Vilniaus universitetas
TARPTAUTINIŲ SANTYKIŲ IR POLITIKOS MOKSLŲ
INSTITUTAS

POLITIKOS IR MEDIJŲ MAGISTRO PROGRAMA

SIMONAS BANSEVIČIUS
II kurso studentas

VERTYBINĖS MAŠININIO MOKYMOSI INŽINERIJOS ĮTAMPOS: PROGRAMUOTOJŲ POŽIŪRIO TYRIMAS

MAGISTRO DARBAS

Darbo vadovė: lekt. Jūratė Kavaliauskaitė

Vilnius, 2025 m.

BIBLIOGRAFINIO APRAŠO LAPAS

Bansevičius S. Vertybinės mašininio mokymosi inžinerijos įtampos: programuotojų požiūrio tyrimas: Politikos ir medijų specialybės, magistro darbas / VU Tarptautinių santykių ir politikos mokslų institutas; darbo vadovė J. Kavaliauskaitė, 2025 m. – 64 p., spaudos ž. (su tarpais) 129 806.

Reikšminiai žodžiai: dirbtinis intelektas, mašininio mokymosi inžinerijos įtampos, technokratinė galia, duomenizavimas, dirbtinio intelekto etika, algoritmai kaip kultūra, sociotechninė visuomenė, mašininis mokymasis, inžinerijos vertybinės įtampos, programuotojų etika.

Šiame darbe, per inžinerinių procesų analizę tiriama, kaip kyla mašininio mokymosi programų poveikis visuomenei. Šių programų poveikiui analizuoti jų inžinerija tiriama kaip kultūrinis procesas, kurį sudaro įvairūs konfliktai, lemiantys galutinį šių programų poveikį visuomenei. Darbe pristatomi mašininio mokymosi programuotojų patiriami sprendimų įtampos taškai, kokiomis sampratomis ir vertybėmis jie orientuojasi priimdami sprendimus bei interpretuojama šių sprendimų įtaka mašininio mokymosi programų socialiniam poveikiui.

TURINYS

TRUMPINIŲ ŽODYNAS

ML - mašininio mokymosi programa (angl. „machine learning“);

LLM - didieji kalbos modeliai (angl. „large language models“). ML programų rūšis;

DI - dirbtinio intelekto programa.

ĮVADAS

Pastarąjį dešimtmetį vis gausėja teigiančiųjų, kad gyvename jau 4-osios pramonės revoliucijos laikais[1]. Vienas esminių šios pramonės techninių laimėjimų yra duomenų mokslas bei dirbtinis intelektas[2]. Šios technologijos apima „tiek ekonomikos, tiek verslo, visuomenės ir paties žmogaus paradigmų pokyčius“[3], bet kartu ir „valstybių, įmonių, pramonės sektorių ir visos visuomenės sistemų pertvarką“[4]. Taigi, dirbtinio intelekto technologijos yra naujas dėmuo politiniame kontekste, darantis pastovią įtaką žmonių elgesio pasirinkimams kartais net ir globaliu mastu.

Kaip bebūtų, nors diskusijų apie dirbtinį intelektą pastaraisiais metais netrūksta, retas kuris gerai supranta, kaip šios programos veikia iš tikrųjų. Viena dažniausių dirbtinio intelekto rūšių yra mašininio mokymosi programos[5] (toliau ML, pagal angl. „Machine learning“). ML programos kuria statistines prognozes, paremtas automatiškai duomenyse aptinkamomis tendencijomis[6]. Tai yra praktiškai panaudojama statistika.

Remiantis gausiais duomenimis, programa apskaičiuoja statistiškai labiausiai tikėtinas duomenų prognozes. Šios prognozės, siauresniu ar platesniu mastu, daro įtaką individams ar visuomenei – nuo pirkinių pasirinkimų rekomendacijų ir socialinių tinklų aplinkinių rato formavimo iki stebimo politinio turinio parinkimo, teisinių argumentų, policijos darbo asistavimo ir pan.

Tyrimo problema. Spartėjanti ML technologinė plėtra reiškia algoritmizuoto socialumo plėtimąsi į vis įvairesnes gyvenimo sferas. Šia problemas yra susirūpinta, tačiau nepakankamai. Pilnam ML socialinio poveikio suvokimui būtina pažinti ne tik šių programų rezultatų poveikį, bet ir jų priežastis – kokie vidiniai ML kūrimo procesai sukuria atitinkamą poveikį naudotojams.

Apie tai kalbėti nėra paprasta, nes ML yra inžineriškai kompleksiška ir be tinkamo išsilavinimo, sunkiai suprantama technologija. Kas viena vertus ją apipina įvairiais mitais[7], kurie trukdo realiai matyti

[1] Heiner Lasi ir kt., „Industry 4.0“, *Business & Information Systems Engineering* 6, nr. 4 (2014 m. rugpjūčio): 239–42, https://doi.org/10.1007/s12599-014-0334-4.

[2] Cristina Orsolin Klingenberg, Marco Antônio Viana Borges, ir José Antônio Valle Antunes Jr, „Industry 4.0 as a Data-Driven Paradigm: A Systematic Literature Review on Technologies“, *Journal of Manufacturing Technology Management* 32, nr. 3 (2019 m. birželio 21 d.): 586–87, https://doi.org/10.1108/JMTM-09-2018-0325.

[3] „Pramonė 4.0 - Lietuvos Respublikos ekonomikos ir inovacijų ministerija“, žiūrėta 2024 m. kovo 20 d., https://eimin.lrv.lt/lt/veiklos-sritys/pramone/pramone-4-0/.

[4] „Pramonė 4.0 - Lietuvos Respublikos ekonomikos ir inovacijų ministerija“.

[5] Alex Castrounis, *AI for People and Business: A Framework for Better Human Experiences and Business Success*, 2019, 53.

[6] Castrounis, 53.

[7] Nick Seaver, *Computing Taste: Algorithms and the Makers of Music Recommendation* (University of Chicago Press, 2022), 19, https://doi.org/10.7208/chicago/9780226822969.001.0001.

šių programų tikrąjį poveikį. O kartu, ją suprantantys inžinieriai retai turi pakankamai socialinių mokslų žinių, kad suprastų savo kuriamų programų socialinį poveikį.

Be to, egzistuoja paradoksas – kalbama apie ML galimybę kurti diskriminaciją, išnaudojimą ir kitokias socialines grėsmes, tačiau iš ML programų tikimasi inžinerinio objektyvumo, kadangi formaliai jos kuriamos remiantis faktiškai apibrėžiamais matematiniais vertinimais. Šis faktiškumas atrodo priešingas normatyviems pasirinkimams, kaip pavyzdžiui išimtims vienos socialinės grupės atžvilgiu prieš kitas.

To priežastis – ML veikia remiantis ir duomenyse aptiktomis tendencijomis. Duomenų dėmuo šiai problemai suteikia socialinio poveikio klausimą bei papildomas sprendimų priėmimo įtampas. Kurių asmenų duomenys turėtų būti gaunami, kurį naudotojų elgesį verta ar neverta paversti duomenimis, ar visus duomenis galima gauti, galiausiai kokias išvadas padaryti iš šių duomenų? Nuo to, kaip bus atsakyta į pastaruosius klausimus, priklauso tai, kokį poveikį naudotojui darys tam tikras ML dizainas. Į šiuos duomenizavimo klausimus atsako programuotojai.

Asmenų, kaip mechaninių įpročių, kuriuos galima kreipti saviems tikslams pasiekti suvokimas yra laikomas technokratija[8]. Tuomet ML taip pat gali būti laikoma technokratine galios forma, kadangi per duomenis šios technologijos turi galią veikti naudotojų pasirinkimus ir elgesį bei algoritmiškai formuoti individų ar visuomenių elgesį[9]. Atitinkamai ir IT industriją galima laikyti tam tikra technokratine galia. Tačiau būtina pastebėti, kad šį įpročių poveikį formuoja ne tik IT verslų valdytojai, bet ir patys programuotojai. Kiekviename ML kūrimo žingsnyje skirtingi technologiniai sprendimai daro vis kitokį poveikį ML prognozei, o galų gale – ir poveikiui naudotojui.

Tad problema yra ši – ML socialinis poveikis visuomenei ir asmeniui yra realus, kadangi ši technologija remiasi asmenų įpročių formavimu, tačiau kaip šie socialinę įtaką formuojantys procesai veikia iš tikrųjų, teisininkai, politikai, socialinių mokslų atstovai bei kiti tuo susirūpinę asmenys supranta nepakankamai ar tik paviršutiniškai, trūksta socialinių ML praktikų tyrimų, kadangi ML procesai yra kompleksiški ir reikalauja inžinerinių žinių. Šias žinias turi patys ML programuotojai, tačiau būtent jiems trūksta socialinių mokslų žvilgsnio, galinčio padėti suvokti platesnį ML poveikį visuomenei.

[8] Jathan Sadowski ir Evan Selinger, „Creating a Taxonomic Tool for Technocracy and Applying It to Silicon Valley“, *Technology in Society* 38 (2014 m. rugpjūčio): 164, https://doi.org/10.1016/j.techsoc.2014.05.001.

[9] Henrik Skaug Sætra, „A Shallow Defence of a Technocracy of Artificial Intelligence: Examining the Political Harms of Algorithmic Governance in the Domain of Government“, *Technology in Society* 62 (2020 m. rugpjūčio): 4, https://doi.org/10.1016/j.techsoc.2020.101283.

Tad pati ML technologija yra tarpdisciplinišką. Tai reiškia, jog nors ir kuriamos dirbtinio intelekto etikos gairės[10] bei ES įstatymai[11], nors ML verslai gali skatinti algoritmų skaidrumą, tai yra daugiau nei socialinės įtakos veikėjas. Neužtenka vien riboti ML, bet reikia ir suprasti ML socialinio poveikio priežastis. Pripažinti, kad patys ML kūrimo procesai yra ne vien inžineriniai, bet ir paremti socialinio gyvenimo vertybėmis. O šių vertybių įgyvendinimas bene smarkiausiai priklauso nuo pačių programuotojų ir to, koks yra jų supratimas, kas yra geras ML algoritmas.

Analitinė literatūros apžvalga. Darbe analizuota literatūra apima algoritminės visuomenės temas, sociologines duomenų analizes, antropologinį žvilgsnį į ML. Taip pat, analizuojamos ML etikos literatūros apžvalgos tiek socialinių, tiek glaustu techninių mokslų žvilgsniu. Galiausiai, apžvelgiami tyrimai atlikti su pačiais ML programuotojais.

ML kaip politinę sistemą kuriantis įrankis. Buvusi ML algoritmų progrogramuotoja Cathy O'Neil, knygoje „Weapons of math destruction" ML analizuoja kaip technologiją dėl kurios stiprieji visuomenės nariai pasipelno ir tampa dar stipresniais, o silpnieji tampa dar labiau išnaudojami[12]. Autorė pabrėžia, jog ML algoritmai neturi kompleksiško pasaulio supratimo, bet yra linkę duoti pirmumą efektyvumui[13]. Dėl to algoritmai neatsižvelgia į lygybės vertybes, o tik efektyvumą. Pabrėžiama problema, jog ML sukuriami rekomenduoti tam tikro lokalaus konteksto situaciją, tačiau šie algoritmai naudojami įvairiuose kontekstuose ir dėl to, kiti kontekstai turi prisitaikyti prie šių algoritmų skatinamų elgesio normų[14].

Pateiktas pavyzdys, kaip JAV universitetų reitingavimo algoritmas veikė pagal aukščiausio lygio universitetų vertinimus. Tačiau programa buvo naudojama reitinguoti ne vien prestižiniams, bet ir vidutinio ar žemiausio lygio universitetams. Rezultatas – prestižiniai universitetai ir liko viršuje, tačiau vidutiniai ir žemesnio lygio gavo tokius žemus įvertinimus, kad pastarieji pradėjo prarasti naujus studentus, finansavimą ir susidarė atskirtis. Algoritmo problema tame, jog jis atsižvelgė tik į aukščiausios klasės studentų poreikius, bet ne į vidutinės ar žemesnės klasės studentų poreikius (pavyzdžiui universiteto studijų įperkamumas)[15]. Autorės požiūryje galima įžvelgti klasių kovos sampratos ir į ML technologiją neretai autorė žvelgia kaip į įrankį, kurios tikslas yra išnaudoti klasių skirtumus.

Wendy Hui Kyong Chun knygoje „Updating to Remain the Same: Habitual New Media" naudoja naujųjų medijų sąvoką, apibrėžiančią interneto technologijas, ir teigia, jog šios medijos daro didžiausią

[10] UNESCO, „Rekomendacija dėl dirbtinio intelekto etikos", 2021 m. 23.
[11] Europos Komisija, „Dirbtinio intelekto aktas", 2024 m. 13.
[12] Cathy O'Neil, *Weapons of Math Destruction,* s.a., 78–81, 92.
[13] O'Neil, 94.
[14] O'Neil, 78–81.
[15] O'Neil, 52.

poveikį naudotojų pasirinkimams, kai tampa dalimi mūsų kasdienybės įpročių, kaip pavyzdžiui išmanieji telefonai tapo mūsų kasdienybės dalimi[16]. Tai sukuria erdvę, kur naudotojo privatūs veiksmai, dėl to, kad yra atliekami skaitmeninėje erdvėje, tampa viešais[17]. Galiausiai autorė mano, jog naujosios medijos, kurdamos naudotojų įpročius ir šią privataus gyvenimo dalį perkeldamos į viešą, stipriną neoliberalistinę sistemą[18]. Kitoje knygoje autorė plečia minimą teoriją kalbant apie medijų naudotojų rolių įvairumą[19]. Galiausia jau kiti autoriai tai išplečia kalbant apie naujas visuomenės formas, kai įpročių skaitmenizavimas kuria „dividų" individualizaciją - vienas asmuo reprezentuojamas įvairiais atžvilgiais[20].

Duomenizacijos analizės literatūra. Svarbus bet kokių ML algoritmų kūrimo dėmuo yra naudotojų elgesio duomenys, pagal kuriuos programa nuspėja, kokį rezultatą naudotojui pateikti. Visuomenės reprezentavimo duomenyse problematiką analizavo Gilles'as Deleuze'as pristatydamas prieš tai minėtą divido sąvoką. Deleuze'as teigė, jog Michelio Foucault pristatyta disciplinos visuomenė pavirto „kontrolės" visuomene[21]. Dividas, tai asmens reprezentacija tam tikrame konkrečiame gyvenimo kontekste[22] - asmuo kaip įmonės darbuotojas, kaip banko klientas, ar šiais laikais kaip socialinės medijos naudotojas ir pan. Šiomis dienomis ši teorija plačiai pritaikoma kalbant apie duomenų mokslą[23], bei jo įtaką naujoms asmens sampratos formoms[24].

ML programuotojų analizės literatūra. Kaip matoma ML politinis paveikumas yra pastebėtas. Tačiau dažnai šis politinio poveikio potencialas grindžiamas ML analizuojant kaip sisteminės politinės struktūros elementą ar pačiame ML ieškant techniniuose principuose slypinčių politinių jėgų, bet reta analizė atsižvelgia į pačius programuotojus.

Nick'as Seaver'is yra vienas iš nedaugelio autorių, tyrusių būtent pačius ML programuotojus. Knygoje „Computing Taste: Algorithms and the Makers of Music Recommendation" autorius aprašo antropologinį tyrimą, kurio metu Seaver'is vykdė interviu su JAV muzikos rekomendacinių algoritmų programuotojais. Pats autorius knygoje teigia, jog kalbant apie ML, „mums netrūksta paaiškinimų, kurie

[16] Chun Wendy Hui Kyong, *Updating to Remain the Same: Habitual New Media*, s.a., 11.
[17] Wendy Hui Kyong, 89.
[18] Wendy Hui Kyong, 16.
[19] Wendy Hui Kyong Chun ir Alex H. Barnett, *Discriminating Data: Correlation, Neighborhoods, and the New Politics of Recognition* (Cambridge, Massachusetts: The MIT Press, 2021), 252–250.
[20] Celia Lury ir Sophie Day, „Algorithmic Personalization as a Mode of Individuation", *Theory, Culture & Society* 36, nr. 2 (2019 m. kovo): 31, https://doi.org/10.1177/0263276418818888.
[21] GIlles Deleuze, „Postscript on the Societies of Control", 1992 m., 3–5.
[22] Deleuze, 6.
[23] James Brusseau, „Deleuze's Postscript on the Societies of Control Updated for Big Data and Predictive Analytics", *Theoria* 67, nr. 164 (2020 m.).
[24] Jenna Burrell ir Marion Fourcade, „The Society of Algorithms", *Annual Review of Sociology* 47, nr. 1 (2021 m. liepos 31 d.): 227–28, https://doi.org/10.1146/annurev-soc-090820-020800.

kalte verstų kapitalizmui“[25]. Ir nors jis neteigia, kad šie paaiškinimai būtų klaidingi, jis rašo, kad tokie paaiškinimai „neatskleidžia su šiomis sistemomis dirbančių žmonių kasdienių argumentų“[26]. Dėl to, šioje knygoje jis siekia suprasti, kaip ML kūrėjai suvokia savo kuriamas programas ir kam jos yra reikalingos[27].

Kitame darbe, N. Seaver‘is aptaria etnografinius algoritminių sistemų tyrimo metodus. Pripažįstama, jog egzistuoja tam tikra dichotomija tarp algoritmus kuriančiųjų bei jų poveikį analizuojančiųjų. Skirtis, kurioje IT inžinieriai algoritmus mato tik kaip technologinį objektą, o algoritmų kritikai, šias programas kaip grėsmę kultūrai[28].. Seaveris teigia, jog ši skirtis remiasi klaidinga prielaida, kad technologija ir kultūra yra atskiros[29]. Autorius siūlo pripažinti, kad algoritmai nėra tiesiog kultūrą veikiantys objektai, bet patys yra sukurti iš kolektyvių žmonių veiksmų (t.y. programuotojų pasirinkimų). Tad algoritmų kūrimas savaime yra kultūrinis veiksmas[30]. Tai pripažinus, antropologiniams algoritmų tyrimams Seaveris siūlo analizuoti, kaip per algoritmus materializuojasi vertybės bei kultūrinės prasmės[31].

Tačiau toks žvilgsnis dar yra iš ties naujas. Nors Seaverio algoritmų kaip kultūros panaudojimų gausėja tiriant DI naudojimą teisinėse programose[32], sveikatos srityje[33], tiesiogiai ML kūrime[34] ir pan. srityse, vis dėlto ML kaip kultūros analizių dar nėra daug, o ypač tokių, kurios analizuotų būtent pačių programuotojų vertybes – taip, kaip ir siūlo pats N. Seaver. Kas rodo neištirtą, bet žadančią nišą.

ML inžinerijos problemų apžvalga. Dažnos ML inžinerijos problemos: verslo problemos suvokimas, duomenų gavimas, apdorojimas[35]. Tai įtraukia ir kitus ML kūrimo veikėjus, kaip verslą, ne ML programuotojus ir t.t. Taip pat daug literatūros vertina ir lygina įvairius ML kūrimo metodus. 2011-

[25] Seaver, *Computing Taste*, 29.
[26] Seaver, 29.
[27] Seaver, 29.
[28] Nick Seaver, „Algorithms as Culture: Some Tactics for the Ethnography of Algorithmic Systems“, *Big Data & Society* 4, nr. 2 (2017 m. gruodžio): 8, https://doi.org/10.1177/2053951717738104.
[29] Seaver, 8.
[30] Seaver, 11.
[31] Seaver, 11.
[32] Sem Nouws ir Roel Dobbe, „The Rule of Law for Artificial Intelligence in Public Administration: A System Safety Perspective“, *Digital Governance*, sud. Kostina Prifti ir kt., t. 39, Information Technology and Law Series (The Hague: T.M.C. Asser Press, 2024), 194, https://doi.org/10.1007/978-94-6265-639-0_9.
[33] Javier Guerrero-C ir kt., „Brave Global Spaces: Researching Digital Health and Human Rights through Transnational Participatory Action Research“, *Journal of Responsible Technology* 20 (2024 m. gruodžio): 6, https://doi.org/10.1016/j.jrt.2024.100097.
[34] Emily Wanderer, „Bearly Recognizable: Facial Recognition and the Wild“, *Science, Technology, & Human Values*, 2024 m. gruodžio 15 d., 4, https://doi.org/10.1177/01622439241304141.
[35] Elizamary De Souza Nascimento ir kt., „Understanding Development Process of Machine Learning Systems: Challenges and Solutions“, *2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)* (2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), Porto de Galinhas, Recife, Brazil: IEEE, 2019), 5–6, https://doi.org/10.1109/ESEM.2019.8870157.

2021 m. rašytų techninių ML mokslinių straipsnių apžvalgoje matoma, jog kuriant ML kyla daug techninių ribotumų bei iššūkių, kurių kiekvienam spręsti yra keletas sprendimų. Taip pat pasirinkti, kuris sprendimas yra tinkamiausias – sudėtingas uždavinys, nes kiekviena alternatyva turi savų pranašumų, bet ir problemų[36]. Apžvalga rodo, kad ML kūrime nėra vyraujančios metodikos, bet daug techninių sprendimų[37]. Taip pat pastebėta, jog nėra vieno universalaus ML efektyvumo vertinimo[38]. Tai parodo, jog ML kūrimas kiekvienoje aplinkybėje vis kitoks ir renkantis techninį problemos sprendimą, programuotojui dažnai tenka daryti kompromisus. Šių kompromisų sprendimai lemai skirtingą ML galutinį rezultatą, o kartu ir skirtingą socialinį poveikį.

ML etikos literatūra. Sisteminė ML etinių iššūkių analizė išskiria etines problemas: neatitinkamo turinio rekomendacija, privatumas, naudotojo autonomija (ar naudotojas nėra tam tikra prasme manipuliuojamas priimti rekomendacijas), skaidrumas (ar naudotojas supranta, kaip veikia algoritmas) ir teisingumas (kiek šališki yra algoritmai)[39]. Išskiriamos šios ML interesų grupės bei jų interesai: ML kūrėjai (siekia kurti ML), pačios ML (tikslas kurti rekomendacijas), naudotojai (siekis gauti ir naudotis rekomendacijomis) bei visuomenė[40]. Autoriai pabrėžia, kad tuometinėje ML etikos literatūroje daugiausia dėmesio skiriama ML poveikiui naudotojams individualiai, tačiau mažai dėmesio skiriama kitoms grupėms kaip pačių ML kūrėjai ar visuomenė plačiąja prasme[41]. Ir pastebima, jog privatumo, teisingumo ir skaidrumo etinės problemos dažniausiai randamos techninėje ML literatūroje, tuo tarpu socialinio paveikumo etinės problemos, kaip manipuliatyvumas ir autonomija, dažniau pastebimos socialinių, humanitarinių mokslų ML tyrimuose[42].

Techninė ML iššūkių literatūra. Kaip minėta, įvairioms ML sprendžiamoms problemoms egzistuoja keletas skirtingų sprendimo būdų ir kiekvienas iš šių sprendimo būdų turi savų pranašumų, bet ir savų trūkumų[43]. Pastebėta, jog techninė ML iššūkius analizuojanti literatūra neretai siūlo dėl ML atsirandančia etines problemas tiesiog spręsti kitais techniniais sprendimais. Pavyzdžiui ML problemoms spręsti programos viduje panaudoti dar vieną ML algoritmą ar pan.[44].

---

[36] Deepjyoti Roy ir Mala Dutta, „A Systematic Review and Research Perspective on Recommender Systems“, *Journal of Big Data* 9, nr. 1 (2022 m. gruodžio): 1, https://doi.org/10.1186/s40537-022-00592-5.

[37] Roy ir Dutta, 33.

[38] Roy ir Dutta, 33.

[39] Silvia Milano, Mariarosaria Taddeo, ir Luciano Floridi, „Recommender Systems and Their Ethical Challenges“, *AI & SOCIETY* 35, nr. 4 (2020 m. gruodžio): 960–63, https://doi.org/10.1007/s00146-020-00950-y.

[40] Milano, Taddeo, ir Floridi, 958–59.

[41] Milano, Taddeo, ir Floridi, 966.

[42] Milano, Taddeo, ir Floridi, 965.

[43] Roy ir Dutta, „A Systematic Review and Research Perspective on Recommender Systems“, 1.

[44] Fjolla Berisha ir Eliot Bytyçi, „Addressing Cold Start in Recommender Systems with Neural Networks: A Literature Survey“, *International Journal of Computers and Applications* 45, nr. 7–8 (2023 m. rugpjūčio 3 d.): 485, https://doi.org/10.1080/1206212X.2023.2237766.

ML programuotojų poveikis sprendimų priėmimui. Kaip bebūtų, dalis ML programuotojų pastebi šių algoritmų socialinį poveikį ir iš to kylančias etines problemas, tačiau kartais, imtis veiksmų jas spręsti, trukdo pačios įmonės. Vertybių konfliktus tarp technologinių kompanijų bei jų darbuotojų tiriančios literatūros apžvalga atskleidžia, kokie etiškumo išsaugojimo žingsniai praktikuojami IT versle bei kaip juos vertina individualūs šių įmonių darbuotojai[45]. Nors rasti apklausų su ML programuotojais tyrimai rodo, kad dalis ML programuotojų yra susirūpinę dėl etinių šių programų problemų[46], programuotojams dažnai vis tiek yra sunku pastebėti subtiliais ir ne tokias dažnas ML etikos problemas[47]. Be to ir pačios įmonės kartais gali būti trukdžiu etiškam ML vystymui, nes literatūroje pastebėta nuomonė, jog tikėtina, kad ML etikos priemones verslai įgyvendins tik tada, jei tai nebus ekonominiu trukdžiu, taip pat, ML įmonės pastebėtos tik kaip reaguojančios į teisinius ML ar technologijų reikalavimus, bet ne kaip aktyviai to siekiančios[48]. Nors ML etikos pritaikymo versle literatūros yra, dažnai ši literatūra akcentuoja teisinį ML etikos įgalinimą, bet retai kreipiamas dėmesys į pačius ML programuotojus, tad apžvalgos autoriai skatina daugiau dėmesio skirti pačių programuotojų etinių įžvalgų tobulinimui[49].

Tyrimo niša. Analizuota literatūra, tirianti ML kūrimo procesus, programuotojų aplinką jų kasdienybėje bei kaip žvelgti į ML sistemų socialinį poveikį, parodė, jog šių technologijų politinis paveikumas yra pastebėtas. Tačiau jis dažnai matomas, kaip kylantis iš egzistuojančios politinės sistemos ar pačios technologijos savaime, tačiau mažai dėmesio skiriama patiems programuotojams ir ML inžinerijos praktikai. Programuotojams dėmesys turėtų būti skiriamas, nes kiekvienas jų techninis pasirinkimas prisideda prie to, kokį poveikį naudotojo pasirinkimams darys algoritmas ir galiausiai, kokį politinį poveikį tai sukurs per įtaką naudotojų elgesio įpročiams kaip individams bei visuomenei. Taigi, bet koks darbas tiriantis ML programuotojų techninių sprendimų įtaką socialiniam poveikiui jau yra aktualus ir liečiantis mažai tirtą, nes ne taip intuityviai pastebimą, tačiau dėl to ypač aktualią nišą.

Arčiausiai ši problema yra pastebėta N. Seaverio darbuose. Seaveris siekia išsiaiškinti, kas programuotojų žvilgsniu yra ML rekomendacija. Tačiau jo darbas nekalba apie tai, kaip programuotojai supranta, kas yra geras ML algoritmas. Bet tai yra svarbu, nes kitame tekste pats autorius pripažįsta, jog programuotojai darbe daro vertybinius sprendimus[50], o šių vertybių pažinimas skatina ištirti algoritmų kūrimo praktikas.

[45] Mark Ryan ir kt., „An AI Ethics 'David and Goliath': Value Conflicts between Large Tech Companies and Their Employees", *AI & SOCIETY* 39, nr. 2 (2024 m. balandžio): 558, https://doi.org/10.1007/s00146-022-01430-1.
[46] Ryan ir kt., 560.
[47] Ryan ir kt., 560.
[48] Ryan ir kt., 560.
[49] Ryan ir kt., 569.
[50] Seaver, „Algorithms as Culture", 9.

Programuotojai sprendimus priima gausiose komandose bei siekia atliepti skirtingų asmenų poreikius. Būtent dėl to ML kūrimas yra kultūrinis procesas[51]. Daugybė skirtingų veikėjų – programuotojai, verslo užsakovai, naudotojai ir pan. – turi skirtingus įsivaizdavimus, kaip turi atrodyti gera ML programa. Tačiau technologinius sprendimus atlieka tik programuotojai. Be to, vertybinių ML vertinimo žvilgsnių programuotojai turi įvairių. Kaip rodo literatūros analizė, viena vertus, programuotojai siekia gerųjų ML inžinerijos praktikų. Kita vertus, dažnai skirtingi techniniai sprendimai prieštarauja vienas kitam, kas reikalauja programuotoją rinktis tarp kelių pasirinkimų. Galiausiai, programuotojo tikslai turi derėti ir su verslo užsakovo, naudotojo bei kitų asmenų lūkesčiais, o kaip rodo etinių kivirčų literatūra, šį sutarimą kartais rasti sunku.

Taigi, ML kūrimas yra kultūrinis veiksmas, apimanti daugybės skirtingų veikėjų tikslus bei lūkesčius, kaip turėtų veikti galutinis ML produktas. Šie lūkesčiais neišvengiamai atsiduria įtampoje, kadangi technologinis sprendimas konkrečiu atveju gali būti priimtas tik vienas. Ir šį sprendimą galiausiai atlieka programuotojai.

Darbo tikslas: rasti įtampų taškus ML inžinerijos praktikoje – sprendimų priėmimo momentus, kuriuose programuotojai turi rinktis tarp kelių technologinių sprendimų, bei atskleisti, kaip ir kokiomis vertybėmis jie remiasi darydami šiuos pasirinkimus.

Uždaviniai:

1. Suformuoti tyrimo teorinį pagrindą: išnagrinėti pagrindinius ML inžinerijos procesus, juose priimamų sprendimų alternatyvas ir jų tarpusavio įtaką bei įtampos taškus.
2. Apibrėžti tyrimo metodą ML inžinerijos įtampų analizei, tyrimo dalyvių atrankos bei interviu vykdymo gaires.
3. Atlikti pusiau struktūruotus giluminius interviu – ištirti programuotojų patiriamus ML inžinerijos įtampos taškus bei kokiomis vertybėmis ir sampratomis jie orientuojasi priimdami sprendimus.
4. Išanalizuoti tyrimo rezultatus ir pristatyti rastus ML inžinerijos įtampos taškus bei juose veikiančias programuotojų vertybines nuostatas.

Tyrimo metodas: pusiau struktūruoti giluminiai interviu su ML programuotojais, dirbančiais įvairiose kompanijose ir kuriančiais skirtingų paskirčių sprendimus.

Darbo struktūra: Pirmiausia pristatomas tyrime naudojamas teorinis įrankis, kuris yra klausimyno sudarymo pagrindu ir interviu analizavimo įrankiu. Tuomet pristatomi tyrimo rezultatai – ML duomenizavimo (angl. „datafication“) bei vertinimo procesuose kylančios sprendimų priėmimo

[51] Seaver, 11.

įtampos bei pastebėtų programuotojų ML vertinimo žvilgsnių analizė (jų konfliktai, vidinės prieštaros). Galiausiai, pateikiamos išvados.

1. ML INŽINERIJOS VERTYBINĖS ĮTAMPOS: TEORIJA

1.1. Algoritmų kaip kultūros apibrėžimas

Verta nuodugniau apibrėžti, ką N. Seaveris turi omenyje, skatindamas į algoritmus žvelgti kaip į kultūrą. Jis pradeda pristatydamas dilemą, jog tai, kas yra programa ar algoritmas, skirtingų disciplinų atstovai supranta įvairiai. Tai yra centrinė jo teksto problema. Dėl to, autorius pateikia ne apibrėžimą, kas yra algoritmas (kadangi universalaus, visoms disciplinoms prieinamo apibrėžimo turbūt ir nepavyktų rasti), o būdą, apie juos galvoti – etnografinį algoritmų tyrimo metodą, leidžiantį algoritmų kūrimą patirti praktiniu žvilgsniu[52].

Pirmiausia Seaveris analizuoja „algoritmų kultūroje“ žvilgsnį, paremtą prielaida, jog technologijos ir kultūra yra atskiros, bene priešingos sritys[53]. Toks žvilgsnis remiasi mąstymu, jog algoritmas yra apibrėžtinas objektas, tarsi veikėjas[54]. Tačiau autoriaus pokalbiai su programuotojais parodo, kad algoritmas yra „kolektyvinis produktas“, kurį yra sunku apibrėžti konkrečiai ir specifiškai[55]. Kitaip tariant, Seaveris parodo, jog algoritmas (taip pat ir ML) nėra tiesiogiai lengvai apibrėžiamas matematinių formulių rinkinys. Algoritmus kuria daugybė žmonių, su savais tikslais ir motyvais, dėl to tokios sudėtingos programos kaip ML yra greičiau ne konkrečiai suvokiami objektai, bet išsisklaidę, daugelį veikėjų apimantys procesai[56].

Tad Seaveris pateikia alternatyvų žvilgsnį – siūlo algoritmus matyti ”kaip kultūrą“. Kultūrą ta prasme, jog algoritmų kūrimas yra padrikai sujungtų praktikų (pavyzdžiui sociotechninių pasirinkimų) rinkinys. Kiekviena šių praktikų įgyvendina tam tikrą tikslą – kultūrinį (kaip pelnas) arba techninį. Atitinkamai, kai kurios praktikos dera tarpusavyje, o kai kurios priešinasi viena su kita dėl savo galutinio tikslo[57]. Pritaikant tai ML kūrime, galima sakyti, jog ML inžinerija yra paremta pastovia vertybine kova – konfliktais ir kompromisais dėl skirtingų ML techninių ar kultūrinių sprendimų.

[52] Seaver, 4.
[53] Seaver, 8.
[54] Seaver, 8.
[55] Seaver, 6.
[56] Seaver, 11.
[57] Seaver, 9.

1.2. ML sprendimų priėmimo taškų apžvalga

Norint atrasi ML inžinerijos įtampų taškus, reikia apibrėžti ML kūrimo praktikas. Tam neapsieinama be glausto ML technologijos paaiškinimo, kadangi tik susipažinus su pagrindiniais ML veikimo principais įmanoma kurti efektyvų klausimyną, leidžiantį ištirti, tarp kokių inžinerinių alternatyvų bei kokiomis galimomis vertybėmis remiasi programuotojai.

ML programas galima išskirti į dvi dalis – duomenis bei pagal duomenis statistines prognozes kuriančią programos dalį vadinamą ML modeliu[58]. Atitinkamai ir ML kūrimo praktikos apima du pagrindinius procesus: duomenizavimą bei modelio kūrimą.

1.2.1 Duomenizavimo alternatyvų įtampos

Darbas su duomenimis – tai procesai, tokie kaip duomenų apie naudotojus surinkimas, gautų duomenų apdorojimas ar koregavimas[59]. Darbas su duomenimis aktualus, kadangi norint kiekybiškai užkoduoti tam tikrą asmens elgesį, reikia sukurti pasaulio suvokimo abstrakciją[60]. Ši abstrakcija, tiek riboja supratimą apie žmogų, tiek kartu ir sukuria naujas supratimo apie asmenį formas.

Duomenų gavimo procesai. Norint gauti duomenis apie naudotoją, programa kuriama taip, kad tam tikras elgesio fenomenas galėtų būtų užfiksuojamas kiekybiškai. Duomenys patys savaime neegzistuoja – jie yra sukuriami per abstrakcijų sukūrimą[61]. Verta paminėti, jog net egzistuoja diskusijų, kad duomenų pavadinimas „data“ turėtų būti peržvelgtas. Diskusija remiasi tuo, jog „data“ yra paremtas idėja, kad duomenys egzistuoja savaime (*lot. datum* - tai, kas duota[62]). Kaip bebūtų, neretai per šį abstrahavimo procesą duomenys veikiau yra sukuriami nei randami, kas diskutuojančiuosius net skatina mąstyti apie naujas duomenų sąvokas[63]. Ši diskusija parodo, kaip svarbu yra analizuoti duomenų gavimo/kūrimo procesus.

Kuriant ML, asmenų abstrahavimą į duomenys atlieka tiek programuotojai, įvairiai modifikuojantys ir apdorojantys duomenis, tiek ML programa, skaičiuojanti prognozes pagal šiuos duomenis, tiek vartotojo sąsaja, kuria naudojantis, asmenys gauna ML rezultatus, o kartu, jų elgesys šioje sąsajoje yra verčiamas duomenimis ir panašiai. Tad programuotojai programas kuria taip, kad žmonės tam tikrus, su rekomendacijos duomenimis susijusius sprendimus, darytų naudodamiesi ML teikiančia

[58] Agnė Paulauskaitė-Tarasevičienė ir Kristina Šutienė, *Intelektikos pagrindai* (Kaunas : Technologija, 2022), 14–19.
[59] Paulauskaitė-Tarasevičienė ir Šutienė, 91–117.
[60] Ulises A. Mejias ir Nick Couldry, „Datafication“, *Internet Policy Review* 8, nr. 4 (2019 m. lapkričio 29 d.): 2, https://doi.org/10.14763/2019.4.1428.
[61] Mejias ir Couldry, 2.
[62] „Definition of DATUM“, 2025 m. sausio 3 d., https://www.merriam-webster.com/dictionary/datum.
[63] Johanna Drucker, „Humanities Approaches to Graphical Display“, *Digital Humanities Quarterly* 5, nr. 1 (2011 m.), https://www.proquest.com/docview/2555208513/abstract/B61E4CD96292457EPQ/1.

programa, t.y. programuotojai organizuoja naudotojų elgesio įpročius[64]. Kitaip tariant, jau vien norint gauti duomenis reikalingus ML kūrimui, programa kuriama tokiu būdu, kuris daro įtaką naudotojų elgesiui, sprendimams, o tam tikrais atvejais net ir mąstymo būdams.

Šios naudotojo elgesio abstrakcijos įprastai kuriamos remiantis viena iš dviejų metodikų – akivaizdine arba neakivaizdine. *Akivaizdinis duomenų rinkimas* yra tada, kai programos naudotojas aiškiai žino, jog jo įvertinimas bus užfiksuotas ir išsaugotas kaip duomenys[65]. Pavyzdžiui naudotoja turi užpildyti klausimyną apie jai labiausiai patinkančius filmų žanrus. Šiuo atveju sąmoninga naudotoja gali nesunkiai suprasti, jog nuo jos pasirinkimų priklausys ML teikiamų pasiūlymų turinys. Tačiau dalis programuotojų tokį potencialų naudotojo sąmoningumą rekomendacijos rezultatams mato kaip grėsmę algoritmo kokybei. Manoma, kad naudotojas turėtų kuo mažiau pastebėti algoritmo buvimą, nes toks sąmoningumas pakenktų naudotojo elgesio autentiškumui, kadangi naudotojas darytų sprendimus ne pagal savo poreikius, bet juos pritaikytų algoritmo vertinimui[66].

Alternatyva yra *neakivaizdinis duomenų rinkimas*. Šiuo atveju naudotojas ne pildo klausimyną, bet duomenys surenkami netiesiogiai – naudotojo elgesio su sistema įpročius užfiksuojant kaip kiekybinius duomenis[67]. Pavyzdžiui, skaičiuojama, kiek laiko naudotojas praleidžia naršydamas kiekvieno skirtingo žanro filmų paieškose, ar tam tikrą filmą peržiūrėjo iki galo, o gal išjungė jau filmo pradžioje. Ne taip kaip filmų įvertinimo formos atveju, čia naudotojas nežino, ar/kurie jo pasirinkimai daro įtaką rekomendacijos rezultatui. Tad ML programavimo kultūroje egzistuoja nuomonė, kad neakivaizdinis duomenų rinkimas, dėl jo nepastebimumo, leidžia surinkti duomenis apie naudotoją objektyviau, nei akivaizdinis[68]. Tačiau toks požiūris yra problemiškas dėl kelių priežasčių.

Pirmiausia dėl to, nes toks pasitikėjimas neakivaizdiniu duomenų rinkimu yra paremtas per daug siaura objektyvumo samprata. Ši samprata remiasi prielaida, kad jei naudotojas nežino, kad jo elgesys yra fiksuojamas, tuomet galima laikyti, jog jis elgiasi visai nesuvaržytai ir jokie išoriniai veiksniai nekreipia naudotojo pasirinkimų. Šiuo atveju objektyvumas laikomas autentiškų pasirinkimų užtikrinimu.

Tačiau tai yra klaidinga, nes jau tas faktas, jog asmuo naudojasi programa, yra išorinis veiksnys jo pasirinkimams – nesvarbu, asmuo žino apie jo elgesio vertimą duomenimis ar ne. Juk norint duomenis apie naudotoją surinkti neakivaizdiniu būdu, pačios programos, kuria naudosis asmuo, mygtukai,

[64] Mejias ir Couldry, „Datafication“, 3.
[65] Roy ir Dutta, „A Systematic Review and Research Perspective on Recommender Systems“, 2.
[66] Tyler Reigeluth, „Recommender Systems as Techniques of the Self?“, *Le Foucaldien* 3, nr. 1 (2017 m. rugsėjo 1 d.): 7, https://doi.org/10.16995/lefou.29.
[67] Roy ir Dutta, „A Systematic Review and Research Perspective on Recommender Systems“, 2.
[68] Reigeluth, „Recommender Systems as Techniques of the Self?“, 10.

naršymo juostos ir kiti programos naudojimosi elementai turi būti išdėstyti elgesio sekimui patogia forma. Tai savaime kreipia naudotojus elgtis tam tikru struktūruotu būdu. Minėtuoju filmų pavyzdžiu, skirtingi filmų žanrai gali būti specialiai pateikiami atskirose grupėse tam, kad būtų galima nustatyti, kurio filmų žanro naršyme asmuo praleidžia daugiausiai laiko.

Taip pat įdomu, jog renkant duomenis neakivaizdžiai, sekami net ir nepasirinkimai. Pavyzdžiui, jei socialiniame tinkle yra naudojama ML, socialinio tinklas gali būti suprogramuotas taip, kad galėtų užfiksuoti, ne tik kurios politinės partijos pasisakymus naudotojas skaito, bet ir kurios partijos minčių nesustojo paskaityti net trumpai. Kaip rašo W. Chun, tokiose sistemose naudotojų politinis pasipriešinimas tyla ir neaktyvumu neįmanomas[69]. Asmuo niekada neturi teisės į privatumą, nes net kiekvienas naudotojo nepasirinkimas yra fiksuojamas duomenyse.

Tačiau net jei šis objektyvumas iš tiesų galėtų būti pasiektas, ML kūrimo procesas visada siekia dualistinio natūralumo ir strategiškumo santykio. Toks santykis apibrėžtas analizuojant Google paieškos algoritmą. Kaip rašo D. Caradon, Google siekia kurti savo sistemas taip, kad naudotojas niekada nedarytų savo sprendimų pagal Google sistemą, bet elgtųsi pagal savo natūralius elgesio sistemoje poreikius. Tačiau tuo pat metu Google panaudoja naudotojų elgesių duomenis tam, kad optimizuotų pateikiamas reklamas, kas naudotojo atžvilgiu yra nebe natūrali elgesio erdvė, bet strategiškai reklamos naudai suprojektuota sistema[70].

Atitinkamai yra ir ML projektavime – siekiama sistemas kurti taip, kad naudotojas elgtųsi kuo natūraliau ir neakivaizdžiai surinkti duomenys kuo labiau atskleistų naudotojo poreikius. Tačiau tuo pat metu sistemos yra ir strategiškos, siekiančio iš naudotojo elgesio išgauti duomenis, kurių iki tol net nebuvo galima gauti. Norint tai pasiekti, sistema kuriama taip, kad iki tol programiškai neužfiksuoti įpročiai būtų atliekami per programą, t.y. sistema turi organizuoti naudotojo įpročius[71].

Galiausiai, tai turi įtakos asmens privatumui, bei gali daryti įtaką net visuomenėje vyraujančiai politinei formai. Organizuojant naudotojo įpročius bei juos užfiksuojant duomenyse, įvyksta kai kas nenatūralaus. Asmens pasirinkimų, mąstymo procesas – tai, kas iki šiol buvo privatu, dabar tampa vieša. Tad kaip rašo Chun, tokios sistemos yra neoliberalistinės, nes iškelia privatų gyvenimą į viešumą[72]. Tai ne tik komplikuoja supratimą, kas yra asmens teisė į privatumą, bet ir kuria ekonominių vertybių smelkimąsi į etikos ar net asmenų moralės supratimą.

[69] Wendy Hui Kyong, *Updating to Remain the Same: Habitual New Media*, 29.
[70] Dominique Cardon, „Dans l'esprit du PageRank", 2013 m., 80.
[71] Mejias ir Couldry, „Datafication", 3.
[72] Wendy Hui Kyong, *Updating to Remain the Same: Habitual New Media*, 17.

Taigi, kuriant duomenis, iki šiol privatūs ir kokybiškai vertinami asmenų mąstymo procesai yra verčiami į kiekybiškai apibrėžiamą informaciją. Tai yra abstrahavimo procesas, kuriam sukurti sistemos turi būti projektuojamos taip, kad kreiptų naudotojo pasirinkimus. Galiausiai, šių abstrakcijų kūrimo metu, iki šiol privatūs naudotojų elgesio/pasirinkimų įpročiai yra išviešinami. Kaip bebūtų, nors duomenų kūrimas yra akivaizdžiai besismelkiantis į naudotojo gyvenimą, programavimo kultūroje egzistuoja nuostata, jog neakivaizdinis duomenų rinkimas sukuria natūralaus veikimo erdvę naudotojams[73]. Tačiau būtent pastarasis duomenų rinkimas atrodo neutralus tik paviršutiniškai ir būtent dėl šio apgaulingumo reikalauja gilesnės analizės: kaip kad kodėl programavimo kultūroje vieni duomenizavimo procesai laikomi objektyvesniais, kaip programuotojai supranta šį objektyvumą, kaip programuotojai supranta naudotojų teisę į privatumą.

Duomenų apdorojimo procesai. Duomenizavimo inžinerijos teorija kalba apie papildomą duomenų apdorojimą pagal įvairius korektiškumo poreikius[74]. Atrasti šie duomenų koregavimo pasirinkimų modeliai, paremti konkrečiomis duomenizavimo problemomis.

Pirmasis – trūksta reikalingų duomenų[75]. Hipotetinis pavyzdys – įmonių įdarbinimo programa: sakykime, kad turime programuotojo darbui prašymą teikiančio asmens duomenis, bet juose trūksta informacijos apie asmens lytį. Atrodytų, kad tokiu atveju būtina gauti šiuo naudotojo duomenis, kitaip programa negalės veikti. Tačiau pastebėta, jog šiuo atveju programuotojai problemą sprendžia jau patiems programuotojams ,,nuspėjant“, kokie galėtų būti šie trūkstami duomenys.

Pastebėtas to sprendimo būdas yra trūkstamus duomenis užpildyti vidutiniškai dažniausiai pasikartojančia reikšme[76]. Minėtuoju atveju programuotojas galėtų elgtis taip – kadangi IT rinkoje statistiškai dažniausiai dirba vyrai, tokiu atveju į trūkstamą duomenų vietą būtų įrašoma vyro lytis. Ir nors tai ir atitiktų statistinį vidurkį, toks trūkstamų duomenų užpildymas dar labiau padidintų šansą, kad ML pasiūlys įdarbinti vyrą, o ne moterį ir taip dar labiau sustiprinti statistinį skirtumą bei šališkumą.

Kitas pastebėtas sprendimas – visiškai panaikinti duomenų vienetą, kuriame trūksta kokios nors informacijos[77]. Čia duomenų vienetas – informacija apie vieną asmenį, pavyzdžiui vieno naudotojo lytis, amžius, darbo patirtis ir t.t. Nors šis sprendimas ir neskatintų kokios nors išankstinės nuomonės, kaip pastarasis pavyzdys, tokiu atveju kyla grėsmė iškreipti statistiką, kas lygiai taip pat gali sukurti

[73] Reigeluth, „Recommender Systems as Techniques of the Self?“, 10.
[74] Paulauskaitė-Tarasevičienė ir Šutienė, *Intelektikos pagrindai*, 116.
[75] Paulauskaitė-Tarasevičienė ir Šutienė, 116.
[76] Irfan Pratama ir kt., „A review of missing values handling methods on time-series data“, *2016 International Conference on Information Technology Systems and Innovation (ICITSI)*, 2016, 1–2, https://doi.org/10.1109/ICITSI.2016.7858189.
[77] Paulauskaitė-Tarasevičienė ir Šutienė, *Intelektikos pagrindai*, 116.

šališkumą. Pavyzdžiui, asmenų įdarbinimo ML atveju, pašalinamas duomenų vienetas gali būti būtent tas statistiškai išskirtinis atvejis, kai IT rinkoje dirba moteris.

Tad duomenų apdorojimas programuotojams suteikia pasirinkimų įtaką tarp to, kurie naudotojų įpročiai yra verti duomenizavimo, kaip juos apibūdinti, kuriuos duomenis pažymėti kaip netinkamus ir pan. Neaišku, kada/ar duomenys pritaikomi tik dėl to, kad pavyktų įgyvendinti technologinius procesus, o gal slypi ir programuotojų asmeniniai socialiniai motyvai – kurie duomenys ir veiklos sritys yra vertingesnės už kitas ir kurias problemas spręsti automatizuotai, o kur duomenis dar peržiūrėti patiems?

1.2.2. ML modelio kūrimo alternatyvų įtampos

Prognozavimo algoritmo parinkimas. Kitas svarbus procesas yra ML modelio kūrimas. Modelis – programos dalis duomenyse atrandanti matematines tendencijas ir pagal jas siūlanti galimas prognozes[78]. Prognozės atliekamos remiantis įvairiomis matematinėmis formulėmis, kurių kiekviena yra skirta įvairaus tipo prognozavimo atvejams[79].

Rekomendacijų veikimo teorija lengviausiai suprantama per pavyzdžius ir dažnai minimas pavyzdys yra filmų rekomendacijos. Remiantis turimais duomenimis apie sistemos naudotoją, ML bando nuspėti, kokia tikimybė, kad žmogus pasirinks tam tikrą filmą[80]. Pavyzdžiui jei naudotojas yra teigiamai įvertinęs keletą veiksmo filmų, logiška, kad ML gali vėl rekomenduoti kitą veiksmo filmą. Tačiau, jei naudotojas sistemoje dar nėra įvertinęs nė vieno romantinio filmo, kaip ML prognozuoja, ar naudotojui patinka tokio žanro filmai ar ne? Juk iš pirmo žvilgsnio atrodo nėra jokios statistinės informacijos apie romantinių filmų grupę ir bet kokia tokių filmų „prognozė“ šiame kontekste tebūtų prielaida, o ne tikimybėmis grįsta logika. ML inžinerijos teorija tam turi įvairių sprendimų.

Turiniu paremtos prognozės. Kuriant tokią prognozę, prognozuojamiems objektams yra priskiriami įvairūs bruožai, kaip kad filmų atveju būtų žanras, šalis, režisierius ir pan. Naudotojui teigiamai įvertinus objektą, padidėja rekomendacijos tikimybė kitiems objektams, turintiems panašų bruožą[81]. Minėtuoju pavyzdžiu, jei naudotojas teigiamai įvertino veiksmo filmą, tačiau šio filmo režisierius yra sukūręs ir romantinių filmų, padidėja tikimybė, jog ML rekomenduos šio režisieriaus romantinį filmą. Nors tai ir yra techninis sprendimas, akivaizdu, jog jis remiasi socialiniu supratimu apie tai, kas yra rekomendacija. Šio metodo atveju remiamasi suvokimu, jog rekomendacija yra paremta daiktų tarpusavio panašumu.

Bendradarbiavimo prognozės. Tokios rekomendacijos jau remiasi ne rekomenduojamų objektų, bet šiuos objektus vertinusių asmenų panašumu. Randami naudotojai, kurių elgesio istorijos duomenys panašiausi į esamo naudotojo. Atitinkamai ir rekomenduojami tie produktai, kuriuos dažniausiai renkasi šie panašūs naudotojai[82]. Pavyzdžiui, socialinės medijos naudotoja seka ekologiško gyvenimo būdo turinį. Pagal turimus duomenis ML užfiksuojama, jog panašūs į naudotoją žmonės (t.y. žmonės, kurie taip pat domisi ekologija), seka ir tam tikros politinės partijos veiklą. Tad pirmajai naudotojai ML rekomenduotų tos politinės partijos turinį.

[78] Castrounis, *AI for People and Business: A Framework for Better Human Experiences and Business Success*, 53.
[79] Paulauskaitė-Tarasevičienė ir Šutienė, *Intelektikos pagrindai*, 14–22.
[80] Roy ir Dutta, „A Systematic Review and Research Perspective on Recommender Systems“, 2.
[81] Roy ir Dutta, 3.
[82] Roy ir Dutta, 4.

Šiuo atveju metodas remiasi kitokiu rekomendacijos kaip socialinio veiksmo supratimu, pagrįstu žmonių skirstymu į socialines grupes. Taip pat, pastaruoju pavyzdžiu lengviau pastebėti, jog toks rekomendavimas nebūtinai yra teisingas, nes vien dėl to, kad naudotojų elgesio istorija yra panaši, nereiškia, jog asmenys turi tas pačias pasaulėžiūras.

Naudinga prisiminti N. Seaverio teoriją, kad „algoritmai yra kaip kultūra“, nes jų kūrimo procesas yra sudarytas iš daugybės kultūriniu arba techniniu motyvu paremtų pasirinkimų[83]. Atitinkamai, į aptartus pavyzdžius naudinga pažvelgti per algoritmų kūrimo praktikų konfliktavimo/dėrėjimo santykį.

Pastebėta, jog turiniu paremtų prognozių kūrimui reikia turėti daug duomenų apie pačius produktus[84]. Tad kai tokių duomenų trūksta, bendradarbiavimo prognozės gali būti kaip lengviau įgyvendinama alternatyva. Žvelgiant iš programavimo kaip kultūros perspektyvos, modelio metodas būtų renkamas dėl techninio įgyvendinamumo, o ne dėl socialinių motyvų. Kaip bebūtų, šis pasirinkimas vis tiek daro įtaką ne tik techniškumui, bet ir socialiniam poveikiui (net jei pasirinkimo procese to ir nebuvo siekta).

Kol kas aptarti ML iššūkių sprendimų tipai leidžia pastebėti, jog net ir kiekvienas techninis pasirinkimas daro socialinę įtaką ir jau vien dėl to yra verta tirti, kokiomis normomis remiasi programuotojai, darydami sprendimus. Kaip bebūtų, įdomu apžvelgti, ar programuotojai turi ir išankstinių socialinių nuostatų, darančių įtaką jų techniniams pasirinkimams.

Koreliacijos statusas. ML yra paremta tam tikra statistikos forma – duomenų kismo tarpusavio priklausymo tendencijų pastebėjimu. Tačiau, būtent dėl to, kad ML remiasi statistika, reikia prisiminti, jog paprasčiausias duomenų sutapimas dar nereiškia, kas šie duomenys daro įtaką vieni kitiems. Ši mintis, radusi savo pradžią Deivido Hume‘o „A Treatise of Human Nature“[85], vėliau tapo statistikos teiginiu, jog koreliacija nereiškia priežastingumo. Nors dabar šis teiginys tiesiogiai siejamas su statistika, naudinga suprasti, kad Hume‘as ją plėtojo kaip epistemologijos klausimą[86]. Taigi, tai toli gražu nėra tiesiog siaura inžinerinė praktika, bet greičiau mąstymo būdas, pasirinkimas matyti pasaulį koreliacijos ir priežastingumo nebūtino ryšio žvilgsniu. Tai reiškia, jog ši mąstymo samprata gali susidurti su nepritarimu, ypač jei pasikeistų aplinkybės.

Anot britų verslininko Kriso Andersono aplinkybės iš tiesų pasikeitė[87]. 2008 m. jis parašė straipsnį „The End of Theory: The Data Deluge Makes the Scientific Method Obsolete“. Tekstas sukėlė didelę

[83] Seaver, „Algorithms as Culture“, 9.
[84] Roy ir Dutta, „A Systematic Review and Research Perspective on Recommender Systems“, 3.
[85] John. P. Wright, *Hume’s „A Treatise of Human Nature“: An Introduction* (Cambridge: Cambridge University Press, 2009), 95–96.
[86] Wright, 79.
[87] Anderson Chris, „The End of Theory. The Data Deluge Makes the Scientific Method Obsolete“, 2008 m., 2.

diskusiją, nes jo autorius pasiūlė netikėtą idėją – mūsų technologijos dabar gali analizuoti tiek daug duomenų, kad užtenka atrasti koreliacijas ir tradicinis mokslinis metodas bus nebeaktualus. 2019 m. panašiai kalbėjo jau ir kompiuterių mokslininkas Rikas Sutonas, teigdamas, jog didžiausią įtaką padaro ne algoritmo matematinis profesionalumas, bet vidutiniškos kokybės algoritmai. Turėti ypač daug duomenų yra būtent tai, kas daro didžiausią įtaką ir kuria didžiausius rezultatus[88].

Šie pasisakymai kilo dėl to, nes pirmą kartą per žmonijos istoriją mokslininkai turi tokius didžiulius duomenų kiekius. Ir nors koreliacija tikrai nereiškia priežastingumo, turint tokį didelį duomenų kiekį, paprasčiausiai atkartojant koreliacijas, tam tikrus rezultatus gauti įmanoma. Ir šie straipsniai perša mintį, jog galbūt galime „peršokti“ priežastingumo atradimo dalį, jei rezultatas vis tiek pasiekiamas[89].

Tačiau, ar tai nesukels šalutinių poveikių apie kuriuos net nepagalvojome? Pavyzdžiui, jei ML atrastų koreliaciją tarp naudotojo gyvenamojo rajono, rasės bei nusikalstamos veiklos galimybės, ar tokiu atveju priežastingumas irgi tėra kliūtis? Tokios sistemos jau yra naudojamos skirtingų šalių policijos institucijų ir kaip tai kritikavo W. Chun, neįmanoma sužinoti, kiek žmonių buvo neteisingai apkaltinti ar suimti[90]. Taip yra dėl to, nes tokios programos neanalizuoja priežasčių, o tiesiog remiasi iš priežasčių kylančių rezultatų koreliacijomis[91].

Iki šiol aprašyti ML kūrimo žingsniai rodo, jog bene kiekvienas techninis sprendimas remiasi ne tik į matematiką, bet ir į socialinį pasaulį. Prognozės metodai siekia reprodukuoti socialinio rekomendavimo fenomeno socialinę sampratą. Pavyzdžiui, turiniu paremtos prognozės veikia pagal renkamų daiktų panašumą, o bendradarbiavimo prognozės - pagal naudotojų priklausymą atitinkamai socialinei grupei. Šie du rekomendavimo metodai yra tik keli iš šimtų pasirinkimų, bet jie iliustruoja, jog algoritminis mąstymas neretai remiasi socialinio pasaulio mąstymo logika.

[88] Rich Sutton, „The Bitter Lesson“, 2019 m., 1–3.
[89] Chris, „The End of Theory. The Data Deluge Makes the Scientific Method Obsolete“, 3.
[90] Wendy Hui Kyong, *Updating to Remain the Same: Habitual New Media*, 54.
[91] Wendy Hui Kyong, 54.

1.3. ML sprendimų vertybinių įtampų vertinimo sąvokos

Tiriant ML inžinerijos sprendimų priėmimo proceso įtampas, aktualu susipažinti ne tik su pačiais procesais, bet ir teoriniais žvilgsniais, tiriančiais algoritminį visuomeniškumą. Šiame skyriuje pristatoma teorija, nagrinėjanti visuomenės kodavimo duomenimis ir šiais duomenimis paremto įpročių organizavimo poveikį visuomenei bei asmeniui.

Algoritminių įpročių politiškumas. Wendy Chun rašo, jog formulė *įprotis* + *krizė* = *atnaujinimas* ("habit + crisis = update"), apibūdina naujųjų medijų elgesio principą, kuris tampa svarbiu neoliberalios visuomenės stabilumo išlaikymui[92]. Tokioje visuomenėje žmonės yra individai, vis mažiau turintys juos jungiančias bendruomenes, o neoliberalioje visuomenėje ekonominis mąstymas yra etikos ir demokratijos esmė[93]. Taigi, tokioje sistemoje ekonominis progresas yra būtinas. Tačiau norint, kad vyktų ekonominiai mainai, kažkokia *krizė* juos turi paskatinti imtis pirkimo (atnaujinimo) veiksmo, bet kiekviena krizė atneša suirutę. Tad egzistuoja įtampa, dėl pastovaus atnaujinimo poreikio, tačiau šis atnaujinimas visuomet yra griaunantis stabilumą. O visuomenės stabilumą anksčiau išlaikydavo bendruomenės, tačiau jos to padaryti nebegali, nes visuomenės darosi individualistinės. Kaip rašo Chun, šį stabilumą išsaugo būtent įpročiai. T.y. atsinaujinimas skatinamas per krizę, kuri yra pateikta egzistuojančių įpročių pagrindu. Krizės čia ne katastrofos, bet įpročių pokyčiai. Pokyčiai, pastoviai atnaujinantys esamus įpročius, o kartu ir kuriančios naujus įpročius[94].

ML taip pat yra viena iš naujųjų medijų, kuriai analizuoti tinka autorės teorija. Juk ML tikslas yra analizuojant naudotojo elgesį, nuspėti, koks galėtų būti kitas naudotojo poelgis. Kitaip tariant, remiantis duomenyse užkoduotais naudotojo įpročiais, ML pateikia ateities prognozes (pavyzdžiui pirkinio rekomendaciją), kuri naudotojui sukuria minimalią krizę. Šiuo atveju krizę, kad asmuo nori siūlomo produkto, tačiau jo neturi. Šiai krizei įveikti, naudotojui reikia paskatinti pirkimo įprotį ir taip įvyksta ekonominių mainų atsinaujinimas.

Taip pat Chun teorija atskleidžia ML dualumo įtampą. Autorė mini, jog naujųjų medijų naudotojai visuomet yra vienaskaitinėje daugiskaitoje („singular yet plural"), nes jie visi tarpusavyje yra susieti duomenų tinkle, bet kartu naujosios medijos siekia įgalinti individualius naudotojo įpročius[95]. Bet tai paskatina suprasti, jog ir prieš tai pristatyta naujųjų medijų formulė rodo, kad tokios technologijos yra dualistinės, nes vienu metu siekia išlaikyti senus naudotojų įpročius, kita vertus, senųjų įpročių pagrindu

[92] Wendy Hui Kyong, 12–14.
[93] Wendy Hui Kyong, 16.
[94] Wendy Hui Kyong, 11–13.
[95] Wendy Hui Kyong, 12.

kurti ir naujus[96]. Atitinkamai panašu, jog ir ML turi šių dualumų, kadangi yra kuriama remiantis tiek duomenimis, tiek statistine logika. Procesais, kurių vieno (duomenizavimo) šaltiniu yra naudotojų socialinė elgsena, o kito šaltiniu (modelio) yra matematika bei statistinė logika.

Duomenizavimo procesų sąvokos. Nors Chun daugiau dėmesio kreipia į pačią technologiją, kiti autoriai pastebi, jog jos aprašytas įpročių skaitmenizavimas kuria naują individualizacijos formą – „dividą“[97]. Ši sąvoka pristatyta Gilles'o Deleuze'o tekste „Postscript on the Societies of Control“. Deleuze'o tekstas yra teorinė darbo prieiga mąstyti apie duomenis ir asmenų reprezentavimą juose.

Ši teorija gali būti naudojama aptarti motyvus, kurie sukuria bei išlaiko tikėjimą, kad duomenyse slypi asmens reprezentacija. Šiuolaikinis asmuo yra užkoduojamas daugybėje skirtingų duomenų formatų pagal įvairius gyvenimo aspektus – asmuo kaip banko naudotojas, socialinės medijos naudotojas ar pan. Šiais skirtingais asmens gyvenimo rėžių kodavimais reprezentuojamus asmenis Deleuze'as vadina „dividais“[98].

Taip pat autorius analizuoja tokio asmenų reprezentavimo socialinį poveikį. Deleuze'as pristato kontrolės visuomenės sąvoką, kuria teigia, jog jei Foucault disciplinos visuomenėje individai būdavo kontroliuojami erdvėje per institucijas[99], kontrolės visuomenėje institucijų nebereikia, nes asmenys save seka ir kontroliuoja patys. Taip yra dėl to, nes pagal autorių, asmenų savistabą motyvuoja visuomenės noras konkuruoti[100]. Disciplinos visuomenėje kiekvienoje institucijoje asmuo pradėdavo „nuo nulio“ ir ilgainiui galėjo kilti galios laiptais, o kontrolės visuomenėje progresas niekada nesibaigia – asmenų sekimas ir „egzaminavimas“ pakeičiamas „tęstine kontrole“, paremta asmenų tarpusavio konkurencija[101]. Kas rodo, jog pats asmens reprezentavimo kūrimas yra konkurencijos, kaip vertybės keliama įtampa.

Čia galima įžvelgti paralelių su Chun teorija. Abu autoriai kalba apie nesibaigiantį progreso siekį – Chun kalba apie tai kaip pastovų įpročių atnaujinimą, Deleuze'as, kaip apie amžiną konkuravimą tarpusavyje. Taip pat, abiejose teorijose kalbama apie individus, kurie turi ryšį vieni su kitais ne dėl bendruomeniškumo, bet dėl to, kad jų duomenyse yra susieti tarpusavyje tinkle – ar tai būtų įpročių duomenys ar dividai.

Divido teorija taip pat leidžia diskutuoti apie tai, kurie asmenys daro didžiausią įtaką galutiniam ML rezultatui – programuotojai ar pats naudotojas. Pirma naudinga apsibrėžti individo ir divido santykį.

---

[96] Wendy Hui Kyong, 11–13.
[97] Lury ir Day, „Algorithmic Personalization as a Mode of Individuation“, 31.
[98] Deleuze, „Postscript on the Societies of Control“, 5.
[99] Michel Foucault, *Disciplinuoti ir bausti : kalėjimo gimimas* (Vilnius : Baltos lankos, 1998), 172.
[100] Deleuze, „Postscript on the Societies of Control“, 3–5.
[101] Deleuze, 5.

Nors dividas yra tarsi individo evoliucija, dividas nepanaikina individo egzistavimo. Taip teigia F. Bruno bei P. M. Rodriguez, sakydami, jog dividai yra dalis individų. Patys netapdami individais – jie tarsi „sujungia galimas trajektorijas“ tų individų įpročių. O duomenys tampa ta vieta, kur asmeniškumas ir individualumas nesusijungia[102].

Tai iliustruoja tą faktą, jog duomenims atsirasti reikalingas gyvo individo įsitraukimas, kadangi būtent individų įpročiais skaitmeninėje erdvėje yra fiksuojami ir verčiami duomenimis. Kaip rašo autoriai, kuriant dividą, individo subjektyvūs bruožai (polinkiai, jausmai, atmintis, t.t.) yra apibrėžiami agentiškumais (angl. „agencies“)[103], kaip kad pasirinkimai, įpročiai ar kiti elgesio fiksavimai. Tad jei divido (pavyzdžiui ML duomenų rinkinio) sukūrimas priklauso nuo individo veiksmų, atrodytų jog ML rezultatams įtakos tikrai turi pats naudotojas.

Kaip bebūtų, duomenys yra vieta, kur asmeniškumas ir individualumas nesusijungia[104]. Tai primena, jog nors duomenys ir kuriami remiantis naudotojo elgesiu, duomenų rezultatas smarkiai priklauso nuo to, kokius duomenis programuotojas nuspręs palikti bei kaip juos reprezentuoti. Taigi, nors dividas ir sukuriamas iš individo, šis veiksmas tėra vienkryptis ir pats asmuo negali daryti įtakos divido elgesiui. Tai logiška, kadangi divido sukūrimas priklauso nuo pačių programuotojų suvokimo, kaip ir kokie duomenys bus panaudoti. Tad rodos, jog duomenų kūrėjo priskyrimas taip pat yra įtampos vieta - jog ML rezultatas priklauso ir nuo naudotojų, kadangi jų skaitmeniniai veiksmai yra duomenų šaltinis, bet ir nuo programuotojų pasirinkimų, kokius duomenis išgauti bei kaip juos apdoroti.

[102] Fernanda Bruno ir Pablo Manolo Rodríguez, „The Dividual: Digital Practices and Biotechnologies“, *Theory, Culture & Society* 39, nr. 3 (2022 m. gegužės): 43, https://doi.org/10.1177/02632764211029356.

[103] Bruno ir Rodríguez, 36.

[104] Bruno ir Rodríguez, 43.

2. TYRIMO METODOLOGIJA

Tyrimo metodu pasirinkti kokybiniai pusiau struktūruoti giluminiai interviu su ML programuotojais, dirbančiais įvairiose kompanijose ir kuriančiais skirtingų paskirčių sprendimus. Remiantis N. Seaver teorija, tyrimo klausimai kurti siekiant atskleisti ML programų kūrimą „kaip kultūrą“ – t.y. pašnekovų klausti apie pagrindinius ML kūrimo procesus ir jų sprendimų priėmimo problematikas. Šitaip siekta atpažinti, kuriuose ML kūrimo žingsniuose tenka daryti pasirinkimus ir kompromisus (bei vardan ko), kurie inžinierių sprendimai prieštarauja pačių programuotojų tikslams ar prieš tai padarytiems pasirinkimams, su kokiais asmenimis taip pat tenka dirbti ar kieno dar vertybinius motyvus (ar lūkesčius, poreikius) atliepti.

Tyrimas atliktas lyginant pašnekovų pasisakymus vienus su kitais, ieškant bendrybių arba kaip tik skirčių. Tyrimo klausimai (Priedas Nr. 1. Interviu klausimų gairės) aptaria teorijoje apibrėžtas ML sprendimų grupes, kur labiausiai galimos įtampos – darbo su duomenimis bei modelio kūrimo ir vertinimo procesus. Taip pat programuotojų klausiama, kaip jie vertina ML kokybę, kurie programų kūrimo procesai svarbiausi, sudėtingiausi ir pan. Visų interviu metu siekiama pažinti, kas yra pagrindiniai vertybiniai žvilgsniai, kuriais orientuodamiesi programuotojai ir priima sprendimus. Tiek kuriant klausimyną, tiek interviu metu, klausimai formuluoti ML procesus analizuojant W. Chun algoritminių įpročių teorijos bei G. Deleuze‘o duomenizuotų asmenų sąvokų žvilgsniu. Galiausiai, rezultatai interpretuojami remiantis šiuo teoriniu įrankiu.

Taip pat svarbu pastebėti, jog nors tyrimas yra socialinių mokslų, jo autorius turi informatikos mokslų bakalaurą bei virš 2 m. patirties IT rinkoje kaip programuotojas. Tačiau darbo patirtis buvo ne ML kūrime. Tad ankstesnis išsilavinimas leido ML kūrimo procesus suvokti ne tik socialinių, bet ir informatikos mokslų prizmėje bei lengviau pastebėti socialiai konfrontuojančius inžinerinius sprendimus. Tačiau dėl to tyrimo objektyvumas nenukentėjo, kadangi pats tyrimo autorius niekada nėra dirbęs ML programuotoju ir būtent šios srities darbo procesai jam buvo nepažįstami taip pat, kaip būtų ir tik socialinių mokslų atstovui.

Tyrime dalyvavo 14 ML programuotojų. Interviu skaičius šio darbo tikslams yra pakankamas, kadangi jau apklausus pusę pašnekovų, pradėtas pastebėti atsakymų prisotinimas. Vidutinė interviu trukmė valanda. Kadangi rinkoje dauguma programuotojų yra vyrai (kai kuriais tyrimais moterų ML programuotojų yra apie 22%[105]), lygaus lyčių santykio sukurti nepavyko, tačiau 4 pašnekovės vis tiek

[105] Siddhi Pal, Ruggero Marino Lazzaroni, ir Paula Mendoza, „AI’s Missing Link: The Gender Gap in the Talent Pool“, Interface, žiūrėta 2024 m. gruodžio 9 d., https://www.stiftung-nv.de/publications/ai-gender-gap.

buvo moterys. Apklausiamieji buvo dirbantys skirtingose ML panaudojimo srityse: vaizdo atpažinimo programų, produktų pardavimo prognozių, rekomendacinių sistemų programų kūrime, taip pat naudotojų įpročių analizavime. Dauguma inžinierių dirbo versluose, tačiau vienas pašnekovas dirbo universitete, o dar vienas kūrė savo ML startuolį.

Pašnekovų užimamų pozicijų lygis varijavo nuo vidutinio lygio iki vyriausiųjų („staff“) programuotojų ar ML komandų vadovų. Amžius 25-35 m.[106] Dauguma pašnekovų buvo lietuviai, tačiau dalyvavo 2 užsieniečiai. Kaip bebūtų, tyrimui tai neturi svarios įtakos, kadangi dalis lietuvių programuotojų taip pat dirbo užsienio įmonėse, o ir tyrimo klausimynas buvo paruoštas taip, kad tirtų universalias ML darbo praktikas. T.y. klausimai paruošti aptarti universalius, skirtingų sričių bei įvairų ML patirties stažą apimančius klausimus.

[106] Ne visi pašnekovai norėjo atskleisti amžių, tad amžiaus rėžiai yra apytiksliai.

3. ML INŽINERIJOS VERTYBINIŲ ĮTAMPŲ TYRIMO REZULTATAI

3.1. Duomenizavimo iššūkiai

Ar rinkti kuo daugiau duomenų? Dauguma pašnekovų dalijosi troškimu turėti kuo daugiau duomenų apie naudotojus. „S*tengdavomės gauti duomenis iš įmanoma, kur tik tais įmanoma*“(I.3.). Dalis tai argumentavo tuo, jog duomenyse slypi naudotojų požymiai, kurių nesurinkus, gali būti ignoruojama tam tikra dalis asmenų. „J*eigu jų* (duomenų) *trūksta sistemingai, tai tu gali sukurti kažkokį sprendimą, <...> kuris visiškai ignoruoja kažkokią dalį ar ar klientų, ar kažko. <...> Tai kontroliuok, ką gali kontroliuoti, ko negali – tiesiog yra kaip yra*“ (I.1.). Tai leidžia įžvelgti manymą, kad duomenys yra ne kuriami, kaip sako duomenizacijos teorija[107], bet egzistuoja patys savaime.

Tačiau buvo ir priešingų požiūrių, kalbančių apie per gausaus duomenų rinkimo riziką tapti manipuliacijos įrankiu. „*Nemėgstu ir nenorėčiau dirbti kažkur, kur yra grynai. Toks per daug optimizavimas. <...> Pasigriebt viską. Sakykim, visą dėmesį žinai ar kažką tokio iš iš vartotojų*“ (I.4.). Įdomu, jog pripažįstama, kad kiti programuotojai gali turėti savų vertybinių argumentų gausiam duomenų rinkimui. „T*urbūt kas dirba toj srity, žinai irgi gal gali kaip tik atvirkščiai galvoti, kad jie nenori siūlyti produkto tiem kam nereikia. Tai dėl to yra labiau targetinimas tik tiems žmonėms, kurie potencialiai reikia arba norėtų tokio produkto. <...> Turbūt skirtingi žmonės skirtingai galvoja“* (I.4.).

Be to, pastebėta, jog skirtingos programuotojų vertybės gali daryti įtaką viena kitai. Čia pateikiamas pavyzdys, kai duomenų privatumo vertybė skatino programuotoją rinkti duomenis saikingiau: “*Data scientist‘as visada nori kuo daugiau duomenų <...> bet irgi realistiškai žiūrint, tiesiog neįmanoma. Visų duomenų visada visur apsaugot. <...> vis tiek reikia žinot kažkokias irgi sau ribas, kad ką reikia saugot, ko nereikia, ko reikia*” (I.4.). Tad nors duomenų gausos troškimas yra dažna vertybė, kur/ar nustatomos duomenų rinkimo ribos, yra pačių programuotojų vertybinis pasirinkimas.

Duomenų gausa nereiškia jų kokybės. Kita pašnekovų dalis, nors ir pripažino egzistuojantį norą turėti daug duomenų, tai vertino kritiškai. Pastebėta, jog svarbiau, kad duomenys būtų naudingi, negu gausūs. „*Dažniausiai žmonės ir galvoja, kad tokiose įmonėse kaip Facebook yra daug duomenų ir vien iš koreliacijos labai daug gali pamatyti. <...> Ta prasme duomenų yra daug, bet signalų juose nėra daug.”* (I.2.). „*Aktualūs duomenys tai problemai. Tai yra ne per daug duomenų. Nes visi sako ai, kuo daugiau duomenų, tuo geriau. Tikrai ne. Tai turi būt aktualūs duomenys*.” (I.6.). Tai iliustruoja, jog duomenys neegzistuoja patys savaime, bet yra sukuriami[108]. Norint gauti ML tinkamus duomenis,

[107] Mejias ir Couldry, „Datafication“, 2.
[108] Mejias ir Couldry, 2.

neužtenka „rasti“ tarsi jau egzistuojančius duomenis apie asmenis, bet jie turi būti užfiksuojami taip, kad atlieptų ML sprendžiamą tikslą.

Klausta, kaip parenkama, kuriuos naudotojų veiksmus sistemoje fiksuoti ir paversti duomenimis. Pašnekovai atsakė, jog tai matuojama verslo tikslų ir naudotojų įpročių įtaka vienas kitam. Iškeliamas tikslas, kokio poveikio verslui siekiama, tuomet apibrėžiami naudotojų įpročiai, kurie galėtų daryti įtaką šiam verslo tikslui ir kuriamas ML, skatinantis šį įprotį. „*Kokią problemą turi vartotojai? Kaip tai susiję su tavo verslo problema? <...> Iškeli hipotezę, kad jeigu mes padarysime chatbotą* (hipotetinis pavyzdys)*, tada išspręsime tokią vartotojo problemą ir tą vartotojo problemą tu matuosi per tam tikrą metriką – kaip transakcijų kiekis padidėja arba, kad galbūt user‘ių vartotojų pasitenkinimas platforma padidės <...>, pastatai tada tą sprendimą, kurį tu sugalvoji, paleidi ir žiūri, ar ta metrika pagerės, ar ne*.“ (I.2.).

Tai parodo, jog duomenų kūrimas yra paremtas verslo tikslų atliepimu. Tačiau, paprasčiausiai daryti prielaidas, jog tam tikrų naudotojų elgesių pokytis pagerins siekiamą verslo tikslą, neužtenka. Kaip ir W. Chun teorijoje[109], procesas yra paremtas pastoviu naudotojų elgesio pokyčių atnaujinimu ir rezultatų sekimu. Tad viena vertus ML duomenų kūrimas apriboja naudotojo pasirinkimų laisvę, kadangi duomenys kuriami atliepti iš anksto apibrėžtam tikslui. Kita vertus, pačių naudotojų reakcija į ML pasiūlymus, daro įtaką tam, kokie duomenys bus kuriami. Taip yra dėl to, nes programuotojai visuomet patiria tam tikrą įtampą, kad kuriant duomenis nebūtų per daug nenutolstama nuo naudotojų poreikių ar priimtinų ML pasiūlymų, kurie naudotoją paskatintų elgtis pagal siekiamą verslo tikslą.

Apibendrinimas. Pastebėta, jog duomenų gausa nutuokiama kaip tam tikra vertybė ar siekiamybė. Tačiau egzistuoja įtampa, kaip šis troškimas yra vertinamas – dalis to siekia be išimčių, kai kurie šį troškimą apriboja dėl kitų vertybių, o trečioji grupė, nors ir pripažino tokį troškimą, jį laikė apgaulingu ir tokiu, kurį reikia apriboti.

Pagrindinė skirtis, tarp priėmusiųjų duomenų gausos troškimą ir jį apribojančiųjų yra duomenų kūrimo strategija. Atrodo, jog duomenų gausos troškimas kyla iš manymo, kad duomenys egzistuoja patys ir juos reikia atrasti. Dėl to ir norisi jų turėti kuo daugiau. Tačiau kitos grupės pastebėta, jog neribotas duomenų rinkimas turi grėsmę manipuliuoti naudotojo pasirinkimus bei prarasti privačią informaciją.

Priešingai maniusieji propaguoja ne duomenų gausą, bet tikslingą duomenų atliepimą kuriamam ML tikslui. Tai atitinka duomenizacijos teoriją, jog duomenys iš tikrųjų neegzistuoja, bet visuomet yra kuriami. Šiuo atveju, duomenys kuriami taip, kad atlieptų ML sprendžiamus verslo tikslus. Tačiau, nors

[109] Wendy Hui Kyong*, Updating to Remain the Same: Habitual New Media,* 12–14.

čia duomenys kuriami pasiekti aiškiam tikslui, pats naudotojas čia nėra be galios. Šių programuotojų vertybė – kad fiksuojami įpročių duomenys turėtų ryšį su siekiamu verslo tikslu. Dėl šios vertybės programuotojai stengiasi visad atliepti nuolat besikeičiančius naudotojų troškimus.

3.2. ML rezultatų vertinimo dilemos

Kadangi ML tikslas yra pateikti spėjimų prognozes, šios prognozių kokybė turi būti įvertinta. Kaip įvertinti prognozės tikslumą, kuo remiantis ją matuoti, kiek procentų tikslumo yra pakankama? Norint tai suprasti, programuotojų buvo prašyta pasidalinti procesais, kaip jie įvertina savo sukurtas ML.

3.2.1. Laiko ir kokybės santykis

Pirmiausia pastebėta, jog egzistuoja vertinimo žvilgsnis, paremtas kokybės bei investuoto laiko santykiu.

Kada kokybė yra „good enough“. Aptariant, kada pasiektas tinkamas ML tikslumas, dažnai išgirsta frazė „good enough“. Ši frazė kitais žodžiais reiškia ribą (vadintą „treshold“) nuo kurios programuotojai jau teigia, jog programos rezultatai yra pakankamo procentinio tikslumo arba riba, kiek ML gali suklysti. ”D*ažniausiai tas 90% yra good enough <...> kad ten iki 93 pakelsi praleisdamas ten prie jo dar papildomai kokius šešis mėnesius, neduos papildomai tiek tos naudos <…> Pas mus tokie būna dauguma sprendimų good enough, o ne ne ten amazing ir panašiai*” (I.1.). „K*ažkokį stengiesi turėti iš anksto treshold'ą, <...> kas tau atrodo pakankamai gerai <...>, kad būtų čia naudingas šitas dalykas*“(I.4.). Tačiau toks vertinimas kelia įtampą: „*Kur dėsim tą treshold'ą, kuris nuspręs, kiek mes galim leisti modeliui dažnai suklysti?“*(I.3.).

Pavyzdžiui, duotas automatinio čekių skanavimo pavyzdys: „*20 žmonių, kurie approve'ina tuos čekius <...> Ir mes tarkim norim palikti tik 2 žmones, kurie approve'intų <...> Na, maždaug jo, čia jau dabar gerai turbūt, nes geriau negu buvo* ”(I.1.). Kitaip tariant, tikslaus apibrėžimo, koks procentas yra pakankamas pagerėjimas, nėra, bet tai nustatoma priklausomai nuo sprendžiamos problemos konteksto. Tikslas yra ne tam tikras pasiektas procentas, bet jog su ML būtų „geriau negu buvo“. Šios ribos nustatymas yra kitas vertybinių įtampų taškas, nes „geriau negu buvo“ yra reliatyvus vertinimas, skirtingai suprantamas įvairių asmenų. Taip pat vieta, kurioje gali atsirasti išimčių ar nuolaidų skirtingų atvejų vertinimui.

Laikas – nedirbti ilgiau nei būtina. Kokybės vertinimas ne pagal iš anksto nustatytus kriterijus, o paslankias vertinimo ribas yra susijęs su laiku. Klausiant, ar rezultatai jau pakankamai geri, kartu ir klausiama, kiek laiko užtruko to siekimas ir pokalbiai rodo, jog tai taip pat kinta įvairiose situacijose. *„Turi būti return of investment. Jei tu developinsi kažką savaitę ar pora dienų ir bus visai būsi patenkintas su mažu pagerėjimu. Bet jei developini kažką metus, tai tu turi turėti ir aukštesnį return of investment”*(I.2.). Taip pat pastebėta, jog programuotojai vertina mažiausiu įdirbiu pasiektus didžiausius įmanomus rezultatus: „*kažkokiu metu žinai, yra tas <...> diminishing returns <...> Tuos pirmus <...>*

low hanging fruits susirenki žinai, paskui tu vis dar gali ten kažką tobulinti, bet labai ilgai užtrunka“ (I.4.).

Vis dėlto, išskirtiniais atvejais yra poreikis sprendimus analizuoti ir ilgiau. Tačiau tuomet programuotojai jaučia įtampą, nes nėra užtikrinti, ar jų investuotas darbas parodys tiesioginius, verslo naudą teikiančius rezultatus: „*ilgai, kad dvejojau tikriausiai nebūtų, nes tiesiog nelabai, kaip čia finansiškai apsimoka ilgai apie tokius dalykus galvoti <...> per nagus gaučiau, jei visą dieną galvočiau apie tokį dalyką*“ (I.10.). Tai leidžia spėlioti ar lygiai taip pat mažai būtų mąstoma ir apie etiškai sudėtingą atvejį, o galbūt ir šis minėtasis iki galo neapmąstytas atvejis taip pat turėjo neapgalvotų pasekmių naudotojui? Atsakyti sunku, nes vertinimas yra abstraktus ar paremtas intuicija, kaip kad pasidalinta paklausus, ar darbe turi tikslių kokybės vertinimo apibrėžimų*„Nėra. Tai ateina man atrodo su patirtim.”(*I.2.).

Apibendrinimas. Seaver algoritmų kaip kultūros apibrėžime pastebėta, jog dalis algoritmų kūrimo praktikų dera, o dalis prieštarauja vienos kitom, nes siekia skirtingo galutinio tikslo[110]. ML kūrimui skirtas laikas bei rezultato tinkamumo riba yra toks vertybių konflikto taškas.

Pirmiausia, ML vertinama apibrėžiant prognozės patikslėjimo ribą, kada rezultatas pasidarė norimai geresnis, nei buvo prieš tai. Šis vertinimas paremtas ne konkrečiais kokybės apibrėžimais, bet kinta priklausomai nuo konteksto ir gali būti vertinamas profesine intuicija. Taip pat, intuityviai vertinama ir ar pagerėjimas buvo pasiektas nedirbant per ilgai.

Galiausiai, laiko ir kokybės siekiai susiduria konflikte. Viena vertus programuotojai siekia didinti ML rezultato kokybę. Kita vertus, jie siekia ir dirbti kuo greičiau, kas savaime mažina maksimalios pasiekiamos kokybės galimybę. Kam bus skiriama daugiau dėmesio - kokybės maksimizavimui ar darbo laiko minimizavimui - priklauso nuo sprendimą priimančio programuotojo ir jį supančio konteksto vertybių - ar tai būtų verslo greičio poreikis, programuotojo profesionalumo siekis.

3.2.2. Kada (ne)pakanka automatizuoto ML vertinimo

ML vertinimas apima tiek automatizuotą, matematinėmis formulėmis grįstą vertinimą, tiek pačių programuotojų vertinimo žvilgsnį, kuris, kaip rodo ankstesnė įžvalgos, neretai būna ir intuityvus. Gali kilti noras sakyti, jog automatizuotas ML vertinimas yra labiau paremtas faktais ar objektyvumu. Tuo tarpu programuotojų asmeninio žvilgsnio vertinimai yra labiau subjektyvūs. Tačiau tyrimas rodo, jog ML kūrime, riba tarp objektyvumo ir subjektyvumo bei patikimumo yra daug sunkiau apibrėžiama.

Matematinis objektyvumas ir žmogiškas subjektyvumas. Keli pašnekovai, kalbėdami apie ML vertinimą, patys (interviuotojui sąmoningai šių sąvokų nevartojant) paminėjo objektyvumo sąvoką,

[110] Seaver, „Algorithms as Culture“, 9.

priskirdami ją automatizuotam ir matematiniam ML vertinimui: „*Pirma, tiesiog vertiname <..> objektyvų modelio performansą. <...> Ar turime labiau kažkokias konkrečias vieno skaičiaus metrikas, kaip mean average“(*I.3.). „*Stengiamės visur viską daryti kuo labiau automatizuoti <...> kuo mažiau žmogaus įsikišimo. Jeigu sistema taip modelį vertina naudodama tokius pačius procesus, tai skaičiai nemeluoja*“ (I.12.). Atitinkamai, žmogaus vertinimas laikomas mažiau objektyviu: „*Po to eina <...> žmogiškesni, mažiau objektyvūs vertinimai <...> Kur dėsim tą treshold'ą, kuris nuspręs, kiek mes galim leisti modeliui dažnai suklysti*“(I.3.) Taigi atrodytų, jog programuotojas daro aiškią skirtį, kad tai, kas yra matematiška, t.y. nereikalauja žmogaus asmeninio vertinimo, yra objektyvu, o vertinimai, kuriuos programuotojas priima remiantis savo nuovoka, yra „mažiau objektyvūs“.

Dėl to įdomu, kad beveik iš karto pašnekovas tarsi pradeda prieštarauti sau, kalbėdamas apie tolimesnį ML testavimą. Kadangi šis pašnekovas kuria vaizdų atpažinimo ML, tokio tipo programos duomenims naudoja nuotraukas. Apie nuotraukų analizavimą jis tęsia: „*Žiūrim grynai akimis, grynai į konkrečias nuotraukas, ne į kreives, ir vertinam, kur modelis suklydo. Ar jis yra, kad klystų visą laiką ties kažkokiom konkrečiom nuotraukom <...> Pirmiausia toks labai objektyvus ir metrikom grįstas. Bet paskui <...> labiau toksai žmogiškais elementais“(*I.3.). Tad galiausiai atrodo, jog būtent žmogaus vertinimu pašnekovas pasitiki labiau. Ar tai reiškia, jog šis programuotojas nepasitiki objektyviomis metrikomis? O gal tai priklauso nuo konteksto, kad pirmuose testavimo žingsniuose jis labiau pasitiki matematika, bet kai ML yra paruošta ir testuojama realiam naudojimui praktikoje, čia nebepasitikima tik formulėmis, o kliaujamasi žmogišku žvilgsniu. Tai kelia klausimų, kas iš tikrųjų programuotojo nuomone yra objektyvumas.

Tolimesniame pasidalijime dar aiškiau atskleidžia žmogiško žvilgsnio vertingumas. Duodabt veido atpažinimo ML pavyzdį, minima, kad jei programos tikslumo vertinimas yra abejotinas, papildomai programos rezultatus patikrina ir žmogus – ar programa gerai atspėjo veidą, ar ne: „*Gali būti du setup'ai. Pirmas, kur modelis <...> autonomiškai veikia ir ką jis pasako, tas yra kaip objektyvi tiesa. Bet kartais <...> aš turiu Face Detection modelį <...> Bet jei jis neapsisprendęs yra kažkur per vidurį, mes galim tą tą viduriuką paimti ir suprasti, kad jo, jeigu tas confidence tarp 0,6 ir 0,4, duodam žmogui peržiūrėti ir nuspręsti ar čia tikrai?“(*I.3.).

Vis dėlto, iš pradžių programuotojas mini, kad taip pat galimas atvejis, kur ML rezultatai laikomi „kaip objektyvi tiesa“. Kaip nustatoma, kada užtenka pasikliauti vien automatizuotu „objektyviu“ vertinimu, o kada reikia ir žmogaus žvilgsnio? Šiame pasidalijime minima, jog žmogaus analizės reikia tada, kai ML tikslumas yra neužtikrintas. Tačiau tai kelia klausimų, kaip nustatoma riba, koks tikslumo

procentas yra pakankamas? Šiame pavyzdyje 60% tikslumas laikomas nepakankamu, bet ar 80% jau būtų pakankamai tikslu?

Kada neužtenka vien „objektyvios“ matematikos vertinimo. Pirma naudinga giliau ištirti, kodėl programuotojams ne visada pakanka automatizuoto ML vertinimo, kuris, kelių iš jų nuomone, yra objektyvus. Ar ši matematika tikrai yra objektyvi? Iš pirmo žvilgsnio atrodytų taip - juk matematinės formulės (čia vadinamos „metrikomis“) iš tiesų nėra sugalvotos paties programuotojo. Tačiau verta pastebėti, jog kokiomis formulėmis sistema bus vertinama – tai jau žmogaus pasirinkimas. Atitinkamai net ir šių formulių rezultatų interpretacija taip pat priklauso nuo žmogaus.

„Dažniausiai <...> reikėdavo vertinti labiau balanso kontekste. <...> Jeigu tu vertini metrikas <...> gal į jas smagu pažiūrėti ir smagu kartais nuspręsti, kad ok šitas vienas skaičius yra geriau už tą vieną skaičių, bet <...> ta metrika atrodo šiek tiek kitaip <...> Tam tikram kontekste gali būti gera, kitam kontekste gali būti bloga <...> Man atrodo labiau yra problema ne metrikos blogumas savaime, bet kaip tos metrikos vertinamos <...> subjektyviai <...> žmogaus. Jo, problema būna labiau interpretavime, o ne pačioj metrikoj.“ (I.3.)

Pašnekovai pripažino, jog „objektyvių“ matematinių formulių yra įvairių ir programuotojai bei verslo atstovai skirtingai nustato, kuriomis iš jų vertinti ML. *„Kažkiek gal produkto žmonės labiau linkę viską vertinti <...> kiek tas pinigų kainuos, kiek lėtai veiks, <...> Kiek ten userių bus nepatenkinti <...>. O mes labiau vertiname <...> tokiom inžinerinėm metrikom, kad kiek modelis ten dažnai klysta <...> precission recall curves ir pan. Ir dažniausiai bandydavome tai sujungti kažkiek. Jeigu modelis bus tiek netikslus, kiek userių bus nepatenkinti ir pan. <...> Kokį žmogišką tą poveikį turės. Ta tokia matematine metrika sakykim apjungti du konceptus.“(*I.3.). „*Kiekvienas priimamas sprendimas turi būti balansas tarp vienų arba kitų vartotojų grupių*” (I.12.)

Taigi, programuotojai pripažįsta, kad „objektyvios“ metrikos, kartais tik atrodo perteikiančios kokybės įvertinimą, tačiau pilnas ML kokybė vertinimas yra „balanso“ procesas. Balanso tarp verslo poreikių (pateikiamų „produkto žmogaus“) bei matematinių metrikų ar skirtingų naudotojų poreikių. Kalbant pašnekovo žodynu, tarsi balanso tarp subjektyvaus ir objektyvaus. Ir matematinėje metrikoje siekiama „apjungti du konceptus“ – pašnekovų suvokimu subjektyvų ir objektyvų.

Be to, vėliau ir pašnekovas pastebi, kad net ir neva užtikrintus rezultatus rodantis ML gali būti šališkas („biased“). Taip gali nutikti, jei ta nedidelė paklaidos dalis aprėps būtent tuos etiškai jautrius atvejus, kaip ML neatitikimą tautinėms mažumoms ar pan. „*Tarkim, tavo metrikos rodo, kad tau modelis 90 % tikslus. Bet <...>, jei tavo modelis yra face recognition, tai tu matai, kad iš tų 10%, kur modelis klysta, 5% yra <...> kažkokios lyties ar ar rasės žmonės <...>. Tai toks gal labai paviršutiniškas*

*testavimas. Gali, gali nepagauti tokių problemų kaip modelio bias'as <...>. Dėl to svarbu yra stebėt grynai konkrečiai pačiam akimis, kur klaidos vyksta"(*I.3.). Galiausiai pastebima, jog ne automatizuota, o žmogaus patikra gali pastebėti tokį šališkumą.

Šiame kontekste programuotojas šališkumu laiko nevienodą skirtingų situacijų padengimą duomenimis. Tačiau verta pagalvoti, kad manymas, jog įvairias situacijas reikia reprezentuoti vienodu kiekiu duomenų, taip pat yra kita šališkumo forma. Juk pateiktame pavyzdyje socialinės mažumos, išimtiniai elgesio atvejai ar panašūs „biased" atvejai iš tikrųjų yra rečiau pasitaikantys atvejai. Ir duomenyse išlyginus šį netolydų pasiskirstymą, duomenys savo procentiniu skirtingų atvejų padengimo pasiskirstymu nutolsta nuo realybės, kurią siekiama reprezentuoti.

Jei ML tikslas yra siūlyti prognozes remiantis duomenimis, šios programos objektyvumas priklauso nuo to, kaip ML kūrėjai suvokia, kas yra objektyvūs duomenys. Žinant, jog duomenys neegzistuoja patys savaime, bet visada yra kuriami[111], galima teigti, jog duomenys, kaip objektyvumo faktorius, visada savyje turės tam tikrą šališkumą. Tad dabar dar rimčiau kyla klausimas, ką pašnekovas turėjo omenyje sakydamas, jog „*Gali būti du setup'ai. Pirmas, kur modelis <...> autonomiškai veikia ir ką jis pasako, tas yra kaip objektyvi tiesa".* Juk žvelgiant į pastarąjį pasidalinimą sunku įsivaizduoti atvejį, kada modelis iš tikro galėtų būti objektyvus ir atrodo, jog programuotojo atžvilgiu, žmogiškas žvilgsnis visada praverstų.

