

Sadržaj

Uvod	1
1. Opis projektnog zadatka	2
1.1. Povijest	2
1.2. Karakteristike sustava preporučivanja	3
1.3. Preporučivanje Spotify	5
1.4. Općenito o evaluaciji sustava preporučivanja	7
1.5. Ciljevi rada	9
2. Teorija.....	11
2.1. Filtriranje temeljeno na sadržaju	12
2.1.1. TF-IDF.....	13
2.2. Suradničko filtriranje.....	14
2.2.1. Algoritam temeljen na memoriji	15
2.2.1.1. k-Najbližih susjeda (KNN).....	16
2.3. Algoritam temeljen na modelu	17
2.3.1. Singularna dekompozicija matrice (SVD).....	18
2.4. Strojno učenje u sustavima preporučivanja	19
2.4.1 Logistička regresija	20
2.4.2 Stablo odlučivanja (DTC).....	21
2.4.3 Klasifikator K susjeda	22
2.4.4 Epsilon pohlepni algoritam.....	22
2.5. Načini evaluacije rezultata sustava preporučivanja.....	23
2.5.1. Računanje procjene pogreške u evaluaciji.....	25
2.5.2. Evaluacija regresije.....	27
2.5.3. Evaluacija klasifikacije.....	28

2.5.4	Evaluacija rangiranja	30
2.5.5	Evaluacijske mjere za performanse sustava preporučivanja	31
3.	Programska izvedba sustava preporučivanja.....	33
3.1.	Skup podataka za sustav preporučivanja	33
3.2.	Korištene tehnologije i alati.....	34
3.3.	Metodologija izrade preporuka.....	36
3.4.	Metodologija provedbe evaluacija.....	37
4.	Rezultati.....	40
4.1.	Klasifikacija kvalitete preporuka.....	40
4.2.	Opis rezultata prema izboru mjere evaluacije	42
4.2.1.	RMSE	43
4.2.2.	MAE	44
4.2.3.	NDCG.....	44
4.2.4.	Srednja prosječna preciznost (MAP).....	45
4.2.5.	Preciznost	45
4.2.6.	Odziv	46
4.2.7.	F1	47
4.2.8.	Evaluacijske mjere za performanse sustava preporučivanja	47
4.3.	Opis rezultata po metodi preporuke.....	48
4.3.1.	TF-IDF.....	48
4.3.2.	SVD	49
4.3.3.	KNN	49
4.3.4.	Logistička regresija.....	50
4.3.5.	Stablo odlučivanja	50

4.3.6.	Klasifikator K susjeda	51
4.4.	Najbolja metoda preporuke	51
5.	Zaključak	53
6.	Literatura	54

Uvod

U današnje vrijeme sve se više aktivnosti iz stvarnog svijeta premješta u virtualni te s time dolaze razne novosti u svijet međuljudskih interakcija. Tradicionalno smo se u odlukama i preporukama uvijek oslanjali na prijatelje, obitelj, poznanike i kolege te bi po njihovim savjetima najčešće temeljili naše odluke. Ovo se sve preokrenulo pojavom interneta, jer su se počele stvarati poveznice između različitih ljudi koji bi bez tehnologije ostali nepovezani. S novim poveznicama između ljudi dolaze i različite mogućnosti koje te poveznice stvaraju pa su tako i nastali sustavi preporučivanja. Sustavi preporučivanja uzimaju podatke različitih korisnika i potom pronalaze sličnosti između korisnika ili stvari koje je korisnik koristio. Na ovaj način funkcioniraju razne aplikacije u optimizaciji svog poslovanja. Neke od tih aplikacija su: društvene mreže, internetske trgovine, novinski portali te razne druge stranice. Štoviše, danas bi veliki udio poslova bez sustava preporučivanja bio naprosto nemoguć te je zato tim više važno razumijeti na koji način oblikuju svijet oko nas.

Baš je zbog te važnosti koju sustavi preporučivanja imaju u današnjem svijetu napravljen ovaj rad. U praktičnom je dijelu rada prvo kreiran skup podataka (*engl. dataset*) sa 150 popisa za reproduciranje koji ujedno predstavljaju 150 korisnika, gdje je svaki korisnik odslušao od 15 do 200 pjesama. Nakon toga se na temelju tog skupa podataka preporučuju pjesme pa se potom te iste preporuke evaluiraju. Na kraju su rada rezultati evaluacija prezentirani, objašnjeni i uspoređeni. Svrha rada je ponajviše usmjerena za istraživanje i dijagnostiku.

Rad je kao cjelina podijeljen u 7 poglavlja. U prvom je poglavlju napisan uvod u radu, a u drugom je drugom poglavlju opisan projektni zadatak, činjenice koje ga pobliže opisuju i ciljevi rada. Nakon toga je u trećem poglavlju opisana teorija metodologije i tehnika koje su se koristile za izradu rada. Ovdje su opisani korišteni koncepti filtriranja po sadržaju, suradničkog filtriranja, modela strojnog učenja i načina evaluacije sustava preporučivanja. Potom se u četvrtom poglavlju opisuje skup podataka, korišteni programi i metodologija za praktičnu izvedbu rada. U petom se poglavlju prikazuju dobiveni rezultati evaluacija, nakon čega se opisuju i međusobno uspoređuju. U šestom se poglavlju nalazi zaključak i u sedmom poglavlju je opisana literatura.

1. Opis projektnog zadatka

1.1 Povijest

Razvojem interneta su se pojavili pretraživački alati koji su s vremenom sve više rasli, zbog čega se pojavila potreba za poboljšavanjem istih. Neka od tih poboljšavanja su iznjedrila prve sustave preporučivanja početkom 90-ih godina prošlog stoljeća. 1992. godine se pojavljuje eksperimentalni sustav slanja poruka Tapestry koji je ujedno bio prvi sustav preporučivanja temeljen na suradničkom filtriranju. U Tapestryju se korisnicima omogućilo da napišu svoja pravila filtriranja za povezivanje s mišljenjima i ponašanjima drugih korisnika. Ova je tehnologija nadograđena 1994-e godine automatizacijom suradničkog filtriranja u GroupLens sustavu filtriranja vijesti.

Nastavkom desetljeća se pojavljuje sve više sustava preporučivanja, svaki napredniji od prethodnog. Razvoj sustava preporučivanja se u tom vremenu uglavnom odnosio na tehnike faktorizacije matrice (pogotovo u suradničkom filtriranju, odnosno selekciji sadržaja prilagođenoj uočenoj sličnosti između različitih korisnika). Među tim sustavima posebno je dojmljiv sustav kojeg je Amazon primijenio, koji mu je ujedno i pomogao da se plasira na vrhove na kojima se nalazi danas.

Sredinu 2000-ih godina je obilježilo sve češće primjenjivanje metoda strojnog učenja u sustavima preporučivanja. Ovaj razvoj je posebno ubrzalo Netflixovo natjecanje Netflix Prize (2006.,2009), kojim je Netflix htio optimizirati svoj sustav preporuke filmova. Strojno učenje je danas jedan od temelja u sustavima preporučivanja te se sve više razvija i proučava.

Protokom vremena su se pojavila neka nova mjerila kvalitete algoritma poput slučajnog otkrivanja (*engl. serendipity*) ili noviteta (*engl. novelty*). Osim toga su se počeli uzimat u obzir i neki drugi faktori za odlučivanje: društvena mreža korisnika, ocjene ili vrijeme. Danas se recimo umjesto matrice korisnik-stvar (*engl. user-item matrix*) koriste sekvencijalni logovi aktivnosti korisnika (*engl. sequence-aware recommendation*) [42].

1.2 Karakteristike sustava preporučenja

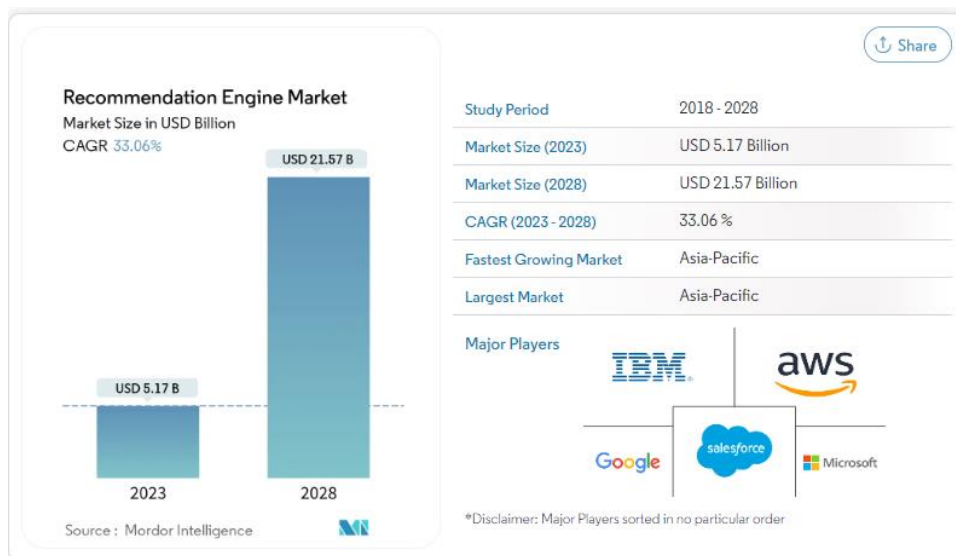
Sustavi preporučenja su algoritmi koji služe za preporučenje, izbacivanje nerelevantnih opcija te pronalazak stvari koje bi mogle zanimati korisnika. Koriste najčešće umjetnu inteligenciju (*engl. Artificial Intelligence - AI*) s velikom količinom podataka za preporučenje dodatnih proizvoda korisnicima. Korišteni podaci za preporuke mogu biti različitog porijekla: prethodne interakcije, prethodna povijest pretraživanja, demografske informacije itd. Mnogi pružatelji internetskih usluga koriste sustave preporučenja zbog njihove sposobnosti predviđanja interesa klijenta koji im služe u upućivanju klijenta prema bilo kakvom proizvodu ili usluzi koja bi ih mogla zanimati [44]. Brojne je činjenice potrebno uzeti u obzir prilikom praktične primjene, istraživanja ili razvoja sustava preporučenja. Neke od njih su: idiosinkracija sustava preporučenja, dobro razumijevanje tipova preporučenja koji generiraju vrijednost (za korisnika i razvijatelja), kako mjeriti utjecaj preporučitelja te postojanost rizika u generiranim preporukama [42].

Sustavi se preporučenja uglavnom dijele u dvije skupine: suradničko filtriranje i filtriranje po sadržaju. Preporuke u suradničkom filtriranju se baziraju na interesima korisnika koji su slični korisniku kojem se preporučuje. U ovoj se metodi rade preporuke izračunom težinske sume preferenca sličnih korisnika za korisnika kojem se preporučuje. Težine preferenci su proporcionalne korelacijama sa zajedničkim skupom stvari kojeg su oba korisnika ocijenili.

S druge strane u filtriranju po sadržaju sustav radi preporuke uspoređujući korisničke profile i informacije o proizvodu (npr. opisi i značajke, odnosno svojstva proizvoda). Ovaj sustav preporučenja izražava korisnikove preference koristeći matricu korisnik-stvar preko koje potom preporučuje proizvode u skladu s korelacijom između atributa proizvoda i korisnikovih preferenci [45]. Također postoji i hibridni način preporučenja koji kombinira različite sustave preporučenja za preporučenje. Obično se ovo radi u slučajevima gdje se istovremenim korištenjem različitih metoda optimizira sustav da bi se nadomjestila ograničenja i problemi koji su zateknuti prilikom preporučenja [46].

Sustavi preporučenja imaju razne upotrebe, što se ogleda i u veličini tržišta. Trenutno je veličina tržišta za sustave preporučenja 5.17B\$, dok je predviđeno da bi se ovaj broj 2028-e godine trebao popeti na 21.57B\$ što je vizualno istaknuto na slici 1 [48]. Za usporedbu, 2018-e

godine je veličina tržišta bila 1.14B\$, što znači da se veličina tržišta skoro pa upeterostručila od tada [47].



1. Predviđeni rast tržišta sustava preporučivanja od 2023. do 2028. [48]

Razlog tomu je stalno ustrajanje od strane tvrtki u poboljšanju svojih metoda preporučivanja ne bi li poboljšale korisničko iskustvo i sukladno financijski više profitirale. U poboljšavanju kvalitete sustava preporučivanja u zadnjih nekoliko godina najveću su ulogu imali razvoj internetskih trgovina, razvoj multimedijских platformi i integracija strojnog učenja u web platforme [48].

Jedan od poznatih sustava preporučivanja koristi Amazon. Amazon koristi suradničko filtriranje bazirano na matrici stvar-stvar na većini svojih web stranica i na promidžbi preko e-pošte. Navodno čak 35% dodatnih prodaja koje Amazon ostvari su dobivene zbog njegovog sustava preporučivanja. Osim Amazona, jako poznati sustav preporučivanja ima Netflix. Netflix je čak dvaput imao natjecanje na kojem su natjecatelji predstavljali svoje algoritme te bi za to dobili 1M\$ nagrade. Od drugih sustava bi bilo bitno napomenuti LinkedIn te Spotify, o kojem ćemo više u idućim poglavljima [49].

Neke od taktika koje internetske trgovine koriste za povećanje prodaja preko sustava preporučivanja su: uzimanje podataka iz škartiranih elektroničkih kolica, preporučivanje stvari s kojima je slični korisnik imao interakcije, preporučivanje po komentarima korisnika, komentarima sličnog korisnika i mnoštvo drugih načina.

