



VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
FUNDAMENTINIŲ MOKSLŲ FAKULTETAS
INFORMACINIŲ SISTEMŲ KATEDRA

Milda Vencevičiūtė

**MAŠININIO MOKYMOSI ALGORITMAIS GRINDŽIAMŲ
FILMŲ REKOMENDACINIŲ SISTEMŲ TYRIMAS
RESEARCH OF MOVIE RECOMMENDATION SYSTEMS
BASED ON MACHINE LEARNING ALGORITHMS**

Baigiamasis magistro darbas

Informacinių sistemų programų inžinerijos studijų programa
valstybinis kodas 6211BX017
Informatikos inžinerijos studijų kryptis

Vadovas prof. dr. Diana Kalibatienė

Vilnius, 2023

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

FUNDAMENTINIŲ MOKSLŲ FAKULTETAS

INFORMACINIŲ SISTEMŲ KATEDRA

Milda Vencevičiūtė

**MAŠININIO MOKYMOSI ALGORITMAIS GRINDŽIAMŲ FILMŲ
REKOMENDACINIŲ SISTEMŲ TYRIMAS**
**RESEARCH OF MOVIE RECOMMENDATION SYSTEMS BASED ON
MACHINE LEARNING ALGORITHMS**

Baigiamasis magistro darbas

Informacinių sistemų programų inžinerijos studijų programa

valstybinis kodas 6211BX017

Informatikos inžinerijos studijų kryptis

Vadovas

prof. dr. Diana Kalibatienė
(Moksl. laipsnis/pedag. vardas, vardas, pavardė)

(Parašas)

(Data)

Lietuvių kalbos konsultantas

dr. Vaida Buivydienė
(Moksl. laipsnis/pedag. vardas, vardas, pavardė)

(Parašas)

(Data)

Vilnius, 2023

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
FUNDAMENTINIŲ MOKSLŲ FAKULTETAS
INFORMACINIŲ SISTEMŲ KATEDRA

Studijų kryptis: Informatikos inžinerija

Studijų programa: Informacinių sistemų programų inžinerija , valstybinis kodas
6211BX017

Specializacija: Informacinių sistemų programų inžinerija

TVIRTINU

Katedros vedėjas
Nikolaj Goranin

2023-05-27

MAGISTRO BAIGIAMOJO DARBO UŽDUOTIS

Nr. ISIfm-21–3934

Vilnius

Studentas (-ė): Milda Vencevičiūtė

Baigiamojo darbo tema: Mašininio mokymosi algoritmais grindžiamų filmų rekomendacinių sistemų tyrimas Baigiamojo darbo užbaigimo terminas pagal numatytą studijų kalendorinį grafiką.

BAIGIAMOJO DARBO UŽDUOTIS:

Tikslas: Pagerinti rekomendacinės sistemos tipo pasirinkimo procesą, siūlant rekomendacinių sistemų kūrimo, išbandymo bei vertinimo metodą.

Uždaviniai:

1. Išanalizuoti skirtingų tipų rekomendacines sistemas bei jose taikomus mašininio mokymosi metodus bei algoritmus.
2. Pasiūlyti metodą, leidžiantį palyginti rekomendacines sistemas, grindžiamas skirtingais mašininio mokymosi algoritmais.
3. Išbandyti pasiūlytą metodą, įgyvendinant bent tris skirtingo tipo algoritmais grindžiamas rekomendacines sistemas, ir pasiūlyti jų pasirinkimo rekomendacijas.

Planuojamas rezultatas:

1. Išanalizuotos ir palygintos rekomendacinės sistemos, grindžiamos skirtingo tipo algoritmais, pagal apibrėžtus kriterijus.
2. Pasiūlytas metodas, leidžiantis palyginti rekomendacines sistemas, grindžiamas skirtingais mašininio mokymosi algoritmais. Siūlomas metodas aprašytas BPMN diagramomis.
3. Pasiūlytas metodas įgyvendintas ir išbandytas bent su trimis skirtingo tipo algoritmais grindžiamomis rekomendacinėmis sistemomis, ir pasiūlytos rekomendacinių sistemų pasirinkimo rekomendacijos.

Vadovas profesorius Diana Kalibatienė

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

	Milda Vencevičiūtė, 20210874	
	(Studento vardas ir pavardė, studento pažymėjimo Nr.)	
	Fundamentinių mokslų fakultetas	
	(Fakultetas)	
	Informacinių sistemų programų inžinerija, ISIfm-21	
	(Studijų programa, akademinė grupė)	

BAIGIAMOJO DARBO (PROJEKTO)

SAŽININGUMO DEKLARACIJA

2023 m. gegužės 28 d.

Patvirtinu, kad mano baigiamasis darbas tema „Mašininio mokymosi algoritmais grindžiamų filmų rekomendacinių sistemų tyrimas“ yra savarankiškai parašytas. Šiame darbe pateikta medžiaga nėra plagijuota.

Tiesiogiai ar netiesiogiai panaudotos kitų šaltinių citatos pažymėtos literatūros nuorodose.

Mano darbo vadovas profesorius daktaras Diana Kalibatienė.

Kitų asmenų indėlio į parengtą baigiamąjį darbą nėra. Jokių įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs (-usi).

			Milda Vencevičiūtė
	(Parašas)		(Vardas ir pavardė)

Vilniaus Gedimino technikos universitetas
Fundamentinių mokslų fakultetas
Informacinių sistemų katedra

ISBN ISSN
Egz. sk.
Data-....-....

Informacinių sistemų programų inžinerijos studijų programos baigiamasis magistro darbas
Pavadinimas **Mašininio mokymosi algoritmais grindžiamų filmų rekomendacinių sistemų tyrimas**

Autorius **Milda Vencevičiūtė**

Vadovas prof. dr. **Diana Kalibatienė**

Kalba: lietuvių

Anotacija

Baigiamajame magistro darbe nagrinėjama rekomendacinių sistemų sudarymo tema. Skirtingiems rekomendacinių sistemų tipams įgyvendinti pasirinkti mašininio mokymosi algoritmai: k-artimiausio kaimyno, k-reikšmių klasterizavimo bei neuroninių tinklų. Sudarytos filmų rekomendacinės sistemos buvo apmokomos bei išbandomos „MovieLens“ duomenų baze. Rekomendacinių sistemų tikslumas įvertintas vidutinės absoliučios paklaidos (MAE) bei šaknies iš vidutinės kvadratinės paklaidos (RMSE) metrikomis. Taip pat rekomendacinių sistemų modeliai buvo palyginti apmokymo trukmės įvertinimu. Tyrimo metu nustatyta, kad bendrumo filtravimu paremta rekomendacinė sistema yra efektyviausia, dėl žemo MAE ir RMSE įverčio, taip pat tokio modelio apmokymo trukmė yra trumpiausia ir įvesties duomenų kiekis yra mažiausias.

Darbą sudaro 7 dalys: įvadas, rekomendacinių sistemų tipų analizė, mašininio mokymosi algoritmų taikymo rekomendacinėse sistemose analizė, filmų rekomendacinių sistemų projektavimas, rekomendacinių sistemų įgyvendinimas ir testavimas, išvados ir siūlymai, literatūros sąrašas.

Darbo apimtis – 55 p. teksto be priedų, 16 iliustr., 8 lent., 33 bibliografiniai šaltiniai.

Prasminiai žodžiai: rekomendacinė sistema, mašininio mokymosi algoritmai, neuroniniai tinklai, rekomendacinių sistemų palyginimas

Vilnius Gediminas Technical University
Faculty of Fundamental Sciences
Department of Information Systems

ISBN ISSN
Copies No.
Date-....-....

Information Systems Software Engineering study programme Master Graduation Thesis
Title **Research of Movie Recommendation Systems Based on Machine Learning Algorithms**

Author **Milda Vencevičiūtė**

Academic supervisor prof. dr. **Diana Kalibatiėnė**

Thesis language: Lithuanian

Annotation

The final master's thesis describes the topic of recommendation systems. For different types of movie recommendation systems implementation there were chosen three different machine learning algorithms: k-Nearest Neighbor, k-Means Clustering and neural networks. Implemented movie recommendation systems were trained and tested based on „MovieLens“ dataset. The accuracy and efficiency of movie recommendation systems were measured and compared by Mean Absolute Error (MAE), Root Mean Absolute Error (RMSE) metrics and evaluation of model's training time. Based on the results, the collaborative filtering based movie recommendation system was selected as most efficient because of the low MAE, RMSE, training time evaluations and also required input data is very small.

Thesis consists of 7 parts: introduction, analysis of recommendation system types, analysis of machine learning algorithms used in recommendation systems, design of movie recommendation systems, implementation and testing of recommendation systems, conclusions, references.

Thesis consists of 55 p. text without appendixes, 16 pictures, 8 tables, 33 bibliographical entries.

Prasminiai žodžiai: recommendation system, machine learning algorithms, neural networks, comparison of recommendation systems

Turinys

Paveikslų sąrašas	8
Lentelių sąrašas.....	9
Santrumpos	10
1 Įvadas	11
2 Rekomendacinės sistemos, paremtos naudotojo elgsenos analize	14
2.1 Naudotojo elgsenos analizė	14
2.2 Rekomendacinės sistemos	14
2.3 Rekomendacinių sistemų tipai	15
2.4 Rekomendacinių sistemų palyginimas	16
2.5 Gauti rekomendacinių sistemų analizės rezultatai	18
3 Mašininio mokymosi metodai taikomi rekomendacinėse sistemose	20
3.1 Mašininis mokymasis	20
3.2 K-artimiausio kaimyno metodas	21
3.3 K-reikšmių klasterizavimas	22
3.4 Neuroniniai tinklai.....	23
3.5 Kosinuso panašumas	24
3.6 Gauti mašininio mokymosi metodų rezultatai.....	25
4 Filmų rekomendacinių sistemų projektavimas.....	26
4.1 Duomenų surinkimas.....	27
4.2 Duomenų apdorojimas	27
4.3 Duomenų analizė.....	28
4.3.1 Rekomendacinės sistemos paremtos duomenų analizės metodais	34
4.4 Siūlomas rekomendacinės sistemos kūrimo ir išbandymo modelis	35
4.5 Gauti projektavimo rezultatai.....	38
5 Rekomendacinių sistemų įgyvendinimas ir išbandymas.....	39
5.1 Modelių vertinimo struktūra.....	39
5.1.1 Vidutinė absoliuti paklaida.....	39
5.1.2 Šaknis iš vidutinės kvadratinės paklaidos	40
5.2 Bendrumo filtravimu paremta sistema	40
5.3 Turiniu paremta rekomendacinė sistema.....	42
5.4 Hibridinis rekomendacinės sistemos modelis	44
5.5 Modelių išbandymas su sumažinta duomenų baze.....	47
5.6 Sukurtų rekomendacinių sistemų palyginimas.....	49
5.7 Gauti rekomendacinių sistemų įgyvendinimo bei testavimo rezultatai.....	50
6 Išvados ir siūlymai	52
Literatūros sąrašas	53

Paveikslų sąrašas

2.1 pav. Rekomendacinių sistemų tipai (pagal Isinkaye et al., 2015)	16
3.1 pav. K-artimiausio kaimyno algoritmas (pagal Chomboon et al., 2015)	22
3.2 pav. Neuroninių tinklų modelio struktūra (pagal Tanty Tanweer Desmukh, 2015).....	24
3.3 pav. Vektorių kosinuso panašumas	25
4.1 pav. Tyrimo procesų diagrama	26
4.2 pav. Naudotojų amžiaus pasiskirstymas.....	29
4.3 pav. Filmų žanrų pasiskirstymas	30
4.4 pav. Įvertinimų skaičius pagal įvertį	31
4.5 pav. Spearman koreliacija tarp filmų žanrų ir įvertinimų vidurkio	32
4.6 pav. Šilumos grafikas, atvaizduojantis koreliacijos koeficientus tarp skirtingų filmo žanrų ir jų įvertinimų vidurkio.....	33
4.7 pav. „Star wars“ koreliacija su kitais filmais	35
4.8 pav. Koreliacija, nustačius įvertinimų kiekį	35
4.9 pav. RS komponentų diagrama	36
4.10 pav. Rekomendacinės sistemos kūrimo ir išbandymo procesų diagrama	37
5.1 pav. Filmų pasiskirstymas grupėse.....	43
5.2 pav. Neuroninių tinklų modelio nuostoliai.....	47

Lentelių sąrašas

2.1 lentelė. Rekomendacinių sistemų palyginimas	19
3.1 lentelė. RS naudojamų mašininio mokymosi algoritmų palyginimas	25
4.1 lentelė. Duomenų struktūra	27
4.2 lentelė. Daugiausiai įvertinimų turintys filmai	34
4.3 lentelė. Komponentų aprašymas	36
5.1 lentelė. Modelių išbandymas su sumažinta duomenų baze	48
5.2 lentelė. Rekomendacinių sistemų modelių palyginimas	49
5.3 lentelė. Bendra rekomendacinių sistemų įvertinimo lentelė.....	50

Santrumpos

RS (*angl. recommendation system*) – rekomendacinė sistema.

ML (*angl. machine learning*) – mašininis mokymasis.

KNN (*angl. k-Nearest Neighbor*) – k-artimiausio kaimyno algoritmas.

RMSE (*angl. Root mean squared error*) – vidutinis kvadratinis nuokrypis.

MAE (*angl. Mean absolute error*) – vidutinė absoliuti paklaida.

DB – duomenų bazė.

1 Įvadas

Tobulėjant informacinėms technologijoms, vis didesnė dalis prekių ir paslaugų teikėjų ima perkelti savo veiklą į virtualią erdvę, kurioje gali pasiekti įvairesnę ir platesnę klientų auditoriją. Tačiau prekių bei paslaugų paklausa ir pasiūla su laiku sparčiai auga, o tai verslo atstovams tampa dideliu iššūkiu, kaip pritraukti pirkėjus, juos sudominti naujomis prekėmis, padidinti pardavimų kiekius ir pan. Šiuos verslo tikslus įgyvendinti gali palengvinti naudotojo elgsenos analizė. Naudotojo elgsenos analizė yra procesas, kurio metu renkami ir analizuojami naudotojo duomenys tam, kad suprasti kaip ir kodėl naudotojas sąveikauja su sistema. Šiai duomenų analizei gali būti taikoma viena iš efektyviausių ir sparčiausiai tobulėjančių mokslo sričių – mašininis mokymasis – kurios algoritmais būtų galima įgyvendinti rekomendacines sistemas. Rekomendacinės sistemos padėtų suprasti bei prognozuoti naudotojo elgesį ir tokiu būdu rasti verslo tikslinę auditoriją, perprasti bei individualizuoti kliento poreikius, pasiūlyti labiau pagal poreikius atitinkančias prekes ar paslaugas. Tačiau mašininio mokymosi algoritmų yra ganėtinai daug, o ir verslo atstovų saugomi duomenys apie klientus gali skirtis, todėl yra svarbu suprasti, kokių algoritmų bei duomenų panaudojimas suteiktų didžiausią naudą verslo atstovams, įgyvendinant rekomendacines sistemas bei siekiant savo klientams pasiūlyti kuo labiau jų poreikius atitinkančias prekes ar paslaugas.

Tyrimo objektas – skirtingų tipų rekomendacinės sistemos.

Darbo tikslas – pagerinti rekomendacinės sistemos tipo pasirinkimo procesą, siūlant rekomendacinių sistemų kūrimo, išbandymo bei vertinimo metodą, kuris padėtų nustatyti efektyviausią rekomendacinį modelį.

Darbo uždaviniai:

1. Išanalizuoti skirtingų tipų rekomendacines sistemas bei jose taikomus mašininio mokymosi metodus.
2. Pasiūlyti metodą, leidžiantį įgyvendinti ir palyginti skirtingo tipo rekomendacines sistemas, grindžiamas skirtingais mašininio mokymosi algoritmais.
3. Išbandyti ir palyginti pagal pasiūlytą metodą sudarytas skirtingo tipo rekomendacines sistemas.

Temos naujumas – mašininis mokymasis yra nuolat tobulėjanti ir plačiai taikoma mokslo sritis. Šiais laikais yra gausybė įvairių mašininio mokymosi metodų, jų modifikacijų bei kombinacijų, naudojamų rekomendacinėse sistemose. Tačiau vartotojų ir verslo poreikiai,

duomenų gavimo būdai ar rekomendacinių sistemų savybės dažnai keičiasi ir tam tikrais atvejais yra sudėtinga žinoti ar nustatyti, koks mašininio mokymosi metodas yra tinkamiausias rekomendacijoms pateikti. Todėl yra naudinga apžvelgti šiais laikais taikomus mašininio mokymosi algoritmus, suprasti kaip jie veikia ir kaip juos taikyti, kad pasiekti geriausių rezultatų.

Temos aktualumas – praėjus pasaulį sukausčiusio Covid-19 viruso bangai, galima pastebėti, kad didžioji dalis parduotuvių persikėlė į internetinę prekybą, todėl verslo atstovai yra priversti domėtis ir ieškoti būdų, kaip išsiskirti iš rinkos, atrasti savo pirkėją ir padidinti pardavimų kiekį, todėl turi būti kuriamos įvairesnės, skirtingas verslo sritis analizuojančios, rekomendacinės sistemos ir jose taikomi metodai turi būti apmokomi ir išbandyti su kuo naujesniais duomenų rinkiniais.

Tyrimo metodika – analitinėje darbo dalyje buvo naudojama su tema susijusių mokslinių straipsnių analizė bei jų palyginimas.

Mokslinė darbo vertė – pasiūlytos, išbandytos ir palygintos skirtingo tipo rekomendacinės sistemos bei sudarytas palyginimo metodas, kuriuo būtų galima sklandžiau pasirinkti efektyviausią rekomendacinę sistemą.

Darbo rezultatai:

1. Atlikta mokslinių literatūros straipsnių, susijusių su naudotojo elgsenos analize rekomendacinėse sistemose bei jose taikomų mašininio mokymosi metodų, analizė.
2. Suprojektuota bendroji rekomendacinės sistemos kūrimo ir išbandymo eiga.
3. Sukurti trys skirtingo tipo rekomendacinės sistemos:
 - Bendrumo filtravimu paremta sistema, pagrįsta k-artimiausių kaimynų algoritmu;
 - Turiniu paremta sistema, pagrįsta k-reikšmių klasterizavimo algoritmu;
 - Hibridinė sistema, pagrįsta neuroniniais tinklais.
4. Ištestuotos rekomendacinės sistemos, apskaičiuota rekomendacinių sistemų modelių apmokymo trukmė bei tikslumo įverčiai:
 - Apskaičiuotas vidutinės absoliučios paklaidos įvertis;
 - Apskaičiuotas šaknies iš vidutinės kvadratinės paklaidos įvertis.
5. Palyginti gauti rekomendacinių sistemų įvertinimai.

Darbo struktūra:

Darbą sudaro šeši skyriai – įvadas, su tema susijusių literatūros šaltinių analizė, mašininio

mokymosi metodų apžvalga, bendros darbo struktūros nustatymas, rekomendacinės sistemos modeliavimo ir įgyvendinimo bei išvadų ir siūlymų.

2 Rekomendacinės sistemos, paremtos naudotojo elgsenos analize

Šiame skyriuje apibrėžiamos naudotojo elgsenos ir jos analizės sąvokos. Apžvelgiamos verslo sritys, kurioms yra aktualus naudotojo elgsenos tyrimas bei plačiau išanalizuojamos šiuo metu taikomos prekių ir paslaugų rekomendavimo sistemos. Aprašomas mašininis mokymasis ir jo metodai, taikomi naudotojo elgsenos analizėje ir prekių bei paslaugų rekomendacinių sistemų kūrimo procesuose.

2.1 Naudotojo elgsenos analizė

Elgsenos analizės (angl. *behavior analysis*) pradmenys siekia 1930-uosius metus. Šis analizės tipas nuo pat pradžių buvo koncentruotas į žmogaus elgsenos tyrimus, kurie turi glaudų ryšį su psichologijos sritimi – tam tikri mokslo metodai buvo taikomi siekiant geriau suprasti žmogaus elgseną (Cooper et al., 2020). Su laiku, tobulėjant technologijoms ir joms užimant vis didesnę dalį mūsų kasdienio gyvenimo, žmonių elgsenos tyrimai tobulėjo – buvo pradėti taikyti duomenų gavybos metodai, mašininio mokymosi algoritmai. Dabar elgsenos analizei gali būti naudojami tokie terminai, kaip elgsenos informatika (angl. *behavior informatics*), kliento elgsenos analizė (angl. *customer behavior analysis*) arba naudotojo elgsenos analizė (angl. *user behavior analysis*).

