

**VILNIAUS UNIVERSITETO
KAUNO FAKULTETAS**

**SOCIALINIŲ MOKSLŲ IR TAIKOMOSIOS INFORMATIKOS
INSTITUTAS**

Finansų technologijų studijų programa

Kodas 6211BX022

RUGILĖ PRANSKŪNAITĖ

MAGISTRO BAIGIAMASIS DARBAS

**SUKČIAVIMO ATVEJŲ NUSTATYMO AUTOMATIZAVIMAS
FINANSINIŲ PASLAUGŲ ĮMONĖSE**

Kaunas 2025

**VILNIAUS UNIVERSITETO
KAUNO FAKULTETAS**

**SOCIALINIŲ MOKSLŲ IR TAIKOMOSIOS INFORMATIKOS
INSTITUTAS**

RUGILĖ PRANSKŪNAITĖ

MAGISTRO BAIGIAMASIS DARBAS

**SUKČIAVIMO ATVEJŲ NUSTATYMO AUTOMATIZAVIMAS
FINANSINIŲ PASLAUGŲ ĮMONĖSE**

Magistrantas _____
(parašas)

Darbo vadovas _____
(parašas)

(darbo vadovo mokslo laipsnis, mokslo
pedagoginis vardas, vardas ir pavardė)

Darbo įteikimo duomenys

Registracijos Nr.

Kaunas 2025

TURINYS

SANTRUMPŲ SĄRAŠAS.....	5
LENTELIŲ SĄRAŠAS.....	6
PAVEIKSLŲ SĄRAŠAS.....	7
SANTRAUKA (užsienio kalba)	8
ĮVADAS	9
1. SUKČIAVIMO ATVEJŲ NUSTATYMO AUTOMATIZAVIMO, FINANSINIŲ PASLAUGŲ ĮMONĖSE, APŽVALGA	11
1.1. Sukčiavimo atvejų proceso apžvalga	11
1.2. Sukčiavimo tipai ir kategorijos	12
1.3. Klientų pažinimo procesas ir jo etapai	15
1.4. Mašininio mokymosi algoritmai ir metodai sukčiavimo aptikimui	17
1.4.1. Prižiūrimi mokymosi metodai.....	18
1.4.2. Neprižiūrimi mokymosi metodai	20
1.4.3. Pusiau prižiūrimi mašininio mokymosi metodai	21
1.5. AutoML įrankiai	22
1.5.1. H2O AutoML.....	23
1.6. Modelio tikslumo vertinimo kriterijai.....	25
1.7. Funkcijos parametrai modeliuose ir susiję darbai.....	27
2. SUKČIAVIMO ATVEJŲ NUSTATYMO AUTOMATIZAVIMO, FINANSINIŲ PASLAUGŲ ĮMONĖSE, METODOLOGIJA	32
2.1. Sukčiavimo atvejų nustatymo automatizavimo tyrimo pobūdis.....	32
2.2. Sukčiavimo atvejų nustatymo automatizavimo tyrimo metodas	32
2.3. Sukčiavimo atvejų nustatymo automatizavimo metodo našumo metrikos.....	34
2.4. Empirinio taikomojo sukčiavimo atvejų nustatymo automatizavimo tyrimo atlikimas	37
3. SUKČIAVIMO ATVEJŲ NUSTATYMO AUTOMATIZAVIMO, FINANSINIŲ PASLAUGŲ ĮMONĖSE, REZULTATAI	39
3.1. Duomenų rinkimas ir jų reikšmė.....	39
3.2. Duomenų paketų paruošimas naudojant „Tableau Prep“	41
3.3. Sukčiavimo nustatymo modelis	43
3.3.1. Sukčiavimo atvejų nustatymo modelio kintamieji.....	44
3.3.2. Sukčiavimo atvejų modelio automatizavimo kodo aprašymas.....	45

3.3.3.	Visų nepriklausomų kintamųjų reikšmė modeliui	47
3.3.4.	Atrinktų nepriklausomų kintamųjų reikšmė modeliui	49
3.3.5.	Papildomų nepriklausomų kintamųjų reikšmė ir svoris modeliams	50
3.3.5.1.	Kintamojo, (ne)patvirtintų transakcijų vidurkio, reikšmė ir svoris modeliams	50
3.3.5.2.	Kintamojo, mažų transakcijų vidurkio, reikšmė ir svoris modeliams	52
3.3.5.3.	Kintamojo, visa kliento transakcijų suma, reikšmė ir svoris modeliams.....	54
3.3.5.4.	Kintamojo, visas kliento transakcijų kiekis, reikšmė ir svoris modeliams	57
3.3.5.5.	Pasirinktų nepriklausomų kintamųjų eksperimentų apibendrinimas	59
3.3.6.	Reikšmingų kintamųjų ir IF balo įtaka galutinio modelio pasirinkimui.....	60
3.3.6.1.	Modelio su atrinktais kintamaisiais metrikos, parametrai ir svoris	60
3.3.6.2.	IF balo reikšmė ir svoris modeliui	62
3.3.6.3.	Galutiniai modeliai, jų tikslumas, parametrai ir svoris	65
3.3.6.4.	Galutinio modelio pasirinkimo analizė.....	67
IŠVADOS IR PASIŪLYMAI		70
LITERATŪRA.....		71
PRIEDAI.....		76
1	PRIEDAS.....	76
2	PRIEDAS.....	84
3	PRIEDAS.....	85
4	PRIEDAS.....	86

SANTRUMPŲ SĄRAŠAS

KYC – „Pažink savo klientą“
CDD – klientų patikra
SCDD – supaprastinta klientų patikra
EDD – sustiprinta klientų patikra
RF – atsitiktinis miškas
XRF – paaiškinamasis atsitiktinis miškas
GBM – gradientinis stiprinimo mechanizmas
XGBoost – ekstremalus gradiento padidinimas
GLM – apibendrintas linijinis modelis
OCSVM – vienos klasės palaikymo vektorinė mašina
AE – automatinis kodavimo algoritmas
IF – izoliaciniai miškai
ML – mašininis mokymasis
AutoML – automatinis mašininis mokymasis
TP – tikrosios teigiamos reikšmės
TN – tikrosios neigiamos reikšmės
FP – klaidingos teigiamos reikšmės
FN – klaidingos neigiamos reikšmės
AUC – plotas po kreive
DL – giliausias mokymasis
OLAF – Europos kovos su sukčiavimu tarnyba
ES – Europos Sąjunga

LENTELIŲ SĄRAŠAS

1 lentelė. Sukčiavimo skirstymas pagal sektorius	12
2 lentelė. CDD ir EDD palyginimas	17
3 lentelė. Modelių metrikos ir jų aprašymas	26
4 lentelė. „ <i>users.csv</i> “ duomenų failo analizė	39
5 lentelė. „ <i>transactions.csv</i> “ duomenų failo analizė	40
6 lentelė. „ <i>fx rates.csv</i> “ duomenų failo analizė	40
7 lentelė. Trylikos kintamųjų metrikos	47
8 lentelė. Trylikos kintamųjų modelių parametrai	48
9 lentelė. Nepriklausomų kintamųjų svoriai ir metrikos	49
10 lentelė. Kintamojo, (ne)patvirtintų transakcijų vidurkio, metrikos	50
11 lentelė. Kintamojo, (ne)patvirtintų transakcijų vidurkio, parametrai	51
12 lentelė. Kintamojo, mažų transakcijų vidurkio, metrikos	53
13 lentelė. Kintamojo, mažų transakcijų vidurkio, parametrai	53
14 lentelė. Kintamojo, visa kliento transakcijų suma, metrikos	55
15 lentelė. Kintamojo, visa kliento transakcijų suma, parametrai	55
16 lentelė. Kintamojo, visas kliento transakcijų kiekis, metrikos	57
17 lentelė. Kintamojo, visas kliento transakcijų kiekis, parametrai	57
18 lentelė. Šešių kintamųjų modelių metrikos	60
19 lentelė. Šešių kintamųjų modelių parametrai	61
20 lentelė. Modelių, su papildomu IF balo kintamuoju, metrikos	63
21 lentelė. Modelių, su papildomu IF balo kintamuoju, parametrai	64
22 lentelė. Modelių, su papildomu IF balo kintamuoju, svoriai	64
23 lentelė. Galutinių modelių metrikos	65
24 lentelė. Galutinių modelių parametrai	66
25 lentelė. Geriausio galutinio modelio sumaišties matrica	68
26 lentelė. Trylikos kintamųjų visų imčių tikslumas	84
27 lentelė. Šešių kintamųjų visų imčių metrikos	85
28 lentelė. Galutinių modelių kintamųjų visų imčių metrikos	86

PAVEIKSLŲ SĄRAŠAS

1 pav. Tapatybės sukčiavimas ir jo procesas.....	14
2 pav. Mašininio mokymosi metodai	18
3 pav. AutoML technologijų charakteristikų santrauka	23
4 pav. H2O AutoML įrankio veikimas.....	25
5 pav. H2O AutoML kodo pavyzdys Python programavimo kalba.....	34
6 pav. Sukčiavimo atvejų nustatymo automatizavimo modelio parinkimo etapai.....	38
7 pav. Klientų duomenų paketų tvarkymas	41
8 pav. Nepriklausomų kintamųjų svoris modeliui.....	52
9 pav. Nepriklausomų kintamųjų svoris modeliui.....	54
10 pav. Nepriklausomų kintamųjų svoris modeliui.....	56
11 pav. Nepriklausomų kintamųjų svoris modeliui.....	58
12 pav. Nepriklausomų kintamųjų svoris modeliui.....	62
13 pav. Nepriklausomų kintamųjų svoris modeliui.....	67

SANTRAUKA (užsienio kalba)

Pranskūnaitė, Rugilė (2025). *Automating Fraud Detection in Financial Services Companies*. MBA Graduation Paper. Kaunas: Vilnius University, Kaunas Faculty, Institute of Social Sciences and Applied Informatics. 70 p.

SUMMARY

This research paper is carried out with the aim of proposing and implementing an automated fraud detection model that can be used in financial service companies. The goal of this work is achieved through five main tasks. First, the types and categories of fraud cases, the principles of their detection, the criteria for evaluating the models' performance and related works by other authors are summarized. The workflow and methods based on which a fully automated fraud detection model is developed are also presented. The data used in the study were processed using the data preparation software package "Tableau Prep". Model development is achieved using the supervised machine learning tool H2O AutoML. This research also investigates the impact of a new variable, obtained by the unsupervised machine learning algorithm "Isolation Forest", on the model metrics and the percentage distribution of weights of independent variables. Finally, the best automated fraud detection model is proposed. The results obtained in this empirical research work show that models often tend to overfit if they contain many variables. The most realistic model with the lowest overfitting rate (before the additional independent variable was added) was obtained when the model included six independent variables, using the GBM algorithm with 140 trees, a maximum depth of 10, and a sample rate of 0,8, but one variable was strongly dominant in terms of percentage weight. Adding the additional variable to the model normalized the distribution of the weights of the independent variables, but no positive impact on the performance of the models was observed. Adding the additional variable to the analysis showed that the models overfitted more often due to the better distribution of weights, so it is assumed that the dominance of one variable prevents the model from overfitting. As the best and most realistic model, with a better distribution of variable weights, was proposed another model, which also obtained with the GBM algorithm, but it includes seven independent variables including additional "Isolation Forest score" feature, the number of trees is 180, the maximum depth is 8, and the sample rate is 0,8. The model predicted 96 472 normal predictions and 1 369 fraud cases, with an error score of 25 %. This research paper consists of 67 pages, 28 tables, and 13 figures.

IVADAS

Nors ir šiandieniniame pasaulyje, vis greičiau atsiranda pažangios ir naujos technologijos efektyviai kovai su įvairiais nusikaltimais, tuo pačiu greičiu daugėja ir nusikaltimų atlikimo būdų. Vienas tokių nusikaltimų, su kuriuo labai dažnai susiduriama šiandieninėmis dienomis, yra sukčiavimas. Ši sąvoka apibūdina elgsenos spektrą, apgaulės būdu siekiant gauti naudą. Sukčiavimas gali būti susijęs ir su mokėjimais, ir su tapatybės pasisavinimu, investicijomis, rinkodara bei kt. (Button et al., 2009). Europos bankininkystės institucijos ir Europos centrinio banko pateiktos analizės duomenimis, skelbiama, kad 2023 metų pirmajame pusmetyje, sukčiavimas per kredito pavedimus ir mokėjimus kortele sudarė didžiausią dalį sukčiavimo atvejų: apgaulingi kredito pavedimai siekė net 1,131 mlrd. eurų, o sukčiavimas per kredito korteles – 633 mln. eurų (The Global Treasurer, 2024).

Tam, kad būtų galima sumažinti dėl sukčiavimo patiriamus nuostolius, kas šiais laikais, yra labai **aktuali tema**, yra būtina kurti efektyvias sukčiavimo atvejų nustatymo sistemas ir kuo greičiau ir efektyviau užkirsti kelią šiam nusikaltimui. **Problemos ištyrimo lygis:** automatizuotas sukčiavimo atvejų nustatymas dažniausiai atliekamas naudojant mašininio mokymosi paremtus prižiūrimus ir neprižiūrimus metodus (Carcillo et al., 2021). Tam, kad minėtos problemos sprendimo efektyvumas būtų geresnis ir naudojami metodai kuo tikslesni, būtinai tobulinti ar ieškoti naujų būdų, algoritmų ar sistemų, kuriais šį procesą būtų galima padarytų kuo tikslesnį ir reikalaujantį mažiau rankinio darbo, tad šiame **darbe nagrinėjama problema**, kuria ieškoma kaip patobulinti jau esamus sukčiavimo atvejų nustatyto automatizavimo metodus. **Darbo objektas** – sukčiavimo atvejai finansinių paslaugų įmonėse.

Darbo tikslas: pasiūlyti automatizuotą sukčiavimo atvejų nustatymo modelį, naudotiną finansinių paslaugų įmonėse.

Darbo uždaviniai:

1. Apibendrinti sukčiavimo atvejų tipus ir kategorijas, nustatymo principus bei palyginti sukčiavimo atvejų nustatymo automatizavimo metodus.
2. Sudaryti sukčiavimo atvejų nustatymo metodologiją, kuri prižiūrimo mašininio mokymosi pagalba leistų automatizuoti sukčiavimo atvejų nustatymą.
3. Eksperimento būdu išsiaiškinti sukčiavimo atvejų nustatymo modelio papildomą nepriklausomą kintamąjį, jo reikšmę ir svorį modeliui.
4. Nustatyti, papildomo nepriklausomo kintamojo, gauto neprižiūrimo mašininio mokymosi pagalba, įtaką modelio metrikos ir procentiniam svorių pasiskirstymui.
5. Pasiūlyti galutinį automatizuotą sukčiavimo atvejų nustatymo modelį.

Darbo naujumas grindžiamas tuo, kad darbe pateikiamas H2O AutoML įrankio pagalba sukurtas automatizuotas sukčiavimo atvejų nustatymo modelis, kuriame taip pat tiriama naujo kintamojo įtaka, gauto neprižiūrimo mokymosi metodu. **Darbo struktūra** susideda iš literatūros analizės, kuria apibrėžiamas sukčiavimas, jo praktika, lyginami mašininio mokymosi metodai, sukčiavimo atvejų nustatymo automatizavimui, darbo metodologijos sudarymo, kuria aprašoma viso šito darbo eiga, sukčiavimo atvejų nustatymo automatizuoto modelio sukūrimo, ištyrimo ir jo pritaikymo bei papildomo kintamojo, gauto neprižiūrimo mokymosi pagalba, įtaka modelio metrikoms bei procentiniam svorių pasiskirstymui. **Naudoti literatūros šaltiniai** šiame tiriamajame darbe daugiausiai yra užsienio šalių moksliniai straipsniai bei jų tyrimų duomenys, bei taip pat remtasi ir publicistiniais, internetiniais šaltiniais. Moksliniai straipsniai apima 1999-2024 metus, tačiau dauguma jų yra nauji, išleisti 2015-2024 metai. Tuo tarpu, publicistiniai straipsniai, yra tik nauji, išleisti 2021-2024 metais. **Darbo ir tyrimo metodai** apėmė mokslinės literatūros analizę, kurios pagalba nustatyti sukčiavimo tipai, principai ir išskiriami bei palyginami sukčiavimo nustatymo metodai empiriniam taikomajam tyrimui. **Darbo praktinėje dalyje** sukurtas ir patikrintas automatizuotas sukčiavimo atvejus nustatantis modelis, paremtas prižiūrimo mokymosi algoritmu bei ištirta naujo kintamojo, gauto neprižiūrimo mokymosi algoritmu, reikšmė modelio metrikoms ir kintamųjų svorių pasiskirstymui. Pasiūlytas automatizuotas modelis geba tiksliai ir gerai aptikti realius sukčiavimo atvejus.

Tyrimo ribotumai iškilę darant šį tiriamąjį darbą: duomenų ir jų detalių trūkumas – gautas automatizuotas sukčiavimo aptikimo modelis geba mokytis ir aptikti atvejus tik iš tyrime naudotų duomenų, kurie pagrįdė apima klientų transakcijų, jų tipus, sumas, susijusias šalis ir kita, tačiau, dėl duomenų apsaugos įstatymo, pateikti duomenys neturėjo pilno spektro klientų duomenų, kurie gauti iš „Pažink savo klientą“ anketos metu. Būtent dėl to modelis apsiriboja tik transakcijų duomenimis, o norint kad jis būtų dar geresnis, ir aptiktų sukčiavimą ne tik pagal transakcijas, bet ir pagal bendrą kliento profilį, būtų galima papildyti minėtais duomenimis. Gautas modelis pasižymi patenkinamomis ir realiomis metrikomis bei kintamųjų svorių pasiskirstymu, o norint modelį pagerinti reikėtų naudoti tikslesnius analitinius įrankius, kurie nebuvo naudoti dėl finansinių ribojimų.

Šio **darbo struktūrą ir apimtį** sudaro trys pagrindiniai skyriai, tokie kaip literatūros analizė, metodologijos aprašymas bei jos pritaikymas praktiškai, kuriant automatizuotą sukčiavimo atvejų nustatymo modelį. Tiriamąjį darbą sudaro 70 puslapių, 13 paveikslų bei 28 lentelės.

1. SUKČIAVIMO ATVEJŲ NUSTATYMO AUTOMATIZAVIMO, FINANSINIŲ PASLAUGŲ ĮMONĖSE, APŽVALGA

Norint automatizuoti sukčiavimo atvejų nustatymą, sukuriant tai darantį modelį, pirmiausia reikia suprasti šio termino reikšmę. Magistro tiriamojo darbo literatūros apžvalgoje pateikiama sukčiavimo atvejų koncesija, susijusių darbų analizė bei galimi metodai, kuriais naudojantis galima skurti ar patobulinti sukčiavimo atvejų nustatymo automatizavimą. Taip pat pateikiamas ir modelių našumo metrikos, kuriomis remiantis vertinamas modelio veikimas.

1.1. Sukčiavimo atvejų proceso apžvalga

Šiais technologijų laikais, elektroninės prekybos naudojimas yra neatsiejamas nuo daugelio organizacijų, įmonių ir vyriausybinių agentūrų. Tokiose srityse, kaip telekomunikacijos, sveikatos ar automobilių draudimas, nuotoliniai aukcionai ir kitos sritys, kuriose, prekiaujant produktais ar paslaugomis, naudojamos kredito kortelės, elektroninė prekyba yra neatsiejama dalis, kuria norima padidinti produktyvumą bei efektyvumą (Bolton ir Hand, 2002). Tačiau, elektroninės prekybos sistemomis naudojasi ne tik teisėti, įprasti, su nusikalstama veikla nesusiję vartotojai, bet tuo pačiu ir sukčiai, dėl kurių, rinka tampa labiau pažeidžiama (Philip ir Sherly, 2012).

Yra daug sukčiavimo ir nesąžiningos veiklos apibrėžimų, tačiau šis procesas vis tiek reiškia tą patį – sunkus nusikaltimas, kuriuo siekiama pasisavinti lėšas neteisėtomis priemonėmis (Bolton ir Hand, 2002). Sertifikuotų sukčiavimo ekspertų asociacija (angl. *The Association of Certified Fraud Examiners*) apibrėžia sukčiavimą kaip: „savo profesijos naudojimą asmeniniam praturtėjimui, tyčia netinkamai naudojant įdarbinančios organizacijos išteklius ar turtą“ (ACFE, 2002). Autoriai Alexopoulos et al. (2007) sukčiavimą apibūdina kaip melagingą naudos siekimą neteisėtomis priemonėmis, tad šis procesas daro dramatišką poveikį ekonomikai, teisei ir net žmogaus moralinėms vertybėms. Plačiau nagrinėjant visą rinką, beveik visos technologinės sistemos, susijusios su pinigais ir paslaugomis, gali būti pažeistos nesąžiningų veiksmų, todėl sukčiavimas šiose sistemose yra vertinamas kaip kibernetinis nusikaltimas, sukeliantis didžiulius finansinius nuostolius (Abdallah A. et al., 2016).

Bazelio bankų priežiūros komiteto teigimu, yra įvairių sukčiavimo rūšių, kurios visos patenka į dvi kategorijas (Bhowmik, 2011):

1. Vidinis (arba, kitaip tariant, profesinis) sukčiavimas, kuris įvyksta, kai darbuotojas (-ai) sukčiauja prieš savo organizaciją;

2. Išorinis sukčiavimas, kuris visiškai kitaip nei vidinis, apima daugybę schemų, įskaitant pardavėjus, klientus arba kitų trečiųjų šalių įvykdytas vagystes. Yra trys išorinių sukčių tipai:

- vidutiniai nusikaltėliai, priskiriami „švelniam“ sukčiavimui;
- kriminaliniai nusikaltėliai;
- organizuoto nusikalstamumo nusikaltėliai, priskiriami „sunkiam“ sukčiavimui (Bhowmik, 2011).

1.2. Sukčiavimo tipai ir kategorijos

Kaip ir minėta 1.1. poskyryje, kadangi yra daugybę sukčiavimų tipų, bei rūšių, juos skirstyti galima labai įvairiai, ir kiekvienas autorius labai skirtingai juos įvardija savo moksliniuose tyrimuose. Button, Lewis ir Tapley (2009) teigimu, sukčiavimą galima skirstyti į privatą ir viešąjį, kurie pateikiami 1 lentelėje.

1 lentelė

Sukčiavimo skirstymas pagal sektorius

Privatus			Viešasis		
Finansinis	Nefinansinis	Individualus	Nacionalinis	Vietinis	Tarptautinis
Draudimo	Žaidimų	Vartotojų	Išmokų	Tarybos mokesčių	Viešieji pirkimai
Mokėjimo kortelių		Labdaros			
Skolinimo		Pensijų	Mokesčių		ES fondų
Pirkimo		Investicijų			
Padirbti pinigai			Pasisavinimas		
Suklastota intelektualinė nuosavybė			Pirkimų		

Šaltinis: sudaryta autoriaus remiantis Button et al. (2009).

Yra daugybė variantų, kaip gali būti atliktas sukčiavimas. Toliau pateikiamos kategorijos, pagal kurias skirstomi sukčiavimo atvejai.

Masinės rinkodaros sukčiai. Button et al. (2009) analizuoja, kad nėra vienareikšmio apibrėžimo masinės rinkodaros sukčiams ir apgavikams, bet autorius juos išskiria į:

- a) Azartinių lošimų sukčiavimas – asmuo kviečiamas dalyvauti įvairiose loterijose, pavyzdžiui, kai gaunami elektroniniai laiškai iš nežinomų ar nesusijusių įstaigų, kuriuose teigiama, kad auka laimėjo prizą ir galės jį pasiimti tik susimokėjęs vienkartinį mokestį. Tokiu būdu nusikaltėliai gauna tiesioginę prieigą prie aukos lėšų. Taip pat aukos gauna tekstinius pranešimus, su raginiu paskambinti nurodytu numeriu norit atsiimti prizą, tačiau šiuo atveju,

kaip išskiria autoriai, prizas visada yra gaunamas – tik ne tokia suma, kuri buvo žadėta (Button et al., 2009).

- b) Lengvų pinigų užsidirbimo sukčiai – sukčiavimo kategorija labai dažnai sutinkama, kai siūlomas greitas uždarbis dirbant iš savo gyvenamosios vietos. Sukčiai vilioja siūlydami gerą atlygį, geras darbas sąlygas, lakstų grafiką, dažną algos mokėjimą, nors iš tikrųjų tikrasis atlygis yra žymiai mažesnis nei buvo siūloma. Taip pat, dažnai dar įvardijamas „piramidinis“ pardavimas, kada teigiama, jog aukos gaus didesnę atlygį, įtraukus daugiau naujų darbuotojų, tačiau iš tikro, uždirba tik nedidelė dalis, esančių „piramidės“ viršuje (Button et al., 2009).
- c) Netikri produktai ir paslaugos – su sveikata ar lieknėjimu susiję produktai, kuomet klientams siūloma pirkti produktą, kuris tikrai žinoma, kad nėra veiksmingas. Sukčiai taip pat kėsinaisi net ir į karjeros galimybių siekiančius žmones, kurie įdarbinami kaip modeliai, išradimų patentuotajai, dalyviai konferencijose, o galiausiai aukos pačios turi susimokėti (Button et al., 2009).
- d) Neteisėti sukčiai – sukčiai, kurie apgaulingai užsiima neteisėta veikla, ieškodami bendrininkų. Potencialioms aukoms dažniausiai fiktyviai vaizduoja korumpuotą vyriausybės pareigūną, kuriam reikia naujos banko sąskaitos, kad persivestų pasisavintą didelę pinigų sumą. Aukų prašoma sumokėti mokesčius, o už jų padengimą žadamas atlygis (Button et al., 2009).
- e) Technologiniai sukčiai – siunčiami elektroniniai laiškai arba sukuriama iššokantys langeliai svetainėse, kurias spustelėjus, atsiunčiama programinė įranga, pakeičianti interneto nustatymus (Button et al., 2009).

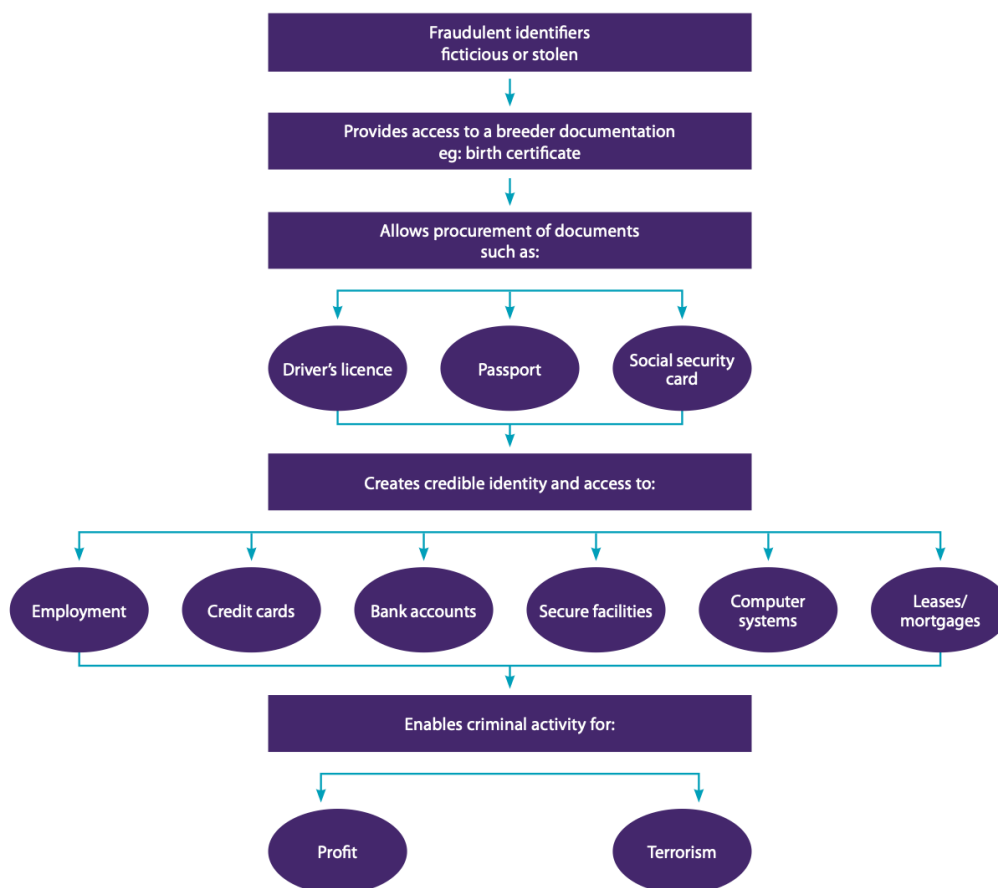
Investiciniai sukčiavimai – aukoms siūloma dalyvauti investicijų schemose, kurios yra labai pelningos, nors iš tikro jos yra bevertės arba labai greitai netenka savo vertės. Siūloma pirkti nekilnojamąjį turtą su nuolaida, kuomet siekiama privilioti investuotojus į neegzistuojančią nekilnojamojo turto pirkimo ir nuomos rinką. Taip pat, dar gerai žinomos „Ponzi“ schemos, kai sukčius sukuria „teisėtą“ investavimo schemą, kurios grąža paprastai vos viršija vidutinę (Button et al., 2009).

Kiti sukčiavimai, nepriskiriami konkrečiai kategorijai, ypač pagarsėję šiame dešimtmetyje, kuomet asmenys per pažinčių platformas apsimeta kitais asmenimis, prašo už juos mokėti, imti paskolas ir kita (Button et al., 2009).

Tapatybės sukčiavimas – „tapatybės vagys“ turi būti atskiriami nuo „tapatybės sukčiavimo“ nes tai nėra tą patį reiškiančios sąvokos. Daug labiau paplitę įvairūs tapatybės sukčiavimo tipai, kai neteisėtai panaudojami kito asmens tapatybės duomenys be jo žinios. Nacionalinė sukčiavimo institucija, sukčiavimą tapatybe apibrėžė kaip „asmens ar įmonės tapatybės informacijos naudojimas siekiant atidaryti sąskaitas, apgaule gauti socialinio draudimo išmokas (fizinį asmenų atveju),

kreiptis dėl kredito ir (arba) gauti prekių bei paslaugų.“ (Button et al., 2009). Toliau išvardijami atvejai, kada yra susiduriama su tapatybių sukčiavimu, o šio nusikaltimo kelias pateikiamas 1 pav.

- a) Pamestos arba pavogtos kortelės/dokumentai – vartotojui pametus savo kortelę ar dokumentus, sukčiai jais pasinaudoja tam, kad gautų reikalingas prekes ar paslaugas (Button et al., 2009).
- b) Pavogti kortelės duomenis jos neturint – vartotojui įrašius savo kortelės duomenis pavojuose internetiniuose, sukčių sukurtuose, puslapiuose, sukčius iš aukos banko sąskaitos duomenų gauna pakankamai informacijos, kad galėtų atlikti mokėjimus jo vardu (Button et al., 2009).
- c) Padirbtos kortelės/dokumentai – sukčiai padirbinėja esamas korteles ar vartotojų dokumentus ir naudoja juos aukos vardu įsigyjant prekes ar paslaugas (Button et al., 2009).
- d) Sąskaitos perėmimas – sukčius perima teisėto kliento sąskaitą ir naudoja joje esančias lėšas.
- e) Naujų sąskaitų kūrimas, paskolų ir kreditų ėmimas – sukčiai naudoja aukų asmens duomenis, kad sukurtų naujas banko sąskaitas ir prašytų kredito jų vardu (Button et al., 2009).
- f) Tapatybės vagystė – sukčius visam laikui perima asmens tapatybę (Button et al., 2009).



