

VENTSPILS AUGSTSKOLA  
INFORMĀCIJAS TEHNOLOĢIJU FAKULTĀTE

BAKALAURA DARBS

**Ziņu portālu rakstu klasifikācija (ar  
mašīnmācīšanās algoritmiem)**

Autors:

Ventspils Augstskolas  
Informācijas tehnoloģiju fakultātes  
bakalaura studiju programmas  
„Datorzinātnes”  
3. kursa students  
Matīss Kalniņš  
Matrikulas Nr. 23020018

---

(paraksts)

Fakultātes dekāns:

doc. Dr.sc.comp. Vairis Caune

---

(paraksts)

Zinātniskais vadītājs:

Mg.sc.comp. Agris Traškovs

---

(paraksts)

Recenzents:

---

(Ieņemamais amats, zinātn. nosaukums, vārds, uzvārds)

---

(paraksts)

Ventspils, 2023

# Saturs

<b>Anotācija</b>	<b>4</b>
<b>Abstract</b>	<b>5</b>
<b>Izmantotie saīsinājumi un termini</b>	<b>6</b>
<b>Ievads</b>	<b>7</b>
<b>1. Teorija</b>	<b>8</b>
1.1. Dabiskās valodas apstrāde	8
1.2. Mašīnmācīšanās	8
1.3. Tekstu klasifikācija	9
1.3.1. Klasiskie algoritmi tekstu klasifikācijai	10
1.3.2. Neironu tīkli	13
1.4. Modeļu novērtēšana un validācija	14
1.4.1. Klasifikācijas mēri	14
1.5. Tekstu priekšapstrādei	16
1.5.1. Vārdu atdalīšana	16
1.5.2. Sakņu atdalīšana un lemmatizācija	16
1.5.3. Stopvārdu dzēšana	16
1.5.4. Vektorizācija	17
1.5.5. Vārdlietojuma kartējums	18
1.5.6. Pazīmju izvēle	19
1.6. Biežākās problēmas tekstu klasifikācijā	19
1.6.1. Vārdu neskaidrība	19
1.6.2. Tekstu nevienmērība	20
1.6.3. Pārmērīga pielāgošana	20
1.7. Mašīnmācīšanās rīki	21
1.7.1. scikit-learn	21

1.7.2. Tensorflow . . . . .	21
<b>2. Praktiskā daļa . . . . .</b>	<b>22</b>
2.1. Datu izgūšana no ziņu portāliem ar rāpuli . . . . .	22
2.2. Tekstu priekšapstrāde . . . . .	23
2.2.1. Stopvārdu atmešana . . . . .	23
2.3. Algoritmu rezultāti . . . . .	24
2.3.1. Atbalsta vektora mašīnas . . . . .	24
2.3.2. Naivā Bajesa metode . . . . .	25
2.3.3. Loģistiskā regresija . . . . .	26
2.3.4. Lēmumu koki . . . . .	27
2.3.5. Konvolūcijas neironu tīkli . . . . .	28
2.4. Modeļu salīdzinājums . . . . .	29
<b>Secinājumi un priekšlikumi . . . . .</b>	<b>30</b>
<b>Izmantotās literatūras un avotu saraksts . . . . .</b>	<b>31</b>
<b>Galvojums . . . . .</b>	<b>32</b>
<b>Pielikumi . . . . .</b>	<b>33</b>
1. Ar rāpuļa palīdzību izgūta raksta piemērs . . . . .	33
2. Klasificējamo kategoriju rakstu garumi . . . . .	34

## ANOTĀCIJA

<b>Darba nosaukums:</b>	Ziņu portālu rakstu klasifikācija (ar mašīnmācīšanās algoritmiem)
<b>Darba autors:</b>	Matīss Kalniņš
<b>Darba vadītājs:</b>	Mg.sc.comp. Agris Traškovs
<b>Darba apjoms:</b>	32. lpp, XX tabulas, 12 attēli, XX formulas, 10 bibliogrāfiskās norādes, 2 pielikumi
<b>Atslēgas vārdi:</b>	Dabiskās valodas apstrāde, mašīnmācīšanās, tekstu klasifikācija

Bakalaura darbā ir aprakstīta dabīgās valodas apstrāde un kā mašīnmācīšanās metodes var palīdzēt risināt teksta klasifikācijas problēmu, konkrēti apskatot tieši ziņu klasifikāciju latviešu valodas rakstiem.

Darba ietvaros tiek ievākta rakstu kopa no ziņu portāliem un pārbaudīts kāda pieeja sniedz augstāko precizitāti teksta klasifikācijai latviešu valodā. Tiek izvērtētas un salīdzinātās dažādas pazīmju izvēles pieejas un apmācības algoritmi (naivā Bajesa metode, loģistiskā regresija, lēmumu koki, atbalsta vektora mašīnas, neironu tīkli).

Papildus apskatītas dažādas atvērtā pirmkoda bibliotēkas mašīnmācīšanās problēmu risināšanai kā scikit-learn un Tensorflow, to praktiskais pielietojums dabīgo valodu apstrādei.

## ABSTRACT

<b>The title:</b>	Classification of news articles (with machine learning algorithms)
<b>Author:</b>	Matīss Kalniņš
<b>Academic Advisor:</b>	Mg.sc.comp. Agris Traškovs
<b>The volume of the work:</b>	32. pages, XX tables, 12 images, XX equations, 10 literature sources, 2 appendices
<b>Keywords:</b>	Natural language processing, machine learning, text classification

The bachelor thesis describes natural language processing and how machine learning methods can help to resolve text classification problems, focusing specifically the classification of news articles in the Latvian language.

As part of this work a data set of articles is gathered from Latvian news websites and the best approach is researched for achieving the highest accuracy of Latvian text classification. Various feature generation approaches and learning algorithms (e.g. Naïve Bayes, logistic regression, decision trees, support vector machines, neural networks) are evaluated and compared.

In addition, various open source libraries for machine learning as scikit-learn and Tensorflow along with their practical applications for natural language processing are explored as part of this work.

## **IZMANTOTIE SAĪSINĀJUMI UN TERMINI**

**DVA** - Dabisko valodu apstrāde (angliski - natural language processing)

**TF-IDF** - Terminu biežums - inversais dokumentu biežums (angliski - term frequency  
- inverse document frequency)

**PA** - pareiza atbilde (angliski - true positive)

**PA** - pareiza atbilde (angliski - true positive)

**PN** - pareiza neatbilde (angliski - true negative)

**KA** - kļūdaina atbilde (angliski - false positive)

**KN** - kļūdaina neatbilde (angliski - false negative)

## IEVADS

Mūsdienās internets ir kļuvis par galveno informācijas avotu lielai daļai cilvēku, kuri ikdienā ar dažādu mediju palīdzību caur to gūst informāciju par jaunākajām aktualitātēm savā rajonā, valstī un pasaulē. Svarīga loma informācijas iegūšanā un izplatīšanā ir arī pareizai teksta klasifikācijai, lai šī informācija sasniegtu vēlamā lasītāju. Pārsvārā problēma tiek atrisināta autoram klasificējot savu darbu jau izveides procesā, tomēr bieži ar to vien nepietiek – tiek pārpublicēti raksti no ārējiem resursiem, mainās kategoriju iedalījums, aktuālas kļūst jaunas tēmas u.t.t. Lai gan arī šādos gadījumos klasifikāciju iespējams darīt manuāli, pie liela informācija apjoma kļūst jēgpilni šo klasifikāciju automatizēt ar mašīnmācīšanās algoritmiem, ietaupot laiku un resursus. Autors plāno izpētīt metodes ar kurām iespējams veikt šādu tekstu klasifikāciju un kā tās ir piemērotas latviešu valodas tekstu apstrādei, konkrētāk apskatot tieši ziņu portālu rakstu klasifikācijas problēmu. Darba mērķis ir izveidot mašīnmācīšanās modeli, kas ar augstu precizitāti spētu klasificēt ziņu portālu rakstus. Lai sasniegtu šo mērķi, tiek izvirzīti sekojoši uzdevumi:

1. Veikt literatūras izpēti par mašīnmācīšanos un tekstu klasifikāciju
2. Izveidot rāpuli ar kura palīdzību izgūt un marķēt ziņu portālu rakstus, pielietojamus modeļu apmācībā
3. Implementēt dažādus mašīnmācīšanās algoritmus tekstu klasifikācijai
4. Izpētīt kā dažāda tekstu priekšapstrāde / pazīmju izveides metodes ietekmē klasifikācijas rezultātus
5. Veikt precizitātes novērtējumus un salīdzināt cik labi dažādi algoritmi spēj veikt latviešu valodas tekstu klasifikāciju

Lai veiktu algoritmu implementēšanu un analīzi tiks izmantota programmēšanas valoda Python un plaši pielietotas bibliotēkas mašīnmācīšanās problēmu risināšanai (scikit-learn, Tensorflow)

# 1. TEORIJA

## 1.1. Dabiskās valodas apstrāde

Dabiskās valodas apstrāde (angliski – natural language processing jeb NLP) ir daudzozaru joma, kas apvieno lingvistikas, datorzinātnes un mašīnmācīšanās elementus, lai ļautu datoriem saprast, interpretēt un ģenerēt cilvēka valodu.

