



VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
FUNDAMENTINIŲ MOKSLŲ FAKULTETAS
INFORMACINIŲ SISTEMŲ KATEDRA

Džiugas Maslenikas

**SU SAUGA SUSIJUSIŲ TEKSTŲ KLASIFIKAVIMAS Į GRĖSMIŲ,
APSAUGOS AR NEUTRALŲ KONTEKSTĄ**

**SECURITY AREA TEXT CLASSIFICATION TO THREAT,
MITIGATION AND NEUTRAL CONTENT**

Baigiamasis magistro darbas

Informacinių technologijų studijų programa, valstybinis kodas 6211BX016

Informatikos inžinerijos studijų kryptis

Vilnius, 2024

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
FUNDAMENTINIŲ MOKSLŲ FAKULTETAS
INFORMACINIŲ SISTEMŲ KATEDRA

Džiugas Maslenikas

**SU SAUGA SUSIJUSIŲ TEKSTŲ KLASIFIKAVIMAS Į GRĖSMIŲ,
APSAUGOS AR NEUTRALŲ KONTEKSTĄ**
**SECURITY AREA TEXT CLASSIFICATION TO THREAT,
MITIGATION AND NEUTRAL CONTENT**

Baigiamasis magistro darbas

Informacinių technologijų studijų programa, valstybinis kodas 6211BX016

Duomenų gavybos technologijų specializacija

Informatikos inžinerijos studijų kryptis

Vadovas _____ Prof. dr. Simona Ramanauskaitė
(Moksl. laipsnis/pedag. vardas, vardas, pavardė)

Lietuvių kalbos konsultantas _____ Dr. Vaida Buivydienė
(Moksl. laipsnis/pedag. vardas, vardas, pavardė)

Vilnius, 2024

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
FUNDAMENTINIŲ MOKSLŲ FAKULTETAS
INFORMACINIŲ TECHNOLOGIJŲ KATEDRA

Studijų kryptis: Informatikos inžinerija

Studijų programa: Informacinės technologijos, valstybinis kodas 6211BX016

Specializacija: Duomenų gavybos technologijos

TVIRTINU

Katedros vedėjas

Dmitrij Šešok

2024-05-16

MAGISTRO BAIGIAMOJO DARBO UŽDUOTIS

Nr. DGTfm-22-7738

Vilnius

Studentas (-ė): Džiugas Maslenikas

Baigiamojo darbo tema: Su sauga susijusių tekstų klasifikavimas į grėsmių, apsaugos ar neutralų kontenstą

Baigiamojo darbo užbaigimo terminas pagal numatytą studijų kalendorinį grafiką.

BAIGIAMOJO DARBO UŽDUOTIS:

Tikslas: Magistro baigiamajame darbe turėtų būti siekiama automatizuoti su sauga susijusių tekstų analizę, klasifikuojant juos į kategorijas pagal paskirtį (apibrėžimas, grėsmės aprašymas, apsaugos priemonių nurodymas ar pan.).

Uždaviniai:

Tikslo pasiekimui, magistro darbe turėtų būti išspręsti šie uždaviniai:

1. Atlikti srities analizę, identifikuojant esamus panašius sprendimus ar jo įgyvendinimui galimas naudoti technologijas.
2. Sudaryti duomenų rinkinį, skirtą tekstinių duomenų analizės modelio apmokymui ir/ar testavimui.
3. Ištirti galimų panaudoti dirbtinio intelekto sprendimų tinkamumą su sauga susijusių tekstų klasifikavimui pagal paskirtį.

Planuojami rezultatai: Atlikus visas užduotis ir aprašius magistro baigiamojo darbo rezultatus, turėtų būti aišku kokią klasifikavimo tikslumą galima pasiekti su pasirinktais dirbtinio intelekto metodais ir sudaryti duomenų rinkiniu.

Kita: Tyrimui rekomenduojama orientuotis į anglų kalba rašytus tekstus iš bent dviejų duomenų šaltinių, kurie pateikia skirtingo tipo tekstinius aprašymus, susijusius su saugos tematika.

Vadovas profesorius Simona Ramanauskaitė

Vilniaus Gedimino technikos universitetas		ISBN	ISSN
Fundamentinių mokslų fakultetas		Egz. sk.	
Informacinių technologijų katedra		Data-.....-.....	

Antrosios pakopos studijų Informacinių technologijų programos magistro baigiamasis darbas 4	
Pavadinimas	Su sauga susijusių tekstų klasifikavimas į grėsmių, apsaugos ar neutralų kontenstą
Autorius	Džiugas Maslenikas
Vadovas	Simona Ramanauskaitė

	Kalba: lietuvių
--	------------------------

Anotacija

Šiame darbe buvo atliktas su sauga susijusių anglišų dokumentų tekstų klasifikavimo į tris grupes tyrimas. Klasės į kurias skirtsomi tekstai yra grėsmių scenarijai, apsaugos priemonės bei apibrėžimai. Tikslui pasiekti buvo padaryta srities analizė, apžvelgti skirtingi duomenų paruošimo būdai, apžvelgti klasifikavimo metodai, tuomet sudarytas tinkamas duomenų rinkinys, pasirinkti metodai ir atliktas pats tyrimas. Buvo sudarytos 21 skirtingos klasifikatorių kombinacijos pritaikant tiek skirtingus duomenų paruošimo būdus, skirtingus požymių išskyrimo metodus, tiek pasirenkant skirtingus klasifikavimo algoritmus, o pažangesniems BERT modeliams taikant skirtingus optimizatorius. Rezultatų analizėje klasifikatoriai buvo lyginami tikslumo ir laiko aspektais, taip buvo išsiaiškinta kokią klasifikavimo tikslumą galima pasiekti su pasirinktais skirtingais klasifikavimo metodais ir sudarytu duomenų rinkiniu.

Darbo apimtis yra 51 p. teksto be priedų. 16 iliustracijos, 8 lentelės, 24 šaltiniai.

Prasminiai žodžiai: klasifikavimas, teskto klasifikavimas, požymių išskyrimo metodai, klasifikavimo metodai, duomenų paruošimas, dirbtinis intelektas

Vilnius Gediminas Technical University		ISBN	ISSN
Faculty of Fundamental Sciences		Copies No.	
Department of Information Technologies		Date-.....-.....	

Master Degree Studies Information Technologies study programme Master Graduation Thesis	
Title	Security Area Texts Classification to Threat, Mitigation and Neutral Content
Author	Džiugas Maslenikas
Academic supervisor	Simona Ramanauskaitė

	Thesis language: Lithuanian
--	------------------------------------

Annotation
In this work, a study was carried out to classify English safety-related texts into three groups: threat scenarios, mitigations, and neutral context. To achieve this objective, a domain analysis was conducted, various data preparation methods were reviewed, and classification methods were evaluated. Subsequently, a suitable dataset was compiled, appropriate methods were selected, and the study was performed. Twenty-one different combinations of classifiers were created, utilizing various data preparation methods, feature extraction techniques, classification algorithms, and for more advanced BERT models, different optimizers. The results were analyzed by comparing the classifiers in terms of accuracy and timing to determine the classification accuracy achievable with the chosen methods and datasets. The thesis consist of 51 pages of text without appendices, 16 illustrations, 8 tables, 24 references.

Keywords: classification, text classification, feature extraction methods, classification methods, data preparation, artificial intelligence

TURINYS

Įvadas.....	10
1. klasifikavimo metodų, algoritmų analizė, jų palyginimo bei panaudojamumo apžvalga	11
1.1. Teksto klasifikavimas ir jo sprendžiamos problemos	11
1.2. Teksto klasifikavimo panaudojamumas	11
1.2.1. Informacijos paieška.....	12
1.2.2. Stebėti pavojingas situacijas socialinėje erdvėje	12
1.2.3. Lengvesnis duomenų valdymas.....	12
1.2.4. Automatinis el. laiškų ar bilietų rūšiavimas	12
1.2.5. Dokumentų apibendrinimas.....	12
1.2.6. Klientų aptarnavimas.....	13
1.2.7. Versle ir rinkodaroje.....	13
1.2.8. Sveikatos sritis.....	13
1.2.9. Socialiniai mokslai	14
1.2.10. Teisė	14
1.3. Saugos standartų svarba.....	14
1.4. Teksto klasifikavimo algoritmo eiga	15
1.4.1. Požymių išskyrimas (angl. Feature Extraction).....	16
1.4.2. Dimensijų mažinimas (angl. Dimensionality Reduction).....	16
1.4.3. Klasifikavimas (angl. Classification)	16
1.4.4. Modelio vertinimas (angl. Evaluation).....	16
1.5. Teksto klasifikavimo uždaviniams skirtų metodų aprašymas	16
1.5.1. Požymių išskyrimo metodai	17
1.5.2. Dimensijų mažinimo metodas	18
1.5.3. Klasifikavimo metodai	18
1.5.4. Įvertinimas	20
1.6. Teksto klasifikavimo metodų palyginimas.....	21
1.6.1. Požymių išskyrimo modelių palyginimas.....	21
1.6.2. Teksto klasifikavimo metodų palyginimas.....	22
1.7. Panašaus tyrimo apžvalga.....	24
2. Tyrimo metodika	25
2.1. Įrankiai reikalingi tyrimui atlikti	25
2.2. Tyrimo etapai.....	27
Duomenų surinkimas.....	27
2.3. Duomenų praplėtimas.....	29
2.4. Duomenų paruošimas	31
2.5. Metodų pasirinkimas	32
2.6. Metodų taikymas	35
2.6.1. Modelių kūrimo variacijos.....	36
2.6.2. Vertinimo variacijos	37
3. Tyrimo rezultatai	39
3.1. Tyrimas naudojant pagrindinį duomenų rinkinį <i>COMBINED</i>	39
3.1.1. Rezultatai taikant naivaus Bajeso metodą	39
3.1.2. Rezultatai taikant daugianarės logistinės regresijos metodą	39
3.1.3. Rezultatai taikant atraminių vektorių mašinos metodą.....	40
3.2. Tyrimas naudojant rinkinį su papildomu paruošimu.....	40
3.2.1. Rezultatai taikant naivaus Bajeso metodą	41

3.2.2. Rezultatai taikant daugianarės logistinės regresijos metodą	41
3.2.3. Rezultatai taikant atraminių vektorių mašinos metodą.....	41
3.3. Tyrimas taikant BERT metodą, bet su skirtingais optimizatoriais	42
3.3.1. Rezultatai taikant AdamW optimizatorių	42
3.3.2. Rezultatai taikant Adam optimizatorių.....	42
3.3.3. Rezultatai taikant RMSprop optimizatorių.....	43
3.4. Gautų rezultatų įžvalgos taikant klasikinius metodus	43
3.4.1. Duomenų paruošimas	43
3.4.2. Klasifikatoriaus pasirinkimas	44
3.4.3. Požymių išskyrimo metodo pasirinkimas.....	45
3.4.4. Geriausių ir prasčiausių tikslumą gavusių klasikinių modelių analizė	47
3.5. Gautų rezultatų įžvalgos taikant pažangųjį BERT modelį	48
Išvados	51
Literatūros sąrašas	52

PAVEIKSLŲ SĄRAŠAS

1 pav. Teksto klasifikavimo algoritmo eiga.....	15
2 pav. Artimiausio kaimyno (K-Nearest Neighbour) modelis 2D erdvėje	19
3 pav. SVM modelis 2D erdvėje.....	19
4 pav. Klasifikatoriaus kūrimo etapai	27
5 pav. Surinktų duomenų iškarpa.....	28
6 pav. Duomenys prieš ir po pritaikyto sinonimų ketimo metodo.....	30
7 pav. Duomenys prieš ir po taikyto vertimo metodo.....	30
8 pav. Naujai sugeneruoti duomenys naudojant GPT-2	31
9 pav. Duomenų paruošimo schema	32
10 pav. Skirtingų metodų taikymų variacijos su skirtingais duomenų rinkiniais	35
11 pav. Painiavos matrica ir metrikų skaičiavimo formulės.....	38
12 pav. Skirtingų metodų taikymo tikslumo rezultatai	44
13 pav. Laiko verčių palyginimai	45
14 pav. Tikslumas gautas su skirtingais požymių išskyrimo metodais naudojant COMBINED	45
15 pav. Tikslumas gautas su skirtingais požymių išskyrimo metodais	46
16 pav. Laiko vertės taikant skirtingus požymių išskyrimo metodus	46
17 pav. Painiavos matricos. Geriausias ir blogiausias rezultatas	47
18 pav. Gautas tikslumas su skirtingais optimizatoriais	49
19 pav. Nuostolių verčių pokytis su skirtingais optimizatoriais	49

LENTELIŲ SĄRAŠAS

1 lentelė. Požymių išskyrimo metodų palyginimas	22
2 lentelė. Teksto klasifikavimo metodų palyginimas	23
3 lentelė. Naivaus Bajeso su duomenų rinkiniu COMBINED rezultatai.....	39
4 lentelė. Daugianarės logistinės regresijos su duomenų rinkiniu COMBINED rezultatai.....	40
5 lentelė. SVM su duomenų rinkiniu COMBINED rezultatai.....	40
6 lentelė. Naivaus Bajeso su paruoštu duomenų rinkiniu rezultatai	41
7 lentelė. Daugianarės logistinės regresijos su paruoštu duomenų rinkiniu rezultatai	41
8 lentelė. SVM su paruoštu duomenų rinkiniu rezultatai	42

SANTRUMPOS

PCA (angl. *Principal Component Analysis*) – pagrindinių komponentų analizė.

SVM (angl. *Support Vector machines*) – atraminių vektorių mašina.

IT – informacinės technologijos.

ISO (angl. *International Organization for Standardization*) – Tarptautinė standartizacijos organizacija.

COBIT (angl. *Control Objectives for Information Technologies*) – Informacinių technologijų kontrolės tikslai.

NIST (angl. *National Institute of Standards and Technology*) – nacionalinis standartų ir technologijų institutas.

TF-IDF (angl. *Term Frequency – Inverse Document Frequency*) – Terminų dažnis - atvirkštinis dokumento dažnis.

BERT (angl. *Bidirectional Encoder Representations from Transformers*) – dvikryptis transformacinio kodavimo būdas.

GloVe (angl. *Global Vectors for Word Representation*) – bendrinių žodžių vektorių reprezentacija.

Adam (angl. *Adaptive Moment Estimation*) – adaptyvus momentų įvertinimas.

AdamW (angl. *Adam with Weight Decay*) – Adam su svorių mažėjimu.

RMSprop (angl. *Root Mean Square Propagation*) – optimizavimo algoritmas su momentu.

GPT (angl. *Generative Pre-trained Transformer*) – generatyvinis iš anksto apmokytas transformatorius.

JSON (angl. *JavaScript Object Notation*) – JavaScript objektų notacija.

NLTK (angl. *Natural Language Toolkit*) – naturalios kalbos įrankių rinkinys.

CPU (angl. *Central Processing Unit*) – centrinio apdorojimo vienetas.

GPU (angl. *Graphics Processing Unit*) – grafinio apdorojimo vienetas.

IVADAS

Skaitmeniniais laikais, kiekvienos įmonės duomenų saugumas neabejotinai yra labai svarbus. Todėl įmonės vis dažniau taiko tam tikrus saugumo standartus. Standartai dažniausiai yra saugojami įmonės viduje, informacinėse sistemose ir kitose aplinkose.

Didėjant su sauga susijusių dokumentų skaičiui, darbuotojams darosi sudėtinga atsirinkti jiems reikalingą informaciją, o įrankio, kuris padėtų išspręsti šią problemą, nėra. Būtent taikant teksto klasifikavimo metodus, galima sukurti automatizuotus įrankius, kurie padėtų greičiau atsirinkti reikiamą informaciją, bei padėtų suskirstyti tekstą į atitinkamas kategorijas.

Problema – šiuo metu nėra aiškaus duomenų rinkinio, kuris tiktų šiam teksto klasifikavimo uždaviniui spręsti. Todėl reikalingi klasifikavimo sprendimai, kurie leistų sudaryti šio klasifikatoriaus modelį.

Tyrimo objektas – oficialių įmonės dokumentų parašytų anglų kalba automatinis klasifikavimas.

Darbo tikslas – automatizuoti tekstinių dokumentų analizę klasifikuojant juos į grėsmių, apsaugos ar neutralų kontekstą.

Šiam tikslui pasiekti buvo iškelti tokie **uždaviniai**:

1. Atlikti srities analizę, apžvelgiant teksto klasifikavimo metodus ir jų panaudojimo galimybes.
2. Sudaryti tyrimo metodiką skirtą kuriamų klasifikavimo metodų variacijoms nustatyti.
3. Sudaryti tinkamą duomenų rinkinį skirtą tekstinių duomenų analizės modelio apmokymui bei testavimui
4. Atlikti tyrimą pritaikant atitinkamus teksto klasifikavimo metodus ir pateikti išvadas.

