UNIVERZITET U ZENICI

POLITEHNIČKI FAKULTET

SEMINARSKI RAD IZ PREDMETA:
Vještačka inteligencija

Tema rada:	**Generisanje sistema preporuke za filmove**

Predmetni nastavnik:	van. prof. dr. Nermin Goran

Student:	Ajdin Bukvić
Broj indeksa:	II-89
Usmjerenje:	Softversko inženjerstvo
Godina studija:	1. godina, 2. ciklus

Rezultat rada:	Analiza skupa podataka i implementacija različitih metoda i algoritama vještačke inteligencije za generisanje sistema preporuke za filmove, uz poređenje dobijenih rezultata i pregled performansi

Datum: 24.01.2024.

SADRŽAJ

1. UVOD

U zadnjih nekoliko godina područje vještačke inteligencije je napravilo veliku ekspanziju u raznim sferama ljudskog života. Pošto se tehnologija sve brže razvija, u mnogim djelatnostima dolazi do potrebe za primjenom vještačke inteligencije. Upotreba vještačke inteligencije može znatno unaprijediti svakodnevne životne procese, kako u privatnom, tako i u poslovnom smislu. Mnoge grane nauke i tehnike se danas u visokoj mjeri oslanjaju na vještačku inteligenciju, kao što su auto industrija, medicina, vojna industrija, programiranje i druge. Iako su ovo relativno novi pristupi, daju znatna unaprijeđenja i mogu ubrzati sam proces rada.

Kroz ovaj seminarski rad će biti prikazan primjer upotrebe vještačke inteligencije za generisanje sistema preporuke za filmove. Sistemi preporuke su posebna grana vještačke inteligencije koji koriste velike skupove podataka, kako bi korištenjem različitih algoritama predvidjeli ili klasifikovali, korisničke podatke i dali mu što precizniju sugestiju onoga što on inače preferira.

2. SISTEMI PREPORUKE

Sistemi preporuke su bazirani na praćenju ponašanja pri pregledavanju i historiji pregledavanja od strane korisnika kako bi kreirali različite obrasce i pronašli sličnosti u podacima. Na osnovu toga se korisniku preporučuje sadržaj koji bi ga mogao zanimati. Sistemi preporuke se koriste svugdje gdje bi mogli unaprijediti korisničko iskustvo. Poznato je da sve velike kompanije i organizacije koriste ovaj pristup kako bi svojim članovima ili kupcima preporučili sadržaje koje ih zanimaju, a sve u svrhu povećanja profita i zadržavanja korisnika.

Najjednostavniji pristup na kojem rade ovi sistemi jest da korisnik nakon korištenja nekog softvera ili sistema, ostavlja dosta svojih privatnih podataka, o svojim navikama ili proizvodima i uslugama koji ga interesuju. Organizacije zatim koriste ove prikupljene podatke kako bi iz tih podataka izvukli relevantne podatke i igradili jedinstveni opis korisnika, koji im pomaže da poboljšaju svoju ponudu na temelju stečenih saznanja. Neke od vodećih kompanija koje koriste ove sisteme u svojim aplikacijama su Netflix, Amazon, Spotify, Google (YouTube), Pandora, Apple Music i mnogi drugi. [1]

Podaci koji su prikupljeni, te koji se koriste se još nazivaju i „Big Data“. To su veliki skupovi podataka, višemilionski, koji služe kako bi filtrirali različite korisnike, proizvode, usluge, te ih međusobno uparili na osnovu sličnosti. Faktori koji su sastavni dijelovi ovih podataka, mogu varirati od različitih tipova informacija, kao što je već rečeno historija pregledavanja ili kupovine, ali i dosta dublji podaci koji mogu dati preciznije rezultate, koji uključuju demografske podatke. Ovim pristupom se mogu postići preporuke, koje se ne odnose samo na korisnikove lične preferencije, nego i na njegov status, porijeklo, kulturu, jezik i mnoge druge parametre do kojih se može doći. Na ovaj način korisnik može doći i do proizvoda ili usluga koje inače ne bi sam pronašao, ali sistem može zaključiti da bi mu se takvi rezultati preporuke mogli svidjeti.

S druge strane kada su u pitanju sami proizvodi ili usluge, za njihovo filtriranje i klasifikaciju koriste se parametri kao što su broj pregleda, klikova, lajkova, komentara, recenzije ili kupovina. Područja primjene sistema za preporuku su široko rasprostranjena u različitim industrijama, koje uključuju lance za nabavku, e-trgovinu, medije i zabavu, društvene mreže, finansijske usluge, putovanja i ugostiteljstvo. Prednosti korištenja sistema preporuke u aplikacijama i sistemima mogu značajno povećati prihode, kao i broj pregleda (klikova), te razne druge metrike. Ključni faktor je zadovoljstvo i zadržavanje već postojećih članova i kupaca.

2.1. Filtriranje zasnovano na sadržaju

Filtriranje zasnovano na sadržaju je jedan od tipova za generisanje sistema preporuke, koji se kako sam naziv kaže bazira na samom sadržaju. Pitanje je šta u konkretnom kontektstu predstavlja „sadržaj“. Sadržaj su zapravo stavke ili karakteristike pojedinačnog proizvoda ili usluge. Ovaj princip uzima sve karakteristike nekog proizvoda ili usluge i upoređuje ih s onim već poznatim podacima (proizvodima ili uslugama), s kojima je korisnik već ranije imao neku interakciju. [2] Naprimjer, ako se uzmu u obzir korisnikove osobine, broj godina, spol, status i drugi specifični atributi, ovaj tip će pronaći sve proizvode ili usluge koji odgovaraju gore navedenim uslovima. U konkretnom primjeru, ako se radi o muškoj osobi od 20. godina koja je student i dolazi iz Zenice, sistem bi mogao preporučiti samo proizvode koji su namjenjeni mlađim osobama (između 18 i 25 godina), muškog spola, koje pritom studiraju, a da su dostupni u njihovoj radnji u Zenici. Posebno, ako je već od ranije poznato da je korisnik zainteresovan za ovakav tip proizvoda, obzirom da je već jednom kupovao slične proizvode. Naravno, ovo je samo grubi (naivni) primjer primjene ovog tipa preporuke. U praksi bi ovaj postupak bio znatno napredniji i kompleksniji. Glavni segment kod ovog tipa preporuke jeste izdvajanje ključnih karakteristika iz samog sadržaja. Modeliranje ovog sistema zasniva se na određivanju relevantnih atributa, koji bi mogli biti zanimljivi za proučavanje. Jedan od klasičnih primjera primjene ovog tipa je model baziran na ključnim riječima, koji se još naziva i model vektorskog prostora s ponderom frekvenzije inverzne frekvencije dokumenta. Ovaj algoritam će biti detaljnije objašnjen u nastavku rada.

