VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

FUNDAMENTINIŲ MOKSLŲ FAKULTETAS

INFORMACINIŲ TECHNOLOGIJŲ KATEDRA

Paulius Matulevičius

# NAUJIENŲ PORTALŲ ĮRAŠŲ REKOMENDAVIMAS VARTOTOJUI, TAIKANT PRITAIKOMŲJŲ PASIŪLYMŲ METODUS

# NEWS PORTAL RECORD PROPOSITION FOR USER BY APPLYING PERSONALIZED PRODUCT METHODS

Baigiamasis magistro darbas

Informacinių technologijų studijų programa, valstybinis kodas 6211BX016

Duomenų gavybos technologijų specializacija

Informatikos inžinerijos studijų kryptis

Vilnius, 2021

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
FUNDAMENTINIŲ MOKSLŲ FAKULTETAS
INFORMACINIŲ TECHNOLOGIJŲ KATEDRA

TVIRTINU
Katedros vedėjas

(Parašas)
Dmitrij Šešok
(Vardas, pavardė)
2021-05-24
(Data)

Paulius Matulevičius

# NAUJIENŲ PORTALŲ ĮRAŠŲ REKOMENDAVIMAS VARTOTOJUI, TAIKANT PRITAIKOMŲJŲ PASIŪLYMŲ METODUS

# NEWS PORTAL RECORD PROPOSITION FOR USER BY APPLYING PERSONALIZED PRODUCT METHODS

Baigiamasis magistro darbas

Informacinių technologijų studijų programa, valstybinis kodas 6211BX016
Duomenų gavybos technologijų specializacija
Informatikos inžinerijos studijų kryptis

**Vadovas** doc. dr. Simona Ramanauskaitė _____ _____
(Pedag. vardas, vardas, pavardė) (Parašas) (Data)

Vilnius, 2021

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
FUNDAMENTINIŲ MOKSLŲ FAKULTETAS
INFORMACINIŲ TECHNOLOGIJŲ KATEDRA

TVIRTINU
Katedros vedėjas

_____
(Parašas)
Dmitrij Šešok
(Vardas, pavardė)
2021-05-24
(Data)

Informatikos inžinerijos studijų kryptis

Informacinių technologijų studijų programa, valstybinis kodas 6211BX016

Duomenų gavybos technologijų specializacija

**BAIGIAMOJO MAGISTRO DARBO UŽDUOTIS**

2021-05-24 Nr. .....
Vilnius

Studentui (ei) PAULIUI MATULEVIČIUI
(Vardas, pavardė)

Baigiamojo darbo tema: NAUJIENŲ PORTALŲ ĮRAŠŲ REKOMENDAVIMAS VARTOTOJUI, TAIKANT PRITAIKOMŲJŲ PASIŪLYMŲ METODUS (*angl.* NEWS PORTAL RECORD PROPOSITION FOR USER BY APPLYING PERSONALIZED PRODUCT METHODS)

patvirtinta 2021 m. balandžio 28 d. dekano potvarkiu Nr. 96fm

Baigiamojo darbo užbaigimo terminas 2021 m. gegužės 24 d.

BAIGIAMOJO DARBO UŽDUOTIS:

Susipažinti su pritaikomųjų pasiūlymų formavimo metodais ir jų taikymo sritimis. Aprašyti natūralios kalbos apdorojimo galimybes ir teksto klasifikacijos problematikos sprendimo būdus, kurie gali būti taikomi rekomendacinėje sistemoje. Pasirinktam naujienų portalui, ar jų grupei, atrinkti (ir poreikiui esant sužymėti) tyrimui naudojamus duomenis. Su atrinktais naujienų portalo įrašais atlikti tyrimą, kuriame būtų taikomi bent du skirtingi pritaikomųjų pasiūlymų metodai, taip įvertinant jų tinkamumą, tikslumą naujienų portalų įrašų rekomendavimui. Apibendrinti tyrimo duomenis ir dokumentuoti tyrimo eigą, gautus rezultatus bei rekomendacijas praktiniam pasirinktų sprendimų taikymui.

Vadovas _____ doc. dr. Simona Ramanauskaitė
(Parašas) (Pedag. vardas, vardas, pavardė)

Užduotį gavau

_____
(Parašas)
Paulius Matulevičius
(Vardas, pavardė)
2019-10-01
(Data)

Vilniaus Gedimino technikos universitetas

**Fundamentinių mokslų** fakultetas

**Informacinių technologijų** katedra

ISBN ISSN

Egz. sk. .....

Data .....-....-....

**Informacinių technologijų** studijų programos baigiamasis magistro darbas

Pavadinimas: **Naujienų portalų įrašų rekomendavimas vartotojui, taikant pritaikomųjų pasiūlymų metodus**

Autorius: **Paulius Matulevičius**

Vadovė: **doc. dr. Simona Ramanauskaitė**

**Kalba**: lietuvių

**Anotacija**

Baigiamojo darbo tikslas – taikant pritaikomųjų pasiūlymų formavimo metodus ir surinktus naujienų duomenis ištirti vartotoją dominančių naujienų atranką ir rekomendacijų galimybes. Teorinėje dalyje išnagrinėtos rekomendacinės sistemos ir pritaikomųjų pasiūlymų metodai, apžvelgta natūralios kalbos apdorojimo nauda ir atlikta jos problematikos analizė. Pagal teorines gaires sudarytas ir aprašytas tyrimo metodikos planas. Praktinėje dalyje atliktas savarankiškai surinkto duomenų rinkinio klasterizavimas ir pateikti statistiniai jo rezultatai. Pasirinktais metodais apdorotam rinkiniui įgyvendintas teksto klasifikacijos uždavinys, pateiktos rezultatų ir bendrosios tyrimo išvados.

Darbą sudaro 8 dalys: įvadas, pritaikomųjų pasiūlymų formavimo apžvalga, natūralios kalbos apdorojimo ir problematikos analizė, tyrimo metodikos planas, klasterizavimo proceso aprašymas ir statistiniai rezultatai, klasifikacijos užduoties realizacija ir rezultatų palyginimas, išvados, naudotos literatūros sąrašas.

Darbo apimtis – 60 puslapių, 31 iliustracija, 13 lentelių ir 26 bibliografijos šaltiniai.

**Prasminiai žodžiai**: rekomendacinės sistemos, NLP, klasterizavimas, teksto klasifikacija

Vilniaus Gediminas Technical University

Faculty of Fundamental Sciences

Department of Information Technologies

ISBN ISSN

Copies No. .....

Date .....-….-....

Master Degree Studies **Information Technologies** study programme Master Graduation Thesis

Title: **News Portal Record Proposition for User by Applying Personalized Product Methods**

Author: **Paulius Matulevičius**

Academic supervisor: **doc. dr. Simona Ramanauskaitė**

**Language**: Lithuanian

**Annotation**

Purpose of master‘s thesis: to research the selection of news of consumer interest and possibilities for recommendations by applying personalized product methods and gathered data of news records. In the theoretical part, recommendation systems and methods of personalized proposals were examined. Also, reviewed the benefits of natural language processing and its problems. A plan of research methodology was developed and widely described according to the theoretical guidlines. In the practical part, the self-collected data set was clustered and its statistical results were described. The text classification task was implemented by using selected methods with results and general conclusions of the research presented after.

Whole paper consists of 8 parts: introduction, overview of personalized proposals formation, natural language processing and problem analysis, detailed research methodology plan, clustering process description and statistical results, classification task realization and comparison of results, conclusion, list of used literature references.

Thesis consist of total: 60 pages, 31 figures, 13 tables and 26 bibliographical entries.

**Keywords**: recommendation systems, NLP, clustering, text classification

## Turinys

## Paveikslų sąrašas

## Lentelių sąrašas

## Santrumpų sąrašas

| | |
|---|---|
| **IT** | Informacinės technologijos (angl. *Information Technology*) |
| **NLP** | Natūralios kalbos apdorojimas (angl. *Natural Language Processing*) |
| **NLU** | Natūralios kalbos supratimas (angl. *Natural Language Understanding*) |
| **NLG** | Natūralios kalbos generavimas (angl. *Natural Language Generation*) |
| **ML** | Mašininis mokymas (angl. *Machine Learning*) |
| **UML** | Unifikuota modeliavimo kalba (angl. *Unified Modeling Language*) |
| **HTML** | Hiperteksto žymėjimo kalba (angl. *Hyper Text Markup Language*) |
| **IDE** | Integruota kūrimo aplinka (angl. *Integrated Development Environment*) |
| **GUI** | Grafinė naudotojo sąsaja (angl. *Graphical User Interface*) |
| **ROC** | Sprendimus priimančiojo ypatybių kreivė (angl. *Receiver Operating Characteristic*) |
| **AUC** | Plotas po ROC kreive (angl. *Area Under the ROC Curve*) |

# Įvadas

Šiais skaitmeninių technologijų laikais žmonės vis daugiau savo verslo ir laisvalaikio veiklių perkelia į interneto erdves. Duomenų kiekiai internete nuolatos auga, o jo naudotojams tampa vis sunkiau sekti jiems reikšmingą ir aktualią informaciją. Kad pasiektų reikalingą informaciją, naudotojai turi tikrinti arba apeiti didelius kiekius jiems neįdomios informacijos.

Duomenų pertekliui spręsti naudojami pritaikomieji metodai, kurie kiekvienam naudotojui pateikia tik jiems individualiai pritaikytus pasiūlymus ir rekomendacijas. Tai principas, kurį naudoja praktiškai visos didžiosios naujienų, prekybos ir kitos informacinių technologijų įmonės pasaulyje.

Šio baigiamojo darbo tema orientuojasi į naujienų portalų įrašų pritaikomuosius pasiūlymus straipsnių skaitytojams. Todėl didelio naujienų įrašų kiekio problemai spręsti pritaikomas duomenų gavybos principas, kuris, taikant automatinio apsimokymo programinę įrangą, sugebės pateikti skaitytojams tik jiems aktualias ir pagal jų skaitytų straipsnių patirtį pritaikytas naujienų įrašų rekomendacijas.

**Tyrimo objektas.** Pritaikomųjų pasiūlymų ir rekomendacijų formavimas.

**Darbo tikslas.** Taikant pritaikomųjų pasiūlymų formavimo metodus ir surinktus naujienų duomenis ištirti vartotoją dominančių naujienų atranką ir rekomendacijų galimybes.

**Uždaviniai:**

1. Susipažinti su rekomendacinių sistemų paskirtimi ir pritaikomųjų pasiūlymų formavimo metodais.
2. Aprašyti natūralios kalbos apdorojimo galimybes ir teksto klasifikacijos problematikos sprendimo būdus.
3. Pateikti dokumentuotą tyrimo eigos planą lietuviškų tekstų klasifikacijos uždaviniui išspręsti.
4. Atlikti paruošto duomenų rinkinio klasterizavimo veiksmus ir išanalizuoti statistines rezultatų savybes.
5. Atlikti klasifikacijos uždavinį su keliais pasirinktais klasifikatoriais, palyginti gautus rezultatus ir pateikti išvadas apie lietuviškų tekstų klasifikacijos galimybes.

**Laukiami darbo rezultatai ir nauda.** Šio rezultatai leidžia įvertinti duomenų gavybos ir pritaikomųjų metodų taikymo naudą, siekiant pateikti kuo tikslesnius ir individualiai pritaikytus pasiūlymus naujienų portalų skaitytojams. Pritaikomi pasiūlymai moderniame internete taupo naudotojų laiką, dėmesį ir resursus, kurie būtų skirti savarankiškam aktualios informacijos ieškojimui.

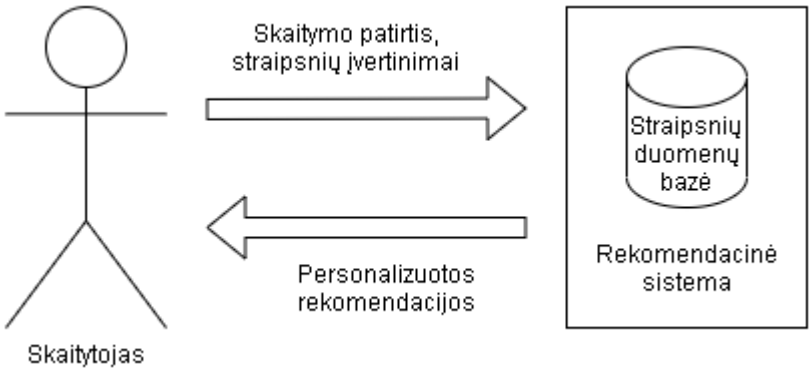
# 1. Pritaikomųjų pasiūlymų formavimas

Pritaikomųjų pasiūlymų formavimas yra informacijos apdorojimo kryptis, kuri stengiasi tiksliai prognozuoti savo vartotojų pomėgius ir pasirinkimus (Ricci, Rokach, Shapira ir Kantor, 2011). Kiekvienoje rekomendacinėje sistemoje dalyvauja vartotojo ir produkto subjektai. Vartotoju laikomas subjektas naudoja sistemą ir suteikia jai duomenis apie savo nuomonę ir įvertinimus, o atgal gauna rekomendacijas.

## 1.1. Rekomendacinės sistemos

Pritaikomosios rekomendacinės sistemos nuo nepritaikomųjų sistemų skiriasi tuo, jog pritaikomosios rekomendacinės sistemos fiksuoja kiekvieno naudotojo individualią veiksmų istoriją.

Rekomendacinės sistemos turi tris etapus: įvesties, pritaikomųjų pasiūlymų formavimo ir išvesties (Vozalis ir Margaritis, 2003). 1.1 pav. vaizduojamas ryšys tarp rekomendacinės sistemos ir jos naudotojo (straipsnių skaitytojo). Skaitytojas pateikia informaciją, kokius straipsnius skaito, kiek laiko juos skaito ir kurie straipsniai skaitytojui patiko ar nepatiko, o rekomendacinė sistema gali išanalizuoti gautus individualius duomenis ir pritaikyti juos kitų straipsnių rekomendacijoms. Kiekvieno skaitytojo individualios informacinės įvestys suteikia galimybę sistemai grąžinti visiškai pritaikytus rezultatus. Rekomendacinių sistemų efektyvumą geriausiai įvertina tikslių ir reprezentatyvių rekomendacijų pateikimas, sugebant išskirti tinkamus duomenis iš didelės apimties duomenų.




***1.1 pav.*** *Rekomendacinė sistema*

Pagrindinės ir esminės duomenų gavybos ir rekomendacinių sistemų užduotys yra klasterizavimas (angl. *clustering*) ir klasifikacija (angl. *classification*). Klasifikacija dažniausiai yra prižiūrimo mokymo metodas, o klasterizavimas – neprižiūrimo mokymo metodas (Rokach ir Maimon, 2005). Šie du metodai yra taikomi praktinei darbo daliai įgyvendinti, siekiant suformuoti pritaikomųjų naujienų pasiūlymus.

## 1.2. **Mašininis mokymas**

Mašininis mokymas (angl. *machine learning*) yra mokslo sritis, kuri nagrinėja kompiuterių apmokymą dirbti su duomenų šablonais ir atskirti ryšius tarp jų objektų. Mašininis mokymas skiriasi nuo klasikinio programavimo savo įvestimi ir išvestimi. Įprastos programos apdoroja pateiktus duomenis pagal aprašytas taisykles ir pateikia rezultatus, o mašininio mokymo sistemos įvesties duomenis panaudoja taisyklėms generuoti, tolesniam naujų duomenų apdorojimui.


***1.2 pav.*** *Mašininio mokymo tipai*

Pagal žmogaus įsikišimo kiekį, mašininis mokymas skirstomas į tris pagrindinius tipus (Stinis, 2019):

1. Prižiūrimas mokymas (angl. *supervised learning*). Tai pats elementariausias mokymo tipas, kuriam pateikiami įvesties duomenys ir juos atitinkantys išvesties duomenys. Mokymo užduotis yra rasti ryšį tarp įvesties ir išvesties duomenų bei išmokti bendras taisykles.

2. Neprižiūrimas mokymas (angl. *unsupervised learning*). Tai mokymo tipas, kuriam nėra pateikiamos duomenų išvestys. Neprižiūrimo mokymo algoritmai patys analizuoja įvesties duomenis, ieško šablonų ir išveda nuspėjamus rezultatus. Priešingai, nei prižiūrimo mokymo metu, negalima iš anksto įvertinti modelio tikslumo.

3. Sustiprintas mokymas (angl. *reinforcement learning*). Mašininio mokymo tipas, kuomet apsimokanti programa savarankiškai mokosi nenuspėjamoje ir sudėtingoje aplinkoje. Šio tipo algoritmai dinaminėje aplinkoje turi pasiekti tikslą be žmonių įsikišimo.

## 1.3. **Klasterizavimo tipai**

Klasterizavimas (angl. *clustering*) yra duomenų gavybos sritis, nagrinėjama duomenų gavybos, mašininio mokymo ir duomenų bazių bendruomenėse, ir apimanti metodus, kurie sugeba sugrupuoti duomenų įrašus į skirtingas grupes (Pujari, 2013). Todėl galima apibrėžti, kad

klasterizacijos metodų tikslas yra sugebėti surasti panašumus tarp informacinių vienetų ir juos sugrupuoti į klasterius (angl. *cluster*). Kiekviename klasteryje esantys duomenų vienetai yra panašūs į kitus tos pačios grupės įrašus, tačiau identifikuotais bruožais skiriasi nuo kitų klasterių įrašų.

Galima išskirti kelis dažniausius duomenų klasterizacijos tikslus:

1. Nustatyti panašumo laipsnį tarp duomenų objektų.
2. Išskaidyti duomenų rinkinį į atskiras grupes, siekiant dirbti su individualiomis grupėmis atskirai.
3. Identifikuoti duomenų rinkinio anomalijas.
4. Valdyti saugojamų duomenų kiekius, paliekant tik pasirinktus klasterių objektus.
5. Atrinkti tinkamiausius požymius, pasiliekant tik tinkamiausius požymius iš jų klasterių.

Klasterių sudarymas yra neprižiūrimo mokymo užduotis, kai siekiama nustatyti baigtinį kiekį klasterių, apibūdinančių duomenis. Kitaip nei klasifikavimas, kuris analizuoja klasių žymes, klasterizavimas neturi apsimokymo etapo ir dažniausiai yra naudojamas, kai klasės net nėra iš anksto žinomos. Šio metodo metu yra nustatoma duomenų panašumo metrika, pagal kurią panašūs duomenų įrašai yra padalinami į klasterius. Geriausi rezultatai yra gaunami tuomet, kai iš anksto nustatomi grupes identifikuojantys duomenų atributai. Duomenų grupavimas yra paremtas maksimaliu panašumu tarp klasterio narių ir minimaliu panašumu tarp skirtingų klasterių. Vėliau, klasterių profilių analizė padeda pamatyti, kokie yra esminiai klasterių duomenų skirtumai. Aukštos kokybės klasterizavimas yra atliktas tada, kai viename klasteryje esantys įrašai yra labai panašūs, tačiau smarkiai skiriasi nuo kitų klasterių įrašų (Han, Kamber ir Pei, 2011).

Klasterizavimo algoritmus nagrinėjančios literatūros šaltiniai suskirsto juos į 4 kategorijas:

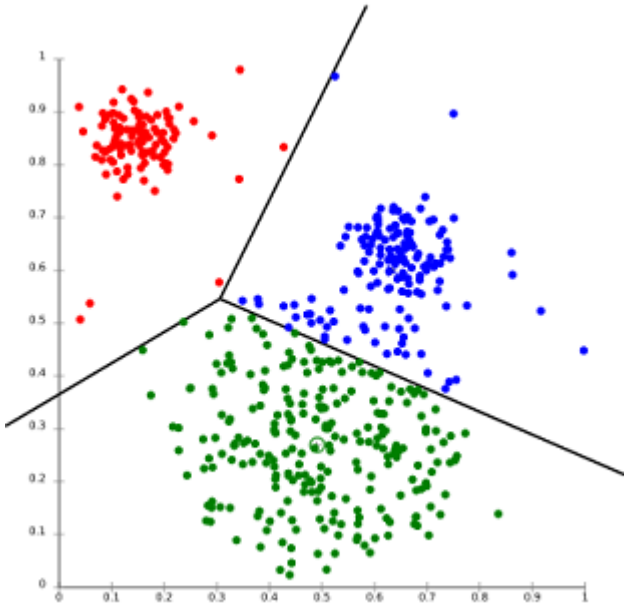
1. Suskaidymo klasterizavimo algoritmai, kurių veikimas priklauso nuo atstumais tarp duomenų paremtų funkcijų.
2. Hierarchinio klasterizavimo algoritmai, kurie sukuria hierarchinę duomenų rinkinio dekompoziciją, atvaizduojamą su dendrograma.
3. Tankiu paremto klasterizavimo algoritmai, kurie ieško tankiai užimtų regionų duomenų erdvėje.
4. Tinklu paremto klasterizavimo algoritmai, kurie identifikuoja labiausiai užimtas tinklelio struktūros ląsteles (angl. *cells*).

### 1.3.1. *Suskaidymo klasterizavimas*

Suskaidymo klasterizavimo (angl. *partitioning clustering*) algoritmai bando išskaidyti objektus į pasirinktą klasterių kiekį taip, kad būtų optimizuota klasterizavimo funkcija. Funkciją sudaro duomenų rinkinys *D*, objektų kiekis *N* ir norimas klasterių kiekis *K*, kurie klasterizavimo pabaigoje pasidalina į *K* skaičių atitinkantį grupių kiekį (žr. 1.3 pav.). Kiekviena sugeneruota grupė atitinka atskirą klasterį. Klasteriai yra sudaryti optimizavus atstumus tarp objektų – objektai viename klasteryje yra arti vienas kito, tačiau toli nuo kitų klasterių objektų.