Temos aktualumas – vis daugiau įmonių pradeda taikyti saugos standartus. Reikalingas automatinis įrankis, kuris padėtų greičiau atsirinkti reikalingą informaciją.

Tyrimo metodika – užsienio mokslinės literatūros šaltinių analizė, taikomų metodų aprašymas.

Darbo rezultatai – turėtų būti aišku kokį klasifikavimo tikslumą galima pasiekti su pasirinktais klasifikavimo ir dirbtinio intelekto metodais ir sudarytu duomenų rinkiniu.

Darbo struktūra – darbą sudaro įvadas, 3 skyriai, išvados, literatūros sąrašas ir priedai. Viso yra 18 poskyrių, kurie skirti klasifikatorių modelių aptarimui, tyrimams bei rezultatų analizei.

1. KLASIFIKAVIMO METODŲ, ALGORITMŲ ANALIZĖ, JŲ PALYGINIMO BEI PANAUDOJAMUMO APŽVALGA

Šiame skyriuje aprašomi ir lyginami įvairūs klasifikavimo metodai bei algoritmai, jų pritaikymo galimybės įvairiose srityse. Apžvalga apims tiek tradicinius, tiek šiuolaikinius modelius. Pateikta palyginimo lentelė parodys, kurie algoritmai labiausiai tinka konkrečių problemų sprendimui.

1.1. Teksto klasifikavimas ir jo sprendžiamos problemos

Tekstas gali būti ypatingai turtingas informacijos šaltinis, bet išgauti įžvalgas ir konstruktyvią susistemintą informaciją, yra gana sudėtinga ir kainuoja daug laiko bei pastangų. Vienas iš būdų kaip greičiau atsirinkti reikiamą informaciją, būtų teksto automatinis suskirstymas į atskiras klases. Būtent šią problemą išspręsti padeda vienas iš mašininio mokymosi (angl. *Machine Learning*) metodų ir teksto gavybos (angl. *Text Mining*) formų – teksto klasifikavimas (angl. *Text Classification*).

Teksto klasifikavimas (dar neretai vadinama ir teksto kategorizavimu) yra procesas, kurio metu tekstas yra automatiškai analizuojamas ir tuomet, taikant mašininio mokymosi algoritmus, priskiriamas sukurtoms grupėms, klasėms ar kategorijoms (priklausomai nuo konteksto). Vien tik dėl didžiulių duomenų atsiradimo ir jų šaltinių tokių kaip socialiniai tinklai, forumai, tinklalapiai, el.laiškai, skaitmeninės bibliotekos, įmonių dokumentai, reglamentai ir t.t., tekstų klasifikavimas tapo nepaprastai svarbus per paskutinius metus (Subasi, 2020).

1.2. Teksto klasifikavimo panaudojamumas

Mašininio mokymo ir dirbtinio intelekto pradžioje, teksto klasifikavimo technikos daugiausiai buvo naudojamos tik informacijos paieškos sistemose. Visgi, laikui bėgant, technologijoms tobulėjant, tekstų klasifikavimas ir kategorizavimas tapo pasaulyje naudojamas daugelyje sričių: medicina, sveikata, socialiniai mokslai, inžinerija ir t.t.

Šioje dalyje aptarsiu kelis pagrindinius teksto klasifikavimo pritaikymus realiame pasaulyje:

1.2.1. Informacijos paieška

Informacijos paieška – tai nestruktūruotų duomenų dokumentų, kurie atitinka informacijos poreikį, paieška dideliuose dokumentų rinkiniuose. Sparčiai augant informacijai internete (ypač, kuri pateikta tekstiniu formatu), teksto klasifikavimas tapo svarbiu tokio duomenų formato tipo tvarkymo metodu. Vieni iš dažniau naudojamų metodų šioje srityje yra: N. Bajeso metodas, atraminių vektorių klasifikatoriai, sprendimų medis (Kowsari, Meimandi, Heidarysafa, Mendu, Barnes, Brown, 2019).

1.2.2. Stebėti pavojingas situacijas socialinėje erdvėje

Teksto klasifikacija plačiai naudojama teisėsaugos institucijų. Jos analizuoja socialinių tinklų įrašus ir pokalbius. Taikydami teksto klasifikavimo įrankius, jie gali aptikti rizikingus pokalbius, filtruoti pagal kalbos skubumą, nustatydami neigiamas emocijas, ar kitokias neįprastas situacijas (Shaip Team, 2022).

1.2.3. Lengvesnis duomenų valdymas

Teksto klasifikavimas palengvina duomenų tvarkymo procesą. Tekstų klasifikavimas naudingas akademinei bendruomenei, mokslininkams, administracijos, valdžios ir teisės specialistams, kai nestruktūruoti duomenys suskirstomi automatiškai į grupes (Shaip Team, 2022).

1.2.4. Automatinis el. laiškų ar bilietai rūšiavimas

Pagalbos centrai gauna didelį kiekį el. laiškų ar bilietai, kuriuose yra nestruktūruotas tekstas. Laiku peržiūrėti, patvirtinti ir nukreipti pas atitinkamus ekspertus yra labai svarbu. Tokiu mastu el. laiškų rūšiavimui reikia daug žmonių, kurie peržiūrėtų ir nukreiptų į tinkamus skyrius, o tai užima daug laiko ir resursų. Naudojant pritaikytą teksto klasifikaciją galima analizuoti gaunamą tekstą ir automatiškai rūšiuoti bei nukreipti turinį į atitinkamus skyrius tolimesniems veiksams atlikti (Microsoft, 2023).

1.2.5. Dokumentų apibendrinimas

Tekstų klasifikavimas naudojamas dokumentams apibendrinti. Apibendrinime nebūtinai privalo būti naudojami žodžiai iš originalaus dokumento, svarbu būtų tinkamai perteikta esminė

mintis. Taip pat, dėl sparčiai didėjančio informacijos kiekio internete, daugiau nei vieno dokumento apibendrinimas tampa irgi vis svarbesnis (Kowsari, Meimandi, Heidarysafa, Mendu, Barnes, Brown, 2019).

1.2.6. Klientų aptarnavimas

Klientai tikisi laiku ir tikslių atsakymų, kai kreipiasi pagalbos į klientų aptarnavimą. Naudojant mašininio mokymosi pagrindu veikiančią teksto klasifikaciją, klientų aptarnavimo komanda gali nukreipti gaunamus užklausimus tinkamiems specialistams. Pavyzdžiui, teksto klasifikatorius aptinka žodį „keitimas“ palaikymo biliete ir siunčia užklausą garantijų skyriui (Amazon Web Services, 2024).

Šioje dalyje aptarsiu kelias pagrindines sritis, kur yra dažniausiai pritaikomi tekstinio klasifikavimo algoritmai:

1.2.7. Versle ir rinkodaroje

Pelningos kompanijos palaipsniui vis daugiau naudoja socialinius tinklus rinkodaros tikslais. Duomenų rinkimas apie klientų poreikius, norus, taip pat apie rinkoje vyraujančias tendencijas iš socialinių tinklų padidina pelną kelis kartus. Būtent tu duomenų surinkimas ir klasifikavimas yra efektyvus įrankis paprasčiau surasti įmonėms savo klientus ir teikti paklausiausias prekes tuo metu (Kowsari, Meimandi, Heidarysafa, Mendu, Barnes, Brown, 2019).

1.2.8. Sveikatos sritis

Didžioji dalis informacijos sveikatos srityje yra pateikiama nestruktūruota arba pasakojamąja forma su dviprasmiškais terminais ir rašybos klaidomis. Tokia informacija privalo būti pasiekiamą akimirksniu per paciento ir gydytojo susitikimą įvairiuose diagnozės nustatymo ir gydymo etapuose. Klinikinis kodavimas, kurį sudaro medicininių diagnozių priskyrimas konkrečioms klasių reikšmėms, gautoms iš didelio kategorijų rinkinio yra sveikatos priežiūros taikymo sritis, kurioje teksto klasifikavimo metodai yra labai vertingi (Kowsari, Meimandi, Heidarysafa, Mendu, Barnes, Brown, 2019).

1.2.9. Socialiniai mokslai

Tekstų klasifikavimas ir dokumentų kategorizavimas vis dažniau taikomas siekiant suprasti žmogaus elgesį. Duomenys daugiausiai yra renkami iš neformalios kalbos (trumpųjų žinučių, socialinės medijos ir t.t.), kurių tyrimuose daugiausiai dėmesio yra skiriama žodžių pasikartojimo dažnumui stebėti (Kowsari, Meimandi, Heidarysafa, Mendu, Barnes, Brown, 2019).

1.2.10. Teisė

Kasmet sukuriami daug naujų teisinių dokumentų. Pagrindinis teisininkų bendruomenės iššūkis yra šių dokumentų kategorizavimas. Šios informacijos gavimas ir automatinis klasifikavimas padeda ne tik teisininkams, bet ir jų klientams (Kowsari, Meimandi, Heidarysafa, Mendu, Barnes, Brown, 2019).

1.3. Saugos standartų svarba

Ankstyvaisiais kompiuterijos laikais, pagrindiniai kompiuteriai turėjo viena procesorių, vienu metu buvo vykdoma tik viena programa. Nebuvo nei bendrų duomenų bazių, nebuvo dalijamasi atmintimi. Apsaugoti tokią aplinką buvo lengva (Solms, 2019).

Kompiuterių revoliucijos metu, atsiradus daugiaprocesinėms kompiuterių sistemoms, atsirado daugybę papildomų techninių saugumo mechanizmų, pavyzdžiui, naudotojo identifikavimas ir autentiškumo nustatymas, atminties išvalymas, prieigos prie duomenų ir kt. Siekiant užtikrinti tinkamą informacijos saugumą šioje aplinkoje, reikia daugiau techninių ir procedūrinių saugumo mechanizmų. Todėl, įmonės pradėjo įvedinėti vidinę informacijos saugumo politiką. Ši politika buvo rinkinys taisyklių ir reglamentų, kur nurodyta, kas yra leistina, o kas ne (Solms, 2019).

Technologijoms toliau plečiantis, įmonės savo kompiuterių tinklus prijungia prie interneto, taip pat prie verslo partnerių tinklų, įmonėje atsiranda didžiulės IT infrastruktūros, prie kurių galima prieiti tiek per vidines sistemas, tiek iš išorės (vartotojų pusės). Todėl, atsiranda tiek didžiulės rizikos, kai duomenys gali būti prarasti, pavogti, tiek daug neaiškumų, kas turi prieigą prie duomenų, kaip duomenys turi būti saugomi ir kaupiami. Atsiradus rizikoms, spragoms bei neaiškumams, įmonė praranda stabilumą, efektyvumą bei pasitikėjimą klientais ir verslo partneriais (Solms, 2019).

Užtikrinti saugią IT aplinką, privaloma įdiegti saugos standartų, taisyklių, reglamentų rinkinius visiems tos sistemos vartotojams, ir tam tikros institucijos privalo pasirūpinti, kad visos šalys jų laikytųsi.

Vieni iš plačiau naudojamų (tarptautinių) saugos standartų šiomis dienomis:

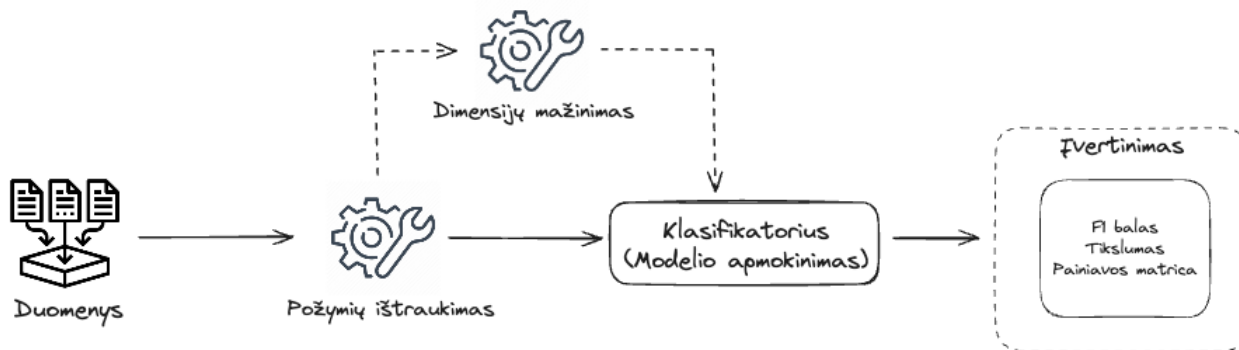
1. ISO 27000 serijos – buvo sukurta tarptautinės standartizacijos organizacijos. Tai lankstus informacijos saugumo karkasas, kuris gali būti naudojamas įvairaus tipo ir dydžio organizacijose (Kirvan, 2021).
2. COBIT – sukurtas fokusuojantis į IT rizikų mažinimą (Kirvan, 2021).
3. NIST CSF – daugiausia dėmesio skiriama rizikos analizei ir rizikos valdymui (Kirvan, 2021).

Organizacija gali taikyti skirtingą kiekį saugos standartų. Tai priklauso nuo įmonės veiklos, turimos IT sistemos kompleksiskumo, vartotojų kiekio, kokius duomenis saugo, kokią prieigą suteikia ir t.t.

Įmonei turint didžiulį kiekį saugos standartų, vartotojams atsiradus poreikiui susipažinti su tam tikra informacija juose, jie privalo viską skaityti, kas gali kainuoti daug laiko ir pastangų. Kur kas paprasčiau būtų, jeigu atsirastų automatinis įrankis, kuris galėtų pakankamai tiksliai išklasifikuoti informaciją esančią tuose standartuose.

1.4. Teksto klasifikavimo algoritmo eiga

Per paskutinius dešimtmečius teksto klasifikavimo problemos buvo plačiai nagrinėjamos ir sprendžiamos tyrėjų. Daugumą teksto klasifikavimo ir dokumentų kategorizavimo sistemų galima suskirstyti į šiuos keturis etapus: požymių išskyrimas, dimensijų mažinimas, klasifikatoriaus parinkimas ir vertinimas (žr. 1 pav.) (Kowsari, et al., 2019).



1 pav. Teksto klasifikavimo algoritmo eiga

1.4.1. Požymių išskyrimas (angl. Feature Extraction)

Beveik visais atvejais, tekstai ir dokumentai yra nestruktūruoti duomenų rinkiniai. Prieš taikant klasifikavimo metodus, nestruktūruoti duomenys turi būti transformuoti į skaitinį pavidalą (struktūruotą požymių aibę). Pirmiausiai reikėtų duomenis išvalyti, tai yra, išimti nereikalingus simbolius ir žodžius. Po valymo galime taikyti požymių išskyrimo metodus (Kowsari, Meimandi, Heidarysafa, Mendu, Barnes, Brown, 2019).

1.4.2. Dimensijų mažinimas (angl. Dimensionality Reduction)

Tekstai ar dokumentų rinkiniai visada turi tam tikrų išskirčių, todėl pirmi duomenų apdorojimo etapai gali vėluoti, striginėti, naudoti daugiau atminties. Dažnu atveju, tokios problemos išsprendžiamos naudojant nesudėtingus, pigius algoritmus. Visgi, ne visada jie pasiteisina, todėl tyrėjai pradeda naudoti dimensijų mažinimo metodus (bando sumažinti gaunamų požymių vektorių matus) (Kowsari, Meimandi, Heidarysafa, Mendu, Barnes, Brown, 2019).

1.4.3. Klasifikavimas (angl. Classification)

Svarbiausias žingsnis teksto klasifikavimo etapuose. Labai svarbu turėti bent konceptualų supratimą apie kiekvieną klasifikavimo algoritmą, kad žinotume, kuris modelis yra pats efektyviausias uždaviniui spręsti (Kowsari, Meimandi, Heidarysafa, Mendu, Barnes, Brown, 2019).

1.4.4. Modelio vertinimas (angl. Evaluation)

Galiausiai labai svarbu išsiaiškinti kaip veikia sukurti klasifikavimo modeliai. Tam yra analizuojamos gautos įvairios modelio vertinimo metrikos.

1.5. Teksto klasifikavimo uždaviniams skirtų metodų aprašymas

Šiame skyrelyje aptarsiu kelis dažniausiai taikomus metodus teksto klasifikavimo algoritmams kurti.