DVA sastāv no vairākām pamata komponentēm:

- Tokenizācija: teksta sadalīšanas process vārdos vai frāzēs (tokenos)
- Morfoloģiskā marķēšana: gramatikas marķējumu piešķiršana vārdiem
- Sintakses parsēšana: teikumu gramatiskās struktūras analīze
- Nosaukto entitāšu atpazīšana: nosaukumu atpazīšana un kategorizācija, piemēram, personvārdi, datumi un atrašanās vietas
- Lemmatizācija: vārdu pārveidošana pamatformā

Pielietojot daļu no šīm komponentēm tālāk iespējami sarežģītāki pielietojumi teksta apstrādei – kategorizācijai, noskaņojuma analīzei, mašīntulkošanai, čatbotu izveidei.

Sākotnējās DVA sistēmas balstījās uz manuāli izstrādātām noteikumiem, taču šīs sistēmas ir ierobežotas ar savu nespēju apstrādāt cilvēka valodas daudzveidību un sarežģītību, kā rezultātā DVA sistēmas mūsdienās bieži tiek veidotas tieši ar mašīnmācīšanās iesaisti.

## 1.2. Mašīnmācīšanās

Mašīnmācīšanās ir mākslīgā intelekta nozare, kas nodarbojas ar datorprogrammu izstrādi, kuras, izmantojot algoritmus un statistikas modeļus, mācās no datiem un uzlabo savu precizitāti. Toms Mičels savukārt apraksta mašīnmācīšanās jomu, izvirzot centrālo jautājumu, ko tā pēta: "Kā mēs varam izveidot datoru sistēmas, kas automātiski uzlabojas, iegūstot pieredzi, un kādi ir pamatlikumi, kas nosaka visus mācīšanās procesus?" [1].



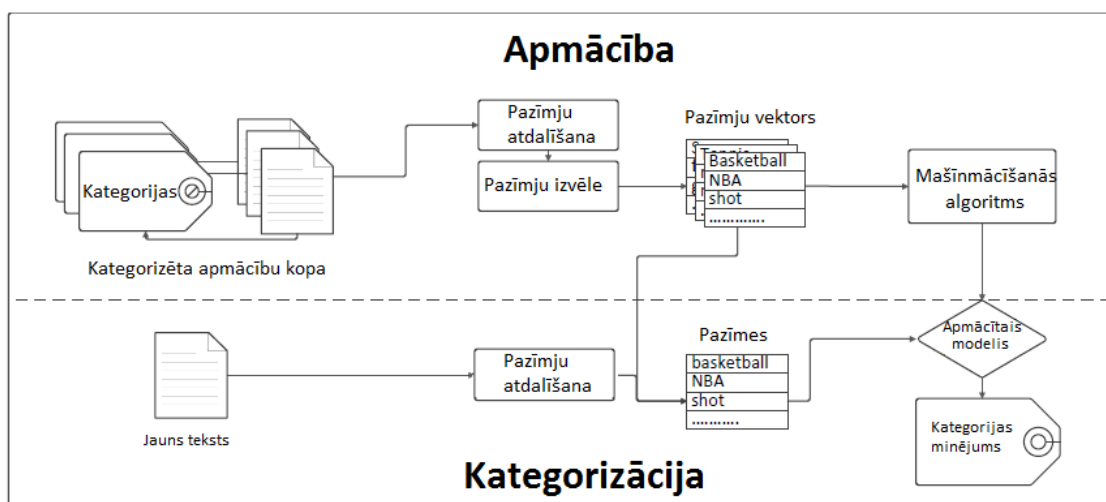
Induktīvā mācīšanās ir mašīnmācīšanās apakšnozare, kas specifiski nodarbojas ar modeļu mācīšanos no novērojumiem. Šajā nozarē mācīšanās uzdevumi bieži tiek raksturoti, pamatojoties uz atgriezenisko saiti, kas tiek sniegta apmācības veicējam [2], šādi:

- Uzraudzītā mācīšanās: Apmācības procesā tiek sniegts vēlams izvads katram novērojumam. Mērķis ir iemācīties funkciju, kas paredz pareizo izvades vērtību brīdī kad tiek sniegts iepriekš neredzēts novērojums.
- Neuzraudzītā mācīšanās: Apmācības laikā netiek sniegts izvads. Mērķis ir atklāt paraugus un regulāras iezīmes datus.
- Stimulētā mācīšanās: Šis ir īpašs uzraudzītās mācīšanās gadījums, kur apmācības posmā tiek sniegta atlīdzība pēc katras darbības.

Uzraudzītās mācīšanās gadījumus, kad uzdevums ir iemācīties diskrēti vērtētu funkciju, sauc par klasifikāciju. Uzdevums, kad jāiemācās nepārtraukti vērtēta funkcija, tiek saukts par regresiju. Savukārt klasterošana ir neuzraudzītās mācīšanās uzdevums, kas atrod līdzīgu objektu grupas datu kopā.

### **1.3. Tekstu klasifikācija**

Teksta klasifikācijas mērķis ir tekstu piesaiste konkrētai kategorijai, balstoties uz teksta saturu. Šāda klasifikācija ir ļoti izplatīta ziņu portālos un citur, kur nepieciešams kategorizēt lielu rakstu daudzumu, piemēram akadēmisko darbu datubāzēs. Mašīnmācīšanās algoritmi ļauj rast risinājumu automātiskai šādu tekstu kategorizēšanai un daudzām citām klasifikācijas problēmām, piemēram, ar augstu precizitāti noteikt vai ienākošais e-pasts ir vai nav mēstule. Iespējams arī veikt sentimentu analīzi un noteikt cilvēku attieksmi par kādu konkrētu tematu, piemēram, M. Kandas ir apskatījis kā ar tekstu klasifikācijas palīdzību noteikt negatīvu attieksmi pret likumsargiem, balstoties uz konkrēta lietotāja ierakstiem vietnē YouTube [3].



1.1. att. Mašīnmācīšanās tekstu klasifikācijai

Teksta klasifikācijas piemēru ar mašīnmācīšanās pielietojumu iespējams redzēt attēlā 1.1.. Pieņemsim, ka dota datu kopa ar sporta ziņām, un risināmā problēma ir - kā klasificēt jaunu dokumentu, piešķirot tam atbilstošā sporta veida kategoriju. Sākumā būs nepieciešami apmācības dokumenti (ar klases marķējumiem) no kuriem mācīties. Tālāk katru ziņu mēs pārveidojam par pazīmju kopu, kuru vektorizētā veidā mēs varam padot tālāk mašīnmācīšanās algoritmam. Ar šo informāciju algoritms izveido modeli, kas var paredzēt iepriekš neredzētu tekstu kategoriju.

### 1.3.1. Klasiskie algoritmi tekstu klasifikācijai

#### Naivā Bejesa metode

Naivā Bejesa mašīnmācīšanās algoritms bieži tiek lietots tieši klasifikācijas uzdevumos tā veikspējas un efektivitātes dēļ. Tas balstās uz Bejesa teorēmu, kas ir viena no pamat-teorēmām varbūtību teorijā. Šī teorija apraksta notikuma varbūtību, pamatojoties uz iepriekš zināmiem datiem. Teksta klasifikācijas kontekstā teorēma palīdz mums aprēķināt varbūtību dokumenta piederībai noteiktai klasei.

Bejesa teorēmu var izteikt šādi:

$$P(klase|dokuments) = \frac{P(klase) \cdot P(dokuments|klase)}{P(dokuments)} \quad (1.1)$$

Kur formulā 1.1.:

- $P(klase|dokuments)$  ir varbūtība, ka dokuments pieder norādītajai klasei.

- $P(klase)$  ir apriorā klases varbūtība.
- $P(dokuments|klase)$  ir varbūtība novērot dokumentu, zinot klasi.
- $P(dokuments)$  ir varbūtība, ka dokuments parādās datu kopā.

”Naivais” aspekts Naivajā Bejesā nāk no pieņēmuma, ka pazīmes (vārdi tekstā) ir neatkarīgas. Citiem vārdiem sakot, mēs pieņemam, ka katra vārda klātbūtne vai neesamība dokumentā ir neatkarīga no citu vārdu klātbūtnes vai neesamības. Šis ir vienkāršs pieņēmums, bet tāds kurš bieži darbojas praksē.

## Loģistiskā regresija

Loģistiskā regresija modelē varbūtību, ka dokuments pieder konkrētai klasei, izmantojot loģistisko (sigmoidālo) funkciju [4], kas nodrošina, ka izvades varbūtība ir starp 0 un 1. Loģistiskā funkcija tiek definēta šādi:

$$P(y = 1|x) = \frac{1}{1 + e^{-z}} \quad (1.2)$$

Kur formulā 1.2.  $P(y = 1|x)$  ir varbūtība, ka dokuments pieder klasei 1 un  $z$  ir lineāra kombinācija no ievades iezīmēm un modeļa parametriem. Lineāra kombinācija tiek aprēķināta šādi:

$$z = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n \quad (1.3)$$

Kur formulā 1.3.  $x_1, x_2, \dots, x_n$  ir skaitliskās iezīmes, kas izgūtas no teksta dokumenta un  $\theta_0, \theta_1, \dots, \theta_n$  ir modeļa parametri, arī saukti par svāriem vai koeficientiem.