Tikriausiai pats populiariausias metrinių erdvių klasterizavimo metodas yra vadinamas k-vidurkių metodu (angl. *k-means*). K-vidurkių metodo žingsniai (Sisodia, Singh ir Saxena, 2012):

1. Pasirenkamas norimas klasterių kiekis (*k*).
2. Atsitiktinai generuojami klasteriai ir nustatomi klasterių centrai.
3. Erdvės taškai priskiriami prie artimiausių klasterių centrų.
4. Perskaičiuojami nauji klasterių centrai.
5. Nuolatos kartojami paskutiniai du žingsniai, kol pasiekiamas reikalingas rezultatas ir nepasikeičia klasterių centrai.



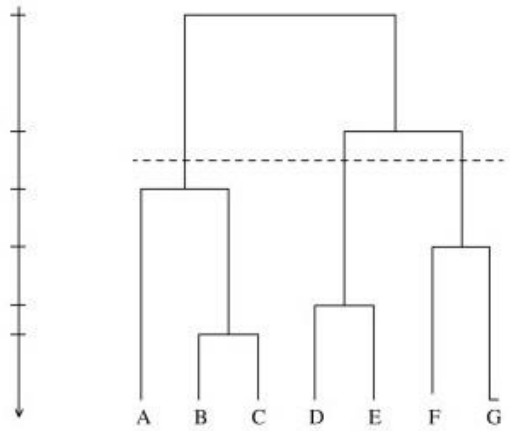

***1.3 pav.*** *Suskaidymo klasterizavimas*

### 1.3.2. *Hierarchinis klasterizavimas*

Hierarchinis klasterizavimas (angl. *hierarchical clustering*), kitaip nei suskaidymo klasterizavimo metodai, sugeba sugeneruoti klasterių sekas, atvaizduojamas dendrograma (žr. 1.4 pav.). Dendrogramos viršūnėje yra atvaizduojamas visus apimantis klasteris su smulkesniais klasteriais žemesniuose diagramos lygiuose (Karypis, Han ir Kumar, 1999). Diagramos hierarchija

gali kylanti aukštyn arba besileidžianti žemyn. Pasiekus norimą klasterių kiekį, hierarchinio klasterizavimo metodas gali būti sustabdytas. Dažniausiai kiekviena algoritmo iteracija vykdo klasterių porų jungimą ir atskyrimą, pagrįstą atstumais tarp klasterių. Hierarchinio klasterizavimo trūkumas - anksčiau atlikti sujungimo ir suskaidymo veiksmai gali būti klaidingi, tačiau nebegali būti sugrąžinti atgal.

Dažnai naudojami hierarchinio klasterizavimo metodai: CURE, CHAMELEON ir BIRCH.

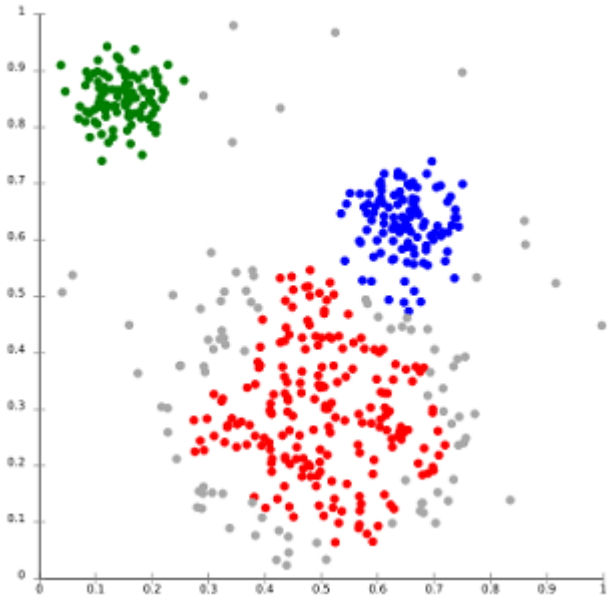



***1.4 pav.*** *Hierarchinio klasterizavimo dendrograma*

### 1.3.3. *Tankiu paremtas klasterizavimas*

Tankiu paremto klasterizavimo (angl. *density-based clustering*) metodai sugrupuoja objektus į klasterius atsižvelgiant į objektų populiacijos tankius, o ne atskirus atstumus tarp objektų (Nasibov ir Ulutagay, 2009). Šie metodai klasteriais laiko tankius erdvės regionus, aplink kuriuos nėra tankiai išsidėsčiusių objektų (žr. 1.5 pav.). Didžiausias šio metodo pranašumas yra tas, jog tankiu paremtas klasterizavimas gali identifikuoti įvairių formų klasterius.

Dažniausiai naudojami tankiu paremto klasterizavimo algoritmai yra DBSCAN, OPTICS ir DENCLUE.



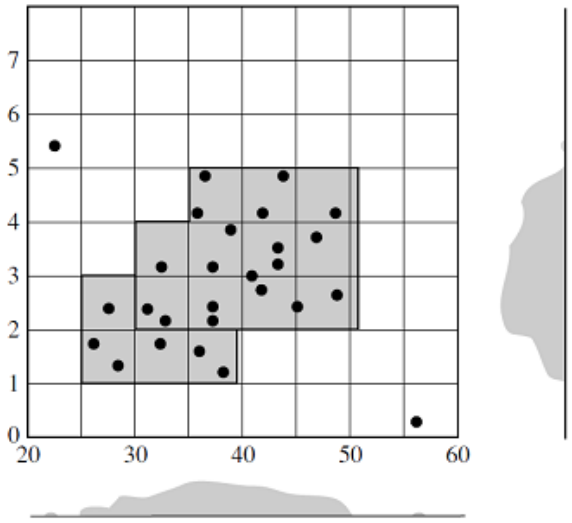

***1.5 pav.*** *Tankiu paremtas klasterizavimas*

### 1.3.4. *Tinklu paremtas klasterizavimas*

Prieš tai apžvelgti klasterizavimo metodai yra atsižvelgiantys į turimus duomenis – jie padalina objektus į atskiras dalis ir sukuria duomenis atitinkančią erdvę. Tuo tarpu tinklu paremtas klasterizavimas (angl. *grid-based clustering*) yra paremtas vietos pasidalinimu – padalina erdvę į ląsteles (angl. *cells*), nepriklausomai nuo duomenų objektų pasiskirstymo.

Tinklu paremto klasterizavimo būdas naudoja kelių sluoksnių tinklinę duomenų struktūrą, taip padalindamas objektų erdvę į baigtinį ląstelių kiekį (žr. 1.6 pav.). Šiose ląstelėse yra atliekamos visos metodo klasterizavimo operacijos. Pagrindinis tokio metodo pranašumas yra labai greitas apdorojimo laikas, kuris dažniausiai yra visiškai nepriklausomas nuo duomenų objektų kiekio. Skaičiavimų greitis priklauso tik nuo ląstelių kiekvienoje tinklo dimensijoje kiekio.

Egzistuoja keli tinklu paremto klasterizavimo būdai. Algoritmas STING analizuoja statistinę ląstelių informaciją, o algoritmas CLIQUE atsižvelgia į kiekvienos ląstelės tankio informaciją.




***1.6 pav.*** *Tinklu paremtas klasterizavimas*

Klasterizavimo algoritmai taip pat yra skirstomi į dvi rūšis: algoritmai, kurie skirsto duomenis į nustatytą klasterių kiekį (pavyzdžiui, k-vidurkių algoritmas), arba algoritmai, kurių įvestis neapibrėžia klasterių kiekio, o jis yra nustatymas algoritmo vykdymo metu (pavyzdžiui, *Forel* algoritmas).

### 1.4. **Klasifikacijos metodai**

Klasifikavimas yra klasikinė duomenų gavybos užduotis, dažnai naudojama mašininio mokymo darbuose (Sundar, Latha ir Chandra, 2012). Tai duomenų analizės forma, kuri išgauna modelius, apibūdinančius duomenų klases. Šie modeliai, dar vadinami klasifikatoriais, nuspėja kategorinių, pažymėtų klasių reikšmes. Pavyzdžiui, galima sukurti tokį klasifikavimo modelį, kuris sugebėtų kategorizuoti naujienų straipsnius į rekomenduojamus ir nerekomenduojamus. Klasifikavimo taikymas padeda atlikti daug išsamesnę duomenų analizę.

Teksto klasifikacijai yra taikomi keli skirtingi metodai, kurie pasirenkami atsižvelgiant į metodų privalumus. Pagrindiniai klasifikavimo metodų aspektai yra veikimo greitis, skaičiavimų paprastumas ir gaunamų rezultatų tikslumas. Klasifikacijai *RapidMiner Studio* sistemoje dažniausiai naudojami metodai: *Naive Bayes, Generalized Linear Model, Logistic Regression, Fast Large Margin, Deep Learning, Decision Tree, Random Forest, Gradient Boosted Trees* ir *Support Vector Machine*.

Dažniausiai naudojami teksto klasifikavimo metodai:

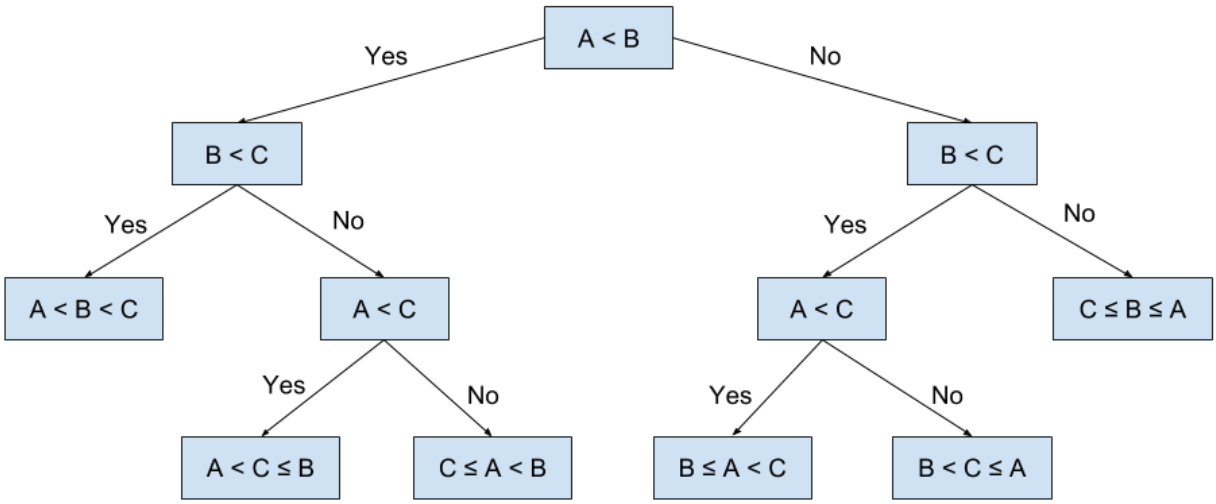
1. Naivusis Bajeso (angl. *Naive Bayes*) klasifikavimo metodas yra vienas iš dažniausiai naudojamų ir paprastų metodų. Jis labai supaprastina mokymąsi priimdamas prielaidą, kad ypatybės yra nepriklausomos nuo nurodytos klasės. Nors tokia nepriklausomybės prielaida yra prastas būdas klasifikuoti, tačiau Naivusis Bejeso metodas dažnai

nenusileidžia sudėtingesniems klasifikatoriams (Rish, 2001). Didelis Naiviojo Bejeso klasifikatoriaus privalumas yra, kad norint įvertinti klasifikacijos parametrus, reikalingas nedidelis kiekis apmokymo duomenų.

2. Bendro linijinio modelio (angl. *Generalized Linear Model*) metodai yra tradicinių tiesinių modelių pratęsimas. Šis algoritmas duomenims pritaiko apibendrintus tiesinius modelius, maksimaliai padidindamas logaritminį panašumą. Modelio tinkamumas yra apskaičiuojamas paraleliai ir ypač greitai.

3. Greitasis didelės ribos (angl. *Fast Large Margin*) metodas yra pagrįstas tiesinio atraminių vektorių klasifikatoriaus schema, kurią pasiūlė Fan, R. E., Chang, K. W., Hsieh, C. J., Wang, X. R. ir Lin, C. J. Nors greitojo didelės ribos klasifikatoriaus rezultatai yra panašūs į klasikinio atraminių vektorių klasifikatoriaus ir logistinės regresijos rezultatus, šis klasifikatorius sugeba dirbti su duomenų rinkiniu, kuriame pateikiami milijonai pavyzdžių ir atributų.

4. Sprendimo medžiai (angl. *Decision Trees*) yra medžio pavidalą turintis klasifikatorius. Metodas klasėms priskiria daugiamačius vektorius, kurie nebūtinai yra kategoriniai. Medžius sudaro šakos (pastabos apie objektą) ir lapai (išvados apie elementus). 1.7 pav. pavyzdyje matoma, kaip mazgai yra apibrėžiami testo pavidalu, o po jais esantys pomedžiai atstovauja visas testų galimas baigtis.

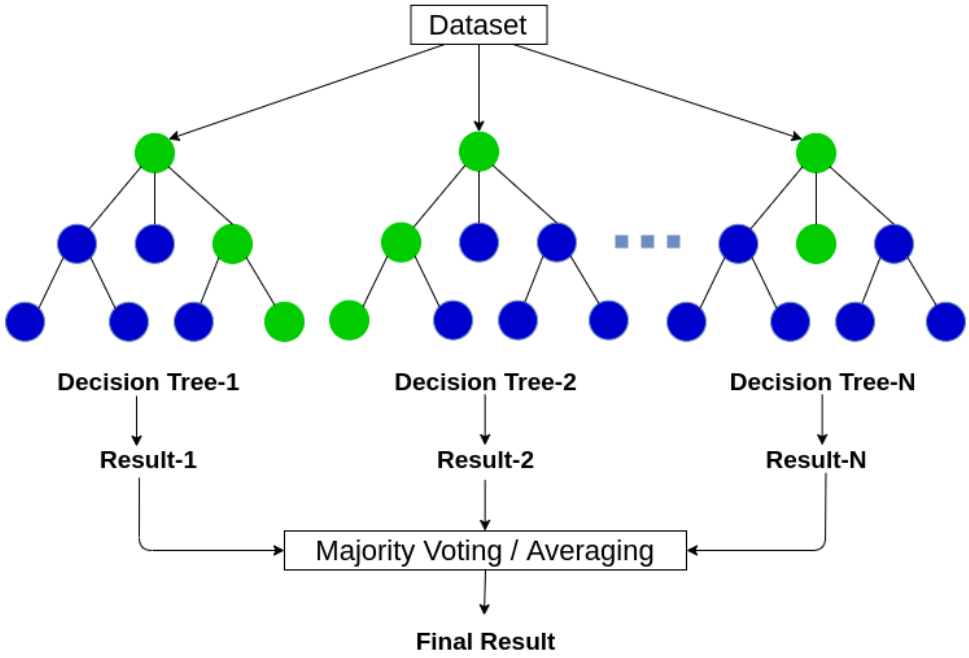



***1.7 pav.*** *Sprendimo medžių pavyzdys*

Klasifikuojančiais medžiais vadinami tie sprendimo medžiai, kuriuose priklausomas kintamasis gali gauti tik vieną reikšmę iš baigtinės reikšmių aibės. Klasifikuojančiuose medžiuose šakos simbolizuoja sutampančias ypatybes, kurios baigiasi lapais, simbolizuojančiomis klases. Regresijos medžiais vadinami sprendimų medžiai,

kuriuose priklausomas kintamasis gali įgauti tik nuolatines reikšmes (dažniausiai skaičius).

5. Atsitiktinių medžių rinkinys (angl. *Random Forest*) – taip pat paremtas sprendimo medžiu modelis. Klasifikatorius veikia apmokydamas skirtingus ir atsitiktinius požymių poaibius ir sugeneruodamas atskirus medžius (žr. 1.8 pav.). Galutiniai rezultatai yra priimami atsižvelgus į visų medžių rezultatus. Atsitiktinių medžių rinkinio klasifikatoriaus rezultatai yra pasiekiami dviem būdais – skaičiuojamas visų medžių rezultatų vidurkis (angl. *averaging*) arba tikrinama, kuris medis surinko didžiausią kiekį balsų (angl. *majority voting*).




***1.8 pav.*** *Palaipsniui išpūstų medžių pavyzdys*

6. Palaipsniui išpūstų medžių (angl. *Gradient Boosted Trees*) yra regresijos arba klasifikavimo medžio modelių ansamblis. Abu šie metodai yra apmokomi į priekį, o prognozuojami rezultatai gaunami palaipsniui tobulinant vertinimus. Taikoma ir lanksti netiesinė regresijos procedūra, padedanti pagerinti medžių tikslumą. Nuosekliai taikant silpnus klasifikavimo algoritmus palaipsniui besikeičiantiems duomenims, sukuriama daugybė sprendimų medžių, kurie sukuria daugybę prastų prognozavimo modelių ansamblį. Nors medžių tikslumas didėja, tačiau lėtėja greitis ir mažėja suprantamumas. Palaipsniui išpūstų medžių metodas apibendrina medžius ir siekia sumažinti minėtas problemas.

7. Atraminių vektorių mašinos (angl. *Support Vector Machine*) klasifikatorius yra skirtas klasifikuoti jau sužymėtus iš anksto žinomų klasių duomenis. Standartinis SVM metodas ima įvesties duomenų rinkinį ir kiekvienai nurodytai įvesčiai numato, kurią

iš dviejų galimų klasių sudaro įvestis, todėl SVM yra netikimybinis dvejetainis tiesinis klasifikatorius. Pateikiant mokymo pavyzdžių rinkinį, kur kiekvienas pavyzdys priskirtas vienai iš dviejų kategorijų, SVM apmokymo algoritmas sukuria modelį, kuris priskiria naujus pavyzdžius kažkuriai vienai kategorijai. SVM modelis yra reprezentacija, kurioje pavyzdžiai yra tarsi erdvėje esantys taškai, kurie išsidėstę erdvėje taip, kad vienos kategorijos taškai yra atskirti nuo kitos kategorijos taškų kaip įmanoma akivaizdžiau. Tuomet nauji pavyzdžiai yra priskiriami tai kategorijai, kuri patenka į tą pačią erdvės vietą.

# 2. Natūralios kalbos apdorojimas

Natūralios kalbos apdorojimas (angl. *natural language processing*) tiria kompiuterių panaudojimą, siekiant apdoroti ir suprasti natūralią žmonių kalbą, kad būtų galima atlikti naudingas užduotis (Deng ir Liu, 2018). Natūrali kalba, kurią NLP analizuoja lingvistiniais metodais, gali būti rašytinė arba tariamoji.

NLP informaciniuose šaltiniuose yra skaidomas į dvi pagrindines dalis (Khurana, Koli, Khatter ir Singh, 2017): natūralios kalbos supratimą (angl. *natural language understanding*) ir natūralios kalbos generavimą (angl. *natural language generation*). NLU tiria žmonių kalbos fonologiją, sintaksę, morfologiją, semantiką ir pragmatiką. NLG tiria natūralios kalbos suvokimą ir jos generavimą.

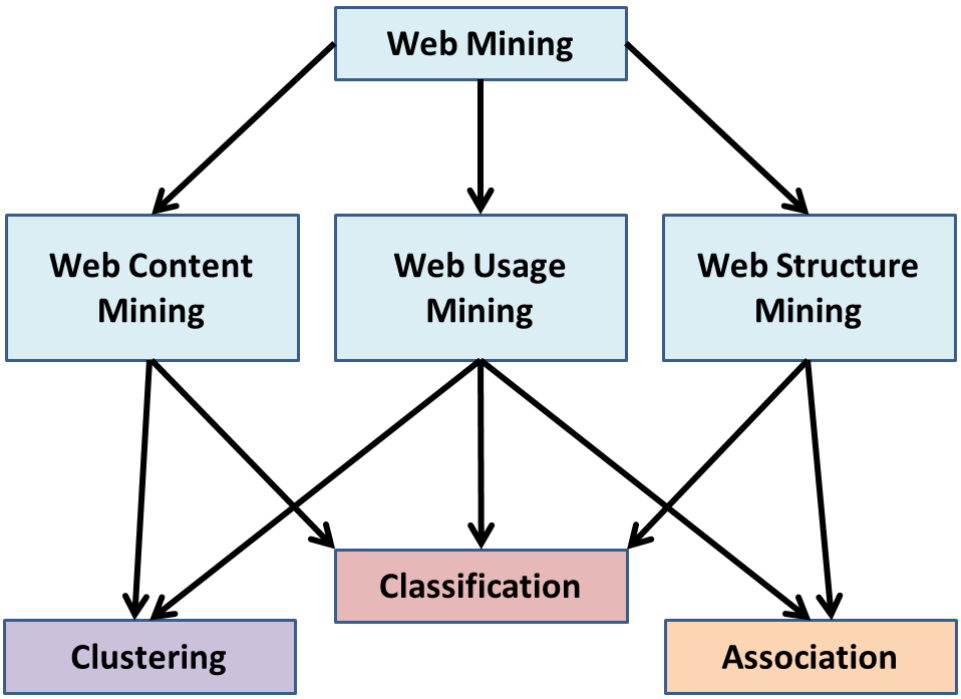
Dažnai įvardijama, jog pagrindinis NLP tikslas yra natūralios kalbos supratimas (angl. *natural language understanding*), kurį galima dalinti į keturis smulkesnius tikslus (Liddy, E. D., 2001):

1. Įvesties teksto perfrazavimas.
2. Teksto išvertimas į kitas kalbas.
3. Sistemos atsakinėjimas į klausimus, susijusius su teksto turiniu.
4. Išvadų apie tekstą ir jo turinį formavimas.