Šaltinis: Button et al. (2009). *Fraud typologies and victims of fraud*, p.12.

1 pav. Tapatybės sukčiavimas ir jo procesas

1.3. Klientų pažinimo procesas ir jo etapai

Remiantis Lietuvos Respublikos pinigų plovimo ir teroristų finansavimo prevencijos įstatymu, visos finansinės institucijos, susijusios su trečiųjų asmenų lėšų fasilitavimu ir apdorojimu, privalo užtikrinti lėšų teisėtumą bei savo klientų tapatybę dėl ko gali būti sumažinamas sukčiavimo atvejų skaičius (E-seimas, 1997). „Pažink savo klientą“ (angl. *Know your customer* (KYC)) yra tam tikras standartas, arba kitais žodžiais tariant, taisyklių rinkinys, kuriuo remiasi visų tipų finansų įstaigos, siekdamos nustatyti savo naujų ar esamų klientų profilius, t.y. tapatybes bei su ja susijusią riziką (Rajput 2013).

Profesorius Venkatesh U. Rajput (2013) išskiria du pagrindinius KYC proceso komponentus, kurie yra būtini taisyklingam šio minėto proceso atlikimui, tačiau kiti, naujesni moksliniai straipsniai, išskiria tris pagrindinius KYC proceso komponentus (Aziz ir Md Daud, 2022). Bendrai analizuojant pateiktą informaciją, komponentų etapai yra lygiai tokie patys, tačiau skirtingose šaltiniuose išskirstyti į skirtingas klases. Profesoriaus Rajput (2013) teigimu, KYC procesą sudaro: klientų patikra (angl. *Customer due diligence* (CDD)) ir sustiprinta klientų patikra (angl. *Enhanced due diligence* (EDD)), o mokslininkai Z. Aziz ir S. Md Daud (2022) paminė dar ir klientų identifikavimo procesą (angl. *Customer identification procedures*), kuris prieš tai minėtas profesorius priskyrė prie CDD.

Klientų identifikavimas

Tai pats pirmasis KYC žingsnis, kurio metu, kaip ir galima suprasti iš etapo pavadinimo, yra nustatoma klientų tapatybė. Šiame etape, klientai privalo pateikti savo, arba įmonės atstovo, vyriausybės išduotus asmens tapatybės dokumentus, tokius kaip pasą, asmens tapatybės kortelę, kai kuriuose kitų šalių finansų įstaigose, priklausomai nuo įstatymo – gali būti priimtas ir laikinas leidimas gyventi šalyje, vairuotojų ar net studentų pažymėjimai. Pateiktas dokumentas leidžia verifikuoti klientų asmeninę informaciją, tokią kaip vardas, pavardė, gimimo data ir kartais asmens kodas bei kitą (Aziz ir Md Daud, 2022). Taip pat aktualus yra ir klientų registruotos bei esamos registruotos gyvenamosios vietos adresas, kurio patvirtinimui gali būti prašomi papildomai dokumentai, tokie kaip sąskaitos už komunalines paslaugas, notarizuotos nuomos sutartys, komercinių bankų sąskaitos išrašai ir kita (Rajput, 2013). Kliento tapatybės nustatymo metu, finansų įstaigos taip pat reikalauja papildomos informacijos, susijusios su klientų ar atstovų pajamomis, užimtumu ir kartais net tai pagrindžiančių dokumentų (Schlatt et al., 2022). Šis etapas yra pirminis KYC etapas, net ir galima išskirti kaip vieną svarbiausių etapų, kurio metu užkertamas kelias sukčiavimui, t.y. tapatybės vagystėms bei su tuo susijusiam pinigų plovimui. Gavusios tikslią

informaciją iš klientų, finansų įstaigos gali užtikrinti, kad vykdo veiklą su teisėtais asmenimis ar įmonėmis, bei laikosi teisės aktų nustatytų reikalavimų (Kurum, 2023).

Klientų patikra

Klientų patikra (angl. *customer due diligence* (CDD)) yra labai svarbus KYC proceso komponentas, kuris tuo pačiu yra neatsiejama sukčiavimo nustatymo dalis, kuri implementuojama visos finansinės paslaugas teikiančios įstaigos. Shust ir Distov (2020) teigimu, CDD yra procesas, kurio metu yra surenkama informacija apie klientą, stengiantis nustatyti pastarojo tapatybę, tuo pačiu bandant mitiguoti galimą susijusią riziką bei atvejus su sukčiavimo veikla. Šis procesas nustato tiek individualių, tiek juridinių asmenų tapatybes bei jų rizikos profilius, tam kad būtų galima užmegzti dalykinius santykius (Shust ir Distov, 2020).

- Supaprastinta klientų patikra (angl. *Simplified customer due diligence* (SCDD)).

Ši patikra paprastai naudojama, kai kliento rizika, sąskaitos atidarymo metu, yra žema ir nekelia papildomos rizikos, kad klientas gali dalyvauti sukčiavimo, pinigų plovimo ar teroristų finansavimo veiklose (Chitimira ir Munedzi, 2023). Kadangi visos finansų įstaigos privalo atlikti bent jau minimalią klientų patikrą, jos metu, kaip ir galima suprasti, yra atliekamas žemiausias galimas patikrinimo lygis. Mažos rizikos klientams priskiriami dirbantys fiziniai asmenys ar įmonės su palyginus mažomis mėnesinėmis pinigų operacijomis ir nesusijusiomis rizikingomis šalimis. Šios patikros metu, yra būtina gauti tapatybės dokumentus, kas plačiau aprašyta „*Klientų identifikavimas*“ skyrelyje, nustatyti verslo santykių pobūdį (Njotini, 2009). Jeigu klientas visos patikros metu, lieka kaip žemos arba vidutinės rizikos (rizikos klasifikacijos balai apibūdinami kiekvienos įmonės vidinėje politikoje), FĮ neprivalo rinkti kliento ar atstovo lėšas pagrindžiančius dokumentus (Kurum, 2023).

- Sustiprinta klientų patikra (angl. *Enhanced due diligence* (EDD)).

Jei klientų rizikos nustatymo metu, rizika apibūdinama kaip aukšta, finansų įstaigos privalo imtis papildomų priemonių sąskaitos atidarymo ar jos patikrinimo metu, nes tokie klientai turi didesnę galimą riziką prisidėti prie sukčiavimo procesų (Chitimira ir Munedzi, 2023). Finansinių veiksmų darbo grupė (angl. *Financial Action Task Force*) numato, kad EDD priemonės turėtų būti naudojamos konkrečiose situacijose ir (arba) sandoriuose, kuriuose dalyvauja didelės rizikos klientai (Njotini, 2009). Šios patikros metu, yra atliekamas ypač detalus ir išsamus kliento patikrinimas, kuris apima tokias priemones, kaip išsamią biografijos analizę, konfidencialios informacijos rinkimą su tai pagrindžiančiais dokumentais, tokiais kaip sutartys su verslo partneriais, algalapiai, banko sąskaitų išrašai, įdarbinimo kontraktai ir t.t., kad klientų veikla būtų ne tik nustatyta, bet dar ir pagrįsta (Rajput, 2013). Išnagrinėjus pateiktus sąvokų (t.y. CDD ir EDD) apibūdinimus bei jiems taikomus metodus, apačioje pateikiama 2 lentelė, sutrumpintam procesų apibūdinimui.

CDD ir EDD palyginimas

Klientų patikra (CDD)	Sustiprinta klientų patikra (EDD)
Paprasta klientų patikra žemos ir vidutinės rizikos klientams	Aukštos rizikos, sudėtingų klientų patikra
Įprasta standartizuota procedūra	Papildomų dokumentų ir informacijos rinkimas
Patikros metu suteikiama įprasto lygio apsauga	Patikros metu suteikiama aukšto lygio apsauga

Šaltinis: sudaryta autoriaus pagal Chitimira ir Munedzi (2023); Njotini (2009).

Nuolatinis patikrinimas

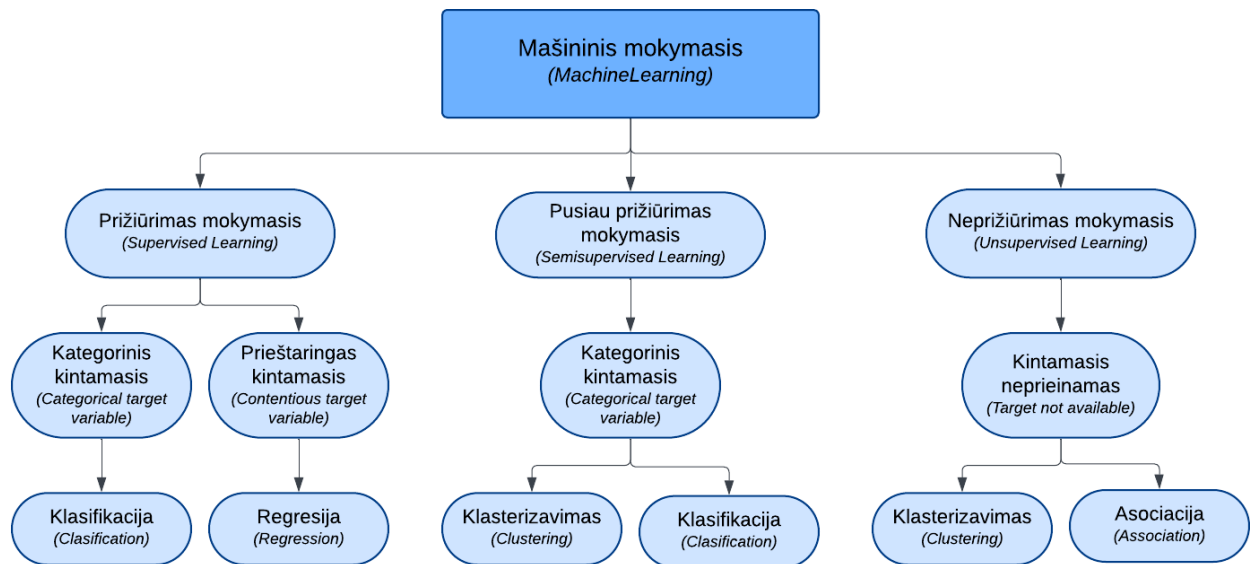
Finansų įstaigoms, nepriklausomai nuo kliento rizikos, privalo stebėti visą jų finansinį kelią jau po sąskaitos(-ų) atidarymo proceso, nes net ir mažiausios rizikos klientai gali dalyvauti sukčiavimo procese. Toks klientų stebėjimas jau užmezgus dalykinius santykius su klientais yra vadinamas nuolatinio klientų patikrinimu (angl. *Ongoing monitoring*). Šio proceso metu užtikrinama, kad klientų atliekamos transakcijos atitiktų sąskaitos atidarymo metu nustatytą klientų rizikos profilį ir nurodytą lėšų šaltinį. Nuolatinė klientų patikra suteikia bankams ir susijusioms finansų įstaigoms atnaujintus svarbių konfidencialių duomenų ir informacijos iš savo finansinių klientų įrašus, kurie naudojami aptikti ir kovoti su sukčiavimu daugelyje šalių ir jurisdikcijų (Chitimira ir Munedzi, 2023). Taip pat, autoriaus teigimu, tokių finansų įstaigų darbuotojai turi turėti atitinkamą patirtį ir nuolat vykstančius mokymus, kad būti galima laiku ir teisingai stebėti, vertinti bei aptikti nusikaltimą ir su tuo susijusią riziką (Njotini, 2009).

1.4. Mašininio mokymosi algoritmai ir metodai sukčiavimo aptikimui

Finansinio sukčiavimo, vykstančio per kreditines korteles, aptikimo tikslas – remiantis istoriniais duomenimis nuspręsti, ar operacija yra apgaulinga (Carcillo et al., 2021). Kiekvienos transakcijos ar per kredito kortelę įvykdytos operacijos tikrinimas, šiais laikais jau senai nebenaudojamas procesas – operacijų suma nedidelė, o laiko užtunka daug (Niu et al., 2019). Pačiam procesui įtakos turi ir klientų elgsena, pavyzdžiui prieš šventes pasikeičia pinigų leidimo įpročiai, o ir pačių sukčių metodų, ypač tų, kuriuos jie naudoja prisitaikydami prie sukčiavimo nustatymo metodų, vis daugėja (Carcillo et al., 2021). Dabar žinoma, kad mašininio mokymosi metodai yra veiksmingas būdas spręsti su sukčiavimo procesu susijusias problemas (Abdallah Z. et al., 2016).

Įprastą sukčiavimo aptikimo sistemą sudaro keli valdymo lygiai, kurių kiekvienas gali būti automatizuotas arba prižiūrimas žmonių (Carcillo et al., 2021). Dalis automatizuotų metodų apima

mašininio mokymosi algoritmus, kurie sukuria nuspėjamuosius modelius, pagrįstus jau pažymėtomis „sukčiavimas“ operacijomis. Per pastarąjį dešimtmetį intensyvūs mašininio mokymosi tyrimai, skirti aptikti sukčiavimą finansų įstaigose, paskatino sukurti prižiūrimus (angl. *Supervised Learning*), neprižiūrimus (angl. *Unsupervised Learning*) ir pusiau prižiūrimus (angl. *Semi-Supervised Learning*) mokymosi metodus (Sethi ir Gera, 2014), kurie pateikti 2 pav.



Šaltinis: sudaryta autoriaus remiantis Sethi ir Gera, 2014.

2 pav. Mašininio mokymosi metodai

1.4.1. Prižiūrimi mokymosi metodai

Tai yra metodai, kurie gali būti naudojami tik tada, kai jau įvykusiuose mokėjimuose ar operacijose buvo nustatyta, kad tai yra asmenys, sąskaitos, transakcijos ar kita, susiję su konkrečia ieškoma problema (šiuo atveju – sukčiavimu). Kitaip tariant, metodas priklauso nuo istorinių duomenų, kuriuose yra „etiketė“ (angl. *label*), kuri pažymi rezultatą arba klasę (Carcillo et al., 2021). Sukčiavimo aptikimo problemų atveju, etiketė yra arba tikra (operaciją atliko pats klientas), dažniausiai žymima – „0“, dar laikoma įprasta operacija, arba apgaulinga (operaciją atliko sukčius, be savininko žinios) ir dažniausiai žymima – „1“ bei apibūdinama kaip neįprasta operacija (Niu et al., 2019). Etiketę dažniausiai sužinoma iš kliento skundų, kuriuos jis pats pateikė ir kreipėsi į finansų įstaigą, arba bendrovė tai nustatė pati iš nustatytos įtartinos ar pasikeitusios kliento veiklos. Taikant prižiūrimus mokymosi metodus, kada, kaip ir minėta, naudojami jau įvykę, pažymėti sandoriai, pagal kuriuos bandoma išmokyti modelį atpažinti sukčiavimą ateityje (Carcillo et al., 2021). Toliau aprašoma keletas pagrindinių prižiūrimo mokymosi algoritmų, jų veikimas bei naudojimo galimybės.

„Atsitikiniai miškai“ (angl. *Random Forest (RF)*) – vienas seniausių ir populiariausių žinomų algoritmų. 2001 metais L. Breimano pasiūlytas RF algoritmas pasiteisino kaip bendrosios paskirties klasifikavimo ir regresijos metodas. Algoritmas, sujungia kelis atsitiktinių imčių sprendimų medžius ir apibendrina jų prognozes vidurkiu. Algoritmas yra labai tinkamas, kai nepriklausomų kintamųjų skaičius yra daug didesnis nei priklausomų kintamųjų skaičius. Be to, jis yra pakankamai universalus, nes pritaikomas didelės apimties problemoms, susijusioms ir su klasifikacijos, ir su regresijos uždutims, spręsti. Algoritmas dažnai naudojamas sukčiavimo nustatymui, medicinos diagnostikai (Niu et al., 2019).

„Paiškinamasis atsitiktinis miškas“ (angl. *Explainable Random Forest (XRF)*) yra RF algoritmo atmaina, kuri leidžia paveikti modelio priklausomybę specifinėmis mokymosi duomenų ypatybėmis, dėl kurių pagerėja modelio gerumas, pagerinamas jo tikslumas. Šis algoritmas puikiai tinka tada, kai norima, kad modelis kuo tiksliau būtų apmokomas ir tinkamai atspėtų ieškomą problemą, ypač naudojamas finansų ir sveikatos priežiūros srityse, kaip ir RF algoritmas. Šis metodas pagrįstas tuo, kad leidžia vartotojui nurodyti funkcijos parametrų reikšmes, kuris koreguoja modelį taip, kad būtų teikiama pirmenybė pagal šias reikšmes. Metodas remiasi dviem pagrindiniais veiksmiais, kuriuose kuria sprendimų medžius bei optimizuoja objektyvią funkciją (Amit ir Gur, 2024).

„Gradientinis stiprinimo mechanizmas“ (angl. *Gradient Boosted machine (GBM)*). Šis algoritmas turi dar kitokį pavadinimą, kartais jis dar vadinamas „gradientiniu stiprinimo (regresijos) medžiu“. Šis algoritmas buvo sukurtas siekiant pagerinti klasifikavimo ir regresijos, t. y. vieno klasifikatoriaus, kurio klasifikavimas ir regresija vyksta vienu metu, našumą. GBM geba iš kelių to paties tipo silpnų klasifikatorių, sudaryti prognozavimo modelį. Jis remiasi tokiu principu, kad nuosekliai augina „medžius“, kurių vėlesni „medžių“ rezultatai remiasi ankstesnių „medžių“ rezultatais. Algoritmą siūloma naudoti tada, kai turimos užduoties atlikimui yra labai svarbus didelis tikslumas bei turimi duomenys yra kompleksiniai ir sudėtingi. Algoritmas dažnai pasitarnauja anomalijų aptikime, nes pasikliauja kintamųjų svarba ir sąveika (Tama ir Rhee, 2017).

„Ekstremalus gradiento padidinimas“ (angl. *Extreme Gradient Boosting (XGBoost)*) yra algoritmas, kuris paremtas RF technika sprendžiant regresijos, klasifikavimo problemas, tačiau XGBoost įtraukia dar ir reitingavimą, kurio pagalba sukuriamas naujas „silpnasis besimokantysis“, kad kompensuotų esamų „silpnų besimokančiųjų“ trūkumus. XGB unikalumas yra tas, kad jis naudoja labiau įprastą modelio formalizavimą, kad būtų galima kontroliuoti perteklinį pritaikymą ir pasiekti geresnį našumą. Ypač pritaikomas elektronines prekybos, finansų srityse, bei puikiai tinkamas sukčiavimo nustatymui. Gradiento didinimas priklauso nuo regresijos medžių, kur optimizavimo žingsnis sumažina vidutinę kvadratinę paklaidą, o dvejetaini klasifikacijai

naudojamas standartinis „log“ nuostolis. Sujungus nuostolių funkciją su reguliavimo terminu, kuris kontroliuoja sudėtingumą ir mažina persimokymo riziką, gaunama objektyvi funkcija. Algoritmą siūloma naudoti tada, kai turimi duomenys yra didelio masto bei reikia juos greitai ir tiksliai apdoroti, tinka tvarkant duomenis, kuriuose yra daug trūkstančių reikšmių (Niu et al., 2019).

„Apibendrintas linijinis modelis“ (angl. *Generalized Linear Model* (GLM)) yra algoritmas, kuris puikiai tinka su sukčiavimu susijusioms transakcijoms nustatyti ir pasižymi dideliu tikslumu. Šis modelis dažniausiai naudojamas aiškinant ryšį tarp atsako kintamojo ir regresorių (lydinčių veiksmų) tuo atveju, kai pažeidžiamos klasikinio tiesinio modelio prielaidos. Algoritmas padeda suprasti tiesinius ryšius, kai gaunamas rezultatas yra dvinaris arba tęstinis. Priešingai nei prieš tai analizuoti algoritmai, jis gerai veikia duomenų rinkiniuose, kurie yra maži arba vidutiniai. Modelis gerai pritaikomas, kai reikia tirti prognozavimą, rizikos modeliavimą (Čampulova et al., 2019).

1.4.2. Neprižiūrimi mokymosi metodai

Neprižiūrimi mokymosi metodams, priešingai nei prižiūrimiems, nereikia žinių apie nustatytas operacijų etiketes. Šie metodai labiau orientuojasi į operacijų duomenų pasiskirstymą bei remiasi prielaida, kad sandorių pasiskirstymo nuokrypiai ir yra anomalija (šiuo atveju – sukčiavimas) (Niu et al., 2019). Pagal anomalijas, algoritmai gali aptikti visiškai kitokius sukčiavimo tipus, nes jie visiškai nesivadovauja nustatytais praeityje sandoriais. Verta paminėti, kad jų naudojimas taip pat apima klasterizacijos ir glaudinimo algoritmus (Chandola et al., 2009). Klasterizavimas leidžia nustatyti atskirus duomenų paskirstymus, kuriems turėtų būti naudojami skirtingi nuspėjamieji modeliai, o glaudinimas sumažina mokamosios problemos matmenis, tad abu algoritmai linkę pagerinti prižiūrimų metodų veikimą (Carcillo et al., 2021). Toliau aprašoma keletas pagrindinių metodų bei jų naudojimo galimybės:

„Vienos klasės palaikymo vektorinė mašina“ (angl. *One-Class Support Vector Machine* (OCSVM)) algoritmas pasiūlytas 2001 metais, kurio tikslas buvo identifikuoti anomaliją neprižiūrimais metodais, kuriems nereikia pažymėtų duomenų. Algoritmas išmoksta ribą, kad apimtų įprastus duomenų egzempliorius naudodamas mokymo rinkinį, o tada, naudodamas bandomąjį egzempliorių, sureguliuoja save taip, kad nustatytų anomalijas, kurios nepatenka į išmoktą sritį (Niu et al., 2019). Autorių teigimu, algoritmas turėtų geriausiai veikti, kai yra naudojamas mažoms duomenų bazėms, kuriose pateikiami duomenys yra struktūrizuoti (Smolen ir Benova, 2023).

„Automatinis kodavimo algoritmas“ (angl. *Auto-Encoder* (AE)) išmoksta pažymėti ir susieti duomenis nuo įvesties iki išvesties per kodavimo ir dekodavimo fazių porą. Kodavimo įrenginys susieja įvestį, dekoderis susieja išvestį, kad atkurtų įvestis. Paslėpti automatinio kodavimo sluoksniai

yra mažo matmens ir netiesiniai įvesties duomenų atvaizdavimai (Niu et al., 2019). Šis algoritmas turėtų būti labiau tinkamas didelio mastelio, kompleksinės struktūros duomenims, tačiau jo, autorių teigimu, naudojimui reikėtų didesnės galios, vien dėl to, kad duomenų kiekis turėtų būti didelis, norint gauti geriausius rezultatus (Smolen ir Benova, 2023).

„Izoliacinis miškas“ (angl. *Isolation Forest*) – anomalijas atskiria be jokio profiliavimo, kai kitiems algoritmams šis etapas būtų būtinas. IF sukuria mišką iš izoliacinių medžių, kur kiekvienas medis atitinka testą, kuris rekursyviai atskiria egzempliorius. Kadangi metodas daro prielaidą apie duomenis, kad anomalijų yra daug mažiau nei įprastų duomenų atvejų, anomalijos turėtų būti išskirtos anksčiau ir jų keliai medžiuose turėtų būti trumpesni. Anomalijos balas apskaičiuojamas pagal šių takų ilgį. Vienas naudojamiausių ir paprasčiausių algoritmų (Smolen ir Benova, 2023).

1.4.3. Pusiau prižiūrimi mašininio mokymosi metodai

Prieš tai minėti du metodai (prižiūrimi ir neprižiūrimi) vienas kitą gali papildyti: prižiūrimi metodai mokosi iš ankstesnio nesąžiningo elgesio, o neprižiūrimi metodai yra skirti aptikti naujus sukčiavimo tipus. Šie du, vienas kitą papildantys, metodai yra derinami pusiau prižiūrimuose metoduose (Carcillo et al., 2018). Jie dažnai naudojami, kai yra daug nepažymėtų duomenų, o pažymėtų – mažai. Jų tikslas – veikti geriau nei prižiūrimas modelis, kuriame naudojamas tik iš kelių galimų etikete pažymėtų duomenų, arba neprižiūrimas modelis, kuris neduoda tokios geros naudos sekti tik anomalijas (Carcillo et al., 2021).

Vienas tokių pavyzdžių, būtų Carcillo et al., 2021 metais aprašytas tyrimas, kuris remiasi nepriklausomo mokymosi metodo sujungimu su priklausomu mokymosi, gaunant kitokį hibridinį modelį, kai į modelio nepriklausomus kintamuosius yra įvedamas kintamasis gautas iš nepriklausomo mokymosi algoritmų. Autorių teigimu, prižiūrimo mokymosi metodai yra plačiai naudojami sukčiavimo nustatymui, nes daroma prielaida, kad modeliai gali būti išmokyti iš ankstesnių operacijų analizės. Tačiau, tobulėjant sukčiavimo procesams, užduotis tampa sudėtinga, kai reikia atsižvelgti į klientų elgsenos pokyčius ir sukčių galimybes, bei sugalvoti naujus sukčiavimo aptikimo modelius, tad neprižiūrimo modelio algoritmai gali labai pasitarnauti. Autoriai savo tiriamajame eksperimente pasiūlo sujungti prižiūrimus ir neprižiūrimus mokymosi algoritmus, kad būtų pagerintas sukčiavimo aptikimo tikslumas. Neprižiūrimo mokymosi pagalba apskaičiuojamas anomalijų balas, kuris įvedamas kaip papildomas kriterijus į prižiūrimą mokymąsi. Šiame tyrime, balai apskaičiuojami skirtingais detalumo lygiais, lyginami ir tikrinami naudojant skirtingus duomenų rinkinius. Balas apskaičiuojamas naudojant IF algoritmą. Eksperimentiniai rezultatai rodo, kad derinys yra efektyvus ir iš tikrųjų pagerina aptikimo tikslumą (Carcillo et al., 2021).

1.5. AutoML įrankiai

Mašininis mokymasis (angl. *Machine Learning* (ML)), atvėrė visiškai naujas galimybes dėl vis didėjančių duomenų kiekių, reikalingos didesnės apdorojimo galios. Sudėtingų mokymosi algoritmų kūrimas suteikė įrankius tuos didelius duomenų kiekius apdoroti greičiau bei teikti vis aiškesnes prognozes. Mašininis mokymasis yra labai svarbus daugelyje aspektų, tačiau jis ypač naudingas paskirstytajame mokymąsi (angl. *Distributed Learning*) bei automatiniam mašininiam mokyme (angl. *Auto Machine Learning* (AutoML)). Paskirstytą mokymąsi galima apibūdinti kaip didelio masto problemų sprendimą, kai tuo pačiu susiduriama su atminties ir apdorojimo apribojimais. AutoML įrankis gali įveikti atminties ir laiko problemas, nes prideda naujas mašinas ar procesorius tiesiogiai auginant duomenis (Peteiro-Barral ir Guijarro-Berdinas, 2012). AutoML gali parinkti algoritmus atitinkančius sprendžiamą problemą (Ferreira et al., 2020).

AutoML įrankį sudaro pradiniai penki modernūs atvirojo kodo AutoML įrankių rinkinys (Ferreira et al., 2020):

„Auto-Keras“ – AutoML Python biblioteka. „Auto-Keras“ pagrindinis dėmesys skiriamas giluminiam mokymuisi ir automatinei giluminio mokymosi modelių architektūros ir hiperparametrų paieškai. Bibliotekos tikslas – leisti domėnų ekspertams, turintiems ribotą ML pagrindą, lengvai kurti ir derinti giluminio mokymosi modelius (Jin et al., 2018).

„Auto-Sklearn“ įrankio metu, naudojama ML „*scikit-learn*“ biblioteka, kad atliktų algoritmų pasirinkimą ir hiperparametrų derinimą. „Auto-Sklearn“ yra *Python* modulis, naudojantis naujausius Bajeso optimizavimo pasiekimus, taip padidinant modelio efektyvumą. Į „Auto-Sklearn“ įeina penkiolika ML algoritmų, keturiolika funkcijų apdorojimo ir keturi duomenų išankstinio apdorojimo metodų (Ferreira et al., 2020).

„Auto-Weka“ – „*WEKA*“ ML įrankio modulis. „Auto-Weka“ modulis yra skirtas klasifikavimo užduočių kombinuoto algoritmo pasirinkimui spręsti. Naujausioje versijoje modelis jau palaiko regresijos užduotis (Kotthoff et al., 2017).

„H2O.ai“ yra platforma, kurioje yra integruotas ML ir nuspėjamoji analizė. Ji veikia su atmintyje esančiais duomenimis, naudodama paskirstytą ir keičiamo dydžio architektūrą, leidžiančią vartotojams kurti ML modelius didelių duomenų aplinkoje. H2O tikslas yra padėti vartotojams, kurie nėra ML ekspertai, sukurti didelio našumo bei tikslumo modelius. Ieškomų modelių skaičius priklauso nuo pasirenkamų vartotojo pasirinktų parametrų (pvz., laiko limitas, algoritmų išskyrimas, modelio kiekis). H2O AutoML taiko daugumą H2O algoritmų šeimų, tokių kaip GBM, GLM, XGBoost, RF bei taip pat ir gilaus mokymosi algoritmus (H2O.ai, 2024). Plačiau aprašomas 1.5.1 paragrafe.

„TransmogriAI“ yra „Scala“ programavimo kalba parašyta priemonė, kuri veikia „Apache Spark“ platformoje. Šis įrankis orientuotas į ML programų automatizavimą, leidžiantį vartotojams greitai gauti ML modelius su keliais rankiniais nustatymais. Naudodamasis šiuo įrankiu, vartotojas turi tik nurodyti duomenų rinkinį, schemą ir tikslinį (angl. *target*) stulpelį. „TransmogriAI“ naudoja šiuos parametrus, kad nustatytų tikslesnius atributų tipus (pvz., telefono numerius, pašto kodus) ir automatiškai atmeta įvesties funkcijas, kurios nepateikia nuspėjamosios reikšmės. Priklausomai nuo modeliujamos problemos tipo (t.y. klasifikacija ar regresija), TransmogriAI parengia algoritmų rinkinį ir iš anksto nustatytą hiperparametrų rinkinį. Išmokytų algoritmų sąrašas apima RF, GLM ir kt. (Ferreira et al., 2020). 3 pav. Pateikiami AutoML įrankių skirtumai.

	Auto-Keras	Auto-Sklean	Auto-Weka	H2O AutoML	TransmogriAI
Interface language	Python	Python	Python R	Python R Scala AWS Azure	Scala
Associated platforms	-	-	WEKA	Google Cloud Spark	Spark
Current version	pre-release	0.6.0	2.6.1	3.26.0.6	0.6.1
Graphical User Interface	-	-	✓	✓	-
Distributed mode	-	-	-	✓	✓

Šaltinis: Ferreira et al. (2020). *An Automated and Distributed Machine Learning Framework for Telecommunications Risk Management*, p. 103.