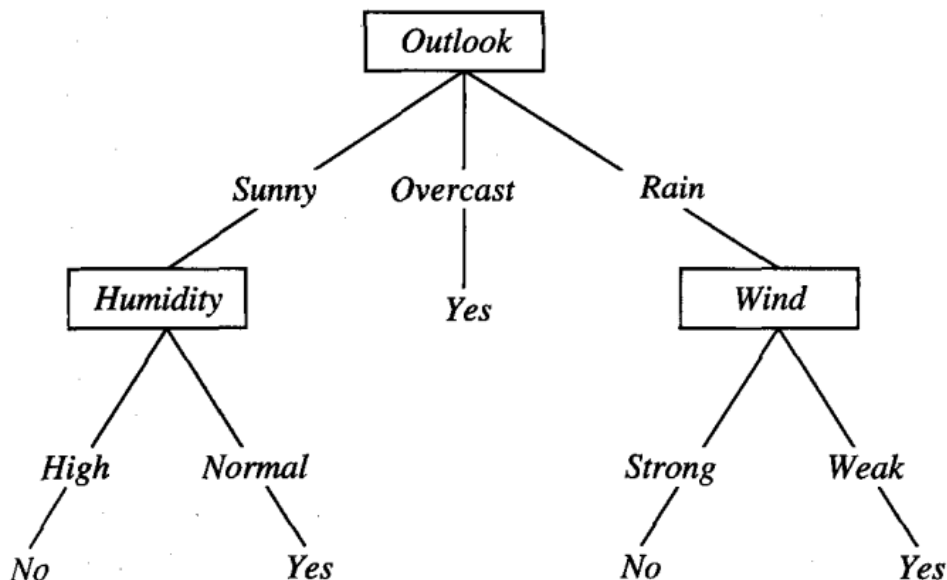
Apmācības fāzē loģistiskās regresijas modelis mācās optimizēt savus parametrus ( $\theta$ ) no marķētajiem datiem, lai iegūtu pēc iespējas pareizākus minējumus.

## Lēmumu koki

Lēmumu koka klasifikators izmanto koka modeli, lai prognozētu teksta klasi. Koks sastāv no viena saknes mezgla, kas ir uzskatāms par klasifikatora sakuma punktu. Pārējie mezgli ir lapu mezgli, ja tiem nav zaru, vai iekšējie mezgli. Iekšējie mezgli un saknes mezgls ir iezīme un pārbaude, kas jāveic šai iezīmei. Katrs iespējamais testa rezultāts ir mezgla atzars, kas ved uz nākamo mezglu. Šādi veicot pārbaudes uz katra mezgla, tiek iziets caur visiem mezgliem līdz pirmajam lapu mezglam. Lapu mezgli galu galā norāda uz klasi, kurai

šis teksts pieder. Citiem vārdiem – klase tiek paredzēta, sekojot ceļam no koka saknes mezgla, līdz tas saskaras ar lapas mezglu [5].

Apmācības algoritma mērķis šajā gadījumā ir izveidot lēmumu koku, pamatojoties uz apmācību datu piemēriem. Tomēr algoritmam ir jāizvairās veidot koku, kas pārmērīgi atbilst apmācības datiem. Tāpēc optimālais koks ir mazākais lēmumu koks, kas vislabāk atšķir klases.

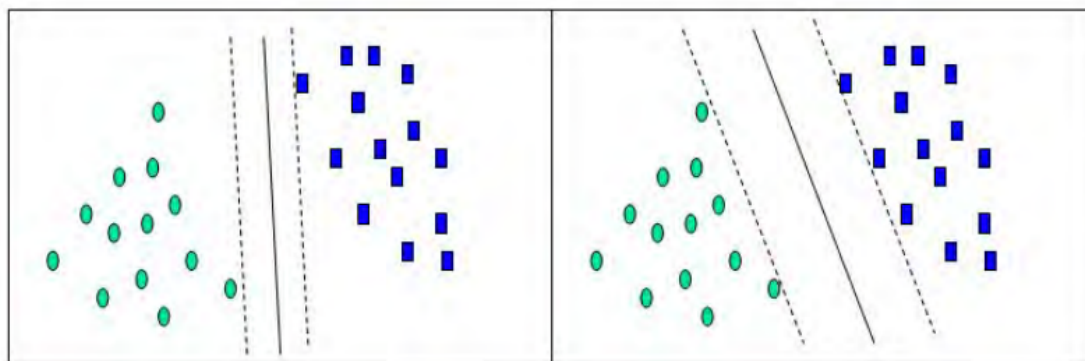


1.2. att. Lēmumu koka ilustrācija [5]

Piemērā 1.2. apskatām vienkāršu šāda koka reprezentāciju, kur veicam klasifikāciju par to vai šis ir piemērots laiks tenisa spēlei ārpus telpām. Ar ieejas datiem kā laikapstākļi (outlook) – saulaini (sunny), mitrums (humidity) – augsts (high), mēs virzītos pa mezgliem “Laikapstākļi”, “Mitrums” līdz lapas mezglam kurš klasificētu laiku kā nepiemērotu tenisa spēlei.

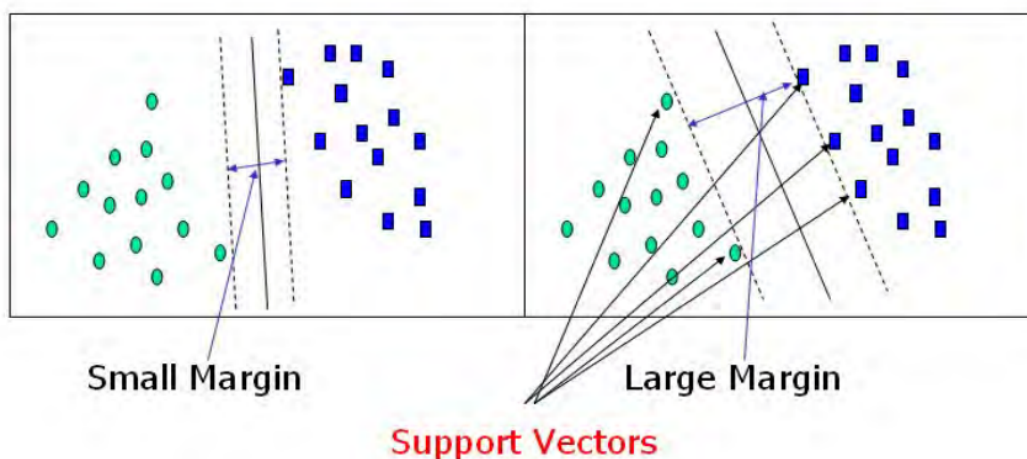
### Atbalsta vektoru mašīnas (SVM)

Atbalsta vektora mašīnas [6] (angliski - support vector machines jeb SVM) ir pārraudzītās mācīšanās algoritms kurš ir diezgan populārs tieši klasificēšanas problēmu risināšanā. Šis algoritms veic klasifikāciju konstruējot n-dimensiju hiperplakni kura optimāli ierobežo datus divās nošķirtās kategorijās. Lai vieglāk ilustrētu algoritma darbību, varam apskatīt divdimensiju piemēru.



1.3. att. Atbalsta vektora mašīnas algoritma ilustrācija [6]

Šajā piemērā 1.3. gadījumi ar vienu kategoriju atrodas pa kreisi (apzīmēti ar zaļiem apliem) un ar otru kategoriju – pa labi (apzīmēti ar ziliem kvadrātiem). Atbalstu vektora mašīnas analīze mēģinās atrast 1-dimensijas hiperplakni (līniju) kura atdala datus balstoties uz kategoriju kurai tie pieder. Ir praktiski neierobežots līniju skaits, kas spētu veikt šādu nodalījumu, attēlā norādīti 2 piemēri un atliek gūt atbildi uz jautājumu – kura līnija ir labāka kategorizācijas veikšanai vispārīgā gadījumā. Raustītās līnijas, kuras zīmētas paralēli atdalošajai līnijai, iezīmē attālumu starp atdalošo līniju un tai tuvāko vektoru. Šo attālumu sauc par robežu (angliski – margin). Vektori kas atrodas pie šīs robežas ir atbalsta vektori.



1.4. att. Robežas un atbalsta vektoru ilustrācija [6]

Atbalstu vektora mašīnas analīze atradīs līniju (vispārīgi – hiperplakni), kas novietota pēc iespējas lielāku robežu starp atbalsta vektoriem.

### 1.3.2. Neironu tīkli

TBD

## 1.4. Modeļu novērtēšana un validācija

Pēc apmācīta modeļa iegūšanas ir svarīgi novērtēt izveidotā modeļa veikspēju un to, cik labi tas spēs veikt tekstu klasifikāciju ar jauniem datiem. Viena no izplatītākajām novērtēšanas metodēm ir metode ar noturēšanu (holdout method), kur paraugu kopa tiek sadalīta divās daļās – apmācības kopa un testa kopa. Klasifikators tad tiek apmācīts ar apmācību kopu un validēts ar testa kopu. Šīs metodes mīnuss ir tas, ka apmācībai pieejama mazāka datu kopa. Cita negatīvā īpašība šai metodei ir arī tā, ka rezultāti ir atkarīgi no nevienlīdzīgā datu sadalījuma starp šīm abām kopām.