Slika 1 – Primjer filtriranja zasnovanog na sadržaju [3]

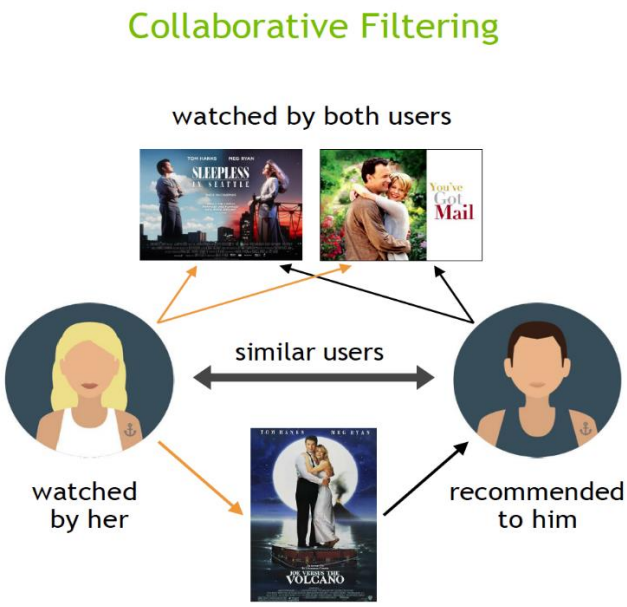
2.2. Kolaborativno filtriranje

Jedan od najefikasnijih tipova koji se koriste u sistemima preporuka je kolaborativno filtriranje. Metoda kolaborativnog filtriranja se zasniva na analizi ponašanja korisnika, te pronalasku sličnosti u njihovim preferiranim izborima. U ovom pristupu u obzir se uzima koncept koji pokazuje da se će ljudi koji su u prošlosti pokazali slično ponašanje (preferencije), isto tako pokazati i u budućnosti (vjerovatno).

Sam naziv ovog tipa sastoji se od dva dijela: kolaborativno, koje se odnosi na „saradnju“ (prepoznavanje sličnosti između korisnika), te filtriranje, odnosno preporučivanje stavki (sadržaja, proizvoda, usluga) na osnovu prikupljenih informacija o korisnicima. Interakcije između korisnika i stavki, odnosi se na kreiranje modela koji će biti u stanju predvidjeti buduće interakcije korisnika, na osnovu onih koje su imali slični korisnici. Sličnosti između korisnika mogu biti određene, naprimjer proizvodima koje su neki korisnici kupili ili ocjenama koje su korisnici ostavili na neki proizvod. [4]

Jednostavan primjer, jeste da ako postoje korisnici koji dijele slične ukuse prema hrani, tako da su već ranije kupovali i ocjenjivali proizvode od iste kompanije. Naprimjer, korisnici su probali neki broj istih proizvoda, te podijeli mišljenje da su ti proizvodi za njih zaista prihvatljivi (ukusni, zanimljivi ili bilo koji drugi opisni atribut). Tako će sistem vrlo lahko ove korisnike označiti, kao „slične“. Naredni korak jeste da, ako postoji neki proizvod koji je jedan korisnik probao (ocijenio, kupio), a da drugi korisnik nije. Ovdje ovaj sistem dolazi do izražaja tako što se „pretpostavlja“ da će se ovaj proizvod svidjeti i drugom korisniku, te mu se zbog toga i preporučuje. Također, kao i u prethodnom tipu, ovo je samo jedan banalan primjer, s dva korisnika i nekoliko proizvoda, ali se u realnosti radi o velikom broju različitih korisnika i proizvoda. Ovaj tip preporuke se još naziva i „korisnik-korisnik“ (user-user) kolaborativno filtriranje, jer se zasniva na sličnostima samih korisnika.

Također, postoji i drugi tip koji se naziva „stavka-stavka“ (item-item) koji se odnosi na sličnost između dvije stavke (šta god da one predstavljaju). Analogno prvom primjeru, drugi tip određuje sličnosti između stavki, tako da se neka stavka preporučuje korisniku koji je već ranije pokazao interes za slične stavke. Postoje razne metode i metrike, za mjerenje „sličnosti“ između korisnika (ili stavki), koje se koriste za predikciju rezultata. Najčešći pristup je kreiranje matrica korisnik-stavka-korisnik ili stavka-korisnik-stavka. Preporuke nastale ovim pristupom je potrebno još dodatno filtrirati, dodavanjem nekih težinskih vrijednosti, kao što su srednje ocjene, kako bi se izvršila normalizacija rezultata. Primjene ovih algoritama će biti prikazane u nastavku rada.



Slika 2 – Primjer kolaborativnog filtriranja [5]

2.3. Ostali tipovi sistema preporučivanja

Postoje još neki tipovi na kojima se temelje sistemi preporučivanja, koji će ovdje biti samo ukratko predstavljeni. Poseban tip je hibridno filtriranje, koje kobinuje koncepte prethodno objašnjenih tipova i još nekih dodatnih tipova. Pošto svaki od ovih tipova, sadrži određeni broj problema, ova metoda kombinira sve prednosti i uzima pojedinačne segmente ovih tipova, kako bi se kreirao napredniji sistem, koji prevazilazi njihove nedostatke. Na ovaj način se postižu bolje performanse i dosta bolji i precizniji rezultati. Naprimjer, ako kolaborativnom filtriranju nedostaju informacije o zavisnostima korisnika (ili stavki), dok filtriranju zasnovanom na sadržaju nedostaju informacije o preferencijama korisnika, hibridni sistem će iskoristiti poznate podatke iz oba tipa kako bi generisao preporuke. [6]

Kontekstno filtriranje je tip koji uključuje kontekstualne informacije korisnika u procesu preporuke. Ovdje se koristi niz kontekstualnih radnji korisnika, zajedno s trenutnim kontekstom, kako bi se predvidjela vjerovatnoća sljedeće akcije. Naprimjer, na osnovu nekih parametara kao što su zemlja, uređaj, vrijeme, datum prethodnih radnji (akcija), model može predvidjeti korisnikove naredne radnje (akcije). Isto tako jedan od tipova je i sistem zasnovan na znanju, kod kojeg se preporuke zasnivaju na uticaju na potrebe korisnika. To mogu biti neki kriteriji koji definiraju, kada bi neki proizvod (usluga) mogla biti od koristi za korisnika. Tehnike dubokog učenja su napredne tehnike koje koriste umjetne neuronske mreže za proučavanje svih vrsta podataka povezanih s određenim domenima vezanim za tematiku samog sistema. Postoje različite vrste neuronskih mreža koje se koriste za ovu namjenu (RNN, CNN, DNN). [7]

3 PRAKTIČNI DIO

Praktični dio se odnosi na implementaciji sistema preporuke za filmove. Pošto su domen ovog sistema filmovi, koristit će se već gotovi (prikupljeni) skup podataka, koji će u nastavku biti detaljnije objašnjen. Također, iskoristit će se prethodno navedeni tipovi, zajedno s pratećim algoritmima za preporuku. Iako se radi o sistemu preporuke za filmove, na sličan način bi se mogao implementirati i neki drugi sistem preporuke, naravno s adekvatnim podacima, kao što su muzika, knjige, online trgovina i drugi.