1.5.1. Požymių išskyrimo metodai

Išskiriant požymius, pradinė požymių erdvė paverčiama kompaktiškesne nauja erdve. Visi pradiniai požymiai transformuojami į naują sumažintą erdvę jų neištrinant, bet pakeičiant pradinius požymius mažesne reprezentatyvia aibe. Tai reiškia, kad kai įvesties duomenyse esančių požymių skaičius yra per didelis, kad juos būtų galima apdoroti, tada įvesties duomenys bus transformuojami į sumažintą reprezentacinę požymių aibę (Zareapoor, 2015).

Teksto valymas ir paruošimas (Kowsari, Meimandi, Heidarysafa, Mendu, Barnes, Brown, 2019):

- Žetonavimas (angl. *Tokenization*) – tai metodas, kurio metu teksto srautai yra suskaidomi į frazes, žodžius, simbolius ar kitus reikšmingus elementus vadinamais žetonais. Pagrindinis šio metodo tikslas – ištirti sakinyje esančius žodžius. Pavyzdžiui: „*Draudžiama siųsti ekrano nuotraukas*“ būtų išskaidyta į {„*Draudžiama*“, „*siųsti*“, „*ekrano*“, „*nuotraukas*“}.
- Nereikalingi žodžiai (angl. *Stop Words*) – neturinčių didelės reikšmės žodžių pašalinimas (dalelytės, jungtukai ir t.t.).
- Kapitalizacija (angl. *Capitalization*) – visų raidžių sumažinimas, kad būtų sumažinta duomenų įvarovė ir rezultatai būtų tikslesni.
- Triukšmo šalinimas (angl. *Noise Removal*) – skyrybos ženklų ar specialiųjų simbolių pašalinimas.
- N-gram metodas – šis metodas žetonavimo metodui naudoja nustatytą kiekį žodžių. Pateikiamas 2-gram pavyzdys: „Nėra blogo oro“, po žetonavimo atrodytų {„Nėra blogo“, „blogo oro“}.

Kadangi mašininio mokymosi modeliai geriau veikia su skaitinėmis vertėmis, nei su tekstiniais duomenimis, todėl tekstą visada reikia sustruktūruoti. Tam yra taikomi požymių išskyrimo metodai:

- Žodžių maišas (angl. *Bag of Words*) – naudojantis šia technika, mes sekame ir skaičiuojame pasikartojančius žodžius tekste. Taip galime konvertuoti kintamo ilgio tekstus į fiksuoto ilgio vektorius (Great Learning Team, 2020).
- Terminų dažnis – atvirkštinis dokumento dažnis (angl. *Term Frequency - Inverse Document Frequency*). Kai apskaičiuojame terminų dažnumą, galima pastebėti, kad algoritmas visus raktažodžius vertina vienodai (nesvarbu, jog tai yra nereikšmingas

žodis). Būtent šio metodo pagalba, tokiems nereikšmingiems žodžiams yra suteikiamas mažesnis svoris, o rečiau pasikartojantiems didesnis (Qaiser, 2018).

- Word2Vec – tai algoritmas susidedantis iš dviejų mokymosi modelių, tai yra CBOW (angl. *Continuous Bag of Words*) ir Skip-gram. Kol CBOW modelis nuspėja žodį pagal jo kontekstą, Skip-gram modelis nuspėja kontekstą pagal duotą žodį. Pradžioje algoritmas paskaičiuoja žodžių panašumą ir sudeda juos į klasterius. Sugeneruoti klasteriai yra sudedami į naują dimensiją. Tokie sukurti modeliai vėliau gali atpažinti sinoniminius žodžius (Ma, 2015).
- BERT (angl. *Bidirectional Encoder Representations from Transformers*) tokenizacija yra pažangi priemonė, naudojama natūralios kalbos apdorojimo užduotyse. Skirtingai nuo tradicinių metodų, tokių kaip jau minėtas žodžių maišas, BERT tokenizacija suskaido tekstą į mažesnius segmentus, vadinamus tokenais, atsižvelgiant į kontekstą. Tai leidžia modeliui suprasti ne tik pavienius žodžius, bet ir jų reikšmę sakinyje bei kontekstą. Dėl šios savybės BERT gali geriau suprasti daugiaprasmius žodžius ir frazes, kas padidina modelio tikslumą ir efektyvumą (Kohli, 2023).

1.5.2. Dimensijų mažinimo metodas

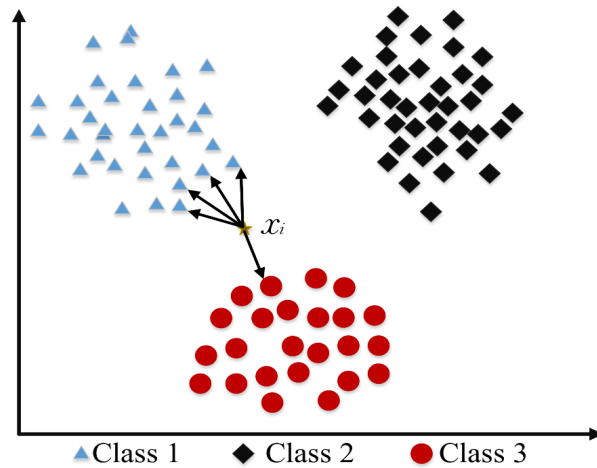
Viena didesnių teksto klasifikavimo problemų, jog yra labai didžiulis skaičius požymių. Didžioji dalis šių požymių yra nereikalingi, nereikšmingi. Jie tik perpildo nereikalingai klasifikatorius ir dėl to modelių tikslumas sumažėja. Taip pat, su didelės dimensijos duomenimis, modelio veikimas gali užtrukti. Vienas iš populiariesnių metodų šioms problemoms išspręsti – PCA (angl. *Principal Component Analysis*):

- PCA – yra įrankis skirtas sumažinti požymių vektorių į žemesnę dimensiją (žemesnius matmenis), bet išlaikant didžiąją dalį informacijos. Viena iš pagrindinių šio metodo idėjų yra rasti kintamųjų poaibį iš didesnio rinkinio, kurie turi didžiausią koreliaciją su vadinamaisiais pagrindiniais komponentais (Taloba, 2018).

1.5.3. Klasifikavimo metodai

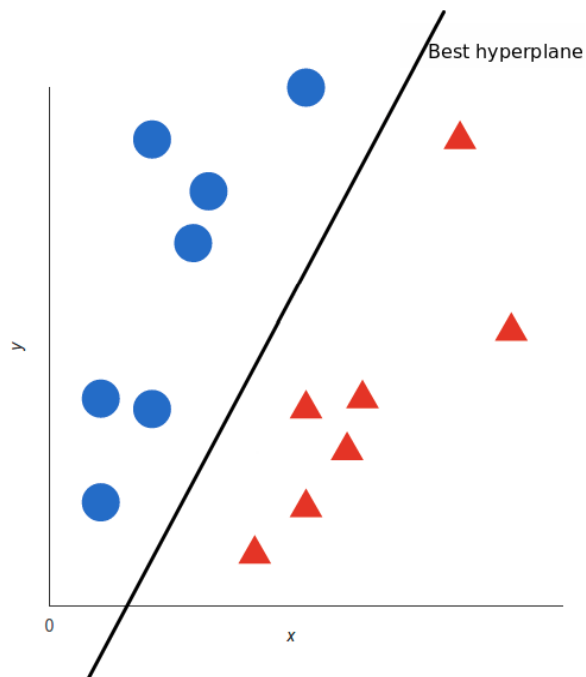
Vienas iš pagrindinių žingsnių visoje teksto klasifikavimo eigoje. Dažnai naudojami klasifikavimo algoritmai:

- Artimiausio kaimyno (angl. K-Nearest Neighbour) – yra metodas skirtas klasifikuoti objektus pagal artimiausius mokymosi pavyzdžius požymių erdvėje (žr. 2 pav.). Šis metodas yra vienas iš pagrindinių, kai išankstinių žinių apie duomenų pasiskirstymą yra mažai arba nėra (Imandoust, 2013).



2 pav. Artimiausio kaimyno (K-Nearest Neighbour) modelis 2D erdvėje

- SVM (angl. *Support Vector Machine*) – tai algoritmas, kuriuo nustatoma geriausia sprendimo riba tarp vektorių (žr. 3 pav.) priklausančių tam tikrai grupei (arba kategorijai), ir vektorių, kurie jai nepriklauso (Text Classification Using Support Vector Machines (SVM), n.d.).



3 pav. SVM modelis 2D erdvėje

- Sprendimų medžiai (angl. *Decision trees*) – šio metodo struktūra yra hierarchinė duomenų erdvės dekompozicija. Pagrindinė idėja - sukurti medį pagal kategorizuotų duomenų taškų požymį (Kowsari, Meimandi, Heidarysafa, Mendu, Barnes, Brown, 2019).
- Naivusis Bajeso klasifikatorius (angl. *Naive Bayes*) – metodas, kuris naudojamas objektams (tekstiniais dokumentams) suskirstyti į dvi ar daugiau klasių, remiantis Bajeso teorema. Bajeso klasifikatorius geriausiai veikia tada, kai kintamieji yra sąlygiškai nepriklausomi.
- Daugianarė logistinė regresija – tai klasifikavimo metodas, kuris naudojamas dažniausiai tekstiniais dokumentams klasifikuoti. Šis metodas išplečia binarinę logistinės regresijos metodą, leidžiantį dirbti su daugiau nei dviem klasėmis (Team, Great Learning, 2023).
- BERT modelio apmokymas – tai procesas, kurio metu BERT (angl. *Bidirectional Encoder Representations from Transformers*) modelis pritaikomas konkrečioms šnekamosios kalbos apdorojimo užduotims. Pradžioje BERT mokomas naudojant didelius tekstų kiekius. Po pirminio mokymo modelis tobulinamas (angl. *fine-tuning*) naudojant jau konkrečios srities duomenis ir užduotis, kad būtų galima tiksliai atlikti specializuotas užduotis, tokias kaip teksto klasifikavimas ar sentimentų analizė (Hashemi-Pour, 2024).

1.5.4. Įvertinimas

Paskutinis etapas teksto klasifikavimo eigoje. Rezultatų apžvelgimas ir modelio įvertinimas. Pagrindinės metrikos norint įvertinti sukurtą modelį:

Tikslumas (angl. *Accuracy*) – procentinė dalis tekstų, kurie buvo priskirti teisingai žymai (Text Classification with Machine Learning & NLP, n.d.).

Tikslumas (angl. *Precision*) – tai procentinė dalis pavyzdžių, kuriuos klasifikatorius išsprendė teisingai pagal tam tikrą žymę iš visų turėtų pavyzdžių (Text Classification with Machine Learning & NLP, n.d.).

Atsiminimas (angl. *Recall*) – procentinė dalis pavyzdžių, kuriuos klasifikatorius nuspėjo pagal tam tikrą žymą, nuo bendro pavyzdžių skaičiaus, kuriuos jis turėjo nuspėti pagal tam tikrą žymą (Text Classification with Machine Learning & NLP, n.d.).

F1 Score – tikslumo ir atsiminimo harmoninis vidurkis (Text Classification with Machine Learning & NLP, n.d.).

Painiavos matrica (angl. *Confusion matrix*) – tai lentelė, arba $N \times N$ matrica (kur N klasių skaičius), naudojama klasifikatorių veikimui įvertinti. Ji rodo teigiamų atvejų (angl. *true positives*), neigiamų atvejų (angl. *true negatives*), klaidingų teigiamų atvejų (angl. *false positives*) ir klaidingų neigiamų atvejų (angl. *false negatives*) skaičius. Painiavos matrica padeda analizuoti modelio veikimą, leidžia apskaičiuoti įvairių metrikų reikšmes bei nustatyti su kuriomis klasėmis modeliui sekasi dirbti sunkiausiai (Bhandari, 2020).

Validacijos nuostolis (angl. *Validation Loss*) – tai metrika, naudojama vertinant modelio veikimą su atskirais validacijos duomenimis, kurie nebuvo naudojami mokymo metu. Šis nuostolis padeda stebėti, ar modelis per daug neprisitaiko (nepertreniruojamas) prie mokymo duomenų, ir ar jis gerai apibendrina naujus duomenis. Mažėjantis validacijos nuostolis rodo, kad modelis gerai apmokytas (aniketguhnn4, 2024).

Validacijos tikslumas (angl. *Validation Accuracy*) – tai metrika, rodanti, kokia dalis validacijos duomenų pavyzdžių buvo teisingai suklasifikuota modelio. Ji padeda įvertinti, kaip gerai modelis veikia su naujais, nematytais duomenimis. Aukštas validacijos tikslumas rodo, kad modelis tiksliai klasifikuoja naujus sakinius (aniketguhnn4, 2024).

1.6. Teksto klasifikavimo metodų palyginimas

Vienas iš teksto klasifikavimo sunkumų yra tinkamo algoritmo pasirinkimas. Prieš pasirenkant tinkamą algoritmą, reikia suprasti kokie yra pagrindiniai tų algoritmų skirtumai, kokia yra jų paskirtis. Šiame skyrelyje bus trumpai išryškinti anksčiau minėtų algoritmų bruožai.

1.6.1. Požymių išskyrimo modelių palyginimas

Šioje dalyje bus lyginami anksčiau minėti požymių išskyrimo modeliai (BoW, TF-IDF, Word2Vec, GloVe bei BERT) (žr. 1 lentelė)

1 lentelė. Požymių išskyrimo metodų palyginimas

Metodai	Privalumai	Apribojimai
Bag of Words (Weighted Words)	– Lengva apskaičiuoti; – Šio modelio pagalba nesudėtinga apskaičiuoti dviejų dokumentų panašumus; – Veikia su nežinomais naujais žodžiais; – Viena iš pagrindinių metrikų, kuri leidžia išskirti labiausiai aprašomus terminus dokumente.	– Nefiksuoja padėties tekste (sintaksės); – Nefiksuoja teksto reikšmės, prasmės (semantika); – Bendriniai žodžiai (dalelytės bei jungtukai) turi poveikį rezultatams.
TF-IDF	– Lengva apskaičiuoti; – Šio modelio pagalba nesudėtinga apskaičiuoti dviejų dokumentų panašumus; – Veikia su nežinomais naujais žodžiais; – Dažnai pasikartojantys žodžiai neturi didžiulės įtakos rezultatams.	– Nefiksuoja padėties tekste (sintaksės); – Nefiksuoja teksto reikšmės, prasmės (semantika).
Word2Vec	– Fiksuoja žodžių padėtį tekste (sintaksė); – Fiksuoja žodžių prasmę (semantika).	– Negali užfiksuoti žodžių (iš dokumentų), kurie nėra žodyno dalis; – Reikalauja didesnės dokumentų kolekcijos, kad tinklas susiliėtų.
GloVe	– Fiksuoja žodžių padėtį tekste (sintaksė); – Fiksuoja žodžių prasmę (semantika); – Efektyvus dideliuose duomenų rinkiniuose.	– Negali užfiksuoti žodžių, kurie nėra žodyno dalis; – Fiksuotas žodyno dydis; – Nefiksuoja konteksto.
BERT	– Fiksuoja žodžių padėtį tekste (sintaksė); – Fiksuoja žodžių prasmę (semantika); – Dviejų kryptų konteksto supratimas.	– Dideli skaičiavimo ištekliai; – Sudėtingas ir lėtas mokymas.