## 2.1. Interneto duomenų gavyba

Duomenų gavyba – tai duomenų apdorojimas naudojant sudėtingas duomenų paieškos galimybes ir statistinius algoritmus pasikartojantiems šablonams bei koreliacijoms rasti didelėse duomenų bazėse. Duomenų gavyba apibūdinama kaip naujų prasmių duomenyse aptikimo, nustatymo ir atradimo būdas. Kai duomenų gavybos mokslas pritaikomas internetui ir žiniatinklyje prieinamai informacijai, tai vadinama interneto duomenų gavyba (Srinath, 2017).

Interneto duomenų gavyba yra suskirstyta į tris pagrindines kategorijas (žr. 2.1 pav.), priklausomai nuo išgaunamų duomenų tipo (Sharma, Arvind ir Gupta, 2012). Tai yra interneto turinio duomenų gavyba, interneto struktūros duomenų gavyba ir interneto naudojamumo duomenų gavyba. Visą interneto duomenų gavybos mokslą apima du pagrindiniai aspektai: informacijos radimas (aktualių dokumentų ir duomenų paieška) ir informacijos gavyba (aktualių duomenų ir faktų išgavimas).




***2.1 pav.*** *Ryšys tarp duomenų gavybos kategorijų ir jų tikslų*

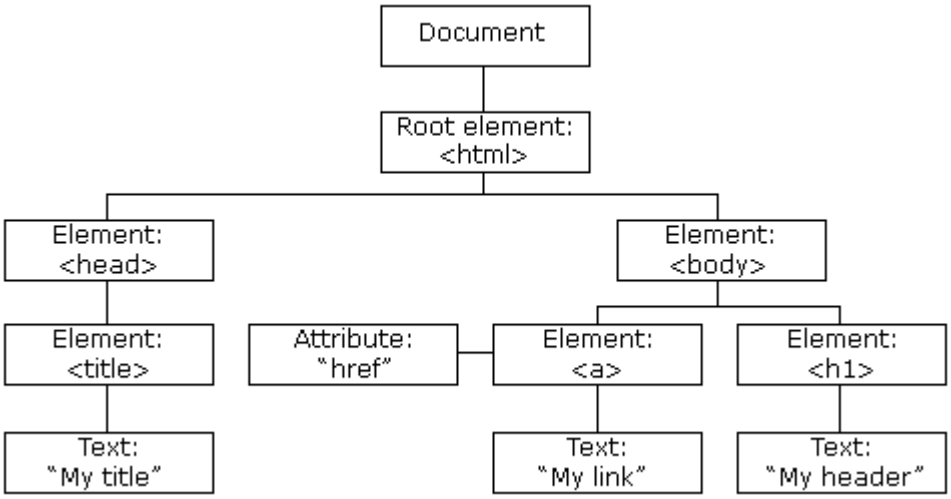
2.1.1. *Interneto turinio duomenų gavyba*

Interneto turinio duomenų gavyba yra teksto, vaizdo įrašų, grafikų ar paveikslėlių gavyba ir ištraukimas iš internete pasiekiamų dokumentų. Taip pat, vadinamas ir teksto gavyba. Interneto turinio duomenų gavyba analizuoja interneto išteklių turinį ir faktus taip, kaip šie yra pateikti naudotojams. Didžioji dalis internete esančių duomenų yra nestruktūrizuoti duomenys. Pagrindiniai turinio duomenų gavybos tikslai yra pagerinti informacijos filtravimą ir suradimą vartotojams ir valdyti žiniatinklyje rastus duomenis (Hussein, Mohamed ir Mousa, 2010).

2.1.2. *Interneto struktūros duomenų gavyba*

Interneto struktūros duomenų gavybos tikslas yra sukurti struktūrinę interneto svetainių santrauką, kuri rodytų vartotojų ir interneto svetainių tarpusavio ryšį. Taip pat aiškinamasi apie puslapių struktūras ir pagal kokias schemas yra išdėstyti bei susieti puslapių dokumentai. Du pagrindiniai struktūriniai tipai (Srivastava, Prasanna ir Vipin, 2005):

- Hipertekstas. Tai tekstas, kuris savyje gali turėti nuorodas į kitus tekstus ar puslapius. Tos nuorodos gali nukreipti naudotoja į kitas puslapio ar svetainės vietas (ir kitu svetainių puslapius);
- Dokumentų struktūros. Turinys esantis puslapio dokumente gali būti suorganizuotas į medžio struktūrą (žr. 2.2 pav.). Medžiai gali būti HTML, XML ir kitų dokumentų formatų.




***2.2 pav.*** *HTML dokumento medžio struktūros pavyzdys*

2.1.3. *Interneto naudojamumo duomenų gavyba*

Interneto naudojamumo duomenų gavyba yra procesas, kurio metu bandoma sužinoti, ko interneto naudotojai ieško internete. Taip pat siekiama atrasti naudingos informacijos iš naudotojų sąveikos su interneto naršymo įrankiais. Yra trys etapai sudarantys interneto naudojamumo duomenų gavybos procesą (Sharma, Kavita, Gulshan ir Vikas, 2011):

- Išankstinis apdorojimas. Reikalingas išgauti neapdorotus duomenis iš interneto išteklių.
- Šablonų nustatymas. Po išankstinio apdorojimo tie duomenys yra naudojami naudotojų įpročių šablonams atrasti.
- Šablonų analizė. Atradus modelius, jie yra analizuojami ir tikrinami. Jei modelis teisingas, jis yra implementuojamas į žiniatinklį, kad būtų galima išgauti dar daugiau informacijos iš interneto.

## 2.2. **Teksto klasifikacijos problematika**

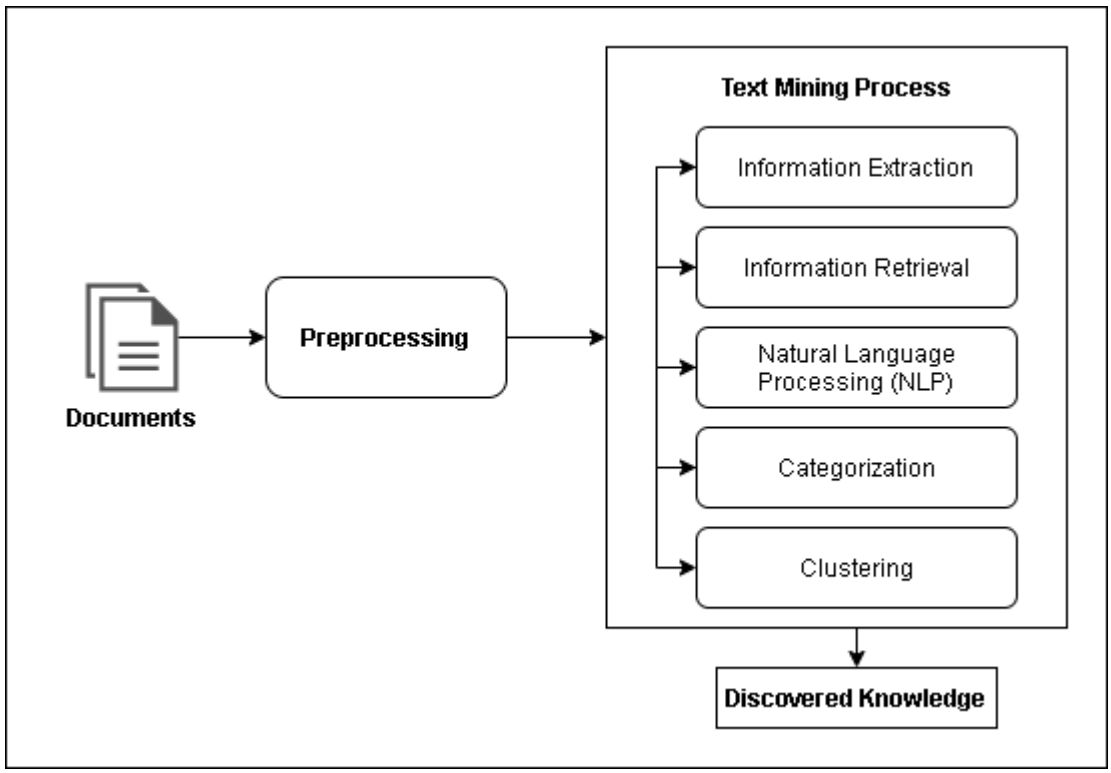
Didžioji dalis mašininio mokymosi algoritmų naudoja vektorius, paremtais realiaisiais skaičiais. Tokių vektorių komponentės dar yra vadinamos požymiais (angl. *features*). Kadangi tekstai nėra skaitinės reikšmės, reikalingi specialūs požymių išskyrimo algoritmai (angl. *feature extraction algorithms*), kurie sugebėtų transformuoti tekstus į skaitinius pavidalus ir paruoštų juos klasifikacijos metodams.

Pagrindinės teksto klasifikacijos problemos:

1. Tekstai yra sudaryti iš daugybės skirtingų simbolių, kurių ne visi yra skaitinės reikšmės, bet kategoriniai duomenys;

2. Tekstų ilgiai yra nepastovūs. Tuo tarpu didžiosios daugumos mašininio mokymosi algoritmų įvestys būna fiksuoto ilgio vektoriai;
3. Mašininio mokymo algoritmai yra linkę išnaudoti didelius kiekius kompiuterinių resursų, ypač, kai dirbama su milžinišku kiekiu požymių, kuris ir yra būdingas natūralios kalbos tekstams.

Visoms šioms problemoms spręsti yra naudojami keli tekstinių duomenų gavybos metodai (žr. 2.3 pav.): informacijos išgavimas, natūralios kalbos apdorojimas, kategorijų nustatymas ir klasterizavimas.




***2.3 pav.*** *Teksto duomenų gavybos žingsniai*

## 2.3. **Lietuviškų tekstų komponentai ir apdorojimas**

NLP teksto klasifikavimas yra pakankamai gerai išnagrinėtas ir išvystytas anglų kalbos tekstams. Dėl lietuvių kalbos sudėtingumo ir unikalumo, jos tekstų klasifikavimas yra daug sudėtingesnis ir reikalaujantis daugiau tyrimų.

Norint pateikti tinkamai sukategorizuotus lietuviškų straipsnių pasiūlymus, naudojantis tam pritaikytus klasifikacijos metodus, reikia surinkti tinkamos apimties duomenų rinkinį (šio darbo tyrimui - naujienų straipsnių tekstus). Pagrindiniai reikalavimai duomenų rinkiniui:

- Santykiniai didelis unikalių tekstų (straipsnių) kiekis;
- Kiekvienas tekstas turi būti paruoštas mašininio mokymo metodams taikyti;

- Surinkti metaduomenys apie skaitytojų patirtis su nepritaikytais straipsnių pasiūlymais;
- Tekstai turi išlaikyti savo sintaksę, simboliką, morfologinę ir stilometrinę vertę.

Natūralios kalbos struktūra apima kelis esminius komponentus. Kuo daugiau šių komponentų NLP sistema padengia – tuo natūralios kalbos atpažinimas tikslesnis. 7 natūralios kalbos komponentai (Briscoe, 2013):

1. Fonetika. Tai kalbos ir jos garsų interpretacija. Norint atpažinti natūralios kalbos fonetiškumą, reikia sugebėti išspręsti problemas, kai greitai pasakyti žodžiai persidengia vieni su kitais, yra ištariami skirtingai, ar naudojant skirtingas intonacijas bei nuotaikas.
2. Fonologija. Nagrinėja kalbos garsinių elementų funkcijas ir garsų tarpusavio sąveikos modelius. Fonologijos vienetai fonemos kartu sudaro fonologinę kalbos sistemą.
3. Morfologija. Jos tikslas yra išspręsti problemas, kurios kyla, kuomet pakeitus žodžio galūnę ar priesagą pasikeičia žodžių prasmė.
4. Leksikonas. Tai žodžių reikšmės supratimas. Žodžiams priskiriamos kalbos žymės, tačiau problema atsiranda, kai tas pats žodis gali turėti kelias kalbos žymes. Tam gali būti naudojami ir statistiniai metodai, nurodantys, kokios yra tikimybės, kad tam tikri žodžiai gali būti vienas šalia kito.
5. Sintaksė. Tai gramatinės sakinių struktūros, nusakančios, kokios kalbos dalys turi sekti kitas kalbos dalis. Didžiausia sintaksės problema ta, kad dvi identiškos sakinių struktūros gali visiškai pakeisti sakinių prasmes.
6. Semantika. Žodžių prasmės nebūtinai yra tokios, kokios jos yra apibrėžtos kalbų žodynuose. Dažnai, norint apibrėžti tas prasmės, reikia žinoti kontekstą ir ryšius tarp naudojamų sąvokų. Tokioms problemoms spręsti kuriamos ontologijos, kurios naudojamos kompiuterinėms programoms, bandančioms perprasti minėtuosius ryšius tarp žodžių prasmių ir bendrus jų kontekstus.
7. Pragmatika. Jos tikslas yra atsakyti į klausimus, kas buvo norėta pasakyti ir kas galėtų būti pasakyta tais pačiais žodžiais ar kitais kalbos ženklais. Elementarūs pavyzdžiai yra pasišaipymas ir ironija. Didelę reikšmę suteikia kalbėtojo intonacija, tarimo būdas, kalambūrų naudojimas, kurių supratimui reikia ir žinoti foniškumo ypatumus. Tai bene sudėtingiausias natūralios kalbos komponentas, nes žmonės ne visada supranta juokus, anekdotus ar ironiškus pasakymus.

## 2.4. **Lietuviško teksto klasifikacija**

Natūralios kalbos teksto klasifikacijos uždaviniai yra plačiai aprašyti ir dažnai tiriami, tačiau tyrimai dažniausiai orientuojasi į anglų kalbos tekstus ir jos ypatumus. Lietuvių kalba yra gramatiškai sudėtingesnė, todėl pasiekti teigiamų rezultatų yra daug sunkiau. Dėl šios priežasties lietuviško teksto klasifikacijai vis dar reikia skirti daug dėmesio ir tirti geriausiai tam tinkamus metodus. 2019 metais buvo publikuoti keli darbai, kurie siekė įgyvendinti lietuviško teksto klasifikacijos uždavinį.

Pirmajame straipsnyje, pavadinimu „Didelį kategorijų kieki turinčių draudimo bendrovės klientų užklausų, gautų elektroniniais laiškais, lietuviško teksto klasifikavimas“, Karolis K. ir Simona R. aprašo tyrimą, kurio tikslas yra įvertinti didelį kategorijų kiekį turinčių draudimo bendrovės klientų užklausų, parašytų lietuvių kalba, klasifikavimo tikslumą, taikant skirtingus teksto apdorojimo ir klasifikacijos metodus. Šio tyrimo metu buvo atliekami eksperimentai, siekiant kuo tiksliau suklasifikuoti 51 tūkstančius elektroninių laiškų pagal 33 kategorijas. Praktiškai pritaikius teksto transformacijos teoriją, išbandžius kelis metodus ir identifikavus tinkamiausius parametrus, nustatyta, kad išsikeltam tikslui labiausiai tiko atraminių vektorių mašinų algoritmas, kuris sugebėjo pateikti 66,75 % tikslumo rezultatus.

Kitame straipsnyje „Sentiment Analysis of Lithuanian Texts Using Traditional and Deep Learning Approaches“ Jurgita K.-D., Robertas D. ir Marcin W. analizuoja naujienų portale esančių lietuviškų komentarų klasifikacijos galimybes. Duomenų rinkinį šiame tyrime sudarė virš 10 tūkstančių įrašų, kuriuos skirstė pagal 3 kategorijas. Nors buvo klasifikuojamas lietuviškas tekstas, daug įtakos turėjo ir tekste naudojamos grafinės emocijos. Geriausią rezultatą tyrimo metu parodė daugialypio naiviojo Bajeso (angl. *naive Bayes multinominal*) algoritmas, sugebėjęs pasiekti net 73,50 % tikslumą.

Bendra panašių tyrimų apžvalga leidžia susidaryti įspūdį, kokie duomenų gavybos metodai lietuviško teksto klasifikacijos uždaviniuose yra tinkamiausi. Taip pat, abiejuose tyrimuose identifikuota teksto transformacijos svarba, reikalinga tiksliausiems rezultatams gauti.

## 2.5. **Teksto transformacijos žingsniai**

Kad tekstas taptų struktūriškai paprastesnis, o teksto požymių išskyrimas tikslesnis, yra vykdomos kelios esminės teksto transformacijos (Vijayarani ir Janani, 2016) (žr. 2.5 pav.):

1. Suskaidymas į žodžius (angl. *tokenization*). Tai transformacijos forma, kurios metu žodžiai yra atskiriami vienas nuo kito, nuo skyrybos ženklų ir kitų simbolių formų. Skyrybos ženklai gali būti pašalinami arba interpretuojami kaip savarankišką prasmę turintys vienetai.

2. Žodžių šalinimas (angl. *word removal*). Tai teksto transformacijos žingsnis, kurio metu iš teksto žodžių sąrašo yra šalinami nereikšmingi, prasmės neturintys ir labai dažnai randami žodžiai. Dažniausiai žodžių šalinimas yra atliekamas turint minėtų žodžių sąrašus, į kuriuos įtraukiami prielinksniai, jungtukai, artikeliai, įvardžiai ir kitos savarankiškai nereikšmingos kalbos formos.

3. Žodžių kamienų išgavimo (angl. *stemming*) algoritmai yra naudojami norint transformuoti teksto žodžius į jų gramatinių šaknų formas (Ramasubramanian ir Ramya, 2013). Pavyzdžiui, žodžių "kompiuteris", "kompiuteryje" ir "kompiuterizuotas" šaknis yra "*kompiuter*". Žodžių kamienų išgavimo algoritmų tikslas yra pašalinti įvairias žodžių galūnes, tokiu būdu sumažinant turimų žodžių kiekį. Sumažėjęs žodžių kiekis taupo atminties vietą ir laiką, vykdant apdoroto teksto klasifikacijos metodus. Tai ypač aktualu gramatiškai sudėtingų kalbų tekstams, tokiems kaip lietuvių kalba. Žodžių kamienų išgavimas yra kanonizacijos (angl. *canonicalization*) dalis. Kanonizacija, tai teksto standartizacijos ir normalizacijos procesas, kurio metu tekstas, kuris gali būti perfrazuotas keliais būdais, yra pakeičiamas standartizuotą ir paprasčiausią galimą formą. Kanonizacija sumažina teksto požymių skaičių ir suefektyvina klasifikatorių veikimą. Žodžių kamienų išgavimo pavyzdys matomas 2.4 paveikle.

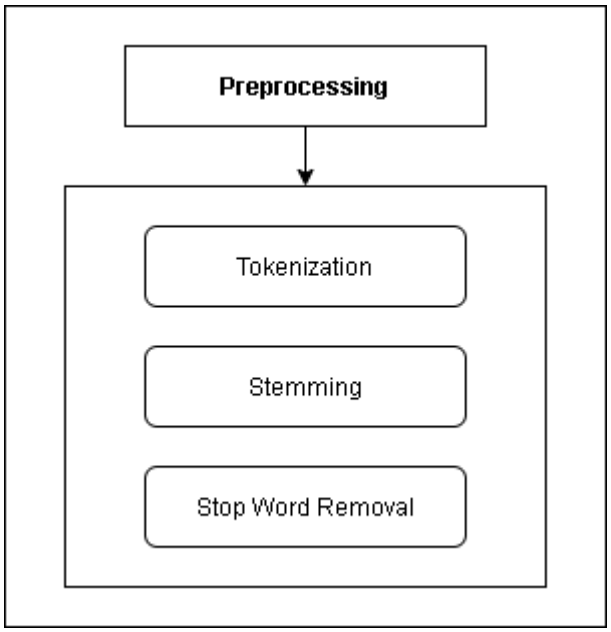
| **Prieš metodo panaudojimą:** | **Po metodo panaudojimo:** |
|---|---|
| "Tai yra pavyzdinis tekstas, kuriam taikomas duomenų paruošimo metodas "Stemming". Duomenų gavyboje šis metodas naudojamas, siekiant išgauti rašytinių žodžių šaknis. Iš apdorojamo teksto pašalinami skaičiai (pavyzdžiui "23"), skyrybos ženklai ir kiti simboliai (pavyzdžiui "$"). Apdoroti metodo rezultatai gali būti naudojami statistiniams ir kitiems kategorizavimo uždaviniams." | "tai yr pavyzdin tekst kur taikom duomen paruošim metod stemm duomen gavyb šis metod naudo siek išg rašytin žod kamien iš apdoro tekst pašalin skait pavyzd skyryb ženkl ir kit simbol pavyzd apdor metod rezultat gal būt naudo statistin ir kit kategorizavim uždavin" |

***2.4 pav.*** *Žodžių kamienų išgavimo įvesties ir išvesties pavyzdys*

Žodžių kamienų išgavimo algoritmai turi porą pagrindinių neigiamų pusių bei trūkumų (Jivani, 2011):

- Per smarkus algoritmų panaudojimas. Tai atvejis, kuomet skirtingi žodžiai yra transformuojami iki vienodo žodžio kamieno. Tai vadinama klaidingai teigiamu rezultatu.

- Per silpnas algoritmų panaudojimas. Tai atvejis, kuomet du žodžiai, kurie turi vienodą kamieną, yra transformuojami neteisingai ir gaunami skirtingi žodžių kamienai. Tai vadinama klaidingai neigiamu rezultatu.

