VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS INSTITUTAS
KOMPIUTERINIO IR DUOMENŲ MODELIAVIMO KATEDRA

Baigiamasis bakalauro darbas

Klientų atsiliepimų sistema

Atliko:
Aleksandras Veretenikas parašas

Vadovas:
J. Asist. Andrius Vytautas Misiukas Misiūnas

Vilnius
2020

Turinys

Sutartinis terminų žodynas

TS (angl. *TypeScript*) - Microsoft sukurta programavimo kalba. Viršutinis rinkinys JavaScript kalbai. Kompiliavimo metu TypeScript kodas yra paverčiamas į JavaScript.
UML (angl. *Unified Modeling Language*) - Vieninga modeliavimo kalba. Skirta atvaizduoti ir konstruoti objektiškai orientuotų programų dokumentus.
HTML (angl. *Hyper text Markup Language*) - Interneto turiniui pateikti skirta hiperteksto žymėjimo kalba.
CRUD (angl. *Create, Read, Update and Delete*) - Sukurti, skaityti, atnaujinti ir ištrinti. Keturios pagrindinės funkcijos dirbant su duomenų saugojimo technologijomis.
CSS (angl. *Cascading Style Sheets*) - Pakopiniai stilių šablonai. Kalba, kuri yra skirta a nurodyti dokumento vaizdavimą, parašyta kita struktūrine kalba.
SQL (angl. *Structured Query Language*) - Struktūrizuota užklausų kalba. Kalba, skirta manipuliuoti duomenimis reliacinių duomenų bazių valdymo sistemose.
CLI (angl. *Command Line Interface*) - Komandinės eilutės sąsaja.
JSON (angl. *JavaScript Object notation*) - Duomenų perdavimo formatas, kuriame objektas yra sudarytas iš reikšmių porų.
PBKDF2 (angl. *Password-Based Key Derivation Function*) - Rakto formavimo funkcija, skirta slaptažodžių užkodavimui. Priklauso viešųjų raktų kriptografijos standartui.
EF (angl. *Entity Framework*) - Objektų ir reliacijų priskyrimų sistema.

Santrauka

Šio baigiamojo darbo tikslas yra klientų atsiliepimų sistemos sukūrimas. Sistema skirta padėti įmonėms peržiūrėti, analizuoti ir atsakyti į jų klientų suformuotas nuomones apie produktą. Tuo tarpu pats klientas sistemoje gali lengvai palikti atsiliepimą ir susisiekti su atsiliepimų analitiku. Nuomonių pozityvumas yra nuspėjamas naudojantis ištreniruotu daugiaklasės klasifikacijos modeliu, o negatyviausi atsiliepimai sistemoje yra paryškinami. Tikslui įgyvendinti buvo nagrinėjamos panašios sistemos Lietuve ir užsienyje. Taip pat nagrinėjami tokie kompiuterio mokymo algoritmai ir klasifikatoriai, kaip maksimali entropija, naivus Bajesas, sprendimų medžiai. Geriausius rezultatus parodęs klasifikatorius - maksimali entropija. Darbui atlikti naudojamos C#, Angular, ASP.NET, ML.NET, EF technologijos. Gautas rezultatas yra klientų atsiliepimų sistema, kurioje atsiliepimai yra vertinami ištreniruotu daugiaklasės klasifikacijos modeliu ir vaizduojami analitikui, kuris atsižvelgęs į atsiliepimo negatyvumą gali palaikyti ryšį su klientu. Taip pat sistemoje yra ir paprasti naudotojai, kurie gali kurti ir redaguoti atsiliepimus, o administratoriai be analitiko privilegijų, taip pat turi prieigą prie naudotojų valdymo. Sistema kuriama anglų kalba, kadangi modeliui apmokyti reikia daug duomenų, kuriuos gali pateikti tik stambios užsienio įmonės.

Summary

Customer Feedback System

The main goal of this work is to create a customer feedback system, which allows companies to look through, analyse and respond to the opinions that are formed about their product. Meanwhile the system also allows for clients to easily leave some feedback about the product and lets the user contact a feedback analyst. The positivity of opinions is predicted by using a trained multiclass classification model and the most negative feedback is highlighted in the system. To achieve this, the analysis of similar systems that are currently operating in Lithuania, as well as abroad, has been conducted. Machine learning algorithms and classifiers are examined, such as, max entropy, naive Bayes, decision trees. Best performer proved to be max entropy classifier. The work is done using C#, Angular, ASP.NET, ML.NET, EF technologies. The achieved result is a customer feedback system in which the feedback is evaluated by a trained multiclass classification model and presented to an analyst, who takes into account the negativity of the feedback and keeps contact with the client who raised it. System also contains regular users, who are able to create and amend feedback items. The are also administrator type users who not only have analyst privileges, but also have access to control of system users. System is developed using English language, as in order to train my model i need a lot of data, which can only be provided by international companies.

Įvadas

Palaikyti ryšį su klientu yra be galo svarbi verslo dalis. Jei klientas nesijaus, kad jo nuomonė apie produktą nėra išklausyta ir jokių veiksmų, jei jis teikia nepasitenkinimą produktu nėra imamasi, atsiranda didelė galimybė, kad tas klientas greičiausiai pasirinks kitą produktą.

Nagrinėti klientų atsiliepimus yra daug laiko reikalaujantis darbas. Taip pat didelėse ir sėkmingose verslo organizacijose atsiliepimai gali kauptis per dideliu greičiu, kad galima būtų juos visus išanalizuoti rankiniu būdu. Būtent todėl atsiranda vis daugiau sistemų, kurios padeda juos apdoroti ir taip gerinti klientų nuomonę apie kuriamą produktą ar paslaugą.

Šio darbo tikslas: apmokyti daugiaklasės klasifikacijos modelį ir sukurti sistemą, kuri atskiria negatyviausius atsiliepimus ir padeda naudotojui inicijuoti atitinkamus veiksmus. Sistema kuriama naudojantis "ASP.NET" ir "Angular" karkasais, C#, TypeScript, HTML ir CSS kalbomis.

Darbo uždaviniai:

- Įvertinti ir palyginti panašias sistemas kurios jau egzistuoja.
- Pasirinkti tinkamas technologijas sistemos įgyvendinimui.
- Išanalizuoti įvairius klasifikacijų algoritmus ir klasifikatorius.
- Surasti tinkamą duomenų rinkinį modeliui apmokyti.
- Išsiaiškinti, kuris daugiaklasės klasifikacijos algoritmas yra labiausiai tinkantis turimam duomenų rinkiniui.
- Apmokyti modelį pasirinktu algoritmu.
- Suprojektuoti duomenų bazės modelį, sistemos architektūrą ir funkcionalumą.
- Įgyvendinti praktinę darbo dalį, sukuriant funkcionalią klientų atsiliepimų valdymo sistemą.

Pasiekti rezultatai ir atlikti uždaviniai:

Amazon.com klientų atsiliepimų duomenimis apmokytas daugiaklasės klasifikacijos modelis, kuris pagal parašytos nuomonės apie produktą turinį ir atspėja jo pozityvumo lygį. Šis modelis integruotas į sukurtą klientų atsiliepimų sistemą, kurioje klientas gali prisijungti ir palikti savo nuomonę apie produktą. Atsiliepimų analistas tuo tarpu turi galimybę peržiūrėti visus sistemoje paliktus atsiliepimus, ant kiekvieno iš jų palikti naudotojui po žinutę, apie kurią jis informuojamas elektroniniu paštu, bei palikti užrašą, kurį matys kiti analistai ir administratoriai.

1. Panašių sistemų analizė

Siekiant sukurti sistemą su norimu funkcionalumu reikia iš pradžių išanalizuoti, kas jau yra sukurta rinkoje. Kuriamos programos pagrindinis tikslas yra suteikti naudotojams funkcionalumą lengvai kurti atsiliepimus, o analitikui paliktus atsiliepimus analizuoti ir palaikyti tolimesnį ryšį su tuo klientu. Būtent šio tipo sistemų ir reikia surasti. Kadangi svarbi sistemos dalis yra ir sentimentų analizės naudojimas, tai reikia atsižvelgi ar jau rinkoje esantys sprendimai taip pat naudoja šią technologiją.

1.1. Panašios sistemos Lietuvoje

Artimiausias mano planuojamam funkcionalumui sprendimas Lietuvoje ko gero yra evertink.lt. Šioje sistemoje naudotojai gali peržiūrėti įvairius atsiliepimus, taip pat po jais palikti komentarus. Žinoma, nors komentarai ir žinutės nėra visai vienodas funkcionalumas, tačiau tai yra vienas iš būdų susisiekti su klientu. Ši sistema klientų atsiliepimų analizavimui nepasitelkia sentimentų analizės. Neatrodo, kad egzistuotų funkcionalumas analitikui prie atsiliepimo palikti privatų užrašą. Taip pat klientų nuomonės yra viešai prieinamos visiems naudotojams, kas ne visada yra gerai paslaugos tiekėjui, kuris nori pakeisti naudotojo nuomonę prieš skelbiant ją viešai.

1.2. Panašios sistemos užsienyje

Daug panašesnė į mano planuojamą sistemą yra reviewtrackers.com. Ši sistema turi daug įvairių funkcijų, tokių kaip: reputacijos valdymas, apklausų kūrimas, marketingas ir t.t. Tačiau didelė dalis šios sistemos funkcionalumo sutampa su mano planuojama kurti sistema. Reviewtrackers.com rodo atsiliepimus, naudoja sentimentų analizę jiems apdoroti, taip pat ši sistema leidžia atsakyti į kliento įkeltą atsiliepimą. Tačiau šioje sistemoje taip pat pasigedau privatesnio nuomonių valdymo, kadangi jos vis tiek yra viešai prieinamos. Ši sistema taip pat nėra nemokama ir blokuoja įvairų funkcionalumą pagal mokamo abonemento lygį. Taip pat kai kuriems naudotojams sistemos navigacija pasirodė gana paini.

	reviewtrackers.com	evertink.lt
Nemokama versija	Ne	Taip
Atsiliepimai analizuojami pasinaudojant sentimentų analize	Taip	Ne
Galimybė atsakyti klientui	Taip	Taip (mokama)
Galimybė palikti užrašus apie atsiliepimus, kurių nemato klientas	Taip	Ne
Ar atsiliepimai yra privatūs?	Ne	Ne

Iš aptartų jau egzistuojančių rinkoje sistemų galime pastebėti, kad jos abi turi trūkumų. Taip pat norint pasinaudoti pilnu šių sistemų funkcionalumų reikia mokėti uz aukščiausio lygio abonementus, kurių kaina yra gana aukšta. Galima padaryti išvadą, kad rinkoje vis dar yra vietos naujoms tokio tipo sistemoms.

2. Sistemos projektavimas

2.1. Numatomas sistemos funkcionalumas

- Registracija naujiems naudotojams.
- Naudotojų prisijungimas prie sistemos.
- Galimybė sukurti, pakeisti, peržiūrėti ir ištrinti atsiliepimą apie produktą (Šių veiksmų prieinamumas priklauso nuo naudotojo tipo).
- Sukurto arba atnaujinto atsiliepimo turinys yra įvertinamas penkių balų sistemoje pagal ištreniruotą sentimentų analizės modelį.
- Atsiliepimų sąrašo filtravimas, rikiavimas ir puslapiavimas.
- Administratoriaus panelė su galimybe peržiūrėti, redaguoti, sukurti ir ištrinti naudotojus.
- Galimybė palikti užrašus prie kiekvieno atsiliepimo.
- Galimybė palikti žinutę atsiliepimo kūrėjui apie kurią jis bus informuotas elektroniniu paštu.