Cita metode ir nejauša paraugu atlase (Random Subsampling). Šī metode atkārtoti metodi ar noturēšanu vairākas reizes, lai labāk noteiktu modeļa veikspēju, tomēr arī šai metodei piemīt pirmās metodes negatīvās īpašības. Modeļa novērtēšana ar apmācību datiem nav ieteicama, jo tādējādi notiks pārmērīga pielāgošana (overfitting) un mašīnmācīšanās modelis iegaumēs apmācības datus, bet nespēs pareizi klasificēt jaunus datus.

Visbeidzot var izmantot arī šķērsvalidāciju (cross-validation). Šī metode sadala paraugu kopu  $k$  vienādās daļās un izmanto katru no šīm daļām tieši vienu reizi priekš testēšanas. Citas,  $k-1$ , reizes katra daļa tiek izmantota apmācībai.

### 1.4.1. Klasifikācijas mēri

Klasifikācijas problēma ir bieži apskatīts mašīnmācīšanās temats un visizplatītākie mēri, ar kuriem novērtēt modeļa precizitāti ir aprakstīti zemāk.

#### Pārpratumu matrica

Pārpratumu matrica ļauj pārredzami attēlot modeļa precizitāti modelim ar 2 un vairāk klasēm. Matrica sastāv no 2 asīm, kur uz  $x$  ass tiek attēlotas visas klases un uz  $y$  ass tiek attēloti klases minējumi. Katra matricas šūna satur minējumu skaitu attiecīgajai klasei un minētās klases kombinācijai.

Bināras klasifikācijas gadījumā pārpratumu matrica varētu būt attēlojama ar šādu tabulu:

**Pārpratuma matrica**

	+	-
+	PA	KA
-	KN	PN

Kur tabulas 1.1. vērtības raksturojamas šādi:

- PA - pareiza atbilde (tabulā - gadījumi, kad '+' tiek pareizi klasificēts)
- PN - pareiza neatbilde (tabulā - gadījumi, kad '-' tiek pareizi klasificēts)
- KA - kļūdaina atbilde (tabulā - gadījumi, kad '-' tiek nepareizi klasificēts kā '+')
- KN - kļūdaina neatbilde (tabulā - gadījumi, kad '+' tiek nepareizi klasificēts kā '-')

### Akurātums

Akurātums ir mērs, kurš norāda cik no minējumiem ir bijuši pareizi. Tas ir pareizo minējumu dalījums ar visu minējumu skaitu.

$$Akurātums = \frac{PA + PN}{PA + PN + KA + KN} \quad (2.4)$$

### Precizitāte

$$Precizitāte = \frac{PA}{PA + KA} \quad (2.5)$$

### Pārklājums

$$Pārklājums = \frac{PA}{PA + KN} \quad (2.6)$$

### F1 mērs

F1 mērs ir precizitātes un pārklājuma mēru harmoniskais vidējais.

$$F1 = \frac{(2 * precizitāte * pārklājums)}{(precizitāte + pārklājums)} \quad (2.7)$$

Atšķirībā no aritmētiskas vidējās vērtības, harmoniska vidējā vērtība tiecas uz mazāko vērtību no diviem mēriem, no tā izriet, ka F1 mērs būs zems ja precizitāte vai pārklājums ir zems.

## **1.5. Tekstu priekšapstrādei**

Pirms iespējams izveidot klasifikācijas modeli, DVA problēmvidē nepieciešams veikt teksta priekšapstrādi, lai dati būtu pielietojami tālākā apstrādē. Tas sevī ietver gan dažādu teksta fragmentu atmešanu, gan pārveidošanu formā kuru spētu saprast apmācības algoritmi. Tālāk apskatīti konkrēti priekšapstrādes soļi.

### **1.5.1. Vārdu atdalīšana**

Lai tekstu izmantotu klasificēšanā, ir jāspēj atdalīt atsevišķas šī teksta daļas. To iespējams paveikt dažādos veidos – tekstu iespējams sadalīt pa individuāliem vārdiem vai arī secīgu vārdu grupām. Vārdu atdalīšanai var izmantot dažādus atdalošos simbolus, piemēram atstarpī un jaunas līnijas sākuma simbolu (n'). Tomēr ne visi atdalošie simboli ir viennozīmīgi, piemēram punkts var būt gan kā teikuma beigas, gan kā daļa no konkrētas vērtības (skaitlis 10.5). Vārdu atdalīšana secīgās vārdu grupās izmanto n-grammas, kas ir n secīgu vārdu un/vai simbolu kopa tekstā. Izmantojot n-grammas, iespējams iegūt plašāku teksta kontekstu no atdalītajiem vārdiem.

### **1.5.2. Sakņu atdalīšana un lemmatizācija**

Apstrādājot tekstus, svarīgi ir arī ņemt vērā faktu, ka dažādos tekstos viens un tas pats vārds bieži tiek lietots dažādos locījumos un visbiežāk locījumam nav ietekme uz to vai teksts pieder konkrētai kategorijai vai nē. Svarīgi arī ņemt vērā, ka liels individuālo vārdu atkārtojums dažādos locījumos palielina klasifikācijai nepieciešamo laiku un resursus. Sakņu atdalīšana un lemmatizācija risina šo problēmu, pārvēršot vārdus to pamatformā.

### **1.5.3. Stopvārdu dzēšana**

Bieži vien noderīga ir arī tā sauktā stopvārdu (angliski - stopwords) dzēšana no apstrādāmajiem datiem. Tie ir vārdi kuri neietekmē teksta saturu un to biežais lietojums var atstāt negatīvu ietekmi uz klasifikācijas akurātumu. Latviešu valodā piemērs šādiem vārdiem būtu palīgvārdi, kuri tiek saprasti kā saikļi (un, bet, vai u.c.), prievārdi (uz, no u.c.), partikulas (arī, diezin, gan u.c.).



### 1.5.4. Vektorizācija

Teksta vektorizācija ir process, kurā teksta informācija tiek pārveidota skaitļu formā, ko tālāk savukārt var izmantot mašīnmācīšanai. Šis solis ir būtisks, jo lielākā daļa mašīnmācīšanās algoritmu strādā ar skaitļiem, bet teksta dati ir paši par sevi neskaitliski. Teksta vektorizācijai ir vairākas metodes, populārākās no tām uzskaitītas zemāk.

#### Vārdu maiss

Vārdu maiss – tā ir nesakārtota vārdu kopa, kur vārdu secība tiek ignorēta, saglabājot tikai vārdu biežumu dokumentā [7]. Teksts tiek pārveidots vektorā, kur katram unikālam vārdam dokumentā tiek piešķirts indekss, kā vērtību indeksā norādot konkrētā vārda biežumu. Ar šo pieeju mēs veicam šādus pieņēmumus:

- Tekstu iespējams analizēt, ignorējot vārdu / tekstvienību secību.
- Nepieciešams zināt tikai kuri vārdi / tekstvienības atrodas tekstā un cik bieži tie atkārtojas.

#### TF-IDF

Terminu biežums - inversais dokumentu biežums (angliski, saīsināti - TF-IDF) ir vektorizācijas paveids ko izmanto, lai novērtētu termina (vārda) svarīgumu dokumentā attiecībā uz dokumentu kopu (korpusu) [8].

Termina biežums (TF):

Termina biežums nosaka, cik bieži konkrēts termins parādās konkrētā dokumentā. To aprēķina kā attiecību starp termina parādīšanos dokumentā un kopējo terminu skaitu šajā dokumentā. Termina biežuma (TF) formula ir šāda:

$$TF(t, d) = \frac{f(t, d)}{|d|} \quad (2.8)$$

Kur:

- $TF(t, d)$  ir termina biežums terminam  $t$  dokumentā  $d$ .
- $f(t, d)$  ir termina  $t$  biežums dokumentā  $d$ .
- $|d|$  ir kopējais terminu skaits dokumentā  $d$ .

Inversais dokumenta biežums (IDF): Inversais dokumenta biežums nosaka cik unikāls vai svarīgs ir termins visā dokumentu kopā (korpusā). To aprēķina kā logaritmisku attiecību starp visu dokumentu kopā esošo dokumentu skaitu un dokumentu skaitu, kuros šis termins parādās. Inversās dokumentu biežuma (IDF) formula ir šāda:

$$IDF(t, D) = \log \left( \frac{|D|}{|d \in D : t \in d|} \right) \quad (2.9)$$

Kur:

- $IDF(t, D)$  ir termina  $t$  inversais dokumenta biežums kopā  $D$ .
- $|D|$  ir kopējais dokumentu skaits korpusā.
- $|d \in D : t \in d|$  ir dokumentu skaits, kas satur terminu  $t$ .