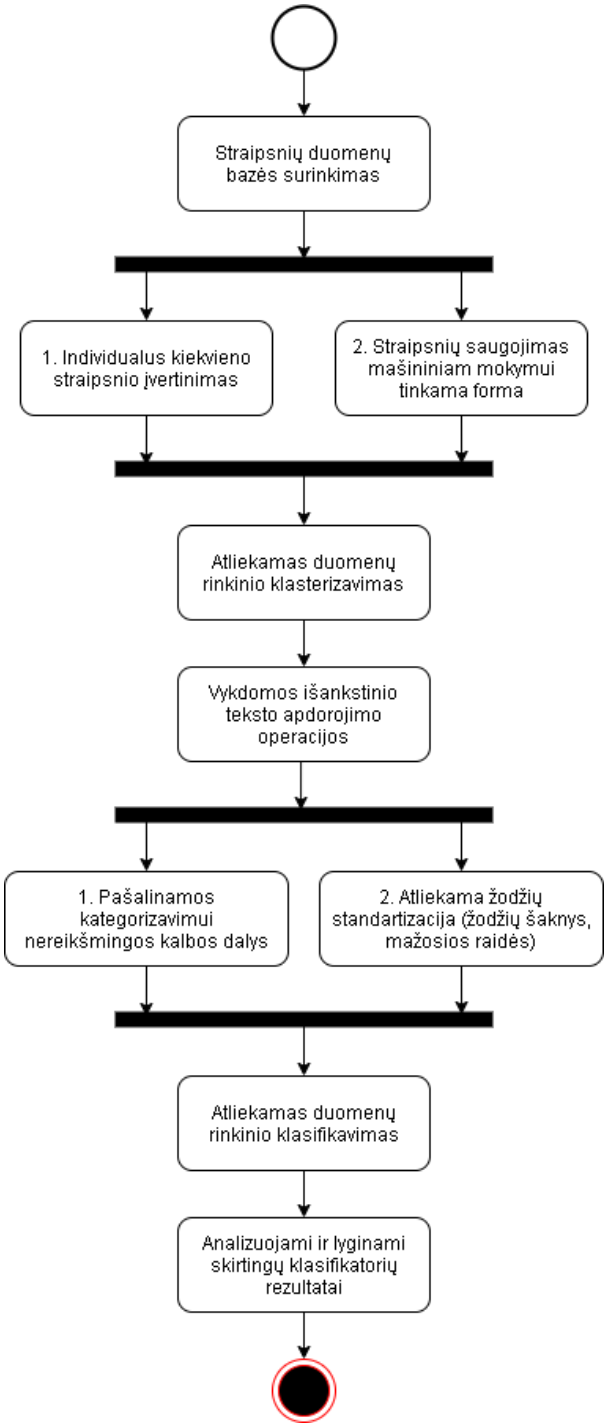



***2.5 pav.*** *Išankstinio teksto apdorojimo operacijos*

## 3. Tyrimo metodika

Tyrimo metu yra surenkamas, apdorojamas ir panaudojamas unikalus duomenų rinkinys. Šis panaudojamas dviem duomenų gavybos ir mašininio mokymo uždaviniams spręsti – klasterizavimui ir klasifikavimui. Tyrimas yra išskaidomas į penkias pagrindines dalis (žr. 3.1 pav.):

1) Iš pasirinktų naujienų portalų renkama internetinių straipsnių duomenų bazė. Straipsniai siunčiami į lokalią duomenų bazę ir yra paruošiami skaitytojo įvertinimui.
    a) Kiekvienas parsiųstas straipsnis individualiai pateikiamas naujienų skaitytojui ir saugojami duomenys apie jo interakciją su duomenimis (skaitymo statistika ir įvertinimai).
    b) Kiekvieno straipsnio metaduomenys, pavadinimas, antraštė, naujienos tekstas ir skaitytojo įvertinimai bei skaitymo laikas yra saugojami duomenų rinkinio forma, kuri bus tinkama mašininio mokymo programinės įrangos įvesčiai.
2) Pasirinktu metodu atliekamas duomenų rinkinio įrašų klasterizavimas. Įrašai sugrupuojami į du klasterius – rekomenduotinus ir nerekomenduotinus naujienų straipsnius. Klasterizavimas vykdomas atsižvelgiant į tris duomenų rinkinio atributus (ar straipsniai skaityti, kaip straipsniai įvertinti ir kiek laiko straipsniai skaityti). Suklasterizuoti duomenys bus naudojami siekiant nustatyti, ar klasifikavimas pagal tekstą koreliuoja su skaitytojo įvertinimais.
3) Vykdomos išankstinio teksto apdorojimo operacijos. Neapdorotas tekstas yra netinkamas klasifikavimo uždaviniams spręsti, todėl yra apdorojami prieš jų panaudojimą. Geriausiems rezultatams pasiekti reikia pašalinti nereikšmingus žodžius, o reikalingus – standartizuoti pagal tas pačias taisykles.
    a) Pašalinamos kategorizavimui nereikšmingos kalbos dalys. Tokiomis dalimis yra vadinami prielinksniai, jungtukai, ištiktukai, jaustumai, dalelytės ar bet kokie kiti rašytiniai simboliai.
    b) Atliekama teksto žodžių standartizacija. Kad lietuvių kalbos sudėtingumas minimaliai apsunkintų tyrimą ir turimi tekstiniai duomenys būtų tinkamesni klasifikavimo uždaviniams spręsti, žodžiai perrašomi mažosiomis raidėmis ir paliekamos tik žodžių šaknys.
4) Keliais metodais atliekamas apdoroto duomenų rinkinio klasifikavimas. Klasifikavimas vykdomas naudojant tik standartizuotus straipsnių tekstus. Klasifikatoriai nuspėja, kokiam klasteriui kiekvienas straipsnis priklauso, o naudojama programinė įranga patikrina, ar spėjimai atitinka anksčiau klasterizavimo nustatytas straipsnių grupes.
5) Analizuojami ir lyginami skirtingų klasifikatorių rezultatai ir identifikuojami tinkamiausi užduočiai klasifikavimo būdai. Klasifikatoriai lyginami pagal tikslumo matus, pateikiant klasifikatorių vertinimo matricas ir ROC kreives.




***3.1 pav.*** *Tyrimo eigos schema*

## 3.1. **Įrankiai ir technologijos**

Tyrimui atlikti naudojama turima kompiuterinė įranga ir tinkamą funkcionalumą turinti programinė įranga. Programavimo užduotims naudojama C# programavimo kalba ir Microsoft Visual Studio aplinka, duomenų gavybos užduotims naudotas RapidMiner Studio platforma, o rezultatų vizualizacijoms naudotas Orange įrankis.

### 3.1.1. *Tyrimui naudoti kompiuteriniai resursai*

1. Asmeninis kompiuteris:

- Procesorius: *Intel i5-4460 (3.20 GHz)*.
- Darbinė atmintis: *8.00 GB (1333 MHz)*.
- Vaizdo plokštė: *NVIDIA GTX 1060 3GB*.
- Pirmojo monitoriaus rezoliucija: *16:9 Full HD (1920x1080)*.
- Antrojo monitoriaus rezoliucija: *16:9 Full HD (1920x1080)*.
- Operacinė sistema: *Microsoft Windows 10 Pro 64-bit*.

2. Nešiojamasis kompiuteris:
   - Procesorius: *Intel i7-2760QM (2.40 GHz)*.
   - Darbinė atmintis: *8.00 GB (1333 MHz)*.
   - Vaizdo plokštė: *NVIDIA GT 555M 3GB*.
   - Monitoriaus rezoliucija: *16:9 Full HD (1920x1080)*.
   - Operacinė sistema: *Microsoft Windows 10 Home 64-bit*.

#### 3.1.2. *Tyrimui naudotos technologijos*

Sistemai kurti naudota programinė įranga:

1. Programavimo aplinka: *Microsoft Visual Studio 2019 Community IDE*.
2. Programavimo kalba: *C# (.NET Framework)*.
3. Duomenų gavybos platforma: *RapidMiner Studio Educational*.
4. Duomenų gavybos įrankis: *Orange*.
5. Duomenų gavybos plėtinys: *Text Processing (v9.3.001)*.
6. Duomenų gavybos plėtinys: *Web Mining (v9.7.000)*.

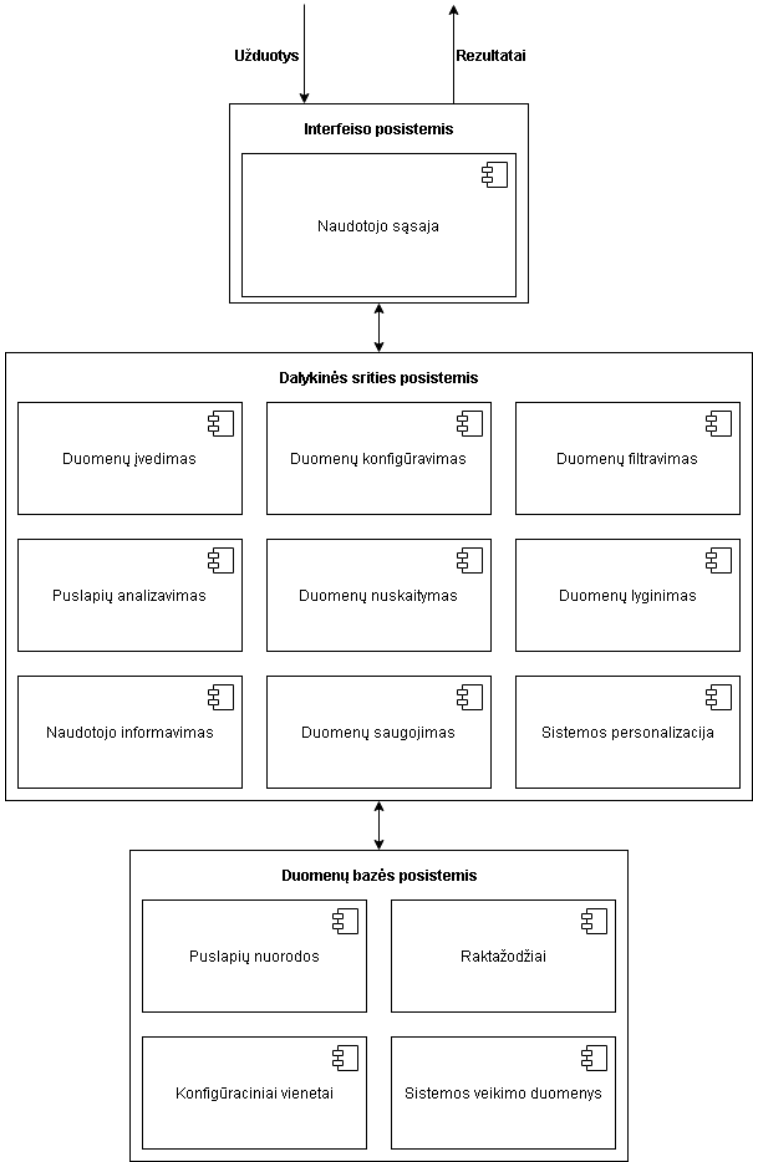
Pagrindinėms tyrimo užduotims, duomenims klasterizuoti ir klasifikuoti, yra sukurta daugybė duomenų gavybos įrankių, tačiau šiame darbe pasirinkta naudoti *RapidMiner Studio*, kuris pasižymi savo algoritmų gausa, lengvomis duomenų importavimo ir eksportavimo galimybėmis, GUI kokybe ir plačiu duomenų gavybos rezultatų atvaizdavimo funkcionalumu.

### 3.2. **Duomenų rinkinio paruošimas**

Norint identifikuoti aktualias naujienas ir pateikti tinkamas internetinių straipsnių rekomendacijas reikia parengti tinkamą duomenų rinkinį. Šie duomenų dokumentai yra naudojami dirbtinių neuroninių tinklų apmokymui ir jų testavimui.

Didžiulis internetinių naujienų straipsnių paketas buvo surinktas naudojant bakalauro baigiamajam darbui sukurtoje sistemoje. Internete viešinamų duomenų stebėjimo sistema yra reikalinga šiuolaikiniams žmonėms, kurie yra priversti stebėti ir sekti didžiulius kiekius duomenų. Darbas, laisvalaikis, hobiai ir kiti laiko leidimo būdai verčia domėtis ir ieškoti naudingos informacijos, kurios yra užtektinai, tačiau labiausiai ribotas yra duomenų ieškojimo laikas. Būtent tam ir reikalingos tokios sistemos kaip ši, kurios taupytų žmonių naudotojų laiką ir rastų reikiamą informaciją už juos.

Žemiau pateikta UML komponentų diagrama (žr. 3.2 pav.), kuri vaizduoja į kokius komponentus yra dekomponuota internete viešinamų duomenų stebėjimo sistema.

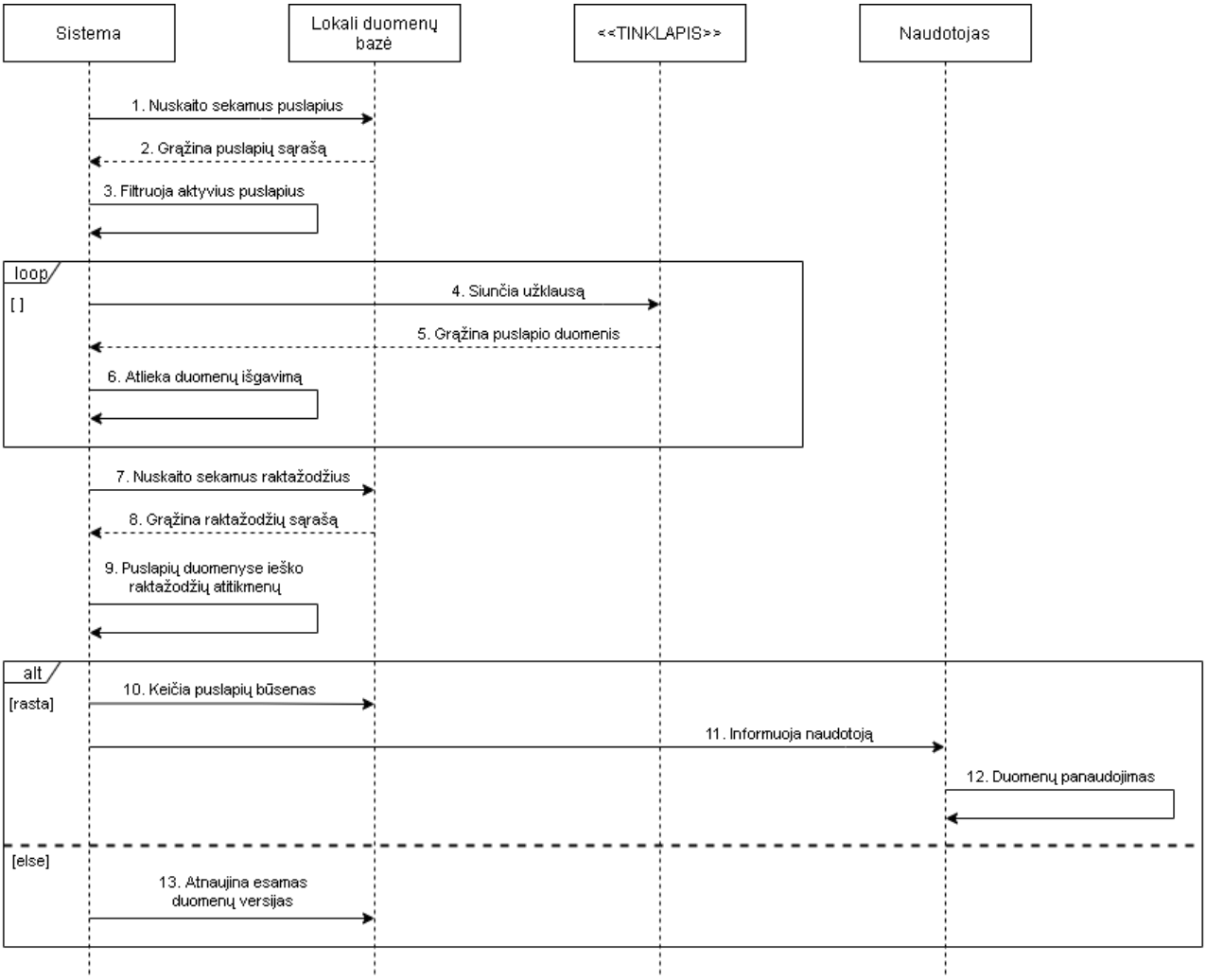



***3.2 pav.*** *Duomenų surinkimo sistemos dekompozicija*

Sistema reguliariai tikrina naudotojo sekamus puslapius (žr. 3.3 pav.). Vyksta tikrinimas, ar puslapiai yra aktyvūs (ar naudotojas neišjungė konkretaus puslapio sekimo) ir kreipiasi į visų aktyvių puslapių nuorodas, siekdamas pasisiųsti puslapių HTML dokumentus. Po duomenų nuskaitymo

vykdomas tų duomenų karpymas, reikalingų dalių išgavimas. Sutvarkius puslapio informaciją taip, kaip reikia, sistema pasiruošia naudotojo sekamus raktažodžius. Tuomet suformuotuose duomenyse pradedama ieškoti paminėtų ieškomų raktažodžių. Šioje vietoje įvyksta loginė sąlyga, priklausomai, ar buvo rasta atitikmenų:

- Jeigu atsakymas „TAIP“: puslapio konfigūracijoje puslapis pažymimas kaip turintis naujienų, o naudotojui imama rodyti pranešimą apie rastą raktažodį su nuoroda į šaltinio puslapį;
- Jeigu atsakymas „NE“: atnaujinami turimi puslapio duomenys pagal suformuotą ir surinktą šio proceso eigoje informaciją.

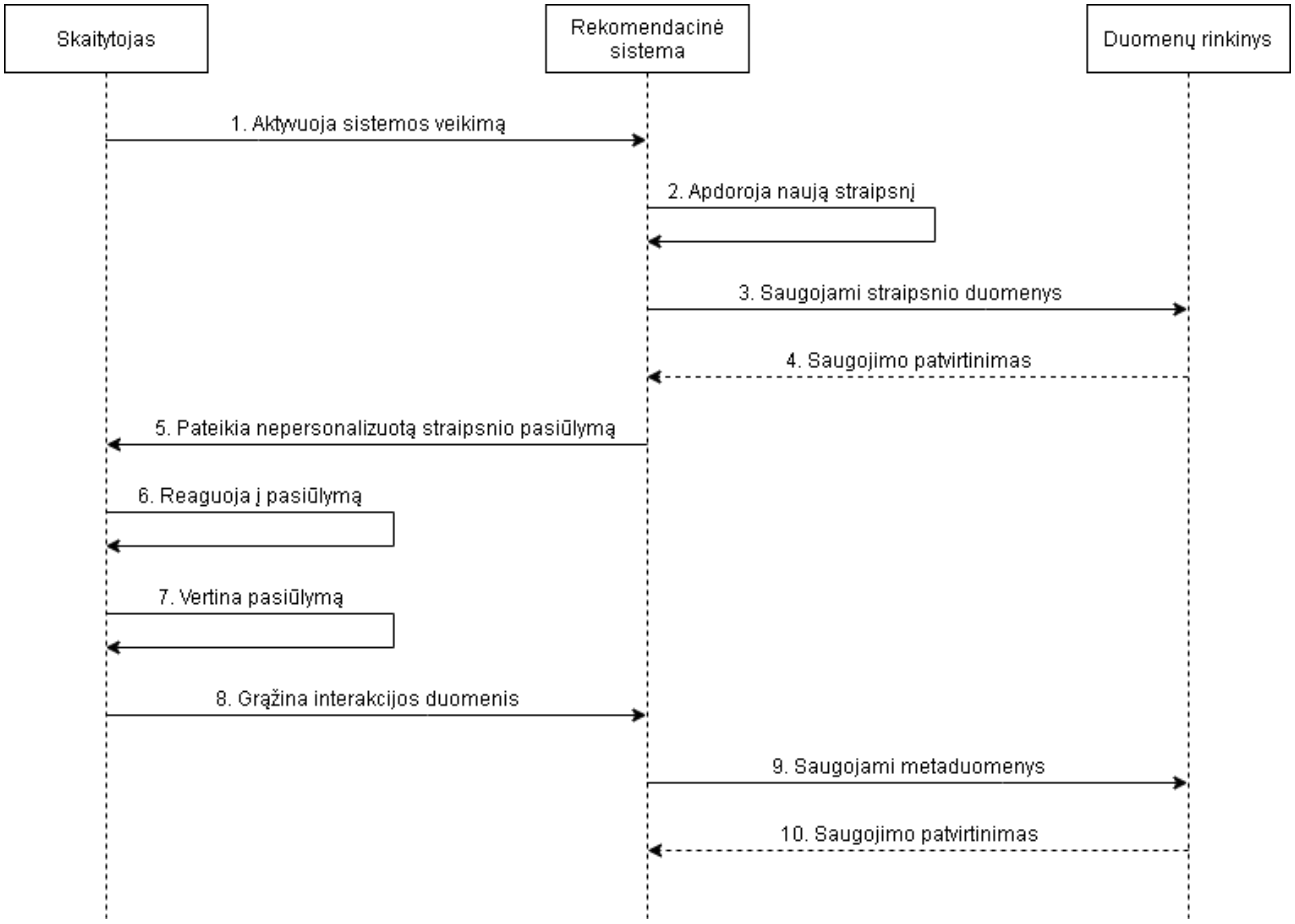



***3.3 pav.*** *Duomenų nuskaitymo ir raktažodžių paieškos sekų diagrama*

Magistro baigiamasis darbas toliau dirba ties minėta sistema, tačiau vietoje elementaraus aktualių straipsnių ieškojimo pagal raktažodžius, yra naudojami duomenų gavybos ir mašininio mokymo metodai, leidžiantys suformuoti pritaikomąsias rekomendacijas sistemos naudotojams.