2.2. Sistemos naudotojų tipai ir jiems prieinamas funkcionalumas

Sistemoje egzistuoja 3 naudotojų tipai: naudotojas, atsiliepimų analitikas ir administratorius. Naudotojo rolė yra skiriama sistemos lankytojams, kurie nori palikti atsiliepimą apie kokį nors produktą. Atsiliepimo analitiko rolė skiriama asmenims, kurie peržiūri naudotojų paliktus atsiliepimus ir atlieka atitinkamus veiksmus. Administratoriui prieinama didžiausia dalis sistemos funkcionalumo. Paprastas naudotojas gali atlikti šiuos veiksmus (Žiūrėti 1pav.):

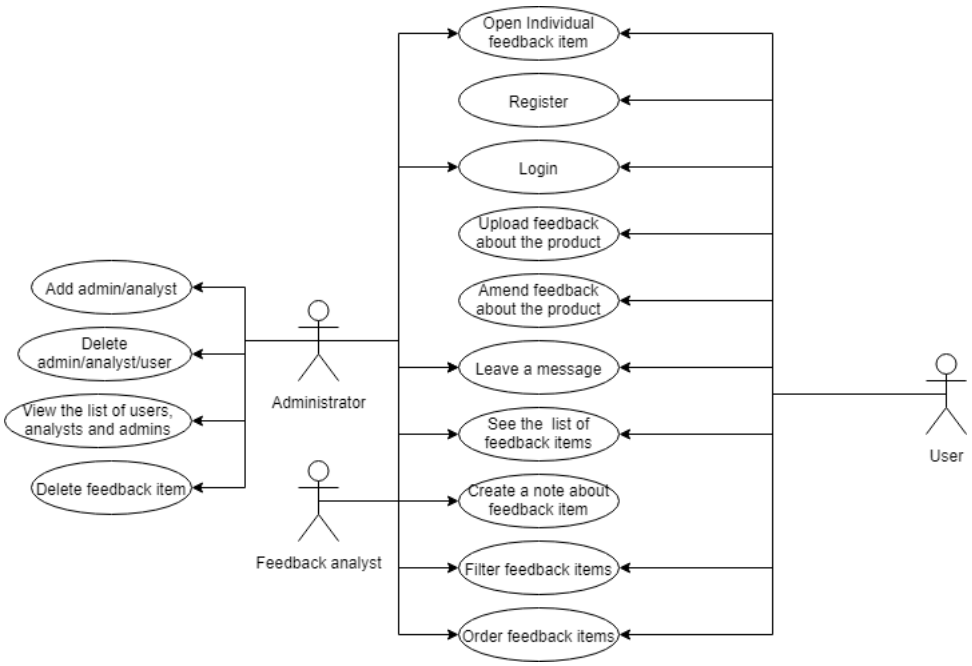
- Registruotis;
- Prisijungti;
- Sukurti atsiliepimą apie produktą;
- Matyti savo atsiliepimų sąrašą;
- Peržiūrėti savo individualius atsiliepimus;
- Filtruoti savo atsiliepimų sąrašą;
- Rūšiuoti savo atsiliepimų sąrašą;
- Papildyti atsiliepimo turinį;
- Palikti žinutę atsiliepimų analitikui.

Atsiliepimų analitikas gali atlikti šiuos veiksmus:

- Prisijungti;
- Peržiūrėti atsiliepimų sąrašą
- Peržiūrėti individualius atsiliepimus;
- Filtruoti savo atsiliepimų sąrašą;
- Rūšiuoti savo atsiliepimų sąrašą;
- Palikti užrašus prie atsiliepimų (Naudinga jei kontaktas su naudotoju vyksta už sistemos ribų, pavyzdžiui, susitikimas, skambutis);
- Palikti žinutę atsiliepimo autoriui.

Administratorius gali atlikti šiuos veiksmus:

- Prisijungti;
- Peržiūrėti atsiliepimų sąrašą
- Filtruoti savo atsiliepimų sąrašą;
- Rūšiuoti savo atsiliepimų sąrašą;
- Peržiūrėti individualius atsiliepimus;
- Palikti užrašus prie atsiliepimų (Naudinga jei kontaktas su naudotoju vyksta už sistemos ribų, pavyzdžiui, susitikimas, skambutis);
- Palikti žinutę atsiliepimo autoriui.
- Uždaryti atsiliepimą.
- Peržiūrėti, kurti ir trinti sistemos naudotojus.



1 pav. UML Panaudojimo atvejų diagrama

2.3. Duomenų bazės modelis

Sistemos duomenų saugojimui yra naudojama SQL server duomenų bazių valdymo sistema. *AspNetUsers* lentelė sugeneruota ASP.NET identity įrankiu, todėl dažniausiai naudotojo autentifikacijai reikalingi laukai yra sugeneruojami automatiškai. Taip pat automatiškai sugeneruotos ir *AspNetRoleClaims*, *AspNetRoles*, *AspNetUserClaims*, *AspNetUserLogins*, *AspNetUserRoles*, *AspNetUserTokens* lentelės, tačiau jos nėra naudojamos.

Duomenų bazė yra sudaryta iš šių lentelių (Žiūrėti 2pav.):

- **AspNetUsers** - lentelė, kurioje saugomi su sistemos naudotojais susiję duomenys. Šią lentelę sudaro identifikacinis raktas *Id*, naudotojo vardas saugomas lauke *UserName*, o didžiosiomis raidėmis išsaugotas normalizuotas variantas *NormalizedUserName*. Elektroninis paštas ir normalizuotas jo variantas atitinkamai išsaugomi *Email* ir *NormalizedEmail*. Slaptažodis saugomas *PasswordHash* lauke ir yra užkoduojamas PBKDF2 maišos funkcija. Telefono numeris išsaugomas *PhoneNumber* lauke. Naudotojas taip pat turi vardui skirtą lauką *Name* bei jam tinkamiausią susisiekimo metodą *PreferredContactMethod*. Taip šioje lentelėje yra saugomas ir naudotojo tipas *UserType*. Ši lentelė sujungta su *Reviews* 1:N ryšiu.

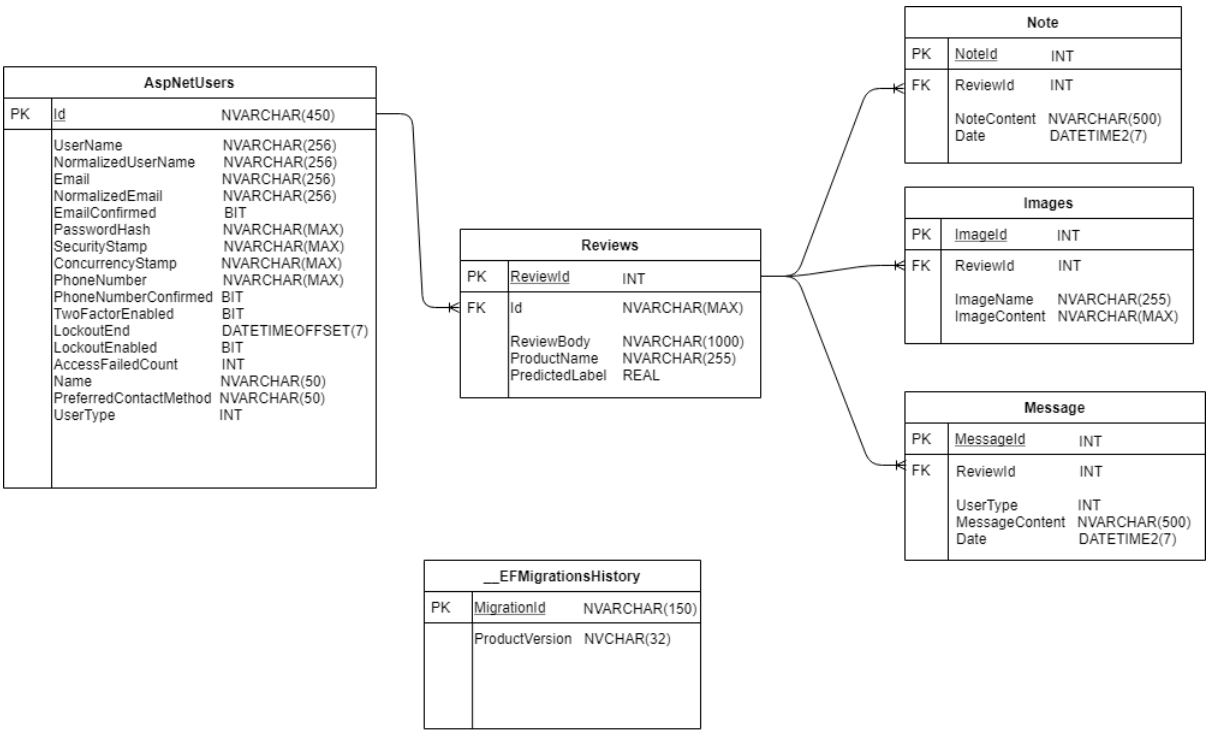
- **Reviews** - lentelė, kurioje saugomi atsiliepimai. Šios lentelės identifikacinis numeris yra saugomas *ReviewId*. *Id* yra išorinis raktas, kuris atitinka *AspNetUsers* lauką *Id*. *ReviewName* saugo atsiliepimo pavadinimą, o *ReviewBody* - atsiliepimo turinį. Ši lentelė sujungta su *AspNetUsers* lentele N:1 ryšiu, taip pat su lentelėmis *Note*, *Images* ir *Message* 1:N ryšiais. *PredictedLabel* yra apmokyto daugiaklasės klasifikacijos modelio pozityvumo spėjimas.

- **Note** - lentelė, kurioje saugomi užrašų duomenys. Šios lentelės identifikacinis numeris yra saugomas *NoteId*, o išorinis raktas *ReviewId* atitinka *Reviews* lentelės *ReviewId*. *NoteContent* saugomas užrašo turinys. *Date* - sugeneruota užrašo įkėlimo data. Ši lentelė sujungta su *Reviews* lentele N:1 ryšiu.

- **Images** - lentelė, kurioje saugomi nuotraukų duomenys. Šios lentelės identifikacinis numeris yra saugomas *ImageId*, o išorinis raktas *ReviewId* atitinka *Reviews* lentelės *ReviewId*. Nuotraukos pavadinimas saugomas lauke *ImageName*. Nuotraukos turinys *ImageContent* duomenų bazėje yra užkoduotas base64 formatu. Ši lentelė sujungta su *Reviews* lentele N:1 ryšiu.

- **Message** - lentelė, kurioje saugomi žinučių duomenys. Šios lentelės identifikacinis numeris yra saugomas *MessageId*, o išorinis raktas *ReviewId* atitinka *Reviews* lentelės *ReviewId*. *MessageContent* lauke yra saugomas žinutės turinys. Norint žinoti, kokio tipo sistemos naudotojas paliko žinutę, naudotojo kodas yra išsaugojamas kaip *UserType Date* - sugeneruota žinutės įkėlimo data. Ši lentelė sujungta su *Reviews* lentele N:1 ryšiu.

- **__EFMigrationsHistory** - lentelė, kurioje saugoma duomenų bazės pokyčių istorija. Sugeneruota naudojantis Entity Framework. *MigrationId* - unikalus raktas migracijai identifikuoti. Taip pat yra išsaugoma produkto versija, kuri saugoma *ProductVersion* lauke.