Sustavi preporučivanja imaju razne prednosti i mane. Neke od prednosti su: povećanje retencije korisnika na određenoj web platformi, povećanje profita na platformama za 10-50%, formiranje navika korisnika i ubrzavanje posla [44]. S druge strane postoje i negativne strane sustava preporučivanja. Jedna od najznačajnijih negativnih strana je mogućnost gubitka autonomije i osobnog identiteta. Sustavi preporučivanja vrlo lako mogu utjecati na autonomiju korisnika preporučujući stvari koje bi usmjerile korisnika u određenom smjeru. Ovime se vrlo lako može izazvati ovisnost ili limitaciju njegovih opcija što može imati benigne ili čak zlonamjerne posljedice. Druga negativna strana bi bili socijalni efekti koje bi preporučivanje moglo imati. Ovo je posebno izraženo kod sustava preporučivanja na novinskim portalima i na društvenim mrežama gdje se vrlo lako može uskratiti korisnike da vide stavove suprotnih mišljenja. Neizlaganje suprotnom mišljenju narušava javnu debatu i demokratske institucije uopće. Treća velika negativna strana je moguće narušavanje privatnosti uporabom sustava preporučivanja. S obzirom da su kod suradničkog filtriranja potrebne informacije drugih korisnika za izradu preporuka, narušavanje privatnosti naprosto postaje neizbježno te se događa u 4 faze.

U prvoj se fazi narušavanje privatnosti događa prilikom spremanja podataka bez dopuštenja korisnika. U drugoj fazi dolazi do opasnosti da preuzeti podaci slučajno ne procure ili da podaci više ne budu anonimni. U trećoj je fazi opasnost u izradi preporuka gdje se iz informacija potrebnih za izradu preporuke mogu izvući neki zaključci o korisniku koji bi mogli narušiti njegovu privatnost. I u zadnjoj fazi sustav ima mogućnosti stvaranja modela korisnika iz podataka koje je prikupio od njega i njemu sličnih korisnika [50].

1.3 Preporučivanje Spotify

S obzirom da je s platforme Spotify preuzet korišteni skup podataka, potrebno je opisati platformu Spotify i sustave preporučivanja koje koristi. Spotify je jedna od najvećih platformi na tržištu streaminga glazbe uz druge konkurente poput Apple Musica, Amazon Musica te Youtube-a. Za zadržavanje korisnika i poboljšavanje korisničkog iskustva Spotify koristi algoritme preporučivanja i strojno učenje. Tijekom godina Spotify uporno ulaže u svoje sustave preporučivanja ne bi li povećao svoju konkurentnost.

Preporučivanje na platformi Spotify se događa na kućnom ekranu gdje se prikazuju liste preporučivanja poput: *Discover weekly*, *B Side*, *Release Radar*, *your mixtapes* i tako dalje. Podaci se za Spotify prikupljaju jako ekstenzivno te se mnoge akcije koje korisnik radi na

platformi strogo prate. Spremaju se preferirani umjetnici, novi umjetnici, najslušaniji popise za reproduciranje, broj minuta koje je korisnik proveo na platformi, povijest korisnikovih reprodukcija pjesama, preskočene pjesme, koliko puta je određena pjesma bila svirana, spremljeni popisi za reproduciranje, interakcije s drugim korisnicima i mnoge druge stvari. Sustav preporučivanja na platformi Spotify daje preporuke s obzirom na ponašanje i karakteristike iz prikupljenih podataka. Istraživački tim u Spotifyju je ovo objasnio ovim riječima: “Korisnici su preplavljeni odabirom onoga što mogu gledati, kupiti, pisati ili slušati te ih sustavi preporučivanja samo navigiraju u tom cijelom procesu”. Spotify koristi suradničko filtriranje i filtriranje temeljeno na sadržaju, a sustav po kojem se posebno ističe je BaRT (*engl. Bandits for recommendations as treatments*).

Sustav BaRT može raditi na dva načina: Jedan je tzv. eksploatacija, a drugi istraživanje. Eksploatacija se upotrebljava kad sustav koristi informacije koje je dobio o korisniku na isti način kako se to radi u običnom suradničkom filtriranju. Neki primjeri podataka koji se koriste su recimo preskočene pjesme, podijeljene pjesme, popisi za reproduciranje i tako dalje. Jedan od glavnih problema sustava koji se isključivo bazira na eksploataciji je manjak relevantnosti kada nema puno informacija o korisniku i o pjesmama s kojima je ovaj imao interakciju.

Za rješavanje ovog problema se koristi istraživanje. Istraživanje se koristi kad sustav preporučivanja preporučuje korisniku pjesme na temelju ostalih informacija koje bi sustav mogao koristiti, odnosno na temelju informacija kojima sustav raspolaže. Neke od takvih informacija su: povijest slušanja, kreirani popisi za reproduciranje, što je u trendu, što je objavljeno i tako dalje. Istraživanje se počelo koristiti zbog nedostataka u funkcijama eksploatacije kod slučajeva gdje nema dovoljno podataka za dovoljno točna predviđanja. Model BaRT-a je sposoban učiti i predvidjeti zadovoljstvo koje se mjeri preko mjerila poput brzine klikanosti (*engl. click-through rate - CTR*) ili vjerojatnosti uporabe te cijelo vrijeme uči i popravljajući svoje greške. BaRT se bazira na poticanom učenju i za to koristi model višeručnog bandita (*Multi-Armed Bandit - MAB*).

MAB uči izvršavanje određene akcije A, s kojom bi imao najveću vjerojatnost da će dovesti do nagrade R i svaka akcija A ovisi na prethodnim akcijama i nagradama. Cilj MAB-a je izabrati akcije koje bi maksimizirale totalnu sumu nagrada. Zbog ignoriranja konteksta, vremena, uređaja, popisa za reproduciranje i ostalih sličnih stvari u procesu izvršavanja MAB-a,

programeri iz Spotify su kreirali kontekstualni MAB. Kontekstualni MAB pamti kontekstualne informacije i evaluira informacije prije negoli se akcija izvrši. Sastoji se od konteksta, modela nagrada, procedure učenja i politike koja uključuje eksploataciju i istraživanje te se ove dvije akcije izvršavaju odvojeno. Spotify za eksploataciju koristi epsilon pohlepni algoritam (*engl. Epsilon greedy algorithm*)

Što se suradničkog filtriranja tiče, Spotify ga koristi za korisnike i glazbu za koje se već prikupila veća količina podataka. Osim ovih navedenih modela, Spotify za preporučivanje koristi obradu prirodnog jezika (*engl. Natural Language processing - NLP*). On se pogotovo koristi u svrhe pretraživanja riječi po forumima, diskusijama, preferencama ljudi za baziranje preporuka. Sustav identificira riječi povezane uz pjesme ili umjetnike te ih klasificira, a rezultat je skup izraza koji pokazuje važnost svakog od izraza u brojčanoj vrijednosti [88].

1.4 Općenito o evaluaciji sustava preporučivanja

Prilikom razvoja sustava preporučivanja uvijek se postavlja pitanje najboljeg sustava preporučivanja za generiranje što veće ekonomske vrijednosti. Kako se unešeni podaci u sustavu preporučivanja konstantno mijenjaju, mali nedostaci u metodi preporučivanju mogu izazvati velike probleme u preporučenim podacima. Stoga je potrebno stalno provjeravati kvalitetu preporuka za optimizaciju sustava i generiranje što kvalitetnijih preporuka. U tu svrhu služi evaluacija [51].

Sposobnosti sustava preporučivanja se mogu kvantificirati raznim mjerama evaluacije te bi se svaki sustav preporučivanja trebao evaluirati prije nego što bude pušten u rad. Evaluacija se dijeli na prediktivnu evaluaciju (evaluacija regresije), evaluaciju klasifikacije, evaluaciju rangiranja, evaluaciju karakterističnu za sustave preporučivanja i poslovne mjere evaluacije.

Mjere evaluacije regresije provjeravaju koliko su bliske ocjene sustava preporučivanjima stvarnim ocjenama koje su ostavili korisnici. Dvije najpoznatije mjere izračunavaju se kao drugi korijen srednje kvadratne pogreške (*engl. Root mean square error - RMSE*) i kao srednja apsolutna pogreška (*engl. Mean absolute error - MAE*). U evaluaciji rangiranja se provjerava kvaliteta ljestvice predviđenih stvari. Neke od mjera evaluacije rangiranja su prosječna preciznost (*engl. Average precision - AP*), srednja prosječna preciznost (*engl. Mean average*

precision - MAP) te normalizirani diskontirani kumulativni dobitak (*engl. Normalized discounted cumulative gain* - NDCG).

Evaluacija klasifikacije procjenjuje sposobnost odlučivanja sustava preporučivanja te je korisna za zadatke poput identifikacije relevantnih stvari. Bazira se na 4 vrste predviđanja: istinita pozitivna (*engl. true positive*), lažna pozitivna (*engl. false positive*), lažna negativna (*engl. false negative*) te istinita negativna (*engl. true negative*) te se iz njih izvlače sve mjere evaluacije klasifikacije. Neke od poznatih mjera evaluacija klasifikacije su preciznost, odziv i F1.

Pošto ove prethodno navedene metode evaluacije samo pokrivaju točnost pojedinih modela, razvijene su metode koje pobliže mogu evaluirati sustave preporučivanja. Prve takve su mjere evaluacije karakteristične za sustave preporučivanja. Dije se na mjere kojima je glavni fokus na preporukama (različitost i pokrivenost) i na mjerama fokusiranim na korisniku (novitet, povjerenje i mjere “churna”).

Također još postoje poslovne mjere evaluacije koje evaluiraju sustav preporučivanja po mjerama koje optimiziraju poslovnu solventnost i uspjeh. Neke od tih mjera su CTR, posvojenost i konverzija (*engl. adoption and conversion*), prodaja i prihod (*engl. sales and revenue*), efekti na distribuciju prodaje (*engl. effects on sales distribution*) i ponašanje korisnika i angažman (*engl. user behaviour and engagement*) [99].

Osim ovih spomenutih, postoje online evaluacije i mjere evaluacije preko korisničkih istraživanja. Online evaluacije se provode u radnom okruženju te se uobičajeno koriste u internetskom marketingu i internetskim trgovinama. Najpoznatiji primjeri takvih metoda su stopa preporuka i profit. Mjere evaluacije preko korisničkih istraživanja koriste metode istraživanja koje se standardno provode u znanstvenim istraživanjima te su najpoznatiji primjeri ResQues i subjektivna korisnička iskustva [100].

Posebno se ističe važnost evaluacije sustava preporučivanja ako uzmemo u obzir sve potrebe određene platforme. Ako je važno da preporuke budu u određenom poretku, onda je važna mjera evaluacije rangiranja. Ako je klasifikacija u pitanju, onda se preporučuje koristiti mjere evaluacije klasifikacije, a za vrijednost regresije se preporučuje koristiti mjere evaluacije regresije. Za filtriranje temeljeno na sadržaju se preporučuje koristiti Jaccardova sličnost u problemima klasifikacije. Ako je važan poredak TOP-N preporuka, onda se koristi NDCG. Ako su prisutne preporuke samo za TOP-N stvari, onda je brzina klikanja korisna [99].

1.5 Ciljevi rada

Cilj ovog diplomskog rada je razvoj različitih metoda preporučivanja glazbe te njihova evaluacija prema različitim mjerilima. U radu se iz podataka generiranih u aplikaciji Spotify za svakog pojedinog korisnika preporučuju adekvatne pjesme.

U sustavima preporučivanja korišteni su sljedeći algoritmi za pronalaženje grupa:

- Singularna dekompozicija matrice
- K-najbližih susjeda
- TF-IDF
- Logistička regresija
- Stablo odlučivanja

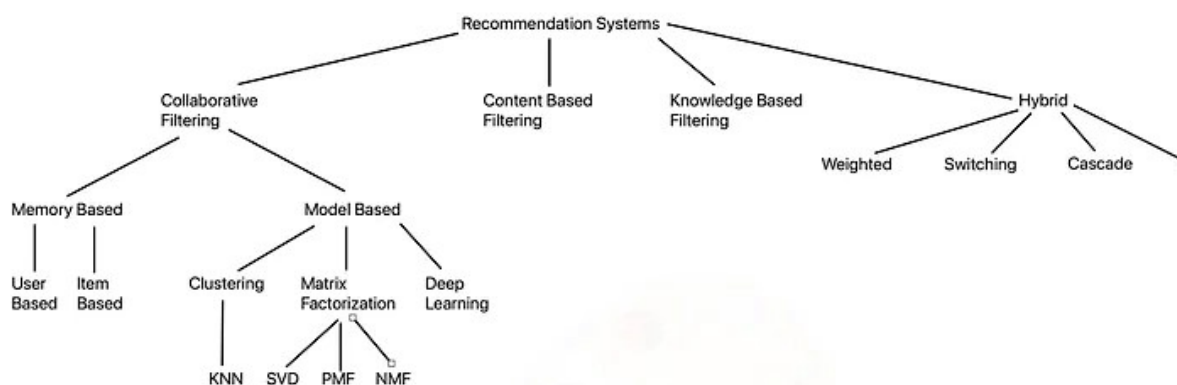
Predviđanja dobivena preko prethodno navedenih metoda preporučivanja se potom evaluiraju sljedećim metodama evaluacije:

- Evaluacija regresije
 - RMSE
 - MAE
- Evaluacija rangiranja
 - NDCG
 - MAP
- Evaluacija klasifikacije
 - Preciznost
 - Odziv
 - F1
- Evaluacija mjera karakterističnih za sustave preporučivanja
 - Novitet (*engl. novelty*)
 - Pokrivenost kataloga (*engl. catalog coverage*)
 - Personalizacija (*engl. personalization*)
 - Različitost (*engl. diversity*)

Svaka od ovih mjera ima drukčiju važnost i ocjenjuje različita obilježja pojedinog sustava preporučivanja. Nakon provedene evaluacije, provedena je usporedba dobivenih vrijednosti evaluacija i donesen je zaključak o najboljim metodama za korištenje.