Šiai laikais elgesio analizė yra laikoma moksline disciplina, plačiau taikoma informacinių technologijų srityje. Elgsenos analizė padeda geriau suprasti bei analizuoti žmogaus ir sistemos sąveiką, remiantis surinktais duomenimis. Šiuo atveju, informacinių technologijų srityje, žmogaus ir sistemos sąveikos duomenimis gali būti paspaudimai, klaviatūra įvedami žodžiai ar jų fragmentai, naudotojo navigacijos sistemoje sekimo duomenys ir t. t. (Terragni & Hassani, 2018). Tad detali ir tinkamai panaudojama vartotojo elgesio analizė gali vesti į lengviau ir sklandžiau pasiekiamą verslo tikslų įgyvendinimą.

Elgsenos analizė gali būti naudojama daugelyje sričių, tokių kaip: įtartinos sąveikos vidiniuose tinkluose aptikimas, tam tikrų prekių ar paslaugų rekomendacijos klientams, pagalba vyresniems žmonėms atlikti kasdieninius darbus ir pan. Šiame darbe detaliau bus nagrinėjamas vartotojo elgesio analizės naudojimas prekių, produktų bei paslaugų rekomendavimo sistemų srityje.

2.2 Rekomendacinės sistemos

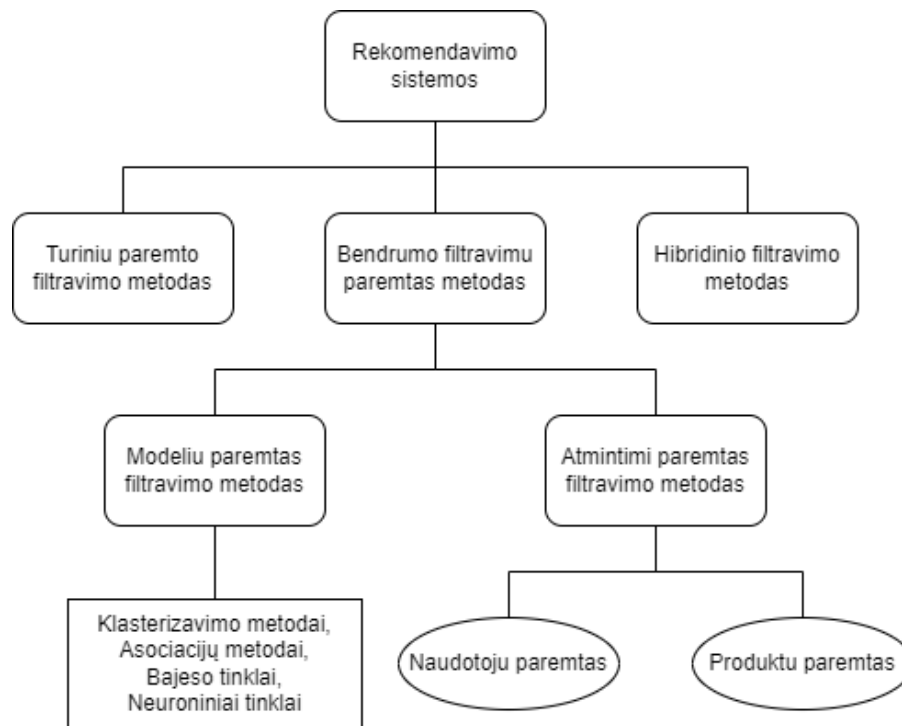
Rekomendacinės sistemos (RS) yra vienos iš svarbiausių didžiųjų duomenų (angl. *big data*) analizės tipų ir yra tapusios viena iš verslo sudedamųjų dalių tokio dydžio kompanijose, kaip „Amazon“, „Facebook“, „Google“ ir kt. (Hwangbo et al., 2018). RS yra skirstomos į dvi kategorijas – suasmenintas ir nesuasmenintas (Sahoo et al., 2017). Nesuasmenintos

rekomendacinės sistemos dažniausiai yra naujų tinklalapių pradinis pasirinkimas. Organizacijos, tik pradedančios verslą, dažnai remiasi tokiomis sistemomis, nes jos yra paprastos ir joms sukurti nereikia duomenų apie naudotojus ar jų poreikius. Nesuasmenintos rekomendacijos yra sudaromos pagal daugiausiai nupirktų produktų kiekius arba pagal prekės populiarumo reitingus. Tokios rekomendacijos yra vadinamos nesuasmenintomis dėl to, nes visiems naudotojams, sąveikaujantiems su tokia rekomendacine sistema, bus pateikiamos vienodos rekomendacijos (Falk, 2019).

Suasmenintos rekomendacinės sistemos yra paremtos asmeniniais vartotojų duomenimis, vartotojų elgsena bei poreikiais. Tokio tipo rekomendacinės sistemos reikalauja įvesties duomenų, tokių kaip: vartotojo vietovės duomenys, amžius, lytis, transakcijų ar vartotojo sąveikos su internetine svetaine duomenys ir pan. Tokio tipo RS gali būti skirstomos į subkategorijas, iš kurių dvi pagrindinės bei dažniausiai naudojamos yra turiniu paremtos (angl. *content-based*) ir bendrumo filtravimo (angl. *collaborative filtering*) rekomendacinės sistemos.

2.3 Rekomendacinių sistemų tipai

Kaip ir minėta ankstesniame poskyryje, suasmenintos rekomendacinės sistemos yra skirstomos į dvi pagrindines kategorijas: turiniu paremtas bei bendrumo filtravimo (žr. 2.1 pav.). Taip pat kai kurios organizacijos, norinčios pagerinti savo produktų rekomendavimą vartotojams ir sukurti kompleksines rekomendacines sistemas naudoja hibridinį būdą joms formuoti, tai reiškia, kad sujungia turiniu bei bendrumo filtravimu paremtų sistemų elementus į vieną – tokios sistemos remiasi asmeniniais vartotojo profilio duomenis bei produktų turinio charakteristikomis (Isinkaye et al., 2015).



2.1 pav. Rekomendacinių sistemų tipai (pagal Isinkaye et al., 2015)

Turiniu paremtos rekomendavimo sistemos atsižvelgia į paties produkto turinį – vartotojui yra rekomenduojamos prekės ar paslaugos, kurių turinys ir jo atributai yra panašūs į prieš tai, to paties vartotojo pirktas, pamėgtas ar peržiūrėtas prekes. Bendrumo filtravimu pagrįstos RS, remiasi vartotojų produktų vertinimais (Shi et al., 2014), sąveika ir modeliuoja tam tikras vartotojų profilio charakteristikas. Vėliau, remiantis sudarytais profilių modeliais, sistema bando išvelgti panašumus tarp skirtingų vartotojų ir jų vartojimo ypatumų, tokiu būdu sistema gali vienam vartotojui rekomenduoti prekę ar paslaugą, kurią įsigijo, peržiūrėjo, pamėgo ar pan. kitas, panašius pomėgius turintis, vartotojas. Bendrumo filtravimo technika gali būti išskiriama į modelių paremtą ir atmintimi paremtą technikas.

Atmintimi pagrįstas bendrumo filtravimas naudoja visus vartotojo ir produkto duomenis tam, kad sugeneruotų tikimybes – sistema naudoja statistinius metodus vartotojų rinkiniui nustatyti, kurie turėtų panašius sąveikos ypatumus su aktyviu, tiksliniu vartotoju (Aditya et al., 2016), paprastai tariant, tokia sistema naudoja tik atmintyje įrašytus duomenis ir rekomenduoja produktą, kuris patiko kitam panašiam vartotojui. Modelių paremtas bendrumo filtravimas yra tokia technika, kurios metu iš turimų duomenų ir pasitelkus mašininio mokymosi algoritmus yra sudaromas modelis. Modelis yra apmokomas turimais duomenimis ir vėliau geba sudaryti bei apskaičiuoti tikimybes ateities duomenims, tai reiškia, kad toks modelis gali įtraukti ir rekomenduoti naujus, nei vienam vartotojui dar nematytus, produktus.

2.4 Rekomendacinių sistemų palyginimas

Didelės kompanijos, tokios kaip „Amazon“ ar „LinkedIn“ dažnai naudoja

rekomendacines sistemas, paremtas bendrumo filtravimu. „Amazon“ analizuoja vartotojų pirktas prekes ir jas lygina su kitomis, panašiomis prekėmis bei jų atsiliepimais, o „LinkedIn“ platformoje, kuri orientuojasi į verslą ir jo santykius, taip pat yra naudojama rekomendacinė sistema, pagrįsta bendrumo filtravimu, „LinkedIn“ rekomendacinė sistema sudaro tam tikrus naudotojų profilius, lygina juos tarpusavyje ir rekomenduoja asmenis, darbo pasiūlymus ar paslaugas paremtas, kitų panašių naudotojų elgsenos platformoje ypatumais. Tokių rekomendacinių sistemų didžiausias trūkumas yra reitingavimo matricos retumas (angl. *sparsity*) bei „šalto paleidimo“ (angl. *cold start*) problemos (Wei et al., 2017). Reitingavimo matricos retumas ar negausumas apibrėžia problemą, kai sistemos rekomendacijos gali būti netikslios, nes yra nepakankamas kiekis vartotojų įvertinimų ar atsiliepimų, kuo iš esmės ir remiasi tokio pobūdžio sistemos. Kita opi problema yra „šaltas paleidimas“, tai reiškia, kad tokia rekomendacinė sistema gali būti itin neefektyvi naujoms prekėms bei paslaugoms ar naujiems vartotojams, dėl įvertinimų ar atsiliepimų duomenų trūkumo (Google Developers: Recommendation Systems, 2020). Taip pat tokioms RS yra sudėtingiau įtraukti pašalinius duomenis, pavyzdžiui, filmų svetainėje yra skirtingo amžiaus ar skirtingose šalyse gyvenančių naudotojų ir tai galėtų dar labiau pagerinti rekomendacijų tikslumą, tačiau rekomendacinėse sistemose, paremtose bendrumo filtravimu, prioritetą yra sumodeliuoti naudotojų profilius pagal jų žiūrėjimus filmus, o ne pagal asmeninius duomenis.

Kitą rekomendacinių sistemų kategoriją, paremtą turiniu, reprezentuoja „Google“ bendrovei priklausanti „Google Play store“ platforma, kurioje yra renkami ir analizuojami su naudotoju susiję duomenys ir pagal mašininio mokymosi algoritmus yra atrenkamos bei siūlomos, pavyzdžiui, mobiliosios programėlės, panašios į prieš tai naudotojo atsisiųstas ar peržiūrėtas programėles – taip sudaromos suasmenintos, naudotojo poreikius ar naudojimo įpročius atitinkančios, rekomendacijos (Gong & Zhernov, 2019). Stambi amerikiečių kompanija „Best Buy“, parduodanti elektroninius įrenginius, taip pat naudoja turiniu paremtą rekomendacinę sistemą. Ši organizacija stebi naudotojų elgseną sistemoje – paieškos, naršymo, transakcijų duomenis – ir paruošia individualias rekomendacijas kiekvienam naudotojui pagal būtent jo elgsenos duomenis. Pažinčių programėlėje „Tinder“, taip pat naudojama turiniu paremta rekomendacinė sistema. Šioje programėlėje yra stebima, analizuojama asmeninė naudotojo informacija, jo lokacija, amžius, kiti profilio duomenys, be to rekomendacijos yra sudaromos pagal tai, kokie kiti programėlės naudotojai patiko ar nepatiko tiksliniam naudotojui. Tokio tipo rekomendacinių sistemų privalumas yra tai, kad nereikia rinkti duomenų apie kitus naudotojus – nereikia analizuoti ar lyginti skirtingų naudotojų duomenų tarpusavyje, todėl tokia sistema yra tinkama didelį naudotojų kiekį turinčioms kompanijoms. Šio tipo RS, priešingai nei bendrumo filtravimo technikos, nesudaro „šalto paleidimo“ problemos, susijusios

su pačiais produktais – sistema gali rekomenduoti visiškai naujus, dar nebandytus, produktus, taip pat turiniu paremtos sistemos yra itin adaptyvios, tai reiškia, kad jos keičiasi ir prisitaiko prie atsinaujinusių tikslinio naudotojo poreikių. Turiniu paremtos rekomendacinės sistemos, nagrinėjančios konkretaus naudotojo domėjimosi ar pirkimo istoriją gali jam pasiūlyti išskirtinių prekių ar paslaugų, kurių, tarkime, kiti naudotojai neieško ir nesidomi, tačiau tai kartu gali būti ir trūkumas, nes naudotojui yra sudėtingiau pasiūlyti naujų, jo pirkimo duomenų neatitinkančių, prekių, todėl tokiai rekomendacinei sistemai bus kiek sunkiau perprasti ir pridėti ar pakeisti rekomenduotinas prekes bei paslaugas. Taip pat tokio tipo sistemoms yra sunku veikti efektyviai, jei naudotojas yra naujas, nes sistemai ir tikslesnėms rekomendacijoms paprasčiausiai trūksta reikiamų duomenų.

Kita didelė kompanija – „Spotify“ – naudoja hibridinio tipo rekomendacinę sistemą, paremtą panašių naudotojų profilių modeliavimu, tačiau, kad rekomendacijos būtų tikslesnės taip pat yra analizuojama garso įrašų charakteristika bei produktų, šiuo atveju dainų, duomenys – atlikėjas, dainos ar albumo pavadinimas (Dilmegani, 2017). Tokiu būdu rekomendacijos yra sudaromos, remiantis skirtingų vartotojų naudojimosi programėle ypatumais bei pačių dainų turiniu. Panašiu principu veikia ir „Netflix“, filmų bei serialų peržiūros sistema. „Netflix“ modeliuoja tam tikrus naudotojų profilius, kurie apibūdina, kokius filmus ir jų žanrus žiūri naudotojai, analizuodami šiuos duomenis „Netflix“ atrenka ir siūlo atitinkamas rekomendacijas, paremtas skirtingų naudotojų profilių duomenimis, be to sistemoje yra naudojami algoritmai, analizuojantys pačių filmų ar serialų turinį ir tokiu būdu sudaro dar tikslesnes rekomendacijas, nes remiasi filmų bei serialų panašumu. „Netflix“ organizacijos sudaryta rekomendacinė sistema internetiniuose šaltiniuose dažnai minima, kaip viena geriausių, tiksliausių, tokio tipo sistemų.

2.5 Gauti rekomendacinių sistemų analizės rezultatai

Rekomendacinių sistemų yra įvairių tipų, tačiau pagrindinis tokios sistemos komponentas yra vartotojo elgsenos analizė. Atlikus literatūros analizę, šiame skyriuje yra pateikiamos pagrindinės sąvokos, aprašomi naudotojo elgsenos analizės principai bei rekomendacinių sistemų tipai ir jų savybės (žr. 2.1 lentelė). Taip pat yra apžvelgiamos ir palyginamos įvairių kompanijų naudojamos rekomendacinės sistemos.

Atlikus literatūrinę analizę ir palyginus skirtingų tipų rekomendacines sistemas ir jų pritaikymą realiose platformose, galima išvelgti, kad kiekviena sistema turi savų privalumų bei trūkumų ir kad pasirinkti, kokio tipo sistemą yra tikslingiau taikyti – reikia atsižvelgti į daugelį kriterijų.

2.1 lentelė. Rekomendacinių sistemų palyginimas

RS tipas	Privalumai	Trūkumai	Sistemų pavyzdžiai
Turiniu paremtas	• Prisitaiko prie konkretaus naudotojo poreikių • Atsižvelgia į produkto turinį • Gali rekomenduoti naujus produktus, kurie dar nėra įvertinti, tačiau atitinka naudotojo poreikius	• Šalto paleidimo problema, kai naudotojas yra naujas • Sunku rekomenduoti kitokias/neįprastas prekes	• Best Buy • Tinder • Google Play
Bendrumo filtravimu paremtas	• Gali teikti įvairias/išskirtines rekomendacijas • Nereikalauja produktų kontekstinės informacijos	• Priklausomas nuo naudotojų vertinimų • Šalto paleidimo problema, kai sistemoje atsiranda nauji naudotojai ar produktai	• Amazon • LinkedIn
Hibridinis	• Galima išvengti šalto paleidimo problemų • Galima pritaikyti pagal konkrečios sistemos dalykinę sritį	• Sudėtingesnis įgyvendinimas • Reikia saugoti daugiau duomenų	• Spotify • Netflix

3 Mašininio mokymosi metodai taikomi rekomendacinėse sistemose

Kuriant rekomendacines sistemas, pagrindinis tikslas yra kuo geriau išanalizuoti turimus duomenis ir taip pateikti kuo tikslesnes rekomendacijas vartotojams, todėl mašininis mokymasis tampa neatsiejama to dalimi. Taikomi mašininio mokymosi metodai geba apdoroti didelį kiekį duomenų ir taip rekomenduoti prekes bei paslaugas, kurios galimai labiausiai tenkins vartotoją.

3.1 Mašininis mokymasis

Mašininis arba kompiuterio mokymasis (angl. *machine learning* – ML) yra dirbtinio intelekto atšaka, kuri apima daugybę algoritmų ir modeliavimo įrankių, naudojamų įvairioms duomenų apdorojimo, analizės užduotims spręsti (Carleo et al., 2019). Rašoma, kad ML per pastarąjį dešimtmetį patyrė didžiausią „atgimimą“ dėl milžiniško kiekio turimų duomenų ir reikšmingos pažangos skaičiavimuose (Edgar & Manz, 2017). ML paremti modeliai yra taikomi įvairiose pramonės bei mokslo srityse, pavyzdžiui, biologijos, chemijos, statistikos ir pan.

Pagrindinis mašininio mokymosi principas yra toks, kad kompiuteris gali mokytis pats ir atlikti jam paskirtas užduotis, jo neprogramuojant. Tad toks kompiuteris yra nuolat tobulėjantis iš patirties, nes jam yra suteikiami tik įvesties duomenys ir norimi gauti rezultatai, o visas taisykles, kaip apdoroti gautus duomenis, sukuria pats kompiuteris (Mohri et al., 2018). Kompiuterio mokymusi paremti modeliai yra taikomi sprendžiant klasifikavimo, klasterizavimo, regresijos ir panašias užduotis. Mašininis mokymasis yra skirstomas į tris pagrindines kategorijas:

- Prižiūrimas mokymasis (angl. *supervised learning*)

Ši kompiuterio mokymosi kategorija yra labiausiai paplitusi sprendžiant praktinius uždavinius. Prižiūrimo kompiuterio mokymosi metu modeliui yra paduodami įvesties (X) bei išvesties (Y) kintamieji ir naudojant algoritmus modelis yra apmokomas sudaryti taisykles, kaip pasiekti norimą rezultatą, kitaip tariant yra sudaroma $Y = f(X)$ funkcija. Pagrindinis tikslas yra taip gerai apmokyti kompiuterį, kad padavus naujas įvesties reikšmes, kompiuteris galėtų pats pasakyti, kokios bus išvesties reikšmės. Prižiūrimo kompiuterio mokymusi paremti modeliai yra naudojami sprendžiant klasifikavimo ir regresijos uždavinius. Skirtumas tarp šių uždavinių yra toks, kad klasifikacijos metu išvesties kintamieji yra kategoriniai, pavyzdžiui, raudona ar mėlyna, liga ar ne liga ir pan., o regresijos uždaviniuose išvesties kintamieji gali būti bet kokios skaitinės reikšmės (Schrider & Kern, 2018).

- Neprižiūrimas mokymasis (angl. *unsupervised learning*)

Neprižiūrimas mokymasis yra toks mašininio mokymosi tipas, kuriame modeliui apmokymo metu, skirtingai nuo prižiūrimo mokymosi, yra paduodami tik įvesties duomenys ir kompiuteris turi mokytis pats, be priežiūros, atrasti duomenų rinkinio struktūrą, atpažinti duomenų panašumus ir sugebėti juos apdoroti bei grupuoti. Šioje kompiuterio mokymosi kategorijoje pagrindiniai sprendžiami uždaviniai yra susiję su klasterizavimu (grupavimu), anomalijų aptikimu ir pan. (Ceriotti, 2019).