1.6.2. Teksto klasifikavimo metodų palyginimas

Šioje dalyje bus pateikta teksto klasifikavimo metodų palyginimo lentelė su kiekvieno modelio privalumais bei apribojimais (žr. 2 lentelė):

2 lentelė. Teksto klasifikavimo metodų palyginimas

Metodai	Privalumai	Minusai (apribojimai)
Artimiausio kaimyno (angl. <i>K-Nearest Neighbor</i>)	– Veiksminga tekstinių duomenų rinkiniams; – Neparametrinis; – Atsižvelgiama į daugiau lokalių teksto ar dokumento savybių; – Natūraliai apdoroja kelių klasių duomenų rinkinius;	– Šio modelio skaičiavimas labai brangus; – Teksto duomenų rinkiniams sunku rasti prasmingą atstumo funkciją; – Sudėtinga rasti optimalią k reikšmę.
SVM (angl. <i>Support Vector Machine</i>)	– Galima modeliuoti netiesines sprendimų ribas; – Atsparus perteklinio pritaikymo problemoms (ypač, kai duomenys nustatyti aukštoje dimensijos erdvėje).	– Rezultatų aiškumo trūkumas dėl didelio dimensijų skaičiaus; – Neveikia labai gerai, kai duomenų rinkinyje yra daug triukšmo (skyrybos ženklų, nereikalingų simbolių ir pnš.).
Naivaus Bajeso klasifikatorius	– Gerai veikia su tekstiniais duomenimis; – Lengva įgyvendinti; – Greitas, palyginti su kitais algoritmais.	– Naivusis Bajeso metodas daro prielaidą, kad visi prognozuojamieji veiksniai (arba požymiai) yra nepriklausomi; – Atsiranda problemų, kai vieno iš kategorinio kintamojo tikimybė yra nulinio dažnio.
Sprendimų medis	– Palyginus su kitais algoritmais, reikalauja mažiau pastangų duomenų paruošimui; – Gali apdoroti tiek tolydžius, tiek kategorinius kintamuosius.	– Nedidelis duomenų pokytis gali sukelti didelį sprendimų medžio struktūros pokytį ir sukelti nestabilumą; – Dažnai reikalauja daugiau laiko modeliui apmokyti.
Daugianarė logistinė regresija	– Metodas yra paprastas ir lengvai įgyvendinamas naudojant daugelį mašininio mokymosi bibliotekų; – Greitas treniravimas; – Tinka linijinėms problemoms.	– Negali gerai spręsti sudėtingų problemų; – Gali prastai veikti su nelygiu klasių pasiskirstymu.
BERT	– Naudoja dviejų krypčių kontekstą, kas leidžia geriau suprasti žodžių reikšmę kontekste – Dažniausiai aukštas tikslumas; – BERT tokenizacija efektyviai suskaido tekstą į tokenus, kurie gerai veikia su retais ar sudurtiniais žodžiais.	– Dideli skaičiavimo ištekliai; – Sudėtingas ir lėtas mokymas; – Nors BERT puikiai supranta kontekstą, jis gali būti ribotas ilgiems tekstams, nes turi nustatytą maksimalų tokenų skaičių (paprastai 512 tokenų).

1.7. Panašaus tyrimo apžvalga

Straipsnyje „*A Survey on Text Classification: From Traditional to Deep Learning*“ (Li, et al., 2022) yra išsamus teksto klasifikavimo metodų apžvalgos tyrimas. Pagrindinis dėmesys skiriamas tekstiniams duomenims tokie kaip: „*Movie Review*“, „*Stanford Sentiment Treebank*“, „*20 Newsgroups*“, „*AG News*“ ir „*DBpedia*“, kurie vėliau yra analizuojami, bei klasifikuojami.

Tyrimo apžvelgiami tiek tradiciniai, tiek pažangesni modernūs giluminio mokymosi (angl. *Deep Learning*) metodai, naudojami teksto klasifikavimui. Tarp tradicinių metodų galime rasti žodžių maišo metodą (BoW), požymių išskyrimo metodą TF-IDF, klasifikatorius kaip naivusis Bajesas, atraminių vektorių mašina (SVM) ir kiti. Šie metodai buvo plačiai naudojami iki giliojo mokymosi eros pradžios. Tuo tarpu giluminio mokymosi metodai, tokie kaip konvoliuciniai, rekurentiniai neuroniniai tinklai ir transformeriai (pavyzdžiui: BERT, GPT), revoliucionavo teksto klasifikavimo sritį, ženkliai pagerindami modelių efektyvumą bei tikslumą.

Straipsnyje daroma išvada, kad giluminio mokymosi metodai, kurie remiasi transformeriais, yra reikšmingai pagerinę teksto klasifikavimo uždavinių sprendimo efektyvumą bei tikslumą, lyginant su tradiciniais metodais. Transformeriai, tokie kaip BERT ir GPT, leidžia modeliams geriau suprasti kontekstą ir semantiką tekstuose, todėl jie tapo dominuojančiais metodais šioje srityje. Tačiau tradiciniai metodai vis dar išlieka aktualūs, ypač kai reikia mažesnių skaičiavimo resursų ar didesnio modelių interpretabilumo. Straipsnyje pabrėžiama, kad, nepaisant didelio progreso giluminio mokymosi srityje, teksto klasifikavimas išlieka aktyvia ir nuolat besivystančia tyrimų sritimi.

2. TYRIMO METODIKA

Šiuo tyrimu yra siekiama atlikti automatizuotus tekstinių dokumentų analizes, klasifikuojant juos į grėsmių, apsaugos ar neutralų kontekstą. Kad tyrimas būtų vykdomas nuosekliai ir sklandžiai, yra sudaromas tyrimo planas:

- Pasirenkami tinkami įrankiai;
- Sudaromas duomenų rinkinys;
- Paruošiami duomenys toliau dirbti su klasifikatoriais;
- Pasirinktų algoritmų pasirinkimas ir taikymas;
- Klasifikavimo modelio mokymas;
- Modelio rezultatų vertinimas.

2.1. Įrankiai reikalingi tyrimui atlikti

Tyrimo atlikimui pasirinkta nemokama „Google Colab“ interaktyvioji aplinka, kuri leidžia programuoti ir vykdyti *Python* kalba parašytą kodą. Dažniausia skirta mašininio mokymo modelių kūrimui, bei duomenų mokslo užduotims atlikti. Skaiciavimams yra suteikiama prieiga prie *CPU*, ir *GPU* technologijų. Pagrindiniai privalumai naudojantis šia platforma (souvikmandal, 2024):

- Nemokama prieiga prie *GPU* technologijos, kuri kritiškai reikalinga sudėtingesniems mašininio mokymo modeliams apmokyti.
- Platformos naudojimosi paprastumas, eliminuoja papildomus žingsnius, kurie būna reikalingi kuriant ar konfigūruojant kodo vystymo aplinką.
- Kadangi „Google Colab“ integruota kartu su „Google Drive“, tai leidžia patogiai, greitai ir nesudėtingai saugoti savo sukurtus projektus vienoje aplinkoje.
- Palaiko daugybę pagrindinių *Python* bibliotekų skirtų mašininio mokymo uždaviniams spręsti, duomenų analizei, bei duomenų vizualizacijai.
- Kadangi aplinka susieta su „Google Drive“, tai leidžia dalintis projektais ir su kitais vartotojais.

Programavimui yra pasirinkta *Python* kalba, kuri yra itin plačiai naudojama duomenų moksle. Įvairios bibliotekos (angl. *libraries*) padeda standartizuoti tam tikras kodo dalis, taip pat palengvina programuotojo darbą.

Duomenims išgauti naudojamos pagrindinės bibliotekos:

- *Requests*, ši biblioteka leidžia siųsti HTTP užklausas. Naudinga, kai reikia automatinio modulio, kuris ištrauktų tam tikrus duomenis iš internetinio puslapio (Reitz, 2023).
- *Beautiful Soup*, ši biblioteka leidžia analizuoti iš tinklapio serverio gautą informaciją, t.y. duomenis, lenteles, tekstus, ir kitas *HTML* struktūras (Richardson, 2024).

Duomenų (sakinių) priauginimui ir tvarkymui skirtos bibliotekos:

- *NLTK*, tai biblioteka skirta dirbti su natūraliosios kalbos duomenimis. Leidžia atlikti tokias funkcijas kaip klasifikavimas, tokenizacija, sinonimų radimas ir daug kitų (Bird, Loper, & Klein, 2023).
- *Googletrans*, tai biblioteka, kuri leidžia naudotis vertėjo funkcija. Naudinga, kai reikia generuoti naujus sakinius verčiant originalų sakinį į kitas kalbas, ir galiausiai gražinant į pradinę kalbą (Han, 2020).
- *Transformers*, tai biblioteka sukurta *Hugging Face* bendruomenės, kuri leidžia naudotis sukurtais modeliais, kurie gali atlikti įvairias užduotis. Viena iš jų, reikalinga šiame darbe, naujų sakinių generavimas pasinaudojant GPT-2 modelį (The Hugging Face Team, 2024).

Klasifikatorių kūrimui ir rezultatų atvaizdavimui skirtos bibliotekos:

- *Scikit-learn*, viena pagrindinių bibliotekų suteikianti prieigą prie daugybės mokymo (tiek *unsupervised*, tiek *supervised*) algoritmų. Taip pat papildo duomenų apdorojimui, modelio vertinimui skirtais įrankiais (DataScientest Team, 2023).
- *Seaborn* ir *Matplotlib* bibliotekos abi bibliotekos skirtos vizualizuoti duomenis. Pagrindinis skirtumas, kad *Matplotlib* leidžia daugiau įvairaus pritaikymo galimybių kaip skirtingos spalvos, stulpelių plotis ir pan (Pierre & Whitfield, 2023).
- *PyTorch*, tai biblioteka suteikianti prieigą prie giliojo tinklo kūrimo struktūrų bei jo karkasų, kurie padeda kurti sudėtingus mašininio mokymo modelius (PyTorch Team, 2024).

Papildomos bibliotekos skirtos duomenų analizei, duomenų saugojimui ir tvarkymui:

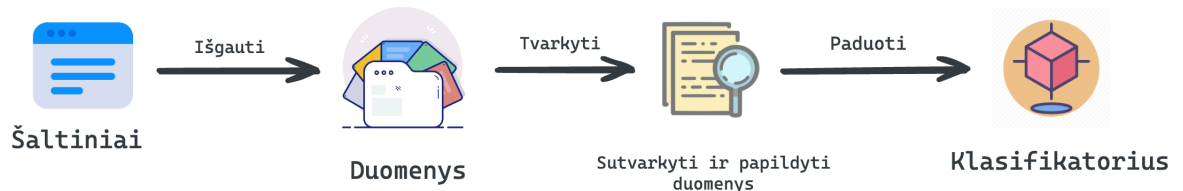
- *Numpy*, viena iš dažnai sutinkamų bibliotekų duomenų moksle, kuri leidžia efektyviai dirbti su dideliais duomenų masyvais. Būtent *NumPy* masyvai yra

pagrindinė duomenų struktūra, kuri dažnai naudojama kitose mašininio mokymo algoritmuose (W3Schools, 2023).

- *Pandas*, dar viena iš populiariesnių bibliotekų, kuri puikiai tinka dirbti su lentelės duomenimis ir kitais struktūruotais duomenimis.

2.2. Tyrimo etapai

Šiame skyriuje yra aprašomi magistrinio darbo tyrimo etapai (žr. 4 pav.). Pradedant duomenų surinkimu iš *CAPEC* ir *MITRE ATTACK* žinių bazių, naudojantis *Python* bibliotekomis kaip *Requests* ir *BeautifulSoup*. Šie duomenys bus nuskaityti iš lentelių ir išsaugojami *JSON* tipo failo formatu. Tuomet duomenys yra tvarkomi, naudojant *NLTK* įrankį (t.y. sumažinamos raidės, pašalinami nereikalingi simboliai) ir apjungiami į vieną didžiulį duomenų failą. Esant poreikiui, taip pat taikomi įvairūs metodai duomenų kiekiui praplėsti bei subalansuoti klases. Po duomenų tvarkymo ir paruošimo yra kūriami klasifikatoriai. Pradedama nuo paprastesnių modelių, kaip linijinė regresija ir naivaus Bajeso klasifikatoriai, pereinant prie kiek sudėtingesnių, kaip SVM, ir baigiant tobulinant BERT, kuris yra vienas iš pažangesnių modelių natūralios kalbos apdorojimo srityje. Tokenizacijai, arba kitaip teksto skaidimui ir paruošimui dirbti su modeliais yra naudojami taip pat skirtingi požymių išskyrimo algoritmai kaip TF-IDF, Word2Vec ar BERT. Modeliui vertinti pagrindinės metrikos yra tikslumas, F1 balas, taip pat stebimas ir laikas modeliui apmokyti. Dar yra naudojama painiavos matrica (angl. *confusion matrix*), kuri padeda vizualiai išnagrinėti, kaip modelis klasifikuoja į atskiras klases. Modelio patikimumo ir stabilumo užtikrinimui skirtinguose duomenų rinkiniuose naudojama kryžminė patikra. Galiausiai, testuojama su savo sukurtais testiniais sakiniais arba iš dokumentų paimtais teksta, kurie padeda įvertinti veiksmingumą realaus pasaulio situacijose.



4 pav. Klasifikatoriaus kūrimo etapai

Duomenų surinkimas

Vieno didžiulio duomenų rinkinio laisvai prieinamo internete, kuris tiktų, ir kurio pakaktų šiam tyrimui nėra. Duomenų surinkimas iš skirtingų šaltinių tampa gana svarbiu etapu šio tyrimo

procesu. Tyrimu duomenys yra surenkami, ištraukiami iš dviejų pagrindinių tinklalapių, žinių bazių *CAPEC* ir *MITRE ATT&CK*, ir kelių su sauga susijusių PDF dokumentų.

CAPEC (angl. *Common Attack Pattern Enumeration and Classification*), tai žinių bazė, kurioje galima rasti sąrašą scenarijų kaip potencialūs kenkėjai gali pasinaudoti pažeidžiamumais programinėje įrangoje, įmonės vidinėje sistemoje. Taip pat yra pateikti pasiūlymai, kaip galima atpažinti bei apsisaugoti nuo tokių scenarijų.

MITRE ATT&CK yra kitas žinynas, kuriame labai panašiai galima rasti daug informacijos apie kenkėjiškas galimas operacijas kibernetinėje erdvėje. Čia suteikiamos išsamios taktikų ir technikų aprašymai, kurie turėtų padėti sumažinti riziką nuo galimų atakų.

Abu šiuose šaltiniuose randami visoms klasėms reikalingi duomenys, t.y. tekstai susiję su grėsmėmis (ką gali kenkėjas padaryti), su sauga (kaip apsisaugoti nuo potencialių atakų) ir tam tikrų terminų paaiškinimai (neutralus kontekstas). Visai tai reikalinga šio darbo tolimesniam tyrimui.

Duomenų gavimas iš *CAPEC* ir *MITRE ATT&CK* žinių bazių yra vykdomas naudojant *Python* programavimo kalbą ir bibliotekas, tokias kaip *Requests* ir *BeautifulSoup*, kurios padeda automatizuoti tinklalapių turinio nuskaitymą. Taktikos, potencialios atakos ir terminų paaiškinimai yra aprašomi atskiruose puslapiuose, lentelėse arba išskiriamos kaip tam tikros paragrafo dalys. Pavyzdžiui, *MITRE ATT&CK* žinių bazėje, naudojant *Python* kodą, galima peržiūrėti taktikų sąrašą keičiant ID esantį subdomene (pvz.: attack.mitre.org/tactics/TA0041/). Dauguma duomenų yra struktūrizuota pagal numeracijos šabloną. Todėl kodas sukurtas taip, kad automatiškai iteruotų per visus galimus taktikų ID naudojant *for* operaciją. Kiekvienas puslapis apdorojamas, o rasti duomenys, t.y. tekstai apie grėsmes, apsaugos metodus ir neutralaus konteksto sakiniai yra įrašomi į *JSON* formatu sukurtą failą, kuris vėliau gali būti naudojamas tolesnei analizei.

JSON faile esančių duomenų pavyzdys (žr. 5 pav.):

```
{
  "Description": "Adversaries may use Valid Accounts to remotely control machines using Virtual Network Computing (VNC).",
  "Label": "attack"
},
{
  "Description": "Adversaries may use Valid Accounts to interact with remote systems using Windows Remote Management (WinRM).",
  "Label": "attack"
},
{
  "Description": "Implementation: Perform output validation for all remote content.",
  "Label": "security"
},
{
  "Description": "Implementation: Disable scripting languages such as JavaScript in browser",
  "Label": "security"
},
{
  "Description": "The Software Development Life Cycle (SDLC) is a process used by the software industry to design, develop, and test high-quality software.",
  "Label": "definition"
},
{
  "Description": "Turing Machine is a theoretical machine that manipulates symbols on a strip of tape according to a table of rules.",
  "Label": "definition"
}
```

5 pav. Surinktų duomenų iškarpa

Duomenys, tai informacija ištraukta iš anksčiau nurodytų šaltinių. Duomenys saugojami žodynų (angl. *dictionaries*) pavidalu. Žodyną sudaro du raktai su jiems priklausančiomis reikšmėmis:

- Sakinys arba aprašymas (angl. *description*), kuriame yra tekstas, kuris bus naudojamas tolimesnei analizei.
- Etiketė (angl. *label*), kuri nurodo kuriai klasei priklauso sakiny. Etiketė yra privaloma dalis norint sukurti ar apmokyti klasifikavimo modelį.