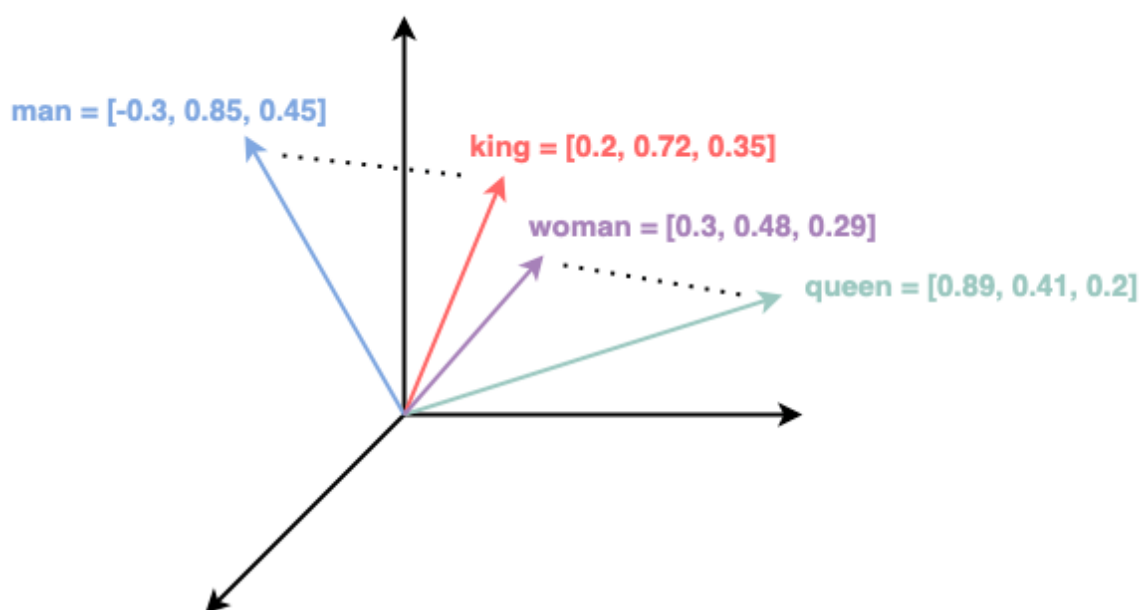
TF-IDF beigu rezultāts ir termina biežuma (TF) un inversā dokumenta biežuma (IDF) reizinājums:

$$TF-IDF(t, d, D) = TF(t, d) \times IDF(t, D) \quad (2.10)$$

TF-IDF rezultāts atspoguļo, cik svarīgs ir vārds konkrētajā dokumentā, ņemot vērā visu tekstu kopu. Augstāki TF-IDF vērtējumi tiek piešķirti terminiem, kas bieži parādās dokumentā, bet reti visā kopā. Tas palīdz uzsvērt terminu svarīgumu, kuri ir raksturīgi tieši konkrētiem dokumentiem, vienlaikus samazinot kopīgu terminu nozīmi, kas parādās daudzos dokumentos.

### 1.5.5. Vārdlietojuma kartējums

Vārdlietojuma kartējums (angliski - word embedding) ir jaunāka metode, kur vārdi tiek attēloti kā skaitliski vektori daudzdimensiju telpā. Šie vektori spēj saglabāt informāciju par vārdu kontekstu / nozīmi / saikni ar citiem vārdiem, respektīvi - vārdi ar līdzīgu nozīmi vai pielietojumu ir attēloti ar vektoriem, kas ģeometriski tuvi viens otram vektoru telpā. Piemēram, labi apmācītā vārdu iegulšanas modelī vārdi "karalis" un "karaliene" tiek attēloti kā vektori, kas ir tuvu viens otram, norādot to semantisko līdzību [9].



1.5. att. Vārdlietojuma kartējums vektora telpā [9]

Vārdlietojuma kartējums ļauj arī veikt dažādas operācijas ar vārdiem vektoru telpā, to skaitā arī saskaitīšanu un atņemšanu. Piemēram, "karalis - vīrietis + sieviete" varētu izveidot vektoru, kas vektora telpā ir tuvu vārdam "karaliene".

### 1.5.6. Pazīmju izvēle

Iepriekš tika apskatīts kā atlasīt pazīmes no dokumentu kopas. Atkarībā no apskatīto tekstu daudzuma un sarežģītības, rezultātā var tikt iegūts liels pazīmju skaits, kas var apgrūtināt mašīnmācīšanās algoritmu pielietošanu. Pārāk plaša vai pārāk maza pazīmju kopa var atstāt negatīvu iespaidu uz modeļa veiktspēju. Lai risinātu šo problēmu tiek apskatīta pazīmju izvēle.

Viena no izplatītākajām metodēm, kas samazina pazīmju skaitu ir retu vārdu izņemšana. Dēļ to retuma, tās visdrīzāk nav pazīmes, kas ir raksturīgas visiem kategoriju tekstiem, un nepalīdzēs izveidot precīzāku modeli.

## 1.6. Biežākās problēmas tekstu klasifikācijā

### 1.6.1. Vārdu neskaidrība

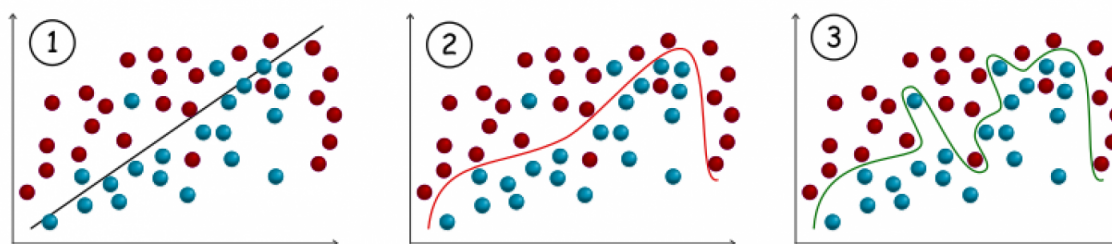
Dabiskā valoda ir būtībā neskaidra, ar vārdiem un frāzēm, kuriem ir vairākas nozīmes atkarībā no konteksta. Šīs neskaidrības precīza risināšana ir izaicinājums teksta klasifikācijas modeļiem.

### 1.6.2. Tekstu nevienmērība

Teksta dati ir dažādi un atšķiras pēc garuma, struktūras un kvalitātes. Tie var ietvert rakstos raksturīgas kļūdas, slengu, saīsinājumus un plašu rakstīšanas stilu klāstu, kas padara tos grūti standartizējamus.

### 1.6.3. Pārmērīga pielāgošana

Pārmērīga pielāgošana nozīmē to, ka klasifikators ir pārāk labi modelējis apmācības datus un nedarbojas labi uz iepriekš neredzētiem datiem. Kļūdas, ko klasifikators pieļauj uz apmācības datiem sauc par apmācības kļūdām, savukārt kļūdas, kuras tiek pieļautas uz iepriekš neredzētiem datiem, sauc par vispārināšanas kļūdām. Labam modelim ir gan zems apmācības kļūdu skaits, gan zems vispārināšanas kļūdu skaits. Nepietiekama pielāgošana notiek ja modelim ir gan augsts apmācības kļūdu skaits, gan arī augsts vispārinājuma kļūdu skaits. No otras puses - pārmērīga pielāgošana notiek kad modelim ir zems apmācības kļūdu skaits, bet augsts vispārināšanas kļūdu skaits [10]. Zemāk apskatāmajā attēlā 1.6. attēloti piemēri divdimensiju klasifikācijas scenārijā.



1.6. att. Pielāgošanas scenāriji (1 - nepietiekama, 2 - optimāla, 3 - pārmērīga)

Visbiežāk šāda veida kļūdas rodas no apmācību datiem, kuros ir pārāk daudz ar konkrēto klasifikāciju nesaistīti dati (lieks fons “troksnis”) vai arī izvēlētais apmācību datu apjoms ir pārāk mazs.

## **1.7. Mašīnmācīšanās rīki**

### **1.7.1. scikit-learn**

Viena no Python valodas populārākajām mašīnmācīšanās bibliotēkām, kas palīdz risināt problēmas kā klasteru veidošana, regresija, klasifikācija, dimensiju skaita samazināšana, ir 'scikit-learn'. Sākotnēji bibliotēku izstrādājis Dāvids Kornepū (David Cournapeau) 2007. gadā, tomēr ātri vien projekts ir izaudzis par atvērtā pirmkoda projektu kuru uztur vairāki simti izstrādātāju. Bibliotēku izmanto daudzi lieli uzņēmumi kā J.P. Morgan, Spotify u.c. Autors ir izvēlējis lietot šo bibliotēku lai atvieglotu plaši lietotu klasifikācijas algoritmu implementāciju (Naivā Bejesa metode, loģistiskā regresija, lēmumu koki, atbalsta vektoru mašīnas).

### **1.7.2. Tensorflow**

TBD

## 2. PRAKTISKĀ DAĻA

### 2.1. Datu izgūšana no ziņu portāliem ar rāpuļi

Lai veiktu izpēti, sakumā ir nepieciešams ievākt treniņdatus / valodas korpusu, kas raksturo problēmvidi – ziņu portālu rakstus. Praktiskai rāpuļa implementācijai tika izvēlēts Python ietvars “Scrapy”, ar kura palīdzību iespējams izveidot tīmekļa rāpuļus, kas pārmeklē mājaslapas un izvelk no tām datus strukturētā formā. Šis ietvars izvēlēts, jo tas ir viens no populārākajiem rīkiem šajā kategorijā un tas labi spēj apstrādāt un formatēt lielu datu apjomu. Tā kā tīmekļa rāpuļi ir jāpielāgo konkrētai mājaslapas struktūrai, lai iegūtu vēlamos datus, tika izvēlēts konkrēts portāls - delfi.lv, dēļ tā daudzveidīgā kategoriju klāsta un rakstu daudzuma. Darba ietvaros izveidots rāpuļis, kas ievāc datus no šī portāla un saglabā tos JSON formā ar 4 pamatlaukiem – virsraksts, kategorija, saturs, hipersaite. Lai sašaurinātu problēmvidi un ierobežotu nepieciešamos resursus tika izvēlētas 5 apskatāmās kategorijas – kriminālziņas, biznesa, izklaides, politikas un sporta ziņas. Izgūta raksta piemēru JSON formātā iespējams apskatīt 1. pielikumā.