3.1 Alati, biblioteke i radno okruženje

Implementacija je urađena koristeći Python programski jezik (verzija 3.10.0) [8]. Python je najbolji izbor za ovakve projekte, jer ima jako velik ekosistem biblioteka, mogućnosti vizualizacije, fleksibilnost, podršku zajednice, te nezavisnost platforme. Iz tog razloga je idealan kandidat za potrebe mašinskog učenja i analize podataka. Korištene su napredne Python-ove biblioteke koje pružaju efikasnost i jednostavnost u svojim primjenama, što karakteriše čitljivost i lahka sinatksu. U nastavku je lista korištenih biblioteka:

- numpy
- pandas
- mathplotlib
- seaborn
- scikit-learn
- tensorflow/keras
- surprise

Za pisanje samog koda korišten je Visual Studio Code [9], korištenjem posebnog formata (ekstenzije) prilagođenog Python-u koji se naziva Jupyter Notebook (.ipynb). Jupyter Notebook pruža podršku za kreiranje i dijeljenje dokumenta koji sadrži „živi“ (izvršeni) kod, koji je popraćen vizualizacijom i opisnim dijelom teksta.

Kada je u pitanju sama implementacija prvo je odrađen kratak osvrt na skup podataka, te njihova kratka analiza i obrada za daljnju upotrebu. Prva metoda (algoritam) jeste kreiranje i testiranje modela neuronske mreže, a zatim primjena filtriranja baziranog na sadržaju korištenjem TF-IDF algoritma. Na kraju je prikazano poređenje dva algoritma KNN i SVD na primjeru korištenja kolaborativnog filtriranja. Svi podaci, modeli i metode su organizovani u zasebnim folderima i fajlovima, te sadrže kratka uputstva i objašnjenja samog koda.

3.2 Skup podataka (dataset)

Skup podataka koji je korišten za implementaciju praktičnog dijela je MovieLens Dataset. [10] Iza ovog skupa podataka stoji organizacija GroupLens Research koji su prikupili i stavili na raspolaganje podatke o recenzijama (ocjenama, rejtingu) s web stranice MovieLens (https://movielens.org). Podaci su prikupljeni u različitim vremenskim periodima, te postoje skupovi različitih veličina (broja podataka).

Podaci su preuzeti s web stranice Kaggle (https://www.kaggle.com). Kaggle je stranica koja omogućava korisnicima da pronađu skupove podataka za korištenje u izgradnji modela vještačke inteligencije. Također, moguće je i objaviti svoje skupove podataka, raditi s drugim naučnicima i inženjerima, kao i učestvovati u raznim takmičenjima i izazovima koji se odnose na rješavanje problema iz sfere „nauke o podacima“ (data science).

Sam MoviLens skup podataka sadržava ocjene korisnika na određene filmove (još dodatno i dodjeljene oznake korisnika na filmove, ali oni neće biti razmatrani u ovom radu). Period u kojem su kreirani podaci je između 1995-2015. godine, a skup podataka je generisan 2016. godine. Korisnici su odabrani nasumično, te njihovi podaci ne uključuju nikakve demografske podatke, već samo jedinstveni identifiaktor (ID). U nastavku će detaljnije biti proučen i prikazan skup podataka.

3.3 Priprema i analiza podataka

Preuzeti podaci su raspoređeni unutar zasebnih .csv datoteka, a glavni skup podataka se sastoji od 6 datoteka, ali će se u trenutnom primjeru koristiti samo podaci iz dvije datoteke i to: movie.csv i rating.csv. Podaci o filmovima unutar movie.csv sastoje se od: movieId (jedinstvenog identifikatora filma), title (puni naziv filma), genres (lista pripadajućih žanrova filma). Podaci o ocjenama (rejtingu) unutar rating.csv sadrže četiri kolone: userId (jedinstveni identifikator korisnika), movieId (jedinstveni identifikator filma), rating (ocjenu korisnika za dati film u rasponu od 0.5 – 5), timestamp (datum i vrijeme kreiranja/davanja ocjene). Za generisanje sistema preporuke prvi korak jeste istraživanje podataka, kako bi se utvrdi odnos između podataka, ali kako bi se i razumjela svrha i vrijednost svakog pojedinačnog atributa. U ovom primjeru radi se o vrlo malom broju atributa, tako da je vrlo intuitivno zaključiti povezanost između ovih podataka. Radi se o tome da su ove dvije datoteke jednostavno povezane putem jedinstvenog identifikatora filma movieId (koji se pojavljuje u obje datoteke). Kada se radi o samoj količini (broju) podataka unutar datoteka radi se o:

- 27278 različitih filmova
- 20000264 različitih recenzija

U nastavku analize može se utvrditi da je ukupno 138494 različitih korisnika ostavilo ocjenu (minimalno jednu). Prije generisanja sistema preporuke važno je pripremiti same podatke, kako bi rezulati preporuke bili što tačniji. Ovo se odnosi na uklanjanje ili čišćenje suvišnih ili nepotpunih podataka. Ove operacije se mogu odnositi na:

- uklanjane duplikata (ako postoje)
- uklanjanje N/A vrijednosti (ako postoje)
- izostavljanje filmova ili korisnika koji nisu „aktivni“ (korisnici koji su ostavili mali broj recenzija ili filmovi koji su dobili mali broj recenzija)