Be konkrečių metodų taikymo, padarytoje duomenų apžvalgoje yra pastebėtos tokios problemos:

- Duomenų rinkinys yra homogeniškas. Grėsmių klasės sakiniai labai panašūs, dažnai pasikartoja tam tikri raktažodžiai, tai gali sukelti modelio persimokymą.
- Palyginus surinktų duomenų skaičius kiekvienos klasės, pastebėtas disbalansas. Vienos klasės duomenų yra kelis kartus daugiau, kas gali iškreipti mokymosi procesą ir turėti įtakos tyrimo rezultatams.
 - Grėsmių kontekstas 2400;
 - Saugos kontekstas 1300;
 - Neutralus kontekstas 600.
- Pastebėti identiški duomenys (sakiniai), kurie turėtų būti pašalinti, siekiant išvengti modelio klaidingo mokymosi ir išlaikyti duomenų autentiškumą.
- Yra daug nereikalingų simbolių, kurie nesuteikia papildomos vertės tekstui, o tik sukelia triukšmą duomenyse. Būtina juos išvalyti prieš mokymo procesą.

Surinktas duomenų rinkinys nėra pilnai tinkamas ir paruoštas tolimesnei analizei. Tam bus taikomi duomenų praplėtimo ir duomenų paruošimo metodai bei algoritmai.

2.3. Duomenų praplėtimas

Analizuojant duomenų rinkinį, buvo nustatyta, kad jame yra trūkumų, tokių kaip tam tikros duomenų klasės homogeniškumas, disbalansas tarp klasių. Siekiant išspręsti šias problemas, buvo taikomi pagrindinių raktažodžių sinonimų keitimo, atgalinio vertimo metodai bei teksto generavimas naudojant *GPT-2* modelį.

Sinonimų keitimo metodas. Taikant iš anksto apibrėžtą sinonimų žodyną, buvo pakeisti raktiniai žodžiai tekstuose, pakeičiant juos sinonimais. Pavyzdžiui, žodis „*adversary*“ buvo keičiamas į tokias alternatyvas kaip „*opponent*“ ar „*enemy*“. Taip pat žodis „*attacker*“ keičiamas į

„aggressor“ ar „intruder“. Šie pakeitimai leidžia išlaikyti esminę teksto prasmę, bet taip kartu sukuria formulavimo variacijas, kurios padeda išvengti modelio persimokymo ir padidina tekstų įvairovę duomenų mokymo rinkinyje (žr. 6 pav.).

```
{ "Description": "Adversaries may gather credentials that can be used during targeting.", "Label": "attack" }
{ "Description": "Adversaries may gather email addresses that can be used during targeting.", "Label": "attack" }
{ "Description": "Adversaries may gather employee names that can be used during targeting.", "Label": "attack" }

Pakeičiam sinonimais

{ "Description": "Rivals might gather credentials that can be used during targeting.", "Label": "attack" }
{ "Description": "Opponents could gather email addresses that can be used during targeting.", "Label": "attack" }
{ "Description": "Enemies might gather employee names that can be used during targeting.", "Label": "attack" }
```

6 pav. Duomenys prieš ir po pritaikyto sinonimų ketimo metodo

Atgalinio vertimo metodas buvo taikomas verčiant tekstą iš anglų į vokiečių kalbą ir atgal, generuojant naujas teksto formas. Automatinis vertimas taip pat įnešą įvairovę ir praturtina duomenų rinkinį. Šis metodas buvo taikomas tiek grėsmių, tiek apsaugos konteksto klasėms (žr. 7 pav.).

```
{ "Description": "Adversaries may alter data en route to storage or other systems in order to manipulate external outcomes or hide activity.", "Label": "attack" }

Angreifer können Daten auf dem Weg zu Speicher- oder anderen Systemen ändern, um externe Ergebnisse zu manipulieren oder Aktivitäten zu verbergen.

{ "Description": "Attackers can modify data en route to storage or other systems to manipulate external results or hide activity.", "Label": "attack" }
```

7 pav. Duomenys prieš ir po taikyto vertimo metodo

Teksto generavimas naudojant GPT-2 modelį. Naudojant šį modelį yra kuriami nauji sakiniai, turintys panašų stilių ir kontekstą kaip originalūs duomenys (žr. 8 pav.). Šis metodas yra itin naudingas, nes jis leidžia automatiškai generuoti tekstus, kurie yra stiliumi ir turinio prasme panašūs į pradinę duomenų bazę, tačiau tuo pačiu įneša ir įvairovę, taip padeda išvengti persimokymo. Generuojant tekstus, GPT-2 modelis analizuoja ir mokosi iš duomenų rinkinio, konteksto bei žodžių vartojimo ypatumų, o vėliau sukuria naujas eilutes, kurios atitinka pradinio teksto semantiką ir stilistiką. Tokiu būdu, naujai sugeneruoti įvairių struktūrų sakiniai padeda modeliui geriau generalizuoti ir mokytis. Šis duomenų praturtinimas yra labai svarbus, siekiant sukurti robustiškesnius mašininio mokymosi modelius.

```
{ "Description": "Security Architecture is a unified security design that addresses the necessities and potential risks involved in a certain scenario or environment.", "Label": "definition" }
{ "Description": "Network Security involves implementing both hardware and software mechanisms to protect the network and infrastructure from unauthorized access, disruptions, and misuse.", "Label": "definition" }
{ "Description": "Cloud Security is a set of control-based safeguards and technology protection designed to protect resources stored online from leakage, theft, or data loss.", "Label": "definition" }
```

8 pav. Naujai sugeneruoti duomenys naudojant GPT-2

Šiame žingsnyje pritaikyti trys duomenų praplėtimo metodai: sinonimų keitimas, atgalinis vertimas ir teksto generavimas su *GPT-2* pagalba. Visi šie pritaikyti metodai suteikė pakankamai kokybišką duomenų įvairovę, leidžiančią pasiruošti efektyviai treniruoti mašininio mokymo modelius. Toliau bus taikomi duomenų paruošimo ir valymo procesai, kurie yra būtini siekiant užtikrinti aukščiausią modelio efektyvumą.

2.4. Duomenų paruošimas

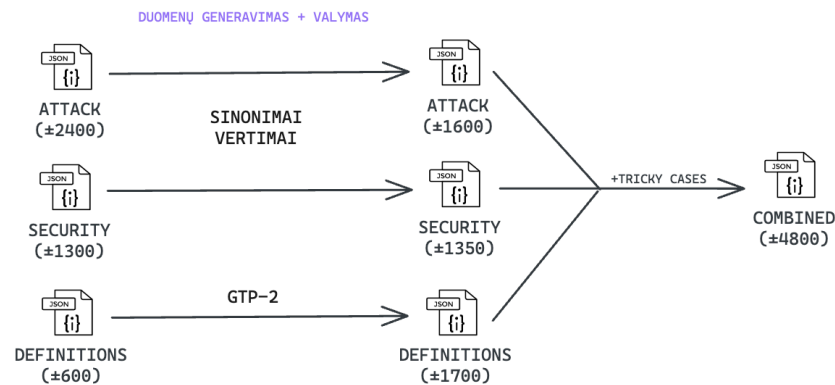
Po duomenų surinkimo yra kitas, ne ką mažiau svarbus žingsnis – duomenų paruošimas. Tai užtikrina aukštesnę duomenų kokybę ir efektyvesnę mašininio mokymosi modelių treniravimą. Šiame darbe buvo taikomi keli duomenų valymo metodai:

- Specialių simbolių šalinimas. Iš duomenų buvo pašalinti visi nereikalingi simboliai, kurie galėjo trukdyti duomenų analizei ir modelio mokymui. Šis procesas apima simbolių ir specialių ženklų, kurie nesukuria jokios papildomos vertės tekstui, o tik padidina triukšmą duomenyse.
- Mažųjų raidžių naudojimas. Visas tekstas buvo konvertuotas į mažąsias raides. Tai standartizuoja duomenis ir pašalina bet kokias nevienodumų, susijusių su raidžių dydžiu, problemas.
- Tarpų koregavimas. Tarpai tekste buvo sutvarkyti, pašalinant perteklinius tarpus ir užtikrinant, kad tarp žodžių būtų vienodas atstumas. Tai svarbu, nes papildomi tarpai gali sukelti klaidų tekstų apdorojime ir analizėje.
- Dublikatų šalinimas. Atliekamas duomenų patikrinimas dėl dublikatų, identifikuojant ir šalinant identiškų duomenų įrašus. Padeda užtikrinti duomenų unikalumą ir sumažina klaidingą modelio mokymą.

Po valymo gautas duomenų kiekis taip pat pasikeitė. Gauti galutiniai rezultatai po duomenų apvalymo ir dublikatų pašalinimo:

- Grėsmės kontekstas, apie 1600 sakinių;
- Saugos kontekstas, apie 1350 sakinių;
- Neutralus kontekstas, apie 1700 sakinių.

Dar papildžius keletais sudėtingesniais sakiniais, galutiniame variante, apjungus visas klases, pavyko gauti apie 4800 sužymėtų sakinių (žr. 9 pav.).



9 pav. Duomenų paruošimo schema

Duomenų ypatumai po valymo:

- Duomenų klasės labiau subalansuotas ir praplėstos;
- Duomenys švaresni, mažiau nereikalingų simbolių, nereikalingų tarpų;
- Duomenis papildžius sudėtingesniais scenarijais, atsirado didesnė duomenų įvairovė.

Šie valymo veiksmai ne tik padeda sumažinti triukšmą duomenyse, bet ir paruošia tekstą tolesniam efektyviam apdorojimui, užtikrinant, kad mašininio mokymosi modeliai galėtų veiksmingiau mokytis iš švarių, gerai struktūrizuotų duomenų.

2.5. Metodų pasirinkimas

Skyrelyje aptariami teksto klasifikavimui skirti metodai. Atsižvelgiant į duomenų ypatumus ir analizės tikslus, yra pasirinkti keturi pagrindiniai metodai: daugianarės logistinės regresijos metodas, naivus Bajeso metodas, atraminių vektorių mašinos metodas bei *BERT* mokymosi modelis.

Šiame darbe tikslas yra klasifikuoti su sauga susijusius tekstus į tris grupes. Modeliams kurti ir apmokyti duomenys yra paimti iš žinių bazių *CAPEC* ir *MITRE ATT&CK*. Šių bazių tekstai yra daugiau bendriniai ir generalizuoti, neturintys tikrinių daiktavardžių. Pateikti scenarijai veikia kaip karkasai, kurie naudojami įmonėse aprašant standartus ar kitus su sauga susijusius dokumentus. Kadangi įmonė gali turėti įvairias tu karkasų variacijas, remtis vien žodžių reikšmėmis, gali būti rizikinga. Todėl svarbu koncentruotis į metodus, kurie remiasi ne tik žodžių semantika, bet ir sakinių struktūra. Šiame darbe bus analizuojami įvairūs modeliai – nuo paprastesnių iki sudėtingesnių, siekiant nustatyti, kokia įtaka turi modelio pasirinkimas bei skirtingi požymių išskyrimo metodai.

Pasirinkti teksto klasifikavimo metodai:

- Daugianarė logistinė regresija (arba tiesiog daugianarė regresija), tinkama trijų klasių teksto klasifikavimui. Ji kiek sudėtingesnė nei įprasta binarinė logistinė regresija, bet vis vien yra greitai ir efektyviai apmokama. Taip pat modelis leidžia pamatyti kokią įtaką kiekvienas žodis (arba kintamasis) daro tikimybę priskiriant sakinį tam tikrai kategorijai
- Naivusis Bajeso klasifikatorius, taip pat gana greitas ir nesudėtingas algoritmas. Klasifikatorius gerai veikia todėl, nes su teksto duomenimis laikosi prielaidos, kad savybės (žodžiai) yra nepriklausomi vieni nuo kitų.
- Atraminių vektorių mašinos klasifikatorius, vidutinio sudėtingumo metodas. Dėl branduolio funkcijų (angl. *kernel*) panaudojimo, gali veiksmingai tvarkyti netiesines duomenų erdves, kas yra naudinga sudėtingesniuose uždaviniuose spręsti. Nors tai yra gana efektyvus metodas, tačiau jau reikalauja daugiau žinių apie duomenų savybes, reikalauja daugiau eksperimentų su skirtingais parametrais.
- *BERT*, (angl. *Bidirectional Encoder Representations from Transformers*) yra vienas pažangesnių atviro kodo mašininio mokymosi modelis, naudojamas natūralios kalbos apdorojimui, supratimui ir generavimui. Gali atlikti įvairias užduotis, įskaitant teksto klasifikavimą.

Prieš perduodant duomenis modeliams, tekstiniai duomenys turi būti paversti į skaitines vertes. Tam yra taikomi požymių išskyrimo metodai. Šiame darbe bus tikrinama, kokią įtaką daro skirtingų požymių išskyrimo metodų pasirinkimas. Šiame darbe nagrinėjami metodai:

- *TF-IDF* (angl. *Term Frequency-Inverse Document Frequency*), vienas iš paprastesnių metodų, kuris apskaičiuoja žodžio svarbą dokumente atsižvelgdamas į tai, kiek kartų žodis pasikartoja dokumente, ir kiek kituose dokumentų rinkiniuose yra tas pats žodis.
- *Word2Vec* yra kiek sudėtingesnis metodas, kuris sukuria žodžių vektorines vertes, kurios atspindi žodžių tarpusavio ryšius. Žodžiai, kurie bus panašūs pagal reikšmę, yra atvaizduojami panašiais vektoriais.
- *Glove* (angl. *Global Vectors for Word Representation*), taip pat sudėtingas metodas, kuris naudoja didesnius duomenų rinkinius. Dažnai naudojama aukštos kokybės teksto generavimo ir mašininio vertimo užduotyse.

Galiausiai modelių įvertinimui pasirinkti šie būdai:

- Tikslumas (angl. *accuracy*), tai yra pagrindinė klasifikavimo metrika, kuri parodo teisingai klasifikuotų objektų (teigiamų ir neigiamų) procentą.
- F1 balas (angl. *F1 score*) yra tikslumo ir atšaukimo (angl. *recall*) vidurkis, skirtas įvertinti klasifikavimo modelio našumą. Naudingas, kai klasių pasiskirstymas yra nevienodas.
- Laikas, taip pat svarbi vertinimo metrika, kuri matuoja, kiek laiko reikia modeliui apmokyti ir atlikti prognozes.
- Validacijos nuostolio vertė (angl. *Validation Loss*) yra metrika, naudojama modelio apmokymo metu siekiant įvertinti modelio gebėjimą generalizuoti. Ji matuoja skirtumą tarp tikrųjų klasių ir modelio prognozių validacijos duomenų rinkinyje.

Vizualiam atvaizdavimui bus naudojami du įrankiai:

- Painiavos matrica (angl. *confusion matrix*), kuri išsamiai pateikia informacija apie kiekvienos klasės klaidas.
- Taip pat bus pateikiami įvairūs grafai (stulpelinės diagramos), kurie padės palyginti rezultatus.

Dirbant su tradiciniais klasifikavimo metodais, duomenys bus skaidomi taikant kryžminį patikrinimą (angl. *X-Fold Cross-Validation*). Šis metodas padalija duomenų rinkinį į k dalis. Kiekvieną kartą viena dalis naudojama testavimui, o likusios $k-1$ dalys skirtos mokymui. Šis procesas kartojamas k kartų, o galutinis rezultatas gaunamas kaip vidurkis iš visų k rezultatų.

Kitame poskyryje yra aptariami, kaip pasirinkti metodai yra taikomi praktikoje, įskaitant įvairius duomenų paruošimo būdus ir galiausiai jų įtaką klasifikavimo rezultatams.

Šiame poskyryje dėmesys skiriamas pasirinktų metodų taikymui praktikoje aprašymui. Detaliai aptariamos įvairios kiekvieno modelio sukūrimui taikomas metodų variacijos (žr. 10 pav.). Nors duomenys jau dalinai yra paruošti, tačiau tyrimui atlikti yra sudaromas dar vienas atskiras rinkinys. Kaip matyti scheme (pav...), duomenys yra išskaidomi į dvi rūšis: pagrindinis duomenų rinkinys *COMBINED* ir duomenų rinkinys, kuriam yra pritaikyta žodžių formų normalizavimas ir nereikalingų žodžių pašalinimo funkcija. Tuomet kiekvienam duomenų rinkiniui pritaikomi skirtingi požymių išskyrimo metodai. Ir galiausiai testuojama su skirtingais klasifikavimo metodais. Vienintelis BERT modelis turi savo atskirą būdą, nes naudoja savo integruotus požymių išskyrimo algoritmus, kurie leidžia automatiškai generuoti reikalingas teksto savybes. Vienas iš didesnių BERT parametrų, kuris keičiamas, yra optimizatorius. Išsitestavus visas 20 variacijų, galima pamatyti, kokią įtaką turės pasirinkti skirtingai paruošti duomenų rinkiniai, požymių išskyrimo bei skirtingi klasifikavimo metodai.