2. Teorija ili struktura sustava za preporučivanje

Sustavi se preporučivanja dijele u nekoliko kategorija: suradničko filtriranje, filtriranje bazirano na sadržaju, filtriranje bazirano na znanju i hidridno filtriranje. Ova je podjela opisana na slici 2 te su ispod slike detaljnije opisane grane na ovom spomenutom stablu [105]. U radu su obuhvaćene metode suradničkog filtriranja i filtriranja baziranom na sadržaju te su objašnjeni u nastavku rada



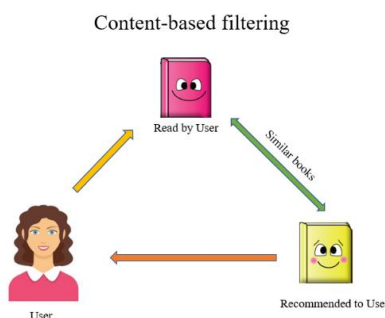
2. Vrste i kategorizacija sustava preporučivanja

Hibridni sustav preporučivanja implementira funkcionalnosti iz suradničkog filtriranja i filtriranja temeljenog na sadržaju te se dijeli na više podvrsta: težinski (*engl. weighted*), pomični (*engl. switching*), mješani (*engl. mixed*), kaskadni (*engl. cascade*) i mnogi drugi. U težinskom se sustavu preporučivanja uzimaju rezultati iz svakog od modela te se kombiniraju po težini bez promjene težine na skupovima za učenje i ispitivanje. Velika je prednost ovog sustava mogućnost integracije više modela preporuka. Pomični hibrid s druge strane uzima jedan sustav preporučivanja za generiranje preporuka ovisno o situaciji. Nakon toga slijedi hibrid koji se klasificira kao miješani koji uzima profil korisnika i njegovih značajki da bi generirao skup preporuka istovremeno koristeći različite sustave preporučivanja [89].

Sustav preporučivanja koji je baziran na znanju je specifičan tip sustava preporučivanja kojem je temelj na eksplicitno znanje o asortimanu stvari, preferencama korisnika i kriteriju preporučivanja. Dijeli se na sustav preporučivanja koji je baziran na ograničenjima te onaj baziran na slučajevima [90].

2.1 Filtriranje temeljeno na sadržaju

Filtriranje temeljeno na sadržaju je način preporučivanja kojem se preporuke temelje na podacima o sadržaju kojeg je korisnik obrađivao ili osvrtima koje je ostavio. Kreira se profil korisnika koji sadrži informacije o preferencama korisnika i o povijesti interakcija korisnika na pojedinim web platformama. Odabiru se stvari na temelju korelacije između sadržaja stvari i interesa korisnika, kao što se može viditi na slici 3 gdje je opisan taj cijeli proces [106]. Prednosti ovog tipa sustava preporučivanja su to da je preporučivanje neovisno o drugim korisnicima i da se поближе mogu preporučiti sadržaji po mjeri određenog korisnika. S druge strane negativna strana ovog algoritma je to da preporučuje samo po već iskazanim interesima korisnika što ga limitira po nekim pitanjima [4].



3. Način preporučivanja u sustavu preporučivanja temeljenom na sadržaju

Filtriranje na temelju sadržaja preko naučenih podataka izrađuje korisnički model koji omogućuje da sustav filtriranja klasificira stvari s kojima korisnik nije imao interakcije na pozitivnu (relevantnu korisniku) i negativnu (nerelevantnu korisniku) klasu. Naučeni podaci ili skupovi se sastoje od stvari koje zanimaju tog određenog korisnika, dok atribut поближе specificira klasu stvari po rejtingu korisnika ili po implicitnim dokazima. Formalno se stvar opisuje kao vektor n komponenti, a komponente mogu imati binarne, nominalne ili numeričke attribute. Atributi se dobijaju ili preko sadržaja stvari ili preko informacija o korisnikovim preferencama. Zadatak učenja metode je odabrati funkciju bazirajući se na naučenom skupu od m vektora unosa koji mogu klasificirati bilo koju stvar u kolekciji. Funkcija $h(X)$ će klasificirati neviđenu stvar kao pozitivnu ili negativnu vraćajući binarnu ili numeričku vrijednost. Filtriranje temeljeno na sadržaju odabire stvar po korelaciji između sadržaja stvari i korisnikovih preferencama nasuprot suradničkom filtriranju koje odabire stvari po korelaciji sa sličnim korisnicima [5].

2.1.1 Frekvencija izraza – Inverzna frekvencija dokumenta

Frekvencija izraza - Inverzna frekvencija dokumenta (*engl. term frequency-inverse document frequency – TF-IDF*) je statistička mjera koja evaluira relevantnosti riječi u dokumentu ili kolekciji dokumenata te je dio sustava preporučivanja temeljenih na sadržaju. Postoje dva načina kojima se mjeri frekvencija ponavljanja riječi: brojanje ponavljanja određene riječi u jednom dokumentu (TF) i inverzno brojanje frekvencija ponavljanja određenih riječi kroz više dokumenata (IDF) [3]. Velika prednost korištenja IDF-a je korekcija u spremanju i procesiranju riječi koje se često ponavljaju, a smetaju u procesu klasifikacije. Najčešće se radi o prijedlozima, veznicima i česticama.

U jednadžbi (1) je predstavljena formula za IDF koja se računa kao logaritmirani količnik broja dokumenata i sume ponavljanja određenih riječi u dokumentu. U jednadžbi (1) t predstavlja broj ponavljanja određenih riječi u svim dokumentima i N je broj dokumenata unutar jedne kolekcije dokumenata.

$$idf(t, D) = \log\left(\frac{N}{count(d \in D : t \in d)}\right) \quad (1)$$

Izračun vrijednosti TF-IDF se dobije umnoškom vrijednosti TF i IDF kao što se vidi u jednadžbi (2). Što je veća vrijednost TF-IDF to je izraz važniji (relevantniji), a kad je rezultat manji (bliži nuli) onda izraz postaje manje važan [38].

$$tfidf(t, d, D) = tf(t, d) * idf(t, D) \quad (2)$$

Ova je metoda popularna kod dohvata podataka i ekstrakcije ključnih riječi u raznim aplikacijama i programima [39], među kojima je najpoznatiji Google [40]. Također se koristi u procesiranju prirodnih jezika s raznim alatima u Pythonu, kao što je to napravljeno u ovom radu [39]. TF-IDF algoritmi pomažu dešifrirati riječi dobivene iz NLP-a alocirajući ih s numeričkom vrijednosti ili vektorom te onda sortiraju te iste riječi u kategorije pa izvlače ključne riječi [3].

Ova metoda ima razne prednosti i mane. Jedna od glavnih prednosti TF-IDF-a je svakako jednostavnost korištenja te se zato često koristi kao polazna točka za izračun nekih drugih vrijednosti, poput kosinusne sličnosti. Kosinusna sličnost je mjera sličnosti koja se koristi u usporedbi dvaju različitih vektora računajući kut između različitih vektora. Također se dosta koristi u kontekstu sustava preporučivanja.

S druge strane negativna je strana TF-IDF-a neprepoznavanje semantičkih značenja i konteksta izraza. Tom se problemu može doskočiti koristeći NER (*engl. Named Entity Recognition*). Osim toga, ne uzima u obzir poredak riječi u izrazu i neefikasno koristi memoriju u matrici malih dimenzija [38].

2.2 Suradničko filtriranje

Suradničko filtriranje (*engl. collaborative filtering*) temelji svoje preporuke na interakcijama između sličnih korisnika kroz povijest u svrhu preporučivanja novih stvari. Svi se korisnici uzimaju u obzir te se preference korisnika sa sličnim interesima i ukusima koriste za preporuku proizvoda svakom od korisnika. Značajke koje opisuju stvari nisu važne jer se za preporučivanje isključivo koriste interakcije između korisnika i stvari. Interakcije se bilježe u matrici korisnik-stvar gdje retci predstavljaju korisnike, a stupci predstavljaju stvari.

U jednadžbi (3) je predstavljen matematički izraz $K \times M$ matrice korisnik-stvar za M stvari i K korisnika, koju smo nazvali X . Svaki element $x_{k,m} = r$ označava to da je korisnik k ocjenio stvar m s ocjenom r , $r \in \{1, \dots, |r|\}$, dok je u slučaju nepoznate ocjene $x_{k,m} = \emptyset$. Svaki vektor reda u_k^T predstavlja profil korisnika i ocjenu pojedinog korisnika za pojedinu stvar [8]. Matrica korisnik - stvar se može predstaviti s vektorima redaka na ispod objašnjen način.

$$X = [u_1, \dots, u_k]^T, u_k = [x_{k,1}, \dots, x_{k,M}]^T, k = 1, \dots, K \quad (3)$$

Postoje različiti načini kojima se spremaju interakcije između korisnika i stvari. Prvi način je preko korisnikovih pretraživanja i implicitnih povratnih informacija kao što su klikovi, poredak povijesti, pregledavanje određenog sadržaja i tako dalje. Drugi način kojim se mogu spremati interakcije je preko direktnih povratnih informacija od korisnika.

Suradničko se filtriranje dijeli na suradničko filtriranje bazirano na memoriji (*engl. memory-based collaborative approach*), suradničko filtriranje bazirano na modelu (*engl. model-based collaborative approach*) i na hibridno koje kombinira ova dva navedena pristupa. Razlika između dvije glavne verzije suradničkog filtriranja je u tome što se u suradničkom filtriranju baziranom na memoriji koristi isključivo matrica korisnik-stvar za izradu preporuka te se cijeli proces bazira na prethodnim ocjenama i interakcijama. S druge se strane u suradničkom filtriranju baziranom na modelu koriste modeli strojnog učenja za predviđanje i rangiranje interakcija između korisnika i stvari s kojima još nisu imali interakcije [82].

Suradničko filtriranje ima razne prednosti i nedostatke. Prednosti uključuju povećanu izloženost korisnika različitim proizvodima te činjenicu da se tijekom pretraživanja korelacija u matrici korisnik-stvar ne traže detalji o stvarima koje se preporučuju. S druge strane nedostaci suradničkog filtriranja uključuju rijetkost podataka, problem hladnog starta (*engl. cold-start problem*), lošu skalabilnost, nemogućnost identifikacije sinonima te različitost. Problem rijetkosti podataka uzrokuje problem hladnog starta koji opisuje poteškoće suradničkog filtriranja da radi preporuke kad je prevelik broj novih korisnika ili stvari. Razlog tomu je što algoritam suradničkog filtriranja ovisi o povijesti pretraživanja korisnika. Drugi nedostatak je skalabilnost koja ukazuje na problem smanjenja performansi s povećanjem ulaznih podataka [83].

Suradničko se filtriranje koristi na raznim mjestima. Jedan od dojmljivih primjera je Facebook gdje se koristi u preporučivanju prijatelja ili Netflix gdje se koristi za preporuku filmova po preferencama korisnika. Osim ove dvije aplikacije, dobro je poznat sustav suradničkog filtriranja u Amazonu [84].

2.2.1 Algoritam temeljen na memoriji

Algoritam temeljen na memoriji koristi ocjene korisnika kroz povijest da bi pronašao sličnosti između korisnika i stvari te potom preporučuje najbližnjemu korisniku još neviđenu stvar. Postoje dvije vrste algoritma temeljenog na memoriji: baziran na korisniku (*engl. user-based*) i baziran na stvari (*engl. item-based*).

U algoritmu baziranom na korisniku se pronalaze korisnici koji su vidjeli ili ocijenili sličan sadržaj. Nedostatak ovog pristupa je činjenica da postoji više korisnika nego stvari što dovodi do velikih matrica sličnosti što uzrokuje probleme s performancama i memorijom na većim skupovima podataka. Drugi je problem hladnog starta gdje se u sustavima s malom količinom podatka ne mogu dobro obrađivati podaci.

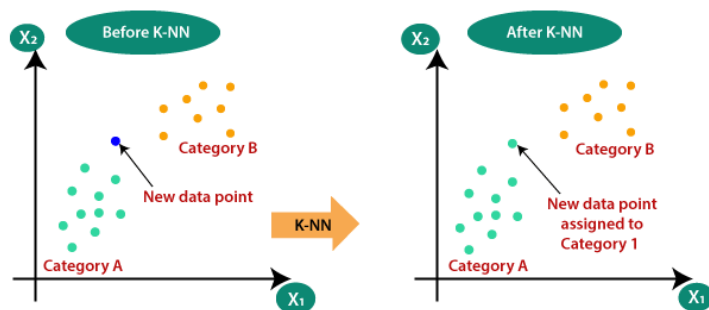
U algoritmu baziranom na stvari se izrađuju preporuke za određenu stvar na temelju drugih preferenci koje određen korisnik ima. Prednost ovog pristupa je manja matrica sličnosti što smanjuje troškove pronalaska susjeda u matrici sličnosti. Isto tako s ovim pristupom nema problema hladnog starta. Nedostatak ovog pristupa je smanjenje različitosti preporuka u usporedbi s pristupom baziranom na korisniku [87].