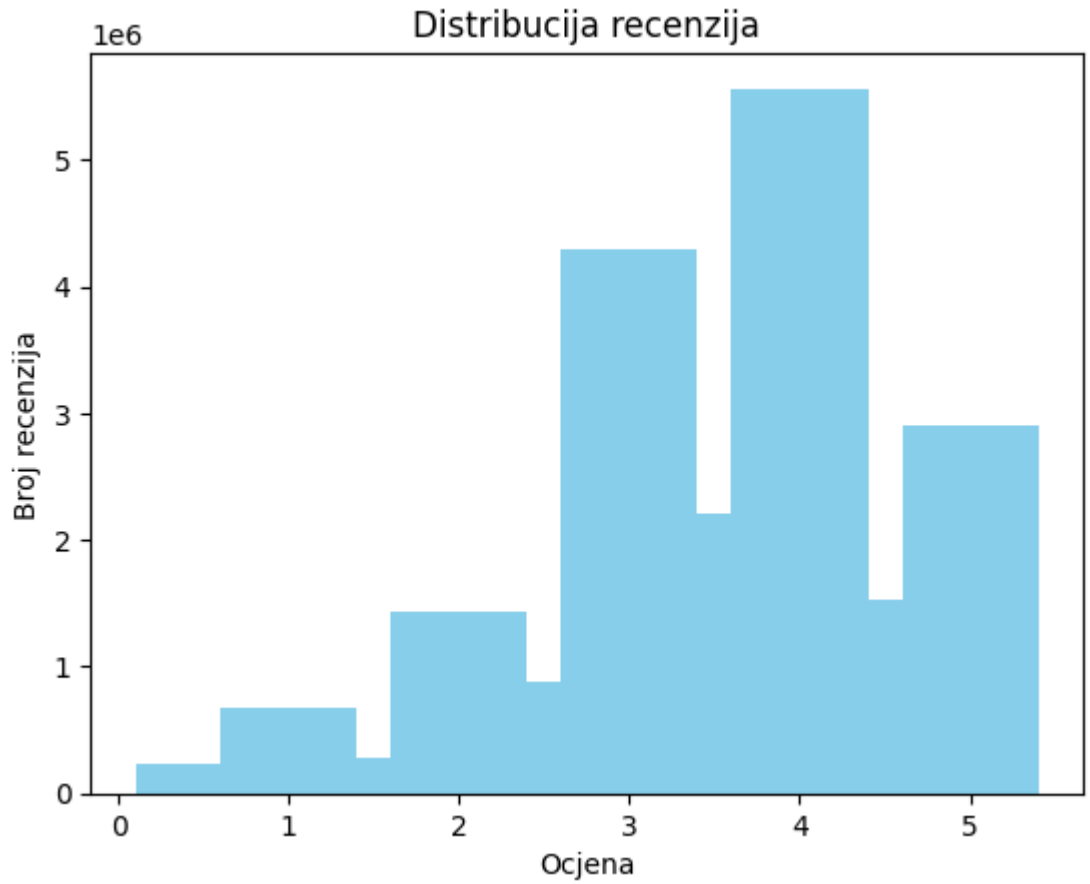
```
   movieId                                title  \
0        1                    Toy Story (1995)
1        2                      Jumanji (1995)
2        3             Grumpier Old Men (1995)
3        4            Waiting to Exhale (1995)
4        5  Father of the Bride Part II (1995)

                                        genres
0  Adventure|Animation|Children|Comedy|Fantasy
1                   Adventure|Children|Fantasy
2                              Comedy|Romance
3                        Comedy|Drama|Romance
4                                      Comedy
------------------------------------------------------
   userId  movieId             tag            timestamp
0      18     4141    Mark Waters  2009-04-24 18:19:40
1      65      208      dark hero  2013-05-10 01:41:18
2      65      353      dark hero  2013-05-10 01:41:19
3      65      521  noir thriller  2013-05-10 01:39:43
4      65      592      dark hero  2013-05-10 01:41:18
------------------------------------------------------
   userId  movieId  rating            timestamp
0       1        2     3.5  2005-04-02 23:53:47
1       1       29     3.5  2005-04-02 23:31:16
2       1       32     3.5  2005-04-02 23:33:39
3       1       47     3.5  2005-04-02 23:32:07
4       1       50     3.5  2005-04-02 23:29:40
```

Slika 3 – Pregled skupa podataka

Naredni korak jeste vizualizacija dostupnih podataka, kako bi se lakše mogli uvidjeti određeni trendovi i poveznice. Naprimjer, za potpuno razumijevanje je korisno saznati neke grupisane podatke koji se ne mogu vrlo lahko saznati samim pregledom „sirovih“ podataka. Neki od zanimljivih grafičkih prikaza mogu uključivati:

- distribuciju recenzija (pregled broja recenzija po kategoriji)
- računanje srednje i medialne vrijednosti recenzija
- prosjek recenzija po korisniku i filmu
- prosjek količine recenzija po korisniku i filmu
- distribucija žanrova filmova
- stanje recenzija po godinama



Slika 4 – Distribucija recenzija

4 IMPLEMENTACIJA

Nakon pripreme i analize podataka može se započeti s kreiranjem sistema preporuke za filmove. Pošto je skup podataka istražen, filtriran i spreman za primjenu, može se krenuti s implementacijom različitih algoritama. Kroz ove algoritme bit će prikazana njihova tačnost i preciznnost i performanse, koje se odnose na predviđanje ocjena koje bi korisnik ostavio na filmove koje nije gledao (ocijenio), te generisanje preporuka na osnovu filmova koje je korisnik gledao (ocijenio), odnosno na osnovu sličnosti između korisnika s drugim korisnicima i filmova s drugim filmovima.

4.1 Model neuronske mreže

Za kreiranje modela neuronske mreže bilo je neophodno još dodatno pripremiti podatke. Zbog velikog broja podataka (preko 20 miliona recenzija), proces treniranja neuronske mreže je spor, odnosno performanse znatno zavise od samih specifikacija hardvera. Iz tog razloga ocjene su dodatno filtrirane na način da se „odbace“ manje ocjene, tako da ostaju samo ocjene 4 i 5. Ovim pristupom ostalo je 12 miliona recenzija. Na ovaj način se može simulirati binarna klasifiakcija, tako da se ocjenom 5 smatra film koji bi se korisniku svidio (odnosno kojeg bi on kao takvog ocijenio), dok ocjena 4 predstavlja film koji mu se manje „sviđa“. U ovom kontekstu radi se o filmu koji mu se „ne sviđa“, jer su ostale ocjene samo 4 i 5, mogu se mapirati u vrijednosti 0 i 1. Dodatni korak jeste filtriranje korisnika, kako bi se broj recenzija još smanjio. Za ovo se koristi filtriranje kojim se izuzima 30% slučajno odabranih korisnika, a tako i njihovih pratećih recenzija. Na samom kraju konačno se došlo do „zadovoljavajućeg“ broja podataka za treniranje modela (oko 3.6 miliona recenzija). Prije kreiranja modela potrebno je još transformisati podatke koristeći LabelEncoder. Ovaj postupak je jako bitan, jer se podaci transformišu u normalizirane vrijednnosti, kako bi ih model lakše i bolje procesirao. Zatim se podaci dijele u dvije grupe: trening i test podaci (80% trening i 20% test podataka). Model neuronske mreže se kreira uz pomoć tensorflow i keras biblioteka, a sastoji se od sljedećih dijelova:

- ulazni slojevi - user_input i movie_input, koji predstavljaju jedinstvene identifikatore korisnika i filmova
- embedding slojevi – user_embedding i movie_embedding, koji se koriste za učenje reprezentacija korisnika i filmova, odnosno transformaciju identifikatora u gusto raspoređene vektore

- flatten slojevi – user_flatten i movie_flatten, koji služe kako bi se „izravnali“ rezultati iz embedding slojeva, što se postiže pretvaranjem matrica u vektore
- sloj konkatenacije – služi za spajanje rezultata iz flatten slojeva u jedan vektor
- dense slojevi – potpuno povezeni slojevi (128 neurona i „relu“ aktivacijska funkcija)
- izlazni sloj – sadrži 1 neuron (što predstavlja jedan izlaz) i ima linearnu aktivacijsku funkciju

Na kraju model je definisan prosljeđivanjem prethodno kreiranih ulaznih i izlaznih slojeve.

```
user_input = tf.keras.layers.Input(shape=(1,))
user_embedding = tf.keras.layers.Embedding(input_dim=len(user_encoder.classes_), output_dim=50, input_length=1)(user_input)
user_flatten = tf.keras.layers.Flatten()(user_embedding)

movie_input = tf.keras.layers.Input(shape=(1,))
movie_embedding = tf.keras.layers.Embedding(input_dim=len(movie_encoder.classes_), output_dim=50, input_length=1)(movie_input)
movie_flatten = tf.keras.layers.Flatten()(movie_embedding)

concatenated = tf.keras.layers.Concatenate()([user_flatten, movie_flatten])

dense_layer = tf.keras.layers.Dense(128, activation='relu')(concatenated)
output_layer = tf.keras.layers.Dense(1, activation='linear')(dense_layer)

model = tf.keras.Model(inputs=[user_input, movie_input], outputs=output_layer)
```

Slika 5 – Kreiranje modela neuronske mreže

Sljedeći korak je „kompajliranje“ modela koji koristi „adam“ optimizator, za gubitak koristi „mean squared error“, te kao metriku ima „mean average error“. Model se zatim trenira korištenjem prethodno podijeljenih trening podataka na ulazne (korisnik i film) i izlazni (rejting). Proces treniranja je podijeljen u 10 epoha, koje su otprilike trajale 3 i pol sata. Naravno, uz različite specifikacije hardvera moguće su varijacije u odnosu na vrijeme izvođenja procesa treniranja. Nakon treniranja, model kao takav se može spremiti (odnosno istrenirane težine) na sam disk. Kako bi se u budućnosti ove istrenirane težine mogle ponovo koristiti, te kako bi se model mogao unaprijeđivati. Nakon učitavanja spremljenog modela (ili nastavka rada na trenutnom modelu) vrše se predikcije na osnovu testnih podataka. Na osnovu ovih predikcija mogu se izračunati „srednja kvadrata greška“ i „preciznost“ modela, koji u ovom slučaju iznose 0.33 (zaokruženo na dvije decimale), odnosno 0.48 (zaokruženo na dvije decimale). Kako bi se model testirao na nekim „stvarnim“ podacima, odnosno koristeći podatke iz rating.csv datoteke koji nisu uključeni u proces treniranja modela, mogu se odabrati neki nasumični identifikatori za korisnike i filmove. Cilj ovoga je osnovu slučajno odabranog identifikatora korisnika, prvo odrediti filmove koje on nije ocijenio, a zatim na osnovu identifikatora tih filmova predvidjeti koje bi ocjene korisnik dao tim filmovima. U konkretnom

primjeru odabran je ID 11 za korisnika, te su uzeti slučajni ID-ovi njegovih neocjenjenih filmova (2178, 18, 973, 320, 5411 – njih 5 u ovom slučaju). Kada se modelu proslijede ovi podaci rezultat su predviđene ocjene za te filmove od strane odabranog korisnika. Pošto su na početku ostavljene samo ocjene 4 i 5, tako i rezultati variraju između 4 i 5 (u vidu decimalnih brojeva zaokruženih na nekoliko decimala), a radi jednostavnosti prikaza zaokružene su na približne vrijednosti.

```
print(f"Predviđene ocjene za korisnika {user_id_to_predict}:")
for movie_id, rating in sorted_predictions:
    movie_title = movies_data[movies_data['movieId'] == movie_id]['title'].values[0]
    print(f"Film ID {movie_id}: {movie_title}, Predviđena ocjena {int(rating)}")
```

```
Predviđene ocjene za korisnika 11:
Film ID 973: Meet John Doe (1941), Predviđena ocjena 5
Film ID 2178: Frenzy (1972), Predviđena ocjena 5
Film ID 5411: Summer Holiday (1963), Predviđena ocjena 4
Film ID 320: Suture (1993), Predviđena ocjena 4
Film ID 18: Four Rooms (1995), Predviđena ocjena 4
```

Slika 6 – Predviđanja ocjena modela neuronske mreže

Na osnovu ovih podataka sada je potrebno predvidjeti ocjene od svih neocjenjenih filmova za nekog korisnika, te onda jednostavno sortirati te rezultate od najvećeg do najmanjeg. Preporuka filmova korisniku se u ovom slučaju bazira na prikaz prvih N rezultata (u većini primjera koristi se prvih 10 rezultata). Na identičan način kao i u prethodnom slučaju odabire se ID korisnika, određuju se ID-ovi njegovih neocjenjenih filmova, te se nakon predikcije vrši sortiranje na osnovu ocjene, te se ispisuje prvih 10 rezultata.

```
top_n_movies = 0
print(f"Preporuke za korisnika ID {user_id_to_recommend}:")
for movie_id, rating in sorted_recommendations:  # Prikazujemo prvih 10 preporuka
    if movie_id in movies_data['movieId'].values:
        movie_title = movies_data[movies_data['movieId'] == movie_id]['title'].values[0]
        print(f"Naziv filma (ID {movie_id}): {movie_title}")
        top_n_movies += 1
        if top_n_movies == 10:
            break
```

```
Preporuke za korisnika ID 12:
Naziv filma (ID 1154): T-Men (1947)
Naziv filma (ID 255): Jerky Boys, The (1995)
Naziv filma (ID 3659): Quatermass 2 (Enemy from Space) (1957)
Naziv filma (ID 2210): Sabotage (1936)
Naziv filma (ID 6256): House with Laughing Windows, The (Casa dalle finestre che ridono, La) (1976)
Naziv filma (ID 1356): Star Trek: First Contact (1996)
Naziv filma (ID 313): Swan Princess, The (1994)
Naziv filma (ID 1093): Doors, The (1991)
Naziv filma (ID 6803): Phenomena (a.k.a. Creepers) (1985)
Naziv filma (ID 2637): Mummy's Hand, The (1940)
```