Remiantis minėtosios mobiliosios programėlės funkcionalumu, jai pritaikomas ir šio tyrimo modelis (žr. 3.4 pav.). Prie egzistuojančių funkcijų pridedama galimybė vartotojui įvertinti jam

pateiktą nepritaikytą pasiūlymą. Šie įvertinimai saugojami metaduomenų pavidalu, kaip duomenys, apibūdinantys straipsnių įrašus.




***3.4 pav.*** *Pasiūlyto modelio pritaikymo sekų diagrama*

Duomenų rinkiniui paruošti buvo naudojami trys pagrindiniai informacijos šaltiniai:

- *Delfi* (www.delfi.lt).
- *15min* (www.15min.lt).
- *Lietuvos Rytas* (www.lrytas.lt).

Duomenų rinkinį sudaro 6000 unikalių įrašų, iš kurių kiekvienas reprezentuoja atskirą naujienų straipsnį. 2593 straipsniai surinkti iš *Delfi.lt* portalo, 2093 straipsniai iš *15min.lt* portalo ir 1314 straipsniai iš *LRytas.lt* portalo.

Duomenų rinkinys buvo parengtas pagal iš anksto nustatytus reikalavimus. Kiekvieno straipsnio įrašas turėjo šiuos atributus:

- Straipsnio pavadinimas.
- Straipsnio šaltinis (naujienų portalo pavadinimas).
- Nuoroda, vedanti į straipsnį.

- Straipsnio identifikacinis numeris.
- Tekstinis straipsnio dokumentas (netekęs HTML žymių).

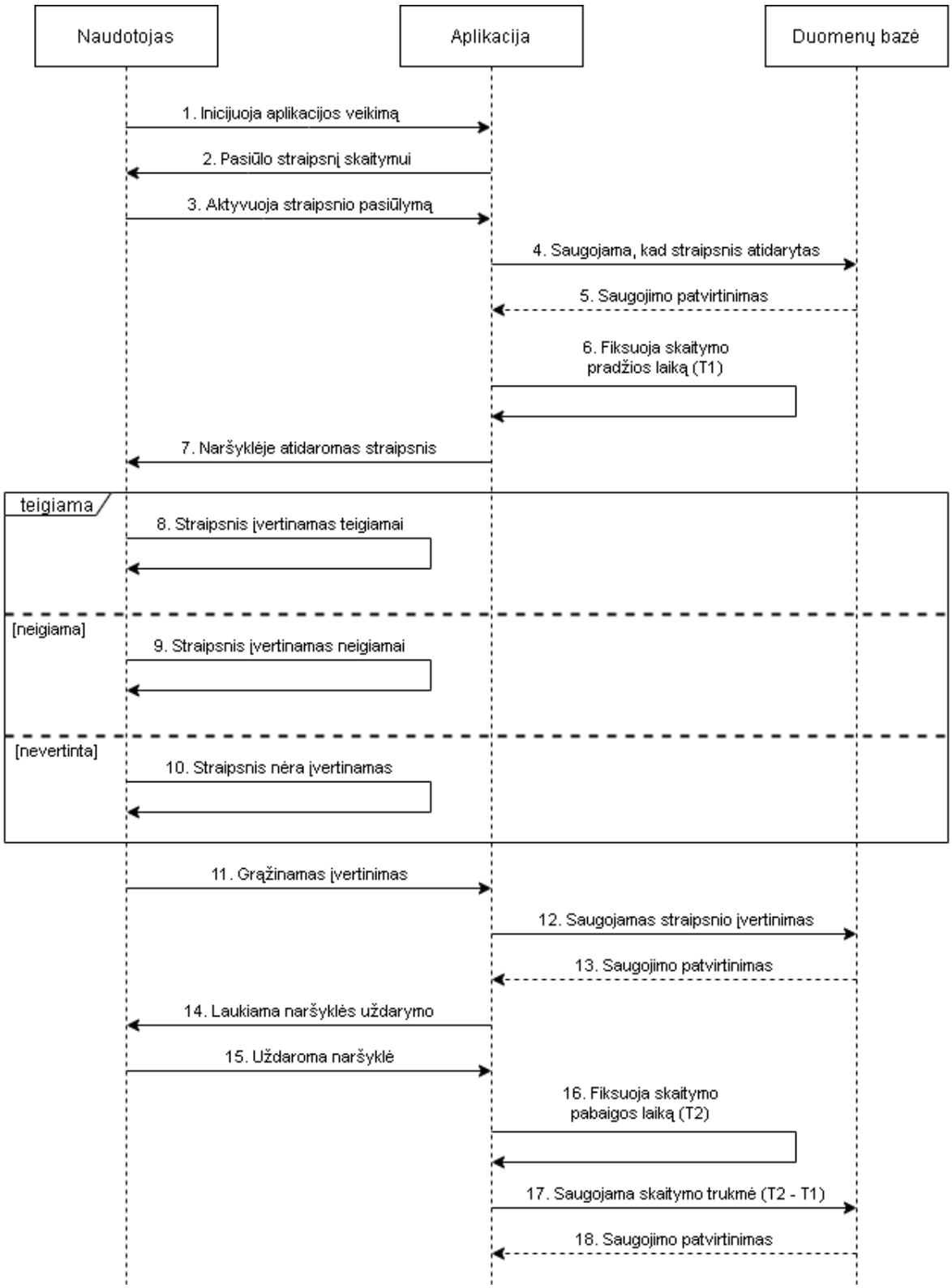
### 3.3. **Atributų reikšmių rinkimas**

Kad būtų tinkamai atliktas duomenų įrašų klasterizavimas, reikėjo surinkti duomenis apie skaitytojų interakcijas su naujienų straipsniais iš originalaus duomenų rinkinio. Šiam tikslui prie kiekvieno straipsnio buvo priskirti ir duomenimis užpildyti dar 3 atributai:

1. Atributas *Opened*, reiškiantis, ar skaitytojas iš viso atidarė jam pateiktą (ne pritaikytą) puslapio rekomendaciją. Tai binominis atributas, kuriame reikšmė 0 reiškia, jog straipsnis nebuvo atidarytas, ir reikšmė 1, reiškianti, kad straipsnis buvo atidarytas;
2. Atributas *Liked*, reikšiantis, ar skaitytojui patiko jo atidarytas ir skaitytas straipsnis. Tai trinaris atributas su 3 galimomis reikšmėmis. Reikšmė 1 reiškia, kad straipsnis skaitytojui patiko. Reikšmė 0 reiškia, kad skaitytojas neįvertino skaityto straipsnio arba jo nuomonė yra neutrali. Reikšmė -1 reiškia, kad skaitytojui straipsnis nepatiko arba tiesiog buvo neaktualus.
3. Atributas *TimeRead*, reiškiantis, kiek laiko skaitytojas skaitė straipsnį, arba buvo jį atidaręs. Atributo reikšmės pateiktos sekundžių tikslumu, o laikas buvo skaičiuotas nuo straipsnio atidarymo iki grįžimo į aplikaciją momento.

Klasterizavimui naudojamų aprašytų atributų reikšmės surenkamos minėtoje mobiliojoje programėlėje (žr. 3.5 pav). Sistema pasiūlydama straipsnį skaitymui nustato reikšmę T1, kuri reiškia skaitymo pradžios laiką. Ši reikšmė fiksuojama skaitytojui atidarius straipsnį savo naršyklėje. Perskaitęs/uždaręs straipsnį skaitytojas gauna pasiūlymą įvertinti turinį. Įvertinimas gali būti teigiamas, neigiamas arba neutralus. Uždarius naršyklę sistema fiksuoja laiką T2, reiškiantį skaitymo pabaigą. T1 ir T2 skirtumas laikomas skaitymo trukme. Šios sekos pabaigoje sistema saugo atributų reikšmes: atributas *Opened* gauna reikšmę 1, reikšiančią, jog straipsnis skaitytas, atributas *Liked* gauna reikšmę, atitinkančią skaitytojo įvertinimą, o atributas *TimeRead* gauna sekundėmis išreikštą laiko tarpą (skaitymo trukmę).

Tokiu būdu surinkus 6000 unikalių straipsnių įrašų, kurie reikalinga klasterizavimo uždaviniui įgyvendinti.




***3.5 pav.*** *Atributų reikšmių rinkimo sekų diagrama*

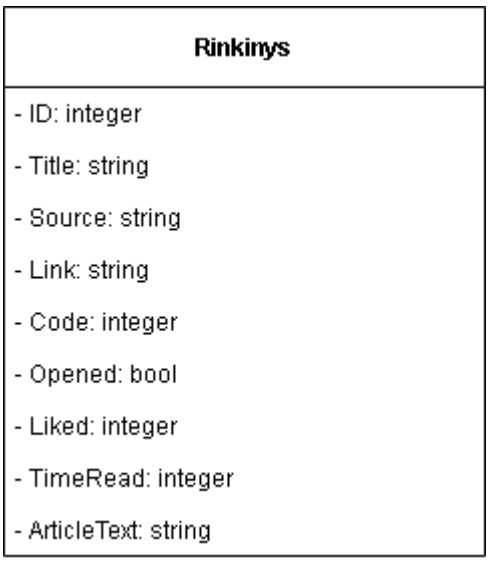
### 3.4. **Duomenų rinkinio struktūra**

Duomenų rinkinys ima kauptis iš karto po sistemos naudojimo pradžios. Sistema nuolatos ieško naujienų straipsnių tyrimo aprašymo nurodytuose naujienų portaluose. Vos radusi naujieną, sistemą apdoroja straipsnį – išgauna grynąjį tekstą (pašalinamos HTML žymės ir kodas). Išgryninti duomenys išsaugojami duomenų rinkinio bazėje. Kadangi tai nėra kategorizuotas straipsnis, sistema

pateikia jį skaitytojui kaip nepritaikytą jam pasiūlymą. Skaitytojas reaguoją į jam pateiktą pasiūlymą, o jeigu straipsnis buvo skaitytas – jį įvertina. Visi apie interakciją surinkti duomenys apie straipsnį pateikiami sistemai, kuri saugo juos duomenų rinkinyje šalia konkretaus straipsnio įrašo. Tokia seka kartojama, kol yra surenkamas pakankamas straipsnių kiekis duomenų rinkinyje tyrimui vykdyti.

Galutinį duomenų rinkinį (naujienų straipsnių duomenų bazę) sudaro vienintelė lentelė (žr. 3.6 pav.). Lentelės atributai:

- Atributas *ID,* kuris yra skaitinio tipo ir identifikuoja įrašą (konkretų straipsnį).
- Atributas *Title*, kuris yra tekstinio tipo ir saugoja straipsnio pavadinimą.
- Atributas *Source*, kuris yra tekstinio tipo ir saugoja naujienų portalo pavadinimą.
- Atributas *Link,* kuris yra tekstinio tipo ir saugoja nuorodą, vedančią į konkretų straipsnį.
- Atributas *Code*, kuris yra skaitinio tipo ir yra identifikacinis numeris portalų sistemose.
- Atributas *Opened*, kuris yra loginio tipo ir žymi, ar skaitytojas skaitė straipsnį.
- Atributas *Liked,* kuris yra skaitinio tipo ir žymi, kaip skaitytojas įvertino straipsnį.
- Atributas *TimeRead*, kuris yra skaitinio tipo ir žymi, kiek laiko (sekundėmis) skaitytojas skaitė pateiktą straipsnį.
- Atributas *ArticleText*, kuris yra tekstinio tipo ir saugo visą tekstinį straipsnio dokumentą.




***3.6 pav.*** *Duomenų rinkinio įrašų lentelė*

### 3.5. **Efektyvumo vertinimas**

Siekiant nustatyti, kaip tiksliai sistema nuspėja, kuriuos straipsnius skaitytojas perskaitys, yra taikomi rekomendacinių sistemų efektyvumo įverčiai. Rekomendacinė sistema veikia kaip klasifikatorius, kuris nustato ar konkretūs straipsniai bus perskaityti ar jų net nėra verta rekomenduoti. Klasifikatoriaus gebėjimą teisingai priskirti straipsnius tinkamoms klasėms galima vizualiai pateikti keliais būdais.

Pirmasis būdas įvertinti klasifikatorių rezultatus yra painiavos matrica (angl. *confusion matrix*). Joje pirma klasė (skaičiavimuose žymima *cluster_1* pavadinimu) yra rekomenduojami straipsniai, o antra klasė (skaičiavimuose žymima *cluster_0* pavadinimu) yra nerekomenduojami straipsniai. Klasifikatoriaus tikslas – kuo tiksliau nustatyti, kurie straipsniai bus skaitytojui aktualūs, o kurių nėra verta jam rekomenduoti. Dažnai klasifikatoriai priskiria vienos klasės elementus prie kitos klasės rezultatų – tai vadinama klasifikavimo klaidomis.

Keturi spėjimų įvertinimų atvejai šiame tyrime:

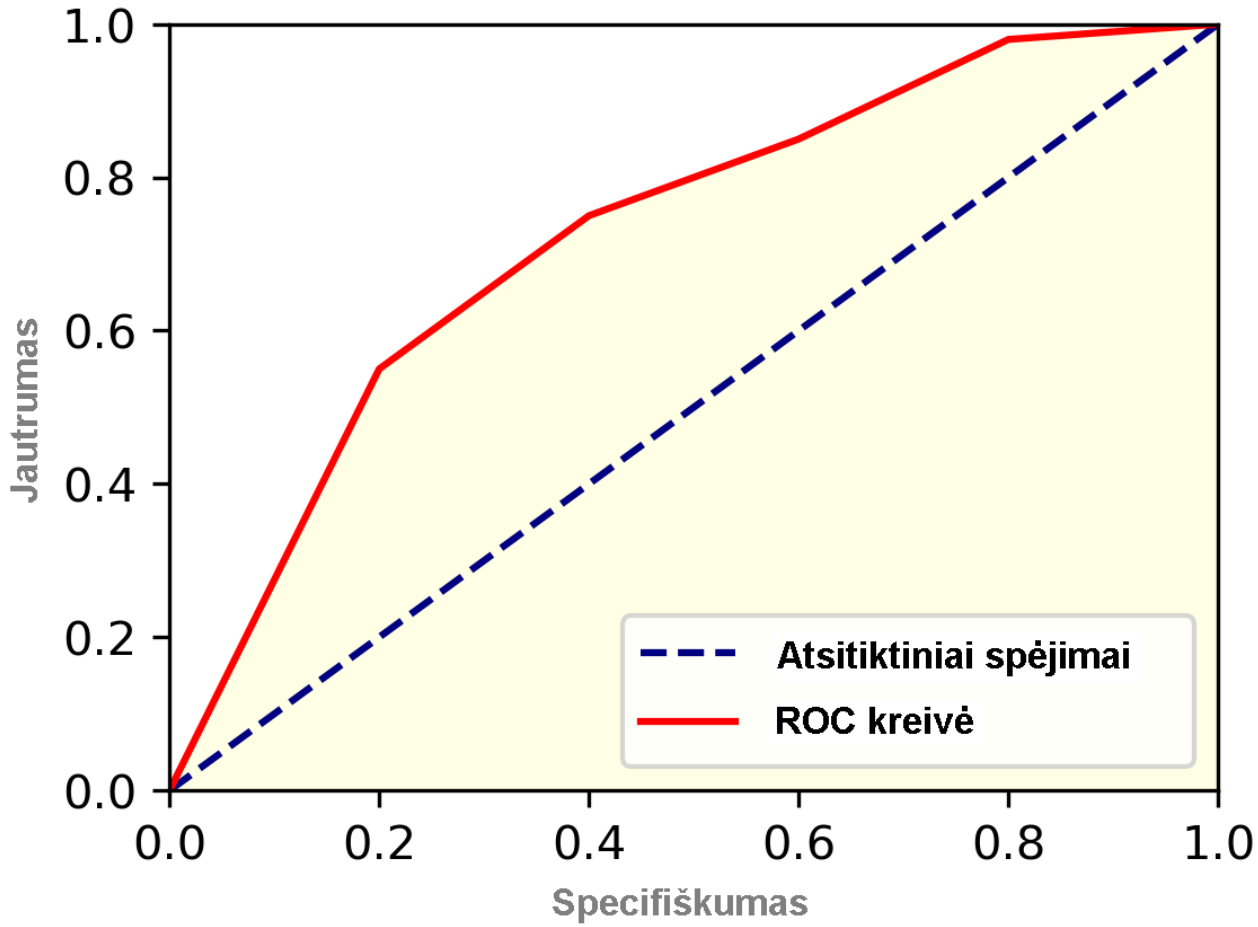
- TP (angl. *true positive*) – rekomenduojamus straipsnius rekomenduoja ir klasifikatorius.
- TN (angl. *true negative*) – nerekomenduojamų straipsnių nerekomenduoja ir klasifikatorius.
- FN (angl. *false negative*) – nerekomenduojami straipsniai, kuriuos klasifikatorius rekomenduoja.
- FP (angl. *false positive*) – rekomenduojami straipsniai, kurių klasifikatorius nerekomenduoja.

***3.1 lentelė.*** *Painiavos matricos struktūra*

| | | **Klasifikatoriaus rezultatai** | |
|---|---|---|---|
| | | Pirma klasė (rekomenduojami straipsniai) | Antrai klasė (nerekomenduojami straipsniai) |
| **Straipsnių klasės** | Pirma klasė (parekomenduoti straipsniai) | **TP** (angl. *true positive*) Teisingai priskirti pirmos klasės elementai | **FN** (angl. *false negative*) Neteisingai prie pirmos klasės priskirti antros klasės elementai |
| | Antra klasė (neparekomenduoti straipsniai) | **FP** (angl. *false positive*) Neteisingai prie antros klasės priskirti pirmos klasės elementai | **TN** (angl. *true negative*) Teisingai priskirti antros klasės elementai |

Kitas būdas grafiškai iliustruoti atvaizduoti klasifikatoriaus rezultatus yra ROC (angl. *Receiver Operating Characteristic*) kreivė (žr. 3.7 pav.). Šis grafikas rodo naudojamo metodo priklausomybę tarp jautrumo ir specifiškumo. Tiesi linija, dalinanti diagramą į dvi dalis reiškia, jog klasifikatorius spėjimus vykdė atsitiktinai ir nėra tinkamas užduočiai įgyvendinti. Teigiamo rezultato klasifikatorių ROC kreivės staigiai kyla ir laikosi viršuje.

Kitas svarbus matas, egzistuojantis ROC diagramoje yra plotas po ROC kreive, dar vadinamas AUC (angl. *Area Under the ROC Curve*), kuris rodo rekomendacijų gerumą įvertinant plotą po aprašyta kreive. Maksimali AUC reikšmė gali būti 1. Jei reikšmė ~0.5 – vadinasi klasifikatorius „spėliojo“. Idealiu atveju, kai rezultatai teigiami, AUC reikšmė yra tarp 0.5 ir 1. Tyrimo atveju, AUC reikšmė reiškia jog klasifikatorius rekomenduos atsitiktinį straipsnį, kuris ir turėtų būti rekomenduojamas, dažniau nei atsitiktinį straipsnį, kuris neturėtų būti rekomenduojamas.




***3.7 pav.*** *ROC kreivės pavyzdys*

Horizontali ROC diagramos ašis reiškia specifiškumą. Specifiškumas yra klasifikacijos įvertis, parodantis santykį, kiek konkrečiai klasei priskirtų įrašų iš tikrųjų priklauso tai klasei (tyrimo atveju – kiek rekomendacijai tinkamų straipsnių klasifikatorius tikrai parekomenduotų ir atvirkščiai). Kuo daugiau įrašų teisingai priskirtų įrašų patenka į konkrečią klasę, tuo specifiškumas didesnis ir rekomendacinė sistema tikslesnė.

Vertikali ROC diagramos ašis reiškia jautrumą. Jautrumas rodo teisingai prie konkrečios klasės priskirtų įrašų kiekio santykį su visu tai klasei priklausančių įrašų skaičiumi (tyrimo atveju – kiek įrašų klasifikatorius parekomendavo ir kiek įrašų tikrai turėjo būti parekomenduota). Kuo daugiau konkrečios klasės elementų yra jai priskirta, tuo jautrumas didesnis, o rekomendacinė sistema tikslesnė.

# 4. Klasterizacija

Tyrimo metu taikomo klasterizavimo tikslas – nustatyti duomenų objektų panašumą ir sugrupuoti panašius įrašus į atskirus klasterius. Tokiu būdu bus nustatyta, kurie internetiniai straipsniai skaitytojo buvo įvertinti panašiai ir yra verti rekomendacijos.

Kadangi duomenis aktualu suskirstyti tik į dvi grupes (rekomenduoti ar nerekomenduoti) ir yra žinomas galutinių klasterių kiekis, todėl tyrimui pasirinktas k-vidurkių metodas (angl. *k-means*).

Klasterizavimo metodas taikomas remiantis prielaida, kad straipsniai, kurie bus priskirti į vieną grupę, yra panašiai įvertinti skaitytojo. Skaitytojo vertinimas atsispindi trijuose atributuose: *Opened, Liked* ir *TimeRead*. Atributas *Opened* nurodo, ar skaitytojas skaitė/atidarė straipsnį naujienų portale. Atributas *Liked* nurodo, ar skaitytojas įvertino straipsnį ir kaip įvertino (patiko/nepatiko). Atributas *TimeRead* nurodo, kiek laiko skaitytojas skaitė atidarytą straipsnį iki jo uždarymo arba įvertinimo.