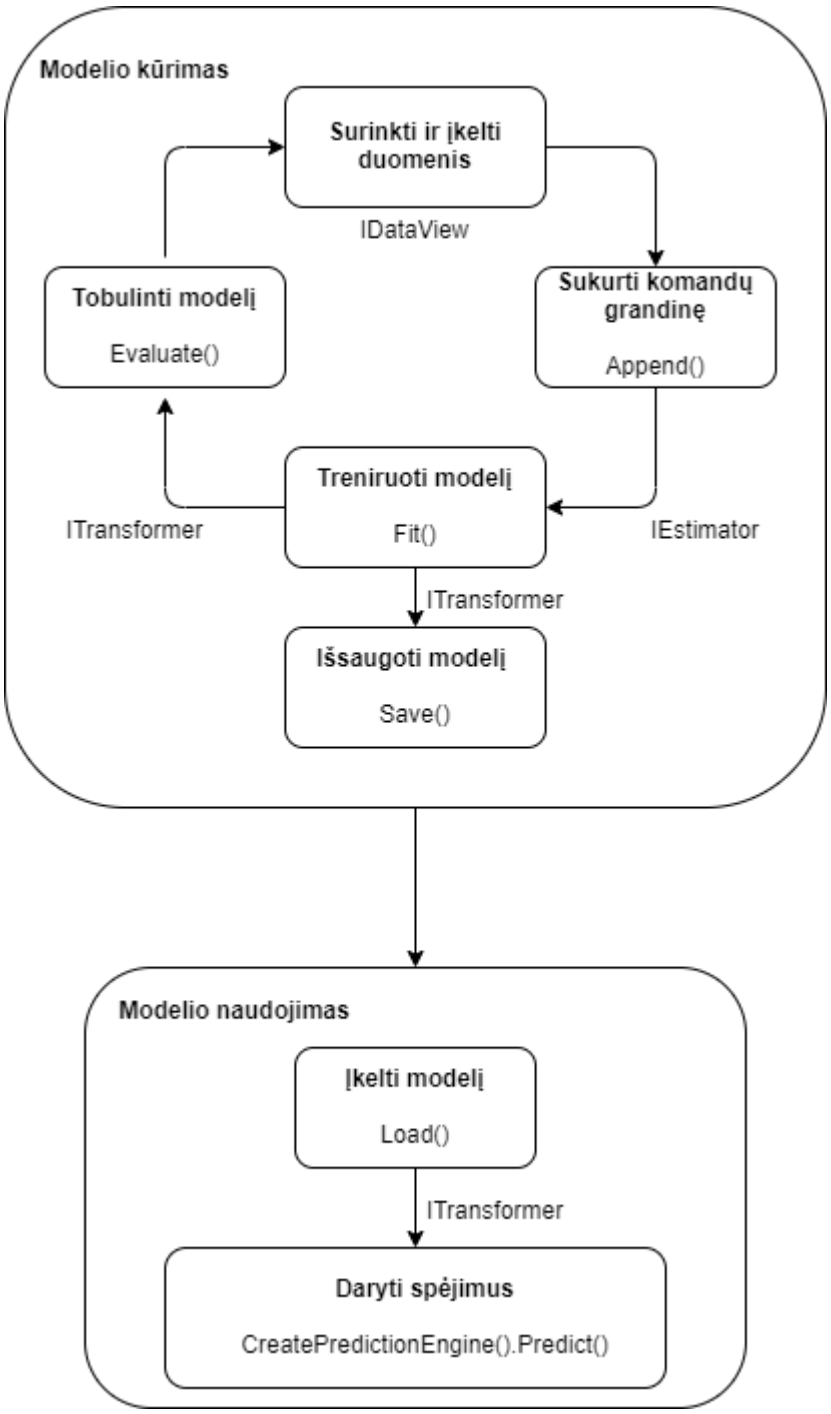
2 pav. Reliacinė schema

3. Klasifikacijos modelių analizavimas ir treniravimas

Sentimentų analizei atlikti naudoju ML.NET. ML.NET suteikia galimybę nesunkiai pridėti kompiuterio mokymą (machine learning) į .NET aplikacijas. Kadangi kurdamas savo sistemą naudoju ASP.NET core, kompiuterio apmokymui nusprendžiau naudoti būtent šį karkasą. ML.NET taip pat yra atviro kodo ir palaiko kitas populiarias kompiuterio apmokymo bibliotekas (TensorFlow, ONNX ir kt.).

Modelio kūrimo procesas naudojantis ML.NET:

- Surinkti ir įkelti duomenis, skirtus modelio treniravimui į IDataView objektą
- Nurodyti operacijų eigą savybėms ištraukti ir pritaikyti kompiuterio mokymo algoritmą
- Treniruoti modelį pasinaudojant Fit()
- Įvertinti modelį, kartoti iteraciją jo pagerinimui
- Išsaugoti modelį į binarinį formatą kad galima būtų jį aplikacijoje
- Įkelti modelį į ITransformer objektą
- Daryti spėjimus pagal ištreniruotą modelį naujiems duomenims pasinaudojant CreatePredictionEngine.Predict()



3 pav. ML.MET darbo eiga

Norint treniruoti savo modelį iš pradžių reikia pasirinkti kuri iš šių dviejų klasifikacijų labiau tinka mano turimiems duomenims.

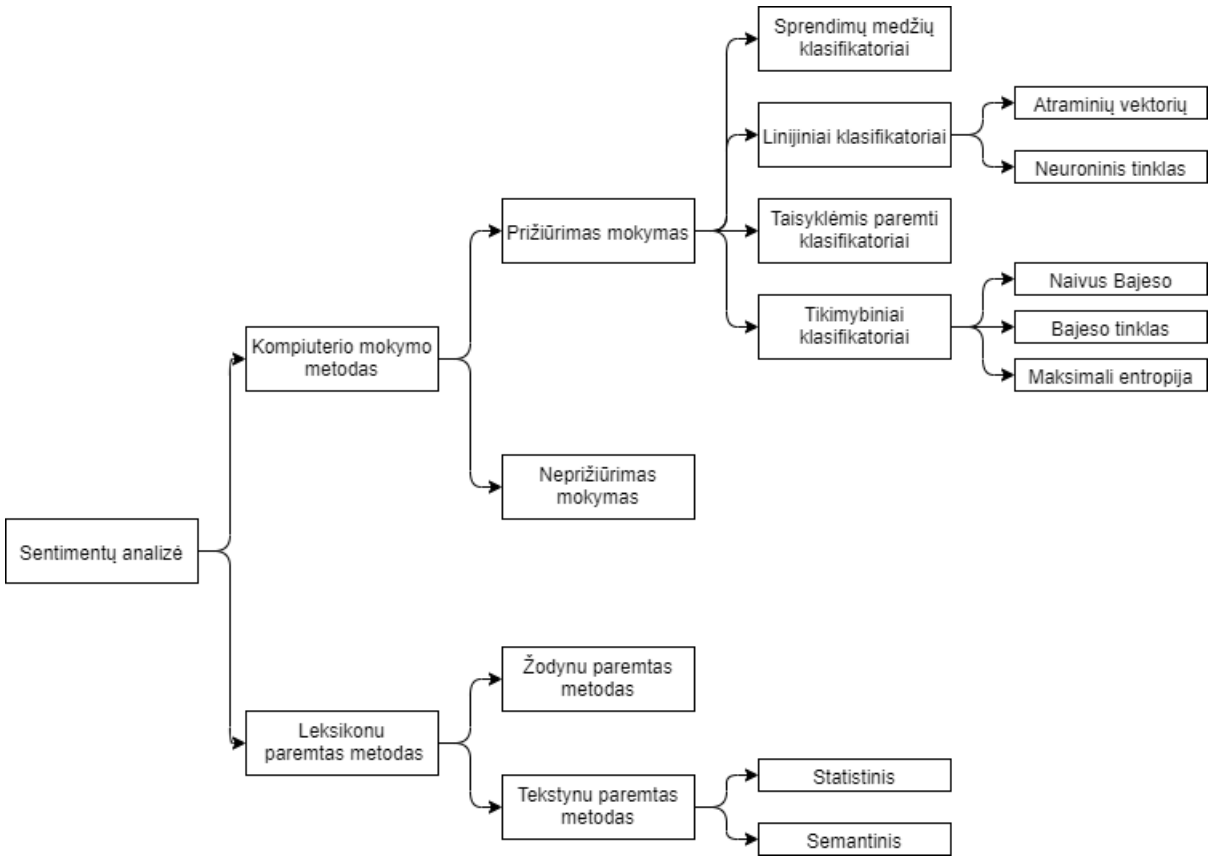
Binarinė klasifikacija (Binary classification)

Binarinė klasifikacija, kaip galima nuspėti iš pavadinimo, specializuojasi kategorizuojant gautą informaciją tik į pozityvią arba negatyvią t.y įvestis priskiriama 0 arba 1. Ši klasifikacija gali būti naudinga:

- Suprasti ar komentarai yra pozityvūs ar negatyvūs
- Nustatyti ar pacientas turi specifinę ligą ar ne
- Nustatyti ar gautas elektroninis laiškas yra reklaminio pobūdžio

Daugiaklasė klasifikacija (Multiclass classification)
Daugiaklasė klasifikacija, skirtingai nei binarinė gali nuspėti duomenis suskirstytus į daugiau nei 2 kategorijas. Šios klasifikacijos algoritmo įvestis yra rinkinys sugrupuotų pavyzdžių. Algoritmas gražina klasifikatorių, kuriuo pasinaudojant galima nuspėti klasę naujai įvestiems duomenims. Šios klasifikacijos pavyzdžiai:

- Nustatyti gyvūno veislę
- Nustatyti ar kliento atsiliepimas yra teigiamas, neutralus arba neigiamas
- Kategorizuoti automobilių atsiliepimus pagal kainą, degalų sunaudojimą , taršumą ir t.t



4 pav. Sentimentų analizės metodai

3.1. Klasifikatorių pavyzdžiai:

3.1.1. Maximum Entropy

Maximum Entropy yra tikimybinis klasifikatorius, priklausantis eksponentinių modelių klasei. Šis klasifikatorius pasižymi tuo, kad nesitiki jog ypatybės yra sąlygiškai nepriklausomos viena nuo kitos. Tai ypač aktualu teksto klasifikavimo problemose, kai naudojami bruožai yra žodžiai, kurie nėra savarankiški.

Norint suprasti, kas yra maksimali entropija (maximum entropy), iš pradžių reikia išanalizuoti kas yra paprasta entropija.

$$-\sum \pi * log(\pi)$$

Paprastai paaiškinus, entropija yra nežinomumo arba atsitiktinumo matavimas. Kuo daugiau egzistuoja baigčių objektui, tuo jo entropija yra aukštesnė t.y galutinė objekto būsena yra labai sunku nuspėti.

Maksimalios entropijos principas teigia[7], kad pats tinkamiausias duomenų modeliui pasiskirstymas yra su aukščiausia entropija pagal jau nustatytas ribas, kurias apibrėžiame iš jau turimų žinių. Turimos žinios šiuo atveju gali būti išreikštos kaip apribojimai tikimybiniams pasiskirstymams, kurie apibūdina sistemos rezultatus. Dažnai šios ribos yra

apibrėžtos kaip norimo pasiskirstymo momentų lygtys. Duotoms funkcijoms $f_1, \ldots, f_K$ ir skaičiams $c_1, \ldots, c_K$, ribos yra tokios formos:

$$E_p[f_k(X)] = \sum_{n=1}^{N} f_k(n)p(n) = c_k, \quad k = 1, \ldots, K, \;\; X \sim p$$

Maximum entropy paprastai yra laikomas tiksliu klasifikatoriumi, tačiau egzistuoja ir atvejų, kai jis nesugeba pateikti teisingus spėjimus. Taip dažniausiai atsitinka, kai jau turimos žinios, gautos iš sistemos buvo neteisingai pamatuotos, šališkos. Taip pat klaidingus spėjimus šis klasifikatorius gali teikti ir dėl su klaidomis atliktos analizės. Dar vienas variantas yra staigūs sistemos pokyčiai, dėl kurių ankstesnės žinios tampa neaktualios.

3.1.2. Naive Bayes

Naive Bayes[5] yra tikimybinis klasifikatorius, paremtas Bajeso teorema. Skirtingai nei Maximum Entropy, šis algoritmas stipriai tikisi, jog ypatybės yra sąlygiškai nepriklausomos viena nuo kitos.

Bajeso teorema yra aprašoma taip:

$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)}$$

Čia A ir B nepriklausomi įvykiai

- $P(A \mid B)$ - tikimybė įvykiui A, kai B jau įvyko
- $P(B \mid A)$ - tikimybė įvykiui B, kai A jau įvyko
- $P(A)$ - įvykio A tikimybė
- $P(B)$ - įvykio B tikimybė

Šis klasifikatorius veikia gana greitai, tačiau dažnai yra mažiau tikslesnis nei kiti klasifikatoriai.