3 pav. AutoML technologijų charakteristikų santrauka

1.5.1. H2O AutoML

Mašininio mokymosi ekspertų paklausai gerokus viršijus pasiūlą, atsirado didelis poreikis šios problemos sprendimui. Darbuotojų daug, technologijų rinka sparčiai plečiasi, tikrųjų ekspertų mažai, o poreikis užduotis spręsti vis greičiau ir efektyviau kyla. Dėl šios priežasties, buvo sukurta automatizuota mašininio mokymosi programinė įranga, kurią gali naudotis ne tik mašininio mokymosi ekspertai. Pirmieji žingsniai siekiant supaprastinti mašininį mokymąsi buvo sukurti paprastas, vieningas sąsajas su įvairiais mašininio mokymosi algoritmais (pvz., H2O). H2O.ai yra programinės įrangos visiškai automatizuota programa, kurioje tyrėjams tereikia užkrauti savo turimus duomenis, pasirinkti užduoties tipą (t.y. klasifikavimo ar regresijos uždavinys), pasirinkti priklausomą kintamąjį ir laukti atsakymų. Sistema pati automatiškai parenka algoritmus, nepriklausomus kintamuosius, parametrus, kad pagal turimus duomenis būtų gautas kuo tikslesnis modelis tiriamai užduočiai (H2O.ai, 2024). Kadangi sistema yra mokama, o nemokama sistemos

versiją susiinstaliuoti užtrunka laiko ir šiek tiek programavimo žinių, failo paketas užima nemažai kompiuterio atminties, yra galima šį automatizuotą modelį naudoti ir virtualiose programavimo erdvėse, naudojant Python, R arba Scala programavimo kalbas (3 pav.) (Ferreira et al., 2020).

H2O AutoML veikimo principas pagrįstas naudojant automatizuotą mašininio mokymosi darbo eigą, kuri apima automatinį daugelio modelių mokymą ir derinimą per vartotojo nurodytą laiką. H2O siūlo daugybę modelio paaiškinimo metodų, taikomų AutoML objektams (modelių grupėms), taip pat atskiriems modeliams (pvz., lyderio modeliui). Paaiškinimai gali būti generuojami automatiškai, naudojant vieną funkcijos iškviatimą, suteikiant paprastą sąsają AutoML modeliams tyrinėti ir paaiškinti (H2O.ai, 2024).

Kaip ir minėta prieš tai, yra keletas parametrų, kuriuos yra privaloma nurodyti modelyje, kad jis galėtų pradėti automatizavimo procesą. Tokie parametrai yra (H2O.ai, 2024):

1. Duomenų parametrai:

- Y – priklausomas kintamasis (dar vadinamas „*target*“). Šiame, sukčiavimo atveju nustatymo automatizavime, priklausomas kintamasis yra jau nustatytas sukčiavimas iš istorinių duomenų;
- Mokymosi rėmas (angl. *training frame*) – nurodo duomenų kiekį, kuris bus naudojamas modelio apmokymui (likusi dalis – testavimui).

2. Sustojimo parametrai:

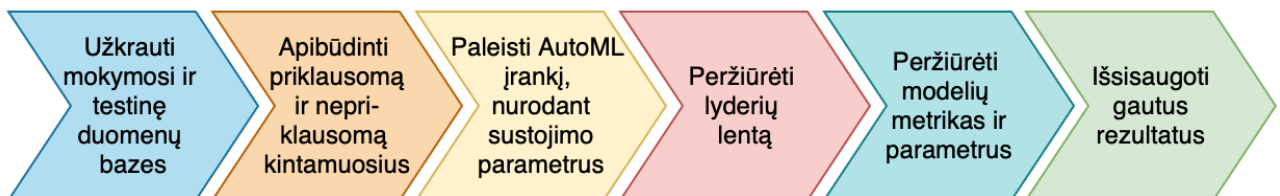
- Maksimalus modelių kiekis – nurodo maksimalų modelių skaičių, kuriuos reikia sukurti. Jeigu parametras nebus nurodytas, nebus apibrėžta, kiek modelių tikimasi gauti.
- Modelio veikimo maksimalus laikas – nurodo laiką, kiek ilgiausia modelis gali mokytis. Numatytoji modelio vertė yra be apribojimų, tačiau jeigu naudotojas nenustatytų maksimalaus laiko ir maksimalaus modelių skaičiaus, būtų laikoma, kad modelio treniravimo trukmė yra 1 valanda nuo jo paleidimo (H2O.ai, 2024).

Taip pat, į modelį galima įdėti daugelį kitų parametrų, kurie nėra privalomi. Keletas tokių parametrų apibūdinimų:

- Sėkla (angl. *seed*) – tai yra sveikasis skaičius. Pagal numatytuosius parametrus, sėklos reikšmė yra nulinė, tačiau ji pasitarnauja kuriant atkuriamuosius rezultatus. Nenustačius sėklos ir leidžiant kodą pakartotinai, galima gauti vis kitokius rezultatus, dėl galimo atsitiktinumo (H2O.ai, 2024). Dažniausiai naudojamas sėklos parametras yra „42“, tarsi reiškiantis savavališką pradinę vertę programavimo kalboje (Truong et al., 2019).

- Klasių automatinis balansavimas – šis parametras gali būti nurodomas kaip „True“ reikšmė, jei norima, kad klasifikacijos uždavinio sprendime, algoritmas sukurtų papildomai duomenų į mažąją klasę ir duomenys būtų subalansuoti (H2O.ai, 2024). Pavyzdžiui, jeigu su sukčiavimu susijusių klientų yra labai mažai, o visa duomenų imtis yra labai didelė, duomenų bazė gali būti netinkamai subalansuota ir duoti nekokybiškus rezultatus.
- Taip yra daugybė kitų parametru, kuriuos galima nurodyti, tokie kaip konkrečių algoritmų įtraukimas ar neįtraukimas į patį modelį, įvairios metrikos bei kita (H2O.ai, 2024).

Modelio treniravimo pabaigoje yra gaunama modelio lyderių lentelė su gautais modeliais. Joje pavaizduojami modeliai, gauti su skirtingais algoritmais, parametrais, modelių tikslumas. Taip pat, galima atlikti ir spėjimo funkciją ant testuojamos duomenų dalies, pagal kuriuos stebimas modelio gerumas nustatant ar šiuo atveju transakcija bus susijusi su sukčiavimu ar ne (H2O.ai, 2024). 4 pav. Pateikiamas H2O AutoML proceso kelias.



Šaltinis: sudaryta autoriaus, remiantis aprašais H2O AutoML įrankio etapais.

4 pav. H2O AutoML įrankio veikimas

1.6. Modelio tikslumo vertinimo kriterijai

Norint tinkamai suprasti gautų modelių rezultatus ir juos tinkamai interpretuoti, būtina nustatyti jų našumo metriką. Priklausomai nuo nagrinėjamos problemos tyrime, ar tai klasifikacijos problema, ar regresijos problema, yra gaunami kriterijai, kuriais galima apibūdinti modelio gerumą. Pagal turimus kintamuosius yra atrenkama ar nagrinėjama problema priklauso regresijos užduočiai, ar klasifikacijos. Sukčiavimo nustatymas, kada turimos etiketės, žyminčios sukčiavimą, yra priskiriamas klasifikavimo problemos sprendimui, nes tokio kintamojo turėjimas patenka į klasifikacijos užduoties sprendimą (Yousif ir Samawi, 2024). Toliau pateikiama 3 lentelė, kurioje pavaizduoti modelio našumo metrikai, priklausomai nuo sprendžiamos problemos tipo.

Modelių metrikos ir jų aprašymas

Nr.	Metrika	Aprašymas
Klasifikacija		
1.	Tikslumas (angl. <i>accuracy</i>)	Parodo teisingų prognozių dalį iš visų atliktų prognozavimų
2.	Tikslumas (angl. <i>precision</i>)	Matuoja spėjimų tikslumą, nustatant TP (šiuo atveju sukčiavimą)
3.	Atšaukimas	Modelio gebėjimas identifikuoti spėjamą klasę (šiuo atveju sukčiavimą)
4.	F1 balas	Tikslumo (angl. <i>precision</i>) ir atšaukimo vidurkis, subalansuojantis abi klases
5.	AUC	Kreivė, nurodanti TP ir FP rezultatų santykį
6.	Sumaišties matrica	Lentelė, parodanti TP, FP, TN ir FN kiekvienai tirtai klasei
7.	Logaritminis nuostolis	Matuoja prognozių neapibrėžtumą
Regresija		
1.	Vidutinė absoliutinė paklaida (MAE)	Absoliutinių paklaidų vidurkis tarp numatytų ir tikrųjų reikšmių
2.	Vidutinė kvadratinė klaida (MSE)	Vidurkis tarp kvadratinų skirtumų tarp prognozių ir faktinių verčių
3.	Vidutinė kvadratinė paklaida (RMSE)	Kvadratinė šaknis iš MSE
4.	R kvadratas (R^2)	Tikslinės reikšmės nuokrypio dalis
5.	Vidutinė absoliutinė procentinė paklaida (MAPE)	Vidutinė absoliutinė procentinė paklaida

TP – tikrosios teigiamos reikšmės; TN – tikrosios neigiamos reikšmės, FP – klaidingos teigiamos reikšmės, FN – klaidingos neigiamos reikšmės. Šaltinis: sudaryta autoriaus pagal Aragao et al. (2024), Ma et al. (2022), Khan et al. (2023) bei Regression Metrics (2024).

Kuo aukštesnės gautos reikšmės (išskyrus sumaišties matricą ir logaritminę nuostolį), sprendžianti klasifikavimo užduotį, tuo modelis geresnis (Yousif ir Samawi, 2024). Sprendžiant regresijos uždavinius – atvirkščiai, kuo vertės mažesnės (išskyrus R^2), tuo modelis laikomas geresniu (Regression Metrics, 2024). Vertinimo metrikos pasirinkimas priklauso nuo konkretaus problemos konteksto ir duomenų rinkinio charakteristikų. Jeigu turimo duomenų rinkinio klasių pasiskirstymas nėra subalansuotas, našumo metrikos, tokios kaip tikslumas (angl. *precision*), atšaukimas ir AUC vertės gali duoti daugiau informacijos nei kitas tikslumas (angl. *accuracy*) ar F1 balas (Walker, n.d.). AUC-PR reikšmė, sujungia ir tikslumą ir atšaukimą, tad gali būti informatyvesnė metrika. Abu kriterijai, ir AUC ir AUC-PR skirti klasifikavimo modelių prognozavimui, o ne slenkstinės klasės užduotims. Skirtumas yra tas, kad AUC atsižvelgia į tikrą teigiamas reikšmes ir klaidingai teigiamas reikšmes, o AUC-PR – į teigiamas nuspėjamas reikšmes ir tikrąsias teigiamas reikšmes (Czaron, 2024).

Tam, kad būtų galima stebėti ar modeliai veikia gerai, mašiniame mokymesi yra labai svarbu atkreipti dėmesį į modelių persimokymą. Kaip teigia Khoshgoftaar ir Allen (2001) modelių

persimokymas atsiranda tada, kai yra didelis metrikų skirtumas tarp mokymosi ir validacijos bei testavimo imčių. Ogbemi (2023) teigimu, 5 % skirtumas tarp minėtųjų imčių yra laikomas priimtiniu. Khoshgoftaar ir Allen (2001) teigimu, ši mašininio mokymosi problema galima būti koreguojamas keičiant į medžių skaičius. Barkved (2022) teigimu, metrikų rezultatai virš 70 % laikomi realiu ir geru.

1.7. Funkcijos parametrai modeliuose ir susiję darbai

Tam, kad būtų galima efektyviai nustatyti sukčiavimą ir padaryti jo aptikimo modelį, būtina atkreipti dėmesį į turimus duomenis, pasirinkti tinkamus metodus, bei panaudoti tinkamus nepriklausomus kintamuosius, kad modelio tikslumas ir preciziškumas būtų kuo tikslesnis. Literatūroje yra nemažai susijusių darbų, kuriuose autoriai naudoja vis skirtingus kintamuosius sukčiavimo aptikimui iš savo turimų duomenų. Kad kriterijai atitiktų realią turimą duomenų bazę, būtina išanalizuoti duomenų paketą bei pagal juos nustatyti, kuo tikslesnius kintamuosius. Sukčiavimo aptikimui dažniausiai naudojamos klientų transakcijos, jų datos, išsigryninimo dažnumas, bei kiti kriterijai, kurie indikuoja, kad klientų veikla gali būti susijusi su šiuo nusikaltimu (Abbassi et al., 2023). Toliau šiame poskyryje yra aprašomi susiję su sukčiavimu darbai, bei kokie nepriklausomi kintamieji buvo naudoti sukčiavimo aptikimui.

1 modelio funkcijos parametrai

Abbassi, Berkaoui, Elmendili, ir Gahi (2023) tyrė bei bandė sukurti pažangią sukčiavimo aptikimo architektūrą, kuri remiasi skaitmeninėmis operacijomis. Ši, autorių įvardijama architektūra, paremta neprižiūrimu mokymusi (angl. *unsupervised learning*) (t.y. duomenys nėra pažymėti etikete ar tai nusikaltimas ar ne). Duomenų rinkinį sudaro didelių duomenų analizės rinkiniai, kurių pagalba realiu laiku galima aptikti sukčiavimą, sutrumpinti mokymo laiką. Autoriai savo tyrime naudojo funkcijos kintamuosius pagrįstus operacijų dažnumu, ir sudėtingų charakteristikų išskyrimą, įskaitant operacijos datą, iš kurios išskiriant mėnesius, dienas, vietas ir pan. Autorių turima duomenų bazė taip pat turėjo duomenis, tokius kaip klientų bandymus prisijungti prie sistemų, kortelių duomenis, klientų vidinį praleidimo laiką jų sąskaitose, naudotus prietaisus bei prie interneto prisijungimo adresus (Abbassi et al., 2023).

Autorių modelis buvo apmokomas naudojant nepriklausomus kintamuosius, tokius kaip klientų identifikacijos numeris, sąskaitos tipas, prisijungimų prie sąskaitos skaičius, paskutinis prisijungimas, paskutinė transakcija, gavėjo sąskaita, transakcijos suma, prietaisas naudotas prisijungimui, laikotarpis kada buvo prisijungta prie prietaiso. Iš pateiktų duomenų matyti, kad autoriai savo modelyje daugiausia naudojo kriterijus, susijusius su prisijungimo prietaisais ir jo informacija, bei transakcijų sumomis, tačiau modelyje buvo naudota minimaliai datos parametrai.

Autorių tyrimas buvo atliktas su vienu iš neprižiūrimo mokymosi algoritmu – IF, kurio pagalba buvo nustatytos ir uždėtos sukčiavimo etiketės su nusikalstama veikla susijusiems klientams. Toliau autoriai atliko tyrimus su H2O programa, kuri kaip ir H2O AutoML įrankis parenka geriausius modelius parametrus, kad modelis būtų kuo tikslesnis (Abbassi et al., 2023).

II modelio funkcijos parametrai

Aditya Oza, 2018 metais taip pat atliko tyrimą susijusį su sukčiavimo aptikimu ir mašininio mokymosi panaudojimu, kuriame bandė atrinkti geriausią metodą tiksliai šio nusikaltimo aptikimui, kurio metu gaunamas mažas klaidingų rezultatų aptikimas. Autorei, atliktame tyrime, jau buvo žinomos transakcijų tipo etiketės, kurios susijusios su sukčiavimu, tad naudojo prižiūrimo mokymosi algoritmus, tokius kaip logistinė regresija (angl. *logistic regression*), linijinė atraminių vektorių mašina (angl. *linear support vector machine*) ir taip pat spindulio pagrindo funkcijos atraminių vektorių mašina (angl. *support vector machine with radial basis function*). Modelio apmokymui autorė naudojo tokius kintamuosius, kaip transakcijų tipas, transakcijų suma, siuntėjo bei gavėjo sąskaitos balansas prieš ir po transakcijos. Transakcijų tipas buvo pažymėtas kaip sukčiavimas arba nėra susijęs su sukčiavimo veikla. Iš šio modelio galima matyti, kad autorė visiškai nenagrinėjo datos, susijusios su transakcijomis, tačiau tyrė tik susijusius mokėjimų tipus, jų sumas bei likutį siuntėjų ir gavėjų sąskaitose. Iš transakcijų tipų duomenų, galima matyti, kad sukčiavimas buvo vykdomas per lėšų pavedimus ir lėšų išgryninimus (Oza, 2018).

III modelio funkcijos parametrai

Roy et al. (2018) tyrė sukčiavimą, atliekamą per kredito kortelės panaudojant ML algoritmus. Autoriai atliko savo tyrimą naudodami didelio našumo paskirstytą debesų kompiuterijos aplinką su tikslu, kad išspręstų įprastas sukčiavimo aptikimo problemas, tokias kaip klasių disbalansą ar mastelio keitimą. Taip pat pristato topologijų parametrų derinimo sistemą, skirtą kredito kortelių sukčiavimo aptikimui, kad finansinės institucijos galėtų sumažinti nuostolius užkertant kelią nesąžiningai veiklai. Savo tyrime autoriai naudojo duomenų rinkinį, kuris buvo gautas iš finansų įstaigos, užsiimančios mažmenine bankininkyste. Jame pateikiama informacija apie beveik 80 milijonų anoniminių kredito kortelių operacijų per aštuonių mėnesių laikotarpį. Pateikta informacija apima tiek operacijų, tiek sąskaitos informaciją, tokių kaip operacijos suma, prekybininko tipą, nuo kurio buvo inicijuota operacija, sąskaitos atidarymo datą ir kita. Autoriai taip pat naudojo prižiūrimo mokymosi metodus, nes jų turima duomenų bazė jau turėjo etiketes žyminčias nesąžiningas operacijas, kurių buvo žymiai mažiau nei sąžiningų, tad jų pasiūlytas modelis taip pat turėjo spręsti ir disbalanso problemą (Roy et al., 2018). Autorių naudoti funkcijos parametrai apėmė tokius kriterijus, kaip:

- Operacijų dažnis per mėnesį, kurie prideda informacijos apie paskyros išlaidų elgesį;

- Trūkstamų duomenų fiktyvūs kintamieji, kurie nebuvo išimti iš modelio, nes autorių teigimu tai yra svarbus veiksnys nustatant sukčiavimą kredito kortelėmis, nes perkant sukčiai nuslepia informaciją apie kortelių programas;
- Kintamieji, nurodantys, kada perkama pas konkretų prekybininką – tam tikri prekybininkai, pvz., degalinės ir restoranai, yra dažnos vietos, kuriose sukčiai išbando pavogtą kortelę prieš pirkdami didesnę pirkinių;
- Sąskaitos istorijos kintamieji, pvz., operacijų iš sąskaitos skaičius per 8-ių mėnesių laikotarpį, maksimali ir vidutinė paskyros autorizacijos suma;
- Kintamasis, nurodantis, ar tam tikro prekybininko sandorio autorizacijos suma yra didesnė nei 10 % prekybininko nesąžiningų sandorių vidurkio standartinio nuokrypio (Roy et al., 2018).

Modelio nustatymui autoriai naudojo visiškai kitokius algoritmus, negu prieš tai minėtieji 1.4. poskyryje, nes jie apėmė giliojo mokymosi (angl. *Deep Learning* (DL)) topologijas, tokia kaip dirbtinius neuroninius tinklus (angl. *Artificial Neural Networks*), pasikartojančius neuroninius tinklus (angl. *Recurrent Neural Networks*), ilgalaikę trumpalaikę atmintį (angl. *Long Short-term Memory*) ir ribotus pasikartojančius vienetų (angl. *Gated Recurrent Units*) (Roy et al., 2018).

IV modelio funkcijos parametrai

Mienye ir Swart, 2024 metais tyrė per kredito kortelės įvykusio sukčiavimo nustatymo modelius, pasitelkiant DL algoritmus. Autoriai savo tyrime jau taip pat turėjo pažymėtus kategorinius duomenis, žyminčius sukčiavimo atvejus, tačiau priešingai nei jau aprašyti modeliai, autoriai naudojo kitokius kintamuosius savo tyrime. Tyrimas buvo atliktas su Brazilijos banko kortelių duomenimis. Autoriai į modelį įtraukė kintamuosius, tokius kaip laiką nuo paskutinės transakcijos, sumą, transakcijos bei kortelės tipą, prekybininko kodą, lokacijos zip kodą, prieš tai įvykusių transakcijų sumą, kliento svarbą ir kortelės limitą. Autorių tirtą modelį iš viso sudaro 17 kintamųjų bei jis turėjo stipriai nesubalansuotą duomenų bazę. Autoriai, šiame darbe, priešingai nei jau aprašytuose modeliuose, bandė kurti modelį hibridą, kuriame sujungė du skirtingus algoritmus bei tokiu būdu pasiekė žymiai geresnius rezultatus, nei naudojant įprastus algoritmus. Šią savo technologiją išbandė ir ant iš Europos domenų gautos duomenų bazės, kurios gauti rezultatai taip pat įrodė jog algoritmų sujungimas modelyje duoda didesnę duomenų preciziškumą, specifikaciją bei tikslumą (Mienye ir Swart, 2024).

1.8. Sukčiavimo prevencijos teisinis aspektas

Sukčiavimo atvejų nustatymas bei jų automatizavimas yra labai svarbus kriterijus visoms reguliuojamoms finansų įstaigoms. Tokio pobūdžių nusikaltimų tik daugėja, tad finansinėms organizacijoms yra labai svarbu tinkamai įvertinti sukčiavimo atvejus, juos nustatyti kuo greičiau ir užkirsti šio nusikaltimo tolimesniems veiksams. Visos finansų įstaigos, įsteigtos Europos Sąjungos valstybėse narėse, privalo laikytis išleistų Europos bankininkystės institucijos gairių. Šias gaires skelbia Europos kovos su sukčiavimu tarnyba (angl. *The European anti-fraud office* (OLAF)). Ji yra pagrindinė įstaiga, kovojanti su sukčiavimu, korupcija ir kita, tokio pobūdžio neteisėta veikla Europos Sąjungoje (ES). Ši taryba atlieka tyrimus dėl galimų sukčiavimo, korupcijos ir finansinio nusižengimo atvejų ES institucijose. OLAF tyrimus atlieka glaudžiai bendradarbiaudama su atitinkamomis valstybių narių įstaigomis įsteigtomis toje šalyje (EU Monitor, 2024) (pavyzdžiui, Lietuvoje tai būtų Lietuvos bankas (Lietuvos Bankas, 2024)). OLAF taip pat remia ES institucijas, ypač Europos Komisiją, kuriant ir įgyvendinant teisės aktus bei politikas, susijusias su sukčiavimu bei kita neteisėta veikla. Taip pat buvo minėta, kad OLAF atlieka tyrimus, susijusius su sukčiavimu, tad tyrimai apima (EU Monitor, 2024):

- Vidaus tyrimus, kurie apibūdinami kaip administraciniai tyrimai ES institucijose ir įstaigose;
- Išorės tyrimai – tai administraciniai tyrimai už ES institucijų ir įstaigų ribų;
- Koordinavimo atvejai – OLAF prisideda prie nacionalinių institucijų ar kitų padalinių atliekamų tyrimų, palengvindama informacijos rinkimą ir keitimąsi informacija (EU Monitor, 2024).

Toliau kalbant apie teisės aktus ir tarybas, susijusias su sukčiavimo prevencija, Europos Parlamentas ir Taryba parengė sukčiavimo prevencijos programą, kuri apimta 2021-2027 metų laikotarpį. Pagrindinis reglamentas (ES) 2021/785 buvo paskelbtas 2021 m. gegužės 17 d. oficialiajame leidinyje L 172. Reglamente nustatomi kovos su sukčiavimu programos tikslai, minėtojo laikotarpio biudžetas, Sąjungos finansavimo formos ir tokio finansavimo teikimo taisyklės. Konkretūs iškelti kovos su sukčiavimu programos tikslai yra šie (Wahl, 2021):

- Kova ir prevencija su sukčiavimo procesu, korupcija ir bet kokios kitos neteisėtos veiklos, pažeidžiančios Sąjungos finansinius interesus;
- Paramos tęsimas įvairiems pranešimams apie pažeidimus, įskaitant sukčiavimą, susijusius su Sąjungos biudžeto pasidalijamojo valdymo lėšomis ir pasirengimo narystei pagalbos lėšomis (Wahl, 2021).

Apibendrinant išanalizuotą literatūros apžvalgą, galima teigti, kad sukčiavimas yra sunkus nusikaltimas, kuris skirstomas į daugybę rūšių, kategorijų bei tipų. Dažniausiai, sukčiavimo nustatymas atliekamas mašininio mokymosi metodų pagalba. Kad pilnai automatizuoti sukčiavimo atvejų aptikimą, galima naudoti H2O AutoML įrankį, kuris pats parenka tinkamiausius algoritmus, duomenų parametrus, bei jei norima – nepriklausomus kintamuosius. Atsižvelgiant į tai, ar nagrinėjamas klasifikacijos, ar regresijos problema, pagal tai parenkamos atitinkamos modelio vertinimo metrikos. Literatūros apžvalgoje, taip pat pateikiami ir aprašomi susiję panašūs tyrimai, tačiau visi jie naudoja kitokius nepriklausomus kintamuosius. Vieno autoriaus atliktame tyrime, buvo pastebėta, kad modelio metrikų našumas išauga, kai į modelį, atliekamą prižiūrimo mašininio mokymosi įrankiu, įvestas naujas nepriklausomas kintamasis, gautas neprižiūrimo mokymosi metodų pagalba. Sukčiavimo prevencija yra labai svarbus aspektas visoms finansų įstaigoms, tad jos privalo laikytis Europos Sąjungos nustatytų teisės aktų ir reikalavimų.

2. SUKČIAVIMO ATVEJŲ NUSTATYMO AUTOMATIZAVIMO, FINANSINIŲ PASLAUGŲ ĮMONĖSE, METODOLOGIJA

Automatizuotam sukčiavimo atvejų nustatymui atlikti pateikiama keturių etapų eiga. Tiriamojo darbo metodologijoje apibūdinama darbo paskirtis bei tipas, aprašomos našumo metrikų formulės, kuriomis remiantis bus įvertinamas modelio veikimas. Pateikiamas mašininio mokymosi įrankio ir jo programinio kodo šablono aprašymas.

2.1. Sukčiavimo atvejų nustatymo automatizavimo tyrimo pobūdis

Tyrimo tikslas ir paskirtis

Tam, kad būtų galima efektyviai kovoti su finansiniu sukčiavimu, šiame darbe atliekamas empirinis taikomasis tyrimas, kurio tikslas – pasiūlyti automatizuotą sukčiavimo atvejų aptikimo modelį, kurio pagalba sumažėtų finansinė rizika įvairioms institucijoms. Šis metodas galėtų būti panaudojamas įvairiuose finansų įstaigose ir būti pritaikomas praktiškai.

Tyrimo tipas

Šis tyrimas yra atliekamas tam, kad būtų pasiūlytas automatizuotas sukčiavimo atvejus nustatantis modelis, kuris pilnai gali veikti praktiškai. Šio tyrimo tipas yra taikomojo empirinio pobūdžio dėl to, nes modelis analizuoja bei jau mokosi iš įvykusių duomenų, tačiau tuo pačiu jis yra ir automatizuojamas ir pritaikomas praktiškai realioms sukčiavimo atvejų problemoms spręsti, tad modelio pobūdis apibūdinamas kaip taikomasis empirinis tyrimas.

Tyrimo objektas

Šio tyrimo objektas – sukčiavimo atvejai finansinių paslaugų įmonėje, juos nusakantys kintamieji, jų reikšmės ir svoriai.

2.2. Sukčiavimo atvejų nustatymo automatizavimo tyrimo metodas

Duomenų šaltinis

Šio tiriamojo darbo duomenų masyvą sudaro elektroninės pinigų įstaigos „N26“, įsteigtoje Vokietijoje, viešai prieinami duomenys, kurie buvo rasti internetinėje mokslinių duomenų bazėje „Kaggle“¹. Duomenyse pateikiama informacija, apie jau nustatytus sukčiavimo atvejus.

¹ <https://www.kaggle.com/datasets/zywald/n26-kyc-challenge>

Duomenų analizė

Analizuojami duomenys apima 2015-2018 metų periodą, juose yra 10 000 unikalių, nesusijusių su sukčiavimu, klientų sąskaitų, bei papildomos 300 sąskaitų, kuriose klientai buvo pažymėti kaip susiję su sukčiavimu (duomenų kiekis sumažėja, kai skirtingi duomenų failai sujungiami ir išsivalomi (žr. 3.2. poskyryje)). Duomenyse taip pat pateikiamos klientų transakcijos, kurios įvyko prieš tai minėtų klientų sąskaitose. Pateikiami ir klientų sąskaitos tipai, transakcijų pobūdis, tipas, prekybinkai, šalių kodai ir t.t. Duomenys yra geros kokybės, struktūrizuoti, pilnai tinkantys sukčiavimo atvejų nustatymo automatizavimo aptikimo panaudojimui. Duomenyse, priklausomas kintamas yra sukčiavimas, kuris žymimas „1“, jei sukčiavimas įvykęs, ir „0“, jei kliento sąskaita nebuvo pažymėta kaip susijusi su sukčiavimo procesu.

Metodas

Šis tyrimas yra apibūdinamas kaip empirinis taikomas metodas, skirtas automatizacijai, nes jame pateiktas tyrimas, atliekamas su automatizuotų mašininio mokymosi įrankiu – H2O AutoML, kurio pagalba galima išbandyti įvairius algoritmus bei jų parametrus, kad būtų parinktas optimalus, labiausiai duomenis ir sprendžiamą problemą atitinkantis metodas, geriausiai leidžiantis įvertinti sukčiavimo atvejams keliamą riziką ir ją efektyviai nustatyti. Šis pasirinktas priklausomo mašininio mokymosi metodas pats tikrina įvairius algoritmus (dalis jų aprašyti 1.4. poskyryje) ir sudėlioja parametrus taip, kad būtų pasiektas geriausias rezultatas.

Šiame darbe tyrimas buvo atliktas naudojant Python programavimo kalbą, kurios pagalba integruotas H2O AutoML įrankis, leidžiantis atlikti tiriamąjį darbą. Įrankio naudojamas ir jo kodo šablonas pateikiamas pačiame H2O.ai puslapyje (H2O.ai, 2024).

H2O AutoML veikimas

Šis metodas yra visiškai automatizuotas, sukurtas taip, kad vartotojas turėtų nurodyti kuo mažiau parametrų. Metodas reikalauja pateikti duomenų rinkinį ir identifikuotą priklausomą kintamąjį. Kadangi šiame tiriamajame darbe, buvo tirta nepriklausomų kintamųjų įtaką ir svoris modeliui, nepriklausomi parametrai buvo sudaryti paties autoriaus. Taip pat, kad būtų užtikrintas duomenų atsikartojimas, buvo parinkta sėkla (angl. *seed*) lygi 42, modelio treniravimo laikas iki 2 val., parinktas duomenų balansavimas ir skaičių kartotinis lygus 0, kadangi tam buvo skirta validacijos duomenų imtis.