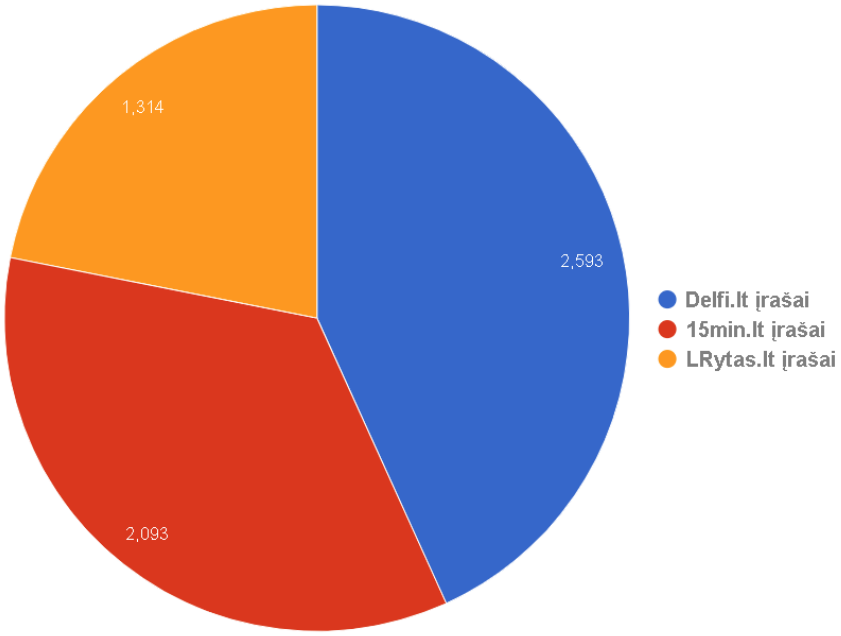
Grupės, į kurias klasterizacijos metu yra paskirstomi straipsniai, yra vadinamos klasteriais (angl. *clusters*). Klasteriai apibrėžiami kaip panašių straipsnių rinkiniai.

Klasterizacija atlikta naudojant *RapidMiner Studio* įrankį.

## 4.1. Klasterizacijos įvestis

Klasterizacijos įvestis šiame tyrime yra jau sukurta aplikacija surinktas duomenų rinkinys. Duomenų rinkinį sudaro 6000 unikalių įrašų apie naujienų straipsnius. Duomenų rinkinys surinktas iš trijų naujienų portalų (*Delfi.lt, 15min.lt* ir *LRytas.lt*). Statistiniai įvesties duomenys pagal naujienų įrašų kilmę (žr. 4.1 pav.):

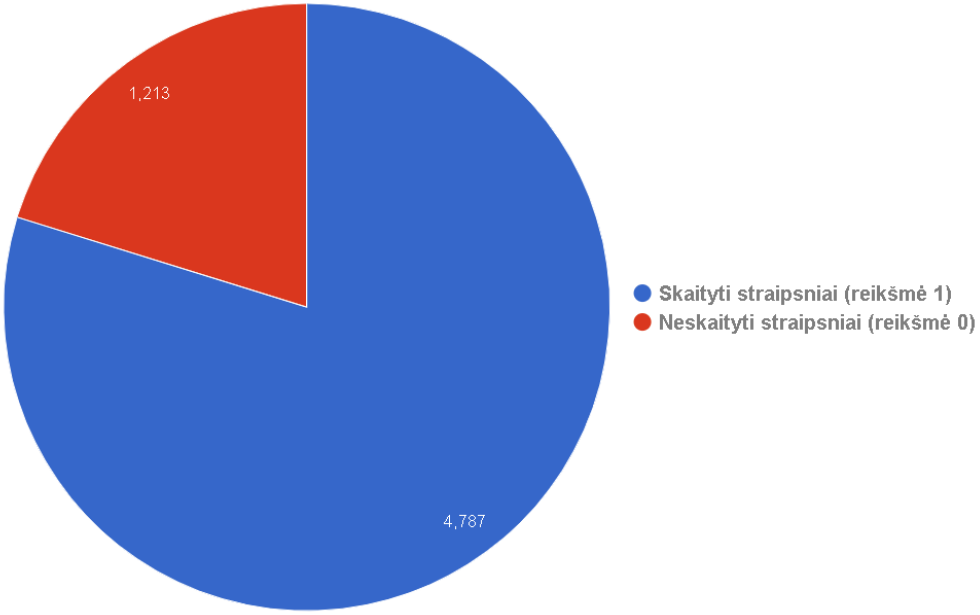
- 2593 įrašai yra surinkti iš *Delfi.lt* naujienų portalo (~43.22 %).
- 2093 įrašai yra surinkti iš *15min.lt* naujienų portalo (~34.88 %).
- 1314 įrašai yra surinkti iš *LRytas.lt* naujienų portalo (~21.90 %).




***4.1 pav.*** *Naujienų portalų įrašų kiekio skritulinė diagrama*

Pirmasis klasterizavimui naudojamas atributas yra *Opened*. Įrašai, turintys reikšmę „1“ buvo skaitytojo atidaryti, o turinys reikšmę „0“ nebuvo atidaryti. Kiekybinė atributo *Opened* statistika duomenų rinkinyje (žr. 4.2 pav.):

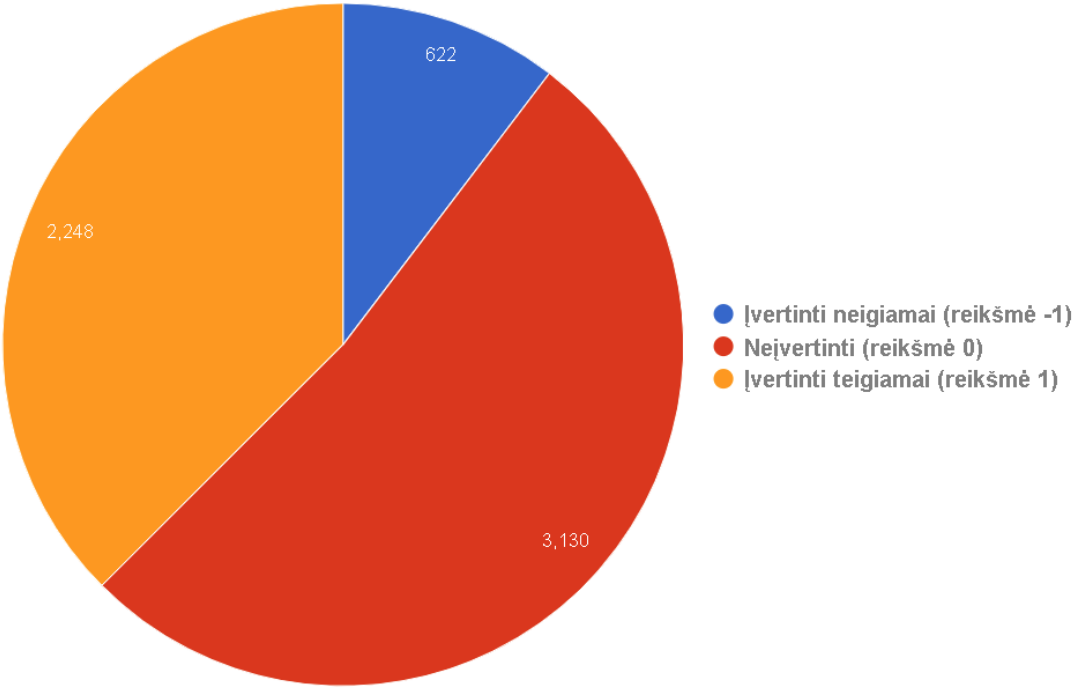
- 4787 straipsnių skaitytojas yra atidaręs/skaitęs (~79.78 %).
- 1213 straipsnių skaitytojas yra neatidaręs (ignoravęs pasiūlymą) (~20.22 %).




***4.2 pav.*** *Skaitytų ir neskaitytų straipsnių pasiskirstymo skritulinė diagrama*

Antrasis klasterizavimui naudojamas atributas yra *Liked*. Įrašai, turintys reikšmę „1“ buvo skaitytojo įvertinti teigiamai, turinys reikšmę „-1“ buvo įvertinti neigiamai, o turinys reikšmę „0“ nebuvo įvertinti. Kiekybinė atributo *Liked* statistika duomenų rinkinyje (žr. 4.3 pav.):

- 2248 straipsnių skaitytojas yra įvertinęs teigiamai (~37.47 %).
- 622 straipsnių skaitytojas yra įvertinęs neigiamai (~10.37 %).
- 3130 straipsnių skaitytojas nėra įvertinęs (~52.17 %).



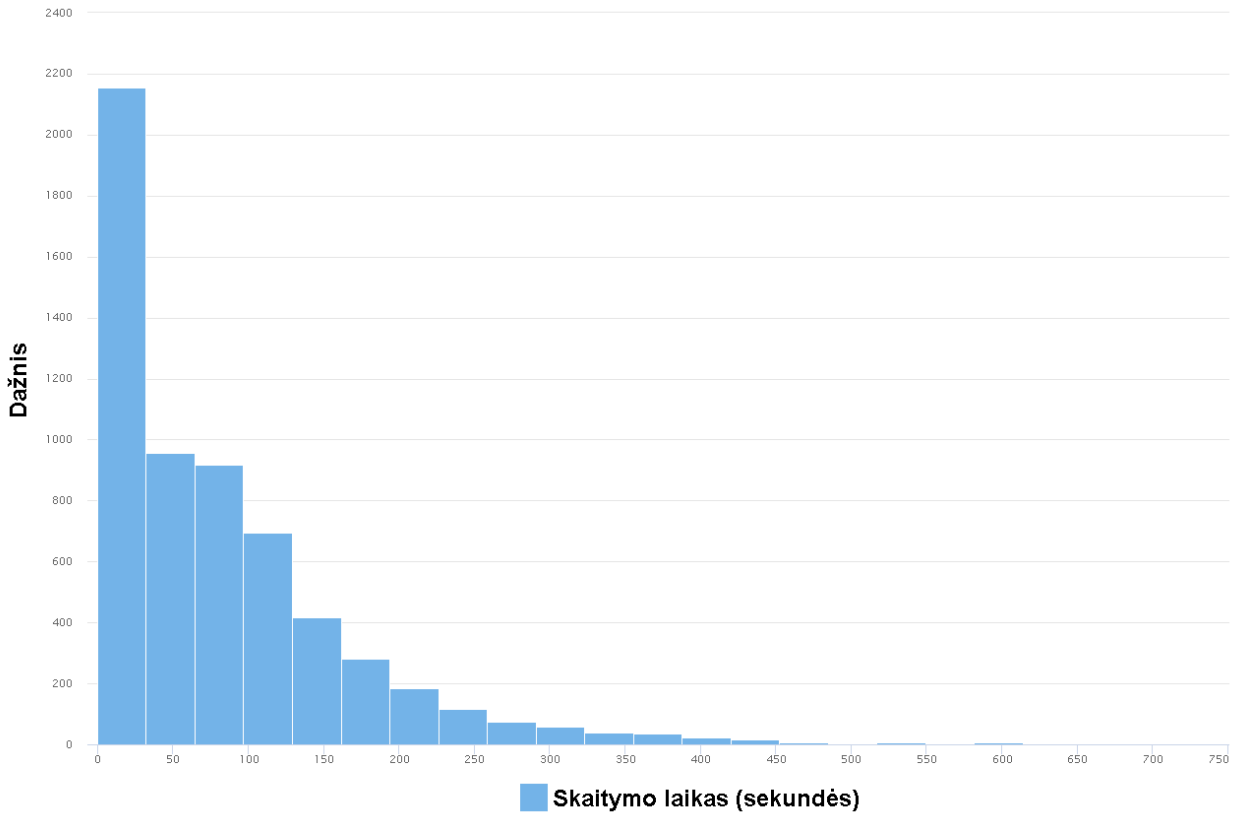

***4.3 pav.*** *Straipsnių įvertinimų pasiskirstymo skritulinė diagrama*

Trečiasis klasterizavimui naudojamas atributas yra *TimeRead*. Atributo reikšmės yra pateiktos skaitinėmis reikšmėmis, simbolizuojančiomis skaitymo trukmes. Atributo *TimeRead* statistika duomenų rinkinyje:

- Vidutinė visų straipsnių skaitymo trukmė: 74.78 sekundės.
- Vidutinė atidarytų straipsnių skaitymo trukmė: 102.63 sekundės.
- Trumpiausia straipsnio skaitymo trukmė: 0 sekundžių.
- Trumpiausia atidaryto straipsnio skaitymo trukmė: 0.55 sekundės.
- Ilgiausia straipsnio skaitymo trukmė: 743.67 sekundės (~12 minučių).
- Visų straipsnių skaitymo trukmės standartinis nuokrypis: 89.36.
- Atidarytų straipsnių skaitymo trukmės standartinis nuokrypis: 88.76.

4.4 pav. esančioje histogramoje matomas straipsnių įrašų pasiskirstymas pagal straipsnių skaitymo trukmę. Vertikali ašis rodo straipsnių kiekį, horizontali ašis rodo skaitymo laiką išmatuotą sekundėmis. Pirmasis histogramos stulpelis yra išskirtinai aukštas – jis aprėpia ir visus skaitytojo neatidarytus straipsnius. Didžioji dauguma skaitytų straipsnių patenka į 50 – 125 sekundžių skaitymo

trukmės ribą. Histogramoje matoma, kaip palaipsniui mažėja santykiniai ilgai skaitytų straipsnių kiekiai.




***4.4 pav.*** *Straipsnių skaitymo laiko pasiskirstymo histograma*

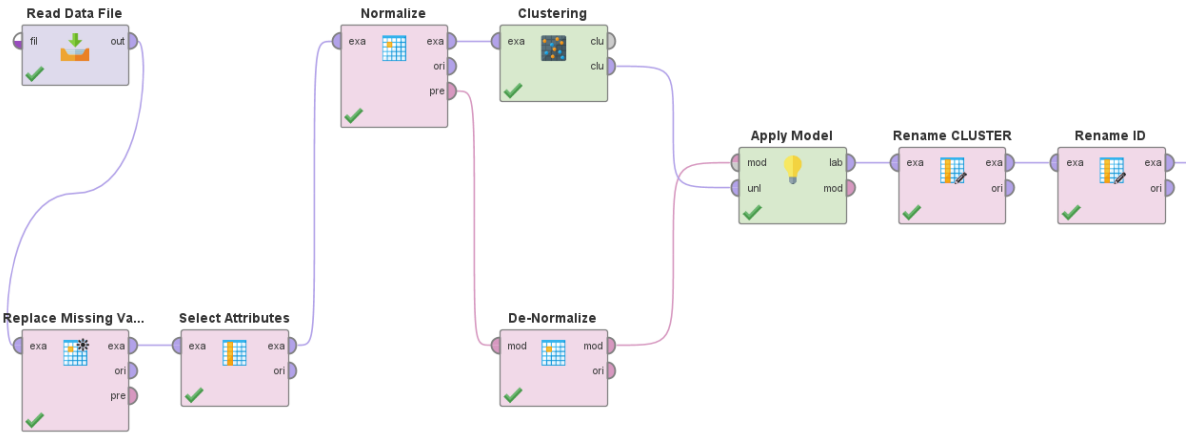
## 4.2. **Klasterizacijos procesas**

Atlikto klasterizacijos proceso (žr. 4.5 pav.) žingsniai:

1. Nuskaitomas duomenų rinkinio failas su straipsnių įrašais ir juos apibūdinančiomis atributų reikšmėmis.
2. Trūkstamos reikšmės pakeičiamos atributų vidurkiu. Nors pateiktas duomenų rinkinys neturi praleistų reikšmių, tačiau šio žingsnio realizacija yra gera praktika prieš vykdant klasterizavimą.
3. Pasirenkami atributai, kurie naudojami straipsnių priskyrimui prie konkrečių klasterių. Pasirinkti atributai: *Opened, Liked* ir *TimeRead*.
4. Normalizuojami duomenys. Duomenų normalizacija yra naudojama siekiant pašalinti nereikalingus duomenis ir užtikrinti, kad duomenys bus paruošti efektyviam klasterizavimo algoritmų vykdymui (Patel ir Mehta, 2011). Taip pat, tai padeda suteikti atributams vienodus svorius ir reikšmę klasterizavimo algoritmuose, net kai atributų tipai yra skirtingi, o jų reikšmės turi didelį skirtumą (skirtingos matavimų skalės). Šiam etapui pasirinkta Z-įverčio (angl. *Z-transformation*) metodas, kuris dar

yra vadinamas statistiniu normalizavimu. Metodas apskaičiuojamas iš atskirų reikšmių atimant populiacijos vidutinę reikšmę, o jų skirtumą padalijant iš populiacijos standartinio nuokrypio. Ši normalizavimo technika yra labai naudinga, nes išsaugo pirminį duomenų pasiskirstymą ir jam mažai įtakos turi pašaliniai rodikliai.

5. Klasterizacijos etapas, kurio metu konfigūruojami klasterizacijos parametrai. Atliekamas k-vidurkių klasterizavimas su $k = 2$ (norimas klasterių kiekis, reprezentuojantis rekomenduojamus ir nerekomenduojamus straipsnius). Atstumams matuoti naudojamas populiarus tokiuose tyrimuose Euklido atstumo kvadratų metodas, kurio rezultatų praktiškai neįtakoja išskirtys. Kiti klasterizavimo parametrai paliekami tokie, kokius numatė naudojama programinė įranga.
6. De-normalizacijos (angl. *de-normalize*) žingsnis yra skirtas įvesties duomenis atstatyti po jų normalizavimo etapo. Tai reikalinga galutiniams rezultatams išvesti ir statistiniams duomenims išgauti.
7. Paruoštiems standartizuoto formato duomenims yra pritaikomas sukonfigūruotas k-vidurkių klasterizavimo modelis.
8. Atliekami kiti kosmetiniai rezultatų lentelės pakeitimai, tinkamesnei rezultatų peržiūrai ir tinkamai klasifikacijos įvesčiai.




***4.5 pav.*** *Klasterizacijos proceso modelis*

## 4.3. **Klasterizacijos rezultatai**

Visi 6000 pateikti straipsnių įrašai ir jų atributai buvo grupuojami į du klasterius – nerekomenduojamų straipsnių (*cluster_0*) ir rekomenduojamų straipsnių (*cluster_1*). Statistiniai klasterizacijos rezultatų bruožai:

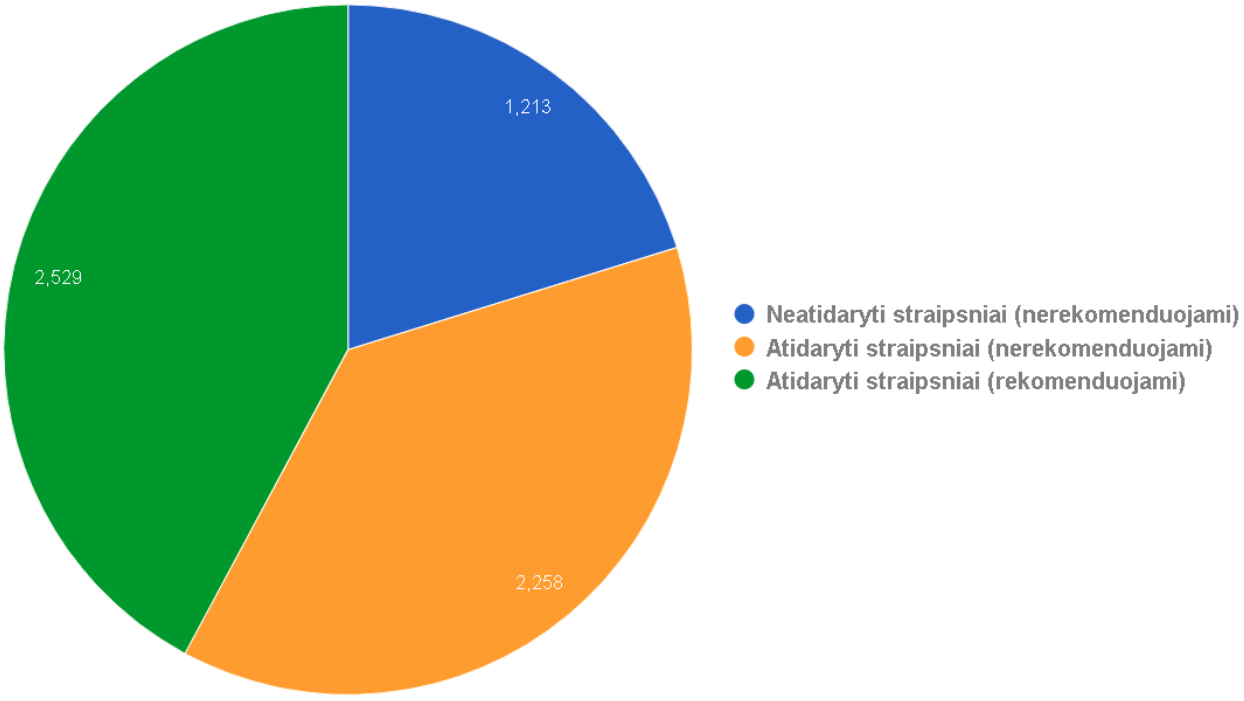
- Į nerekomenduojamų straipsnių klasterį *cluster_0* pateko 3471 straipsnis.

- Į rekomenduojamų straipsnių klasterį *cluster_1* pateko 2529 straipsniai.