Rezultātā tika ievākti 6526 raksti ar sadalījumu pa kategorijām kāds redzams tabulā 2.1.

*2.1. tabula*

**Ievākto rakstu sadalījums pa kategorijām**

Kategorija	Raksti
politika	1633
kriminālziņas	1286
izklaide	1261
sports	1180
business	1166

Ievācot rakstus novērots, ka bez rakstu satura atšķiras arī vidējie rakstu garumi katrā kategorijā. Politikas un biznesa ziņām raksturīgi gari raksti ar vidēji vairāk nekā 350 vārdiem, savukārt izklaides, sporta un kriminālziņām – krietni īsāki raksti (īpaši kriminālziņām

ar vidējo rakstu garumu ap 115 vārdiem). Šāda atšķirība garumos varētu atstāt ietekmi uz konkrētu kategoriju klasifikāciju akurātumu. Rakstu iedalījumu garumos sīkāk iespējams apskatīt 2. pielikumā.

## **2.2. Tekstu priekšapstrāde**

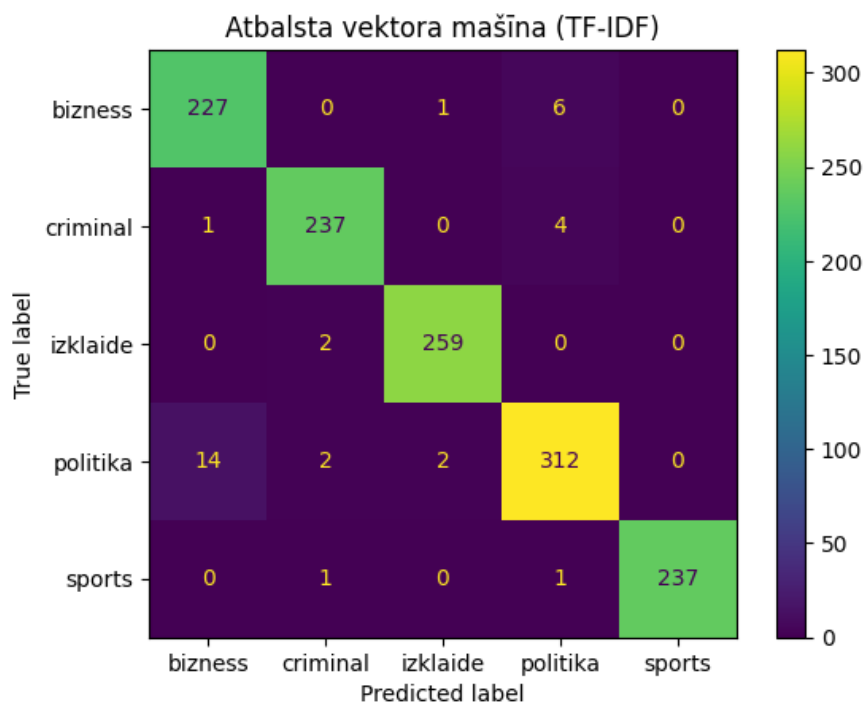
### **2.2.1. Stopvārdu atmešana**

Izmantotajās Python bibliotēkās ir iekļauti saraksti ar stopvārdiem daudzām izplatītām valodām, tomēr latviešu valodai šāds saraksts jādefinē neatkarīgi. Tika veikta izpēte par to vai šāds saraksts jau ir publiski pieejams un kā viens no populārākajiem atrasts ‘stopwords-lv’ repozitorijs iekš github. Lai gan tas ir izmantojams kā labs pamats un uzskaita palīgvārdus (saikļus, prievārdus, partikulas), trūkst citas svarīgas morfoloģiskās grupas kā vietniekvārdi (attieksmes vietniekvārdi – kurš, kura u.c., norādāmie vietniekvārdi – šis, šī, tas, tā, viņš u.c, kā arī locījumi šiem vārdiem), jo arī šo vārdu esamība neraksturo teksta fragmenta jēgu vai piederību kādai kategorijai. Darba ietvaros izveidots uzlabots stopvārdu saraksts latviešu valodai, kas labāk spētu veikt vārdu filtrēšanas soli teksta priekšapstrādē, un pielietots uz apmācības datiem.

## 2.3. Algoritmu rezultāti

### 2.3.1. Atbalsta vektora mašīnas

Atbalsta vektora mašīnas apmācības algoritms ir implementēts ar scikit-learn SVM klases komponenti LinearSVC (sklearn.svm.LinearSVC). Sasniegtais akurātums ar testa rakstu kopu un izmantojot vārdu maisa vektorizācijas pieeju - 0.952, pielietojot TF-IDF pazīmju izveidē savukārt tiek iegūts augstāks akurātums - **0.966**. Kā redzams pārpratuma matricā 2.1. - piecu kategoriju klasifikācija tiek veikta ļoti precīzi un biežākā kļūda ir nepareizi klasificētas politikas ziņas, klasificējot tās kā biznesa ziņas. Precīzāki novērtējumi pa kategorijām apskatāmi 2.2. tabulā.



2.1. att. Atbalsta vektora mašīnas pārpratuma matrica

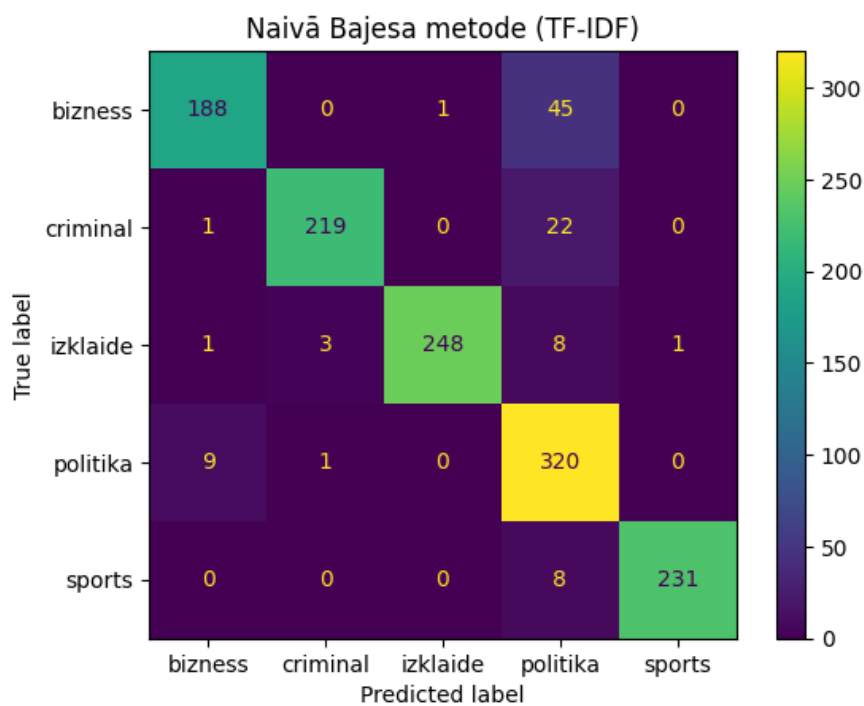


AVM algoritma novērtējums pa kategorijām

kategorija	precizitāte	pārklājums	F1 mērs
business	0.938017	0.970085	0.953782
criminal	0.979339	0.979339	0.979339
izklaide	0.98855	0.992337	0.99044
politika	0.965944	0.945455	0.95559
sports	1	0.991632	0.995798

### 2.3.2. Naivā Bajesa metode

TBD



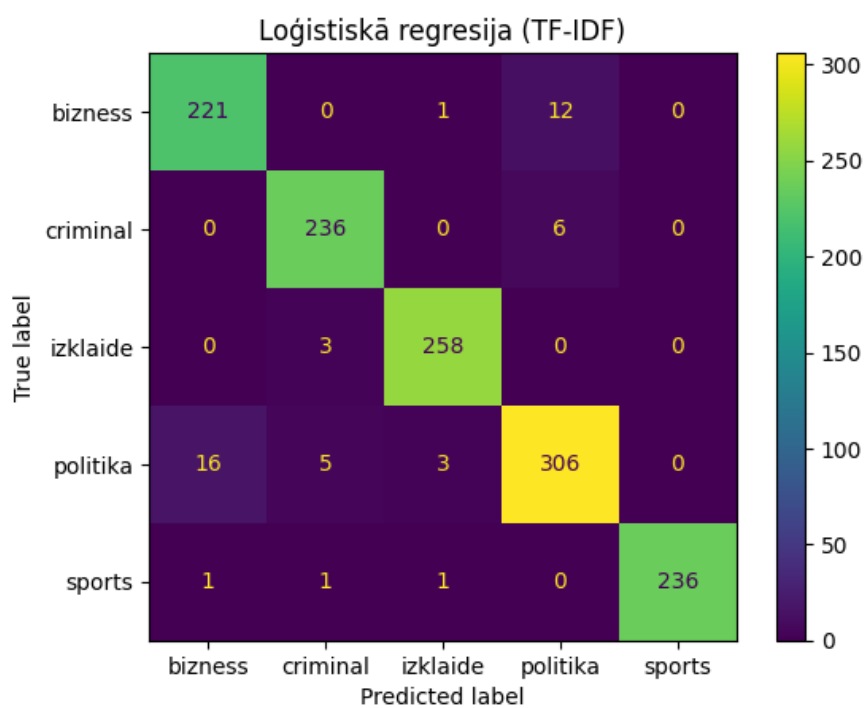
2.2. att. Naivā Bajesa metode pārpratuma matrica