3.2. Daugiaklasės klasifikacijos algoritmų pavyzdžiai:

3.2.1. LightGbmMulticlassTrainer

LightGBM[6] yra atviro kodo implementacija gradient boosting decision tree algoritmo. Šis algoritmas suprojektuotas tolygiam paskirstymui ir didesniam našumui. LightGBM, kitaip nei kiti sprendimų medžių algoritmai, neatlieka augimo kiekvienam lygmeniui, o pasirenka medžio "lapą", kuris turi didžiausią deltos praradimą.

3.2.2. SdcaMaximumEntropyMulticlassTrainer

Ši klasė naudoja empirinę rizikos minimizaciją (Empirical risk minimization arba ERM) kad suformuluoti optimizacijos problemą iš surinktų duomenų. ERM dažniausiai yra matuojamas pritaikant nuostolių funkciją modelio spėjimams API surinktus taškus. Jei duomenys neturi pakankamai taškų gali atsitikti taip, kad modelis išmoks labai gerai spėlioti apie duomenis, kurie yra panašūs į treniravimo duomenis, tačiau teiks labai prastus sprendimus jei duomenys yra nematyti. Šiai problemai sušvelninti dažnai naudojama reguliarizavimas. Reguliarizavimas yra papildomos informacijos panaudojimas sprendžiant nekorektiškus uždavinius. Jį naudojant verta atminti kad naudojant labai agresyvų reguliarizavimą visi parametrai gali būti paversti nuliais ir pakenkti modeliui.

3.2.3. LbfgsMaximumEntropyMulticlassTrainer

Ši klasė atlieka spėjimus naudodama maximum entropy klasifikaciją, ištreniruotą L-BFGS metodu. L-BFGS - optimizacijos algoritmas, priklausantis quasi-Niutono metodams, kuris apvalina BFGS naudodamas nedidelį kiekį kompiuterio atminties. Šis algoritmas neskaičiuoja pilnos Hesijo matricos, o naudoja apytiksles jos reikšmes. Ši klasė taip pat naudoja empirinę rizikos minimizaciją optimizacijos problemai suformuluoti. Verta duomenis normalizuoti arba naudoti kryžminę validaciją, kadangi egzistuoja persimokymo tikimybė.

3.2.4. PairwiseCouplingTrainer

Šis binarinis algoritmas atlieka treniravimą kiekvienai klasių porai[8]. Šios poros nėra surūšiuotos, tačiau tos pačios vertės poros nesikartoja. Kiekvienam binariniui klasifikatoriui duomenų įvesties taškas yra skaitomas pozityviu jei jis atitinka bet kurią reikšmę poroje, ir laikomas negatyviu jei jo poroje neegzistuoja. Šis algoritmas suteikia galimybę ML.NET spręsti lengvesnes problemas efektyviau. Nors minėti algoritmai turi galimybes vykdyti daugiaklasę klasifikaciją, kartais jos neverta naudoti tiesiogiai dėl atminties apribojimų.

3.2.5. NaiveBayesMulticlassTrainer

Naive Bayes yra tikimybinis klasifikatorius kuris gali būti pritaikytas daugiaklasėms problemoms. Kaip galima nuspėti iš pavadinimo, šis algoritmas paremtas Bayes' teorema. Sąlyginei klasei priklausančio pavyzdžio tikimybė gali būti apskaičiuojama remiantis kiekvienos ypatybių derinių grupės imčių skaičiumi. Šis algoritmas yra efektyvus kai ypatybių skaičius ir joms priskirtų reikšmių dydis yra ganėtinai mažas. Nepriklausomumas tarp ypatybių klasėje yra algoritmo atžvilgiu yra būtinas, netgi kai jos iš tikrųjų yra priklausomos viena nuo kitos.

3.2.6. OneVersusAllTrainer

OVA Strategijoe, binarinis klasifikacijos algoritmas yra panaudojamas treniruoti klasifikatorių kiekvienai klasei, kuris atskiria tą klasę nuo kitų likusių. Tada yra padaromas spėjimas naudojant šiuos binarinius klasifikatorius ir pasirenkamas spėjimas su aukščiausia pasitikėjimo reikšme.Svarbu atsiminti, kad OVA visada reikalaus laikinos talpyklos, kadangi šis algoritmas duomenis analizuoja kelis kartus. Šis algoritmas, kaip ir PairwiseCouplingTrainer suteikia galimybę ML.NET efektyviai spręsti lengvesnes problemas efektyviau.

3.3. Duomenų modelio treniravimui pasirinkimas

Kadangi norėjau kad mano sistema būtų pritaikyta elektronikos prekių atsiliepimams, nusprendžiau panaudoti duomenų rinkinį iš ko gero populiariausios elektroninės parduotuvės - amazon.com. Amazon leidžia atsisiųsti duomenų rinkinius apie naudotojų atsiliepimus skirtingoms parduotuvės kategorijoms. Atsisiųstas duomenų rinkinys yra gana didelis (1.6 GB, 3091104 eilučių). Atsirinkti reikalingus "star_rating" ir "review_body" stulpelius su duomenimis pasinaudojau PowerShell Import-Csv ir Export-Csv komandomis, kadangi Microsoft Excel limitas šiam failui yra milijonas eilučių, o kitos, panašaus pobūdžio programinės įrangos nesugebėjo šio failo atidaryti.

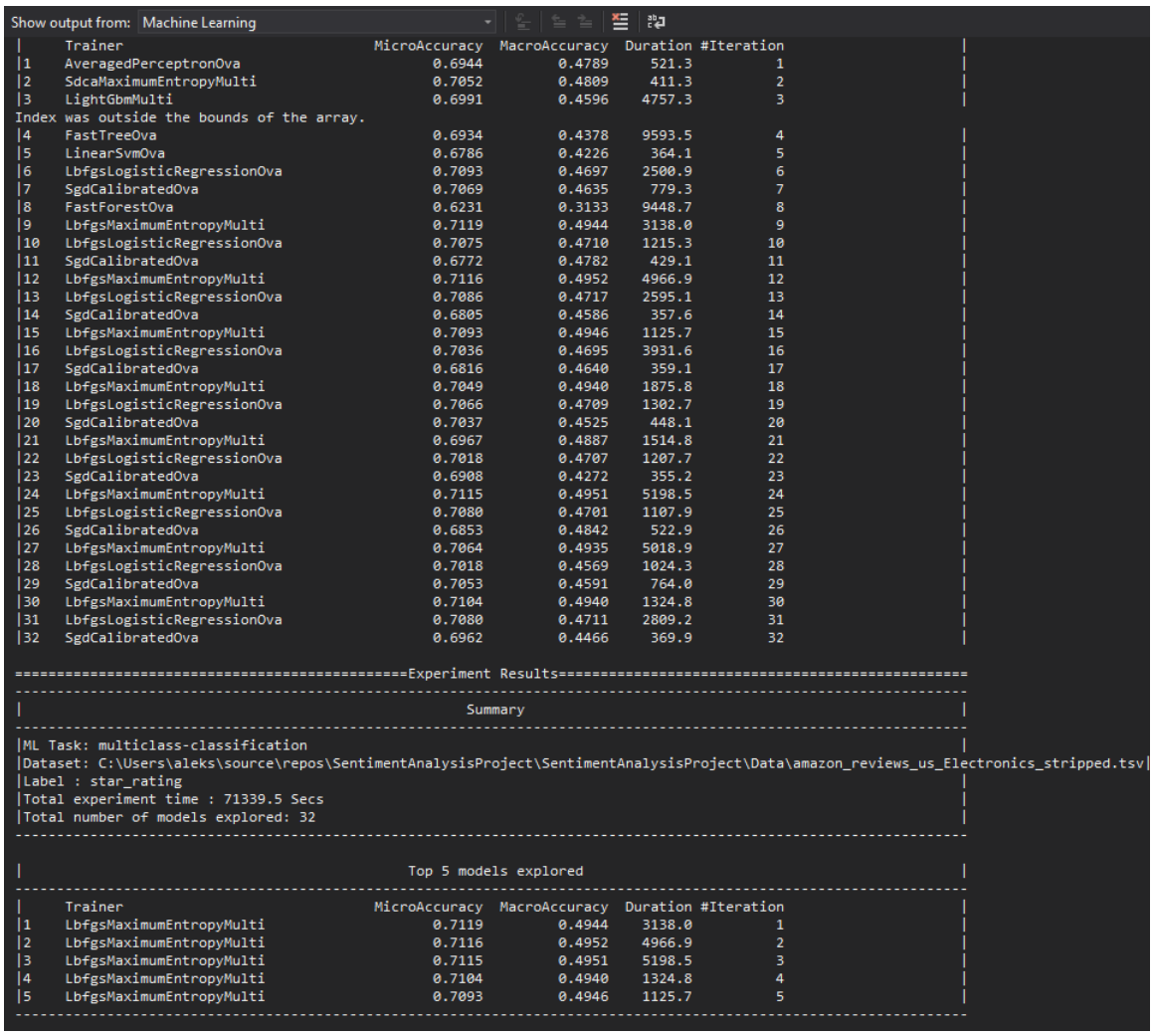
3.4. Modelio efektyvumo metrikos

- Micro accuracy average - skaičiuoja visų klasių grąžinamas reikšmes ir suveda jas į vidurkį.
- Macro accuracy average - skaičiuoja metrikas nepriklausomai kiekvienai klasei ir gražina vidurkį. Šiuo atveju visos klasės laikomos lygiavertiškos.
- Precision - procentas visų reikšmių, kurios yra aktualios.
- Recall - procentas visų reikšmių, kurios yra teisingai suklasifikuotos naudojamo algoritmo.
- LogLoss - skaičiuoja klasifikatoriaus tikslumą, baudžiant už kiekvieną netikslų spėjimą.

3.5. Tinkamiausias modelis

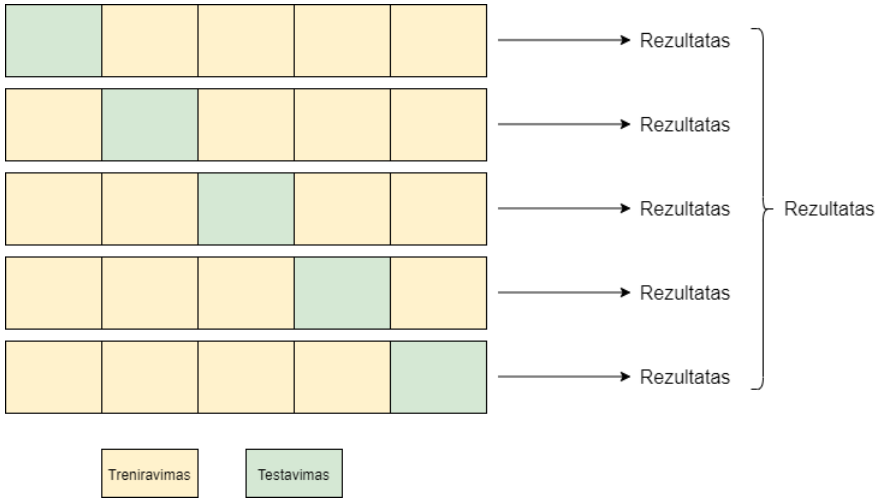
Kadangi turimame duomenų faile atsiliepimai yra vertinami nuo 1 iki 5 balų, šiai užduočiai netikslinga naudoti binarinį klasifikatorių. Norint šiems duomenims naudoti binarinį klasifikatorių reikėtų atsiliepimų balus sugrupuoti į teigiamus arba neigiamus. Žinoma, modelis, ištreniruotas atspėti vieną iš dviejų galimybių bus daug tikslesnis, nei modelis, spėjantis iš penkių. Mano tikslas vis dėlto yra sukurti lankstesnį modelį, kuris gali nuspėti kliento pasitenkinimo laipsnį, todėl pasirinkau mokyti daugiaklasį modelį.