- Skatinamasis mokymasis (angl. *reinforcement learning* RL)

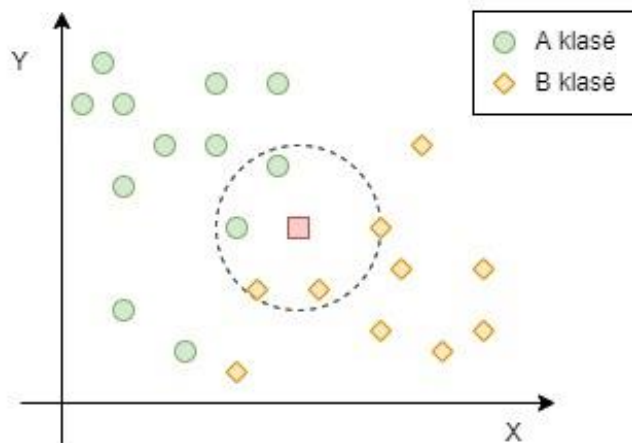
Šis kompiuterio mokymosi tipas yra ganėtinai ribotas sprendžiant realaus gyvenimo problemas dėl to, nes tokiose problemose reikia konkretaus sprendimo iš daugumos galimų. Skatinamasis mokymasis, kaip ir sufleruoja pavadinimas, yra paremtas modelio paskatinimu priimti kuo geresnius sprendimus. Tokiu būdu apmokytam kompiuteriui yra nurodomi veiksmi, kurie yra „skatinami“ (angl. *rewarding*) ir kurie yra „nubaudžiami“ (angl. *punishing*), tad su laiku kompiuteris apdorojant įvesties duomenis mokosi, kurie veiksmi yra skatinami ir taip bando priimti geriausius sprendimus sprendžiant paskirtą užduotį. Skatinamasis mašininis mokymasis gali būti naudojamas žaidimuose, valdant išteklius, robotikoje, kuriant rekomendacines sistemas ir pan. (Chandak et al., 2019).

Rekomendacinėse sistemose gali būti naudojami visi paminėti kompiuterio mokymosi tipai, nes šių tipų naudojimas priklauso nuo RS pobūdžio, turimų duomenų ir norimų gauti rezultatų. Toliau yra aprašomi kai kurie, rekomendacinėse sistemose taikomi, mašininio mokymosi metodai.

3.2 K-artimiausio kaimyno metodas

K-artimiausio kaimyno (angl. *k-Nearest Neighbor* KNN) algoritmas dažnai yra naudojamas bendrumo filtravimu paremtose rekomendacinėse sistemose. Šis algoritmas pirmiausia yra laikomas klasifikavimo metodu, tačiau dėl jo paprastumo, efektyvumo ir pritaikymo jį buvo pradėta naudoti duomenų struktūrai atpažinti bei duomenims išgauti (Gou et al., 2019). Paprasčiausias šio algoritmo pritaikymas yra toks, kai kompiuteris yra apmokomas duomenų rinkiniu, kuriame yra skirtingų klasių duomenų ir tos klasės yra žinomos. Tuomet skirtingomis klasėmis apmokytam kompiuteriui yra paduodami nauji duomenys. Kompiuteris naujų duomenų taškus lygina su senais duomenimis ir naujus duomenis priskiria tai klasei, kurios senų duomenų aplinkui yra daugiausiai. Šiame algoritme yra naudojamas parametras – k , kuris nurodo artimiausių kaimynų, įtraukiamų į sprendimo aibę, skaičių (Chomboon et al.,

2015). Kaip pavaizduota 3.1 pav., turime A ir B klases, tarkime parametras $k=5$, naujas duomuo – raudonas kvadratas. Tai reiškia, kad kompiuteris išrinks raudonam kvadratui artimiausius 5 taškus ir raudoną kvadratą priskirs tai klasei, kurios duomenų yra daugiau, šiuo atveju naujas duomuo priklausys B klasei.



3.1 pav. K-artimiausio kaimyno algoritmas (pagal Chomboon et al., 2015)

Šis algoritmas gali būti naudojamas rekomendacinėse sistemose. Pavyzdžiui, KNN buvo panaudotas stebint vartotojų paspaudimų srautą ir sudarant tam tikras vartotojų grupes pagal naršymo duomenis, tokiu būdu galima pasiūlyti kitas, vartotojų grupės poreikius atitinkančias, naršymo parinktis (Adeniyi et al., 2016).

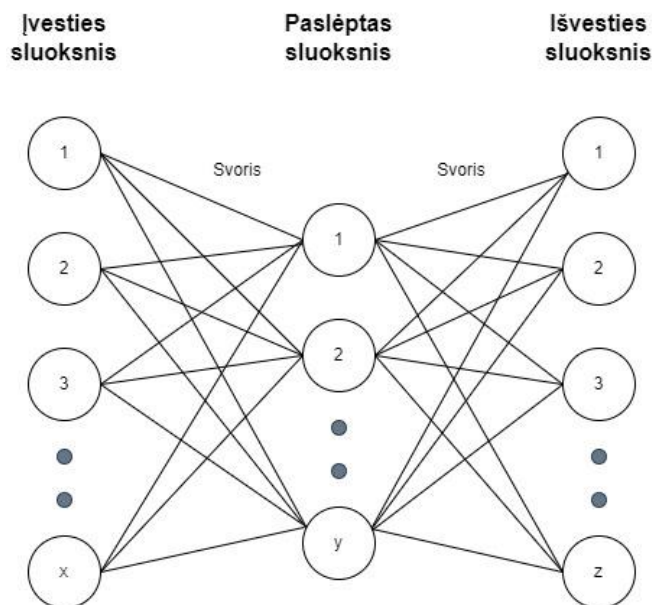
3.3 K-reikšmių klasterizavimas

Rekomendacinių sistemų pagrindinis principas yra atrasti panašumus tarp įvairių skirtingų objektų, tarsi sugrupuoti skirtingus, pavyzdžiui, produktus į tam tikras grupes, kurios galimai patiks vienam ar kitam naudotojui. Tad vienas iš metodų, rekomendacijoms sudaryti, yra klasterizavimas (angl. *clustering*). Klasterizavimas, kitaip tariant grupavimas, yra procesas, kurio metu tam tikri objektai (ar duomenys) yra sugrupuojami į grupes, kuriose esantys objektai tarpusavyje yra panašesni, nei su kitos grupės objektais (Ahuja et al., 2019). Mašininio mokymosi kontekste klasterizavimas gali būti ne vieno tipo. Kuriant rekomendacines sistemas, dažnai yra naudojamas būtent k-reikšmių klasterizavimas (angl. *k-means clustering*). Galbūt iš pavadinimo, tai gali atrodyti panašus mašininio mokymosi metodas, kaip ir k-artimiausio kaimyno, tačiau šie metodai turi didelį ir svarbų skirtumą – k-artimiausio kaimyno metodas yra prižiūrimo kompiuterio mokymosi klasifikacijos algoritmas, o k-reikšmių klasterizavimo metodas yra priskiriamas neprižiūrimam kompiuterio mokymuisi. Kuomet k-artimiausio kaimyno algoritmas klasifikuoja žinomus duomenis į jau žinomas klases – k-reikšmių klasterizavimo metodas sukuria naujas klases iš nepažymėtų duomenų. K-reikšmių klasterizacijos algoritmas apskaičiuoja centroidus – duomenų aibės plokštumoje esančius taškus, kurie žymi tam tikros grupės duomenų centrą – ir kartoja šį procesą iki tol, kol yra

randamas optimalus centroidų kiekis. Tie centroidai žymi skirtingas duomenų grupes, klasterius. Šiame metode klasterių, kitaip tariant – branduolių ar grupių, kiekis yra apibrėžiamas raide „K“. Viso k-reikšmių klasterizavimo metodo principas yra toks, kad duomenų taškai yra priskiriami tam tikram klasteriui taip, kad pakėlus kvadratu atskirai visas atkarpas nuo duomenų taško iki centroido ir tas reikšmes sudėjus kartu – ji būtų kiek įmanoma mažesnė (Sinaga & Yang, 2020). K- reikšmių klasterizacijos metodas rekomendacinėse sistemose gali būti naudojama įvairiais būdais sudarant tam tikrų duomenų grupes, pavyzdžiui, iš duomenų apibūdinančių naudotojų elgseną ar tam tikrų prekių bei paslaugų informacijos.

3.4 Neuroniniai tinklai

Gilusis mokymasis (angl. *deep learning*) yra mašininio mokymosi atšaka ir būtent neuroniniai tinklai (angl. *neural networks*) yra priskiriami šiai atšakai, dėl jų veikimo principo. Ši mokymosi technika istoriškai buvo įkvėpta tuo, kaip skaičiavimai veikia žmogaus smegenyse. Mašininio mokymosi metodai gali būti apibūdinami, kaip modeliai, kurie mokosi prognozuoti pagal praeities stebėjimus, kai giliojo mokymosi metu kompiuteris yra apmokomas ne tik prognozuoti, bet ir tai, kaip teisingai pateikti duomenis, kad jie būtų tinkami tolesniam prognozavimui (Goldberg, 2017). Paprastas neuroninis tinklas yra sudarytas iš dirbtinių neuronų, kurie yra suskirstyti į tarpusavyje susijungiančius sluoksnius. Apmokymo metu duomenys yra paduodami įvesties sluoksniui, tuomet paslėptame sluoksnyje (angl. *hidden layer*) vyksta mokymasis iš duomenų, jų atpažinimas bei apdorojimas ir galiausiai rezultatai yra perleidžiami išvesties sluoksniui (3.2 pav.). Dažniausiai visi neuroninių tinklų modeliai yra visiškai sujungti, tai reiškia, kad visi sluoksniai turi neuronų ryšius. Tokių ryšių tarp sluoksnių stiprumas yra apibūdinamas svoriu (angl. *weight*), kuris pažymi kokią svarbą turi įvesties reikšmė kitos atžvilgiu (Tanty Tanweer Desmukh, 2015). Būtent šis mašininio mokymosi metodas buvo panaudotas kompanijos „YouTube“ vaizdo įrašų rekomendacinėje sistemoje, kurioje yra analizuojami vartotojo peržiūrėti vaizdo įrašai ir remiantis bendrumo filtravimu pateikiamos kitos vaizdo įrašų rekomendacijos (Covington et al., 2016).



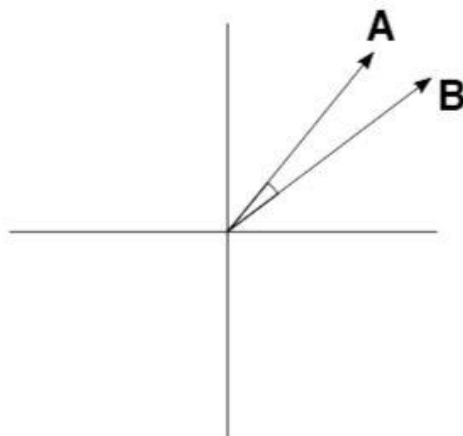
3.2 pav. Neuroninių tinklų modelio struktūra (pagal Tanty Tanweer Desmukh, 2015)

3.5 Kosinuso panašumas

Panašumas (angl. *similarity*) yra viena pagrindinių sąvokų mašininio mokymosi algoritmų - klasifikacijos ir klasterizavimo - uždaviniuose. Panašumas tokiuose uždaviniuose matuoja, apibrėžia bendras analizuojamų kintamųjų charakteristikas. Taip pat viena iš sąvokų, glaudžiai susijusi su panašumu, yra atstumas ar atitolimas (angl. *distance*) tarp kintamųjų. Tokia sąvoka žymi atstumą tarp analizuojamų kintamųjų – jų skirtumus. Sujungus šias sąvokas į vieną visumą, pradžioje galime apskaičiuoti kintamųjų skirtumus, atstumą tarpusavyje, o tuomet išreikšti ribą, kurioje tuos kintamuosius laikome panašiais, visai tai reiškia, kad kuo atstumas tarp kintamųjų yra mažesnis tuo jie yra panašesni (Lahitani et al., 2016).

Kaip minėta anksčiau, klasifikacijos ar klasterizavimo uždaviniams spręsti yra naudojamas kintamųjų panašumo matas. Panašumui matuoti galima naudoti daugybę skirtingų technikų, viena tokių, dažnai taikomų technikų yra Euklido erdvė (angl. *Euclidean space*). Euklido erdvėje, atstumas tarp dviejų taškų yra matuojamas linijos segmentu tarp jų, tačiau šio technikos trūkumas yra didelis jautrumas dydžiams, įvairovei (angl. *magnitudes*) (Xia et al., 2015). Kita, dažnai naudojama matavimo technika yra kosinuso panašumas. Kosinuso panašumo matas išreiškia panašumą, kaip kampą tarp dviejų vektorių. Euklido erdvėje du modeliai yra laikomi mažiau panašūs, kai Euklido atstumas tarp jų didėja, tačiau jie laikomi labiau panašūs, kai kosinuso panašumas tarp jų padidėja. Kosinuso panašumo matas, priešingai nei Euklido erdvės, yra nejautrus dydžiams, įvairovei, tačiau ši savybė ne visada yra privalumas, nes, pavyzdžiui, jei turime du modelius su labai skirtingomis atributų vertėmis, vis tiek kosinuso panašumas galimai bus didelis. Šiam trūkumui pašalinti galima pakoreguoti kosinuso panašumo metriką. Pagrindinis kosinuso panašumo metrikos principas yra – kuo mažesnis kampas tarp sudarytų duomenų vektorių, tuo didesnis kosinuso panašumas. Kosinuso

matavimo imtis yra nuo -1 iki 1, jei kampas tarp vektorių lygus nuliui, tuomet kosinuso panašumas yra 1, o jei kampas tarp vektorių yra 180 laipsnių, tuomet kosinuso funkcijos reikšmė laikysime -1. Pavyzdžiui, kaip pavaizduota 3.3 pav., A ir B vektorių(duomenų) kosinuso panašumas yra didelis, nes kampas tarp jų yra mažas ir tie vektoriai nukreipti į panašią pusę. Tokia kosinuso panašumo metrika, kuriamose bendrumo filtravimo tipo rekomendacinėse sistemose, gali būti naudojama skirtingų naudotojų sudaromų profilių panašumo nustatymui. Tokiu būdu, apskaičiavus kosinuso panašumus, būtų galima rekomenduoti prekes ar paslaugas naudotojui, kurio naudojimosi įpročiai yra panašūs.



3.3 pav. Vektorių kosinuso panašumas

3.6 Gauti mašininio mokymosi metodų rezultatai

Šiame skyriuje, atlikus literatūrinę analizę, yra pateikiamos pagrindinės mašininio mokymosi sąvokos, tipai, jų aprašymai ir pagrindiniai tikslai. Taip pat pateikiami skirtingi mašininio mokymosi metodai – jų aprašymai ir principai – kurie gali būti taikomi rekomendacinėse sistemose (žr. 3.1 lentelė).

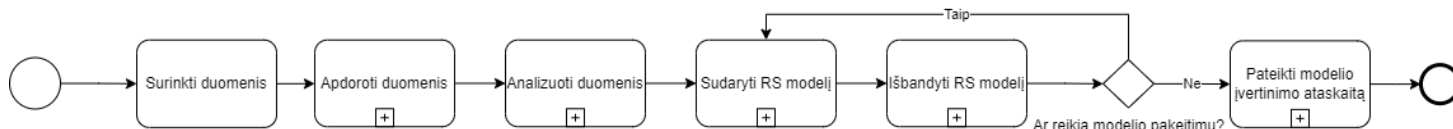
Taigi atlikus literatūrinę analizę, galima teigti, kad rekomendacinėse sistemose gali būti naudojami įvairūs mašininio mokymosi algoritmai, nes tai priklauso nuo rekomendacinės sistemos rūšies, poreikio bei turimų duomenų.

3.1 lentelė. RS naudojamų mašininio mokymosi algoritmų palyginimas

RS tipas	Dalykinės srities duomenys	Taikomi metodai/algoritmai
Turiniu paremtas	• Produktų duomenys • Naudotojo profilio duomenys	• Klasterizavimas • Kosinuso panašumas • Pearson koreliacija • Bajeso klasifikacija • Sprendimų medis
Bendrumo filtravimu paremtas	• Naudotojo pateikti įvertinimai • Kitų naudotojų profiliai	• Klasterizavimas • Kaimynais pagrįstas metodas • K-artimiausių kaimynų algoritmas • Koreliacija paremtas panašumas

4 Filmų rekomendacinių sistemų projektavimas

Viso atliekamo darbo struktūrą galima apibrėžti tam tikrais procesais. 4.1 pav. yra pavaizduota atliekamų procesų struktūra ir eiga (žr. 4.1 pav.).



4.1 pav. Tyrimo procesų diagrama

Atliekamo tyrimo esmė yra išanalizuoti skirtingų rekomendacinių sistemų tipų – pagrįstų turiniu ar bendrumo filtravimu – pagrindinius veikimo principus, sistemų skirtumus, privalumus ar trūkumus ir palyginti juos tarpusavyje. Analizė pradedama nuo duomenų aibės pasirinkimo. Šiam darbui atlikti buvo pasirinkta atviros prieigos „MovieLens“ duomenų bazė (Maxwell Harper & Konstan, 2015). Duomenų rinkinys buvo surinktas Minesotos Universiteto „GroupLens“ tyrimų projektų komandos, kuri suteikia galimybę šiuos duomenis naudoti asmenims, norintiems nagrinėti, analizuoti ar atlikti tam tikrus tyrimus su naudotojų elgsenos duomenimis „MovieLens“ filmų internetinėje svetainėje.

Pasirinkus duomenų rinkinį, tinkamą tyrimui atlikti, yra atliekamas duomenų rinkinio apdorojimo procesas (angl. *preprocessing*). Šio proceso metu yra peržiūrima bei tvarkoma pati duomenų struktūra, gali būti pakeičiami ar pašalinami tam tikri duomenys, kurie, pavyzdžiui, tyrimui atlikti yra nereikalingi arba jų struktūra tolimesnei analizei nėra patogi.

Apdorojus ir pritaikius duomenų rinkinį tolesniam darbui yra atliekama duomenų analizė. Tai yra itin svarbus tyrimo sudedamasis procesas. Šio proceso metu yra detalčiai nagrinėjami patys duomenys, vykdoma statistinė, kokybinė bei kiekybinė duomenų analizė. Duomenų analizės metu mes galime išskirti tam tikrus duomenis, kurie bus tikslingai panaudoti tolimesniuose tyrimuose sudarant ir vertinant rekomendacines sistemas. Atlikus duomenų analizės procesą, galima lengviau apibrėžti tolesnius uždavinius viso darbo tikslui įgyvendinti. Vėliau, po duomenų analizės, yra atliekamas vienas iš pagrindinių viso darbo uždavinių – rekomendacinės sistemos modelio sudarymas.

Rekomendacinės sistemos modelio sudarymas, šio darbo kontekste, yra toks procesas, kurio metu pasirinktam duomenų rinkiniui bus pritaikomi įvairūs mašininio mokymosi algoritmai, būdingi skirtingo tipo rekomendacinėms sistemoms. Šio procesu metu bus tyrinėjami skirtingi duomenys ar jų rinkiniai, atsižvelgiant į kuriamos rekomendacinės sistemos tipą. Pritaikius tam tikrus mašininio mokymosi algoritmus ar duomenų analizės, filtravimo funkcijas, bus vykdomas sukurtą preliminarų modelio ar modelių išbandymas. Išbandžius sukurtus rekomendacinės sistemos modelius, galėsime pateikti tam tikras išvagas, kaip būtų galima dar patobulinti darbo metu kuriamą rekomendacinės sistemos modelį, jei to reikia.

Patobulinius sukurtus modelius, juos turėsime įvertinti ir palyginti. Šio proceso metu turėtumėme pateikti tam tikras išvadas apie skirtingo tipo rekomendacines sistemas, jų struktūrą, veiklą bei veikimo principus, trūkumus ar privalumus.