2.2.1.1 Algoritam K-najbližih susjeda

Algoritam KNN je jednostavna i efektivna metoda klasifikacije koja za klasifikaciju stvari t pronalazi k najbližih susjeda i formira susjedstvo veličine t . Susjedi klasificiraju stvar t bez obzira na razliku težina udaljenosti. Najviše o točnosti i kvaliteti metode odlučuje vrijednost k te se u ovisnosti o točnosti mijenja vrijednost broja k .

KNN je lijena metoda učenja (*engl. lazy learning method*). Lijena metoda učenja je metoda učenja u kojoj se generalizacija treniranih podataka ne događa sve dok se se pojavi upit sustavu. Ova činjenica stoga sprječava KNN u puno uporaba poput dinamičkog rudarenja u velikim repozitorijima. S druge strane se nalazi metoda ustrajnog učenja (*engl. eager learning*) gdje sustav pokušava generalizirati trenirane podatke prije nego dobije upite [26, 27].

Algoritam KNN se sastoji od tri koraka: računanje sličnosti korisnika, selekcija k -najbližih susjeda i predikcija rezultata. U prvom se koraku sličnost među korisnicima računa evaluacijom vrijednosti stvari koje su ocijenili različiti korisnici. Stvara se matrica korisnik-stvar te se pomoću kosinusne ili Pearsonove sličnosti računa sličnost između korisnika i stvari. Nakon računanja sličnosti, odabiru se susjedi s najvećom sličnosti te nakon toga slijedi predviđanje rezultata preko vrijednosti njegovih susjeda [29]. U fazi se predviđanja uzima točka za koju želimo napraviti predviđanje te se računaju udaljenosti te točke od drugih točaka. Tad se pronalazi k najbližih točaka ili susjeda te se ispituju njihove oznake prema kojima se predviđa rezultat. Ovaj je cijeli proces vizualno prikazan na slici 4, gdje se može vidjeti kako se nova točka grupira na temelju ostalih zabilježenih podataka. Za računanje sličnosti kod predviđanja također se koriste jednadžbe za udaljenost poput Euklidove udaljenosti, udaljenosti Minkowskog i Manhattana [28].



4. Grupiranje u k -Najbližih susjeda [107]

KNN ima razne prednosti i nedostatke. Prednosti su da je algoritam jednostavan za implementaciju, da ne zahtjeva dodatna namještanja te da na različite načine koristi razne algoritme (klasifikacija, regresija, pretraga). S druge strane negativne su strane: mala efikasnost, veća cijena komputacije, neprikladnost za velike skupove podataka te ovisnost o izboru prikladne vrijednosti k . [26, 30].

U praksi se KNN preferira koristiti u rješavanju problema čije rješavanje ovisi o identifikaciji sličnih objekata [79]. Unatoč svojim kvalitetama, KNN se danas manje koristi nego prije primarno zbog svoje neefikasnosti i malih performance. Unatoč toj činjenici, danas se još uvijek koristi u sustavima preporučivanja, prepoznavanju uzoraka, rudarenju podataka, predviđanju financijskih tržišta i u mnoge druge svrhe [80].

Uz osnovnu verziju KNN-a koja je obrađena u sklopu ovog rada, postoje i druge verzije ovog algoritma. Neke od tih verzija su: adaptivni KNN (*engl. adaptive KNN*), nejasni KNN (*engl. fuzzy KNN*), težinski raspoređen KNN (*engl. weighted KNN*), zajednički KNN (*engl. futual KNN*), KNN s Hassanatovom udaljenosti (*engl. Hassanat KNN*) itd. [81].

2.3 Algoritam temeljen na modelu

Algoritam temeljen na modelu koristi statistički sustav i strojno učenje za minimizaciju matrice korisnik-stvar, komprimira veliku bazu podataka u model s korisnicima, stvarima i ocjenama te obavlja zadatak preporučivanja koristeći mehanizme odabranog modela [16]. Ova se vrsta algoritma najčešće dijeli na 5 vrsta: grupiranje (*engl. clustering*), klasifikacija (*engl. classification*), latentni model (*engl. latent model*), Markovljev proces odlučivanja (*engl. Markov decision process*) i faktorizacija matrice (*engl. matrix factorization*) [15]. U ovom radu je opisan SVD, koji je podvrsta faktorizacije matrice.

Faktorizacija matrice se bazira na visokoj korespondenciji između korisnika i stvari na kojoj se temelji preporuka. Faktorizacija matrice modelira korisnike i stvari u zajednički latentni prostor faktora dimenzionalnosti f u kojem su interakcije korisnika i stvari modelirane kao unutarnji dijelovi tog prostora. Prema tome je svaka stvar i u prostoru povezana s vektorom $q_i \in R^f$ i svaki je korisnik u povezan s vektorom $p_u \in R^f$. Za svaku stvar i , elementi iz q_i mjere koliko faktora dimenzionalnosti ima ta određena stvar, bilo pozitivnih ili negativnih. Za svakog korisnika u elementi p_u mjere koliko mu je interes za stvari koje imaju visoke vrijednosti korespondentnih

faktora. Preko jednadžbe (29) se dobije rezultat interakcije između korisnika i stvari te interes korisnika za pojedinu stvar [86].

$$\hat{r}_{ui} = q_i^T p_u \quad (29)$$

Neki od najpoznatijih modela vrste faktorizacije matrice su: SVD, analiza glavnih komponenti (*engl. Principal Component Analysis - PCA*), faktorizacija matrice vjerojatnosti (*engl. Probabilistic Matrix Factorization - PMF*), nenegativna faktorizacija matrice (*engl. Non-Negative Matrix Factorization - NMF*) i tako dalje [17]. Faktorizacija je matrice postala sve popularnija zadnjih godina zato što uspijeva u kombiniranju dobre skalabilnosti s prediktivnom točnosti [15].

2.3.1. Singularna dekompozicija matrice

Singularna dekompozicija matrice (SVD) je metoda iz linearne algebre koja se koristi u smanjenju dimenzionalnosti u strojnom učenju. Smanjuje broj podataka u matrici smanjujući veličinu matrice s N-dimenzija na K-dimenzija ($K < N$). Također izvršava faktorizaciju pronalazeći faktore iz glavne matrice korisnik-stvar u kojoj se matrica rastavlja u tri druge matrice [7]. Prva od njih je svakako najvažnija matrica za SVD, a to je matrica korisnik-stvar A (4). Matrica A ima dimenzije $m \times n$ te se izražava kao umnožak matrica U , S i V^T . S druge strane matrica U s dimenzijama $m \times n$ je ortogonalno lijeva singularna matrica koja opisuje snagu svakog od latentnih faktora (5). Latentni faktori označavaju karakteristike stvari, npr. žanr muzike. Matrica V je dijagonalno desna singularna matrica veličine $n \times n$ koja prikazuje sličnost između stvari i latentnih faktora (6). Matrica Σ koja je predstavljena u jednadžbi (7) prikazuje sumu umnoška vektora, gdje je s_i i-ta singularna vrijednost te su u_i i v_i odgovarajući lijevi i desni singularni vektori. Matrica Σ je dijagonalna matrica s nenegativnim vrijednostima i s dijagonalnim vrijednostima sortiranim od viših do nižih [20]. SVD isto tako smanjuje dimenzije matrice A izbacujući latentne faktore [7].

$$A = USV^T \quad U = A * A^T \quad V^T = A^T * A \quad (4, 5, 6)$$

$$A = \sum_{i=1}^{\min(m,n)} s_i * u_i * v_i^T \quad (7)$$

Osim prethodno prikazane, postoje i razni drugi načini primjene SVD-a. Prvi takav primjer je derivativna imitacija SVD-a (SVD++) koja uz osnovne funkcije SVD-a računa implicitne ocjene [21]. SVD++ je prikazan u jednadžbi (8), gdje y_j predstavlja skup faktora stvari koji opisuju implicitne ocjene. Implicitne ocjene opisuju činjenicu da je korisnik u ocijenio stvar j , bez obzira na vrijednost stvari. Ako bi korisnik u bio nepoznat onda bi pristranost (*engl. bias*) b_u i faktori (*engl. factors*) p_u bili nula. Isto vrijedi i za stvar i s b_i, q_i, y_i [22].

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T * (p_u + |I_u|^{-0.5} * \sum_{j \in I_u} y_j) \quad (8)$$

Druga takva varijacija je rijetka singularna dekompozicija matrice (*engl. Sparse singular value decomposition* – SSVD) koja je zapravo analitički alat za biklasteriziranje (*engl. biclustering*) ili identificiranje interpretabilnih asocijacija red-stupac (*engl. identifying interpretable row-column associations*) unutar višedimenzionalnih podatkovnih matrica [24].

Prednosti SVD-a su redukcija dimenzionalnosti, aplikabilnost u različite svrhe te interpretabilnost. Nedostaci su ograničenja u nelinearnim odnosima te osjetljivost na skalu značajki [104].

2.4 Strojno učenje u sustavima preporučivanja

Strojno je učenje vrsta programiranja u kojoj se optimiziraju kriteriji uspješnosti na temelju podatkovnih primjera ili prethodnog iskustva. Tijekom procesa raspolažemo modelom koji je definiran po parametrima koji se definiraju učenjem podataka za učenje. Na temelju tih podataka, model bi trebao predvidjeti svojstva novih neviđenih (ispitnih) podataka te mu je cilj izgrađivanje modela koji dobro generaliziraju.

Strojno se učenje dijeli na tri vrste pristupa: nadzirano, nenadzirano i podržano učenje. Nadzirano učenje je definirano kao funkcija $\hat{y} = f(x)$ te se može dodatno podijeliti u klasifikaciju i regresiju. U klasifikaciji je y diskretna vrijednost, dok je kod regresije y kontinuirana vrijednost. Kod nenadziranog učenja su dani podaci bez ciljne vrijednosti u kojima se traži pravilnost u podacima. Ono se dijeli na grupiranje, procjenu gustoće i smanjenje dimenzionalnosti. Podržano učenje je učenje optimalne strategije na temelju pokušaja s odgođenom nagradom. Strojno učenje se koristi u raznim primjenama, od složenih problema

gdje ljudsko znanje nije dovoljno, sustava koji se dinamički mijenjaju pa sve do sustava s ogromnom količinom podataka [78].

2.4.1 Logistička regresija

Logistička je regresija (*engl. Logistic regression*) definirana kao skup statističkih modela koji opisuju odnose između kvalitativnih zavisnih ili nezavisnih varijabli. Modeli logističke regresije se koriste za proučavanje efekata varijabli prediktora na kategoričke ishode [52]. Ovaj se tip statističkih modela često koristi za klasifikaciju i prediktivnu analitiku u svrhe procjene vjerojatnosti nekog događaja na bazi skupa nezavisnih varijabli. Vrijednost rezultata logističke regresije je ili nula ili jedan, tj. zavisna je varijabla binarna [53]. Logističku regresiju ne treba miješati s linearnom regresijom kod koje je zavisna varijabla kontinuirana.

Logistička se regresija sastoji od 4 koraka. U prvom se koraku podaci formatiraju u oblik gdje svaki redak predstavlja jednu observaciju te svaki stupac predstavlja različitu varijablu. Iza toga slijedi učenje modela gdje se model uči na odabranim podacima. Iza toga slijede evaluacija i korištenje modela u svrhu stvaranja predviđanja. Logistička regresija se izražava logističkom funkcijom koja je prikazana jednadžbom (9).

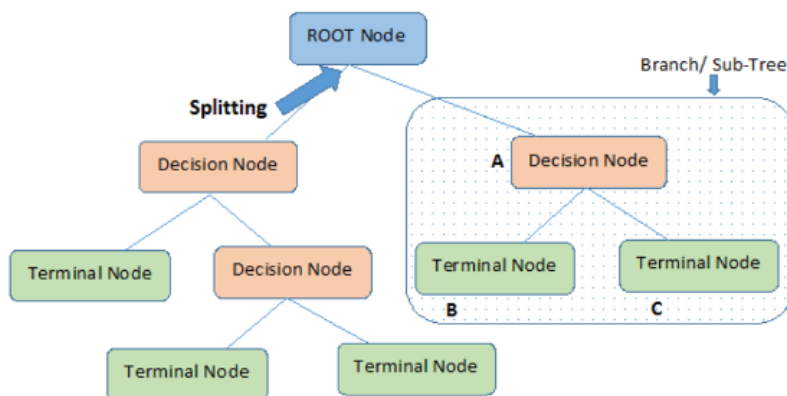
$$P = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}} \quad (9)$$

Ova se funkcija tijekom izvođenja stalno iterativno optimizira tako što se traže optimalne vrijednosti koeficijenata β za minimizaciju razlike između predviđenih i pravih vrijednosti. Najčešće korišteni algoritam za minimizaciju razlike je algoritam gradijentnog spusta (*engl. gradient descent*). Ovaj algoritam mijenja vrijednosti koeficijenata β tako da funkcija uvijek konvergira u točku minimalne vrijednosti, tj pokušava pronaći optimalne vrijednosti koje minimiziraju funkciju gubitka (*engl. loss function*) ovog modela [54].