Naivā Bajesa algoritma novērtējums pa kategorijām

kategorija	precizitāte	pārklājums	F1 mērs
business	0.944724	0.803419	0.86836
criminal	0.982063	0.904959	0.941935
izklaide	0.995984	0.950192	0.972549
politika	0.794045	0.969697	0.873124
sports	0.99569	0.966527	0.980892

### 2.3.3. Loģistiskā regresija

TBD



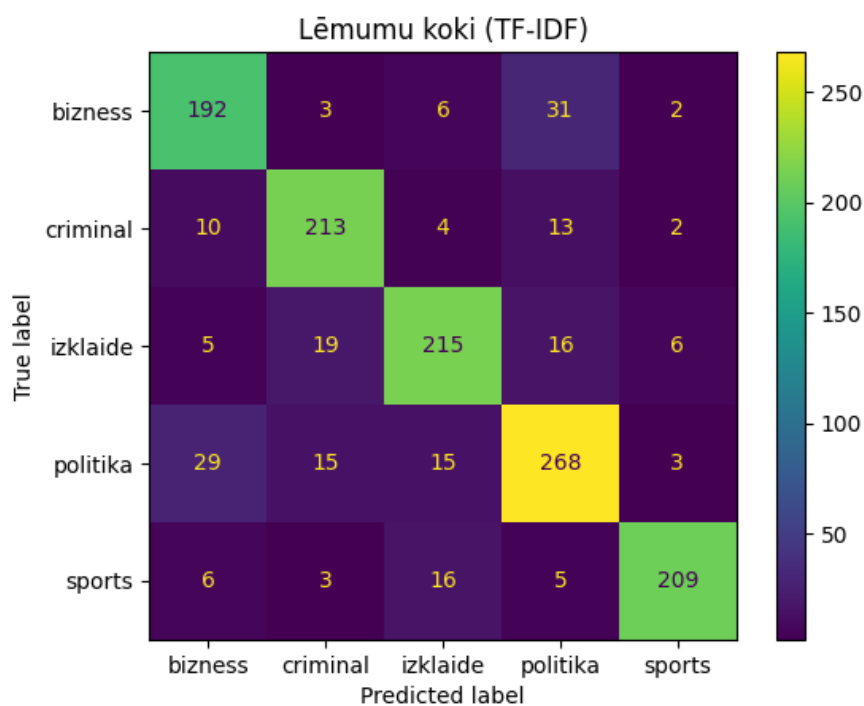
2.3. att. Loģistiskās regresijas pārpratuma matrica

**Loģistiskā regresijas algoritma novērtējums pa kategorijām**

kategorija	precizitāte	pārklājums	F1 mērs
business	0.928571	0.944444	0.936441
criminal	0.963265	0.975207	0.969199
izklaide	0.980989	0.988506	0.984733
politika	0.944444	0.927273	0.93578
sports	1	0.987448	0.993684

**2.3.4. Lēmumu koki**

TBD



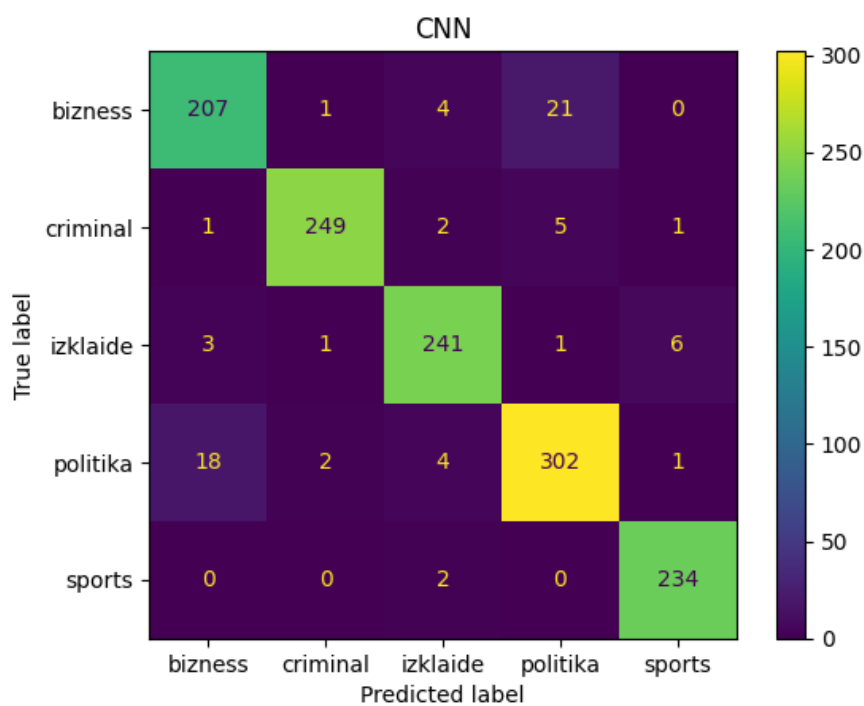
2.4. att. Lēmumu koki - pārpratuma matrica

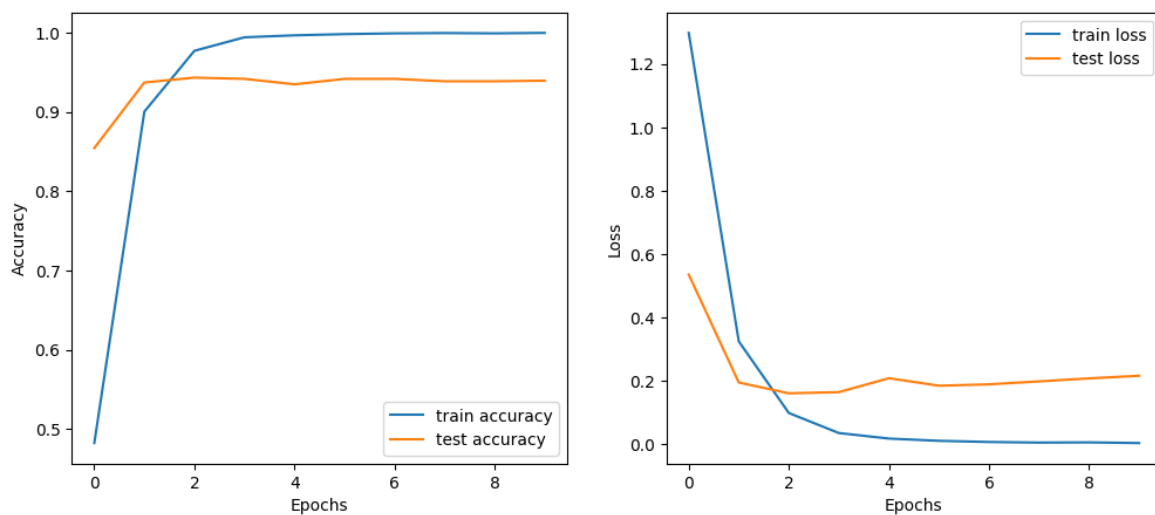
**Lēmumu koku algoritma novērtējums pa kategorijām**

kategorija	precizitāte	pārklājums	F1 mērs
business	0.840517	0.833333	0.83691
criminal	0.823529	0.867769	0.84507
izklaide	0.848249	0.835249	0.841699
politika	0.817365	0.827273	0.822289
sports	0.912281	0.870293	0.890792

### 2.3.5. Konvolūcijas neironu tīkli

TBD

**2.5. att. Konvolūcijas neironu tīkli - pārpratuma matrica**



2.6. att. Konvolūcijas neironu tīkli - novērtējums pa apmācības posmiem

2.6. tabula

Konvolūcijas neironu tīkli - novērtējums pa kategorijām

kategorija	precizitāte	pārklājums	F1 mērs
bizness	0.891775	0.88412	0.887931
criminal	0.96124	0.96124	0.96124
izklaide	0.968	0.960317	0.964143
politika	0.920245	0.917431	0.918836
sports	0.958506	0.978814	0.968553

## 2.4. Modeļu salīdzinājums

TBD

## SECINĀJUMI UN PRIEKŠLIKUMI

Darba procesā tika noskaidrots ka ziņu klasifikācijā pielietot mašīnmācīšanās algoritmus ir noderīgi, jo iespējams veikt šo klasifikāciju ļoti precīzi, augstāko akurātuma rādītāju 0.966 sasniedzot ar atbalsta vektora mašīnas algoritmu un TF-IDF pielietojumu pazīmju ģenerēšanā.

Novērots arī tas, ka ne visas kategorijas ir vienlīdz viegli klasificēt. Sporta ziņas visiem algoritmiem sanāca klasificēt daudz veiksmīgāk nekā politikas un biznesa ziņas. Tas izskaidrojams ar saturisku pārklājumu starp tēmām (politikas un biznesa ziņas bieži kļūdīgi tika klasificētas kā pretējā kategorija) un rakstu garumiem (sporta ziņas pārsvarā ir īsākas).