4.1 Duomenų surinkimas

Kaip ir minėta anksčiau, tyrimui atlikti buvo pasirinktas „MovieLens“ atviros prieigos duomenų rinkinys. Kadangi duomenų rinkinys yra atviros prieigos ir lengvai parsisiunčiamas, tolimesniems darbams jis buvo parsisiųstas ir įrašytas į kompiuterio lokalią atmintį. Pasirinktą duomenų rinkinį sudaro trys pagrindinės duomenų lentelės pavadinimais: *data*, *item* ir *user*. „Data“ lentelėje yra saugomi duomenys apie tai, koks naudotojas, kokį filmą ir koku balu įvertino. Šiame duomenų rinkinyje yra 100 tūkstančių įvertinimų, kuriuos bendrai paliko 943 naudotojai sistemoje patalpintiems 1682 filmams, taip pat duomenų rinkinyje yra tik tie naudotojai, kurie yra įvertinę mažiausiai po 20 skirtingų filmų. Duomenų lentelė „item“ apibrėžia filmų detalesnius duomenis – pavadinimą, žanrus, išleidimo datą ir pan. Svarbu pastebėti, kad pateiktame duomenų rinkinyje, filmų žanrų tipai yra išdėstyti į atskirus stulpelius, pažymint reikšmę 1 arba 0, kur atitinkamai filmas priklauso nurodytam žanrui arba ne. Kita, „user“ lentelė, aprašo demografinius naudotojų duomenis, tokius kaip: naudotojo amžius, lytis, profesiją ir t.t. Visos lentelės siejasi tarpusavyje su identifikaciniais naudotojo ir filmo numeriais.

4.2 Duomenų apdorojimas

Duomenų apdorojimo procese yra svarbu peržiūrėti visus duomenis, išanalizuoti ir suprasti, kurie duomenys ar jų formatai gali daryti įtaką tolimesnių darbų tikslesniam įgyvendinimui.

Pasirinktą „MovieLens“ duomenų rinkinį, kaip ir minėta anksčiau, sudaro trys duomenų lentelės, jas visas galima sujungti į bendrą duomenų struktūrą per filmų ir naudotojų identifikacinius numerius. Bendrą duomenų lentelę sudaro 13 stulpelių. Detalesnis duomenų rinkinio struktūros aprašymas pateiktas 4.1 lentelėje.

4.1 lentelė. Duomenų struktūra

Nr.	Kintamojo pavadinimas	Paaiškinimas
1.	user_id	Naudotojo identifikacinis numeris
2.	age	Naudotojo amžius
3.	gender	Naudotojo lytis
4.	occupation	Naudotojo profesija
5.	zip_code	Naudotojo gyvenamosios vietos pašto kodas
6.	movie_id	Filmo identifikacinis numeris
7.	rating	Įvertinimas balais nuo 1 iki 5
8.	timestamp	Laiko žyma, kuomet buvo įvertintas filmas
9.	movie_title	Įvertinto filmo pavadinimas
10.	release_date	Filmo išleidimo data
11.	video_release_date	Šis duomenų lentelės stulpelis, realiame duomenų rinkinyje

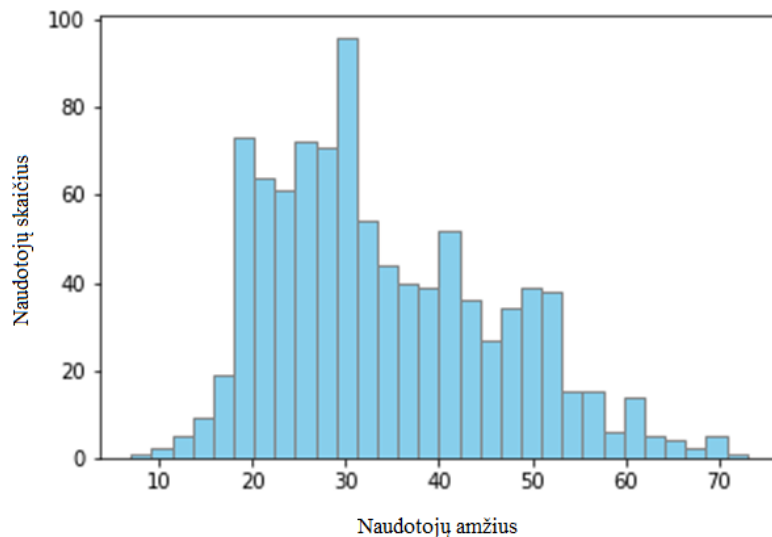
Nr.	Kintamojo pavadinimas	Paiškinimas
		buvo pateiktas su tuščiomis reikšmėmis, tad galima daryti prielaidą, kad būtent šiame duomenų rinkinyje jis neturi reikšmės ir gali būti pašalinamas
12.	imdb_url	„IMDb“ internetinės svetainės nuoroda. Šioje svetainėje yra patalpinta filmų, serialų ir t.t. duomenų bazė, kurioje laikoma detali informacija. Tačiau šio darbo kontekste, svetainėje esantys duomenys nebus naudojami ir kadangi šio stulpelio reikšmės nedaro įtakos kitiems duomenų rinkinio duomenims – galime pašalinti šį stulpelį
13.	žanrai	Žanrai, kaip ir minėta anksčiau, šiame duomenų rinkinyje yra išskirti į atskirus stulpelius, pažymint reikšmę 1 arba 0, atitinkamai ar tas filmas priklauso nurodytam žanrui ar ne. Iš viso yra 19 žanrų. Šį skaičių sudaro ir žanras „unknown“ (liet. nežinoma), tačiau šiam žanrui yra priskirtas tik vienas filmas, tad buvo nutarta šią reikšmę pašalinti iš tolimesnio tyrimo, kadangi palyginus su kitų duomenų turimu kiekiu, tai neturėtų daryti įtakos tolimesniems tyrimams.

Nagrinėjamą duomenų rinkinį sudaro tiek kokybiniai, tiek kiekybiniai duomenys. Tiesa, didžioji dalis duomenų yra kokybinio tipo ir nusako tam tikrą kategoriją, pavyzdžiui, naudotojų lytis, profesija, filmų žanrai ar pan. Detaliau peržvelgus pasirinktą duomenų rinkinį, buvo nuspręsta panaikinti tam tikrus duomenis, kurie yra nereikalingi tolimesniam tyrimui. Tad iš šio duomenų rinkinio bus panaikinta: *video_realease_date*, *imdb_url* bei *unknown* filmų žanras.

4.3 Duomenų analizė

Išsirinkus tinkamą duomenų rinkinį ir apdorojus jo duomenis, toliau galima pereiti prie duomenų analizės. Duomenų analizės procesas padės susidaryti bendrą vaizdą apie duomenų rinkinį, jo struktūrą, statistines galimybes. Viso šio proceso rezultatas yra naudingas tolimesniems darbams.

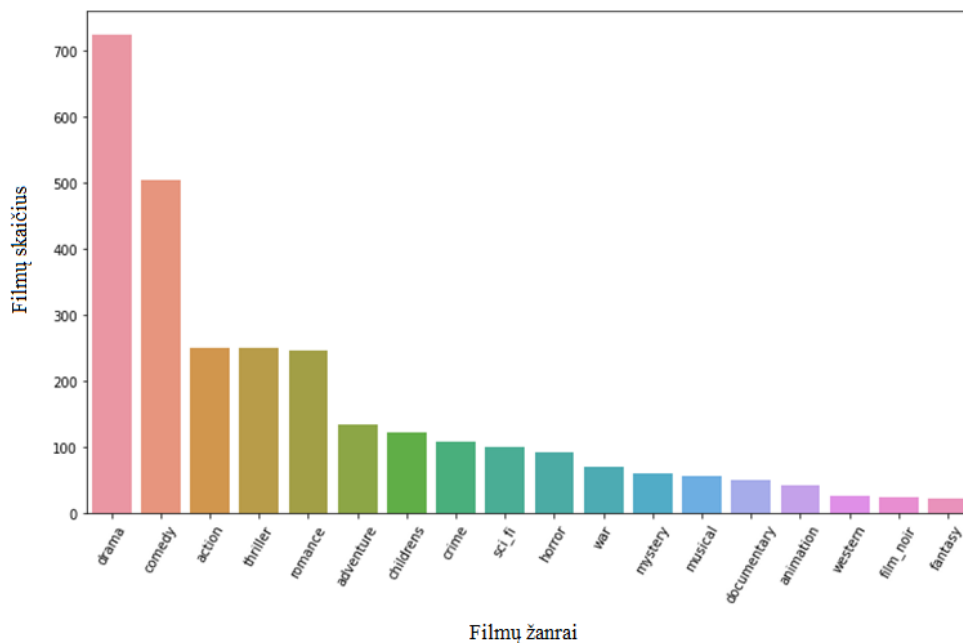
Pirmiausia, galima atidžiau pažvelgti į naudotojų duomenis. Pasirinktame duomenų rinkinyje yra 943 naudotojų identifikaciniai numeriai, tai reiškia, kad būtent tiek naudotojų peržiūrėjo ir įvertino ne mažiau kaip po 20 filmų. Kaip aprašyta 4.1 lentelėje naudotojų duomenis šiame rinkinyje apibūdina naudotojo identifikacinis numeris, amžius, lytis, profesija bei pašto kodas. Kuriant tam tikro tipo rekomendacines sistemas yra svarbu atsižvelgti į tikslinio naudotojo amžių. Amžius gali tapti svarbiu kriterijumi tik pradedant kurti rekomendacines sistemas ar jas tobulinant, nes pavyzdžiui, kas patiks 10 metų vaikui, galimai nepatiks vyresnio amžiaus žmonėms, nes skiriasi patirtis, suvokimas ir t.t. Panagrinėjus duomenų rinkinyje esančių naudotojų amžių, galima teigti, kad didžiausią naudotojų dalį sudaro jaunesnio amžiaus – 20-35 metų – asmenys (žr. 4.2 pav.). Taip pat galima pastebėti, kad 9 – 16 metų vaikai sudaro mažąją dalį naudotojų kiekio bei tai, kad naudotojų skaičius nuo 50 metų yra linkęs sparčiai mažėti.



4.2 pav. Naudotojų amžiaus pasiskirstymas

Diagramos rezultatus taip pat galima pagrįsti ar apskaičiuoti kitais statistiniais matavimo vienetais, tokiais kaip: naudotojų amžiaus vidurkis, mediana ir moda. Mediana apibrėžia vidurinių eilės skaičių, kai duomenų eilė yra išrikiuota didėjančia tvarka, o moda yra daugiausiai kartų pasikartojęs skaičius visoje eilėje. Šiuo atveju, naudotojų amžiaus vidurkis yra 34 metai, mediana lygi 31, o moda – 30, tai taip pat matosi ir iš vizualiai pateiktų rezultatų 4.2 pav., ši nustatyta moda skaičių aibėje pasikartoja 39 kartus iš 943 atvejų.

„MovieLens“ duomenų rinkinio esmė yra surinkta informacija apie filmus, kuriuos įvertino atitinkami naudotojai. Duomenų rinkinyje esančią filmų lentelę sudaro: filmo identifikacinis numeris, pavadinimas, išleidimo data ir žanrai. Kaip paminėta ankstesniame poskyryje iš šios duomenų lentelės yra pašalinamas vaizdo įrašo datos stulpelis, kuris buvo tuščias, bei „IMBd“ nuorodos stulpelis, kurio duomenų tolimesniems tyrimams neprireiks. Filmų duomenų lentelėje žanrai yra padalinti į atskirus stulpelius, reikia pažymėti tai, kad nežinomas (angl. *unknown*) filmo žanras buvo pašalintas iš duomenų rinkinio, nes tai nesuteikia jokios naudingos informacijos. Bendrai, be nežinomo filmo žanro, filmai yra priskiriami 18 žanrų kategorijų, taip pat filmai gali būti priskiriami ne vienai kategorijai. Norint atlikti statistinę filmų analizę, galima paskaičiuoti, koks filmų žanras yra populiariausias – labiausiai žiūrimas – tarp naudotojų (žr. 4.3 pav.). Kaip galima pastebėti iš stulpelinės diagramos pavaizduotos žemiau, populiariausias filmo žanras yra drama, o kiek mažiau populiarius – komedija. Mažiausiai dėmesio iš naudotojų susilaukia fantastinio žanro filmai. Internetinėse svetainėse, kurioje, pavyzdžiui, būtų rekomenduojami filmai – jų žanras – yra labai svarbus aspektas pateikiant tikslias rekomendacijas, nes naudotojai, tarkime, linkę žiūrėti tik romantinius filmus, galimai siaubo filmų žiūrėti nenorės, tad ir rekomenduoti tokius filmus yra ne itin tikslinga.



4.3 pav. Filmų žanrų pasiskirstymas

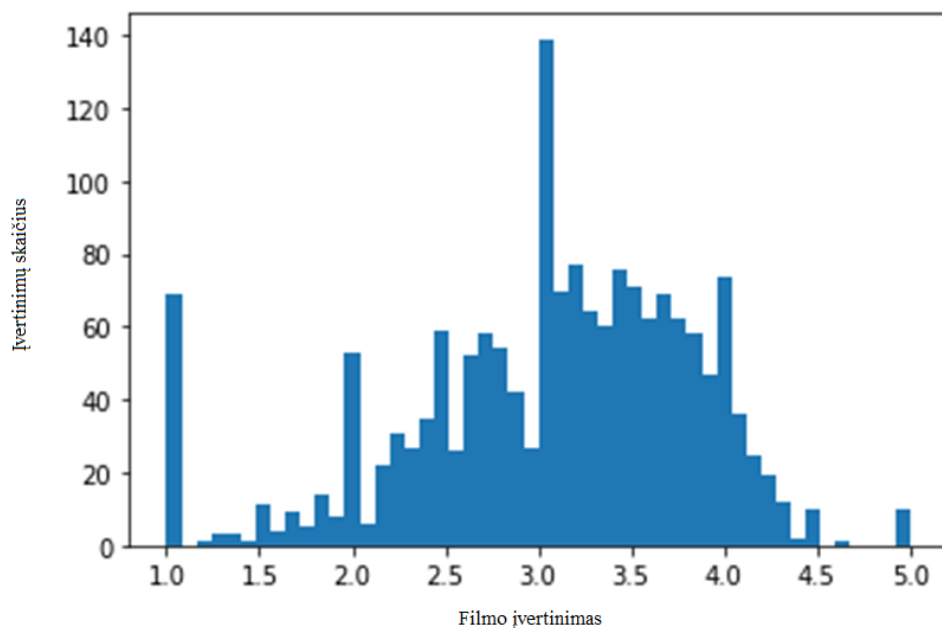
Pagrindinę duomenų lentelę „data“ sudaro: naudotojo bei filmo identifikaciniai numeriai, filmo įvertinimas bei pateikto įvertinimo data ir laikas. Įvertinimai skiriami peržiūrėtiems filmams penkių balų sistemoje – konkrečiai šiame duomenų rinkinyje filmų įvertinimai svyruoja nuo 1 iki 5 balų. Įvertinimų analizė, filmų kontekste, yra itin svarbus aspektas, nes tokiu būdu naudotojai, nematę tam tikro filmo, gali susidaryti nuomonę iš pateiktų įvertinimų, pavyzdžiui, jei filmo įvertinimų vidurkis yra labai žemas, didėja tikimybė, kad naudotojas, peržiūrėjęs tokius įvertinimus, to filmo nežiūrės arba jei filmas yra ganėtinai senas, tačiau įvertinimų kiekis labai mažas, tokiu atveju naudotojui taip pat gali kilti dvejonų, kodėl žmonės nežiūri to filmo. Tad filmų įvertinimai gali suteikti daug naudingos informacijos naudotojams. Iš tiesų, šie duomenys gali būti aktualūs ne tik naudotojams, bet ir kuriant produktų ar paslaugų rekomendavimo sistemas. Tarkime, bendrumo filtravimo rekomendacinėje sistemoje, sudarant skirtingų naudotojų tam tikrus profilius, jų pateikti įvertinimai, gali būti naudingi sudarant naujas rekomendacijas kitiems naudotojams – geriau vieno naudotojo įvertintą filmą rekomenduoti kitam panašiam naudotojui yra tikslingiau nei rekomenduoti tą filmą, kurį naudotojas įvertino blogai ar išvis neįvertino.

Peržvelgus ir atlikus pirminę filmų įvertinimų, esančių „MovieLens“, analizę galima apskaičiuoti tam tikrus svarbius parametrus: įvertinimų vidurkį, medianą ir modą. Tad būtent pasirinkto duomenų rinkinio statistiniai įvertinimų duomenys yra:

- Vidurkis – 3,53;
- Mediana – 4;
- Moda – 4.

Naudotojų peržiūrėtų filmų įvertinimų vidurkis yra 3,53, tai reiškia, kad didesnę dalį filmų naudotojai įvertini geru – daugiau nei 3 – įvertinimu, tai įrodo ir mediana bei moda, kurios lygios – 4. Taip pat 4 moda, įvertinimų aibėje pasikartoja 34174 kartus iš 100 tūkstančių.

Panagrinėjus „MovieLens“ duomenų rinkinio įvertinimus (žr. 4.4 pav.) – įvertinimų kiekio pasiskirstymą atsižvelgiant į įvertčio balą – galima pastebėti, kad naudotojai dažniausiai filmus vertina 3 – 4 balų intervale, taip pat, kad beveik 70 įvertinimų sudarė įvertinimas 1 balu, o didžiausias – penkių balų – įvertinimas buvo pateiktas vos 10 kartų.



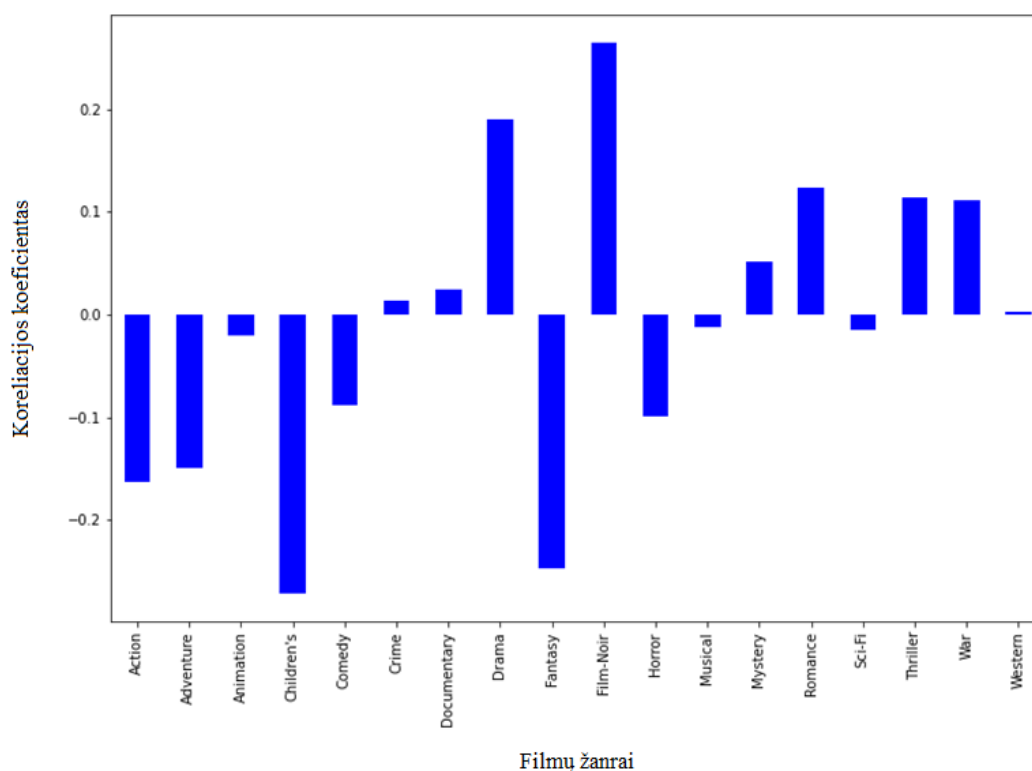
4.4 pav. Įvertinimų skaičius pagal įvertį

Analizuojant įvairaus tipo duomenis yra naudinga apskaičiuoti koreliacijas tarp skirtingų duomenų. Tiesa, koreliacijų skaičiavimas yra dažniau taikomas tarp kiekybinių duomenų. Nors pasirinktame duomenų rinkinyje kiekybinių duomenų nėra itin daug, galime paanalizuoti ir apskaičiuoti keletą koreliacijų, kurios galimai būtų naudingos tolimesniems tyrimams.