Pasinaudojus ML.Net model builder, 20-čiai valandų paleistas skirtingų algoritmų vertinimas (5 pav.). Per duotas 20 valandų kiekvienas klasifikatorius buvo ištestuotas po kelis kartus. Žiūrint į gautus rezultatus, galima teigti, kad prasčiausiai pasirodė OVA (angl. One versus all - vienas prieš visus) klasifikatorių naudojantys algoritmai. Nuosekliai didžiausią tikslumą išvystė Lbfgs Maximum Entropy Multi algoritmas, kuris pritaiko Maximum Entropy klasifikatorių. Atsižvelgiant į turimus rezultatus modelio treniravimui buvo pasirinktas maksimalios entropijos klasifikatorius.



5 pav. Modelio kūrimo rezultatai

Modelio vertinimui buvo naudojama k-grupių kryžminė validacija (K-Fold Cross Validation). Kryžminė validacija sumaišo duomenis ir sugrupuoja juos į pasirinktą skaičių grupių. Paprastai k-grupių validacija yra atliekama pasiimant vieną grupę testavimui, o kitos k-1 grupės imamos kaip treniravimo duomenys, kuriais naudojantis atliekamos fit ir evaluate funkcijos. Šis procesas pakartojamas kiekvienai turimai grupei. Kryžminė validacija taip pat padeda sumažinti persimokymo (over-fitting) efektus, kadangi testavimo duomenys keičiasi su kiekviena grupe.



6 pav. Kryžminės validacijos pavyzdys, k = 5

Savo modelio vertinimui pasirinkau k = 5. Kad įvertinti modelio treniravimo rezultatus kiekvienai testavimo grupei k sugeneravau po klaidų matricą.

```
============== Cross-validating to get model's accuracy metrics ==============
*******************************************************************************************************************
*       Metrics for Multi-class Classification model
*-------------------------------------------------------------------------------------------------------------------
*       Average MicroAccuracy:    0.712  - Standard deviation: (.001)  - Confidence Interval 95%: (.001)
*       Average MacroAccuracy:    0.495  - Standard deviation: (.001)  - Confidence Interval 95%: (.001)
*       Average LogLoss:          .743  - Standard deviation: (.001)  - Confidence Interval 95%: (.001)
*       Average LogLossReduction: .398  - Standard deviation: (.001)  - Confidence Interval 95%: (.001)
*******************************************************************************************************************

Confusion table
          ||========================================
PREDICTED ||     0 |     1 |     2 |     3 |     4 | Recall
TRUTH     ||========================================
        0 || 332,747 | 3,988 | 15,842 |   690 | 2,931 | 0.9342
        1 || 6,927 | 55,808 | 1,372 | 4,477 | 3,062 | 0.7789
        2 || 65,118 | 2,638 | 31,796 | 1,086 | 6,693 | 0.2962
        3 || 5,991 | 14,635 | 2,674 | 6,333 | 5,962 | 0.1779
        4 || 13,288 | 6,596 | 10,756 | 3,780 | 13,510 | 0.2819
          ||========================================
Precision ||0.7846 |0.6670 |0.5092 |0.3870 |0.4201 |


Confusion table
          ||========================================
PREDICTED ||     0 |     1 |     2 |     3 |     4 | Recall
TRUTH     ||========================================
        0 || 332,934 | 4,051 | 15,900 |   717 | 2,863 | 0.9340
        1 || 6,995 | 55,922 | 1,285 | 4,350 | 3,006 | 0.7815
        2 || 64,407 | 2,722 | 32,016 | 1,044 | 6,727 | 0.2995
        3 || 5,907 | 14,668 | 2,652 | 6,285 | 6,136 | 0.1763
        4 || 13,239 | 6,594 | 10,739 | 3,753 | 13,505 | 0.2824
          ||========================================
Precision ||0.7862 |0.6661 |0.5115 |0.3892 |0.4189 |


Confusion table
          ||========================================
PREDICTED ||     0 |     1 |     2 |     3 |     4 | Recall
TRUTH     ||========================================
        0 || 331,649 | 3,976 |   668 | 15,940 | 2,881 | 0.9339
        1 || 7,060 | 56,063 | 4,431 | 1,372 | 3,053 | 0.7789
        2 || 6,019 | 14,560 | 6,375 | 2,633 | 6,239 | 0.1779
        3 || 65,069 | 2,635 | 1,078 | 31,946 | 6,988 | 0.2966
        4 || 13,136 | 6,506 | 3,787 | 10,627 | 13,779 | 0.2881
          ||========================================
Precision ||0.7842 |0.6695 |0.3902 |0.5110 |0.4183 |


Confusion table
          ||========================================
PREDICTED ||     0 |     1 |     2 |     3 |     4 | Recall
TRUTH     ||========================================
        0 || 331,617 | 3,909 | 15,926 |   617 | 2,864 | 0.9343
        1 || 6,839 | 55,268 | 1,256 | 4,295 | 3,100 | 0.7811
        2 || 65,055 | 2,629 | 31,853 | 1,057 | 6,835 | 0.2965
        3 || 5,950 | 14,795 | 2,691 | 6,318 | 6,232 | 0.1756
        4 || 13,181 | 6,460 | 10,379 | 3,592 | 13,578 | 0.2877
          ||========================================
Precision ||0.7846 |0.6654 |0.5129 |0.3979 |0.4164 |


Confusion table
          ||========================================
PREDICTED ||     0 |     1 |     2 |     3 |     4 | Recall
TRUTH     ||========================================
        0 || 333,115 | 16,042 |   649 | 2,868 | 4,042 | 0.9338
        1 || 64,833 | 31,558 | 1,026 | 6,926 | 2,688 | 0.2948
        2 || 6,066 | 2,565 | 6,410 | 6,116 | 14,822 | 0.1782
        3 || 13,141 | 10,602 | 3,802 | 13,462 | 6,601 | 0.2828
        4 || 7,035 | 1,271 | 4,302 | 3,144 | 56,134 | 0.7809
          ||========================================
Precision ||0.7853 |0.5087 |0.3959 |0.4140 |0.6660 |
```

7 pav. Modelio kūrimo metrikos, klaidų matricos

Čia indeksų reikšmės:

Indeksas	Atsiliepimo vertinimas
[0]	5
[1]	1
[2]	4
[3]	2
[4]	3

Iš pavaizduotų matricų galima pamatyti, kad daugiausiai teisingai atspėtų įvertinimų yra 5 balų. Šis rezultatas yra

gana nuspėjamas, kadangi duomenų rinkinyje daugiau nei pusė atsiliepimų (1779426/3091104) yra įvertinti būtent 5 balais. Pakankamai didelis kiekis teisingai atspėtų ir 1 balo atsiliepimų, tačiau įdomu pastebėti kad 5 grupės matricoje ženklus kiekis 1 balo atsiliepimų buvo neteisingai nuspėti kaip 5. Nenaudojant kryžminės validacijos tokį neatitikimą būtų sunku išanalizuoti ir padaryti teisingas išvadas apie modelį. Atsižvelgus į mikro ir makro neatitikimus, galima spėti kad duomenų rinkinyje vis dėl to gali būti per daug 5 balų atsiliepimų. Ar verta šiuo atveju atsižvelgti į LogLoss funkcijos rezultatus sunku pasakyti, kadangi ji stipriai baudžia menkiausius neatitikimus, kurie daugiaklasėje klasifikacijoje įvyksta gana dažnai, kadangi yra vertinamas modelis su net penkiomis klasėmis. Pavyzdžiui, jei tikrasis atsiliepimo įvertinimas yra 3, o modelis spėja, kad reikšmė yra 4, tai jau yra laikoma klaida. Įvertinus bendrus rezultatus manau kad modelio spėjimų tikslumas yra patenkinamas. Kuriamoje atsiliepimų sistemoje yra akcentuojamas prasčiausių atsiliepimų suradimas, kadangi dėl jų reikia imtis greičiausių veiksmų. Tokius atsiliepimus sistema atspėja gana gerai. Šio klasifikatoriaus silpnesnė vieta yra vidutinių vertinimų tikslus klasifikavimas

Norint patikrinti, ar rezultatai nebūtų geresni jei duomenų rinkinyje būtų labiau suvienodintas atsiliepimų skaičius, buvo apytiksliai suvienodintas visų atsiliepimų dažnis. Įvertinus maksimalios entropijos modelio metrikas (7 pav.) apskaičiuotas su atnaujinta duomenų imtimi, galima pasakyti kad modelis buvo apmokytas tolygiau. Makro tikslumas šiek tiek pakilo, tačiau mikro tikslumas ženkliai krito. Taip pat padidėjo ir LogLoss reikšmė. Vertinimų spėjimų išsidėstymas matricoje panašus į modelio su dominuojančiais 5 balų atsiliepimais. Dažniausiai čia teisingai atspėjamos 1 ir 5 balų reikšmės, o 2,3,4 čia vis dar atspėjami ne taip dažnai, tačiau dažniau negu didesniame modelyje.

Indeksas	Atsiliepimo vertinimas
[0]	1
[1]	4
[2]	2
[3]	3
[4]	5

```
=============== Cross-validating to get model's accuracy metrics ===============
*******************************************************************************************************************
*       Metrics for Multi-class Classification model
*-------------------------------------------------------------------------------------------------------------------
*       Average MicroAccuracy:    0.589  - Standard deviation: (.001)  - Confidence Interval 95%: (.001)
*       Average MacroAccuracy:    0.54  - Standard deviation: (.001)  - Confidence Interval 95%: (.001)
*       Average LogLoss:          .99  - Standard deviation: (.001)  - Confidence Interval 95%: (.001)
*       Average LogLossReduction: .375  - Standard deviation: (.001)  - Confidence Interval 95%: (.001)
*******************************************************************************************************************

Confusion table
             ||========================================
PREDICTED    ||     0 |     1 |     2 |     3 |     4 | Recall
TRUTH        ||========================================
           0 || 58,803 | 1,077 | 5,060 | 5,090 | 1,553 | 0.8215
           1 || 2,155 | 22,218 |   843 | 9,272 | 14,606 | 0.4526
           2 || 15,973 | 1,754 | 7,084 | 9,509 | 1,166 | 0.1996
           3 || 7,865 | 8,948 | 4,469 | 23,267 | 3,199 | 0.4873
           4 || 1,868 | 9,991 |   360 | 2,601 | 42,000 | 0.7392
             ||========================================
Precision    ||0.6785 |0.5051 |0.3976 |0.4678 |0.6717 |


Confusion table
             ||========================================
PREDICTED    ||     0 |     1 |     2 |     3 |     4 | Recall
TRUTH        ||========================================
           0 || 59,022 | 1,033 | 4,858 | 5,045 | 1,510 | 0.8259
           1 || 2,134 | 21,937 |   802 | 9,373 | 14,978 | 0.4457
           2 || 16,095 | 1,690 | 7,176 | 9,660 | 1,157 | 0.2006
           3 || 7,827 | 8,728 | 4,497 | 23,230 | 3,296 | 0.4883
           4 || 1,923 | 9,708 |   358 | 2,675 | 42,449 | 0.7432
             ||========================================
Precision    ||0.6784 |0.5090 |0.4056 |0.4648 |0.6696 |
```

8 pav. Modelio kūrimo metrikos, sumažinus dominuojančių vertinimų skaičių

3.6. Apmokyto modelio rezultatų palyginimas su kituose tyrimuose pasiektais rezultatais

Norint gauti platesnį vaizdą, kaip pavyko apmokyti turimą modelį, reikia ištirti šia tema jau parašytus darbus.

Rašto darbe "Amazon Review Classification and Sentiment Analysis" [3] yra aprašyta panaši idėja - ištraukti iš Amazon.com atsiliepimų duomenų rinkinį ir pagal jį suklasifikuoti duomenis. Nors idėja ir yra panaši, tačiau užduoties realizavimas yra gana skirtingas. Visų pirma, čia sentimentų analizė yra vykdoma leksikonu paremtu metodu. Kad

gautas modelis būtų tikslus, čia atsiliepimai yra subendrinami į 3 klases - (geri, vidutiniai ir blogi). Mano modelis yra apmokytas penkiomis klasėmis, kas leidžia smulkiau ir tiksliau kategorizuoti atsiliepimus.