Žmogaus testavimas – prabanga. Vienas iš atsakymų vėl būtų laiko sąnaudos. Atviraujama, kad žmogaus testavimui ne visada užtenka laiko resursų ir tai atliekama tik kritiniais atvejais: *„ne visąlaik yra tam prabanga, nes kartais produktas turi būti greitas ir pigus, bet kartais, jeigu tai yra labiau toks jautresnis use case'as, galim leisti sau tai padaryti"(*I.3.).

Tuomet kurie atvejai yra pakankamai jautrūs? Dalijamasi, jog jautrumas matuojamas pagal tai, „*Kiek bus klaidų padaryta ir kiek brangiai tos klaidos kainuos?"(*I.3.) Pasitikslinus, kas sveria, kiek kainuos ML klaidos, atsakyta, jog tai apsprendžiama remiantis verslo rekomendacijomis – kurios ML programų klaidos sukeltų didžiausias problemas verslui: *„Šnekėtume su produkto <...> žmogumi, nes jis žino, kiek tas kainuos, kokios problemos bus ir panašiai. <...>. Ir jis padeda mums, kaip inžinieriams, suprasti kiek įmanoma labiau, kokios yra rizikos, kokios yra kainos, kokias metrikas mes turime pasiekti ir apskritai kokios metrikos yra svarbios ir nesvarbios"(*I.3.)

Kitaip tariant, mąstant, kurie atvejai yra verti nuodugnaus žmonių testavimo, tai nusprendžiama ne remiantis kokybės didinimo vertybe, bet klaidų minimizavimu. Klaidos čia apibrėžiamos verslo, o ne naudotojų poreikiais. Tai leidžia manyti, jog pačių naudotojų interesai (kaip minėtasis duomenų

[111] Mejias ir Couldry, „Datafication", 2.

šališkumas mažumų atžvilgiu) testavime bus apsaugoti tik tada, jei jų nepasaugojimas turės tiesioginį ir grėsmę keliantį ryšį verslo siekiams.

Panašiai apie testavimą kalba ir kiti, jog testuojant būna neužtikrinti, koks ML rezultatas turėtų būti teisingas: „A*š būna skaitau ir man atrodo. Nu nežinau, čia tikrai taip ar ne? Tu tiesiog nežinai. Tai, nes tu nesi domain tas expert."* (I.14.) Čia minimas atvejis, kai programuotojai neužtenka jos žinių apie sritį, kurioje naudojamas ML ir klausiama „*domain experts. Jie gali iškart pasakyti*" (I.14.). Trumpai tariant, klausiama tų žmonių, kurie yra tos srities, kurioje naudojamas ML, žinovai.

Taip pat minima, jog net ne visi klausimų keliantys atvejai yra tikrinami pačių programuotojų, kadangi tokių atvejų kartais būna per daug: „*Netikrini kiekvieno <...>, nes ten tūkstančiai tų. Patikrini tiktai 100 kokį ir daugmaž žinai, kad neblogai veikia ir viskas <...> I.: 100 iš kiek maždaug? P.: Oi. Tūkstančių*" (I.14.). Dėl to net ir tada, kai testuojama ne tik automatizuotai, bet ir žmonių patikra, klaidų galimybė išlieka visada – ar dėl srities nesuvokimo, ar dėl visų atvejų nepadengimo. Tai dar kartą atskleidžia, jog šiame ML testavimo žingsnyje siekiama ne didinti programos kokybę, bet mažinti rimtų klaidų riziką.

Apibendrinimas. Kitas vertybinių įtampų taškas, kylantis vertinant ML, yra nuspręsti, kaip programos prognozių rezultatai bus testuojami – tik automatizuotai ar ir papildomai vertinant žmogui.

Automatizuota, matematinėmis formulėmis paremta ML patikra atliekama visais atvejais. Iš pirmo žvilgsnio programuotojams ji atrodo „objektyvi", kadangi yra apibūdinama faktiškai. Tačiau pastebėta, jog net automatinei patikrai rodant, kad ML yra tikslus, kritiniai atvejai gali būti itin netikslūs, pavyzdžiui neatliepti mažumų poreikių. Tokie atvejai yra pastebimi tik žmogaus žvilgsniu. Dėl to atrodo, jog žmogiška patikra būtų naudinga praktiškai visoms ML programom. Tačiau žmonių darbo laikas kainuoja ir tai sukelia įtampą – kurie ML yra verti tik automatizuotos, o kurie ir papildomos žmogaus patikros.

Šiuos sprendimus lemianti vertybė yra ne ML kokybės maksimizavimas bet rimtas rizikas keliančių klaidų minimizavimas. Tai turi kelias pasekmes. Pirmiausia, ML rezultatai niekada nėra apsaugoti nuo klaidų – ar dėl to, kad žmogiškoji patikra dėl normatyvaus vertinimo gali nepastebėti tam tikrų ML klaidų, ar dėl to, kad ML apima daugybes duomenų ir žmogiškų resursų visiems atvejams ištestuoti nepakanka.

Antroji pasekmė – kadangi ML yra kuriami verslo tikslams, rimčiausios rizikos taip pat yra susijusios su verslo grėsmėmis. Dėl to, tais atvejais, kai ML gali kelti riziką tam tikrai naudotojų grupei, bet ši rizika nebūtinai kels tiesioginį piniginį nuostolį kompanijai, naudotojų poreikių neatliepimas gali būti nepastebėtas arba nelaikomas pakankama rizika, verta papildomo testavimo.

3.3. Programuotojų pastebimos ML etinės grėsmės

3.3.1. Ne visa, kas leistina, netrikdo sąžinės

Kita pašnekovė pastebėjo, jog prieš maždaug 10 m., jos darbo patirtyje buvo gana dažna personalizuoti naudotojų pasirinkimų galimybes pagal jų asmeninę informaciją. Pavyzdžiui dirbant draudimo įmonėje turėta papildomų asmens duomenų (kaip amžius, lytis), pagal kuriuos ML buvo sukurtas daryti spėjimus, jog tam tikras amžius ar lytis yra rizikingesni. "*Draudimo įmonės iš esmės gali gauti daugiau duomenų <...> Gali net tai panaudoti automobilio kainodaros tarifuose įvertinti, kokia tikimybė, kad tu padarysi didesnę žalą arba, kad tavo yra rizikingas elgesys. <...> mes padarome ar ne tas išvadas iš duomenų, kad ten Porsche vairuojanti moteris 35 metų, tikėtina, bus, sukels mums žalą, kurią neapsimokės už mažesnę kainą laiduoti."* (I.5.). Svarbu pastebėti, kad ji tam nepritarė, nes sakė: „*aš pati suprasdavau, kaip <...> aš save pasmerkiu didelei draudimo kainai*" (I.5.). Vis dėlto, produktą toliau kūrė.

Pašnekovė vėliau pripažino, kad dabar tokie duomenys "*jau yra uždraudžiami, ypatingai lyties diskriminaciniai, kad negalima jų naudoti*" (I.5.). Vis dėlto, ji ir kiti pašnekovai pripažįsta, jog ML kūrime galima panaudoti tarpinius (tarsi legalius) duomenis, kurie atitiktų draudžiamus naudoti duomenis, kaip lytis. "A*š pati žinau, kad tu gali netiesiogiai <...> iš kitų kintamųjų išsivesti, kad čia yra, tarkim, moteris arba vyras*." (I.5.). „*If you want to use any sort of demographics, probably we could do it in a way*."[112] (I.11.). Nors pašnekovės renkasi to nedaryti, pripažįsta, kad tai panaudoti įmanoma „*I wouldn't do that. But I know that it's not kind of out of reach*"[113] (I.11.).

Apibendrinimas. Aptarti atvejai, kai technologiniai sprendimai programuotojų žvilgsniu atrodė neetiški. Vienu atveju, nepaisant nepritarimo, programuotoja dirbo toliau, kadangi jos žvilgsniu neetiška inžinerija tuo metu buvo legali ir verslo siekiama. Kitu atveju, pastebėjusios, jog inžineriniu būdu gali apeiti teisinius apribojimus, programuotojos rinkosi šių išimčių nedaryti.

Šiuo atveju sprendimų įtampa remiasi į programuotojų profesinę etiką. Tai parodo, jog ML inžinerija kaip kultūra susiduria ne vien su technologinių vertybių konfliktais, bet kartais ir konfliktu tarp technologinių siekių bei asmeninių etinių įsitikinimų. Šiuose pavyzdžiuose programuotojų pasirinkimams darė įtaką verslo siekiai, teisinės aplinkybės, asmeniniai įsitikinimai.

[112] Vertimas: „Jei norėtume naudoti demografinius duomenis, tikriausiai galėtume tai padaryti"
[113] Vertimas: „Aš to nedaryčiau. Bet žinau, kad tai nėra nepasiekiama"

3.3.2. Nelauktas poveikis naudotojų įpročiams

Kalbant apie ML procesus, kuriuose gali atsirasti etinių grėsmių, paminėtas ML poveikio naudotojams vertinimas. Pašnekovas čia mini metrikas, kaip įrankį sekti ML rezultatams – ar ML padidėjo naudotojų praleidžiamas laikas programoje, pirkimų skaičius ar panašiai. Pripažįstama, jog jei metrikos bus nukreiptos tam tikro žmonių elgesio skatinimui, tai naudotojams gali sukurti naujus įpročius, kaip kad priklausomybę.

„*Jeigu metrikos pasirenkamos, kurios turi long term kažkokias tai pasekmes, kurios nėra apgalvotos. Tas gali ir sukurti kažkokių pasekmių, kurių mes nesitikėjome. <...> Mes turime pavyzdį feisbukas arba instagramas, kur lyg ir galvojo daug apie metrikas. Bet jeigu šiek tiek pasveria labiau į engagement'o optimizavimą, tai gauna labai addictive appsus*“(I.2.).

Kad apsisaugotų nuo neprognozuotų naudotojų elgesio pokyčių, kompanijos atlieka tyrimus su naudotojais: „*Useriams tu skambini, darai interviu ir taip toliau. Tai tas irgi formuoja metrikas. Ir į tą investuojama žiauriai daug, žiauriai daug*” (I.2.) „*daro šitas apklausas vartotojų, ta prasme, tiesiog vat ant gyvų žmonių testavimą*“ (I.9.)

Taigi metrikos – matematinės formulės skirtos specifinių naudotojų įpročių pokyčio vertinimui – yra pagrindinis ML kūrimo įrankis, bet kartu ir potenciali grėsmė netikėtam naudotojų elgesio pasikeitimui. Iš to kyla įtampa nepasitikėti savo paties kuriamu produktu ir nuolat jį tikrinti. Kitaip tariant šiuo atžvilgiu programuotojas etišką darbą suprastų kaip tokį, kuris yra paremtas pastoviu nepasitikėjimu savo produktu.

W. Chun teorija leidžia ML veikimą interpretuoti kaip „įprotis + krizė = atnaujinimas”. Chun kalba, jog naujosios medijos, švelniai skatindamos tam tikrus naudotojų įpročius, po truputį kuria pokytį (krizę) naudotojų aplinkoje – t.y. stipresnį poreikį šiam įpročiui. Toks nedidelis pasikeitimas, skatina naudotojus prisitaikyti prie šio pokyčio ir įgyti naujus įpročius[114]. Pavyzdžiui ML kuriamas optimizuoti programėlėje peržiūrėtų vaizdo įrašų laiką (šiuo atveju metrika būtų vaizdo įrašų peržiūros trukmė). Ilgainiui tai sukuria krizę – naudotoja nori daugiau laiko praleisti programėlėje, nei tai darė anksčiau. To rezultatas yra naudotojos įpročių atnaujinimas – dabar jos bazinis poreikis praleisti tam tikrą laiką programėlėje padidėja.

Tai atitinka interviu įžvalgas, kuriose pabrėžiama, jog per didelis metrikos optimizavimas gali sukelti priklausomybę. Tačiau pateikiama ir nauja įžvalga – kompanijos atlieka gausius tyrimus su savo naudotojais tam, kad atpažintų jų elgesio pokyčius ir ar juose nėra nepageidautinų rezultatų. Tai reiškia,

[114] Wendy Hui Kyong, *Updating to Remain the Same: Habitual New Media*, 12–14.

jog formulėje „įprotis + krizė = atnaujinimas” santykis tarp įpročio ir jo sukurto atnaujinimo nėra toks akivaizdus, kadangi skatindami tam tikrą įprotį, programuotojai patys nėra tikri, kokį rezultatą tai sukurs.

Apibendrinimas. Elgesio rekomendacijas siūlančių ML veikimas remiasi tam tikrų naudotojų įpročių skatinimu, taip siekiant paskatinti naudotojus elgtis pagal verslo poreikius. Tačiau pastebėta, jog kartais dėl ML naudotojai įgyja naujų įpročių, kurių programos kūrėjai nesitikėjo. Tai vertybinės įtampos taškas, kur ML kūrėjai turi nuspręsti, kurie naudotojų elgesio pokyčiai yra priimtini, kurie ne. Priimtinais laikomi tie pokyčiai, kurie atitinka pačių programuotojų spėjimą apie tai, kokią įtaką naudotojų elgesio kaitai turės tam tikro įpročio skatinimas.

3.4. ML technologijos sampratų iššūkiai

3.4.1. Trukdžiai dėl verslo užsakovų nenuovokos apie ML technologiją

ML komandų vadovė, turėjusi daug ML projektų vadovavimo patirties, išsiplėtė, jog ML yra ne tik inžinerija, bet ir skirtingų asmenų interesų derinimo procesas. „*Tai yra truputį plačiau negu tik kodas <...> Iš finansų ateina, sako mes norim to, ten iš marketingo ateina, sako mes norim to <...> O tarkim yra 5 žmonės, o reikia 10 žmonių. Ir tada reikia sudėlioti ir nuspręsti kas gausis. Kažkas kažką pirmą, kažką antra <...>, o kažkas išvis nieko negaus.*" (I.6.). Tai rodo, kad vienas iš darbų yra išskirti prioritetus, kieno siekiai bus įgyvendinti bei iki kokio lygio, o galbūt kurių interesų grupių siekių ML kūrimui iš vis turi būti atsisakyta.

ML kūrime tai pat tenka susidurti su nerealistiškais lūkesčiais ar inžinieriams nelogiškais verslo prašymais. „*Kartais ir kažkokių ezoterinių, gal kartais reikalavimų ateina, kur nesupranti, net kodėl reik daryt*" (I.10.) „*Tikriausiai, kad dar žmonės nelabai supranta, kas yra tas dirbtinis intelektas <...> Kad kažkas, prisėdęs prie magiško rutulio, jiems išburs kažką ir tada jiem išvis nieko nebereikės daryti ir viskas savaime susitvarkys*" (I.6.). Tai rodo, jog ML užsakovai gali manyti, kad ML technologija yra daugiau nei inžinerija, bet kažkas mistiško. Panašu, jog ne verslo sferoje egzistuoja tam tikras ML, kaip visą išsprendžiančio atsakymo, mitologizavimas. Žinoma, tai nėra tiesa: „*Tai nėra panacėja. Tai nėra taip, kad iš karto sukuri kažką, kas sprendžia visas tavo gyvenimo problemas*" (I.6.).

Tad panašu, jog ML kūrime kartais svarbiau už technologinius sprendimus yra vadybinių klausimų, tikslų ir lūkesčių derinimas. "*Kad projektas būtų neįgyvendintas dėl <...> kažkokių ten techninių ypač dalykų, nu niekada nesu girdėjusi dėl techninių dalykų. Visą laiką būna <...> Kažkas susipyko, kažkas nepasidalino*" (I.6.).

Apibendrinimas. Pastebėjimai atliepia N. Seaver teoriją, jog ML tikrai yra kultūrinis procesas, apimantis įvairių vertybių[115], o šiuo atveju ir lūkesčių bei prioritetų derinimą ar atmetimą. Taip pat, kitoje literatūroje tas pats autorius analizavo įvairius mitus, kuriais remdamiesi tam tikrą ML sritį suprato programuotojai[116]. Šio darbo pasidalinimai rodo, jog ne tik programuotojai, bet ypač verslo užsakovai mitologizuoja ML kaip visas problemas išsprendžiantį atsakymą ir tarsi užsimiršta, kad tai yra inžinerinis procesas.

Dėl šių priežasčių programuotojai atsiduria prioritetų ir tikslų nustatymų įtampoje. Šios įtampos pagrindinė priežastis – ne programuotojų, bet verslo užsakovų žinių trūkumas apie ML inžineriją ir nepagrįsti lūkesčiai. Tai svarbu, nes bendros pasaulėžiūros tarp inžinierių ir užsakovų stygius ne tik

[115] Seaver, „Algorithms as Culture", 9.
[116] Seaver, *Computing Taste*, 19.

trikdo darbą, bet ir gali apsunkinti diskusijas iki šiol minėtais etiškais atvejais, liečiančiais poveikį naudotojui.

3.4.2. ML kaip neutralus ir kartu šališkas problemai

ML nešališkumas problemai. Daug pašnekovų paminėjo, jog pati ML technologija yra neutrali sprendžiamoms problemoms: „*ML nėra nei geras, nei blogas*“ (I.2.), „*Pats algoritmas nėra nei geras, nei blogas arba tinka tai problemai spręsti, arba netinka tai problemai spręsti*” (I.6.), „*The model is agnostic to the problem*[117]” (I.7.).

Pavadinęs ML agnostišku (t.y. nepažįstančiu problemos), pašnekovas tęsė, jog svarbiau dėmesį skirti tikslui, kuriam pasiekti ML yra naudojamas. Pašnekovas sulygino verslo ir nevyriausybines organizacijas sakydamas, kad abiejų žinomumui padidinti ML veiktų panašiai, tačiau tikslas skirtųsi. „*It's about the purpose. <…> Companies want to understand where to put their next euro. Meaning should I put like a Facebook ad? Should I put like a TV ad <...> But NGOs were faced with a similar problem where they have limited budget and they have to organize events to engage with people <…> Measuring these you can use pretty much the same techniques*[118]” (I.7.).

Kaip bebūtų, svarbu suprasti, jog agnostiškumas nereiškia modelio neutralumo. Kitoje vietoje tas pats pašnekovas pripažįsta, jog sąmoningai parenka, kad ML teiktų prioritetą vieniems sprendimams prieš kitus. Pavyzdžiui: “*I would argue that in this case it's better to over forecast because under forecasting it's really bad for the user.*”[119] (I.7.). Tai pašnekovas pasakė duodamas pavyzdį, jog kurdamas inventorizacijos ML, jis geriau jį kurtų taip, kad produktų liktų daugiau nei gali būti paklausos, nes nenupirktų produktų sugedimas yra mažesnė grėsmė nei pasiūlyti pirkėjui produktą, kurio sandėlyje nebėra. Tad nors skirtingiems tikslams kuriami ML „nepažįsta“ problemos ir gali naudoti bene tą pačią technologiją, programuotojas sprendžiamą problemą žino. Kas rodo, jog manymas, kad ML yra agnostiškas, nei geras ar blogas, tėra dalinė tiesa – pats ML, kaip technologija, toks ir yra, tačiau ML kūrimo procesas ir konkretus ML sprendimas nėra neutralus problemai, bet kreipiantis į tam tikrą tikslą.

ML gali iškreipti duomenis. Kita programuotoja išplečia šią temą. Ji iki galo nepasitiki jokių ML rezultatais, nes žino, kad programą kūręs programuotojas galėjo pakreipti ML taip, kad rezultatai būtų panašesni į jo siekiamą rezultatą. „*That is a stupid machine, basically and very probabilistic and might*

[117] Vertimas: „Modelis agnostiškas problemai“

[118] Vertimas: „Tai susiję su tikslu. <...> Įmonės nori suprasti, kur naudoti kitą eurą. Ar turėčiau įdėti „Facebook“ reklamą? Ar turėčiau įdėti televizijos reklamą <...> Tačiau NVO susiduria su panašiomis problemomis - jos turi ribotą biudžetą ir turi organizuoti renginius, kad užmegztų ryšius su žmonėmis <...> Čia galima naudoti beveik tuos pačius metodus.“

[119] Vertimas: „Sakyčiau, kad šiuo atveju geriau prognozuoti daugiau nei reikia, nes nepakankamai prognozuojant tikrai blogai vartotojui.“

make a lot of mistakes". "I*f you see any statistics <...> model achieved like 60% <...> I am sometimes reading it like someone believed and wanted that model to achieve like 60% <...> In the data you can bend a lot of things*"[120] (I.11.). Tam antrina ir kiti: „S*tatistika yra didžiausias melas. Tikrai labai dažnai pasiima ir iš vienos pusės ir galima sumanipuliuoti*" (I.6.).

Prisitaikymas prie naudotojų unikalumo. Iki dabar minėtieji technologijų analizės žvilgsniai kalbėjo apie duomenų mokslą ar ML prognozes. Kalbant su kuriančiais didžiuosius kalbos modelius (LLM – ML programų rūšis, dirbanti su kalbos atpažinimu ir reprodukavimu[121], kurios pavyzdys būtų „ChatGPT"), išgirstas kitoks žvilgsnis: „(DI) *suteikia pilnai suasmenintą, personalizuotą patirtį, nes informacija, kuri yra pasiekiama internete <...> yra bendrinė, o kiekvienas žmogus yra unikalus <...> todėl DI padeda įsigilinti į žmogaus situaciją ir pritaikyti patarimus, klausimus jam*" (I.13). Atrodo, jog programos pritaikymas subjektyviems žmonių poreikiams yra laikomas šios technologijos pranašumu. Kas būtų priešinga tam, ką kalbėjo gryno ML programuotojai, apie šios technologijos agnostiškumą bei potencialią statistikos iškreipimo galimybę.

Tačiau gilinantis į tai, kaip kuriami šie „lankstūs" LLM modeliai, buvo išgirsta: „*Pateikiame pavyzdžių, kaip jie turėtų bendrauti tame pokalbyje, kokius atsakymus turėtų suteikti maždaug tam tikrose situacijose*" (I.13). Paradoksalu, kad LLM yra giriamas dėl to, kad prisitaiko prie naudotojų poreikių, tačiau tai yra kuriama programuotojams patiems apibrėžiant šių naudotojų elgesio situacijas. Tai daroma remiantis savo supratimu apie tai, ko turėtų norėti naudotojas.

Apibendrinimas. Programuotojų suvokimas apie tai, kas yra ML, yra dvilypis. Viena vertus didžioji dauguma sako, jog pati ML technologija yra neutrali problemoms ir negali būti laikoma nei gera, nei bloga. Kita vertus, kalbant apie ML kūrimo procesus, programuotojai pripažįsta, jog tam tikrais atvejais ML pakreipiamas laikytis tam tikrų preferencijų, ar net gali būti sąmoningai pakreipiamas sukurti kūrėjų motyvams palankų ML rezultatą. Tad nors pati ML technologija kaip inžinerija ir yra neutrali, jos kūrimo procesai ir konkretūs rezultatai tokie nėra.

Tai dar stipriau parodo, jog ML kūrimas yra ne vien inžinerinis veiksmas, bet ir procesas atliepiantis kiekvieno prie ML prisiliečiančio kūrėjo tikslus. Šiuo atveju įdomu pastebėti pačių programuotojų nuomonių nepastovumą – vienu metu ML laikant neutraliu, bet po kelių sakinių kalbant apie aiškius ML pritaikymo problemai veiksmus.

[120] Vertimas: „Iš esmės tai yra kvaila mašina, smarkiai remiasi tikimybės ir galinti padaryti daug klaidų". „Jei matote kokią statististiką <...> modelis pasiekė pavyzdžiui, 60 % <...> Aš kartais tai skaitau taip, lyg kažkas tikėjo ir norėjo, kad tas modelis pasiektų pavyzdžiui, 60 % <...> Duomenyse galima daug ką iškreipti"

[121] Mohaimenul Azam Khan Raiaan ir kt., „A REVIEW ON LARGE LANGUAGE MODELS: ARCHITECTURES, APPLICATIONS, TAXONOMIES, OPEN ISSUES AND CHALLENGES", s.a., 1–2.

Šis sau prieštaraujantis dualumas gali būti dėl to, kad ML inžinerija nėra tiesiog faktiškai apibrėžiama matematika, bet yra inžinerijos praktika, kuri remiasi į duomenis. Kaip rodo Duomenizavimo , pats duomenų gavimo procesas yra tik siekiantis būti realistiškas naudotojų duomenų atžvilgiu - juk programuotojai įdeda daug pastangų užtikrinti, kad duomenys būtų gaunami jų suvokimu „gerai“. Bet kartu žinoma, kad duomenys patys neegzistuoja[122] ir kaip pastebėta, jų kūrimas siekia atliepti galutinį ML kūrimo tikslą.

Šį dualumą padeda analizuoti ir W. Chun teorija. Ji kalbėjo apie medijų kaip ML pastovų atsinaujinimą, kai vienu metu siekiama išlaikyti senus naudotojų įpročius, o kartu senųjų įpročių pagrindu kurti ir naujus[123]. Kitaip tariant, ML vienu metu turi išlaikyti „agnostiškumą problemai“, bet kartu ir atliepti ML kūrimo tikslą. ML tikslo (įprastai verslo siekiai kaip, pavyzdžiui, naudotojų pirkimo skatinimas) siekimas jau nebėra neutralus, bet pritaiko duomenyse rastas tendencijas ML tikslų skatinimui. Tai įneša tam tikrą suirutę į naudotojo įpročius, tačiau pakankamai švelniai, kad naudotojas prie jų prisitaikytų ir pradėtų elgtis pagal ML kūrėjų kreipiamą elgesio pokytį.

Taigi, žvelgiant platesniu kampu, pats ML kūrimas yra pastovios įtampos taškas tarp atliepimo naudotojų įpročiams ir poreikiams bei ML kūrimo tikslų. Šią įtampą galima apibūdinti W. Chun mintimi, jog naujosios medijos siekia pastoviai atsinaujinti tam, kad išliktų tokios pat („updating to remain the same“)[124].

[122] Mejias ir Couldry, „Datafication“, 2.
[123] Wendy Hui Kyong, *Updating to Remain the Same: Habitual New Media*, 11–13.
[124] Wendy Hui Kyong, 11–13.

3.5. Požiūris į ML naudotoją

3.5.1. Programuotojų suvokimas apie naudotoją – tarp asmenybės ir įrankio

Klausiant, kada programuotojai mąsto apie galutinį naudotoją, dauguma įvairiais būdais atsakė, jog bene visada. „V*ienas iš pirmų dalykų, kuriuos pagalvoti turi, nes iš esmės turi arba pats kažkaip įsijausti į tą vaidmenį*“ (I.1.). „*Aš asmeniškai stengiuos beveik visada turbūt galvot*“ (I.4.). „Tai visada” (I.6.) Vis dėlto, įvairūs pašnekovai skirtingai apibūdino jų programas naudojančius asmenis. Remiantis Deleuze‘o divido sąvoka - jog iš vienos asmens galima sukurti skirtingų reprezentacijų[125] – ieškota kokiomis reprezentacijomis ML programuotojai suvokia jų produktų naudotojus ir kokiais vertybines įtampas tai sukelia.

Naudotojas kaip agentiškumas. Vienas pašnekovų, dažnai minėjęs žodį „rizika“, į naudotoją žiūrėjo kaip į potencialų pelną, kuris bet kada gali pranykti (atsisakyti ML programos paslaugų), dėl to ir savo ML kūrimo veiksmus projektavo taip, kad naudotojas neatsisakytų paslaugų. „*Tarkim rizika, kad tavo rekomendacijos padarys taip, kad klientai churn'ins* (churn – naudotojai, atsisakantys verslo paslaugų[126]) *kažkur vidury proceso, tai tarkim pamatys tavo rekomendacijas ir pradės išeidinėt, <...> Tai iš esmės rizika prarasti klientą, prarasti pinigus.“ (I.1.)*. Tad šis pašnekovas naudotojus mato kaip paslaugų atsisakymo rizikos bei pelno dividą. Šiuo atveju šie dividai yra įtampoje, nes siekiama didinti pelną aktyvinant naudotoją kaip pelningo elgesio dividą. Bet kartu tai daroma atsargiai, kad pelningumą skatinantys ML pokyčiai neatbaidytų naudotojo. Tai panašu į anksčiau minėtą W. Chun teorijos paralelę, jog vienu metu siekiama išlaikyti senus naudotojų įpročius, o kartu senųjų įpročių pagrindu kurti naujus[127].

Taip pat pastebėta, jog kalbant apie pačius naudotojus, nejučia jie būdavo sulyginami su skirtingomis veiksmo apibrėžtimis. Pavyzdžiui, paklausus, kada galvoja apie naudotoją, atsakė: „*Visą laiką, nes visą laiką galvoju apie metrikas.“*(I.2.). „*Tikriausiai visuose, nes <...> mąstai apie tam tikras metrikas. O metrikos yra <...> generuojamas tų pačių vartotojų*“ (I.12.) Čia padaromas naudotojo sulyginimas su metrika. Šiame kontekste metrika yra tam tikro įpročio sekimo apibrėžimas – pavyzdžiui, kiek laiko praleidžiama programėlėje, kiek produktų nusipirkta ir taip toliau.

Toliau naudotojas sulyginamas su įpročiais ar veiksmo intencija. *„Metrikų tikslas <..> kuo įmanoma apčiuopiamai apsibrėžti vartotojų sėkmę. Jeigu tu nori eiti paklausyti, atsidarai Spotify <...>,*

[125] Deleuze, „Postscript on the Societies of Control“, 5.

[126] „Cambridge Dictionary - Churn“, 2024 m. lapkričio 27 d., https://dictionary.cambridge.org/dictionary/english/churn.

[127] Wendy Hui Kyong, *Updating to Remain the Same: Habitual New Media*, 11–13.

tu jau turi intenciją. Ir klausimas ar mes galim išmatuoti, ar darydamas tą veiksmą mūsų platformoje, ar sugebėsi išpildyti tą intenciją?“ (I.2.). „*We can utilize that kind of information that is performed by someone like more like behavioral*”[128] (I.11.) Naudotojų suvokimas per elgsenos įpročius sutampa su Bruno ir Rodriguez įžvalgomis, jog divido kūrime, individo subjektyvūs bruožai yra apibrėžiami agentiškumais (angl. „agencies“)[129]. Tai lygiai taip pat koreliuoja su W. Chun kalbėjimu apie įpročių fiksavimą ir atnaujinimą[130].

Naudotojas kaip metrikos. Šios įvairios naudotojų sampratos yra svarbios, nes jos prisideda prie platesnių ML kūrimo tikslų apibrėžimo. Pavyzdys, kai pašnekovas, dalindamasis, ką matuoja metrikos, šį įpročių fiksavimo įrankį sulygino su būdu matuoti naudotojo pasitenkinimą: „k*aip <...> galbūt user'ių/vartotojų pasitenkinimas platforma padidės“ (*I.2.).

Pastebėjus, jog naudotojų įpročiai ir naudotojų pasitenkinimas nėra tapačios sąvokos, užklausta, kaip nustatoma, kad šis jo paminėtas pasitenkinimas/laimė padidėja. Pašnekovas iš kart pripažįsta, jog sunku pamatuoti, kas yra laimė, bet tai bando įvertinti pagal tai, ar ML prognozės padėjo naudotojui įgyvendinti programos naudojimo tikslus. *„Kas yra laimingesnis. Manau, čia yra labai sudėtingas klausimas. Mes žiūrime, kad vartotojai ateina į mūsų produktą daryti kažkokį pardavimą arba pirkimą. Ir jeigu jie sugeba nupirkti ar parduoti daugiau ar mažiau, tai reiškia, jie pasiekia savo tikslus“*(I.2.). *„Mąstai apie kažkokius tikslus, o tikslai yra pasiekiami per žmones <...> Kuriame sistemas, kad naudotų žmonės arba padėtų žmonėms.“*(I.12.)

Toliau klausiant, ar yra kitų būdų nustatyti laimę, pašnekovas dar aiškiau pripažįsta, jog „*klausimas yra sudėtingas, nes neaišku, kas yra laimingumas. <...> Jeigu būtų labai konkretus apibrėžimas, tai galėčiau duoti labai konkretų atsakymą į šitą klausimą*“. (I.12.)

Viską peržvelgus, galima sudėti tokią mąstymo seką. Pašnekovas davė vieną pavyzdį, kada ML rezultatai vertinami pagal tai, ar didėja naudotojų pasitenkinimas. Tam nustatyti reikia suprasti, kas yra pasitenkinimas/laimė. Tačiau pripažįstama, jog į tai atsakyti sunku. Nepaisant to, minėtoje ML sistemoje laimė apibrėžiama per naudotojų tikslų (t.y. nupirkti ar parduoti) įvykdymą.

Programuotojas iškelia hipotezę, kad daugiau parduotų produktų naudotojams suteikia pasitenkinimo ar net laimės. Kodėl jis pasirenka tokias žmogiškas, plačias bei sunkiai matematiškai apibrėžiamas sąvokas, o ne ką nors lengviau apibrėžiamo, kaip pardavimo sėkmė? Juk išgirdus programuotojo mintį, kyla klausimas, ar pasitenkinimas ir laimė tikrai yra tapatu pragmatinių tikslų įgyvendinimui? Atrodo, jog ir pašnekovas tai supranta ir galbūt savo pasirinkimą pagrindžia tuo, jog

[128] Vertimas: „Galime panaudoti tokią informaciją, kuri yra panaši į elgesį“
[129] Bruno ir Rodríguez, „The Dividual“, 36.
[130] Wendy Hui Kyong, *Updating to Remain the Same: Habitual New Media*, 11–13.

laimė neturi konkretaus apibrėžimo dėl to geriau, nei per tikslų įgyvendinimą, naudotojo laimės reprezentuoti neįmanoma. Bet geresnės interpretacijos nebuvimas nepaaiškina, kodėl pačioje pradžioje, išgirdęs klausimą, kaip vertina ML rezultatus, programuotojas pasitelkė naudotojo pasitenkinimo kriterijų.

Galima interpretacija, jog ML siekia optimizuoti, atkartoti ar kitaip technologiškai įprasminti plačias kultūrines kategorijas, kaip intelektas, pasitenkinimas, laimė ir pan. Tačiau šios sąvokos yra kompleksiškos. Tuo tarpu palyginus su šiomis sąvokomis, ML veikimo principai yra gana primityvūs ir apibrėžiami siauromis matematinėmis formulėmis. Kitaip tariant atrodo, jog kurdami ML, programuotojai mąsto apie šių plačių žmogiškumo sąvokų įgyvendinimą, bet nesugebėdami jų reprezentuoti technologiškai, pasirenka vieną iš tos plačios sąvokos bruožų ir jį laiko tapačiu pačiai sąvokai, o ne tik kaip bruožą.

Bet klausimas išlieka – kodėl programuotojams konceptualizuojant ML tikslus, norisi apie juos kalbėti plačiomis ir kompleksiškomis asmens patirčių sąvokomis (kaip kad „*vartotojų pasitenkinimas“)*, nors praktiniu žvilgsniu, šios programos kuria ne tai, o tiks siaurus tų sąvokų bruožų aprašymus (pardavimo ar pirkimo įvykdymas).

Divido sąvoka suteikia papildomą interpretacinį lygį. Teorija sako, jog dividas (kaip konkreti asmens veiklos informacija) gaunamas iš individo[131], kitaip tariant veiklos abstrakcija sukuriama iš integralaus asmens. Tačiau nors individas vis dar yra pats asmuo, dividas tuo tarpu jau tėra to asmens siauros veiklos formos reprezentacija, iš kurios nebeįmanoma atkurti integralaus asmens[132].

Šiame atvejyje būtų galima iškelti paralelę, jog ML programuotojai bendruosius ML kūrimo tikslus konceptualizuoja asmens (o ne dividų) bruožų lygmeniu – intelektas, naudotojo pasitenkinimas, laimė. Tačiau praktiniu lygiu, šie bruožai yra redukuojami į dividines, įpročių kategorijas, kaip kad parduotų/nupirktų produktų skaičius ir taip toliau. Būtent, kaip rašė Bruno ir Rodriguez, jog dividas nėra atskirtas nuo individo, bet gyvuoja santykyje su individu[133], taip ir praktiniai ML įgyvendinimo tikslai (kaip pardavimas/pirkimas) egzistuoja tik atsiremdami į platesnį ML tikslo suvokimą, paremtą naudotojo kaip asmens ar individo, integraliu suvokimu.

Naudotojų suvokimų įvairumas. Divido teorija kalba, jog iš vieno asmens (šiuo atveju per duomenizavimą) sukuriama daug skirtingų interpretacijų[134]. Tai paminėjo ir programuotojas,

[131] Bruno ir Rodríguez, „The Dividual“, 43.
[132] Bruno ir Rodríguez, 43.
[133] Bruno ir Rodríguez, 42–43.
[134] Deleuze, „Postscript on the Societies of Control“, 5.

kalbėdamas apie IT inžinerijoje naudojamą personų kūrimo praktiką[135], kai programuotojai vienos sistemos naudotojams bando nuspėti įvairias galimas sistemos naudotojų kombinacijas (įpročius, poreikius ir pan.) „B*andai <...> identifikuoti tas personas skirtingas. <...> Sakai, mano persona yra toks ir toks apsibrėži, kokis tris keturias skirtingas personas ir. Ir iš kiekvienos tos personos perspektyvos žiūri, kad tas sprendimas atitiktų jų lūkesčius*“ (I.1.).

Apibendrinimas. Programuotojai apie ML naudotojus kalba kaip apie integralius asmenis ir atitinkamai ML kūrimo tikslus sieja su žmogiškų savybių reprodukavimu – naudotojo pasitenkinimą, tikslų įgyvendinimą palengvinančių statistinių prognozių kūrimu. Tačiau pradedant kalbėti apie praktinį ML įgyvendinimą, pastebima, jog naudotojų suvokimas staiga redukuojamas iki jų agentiškumo – duomenyse užfiksuotinų įpročių, kaip atliktų konkrečių veiksmų kiekis ir panašiai. Taigi, egzistuoja naudotojų konceptualizavimo įtampa – viena vertus naudotojai suprantami kaip pilnavertės asmenybės, tačiau kartu ir tik kaip duomenų rėžiai.

Tai leidžia apžvelgti pačių naudotojų galių santykį su ML kūrimu ir rezultatais, ML kaip technokratinės galios įrankyje. Atsakymas yra ML naudotojo kaip asmens redukavimas į vis siauresnius asmenybės abstrakcijos lygius. Pirmasis lygmuo yra pats ML naudotojas kaip asmuo - integralus, neredukuotinas, daugiau nei tiesiog tam tikro asmeniškumo egzempliorius[136]. Tačiau norint naudotis ML, šiam žmogui tenka savo asmeniškumą susiaurinti iki individo lygmens. Dėl to, nes reikia naudotis institucija (šiuo atveju institucija yra skaitmeninė programa), kuri, pagal Foucault, jį kontroliuoja ir paverčia individu[137]. Vis dėlto, nors ir veikdamas kaip kontroliuojamas ir duomenizacijos sekamas individas, jis vis dar yra asmuo – t.y. asmuo turi tiesioginę įtaką tam, kaip veiks individas ir gali suprasti, kokiose aplinkybėse jis kaip individas veikia.

Tačiau vėliau ML panaudoja šio individo duomenis ir iš jų sukuria daugybes šio asmens veiksmus atitinkančių duomenų reprezentacijų (dividų). Čia asmuo/individas jau praranda tiesioginį ryšį su jo dividu (duomenizuota reprezentacija), kadangi ML naudotojas nežino kokie duomenys apie jį sukuriami, kaip apdodorojami ar panaudojami. T.y. asmeniškumas ir individualumas nesusijungia[138] ir veiksmas yra tik vienkryptis – dividas sukuriamas iš individo, tačiau pats individas šio divido nepažįsta.

Tai papildo W. Chun mintį, jog sistemose, kaip ML, naudotojų politinis pasipriešinimas tyla ir neaktyvumu neįmanomas. Chun sako, jog taip yra dėl to, nes net naudotojo nepasirinkimai yra

[135] Edna Dias Canedo ir kt., „Use of Journey Maps and Personas in Software Requirements Elicitation“, *International Journal of Software Engineering and Knowledge Engineering* 33 (2023 m.): 1.

[136] Robert Spaemann ir Oliver O’Donovan, *Persons: The Difference between `Someone’ and `Something’* (Oxford University Press, 2017), 32.

[137] Foucault, *Disciplinuoti ir bausti : kalėjimo gimimas*, 172.

[138] Bruno ir Rodríguez, „The Dividual“, 43.

paverčiami duomenimis[139]. Divido teorija tai papildytų ir tuo, kad ML naudotojas (kaip individas) negali pasipriešinti, kadangi jis nepažįsta tų dividų, kurie iš jo yra sukuriami. Tad asmuo net negali žinoti, kaip jis iš tiesų yra interpretuojamas pačioje ML sistemoje.

3.5.2. Programuotojų nemąstymas apie naudotoją

Nors dauguma atsakė, jog apie naudotoją mąsto praktiškai visada, dalis pašnekovų pasidalijo priešingai arba pasimetė. Pavyzdžiui: „*Nu jo, kaip ir pagalvoju, bet tai dažniausiai būna tas... <...> Čia galbūt toks labiau klausimas turėtų būti, kai į website'ą kažkokį modelį įdedi <...> I.: Tai realiai taip išeina, kad <...> kai įmonei darot šituos modelius, tai tada kažkiek mažiau tokio akcento apie galutinį naudotoją gaunasi? P.: Jo jo*" (I.8.) Pašnekovas pripažino, jog kai ML kuriamas ne tiesiogiai klientui, bet tarpinei įmonei, pavyzdžiui prognozuoti įmonės veiklą, bet grįžtamojo ryšio apie prognozių panaudojimą nėra, tada apie naudotoją mąstoma mažiau.

Dalintasi ir spėjimu, kad mažiau apie naudotoją gali mąstyti tie, kuriems svarbiau tik įgyvendinti techninius tikslus. „*Yra analitikų, kuriem galbūt nėra tas labai svarbu. Galbūt daugiau į metrikas, kaip greičiau pasibaigt tuos darbus, kad būtų galima eit kitus pakeitimus daryti.*" (I.9.). Kitaip tariant atvejai, kada apie naudotoją mąstoma mažiau, yra kai trūksta tiesioginio ryšio su pačiu naudotoju.

Tai iliustruoja pašnekovo, dirbančio stipriai hierarchinėje kompanijoje pavyzdys. Dėl šio struktūros, jis nesulaukia grįžtamojo ryšio, kaip naudotojams sekasi naudoti ML: „*Aš gana žemai toje hierarchijoje <...> Tokio grynai tiesioginio ryšio per daug negirdžiu*" (I.10.). Taip pat, kadangi pašnekovas dirba Japonijoje, pripažįsta, jog tokia hierarchija ir kliento nepažinojimas gali būti šalies specifika.

Pašnekovas dalijasi, jog dėl tokio atskirtumo "*kartais atsiranda sunkumų tiesiog prioritetus dėliojant užduotyse*" (I.10.). Norima suprasti, kokie yra būsimo ML naudotojo poreikiai, tačiau to padaryti efektyviai nepavyksta negaunat grįžtamojo ryšio: "*Mes negalime net jų pačių* (naudotojų) *paklausti apie tai <...>. Bet tada mes negalime jiems duoti pagal jų poreikius*" (I.10.).

Čia matoma, jog verslui laikant kliento privatumą svarbiausia vertybe, programuotojai neturi galimybės net sužinoti, kur yra naudojamas jų kuriamas ML. Dėl to susiduriama su iššūkiais įgyvendinant naudotojų poreikius. Šiuo atveju sprendimų priėmimas panašu vyksta jau nebe programuotojų, o verslo lygyje. Tai parodo kitokią programavimo kaip kultūrinio veiksmo perspektyvą - klientų bei programos panaudojimo situacijų neatskleidimas egzistuoja būtent dėl vietinės hierarchinės kultūros ir kompanijos galių pasiskirstymo. Norint analizuoti ML kaip technokratinės galios veikėją,

[139] Wendy Hui Kyong, *Updating to Remain the Same: Habitual New Media*, 29.

galutinė tokių ML įtaka naudotojui labiau yra apsprendžiama paties verslo, kaip jie nuspręs panaudoti produktą.

Kaip bebūtų, programuotojai, nors ir nežinodami apie savo klientą, vis tiek yra tie, kurie duoda pradžią ML sukūrimui ir tai vis tiek prisideda prie galutinio ML veikimo rezultato, o ir poveikio naudotojui. Tad tokia hierarchinė struktūra komplikuoja galios pasiskirstymo ML inžinerijoje suvokimą.

Taigi šie pavyzdžiai parodo, kad jei ML programuotojai neturi galimybės gauti grįžtamojo ryšio iš jų programų naudotojų, tuomet ir ML sprendimai yra kuriami vis mažiau mąstant apie galutinį naudotoją bei įtaką jam. Tokiu atveju verslas, kaip tarpininkas, daro didesnę įtaką ML panaudojimui ir šių programų poveikiui naudotojams.

Apibendrinimas. Kai kuriais atvejais ML programuotojai praktiškai neturi galimybės susipažinti su programą naudojančiais asmenimis. Tai nutinka tada, kai ML kuriantieji tiesioginio grįžtamojo ryšio nelaiko vertybe.

Šiais atvejais programuotojai mažiau dėmesio skiria įsijausti į pačių sistemos naudotojų perspektyvą, o daugiau dėmesio skiria galvoti tik apie techninį ML įgyvendinimą. Tai sukuria didesnę skirtį tarp ML naudojančio asmens ir jo duomenų reprezentacijos divido. Dabar net ir patys programuotojai neturi informacijos apie naudotojus ir jų reakcijas į ML prognozes. Tad šioje situacijoje divido sukūrimo veiksmas nebe vienkryptis (kai dividas kuriamas iš individo, bet ne atvirkščiai), bet atrodo santykis iš vis nutrūksta. Naudotojo dividiška reprezentacija sukuriama, tačiau kiek ji tiksli realiam asmeniui – to pasakyti negali net pats jos kūrėjas. Šiuo atveju ML naudotojų galia suvokti įtaką, daromą ML rezultatams yra dar silpnesnė ar potencialiai net neegzistuoja.

3.6. Išimčių taikymas įprastoms ML kūrimo praktikoms

3.6.1. Koreliacija = priežastingumui?

Statistikos moksle (kuriuo paremta ML) žinoma, jog paprasčiausias duomenų statistinis reikšmingumas dar nereiškia, kad jie daro įtaką vieni kitiems. Jei kuriant ML, koreliacijos priežastingumas bus nepatikrintas, kyla grėsmė naudotojams pateikti prognozes, nepagrįstas jokiais faktais ar logika. Tai ne tik būtų neteisinga naudotojų atžvilgiu, bet ir sukurtų neprognozuojamą poveikį jiems. Tačiau taip pat kyla klausimų, ar iš vis įmanoma užtikrinti, kad ML prognozės visada būtų paremtos patikima empirika. Pašnekovų paklausus, ar tai dar tiesa, t.y. „ar koreliacija vis dar nelygu priežastingumui“ (fraze „vis dar“ sufleruojant klausimą, ar ML praktikoje šios teorijos dar laikomasi, kai atsirado „big data“) - dalis tai pripažino, kai kas pateikė išimčių, o dalis patys nepastebėjo, jog praktikoje daro išimtį šiai taisyklei.

Koreliacija – ne priežastingumas, bet panaudoti galima. Kai kas neneigė statistinės taisyklės, tačiau pripažino, kad priežastingumu neįrodyta koreliacija gali duoti naudos ML prognozėms, tuomet tai ir panaudos. „J*eigu tu tris ar keturis metus iš eilės gali pasakyti apie tam tikrus dalykus, kad jie koreliuos, bet nėra priežastiniai, ar ne, tu tuos tris ar keturis metus gali juos gerai papredict'int <...> Nesvarbu, ar ten priežastinis ryšys, ar ne. <...> nereiktų visiškai užsimerkti ir sakyti, kad iš to negalima jokios naudos išpešti.*“ (I.1.). Čia vėl pasikartoja duomenų potencialios naudos motyvas. Šiuo motyvu pašnekovas tarsi argumentuoja, jog būtų neapdairu palikti nepanaudotą duomenų ryšį.

Kitas pašnekovas pritarė, jog ML prognozėms kurti koreliacija gali būti panaudota, tačiau svarbu įsivertinti, kad tai nebūtų daroma didelės rizikos situacijose, kaip medicinoje. “*Bandant nuspėti kažką, tai jo koreliacijos pilnai užtenka*. *<...> Šnekant apie tą pačią mediciną <...> Žinai, kad tu gali surast kažkokios koreliacijos, bet nereiškia, kad ta koreliacija iš tikrųjų sukelia tą. <...> medicinoje tau labai svarbu žinoti, kas sukelia, nes tu žinai, tu negali tiesiog sakyti šitas dalykas koreliuoja*“ (I.4.). Tačiau ar tai reiškia, jog mažiau rizikingose srityse galima pasikliauti ir vien koreliacija? Kaip nuspręsti, kuri sritis verta priežastingumo pažinimo, o kuri ne?

Vėliau pašnekovas atskleidė, jog kasdienybėje iš ties tenka panaudoti tiesiog koreliaciją. Tai nutinka tais atvejais, kai dėl privatumo tam tikrų duomenų naudoti negalima. Tuomet naudojami tarpiniai duomenys – duomenys skirtingi nuo tų, kurių naudoti negalima, bet tokie, kurie koreliuoja su dėl privatumo saugomais. „*Kartais <...>, tu vis tiek jį (*duomenis, kurių dėl privatumo tiesiogiai naudoti negalima) *turi pasiimti kažkaip <...> kartais gali būti net ne išvestinis iš tam tikro dalyko <...> bet tiesiog kažkoks panašus feature'as, kuris koreliuoja*“(I.4.). Atrodytų, jog šiuo atveju pašnekovas nemato grėsmės naudoti tarpinius duomenis, tačiau yra ir priešingų nuomonių.

Tarpinių reikšmių naudojimas toks pat neetiškas kaip tiesioginė koreliacija. Pasidalijimas pabrėžia, jog nors prognozuoti pagal jautrius bruožus, kaip lytį, jau yra draudžiama, pašnekovė pripažįsta, kad per tarpines reikšmes šiuos jautrius duomenis panaudoti vis tiek įmanoma: „*Jau yra uždraudžiami, ypatingai lyties diskriminaciniai, kad negalima jų naudoti <...> Bet aš pati žinau, kad tu gali netiesiogiai panaudoti iš kitų kintamųjų išsivesti, kad čia yra, tarkim, moteris arba vyras*.” (I.5.). I.4. atvejis rodo, jog programuotojas naudotojų duomenų privatumą laikė vertybe. Tačiau jis nesusimąstė, jog tarpinių išvesčių naudojimas pažeidžia jo paties aukštai laikomą asmenų privatumo vertybę. Šis atvejis iliustruoja, kaip ML etika susirūpinęs programuotojas nepastebi slypinčių etinių grėsmių.

Galimos to priežastys yra per daug siauras savo vertybių suvokimas. Kas jei I.4. pašnekovas privatumą būtų laikęs platesne sąvoka, nei asmeninių duomenų nutekėjimo prevencija, bet išplėtęs iki I.5. sąvokos – socialinių grupių skirstymo ir diskriminacinės personalizacijos prevencijos? Gal tada jis būtų supratęs, jog tarpinių duomenų naudojimas pasiekti „naudingiems“ duomenims, taip pat pažeidžia jam rūpinčią vertybę.

Nepasitikėjimas koreliacijomis. Buvo ir pripažinusių bet kokių koreliacijų nepatikimumą. Pirmiausia dėl to, nes statistiką duomenų mokslininkai dažnai parodo jiems parankiu kampu. O net ir tais atvejais, kai koreliacija iš tikro atskleidžia priežastį, pašnekovės nuomone dažnai tai kažkas savaime suprantamo. „*Parodo, tarkim, kažkokią kreivę <...> vienas sako, kad žiūrėk, kaip auga, kitas sako žiūrėk, ten stovi vietoje <...> čia tiesiog arba tiki, arba netiki <...> Aš išvis koreliacijom net nu nelabai tikiu. <...> Sako, žiūrėk, radau, koreliuoja. <...> Bet ar tai paaiškina? Retais atvejais taip, tai paaiškina. Gali atskleisti kažkokią priežastį, kad, pavyzdžiui, kad kai lyja, gali sušlapti.*”. (I.6.) Pašnekovė teigia, jog duomenų interpretacija labiau priklauso nuo to, kuo žmogus tiki – t.y. kokius rezultatus nori rasti. Ir atitinkamai, tokius rezultatus juose ir ras.

Toliau ji tęsia apie atvejį, kai koreliacija buvo sąmoningai pateikta taip, kad įrodytų asmens įsitikinimus: „*Vienas vadovas <...> susigalvojo, kad ten vienas rodiklis turi kristi. Ir tada jis turi tapti to rodiklio gelbėtoju. Ir jis paprašė <...>, kad jam padarytų tokį grafikėlį. <...> Jis nuėjo pas CEO, ten padarė darbo grupę <...> Žiūriu, ir man kažkas yra tikrai ne taip tame grafike. <...> Paėmiau kompe. Greitai padariau tas ašis, kad abi nuo nulio ir taip gražiai taip tiesė <...> žemyn ėjo, ir taip gražiai išsitiesina <...> nėra problemos*” (I.6.). Tad galiausiai, pašnekovė teigia, jog įsitikinimai yra pirmesni, statistika naudojama ne pažinti pasaulį, bet įrodyti savo pasaulio suvokimo teisingumą.

Koreliacija tinka prognozės, bet ne intervencijos tikslui. Pastarąją įžvalga papildo pasidalijimas, jog koreliaciją galima naudoti kuriant spėjimus. Tačiau dažniausia užsakovai ML kuria ne dėl prognozių, o dėl intervencijų.t.y. siekia kurti ML, kuris darys įtaką naudotojų įpročiams, o ne vien

juos nuspės. „T*hat's fine, if you just want to predict. But in many cases people don't want to just predict but also intervene <…> People say that they want prediction models, 80% of the time, they don't. They actually want causal models*“[140] (I.7.)

Pagal W. Chun teoriją, tokios technologijos kaip ML kuria prognozes, remiantis egzistuojančiais naudotojų įpročiais[141], t.y. remiantis priežastingumu. Kaip rodo skyrius „Programuotojų suvokimas apie naudotoją – tarp asmenybės ir įrankio“, dėl naudotojų redukavimo į jų įpročius (duomenizavimo), net jau priežastingumu paremtas ML sukuria erdvę, kur pats ML naudotojas turi mažai galios pažinti apie jį sukurtą dividą, tad ir mažesnę galią pasipriešinti[142]. Tad jei ML yra kuriamas nebe remiantis patikrintomis priežastimis, o tik koreliacijų spėjimais, skirtis tarp naudotojo kaip asmens ir jo divido išsiplečia dar labiau. Nes ML kuriamos prognozės gali nebeturėti jokio ryšio tarp naudotojo ir jo reprezentacijos ML programoje.

Nepastebi, jog patys naudoja koreliacijas. I.8. pašnekovas pasitikinčiai atsakė, jog koreliacija tikrai nelygi priežastingumui. Tačiau vėliau atviravo apie ML rezultatų interpretavimą, t.y. priežastingumo nuspėjimą pagal koreliacijas. „*Naudingiausi yra modeliai tie, kuriuos gali žmogus interpretuoti <…> Kur matai, kokie kintamieji labiausiai įtakoja ir tada. Gali pagal tai dar papildomą analizę daryti ar kažkokias išvadas*“ (I.8.). Tai prieštarauja ankstesniems pašnekovų pasidalijimams, jog interpretacija dar nereiškia priežastingumo (I.6.) ar, kad koreliacijos yra gerai tol, kol tai tik prognozė, bet ne įpročių intervencija (I.7.). Nors šiuo atveju pašnekovas nesakė, ar interpretacijas naudoja asmenų įpročių keitimui, interpretacijos veiksmas yra aiškus, kas rodo nepakankamą įsigilinimą į koreliacijų ir priežastingumo ryšio trapumą.

Verta pastebėti, jog I.8., ne taip kaip dauguma kitų pašnekovų, neturi tiesioginio ryšio su klientu. Galbūt tai yra viena iš priežasčių, kodėl šiam klausimui neskiriama tiek daug dėmesio ir kyla rizika nepastebėti, jog kartais koreliacijos yra palaikomos priežastimis.

Priežastingumai išgryninami naudotojų grįžtamuoju ryšiu. Kaip bebūtų, kai kurie programuotojai priežastingumui užtikrinti naudoja testus, paremtus naudotojų grįžtamuoju ryšiu. Tai vadinamieji „AB testai“, kai atlikus ML pakeitimą dalis naudotojų naudoja seną programos versiją, dalis naują. Tik jei pastebimas pokytis naujosios versijos naudotojų elgesyje, tada tikima, kad yra priežastingumas[143]. „*Negali žinoti, nepadaręs AB testą, kur tu pamatuoji seną ir naują modelį*“ (I.12.);

[140] Vertimas: „Gerai, jei norite tik nuspėti. Tačiau daugeliu atvejų žmonės nori ne tik prognozuoti, bet ir daryti įtaką <...> Žmonės sako, kad nori prognozavimo modelių, tačiau 80 proc. jie nori ne toIš tikrųjų jie nori priežastinių modelių“

[141] Wendy Hui Kyong, *Updating to Remain the Same: Habitual New Media*, 11–13.

[142] Wendy Hui Kyong, 29.

[143] Federico Quin ir kt., „A/B Testing: A Systematic Literature Review“, *Journal of Systems and Software* 211 (2024 m. gegužės): 1, https://doi.org/10.1016/j.jss.2024.112011.

„*Tu gali optimizuoti kažkokią trumpalaikę metriką, kuri tau atrodys ji koreliuoja <...>, bet iš tikrųjų tai nėra taip. Ir tau dėl to reikia daryti eksperimentus*“ (I.2.).

Atrodytų, jog dirbant grįžtamuoju ryšiu paremtu AB testavimu, ne tik nedidėja grėsmė dar labiau didinti skirtį tarp realaus naudotojo ir jo reprezentacijos divido, bet ši skirtis potencialiai netgi mažėja. Juk atliekant testus, pagal padarytas prielaidas sukurtas ML pateikiamas jau nebe dividui, bet realiam naudotojui – ar ML prognozės priežastingumas bus patvirtintas priklauso nuo to, kaip naudotojas elgsis gavęs ML atnaujinimą. T.y. ar jo įpročiai pasikeis pagal padarytas hipotezes ar ne. Tad nors ir netiesiogiai, suteikiama galimybė naudotojų reakcijoms į ML prognozes daryti įtaką ML kūrimo procesams. Nes jei naudotojams ML prognozės bus nepriimtinos – t.y. ML pasiūlymų jie nepaklausys ir jų įpročiai niekaip nepasikeis – programuotojai tai pastebėję atsisakys ML pakeitimų.

Žinoma, patys naudotojai nežino, būtent kurie jų įpročiai yra matuojami testuose. Dėl to ir negali tiesiogiai daryti įtakos ir tai vis dar kelia problemų naudotojų galios atžvilgiu. Tačiau vis tiek žinoma, jog ML kūrimas remiasi ne tik asmenų abstrahavimu ir vertimu iš individų į dividus (t.y. duomenis), bet ir atliekamas divido (ML prognozės) atitikimo asmeniui testavimas.

Apibendrinimas. ML technologija tam tikra prasme yra paremta statistinių koreliacijų atpažinimu ir prognozių kūrimu pagal jas. Tačiau statistikos mokslas sako, jog vien koreliacijų atrasti nepakanka – reikia ir įrodyti, jog jos turi priežastinį ryšį. Šis įrodymas yra dar vienas įtampos taškas ML inžinerijoje – kaip programuotojams užtikrinti, jog jų programų prognozės iš tiesų paremtos priežastingumu, o ne vien koreliacijomis.

Pirmiausia pastebėta, jog pašnekovai iš tiesų susipažinę su koreliacijos ir priežastingumo teorija. Tačiau tik dalis šios taisyklės laikosi be išimčių. Kiti mano, jog jos laikosi taip pat, tačiau iš tikrųjų daro išimčių tais atvejais, kai norėdami išsaugoti naudotojų privačią informaciją, bet kartu ir norėdami ją panaudoti, padaro išimtį ir su privačiais duomenimis koreliuojantį bruožą laiko beveik tapačiu. Šiuo atveju padaroma išimtis ne tik koreliacijos taisyklei, bet ir privatumo vertybei, kadangi rezultatas lieka tas pats – nors ir per tarpinius duomenis, panaudojami tie asmens bruožai, kurie iš tikrųjų turėtų likti privatūs. Galiausiai, buvo kas pripažino, jog koreliacija nereiškia priežastingumo, tačiau tokius duomenis vis tiek galima panaudoti verslo naudai sukurti. Čia programuotojams verslo nauda buvo stipresnė vertybė nei statistikos mokslo taisyklių išsaugojimas.

Dalis kalbintųjų šio klausimo potekstės iš vis nesuprato. Šių pašnekovų darbo aplinka skiriasi nuo anksčiau minėtųjų tuo, jog grįžtamojo ryšio iš naudotojų jie negauna, o tiesiog sukuria ML, jį pateikia verslui ir patys nežino, ar ši programa patenkino naudotojų poreikius, ar ne. Gali būti, jog neturint grįžtamojo ryšio iš naudotojų, programuotojai net nesusimąstydami patiria polinkį išnaudoti koreliacijas.

Taip yra dėl to, nes jie neturi galimybės patikrinti, ar tikrai tam tikrų duomenų statistinis panašumas iš tikrųjų reiškia priežastinį ryšį.

Vis dėlto pastebėta, jog inžinieriai, kurie remdamiesi naudotojų grįžtamuoju ryšiu daro koreliacijų hipotezių patikrinimus su naudotojais („AB testus“) tiki, jog gali iš tikrųjų pastebėti, kurie duomenys tiesiog statistiškai panašūs, o kurie iš tiesų turi priežastinį ryšį. Toks darbo procesas atrodo kaip potencialus būdas sumažinti skirtį tarp realaus sistemos naudotojo ir jo reprezentacijos (divido). Divido pritaikymas duomenizavimui leidžia suvokti, kad iš naudotojo kuriant duomenis (jo dividą), dividas galbūt ir kuriamas pagal asmens bruožus, tačiau pats asmuo nežino pagal kuriuos jo bruožus bei kaip šie bruožai, slypintys divide, bus panaudoti ML prognozėms kurti.

Pavyzdžiui, naudotoja nežino, ar duomenizuojama, kiek laiko ji praleido žiūrėdama vaizdo įrašus ar kiek skirtingų vaizdo įrašų ji peržiūrėjo. Jei ji pamatys vaizdo įrašo rekomendaciją su pavyzdžiui politiniu turiniu, ji nežinos, ar tai dėl to, kad tokio tipo turinį žiūrėjo kelias valandas ar dėl to, kad tik netyčia spustelėjo vieną kartą. Dėl to divido ir paties asmens santykis yra vienpusis – asmuo nepažįsta divido.

Tuo tarpu atliekant AB testus, atliekamos prognozės lyginamos su tikro naudotojo reakcijomis. Jei naudotojui nepatiko ML prognozė (t.y. rekomenduojamas minėtas politinis turinys nėra žiūrimas), programuotojai tai pamatę supranta, jog prognozė buvo paremta prielaida. Pavyzdžiui prielaida, jog vieno vaizdo įrašo spustelėjimo gali užtekti įrodyti pomėgį, neįtraukiant atsitiktinio paspaudimo galimybės. Tad šiuo atveju svarbus ir paties asmens, pagal kurį kuriami duomenys-dividas, reakcija į šį dividą. Tokiu atveju pradedama artėti prie abipusio ryšio. Dividas kuriamas pagal naudotojo abstrakciją (dividas „pažįsta“ asmenį), tačiau, ar tas dividas bus priimtas, priklauso nuo asmens reakcijos į dividą (asmuo pažįsta ir patį dividą, per ML prognozę).

Tai kelia klausimų W. Chun minčiai, jog ML tipo medijose naudotojai neturi galimybės pasipriešinti net tyla[144]. AB testų atvejis kaip tik rodo, jog aktyvus naudotojų tylos fiksavimas leidžia naudotojams daryti įtaką tam, kokie ML pakeitimai bus priimti ar atmesti.

3.6.2. Išimtys procesams dėl verslo tikslų

Išimtys daugiau pelno kuriantiems klientams. Kaip jau pastebėta, ML kokybė yra vertinama apibrėžiant kokybės ribą, kuri kinta priklausomai nuo konteksto ir situacijos. Šioje dalyje pateikiamas pašnekovės atvejis, kai įmonė, šias ribas skirtingiems klientams pritaikydavo įvairiai, galimai priklausomai nuo kliento pelningumo.

[144] Wendy Hui Kyong, *Updating to Remain the Same: Habitual New Media*, 29.

Pašnekovė kūrė programą, teikiančią įvairių įmonių vertinimus. Šį ML kurianti kompanija save reprezentavo kaip tokią, kuri naudojantis ML panaikina visus melagingus vertinimus: „*pasižadėjusi, kad mes visus fake reviews triname*" (I.5) Programuotoja dalijosi, jog ML prognozės tikslumo riba, nuo kurios įvertinimas buvo pripažįstamas melagingu, buvo nustatyta gana aukšta. Taip buvo dėl to, nes šias paslaugas naudojančios įmonės ML kuriančiai kompanijai nešė pelną ir šios programos kūrėjai nenorėjo pykdyti savo klientų bei prarasti pelno, tad rinkosi pranešti tik apie itin užtikrintus melagienų atvejus. „M*atydavome, kur <...> didžiulį probability parodo, kad čia yra fake'iniai reviewsai <...> Bet tas treshold'as <...> buvo gana aukštas, nes tos įmonės perka pas mus tada paketą*" (I.5.).

Tad norint apsisaugoti nuo neteisingų įvertinimų apkaltinimų melagingais, buvo atliekamos papildomos žmonių patikros. „B*uvo ir atskira komanda, kurie <...> eidavo dar kart double check'int, nes labai bijodavo ką nors <..> apšaukti, kad iš tikrųjų jie nėra fake'iniai <...>*" (I.5.). Tačiau panašu, jog šiai papildomai patikrai buvo skiriama skirtingai dėmesio priklausomai nuo to, kiek pelno neša ML naudojantis klientas. „*Priklausomai nuo to <...> ar ji mums moka duos pinigus, ar ne - būdavo handle'inamos tos situacijos, kartais skirtingai. Tikrai ne visada. Ir aš žinau, kad trust buvo labai svarbi dalis. Bet tuo pačiu svarbi dalis yra ir ir tos pajamos ir yra ta baimė, kaip įrodyti, nes tada prasideda ir visokie kartais ir bylinėjimais*" (I.5.). Kitaip tariant, mažiau mokantieji iš ML galėdavo susilaukti dažnesnių melagingų įvertinimų užfiksavimų.

Šis atvejis parodo iki šiol aptartus įvairius ML inžinerijos įtampos taškus veikiančius kartu – kokybės ribos nustatymas, automatizuotos ar papildomos žmonių patikros parinkimas. Šie pasirinkimai buvo daromi remiantis pelno vertybe, kaip padaryti, kad klientai mokėtų kuo daugiau. Kaip bebūtų, tikslas tikrai nebuvo melagingus įvertinimus palikti daug mokantiems klientams, tačiau buvo siekiama apsisaugoti nuo konfliktų su klientais. Dėl to, kuo daugiau mokėjo klientas, tuo daugiau papildomos patikros buvo daroma melagienų paieškos rezultatams.

Regis problema slypi ne tame, jog dalis klientų įvertinimų liko su melagienomis. Pasak pašnekovės to išvengti neįmanoma, nes kaip ji sako, tikrą įvertinimą pavadinti melagingu taip pat yra nepriimtinas rezultatas. „T*reshold'us gali sumažinti, tu gali nemažai tų false positive gaut, o tai jau irgi yra visiškai not acceptable, bet lygiai taip pat not acceptable turėti įmones, kurios turi vien prifake'intus reviews*" (I.5.). Tad vėl pasireiškia ML neutralumo ir šališkumo įtampa – ar leisti programai veikti „neutraliai", tačiau su rizika, jog teisingas įvertinimas bus palaikytas melagingu. O gal pakelti ML melagingo įvertinimo ribą, kad būtų apsisaugota nuo nepelnytų apkaltinimų, bet prisiimti riziką, jog kai kurie melagingi įvertinimai liktų nepanaikinti?

Tačiau ši vertybinė įtampą susipina su kita – ar elgtis vienodai su visais klientais, nepriklausomai nuo jų pelningumo? Kaip dalintasi, kad svarstyta ar atsižvelgti į tai, kad daugiau pinigų mokantis klientas turėtų mažesnę rizikos tikimybę: „*Ar jis turi dar atsižvelgti, ar nu tipo žmogus moka pinigus ir <...> čia tada less less risky*" (I.5.).

Tai ypač keblu dėl to, nes ML kurianti įmonė jokių pažadų nesulaužo. Įvertinimo ribų nustatymas, tai įprastas ML kūrimo procesas, kas kart priklausantis nuo konteksto. Tad tai, jog skirtingais atvejais įmonė nustato skirtingas tikslumo ribas, iš tiesų nieko keisto.

Sąmoningas verslo tikslų skatinimas. Taip pat minėti atvejai toje pačioje kompanijoje, kai ML buvo koreguojamas taip, kad prognozė naudotojui sukeltų atitinkamą verslui naudingą elgesį. Dalintasi kaip siekta sugeneruoti ML prognozę, kuri rodytų kuo prastesnę kliento verslo ateitį. „*Tikslas buvo kaip prapushint tarkim brangesnius planus pagal kažkokį tai ten jų nesėkmės behavior ir panašiai. Buvo iš esmės rasti tuos neigiamus dalykus būtent tose įmonėse, kad kažkas joms ten neveikia*" (I.5.). Tai buvo daroma dėl to, kad klientą įtikintų, jog jų ML naudojantis produktas padės kliento verslui: "J*eigu jūs nenusipirksite šito, tai jūs <...> negausite <...> populiarumo. Ir useriai jumis nepasitikės*" (I.5.). Tam pasiekti, ML prognozei sukurti reikėjo naudoti ne visus kliento duomenis, bet tuos, kurie kurtų prasčiausią ateities prognozę: „L*abai daug reikėjo iš tiesų neigiamo input'o dėti, tarkim, į modelį. Taip jį diskriminacinį padaryt*" (I.5.)

Galbūt kai kam tai galėtų atrodyti kaip manipuliacija. Nors sunku būtų tai laikyti tiesa, duomenyse nebuvo meluojama ta prasme, jei melas būtų netikrų duomenų sukūrimas. Čia buvo parenkami parankūs duomenų rėžiai. Tačiau tai kelia duomenų objektyvumo-šališkumo įtampą. Tam tikri duomenys buvo panaikinti, kai kur koreliacija buvo palaikyta priežastingumu. Tačiau šie veiksmai pastebėti ir kituose ML kūrimo atvejuose, kurių manipuliacijomis vadinti nesinori. Galiausiai, juk duomenys vis tik nebuvo suklastoti – tik atrinkta tam tikra jų dalis. Kaip rodo tyrimas, visuose ML procesuose dalis duomenų lieka nepanaudoti, nes jau pats duomenų gavimas yra sukūrimas, o ne jų radimas[145].

Apibendrinimas. Darbe aptartos vertybinės įtampos neretai susipina į vieną, o galutiniam sprendimui įtaką gali daryti platesnės kultūrinės įtaka, kaip aptartuose atvejuose buvo pelno vertybė. Pateikti pavyzdžiai rodo duomenų rinkimo, ML vertinimo ir kitų inžinerinių sprendimų parinkimą remiantis ne tik iki šiol ištirtais vertybiniais pasirinkimais, bet ir atliepiant siekį minimizuoti kliento nuvylimo ir potencialaus pelno praradimo galimybę. Galiausiai apžvelgtas atvejis, kai ML rekomendacija modifikuota taip, kad paveiktų klientą rinktis siūlomą produktą kaip galimų problemų sprendimą.

[145] Mejias ir Couldry, „Datafication", 2.

Palyginus šį atvejį su kitais ML panaudojimais, pavyzdžiui minėtu rūbų pirkinių rekomendacijos ML. Kuo šie atvejai skiriasi? Vienu atveju tikslas yra skatinti naudotoją pagalvoti, jog jo organizacijai sekasi prastai, dėl to jam reikia įsigyti siūlomas paslaugas. Kitu atveju tikslas pasiūlyti įsigyti rūbų, kuriuos pamatęs naudotojas tikrai pagalvotų, jog tai vertingas pirkinys. Abu sprendimai siekia įtikinti pirkti.

Žinoma, situacijos tikrai nėra tapačios. Bet kas skiriasi, būtent ML kuriančių verslų bei inžinierių intencijos. Tuo tarpu inžineriniai procesai – panašūs. Abi ML atrenka tam tikrą siaurą naudotojų duomenų grupę, nustato ribą, kada programa yra pakankamai tiksli bei siekia tokių ML rezultatų, kurie išpildytų verslo siekiamą rekomendaciją. Tai parodo, jog ML kūrimas iš tikrųjų yra kultūrinis procesas ir kad nuo ML kūrėjų vertybinių pasirinkimų priklauso programos rezultatas. Taip pat liudija, kad tiriant šių programų socialinį poveikį, svarbu analizuoti priimtus techninius sprendimus, bet dar svarbiau pažinti, kokia buvo šių inžinerinių sprendimų intencija. Šiuo atžvilgiu galbūt programuotojai tikrai sako tiesą, jog pats ML yra „agnostiškas problemai“, „nei geras, nei blogas“ ir viskas priklauso nuo panaudojimo tikslų bei intencijos. Vis dėlto, ši situacija skatina apmąstyti ir platesnį ML kultūros dėmenį – ar programuotojai intenciją rinkosi patys, ar spaudžiami verslo siekių, ar/kokia buvo galimybė pasipriešinti šiam spaudimui.

IŠVADOS

Mašininio mokymosi (ML) programų inžinerija – ne tik naujas technologinis laimėjimas, bet ir naujos sociotechninės problemos. ML programos, analizuodamos didelius duomenų rinkinius, pateikia prognozes apie ateitį, asmenų įpročius, būdus juos keisti ir tai daro socialinę įtaką individams ir visuomenei. Nors šia problema susirūpinta, esami sprendimai nėra pakankami, kadangi apsiriboja tik ML rezultatų analize *post factum,* kai ML produktai jau yra sukurti. Tačiau mažai gilinamasi į pačią ML socialinės įtakos atsiradimo priežastį – kokia inžinerinių sprendimų seka sukuria vienokią ar kitokią socialinę įtaką naudotojui darančią programą.

Darbo tikslas: rasti įtampų taškus ML inžinerijos praktikoje – sprendimų priėmimo momentus, kuriuose programuotojai turi rinktis tarp kelių technologinių sprendimų, bei atskleisti, kaip ir kokiomis vertybėmis jie remiasi darydami šiuos pasirinkimus. Darbe remtasi Nick Seaver „algoritmų kaip kultūros“ samprata, siūlančia ML reikia matyti ne kaip vientisą kultūrą veikiantį veikėją, bet padriką, kompleksišką kultūrinį procesą savaime, veiksmą, sudarytą iš daugelio techninių/vertybinių sprendimų kovų, kurių rezultatai nemaža dalimi lemia ML socialinį poveikį. Norint atrasti įtampos taškus, analizuota ML kūrimo logiką pristatanti literatūra, taip pat - remtasi Wendy Chun algoritminių įpročių teorija bei Gilles’o Deleuze’o „divido“ teorija apie duomenizavimą.

Šios teorijos naudotos formuluoti klausimynui, atlikti pusiau struktūruotiems giluminiams interviu su ML programuotojais, dirbančiais įvairiose kompanijose ir kuriančiais skirtingų paskirčių ML sprendimus. Apklausta 14 asmenų (iš jų 4 moterys), kurių amžius 25-35 m., jie užima vidutinio lygio, vyriausiojo ML programuotojo ar ML komandų vadovų pareigas įmonėse Lietuvoje bei užsienyje.

Darbo rezultatai rodo, jog ML inžineriją iš tiesų galima laikyti kultūriniu procesu. Tai kultūra, apimanti matematine, inžinerine logika suvokiamų procesų parinkimą. Tačiau pastebėta, jog dažnai jie parenkami remiantis ne vien šia faktine, bet ir pačių programuotojų socialinėmis, kultūrinėmis vertybėmis paremta logika. Ir šios vertybės daro įtaką ML dizainui bei paties ML poveikiui naudotojui. Pastebėtos šios, ištirtas įtampas apimančios tendencijos:

1. Dauguma įtampų sprendimų vienu metu apima ir faktinę (mokslinę, techninę), ir normatyvią (programuotojų vertybėmis paremtą) sprendimų logiką. Tai sukuria problemų, kai tas pats argumentas yra apmąstomas ne inžinerinio bei socialinio poveikio prizmėje kartu, bet tik kaip vienas arba kitas. Įprastai tai sukuria situacijas, kur programuotojas yra įsitikinęs saugoti tam tikrą vertybę (pavyzdžiui, nenaudoti privačios asmenų informacijos), kai apie ją kalba socialinio poveikio kontekste. Tačiau, kai mąstant apie praktinį problemų sprendimą, sprendžiamas klausimas aptarinėjamas pirmiau jį

suvokiant tik kaip inžinerinę problemą, atrodo, nebepastebima, jog šis inžinerinis pasirinkimas daro ir socialinę įtaką prieš tai minėtai vertybei.

2. Pastebėta, kad įvairiuose įtampos taškuose siekiama atliepti tiek verslo tikslų, tiek poveikio naudotojui saugumo motyvus. Verslo poreikis – maksimizuoti pelną minimizuojant darbuotojų išlaidas, darbo jėgos jėgos kaštus. Naudotojų saugumo poreikis – minimizuoti etinių klaidų (privačios informacijos panaudojimo, socialinių grupių diskriminavimo ir panašiai) riziką, maksimizuojant papildomą ML patikrą (kainuojančią išlaidas). Šie poreikiai yra priešingi vienas kitam ir rezultatas priklauso nuo to, ar verslas, ar naudotojų saugumas yra pripažįstami didesne vertybe. Šio darbo kontekste dažniausiai verslo motyvai buvo svarbesni – pirmiausia siekta maksimizuoti pelną, o tik po to – klaidas. Tad klaidos būdavo minimizuojamos ne kiek įmanoma labiau, bet tik tiek, kad potencialiai įmanomas pasiekti maksimalus klaidų kiekis nebūtų toks kritinis, jog sukeltų teisinių/etinių keblumų.
3. Keliuose įtampų taškuose pastebėta priešprieša tarp mokslinės ML teorijos bei praktinio jos pritaikymo. Nors mokslinės teorijos neatmetamos, kai kalbama tik moksliniu žvilgsniu, praktiniame įgyvendinime jos pritaikomos tik tol, kol jos netrukdo siekti pagrindinių programuotojų motyvų, kuris dažnai yra ML kuriamo pelno maksimizavimas.
4. Apie naudotoją programuotojai kalba kaip apie asmenį, kai diskusija vyksta socialinio žvilgsnio kontekste – naudotojas apibūdinamas kompleksiškomis socialinėmis kategorijomis, kaip laimė, pasitenkinimas. Tačiau tą patį sprendimą analizuojant jau inžineriniu žvilgsniu, naudotojas įvardijamas kaip kiekybiškai paskaičiuojamų įpročių atlikimo metrikos, laikomos anksčiau minėtų žmogiškumo bruožų reprezentacijomis. Tai kuria keblumų, nes mąstant apie ML socialinį poveikį, programuotojai toliau vartoja plačias žmogiškumo sąvokas. Tačiau tikrasis ML poveikis naudotojui yra paremtas ne žmogiškų savybių, bet įpročių optimizavimu, tokių kaip paspaudimai, jų skaičius. Pastarasis gali būti pritaikytas daugybei įvairių asmens sampratų ir, kaip rodo pasidalijimai, vieno konkretaus įpročio skatinimas ML programoje gali realų naudotoją paveikti keliose skirtingose, iki tol nenumatytose sferose.
5. Išsiskyrė dvi grupės pašnekovų, priklausomai nuo to, ar rūpinamasi, kad ML rezultatai būtų kuo panašesni į realių naudotojų poreikius, ar ne. Skirtis pastebima priklausomai nuo to, ar ML programų kūrimas yra paremtas pastoviu naudotojų grįžtamuoju ryšiu, informuojančiu inžinierius apie programos rezultatus. Grįžtamojo ryšio gali nebūti, jei

ML kuriamas vienkartinėms prognozėms gauti (pavyzdžiui, duomenų mokslininkų darbe) arba jei tarp programuotojų ir naudotojų egzistuoja verslo ar kitų institucijų tarpininkas (pavyzdžiui, kai ML kuriamas ne darbdaviui, bet įmonei kaip užsakovui), nesuteikiantis grįžtamosios naudotojų informacijos. Atitinkamai, kuo glaudesnis ML kūrėjų ir naudotojų grįžtamojo ryšio santykis, tuo didesnė tikimybė, jog ML prognozės bus paremtos ne prielaidomis apie naudotojus, bet realesniais naudotojų poreikiais ar elgesio įpročiais.

Šios įžvalgos pirmiausia rodo, jog programuotojams sprendimų priėmimo metu dažnai aktyviai nesuvokiant, kad ML yra tiek inžinerinė, tiek sociali technologija, kyla rizikų nepastebėti ML socialinio poveikio galimybių. Mano pasiūlymas ateičiai – tirti, ar aktyvaus tarpdiscipliniško žvilgsnio skatinimas ML inžinerijoje galėtų būti prevencija atvejams, kai programuotojai sukuria etines ML klaidas ir pažeidimus net siekdami jų vengti.

Taip pat skatinčiau tirti santykio tarp programuotojų ir naudotojų išlaikymo per grįžtamojo ryšio procesus naudą. Tyrime minėti metodai, kaip „AB testai“ bei metrikomis grįstas ML projektavimas. Siūlyčiau toliau tirti jų naudą ML socialinio poveikio suvokimui.

Galiausiai, ateities tyrimuose verta daugiau dėmesio skirti įtampoms tarp programuotojų bei verslo poreikių. Teorijos ir klausimyno kūrimo metu šis įtampos momentas nebuvo plačiau įtrauktas, kadangi darbas orientuotas būtent inžinerinių įtampų analizei, tačiau išryškėjo interviu metu. Programuotojų asmeninių motyvų aukojimas dėl verslo tikslų dažnai minėtas net šiame tyrime ir indikuoja būtent šio įtampos ar net galimų konfliktų analizės poreikį.

LITERATŪRA

Akademinė literatūra

1. Berisha, Fjolla, ir Eliot Bytyçi. „Addressing Cold Start in Recommender Systems with Neural Networks: A Literature Survey“. *International Journal of Computers and Applications* 45, nr. 7–8 (2023 m. rugpjūčio 3 d.): 485–96. https://doi.org/10.1080/1206212X.2023.2237766.
2. Bruno, Fernanda, ir Pablo Manolo Rodríguez. „The Dividual: Digital Practices and Biotechnologies“. *Theory, Culture & Society* 39, nr. 3 (2022 m. gegužės): 27–50. https://doi.org/10.1177/02632764211029356.
3. Brusseau, James. „Deleuze’s Postscript on the Societies of Control Updated for Big Data and Predictive Analytics“. *Theoria* 67, nr. 164 (2020 m.).
4. Burrell, Jenna, ir Marion Fourcade. „The Society of Algorithms“. *Annual Review of Sociology* 47, nr. 1 (2021 m. liepos 31 d.): 213–37. https://doi.org/10.1146/annurev-soc-090820-020800.
5. Canedo, Edna Dias, Angelica Toffano Seidel Calazanas, Geovana Ramos Sousa Silva, Pedro Henrique Teixeira Costa, ir Eloisa Toffano Seidel Masson. „Use of Journey Maps and Personas in Software Requirements Elicitation“. *International Journal of Software Engineering and Knowledge Engineering* 33 (2023 m.).
6. Cardon, Dominique. „Dans l’esprit du PageRank“, 2013 m.
7. Castrounis, Alex. *AI for People and Business: A Framework for Better Human Experiences and Business Success*, 2019.
8. Chun, Wendy Hui Kyong, ir Alex H. Barnett. *Discriminating Data: Correlation, Neighborhoods, and the New Politics of Recognition*. Cambridge, Massachusetts: The MIT Press, 2021.
9. Deleuze, GIlles. „Postscript on the Societies of Control“, 1992 m.
10. Drucker, Johanna. „Humanities Approaches to Graphical Display“. *Digital Humanities Quarterly* 5, nr. 1 (2011 m.). https://www.proquest.com/docview/2555208513/abstract/B61E4CD96292457EPQ/1.
11. Foucault, Michel. *Disciplinuoti ir bausti : kalėjimo gimimas*. Vilnius : Baltos lankos, 1998.
12. Guerrero-C, Javier, Nomtika Mjwana, Sebastian Leon-Giraldo, ir Sara L.M. Davis. „Brave Global Spaces: Researching Digital Health and Human Rights through Transnational Participatory Action Research“. *Journal of Responsible Technology* 20 (2024 m. gruodžio): 100097. https://doi.org/10.1016/j.jrt.2024.100097.
13. Klingenberg, Cristina Orsolin, Marco Antônio Viana Borges, ir José Antônio Valle Antunes Jr. „Industry 4.0 as a Data-Driven Paradigm: A Systematic Literature Review on Technologies“. *Journal of Manufacturing Technology Management* 32, nr. 3 (2019 m. birželio 21 d.): 570–92. https://doi.org/10.1108/JMTM-09-2018-0325.
14. Lasi, Heiner, Peter Fettke, Hans-Georg Kemper, Thomas Feld, ir Michael Hoffmann. „Industry 4.0“. *Business & Information Systems Engineering* 6, nr. 4 (2014 m. rugpjūčio): 239–42. https://doi.org/10.1007/s12599-014-0334-4.
15. Lury, Celia, ir Sophie Day. „Algorithmic Personalization as a Mode of Individuation“. *Theory, Culture & Society* 36, nr. 2 (2019 m. kovo): 17–37. https://doi.org/10.1177/0263276418818888.
16. Mejias, Ulises A., ir Nick Couldry. „Datafication“. *Internet Policy Review* 8, nr. 4 (2019 m. lapkričio 29 d.). https://doi.org/10.14763/2019.4.1428.
17. Milano, Silvia, Mariarosaria Taddeo, ir Luciano Floridi. „Recommender Systems and Their Ethical Challenges“. *AI & SOCIETY* 35, nr. 4 (2020 m. gruodžio): 957–67. https://doi.org/10.1007/s00146-020-00950-y.
18. Nascimento, Elizamary De Souza, Iftekhar Ahmed, Edson Oliveira, Marcio Piedade Palheta, Igor Steinmacher, ir Tayana Conte. „Understanding Development Process of Machine Learning

Systems: Challenges and Solutions“. *2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, 1–6. Porto de Galinhas, Recife, Brazil: IEEE, 2019. https://doi.org/10.1109/ESEM.2019.8870157.
19. Nouws, Sem, ir Roel Dobbe. „The Rule of Law for Artificial Intelligence in Public Administration: A System Safety Perspective“. *Digital Governance*, sudarė Kostina Prifti, Esra Demir, Julia Krämer, Klaus Heine, ir Evert Stamhuis, 39:183–208. Information Technology and Law Series. The Hague: T.M.C. Asser Press, 2024. https://doi.org/10.1007/978-94-6265-639-0_9.
20. O’Neil, Cathy. *Weapons of Math Destruction*, s.a.
21. Pal, Siddhi, Ruggero Marino Lazzaroni, ir Paula Mendoza. „AI’s Missing Link: The Gender Gap in the Talent Pool“. Interface. Žiūrėta 2024 m. gruodžio 9 d. https://www.stiftung-nv.de/publications/ai-gender-gap.
22. Paulauskaitė-Tarasevičienė, Agnė, ir Kristina Šutienė. *Intelektikos pagrindai*. Kaunas : Technologija, 2022.
23. Pratama, Irfan, Adhistya Erna Permanasari, Igi Ardiyanto, ir Rini Indrayani. „A review of missing values handling methods on time-series data“. *2016 International Conference on Information Technology Systems and Innovation (ICITSI)*, 1–6, 2016. https://doi.org/10.1109/ICITSI.2016.7858189.
24. Quin, Federico, Danny Weyns, Matthias Galster, ir Camila Costa Silva. „A/B Testing: A Systematic Literature Review“. *Journal of Systems and Software* 211 (2024 m. gegužės): 112011. https://doi.org/10.1016/j.jss.2024.112011.
25. Raiaan, Mohaimenul Azam Khan, Saddam Hossain Mukta, Kaniz Fatema, Nur Mohammad Fahad, ir Sadman Sakib. „A REVIEW ON LARGE LANGUAGE MODELS: ARCHITECTURES, APPLICATIONS, TAXONOMIES, OPEN ISSUES AND CHALLENGES“, s.a.
26. Reigeluth, Tyler. „Recommender Systems as Techniques of the Self?“ *Le Foucaldien* 3, nr. 1 (2017 m. rugsėjo 1 d.): 7. https://doi.org/10.16995/lefou.29.
27. Ryan, Mark, Eleni Christodoulou, Josephina Antoniou, ir Kalypso Iordanou. „An AI Ethics ‘David and Goliath’: Value Conflicts between Large Tech Companies and Their Employees“. *AI & SOCIETY* 39, nr. 2 (2024 m. balandžio): 557–72. https://doi.org/10.1007/s00146-022-01430-1.
28. Roy, Deepjyoti, ir Mala Dutta. „A Systematic Review and Research Perspective on Recommender Systems“. *Journal of Big Data* 9, nr. 1 (2022 m. gruodžio): 59. https://doi.org/10.1186/s40537-022-00592-5.
29. Sadowski, Jathan, ir Evan Selinger. „Creating a Taxonomic Tool for Technocracy and Applying It to Silicon Valley“. *Technology in Society* 38 (2014 m. rugpjūčio): 161–68. https://doi.org/10.1016/j.techsoc.2014.05.001.
30. Sætra, Henrik Skaug. „A Shallow Defence of a Technocracy of Artificial Intelligence: Examining the Political Harms of Algorithmic Governance in the Domain of Government“. *Technology in Society* 62 (2020 m. rugpjūčio): 101283. https://doi.org/10.1016/j.techsoc.2020.101283.
31. Seaver, Nick. „Algorithms as Culture: Some Tactics for the Ethnography of Algorithmic Systems“. *Big Data & Society* 4, nr. 2 (2017 m. gruodžio): 205395171773810. https://doi.org/10.1177/2053951717738104.
32. ———. *Computing Taste: Algorithms and the Makers of Music Recommendation*. University of Chicago Press, 2022. https://doi.org/10.7208/chicago/9780226822969.001.0001.
33. Spaemann, Robert, ir Oliver O’Donovan. *Persons: The Difference between `Someone’ and `Something’*. Oxford University Press, 2017.

34. Wanderer, Emily. „Bearly Recognizable: Facial Recognition and the Wild“. *Science, Technology, & Human Values*, 2024 m. gruodžio 15 d., 01622439241304141. https://doi.org/10.1177/01622439241304141.
35. Wendy Hui Kyong, Chun. *Updating to Remain the Same: Habitual New Media*, s.a.
36. Wright, John. P. *Hume's „A Treatise of Human Nature“: An Introduction*. Cambridge: Cambridge University Press, 2009.

Kita literatūra

1. Europos Komisija. „Dirbtinio intelekto aktas“, 2024 m. 13.
2. UNESCO. „Rekomendacija dėl dirbtinio intelekto etikos“, 2021 m. 23.
3. Pal, Siddhi, Ruggero Marino Lazzaroni, ir Paula Mendoza. „AI's Missing Link: The Gender Gap in the Talent Pool“. Interface. Žiūrėta 2024 m. gruodžio 9 d. https://www.stiftung-nv.de/publications/ai-gender-gap.
4. Chris, Anderson. „The End of Theory. The Data Deluge Makes the Scientific Method Obsolete“, 2008 m.
5. „Pramonė 4.0 - Lietuvos Respublikos ekonomikos ir inovacijų ministerija“. Žiūrėta 2024 m. kovo 20 d. https://eimin.lrv.lt/lt/veiklos-sritys/pramone/pramone-4-0/.
6. Sutton, Rich. „The Bitter Lesson“, 2019 m.
7. „Cambridge Dictionary - Churn“, 2024 m. lapkričio 27 d. https://dictionary.cambridge.org/dictionary/english/churn.
8. „Definition of DATUM“, 2025 m. sausio 3 d. https://www.merriam-webster.com/dictionary/datum.

SUMMARY

Value Tensions in Machine Learning Engineering: a Study of Developers' Attitudes

One of the most impactful contemporary technologies are data science and machine learning (ML) programs. Being based on data about people, these programs have the capacity to impact behaviours of individuals and societies, thus becoming new instances of technocratic power.

Problem: Even though academically acknowledged as an actor in social lives, ML is emphasised more as programs as concrete, atomic sociotechnical actors, shaping our lives but not in themselves being made by social actions. This work acknowledges, that ML development itself is a cultural action, because it consist of various decision making, that is not merely engineering based but also defined by values of engineers themselves. Therefore, a social science analysis of these value tensions is crucial for a better understanding of how various technocratic effects of ML are created.

Object of the thesis: value tension points experienced by machine learning engineers

Aim: to find the value tension points in ML engineering - the decision points at which programmers have to choose between several technological solutions, and what values they base their choices on and how they make them.

Objectives:

1. Formulate the theoretical basis of the study by examining the main ML engineering processes, decision-making alternatives, points of tension.
2. Define the research method for analysing the tensions in ML engineering, the selection of research participants and the guidelines for conducting interviews.
3. Conduct semi-structured in-depth interviews to explore the ML engineering tension points experienced by programmers and the concepts they use to guide their decision-making.
4. Analyse the results of the study and present the ML engineering tension points found and the programmers' value attitudes influencing them.

Work was conducted by semi-structured in-depth interviews with ML developers working in different companies and developing solutions for different purposes. Based on N. Seaver's theory, the research questions were designed to uncover the 'culture' of ML development, i.e. to ask the interviewees about the key processes of ML development and the decision-making issues they face. Design of the questionnaire and analysis were formulated in the light of W. Chun's theory of algorithmic habits and G. Deleuze's concepts of dataisation.

First of all, the results show that ML engineering itself can indeed be considered a cultural process. The tension points discussed show that different programmers behave differently in similar situations and that their decisions depend on their internal beliefs or on the contexts around them.

This gives a new perspective to the sociotechnical analysis of ML. For future research, it is suggested to continue the analysis of these tension points, as the discovered tension points, although they are the main ones, are worth investigating whether there are more and whether the study should be carried out for individual ML types. In analysing the value tension points of ML programmers, it is suggested that the feasibility of implementing ethical ML rules should be analysed in more depth. For example, an analysis of the conflicting values of programmers would reveal cases where programmers, in order to develop what they consider to be an ethical ML program, do not notice that they themselves are making choices that are contrary to their ethical values. Also, points of value tension could become actions worthy of legal ethical investigation, as it is here that decisions are made that influence users. Finally, it could provide a guideline for research on ethical ML engineering management.

In addition, it has been noticed that the closer the feedback relationship between ML developers and users, the more likely it is that ML predictions will not be based on assumptions about users, but on actual user needs or behavioural patterns. These insights add to the discussion on the power relation of data-based representations of persons, analysed in the framework of Deleuze and Chun's theories. Strong user-feedback-based ML design empowers the users themselves to influence the kind of ML they will use and bridges the gap between the real person and their representation. Finally, it encourages future research to focus more on ML development based on "AB tests" and "metrics" as a branch of the discipline with the potential to reduce the power imbalance between users and the ML industry.

PRIEDAI

Priedas Nr. 1. Interviu klausimų gairės

1. Jei galite prisistatykite, pasakant savo amžių, kokioje srityje dirbate šiuo metu, kiek laiko tuo užsiimate. Taip pat, kokioje darbovietėje dirbate, ar tai lietuviška ar užsienio kompanija, kokia įmonės struktūra, dydis bei kaip atrodo jūsų komanda?
2. Papasakokite, kaip ir iš kur gaunate ML naudojamus duomenis.
3. Papasakokite apie apdorojimo veiksmus, kuriuos darote su duomenimis.
4. Papasakokite, kaip parenkate ML modelį.
5. Papasakokite, kaip vertinate ML rezultatus.
6. Kuriuose ML kūrimo procesuose matote rizikas programos kokybės užtikrinimui?
7. Kaip apibūdintumėte gerai ir kaip apibūdintumėte blogai sukurtą ML?
8. Kokiose situacijose galvojate apie programos naudotoją?
9. Kurie ML kūrimo procesai turi daugiausiai įtakos galutiniam programos rezultatai?
10. Kurie asmenys (ML programuotojai, verslo atstovai, naudotojai ar pan.) turi daugiausiai įtakos galutiniam programos rezultatai?
11. Ar koreliacija vis dar nelygu priežastingumui?
12. Kuriuose ML kūrimo procesuose matote etikos rizikų?
13. Kaip pats(i) jaučiatės naudojantis kitais ML produktais?
14. Kas darbe jums kelia stresą?
15. Ką darote, jei artėja „deadline‘as“, bet modelis dar neveikia pakankamai gerai?

Priedas Nr. 2. Interviu transkripcijos

Naudojami ženklai: I – interviuotojas, P – pašnekovas(ė).