Koreliacija yra statistinis matas, parodantis, kiek kintamieji yra priklausomi vienas nuo kito, o koreliacinės analizės tikslas yra išsiaiškinti, ar vieno kintamojo pokyčiai sukelia kito kintamojo pokyčius. Šio statistinio mato reikšmė svyruoja nuo -1 iki 1. Koreliacijos koeficiento reikšmė 0 reiškia, kad kintamieji yra statistiškai nepriklausomi tarpusavyje ir tarp jų koreliacijos nėra. Koeficientas -1 parodo, kad tarp kintamųjų yra labai stipri neigiama koreliacija ir indikuoja, kad vieno kintamojo didėjimas prognozuoja kito kintamojo mažėjimą ir atvirkščiai – tų kintamųjų ryšys yra atvirkščiai proporcingas. Tuo tarpu koreliacijos koeficientas 1 rodo, kad kintamieji labai stipriai koreliuoja tarpusavyje ir jų, pavyzdžiui, didėjimas yra vienkryptis – jei vienas didėja, kitas irgi padidės. Spearmano ranginė koreliacija (angl. *Spearman rank correlation*) yra neparametrinis testas, naudojamas dviejų kintamųjų ryšio laipsniui matuoti.

Toks testas yra tinkamas šiame darbe analizuojamam duomenų rinkiniui, nes turi reitinguojamus kintamuosius.

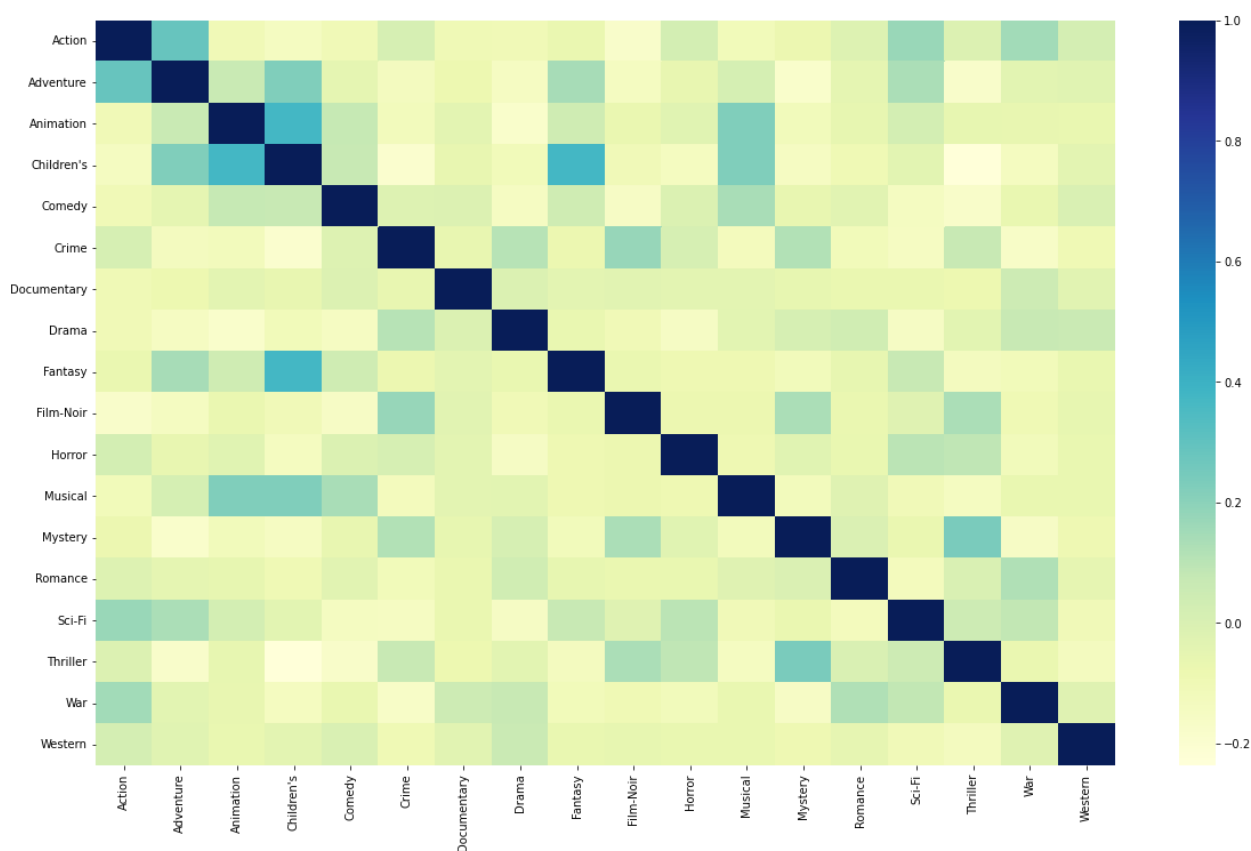
Šio darbo uždaviniai ir pagrindinis tikslas yra glaudžiai susijęs su naudotojų peržiūrėtais filmais ir jų vertinimais, tad galime paskaičiuoti, kaip filmų vertinimų vidurkis koreliuoja su filmo žanru. Panaudojus *Python* funkciją *corr(method = „spearman“)*, galime paskaičiuoti Spearmano ranginę koreliaciją tarp įvertinimų vidurkio ir filmo žanro. Peržiūrėjus rezultatus (žr. 4.5 pav.), galima pastebėti, kad koreliacijos koeficientų diapazonas nėra itin platus. Didžiausia teigiama koreliacija yra tarp filmo žanro „Film-Noir“ ir tokio žanro įvertinimų vidurkio – 0,27 – tokiu atveju galima daryti prielaidą, kad naudotojai šio žanro filmus yra linkę vertinti kiek geriau, taip pat kaip ir dramos tipo filmus. Tiesa, šios koreliacijos nėra itin stiprios, todėl ir prielaidos ne visada gali būti teisingos bei priklausyti nuo analizuojamos duomenų bazės. Spearmano koreliacijos rezultatų diagramoje taip pat galima pastebėti, kad koeficientas yra žemesnis tarp vaikiškų bei fantastikos filmų žanrų ir jų vertinimų vidurkio, galbūt tokio tipo filmus naudotojai yra linkę vertinti prasčiau.



4.5 pav. Spearman koreliacija tarp filmų žanrų ir įvertinimų vidurkio

Taip pat filmų įvertinimų vidurkių duomenis galima panaudoti skaičiuojant ir kitą Spearman koreliaciją. Šiame darbe, kuriant ir lyginant rekomendacines sistemas yra svarbu, kad rekomendacijos, pateiktos naudotojams, būtų kuo tikslesnės. Galima daryti prielaidą, kad naudotojui, mėgstančiam vienokį filmo žanrą labiau patiks ir kiti to paties ar panašaus žanro filmai. Išanalizavus turimą duomenų bazę, galime sugrupuoti filmų žanrus ir jų įvertinimų

vidurkių, tuomet palyginti tuos įvertinimų vidurkių tarp skirtingų žanrų, tokiu būdu būtų galima apskaičiuoti ir palyginti koreliacijos koeficientus. Šiam palyginimui taip pat buvo naudojama *Python* funkcija *corr(method = "spearman")*, o apskaičiuotų koreliacijų koeficientų rezultatai vizualizuojami pasitelkus šilumos grafiką (angl. *heat map*) (žr. 4.6 pav.). Pateiktame šilumos grafike galima pastebėti, kad kai kurie filmų žanrai tarpusavyje turi didesnį koreliacijos koeficientą ir jų langeliai yra šiek tiek ryškesnės mėlynos spalvos. Pavyzdžiui, koreliacijos koeficientas tarp animacijos ir vaikų filmo žanrų yra 0,38, tai reiškia, kad naudotojas gerai vertinantis animacijos žanro filmus yra linkęs geriau vertinti ir filmus, kurie yra skirti vaikams. Taip pat koreliacija yra šiek tiek didesnė nei nulis tarp tokių žanrų porų, kaip nuotykių ir veiksmo, vaikų ir nuotykių, fantastikos ir vaikų, mistinių ir trilerių ir t.t. Nors šių koreliacijų tarp žanrų koeficientai ir nėra itin dideli, tačiau tai gali parodyti tam tikrą panašumą tarp skirtingų žanrų atsižvelgiant į naudotojų vertinimus. Tokie duomenys gali būti vertingi kuriant rekomendacinę sistemą, nes tai parodo, kad jei naudotojas, pavyzdžiui, labiau mėgsta – geriau vertina – miuziklo žanro filmus, galimai siaubo žanro filmų jam nederėtų siūlyti, nes mažėja tikimybė, kad tokį filmą, net ir peržiūrėjęs, naudotojas bus linkęs jį vertinti prasčiau, tad tokiu atveju būtų palankiau pirmiausia pasiūlyti animacijos, komedijos ar vaikams skirtus filmus, nes koreliacija su šiais filmo žanrais yra šiek tiek didesnė.



4.6 pav. Šilumos grafikas, atvaizduojantis koreliacijos koeficientus tarp skirtingų filmo žanrų ir jų įvertinimų vidurkio

4.3.1 Rekomendacinės sistemos paremtos duomenų analizės metodais

Remiantis duomenų analizės metodais, galima sudaryti paprastą nesuasmeninto tipo rekomendacinę sistemą, kuri buvo paminėta 2.2 poskyryje. Nesuasmeninta rekomendacinė sistema remiasi produkto populiarumu ir gali rekomenduoti naudotojui produktą, kuris yra populiarus tarp kitų naudotojų. Šiuo atveju, pasirinktame duomenų rinkinyje panagrinėjus pagrindinę „data“ duomenų lentelę, galima nustatyti, kurie filmai yra populiariausi, žiūrimiausi kitų naudotojų. Paskaičiavus įvertinimų kiekius kiekvienam filmui, galima nustatyti, kad daugiausiai įvertinimų turi filmas „Žvaigždžių karai“ (angl. *Star Wars*) – 583 įvertinimus, kurių įverčio vidurkis yra 4,36 (žr. 4.2 lentelė). Tad nustačius tokius duomenis, juos būtų galima panaudoti, pavyzdžiui, naujai prisiregistravusiam sistemos naudotojui, kuris dar nežiūrėjo jokių filmų ir jam pateikti suasmenintas rekomendacijas yra beveik neįmanoma.

4.2 lentelė. Daugiausiai įvertinimų turintys filmai

Filmo pavadinimas	Bendras įvertinimas	Įvertinimų kiekis
Star Wars	4,36	583
Contact	3,08	509
Fargo	4,16	508
Return of the Jedi	4,00	507
Liar Liar	3,16	485

Taip pat atlikus duomenų analizę, koreliacijų skaičiavimo metodus galima pritaikyti ir paprastos rekomendacinės sistemos kūrimo. Bendrumo filtravimo rekomendacinės sistemos, kaip ir minėta 2.3 poskyryje, gali būti kuriamos remiantis naudotojų pateiktais įvertinimais, sudarant tam tikrus skirtingų naudotojų profilius. Šiuo atveju „MovieLens“ duomenų rinkinyje yra duomenys su naudotojų identifikaciniais numeriais, bei jų pateiktais įvertinimais atitinkamiems filmams, tai reiškia, kad sudarant bendrumo filtravimu paremtą RS galima panaudoti šiuos duomenis.

Šioje užduotyje yra bandoma sudaryti tam tikras koreliacijos lenteles, kurios parodytų, kokie filmai, palyginus su kitų naudotojų įvertinimais, galėtų koreliuoti su konkrečiu filmu. Šiai užduočiai atlikti buvo pasirinktas populiariausias, prieš tai minėtas, filmas – „Žvaigždžių karai“. Pirmiausia, pasidarome matricą, reikalingą koreliacijai tarp skirtingų naudotojų nustatyti. Matricos eilutės yra kiekvieno naudotojo identifikacinis numeris, o stulpeliai nusako, kokiam filmui (pagal pavadinimą), kokį įvertinimą skyrė konkretus naudotojas. Naudojant *python* funkciją *corrwith()*, galime apskaičiuoti koreliaciją tarp pasirinkto filmo ir kitų filmų. Apskaičiavus koreliacijas, galime pastebėti, kad jos yra itin netikslios vien pagal žanro specifiką – ten kur koreliacija yra 1, nemaža dalis filmų yra komedijos žanro, kas ne itin siejasi su filmu „Žvaigždžių karai“, taip pat šioje „MovieLens“ duomenų bazėje yra kitos „Žvaigždžių karų“ filmo serijos, tad galima daryti prielaidą, kad rekomendacijose pirmiausia turėtų rodyti

būtent tokius filmus (žr. 4.7 pav.).

title	correlation	no_of_ratings
Commandments (1997)	1.0	3
Cosi (1996)	1.0	4
No Escape (1994)	1.0	5
Stripes (1981)	1.0	5
Man of the Year (1995)	1.0	9
Hollow Reed (1996)	1.0	6
Beans of Egypt, Maine, The (1994)	0.9999999999999999	2
Good Man in Africa, A (1994)	0.9999999999999999	2
Old Lady Who Walked in the Sea, The (Vieille qui marchait dans la mer, La) (1991)	0.9999999999999999	5
Outlaw, The (1943)	0.9999999999999999	2
Line King: Al Hirschfeld, The (1996)	0.9999999999999999	2
Hurricane Streets (1998)	0.9999999999999999	9
Scarlet Letter, The (1926)	0.9999999999999999	2

4.7 pav. „Star wars“ koreliacija su kitais filmais

Tačiau priežastis to, kodėl šios rekomendacijos yra netikslios yra įvertinimų kiekis – jis yra itin mažas. Vidutinis siūlomų filmų įvertinimų kiekis yra apie 5, tad tai tikrai nėra svarus kiekis. Todėl, norint pateikti tikslesnes rekomendacijas pagal filmą, reikėtų atsižvelgti ir į kitų filmų įvertinimų skaičių. Apribojus įvertinimų minimalų kiekį, pavyzdžiui 80, galime gauti jau visai kitas rekomendacijas (žr. 4.8 pav.). Pirmoje vietoje be abejo, matome mūsų pasirinktą filmą, tuomet su didžiausia koreliacija, tai reiškia, kad tų filmų įvertinimai labiausiai koreliuoja tarpusavyje – matome kitas dvi „Žvaigždžių karų“ serijas: „Empire Strikes Back“ ir „Return of the Jedi“. Tad iš esmės, naudotojui peržiūrėjus filmą „Žvaigždžių karai“, turėtų rekomenduoti kitas to filmo serijas, kas yra logiška ir matematiškai pagrįsta.

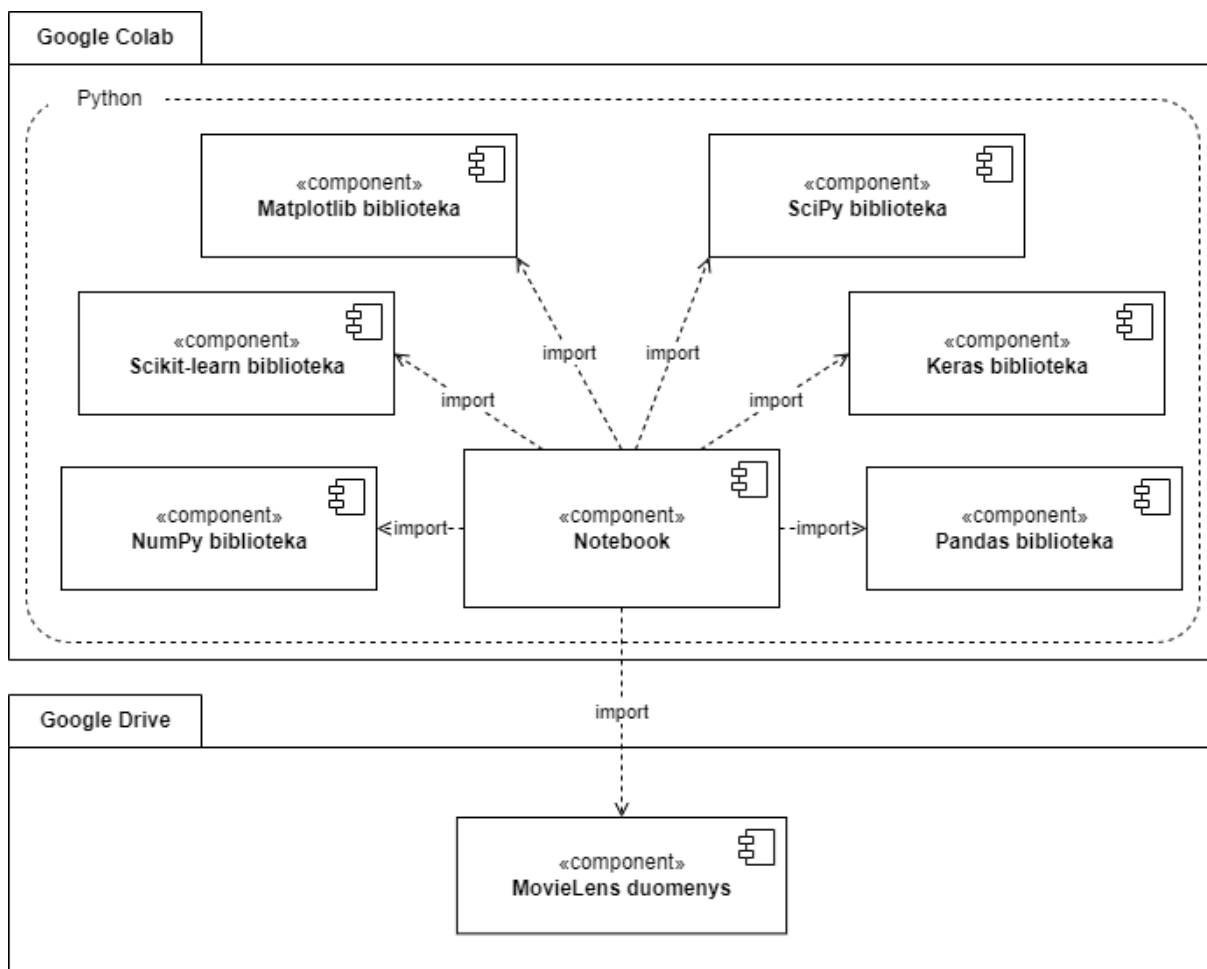
title	correlation	no_of_ratings
Star Wars (1977)	1.000000	583
Empire Strikes Back, The (1980)	0.747981	367
Return of the Jedi (1983)	0.672556	507
Raiders of the Lost Ark (1981)	0.536117	420
Austin Powers: International Man of Mystery (1997)	0.377433	130
...	...	...
Dirty Dancing (1987)	-0.156920	98
G.I. Jane (1997)	-0.176734	175
Evil Dead II (1987)	-0.190719	89
First Wives Club, The (1996)	-0.194496	160
Rosencrantz and Guildenstern Are Dead (1990)	-0.197935	90

4.8 pav. Koreliacija, nustačius įvertinimų kiekį

4.4 Siūlomas rekomendacinės sistemos kūrimo ir išbandymo modelis

Šiame darbe atliekamo tyrimo tikslas yra sukurti skirtingo tipo rekomendacines sistemas, jas išbandyti ir palyginti gautus rezultatus. Tyrimas bus atliekamas naudojant „Google

Colaboratory“ (arba Google Colab) platformą. „Google Colab“ yra nemokama „Jupyter Notebook“ aplinka, kuri veikia debesyje, palaiko didelę dalį mašininio mokymosi algoritmų ir dažniausiai yra naudojama duomenų analizei. Taip pat svarbu paminėti, kad „Google Colab“ platformoje kodas yra rašomas ir vykdomas Python programavimo kalba. Šioje platformoje kiekvienai rekomendacinei sistemai bus kuriama atskira užrašinė (angl. *Notebook*) – Python kodo failas. Analizuojant „MovieLens“ duomenis ir kuriant skirtingas rekomendacines sistemas bus naudojamos įvairios Python bibliotekos, kurios yra pavaizduotos komponentų diagramoje (žr. 4.9 pav.) ir aprašytos lentelėje žemiau (žr. 4.3 lentelė).