10 pav. Skirtingu metodu taikymų variacijos su skirtingais duomenų rinkiniais

Schemoje (žr. 10 pav. Skirtingų metodų taikymų variacijos su skirtingais duomenų rinkiniais), geltonai pažymėta spalva ikonėlė su daugiau nei 4800 įrašų, yra pagrindinis duomenų failas. Šalia jo, linija sujungta mėlynos spalvos ikonėlė yra žymimas tas pats duomenų failas, tik jau su pritaikytomis papildomomis funkcijomis, kaip nereikalingų žodžių (angl. *stopwords removal*) pašalinimas bei lematizacija (angl. *lemmatization*), kas yra panašių (vienodą bazinę šaknį turinčių) žodžių grupavimas. Iš skirtingų failų, einančios linijos ties žaliai pažymėtomis ikonėlėmis, nurodo skirtingus požymių išskyrimo metodus, kurių pritaikymas yra reikalingas klasikiniams klasifikatoriams, kad duomenys būtų pateikiami skaitiniu pavidalu. Tuo tarpu kalbant apie pažangų klasifikatorių BERT, raudonos spalvos ikonėlėmis yra žymimi būtent skirtingi optimizatoriai. Ir schemos galuose galime matyti skirtingų klasifikatorių pavadinimus. Viską suskaičiavus, gauname 21 skirtingą variaciją, kuri bus tyriama.

2.6.1. Modelių kūrimo variacijos

Klasifikavimo modelių kūrimo variacijos taikant įvairius duomenų paruošimo, požymių išskyrimo ir klasifikavimo metodus:

- Duomenų rinkinys pritaikius lematizaciją ir nereikalingų žodžių pašalinimą:
 - Požymių išskyrimo metodas: TF-IDF;
 - Klasifikavimo metodas: SVM;
 - Klasifikavimo metodas: daugianarė (arba daugianarė) logistinė regresija;
 - Klasifikavimo metodas: Naivo Bajeso;
 - Požymių išskyrimo metodas: Word2Vec;
 - Klasifikavimo metodas: SVM;
 - Klasifikavimo metodas: daugianarė logistinė regresija;
 - Klasifikavimo metodas: Naivo Bajeso;
 - Požymių išskyrimo metodas: GloVe;
 - Klasifikavimo metodas: SVM linear;
 - Klasifikavimo metodas: daugianarė logistinė regresija;
 - Klasifikavimo metodas: Naivo Bajeso;
- Duomenų rinkinys COMBINED:
 - Požymių išskyrimo metodas: TF-IDF;
 - Klasifikavimo metodas: SVM;

- Klasifikavimo metodas: daugianarė logistinė regresija;
- Klasifikavimo metodas: Naivo Bajeso;
- Požymių išskyrimo metodas: Word2Vec;
 - Klasifikavimo metodas: SVM;
 - Klasifikavimo metodas: daugianarė logistinė regresija;
 - Klasifikavimo metodas: Naivo Bajeso;
- Požymių išskyrimo metodas: GloVe;
 - Klasifikavimo metodas: SVM;
 - Klasifikavimo metodas: daugianarė logistinė regresija;
 - Klasifikavimo metodas: Naivo Bajeso;

Verta paminėti, jog SVM modeliui bus naudojamas tiesinio branduolio parametras.

Pažangesnio klasifikavimo modelio BERT kūrimo variacijos taikant tris skirtingus optimizatorius:

- Tiek Adam, tiek AdamW, tiek RMSprop optimizatoriams yra parinkti standartiniai modelio parametrai:
 - Epochų skaičius lygus 4;
 - Partijos dydis (ang. batch size) lygus 16;
 - Pasirinktas BERT modelio tipas yra *bert-base-uncased*;
 - Tokenizacijos metodas tai pat *bert-base-uncased*;
 - Mokymosi greitis nustatytas $2e-5$.

2.6.2. Vertinimo variacijos

Klasikiniams SVM, naivaus Bajeso ir daugianariai logistinei regresijai yra taikomi šie vertinimo būdai:

- Tikslumas (angl. *accuracy*), tai yra pagrindinė klasifikavimo metrika, kuri parodo teisingai klasifikuotų objektų (teigiamų ir neigiamų) procentą (Dharani, 2020).
- F1 balas (angl. *F1 score*) yra tikslumo ir atšaukimo (angl. *recall*) vidurkis, skirtas įvertinti klasifikavimo modelio našumą. Naudingas, kai klasių pasiskirstymas yra nevienodas (Dharani, 2020).
- Painiavos matrica (angl. *confusion matrix*) yra lentelė, kuri padeda įvertinti modelio veikimą (žr. 11 pav.). Sudaroma iš dviejų dimensijų: faktinių klasių ir prognozuojamų klasių, kur kiekvienas elementas atspindi, kiek pavyzdžių buvo

prognozuota į tam tikrą klasę ir faktiškai priklausė tam tikrai klasei (Dharani, 2020).

		Predicted		
		0	1	
Actual	0	TN	FP Type I error	Specificity = $TN/(TN+FP)$
	1	FN Type II error	TP	Recall or Sensitivity = $TP/(TP+FN)$
		Negative Rate = $TN/(FN+TN)$	Precision = $TP/(TP+FP)$	
$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN}$				
$\text{F1 - Score} = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}}$				

11 pav. Painiavos matrica ir metrikų skaičiavimo formulės

BERT modelių rezultatų vertinimui, be anksčiau aprašytų tikslumo ir F1 balo metrikų, modeliui įvertinti dar yra naudojama validacijos nuostolio vertė.

3. TYRIMO REZULTATAI

Šiame skyriuje yra aprašomi taikyti, tiek klasikiniai metodai kaip atraminių vektorių mašina, daugianarė logistinė regresija ar naivaus Bajeso metodas, tiek pažangesni klasifikavimo būdai kaip naudojantis *BERT* modeliu. Rezultatai yra lyginami, daromos išvados kokią įtaką daro įvairūs požymių išskyrimo metodai, pasirinktas klasifikatorius ar skirtingai paruoštas duomenų rinkinys.

3.1. Tyrimas naudojant pagrindinį duomenų rinkinį *COMBINED*

Šiame poskyryje analizuojamas pagrindinis rinkinys, kuriam yra pritaikyti tik minimalūs teksto tvarkymo metodai: raidžių sumažinimas, nereikalingų tarpų pašalinimas, nereikalingų simbolių pašalinimas.

3.1.1. Rezultatai taikant naivaus Bajeso metodą

Kuriant naivaus Bajeso klasifikatorių yra taikomi trys skirtingi požymių išskyrimo metodai: TF-IDF, Word2Vec ir GloVe. Kiekvieno jų yra gaunamas tikslumas, F1 balas ir apytikslis apmokymo laikas (žr. 3 lentelė. Naivaus Bajeso su duomenų rinkiniu COMBINED rezultatai.

3 lentelė. Naivaus Bajeso su duomenų rinkiniu COMBINED rezultatai

Naivus Bajesas	Metrikos	TF-IDF	Word2Vec (savo)	GloVe
	Tikslumas (angl. Accuracy)	0.9	0.66	0.67
	F1 balas	0.9	0.66	0.66
	Apmokymo laikas (sekundėmis)	0.06	0.09	0.14

Iš lentelėje pateiktų duomenų galima matyti, jog didžiausias tikslumas yra pasiektas naudojant būtent TF-IDF požymių išskyrimo metodą, kurio tikslumas siekia net 0.9, o apmokymo laikas irgi minimalus.

3.1.2. Rezultatai taikant daugianarės logistinės regresijos metodą

Kuriant daugianarės logistinės regresijos klasifikatorių taip pat yra taikomi trys skirtingi požymių išskyrimo metodai: TF-IDF, Word2Vec ir GloVe. Kiekvieno jų yra gaunamas tikslumas,

F1 balas ir apytikslis apmokymo laikas (žr. 4 lentelė. Daugianarės logistinės regresijos su duomenų rinkiniu COMBINED rezultatai).

4 lentelė. Daugianarės logistinės regresijos su duomenų rinkiniu COMBINED rezultatai

Daugianarė logistinė regresija	Vertinimai	TF-IDF	Word2Vec (savo)	GloVe
	Tikslumas (angl. Accuracy)	0.92	0.76	0.84
	F1 balas	0.91	0.76	0.84
	Apmokymo laikas (sekundėmis)	3.2	3.33	3.6

Iš lentelėje pateiktų duomenų galima matyti, jog didžiausias tikslumas yra pasiektas vėlgi naudojant *TF-IDF* požymių išskyrimo metodą, kurio tikslumo vertė siekia 0.92 ir apmokymo laikas yra pats mažiausias palyginus su kitais požymių išskyrimo metodais.

3.1.3. Rezultatai taikant atraminių vektorių mašinos metodą

Kuriant atraminių vektorių mašinos klasifikatorių vėlgi yra taikomi trys skirtingi požymių išskyrimo metodai: TF-IDF, Word2Vec ir GloVe. Kiekvieno jų yra gaunamas tikslumas, F1 balas ir apytikslis apmokymo laikas (žr. 5 lentelė. SVM su duomenų rinkiniu COMBINED rezultatai).

5 lentelė. SVM su duomenų rinkiniu COMBINED rezultatai

SVM	Vertinimai	TF-IDF	Word2Vec (savo)	GloVe
	Tikslumas (angl. Accuracy)	0.92	0.76	0.85
	F1 balas	0.92	0.76	0.84
	Apmokymo laikas (sekundėmis)	44.53	20.98	15.17

Iš lentelėje pateiktų duomenų galima matyti, jog didžiausias tikslumas yra pasiektas naudojant *TF-IDF* požymių išskyrimo metodą, kurio tikslumo vertė siekia 0.92, tačiau apmokymo laikas yra didžiausias. Su mažiausiu apmokymo laiku, bei taip pat gana aukštą tikslumo vertę pasiekė *GloVe* metodas.

3.2. Tyrimas naudojant rinkinį su papildomu paruošimu

Šiame poskyryje analizuojamas pagrindinis rinkinys, kuriam yra pritaikyti ir papildomi teksto tvarkymo metodai: žodžių sutrumpinimas iki pagrindinės formos, ir nereikalingų arba mažą vertę turinčių žodžių pašalinimas.

3.2.1. Rezultatai taikant naivaus Bajeso metodą

Kuriant naivaus Bajeso klasifikatorių yra taikomi trys skirtingi požymių išskyrimo metodai: TF-IDF, Word2Vec ir GloVe. Kiekvieno jų yra gaunamas tikslumas, F1 balas ir apytikslis apmokymo laikas (6 lentelė. Naivaus Bajeso su paruoštu duomenų rinkiniu rezultatai).

6 lentelė. Naivaus Bajeso su paruoštu duomenų rinkiniu rezultatai

Naivus Bajesas	Metrikos	TF-IDF	Word2Vec (savo)	GloVe
	Tikslumas (angl. Accuracy)	0.88	0.65	0.72
	F1 balas	0.88	0.65	0.72
	Apmokymo laikas (sekundėmis)	0.09	0.11	0.3

Iš lentelėje pateiktų duomenų galima matyti, jog didžiausias tikslumas yra pasiektas naudojant *TF-IDF* požymių išskyrimo metodą, kurio tikslumas siekia 0.88, apmokymo laikas irgi minimalus. Taikant Word2Vec požymių išskyrimo metodą, tikslumas buvo mažiausias, ir laikas modelio sukūrimui užtruko ilgiausiai.

3.2.2. Rezultatai taikant daugianarės logistinės regresijos metodą

Kuriant daugianarės logistinės regresijos klasifikatorių toliau taikomi trys skirtingi požymių išskyrimo metodai: TF-IDF, Word2Vec ir GloVe. Kiekvieno jų yra gaunamas tikslumas, F1 balas ir apytikslis apmokymo laikas (žr. 7 lentelė. Daugianarės logistinės regresijos su paruoštu duomenų rinkiniu rezultatai).

7 lentelė. Daugianarės logistinės regresijos su paruoštu duomenų rinkiniu rezultatai

Daugianarė logistinė regresija	Vertinimai	TF-IDF	Word2Vec (savo)	GloVe
	Tikslumas (angl. Accuracy)	0.89	0.74	0.81
	F1 balas	0.89	0.74	0.81
	Apmokymo laikas (sekundėmis)	7.65	4.62	10.92

Nors daugiausiai laiko truko sukurti modelį su GloVe metodu, tačiau didžiausias tikslumas buvo pasiektas taikant būtent *TF-IDF* metodą 0.89.

3.2.3. Rezultatai taikant atraminių vektorių mašinos metodą

Vėl kuriant atraminių vektorių mašinos klasifikatorių taikomi tie patys trys skirtingi požymių išskyrimo metodai: TF-IDF, Word2Vec ir GloVe. Kiekvieno jų yra gaunamas tikslumas,

F1 balas ir apytikslis apmokymo laikas (žr. 8 lentelė. SVM su paruoštu duomenų rinkiniu rezultatai).

8 lentelė. SVM su paruoštu duomenų rinkiniu rezultatai

SVM	Vertinimai	TF-IDF	Word2Vec (savo rinkinį)	GloVe
	Tikslumas (angl. Accuracy)	0.9	0.72	0.71
	F1 balas	0.9	0.72	0.7
	Apmokymo laikas (sekundėmis)	80	27.32	27.54

Iš lentelėje pateiktų duomenų galima matyti, jog didžiausias tikslumas yra pasiektas naudojant *TF-IDF* požymių išskyrimo metodą, kurio tikslumo vertė siekia 0.9 ir apmokymo laikas yra kelis kartus didesnis, kai yra naudojami kiti požymių išskyrimo metodai. Tiek naudojant Word2Vec, tiek GloVe rezultatai tiek tikslumo 0.7, tiek laiko atžvilgiu 27 yra gana panašūs.

3.3. Tyrimas taikant BERT metodą, bet su skirtingais optimizatoriais

Šiame poskyryje BERT modeliui yra paduodamas pagrindinis duomenų rinkinys *COMBINED*, kuriam yra pritaikyti tik minimalūs teksto tvarkymo metodai: raidžių sumažinimas, nereikalingų tarpų pašalinimas, nereikalingų simbolių pašalinimas.

3.3.1. Rezultatai taikant AdamW optimizatorių

Tobulinant BERT modelį ir jam pritaikant AdamW optimizatorių bei įprastus modelio parametrus, gauti tokie rezultatai:

- Apmokymo laikas tarp 1751 sekundžių;
- Tiek tikslumas, tiek F1 balas siekė 1, o nuostolio vertė (angl. validation loss) lygi 0.01.

3.3.2. Rezultatai taikant Adam optimizatorių

Tobulinant BERT modelį ir jam pritaikant Adam optimizatorių bei įprastus modelio parametrus, gauti tokie rezultatai:

- Apmokymo laikas tarp 1760 sekundžių;

- Tiek tikslumas, tiek F1 balas taip pat siekė 1, o nuostolio vertė (angl. validation loss) lygi 0.01.

3.3.3. Rezultatai taikant RMSprop optimizatorių

Tobulinant BERT modelį ir jam pritaikant RMSprop optimizatorių bei įprastus modelio parametrus, gauti tokie rezultatai:

- Apmokymo laikas tarp 1741 sekundžių;
- Tikslumas siekė vos 0.68, F1 balas nukrito iki 0.58, o nuostolio vertė siekia net 0.64. Prasčiausias rezultatas iš visų optimizatorių.

Nors rezultatai yra puikūs tiek su Adam, tie su AdamW optimizatoriais, tačiau tai kelia įtarimą, jog modelis yra permokytas, kas reiškia, jog pernelyg gerai prisitaikė prie treniravimo duomenų, ir gali prastai veikti su naujais duomenimis.

3.4. Gautų rezultatų įžvalgos taikant klasikinius metodus

Šiame poskyryje yra pateikiami pastebėjimai atlikus visas pasirinktų modelių ir požymių išskyrimo metodų variacijas. Išanalizavus duomenis pateikiamos įžvalgos kokią įtaką rezultatams daro klasifikavimo, požymių išskyrimo metodų pasirinkimas ir skirtingas duomenų paruošimas.