Slika 7 – Prikaz preporuka modela neuronske mreže

4.2 TF-IDF algoritam

Za implementaciju filtriranja zasnovanog na sadržaju korišten je TF-IDF (Term Frequency – Inverse Document Frequency) algoritam. TF (učestalost termina) predstavlja broj pojavljivanja riječi u dokumentu podijeljen s ukupnim brojem riječi u dokumentu. Svaki dokument ima svoju frekvenciju termina. IDF (inverzna frekvencija podataka) je dnevnik broja dokumenata podijeljen s brojem dokumenata koji sadrže tu riječ. Inverzna frekvencija podataka određuje težinu rijetkih riječi u svim dokumentima u korpusu. TF-IDF algoritam se koristi u obradi prirodnog jezika (natural language processing) i služi za pronalaženje informacija, kako bi se odredila važnost termina u dokumentu. Riječi koje se pojavljuju često u dokumentu imat će nisku težinsku vrijednost, dok će riječi koje su rijetke imati višu rijetkost. [11]

Prvi korak kod korištenja ovog algoritma jeste vektorizacija teksta, odnosno transformacije teksta u numeričke vektore koji se mogu obraditi algoritmima mašinskog učenja. Pristup koji se ovdje koristi jeste uklanjanje „stop“ riječi i ostalih nepotrebnih pojmova (simbola). Stop riječi predstavljaju riječi koje se često pojavljuju, ali ne nose sa sobom nikakvu značajnu informaciju, tako da mogu znatno utjecati na rezultate procesiranja. Ove riječi su zapravo riječi iz standarnog jezika, kao što su veznici i prijedlozi (primjeri na engleskom jeziku: a/an, the, and, in, at, on). Nakon toga riječi se tokeniziraju u listu pojmova, pa se zatim kreira „rječnik“ svih jedinstvenih pojmova, gdje se svakom terminu dodijeljuje težinska vrijednost na osnovu broja pojavljivanja u dokumentu. Biblioteka scikit-learn u Pythonu omogućava transformator pod nazivom „TfidfVectorizer“ za računanje težinskih vrijednosti. [12]

Drugi korak je kreiranje matrice na osnovu kosinusne sličnosti. Kosinusna sličnost je mjera sličnosti između dva vektora različita od nule u visokodimenzionalnom prostoru. Računanje kosinusne sličnosti se svodi na računanje dot proizvoda dva vektora, koji je zbir proizvoda njihovih odgovarajućih elemenata. Zatim se dot proizvod dijeli s proizvodom njihovih veličina kako bi se dobio kosinus ugla između dva vektora. Rezultat je mjera sličnosti dva vektora, pri čemu vrijednost 1 označava da su vektori identični, a vrijednost 0 označava da su vektori potpuno različiti. U konkretnom primjeru poređenje kosinusne sličnosti između više dokumenata, identificiraju se oni koji su međusobno najsličniji i grupišu se u skladu s tim. [13] Jednostavan primjer u implementaciji gore navedenog jeste TF-IDF vektorizacija teksta koji je sastavljen od naziva i žanrova filma. Nakon vektorizacije teksta i računanja kosinusne sličnosti metodi za određivanje preporuka se proslijeđuje naziv filma, na osnovu kojeg se pronalaze slični filmovi, te se ispisuju rezultati 10 najsličnijih filmova. Iz rezultata se može zaključiti da pojavljivanje žanrova određuje rezultate pretrage, a ovaj tip preporuke može biti koristan za filmove koji imaju isti naziv (nastavci filmova po brojevima i godinama).

```
# Primjer: Generisanje preporuka za film "Contact"
movie_title = "Contact"
print(movies[movies.title.eq(movie_title)])
```

```
      movieId    title        genres              combined
1532     1584  Contact  Drama|Sci-Fi  Contact Drama|Sci-Fi
```

```
recommendations = get_recommendations(movie_title)

print(f"Preporuke za film '{movie_title}':")
for i, (title, genres) in enumerate(recommendations):
    print(f"{i + 1}. {title} ({genres})")
```

```
Preporuke za film 'Contact':
1. V (Drama|Sci-Fi)
2. It's Me, It's Me (Comedy|Drama|Sci-Fi)
3. Love (Drama|Sci-Fi)
4. Last Night (Drama|Sci-Fi)
5. Day After, The (Drama|Sci-Fi)
6. Everything I Can See From Here (Adventure|Animation|Drama|Sci-Fi)
7. Day, The (Drama|Sci-Fi|Thriller)
8. Making Contact (Fantasy|Horror|Sci-Fi)
9. Beyond the Stars (Drama|Sci-Fi)
10. Face of Another, The (Drama|Sci-Fi)
```

Slika 8 – Prikaz preporuka na osnovu naziva filma

Napredniji primjer korištenja ovog algoritma je preporuka filmova određenom korisniku (što je i cilj ovog rada). Modifikacija prethodnog primjera odnosi se na to što se odabire nasumičan identifikator korisnika (userId), koji se proslijeđuje metodi za određivanje preporuke. Početni dio koda, koji se odnosi na vektorizaciju teksta i računanje kosinusne sličnosti ostaje isti. Naredni korak je pronalaženje filmova koje je korisnik ocijenio, te dodatno filtriranje tako da ostanu samo filmovi koji su visoko ocijenjeni (ocjena veća od 3), koji će se koristiti za pronalaženje sličnih filmova. Poslije toga dodatno se računa i prosječna ocjena ocjenjenih filmova po žanru. Ovo je bitan korak, jer se konačna lista filmova za preporuku dobija množenjem sličnosti filma s prosječnom ocjenom žanra, čime se dobija težinska sličnost vrijednosti. Na osnovu analize rezultata za nasumično odabranog korisnika, utvrđeno je da 3 od 4 visoko ocijenjena filma pripadaju žanru „Drama“, pa prvih 10 preporuka pripada upravo tom žanru (većina filmova ima više od jednog žanra).

```python
# Generiranje preporuka za određenog korisnika
recommended_movies = get_movie_recommendations(user_id)

print(f"Preporuke za korisnika sa ID {user_id}:")
for i, (title, genres) in enumerate(recommended_movies):
    print(f"{i + 1}. {title} ({genres})")
```

```
Visoko ocijenjeni filmovi korisnika ID 1:
['Cinema Paradiso', 'Dracula', 'French Connection, The', 'Tron']
Preporuke za korisnika sa ID 1:
1. Othello (Drama)
2. Dangerous Minds (Drama)
3. Cry, the Beloved Country (Drama)
4. Restoration (Drama)
5. Georgia (Drama)
6. Home for the Holidays (Drama)
7. Mr. Holland's Opus (Drama)
8. Two Bits (Drama)
9. Journey of August King, The (Drama)
10. Margaret's Museum (Drama)
```