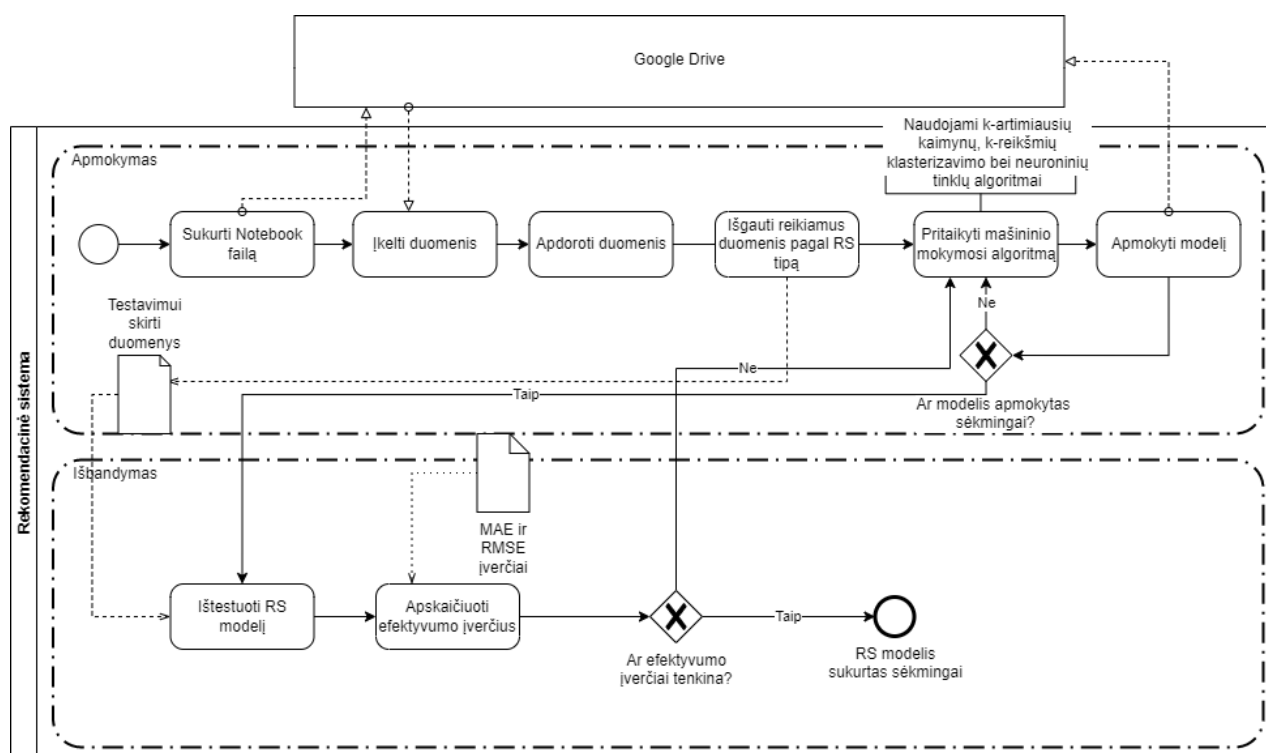
4.9 pav. RS komponentų diagrama

4.3 lentelė. Komponentų aprašymas

Komponentas	Aprašymas
Notebook	Python kodo failas, kuris sukuriamas kiekvienai rekomendacinei sistemai
MovieLens duomenys	Rekomendacinės sistemos apmokymui skirti duomenys, kurie yra pasiekiami per Google Drive asmeninę paskyrą
NumPy biblioteka	Bibliotekos komponentas, naudojamas daugiamačių bei vienmačių masyvų apdorojimui bei įvairiems skaičiavimams atlikti
Scikit-learn biblioteka	Bibliotekos komponentas, kuris suteikia įrankius nuspėjamajai (angl. <i>predictive</i>) duomenų analizei, naudojama atliekant uždavinius su mašininio mokymosi algoritmais

Komponentas	Aprašymas
Matplotlib biblioteka	Bibliotekos komponentas, kuris suteikia funkcijas, reikalingas duomenų atvaizdavimui diagramomis, histogramomis ir kitais vizualiais būdais
SciPy biblioteka	Bibliotekos komponentas, skirtas statistinei analizei, skaičiavimams, skaitinei integracijai, optimizacijai ir pan.
Pandas biblioteka	Bibliotekos komponentas, kuris suteikia paruoštas naudoti didelio našumo duomenų struktūras ir duomenų analizės įrankius
Keras biblioteka	Bibliotekos komponentas, naudojamas neuroninių tinklų modelių kūrimui

Tyrimui atlikti yra sudaromas bendras rekomendacinės sistemos kūrimo ir išbandymo metodas, kuris atvaizduojamas procesų diagramoje (žr. 4.10 pav.). Rekomendacinės sistemos modelio apmokymo etapas prasideda sukuriant „Notebook“ failą, kuriame bus rašomas sistemos programinis kodas, tuomet iš Google Drive saugyklos yra įkeliami „MovieLens“ duomenų failai. Šie duomenys yra apdorojami, kaip ir paminėta 4.2 poskyryje. Skirtingiems rekomendacinių sistemų tipams yra reikalingi ir skirtingi duomenys – kintamieji, tad prieš apmokant modelį yra atrenkami ir iš sujungtų lentelių išgaunami reikalingi duomenys. Vėliau pritaikius pasirinktą mašininio mokymosi algoritmą atliekamas modelio apmokymas ir yra stebimi gaunami rezultatai. Jei modelis yra apmokytas sėkmingai, tuomet RS modelis yra išbandomas testavimui skirtais duomenimis. Atlikus modelio testavimą, apskaičiuojamos rekomendacinio modelio efektyvumo metrikos. Jei tikslumo įverčiai nėra tenkinami, tuomet galima dar pakoreguoti mašininio mokymosi algoritmų parametrus. Jei modelio tikslumo įvertinimas tenkina, tuomet laikoma, kad rekomendacinis modelis yra sukurtas sėkmingai.



4.10 pav. Rekomendacinės sistemos kūrimo ir išbandymo procesų diagrama

4.5 Gauti projektavimo rezultatai

Šiame skyriuje aprašoma bendroji darbo struktūra, įgyvendinami procesai, uždaviniai, reikalingi darbo tikslui pasiekti. Tyrimui atlikti yra pasirinkta „MovieLens“ duomenų bazė, kurioje saugomi duomenys apie naudotojus, filmus bei tų filmų įvertinimus. Šiam pasirinktam duomenų rinkiniui yra atliktas duomenų apdorojimas bei duomenų analizė. Duomenų analizės proceso metu, buvo nustatyta, kad didžiosios dalies naudotojų amžiaus vidurkis yra 34 metai, populiariausias filmų žanras tarp visų naudotojų yra dramos ir kad naudotojai dažniausiai yra linkę filmus vertinti nuo 3 iki 4 balų. Taip pat buvo atlikti Spearman koreliacijos koeficiento vertinimai tarp skirtingų filmo žanrų ir jų įvertinimų vidurkių. Buvo pastebėta, kad tam tikros filmo žanrų poros šiek tiek koreliuoja tarpusavyje, pavyzdžiui, animacijos ir vaikų, mistinių ir trilerių ir t.t. Be to, naudotojai pagal turimus duomenis yra linkę nežymiai geriau vertinti „Film noir“ ir dramos tipo filmus, nes koreliacijos koeficientas yra šiek tiek didesnis, nei kitų. Taip pat buvo išbandytas koreliacijų skaičiavimo taikymas bendrumo filtravimu paremtoje rekomendacinėje sistemoje, kuriai pavyko nustatyti konkrečiam filmui didelę teigiamą koreliaciją turinčius kitus filmus, kuriuos būtų galima rekomenduoti naudotojui.

Buvo pasiūlytas rekomendacinės sistemos sukūrimo modelis, kuris nurodo, kaip rekomendacinės sistemos modelis turėtų būti apmokomas ir išbandomas. Tolimesnis uždavinys yra pagal sudarytą procesų diagramą įgyvendinti skirtingo tipo rekomendacines sistemas, jas išbandyti ir palyginti gaunamus rezultatus, naudojant tuos pačius įvesties duomenis – „MovieLens“ duomenų bazę.

5 Rekomendacinių sistemų įgyvendinimas ir išbandymas

Šiame skyriuje, pagal sudarytą procesų diagramą (žr. 4.10 pav.), yra kuriamos ir išbandomos skirtingo tipo rekomendacinės sistemos. Pagal nurodytą procesų diagramą, yra kuriami trys rekomendacinių sistemų modeliai, kurie taip yra apmokomi bei išbandomi, naudojant „MovieLens“ duomenų rinkinį. Taip pat šiame skyriuje yra aprašoma modelių vertinimo struktūra, kuri bus reikalinga RS modelių palyginimui.

5.1 Modelių vertinimo struktūra

Kuriant mašininio mokymosi algoritmais paremtus modelius ir norint pasiekti sėkmingus rezultatus yra keli svarbūs faktoriai, galintys turėti tam įtakos, pavyzdžiui, kokybiškam modelio apmokymui yra svarbu perduoti tinkamą duomenų rinkinį, kurį sudarytų pakankamas kiekis duomenų ir tie duomenys būtų kokybiškai surinkti. Sėkmingai sukurtas modelis turi gebėti mokytis iš savo klaidų, atskirti tam tikras duomenų tendencijas bei priimti tinkamus – kuo geresnius – rezultatus validacijos etape. Ar modelis išbandymo etape geba priimti tinkamus sprendimus ir yra sėkmingas galima įvertinti ir tam tikromis metrikomis. Mašininio mokymosi modelių srityje gali būti naudojama daugybė įvairių vertinimo metrikų, kurios apibūdina modelio tikslumą, gebėjimą sėkmingai klasifikuoti ar kitaip nuspėti atitinkamas reikšmes. Klasifikavimo uždaviniams sprendžiantiems modeliams įvertinti dažniausiai yra taikomos tikslumo (angl. *accuracy*), preciziškumo (angl. *precision*), jautrumo (angl. *recall*) ar panašios metrikos. Tokios tikslumo vertinimo metrikos yra tinkamos, kai modelio išvesties duomenys yra kategoriniai, tačiau šiuo atveju, kai kuriamas modelis yra filmų rekomendacinė sistema, kurios išvesties duomenys yra naudotojų ir filmų matrica, o matricos reikšmės yra skaitiniai filmų įvertinimai, anksčiau išvardintos metrikos nėra tinkamos. Rekomendacinės sistemos modeliui, kurio išvesties duomenys yra kintančios skaitinės reikšmės, įvertinti galima naudoti tokias metrikas kaip: šaknis iš vidutinės kvadratinės paklaidos (angl. *Root mean squared error*) arba vidutinę absoliuti paklaidą (angl. *Mean absolute error*) (Kiran et al., 2020).

5.1.1 Vidutinė absoliuti paklaida

Vidutinė absoliuti paklaida arba MAE yra metrika apibūdinanti mašininio mokymosi algoritmais grindžiamo modelio efektyvumą. MAE metrika apskaičiuojamas vidutinis absoliutus skirtumas tarp numatomų ir faktinių verčių. MAE matematinė išraiška (Newton et al., 2018) (5.1):

$$MAE = \frac{\sum_{n=1}^N |\hat{r}_n - r_n|}{N} \quad (5.1)$$

čia $\hat{r}_n$ yra prognozuojama vertė; r_n faktinė vertė; N yra bendras prognozuojamų ir faktinių

verčių skaičius. Vidutinės absoliučios paklaidos intervalas yra nuo 0 iki begalybės, kuo MAE įvertis mažesnis – tuo geriau, ir toks modelis laikomas efektyvesniu. MAE žymi vidutinį tikrųjų reikšmių nuokrypį nuo prognozuojamųjų.

5.1.2 Šaknis iš vidutinės kvadratinės paklaidos

Šaknis iš vidutinės kvadratinės paklaidos arba RMSE yra tikslumo įvertis apskaičiuojantis kvadratinę visos paklaidos kvadrato vidurkio šaknį (Christie & Neill, 2022). Toks įvertis apibūdina, kokią paklaidą modelis daro, atlikdamas prognozavimo uždavinius. RMSE matematinė išraiška (Newton et al., 2018) (5.2):

$$RMSE = \sqrt{\frac{\sum_{n=1}^N (\hat{r}_n - r_n)^2}{N}} \quad (5.2)$$

čia $\hat{r}_n$ yra prognozuojama vertė; r_n faktinė vertė; N yra bendras prognozuojamų ir faktinių verčių skaičius. RMSE įverčio intervalas, kaip ir MAE, taip yra nuo 0 iki begalybės ir kuo mažesnė RMSE reikšmė, tuo modelis laikomas tikslesniu. Šaknies iš vidutinės kvadratinės paklaidos metrika yra jautresnė didesnių klaidų atžvilgiu, nes iš gaunamo rezultato yra traukiama kvadratinė šaknis, tai reiškia, kad RMSE įvertis yra labiau paveikiamas neteisingų modelio spėjimų, klaidų ir todėl gali sparčiai padidėti, pavyzdžiui, jei prognozuojami duomenys yra didesniu atstumu nutolę nuo prognozuojamų, tai toks duomenų pasiskirstymas padaro didesnę įtaką RMSE įverčiui į blogąją pusę.

5.2 Bendrumo filtravimu paremta sistema

Šiame poskyryje bus kuriama ir demonstruojama rekomendacinė sistema paremta bendrumo filtravimu. Bendrumo filtravimu paremtoje rekomendacinėje sistemoje bus naudojami tik naudotojų įvertinimų duomenys konkreitiems filmams.

Pradedant šį tyrimo etapą yra sukurama naudotojų ir filmų matrica, kurios kiekviena eilutė parodo naudotojo suteiktą įvertinimą atitinkamiems filmams (stulpeliams). Tokia įvertinimų matrica turi daug tuščių reikšmių, todėl yra panaudojama SciPy bibliotekos funkcija *csr_matrix*, kuri matricą suspaudžia – įvertinimų matricoje palieka tik užpildytas reikšmes, tokiu būdu duomenys užima mažiau atminties ir modelio apmokymas gali vykti sklandžiau.

Modeliui sukurti bus naudojamas Scikit-learn bibliotekos kaimynais paremtas (angl. *neighbors-based*) metodas, kurio pagalba bus kuriamas k-artimiausio kaimyno algoritmu pagrįstas modelis:

```
knn_model = NearestNeighbors(metric='cosine', algorithm='brute').
```

Artimiausio kaimyno (angl. *NearestNeighbors*) klasė turi keletą parametrų, kuriuos pridėjus galima modifikuoti modelio veikimą, vieni tokių parametrų yra „metric“ ir „algorithm“, metrika ir algoritmas atitinkamai:

- Parametras „metric“ – naudojamas išreikšti atstumo matavimą, kurį modelis turėtų naudoti tam, kad apskaičiuotų panašumą, artumą tarp duomenų taškų. Šiuo atveju naudojama metrika buvo pasirinkta „cosine“ – kosinusas. Nurodyta „cosine“ metrika apskaičiuoja kosinuso panašumą tarp dviejų duomenų taškų, detaliau kosinuso panašumas aprašomas 3.5 poskyryje. Kosinuso panašumo metrika yra dažnai naudojama rekomendacinėse sistemose, nes toks matavimas gali veikti, kai duomenys nėra gausūs ir kartu šiai metricai įtakos neturi itin didelis duomenų kiekis, tad tokia metrika yra ganėtinai universali.
- Parametras „algorithm“ – naudojamas, norint specifikuoti, kokį algoritmą modelis turėtų naudoti atrandant k-artimiausius kaimynus duomenų taškui. Šiame uždavinyje buvo pasirinktas „brute“ algoritmas. Toks algoritmas atlieka „brute – force“ paiešką visame duomenų rinkinyje, kol atranda k-artimiausius kaimynus.

Modelis apmokomas su mokymui skirtais duomenimis, prieš tai 20% duomenų bazės atskyrėme ir palikome testavimui. Ši bendrumo filtravimu paremta RS sudaro tam tikrus naudotojų profilius, su jų peržiūrėtais ir įvertintais filmais, tuomet modelis apmokytas tokiais duomenimis bando atrasti panašų profilį – k-artimiausią kaimyną.

Python programiniame kode aprašome funkciją *get_recommendations*, kuri kreipiasi į apmokytą rekomendacinės sistemos, paremtos k-artimiausių kaimynų algoritmu, modelį ir kaip įvesties duomenis paima naudotojo identifikacinį numerį ir norimų rekomendacijų kiekį, *user_id* ir *n_recommendations=5* atitinkamai – šiuo atveju nustatome, kad kiekvienam naudotojui funkcija atrinktų penkias rekomendacijas iš sudarytų panašių naudotojų profilių. RS modelio išbandymo etape yra panaudojama tikrinimo duomenų aibė (angl. *test data*). Išbandymo metu yra paleidžiamas procesas, kuris paima kiekvieną naudotojo identifikacinį numerį iš išbandymo duomenų aibės ir atranka bei nustato prognozuojamą naudotojo įvertinimą pateiktoms filmų rekomendacijoms, gaunami rezultatai – naudotojų ir filmų matrica – yra išsaugomi į atskirą, prognozuojamų įvertinimų duomenų rinkinį – *y_pred*.

Po rekomendacinės sistemos modelio apmokymo ir išbandymo su testavimui skirtais duomenimis, turime dvi duomenų aibes – *y_true* ir *y_pred* – kuriose saugomi tikrieji testavimui atrinktų naudotojų įvertinimai ir prognozuojami įvertinimai filmams, atitinkamai. Turint tikruosius ir prognozuojamus naudotojų įvertinimus, galima juos palyginti bei įvertinti, ar bendrumo filtravimu paremtas modelis teisingai atliko prognozavimo užduotį. Modelio efektyvumas yra įvertinamas pagal anksčiau skyriuje pateiktą vertinimo struktūrą – naudojami Scikit-learn bibliotekoje esantys metodai apskaičiuoti modelio vidutinę absoliučią paklaidą bei šaknį iš vidutinės kvadratinės paklaidos:

- $MAE = 0,054;$

- RMSE = 0,457.

MAE įvertis šiuo atveju yra itin žemas – apytiksliai 0,05, tai reiškia, kad modelis geba itin tiksliai sudaryti filmų rekomendacijas naudotojams ir klaidų kiekis tarp tikrųjų bei prognozuojamų filmų įvertinimų yra mažas. RMSE įvertis yra didesnis už MAE, tai gali reikšti, kad modelis, kuris klaidingai prognozavo rekomenduojamų filmų įvertinimus, padarė didesnes klaidas, pavyzdžiui, modelis rekomenduojamam filmui suteikė 4 balų įvertinimą, nors iš tikrųjų buvo duotas 1 balas ar pan.

5.3 Turiniu paremta rekomendacinė sistema

Turiniu paremtų rekomendacinių sistemų pagrindinis objektas yra produktų savybės, kurios apibūdina produkto specifiškumą. Filmų kontekste pagrindinės savybės yra jų žanrai. Kadangi „MovieLens“ duomenų bazėje nėra itin daug papildomos informacijos apie filmų savybes, tai tyrimo metu kuriamai rekomendacinei sistemai bus perduodami filmų žanrai.