Informantas Nr. 1

Amžius: 37 m.

Lytis: vyras

Darbo patirtis: dabar dirba Jungtiniuose Arabų Emyratuose įkurto holdingo vyresniuoju („staff“) duomenų mokslininku. Bendrai 8 m darbo patirties versle ir 5 m parties akademinėje aplinkoje.

Gyvenamoji šalis: Lietuva

Pilietybė: lietuvis

Formatas: nuotolinis skambutis

Trukmė: 1h 16min 13s

Data: 2024 m. spalio 8 d.

I.: Tai prisistatau, kad aš esu Simonas Bansevičius, politikos ir medijų magistro studentas. Darau tyrimą apie mašininio mokymosi programų etiką. Informuoju, jog bet kada galima nuspręsti nutraukti pokalbį. Ar sutinki, kad šitas pokalbis būtų įrašytas, jo medžiaga būtų naudojama tyrimo tikslais?

P.: Hm, Sutinku.

I.: Gerai. Jeigu galima prisistatyti pasakant savo amžių. Ką dirbi šiuo metu? Kiek laiko dirbi šioje srityje?

P.: A, tai labas, aš (anonimizuota). Tuoj bus 38 už savaitėlės. Su mašininio mokymosi ar duomenų mokslo duomenimis dirbu versle nuo 2016 m. berods. O prieš tai dar kelis metus dirbau akademijoj, tai pradėjau viską doktorantūros studijų metu, tai čia būtų buvę 2011 m. ir iš esmės turėjau daug skirtingų pozicijų, bet visose kurių duomenimis grįstus sprendimus ir ir tas buvo labiau iš modeliavimo pusės. Tai ne tik kad analizuoti duomenis ir pateikti kažkokiass ataskaitas, bet kurti sistemas, kurios apsimokėtų iš duomenų ir priimtų pačios sprendimą.

I.: Gerai, ačiū. Taip pat trumpai prašyčiau pristatyti, kur dabar dirbi ir kokia tai ar lietuviška, ar užsienio kompanija. Kokio dydžio.

P.: Tai dabar dirbu (anonimizuota) holdinge. Viena iš didžiausių privataus kapitalo kompanijų Jungtiniuose Arabų Emyratuose. Jie, nežinau, ten tos didelės kompanijos, kaip ten skirstoma, bet manau drąsiai turi tuos 10 tūkstančių plius žmonių. Jeigu imant visas tas įmones, kurios priklauso holdingui ir.

Iš esmės fokusuojasi jie "retail'e", bet turi, tarkim, kino teatrų, verslo, viešbučių, prekybos centrų ir panašiai.

I.: Supratau, Kiek komandoje tavo būtent yra žmonių ir kokia komandos sudėtis?

P.: Tai aš esu duomenų mokslininkų komandoje arba "data scientist" komandoje. Dabar mūsų yra vienuolika žmonių. Kiek žinau, praktiškai visi yra "staff data scientist". Tai ganėtinai jau to aukštesnio "seniority" lygio. Ir mes, kaipo dalis viso "dat'os" komandos, ten kartu yra data inžinieriai ir platform inžinieriai. Tai jeigu juos visus dar prijungus, tai kiekvienoje iš tų komandų ten bent po dešimt žmonių. Tai visas tas tiesa. Visa tos datos komanda gerokai virš 30 turbūt būtų.

I.: Gerai. Tai einu dabar iškart prie tokių techninių klausimų ir taip pat norėčiau, kad pasidalintum procesu. Kaip gaunate duomenis, su kuriais dirbsite?

P.: Ta prasme, kaip konkrečiai mūsų komanda?

I.: Ta prasme ML'ui reikia duomenų, tai iš kur tuos duomenis gaunate? Aišku, čia viskas, kiek jums leidžia teisiniai visi teisinės sutartys.

P.: Tai jau minėjau, kad. Čia yra įmonių grupė arba holdingas. Tai labai priklauso nuo to, su kuria tos įmonės dalimi dirbama. Bet iš esmės kaip ir daugumoje kompanijų, tai tarkim, jeigu "retail'e" prekybos centruose, tai yra tam tikros lojalumo programos, kaip tarkim, Lietuvoje, ten Maxima turi savo ačiū kortelę. Rimi turi savo kortelę, tai ir ten turim savo lojalumo lojalumo programą, kur už tai, kad ja vartotojai naudojasi, jie tipo ir sutinka, kad jų duomenys būtų naudojami personalizacijai ir analizei. Tai iš esmės duomenys ateina iš ten. Aišku, turim visas transakcijas, kas kas perkama parduotuvėse ir pan. Labai daug tų skirtingų, kaip mes vadiname biznis "unitų", tai juose kiekvienuose yra tam tikri tam tikra specifika ir. Šiek tiek skirtingai. Kokie duomenys prieinami ir kaip jie gaunami, bet iš esmės mes naudojame praktiškai vien tik tuos duomenis, kurių esame patys valdytojai. Tai, tarkim, ten iš "third party" kažkokių nieko neimame. Jeigu gauname duomenis tiktai iš to, kas perka. Tarkim, parduotuvėse ar pramogauja kažkokiose vietose, kurios priklauso kompanijai.

I.: Ar tenka pačiam prisiliesti prie tų sistemos dalių, kurios, vat sukuria tuos duomenis? Ten transakcijų sekimas ir vat tos programos lojalumo?

P.: Tai šitu reikalu pas mus data inžinieriai užsiima. Būtent tų duomenų visų patekimu į tą analitikos data "warehouse'ą". Jie tuo užsiima ir turi tam tikras specifines komandas, kurios ten atsakingos, tarkime, už visos infrastruktūros priežiūrą tiems duomenims paimti. Ta komanda, kuri atsakinga už tam tikrų business vienetų duomenų pateikimą į tą analitinį Warehouse tai mūsų komanda dažniausiai tik jau transformuoja tuos duomenis, kurie jau yra surinkti, bet patys mes prie duomenų rinkimo praktiškai niekada net prisiliečiame labai retais atvejais, kada būna kartais, kad kai kuriems projektams reikia

"pascrape'int" duomenų, tai tada kažkiek prisidedame. Bet iš esmės į mūsų. Tas atsakomybes neįeina šita dalis. Bet inžinieriai tą daro.

I.: O sakei, kad modifikuoti reikia. Tai gali papasakoti apie šitą procesą?

P.: Tai čia kaip ir visur, turbūt tie duomenys dažniausiai renkami nebūtinai galvojant apie analitiką. Arba bent jau, kad jeigu galvojama apie analitiką, tai apie tokią, kuri labiau "BI", ar reporting kažkokiam, bet ne kažkokių ML sprendimų įgyvendinimui. Tai tam, kad galėtume kažkokius modelius kurti ar kažkokius duomenimis grįstus sprendimus, tai aišku, tie duomenys, kurie yra surinkti tuo formatu, tuo labiausiai raw ir ne visada tinka. Tiksliau niekada praktiškai netinka, tai reikia sugeneruoti kažkokių naujų požymių, a ne apjungti skirtingus duomenis. Tarkim, kaip pavyzdys. Grupės mastu yra skirtingos lojalumo programos, bet per tam tikrus identifikatorius galima sužinoti, kad, tas asmuo atliko tas transakcijas maisto prekių parduotuvėje ir tas pats asmuo lankėsi kino teatre. Na, tai dėl to, kaip yra apjungiamos lojalumo programos viduje sistemos, gali iš tų skirtingų "data set'ų" pasidaryti kažkokių požymių, kokių tau reikia. Tai vat. Labiau eina kalba apie tas transformacijas tokias, kad susijungtų tuos duomenis, kad tokį pilnesnį kliento vaizdą turėtume. Ir po to jau nuo projekto specifikos priklauso. Ko tau reikia, kokių požymių ir panašiai. Tai gali būti kažkoks paskaičiavimas klientų srautų per dieną ar ten per valandą. Jeigu ten su marketingu kažkas susiję, tai vėl ten tie standartiniai, visi frequency ir panašiai. Dalykai, kuriuos pačiam reikia pasiskaičiuoti. Tai tokio vieno apibendrinančio dalyko, kad va taip ir ne kitaip transformuoju dažniausiai nėra. Tai labai tas yra jų specifinis reikalas.

I.: O vat šitas apjungimas minėto pavyzdžio, kad kino teatras tenais prekes perka, tai kas nusprendžia, kokius duomenis jungti tarpusavyje?

P.: Tai iš esmės nusprendi tu kaip kažkokios sistemos kūrėjas. Esmė yra tau žinoti, kokių tų duomenų yra ir kur jie yra ir kaip juos rasti, O po to kaip juos apjungti, kaip juos kartu panaudoti dažniausiai sprendimo tu kaip kažkokio sprendimo, tais duomenimis grįstais kūrėjas.

I.: Taip, tai grynai tavo, kaip ML programuotojo, vieno atsakomybė čia.

P.: Jo, iš esmės tavo atsakomybė žinoti, kur yra kokie duomenys, ką su jais galima nuveikti ir tada žinoti, ką, kaip, kaip juos apjungti ir kaip juos panaudoti būtent savo projekte.

I.: Jeigu dabar, taip kokį nors realų atvejį pabandytum prisiminti ir pagalvoti, vat kaip vyksta tavo mąstymo procesas? Kaip nusprendi, kuriuos duomenis jungti, kuriuos naudoti, o kurių ne?

P.: Tai iš esmės pas mus dauguma darbų apibrėžia pagal tai, su kokiu to verslo vienetu dirbi to biznio unitų. Dažniausiai, tarkim, jeigu su su marketingu kažkuo dirbi, tai turbūt vienas iš tų keistų, kur reikia apjungti daugiausiai skirtingų business unitų, Nes iš esmės tu nori identifikuoti klientus, kurie galbūt yra labiau aktyvūs vienoje sferoje, bet žinai, kad jie pas tave leidžia pinigus ir tarkim, kaip ir yra

lojalūs klientai, bet tarkim, nesinaudoja Kažkurioje iš kitų business unitų paslaugų, nes gal nežino, gal naudojiesi konkurento ir pan. Tai vat tokiu atveju dažniausiai iš tų skirtingų visų verslo vienetų apsijungti duomenys, O. Jau dirbant su vienu konkrečiu tai irgi. Tarkim, jeigu dirbant su maisto prekių parduotuvėms, tai vis vien pagrindinis tas duomenų masyvas būna transakcijos. Ane. Tai gal tu žiūri kaip kokiu tuo "granularity" tu sprendi dalykus? Ar tu žiūri iš pagal kažkokį vieną klientą ar sprendimą, kuri pagal tai, kaip klientas elgiasi ar ar tau įdomu visų klientų bazė ar kažkoks specifinis "clusteris" klientų. Tai nuo to ir keičiasi iš esmės ar kartu jungia skirtingus "data setus" ar agreguoji juos kažkaip, ar naudoji grynai transakcijų lygmenyje. Na iš esmės apsprendžia, kokią tu verslo problemą sprendi ir ir tas apsprendžia, kokių tuo "granularity" tau duomenų reikia. Ir pagal tai tada jau sprendi, ar agreguot smarkiau, ar ne?

I.: O kas apsprendžia, kokia verslo problema?

P.: Tai verslo problemas dažniausiai verslas apsprendžia. Skirtingų tų būdų yra šitam dalykui. Tarkim, kaip dabar viskas vyksta pas mus, tai verslas ateina jau jau su problema, kurią nori išspręsti, tai mes šitoj dalyje nedalyvaujam visiškai. Mes iš esmės tik verslo problemą padedam. Ne padedame, o tiksliau išverčiame savo pusę į matematinę problemą, kurią išsprendžiame. Bet ta verslo problema ateina apibrėžta jau iš verslo pusės. Kai kuriose prieš tai buvusiose darbovietėse, tai šitoje dalyje lygiai taip pat dalyvauji ir dažniausiai būna gerokai sklandžiau viskas, kai pats irgi dalyvauji, nes verslas galvoja, kad jie turi tokią problemą, bet ne - dažnai nebūna patikrinę elementarių hipotezių apie tuos dalykus ir pasirodo, kad problemos yra visiškai kitoje vietoje. Tai kai jie jau ateina su ta problema, pradedi ją spręsti. Būna, kad sugaišti laiko be reikalo, bandydamas kažką išspręsti. Kur realiai nėra iš esmės tos problemos.

I.: O gal galėtum kokį pavyzdį duoti? Problemų neapibrėžimo problemos?

P.: Gal net ne tiek kiek problemų neapibrėžimo, bet kai ateina ir tarkim galvoja, kad būtinai tą problemą reikia ML spręsti. Vat turim, turim tokią problemą ir čia viską turime, visą data turim ir sprendžiam. Tarkim, vienas iš projektų, prie kurio teko prisidėti, tai buvo kasininkų tvarkaraščio optimizavimas. Parduotuvėje, tarkim, ten ir vienas iš dalykų, kuris buvo, kur paaiškėjo jau ganėtinai vėlai projekte, kad, tarkim. Negalime pasakyti, koks kasininkas, kurioje kasoje kada prisijungęs būna. Pagal tai, kokius duomenis dabar "capturina". Tai tarkim, tie duomenys kažkur sistemoje yra, bet iš esmės jie niekad nepagalvojo, kad čia būtų kažkam toks dalykas naudingas. Na, tai kažkur sisteminis reikalas yra, bet vat tokius visokius dalykus labai greitai galėtum pastebėti ir išsiaiškinti. Jeigu ten prieš pradedant kažkokį projektą kelias dienas workshop pasidarytų su data "owneriais" su tais pačiais kasininkais, tuos pačius kasininkus įtraukiant, sužinotų apie specifikas tam tikras, kur, tarkim, parduotuvės vadovas nebūtinai viską žinos, kas veikia tom smulkioms detalėm. Tai vat toks iš esmės tas toksai projektas gal

ne neapibrėžtumas ar neapmąstymas iš skirtingų perspektyvų. Dažnai būna ta ta problema, kai neįsitraukia tarkim, įmonės, kurie supranta tą verslą, žmonės, kurie turi išspręsti tą problemą, ir žmonės, kurie turi naudotis tuo sprendimu, ir žmonės, kurie iš tikro atlieka tą darbą, kažkokį.

I.: Tai pasitikrinu. Šituo atveju buvo neapmąstyta iš verslo pusės, kokių dar duomenų reikia?

P.: Iš esmės jie žinojo, ką nori pasiekti. Bet, tarkim, irgi nežinojo, kaip tiksliai išmatuota, ar tas pasiekta, ar nepasiekta. Neturėjo "commitmento" kažkokio iš parduotuvių, įsipareigojimo iš parduotuvių vadovų, kad jie naudosis tuo įrankiu. Tai, tarkim, buvo sukurtas tas sprendimas paleistas ir keliose parduotuvėse jis pasitvirtino ir po to buvo ganėtinai sunku, tarkim, visoj šalyje įdiegti per visas parduotuves, nes tos kelios parduotuvės, kurios sutiko dalyvauti toje pradinėje stadijoje - su jomis buvo kalbėta, diskutuota. Visos kitos net nežinojo, kad toks dalykas apskritai vystomas ir daromas. Tai vat, dėl tokių visokių niuansų, nesusikalbėjimų, kažko ne įtraukimo į visą procesą. Dažnai atsiliepia skirtingose fazėse arba sudėtinga būna kažką sukurti, arba sudėtinga būna po to, kad jį "adpotintų" kažkas.

I.: O pats sakei apie tą pačią problemą, kad. Duoda su ML spręsti kažką, ko net nereikia su ML spręst. Yra dar tokių pavyzdžių buvę? Ne iš tos pusės, kad neišėjo įgyvendinti inžinerinės problemos, bet tiesiog tau atrodo, kad nereikėjo ten ML būtent.

P.: Jo tai būna, kad žinai, jie patys susigalvoja kažkokį scenarijų ir pagalvoju, kad jo vat čia reikia būtinai taip daryti. Ir tada paklausi ,,O taip paprasčiau bandėt?" Ai ne. Nu pabandom, pažiūrėsim paprasčiau. Gal ten tiesiog užtenka kelių taisyklių kažkokių ar ne? Ar ten kažkokio kažkokios logikos SQL parašytos, kur realiai padarys tą patį reikalą. Tiktai gerokai paprasčiau viskas ir ir suprantama ir patiems verslo naudotojams, tai karts nuo karto būna tokių "use case'ų". Bet paskutiniu metu jau mažiau truputį, nes pas mus irgi kažkiek turim tokius kaip delivery manager, tai jie kažkiek validuoja tuos "use case'us" , kurie pas mus ateina, tai jau jie turi pakankamą tą techninį bagažą, kad suprastų, ar ar čia reikia kažkokio ML sprendimo, ar čia galima kažką paprastai padaryti ir neverta apsiimti.

I.: O taip trumpai kokį pavyzdį iš šito atvejo, kur nereikėjo ML.

P.: Dabar kažkokio bandyt prisimint...

I.: Nu tada nesvarbu.

P.: Dabar kažkaip grietai nešauna taip į galvą.

I.: Papasakok apie patį procesą, kaip apdoroji tuos duomenis? Va tą atvejį, kai duomenys būna nešvarūs. Tai papasakok apie tą nešvarų duomenų apdorojimą.

P.: Tai tie nešvarūs. Čia toks irgi slidus tas terminas. Iš esmės tie kažkokie keistumai tuose duomenyse gali būt kelių tipų. Tai reik pabandyti išsiaiškinti ar tie netikslumai visokie atsiranda

sistematiškai ar ne. Kas kas gali indikuot apie kažkokias problemas duomenų surinkimo procese ar kažkokios temos kažkokius neatitikimus, ar tie ar kažkokie gedimai įvykę tam tikrais Laiko tarpais. Ar tie duomenys turi kažkokių neatitikimų, kurie neturi kažkokių akivaizdžių trendų ir "pattern'ų"? Tai priklausomai nuo nuo šitų vat pobūdžių priklauso tai, ką tu darysi. Net jeigu nėra kažkokių akivaizdžių sisteminių trendų, kur matytum, kad kažkoks pasikartojantis dalykas yra. Tai dažniausiai, jeigu nėra didelis duomenų kiekis, kuris probleminis, tai jį tiesiog meti lauk ir tiek. Nieko ten per daug netvarkai, nes visvien duomenų srautai ganėtinai nemaži yra, bent jau šitoje kompanijoje ten tų klientų būna sočiai tiek kino teatruose, tiek prekybos centruose. Transakcijų per dieną pakankamai, kad nedidelis kiekis duomenų ten iš esmės nepakistų. Dabar sudėtingesnis atvejis yra jeigu pamatai, kad yra sisteminis ar dar kažkoks reikalas, kur pasikartojantys trendai. Kur ten nežinau. Kažkokiai specifinei vartotojų grupei, ten koks nors požymis niekad nebūna užpildytas. Tai bandai tada suprast ar ar tas neužpildytas laukas turi būti užpildytas kažkuo, ar kažkokia specifine reikšme, ar kažkur apskritai kažkur "pipeline'as", kur kur šitą data visą "capturine'a" kažkur sulūžęs ar nežinau ar dar kažkokios problemos. Tai vat šitus sisteminius reikalus tai reikia vienokiu ar kitokiu būdu išsispręsti, nes jei iš esmės išmetant juos, kad ir, tarkim, ten null values, kažkokia butų ar kažkokios labai keistos reikšmės, jie galbūt indikuoja tam tikrus dalykus, kur realiai jie atrodo kaip tokie negeri ar netvarkingi duomenys, bet iš esmės jie tau irgi gali būti kaip vienas iš požymių, kad kuris padeda atskirti tam tikrus klientus. Tai vat, sakau realiai dvi tas klases reik atskirti. Ir nuo to priklauso kaip tvarkysies su jai, ar ar tas daiktas toks "missing at random", ar yra tam tikri specifiniai dalykai, dėl kurių ten tų duomenų gali nebūti? Ar jie yra netvarkingi?

I.: Tai kodėl būna labiausiai gaila, kad trūksta duomenų?

P.: Tai ne tai, kad gaila, bet iš esmės reiktų suprasti, ar jie nėra, kad jų trūktų sistemingai, nes jeigu jų trūksta sistemingai, tai tu gali sukurti kažkokį sprendimą, kuris vis tarkim, juos atmetus, sukurs sprendimą, kuris visiškai ignoruoja kažkokią dalį ar ar klientų, ar kažko, kurie turi kažkokį požymį, kuris yra sistemingai trūkstamas ar ne. Ir kai paleisi tai gyvai, vis vien jie niekur nedings. Ar ne? Nebus taip, kad jeigu jie buvo "capturinami" visą laiką, kad ta ta pati dalis duomenų lygiai taip pat gyvai ir production eis Kaip, kaip missing, ar ne kažkokia, ar ten null values kažkur panašiai, tai iš esmės tam segmentui nieko negalėsi gero sukurt, nes niekad su jais nieko nedarai. Tarkim training toje fazėje. O jeigu toks "at random missing", tai tiesiog bus nedidelė dalis dažniausiai ir bus, kad kartais negali priimti sprendimo ir tiek. Dėl to kažkokios didelės bėdos ar trendo nebus. Tai kontroliuok, ką gali kontroliuoti, ko negali - tiesiog yra kaip yra.

I.: O būna, kad tiesiog kai trūksta duomenų, užpildot kitomis reikšmėmis.

P.: Šiaip dažniausiai ne, nes pildant kitomis reikšmėmis tą gali visą tą pasiskirstymą sugadinti. Nes, tarkim yra tas įprotis, kad ir ten tutorial visokiose, kad užpildyti vidurkiu ar mediana ar pan. kas dažniausiai nekeičia ten tos imties vidurkio, tos pačios medianos nepakeičia, bet patį pasiskirstymą pakeičia, ypač jei ten didesnis kiekis duomenų. Tai. Yra net ir tokių tų užpildymo algoritmų, kurie irgi ML paremti. Kur realiai tuos „missing values" jie prognozuoja? Koks labiausiai "likely" tas missing value turėtų būti? Tai. Tas tas užpildymas, dažniausiai tas toks visiškai "naive", daugiau žalos padaro tik negu naudos. Ir iš esmės sakau, kartais, tarkim, tas užpildymas, tarkim, jeigu skaitinė reikšmė ten kažkokia nesąmoninga, tai geriau visus juos įrašyti kaip kažkokią vertę, kuri niekada nepasirodo "data sete". Tai tarkim, jeigu amžius kažkoks ane, tai įrašyti kokį tokiems vat keistai atrodančiam duomenims, kur tarkim amžius sako, kad tau 120 turėtų būt metų, tai tiesiog įrašyti kokį minus 99 ar kažkokia tokią reikšmę, kuri labai aiškiai išsiskiria nuo tų normalių reikšmių. Ir tarkim, jeigu kažkoks kategorijos kintamasis ar "stringas", tai tiesiog vietoje to, kad jis būtų null grynai turėti kategoriją not available ar missing ar ten nežinau damage daiktą ar vat kad aiškiai atskirti, kad čia yra kažkokia klaida, o ne ne įdėt kažkokią vidutinę ar labiausiai tikėtiną reikšmę.

I.: Tai kai minus 99 įrašai, tai nenaudoji tos reikšmės?

P.: Iš esmės tau tau ją labai lengva atskirti.

I.: Čia tiesiog kaip tokią kategoriją pastebėti. Ir dar labai trumpai tiesiog paminėjai tuos ML modelius, i eškančius missing "values". Kaip juos tu vertinai?

P.: Tai šitie. Dažniausiai ten irgi būna koks ar tai neuroninis tinklas, ar ar koks nors klasterizacijos algoritmas, kur bando kažkokius panašius ar ne pagal kitus požymius bando surasti panašias eilutes iš esmės ir pagal tų panašių eilučių nežinau ten ar vidurkį, ar dar kažką. Tada užpildytas reikšmes iš esmės apmoka į modelį. Tarkim man ant dešimties požymių ir tavo target yra 11 tas požymis, kurį prie tos normalios eilutės visos turi užpildyti, ar ne? Ir tada tai eilutei, kur tos reikšmės trūksta, tu "predictini" su ML modeliu pagal tai, kokie tie prieš tai 10 buvusių modelių, tai. Tai toks labiau advanced reikalas, kurį galima naudoti ir.

I.: Ką tu apie jį manai?

P.: Nu čia geriau negu tas "approach'as", kad vidurkio ar mediana. O bet vėl čia reikia tada turėti pakankamai duomenų, kad galėtum tu "impute'int".

I.: Pasidalink tiesiog kaip, kaip sugalvoji, kokio tipo modelį naudot?

P.: Tai iš esmės. Apsprendžia tai, kokia problema sprendžiama. Tai pirmas dalykas suprasti, ar tau reikia naudot kažką iš klasifikatorių, klasifikacijos modelio ar regresijos paprastos ar tarkim laiko eilučių. Tai čia tas visas labiau galioja, kai dirbi su kažkokia structured data, kai vėl, jeigu ten dirbi su vaizdais,

su garsais. Vėl yra skirtingi tipai modelių, ką naudotum skirtingos tos klasės modelių ar ar šeimos. Nežinau kaip pavadinti ką naudotum. Tai. Iš esmės turi pradžioje nuspręsti, ką tu sprendi, ar klasifikacijos, ar regresijos uždavinį, ar ar laiko eilučių uždavinį ir tada jau. Realiai labai daug skirtumo nėra kokį, kokį ten tą modelį pasirinksi. Paskutiniu metu ten tarkim klasifikacijai kokiai tai tie standartiniai pasirinkimai xgboost ar koks random forest. Ar "gradient boosted" trys. I`š esmės kažkurie iš tų medžių modelio. Skirtingų variacijų. Ir regresijai bandai pačią paprasčiausią tiesinę regresiją iš pradžių, jeigu ten galima ir jeigu nepažeidžiamos tos tiesinės regresijos visos prielaidos, tada vėl, jeigu netinka. Tinka ir jeigu tai nėra laiko eilutės, o tiesiog regresija. Tai tų pačių medžių skirtingais metodais tas pats xgboost yra regresijai ar random forest regresija. O jei laiko eilutėmis, tai tada pradedi vėl nuo tų klasikinių visų ar auto regresija, moving average po to sudėtingėja, apjungia šiuos Arima, Arma ir Max taip toliau. Iš esmės viskas dabar pavilkta būna į kokią nors biblioteką. Tai. Tarkim, yra tų sprendimų, tokių ypač laiko eilutėmis, tai tų tokių labiau kaip auto ML, tai ten feisbuko profit ta biblioteka yra Uber Orbit berods kur realiai parenka už tave ir parametrus optimalius. Tau tik reikia iš esmės pasirinkti. Kas tas tavo kintamasis ir kokie ten išoriniai kintamieji yra? Jei turi laiko eilutėmis.

I.: Čia angliškai laiko eilutė būtų kas?

P.: Time series.

I.: Aha. Kaip sprendžia cold start, šaltą startą?

P.: Šiaip paskutiniu metu aš kiek projektų kur dariau, tai nebuvo, cold start problemos. Nes čia labiau tarkim rekomendacinio pobūdžio sprendimas, kai kai kuri ane. Tai vėl ten tų strategijų skirtingų yra, bet tos tos bazinės. Iš esmės susirandi kažką panašaus į. Kažkokį kiekį duomenų jau turi. Bet bet kuriuo atveju kitaip net nepradėsi. Kažkokio ML sprendimo kurt. Tai tarkim, nekalbama apie tą atvejį, kai visiškai jokių duomenų neturi, nes tada reikia pradėti visą datos "pipeline'ą" pradžiai dar prieš pradedant kažkokius modelius. Tarkim, turint kažkokį kiekį duomenų, kai turi kažkiek duomenų bent jau apie tam tikrą aibę user, tai realiai, jeigu kažkokiam naujam useriui reikia pritaikyti tas rekomendacijas, tai gal tada yra keli variantai vienas defaultinis, kai pats paprasčiausias, kai tu tiesiog visiems siūlai kažką to pačio, tam naujam useriui kažkokį tarkim savo populiariausius kažkokius. Jeigu, tarkim, kalbant apie maisto prekių parduotuvę, tai iš tam tikrų kategorijų tiesiog siūlai geriausiai perkamus produktus. Ir gali prioritizuoti ten siūlyti savo brendų produktų kai naujam useriui, apie kurį nelabai ką žinai, kai jau kažkiek tų transakcijų padaro tada gali pradėti lygint tą userį su turimais savo useriais, kurių pirkimų istoriją turi ir siūlyt kažką, ką tie useriai pirkdavo. Tai nebūtina žiūrėti šito naujo userio, ką dažniausiai perka, bet iš esmės pagal tai, ką jis pirko per paskutines kelias transakcijas, sulyginti su tuo, ką tu turi jau savo userių bazėje ir pasiūlyti, ką tie useriai dažniausiai perka iš tam tikrų kategorijų. Tai irgi toks. Vienas

iš variantų, kaip išspręsti šią problemą. Sakau, priklauso turbūt nuo to, kurioje vietoje tu tą cold startą turi. Kokį sprendimą, kurį kuria.

I.: Bet tavo darbe su predictionais taip nebūna?

P.: Ne, praktiškai ne, nepagalvojau. Neturėjau aš. Bent jau šitoje darbovietėje, kur dabar dirbu, tai nebuvo tikrai turbūt kažkokios cold start problemos. Čia labiau rekomendacinio algoritmo. Tas dalykas.

I.: Tai vis tiek, jeigu vos tik pradedu sistemą kurti, kad ir ne rekomendacija, bet tada vis tiek duomenų iš kažkur reikia.

P.: Tai. Labai retai būna taip, kad tu pradėsi projektą kažkokį. Kuriam neturi visiškai duomenų, kur net nepradėjo rinkt duomenų. Tai sakau šitoje vietoje, tai tada tu realiai nelabai gali pradėti tą projektą. Tai. Iš esmės tada viskas sukasi labiau link to, kad tu pradedi tuos duomenis rinkti. Bet čia. Su ML turi bendro tik tiek, kad tu gali kažkiek pasakyti kokius duomenis capturint ar ne, bet iš esmės ten kažkokį sprendimą. Jeigu nori padeployint, tai ten bus labai kažkas basic ir dažniausiai taisyklėmis paremta, bet labiau tik tam, kad tą datą susirinkti.

I.: Na gerai.

P.: Bet mes kiek darom tai sakau, vis tiek pagrindas pas mus iš transakcijų viskas eina, Tai tiesiog tas naujas klientas, kuris atėjęs, tai pagal tam tikrus požymius turi arba ką jisai kituose business unituose veikia arba naujam klientui turi tiesiog kažkokį default elgesį, kuris dažnai su ML nelabai ką bendro turi.

I.: Gerai. Ir dabar apie tai, kaip vertini sukurtą ML?

P.: Tai šitoj vietoj šiek tiek du gal atskiri dalykai yra. Yra Vienas iš grynai tos techninės pusės ML modelių ir tada kitas dalykas yra iš verslo pusės, verslo KPI kažkoks. Tai tuos abu dalykus iš esmės reikia stebėti. Ir jeigu tarkim iš tos techninės pusės gali. Plius minus pasitikrinti dar prieš paleisdamas kažkokį sprendimą. Kaip ten kas veiks ir bent jau ant tų nematytų duomenų gali pasitikrinti ir pažiūrėti ar ten tam tikros metrikos, ar ne tai. Tarkim klasifikavimą į tą precision recall kreives pasitikrinti regresijai tą vidutinę standartinę paklaidą kažkokią žiūrėti ir pan. Tų matematinių metrikų yra sočiai. Ką ten gali pasitikrinti. Didesnė bėda būna, kaip pažiūrėti kokį, kokią įtaką tiems verslo KPI turės tai tam reikalui ypač kompanijos, kurios kuria produktą, tai tas naudojamas AB eksperimentavimas, AB testing testing, kur iš esmės. Taip, grubiai tariant, yra dvi grupės žmonių vienai grupei. Rodai kažkokį seną variantą arba esamą variantą, o kitai grupei rodai kažkokį naują variantą ir tada tą eksperimentą Savaitę ar kelias savaites esi praleidęs gyvai ir ir seki tam tikras metrikas, verslo metrikas ir po to statistiniais metodais gali patikrinti, ar ar statistiškai reikšmingai skiriasi tas impactas tavo naujo feature'o ar ne tai. Tai vat šita dalis. Iš verslo pusės tas įvertinimas svarbesnis ir gerokai sudėtingesnis. Dažniausiai būna. Ir rizikos šiokios tokios būna. Kur negali dažniausiai negali to feature paleist, tarkim, visai aibei savo klientų, nes

nežinai, koks impactas gali būti. Tai iš pradžių paleidi, tarkim, kažką naujo 5 proc. srauto visų klientų, po to 10 proc. ir laipsniškai didini tą srautą tol, kol pamatai, kaip reaguoja. Kad jei matai, kad pradeda verslo rodikliai važiuoti žemyn, tai galėtum tiesiog išjungt tą dalyką ir negadinti visko. Tai vat iš tos verslo pusės per tą eksperimentavimo, AB testing tikrinji. O iš techninės pusės sakau ten iš esmės tas matematines metrikas tikrini ir žiūri ar ar atitinka lūkesčius, ar ne. Dar prieš paleidžiant ar po to paleidus stebi, kaip jos ar keičiasi laike, ar ne.

I.: O kas nustato AB testingo metrikas?

P.: Tai tuos dažniausiai kartu su verslu nusistato. Nes nu iš esmės vis vien jie geriau žino. Savo klientus jie geriau žino. Kas jiems svarbu, a ne, tarkim, žino hipotezę, kažką, kas galvoja, kad turėtų pasikeisti, jeigu įgyvendintų vienokį ar kitokį sprendimą, ar ne. Tai verslas gal labiau apibrėžia kas tai, kas ta metrika ir kaip, ir iš esmės, kaip ją vertinti. Bet tą techninį sprendimą, kaip surinkti duomenis, kaip tai statistiškai įvertinti tuos pokyčius? Ar jie reikšmingi, ar ne, tai jau. Duomenų mokslininkų komanda labiau apibrėžia.

I.: O yra buvę, kad nesutarėt, reikėjo daug diskutuoti, kol nusprendėte metriką.

P.: Tai sakau čia tas sutarei nesutirei, tai mūsų komanda labiau tuo techniniu įgyvendinimu užsiima. Tos pačios metrikos parinkimas iš verslo pusės tai. Tokia labiau nuleidžiama iš viršaus dažnai būna nebent labai kažkokia nesąmonė pasiūlo. Kur tu tarkim, gali pamatyti iš anksčiau buvusių duomenų, kad tarkim, ta metrika ten niekada nesikeičia, nesvarbu, ką darai ir ji nesusijusi visiškai su tuo, ką tu bandai daryti. Gali tada žinai su duomenimis parodyta argumentuoti, kad turbūt nebūtų verta.

I.: Dabar jau toks bendresnis klausimas. Kuriuose ML kūrimo procesuose įžvelgtum potencialias rizikas algoritmo kokybės užtikrinimui?

P.: Turbūt didžiausia bėda duomenų surinkime būna. Gal sakau ten ar per klaidą, ar dėl kažkokio sistemos funkcionalumo. Gali būt, kad. Tas kažkokie duomenys corrupted renkami. Kažko neužfiksuoja ar iš esmės tu nefiksuoji kažkokių duomenų, kurie galbūt tau padėtų išspręsti ar pamatyt kažkokią problemą? Tai turbūt didžiausia problema yra dizaine. Tai kokių tau duomenų reikia ir kaip juos surinkti, ir iš kur paimti? Po to jau, tarkime, ML techninis įgyvendinimas. Tai. Dabar, manau, jau šiais laikais ganėtinai išspręsta problema, nes programuoti kažkokį savo algoritmą dažniausiai nereikia. Yra pilna bibliotekų, tai ten klaidos tikimybė ganėtinai sumažėja, nebent visiškai nežinai, kaip ten parinktas algoritmas. Bet čia vėl kita klausimo dalis, tai turbūt pasirenkant kokius duomenis naudoti ir kokius duomenis surinkinėti ir po to turbūt pasirenkant kaip įvertinti ar yra efektas kažkoks to sprendimo ar ne. Nes kaip kai kurie projektai būna, kad labai sunku pamatuoti, tarkim tiesioginę įtaką kažkokią ten tiems KPI. Visur, kur žmogiško įsikišimo kažkokio reikia, ten dažniausiai ir klaidos tikimybė atsiranda didesnė.

I.: O kur matytum didžiausią riziką etikos atžvilgiu. Irgi visuose procesuose.

P.: Aš bent jau kaip galvoju, tai kad didžiausia rizika yra, kai mes tiesiog aklai bandome atkartoti tą procesą, kuris jau dabar vyksta ir, tarkime, tiesiog automatizuoti kažkokį sprendimą, kuris jau dabar daromas. Tai. Tarkim, yra tie duomenų rinkiniai, kurie iš esmės yra kažkokie biased. Vien tai, kaip visas procesas sustatytas. Nesvarbu, kad tu, tarkim, parenki modelius adekvačiai, bandai reprezentuoti ten, neįveli tarkim duomenyse kažkokių papildomų reikalų, ar ne, bet patys duomenis iš savęs jau jau turi savyje kažkokį biasą. Tai vat šitoje vietoje turbūt yra didžiausia rizika, nes kaip ML kūrėjas tai nežinau, ar tu labai kažkokiu įdedi biased dalykų. Nebent visiškai vienas projektą darai ir tada gali truputį papulti į tą tokį netinkamą mąstymą, kur galvoji, (kas dažnai atsitinka, jeigu taip galvoji, kur galvoji, „jeigu aš taip darau, tai visi taip daro"). Tai čia turbūt labiausiai atvėrė akis, kai su produ projektą turėjome banke ir iš esmės bandėme apibrėžti, kas yra normalus sąskaitos naudojimas, tarkime, banko sąskaita. Ir mes tada komandoje buvome dešimt žmonių. Ne visi dirbo prie to pačio projekto, bet tiesiog galbūt 10 data scientistų ir visi turėjo savo įsivaizdavimą ir savo specifinį būdą, kaip naudoti sąskaitas. Tarkim, ten vienas tą sąskaitą naudoja atlyginimui gauti ir pervesti iškart kažkur kitur visa suma ir iš kitų sąskaitų dirbti. Kitas sako ne, aš ten gaunu tuos pinigus ir ten dešimtą dieną visą laiką padarau pavedimus už, tarkim, komunalinius ir daugiau niekam nenaudoja. Kitas sako aš viskam naudoju vien tik šitą sąskaitą ir iš esmės. Kiek žmonių, tiek skirtingų būdų, kaip tai panaudoti. Realiai visi yra kaip ir legit scenarijai, kaip kad to normalaus panaudojimo. Tai vat, jeigu dirbi vienas ir nepasitikrini su kažkuo, tai šitoj vietoj gal gali kažkiek tą biasą įvelti, Kai galvoji, kad jeigu aš taip galvoju, tai dauguma taip galvos. Bet dažniausiai, jeigu tu taip galvoji, tai dar kokie 5 procentai taip pat galvos. O likę 95 dar pasidalins į 20 skirtingų grupių. Tai va, turbūt šitoje vietoje dar yra ganėtinai didelė rizika kažką padaryti tokio, kas nelabai etiška tą tavo sprendimą padaryt.

I.: Ar darbo metu būna, kad pagalvoji apie tą žmogų, kuris naudos galutinį produktą.

P.: Tai jei nepagalvotum, tai tada geriau nedaryt tokių produktų. Čia turbūt vienas iš pirmų dalykų, kuriuos pagalvoti turi, nes iš esmės turi arba pats kažkaip įsijausti į tą vaidmenį. Kas tas galutinis vartotojas? Arba tarkim, kaip kai kurios kompanijos kur kad dirbau, kad ką mes darydavome, tai tokius Discovery Discovery Workshop, tai ten tu dalyvaudavai kaip tokio sprendimo kūrėjas ir iš esmės fasilituodavai viską ir tada surinkdavai žmones, kurie tarkim, dirba, tam biznis unite, žino savo klientus geriau. Žmonės iš vadovavimo grandies, visiškai iš tų operations grandies, kur arčiausiai pačio kliento yra, tai jie irgi turi skirtingus tuos požiūrius ir bandai kažkaip susidėlioti ar identifikuoti tas personas skirtingas. Tarkim tie visi POC, kur prasideda. Tai va sakai, mano persona yra toks ir toks apsibrėži, kokis tris keturias skirtingas personas ir. Ir iš kiekvienos tos personos perspektyvos žiūri, kad tas

sprendimas atitiktų jų lūkesčius. Tai tas. Jeigu negalvoja apie galutinį vartotoją, tai labai dažnu atveju prašauna su tuo, ką sukuri.

I.: O lūkesčius Kieno? Kai sakai lūkesčius, apgalvoji.

P.: Jo tai lūkesčius galutinio vartotojo.

I.: Kai pats naudoji programą, kuri žinai, kad naudoja kažką machine Learning. Kaip jautiesi?

P.: Aš šitoje vietoje turbūt per daug lankstus ir nežinau man tai dažniausiai labai mažai skauda širdį, kad ten kažkas netinkamai ar tinkamai mano duomenis panaudoja. Bet tas dažniausiai priklauso nuo to, kiek naudos aš iš to gaunu. Tai. Aš tikrai būčiau vienas iš tų, turbūt, kuris atiduočiau vos ne visus duomenis, ir ganėtinai private, jeigu iš to gaučiau kažkokį gerą servisą. Jeigu tikrai pagal mane customise'intų dalykus ir padarytų taip, kad man būtų patogu, tai. Nežinau, bet aš sakau aš šitoje vietoje turbūt dėl to, kad dirbu toje srityje dėl to, kad yra tokia dalis tyrėjo gyslelės likus. Kur man tas progresas iš tos dalies rūpi šiek tiek daugiau, negu, tarkim, ten privatumas kažkoks ar etika. Bet čia turbūt toks labiau išskirtinis atvejis negu taisyklė.

I.: Jeigu tavo vaiko duomenys visi būtų?

P.: Ne, tai matai aš. Galbūt į tai žiūriu grynai per savo prizmę ir per savo sutikimą. Aš tokį padarau savo conscience decision'ą ane. Kad savo aš duodu ir panašiai, bet aš manau, kiekvienas turi tą pasirinkimą turėti. Na tai kaip aišku aš kaip tarkim kaip tėvas ten iki aštuoniolikos atsakau už vaikus. Bet vėl. Vis vien nori nenori, tai yra kitas žmogus, kaip kad ir nesvarbu, kiek ten ar kraujo ryšys ar kas kitas, bet kitas žmogus turi savo poreikius, savo, savo lūkesčius ir panašiai. Tai tam tikrų dalykų, ypač kurie tokie šiek tiek slidūs. Tu, geriau ne nepriiminėti tų sprendimų už juos. Tai, ko aš, aš iš savo asmeninės prizmės sakau. Manau, labai daug ką leisčiau, bet. Puikiai suprantu ir tuos, kurie visiškai nenori niekuo dalintis ir ir visiškai prieš tokius dalykus.

I.: Kas manai, padaro didžiausią įtaką galutiniam modelio rezultatui, t.y. kokie žmonės? Na ta prasme imant grynai visą pilną kontekstą nuo pat asmens, kuris naudojasi sistema iki sistemos projektavimo, duomenų rinkimo, verslo. Žodžiu, visi veikėjai įmanomi nuo programuotojų, verslo iki pat naudotojo.

P.: Turbūt truputį priklauso ir nuo nuo tos srities, kas ten daroma, bet iš esmės galutinis vartotojas turbūt turi vos ne daugiausiai svertų. Na, bet čia dar priklauso nuo to, kas kas per produktas. Nes ypač tose ankstyvose stadijose, tai jeigu tu sukursi sukursi kažką, kas nėra įdomu galutiniam vartotojui, tai tas nebus vartojama ir nebus naudojama nieko. Tai. Tu gali parinkti, tarkim, tam tikrus metodus, turi parinkti technologijas ir taip toliau. Bet tas pats poreikis visvien anksčiau ar vėliau ateina iš vartotojo. Ir. Turbūt vat ir tos visos. Visi startupų pavyzdžiai irgi rodo, kad tie, kurie pataiko į kažkokią nišą, kur kuri

netatverta yra ir kur kur vartotojui tikrai naudinga, tie būna sėkmingi ir. Grynai yra labai didelė dalis sėkmės, o ne tai, kokius ten protingus žmones susirinkęs esi, kokias technologijas naudoji ir panašiai. Tai. Turbūt ta pati pradinė idėja. Kiek ko kita markets šita turi ir toliau. Aš tai sakyčiau, kad nuo galutinio vartotojo super daug priklauso, kur link tavo produktas vystysis, nes jeigu neatsiliepsi jo poreikių, tai ten gali kiek nori būti protingas ir kažką įmantraus gali daryt, bet jeigu to niekam nereikia, tai tas ir liks nenaudojama.

I.: Kas darbe kelia daugiausiai streso?

P.: Streso turbūt Kaip daugumai tarkim kas kuria kažkokius sprendimus, tai tas kontekstų switchinimas, jeigu prie tarkim kelių skirtingų projektų dirbi ir ir ten per tą pačią dieną reikia dvi valandas prie to praleist, dvi valandas prie kito, ten emailai, meet'ai ir taip toliau, tai tas pastoviai galvoj perjungiamas į kažkokį kitą kontekstą. Turbūt didžiausia dalis to streso. Ir turbūt čia didžiausias nesutarimas tarp management'o ir tų, tų kurie labiau kaip management darbą dirba, ir tų, kurie labiau kaip kūrėjai. Kuriems reikia to nepertraukiamo darbo.

I.: Ką darai, jeigu atėjo deadline'as, o tavo modelis dar neveikia, kaip reikėtų?

P.: Tai nežinau. Šiaip, dažniausiai nebūna tokių hard hard deadline'ų. Kažkokia vis vien versiją turi plius minus kažką darančią, tai blogiausiu atveju visą laiką gali paleist kažką, kas nevisai veikia ar tarkim veikia gerai kažkokiam mažam segmentui tiktai, tai paleisk tik tam segmentui, kad kuo greičiau nueit į tą live prodą, kad pasitikrint kaip veikia ir gauti kuo daugiau feedback, kad galėtum tobulinti. Iš esmės tai sakau, turbūt priklauso irgi nuo nuo kompanijos, bet dauguma kompanijų, kurios labiau tokios tech orientuotos ir inovatyvios. Jos nori kuo greičiau kažką paleisti, nebūtinai tobulo, nebūtinai labai gerai veikiančio, ar gerai veikiančio tik labai mažam segmentui. Tai dirbi ties tuo, kad kažkokią pradinę versiją turėtum kažkuriuo metu ir dažniausiai būna susitarimas, kad nelauksim, kol ten išdirbsi iki kažkokio tikslumo ar panašiai. Bet va praėjo toks laiko tarpas. Reikia paleisti pirmą versiją ir paleidi tą pirmą versiją, žinodamas - jeigu ypač, jeigu tu gali tiksliai pasakyti, kokie ribojimai dabar yra ir aiškiai gali tą iškomunikuoti ir ta rizika nėra ten super kažkokia didelė jėga. Aišku, jeigu rizika per didelė, kad ten kažkokį didelį efektą verslui turėtų neigiamą, tai tiesiog tas deadline'as pasislenka tol, kol turi kažką, kas yra plius minus veikiančio.

I.: Rizikos pavyzdžių - kas būtų rizika?

P.: Rizika, kai kažką paleidi ir nežinai, tarkim visiškai kaip veiks. Ir tarkim nežinau, iš tie, kurių rekomendacijos algoritmu, ar sprendimų kažkokių tai. Tarkim. Rizika, kad tavo rekomendacijos padarys taip, kad klientai churn'ins kažkur vidury proceso, tai tarkim pamatys tavo rekomendacijas ir pradės

išeidinėt, nes ten siūlys visiškai kažkokius nesusijusius dalykus. Tai iš esmės rizika prarasti klientą, prarasti pinigus.

I.: Kai vertini ir algoritmas jau artėja prie tikslumo. Tai yra šitas overfitting'as, kaip nusprendžiat, kada jau tos ribos, kur reikalingas tikslumas, bet kur overfitting'o visas tikslumas?

P.: Nežinau mes šiaip neturim labai daug projektų, kuriuos nuolatos tobulintum, tai mes plius minus kaip darom, kad pasiekus tam tikrą tikslumą, kuris kaip ir yra okay. Paliekam tą projektą run'int tiesiog ir labiau kažkokius bug'us gaudome ar tiesiog stebime, kad nekristų tas tikslumas žemiau. Bet bent jau pas mus nėra tų projektų tokių, kad tarkim tą tikslumą užkeltum iki 90 proc., tarkim, kaip pavyzdys, kad būtinai reikia iki 91 dienų, o tada iki 92, tada iki 93. Nes dažniausiai tas 90 yra good enough jau ir tai, kad ten iki 93 pakelsi praleisdamas ten prie jo dar papildomai kokiu šešis mėnesius, neduos papildomai tiek tos naudos. Tai nežinau pas mus... Kadangi mes darome daug, labai daug skirtingų projektų su daug business unit'ų, nėra, kad kažkokį vieną specifinį produktą, kurį turime ir nuolat jį tobuliname, tai pas mus tokie būna dauguma sprendimų good enough, o ne ne ten amazing ir panašiai.

I.: O kaip nustato, kada jau good enough?

P.: Čia geras klausimas. Iš esmės tai. Tu. Žiūri ar tarkim. Nu Priklauso nuo projekto. Jeigu ten kažkoks automatizavimo tik tai projektas, tai plius minus žino kiek. Kiek ten to proceso. Sėkmingai automatizuota. Tarkim yra buvo projektas. Receipt scannning'o, ten gali nuskanuoti savo čekį ir tau pagal tai ten bonus points kažkokių priskiria. Tai nufotkinti čekį. Identifikuojame, kokia suma kokioje parduotuvėje išleista ir taip toliau. Ir ir paprocess'ini. Tai, tarkim šitam dalykui pradinis reikalas buvo, kad 70% čekių galėtų visiškai be žmogaus įsikišimo pravaliduot, kad ten tiesiog app'se nufotkinti ir, ir iš esmės per kelias sekundes tas taškų balansas atsinaujina, tai vat tiesiog verslas ateina ir sako, kad vat mūsų tikslas, kad dabar yra komanda, kur ten 20 žmonių, kurie approve'ina tuos čekius, peržiūri ar ne. Ir mes tarkim norim palikti tik 2 žmones, kurie approve'intų, tai tu pagal tai gali pasiskaičiuoti plius minus kiek koks tas success rate'as turi būti, kad iš viso to srauto, kiek dabar tų čekių yra, kad ir įvertinus, tarkim, paleidus tą sprendimą, kiek išaugtų tas tas poreikis tokio dalyko, kiek daugiau žmonių tų čekių pradėtų skanduoti. Kiek koks tau success rate'as turi būti, kad liktų tik du žmonės, kurie, tarkim, pavaliduotų tuos sudėtingus atvejus. Tai skirtinguose projektuose tas irgi tas toks pasakymas kas yra good enough skiriasi. Ir kartais būna jis toks ganėtinai scientific, kartais jis toks būna. Na, maždaug jo, čia jau dabar gerai turbūt, nes geriau negu buvo. Iš esmės tai labai skiriasi irgi.

I.: Ar koreliacija nelygu priežastingumui?

P.: Taip, jeigu trumpai atsakant.

I.: O jeigu vis dar. Ar vis dar nelygu?

P.: Iš esmės tai kaip. Turbūt reikia... Dažnu atveju tas nelygu, bet turbūt reikia įsivertinti tą laiko tarpą, tą stebėjimą, ar ne. Nes, tarkim jeigu tu tris ar keturis metus iš eilės gali pasakyti apie tam tikrus dalykus, kad jie koreliuos, bet nėra priežastiniai, ar ne. Tu tuos tris ar keturis metus gali juos gerai papredict'int, kas yra ganėtinai gerai ar ne. Nesvarbu, ar ten priežastinis ryšys, ar ne. Tai jeigu, bent jau mano požiūriu, jeigu tas vyksta ganėtinai ilgą laiką, tai nėra būtina, kad tas būtų priežastingumas. Užtenka ir koreliacijos, nes iš esmės visi ML modeliai jie anksčiau ar vėliau degrade'ina, nes keičiasi ir vartotojų kažkokie lūkesčiai keičiasi. Ir kaip jie vartoja kažkokią sistemą, produktą. Tai jeigu tu gali tikrai moksliškai patikrinti tą priežastingumą, tai super. Bet kartais ir ta koreliacija, ypač jeigu ji trunka šiek tiek ilgesnį laiko tarpą, tai nereiktų visiškai užsimerkti ir sakyti, kad iš to negalima jokios naudos išpešti.

Informantas Nr. 2

Amžius: 30 m.

Lytis: vyras

Darbo vieta: dabar dirba skaitmeninės rūbų pardavimo platformos kūrime; 8 m. darbo patirties įvairiose vietose. Vyresnysis sprendimų mokslininkas („staff decision scientist“).

Gyvenamoji šalis: Lietuva

Pilietybė: lietuvis

Formatas: gyvas pokalbis

Trukmė: 42min 44s

Data: 2024 m. spalio 31 d.

I.: Esu Simonas Bansevičius, politikos ir medijų magistro studentas. Darau tyrimą apie mašininio mokymosi programų etiką. Informuoju, jog bet kada gali nuspręsti nutraukti pokalbį. Ar sutinki, kad šis pokalbis būtų įrašytas ir jo medžiaga būtų naudojama tyrimo tikslais?

P.: Taip.

I.: Prisistatyk, jei gali, pasakydamas amžių, ką dirbi šiuo metu, kiek laiko dirbi šioje srityje.

P.: Taip, tai esu (anonimizuota). Man yra man atrodo 30 metų. Dirbu šitoj srity nuo 2016 m. Dirbau skirtinguose taikymuose dirbtinio intelekto: medicinoj, klimato kaitos analizėj ir consumer products.

I.: Klausimas pasidalinti apie dabartinę vietą kur dirbi? Kiek žmonių komandoje? Tiesiog kaip komanda atrodo visi tie specifiniai dalykai.

P.: Dabar dirbu įmonėje, kuri yra turgavietė, žmonėms prekiauti rūbais arba kažkokiais kitais. Nežinau item'ais. Komanda, kurioje dirbu. Ji nedirba, dirba ne tiek su dirbtiniu intelektu, kiek su matavimais, statistiniais, taip vadinamais eksperimentais, AB testais. Komandoje yra gal apie 10 žmonių inžinieriai ir 3 duomenų mokslininkai. Pagrinde mes dirbam, kaip pritaikyti tuos matavimo būdus visai įmonei, įskaitant dirbtinio intelekto modeliams, kurie yra panaudojami produkte. Nežinau, ar atsakiau.

I.: Jo gerai. Bet tai tau vis tiek reikia gauti duomenis, daryt dalykus.

P.: Duomenys yra. Nes ta prasme įmonė turi tuos duomenis ir tuos duomenis, yra iš tų duomenų yra sukuriamos tam tikros metrikos, pavyzdžiui, kiek yra spaudžiama ant tam tikros vietos, ir iš to tu gali sukurti kažkokią tai metriką.

I.: Tai man buvo klausimas, man yra klausimas, kaip gauna duomenis?

P.: Ai, kaip gaunama, kaip gaunama duomenys. Duomenys ateina. Pavyzdžiui, jeigu useris, vartotojas nuperka kažką, tai tai jau yra duomenys, nes tai tas veiksmas, kad kažkas buvo nupirkta, yra užregistruojamas duomenų bazėje ir tai jau yra duomenys, kuriuos tu gali naudoti analizėje, modeliams ar pan.

I.: Pvz., kaip vyksta dinamika tarp implicit explicit duomenų rinkimo.

P.: Implicit ir explicit duomenų?

I.: Nu pvz. ar užpildyti formą, ten pasirinkt kažką, kas tau labiausiai patinka, ar tiesiog žmogui sąmoningai nežinant, kad jo elgesys sekamas, tai irgi paverčiamas duomenim.

P.: Šito nežinau, čia priklauso nuo to, ar buvo implementuoti kažkokie event'ai, kurie leistų matuoti, pvz. kiek užtrunka viena ar kita užduotis. Tam vartotojui ten nueiti iš vienos formos į kitą formą. Kiek tai gali užtrukti laiko? Tai ne visada ir ne visos komandos yra suimplementavę kažkokius eventus, kad tą išmatuoti ir kartais reikia tą suimplementuot, kad sugebėtų išmatuoti. Tai vat čia ir skirtumas - yra duomenys, kurie jau yra matuojami, yra kurie nėra matuojami.

I.: O kaip nusprendžiate, kuriuos matuot, kurių ne?

P.: Priklauso nuo hipotezės. Kai tu statai kažkokį produktą, tu turi pradėti nuo strateginės vizijos, kad mes ten norim pastatyti pvz. chatbotą. Netgi ne tai. Tu nori pradėti nuo problemos. Kokią problemą turi vartotojai? Kaip tai susiję su tavo verslo problema? Na ir turi būti kažkoks geras santykis tarp jų. Ir tada tu iš to iškeli hipotezę, kad jeigu mes padarysime chatbotą, tada išspręsime tokią vartotojo problemą ir ta vartotojo problemą tu matuosi per tam tikrą metriką. Kaip transakcijų kiekis padidėja arba kad galbūt user'ių vartotojų pasitenkinimas platforma padidės, ar rodant per kažkokią kitą metrika ir tu pastatai tada tą sprendimą, kurį tu sugalvoji, paleidi ir žiūri, ar ta metrika pagerės, ar ne.

I.: O kaip nustatot, ar naudotojas dabar laimingesnis, ar ne?

P.: Pirma, reikia atsakyti, kas yra laimingesnis. Manau, čia yra labai sudėtingas klausimas. Mes žiūrime, kad vartotojai ateina į mūsų produktą daryti kažkokį pardavimą arba pirkimą. Ir jeigu jie sugeba nupirkti ar parduoti daugiau ar mažiau, tai reiškia, jie pasiekia savo tikslus. Čia yra pakankamai paprastas, taip pat, kaip ateiname į Maximą ir suradome tuos pomidorus, kuriuos norėjome surasti. Jeigu suradome, tai pirkimas vyksta, jeigu nesuradome ir dalykai nebuvo sudėlioti labai gerai Maximoje, tai tas pardavimas, pirkimas ir pardavimas neįvyks.

I.: Ar yra kitokių būdų nustatyti laimingumą, be pirkimų?

P.: Manau klausimas yra sudėtingas, nes neaišku, kas yra laimingumas. Čia yra taip pat, kaip kalbėti apie dirbtinį intelektą. Kas yra intelektas? Niekas negali atsakyti, kas yra intelektas. Tai čia taip pat ir su laimingumu. Jeigu būtų labai konkretus apibrėžimas, tai galėčiau duoti labai konkretų atsakymą į šitą klausimą.

I.: Bet šituo atveju tai realiai tik tais pirkimai?

P.: Šitų dalykų negaliu atsakyti. Yra daug skirtingų metrikų. Jų yra keli šimtai. Ir kokios jos yra, aš negaliu sakyti. Jo, bet bendrai yra daug pakankamai literatūros, pavyzdžiui, iš Netflix įmonės iš LinkedIn, įmonė įmonių iš Spotify, Booking.com, kuri parodo, kad bendrai įmonė stengiasi optimizuoti ne vien tik pirkimus, bet ir vartotojo sugrįžimą į platformą, kad jeigu jie sugrįžta, reiškia jiems jų patirtis prieš tai buvo gera. Jiems patinka platforma ir tada jie optimizuoja dažniausiai taip vadinamą retention. Nežinau, kaip retention'as lietuviškai. Galima šitaip. Tai optimizuoja retention'ą. Bet tam reikia sukurti kitus AI modelius, kurie predictin'a, koks bus retention'as, žinant kažkokias trumpalaikes metrikas ir iš to. Ir tą bando optimizuoti.

I.: Gerai. Nors negali sakyti, kokios yra metrikos, bet kai galima pasirinkti, kaip nustatote, kokios metrikos? Tipo procesą metrikos sugalvojimo.

P.: Jis yra labai visapusiškas procesas. Kartais tai yra labai strateginis klausimas. Pavyzdžiui, jeigu tu esi Spotify, tu žinai, kad tau yra labai svarbu, kad būtų daugiau naujų menininkų, kurie post'ina savo muziką į tavo platformą. Nes tada tu žinai, kad bus, jeigu bus, tada žmonės, kurie ateina klausyti, jie galės surasti kuo daugiau skirtingų menininkų, kurie galės patenkinti jų norus, kokią muziką jie nori klausyti. Jie gali turėti metriką, kad kuo daugiau būtų ne vien tik naujų menininkų, bet kad ir būtų, kad tie tą muziką kurie, kurią pauploadina nauji menininkai, ji būtų taip pat klausoma. Tai čia vienas iš, iš kada tu iš to gali sugalvoti metriką, kiek yra unikalių perklausų unikaliems kažkokiems menininkams, nu ir ten dar pritaikai kažkokią normalizaciją to metriko.

I.: Pala kaip, kokia metrika?

P.: Pavyzdžiui perklausų skaičius jų unikaliems menininkams per mėnesį arba ten tu kaip tik paėmei su mažiausiai perklausų percentile menininkus ir tu bandai tai percentilei pakelti perklausas. Tai čia labai priklauso, kaip tu tą metriką suformuoji ir kaip tu ją formuoji, kokią formulę panaudoji. Labai turi implicit įtaka tam, ką verslas optimizuoja. Tai, pavyzdžiui, kitas pavyzdys yra YouTube. Youtube gali gerinti tiesiog rekomendacijas, labiausiai peržiūrimų video, bet tai reiškia, kad jie tik rekomenduos tada didžiausius YouTube'erius. Bet jie tikriausiai susidomėjo, kad ir maži YouTube'eriai, kurie tik pradeda, turėtų galimybę pasiekti savo žiūrovus. Ir tam reikia sukurti kažkokią metriką, kuri tai apibrėžia labai gerai.

I.: O šitam procese, kiek žmonių dalyvauja kuriant metrikas?

P.: Labai daug - visi dalyvauja šiame procese. Kaip Druckeris sakė "You get what you measure" tai yra verslo, man atrodo dėstytojo Amerikoje univere posakis. You get what you measure, tai kuo daugiau žmonių dalyvauja tame procese apibrėžti metrikas, tuo geresnį visi turės supratimą, ką jie bando pasiekti versle. Metrikos gali būti dizaininamos lokaliai, produktinėse komandose. Kažkokios metrikos gali būti strateginiai ir būti aptariamos kažkur labiau top level. Čia yra back and forth constantly.

I.: Tau pačiam tenka apdoroti kažkiek duomenis, kuriuos gauni, tvarkyti?

P.: Jo, jo. Aš esu analizavęs ir testus, skirtingų modelių testus ir kažkokių produktinių pakeitimų testus.

I.: Pavyzdžiui, turi, raw data ir ten trūksta kažko.

P.: Dažniausiai nėra, kad trūksta, nes vėlgi yra labai akademinis approach atsakyti į klausimus. Mes sugalvojame kažkokią strateginę kryptį. Mums reikia pagerinti kažkokią metriką ir per tai mes žinome, kad mes sprendžiame vartotojo problemą. Tada mes sugalvojome hipotezę, kad jeigu mes padarysime šitą produkto pakeitimą arba padarysime šitą modelį, tada mes pagerinsime šitą metriką. Ir tada ta hipotezė, kurią mes sukūrėme by default, reiškia, kad mes turime pasirūpinti, kad būtų duomenys atsakyti į tą hipotezę. Dažniausiai jau tie duomenys yra ir tada mes darome analizę, kur palygini userius su naujo modelio ir be naujo modelio ir tada. Vertini, ar tas sprendimas, kurį sugalvoji, patvirtina tau hipotezę, ar ne.

I.: O kai lygina tuos userius, tada irgi pagal tą metriką?

P.: Userius nelygina niekas.

I.: Kai sakei su nauju be naujo modelio?

P.: Tu padalini žmones, grupę. Taip pat kaip tu, pavyzdžiui, kaip testuoji vakciną naują. Turi placebo ir tu turi žmones, kuriems tu davei naują vakciną. Ir tu, nelyginant kiekvieną userį su kiekvienu useriu, tu paskaičiuoji vidurkį vienai grupei, kitai grupei ir palygini tuos vidurkius.

I.: Tai čia čia per AB testing?

P.: Jo.

I.: O AB testingo kūrimas. Ir šitą gali papasakoti? Ar čia realiai tas pats kas metrika tiesiog?

P.: Na, AB testas. AB testo idėja yra suformuojama tada, kai suformuojama hipotezė, kad jeigu mes padarysime tą, mes pagerinsime šitą metriką. Tada kad atsakyt tą hipotezę mums reikia padaryti AB testą arba kažkokį kitą eksperimentą. Yra daug skirtingų. Ir padarom pirma tu pastatai tą sprendimą, kuris tu nori patestuoti, o tada tu sudizainini AB testą. Ir suimplementuoji, tai ten nėra daug darbo, keleta eilučių kodo.

I.: Kaip vyksta modelio vertinimas?

P.: Jeigu tai yra kažkoks modelis. Visų pirma jis yra testuojamas kaip vadinamas offline testavimas, kai visi turi kažkokią offline metriką, kuri, kur tu vertini tą modelį pagal istorinius duomenis? Tada, kai offline modelio performansas tave tenkina, tu sakai OK. Dabar mes matom, tai jis yra good enough ir tada tu darai online testą ir online testas dažniausiai bus AB testas.

I.: O susiduriat su overfitting'ais?

P.: Sunku pasakyti, priklauso, kas yra overfitting'as. Pagal šiuolaikinę teoriją. Neuroninių tinklų tu turi gali turėt double descent, kai tu pradžioj overfittini, o po to modelis tavo realizuojasi su overfitting'u.

I.: Gali pakartoti garsiai?.

P.: Pagal šiuolaikinę teoriją, tu gali turėti double descent. Kai modelis yra over parametrizuotas, bet kadangi tu turi daug duomenų, jisai pradžioj overfittina. Po to kažkaip save pats reguliarizuoja ir po to vėl tą reikia generalizuoti. Tas overfittinimas nėra black and white, bet žinoma bendrai žmonės nori, kad modelis generalizuotų iš tavo treniruojamų duomenų į test duomenis. Tai yra standartinis procedure'as test validation ir train validation.

I.: Bet, tai kažkokios ribos, nusistatymai turi?

P.: Ovefittinimui?

P.: Jo jo jo.

P.: Čia taip pat, kaip ir su žmonių laime. Žmonių overfittinimą reikia labai aiškiai apsibrėžti. Overfittinimas turi tam tikrus matematinius apibrėžimus, bet priklauso nuo literatūros, kurios tu skaitai. Yra ten back base ir taip toliau. Visokie matematiniai terminai. Bet dažniausiai tai yra labiau, kad jeigu mes už treniruojam ant training duomenų, ar modelis performina taip pat gerai ant test duomenų. Tai iš tos pusės dažniausiai.

I.: O tada į priešingą pusę. Kaip nustato, kiek reikia pagerėti metrikai, kad būtų pasiektas rezultatas.

P.: Turi būti return of investment. Jei tu developinsi kažką savaitę ar pora dienų ir bus visai būsi patenkintas su mažu pagerėjimu. Bet jei developini kažką metus, tai tu turi turėti ir aukštesnį return of investment.

I.: Bet yra kažkokių tikslių specifinių apibrėžimų?

P.: Nėra. Tai ateina man atrodo su patirtim.

I.: Būna, kad nesutariat, kad vienas sako, kad jau čia pakankamai gerai buvo, o kiti sako, kad ne.

P.: Tai visą laiką.

pauzė

I.: Čia daugiau netęsi, taip?

P.: Nu ta prasme tai yra normali, kaip ir hipotezė testuojama duomenų moksle. Mes vis tiek, nors ir testuojame hipotezę. Daug diskusijų vyksta aplink tą hipotezę.

I.: Kuriuose tavo darbo procesuose matai daugiausiai galimos rizikos produkto kokybės užtikrinimui?

P.: Koks įdomus klausimas.

I.: Kokybės ta prasme, kaip tu supranti pats.

P.: Viskas priklauso nuo to, ant kiek geros mūsų metrikos. "You get what you measure", jeigu tu nesusikuri pakankamai gerą metriką ir jinai arba teiki metrikų, ir jie nepacover'ina viską, ką reikėtų paceover'int nu tai tu gali ir optimizuoti vieną metriką, bet tai atves iki kažko, ko tu nenorėjai, kad būtų toks rezultatas. Bet gerai su metrikom, kai tu jas bent jau turi, kad tu ne šiaip darai kažką ir negali įvertinti, kur tai veda. Tai geriau turėti metrikas ir turėti jų daug, negu neturėt jų išvis.

Pauzė.

P.: Bet čia daug apie AB testus šnekam. Tau labiau apie AI reikia ane?

I.: Nu tai čia viskas sueina. Kada galvoji dirbdamas apie galutinį naudotoją?

P.: Visą laiką, nes visą laiką galvoju apie metrikas. Ta prasme, mūsų metrikų tikslas metrikų yra kuo įmanoma apčiuopiamai apsibrėžti vartotojų sėkmę. Jeigu tu nori eiti paklausyti, atsidarai Spotify ir nori kažką paklausyt, tu jau turi intenciją. Ir klausimas ar mes galim išmatuoti, ar darydamas tą veiksmą mūsų platformoje, ar sugebėsi išpildyti tą intenciją? Viskas yra apie tai.

I.: Tada klausimas, kas turi daugiausiai įtakos pačiam galutiniam produkto rezultatui imant nuo paties naudotojo, tada duomenų rinkimo, modelio (nespėta užbaigti sakinio).

P.: Tai vartotojas. Vartotojai, jie gali tiesiog nenaudoti produkto ir viskas, ir viskas tuo pasibaigs.

I.: O be jų?

P.: Ekonomika.

I.: O, šito dar nebuvo atsakymo.

P.: Na taip, jeigu ten tavo produktas yra countercyclical ar cyclical, tai jeigu ekonomika kyla viskas gerai, visi naudos, jeigu ekonomika krenta ir niekas nenaudos. Niekas nesako, nes žmonės overattribute'ina success faktorių savo pačių veiksmams. Nors mūsų veiksmų rezultatai labai priklauso nuo daug external faktorių.

I.: Bet jeigu labiau žiūrint iš visų kūrimo procesų. Kaip sakau, nuo duomenų surinkimo iki modelio kūrimo ir panašiai, tai kuris iš tų?

P.: Hipotezių kokybė. Jeigu tu pradedi hipoteze, kad. Nu jeigu tu neturi hipotezių, tiesiog darai bet ką, ten kažką paship'ini ar kažką ne, tu tada neturi galimybės turėti aiškių learning'ų, kas veikia, kas neveikia nu ir tu negauni to feedback'o iš vartotojų, ką jie norėtų. Ir tada tu tiesiog prisikiši dalykų į produktą, ir viskas tuo ir pasibaigs. Gausi Microsoft Windows.

I.: Ką turi tuo omeny?

P.: Na, tarsi daug visokių dalykėlių.

I.: Galvojau, čia asmeninė nuomonė į Windows?

P.: Ne ne. Ne, čia blogas palyginimas. Šiaip yra tyrimų. Man atrodo bent keli, kurie parodo, kad startuoliai, kurie naudoja hypothesis driven development ar AB testus, jie generally tend to succeed more than those that don't.

I.: Kas tau darbe kelia stresą?

P.: Žmonės nekuria hipotezių. Nu bet koks, bet koks, bet kokia verslo ar strateginė kryptis, kurią mes pasirenkame, jinai yra pastatoma ant hipotezės ir ant prielaidų. Mums reikia. Kuo raiškiau mes įvardijame tas hipotezes ir prielaidas, tuo - (pabrėžianti pauzė) greičiau - mes galime įvertinti, ar ta strateginė kryptis yra verta mūsų dėmesio ir kuo mes greičiau tą padarome įvertinimą, tuo geriau mes galime perskirstyti resursus ir tuo greičiau galime spręsti tada vartotojų problemas, kurios iš tikrųjų rūpi. Todėl reikia įvardinti visada hipotezes ir prielaidas ir surasti būdą patestuoti svarbiausias prielaidas kuo greičiau, nes tada tu žinai, ok, galiu judėti toliau negu tu pacommit'ini statyt kažką metus, tada padarai ir pasirodo, kad visiškai nereikėjo to daryti ir tai yra labai sunku. Tai labai kitoks mąstymas, negu. Tai yra labai moksliškas mąstymas.

I.: Ką darai, jeigu artėja deadline'a, bet dar neveikia programa,produktas, kaip reikia?

P.: Sakai, kad dar nespėjai. Dažniausiai tai yra kartu sprendžiamas klausimas su stake holderiais - "ok, kodėl mes nespėjom, ką reikia dėl to padaryti ir t.t."

I.: Aš girdėjęs tokį pasakojimą, kad ten kažkas darė modelį ir negavo rezultato, kaip turėjo gauti, bet Deadline'as spaudė, tai žmogus paėmė savo testą, pakeitė, pakeitė biški, kad pakeltų procentą ir išleido produkciją tokį reikalą. Ar tokia teoriškai galėtų išvis nutikti situacija?

P.: Įmonėse, kur nenaudojami AB testai, online testai - jo. Bet įmonėse, kur viskas turi praeiti pro AB testą, nelabai suveiktų, nes in the end tau rūpi ne tavo training ir test data setas ar validacijos data setas performins. Tau rūpi, ar tas modelis pagerina metriką, kuri buvo apsibrėžta hipotezėj. Ir yra labai geras straipsnis iš Booking.com. Man atrodo figure 2 yra tam straipsny, kurie turi x ašy performance training teste ar test, ar offline teste ir y ašy performance ties verslo metrika ir jie rodo, kad santykis jis labai netiesinis.

I.: Straipsnis?

P.: Hard Fifty lessons from Booking.com. 150 Machine Learning Lessons from Booking.com.

I.: Gerai, o kur matytum grėsmių etikos užtikrinimui ML procese?

P.: Viskas yra apie metrikas. Jeigu metrikos pasirenkamos, kurios turi long term kažkokias tai pasekmes, kurios nėra apgalvotos. Tas gali ir sukurti kažkokių pasekmių, kurių mes nesitikėjome. Nors mes ir visos įmonės, tiek įmonės, kurios naudojasi hypothesis driven development, skiria žiauriai daug laiko mąstant apie metriką. Bet mes turime pavyzdį feisbukas arba instagramas, kur lyg ir galvojo daug apie metrikas. Bet jeigu šiek tiek pasveria labiau į engagement'o optimizavimą, tai gauna labai addictive appsus.

I.: Tai tu sakai, kad jūs šiaip. Kai kuriate metrikas, galvojate apie tai, kad nebūtų vien engagement'as?

P.: Jo jo jo. Visi galvoja apie tai.

I.: Nuo developerio iki?

P.: Nu priklauso nuo žmonių envolvment'o, bet bent jau žmonės, kurie dirba su modeliais, su duomenų analitika, tai daug apie tai svarsto.

I.: Bet yra kažkokios gairės, kaip apie tai galvoti?

P.: Vienas yra qualitative mąstymas. Ok, kas bus, jeigu mes darysime šitą metriką taip, ką mes tikimės? Ar turim praeitų testų, kurie parodo, kaip pasikeičia vartotojų elgesys, priklauso nuo to, ar padidėjo, ar pagerėjo šita metrika, koks šios metrikos santykis su kitomis metrikom. Bet sunku apgalvoti tokius dalykus, kad nu kaip Instagram atveju. Jeigu engagement didės, tai tada tas turės problemų su žmonių psichine būklė. Šituos dalykus apgalvoti sunku. Aš manau, niekas pasaulyje dar nėra, neturi būdų tą apmastyti.

I.: O neturint kokio nors savo privataus filosofo kaip Google?

P.: Ne, neturim. Aš net nežinojau, kad Google turi tokį.

I.: Jo jo, jie turi savo.

P.: Nu bet nelabai jam filosofui tam sekasi.

I.: Nu tipo ten etikos filosofas kažkoks, bet jie ten nieko neskelbia.

P.: Bet jie atleido gi etikos visą komandą, kurią jie turėjo.

I.: Ten, kur tas skandalas?

P.: Nu jo.

I.: Nu tai, dėl to neskelbia savo proceso.

P.: (nusijuokia) Jo.

I.: Bet iš esmės jo. Metrikas vertinat per etikos prizmę patys inžinieriai?.

P.: Jo

Tai tik tik čia šitos disciplinos? Verslas ir inžinerija? Psichologo neturit?

P.: Verslas ir Inžinerija, duomenų analitika yra. Dar yra didelė sfera visose tech įmonėse user research, kai tu darai apklausas. Useriams tu skambini, darai interviu ir taip toliau. Tai tas irgi formuoja metrikas. Ir į tą investuojama žiauriai daug, žiauriai daug.

I.: O kaip dažnai vyksta reguliariai.

P.: Kiekvieną dieną.

I.: Ta prasme, kad kasdien skambina?

P.: Ir skambina, ir daro surveys, ir optimizuoja pačius surveys. Kaip surinkti daugiau feedback? Tai tikrai daroma labai daug.

I.: Tai čia ne kaip planas, kad kas ketvirtį daryti.

P.: Ne čia yra reguliarus dalykas. Kiekviena produktinė komanda dažniausiai turi kažkokį paskirtą user researcherį, kurias visą laiką renka informaciją, skambinasi su useriais ir susitinka su useriais.

I.: Gyvai?

P.: Jo. Ta prasme Kasmet žmonės yra skraidinami į ofisą ir interview'inami.

I.: Bet ta prasme paprasčiausiai ne ten kad verslas, bet paprasčiausias žmogus?

P.: Jo jo. Per paprastus vartotojus reguliariai tas daroma. Tai yra visas mokslas, kaip tą daryti. Taip pat statistikos.

I.: O šitie researcher'iai, jie dažniausiai būna kokios disciplinos žmonės?

P.: Sociologijos. Sociologijos, psichologijos, statistikos, neuromokslo kartais. Tai jau netoli filosofų. Jo jie dažniausiai apibūdina informaciją tokią labiau jau abstraktesnę. Yra labai geras straipsnis iš Spotify Research triangle. Kad tu metrikom gali tik išmatuoti tik tiek iš produkto, kiek galėjai iš to

arba patvirtinti arba sugalvoti tam tikrų hipotezių tipus iš to. O su user research gali irgi paneigti arba sugalvoti tam tikrų user ir tam tikrų hipotezių ir jos turės vienas kitą, informuoti. Taip ir yra dažniausiai. Netgi kai Google, pvz., developina savo Search Engine jie gi turi žiauriai daug research, kur jie sėdi su žmonėm ir žiūri kaip jie jiems suveikė ar nesuveikė, tas search query'is kurį jie turėjo, ar ne. Man atrodo, jie ten turi visą komandą žmonių, kurie vien tik tuo ir užsiima. Aš manau, Instagramas ir Feisbukas turi tą patį.

I.: Ką atsakytum į tokį keistą klausimą. Ar koreliacija vis dar nelygu priežastingumui?

P.: Mes tą matom labai aiškiai iš duomenų, kad tai nėra. Todėl ir, pavyzdžiui, kaip sakiau, vienas iš svarbiausių metrikų visoms įmonėms yra retention. Kiek useriai sugrįžta, bet tu gali optimizuoti kažkokią trumpalaikę metriką, kuri tau atrodys ji koreliuoja su retention, bet iš tikrųjų tai nėra taip. Ir tau dėl to reikia daryti eksperimentus, kad ir iš tikrųjų išsiaiškinti, kokios yra trumpalaikės metrikos, kurios labiausiai causin'a userius sugrįžt.

I.: O skaitei gal tą straipsnį tokį. nežinau 2010 gal metų, kur vadinasi correlation is the new causation. Man rodos, ar kažkas tokio.

P.: Ne. Gal ir skaitęs, bet dabar neprisimenu.

I.: Žodžiu, žmogus ten sako, kad turim tiek daug duomenų ir ten galim surast. Ir koreliacijos mums užtenka.

P.: Ai ne. No. Bet čia dažniausiai žmonės ir galvoja, kad tokiose įmonėse kaip Facebook yra daug duomenų ir vien iš koreliacijos labai daug gali pamatyti. Tai jo, tu gali daug hipotezių iš to sugalvot, bet tas hipotezes vis tiek turi patestuoti. Tai tu darai kažkokią intervenciją ir pažiūri, ar tikrai tos metrikos juda taip, kaip tu tikėjaisi? Ir kitas svarbiausias dalykas, tai yra big data ir whatever we wanna call it. Tai yra taip pat low signal to noise. Nu ta prasme duomenų yra daug, bet signalų juose nėra daug. Yra netgi Google unofficial Data Science Blog, kur jie turi visą straipsnį apie tai, kad jo duomenų yra daug, bet signalas yra žiauriai mažas. Tai istoriškai statistika dealino. Statistikos mokslas dealino su mažai duomenų, stipriu signalu. O čia visiškai kita situacija. Daug duomenų, bet mažai signalų. Jo istoriškai statistika dirbo su mažai duomenų, daug signalo, o dabar yra daug duomenų, mažai signalų.

I.: Kaip manai, kiek patį naudotoją tiksliai vaizduoja tie duomenys? Kurie apie jį surenkami?

P.: Geras klausimas. Kadangi mes prieš tai diskutavome apie High amount of data low signal to noise. Tai reiškia, kad nepakankamai gerai. (nusijuokia) Nes low signal to noise.

I.: Ką reikia daryti, kad būtų geriau?

P.: Tai geresnes hipotezes turėt.

I.: Viskas, jau tikėjausi.

P.: (nusijuokia) Viskas atsitrenkia į tai. Nes tau ne visada ir ne visada ir reikia tų visų duomenų. Tai priklauso nuo to, kokia tavo hipotezė. Tau reikės vienos tam tikros metrikos, kurią reikia išmatuoti. Ir tiek.

I.: O, pavyzdžiui, kaip jautiesi pats naudodamas produktus, kurie naudoja ML AI?

P.: ChatGPT man patinka naudoti. Ten viskas yra AI. Labai padidina produktyvumą. Patinka naudoti YouTube, kai man geras rekomendacijas duoda. Google irgi patinka naudoti, kaip pavyzdžiui bandai switchinti iš Google Search engine'ą į kokį nors Duck duck go, tu supranti, kiek geresnis yra Google. Tai tikrai yra tų privalumų. Tai gerai jaučiuosi. Ką tik trinu, tai trinu visą laiką social media, nes engagement yra tikrai overoptimised to the point, kad neįmanoma naudotis. Šiaip manau social media yra blogai. Pavyzdžiui, važiuoji autobusu ir visi sėdi Tik Toke. Man atrodo čia yra epidemija, kurią mes dar kaip visuomenė neįsivaizduojam, kad ji vyksta ir kokią didžiulę įtaką ji turi. Ir manau problema dažniausiai yra tame, kad žmonės nežino, kaip šitos platformos veikia. Čia kaip ir su visom technologijom mes sugalvojom, bet bendra visuomenė ne visada yra tam pasiruošusi tom technologijom.

I.: Nesusimąstai kartais, kad jūs irgi kažkiek stengiantis, engagement (nebaigtas sakinys).

P.: Pastoviai apie tai mąstome. Ar tikrai mes darome gerai, ar negerai ir tam daromos user research. Kaip sakiau visur, kur šnekama su vartotojais ir vertinama tai.

I.: Čia grynai iš engagement'o pusės?

P.: Nu matai, produktas šiek tiek skiriasi. Tai nėra social media, kur tu ten, labai aiškus tikslas. Žmonės dažniausiai ateina nupirkt arba parduot ir tiek.

I.: Nu bet vistiek tarsi galėtum sukurt.

P.: Nelabai veikia.

I.: Kaip tas per didelis pirkimo.

P.: Jau kaip ir testuotos hipotezės, kad žmonės ateina dažniausiai su labai aiškia intencija ir tu gali tik tą intenciją padėt arba nepadėt. Tai čia ir yra ta visą laiką. Ar pirma yra paklausa, ar pirma yra supply. What comes first, supply ar demand? Tai yra ir apie tuo social media puslapius galima pasakyti, kad galbūt jos nėra, nu jos tik atsako į ta demand su savo supply, negu jie duoda supply ir tada susikuria demand. Nes, pavyzdžiui, Facebook'as, kai daro tyrimus apie social media įtakos psichologinę. Šitie tai jie nesuranda jokio efekto.

I.: Ką turi omeny neranda?

P.: Labai daug tyrimų buvo iš Facebook per paskutinius kelis metus apie poveikį Facebook ir Instagram psichologijai žmonių, ir jie claim'ina, kad nėra efekto, nors sunku šitą iš tikrųjų ištirti.

I.: Nu jo, čia.

P.: Ir čia yra labai stiprus network efektai, tai yra network efektai padaryti aišku causal matavimus yra labai sunku. Tad ar kitas žmogus šalia tavęs naudoja tą platformą, ar ne, tai įtakoja, kaip tu jautiesi ir taip toliau.

I.: Tai tu sakai, kad soc. medijos atsako į demand, o ne supply?

P.: Nežinau. That's the point - I don't know.

I.: O jūs atsakote į?

P.: Irgi nežinau, bet labai pas mus matosi, kad yra demand. Žmonės ateina, nori palistint, ir tu padedi jiems palistint. Bet čia labai vague, labai nekonkretu, tiesiog kaip vieną iš mąstymo būdų apie tai pasiūlo.

I.: Turbūt jau žinosiu atsakymą, bet dėl visa ko. Kas yra geras, geras ML?

P.: Kuris... Ne, tai ML nėra nei geras, nei blogas.

I.: Nu jo, kokybiškai sukurtas?

P.: Nu čia jau daug kampų galima. Kas yra ta kokybė? Žinoma, įmonės kontekste. Ar tai yra modelis, kuris gerina tam tikrą metriką, ar blogina? Priklauso nuo hipotezės vėl, bet bendrai ar geras ar blogas modelis jau priklauso nuo tų. Jo, tai viskas priklauso nuo tų kriterijų, kuriuos mes iškeliame tam modeliui. Pavyzdžiui, ta knyga yra Weapons of math destruction. Ten labai daug gerų pavyzdžių duoda, kur jo, kaip ir kuria modelius su gera intencija. Bet tada tie modeliai turi pasekmes, apie kurias kūrėjai nebuvo pagalvoję.

Informantas Nr. 3

Amžius: 27 m.

Lytis: vyras

Darbo vieta: šiuo metu kuria savo ML duomenų generavimo produktą; iki tol 4.5 metų dirbo versle kompiuterinės regos ML programų kūrime (1 m. Lietuvoje, po to Nyderlanduose)

Gyvenamoji šalis: Nyderlandai

Pilietybė: lietuvis

Formatas: nuotolinis pokalbis

Trukmė: 49min 38s

Data: 2024 m. lapkričio 18 d.

I.: Esu Simonas Bansevičius, politikos ir medijų magistro studentas. Darau tyrimą apie mašininio mokymosi programų etiką. Informuoju, jog bet kada gali nuspręsti nutraukti pokalbį. Ar sutinki, kad šis pokalbis būtų įrašytas ir jo medžiaga būtų naudojama tyrimo tikslais?

P.: Taip, sutinku.

I.: Prisistatyk pasakydamas jei gali savo amžių, ką dirbi šiuo metu, kur dirbai prieš tai, jei tai aktualu, kiek laiko? Žodžiu, viską, kas tau atrodo į temą.

P.: Esu (anonimizuota), mašininio mokymosi inžinierius, kuris daugiausiai savo karjeroje laiko praleido su kompiuterine rega. Mokydamas Mašininio, mašininius modelis su nuotraukomis ir vaizdo įrašais. Dirbau Lietuvoje metus ties veidų atpažinimo ir veidų paieškos algoritmais ir po to išsikrausčiau į Amsterdamą, kur dirbau apie 2 metus prie panašių sistemų. Tai irgi veidų atpažinimas. Nežinau kaip lietuvių kalba, bet liveliness Detection, kai atpažįstamas žmogus yra gyvas ir nedėvi kažkokių kaukių ir panašiai. Ir prie dokumentų apdorojimo. Dokumentų turiu omenyje tai pasų, asmens identifikavimo dokumentų ir pan.. Ir paskutinius metus mažiau leidžiu dirbdamas. Kuriu savo įmonę. Ką mes darome tai. Kuriant nuotraukų duomenų rinkinius įmonėms, kurios nori apmokyti savo kompiuterinės regos modelius, bet tai darome kurdami sintetinius duomenis. Čia trumpai apie mane. Turiu apie keturis su puse metų komercinės patirties darbo. Tai, mažiau šiek tiek gal, tai.

I.: Gali daugiau išsiplėsti apie dabartinį darbą, savo dabartinę įmonę?

P.: Jo kol kas, kaip mes, pačios įmonės įkūrę nesam, tiesiog per freelance ten tą account dirbam, nes tiesiog dar neteko tiek išsiscale'int, nebuvo tokių labai kažkokių apčiuopiamų kontraktų ar kažko panašaus. Bet ką mes darome tai? Pavyzdžiui, įmonės kai jos pradeda kažkokį projektą su mašininio mokymosi, treniruoja mašininio mokymosi modelį su nuotraukomis ar vaizdo įrašais, jam reikia daug duomenų, daug nuotraukų ir jas reikia sužymėti. Ir tas užtrunka daug lėšų ir daug žmogiško darbo. Ką mes darome ir dauguma duomenų, kuriuos įmonės surenka ir paskui toliau žymimi, yra gana panašūs į tuos, kuriuos jau prieš tai žymėjo, tai tiesiog pridėtos naujos nuotraukos į tą duomenų rinkinį. Nepadeda tiek daug tam modeliui. Ir tas užtrunka ilgai. Ką mes darome, tai darome taip, kad įmonėms reikėtų maždaug tik apie 10 procentų duomenų sužymėti ir surinkti. O visus kitus duomenis mes tiesiog pasiimdami, kaip pavyzdžiui, esamas nuotraukas, sugeneruojam su įvariais įvairiais modeliais, sugeneruojam tuo pačiu ir nuotraukų annotationus. Tuos label'ius ir taip sutaupoma apie kokį 80% laiko ir lėšų, susijusių su duomenų rinkimu ir žymėjimu. Tai įmonės gali tiesiog sugaudyti duomenis, kurie yra gan paprasti ir ir tam tikram smulkiam domenui. Kaip čia pasakyti iš tos siauros srities ir mes tuos duomenis daug labiau išplečianm, išdiversifikuojam ir sukuriam naujus ir modeliai tiesiog veikia geriau, tiksliau ir už mažesnę kainą. Sakykim taip.

I.: O yra su tuo atsirandančių kažkokių problemų?

P.: Pavyzdžiui?

I.: Tai ir klausiu, ar yra?

P.: Jo, bet turiu omenyje, kokias problemas turi omenyje, ar komercines, ar inžinerines ir?

I.: Nu inžinerines pavyzdžiui.

P.: Sunku pasakyti taip, nes žinai tas problemų spektras yra labai platus. Potencialiai tai. Sunku iš tikrųjų pasakyti, kad nu tiesiog nėra kažkokių tokių problemų, kurias aš pasakyčiau. Ir kai tu pasakai žinai šitą klausimą, kurios tiesiog ant liežuvio galo būtų, tai sunku pasakyti. Jeigu turėtum kokias, ta prasme labiau koncentruotą sritį, tai būtų lengviau atsakyti.

I.: Tai, pavyzdžiui. Tai gal labiau. Gerai, jūs gaunate 10 procentų tų duomenų, tai tada likusius, kai generuojat, generuojat, kad būtų panašūs į tuos 10 proc.? Ar pagal. Vis tiek, kur ta bazė, nuo kurios jūs atsispiriat?

P.: Būna kartais tai, kad, tarkim, klientas nori plėstis į kitas rinkas ir jam reikia, tarkim, atsiranda apmokymo duomenys iš vieno konteksto, bet jam reikės modelio deployint kitame kontekste, ir mes norime padėti sugeneruoti duomenis kuo panašesnius į tą kitą kontekstą. Bet kartais klientas puikiai dar pilnai nesupranta, kaip tas kontekstas atrodys. Tai mes tiesiog generuojame daug labai skirtingų duomenų ir tiesiog bandome sužvejoti, sakykime, vieną žuvį su plačiu tinklu. Tai gal tai sakykim tokia problema yra, bet dar didesnė problema yra, jeigu tie duomenys išvis nebūtų sugeneruoti ir klientas jų neturėtų pačioj pradžioj, tai tokie bandom spręst didelę problemą ir iš jos lieka mažesnė problema. Gal jeigu, tarkim, papasakočiau kažką iš praeitų darbų, kuriuos dirbau įmonėse, gal net būtų paprasčiau atsakyti, nes man atrodo, būdavo labiau specializuoti tokie task'ai ir panašiai.

I.: Tai apie tai irgi pašnekėsim, bet buvo įdomu dabar apie šitą.

P.: Liuks!

I.: Bet kad jau paminėjai, tai ir galima eiti prie praeities.

P.: Kaip, kaip nori.

I.: Tai gerai, dabar generuoji duomenis, o anksčiau iš kur gaudavai duomenis, kai dirbai jau rinkoje?

P.: Daugiausiai būdavo arba įmonės privatūs duomenys, kuriuos įmonė turėdavo iš savo produkto arba nusipirkdavo, arba iš kitų šaltinių, arba iš Open Source visokių Research duomenų rinkinių, kurie buvo leidžiami komerciniam naudojimui. Pagrinde iš tų sprendimų. Dar būdavo iš third party visokių įmonių, kurios pardavinėdavo duomenis, dažniausiai tiesiog stengdavomės gauti duomenis iš įmanoma, kur tik tais įmanoma. Galima sakyti, ar taip patys susigeneruoti, ar iš savo produkto, ar iš third party, ar

iš open source. Tiesiog rinkdavomės sekantį pigiausią variantą ir lengviausią variantą. Tą tą sekantį variantą pasiėmę siekdavom kito lengviausio varianto. Ir taip tiesiog palaipsniui, vis.

I.: Tipo sekantį, kad ne pats pats lengviausias būdas nuo galo?

P.: Ne, turiu omeny imti patį lengviausią variantą, jį išnaudodavom tą resursą, ir tada siekdavom kito lengviausio varianto ir tiesiog ar tai third party, ar tai būtų iš vidinių resursų ir pan. Tiesiog taip efektyviai laiką ir lėšas naudojant.

I.: O kai pirkdavot. Ten, kur pirkdavot, ten būdavo realūs ar irgi kartais generuoti duomenys?

P.: Būdavo realūs.

I.: Bet vis tiek ten nuasmeninti duomenys?

P.: Priklausė. Kartais būdavo ir duomenys pilnai su. Tarkim, jeigu būdavo žmonių veidų duomenys būdavo. Sakykim, tu nori nusipirkt tokių duomenų ar tai tarkim iš įvairių data providers ten nežinau, pradedant nuo visokių Shutterstock ir pan., kurie pardavinėja juos modelių, trainint duomenis. Ar ten nežinau. Šiaip kitokių įmonių, kurios specializuojasi ties data source'inimu, tai jos nepadarys taip, kad jie būtų duomenis anonimizuoti, bet jie padarys taip, kad žmonės duoda pilnai consent'ą ir taip toliau. Ir tu perki duomenis ir tu, jo, tiesiog. Visi žmonės, kurie parduoda duomenis, jie sutiko, kad jų duomenys būtų naudojami.

I.: Bet tokiu atveju, bet iš savo pačios platformos savos pačios sistemos suprantu nerinkdavot duomenų iš savo naudotojų.

P.: Nežinau, ar galiu atsakyti į šitą klausimą ir ties kiek galiu atsakyti.

I.: Tai jau kiek tau galima, neik už ribos.

P.: Jo bet šiaip dauguma įmonių renka ir naudoja taip galėčiau pasakyti. Dažniausiai čia viskas normalu ir. Jo. Dažniausiai vartotojai duoda concent'ą, kad jų duomenys gali būti naudojami ar tiesiog tai tie duomenys gaunami iš partnerių ir įvairiai įvairių panašių resursų, kad tobulinti AI modelius ir tas pats user experience būtų patogesnis.

I.: Susidurdavai su kokiom nors problemom anotuojant nuotraukas?

P.: Taip, dažniausiai tai būdavo tiesiog, kad tas ilgai užtrunka, ypatingai jeigu duomenų kiekiai dideli, tai tarkim milijonais nuotraukų. Jeigu matuosime, tai tikrai. Įmonėse, kuriose dirbau, reikdavo daug žmonių tam skirti - dešimtimis. Antras dalykas tai, kad kuo labiau specializuotas sprendimas AI, tai tuo labiau specializuotas ir duomenų žymėjimo protokolas būna. Ir kartais jam reikdavo net ir atskirą softą parašyti. Kiekvienam iš tų žymėjimo duomenų žymėjimo use case'ų. Nes tiesiog atimdavo dar ir inžinerijos inžinieriaus laiką, tai reikėdavo tiesiog ir daug žmonių laiko ir daug infrastruktūros tam. Ir gal būdavo tas, kad kai kurie žmonės tiesiog gal ne taip tiksliai žymėdavo duomenis ir būdavo visą laik

tas žmogiškasis elementas, kad tau reikia duomenų žymėjimo tiksliai ir išsamiai. Bet būdavo, kad skirtingi žmonės žymi skirtingai arba kai tas pats žmogus skirtingu paros metu ar kažką mažiau kokybiškai, išsiblaško ar pavargsta, su tuo būdavo šiek tiek sunkiau susidoroti ir susidoroti su tuo. Tas kainuodavo dar papildomas lėšas, nes reikėdavo, kad, pavyzdžiui, trys skirtingi žmonės žymėtų tą patį duomenį ir iš jų išvestų kažkokį Konsensusą ar sprendimą.