Klasterizacijos tinkamumui ir jos rezultatams verifikuoti buvo išanalizuotas ryšys tarp klasterių ir jiems priskirtų straipsnių įrašų atributų ryšys ir jų statistiniai duomenys. Klasterizacija yra tinkama, kai jos rezultatuose randamos anomalijos gali būti logiškai paaiškintos ir pagrįstos straipsnius apibūdinančiomis atributų reikšmėmis.

Straipsnių pasiskirstymas klasteriuose pagal atributą, identifikuojantį, ar skaitytojas skaitė straipsnį, gali būti analizuojamas pagal keturias dalis – nerekomenduojami straipsniai, kurių skaitytojas neskaitė, nerekomenduojami straipsniai, kuriuos skaitytojas skaitė, rekomenduojami straipsniai, kurių skaitytojas neskaitė ir rekomenduojami straipsniai, kuriuos skaitytojas skaitė (žr. 4.6 pav.):

- Neskaityti nerekomenduojami straipsniai – 1213/6000 (~20.22 %).
- Neskaityti rekomenduojami straipsniai – 0/6000 (0.00 %).
- Skaityti nerekomenduojami straipsniai – 2258/6000 (~37.63 %).
- Skaityti rekomenduojami straipsniai – 2529/6000 (~42.15 %).

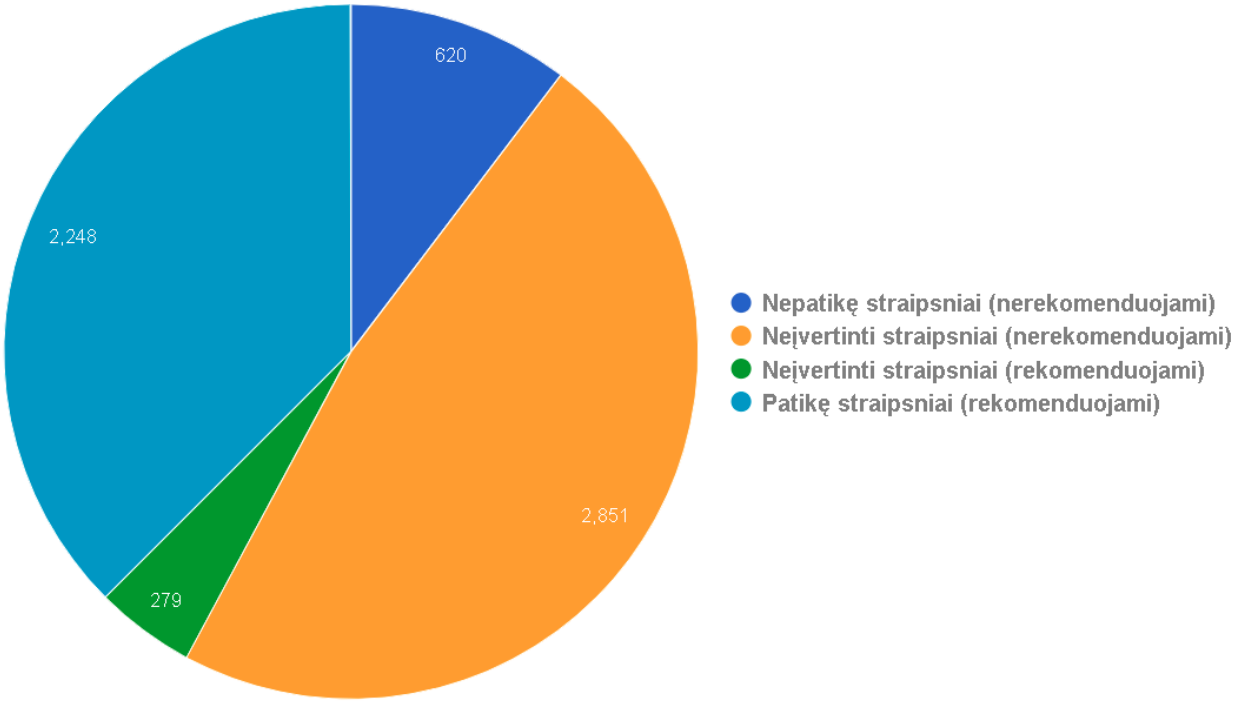



***4.6 pav.*** *Straipsnių skaitymo ir klasterizacijos rezultatų ryšio skritulinė diagrama*

4.6 pav. esanti diagrama sufleruoja klasterizacijos rezultatų teisingumą, nes nei vienas neskaitytas straipsnis nepateko į rekomenduojamų straipsnių klasterį. O skaitytų straipsnių pasiskirstymą po klasterių akivaizdžiai nulėmė detalesni skaitytojo interakcijos su straipsniais atributai (įvertinimai ir skaitymo trukmės).

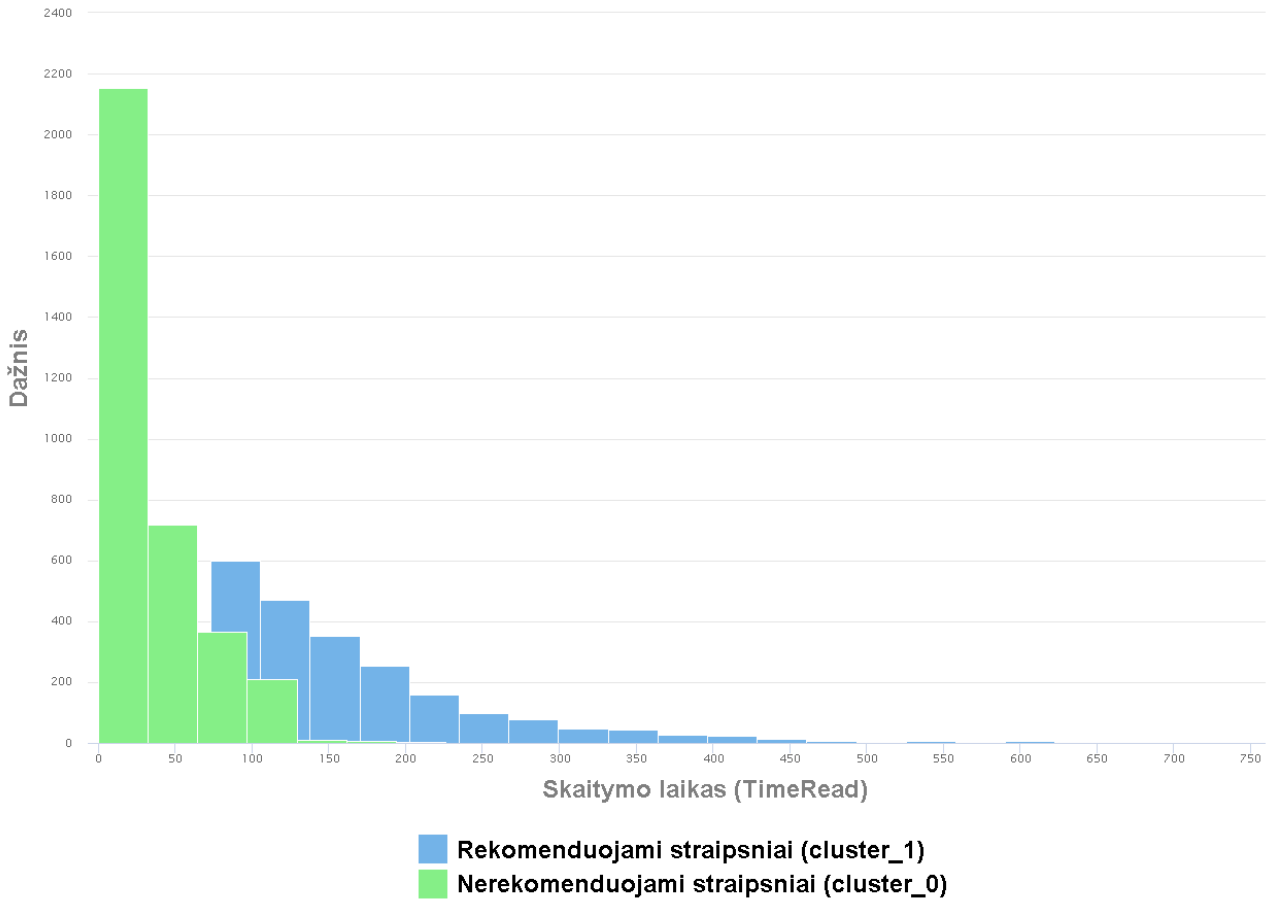
Straipsnių pasiskirstymas pagal skaitytojo įvertinimo atributą gali būti analizuojamas pagal šešias dalis – nerekomenduojami straipsniai, kuriuos skaitytojas įvertino neigiamai, nerekomenduojami straipsniai, kuriuos skaitytojas įvertino teigiamai, nerekomenduojami straipsniai, kurių skaitytojas neįvertino, rekomenduojami straipsniai, kuriuos skaitytojas įvertino neigiamai, rekomenduojami straipsniai, kuriuos skaitytojas įvertino teigiamai ir rekomenduojami straipsniai, kurių skaitytojas neįvertino (žr. 4.7 pav.):

- Nerekomenduojami straipsniai su neigiamu įvertinimu – 620/6000 (~10.33 %).
- Nerekomenduojami straipsniai su teigiamu įvertinimu – 0/6000 (0.00 %).
- Nerekomenduojami straipsniai be įvertinimo – 2851/6000 (~47.52 %).
- Rekomenduojami straipsniai su neigiamu įvertinimu – 2/6000 (~0.03 %).
- Rekomenduojami straipsniai su teigiamu įvertinimu – 2248/6000 (~37.47 %).
- Rekomenduojami straipsniai be įvertinimo – 279/6000 (~4.65 %).



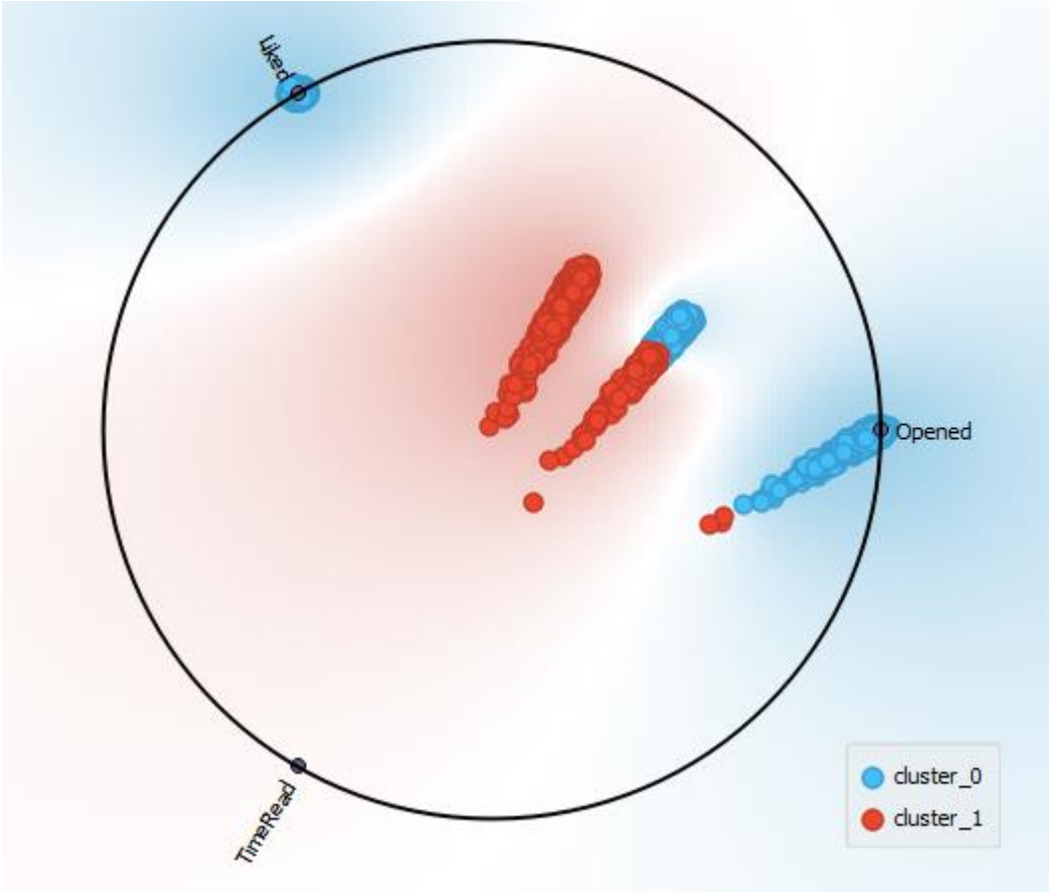

***4.7 pav.*** *Straipsnių įvertinimų ir klasterizacijos rezultatų ryšio skritulinė diagrama*

4.7 pav. esančioje diagramoje nepateikiami du straipsnių įvertinimų ir klasterizacijos rezultatų ryšio pastebėjimai: visi skaitytojui patikę straipsniai pateko į rekomenduojamų klasterį, tačiau 2 nepatikę straipsniai pateko į rekomenduojamų straipsnių klasterį. Šių dviejų straipsnių priežastis yra santykinai ilgos skaitymo trukmės, kurios buvo 327 ir 303 sekundės (vidutinė atidarytų straipsnių skaitymo trukmė yra 102 sekundės).




***4.8 pav.*** *Skaitymo trukmės ir klasterizacijos rezultatų ryšio histograma*

Duomenų rinkinio klasterizacijos rezultatai gali būti vizualiai atvaizduoti naudojant šilumos žemėlapį (angl. *heatmap*). Paveikslėlyje (žr. 4.9 pav.) duomenų įrašai, kurie simbolizuoja rekomenduojamus naujienų straipsnius yra sužymėti raudona spalva, o nerekomenduojamus – mėlyna spalva. Tos pačios spalvos atstovauja ir šilumos zonas.




***4.9 pav.*** *Klasterizuotų duomenų šilumos žemėlapis*

Vizualiai matomos šilumos žemėlapio išvados apie klasterizavimo rezultatus:

1. Visi straipsniai, kurių skaitytojas nebuvo atidaręs (straipsnis nėra įvertintas ir skaitymo laiko nėra), patenka į mėlynąją zoną (straipsniai nerekomenduojami).
2. Straipsniai, kuriuos skaitytojas įvertino teigiamai, pateko į raudonąją zoną (straipsniai rekomenduojami).
3. Straipsniai, kurių vartotojas neįvertino ir kurie turi santykinai ilgą skaitymo laiką, pateko į raudonąją zoną (straipsniai rekomenduojami).
4. Straipsniai, kurių vartotojas neįvertino ir kurie turi santykinai trumpą skaitymo laiką, pateko į mėlynąją zoną (straipsniai nerekomenduojami).
5. Didžioji dauguma straipsnių, kuriuos skaitytojas įvertino neigiamai, pateko į mėlynąją zoną (straipsniai nerekomenduojami).
6. Straipsniai, kuriuos skaitytojas įvertino neigiamai ir kurie turi išskirtinai ilgus skaitymo laikus, pateko į raudonąją zoną (straipsniai rekomenduojami).

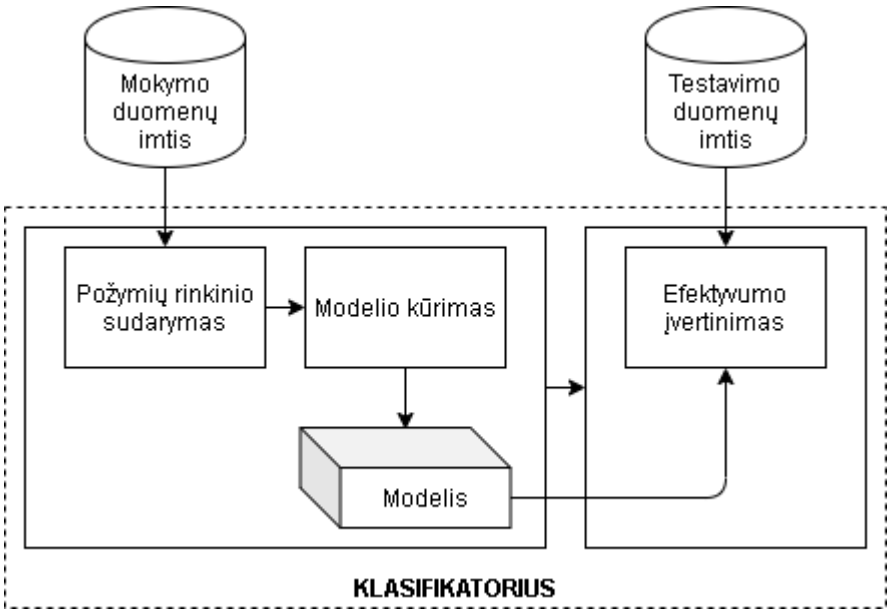
# 5. Klasifikacija

Tyrimo metu taikomos klasifikacijos tikslas – naudojantis skirtingais klasifikatoriais ir turimų straipsnių tekstų informacija, sugeneruoti pritaikytus pasiūlymus, o tyrimo pabaigoje palygintų klasifikatorių našumą ir lietuviškų tekstų klasifikavimo galimybes.

## 5.1. **Klasifikacijos principas**

Klasifikacijai buvo naudojami klasterizacijos dalyje gauti rezultatai. Kad klasifikacijai nebūtų nepadaryta jokia įtaka, naudoti ne 6000, o 5000 suklasterizuotų įrašų (po 2500 atsitiktinai pasirinktų įrašų iš rekomenduojamų ir nerekomenduojamų straipsnių klasterių). Tokiu būdu, tiriami klasifikatoriai turės vienodas sąlygas identifikuoti straipsnių tekstų ypatybes ir kaip jos reprezentuoja, kurie straipsniai skaitytojui yra aktualūs ar neaktualūs.

*RapidMiner Studio* programinėje įrangoje naudojami klasifikatoriai veikia 5.1 pav. pavaizduotos schemos principu – duomenų rinkinys padalinamas į dvi dalis (mokymo ir testavimo). Pagal mokymo duomenų imtį yra sudaromi požymių rinkiniai, kuriems pritaikomi klasifikacijos modeliai, kurių tikslumas vėliau tikrinamas su testavimui skirtomis duomenų imtimis. Išbandžius kelis klasifikatorius, visų jų rezultatai bus sureitinguoti nuo geriausiai pasirodžiusio, iki prasčiausio. Tyrimo pabaigoje pateiktas reitingas leis suformuluoti išvadas, kuris klasifikatorius labiausiai tinka pritaikytų pasiūlymų ir rekomendacijų formavimui, ir aš lietuvių kalbos tekstai yra tinkami tokiems uždaviniams spręsti.




***5.1 pav.*** *Klasifikatoriaus veikimo schema*

## 5.2. **Taikyti metodai ir rezultatai**

Naivaus Bajeso (angl. *Naive Bayes*) metodo rezultatai (žr. 5.1 lentelė) rodo, jog klasifikatorius „spėliojo“, nes klasifikatorius sugebėjo pateikti vos 51,20 % taiklumą su 1,20 % paklaida. AUC ploto reikšmė reiškia, jog klasifikatorius atsitiktinai pasirinktą įrašą dažniau įvertins tinkamai.

***5.1 lentelė.*** *Naivaus Bajeso klasifikatoriaus našumas*

| **Kriterijus** | **Reikšmė** | **Standartinis nuokrypis** |
|---|---|---|
| Taiklumas | 51.2 % | ± 1.4 % |
| Klasifikavimo klaida | 48.8 % | ± 1.4 % |
| AUC | 57.2 % | ± 1.2 % |
| Tikslumas | 74.2 % | ± 19.6 % |
| Teisingai teigiamų įverčių rodiklis | 3.4 % | ± 1.8 % |
| F-matas | 6.4 % | ± 3.3 % |
| Jautrumas | 3.4 % | ± 1.8 % |
| Specifiškumas | 98.7 % | ± 1.0 % |

5.2 lentelė atvaizduojama painiavos matrica rodo, jog klasifikatorius didžiąją daugumą įrašų priskyrė nerekomenduojamų straipsnių klasteriui (*cluster_0*). Vos geresnį nei vidutinį tikslumą lėmė sutapimas, jog nerekomenduojamų straipsnių įrašų kiekis testavimui buvo didesnis nei rekomenduojamų straipsnių.

***5.2 lentelė.*** *Naivaus Bajeso klasifikatoriaus painiavos matrica*

| | Tikrasis ***cluster_0*** | Tikrasis ***cluster_1*** | Spėjimo tikslumas |
|---|---|---|---|
| Nuspėtas ***cluster_0*** | 707 | 688 | 50.68 % |
| Nuspėtas ***cluster_1*** | 9 | 24 | 72.73 % |
| Klasės tikslumas | 98.74 % | 3.37 % | |

Bendro linijinio modelio (angl. *Generalized Linear Model*) metodo rezultatai (žr, 5.3 lentelė) rodo, jog klasifikatorius sugebėjo pateikti net 70,00 % taiklumą su 2,30 % paklaida, o plotas po ROC kreive siekia net 77,80 %. Šis klasifikatorius kur kas tinkamesnis už Naivųjį Bajesą.

***5.3 lentelė.*** *Bendro linijinio modelio klasifikatoriaus našumas*

| **Kriterijus** | **Reikšmė** | **Standartinis nuokrypis** |
|---|---|---|
| Taiklumas | 70.0 % | ± 2.3 % |
| Klasifikavimo klaida | 30.0 % | ± 2.3 % |
| AUC | 77.8 % | ± 1.8 % |
| Tikslumas | 75.8 % | ± 5.1 % |
| Teisingai teigiamų įverčių rodiklis | 59.6 % | ± 2.0 % |
| F-matas | 66.7 % | ± 2.4 % |
| Jautrumas | 59.6 % | ± 2.0 % |
| Specifiškumas | 80.6 % | ± 4.1 % |

5.4 lentelė esanti painiavos matrica rodo, kad nerekomenduojami straipsniai (*cluster_0*) buvo atspėti daug tiksliau nei rekomenduojami straipsniai (*cluster_1*) (atitinkamai 80,56 % ir 59,61 %).