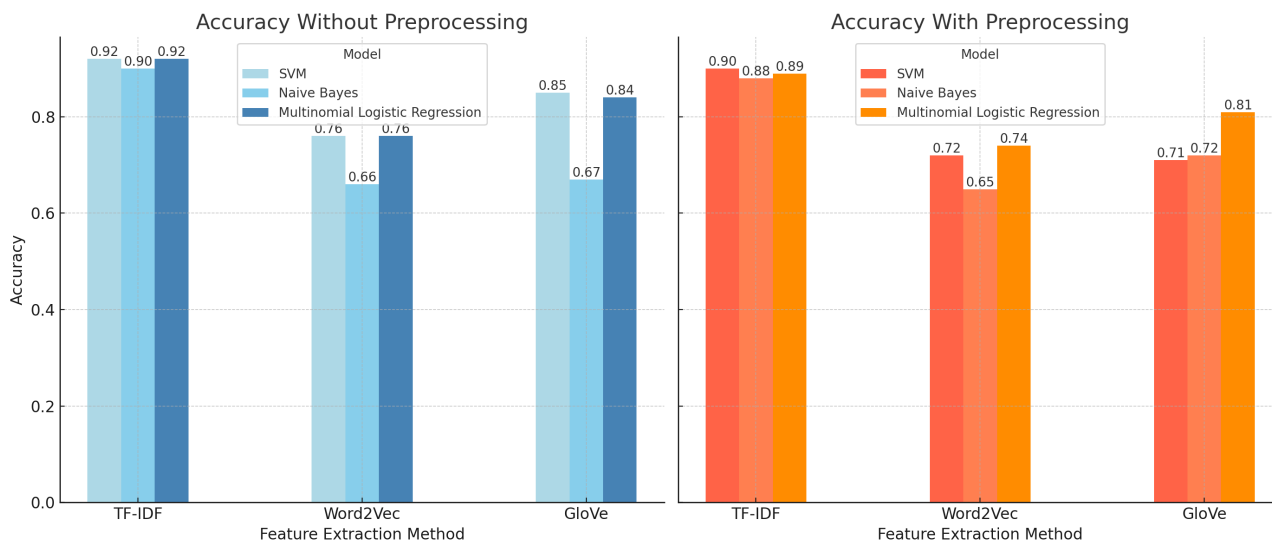
3.4.1. Duomenų paruošimas

Duomenų paruošimas, įskaitant lematizavimą ir nereikšmingų žodžių pašalinimą, iš gautų rezultatų turi įtaką galutiniam rezultatui. SVM atveju, papildomai apdoroti duomenys mažina tikslumo rodiklį su visais taikomais požymių išskyrimo metodais, kai tuo tarpu naivusis Bajesas, naudojant būtent GloVe požymių išskyrimo metodą, siek tiek pagerina tikslumą. Papildomas apdorojimas padidina apmokymo laikus. Pavyzdžiui, SVM klasifikatoriaus su TF-IDF požymių išskyrimo metodu, apmokymo laikas padidėja nuo 44.53 sekundžių iki 80 sekundžių pritaikius papildomą apdorojimą, daugianarės logistinės regresijos su Word2Vec metodu nuo 3.33 sekundės iki beveik 5 sekundžių.

Atsižvelgiant į gautus rezultatus, galima teigti, jog duomenų paruošimas daro minimalią įtaką modelio tikslumui (tiek sumažina, tiek padidina). Duomenų apdorojimas labiau atsiliepia apmokymo greičiui, kuris didėja. Tai gali būti aktualu kuriant kompleksiškesnius modelius su didesniais duomenų kiekiais.

3.4.2. Klasifikatoriaus pasirinkimas

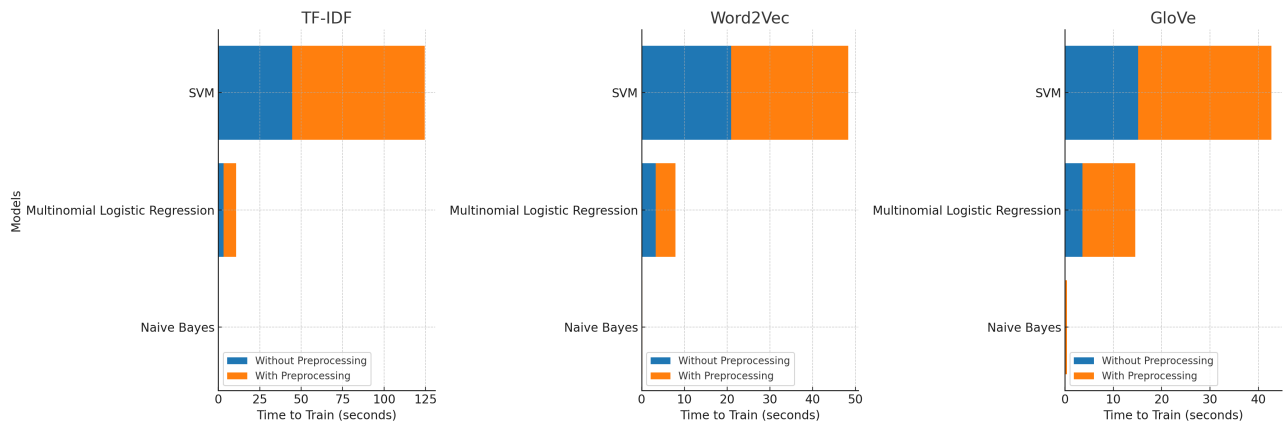
Skirtingų modelių pasirinkimas turi įtaką rezultatams. Stebint žemiau pavaizduotą grafiką matome (žr. 12 pav.), nors visi modeliai gerai veikia su TF-IDF, tačiau turi skirtingus tikslumo įverčius su Word2Vec ir GloVe požymių išskyrimo metodais. Jeigu taikant GloVe metodą, su SVM ir daugianarinę logistinę regresiją tikslumas viršija 0.8, tai tuo tarpu naivusis Bajesas su tuo pačiu požymių išskyrimo metodu nesiekia net 0.7 vertės. Panaši situacija ir su kitais metodais, tik skirtumas tarp tikslumų yra mažesnis. Taip pat ryškūs skirtumai pastebimi, kai dirbama su papildomai apdorotais duomenimis. Taikant GloVe požymių išskyrimo metodą daugianarės logistinės regresijos modelyje gauname 0.81 tikslumą, o su SVM modeliu jau tikslumas siekia 0.71.



12 pav. Skirtingų metodų taikymo tikslumo rezultatai

Iš vizualiai pateiktų duomenų grafuose, galime teigti jog SVM klasifikatoriai sukuria geriausia tikslumą palyginus su kitais metodais. Net iš 6-šių skirtingų variacijų, 4-iose variacijose SVM gavo aukščiausią tikslumą, ir tik su papildomais apdorotais duomenimis ir GloVe požymių išskyrimo metodu gavo mažiausią 0.71 tikslumą.

Modelio pasirinkimas taip pat veikia ir apmokymo laikus. Grafai puikiai iliustruoja, jog šablonas su visais modeliais išlieka toks pat (žr. 13 pav.). Daugiausiai laiko užtrunka apmokyti SVM modelį, kelis kartus mažiau laiko reikia daugianarės logistinės regresijos modeliui, bei minimaliai laiko reikia Bajeso modeliui apmokyti.

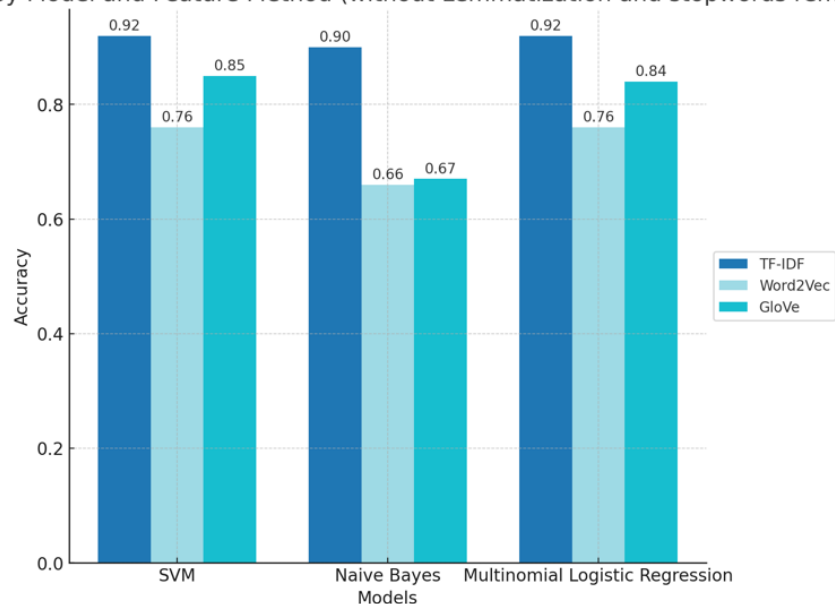


13 pav. Laiko verčių palyginimai

3.4.3. Požymių išskyrimo metodo pasirinkimas

Skirtingi požymių išskyrimo metodai taip pat lemia modelio tikslumą. Stebint grafą (žr. 14 pav. **Error! Reference source not found.**), kur duomenys nėra papildomai apdoroti, galime matyti, jog taikant TF-IDF požymių išskyrimo metodą, yra gaunamas aukštesnis tikslumas su visais sukurtais modeliais. Kai tuo tarpu Word2Vec rodo prasčiausią tikslumą visais atvejais.

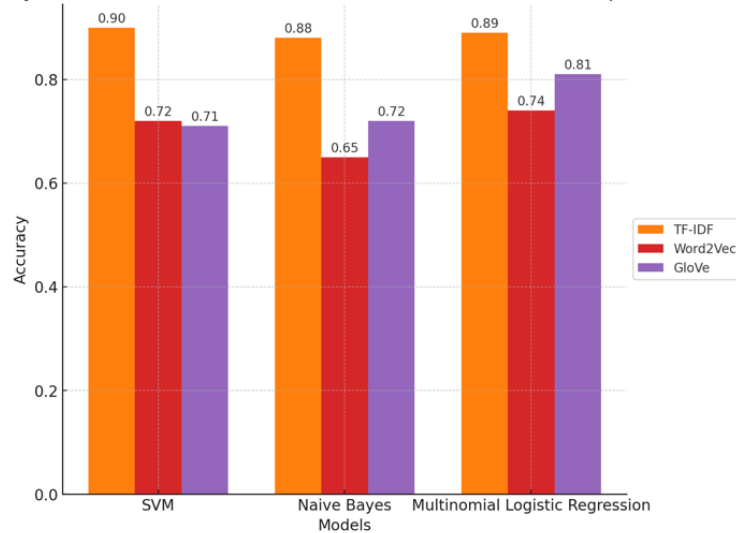
Accuracy by Model and Feature Method (without Lemmatization and stopwords removal)



14 pav. Tikslumas gautas su skirtingais požymių išskyrimo metodais naudojant COMBINED

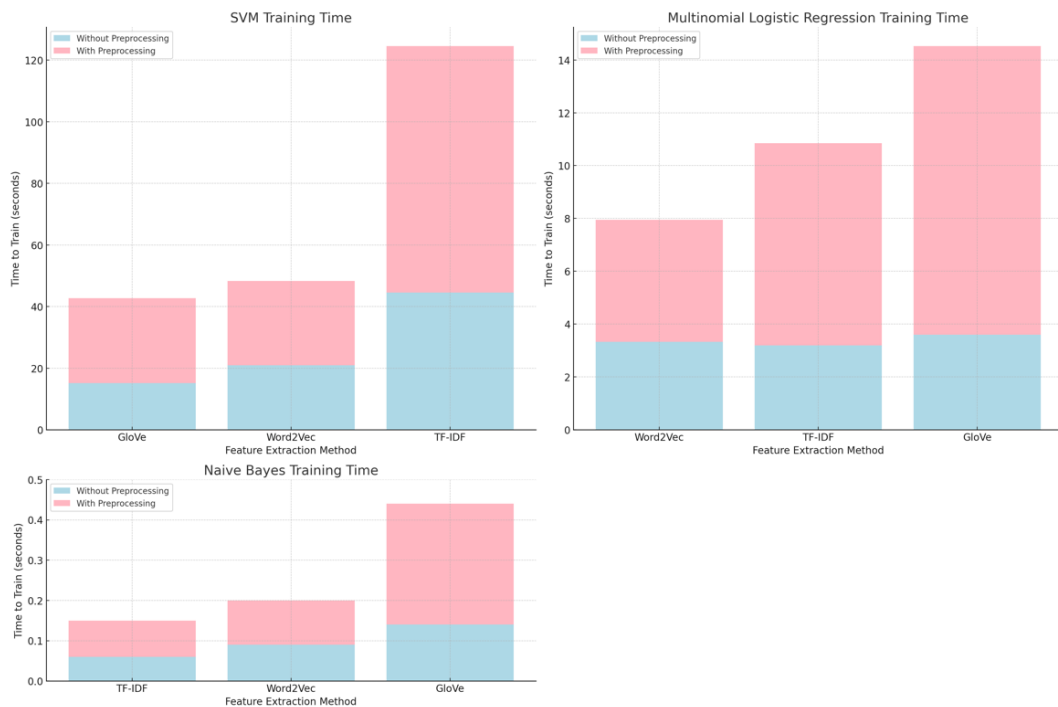
Su papildomai apdorotais duomenimis, situacija išlieka panaši. Taikant TF-IDF metodą, tikslumas gaunamas didžiausias visais atvejais (žr. 15 pav. **Error! Reference source not found.**). Prasčiausios variacijos, kai Word2Vec metodą taikome naivaus Bajeso modeliui.

Accuracy by Model and Feature Method (with Lemmatization and stopwords removal)



15 pav. Tikslumas gautas su skirtingais požymių išskyrimo metodais

Požymių išskyrimo metodas veikia apmokymo laikus taip pat, nors aiškos sekos irgi nėra matyti. Kiekviename modelyje taikant skirtingus požymių išskyrimo metodus, laikas skirtas apmokyti modelį skiriasi, ir su kiekvienu modelių metodai išsidėsto kitaip (žr. 16 pav.). Apmokant SVM modelį daugiausiai laiko užtrunka, kai pasirenkame TF-IDF vektorizatorių, mažiausiai su GloVe. Apmokant tiek daugianarės logistinės regresijos, tiek naivaus Bajeso modelį, daugiausiai laiko užima su GloVe vektorizatoriumi, bet skirtumai nėra žymūs.



16 pav. Laiko vertės taikant skirtingus požymių išskyrimo metodus

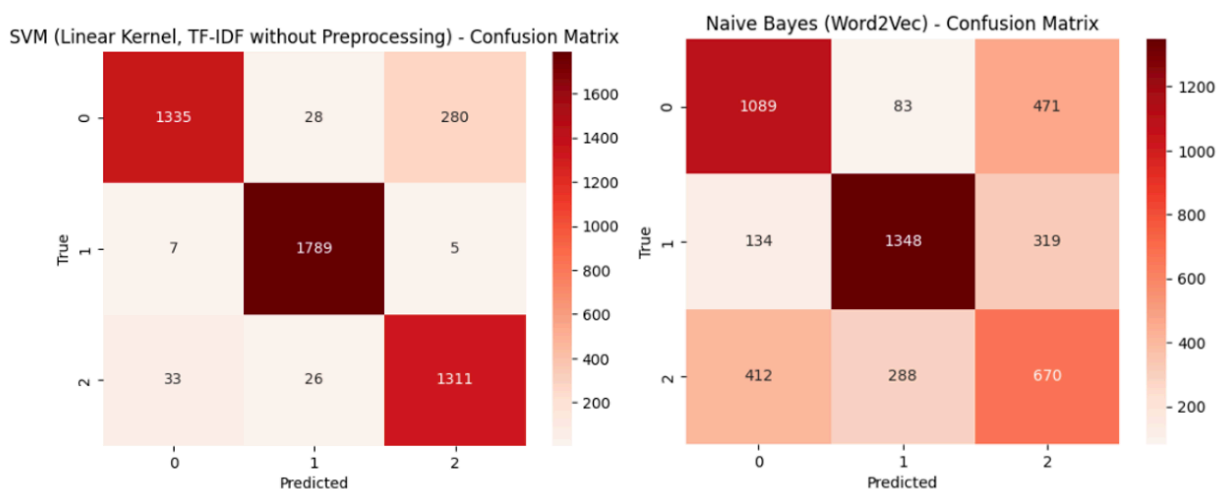
3.4.4. Geriausių ir prasčiausių tikslumą gavusių klasikinių modelių analizė

Šiame skyrelyje lyginami du klasikinių klasifikatorių rezultatai. Kairėje paveikslo pusėje yra pateikta geriausio rezultato painiavos matrica, dešinėje, blogiausio (žr. 17 pav.).

SVM modelis demonstruoja gerokai geresnius rezultatus, palyginti su naivaus Bajeso modeliu. Kairėje pusėje esančioje painiavos matricoje matome, kad SVM teisingai klasifikuoja žymiai didesnę skaičių sakinių. 0 klasėje SVM teisingai suklasifikuoja 1335 sakinius, o suklysta 28 ir 280 kartų. 1 klasėje SVM beveik idealiai veikia su 1789 teisingomis klasifikacijomis ir viso tik 12 klaidingų priskyrimų. 2 klasėje taip pat matome gerus rezultatus su 1311 teisingų klasifikacijų su 59 klaidingų priskyrimų.