I.: Kai sakai, kad reikia software rašyti, tai jis automatiškai anotuodavo. Ar čia kaip įrankis?

P.: Tiesiog įrankis tiems patiems žmonėms anotuoti. Tarkim, gal, tarkim, kai kuriais atvejais, kad žymėti duomenis, reikia parodyti žmogui 10 nuotraukų ir pasakyti, ar čia tas pats daiktas, ar ne tas pats daiktas. Ir čia labai retas, use case būtų. Dėl to reikia parašyti atskirą user interfeisą tam ir kažkiek back end dalykų. Ar tarkim, jo kažkas tokio.

I.: Ar tekę anotuojant susidurti su kokia nors situacija, kad renkiesi tarp dviejų galimų pasirinkimų ir tenka. Nu jo, pasirinkti arba tą, arba tą kažkokį kompromisą priimti?

P.: Labiau turiu omeny, kad būdavo kažkoks nesutarimas ar?

I.: Nu jo pvz. nesi tikras, ar čia tas, ar tas atvejis.

P.: Jo. Būdavo, kad realiai visą laiką būdavo kiekviename projekte. Kartais net padarydavom specialią klasę tam, kad tiesiog būdavo duomenys, kurie neturi aiškaus pripažinimo. Kiekvienas projektas, ties kuriuo dirbau, bet kokiu atveju vią laiką turėdavo, tai ir tas atimdavo kartais net nemažai laiko, nes reikėdavo nuspręsti labai aiškias ir tikslias taisykles. Taip būdavo.

I.: Ir tas taisykles nuspręsdavot jūs patys. Ar jums kažkas padėdavo, patardavo?

P.: Dažniausiai patys. Neretai tiesiog bendrauji su produkto žmonėmis, kurie nusprendžia, kaip. Kaip šis produktas, kurį mes buildinam. Šitas šitas AI modelis, kurį treniruosim, kaip mes norim, kad jis prognozuotų dalykus, kai ateis duomenys. Ir stengiamės nuo to tiesiog. Tą taisyklę pernešti į duomenų žymėjimą, kad duomenys būtų žymimi lygiai taip pat, lyg ateitų iš iš vartotojo. Ir kuo labiau sutaptų, kaip naudojamas bus modelis production'e.

I.: O ar tekdavo kaip nors manipuliuoti, tvarkyti ar kažkaip kitaip keisti turimus duomenis?

P.: Visą laiką, kai turėdavome duomenų, juos augmentuodavom vienaip ar kitaip, tiesiog, kad išgauti daugiau duomenų, tai. Duomenų augmentacijos čia nuo patamsinimo, pašviesinimo, visokių rotation'ų ir pan. Tiesiog primityvus basic duomenų augmentavimas. Kartais prašydavom duomenų žymėtųjų, kad jie sužymėtų duomenis tiksliau, tai tarkim. Dabar galvoju. Ai! Tarkim prašydavau, kad sužymėti, tarkim 3 žmonės tuos pačius duomenis ir surasdavom, kurie duomenys tiksliai sužymėti ir kurie turėjo mažesnį konsensusą. Pavyzdžiui, du žmonės pasakė, kad šita nuotrauka, tarkim, yra prastos kokybės, o vienas žmogus pasakė, kad šita nuotrauka yra geros kokybės ar kažkas tokio. Ir pagal tai,

paskui bandydavome konstruoti skirtingus duomenų rinkinius treniravimui, tarkim, kartais tik su labai švariais duomenimis, o kartais su ir tais, kurie nelabai švarūs. Ir pavyzdžiui, manipuliuojant tuos duomenų rinkinius, kad išsiaiškint, kurie geriausiai apmokami modeliai. Tai tarkim toks pavyzdys. Aišku, jeigu labiau specifikuotum tam, ką turiu omeny sakydamas, kad manipuliuoti tuos duomenis, galėčiau išsiplėsti šituo klausimu. Bet čia kol kas kas tik kyla man.

I.: Na, aš pats nelabai žinojau, tai tiesiog paklausiau ir tu atsakei.

P.: Ai, gerai. Gerai, gerai.

I.: Kaip atrodo sukurto modelio vertinimo procesas?

P.: Hm. Visų pirma labai priklauso nuo paties modelio ir nuo užduoties, kurią sprendžiame. Pavyzdžiui, objekto aptikimas gali skirtingas metrikas turėti nuo klasifikacijos ir pan. Pirma, tiesiog vertiname objektyvų. Objektyvų modelio performansą tiesiog prie paprastų metrikų, kaip jis? Koks jo true positive rate'as, tarkim palyginti su false positive ir tiesiog žiūrime į visą tą kreivę, ar turime labiau kažkokias konkrečias vieno skaičiaus metrikas? Kaip mean average ar kažkas tokio. Po to eina labiau tokie specifiniai ir gal žmogiškesni, mažiau objektyvūs vertinimai, kad tarkim, reikia išsirinkti, kad. Tarkim, turim tą TPR FPR kreivę. Kur dėsim tą treshold'ą, kuris nuspręs, kiek mes galim leisti modeliui dažnai suklysti? Kokia teisingų sprendimų kaina? Ir tiesiog tai tas jau būna labiau žmogiškas elementas. Ar tarkim, pasiėmėm daug nuotraukų ir tiesiog jas duodame modeliui arba tiesiog duodame tą duomenų rinkinį testui skirti ir žiūrim grynai akimis, grynai į konkrečias nuotraukas, ne į kreives, ir vertinam, kur modelis suklydo. Ar jis yra, kad klystų visą laiką ties kažkokiom konkrečiom nuotraukom. Ir jeigu mes žinome, kad production'e bus daug tokių nuotraukų. Ar galim leisti modelį į production'ą? Tai toks būna pirmiausia toks labai objektyvus ir metrikom grįstas. Bet paskui jeigu jau, kai reikia arba toliau pertreniruoti modelį, arba dėti į production'ą, labiau toksai žmogiškais elementais paverstas ir priklausomai nuo to, ar pvz. tas modelis bus grynai veiks autonomiškai, ar bus kažkoks Human in the Loop setup'as. Priklausomai pagal tai, mes nusprendžiam, kiek modelis daug gali klysti, kokios klaidos labiau pavojingesnės modeliui ir panašiai, ir pagal tai sukalibruojam.

I.: Gali išsiplėsti ties tuo atveju, kaip sakei, priklauso nuo to, ar bus Human in the loop. Tai, ką Human in the Loop padarytų?

P.: Bendrai. Tiesiog gali būti du setup'ai. Pirmas, kur modelis, kaip sakiau, autonomiškai veikia ir ką jis pasako, tas yra kaip objektyvi tiesa. Bet kartais galima padaryti, kad, pavyzdžiui, jeigu modelis yra užtikrintas, kad taip, kad, tarkime, ką aš žinau, aš turiu Face Detection modelį ir jis bando atpažinti tave kaip Simoną ir jisai, modelis sako griežtai, kad šitoj nuotraukoj yra Simonas arba labai griežtai, kad šitoj nuotraukoj ne Simonas. Bet jei jis neapsisprendęs yra kažkur per vidurį, mes galim tą tą viduriuką paimti

ir suprasti, kad jo, jeigu tas confidence tarp 0,6 ir 0,4, duodam žmogui peržiūrėti ir nuspręsti ar čia tikrai Simonas ar ne? Ne visada ne visąlaik yra tam prabanga, nes kartais produktas turi būti greitas ir pigus, bet kartais, jeigu tai yra labiau toks jautresnis use case'as, galim leisti sau tai padaryti ir. Tuo atveju absorbuojam tas modelio klaidas. Tai pagal tai galima spręsti, tiesiog kaip visas tas bendras workshop, sakykim tas visas pipeline'as, kiek jis bus netikslus, Įskaitant ir tą žmogų, kuris atliks sprendimus, kai modelis neteisus. O jeigu žmogaus nėra, kaip tada? Kiek bus klaidų padaryta ir kiek brangiai tos klaidos kainuos? Tai tiesiog. Mes bandome suprasti setup'ą visą tą ir pagal tai išimt klaidas ir tiesiog minimizuoti bendrai visos sistemos tą negatyvų poveikį. Statistiškai įvertinti.

I.: Kaip sakei paskutinį sakinį minimizuoti sistemos negatyvų poveikį.

P.: Taip, sakykim, tiesiog minimizuoti modelio klaidas. Bet ne tik modelio, bet ir modelio + žmogaus. Sakykim taip. Arba apskritai tiesiog suprasti, koks bus error rate paties modelio ir absorbuoti tai leisti absorbuoti tai žmogui. Ir dar galų gale įmanoma suprasti, kiek žmogus gali atlikti, kad absorbuoti tas klaidas, ką jis gali daryti ir bendrai suprasti visą sistemą, įskaitant ir žmogų. Kiek ji ir kokia jos klaida gali būti ir kokia ta klaidos kaina? Ir čia su visokiais produkto žmonėmis šnekėti ir bandyti tai išspręsti.

I.: O kai sakai, kad sveria, kiek brangiai klaidos kainuos, tai čia irgi tik developeriai sveria, ar platesnė komanda?

P.: Dažniausiai platesnė komanda, Dažniausiai šiaip. Tokiam gal labiau tradiciniame komandos etape, kaip pavyzdžiui. Tarkim, yra komanda, kuri atsakinga už kažkokį produktą, tai turės, tarkim vieną product ownerį, kokius du ML inžinierius, du back end inžinierius ir ten dar kokį vieną kitą žmogų kaip developerį ar kažkas tokio ar jo, ar ne developerį. Dažniausiai šiuo atveju mes šnekėtume su produkto produkto žmogumi, nes jis žino, kiek tas kainuos, kokios problemos bus ir panašiai. Ir jis tuo tarpu šneka su įvairiais verslo žmonėmis arba kitų, kitais produkto žmonėmis. Ir jis padeda mums, kaip inžinieriams, suprasti kiek įmanoma labiau, kokios yra rizikos, kokios yra kainos, kokias metrikas mes turime pasiekti ir apskritai kokios metrikos yra svarbios ir nesvarbios.

I.: Paminėjai metrikas. Metrikas gaudavot iš jų ar ir patys kurdavot?

P.: Dažniausiai mes nuspręsdavom, šiaip šiaip įvairiai. Dažniausiai būdavo bendras darbas nusprendžiant. Kokios metrikos bus, kokias mes vertinsime metrikas. Kažkiek gal produkto žmonės labiau linkę viską vertinti labiau tokiam vartotojų ir piniginėm metrikom. Tai kiek tas pinigų kainuos, kiek lėtai veiks, kiek tas greitai veiks. Kiek ten userių bus nepatenkinti ir panašiai. O mes labiau vertiname tomis tokiom inžinerinėm metrikom, kad kiek modelis ten dažnai klysta ir taip toliau ir panašiai. Ten precission recall curves ir pan. Ir dažniausiai bandydavome tai sujungti kažkiek. Jeigu modelis bus tiek netikslus, kiek userių bus nepatenkinti ir pan. Ir su žmonėm komunikuoti tokia bendra

kalba ir kiek įmanoma labiau apskaičiuoti iš. Kokį žmogišką tą poveikį turės. Ta tokia matematine metrika sakykim apjungti du konceptus.

I.: Ar esi susidūręs su prastomis metrikomis?

P.: Esu, dabar tik tai neatsimenu, gal kada ir kaip. Dažniausiai būdavo problemos, kaip pavyzdžiui. Dažniausiai, kai mes kurdavom modelius, juos reikėdavo vertinti labiau balanso kontekste. Tarkim. Žinai, tos metrikos, kaip būna, pavyzdžiui, true positive rate ir false positive rate., yra toks kaip trade off'as ir precission and recall irgi. Bet jeigu tu vertini metrikas kaip vieną objektyvų skaičių, tarkim mean average precision ar Average Precision, tai jo, jas gal į jas smagu pažiūrėti ir smagu kartais nuspręsti, kad ok šitas vienas skaičius yra geriau už tą vieną skaičių, bet kai žiūri į trade off'ą, tai ta ta metrika atrodo šiek tiek kitaip ir jinai. Jo tam tikram kontekste gali būti gera kitam kontekste gali būti bloga ir dėl to visąlaik žiūrėti į tas platesnes metrikas dažniausiai apsimokėdavo kuriant modelius, tik jas gal sunkiau objektyviai įvertinti. Jo ir pagrinde tas. Bet šiaip kažkaip nelabai. Man atrodo labiau yra problema ne metrikos blogumas savaime, bet kaip tos metrikos vertinamos objektyviai žmogaus ar subjektyviai atsiprašau žmogaus. Jo problema būna labiau interpretavime, o ne pačioj metrikoj.

I.: Tai ta interpretavimo problema. Čia tai ką jau minėjai?

P.: Kažkiek tas jo, bet tuo pačiu vis tiek kiti žmonės nori išgirsti, koks modelio tikslumas? Aš nežinau, koks modelio tikslumas, nes modelis gali būti, ką aš žinau? Labai tikslus atpažįstant nežinau, kaip čia pasakyti. Labai tiksliai gali atpažinti kokius 90% klasių, bet kitame 10% klasių jis gali būti 100% netikslus. Tai mes galim sakyt, kad modulis 90% tikslus, tai irgi toks. Nežinau kaip paaiškint, bet. Tas gali sukelti tokių objektyvumo problemų ir ypač jeigu duomenų rinkiniai nesubalansuoti ir panašiai.

I.: O su naudotojų grįžtamuoju ryšiu. Pagal tai testuodavot po to? AB testing ir pan.?

P.: Na, dažniausiai nelabai. Kažkaip nelabai prireikdavo iš mūsų pusės. Labiau darydavo tai žmonės, kurie arčiau produkto yra. Bent jau iš mūsų srities tas AB testingas neužima tiek daug darbo, nes mes nekuriame kažkokių user facing modelių. Labiau mes užsiimdavom ta gilesne logika šiek tiek. Tai dažniausiai to nereikdavo daryti. O feedback'as iš userių dažnai ateidavo iš iš produkto žmonių. Tarkim, modelis veikia per lėtai, per greitai ir tokius dalykus dažnai žinai iš anksto, tai jo kažkaip nebūdavo problemų daug.

I.: Turi omenyje, kad jau prieš gaunant grįžtamąjį ryšį jau žinojote, kad jis ir veiks lėčiau tas modelis? Pavyzdžiui, jei pasakė, kad per lėtai veikia.

P.: Tarkime jo, arba būdavo kažkokie produkto reikalavimai ir mes bandom, žinom, kad produktas turi būti toks, veikti greitai ir atrodyti taip. Ir dažniausiai bandome padaryti AI modulį tą, kad jis tiesiog tilptų į to produkto rėžius. Ir dažniausiai vartotojas tiesiogiai nebendrauja su mūsų produktu. Tarkime,

tai nėra kažkokia kaina, kurią jis mato website. Ar ten kažkoks produktų išdėstymas ar nuotraukų paskirstymas. Ne visada, bet dažniausiai to nebūdavo. Tai būdavo toks modelis, kuris optimizuoja kažką background'e arba atlieka kažkokį kažkokius sudėtingesnius veiksmus, kuriuos tarkim naudos kažkokia sistema ar gal darbuotojai ir pan., tarkim face recognition ar kažkas tokio. Tai. Dažniausiai iš vartotojo tas ryšys nelabai net ir gali praverstų sakyčiau, kai yra tai labiau jo, vidinė sistema.

I.: Kuriuose darbo procesuose matytum didžiausias rizikas algoritmo kokybės užtikrinimui?

P.: Tai turi omeny, kad kokiame kontekste modelio klaidos gali labiausiai pakenkti žmogui ar kaip turi omenyje?

I.: Nu programos galutinio rezultato kokybėje, o kokybėje, tame kontekste, kaip tu supranti, kas yra kokybė. Ir tame, kaip tu supranti, kas yra kokybiškas modelis.

P.: Ne, aš turiu labiau omenyje, kad jeigu modelis padaro klaidą, kokia bus kaina, tos klaidos ir tipo rizika tame? Ar rizika mano kaip developerio procese, kad jeigu aš neprižiūriu tos kokybės, kokia bus rizika abstrakčiam kontekste, aišku? Iš kurio pusės, čia turiu omenyje labiau?.

I.: Ne, čia labiau ne tai, kad kur? Ne, kas yra rizika, bet kurie tiesiog proceso development'o pasirinkimai, stadijos turi daugiausiai reikalauti, daugiausiai gal dėmesio, gali daugiausiai klaidų ten įsivelti ir panašiai?

P.: Nu jo, šiaip sakyčiau, gal pirmiausiai tiesiog tas pirminis ir toks reguliarus stage'as. Kur tu tiesiog kaip inžinierius bendrauji su produkto žmonėmis ar su verslo žmonėmis, ir tu turi pilnai suprasti, ko jie nori ir kad tu nepadarytum dirbtinio intelekto modelio, kuris tiesiog gal ne taip atitinka verslo use case. Pilnai suprast jų reikalavimus, ko reikia vartotojui ir kokie dalykai yra prasti, kad nebūtų taip, kad tiesiog, kad kuo labiau kuo daugiau įvykių tu užbėgtum už akių ir kad visi dalykai būtų sustyguoti tarp žmonių, kurie žymi duomenis, treniruoja modulius, naudoja, produktizuoja modelius ir pan., nes tame procese gali dažnai klaida įsivelti. Tiesiog elementari komunikacija ir bendravimas su stake holder'iais ta svarbi dalis.

Tada kitas tai jau man atrodo gal modelio testavimas. Man atrodo šitas etapas, kai gali pagaut daugiausia problemų, kur modelis, tarkim, biased yra ir pan. Tarkim, tavo metrikos rodo, kad tau modelis 90 % tikslus. Bet jisai tarkim, jei tavo modelis yra face recognition, tai tu matai, kad iš tų 10%, kur modelis klysta, 5% yra kažkokio tai kažkokios lyties ar ar rasės žmonės, ar kažkas tokio. Tokius dalykus turi tame etape sugaudyti. Tai toks gal labai paviršutiniškas testavimas. Gali, gali nepagauti tokių problemų kaip modelio bias'as ar pan., šališkumas kažkoks. Dėl to svarbu yra stebėt grynai konkrečiai pačiam akimis, kur klaidos vyksta ir panašiai. Ir bandyt gal išvesti kažkokias konkretesnes statistikos klases ir pan. Tai

čia gal kokie du svarbiausi dalykai. Kažkaip duomenų rinkime gali ir duomenų rinkime. Gali būti šiek tiek problemų, bet dažniausiai iki machine learning inžinieriaus mano nebūdavo tiek daug gal įtakos tame, bet turi rinkt duomenis, kurie diversifikuoti, diversifikuoti, abstraktūs, teisingai sužymėti. Nėra taip, kad įsivelia kažkokios sisteminės klaidos. Tame modulio treniravimui dažniausiai problemų nebūna.

I.: O tada, grįžtant prie tos rizikos, kur prieš tai kitaip norėjai atsakyti. Tai tada kas? Kas yra rizikos?

P.: Nu aš kaip įsivaizdavau, kad tu gal turi omeny, kokia rizika yra vartotojui, jeigu modelis padaro klaidą?

I.: Nu jo, sakykim šitas pavyzdžiui.

P.: Irgi priklauso tai jo. Apie ką norėtum kad. Turiu omeny, jo, kokie modeliai labiau ar kaip mes į tai žiūrime, kaip rinka su tuo deal'ina ir panašiai ar ką turi omenyje?

I.: Ne, tiesiog kaip naudotojas, su kokiom dažniausiai gali susidurt problemomis, rizikomis.

P.: Dabar galvoju man atrodo, kad jo dirbtinio intelekto sprendimai, kurie gal naujesni rinkai ir panašiai, gali daugiau sukelti rizikų, apie kurias net nepagalvota ar nepastebėtos. Nežinau, gal dabar tie įvairūs gen AI modeliai labiau linkę įvairių pridaryti visokių nesąmonių ir klaidų, kurios gali netyčia sukelti kažkokių blogų dalykų vartotojui. Bet senesnės sistemos labiau tokios subrendusios kaip tas pats face Recognition, tai jos vis dar daro klaidų, bet jų daro mažiau. Bet tos klaidos gali būti tokios, kad jo gal kažkokios rasės ar lyties žmonės, tarkim, ant jų ne taip gerai tas modelis veikia, kaip kad turėtų veikt, bet tokie dalykai jau senai gerai žinomi. Ir tas jau vis rečiau pasitaiko, tai jo Kažkaip. Tos klaidos jau šitose senesnėse atrodo labiau subrendusiose srityse labiau struktūrizuotos, suprastos, susistemintos. Tokios sistemiškos jau ir yra sprendimai tas spręsti. Bet kitose srityse naujesnėse dar tas nėra ir tiesiog toksai nuolatos besisukantis ciklas, kad išlindus naujam panaudojimo būdui, naujam naujų modelių tipui ir taip toliau, vartotojai gauna daug vertės, bet tuo pačiu ir tos rizikos atsiranda visiškai naujos ar nesuprastos ir su laiku sprendžiama. Tai nežinau.

I.: Tada kaip apibūdintum gerai sukurtą modelį?

P.: Čia jo, labai abstraktus klausimas, bet toks gerai. Turbūt sakyčiau, gal iš vienos pusės tai toks, kuris tiesiog veikia geriau už tavo praeitą modelį. Tiesiog žiūri į metrikas ir matai, kad jos aukščiau už tavo praeito modelio metrikas ir pasiekti tavo tikslai. Ir ką aš žinau. Žiūrint į duomenų rinkinį, testavimui skirtą, klaidos pasitaiko rečiau. Tiesiog tokiom vertinant objektyviom, ką aš žinau objektyviom metrikom ir kažkokiais loginiais reikalavimais, iškeltais verslo ar produkto, žmonių ar pan. Modelis turi padaryti tai ir jis tai padaro. O kitas ką aš žinau? Kitas atvejis galėtų būti, kad. Tiesiog tas nekelia kažkokių neatpažintų, neiškeltų, nesuprastų problemų, kaip vartotojas vartoja tą modulį, kai - ar naudotojas naudoja, nežinau kaip čia reik sakyti. Ir jis veikia production'e patikimai, vartotojas naudoja ir jis

laimingas. Ar tai tiesiogiai ar netiesiogiai, ir tas tiesiog sprendžia tą abstrakčią iškeltą problemą? Tiesiog modelis uždirba pinigus, sakykim taip. Nieko niekam nekenkdamas ir darydamas gera. Tai toks kažkaip sunkiau gal vertint šitą dalyką. Ir bet jo tiesiog. Kalbant apie kažkokį konkretų pritaikymo atvejį, lengviau išreikšti žodžiais, ką aš turiu omeny. Bet abstrakčiai kalbant, man atrodo, supranti, ką tai.

I.: Tai, jei turi kokį pavyzdį, gali duoti.

P.: Ką žinau. Tarkim, dabar galvoju. Koks nors, tarkim, modelis, kuris tavo veidą padaro juokingu, kai tu pasiimi telefoną ir nufilmuoji savo veidą su selfių kamera ir tarkim tavo app turi 10 milijonų vartotojų ir taip toliau ir tavo modelis tiesiog, jis tiesiog daro žmonių veidus juokingais, visada nuolatos yra patikimas ir žmonėms smagu jį naudoti, bet tuo pačiu jis sugeba aptarnauti dešimtis tūkstančių vartotojų vienu metu. Ir gal ne tiek pats modelis, bet visa modelio sistema production'e veikia patikimai, teisingai, taip toliau ir panašiai. Ji nenulūžta vidury proceso, paduoda gerą user experience ir pan.. Žmonės nuoširdžiai džiaugiasi, niekas neįsižeidžia ir taip toliau ir panašiai. Kažkas, ką sunkiau tiesiog sugaudyti kažkokiomis abstrakčiomis metrikomis ar, atsiprašau, konkrečiom metrikom. Bet tiesiog iš vartotojo pusės tai yra atliekama. Suteikia gerą produkto patirtį. Tiesiog taip.

I.: Kaip apibūdintum blogą modelį?

P.: Tai visiškai priešingai negu kaip apie gerą modelį. Tiesiog. Tai modelis, kuris, ką aš žinau, turi prastas metrikas, prastesnes negu tavo praeito modelio. Jis turi kažkokių daug pilnai nesuprastų, gal elgesio bruožų. Tarkim, gali kelti kažkokias konkrečias problemas, apie kurias mes nežinom, bet žinom, kad jos gali būti - jos neištirtos, nesuprastos. O jis, ką aš žinau, production veikia netolygiai, gali nulūžti ir user'iai tuo nepatenkinti, ir tas tiesiog nesprendžia jų problemos.

I.: Kai dirbai, kada galvodavai apie galutinį naudotoją?

P.: Labai priklausydavo nuo to, kiek produktas ar pats dirbtinio intelekto modelis būdavo arti to paties vartotojo. Tai kartais tarkim, jeigu tavo produktas yra skirtas filtruoti duomenis, kurie yra vidinėse įmonės duomenų bazėse ar kažkas tokio, tai tu per daug apie jį ir negalvoji. Tau rūpi tiesiog ta metrika ir tikslumas. Bet jeigu tavo produktas aktyviai vartojamas, aktyviai naudojamas, tai daug ir galvoji apie naudotoją. Viskas priklausys tiesiog, kiek arti naudotojų esi ir kiek į produktą orientuota komanda yra.

I.: Ir galėtum pateikti pavyzdį, kaip atrodo tavo mąstymas apie naudotoją?

P.: Mano mąstymas apie naudotoją būdavo gan. Priklausydavo nuo to, ar aš, ar darau kažką, ką. Ar mano asmeninė patirtis, ar darbo paskirtis yra grynai kontaktuoti su naudotoju ir užtikrinti, kad jam tas veikia tiesiogiai? Ar yra tarpininkas, kaip pavyzdžiui produkto žmogus, product owner'is, kuris už tai atsakingas, ir jis pateikia man svarbiausius dalykus, kad aš galėčiau užsiiminėti labiau development. Tai nuo to priklausydavo. Dažniausiai, jeigu produkto owner'is yra tam skirtas ar kažkokie user experience

žmonės ir pan., visokie research'eriai, tai stengdavausi priimti iš jų labiau supratimą apie naudotoją, o ne pats tiesiogiai jį suprasti. Ir dažniausiai jie užsiimdavo tuo, kad jie testuodavo, ar tas dalykas sprendžia vartotojo problemą, vartotojai patenkinti ir pan. Aš labiausiai stengdavausi fokusuotis ties tokiomis labiau lengviau suprantamomis, apčiuopiamom problemomis ir, aišku, bendradarbiaudavau su jais. Bet. Tas gal nebūdavo pagrindinis mano susifokusavimas, nes tiesiog tai būtų nelabai efektyvu.

I.: Žvelgiant į visą kūrimo, development'o procesą, kokie asmenys, asmenų grupė daro didžiausią įtaką galutiniam rezultatui? Nuo naudotojo iki tavęs.

P.: Irgi labai priklauso nuo produkto, ar jis labai į naudotoją orientuotas, ar jisai į, tarkim, vidinių sistemų efektyvumą, ar kažkas tokio. Tai čia sakyčiau pagrindinis toksai kaip. Viską sprendžiantis dalykas. Ir kuo labiau į žmogų orientuota, tuo labiau ir aš rūpinsiuos tuo. Toks negaliu tiesiogiai atsakyti, nes tiesiog tai koreliuoja, kinta tiesiog.

I.: Jo, bet čia labiau toks klausimas. Tiesiog nuo kokio asmens elgesio? Pavyzdžiui, sakykime, kuri į naudotoją sistemą orientuotą. Tai šituo atveju, nuo kokio žmogaus - tiesiog kaip jis elgiasi, kaip, koks yra?

P.: Turi omeny, į kokius dalykus kreipiu dėmesį ar kaip?

I.: Ne. Tai sakykim, nu vat yra skirtingi asmenys, tai naudotojas, produkto owner'is, tu, back end'ai visokie, frontend'ai, dar kas nors. Ir tai yra sistema, nukreipta į naudotoją. Ir šiuo atveju galutinis produkto rezultatas - ką jis rekomenduos - nuo kurio iš šitų veikėjų tiesiog egzistavimo labiausiai priklauso?

P.: Visų pirma nuo vartotojo. Antra, tai man atrodo, tų žmonių, kurie arti to vartotojo naudotojo būna ir sprendžia tiesiog, kas jiems patinka, kas ne ir gal žiūri į tai iš tokio abstraktaus makro konteksto. Tai gal įvairūs produkto žmonės, duomenų analitikai, tie patys frontendo žmonės ar žmonės, kurie supranta ne tik kaip vartotojas jaučiasi, bet kaip jaučiasi daug vartotojų ir gali tai gal apibūdinti skaičiais statistiškai, o ne tik konkrečiai.

I.: Kur matytum, kuriuose development'o žingsniuose matytum daugiausiai etikos problemų? Galimų potencialiai.

P.: Galbūt duomenų rinkimas, sakyčiau, ypatingai su duomenų apsauga ir GDPR ir pan. Nes kiekvienas duomenų vienetas turi kažkokią savo ten teisę ar prieigą ar pan. ir į kiekvieną iš jų reikia atsižvelgti. Pavyzdžiui, jei ten apmokai modelį su duomenimis, kurie priklauso GDPR'ui ar pan. Tu gali su tokiais duomenimis apmokyti ar kažkas tokio. Tai tarkim, jeigu po metų tas GDPR nebeleidžia tų duomenų naudoti, tau po to reikia pertreniruot modelį ir pan. Tai aš įsivaizduoju ir nenustebčiau, kad kai kurie tiesiog žmonės nesirūpina tuo. Bet nežinau, priklauso viskas nuo kokias ten, kokius leidimus ir panašiai turi įmonė.

Kitas gal modelių testavimas. Ne tiek, kad jis pats galėtų etikos problemas sukelt, bet jo trūkumas gali sukelti etikos problemas. Ir jo, mes gal galime matyti, kad modelis veikia gerai skaičiais, bet labiau nesigilindami praleisti kažkokias konkretesnes problemas, kažkokią diskriminaciją ar panašiai. Ir diskriminacija neturiu omeny, kad čia tiesiog diskriminuoja žmones, bet gal tam tinkamus žmonių pattern'us ir panašiai, kaip jie elgiasi vienaip ar kitaip. Ar modelis labiau linkęs kažkaip, nesugeba kažkokių duomenų taip gerai klasifikuoti ir įvertinti kaip kitų. Ir tiesiog tai galima praleisti žiūrint į abstrakčius skaičius. Ir vartotojai ar kitokie subjektai su tomis savybėmis gali būt, tiesiog gauti prastesnį produktą negu kiti. Sakyčiau, tas jo, man atrodo, šitos dvi svarbiausios.

I.: Ar koreliacija nelygu priežastingumui?

P.: Tu teisus. Taip, tai nelygu. Nu jo abstraktu, bet tiesiog gal aš net neinu iš tokio duomenų, mokslo ar analitiko background'o, Bet jo, man atrodo visur tiesiog reik elementaraus judgemen'o, o ne tik į abstrakčias metrikas žiūrėt. Ir iš to ir nusprendi, kas, kas koreliacija, o kas priežastingumas.

Informantas Nr. 4

Amžius: 26 m.

Lytis: vyras

Darbo vieta: šiuo metu dirba duomenų mokslininku („data scientist) lietuviškoje įmonėje, prieš tai 3 m. ML inžinieriaus patirties

Gyvenamoji šalis: Lietuva

Pilietybė: lietuvis

Formatas: nuotolinis pokalbis

Trukmė: 1h 18min 48s

Data: 2024 m. spalio 14 d.

I.: Esu Simonas Bansevičius, politikos ir medijų magistro studentas. Darau tyrimą apie mašininio mokymosi programų etiką. Informuoju, jog bet kada gali nuspręsti nutraukti pokalbį. Ar sutinki, kad šis pokalbis būtų įrašytas ir jo medžiaga būtų naudojama tyrimo tikslais?

P.: Sutinku.

I.: Puiku! Pradėsiu nuo prašymo prisistatyti kiek gali, pasakant savo amžių, ką dirbi šiuo metu, kiek laiko dirbi šioje srityje?

P.: Jo tai esu (anonimizuota). 26-erių metų. Šiuo metu dirbu data scientist'u. Jo. Paskutinis gal grynai oficiali pozicija dabar yra data scientist'as, prieš paskutinius kokius du mėnesius prieš tai, apie 3 metus dirbau labiau tokiu mašininio mokymosi inžinieriumi. Daugiau prie prie inžinerijos sakykim, negu gal šiuo metu, bet ir šiuo metu nemažai tenka irgi kažkiek prie to prisiliesti.

I.: Ar gali papasakoti apie vietą, kurioje dirbi? Ar tai lietuviška, ar užsienio kompanija, kiek darbuotojų, kokia komanda, kokio dydžio, kokia struktūra?

P.: Tai šiuo metu dirbu lietuviškoje įmonėje. Na šiaip tarptautinė įmonė turbūt, bet pagrinde lietuviškoje. Tai mūsų komanda šiuo momentu iš tikrųjų yra sudaryta tik iš lietuvių, bet greitu metu turėsime ir kolegų iš kitų šalių. O jo bendrai tai. Jo nežinau, dabar turbūt vis tiek koks 70 proc. iš tikrųjų yra lietuvių. Pas mus šiuo momentu.

I.: Gerai. Tu gali tiesiog papasakot apie savo darbo kasdienybę. Tiek dabar, kaip data science, tiek prieš tai, kaip ML inžinieriaus.

P.: Tai jo nežinau. Šiaip labai skirtinga būna kasdienybė kiekvieną dieną gan skirtinga nu jo, bet turbūt standartas yra visur šiaip dirbt sprintais Scrum'u dirbt. Tai nežinau. Prieš tai dirbau tokioj labiau tarpdisciplininėj komandoj. Ta prasme su su frontend'u, su backend'u, kartu žmonėm. Šiuo momentu esam tokie grynai data science komanda, kur komandos visi nariai yra data scientist'ai ir tada labiau kažko prireiks bendraujant su backend'o ar data inžinieriais ar kažkuo kituo. Ir nežinau, kasdienybė tai tiesiog turbūt, kurtis kas kažkiek laiko kažkokiu, išlendant kažkokioms idėjom. Ta prasme, kur kūries taskus. O kai nesikuri task'ų, tai daugiau dirbi prie tų task'ų, tai ta prasme čia ir. Ta prasme reikia ir visokių modeliavimo, testavimo, eksperimentavimo su modeliais, dokumentavimo, tų modeliavimo būdų. Toliau kūrimas, braižymas schemų, kaip tie planai judės, vis visos tų duomenų iš kažkokios vienos vietos į kitą ir kaip tas modelis galės juos apdoroti ir paskui sudėlioti kažkur į kitas vietas. Tai tada čia kažkiek įsijungia dar ir data engineering'o, nes irgi reikia kažkur saugoti kažkokius tarpinius rezultatus ar dar kažką tai. Jo, tai irgi turbūt čia kažkiek yra laisvės kiek, kiek tu nori skirti laiko arba kiek turi laiko skirti sau prie to kažkokio data engineering'o. Ar tu nori pats ten kurtis visas lentas visokias, ar nori perduoti tą kitai komandai data engineering'o būtent? Gal būtent aišku, tada tau tiesiog sukuria daug daugiau laiko prie kitų dalykų gali dirbt, bet iš kitos pusės reikia laukti, kol kažkas kitas apsiims tą task'ą, kur galbūt jiems ten nėra to prioritetas kaip tau tai. Tai tiesiog kartais būna, kad gali reikėti pačiam daryt. Kartais atrodo, kad tai geriau pačiam daryti, kartais kažkam perduoti. Turbūt panašiai būna ir su deployment'ų modeliu visokių, kad kažkiek pats darai, kažkiek yra pas mus ML ops darbuotojų, kurie tą infrastruktūrą sudėlioja, kad adaptuojant tuos modelius, tai tiesiog kartais būna, kad žinai tu imi ir pats bandai sudėlioti visokius CI/CD pipeline'us arba arba. Jo kažkokius Docker failus ar kažką tokio, kur kur

padeploy'ins tavo tuos modelius. Kartais tai sakau gal perduoti tiems labiau tiems ML ops komandom kartais tiesiog labiau konsultuojies su jais, jei kažkiek turi daugiau laiko prie pats prisėst arba labiau nori tą tą daryti, tai tada daugiau labiau konsultuojiesi su jais. Jeigu kažkokių turi kitų darbų labiau svarbesnių, tai tada arba tiesiog nenori, tai tada labiau perduoti jiems tą dalį padaryti ir atsidedi ten labiau prie to modeliavimo, kažkur kitur ir panašiai. Nežinau. Šiaip nemažai laiko yra ir vis tiek su visokiais stakeholder'iais, bendravimu. Ta prasme išsiaiškinti, ką reikia padaryti, ko nereikia padaryti, eina visokie request'ai žinai, kur kartais tiesiog yra, per kažkiek pabendravus išsiaiškinti, kad ten iš tikrųjų nereikalingas dalykas, gal ten kažkas ne taip suprasta.

Kartais išsiaiškinti, kad iš tikrųjų reikalingas dalykas. Ir tiesiog vat sakau, planuojies tada jau kaip tą įgyvendinsi, užsidedi kažkokius task'us ar OKR'us. Tai vat tas pats ir su tais OKR kažkiek planavimas irgi yra ta prasme. Visa komanda kažkiek prisidedame prie prie tų OKR'ų, bent jau aprašymo gero ir ir planavimo kaip įgyvendinsim juos. Tas pats su sprintais. Ta prasme, tokiam granular'esniame lygmeny tai, tada juos apie tuos sprintus, ta prasme labiau giliniesi ir aiškinies dalykus, ką reikia padaryti, ir kiekvienas komandos narys prie to sėdi ir žiūri, kaip ten kažką įgyvendinti, aprašyti ir suplanuoti. Jo nežinau kažkokia vis tiek turbūt nedidelė darbo dalis aišku yra irgi su ir su kitų komandų nariais kažkokių tokių paralelinių beveik komandų nariais bendrauti, pasisemti žinių, pasidalint žiniomis su jais. Tai vat, su data egnineering pakalbėti jiems, kad jie ką nors pasidalina kokiais nors savo žiniomis, mes pasidalinam savo žiniomis. Ten yra kitas data science komandos, kuri labiau dirba specializuotai kažkokiam specializuotam produkte ar kažkas tokio, kurie pasidalina arba iš kitų ten. Įmonių, su kuriomis bendradarbiaujame. Kartais pasidaliname dakau žiniom, tai vat kažkiek yra ir tos dalies, kad žiniom dalintis ir tuo pačiu, nežinau, pasiklausyti iš kitų, kuo pasidalinti. Tai čia čia tokia nedidelė dalis, bet vis tiek kažkiek kažkiek būna šito. Jo, tai čia turbūt pagrindiniai dalykai. Nežinau, gal kažkur gilintis, bet jei labiau tai.

I.: Gerai, tai tada, pavyzdžiui. Tau reikia vis tiek duomenis gauti savo darbui. Iš kur gauni duomenis?

P.: Tai čia truputį priklauso.

I.: Ir čia tebūnie atsakymas tiek iš to, tiek data science, tiek engineering'o patirties. Tiesiog ir gali pasakyti, čia labiau atsakai iš kurios perspektyvos.

P.: Jo, tai čia sakau irgi. Kadangi bendrai taip šnekant iš mano patirties visos gal nebūtinai čia kažkur šitoj įmonėj. Aš ta prasme viską labiau gal taip bandau sakyti ne kad kur dabar dirbu, bet bendrai tiesiog iš mano patirties tai sakau, tai vat jo data egnineering'e, tai tu vis tiek renkiesi iš visiškai visur. Tai tu bandai surinkti data iš visų įmanomų šaltinių, kur manai, kad kitiems žmonėms reikės tos datos?

Tai dažniausiai vis tiek tas prasideda nuo kažkokių vidinių duombazių, tai yra ta prasme operacinių duombazių, kur yra nu ta prasme grynai SQL kažkokios duombazės ar ten NoSQL, žinai ten Mongo, dokumentų duombazės ir pan. Tai iš šitų vietų pirmiausia turbūt renkies duomenis. Aišku, yra svarbiausi duomenys būtent tau su tavimi susiję, su tavo įmone, tai tu pradedi nuo šitų dalių ir tada tik tais aišku žiūri ir, nu ir jo bandai patalpinti kažkokiam data lake'e, data warehouse'e ar pan. Tai. Tai tik tais jo. Tai čia tik tais tada priklauso kokius ten duomenis laikai ir ko tau reikia. Tai aišku vis tiek toje operacinėje duomenų bazėje saugumas yra didesnis, prieinamumas yra mažesnis prie tų duomenų. Tai tu aišku ten. Laikai nu visus duomenis, bet vat kai keli viską ten į kažkokį data warehouse'ą ar data lake'ą tai tu ten dalį tų duomenų negali laikyti. Tada bandai šalinti kažką, ko negali laikyti data lake'e. Tai tu jo arba kažkaip bandai apibendrinti.

Pavyzdžiui, suagreguoti. Aišku. Šiaip dažniausiai šito bandai išvengti agreguoti, tai tam pirmame step'e, kai iš duombazės keli į data lake'ą, tu bandai visus duomenis sukelt, bet tai ta prasme neagreguotus duomenis įkelt. Bet pašalinus tuos dalykus, kurių, sakykim negali laikyti ar kažkokius identifikacijos dalykus, ten adresus iš esmės tiesiog privačius sensitive duomenis, tai šituos kažkaip bandai pašalinti, bet išlaikyti kažkokią struktūrą, kad vis tiek kažkaip skirtingas "lentas" galėtum sujoin'int per kažkokius ID, bet bet nebūtinai, kad galėtum žinoti, kas ten per žmogus. Čia vėlgi turbūt skirtingos įmonės priklauso, kokius duomenis apskritai kai kur nelabai ir tų sensitive duomenų apskritai ten yra, bet jo tai čia pradedi turbūt nuo to, iš kur duomenis pasiimsi, iš data engineering'o pusės. Tada kitas dalykas yra dažnai ir kažkokių open source'inių vietų ieškai. Nežinau, gali API kažkokių naudoti papildomų, kurių, pavyzdžiui, ten tau nereikia toje operacinėje duomenų bazės SQL, ar kažkur tai, bet kažkokioj analizėj tau ar ar sakau ten modeliams tau reikia kažkokių papildomų dalykų. Kažkaip tai nežinau sumap'int kažkokius tai skirtingus nežinau duomenis. Arba tiesiog pasiimti sakau kažko, ko nereikia toj operacinėj duombazėj, tai tu kažkokius API naudoji iš esmės kaip data engineeringe, kad tu irgi renkies duomenis ir paskui gali join'int su tais savo vidiniais duomenimis, kad kažkokias nežinau papildomas. Papildomą analizę galėtum daryti tai.

Tai kartais yra šitas vat sakau kažkokiem modeliams, pavyzdžiui, modeliavimui gali būti naudingas. Tada... Jo labiau iš to data science modeliavimo pusės. Tai tada tu esi labiau vartotojas to, ką ten data engineering'as daro ar analitikas, engineeringas ten irgi čia priklauso, kokio dydžio įmonė ar turi išsiskirsčius į tas sritis, ar tiesiog yra tik data engineering'as, tai tu tada kaip data scientist labiau esi vartotojas tos dalies, tai tai jo, tai tada tu ir naudoji tas tas visokias analitines, ten data lake'us, data warehouse'us, kuriuos kuria tie, data inžinieriai, tai tai jo, tada naudoji tą datą ir yra kartais galbūt kažkokių tai modelių, kurie vat gali ant tų generic dalių neveikti. Ta prasme ant generic duomenų. Tai

tada irgi reikia ta prasme modeliuoti ant tų sensitive duomenų kažką tai. Tai tada tu jo. Aišku, data lake to negali laikyti. Ta prasme, tai irgi tada pats modeliavimas yra žymiai sudėtingesnis, nes tu visiškai kitaip turi dirbti, ta prasme, nes su lokaliu kompu negali ten daryti, turi per kažkokius rėmus dirbt ir ten nesaugoti tarpinių rezultatų, žinai viską laikyti tik atmintyje, run'inant kažkokį pipeline'ą, nes paskui produkcijai tas pats ta prasme veiks. Tai nu tu labiau testuoji iš karto ir turi dirbti lyg produkciniam tokiam dalyke tai yra žymiai sudėtingiau modeliuoti ir dirbti su tokiais feature'ais, sakykim kur reikės procesint tuos sensitive duomenis.

Jo, tada tiesiog yra žymiai daugiau restriction'ų ir tu turi kitaip biški dirbti, tai tada tavo tas svoris gali būti tiesiai į tą operacinę duombazę, bet tada aišku, kad darbo specifika sudėtingėja. Tai jo, tai dažniausiai stengiesi išvengti šitos dalies, kad galėtum tiesiog su data lake'o duomenim dirbti ir jo, ir kažkaip tai viską padaryti. Nu, bent jau sakau bent jau tą modelį pasikurt, kažkaip ne ne nenaudojant tų private, tų sensitive duomenų.

I.: O pavyzdžiui, tenka kažkiek galvot, kaip daryti interfeisą naudotojo, kad pavyktų gauti duomenis? T.y. UI?

P.: Tai jo, būna tokių dalykų, tai čia ypatingai gal sakykim. Nu, ta prasme, toks atvejis sakyčiau, gal populiariau, populiariausias gali būti tokiu atveju, jeigu tu turi kažkokį human in the loop. Tokį treniravimo būdą, kur kažkas turi review'int, ką tas modelis pavyzdžiui daro. Tai vat tokiu atveju tiesiog tu kažkaip vis tiek, turi review'int ką tas modelis ten daro ir kažkiek irgi su tais duomenim turi žiūrėti, kas buvo paduota, kas kas rašyta, tai jo tu iš esmės turi kurti vos ne tokį vidinį interfeisą kaip tiesiog admino portalas. Sakykim, kažkas tokio kaip tu, kaip įmonei vis tiek negali, tam vat sakau data lake'e nieko tokio negali laikyt kur visi prieina, bet turi tokį labiau kažkokiu būdu vis tiek atvaizduoti tą visą specifiką. Tai jo, tada gali reikėti kurti kažkokį interfeisą, kad kuris ten kažką iš tos duombazės, sakykim vis tiek, žiūri ir. Ir tu gali sakau pavaliduot, kaip ten viskas atrodo. Tai nu tai jo, čia jau data science, tiksliai to šiaip ne ne ne nereikėtų turbūt man daryt, bet tiesiog tai yra susiję ta prasme su modeliavimu, pavyzdžiui ir ką reikia padaryti kažkokiems gal. Ar tau kaip, ta prasme. Tu esi tas vartotojas arba kažkam kitam. Bet tai nėra turbūt ką data science patys daro kažkokį front end'ą.

I.: Bet tau kaip vat inžinieriui, šitam nereikėjo galvot, pavyzdžiui, tiesiog user interfeisą, net ne tiek šito modelio kaip tu sakai, bet tiesiog paprastą naudotojo sąsajos sistemą. Ir, nu pavyzdžiui, Netflix'as koks nors kaip yra, kad tai dabar sekam, kiek laiko žiūri vaizdo įrašą, ar kiek laiko scrollin'a kažką. Tai vat tokių?

P.: Ne tu turi omeny, kad gal bendradarbiauti su front end'o team'u, kad paskakyti jiems, kas ką track'int, pavyzdžiui? Tai kažkiek jo reikia apie tą bendrauti, bet. Bet turbūt. Nu čia priklauso labai koks

field'as. Mūsų field'as yra toks žinai kur, šiaip mes labai stengiamės išvis nieko netrack'int beveik apie savo vartotojus, tai čia biški tas yra, bet taip bendrai tai irgi ta prasme gal iš pažįstamų irgi patirties ką žinau, tai kad jo jo, tai aišku čia yra tas ir tada nu čia tokie gaunasi, kad kelių dalių diskusija. Tai yra frontend'ui žinai vat, visokių data scientist'ų, kurie nori kurti modelį. Tada yra frontend'ų, kuris kažką track'int žinai. Ir tada, bet tai irgi ten nebūtinai ten frontend'as, gali ir backend'o kai kuriuos dalykus sekti. Ir tada yra žinai kažkoks legal ar kažkas tokio, kuris irgi tada sako žinai, ką gali sekti, ko negali sekti ir panašiai. Tai tada žinai yra diskusija. Tiesiog, kad atrasti tą kompromisą, kur tu galėtum kažką naudingo sukurti, bet tuo pačiu ir iš, sakykime, iš tų duomenų, kuriuos realiai gali naudoti iš tikrųjų. Tai jo, tai dažniausiai čia yra tokia kelių šalių diskusija, kur žinai į skirtingas puses eina diskusija. Bet. Tai jo, tai šiaip yra dažnas dalykas, ypač pas mus, kad tų duomenų yra tokių sensitive. Realiai esi labai ribotas, kad negali beveik nieko sensitive turėti ir iš tokių gan bendrinių žinai kažkokių duomenų turi maždaug kur tas feature'as, kurie būtų naudingi. Tai vat čia yra toks didelis gan challenge'as, kad bet tu neturi prieigos prie tų sensitive duomenų. Tai tu jeigu kažkokį sakau kad, nori feature sukurt, tai iškart labai sudėtingesnis darbas tampa. Ir aišku, tie modeliai nėra tokie tikslūs, bet nu bet sakau, kad būtent ten mūsų fielde tiesiog taip jau yra, kad. Žinai, ta prasme negalima naudoti sensitive duomenų, tai kažkokius bandome per tokius labai bendrinius kažkokius dalykus kažką pasakyti.

I.: O kodėl būtent jūsų komanda praktiškai nerenka duomenų?

P.: Nu mūsų pavyzdžiui būtent mūsų produktas. Kur aš dabar dirbu, tai iš viso yra būtent trynimui marketingo, visokių duomenų ir apskritai tada brokerių duomenų ir visko ir pan. trynimui iš iš "intiko", iš data brokerių. Ta prasme mes bandome pašalinti kiek įmanoma marketingo duomenis iš interneto apie žmones. Tai vat ta prasme mūsų tikslas toks yra. Tai mes ir patys irgi tuo rūpinamės, kad mes patys neturėtume per daug tų duomenų ir tiesiog toks žinai, visa idėja yra produkto. Jo, tai sakau, tai turbūt irgi ir sakau tada irgi priklauso turbūt koks tas produktas, nežinai kitur tu gali irgi nebūtinai, kaip data science nebūtinai išvis su front end'u kažką susijusio kurį, tai žinai. Pavyzdžiui, iš mano patirties tai yra koks nors API kūrimas. Tai tu irgi tada iš viso apie frontend'ą realiai nesirūpini ir tu labiau, kuri feature'sus žinai iš visiškai kitų duomenų, o ne userio kažkokių interaction'o. O visiškai žinai kažkokių duomenų, kurie tiesiog yra nesusiję su tavo produktu. Tiesiog bendrai kažkokie yra duomenys ir tu bandai kažkokį naudingą dalyką sukurti useriai iš tų duomenų, kurie nėra ten. Tu sakau frontend'o gali duomenų visiškai jų neturėti. Čia sakau kaip pavyzdys ten toj energetikos visoj srityje žinai, tiesiog iš energetikos kažkokių duomenų tu bandai kažką sukurti naudingo useriui. Ir tu ten front end'o gali net neturėti jokių duomenų. Tu tiesiog žiūri į kažkokį suvartojamą elektros ar kažką kitokio.

I.: O kaip vyksta tas kažko naudingo naudotojui ieškojimas? Kaip vat nustatai, kas gali būti naudinga?

P.: Nu šiaip dažniausiai tai sakau tai šituo atveju, jeigu tu kažkokį API žinai kuri, tai jei tu turi kažkokiam business to business didesniems klientams, tai dažniausiai iš klientų ateina ir žinai. Čia labiau toks product ownerio yra užsiėmimas žinai, susisteminti. Žinai, ko klientai nori ta prasme, bet čia būtent business to business. Tuo metu business to consumers panašiai būtų. Tu žinai, kad tu irgi turi daug klientų, kurie tik tai žinai, sakau biznis to consumers, turi ten jau tūkstančiais ir ten šimtais tūkstančių klientų, kurie kažko tai prašinėja. Tai ten gal truputį sudėtingiau susisteminti, ką tu tiksliai, ką jie, ko jie tiksliai prašinėja. Tai vat, pvz. čia irgi gali įsilieti žinai dalykas, kad gali tą patį mašininį mokymąsi naudot, kad jeigu tu turi free forumą atsiliepimus apie savo produktą, žinai, tu gali juos sisteminti pasinaudodamas kažkokia ML, kad maždaug kategoriją. Ką čia tas žmogus bando parašyti. Nes jeigu tu ten gauni dešimtis tūkstančių atsiliepimų apie savo produktą nuolat, tai ten kažkoks žmogus ten nelabai pereis visko ir neperžiūrėjęs tu gali sisteminti per šitą ir tada žiūrėt, ko tau reikia kaip produktui, ko daugiausiai žmonės prašinėja ir tai jo, tai labiau iš to feedback gali tada žiūrėt. Tada jeigu labiau toks business to consumer, tai žinai, paprasčiau klausinėji. Šito product owneris, nes kažkoks su klientais dideliais klientais šneka ir mato, kad keli klientai to paties maždaug nori. Mes tada irgi įvertiname, ar čia yra įmanoma, ką ką parašo. Žinai, ar tai irgi mums patiems atrodo naudinga? Kartais, nes kartais žinai, irgi gali būti tas, kad tai vienam klientui atrodo naudinga. Mums bendrai atrodo, kad biški švaistymas laiko bus, nes niekas kitas to neprašo. Žinai, tai tada gal nebūtinai darysi tai, bet labiausiai tai turbūt drive'ina, sakyčiau, iš mano patirties tokie dalykai būna iš kliento pusės, o ne tiesiog iš, data scientst'ai galvoja čia jau fun būtų padaryti kažką žinai ir darai. Dažniausiai kažko nori klientai ir tu bandai išspręsti.

I.: O čia kur minėjai tą pavyzdį? Kur buvo tas free formos atsiliepimų sisteminimas? Tai čia tavo buvo darytas projektas?

P.: Neišgirdau pabaigos klausimo.

I.: Ar tu tokį projektą esi daręs, ar čia hipotetiškai buvo?

P.: Ne, aš pats nesu daręs, bet prasme esu dirbęs, kur turėjom, sakykim, tokį dalyką, tai tai čia.

I.: Kažką tokio panašaus tu pats darei kada nors?

P.: Na, galvoju dabar nu jo.

I.: Kaip kad žinai, kai iš tokių abstrakčių duomenų gauti?

P.: Ne struktūrizuoto tipo.

I.: Jo jo, iš nestruktūrizuoto padarai struktūrą.

P.: Jo jo, nu tai jo, tai šitas jo kažkiek esu, bet nėra turbūt pagrindinis dalykas ties daugiausiai kuo esu dirbęs. Jo tikrai yra tekę. Tai jo jo, yra su NLP ir panašiai. Kažką daryti tai. Klasifikuoti. Nu tai va, čia klasifikavimas ir buvo, kad klasifikuoti kažkokį tekstą. Turbūt mažai bėra jau. Šiais laikais, apskritai įmonių, kurios neturėtų kažko žinai su klasifikavimo. Ir kažkokį žinai, ten transformerių NLP klasifikavimo kažko padarę. Kad žinai, kaip sakau, vat būtent. Nes žinai, labai dažnai yra lengva.. Sudėtinga, labai gauti tikslius duomenis. Dažniausiai yra lengviau susirinkti kažkokius truputį tokius vague žinai, tekstinius, free formos duomenis. Net nebūtinai susirinkt. Žinai, tai tiesiog gali būti tiesiog kažkoks ypač didesnis enterprise'as, ar kažkas tokio, tiesiog jas turi net prieš tai, gal net negalvodamas, kad reikės čia kažką naudoti, kažkiai, modelius ant jų. Tiesiog kažkada išauga tokie kiekiai. To to tokių duomenų ne struktūrizuotų, kad tiesiog kažkada reikia pradėti kažką daryti automatiškai, o ne vien tik manual žiūrėjimui. Tai tai jo, sakyčiau, labai daug. Kažkas vis tiek yra daroma su tokiu tuo NLP ir pasirenkama taktika.

I.: Tai vat reiktų klasifikuoti tau tai. Kaip nusprendi, kaip klasifikuoti? Tavo tikslai kokie yra?

P.: Jo. Čia dažniausiai labai priklauso, kokia užduotis yra. Tu beveik visada žinai gan labai tiesiogiai, beveik žinai turi bendrauti su kažkokiais stake holderiais, kurie daro sprendimus žinai, iš tų atsiliepimų. Tai vat saku, iš to teksto sakykim. Žinai, vat sakau, jeigu pavyzdys yra, kas duoda vat žinai kažkokius atsiliepimus, tai vis tiek kažkas turbūt daro sprendimus pagal tuos atsiliepimus. Tai tu bendrauji su tais, kas daro tuos sprendimus apie tuos atsiliepimus ir maždaug pasakai, kad iš tavo patirties, pavyzdžiui, kokios dažniausiai žinai kategorijos yra, tiesiog pirmiausia apsibrėžti kategorijas, kokios žinai gali būti? Pats pasižiūri į tuos duomenis ar ar čia atrodo šitos kategorijos pacover'ina beveik viską žinai ar ne ir ar kažko trūksta, ar kažkas gal nėra pakankamai aiški atskirtis tarp kažkokių kategorijų. Tai jo bendrauji su tais, kurie paskui naudos, tą, tą informaciją. Nes žinai, dažniausiai būna taip, kad kažkas vat žiūri į tuos. Tą tekstą, kažkokį laikotarpį iki kol tas kiekis būna per didelis, kad tas žmogus galėtų žiūrėti. Jis turi kažkokį neblogą įsivaizdavimą, kas ten standartas yra, kokie ten tai dažniausiai kategorijos. Ir tu tada sakau, išsiaiškini iš to žmogaus, kokios tos kategorijos gali būti. Paskui pasivaliduoji, ir tada žiūri ir tuos rezultatus, ar ar ar gerai susidėlioja viskas. Jeigu ten paskui kažkiek gali tekti koreguoti, pagal irgi pagal tuos rezultatus, kaip gerai pavyksta išvis ten viską sudėlioti.

I.: O kaip apibrėžtum kada yra grėsmė blogai suklasifikuoti.

P.: Jo. Tai čia turbūt jo sudėtingiausia dalis yra būtent tai, kad. Ten gan sudėtinga. Be kažkokio stebėjimo, labai sudėtinga iš tikrųjų yra pagauti kažkokias vietas tai. Tai nežinau, mano patirtimi tai realiai visur, kur yra kažkokie panašūs dalykai naudojami kažkur, vis tiek yra žmogus, kuris žiūri į šituos dalykus ir kažkiek žinai review'ina pats. Ar čia viskas atrodo logiška, žinai? Tai tu iš esmės tu dažniausiai

nebūtinai visiškai pakeisi to žmogaus darbą, bet labiau sumažinti 10 kartų krūvį į mažesnį. Ir nu vis tiek žinai, kad ir ne viską review'indamas žinai, bet kad ir ten kelis tuos pavyzdžius pasižiūrėdamas gali žinai pamatyti kažką ne to ir bandyt inai pagerinti tuos rezultatus, pavyzdžiui. Jo. O šiaip tai vis tiek kažkokias monitoringą darai ir tokį automatinį kažkiek, bet šitas yra sudėtinga gan dalis, nes sudėtinga išvis metrikas, tai šitą visą bandai pasidaryt, Bet. Visur labai specifinis dalykus reikia stebėti. Ir čia vėl dažnai gali išlįsti tas duomenų privatumo klausimas. Kad tu irgi žinai, tu negali ten review'int dažniausiai duomenų grynai tokių, kokių yra, kažkokių turi paprocessint juos ten ištrinti kažkokius. Ten yra visokių tools'ų šiaip kurie žinai, irgi ištrina tas privačias dalis duomenų, Facebook'o yra vienas iš populiariausių atrodo. Kur pašalina visą tą sensitive informaciją iš kažkokio teksto, tai žinai tai. Tai tas dar irgi pasunkina. Paskui monitoringus žinai, nes nu tiesiog su. Gali žinai, vis tiek dalį tos informacijos ištrina ir tada irgi tenka iš to likusio konteksto kažkaip ten žiūrėti kas čia ar čia viskas ok ar ne. Ne viską iš esmės matai.

I.: Papasakok, kaip tu jį vertini, kaip įvertinate galiausiai tą galutinį rezultatą? To, ką sukūrėte.

P.: Tai nežinau ar čia irgi šiaip labai sudėtinga. Gal vieną vienaip žinai atsakyt, bet nes labai skirtingi būna ir modeliai, ir metrikų ką tu ten seki. Ir tas kiekvienas tas feature'as gali žinai labai turėti skirtingus dalykus, bet šiaip dažniausiai vis tiek kažkokį stengiesi turėti iš anksto treshold'ą, žinai, kad kas tau atrodo pakankamai gerai, kad ką reikia pasiekti, kad būtų čia naudingas šitas dalykas tai dažniausiai kažkokia metrika, bandoma nustatyt iš anksto, bet dažniausiai, jeigu tai yra visiškai nauja kažkas, kur nei tu, nei įmonė kažkokia žinai, nėra turėjus patirties tokios, tokį dalyką sprendžiant, tai gali labai ten prašauti su tuo savo tikslu, kiek ką pasieksi. Bet dažniausiai yra siekiamybė turėti iš anksto kažkokį tą, nes. Tiksliau skirtingai negu su visu kitu software inžinieringu, sakyčiau data sciencee modeliavimo ir pan. Tai tu čia gali. Tiesiog nuolat dirbti prie kažkokio to dalyko ir jis nuolat greičiausiai gerės, kažkiek kažkokiu. Bet kažkokiu metu žinai, yra tas tiesiog diminishing returns žinai atsiranda ir vis tiek. Tuos pirmus kažkokius tuos low hanging fruits susirenki žinai, paskui tu vis dar gali ten kažką tobulinti, bet labai ilgai užtrunka. Dažniausiai žymiai sudėtingiau yra vėliau kažką tobulinti, negu kad tas pridėtiniai rezultatai. Tai. Tai jo dažniausiai sakyčiau, kad. Viena gal gera praktika yra, ką irgi sakyčiau, nemažai gal žmonių daro, tai yra pirmiausia susikurti kažkokį žiauriai paprastą modelį tos problemos, kurią sprendi. Ir pažiūrėti, kokie rezultatai gaunasi. To žiauriai paprasto modelio ta prasme, pačiu standartiškiausiu žinai kur nieko beveik nedarai to papildomo, bet visiškai tas toks. Lengviausias variantas išspręsti tą problemą tai visur. Kažkiek bent kažkiek padės, nes žmonės žino kažkokį standartinį modelį, kuris šitai problemai taikomas.

Tai, kai pasižiūri ir tada turi kažkokį žinai tą tokį baseline'ą, nuo kurio žiūrėsi. Bandysi žinai gerinti. Žiūrėsi, kiek smarkiai gali pagerinti tuos rezultatus. Tai dažniausiai pradedi nuo to ir tada galvoji, kiek tu čia realistiškai gali pagerinti. Nes kartais gali būti, kad tas paprasčiausias sprendimas yra pakankamai geras. Ir tada žinai galvoji, ar čia man verta dar, žinai laiką leisti prie šito dalyko, jeigu čia pakankamai gerai veikia daug neįdėjus darbo ir ypač negavus kažkokio feedback iš kažko kito, ar čia, ar čia gerai, ar ne, tai tada bandai gerinti. Ir jo dirbant sprintais dažniausiai tai tu irgi stengiesi, kažkiek žinai komunikuoti su stake holder'iais, kuriems bus svarbus tas feature'as, kurį tu čia kuri tai ar koks vidinis, ar čia būtų išorinis, kažkoks feature'as žinai ir jo, tai bandai šnekėtis, ar čia atrodo gerai. Kartais tokiu metu atsiranda, žinai kažkokios papildomos metrikos arba gal kažkokie skirtingi biški benchmark'ai, ką tu ten bandai padaryt? Nes iš pradžių gali atrodyti kažkoks. Nežinau, su tuo pačiu klasifikavimu žinai. Gal ta prasme žinai, skirtingi error'ai ten, skirtingas reikšmes truputį turi, kad pvz. false positive'as ar false negative. Skirtingoj srity gali labai didelį impact'ą turėti vienam ar kitam dalykui, tai paskui jo, pradedi žiūrėti, kad kažkokia tai metrika, kurią pasirenki, iš pradžių atrodo šiaip gerai, bet, sakykim, tas false positive'as koks nors ten yra labai didelis, kuris yra žinai, tau atrodo, dabar truputį jau įsigilinus į problemą, matai, kad tas false pozityvas yra žymiai blogiau negu false negative'as. Tai tau tiesiog vieną klaidą daryti yra žymiai skaudžiau nei kitą klaidą daryti. Pradedi kažką, tada gali koreguoti savo metrikose ir žiūrėti kažką gal kitą reiktų kažkokią kitą metriką optimizuoti ir gerinti. Tai tada gali pradėti biški stumdytis tas tavo benchmark'as nuo to pradinio žinai.

Bet jo tiesiog šita sritis, būtent data science labai skiriasi nuo tipinio software engineering dėl to, kad žinai, tai labai neapibrėžta sritis, ypač kai software engineering tau reikia padaryti feature'ą, tu jį padarai ir žinai, kad padarei. Ir čia viskas gerai. Su data science yra taip, kad tu gali labai greitai kažką padaryti. Bet tu gali. Tu žinai, kad paskui gali pagerinti tą rezultatą ir tiesiog nuolat gerinti rezultatą. Ir ir čia yra dažniausiai sudėtingiausia dalis, kad nuspręsti, kada sustoti tobulinti arba kad paleisti kažkokį žinai ten, tokį pilotinę versiją žinai, pasitestuoti, žiūrėti ar gal to, ką turi dabar užtenka. Tai vat jo, čia dažnas dalykas, sakyčiau gan populiarėjantis yra. Tai vat tie visokie canary deployments, kur tu, žinai, tau gali irgi metrikos rodyt vieną kažkokį statinio data set'o, bet paleidi, padeploy'ini, sakykim 10, nu tu turi jau čia bandydamas pagerinti feature'ą, nebūtinai naują feature'ą paleidęs. Tai tu žinai, tu senasis tavo modelis. Sakykim, ten dar eina 90% userių, 10% userių tu randomly žinai paduodi tą naują modelį. Ir tu žiūri, kaip žmonės reaguoja iš kažkokių metrikų arba tiesiog paprasčiausiai ar ten jau tam live feature. Žinai, tas tas rezultatas, kur tau statiniam data sete atrodė geriau ar ar live padeploytintas irgi atrodo geriau. O gal ten kažkas kitaip veikia negu tas tavo pasenęs data setas pvz. Ir kažką bandai atšaukti, tada

grąžinti prie seno ir panašiai. Tai jo sakyčiau tie canary deployments gan populiarėja, kad tu bandai po truputį žmonėms kažkokį naują dalyką ir žiūrėti, kaip žmonės reaguoja, ar nėra kažkas labai blogai.

I.: O buvai minėjęs, tą kad, vat pavyzdžiui, pamatai, kad viena klaida yra blogesnė negu kita klaida. Gal turi kokį pavyzdį specifinį?

P.: Nežinau. Aš dabar galvoju iš. Kažkokių gal. Šiaip yra, bet aš dabar galvoju ką čia. Nežinau. Čia šiaip yra žiauriai senas projektas, kurį kažkada esu daręs. Tai buvo irgi su amžiaus, bet čia išvis. Pavyzdžiui, čia yra toks pavyzdys, kur čia buvo išvis super pilotinė versija. Čia realiai buvo testas pasižiūrėti ar ar, ar tai tiesiog iš viso naudinga dirbti prie šito dalyko, ar ne. Tai čia buvo. Realiai parduotuvėse žinai, savitarnoje, žiūri į žmogaus veidą ir patikrinti, ar ten, sakykim jis turi virš 18 žmogus ar mažiau nei 18 žmogus tam tikras prekes, kurių gali parduoti tik 18-mečiams ar nu tai čia skirtingose šalyse aišku skirtingi tie rėžiai. Tai vat šitoje srityje pvz. yra gan labai svarbu. Yra viena klaida yra žymiai didesnė už kitą. Dažnai tų ne jaunesniam nei 18 metų žmogui leidęs nusipirkti kažko, ko jis negali, tai tu sau patiri žymiai žymiai didesnę finansinę naštą, galimai kažkas jei paskui pastebi šitą ir dar kažkas kitas pastebi. Negu neleidęs kažkam nusipirkti, kas yra 18. Ir tada tiesiog prieina pardavėja ir patikrina, ir viskas tuo baigėsi. Tai čia yra vienas dalykas, kuris toks labai didelis skirtumas tarp vienos ir kitos klaidos. Kitas dalykas irgi tame pačiame projekte, pavyzdžiui buvo.

Tai čia nebūtinai gal kad false positive, false negative, bet kad. Patiems dažnai modeliai irgi vat turi tokį labai bias'ą, data set'ams ir iš esmės, kad irgi visi žinai, visokie veidų ir amžiaus data set'ai yra labai tokie. Pasislinkę į vyresnių žmonių pusę. Vidutinio amžiaus žmonių yra dažniausiai daugiausiai nuotraukose. Tai vat būtent tokioje srityje irgi tu. Jeigu pradedi žiūrėti apie tas klaidas, kiek daro modelis klaidų tam tikrose amžiaus range'uos. Bet tu matai, žinai, kad ten vidutinio amžiaus žmogui jis daro labai pakankamai mažas klaidas, gan gerai pasako, bet kai pradedi žiūrėti būtent 18 metų, ten yra didžiausias. Didžiausios klaidos arba iš viso žinai, dešimties metų amžiaus range'e, ten yra didžiausios klaidos, nes tiesiog mažiausiai nuotraukų žinai, yra bendrai prieinamų ant kurių galėtum bendrai treniruoti modelį, kad jis pasakytų ir yra pauzė šitoje vietoje. Būtent čia ir buvo pagrindinė problema. Žinai, kai tu pastebi, kad tam range, kur tau yra svarbiausia, tai tam 18 metų range, prasčiausiai modelis performina ir iš tos pusės yra labai svarbu kartais pastebėti, kad modelis, ta prasme kažkokio modelio performance. Nebūtinai tiesiog bendrai, bet tam tikruose kažkokiose grupėse kažkieno tai žinai, ir koks būtų tavo sritis, žinai.

I.:

Kokiose situacijose įprastai galvoji apie naudotoją, galutinį, tiesiog savo darbe?

P.:

Nu nežinau, aš asmeniškai stengiuos beveik visada turbūt galvot. Turbūt čia su gal su patirtim vis labiau ateina tas dalykas. Gal ten pačioj pradžioj karjeros ne tiek galvodavau, kiek tiesiog buvo labai įdomu išbandyti skirtingas visokias technologijas ir viską. Su laiku sakyčiau, kad gal vis labiau galvoju apie, tiesiog kad naudą neštų realiai, o ne tiesiog žaist su technologijom. Nežinau, gal asmeniniuose projektuose, nebent būna, kad turbūt ir su dauguma pažįstamų ir kiek esu šnekėjęs, tai dažniausiai asmeniniuose projektuose žmonės bando žinai kažkokius, kažką fun padaryti. O vat darbe tai dažniausiai, iš tikrųjų sakyčiau, kad data scienist'ai nu gan gerai sakyčiau šitą šitą dalį daro, kad būtent galvoja apie vartotoją. Iš mano patirties tai sakau ir pats beveik visada realiai galvoju, kur būtent, kad sekanti kurs kažką, kas bus naudinga galutiniam vartotojui, o ne, o ne tiesiog žinai, kad kažkas bus įdomaus sukurt.

Nes tiesiog sakau, nes dažnai būna tas, kad labai pastebi, kad tas, tų sakau low hanging fruits dažnai labai daug būna. Tai vat būtent, kad gali pamatyti, kad su labai mažai darbo tu gali kažkokią didelę naudą atnešti su tais feature'ais kur kažkas, kas dabar daroma, žinai, tu ten gali automatizuoti ir, sure žinai gali, greit nepadarysi, nieko ten tobulo, bet kad žymiai geriau padarysite tikrai ir gan greit. Tai vat čia gan dažnas dalykas yra tas, kad gali tokį staigų pagerinimą gan greit atnešti. Daugumai sričių.

I.:

Tai ir dabar gali tokį pavyzdį greit duoti tokio vat pagerinimo?

P.:

Kad pavyzdžiui kažkas su tuo nesugeba, nes čia turbūt ne vieną tokį dalyką esu pats daręs arba kad komanda kažkoks narys darytų. Tai vat būtent su tais teksto klasifikavimu kažkokiu kur tu turi 1000-čius, tiesiog žmogui jau realiai vos ne reikia ne vieno etato žmogaus, kad eitų žiūrėti į kažkokius tai tekstinius dalykus. Tu gali labai greitai tą susistemazituot. Žinai, labai greitai padaryti tokį modelį, kuris ten sakau, darys klaidų, bet bendrai tai žymiai paefektyvins darbą. Šiaip sakyčiau, kad labai dažnai šitie yra. Vidiniai įmonės dalykai kur, kur kažkokį gali modelį pritaikyti, kur greit, kažkokį naudą atneša. Tie tie patys pavyzdžiui, irgi čia žinai, kiek yra tokia data driven gal sakyčiau įmonė, tai ta prasme labai dažnai būna žinai, kad dar kažką kažkokie žinai, ten kažkokias metrikas, kažką tokio būna dažnai kažkokie menedžeriai ar kažkas tokio skaičiuoja. Žinai ten rankom ar kažkur tai eina žinais, paprašo kažko, kad surinktų kažkokius duomenis iš SQL, ten paskaičiuotų ir panašiai. Ir tai vienas dalykas, tai irgi nemažai manual darbo kažko parašyti, kad kažkokį surinkti SQL data, pasakyti čia procentą kažko, tai žinai, kažkokią metriką.

Kitas dalykas, tai gali atsirasti dažnai klaidų šitame dalyke, nes tiesiog tu nuolat prašinėji ir kiekvieną kartą truputį kitokį dalyką. Ten labai lengva įsivelti klaidoms ir panašiai. Tai vat ir čia iš data engineering'o pusės. Pavyzdžiui, čia irgi labai lengva yra paefektyvint tiesiog, surinkti tuos duomenis visur, tiesiog turėti vieną žinai kažkokį tiesos šaltinį. Kur žinai, kad ten tikrai yra teisinga informacija. Ir tau tada irgi nereikia daug custom dalykų daryti. Tu žinai, kad teisingą info gauni ir nereik ten nieko prašinėti, tai čia irgi sakyčiau toks dažnas dalykas labai kur gali labai paefektyvint kitų žmonių darbą, kad jie galėtų, tie vat sakau kažkokie menedžeriai ar kažkas tokio. Labiau užsiimti irgi galvojimu apie tą user'į, mažiau apie kažkokius ten techninius: susirinkimą duomenų iš skirtingų vietų. Tai.

I.:

Kaip tu pažintum, kad vat modelis yra blogai sukurtas?

P.:

Turbūt iš dalies tai pirmas turbūt yra žingsnis. Tai yra dažniausiai kažkaip komandoj apsitari. Tai nežinau, bent jau irgi iš mano patirties turbūt data scientist apskritai tiesiog yra labai kritiški žinai, žmonės dažniausiai moka gerai pamatyti problemas, galimas tavo modelyje, ta prasme ir iš techninės pusės, ir iš, ir iš nebūtinai techninės pusės, ir iš kažkokių. Ta prasme irgi duomenų panaudojimo, saugumo pusių. Iš visur sakyčiau, kad bent jau mano patirtimi. Ta prasme aš ir pats mėgstu ieškoti, prisikabinti prie kažko. Ieškoti kažkokių problemų ir tiesiog bendrai tiesiog. Tiek, kiek pažįstų, kiek esu dirbęs skirtingose komandose, tai dažniausiai visi gan gerai moka. Žinai, turi skirtingų patirčių, kur kažkokių yra buvę problemų ir moka pasidalinti tomis žiniomis, tai dažniausiai gan gerai pagauna. Komandos nariai žinai tokias problemas. Tai kitas dalykas sakau irgi aišku yra legal čia visi dalykai, bet šitas dažniausiai žinai būna bent iš anksto čia. Ir jeigu susiję su būtent su kažkokiais duomenų saugumais, kad žinai, ką tu gali naudoti, žinai, ko negali naudoti. Ten turbūt kiekviena įmonė turi savo.

Vienas dalykas, aišku, yra GDPR, kurį visi turi laikytis. Bet kitas dalykas yra tai, kad kartais kai kurios įmonės kažką biški papildomai daro kažkokius turi savo įsipareigojimus, galbūt įsipareigojimus su partneriais, kur irgi negali jų naudoti duomenų kažkur kitur, tai iš kitos pusės jo. O šiaip. Tada vėl iš techninės grįžtant. Tai. Dažniausiai irgi žinai tu monitorini, nes šiaip stengiesi visada paleidęs tą žinai modelį irgi monitorint ne tik tuo metu, kai tu paleidi. Dažniausiai žinai, kai paleidi, tu įsitikinęs, kad jis šiuo metu geras, bet kai paleidžia į production, tu irgi monitorini ir stebi tiek ar input data nesikeičia? Žinai, tai ta driftas, ar nevyksta ir tuo pačiu kažkoks concept driftas ar nevyksta kur tiesiog vos ne žinai, atsiranda kažkoks pakitimas iš vis, ką tu čia bandai išspręsti ir kažkas tokio, tai bandai monitorinti ir

žiūrėt žinai ar tam tikros metrikos nekinta. Ir kaip ten duomenys pasiskirstę. Ir kada tau, vis tiek kažkada visus modelius kažkada reik retreniruoti, jeigu jie yra production'e ir tiesiog stebi tada kada jis. Kažkokį tiesiog žinai dažniausiai nusistatai treshold'ą, kad ties ties kažkokiu treshold'u tau nebetenkina tikslumas to modelio ir tiesiog tada jis nukrenta. Bandai retreniruoti, ieškoti naujų kažkokių arba tiesiog su naujais duomenimis, arba gali reikėti ir kažko gal daugiau, kad pavyzdžiui kažkokių naujų duomenų šaltinių ir ieškot kartais.

Na šiaip gal aš dar jo gal dar pridėsiu. Tai ta prasme aišku irgi yra, nu vat sakau stake holder tas nusistatymas, kad kas yra jiems pakankamai gerai. Bet čia sakau čia dažniausiai čia yra man dar prieš padeployinant, nes dažniausiai būni nusistatęs, kas yra kas yra gerai tam modeliui, kas ne. Tai dažniausiai jie tokį turi pajautimą, kad, kaip minėjau, tas tam tikras klaida žinai kad, kad kurios čia klaidos skaudžios, kurios ne. Ir kad ką reikia pasiekti, tai dažniausiai žinai, nusistatai tą, tuos visus dalykus žinai dar prieš prieš padeployindamas tai. Jo ir tiesiog vat sakau tikrai būna taip, kad žinai, tu pamatai, kad neišeis pasiekti ten tų metrikų, kurias norėjai, ir tiesiog gali paleisti kartais tam tikrą feature'ą ir net nebedaryti jo, jei matai, kad tikrai neišeis bet. Bet kai jau buvo padeployini, kai jau sakykim nusprendei, kad gerai, tai dažniausiai labiau per stebėjimą.

I.: O čia šito klausimo pradžioj sakei, kad vat data science žmonės moka gerai pamatyti ne tik techninius dalykus, tai vat kad tu būna prikimbi prie ko nors, kaip sakei, tai vat pavyzdžiui, prie ko?

P.: Tai, vat nu sakau. Pradedant nuo to, kad. Dažnas dalykas yra tas, kad tu testiniam data sete gali turėti tam tikrų duomenų ir žinai, kad productione prie jų net neturėsi prieigos. Arba ypatingai pavyzdžiui forcastinime irgi tas aktualu, kad tu žinai turi irgi daug istorinių duomenų, bet tu žinai, kad į ateitį žiūrint tu ten jų neturėsi, tai tu negalėsi to naudoti. Kitas dalykas, tai vėl sakau irgi iš tų duomenų saugumo, kad tiesiog tu žinai, kad tie duomenys, tu žinai, tiesiog negali, pavyzdžiui, dėl kažkokių priežasčių žinai naudoti. Čia dažniausiai irgi yra kažkiek šita dalis susijusi su patirtimi, specifinėj įmonėj. Ir žinojimas kažkokių taisyklių iš patirties, kad, kaip sakiau, žinai, kad kažkokių duomenų negali naudoti tam tikram dalykui, žinai.

Jo kitas dalykas tai yra irgi žinai. Sakyčiau, kad nemažai. Vis tiek įmonės dirba su kažkiek data science, irgi daro ir tuos pačius data primacy kažkokius tai arba GDPR mokymus. Tai irgi skirtingi žmonės per skirtingas savo patirtis dažniausiai būna kažkiek susipažinę su tais dalykais, kad. Labiau vienas patyręs arba vienas vienoj srity dirbdamas, kitas kitoje turi patirties. Tada tiesiog su tam tikrais duomenimis, kur atrodo žinai, čia ta prasme Mažiau. Mažiau patyrusiam kartais žmogui gali atrodyti, kad tie duomenys prieinami, tai aš ten kažkur galiu juos naudot, bet. Ne visada žinai kaip gal veikia,

kartais ten žinai, kad šiaip ten tų duomenų modeliavime negali naudoti. Sakau, jei kažkokie ten yra susitarimai ar kažkas tokio, arba kad sakau, kad jie tavo, tas pipeline'as privalo eit per tą data lake'ą, kaip sakiau, žinai ir ir. Ir tu žinai, kad data lake'e ten sukauptos private informacijos tiesiog negali būti. Tu tiesiog tai žinai. Tai gali atrodyti, kad modeliuojant žinai, ten turi statinį data setą, kuris atrodo žiauriai geras. Žinai, bet žinai, kad production'e tu negalėsi to naudoti, turėsi eiti per (nesuprantamas žodis) ir ten bus kažkokia biški pašalinta dalis ten tos informacijos. Tai. Tai jo tu tiesiog per inputus žiūri. Žinai, kad sakau patart kartais, bet tai jo tai sakau čia ir bendra patirtis dažnai ateina ir dažnai. Įmonės specifinė patirtis, kad. Žinai, tiesiog turi žinoti, kas įmonėje yra saugoma ir kur ir ką galima naudoti, ko ne.

I.: Pagalvok apie savo discipliną ir kur tu matytum gal, didžiausias rizikas etines, kad ir, arba tiesiog inžinerinėje kokybėje irgi grėsmes? Ypač tokias, kur kažkas dar nėra nustatyta, kaip norma, kaip reikia elgtis. Pavyzdžiui, tarsi tas know how vaikšto, kad jūs taip darote, bet tai nebūtinai kitas žmogus taip elgsis.

P.: Visiškai sutinku su tuo, kad tas know how nėra kažkoks standartizuotas gal, sakykim. Ir tikrai sakau bendrai mano patirtis yra tokia gan pozityviai sakyčiau šituo klausimu, bet tikrai yra žmonių iš data science tikrai nemažai yra žmonių, kurie yra iš matematikos labiau ar tos pačios fizikos kaip aš. Ne visi gal tiek domisi ta software engineering dalis, kuri yra su security tiek susijusi. Tad tikrai yra. Būna žmonių, kurie, pažįstu kur, nebūtinai gal tiek rūpinasi irgi ta vat duomenų saugumu. Jo ir. Sakyčiau, ypač didesnės įmonės sakai kur turi daugiau to tos data engineering, ypač dažnai būna, kad data engineering labai prižiūri šitą dalyką. Tai, vat. Data engineering dažnai įneša vat tą tokį standartus, kad žinai, kad data science, kurie yra mažiau techniniai, kad jie žinai žinotų, kad tie, ką ko galima daryti, ką negalima daryti, žinai ką su duomenimis tai. Tai yra kažkiek šios problemos. Sakyčiau, kad ne visiems. Ne visi yra tokie gal techniniai iš tos software engineering'o pusės data scientist'ai, bet nežinau, tikrai sakyčiau gerėja ta ta tas ta problema ir nu jo. Bendrai tai bent jau man taip. Aš irgi sakau, iš pažįstamų gal kiek esu šnekės, tai man atrodo tiesiog pataiko ir aš, kaip turbūt specifiškai pats ieškau. Dažniausiai kur dirbu, kad būtent žmonės būtų labai tokie labiau inžinieriai negu tie tokie moksliški data scientistai, kur dirba, įpratę dirbti toje mokslinėje aplinkoje, kur žinai, tu neturi tiek judančių dalių ir tokių privačių duomenų dažniausiai turi labiau statinius data setus ir ten žinai. Tuos visokius. Nu tikrai sakyčiau akademinėje pusėje mažiau yra to tokio inžinerinės dalies, kuri prižiūri tą duomenų saugumą. Sakyčiau. Ir aš pats sakau dažniausiai ieškau labiau jo tie inžineriniai, tokie žmonės, kurie prisižiūri šituos dalykus, tai bent jau mano patirtim. Tai aš turbūt visur dirbu, kur gal geriau negu vidurkis sakyčiau. Tai gal žinai mano išreikšta biški nuomonė šituo klausimu?

Tada nežinau. Iš Probleminių sričių šiuo atveju turbūt aš asmeniškai turbūt nelabai man. Man bent asmeniškai aš nemėgstu ir nenorėčiau dirbti kažkur, kur yra grynai. Toks per daug optimizavimas. Žinai, kaip čia pasakyt? Pasigriebt viską. Sakykim, visą dėmesį žinai ar kažką tokio iš iš vartotojų naudojant žinai tuos tools'us, mašininį mokymąsi ir panašiai. Tai žinai, aš nežinau, aš tai žinai, kažko ieškau, kur tą mašininį mokymąsi pritaikai arba dažnai kažkokiems visai nesusijusiems dalykams su marketingu ar kažko kito, ar kaip tik kartais, gal net su šituo atveju tai reik labiau kovoti prieš prieš tą visą marketingą dalykų. Kur sakau, mes bandome kažkiek taikyti mašininį mokymąsi, kad. Tiesiog lengviau ir efektyviau galėtume userių duomenis trinti iš intiko. Tai jo, tai aš sakyčiau, kad man bent asmeniškai sakau, kad turbūt nuo to viskas ir prasidėjo. Vat būtent kažkoks marketingo padėti ir žinai, kad kuo labiau pritrauktas žinai būtent su data science field'u. Ir dėl to aišku yra taip, kad čia turbūt vis dar yra populiariausias pritaikymo būdas mašininio mokymosi. Būtent kažkokius tai išspausti kiek įmanoma daugiau žinai iš vartotojų savo produktus. Ir čia aš manau, kad marketingas bendrai nėra blogas dalykas. Tai yra teigiamas dalykas, bet kai kur kartais jau truputį gali peržengti ribas, tam tikras per daug žinai su. Bandydamas įbrukti kažką, ko nebūtinai žmonėms reikia. Tai kažkoks disbalansas turi būti. Ir man atrodo, kad būtent tas mašininis mokymasis dažnai yra pritaikomas šitoj srity. Būtent gal biškį per stipriai, kad.

Iš kitos pusės, kaip tik čionais turbūt kas dirba toj srity, žinai irgi gal gali kaip tik atvirkščiai galvoti, kad jie nenori siūlyti produkto tiem kam nereikia. Tai dėl to yra labiau targetinimas tik tiems žmonėms, kurie potencialiai reikia arba norėtų tokio produkto. Tai nežinau. Turbūt skirtingi žmonės skirtingai galvoja, tai aš toks per viduriuką, kad man atrodo, kad čia turi būti kažkokios irgi biški, kad kartais man atrodo yra biški peržengiamos ribos. Iš techninės pusės, Tai vat būtent duomenų saugumas. Turbūt yra vienas dalykas, kuo aš labai mėgstu domėtis. Tai būtent man atrodo, čia yra problema. Žinai, kad kai kurios įmonės renka daugiau duomenų negu joms tikrai reikia. Žinai sistematizuoja ir kažkur bando išlaikyti ilgiau negu reikia tuos duomenis. Nebūtinai saugiausiais būdas saugo tuos duomenis, tai tai šitas tikrai galėtų man atrodo būti geriau daroma. Jo nesakau, nes bendrai tai man atrodo. Aš sakyčiau, kad. Žinai tą, kaip sakau, tą, mašininio mokymosi, ar ten kažką visa data science field'ą žinai galiu pritaikyti teigiamai. Į teigiamą pusę būtent ir tų pačių duomenų saugumo klausimu. Bet jo tiesiog reikia aktyviai turbūt dirbti ties tuo. Nes sakau visada. Visada žinai, data scientist'as visada nori kuo daugiau duomenų visur žinai, žinai, kad tą kažką geresnio galėsi sukurti, bet irgi realistiškai žiūrint, tiesiog neįmanoma. Visų duomenų visada visur apsaugot. Žinai, visada kažkur yra kažkokia silpnoji grandis ir grandis. Žinai, kažkur tie duomenys vis tiek gali kažkada nusileak'int. Tai tiesiog irgi sakau žiūrint realistiškai, vis tiek reikia žinot kažkokias irgi sau ribas, kad ką reikia saugot, ko nereikia, ko reikia vengti. Tai jo, tai čia turbūt iš techninės pusės pagrindinė problema būtent.

I.: Gerai, imant visą procesą nuo pat pradžios, kai naudotojas naudoja kažkokį įrankį, kaip tada duomenys apie jį yra surenkami, tada duomenys yra agreguojami ir visaip apdorojami. Modelis daro rezultatą ir pateikiamas rezultatas. Tas galutinis rezultatas labiausiai nuo kokių žmonių priklauso, naudotojų, inžinierių, data science'o ar kaip tau atrodo. Kas turi didžiausią įtaką pačiam galutiniam rezultatui to produkto, kurį kuria?

P.: Čia geras klausimas. Šiaip manyčiau. Kad greičiausiai vis tiek. Nors šiaip čia labai sudėtinga pasakyti, nes man atrodo, kad tai skirtingai gali atsitikti. Bet aš vis tiek manau, kad iš tos pusės, kad. Data quality yra svarbiau negu data kiekis. Tai tai čia sakyčiau, kad kažkas tarp data engineering, nors net galbūt ir nebūtinai data engineering'o, gal tiesiog frontendo ar backendo, kuris kažką surenka tuos duomenis, pirminio šaltinio duomenų. Tai čia data engineering'as, ir userių, nes sakau, kaip useris žinai vartojo, kaip kažkokius duomenis pateikia. Tai nuo to labai priklauso tas galutinis rezultatas, tiesiog duomenų kokybė, sakyčiau, labai daug lemia.

Bet iš kitos pusės, dažnai tą duomenų kokybę tu kaip inžinierius gali sukurt arba nežinau pakuruot į gerąją pusę. Tai sakau, jeigu tai yra kažkoks duomenų surinkimas, būtent kur, kur ne automatiškai renki duomenis. Vat kaip tu minėjai per frontendą, kur spaudinėja žmogus, o tai nežinau kokia nors apklausa žinai yra, ar kažkas tokio. Arba tavo appsas, kur žmogus pats įveda kažkokius duomenis, tai šitoje vietoje žinai pati įmonė turi sprendimą, kaip leidžia įvesti žmogui duomenis ir tiesiog gali teisingiau, teisingesnius sprendimus daryti, kad surinkt tuos duomenis tvarkingus ir geros kokybės ar mažiau kokybiškus. Tiesiog pagal pagal tai, koks kokie klausimai yra Klausinėjami, kokie, nežinau, kokie field'ao, ką leidžia įrašyti į tuos field'us ir pan. tai nuo to labai priklauso, kiek ką useris įsideda, bet ir ką įmonė leidžia įvest žinai. Tai sakyčiau, kad nu kažkur tarp userio ir nežinau, frontend'o kažkas tokio, bet sakau, labai priklauso turbūt nuo to, kad kokia sritis, nes kartais būtent minėjai, kad nebūtinai useris pats įveda tuos duomenis, kurie surenkami iš cookies ir panašiai.

I.: Ar koreliacija nelygu priežastingumui, vis dar.

P.: Jo, sakyčiau vis dar. Tu gal turi kokį pavyzdį ar kažką? Kodėl tu tokį klausimą klausi?

I.: Nes 2000 metų pradžioje parašė vienas žmogus straipsnį, kuri sako "Mes dabar turime tiek daug duomenų, kad mums nebesvarbu suprasti, kodėl, kas daro, kam įtaką".

P.: Jo. Nu, čia yra tiesos, turbūt modeliavimo, kad. Ta prasme. Bandant nuspėti kažką, tai jo koreliacijos pilnai užtenka. Nereikia to dažniausiai žinai tos, priežastingumu išvertei, man atrodo ane? Causation. Tai dažniausiai tai jo. Ta prasme kažką bandant nuspėti, tai tau tikrai to nereikia. Bet bet bendrai vis tiek tai pats tas procesas iš sakyčiau, gal grynai iš mokslinės pusės žinai, kad tu kai žiūri tiesiog, kas nulemia kažką? Žinai tu čia visiškai apie kitus fieldu irgi žinai, šnekant apie tą pačią mediciną

ar kažką tokio. Tai vis tiek tai yra visiškai tiesa. Žinai, kad tu gali surast kažkokios koreliacijos, bet nereiškia, kad ta koreliacija iš tikrųjų sukelia tą. Tai tau tam, medicinoje tau labai svarbu žinoti, kas sukelia, nes tu žinai, tu negali tiesiog sakyti šitas dalykas koreliuoja. Ir čia darbas baigtas išrašysiu kažką, kas tą koreliaciją sumažina. Ir tu žinai, ir ten pasveiksi, bet taip neveiks. Bet. Bet iš būtent tos mašininio mokymosi pusės ar tiesiog modeliavimo pusės, kad jeigu tu bandai kažką nuspėti, tai beveik visada tau kažkokią koreliaciją. Pilnai užtenka, tiksliau data science statistikos tikrai lieka. Būtent mašininis mokymasis turbūt ir skirias tuo nuo tradicinio irgi tų visų modelių senųjų, kur nebereikia tiek daug žiūrėti į tas senąsias, statistines visokias. Ten normalumus tikrinti ir panašiai. Tu žinai tuos dalykus, gali išspręsti tuo tiesiog, kad savo monitorini žinai tu. Tai žinai, tu nebandai kažkokias nulines hipotezes paneigti ar kažką tokio tu bandai nuspėti tiesiog, tam tikrą dalyką. Tu žinai, kad tie dalykai yra dalykai, kuriuos tu turi, koreliuoja su tavo norimu dalyku nuspėti. Ir tau visiškai to užtenka. Jeigu jie nustos, koreliuoti, tu matysi tą, žinai savo monitorinime, išmesi tą dalyką arba pasiimsi kažkokį kitą feature'ą, kuris koreliuos geriau. Gal iš to feature'o žinai kažką biški apdirbsi, jį truputį pakeisi ir jis vėl pradės koreliuot ir tau, jo, tau visiškai užtenka koreliacijos. Tau nereikia žinoti, kas sukelia kažką tai. Ir galvoju dar. Nu jo, tiek turbūt. Kažką gal norėjau dar pridurt.

I.: O ar yra tokių vietų, kur ir data science vis tik svarbu kartais suprasti priežastingumą?

P.: Tai suprasti turbūt visad bandai suprast, stengiesi, žinai. Nes sakau kažkiek yra va tos, sakau kad kažkas koreliuoja žinai, ir nereiškia, kad būtinai tu tą ir naudosi. Čia į vieną feature'ą, kuris koreliuoja, nes gali būt žinai, kad jis labai nestabilus ten ir žinai. Iš tos domain'o perspektyvos kažkoks ekspertas pasakys, kad šitas dalykas dabar koreliuoja, bet kitą mėnesį nekoreliuos, trečią dar kitas dalykas pradės koreliuot žinai, kažką tokio, tai tiesiog iš anksto tu gali žinoti, kad tau nereiks paskui monitorint. Be atidėt dalykų, tai iš dalies ir iš tos pusės gali būti kitas. Tai gali būti, kad. Tu du skirtingi dalykai gali koreliuoti ir kartais ten irgi jie tarpusavyje irgi koreliuot, ta prasme tas multikoliniarumas kur tarpusavy koreliuoja. Bet ta prasme ir tavo norimam dalykui koreliuoja. Tai tada tu dažniausiai irgi gali kartais pakenkti tam tikriems modeliams, kai kuriems visiškai nesvarbu. Tai žiūrint kokį modelį naudoji, gali kartais modeliui reikėti šituos dalykus pasižiūrėti ir išsinagrinėt. Arba sakau, kad tokio tipo procesai, kas kitaip tik atrodytų.

Kitas dalykas kartais gali būti, kad kaip sakiau, tu kažkokį feature'ą gali turėt, kuris ne ne su. Nebus prieinama su production. Tu dabar žinai, kad tu gali patestuoti su tuo, bet žinai, kad nebus prieinamas production, Tad tu gali. Sakau, jei nes pavyzdžiui sensitive, tai tu turi kažkokiu būdu jį apdirbti. Ir kartais, ir tai jeigu tu žinai, kad jis tikrai yra žiauriai svarbus dalykas, tu vis tiek jį turi pasiimti kažkaip, bet žinai, tai kažkaip turi visiškai pašalinti kažkokį sensitive dalį iš to dalyko, kad jis liktų naudingas.