Toliau 5 pav. pateikiamas H2O AutoML įrankio kodo pavyzdys Python programavimo kalba, kuris buvo naudotas šiame empiriniame taikomajame tyrime. Kodo viršuje aprašyta H2O AutoML kodo pradžia, kada į virtualią kodo rašymo direktoriją importuojamos H2O AutoML bibliotekos, paleidžiamas klasteris ir užkraunami tiriamojo darbo duomenų rinkiniai. Toliau identifikuojami nepriklausomi kintamieji, jei yra poreikis jie apibūdinami, jei ne – sistema nustato ir parenka

automatiškai, išskiriamas priklausomas kintamasis. Šis metodas, taip pat leidžia pasirinkti gaunamųjų modelių kiekį, sėklos parametą, laiką, kada modelio treniravimas turėtų sustoti. AutoML objektas apima pirmaujančiųjų sąrašą lyderių lentoje, kuriame reitinguojami visi modeliai, parengti pagal modelio našumą. Pagal numatytuosius nustatymus jie reitinguojami pagal kryžminio patvirtinimo našumą, nebent pirmaujančiųjų sąrašo rėmelio argumente pateikiamas išlaikymo rėmelis balams ir modeliams reitinguoti (H2O.ai, 2024). Pagal aprašytą kodą, sukurtas automatizuotas modelis, leidžiantis nustatyti sukčiavimo atvejus, pateikiamas 1 PRIEDAS.

```
import h2o
from h2o.automl import H2OAutoML

# Start the H2O cluster (locally)
h2o.init()

# Import a sample binary outcome train/test set into H2O
train = h2o.import_file("https://s3.amazonaws.com/h2o-public-test-data/smldata/higgs/higgs_
test = h2o.import_file("https://s3.amazonaws.com/h2o-public-test-data/smldata/higgs/higgs_t

# Identify predictors and response
x = train.columns
y = "response"
x.remove(y)

# For binary classification, response should be a factor
train[y] = train[y].asfactor()
test[y] = test[y].asfactor()

# Run AutoML for 20 base models
aml = H2OAutoML(max_models=20, seed=1)
aml.train(x=x, y=y, training_frame=train)

# View the AutoML Leaderboard
lb = aml.leaderboard
lb.head(rows=lb.nrows) # Print all rows instead of default (10 rows)
```

Šaltinis: H2O.ai (2024). H2O AutoML: Automatic Machine Learning.

5 pav. H2O AutoML kodo pavyzdys Python programavimo kalba

2.3. Sukčiavimo atvejų nustatymo automatizavimo metodo našumo metrikos

Kadangi šiame darbe yra tiriamas klasifikavimo uždavinys, jo modelių našumo metriką sudaro šiame poskyryje aprašyti pagrindiniai matmenys, apibūdinantys modelio gerumą bei jo našumą. Kuo aukštesnės šių modelių gautos reikšmės, tuo modelis geresnis (išskyrus logaritminį nuostolį (angl. *log loss*)).

Pirmasis jų yra vadinamas „tikslumu“ (angl. *accuracy*). Tai yra rodiklis, kuriuo matuojama teisingų modelio prognozių dalis iš visų prognozių skaičiaus. Jis apskaičiuojamas kaip teisingų teigiamų ir teisingų neigiamų reikšmių prognozių suma kintamajam „a“, padalyta iš bendro atvejų skaičiaus, kaip parodyta 1je. Nors tikslumas yra naudingas bendras matas, jis gali būti klaidinantis nesubalansuotuose duomenų rinkiniuose, kuriuose viena klasė gerokai viršija kitą (Ma et al., 2022).

$$Accuracy(a) = \frac{TP(a) + TN(a)}{TP(a) + TN(a) + FP(a) + FN(a)}, \quad (1)$$

čia $Accuracy(a)$ – modelio tikslumą rodantis parametras kintamajam „a“;

$TP(a)$ – tikrosios teigiamos reikšmės iš kintamojo „a“;

$TN(a)$ – tikrosios neigiamos reikšmės iš kintamojo „a“.

$FP(a)$ – klaidingos teigiamos reikšmės iš kintamojo „a“.

$FN(a)$ – klaidingos neigiamos reikšmės iš kintamojo „a“.

Dar vienas svarbus parametras yra kitas modelio „tikslumas“ (angl. *precision*), kuris yra metrika, matuojanti tikrų teigiamų prognozių dalį iš visų modelio pateiktų teigiamų prognozių, nurodant teigiamų prognozių tikslumą. Parametras parodo kiek teigiamų prognozių iš tikrųjų yra teigiamų tam tikrai klasei (t.y. kintamajam), kuri pavadinta „a“. Šis parametras apskaičiuojamas tikrąsias teigiamas reikšmes padalijus iš tikrųjų teigiamų reikšmių ir klaidingų teigiamų sumos, kaip parodyta 2je (Aragao, 2024).

$$Precision(a) = \frac{TP(a)}{TP(a) + FP(a)}, \quad (2)$$

čia $Precision(a)$ – tikslumas kintamajam „a“;

$TP(a)$ – tikrosios teigiamos reikšmės iš kintamojo „a“;

$FP(a)$ – klaidingos teigiamos reikšmės iš kintamojo „a“.

Kitas rodiklis, kuris labai svarbus modelio gerumui, yra „atšaukimas“ (angl. *recall*), kuris matuoja faktinių teigiamų atvejų, kuriuos teisingai nustatė modelis, dalį, atspindinčią jo gebėjimą rasti visus susijusius atvejus. Kitaip sakant, parametras indukuoja, kiek teigiamų rezultatų buvo teisingai prognozuota iš visų faktinių teigiamų tam tikros a klasės (t.y. kintamojo). Jis apskaičiuojamas kaip tikrų teigiamų reikšmių skaičius, padalytas iš tikrųjų teigiamų ir klaidingai neigiamų reikšmių sumos, kaip parodyta 3je (Aragao, 2024).

$$Recall(a) = \frac{TP(a)}{TP(a) + FN(a)}, \quad (3)$$

čia $Recall(a)$ – atšaukimo parametras kintamajam „a”;

$TP(a)$ – tikrosios teigiamos reikšmės iš kintamojo „a”;

$FN(a)$ – klaidingos neigiamos reikšmės iš kintamojo „a”.

Parametras, kuris susideda iš tikslumo ir atšaukimo rodiklio yra F1 balas, kuris yra tikslesnė ir informatyvesnė metrika. Parametras suteikia vieną metriką, subalansuojančią teigiamų prognozių tikslumą ir modelio gebėjimą identifikuoti visus susijusius atvejus, ypač naudingus nesubalansuotuose duomenų rinkiniuose. Jį sudaro tam tikros a klasės (t.y. kintamojo) tikslumo ir atšaukimo harmoninis vidurkis, kaip parodyta 4je (Aragao, 2024).

$$F_1 \text{ score}(a) = 2 * \frac{Precision(a) * Recall(a)}{Precision(a) + Recall(a)}, \quad (4)$$

čia $F_1 \text{ score}(a)$ – F1 parametras kintamajam „a”;

$Precision(a)$ – tikslumas kintamajam „a”;

$Recall(a)$ – atšaukimo parametras kintamajam „a”.

$TPR(a)$ – teisingos neigiamos reikšmės rodiklis kintamojo „a” ribose.

Taip pat dar vienas kriterijus labai svarbus modelio gerumui vertinti yra „AUC“ (angl. *Area Under the Receiver Operating Characteristic*), kuris yra apibūdinamas kaip plotas po kreive. Tai yra vidutinis modelio našumo matavimas kintamajam „a”, atliktas tam tikru slenksčio lygiu, o jo reikšmė svyruoja nuo 0 iki 1. Vertė 0 reiškia, kad visos modelio prognozės yra neteisingos ir atvirkščiai, o tuo tarpu vertė 0,5 rodo, kad nėra gebėjimo atskirti, o tai prilygsta atsitiktiniam spėjimui. Parametro apskaičiavimas parodytas 5je (Khan et al., 2023).

$$AUC(a) = \sum_{i=1}^{n-1} \frac{(FPR_i - FPR_{i-1}) * (TPR_i + TPR_{i-1})}{2}, \quad (5)$$

čia $AUC(a)$ – modelio prognozavimą rodantis parametras kintamajam „a”;

$FPR(a)$ – klaidingos teigiamos reikšmės rodiklis kintamojo „a” ribose;

Dar viena nelinijinė našumo metrika vadinama „plotu po tikslumo ir atšaukimo kreive (angl. *area under the precision-recall curve* (AUC-PR)). Ši metrika, priešingai nei jau aukščiau aprašytosios metrikos (išskyrus AUC) negali būti išskaičiuojama iš sumaišties matricos (angl. *confusion matrix*), nes jai reikia numatyti reitingavimo funkciją (Ma ir Lee, 2008). Ši metrika apibūdina bendrąjį modelio našumą, kur metrikos reikšmė „1“ reikštų tobulą modelio teigiamos ir

neigiamos klasių atskyrimą. Kaip ir sako šios metrikos pavadinimas, jos apskaičiavimui būtina žinoti tikslumo ir atšaukimo metrikas (Roberts, 2022).

„Logaritminis nuostolis“ (angl. *Log Loss*) yra logaritminė tikimybės funkcijos transformacija, kai pirmiausia naudojama tikimybinių klasifikatorių veikimui įvertinti. Skirtingai nuo kitų metrikų, pvz., tikslumo, logaritminio nuostolio praradimas atsižvelgia į prognozių neapibrėžtumą, labiau akcentuodamasis į patikimai neteisingas prognozes (DataScience-ProF, 2024). Logaritminio nuostolio formulė matoma 6je.

$$LogLoss = -\frac{1}{N} \sum_{i=1}^N (y_i * \log(p_i) + (1 - y_i) * \log(1 - p_i)) \quad (6)$$

čia *LogLoss* – logaritminis nuostolis;

N – stebėjimų skaičius;

y_i – tikrasis dvejetainis rezultatas (0 arba 1) *i*-ajam stebėjimui;

p_i – numatoma tikimybė, kad *i*-tas stebėjimas priklauso 1 klasei.

2.4. Empirinio taikomojo sukčiavimo atvejų nustatymo automatizavimo tyrimo atlikimas

Sukčiavimo atvejų nustatymo modelio automatizavimas bus pasiektas kelias etapais, kurie pavaizduoti 6 pav. ir aprašomi toliau pateiktuose punktuose:

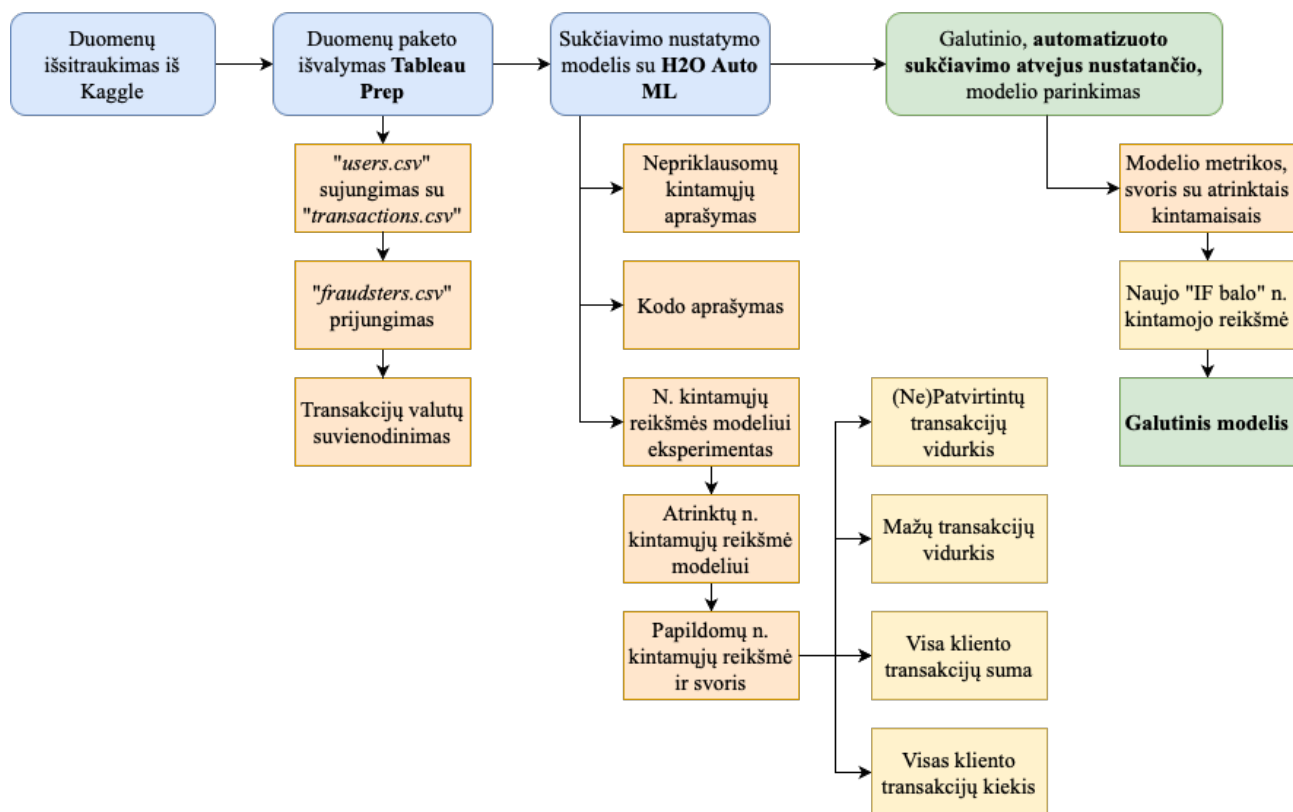
1 etapas. Duomenų failų išsitraukimas iš viešai prieinamos duomenų bazės.

2 etapas. Skirtingų duomenų failų išsivalymas ir jų sujungimas į vieną duomenų rinkinį atliekamas naudojant duomenų paruošimo programinę įrangą „Tableau Prep“.

3 etapas. Sukčiavimo atvejų nustatymo modelio kūrimas naudojant H2O AutoML modelį bei eksperimentai su juo.

4 etapas. Naujo nepriklausomo kintamojo, įsivedimas į modelį, kuris gautas neprižiūravimo mokymosi algoritmu bei jo įtaka modeliui.

5 etapas. Galutinio sukčiavimo atvejų nustatymo automatizavimo modelio parinkimas, pačio modelio ir jo metrikos pavaizdavimas.



Šaltinis: sudaryta autoriaus remiantis empirinio taikomojo tyrimo atlikimo etapais.

6 pav. Sukčiavimo atvejų nustatymo automatizavimo modelio parinkimo etapai

Apibendrinant sukčiavimo atvejų nustatymo metodologiją, galima teigti, kad empirinis tiriamasis darbas bus atliekamas per penkis pagrindinius etapus, kuomet bus išsitraukiami duomenys iš viešai prieinamų duomenų bazių, jie išsivalomi naudojant duomenų paruošimo programinį įrankį, sukuriamas automatizuotas sukčiavimo atvejus nustatantis modelis H2O AutoML įrankio pagalba. Toliau bus tikrinama naujo kintamojo, gauto IF algoritmu reikšmė modelių našumui bei kintamųjų svoriams bei parekomenduojamas galutinis modelis. Kadangi darbe yra nagrinėjamas klasifikacijos uždavinys, šiame skyriuje aprašomos modelio vertinimo metrikos, jų reikšmės bei pateikiamas H2O AutoML kodo pavyzdys Python programavimo kalba.

3. SUKČIAVIMO ATVEJŲ NUSTATYMO AUTOMATIZAVIMO, FINANSINIŲ PASLAUGŲ ĮMONĖSE, REZULTATAI

Išaiškinus sukčiavimo atvejų koncesiją bei etapus, kuriais galima automatizuoti sukčiavimo atvejų nustatymą, atliekamas eksperimentinis tyrimas. Tyrimo metu atliekami eksperimentai remiantis įrankiu, kuris naudoja prižiūrimą mašininių mokymąsi, taip nustatant modelių našumo metrikas bei nepriklausomų kintamųjų svorių pasiskirstymą. Parenkamas geriausiai veikiantis automatizuotas, sukčiavimo atvejus nustatantis, modelis. Tiriama papildomo nepriklausomo kintamojo, gauto neprižiūrimo mašininio mokymosi metodu, įtaka modeliams.

3.1. Duomenų rinkimas ir jų reikšmė

Duomenys buvo gauti iš internete prieinamos duomenų bazės „Kaggle.com“, kurioje atsispindi Vokietijoje registruotos licencijuotos elektroninių pinigų institucijos „N26“ klientų duomenys. Tyrimui naudojamų duomenų paketą sudaro keturių failų rinkinys (failų pavadinimai: *users.csv*, *transactions.csv*, *fx_rates.csv*, *fraudsters.csv*). Šiuose duomenyse yra pavaizduoti klientų, jų transakcijų bei valiutų kursų duomenys.

Pirmajame duomenų faile, pavadinimu „*users.csv*“, yra pateikiami duomenys, susiję su klientų asmeniniais bei sąskaitos identifikacijos duomenimis, kurie aprašomi 4 lentelėje.

4 lentelė

„*users.csv*“ duomenų failo analizė

Pavadinimas	Reikšmė	Aprašymas
F1	Skaičius	Eilutės skaičius
FAILED_SIGN_IN_ATTEMPTS	Skaičius, rodantis nesėkmingus prisijungimus prie savo anketos	Reikšmė tarp 0 ir 6, 0 reiškiantį, kad klientas nei karto nesusidūrė su nesėkmingu prisijungimu
KYC	Kliento KYC etapo statusas	„Failed“ – gražintas, „None“ – nežinoma, „Passed“ – patvirtintas, „Pending“ – laukiantis atsakymo
BIRTH_YEAR	Gimimo metai	Įvardyti gimimo metai YYYY formatu
COUNTRY	Kliento rezidencijos šalies kodas	60 šalių, parašytos tarptautiniais sutrumpinimais
STATE	Sąskaitos statusas	Aktyvi arba užrakinta
CREATED_DATE	Sąskaitos sukūrimo data ir laikas	Terminas nuo 2015-03-03 iki 2018-07-31
TERMS_VERSION	„Revolut“ programėlės versija	Terminas nuo 2017-06-01 iki 2018-09-20
PHONE_COUNTRY	Kliento telefono nr. šalies kodas	87 šalių, parašytos tarptautiniais sutrumpinimais
HAS_EMAIL	Kliento el. pašto turėjimas	0 – neturi, 1 - turi
ID	Kliento identifikacijos numeris	Klientų užslaptinti pavadinimai

Šaltinis: sudaryta autoriaus, remiantis Kaggle.com, „N26“ viešai prieinamais duomenimis.

Antrajame duomenų faile, pavadinimu „*transactions.csv*“, yra pateikiami duomenys, susiję su klientų mokėjimų informacija, kurie aprašyti 5 lentelėje.

5 lentelė

„*transactions.csv*“ duomenų failo analizė

Pavadinimas	Reikšmė	Reikšmė
CURRENCY	Transakcijos valiutos kodas	29 valiutos, parašytos tarptautiniais sutrumpinimais
AMOUNT	Transakcijos suma	Kiekvienos transakcijos suma, priskiriama konkrečioms klientų transakcijoms
STATE	Transakcijos statusas	„ <i>Canceled</i> “ – atšaukta kliento, „ <i>Completed</i> “ – įvykusi, „ <i>Declined</i> “ – atšaukta banko, „ <i>Failed</i> “ – nepavykusi, „ <i>Pending</i> “ – laukianti, „ <i>Recorded</i> “ – įrašyta, „ <i>Reverted</i> “ – grąžinta
CREATED_DATE	Transakcijos data ir laikas	Terminas nuo 2015-03-03 11:55:44:223 iki 2018-07-31 13:28:21:387
MERCHANT	Prekybininkas	Prekybininkų pavadinimai
MERCHANT_COUNTRY	Prekybininko šalis	253 šalys, parašytos tarptautiniais sutrumpinimais
ENTRY_METHOD	Mokėjimo patekimas	6 reikšmės, apibūdinančios kortelių mokėjimus
USER_ID	Kliento identifikacijos numeris	Klientų užslaptinti pavadinimai
TYPE	Mokėjimo metodas	5 reikšmės, apibūdinančios mokėjimo tipą (kortelių mokėjimas, banko pavedimas, pinigų įnešimas)
SOURCE	Išorinė šalis	11 išorinių šalių, naudojama sandoriams tvarkyti
Transaction_ID	Transakcijos identifikacijos numeris	Unikali reikšmė priskirta transakcijai

Šaltinis: sudaryta autoriaus, remiantis Kaggle.com, „N26“ viešai prieinamais duomenimis.

Trečiajame duomenų faile, pavadinimu „*fx rates.csv*“, yra pateikiami valiutų kodai ir jų kursai (6 lentelė).

6 lentelė

„*fx rates.csv*“ duomenų failo analizė

Pavadinimas	Reikšmė	Reikšmė
Base_ccy	Transakcijos valiutos kodas	Valiutos, parašytos tarptautiniais sutrumpinimais
ccy	Transakcijos valiutos kodas	Valiutos, parašytos tarptautiniais sutrumpinimais
rate	Valiutos kursas	Valiutos kurso reikšmė

Šaltinis: sudaryta autoriaus, remiantis Kaggle.com, „N26“ viešai prieinamais duomenimis.

Ketvirtajame duomenų faile, pavadinimu „*fraudsters.csv*“ yra pateikiami klientų identifikacijos duomenys, kurie pažymėti kaip atvejai, susiję su sukčiavimu.

Duomenų masyvas yra platus, leidžiantis visapusiškai ištirti ir nustatyti klientų įsitraukimą į su sukčiavimu susijusią veiklą. Tyrimo metu gauti sukčiavimo atvejų nustatymo modeliai gali būti panaudoti ne tik tiriant sukčiavimą, bet tuo pačiu ir kitus nusikaltimus, kaip pinigų plovimą ar terorizmo finansavimą, atitinkamai pritaikant modelį pagal savo turimus duomenis. Sukčiavimo atvejų aptikimo modelis gali būti naudojamas ir kitoms finansų įstaigoms, jei jų klientų duomenų masyvas turi modelio metu naudotą informaciją.

Tyrimo apribojimai, remiasi į tai, kad duomenys yra viešai prieinami, tad jų tikrumas ir realumas negali būti patikrintas. Taip pat, sukčiavimo modelio nustatymui yra labai svarbu bendras kliento profilis, kuris apibūdina kliento klausimyne pasirinktus atitikmenis apie jo lėšas, veiklą ir kitą, kas leistų dar labiau įvertinti kliento įsitraukimą į sukčiavimo veiklą.

3.2. Duomenų paketų paruošimas naudojant „Tableau Prep“

Kaip ir minėta šio skyriaus ankstesniame poskyryje, duomenų masyvą sudaro 4-ių failų rinkinys. Tam, kad būtų galima analizuoti pilnus duomenis ir matyti bendrą duomenų vaizdą, būtina juos susijungti į vieną failą bei juos išsivalyti. Būtent šis procesas yra pasiekimas su duomenų paruošimo programinės įrangos paketu „Tableau Prep“.

Duomenų masyvų tvarkymas ir paruošimas eksperimentui

Duomenų tvarkymo pradžioje, visiems failams buvo pakeistas jų tipas iš tekstinio į elektroninės skaičiuoklės (t.y. *excel*) formatą bei tokie duomenų failai atidaryti „Tableau Prep“ programinio paketo įrankyje. Duomenų failai, pavadinimu „*users.xlsx*“ ir „*trasactions.xlsx*“, iškelti į bendrą „Tableau Prep“ vaizdavimo plotą (7 pav.), jiems inicijuoti valymo etapai, kurie automatiškai suskirsto duomenis į kategorijas, priskiria duomenų tipą bei pavaizduoja kokius duomenis turi kiekvienas duomenų paketo stulpelis.



Šaltinis: sudaryta autoriaus, naudojant „Tableau Prep“ programinės įrangos paketą.

7 pav. Klientų duomenų paketų tvarkymas

7 pav. pavaizduota, kaip pirmajame duomenų faile, pavadinimu „users“ valymo etape (*Clean_1*) buvo pakeisti duomenų pavadinimai, dauguma jų perrašyti mažosiomis raidėmis, kadangi originaliuose duomenyse buvo parašyti iš didžiųjų. Taip pat jie pervardinti, kad labiau atitiktų reikšmes, apibūdintas 4 lentelėje – klientų identifikacijos numeriui pakeistas duomenų tipas, kad labiau atitiktų turimus duomenis, tad pirmasis duomenų failas paruoštas tolesniems procesams. Antrajame duomenų faile, pavadinimu „transactions“, valymo etape buvo pašalinti tyrimui nenaudotini duomenys, tokie kaip transakcijos eilutės numeris bei transakcijos identifikacijos kodas. Taip, pat prekybininko šalies duomenyse buvo įsimaišiusių reikšmių, kurios neapibūdina šalies, tad valymo procese jos buvo pašalintos iš bendro duomenų masyvo. Kai jau abu duomenų failai išvalyti, inicijuojamas pirmojo ir antrojo duomenų failo sujungimas naudojant vidinio prisijungimo funkciją (angl. „inner join“), kurios metu pirmasis duomenų failas yra prijungimas šalia antrojo duomenų failo pagal vieną kartinį kriterijų – klientų identifikacijos numerį ir paliekami tik tie numeriai ir su jais susiję duomenys, kurie yra abiejuose duomenų failuose, o kiti nesutapę atmetami automatiškai. Kad būtų galima lengviau pertvarkyti gautą naująjį duomenų masyvą, inicijuojamas valymo etapas. Kadangi prisijungimo funkcijos metu, pirmasis duomenų failas buvo prijungtas prie antrojo, minėtame valymo etape, pavadinimu „Clean_3“, pirmiausia stulpelių eiliškumas perskirstomas taip, kad būtų patogų jais naudotis ir analizuoti. Kai jau duomenų stulpelių eiga yra išdėstyta pagal poreikius, valymo etape yra panaikinami kiti susidubliavę vidinio prisijungimo funkcijos metu gauti duomenys, klientų identifikacijos numeris pervardinamas, kad labiau atitiktų reikšmes apibūdintas 5 lentelėje.

Tam, kad būtų galima atlikti tolimesnį duomenų valymo procesą, buvo pasiruoštas ir trečiasis duomenų failas, pavadinimu „fraudsters“, kuriame yra įvardyti klientų identifikacijos numeriai, kurie jau iš seniau nustatyti kaip susiję su sukčiavimu klientai. Valymo etape pervardintinas klientų identifikacijos numeris, bei sukurtas papildomas suskaičiuotas stulpelis, kuris visiems šiems su sukčiavimu susijusiems klientams, priskiria reikšmę „1“. Sutvarkytas failas prijungimas prie pirmųjų dviejų failų valymo etapo „Clean_3“, naudojant kairiąją prisijungimo funkciją (angl. *left join*), reiškiančią, kad visos reikšmės, esančios „fraudsters“ faile yra prijungiamos prie „Clean_3“ duomenų.

Toliau inicijuotame valymo etape yra pašalinamas susidubliavęs stulpelis – klientų identifikacijos numeris. Gimimo datos stulpelio formatas buvo pakeistas į datos formatą, o iš jo suskaičiuotas naujas stulpelis, nusakantis klientų amžių pagal jų gimimo datą. Kadangi duomenų masyvas apima 2015-2018 metus, klientų amžiaus skaičiavimas buvo atliekamas ne pagal dabartinius metus, bet pagal 2018 metų ribą. Atliekamas dar vienas skaičiuojamasis veiksmas, kurio metu prie klientui priskirto eilės numerio, yra pridedamas „C_“ inicialas, taip gaunant trumpesnius anoniminius

klientų pavadinimus, pavyzdžiui „C_I“. Šis veiksmas buvo atliekamas, nes klientų identifikacijos numeris yra labai ilgas, kurį sudaro apie 20 raidžių ir skaičių kratinys. Tai atlikus, pašalinamas stulpelis su ilgu klientų identifikacijos numeriu ir paliekamas naujasis, pavadintas – „*kliento_ID*“. Taip, pat pašalinta nulinė reikšmė iš stulpelio, apibūdinančio naudotos „Revolut“ programėlės versiją. Kai jau duomenų failas pilnai sutvarkytas, išsisaugota jo galutinė versija tekstiniu formatu. Galutinį, išvalytą, sutvarkytą ir sujungtą į vieną duomenų bazę, duomenų failą iš viso sudaro 7 034 klientai, iš kurių 232 buvo pažymėti kaip susiję su sukčiavimu.

Valiutų suvienodinimas

Kadangi duomenų faile, atsispindintys klientų mokėjimai, yra skirtingomis valiutomis, yra netikslu vertinti klientų mokėjimų sumas. Tam, kad sukčiavimo atvejų modelio nustatymo rezultatai būtų tikslūs, visų klientų mokėjimai yra perverčiami į vieną valiutą – eurą. Failų sujungimas ir perversimas į vieną valiutą nebuvo atliktas „Tableau Prep“ programiniame įrankyje, kaip kiti duomenų failai, nes valiutų kursai neturi identifikacijos numerio. Failų sujungimo kodas buvo aprašytas Python programine kalba. Pirmiausia prieš tai gautas galutinis duomenų bei valiutų kursų failas, pavadinimu „*fx rates.csv*“ atsidaromas bei užkraunamas „Kaggle“ internetinėje erdvėje. Toliau apibūdinami funkcijos parametrai, pagal kuriuos modelis turėtų orientuotis ir paversti transakcijas į bendrą valiutą – eurus. Modelyje patikrinama ar transakcija jau yra eurais, jeigu yra – funkcija grąžina tą pačią reikšmę, kokia ir buvo nurodyta. Jeigu modelis randa, kad yra transakcijų ne eurais, valiutų faile ieško transakcijos normos iš eurų į turimą transakciją, bei atvirkščiai. Modelis tai padaro kiekvienai transakcijai, ir sukuria naują stulpelį, kuriame yra naujos įvykusių mokėjimų reikšmės, visos pervestos į eurus. Gautas galutinis duomenų failas, pavadintas „*galutiniai_duomenys_eur.csv*“, kuris bus naudojamas tolimesniems eksperimentams.

3.3. Sukčiavimo nustatymo modelis

Tam, kad būtų galima sukurti gerai veikianti sukčiavimo atvejų nustatymo modelį, būtina atkreipti dėmesį į daugelį dalykų. Šiame poskyryje apibūdinami naudoti nepriklausomi kintamieji, aprašomi atlikti eksperimentai bei modelių analizės ir vertinimas, pagal kuriuos pasirenkamas geriausiai veikiantis modelis, atsižvelgiant į modelių metrikas, procentinius nepriklausomų kintamųjų svorius ir modelių persimokymo procentą.

3.3.1. Sukčiavimo atvejų nustatymo modelio kintamieji

Kad tiksliai nustatyti sukčiavimo atvejus, yra labai svarbu parinkti tinkamus nepriklausomus kintamuosius, pagal kuriuos sukurtas modelis turės ieškoti anomalijų, mokytis ir gebėti atspėti sukčiavimo atvejus. Toliau aprašomas kintamųjų pasirinkimas bei jų reikšmė galimam sukčiavimo atvejų nustatymo automatizuotam modeliui.