"Movie Review Mining: a Comparison between Supervised and Unsupervised Classification Approaches" [4] Darbe vykdytas klasifikatorių mokymas iš filmų atsiliepimų duomenų rinkinio. Pastebėjau, kad darbe tikslumo vertinimas yra taip pat vykdoma kryžmine validacija. Čia buvo susitelkta atsiliepimus klasifikuoti binariškai - į teigiamus arba neigiamus. Ką verta pastebėti, kad mano apmokyto 5 klasių klasifikatoriaus tikslumas yra gana palyginamas - geriausias darbe pasiektas tikslumas yra 77% naudojant semantinį algoritmą 85% su n-gram modeliu, tas tikslumas nukrito iki 66% procentų naudojant testinius duomenis. Darbe taip pat naudojamos gana mažos duomenų imtys ir rankiniu būdu nustatomas ribos tarp gero ir blogo atsiliepimo.

Panagrinėjus kitus mokslinius darbus galima spręsti, kad mano daugiaklasės klasifikacijos modelis yra apmokytas pakankamai gerai.

4. Sistemos praktinė dalis

4.1. Naudojamos technologijos

Atsiliepimų sistema buvo kuriama pasitelkus ASP.NET ir Angular platformas. Naudojamos C#, TypeScript, CSS, HTML kalbos.

Naudojami įrankiai: Microsoft Visual Studio 2019 programavimo aplinka, Microsoft SQL Management Sudio 2018 aplinka darbui su duomenų baze, www.draw.io tinklapis uml diagramų braižymui, Postman API testavimo įrankis.

4.1.1. ASP.NET core

ASP.NET core[2] [9] yra daugiaplatformis, didelio našumo ir atviro kodo karkasas skirtas modernioms, su internetu sujungtoms aplikacijoms kurti. Šis karkasas puikiai integruojasi su Angular, būtent kurį ir pasirinkau naudoti kuriant išorinę sistemą.

4.1.2. Angular

Sistemos išorinei sistemai naudojamas Angular[1]. Angular yra TypeScript kalba paremtas, atviro kodo web aplikacijų karkasas. Jis pasižymi vientisumu visuose programos komponentuose. Dėl šios savybės kodą yra gana lengva taisyti ir prižiūrėti. Kiekvienas Angular komponentas susideda iš 4 dalių: html failo, css stiliaus failo, ts failo ir unit test failo. Aplikacijoje naudojamos bibliotekos ir komponentai yra aprašomi viename faile, kas taip pat padeda kodo skaitomumui ir kitaip palengvina darbą programuotojams, kurie jau yra susipažinę su Angular karkasu.

4.1.3. Angular CLI

Norint užtikrinti programos vientisumą, kiekvieną komponentą galima generuoti naudojant Angular CLI. Angular CLI yra komandinės eilutės įrankis, skirtas palengvinti aplikacijos kūrimą. Juo galima sugeneruoti komponentus, atsisiųsti reikalingus modulius, paleisti pačią aplikaciją, kuri atsinaujina po kiekvieno kodo pakeitimo išsaugojimo.

4.1.4. Angular Material

Angular material yra UI komponentų rinkinys, paremtas Goole "Material" dizainu. Šie komponentai yra optimizuoti naudojimui su Angular, suteikia aplikacijai modernų dizainą.

4.1.5. ML.NET

ML.NET - Kompiuterių mokymo karkasas padedantis integruoti kompiuterių mokymo algoritmus .NET platformoje.

4.1.6. ASP.NET Identity

ASP.NET Identity padeda kontroliuoti sistemos naudotojų autorizaciją ir autentifikaciją. Ši technologija padeda:

- Autentifikuoti naudotojus, kurti autorizacijos žetonus.
- Su Entity Framework padeda sugeneruoti duomenų bazės lenteles naudotojams su naudingais laukais.

- Suteikia pagalbines funkcijas sąveikauti su duomenų baze.
- Koduoti naudotojų slaptažodžius PBKDF2 maišos funkcija.

4.1.7. Entity Framework

Entity Framework (EF) technologija padeda atlikti įvairius veiksmus, susijusius su duomenų bazės sąveika. EF leidžia lengvai ir paprastai rašyti LINQ užklausas, generuoti valdiklius su CRUD operacijom, kurti arba atnaujinti duomenų bazės lenteles pagal jau sukurtą duomenų modelį.

4.1.8. SQL Server

SQL Server yra santykinė duomenų bazės valdymo sistema. Sukurta Microsoft, todėl turi gerą integraciją su .NET platforma.

4.2. Sistemos struktūra

4.2.1. Naudotojo sąsajos sistemos struktūra (front-end)

Aplikacijos išorinė sistema realizuota su Angular, todėl aplikacijos struktūroje kiekvienam komponentui yra sugeneruota po css, html, spec.ts ir ts failą. Dažnai tarp įvairių komponentų naudojamos funkcijos yra iškeltos į servisų failus, kurie yra lengvai pasiekiami iš bet kurio komponento. Būtent iš servisų failų yra aprašytos funkcijos, skirtos vidinės sistemos API kvietimui. Visi naudojami komponentai, navigacijos keliai yra aprašyti app.module faile lengvam prieinamumui.

4.2.2. Vidinės sistemos struktūra (back-end)

Aplikacijos vidinė sistema realizuotas ASP.Net Ši aplikacijos dalis padalinta į modelius (models) ir valdiklius (controllers). Modeliuose yra aprašomos naudojamos klasės ir jų savybės, kurios naudojamos sąveikauti su duomenų baze. Valdikliuose aprašomos funkcijos, kurios yra pasiekiamos per API užklausas iš front-end. Dažniausiai vykdomos CRUD operacijos.

4.2.3. Kompiuterio mokymo projektai

Naudojantis ML.NET sugeneruoti du projektai: projektas, kuriame modeliai, skirti daugiaklasės klasifikacijos duomenų įvedimui ir išvedimui. Taip pat sukurtas modelis skirtas spėjimo pagal ištreniruotą modelį pateikimui kitiems projektams. Taip pat sugeneruota konsolinė aplikacija, iš kurios galima keisti ir iš naujo mokyti klasifikacijos modelį.

4.3. Pagrindinės sistemos funkcijos

4.3.1. Autentifikavimas ir autorizavimas

Autentifikavimas vyksta kai naudotojas įveda savo naudotojo vardą ir slaptažodį į prisijungimo formą. "Login" mygtukas yra neaktyvus kol abu laukai nėra užpildyti. Paspaudus mygtuką login yra kviečiama *UsersController* esanti *CreateToken*. Šioje funkcijoje yra tikrinama ar teisingai įvestas naudotojo vardas ir slaptažodis. Jei toks naudotojas iš tiesų egzistuoja duomenų bazėje, yra sukuriamas Jwt žetoną su 30 minučių galiojimo laiku ir naudotojas nukreipiamas į atsiliepimų vaizdavimo langą.

Visi sistemos navigacijos keliai aprašyti Angular app.module išskyrus "login" ir "register" yra apsaugoti *AuthGuard* serviso, kuris tikrina ar egzistuoja galiojantis žetonas. Jei galiojančio žetono nėra, apsaugoti navigacijos keliai naudotojui nėra prieinami. vidinės sistemos valdiklių funkcijos yra apsaugotos asp.net [Authorize] atributu, todėl API yra prieinamas tik prisijungusiems naudotojams.

4.3.2. Registracija

Prisijungimo ekrane paspaudus mygtuką "Register" atsidaro registracijos forma. Šią formą sudaro šie laukai:

- Name - privalomas laukas naudotojo vardui nurodyti.
- Email - privalomas laukas naudotojo elektroniniam paštui.

- Phone Number - naudotojo telefono numeris.
- Preferred contact method - priimtiniausias naudotojui susisiekimo būdas.
- UserName - privalomas laukas naudotojo prisijungimo vardui.
- Password - privalomas laukas slaptažodžiui. Yra tikrinama ar slaptažodis turi bent 6 simbolius, nors 1 skaitmenį, nors 1 didžiąją raidę, nors 1 mažąją raidę, nors 1 specialų simbolį. Neatitikimai vaizduojami po lauku.
- Confirm Password - privalomas laukas slaptažodžio patvirtinimui. Yra tikrinama ar lauko reikšmė sutampa su "Password" reikšme. Neatitikimas vaizduojamas po lauku.

Mygtukas "Register" yra neaktyvus kol nėra įvykdytos visos validacijos taisyklės. Prieš naudotojo sukūrimą yra patikrinama, ar nėra pažeistas El. pašto ir naudotojo vardo unikalumas. Jei duomenų bazėje tokios reikšmės jau egzistuoja, naudotojas nėra sukuriamas, o klaidos pranešimas yra pavaizduojamas ekrane. Prieš naudotojo išsaugojimą duomenų bazėje jo slaptažodis užkoduojamas naudojant PBKDF2 maišos funkciją.

4.3.3. Naudotojų valdymas

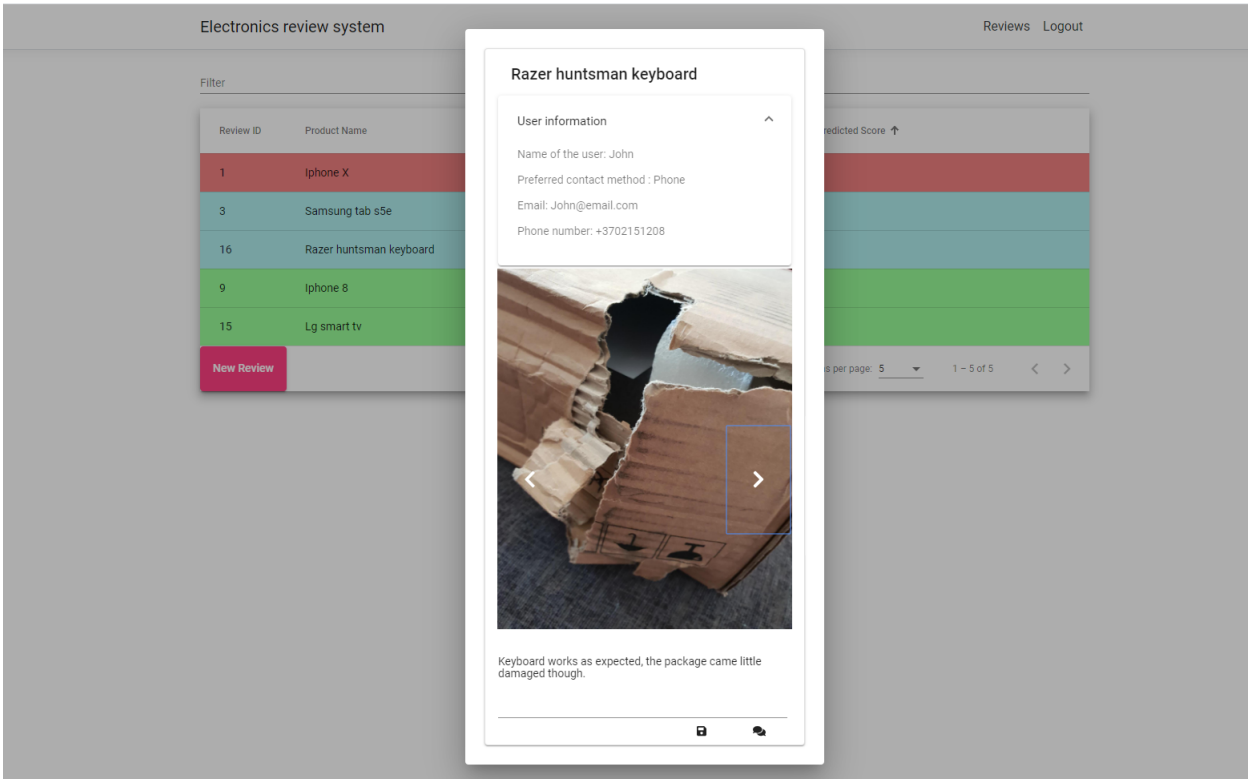
Tik administratoriui prieinamas puslapis, kuriame pavaizduojama lentelė su sistemoje esančiais naudotojais. Lentelės duomenys supuslapiuoti, juos galima filtruoti, kiekvieną stulpelį galima rūšiuoti didėjimo arba mažėjimo tvarka. Lentelėje pavaizduota: naudotojo ID, naudotojo vardas, elektroninis paštas, naudotojo tipas, telefono numeris. Administratorius taip pat turi galimybę trinti bet kurį naudotoją, išskyrus save. Paspaudus mygtuką "New User" atsidaro naujo naudotojo kūrimo forma. Šią formą sudaro 5 privalomi laukai:

- User Type - naudojo tipas. Galima pasirinkti tarp "Review Analyst" ir "Admin". Paprastas naudotojas registruojasi pats, o "Review Analyst" ir "Admin" yra kuriami per administratoriaus panelę.
- Email - naudotojo elektroninis paštas.
- UserName - naudotojo prisijungimo vardas.
- Password - slaptažodis. Yra tikrinama ar slaptažodis turi bent 6 simbolius, nors 1 skaitmenį, nors 1 didžiąją raidę, nors 1 mažąją raidę, nors 1 specialų simbolį. Neatitikimai vaizduojami po lauku.
- Confirm Password - Slaptažodžio patvirtinimas. Yra tikrinama ar lauko reikšmė sutampa su "Password" reikšme. Neatitikimas vaizduojamas po lauku.