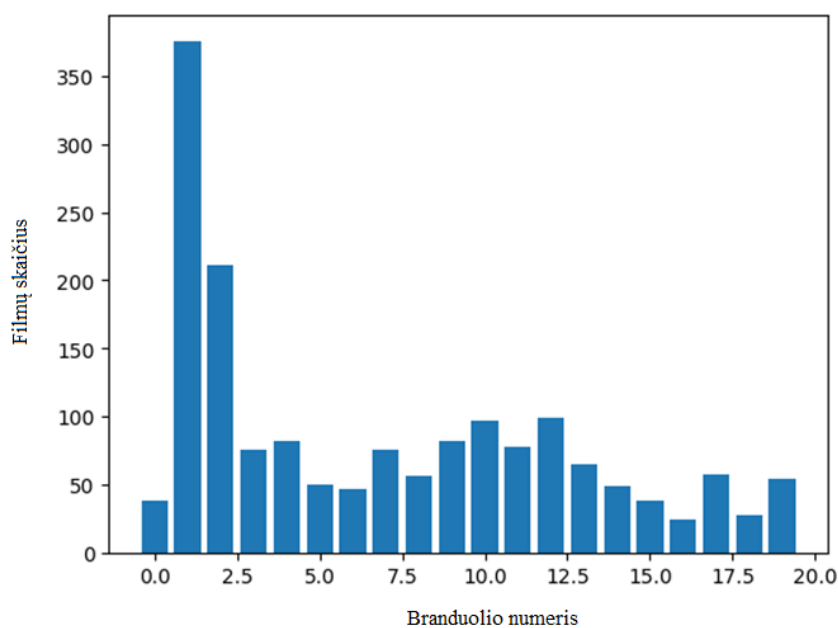
Šio tyrimo metu bus kuriamas rekomendacinis modelis, paremtas K-reikšmių klasterizavimo algoritmu. K-reikšmių klasterizavimo metodas yra naudojamas grupuojant filmų žanrus. Žanrų duomenys padeda modeliui sudaryti rekomendacijas naudotojams pagal jų žiūrėjimus filmų žanrus, tad jei naudotojas yra linkęs žiūrėti ir geriau vertinti, pavyzdžiui, dramos bei romantinio tipo filmus, tai sukurtas RS modelis turėtų tokiame naudotojui sudaryti rekomendacijas paremtas žanrų duomenimis bei tokiems filmams skirti aukštesnį įvertinimą.

Pirmiausia tam, kad modelis būtų tinkamai apmokytas filmų žanrų duomenimis – jie turi būti apdorojami. Duomenų lentelėje yra sukuriamas naujas stulpelis *genre_data*. Šiame stulpelyje patalpinamos žanrų reikšmės – į vieną tekstinę eilutę sujungti atitinkami žanrų deriniai atskirti tarpu, pavyzdžiui: „Crime Drama Thriller“ arba „Drama Romance“ ir pan. Tokiu būdu kiekvienam filmui yra sukuriamas filmų žanrų identifikacinė eilutė. Filmų žanrų eilutė yra transformuojama, panaudojus Scikit-learn bibliotekos metodą *TfidfVectorizer*, šis metodas yra vadinamas termino pasikartojimo dažnio atvirkščio dokumento pasikartojimo dažniui vektorizatorius (angl. *Term Frequency-Inverse Document Frequency Vectorizator*). Šis metodas žanrus iš tekstinio formato transformuoja į skaitinį, toks procesas atliekamas kiekvienam žanrui suteikiant tam tikrą svorį pagal svarbumą, kiek kartų šis žanras ar žanrų pora pasikartoja duomenų bazėje. Kai filmų žanrai yra transformuojami į skaitines reikšmes yra panaudojama kita Scikit-learn bibliotekos funkcij *fit_transform*, ši funkcija kaip išvesties duomenis grąžina matricą, kurios eilutės atitinka filmus, o stulpeliai žanrus, taip pat matricoje saugomos tik užpildytos reikšmės.

Atlikus žanrų vektorizaciją, duomenys yra perduodami rekomendacinės sistemos modeliui. Modelis sukuriamas metodu – *Kmeans(n_clusters=20, random_state=42)*. *N_cluster* parametras reiškia, kad rekomendacinės sistemos modelis, paremtas k-reikšmių klasterizavimu

bandys sudaryti 20 branduolių, prieš tai transformuotus žanrų derinius susiskirstys į 20 grupių ir tokiu būdu vėliau, sudarant rekomendacijas, naudotojams bus atrenkami filmai iš arčiausiai esančių branduolių. *Random_state* parametras, kaip įprasta mašininio mokymosi uždaviniuose, nustatomas į 42, kad paleidinėjant modelio programinį kodą, centroidai būtų sugeneruojami toje pačioje vietoje.

Sukūrus ir apmokius modelį transformuotais žanrų duomenimis yra sukuriamas dar vienas papildomas stulpelis duomenų bazėje „cluster“, į šį stulpelį įrašome k-reikšmių klasterizavimo modelio sugeneruoto branduolio numerį – nuo 0 iki 19 – tai reiškia, kad filmus modelis susiskirstė į 20 grupių pagal žanrų derinius. Sudarius filmų pasiskirstymo grupėse grafiką, galima pastebėti, kad daugiausiai filmų yra pirmame branduolyje, patikrinus išsaugotą duomenų lentelę, matoma, kad toks branduolio numeris buvo priskirtas dramos žanrui. Vienas žanras į vieną branduolį pateko dėl to, kad transformavimo funkcija apskaičiavo, kad šis žanras yra dažniausiai pasikartojantis duomenų bazėje, todėl šiam žanrui skyrė didelį svorį bei sukūrė atskirą branduolį.



5.1 pav. Filmų pasiskirstymas grupėse

Sudarius filmų žanrų grupes rekomendacinis modelis sugeneruoja rekomendacijas kiekvienam naudotojui, apskaičiuojant kiek kiekvienas naudotojas yra peržiūrėjęs ir įvertinęs filmų iš tam tikro branduolio, taip pat ar tie filmai yra įvertinti gerai, jei modelis mato, kad tame pačiame branduolyje yra dar likę filmų, kurių konkretus naudotojas dar neįvertinęs, nors jam ir patiko tos grupės filmai, tai greičiausiai rekomendacinis modelis pasiūlys naudotojui filmą iš to paties branduolio, jei tame branduolyje jo neįvertintų filmų nebelikę – tuomet būtų siūlomas filmas ir arčiausiai esančio kito branduolio.

Modelio prognozuojami filmų įvertinimai yra išsaugomi *y_pred* duomenų aibėje, kurią

vėlgi galime patikrinti su tikraisiais naudotojų įvertinimais. Modelio tikslumo įvertinimai buvo apskaičiuoti tokie:

- $MAE = 0,054$;
- $RMSE = 0,457$.

Gauti MAE ir RMSE tikslumo įvertinimai yra ganėtinai geri, taip pat jie sutampa su pirmosios, bendrumo filtravimu paremtos, sistemos įverčiais. Tai reiškia, kad ir šis rekomendacinis modelis ganėtinai gerai geba sudaryti filmų rekomendacijas naudotojams ir jo prognozuojami filmų įvertinimai yra arti tikrųjų įvertinimų.

5.4 Hibridinis rekomendacinės sistemos modelis

Tyrimo metu buvo sukurti du modeliai: bendrumo filtravimu bei turiniu paremtas. Šioje dalyje bus kuriamas trečiasis – hibridinis – rekomendacinės sistemos modelis. Hibridiniai rekomendaciniai modeliai, gali būti sudaryti naudojant įvairaus tipo duomenis, pavyzdžiui, naudotojo bei filmo matricas, kaip bendrumo filtravimu paremtose sistemose, taip pat gali būti naudojami su filmais susijusių savybių duomenys – filmų žanrai, aktoriai, režisieriai ir t.t., taip pat hibridinėse rekomendacinėse sistemose gali būti naudojami ir demografiniai duomenys, pavyzdžiui, amžius, profesija, lytis ir pan. Šiuo atveju, kuriant mišrią filmų rekomendacinę sistemą, bus panaudoti naudotojų ir filmų matricų duomenys bei demografiniai, naudotoją labiau apibūdinantys, duomenys: lytis, profesija ir amžius. Taip pat tam, kad rekomendacinis modelis galėtų apimti ir sklandžiau apdoroti didesnę kiekį įvesties duomenų, bus naudojamas neuroninių tinklų metodas. Neuroninių tinklų modelis yra sudaromas iš tam tikrų neuronų sluoksnių, kurie priima atitinkamus sprendimus, paremtus skirtingo tipo duomenimis, ir tokiu būdu gali generuoti dar labiau suasmenintas filmų rekomendacijas.

Neuroninių tinklų modeliui patogiau yra priimti bei apdoroti skaitinio formato duomenis, todėl turime duomenų rinkinyje pirmiausia kategoriniai duomenys, lytis ir profesija, bus paverčiami į skaitinius. Šiam uždaviniui atlikti bus panaudotos Scikit-learn bibliotekos funkcijos *LabelEncoder()* ir *fit_transform()*. Šių dviejų funkcijų kombinacija yra sukuriama klasės objektas ir tuomet tas objektas yra užkoduojamas, tokiu būdu lyties ir profesijos duomenis paverčiant skaitinėmis reikšmėmis. Tokių kategorinių duomenų skaitinės reikšmės yra lygios sveikiesiems skaičiams ir yra paskirstomos nuo nulio iki tokio skaičiaus, kiek klasių duomenų rinkinyje yra aptinkama.

Tyrimo metu kuriamam hibridiniam modeliui yra taikoma Keras bibliotekos siūloma neuroninių tinklų struktūra, kuri yra viena populiariausių kuriant neuroninių tinklų modelius bei yra laikoma intuityvia bei lengviau kuriama. Keras bibliotekos metodais yra sudaromi neuroninių tinklų sluoksniai, kurie įtraukia įvesties duomenis:

- Naudotojo identifikacinį numerį;

- Filmo identifikacinį numerį;
- Naudotojo lytį;
- Naudotojo profesiją;
- Naudotojo amžių.

O kaip išvesties duomenis, sudarytas neuroninių tinklų RS modelis, pateikia filmo prognozuojamą įvertinimą. Duomenų įvesties sluoksniams aprašyti yra taikomos *Input()*, *Embedding()* ir *Flatten()* funkcijos. Funkcija *Input()* apibrėžia įvesties duomenų formą – įvesties duomenys yra vienos dimensijos. *Embedding()* funkcija yra aprašomi tam tikri modelio įterpiamieji sluoksniai (angl. *embedding layers*). Įterpiamuosiuose sluoksniuose neuroniniuose tinkluose duomenys yra vektorizuojami, šio tyrimo metu pritaikoma *Flatten()* funkcija, kuri visus įvesties duomenis transformuoja į vienos dimensijos vektorius. Apmokymo metu modelis su įterpiamaisiais sluoksniais bando atkurti tam tikrus panašumus tarp skirtingų įvesties duomenų, geba atrasti ir suprasti ryšius tarp skirtingų duomenų vektorių. Šiuo atveju įterpiamieji sluoksniai, į kuriuos yra perduodami prieš tai minėti įvesties duomenys, RS modelio mokymo metu turi gebėti atrasti ryšį tarp skirtingų naudotojų, filmų bei demografinių duomenų, pavyzdžiui, jei moteriai, dirbančiai daktare ir mėgstančiai žiūrėti dramos žanro filmus, tokie įterpiamieji sluoksniai turėtų padėti atrasti panašius duomenų taškus, vektorius plokštumoje ir rekomenduoti filmus, kurių duomenys – kiti vektoriai – yra artimiausi, tokiu būdu sudarant filmų rekomendacijas, pavyzdžiui, paremtas kitos moters, daktarės žiūrėtais filmais.

Tolimesnė filmų rekomendacinės sistemos, paremtos neuroniniais tinklais, struktūra susideda iš: sujungimo sluoksnio (angl. *concatenate layer*), dviejų tankumo sluoksnių (angl. *dense layer*), atmetimo sluoksnio (angl. *dropout layer*) bei paskutinio tankumo sluoksnio, kuris kartu yra ir išvesties sluoksnis. Sujungimo sluoksnis paima visus prieš tai sulygintus (angl. *flattened*) įvesties duomenis ir sujungia į vieną vektorių. Tuomet du tankumo arba dar kitaip vadinami pilnai sujungtaisiais sluoksniais (angl. *fully connected*), yra skirti neuronų išdėstymui. Duomenys keliauja per nurodytą neuronų kiekį, pirmajame sluoksnyje nurodyti 130, o antrajame – 70 neuronų, ir priima atitinkamus sprendimus, kurie lydetų link galutinio išvesties sluoksnio. Svarbu pasirinkti ne per didelį kiekį neuronų, kad modelis nebūtų perpildytas (angl. *overfitted*) ir mokėtų savarankiškai pastebėti duomenų tendencijas ir ryšius, tad atlikus kelis bandymus su skirtingais neuronų kiekiais, buvo pasirinktas optimaliausias neuronų skaičius. Taip pat pilnai sujungtuose sluoksniuose buvo pridėtas L2 reguliarizacijos (angl. *L2 regularization*) parametras, kuris tarytum prideda bausmes (angl. *penalty*) modelio sprendimams apmokymo metu tam, kad modelis nepridėtų papildomo svorio – svarbumo tam tikriems duomenims ir gebėtų geriau dirbti su duomenimis, kurių yra mažiau arba jie nauji

modeliui. Atmetimo sluoksnio veikimo logika yra tokia pati, kaip regularizacijos – šis sluoksnis yra skirtas tam, kad modelis gebėtų geriau atpažinti naujus duomenis apmokymo metu bei būtų išvengiama modelio perpildymo problemos. Paskutinis sluoksnis rekomendacinio modelio struktūroje yra išvesties, kuriam priskiriamas 1 neuronas ir tai turėtų būti prognozuojamas filmo vertinimas, be to pridamas parametras *activation = „linear“*, kuris reiškia, kad paskutiniame sluoksnyje yra naudojama linijinė aktyvacijos funkcija (angl. *linear activation*), tai leidžia modeliui, kaip išvestį, pateikti filmo prognozuojamą įvertinimą, kuris yra tęstinė skaitinė reikšmė (angl. *continuous value*). Sudarius neuroniniu tinklų modelio struktūrą, modelis yra sukompiliuojamas (angl. *compile*). Kompiliuojant modelį yra parenkamas „Adam“ optimizatorius (angl. *optimizer*), kuris optimizuoja, atnauja modelio suteiktus svorius duomenims, gerinant jo apmokymą, taip pat modelio kompiliavimo metode yra įrašomas parametras *loss = „mse“*, kuris apmokymo metu stebėtų MSE rodmenis (ši metrika yra panaši į RMSE, tik netraukiama kvadratinė šaknis) ir bandytų išvengti kiek įmanoma daugiau klaidų tarp prognozuojamų ir tikrųjų verčių.

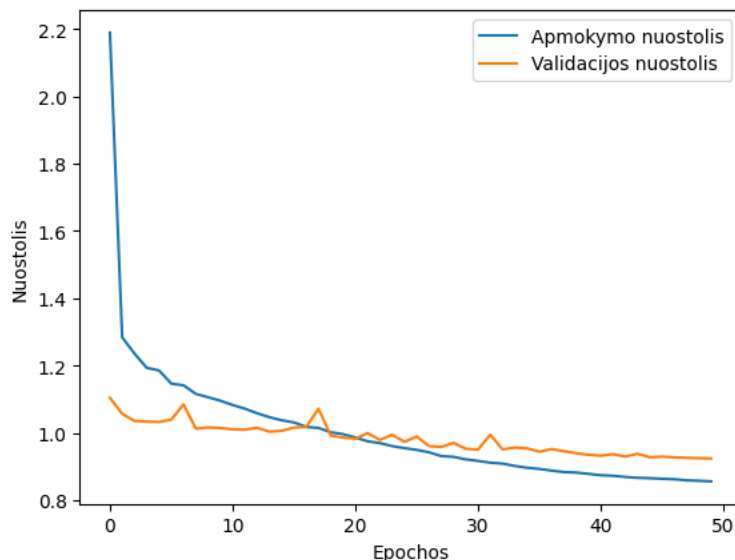
Sudarius neuroninių tinklų modelio struktūrą yra paruošiamas modelio apmokymo metodas, kuriuo yra perduodami reikiami įvesties duomenys bei nurodomi keli kiti parametrai, reikalingi modelio apmokymui:

- *epochs = 50*;
- *batch_size = 128*;
- *validation_split = 0.1*;
- *callbacks = [early_stopping]*.

Epochos arba iteracijos parametras yra skirtas nurodyti, kiek iteracijų modelis turėtų pereiti per apmokymo duomenis, atlikus kelis bandymus, buvo pastebėta, kad geriausi rezultatai yra pasirinkus parametą 50, nes svarbu yra tai, kad modelis išvengtų perpildymo problemos. Paketo (angl. *batch*) dydžio parametru yra išreiškiamas į kokio dydžio paketus yra suskirstomi apmokymo duomenys. Validacijos padalijimo parametras kiekvienos iteracijos metu paima nurodytą kiekį duomenų ir palieka juos validacijai, šie validacijos duomenys yra tikrinami ir stebimi viso apmokymo metu. Atšaukimo (angl. *callback*) funkcija yra nustatyta taip, kad jei per 5 iteracijas modelis nepagerina validacijos rezultato, tuomet apmokymas yra atšaukiamas tam, kad modelis nebuvo perpildytas.

Kai modelis yra apmokytas, galima apskaičiuoti mokymo metu patirtus nuostolius (angl. *loss*) bei tikslumo metrikas MAE ir RMSE. Modelio apmokymo ir validacijos nuostoliai parodo, kaip gerai modelis yra pamokytas per nurodytas epochas. Šiuos duomenis galima atvaizduoti vizualiai, panaudojus Matplotlib bibliotekos funkciją *plot()* (žr. 5.2 pav.). Apmokymo bei validacijos nuostoliai yra atvaizduojami kreivės pavidalu, todėl galima aiškiai

matyti, kad su kiekviena apmokymo iteracija modelis padarė vis mažiau klaidų tarp tikrųjų ir prognozuojamų verčių ir todėl praradimo kreivė yra mažėjanti, o tai galėtų reikšti, kad modelis gali būti ganėtinai gerai apmokytas.



5.2 pav. Neuroninių tinklų modelio nuostoliai

Taip pat apskaičiuotos ir modelio efektyvumo metrikos:

- $MAE = 0,751$;
- $RMSE = 0,951$.

Palyginus gautus rezultatus su prieš tai tyrime sudarytais modeliais, šio modelio efektyvumo metrikos MAE ir RMSE yra didžiausios, o tai reiškia, kad modelis padarė didesnę klaidų prognozuojant filmų įvertinimus, nei praeiti modeliai. Tiesa, MAE ir RMSE įvertinimų intervalas gali būti nuo nulio iki begalybės, todėl negalima teigti, kad modelis generuoja netinkamas rekomendacijas. Taip pat dažniausiai neuroninių tinklų modeliai yra naudojami turint itin dideles duomenų bazes, o „MovieLens“ duomenų bazė, palyginus yra maža/vidutinė. Neuroninių tinklų modelio nuostolių kreivėje galima pastebėti gerą tendenciją, kad modelis geba tobulėti ir gerinti savo išvesties rezultatus, tačiau šiuo atveju galimai duomenų rinkinys yra per mažas, kad modelis būtų pakankamai apmokytas.

5.5 Modelių išbandymas su sumažinta duomenų baze

Atlikus tyrimą su „MovieLens“ duomenų baze, kurioje yra 100 tūkstančių įrašų, galima pastebėti, kad tyrimo rezultatai yra ganėtinai neblogi – modelių MAE ir RMSE įverčiai yra nesiekia vieneto reikšmės. Geriausiai buvo įvertintos bendrumo filtravimu bei turiniu paremtos rekomendacinės sistemos. Norint šiek tiek praplėsti tyrimą, galima panagrinėti modelių efektyvumą, kai duomenų bazė yra mažesnė.

Šiam bandymui įgyvendinti buvo atlikti kvartilų (angl. *quartiles*) skaičiavimai. Kvartilis yra matematinėje statistikoje naudojama charakteristika, kuri apibūdina duomenų

pasiskirstymą į keturias dalis (Žerovnik & Rupnik Poklukar, 2017). Sujungus *data* ir *item* duomenų lenteles, galima apskaičiuoti, kiek filmų yra peržiūrėjęs bei įvertinęs kiekvienas naudotojas. Turint kiekvieno naudotojo įvertinimų kiekį yra paskaičiuojami kvartilai, tai atliekama panaudojus NumPy bibliotekos funkciją *percentiles()*, procentilių (angl. *percentiles*) reikšmės yra nustatomos į 25%, 50% ir 75%, tai reiškia, kad duomenų rinkinys yra padalijamas į nurodytų procentilių dalis, kurios ir yra duomenų ketvirtadaliai procentais. Gauti *percentiles()* funkcijos rezultatai:

- 25 procentilis – 33;
- 50 procentilis – 65;
- 75 procentilis – 148.