1. Mažos, dažnos transakcijos – vienas iš pagrindinių požymių, kad klientas galimai susijęs su sukčiavimu yra labai dažnos transakcijos, kuriose pervedama suma yra maža. Šis kintamasis nebūtinai turi būti tinkamas tik sukčiams, tačiau, kadangi sukčiai dažnai nežino esamo lėšų likučio klientų sąskaitose, inicijuoja mažas, dažnas transakcijas. Modelyje parinktas transakcijos slenkstis iki 20 piniginių vienetų / eurų per minutę.
2. Didelė suma (piniginiais vienetais) – kitas kintamasis, kuris taip pat dažnai išskiriamas kaip vienas kertinių kriterijų, tai yra didelių sumų transakcijos. Šiuo atveju, į modelį įvestas kintamasis, kuris analizuoja transakcijas, kurių vienos transakcijos suma yra didesnė nei 100,000.00 piniginių vienetų.
3. Didelė suma (eurais) – kadangi modelyje pateiktos sumos ne visos buvo vienos valiutos, visos sumos buvo perskaičiuotos į eurus, kas aprašyta „*Valiutų suvienodinimas*“ paragrafe. Taip pat, kaip ir minėta prieš tai, didelės transakcijos, gali taip pat būti susijusios su sukčiavimu. Šiuo atveju, į modelį įvestas dar vienas kintamasis, kuris analizuoja transakcijas, kurių suma didesnė nei 100,000.00 eurų.
4. Pavedimai ne darbo valandomis – apgavikai dažnai veikia naktį, arba bent jau ne darbo valandomis, kad klientai mažiau pastebėtų ir ne taip greitai kreiptųsi į finansų įstaigas ir užsiblokuotų korteles ar sąskaitas. Dėl šios priežasties, į modelį paimtas kintamasis, kuris analizuoja transakcijas neįprastomis darbo valandomis, t.y. nuo 23 val. vakaro iki 6 val. ryto.
5. Apvalūs skaičiai – apgavikai, dažnai inicijuoja transakcijas, kuriose sumos buvo apvalios bei lygios, t.y. be jokių skaičių po kablelio. Dėl šios priežasties, modeliui parinktas dar vienas kintamasis, kuris pažymėtų transakcijas, kuriuose yra dešimtainiai skaičiai.
6. Transakcijų suma per dieną – didelių mokėjimų sumos gali turėti didelę įtaką sukčiavimo nustatymui, ypač jeigu tos sumos nukrypsta nuo įprastų mokėjimų, būtent tam, į modelį įvestas kintamasis, kuris stebi transakcijų sumas, įvykusias per dieną.
7. Transakcijų kiekis per dieną – transakcijų kiekis taip pat yra labai svarbus kintamasis pagal kurį būtų galima nustatyti sukčiavimo atvejus. Didelis transakcijų skaičius gali simbolizuoti prasidėjusį sukčiavimą.

8. Nuokrypis nuo vidurkio – šis kintamasis turėtų būti vienas pagrindinių kintamųjų, nes jo pagalba bus skaičiuojamas sumų nuokrypis klientų sąskaitose nuo vidurkio.
9. Šalių skirtumas – didelė šalių kaita yra stiprus kintamasis, simbolizuojantis, kad galimai klientų sąskaita galėjo būti pasisavinta. Šiame kintamajame yra analizuojama ar kliento rezidavimo, telefono ir pardavėjo šalies kodai skiriasi.
10. (Ne)Patvirtintų transakcijų vidurkis – turimoje klientų duomenų bazėje buvo nemaža dalis klientų, kuriuose dauguma transakcijų nebuvo patvirtintos. Tai yra vienas iš pagrindinių rodiklių, keliantis įtarimą, kad klientai ar sukčiai galima galėjo užsiimti nelegalia veikla, kai didelė transakcijų suma nebuvo patvirtinta finansų įstaigos darbuotojų.
11. Mažų transakcijų vidurkis – kaip ir minėta pirmajame punkte, mažos, dažnos transakcijos gali rodyti galimus sukčiavimo atvejus, dėl šios priežasties, į modelį buvo įvestas dar vienas kintamasis, kuris tikrina ne atskiras transakcijas, o transakcijų vidurkius.
12. Visas kliento transakcijų kiekis – didelis transakcijų kiekis per visą laikotarpį gali būti laikomas vienu iš kriterijų sukčiavimo atvejų nustatymui.
13. Visa kliento transakcijų suma – kaip ir minėta prieš tai, didelės transakcijos gali rodyti klientų galimą įtraukimą į sukčiavimą, ypač jei tai būtų staigus pasikeitimas.
14. IF balas – tai yra papildomas kintamasis modeliui, kuris duoda balą, apskaičiuotą IF algoritmu. Jis aptinka galimus iškrypimus arba anomalijas operacijų duomenų rinkinyje remiantis neprižiūrimu mašininio mokymosi.

Visi kintamieji bei juose įvardyti slenksčiai buvo parinkti, pagal turimus duomenis aprašytus 3.1. poskyryje. Atrinkti tokie kintamieji, kurie gali daryti didelę įtaką sukčiavimo atvejų nustatymui, tam, kad modelis galėtų mokytis ir gerai aptikti bei prognozuoti sukčiavimą.

3.3.2. Sukčiavimo atvejų modelio automatizavimo kodo aprašymas

Automatizuotam sukčiavimo atvejų modeliui gauti buvo parašytas kodas virtualioje erdvėje naudojant Python programavimo kalbą. Kodas, nustatantis sukčiavimo atvejus, buvo atliktas naudojant H2O AutoML įrankį, kurio šablonas paimtas iš jų oficialios internetinės svetainės (**5 pav.**). Toliau pateikiamas detalus kodo aprašymas žingsniais, kas buvo daroma ir funkcijų reikšmės. Pilnas kodas, su kuriuo gautas galutinis modelis, pateikiamas 1 PRIEDAS.

1 žingsnis. Pirmiausia, kad kodas galėtų tinkamai veikti, buvo importuoti duomenų analizės bei anomalijų aptikimo įrankiai, tokie kaip IF algoritmas, kurio pagalba buvo aptinkamos anomalijos balas bei H2O AutoML įrankis, kuris yra pilnai automatizuotas ir gali pats parinkti tinkamiausius algoritmus ir parametrus pagal sprendžiamą problemą. Buvo paleistas H2O įrankio veikimas.

2 žingsnis. Įsiimportuotas duomenų failas „galutiniai_duomenys_eur.csv“, naudojamas analizei ir eksperimentų aptikimui, nurodyti parametrai, kaip duomenų failą sistemai skaityti, kad jis būtų tinkamai užkrautas ir suprastas.

3 žingsnis. Konvertuojamas datos ir laiko formatas, kad būtų galima atlikti tinkamai apsirašyti nepriklausomus kintamuosius.

4 žingsnis. Apsirašomi nepriklausomi kintamieji, kuriuos norima, kad H2O AutoML įrankis įtrauktų į modelius. Kintamieji apibūdinami įrankiui, išskiriant pagal kuriuos kintamuosius norima, kad sistema atliktų skaičiavimus.

5 žingsnis. Apibūdinami IF algoritmo parametrai bei nepriklausomi kintamieji, pagal kuriuos parenkamas anomalijos balas.

6 žingsnis. Išskiriama kas yra priklausomas kintamasis (šiuo atveju jau nustatytas sukčiavimas iš istorinių duomenų) bei kokie nepriklausomi kintamieji dalyvauja modelio kūrime. Jei kažkokio kintamojo nereikia, jis pažymimas „#“, kad šio nepriklausomo kintamojo sistema į modelį netrauktų.

7 žingsnis. Parašoma, kad sistema priklausomą kintamąjį imtų kaip kategorinį kintamąjį tam, kad įrankis šį uždavinį spręstų kaip klasifikacijos, o ne regresijos problemą, tad tam pakeičiamas duomenų rėmelis į H2O skaitymui suprantamą rėmelį.

8 žingsnis. Duomenų bazė padalinama į tris dalis, atskiriant duomenis, pagal kuriuos modelis mokysis, validuos ir kuriuos testuos. Modelio treniravimuisi palikta 0,7 visos duomenų bazės, validacijai ir testavimui po 0,15, bei parinkta „sėkla“ (angl. *seed*), lygi 42. Iš literatūros analizės, nustatyta, kad tai yra standartinė naudojama sėkla tokio pobūdžio užduotims spręsti. Taip pat parinkti parametrai, kad įrankis vienai užklausai maksimaliai galėtų veikti 2 val., balansuotų duomenų bazę, sudarytą 10 modelių, bei skaičių kartotinis (angl. *nfolds*) nurodyta – 0, nes tam buvo skirta atskira duomenų validacijos imtis. Nurodyta, kad X raidė atitinka nepriklausomus kintamuosius, o Y – priklausomą bei kad įrankis treniruotų modelius remdamasis ir mokymosi ir validacijos imtimis.

9 žingsnis. Parašoma komanda, kurios pagalba, sugeneravus 10 modelių, jie yra pateikiami toje pačioje teritorijoje. H2O AutoML įrankyje, tai apibūdinama kaip „lyderių lenta“ (angl. *leaderboard*).

10 žingsnis. Sukuriamas sąrašas, kuriame bus saugomos visų gautų modelių svoris. Įrankis tikrina gautus modelius, jeigu jų nepriklausomiems kintamiesiems įmanoma priskirti svorį, jis tai padaro bei prideda modelių ID, pakeičia į „DataFrame“ rėmelį. Kombinuoja visus modelių svorius į vieną failą ir jį išsaugo į atskirą tekstinį failą.

11 žingsnis. Šiame etape inicijuojamas tuščias sąrašas, kuriame bus išsaugoti modelių parametrai. Kadangi, įrankis sugeneruodavo labai daug parametrų, kurie nebuvo reikalingi modelio analizėms, tai buvo nurodyta, kokie parametrai yra reikalingi. Kodas ima modelius, sugeneruotus

lyderių lentoje, ištraukia tik reikalingus parametrus, priskiria modelių ID, pakeičia į „*DateFrame*“ rėmelį, vėliau kombinuoja su jau esančiu duomenų rėmu, nes jis buvo keičiamas bei išsaugo rezultatus į atskirą tekstinį failą.

12 žingsnis. Parašyta sistemai peržiūrėti lyderių lentą ir ištraukti modelio metrikas, padarytas remiantis mokymosi, validacijos ir testavimo duomenų imtimis, pakeičia į „*DateFrame*“ rėmelį. Visos nurodytos metrikos sujungtos ir išsaugotos į vieną tekstinį failą.

13 žingsnis. Sukuriamas dar vienas papildomas sąrašas, kuriame kodas vėl peržiūri lyderių lentą ir modelių testuojamą imtį, pagal ją ištraukia sumaišties matricos rezultatus, pakeičia į „*DateFrame*“ rėmelį, sujungia visas matricas į vieną failą ir išsaugo tekstiniu formatu.

3.3.3. Visų nepriklausomų kintamųjų reikšmė modeliui

Sukčiavimo atvejų nustatymo modelis pirmiausia pradėtas vykdyti nuo visų turimų kintamųjų įsivedimo į modelį ir stebint jų bendrą įtaką pačioms modelių metrikoms. Eksperimentas buvo atliekamas H2O AutoML įrankio pagalba su visais 13-a nepriklausomų kintamųjų (14-as nepriklausomas kintamas yra papildomas tolimesniuose eksperimentuose), kurie aprašyti 3.3.1. paragrafe. Šiame eksperimente buvo gauti 10 modelių, kuriuose minėtasis įrankis iš viso parinko penkis algoritmus bei skirtingus jų parametrus. Modelių testavimo imties metrikų rezultatai aprašyti 7 lentelėje.

7 lentelė

Trylikos kintamųjų metrikos

Modelio pav.	Imtis	Tikslumas	F1 balas	AUC	AUC-PR	Log. nuostolis
GBM_4	Testavimo	0,9997	0,9927	1,0000	0,9995	0,0030
GBM_2	Testavimo	0,9996	0,9880	1,0000	0,9989	0,0045
XGBoost_2	Testavimo	0,9997	0,9916	1,0000	0,9992	0,0017
GBM_3	Testavimo	0,9997	0,9916	1,0000	0,9993	0,0035
GBM_1	Testavimo	0,9996	0,9883	1,0000	0,9986	0,0039
XGBoost_1	Testavimo	0,9996	0,9897	1,0000	0,9989	0,0020
XGBoost_3	Testavimo	0,9998	0,9957	0,9999	0,9992	0,0020
XRT_1	Testavimo	0,9971	0,9197	0,9989	0,9705	0,0535
DRF_1	Testavimo	0,9975	0,9298	0,9988	0,9721	0,0307
GLM_1	Testavimo	0,9813	0,2407	0,8361	0,1352	0,0774

Šaltinis: sudaryta autoriaus remiantis pirmojo eksperimento rezultatais.

Analizuojant gautus rezultatus, galima matyti, kad buvo gauti 10 modelių, kuriuose naudoti penki algoritmai: GBM, XGBoost, XRT, DRF ir GLM (algoritmų aprašymai pateikiami

1.4. poskyryje). Iš 7 lentelės, galima matyti, kad geriausi modelių metrikų rezultatai gauti su XGBoost arba GBM algoritmu, prasčiau su GLM. Gauti metrikų rezultatai ant testuojamos duomenų bazės yra labai aukšti, ir praktiškai niekuo nesiskiria nuo apmokymo ir validacijos duomenų imčių (2 PRIEDAS), tačiau mašininio mokymosi pagalba gautuose modeliuose yra labai svarbu stebėti modelių persimokymą, o tokie aukšti rezultatai to padaryti neleidžia ir net gali rodyti persimokymą vien dėl per gerų metrikų. Kadangi pats H2O AutoML įrankis neleidžia mažinti medžių skaičiaus, tolimesniuose eksperimentuose bus pašalinami didžiausią svorį modeliams turintys nepriklausomi kintamieji, kad gautos galutinės modelių metrikos leistų įvertinti galimą modelių persimokymą, tačiau ir toliau išliktų geros (daugiau aprašyta 1.6. poskyryje).

8 lentelėje yra pateikiami modelių parametrai, su kuriais buvo gautos metrikos, aprašytos 7 lentelėje ir 2 PRIEDAS.

8 lentelė

Trylikos kintamųjų modelių parametrai

Nr.	Modelio pav.	Klasių balansavimas	Medžių skaičius	Maksimalus gylis	Minimalius eilučių skaičius	Imties dažnis
1.	GBM_4	+	180	10	10	0,8
2.	GBM_2	+	350	7	10	0,8
3.	XGBoost_2	-	200	10	5	0,6
4.	GBM_3	+	290	8	10	0,8
5.	GBM_1	+	130	15	100	0,8
6.	XGBoost_1	-	165	15	10	0,6
7.	XGBoost_3	-	670	5	3	0,8
8.	XRT_1	+	35	20	1	0,632
9.	DRF_1	+	35	20	1	0,632
10.	GLM_1	+	-	-	-	-

Šaltinis: sudaryta autoriaus pagal pirmojo eksperimento rezultatus. Visi stulpeliai su vienodomis reikšmėmis: Pasiskirstymo (angl. *distribution*) reikšmė „*bernoulli*“ (išskyrus 10-oje eilutėje) ir skaičių kartotinis (angl. *nfolds*) – „0“.

Iš 8 lentelės, galima matyti, kad modelyje, gautame su GBM algoritmu, naudotas medžių skaičius siekė tarp 130-350, o XGBoost net iki 670. Su XRT, DRF algoritmu H2O AutoML parinktas medžių skaičius buvo žymiai mažesnis, tik – 35, tačiau maksimalus medžių gylis gerokai didesnis nei kitų algoritmų. Iš aprašytų rezultatų, taip pat galima matyti, kad XGBoost algoritme duomenų bazė niekada nėra balansuojama, kai su visais kitais algoritmais – balansuojama. Atliekant eksperimentus tik su GLM algoritmu nereikia nurodyti nei medžių skaičiaus, nei jų gylio ar kitų parametrų.

3.3.4. Atrinktų nepriklausomų kintamųjų reikšmė modeliui

Kaip ir minėta 3.3.3. paragrafe, tam, kad būtų galima analizuoti modelio persimokymą, buvo atliekami eksperimentai nustatant nepriklausomų kintamųjų svorį, ir šalinant po vieną, didžiausią svorį turintį, kintamąjį, bandant nustatyti ribą, kur modelio metrikų rezultatai ženkliai keičiasi. 9 lentelėje pavaizduotas nepriklausomų kintamųjų numeris, kurių pavadinimai ir reikšmės aprašytos 3.3.1. paragrafe, tų kintamųjų svoris, gautas su nurodytu algoritmu, bei modelių metrikos.

9 lentelė

Nepriklausomų kintamųjų svoriai ir metrikos

Kintamieji	Kintamojo svoris	Modelio pav.	Imtis	Tikslumas	F1 balas	AUC	AUC-PR	Log. nuotolis
13-is	37,47 %	GBM_4	Testavimo	0,9997	0,9927	1,0000	0,9995	0,0030
12-is	37,25 %	GBM_4	Testavimo	0,9996	0,9897	1,0000	0,9987	0,0049
11-is	50,15 %	GBM_4	Testavimo	0,9996	0,9880	0,9999	0,9981	0,0079
10-is	50,08 %	XGBoost_2	Testavimo	1,0000	0,9995	0,9996	1,0000	0,0022
10-is	60,41 %	GBM_4	Testavimo	0,9913	0,7549	0,9857	0,8352	0,0400
9-is	50,48 %	XGBoost_3	Testavimo	0,9866	0,6162	0,8866	0,6805	0,1885
9-is	45,10 %	GBM_4	Testavimo	0,9827	0,3240	0,8361	0,2820	0,0721

Šaltinis: sudaryta autoriaus remiantis antrojo eksperimento rezultatais.

Iš gautų rezultatų galima matyti, jog visų modelių testavimo imties tikslumas buvo aukštas, logaritminis nuotolis – minimalus. 13-ojo ((ne)patvirtintų transakcijų vidurkis) kintamojo svoris, nustatytas modelyje su GBM_4 algoritmu, siekė 37,47 %, jo F1 balo, AUC ir AUC-PR rodiklis, buvo arti maksimalios reikšmės. Tokie patys rezultatai gauti ir su 12-ju (mažų transakcijų vidurkis) ir 11-ju (visas kliento transakcijų kiekis) kintamaisiais, kuriuose jų svoris sudarė 37,25 % ir 50,15 % atitinkamai. Modelių su dešimt ir devyniais nepriklausomais kintamaisiais, geriausios metrikos buvo naudojant XGBoost algoritmą, o reikšmės, gautos su GBM algoritmu aprašytos taip pat, kad būtų galima lyginti su kitų kintamųjų gautais duomenimis. Su 10 kintamųjų, modelio metrikos su XGBoost algoritmu buvo labai aukštos, o su GBM jau pradėjo žemėti, ko ir buvo siekiama šiuo eksperimentu. Pašalinus ir 10-ąjį kintamąjį, modelio metrikos krito net ir su XGBoost algoritmu, o su GBM – jau buvo prastos, tad galima teigti, jog čia yra riba, kad modelis veikia prastai ir reikia įsivesti papildomus kintamuosius, kad metrikos būtų aukštesnės ir modelio gebėjimas atskirti sukčiavimo atvejus nuo įprastų klientų, liktų realus ir geras.

3.3.5. Papildomų nepriklausomų kintamųjų reikšmė ir svoris modeliams

Kadangi, iš praeito eksperimento nustatyta, kad su devyniais kintamaisiais modelio metrikų rezultatai ženkliai krenta, o su visais trylika kintamųjų gauti modeliai stipriai persimokę, buvo atliekami tolimesni eksperimentai, bandant atrasti ir pasirinkti vieną papildomą kintamąjį iš 4-ių (t.y. (ne)patvirtintų transakcijų vidurkio, mažų transakcijų vidurkio, viso kliento transakcijų kiekio ir visos kliento transakcijų sumos), kurį pridėjus prie turimų devynių kintamųjų, modelis veiktų gerai ir būtų mažiausiais persimokęs. Galutinių rezultatų apžvalga pateikiama po 3.3.5.5 paragrafe.

3.3.5.1. Kintamojo, (ne)patvirtintų transakcijų vidurkio, reikšmė ir svoris modeliams

Nepriklausomas kintamasis, pavadintas „(ne)patvirtintų transakcijų vidurkis“, buvo papildomai įvestas į modelius, kuriuose jau buvo devyni nepriklausomi kintamieji. 10 lentelėje pateikiamos trijų pasirinktų modelių metrikos, kurie taip pat bus aprašomi ir kituose eksperimentuose bandant išsirinkti papildomą kintamąjį. Lentelėje taip pat pateikiama ir mokymosi, ir validacijos, ir testavimo imtys, pagal kurias galima spręsti modelio persimokymo procentą, 11 lentelėje pateikiami tų modelių parametrai.

10 lentelė

Kintamojo, (ne)patvirtintų transakcijų vidurkio, metrikos

Modelio pav.	Imtis	Tikslumas	F1 balas	AUC	AUC-PR	Log. nuotolis
XGBoost_2	Mokymo	0,9999	0,9979	1,0000	0,9997	0,0056
XGBoost_2	Validacijos	0,9959	0,8868	0,9938	0,9333	0,0169
XGBoost_2	Testavimo	0,9958	0,8804	0,9939	0,9068	0,0169
GBM_4	Mokymo	0,9946	0,8532	0,9975	0,9243	0,0251
GBM_4	Validacijos	0,9934	0,8112	0,9940	0,8805	0,0293
GBM_4	Testavimo	0,9935	0,8090	0,9934	0,8684	0,0299
DRF_1	Mokymo	0,9968	0,9141	0,9990	0,9680	0,0405
DRF_1	Validacijos	0,9911	0,7348	0,9742	0,7810	0,0557
DRF_1	Testavimo	0,9911	0,7308	0,9745	0,7679	0,0554

Šaltinis: sudaryta autoriaus remiantis trečiojo eksperimento rezultatais.

Iš pateiktų rezultatų 10 lentelėje, galima matyti, jog visų modelių ir imčių tikslumas yra aukštas, o visų validacijos ir testavimo imčių reikšmės yra panašios. Kaip minėta literatūros apžvalgoje, didelis skirtumas tarp testavimo imties nuo validacijos ir testavimo, rodo modelio persimokymą. Sprendžiant iš FI balo rodiklio, rodo, gautas su XGBoost algoritmu, yra persimokęs,

nes mokymosi imtis yra labai aukšta, o testavimo – sumažėjus net 11 %. Šio modelio abi AUC reikšmės aukštos, logaritminis nuotolis minimalus, nors skirtumas tarp AUC-PR imčių rodo 7 % skirtumą. Mažiausią persimokymą turėjo modelis, gautas su GBM algoritmu, skirtumas tarp mokymosi ir testavimo imties yra apie 5-6 %, kas šiek tiek vis tiek indukuoja minimalų modelio persimokymą. Modelis, gautas su DRF algoritmu, yra labiausiai persimokęs modelis iš visų – net apie 18-20 %, turi didžiausią logaritminį nuostolį iš visų trijų analizuojamų modelių. Apibendrinus 10 lentelę, būtų galima teigti, kad modelis, pavadintas GBM_4 veikia geriausiai, o DRF_1 – prasčiausiai.

11 lentelėje pateikti parametrai, su kuriais buvo gautos modelių metrikos aprašytos 10 lentelėje. 11 lentelėje pateikti parametrai, kuriuos H2O AutoML įrankis parinko modeliams, nes kaip ir minėta prieš tai, įrankis neleidžia pasirinkti medžių skaičiaus.

11 lentelė

Kintamojo, (ne)patvirtintų transakcijų vidurkio, parametrai

Modelio pav.	Klasių balansavimas	Medžių skaičius	Maks. gylis	Min. eilučių skaičius	Imties dažnis
XGBoost_2	-	395	10	5	0,6
GBM_4	+	155	10	10	0,8
DRF_1	+	40	20	1	0,632

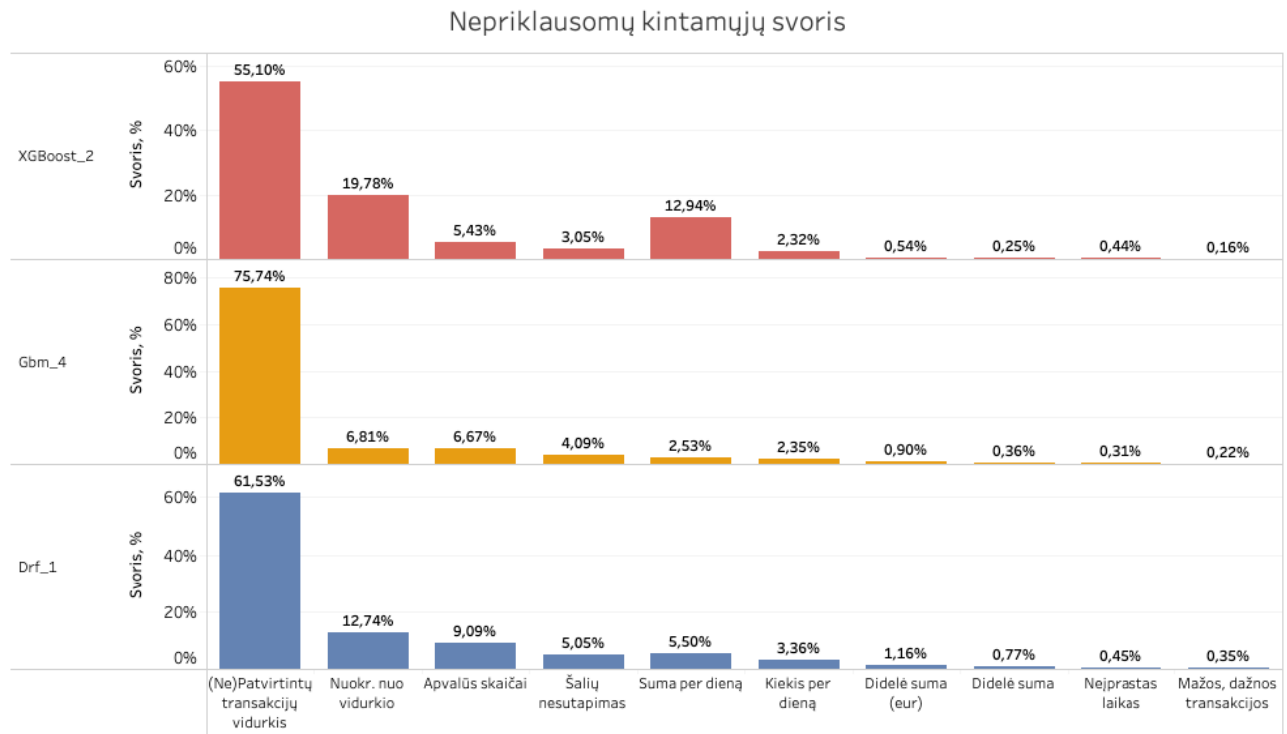
Šaltinis: sudaryta autoriaus remiantis trečiojo eksperimento rezultatais. Pasiskirstymo (angl. *distribution*) reikšmė „bernoulli“ ir skaičių kartotinis (angl. *nfolds*) – „0“.

Analizuojant 11 lentelę, galima matyti, kad modeliui, kuris pavadintas XGboost_2 nebuvo balansuojama duomenų bazė, kaip ir pastebėta prieš tai atliktuose eksperimentuose. Parinktas medžių skaičius yra 395, kai maksimalus jų gylis 10, eilučių skaičius 5, o imties dažnis 0,6. Su GBM algoritmu, parinktas medžių skaičius buvo mažesnis, siekė – 140, bei didesnis minimalus eilučių skaičius. DRF_1 modelio medžių skaičius tik – 40, tačiau buvo didžiausias maksimalus medžių gylis ir mažiausias minimalus eilučių skaičius.

Tam, kad būtų galima stebėti ne tik modelių metrikų rezultatus bei parametrus, su kuriais jos buvo pasiektos, taip pat svarbu atkreipti dėmesį ir į modelio nepriklausomų kintamųjų svorius. 8 pav. yra pavaizduoti dešimt nepriklausomų kintamųjų bei jų visų procentiniai svoriai. Paveiksle taip pat matomi visi trys algoritmai (XGBoost pažymėtas raudona spalva, GBM – oranžine, o DRF – mėlyna) kurie ir yra pasirinkti nagrinėti per visus eksperimentus.

Iš 8 pav. galima matyti, jog įvestas naujasis kintamasis „(ne)patvirtintų transakcijų vidurkis“ sudaro didžiausią svorį modelyje, pvz. XGBoost_2 modelyje šio nepriklausomo kintamojo svoris yra apie 55 %, o GBM_4 modelyje net – virš 75 %, DRF – apie 62 %, tad visuose modeliuose dominuoja vienas kintamasis. Antrasis kintamasis pagal svorio procentinį dydį yra „nuokrypis nuo vidurkio“,

kuris XGBoost_2 modelyje sudaro didžiausią svorį, o GBM – mažiausią. Kitų kintamųjų svoris yra daugmaž panašus, tačiau XGBoost_2 modelyje, kintamasis „suma per dieną“ sudaro didesnę procentinę svorį nei kituose modeliuose. Iš 8 pav. taip pat galima matyti, jog keturi paskutiniai kintamieji sudaro labai mažą svorį ir neduoda didelės naudos modeliams, dauguma iš jų net nesiekia 1 % modelių svorio.



Šaltinis: sudaryta autoriaus, remiantis trečiojo eksperimento rezultatais, naudojant „Tableau Desktop“ vizualizacijos įrangos paketą.

8 pav. Nepriklausomų kintamųjų svoris modeliui

3.3.5.2. Kintamojo, mažų transakcijų vidurkio, reikšmė ir svoris modeliams

Šiame paragrafe atliktas dar vienas eksperimentas, kuriame į modelį su devyniais nepriklausomais kintamaisiais buvo įvestas papildomas kintamasis, kuris pavadintas „mažų transakcijų vidurkis“, taip gaunant dešimt nepriklausomų kintamųjų. 12 lentelėje taip pat pateikiami tie patys modeliai, kurie buvo aprašomi ir 3.3.5.1. paragrafe. Lentelėje pateikiamos metrikos, gautos mokymosi, validacijos ir testavimo duomenų imtyse. 13 lentelėje pateikiami modelių parametrai parinkti H2O AutoML įrankio, su kuriais gautos modelių metrikos.

Kintamojo, mažų transakcijų vidurkio, metrikos

Modelio pav.	Imtis	Tikslumas	F1 balas	AUC	AUC-PR	Log. nuostolis
XGBoost_2	Mokymo	0,9998	0,9960	1,0000	0,9997	0,0070
XGBoost_2	Validacijos	0,9950	0,8584	0,9888	0,9044	0,0225
XGBoost_2	Testavimo	0,9948	0,8508	0,9905	0,9001	0,0220
GBM_4	Mokymo	0,9941	0,8316	0,9945	0,8984	0,0274
GBM_4	Validacijos	0,9930	0,8019	0,9912	0,8673	0,0308
GBM_4	Testavimo	0,9930	0,7981	0,9915	0,8569	0,0305
DRF_1	Mokymo	0,9958	0,8873	0,9979	0,9536	0,0427
DRF_1	Validacijos	0,9904	0,7156	0,9742	0,7687	0,0568
DRF_1	Testavimo	0,9903	0,7101	0,9766	0,7611	0,0555

Šaltinis: sudaryta autoriaus remiantis ketvirtąjo eksperimento rezultatais.

Iš pateiktų metrikų 12 lentelėje, galima matyti, jog XGBoost_2 ir DRF_1 modeliai yra taip pat persimokę, kaip ir 3.3.5.1. paragrafe gautuose rezultatuose. Visų modelių tikslumas yra aukštas, logaritminis nuostolis minimalus. XGBoost_2 F1 balo skirtumas tarp mokymosi ir testavimo imties yra apie 14 %, DRF_1 apie 16 %, o AUC-PR – 9 % ir 19 % atitinkamai, tad modelių persimokymo procentas yra tikrai didelis. GBM_4 modelio skirtumas tarp imčių – tik apie 4 %, tad pagal šiuos abu kriterijus, modelį būtų galima laikyti labai geru. Taip pat šiame modelyje, yra mažiausias skirtumas ir tarp AUC reikšmių, bei logaritminio nuostolio, tad šis modelis, gautas su GBM algoritmu, galėtų būti laikomas geriausiu.