Postoje različite prednosti logističke regresije. Logistička regresija je jednostavnija za korištenje naspram drugih modela strojnog učenja zbog svoje fleksibilnosti, jer daje odgovore na pitanja koja imaju dva ili više ishoda. Osim toga, ovo je dosta brz algoritam koji se lako koristi. S druge strane se u nedostatke ubrajaju stvaranje linearnih granica, asumpcija linearnosti te nemogućnost rješavanja nelinearnih problema. Logistička se regresija koristi u raznim poljima: od proizvodnje, zdravstva, financija, marketinga, sustava preporučivanja itd. [55].

2.4.2 Stablo odlučivanja

Stablo odlučivanja (*engl. Decision tree classifier*) je algoritam nadziranog učenja bez parametara koji se koristi za klasifikacijske i regresijske zadatke. Sastoji se od hijerarhijske strukture stabla koja u sebi ima korijenski čvor (*engl. root node*), grane (*engl. branch*), unutarnje čvorove (*engl. internal node*) i krajnje čvorove (*engl. leaf node*) te koristi uvjete za raspoređivanje čvorova. Korijenski čvor je početni čvor u stablu odlučivanja te u tom čvoru započinje podjela na stablu. Unutarnji čvor je čvor koji nije korijenski, a ima druge čvorove ispod sebe. Krajnji čvor je posljednji čvor ispod kojeg nema drugih čvorova [56]. Ova je podjela vizualno opisana na slici 5.



5. Stablo odlučivanja [108]

Stabla odlučivanja imaju nekoliko pravila. Jedno od tih pravila su binarne podjele, gdje stablo odlučivanja radi binarne podjele na čvorovima u dva podskupa na temelju jedne značajke ili uvjeta. Drugo važno pravilo je rekurzivna podjela gdje se svaki čvor dijeli u čvorove djecu sve dok ne bude ispunjen kriterij stajanja (*engl. stopping criterion*). Treće važno pravilo je pohlepni pristup (*engl. top-down greedy approach*) koji kaže da se stabla odlučivanja izgrađuju odozgora prema dolje s mišlju da se što više poveća dobitak informacija i da se što više smanji nečistost trenutnog čvora. Ostala važna pravila su: neovisnost značajki, homogeničnost, kategoričke i numeričke značajke, prevelika naučenost, jednaka važnost značajki, osjetljivost na veličinu sadržaja itd. [56].

Proces izgradnje stabla odlučivanja funkcionira na način da se prostor značajki sve više smanjuje u sve manje i manje podskupe. Na svakom koraku stablo odlučivanja odabire

najvažnije značajke za odabir tražene varijable te potom algoritam dijeli prostor značajki po izabranoj značajki stvarajući dva nova podskupa. Ovaj se proces ponavlja sve dok prostor značajki ne postane dovoljno malen i homogen.

Sustavi preporučivanja koji su bazirani na stablu odlučivanja su jednostavni za razumijevanje i interpretaciju, jer se za svako predviđanje može vidjeti kako se do njega došlo. Uz to, ovaj je algoritam jako brz i efikasan što ga čini jako korisnim za sustave preporučivanja gdje ima mnogo podataka. S druge strane nedostatak ovakvog sustava preporučivanja jest da se vrlo lako može dogoditi prenaučenosť algoritma, posebno u slučaju kad je stablo preduboko. Razlog iz kojeg se to događa se nalazi u prevelikoj kompleksnosti pojedinačnih modela koji dobro uče podatke za učenje uz lošu generalizaciju i nepodobnost uporabe s novim podacima. Stoga je jako važno znati dobro unijeti ulazne podatke da bi se dobila što preciznija vrijednost [57].

2.4.3 Klasifikator K susjeda

Klasifikator K-susjeda (*engl. K-Neighbors Classifier*) se upotrebljava kao algoritam strojnog učenja koji koristi algoritam K-najbližih susjeda za klasifikacijske zadatke. Klasifikator K-susjeda je vrsta nadziranog učenja koja ne izgrađuje svoj interni model, već samo sprema instance naučenih podataka. Klasifikacija se određuje iz većine glasova najbližih točaka za svakog elementa te svaka točka ima jednaku težinu. Postoji također i verzija gdje se daje veća težina točkama koje su bliže odredišnoj točki i verzija za regresiju[103].

2.4.4 Epsilon pohlepni algoritam

Epsilon pohlepni algoritam je algoritam tipa MAB koji istražuje i eksploatira podatke. Daje upute programu u procesu istraživanja (odabir nasumične opcije s epsilonom vjerojatnosti) te eksploatira (uzima opciju koju smatra za najbolju). Kako vrijeme protiče, program odabire različite opcije te dobija smisao toga koje bi opcije mogle imati najveću povratnu nagradu. No ponekad će algoritam izabrat nasumičnu akciju da bi algoritam bio uvjeren da slučajno ne postoje neki zaostali elementi [58].

Epsilon pohlepni algoritam na svakom koraku odlučuje tako da preko AI-ja generirani agent nasumično odabire akciju s vjerojatnosti epsilon [59]. Algoritam provjerava je li vjerojatnost manja ili veća od epsilona. S vjerojatnosti manjom od epsilona izvodi nasumičnu operaciju, a ako je vjerojatnost veća od epsilona, onda izvodi trenutno najbolju moguću operaciju. Epsilon

se najčešće stavlja na vrijednost 0.1 čime se osigurava da se u otprilike 90% slučajeva izabire najbolja trenutno moguća operacija. Pseudokod ovog je algoritma opisan na slici 6.

```
p = random()

if p < ε:
    pull random action
else:
    pull current-best action
```

6. Algoritam Epsilon [109]

Ovakvim se procesom osigurava stalno traženje alternativnih akcija za prikupljanje podataka dok se istovremeno kapitaliziraju akcije koje su polučile najbolje uspjehe. Epsilon pohlepni algoritam se koristi u razne svrhe: treniranje autonomnih robota u kompleksnim okruženjima, razvoj adaptivne i progresivne umjetne inteligencije za računalne igrice te unapređenje sustava preporučivanja [59]. Epsilon pohlepni algoritam nije korišten u kontekstu ovog rada, ali je važna komponenta u preporučivanju na platformi Spotify.

2.5 Načini evaluacije rezultata sustava preporučivanja

Evaluacija sustava preporučivanja služi provjeri kvalitete sustava preporučivanja, a odabir metode evaluacije ovisi o problemu koji se rješava. Problemi koji se rješavaju mogu se podijeliti u dvije kategorije: regresija i klasifikacija. Regresija odgovara na pitanje koliko je nečega te daje numerički odgovor. S druge strane klasifikacija daje odgovor u obliku pripadnosti jednoj ili drugoj klasi [61].

Tijekom procesa evaluacije sustava preporučivanja najbitnije su tri stvari: koju mjeru evaluacije koristiti, kako realistično procijeniti pogrešku modela te na koji način napraviti statističku analizu rezultata da bi se uvjerali da rezultat nije puka slučajnost. Mjera vrednovanja evaluacije kvantificira točnost ili pogrešku te se uvijek pokušava pronaći metoda koja bi to najbolje napravila. Tijekom ovog se procesa sustav upotrebljava na skupu primjera (ispitni skup) tako da se za svaki primjer dobije njegova predikcija. Rezultati dobiveni predikcijom se predstavljaju kao vektori predviđenih oznaka (10), koji se u evaluaciji potom uspoređuju s vektorima stvarnih oznaka (11)

$$y_{pred} = (y_{pred}^1, \dots, y_{pred}^i, \dots, y_{pred}^N)^T \quad (10)$$

$$y_{true} = (y_{true}^1, \dots, y_{true}^i, \dots, y_{true}^N)^T \quad (11)$$

Matrica zabune (*engl. confusion matrix*) posebno pomaže kod odabira mjere evaluacije algoritma. Matrica zabune je kvadratna matrica koja prikazuje broj podudaranja i nepodudaranja predikcija i stvarnih oznaka. Najjednostavniji primjer matrice zabune je u slučaju binarnog klasifikatora, gdje bi matrica zabune izgledala kao u primjeru ispod (12). No više o tome u poglavlju o klasifikaciji

$$\begin{matrix} & y_{true} = 1 & y_{true} = 0 \\ \begin{matrix} y_{pred} = 1 \\ y_{pred} = 0 \end{matrix} & \begin{pmatrix} TP & FP \\ FN & TN \end{pmatrix} \end{matrix} \quad (12)$$

Prema ovome možemo napraviti klasifikaciju u četiri slučaja:

- stvarno pozitivni (*engl. true positive* - TP) – predviđeno je 1, stvarna oznaka je 1
- lažno pozitivni (*engl. false positive* - FP) – predviđeno je 1, stvarna oznaka je 0
- lažno negativni (*engl. false negative* - FN) – predviđeno je 0, stvarna oznaka je 1
- stvarno negativni (*engl. true negative* - TN) – predviđeno je 0, stvarna oznaka je 0

Uz odabir mjere evaluacije, mora se dobro procijeniti pogreška modela. Za dobru procjenu pogreške modela je potrebna nepristranost, poštenost i konzistentnost procjenitelja. Nepristranost označava činjenicu jednakosti očekivanja na pravoj vrijednosti populacije, konzistentnost podrazumijeva smanjenje varijance procjene rastom veličine uzorka, dok poštenost označava sposobnost generalizacije pojedinog modela. Također tim je bolje ukoliko je procjena pogreške pesimistična nego optimistična, zato što će sustav preporučivanja onda raditi tako ili još bolje od te vrijednosti [62].

Osim prethodno navedenih mjera evaluacije, za evaluaciju sustava preporučivanja koriste se posebne mjere karakteristične za sustave preporučivanja. Kroz rad i evaluaciju sustava preporučivanja je primjećeno kako samo računanje točnosti pojedine evaluacije nije dovoljno za potrebe evaluacije sustava preporučivanja. Stoga su stvorene mjere evaluacije sustava preporučivanja kojima se evaluira opseg istraživanja novih stvari. Neke od tih mjera evaluacije su pokrivenost, novitet, različitost, nDCG, MAP, prosječno recipročno rangiranje (*engl. mean reciprocal rank* - MRR) itd. [64].

Važnost evaluacije modela je u osiguravanju performansi modela za njegovu optimalnost i pouzdanost. Netočna ili nepotpuna evaluacija modela može biti katastrofična za korisničko iskustvo i prihode tvrtke. Evaluacija se izvodi i tijekom razvijanja programa i tijekom uporabe programa [63].

Proces evaluacije je dobro opisan na slici 7, gdje se evaluacija sustava preporučivanja prikazuje kao proces usporedbe stvarnih korisničkih podataka i predviđenih vrijednosti na temelju stvarnih korisničkih podataka.

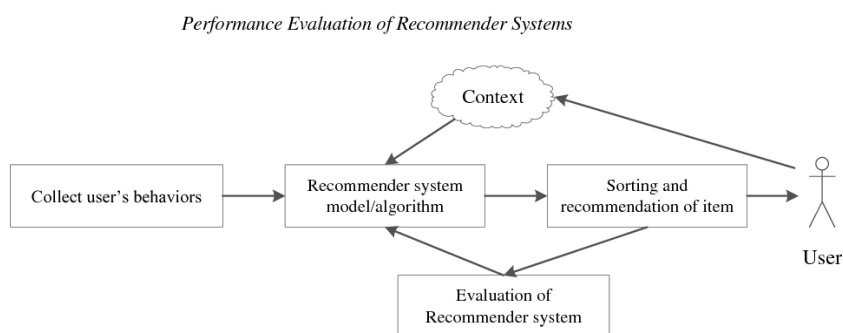


Figure 1. The module diagram for recommender system's design and evaluation

7. Evaluacija [110]

2.5.1 Računanje procjene pogreške u evaluaciji

Za provođenje evaluacije modela sustava preporučivanja je potreban dobar način procjene pogreške, za što se uglavnom koriste statističke metode.

Prva metoda koja je korištena u radu je metoda izdvajanja (*engl. holdout method*) ili train-test metoda, gdje se skup primjera razdvaja u skup za učenje i ispitni skup [62]. Skup za učenje se koristi za učenje modela, dok se ispitni skup koristi za evaluaciju modela. Postotci po kojima se raspoređuje skup podataka ovise o količini dostupnih podataka, najčešće je to podjela 70%-30% dok se u ovom radu koristila podjela 80%-20%. Ova tehnika na ispitnim podacima odabire najbolji model i optimizira proces [77]. Velika je prednost ove metode u tome što se ovdje računa pogreška generalizacije umjesto pogreške učenja. Pogreška generalizacije računa točnost algoritma na neviđenim (ispitnim u ovom slučaju) podacima, dok pogreška učenja računa točnost naučenih podataka. S druge strane nedostaci ove metode su u slučajevima izostanka primjera za učenje čime se onda gubi velik broj vrijednih informacija za poboljšavanje modela. Drugi veliki nedostatak je lošija točnost procjene [62].

Druga metoda koja je korištena u radu je k-struka unakrsna provjera gdje se skup primjera dijeli u K podskupova te je svaki od tih podskupova podijeljen u k preklopa. Model se uči na k-1 preklopa, dok se k-ti preklop uzima za ispitni skup. Nakon toga se ovaj algoritam provede k puta za sve preklope. Na kraju se izračuna prosjek svih dobivenih vrijednosti, što je ujedno i krajnji rezultat.

Prednost ove metode je u tome što se svaki primjer koristi za ispitivanje i to točno jednom, jednostavno se provodi, točan je, efikasno koristi podatke i daje dobar uvid u performance modela. S druge strane nedostaci su mu nedostatak nezavisnosti između k klasifikatora što dovodi do visoke varijance procjene pogreške [62]. Način podjele ovog načina evaluacije je prikazan na slici 8



8. Unakrsna validacija [62]

Uz ove navedene metode postoje brojne druge metode koje nisu korištene u radu. Jedan zanimljiv primjer je unakrsna provjera “izdvoji jednog” (*engl. Leave-One-Out Cross validation - LOOCV*). LOOCV funkcioniра na isti način kao k-struka unakrsna provjera, samo što u podjelama svaki primjer postaje svoj preklop ($k=N$).