Slika 9 – Prikaz preporuka za nasumično odabranog korisnika

4.3 KNN i SVD algoritmi

Za implementaciju kolaborativnog filtriranja korištena su dva algoritma: KNN (K – Nearest Neighbour) i SVD (Singular Value Decomposition). KNN algoritam, kako i sam naziv kaže odnosi se na pronalazak K „najbližih susjeda“, u konkretnom slučaju to mogu biti ili korisnici ili filmovi (pošto su korištena dva već objašnjena pristupa „user-user“ i „item-item“). S druge strane SVD algoritam koristi tehniku faktorizacije matrice koja razlaže matricu u proizvod matrica niže dimenzionalnosti, a zatim se izdvajaju karakteristike od najveće do najmanje važnosti. [14] KNN predstavlja n filmova u dimenzionalnom prostoru definiranom od n korisnika (ili obrnuto u zavisnosti od korištenog pristupa). Udaljenost između pojedinačnih tačaka može se izračunati na više načina, kao što je euklidska udaljenost, ali u ovom slučaju koristit će se već objašnjeni pristup primjene kosinusne sličnosti, jer daje bolje rezultate po pitanju performansi na skupovima podataka visoke dimenzionalnosti. [15] Dakle, KNN iterira kroz pojedinačne elemente, dok SVD promatra matricu kao cjelinu. Dekompozicijom korisnik-film matrice, dobijaju se dvije matrice korisnik i film. Nakon toga, na osnovu nepraznih ćelija u matrici korisnik-film predviđaju se vrijednosti ćelija koje nemaju vrijednosti.

Ova dva algoritma se mogu jednostavno implementirati korištenjem surprise biblioteke u Pythonu, koja pruža mogućnosti kao što je konvertovanje skupa podataka u validan format. Poslije učitavanja podataka, podaci se dijele u dvije grupe: trening i test (80% trening i 20% test). Modeli oba algoritma se jednostavno inicijaliziraju te se vrši njihovo treniranje s trening podacima. Primjena će biti prikazana samo na primjeru „user-user“ pristupa. Dakle, to znači da će se koristiti sličnost između korisnika.

U metodi za određivanje preporuka izvodi se nekoliko koraka. Pošto se metodi prosljeđuje jedinstveni identifikator korisnika (userId), prvo se određuju filmovi koje je korisnik nije ocijenio. Drugi korak je pronalaženje sličnih korisnika prosljeđenom korisniku. Treći korak je predikcija ocjena koje bi korisnik dao neocjenjenim filmovima, na osnovu ocjena koje su tom filmu već dali njemu slični korisnici. Na kraju se te predikcije sortiraju u opadajućem redoslijedu, te se ispisuje prvih N preporuka. Za nasumično odabranog korisnika, ova dva algoritma su dala različite rezultate za prvih 10 preporuka. Naravno, ovo je bilo i očekivano, obzirom da se koriste drukčiji pristupi. Naknadnom analizom utvrđeno je da KNN daje više „generičnije“ predikcije po pitanju ocjena. Poseban dio koji se odnosi na same metrike i efikasnosti ovih algoritama će biti prikazan u nastavku.

```
display_recommendations(recommendations_data_svd)
```

```
ID: 6016, Naslov: City of God (Cidade de Deus) (2002) (Action|Adventure|Crime|Drama|Thriller)
ID: 1221, Naslov: Godfather: Part II, The (1974) (Crime|Drama)
ID: 2542, Naslov: Lock, Stock & Two Smoking Barrels (1998) (Comedy|Crime|Thriller)
ID: 318, Naslov: Shawshank Redemption, The (1994) (Crime|Drama)
ID: 858, Naslov: Godfather, The (1972) (Crime|Drama)
ID: 1212, Naslov: Third Man, The (1949) (Film-Noir|Mystery|Thriller)
ID: 50, Naslov: Usual Suspects, The (1995) (Crime|Mystery|Thriller)
ID: 1213, Naslov: Goodfellas (1990) (Crime|Drama)
ID: 2329, Naslov: American History X (1998) (Crime|Drama)
ID: 922, Naslov: Sunset Blvd. (a.k.a. Sunset Boulevard) (1950) (Drama|Film-Noir|Romance)
```

```
display_recommendations(recommendations_data_knn)
```

```
ID: 6, Naslov: Heat (1995) (Action|Crime|Thriller)
ID: 68, Naslov: French Twist (Gazon maudit) (1995) (Comedy|Romance)
ID: 108, Naslov: Catwalk (1996) (Documentary)
ID: 136, Naslov: From the Journals of Jean Seberg (1995) (Documentary)
ID: 186, Naslov: Nine Months (1995) (Comedy|Romance)
ID: 203, Naslov: To Wong Foo, Thanks for Everything! Julie Newmar (1995) (Comedy)
ID: 217, Naslov: Babysitter, The (1995) (Drama|Thriller)
ID: 263, Naslov: Ladybird Ladybird (1994) (Drama)
ID: 274, Naslov: Man of the House (1995) (Comedy)
ID: 279, Naslov: My Family (1995) (Drama)
```

Slika 10 – Prikaz preporuka KNN i SVD algoritama

4.4 Poređenje rezultata i pregled performansi

U prethodnom dijelu izostavljen je dio koji se odnosi na evaluaciju modela KNN i SVD algoritama. Metrike koje su korištene za procjenu performansi algoritama su MAE (Mean Absolute Error), MSE (Mean Squared Error), RMSE (Root Mean Squared Error)). MAE ili srednja apsolutna greška predstavlja prosjek apsolutne razlike između stvarnih i predviđenih vrijednosti u skupu podataka, odnosno mjeri prosjek ostataka u skupu podataka. MSE ili srednja kvadratna greška predstavlja prosjek kvadratne razlike između originalne i predviđene vrijednosti u skupu podataka, odnosno mjeri varijansu reziduala. RMSE ili korijenska srednja kvadratna greška je kvadratni korijen srednje kvadratne greške, odnosno mjeri standardnu devijaciju ostataka. Pošto tehnika kolaborativnog filtriranja predstavlja regresijski nadzirani model, izlaz je numerička vrijednost (ocjena). [16]