13 lentelėje, yra pateikiami modelio parametrai, kuriuos H2O AutoML įrankis parinko atliekant šį eksperimentą.

Kintamojo, mažų transakcijų vidurkio, parametrai

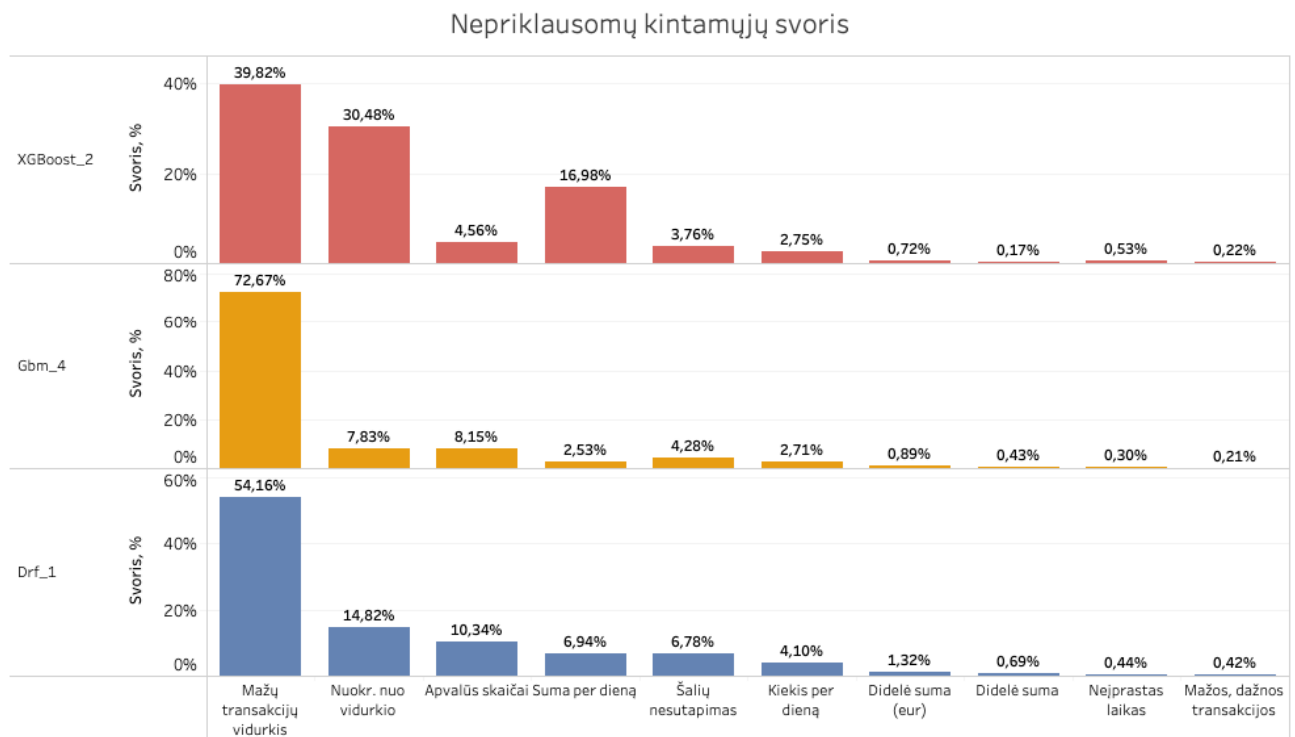
Modelio pav.	Klasių balansavimas	Medžių skaičius	Maks. gylis	Min. eilučių skaičius	Imties dažnis
XGBoost_2	-	420	10	5	0,6
GBM_4	+	130	10	10	0,8
DRF_1	+	35	20	1	0,632

Šaltinis: sudaryta autoriaus remiantis ketvirtąjo eksperimento rezultatais. Pasiskirstymo (angl. *distribution*) reikšmė „bernoulli“ ir skaičių kartotinis (angl. *nfolds*) – „0“.

Kaip ir matoma ir iš praeitų skyrių, modelis, gautas su XGBoost algoritmu, nebuvo balansuojamas, o medžių skaičius – didelis, net 420. Mažiausias medžių skaičius parinktas su DRF algoritmu, o ganėtinai panašus į buvusį prieš tai – su GBM algoritmu, kuris ir buvo geriausias

modelis. Kadangi pats H2O AutoML įrankis neleidžia parinkti medžių skaičiaus, tai modelių rezultatuose dėl to gali atsirasti minimalus skirtumas.

Šiame eksperimente taip pat buvo tikrinamas ir modelyje esančių nepriklausomų kintamųjų svoris. 9 pav. pateikiami gauti procentiniai svoriai iš visų trijų modelių.



Šaltinis: sudaryta autoriaus, remiantis ketvirtojo eksperimento rezultatais, naudojant „Tableau Desktop“ vizualizacijos įrangos paketą.

9 pav. Nepriklausomų kintamųjų svoris modeliui

Iš 9 pav. galima matyti, kad kaip ir praeitame paragrafe, įvestas naujasis papildomas nepriklausomas kintamasis sudaro didelę modelio svorio dalį. Nors ir XGBoost_2 modelis stipriai persimokęs, tačiau jo kintamųjų pasiskirstymas modelyje yra geriausias, nes ne vienas kintamasis sudaro pagrindinį modelio svorį. Su GBM algoritmu, gautas geriausiai veikiantis modelis, kuris visiškai nėra persimokęs ir su geromis modelio metrikomis, tačiau vienas kintamasis sudaro apie $\frac{3}{4}$ modelio svorio. DRF_1 modelis buvo prasto veikimo, bei papildomas kintamasis sudaro $\frac{1}{2}$ modelio svorio, tad modelis gautas su DRF algoritmu, negalėtų būti išrinktas geriausiu. Kaip ir praeitame eksperimente, paskutiniai keturi kintamieji turi mažiausią procentinį svorį.

3.3.5.3. Kintamojo, visa kliento transakcijų suma, reikšmė ir svoris modeliams

Šiame eksperimente, į modelį, kurį sudaro devyni nepriklausomi kintamieji, buvo įvestas papildomas kintamasis, pavadintas „visa kliento transakcijų suma“. 14 lentelėje taip pat pateikiami

tie patys modeliai ir metrikos, kurios buvo aprašomos ir 3.3.5.1. ir 3.3.5.2. paragrafuose. 15 lentelėje pateikiami modelių parametrai, kuriuos parinko H2O AutoML įrankis.

14 lentelė

Kintamojo, visa kliento transakcijų suma, metrikos

Modelio pav.	Imtis	Tikslumas	F1 balas	AUC	AUC-PR	Log. nuostolis
XGBoost_2	Mokymo	1,0000	0,9995	1,0000	1,0000	0,0022
XGBoost_2	Validacijos	0,9987	0,9648	0,9995	0,9901	0,0055
XGBoost_2	Testavimo	0,9984	0,9561	0,9996	0,9882	0,0060
GBM_4	Mokymo	0,9913	0,7549	0,9920	0,8352	0,0400
GBM_4	Validacijos	0,9900	0,7089	0,9856	0,7844	0,0434
GBM_4	Testavimo	0,9900	0,7132	0,9857	0,7810	0,0427
DRF_1	Mokymo	0,9900	0,7212	0,9911	0,8077	0,0574
DRF_1	Validacijos	0,9850	0,5395	0,9488	0,5495	0,0694
DRF_1	Testavimo	0,9850	0,5353	0,9494	0,5453	0,0681

Šaltinis: sudaryta autoriaus remiantis penktojo eksperimento rezultatais.

14 lentelėje, galima matyti, jog XGBoost_2 modelio rezultatai yra šiek tiek geresni nei prieš tai aprašytų eksperimentų. Lentelėje pateikiama, kad šio modelio tikslumas ir abu AUC rodikliai yra maksimalios vertės, logaritminis nuostolis minimalus, tačiau, sprendžiant iš F1 balo, stebimas mažesnis modelio persimokymas (skirtumas tik 4 %). Analizuojant GBM_4 modelį, galima matyti, kad modelio rezultatai nėra tokie geri, tačiau F1 balo procentinis skirtumas sudaro apie 4 %, AUC-PR skirtumas apie 5 %), logaritminis nuostolis mažas, tad modelio metrikos tikrai geros, persimokymas nepastebimas. Kaip ir prieš tai aprašytuose eksperimentuose, modelis DRF_1 veikia prasčiausiai, vis dar matomas stiprus modelio persimokymas ir iš F1 balo, ir iš AUC-PR metrikų skirtumo. Taip pat matomos ir prastos testavimo ir validacijos imčių metrikos.

Toliau 15 lentelėje pateikiami aprašytų modelių parametrai, kuriuos H2O AutoML įrankis parinko atliekant šį eksperimentą.

15 lentelė

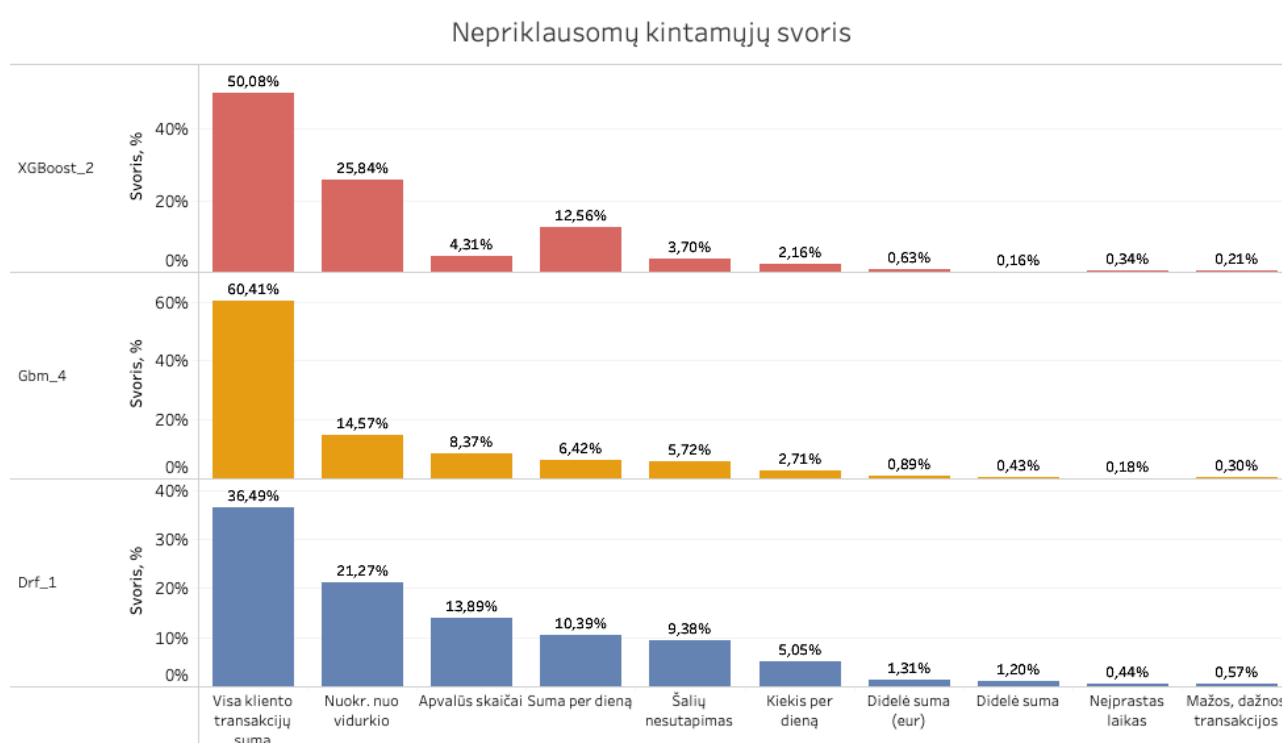
Kintamojo, visa kliento transakcijų suma, parametrai

Modelio pav.	Klasių balansavimas	Medžių skaičius	Maks. gylis	Min. eilučių skaičius	Imties dažnis
XGBoost_2	-	455	10	5	0,6
GBM_4	+	100	10	10	0,8
DRF_1	+	40	20	1	0,632

Šaltinis: sudaryta autoriaus remiantis penktojo eksperimento rezultatais. Visi stulpeliai su vienodomis reikšmėmis: Pasiskirstymo (angl. *distribution*) reikšmė „bernoulli“ ir skaičių kartotinis (angl. *nfolds*) – „0“.

Kaip ir buvo pastebėta praeituose skyriuose, modelis, gautas su XGBoost algoritmu, nebuvo balansuojamas, o medžių skaičius parinktas dar didesnis, kai kiti parametrai išlaikomi tokie patys. Kadangi pats H2O AutoML įrankis neleidžia parinkti medžių skaičiaus, tai modelių rezultatuose dėl to gali atsirasti minimalus metrikų skirtumas ar stipresnis persimokymas, nors šiuo atveju – medžių skaičius didesnis, o modelio persimokymas, sprendžiant iš F1 balo ir AUC-PR reikšmių, priešingai – mažesnis. Geriausias veikimas stebimas vėl su GBM algoritmu, modelio metrikos mažesnis, bet ir naudotas medžių skaičius mažesnis.

Šiame eksperimente taip pat buvo tikrinamas ir modelyje esančių nepriklausomų kintamųjų svoris. 10 pav. pateikiami gauti procentiniai svoriai iš visų trijų modelių.



Šaltinis: sudaryta autoriaus, remiantis penktojo eksperimento rezultatais, naudojant „Tableau Desktop“ vizualizacijos įrangos paketą.

10 pav. Nepriklausomų kintamųjų svoris modeliui

Iš 10 pav., galima matyti, kad kaip ir prieš tai aprašytuose paragrafuose, įvestas naujasis papildomas nepriklausomas kintamasis sudaro didelę modelio svorio dalį. 10 pav. priešingai nei 9 pav., XGBoost_2 modelyje papildomas kintamasis sudaro jau didesnę dalį viso svorio, o kituose modeliuose priešingai – GBM_4 apie 15 % mažesnę svorį, DRF_1 apie 18 %. Modelio metrikos mažesnės, tačiau stebimas geresnis svorių pasiskirstymas. Kaip ir prieš tai aprašytuose eksperimentuose, XGBoost_2 modelyje „suma per dieną“ kintamasis sudaro didesnę svorį nei „apvalūs skaičiai“. Kaip ir praeitame eksperimente, paskutiniai 4 kintamieji turi mažiausią svorį.

3.3.5.4. Kintamojo, visas kliento transakcijų kiekis, reikšmė ir svoris modeliams

Šiame eksperimente buvo nustatomos modelio metrikos, parametrai bei kintamųjų svoriai, kai į modelį su devyniais nepriklausomais kintamaisiais yra įvedamas papildomas nepriklausomas kintamasis, pavadintas „visas kliento transakcijų kiekis“. Modelio metrikos pateikiamos 16 lentelėje.

16 lentelė

Kintamojo, visas kliento transakcijų kiekis, metrikos

Modelio pav.	Imtis	Tikslumas	F1 balas	AUC	AUC-PR	Log. nuotolis
XGBoost_2	Mokymo	0,9999	0,9984	1,0000	0,9998	0,0061
XGBoost_2	Validacijos	0,9935	0,8171	0,9911	0,8815	0,0252
XGBoost_2	Testavimo	0,9934	0,8079	0,9911	0,8732	0,0255
GBM_4	Mokymo	0,9902	0,7214	0,9938	0,8138	0,0329
GBM_4	Validacijos	0,9889	0,6792	0,9885	0,7549	0,0370
GBM_4	Testavimo	0,9887	0,6609	0,9884	0,7415	0,0372
DRF_1	Mokymo	0,9934	0,8166	0,9963	0,8942	0,0448
DRF_1	Validacijos	0,9876	0,6287	0,9721	0,6578	0,0588
DRF_1	Testavimo	0,9878	0,6179	0,9725	0,6575	0,0573

Šaltinis: sudaryta autoriaus, remiantis šeštojo eksperimento rezultatais.

Iš gautų modelio metrikų rezultatų, galima matyti, jog modeliai, su šiuo papildomu nepriklausomu kintamuoju, veikia prasčiausiai, lyginant su visais kitais atliktais eksperimentais. Modeliai išlieka tikslūs, su aukštais AUC rodikliais ir ganėtinai mažu logaritminiu nuostoliu, tačiau kiti kriterijai ženkliai mažesni. Analizuojant modelį XGBoost_2, galima matyti, kad modelis stipriai persimokęs: F1 balo skirtumas tarp mokymosi ir testavimo imčių yra apie 20 %, o AUC-PR apie 12 %. GBM_4 modelio F1 balo procentinis skirtumas yra tik apie 6 %, AUC-PR rodiklio – apie 7 %, kas rodo patenkinamą modelio veikimą, tad modelis tikrai nepretenduoja į geriausiąjį. Kaip ir prieš tai aprašytuose eksperimentuose, modelio, gautas su DRF algoritmu, stebimas prasčiausias veikimas.

17 lentelėje aprašyti modelio parametrai, parinkti H2O AutoML įrankio pagalba.

17 lentelė

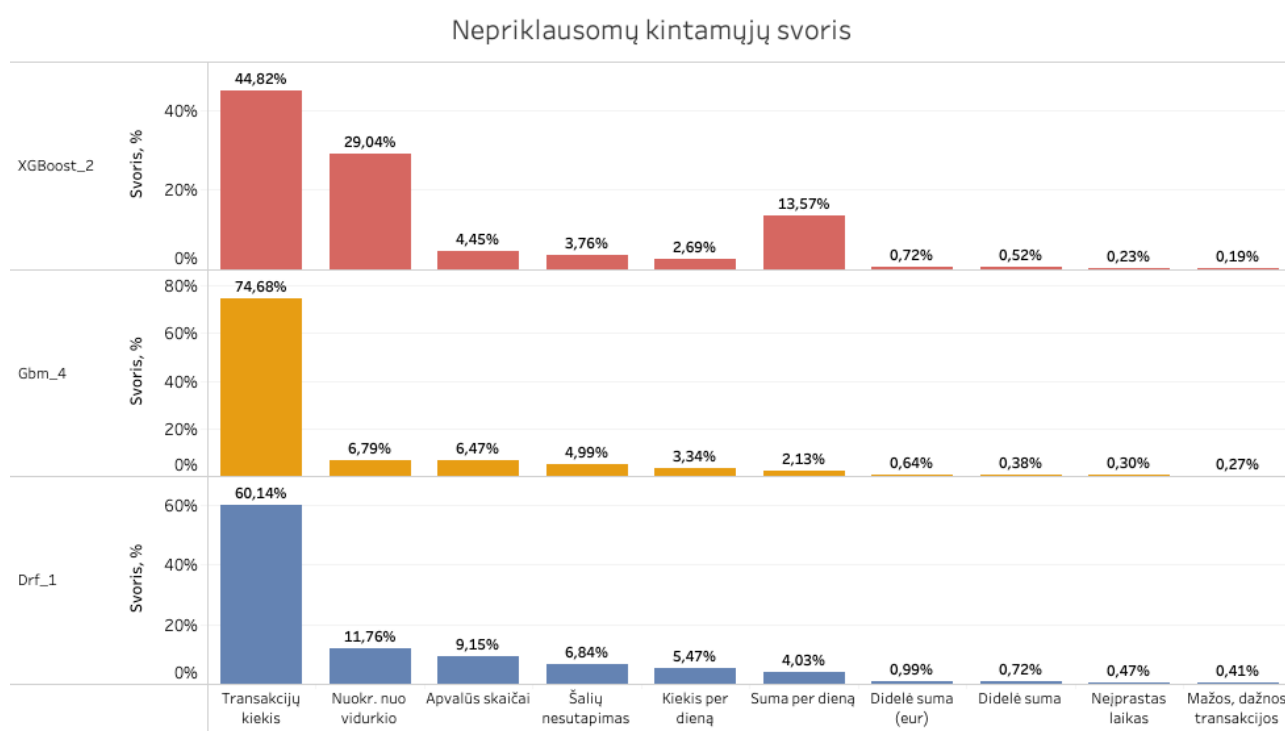
Kintamojo, visas kliento transakcijų kiekis, parametrai

Modelio pav.	Klasių balansavimas	Medžių skaičius	Maks. gylis	Min. eilučių skaičius	Imties dažnis
XGBoost_2	-	505	10	5	0,6
GBM_4	+	125	10	10	0,8
DRF_1	+	40	20	1	0,632

Šaltinis: sudaryta autoriaus, remiantis šeštojo eksperimento rezultatais. Visi stulpeliai su vienodomis reikšmėmis: Pasiskirstymo (angl. *distribution*) reikšmė „bernoulli“ ir skaičių kartotinis (angl. *nfolds*) – „0“.

Kaip ir matoma iš 17 lentelėje pateiktų duomenų, modelis, gautas su XGBoost algoritmu, nebuvo balansuojamas, o medžių skaičius parinktas panašus lyginant lyginant su praeitais eksperimentais, nors modelio persimokymas stebimas didelis. Geriausias veikimas stebimas vėl su GBM algoritmu, kuomet naudoti 125 medžiai, tačiau šio modelio, būtent su šiuo algoritmu, veikimas yra prasčiausias iš visų gautų eksperimentų.

Šiame eksperimente taip pat buvo tikrinamas ir modelyje esančių nepriklausomų kintamųjų svoris. 11 pav. pavaizduoti gauti procentiniai svoriai iš visų trijų modelių.



Šaltinis: sudaryta autoriaus, remiantis šeštojo eksperimento rezultatais, naudojant „Tableau Desktop“ vizualizacijos įrangos paketą.

11 pav. Nepriklausomų kintamųjų svoris modeliui

Iš 11 pav. pavaizduotų rezultatų galima matyti, jog kaip ir praeituose eksperimentuose, papildomas nepriklausomas kintamasis sudaro didžiausią dalį viso modelio svorio. Geriausias kintamųjų svorio pasiskirstymas gautas su XGBoost algoritmu, kadangi dominuoja ne tik vienas nepriklausomas kintamasis, tačiau kaip buvo matoma iš 16 lentelės, kad modelis persimokęs, kaip ir visi kito modeliai gauti su XGBoost algoritmu. Geriausi metrikų rezultatai buvo gauti su GBM algoritmu, tačiau naujasis kintamasis sudaro didžiausią svorį viso modelio svorio. Tas pats matoma ir modelyje, gautame su DRF_1 algoritmu. Kaip ir visuose aprašytuose eksperimentuose, ir šiuo atveju, paskutiniai keturi kintamieji sudarė mažiausią svorį, tad toliau jie bus pašalinti iš modelių.

3.3.5.5. Pasirinktų nepriklausomų kintamųjų eksperimentų apibendrinimas

Apibendrinus visus gautus rezultatus iš atliktų keturių eksperimentų, kuriuose bandoma įsivesti papildomą nepriklausomą kintamąjį, galima teigti, jog aukščiausi modelių metrikų rezultatai pastebėti su XGBoost algoritmu, tačiau visuose eksperimentuose, sprendžiant iš metrikų skirtumo tarp mokymosi, validacijos ir testavimo imčių, stebimas didelis modelių persimokymas. Visuose eksperimentuose, modeliai, kurie gauti su XGBoost algoritmu, duomenų bazė nebuvo balansuojama, o medžių skaičius labai didelis, dėl ko ir gali būti matomas modelių persimokymas. Prasčiausi rezultatai gauti su DRF algoritmu, kuomet gautos ir prastos modelių metrikos, ir taip pastebėtas stiprus modelių persimokymas visuose eksperimentuose. Geriausi rezultatai gauti su GBM algoritmu, kuriuose matoma, kad modelių tikslumas geras (70-80 %), kas pabrėžta ir literatūros apžvalgos skyriuje, gauti modeliai nėra persimokę, nors ir F1 balo bei AUC-PR reikšmės šiek tiek mažesnės. Visuose eksperimentuose stebima, kad metrikų rezultatai tarp validacijos ir testavimo imčių praktiškai nesiskiria ar skiriasi labai minimaliai.

Stebint modelių svorius, geriausias svorių pasiskirstymas matomas, gaunant modelius su XGBoost algoritmu, kuomet ne tik vienas kintamasis turi didžiausią modelio svorį. Priešingai nei su GBM ar DRF algoritmais gauti rezultatai, jų svorių pasiskirstymas prastesnis, dažnai dominuoja vienas papildomas nepriklausomas kintamasis, kuris ir sudaro pagrindinę modelio dalį. Verta taip pat paminėti, jog nustatant nepriklausomų kintamųjų svorius, kintamųjų procentinis svorių išsidėstymas tarp GBM ir DRF algoritmų yra panašus, tačiau modeliuose, gautuose su XGBoost algoritmu, nepriklausomas kintamasis, pavadintas „suma per dieną“ sudaro didesnę procentinę svorį nei kituose modeliuose.

Kadangi mašiniame mokymesi labai aktualu stebėti modelių persimokimą, literatūros apžvalgoje pabrėžta, kad jį nustatant svarbu orientuotis į metrikų visų imčių skirtumus. Kaip ir teigiama, skirtumas neturėtų būtų didesnis nei 5 %, bei metrikų rezultatai turėtų būti bent jau geri ir realūs (apie 80 %). Dėl šių priežasčių, kaip geriausias algoritmas pasirinktas – GBM, nepaisant didelio vieno kintamojo svorio. Iš gautų rezultatų taip pat galima matyti, kad yra du eksperimentai, kuriuose GBM_4 modelis gali būti pasirinktas kaip geriausias, tai modelis, kuriame papildomas nepriklausomas kintamasis buvo „mažų transakcijų vidurkis“ bei „(ne)patvirtintų transakcijų vidurkis“. Modelio su „mažų transakcijų vidurkis“ kintamuoju, F1 balas tarp imčių buvo 79-83 %, AUC-PR – 85-89 %, o su „(ne)patvirtintų transakcijų vidurkis“ tarp 80-85 %, AUC-PR – 90-96 %. Kadangi modelio, gauto su „mažų transakcijų vidurkis“ kintamuoju, svoris keliais procentais buvo mažesnis nei kito minėtojo modelio, mažesnis procentinis skirtumas tarp imčių, tad šis modelis pasirinktas kaip tinkamiausias tolimesniems tyrimams.

3.3.6. Reikšmingų kintamųjų ir IF balo įtaka galutinio modelio pasirinkimui

Kadangi 3.3.5. skyrelyje buvo analizuojama papildomų nepriklausomų kintamųjų įtaka modeliams bei pasirinktas vienas papildomas kintamasis, su kuriuo modeliai turi gerą tikslumą bei mažiausią persimokymo procentą, tad šiame skyrelyje yra pasirenkamas geriausias modelis, kai į jo sudėtį įeina tik reikšmę turintys nepriklausomi kintamieji. Kaip ir buvo minėta literatūros apžvalgoje, naujo nepriklausomo kintamojo, gauto IF algoritmu, įtaka modelio metrikoms turėtų būti teigiama. Šiame skyrelyje atliekami eksperimentai, kuriuose tiriama papildomo naujo kintamojo įtaka modelio metrikoms, parametrams bei svorių pasiskirstymams.

3.3.6.1. Modelio su atrinktais kintamaisiais metrikos, parametrai ir svoris

Kaip ir buvo minėta 3.3.5.5. paragrafe, geriausių metrikų modelis, kuriame nėra didelio persimokymo, gautas kuomet į modelio sudėtį įeina devyni nepriklausomi kintamieji ir įvestas papildomas nepriklausomas kintamasis, pavadintas „mažų transakcijų vidurkis“. Iš visų 3.3.5. skyrelyje aprašytų modelių kintamųjų svorių pavaizdavimo grafikų, galima matyti, kad paskutiniai keturi kintamieji turėjo labai mažą svorį, tad šiame eksperimente jie buvo pašalinti. Modelių tikslumai pateikiami 18 lentelėje.

18 lentelė

Šešių kintamųjų modelių metrikos

Nr.	Modelio pav.	Tikslumas		F1 balas		AUC-PR		Log, nuostolis	
		Mokymo	Testavimo	Mokymo	Testavimo	Mokymo	Testavimo	Mokymo	Testavimo
1.	GBM_1	0,9971	0,9948	0,9218	0,8459	0,9794	0,9010	0,0142	0,0234
2.	XGBoost_3	0,9988	0,9951	0,9693	0,8607	0,9949	0,9076	0,0195	0,0294
3.	GBM_4	0,9942	0,9931	0,8298	0,7904	0,8986	0,8586	0,0280	0,0311
4.	XGBoost_1	0,9999	0,9947	0,9966	0,8466	0,9997	0,8894	0,0067	0,0231
5.	GBM_3	0,9923	0,9914	0,7792	0,7497	0,8384	0,8154	0,0347	0,0361
6.	XGBoost_2	0,9997	0,9948	0,9929	0,8493	0,9992	0,8958	0,0090	0,0235
7.	GBM_2	0,9918	0,9913	0,7539	0,7338	0,8226	0,8044	0,0362	0,0369
8.	DRF_1	0,9980	0,9909	0,9444	0,7274	0,9849	0,7858	0,0361	0,0538
9.	XRT_1	0,9965	0,9902	0,9038	0,7062	0,9579	0,7588	0,0603	0,0681
10.	GLM_1	0,9812	0,9813	0,1495	0,1525	0,0742	0,0744	0,0852	0,0863

Šaltinis: sudaryta autoriaus, remiantis septintojo eksperimento rezultatais.

18 lentelėje pateikiami 10 modelių, kurie gauti H2O AutoML įrankio pagalba. Šioje lentelėje pateikiamos modelio metrikos, gautos iš apmokymo ir testavimo duomenų imčių, o pilni rezultatai ir su validacijos duomenų imtimi pateikiami 3 PRIEDAS. Iš pateiktų rezultatų galima matyti, kad

modelis geriausiai veikia su GBM, XGBoost ir DRF algoritmais, o prasčiausiai su XRT bei GLM. Analizuojant mokymosi imtį, galima matyti, kad geriausi rezultatai pasiekiami su XGBoost algoritmu, tačiau visuose modeliuose stebimas didelis persimokymas, tas pats matoma ir su DRF algoritmu. Modeliai gerai veikia, kai jie gaunami su GBM algoritmu – GBM_4 skirtumas tarp mokymo ir testavimo imčių yra apie 3-4 %, kas rodo, kad modelis persimokymo nepatiria. Lyginant GBM_4 modelio metrikas tarp 12 lentelėje ir 18 lentelėje, galima matyti, kad nepriklausomų kintamųjų pašalinimas iš modelio praktiškai neturėjo įtakos modelio metrikos. Kaip matoma iš 3 PRIEDAS pateiktos lentelės, modelių skirtumas tarp testavimo ir validacijos duomenų imčių yra minimalus, kas pastebima ir visuose prieš tai aprašytuose eksperimentuose.

19 lentelėje pateikiami modelių parametrai, kuriuos H2O AutoML įrankis parinko automatiškai sudarydamas modelius.