Prednost ove metode je iskoristivost cijelog skupa primjera i bolja procjena pogreške, a nedostaci su računalna zahtjevnost i velika varijanca procjene pogreške. Druge dvije zanimljive metode su unakrsna provjera uz odabir modela i ugniježdена k-struka unakrsna provjera.

2.5.2 Evaluacija regresije

Mjere evaluacije regresije su kvantitativna mjerila evaluacije koja se uglavnom koriste za evaluaciju regresijskih modela. Dva načina regresijskih mjera evaluacije koji su korišteni u ovome radu su srednja pogreška korijena kvadrata (RMSE) i srednja apsolutna pogreška (MAE).

RMSE se izražava kao korijen prosječne kvadrirane razlike predviđenih i stvarnih vrijednosti te je izražen u jednadžbi (13) [61]. Kad podaci imaju bliske vrijednosti, onda je pogreška RMSE manja. A kad je međusobna udaljenost veća, onda je ujedno i pogreška veća. Vrijednost RMSE može varirati od 0 do pozitivne beskonačnosti, gdje 0 označava veću točnost dok veće vrijednosti označavaju manje točan model ili algoritam. Neke od prednosti RMSE-a su intuitivna interpretacija i standardnost (može se u razne uporabe upotrijebiti). S druge strane nedostaci ovakvog mjerenja pogreške uključuju osjetljivost na prenaučenosť, osjetljivost na iznimke i veličinu uzorka. Razlog tome je što se svaka razlika između predviđenih i stvarnih vrijednosti gleda nezavisno jedna od druge. RMSE je jedna od ključnih mjera u prediktivnom modeliranju [65].

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_{pred} - y_{true})^2} \quad (13)$$

Drugi način računanja evaluacije regresije koji je korišten u ovom radu je MAE. MAE računa prosječnu vrijednost pogreške unutar skupa predviđanja bez uzimanja njihovog smjera u obzir (14). Ova mjera označava prosjek apsolutnih razlika predviđanja i stvarnih vrijednosti unutar ispitnog skupa između gdje sve razlike imaju istu težinu [66]. Isto kao i kod RMSE-a, što je manja vrijednost MAE to je točnost predviđanja veća, a veća vrijednost MAE označava manju točnost. Neke od prednosti MAE su jednostavnost korištenja, interpretabilnost te pogodnost korištenja sa skupom podataka koji sadrži ekstremne vrijednosti. Neki od nedostataka MAE su nedostatak računanja smjera greške te neosjetljivost na precijenjenost i podcjenjenost. Što se tiče praktičnih upotreba, MAE se koristi kod predviđanja vremenskih prognoza i u finansijskom sektoru [66].

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_{pred} - y_{true}| \quad (14)$$

Uz ove dvije navedene postoje i razni drugi načini evaluacije regresije. Neki od primjera su: R na kvadrat (*engl. R-Squared*), srednja apsolutna devijacija (*engl. mean absolute deviation - MAB*), srednja apsolutna pogreška postotka (*engl. mean absolute percentage error - MAPE*), srednja apsolutna skalirana pogreška (*engl. mean absolute scaled error - MASE*) i srednja kvadrirana greška predviđanja (*engl. mean squared prediction error – MSPE*). Najviše se koriste u suradničkom filtriranju na unakrsnoj validaciji preko K-folda [99].

2.5.3 Evaluacija klasifikacije

Klasifikacija je predviđanje klasa u koje bi se pojedini podaci mogli svrstati te se dijeli u binarnu (*engl. binary*) i višeklasnu (*engl. multiclass*). Postoje različiti načini izražavanja evaluacije klasifikacije kao što su: preciznost, odziv i mjera F1. Vrijednosti evaluacije klasifikacije izlaze iz matrice zabune koja je objašnjena u prethodnim poglavljima.

Mjera preciznosti prikazuje količinu točno predviđenih slučajeva koji su i stvarno pozitivni. Računa se kao količnik stvarno pozitivnih i zbroja stvarno i lažno pozitivnih predviđanja (15). Posebno je korisna u slučajevima gdje su lažno pozitivni rezultati važniji nego lažno negativni rezultati [68]. Vrijednosti preciznosti variraju od 0 do 1, gdje je 1 najtočnija vrijednost.

$$PRECIZNOST = \frac{TP}{TP + FP} \quad (15)$$

S druge strane mjera odziva se računa kao količnik stvarno pozitivnih i zbroja stvarno pozitivnih i lažno negativnih vrijednosti (16). Mjera se zove odziv jer govori koliko se pozitivnih primjera odazvalo klasifikatoru. Vrijednosti odziva variraju od 0 do 1, gdje je 1 najtočnija vrijednost kad su svi pozitivni primjeri označeni kao odazvani.

$$ODZIV = \frac{TP}{TP + FN} \quad (16)$$

Kao što se vidi u prethodnim primjerima, odziv i preciznost daju različite informacije te su ove dvije mjere često izravno suprotstavljene. Ovo je posebno vidljivo u činjenici da s višim odzivom dolazi i do smanjenja preciznosti i obrnuto. Da bi se bolje utvrdila točnost klasifikatora na temelju ove dvije mjere je osmišljena mjera F1. Mjera F1 je definirana kao harmonijska sredina preciznosti i odziva zbog različitosti rezultata preciznosti i odziva (17).

Kroz računanje harmonijske sredine daje se jednaka važnost i preciznosti i odzivu te je rezultat F1 uvijek bliži manjoj vrijednosti od te dvije[62].

$$F1 = \frac{2 * PRECIZNOST * ODZIV}{PRECIZNOST + ODZIV} \quad (17)$$

Postoje osim navedenih i razni drugi načini evaluacije klasifikacije kao što su računanje točnosti (*engl. accuracy*) (18), specifičnosti (*engl. specificity*) (20), ROC (*engl. receiver operating characteristics*), AUC (*engl. area under ROC curve*), ispadanja (*engl. fall-out*) (19) itd. [62, 68].

$$TOCNOST = \frac{TP + TN}{TP + FP + FN + TN} \quad (18)$$

$$ISPADANJE = \frac{FP}{FP + TN} \quad (19)$$

$$SPECIFICNOST = \frac{TN}{FP + TN} \quad (20)$$

U svrhu rada su također provedeni izračuni za mikro (*engl. micro*), makro (*engl. macro*) i težinski (*engl. weighted*) način izračuna evaluacije klasifikacije. Mikro evaluacija klasifikacije računa globalni prosjek bez obzira na klasu svake od vrijednosti. Zbog toga je mikro evaluacija klasifikacije pogodna u kontekstu kad je korišteni skup podataka balansirani. Makro evaluacija se dobije računajući aritmetičku sredinu mjere evaluacije po svakoj klasi rezultata. Metoda tretira sve klase jednako bez obzira na njihove potporne vrijednosti i preporučuje se koristiti kad su u pitanju nebalansirani skupovi podataka.

Osim ova dva spomenuta, u radu je korišten težinski izračun evaluacije klasifikacije koji balansira perspektivu izračuna. Težinski se način preporučuje koristiti kad su u pitanju nebalansirani skupovi podataka gdje je potrebno raspodijeliti težinu da se prednost daje klasama s više primjeraka [101]. Osim ovih navedenih može se računati još i binarni prosjek (*engl. binary*) i prosjek primjera (*engl. samples*). Binarni isključivo računa rezultate odabranih klasa, dok prosjek primjera računa svaku instancu te pronalazi njihove prosjeke.

2.5.4 Evaluacija rangiranja

Cilj uspostavljanja evaluacije rangiranja je sortiranje liste dokumenata na način koji najviše odgovara krajnjem korisniku, a to se postiže tako da najrelevantniji dokumenti završe na vrhu ljestvice [69]. Rangiranje je svakako jedna od kritičnih točki u svakom sustavu ili aplikaciji zbog potrebe poboljšavanja korisničkog iskustva i efikasnog sortiranja relevantnih informacija [70]. U sklopu ovog rada su korištena mjerila normaliziranog diskontiranog kumulativnog dobitka (NDCG) i srednje prosječne preciznosti (MAP).

NDCG je mjera rangiranja kvalitete ili relevantnosti proizvoda. NDCG stavlja relevantnije stvari ispred nerelevantnih te visoka vrijednost NDCG pokazuje veću relevantnost pojedinih stvari. Osim toga, može se koristiti kao proxy evaluacija korisničkog zadovoljstva, jer razumije razlog bolje kvalitete jednog proizvoda naspram drugog [71]. NDCG se dobije kao količnik diskontiranog kumulativnog dobitka (*engl. discounted cumulative gain* - DCG) i idealnog kumulativnog dobitka (*engl. ideal distcounted cumulative gain* - IDCG) kao što je prikazano u jednadžbi (21).

$$NDCG = \frac{DCG}{IDCG} \quad (21)$$

DCG računa rezultat relevantnosti za pojedino predviđanje. IDCG je s druge strane maksimalna vrijednost DCG-a koja bi se mogla dobit ako bi rezultati bili idealno rangirani. Maksimalna se vrijednost DCG-a dobije kad je najviša moguća vrijednost na vrhu, a najniža na dnu ljestvice. U jednadžbi (12) je prikazana formula za DCG, u kojoj se preko različitih varijabli računa vrijednost izraza DCG. Varijabla rel_1 prikazuje relevantnost prve stvari na ljestvici te se s njom zbraja izraz koji prikazuje sumu rezultata diskontirane relevantnosti od druge do zadnje pozicije. Varijabla rel_i izražava relevantnost stvari na poziciji i , a izraz $\log_2(i+1)$ diskontira rezultat relevantnosti tako što daje veću težinu stvarima na višim pozicijama na ljestvici

$$DCG_p = rel_1 + \sum \left(\frac{rel_i}{\log_2(i+1)} \right) for \dots i \dots in \dots [2, p] \quad (22)$$

NDCG je posebno koristan za praćenje performanci modela rangiranja. Stalnim se praćenjem vrijednosti NDCG-a može primijetiti bilo kakav nedostak u performancama modela [72].

MAP je mjera za kvantifikaciju kvalitete rangiranja predviđenih rezultata nasuprot stvarnim vrijednostima. MAP se dobija od vrijednosti prosječne preciznosti (engl. average precision - AP) koja predstavlja sumu umnožaka svih vjerojatnosti i relevantnosti za pojedine stvari k te potom umnožak dijeli s ukupnim brojem stvari m (23). MAP je prosjek izračuna AP za sve korisnike $|U|$ u skupu korisnika, tj. prosjek prosjeka AP (24) [73].

$$AP@N = \frac{1}{m} \sum_{k=1}^N p(k) * rel(k) \quad (23)$$

$$MAP@N = \frac{1}{|U|} \sum_{u=1} |U| * (AP@N)_u = \frac{1}{|U|} \sum_{u=1} |U| * \frac{1}{m} \sum_{k=1}^N P_u(k) * rel_u(k) \quad (24)$$

Uz ove navedene mjere se koriste i razne druge mjere rangiranja kao što su MRR, rangom pristrana preciznost (engl. rank biased precision - RBP), očekivani recipročni rang (engl. expected reciprocal rank - ERR) i tako dalje [69]. Vrijednosti MAP-a i NDCG-a se kreću od 0 do 1 te veća vrijednost upućuje na bolju evaluaciju rangiranja.

2.5.5 Evaluacijske mjere za performance sustava preporučivanja

Evaluacijske mjere karakteristične za performance sustava preporučivanja su posebne mjere koje služe boljoj evaluaciji performanci sustava preporučivanja. Stvorene su zato što prethodno opisane mjere opisuju samo točnost modela, a usput ne obuhvaćaju širi kontekst u kojem se sustav preporučivanja nalazi [99]. Posebnost mjera evaluacije za performace sustava preporučivanja je uračunavanje u obzir korisničkih potreba što čini korisničko iskustvo boljim [74]. U radu su korištene mjere za različitost, pokrivenost kataloga, novitet i personalizaciju.