Mygtukas "Create" yra neaktyvus kol visos validacijos taisyklės nėra įvykdytos. Taip pat prieš naudotojo sukūrimą yra patikrinama, ar nėra pažeistas El. pašto ir naudotojo vardo unikalumas. Jei duomenų bazėje tokios reikšmės jau egzistuoja, naudotojas nėra sukuriamas, o klaidos pranešimas yra pavaizduojamas ekrane. Prieš naudotojo išsaugojimą duomenų bazėje jo slaptažodis užkoduojamas naudojant PBKDF2 maišos funkciją.

4.3.4. Atsiliepimo peržiūra

Atsiliepimo lentelėje ant atsiliepimo paspaudus mygtuką "More..." atsidaro modalus langas su išsamesne informacija apie atsiliepimą ir informaciją apie jį palikusį naudotoją. Šį langą sudaro produkto pavadinimas, išsiplečianti panelė su informaciją apie naudotoją, įkeltos nuotraukos (jei įkeltų nuotraukų nėra, rodoma numatyta) ir pats atsiliepimas. Atsiliepimą redaguoti gali tik jį sukūręs naudotojas, todėl tik atitikus šią sąlygą yra aktyvuojamas atsiliepimo turinio lauko redagavimas ir vaizduojamas išsaugojimo mygtukas. Nuspaudus išsaugojimo mygtuką yra padaroma užklausa į *ReviewsController PutReview* asinchroninę funkciją.



9 pav. Atsiliepimo peržiūra

4.3.5. Žinutės palikimas

Prisijungę naudotojai vieni kitam gali palikti žinutes, susijusias su specifiniu atsiliepimu. Paspaudus mygtuką su žinučių burbuliukų piktograma šalia atsiliepimo atsidaro naujas langas, kuriame vaizduojamos visos iki tol gautos žinutės. Kiekvieną žinutę sudaro:

- Žinutės turinys
- Sukūrimo data

Žinutės spalva ir pozicija priklauso nuo to, kokio tipo naudotojas yra prisijungęs prie sistemos. Naudotojas savo žinutes mato surikiuotas kairėje pusėje mėlynos spalvos stačiakampiuose, o gautas žinutes mato surikiuotas dešinėje, baltos spalvos stačiakampiuose. Jei į gautą žinutę dar nėra atsakyta, prie mygtuko yra vaizduojamas šauktukas - pritraukti naudotojo dėmesiui.

Jei naudotojui, sukūrusiam atsiliepimą, yra paliekama nauja žinutė, be to, kad yra daroma užklausa į *MessagesController PostMessage* funkciją, dar yra kviečiama ir *EmailController sendEmail* funkcija. Ši funkcija išsiunčia elektroninį laišką naudotojo registracijos metu nurodytą elektroninį adresą, su informacija, kad prie atsiliepimo yra palikta nauja žinutė. El. Laiške taip pat yra nurodytas ir atsiliepimo id, kad naudotojas žinotų, prie kurio atsiliepimo yra palikta žinutė. Elektroninių laiškų siuntimui naudojama openpop biblioteka.

4.3.6. Užrašo sukūrimas

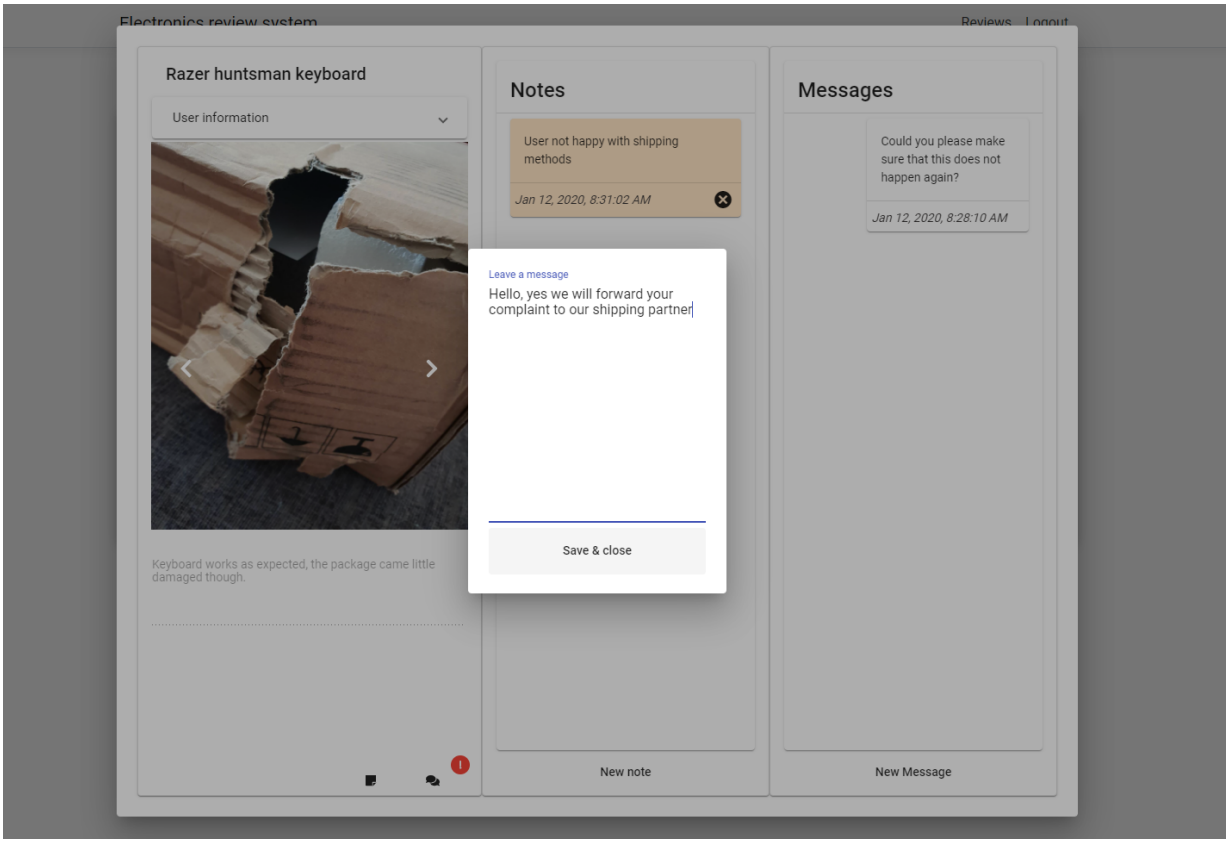
Jei prisijungusio naudotojo tipas yra "Administratorius" arba "Atsiliepimų analitikas", paspaudus mygtuką su užrašų lapelio piktograma atsidaro langas, kuriame yra vaizduojami visi apie tą atsiliepimą palikti užrašai. Kiekvieną užrašą sudaro:

- Užrašo turinys
- Sukūrimo data
- Mygtukas užrašui ištrinti

Naudotojas taip pat gali pridėti naują užrašą paspaudęs mygtuką "New note" ir atsidariusiame lange įvedant norimą turinį. Užrašas išsaugomas paspaudus "Save & close" mygtuką, jei turinio ilgis yra didesnis už 0.

Užrašų gavimas, kūrimas ir trynimas vyksta darant API užklausas į *NotesController* asinchronines funkcijas:

- Gavimas - *GetNoteByReview*. Ištraukia iš duomenų bazės užrašus, kurių "ReviewId" sutampa su dabartinio atidaryto atsiliepimo Id.
- Trynimas - *DeleteNote*. Trina užrašą iš duomenų bazės pagal nurodytą užrašo id.
- Kūrimas - *PostNote*. Užrašas įdedamas į duomenų bazę.



10 pav. Užrašų vaizdavimas, žinutės kūrimas

4.3.7. Atsiliepimo kūrimas

Prisijungusiam naudotojui su role "User" suteikiama galimybė pridėti naują atsiliepimą. Atsiliepimų lentelės apatiniame kairiame kampe paspaudus mygtuką "New Review" yra atidaromas modalus langas su atsiliepimo sukūrimo forma, kuri yra suskirstyta į keturis žingsnius:

- What is the name of the product? - prašoma įvesti produkto pavadinimą.
- What is your opinion of the product? - prašoma įvesti atsiliepimą apie produktą.
- Would you like to upload some images? - prašoma įkelti nuotraukas, jei naudotojas pageidauja.
- Submit - atsiliepimas yra sukuriamas.

Paspaudus mygtuką "Submit" sistema patikrina, ar būtinuose laukuose (produkto pavadinimo ir produkto atsiliepimo turinio) yra įvesti duomenys. Jei duomenys yra įvesti teisingai, yra padaromos dvi post užklausos - viena į *ReviewsController*, kita į *ImagesController*.

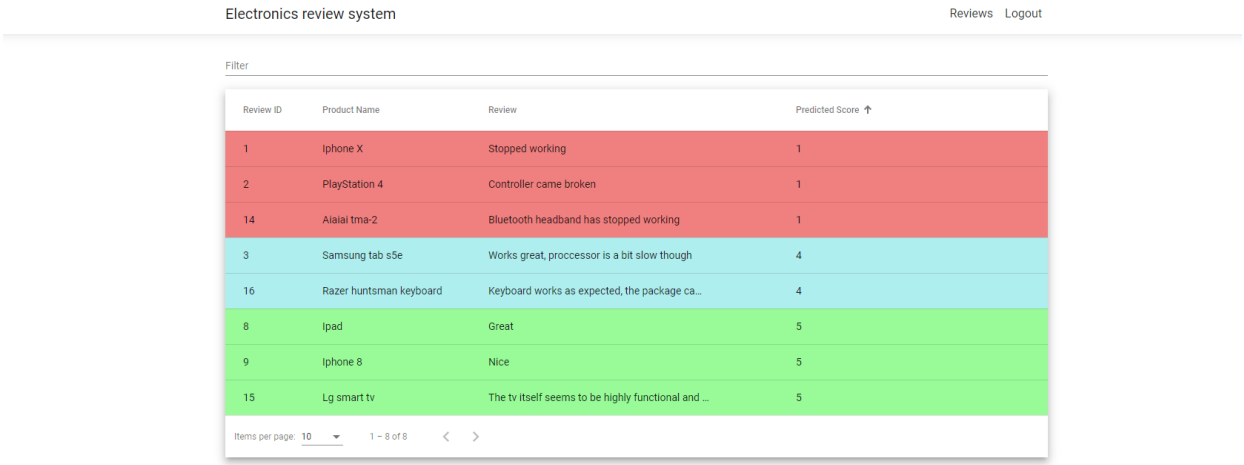
Į *ReviewsController* padaryta užklausa kviečia asinchroninę funkciją *PostReview*. Prieš išsaugant gautą įrašą duomenų bazėje čia dar yra kviečiama apmokyto daugiaklasės klasifikacijos modelio spėjimo funkcija, paduodant atsiliepimo turinį. Gautas atsakymas yra priskiriamas į "PredictedLabel" kintamąjį ir tik tada visas objektas yra išsaugomas duomenų bazėje.