19 lentelė

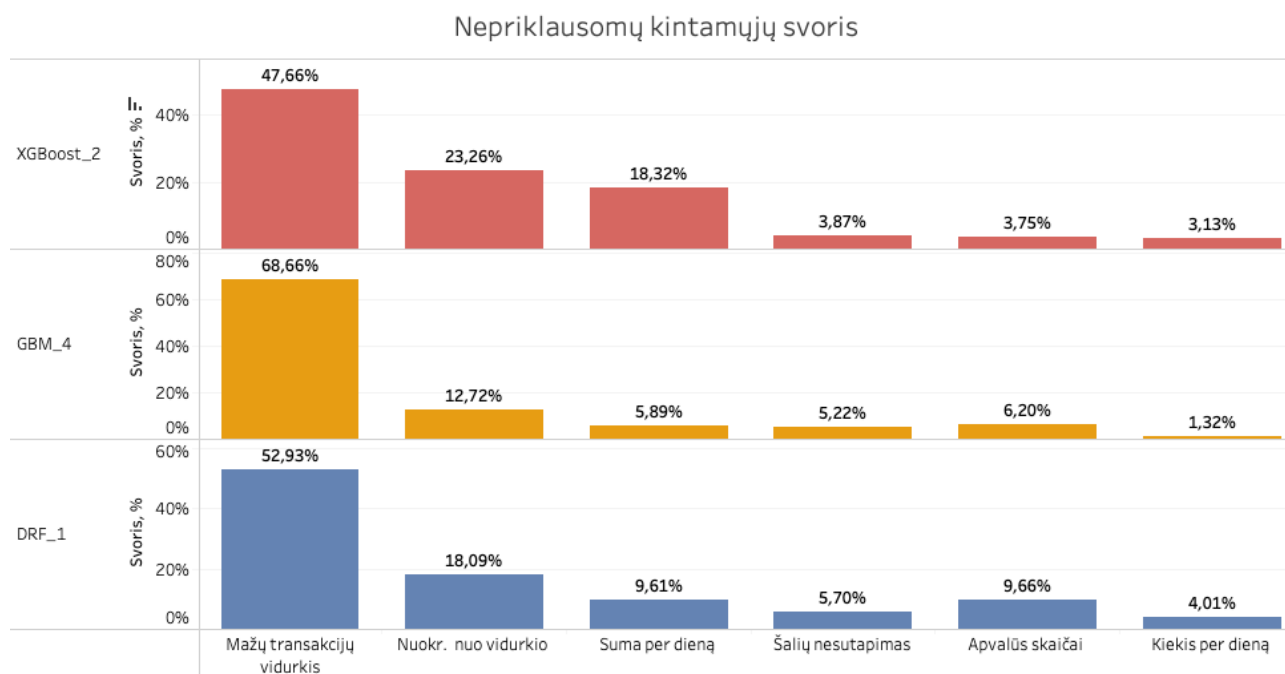
Šešių kintamųjų modelių parametrai

Nr.	Modelio pav.	Klasių balansavimas	Medžių skaičius	Maksimalus gylis	Minimalus eilučių skaičius	Imties dažnis
1.	GBM_1	+	140	15	100	0,8
2.	XGBoost_3	-	1360	5	3	0,8
3.	GBM_4	+	140	10	10	0,8
4.	XGBoost_1	-	295	15	10	0,6
5.	GBM_3	+	185	8	10	0,8
6.	XGBoost_2	-	365	10	5	0,6
7.	GBM_2	+	250	7	10	0,8
8.	DRF_1	+	50	20	1	0,632
9.	XRT_1	+	35	20	1	0,632
10.	GLM_1	+	-	-	-	-

Šaltinis: sudaryta autoriaus, remiantis septintojo eksperimento rezultatais. Visi stulpeliai su vienodomis reikšmėmis: Pasiskirstymo (angl. *distribution*) reikšmė „bernoulli“ (išskyrus 10-oje eilutėje) ir skaičių kartotinis (angl. *nfolds*) – „0“.

Lyginant 19 lentelėje pateiktus GBM_4, XGboost_2 bei DRF_1 modelius, kuriuose pašalinti nereikšmingi kintamieji, su 13 lentelė, galima matyti, kad medžių skaičius tarp XGBoost_2 ir GBM_4 skyrėsi, tačiau metrikos praktiškai ne, tad medžių skaičiaus skirtumas, kurio negalima kontroliuoti su H2O AutoML pagalba, nedaro didelės įtakos modelio metrikoms.

Taip pat buvo tikrinamas ir modelyje esančių nepriklausomų kintamųjų svoris. 12 pav. pavaizduoti gauti procentiniai svoriai iš visų trijų modelių.



Šaltinis: sudaryta autoriaus, remiantis septintojo eksperimento rezultatais, naudojant „Tableau Desktop“ vizualizacijos įrangos paketą.

12 pav. Nepriklausomų kintamųjų svoris modeliui

Iš 12 pav. galima matyti, kad kaip ir prieš tai aprašytuose eksperimentuose, geriausias svorių pasiskirstymas matomas su XGBoost algoritmu, o prasčiausias su GBM. Analizuojant 12 pav., galima matyti, kad visų modelių nepriklausomų kintamųjų svorių išsidėstymas panašus, „apvalių skaičių“ nepriklausomas kintamasis sudaro didesnę procentinę svorį nei „suma per dieną“ kintamasis. Taip pat galima matyti, kad pašalinus paskutinius keturis nepriklausomus kintamuosius, dabar visi kintamieji turi reikšmingą svorį modeliams.

3.3.6.2. IF balo reikšmė ir svoris modeliui

Šiame eksperimente tiriama naujo nepriklausomo kintamojo įtaka modelio metrikoms, parametrų bei kintamųjų svoriams. Naujasis kintamasis gautas neprižiūravimo mašininio mokymosi IF (angl. *Isolation Forest*) algoritmo pagalba. Literatūros analizėje apžvelgiama, kad tokio kintamojo įsivedimas į modelį turėtų duoti didelę naudą modelio metrikoms.

20 lentelėje yra pateikiamos trijų algoritmų modelių metrikos, kurie buvo analizuojami ir praeituose skyreliuose. Eksperimentas buvo atliktas keičiant prognozuotojų skaičių (angl. *n-estimators*) IF algoritme, su tikslu stebėti modelio metrikų pasikeitimą. Lentelėje pateikiamos modelių metrikos, gautos tik ant mokymo ir testavimo duomenų imties, nes rezultatai tarp testavimo ir validacijos imčių praktiškai nesiskyrė.

Modelių, su papildomu IF balo kintamuoju, metrikos

Nr.	Modelio pav.	Tikslumas		F1 balas		AUC-PR		Log, nuostolis	
		Mokymo	Testavimo	Mokymo	Testavimo	Mokymo	Testavimo	Mokymo	Testavimo
50	XGBoost_2	0,9998	0,9945	0,9945	0,8405	0,9993	0,8852	0,0080	0,0242
	GBM_4	0,9949	0,9930	0,8582	0,7971	0,9364	0,8676	0,0228	0,0293
	DRF_1	0,9965	0,9884	0,9081	0,6443	0,9623	0,6773	0,0418	0,0600
100	XGBoost_2	0,9998	0,9945	0,9942	0,8411	0,9993	0,8830	0,0082	0,0247
	GBM_4	0,9948	0,9930	0,8571	0,7969	0,9337	0,8633	0,0232	0,0298
	DRF_1	0,9966	0,9886	0,9096	0,6540	0,9638	0,6791	0,0416	0,0602
140	XGBoost_2	0,9998	0,9945	0,9957	0,8392	0,9995	0,8807	0,0078	0,0248
	GBM_4	0,9952	0,9931	0,8660	0,8026	0,9429	0,8670	0,0218	0,0291
	DRF_1	0,9968	0,9886	0,9152	0,6505	0,9671	0,6859	0,0414	0,0604

Šaltinis: sudaryta autoriaus, remiantis aštuntojo eksperimento rezultatais.

Iš pateiktų duomenų 20 lentelėje, galima matyti, kad kaip ir kituose eksperimentuose, geriausios metrikos ant testavimo duomenų imties gautos su XGBoost algoritmu, tačiau tuo pačiu stebimas ir didelis modelių persimokymas. XGBoost_2 modelių procentinis F1 balo skirtumas tarp mokymosi ir testavimo imčių siekia apie 15 %, AUC-PR – 11 %. Modelių, gautų su DRF algoritmu persimokymo procentas dar didesnis – net apie 25-30 %, kas rodo prastą modelio gebėjimą atskirti sukčiavimo atvejus nuo įprastų klientų.

Priešingai nei kituose eksperimentuose, GBM_4 modelių veikimas prastesnis – metrikos išlieka panašios, tačiau F1 balo skirtumas tarp mokymosi ir testavimo duomenų imties yra apie 6 %, AUC-PR – 7 %, kai ankstesniuose eksperimentuose būdavo apie 3-4 %, tad šiuo atveju modelio metrikos kaip tik suprastėjo, priešingai nei buvo teigiama kitų autorių eksperimentuose. Kadangi šio paragrafo eksperimente buvo keičiami IF algoritmų prognozuotojų skaičiai (pažymėti „Nr.“), galima matyti, jog jų pokytis praktiškai neturėjo įtakos modelio metrikoms, nes visame eksperimente jos išliko beveik vienodos, aukščiausios, kai prognozuotojų skaičius – 140.

21 lentelėje yra pateikiami modelių parametrai, kuriuos H2O AutoML įrankis pats parinko automatiškai. Visi parametrai yra vienodi su prieš tai minėtais eksperimentais, tačiau skiriasi medžių skaičius, kurį pats įrankis pakeisti neleidžia. Nors ir buvo parinkti skirtingi medžių skaičiai, o kai kur vienodi, modelio metrikos išlieka beveik tokios pat visuose atitinkamuose modeliuose, tad minimalus medžių skaičiaus pokytis dėl nepilno įrankio funkcionalumo nesudaro didelės įtakos modelio bendram tikslumui.

Modelių, su papildomu IF balo kintamuoju, parametrai

Nr.	Modelio pav.	Klasių balansavimas	Medžių skaičius	Maksimalus gylis	Minimalius eilučių skaičius	Imties dažnis
50	XGBoost_2	-	420	10	5	0,6
	GBM_4	+	130	10	10	0,8
	DRF_1	+	40	20	1	0,632
100	XGBoost_2	-	420	10	5	0,6
	GBM_4	+	125	10	10	0,8
	DRF_1	+	45	20	1	0,632
140	XGBoost_2	-	420	10	5	0,6
	GBM_4	+	140	10	10	0,8
	DRF_1	+	45	20	1	0,632

Šaltinis: sudaryta autoriaus, remiantis aštuntojo eksperimento rezultatais. Visi stulpeliai su vienodomis reikšmėmis: Pasiskirstymo (angl. *distribution*) reikšmė „bernoulli“ ir skaičių kartotinis (angl. *nfolds*) – „0“.

22 lentelėje yra pateikiami modelių procentiniai svorių pasiskirstymai pagal modelius, gautus su skirtingais algoritmais bei pagal skirtingą prognozuotojų skaičių, kuris nurodomas IF algoritme.

Modelių, su papildomu IF balo kintamuoju, svoriai

Kintamasis/modelis	XGBoost_2			GBM_4			DRF_1		
Prognozuotojų sk.	50	100	140	50	100	140	50	100	140
Mažų trans. vidurkis	34,33	34,43	34,28	46,22	46,01	45,93	35,84	36,26	35,96
ISO balas	16,66	17,93	17,63	24,09	24,79	24,43	27,69	28,32	28,34
Nuokr. nuo vidurkio	26,58	25,27	26,36	13,18	13,00	13,63	13,04	12,34	12,43
Suma per dieną	11,41	11,28	10,74	5,73	5,54	5,51	7,40	7,16	7,37
Apvalūs skaičiai	4,29	4,24	4,27	4,49	4,32	4,24	7,22	7,05	7,01
Šalių nesutapimas	3,70	3,84	3,78	3,15	3,25	3,29	4,37	4,36	4,30
Kiekis per dieną	3,02	3,01	2,94	3,13	3,09	2,97	4,42	4,51	4,59

Šaltinis: sudaryta autoriaus, remiantis aštuntojo eksperimento rezultatais.

Iš pateiktos 22 lentelės, galima matyti, kad prognozuotojų skaičius modelių nepriklausomiems kintamiesiems įtakos neturi, nes visų modelių kintamųjų procentinis svoris praktiškai vienodas. Nors ir IF balo įvedimas į modelį, jų metrikas minimaliai pablogino, tačiau turėjo labai teigiamos įtakos nepriklausomų kintamųjų pasiskirstymui. Kaip galima matyti iš 22 lentelės, visuose modeliuose naujas papildomas kintamasis užima antrąją vietą pagal svorių pasiskirstymą. Šis kintamasis labai didelę įtaką turėjo GBM_4 modeliui, kuriame buvo gautos geros modelio metrikos su sąlyginai mažu

modelio persimokymo procentu. Prieš tai, be šio papildomo kintamojo, „mažų transakcijų vidurkis“ sudarė 72,67 % (9 pav.), o dabar – apie 46 %.

3.3.6.3. Galutiniai modeliai, jų tikslumas, parametrai ir svoris

Visuose prieš tai aprašytuose eksperimentuose, buvo išskiriami trys pagrindiniai modeliai: XGBoost_2, GBM_4 ir DRF_1. GBM_4 modelis visada pasižymėjo mažiausiu persimokymo procentu, kas yra labai svarbu mašininiam mokymesi. Kaip buvo matomai iš 3.3.6.2. paragrafe atlikto eksperimento, IF balo pridėjimas į modelį turėjo labai teigiamą įtaką procentiniam nepriklausomų kintamųjų pasiskirstymui, tačiau geriausio modelio (GBM_4) persimokymo laipsnis padidėjo nuo 3 % iki 6 %, kas jau rodo minimalų modelio persimokymą. Šiuo tikslu buvo atliekamas dar vienas eksperimentas, kada parodomo visi 10 gautų modelių, kai į modelio sudėtį įeina 7 kintamieji, kai vienas iš jų – IF balas. Modelio metrikos aprašytos 23 lentelėje.

23 lentelė

Galutinių modelių metrikos

Nr.	Modelio pav.	Tikslumas		F1-Balas		AUC-PR		Log, nuostolis	
		Mokymo	Testavimo	Mokymo	Testavimo	Mokymo	Testavimo	Mokymo	Testavimo
1.	GBM_1	0,9978	0,9942	0,9421	0,8310	0,9886	0,8866	0,0126	0,0268
2.	GBM_4	0,9952	0,9931	0,8660	0,8026	0,9429	0,8670	0,0218	0,0291
3.	GBM_3	0,9930	0,9920	0,8060	0,7715	0,8824	0,8365	0,0293	0,0327
4.	XGBoost_3	0,9990	0,9947	0,9742	0,8496	0,9959	0,8966	0,0188	0,0300
5.	GBM_2	0,9922	0,9915	0,7859	0,7505	0,8594	0,8226	0,0321	0,0344
6.	XGBoost_2	0,9998	0,9945	0,9957	0,8392	0,9995	0,8807	0,0078	0,0248
7.	XGBoost_1	0,9999	0,9943	0,9971	0,8350	0,9997	0,8776	0,0063	0,0245
8.	DRF_1	0,9968	0,9886	0,9152	0,6505	0,9671	0,6859	0,0414	0,0604
9.	XRT_1	0,9914	0,9865	0,7605	0,5795	0,8608	0,6192	0,0557	0,0662
10.	GLM_1	0,9812	0,9813	0,1956	0,2004	0,1103	0,1165	0,0817	0,0805

Šaltinis: sudaryta autoriaus, remiantis devintojo eksperimento rezultatais. Prognozuotojų skaičius – 140.

Iš gautų metrikų rezultatų, pateiktų 23 lentelėje, galima matyti, kad XGboost modeliai pirmąją savo metrikų rezultatais, kaip ir visuose eksperimentuose, tačiau stebimas didelis persimokymo procentas. Tas pats matomas ir su DRF modeliu, XRT modelio tikslumas vidutiniškas, o GLM – prastas. Modelių metrikos tarp mokymosi, validacijos ir testavimo imčių pateikiamos 4 PRIEDAS. Modeliai, gauti su GBM algoritmu, veikia geriausiai, su mažiausiu persimokymo laipsniu sprendžiant iš F1 balo, tačiau pokytis matomas lyginant AUC-PR reikšmes, kai kur skirtumas tarp imčių – net 10 %. Nors ir visuose eksperimentuose, buvo nagrinėjamas GBM_4 modelis, jo metrikos

suprastėjo įvedus papildomą kintamąjį, o iš šių aprašytų 10 modelių geriausiai veikia kitas GBM algoritmu gautas modelis – GBM_3. Šio modelio F1 balo reikšmė mokymosi imtyje yra apie 80 %, o testavimo – 77 %, kas rodo gerą modelio gebėjimą atskirti sukčiavimo atvejus nuo ne sukčiavimo, AUC-PR reikšmė aukšta su mažu skirtumu tarp testavimo ir mokymosi imties, tad modelio persimokymo laipsnis taip pat mažas. Priešingai nei su GBM_4 modelio, šio modelio metrikos, pridėjus IF balo kintamąjį – suprastėjo, F1 balo skirtumas tapo 6 % (iš buvusių 3 % (18 lentelė)), bei jo AUC-PR reikšmė stipriai pasikeitė į blogąją pusę, kas rodo didelį modelio persimokymą.

24 lentelėje aprašyti, parametrai, kuriuos H2O AutoML įrankis parinko automatiškai atliekant šį eksperimentą. GBM_4 ir GBM_3 tarp tarpusavyje vienodų modelių parametrai beveik nesiskyrė, lyginant su atliktu eksperimentu, aprašytu 19 lentelėje, tačiau parametrai tarp GBM_3 ir GBM_4, tokie kaip maksimalus gylis bei medžių skaičius skyrėsi, kas ir galėjo lemto šį skirtumą.

24 lentelė

Galutinių modelių parametrai

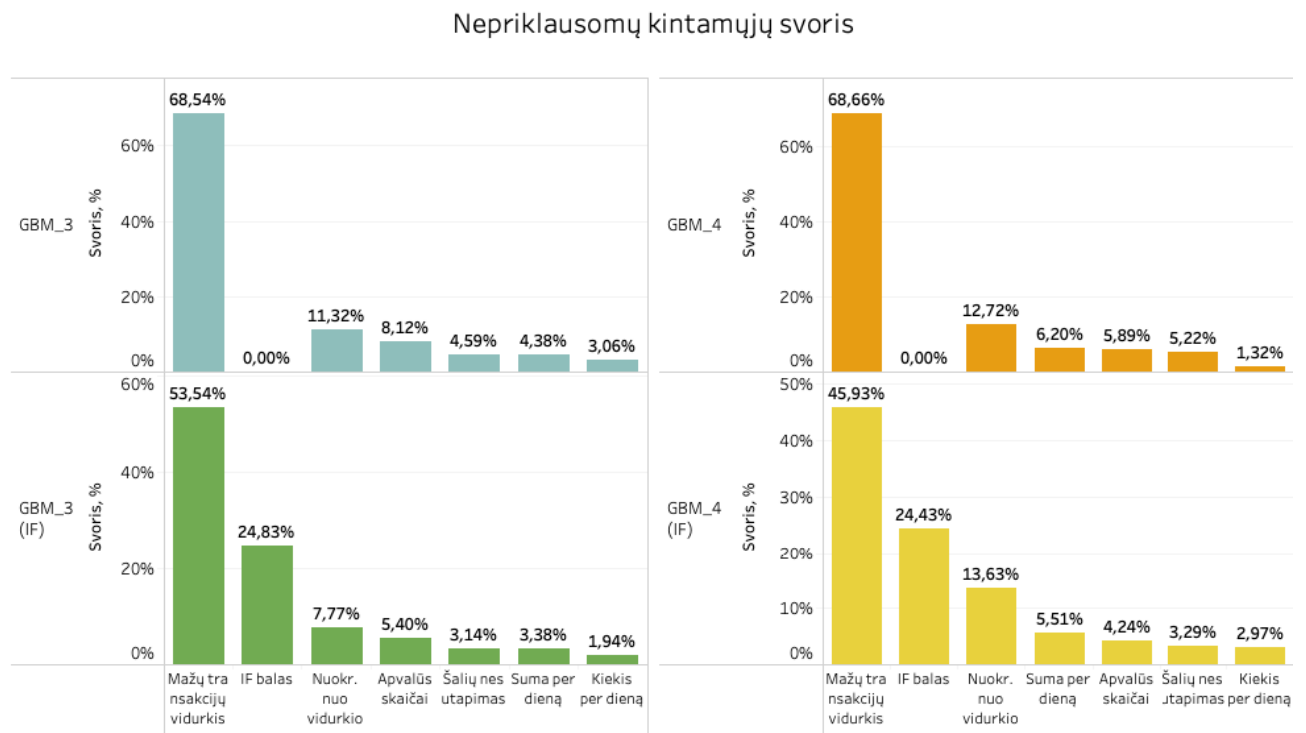
Nr.	Modelio pav.	Klasių balansavimas	Medžių skaičius	Maksimalus gylis	Minimalius eilučių skaičius	Imties dažnis
1.	GBM_1	+	130	15	100	0,8
2.	GBM_4	+	140	10	10	0,8
3.	GBM_3	+	180	8	10	0,8
4.	XGBoost_3	-	1365	5	3	0,8
5.	GBM_2	+	240	7	10	0,8
6.	XGBoost_2	-	420	10	5	0,6
7.	XGBoost_1	-	315	15	10	0,6
8.	DRF_1	+	45	20	1	0,632
9.	XRT_1	+	50	20	1	0,632
10.	GLM_1	+	-	-	-	-

Šaltinis: sudaryta autoriaus remiantis devintojo eksperimento rezultatais. Visi stulpeliai su vienodomis reikšmėmis: Pasiskirstymo (angl. *distribution*) reikšmė „bernoulli“ (išskyrus 10-oje eilutėje) ir skaičių kartotinis (angl. *nfolds*) – „0“.

13 pav. yra pavaizduotas procentinis svorių pasiskirstymas GBM_3 ir GBM_4 modeliuose, bei IF balo naudo modelių svorių pasiskirstymui.

Neįvedus IF balo nepriklausomo kintamojo, galima matyti, kad procentinis svorių pasiskirstymas tarp abiejų modelių yra panašus ir labai didelis, dominuoja vienas nepriklausomas kintamasis. GBM_3 modelyje mažesnę svorį turi du paskutiniai kintamieji, nei kad GBM_4. Įvedus IF balo kintamąjį, abiejų modelių svorių pasiskirstymas pagerėjo, abėjuose modeliuose jis sudarė panašų procentinį svorį, tačiau GBM_3 modelyje, didesnę „mažų transakcijų vidurkis“ svorį nei

GBM_4 modelyje, kas rodo prastesnį svorių pasiskirstymą ir vis dar visišką vieno kintamojo dominavimą.



Šaltinis: sudaryta autoriaus, remiantis devintojo eksperimento rezultatais, naudojant „Tableau Desktop“ vizualizacijos įrangos paketą.

13 pav. Nepriklausomų kintamųjų svoris modeliui

3.3.6.4. Galutinio modelio pasirinkimo analizė

Apibendrinus 3.3.6. skyrelio rezultatus, matoma, kad visą laiką laikyto geriausio modelio (GBM_4) metrikos suprastėjo, kai lyginama metrikų, tokių kaip F1 balo bei AUC-PR reikšmės bei jų pokyčius tarp mokymosi, validacijos bei testavimo imčių (3 PRIEDAS ir 4 PRIEDAS). Nors ir procentinis svorių pasiskirstymas GBM_4 modelio yra geresnis, nei GBM_3, GBM_4 modelis tapo persimokęs, kai iki „IF balo“ įvedimo jis veikė labai gerai. Priešingai nei su GBM_4 modeliu, „IF balo“ įvedimas į GBM_3 šiek tiek pagerino modelio metrikas (nuo 74 % iki 77 % testavimo imtyje), išliko mažas skirtumas tarp F1 balo bei AUC-PR reikšmių.

Šis metrikų pasikeitimas galėjo būti dėl kelių priežasčių – vieną iš jų, tai, kad parametrai, tokie kaip medžių skaičius bei maksimalus medžių gylis skiriasi, tačiau tarp GBM_3 ir GBM_4 tarpusavyje vienodų modelių šie parametrai beveik nesiskyrė (19 lentelė ir 24 lentelė), tad neturėtų turėti tokios didelės įtakos ir atsakyti į klausimą, kodėl vieno modelio metrikos pagerėjo, o kito – suprastėjo.

Kitas svarbus aspektas, kuris galėjo turėti didelę įtaką tokiems gautiems rezultatams, tai yra procentinis svorių pasiskirstymas. Kaip ir matoma iš visų atliktų eksperimentų, procentinis svorių pasiskirstymas (dar papildomai nepridėjus „IF balo“ nepriklausomo kintamojo), visada buvo geriausias, tačiau buvo stebimas žymus modelių persimokymo procentas. Ta pati situacija matoma ir 3.3.6.3. paragrafe, kai įvedus „IF balo“ nepriklausomą kintamąjį, modelio GBM_4 procentinis svorių pasiskirstymas tapo geras, kaip ir XGBoost algoritmo pagalba gautų modelių, o persimokymo laipsnis – padidėjo. Priešingai nei su GBM_3, svorių pasiskirstymas prastesnis, o persimokymo beveik nėra. Iš gautų rezultatų galbūt būtų galima daryti prielaidą, kad vieno nepriklausomo kintamojo dominavimas stabdo modelį nuo persimokymo. Tai būtų galima matyti ir visuose atliktuose eksperimentuose, GBM algoritmo modeliai visada veikdavo geriausiai pagal F1 balą bei AUC-PR reikšmes, nors juose visada stipriai dominuodavo vienas nepriklausomas kintamasis.

Atsižvelgiant į tai, kad modelis GBM_4 turi didesnę skirtumą tarp F1 balo bei didelį skirtumą tarp AUC-PR, kas rodo modelio gebėjimą atskirti sukčiavimo atvejus nuo įprastų klientų, geriausiu modeliu geriau būtų laikyti GBM_3 modelį, nors ir vienas kintamasis sudaro pusę viso modelio svorio.

Nors ir literatūroje minimų autorių teigimu, nepriklausomo kintamojo, gauto IF algoritmu pridėjimas į modelį jiems labai pagerino modelio metrikas, atlikti eksperimentai rodo, kad modelio metrikos gali tik minimaliai pagerėti, o kitais atvejais net ir suprastėti, jei galimai stipriai pasikeičia modelių procentinis svorių išsidėstymas. Tačiau, visais atvejais, IF balo pridėjimas į modelį, labiau stabilizavo procentinį svorių pasiskirstymą.

25 lentelėje pavaizduotas geriausio modelio – GBM_3 sumaišties matrica (angl. *confusion matrix*), rodanti modelio gebėjimą prognozuoti sukčiavimo atvejus bei klaidų balą.

25 lentelė

Geriausio galutinio modelio sumaišties matrica

Reikšmė	0	1	Klaidų balas
0	96 472 (TN)	337 (FP)	0,0035
1	474 (FN)	1 369 (TP)	0,2572
Iš viso	96 946	1 706	0,0082

*TN – tikrai neigiamas, TP – tikrai teigiamas, FN – klaidingai neigiamas, FP – klaidingai teigiamas reikšmės.

Šaltinis: sudaryta autoriaus remiantis paskutinio eksperimento rezultatais.

Iš gautų eksperimento rezultatų 25 lentelėje, galima matyti, kad modelis nustatė 96 472 prognozes, kurios yra įprastos transakcijos, iš jų 337 buvo nustatyta kaip sukčiavimas, kas buvo klaidingai teigiamas nustatymas, todėl modelio klaidų balas siekia 0,035 %. Taip pat, modelis nustatė 1 369 sukčiavimo atvejus, iš kurių 474 buvo iš tikro nebuvo sukčiavimas, tad klaidų balas yra lygus

25 %, kas rodo, kad modelis veikia gerai, bet tikrai dar galima jį tobulinti. Sprendžiant iš gautų rezultatų, modelis yra pakankamai geras, gerai prognozuoja sukčiavimą.

Apibendrinant sukčiavimo atvejų nustatymo automatizavimo eksperimentinį skyrių, galima teigti, kad vienas iš svarbiausių dalykų, leidžiančių gauti gerų metrikų modelį, yra nepriklausomi kintamieji, sudaryti pagal turimus duomenis. Šiame skyriuje sukčiavimo atvejų nustatymo automatizavimas buvo atliekamas prižiūrimo mokymosi pagalba gautu įrankiu – H2O AutoML. Kuriant modelius, buvo atrenkami nepriklausomi kintamieji, stebint mažiausią persimokymo procentą tarp mokymosi, validacijos ir testavimo imčių. Tikrinant modelius, kai jie gaunami su visais turimais nepriklausomais kintamaisiais, modelių rezultatai rodė stiprų persimokymą. Geriausių metrikų modelis gautas tada, kai į modelį su penkiais nepriklausomais kintamaisiais papildomai įvestas naujas kintamasis – „mažų transakcijų vidurkis“, tačiau pastebėtas prastas svorių pasiskirstymas. Šiame skyriuje taip pat buvo tirta naujo nepriklausomo kintamojo įtaka modelio metrikoms ir svorių pasiskirstymams. Gauti rezultatai ir jų analizė parodė, kad naujo kintamojo įvedimas į modelį nekeičia arba minimaliai keičia modelių metrikas, tačiau turi teigiamą įtaką procentiniam svorių pasiskirstymui. Gauti rezultatai taip rodo, kad kuo geresnis svorių pasiskirstymas, stebimas didesnis modelių persimokymas, taip darant prielaidą, kad vieno kintamojo dominavimas gali stabdyti modelius nuo persimokymo. Geriausias modelių veikimas stebimas su GBM algoritmu.

IŠVADOS IR PASIŪLYMAI

1. Atlikus sukčiavimo atvejų nustatymo automatizavimo literatūros apžvalgą, nustatyta, kad pagal tipus sukčiavimas skirstomas į švelnų, kriminalinį ir sunkų, o pagal kategorijas į masinės rinkodaros, investicinius, tapatybės bei kitus sukčiavimus. Kad nustatyti sukčiavimą, reikia gerai išanalizuoti turimą duomenų bazę, pritaikyti tinkamiausią algoritmą bei aktualius nepriklausomus kintamuosius. Sukčiavimo atvejai dažniausiai nustatoma remiantis mašininio mokymosi pagalba, naudojant prižiūrimus arba neprižiūrimus mokymosi metodus.
2. Sudaryti tiriamojo darbo atlikimo etapai ir metodai, pagal kuriuos pirmiausia, su programiniu duomenų paruošimo įrankiu „Tableau Prep“, sutvarkomi turimi duomenys, gauti iš viešai prieinamos duomenų bazės. Atliekami įvairūs eksperimentai, naudojant H2O AutoML įrankį, tikrinama naujo kintamojo, gauto neprižiūrimo mokymosi metodu įtaka modeliams bei pasiūlomas galutinis, realiausiai veikiantis modelis automatizuotam sukčiavimo atvejų nustatymui.
3. Eksperimento būdu nustatyta, kad modelyje turint daug didelę įtaką modeliui darančių nepriklausomų kintamųjų, modeliai linkę persimokyti. Papildomas kintamasis „mažų transakcijų vidurkis“ buvo reikšmingas kintamasis iš keturių tikrintų, su kuriuo geriausias modelis, gautas su GBM algoritmu, turėjo mažiausią persimokymo procentą, realias metrikas (F1 balas tarp imčių buvo 79-83 %, AUC-PR – 85-89 %), tačiau pagal procentinius svorių pasiskirstymus stebimas visiškas vieno papildomo kintamojo dominavimas.
4. Papildomo nepriklausomo kintamojo, gauto IF algoritmu, įtaką modelio metrikoms buvo minimali – vienais atvejais minimaliai pagerėjo, kitais suprastėjo, tačiau pastebėtas ženkliai pasikeitę nepriklausomų kintamųjų procentiniai svoriai. Šio kintamojo pridėjimas į modelį lemdavo gerą svorių pasiskirstymą ir ne visišką vieno kintamojo dominavimą.
5. Geriausias automatizuotas modelis, nustatantis sukčiavimo atvejus, kurį siūloma naudoti, yra gautas, kai į modelio sudėtį įeina septyni nepriklausomi kintamieji (vienas iš jų „IF balas“), naudojamas GBM algoritmas, duomenys balansuojami, medžių skaičius – 185, maksimalus gylis – 8, o mėginių dažnis – 0,8. Stebimas ir normalus procentinis svorių pasiskirstymas, kur visi nepriklausomi kintamieji turi reikšmingą svorį. Modelis, su pasirinktu algoritmu, prognozavo 96 472 įprastas reikšmes, ir 1 642 kaip sukčiavimo atvejus, klaidų balas siekia 25 %.