Skirtingai veikia naivusis Bajeso modelis, kurio painiavos matrica vaizduojama dešinėje pusėje, rodo žymiai daugiau klaidingų priskyrimų. Jau 0 klasėje jis teisingai klasifikuoja 1089 pavyzdžius, tačiau suklysta 83 ir 471 kartus. 1 klasėje matome 1348 teisingus priskyrimus, bet daugiau nei trečdalis yra neteisingai priskirti 134 ir 319. Ir 2 klasėje rezultatai blogiausi, nes teisingų priskyrimų 670 yra kur kas mažiau nei klaidingų priskyrimų, kurių kartu yra 700.

Analizė rodo, kad SVM modelis su TF-IDF funkcijomis žymiai pranoksta naivaus Bajeso modelį su Word2Vec požymių išskyrimo metodu. SVM modelis su TF-IDF vektorizacijos metodu yra tvirtesnis ir efektyvesnis šiai klasifikavimo užduočiai spręsti.



17 pav. Painiavos matricos. Geriausias ir blogiausias rezultatas

Gauti naivaus Bajeso rezultatai apskaičiuojant įvairias metrikas:

- Klasė 1:
 - Tikslumas (angl. Precision) gautas 0.6661;
 - Atšaukimas (angl. Recall) gautas 0.6628;

- F1 balas (angl. F1 Score) gautas 0.6644.
- Klasė 2:
 - Tikslumas (angl. Precision) gautas 0.7842;
 - Atšaukimas (angl. Recall) gautas 0.7485;
 - F1 balas (angl. F1 Score) gautas 0.7659.
- Klasė 3:
 - Tikslumas (angl. Precision) gautas 0.4589;
 - Atšaukimas (angl. Recall) gautas 0.4891;
 - F1 balas (angl. F1 Score) gautas 0.4735.

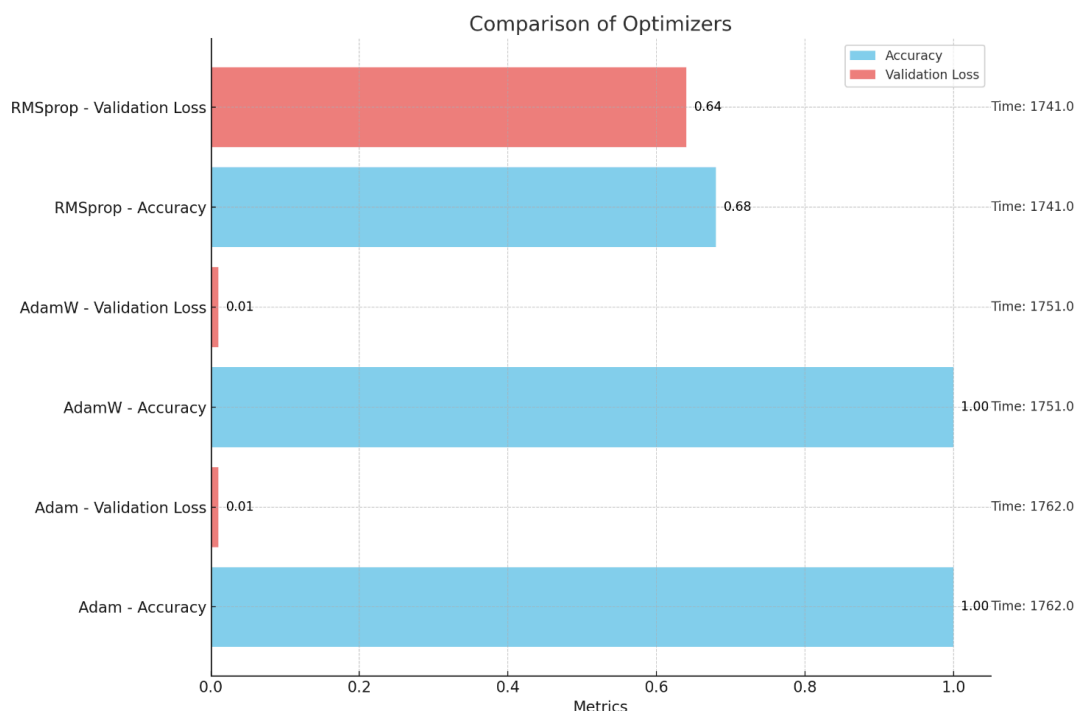
Klasė 0 pasiekė tikslumą 0.6661, atšaukimą 0.6628 ir F1 balą 0.6644, kas rodo vidutinišką modelio gebėjimą klasifikuoti. Klasė 1 pasiekė tikslumą 0.7842, atšaukimą 0.7485 ir F1 balą 0.7659, indikuodama gana gerą klasifikaciją. Klasė 2 pasiekė tikslumą 0.4589, atšaukimą 0.4891 ir F1 balą 0.4735, kas rodo žemą modelio gebėjimą su daug klaidingų teigiamų ir neigiamų

Įtaką blogiems rezultatams galėjo nulemti tam tikras duomenų disbalansas, panašios 3 klasės savybės su kitų klasės savybėmis arba tiesiog modelio ribotumas tokiems uždaviniams spręsti.

3.5. Gautų rezultatų įžvalgos taikant pažangųjį BERT modelį

Šiame poskyryje yra pateikiami pastebėjimai sukurtų BERT modelių su skirtingais optimizatoriais variacijos. Išanalizavus duomenis pateikiamos įžvalgos kokią įtaką daro skirtingi optimizatoriai.

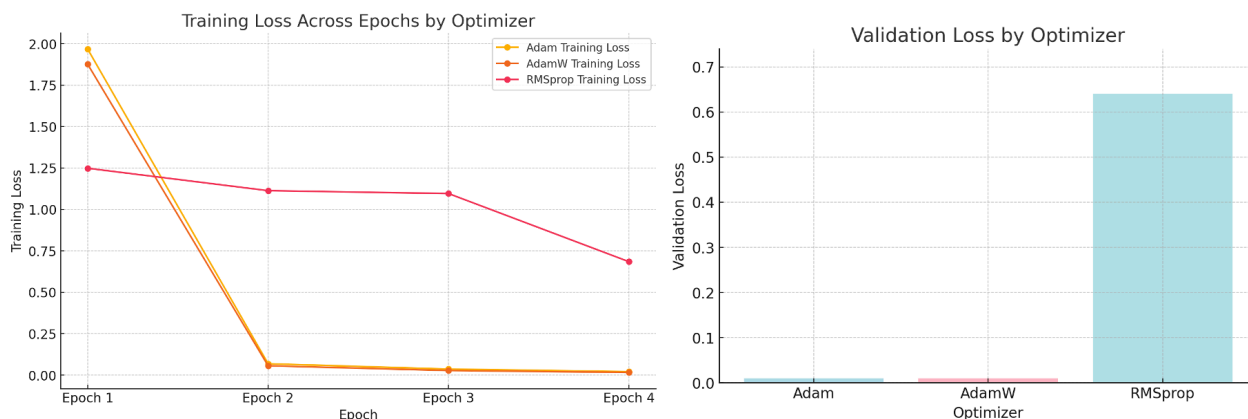
Iš pateiktų BERT modelio rezultatų su skirtingais optimizatorių algoritmais (Adam, AdamW ir RMSprop) matyti (žr. 18 pav.), kad optimizatoriaus pasirinkimas turi įtaką modelio mokymosi efektyvumui. Adam ir AdamW algoritmai pasirodė itin efektyviai, pasiekdami maksimalų tikslumą (1) ir F1 įverčius (1), kai tuo tarpu RMSprop algoritmas parodė ženkliai silpnesnius rezultatus su didesniais mokymosi nuostoliais (angl. training loss) ir gautu žemesniu tikslumo rezultatu (0.68). Galima teigti, jog RMSprop nėra efektyvus šiam specifiniam duomenų rinkiniui. Apmokymo laikai tarp skirtingų optimizatorių beveik nesiskiria, tačiau našumo skirtumai yra akivaizdūs. Vadinasi, Adam ir AdamW yra geriau pritaikyti siekiant aukšto modelio našumo šioje specifinėje užduotyje.



18 pav. Gautas tikslumas su skirtingais optimizatoriais

Kalbant apie nuostolių vertę, Adam ir AdamW optimizatoriai pasiekia beveik nulinę treniravimo nuostolį jau po antros epochos, tuo tarpu RMSprop optimizatorius rodo lėtą nuostolio mažėjimą (žr. 19 pav.). Galime manyti, jog padidinus epochų skaičių, nuostolių vertė dar labiau priartėtų prie nulio, tačiau tokiu atveju apmokymo laikas išaugtų labai smarkiai.

Dešinėje grafikas rodo validacijos nuostolius, kur Adam ir AdamW optimizatoriai turi žymiai mažesnę nuostolį, palyginti su RMSprop. Adam ir AdamW optimizatoriai yra efektyvesni šiam treniravimo duomenų rinkiniui.



19 pav. Nuostolių verčių pokytis su skirtingais optimizatoriais

Apibendrinant visus rezultatus, galime teigti, jog iš klasikinių klasifikavimo metodų su TF-IDF vektorizatoriumi stabiliai gerai pasirodė visi sukurti modeliai, nes daugumos tikslumas viršijo

0.9. Taikant kitus požymių išskyrimo metodus tikslumas krenta. Taikant Word2Vec tikslumas svyruoja tarp 0.65 ir 0.76, tuo tarpu taikant GloVe metodą svyruoja tarp 0.66 ir 0.85. Papildomas duomenų apdorojimas daro įtaką tik, kai yra taikoma GloVe vektorizacija, kitais atvejais įtaka yra labai minimaliai (0.1-0.2 tikslumo skirtumas). Kalbant apie pažangųjį modelį BERT ir jo tikslumo rezultatus, matome, jog geriausiai ir efektyviausiai veikia būtent su Adam ir AdamW optimizatoriais, kurių tikslumas siekia 1, nors tai kelia abejonių dėl permokymo. Norint patikrinti ar modelis persimokė, vienas iš būdų yra sudaryti atskirą testavimo rinkinį, su visiškai nematytais modeliui duomenimis. Ir tuomet patikrinti ar geba modelis toliau teisingai klasifikuoti duomenis į nustatytas kategorijas. Šis papildomas tyrimas nebus atliekamas šiame darbe.

IŠVADOS

1. Buvo atlikta srities analizė, apžvelgti įvairūs teksto klasifikavimo metodai ir jų pritaikymo galimybės. Apžvelgti tiek tradiciniai klasifikavimo modeliai, kaip naivaus Bajeso klasifikatorius, daugianarė logistinė regresija ar atraminių vektorių mašina, tiek pažangus modelis, kaip BERT. Aprašyti kiekvieno metodo privalumai bei trūkumai padėjo pasirinkti tinkamo klasifikavimo algoritmus tolesniam tyrimui.
2. Sudaryta tyrimo metodika, kurioje aprašomi skirtingų metodų pasirinkimai ir viso tyrimo variacijos. Tyrimui atlikti parinkti skirtingi požymių išskyrimo metodai TF-IDF, „Word2Vec“, „GloVe“ ir skirtingi klasifikatoriai: Naivusis Bajeso klasifikatorius, daugianarė logistinė regresija, SVM ir BERT. Taip pat BERT su trimis skirtingais optimizatoriais.
3. Tinkamas duomenų rinkinys sudarytas paimant tekstus iš CAPEC ir MITRE ATT&CK žinių bazių. Pradžioje sprendžiant duomenų balansavimo klausimą, duomenų rinkinys buvo tvarkomas taikant sinonimų keitimą, atgalinį vertimą bei naujų tekstų generavimą su GPT-2. Tokiu būdu siekiama padidinti duomenų įvairovę ir subalansuoti klases. Galiausiai duomenų valymas apėmė specialių simbolių šalinimą, raidžių sumažinimą, tarpų koregavimą ir dublikatų šalinimą.
4. Galiausiai atlikus tyrimą, buvo nustatyta, kad klasikiniai metodai su TF-IDF požymių išskyrimo metodu stabiliai pasiekia aukštą tikslumą su šiame tyrime sudarytu duomenų rinkiniu. Pažangusis BERT modelis, naudojant optimizatorius, tokius kaip „Adam“ ir „AdamW“, pasiekė itin aukštą tikslumą. Rezultatai rodo, kad šie pasirinkti metodai yra itin veiksmingi ir gali būti taikomi automatizuotai tekstinių dokumentų analizei ir klasifikavimui.

LITERATŪROS SĄRAŠAS

- PyTorch Team. (2024). *Torch*. Nuskaityta iš PyPI: <https://pypi.org/project/torch/>
- Amazon Web Services. (2024). *What is Text Classification?* Nuskaityta iš AWS: <https://aws.amazon.com/what-is/text-classification/>
- aniketguhnn4. (2024). Why an Increasing Validation Loss and Validation Accuracy Signifies Overfitting? *GeeksForGeeks*.
- Bhandari, A. (2020). Understanding & Interpreting Confusion Matrix in Machine Learning. *Analytics Vidhya*.
- Bird, S., Loper, E., & Klein, E. (2023). *Natural Language Toolkit (NLTK)*. Nuskaityta iš NLTK: <https://www.nltk.org/>
- DataScientest Team . (2023). *Scikit Learn: Discover the Python library dedicated to Machine Learning*. Nuskaityta iš DataScientest: <https://datascientest.com/en/scikit-learn-discover-the-python-library-dedicated-to-machine-learning>
- Dharani. (2020). All About ML — Part 4: Evaluation metrics in classification algorithms. *All About ML*.
- Han, S. (2020). *Googletrans*. Nuskaityta iš PyPI: <https://pypi.org/project/googletrans/>
- Hashemi-Pour, C. &. (2024). BERT language model. *TechTarget*.
- Kirvan. (2021). *Top 10 IT security frameworks and standards explained*. Nuskaityta iš TechTarget: <https://www.techtarget.com/searchsecurity/tip/IT-security-frameworks-and-standards-Choosing-the-right-one>
- Kohli. (2023). Notes on BERT tokenizer and model. *Medium*.
- Kowsari, Meimandi, Heidarysafa, Mendu, Barnes, Brown. (2019). Text Classification Algorithms: A Survey.
- Li, Peng, Li, Xia, Yang, Sun, . . . He. (2022). A Survey on Text Classification: From Traditional to Deep Learning. *ACM Transactions on Intelligent Systems and Technology*, 13(2), 31-41.
- Microsoft. (2023). *Custom Text Classification: Transparency Note*. Nuskaityta iš Microsoft Learn: <https://learn.microsoft.com/en-us/legal/cognitive-services/language-service/ctc-transparency-note>
- Pierre, S., & Whitfield, B. (2023). *Python Data Visualization With Seaborn and Matplotlib*. Nuskaityta iš Built In: <https://builtin.com/data-science/data-visualization-tutorial>
- Reitz. (2023). *Requests*. Nuskaityta iš PyPI: <https://pypi.org/project/requests/>
- Richardson. (2024). *Beautiful Soup*. Nuskaityta iš PyPI: <https://pypi.org/project/beautifulsoup4/>

- Shaip Team. (2022). *Text Classification: Importance, Use Cases, and Process*. Nuskaityta iš Shaip:
<https://www.shaip.com/blog/text-classification-importance-use-cases-and-process/>
- Siddharth. (2021). Feature Extraction and Embeddings in NLP: A Beginner's Guide to Understand Natural Language Processing. *Analytics Vidhya*.
- Solms. (2019). Information security management: why standards are important. 50-58.
- souvikmandal. (2024). How to use Google Colab. *GeeksForGeeks*.
- Subasi, A. (2020). Text Classification. Esantis *Practical Machine Learning for Data Analysis Using Python*. Sciencedirect.com.
- Taloba. (2018). A Comparative Study on using Principle Component Analysis with Different Text Classifiers.
- Team, Great Learning. (2023). Multinomial Logistic Regression. *Great Learning Blog*.
- The Hugging Face Team. (2024). *Transformers*. Nuskaityta iš PyPI:
<https://pypi.org/project/transformers/>
- W3Schools. (2023). *NumPy Introduction*. Nuskaityta iš W3Schools:
https://www.w3schools.com/python/numpy/numpy_intro.asp
- Zareapoor. (2015). Feature Extraction or Feature Selection for Text Classification: A Case Study on Phishing Email Detection. *International Journal of Information Engineering and Electronic Business*, 60-65.