***5.4 lentelė.*** *Bendro linijinio modelio klasifikatoriaus painiavos matrica*

| | Tikrasis ***cluster_0*** | Tikrasis ***cluster_1*** | Spėjimo tikslumas |
|---|---|---|---|
| Nuspėtas ***cluster_0*** | 572 | 290 | 66.36 % |
| Nuspėtas ***cluster_1*** | 138 | 428 | 75.62 % |
| Klasės tikslumas | 80.56 % | 59.61 % | |

Greitojo didelės ribos (angl. *Fast Large Margin*) metodo rezultatai (žr, 5.5 lentelė) rodo, jog klasifikatorius sugebėjo pateikti 70,10 % taiklumą su 1,30 % paklaida. Tai antras geriausias rezultatas iš visų šešių naudotų klasifikatorių. Tačiau plotas po ROC kreive siekia 77,10 %, o tai yra prasčiau nei prieš tai apžvelgtas bendro linijinio modelio metodas.

***5.5 lentelė.*** *Greitojo didelės ribos klasifikatoriaus našumas*

| **Kriterijus** | **Reikšmė** | **Standartinis nuokrypis** |
|---|---|---|
| Taiklumas | 70.1 % | ± 1.3 % |
| Klasifikavimo klaida | 29.9 % | ± 1.3 % |
| AUC | 77.1 % | ± 2.1 % |
| Tikslumas | 73.6 % | ± 4.2 % |
| Teisingai teigiamų įverčių rodiklis | 63.4 % | ± 2.9 % |
| F-matas | 68.0 % | ± 2.1 % |
| Jautrumas | 63.4 % | ± 2.9 % |
| Specifiškumas | 77.0 % | ± 3.8 % |

5.6 lentelė atvaizduojama painiavos matrica rodo, jog spėjimo tikslumai tarp nerekomenduojamų (*cluster_0*) ir rekomenduojamų straipsnių (*cluster_1*) yra santykinai apylygiai (atitinkamai 80,56 % ir 59,61 %).

***5.6 lentelė.*** *Greitojo didelės ribos klasifikatoriaus painiavos matrica*

| | Tikrasis ***cluster_0*** | Tikrasis ***cluster_1*** | Spėjimo tikslumas |
|---|---|---|---|
| Nuspėtas ***cluster_0*** | 546 | 263 | 67.49 % |
| Nuspėtas ***cluster_1*** | 164 | 455 | 73.51 % |
| Klasės tikslumas | 76.90 % | 63.37 % | |

Sprendimo medžių (angl. *Decision Tree*) metodo rezultatai (žr, 5.7 lentelė) rodo, jog klasifikatorius sugebėjo pateikti 65,20 % taiklumą su 1,20 % paklaida. Plotas po ROC kreive siekia 64,90 %. Kadangi taiklumas yra mažesnis už kelis kitus klasifikatorius, galima teigti, kad sprendimo medžių metodas nėra tinkamas lietuviškų tekstų klasifikacijai.

***5.7 lentelė.*** *Sprendimo medžių klasifikatoriaus našumas*

| **Kriterijus** | **Reikšmė** | **Standartinis nuokrypis** |
|---|---|---|
| Taiklumas | 65.2 % | ± 1.2 % |
| Klasifikavimo klaida | 34.8 % | ± 1.2 % |
| AUC | 64.9 % | ± 1.2 % |
| Tikslumas | 61.0 % | ± 2.3 % |
| Teisingai teigiamų įverčių rodiklis | 84.8 % | ± 2.8 % |
| F-matas | 70.9 % | ± 1.5 % |
| Jautrumas | 84.8 % | ± 2.8 % |
| Specifiškumas | 45.7 % | ± 1.0 % |

5.8 lentelė atvaizduojama painiavos matrica rodo, jog klasifikatorius labai prastai spėliojo nerekomenduojamų straipsnių (*cluster_0*) įrašus (45,66 %), tačiau pasirodė labai gerai spėliodamas rekomenduojamus straipsnius (*cluster_1*) (84,76 %).

***5.8 lentelė.*** *Sprendimo medžių klasifikatoriaus painiavos matrica*

| | Tikrasis ***cluster_0*** | Tikrasis ***cluster_1*** | Spėjimo tikslumas |
|---|---|---|---|
| Nuspėtas ***cluster_0*** | 326 | 109 | 74.94 % |
| Nuspėtas ***cluster_1*** | 388 | 606 | 60.97 % |
| Klasės tikslumas | 45.66 % | 84.76 % | |

Atsitiktinių medžių (angl. *Random Forest*) metodo rezultatai (žr, 5.9 lentelė) rodo, jog klasifikatorius sugebėjo pateikti net 72,40 % taiklumą su 1,10 % paklaida. Tai pats geriausias rezultatas iš visų šešių naudotų klasifikatorių. Plotas po ROC kreive siekia net 78,50 %, o tai reiškia, jog klasifikatorius atsitiktinai pasirinktą įrašą dažniausiai įvertins tinkamai.

***5.9 lentelė.*** *Atsitiktinių medžių rinkinio klasifikatoriaus našumas*

| **Kriterijus** | **Reikšmė** | **Standartinis nuokrypis** |
|---|---|---|
| Taiklumas | 72.4 % | ± 1.1 % |
| Klasifikavimo klaida | 27.6 % | ± 1.1 % |
| AUC | 78.5 % | ± 1.3 % |
| Tikslumas | 71.1 % | ± 2.5 % |
| Teisingai teigiamų įverčių rodiklis | 76.2 % | ± 1.7 % |
| F-matas | 73.6 % | ± 1.5 % |
| Jautrumas | 76.2 % | ± 1.7 % |
| Specifiškumas | 68.5 % | ± 1.5 % |

5.10 lentelė atvaizduojama painiavos matrica rodo, jog klasifikatorius spėjimus tarp nerekomenduojamų straipsnių (*cluster_0*) ir rekomenduojamų straipsnių (*cluster_1*) atliko apylygiai (atitinkamai 73,78 % ir 71,15 %).

***5.10 lentelė.*** *Atsitiktinių medžių rinkinio klasifikatoriaus painiavos matrica*

| | Tikrasis ***cluster_0*** | Tikrasis ***cluster_1*** | Spėjimo tikslumas |
|---|---|---|---|
| Nuspėtas ***cluster_0*** | 484 | 172 | 73.78 % |
| Nuspėtas ***cluster_1*** | 223 | 550 | 71.15 % |
| Klasės tikslumas | 68.46 % | 76.18 % | |

Palaipsniui išpūstų medžių (angl. *Gradient Boosted Trees*) metodo rezultatai (žr, 5.11 lentelė) rodo, jog klasifikatorius sugebėjo pateikti vos 49,90 % taiklumą su 1,30 % paklaida. Tai pats prasčiausias rezultatas iš visų pritaikytų klasifikatorių.

***5.11 lentelė.*** *Palaipsniui išpūstų medžių klasifikatoriaus našumas*

| **Kriterijus** | **Reikšmė** | **Standartinis nuokrypis** |
|---|---|---|
| Taiklumas | 49.9 % | ± 1.3 % |
| Klasifikavimo klaida | 50.1 % | ± 1.3 % |
| AUC | 91.3 % | ± 1.6 % |
| Tikslumas | 49.9 % | ± 1.3 % |
| Teisingai teigiamų įverčių rodiklis | 100.0 % | ± 0.0 % |
| F-matas | 66.5 % | ± 1.1 % |
| Jautrumas | 100.0 % | ± 0.0 % |
| Specifiškumas | 0.0 % | ± 0.0 % |

5.12 lentelė atvaizduojama painiavos matrica rodo, jog klasifikatorius visiškai visus spėjimus atliko taip, lyg kiekvienas įrašas priklauso rekomenduojamiems straipsniams (*cluster_1*). Metodas visiškai nesugebėjo rasti lietuviškuose tekstuose esančių požymių.

***5.12 lentelė.*** *Palaipsniui išpūstų medžių klasifikatoriaus painiavos matrica*

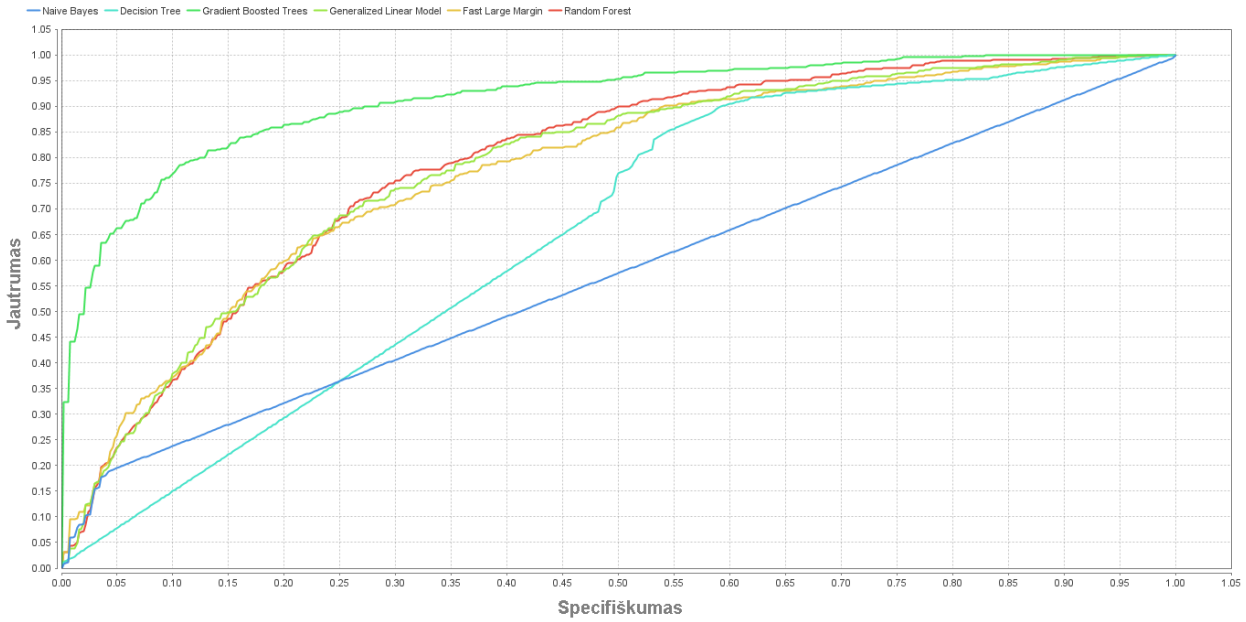
| | Tikrasis ***cluster_0*** | Tikrasis ***cluster_1*** | Spėjimo tikslumas |
|---|---|---|---|
| Nuspėtas ***cluster_0*** | 0 | 0 | 0.00 % |
| Nuspėtas ***cluster_1*** | 716 | 712 | 49.86 % |
| Klasės tikslumas | 0.00 % | 100.00 % | |

### 5.3. **Klasifikacijos rezultatų analizė**

Klasifikacijos rezultatai yra pateikti ROC diagramoje (žr. 5.2 pav.). Klasifikavimo metodų reitingas pagal lietuviškų tekstų duomenų rinkinio klasifikavimo tikslumą (nuo tiksliausios, iki mažiausiai tikslios):

1. Atsitiktinių medžių metodas (angl. *Random Forest*). Tikslumas: 72.40 %;
2. Greitasis didelės ribos metodas (angl. *Fast Large Margin*). Tikslumas: 70.10 %;
3. Bendro linijinio modelio metodas (angl. *Generalized Linear Model*). Tikslumas: 70.00 %;
4. Sprendimo medžių metodas (angl. *Decision Tree*). Tikslumas: 65.20 %;

5. Naivaus Bajeso metodas (angl. *Naive Bayes*). Tikslumas: 51.20 %;
6. Palaipsniui išpūstų medžių metodas (angl. *Gradient Boosted Trees*). Tikslumas: 49.90 %.




***5.2 pav.*** *Klasifikacijos metodų rezultatų ROC kreivė*

Nors tikslumas leidžia sudaryti išvadas, kurie klasifikatoriai labiausiai tinka lietuviškų tekstų klasifikacijos uždaviniams spręsti, tačiau labai svarbi ir klasifikavimo trukmė, kuri dažnai gali lemti pasirenkamo klasifikacijos metodo pasirinkimą. Klasifikavimo metodų reitingas pagal klasifikavimo trukmę (nuo trumpiausios, iki ilgiausios):

1. Bendro linijinio modelio metodas (angl. Generalized Linear Model). Trukmė: 1 minutė 41 sekundė;
2. Sprendimo medžių metodas (angl. Decision Tree). Trukmė: 1 minutė 45 sekundės;
3. Naivaus Bajeso metodas (angl. Naive Bayes). Trukmė: 1 minutė 49 sekundės;
4. Greitasis didelės ribos metodas (angl. Fast Large Margin). Trukmė: 2 minutės 2 sekundės;
5. Atsitiktinių medžių metodas (angl. Random Forest). Trukmė: 8 minutės 52 sekundės;
6. Palaipsniui išpūstų medžių metodas (angl. Gradient Boosted Trees). Trukmė: 20 minučių 3 sekundės.

Išvados, pasiektos įgyvendinus paskutinį tyrimo etapą ir palyginus skirtingų klasifikatorių našumo duomenis:

1. Pasiektas 72.40 % lietuviškų tekstų klasifikacijos tikslumas įrodo, kad tokio tipo uždavinius tikrai galima atlikti ir naudojant lietuvių kalbos tekstus, kurie gramatiškai yra daug sudėtingesni už, pavyzdžiui, anglų kalbos tekstus.
2. Nors tiksliausias klasifikatorius buvo atsitiktinių medžių metodas, jis užėmė priešpaskutinę vietą greičiausiai veikiančių klasifikatorių reitinge, todėl norint taikyti lietuviškų tekstų klasifikacija praktikoje, būtina atsižvelgti į kompiuterinius ir laiko resursus.
3. Didžiausią įtaką teigiamiems tyrimo rezultatams padarė tinkamai atliktos gerosios duomenų gavybos praktikos, skirtos teksto apdorojimui (teksto suskaidymas į atskirus žodžius, jų šaknų išgavimas ir standartizacija). Be šių metodų pritaikymo kiekviena žodžių galūnių variacija sumažintų žodžių svorius vykdant klasifikacijos algoritmus, todėl klasifikatoriams būtų sudėtinga atrasti reikšmingus tekstų požymius.

## Išvados

Magistro baigiamojo darbo rašymo metu suformuluotos šios bendrosios išvados:

1. Rekomendacinės sistemos yra vienos svarbiausių šiuolaikinių programų, interneto svetainių ir vartotojų dažnai naudojamų sistemų sudedamoji dalių, todėl konkretiems atvejams yra labai svarbu ištirti tinkamiausius tai sričiai duomenų klasifikacijos metodus, kurie galėtų naudotojams teikti pritaikytuosius pasiūlymus, tokiu būdu taupant naudotojų laiką ir didinant sistemų praktiškumą.
2. Sprendžiant teksto klasifikacijos uždavinius labai didelę įtaką turi natūralios kalbos apdorojimo taikymas, ypač jeigu tiriama kalba yra tokia sudėtinga, kaip lietuvių kalba, todėl reikia pasirinkti tinkamus duomenų gavybos išankstinio apdorojimo metodus, nes klasifikatoriai pateikia geriausius rezultatus tada, kai tekstai yra apdoroti, o teksto dalys standartizuotos.
3. Teksto klasifikacijos tyrimas atliekamas sklandžiai tuomet, kai ankstesnės tyrimo dalies rezultatai yra iš karto tinkami naudoti tolesnės dalies įvesčiai, vadinasi tyrimo metodika privalo būti suformuluota tyrimo pradžioje, kad nekiltų papildomų darbų, trikdančių tikslinių tyrimo dalių veikimą.
4. Išanalizavus duomenų rinkinio įrašų klasterizavimo rezultatus ir jų statistinius duomenis, nustatyta, kad klasterizavimas atliktas teisingai, nes yra naudojami tinkami klasterizavimo metodai, jų konfigūracijos nustatymai ir pasirinkti tinkami straipsnių įrašus apibūdinantys atributai, vadinasi šio tyrimo etapo išvestis yra tinkama klasifikacijos uždaviniui realizuoti.
5. Tyrimo metu atlikta klasifikacija yra sėkminga, nes iš taikytų klasifikatorių išsiskyręs atsitiktinių medžių metodas pateikė aukšto tikslumo rezultatus ir sugebėjo skaitytojui sugeneruoti pritaikytus pasiūlymus pagal duomenų rinkinyje esančius tekstus, vadinasi lietuviškų tekstų klasifikacija yra įmanoma, jei yra pasirenkami tinkami metodai, jos vykdymui.

## Literatūros sąrašas


1. Ricci, F., Rokach, L., Shapira, B. and Kantor, P. B. (2011). Recommender Systems Handbook. Boston: Springer Science.
2. Rokach, L. and Maimon, O. (2005). Data mining and knowledge discovery handbook: Clustering Methods. Tel-Aviv: Springer Science.
3. Patel, V. R. and Mehta, R. G. (2011). Performance analysis of MK-means clustering algorithm with normalization approach. In *2011 World Congress on Information and Communication Technologies*. Mumbai: IEEE.
4. Deng, L. and Liu, Y. (2018). Deep Learning in Natural Language Processing. New York: Springer.
5. Liddy, E.D. (2001). Natural Language Processing. In *Encyclopedia of Library and Information Science (2nd Ed)*. New York: Marcel Decker, Inc.
6. Vijayarani, S. and Janani, R. (2016). Text Mining: Open Source Tokenization Tools - An Analysis. *Advanced Computational Intelligence: An International Journal*, 3(1), 37-47.
7. Ramasubramanian, C. and Ramya, R. (2013). Effective Pre-Processing Activities in Text Mining using Improved Porter's Stemming Algorithm. *International Journal of Advanced Research in Computer and Communication Engineering*, 2(12), 4536-4538.
8. Jivani, A. G. (2011). A Comparative Study of Stemming Algorithms. *International Journal of Computer Applications Technology and Research*, 2(6), 1930-1938.
9. Stinis, P. (2020). Enforcing Constraints for Time Series Prediction in Supervised, Unsupervised and Reinforcement Learning. Washington: Advanced Computing.
10. Khurana, D., Koli, A., Khatter, K. and Singh, S. (2017). Natural Language Processing: State of The Art, Current Trends and Challenges. Faridabad: Manav Rachna International University.
11. Sisodia, D., Singh, L., Sisodia, S. and Saxena, K. (2012). Clustering Techniques: A Brief Survey of Different Clustering Algorithms. New Delhi: IJLTET.
12. Han, J., Kamber, M. and Pei, J. (2011). Data Mining: Concepts and Techniques. Amsterdam: Elsevier Inc.
13. Pujari, A. K. (2013). Data Mining Techniques. Hyderabad: Universities Press.
14. Nasibov, E. and Ulutagay, G. (2009). Robustness of density-based clustering methods with various neighborhood relations. Izmir: Elsevier.
15. Karypis, G., Han, E. S. and Kumar, V. (1999). Chameleon: A Hierarchical Clustering Algorithm Using Dynamic Modeling. Minneapolis: IEEE Computer.
16. Sundar, N. A., Latha, P. P. and Chandra, M. R. (2012). Performance Analysis of Classification Data Mining. Techniques Over Heart Disease Data Base. Rajam: IJESAT.

17. Rish, I. (2001). An empirical study of the naive Bayes classifier. New York State: IJCAI.
18. Srinath, K. R. (2017). An Overview of Web Content Mining Techniques. *International Research Journal of Engineering and Technology*, 4(11), 1858-1861.
19. Sharma, A. K. and Gupta, P. C. (2012). Study & Analysis of Web Content Mining Tools to Improve Techniques of WebData Mining. *International Journal of Advanced Research in Computer Engineering & Technology*, 1(8), 287-293.
20. Hussein, M.-K. and Mousa, M.-H. (2010). An Effective Web Mining Algorithm using Link Analysis. *International Journal of Computer Science and Information Technologies*, 1(3), 190-197.
21. Srivastava, J., Desikan, P. and Kumar, V. (2005). Web mining–concepts, applications and research directions. Minnesota: University of Minnesota.
22. Sharma, K., Shrivastava, G. and Kumar, V. (2011). Web mining: Today and Tomorrow. *Institute of Electrical and Electronics Engineers*. Chennai: IEEE.
23. Vozalis, E. and Margaritis, K. G. (2003). Analysis of recommender systems' algorithms. *The 6th Hellenic European Conference on Computer Mathematics & its Applications*. Athens: HERCMA.
24. Briscoe, T. (2013). Introduction to Linguistics for Natural Language Processing. JAV: University of Cambridge.
25. Kapočiūtė-Dzikienė, J., Damaševičius, R. ir Wozniak, M. (2019). Sentiment Analysis of Lithuanian Texts Using Traditional and Deep Learning Approaches. Computers.
26. Kiaunė, K. ir Ramanauskaitė, S. (2019). Didelį kategorijų kieki turinčių draudimo bendrovės klientų užklausų, gautų elektroniniais laiškais, lietuviško teksto klasifikavimas. *Jaunųjų mokslininkų darbai*, 49(2), 52-59.