LITERATŪRA

1. Abbassi, H., Berkaoui, A., Elmendili, S., ir Gahi, Y. (2023). End-to-End Real-time Architecture for Fraud Detection in Online Digital Transactions. *International Journal of Advanced Computer Science and Applications*, 14(6), 749-757.
<https://doi.org/10.14569/IJACSA.2023.0140680>
2. Abdallah, A., Maarof, A. M., ir Zainal, A. (2016). Fraud detection system: A survey. *Journal of Network and Computer Applications*, 68, 90-113.
<https://doi.org/10.1016/j.jnca.2016.04.007>
3. Abdallah, S. Z., Gaber, M. M., Srimivasan. B., Krishnaswamy, S. (2016). AnyNovel: detection of novel concepts in evolving data streams. *Evolving Systems*, 7, 73–93.
<https://doi.org/10.1007/s12530-016-9147-7>
4. ACFE (2002). *2002 Report to the Nation Occupational Fraud and Abuse*. Prieiga per: <https://www.acfe.com/-/media/files/acfe/pdfs/2002rttn.pdf>
5. Alexopoulos, P., Kafentzis, K., Benetou, X., Tagaris, T., Georgolios, P. (2007). Towards a generic Fraud ontology in E-government. *Proceedings of the Second International Conference on e-Business*, 1, 269-276. <https://doi.org/10.5220/0002112602690276>
6. Amit, G., ir Gur, S. (2024). eXplainable Random Forest. IBM Research, *Haifa*. 1-15. Prieiga per https://ceur-ws.org/Vol-3765/Camera_Ready_Paper-10.pdf
7. Aragao, M. V. C., Afonso, A. G., Ferraz, R. C., Feirreira, R. G., Leite, S., de Figueiredo, F. A. P., ir Mafra, S. B. (2024). A practical evaluation of AutoML tools for binary, multiclass, and multilabel classification. *Research Square*, 1-44. <https://doi.org/10.21203/rs.3.rs-4172933/v1>
8. Aziz, Z. Z. A. ir Md Daud, S. A. M. (2022). Customer's awareness, trust, discomfort and acceptance of anti-money laundering practices in Malaysian Banks. *Journal of Money Laundering Control*, 25(4), 864-881. <https://doi.org/10.1108/JMLC-08-2021-0087>
9. Bhowmik, R. (2011). Detecting Auto Insurance Fraud by Data Mining Techniques. *Journal of Emerging Trends in Computing and Information Sciences*, 2(4), 156-162. Prieiga per <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=45e6de6fbff3ccf40e93941c72f12e14bf8e5747>
10. Barkved, K. (2022). *How To Know if Your Machine Learning Model Has Good Performance*. Prieiga per <https://www.obviously.ai/post/machine-learning-model-performance>

11. Bolton, R. J., ir Hand, D.J. (2002). Unsupervised Profiling Methods for Fraud Detection. Conference on Credit Scoring and Credit Control. Prieiga per <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=5b640c367ae9cc4bd072006b05a3ed7c2d5f496d>
12. Button, M., Lewis, C., ir Tapley, J. (2009). Fraud typologies and victims of fraud. Centre for Counter Fraud Studies, Institute of Criminal Justice Studies, University of Portsmouth. 1-40. Prieiga per <https://assets.publishing.service.gov.uk/media/5a7ad8c2ed915d670dd7efad/fraud-typologies.pdf>
13. Carcillo, F., Le Borgne, Y.A., Caelen, O., Kessaci, Y., Oble, F., ir Bontempi, G. (2021). Combining unsupervised and supervised learning in credit card fraud detection. *Information Sciences*, 557, 317–331. <https://doi.org/10.1016/j.ins.2019.05.042>
14. Carcillo, F., Pozzolo, D. A., Borgne, L. E., Caelen, O., Mazzer, Y., ir Bontempi, G. (2018). SCARFF: *A scalable framework for streaming credit card fraud detection with spark*. *Information Fusion*, 41, 182-194. <https://doi.org/10.1016/j.inffus.2017.09.005>
15. Chandola, V., Banerjee, A., ir Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys*, 41(3), 1-58. <https://doi.org/10.1145/1541880.1541882>
16. Chitimira, H. ir Munedzi, S. (2023). Overview international best practices on customer due diligence and related anti-money laundering measures", *Journal of Money Laundering Control*, 26(7), 53-62. <https://doi.org/10.1108/JMLC-07-2022-0102>
17. Czakon, J. (2024). *F1 Score vs ROC AUC vs Accuracy vs PR AUC: Which Evaluation Metric Should You Choose?* Prieiga per <https://neptune.ai/blog/f1-score-accuracy-roc-auc-pr-auc>
18. Čampulova, M., Michalek, J., ir Moucka, J. (2019). Generalised linear model-based algorithm for detection of outliers in environmental data and comparison with semi-parametric outlier detection methods. *Atmospheric Pollution Research*, 10(4), 1015-1023. <https://doi.org/10.1016/j.apr.2019.01.010>
19. DataScience-ProF (2024). *Understanding Log Loss: A Comprehensive Guide with Code Examples*. Prieiga per <https://medium.com/@TheDataScience-ProF/understanding-log-loss-a-comprehensive-guide-with-code-examples-c79cf5411426>
20. E-seimas (1997). *Republic of Lithuania Law on the Prevention of Money Laundering and Terrorist Financing*. Prieiga per <https://e-seimas.lrs.lt/portal/legalAct/lt/TAD/2c647332ba5111eb91e294a1358e77e9?jfwid=twcznlk4w>

21. EU Monitor (2024). *European Anti-Fraud Office (OLAF)*. Prieiga per <https://www.eumonitor.eu/9353000/1/j9vvik7m1c3gyxp/vh77kouqbwrr>
22. Ferreira, L., Pilastri, A., Martins, C., Santos, P., ir Cortez, P. (2020). An Automated and Distributed Machine Learning Framework for Telecommunications Risk Management. *Proceedings of the 12th International Conference on Agents and Artificial Intelligence*, 2, 99-107. <https://doi.org/10.5220/0008952800990107>
23. H2O.ai (2024). H2O AutoML: *Automatic Machine Learning*. Prieiga per <https://docs.h2o.ai/h2o/latest-stable/h2o-docs/automl.html>
24. Yousif, S.A., ir Samawi, V.W. (2024). The Effectiveness of Using AutoML in Electricity Theft Detection: The Impact of Data Preprocessing and Balancing Techniques. *Computational Science and Its Applications – ICCSA 2024*, 14814, 68-82. https://doi.org/10.1007/978-3-031-64608-9_5
25. Jin, H., Song, Q., ir Hu, X. (2018). Auto-Keras: An Efficient Neural Architecture Search System. *ArXiv*, 1-11. <https://doi.org/10.48550/arXiv.1806.10282>
26. Khan, A. A., Dwivedi, P., Mugde, S., Sajidha, S. A., Sharma, G., ir Soni, G. (2023). Chapter 8 - Toward automated machine learning for genomics: evaluation and comparison of state-of-the-art AutoML approaches. *Data Science for Genome: Academic Press*, 129-152.
27. Khoshgoftaar, T. M., ir Allen, E. B. (2001). Controlling Overfitting in Classification-Tree Models of Software Quality. *Empirical Software Engineering*, 6, 59-79. <https://doi.org/10.1023/A:1009803004576>
28. Kotthoff, L., Thornton, C., Hoos, H. H., Hutter, F., ir Leyton-Brown, K. (2017). Auto-WEKA 2.0: Automatic model selection and hyperparameter optimization in WEKA. *Journal of Machine Learning Research*, 18, 1-5. Prieiga per <https://www.jmlr.org/papers/volume18/16-261/16-261.pdf>
29. Kurum, E. (2023). RegTech solutions and AML compliance: what future for financial crime?. *Journal of Financial Crime*, 30(3), 776-794. Prieiga per <https://doi.org/10.1108/JFC-04-2020-0051>
30. Lietuvos Bankas (2024). *Sukčiavimo prevencijos gairės*. Prieiga per https://www.lb.lt/uploads/documents/docs/44226_77a36df7cd0c350012778525034c87fa.pdf
31. Ma, J., Jiang, S., Liu, Z., Ren, Z., Lei, D., Tan, C., ir Guo, H. (2022). Machine Learning Models for Slope Stability Classification of Circular Mode Failure: An Updated Database and Automated Machine Learning (AutoML) Approach. *Sensors*, 22(23), 1-29. <https://doi.org/10.3390/s22239166>

32. Ma, C., ir Lee, C. H., (2008). An efficient gradient computation approach to discriminative fusion optimization in semantic concept detection. *19th International Conference on Pattern Recognition, Tampa, FL, USA*. 1-4, <https://doi.org/10.1109/ICPR.2008.4761471>
33. Mienye, D., ir Swart, T. (2024). A Hybrid Deep Learning Approach with Generative Adversarial Network for Credit Card Fraud Detection. *Technologies*, 12(10), 1-17. <https://doi.org/10.3390/technologies12100186>
34. Niu, X., Wang, L., ir Yang, Y. (2019). A Comparison Study of Credit Card Fraud Detection: *Supervised versus Unsupervised*. ArXiv, 1-8. <https://doi.org/10.48550/arXiv.1904.10604>
35. Njotini, M. N. (2009) *The verification and exchange of customer due diligence (CDD) data in terms of the Financial Intelligence Centre Act 38 of 2001*. Prieiga per <https://uir.unisa.ac.za/handle/10500/3202>
36. Ogbemi, M. E., (2023). *What is Overfitting in Machine Learning?* Prieiga per <https://www.freecodecamp.org/news/what-is-overfitting-machine-learning/>
37. Oza, A. (2018). *Fraud Detection using Machine Learning*. Prieiga per <https://cs229.stanford.edu/proj2018/report/261.pdf>
38. Peteiro-Barral, D., ir Guijarro-Berdinas, B. (2012). A survey of methods for distributed machine learning. *Progress Artificial Intelligence*, 2, 1–11. <https://doi.org/10.1007/s13748-012-0035-5>
39. Philip, N., ir Sherly, K. K. (2012). Credit Card Fraud Detection Based on Behaviour Mining. *International Journal of Scientific and Technological Research*, 1(1), 7-12. Prieiga per https://www.researchgate.net/publication/303215980_Credit_Card_Fraud_Detection_Based_on_Behaviour_Mining
40. Rajput, V. U. (2013). Research on Know Your Customer (KYC). *International Journal of Scientific and Research Publications*, 3(7), 1-6. Prieiga per <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=c24ae816b255a550328de0f6f6c6d344739f219d>
41. Regression Metrics (2024). Geeks for geeks. Prieiga per <https://www.geeksforgeeks.org/regression-metrics/>
42. Roberts, A. (2022). *What is PR AUC?* Prieiga per <https://arize.com/blog/what-is-pr-auc/>
43. Roy, A., Sun, J., Mahoney, R., Alonzi, L., Adams, S., ir Beling, P. (2018). Deep learning detecting fraud in credit card transactions. *2018 Systems and Information Engineering Design Symposium (SIEDS)*. 129-134, <https://doi.org/10.1109/SIEDS.2018.8374722>.

44. Schlatt, V., Sedlmeir, J., Feulner, S., ir Urbach, N. (2022). Designing a Framework for Digital KYC Processes Built on Blockchain-Based Self-Sovereign Identity. *Information and Management*, 59(7), 103553. Prieiga per <https://doi.org/10.1016/j.im.2021.103553>
45. Sethi, N., ir Gera, A. (2014). A Revived Survey of Various Credit Card Fraud Detection Techniques. *International Journal of Computer Science and Mobile Computing*. 3(4), 780-791. Prieiga per <https://text2fa.ir/wp-content/uploads/Text2fa.ir-International-Journal-of-Computer-Science-and-Mobile-Computing-1.pdf>
46. Shust, M. P. ir Dostov, V. (2020). Implementing innovative customer due diligence: proposal for universal model. *Journal of Money Laundering Control*, 23(4), 871-884. <https://doi.org/10.1108/JMLC-01-2020-0007>
47. Smolen, T. ir Benova, L. (2023). Comparing Autoencoder and Isolation Forest in Network Anomaly Detection. Proceedings of the 33rd Conference of FRUCT Association, 276-282. <https://doi.org/10.23919/FRUCT58615.2023.10143005>
48. Tama, B. A., Rhee, K. H. (2019). An in-depth experimental study of anomaly detection using gradient boosted machine. *Neural Computing and Applications*, 31, 955-965. <https://doi.org/10.1007/s00521-017-3128-z>
49. The Global Treasure, 2024. *EU Payment Fraud Tops €4.3 Billion, But New Security Measures Show Promise*. Prieiga per <https://www.theglobaltreasurer.com/2024/08/07/eu-payment-fraud-tops-e4-3-billion-but-new-security-measures-show-promise/>
50. Truong, A., Walters, A., Goodsitt, J., Hines, K., Bruss, C. B., Firavar, R. (2019). Towards Automated Machine Learning: Evaluation and Comparison of AutoML Approaches and Tools. *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, 1471-1479. <https://doi.org/10.1109/ICTAI.2019.00209>
51. Wahl, T. (2021). *New EU Anti-Fraud Programme*. Prieiga per <https://eucrim.eu/news/new-eu-anti-fraud-programme/>
52. Walker, M. S. (n.d.). F-Score: *What are Accuracy, Precision, Recall, and F1 Score?* Prieiga per <https://klu.ai/glossary/accuracy-precision-recall-f1>

PRIEDAI

1 PRIEDAS

```
# 1 zingsnis
import pandas as pd
from sklearn.ensemble import IsolationForest
import h2o
from h2o.automl import H2OAutoML

h2o.init()

# 2 zingsnis
file_path = "/kaggle/input/naujaspaskutinis/converted_amounts_in_eur_full.csv"
data_pd = pd.read_csv(file_path, delimiter=';', encoding='utf-8', on_bad_lines='skip')

# 3 zingsnis
data_pd['trans_date'] = pd.to_datetime(data_pd['trans_date'], format='%d/%m/%Y')
data_pd['trans_time'] = pd.to_timedelta(data_pd['trans_time'])

# 4 zingsnis

# 1 kintamasis
data_pd['time_diff'] = data_pd.groupby('Client_ID')['trans_date'].diff().dt.total_seconds() / 60
data_pd['small_transaction'] = data_pd['amount'] < 20
data_pd['frequent_small_trans'] = data_pd['small_transaction'] & (data_pd['time_diff'] < 60)

# 2 kintamasis
data_pd['large_sum'] = data_pd['amount'] > 100000

# 3 kintamasis
data_pd['large_sum_in_eur'] = data_pd['amounts_in_eur'] > 100000

# 4 kintamasis
data_pd['hour_of_day'] = data_pd['trans_time'].dt.total_seconds() // 3600
```

```

data_pd['unusual_time'] = (data_pd['hour_of_day'] < 6) | (data_pd['hour_of_day'] > 23)

# 5 kintamasis
data_pd['round_number'] = (data_pd['amount'] % 100 == 0)

# 6 kintamasis
data_pd['day'] = data_pd['trans_date'].dt.date
data_pd['amount_per_day'] = data_pd.groupby(['Client_ID', 'day'])['amount'].cumsum()

# 7 kintamasis
data_pd['transaction_count_per_day'] = data_pd.groupby(['Client_ID', 'day']).cumcount() + 1

# 8 kintamasis
data_pd['average_transaction_amount'] = data_pd.groupby('Client_ID')['amount'].transform('mean')
data_pd['mean_deviation'] = abs(data_pd['amount'] - data_pd['average_transaction_amount'])

# 9 kintamasis
data_pd['country_mismatch'] = (
    (data_pd['client_country'] != data_pd['phone_code']) &
    ((data_pd['merchant_country'].notnull()) & (data_pd['client_country'] !=
data_pd['merchant_country'])))
)

# 10 kintamasis
data_pd['DECLINED'] = data_pd['transaction_status'].apply(lambda x: 1 if x == 'DECLINED' else
0)
data_pd['declined_mean'] = data_pd.groupby('Client_ID')['DECLINED'].transform('mean')

# 11 kintamasis
data_pd['small_transactions_mean'] =
data_pd.groupby('Client_ID')['small_transaction'].transform('mean')

# 12 kintamasis
data_pd['total_transactions_per_client'] = data_pd.groupby('Client_ID')['amount'].transform('count')

```

```

# 13 kintamasis
data_pd['total_amount_per_client'] = data_pd.groupby('Client_ID')['amount'].transform('sum')

# 5 zingsnis
IFscore = IsolationForest(n_estimators=10, max_samples='auto', contamination='auto',
random_state=42)
IFscore.fit(data_pd[[
    'round_number',
    'amount_per_day',
    'transaction_count_per_day',
    'mean_deviation',
    'country_mismatch',
    'small_transactions_mean']]).fillna(0))

data_pd['IF_score'] = IFscore.decision_function(data_pd[[
    'round_number',
    'amount_per_day',
    'transaction_count_per_day',
    'mean_deviation',
    'country_mismatch',
    'small_transactions_mean']]).fillna(0))

# 6 zingsnis
target = 'Is_fraud_raw_data'
features = [
    'round_number',
    'amount_per_day',
    'transaction_count_per_day',
    'mean_deviation',
    'country_mismatch',
    'small_transactions_mean',
    'IF_score'
]

```

```

# 7 zingsnis
data_pd = data_pd.dropna(subset=[target])
data_pd[target] = data_pd[target].astype('category')

h2o_data = h2o.H2OFrame(data_pd)
h2o_data[target] = h2o_data[target].asfactor()

# 8 zingsnis
train, valid, test = h2o_data.split_frame(ratios=[0.7, 0.15], seed=42)

aml = H2OAutoML(
    max_runtime_secs=7200,
    seed=42,
    balance_classes=True,
    max_models=10,
    nfolds=0,
)

aml.train(x=features, y=target, training_frame=train, validation_frame=valid)

#9 zingsnis
lb = aml.leaderboard
print(lb.head(rows=10))

#10 zingsnis
all_feature_importance = []

for model_id in lb['model_id'].as_data_frame()['model_id']:
    model = h2o.get_model(model_id)

    if hasattr(model, 'varimp'):
        feature_importance = model.varimp(use_pandas=True)
        feature_importance['Model ID'] = model_id
        all_feature_importance.append(feature_importance)

```

```

else:
    print(f"Feature importance not available for model {model_id}.")

if all_feature_importance:
    combined_feature_importance = pd.concat(all_feature_importance, ignore_index=True)
    combined_feature_importance_file_path =
"/kaggle/working/all_models_feature_importance.csv"
    combined_feature_importance.to_csv(combined_feature_importance_file_path, index=False)
    print(f"Feature importance for all models saved to {combined_feature_importance_file_path}")
else:
    print("No feature importance available for any model.")

# 11 zingsnis
parametres = []

important_params = [
    "balance_classes",
    "max_runtime_secs",
    "ntrees",
    "max_depth",
    "min_rows",
    "sample_rate",
    "distribution"
]

for model_id in lb['model_id'].as_data_frame()['model_id']:
    model = h2o.get_model(model_id)

    params = {param: model.params.get(param, {}).get('actual', None) for param in
important_params}
    params["Model ID"] = model_id
    params_df = pd.DataFrame([params])

```

```

if 'all_params_df' in locals():
    all_params_df = pd.concat([all_params_df, params_df], ignore_index=True)
else:
    all_params_df = params_df

params_df = pd.DataFrame(parameters)
params_file_path = "/kaggle/working/model_parameters.csv"

# Save model parameters to CSV
all_params_df.to_csv(params_file_path, index=False)
print(f"Model parameters saved to {params_file_path}")

# 12 zingsnis
all_models_metrics = []

for model_id in lb["model_id"].as_data_frame()["model_id"]:
    model = h2o.get_model(model_id)

    performance_train = model.model_performance(train)
    performance_valid = model.model_performance(valid)
    performance_test = model.model_performance(test)

    train_metrics = {
        "Model ID": model_id,
        "Dataset": "Train",
        "Accuracy": performance_train.accuracy()[0][1],
        "F1-Score": performance_train.F1()[0][1],
        "AUC": performance_train.auc(),
        "AUC-PR": performance_train.aucpr(),
        "Log Loss": performance_train.logloss()
    }

    valid_metrics = {
        "Model ID": model_id,

```

```

        "Dataset": "Validation",
        "Accuracy": performance_valid.accuracy()[0][1],
        "F1-Score": performance_valid.F1()[0][1],
        "AUC": performance_valid.auc(),
        "AUC-PR": performance_valid.aucpr(),
        "Log Loss": performance_valid.logloss()
    }

    test_metrics = {
        "Model ID": model_id,
        "Dataset": "Test",
        "Accuracy": performance_test.accuracy()[0][1],
        "F1-Score": performance_test.F1()[0][1],
        "AUC": performance_test.auc(),
        "AUC-PR": performance_test.aucpr(),
        "Log Loss": performance_test.logloss()
    }

    all_models_metrics.extend([train_metrics, valid_metrics, test_metrics])

all_models_metrics_df = pd.DataFrame(all_models_metrics)

all_models_metrics_file_path = "/kaggle/working/all_models_metrics.csv"

all_models_metrics_df.to_csv(all_models_metrics_file_path, index=False)

print(f"Metrics for all models saved to {all_models_metrics_file_path}")

#13 zingsnis
confusion_matrices = []

for model_id in lb['model_id'].as_data_frame()['model_id']:
    model = h2o.get_model(model_id)
    performance = model.model_performance(test)

```

```
cm_h2o = performance.confusion_matrix()
cm_df = cm_h2o.table.as_data_frame()
cm_df['Model ID'] = model_id
confusion_matrices.append(cm_df)

confusion_matrices_df = pd.concat(confusion_matrices, ignore_index=True)

confusion_matrices_file_path = "/kaggle/working/confusion_matrices.csv"

confusion_matrices_df.to_csv(confusion_matrices_file_path, index=False)

print(f"Confusion matrices saved to {confusion_matrices_file_path}")
```

2 PRIEDAS

26 lentelė

Trylikos kintamųjų visų imčių tikslumas

Modelio pav.	Imtis	Tikslumas	F1 balas	AUC	AUC-PR	Log. nuostolis
GBM_4	Mokymo	1,0000	0,9994	1,0000	1,0000	0,0017
GBM_4	Validacijos	0,9998	0,9939	1,0000	0,9992	0,0027
GBM_4	Testavimo	0,9997	0,9927	1,0000	0,9995	0,0030
GBM_2	Mokymo	0,9998	0,9956	1,0000	0,9998	0,0035
GBM_2	Validacijos	0,9996	0,9896	1,0000	0,9989	0,0042
GBM_2	Testavimo	0,9996	0,9880	1,0000	0,9989	0,0045
XGBoost_2	Mokymo	1,0000	0,9996	1,0000	1,0000	0,0010
XGBoost_2	Validacijos	0,9996	0,9907	1,0000	0,9990	0,0019
XGBoost_2	Testavimo	0,9997	0,9916	1,0000	0,9992	0,0017
GBM_3	Mokymo	0,9999	0,9986	1,0000	0,9999	0,0024
GBM_3	Validacijos	0,9998	0,9936	1,0000	0,9991	0,0032
GBM_3	Testavimo	0,9997	0,9916	1,0000	0,9993	0,0035
GBM_1	Mokymo	1,0000	0,9988	1,0000	0,9999	0,0018
GBM_1	Validacijos	0,9996	0,9886	1,0000	0,9985	0,0037
GBM_1	Testavimo	0,9996	0,9883	1,0000	0,9986	0,0039
XGBoost_1	Mokymo	1,0000	0,9996	1,0000	1,0000	0,0012
XGBoost_1	Validacijos	0,9996	0,9906	1,0000	0,9988	0,0021
XGBoost_1	Testavimo	0,9996	0,9897	1,0000	0,9989	0,0020
XGBoost_3	Mokymo	1,0000	0,9996	1,0000	1,0000	0,0014
XGBoost_3	Validacijos	0,9999	0,9968	0,9996	0,9990	0,0020
XGBoost_3	Testavimo	0,9998	0,9957	0,9999	0,9992	0,0020
XRT_1	Mokymo	0,9988	0,9675	0,9998	0,9940	0,0509
XRT_1	Validacijos	0,9974	0,9295	0,9991	0,9744	0,0538
XRT_1	Testavimo	0,9971	0,9197	0,9989	0,9705	0,0535
DRF_1	Mokymo	0,9993	0,9826	1,0000	0,9983	0,0234
DRF_1	Validacijos	0,9976	0,9365	0,9989	0,9728	0,0306
DRF_1	Testavimo	0,9975	0,9298	0,9988	0,9721	0,0307
GLM_1	Mokymo	0,9811	0,2407	0,8410	0,1396	0,0772
GLM_1	Validacijos	0,9809	0,2417	0,8447	0,1400	0,0776
GLM_1	Testavimo	0,9813	0,2407	0,8361	0,1352	0,0774

3 PRIEDAS

27 lentelė

Šešių kintamųjų visų imčių metrikos

Modelio pav.	Imtis	Tikslumas	F1 balas	AUC	AUC-PR	Log. nuostolis
GBM_1	Mokymo	0,9971	0,9218	0,9995	0,9794	0,0142
GBM_1	Validacijos	0,9950	0,8573	0,9940	0,9106	0,0229
GBM_1	Testavimo	0,9948	0,8459	0,9938	0,9010	0,0234
XGBoost_3	Mokymo	0,9988	0,9693	0,9999	0,9949	0,0195
XGBoost_3	Validacijos	0,9952	0,8654	0,9927	0,9104	0,0300
XGBoost_3	Testavimo	0,9951	0,8607	0,9931	0,9076	0,0294
GBM_4	Mokymo	0,9942	0,8298	0,9945	0,8986	0,0280
GBM_4	Validacijos	0,9931	0,7993	0,9916	0,8700	0,0311
GBM_4	Testavimo	0,9931	0,7904	0,9915	0,8586	0,0311
XGBoost_1	Mokymo	0,9999	0,9966	1,0000	0,9997	0,0067
XGBoost_1	Validacijos	0,9948	0,8537	0,9894	0,8984	0,0225
XGBoost_1	Testavimo	0,9947	0,8466	0,9892	0,8894	0,0231
GBM_3	Mokymo	0,9923	0,7792	0,9904	0,8384	0,0347
GBM_3	Validacijos	0,9918	0,7662	0,9889	0,8276	0,0360
GBM_3	Testavimo	0,9914	0,7497	0,9886	0,8154	0,0361
XGBoost_2	Mokymo	0,9997	0,9929	1,0000	0,9992	0,0090
XGBoost_2	Validacijos	0,9949	0,8561	0,9889	0,9007	0,0236
XGBoost_2	Testavimo	0,9948	0,8493	0,9900	0,8958	0,0235
GBM_2	Mokymo	0,9918	0,7539	0,9890	0,8226	0,0362
GBM_2	Validacijos	0,9915	0,7480	0,9880	0,8197	0,0371
GBM_2	Testavimo	0,9913	0,7338	0,9880	0,8044	0,0369
XRT_1	Mokymo	0,9980	0,9444	0,9995	0,9849	0,0361
XRT_1	Validacijos	0,9911	0,7368	0,9776	0,7985	0,0545
XRT_1	Testavimo	0,9909	0,7274	0,9793	0,7858	0,0538
DRF_1	Mokymo	0,9965	0,9038	0,9984	0,9579	0,0603
DRF_1	Validacijos	0,9904	0,7214	0,9758	0,7720	0,0693
DRF_1	Testavimo	0,9902	0,7062	0,9773	0,7588	0,0681
GLM_1	Mokymo	0,9812	0,1495	0,7560	0,0742	0,0852
GLM_1	Validacijos	0,9809	0,1650	0,7632	0,0814	0,0854
GLM_1	Testavimo	0,9813	0,1525	0,7499	0,0744	0,0850

4 PRIEDAS

28 lentelė

Galutinių modelių kintamųjų visų imčių metrikos

Modelio pav.	Imtis	Tikslumas	F1 balas	AUC	AUC-PR	Log. nuostolis
GBM_1	Mokymo	0,9978	0,9421	0,9998	0,9886	0,0126
GBM_1	Validacijos	0,9942	0,8340	0,9921	0,8925	0,0271
GBM_1	Testavimo	0,9942	0,8310	0,9926	0,8866	0,0268
GBM_4	Mokymo	0,9952	0,8660	0,9983	0,9429	0,0218
GBM_4	Validacijos	0,9932	0,8092	0,9917	0,8760	0,0294
GBM_4	Testavimo	0,9931	0,8026	0,9921	0,8670	0,0291
XGBoost_3	Mokymo	0,9930	0,8060	0,9948	0,8824	0,0293
XGBoost_3	Validacijos	0,9921	0,7873	0,9903	0,8500	0,0328
XGBoost_3	Testavimo	0,9920	0,7715	0,9906	0,8365	0,0327
GBM_2	Mokymo	0,9990	0,9742	0,9999	0,9959	0,0188
GBM_2	Validacijos	0,9947	0,8487	0,9899	0,8982	0,0307
GBM_2	Testavimo	0,9947	0,8496	0,9913	0,8966	0,0300
GBM_3	Mokymo	0,9922	0,7859	0,9930	0,8594	0,0321
GBM_3	Validacijos	0,9915	0,7625	0,9894	0,8332	0,0346
GBM_3	Testavimo	0,9915	0,7505	0,9895	0,8226	0,0344
XGBoost_2	Mokymo	0,9998	0,9957	1,0000	0,9995	0,0078
XGBoost_2	Validacijos	0,9944	0,8396	0,9865	0,8858	0,0252
XGBoost_2	Testavimo	0,9945	0,8392	0,9882	0,8807	0,0248
XGBoost_1	Mokymo	0,9999	0,9971	1,0000	0,9997	0,0063
XGBoost_1	Validacijos	0,9944	0,8421	0,9858	0,8848	0,0244
XGBoost_1	Testavimo	0,9943	0,8350	0,9869	0,8776	0,0245
XRT_1	Mokymo	0,9968	0,9152	0,9992	0,9671	0,0414
XRT_1	Validacijos	0,9883	0,6510	0,9661	0,6943	0,0616
XRT_1	Testavimo	0,9886	0,6505	0,9679	0,6859	0,0604
DRF_1	Mokymo	0,9914	0,7605	0,9953	0,8608	0,0557
DRF_1	Validacijos	0,9862	0,5833	0,9613	0,6225	0,0675
DRF_1	Testavimo	0,9865	0,5795	0,9626	0,6192	0,0662
GLM_1	Mokymo	0,9812	0,1956	0,7903	0,1103	0,0817
GLM_1	Validacijos	0,9810	0,2073	0,7891	0,1220	0,0820
GLM_1	Testavimo	0,9813	0,2004	0,7959	0,1165	0,0805