Arba kartais gali būti net ne išvestinis iš tam tikro dalyko. Nebūtinai išvestinė kažkokia ne išvestinis feature'as, bet tiesiog kažkoks panašus feature'as, kuris koreliuoja. Tai žinai, nu nežinau, koks nors pavyzdys gali būti metų diena su kokiu nors dienos ilgiu. Kokia ilga diena vis tiek labai koreliuoja su tuo, kiek yra, kelinta diena žinai metų yra, nu tai čia nežinau. Šiuo atveju turbūt nelabai įmanomas toks variantas, kad tai yra privatūs duomenys, kažkuri dalis. Bet tiesiog kažkuriuo atveju tu gali neturėti prieigos prie vienų duomenų, bet gali turėti prieigą prie kitų duomenų, tai tada tiesiog pasiimsi tą dalyką, kuris koreliuoja prie to vadinamo proxy feature, kad tas vienas feature'as atstoja tą feature'ą, kuriuo tu negali naudotis pavyzdžiui. Dažnai tokiuose dalykuose tenka pasiaiškinti, kas čia gali atstoti šitą dalyką, kas nėra privatu pavyzdžiui.

I.: Čia ypač aktualu tais kartais, kai dėl privatumo negali naudoti?

P.: Kartais dėl privatumo, kartais dėl tiesiog techninių žinių, galimybių, kad ten ateina tie duomenys per vėlai, ne taip dažnai, kaip tau reikia. Tai jo, bet pala, dar tiksliai primink, koks klausimas buvo? galvoju, ar kažką dar turiu pridėti?

I.: Buvo, ar yra vietų, kur svarbu suprasti priežastingumą?

P.: Ai, Tai dar kur čia gali būt. Jo, nežinau. Ta prasme tu vis tiek kažkoks. Ai, nu čia turbūt visokie AB testai ir panašiai. Eksperimentavimas čia yra labai svarbu, tai bet aš su šituo su ta dalim nelabai dirbu, tai neturiu patirties, bet žinau, kad čia tikrai labai daug stengiasi visi ten dirbt priežastingumu. Nes jo, tai AB testas žinai gali paaiškinti, kažkas tai tau čia yra naudinga šitai pakeist. Bet kodėl tau tai naudinga? Tai nebūtinai pasakys ir tau. Tada nežinosi, kokie kiti pasikeitimai, kokį turės tavo reikšmei.

Informantas Nr. 5

Amžius: 33 m.

Lytis: moteris

Darbo vieta: lead decision scientist

Gyvenamoji šalis: Lietuva

Pilietybė: Lietuvos

Formatas: nuotolinis pokalbis

Trukmė: 1h 1min 14s

Data: 2024 m. gruodžio 11 d.

I.: Esu Simonas Bansevičius, politikos ir medijų magistro studentas. Darau tyrimą apie mašininio mokymosi programų etiką. Informuoju, jog bet kada gali nuspręsti nutraukti pokalbį. Ar sutinki, kad šis pokalbis būtų įrašytas ir jo medžiaga būtų naudojama tyrimo tikslais?

P.: Sutinku.

I.: Kiek gali, kiek jautiesi laisva, prisistatyk, jei nori pasakant savo amžių, ką dirbi šiuo metu, kiek laiko dirbi, kokia tai komanda, kiek žmonių?

P.: Tai aš esu (anonimizuota). Šiuo metu dirbu lead decision scientist (anonimizuota įmonė). Dirbu jau keturis su puse metų. Dabar man atrodo, jei neklystu. Dirbu search sub domene, tai mes iš esmės su paieškos įrankiais ir rekomendacijos sistemom dirbam. Tai va, maždaug taip. Man 33 metai. Mano darbo profilis tai jau yra aš dirbau keletoje skirtingų įmonių kaip data scientist arba aktuare esu dirbus draudimo bendrovėse, tai esu įvairiose srityse, turbūt susidūrusi su įvairiais mašininio mokymosi algoritmais ir įvairiais niuansais, kurie galimai paliečia mūsų šiandieninę temą. Tai jau gal trumpai taip.

I.: Kokios tavo pagrindinės užduotys?

P.: Jeigu šiuo metu, tai aš daugiausiai dirbu su eksperimentavimu ir įvairių produkto pakeitimų, kuriuos darome search'e, o taip pat aš bent jau pusantrų metų, aš dirbau ties projektu, kuriuo kūriau algoritmą, kuris pasiūlo user'iams suggestion'us, vadinamąjį query auto completion'ų suggestion'ai. Tai su tuo projektu dirbau ir jį tęsiu turbūt nuo kitų metų, bet daugiausiai dabar dirbu su eksperimentavimo įvairiais, biznio decision making klausimais, kuriuos turime savo domene. Kad ne tik apie search'ą, bet plačiau žiūrint, kaip teisingai viską atlikti, kaip eksperimentuoti, kaip įsivertinti dalykus, ką galima matuoti, ko negalima matuoti ir panašiai. Ir komanda, kurioje dirbu, tai dabar dydis yra ten jau virš dvidešimt žmonių. Jeigu konkrečiai komandoje būtų pats tas domenas, tai jau yra turbūt šimtas su viršum žmonių. Tai, kaip mano rolė apima labiau patį domeną ir visus tuos darbus.

I.: Tai gal dabar pradėkim nuo prieš tai buvusio darbo, kai kūrei algoritmą, query auto completion'ą. Tai apie šitą, kiek galėtum išsiplėsti.

P.: O kas konkrečiau yra įdomu ir aktualu?

I.: Pavyzdžiui, iššūkiai, su kuriais susidūrei kuriant.

P.: Na iš esmės šitam procese, kur, kur šitam projekte, nes aš esu turėjusi, jeigu bus aktualiau, tarkime, visi su etikos klausimai, aš galėsiu paminėti iš kitų darbų, savo susijusių dalykų. Šitam darbe didžiausias.

I.: Iš visų galima.

P.: Gerai, tai gal aš tada paminėsiu Jeigu yra įdomesnė šita tematika, tai aš visa seksiu iki (esamos darbovietės), nes (esamoj darbovietėj) irgi tų iššūkių. Bet turbūt esu galbūt įdomesnių dar iššūkių ir iš

praeities esu turėjusi. Aš turbūt neįvardinsiu įmonių specifiškai šiuo atveju, bet kalbant apie draudimo įmones, tai aš dirbau draudimo įmonėse su, tarkime, automobilių draudimo kainodaros sudarymu. Ten yra gaminami irgi Machine learning modeliai, pagal kuriuos nustatoma kaina konkrečiai automobiliui ir taip pat asmeniui. Tai iš esmės turbūt man buvo įdomiausia tai, kad tai buvo vienas mano pirmųjų darbų ir tuo metu bent jau tai, sakykim, manau, kad dabar truputį įstatymai pasikeitė. Bet tuo metu, kiek esu po to skaičius, daugelyje šalių atsirado apribojimai, pavyzdžiui, skirstyti, naudoti lyties duomenis arba amžiaus duomenis, kad tu nustatytum arba identifikuotum per netiesiogines koreliacijas, kad, tarkim, automobilį vairuoja ar ne, tarkim, moteris ir moterys daro dažniau auto įvykius ir joms tada draudimo įmoka yra didesnė. Tai tas turbūt buvo vienas pirmųjų mano susidūrimų, kur tu turi panaudoti įvairiausius duomenis. Ir draudimo įmonės iš esmės gali gauti daugiau duomenų negu šiaip turbūt privačios įmonės, nes tu gauni ir apie kredito riziką, ir, tarkim, net su būstu susijusi tavo rizika. Koks tavo kreditingumas. Ir tu gali net tai panaudoti automobilio kainodaros tarifuose įvertinti, kokia tikimybė, kad tu padarysi didesnę žalą arba, kad tavo yra rizikingas elgesys. Tai turbūt šitas buvo vienas tokių įdomesnių projektų, kur aš atsimenu, padarydavau išvadas ir kainodara buvo gana stipriai personalizuota, bet aišku, klientams tokios komunikacijos nebūdavo, kad, tarkim, kiek yra įtraukiama tavo rizikos veiksnių. Visada ten, tarkim yra draudimo, po to išsiunčiamas polisas, kuris ten yra tarkim, net gali būti kaip taisyklės. Reikia atskirai skaityti, tarkim 20 lapų, kuriuose suminėta, kokie tavo dalykai gali būti įtraukiami, bet jie gana būdavo taip sakyčiau per aplinkui kai kurios dalys apibrėžiamos ir neretai po šito, tarkim, kuriant šį modelį, aš turėdavau panaudoti tokius duomenis, kurie, manau, ne visada yra susiję konkrečiai su automobiliu, jo, tarkim, senumo ir žmogaus vairuotojo stažu. Bet tai būdavo ir, tarkim, lyties, kreditingumas arba kažkokios koreliaciniai veiksniai, kurie padėtų identifikuoti, kad šitoje vietovėje, tikėtina, žmogus atliks daugiau eismo įvykių, todėl galutinė jam kaina būtų suskaičiuojama tokia. Gana nemaži tarkime diskriminaciniai būdavo, tarkime, dalykai, kurie yra tam tikrų automobilių savininkai. Nu iš karto padaromos išvados. Tarkime, padarai prognostinį modelį, kad šiokie tokie modeliai automobilių rodo polinkį ar ne, kad bus atlikti, tikėtina, didesnės žalos. Tokiems tada klientams būdavo, tarkime, vietoje tokio output'o kainos išvedama tiesiog yra nesvietiškai didelė, tada kainos toks, sakykim output'as, kad jisai jau skambintų ir tada sakoma, kad jum sudaroma atskira kaina. Bet tai yra tik susiję su tuo, kad mes padarome ar ne tas išvadas iš duomenų, kad ten Porsche vairuojanti moteris 35 metų, tikėtina, bus, sukels mums žalą, kurią neapsimokės už mažesnę kainą laiduoti. Aš žinau, kad bendrai, tarkime, šioje draudimo srityje daug po to pokyčių buvo ir turbūt nuo to laiko, kai aš dirbau 2014-2015 m., manau, daug daugiau atsirado apribojimų, kokius duomenis galima naudoti, kad tu negali naudoti nesusijusių duomenų, tarkim, bet tuo metu ir ne tik šitam, tarkim ne tik su automobiliu.

Po to draudimo teko kurti ir ir kitokių, kur įmonių reitingams. Tu gaudavai tokius duomenis, mes juos, tarkim, pirkdavome iš Creditinfo ir panašiai, kurie iki tos vietos, kur žinai, ten tavo, ten asmeniniai turto ir panašūs dalykai. Ir tu tada ieškodavo tų koreliacijų ir sukurdavai tą modelį, kuris pasakydavo, predict'indavo tai ar galima jam tą paprastą kainodarą duoti ar išskirtinę. Tai vat taip buvo sakyčiau. Ankstesniais laikais. Ar dabar taip yra? Aš negalėčiau užtikrinti, kad visai tokių dalykų nėra. Turbūt vis tiek yra. Tam tikrų turbūt diskriminaciniai modeliai nu tarkim, yra leidžiami. Bet turbūt čia irgi daug klaustukų būtų šitoje vietoje. Kiek, kiek tai yra leidžiama ir kiek legalu. Nes po to, manau, buvau ne kartą jau apie tai skaičiusi, kad jau yra uždraudžiami, ypatingai lyties diskriminaciniai, kad negalima jų naudoti. Bet aš pati žinau, kad tu gali netiesiogiai panaudoti iš kitų kintamųjų išsivesti, kad čia yra, tarkim, moteris arba vyras. Tai sakyčiau, čia toks buvo mano pirmasis, kur aš turėjau kurti modelius ir tokie būdavo įdomesni galbūt etiniai klausimai diskriminaciniai. Tarkime, iššūkis man pačiai būdavo aš juos pristatydavau, ir aš pati suprasdavau, kaip - aš tada dar nevairavau - bet kaip aš save pasmerkiu didelei draudimo kainai ir atsimenu, kai aš po to jau išėjau iš to darbo. Aš nusipirkau automobilį ir supratau, kodėl ta kaina yra kosminė, nes aš ją pati sukūriau tokią. Tokia moteris su mažu stažu ir panašiai. Ir kitais ten apibrėžtais veiksniais, kuriuos mes susirenkam iš kitų duomenų setų.

Tada turbūt kitas mano darbas buvo aš dirbau. Apie tai, kai yra reviews'ai apie įmones. Kaip pasakyti collect'inami, tai kai yra Google, taip yra ir kita įmonė, kuri collect'ina atsiliepimus apie visokiausias įmones. Pradedant šiaip nežinau ten grožio, sveikatos industrijos ir pan. ir Google reitingai pasiima iš tos kompanijos dalį tų reitingų, kad sudarytų tą compound reitingą. Tai aš dirbau dviejuose skyriuose. Pradžiai dirbau su Trust ir Fake news ir ten gaminau iš esmės Machine Learning modelį, kuris buvo apply'inamas nustatyti, ar šita įmonė fake'ina tuos reviews ir tada, jeigu ji fake'ina reviews'us, būdavo perduodamas tas output'as iš to machine learning modelio, kad čia tikėtina, kad turėtume pažymėti raudonai šitą įmonę ir customer support'as turėtų susisiekti ir duoti įspėjimą. Nes šita įmonė yra pasižadėjusi, kad mes visus fake reviews triname. O turbūt ir pats žinai, fake reviews yra viena didžiausių scam'erių, botų ir panašiai vieta ir nu labai ten buvo toks įdomus darbas. Mes tikrai panaudojome šitą modelį, bet jo output buvo labai sunku įrodyt. Nu kaip pasakyti, mes matydavome, kur ten nu literally, aš įrodau, tarkim modelis tą output'ą didžiulį probability parodo, kad čia yra fake'iniai reviewsai pagal įvairius input'us. Bet tas treshold'as - kurį mes priduodame, jau sakyt, kad čia jau gal nu ir užblokuoti tarkim įmonės puslapį, nebeleisti jai rinkti reviews - buvo gana aukštas, nes tos įmonės perka pas mus tada paketą, kuris, pavyzdžiui, kiek tu gali ten žinai gauti visą mūsų customer support'ą, kiek ten gali žinai lengvai reviews'us per mūsų platformą atsakyti, kaip ten pashow'ina tas žvaigždutes.

Nu žodžiu integracija. Ir tada gaudavosi taip, kad, kurie moka daug pinigų, tada tu kaip ir nelabai gali to sakyti - nu jo, jie turi fake reviews, bet jie mums moka pinigus. Ir tada visada buvo šita labai keista jausdavausi, nes aš tokia labai tiesos gerbėja būdavau, bet toks padarai ir pristatai tiem management, tai čia tuos 20% reikia išvis tipo jie yra vien tik fake reviews'ai. Ten buvo ir atskira komanda, kurie pvz. iš to output, to modelio, eidavo dar kart double check'int, nes labai bijodavo ką nors tipo apšaukti, kad iš tikrųjų jie nėra fake'iniai, tai ten tokie vadinami vos ne detektyvai žmonės būdavo, kurie užsiimdavo tuo išsiaiškinimu ir jie identifikuodavo, bet, kaip ir minėjau, priklausomai nuo to, kokio konteksto ši įmonė - ar ji mums moka tuos pinigus, ar ne - būdavo handle'inamos tos situacijos, kartais skirtingai. Tikrai ne visada. Ir aš žinau, kad trust buvo labai svarbi dalis. Bet tuo pačiu svarbi dalis yra ir ir tos pajamos ir yra ta baimė, kaip įrodyti, nes tada prasideda ir visokie kartais ir bylinėjimais. Tai čia pirmas mano buvo toks, sakyčiau, šitoje įmonėje susidūrimas, kur kad ir tu turi tuos input'us ar output'ą gauni kažkokį kur gali suskirstyti, ar čia yra fake review su tarkim ta įmone ar ne, bet tada atsiranda tam tikra diskriminacija turbūt jau iš įmonės politikos. Kaip mes turime handlint šitą situaciją. Kiek mes galime pasitikėti tuo ML? O gal čia nėra taip? O gal yra? Ir jeigu dar priklausomai nuo to, ar ne, ar ten moka pinigus ta įmonė, ar ne, už už tą mūsų kažkokį siūlomą paketą, yra skirtingai tada atsižvelgiama, ar ne, į šitą visą output'ą ir tada tas toks, atsiranda daugybė layer'ių, patikrinimų, ką aš suprantu, kas yra reikalinga. Bet šitoje vietoje, be abejo, labai plona linija tarp to, kiek tu gali pasitikėti. Nes tai yra, nu tu iš esmės build'ini modelius, kurie yra pagrindinis core variklis įmonės, kuri sako, kad tavo yra trust'as numeris 1 ir jeigu tu daugiau kažkokių data input'ų įsidedi į tą modelį, ten treshold'us gali sumažinti, tu gali nemažai tų false positive gaut, o tai jau irgi yra visiškai not acceptable, bet lygiai taip pat not acceptable turėti įmones, kurios turi vien prifake'intus reviews. Ir šitoje srityje, mano nuomone, dirbant. Labai daug buvo tiek etinių tokių dilemų, tiek klausimų kuom galima pasitikėti. Ir kaip aš sakau, kai atsiranda dar dalis mokėjimų pinigų, tai ten ir atsiranda tas toks nemažas klausimas kaip tas modelis veikia? Ar jis turi dar atsižvelgti, ar nu tipo žmogus moka pinigus ir tu ten tipo, čia tada less less risky, ir tu ten pakeiti vos ne output'ą, nors tavo modelis gal ir kitaip sakytų, predict'intų. Tai va, sakyčiau vienas šitas buvo.

Toliau ten dirbant aš buvau, dirbau ir kitam departamente, tai kūriau tokį op selling, kurio selling modelį. Aš dabar visų detalių neatsimenu, bet tai buvo mano paskutinis taškas, kur aš pajutau, kad aš negaliu dirbti tokioje aplinkoje. Ir iš esmės tai buvo vienintelis darbas, kurį aš dėl moralinių sumetimų po to mečiau, nes ten, kur jau atėjau į tą vietą, kur irgi reikėjo man sukurti modelį, aš tiksliai nepasakysiu, kaip tai veikia, bet iš esmės ten ta įmonė siūlo planus, kaip tu gali tuos reviews pildyti ir pan. Ir mano tikslas buvo kaip prapushint tarkim brangesnius planus pagal kažkokį tai ten jų nesėkmės behavior ir panašiai.

Buvo iš esmės rasti tuos neigiamus dalykus būtent tose įmonėse, kad kažkas joms ten neveikia. Tokius inputus sudėti. Ten tada galima duoti tuos call skambučiams. Būtent tos įmonės ir pagal tu ten predict'ini. Dar atsimenu kažkokius Quality insight'us pagal berods ten kintamųjų svarbumą. Galiausiai ten išversdavo kažkokias key tezes, kurias galima ten vos ne pasakyti, žinai, kad jeigu jūs nenusipirksite šito, tai jūs jau čia negausite ten tipo sau populiarumo. Ir useriai jumis nepasitikės. Ir panašiai. Ten būdavo agenda aplipdoma iš to output'o, kurį tu padarai su tuo modeliu. Tai man buvo gana ypatingai, šita jau dalis dar labiau turbūt buvo svetima, nes ten tiesiog. Labai daug reikėjo iš tiesų neigiamo input'o dėti, tarkim, į modelį. Taip jį diskriminacinį padaryt, kad atrasti tiems, kuriems, sakykim, galima dar prapush'int. Ir ten būdavo, tarkim, kiekvieną kartą, jeigu nepasiseka, tie skambučiai: "gal tu gali dar kartą peržiūrėti tą modelį? Galima dar kokį input įdėti?" Gal galim iškasti dar kokios ten tos informacijos, kuri padėtų nustatyti, kas tikrai sugebėtų iš mūsų tų tada skambučių įrodyti, kad jūs nu turite nepakankamai šito, todėl jums free planas netinka, arba kad jums reikia brangesnio plano, kad jūs. O kaip aš sakiau, kad susijęs kuo brangesnis planas, tuo mažiau tada apribojimų gaunasi visam tam trust'ui. Iš esmės tu tada daug labiau praslysti pro šitą visą vietą ir ir tada visi kiti, kuriuos ML modeliai ten buvo. Čia jau ne aš gaminau, bet tarkim sentimentų analizė ir pan. Tai analogiškai irgi būdavo brangiausiai perkantiems tada sentimentų analizė, kurią irgi gali pashift'int, plius minus kaip nori įdedi, dar tų filtrų, parodai galbūt tik tais only positive ir tą išsiunti į tą API, į tuos rezultatus. Tai, sakyčiau toks įdomesnis buvo, kur labai daug buvo AI ML naudojama, bet jis buvo ypatingai diskriminacinis momentais vien dėl to, kad tai yra susiję su pinigais on the other side. Kai nėra visiškai, nors ta problema tai be abejo. Tai sakau tie reviews, Facebook, google ir visos kitos susiduria su ta pačia problema. Bet turbūt irgi yra tas skirtumas iš kur ateina, jeigu tai yra, tarkim įmonė, kurios visas tada pelnas yra iš to, kad tu nusipirktum papildomai kažką. Tada gaunas to pasitikimumo ir sakau, tie visi modeliai gaminami galiausiai jie turi dar būti praleisti pro kažkokį filtrą ir tai, ką tu ten. Koks jaunas žmogus ateini, tai čia visi blogi, tai reikšia čia juos. Su tuo, kad sako, tai gal tu gali dar čia papeakint, šituos palik, bet tuos reikia ir tada gaunasi jau žinai. Ml modelis plius minus su kažkokiais dalykais, kurie kartais mano nuomone tikrai prasilenkdavo su su įvairiais moralės ir etikos klausimais ir ten data nėra taip, kad aš paduodu data, gaunu output'ą ir mes jį arba ten validuojam. Bet užtat tu bandai kažkokius rasti veiksnius, kurie konkrečiai specifiškai padėtų identifikuoti kažką ir už tai, kad būtų dar kažkokia nauda. Tai va. Tai sakyčiau, čia tos mano pirmos patirtys, kurias turėjau šitoje srityje.

I.: Kaip atrodo tavo sukurto sukurtos programos vertinimas? Vertinimo procesai? To ml vajaus ar kad ir ką kuri?

P.: Turi omenyje apie vertinimo tikslumą ir pan..

I.: Jo, sukūrei, reikia vis tiek įvertinti, ar jis geras, pakankamai geras.

P.: Jo, tai mes dažniausiai čia depends, kokį modelį buildinam, tai aišku jeigu šiaip bendrai ML modeliai, tai tu ten pats pat turi ten tas data, tą accuracy F score ir pan. Bet mes ką dažniausiai darome, tai mes eksperimentuojame. Tai su mano modeliu irgi lygiai tas pats buvo pirminiu tuo algoritmu, kuris, sakau, yra tik pagal šalis ir kalbas ten identifikuoja, kaip rank'inti tuos suggestion'us. Tai mes padarome eksperimentą galiausiai, nes jis yra šitas, bent jau modelis. Jis yra labai didelis duomenų inžinerijos, sakyčiau labiau modelis, labiau toks kaip rule based. Tai galutinis jo vertinimas. Mes lyginame su kas production'e buvo, kas yra niekas nerank'inam niekuo, ar ne ten tiesiog yra iš sort'inta septinto pagal number of listing. Na tarkim ir ten visai kitaip veikia tie suggestion'ai. Ttiesiog suknelė žinom, kad ten penki tūkstančiai daiktų. Čia aš šiaip sakau, sijonas ten keturi tūkstančiai, dedame suknelės numeris vienas, tai mano jau šitas visai kitoks algoritmas, kuris ten vadinas labiau kaip most popular completion modelis. Tai mes jį įvertiname tada eksperimento metu, kiek jis atneša naudos useriams. Tada lygina metrikas, tarkime, tarp suggestion usages, click through rate'as, tada transactions per active user ir panašios metrikos, pagal kurias mes nustatome, ar šitas modelis veikia geriau negu esamas production modelis.

Tai va, šitas buvo taip įvertintas ir buvo atlikta labai daug eksperimentų. Jei neklystu, per metus ar apie trisdešimt, kuriuose jo dalyvavo kartais tik vienos šalies, kartais visų šalių useriai, kurie būdavo dalinami į atskiras dalis. Tai dažniausiai būdavo du variantai, kartais keli variantai ir pagal jų userio elgesį mes tada įvertindavom, kuris variantas, koks ten rank'ingas jiems geriausiai veikia. Tai daugybė user'ių buvo labai dažnai testuojami. Čia irgi toks turbūt vienas iš tokių niuansų, kur su eksperimentavimu, kad kai yra labai dažnai eksperimentas, tu matai vis kitokią versiją. Ir tai yra vienas pirmųjų žingsnių, ką tu ateini į search'ą, pradedi type'int ir tu matai, o tam tu kažkiek dienų matai vieną, po to vėl matai kitą. Tai čia useriam irgi toks yra. Su šituo, sakyčiau, nėra buvę. Gal yra buvę kažkokių customer supporte ticket'ų, bet mes tikrai esame gavę iš kitų galbūt tokių eksperimentų, kur, pavyzdžiui, testuojam tai gali būti ypatingai rekomendacinės sistemos. Aš konkrečiai pati nedirbu, bet pakeičiame kažką arba išjungiame useriam, jie tada ateina ir skundžiasi, kad aš tada mačiau prieš savaitę. Aš dabar nematau. Ir panašūs tokie dalykai. Arba kodėl man čia dabar neveikia taip, kodėl buvo anksčiau? Tai turbūt irgi eksperimentavimas tokia viena iš tokių įdomesnių dalių, kurioje susiduriama, kad čia yra geriausias būdas išsitestuoti, suprasti, ar tavo modelis gerai veikia, bet tuo pačiu jis gana sutrikdo aišku, userio to behavior kartais. Jeigu tu ilgai testuoji, tai, pavyzdžiui, mūsų pusantrų metų projektas. Mes labai daug testavom ir mes tą tris dienas paleidžiame, matom bug'as arba kitaip neveikia. Išjungiame, įjungiame vėl.

Tai daug nemąstydavau apie tai, bet po to suprasdavau, kad tai yra irgi kažkoks friction point'as useriam ir jie supranta, kad jie yra nuolat įtraukiami į šitą testavimą. Jeigu tu paklausi, ar čia lengva pasakyti, kad aš noriu būti netestuojamas ir nebūt įtraukiamas, aš šito negalėčiau atsakyti. Čia labai geras klausimas ir aš galvoju, nes čia man, kiek aš žinau, čia irgi turėtų būti useriaiui teisė atsisakyti būti testuojamam. Ir aš nežinau, bet čia geras klausimas, kurį turėčiau turbūt pasidomėti, ar įmanoma atsisakyti, nes tikrai žinau, kad jau yra buvę, kad žmonės jaučia frikciją nuo to, kaip jie dažnai būna išmėtomi. Ir mes dar neturime tokios, turbūt čia irgi į temą, kad tokios sistemos, kur mes dalintume kažkaip userius į grupes, tarkim pailsėt, imtume vieną grupę žmonių, kurie tiesiog galėtų būti netestuojami. Kiek esu buvus ten ir konferencijose, didžiausios jau tos paieškos, tarkim, įmonės Bing, Google ir panašiai. Jie skaido tuos savo userius ir leidžia daliai userių būti testuojamiems tik kažkiek mėnesių ir jie po to nebe nebebūna taip testuojami. Pas mus kol kas, deja, taip nėra ir aš jo nemanau, kad tai labai gerai yra. Tuo labiau, kai sakau kartais aš pagalvoju, kai keli šimtai testų kartais toje pačioje vietoje, kaip aš paminėjau, pavyzdžiui, su mano modeliu, tai useriui kitam būdavo trys dienos, tada penkios dienos, tada dvi savaitės, tada išjungiam pilnai. Tada vėl grįžta prie to seno, keičiam raidžių dydį. Nu ten nu nuo to ką tu ten backende, algoritme, dar pridėdavom ir Frontendo, ir ten ir tu ten džiaugiesi, o kaip matosi metrikose, bet aš tai nežinau. Turbūt patirtis naudotis app'su, kur ten ryte dar vienaip buvo, vakare jau kitaip, turbūt nėra gerai. Ir jei tu pasakytum, man čia jau enough, kaip tą padaryti? Turbūt yra labai sunku padaryti. Mano nuomone, būtent čia. Kas turbūt irgi iškelia tokių klausimų, kad aš negaliu būti šiaip laisvai. Bet jo įdomu, kokia čia reguliacinė. Į šitą negalėčiau atsakyti. Aš tik esu girdėjus, kad kaip ir turėtų būti galimybė būti nebe testuojamiems turbūt. Nežinau, kiek tai yra griežtai reguliuojama. Bet kaip tą padaryti? Aš negalėčiau atsakyti. Ir nesu girdėjusi tokių labai atvejų, kad mes sakytume va, dabar jau dabar eksperimentavimo platformą mes išimame. Gal aš nežinau, galbūt legal po to susisiekė ir tada eksperimentavimo platformoj kažkaip mes tuos userius išimame, bet aš spėju labiau, kad neveikia taip. Tai va, turbūt šita vieta irgi yra truputį apleista. Dar kol kas.

I.: Ar koreliacija vis dar nelygu priežastingumui?

P.: Čia klausimas statistinis ar bendras?

I.: Šitą visiems užduodu.

P.: Vis dar nelygu. Aš norėčiau pasakyti, kad vis dar lygu. Deja, aš sakyčiau. Engineer čia neseniai čia, prieš porą dienų product menedžeris sakė jis parodė koreliaciją ir sako todėl turim tą daryt ir tada jam sako. Bet koreliacija nelygu priežastingumui. Ir labai susipyko žmonės. Tai toks ir turbūt atsakymas. Turbūt sunku dar šitą išimti iš žmonių galvų. Ti padarai koreliaciją ir tu manai, kad tai ir yra priežastis. Ir man atrodo, kad čia yra turbūt vienas sunkiausių dalykų, kurį sunku išgyvendinti. Ypatingai, jeigu

galiausiai tavo output'as iš data žmonių atsiranda kitų rankose, kurie nu ten žino tas sąvokas, bet vis tiek yra tas polinkis padaryti išvadas ir jos yra labai stipriai. Taip taip ir daromos. Tu visada ten žvaigždutes parašai, kad ten tipo, tai nereiškia, kad susiję, bet žmonės tada. O, čia žiūrėk šitas ir šitas važiuoja. Viskas, varom. Tai manau, kad tai yra. Vienas sudėtingiausių dalykų šitoj data srityje, ypatingai turint omenyje, kad tu galiausiai atiduodi kažkam tas išvadas ir jas. Tiesiog koreliacija yra tai, kas žmonėms suprantama daug labiau negu priežastingumas ir pan. Tai va.

I.: O tai tada ką daryt, ką kitaip daryt, kad neišnaudojant koreliacijos kaip priežastingumo. Bet kaip tada panaudot tą informaciją?

P.: Nu geras klausimas. Aš manau, kad tiesiog. Būtent ką mes darom? Turbūt (anonimizuotas kompanijos pavadinimas) mes. Nes aš esu dirbusi įmonėse, kur eksperimentų iš vis nebūdavo. Eksperimentuose daug labiau gali pamatyti, kad čia ir buvo tas priežastis ar ne, tavo pakeitimas, nes tu taip pasižiūri ane iš analizės - nu čia koreliuoja kažkokie dalykai ir jau tu sakai, kad tai dėl to turbūt ir sales'ai auga dėl to, kad ten kiekis search'ų išauga. Nu gerai yra. Pasirun'inam ten tarkim eksperimentą, kur user'iai, juos paskatini daugiau search'int. Niekaip žiūri neįtakoja šitų sales'ų, ten daugiau atsiranda frikcijos point'ų, kurie jie ten turi pasearch'int, pafilter'int - nu niekaip nesusiję. Tai čia yra lengviausias, mano nuomone, būdas ir visa ta bendruomenė, kuri yra ne data žmonių supažindinti, kad ne visada tai, kas koreliuoja, galiausiai yra toj.. priežastingumas lygiai tas pats. Tai man atrodo, pas mus tas tas ir padeda. Dėl to mes stengiamės atsisakyt vien tik tų analizių, nes aš esu buvus didžiumoj įmonių. Tokio dalyko kaip eksperimentas nebūdavo, tai čia visiškas wild west - paimi du kintamuosius, ten kažkas važiuoja. Viskas, darome produktą jau ant to, nes ten matome, kad tas ten. Nežinau, kiek jis ten turi view'ų, lemia sales, o čia nėra, nėra tik taip susiję. Tai va ir tu labiau tą priežastingumą, mano nuomone, per eksperimentus, o eksperimentų kultūrą gerinant. Pas mus tikrai ji labai stipri. Tikrai susibuild'ino ta kultūra ir žmonės, kurie yra ne tik data, labai jau dabar yra, sakyčiau, apsišvietę. Tai čia paprastesnis toks įrankis, kaip apsisaugoti nuo šitų, mano nuomone.

I.: Kuriuose development procesuose matytum daugiausiai grėsmių? ML kokybės užtikrinimui?

P.: Čia turi omeny, kad tarkim jeigu duomenų panaudojimas ar ne ir tu ten įtrauki duomenis, kurie jau iš karto žinom, kad yra labai diskriminaciniai. Čia apie tą tu kalbi?

I.: Na, pavyzdžiui, toks ir kad etikos, bet, pavyzdžiui, nebūtinai net etinis. Tiesiog kažką nepadarai. Dėl to algoritmas predict'ina prastai, pakankamai žemą turi tikslumą.

P.: Tai mes va ką. Ką čia bedirbant esu pastebėjusi, kad, pavyzdžiui, limituotas kiekis duomenų, kurie tik atskleidžia tam tikrą angle'ą dalykų, kuriuos tu jau iš karto nusprendei, kad mes, pavyzdžiui, tikrai žinau patirties ir pati esu padariusi, ir kiti, kur. Tu jau iš karto nusiprendi, kad žinai. Nu vat šie

veiksniai lemia tą. Tai tu tiesiog aptrain'ini būtent į šitą profilį ir kai tu nepaimi daugiau kintamųjų, daugiau tų dėmenų, kurie galėtų pasakyti, kad čia nenuvažiuotų į vieną pusę, į tokį klasifikavimą. Tai irgi yra svarbi dalis. Turbūt tas išankstinis klasifikavimas duomenų neretai gali būti problema, kur iš karto tu tarkim su klasterizuoji ir tu jau paduodi grupes, kurios tada jau tu, nepaduodi tiesiog kartais labiau sakyčiau tokio raw input'o. Ir pabandyt pirmiausia tada pažiūrėti, koks yra predict'inimas. O jeigu iš karto tu apsisprendi tas klases, tu gana suriboji šitą galimybę geresnį output gaut ir neapriboji tai, kad modelis veiks tik tais vienam dalykui ir kuris po to degrade'ins laikui bėgant. Nes jis nebus matęs kažkokios tai klasės ir pan. Tai turbūt viena iš šitų yra dalių.

Kita turbūt yra dalis sakyčiau. Kas yra svarbu, kai dirbi ar privačioj įmonėj ar pan., kad galiausiai ne visus duomenis kaip pasakysi, galėsi vėliau gauti ir naudoti. Kartais būna iš to įdomumo. Nu davai, tarkim imam kokį image search ar kažką, kas čia irgi buvo. Kur ten user'is įkels nuotraukas, aš ten supredict'insiu ir tu ten supranti, kad tau veikia tik limituotam kiekiui, kad tu vėl turi tada su legal suderinti. Ir tas tavo modelis yra toks, kur galiausiai sako. Nu tai gal galima ten tik vieną nuotrauką, kuri ten tokio tokio žinai, kurioje neatsispindi kažkokie dalykai ir tu pamatai. Nu jo, tavo nepanaudojamas modelis, nes jis neatitiko tų reikalavimų, kuriuos tu nusižiūrėjai. Tai mano nuomone, kad pradinis step'sas, kur tu turi - kokie tuos duomenis tu naudoji, jeigu tai yra kažkokie nauji, kuriuos tu nežinai, ar tu galėsi galiausiai naudoti, ar tai yra, pavyzdžiui, nuotraukos ir pan. Tu turi sužinoti, ar tu tavo legal tokius requirements įgyvendina, ar tu galėsi išvis jį ten deployt'int ir atitinkamai naudoti. Kitu atveju, be abejo, tada tampa nelabai panaudojami šituo modeliu arba jis panaudojamas tik paeksperimentuoti, pasižiūrėti. Pas mus taip ne kartą irgi buvo. Ir po to mes jį turim išjungti, nes tu supranti, kad tu nesusižiūrėjai šitų dalių ir tik po to išsiaiškinti, kad tipo negalėsi, tokių duomenų saugoti, negalėsi. Neturi net capacity galbūt išsisaugoti tokius duomenis arba juos process'int teisingai. Bet tada supranti, kad jau čia biški reikėjo anksčiau tą tą išsiaiškinti. Tai gal nežinau, tokia šauna dabar mintis, daugiau nelabai sugalvoju.

Informantas Nr. 6

Amžius: nežinoma (apie 30 m.)

Lytis: moteris

Darbo vieta: duomenų mokslininkų komandų vadovė, 10 m. patirties

Gyvenamoji šalis: Lietuva

Pilietybė: Lietuvos

Formatas: nuotolinis pokalbis

Trukmė: 54min 13s

Data: 2024 m. gruodžio 10 d.

I.: Esu Simonas Bansevičius, politikos ir medijų magistro studentas. Darau tyrimą apie mašininio mokymosi programų etiką. Informuoju, jog bet kada gali nuspręsti nutraukti pokalbį. Ar sutinki, kad šis pokalbis būtų įrašytas ir jo medžiaga būtų naudojama tyrimo tikslais?

P.: Sutinku.

I.: Tai pirmasis klausimas. Prisistatyti kiek norisi. Naudinga, jeigu norisi, pasakant savo amžių, ką dirbi šiuo metu, kiek laiko šioje srityje. Ir aplamai apie komandą, kokio dydžio, kiek žmonių?

P.: Tai aš, dirbu su duomenų mokslu ir mašininio mokymosi algoritmais arba dirbtiniu intelektu 10 metų. Pradėjau dirbti kaip duomenų mokslininkė, pati kūriau prognostinius modelius ir po to tapau vadove. Prieš kokius septynerius šešerius metus vadovavau duomenų mokslininkų komandoms. Buvo įvairios tos komandos. Buvo komandos iš kelių žmonių sudarytos. Buvo didesnių komandų buvo po to tapau duomenų strategijos vadove. Ten jau vadovavau departamentams, kur buvo duomenų inžinerija, duomenų mokslas, duomenų analitika, t.y. viskas įmonėse, kas susiję su duomenimis. Ten būdavo iki 30 žmonių. Departamentai, tai dirbau didžiąją laiko dalį su finansais. Banke, o po to dirbau startup'uose. Tai maždaug tiek.

I.: O dabar?

P.: Dabar nedirbu, nes use atostogose. Bet kuriu startup'ą, tai šiek tiek dirbu, užsiimu visuomenine veikla - mentoriauju.

I.: O paskutinėj darbo vietoje maždaug kaip atrodė darbas, kurį teko daryti.

P.: Tai aš vadovavau kažkur 30 žmonių komandoms. Tai. Buvau vadovų vadovė. Tai buvo komandos duomenų analitikos. Buvo komandos duomenų mokslo. Tada buvo komandos duomenų inžinerijos ir tada buvo tokios specifinės komandos, kur, pavyzdžiui, marketingo analitikai ir duomenų mokslininkai, finansų analitikai. Tai tokių komandų gal kokios šešios, septynios buvo ir aš tų komandų vadovų vadovams, vadovų vadovė buvau. Ir tai tikslas yra departamento vadovo sudaryti strategiją. Nes kai aš atėjau į organizaciją, buvo tenai apie dešimt analitikų. Ir jie tiesiog norėjo pradėti daryti duomenų mokslą ir plėsti analitiką. Nebuvo visai duomenų inžinerijos, tai mano strategija buvo pirma sutvarkyti duomenų inžineriją, kad ji atsirastų, nes reikia dėti pagrindus tam, kad būtų galima dirbtinio intelekto produktus vystyti. Tai atsirado duomenų inžinerija. Tada išsigryninom, kas yra analitika, kur jos reikia,

kur nereikia ir kur reikia dirbtinio intelekto specialistų, t.y. duomenų mokslininkų. Ir tada samdžiau. Tai yra strategijos sudarymas ir jos vykdymas. Kaip, kur, ko reikia. Tai va, toks darbas buvo.

I.: Kokie didžiausi iššūkiai šiame procese buvo?

P.: Iš tikro nelabai daug iššūkių buvo. Tai iš tikro vadovaujamame darbe tai visą laiką žmogiškieji iššūkiai dažniausiai būna. Kaip suderinti kažkokius darbus, kaip. Kaip sudėlioti prioritetus? Nes, tarkim, jeigu bendrai kalbant apie dirbtinį intelektą ir aš gal šiek tiek plačiau kalbėsiu, nes aš dar ir konsultavau kažkiek įmonių norėjau sakyti nemažai. Kažkokį tai šiek tiek plačiau norėčiau atsakyti, nes bėda su dirbtiniu ir. Ir gerai darbuotojams. Ir bėda yra ta, kad dirbtinis intelektas įmonėse tampa kaip servisas, t.y. dažniausiai yra kažkokia komanda. Ir tada, jeigu kažkas nori kažkokio dirbtinio intelekto produkto, aš vadinu produktais iš tikro. Nu tai ten sistema, kaip kai kas vadina, bet aš vadinu produktu dėl to, kad aš visą laiką kalbu, kad tai yra truputį plačiau negu negu tik kodas. Tai va, jeigu nori kažkokio dirbtinio intelekto produkto, ateina pas tos komandos vadovą ir ten tariasi, sako mes norime to. Tada, tarkime, iš finansų ateina, sako mes norim to ten iš marketingo ateina, sako mes norim to iš operacijų ateina, sako norim to. O tarkim yra 5 žmonės, o reikia 10 žmonių. Ir tada reikia sudėlioti ir nuspręsti kas gausis. Kažkas kažką pirmą, kažką antra ir kažkas trečias, o kažkas išvis nieko negaus. Tai čia iš tikro yra iššūkis visada todėl, kad yra ribotas biudžetas, kas yra gerai, kad yra ribotas biudžetas. Ir šiaip vykdant jau tuos dirbtinio, kuriant dirbtinio intelekto produktus, tai didžiausias iššūkis visą laiką yra Lūkesčiai. Tikriausiai, kad dar žmonės nelabai supranta, kas yra tas dirbtinis intelektas. Intelektas versle ir jau tada turi lūkestį, kad kažkas, prisėdęs prie magiško rutulio jiems išburs kažką ir tada jiem išvis nieko nebereikės daryti ir viskas savaime susitvarkys. Ir tada, kai pradedi pasakoti, kad čia jums reikės įdėti darbo, kad čia iteratyvus procesas ir taip toliau. Tada tas toks lūkesčių valdymas yra labai didelė dalis. Po to būna dar ir dar kita dalis, kaip pavyzdžiui, pristatai modelį kažkokį dirbtinio intelekto ir jeigu tai liečia, ypač jeigu kokius finansus liečia. Ir kažkam kažką. Aš visada sakau, kad dirbtinis intelektas yra veidrodis ir parodo, kas buvo daryta. Bet tada jau parodo be asmenų, taip nuasmenintai. Ir tada būna, kad čia nesąmonė. Taip negali būti. Tai tokie žmogiškieji resursai, ištekliai ir taip toliau. Ta prasme ne, tikrai ne techninės problemos. Šiaip bendrai su dirbtiniu intelektu. Kad projektas būtų neįgyvendintas dėl, tarkim, finansų ar ar kažkokių ten techninių ypač dalykų, nu niekada nesu girdėjusi dėl techninių dalykų. Visą laiką būna kažkokia. Kažkas susipyko, kažkas nepasidalino, kažkas dar kažko. Labai tokie buitiniai dalykai būna.

I.: Aha. Aš turiu iš to sekančių papildomų klausimų. Pavyzdžiui. Sakei, kad dirbtinis intelektas tai plačiau nei vien tik kodas. Tai ar gali išsiplėsti ties šia mintimi?

P.: Tai dirbtinio intelekto Sprendimai. Jie iš principo automatizuoja ir įgalina padidinti. Nu šiaip realiai, automate and amplify, jeigu angliškai, tik tai lietuviškai aš bandau kalbėti, tai jie realiai jo automatizuoja ir tiesiog padidina tai, ką daro įmonė, t.y. nu jie gali tai padaryti, jeigu jie veikia. Ir kai mes kalbame apie sprendimus, tai yra ta dalis, kur yra kodas, kur tiesiog mes galime padaryti, kad veiktų. Nu, tarkim, sako pavyzdys: prognozuoti kitų metų biudžetą ir jisai suprognozuos kitų metų biudžetą, bet jeigu sakys, kad dar reikia atsižvelgti į tai... Nu ir tarkim ta prognozė, gali būti, kad jis bus toks pat kaip praeitų metų. O įmonė auga ir. Skaičius, tarkim, vartotojų, auga ir taip toliau. Ir gali būti daug niuansų, kad, pavyzdžiui, planuojama restruktūrizacija įmonės ir taip toliau. Tai tam, kad dirbtinis intelektas neštų naudą, tai turi būti labai gerai su integruota, su vidiniais procesais. Visų pirma tam, kad mes gautume daugiau konteksto, daugiau medžiagos. Antra, tam, kad iš tikro naudotų to dirbtinio intelekto sprendimus, produktus.

Nes ką aš matau dažnai. Tai vėl grįšiu prie to, ką kalbėjau, kad būna labai sukelti lūkesčiai. Čia gal dabar aš jau truputį kalbu iš tų įmonių, kurias teko konsultuoti, kad, pavyzdžiui, mes ateinam. Dažniausiai su vyru konsultuoju arba šiaip dėl to sakau mes, kartais ir viena. Bet žodžiu, būna taip, kad mes susitinkame su kažkokiu žmogumi ir tada sako mums reikia dirbtinio intelekto sprendimo ir jie tada tokie. Ne nu tai čia dirbtinis intelektas, tai čia savaime suprantama, kad bus sėkminga. Čia mes viską pakeisime, čia bus labai inovatyvus, gausim labai daug pinigų. Mes čia netaupome pinigų, galime viską sau leisti ir tada, kai aš jiem praded aiškint, nu žiūrėkit, jūs ten pataupykite, nešvaistykite pinigų. Čia nėra stebuklas. Čia yra juodas darbas. Jums vis tiek reikės padaryti. Jums reikės padaryti, kad veiktų. Ir tada būna "jo jo, suprantu, bet čia mums reikia pačių geriausių specialistų. Čia mes biudžetas neribotas, čia viskas neribota". Žodžiu, greitai tik tai darom čia. Hmm... Tada po pusės metų bankrutuoja. Tai kai yra tokie dideli lūkesčiai ir ateina žmonės ir pradeda dirbti ir tarkim sukuria kažkokį kažkokią versiją vieną ir jie tada būna, sako "ė, bet tai aš pats taip galiu pasidaryti". Tai taip gali pasidaryti, bet ta pirma versija kažkokio modelio. Ji ir turi būti tokia, kad žmogus dar galėtų suprasti tam, kad galėtų praeiteruoti, ir nu dar suvokti, kas yra gerai ir kas yra blogai, ir į kurią pusę tobulinti, nes tai nebus. Tai nėra panacėja. Tai nėra taip, kad iš karto sukuri kažką, kas sprendžia visas tavo gyvenimo problemas. Lygiai taip pat, kaip jeigu skauda gerklę, eini pas lor gydytoją, neini pas kažkokį gydytoją, kuris ten širdies, pavyzdžiui. Na ta prasme, kad kitaip tiesiog neveikia. Tai vat, kai yra labai dideli tie lūkesčiai ir nėra to tokio kultūrinio pokyčio organizacijoje, kuris galėtų priimti tuos, nes, pavyzdžiui, labai daug kartų esu mačiusi, ypač jo konsultuojamose jo įmonėse, kur, pavyzdžiui, sako, visą laiką reikia su pinigais kažką ten prognozuoti vieną ar kitokį elementą, bet viskas turi būti su pinigais. Ir prognozuoja, ir sako Ačiū, žinokit, jūsų

modelis neveikia. Nu neveikia, bet tai jūs pusę žmonių atleidot. Ir tada visą laiką reikia suvokti to dirbtinio intelekto ribotumą, nes lygiai taip pat, ypač versle, jeigu mes darome prognostinius modelius, tai žmonės, žinodami tas prognozes, priima tuos kažkokius tai sprendimus. Nu tarkim, aš gaunu duomenis, kažkokius, padariau prognostinį modelį ir matau, kad įmonei labai bus blogai. Jeigu jeigu jie elgsis taip, kaip elgiasi dabar, bus žiauriai blogai, tai aišku, kad jie kažką pasikeis, kad to išvengtų. Tai tiesiog tas dirbtinis, dirbtinio intelekto sprendimams priimti ir juos naudoti reikia labai didelio darbo iš vadovybės. Reikia labai gerų procesų apibrėžimo tam, kad apskritai ten viskas veiktų ir vaikščiotų, nes būna kur ten kažkur ten guli, pas kažką kompe kažkoks modelis. Bankas žinau, ten vienas apsiskelbė, kad jie ten viską dirbtinio intelekto daro. Pas kažką kompe guli ir niekas nesupranta kaip ten kas veikia. Nu tai žodžiu apie tai, kad kodas yra viena dalis, bet tas jo, aš skaitau, kad dirbtinio intelekto produktas yra tada, kada jis yra naudojamas, kad kažkas gauna naudą iš jo.

I.: Dar prieš tai irgi sakei, kad gerai kai ribotas biudžetas. Tai čia dėl tos priežasties, kad nebankrutuotų?

P.: Apskritai aš mėgstu sakyti, kad dirbtinis intelektas yra apie neapibrėžtumus, mokslas apie neapibrėžtumus, nes realiai tai yra kažkokia labai yra problema, kuri dažniausiai būna kažkokia labai bendrinė. Jeigu ten kažką privestume, tai ten mes norim žinoti, kas bus ateityje. Arba mes norim žinoti, kaip mūsų klientai elgsis ateityje. Realiai tas yra labai bendrinės problemos, kitaip tariant, labai neapibrėžtos problemos. Tada iš kitos pusės. Duomenys irgi yra labai neapibrėžti. Čia iš tos perspektyvos, nes jų yra mažai tokiai didelei problemai spręsti. Taip visada yra. Tarkim, kaip mūsų klientai elgsis ateityje, Bet mes turime tik savo klientų duomenis, kurie yra, Kuriuos visų pirma galima naudoti pagal GDPR. Ir dar taip, kaip jie elgiasi, tarkim, kaip mūsų klientai. Bet mes nežinome, kaip jie elgiasi. Tarkim, bankas. Ten turi duomenis, kaip klientai elgiasi, ar jie ten grąžina paskolą laiku, ar ne, bet jie nežino, gal ten, pavyzdžiui, šitas žmogus yra teistas, ar ne. Ten ta prasme vis tiek yra ribota. Ir kuo labiau apsibrėžia duomenų mokslininkai tą užduotį ir ją net nebūtinai susimažina, bet būtent apsibrėžia, kad aš man reikia šito, bet ne šito. Aš naudosiu šitus duomenis, bet ne šitus duomenis. Ir tada kablelis, nes mano problema yra tokia ir tokia. Ir mano kontekste yra taip ir taip ir taip. Nu tik va, dabar gal su prompt engineerig'u po truputį ateina tai, ką aš kalbėdavau duomenų mokslininkams, kad prieš pradedant daryt kažkokį kodą, jūs tiesiog vos ne ranka parašykit. Šiaip dokumentaciją visada liepdavau rašyti ir kas su manimi dirba, tai labai pamėgo, nes visą laiką apsibrėždavom, kad mes darom. Mūsų užduotis yra tokia ir tokia. Ir tada su verslu sakome ar jums tinka tiesiog raštu apibrėžimas. Ir ten koks būna 10 lapų aprašas apie ką yra modelis Ir kai mes apibrėžiame, tai tada labai lengva tai įgyvendinti, nes atsiranda kažkokios gairės, o kai to nėra ir tos problemos, jos yra labai bendrinės ir. Ir neaprėpiamos.

Tai ten yra keli keli tokie rabbit holes aš vadinu. 1 tai yra laiko, kada - ir čia labai tipinis - kai būna. Yra viena labai juokinga istorija, kai vieną modelį banke daro, net nežinau dabar, nes išėjau jau seniai iš banko, bet nenustebčiau, jeigu dar dabar bando tą modelį padaryt. Žodžiu, buvo taip, kad kažkas iš verslo užsinorėjo tam tikro modelio. Dėl NDA nesakysiu kokio, bet žiauriai paprastas. Įsivaizduokim, kad churn, panašus, tarkim churn. Ir bandė padaryti tą modelį ten ten vieni duomenų mokslai, viena komanda, kita komanda. Ir atkeliavo iki manęs tas modelis. Sako nu, čia pora komandų bandė, nepadarė. Tu pabandyk. Ir ką aš padariau? Aš visų pirma pasižiūrėjau, kiek banko klientų paliestų tas modelis. Ir aš sakau žiūrėkit, tarkim, jeigu kalbam hipotetiškai apie churn, tai įsivaizduokit, kad sakau tai jums čia 10 klientų parodys, kad 10 klientų per metus paliks banką. Ar jums aktualus toks modelis? Nuėjau pas tą vadovą. Žodžiu, sakau ar jums, ar jums vis dar reikia, jei jūsų toks rezultatas bus? Jis sako Oi, ne, ne, ne, sako, aš galvojau, kad daug daugiau. Na ir išėjo taip, kad aš išėjau iš tenai. Tada tas vadovas išėjo ir vėl jie iš naujo pakėlė tą modelį, ir vėl visiems reikia, ir vėl kažkodėl niekaip nepadaro to modelio. Ir tam buvo gal septyni metai darė. Na, tai čia aš galiu sakyti, kad tai yra ten tarkim laiko, bet čia yra čia tokia kuriozinė situacija, kurią aš galėjau papasakoti pavyzdžiu. Bet tokių situacijų yra labai labai daug. Pvz. modelį daro tris metus, keturis metus ir žmonės nesupranta, kad arba jis jau yra padarytas ir jie ten jau dailina detales, kurios visiškai nekeičia esmės, arba jie tiesiog nepadaro, nes yra priežastis, kodėl nepadaro. Dar vieną žinau tokią liūdną istoriją, kai vieną labai protingą žmogų tiesiog atleido, nes jam davė tokią užduotį. Sako, na, išsiaiškink čia maždaug apie mūsų, paprognozuok vat mūsų klientų elgseną. O ten tiesiog yra toks pool'as klientų, kurie sako sako tai gal gal kažkaip ten segmentuojam tuos klientus - sako ne ne, tiesiog visų. Nu ir ten tiesiog neišsprendžiamas uždavinys yra ir sako va tu blogai dirbi, nes tu nepadarei šito. Tai jo viena yra tai, kad laike labai išsiplečia ir tada tiesiog neracionaliai ilgai būna daromi tie produktai ir nepadaroma. O kitas dalykas yra kaštai, nes, tarkim, jeigu mes žinom, kad turim x ten tūkstančių, tai tada viskas yra verčiama į fte, T.y. Darbuotojų skaičių. Ir tas darbuotojų skaičius ten tarkim yra du darbuotojai pusę metų ir tada labai gerai galima pasakyti. Žiūrėkit, jūs galite padaryti tik tai, ar jums vis dar reikia. Tai vat čia labiau apie tai, nes kai yra neribotas biudžetas, tai tikrai. Nesu mačiusi gerų rezultatų su neribotais biudžetais. Mačiau, kaip įmonės bankrutuoja su neribotais biudžetais.

I.: Papasakok apie procesą, kaip galvoji apie naudotoją, kokiose situacijose. Apie galutinį naudotoją programos.

P.: Čia apie dirbtinio intelekto, bet kokio produkto?

I.: Jo jo jo.

P.: Procesą iš tos pusės, kada jis paliečia?

I.: Kaip vyksta galvoje mąstymo procesas. Kaip, kada galvoji apie naudotoją, kokiomis aplinkybėmis?

P.: Ai, pala, tai dabar aš dar kartą patikslinsiu, kai yra kuriamas dirbtinio intelekto produktas, kada yra galvojama apie vartotoją?

I.: Jo

P.: Tai visada. Aš galiu pasakyti, nes čia šiaip pakankamai mano specializacija šitoje vietoje yra tai. Kaip ir kalbėjau, yra apibrėžiama pati užduotis. Ir viena iš apibrėžimo formų. Tai yra tas vartotojo interfeisas. Ką gaus vartotojas? Tai tarkim, kur aš pasakojau, kad ten mes pradedame daryti kažkokį tai dirbtinio intelekto produktą? Aš pati galiu papasakoti. Aš naudoju dizaino sprintus visada. Tai yra Google Ventures parašyta knyga Design Sprint. Ten 2 2, kurie dirba Google Ventures. Čia realiai ši metodologija gal kokiais 2014 metais, nes aš pradėjau naudoti 2015, bet buvo ką tik išleista knyga. Tai žodžiu nepamenu pavardžių. Tai Design sprint vadinasi, po to mačiau net labai juokingai, nes labai daug kas... Aš tą dizaino sprintą prisitaikiau dirbtinio intelekto produktams ir dabar mačiau net veda mokymus ten dizaino sprintas dirbtinio intelekto produktams. Pagal tą pačią knygą. Tai vat aš tą naudoju nuo kokių 15-16 metų. Tikslas yra toks, kad pirmą dieną susėdome su verslu. Jie pasako problemą. Tada duomenų mokslininkai ir aš ten. Aš tuo metu būdavau vadovė. Mes pasitariame iš techninės pusės ar įgyvendinama, ar neįgyvendinama, tada kokių duomenų reikia. Ir tada mes pradedame kalbėti OK, tai iš techninės pusės mes, tarkim, padarom modelį. Bet kokie tie, pvz. filtrai, kurie bus matomi klientui ir kodėl? Ką matys klientas ir kodėl? Tai čia tas toks ne iš dizaino pusės, bet iš tos esminės pusės ką matys, tai jau yra aptariama pirmom dienom, kai yra pradedama galvoti apie problemą.

Ir tada grįžtame pas verslą ir sakom žiūrėkit, jeigu norime, kad vartotojas matytų tą ir tą, tada mums reikia tokių ir tokių duomenų, nes dažniausiai verslas žino, pas ką yra duomenys. Jeigu didesnė organizacija, jeigu ten maža įmonė, tai tada visi daugmaž bendrai žino. Tada, aišku, reikia pasitarti. Tarkim, GDPR, dirbtinio intelekto aktas, reguliacijos ir taip toliau. Ar ten įmanoma gauti tuos duomenis ir tada pradedamas kurti modelis. Bet tiksliau net ne modelis, o tas proof of concept. Tai yra POC konceptas modelio, kurio tikslas yra parodyti, ką matys vartotojas, bet visas backend'as jis sufake'intas gali būti. Pvz., modelis gali veikti ant trijų klientų, bet esmė, kad jis veiktų ir kai sukuria tą POC proof of concept, tai tada duodam verslui arba vartotojams ir tada vartotojai išsitestuoja ir sako jo man tinka arba netinka. Tai pirminė modelio versija turi įvykti kuo greičiau. Mes duodame tam savaitę. Tarkime, mes pirmadienį pradedame, o penktadienį duodame vartotojams. Aš sorry, visą laiką sakau versle, nes mūsų vartotojas verslas būdavo dažniausiai tai. Tai čia šitoje vietoje reikia panote'inti. Ir tada,

penktadienį, tarkim, penktadienį, mes duodame savo vartotojams ir vartotojai sako ok, mums šitas tinka. Bet, tarkim, mums reikia dar tokio ir tokio feature'o. Arba mums tas netinka. Tai šito vartotojo įtraukimas nuo pat pradžių. Jis yra tam, kad kuo greičiau nukill'inti projektą, jeigu iškart matome, kad nesigauna. Nes vėlgi vartotojas gali įsivaizduoti, kad ten labai daug naudos atneš. Bet jeigu mes pasiėmėme duomenis ir žiūrim, čia yra taip ir taip. Mes jums galime parodyti arba prognozuoti tą ir tą. Ir tada sako Ai, nu bet taip, tai čia mums daugiau naudos nėra iš to. Ir tada ačiū viso gero, Bet ir jam gerai, ir mums gerai, ir pinigai sutaupyti įmonei tai ten visiškai win win būna.

Aš visąlaik juokauju, kad duomenų mokslininkų tikslas yra nedaryti modelių, nedaryti to, nedaryti modelių ir paaiškinti, kodėl jų nereikia daryt. Kitas dalykas yra vat vėl apie tą neapibrėžtumą. Dažniausiai, kai ateina ir sako mums čia reikia prognozuoti tą ir tą. Ir realybėje būna taip, kad mes apibrėžiame. Tarkim, per pirmas dvi dienas sėdime kartu su savo klientais, apibrėžiame užduotį, kas jiems yra svarbu ir tada realiai būna kokį 70% arba aštuonis. Jeigu nu ten imam ten tą Pareto taisyklę, tai tarkim 80% užduoties būna labai lengvai įgyvendinama, tai mes pasiimam op ir per savaitę tą POC tikrai labai lengvai padaroma. O tas 20% tai būna. Nėra duomenų, ten kažkas kliūna, kažkur kažko trūksta, kažkas negerai ir tada. To Proof of concpet tikslas yra, kad kas nepasidaro greitai, mes to ir neimame. Out of the scope visą laiką atsimėtom, atsimėtom padarome tai, kas lengva. Ir triukas ką šita, ką aš pastebėjau praktikoje tai, kas lengvai pasidaro, tas ir virsta tuo galutiniu produktu. Nes tai, kas nepasidaro, tai reiškia, kad jau yra problema. Ir dažniausiai kur atrodo. Ai, čia turėtumėm duomenų daugiau ir padarytume. Čia tikrai nesunku ir dažniausiai neatsiranda ten duomenis arba ten, jeigu kažkodėl prastai prognozavo kažkokią vieną šalį, tai ir neprognozuoja tos šalies. Kažkokia problema ten yra su ta šalimi.

Na, žodžiu, šitas reikalingas tam, kad vėl apsibrėžti ir išsigryninti tą užduotį ir tada vartotojai penktadienį, tarkime, išsitestuoja tą Proof of Concept ir tada kitas. Tada mes pereiname į tą kūrimo mode'ą ir kai kuriam tai. Tada mes pasiimame kassavaitinius susitikimus su vartotojais ir kiekvieną savaitę. Ir nesvarbu. Ir čia šitas duomenų mokslininkams yra žiauriai skaudu ir šito nekenčia, kad kiekvieną savaitę parodyti. Nesvarbu, ar ten visą laiką jaučiasi, kad nieko nepadariau arba čia nesąmonė. Čia ne galutinis variantas, aš parodysiu, bet visą laiką parodyti. Ir kai jie parodo kažkokį variantą, tai tada įvyksta ir tas kultūrinis pokytis. Tada įpranta vartotojai, kad aha, čia bus taip ir taip. Ir aš tada esu tame. Ai, o šitas atsirado, nes taip ir taip. Ir tada tokie esminiai sprendimai kartais būna darai, darai ir sakai žiūrėkit, bet čia gal geriau taip padarom. Ir jeigu verslas, vartotojai nedalyvautų tame, jie sakytų Ne, ne, nesąmonė,

tikrai nedarom. Bet kadangi jie yra tame pačiame procese, tai jie tada kažkaip sako Ai, nu bet tikrai taip yra geriau. Nu ir po to tada su tais kas savaitiniais susitikimais stengiamės. Priklausomai nuo projekto dydžio, bet per pusmetį ir iki metų pasibaigti. Jeigu kažkoks sprendimas trunka ilgiau negu metus, tai iš karto sakom čia turi būti 2 3 modeliai. Idealu, jeigu tai per pusę metų ir yra uždaromas tas projektas. Tai va, realiai, bent jau taip, kaip aš dirbau, tai vartotojai nuo pirmos dienos ir po to kas savaite. Nu ta prasme reguliarūs susitikimai ir parodymai nebaigto produkto vartotojams.

I.: Kokių turėjot darbe modelių vertinimo strategijų? AI vertinimo strategijos - kaip vertinate?

P.: Tai aš dirbau dažniausiai reguliacinėj aplinkoje. Tai būdavo reguliatoriai, tai būdavo keli keli etapai. Pirmas yra, kada mes padarom modelį, tai mes patys validuojam modelį. Tai validavimas. Čia gal tikriausiai neit labai į techninius dalykus?

I.: Manau galim eit visur.

P.: Nes validavimo tai ten priklausomai nuo užduoties tipo. Bet visą laiką yra du validavimo tipai, tai yra vienas yra validavimas techninis. Tai, pavyzdžiui, yra algoritmai, specialūs validavimui, priklausomai ar ten kokios laiko eilutės yra, ar ten yra kažkokio klasterinimo algoritmai. Tai yra tiesiog tas algoritminis validavimas, tada visą laiką yra tas techninis validavimas. Apskritai, ar ta sistema veikia, kiek ji stabili yra, tada visą laiką yra tas out of sample validavimas, kada pasiimam kažkokius 100 klientų ir pasižiūrim. Mes dar visą laiką darydavome tuos edge cases, kur žinome, kad kažkokios ten sudėtingi atvejai žiūrim, kaip veikia. Tada visą laiką vartotojai testuoja. Čia jau testavimas, analizavimas. Čia du skirtingi dalykai. Tai validavimas yra procesas, o testavimas kada mes jau atiduodam pravaliduotą modelį ir tada vartotojai tiesiog testuoja ir sako arba veikia, arba ne. Arba. Nu gerai, tas techninis testavimas dar patys darome. Tai yra, kad prieš pradedant daryti modelį 20 procentų duomenų atsidedame ir tada ant tų 20% kurių nematė modelis, tai nu žodžiu, tas klasikinis ml. Ir tada vykdavo audito analizavimas. Yra atskira komanda. Tai čia, kai banke dirbau, yra atskira komanda, kuri nieko nežino nei kaip buvo kuriamas modelis, nei kam, nei kaip. Į ją tiesiog ateina ir validuoja. Tai kartais būdavo taip, kad mes darydavome kokį modelį pusę metų, validuodavo metus. Arba būdavo ten trys mėnesiai ar, trys mėnesiai žiauriai greit čia skaitydavosi. O jei rimtai, pusę metų validuodavo tas modelis ir tada, jeigu tai būdavo modelis, kuris yra naudojamas tam tikruose procesuose, tai būdavo išorinė validacija. Tai yra reguliatoriai, tai finansų reguliatoriai. Jeigu tai yra bankas, tai ten dar kartą praeidavo. Žodžiu, labai labai daug esu susidūrusi su tuo.

I.: Galima dar išsiplėsti, kaip to audito validavimas atrodė?

P.: Jeigu vidinio, tai jie tiesiog pasižiūri procesus, jie, bet čia būdavo tik tai mūsų specifika. Jie techninės validacijos nelabai darydavo. Jie tiesiog pasiimdavo, žiūrėdavo konceptualiai modelį, ar jis

teisingas? Ar jis yra etiškas? Ar teisingai tarkim, pasiima kažkokį, nežinau banko biudžetą ir žiūri, ar nekeičia kažko keistai. Tarkim, didelėje apimtyje, dar visąlaik daromas stress testas. Stress testai yra, bet čia iš tos ne techninės pusės stress testai, kurie irgi šiaip daromi yra, irgi to nepasakiau. Patys darydavome stress testus. Na, ten galima įvairiai daryti, ten uždėti ten 5 milijonus klientų ir t.t. Bet jie darydavo stress testą, kad jeigu ten palūkanų normos krenta tiek ir tiek, ar jūsų modelis dar veikia. O jeigu visi klientai tampa labai turtingi? Ar jūsų modelis dar veikia? Tai o jeigu pandemiją, tai mes tą patį darydavom, modeliuodavom. Tai va, tai tokius stress testus daro ir tada konceptualiai. Ar modelis yra teisingas, ar etiškas? Ar rezultatai stabilūs laike, ar patikimi? Ir kaip? Ir bando tiesiog surasti kur? Kur? Kur gali kažkokios žalos padaryti modelis ir jie ten labai ilgai darydavo. Bet ten sakau, ten yra ir dokumentacijos rašymas, kiekvieno modelio validavimas ir ten dokumentacija būdavo. Nežinau, gal 40 lapų tada yra patvirtinamas komitetas rizikos, patvirtina, kad tas modelis praėjo validaciją. Validacija, beje, turi būti tada, tai pirminė validacija taip ilgai trunka ir po to kasmet tie modeliai yra iš naujo validuojami, bet ten jau toks tada standartizuotas procesas. Ten iš principo peržiūri, ar tas modelis vis dar aktualus yra, ar vis dar geri rezultatai. Perkalibruoja tą modelį, žiūri, bet nu dar ką irgi ten tas auditas daro, tai žiūri kokie duomenys ateina. Input output ir realiai žiūri. Tai va, tai įdomus, beje, labai procesas ir man atrodo, kad šiaip turėtų tie modeliai būti validuojami bet kur, nes ten taip būdavo iš poreikio, nes tai yra reguliacinė aplinka. Bet, tarkim, kiek aš ir kitur dirbau, aš manau, kad atneštų labai daug naudos validacija modelių.

I.: O kodėl?

P.: Nes prisikaupia nesąmonių patys nesupranta kodėl. Nu gerai, kad tarkim jeigu yra, jeigu yra įmonė, kurioje dirba kažkokie patyrę, atsakingi specialistai, tada yra gerai. Irgi vienas konkretus pavyzdys visai neseniai. Viena įmonė organizacija ieškojo, kad kažkas sukurtų dirbtinio intelekto sprendimą sekti jų darbuotojams. Tai man atrodo... Okey jeigu kažkas dirba, kas pasakys, kad nekurkite, nes nu ir neetiška, ir nelegalu man atrodo. Nežinau. Bet, tarkim, kai kuriose šalyse tai yra nelegalu tikrai. Tai bet yra tokių įmonių, kurios sako davai važiuojam. Tai vat kitas dalykas, kada modeliai yra čia toks. Tarkime, netgi tame pačiame banke, kur dirbom, ten buvo nežinau gal kokios dešimt duomenų mokslininkų komandų ir kiti labai nemėgo tos validacijos. Tai aš visąlaik savo darbuotojams sakydavau, kad validacijaį jus išmoko daryti modelius tvarkingai. Ir tokius modelius, kurie veikia ir neša naudą. Ir jie labai tai perėmė. Ir kažkaip mes labai gerai sutariame su tomis validacijos komandomis. Šiek tiek erzindavo, kad jie taip lėtai viską daro, bet čia buvo vienintelis minusas, nes kiti tai pjausdavosi ten, pykdavo, ten jeigu kažkokią gauna, taip vadinasi remark'us. Jie labai kažkaip asmeniškai priimdavo. Bet išmoko, šiaip aš pasakyčiau, kad tie duomenų mokslininkai, kurie dirbo. Bent jau kur aš žinau, kur kur dirbo ir kur modeliai buvo analizuojami. Jie labai varkingai dirba, jie susi dokumentuoja. Jų kodas labai

tvarkingai, nes jie žino, kad jiems po metų reikės grįžti prie to kodo. Tai aš sakyčiau, kad tai įneša tokios disciplinos tvarkos ir bendrai tai labai pakelia kokybę.

I.: Aha, o kai sakei, kad žiūrėdavo, ar modelis teisingas, ar etiškas, pagal ką jie nuspręsdavo, ar jis teisingas, ar etiškas?

P.: Tai čia labai priklausydavo nuo užduoties. Visų pirma žiūri, ar tai yra... Nu kokie duomenys naudojami, ar nėra kažkokių netyčia naudojamų duomenų, kurie, pavyzdžiui, būtų lytis, amžius ar dar kažkas. Tiesiog peržiūri. Tai čia tas paprasčiausias. Po to žiūrėdavo, ar nediskriminuoja ten kažkokios grupės žmonių, kas pvz. gali būti vyresni žmonės? Na, gal pavyzdys būtų čia. Jis nėra tas pavyzdys, kokį aš esu mačius, bet tiesiog sugalvojau dabar. Daromas yra pricing'o modelis, kainodaros modelis ir pasiūlyti, tarkim, vyresniems žmonėms didesnę kainą. Nors, tarkim, jei modelis būtų. Modelyje pačiame nebūtų amžiaus žmonių, bet kažkaip jie ten nusprendžia, kad kad būtent jiems kažkas bus arba ten tarkim didmiesčiuose gyvenantiems, nes jų didesnės pajamos. Nes nu realiai dirbtinis intelektas. Jis linkęs, ypač tam tikri algoritmai. Jie yra linkę pasigauti tuos vadinamus Lazy predictors, kad jie patys labai greitai išmoksta, kad ai, šitas turtingas, šitas neturtingas arba šitas vyras, šita moteris yra, nors kaip ir tų atributų visiškai nėra įdėta. Tai vat ir tokius tikrindavo.

I.: Bet. Ar jie turėjo kažkokias apibrėžtas normas? Gaires?

P.: Ne. Tą vėlgi, aš čia gal labiau susiję su tuo, kad aš dirbau tokioje komandoje, kuri labiau research užsiiminėjo. Yra tokių komandų, kur darydavo taip vadinamus operacinius modelius, kur yra viskas žinoma, aišku ir iš metų metai ten nelabai kas keičiasi, tai tada ta validacija būdavo tikrai tokia standartizuota, kur tiesiog peržiūrėję vos ne tuos pačius tikriausiai klientų set'us turėdavo. Prasižiūri, prasižiūri ir viskas. O aš dirbau tokioje, kur būdavo tokie egzotiniai modeliai, bankui. Dėl to jie taip nestandartizuotai turėdavo žiūrėti. Kitas dalykas, kai aš dirbau ir dabar kokius daugiausiai pavyzdžius, kur paskutinį kartą su validacija dirbau. Tai tuo metu dirbtinis intelektas... Čia buvo kokie 2019, 2020, 2021 m., tai tuo metu nelabai dar buvo kažkokių. Tarkime, mes turime dirbtinio intelekto aktą, o tuo metu nebuvo kažkokio reglamento arba kažkokių net gairių. Tai ta reguliacija. Dirbtinio to mašininio mokymosi algoritmo, ji tokia buvo neapibrėžta, tai dėl to reikėjo kažkaip suktis iš padėties.

I.: Ir. Kuriuose. Kuriuose kūrimo procesuose būtų didžiausia rizika algoritmo kokybės užtikrinimui?

P.: Algoritmo kokybė. Aš gal sakyčiau, kad algoritmas yra. Algoritmai yra kokybiški. Jeigu juos naudoja ten, kur reikia, tai tada algoritmo parinkimo. Nes pats algoritmas nėra nei geras, nei blogas arba tinka tai problemai spręsti, arba netinka tai problemai spręsti. Yra aišku duomenų klausimas, ar tinkami

duomenys yra. Bet jo. Bet aš visiškai algoritmo nekaltinčiau algoritmo. Kaltinčiau tarpinę tarp kompiuterio ir algoritmo.

I.: O kas yra tinkami duomenys?

P.: Aktualūs duomenys tai problemai. Tai yra ne per daug duomenų. Nes visi sako ai, kuo daugiau duomenų, tuo geriau. Tikrai ne. Tai turbūt aktualūs duomenys. Ir iš kitos pusės. Tarkim, jeigu nori prognozuoti kitiems metams į priekį. Ir dar žiūrint ką. Bet jeigu kažkokia yra prognozė į priekį, tai tada turi būti istorinių duomenų pakankamai. Tada dar aišku, reikia atsižvelgti į tai, kad duomenys turi būti etiški, kad nepateikti kažkokių kažkokių atributų, kurie modeliui leistų diskriminuoti. Čia vėlgi galima būtų sakyti dėl algoritmo, bet algoritmas pats nediskriminuoja, tai duomenys implikuoja diskriminaciją.

I.: Toks įdomesnis klausimas - ar koreliacija vis dar nelygu priežastingumas?

P.: Aš manau, kad čia tokia. Ta prasme čia yra visą laiką, kaip ir su modeliais. Aš sakau, kad čia yra horoskopai, nes gali. Čia yra tikėjimo klausimas. Aš manau, kad ne. Aš manau, kad ne. Ir lygiai taip pat, kaip ir su modeliu. Jis parodo, tarkim, kažkokią kreivę, kuri eina į viršų, į apačią, bet ten vienas sako, kad žiūrėk, kaip auga, kitas sako žiūrėk, ten stovi vietoje ir. Ir jeigu žmogus nori tikėti kažkuo, tai. Tai tiesiog neįmanoma pakeisti. Nu bet tas, vėlgi ten irgi yra pasakymas, kad statistika yra didžiausias melas. Tikrai labai dažnai pasiima ir iš vienos pusės ir galima sumanipuliuoti. Ir matau, kad duomenų mokslininkai labai dažnai parenka tokius duomenis arba parodo, kokias ašis išdidina, arba. Giriasi. Po to vieną pavyzdį papasakoti juokinga, bet galima parodyt bet ką, bet kaip ir čia tiesiog arba tiki, arba netiki. Bet. Aš išvis koreliacijom net nu nelabai tikiu, nes tiesiog labai dažnai būna, ateina ir sako va, čia koreliuoja. Sako, žiūrėk, radau, koreliuoja. Sakau, o kas iš to? Nu nieko, šiaip, faina, koreliuoja. Bet ar tai paaiškina? Retais atvejais taip, tai paaiškina. Gali atskleisti kažkokią priežastį, kad, pavyzdžiui, kad kai lyja, gali sušlapti. Na, bet tai tos koreliacijos ir būna labai dažnai. Jeigu rimtai taip, duomenų moksle ką darom, tai tos koreliacijos labai dažnai būna tokios. Kur ten duomenų mokslininkai atranda - koreliacija! Nuneša bizniui parodo, nes duomenų mokslininkai būna, nesupranta to sudėtingo proceso ar ne. Aišku, dabar ten tie start ups, ten kur perku parduodu. Produktai paprasti, tai ten nėra to, bet banke tikrai būna žiauriai sudėtingi procesai, investicijos ten markets žiauriai sunkūs. Nuneša, sako va žiūrėkit, čia radau tas ir tas. Sėdi tie specialistai. Nu jo, tai mes tą žinome. Tai vat šitas.

Kitas dalykas irgi, visai palyginus neseniai susidūriau su tokia va kur vienas vadovas. Nu jam ten sunkiai sekėsi darbe ir jis tiesiog susigalvojo, kad ten vienas rodiklis turi kristi. Ir tada jis turi tapti to rodiklio gelbėtoju. Ir jis paprašė ten vienos mano komandos narės, kad jam padarytų tokį grafikėlį. Sako, man pavaizduoti ten X tas y tas. Ji pavaizdavo ir sako va, žiūrėk, krenta jis. Jis nuėjo pas CEO, ten padarė

darbo grupę, ten padarė tą emergency planą. Žodžiu, vos ne įmonė bankrutuos, jeigu ten tos problemos neišspręs ir taip toliau. Nu ir ten tokia drama didžiulė. Sakau, ten pirmo lygio įmonės problema ten, atrodo, milijonus praranda. Aš atėjau ir žiūriu, ir man kažkas yra tikrai ne taip tame grafike. Čia irgi beje su koreliacijom susiję. Ir aš visiškai kitaip žiūriu, žiūriu, žiūriu ir sakau, nu tikrai kažkas ne taip. Ir po to žiūriu, kad. Kaip ten tiksliai buvo, kad viena ašis, dvigubos ašys ir viena ašis nuo nulio prasideda, kita ne nuo nulio. Ir aš paėmiau kompe. Greitai padariau tas ašis, kad abi nuo nulio ir taip gražiai taip tiesė čia taip, kaip čia parodyta gražiai. Tarkim va žemyn ėjo, ir taip gražiai išsitiesina. Ir aš sakau žiūrėkit, nėra problemos, čia tiesiog nekinta. Šitas rodiklis nekinta metų metais ir tas žmogus toliau vargsta su ta grupe, ten įmonės gelbėjimo planu ir taip toliau. Nu ir aš sakau, čia yra religija. Tai žmogaus tikėjimo nepermuši jokia statistika. Jis gali ją panaudoti, įrodyti savo argumentą. Bet iš kitos pusės, jeigu žmogus yra įsitikinęs, nieko nepadarysi. Tai sakau tai man. Aš nesu koreliacijų ieškotoja.

I.: O tai kaip tada? Kaip rasdavot priežastis šitas, produktuose, kur būdavo?

P.: Vat būtent nebūtinai reikėdavo ir ieškoti tų priežasčių. Bet gal kas ir yra sunkiausia duomenų mokslininkams, kad ne visada reikia surasti priežastį? Nes aš visą laiką kažkaip gretina tą dirbtinį intelektą su medicina. Žmogus susirgo, tarkim, vėžys. Ir gali gydytojas pasakyti. Ten gali būti, kad ten vėžys, dėl to, gali būt genetika. Gali būt, kad ten mėsos valgėte. Gali būt ne tai, kad žmogui pasakys, kad žinai ten 30% tikimybė, kad tu susirgai nuo to ir to. Nu ar tai kažką pakeis? Ne. Tai tikrai nesakau, kad tai yra visiems atvejams. Kai kuriems atvejams reikia rasti priežastingumą ir galima rasti. Bet vėlgi iš mano patirties tai tikrai nebūna kažkokie Nobelio verti atradimai. Būna, kur aaaai! Toks būna - nu jo, aišku, jo čia dėl to. Bet sakau, ten kažkokių stebuklų ten labai daug neišburia. Šitoje vietoje. Vat sakau ir tas susitaikymas su tuo, kad ne visada reikia ieškoti priežasties ir galų gale, o jeigu jau jeigu gilinamės į tai nu tai beveik niekada nebūna viena priežastis. O jeigu būna penkios priežastys. Nu tai va dabar kaip teisingai attribute'inti, kuriai priežasčiai kiek laiko skirti ar ten 2%, ar trys, ar penkiasdešimt? Nu, ten tada prasideda tokių. Tai Analizę darom. Bet sakau kartais būdavo žmonės: nu kodėl čia taip? Sakau viskas - tu nebūtinai tai reikia žinoti.

I.: Tai tada. Kur tau asmeniškai prasmė yra šitame darbe?

P.: Man prasmė yra tame kūrybos procese. Todėl, kad. Galima sukurti produktų. Kurie tikrai padeda žmonėms. Pavyzdžiui, kas man yra aktualu? Man dėl to medicina labai įdomu. Ir taikymai, gal netgi tokie buitiniai, ten palengvinti gyvenimui, automatiškai užsakyti pirkinius, susidaryti mitybos planus ar dar kažką tokio, kur tiesiog nusiimti nuo savęs ir tą laiko dalį praleisti kažkur kitur. Bet čia aš taip, aš visą laiką. Jau seniai taip sakau, kad labai nereikia susireikšminti tiems žmonėms, kurie kuria tuos produktus. Vėlgi, sakau, čia nėra kažkokio stebuklo pasaulio. Tie modeliai, kurie yra sukuriami, tai

yra darbas, kaip ir visi kiti darbai. Ir tai, kad kol kas dar žiūri kaip į kažkokį stebuklą, į dirbtinį intelektą ir jo sprendimus. Tai kartais žmonės klysta tame. Tai man yra labai įdomu. Man yra įdomu dirbti su žmonėmis. Kaip vadovei man yra įdomu mentoriauti. Man yra įdomu tas kūrybinis procesas, kaip padaryt, kad susitartų, kaip padaryt, kad veiktų tie produktai? Nėra lengva padaryti, kad jie veiktų. Kad neštų naudą. Tai va tame procese.

Informantas Nr. 7

Amžius: nežinoma (apie 30 m.)

Lytis: vyras

Darbo vieta: įvairiose šalyse veikianti maisto į namus pristatymo kompanija.

Gyvenamoji šalis: Vokietija

Pilietybė: nežinoma, bet tautybė Kolumbijos

Formatas: nuotolinis pokalbis

Trukmė: 1h 0min 27s

Data: 2024 m. spalio 17 d.

I.: I introduce myself that my name is Simonas Bansevičius. I am a master's student studying politics and media. This interview will be used for my research about machine learning ethics. I inform you that at any time you can decide to end this interview. Knowing this do you agree with me recording this interview and using it for the previously mentioned research purposes?

P.: Yeah. I'm fine.

I.: Okay. for the first question, I'd like to ask you to introduce yourself, if you can, stating your age, where you work right now. What's your current profession?

P.: Yeah, absolutely. So one. I'm originally from Bogota, Colombia, and I came to and I did a bachelor in math and physics, and I came to Berlin to continue my graduate studies in math, and I did my PhD at Humboldt University, Berlin, in the area of something that is called geometric analysis and is more in the direction of pure math. And that was pretty fun. But after finishing my PhD, I decided I wanted to do something else, something more applied, and kind of understand the challenges of, of of the industry in the real world, so to say. And yeah, I got my first job in a marketing consultancy where I was faced to various types of projects and industries like automotive, retail and fashion. So that was quite nice. I was not a proper consultant, I was more like an in-house support team, working on the data science project. But of course I had to meet clients and that gave me like a nice sense of reality and business kind

of mindset. And after a couple of years there, I decided I wanted to move to a product company where I would really develop data products and I took a position at (anonimizuota), which is the meal box delivery company.

And that was March 2020. So I was let's say we didn't know about it, but I was supposed to lead the forecasting and efforts. There's a lot of time series analysis. But then the pandemic started and that was a bit of a mess, trying to do predictions in the middle of a pandemic that was very, very hard. But it was still fun. After a couple of years there, I decided to take a break of this forecasting kind of problems in the middle of the pandemic. And I joined (anonimizuota), where I still work there, which is a food delivery company that expand into Europe. And last year was bought by an American company. So it's quite big. And in my first part there, I worked in the marketing tech department, working in various project, kind of setting up the team, mainly in the area of a media efficiency. And this is where should we put our next euro in? In which channel is it? Facebook. Is it a TV ad and so on. And also customer lifetime value. Let's say now that we have acquired a customer, what's, let's say given their purchase history, how much do we expect this customer to be with us? And again in the food delivery company. It's not clear to churn, let's say, to leave, because you could just pause and maybe order a year from now or have a different purchase pattern. But we wanted to do this with just the purchase history. And since last year, I actually decided to come back to forecasting. So within the same company I'm now working in retail, which are like these market stores on which it sells kinds of bananas, fruits and diapers and so on. It has a big assortment, and I'm working in projects in between pricing. So how to price our items in an intelligent way and also forecasting. And again, kind of as you see, a lot of my work has been surrounding this area of of time series analysis, which I pretty much enjoy. It's quite classic but still very powerful. And I think every company at some point will will work with time series data. So that's kind of on my professional side. I also like since a couple of years they might be around four years. I've been very interested in open source projects, so contributing to many open source libraries like Pymc, Numpyro and other more like smaller projects like X time at the moment, which was about forecasting. And one of these projects, which is called Pymc marketing which mainly handles customer lifetime value and media mix model for marketing marketing efficiency.

Actually grew up quite a lot. I have been working with the folks of Pymc labs, which are kind of it's a Bayesian consultancy, so to say. So most of the consultants there, or like members of the team, are core pymc developers, which is a statistical package for Bayesian modeling. And then we support companies

and clients to adopt this type of, of modeling techniques. So as a matter of fact, I'm working 80% at (anonimizuota kompanija) and the 20% remaining i'm working with Pymc Labs to push this open source project. So this includes kind of handling the community and making sure that people are engaged with the package, that they use it, and if they don't use it for whatever reason, understand the reasons and trying to fix it. Also checking pull requests like code changes and also. Yeah, just just. I'm not doing this by myself, but I'm mostly leading a at least the direction because I started the project and I never expected it to be so big. Yeah. So that's about me. I don't know if that was too long or not, but maybe this would be a good starting point.

I.: That's great. Really interesting. I already have questions.

P.: Yeah. Sure.

I.: From where to start. You mentioned that it was hard to predict during the pandemic? Why?

P.: So, many, manu. Kind of seasonal. Let's say many time series that you find in the industry have kind of three components one, seasonality component, like for example, people order more boxes or food in the cold times because they are kind of they don't want to go outside. That rather than the summer that everyone is keen to just go outside and and been out. So there's like the seasonality component, which for most of the business is very clear. There's a trend component which is kind of the growth. And this could be related to the number of stores that you open, the popularity of your brand, and so on and so forth. So for a business like (anonimizuota), at that time was very linear trend growing nicely and nice seasonality. And if you need to predict that, you just need to capture these things. And I mentioned a third component which is called the remainder, which is what you cannot explain by these two. And this could be marketing events or outliers or maybe something hidden in the data structure like autoregressive terms and so on. During the pandemic, let's say if you want to do a prediction for the future, essentially you want to try to extrapolate what you have seen, but then you have this nice seasonality and trend, and at some point this is just get destroyed, like one order of magnitude up or down depending on the industry.

And then it's pretty random what's going to happen. So no one we don't we didn't have previous data on pandemics to try to say, okay, actually we are expecting a uplift of ten x. And during the first weeks we went, let's say it was at least for (anonimizuota), like something like of the order of ten x increase. So something that you have never seen. And at that point no one knew what that was going to happen. It was a lot of uncertainty. So at the very end, it was really trying to have a clever guesstimate. And then things after the pandemic kind of stabilized a bit, you recovered a bit of the seasonality, but the trend was was distorted. And also we needed to predict, let's say, 1 or 2 months in the future. And a lot of these

changes actually happened very recently. I don't know if you remember, but there was when there was a new variant, people would get scared and start packing food, and that happened within a 1 or 2 weeks. But if we were forecasting for the next month, of course, these forecasts didn't know about that. So that is the idea behind I mean, there are techniques to make this more dynamic, but that's the main challenge.

I.: Okay, okay. Thank you. Wanted just to understand from what perspective? Nice. Yeah. You you've mentioned that in current company, you are doing basically between pricing and forecasting, pricing, how to price items in an intelligent way. What do you mean by intelligent way?

P.: So let us let us assume. Let's play this game. I think it's very interesting. So assume that your task is to sell apples. How would you price apples?

I.: By referencing how much others probably price?

P.: Exactly. So I mean there are various techniques. So one is for example, that you want to price your apple in such a way that you don't lose money. So if buying apple costs you, let's say €1, you probably want to sell it more. But then it's important to see what the market says, right? So maybe you go and the cost of selling it at 1.5 and Lidl maybe 1.25, and then you need to choose. Right. Because if and if you price them lower, you're probably going to sell more. But the revenue which is what you gain is going to be less, but if you price them higher, you're going to sell less. So this is kind of the demand curve where the price and demand and you need to find a way. Because if let's say what if all of the competitors are pricing the apples very low, and if we could do it a little bit higher just to increase our margins without hurting the price perception, and that would be good, but also for growing markets, you don't want to maybe go very expensive. You want to kind of capture audience and take it from there. And it's also important to map it with forecasting because you want to ensure that you have the assortment ready, right? Because if you have a good pricing, but then you cannot sell them because you have not enough apples, that's a bad pricing experience a customer experience. Yeah.

I.: Okay. Thank you I got that. Nice. Okay. Do you yourself have to work with the data gathering?

P.: So what usually happens is that the data comes into what is called like a data warehouse where there's a lot of tables. And it could be like typically something that you can query with SQL. It could be like snowflake, I don't know, redshift, Impala. There are many of them. All of the integration is mainly done by the data engineers or back end engineers. So for example, if you if you are capturing the clicks for example, or the impressions that you see, all of this comes, yeah, the back end and the data engineers put them into raw tables so to say. So depending on the company data scientists like myself would query or gather the data for the models from this very, very raw data. But more recently there's an intermediate layer of people called analytics engineers that cleaned this massive amount of data in a way that is much

usable. But either way, everyone working in data science has to, and as a fundamental part of every project to be able to query and merge this data. Because sometimes you want to merge purchase history with purchase characteristics. Right? And this is not like a huge table. You have a table of users, you have a table of purchases, and you have a table, a table of product characteristics. So if I want to ask the question, how many people in this city have ordered red apples, you need to do the query yourself. So if you mean that by data gathering, yes. If you mean like the data integration that's usually made by the data engineers or backend engineers.

I.: How raw do you think the raw data is?

P.: So that depends on in the company and on the use case. Because what's happening is there's very raw, which is events. So it could be like every click that you do, every thing that you see. So if you have your app and you're scrolling down and scrolling up, even if you don't interact, let's say you don't click like the knowing that you open the app and you're seeing something. It's something that you can, let's say companies can track on your phone. So if you think about event data from your your device, that's really like a lot, a lot, a lot of data. So you this is most likely not in the data warehouse but something else. And kind of there's a big part of piece of work which is making this usable a in a pseudo aggregated way so that you can query because otherwise it's just too much. I don't I haven't had the need to go that granular. So usually I consume the data from tables that have already been cleaned, so to say, by either data engineers or analytics engineers.

I.: So you would say that the action data is quite realistic. Like quite - the most raw that you can go?

P.: Yes. But let's say for it depends on the, on the use case. So if you really need to you and there's this question about data governance let's say who is allowed to go there. So just to give you an example, companies usually have data about their own employees, like for example, salaries and I don't know, personal information that is useful for human resources just to and but employees don't have access to that table, right. I cannot just query an order by a salary. So There's also kind of roles for which you have certain restrictions, access to the data. And so most of the time, at least in these modern times, you have really clean layers done by the engineers. But if needed you can go like very, very raw. But then this becomes very hard because also the data structures are kind of nested. They are not just nice tables with columns and rows, but I could be like this JSON format which are chained. And doing that can be a little bit cumbersome. So usually you want to rely on something that is going to be maintained by the team.

I.: You mentioned the case when even non action is tracked, which is that non-explicit. So. What do you think about explicit and non explicit data gathering?

P.: What do you mean? Like, for example, a click versus an impression?

I.: Yeah. Or like a click versus a survey.

P.: So both of them have biases. So depending on what you want to do, let's say every data source has their like their own caveats. So to say. So every model that you develop, every conclusion you take out of it, it has to be given the data collection. So just to give you a very concrete example, if you want to measure Brand strength, so to say. So for example, they want to know that people feel empowered by the brand, which is something like companies like Adidas and Nike will typically would like to know because this is what's driving potentially sales. They asked surveys. So would you rather go for Nike or Adidas. And they they they these surveys are actually quite a yeah diverse. And they want to capture dimensions of your brand. These surveys have two problems. And this is actually people use this to track a metric called NPS which is called Net Promoter Score. And it's very important for many companies. But it has two main problems. One is that it's expensive to run. And the second is that A you have to do this every quarter. So to say at least because you cannot keep doing it for forever, it's just very expensive. So if you want to compare how you're doing against Nike, for example, between Adidas and Nike, you could do that. But that's very time consuming. What you can do is actually go and try to have a proxy of that in the digital world, and then you can have proxies for that, which is for example, the share of search. So if you go to Google Trends you could look for how many searches Adidas has, how many searches Nike has.

And then you could just take the share and see, okay, is this data actually being a proxy of the interest of people and brand strength? Or you could also look to take a look into a social media engagement. So this a has the benefit that is much more real time and it's way cheaper. So you could say okay, this is the way to go. But something that we also need to be aware of is that there are a lot of people that don't go that online so to say. So for example, myself, I don't use a social media that much and I don't have Instagram, I don't have TikTok. I used to have kind of Twitter, but now I don't like it. So these type of metrics and can actually have a bias. So even if they're super fast, they're cheaper, they're still a small bias. And another challenge that is going to happen with this large language model is that people will be using these data sources less and less in principle. So for example, search at least myself, I'm searching less and going using more ChatGPT or some of these large language models. So you also would need to account for this change in behavior so to say. So I don't think so. Surveys. And there's very good statistical methods to track surveys are still very good. This digital tracking. It's also good, but both of them are the weaknesses. And and you as a modeler or decision maker have to be aware about those.

I.: At which moments during your development do you think about the end user itself?

P.: So I mean, I would say always because at the very end that's what we're aiming for. In my case, users can vary. So for example, when I'm working with pricing and forecasting, actually my end user is the inventory manager who ordered the product. So going back to the apple, I estimate we're going to sell 20 apples in a week. And because I want to make sure that my customers are eating, if they want to order, then then the apples are available. But kind of my direct user in this case is the inventory manager who is going to order that and. But for example, when we're doing forecasting, we always need to think about the customer because you should forecasting models. You could either under forecast which means you estimate less apples that you actually solve or over forecast. And then they are not the same. Because if you under forecast users will go to the app and want to buy, but they cannot buy it. So it's frustrating. And for if you over forecast, then what typically is going to happen is that this food is going to go to waste. But actually I would argue that in this case it's better to over forecast because under forecasting it's really bad for the user. It feels like the data is the app and the assortment is not good for them. Because even if you over forecast, you can actually price these items down like mark down the price and try to sell it to get them out of shell.

So in our time series model, even with one developing, we actually weighting the errors in such a way that on the forecasting gets penalized more than over forecasting. So I would say that in every step sentence. Yeah. So as we know, let's say from because of the kind of business requirements that on the forecasting is worse than over forecasting. Then when we are developing the models and testing the models against others and selecting the model, we want to make sure that these weights, these errors are actually accounted for. And so yeah. So I think everyone at least should be always be thinking about the end user. Because a at the very end this is how they're going to to be consuming the model. So I don't know if the question was in that direction or in the direction of the data gathering part.

I.: In whatever direction you felt like.