Različitost je mjera koja ukazuje na različitost preporuka jedne od drugih. Mala različitost ukazuje na bliskost stvarnih i predviđenih značajki, a s druge strane velika različitost ukazuje na manju povezanost stvarnih i predviđenih značajki. U radu je korištena Jaccardova sličnost (engl. Jaccard similarity) koja je mjera sličnosti između dvaju skupa podataka. Jaccardova sličnost se računa kao veličina presjeka dva skupa podijeljena s veličinom unije dva skupa. Vrijednosti se kreću od 0 do 1, gdje 1 prikazuje veću različitost.

$$JACCARD \cdot RAZLICITOST = \frac{|A \cap B|}{|A \cup B|} \quad (25)$$

Pokrivenost (*engl. coverage*) prikazuje postotak mogućih preporuka koje implementirani algoritam preporučivanja može proizvesti. Pokrivenost je esencijalna iz perspektive novih stvari koje bi se mogle preporučiti te prikazuje brzinu pojavljivanja neke nove stvari na listi preporuka. Pokrivenost se računa kao količnik mogućih stvari koje bi se mogle preporučiti i ukupnog broja stvari (26) [74]. Vrijednosti se kreću od 0 do 1, gdje veća vrijednost prikazuje veću pokrivenost.

$$POKRIVENOST = \frac{MOGUCE \cdot STVARI \cdot ZA \cdot PREPORUKU}{UKUPAN \cdot BROJ \cdot STVARI} \quad (26)$$

Vrsta pokrivenosti koja je obrađena u ovom radu je pokrivenost kataloga. Pokrivenost kataloga se računa kao broj preporučenih stvari podijeljen s ukupnim svih stvari (27). Vrijednosti se kreću od 0 do 1, gdje veća vrijednost prikazuje veću pokrivenost kataloga.

$$POKRIVENOST \cdot KATALOGA = \frac{N(PREPORUCENIH \cdot STVARI)}{N(UKUPAN \cdot BROJ \cdot STVARI)} \quad (27)$$

Novitet je mjera sposobnosti preporučitelja da preporučiti stvari koje nisu relevantne za određenog korisnika i s kojima nije imao interakcije. Najveći se novitet može ostvariti nasumično preporučujući stvari, jer se time najviše smanjuje broj stvari s kojima je korisnik imao interakciju. Unatoč tome što se često računa novitet u evaluaciji, ne postoji jedinstveni način njenog računanja. Novitet je u našem slučaju izračunan kao što je niže prikazano [74, 76]. $Count(i)$ je predstavljen kao zbroj ponavljanja određene stvari u preporukama, U je broj korisnika i N je ukupan broj stvari u skupu podataka (28). Kod noviteta manja vrijednost prikazuje slabiji novitet, dok veća vrijednost prikazuje veći novitet.

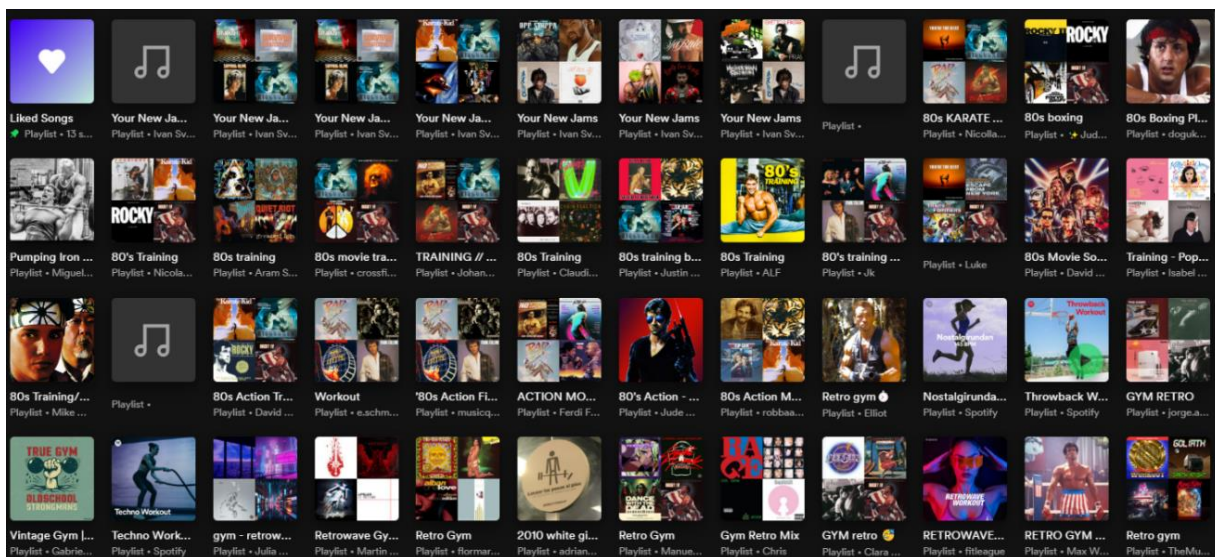
$$NOVOST = \frac{1}{|U|} \sum_{u \in U} \sum_{i \in topN} \frac{\log_2(\frac{count(i)}{|U|})}{|N|} \quad (28)$$

Personalizacija je mjera različitosti između lista preporuka za različite korisnike. Visoki rezultat je indikacija različitosti između preporuka korisnicima, dok mala personalizacija upućuje na sličnost između preporuka različitim korisnicima [76]. Vrijednosti personalizacije se kreću od 0 do 1 te je u radu korištena sličnost kosinusa i rijetka matrica za njeno računanje.

Uz ove navedene mjere, mogu se računati i sretno slučajno otkrivanje, otkrivenost (*engl. discoverability*), neočekivanost (*engl. unexpectedness*) itd.

3. Programska izvedba sustava preporuĉivanja

3.1 Skup podataka za sustav preporuĉivanja



9. Spotify popisi za reproduciranje

Skup podataka korišten za izradu ovog rada se sastoji od preko 10 tisuća pjesama iz 80-ih godina koje ljudi uobičajeno slušaju dok rade tjelesne vježbe. Stvoreno je 150 popisa za reproduciranje koji predstavljaju 150 korisnika te se veličine popisa za reproduciranje kreću od 15 do 200 pjesama po popisu za reproduciranje. Kako je tematika pjesama jako slična, iste se pjesme često nalaze u više popisa za reproduciranje te je ukupan broj jedinstvenih pjesama oko 7 tisuća. Na slici 9 je prikazan jedan dio korištenih popisa za reproduciranje.

Informacije o pjesmama su izvučene iz platforme Spotify preko njihove platforme Spotify for Developers i preko biblioteke Spotipy u Pythonu. Spotify for Developers je platforma koja koristi aplikacijska korisnička suĉelja (*engl. Application programming interface – API*), komplete za razvoj software (*engl. Software development kit - SDK*) i razne druge resurse za izgradnju svoje aplikacije kojom pristupa podacima na platformi Spotify. Neki od podataka koji se koriste za pristup podacima na platformi Spotify su: identifikacija klijenta (*client_id*), tajna identifikacija klijenta (*client_secret*), URI kojim se pristupa Spotify-ju (*redirect_uri*) i prava koja je aplikacija dobila (*scope*). Spotify for Developers ujedno omogućuje prijenos podataka iz platforme Spotify u korisničko suĉelje Pythona preko paketa Spotipy.

Pjesme koje su preuzete s platforme Spotify sadrže različite podatke: o popisu za reproduciranje u kojem je pjesma (id popisa za reproduciranje, ime popisa za reproduciranje te pjesme u popisu za reproduciranje), albumu (ime, jedinstveni id, datum izdavanja, tip glazbe, ime i id izvođača), o izvođaču (jedinstveni id, ime) te zvučnim karakteristikama pjesme (energija, plesnost, tempo, instrumentalnost, popularnost itd). Sve preuzete oznake se mogu vidjeti na slici 10.

```
Index(['id', 'name', 'popularity', 'type', 'is_local', 'explicit',
      'duration_ms', 'disc_number', 'track_number', 'artist_id',
      'artist_name', 'album_artist_id', 'album_artist_name', 'album_id',
      'album_name', 'album_release_date', 'album_tracks', 'album_type',
      'playlist_id', 'playlist_name', 'playlist_tracks', 'added_at',
      'added_by', 'genres', 'album_genres', 'danceability', 'energy', 'key',
      'loudness', 'mode', 'speechiness', 'acousticness', 'instrumentalness',
      'liveness', 'valence', 'tempo', 'uri', 'track_href', 'analysis_url',
      'time_signature', 'isrc', 'genres_str'],
      dtype='object')
```

10. Podaci o pjesmama

Nakon dobave se podaci spremaju u *.pkl* datoteku koja će se koristiti za prosljeđivanje na daljnju obradu i korištenje tih informacija.

3.2 Korištene tehnologije i alati

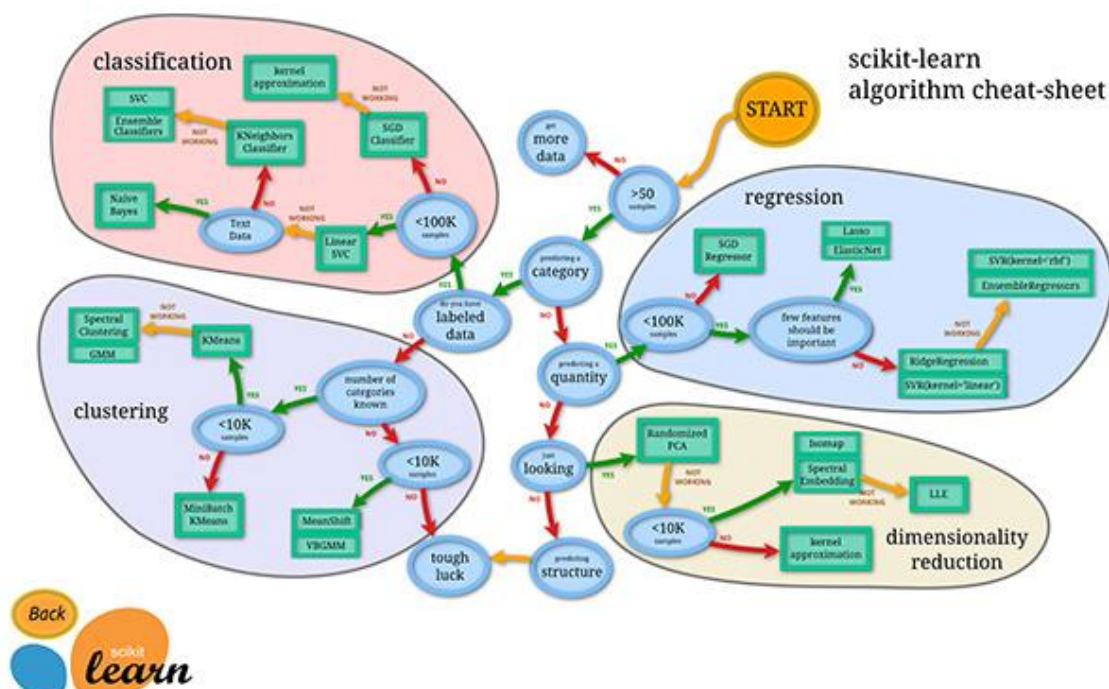
- Spotify
- Programi
 - Python
 - Spotipy
 - Pandas
 - Jupyter Notebook
 - Sklearn
 - Surprise
 - Numpy
 - Pandas
 - Recommenders
 - Math
 - Typing
 - Tqdm
 - Recmetrics

U ovom je radu za programiranje sustava preporučivanja i njihovu evaluaciju korišten program Python preko platforme Jupyter Notebook. Jupyter je Pythonov projekt koji obuhvaća različite softverske alate poput Jupyter Notebooka i JupyterLaba. Jupyter Notebook je aplikacija koja se bazira na formatu komputacijske bilježnice (*engl. computational notebook format*) te omogućuje programiranje, pretraživanje i vizualizaciju. Prednosti Jupytera su u tome što može pokrenuti kod iz preglednika, omogućuje dobar i jasan prikaz rezultata te ima mogućnosti matematičkih zapisa. Dokumenti spremljeni u Jupyter Notebooku su zapravo dokumenti tipa JSON, ali su spremljeni s ekstenzijom *.ipynb* [91]. Posebno je koristan u zadacima gdje postoji više kompleksnih algoritama, zadaća i grafova u istom programskom kodu.

Jedan od paketa koji je korišten u izradi ovog rada je scikit-learn ili sklearn koji je ujedno i jedna od najkorištenijih biblioteka u Pythonu za strojno učenje. Ona sadrži brojne efikasne alate za strojno učenje i statističko modeliranje uključujući klasifikaciju, regresiju, klasteriranje, redukciju dimenzionalnosti, unakrsnu validaciju te algoritme nenadziranog učenja. Uz to, sadrži razne ugrađene skupove podataka te može služiti za ekstrakciju značajki iz slika ili teksta [92]. Paket podržava razne modele strojnog učenja poput logističke regresije, XGBoosta, stabla odlučivanja, klasifikatora K susjeda i tako dalje. Osim toga, pruža razne metode evaluacije poput *cross_val_score*, *pipeline-a* i ostale metode unutar sklearn.metrics paketa. Podrobnija vizualna podjela scikit-learn-a je izražena na slici 11.

U radu je osim scikit-learn-a korištena biblioteka surprise koja služi za izgradnju i analizu sustava preporučivanja i za eksplicitno ocjenjivanje unutar Pythona. Biblioteka surprise u sebi sadrži ugrađene algoritme poput SVD-a, KNN-a, NMF-a, KNN-a te pruža razne alate za evaluaciju, analizu i usporedbu performanci algoritama. Posebno se ovo izražava u *cross_validate* metodi za unakrsnu validaciju [93].

Što se ostalih paketa tiče: Recmetrics je paket za ugrađene mjere evaluacije, iz paketa recommenders se koristi metoda za podjelu skupa podataka, math je matematički paket, tqdm je paket za progresne linije, numpy, typing i pandas su standardni podatkovni paketi za Python.



11. Scikit-learn biblioteka [111]

3.3 Metodologija izrade preporuka

U ovom je radu korišteno 6 različitih metoda preporučivanja: TF-IDF preko sličnosti kosinusa, KNN preko tehnologije surprise, SVD preko tehnologije surprise, stablo odlučivanja tehnologijom sklearn, logistička regresija tehnologijom sklearn te klasifikator K najbližih susjeda tehnologijom sklearn. Nakon što su se proveli algoritmi preporučivanja, ocjenjeni su raznim mjerama evaluacije algoritma. Mjere evaluacije koje su provedene su: preciznost, odziv, mjera F1, RMSE, MAE, NDCG i MAP. Ove sve navedene mjere evaluacije su provedene metodom train-test i metodom unakrsne validacije. Osim ovih navedenih, provedene su metode evaluacije karakteristične za sustave preporučivanja među kojima su personalizacija, različitost, pokrivenost kataloga i novitet.