Į *ImagesController* padaryta užklausa kviečia asinchroninę funkciją *PostImage*, kurioje nuotraukos duomenys išsaugomi duomenų bazėje.

Kol vyksta atsiliepimo kūrimas naudotojui yra vaizduoja progreso animacija, o kai užklausos yra įvykdytos, parodoma informacinė žinutė.

4.3.8. Atsiliepimų vaizdavimas

Prisijungusiam naudotojui pirmiausiai yra pristatomas *review-list* komponentas, kurio pagrindas yra lentelė su sukurtais atsiliepimais. Lentelės atvaizdavimui panaudotas *mat-table* komponentas. Šis komponentas leidžia tvarkingai sudėti iš API gautus duomenis apie atsiliepimus į material stiliaus lentelę, kuri palaiko puslapiavimą, rūšiavimą, bei filtravimą.



11 pav. Atsiliepimų sąrašas

Jei prisijungusio naudotojo rolė yra "User", tai į vidinę sistemą API yra siunčiama GET tipo užklausa *api/Users/id* į *ReviewsController*, kuriame yra kviečiama asinchroninė funkcija *GetReviewsByUser*. Ši užklausa grąžina tik tuos atsiliepimus, kurie buvo sukurti būtent šio naudotojo. Jei naudotojo rolė yra "Feedback analyst" arba "Administrator" - vėl yra siunčiama GET tipo užklausa *api/Users* į *ReviewsController*, kuriame kviečiama funkcija *GetReviews*. Šiuo atveju yra grąžinami visi duomenų bazėje esantys atsiliepimai.

Lentelė susideda iš keturių stulpelių: Review ID, Product Name, Review ir Predicted score.

- Review ID - unikalus atsiliepimo kodas. Kiekvienam sukurtam atsiliepimui šis skaičius vienetu padidėja.
- Product Name - atsiliepime aprašomo produkto pavadinimas.
- Review - atsiliepimo turinys.
- Predicted score - Ištreniruoto daugiaklasės klasifikacijos modelio nuspėjamas atsiliepimo pozityvumo lygis.

Virš lentelės yra filtravimo laukas, kuris leidžia filtruoti jos turinį. Filtravimas veikia šiuo principu: Kiekvienas masyvo elementas, kuris yra JSON formato, yra konvertuojamas į vieną eilutę, pagal kurią ieškoma sutapimų su į filtravimo lauką įrašytu žodžiu. Pavyzdžiui, objektas {"reviewId": 1,"productName": "Iphone X", "reviewBody": "Works as intended", "predictedLabel": 5} yra paverčiamas į "1iphonexworks as intended5". Jei filtravimo lauke yra įrašomas žodis "iphone" arba "intended", tai sutapimas bus surastas ir eilutė palikta lentelėje. Tačiau jei sutapimas nebus rastas, eilutė iš lentelės bus pašalinta.

Po lentele yra integruotas puslapiavimas, kuris leidžia pasirinkti kiek rodyti atsiliepimų puslapyje, rodo kuriame puslapyje naudotojas yra dabar, bei leidžia naviguoti tarp jų. Šis *mat-paginator* komponentas puikiai integruojasi su duomenimis, kurie yra paduoti kaip *MatTableDataSource mat-table* komponente.

Lentelėje taip pat yra implementuotas rūšiavimas. Kiekvieną lentelės stulpelį naudotojas gali rūšiuoti didėjančia arba mažėjančia tvarka. Norint funkcionalumą realizuoti reikia pridėti *matSort* direktyvą į lentelę ir prie kiekvieno header elemento pridėti *mat-sort-header*. Ši direktyva taip pat puikiai integruojasi su *MatTableDataSource*. Kad būtų lengviau atskirti ir analizuoti prasčiausius atsiliepimus jie lentelėje yra pažymėti raudona spalva, o kad naudotojas pamatytų juos pirmiausiai, lentelė iš pat pradžių yra rikiuojama pagal prasčiausią sentimentų analizės modelio priskirtą rezultatą.

Išvados ir rekomendacijos

Darbo atlikimo metu analizuotos panašios sistemos, klasifikacijos algoritmai, klasifikatoriai ir apmokytas kompiuterio mokymo modelis padėjo pasiekti išsikeltą darbo tikslą - sukurti veikiančią klientų atsiliepimų sistemą. Geriausius rezultatus, atlikus modelių vertinimus, parodė maksimalios entropijos klasifikatorius, kurio mikro tikslumas buvo 0.7119, o makro - 0.4944. Išanalizavus klaidų matricas galima teigti, kad dažniausiai atspėjami yra 5 balų įvertinimai, taip pat gana aukštu tikslumu spėjami ir prasčiausi - 1 balo atsiliepimai. Sudėtingiau modeliui sekėsi spėti vidutiniškus įvertinimus. Klasifikatorių galima būtų pagerinti panaudojus didesnę duomenų imptį jo treniravimui. Pasitelkus nagrinėtas technologijas buvo realizuotos ir pagrindinės planuotos sistemos funkcijos - naudotojams suteikta galimybė įkelti, redaguoti peržiūrėti savo atsiliepimus bei susisiekti su analitiku palikus jam sisteminę žinutę. Atsiliepimų analitikams suteikta prieiga prie visų sukurtų atsiliepimų, kurių tūrinių pozityvumas įvertintas ištreniruoto klasifikacijos modelio. Leidžiama peržiūrėti kiekvieną atsiliepimą, prie jo palikti užrašus, palikti naudotojui žinutę. Administratoriui suteiktos analitiko teisės, taip pat leidžiamas sistemos naudotojų valdymas. Apžvelgus atliktus darbus, galima teigti, kad šiame darbe išsikelti uždaviniai buvo pasiekti.

Norint pagerinti esamą sistemą, galima atsižvelgti į šias rekomendacijas:

- Panaudoti didesnį duomenų rinkinį klasifikatorių mokymui pagerinti.
- Įvesti apribojimų apmokant klasifikacijos modelį.
- Iš prieinamų šaltinių reguliariai siųstis naujus atsiliepimų duomenis ir automatizuoti modelių treniravimą nustatytu paros laiku.

Ateities tyrimų planas

Tolimesniems darbams planuojamas toks funkcionalumas:

- Išplėsti elektroninių laiškų gavimo ir siuntimo integraciją. Suteikti sistemos naudotojams galimybę gauti ir rašyti laiškus tiesiogiai iš sistemos.
- Naudinga būtų sukurti funkcionalumą, kuris leidžia analitikam priskirti specifinius atsiliepimus.
- Sistemoje galėtų būti pateiktos įvairios atsiliepimų statistikos vizualizacijos bendram vaizdui apie atsiliepimų pozityvumą sudaryti.
- Leisti importuoti atsiliepimus iš populiarių tinklapių, taip jie visi būtų prieinami vienoje sistemoje.
- Rinkti statistiką apie naudotojus ir analizuoti jų tendencijas.

Literatūros šaltiniai

[1] Angular dokumentacija. https://angular.io/docs.

[2] Asp.net dokumentacija. https://docs.microsoft.com/.

[3] Aashutosh Bhatt, Ankit Patel, Harsh Chheda, and Kiran Gawande. Amazon review classification and sentiment analysis. *International Journal of Computer Science and Information Technologies*, 6(6):5107–5110, 2015.

[4] P. Chaovalit and L. Zhou. Movie review mining: a comparison between supervised and unsupervised classification approaches. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, pages 112c–112c, Jan 2005.

[5] Trevor Hastie Gareth James, Daniela Witten and Rob Tibshirani. "an introduction to statistical learning". Springer-Verlag New York, 2013.

[6] Thomas Finley Taifeng Wang Wei Chen Weidong Ma Qiwei Ye Tie-Yan Liu Guolin Ke, Qi Meng. "lightgbm: A highly efficient gradient boosting decision tree.", 2017.

[7] Erica A. Newman John Harte. "maximum information entropy: a foundation for ecological theory". Energy and Resources Group, University of California at Berkeley, 310 Barrows Hall, Berkeley, CA 94720, USA, 2014.

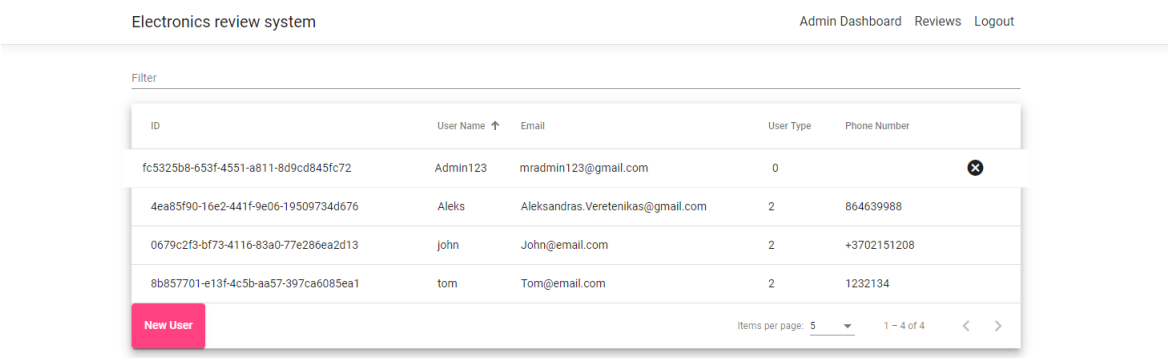
[8] Robert Tibshirani Trevor Hastie. "classification by pairwise coupling ". University of Toronto.

[9] Jack Xu. "practical multiple-page apps with asp.net core and angular elements - building modern multiple-page web applications using asp.net core razor pages, angular elements, webpack, rxjs, and mini-spas". UniCAD Publishing, New York, USA, 2019.

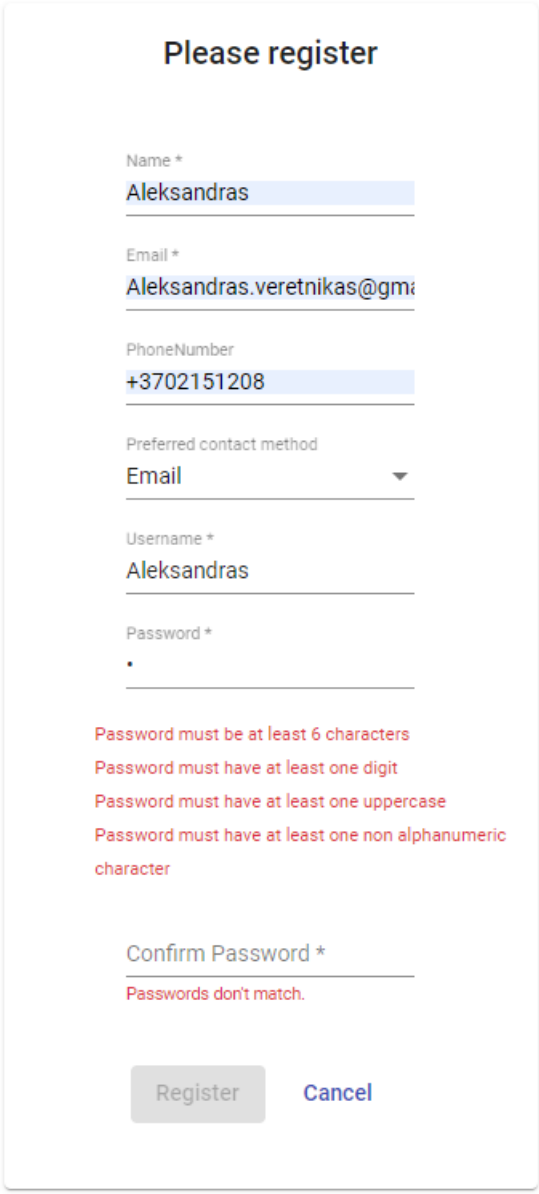
Priedai

.

A. Administratoriaus panelė



B. Registracija



C. Naujas atsiliepimas