P.: So yes, about this is about the output of the user and about the input of the data. I actually really like to work with data which has the less information possible about the user. So for example, and that's why I like statistical models and work with the aggregation. So for example it's not encouraged. And I think it's a bad practice to query for a user specifically or try to to get your email and try to see, Okay, what's your purchase history? Right. We want to be very careful about this sensitive data. And I would say that kind of GDPR has helped kind of change this this mindset. So in the data gathering in the data products, we also think, okay, we want to use data that is not sensitive, right. Even if can potentially

make our models better. And we I actually am very conscious about it. And that's why the models that I do in marketing, at least from my experience, that it was once I worked, it's the fact it's the one are the ones which use more aggregate data and less personal data. And this is not just because it's maybe nicer or not so evil, so to say, but also even with this personalized data, these clicks also have a big bias. So sometimes it's better to get away from the very granular data and see it from from from the bigger picture.

I.: So who do you think would have the biggest impact to the end result that the model gives? And having in mind that the context is from the user using the interface of the program that via which data is gathered and data... Some people then take that raw data, then that data is cleaned and you make something with it. And then we have the end result and the end user uses it. So out of all these actors, who has the biggest impact on the end result itself of the predictions?

P.: So I think everyone and especially but most importantly the data, Because if the data is, is is rubbish. So to say, no matter how clever the data science team, no matter how fancy the infrastructure is, the outcome is going to be bad. So kind of when people talk about data science, talk about these fancy models and this super nice cloud computing and llms and whatnot. But in reality the most critical part is the data. So and it's not just the raw data but the data gathering. And let's say that's why if you do some joins, if you filter by something, all of these steps are really, really key. So and these data manipulations these data wrangling actually depends on the output. So everyone in the process should think about let's say the end user. And but again these end users actually change a bit. So for example for the analytic engineer making the data usable and structured the data. The end user is not kind of someone using data, but actually the data scientist which are to consume that. So kind of as long as this process works a kind of we expect this to to work at the very end. But the critical component is understanding the problem and also knowing we have the data for that. Because in other cases and many cases, even if we have a business case, if we don't have the data before going into any modeling, the step zero should be, okay, let's start tracking the data. And that's typically how it works. It starts for for many projects. So there's no specific person that is more critical than the other is making sure that the process overall works.

I.: So to be precise, from your perspective, what is bad data?

P.: So bad data is for example, it can do many things. So for example incomplete data meaning that if you are tracking events, let's say sales, at some point there's sales are gone and you have a gap between these two. So this is typically a the case also messy format especially dates. This is a mess. So if you have dates in the European way that is day month year. And at some point in time you randomly change the format. Then also it's bad data in the sense that it's not wrong, but it's going to be a mess to, to to to work with that. So that's more on the technical level. But there's a deeper level which is on the kind of

biases. So for example, let us assume that I just collect data for we don't do this. But let's say we we just collect data for male users and we all do the analysis for male users. And then we are expecting this app to be deployed and used by everyone, not just men. Then you are going to have a big bias. And and this is just it's not representative to how this is going to be. So you also want to make sure that the data that you have is representative to the use case, so that even if the data is super clean, they are not missing value. It's bad data in the sense that you cannot really use it for the use case because of the of the bias that you have. Okay. And most of the time you have bad data. So in either of these dimensions. So you just need to make sure that you can do the best that you have like. And of course it's better to enhance the data and make it better than trying to fix it. Fix it with a fancy model.

I.: So when you said that the most important thing about the end result is data. And then you said that most of the time you have bad data. So it basically means that there's a lot of emphasis on the cleaning of data?

P.: Yes, yes. And it's not the sexiest part of the job. And people don't talk about it that much. But in reality, this is what we spend most of the time. And and for example, I like to develop kind of very transparent and explicit models. So our let's... Black boxes or more explicit. So for example how do we encode price discounts and all of this stuff in a way that makes sense. And by looking to the data, cleaning the data, you actually get a lot of insights and ideas on how to model this. Right. And so there are various strange mechanisms where people actually let's say they clear this, they decrease the price to clear the stock. And there's a lag between these two processes. So actually understanding this it's really going. Your hands dirty and seeing which case it works and which case it doesn't. And you will also learn a lot by looking to outliers and things that are away from the wider population. So yes, a lot of data cleaning, a lot of visualization, data analysis to make the end result useful because like otherwise there's no way out.

I.: What do you do when you find an outlier?

P.: So it depends on what you want to do with it. So the simplest thing let's say if it's something random and it could be that, for example, that that day there was an error in a data pipeline and some format was wrong. And you expect this not to happen often or like it's just random. Then you can just call it or just say impute it by taking the mean of these two values or just forget about it. There are other types of outliers which are a much more interesting and you shouldn't cut them out. So, for example, whenever you are working with a maybe a data from the US, or also sometimes in Europe, you have one day a year, which is Black Friday, where people go crazy buying stuff. And you always see this peak in sales. And and let's say you just have two years of data of daily data. These are 1700, 700 points. And

then you see just these two points. You could just say, okay, let's forget about them. But you shouldn't because these are very important points. So you may you always need to understand the source of, of of these outliers. And in many times the outliers actually will probably tell you something about about the underlying data. So sometimes the outliers could happen because you did a join in a wrong way. So kind of what would be a bad idea is have an outlier detection method. And if you see an outlier just call it automatically. That should be like a no go. It's very important to really understand if there are at random or this is a systematic outlier so to say.

I.: Right. So you create a model and how do you do your testing, your evaluation of the results that you get?

P.: So what you always want to aim in your development process is to have a metric that you want to optimize for, and it has to be tied with with the business. So once you have defined this metric, you also need to define what is called like a. Yeah. Like out-of-sample evaluation step on which you mimic what you will it will happen in a production live system. So in the context for example, of time series forecasting, what you do is you develop your model and then what you do is like a process of what is called time slice cross-validation. So you have your complete data set and you chop it in pieces, and then you forget the last pieces. You train just on this, and then you predict and see how it works. And you do this many times, making sure that the you are not kind of leaking information from the future into your training set. That's just the analytical part. But you also need to make sure that the technical part works. So you actually have various development environment, let's say clusters compute clusters like the production cluster which was running. And you have a copy of it which is called development. And in the production it's running the thing that we are happy about and it's solid robust. And if you are testing an alternative, then you deploy this model into the development environment, which should be an exact copy of production, and make sure that technically it runs. For example, if you have in your production environment something that runs in one hour and in your development environment, you have something that could be a slightly better, but it takes 48 hours to run. Then even though in your validation it's better, probably you maybe don't want to deploy just because it's too long. If the requirement that it has to be a valid prediction. So it's just not just the analytical part, but also the technical computation. We test this in a development environment.

I.: Let's say you still have to define some threshold where your result is good enough now or not.

P.: Yes. So that's why about this metric. So but it's not. So again let us assume that the metric is average percentage error. And what I have in production is 5% and it runs in one hour. And in my development environment I have another which is the accuracy. The metric is 4.75, but it takes 23 hours

and a lot of compute power. And that's costly. That's machines that are cost. So you also need to ask yourself, is it worth it. Because you also need to understand what's the monetary value in an increase of this metric. So okay, if we increase the error metric or decrease the error metric by 0.1%, how this will translate in money, for example. And then you need to balance that against what you're going to pay Amazon or Google to fit this model. Right. Because if it takes, I don't know, 20 hours this compute time, you need to make sure that you balance. So as a data scientist, you also need to understand like the monetary value of an increase in your metric. And then you make a decision. It could be that we actually care about the 0.1%, and it's actually paying off the 20 hour compute. It could be or it could be like, no, it's actually a waste because we're going to maintain that and it's just too much. Let's keep it simple for now and try to find another solution.

I.: So basically, you don't have fixed percentages of how much you.

P.: I understand your question. And in theory, yes, we could define when do we want to go into production. But the process is very complex because all of these models actually rely on teams and people. So I'm going to give you another example. Let us assume that everyone kind of knows how to use Python and how to develop everything in Python. And there's one very clever data scientist, and she did this in R, and just because she likes them then it's great. And it's the metric is great. It's fast so why shouldn't we. But then the team itself don't doesn't know how to write R. And then there's the risk. Because if this person quits or goes to holidays or gets sick, then and it breaks, there's a hidden risk about maintaining that. So either we all learn R or understand or there's documentation or maybe we say, okay, even though it's better just because of maintenance challenges, we're going to keep this because at the very end, doing it like having a metric, it's threshold is good for a one time analysis because then you do it. But if you want to deploy this into a production system, you also need to ensure that this is going to run long term. So that's why all of these processes are also part of the decision making.

I.: In which development processes or stages do you see the biggest risks of making sure that your result is of quality?

P.: The data collection part. Because yeah, the data collection part, it's so fundamental. It's so important that if that if you mess that up there, it's you're doomed, right? Because if you have good data, even a bad model can be very reasonable. So in my experience, like 80% of the problems in the industry can be solved with a good linear regression. So you don't need this fancy stuff for many things. Of course you have recommender systems and you have very complex algorithm. But if you want to predict sales, if you want to understand certain segmentation or linear regression or like a B test experimentation, there are linear regressions everywhere. So you don't need. So even with a really not great model you can get

something. But if the data is let's say corrupted, it has biases. It pass them, or the model doesn't understand where it's coming from, then there's no you're doomed. Right. And this could be, for example, a if you are collecting data at zip code level, and then you just train your data in 3D time and or Kreuzberg, which are areas in Berlin, very young. And then you want to use this insight to do your marketing in another neighborhood, which is older, more traditional. It's not a good idea just because the data is not representative. So everything about the the input or the output, expected data and how this makes sense is key. There's this is at least my opinion.

I.: Okay. Would you say that there is a risk for a developer to. Okay, so let's say a developer has a deadline and he needs to update them to improve the metric. And it's not really working. Is there a possibility that he with a bad will, let's say, can manipulate his data that he has, that the percentage of the result would grow, even though actually it doesn't really?

P.: Yes, I understand your question. So everything is about incentives, everything. So and that's why I actually like working in product companies. If I've never been forced to do that work in a product company, because if I, if there's if I do that, I increase the metric just to make my manager happy. And we deploy that, we're going to see the results back like it's going to You're gonna fail. Like, you cannot hide that, right? Because if I'm doing a forecasting, when I look in the back, this is beautiful and everyone's happy. We deploy it in production, and it's rubbish. Then, of course, it's my responsibility that the validation was not good. So in, in in product companies, deadlines will never like they're they're not worth the risk right. Because it's just too much. It's kind of breaking. This is breaking the consumer the customer experience. And and it's just bad. It costs a lot of money. Whereas if you're a consultant that you have need to do like a one time job and you just forget about it, then I could see a the, the incentive of doing that because one no one is going to challenge and test that. And and and to if it's not going to be deployed and everyone's happy, then then it's fine. So that's why working with these data consultancies and myself working at labs, let's say we have a very strict and transparent way of working. So for example, a consultancies data consultancy that don't share the data processing or the code. That's always a bad sign. So the more open they are and this is a key component is reproducibility. So whenever I work I always make sure that the number that I get is reproducible. And anyone can audit the process. But I've also seen cases in consultancies where everything's closed. Let's say there's this fancy consultancy and say we have a fancy model that produces this output and it's great. And okay, can we see the model? No. It's ours. In that case, for me, it's like they could just have written the number on a piece of paper or in Excel. It's like, okay, that's the number that they want. So it's really about the incentive. And yeah, for companies it's probably not a good way. Good practice for consultancies. This might happen.

I.: Okay. A bit more personal question, but you've mentioned that you barely use any social media. How do you feel about using machine learning software?

P.: I mean, I don't think they're they are connected in the sense that I don't use social media because I would rather use my offline world and be with my kids, or not be so plugged with my phone. So really. Kind of being disconnected. For example, not even a watch or any type of device. So I like. This type of freedom for the statistical. When you talk about machine learning, I, I'm very. Think about it in a very classical way. This is kind of statistical modeling. I don't I'm not used to. To to this hype. So it's just doing stats. You can do this in R Matlab. Let's say these are things that are disconnected. One thing that I do see a bit of a contradiction is the fact that for my day to day work, I actually use a lot of these large language models to, for example, improve my or speed my code. This could be GitHub Copilot, which is free, or actually I have a subscription just because I like using it. Or maybe you can ask ChatGPT about okay, how do I fix that? And I know that I don't know if I'm for sure, but I could imagine that Microsoft and GitHub train this large language model with all of the repositories that they have, and probably the whole internet, right? And so in that case, yeah, that's a bit of a of a contradiction. But I actually think that all of these large language models and let me call it AI products are here to stay. And if they make my work better, why not use them? So for example, I use software that corrects my typing because I do a lot of typos when I'm correcting. So that's going to help me actually write better and improve them. So be it. And I don't think that much about which data was trained this on. Yeah I.

I.: What would you think about the statement that correlation equals causation now? But no, no a differet statement. That correlation no longer does not equal causation?

P.: So I guess what you're trying to point out is that the fact that correlation is not causation or what's the question? Or could you repeat the last statement?

I.: Yes. Okay. So the thing I want to ask your opinion about is the thing that even though we know that correlation does not mean causation in the data science, statistics, machine learning, we use correlation.

P.: Yes. And but that's yeah, I think that depends on the use case. So if you want to predict something, usually people think about machine learning as something that that's model that can predict. So for example, if you have a linear kind of time series that grows over time very nicely, you can put my age in days in the model, because my age in days is also growing every day. Again, like I get older, right? And if I put this into a model, which or a time series model which has a linear trend, it's going to be great. It's going to be a great predictor. And I often use not my age but like a linear component just to capture that. And I know it's not causal, but it's good for prediction. There could be other complexities, but I just

use that for prediction. So if it's for prediction purposes, that's fine, if you just want to predict. But in many cases people don't want to just predict but also intervene. So think about the price and demand, right? If we want to understand this, we want to change the price to see how the demand changes. So in the previous example you cannot alter my age. You cannot just intervene my age. And that will actually have no effect on this time series. So even though that people say that they want prediction models, 80% of the time, they don't. They actually want causal models. And that's why I've been working in this area and learning about this and making part of of my work to always start with, with a causal graph or like structure. And so in pricing forecasting and in marketing, I've always had this mindset of, of having this causal structure. And and actually I've written a couple of blog posts about it and maybe I can send that to you. Where where I actually explicitly tell people that this is not just going to give you a slightly bad model, but it's going to actually make you take very bad decisions. So I have an experience on which I've explained the model and people didn't like some stakeholders, didn't like the model because they didn't have all of the variables. For them, it was like, it doesn't make any sense. But in causality there's this concept of confoundingness. And many times you just put all of the variables, you're going to mask effects and it's going to actually be incorrect. And I told them like, actually, no, this is incorrect just because of causal kind of theory and because you get biased results. And they will tell me, yeah, it's fine, but we can actually it's better if you just put them and we'll be safer. So I did a blog post that is called Data Science for Bad Decision Making and show. In a very real case, it was a simulation, but it could feel very real that by using like forgetting the causal structure, you can actually make very bad decisions that can cost you a lot of money. And actually, it's better not to do anything at all than doing this by brute force. And so yeah, I think it's a very fundamental topic. And I'm not surprised that people are talking less and less about pure prediction models and talking more about about causal models.

I.: That's a nice insight.

P.: Yeah. Yeah. And this is this is a trend that hopefully is here to stay.

I.: Now speaking. In relation to ethics. Where would you see the biggest risks in machine learning development from the perspective of ethical impact?

P.: I think it kind of the something that I don't really like is when companies use data, let's say user level data, and the user kind of doesn't know about it. And, and, and especially so there are two of them. So one is on the on the pure product side where they enhance their the data product. So for example imagine that I'm we're talking on LinkedIn. Having private conversations. And LinkedIn use our conversations to train their large language model. That's bad. Like I think this is a big risk because even though we could say, okay, there's no sensitive data like it shouldn't, like it's something private and I

don't know if this is happening. But for example, I saw I read that slack wanted to do this, this messaging platform. And I think this is just terrible. Like keeping the privacy is also important. But in addition to that, something that we've already experienced, to use this data and sell it to other third parties, which, for example, can have an influence in politics. So all of all of this around, for example, Cambridge Analytica and what happened with the US election when Trump was elected? I mean, we all know that this is real and how people can use this, this data to, to manipulate audiences. So, for example, the manipulation that's happening in, in Russia or China where things are very constrained and there's the information is, is restricted in such a way that you just get a half piece of the story. So I think the this and they're very people, clever people working on that. So the misuse of of the data for. Yeah on consent products and, and kind of yeah. Politicians, politicians and and and dictatorship. It's definitely something that is not a risk. We're already seeing it. Yeah. It's for example, I have... Like meta the, the software company Facebook. So to say they're doing a lot of interesting things in research like open source large language models. Yeah. Advancing a deep learning languages like PyTorch. So they do many things. But what they did and the business model that they had about manipulation, addiction and everything that I've done, I cannot simply stand it. So, for example, I would never work for a company like Meta, like, I can't, I simply can't because what they have done and and it's not it's not that it's not mystery. It's like a fact. So these big tech companies and yeah, I don't like working for such big tech companies. I like like more middle size and consumer goods. Something like it's okay. We want to be able to deliver food in a sustainable way so that we merchants, users and and the app, we all win. So I still like for me, it's still very important whenever I'm looking to, for for a job that the purpose is it's a it's not like this the company like that?

I.: So I've asked you what are the risks of creating a good machine learning program, but how would you, what would be a description of a good machine learning program?

P.: So again, everything it's it's about the purpose. So when I was kind of getting into the data field, I actually worked with Organization called Correlate. And kind of their pitch is that they do data science for social good, because all of this social media data, web visits, tracking data that a company like Walmart is also something that like an NGO organization, which could be for helping refugees to gather food. They also have it, but they don't know how to use it. So you can actually use the same methods but for a different purpose. So for example, think about marketing. Companies want to understand where to put their next euro. Meaning should I put like a Facebook ad? Should I put like a TV ad or do something else? But NGOs were faced with a similar problem where they have limited budget and they have to organize events to engage with people, for example, To to to get more visibility. And then they could

pay Facebook or they could do like an event. And measuring these you can use pretty much the same techniques with some tweaks here and there. And they're very nice. I can also share the link about that. A very nice success stories about how to use the same thing, the same model, because the model is agnostic to the problem, right? It's like linear regression would fit for kind of an NGO and like a for, for Trump, like evil Republican Party. So they don't care about it. It's more about the people behind them. It can make them. And of course can be intentional and unintentional, because there's also effects of of the model that has biases. And the model didn't know about that. So that's another risk. So I still think there are many many opportunities for for that.

Informantas Nr. 8

Amžius: 26m.

Lytis: vyras

Darbo vieta: universitetas

Gyvenamoji šalis: Lietuva

Pilietybė: Lietuvos

Formatas: nuotolinis pokalbis

Trukmė: 45min 12s

Data: 2024 m. spalio 28 d.

I.: Esu Simonas Bansevičius, politikos ir medijų magistro studentas. Darau tyrimą apie mašininio mokymosi programų etiką. Informuoju, jog bet kada gali nuspręsti nutraukti pokalbį. Ar sutinki, kad šis pokalbis būtų įrašytas ir jo medžiaga būtų naudojama tyrimo tikslais?

P.: Taip, sutinku.

I.: Tai va, pirmasis klausimas būtų prisistatyti, jeigu gali tai ir pasakyti amžių. Kuo dirbi šiuo metu? Kiek laiko esi šioje srityje?

P.: Man 26-eri metai. Dirbu duomenų mokslininku. Dirbu prie realiai universitete, prie projektų. Dabar jau eina, man rodos, treti metai, kaip dirbu šioje srityje. Esu baigęs taikomąją matematiką, bakalaurą. Dabar studijuoju taikomosios matematikos magistrą. Tai maždaug tiek.

I.: Papasakok apie savo darbo specifiką, ką darai?

P.: Šiaip dažniausiai tai analizuoju tam tikrus duomenis, pavyzdžiui, laiko eilučių duomenis, lentelių duomenis arba paveikslėlių duomenis tenka analizuoti, taikyti visokius mašininio mokymo algoritmus, spręsti klasifikavimo, regresijos, anomalijų aptikimo, klasterizacijos užduotis. Taip pat

atlieku visokius statistinius tyrimus, jeigu to reikia. Kartais prisidedu prie mokslinių straipsnių rašymo. Kažkas tokio.

I.: O kokiuose kontekstuose tavo minėti uždaviniai atliekami? Kažkas paprašo padaryti užduotį ar tiesiog, kiek gali, išsiplėsk.

P.: Tai būna tam tikras projektas, prie kurio dirbu, tai ten dirbam su kitomis įmonėmis, kartais kartais su kitais universitetais ir būna kažkokia tematika. Sakykime, dirbu prie miškų projekto šiuo metu. Taip pat dirbu prie, ten su inovacijų agentūra ir su kitomis agentūromis, kurios finansuoja ir teikia Europos Sąjungos paramą tokiems projektams. Tai tai realiai būna tam tikros, kaip ir temos, kurias galėtų finansuoti Europos Sąjunga ir kartu su kitomis įmonėmis rašomos paraiškos. Ir tada, jei tos paraiškos laimimos, tai ir dirbam prie tų projektų.

I.: O tai, pavyzdžiui, dabar minėtas miškų projektas, tai jį užsakė kas?

P.: Tai čia labai didelis projektas, kuriame dalyvauja keli universitetai iš kelių šalių ir taip pat kelios įmonės. Dabar net negalėčiau pasakyti, kiek tiksliai yra skirtingų atšakų, to projekto, nes jis labai didelis, bet bent jau ta dalis, kur aš dirbu, tai susijusi su Valstybinės miškų tarnybos duombazės skaičiavimų atlikimu. Tai čia toks labiau vienas iš tų projektų, kur ne tik susiję su mašininiu mokymu, bet su pačia matematika ir formulėmis. Čia toks retesnis variantas.

I.: O įprastai keliese tenka dirbti prie vieno projekto?

P.: Būna įvairiai. Esu didžiausioje grupėje dirbęs, tai buvo komandoje, berods, aštuonių žmonių. Kita komanda, kurioje dabar dirbau, buvo šešių žmonių. Bet paprastai tai būna kokie 2, 3, 4 žmonės.

I.: Šitie žmonės, visi machine learning žmonės ar?

P.: Panašiai jo. Tai būna vadovas dažniausiai, kuris organizuoja visą tą dalyką skambučius, šnekasi su įmonėmis. Ir būna tada jo, darbuotojai, kurie dirba prie skirtingų to projekto užduočių. Ten tas, kur kompiuterinės vizijos projektas, tai ten ir turim grafinio dizaino specialistų, kurie supranta apie tekstūras ir panašius dalykus.

I.: Iš kur gauni, gaunate jūsų komandos dažniausiai duomenis?

P.: Tai iš tų įmonių, su kuriomis kartu projektą darome, nes, pavyzdžiui, tai toms įmonėms reikia kažkokio sprendimo ir jos ir duoda mums tuos duomenis. Realiai taip

I.: Jie duoda tokį raw data ar?

P.: Jo jie duoda, ką jie turi iš tikrųjų. Tai kartais pačiam reikia apsidirbt tuos duomenis. Kartais kartais būna jau gražioje formoje. Tie duomenys, kuriuos galima iškart skaityti ir dirbt su jais.

I.: O dabar toks man klausimas, o tai kodėl, pavyzdžiui, tos įmonės kreipiasi į universitetą, o ne pačios daro?

P.: Tai jos neturi tiesiog tų specialistų.

I.: Tai ar galima pasakyti tam tikra prasme, kad tiesiog jus outsource'ina? Ar čia kažkaip kitaip vyksta abu dalykai?

P.: Nu, tikriausiai taip ir yra, kad. Jo.

I.: Tiesiog norėjau susipažinti su situacija. Tai pats neturi įtakos jokios tam, kaip duomenys surenkami?

P.: Buvo vienas projektas, kur dirbau su (anonimizuota į medicinos universitetą). Tai ten mes rinkome duomenis iš pačių pacientų. Jau ėjom į pačią ligoninę ir ten su jų sutikimu instaliuodavom tokias programėles į jų telefonus, kurios rinktų duomenis. Tai vat čia vienintelis buvo toks momentas, kur pats esu dalyvavęs tame duomenų rinkime.

I.: Tada papasakok apie tai, kaip, kaip tiesiog vyksta duomenų apdorojimas. Kai gauni raw data ir tau reikia dar apsidirbti.

P.: Tai tikriausiai pirma reikia susipažinti, kas tie duomenys yra. Dažniausiai tenka klausti, ką reiškia tam tikri kintamieji, ar jie yra naudingi, ar nenaudingi. Tada pagrinde šiaip dažnai vadovas jau būna apžiūrėjęs tuos duomenis ir nusprendęs, kas yra naudinga, ir tada tik tai mus įtraukia, kad jau pradėt darbus. Dažnai vis vien reikia jos apsidirbti tai prasivalyti eilutes, kurių per daug missing values yra. Taip pat gal kažkokias reikšmes reikia užpildyti, galbūt reikia agreguoti, ten paskaičiuoti kažkokias sumas per mėnesį ar ten vidurkius per mėnesį, jeigu tai laiko eilučių duomenys. Ir kažkokias lenteles su kitom lentelėmis sujungti, kad gautų tą duomenų aibę, su kuria jau galima dirbti.

I.: O kaip nusprendi ar užpildyti, ar ten naikint?

P.: Tai tikriausiai pagal kažkokį tą kintamąjį. Stengiamės, aišku, neišmetinėti jokių dalykų, bet kartais yra tiesiog duomenys, galbūt neteisingi, yra neteisingai įvesti. Ten visiškai nelogiški. Gal ten apskritai kažkokios sistemos klaidos tuose duomenyse ir tada tokiais atvejais jau pravalymas tas visas duomenų ir vyksta, o užpildant. Užpildant dabar neatsimenu. Šiuo atveju kur tektų užpildyti, tai negaliu pakomentuoti.

I.: Tai dažniausiai realiai jeigu duomenyse būna klaidų, tai tiesiog išvalo tą eilutę realiai?

P.: Nu jo išvalant tą eilutę, aišku prarandama gal būt ten kitų kintamųjų informacijos, bet jeigu ta eilutė visiškai ten jau tokia netinkama, tada į analizę kaip ją traukti, būtų visai klaidinga sakyčiau.

I.: Bet vis tiek minėjai iš pradžių, kad ten būna, kad agreaguoja.

P.: Taip taip. Jo būna. Bet tai, kartais būna taip, kad tuos duomenis, kur valom, tai jų ir nereikia agreguoti. Tokio atvejo, kur ir reikėtų išvalyti, ir agreguot, dar gal ir nebuvo pasitaikę man. Tai su šita problema nesusidūriau. Tačiau ateityje tikriausiai bus tokia problema. Nežinau, ką darysiu.

I.: O kai jungiat, pagal... Kkaip vyksta mąstymo procesas, kaip nusprendi, kuriuos duomenis jungti?

P.: Tai pagalvoju, jeigu ten tos SQL duomenų bazės duomenys, tai tiesiog pagal kažkokią ID. Apjungia kelias lenteles ir pasirenki kintamuosius, kurių tau reikia.

P.: Tai tikriausiai toks būtų apjungimas. Kiti apjungimai, kur galvojau galėtų gal būt, tai kai yra apmokant kokį kompiuterinės vizijos modelį, kad skirtingus data setus naudoti, bet tokio nesu pats daręs Bet tikriausia įmanoma būtų fine tune'int su kitais duomenimis. Čia nežinau ar čia skaitytūs kaip apjungimas, ar nesiskaitytų.

I.: Ar esi turėjęs kada nors, kad reikėjo tau susidurti su Cold startu?

P.: Kas tai yra tas Cold startas, kur nėra duomenų tikriausiai? šiaip ne, nesu susidūręs, bet žinau, kad tenka susidurti su tokiomis situacijomis, nes dažnai įmonės nori padaryt kažkokį produktą, bet duomenų neturi. Tai tokiais atvejais paprastai jie sugeneruoja kažkokius duomenis, kurie galėtų reprezentuoti tą realybę ir realius duomenis, kurių jie kaip ir neturi. Bet pačiam nėra tekę dirbti su tokiais duomenimis.

I.: Turi omenyje, kad jie, verslas turi ir nori išspręsti problemą, bet neturi jai duomenų, tai patys pagalvoja hipotetiškai kokie tie duomenys galėtų būti?

P.: Jo, jo.

I.: Aha, čia čia tokią istoriją. Girdėjai tokį atvejį iš kokių šaltinių?

P.: Iš pažįstamų duomenų mokslininkų.

I.: Įdomu.

P.: Įdomūs tokie projektai tada gaunasi kur... Bet nu na, jeigu jie ten žino, būtent kokius duomenis jie nori pradėti rinkti ir tiesiog dar nėra jų tos įmonės surinkusios, tiesiog žino kintamuosius ir. Gal ir įmanoma teoriškai kažkokį modelį apmock'inti tokių fake duomenų? Jeigu bent jau kažkokias statistikas įsivaizduoja žmogus, kuris kuria pačius tuos duomenis. Bet klausimas tada, kaip realybėje tas modelis veiks?

I.: Gerai. Tai va, įsivaizduok, dabar sukūrė, sukūrei modelį. Papasakok, kaip jį vertini. Kaip vyksta jo rezultatų įvertinimo procesas?

P.: Tai paprasčiausiai žiūrima į tas visas tikslumo metrikas. Tai jų skirtingų yra priklausomai nuo užduoties ir dažniausiai duomenys skirstomi į tą testinę imtį, valdymo imtį ir treniravimo imtį. Tai treniravimo imtį naudoju tik treniruojant duomenis. Validavio imtį naudoju patikrint, ar teisingai modelis prognozuoja čia. Validavimo su neuroniniais tinklais susiję būna, tai ten irgi mokymosi procese jie kaip ir yra naudojami, o tada testavimo imti, kuri visiškai niekada modelio nematyta, tiesiog ant tų testavimo duomenų ir prognozuoja tas reikšmes ir tada lygini tikras reikšmes su prognozuotomis reikšmėmis ir

taikai įvairias tikslumo metrikas. Ir aišku, aš. Tai su regresijos užduotimi ten būna random mean square metrika, Maya. O čia tokios sutrumpinimai kelių raidžių, bet iš dalies vienas į kitą kažkiek panašūs, o klasifikavimo tai būna accuracy, berods dar būna. Kaip yra toks bendras accuracy, rock kreivės specifiškumas? Yra dar kelios yra, bet šiuo metu neatsimenu. Tiesiog priklausomai nuo to, ar ten tavo gauta duomenų jungtis yra išbalansuota kažkokia. Gal vienos klasės yra daugiau nei kitos klasės, tai ir skirtingas tas metrikas renkamės.

I.: Išbalansuota - gali patikslint?

P.: Tai išbalansuota tai būna, kad tavo duomenyse, sakykim, turi šunų ir kačių duomenis ir pas tave duomenyse ten koks 10000 šunų duomenų ir tik 1000 kačių duomenų. Tai vat tada tokia imtis gaunasi išbalansuota ir kad tą balansą palaikyti žiūrint į tikslumą modelio, tai reikia tos kitokios truputį tikslumo metrikos naudoti.

I.: Tai tos tikslumo metrikos, jos tiesiog būna šiek tiek tarsi hardcode'intos, kad paslinktų pasiskirstymą ar?

P.: Ten kitokios formulės tiesiog taikomos.

I.: Tai šitos formulės grynai nustatytos tam tikriems išbalansavimo atvejams?

P.: Jo, kaip ir.

I.: Hmm. Tada kokių nors kriterijų, kurie darytų įtaką, kokią metriką naudojote, ar yra?

P.: Visokios vizualios vizualizavimo būdai naudojami aišku, pažiūrėti, ar ten teisingai modelis dirba ir aišku tada, kai production išeina. Tada dar žiūri, ar modelis realiai veikia, ar duoda pelną įmonei.

I.: O kaip šitą vertini production'o atžvilgiu?

P.: Tai tos pačios metrikos taikomos. Šiaip aš ten prie to daug nedarau nieko. Ten jau perduodama labiau tos įmonės programuotojai tenais žiūri. Nebent mums reikia dar kažkaip fine tune'int tuos modelius.

I.: Prisilieti prie visų AB testing?

P.: Šiaip minimaliai esu, tik vat su tais pacientais kai minėjau, kad dirbom, tai tikrinom tarp pacientų grupių visokius statistinius testus. Tai gal čia irgi galima būtų pavadinti kaip AB testing. Bet kažkokius tokius tradicinius kaip ten website'uose taikomi tie AB testing'ai, tai tokių nesu daręs.

I.: Kadangi esi toks... Kuri projektą produktą, bet ne iki galo visuose žingsniuose jo kūrimo esi. Ar jauti kažkokios, atitrūkimo nuo viso proceso?

P.: Galbūt atitrūkimas yra pačioj pradžioj projekto, kai reikia išsiaiškinti, ką visi tie duomenys reiškia. O, tai vat ten ir būna sunkiausia suprast tą domeną kaip ir. Ten reikia daug pagalbos iš pačių tos

įmonės specialistų, kurie supranta. Tai čia gal labiausiai toje vietoje pasijaučia atitrūkimas. Ir aišku, jeigu tau yra tas projektas iš kažkokios naujos temos, su kuria nesi dirbęs, tai irgi tame pasijaučiau atitraukimo.

I.: O testavimo atžvilgiu?

P.: Testavimo atžvilgiu Tai. Kaip ir nelabai. Jeigu jau ten supranti tuos duomenis, tai tada. Tada toje vietoje jau kaip ir to didelio atitrūkumo nėra.

I.: Kuriuose savo darbo kasdienybės procesuose matytum didžiausias rizikas algoritmo kokybės užtikrinimui?

P.: Savo darbe tai didžiausios rizikos. Gal. Tai gal su tais duomenimis, kad reikia jų kokybiškų, kad reikia tuos kintamuosius susidaryti gerus, bet ir. Čia buvo tokio, toks atvejis, kur galvojam, ar mes čia gal pažeidžiame Europos Sąjungos tos teisės sudarinėdami kintamuosius, bet pasirodo nepažeidžiami. Tai vat reikia į tuos dalykus atsižvelgti. Čia galbūt iš tos pusės yra rizika. Aišku, pačio modelio gal kūrimas, jis vis vien turi būti kažkoks logiškas. Ten yra visokių istorijų, kur tenais prekybos centrai pradėjo prognozuoti, ar ten moteris nėščia pagal tai, ką perka. Tai vat prie tokių modelių tai nenorėčiau dirbti, nesu dirbęs, tai tikriausiai tame yra rizika pažeisti žmogaus privatumą.

I.: O kai sakei, kad dvejojate, ar kurdami kintamuosius, sudarydami nepažeidžiat teisių, tai čia koks čia atvejis buvo?

P.: Tai čia buvo su. Kadangi rinkome iš telefonų GPS sensorių duomenis, tai ten buvo sunku. Kažkaip nustatinėjo, kur dažniausiai žmogus leidžia laiką ir kaip toli nuo tos vietos būna. Tai vat toj vietoj buvo klausimas ar čia taip galima daryt, bet lyg ir kaip ir pasitariau su draugu, kuris dirba duomenų apsaugos. Agentūroje Lietuvai. Na, nežinau kaip vadinasi, tiksliai tai Tai sakė kaip ir viskas ok. Čia su tuo reikalu. Na ten aišku žinai, žmonėm nėra tikriausiai labai malonu jei kaip ir pasirašo, kad surinks iš jų telefonų duomenis, bet žinai vis vien jam gali būti nejauku, kad tie duomenys kažkur ne taip panaudojami. Gali būti. Tai čia irgi tikriausiai yra rizika apsaugoti tokius duomenis ir stengiamės dirbti tik serveriuose, tų pačių duomenų atsisiųsti į savo kompiuterius ar kažkur. Tai čia gal iš tos pusės dar bandoma apsisaugoti? Ir tų pačių, kaip ir ten tų visokių ten žmonių vardų ir pavardžių? Tokios informacijos niekad ir neturėjom.

I.: Bet šiuo atveju taip atrodo. Viskas priklausė grynai nuo jūsų vos ne moralės.

P.: Na, gal gal ir galima taip pasakyti, bet ne tik nuo mūsų moralės. Mes, kaip ir teisiškai, negalėtume kurti tokių kintamųjų, kurie pažeistų mūsų idėją. Kaip ir buvo iš to viso tyrimo parašyti mokslinį straipsnį. Aišku, iš to į mokslinį straipsnį ten jau tokių kažkokių dalykų nedėsiu, kurie laužo įstatymus. Ir ne tik tai, bet tuo pačiu ir kaip ir nelabai įdomu, ar kur ten tie žmonės gyvena, ar kažkas neturi laiko. Ne tik kad neįdomu, bet dar ir nėra laiko į tokius dalykus gilintis. Tai.

I.: O dar sakei, kad vat problema, kad duomenų reikia kokybiškų, kad būtų geri kintamieji. Tai kaip apibūdintum, kas yra geras kintamasis? Kas yra kokybiški duomenys?

P.: Tai kintamieji, kurie. Reprezentuoja galbūt realybę, kurie susiję vienas su kitu, iš kurių galima daryti tą analizę, kuri nėra visiškai randomly sugeneruoti, būna kai kurie tokie duomenys, kurie visiškai neįtakoja to tavo priklausomo kintamojo, kurį tu turi. Iš tokių duomenų sunku kartais modelį padaryti. Ir aišku, kokybiškai, kad nebūtų visų tų daug praleistų reikšmių, kad. Nebūtų skirtingi tipai stulpeliuose. Aišku, būna įvairiai, bet ir galima sutvarkyt, bet tiek tikriausiai.

I.: Kaip nustatot, kada jau yra over fitting'o riba? Ar išvis ją pasiekiate Kada?

P.: Yra tam tikri rodikliai, pvz treniruojant neuroninius tinklus, kai train'inimo aibės accuracy būna labai aukštas, o validavimo accuracy vis krenta, krenta arba nekyla, tai tada pasireiškia overfittinimas. Aišku, pasižiūrim į jau pritaikomą modelį ant testinių duomenų, pasižiūrim kokios. Kokia to modelio kaip ir išvestis yra pas jį braižant brėžinius. Įvairūs, ten yra visokie brėžiniai ir panašiai. Tai pagal ir histograma, tai galima pagal jas spręsti, ar modelis overfit'ina. Tiesiog per daug priprato prie duomenų.

I.: Bet kažkokios konstantos ten procentinės neturit?

P.: Ne, tokio neteko. Nesu girdėjęs apie tokį, bet gal ir yra.

I.: O turit kokį nors tai nusistatę lygį, kada yra jau pakankamai geras rezultatas?

P.: O šiaip tą lygtį dažniausiai nustato tos įmonės. Mes testuojame visus tuos metodus, kuriuos galime. Daug skirtingų modelių. Ir išrenkame kaip geriausią arba didžiausią potencialą turintį modelį. Ir tada toliau atliekami tyrimai su tuo modeliu. Ir. Aš visada noriu, kad labai gerai prognozuotų viską, bet. Kartais visokių išorinių faktorių galbūt yra, kurie lemia, kad modelis ten, nu jis visai bus geras, bet galbūt ne toks geras, kaip norėtum. Arba atliekant tyrimą galbūt ne visada tas modelis toks geras gaunasi, kaip galvotum, kad galėtų gautis, bet aišku tiesiog toliau bandai, bandai, bandai viską iš naujo. Ištestavai viską, ką galėjai ir paklausi visų žmonių, kaip ir išminties, kurių galėjai paklausti. Ir internetas nebepadeda. Tai ką padarysi. Gal tiesiog tie duomenys nebuvo geros kokybės ir kintamieji nereprezentavo tą priklausomą kintamąjį ir tiek.

I.: Ar galvoji savo darbo kasdienybėje apie galutinį naudotoją?

P.: Šiaip dažniausiai... Dabar galvoju? Dažniausiai net ne... Nu jo, kaip ir pagalvoju, bet tai dažniausiai būna tas... tos, kaip ir įmonės, kurios naudoja tuos modelius. Įmonės darbuotojai, tam tikras analizes vykdant ar ten kažkokie maršrutus sudarinėjant ar pan.. Tai apie tą kaip ir... Čia galbūt toks labiau klausimas turėtų būti, kai į website'ą kažkokį modelį įdedi. Ir tada žmogus paprastas vartotojas naudoja jį nežinodamas, kad ten AI yra. Bet dabar galvoju, ar apskritai man yra tekę prie tokio varianto dirbti? Tikriausiai... Nu tas vienas iš kompiuterinės vizijos modelių. Tai jo galvojama apie galutinį

vartotoją. Tai, kad viskas ten taip ir. Kada žmogus norėtų ten fotkintis ir kad norėtų susikurti avatarą ir panašiai. Galvojame apie tuos dalykus, kad gal nu jo.

I.: Tai sakai, kad čia vat buvo tas atvejis, kai webe, o kitais atvejais kas tavo galutinis vartotojas būna tada?

P.: Tai va tos įmonės, kurios naudoja tuos modelius, kad pažiūrėtų, ar ten. Kaip ten sekasi, ar ten gera marketingo kampanija, ar tenais kokia prognozė kito mėnesio, kiek ten užsakymų bus ar kiek pardavimų?

I.: Tai realiai taip išeina, kad kai įmonė, kai įmonei darot šituos modelius, tai tada kažkiek mažiau tokio akcento apie galutinį naudotoją gaunasi?

P.: Jo jo.

I.: Sakykim, sukuri modelį ir jis niekaip nepasiekia reikiamo tikslumo. Ar tokiu atveju būtų rizika, kad programuotojas galėtų pasiimti savo duomenų rinkinį, jį truputį pakaitalioti, kad tipo pakoreguoti, pagadinti iš tos pusės, kad pasikeltų reikiamas rezultato procentas ir jis tada suspėtų su visais deadline'ais. Ir jeigu niekam nieko nesakytų, tai tiesiog duotų tokį prastą rezultatą, bet niekas nežinotų.

P.: Tai, nežinau, kiek tos rizikos būtų, tai tikriausiai būtų kažkiek. Bet koks tikslas tokio modelio tada reikia klausti.

I.: Pavyzdžiui, jeigu įmonė prašo reprezentuoti jų situaciją ir. Kadangi nenaudos niekas kaip ir production'e, tai tik vienkartinė tokia prognozė. Tai tarsi gali pateikti bet ką ir niekas nepatikrins, ar tai validu, nes production'e neina.

P.: Nu nežinau, nežinau koks to tikslas vis vien. Nu aišku bus patenkinti tie. Ta įmonė tikriausiai šituo hipotetinių atveju, bet tikriausiai gali atiduoti blogą modelį ir jie bus nepatenkinti. Toks tiktais skirtumas.

I.: Supratau, reiškia šitokios mintys nebuvo?

P.: Nelabai.

I.: Tiesiog vienąkart esu girdėjęs tokį atvejį. Galvojau, kiek jis dažnas gyvenime būna. Kas manai turi daugiausiai įtakos galutiniam modelio rezultatui? Tai čia kalbant nuo duomenų rinkimo iki pabaigos.

P.: Jau pats duomenų surinkimas tikrai. Nuo duomenų viskas priklauso. Ar modelis bus geras, ar ne? Aišku, ir modelis, pats modelio parinkimas tikriausiai irgi pagal duomenis parenkamas. Bet irgi tada pagrinde duomenys.

I.: Tada toksai įdomesnis klausimas. Ar koreliacija vis dar nelygu priežastingumui?

P.: Tai, taip. Jo. Nu čia tas Simpsono paradoksas berods ar kaip jis vadinasi? Kur tenais gali būt ruonių populiacija susijusi su kokiu nors konservatorių balsuotojų dydžiu? Tai.

I.: Taip, bet kartu yra žmonių, kurie sako, kad kadangi turime vis daugiau ir daugiau duomenų. Kartais atrodo, net nebereikia mąstyti apie priežastinę jungtį. Jeigu tiesiog turi pakankamai daug duomenų ir gali rast vis tiek kažkokią naudą iš to.

P.: Galbūt. Galbūt neuroniniai tinklai kažkaip geba atpažinti tuos. Kaip ir paslėptus sąryšius tarp duomenų. Bet tada vis vien bent jau man atrodo, kad naudingiausi yra modeliai tie, kuriuos gali žmogus interpretuoti, o neuroniniai tinklai tokios juodosios dėžės yra, kurių niekas per daug nesupranta, kodėl jie veikia vienaip ar kitaip, kodėl ten jie mąsto? Aišku, galima ten labai gilintis, bet kad naudingiausi vis dėlto tie modeliai, kur matai, kokie kintamieji labiausiai įtakoja ir tada. Gali pagal tai dar papildomą analizę daryti ar kažkokias išvadas.

I.: Kur matai daugiausiai etikos rizikos.

P.: Gal kad tie duomenys kažkur nutekės ar kažkur ne ten atsiras. Etikos. na, kad nežinau. Kažkas šiandien bus bloga nuotaika bus supykę ant visko ir kažkur tuos duomenis išsiųs velniop. Tai tikriausiai čia nebent toj vietoj, bet tai. Nežinau.

I.: Jau klausiau, kas yra rizika prastam algoritmui. O tada kaip apibūdintum, kas yra geras machine learning algoritmas?

P.: Tai. Tikriausiai tas algoritmas, kuris turi užtenkamai aukštą tikslumą. Ir tikriausiai svarbiausia yra, kad toms įmonėms kurtų naudą ir atneštų pelno dažniausiai.

I.: O kaip nustatoma ar pakankamai aukšto tikslumo?

P.: Dažniausiai tai įmonė. Priklauso nuo įmonės. Tenais nusprendžia, ar jiem naudingas, ar nenaudingas.

I.: Kaip jauties pats, kai naudoji kokį nors produktą, kur yra machine learning, kokiu nors būdu?

P.: Nežinau, visai normaliai jaučiuos, ta prasme, čia net negalima apsieiti be Machine learning produktų atsidarius telefoną ir atsidarius internetą. Tai net per daug galbūt apie tai negalvoju ir kaip ir neįtakojo manęs, Tai ar ten yra tas mašininio mokymo algoritmas, ar nėra? Gal kartais, kai tas reklamas meta pagal tai, ką tu parašei Facebook'e, žinutės būna toks biškį taip arba kad galbūt galvoji apie kažką, tai jis atrodo nieko ir net nerašei bet vat algoritmas tau išmeta tam tikrą video kažkokį, tai taip keistai kartais būna, kad pasijauti ir nežinai ar ten algoritmai daugiau žino ar nei tu pats, ar kaip yra.

I.: Tai neturi paaiškinimų pateikti. Kodėl taip nutinka?

P.: Ne, ne, ne. Neturiu paaiškinimo. Paaiškinimas būtų nu taip - jo net nejauku, kad tiek daug duomenų apie tave surenkama iš dalies. Bet jei mane tai labai jau grasintų, tai ten registruočiaus į visokius. Man rodos čia Surfshark Incogny yra kur, kad sustabdytų pardavinėt tavo duomenis, išimtų iš vender'ių. Tai vat tokius dalykus ir registruočiausi, jeigu man labai jau tenais erzintų.

I.: Kas tau darbe dažniausiai daugiausiai kelia stresą?

P.: Tai tikriausiai meet'ai su klientais. Reikia pasiruošti meet'ams, jeigu reikia šnekėti, jeigu nereikia šnekėti, tai net nekelia streso. Na, kartais būna, kad būna daug darbo, reikia ten daug visokių literatūros analizei atlikti. Ir deadline reikia įtemptai. Nebent tokiais atvejais būna streso, kad per daug darbo, o bet laiko mažai. Bet ką kartais tenka paaukoti kokią savaitgalio dieną ir padarai tuos darbus.

I.: Klientus paminėjai. O ar yra buvę.

P.: Klientai ir yra įmonės...

I.: Jo jo. Bet ar yra buvę kokių nesusikalbėjimo problemų?

P.: Būna dažnai tikriausiai, kada klientai ne visai supranta, kad, pavyzdžiui, nori tuos projektus daryt. Bet čia kaip geriau. Jeigu nori projekto daryt, bet neturi duomenų, tai čia jau iš esmės man atrodo, kad yra nesusikalbėjimas ta prasme, kad neturėtų būti projektas daromas be duomenų. Bet aišku, būna nesusikalbėjimo. O kas būna, kai tenka su daug skirtingų žmonių bendrauti? Aišku, vadovams pagrinde problemos. Man mažai tenka dalyvauti tame, kad ten vienas žmogus, vienas atsako, kitas žmogus jau kitas atsako. O ir tada bandai išsiaiškinti, ką jie iš tikrųjų nori pasakyti. Būna, neatrašo į laiškus ar dar kažkas. Na, įvairiai būna.

I.: O būna, nu vat sakai. Būna nori projektą daryt, bet duomenų nelabai turi. Ar buvę, kad nepavyko perkalbėti, įrodyt, kad neišeina?

P.: Aš nesu prie tokių dirbęs tai...

I.: Tai su kokiais tu esi sunkumais, nesusikalbėjimais susidūręs?

P.: Tai pagrinde aš tokį sunkumą, nesusikalbėjimo, tai kad neatrašo, kai tau reikia kažkokio paaiškinimo apie duomenis iš pačios įmonės. Ir tiesiog neatrašo, nes nežinau dėl ko. Tai vat tas labiausiai užknisa, nes kaip ir vilkinamas laikas yra ir reikia meet'intis projekto deadline'us ir tau tiesiog įmonė neatrašo su kuria tu realiai dirbi prie to pačio, kur jiems reikia to produkto. Tai labai keista man, bet. Dažniausiai spėjam visiem.

Informantas Nr. 9

Amžius: 25m.

Lytis: moteris

Darbo vieta: skaitmeninė rūbų pardavimo platforma

Gyvenamoji šalis: Nyderlandai

Pilietybė: Lietuvos

Formatas: nuotolinis pokalbis

Trukmė: 56min 14s

Data: 2024 m. gruodžio 3 d.

I.: Esu Simonas Bansevičius, politikos ir medijų magistro studentas. Darau tyrimą apie mašininio mokymosi programų etiką. Informuoju, jog bet kada gali nuspręsti nutraukti pokalbį. Ar sutinki, kad šis pokalbis būtų įrašytas ir jo medžiaga būtų naudojama tyrimo tikslais?

P.: Gerai.

I.: Tai pirmasis klausimas, kiek tau norisi tiek prisistatyti, pasakydama, ką dirbi šiuo metu, kiek laiko dirbi šioje srityje ir pan.?

P.: Tai aš dirbu duomenų analitike produkte. Šitą darbą dirbu jau beveik pusketvirtų metų. Visai taip netyčia. Iš tiesų prisijungiau prie šios srities, nes pamačiau, kad duomenų analitika visai įdomi yra universitete. Tiesiog pabandžiau Coursera kursą ir at the end I'm here žinai. Tai va, tai jo. Mano darbas dažniausiai yra testavimas ant vartotojų būtent tai. Nežinau. Jeigu pavyzdys, kažkokį naują filtrą pridedame, tai mes visą laiką testuojame ir pasižiūrime, kaip vartotojai į tai reaguoja. Tai mano darbas iš esmės yra pasakyti, ar tai yra atsitiktinumas, ar vartotojams tikrai patiko ir kitą darbą. Tada tų analizių rašymas reportavimas, kažkokių metrikų ir pan. Tai maždaug taip apibūdinčiau savo darbą.

I.: Tai sakei pvz., filtrą išleidžiate ir testuoji. Tai gali išsiplėsti ties tuo testavimu?

P.: Aha, o ką būtent norėtum sužinoti?

I.: Papasakok tiesiog procesą, savo darbo, kaip tu testuoji, ar filtras veikia?

P.: Ok, tai pirmiausia turbūt kaip ir testavimas, reikia išsirinkti, kurie vartotojai mums yra svarbūs. Pavyzdžiui, jeigu filtras yra susijęs su, tarkim, pridedam rašto filtrą, medžiagos rašto, tai galbūt jis bus aktualus mados pirkėjams, ta prasme, kurie perka rūbus, bet nebus aktualūs elektronikos pirkėjams. Tai tiesiog nuspręs, kurie vartotojai yra mums reikalingi ir tada iš esmės mūsų developeriai setup'ina, kad tam tikru metu tam tikroje vietoje vartotojas būtų paimtas į tą grupę ir iš tos pusės tada nusistatome šitą. Tada labai svarbu testavime yra parinkti metrikas, kurios galėtų mums parodyti, ar tie vartotojai gerai reaguoja. Negerai. Tai dažniausiai tos pagrindinės, iš kurių mes sprendžiame, tai jos būna artimiausios produktui. Jeigu tai bus filtras, tai bus, pavyzdžiui, filtro naudojimas arba filtro click throught rate'as, iš esmės konversija iš pamatymo, kiek vartotojų paclick'ina. Na ir aišku, dar labai svarbios būna ir verslo metrikos. Jas irgi pasižiūrim ir tada belieka apsirašyti dokumentaciją visą. Čia dažniausiai jau būna mano sritis aprašyti, koks yra oportunity size'as, kiek vartotojų, kiek vartotojų bus ištirti, kiek mums reikės laiko leisti. Pvz. būna vieni filtrai, labai mažai galbūt pasižiūrimi, kiti labai daug, tai nuo to priklauso testo ilgis, tai irgi čia viską nusprendžia. Ir tada, kai jau mums developeriai duoda žalią šviesą, kad viskas

jau yra sudėliota, sutvarkyta ir galim leisti, tada mes jau tą paleidžianm.

Tai va, aišku eigos metu netikrinam metrikų, nes veikiant pagal pagal statistikos visą logiką yra negerai. Bet labai svarbu tada, kad vartotojai būtų po lygiai grupėse ir pan. Ir paskui, praėjus tam laikui, pavyzdžiui, dviem trims savaitėms, pasižiūrime tas metrikas, parašome raportą ir tada nusprendžiame, ar mes tą naują filtrą, pavyzdžiui, paleidžiam visiems vartotojams, ar galbūt taisome, jeigu kažkas buvo netvarkoje. Jeigu vartotojai, pavyzdžiui, nesuprato, kaip suskirstytos raštų grupės, tarkim ar ne ten galbūt jiems nebuvo aišku, kas yra gyvūninės kilmės printas. Tai viską mes turime kaip analitikai, įvertinti ir tada pasakyti, ar mes darom kažkokius pakeitimus, nedarom, ar tas pakeitimas geras. Tai va, toks toks maždaug procesas yra.

I.: Aha. Dar turėsiu papildomų klausimų iš to paties. Tai pirmiausia gali man paaiškint, ką turi omeny su click through rate'u, ten pasakei tą, greit viskas judėjo.

P.: Jo. Click throught rate'as pati kaip metrika?

I.: Šitam kontekste, ką prieš tai sakei. Išsiplėsk biški, kad vėliau suprasčiau tiksliai, ką turi omeny.

P.: Aha, tai click-through rate'as iš esmės yra... Žodžiu, vartotojas tą filtrą ar ne, jis gali pamatyti ir gali ant jo paspausti. Tai yra du veiksmai, kurie mums rūpi. Click-through rate'as. Tai labai panašu, kaip ir iš esmės filtro pačio naudojimas, pvz. per sesiją. Bet tai parodo iš visų kiek iš tiesų, kiek kartų, kiek vartotojas pamato, kiek vartotojas paclick'ino to filtro. Tai pavyzdžiui, jeigu filtras būtų neaiškus žmogui ar ne, jis ten galbūt nesupranta jo iki galo. Tai jis gali labai daug, galbūt ir nespaudinėti, tiesiog pasižiūrės, nesupras, ką ten galima. Arba, pavyzdžiui, nebus opcijų ir jam reikalingų ar ne. Mes pridėsime gyvūninės kilmės kokį nors raštą, pridėsime spalvotą raštą ir, pavyzdžiui, nebus, tarkim, kareiviško rašto. Tai žmogus jo neras, tai galbūt irgi į tą galima atkreipti dėmesį. Jo, tai ta metrika maždaug jo. Iš esmės tai yra santykis iš visų kiek pamato tą filtrą, kiek iš tų packlick'ina. Click'us padalinant iš impresijų, tų pamatymų ir padauginus iš 100 procentų gauname click-through rate'o metriką.

I.: O kaip jūs užfiksuojat ar pamatė?

P.: Tai visam šitam yra event'ai iš esmės. Turbūt kiekviena įmonė ir kompanija turi daug visokios datos, kurią renka. Tai tam dažniausiai yra atskiri event'ai, kurie šauna. Nežinau, kaip paaiškinti event'ą, įvykius. Bet jo, tai žmogus tiesiog paspaudžia arba pamato ir mes... Ta prasme iš esmės turi tas filtras atsirasti ekrane. Ir vartotojas, jeigu jį pamato iš karto, sistema registruoja ir siunčia į duomenų sandėlį tą dalyką.

I.: Tada minėjai, kad kažko negali daryti, kai paleidi testą. Tai gali priminti, ko tu negali daryti?

P.: Tai, pavyzdžiui, būna noras žinai, paleidi testą. Pavyzdžiui, jis turi leistis dvi savaites, bet kartais norisi pasižiūrėti, kaip ten dabar viskas vyksta ir dažnai tai vadinama peak'inimu. Tiesiog žvilgtelėjau į tą būtent rezultatą, bet dažniausiai pirmos dienos... Ypač jeigu kažkokia nauja, naujas kažkoks pakeitimas. Ta prasme mes pridedame, pavyzdžiui, naują bloką, ar ne, ten home page, tai žmonėm būna labai jis naujas ir galbūt pirmom dienom mes galime pamatyti dėl to naujumo efekto vadinamo galime pamatyti labai didelį šuolį metrikose ir tada galima labai apsidžiaugti. Kai kurie žmonės, pvz. vat direktoriai, produkto vadovai nori galbūt sustabdyti AB testą ir iš karto pascale'int tą dalyką, nors tai yra tiktai nauji pamatymai ir pan. Ir tai nėra statistiškai svarbu, nes tai yra tik pirmos dienos. Tai gali tiesiog arba apsidžiaugti labai stipriai, arba nuliūst labai stipriai, bet kad tą tikrą efektą pamatyti, tai turi sulaukt žinai tos pakankamos tos statistinės galios iš esmės ir palaukti. Tiesiog mes nežiūrime į tuos rezultatus pradžioj bent jau savaitę, kad surinktume visą tą dinamiką visos savaitės dienų ir tada nu galbūt po savaitės jau leidžiam pasižiūrėti tuos rezultatus kas nori, bet pirmom dienom tai ne - draudžiama. Čia svarbu tik tai, kad nebūtų to sample ratio mismatch, kurį turbūt irgi galiu paaiškinti.

I.: Paaiškink, paaiškink.

P.: Jo jo, tai galiu papasakoti labai paprastais pavyzdžiais. Pavyzdžiui, turim populiaciją ar ne. Tai pavyzdžiui galėtų būti stiklainis, kuriame yra 10 mėlynų, 10 raudonų ir 10 žalių saldainių. Ir tai čia yra mūsų populiacija. Tai yra visas aruodas tų saldainių, kuriuos mes turim dabar. Ką mes darome? Pavyzdžiui, jeigu norime pasiimti kažkiek saldainių, mes tiesiog kišame ranką, traukiame ir kai ištraukiame, tų saldainių pasiskirstymas gali būti bele koks. Ta prasme, tu gali ištraukti, pavyzdžiui, tris raudonus, du žalius ir vieną žalią. Vieno mėlyno saldainio, tarkim, ane? Tai vat tai ir yra tas sample ratio miss Match arba kitaip tariant, vadinamas. Kur mes jau turim imtį tą, kuri papuolė mums į ranką, kuri nėra tas pasiskirstymas, jos nėra lygus populiacijai. Tai vat šito dalyko mums reikia išvengti, bet testuojant dėl to mes būtent tas vartotojų grupes turime labai atsakingai parinkti, kad tas pasiskirstymas. Pavyzdžiui, jeigu mes turim tris grupes, būtų 33, 33, 33. Ta prasme po lygiai arba jeigu tai yra per pusę, kad būtų 50 ant 50. Nes kitu atveju, jeigu turėsim skirtumą vartotojų skaičiaus, tai tiesiog negalėsime palyginti tų rezultatų, nes grupės bus nevienodos.

I.: Testus leidžiat būtent, būtent tu leidi testus? Tavo atsakomybė juos paleidinėt?

P.: Tai iš esmės iš komandos tą padaryti gali bet kas. Produkte dažniausiai yra inžinieriai, būna apsiskaitę. Mūsų darbas yra juos apšviesti, bet dažniausiai tai yra mūsų atsakomybė sukurti eksperimentą, jį aprašyti su visom hipotezėmis, pasirinktom metrikom. Kodėl mes tą darome? Argumentavimas, galbūt irgi paaiškinimas, kas tai per pakeitimas? Na, ir tada tą eksperimentą paskui jau jau leidi ant visų pavyzdžiui šalių. Tai va, jį paleisti. Pavyzdžiui, kaip yra schedule žodis, nežinau, kaip jį išversti.

Suplanuoti laiką jo paleidimo. Jo, tai šitą dalyką padaryt jau yra mūsų atsakomybė. O sustoję pas mus bent jau nežinau kaip kitose įmonėse, bet automatiškai parenki dvi savaites ir kokiu laiku prasidėjo tokiu laiku pasibaigė po, tarkime, septynių arba keturiolikos dienų.

I.: Dar yra kažkas, ką gali pridėti apie šitą visą testų kūrimo procesą? Ai, nu gerai. Arba tiesiog. Tai va, hipotezės, metrikas - tai irgi tu atsakinga už jū sugalvojimą?

P.: Taip. Tai kai mes pradedame galvoti kažkokį pakeitimą, tai pirmiausia reikia pradėti nuo problemos, ar ne? Tai problemą turbūt dažniausiai sugalvoja arba produkto žmonės, arba dizaineriai. Jie, pavyzdžiui, pamato, kad ten kažkoks laukelis yra labai ten keisto dydžio ir pvz. jo vartotojo pamatyti negali, tai jau ateina pas mumis su problema. Tada yra kalbama su inžinieriais. Kalbant iš esmės, visa komanda kalbasi. Ką mes galėtume padaryti, kad tai neatrodytų keistai ta prasme, kad atrodytų tas dizainas geresnis? Ir tada jau analitikų atsakomybė yra, kad mes vartotojus pasirinktume teisingai, kad neturėtume to, apie kurį prieš tai šnekėjome, kad hipotezė irgi būtų suformuluota teisingai. Pavyzdžiui, jeigu mes, tarkime, jeigu mes pridėsime šitą pakeitimą, mes pagerinsime vartotojo patirtį būtent toje dalyje ir tą mes pamatysime ir nuspręsime, kad tai įvyko. Jeigu pamatysime pokytį tokios ir tokios metrikos. Tai va taip vat susiformuluojam tą hipotezę, pasirenkam metrikas ir tada jau, tada jau yra mūsų dalis aprašyti ir panašiai. O inžinierių dalis yra visą šitą implementuot produkte.

I.: Su kokiais iššūkiais dažniausiai susiduri šitame procese?

P.: Taip, tai labai svarbu. Kaip ir minėjau, tuos vartotojus pasirinkti teisingai, bet labai dažnai tą teisingą parinkimą inžinieriai implementuot tingi, tai būna toks momentas, kad galbūt ten galima tiesiog visus aktyvius userius ir tarkim paimti. Bet tada aš turiu sėdėti jam ir aiškinti, kad na žiūrėkit, tai jeigu jūs dabar sutaupysite 2 valandas laiko, kol implementuotumėt tą vat event, ta prasme tą event'ą, kurio metu mes parenkame tą vartotoją, kai jis ateina jau į mūsų testą. Sakau tai aš. Jeigu jūs nepraleisite dabar tų dviejų valandų, tada aš praleisiu dvi savaites. Rankom tiesiog išsitraukinėdama. Na, žinai tuos vartotojus, tą segmentą, kuris man yra reikalingas. Pavyzdžiui, visi tie, kurie pamatė tą filtrą, tarkime, ir jo. Ir tada sakau man reikės rankom skaičiuot visas tas metrikas ir mes tada vis tiek vėluojame. Tada jau paleist feature'ą vėluosim deliver'int kaip komanda ir pan. Tai va, šitas reikalauja tokio labai didelio. Diskutavimo, nes yra tas noras iš inžinierių eit paprastuoju būdu, bet tada galima sakyt mums būna labai sunku ir mes jau automatiškai metrikų suskaičiuoti sistemoje nebegalime, nes jos yra labai atskiestos tokių vartotojų, kurie mums nelabai svarbūs. Šiuo metu tai šitas būna.

Kitas dalykas irgi yra dar reikalų pasiskaičiuoti testo trukmę. Tai pavyzdžiui, irgi ta prasme, jeigu ten filtras, turbūt geriausias pavyzdys yra tai jeigu jis, tarkime, yra ne pirmas, bet koks nors ten sąraše šeštas,

kur dar reikia vartotojui pascroll'int į šoną pasižiūrėti, kur galima tą filtrą susirast, tai jau iš esmės toje eilutėje, kur, kur matot, tas filtrų sąrašas. Kuo toliau, tuo vartotojų mažiau ten ją nubraukia į šoną ir pasižiūri. Tai reikia visąlaik pasiskaičiuoti, kiek jų ten apskritai ateina, ir pagal tai, kiek mes turim aktyvių narių, spaudžiančių per tam tikrą laiką, tai mes tada galime pasirinkti tą būtent trukmę.

Pavyzdžiui, su tokiais labai populiariais filtrais, pavyzdžiui, kaip kategorija, kaip dydis. Juos visi naudoja. Ta prasme, žmonės ten nusipirktų batų, ten nežinodami savo dydžio. Tai su šitais yra paprasta. Dažniausiai savaitės laiko užtrunka pamatyti pakeitimą, bet jeigu tai yra kažkoks naujas ir kažkoks labai nišinis, tai dabar elektronika paleidome ir buvo sim ar SIM yra užrakinta su pririštas prie tiekėjo telefonas. Tai ten tokie labai nišiniai ir tada juos reikia testuoti jau tris savaites. Keturias, ilgiausiai, keturias. Daugiau ten jau nebeapsimoka tas leisti. Tai va, tai tokie turbūt iššūkiai didžiausi.

Tai labai darbas, reikalaujantis politikos. Daug daug ginčų su komanda. Galiu net iš tiesų pavyzdį duoti Situacija viena. Kur? Kur irgi leidom, leidom filtrą ir tada kažkaip gavosi, kad vartotojus parinkome ne ant vietos kur filtras yra, bet ant fakto, kad tas filtras egzistuoja. Ta prasme, pvz on grupė, kuri tarkim matė rašto filtrą, tai jie ta prasme atsirado pas mus eilutėse, bet pvz. off grupė, kur tas filtras turėjo būti. Mes tiesiog tada kažkaip žiūrėjome, kad neuždėjo inžinieriai ant fakto, kad būtų vartotojai pamatę. Ir tada po trijų dienų mes atsidarome pasižiūrėt ar to sample ratio missmatch neturim ir žiūrim, kad 70% on grupė ir 30% off, nes jie kažkaip atsirado, bet bet kokiu atveju yra didžiulė nesąmonė. Tai tada grįždami pasišnekėjome, kad nesusišnekėjom. Tiesiog praeitą savaitę, kai jau buvo testas paleistas, reikėjo pataisyt ir tada jau buvo viskas tvarkoj. Tai tu jo reikalauja tokių checklist'ų ir taip susidėti tą procesą. Ta prasme, ką vat pasitikrinti kiekvieną kartą, kad viskas būtų gerai.

I.: čia off grupė?

P.: Ta grupė yra kontrolinė, kontrolinė grupė. Ta prasme, mes. Jeigu paprasčiausi pavyzdžiai, tai turim OFF grupę, kuri mato viską taip, kaip yra jau egzistuojančiai platformai. O ON grupė tai jau yra ta testinė, kur mes įvedame, ta prasme tą naują pakeitimą. Taip.

I.: Supratau. Tai šitie dalykai iš esmės yra tavo darbas. Visi šitie testai tarsi?

P.: Hmm. Tai mes taip pat darome ir visokius Discovery mes vadiname tokius dalykus, tai būtent ieškom, ką būtų galima pakeisti, pridėti prie platformos ir pan. Tai dažniausiai būna galbūt vartotojų elgsenos analizė, jeigu mus, pavyzdžiui, domina. Nežinau. Labai geras pavyzdys, kur daro about you berods turi ten pavyzdžiui yra kategorijų, ten visas sąrašas, kur tu, pvz. rūbai, batai, aksesuarai ir pan. Kartais jie prideda dar vieną kategoriją kalėdinę ar ne tai, kad tą kategoriją kalėdinę paleist reikia žinot

būtent tą vartotojo elgesį? Ko jis tuo metu būtent ieško. Galbūt, jeigu tai yra Kalėdos, tai yra, pavyzdžiui, šaltas sezonas, tai tada reikia ateiti pasižiūrėti ir kiekvieną kategoriją, kiek vartotojų tuo metu būtent palyginus su visu kitu metu, kas nėra sezonas ieško, pavyzdžiui, ar yra kažkokie įdomesni paieškos tie tekstai, ar ne? Ar ieško kalėdinio megztinio, pavyzdžiui, arba elfų ausyčių? Tarkime, tokių visokių dalykų? Tai tiesiog ką reikia padaryti, padaryti tą Discovery išsianalizuoti, palyginus su visus kitus sezonus. Taip vyksta, pavyzdžiui, kalėdinis. Tada galima parinkti būtent tas kategorijas. Galbūt užtenka tiesiog įdėti šiltų megztinių kategoriją, bet nebūtinai reikia lįsti giliau, ta prasme į kažkokius gilesnius lygius, kur būna megztiniai su kaklu, megztiniai su keistom rankovėm ir panašiai. Tiesiog galbūt užtenka megztinius sudėt, nes vartotojai giliau nelenda, tai irgi galima tokia papildoma navigacija pridėt ir tam būtent reikalinga išanalizuota vartotojo elgesį, kur jisai spaudinėja, galbūt iš kurios labiausiai perka, galbūt yra nenaudojamų kategorijų tuo metu, tai jas net neapsimoka rodyti. Tai va, tokius dalykus irgi darome.

Ir taip, dar kitas yra, toks labiau report'ingas jau. Jis turbūt vyksta kiekvieną mėnesį, kur mes parodome komandai arba direktoriams irgi. Jeigu jau domenų lygio būna, tai metrikas kaip mūsų komanda iš esmės ten. Jeigu aš dirbu su browsing, tai yra būtent navigacija per kategorijas, tai kaip būtent browsing tos metrikos? Vienu žodžiu kinta per laiką. Ta prasme, ar jos auga? Ar matom kažkokį pokytį vartotojų elgesyje, į kurį mes turėtumėme atkreipti dėmesį, tai tokius darbus irgi darome. Testavimas turbūt didžiausia dalis viso šito yra produkto.

I.: Ar koreliacija vis dar nelygu priežastingumui?

P.: Taip. Taip, jinai vis dar nelygu priežastingumui. Nu tai dėl to iš esmės mes ir darome tą testavimą. Pavyzdžiui, jeigu mes pridėtume kažkokį naują filtrą ir tiesiog matytume, kad metrikos tam tikros kyla, tai mes nežinotume, kodėl jos kyla. Galbūt vartotojai tiesiog daugiau pradėjo spaudinėti ant kažko. Tai dėl to mes ir darome testavimus, kad būtent pasirinkę tam tikras metrikas, mes pamatytume, ar tai yra tikras vartotojų pokyčių, vartotojų elgesio pokytis. Tuo metu per tą laiką su tuo pakeitimu ar galbūt jam apskritai yra nesvarbu. Tai vienintelis dalykas, kas mums gali pasakyti ar vartotojai džiaugiasi ir nesidžiaugia pakeitimais. Tai būtent yra tas langas. Ir šiaip visokių kitokių testų.

I.: Tikėjaus šito atsakymo, nes vis tiek iš tos pačios vietos jau kalbėjau.

P.: Mes data žmonės labai griežtai šituo klausimu esam.

I.: Nu pas jus kompanijoj.

P.: Taip. Turbūt. Bet ir visose kitose didelėse, kur testuoja. Pvz. mums turbūt labai geras pavyzdys yra Netflix. Kai važiuodavome į tas konferencijas, ten savo verslo kelionės, tai būtent ieškom Netflix,

nes jie labai daug testuoja su skirtingais visokiais. Galbūt ne AB testu, galbūt ten kitokiais, visokiomis technikomis. Tai mus irgi labai domina, nes, aišku, ten tas pats testavimas kainuoja brangiai tris savaites mokėdavo už papildomus niuansus, sukūrimas ir pan. Tai tiesiog yra brangu. Yra daug visokių kitokių technikų. Bet dabar pas mus bent jau pats populiariausias yra tas AB testas, kur tiesiog lyginant grupes.

I.: Kada ir kokiose situacijose galvoji apie galutinį naudotoją?

P.: Visada, visada... Yra skirtingo požiūrio. Yra analitikų, kuriem galbūt nėra tas labai svarbu. Galbūt daugiau į metrikas, kaip greičiau pasibaigt tuos darbus, kad būtų galima eit kitus pakeitimus daryti. Man galutinis vartotojas visada svarbu ir turbūt tai iš dalies yra dėl to, kad aš pati (kuriamu produktu) naudojuosi ir dažnai visos idėjos kyla iš to, ko man pačiai reikia. Tai jeigu jeigu aš noriu dabar susirasti tą zebrinę ten kokia nors palaidinę, tai man. Man reikia to filtro ir jeigu aš jo neturiu, tai mane iš karto erzina. Tai va. Bet šiaip mes visą laiką ta prasme galvojam ir aišku, kartais atsiranda ir tų tokių dalykų, kad reikia taip laviruoti tarp galutinio vartotojo ir ir mūsų kompanijos interesų. Kai turbūt dabar labai geras pavyzdys, galima ir Reddit'e iš tiesų pasiskaityti. Labai dabar skundžiasi Vokietijos žmonės, kad išjungia šitą leidimą palikti atsiliepimus, jiems tiesiog, jeigu jie, pavyzdžiui, neperka per app mūsų. Nes problema su Vokietijos žmonėm yra, kad jie labai mėgsta grynus pinigus. O mums, kaip kompanijai, tas nelabai tinka, nes tada atsiranda tos šešėlinės transakcijos, kurių mes negalime užfiksuoti. Ir ypač per tuos dabartinius naujus pakeitimus įstatymo Europos, kur reikia pranešinėti jeigu žmonės daug parduoda, tai mums tiesiog tai nėra suderinama su, tas grynas pinigas, su mūsų įmone. Tai tiesiog dabar keičiame vartotojų elgesį. Aišku, jie yra labai nepatenkinti ir ta prasme dėl to ten sakau. Reddit'ą pasiskaičius labai labai juokingos reakcijos yra, bet kadangi reikia laimėti tą Vokietiją ir kad ten galėtų žmonės naudotis, kad galėtų pirkti iš Prancūzijos, pavyzdžiui, ir pan. Ir kartu prancūzai galėtų pirkti iš Vokietijos ir kad neatšauktų jų pirkimų, tai tiesiog čia buvo pagalvota daugiau apie kompanijos interesus. Pavyzdžiui, jeigu kažkuri dalis vartotojų atkris, tai čia yra rizika, kurią mes sutinkame priimti. Tai va, čia nėra kompanijos paslaptis. Čia viską galima internete rasti visus skundus.

I.: Kada dar tenka daryti kažkokių tokių kompromisų, laviravo tarp skirtingų poreikių implementacijų ar ko nors. Bet kokioje srityje.

P.: Taip. Irgi galėtų galbūt būti pavyzdys. Dabar elektronikos nauja kategorija, kurią mes paleidome prieš du mėnesius. Berods tai, tarkime, galbūt mes kaip komanda nematėme prasmės ten pridėjinėt tų kortelės, pavyzdžiui, atrakinimo prie telefono filtrų, nes tiesiog žinojome, kad žmonės dažniausiai pagal ką ieško pagal kainą arba pagal naujumą telefoną, bet tiesiog reikalavimas buvo toks, kad tie filtrai būtų ir jie. Tai mes galbūt su tuo nesutikome, bet turėjome juos pridėti ir kai kuriose net šalyse buvo skirtumas, kad vienose ar daugiau filtrų geriau veikia. Vartotojai labiau įsitraukia, būna į tą

paiešką kitose šalyse priešingai. Galbūt to vieno užteko, kad jie nori susirasti tą ikoną ir viskas jiems tas vienintelis modelio filtras buvo reikalingas, bet mes vis tiek leidome visus, kadangi tai buvo tiesiog reikalinga pagal dizainą ir pagal tos elektroninės specifikacijos tuos reikalavimus iš esmės, kad galėtų patvirtinti.

I.: Čia darėt viską dėl to reikalavimo?

P.: Taip taip, nors galbūt iš rezultatų tai nebuvo nei kažkoks statistiškai reikšmingas pokytis, nei labai reikalingas tokiam vartotojui.

I.: Koks faktorius turi daugiausiai įtakos galutiniam rezultatui? Produkto.

P.: Produkto ar to fakto, kad mes paleisime pakeitimą, ar ne?

I.: Gal produkto.

P.: Mums, kaip komandai, tai būna svarbu būtent tos mūsų produktų metrikos konversijos, pvz. į pirkimą. Tie būtent click'ai vartotojų, kaip jie naudoja mūsų tuos filtrus ir panašiai. Įmonei jau yra svarbu tos daugiau North Star metrikos. Tai ten iš esmės viską, kas galima išreikšti, išreikšti pinigais. Jei mes pakeliame piniginę vertę. Tai mes darome viską gerai. Jei piniginė vertė nekyla, tada jau galvokite, kaip ją pakelti iš esmės. Tai viskas turbūt kaip ir visur, atsiremia dažniausiai į pinigus, bet mes to daugiau išsprendėme. Šiaip labai gera knyga. Vadinasi Trustworthy Online Control Experiments labai gerai paaiškinta. Ir mes irgi taip apsisprendėme su direktoriais šitą klausimą, kad jeigu mes tas produktų metrikas pakelsime, tai jos dažniausiai turi tą teigiamą koreliaciją su visomis kitomis jau pinigus apimančiom verslo metrikom. Nes jeigu vartotojams patiks naudotis, jeigu jie spaudinės, pirks ir jiems bus paprasčiau daiktą rasti, tai jie, ko gero, išleis daugiau pinigų ir daugiau transakcijų vykdys ir dėl to visiems mums bus geriau. Tai va. Tai taip turbūt.

I.: Nuo kokių asmenų, jau veikėjų, daugiausiai priklauso tada galutinis produktas? čia net ir priimant kartu naudotoją kaip galimą atsakymą. Bet ten kartu inžinieriai visi, tu ir panašiai.

P.: Om. Mes jeigu ta prasme testavimo būdu, turbūt dažniausiai žiūrime į rezultatus ir jeigu rezultatai yra geri, mes tą produktą keičiame ir tuos pakeitimus įvykdom. Bet bendrą dizainą, kaip atrodo tas būtent platforma pati ir kaip ten viskas turėtų veikti, tas patirtis. Tai jau priklauso nuo turbūt generalinės dizainerių komandos, kurie ten sumąsto, kaip viskas turi būti. Na ir aišku nuo stakeholder iš tų visų direktorių, kurie nusprendžia ar apsimoka dabar keisti branding mūsų, ar ne.

I.: O vat šita visa jūsų darbinė komanda inžinieriai ir data, kiek turi įtakos viskam. Tam, kaip galiausiai atrodo.

P.: Tai daug. Mes ta prasme irgi esame tie, kurie būtent mūsų komandos dizainerė. Mes esame atsakingi už tą katalogo navigaciją. Ji tiesiog vaikšto per mūsų programėlę, žiūri, kas gražiai atrodo, kas

negražiai ten atrodo ir daro šitas apklausas vartotojų, ta prasme, tiesiog vat ant gyvų žmonių testavimą. Tiesiog grupė susirenka ir duoda naudotis. Su viena dalim, duoda naudotis su kita ir tiesiog klausinėja, kas žmonėms patinka, kas nepatinka. Jeigu tie mūsų pasiūlymai yra labai gerai uždokumentuoti ir ta prasme, jeigu mes suformuluojam tą verslo problemą gerai, tai mums nėra problemos nuo kito ketvirčio įsitraukti šitą pakeitimą į savo darbų sąrašą ir jį daryti. Bet irgi dažnai būna tokių situacijų, kada jau iš viršaus pavyzdžiui, ten yra tokios aukštesnės komandos, kurie analizuoja. Būtent kitų. Kaip kiti pavyzdžiui, ten tos e-komercijos platformos veikia, tai irgi, pavyzdžiui, nusprendė, kad elektronika čia galėtų būti mūsų nauja kryptis. Pradžiai buvo dizainerių prekės, dabar elektronika. Tai galėtų būti mūsų nauja kryptis. Ir tada tiesiog mūsų domenui, kuris atsakingas būtent už tą veikimą. Su navigacija atnešė ir pasakė dabar mes darome šitą, tai tiesiog darėm tada, ten implementavom tuos filtrus naujus.

I.: Kur matai daugiausia etikos rizikos?

P.: Pas mus nelabai. Aš jos kaip ir bent jau savo komandoj aš nelabai pastebiu, bet yra. Yra kita kita dalis įmonės, kuri dirba su su pasitikėjimu, su trust'u ir įmonės saugumu, ten jau yra reikalų. Aš nelabai galiu išsiplėsti, nes pilnai nežinau. Ten yra tas darbas su visokiomis jautriomis nuotraukomis, kur žmonės vieni kitiems siuntinėja. Tai čia gali būti toksai daugiau, kad reikia labai duomenis saugot ir jų kaupti negalima ten pagal GDPR ir pan. tai čia ko gero labiausiai turim saugotis, kad niekas nenutekėtų ir neišeitų niekur. Tai va. Na ir aišku kadangi naudojam visus jų duomenis, tai irgi turbūt negali matyti žmogaus vardo, pavardės, tai turi būti anonimizuota ir panašiai. Tai čia irgi svarbu, kad šitas neišplauktų, kad paskui nežinotume, kuris ten vardenis Pavardenis ko ieško, nes tai jau yra jautresnė informacija. Na ir aišku ten jeigu kažkokie yra pakeitimai, kaip pavyzdžiui nežinau rekomendacinių sistemų, tai irgi yra jau žmogaus įpročiai, žmogaus kažkokie elgesio modeliai. Tai jie ten, kaip suprantu, irgi ten labai stipriai analizuoja ir tada naudoja ją ten teikdami kažkokius pasiūlymus ir panašiai.

I.: O tai, kad žmogaus įpročiams daroma įtaka. Kaip, tavo nuomone, tai etinė problema, ar ne?

P.: Jo. Jo. Čia, čia manau, kad galėtų būti etinė problema. Pati aš irgi dabar pavyzdys - Facebook'as, ar ne? Aš dabar nesutikau, kad man personalizuotas šitas duotų reklamas. Tai dabar man jie liepia tas reklamas žiūrėti penkias sekundes dėl to, kad aš tiesiog neleidžia jiems naudoti savo data, tai nemanau, kad čia labai etiškas dalykas šito daryti. Tai vat, tiesiog nebesinaudoju feisbuku. Apskritai. Jie iš esmės su tokiu savo elgesiu prarado vartotoją, bet aš tuos dalykus suprantu. Bet galbūt tėvams, pavyzdžiui, ten žmonės, kurie nedirba su duomenimis, jie sutinka su daug visokių dalykų, kad apie juos būtų kaupiama. Ir man tai nėra labai geras dalykas. Ar dar su kuo aš labai nesutinku, kad negalima ištrinti apskritai, pavyzdžiui Messenger žinučių. Jos, ta prasme lieka, nes tai yra dviejų žmonių reikalas arba nežinau septyniolikos, jeigu tai buvo grupinis susirašinėjimas. Ir tai vis tiek yra serveriuose ir kad ir kaip tu turi

tą teisę - irgi Europoje, kad ištrint apie save visus duomenis - šitų duomenų ištrinti negalima. Tai labai nepatinka. Mes net ir forumą, pavyzdžiui, uždarėme. Ten irgi buvo paskelbta, kad jisai yra uždaromas, nes ten tiesiog žmonės daug visokių dalykų rašė ir dalinosi, tai mes tiesiog nenorėjome jų laikyti pas save.

I.: O pačioje kompanijoje stengiatės, kad jūsų kompanija netaptų Facebook.

P.: Tai vat čia tas yra laviravimas tarp šitų dalykų ir. Bet šiaip labai iš tiesų patinka, nes (anonimizuota darbo vieta) visai turi moralinį kompasą. Kartais praslysta tam tikros reklamos. Vienu momentu, buvo lažybų pasirodę. Bet mes iš karto pranešėme, kad tai nėra susiję nei su mūsų kompanija, nei su kažkokia kryptimi, kurią mes turime. Tai tos reklamos buvo labai greitai pašalintos. Tai va, ten, kaip suprantu, irgi yra uždėtas...

I.: Jūs kaip darbuotojai pasakėt darbe, kad jums nepatinka, kad reklamavosi lažybos?.

P.: Taip taip, tai tiesiog. Arba pavyzdžiui kažkokie Aliexpress reklamos ar Temu kur tai yra ta prasme tas fest net nepavadinčiau fashion. Aš net nežinau, kažkokia greitų šiukšlių parduotuves žinai, kur, kur irgi buvo atsiradę reklamos. Ir mes kažkaip visiems sakom, kad mes esam ten jokios greitos mados ir panašiai. Va tada pas mumis yra reklamuojami Aliexpress, kur yra didžiausia šiukšlių platforma apskritai pasaulyje. Tai mes iš karto pasakėme, kad čia yra nesąmonė ir juos buvo pasistengta pašalinti. Kitas dalykas irgi pas mus iš tos etinės pusės, kas yra labai svarbu. Tai, pavyzdžiui, yra draudžiami kailiniai. Mes negalim pardavinėt, pavyzdžiui, kailinių, vien todėl jų atsiranda. Bet jeigu kažkas praneša arba jeigu mes patys randame, tai mes turime vidinius kanalus, kur pasidalinome, kad va, žiūrėkit, radom kailinių, bet pas mus nelegalu parduot juos yra pagal mūsų nuostatas vidinės taisyklės, platformos juos tiesiog išima.

I.: Kaip apibūdint tavo technologinę specifinę, kurią tu? Na aš suprantu procesus, kuriuos tu darai, bet kokį žodį įdėt. Data science ar ne?

P.: Tai yra data analytics labiau. Nes data scientists, jie daugiau kuria modelius, jie daugiau yra inžinerinės pakraipos. Aš daugiau jungiu produktą su duomenimis. Iš esmės. Tai mes viską, ką pakeičiame ir padarome, aš iš esmės turiu iš duomenų pasakyti, ar tai veikia, ar neveikia. Tai va tas noras būti data driven. Tai mano atsakomybė yra, kad mes tokie būtume, kad tai nebūtų kažkokie keisti dalykai daromi.

I.: Tai tada, kaip tu apibūdintum, kas yra gerai daromas data Analytics darbas, o po to apibūdink, kas yra blogai daromas analitikos darbas. Kokie bruožai?

P.: Aha. Nežinau, manau, kad analitikoj, jeigu analitikas yra geras, tai visų pirmiausia. Vat svarbiausia šiame darbe yra apskritai mokėti paaiškinti sudėtingus dalykus labai paprastai. Jeigu mes

rašysime savo sunkiomis frazėmis duomenų analizę, tai ją paskaitęs turi suprasti bet kas? Ta prasme ir inžinierius, kuriam galbūt ta matematika nėra labai svarbi ir tie skaičiai, ir direktorius, kur ten jie, galbūt vadybą baigę ir pan. Apskritai su statistika nieko bendro nėra. Tai reikia labai paprastai tiesiog viską aiškinti ir gebėti tą parodyti. Blogas analitikas turbūt parašytų viską labai akademiškai ir ten su aukštomis frazėmis ir niekas nesuprastų, galų gale, ką jis nori pasakyti.

Kitas dalykas irgi labai svarbus yra vizualizacijos. Jeigu mes kažką rodome ir aiškiname paveikslais, tai jie turi būti suprantami. Ta prasme, tai nėra būtinai daug visokių kažkokių technikų ten parodyti, kad moki šitą smuiko grafiką nupiešti tai dažniausiai nėra reikalinga. Turbūt. Aš nesakau, kad viską galima tiesiog su paprastu stulpeliu diagramą paaiškinti ir ji geriausiai viską parodo su pora linijų. Tai irgi yra labai svarbu. Tada irgi tas gebėjimas komunikuoti ir su aukštesnėm komandom, ir su visais žmonėmis tai turbūt. Gebėjimas. Nežinau, net čia net ne. Dabar tokia diplomatija, kad jeigu, pavyzdžiui, mes nematome prasmės tam tikram dalyke, tai mes turime gebėti tą pagrįsti iš duomenų. Tai neturi kelti kažkokio - ai nu, aš čia manau, kad čia yra nesąmonė ir čia mes to nedarysime.

Tai kartais tiesiog žmonės turi labai tokią atmetimo reakciją į dalykus. Reikia gebėti su jais kalbėtis ir pasakyti, kad žiūrėkite, mes suprantame jūsų motyvaciją, bet mes manome, kad tai yra neteisinga. Ir prašom jums skaičiai, kurie parodo. Kitas dalykas reikia labai mokėti priimti grįžtamąjį ryšį. Nes visą laiką bus protingesnių kolegų, kurie galbūt daugiau žino. Kartais irgi tenka susidurti, kad padarome kažkokius pakeitimus iš karto. Pakeitimams priešinamasi yra tam tikra prasme sako va čia yra nesąmonė, nors net nepabandė ir ir nepadarė. Tai va ir dar kas yra labai svarbu, tai labai tiesiog mokytis ir priimti dalykus tame darbe, nes irgi tarkim vienu momentu, kai aš tik prisijungiau ten, buvo galima pas mus analizę su Skala, su Python, su R daryti, rinktis ką tik nori. Dabar kompanija vis tiek standartizuota, kad mes visi turėtume panašius procesus, kad galėtume vieni kitų darbus daryti, nedaryti, skaityti, suprasti tą kodą ir panašiai. O tai jau turbūt suprantama savaime, kad reikia kažkaip vienodai tą procesą turėti. Tai irgi labai priešinosi. Pavyzdžiui, Python mokytis tie žmonės, kurie galbūt univere visą laiką mokėsi su R. codint. Ir aš to dalyko neturiu. Ta prasme man. Aš pasakiau sau, kad aš esu kaip kaktusas - aš augsiu bet kokiose sąlygose. Ta prasme, ir ką man reikės aš tą išmoksiu. Bet pastebėjau, kad būtent iš tų labiau senior žmonių, kurie ten jau aukštesnėse rolėse yra tas didesnis pasipriešinimas. Kas čia per nesąmonė? Kodėl aš negaliu dirbt? Tai vat tas toks irgi prisitaikymas prie tų kintančių sąlygų irgi yra labai svarbu. Ir ypač dirbant tokiose didelėse įmonėse, kada būna tos reorganizacijos, keičiasi dalykai. Tai irgi labai svarbu yra dinamišku žmogum. Man pradžioj, kai pradėjo reorganizuoti gal prieš 2 metus, tai buvo šokas.

Ta prasme, kaip čia galima taip puikiai dirbama. Dabar keičia viską ir čia bus nesąmonė. Aš labai asmeniškai priėmiau, bet nereikia to priimti asmeniškai. Ta prasme pasaulis keičiasi. Galų gale, jeigu žinai, jeigu norėsi pereiti į kitą įmonę, bus lengva, nes tiesiog pas tave kontekstas kas pusmetį keičiasi. Juk tai tai tiesiog reikia būti tokiam dinamiškame, prisitaikant prie dalykų ir tiesiog nebijoti augti, nebijoti mokytis, domėtis. Argi labai svarbu? Tai va jo. Mokėti kritiką, priimti konstruktyvią. Jeigu yra konstruktyvi, tada ne. Bet čia yra labai svarbu. Tai blogas analitikas, sakyčiau, to nedarytų. Ir dar išvertęs akis sakytų, kad mažas statistinis reikšmingumas čia jau yra, iškart viską parodo, nors neparodo, nes net neparodo. To. Ir dar labai svarbu šitam darbe pasiimti atostogų ir pailsėti nuo visko. Taip. Nes jeigu jeigu žmonės, ypač dar kur, korporacija, bet dar nori startup'o tokio jausmo aplinkui, tai būna labai pervargsta. Ir tada tiesiog iš savo patirties kalbu. Ta prasme buvau perdegus ir teko pailsėti, tai reikia atostogų imti. Žodžiu.

I.: Kaip gerai patį naudotoją atitinka visi duomenys, visi modeliai. Visa šitas algoritmas, jo reprezentacija.

P.: Šiaip labai turbūt priklauso nuo produkto. Jeigu tai yra tiesiog navigacija ir ten pasispaudinėt, kur susirasti suknelę, tai ji visiems yra vienoda. Bet yra ir kita projekto dalis. Aš prie jo nedirbu, tai daug ten labai nežinau, nesigilinau. Bet yra tos būtent rekomendacinės sistemos, kur būtent pradžioj tai buvo tiesiog pagal galbūt pasektus brandus kažkokius ir pan. Dabar tai jau yra daroma, kad pagal tai, ką žmogus spaudinėja, ieško, labiausiai mėgsta, pagal tai būtų tos rekomendacijos daromos. Tai jis tas mūsų produktas po truputį, po truputį. Jis toks vis dar darosi labiau atliepiantis žmogų, bet su rekomendacinėm sistemom yra toks dalykas. Kad yra tas tos šaltos pradžios problemos, tas cold start problem, kad jeigu žmogus nėra nieko ieškojęs prieš tai, tai labai sunku yra kažką pasiūlyt, tai turbūt tokius bendrus dalykus daugiau siūlo ir ypač su tais modeliais irgi būna, kad jeigu tai yra berods iki 5 ar iki 10 daiktų, tai jeigu žmogus nusipirko šlepetės, šlepetės ten ilgai ir nuobodžiai rekomenduos pagrindiniam puslapy dėl to, kad tiesiog nėra daugiau dar ir ko rekomenduoti, tai turbūt dažniems vartotojams, aktyviems vartotojams ta patirtis būna geresnė, bet ir taip turbūt labiau pastebim, kad nauji vartotojai daugiau tiesiog patys ieško. Jiems ta prasme pati programėlė būna įdomesnė, jie mokosi ja dar tik naudotis. Tai jie turbūt tokią patirtį turi bendrą, kur nelabai ten jiems pritaikyta. Bet irgi yra vat galimybė pas mus susipersonalizuoti. Pavyzdžiui, jeigu ten nueisi į pasirinkimus tuos, tai galima pasirinkti, kad kažkokį brendą pafollow'int. Arba galima dydį pasirinkti, kurį galbūt dažniausiai nori matyti ta sistema. Jeigu tiesiog paieškosi suknelės - nežinau, kodėl tu galėtum suknelės ieškoti - jeigu paieškotum, tai tau turbūt duotų tavo dydžio sukneles, o ne kažkokias per dideles arba per mažas.

I.: Jau aš klausiau dalinai, bet gal netyčia dar turi ką papildyti. Kur matytum kur yra grėsmių daugiausiai kokybės užtikrinimui darbo?

P.: Darbo čia mano darbo.

I.: Jo jo jo.

P.: Ai ok. Tai matyčiau didžiausią problemą. Jeigu analitikas komandoje yra vienas, tai vis tiek būna, kad kartais praleidžiame ten tam tikrų skaičiavimų. Galbūt klaidą kode padarome. Tai manau, kad tie peer reviews vadinami yra būtini mūsų darbe. Tai šitą problemą mes irgi turėjom ir labai stipriai išreiškiam, tai dabar irgi mus irgi reorganizavo, tai reikalaujame, kad kiekvienoje komandoje būtų bent jau po 2 žmones, nes tada turi kažkokio palaikymo. Gali paprašyti, kad analizė peržiūrėtų kodą peržiūrėtų ar viskas ten yra sąmoningai parašyta ir tvarkingai? Tai taip. Labai lengva užsisakyti iš tiesų į savo ratą, nepastebėti tam tikrų dalykų ir tada pateikti rekomendacijas, kurios galbūt nelabai naudingas planas. Tai turbūt didžiausia problema yra likti vienam toje srityje, bet yra kas priešinasi, pavyzdžiui, kad kam čia, kas čia, kodėl čia mano kažkas turi peržiūrėti tą darbą? Aš čia gerai dirbo ir taip labai asmeniškai priima, kad čia yra puolimas, bet tai yra tiesiog kokybės užtikrinamas iš esmės ir taip.

I.: Gerai, kas būna, jeigu artėja deadline ir nespėjote?

P.: Pranešame, kad nespėsime ir viskas su tuo tvarkoj. Na ta prasme, jeigu tai jau degantys dalykai, kad pasakyta, kad tam tikra dieną turi būti paleista nauja kategorija ir mes būtinai tą dieną turime pradėti pardavinėti telefonus. Tai tada jau visi kiti darbai yra nukeliami ir komanda dirba prie to, kad mes pakeistume tą dalyką. Bet jeigu tai yra kažkokia paprastesnė ta prasme misijos, epic, tai kaip pavadinsi, taip nepagadinsi. Tai tada jau pasakai tiesiog ta prasme, kad iškomunikuojame. Galbūt per tą mėnesio susitikimą su direktoriais, kad šita misija vėluoja tiesiog dėl kitų prioritetų, kitų priežasčių nespėjom. Su tuo irgi yra viskas tvarkoj. Mes bent jau mūsų komanda yra daugiau už kokybę negu kiekybę. Tai jeigu tam reikia papildomos savaitės, tai viskas tvarkoj. Su analitiniu darbu yra lygiai tas pats. Mano paskutinis mėnesis buvo labai chaotiškas, nes dirbau vieną, tada komandoje. Tą mėnesį tai reikėjo laikyti labai daug kampų. Kadangi kolegė atostogavo, tai irgi. Pavyzdžiui, turėjau pradėti analizę gal prieš savaitę. Aš pradėjau ją savaite vėliau. Bet svarbu labai laiku informuoti ne tą pačią dieną, galbūt prieš tris, kad žiūrėkit, dabar yra kitų prioritetų. Aš neturiu laiko. Aš esu ten viena arba būsiu vertusi kažkuo. Kažkas dega, kažką reikia duomenų kaupimo sutvarkyti ir pan. Tai tiesiog visiems yra viskas su tuo tvarkoj ir jo. Tai galbūt tada tiesiog preliminarius testo rezultatus užmetė akį pamatai, kad viskas pozityviai arba negatyviai ir tada tiesiog priėmė sprendimą, o jau patį analizės raportą gali pristatyti ir vėliau. Dėl to nėra problemos.