Lai gan izpētīt un implementēt konvolūcijas neironu tīklus autora ieskatā bija jēgpilni, izveidotais modelis nespēja sasniegt augstāku precizitāti par vienkāršākiem algoritmiem.

Autora ieskatā publiskas ziņu rakstu datu kopas ir ļoti noderīgas mašīnmācīšanās eksperimentos, piemēram, angļu valodā ziņu kopas kā “20 Newsgroup” tiek plaši pielietotas un pat iekļautas populārās bibliotēkās kā scikit-learn. Latviešu valodā šādas publiskas datu kopas netika atrastas un rakstu kopas izveide ne vienmēr ir triviāls uzdevums. Autora ievāktu datu kopu publiskojot iespējama tālāka tās pielietošana citu autoru darbos.

Teksta priekšapstrāde latviešu valodā ir ierobežota morfoloģisko rīku pieejamības dēļ. Zināmus uzlabojumus priekšapstrādē autoram ir izdevies panākt ar paplašināta stopvārdu saraksta izveidi.

### **Priekšlikumi:**

Autora ieskatā noderīgi būtu uzlabot modeļu apmācību ar papildus tekstu morfoloģisko apstrādi (piemēram, lemmatizāciju). Šāda apstrāde angļu un citu izplatītāku valodas tekstiem ir pieejama dažādās Python bibliotēkās, diemžēl latviešu valodas tekstiem nav tāda atbalsta. Nepieciešams veikt papildus darbu šādu rīku izstrādei.

Nepieciešama tālāka izpēte par neironu tīkliem un iespējām sasniegt augstāku precizitāti – iespējams autora izvēlētie slāņi un parametri tīkla izveidei nebija optimāli. Labākus rezultātus iespējams varētu sasniegt ar LSTM neironu tīkliem.

## IZMANTOTĀS LITERATŪRAS UN AVOTU SARAKSTS

- [1] Tom Mitchell. *The Discipline of Machine Learning*. Carnegie Mellon University, 2006.
- [2] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 3 edition, 2010.
- [3] Miltiadis Kandias, Vasilis Stavrou, Nick Bozovic, and Dimitris Gritzalis. Proactive insider threat detection through social media: The youtube case. pages 261 – 266, 11 2013.
- [4] Ian H.Witten, Eibe Frank, and Mark A.Hall. *Data Mining: Practical Machine Learning Tools and Techniques(Third Edition)*. Morgan Kaufmann, third edition edition, 2011.
- [5] Tom Mitchell. *Machine Learning*. McGraw - Hill Education, 1997.
- [6] Corinna Cortes and Vladimir Vapnik. Support - vector networks. *Machine Learning*, 20(3):273–297, 1995.
- [7] Daniel Jurafsky and James Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. 2008.
- [8] Prabhakar Manning, Christopher D.and Raghavan and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.
- [9] Baeldung. Dimensionality of word embeddings. <https://www.baeldung.com/cs/dimensionality-word-embeddings>, 2023. [Tiešsaistē; Skatīts Okt. 14, 2023].
- [10] Pang Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining*. Addison Wesley, 2005.

## **GALVOJUMS**

Ar šo es, Matīss Kalniņš, galvoju, ka šis bakalaura darbs ir manis paša patstāvīgi izpildīts oriģināls darbs. Visi informācijas avoti, kā arī no tiem ņemtie dati un definējumi ir norādīti darbā. Šis darbs tādā vai citādā veidā nav iesniegts nevienai citai pārbaudījumu komisijai un nav nekur publicēts.

Esmu informēts (-a), ka mans bakalaura darbs tiks ievietots un apstrādāts Vienotajā datorizētajā plaģiāta kontroles sistēmā plaģiāta kontroles nolūkos.

202\_\_gada \_\_\_\_.

Es, Matīss Kalniņš, atļauju Ventspils Augstskolai savu bakalaura darbu bez atlīdzības ievietot un uzglabāt Latvijas Nacionālās bibliotēkas pārvaldītā datortīklā Academia ([www.academia.lndb.lv](http://www.academia.lndb.lv)), kurā tie ir pieejami gan bibliotēkas lietotājiem, gan globālajā tīmeklī tādā veidā, ka ikviens tiem var piekļūt individuāli izraudzītā laikā, individuāli izraudzītā vietā.

Piekrītu \_\_\_\_\_

Nepiekrītu \_\_\_\_\_

202\_\_gada \_\_\_\_.



## PIELIKUMS

### 1. pielikums. Ar rūpuļa palīdzību izgūta raksta piemērs

```
1 [{"title": "'Rīgas Miesnieks' zīmola īpašnieks strādājis ar zaudējumiem",
2  "category": "business",
3  "body": "Gaļas pārstrādes uzņēmums AS 'HKScan Latvia' 2022. gadā apgrozījis 52,91
4  milj. EUR (+5,46% pret 2021. gadu), pārskata gadu noslēdzot ar 829,62 tūkst. EUR
  zaudējumiem, ziņo 'Lursoft' Klientu portfelis. 2021. gadā uzņēmums nopelnīja 702,54
  tūkst. EUR. Lursoft dati rāda, ka pērn gaļas pārstrādes uzņēmums nodokļu iemaksās valsts
  kopbudžetā samaksājis 7,77 milj. EUR. Uzņēmumā 2022. gadā strādāja 175 darbinieki.
  Pagājušajā gadā AS 'HKScan Latvia' savā Jelgavas ražotnē turpināja ražošanas apjomu
  palielināšanu gan vietējam patēriņam, gan eksporta tirgiem, īpaši fokusējoties uz mērķi
  palielināt pārdošanas apjomus eksporta tirgos. AS 'HKScan Latvia' galvenie eksporta
  tirgi ārpus Baltijas ir Vācija un Polija, bet produkti ar dažādiem zīmoliem tiek ražoti
  gan Igaunijas, gan Lietuvas tirgiem. Atbilstoši šim mērķim uzņēmums arī pērn plānojis
  investīcijas un veicis jaunu produktu izstrādi. Aizvadītajā gadā gaļas pārstrādes
  uzņēmums turpināja plašu investīciju programmu, lai paplašinātu saldētās produkcijas
  ražošanas cehu Jelgavas ražotnē. Investīciju plāna ietvaros tika iegādāta jauna saldētās
  produkcijas formēšanas un iepakojšanas līnija, kā arī veikta saldētavas paplašināšana.
  Uzņēmums iegādājies zemi blakus Jelgavas ražotnei aptuveni 10ha platībā, kas sniegs
  iespēju nākotnē, iespējams, attīstīt uzņēmējdarbību Jelgavā. 2022. gadā AS 'HKScan
  Latvia' sasniegta ražošanas rekordus produktu apjomu ziņā tādās kategorijās kā marinēta
  un svaiga vistas gaļa. 2022. gadā viens no visstraujāk augošajiem zīmoliem svaigās un
  marinētās gaļas segmentā bija 'Tallegg'. Aizvadītais bija zīmola 'Rīgas Miesnieks'
  100. jubilejas gads. Uzņēmums, atzīmējot šo notikumu, veica zīmola identitātes maiņu un
  izveidoja jaunu zīmola saukli 'Labs, Labāks, Labākais'. 'Zīmola identitātes maiņu
  novērtēja arī patērētāji, kā rezultātā 'Rīgas Miesnieks' zīmols bija visstraujāk
  augošais zīmols pārstrādātās gaļas kategorijā Latvijā, norādījis AS 'HKScan Latvia'.
  Pērn AS 'HKScan Latvia' pārcēla savas loģistikas funkcijas no loģistikas centra Rīgā uz
  vienoto Baltijas loģistikas centru Igaunijā, netālu no Tallinas. 'Kopējais Baltijas
  loģistikas centrs nodrošina visu Baltijas tirgu, piedāvājot klientiem ātrāku un
  elastīgāku loģistikas pakalpojumu veikšanu,' savā vadības ziņojumā uzsveris AS 'HKScan
  Latvia'. Šogad AS 'HKScan Latvia' fokusēsies uz izmaksu samazināšanu, produktivitātes
  uzlabošanu ražošanā un produktu portfeļa optimizāciju, reaģējot uz izmaiņām patērētāju
  iepirkumu un pārtikas patēriņa paradumos. 'Uzņēmums plāno, ka 2023. gadā turpināsies
  putnu gaļas un putnu gaļas produktu pārdošanas apjomu pieaugums. Pārstrādātās gaļas un
  gatavo maltiņu segmentā uzņēmums plāno koncentrēties uz produktu pievienotās vērtības
  palielināšanu. Viens no mērķiem ir gatavo maltiņu un ēdienu pieauguma apjoms,' vadības
  ziņojumā norādījis AS 'HKScan Latvia'. 'Delfi Bizness' ļauj ieskatīties uzņēmuma
  ražotnē.",
5  "link": "
  https://www.delfi.lv/bizness/biznesa_vida/rigas-miesnieks-zimola-ipasnieks-stradajis-ar-za
  udejumiem.d?id=55822846"
6  }, {
```

## 2. pielikums. Klasificējamo kategoriju rakstu garumi