Šie procentilių duomenys reiškia, kad išrikiavus naudotojų įvertinimų skaičius didėjimo tvarka, 25% duomenų rinkinio naudotojų įvertino 33 filmus ir mažiau, taip pat galima pastebėti, kad duomenų rinkinio naudotojų įvertinimų mediana yra 65, o 25% procentai naudotojų yra įvertinę 148 arba daugiau filmų.

Kuriant rekomendacines sistemas gali atsirasti problemų, kai įvertinimų matrica yra reta arba naudotojai yra nauji, įvertinę mažai filmų, dėl to ir duomenų apie tokius naudotojus turima mažiau, šios duomenų retumo ir šalto paleidimo problemos aprašomos 2.4 skyriuje. Todėl prieš tai sukurtus rekomendacinius modelius galima palyginti apmokant ir išbandant su mažesniu duomenų rinkiniu. Šiam bandymui atlikti bus naudojamas pirmojo kvartilio duomenų rinkinys, modeliai bus apmokomi ir testuojami duomenimis, kuriuose kiekvienas naudotojas yra įvertinęs ne daugiau nei 33 filmus, tai reiškia, kad naudotojai yra mažiau patyrę ir tokiems naudotojams turėtų būti sudėtingiau sudaryti filmų rekomendacijas.

Apmokius rekomendacinius modelius sumažinta duomenų baze, buvo apskaičiuoti MAE ir RMSE tikslumo įverčiai (žr. 5.1 lentelė). Iš bendros rezultatų lentelės galima pastebėti, kad bendrumo filtravimu paremtos bei turiniu paremtos rekomendacinės sistemos rezultatai net šiek tiek pagerėjo, tai reiškia, kad tokio tipo rekomendacines sistemas galima taikyti tuo atveju, kai egzistuoja duomenų retumo ar šalto paleidimo rizikos, nes šie modeliai geba padaryti mažiau klaidų įverčių prognozavime, kai duomenų kiekis apie naudotojus yra mažesnis. Tiesa, hibridinio modelio rezultatai suprastėjo – neuroninių tinklų MAE ir RMSE įverčiai padidėjo, o tai reiškia, kad šiam modeliui buvo sudėtingiau teisingai prognozuoti filmų įvertinimus ir toks modelis padarė daugiau klaidų, nei kiti.

5.1 lentelė. Modelių išbandymas su sumažinta duomenų baze

RS tipas	MAE	RMSE
Bendrumo filtravimu paremtas	0,045	0,423
Turiniu paremtas	0,046	0,423

RS tipas	MAE	RMSE
Hibridinis	0,836	1,041

5.6 Sukurtų rekomendacinių sistemų palyginimas

Šiame poskyryje bus palyginami tyrimo metu sukurti ir išbandyti skirtingo tipo rekomendaciniai modeliai ir jų tikslumo vertinimo metrikos. Tyrimo metu buvo pastebėta, kad bendrumo filtravimu ir turiniu paremtų rekomendacinių sistemų tikslumo įverčiai yra ganėtinai panašūs, todėl modelių palyginimui buvo nuspręsta įtraukti ir modelio apmokymo trukmės skaičiavimus. Šie modelių, pagrįstų skirtingais mašininio mokymosi algoritmais, apmokymo trukmės skaičiavimai buvo atliekami panaudojus Python modulio „Time“ funkciją *process_time()*, ši funkcija suteikia galimybę apskaičiuoti, kiek laiko užtruko modelio apmokymas įvesties duomenimis, rezultatas yra pateikiamas sekundėmis. Apskaičiavus kiekvieno modelio apmokymo laiką, galima sudaryti bendrą rezultatų lentelę (žr. 5.2 lentelė).

5.2 lentelė. Rekomendacinių sistemų modelių palyginimas

RS tipas	MAE	RMSE	Apmokymo trukmė (sek.)
Bendrumo filtravimu paremtas	0,054	0,457	0,001
Turiniu paremtas	0,054	0,457	1,064
Hibridinis	0,751	0,951	255,806

Modelio efektyvumo metrikos – MAE ir RMSE – buvo suapvalintos iki tokios reikšmės, kuri būtų aiškesnė lyginant bendrumo filtravimu ir turiniu paremtas rekomendacines sistemas.

Peržvelgus bendrą rezultatų lentelę, galima matyti, kad hibridinis modelis, pagrįstas neuroninių tinklų metodu, pasirodė prasčiausiai, jo MAE bei RMSE įvertinimai yra didžiausi, o tai reiškia, kad modelis padarė daugiausiai klaidų generuojant filmų rekomendacijas. Taip pat neuroninių tinklų modelio apmokymas truko daugiau nei 255 sekundes, kas lyginant su kitais modeliais yra didelė trukmė. Tiesa, neuroninių tinklų modelis yra sudėtingas savo struktūra bei šiam modeliui yra perduodamas didesnis kiekis skirtingų duomenų, todėl ir jo apmokymo trukmė yra didžiausia.

Bendrumo filtravimu bei turinio paremto rekomendacinio modelio MAE ir RMSE įverčiai yra labai panašūs. Turiniu paremto rekomendacinio modelio MAE įvertis yra šiek tiek didesnis nei bendrumo filtravimu paremtos sistemos, kuomet RMSE įvertis yra šiek tiek mažesnis, tai gali reikšti, kad bendrumo filtravimu paremtas rekomendacinės sistemos modelis padarė vieną didesnę klaidą, pavyzdžiui, vienam filmui prognozavo 5 balų įvertinimą, nors iš tikrųjų naudotojo įvertinimas tokiam rekomenduotam filmui buvo 1. Bendrai šių modelių efektyvumas yra ganėtinai panašus ir „MovieLens“ duomenų bazei galima taikyti abu rekomendacinius modelius, nes jų rezultatai yra geri. Tačiau, norint apsispręsti, kuris

rekomendacinis modelis gali būti laikomas efektyviausiu ir tinkamiausiu naudoti, svarbu atsižvelgti į kitą svarbų įvertinimą – apmokymo trukmę. Bendrumo filtravimu paremtos rekomendacinės sistemos modelio, pagrįsto K-artimiausių kaimynų algoritmu, apmokymo trukmė yra vos viena milisekundė, tad galima teigti, kad tokio modelio apmokymas yra itin greitas. Ši informacija gali būti naudinga, norint apmokyti dar didesniu duomenų kiekiu ir norint efektyviausio rezultato, tiek rezultatų, tiek įgyvendinimo atžvilgiu.

5.7 Gauti rekomendacinių sistemų įgyvendinimo bei testavimo rezultatai

Šiame skyriuje buvo aprašytas atliktas tyrimas. Tyrimo metu buvo kuriamos trys rekomendacinės sistemos: paremta bendrumo filtravimu, paremta turiniu bei hibridinė. Sudarytos skirtingų tipų rekomendacinės sistemos buvo pagrįstos skirtingais mašininio mokymosi algoritmais: k-artimiausių kaimynų algoritmu, k-reikšmių klasterizavimo algoritmu bei neuroninių tinklų metodu, atitinkamai. Rekomendacinių modelių efektyvumas buvo įvertintas vidutinės absoliučios paklaidos bei šaknies iš vidutinės kvadratinės paklaidos metrikomis. Taip pat buvo įvertinta modelių apmokymo trukmė naudojant pilną „MovieLens“ duomenų bazę. Visi rekomendacinių sistemų įvertinimo rezultatai yra pateikiami modelių palyginimo lentelėje (žr. 5.3 lentelė).

5.3 lentelė. Bendra rekomendacinių sistemų įvertinimo lentelė

RS tipas	Apmokymas sumažinta DB		Apmokymas pilna DB			Įvesties duomenys
	MAE	RMSE	MAE	RMSE	Apmokymo trukmė (sek.)	
Bendrumo filtravimu paremtas	0,045	0,423	0,054	0,457	0,001	- Filmo identifikacinis numeris - Naudotojo identifikacinis numeris - Įvertinimai
Turiniu paremtas	0,046	0,423	0,054	0,457	1,064	- Žanrai - Filmo identifikacinis numeris - Naudotojo identifikacinis numeris - Įvertinimai
Hibridinis	0,836	1,041	0,751	0,951	255,806	- Filmo identifikacinis numeris - Naudotojo identifikacinis numeris - Įvertinimai - Lytis - Profesija - Amžius

Modelio efektyvumas gali būti apibūdinamas keliais kriterijais. Šio tyrimo kontekste efektyviu rekomendaciniu modeliu yra laikomas tas, kuris sudarydamas filmų rekomendacijas

geba padaryti kuo mažiau klaidų. Taip pat į modelio efektyvumo sąvoką galima įtraukti ir modelio įgyvendinimo sąlygas, šiuo atveju – modelio apmokymo trukmę bei įvesties duomenų kiekį. Jei, pavyzdžiui, verslo atstovas neturi galimybių rinkti ar saugoti didelį ir įvairų kiekį duomenų, tuomet yra naudinga rasti optimalų kiekį duomenų, kurio pakaktų efektyviam rekomendacinės sistemos modeliui sukurti. Tad remiantis bendros rekomendacinių sistemų įvertinimo lentelės duomenimis, galima teigti, kad bendrumo filtravimu bei turiniu paremti rekomendaciniai modeliai filmų įvertinimų prognozavimo užduotį atliko itin gerai, jų tikslumo įvertinimo metrikos yra beveik vienodos, todėl šie rekomendacinių sistemų modeliai gali būti laikomi efektyviais. Bendrumo filtravimu paremtam modeliui apmokyti reikia mažiausio kiekio įvesties duomenų – filmų įvertinimų, naudotojų ir filmų identifikacinių numerių, kuriais yra sudaromi naudotojų profiliai ir bandomas atrasti panašumas tarp jų, taip pat tokio modelio apmokymo trukmė yra labai maža – viena milisekundė, tad toks rekomendacinis modelis taip pat gali būti laikomas efektyviausiu ir įgyvendinimo atžvilgiu, nes nereikalauja didelio kiekio skirtingų įvesties duomenų ir yra itin greitai apmokomas.

6 Išvados ir siūlymai

1. Atlikus mašininio mokymosi metodų, taikomų rekomendacinėse sistemose, analizę, galima pastebėti, kad yra nemažas kiekis metodų rekomendacinėms sistemoms sudaryti, tačiau vieni iš dažniausiai naudojamų mašininio mokymosi metodų rekomendacinėse sistemose yra k-artimiausių kaimynų, k-reikšmių klasterizavimo bei neuroninių tinklų.
2. Atlikus rekomendacinės sistemos kūrimo ir išbandymo modelio projektavimą, galima teigti, kad sudarytas metodas leidžia sukurti skirtingo tipo rekomendacines sistemas, joms perduodant skirtingus įvesties duomenis, bei pritaikyti vieningą rekomendacinių sistemų, kurių išvesties duomenys yra skaitinės reikšmės, vertinimo struktūrą.
3. Išbandžius ir palyginus pagal sudarytą rekomendacinės sistemos kūrimo ir išbandymo modelį įgyvendintas skirtingo tipo rekomendacines sistemas (bendrumo filtravimu, turiniu paremto ir hibridinio tipo), jos buvo įvertintos vidutinės absoliučios paklaidos, šaknies iš vidutinės kvadratinės paklaidos bei apmokymo trukmės įverčiais, apibūdinančiais rekomendacinės sistemos modelio efektyvumą. Palyginus gautus rezultatus, galima nustatyti, kad bendrumo filtravimu paremtas rekomendacinis modelis, pagrįstas k-artimiausių kaimynų algoritmu, yra efektyviausias dėl itin gerų tikslumo įverčių, mažos modelio apmokymo trukmės bei nedidelio įvesties duomenų kiekio.

Siekiant išplėsti atliktą tyrimą bei galimai pagerinti tyrimo rezultatus, būtų galima atlikti bandymus su papildyta duomenų baze, išgaunant duomenis apie filmų savybes (pavyzdžiui, režisierius, aktorius ir pan.), internetinio duomenų ištraukimo (angl. *web scraping*) metodu. Taip pat norint praplėsti filmų rekomendacinių sistemų modelių įvertinimo struktūrą, būtų galima išvesties duomenų matricą paversti į kategorinius duomenis – patinka ar nepatinka filmas – ir tokiu būdu įvesti papildomas modelių tikslumo įvertinimo metrikas, kurios yra apskaičiuojamos tik su kategoriniais duomenimis.

Literatūros sąrašas

- Adeniyi, D. A., Wei, Z., & Yongquan, Y. (2016). Automated web usage data mining and recommendation system using K-Nearest Neighbor (KNN) classification method. *Applied Computing and Informatics*, 12(1), 90–108.
<https://doi.org/10.1016/J.ACI.2014.10.001>
- Aditya, P. H., Budi, I., & Munajat, Q. (2016). *A Comparative Analysis of Memory-based and Model-based Collaborative Filtering on the Implementation of Recommender System for E-commerce in Indonesia : A Case Study PT X*.
- Ahuja, R., Solanki, A., & Nayyar, A. (2019). Movie recommender system using k-means clustering and k-nearest neighbor. *Proceedings of the 9th International Conference On Cloud Computing, Data Science and Engineering, Confluence 2019*, 263–268.
<https://doi.org/10.1109/CONFLUENCE.2019.8776969>
- Carleo, G., Cirac, I., Cranmer, K., Daudet, L., Schuld, M., Tishby, N., Vogt-Maranto, L., & Zdeborová, L. (2019). Machine learning and the physical sciences. *Reviews of Modern Physics*, 91(4), 045002.
<https://doi.org/10.1103/REVMODPHYS.91.045002/FIGURES/8/MEDIUM>
- Cerioti, M. (2019). Unsupervised machine learning in atomistic simulations, between predictions and understanding. *The Journal of Chemical Physics*, 150(15), 150901.
<https://doi.org/10.1063/1.5091842>
- Chandak, Y., Theodorou, G., Kostas, J. E., Jordan, S. M., & Thomas, P. S. (2019). *Learning Action Representations for Reinforcement Learning*.
- Chomboon, K., Chujai, P., Teerarassamee, P., Kerdprasop, K., & Kerdprasop, N. (2015). *An Empirical Study of Distance Metrics for k-Nearest Neighbor Algorithm*.
<https://doi.org/10.12792/iciae2015.051>
- Christie, D., & Neill, S. P. (2022). Measuring and Observing the Ocean Renewable Energy Resource. *Comprehensive Renewable Energy, Second Edition: Volume 1-9*, 1–8, 149–175. <https://doi.org/10.1016/B978-0-12-819727-1.00083-2>
- Cooper, J. O., Heron, T. E., & Heward, W. L. (2020). *Applied behavior analysis*.
<https://www.goodreads.com/book/show/56933326-applied-behavior-analysis-global-edition>
- Covington, P., Adams, J., & Sargin, E. (2016). Deep neural networks for youtube recommendations. *RecSys 2016 - Proceedings of the 10th ACM Conference on Recommender Systems*, 191–198. <https://doi.org/10.1145/2959100.2959190>
- Dilmegani, C. (2017). *Recommendation Systems: Applications, Examples & Benefits*. Aimultiple.Com. <https://research.aimultiple.com/recommendation-system/>
- Edgar, T. W., & Manz, D. O. (2017). Machine Learning. *Research Methods for Cyber Security*, 153–173. <https://doi.org/10.1016/B978-0-12-805349-2.00006-6>
- Falk, K. (2019). *Practical recommender systems*.
https://books.google.com/books/about/Practical_Recommender_Systems.html?hl=lt&id=FDszEAAAQBAJ
- Goldberg, Y. (2017). Neural Network Methods for Natural Language Processing. *Http://Dx.Doi.Org/ 10.2200/S00762ED1V01Y201703HLT037* , 10(1), 1–311.
<https://doi.org/10.2200/S00762ED1V01Y201703HLT037>
- Gong, M., & Zhernov, A. (2019). *Advanced machine learning helps Play Store users discover personalised apps*. <https://deepmind.com/blog/article/Advanced-machine-learning-helps-Play-Store-users-discover-personalised-apps>
- Google Developers: Recommendation Systems. (2020). In *Google Developers*.

- <https://developers.google.com/machine-learning/recommendation/content-based/summary>
- Gou, J., Ma, H., Ou, W., Zeng, S., Rao, Y., & Yang, H. (2019). A generalized mean distance-based k-nearest neighbor classifier. *Expert Systems with Applications*, 115, 356–372. <https://doi.org/10.1016/J.ESWA.2018.08.021>
- Hwangbo, H., Kim, Y. S., & Cha, K. J. (2018). Recommendation system development for fashion retail e-commerce. *Electronic Commerce Research and Applications*, 28, 94–101. <https://doi.org/10.1016/J.ELERAP.2018.01.012>
- Isinkaye, F. O., Folajimi, Y. O., & Ojokoh, B. A. (2015). Recommendation systems: Principles, methods and evaluation. *Egyptian Informatics Journal*, 16(3), 261–273. <https://doi.org/10.1016/J.EIJ.2015.06.005>
- Kiran, R., Kumar, P., & Bhasker, B. (2020). DNNRec: A novel deep learning based hybrid recommender system. *Expert Systems with Applications*, 144, 113054. <https://doi.org/10.1016/J.ESWA.2019.113054>
- Lahitani, A. R., Permanasari, A. E., & Setiawan, N. A. (2016). Cosine similarity to determine similarity measure: Study case in online essay assessment. *Proceedings of 2016 4th International Conference on Cyber and IT Service Management, CITSM 2016*. <https://doi.org/10.1109/CITSM.2016.7577578>
- Maxwell Harper, F., & Konstan, J. A. (2015). The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst*, 20. <http://grouplens.org/datasets/movielens>
- Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018). *Foundations of machine learning*. 486.
- Newton, E. R., Irwin, J., Charbonneau, D., -, A., Gao, J., Dong, Y.-W., Shang, M.-S., Yang, Z., Zhang, Z.-K., Zhou -, T., Wang, W., & Lu, Y. (2018). *Analysis of the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE) in Assessing Rounding Model Group-based ranking method for online rating systems with spamming attacks Analysis of the Mean Absolute Error (MAE) and the Root Mean Square Error*. <https://doi.org/10.1088/1757-899X/324/1/012049>
- Sahoo, L., Das, D., & Datta, S. (2017). A Survey on Recommendation System. *Article in International Journal of Computer Applications*, 160(7), 975–8887. <https://doi.org/10.5120/ijca2017913081>
- Schrider, D. R., & Kern, A. D. (2018). Supervised Machine Learning for Population Genetics: A New Paradigm. *Trends in Genetics*, 34(4), 301–312. <https://doi.org/10.1016/J.TIG.2017.12.005>
- Shi, Y., Larson, M., & Hanjalic, A. (2014). Collaborative Filtering beyond the User-Item Matrix. *ACM Computing Surveys (CSUR)*, 47(1). <https://doi.org/10.1145/2556270>
- Sinaga, K. P., & Yang, M. S. (2020). Unsupervised K-means clustering algorithm. *IEEE Access*, 8, 80716–80727. <https://doi.org/10.1109/ACCESS.2020.2988796>
- Tanty Tanweer Desmukh, R. S. (2015). *Application of Artificial Neural Network in Hydrology-A Review*. www.ijert.org
- Terragni, A., & Hassani, M. (2018). Analyzing Customer Journey with Process Mining: From Discovery to Recommendations. *Proceedings - 2018 IEEE 6th International Conference on Future Internet of Things and Cloud, FiCloud 2018*, 224–229. <https://doi.org/10.1109/FICLOUD.2018.00040>
- Wei, J., He, J., Chen, K., Zhou, Y., & Tang, Z. (2017). Collaborative filtering and deep learning based recommendation system for cold start items. *Expert Systems with Applications*, 69, 29–39. <https://doi.org/10.1016/J.ESWA.2016.09.040>
- Xia, P., Zhang, L., & Li, F. (2015). Learning similarity with cosine similarity ensemble.

InfXia, P., Zhang, L., & Li, F. (2015). *Learning Similarity with Cosine Similarity Ensemble*. *Information Sciences*, 307, 39–52.

<https://doi.org/10.1016/J.INS.2015.02.024>

<https://doi.org/10.1016/J.INS.2015.02.024>

Žerovnik, J., & Rupnik Poklukar, D. (2017). Elementary methods for computation of quartiles. *Teaching Statistics*, 39(3), 88–91. <https://doi.org/10.1111/TEST.12133>