I.: O tais atvejais, kai sakai, kad jau būtinai rytoj. Nu ne rytoj, kažkada, reik paleisti ir tada visas dėmesys į tai. Būna, kad jeigu tada nespėjat, pavyzdžiui tektų kažkokių nuolaidų daryti trumpalaikių, kad ir.

P.: Būna, būna ta prasme, kad ir nukelia kelioms dienoms. Pavyzdžiui, jeigu ten paleidi tam tikrą pakeitimą, naują kategoriją ir pamatome, kad yra labai jau ten daug tų bug'ų palikta, tai taip tada stabdo tą paleidimą ir tada jau eina kita tvarka. Bet tai vėlgi būna fokusas vien tik į šitą dalyką. Kiti darbai būna padėti į šoną ir mes dirbame prie to. Bet dažniausiai planavimas tikrai yra geras. Ir tokių kažkokių extra atvejų net nebūna. Per paskutinius metus net nemačiau. Ta prasme. Jeigu mes matėme, kad kiti darbai mums trukdo, mes tiesiog fokusavomės tas dvi ateinančias savaites tik į tam tikrą sritį. viską darėme, kad tai būtų padaryta kartu su visom kitom komandom.

I.: O tokių išimčių, kad, pavyzdžiui. Gerai, vat truputį apmažinsim funkcionalumų kiek, bet vis tiek kažką paleisime į prodą. Atrodo, kad bent kažką turėtum.

P.: Om. Negirdėjau tokio dar. Bent jau mano mano darže nepasitaikė.

I.: Kas tau darbe kelia stresą?

P.: Viskas. Nu (juokiasi). Stresas galbūt nežinau, man dažniausiai su prezentacijom galbūt. Aš šiaip esu toksai, adaptuota introvertė. Esu išmokusi bendraut su žmonėmis. Tai kai būna tas tiesiog visas dėmesys į mane ir man reikia kalbėti, tai man tiesiog labai daug nerimo sukelia. Irgi tas galbūt, kodėl tas analisto darbas, jis yra labai smagus, nes gali daug, daug visokių dalykų veikti. Aš galiu daryti darbus ir duomenų analitikos inžinieriaus. Aš galiu naujas lenteles rašyti, kurios mano analizėm padės, bet taip pat aš galiu ir ten AB testus analizuoti. Tai man, kaip labai dinamiškam žmogui, šitas yra labai smagu, bet kartu šitame darbe yra ir labai daug politikos. Ta prasme tikrai būna pasakai, paskelbi rezultatus, ateina kažkas labai išsilavinęs ir klausia o kodėl ne šitą metriką analizavo? O vat šitą ta prasme. Tai gal jūs šitą paanalizuokite? Ir vat pradeda iš esmės net žmogus, kuris su analitika net nesusijęs, tau pradėti aiškint, daryti tavo darbą žinai. Tai kaip dirbt? Tai toksai vat irgi tas pats būna, kad labai reikia laviruoti. Būna tokių žmonių, kurių tiesiog asmenybės tipas yra tave pult ir, pavyzdžiui, eina po tavo kiekvieną analizę, klausinėja tų pačių iš esmės dalykų, nors jie puikiai supranta apie ką čia yra ir pan. Tai tiesiog kartais tas toks. Sakau ir kartais net paaiškint, kad pakeitimas geras, net nėra analitiko darbas. Tai yra produkto vadovo darbas, tarkim, bet ateina ir vis tiek mūsų klausinėja, nors mes sprendimo neturime. Mes turime pasakyti, ar tai yra geras sprendimas, ar ne. Tai tiek žinių. Tai va. O šiaip nežinau. Aš galbūt nesu mėgėja labai daug susitikimų, kas pvz. produktų komandoj mes turim labai daug. Galiu dar dabar tiksliai pasakyti, kiek pas mane yra valandų per savaitę. Tai yra kažkur 8-10 valandų per savaitę, priklausomai nuo savaitės planavimo savaitėje. Tai bus apskritai aš kiekvieną dieną turėsiu padirbti laisvas keturias

valandas, pavyzdžiui, kitą savaitę. Tai pusė dienos yra tiesiog susitikimai su visais. Tai va, tai sveika labai nėra. Bet dalis to yra. Tai va.

Informantas Nr. 10

Amžius: 26 m.

Lytis: vyras

Darbo vieta: vaizdų atpažinimo programų algoritmų inžinierius, 4 m. darbo praktikos

Gyvenamoji šalis: Japonija

Pilietybė: Lietuvos

Formatas: nuotolinis pokalbis

Trukmė: 58min 19s

Data: 2024 m. lapkričio 14d.

I.: Esu Simonas Bansevičius, politikos ir medijų magistro studentas. Darau tyrimą apie mašininio mokymosi programų etiką. Informuoju, jog bet kada gali nuspręsti nutraukti pokalbį. Ar sutinki, kad šis pokalbis būtų įrašytas ir jo medžiaga būtų naudojama tyrimo tikslais?

P.: Gerai.

I.: Kiek tu gali prisistatyk, pasakydamas savo amžių? Ką dirbi šiuo metu? Kiek laiko dirbi šioje srityje? Galbūt ankstesnes darbo vietas, jei tai labai aktualu?

P.: Okey. Man 26 metai yra. Aš bendrai paėmus, jeigu full time, tai apie 2 metus arti dirbu šitoje srityje. Jeigu įskaitant ne pilną etatą, tai taip pat dar bus arti tikriausiai, pala, duok paskaičiuoti. Keturi su trupučiu metų, realiai gal 4 metai suėjo. Tai visi mano realiai darbai, kiek esu turėjęs, buvo plius minus susiję su kažkokiu mašininiu mokymusi, pagrinde su giliojo mokymosi. Su tuo klasikiniu ML nesu užsiėmęs profesionaliai ir jo. Šiuo metu gyvenu ir dirbu Japonijoje ir mokausi taip pat doktorantūroje informatikos inžinerijos. Ten irgi yra su jomis susiję.

I.: Aha, gali išsiplėsti apie darbovietę. Kokio tipo tai darbo vieta?

P.: Tai šiuo metu dirbu mažo dydžio įmonėje, kur maždaug iki 50 žmonių kartu sudėjus visus ir HR taip toliau ir esu tenais. Mano pozicija vadinasi algoritmų inžinierius. Tai realiai už ką esu praktiškai atsakingas. Būtent modelių mokymas. Viskas, kas apeinama modelių mokymo, tai reiškias ir duomenų kažkoks apdorojimas tam tinkamas. Pačių duomenų galbūt nežymiu dažniausiai pats, mes kažkam kitam paliekame darbą, bet aš taip pat atsakingas už duomenų žymėjimo instrukcijų kūrimą. Po to sužymėto duomenų patikrinimo taip toliau dar ir galiausiai tie modeliai yra tikrinami ir pateikiami jau klientui kaip

dalis kažkokio programinio paketo. Realiai pas mus tie modeliai nėra Cloud'e, pas mus viskas yra edge device'ai, tai kažkokius, automobilius pavyzdžiui dedamos eismo sistemos, tai tas yra reiškias vairuotojo stebėjimas. Po to yra eismo pačio stebėjimas į išorę nukreipta kamera. Būtent su šituo aš užsiimu pagrinde su eismo stebėjimo, o ne su vairuotojų stebėjimu ir taip pat yra kitokių ten pas mus produktų. Tai šiek tiek kažkoks ten eksperimentas. Moksliniai tyrimai ir eksperimentinė plėtra. Kaip ten? Žodžiu, tas RnD pas mus irgi vyksta, tai kartais irgi šį bei tą gaunu užsiimti su tuo dalyku iš tiesų gana platu. Pagrinde su eismu užsiimam, bet taip pat yra ir kitų dalykų, kur būtų jau ne mašinų, bet žmonių kažkokių atpažinimui, tai šitas ir ta vairuotojo sistema stebėjimo įeina, bet taip pat yra iš kompanijos produktų, pavyzdžiui, privatizavimas dėl kažkokių žmonių, veidų aptikimas ir užfiksavimas ir taip toliau, ir taip toliau. Bet su šituo neužsiimu pats asmeniškai. Tai 2D kompiuterinė rega. Pagrindinis dalykas, kaip galiu apibūdinti.

I.: Gerai. Gali išsiplėsti, ką turi omeny duomenų žymėjimą?

P.: Taip, tai sakykime tas modelis. Kadangi mokyti reikia, tai reikia kažkokių anotacijų, tai gal anotacijos būtų buvę teisingiau pasakyti, jeigu mums, pavyzdžiui, klientas prifilmuoja su mašinos viduje padėta kamera kažkokių eismo tiesiog medžiagos ir iš to reikia tam tikrą modelį. Sakykim, aš jeigu noriu šviesoforo detektorių padaryti, tai reikia tam video paimti, kadrus, sužymėti kur yra šviesoforai. Kuriose vietose gal ten kažkokius taip pat ten klases ar spalvas priskirti jiems ir po to šitą visą naudoti modelių mokymui, taip pat ir ten validavimui ir taip toliau. Tai duomenų žymėjimas apie šitai. Turiu omenyje anotavimus turbūt korektiška sakyti.

I.: O žymėjimo instrukcijos?

P.: Taip, tai kadangi paprastai yra kaip ir baigtinis kiekis kažkokių klasių, pavyzdžiui, detektoriui arba klasifikatoriui. Tai jas reikia apibrėžti žmogui, kuris tas anotacijas daro. Tai pirmas dalykas yra apibūdinimas užduoties pačios, ką mes darome, kokiuose duomenyse ir yra kiekvienos klasės apibūdinimas. Ir pavyzdžiai kažkokie konkretūs. Ir jeigu yra kažkokių, tai. Nevienareikšmiškų ar tai kažkokį galinčių susimaišymo sukelti, tai tokius atvejus irgi detaliai aprašyti toje instrukcijoje. Bendrai paėmus, tikslas yra dokumentą padaryti tokią, kad aš neturėčiau pats asmeniškai kalbėtis su žmogumi, kuris darys anotacijas, bet jis apie mašininio mokymo užduotį, pasirodančius objektus ir duomenis žinotų tiek pat, kiek aš. Tai. Na ir aišku, evoliucionuoja tos instrukcijos, nes būna, kad, pavyzdžiui, kažką nepagalvojau apie kažkokį kraštutinį atvejį. Tada atsiunčia klausimą, ką su tuo daryti? Ir tada pasipila, pasipildo ta instrukcija.

I.: Kaip tikrini modelį?

P.: Taip, realiai yra tiesiog tas tradicinis kompiuterinės įrangos treniravimo, vizualizavimo, testavimo aibės. Tai kiekvienas iš jų yra kažkoks duomenų rinkinio dalis. Būtinai ne persidengiančių.

Tai reiškia ten nėra tų pačių nuotraukų iš tų pačių scenų, taip toliau. Jie visi turi anotacijas, tai ant jų paleidus tą jau galutinį modelį, žiūrima, kokias klaidas jis daro tame duomenų rinkinyje. Kadangi tas modelis nėra matęs ir jis taip pat. Jis idealiu atveju taip pat nebuvo parinktas pagal tai, kas aišku labiau akademijoj svarbu. Tai ir būna tas modelio įvertinimas. Taip pat to pačio validavimo, kažkokio tai testavimo duomenų aibės skaidymas kažkokiais kitais kriterijais. Sakykim, jeigu su šviesoforais, ką užsiimu konkrečiai, tai pagal atstumus, tai reiškia, kadangi pagal jo dydį patį, kadre. Plius minus galima išskirti kažkokias kategorijas atstumą tolimas, artimas, vidutinio atstumo. Taip pat pagal dienos metą. Mes žinome, pavyzdžiui, ar nuotrauka yra daryta dienos metu ar nakties metu, kai apšvietimo sąlygos skiriasi pagal visokius tokius išskirstymus, galima žinoti kaip kiekvienoj tam tikram atvejy ar tai atstume, ar tai laike, ar tai oro sąlygos ir taip toliau. Šitas modelis, kokios jo yra galimybės, kokios yra ribos, kokias klaidas jis daro? Tai šitą, aišku klientas užsako, pasako kokios nori analizės dėl to, kad kuo detalesnė, tuo daugiau kainuoja, nes tada reikia reprezentatyvios kažkokios aibės kiekvienam atvejui. Sakykim, jeigu man pasakytų dabar, kad jie nori ir ten kai lyja ir kai sninga dar kažką, tai aš, pavyzdžiui, turėsiu duomenų rinkinį tiems 5 kažkokias scenas, kur sninga, to tikrai neužteks. Tada darbo padaugėja. Bet kuo daugiau, tuo mes galime realistiškiau įsivaizduoti, kaip realiame gyvenime visa tai veiks.

I.: Ir šitas kategorijas būtent tik tu vienas priskirinėji?.

P.: Kategorijas tai, pavyzdžiui, ten diena, naktis ir t.t.?

I.: Nu jo jo.

P.: Tas kas anotuoja realiai daro. Idealiu atveju turėtų. Na, kai kurie yra labai vienareikšmiški dalykai. Bet yra kai kurie atvejai, kai reikia keliems duoti žmonėms pamatuoti ir tada žiūrėti, ar jie išsiskiria nuomonės.

I.: Bet bet kas, kas apibrėžia, kokios yra tos kategorijos?

P.: Klientas.

P.: Kuris užsako tą modelį. Jis nori, kad būtų įvertinta kaip dieną, kaip naktį. Neseniai norėjo ir dar lietaus, ir ne lietaus būtų, bet diena, naktis ir lietus, ne lietus keturias kategorijas sukomponavo.

I.: Radau užsirašęs iš tavo to, ką sakei, frazę, kurios dabar nesuprantu. Gal tu padėsi man? Idealiu atveju nebuvo parengta pagal tai ir tada.

P.: Taip, taip taip, tai reiškias frazė. Aš kalbėjau apie testavimo, mokymo ir validavimo aibes. Tai sakau, kad taip pat egzistuoja dar viena aibė, ant kurios nebuvo mokyta. Ir taip pat idealiu atveju tas modelis nebuvo parinktas pagal ją. Ką aš noriu pasakyti, kad parinktas. Sakykim, aš nemokau ant jos, bet aš modelį vis tiek testuoju. Ir tada aš gaunu kažkokius skaičius iš to ir sakykime aš ten 5 skirtingus modelius, turiu kandidatus ir aš juos visus palyginu ant vienos aibės kažkokios. Tai, nors jie nebuvo

mokyti pagal tą aibę, bet aš tampu šališkas jos atžvilgiu dėl to, kad aš parinkau modelį pagal skaičius ant tos aibės. Tai, pavyzdžiui, akademikai. Jeigu jie norėtų lyginti, kuris metodas yra geriausias, tai jie negali to daryti, nes jie tada gaunasi, jog gali tiesiog išsirinkti - tai gerai, šitą modelį paimsiu šitam, kitą modelį kitą ir jie gali koks nors metodas pasakyti, kad geriausiai veikia dėl to, kad jie gali pamatyti skaičius ir pasirinkti. Kadangi mes galiausiai turime klientui pateikti šitą dalyką, tiesiog nėra kažkokio daug ten konkuruojančių modelių. Realiai tiesiog yra, kad tas, kuris geriausiai veikiantis ant testavimo ar validavimo aibės, tai bet idealiu atveju to nereikėtų daryti akademiškai. Bet tuo pačiu metu, jeigu taip papuolė, kad ištestavom ir modelis blogai veikia, tai mes aišku neduosime jam klientui to modelio, tai tas vis tiek įtaka vyksta. Akademikai, aišku, irgi taip daro, bet akademikas turėtų tiesiog į straipsnį įdėt tą modelį ir.

I.: O tai kaip tada reikia tą aibę rinkt, kad tu jos neparinktum?

P.: Nieko. Tiesiog realiai turėtų. Ties tuo momentu, kai mes testuojame, tai ir eksperimentas pasibaigti. Aš ne, aš tada net nebegaliu mokyt naujų modelių ir taip toliau, nes tada gaunasi, kad mano sprendimus tolesnei eksperimento eigai įtakoja, daro įtaką mano sprendimams tie skaičiai, kuriuos gavome, kaip tos metrikos ant testavimo aibės. Tai, kas vėlgi nebūtinai daroma visuose akademiniuose tyrimuose. Bet kai yra ant validavimo aibės išrinkti geriausi modeliai kažkokie. Pats paskutinis etapas yra juos palygint ant testavimo ir nieko nebeliest daugiau. Nuo tada tik turėtų būti aprašymas rezultatų ir jokių naujų eksperimentų neturi vykti.

I.: Tada taip pilnai, kad išsiaiškinčiau. Tai, kaip vyksta procesas, kada renki, kurie duomenys ten apmokymui, kurie testavimo būna?

P.: Tai yra daug metodų, skirtingų. Priklauso labai nuo pačių duomenų. Sakykime, jeigu tai yra tiesiog kažkoks padrikas rinkinys nuotraukų, dar kažko, tai galima tiesiog atsitiktinai suskirstyti, kas yra ten mokymo, kas yra valdymo ir taip toliau. Kitas dalykas tiesiog visiškai atsitiktinai. Bet jeigu, pavyzdžiui, yra klasės nebalansuotos, tai, pavyzdžiui, ten yra labai daug vieno objekto, labai mažai kito. Taip skirstant gali netyčia gautis, kad mes, pavyzdžiui, nė vieno mokymo aibei neturime objekto. Kitas būna pasvertas kažkoks atsitiktinis paėmimas. Tai tiesiog atsitiktinai, bet mes atsitiktinai imame iš vienos klasės kažkiek, atsitiktinai iš kitos ir ta pati proporcija yra išlaikoma. Kas pas mus daroma? Mes negalim nei vieno, nei kito daryti dėl to, kad realiai yra kilmė tų nuotraukų iš vaizdo įrašo. Tai sakykime, jeigu tai yra kažkokie paeilėj einantys kadrai, aš paimsiu atsitiktinai ir gausis ten penkių kadrų skirtumu, sakykim vienas į mokymą, kitas į validavimą. Galima sakyti, kad ta scena pakankamai reikšmingai nepasikeitė. Jeigu aš mokiau ant tokio pačio vaizdo po penkių kadrų ir po penkiolikos kadrų, praktiškai aš tą patį vaizdą rodau, nes fonas tas pats, tos pačios apšvietimo sąlygos, ta pati miesto dalis, taip toliau.

Tai čia yra nekorektiška. Tai ką mes darom tiesiog rankiniu būdu. Mes turim kažkokį vaizdo įrašų kiekį. Aš tiesiog praeinu pro juos ir rašau metaduomenis. Tai reiškia ar diena ar naktis tame vaizdo įraše ten kokie objektai pasirodo gal įdomūs. Pavyzdžiui, jeigu ten kokia ypač sunki scena, tai aš galbūt specialiai testavimui įdedu t.t. tai tiesiog rankiniu būdu pasižiūri ir tada plius minus tokį balansą. Pavyzdžiui, šią dieną filmuoti video bus visi mokymai. Kitą dieną visi testavimai ir taip toliau, bet realiai tiesiog rankiniu būdu ir labai tikslingai skirstoma.

I.: Ar buvo kada nors atvejis, kur ilgai visai dvejojai, kur kurį šaltinį, duomenį, vaizdo įrašą skirti?

P.: Ilgai, kad dvejojau tikriausiai nebūtų, nes tiesiog nelabai, kaip čia finansiškai apsimoka ilgai apie tokius dalykus galvoti. Tiesiog, bet kad pagalvot tenka kartais taip. Kodėl? Dėl to, kad, pavyzdžiui, jeigu randu kažkokią retą sceną, tai yra sunkus atvejis. Aš labai noriu ant jo modelį išmokyti, išbandyti. Sakykim, kad ten vienas linksmas atvejis, kokį yra tekę matyti. Šviesoforo detektoriai važiuoja pro pastatą, kuris stiklinis pastatas ir jame idealiai idealiai atsispindi dar vienas šviesoforas. Ir gaunasi jų kaip ir du yra. Tai man įdomu, ar modelis aptiks dar vieną, ar ne aptiks, bet tuo pačiu metu aš norėčiau ir mokyti jį, kad neaptiktų, jeigu aš galėčiau į mokymosi vietą įdėti. Galbūt toks atvejis bus užkirstas kelias jam.

I.: Tai tuo atveju, kiek laiko tam skyrei?

P.: Bus minučių eilės turbūt. Tiesiog aš per nagus gaučiau, jei visą dieną galvočiau apie tokį dalyką, vietoj to, kad ten mokyčiau kažką, realiai pasikalbi su kažkokiu kitu kolega, galbūt ir greitai nusprendžiat tiesiog po to. Bet būna kartais, kad, pavyzdžiui, persikeliame iš vienos aibės į kitą, permetam dar kažką, bet reikia labai aiškiai žinoti, kuris modelis ant ko buvo mokytas.

I.: Jo, tai pradeda naudotojai naudotis jūsų produktu. Kaip jūs susirenkate grįžtamąjį ryšį iš jų?

P.: Galimas daiktas, kad iki manęs tas neateina. Kartais išgirstu kažką tokio, kad o tai jiems labai patiko. Taip toliau. Bet ašgana žemai toje hierarchijoje, kad su manim ten eitų kažkas kalbėtis apie tai. Tai. Kartais tiesiog kažkas buvo susitikime. Koks bosas prieina, pasako, jo ten gerai buvo. Bet tokio grynai grynai tiesioginio ryšio per daug negirdžiu.

I.: O tai čia gal Japonijos specifika yra tokia?

P.: Gali būt. Šiaip labai privatumas yra kažkoks klientų pačių, kas yra klientai, saugojimas. Aš realiai iki šiemet aš jau pusantrų metų beveik dirbu, tai gal prieš keletą mėnesių tiktais pamačiau, kur kokiame produkte tie modeliai kur mokomi eina. Niekas nežinojo iš mūsų komandos iki tada. Tu realiai kažką darai, siunti ir po to tiesiog neaišku kur tas eina. Kaip ir nu žinau kokia kompanija, bet į kokį produktą deda, kaip jis naudojamas, taip toliau. Čia gal bus įdomus į šoną pasiplėtimas su tuo paslaptingumu yra, kad taip pat. Kadangi mes nežinome, kaip jie naudoja mūsų šitą galutinį produktą,

nes jis yra dalis jų didesnio kažkokio programinio paketo, tai kartais atsiranda sunkumų tiesiog prioritetus dėliojant užduotyse. Tai, sakykim, vienas dalykas su šviesoforo detektorių, tai yra, kad. Nustatyta automatiškai ir įvyko kažkokie eismo pažeidimai. Tai tą galima labai daug būdu skirtingų nustatinėti iš tų kažkokių išvesčių tų modelių, visų kartu sudėjus. Bet kadangi mes nežinome, kaip jie tai daro, tai sakykime, pavyzdžiui, vienas modelis, kuris ten geriau veikia kadro viršuje, kitas ten geriau viduryje kažką daro. Tai, kas jiems yra svarbiau, su kuo jie labiau užsiima? Mes negalime net jų pačių paklausti apie tai, nes tiesiog sakys čia yra verslo paslaptis. Bet tada mes negalime jiems duoti pagal jų poreikius, iki galo kai kurių dalykų, o kartais ir kažkokių ezoterinių, gal kartais reikalavimų ateina, kur nesupranti, net kodėl reik daryt. Atrodo, be to būtų logiškiau ir paprasčiau, bet tiesiog prašo tai darai.

I.: Gali kokį pavyzdį pateikti, jei tau leidžiama?

P.: Negaliu.

I.: Gerai dėl to ir pasakiau, kad jei leidžiama, kad netyčia. Nenoriu, kad pasakytum, ko nereik. Aa, kaip jūs darbus planuojate?

P.: Pas mus kadangi gana yra mažos komandos. Tai pavyzdžiui, mano komandoje yra du žmonės, tai aš ir mano lyderis komandos tiesiog. Tai tiesiog ateina naujas užsakymas. Tada ten sulekia, padaugina mano tą tą lyderį, tada tada man pasakoma aš. Aš jau prieš tai mačiau tam bendram chate, bet tada tiesiogiai man pasakoma yra ir jo yra deadline kažkoks, reikėjo tada reikia atlikti. Realiai struktūros kažkokios. Na pats atrandi tą struktūrą padarydamas čia. Nors kompanija nėra nauja baisiai, bet 100% startupų kultūra vyksta vis tiek.

I.: O kiek metų kampanijai?

P.: Dabar pasižiūrėkime. 2005-aisiais įsteigta.

I.: O tai sakei, kad maždaug pats susikuriate savo struktūrą. Kaip atrodytų tavo struktūra?

P.: Jeigu čia yra iš tų šviesoforų uždavinių. Kadangi tenais gana jau išdirbta viskas realiai, jeigu tas pats klientas, kaip dažniausiai, mes turim iš jų daug duomenų ir viską, tai kartais vien tik pasikeitus reikalavimams galima truputėlį tiesiog paimti kitokį pojūtį tų visų duomenų, kuriuos mums iki dabar yra pateikę ir pagal juos apmokyti gana greitai pateikti atgal tą modelį jiems. Tai realiai pirmas dalykas pražiūrėti, kokius duomenis turim. Jeigu kažko trūksta, galima iš jų užsakyti, kad jie ten važinėtų, dar pafilmuoti. Tai realiai čia jokios magijos nėra labai toksai. Kartais pagalvoju, gal negatyviai. Sizifo darbas truputėlį toksai, nes tiesiog ateina kažkoks naujas. Nu dabar ten tą metriką pakeliame, ten pamokai truputėlį negerai atgal duodi kažką kitą, užsinori ir taip toliau. Tai. Gal įdomumo, jeigu kažkokių yra ne nenumatytų klaidų atsiranda. Pavyzdžiui, ten pradeda daug neteisingų detekcija, ant kažkokio objekto atsiradinėt, kurio net niekada nebuvo. Mokymo aibėj ten kokią, pavyzdžiui, naują reklamą pastato mieste

ir ji baisingai atrodo tam modrliui kaip šviesoforas ir tada greitai pamokyti, atgal paduoti ir. O jeigu koks naujesnis uždavinys, tai realiai iš lyderio vėlgi ten kažkokį to do list'ą tokį plius minus laisvos formos gauni. Ir vis tiek ten pradiniai kažkokie spėjimai galbūt yra. Kiek dienų ten kas turi užtrukti dar kažką, bet šitas po to keičiasi gana laisva forma.

I.: Tai ten kažkokio agile'o ar kažko nenaudojat?

P.: Ne, nėra. Nu tipo yra ten weekly meet'ai, bet čia labiau tiesiog, kad žinotų ką darai. Tas pats jau viršesnis, tiek bosas taip toliau. Tiesiog, nes nuo to metiniai po to tie įvertinimai priklauso taip toliau.

Esu gal prieš tai kitame darbe, tokiame pseudo Agile galbūt dirbęs, bet aš iš to atsimenu tik kad buvo tie daily stand up, bet iš kitų ten Agile'o dalykų aš nežinau, ar ten turėjom tuos. Task'us reikėjo visą laiką apsibrėžti ten gal būt kokie ten gantt chat'ai ar tai buvo. Dabar baisiai neprisimenu.

I.: Kada galvoji apie galutinį naudotoją? Kuriose vietose savo darbo proceso?

P.: Jeigu tai RnD, tai realiai nuo pat pradžių visą laiką, nes tas produktas kaip ir neapibrėžtas būtų, tai tiesiog reikia. Ar kaip čia pasakyt, ar aš apie vartotoją?.. Galima sakyti... Taip. Aš gal galvoju, kaip jisai naudos tą dalyką. Bet toks labai ribotas. Tai aš aš galvoju kaip apie tiesiog tą sąsają tarp. Sąsają tai tipo interfeisą, o ne kažkokį ten ryšį asmeninį ar emocinį. Tai tiesiog, kaip bus kažkokios ar tai komandos, ar tai kažkokie ketinimai perduodami iš žmogaus į programą, bet jau į, kažkokia tolesnė įtaka viso šito, negalvoju.

I.: O pavyzdžiui. Ten apie galimus. Nežinau, ar tai darys įtaką eismui, ar ten kokių nors avarijų rizika galėtų atsirasti, ar ne? Tokie dalykai ateina į galvą?

P.: Jo iš to, ką mes kol kas pateikinėjam, tai niekas nėra sistemos darančios sprendimų kažkokiu realiu laiku. Tai yra viskas stebėjimas, stebėjimas ir raportavimas. Pavyzdžiui, vairuotojo stebėjimo sistema, ar tai eismo stebėjimo sistema, jos tiktai stebi ir įrašinėja kažkokius įvykius. Ar tai buvo eismo pažeidimas, dar kažką, ar vairuotojas kažką nelegalaus darė? Sakykim, ten telefonu skambinti pradėjo, ten vidury važiavimo ar dar ten. Tai tas modelis kaip ir nieko blogo negali padaryti, išskyrus tai, kokį false alarm padaryti. Kažkas po to turės atsidaryti tuos išrašus, pažiūrėti, kad iš tiesų nieks nieko nepadarė, bet aš tikiuosi ten jų ten ir nebaudžia. Tiesiog nuo pirmo alarmo kažkokio. Tiesa, yra etinės kažkokios ar moralinės rizikos mažai, tai aš lengva širdimi galiu tuos dalykus vystyti, žinoti, kad niekas neužsiimuš dėl to, kad aš kažkokį ten neteisingai padariau kažką.

I.: O ar susimąstęs esi. Sakykim situaciją kokią nors. Sakykim, tavo modelis. Kaip sakei, galbūt duoda netikrą false positive, kad kažkas nelegaliai, kalbėjo telefonu žmogus, gauna problemų.

P.: Aha, nesu apie tai galvojęs, nes tikiu tiesiog. Kaip čia, nurašau tą kad ne - tikriausiai neįvyks šitaip, nes. Na čia turėtų būt kaip ir padėjimas kažkoks, bet tai neturėtų būti sprendimo priėmimo pagrindas. Tiesiog lengviau atsirinkti, kada tai iš tiesų įvyko, o kada ne, Nes. Nu jo. Kaip ir neteko susimąstyti. Pirmas darbas, kokį turėjau, buvo institutas. Tai kaip jie irgi daug reklamuojasi, tai čia jokių paslapčių neišduosiu. Rinktinė irgi reklamuojasi, kad jie su krašto gynyba labai užsiima, tai tenais galimas daiktas, kad irgi teko prisidės prie kažkokių dalykų, kurie ten galbūt ten kare kažką dalyvauja. Visokius ten dronų detektorius dar kažką daro. Modelių kažkokių ten per daug neprimokau. Aš pagrinde su duomenų užsiiminėdavau. Ar tai analize, ar tai rašyba kažkokia. Bet vis tiek tai šitas irgi gal čia iš to tokio. Kadangi Lietuvos įmonė čia turbūt Lietuvos interesams tą darytų, tai tiesiog kažkokios kaltės nėra.

I.: Bet tu iš esmės vis tiek, kadangi visai slapta kas tas klientas, tai ir nelabai gauni kokių nors rezultatų ten AB testing ar kokie nors ten tokie dalykai neegzistuoja, ar ne?

P.: Ne. Jeigu atsirado bėdų, tada aš apie tai sužinau. Jeigu nieko nebuvo, tai nieks man nieko ir nepasakys. Nu vėlgi, nebent susitikime pasako, kad gerai buvo taip toliau.

I.: O tu gi minėjai, kad RnD darai. Tai, ką šitame savo darbo procese darai? Papasakok.

P.: Tai šitas yra toks gan ezoterinis procesas dėl to, kad tiesiog čia yra. Galbūt kažkas norės ten tokio arba tokio modelio ir viskas. Važiuojam. Tai. Tam yra. Pirmas aišku kadangi mašininis mokymas, tai reikia tam duomenų ir didelio duomenų kiekio, tai reikia ieškoti pradžiai duomenų rinkinio, kuris būtų pakankamai artimas savo kažkuriuo aspektu tai norimai užduočiai, bet tas nebūtinai egzistuos. Vienas dalykas.

Kitas tai būtų literatūros ieškojimas. Tai reiškia kažkokių mokslinių straipsnių skaitymas, apžvalginių straipsnių skaitymas. Kas šiuo metu panašioje srityje daroma? Jeigu tai yra pakankamai naujas RnD dalykas, kuriuo galbūt net nėra per daug užsiimama, tada reikia tiesiog tą problemą labai suabstraktinti, kažkaip mąstyti ne apie konkrečiai problemą, bet apie problemos specifiką. Sakykim, čia bus pavyzdys tiesiog toks teorinis. Jeigu nori padaryti kažkokią vėlgi tą vairuotojų stebėjimo sistemą, kuri atpažintų veiksmus. Bet tai būtų ne kažkokie akivaizdūs veiksmai, bet veiksmai, kuriems reikėtų laikinės informacijos. Tai sakykime. Užsidėti akinius, nusiimti akinius ar ne? Jeigu aš darysiu tai vien tik su nuotraukomis, tai tiesiog aš laikau akinius, bet neaišku, kas su jais daroma, ar aš juos deduosi? Nusiimu taip toliau. Ir tai yra kitas dalykas. Tai yra smulkus veiksmas, galbūt kažkoks. Na, jis nėra labai arti plačiais mostais darydamas dar kažką, tai tiesiog yra sunkiai aptinkamas. Galbūt. Bet sakykim aš tiesiog neturiu. Kol kas duomenų, kurie turėtų. Šiaip iš tiesų esu ten pilna visokių akademinių rinkinių tam. Tai ką aš tada daryčiau, galvočiau ne. Aš noriu ieškoti, kaip akinius nusiimti ir užsidėti, bet aš noriu ieškoti

kažkokį smulkų veiksmą, kuris nėra baisiai išreikštas judesiais. Tai aš susirašiau kažkokį kitą duomenų rinkinį, kur būtų kažkokie smulkūs judesiai, klasifikuojami, atpažįstami ir taip toliau. Ir ką tada daryti? Tiesiog išsirinkti kažkokias architektūras neuroninių tinklų, kurias manom, kad išspręstų tą uždavinį ir tada būti prieš kažkokį savo duomenų rinkinį, gaminimą, kas yra brangu ir daug laiko užima tiesiog ant tų duomenų pasibandyti. Jeigu tas veikia, tipo duoda kažkokius rezultatus, galima tikėtis, kad tam konceptualiai panašiame uždavinyje taip pat veiktų. Tai toksai. Pirminis. Pirminė stadija, o baisiai toli ten su tuo RnD iki pat kažkokio paleidimo produkto, tai nėra tekę nueiti man. Tai tik tiek ir galiu papasakoti realiai.

I.: Kur matytum daugiausiai etikos rizikos tavo darbe?

P.: Čia pagalvokim, kaip čia gerai atsakyti, nes jeigu jeigu pasakysiu negaliu sakyt, tai tada tiesiog visas tas interviu šuniui ant uodegos nueina. Čia apie etiką ir esmė.

I.: Nenueina ant uodegos, jei negali sakyti, tai viskas gerai.

P.: Etinės problemos. Etinių. Iš pačio modelio taikymo etinių problemų nematau. Bet. Čia gal nežinau, ar čia etninė problema. Tuoj pagalvosiu, kaip čia gerai išsireikšti. Čia nėra etikos turbūt. Čia gal labiau tada paaiškinamumo. Čia yra sfera, explainable AI ane. Tai tiesiog, kad. Yra tam tikrų produktų, kuriuose galbūt norima taikyti tuos LLM. Bet aišku, jie yra žymūs savo nepaaiškinamu, kaip jie prieina prie kažkokio rezultato ir kažkokių išsigalvojimų. Tai tiesiog sakai kur aš galėčiau matyti? Jeigu mes pradėtume LLM'us taikyti, tai tiesiog man čia būtų toks. Pirmas dalykas, jog mes kažką duodame, bet aš jausčiaus tarsi nesąmonė kažkokia duodama tiesiog. O kitas dalykas etika. Tada galbūt jo. Nes aš turėčiau dirbti su technologija, kuria netikiu pats ir ją daryti, ir kažkam pateikti. Tai gal čia etikos problema į save patį nukreipta? Bet aš su ta neužsiimu, tai kol kas rami širdis man.

I.: O kodėl netiki šita technologija? LLM?

P.: Hm. Tai pagrinde dėl haliucinacijų. Tiesiog tiek, kiek yra tekę naudoti. Sakykim, jeigu yra ten tų didžiausių kompanijų, tai jie, aišku, viešai prieinami, tiek nusišneka tai. Tai ką mes su savo ribotu nemilijardiniu biudžetu galim padaryti, aišku, yra irgi. O kitas dalykas čia toks labiau madų vaikymasis. Galbūt yra uždaviniai, kuriems tai yra gerai ir reikalinga, bet tiesiog per didelis pasitikėjimas, ypač kompiuterinės regos srityje. Tam dar galbūt visai toli. Nežinau, kiek čia šitas dalykas sustojo. Dabar girdėjau naujienas, reiškias openAI naujasis modelis Orion vadinasi jis išėjo. Ir jis pats didžiausias, pats brangiausias iki dabar visų mokytojų ir padidėjimas arba menkas, arba kartais ir nusileidžia praeitiems buvusiems tam tikrose kažkokiose metrikose. Dar nespėjau per daug įsigilinti. Šiandien tiesiog paklausiau fone kaip ale radijo naujienas kažkokias per Youtube tai taip gaunasi, tarsi atrodo tas iš plokštėjimas priartėja. Kiek galima ten pinigų ir energijos sukišti į tas temas, kad jie kažkokios naudos

pradėtų duoti? Tai yra tikrai naudinga technologija, tikrai stipri technologija tose srityse, kur ji yra naudinga ir stipri. Nu čia tokią tautologiją pasakiau, bet.

I.: Gerai. Kaip apibūdintum, kas yra geras ir kas yra blogas ML?

P.: Geras ar blogas? Kuria prasme?

I.: Tavo supratimu, ta prasme...

P.: Moralės ar kokybės?

I.: Kokybės, kaip jis atrodo gerai sukurta programa? Čia eina ir procesai visi. Tiesiog yra tokia gera geroji praktika.

P.: Viskas gerai yra taip, kaip aš darau. Viskas blogai yra kaip kiti daro. Čia, aišku, juokauju, bet. Aš nežinau, kad yra toks dalykas, kaip teisingai ten gerai viską daryti. Realiai labai daug ką norisi pavyzdžiui, pamačius, sakykim esi prie projekto prisijungei nuo vidurio ir ten labai viską tvarkytis nori, norisi savaip ten kažkaip pertvarkyti, o čia neaiškiai parašyta ar dar kažką. Bet tas laikas yra ribotas, tas biznis į priekį juda ir tiesiog. Kartais gali būti dalykų, kur nesupranti, kodėl padaryta, bet tam yra kažkokia priežastis. Tiesiog kartais traktuoti reikia tą kodą kaip black box kažkokį ir. Tiesiog su tuo judėti tolyn. Bet aš manau, jo, vis tiek idealiu atveju norėčiau, kad būtų programos ir ML šiaip, bet kokios kitokios, nes ML tai realiai tik pradžia produkto yra. Iš tiesų tikrasis, kuris bus taikomas. Tai, kad būtų. Lengva skaityti irgi toks sunkus pasakymas jeigu sudėtinga problema, aišku, nebus ir jos kodas lengvas, bet žmogui, susipažinusiam su ta problema, bet ne su to code base, kad jis galėtų suprasti tą kodą. Galima šitaip pavadint. Tai čia. Bet čia mano tas ribotas suvokimas. Vėlgi aš, kadangi aš taip vadinu fake programmer, nes aš dirbu programuotoju, bet aš nesu baigęs šitos profesijos studijų, tai aš kadangi iš fizikos turiu pradmenis tipo, kad mano magistras buvo teorinė fizika ir astrofizika. Tai. Ten visai kitokia paradigma kodo rašymo. Kodas nėra esmė, esmė yra kodo išvestis. Jeigu kažkokio kito akademiko kodą reikėtų kada nors skaityti, tai niekam nelinkėčiau. Labai negeras dalykas yra. Ir ML akademikai taip pat į GitHub kelią. Tai reiškia, tai yra programinis kodas, kurį galbūt ir verslai naudoja, bet jį rašė akademikai. Tai labai jaučiasi ir tai labai kelia neigiamas emocijas. Tačiau. Taip pat ir būna. Na, industrijos techniškai yra kažkokios ane. Žmonės, kurie rašė, kurie mokėsi ar, pavyzdžiui software engineering ir t.t. tai vis tiek yra studija kažkokias sufokusuota į kodą kaip produktą, tačiau taip pat kodą, kaip kažkokį projektą, kuris vykdomas kartu su komanda ir taip toliau. Tai aišku tų žmonių, kurie gavo formalų mokymą kodo rašyme. Iš jų galima galbūt geresnių lūkesčių turėti, bet aišku, tas irgi ne visą laiką taip yra, nes tiesiog ypač kažkokiems startup ar dar kažką, kai reikia greitai, greitai, greitai judėti viską Tiesiog viskas taip sumetame ir vėliau sutvarkysime. Tas vėliau niekada neateina, nes vaikšto žmogus, išeina, perleidžia kažkam kitam tą savo kodą ir. Tada jis black box virsta.

I.: Galvoju ar tavo atveju šitas klausimas išvis tinka, bet tebūnie. Ar manai, kad koreliacija vis dar nelygu priežastingumui?

P.: Taip. Galima duot klausimą, kodėl manai, kad mano nuomonė turėjo į tai pasikeisti pastaruoju metu?

I.: Tai va va, sakau, kad ne visai tavo domenas į tą reikalą. Čia būtų galbūt labiau, jei dirbtum su rekomenderiais ar kažkuom. Jo, tai esmė, kad atsiranda vis daugiau žmonių, kurie sako, kad jeigu turim baisiai baisiai daug duomenų, tai mums kartais net nebūtina suprasti, kodėl tie duomenys koreliuoja. Tiesiog panaudokite tą koreliaciją kaip įrankį kažkokiam rezultatui gauti. Net nebūtina galvoti apie priežastį.

P.: A tai, gerai, ne, aš nematau su tuo problemos kaip čia. Tai nebūtinai, kad bus priežastingumo ryšys dėl to, bet jeigu mes tai tiesiog traktuosim taip ir mums gaunasi rezultatai, tai kodėl gi ne? Jeigu tai veikia, tai veikia. Taip, mano toksai. Jeigu tai daro pinigus, tai daro pinigus.

Bet aš įtariu, kad yra kaip čia. Jeigu ar tai žinodami ar nežinodami dirbame su kažkokia neteisinga prielaida. Tai čia yra sėkmės reikalas. Tipo, ar mums amžinai seksis, kad ta iliuzija laikytųsi? Ar nesiseks amžinai, kad ta iliuzija laikytųsi? Bet neturiu iš gyvenimo pavyzdžių jokių čia nei čia patarimai ar dar kažkas. Tiesiog.

I.: Kas manai, tavo kuriamų produktų atveju daro didžiausią įtaką galutiniam rezultatui? Koks faktorius?

P.: Šiuo atveju tiesiog duomenų kiekis, kiekis ir įvairovė.

I.: O jeigu galvojant koks asmuo pagal rolę nuo naudotojo iki kito tavęs?

P.: Didžiausią įtaką... Mano atveju užsakovas didžiausią. Nes jeigu jo nėra, aš negausiu to projekto daryti. Na, čia toks kvailas atsakymas, bet.

I.: Bet tai sakykim, vat gavai projektą. Tada projekto programos rezultatai?

P.: Paprastos... Didžiausią programos rezultatui? Tipo, ar gerai gavos ar negerai gavos?

I.: Ne tiek, kad gerai ar negerai, bet tiesiog. Nu va tą parekomendavo man, ką pasiūlė, nustatė. Ir žiūrint visą darbų eigą, kiek ko, kokių veiksmų buvo atlikta, kokio asmens indėlis daugiausiai padarė?

P.: Aaa, ok ok. Tai reiškias, sakykim produktą vystant. Kas turėjo didžiausią įtaką to produkto galutiniam? Kaip ir sakykim, aš dirbau prie kažkokio projekto nuo pačios pradžios iki pateikimo dalies jau iki pateikimo. Kas turėjo didžiausią įtaką šitam? Turėjo įtaką. Kam konkrečiai šitam? Nu tiesiog bendrai? Viskam?

I.: Nu vat ką tik mašina ten nustatė, kad va čia rodo. Nežinau ką ten nustatot su šviesoforais.

P.: Aa kas atsakingas?

I.: Galima ir taip.

P.: Realiam gyvenime kažkas nutinka, kas atsakingas už kažkokį įvykį ar to įvykio nebuvimą, dar kažką?

I.: Na, tebūnie apie kas atsakingas.

P.: Atsakingas yra tas, kas paleido produktą į rinką. Ne aš negaliu taip vienareikšmiškai atsakyti. Visi turi atsakomybės. Tipo aš norėčiau. Pradžioje mano darbas buvo užtikrinti, kad tas modelis būtų geras. Tada yra jų darbas. Kai jie gauna tą modelį iš manęs pažiūrėti, ar jis tikrai geras, ar aš ten jiems neprisvaigau nesąmonių. Jie ten turi irgi savo būdus tikrinti. Jie turi savo duomenų rinkinius ir taip toliau. Tada jie. Galiausiai pateikia kažkam jau, kas naudoja tą jų galutinį programinį paketą, nes čia patys mes esame outsource'inami. Čia tas kaip ir grandinė tokia, kad tikrasis užsakymas dar toli toli yra, mes dalį kažkokią darome, nes mus užsakė, kad darytume tai. O tas, kuris jo. Visi, kas prisidėjo prie programinės įrangos. Pvz. kaltinti to, kuris nusipirko šitą produktą, negaliu, nes tikriausiai jis viską ką tik girdėjo, kad čia labai gerai. Labai liuks tipo pirkit, tik naudokimės. Čia viską ištestavo. Tai tas, kuris perka. Jam ir neturėtų kristi atsakomybė, kad jis ten dar, pavyzdžiui, ten pusę metų pats rinktų duomenis, ten kažkokius žiūrėtų. Tiesiog ne tam yra, tegu jis pats tegul savo sprendimą susidaro. Jeigu turi tiek pinigų ten testavimas visokiens ir toliau. Tai atsakingas. Labiausiai tas, kuris paskutinis pateikė produktą, nes jis iš visų ten skirtingų surenka kažkokias funkcijas, jas sudeda į vieną ir. Jie jau turėtų tada užtikrinti, nes jie pateikia tą produktą. Jau jie yra paskutinė riba tarp realaus gyvenimo ir to modelio. Tik ten būna laboratorijoj kažkokioj.

I.: Kas darbe tau kelia stresą?

P.: Būtų kvaila pasakyti nieko, nes tikriausiai čia netiesa yra. Aš pakankamai gerai tą stresą užspaudžiu, kad ten baisiausiai kažko nejausčiau tokio. Tiesiog pvz. jei. Čia gal kažkoks fatalizmas dar kažkas, nežinant ta prasme, kad sakykim kažkas. Ok. Būna tikrai, kai RnD stresą kelia. Dėl to, kad labai vėlgi, kaip sakei ezoterinės užduotis. Tai reiškias sakykim aš mėnesį praleidau ten kažkokį bandau rezultatą gauti, nieko nesigauna ir tau pačiam gale tada po mėnesio pasikeitė reikalavimai, dar kažką, o visa kita. Tai čia toksai gaunasi. Praleidau mėnesį, tas mėnesis kaip ir nieko neturiu ką parodyt už jį. O ten tie metiniai įvertinimai, taip toliau ten nuo to ten alga gal ten pakyla, taip toliau. Tokie visokie dalykai tai šitas, sakyčiau, streso šaltinis, nes kitais atvejais, sakykim. Kažkas, jeigu nutinka tiesiog iš viršaus. Pavyzdžiui, ten klientai nelaiku duomenis atsiuntė, vėluoja dar kažką. Tai čia ne mano kaltė. Aš negaliu nieko padaryti dėl to. Tiesiog taip. Sakykime, užduoties neatliko laiku, bet aš neturėjau galimybės ją

atlikti laiku, su realistiška kažkokiais lūkesčiais. Tai aš manau, daug kas galėtų dėl šito irgi stresuoti, nors visiškai ne jų kontrolė. Bet tas RnD yra mano kontrolėje, nes tiesiog iki tam tikros ribos.

I.: Ką darai, jeigu artėja deadline'as ir modelis dar neveikia pakankamai gerai, kaip tikėjaui, kad veiks?

P.: Ką darau, tiesiog iki paskutinės dienos, kaip čia? Na, pirmas dalykas, aišku, aš turėčiau būti iki tada jau pasikalbėjęs su kažkuo apie šitą dalyką, padarysiu prielaid1, kad nėra, kad aš tyliu, tiesiog bandau, kad niekas nesužinotų, kad tas modelis neveikia ir arti deadline'o. Tai jeigu kažkokį protingą patarimą gavau, tai bandysiu tą. Jeigu ne, tai vėlgi. Čia trial and error dalykas. Tiesiog tą modelį ten pakeiti vieną hyper parametrą, kitą, kol galiausiai kažkas gaunasi tiesiog kas kas yra pagal reikalavimus. Aišku, priklausomai dar gali būt kažkokią ten durną klaida ten konfigūracijoje padariau, dar kažką tai, ten nuodugniai, tiesiog viską ten tikrinsiu iš eilės, iš naujo, iš naujo. Dar. Tai viršvalandžių pasėdėt tikriausiai reikės tada, nes tiesiog ten ilgesnės dienos gausis. Viskas tvarkoj, nes paprastai nebūna viršvalandžių pas mane. Bet būtent prieš deadline kartais. Jaučiu, nieko protingo per daug nedarau, tiesiog brute force realiai.

I.: O buvo kada toks, kad brute force'inai, bet vistiek, nesuspėji pasiekt reikalavimus?

P.: Su projektais kaip klientams ne nebuvo, nes tiesiog pasisekė. Kartais būna paskutinę dieną dar ten kažką, bet su RnD yra buvę, kad kažką developinu? Ir ne tai, kad deadline klientui kažkoks pateiktas, bet tiesiog kompanijos vidury kažkokie. Užsibrėžėme, kad mes tiek laiko skirsime tokiam tyrimui. Jeigu matome, kad nesigauna, tiesiog kažką kitą mėginam ar šitą visai į stalčių įsikišam. Tai tokių yra dalykų buvę, na tipo nėra malonu. Bet tiesiog taip buvo.

I.: O šiaip kalbant apie šitą vertinimą. Gana įdomus atvejis, kadangi tu nežinai galutinio naudotojo. Bet turit kažkokius treshold'us, kiek ten padidėjimą, patikslėjimą reikia pasiekti?

P.: Jo, tai šitas yra būtent užsakymo dalis, kad tiesiog jie užsibrėžia skaičius kartais ir tokius skaičius su realybe susipykusius. Tiesiog mes norime tiek tikslumo ten, kažkokio precision'o, tiek recall'o ir taip toliau. Ir važiuojame. Tipo darome. Tai šitas pagrindinis yra užsakymo reikalavimai, kad būna būtent šitoks dalykas, kad, pavyzdžiui, kažkokiai klasei nepatinka ten tikslumas. Nori padidinti arba pavyzdžiui, kad ten atstumą kažkurį nori, pavyzdžiui, praplėsti. Atstumą ten kuriuo veikia tas, taip toliau, taip toliau. Tai kartais būna neįmanoma tokių užsakymų, bet. Tada tiesiog diskusiją su jais kažkas padaro.

I.: Bet šitų reikalavimų tu pats neapibrėžinėji?

P.: Ne čia ne mano reikalavimai.

Informantas Nr. 11

Amžius: 31 m.

Lytis: moteris

Darbo vieta: vyriausioji duomenų mokslininkė ir inžinierė, kuria ML pinigų mokėjimų platformai. 6 m. patirties.

Gyvenamoji šalis: Jungtinė Karalystė

Pilietybė: Turkijos

Formatas: nuotolinis pokalbis

Trukmė: 53min 2s

Data: 2024 m. gruodžio 13d.

I.: My name is Simonas Bansevičius, I am a master‘s student, studying „Politics and media“, this interview wll be used for my research about Machine learning ethics. I inform you that at any time you can decided to end this inverview. Knowing this, do you agree with me recording this interview and using it for the previously mentioned research?

P.: Yes.

I.: I'd like to you to introduce yourself. As much as you want. If you can state your age, also where you work at the moment. What's the company? The sphere. How long are you in this discipline in general? Maybe share about your team ,its size, structure.

P.: Okay. I have a background in industrial engineering which is a good background in Turkey, but it doesn't exist in UK very much. And it was heavily focused on statistics and optimization. And I moved to UK, and I wanted to transition into data science machine learning, as I have this engineering background, and I did several bootcamps and freelance projects with various companies and later I started to my kind of like the first big job. Like data science job with (anonimizuota). And it's a data science AI consultancy company where I had the chance to work on several companies from a variety of industries and variety of the data science topics such as social network analysis, dashboarding, predictive modeling, neural networks, lots of lots of things. And it was quite kind of like generalist role where you are data scientist, but also a little bit data engineer and a little bit project manager. Since I joined there early stage early stage when they were early stage, I had the chance to work with everyone in the company, flat hierarchy, etc. and then later I wanted to specialise in an industry and in a data science topic, and I moved on to my current role, which is a senior data scientist at (anonimizuota). I work for Financial Crime Team, who does the financial crime solutions for the UK banks. And I work with all all payments that are going through payment network. So it's account to account payment transaction data. We don't have

any sorts of PII information or are in the information related to accounts or anything is basically like a bunch of network and a bunch of transactions only. I still work on predictive modeling, machine learning models, mostly things like XGBoost and decision trees. Random forests. Yeah. That's it. My team size is about like ten people maybe, and I am. I'm 31 years old. So it took me it took me like six years to. Yeah. Like since I transitioned into into data science, I've been working on the industry for about seven years since. Since 2018.

I.: Okay. Could you share. Most common or maybe biggest challenges that you have faced in your work?

P.: Do you ask for technical challenges or career wise challenges?

I.: More like technical or team related. Yeah. I don't know you can. You can mention all if it's going way too, too, too much out of scope. I will tell you.

P.: Okay. The challenges usually if it's a data science project, I can't remember any data science project where we didn't have like getting the input data challenges. It always takes some time to get the data from client or transfer data to a certain environment that is safe. Best is that the GDPR, of course, is a big, big thing. We need to be super careful. Sometimes it's a UK data and I cannot work from anywhere in anywhere else in the world because I cannot even see the screen that if my colleague is sharing some data. So I would say biggest challenge is how companies are managing the data. And still, still we don't have so much, I mean, so good, like data practices in place that if we wanted to start a data science project, oh, here is the data and start from day one. I think we are still struggling with it. That's that's kind of a challenge. Another challenge is that kind of more like, career wise challenge. A data science is not a primary thing to do for a company. It's more like secondary thing to do. So a company where you you need to have a backend DevOps and front end, maybe if it's a B2C company. And then of course like business development people, but the data science comes like quite later. That's why I think there is only a handful of companies that does proper data science, if that makes sense. So it's really hard to find like a really good place that understands and does meaningful data science. Yeah. And I think it's also creates, creates a quite ambiguous roadmap for a road for any like, data science professional in the industry about where to go and what to do about their career, basically. Yeah. And then and then I'm not even mentioning that data science is a wide, wide topic. There is like product data science. There's like fraud data science, machine learning engineers. There is, there is there is so many data science types. Yeah. Positions, roles that are so much different than each other. Yeah.

I.: Okay. You mentioned that you need to have your data safe. Could you elaborate? Explain more about what do you mean by safe?

P.: For example, if I work with a client, any client, not just in finance or anything, just any client, I want to build a model, let's say. And then their data has some, some information, let's say, like some people names, addresses, account names. Let's say I'm working with a telecoms company and then I want to figure out some. I want to build a model for them to predict maybe customer behavior or whatever. And in their dataset always have this account information that is super sensitive. They need to either change that data or take out data that, that data or somehow hash it use some cryptography to change the values and then send it to us, etc. and then we do the model we are predicting for some number for some certain customer. And we are like sending that number to the client. And they need to revert that ID to understand which customer is that on their side. Yeah. The really things things like that, like sensitive information is and also, for example, I work at (anonimizuota) and we, we cannot even have like some technologies like GCP, BigQuery, or Amazon Redshift like SQL basic SQL databases because because we cannot share payment data with any, any sort of cloud provider, right. So yeah, really things like that.

I.: So one of the solutions you've mentioned is hashing doing cryptography. What are the others when you need some account info but you cannot use it.

P.: If we if we need some sort of account information and cannot get it sometimes, sometimes we, we build a POC model with a super mock data that is super, super synthetic data set. And then we might use that to convince the client to go for a bigger like, like a bigger kind of POC or project later on. So we do some sort of initial phase that we mock the data and we mock everything, and we show some results that we are capable to do this. And then later on we create that environment. Actually, that's okay to do data transfer under a contract or something. And then we start to work on it.

I.: Okay. Is there some sort of data that you not only have to keep private, but simply just cannot use.

P.: Ooh like, oh, I don't know. Probably. Probably we can, we can I mean, in theory, you can build models on anything, right? Any demographics like gender, age, etc. but there is not one model in the past that I built that using this kind of demographic information. Even if I had access, I would not use them, if that makes sense. Probably demographics, I would say.

I.: But but you said something about using it internally, or did I hear it wrong?

P.: No. No. Internally, it's more like, for example, a customer has an account in a bank, let's say when when that customer opened that account. What kind of account? Events that they performed, like changing address on the account or, you know, this kind of information, more like technical information that is out there. We can use that. We can utilize that kind of information that is performed by someone like more like behavioral. But certainly we wouldn't use like any sort of demographics information about

kind of that person, that specific person. And in the past that was there was a model That that had a, that had a feature that that had a certain type behavior that only performed by women. And it gives the model about like a hint of or understanding of the gender of that model. And I personally, I, I wouldn't, I wouldn't use that kind of feature that introduce any sort of like gender or something. I don't know, that's my personal preference. That's not that. That's nothing to do with data science. That's more like how I envision the models that are. I don't know.

I.: But you said you personally you would not use it. Does that mean that in general, you could have used that behavior that certain women did something?

P.: Yeah, I think there is. There is there is no rule or as as far as I know, there is nothing that could avoid us to use that.

P.: Especially let's say you have an information. Let's say. Let's say you have an information the age of the person, let's say, and then you create another feature, something related to something using that information. And you create a feature in the model like to be input to the model. Does that make sense. You do the data crunching, data wrangling and use that information to create some other information. Let's say I know that you are 30 years old and then I create in my model. I have a feature that that has the probability of the fraud of a man who's 30 years old. So it it is nothing to do with your age, but it is kind of like hints the model, whether you are 30 years old or not. If I calculate that kind of feature, then it's my it's my information and I can use it in my model. After the data wrangling etc.. So there is nothing like if you want to use any sort of demographics, probably we could do it in a way.

I.: So in a way you cannot use demographics per se, but you can make some proxy? Information that you can use as a proxy to that demographic, which you cannot use?

P.: Yes, exactly. Exactly, exactly. And that information, if I understood correctly, that belongs to my company. Like any company, if I created that information like secondary or proxy information and as company, probably I don't need to share it.

I.: Share with whom?

P.: Whom? Like. Like the. With the clients, for example.

I.: Is this a practice that is done using this proxy information or not?

P.: No, I think no. And as a data scientist, I wouldn't do that. But I know that it's not kind of out of reach. It's kind of it could be done. Or like I use that only one one feature kind of proxy feature that is derived from the gender. But but that is like out of the use now. So that is. Like. In the past. By the way I am talking this kind of experiences not about like one certain company but more like my like experience in the last, last years in general.

I.: When do you think about the end user of the program/system?

P.: Like like in terms of like, can you rephrase a little bit more?

I.: Yeah. In which, okay, during the development process, your everyday work. At which moments do you think about the end user?

P.: Since I always worked B2B, I work with the clients kind of the model's end user is kind of always the client. How they receive the predictions out of my model. But the models are. They can be based on behaviors of the people or the customers of that specific client. And if I build a model for that client's customers, that is how I analyze the data. And yeah, I mean, but not like think of the end user. I don't know. I don't know if it's counted as a thinking of the end user. But but it might be my experience because I never work B to C and did not directly give any sort of predictions or results that might affect a customer like directly. My like models effect might be more like in indirect.

I.: Look at your everyday work at the development processes that you do. And where in these processes do you see possibilities for mistakes or something that may decrease the quality of the program.

P.: For the, for the that's a that's a hard question. Anything might, might go wrong, actually. So if, if we talk about the models specifically, there is there has been many times that I've kind of seen that you build a model with the training data set and test, test it with the extension of that training set. And then in every kind of isolated environment that you created carefully, and then it, the model goes live and then you see a drop in the performance. So it's it's usually kind of not a surprising thing that that happens, that you build something and put it on live production and that doesn't perform as expected. So that's one thing I think that definitely can go wrong or like kind of like, yeah, it might not be as expected. And I don't know, another thing is that kind of still related to this one is the Overfitting. And you fit a model, but it doesn't fit really well. And you learn the patterns in your training set too much and too well. It doesn't perform the same as the as in the production environment at the prediction time. That's also like very similar scenario. And the quality and if if it is regarding the quality of the technical tools that we improve as data scientists, I can also say that usually data scientists are less experienced programmers and usually the quality of code. I mean, not usually, but sometimes the quality of code that we produce as data scientists is not as good.

P.: Yeah. So the, the, like adding test cases, test unit tests, whatever test would be really helpful to not to make any major mistakes with the data or the assumptions that you made while building your data engineering pipeline and the data model.

I.: Please tell me about the processes that you do to evaluate your model. Like, how do you define if it's good or bad?

P.: In the development process first of all, you use some sort of cross-validation cross-validation methodology. It could be rolling cross-validation. Time based rolling cross-validation. By using different parts of the data set. You fit model and then test it on different parts of the data set. That's one thing. And then if, if kind of if it's more like time based data that you have like a certain period, you, you try to you try to use some data out of the test time. That's called out of time testing. So, for example, I fit my model using 2023 data set. So I use like some, some some data from like very far future like 2024 to test how it performs in the in the feature. And then out of time testing cross-validation. You could perform some robustness tests depending on the model. Actually these these are depending on the model. But robustness tests are definitely useful. That is where you try to break your model. For example, what happens if I had like money related features? What happens if there is an inflation, let's say 10% inflation? And what happens you know, like some, some certain, some certain real life use cases that might happen, and then you test the model against them by changing the data set a little bit every time, and then see if there is any drop in the model's results. And also recently I tried like adversarial robustness tests. That was fun as well. That's where you where you try to trick the model in certain ways that that's super fun.

I.: Okay. And speaking of in terms of evaluation of when the model is working as you intended or when is it. How do you define when it has reached the required quality?

P.: Yeah, I guess, I guess the the metrics that you created like a like the goal during the goal, like, goal setting or the initial planning of the of the the project. Sometimes there is, like, contractual agreements, like model has to be at least like 25% accuracy and things like that. So when you reach that kind of KPIs that are planned before or if not if there are some reasons for it, then yeah, I would say it's it's okay to it's okay to release that model. And sometimes you can take measures like let's say I have a model predicting some certain label. Let's say I have I'm predicting, let's say like I'm predicting crypto coins and they are which industries they belong. Are they meme coins or are they NFTs and things like that? Let's say I'm labeling them. My model is labeling them. But for some of them, some some of them, I have low prediction power. I might choose that when there is a low prediction power, I am not labeling them overriding the results or I can choose that if the predictions are too high. My model is so confident I am only labeling them and not labeling any other thing. So you can. So on top of building a model, there is a part that how you use that prediction. What kind of threshold thresholds do you set.

I.: Could you could you explain a bit more. The last part when you said it's not very confident and when very confident.

P.: So for example I am predicting whether, whether this coin, this crypto coin is a meme coin or not. My model says that with 10% or like 50% prediction like like probability. This coin belongs to a

meme coin industry or meme coin label. And then I say that 50% is almost like flipping a coin. And it's I don't take your answer seriously. Tell me only tell tell the results or the labels. Only when there is at least like 90% chance. Like only tell me when you are super confident about something. So there is a phase that you can like. You can shape how you use the predictions.

I.: Okay. Yeah. Okay. You've mentioned metrics. Could you explain more how the metrics that you use look like?

P.: I think. It is very it is kind of very out of the textbook. If it's a regression based model, you use accuracy metrics mean squared error, mean absolute error. And if it's a classification model you use coefficient confusion metrics. The precision recall. Yeah that that kind of metrics. And it again it depends on the problem where one metric might be primary goal. Depending on the problem I might not care about recall but I might care about the precision more depending on depending on what I am trying to do. So I have the ability to optimize the model by primarily choosing one metric.

I.: And how do you define which one to use? Which one to throw away?

P.: For example, again the same example I am labeling a coin's industry. It's in that case, it's. For me, it's important that I am not providing any false positives. False positive is that I say a coin belongs to meme coin, but it's not. I don't want to do that. So that's why I can choose Precision over recall for example. So it it depends whether you care about false positives or false negatives.

I.: Do you have. Any metrics that are not purely mathematical but more related with the social goal, not social like the business goal.

P.: I don't know. I'm not sure.

I.: Yeah. I mean, it's okay if you don't.

P.: My world is only only mathematical at this point.

I.: Okay. There's one more interesting question. Which is. Does correlation still not equal causation?

P.: Yeah. Yes. Well I guess yeah. I don't know. It might be super theoretical.

I.: No doubt about that?

P.: No, no.

I.: Okay. Do you have any vague idea why I may be asking this?

P.: No. Is it. Did you find something that. I don't know, correlating in in your research?

I.: I mean, there are some people who now say, in general, I got this idea from reading one article that was written more than ten years ago by one businessman who said that we have so much data now

that we can simply look for correlations, and we don't really care whether they cause each other. We can just use that.

P.: Interesting.

I.: So, so so you it doesn't ring a bell, for your practice? Do you think or try to make sure that the data that you predict is based on causation?

P.: Well, not not really. Like. I mean, of course, but but the data that we have kind of always limited, on the contrary of what people might think. And sometimes the data that companies store and designed to store is already in a very bad shape. They they may They might be stored at. Like. Like values that are. That doesn't make sense. That make any sense? We as data scientists, I think we are already limited. Like what we can use. Yeah. So. So still, like, most of the time. Yeah. We are trying to use what makes sense in terms of causality, I think. And Yeah.

I.: Is there some process for you to define whether it is based on causality?

P.: Not really. The only the only the only place where we actually, I mean, where I actually use the correlation and causality might be that if you have information that are highly, highly correlated in your model, you use one of them, you remove the other one. So if you have for, for example, it might be a stupid example, but let's say you have a gender information and that's one of your features in the model. And another feature that tells you if it if they are a mother or not, then maybe you can think about your data set because only women can be the mothers, etc.. So like if there is like two features of the same source or like, like highly correlated like 0.9 or something, then you might choose that, I'm going to remove this one and then use only the one that that. Yeah. That captures the information already. So there is, there is like a kind of like a practice that where we remove if there is like multiple feature features that are highly correlated and that doesn't bring any more value to the model that the other feature does.

I.: Where would you say that there's the high risk for ethical problems and everyday development processes that you face? Like the possibility to make something unethical. Not that you do.

P.: Yeah. As I, as I said, if I would be building a model for I mean that's just an example. But if I somehow again like the again with the gender, unfortunately, that's my only example. But if I would somehow building a model that is hinting whether this person or this information is related to a person and they might have a certain gender. I would be really uncomfortable with that idea somehow. Like predicting that person's like like a, that something they do in their life and somehow disrupting it with, with high probability of fraud or something. That would be really uncomfortable for me. Other than that, the ethical issue. I mean, and also one ethical issue, maybe somehow that if I work with payments data or transactional data, somehow I work with the data and I use Jupyter notebooks, for example. You print

out the results and then you. Sometimes you push Jupyter notebook to a Bitbucket or like git repo somewhere. And by mistake, there is a chance that you expose someone's transaction to a developer that who had no access to that data. So that's, that's a big, big, big problem. So like usage of Jupyter notebooks in general. So if you have super sensitive data, you might opt out using Jupyter notebooks at all. And you can just use Python scripts and don't show any sort of, any sort of data. So yeah, overall, just to just to summarize, like having any any sort of features in the demographics in the model and plus like exposing the data data to, to other people would be uncomfortable things to do.

I.: Do you yourself have to do any fixes, manipulations or other modifications to the data? And if you do, what kind of.

P.: Yes. So. So creating some sort of summarizing statistics and features out of the information that we have. Any sort of data wrangling. Right. Using that data, creating more meaningful information that is, that is kind of yeah. Any yeah, that might be category of what you what you ask?

I.: But say you have you have your raw data and you see that there's like one column sometimes is missing. What do you do then?

P.: Oh. It depends if. If it's a super sensitive thing. If it's a super sensitive model, then you probably want to remove any any data with missing values instead of instead of filling out with something else, basically. Or if it's if it's more like a secondary information that you created out of other data set and that how can I say that then in that case, if it's a secondary information that you already created from other other other data set and you. You will assign some value to a data data row or data point. Then you can use like mode or median of that data set.

There is also some like I can't give more like comprehensive examples, maybe because some of them is coming from directly my like work experience and, you know, like for example, like there, like there are things that I can expose, I cannot expose, etc.. So my examples might be too abstract sometimes about that. Yeah.

I.: Yeah. I mean, I don't want you to break your non-disclosure agreement. Okay. Maybe the last question how do you yourself feel when you use products that are based on machine learning, AI, etc.?

P.: that's hard to tell. Hard to tell. I mean, first of all, I am just thinking the other side of the whatever I am using the other side of the other side of that is a stupid machine, basically and very probabilistic and might make a lot of mistakes. But there are places that I also really believe. Yeah, believe in in the, the power of the data. And for example, if you see any statistics, any statistics on the paper, like, oh, this, this, this model achieved like 60% or something, I am always reading like I am sometimes reading it

like someone believed and wanted that model to achieve like 60% accuracy because in this in the data you can bend a lot of things. And data science almost is like an art or like a, like a really sensitive science. And I'm, I'm, I work I work like, like years and I still like have a lot of areas that I am not super confident and I try to be careful basically. So yeah, like statistics, there is a lot of books about that. I am not the only one saying like there is like bad data. Like anything the government does or gives any statistics might be wrong or might be different than what is what is shown. So you need to you need to have a full picture. And that is that is hard.

Informantas Nr. 12

Amžius: 25 m.

Lytis: vyras

Darbo vieta: skaitmeninė rūbų pardavimo platforma, eksperimentavimo platformos inžinierius, ML inžinierius. 7m. patirties.

Gyvenamoji šalis: Nyderlandai

Pilietybė: Lietuvos

Formatas: nuotolinis pokalbis

Trukmė: 51min 57s

Data: 2024 m. lapkričio 6 d.

I.: Esu Simonas Bansevičius, politikos ir medijų magistro studentas. Darau tyrimą apie mašininio mokymosi programų etiką. Informuoju, jog bet kada gali nuspręsti nutraukti pokalbį. Ar sutinki, kad šis pokalbis būtų įrašytas ir jo medžiaga būtų naudojama tyrimo tikslais?

P.: Taip.

I.: Jeigu gali prisistatyk pasakydamas savo amžių. Ką dirbi šiuo metu? Kiek laiko dirbai šioje srityje.

P.: Man yra 25. Šiuo metu dirbu kaip inžinierius. Pagrinde darbuojuosi ties eksperimentavimo platformos kūrimu, bet didžiąją darbo dalį įmonėje ir prieš tai dirbau ties mašininio mokymosi algoritmais, jeigu taip lietuviškai išsireikšti. Šitoje srityje esu maždaug septynis metus. Dar mokausi magistrą dirbtinio intelekto srityje.

I.: Gali daugiau išsiplėsti apie savo praktiką, savo patirtį, apie savo komandos struktūrą, kaip atrodė.

P.: Didžiąją laiko dalį, kai dirbau ties šitais visais algoritmais, tai praleidau ties moderavimo ir kontroliavimo sritimis. Konkrečiai šnekant apie. Prekių katalogo moderavimo, bandymu atpažinti padirbinius, netinkamus vaizdus ir panašius dalykus, taip pat bandant aptikti bloguosius aktorius, taip vadinamuosius. Tai žmones, kurie nori ištraukti informaciją iš klientų. Tai spamas, phishing ir visos kitos atakos. Tai darbas iš esmės buvo tiek su tekstu, kad bandyti atpažinti spam tipo žinutes susirašinėjimuose. Su vaizdu, bandant atpažinti netinkamus vaizdus, ar tai būtų kažkas, kas yra netinkamo, nuogybės ir panašūs dalykai, ar tai būtų netinkamos prekės, kaip. Tai, ko mes neleidžiame pardavinėti. Ginklai, narkotikai ir kiti dalykai, ar bent bandymas atpažinti prekių padirbinius iš tų pačių nuotraukų ar tam tikrų papildomų atributų. Darbas buvo dažniausiai produktinėse komandose. Vadinasi, buvau inžinierius, įterptas į produktines komandas, kurios dirba prie tam tikros dalies įmonės arba programos to visos. Ir ten turėjau bendradarbiauti tiek su front end, tiek su back end inžinieriais. Norėdamas tuos visus daromus dalykus įdiegti į esamas sistemas.

I.: Aha. Gerai. Iš kur gaudavai duomenis?

P.: Iš kur gaudavau duomenis, tai iš vartotojo interakcijų. Tai viską, ką vartotojai daro platformoje, yra saugoma įstatymo ribose. Modeliams kurti naudojame tuos duomenis, kad apmokyti algoritmus.

I.: Ar tau būdavo, kad kai gaudavai, reikėdavo juos dar patvarkyti tuos duomenis?

P.: Taip. Didžiąją dalį taip. Kadangi duomenų modeliai tada ir dar iki šios dienos ne visuomet yra pritaikyti tam tikrai konkrečiai užduočiai išspręsti. Tai gali būti, kad reikia duomenis pasiimti iš kelių skirtingų resursų, juos sujungti ir apdirbti į atitinkamą formatą, jau paskui, kuris būtų tinkamas mokyti algoritmus.

I.: Ar būdavo, kad trūksta reikšmių?

P.: Ne, nes ta prasme, ar tai, kad nėra kažkokio atributo, reikalingo, ar tai, kad patys atributai yra tušti?

I.: Abu.

P.: Tai, kad kažkokių reikšmių nėra, galbūt dažniau pasitaikytų. Tai, kad pačios reikšmės būtų tuščios, beveik ne, nes nu tikriausiai pagal tai ir atsirenki, kokius atributus naudoti. Jeigu matai, kad iš jo naudos nėra, tai neimi jo ir nebandai ten kažkokia magija užsiimti kurdamas trūkstamas reikšmes.

I.: Tai kai sakai, kad reikšmių nėra, tai turi omeny, kad tiesiog net ne. Pavyzdžiui, jūs susikūrėte sistemą, kuri renka, kiek laiko praleidžia. Bet neparinko, kiek clickų yra, tai tipo nėra reikšmės, kad clickų nėra?

P.: Na, čia daugiau iš tos pusės, kad nežinau. Pabandyti aptikti, ar žmogus yra spameris, ar geras vartotojas. Mums tikriausiai reikia žinoti jo galbūt tikslią lokaciją. Mes tokių duomenų neturime arba

tokių negalime naudoti. Čia galbūt esame ribojami tos informacijos, kurią galime kaupti ir saugoti ar naudoti modelių mokymo tikslais ar galbūt iš tos pusės. Bet jeigu kažko reikia ir tą galima surinkti, tai tą galima pasidaryti. Ir ilgainiui tokius duomenis turėsim.

I.: Tai ten tokių. O tokių atvejų, kad stulpelyje kažkokios. Vienam stulpelyje vienoje eilutėje nėra kažkokio įrašo, tai taip nebūna?

P.: Ne, nu tai jeigu ir būna dėl kažko, kas atsitinka nežinau. Duomenų rinkimo proceso eigoje, mes dirbame su milijonais įrašų ir tas procentas yra toks minimalus, kad tiesiog tu nudrop'ini tas tikriausiai tas eilutes kažkokias. Tai bet čia nežinau. Ta prasme sunku dabar pasakyti, bet turim daug duomenų ir didžiąja dalimi jie visi yra pilni.

I.: O kaip nustato tada? Kokius duomenis reikia rinkti? Kokias interakcijas pvz sekt?

P.: Nu tai iš esmės bandai galvoti pagal problemą kokią nori išspręsti. Jeigu tu matai, kad žmonės įkelinėja netinkamus daiktus, tai tu tikriausiai nori pagalvoti pats. Tarkim, kaip galėtum nustatyti, ar tas daiktas yra tinkamas, ar netinkamas? Tau greičiausiai gali reikėti nuotraukos, nes nuotrauka yra gan stiprus signalas. Gali būti, kad reikės ir aprašymo. Nu, jeigu matai pagal šituos dalykus, kad tikslumas įrankio nėra gan tikslus, gali būti, kad tu gaudai ir gerus žmones, tai galbūt tada papildomai pabandai gauti kažkokius atributus iš pačio vartotojo, tarkim kaip seniai yra užsiregistravęs platformoje. Ar jis yra, kiek turi atsiliepimų, kokie tie atsiliepimai ir panašūs dalykai? Iš esmės pradedi visuomet nuo problemos. Galvodamas, kaip tu pats, kaip žmogus, tikriausiai bandysi tą problemą išspręsti ir pagal tai tada žiūri, kokių tų duomenų gali reikėti.

I.: O tenka tau apmąstyti ar duomenis rinkti, explicitly ar implicitly. Ta prasme, kad naudotojui sąmoningai žinant, kad jo veiksmas bus užfiksuotas ar ten nesąmoningai, pavyzdžiui, seka, kiek laiko žiūri, ten?

P.: Explicitly aš manau, kaip įmonė mes renkame tik tais atvejais, kai prašome konkrečiai kažkokio atsiliepimo apie kažką, ir tai tikriausiai nėra naudojama modelio kūrimui, bet tokie kaip apklausos formatai esamų sistemų gerinimui, susirenkant ten feedback iš tų pačių vartotojų. Gali būti, kad. Yra kažkoks, tarkim, susijungimas tarp tų explicit ir implicit pavyzdžiui duomenų. Tarkim, kaip pavyzdžiui, sužinome, ar vartotojas yra spameris ar ne? Na tai arba jį kažkaip pagauna patys darbuotojai, kurie prižiūri visą tą modeliavimo sistemą, arba patys žmonės juos reportuoja. Tai gaunasi toks explicit veiksmas iš pačių vartotojų. Bet tai nėra ta prasme su intencija, kad va praneš apie šitą vartotoją, kad mes apmokytume savo algoritmus. Vadinasi, mes signalą vis tiek bet kuriuo atveju gauname iš vartotojo, tai sakyčiau, kad vis tiek yra implicit, bet patys atliekami veiksmai gali būti ir explicit.

I.: O tada kodėl dauguma kitų duomenų yra implicit? Kodėl būtent taip renkatės?

P.: Gal gali išsiplėsti pačiu klausimu, kad suprasčiau, kaip ji atsakyti?

I.: Tai supratau, kad dauguma kitų renkamų duomenų būna surenkami implicitly. Tai, kad formų nepildo žmonės. Ten mėgstamiausi brandai ar kažkas tokio?

P.: Okey, apie rekomendacines sistemas, supratau. Nu tai visų pirma žmonės. Jeigu šnekame apie šališkumą pačių duomenų. Gali būti, kad žmonės patys nežino, ko jie nori. Čia yra vienas dalykas arba, tarkim, jie nežino, kad yra kažkas pasiūloje. Tai jeigu vartotojas turi galimybę pasirinkti kažką iš, tarkim, visos gausumos arba jis to nedaro, arba gali būti, kad padarys nu ne taip tiksliai, kaip mes galėtume su algoritmais padaryti, kadangi mes žinome žymiai daugiau iš jo visų interakcijų. Taip, dauguma platformų pačioje pradžioje, kai nieko nežino apie vartotoją, jo paklausia, ar kad ir tas Spotify, Netflix, kokie filmai yra įdomūs arba kokios muzikos klausytum, kad galėtų pradėti siūlyti kažką. Tai (anonimizuotas darbo vietos pavadinimas) tas iš esmės irgi yra. Galbūt problema didesnė yra, kad mes neturime profilių. Tarkim, vienas žmogus gali apsipirkinėti tiek sau, tiek vyrui. Tarkim, jeigu mes šnekame apie moterį, tai tada labai sunku yra kažką pasiūlyt konkrečiai. Ir tada tokiu atveju jo, galbūt būtų galima galvoti apie kažkokią explicit sistemą, kur žmonės galėtų patys pasirinkti, ko, tarkim, jam nesiūlyti. Tam tikrose katalogo vietose. Explici gali vykti ir su implicit vienu metu. Pavyzdžiui, jeigu mes, tarkime, rekomenduojame tam tikrą dalį rūbų. Žmonės vis tiek gali pasirinkti, kokį dydį mėgsta nešioti. Vadinasi, mes filtruojame didžiąją dalį šiaip tų pačių prekių, kurias mes jam rekomenduotume vien dėl to, kad jos yra netinkamos. Tad gali būti, kad tai veikia ir kartu su explicit it su implicit. Kur implicit yra tarkim, rekomendacinė sistema, explicit yra aiškūs kažkokie filtrai, kurio vartotojas pasirenka, kad jam tikslesni tie patys būtų siūlymai.

I.: Kokių imatės priemonių, kad užtikrintumėte, kad duomenys apie naudotoją būtų gaunami kuo tiksliau, teisingiau.

P.: Na, tai čia tikriausiai yra klausimas pačios duomenų platformos, per kurią keliauja visi duomenys, t.y. Paskaičiuojamas metrikos. Koks yra duomenų praradimas? Kaip greit arba lėtai tie duomenys atkeliauja iš vartotojo? Jo, tikriausiai esamų sistemų gerinimas ir testavimas, kažko papildomai nelabai daroma.

I.: Kuri modelį. papasakok, kaip tu nusprendi kokį modelį patį rinktis, tiesiog kaip kūrimo procesas atrodo tavo.

P.: Na tarkim. Tai jeigu problemos pats domenas yra susijęs su tam tikra rūšimi duomenų. Pavyzdžiui, tarkime, jeigu mes šnekame apie tekstą, greičiausiai nesirinksi modelio, kuris yra pagrinde skirtas dirbti vaizdui ir atvirkščiai. Tai pačio modelio, o paskui pasirinkimas. Tai pagrinde žiūri. Aišku, dabar jau tikriausiai šiais laikais ir pagal licenzijas. Jeigu mes šnekame apie tuos visus LLM, tu gali

naudoti verslui ar negali naudoti verslui. Bet jeigu šnekame apie paprastesnius variantus, tai visuomet seki tendencijas, žiūri, kurių modelių pagrindai yra geriausi pagal tam tikras, nežinau ten. Pagal tam tikras metrikas. Na ir tada pasirenki tą modelį, nuo kurio pradėsi statyti visą kitą sistemą. Tai tikriausiai dar kas imponuoja prieinamumas, paprastumas naudoti ir patikimumas to modelio, nuo kurio kažką jau paskui savo build'ini. Nes didžiąja dalimi, jeigu mes šnekame apie visas modernias ML sistemas, kurios nėra paremtos šitų lentelių duomenimis. Tai tu tikriausiai to paties modelio niekuomet nemokai nuo pradžių. Nuo atsitiktinių skaičių, tai jau pasiimi kažkieno paruoštą modelį, kuris sugeba atlikti tam tikrą arba vieną užduotį ir iš esmės tu jį per pernaudoji savo užduočiai atlikti. Tai renkiesi geriausį, prieinamiausį ir efektyviausį tuo metu, kai statai kažkokią sistemą.

I.: Ar susiduri su Cold startu?

P.: Tai čia tas kaip ir minėtas yra jo visos problemos. Tikriausiai platformos, kurios turi rekomendacinis temas, su tuo susiduria, kai mes nieko nežinome apie vartotoją ir iš esmės iš pirmų kelių interakcijų turi pamanyti vartotojui pasiūlyti kažką. Tai Netflix turi vieną sprendimą. Tarkim, pasirenki filmus, kuriuos žiūri. Spotify kitą, (anonimizuotas darbovietės pavadinimas) to kol kas neturi. Na, ir tada mes turime bandyti spręsti pagal pirmąsias interakcijas apie vartotojo elgseną kažką. Tai tikriausiai nebus labai tikslu. Ir jau tada čia yra tiktai vartotojo sąsajos sprendimas. Kaip tada daug mes norime bandyti parodyti mūsų rekomenduojamas prekes. Kai vartotojas yra gan šviežias. Galbūt tu nori kuo mažiau tada kištis, palaukti, kol surinksi daugiau interakcijų pačioje platformoje. Žinoti, kad tikslumas tavo rekomendacijoj yra žymiai geresnis ir tada bandyti jau aktyviau siūlyti kažkokias rekomenduojamas prekes.

I.: Pavyzdžiui, ar būna, kad kai žmogus ką tik susikūrė, tada tiesiog parekomenduoja, kas populiariausia eina?

P.: Jo gali būti toks sprendimas.

I.: Yra koks nors mąstymo būdas, pagal ką nusprendi, tada - geriau mažiau kištis ar geriau parekomenduoti populiariausius ar?

P.: Padaryti eksperimentus. Darai vieną pakeitimą, lygini su kitais, žiūri, kurios metrikos kyla ir tą sprendimą priimi. Gali būti, kad šiandien vienas sprendimas geriausias, rytoj bus kitas. Tai iš esmės nuolat platformos gerinimo procesas.

I.: O tu pats turi įtakos tam, kokias metrikas siekiate gerinti?

P.: Jo, idealiu atveju iš esmės vis tiek įmonė turi tam tikrą tą Šiaurės žvaigždę, tą metriką, prie kurios nori gerinti, ar tai gali būti aktyvių vartotojų kiekis, ar sugeneruota piniginė vertė visos platformos. Ir jau tada tu iš esmės žiūri ne tas metrikas konkrečiai gerini, bet pasirenki tą metriką, kuri iš esmės galėtų

pakelti tą vat esminę metriką. Tai pačios stambiausios metrikos turėtų būti duodamos jau organizacijos viršuje, ir visos komandos pasirenka galbūt smulkesnes metrikas, kurios yra labiau orientuotos į būtent tos tos šito, vartotojo aplikacijos vietoje, kurioje dirba tavo komanda. Tai, pavyzdžiui, jeigu aš dirbu, tarkime, rekomandacinė sistemos komandoj, aš tikriausiai žiūrėsiu vienas metrikas, kurios galų gale darys įtaką pagrindinei metrikai. Jeigu dirbu ten prie vartotojų komunikacijos, kažkokios sistemos, galbūt didinsiu kitas metrikas, kurios irgi didins vat tas galutines kelias metrikas.

I.: Bet tu pats tada gali pasirinkti iš kelių skirtingų mažų metrikų, kurias nori?

P.: Taip

I.: Bet renkiesi, ar pats gali dar apibrėžti, ar pavyzdžiui, o jeigu šitaip metriką apibrėžkime?

P.: Mes turime ir statome sistemą, kurioje bet kas gali pridėti bet kokią metriką ir ją naudoti ir žiūrėti, kaip tos metrikos pakeitimas daro įtaką galutinei metrikai.

I.: Kaip vertini savo sukurtą modelį?

P.: Kaip vertinu šiaip, ta prasme, ar didžiuojuosi? Ar tarkim, kokias metrikas naudoju?

I.: Gali 5 abu atsakyti šiaip, būtų įdomu. O šiaip tai kaip vertini rezultatą? Ta prasme.

P.: Tai yra begalės būdų. Esminis dalykas, kad tu kai modelius optimizuoji. Tu tikriausiai naudoji tokias metrikas, tas vadinamąsias offline metrikas. Iš esmės tas metrikas, kurios yra daugiau skirtos pačio modelio optimizavimui, bet galų gale pats modelis daro įtaką verslo metrikom. Tai didžiąja dalimi tu negali optimizuoti modelio naudodamas verslo metriką, pavyzdžiui, apmokyti kažkokio klasifikatoriaus nuotraukų, ar jis yra padirbinys arba nepadirbinys, naudodamas sugeneruotą piniginę vertę. Tas nėra įmanoma iš optimizacijos pusės, tai tu stengiesi modelį padaryti kuo tikslesniu, naudodamas, tada vertindamas jį pagal tą testavimo duomenų rinkinį. Ir čia tikriausiai tada yra ta vieta, kur tu turi sukonstruoti kuo reprezentatyvesnį tą testavimo duomenų rinkinį, ant kurio galėtum pabandyti bent jau pritempti, kaip tas modelis galbūt. Gerai veiks, kai tu jau jį padėsi jau į veikiančią kažkur sistemą. Vadinasi, jeigu testavimo duomenys yra gana reprezentatyvus ir modelis veikia gan tiksliai, čia mes šnekame apie tikslumą, metrikas, tą F1 įvertį, precission recall, kitas metrikas. Tai darai assumption, kad tas tikriausiai gerai veiks ir jau esamok sistemoj. Na, ir jeigu jis gerai veikia, vadinasi, galbūt pagal tam tikras arba vienas, arba kitas metrikas. Yra du žingsniai, vieną kartą protestuojant konkrečiu optimizavimo metriku ant testų seto. Na ir tada tikriausiai paleidi AB testą ir žiūri, kaip tas pagerintas modelis kelia jau pačias verslo metrikas. Jeigu pakelia - super. Nepakelia - galbūt tada bandai pakeisti savo hipotezę. Žiūri ir nežinau, ką kitaip padaryti. Esminis dalykas vis tiek, bet kuris pakitimas turi kelti kažkokias verslo metrikas.

O kaip vertinu, ar didžiuojuosi ar nedidžiuojuos, tai nežinau. Kiekvieną kartą, jeigu modelis pakelia verslo metrikas, išsprendžia kažkokią problemą, tai viskas yra žiauriai fainai. Jeigu nepadaro, tai grįžti prie braižymo lentos ir žiūri, kurią dalį pakeisti, kad jis tą padarytų.

I.: O ar būna atvejų, kad. Kaip nors nesusikalbat su kolega ir tu įvertini galutinai modelį vienaip, o kitas kolega kažkaip kitaip?

P.: Amm, tai faktas, kad taip gali būti, bet stengiamės visur viską daryti kuo labiau automatizuoti, vadinasi, būtų kuo mažiau žmogaus įsikišimo. Jeigu sistema taip modelį vertina naudodama tokius pačius procesus, tai skaičiai nemeluoja. Tiesiog mes galime galbūt tada jau nesutarti dėl pačio proceso. Bet jeigu procesas yra sutartas ir tu modelį vertini naudodamas esamą sistemą, nu tai tų nesutarimų yra labai mažai.

I.: Ar reikia prieš testuojant iš savo duomenų atskirt dalį duomenų testavimui, dalį duomenų buildinimui ir pan. Ar jūsų atveju nereikia, kadangi turite visą aplinką?

P.: Ne, tai modelio optimizavimo metu tu tikriausiai turi bent tris kažkokias tas pats, tarkim pyragus tų duomenų. Tai yra training, mokymo duomenys, kuri modelį mokai, validavimo duomenų rinkinys, ant kurio žiūri, kaip modelis elgiasi, tarkim, per kažkokias iteracijas. Iš esmės kiekvienos iteracijos metu ar gerėja, ar negerėja. Vadinasi, tai yra duomenys, kurių modelis nemato, kai yra optimizuojami svoriai. Tarkime, jeigu šnekame apie kažkokius neuroninius tinklus. Ir galų gale jau kai tu pabaigi tą visą mokymo procesą, žiūri visiškai dar kartą atskiro tavo testo duomenų rinkinio, kuris gali būti jau visai kitaip išbalansuotas, kad, tarkime, atitikti realią sistemą. Na, tarkim, kai tu mokai, pavyzdžiui, modelį klasifikuoti, ar tai yra tikras, ar netikras daiktas, tu tikriausiai nori turėti kuo galima geresnį balansą. Na, tarkime, 50% 50% padirbinių ir nepadirbinių. Bet paskui, kai, tarkim modelį testuoji, tu greičiausiai nori pasižiūrėti, kaip jis elgiasi ant realaus duomenų paskirstymo, kas tarkim, gali būti, kad jis yra vienas iš dešimties arba vienas prie tūkstančio tikrų arba netikrų prekių. Ir tada jau naudok tokį testo duomenų rinkinį, kad įvertinti tą patį tikslumą, pažiūrėti kaip tikriausiai modelis tiksliai elgsis, kai duomenų kiekis bus visiškai kitoks.

I.: Tu sukuri, tada vertini metriką. Bet ar yra kažkokios ribos, kad nu bent šitiek turėtų pakilt, kad galėtų skaitytis kaip gerai arba nežinau ten ir overfitting'o kokių atvejų išvis būna pas jus?

P.: Čia kai modelį mokome ar kai modelį testuojame AB testo metu?

I.: Tai visais atvejais.

P.: Tai kiekvieną kartą, kai tu. Tikriausiai modelio mokymas ir modelio paskui vertinimas AB testo metu yra du atskiri dalykai. Tai visų pirma žiūri, jeigu modelio metrikos yra gan geros ant to testo seto, tu tikriausiai gali, pavyzdžiui, pabandyti palyginti seną ir naują modelį, naudodamas kažkokį atskirą testo setą, prieš paleidžiant jį į tą pačią darbinę aplinką, kai jau tu pasidarytum AB testą. Tai čia vienas dalykas,

kaip tu, pavyzdžiui, gali palyginti, ar tavo dabartinis modelis naujai apmokytas yra geresnis? Kitas dalykas yra tiesiog paleisti paprastą AB testą. Gali tada naudoti atskirus test setus modelio mokymo metu, bet tada per AB testą pasižiūrėti ar tas naujas modelis yra geresnis, ar blogesnis, o ar modelis overfit'ina ar underfit'ina, tai čia daugiau yra optimizavimo, kažkokios detalės. Gali būt, kad kartais norint išlošti vienoje mokymo stadijoje, kad modelis overfit'intų, kitoje vietoje bandai tą underfit'inimą kažkaip išlyginti. Tai čia per daug Sunku pasakyti, labai priklauso nuo situacijos.

I.: Bet kažkokių apibrėžtų konstantų nėra šitiem overfit'inimam visiems?

P.: Ne, ne nėra.

I.: Kuriuose darbo procesuose matytum rizikas algoritmo kokybės užtikrinimui?

P.: Na, tai jeigu tu paleidi modelį nedarydamas AB testo, vienas dalykas. Nes tada negali žinoti, ar tikrai geriau performina ar ne. Nežinau, netikslumai duomenų procesinimo stadijoje. Kažkokios, tarkim, duomenų srautai nesuvaldyti ar blogai, nežinau, perdirbti ar kažkas kito blogo jiems yra padaryta. Na ir galų gale, tikriausiai šiaip kažkokių procesų nesilaikymas, kaip tarkim, duomenys, turėtų būti atrenkami. Nepadaryti to paties training validation test splito. Ar nesinaudoti gerosiomis darbo praktikomis? Tikriausiai visa tai veda prie prastų darbo rezultatų.

I.: Gerai tada, kur matytum pats asmeniškai. Rizikų, grėsmių iš etikos pusės. Žvelgiant į machine learning procesus. Tavo darbe.

P.: Na, tai tikriausiai kiekvienas priimamas sprendimas turi būti balansas tarp vienų arba kitų vartotojų grupių. Ypač nežinau, tarkim, čia ne tik (anonimizuotas darbovietės pavadinimas), visur kitur. Jeigu Netflixas pradėtų rekomenduoti tik tam tikrus pelnus. Gali būti, kad trumpuoju laikotarpiu metrikos pakyla, bet ilguoju nukrenta, nes žmonės nebemato įvairovės kažkokios ir pabaigia žiūrėti tam tikrą turinį. Na, pabando kažko kito netyčia, tarkim Netflixas to turinio nebeturi. Su (anonimizuotas darbovietės pavadinimas) tada tarkim jeigu mes pradedame siūlyti tik vienas prekes. Gali būti, kad kiti vartotojai, kurie jas įkelia, labiau nuliūsta, nes neparduoda. Tai vadinasi, trumpuoju laikotarpiu galbūt metrikos pakyla, alguoju nukrenta. Jeigu šnekame vėlgi apie kažkokias ypač modeliavimo sistemas, tai kiekvienas netikslus modeliavimo žingsnis kenkia patikimumui platformos arba ne išmoderuotas dalykas gali pakenkti žmonėms, nes jie dažnai praranda pinigus arba dar kažkas blogo atsitinka. Iš esmės labai daug yra rizikų kiekvienam žingsny. Tai kiekvieną dalyką turi pabandyti pasimatuoti ir įvertinti, kas blogiausiai gali atsitikti.

I.: Čia modeliavimo. Turi omenyje, kad kažkokia preferencija yra tam tikrai grupei produktų.