Jedan od pristupa za računanje ovih metrika je korištenje ugrađene funkcije „cross_validate“ unutar surprise biblioteke gdje se prosljeđuju algoritmi i skup podataka. Kako dodatni parametar prosljeđuje se i broj „unakrsnih preklopa validacije“ (u ovom slučaju njih 5). Na osnovu rezultata može se zaključiti da SVD ima manje vrijednosti za RMSE i MAE na testnom skupu podataka od KNN, a također je potrebno i manje vremena za izračunavanje.

```
cross_validate_KNN = cross_validate(algo_KNN, rating_df, measures=['RMSE', 'MAE'], cv=5, verbose=True)
```

```
Computing the cosine similarity matrix...
Done computing similarity matrix.
Computing the cosine similarity matrix...
Done computing similarity matrix.
Computing the cosine similarity matrix...
Done computing similarity matrix.
Computing the cosine similarity matrix...
Done computing similarity matrix.
Computing the cosine similarity matrix...
Done computing similarity matrix.
Evaluating RMSE, MAE of algorithm KNNWithMeans on 5 split(s).

                  Fold 1  Fold 2  Fold 3  Fold 4  Fold 5  Mean    Std
RMSE (testset)    0.9318  0.9302  0.9201  0.9243  0.9281  0.9269  0.0042
MAE (testset)     0.7146  0.7127  0.7052  0.7068  0.7111  0.7101  0.0035
Fit time          5.32    5.25    6.10    5.60    5.64    5.58    0.30
Test time         4.22    4.23    4.59    4.47    4.34    4.37    0.14
```

```
cross_validate_SVD = cross_validate(algo_SVD, rating_df, measures=['RMSE', 'MAE'], cv=5, verbose=True)
```

```
Evaluating RMSE, MAE of algorithm SVD on 5 split(s).

                  Fold 1  Fold 2  Fold 3  Fold 4  Fold 5  Mean    Std
RMSE (testset)    0.8962  0.9143  0.8875  0.8872  0.9019  0.8974  0.0101
MAE (testset)     0.6903  0.7029  0.6854  0.6833  0.6948  0.6913  0.0070
Fit time          0.89    0.89    0.96    0.85    0.83    0.88    0.05
Test time         0.13    0.18    0.12    0.10    0.09    0.13    0.03
```

Slika 11 – Poređenje rezultata unakrsne validacije

Drugi način računanja je vrlo sličan, ali se za razliku od unakrsne validacije model obučava samo jednom, nakon čega se testira. U ovom primjeru se ispisuju RMSE, MAE, MSE, kao i testni skup podataka s podacima uid (userId), iid (movieId), r_ui (rating) i est (predviđena ocjena). Kao i u prethodnom slučaju na osnovu rezultata uočava se da SVD pruža manje greške, a samim tim i bolje performanse od KNN algoritma.

```
train_test_KNN = train_test_algo(algo_KNN, "algo_KNN")
print(train_test_KNN.head())
```

```
Computing the cosine similarity matrix...
Done computing similarity matrix.
RMSE - algo_KNN 0.9306864728287642
MAE - algo_KNN 0.7122355378981291
MSE - algo_KNN 0.866177310706446
   uid    iid  r_ui       est                                        details
0  547  39307   2.5  3.829056  {'actual_k': 40, 'was_impossible': False}
1  452   3809   3.0  2.725402  {'actual_k': 40, 'was_impossible': False}
2  102   1036   4.0  4.199759  {'actual_k': 40, 'was_impossible': False}
3  184    364   5.0  3.871193  {'actual_k': 38, 'was_impossible': False}
4  472   7070   3.0  4.645791  {'actual_k': 40, 'was_impossible': False}
```

```
train_test_SVD = train_test_algo(algo_SVD, "algo_SVD")
print(train_test_SVD.head())
```

```
RMSE - algo_SVD 0.8912553725652782
MAE - algo_SVD 0.6878460760914464
MSE - algo_SVD 0.7943361391264729
   uid   iid  r_ui       est                    details
0   30  2611   5.0  3.455180  {'was_impossible': False}
1  268  1517   3.5  3.761342  {'was_impossible': False}
2  640  1476   3.0  3.773878  {'was_impossible': False}
3  547  1282   5.0  3.720801  {'was_impossible': False}
4   56  3079   4.0  3.715933  {'was_impossible': False}
```

Slika 12 – Poređenje rezultata izlaznog test skupa podataka

5 ZAKLJUČAK

U današnje vrijeme kada postoji mnogo opcija za odabir nekog proizvoda ili usluge, sistemi preporuke se koriste u personalizaciji, tako što korisniku na vrlo brz i efikasan način pomažu da pronađe nešto što mu se sviđa. Personalizacijom preporuka postiže poboljšano korisničko iskustvo, a za samu platformu rezultira povećanjem prihoda. Postoji nekoliko tehnika koje se koriste za izgradnju jednog ovakvog sistema, koji se može primijenjivati u više različitih oblasti i domena. Dvije popularne tehnike su filtriranje zasnovano na sadržaju i kolaborativno filtriranje, ali najbolje rezultate daje hibridni sistem preporuke, koji kombinira najbolje karakteristike od obje tehnike. Kreiranje što boljeg i preciznijeg sistema zavisi od mnogo faktora, kao što su adekvatan skup podataka, te odabir najboljeg algoritma i modela. Analizom podataka i evaluacijom modela korištenjem raznih metrika, mogu se utvrditi važni parametri, koji za cilj imaju određivanje performansi i tačnosti generisanih preporuka.

6 LITERATURA

[1] https://www.techopedia.com/how-is-ai-used-in-recommendation-systems

[2] https://www.proxet.com/blog/what-is-a-recommender-system-with-ai-and-why-do-you-need-it

[3] https://www.nvidia.com/content/dam/en-zz/Solutions/glossary/data-science/recommendation-system/img-3.png

[4] https://medium.com/voice-tech-podcast/a-simple-way-to-explain-the-recommendation-engine-in-ai-d1a609f59d97

[5] https://www.nvidia.com/content/dam/en-zz/Solutions/glossary/data-science/recommendation-system/img-2.png

[6] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9269752/

[7] https://medium.com/sciforce/deep-learning-based-recommender-systems-b61a5ddd5456

[8] https://www.python.org/downloads/release/python-310/

[9] https://code.visualstudio.com/

[10] https://www.kaggle.com/datasets/grouplens/movielens-20m-dataset

[11] https://www.kaggle.com/code/ramzanzdemir/recommendation-systems-content-based-tf-idf

[12] https://www.ijraset.com/research-paper/movie-recommendation-system-using-tf-idf-vectorization-and-cosine-similarity

[13] https://www.geeksforgeeks.org/movie-recommender-based-on-plot-summary-using-tf-idf-vectorization-and-cosine-similarity/

[14] https://medium.com/@m_n_malaeb/singular-value-decomposition-svd-in-recommender-systems-for-non-math-statistics-programming-4a622de653e9

[15] https://www.freecodecamp.org/news/how-to-build-a-movie-recommendation-system-based-on-collaborative-filtering/

[16] https://medium.com/analytics-vidhya/mae-mse-rmse-coefficient-of-determination-adjusted-r-squared-which-metric-is-better-cd0326a5697e