P.: Ne nu tarkime, modeliavimo pavyzdžiu. Jeigu mes, žmogus įkelia prekę ir mes pasakome, kad čia yra padirbinys, o tai nėra padirbinys. Bet žmogų užblokuojam. Arba atvirkščiai.

I.: Tai kai sakai moderavimas gali kenkti patikimumui?

P.: Na tai patikimumui pačios platformos. Jeigu žmonės kels prekes, mes juos bausime už tai, kad jie nėra kalti, nu tai koks pasitikėjimas ta pačia platforma?

I.: Supratau. O ar tau tenka kada nors spręsti dilemą, tarp geriau galse positivus ar false negative? Ir kuris kuris?

P.: Tai va čia ir yra ta viena iš optimizavimo dalių, kai tu naudoji, tarkim realaus pasiskirstymo duomenis. Pavyzdžiui, kiek mes turime counterfeit padirbinių realiai? Ir tarkim bandai tada tą precission recall paadjust'inti. Ir bandai su tokiu spėjimu galbūt paleisi AB testą? Galbūt net leidi, pavyzdžiui, tris skirtingus AB testus, tą patį AB testątik su skirtingais variantais ir tas tresholdas skiriasi. Žiūri iš esmės, kuris geriau veikia, tarkim, kuris geriau kontroliuoja tam tikras verslo metrikas.

I.: Ar yra vis tiek kažkoks išankstinis sprendimas, kad geriau tegu praeina koks nors fake'as, bet neužmoderuosi ir ne užban'insi nekalto naudotojo. Ar geriau, tegu nebūna jokių fake'ų, bet kartais išmesi gerą naudotoją?

P.: Vėlgi priklauso tikriausiai nuo įmonės situacijos, nes visuomet kompiuterio arba algoritmo priimami sprendimai turi būti patikrinti žmogaus. Tai jeigu žinome, kad mes turime labai daug, tarkim, resursų ir mes galime kiekvieną atvejį du kartus patikrinti, tai tikriausiai mes jau geriau, tarkim daugiau žmonių nubausime ir tada jie bus atleidžiami nuo bausmės. Bet jeigu mes žinome, kad mūsų sistema arba ištekliai nėra pakankami, tai jau galbūt geriau įsivertinti, kad nežinau. Vienų ar kitų vartotojų geriau jau, tarkim nenubausti, nes, tarkim, daugiau nukentės. Viskas galų gale atsiveria į tas metrikas, kur modelį pakeiti ir pamatuoji. Tai.

I.: Kas, tavo nuomone, daro didžiausią įtaką galutiniam rezultatui modelio? Turint omeny nuo naudotojo, kurio elgesys sekamas ir paverčiamas duomenimis, iki duomenų rinkėjų, apdorojimo, viso ko.

P.: Nesakyčiau, kad elgesys yra sekamas. Jo. Tikriausiai, kaip čia pasakyt, yra toks išsireiškimas - shit in, shit out. Jeigu paso duomenys bus gan prasti, nereprezentatyvūs, tikriausiai ir pas tave modelis gali būt prastas. Didžiąja dalimi visų, kas susiję su automatizuotais kažkokiais algoritmais, kurie yra mokomi iš duomenų. Esmės visada yra dalykas, geri duomenys. Kitas tikriausiai sekantis dalykas tai visų pirma hipotezė gera irgi nustatyta. Negali būti, kad išspręsi problemas, kurios nėra arba ne toj vietoj sistemos. Tai vat šitie du dalykai: geros hipotezės suformulavimas ir gerų duomenų turėjimas, sakyčiau yra 95 procentai darbo.

I.: Tai šitas darbas krenta ant kokių žmonių pečių. Jeigu reiktų įvardinti kažkokius asmenis, nuo kurių elgesio labiausiai priklauso rezultatai.

P.: Tai nuo komandos tikriausiai priklausė, vienoj komandoje daugia vat šitus, tarkim dizaino sprendimus, kur ką reikia kontroliuoti arba keisti, jas prisiima tarkim dizaineris ar produkto owner'is, kitur daugiau turi iniciatyvos data scientist'ai. Nėra kažkokios nusistovėjusios dalykas. Kiekviena komanda yra skirtinga ir turi skirtingus procesus iš kur tos idėjos kyla? O dėl pačių duomenų, tai vėlgi turi gerai planuoti ir galvoti. Jeigu matai, kad esamai sistemai kažko trūksta, galbūt tada imiesi iniciatyvos ir tuos duomenis pradedi rinkti. Arba jei matai, kad duomenys yra prastos kokybės, ieškai, kur yra problema. Nežinau, kažkur yra kodo klaidos paliktos, kad tie duomenys prastai atkeliauja ar dar kažkokie dalykai, tai vistiek iniciatyva kyla jau iš tų žmonių, kurie su tom problemom bando spręsti. Ir nėra, kad kažkas paskui pirštais ta prasme mosikuoja ir bando sakyti, kas yra kaltas? Dažniausiai būna team effort'as.

I.: Gerai. Ar koreliacija vis dar nelygu priežastingumo?

P.: Vis dar? Tai niekuomet nelygu koreliacija priežastingumui. Ta prasme. Ne, dažniausias atvejis, nelygu koreliacija priežastingumui.

I.: Kodėl dažniausiai?

P.: Ne, nu tai gali būti, kad tu pataikai. Čia daugiau, kad ta prasme. Ne visa koreliacija yra lygi priežastingumui, bet tikriausiai visas priežastingumas yra lygus koreliacijai, jeigu tai yra teisinga iš tos pusės pažiūrėti.

I.: Kontekstas kodėl šito klausiu. Nes kokiais 2005 metais vienas dėdė parašė straipsnį, sakydamas, kad turime tiek daug duomenų, kad mums neberūpi, ar jie ten priežastingi, ar ne. Tiesiog išnaudokime koreliacijas.

P.: Aha, jo nu tai paskui būna, kad tikriausiai tu sprendi žinai. Kaip čia pasakyt? Džiaugiesi, kad tavo modelis yra geras, nes tarkim pagavo daugiau šitų spamo ir vien dėl to, kad Indijoj internetas buvo atjungtas tuo momentu, kai pakeiti modelio versiją ar ne? Tai vat vat ir negali žinoti, nepadaręs AB testą, kur tu pamatuoji seną ir naują modelį. Tu labai apsidžiaugsi ir pradės iššauti šampaną ir duris spardyt, kad tau algą kelt. Nors realiai kažkas kito atsitiko.

I.: Kuriuose etapuose tavo darbo kasdienybėje mąstai apie naudotoją?

P.: Tai tikriausiai visuose, nes visas metrikas... Visų pirma mąstai apie tam tikras metrikas. O metrikos yra apie galų gale generuojamas tų pačių vartotojų. Jeigu mes šnekame apie produktine komandas. Jeigu mes šnekame apie infrastruktūrines komandas, kur aš daugiau irgi dabar darbuojuosi, tai vėlgi mąstai apie kažkokius tikslus, o tikslai yra pasiekiami per žmones ar pačius tuos darbuotojus. Tai kuriame sistemas, kad naudotų žmones arba padėtų žmonėms. Tai žmonės visuomet yra neatsiejama dalis.

I.: Ar turi kokį nors pavyzdį, kur taip labai ryškiai tau padėjo mąstymas apie žmones padaryti sprendimą.

P.: Nu su tuo pačiu spamo nežinau, gal dabar jau mąstai ne apie gerus, bet apie blogus žmones, kai nes iš esmės šitas dalykas yra katės ir pelės žaidimas kiekvienu metu. Tu padarai kažkokį pakeitimą, jie bando tą pakeitimą apeiti aplink. Bandai gaudyti tekstą, pradeda rašyti kirilica. Ir visuomet tada apie naujus modelio pakeitimus bandai žiūrėti iš va tos elgsenos, kažkokio patterno tų blogų žmonių. Jeigu matai, kad tavo modelis nebesugeba, tarkim, veikt, nes pradeda kirilica rašyt, greičiausiai tu apskritai tada gali visus užblokuoti žmones, kurie kirilica rašo, nes mes nesame tuose marketuose, kur žmonės kalba kalbomis, kuriose yra kirilica susirašinėjimo būdas. O jeigu taip ir yra, tai tikriausiai bus labai mažas procentas. Gal toks pavyzdys.

I.: O kaip pats jautiesi naudodamas kitų sukurtus Machine learning paremtus produktus?

P.: Kuo daugiau pats prie tokių dirbi, tuo mažiau nori Teslos autopiloto. Galbūt aš taip pasakysiu. Nes žinai, kad 99,9% tikslumas. Mes turime gan didelę imtį. Gali būti ir trys avarijos įvažiavus į sunkvežimį, kur susimaišo su dangum. Tas pats yra apskritai, kai matai, kaip yra, taip daug bandoma įdiegti tų pačių to chatGPT ar kokį kitokį panašiu įrankiu. Visame kitame, tarkim, valstybės valdymo procese. Tai irgi nėra labai patikima, juolab, kad atsainumas ir nepatikrinimas tam tikrų dalykų gali kainuoti vienus ar kitus sprendimus. Pavyzdžiui, nežinau. Nuo bylų nagrinėjimo iki sveikatos sistemos. Jeigu nešnekant apie duomenų privatumą, bet šnekant apie sistemos patikimumą, net ir man atrodo 12 tais metais netgi tas pats Jeffrey Ketonas, kuris gavo Nobelio premiją, sakė, kad viskas. KaI buvo išrastI, tie naujesni konvoliuciniai neuroniniai tinklai. Na, kad tokio darbo kaip radiologas nebeliks - matome iki šių dienų, kad po 10 15 metų tas darbas vis gi yra. Bet kas atsitiko, kad galbūt tas darbas buvo padaromas efektyvesnis? Vadinasi, žmonės gali dirbti kartu sistema, kad priimti geresnius sprendimus. Aš matyčiau tokią ateitį ir pats tikiu, kad. Nebent tikrai yra kažkokie aiškūs, trade off'a, kur tu gali paleisti tą sistemą. Pavyzdžiui, tas pats modeliavimas. Nes žinai, kad žmogus galų gale vėl tą peržiūrės. Taip efektyvinti procesus yra gan gerai, bet jeigu tu viską paleidi, ypač kai tu net nežinai, kaip tos sistemos iki galo veikia, tai man atrodo, nėra gerai.

I.: O pats naudoji GPT dalykus?

P.: Taip taip. Nežinau, teksto redagavimui, geriau negu Grammarly ir pigiau. Kodo rašymui. Kaip tik kolega, kur žinai, nori paklausti vieną ar kitą dalyką daryti. Juolab, kai, pavyzdžiui, mano atveju dirbu su trim skirtingom programavimo kalbomis. Ir nežinai žinai, galbūt ten nesi prodigy visų ir tikriausiai nori parašyti kodą geriau, refaktorinimas ir visi kiti dalykai. Galbūt netgi ir testo parašymas tikrai tą darbą paefektyvina. Bet galų gale sprendimus turi priimti tu.

I.: O. Apie privatumą, iš duomenų privatumo pusės. Kaip žiūri, kaip jautiesi? Kiek esi atviras savo duomenis duot dėl programos panaudojimo?

P.: Nu visuomet tai pasveriama yra ir žiūri, ar tau tikrai to reikia. Jeigu žinai parsisiunti app ten žibintuvėlį, kuris flashlight'ą jungia ir jam reikia tavo location ir kontakto access. Na, greičiausiai neduodi to. Bet ar esu paranojiškas, ne. Nu tai Facebook vis dar turiu su Instagram. Jei matau, kad tas benefit, kad kiekvienas nežinau, Instagram reels scroll'as man pasiūlys geresnį content'ą, tai tebūnie. Jau taip ar taip žino tas informacijas. Bet taip, kad duočiau tai įmonei arba institucijai, kurios nepasitikiu. Tai taip nėra. Iš esmės tai perteklinės informacijos niekuomet nesuteikiu.

I.: Ką darai, jeigu artėja deadline? Bet dar modelis neveikia kaip norėjai?

P.: Prasiilginam deadline. Niekuomet ne iki galo iškepto dalyko niekur nedeploy'inam, o tada tikriausiai paskui ateina jau geresnis tik tai planavimas. Galbūt etapais kažkokiais modelį gali daryti ir panašiai.

I.: O yra galimybių kokių nors ten kažką truputį apmažinti reikalavimus, kad truputį greičiau pasidarytų?

P.: Nežinau. Greičiausiai vis tiek tada imi kažkokių alternatyvių sprendimų. Jeigu matai, kad. Nu visų pirma šiaip čia labai blogai, jeigu, tarkim tu pradedi iš karto nuo labai sudėtingų modelių, bet pabandydamas kažko žinai paprastesnio arba kažkokią euristinę taisyklę. Tai gali visuomet bandyt tikriausiai žvalgytis į kažkokias alternatyvas vietoj to, kad turėt prastesnį modelį kažkokį nedadarytą. Tai ten pavyzdžiui, vietoje to, kad turėti kažkokį labai gerą transformerį tekstui procesinti ir bandyti spam'ui aptikti, galbūt tau užtenka nežinau ten, kažkokio entity recognition modelio, kuris tarkim pagal tam tikrus raktažodžius galėtų bandyti aptikti tą spamą ar kažkokį kitokį phishing'ą. Bet taip, kad nedadaryta modelį kažkokį ten bandyti diegti, tai sėkmės atvejis. Labai labai labai retas.

I.: Tai kaipgi apibūdintum, kas yra geras ir kas yra blogas machine Learning'as?

P.: Pats modelis ar geras ar blogas?

I.: Nu jo, nu nu ne tik modelis, bet aplamai visame procese žiūrint.

P.: Dažniausiai kas atsitinka šiuo metu, kai modeliai yra labai dideli, tu visuomet darai trade off'ą tarp explainability arba modelio suprantamumo ir jo, tarkim, gerumo, kaip jis gerai veikia. Dažniausiai taip ir būna, ypač su dabartiniai tarkime, LLM'ais. Tai idealiausiu atveju tu nori turėti gerą modelį, kuriuo gali pasitikėti, nes supranti, kaip jis veikia arba kokius, arba vienus kitus sprendimus daro. Tai čia, sakyčiau, yra idealus atvejis. O paskui darai tradeoff'us tarp to jo suprantamumo ir jo gerumo. Jo pats gerumas yra vertinamas kaip ir šnekėjom, arba kažkokiom sintezinėm metrikom, optimizavimo metrikom ar galų gale testais, ar verslo metrikų prizme.

I.: Kiek patį naudotoją atitinka modelyje naudojami duomenys tavo nuomone?

P.: Gali pakartoti klausimą.

I.: Kaip gerai patį naudotoją atitinka modeliuose naudojami duomenys?

P.: Tai patį kaip žmogų, tarkim, jų kažkokius atributus greičiausiai gauna modelis to ir nelabai naudoja. Jeigu mes šnekame apie rekomendacines sistemas. Vis tiek daugiau tu bandai suprasti iš kažkokio žmogaus elgsenos arba vieno ar kito daikto mėgimo arba nemėgimo. Tas pats, kai ateini į parduotuvę ir paklausia, kokią kavą man galėtumėt pasiūlyti, tai greičiausiai žinai klausi kažko apie kavą, ne apie patį žmogų. Žinai, ne kiek tau metų? Norėdamas tau pasiūlyti. Tai sakyčiau, modelyje naudojami duomenys atitinka žmogų tiek, kiek kiek reikia pačiam modeliui ir kokios informacijos yra naudingos. Kažkokio perteklinės informacijos kiekio niekuomet nenori naudoti.

I.: O kodėl nenori pertekliaus?

P.: Kodėl nenori? Kompleksiškumas. Ne gerai, nu tai pačios sistemos kompleksiškumas. Kuo mažiau kažkokių judančių dalių, tuo geriau. Tada tikriausiai pats duomenų apsaugos dalykas. Jeigu tau nereikia kažko, tu tikriausiai irgi nenori naudoti. Na ir galų gale efektyvumas - paskui vis tiek modelis yra kažkur padėtas tuos pačius duomenis ir tą modelį nusiųsti. Tai vietoj to, kad tau reikėtų nežinau iš vienos vietos juos ištraukti ir siųsti modelį. Tau reikia iš kelių vietų tą daryti, traukti ir siųsti vėlgi modulį. Tai sistemos kompleksiškumas, duomenų apsauga ir saikas.

I.: Kur darbe susiduri su stresu?

P.: Darbe su stresu? Tai visada blogai, kai kažkas neveikia. Čia vienas dalykas. Antras dalykas, tikriausiai nepasvertų lūkesčių neišpildymas. Visuomet, man atrodo, turi būti sąžiningas sau, keldamas lūkesčius ir paskui, vertindamas save pagal už juos, už tuos užsibrėžtus lūkesčius, žiūrėdamas, kiek tu įdėjai pastangų juos pasiekti. Balansą atradęs tarp šitų dalykų, tai to streso būna mažiau. Bet jeigu kažkas išsibalansuoja, stresas iškart auga.

Informantas Nr. 13

Amžius: 27 m.

Lytis: vyras

Darbo vieta: sąmoningumą skatinančio konsultacinio DI startuolio kūrėjas. Prieš tai dirbęs IT komandų vadovu.

Gyvenamoji šalis: Lietuva

Pilietybė: Lietuvos

Formatas: nuotolinis pokalbis

Trukmė: 54min 30s

Data: 2024 m. lapkričio 7d.

I.: Esu Simonas Bansevičius, politikos ir medijų magistro studentas. Darau tyrimą apie mašininio mokymosi programų etiką. Informuoju, jog bet kada gali nuspręsti nutraukti pokalbį. Ar sutinki, kad šis pokalbis būtų įrašytas ir jo medžiaga būtų naudojama tyrimo tikslais?

P.: Sutinku.

I.: Jei gali prisistatyk, pasakant savo amžių, ką dirbi šiuo, kiek laiko tuo užsiimi?

P.: Man šiuo metu yra 27 metai ir šiuo metu kuriu savo startuolį, kuris yra susijęs su dirbtiniu intelektu. Kuriame dirbtiniu intelektu paremtą programėlę, kuri padeda žmonėms reflektuoti ir geriau save pažinti. O prieš tai dirbau keliose rolėse įmonėje, kuri kūrė dirbtinio intelekto sprendimus dideliems verslams. Buvau duomenų analitikas, buvau produkto vadovas ir tiesiogiai dirbau su klientais, kurdamas HR žmogiškiesiems ištekliams skirtą produktą, paremtą dirbtiniu intelektu. Tai tokie pagrindiniai akcentai.

I.: Kas tave paskatino pradėti kurti savo produktą?

P.: Matoma didelė socialinė problema, kadangi produktas platesne prasme yra susijęs su sąmoningumo self awareness ugdymu. Su kolege ilgai analizuodami socialines problemas iš įvairių perspektyvų, einant nuo fizinės sveikatos, sveikų įpročių iki emocinės sveikatos, emocinės sveikatos sutrikimų, motyvacijos trūkumo, nerimo ir taip toliau, einant iki platesnių socialinių problemų, kaip didelis susiskaldymas visuomenėj. Klimato kaitos problemos ir jų ne sprendimas ar konfliktai šeimose, įmonėse ir t.t. Matėme, kad norisi šias problemas spręsti ir kiek analizavome, tai matėme, kad sąmoningumo trūkumas yra viena iš priežasčių toms problemoms. Dėl to nusprendėme, kad reikia patiems bandyti spręsti problemą ir kurti kažkokį produktą ar paslaugą, kuri padėtų spręsti ją. Ir galiausiai, išanalizavę galimybes nusprendėme, kad dirbtinio intelekto paremtas produktas būtų vienas iš geresnių būdų spręsti sąmoningumo trūkumo problemą.

I.: O kokie argumentai jus paskatino? Matyt, manyti, kad dirbtinio intelekto būtų vienas geresnių, kaip sakai. Ir geriausių kaip sakai.

P.: Tuoj sudėsiu. Yra. Trys aspektai tokie esminiai. Tai, kas siejasi ne tik su dirbtiniu intelektu, bet ir bendrai su skaitmeniniu įrankiu. Skaitmeninis įrankis turi daug didesnį pasiekiamumą žmonių, lyginant su terapija arba tradiciniu koučingu. Tai tradiciškais būdais galima tik vieną žmogų vienu metu pasiekti, o skaitmeniniai įrankiai turi daug didesnį mastą. Ir galime tada spręsti problemą ne lokaliai, o globaliu

mastu.

Dirbtinis intelektas tuo tarpu suteikia galimybę gauti kokybišką paslaugą bet kada, bet kuriuo paros metu, bet kurioje vietoje ir daug pigiau, palyginus su tradiciniu koučingu ar terapija. Tai tas irgi padidina pasiekiamumo ir padidina tikimybę, kad žmonės gaus pagalbą tada, kada jiems reikia, o ne tik turėdami vieną susitikimą kas savaitę ar kas mėnesį.

O didžiausia nauda iš dirbtinio intelekto tai, kad jis suteikia pilnai suasmenintą, personalizuotą patirtį, nes informacija, kuri yra pasiekiama internete ir įprastai senesniais būdais ar per knygas, ar per konsultacijas, mokymus ar dar kažką, tai ji yra bendrinė, o kiekvienas žmogus yra unikalus, jo problemos irgi unikalios, todėl dirbtinis intelektas padeda įsigilinti į žmogaus situaciją ir pritaikyti patarimus, klausimus jam, kad išspręstų savo problemas taip, kaip jam geriausiai suveiktų.

I.: Kiek pasitiki jų sukuriamo produkto suasmeninimu?

P.: Iš kokios perspektyvos? Iš kokybės, kiek jis gerai suasmenina?

I.: Jo, bet leisiu tau pasakyti, kaip tau atrodo. Ta prasme, nenoriu susiaurinti klausimo.

P.: Aišku, visuomet yra kur tobulėti. Dabar tas įrankis yra, mano požiūriu, dar ganėtinai primityvus. Jis prisitaiko pagal tai, ką žmogus rašo ir atitinkamai generuoja klausimus, generuoja pastebėjimus, patarimus, reakcijas į žmogaus pasidalinimą. Tai jau yra labiau pritaikyta negu, tarkim, skaitant kokį nors straipsnį, žiūrint video ar panašiai, kur yra vienpusė komunikacija iš kažkokio žinovo žmogui. Šiuo atveju yra dvipusė komunikacija. Žmogus pasidalina ir gauna jau sukonkretintą klausimą, kuris yra aktualus jo kontekstui, tačiau kitas lygis yra atsižvelgti į ilgalaikį suvokimą apie žmogaus patirtis. Tai kad tas mūsų dirbtinis intelektas produktas, jis kauptų žinias apie žmogų ir apie jo įpročius, tam tikras elgesio tendencijas, vertybes, požiūrius, skirtingus silpnybes ir stiprybes ir taip toliau. Problema, su kuriom dažniausiai susiduria ir pagal tą ilgalaikę informaciją dirbtinio intelekto tada jau įrankį mūsų, kuriame galėtų pritaikyti pastebėjimus, pastebėti tendencijas, kurių žmogus pats nesusigaudo, ir tada pritaikyti klausimus. Ir sakyčiau, kad taip galima užtikrinti didesnę, didesnę ir geresnę personalizaciją, negu pats žmogus gali sau suteikti. Neigu psichologas ar treneris galėtų suteikti, nes dirbtinis intelektas, ilgainiui tobulindamas savo gebėjimus atsižvelgti į visą gaunamą kontekstą, vis tiek galės turėti mintyse daugiau konteksto negu bet koks žmogus, kurie dirba tiesiogiai tokiose srityse kaip terapija arba koučingas.

I.: Tai, kad pasitikslint. Tu sakai, kad aplamai šitas įrankis potencialiai gali būti geresnis už psichologą?

P.: Čia reikia... Gerai, kad iškėlei klausimą, nes reikia labai atsargiai daryti atskirtį. Mūsų tikslas nėra pakeisti psichologo ar terapeuto, nes jie sprendžia problemą ir jų psichikosveikatos sutrikimą. Nemanau, kad dirbtinio intelekto įrankis, ypač dabartine forma, kuri yra teksto, vaizdo ar garso, jis negebės išspręsti rimtų psichikosveikatos sutrikimų. Tačiau dirbtinis intelektas galėtų būti tokios pačios arba galbūt net geresnės kokybės negu asmeninio augimo treneris, gyvenimo coach ar pan., nes tada dažniausiai žmonės ateina su ganėtinai apčiuopiamais, paprastais iššūkiais, kuriems nereikia jau gydymo metodo ar tuo labiau medikamentų, ar dar ko nors kito.

I.: Bet minėjai, kad toks įrankis, turėdamas pakankamai daug žinių, gali užtikrinti geresnę personalizaciją nei pats žmogus ar psichologas?

P.: Jo. Tai čia geras spėjimas, nes esmė gydyme, tarkim, psichologo ne vien personalizacijos reikia. Reikia viena dalis yra per socializacija. Kita dalis yra psichologo patirtis ir žinios, kurias jis turi iš mokslinės perspektyvos. Ir trečias, tai yra gyvas kontaktas. Galbūt taip galima įvardinti daugiau informacijos, kurią psichologas gali sugaudyti. Tai žmogaus kūno kalba, jo kvėpavimo dažnis, jo veido išraiškos ir taip toliau. Dirbtinis intelektas šiuo metu to dar negali sugaudyti ir vargu ar galėsime sugaudyti visus signalus ateityje, nes ten įtraukiant ir kvapą, ir garsą, toną ir t.t. tai psichologas turi pranašumą tikrai su tuo žmogišku kontaktu ir saugumo, kurį gali sukurti žmogui kalbantis. Iš tų mokslinių žinių, perspektyvos. Ilgainiui dirbtinis intelektas tikriausiai gali turėti daugiau žinių negu psichologas ir jas atsižvelgti.

I.: Tai tada kaip apibrėžti, kas yra personalizacija?

P.: Personalizacija tai atsižvelgimas į žmogaus kontekstą, į informaciją, kurią jis pasidalina, ir tada atsakymo, klausimo ar dar kitos informacijos pritaikymas pagal kontekstą. Tai čia galbūt gali panašiai skambėti tas žmogiškas ryšys su personalizacija. Bet tą žmogišką ryšį ne tik tais iš. Jį sunkiau šiek tiek apibrėžti, nes jis patenka ir į psichologinį saugumą, ir į tą bendrą žmogiškumą. Galbūt dar tą reikėtų akcentuoti, kad vien kai žmogus ateina į terapijos sesiją pas psichologą, jis jau yra nusiteikęs, kad jis ateina bandyti padėti sau, bandyti gydyti tam tikras problemas, su kuriomis susiduria. Būnant psichologo kabinete yra užtikrinamas konfidencialumas. Yra tam tikra aplinka, kur žmogus jaučiasi saugiai, kad jis gali kalbėtis ir psichologo garsiniai veido, ir visi kiti signalai sukuria tą psichologinį saugumą, kad žmogus galėtų dalintis ir būti atviras pokyčiams. Dirbtinis intelektas. Iki šiol neturime žmogui lygių, kažkokių humanoidų robotų, tol jis negalės dar to saugumo suteikti. Ir tai vargu ar kada nors ateityje galės suteikti tą žmogiškumą. Tai čia gal vat taip norisi patikslinti, kad ne tiek iš vien informacijos, kiek jis gali sugauti, bet labiau iš atmosferos, kurią sukuria.

I.: Tada apie patį kūrimo procesą. Tiesiog trumpai papasakok, kaip jūsų technologija surėsta. Ir kaip tu kuri visą tą programą?

P.: Tai yra vartotojo sąsaja, programėlė, kurią kuria žmogus naudojasi. Tik čia labiau iš architektūros pusės kalbėti ar apie procesą patį. Apie ką?

I.: Na, iš esmės man įdomu, kaip tu sukuri tą chatbotą. Kokį renkiesi įrankį naudoti? Kaip jį modifikuoti? Galbūt kaip pritaikei, kad galiausiai būtų šitas produktas?

P.: Okay, Tai tada gavos, kad pažiūrėjom į tuos visiems prieinamus didžiųjų kalbų modelius, LLM ir pažiūrėjome, kuris turi geriausią infrastruktūrą, su kuriuo lengviausia kurti ir kuris duoda mums patenkinamos atsakymus. Tuo metu, kai pradėjome kurti tai, ką labiausiai suteikė OpenAI chatGPT. Tai toliau jį ir naudojam šiuo metu ir tada pasiėmę jį, naudodami jų api, paruošiame skirtingus prompt'us, instrukcijas tiems modeliams, kaip jie turi reaguoti skirtingose situacijose. Pateikiame pavyzdžių, kaip jie turėtų bendrauti tame pokalbyje, kokius atsakymus turėtų suteikti maždaug tam tikrose situacijose, kad irgi išmoktų, kaip kokią atmosferą norime sukurti pokalbio metu ir tada, kai naudotojas mūsų programėlėje pateikia tam tikrą užklausą, mes ją nusiunčiame į OpenAI, prieš tai ją apdirbam, suteikdami tą kontekstą iš pavyzdžių pusės, iš tai ką žmogus yra pasidalinęs jau pokalbio istorijoje ir t.t. ir kitą aktualią informaciją, priklausomai nuo to, kokią funkciją naudoja naudotojas ir pridedame instrukcijas ir tada nusiunčiamas į openAI, gaunam atsakymą ir tada grąžiname naudotojui.

I.: Ar galėtum? Jei negali viskas gerai, bet pasidalinti keliais pavyzdžiais, kokius ten pridėjot tuos. Apmokymo pavyzdžius.

P.: Sudėtinga šiek tiek, nes tie pavyzdžiai yra ganėtinai išsamūs. Ten, pavyzdžiui, viena iš funkcijų yra, kada naudotojas reflektuoja ir tada tas įrankis suteikia galimybę išanalizuoti naudotojo refleksijos atsakymą. Ir tame refleksijos atsakyme mes esame pateikę tam tikrus pavyzdžius, kaip, pavyzdžiui, jeigu naudotojas ten, konkrečiai ten sakiniais dabar neturiu šalia, bet parašė, pavyzdžiui, apie savo santykius su draugais ir taip toliau ir tada, ką ten toje dienoje veikė ir kodėl patiko susitikimas su draugu. Tada mes esame pateikę pavyzdžius, kokius, kokias sakinio dalis reikėtų pabrėžti, kaip potencialiai analizuojamas, ir tada kokį pasiūlymą pateikti, kaip jas reikėtų išanalizuoti. Pavyzdžiui, užsiminė, kad šiandien susitikau su draugu ir tada smagiai praleidome laiką. Tai tada mūsų pavyzdys tam modeliui yra, kad paėmus tokį sakinį, pasiūlymas analizei yra, kad išsiaiškinti, kodėl patiko leisti laiką kartu su draugu. Analise triggers ar kažkas tokio. Tai pateikiame tiesiog pavyzdžių, panašių, kad jisai suprastų, už kokių sakinio dalių reikia užsikabinti ir tada kokį pasiūlymą pateikti žmogui. Čia vienas iš tų pavyzdžių.

I.: Ir kiek tokių pavyzdžių reikėjo?

P.: Pradinėi versijai ten nėra daug. Yra labiau naudojamas tas few shot mokymasis, tai, kad yra nuo poros iki ten 5 ar daugiau pavyzdžių tame prompte.

I.: O dabar kiek yra?

P.: Tai taip tipo ir yra, skirtingi, daug yra skirtingų promptų, priklausomai nuo funkcijos ir tada kiekvienam yra nuo tų keletas pavyzdžių. Tikslas yra ateityje turėti ir pridėt daugiau pavyzdžių, kad turėtumėm labiau užtikrintą veikimą, bet. Kad tą galėtume įgyvendint. Iš pradžių norisi išsigryninti, kurios funkcijos yra naudingiausios, nes duomenų rinkimas ir teisingas ruošimas užtrunka daug laiko ir resursų.

I.: Va paminėjai duomenų rinkimą, kaip renkat duomenis?

P.: Tai iki šiol buvo, jie sintetiškai generuojami arba rankiniu būdu kuriami. Tai, kad patys pagal tai, ką esame, kalbėję su naudotojais, suprasdami, kaip jie naudojasi mūsų įrankiu, kokiose situacijose dažniausiai ir pagal analitikas. Ten viena iš funkcijų yra ta, kad gali susirašinėti su AI coach ir pagal tam tikrą situaciją. Yra kelios duotos šabloninės situacijos. Matome, kurias šablones situacijos dažniausiai pasirenka naudotojai, ar tai yra, pavyzdžiui, kad turi iššūkių su įpročio pritaikymu, ar sunku yra įgyvendinti savo tikslą ir pan. Tai pagal tokias analitikas, kokį šabloną patys renkasi, pagal naudotojų pačių pateiktus pavyzdžius. Generuojame patys, rašome pokalbių variantus arba įsivaizduojamus atsakymus į refleksijos klausimus ir tada pagal tai juos naudojame apmokymui. Tai kol kas šitaip. Naudotojų pačių duomenų nenaudojame.

I.: Bet yra planas?

P.: Čia dar nėra aiškaus atsakymo, nes grynai pačių atsakymų naudotojų tai tikriausiai ne, nes vis tiek asmeniniai duomenys ir tada sudėtinga tą etiškai ir teisingai naudoti kaip apmokymo duomenis labiau. Nebent pastebėtume bendras tendencijas, kokios temos išryškėjo ir tada sukurti anonimizuotus tokius šablonus, atsakymus, kurie jau gali būti, kurie yra kaip čia... Kurie yra apdirbti taip, kad nebūtų biased atsakymuose ir mokymo pavyzdžiuose, ir tada būtų galima užtikrinti kuo labiau teisingesnį veikimą.

I.: Ką šiame kontekste turiu omeny biased?

P.: Tuoj sugalvosiu per kokį pavyzdį. Tarkime, jeigu dabar pradinių naudotojų tarpe didesnė dalis naudotojų susiduria su nerimo sunkumais ir dažnai patiria nerimą, tada tose situacijose, kurias jie rašo, dažniausiai atsispindi sunkumai su nerimu, stresu ir taip toliau. Ir tada mes tiesiog paimdami jų atsakymus ir apmokydami savo dirbtinio intelekto modelį, kad jis turi šitaip reaguoti situacijose ir tada pažymėdami, kur yra geras atsakymas, kur prastas atsakymas, netiesiogiai užprogramuosim jį, kad jis turi labiausiai gaudyti nerimą visose situacijose, ką žmonės parašo. Ir tada, tarkime, ateityje ateina nauji

naudotojai, kurie susiduria su kitok3iom problemomis. Jie ne tiek dažnai jaučia nerimą, bet, tarkim, turi problemų su pykčio valdymu. Ir tada jie rašo apie savo pykčio situacijas. Tačiau mūsų dirbtinio intelekto modelis labiau yra linkęs ieškoti nerimo atvejų ir taip klaidingai veikia. Tai čia toks dar saugus atvejis, bet, tarkime, ateityje galima žiūrėti ir apie lyčių, rasės, įsitikinimų ir kitokias kitokius skirtumus.

I.: Turite apgalvoję strategiją, kaip užtikrinti, kad to biaso nebūtų?

P.: Tai šiuo metu tiesiog ruošiant mokymuose duomenis ir tada rankiniu būdu. Pradinei versijai žiūrėtum, kad jie apimtų įvairias skirtingas situacijas ir kad būtų jų pasiskirstymas kuo labiau atitinkantis realybę. Tai arba pagal tą normalų pasiskirstymą, arba pagal tai, ką tenka iš mokslinių tyrimų, arba kitokių analizių sužinoti. Tai tokie kol kas metodai dar ilgalaikės, strategijos neturime, kol dar bandome įrodyti įrankio veiksmingumą ir naudingumą.

I.: Normalųjį pasiskirstymą, tai turi omeny, kad ta prasme surastume statistiką, kiek dažniausiai kokie būna ir pagal tai?

P.: Ir kur yra tas statistikos, tas vadinamas bell Curve, kad yra, tarkime, pagal tam tikrą aspektą, kad ten dauguma bus per vidurį maždaug ir tada mažumos bus šonuose, tai pagal tokį.

I.: O dar vis dar šitame liekant klausime. Ar buvo kokių nors momentų, kai galvojote apie šitus bias'us ir susidūrėte su situacija, kad geriau false positive ar falls negative, kad būtų. Ta prasme, kad geriau klaidingai priskirti nesančią diagnozę. Tarsi pykčio žmogui, kuris neturi pykčio arba geriau nepastebėti to, kas yra?

P.: Tokių konkrečių diskusijų dar nesame turėję, tai negaliu patikslinti.

I.: Tai tada aplamai, kur dabar yra didžiausi sunkumai, su kuriais susiduri kurdamas produktą?

P.: Yra tokios turbūt dvi perspektyvos. Tai viena, žiūrint iš to grynai startuolio perspektyvos, kad turim sukurti kažką, kas yra naudinga ir generuojantis pajamas, kad galėtume išgyventi kaip verslas. Tai tas labiau skatina iteruoti greitai, kurti greitai, testuoti, nepulti iš karto kurti etiško AI, kuris atsižvelgtų į visas galimas rizikas, bet tiesiog pasitikėti, kad tie modeliai, kuriuos naudojam, kad jie jau turi pakankamas apsaugas įprogramuotas pas save ir tada fokusuotis į tai, kad suteiktų naudą. Kai matome, kad suteikia naudą, naudotojai naudojasi ir yra pasiruošę mokėti pinigus, tada galim daugiau atsižvelgti į etiškumo ir įvairių tų kraštutinumų kraštutinių situacijų sprendimą. Tai čia sunkumas yra surasti, kaip suteikti naudos naudotojams, kad jie dažniau naudotųsi ir suprasti, kodėl jie nesinaudoja įrankiu ir panašiai. Kaip pritraukti naudotojus, kaip juos aktyvuoti, kad jie išbandytų tas esmines funkcijas, suprastų, kaip jas reikėtų naudoti ir galėtų įtraukti savo esamas rutinas. Tai grynai iš tos startuolio pusės.

Iš labiau tada tokios idėjinės arba programuotojo pusės, techninės pusės. Tai tada iššūkis yra užtikrinti,

kad tas veiktų patikimai, kad duotų. Kuo labiau pasikartojančius rezultatus. Galbūt taip, nežinau, koks teisingesnis žodis būtų, kad būtų labiau deterministinis, kad jeigu davėm tokias instrukcijas, kad jis tokį atsakymą duoda ir kad yra mažiau randomness ar tų haliucinacijų, kurios gali iškilti netikėtai, tai šitą užtikrinti. Tada, kad jisai darytų iš tikro logiškas išvadas pagal tai, kaip jis bando priimti sprendimus, nes kartais vis tiek dar pasitaiko su tais bendriniais modeliais, kad jie neturi dar to common sense. Tai kaip šitą įdiegti? Ir ne tik common sense, bet ir suvokimo iš psichologijos, biologijos, neurologijos perspektyvų, kad atsižvelgtų ir į mokslinę informaciją, ir nepraleistų kažko svarbaus.

Kitas aspektas, kad atsižvelgtų į visą kontekstą, kurį suteikia tai, nors ir tie context windows auga, kiek galima suteikti informacijos modeliui, bet jisai pradeda ignoruoti dalį informacijos - užmiršta. Ypač jei pokalbis tampa ilgesnis. Tai kaip užtikrinti? Kaip naudoti tinkamas prompt'inimo technikas, kad jis atsižvelgtų ir nepraleistų esminių dalykų. Tai ten nuo tokių smulkmenų, kaip, pavyzdžiui. Kaip daug iššūkių buvo, kad jis rašytų tik vieną klausimą, kai koučina, nes rašėme instrukcijas įvairiais būdais, įvairiom frazėm, kad užduok tik vieną klausimą ir ne daugiau negu vieną klausimą. Ir vis tiek. Po keletos apsikeitimų tarp naudotojų ir dirbtinio intelekto pradeda uždavinėti kelis klausimus. Tai tokios minimalios galbūt klaidos iki to, kad suteikiame kontekstą, pavyzdžiui, apie žmogaus seniau turėtas problemas ar kad jų neužmiršti. Ir neuždavinėti pasikartojančių klausimų arba kažkokių nepaliestų, triggerinančių temų, kurios yra jautrios žmogui. Tai tas konteksto atsižvelgimas. Ir tas nu ypač yra aktualu kuriant vis labiau personalizuotą įrankį, kai reikės vis daugiau informacijos jam suteikti.

I.: O jūs, naudodami trečiosios šalies įrankį. Manai, turit pakankamai galimybių savo modelį modifikuoti taip, kad jis vis tik galiausiai pasiektų šitą siekiamą techninį rezultatą, kurį dabar apibrėžei?

P.: Žiūrim iteratyviai ir kiek įmanoma. Taip ir su tuo Define Tuning galima stipriai pakoreguoti modelio veikimą. Tas matosi ir iš kitų sėkmingų produktų kitose industrijos srityse, kur esamus modelius naudoja ir tada pritaiko savo kontekstui ganėtinai unikaliai. O ateityje, jeigu būtų neriboti resursai, tai taip tada būtų galima turėti ir savo modelius. Tik šiuo metu jų treniravimas būtų itin brangus.

I.: Kaip jūs dabar vertinate, pagal ką vertini programos galutinius rezultatus duodamus?

P.: Bendrai sakant, rankiniu testavimu pagal subjektyvius kriterijus. Kas nėra idealu, bet kuriant tokį ganėtinai unikalų produktą, taip jau gaunasi. Tai ir pagal naudotojų atsiliepimus aišku, jeigu jie irgi kažkuo skundžiasi, tai atsižvelgiame. Tai gaunasi, kad koreguodami to modelio veikimą arba gaudami naują šito OpenAI modelio versiją ir testuojame, kaip jis reaguoja skirtingose situacijose, tada kaip susirašinėji. Jeigu nepatinka, tai tada bandom koreguoti instrukcijos pavyzdžius ir pan. Kol mūsų požiūriu jisai veikia taip, kaip atitiktų mūsų lūkesčius, kad tinkamu tonu uždavinėja klausimus, kad.

Uždavinėja reikiamus klausimus, teisingus, tikslius klausimus ar teikia teisingus pastebėjimus ir pan. Tai gaunas toks rankinis testavimas, kuris turi riziką, kad ne visas situacijas sugaudysim, nes vis tiek turim ribotą laiką ir kuris nėra tiek scale'able ateityje irgi. Kad jeigu reikės dažniau daugiau pasikeitimų vyks modeliuose, tai neturėsime laiko tokiam išsamiam testavimui. Tačiau bendriniai tie benchmarks, kurie naudojami modelių testavime. Jie yra labai supaprastinti ir yra ten labiau apibrėžtos aiškios užduotys, kurios neatitinka tai, ką mes bandome sukurti. Tai nebent ateity būtų potencialas kurti savo benchmark atskirus pagal įvairias situacijas, kurias sugalvojame kaip tokias standartines, kurias turi jisai gerai išspręsti ir tinkamai užduoti klausimus. Ir tada jau tenais po kiekvieno pakeitimo praeit pro tuos testus.

I.: O tie benchmark'ai, kur dabar minėjai, tai čia kurie čia - jūsų, čia openAI? Aš pasimečiau.

P.: Vieši. Yra tiesiog tam tikri standartiniai benchmarks, kurie naudojami modelių vertinimui. Tai yra nuoroda. Dabar tiksliai nepasakysiu, bet ten kai kurie yra. Pavyzdžiui, kiek jis gerai sprendžia programavimo užduotis modelis, kiek jis gerai atsakinėja į medicinos egzamino klausimus, kiek jis gerai atsakinėja į teisės egzamino klausimus ir pan. Tai yra pagal standartizuotus testus paruošti. O mūsų situacija yra, kad kiekvieno naudotojo klausimai yra ganėtinai unikalūs, tai reikėtų mums irgi pasiruošti tuos standartinius atvejus, kurie bent tą didžiąją dalį klausimų atsakymų sugaudyti.

I.: Kuriuose savo dabartinio darbo procesuose matai didžiausias rizikas programos galutinio rezultato kokybės užtikrinimui?

P.: Turbūt mokymosi duomenų paruošime ir instrukcijų suteikime, ten tam prompt Engineering, nes kol kas atrodo, kad jie didžiausią įtaką turi rezultatui.

I.: Gali gal pavyzdžių kokių pateikti?

P.: Kad jeigu pagal duomenų paruošimą, tai kiek jau šiek tiek kalbėjome, kad jeigu jie yra biased ar tarkime, nukreipti tik į vieno tipo situacijas, tai tada, kad gali iškreipti modelio veikimą arba jeigu tų pavyzdžių duomenų yra nepakankamai, tai tiesiog nesugaudo tam tikrų situacijų. O iš prompt engineering pusės tai ten įvairiausiai atrodo. Tiesiog modeliai reaguoja į skirtingas instrukcijas įvairiai ir tada jiems reikia atrasti tuos tinkamus raktinius žodžius, kad jie duotų norimą rezultatą. Pavyzdžiui, kaip minėjau, kad ilgai buvo iššūkis padaryti, kad modelis visą laiką po vieną klausimą uždavinėtų einant per coach'inimo procesą, tai reikėjo tada daug skirtingų variacijų išbandyt. Kaip parašyti instrukcijas, kad jis suprastų mumis ir tinkamai veiktų. Arba tarkime, kad kaip paruošti tą tokį vadinamą irgi vieną iš technikų role playing, tada duoti tam tikrą vaidmenį ten tam modeliui, kad jis susirinktų atitinkamą kontekstą iš savo bendrinio žinių rinkinio. Tai ten ar rašyti, kad jis yra Executive Coach ar kad jis yra psychotherapist, ar kad jis yra Personal development Coach ar dar kas nors. Tai irgi daug variacijų reikia išbandyti, kad gautume tinkamo tono, tinkamos intonacijos ir konteksto atsakymus į klausimus.

I.: Kuriuose darbo momentuose galvoji apie galutinį naudotoją?

P.: Turbūt visuose atrodo, dabar sunku susigaudyti. Kai dėliojome, kaip iš idėjinės pusės turėtų veikti, kaip visas tas procesas turėtų būti. Tada atsižvelgėme į naudotojų surinktą grįžtamąjį ryšį, tai, kaip įsivaizduojame savo idealų naudotoją, o tada kaip jis turėtų tikriausiai naudotis mūsų produktu. Į analitikas, kurias esame surinkę, kaip jie naudojasi produktu, tada, kai darome dizainus, irgi atsižvelgiame. Galvojame iš naudotojo perspektyvos. Kai kuriu pačią funkciją, testuojame patys. Jeigu didesnė funkcija, tai testuojame su naudotojais galutiniais irgi žiūrime, kaip jie reaguoja, koks grįžtamasis ryšys. Kai tą patį modelį tobulinu, tada irgi žiūriu iš pradžių iš savo perspektyvos. Ar jis patenkina mano lūkesčius? Ir tada bandau pagalvot, ko tada tikisi mūsų naudotojai, ar suteikti jiems naudos? Tai turime prielaidą, kad visuomet turime į tą atsižvelgti, kad suteiktume kažką naudingo jiems.

I.: Kaip nusprendi, kas jiems yra naudinga?

P.: Didžiąja dalimi šita informacija ateina ir toks požiūrio suformavimas iš interviu, kuriuos turime su naudotojais, ypač su aktyviausiais, kurie daugiau naudojasi produktu ir matosi, kad yra užsikabinę ant jo idėjos. Tai tada per pokalbius ir klausimus bandydami suprasti, kodėl jie naudojasi produktu, kas jiems vertingiausia yra, kokie bendrai jų prioritetai gyvenime, problemos, su kuriomis susiduria ir pan. Pagal tai sukuriame vaizdinį tam tikrą, kaip tas žmogus tikriausiai mąsto. Ir tada bandome tą vaizdinį turėti mintyse, kai vertiname, ar rezultatas atitinka vat tokio žmogaus, kaip mes įsivaizduojame, lūkesčius. Tai čia vėlgi taip žiūrint yra daug rizikos įnešti bias, tai vien dėl, iš to, kaip pats naudotojas atsakė į klausimus, tada iš to, kaip mes juos interpretavom, tada iš to, kaip mes juos prisimenam, ir tada iš to, kaip juos ištransliuojame instrukcijose ir visa kita, tai toks gaunasi jau keletą kartų perdarytas paveikslas, kaip žmogus iš tikro mąsto. Tai pagal tai nusprendžiame turbūt kaip patys testuodami, kas yra naudinga. O tada galutinis įvertinimas yra jau kai išleidžiame funkciją naudotojams ir tada koks jų grįžtamasis ryšys.

I.: Kaip pasitiki savo produktu?

P.: Kiek ar kaip?

I.: Kiek, kaip.

P.: Tipo tiesiog kiek stipriai?

I.: Jo jo jo.

P.: Sudėtinga vertinti, nes niekad nebuvau tokios skalės kažkokios susidaręs, kad būtų realistiškas vertinimas. Jeigu vertinant 10 balų, tai tikrai nėra 10 ar 9, nes matau, jog vis tiek dar kaip ankstyva versija, kurią tobuliname. Turbūt šiek tiek daugiau negu per vidurį, nes jau jaučiasi, kad jis suteikia naudos ir matosi iš naudotojų grįžtamojo ryšio. Bet visąlaik iš to maksimalizmo pusės, tai dar matosi, kad daug kur tobulėti, tai turbūt ties 6 šiuo metu kažkur.

I.: O jei ne skalėj įvertinti, pavyzdžiais kažkokiais?

P.: Nevisai su tuo, kokiais pavyzdžiais galėčiau atsakyti. Ne, galbūt ne pilnai suprantu klausimą?

I.: Nu tiesiog, kad žinai, kad savęs neįrėmintum į ten dešimtbalę sistemą. Nu tiesiog vat, tam tikrame kontekste pasitiki daugiau, tam tikram funkcionalume mažiau ir...

P.: Ok, ok, gerai, dabar aiškiau ačiū. Tai aš manau, kad jis jau tikrai geba generuoti klausimus, kurie gali būti tikrai naudingi, kai žmogus bando reflektuoti arba įsigilinti į tam tikrą situaciją, problemą, su kuria, su kuria dorojasi. Tai tiek ir kiek patys naudojome, tiek ir iš naudotojų grįžtamojo ryšio matosi, kad gauna, gauna klausimus, kurie padeda jiems gilintis į savo mąstymą ir veda per tą procesą. Tai ten jau tikrai gerai. Manau, kad dar būtų ir tikrai erdvės, kur galėtų užduoti gilesnius, taiklesnius klausimus, kurių žmogus nesitiki. Bet čia tokia turbūt idealistinė vieta, kur visąlaik galima tobulinti, bet nėra, kad čia degtų iš kokybės trūkumo. Kur visai neblogai, tai jis tam tikras pvz psichologijos problemas, tarkim perfekcionizmas arba nepasitikėjimas savimi ar panašiai sugaudo, bet matosi, kad dar trūksta konteksto apie pvz. neurologiją arba biologiją žmogaus veikimo ir ten tikrai dar yra daug potencialo. Kur jis galėtų galėtų susigaudyt. Pavyzdžiui, jeigu žmogus sako, kad jam trūksta motyvacijos daryti kažkokį darbą, tai ne tik kad užduotų tokias prielaidas ir klausimus ar yra susijęs su tavo vertybėmis, ar kažkas tokio, bet ir galėtų užklausti, kaip ką vakar veikė ir kaip tavo dopamino lygis buvo paveiktas arba kiek jauties pailsėjęs iš ten fizinės ar psichologinės pusės ir taip toliau. Tai, kad to mokslinio background'o dar trūksta ir ne visada sugaudo galimus trigerius arba dalykus, kurie daro įtaką žmogaus elgesiui. Ir tada trečia dalis, kur dar tikrai trūksta kokybės, nes dar ten nėra funkcionaliai kokybiškai įgyvendinta tai, kad sugaudyti to pačio žmogaus tendencijas, patterns visokius tai, kad ne tik atsižvelgtų į šiuometinį pokalbį, bet atsižvelgtų į visų pokalbių kontekstą ir pagal tai darytų logiškas, pagrįstas išvadas apie žmogaus elgesį ir pagal tai iškeltų prielaidas žmogui, kad jis jau pats įvertintų, ar tai jam rezonuoja, ar ne. Tai dar tikrai nepasitikėčiau sakyt, kad jis sugaudys tavo patterns ir galės padėti gauti reikšmingas įžvalgas.

I.: O kaip tu tada šitą vertinimą padarai, jei nematai žmonių chatų istorijos?

P.: Pagal grįžtamąjį ryšį ir pagal savo patirtį stipriai tai jie. Turiu viziją, kaip idealiu atveju norėčiau, kad įrankis veiktų. Ir tada kai pats susirašinėju matau, ar gaunu tą rezultatą, ar ne?

I.: Tu turi pasidaręs keletą skirtingų savo profilių, kad jis iš naujo išbandytų?

P.: Jo tai yra testavimo aplinka, kur galima kaip nors juos rašinėt.

I.: O naudoji kitų žmonių sukurtus panašaus tipo AI LLM programas?

P.: Jo teko reguliariai nenaudoju, bet teko testuoti skirtingus ir žiūrėti, kaip jie veikia. Tai savo pagrindinius konkurentus peržiūrime, sekame, ką jie kuria, ir tada kartas nuo karto išbandome jų įrankius,

žiūrėti, kaip jie veikia, kad ir iš vienos pusės pasisemti idėjų. Galbūt jie daro kažką kokybiškiau, negu mes įsivaizduojame, kad galima ir pamatytume šiaip, kas jau egzistuoja.

I.: O pats savo tikslams. Asmeninio naudojimo. Ne tiek konkurentai, bet tiesiog kažkokie įrankiai, kurie naudoja LLM?

P.: Grynai kontekstai kuriam mes kuriame, refleksijoms arba koučinimui. Tai buvau tik išbandęs konkurentų įrankius, kad pažiūrėčiau ar gaunu naudą ten po vieną du kartus. O šiaip įrankių su LLM tai aktyviai naudoju, bet tik kituose kontekstuose.

I.: O kokiuose?

P.: Tai pagrindinis įrankis gaunas chatGPT, tai vienas dažniausių kontekstų programuojant, kad galėtų padėti atsakyti į klausimus. Jei kažkas nesigauna, pateikti idėjų, kaip reikia implementuoti tam tikrą funkciją, kurią sugalvoji. Tada kiti yra iš rašymo pusės arba sinonimų pusės, kad jeigu nesugalvoju kaip vienu žodžiu išreikšti tam tikrą idėją, tai tada prašau jo pagalbos. Kaip būtų galima surasti tinkamą žodį? Tada, kai Brainstorminu ir noriu gauti papildomą perspektyvą, išorinę ten, pavyzdžiui, buvo, kad galvojome produkto pavadinimą ir tada prašiau idėjų arba kai ruošiuos mokymams, kuriuos vedu, tada irgi kartais būna, kad pasikreipiu. Kaip būtų galima įgyvendint tokią veiklą ar panašiai? Tai štitie. Ir jeigu būna kokia nors problema, kur nežinau, kaip spręsti, ir tada google'inimas nedavė greito atsakymo. Pvz. buvo problema su kompiuteriu, kad su internetu. Neįsijungė viena svetainė ir nežinau kodėl ji neįsijungia. Ir tada parašiau chatGPT, gavau variantų, kas gali daryti įtaką ir tada per tą jau ėjau per debuginimą. Kol suradau priežastį, tai turbūt čia pagrindinis įrankis.

I.: Tai kur matai daugiausiai etinių rizikų? Susijusių kūrimo procese?

P.: Ok, kol kas tris sugalvoju, tada galbūt dar pakeliui atsiras papildomų. Tai viena yra, kad dirbame su žmonių psichologija ir su psichologija būna įvairiausių problemų. Tai turbūt tas vienas dažniausių atvejų yra su savižudybių rizika. Tai, kad tokiose kritinėse situacijose, kur dirbtinio intelekto įrankis jau tikrai nebėra pajėgus padėti, tai, kad laiku ir tinkamai nukreipiami į pagalbos šaltinius. Dabar, kadangi pats aktyviai domiuosi AI etika ir panašiomis problemomis, tai neseniai pagarsėjo atvejis, kaip character AI naudojosi vienas vaikinas ir galiausiai nusižudė. Ir tada daug kartų buvo cituota, kaip jie kaip character AI visiškai ne laiku ir ne vietoje taikė savižudybių intervencijos algoritmą ir tiesiog klausė, ar turi, ar planuoji nusižudyti, ar turi planą, ne nukreipdami į pagalbos šaltinius. Tai atrodo, kad čia buvo netinkamas pavyzdys, kaip dorojamasi su savižudybės rizika.

Tada antras. Tai, ką aktyviau bandoma spręsti, kad psichologijoje ir bendrai žmonių elgesyje, žmonių elgesiui daro įtaką įvairūs aspektai. Ir, tarkim, kalbant su psichologais arba koučeriais ar pan. kitais

žmonėmis, kurie dirba šioje srityje, jie dažniausiai turi išsirinkę savo vieną mėgstamiausią perspektyvą ir tada per ją vertina visas situacijas. Pavyzdžiui, iš populiarių žmonių imant Hubermanas, kuris dėsto neurologijos paskaitas per YouTube, jis viską vertina pagrinde iš neuro chemijos perspektyvos ir tada, kad jeigu tu prastai jautiesi, tikriausiai dopaminas, dar kažkas kažką. Jeigu kalbėsi su jungistiniu psichoterapeutu, jis sakys, kad yra tavo vaikystės traumos ar dar kas nors kalbės su kitokios psichoterapijos rūšies atstovais, jie teigs iš savo perspektyvos. Tai mūsų tikslas yra suteikti tą tokią, kuo labiau sistemišką perspektyvą, kuri kuo daugiau mokslo šakų apima ir padeda suformuoti tokį visapusišką požiūrį į žmogaus elgesį. Tai, bet čia vis tiek visuomet išlieka bias, kad turėsime savo kažkokį labiau mėgstamą būdą ir tada pasiliksime.

O trečias tai, kas jau ryškiai matosi ir kas visai gąsdina, tai, kad žmonės labai greitai pradėjo formuoti stiprius ryšius su dirbtinio intelekto chat botais. Vėl tas pats character AI pavyzdys, kaip žmonės, ypač paaugliai, suformuoja stiprų ryšį su savo chat botais ir tada jaučia emocinį prisirišimą ir izoliuojami nuo realių ryšių su žmonėmis ir taip toliau. Tai mūsų tikslas. Nuo pat pradžių buvo, kad užtikrintume, kad mes patys žmogų bandome mokyti mąstyti. Ir tikslas, kad tai būtų įrankis, kur jis gali padėti, bet reikiamu momentu jis sustabdytų pokalbį ir sakytų. O, žiūrėk, man rodos, jau turi visas išvadas ir jau galbūt laikas eiti įgyvendinti dalykus. Dėl to ir tikslingai pasirinkome jį vadinti AI Coach. Nedavėme vardo ar kažkokios personos, kad nesikurtų tas personifikuotas ryšys. Tai dar aišku, su tuo žiūrėsime, kaip palaikyti šitą, bet norisi akcentuoti, kad čia yra AI robotas, tai nėra kitas žmogus, tai nėra kažkas, su kuo tu turėtum kurti ryšį ir kas turėtų tave palaikyti. Tai yra tiesiog įrankis, kuris tau padeda analizuoti save ir savo sprendimus.

Tai tas, ir kad va dabar šiek tiek prisiminiau, kad ketvirtas turbūt išskirčiau tai, kad dirbtinis intelektas labai daug kur jau perima žmonių poreikį, gebėjimą mąstyti. Tad kadangi vedu mokymus apie dirbtinį intelektą, jau teko girdėti žmonių pavyzdžių. Ai tai vat, kai tingiu sugalvot kažką, tai paprašau, kad chatGPT man sugeneruotų ir tiesiog nusiunčiu ar laišką ar kažkokį postą, ar dar ką nors. Atrodo žmonės deleguoja mąstymą dirbtiniam intelektui. Tai rizika, kad žmonės tiesiog kuo toliau, tuo labiau tingės mąstyti ir daryti kažkokias sunkias užduotis. Mūsų tikslas yra mokyti žmones mąstyti giliau ir ugdyti sąmoningumą, o ne perimti sąmoningumo funkciją iš žmonių. Nes būtų ganėtinai liūdna, jeigu mes sukurtume žiauriai kietą įrankį, bet kad žmonės tada bet kokią savianalizės užduotį deleguotų įrankiui, o ne sau. Kaip jau teko matyti pavyzdžių iš kitų įrankių, kur kaip vos ne tokią reklaminę naudą ir siūlo, kad, pavyzdžiui, žmogus susirašinėja tenais, reflektuoja ir taip toliau. Ir tada gali paklausti to dirbtinio

intelekto, pavyzdžiui, kokios yra mano vertybės. Tai iš vienos pusės, tai gali būti tiesiog smalsu sužinoti, kokį įspūdį susidarė ir viskas su tuo yra OK. Bet jeigu žmogus vietoj to, kad savęs klausti, kokios yra mano vertybės ir pats žinotų, kas jam yra svarbiausia gyvenime, jis klaustų dirbtinio intelekto. Manau, kad stipriai pramissinau. Esmė, ką bandome pasiekti.

I.: Kokie aspektai turi didžiausią įtaką galutiniam produkto rezultatui? Čia imant viską, nuo to, kaip žmogus elgiasi, nuo to, kaip duomenys gaunami, kaip modelis veikia viską, viską.

P.: Ar gali dar pakartoti klausimą, kad tikrai teisingai mąstyti?

I.: Kokie aspektai daro didžiausią įtaką galutiniam rezultatui?

P.: Viena didžiausių įtakų tai tikrai yra duomenys, skirti apmokymui, nes kiek teko bandyti dirbti su modeliais ar girdėti ekspertų įžvalgų, tai nuo duomenų priklauso, kaip veiks modelis. Tai. Vienas svarbiausių aspektų. Kitas svarbus tai yra pati sistema, kurioje veikia modelis. Tai, kad ypač kai atsirado dabar AI įrankiai, tai atsirado žmonių polinkis visas užduotis perduoti AI, ypač gen AI. Bet realistiškai gen AI yra stiprus vienose užduotyse, bet kitose visiškai prastas. Tai, kaip mūsų sistema tinkamai parenka, kur yra užduotis, skirta generative AI? Pavyzdžiui, klausimo generavime, o kur jau yra užduotis, geriau skiriama tradiciniams algoritmams ar kitokiems mašininio mokymosi būdams, ar dar kažkam. Tai tada, kaip mes tinkamai pasirenkame, kokį įrankį naudojam, kokiai užduočiai? Tai manau, kad tai gali stipriai paveikti, kokį rezultatą suteikiame. Ir šiaip dar didelę įtaką daro tai, kaip įrankis yra pateiktas ir kokiame kontekste žmonės naudojosi juo. Mes turėjome skirtingų atvejų, kaip, pavyzdžiui, žmonės, kurie atėjo išbandyti įrankį ir juo naudojosi grynai kaip chatGPT, tikėjosi kažko tokio. Ir tada parašė klausimą ir tikėjosi, kad jiems atsiųs atsakymą išsamų su nuorodomis. Ten paklausė, pavyzdžiui, kaip man reikėtų išmokti, kaip man reikėtų geriau pasiruošti interviu? Ir mūsų įrankis visiškai nepritaikytas tokiems klausimams. Jis coachina žmogų ir veda per klausimus, kad jis gilintųsi į save, tai tada žmogus, ateinantis su netinkamu klausimu ar netinkamomis intencijomis, lūkesčiais, jis negaus atsakymo ir tada rezultato tikrai negausi ir jis atrodys nekokybiškas. Tai čia jau nuo mūsų, kaip mes komunikuojame ir kaip parodome žmonėms, kaip reikia gauti daugiausiai naudos.

I.: Ar koreliacija vis dar nelygu priežastingumui?

P.: Čia bendrai, ar? Iš mano ribotų statistikos žinių tai vis dar nelygu.

I.: Nesiplėsi daugiau?

P.: Nu, tiesiog atrodo, kad yra galybė tų pavyzdžių, kur yra koreliacija tarp tam tikrų dalykų, bet jie nėra vienas su kitu susiję, ten tų komiškų.

I.: Na gerai, tavo atveju aš kėliau sau galvoj klausimą ar tau apsimoka, šito klaust tavęs tavo atveju? Kontekstas tas, kad tiesiog atsiranda žmonių, kurie sako, kad kai dabar turime tiek tiek daug duomenų,

kartais užtenka. Tiesiog rask koreliaciją ir gerai, kad tavo. O tavo šitam modelio atvejyje čia nebūtinai aišku, vis tiek kažkiek tuo paremtas tas tavo.

P.: Iš dalies yra susiję. Atrodo, kad ypač žiūrint į ilgalaikę perspektyvą, nes mūsų tikslas pvz su tuo koučinimu buvo, kad per klausimus tas dirbtinis intelektas iškelia prielaidas ir padeda suprasti, kas galimai daro įtaką žmogaus elgesiui. Jeigu jam trūksta motyvacijos, tai ne tik klausia, bet ir užmeta užuominas. O va žiūrėk ten, tarkim praeitą savaitę tu mažai miegojai, tai gali būti, kad nuo miego trūkumo tau trūksta miego ir šitas apmokymas jis vyks ir pagal tiesiog tą deterministinius algoritmus. Kas yra standartai, kaip veikia žmogaus organizmas ir panašiai. Bet galimai kažkiek bus ir mašininio mokymosi, kur jis mokosi labiau koreliacijom. Tai jeigu mes matysime, kad dažniausiai, kai žmonės skundžiasi motyvacijos trūkumu ir tada atsakymas yra, kad jiems trūksta miego. Ir tada jie sutinka, kad tai yra tiesa. Tada jau mūsų modelis ir išmoks, kad padarys šita koreliacija tai yra priežastis. Tai čia, manau ir ateina mūsų tas sprendimas, kur norim surasti, kada užduotį duoti mašininiam mokymuisi, kuris, mano suvokimu, paremtas vien tik koreliacijom, o kada duoti algoritmą, kuris yra labiau paremtas priežastingumu, kuris paremtas moksliniais tyrimais, kurie, tikėkimės, kad turi daugiau priežastingumo, ne vien koreliacijos tik psichologiniam kontekste.

Toks gal irgi dar į tą patį klausimą, kad. Nors ir nėra lygu, atrodo, kad kartais vis tiek naudojame kaip prielaidą tam tikrą, tik paliekam galutinį sprendimą pačiam žmogui. Pavyzdžiui, kad jeigu mes, Mes negalime būti užtikrinti, kad žmogui trūksta motyvacijos, nes mažai miegojo. Bet pagal mūsų modelius ir suvokimą apie žmogaus veikimą čia yra tikėtina, nes dažnai žmonės, kai mažai miega, jiems trūksta motyvacijos. Ir tada pateikiame šitą kaip idėją, kaip potencialią priežastį ir tada jau žmogui paliekame sprendimą, ar jis mano, kad tai yra priežastis, ar ne. Arba paskatina padaryti testą, jeigu išsimiegosi kelias dienas, ar pakils motyvacija? Jo, bet čia dabar, kai pradėjau mąstyti, atrodo, kad galima nueiti ir į labai jau ne tiek ezoteriką, bet į tokius kraštutinumus, kad ar išvis priežastingumas egzistuoja? Ar nėra tiesiog labai patikimos koreliacijos? Nes niekad nežinome, kaip iš tikro ten veikia atominiam lygmeny ir taip toliau dalykai, ypač kas liečia žmonių veikimą.

Informantas Nr. 14

Amžius: 29 m.

Lytis: moteris

Darbo vieta: NLP inžinierė, 4 m. patirties.

Gyvenamoji šalis: Vokietija

Pilietybė: Lietuvos

Formatas: nuotolinis pokalbis

Trukmė: 54min 13s

Data: gruodžio 17d.

P.: Man yra 29 metai. Dirbu aš NLP Engineer esu, bet jau esu NLP Engineer jau 6 gal kokie metai sakyčiau taip daugmaž. Tai aš gyvenu ne Lietuvoj ir dirbu ne Lietuvoje. Dirbau Vokietijoje ir ten pradėjau dirbti ir dabar neseniai perėjau į kitą darbą, pradėjau dirbti (universitete). Tai irgi NLP engineer. Tai va, prieš maždaug 4 mėnesius pradėjau čia dirbti. Tai vat, pakeičiau darbą neseniai, bet pareigos išliko tos pačios. Tai pagrinde aš dirbu su conversational AI. Su mūsų universiteto chatbotu ir jo, tenka jį, kaip čia, programuoti: dizainas, content'as. Žodžiu, viskas, kas susiję su tuo Natural Language Processing ir Conversational AI. Jo, nežinau ką čia dar. Ai, komanda klausei, kokio dydžio, tai mūsų komanda yra? Man atrodo maždaug dabar 10 žmonių, tai mes dirbam. Yra daug skirtingų AI komandų, bet aš pagrinde dirbu prie pre sales. Tai reiškia mes dirbame prie chatbot'o, kuris konsultuoja būsimus studentus. Bandome pritraukti juos studijuoti mūsų universitete. Ir mūsų chatbot'o tikslas yra atsakyti į klausimus apie studijas, apie Admission requirements ir visa kita. Tai vat ir yra dar kita AI team, kuri dirba post sales, tai jie jau dirba su studentais, kurie jau yra, jau pradėjo studijuoti ir ten jau kitos temos. Tai va.

I.: Ar jauti pokytį, kai pradėjai dirbti universitetui? Kai prieš tai verslui reikėjo, o dabar...

P.: Nu taip yra. Aš dirbau. Kadangi dirbu pagrinde yra automotive industry. Tai ten šiek tiek yra kitaip. Ir tai yra konsultavimo įmonė. Tai tu dirbi su klientais. Žodžiu, tave samdosi klientai ir tu dirbi jiems. Ten visai kitas procesas yra darbo negu čia. Čia viskas vyksta internall. Žinai, ten mes viską developinam internal, o ten tu dirbi klientui ir su kliento įranga. Žodžiu. Tai vat toks pagrindinis skirtumas. Na ir aišku komanda mūsų univere yra jaunesnė.

I.: Tai papasakok apie savo darbo dieną. Tiesiog kokie procesai.

P.: Labai daug debuginimo. Daug čia visko vyksta. Pagrinde nu ką, tiesiog meeting'ai, pagrinde būna dailies, paskui visokie backlog refinements, spring planning ir pan. kur mes suplanuojame, ką mes darysime ir tada pradedi dirbti tiesiog. Tai aš pagrinde dabar dirbu su mūsų, analizuoju mūsų tuos conversations visus su studentais, reikia statistiką suvedinėti. Kiek studentų, kiek žmonių tapo po, pavyzdžiui, studentais ir pan. Taip pat daug dirbu kasdien beveik su Bot monitoring tema. Tai žodžiu, mes turime užtikrinti, kad mūsų chatbot'as visada veikia visada aktyvus ir nedaro jokių nesąmonių, tikrinam hallucinations pavyzdžiui, ar mūsų, nes viskas yra pagrinde varoma chatGPT ir GPT modelio.

Tai reikia užtikrinti, kad nėra jokių ten haliucinacijų, kad nesąmonių nepasakytų. Jei studentas klausia, "Can I stody for free" chatbot'as "yes, you can" - ne, blogai. Tai reikia užtikrinti tokius dalykus pagrinde. Dabar mano darbas yra tas visas bot monitoring užtikrinti, kad viskas vyksta taip, kaip ir turi. Ir statistika suvedinėti jo. Tai va ir apskritai visų tų conversation analizė.

I.: Tai statistikos suvedinėjimas ne automatizuotas?

P.: Automatizuotas taip. Code'inam.

I.: Tai iš esmės kodini, kaip automatizuotai tai suvedama bus?

P.: Taip, taip, viskas yra. Aš šiuo metu developinu dashboard, tai jis yra pajungtas prie mūsų tų visų conversations, kurie šiuo metu vyksta. Ir tas dashboard'as žodžiu, viską parodo, užfiksuoja kiek ko yra, pvz. kiek yra haliucinacijų, kiek yra negatyvių conversations. Nes mes ir sentimentus tikriname, kiek žmonių yra nepatenkinti kažkuo, pavyzdžiui, ane atsako, "man nepatinka tavo atsakymas" ar kažkas panašaus. Tokius dalykus skaičiuojam. Aišku, viskas automatizuota yra. Tai yra pipeline'as. Taip ir viskas paskui yra display'inama dashboard'e, su grafikais ir pan. Tai va.

I.: O tai, kaip pavyzdžiui, apibrėži, kad nusprendi, jog žmogus buvo nepatenkintas pokalbiu?

P.: Tai turim mes implementinę tą sentiment analizę, tai naudojame transformer pre trained language model. Nežinau, ar tu pažįstamas su šitais dalykais, bet tai yra modelis, kuris yra pre-train'intas jau, ten kažkokios online tarkim data ir mes tiesiog jį naudojam savo užduočiai. Tai kai mums reikia atpažinti, kada yra negatyvi. Kažkokia žinutė, tai dažniausiai būna kažkokie net keywords, kaip toje žinutėje užkoduoti. Pavyzdžiui, jo manymu, "I don't like it. I don't like you answer". O kažkas tokio "yeah, I don't know. I think didn't answer to my question" ar kažkas panašaus. Tai mes fiksuojame tai, kaip žmogus yra nepatenkintas. Ir visa tai daro AI. Jis atpažįsta, kada būtent tie transformer models.

I.: Tai jūs naudojate modelį, kuris jau kažkieno kito yra sukurtas atpažinti sentimentams.

P.: Taip, taip, pre-trainintas jau yra.

I.: Tai tau pačiai nereik galvot kokie keywords ir pan.?

P.: Oi ne, ne.

I.: O jūs, kai naudojat šitą modelį, yra kokia nors dokumentacija, kaip jis veikia? Pagal kokius keywords jis padarytas?

P.: Jo tai yra. Yra ir tie papers visi, čia yra pagrinde tas hugging face transformers, tai ten viskas yra surašyta dažniausiai, kaip veikia ir kodo pavyzdžiai. Jeigu nori daugiau detalių apie tai, kaip veikia tas modelis, aišku tie visi papers jau tie research papers, kur gali pasižiūrėti daugmaž.

I.: Tai haliucinacijas irgi panašiai aptinka?

P.: Ne, su haliucinacijomis mes naudojame GPT modelį. Tai klausiam GPT, ar čia yra haliucinacija, ar ne? Mes turim knowledge base, kurią botas turi naudoti. Pavyzdžiui, kokie yra admission requirements? Tai mes turim knowledge base ir tas knowledge base yra prijungtos prie mūsų chatbot'o ir chatbot'as naudoja tą informaciją, kurią mes turim jau apie tai. Ir mes tikriname ar chatbot'as neįgyvendino kažkokios naujos informacijos. Tai inputas būna visą laiką message atsakymas chatbot'o, ir kažkokia knowledge base dalis, iš kurios chatbot'as tą informaciją ima ir mes lyginame tada ar atitinka. Ar response chatbot'o atitinka tai, kas yra mūsų knowledge base, ar jis haliucinavo ir sukuria kažką, ko net nėra mūsų duomenų bazėje. Pavyzdžiui. Ir tą darome su GPT modeliu OpenAI.

I.: Tai, su tuo pačiu modeliu tikrinat tą patį modelį?

P.: Kažkas panašaus jo.

I.: Ta prasme API tas pats, pasiekt gpt?

P.: Taip. Yra ir kitų dalykų, prie kurių dirbam. Čia ne tik conversational AI. Pavyzdžiui, prieš tai, kur aš dirbau, tai mano klientai buvo draudimo įmonė, tai ten buvo visai kiti reikalai. Ten aš nedirbau su GPT, pavyzdžiui, ir nedirbau su Conversational AI, tai man reikėjo automatizuoti tų draudimo dokumentų visą pre-processinimą. Žodžiu, ką žmonės daro rankiniu būdu. Man reikėjo automatizuoti, tai ten pavyzdžiui, perskaityti kliento emailą, ištraukti tam tikrą informaciją iš kliento emailo, ten draudimo numeris. Kas ten atsitiko? Jeigu apie mašinų draudimą kalbame ir pan. Ir visa tai dariau aš irgi su AI. Bet čia jau buvo kita tema. Čia man reikėjo pačiai sukurti training data, tą patį label'inimą atlikti man pačiai rankiniu būdu. Ir tada aš pre-traininau modeliui. Modelis, kuris jau yra pre-train'intas, aš jį fine tune'inau. Nu, aš nežinau lietuviškai kaip pasakyt. Žodžiu, fine tuning. Vadinasi tas procesas, kai tu paėmei modelį, kuris yra per-train'intas ant online ten kažkokios data kažkieno, research community dažniausiai ane. Ir aš tada tiesiog jį update'inu. Nu ant mano užduoties, mano data, mano duomenų kuriuos turiu. Tai ten buvo žodžiu kitas procesas. Man reikia pačiai train'inti AI modelius. Jokio GPT nieko.

I.: O tai pvz. man dar apie haliucinacijas buvo įdomu. Tai jo, naudojat GPT, kad jis tikrintų haliucinaciją, bet vis tiek po to patikrinat jo patikrinimą?

P.: Taip, reikėjo patikrinti. Čia dar nebaigtas procesas, bet jo, reikia rankiniu būdu. Jo aš turiu skaityti ir žiūrėti, ar tai tikrai yra tiesa. Taip, taip. Ir aš dar. Mes naudojome tokį būdą, metodą, kuris vadinasi explainable AI. Kur paklausi chatGPT, ar čia yra haliucinacija ar ne? Ir prašai dar paaiškinti kodėl. Tai tada tu gauni dažniausiai šiek tiek geresnius atsakymus, nes tada modelis galvoja ir tau duoda net informaciją OK. Todėl ir todėl čia buvo haliucinacija. Tai šiek tiek irgi padeda greičiau patikrinti, bet

netikrini kiekvieno ten to conversation, nes ten tūkstančiai tų. Patikrini tiktai 100 kokį ir daugmaž žinai, kad neblogai veikia ir viskas. Tai tiek.

I.: O šitam tikrinimui automatizacijos kažkokios ar yra?

P.: Suplanuota bus, dar nepradėjome dirbti šita tema.

I.: O kaip tai automatizuoti planuoja?

P.: Dar nežinom kol kas. Dar nesuplanuota, kaip čia. Turim minty, kad reikia tai daryti, tikrinti, ar tikrai viskas atitinka. Nerankiniu būdu skaityti kiekvieną atsakymą, bet dar nepriėjome prie šito. Tai negaliu pasakyt kol kas.

I.: Bet iš esmės dabar kaip sakei paėmė 100 kažkokių. 100 iš kiek maždaug?

P.: Oi. Tūkstančių. Tūkstančiai.

I.: O kai renkiesi, kuriuos rankiniu būdu patikrinti, tai atsitiktinai?

P.: Tiesiog random.

I.: Kažkaip pasiskirstymo kokio tikrinimo?

P.: Ne ne, tiesiog parsisiunti tuos visus conversations ir tiesiog iš eilės pažiūri. Ir viskas.

Tai. Kol kas. Bet turim dar ir tą. Kadangi bot monitoring, tai dažniausiai. Būna kažkas negerai. Jeigu labai ilgas conversation, pavyzdžiui, tai jau tokie būna kartais įtartini ten pavyzdžiui. Nes šiaip conversation nebūna labai ilgas. Studentas būna labai eina prie esmės. Aš noriu tą studijuoti. Ką daryti, kokių dokumentų man reikia ir taip toliau. O kartais būna conversation kur ten 30 žinučių būna išsiųsta, daugiau ir jau būna tokie įtartini, jau reiškia kažkas gali būti negerai. Tai dažniausiai irgi tokius reikia žiūrėti ir skaityti. Viską, ką mes monitorinime, tam dashboard'e, ten daug yra skirtingų dalykų. Haliucinacija viena iš jų, bet ir conversations that are too long? Ten negatyv8s. Juos visus turime peržiūrėti rankiniu būdu šiuo metu.

I.: Visus visus, be išimčių?

P.: Nu, tai priklauso ten nuo. Kuri versija yra chatbot'o? Mes netikriname absoliučiai visų ten 5 metų, tiktai jau tuos naujausius ir naujausios mūsų versijos chatbot'o. O chatbot'o versija keičiasi gan dažnai, kas kelis mėnesius kokius, išleidžiame naują.

I.: O tai kaip gali likti sena versija, jūs ta prasme turit app'są, kurio jeigu user'is neatsinaujina, tai reiškias seną naudoja ar kaip?

P.: Ne, mes naudojam tokius konkrečiai conversation designer. Kaip web tokia kaip aplikacija ir ten mes kuriame template'us, žodžiu skirtingas botų versijas. Ir ten mes galim paskui implementinti ką

mes norim. Nežinau net kaip paaiškint. Vadinasi Conversation Designer. Toks web aplikacija, kuri yra būtent mūsų univero sukurta.

I.: O tai kaip tada gali nutikti, kad žmogus naudos senesnę versiją, kurios?..

P.: Nenutiks, nes versiją live paleidi. Žodžiu, ten daug būna versijų ten. Tu paleidi live tik vieną, nuarba dvi kaip nori, bet ten. Žodžiu, pirma mes testuojame, sukuriame versiją, kuri nėra live ir tik tada paleidžiame live. Kai jau viskas ištestuota ir veikia ta naujausia. Ir naujausia versija dažniausiai turi kažką naujo arba kažkas udpate'inta. Tokie vat dalykai. Knowledge base pvz. kai update'inam, tai atitinkamai ir versiją paleidžiam, nes informacija keičiasi pastoviai. Univere studijos, programos naujos atsiranda ir pan. tai turim pastoviai update'int.

I.: Kai tikrini haliucinaciją galimai atrastą. Ar būna, kad haliucinaciją tikrinantis modelis suklydo. Ir vis tik false positive buvo.

P.: Jo, buvo tokių. Šiaip dažniausiai man pačiai kaip developeriai sunku net pasakyt kartais ar tai yra haliucinacija, ar ne, tai dažniausiai. Nu, mes bent jau kai planuojam daryti, tai tokius conversations prašyti, tikrintis study advisors. Nes jie yra tie domain experts. Jie gali iškart pasakyti ar čia haliucinacija, ar ne. Nes man kaip developer'ei kartais atrodo, ok botas paaiškino, kad čia haliucinacija. Dėl tokių ir tokių priežasčių ir aš būna skaitau ir man atrodo. Nu nežinau, čia tikrai taip ar ne? Tu tiesiog nežinai. Tai, nes tu nesi domain tas expert. Tai tiesiog tai už tokius dalykus dažniausiai yra tie study advisors atsakingi, kur mes planuojame tiesiog siųsti, tikrinti jiems ir paskui automatizuotai viską po kažkiek laiko.

I.: O turi kažkokį mąstymo procesą, kaip nusprendi, kada jau esi užtikrinta, kad žinai, ar čia haliucinacija, ar ne? Kada jau reikia prašyti pagalbos?

P.: Nu tu lygini su informacija, kuri būna jau tam knowledge base ir kartais ji būna labai aiški. Ten būna pasakyta studentas klausia ar galiu studijuoti mediciną? Ir botas atsako Ne, negali. Ir tada būna retrieved information from knowledge base ir ten būna surašytos ten studijų programos ir tu matai, kad nėra tokio dalyko kaip medicina. Tu matai akivaizdžiai ir tu žinai, kad nu ne haliucinacija čia, viskas ok yra. Jeigu botas sakytų taip gali, tu vis tiek atsidarai tą knowledge base. Tu matai informaciją, tu ją skaitai ir tu gali nuspręst. Bet kartais būna tokie labai sudėtingi. Tokie kur, nes viskas vokiečių kalba. Kaip ir sakiau, man kartais būna toks skaitau ir atrodo. Na gerai, aš nežinau. Čia kartais gali tik vienas žodis lemti ar čia haliucinacija, ar ne. Ir kartais man sunku būna pačiai nustatyti vien. Su google vertėju išsiversti, nu bet vis tiek ne taip būna. Tai dėl to toks kalbos barjeras šiek tiek prisideda. Ir jo kartais tiesiog ta informacija taip sudėta knowledge base, kad tu skaitai ir atrodo, net pasimeti. Labai sudėtingai būna aprašytas procesas, tai sunku pasakyt kartais. Jo ir paskaitai boto aprašymą, kad jo čia haliucinacija,

nes tada ta ta ta. Skaitai, tą knowledge base ir atrodo. Na, nežinau, gal gal ne. Tai tada klausiam study advisors. Bet čia nebūna daug tų haliucinacijų, Tai kol kas, kiek pastebėjau, ten labai mažas procentas yra jų.

I.: O yra buvę atvejų, kad tas pats haliucinacijas tikrinantis modelis pateikė knowledge base informaciją ir ją pateikia irgi haliucinavęs?

P.: Oi ne. Ne, taip nebūna.

I.: Čia yra kažkokia prevencija padaryta, kad tikrai taip nebūtų?

P.: Na, knowledge base yra. Kas tai yra data? Tas data base ir modelis negali jos pakeisti. Na ji jau yra define'inta. Viskas. Modelis gali tik klaidingai suprasti kažką ir atsakymą sugeneruoti kažkokį ne tokį, bet pačios knowledge base to viso, nu jis negali pakeisti, jis neturi prieigos net nu, kaip. Negali nieko edit'int ten.

I.: Ta prasme jis pateikia iš šaltinio tiesiai jau turinį iš to knowledge base, jis neperfrazuoja jo?

P.: Perfrazuoja. Tai taip, perfrazuoja.

I.: O tai perfrazuojant, nėra tikimybės, kad blogai perfrazavuos?

P.: Tai dėl to ir tikrini.

I.: Ne, bet ta prasme, tas šaltinis, ant kurio jis stato savo argumentą, tas tikinantysis modelis. Jis pateikia knowledge base, ant kurio jis stato savo argumentą, jog originalus chatbotas meluoja. Tai kaip jisai perfrazuoja tą knowledge base, ar tenais galbūt gali būti klaidos ir tada?

P.: Nu gali, aišku, ten visokių gali būt dalykų. Bet nu kaip ir sakiau, tos haliucinacijos yra tik vienas iš. Tų mūsų bot monitoring tų visų dalykų, tai jo ir čia toks, kur šiuo metu nėra net priority ir aš nelabai galiu daug ten kažką ir pasakyt, nes dirbau prie jo gal tik 2 dienas realiai, o prie visų kitų dalykų ilgai teko dirbti. Sentimentai, visokie conversation ir pan. tai tos haliucinacijos dar kol kas toks naujas dalykas ir. Neturiu daug informacijos.

I.: Iš tų, ką prieš tai, prie ko dirbai daugiausiai?

P.: O daugiausiai, kaip ir sakiau, reikėjo data label'inimą daryti. Žinai, kur tu perskaitai. Priklausomai nuo užduoties aišku. Bet čia jau kitoje įmonėje, kur dirbau, tai būdavo klientas sako - man reikia modelio, kuris išextract'intų žodžiu tam tikrą kliento informaciją iš dokumentų. Ten, pavyzdžiui, kliento adresas, ten insurance number visokie ten, kada nori ten nutraukti sutartį, nes aš pagrinde dirbau su tais sutarčių nutraukimais, kur būdavo klientas rašo draudimui. Sveiki, aš noriu nutraukti draudimą. Žodžiu ten būna visas tas dokumentas. Kodėl. Tai man būdavo reikia ištraukti priežastis, kodėl noriu nutraukti, kada nori nutraukti, adresas, IBAN ten ir taip toliau. Kur pinigus pravesti? Tai man reikėjo šituos visus automatiškai išextract'inti.

Pavyzdžiui, čia vadinasi named entity recognition. Šitas visas procesas ir man reikėjo sulabel'inti dokumentus, aš turėjau rankiniu būdu pažymėti, kur yra adresas, kur yra, ten IBan, kur yra priežastis, kodėl jis nori nutraukti sutartį ir pan. Ir aš tada visą šitą data, apie du tūkstančius dokumentų, sulabel'inau rankiniu būdu ir šitą data visą daviau AI modeliui. Žodžiu. Ir jį fine tune'inau, update'inau jį taip, kad jis žinotų, kaip išextract'int iš to data turėtų pavyzdžius kažkokius. Tai ir pagrinde dirbau prie va šitų procesų, kur aš turėjau daryti visą tą manual label'ing ir tada daryti tą model fine tuning, kur update'inti modelis su mano label'inta data. Čia buvo visai kiti procesai. Jokio GPT tuo metu dar nebuvo. Buvo, bet nebuvo dar toks geras GPT atrodo taip.

I.: Ar buvo problemų darant label'inimą?

P.: Oi, belekiek. Visų pirma vokiečių kalba. Vienas dalykas nebuvo lengva. Ir būdavo daug tokių, kur tu nežinai ką žymėti. Kur tu skaitai, dokumentai, adresas, visa kita. Šitie lengvi dalykai palyginus - gali tu juos dar atpažinti, bet kartais būdavo. Pavyzdžiui, jeigu dirbi su PDF - formatas toks būna. Tu turi tekstą, išextract'int iš PDF ir tu tekstą label'ini, bet ne pdf straipsnį. Žodžiu extractin'tą tekstą. Ir kartais pdf'ai, jie būna tokie. Struktūra tokia sudėtinga ir kai tu extract'ini tekstą su tuo optical character recognition, jis būna visas tiesiog kaip čia, all over the place. Žinai Visas. Nėra struktūros. Adreso viena dalis vienoje puslapio pusėje. Kita adreso dalis kažkur kitoje puslapio dalyje. Ir tu turi taip label'inti. Kas sudėtinga ir aišku, jei modeliui tokius dalykus, tai reikėjo labai didelę dalį dokumentų label'inti, nes tu negali suderinti su ta struktūra pdf. Buvo dalykų, kur, pavyzdžiui, text classification buvo, kur reikėjo dirbti su tam tikrais dokumentais, kur man tiesiog binary classification. Ir kartais tu skaitai dokumentą ir nesupranti, kuri čia klasė ar ta, ar ta. Tu skaitai ir galvoji nu nežinau, nes ten dirbi su kreditų, man atrodo buvo draudimo dokumentai, ir tu nesi domain expert. Tu skaitai ir pasimetęs, tiesiog nežinai, kokia čia label iš vis yra. Tai tokiu atveju reikėjo vėl su draudimo darbuotojais man meeting'us daryt, rodyti jiems šituos dokumentus ir klausti, kokia čia klasė yra, kuriai label reikia priskirti šitą dokumentą. Tai daug kas buvo labai, nes jie patys nenori label'inti dokumentų, o modelio reikia. Ir aš turėjau pati label'intis. Tai buvo daug meetingų su jais.

I.: Kaip jie reaguodavo, kai prašydavai jų pagalbos?

P.: O ne, neturiu laiko. Pats pagrindinis būdavo. Oi, dabar čia neturiu kada. Tai gal kitą savaitę, tada. Tai vat čia toks būdavo sudėtingesnis. Ir tu turėdavai su jais meeting'us organizuoti. Negali tiesiog per teams'us, skype paimt ir parašyt labas, žiūrėk, pasuksiu aš tau dabar. Ten taip neveikdavo. Turi planuotis savaitę, dvi, kad tu galėtum turėt meeting'ą su vienu iš jų. Tai dėl to aš paskui pradėjau Series

of meetings žodžiu daryti, kur kas savaitę automatiškai padaryt, kad visada būtų meeting'as ir kad visada turėtume galimybę pašnekėti ir padiskutuoti, kas neaišku, nes kitaip nu be šansų.

I.: O dėl tokios situacijos nebuvo, kad kartais tiesiog numodavai ranka ir jau iš savo žinių?

P.: Nu negali, nes žinai, yra toks posakis shit in shit out. Tai jeigu tu ten savo nuožiūra prižymėsi visokių nesąmonių, tai modelis tau nesąmones ten predict'ins žinai. Man būdavo dažniausiai, kad aš tiesiog skip'indavau, tiesiog praskipinu ir paskui vis tiek. Tu naudoji labeling tools, paskui sugrįžti. Tu matai, kurie dokumentai nesulabel'inti, tuos tada kartu paklausdavau tų jau domain experts, bet irgi priklausydavo, nes kartais būna tu nežinai, praskipni ir viskas, nes ten tiesiog exception. Bet aš prie jų net negrįždavau. Bet kartais būna, kad tu praskipini ir paskui kažką panašaus vėl matai ir vėl, ir vėl. Ir tu matai, kad nu it's repetitive žinai. Kažkoks yra pattern repetitive, bet tu nežinai, kuriai klasei priklauso. Tai va tokius jau diskutuodavome mes jau su domain experts. Kur tik vieną kartą kažkoks atsirasdavo, neaiškus, tai tiesiog praskipini. O kur jau repetitive, tada jau klausi. Ir čia viskas buvo su tekstu. Kaip ir sakiau, čia natural language processing. Tai teksto visi dokumentai.

I.: Dar sakei su sentimentais reikėjo?

P.: Jo, bet čia jau dabartiniame darbe. Reikėjo atpažinti, kada žmogus atsiunčia kažkokią negatyvią žinutę.

I.: Nu o čia kokie keblumai?

P.: Čia. Šiaip realiai jokių kol kas, tiesiog nereikėjo man nieko label'int nieko. Tiesiog pasiėmėm modelį, kuris pritrain'intas, mes jo neadaptavom nieko ir tiesiog paleidome jį ant savo data, ant savo tų pokalbių. Ir viskas, kol kas normaliai kaip ir suveikė.

I.: Kiek laiko su šituo dirbai?

P.: Su visu šituo boto monitoring dalyku tai mėnesis laiko kažkur.

I.: O tai aplamai šitame universitete kiek laiko?

P.: Keturi mėnesiai. Penktas dabar. Jo, o daugiau tai aš dabar dar dashboard greit atsidarau ir pažiūrėsiu, ką dar mes čia darėm, nes jau pamiršau. Nu tai taip, kaip ir sakiau, tuos negatyvius pokalbius reikia atpažinti ir tie, kurie yra per ilgi. Pavyzdžiui, reikia tikrinti dėl visa ko, kad nebūtų kažkokių nesąmonių. Ir dar kas svarbu buvo. Ai, dar tikrinome, kada yra kalba inconsistent? Pavyzdžiui, user klausia klausimų vokiečių kalba, o botas atsako anglų, tai čia jau yra language inconsistency ir čia nėra gerai. Mūsų moduliui taip daryti. Tai va, tokių turim dar vis.

I.: O šitą testuojat... Pagal, su kuo?

P.: Darome tuos AB test. Jo, nes mes dirbame su skirtingais boto versijomis ir testuojame jas. Tiesiog paleidžiame live ir žiūrim kaip kas. Bet pradžiai internal tai pradžiai nėra paleista, kad studentas

gali naudoti pradžią internally. Mes testuojame viską patys, susirašinėjame su botu ir student Advisors. Ir tada jau žiūrim, tikriname. Tai va vat ir turim tą monitoring dashboard visą, kur matom tuos skaičius ir kas negerai buvo. Ir paskui tikriname tuos visus dalykus. Ir tada paleidžiame live. Jau studentas gali naudot.

I.: O tai su kalbos to nesutapimu, tada kaip nustatote, kad..?

P.: Neišspręsta problema dar kol kas.

I.: O tai dabar kaip sužinojot, kad šitas nutinka?

P.: Tai mes monitorinam šitą dalyką. Čia vienas iš tų, kur haliucinacijos, per ilgi conversation ir pan. Tai mes vienas iš tų metrikų, kur mes monitorinam. Ir mes naudojam irgi pretrain'intą tą tokį modelį Fast text vadinasi ir jo input būna user message. Tada tas bot response ir lyginama ar atitinka kalbos ar ne. Tai irgi pretrainintas modelis tiesiog. Turi daug problemų, nes. Jeigu labai trumpa žinutė, dažniausiai blogai atpažįsta kalbą. Tai jeigu ten vienas žodis ar kažkas, tai dažniausiai klaidingai būna kažką, nes per mažai konteksto yra, kad atpažintų kalbą, tai čia didžiausia problema būna.

I.: O kaip jūs, sužinoję, kad jis turi šitą problemą? Tas modelis?

P.: Man šitą pasakė mūsų product manager, tai jisai turi šitą informaciją. Aš nežinau iš kur. Jis tiesiog reportina mums, kame yra problemos su chatbotu ir mes taisome tas problemas. Nežinau iš kur jisai žino, jo.

I.: Kuriuose savo darbo kasdienybės procesuose matytum didžiausias rizikas kokybės užtikrinimui, programos programos kokybės užtikrinimui?

P.: Didžiausios rizikos kokios yra?

I.: Jo. Kurie darbo procesai yra tokie kebliausi, kur lengviausiai klaidos įsivelia? Kažkokie dalykai, dėl kurių po to kenčia rezultatas?

P.: Visi procesai, kurie. Kuriuos reikia padaryti urgent. Tai va, čia jau didžiausia būna bėda pas mus, nes kai vyksta tas sprint planning visas, defineinam tuos user stories, tuos tickets užduočių ir būna tokių, kur o, šita yra čia ir urgent. Čia reikia kuo greičiau. Ir tokia, tokios užduotys, tokie procesai, kurie labai skubinami yra, tai jie būna patys prasčiausi ir sudėtingiausi, ir daugiausiai bugų, ir daug streso. Tai tie, kurie yra skubinami taip pasakysiu. Gali būt bet kas. Tiesiog jeigu skubiai reikia, čia ir dabar ir dažniausiai būna net neapgalvojama visokie niuansai, exception nebūna apgalvojami, tiesiog va čia ir dabar reikia ir paskui susiduri su tais visais exceptions ten visokie bugai. Žodžiu, labai streso daug ir dažniausiai ir nesigauna. Tai čia toks mano atsakymas būtų abstraktus, toks skubinami procesai.