Razlog odabira baš ovih metoda je prikazivanje algoritama iz iste kategorije korištenjem strojnog učenja i bez korištenja strojnog učenja. KNN i klasifikator K-najbližih susjeda su dio suradničkog filtriranja koje se temelji na susjedstvu, suradničko filtriranje temeljeno na modelu je predstavljeno preko SVD-a i stabla odlučivanja, dok je filtriranje temeljeno na sadržaju prikazano preko TF-IDF-a i logističke regresije [90].

Za metodu TF-IDF je korišten objekt *TfidfVectorizer* iz biblioteke scikit-learn koji pretvara kolekciju dokumenata u matricu s TF-IDF karakteristikama nad kojima se koristi analizator koji analizira ključne riječi [95]. Ključne se riječi sastoje od imena pjesme, imena umjetnika, imena albuma, imena popisa za reproduciranje te imena žanrova kojima pjesma pripada. Potom se ova matrica uči na spomenutim podacima da bi se dobila što veća moć predviđanja nakon čega se elementi matrice uspoređuju preko metode *linear_kernel* mjereći kosinusnu sličnost između različitih stvari. Potom se ovi podaci spremaju i sortiraju u listu preporuka.

Metoda SVD prvo stvara *Dataframe* u kojem se u paru prikazuju korisnici, stvari i njihove ocjene. Korištena je metoda *SVD()* iz biblioteke surprise koja daje preporuke na temelju unesenog *Dataframea*. Potom se rezultati dobiveni korištenjem metode *SVD()* sortiraju te time nastaju preporuke za korisnike. Za metodu KNN se izvodi isti postupak, osim što se u njegovom slučaju koristi metoda *KNNBasic()* (osnovni KNN algoritam).

Osim prethodno navedenih metoda preporučivanja, u radu su korištene razne druge ugrađene metode strojnog učenja u svrhu preporučivanja. U tim su se metodama prvo podaci podijelili na podatke za učenje i ispitivanje nakon čega su podaci za učenje preneseni na model. Potom je stvoren *Pipeline* na koji se stavila oznaka modela koji je skalirao ulazne podatke. *Pipeline* služi orkestriranju operacija strojnog učenja tako što se linearni niz podataka veže skupa u jedan modelirani proces [94]. Modeli koji su korišteni su isto dio scikit-learn biblioteke: *LogisticRegression*, *DecisionTreeClassifier* i *KNeighborsClassifier*. Nakon toga su preporuke dobivene iz prikupljenih modela spremljene i sortirane.

3.4 Metodologija provedbe evaluacija

Nakon preporučivanja provela se evaluacija rezultata algoritama preporučivanja. Evaluacija je provedena za unakrsnu validaciju, train-test metodu i za metode evaluacije karakteristične za sustave preporučivanja.

Što se unakrsne validacije tiče, prvo se provela preko metoda iz paketa sklearn.metrics. U ovoj se metodi prvo podjelio skup podataka u 10 dijelova preko metode *StratifiedKfold* ili *KFold*. *StratifiedKFold* je metoda koja se koristi za nebalansirane skupove podataka kao što je u ovom radu, dok *KFold* to isto ne uzima u obzir. *StratifiedKFold* je proveden kod svih metoda preporučivanja, osim KNN i SVD iz paketa surprise. U njihovom je slučaju korišten *KFold*, jer

se nije mogla tehnički izvesti podjela preko *StratifiedKfold-a*. Nakon podjele se prvo uče podaci za učenje na modelu i onda se na tom istom modelu sa skupom podataka za ispitivanje provodi proces preporučivanja. Nakon što se dobiju rezultati predviđanja, provodi se ispitivanje mjera evaluacije sustava preporučivanja nad predviđenim i stvarnim podacima o slušanosti pojedinih pjesama.

Mjere se evaluacije provode preko metoda koje se nalaze u paketu `sklearn.metrics` i uključuju

- *mean_squared_error* (RMSE)
- *mean_absolute_error* (MAE)
- *ndcg_score* (NDCG)
- *average_precision_score* (MAP)
- *precision_score* (preciznost)
- *recall_score* (odziv)

Postupak se provodi za svih 149 korisnika te se nakon izvođenja petlje računa prosjek vrijednosti pojedine mjere evaluacije. Također se računa rezultat za mjeru F1, koji se dobije s već spomenutom jednadžbom (17) iz vrijednosti preciznosti i odziva. Uz ovaj navedeni način unakrsne validacije provode se u radu i drugi načini unakrsne validacije.

Za metode strojnog učenja provodi se metoda *cross_val_score* iz paketa `sklearn.model_selection`. Kao ulazni parametri se unose: naučeni objekt koji predstavlja odabranu metodu strojnog učenja (klasifikator K-najbližih susjeda, logistička regresija i stablo odlučivanja), razne brojčane vrijednosti koje opisuju pjesmu i prema kojima se radi predviđanje (energija, plesnost, tempo, instrumentalnost, popularnost itd.), stvarni podaci o slušanosti pjesme od strane korisnika, na koliko se dijelova dijeli skup podataka te koji je način provođeno ocjenjivanje. Rezultati su generirani za preciznost, odziv, F1, MAE, RMSE, NDCG i MAP. Prednosti ove metode naspram drugih su uvid u performance modela, efikasnost podataka i efektivna evaluacija.

Osim *cross_val_score* korištena je i *cross_validate* metoda iz paketa `surprise.model_selection`. Ova je metoda implementirana za računanje rezultata unakrsne validacije u algoritmima preporučivanja iz biblioteke `surprise` poput KNN-a i SVD-a. Ova se metoda može upotrijebiti samo za računanje vrijednosti unakrsne validacije za RMSE i MAE. Ulazni su parametri: oznaka algoritma, podaci o slušanosti pjesme, oznaka mjere te broj podjela.

Osim unakrsne validacije, provedena je evaluacija preko metode train-test. Podaci su podijeljeni u omjeru 80/20, gdje je 80% podataka primijenjeno je za učenje, dok je za ispitivanje primijenjeno 20% podataka. Nakon što se model nauči, ispitni se podaci ispituju na već spomenutim metodama iz `sklearn.metrics`. Jedino u metodi TF-IDF nije provedeno učenje modela, već je provedeno ispitivanje mjera nad preporučenim podacima.

Također su provedene mjere evaluacije koje su karakteristične za sustave preporučivanja: novitet, pokrivenost kataloga, različitost i personalizacija. Primijenjeni načini računanja ovih vrijednosti su isti onima koji se primijenjuju u biblioteci `Recmetrics`. Za računanje ovih mjera je bilo potrebno izračunati vrijednosti prosječne stote stvari koja je preporučena, lista prvih 100 preporučenih stvari te broj ponavljanja za svaku od stvari u preporukama. Prvih 100 vrijednosti ujedno prikazuje prvih 100 elemenata koji će se korisniku prikazati (TOP-N). Stoga je uzeta vrijednost stotog elementa kao odlučujući faktor kod evaluacije klasifikacije.

4. Rezultati

U sljedećim se poglavljima prikazuju, objašnjavaju i uspoređuju rezultati evaluacije različitih metoda sustava preporučivanja. Prvo su objašnjeni kriteriji klasifikacije ocjenjivanja rezultata evaluacije i klasifikacija kvalitete evaluacije za svaku mjeru evaluacije. Nakon toga su u tablicama prikazani rezultati za train-test metodu, metodu unakrsne validacije i za metode evaluacije karakteristične za sustave preporučivanja.

Rezultati su prvo prikazani i objašnjeni iz ugla izbora pojedine metode evaluacije. Objašnjen je rezultat preporuke, nakon čega se taj isti rezultat klasificirao po različitim mjerilima. Nakon klasifikacije, uslijedila je usporedba s rezultatima dobivenim drugim metodama preporučivanja. Na kraju su ponuđena objašnjenja na što bi mogao upućivati takav rezultat.

U idućem poglavlju su isti registrirani rezultati prikazani iz ugla pojedine metode preporučivanja. Rezultati su objašnjeni, klasificirani po kvaliteti te su dani razlozi zbog kojih bi rezultati mogli imati takve vrijednosti. Na kraju poglavlja je opisan odabir najbolje metode preporuke za svaku metodu evaluacije, što to znači za pojedini sustav te je naposljetku zaključen odabir najbolje metode koja bi se koristila za rješavanje problematike ovog rada.

Rezultati dobiveni evaluacijom regresije i klasifikacije u različitim metodama preporučivanja su sličnih vrijednosti te su blisko kategorizirani. S druge strane kod mjera evaluacije rangiranja i mjera evaluacije karakterističnih za sustave preporučivanja rezultati variraju od metode do metode. No o tome više u nastavku rada.

4.1 Klasifikacija kvalitete preporuka

Prilikom promatranja rezultata evaluacije uvijek se gleda rezultat po pojedinoj metodi preporučivanja te se potom taj rezultat klasificira kao dobar, loš ili odličan, jako loš.

Postoje različite vrijednosti po kojima se vrijednost RMSE smatra jako dobrom. Prema nekim izvorima ta je vrijednost od 0 do 0.75, po drugim izvorima je to od 0 do 0.7 ili čak od 0 do 0.5. U ovom je radu primijenjen stroži kriterij (0-0.5) da bi bili sigurniji u klasifikaciju vrijednosti RMSE. S druge strane, smatrat ćemo vrijednosti RMSE veće od 0.5 kao lošije vrijednosti [96]. Kod metode MAE, dobrim se rezultatom smatra vrijednost koja je manja od dvije trećine vrijednosti RMSE. U ovom radu bi ta vrijednost bila $1/3$.

Nasuprot tome kod evaluacije rangiranja, tj NDCG-a i MAP-a, se vrijednosti evaluacije kreću od 0 do 1. Što je vrijednost NDCG-a ili MAP-a veća, to se ona smatra boljom u smislu kvalitete rangiranja. Vrijednosti NDCG-a ili MAP-a koje su između 0.8 i 1 se smatraju odličnim vrijednostima, vrijednosti između 0.6 i 0.8 se smatraju prihvatljivim ili dobrim vrijednostima, dok se vrijednosti između 0 i 0.6 smatraju lošim vrijednostima [97].

Kod evaluacije klasifikacije u radu (preciznost, odziv i F1 vrijednost), vrijednosti se također prostiru od 0 do 1 te 1 predstavlja najbolju moguću vrijednost dok 0 predstavlja najgoru moguću vrijednost. Vrijednosti od 0.8 do 1 se smatraju odličnom vrijednosti, vrijednosti od 0.7 do 0.8 se smatraju dosta dobrim, vrijednosti između 0.6 i 0.7 se smatraju dobrim vrijednostima, vrijednosti između 0.5 i 0.6 se smatraju lošim, ali donekle prihvatljivim vrijednostima, dok se vrijednosti između 0 i 0.5 smatraju lošim vrijednostima [98]. Sustavi koji imaju veću preciznost od odziva se preferiraju koristiti u svrhama gdje je jako bitno da su podaci točni (medicina). U sustavima gdje je važnije prikupiti što više podataka koji se odzivaju, preferira se veći odziv od preciznosti. Takav je sustav poželjan i u ovom radu.

Što se tiče mjera za evaluaciju karakterističnih za sustave preporučivanja, što je manja njihova vrijednost to je rezultat lošiji. Isto tako vrijedi i suprotno, što je veći rezultat on se smatra boljim. Kod različitosti, personalizacije i pokrivenosti kataloga se vrijednosti prostiru od 0 do 1, dok se kod noviteta moguće vrijednosti prostiru od 0 do beskonačnosti.

4.2 Opis rezultata prema izboru mjere evaluacije

UNAKRSNA VALIDACIJA							
		TF-IDF	SVD SURPRISE	KNN SURPRISE	LOGISTIČKA REGRESIJA	STABLO ODLUČIVANJA	KLASIFIKATOR K SUSJEDA
Sklearn.metrics	RMSE		0.1107	0.0989	0.0989	0.0757	0.0825
	MAE		0.0792	0.02	0.013	0.0139	0.0131
	NDCG		0.0	0.6205	0.6182	0.2388	0.4806
	MAP		NAN	0.0386	0.0371	0.0259	0.2184
	PRECIZ NOST	micro	0.8106	0.97	0.9744	0.9868	0.9908
		macro	0.45	0.5323	0.5335	0.5074	0.6462
		weighted	0.9	0.9825	0.9822	0.9873	0.9918
	ODZIV	micro	0.8106	0.97	0.9744	0.9868	0.9908
		macro	0.4053	0.577	0.5609	0.5144	0.6638
		weighted	0.8106	0.97	0.9744	0.9868	0.9908
	F1	micro	0.8106	0.97	0.9744	0.9868	0.9908
		macro	0.4265	0.5541	0.5467	0.5109	0.6549
		weighted	0.853	0.976	0.9783	0.987	0.9913
Cross val score	RMSE					0.0757	0.095
	MAE					0.0067	0.0107
	NDCG					0.2092	0.2423
	MAP					0.0071	0.024
	PRECIZNOST					0.5051	0.5292
	ODZIV					0.5076	0.5262
	F1					0.5064	0.5277
Cross_validate	RMSE			0.0987	0.0989		
	MAE			0.02	0.013		

TRAIN-TEST								
		TF-IDF	SVD	KNN	LOGISTIČKA	STABLO	KLASIFIKATOR	
			SURPRISE	SURPRISE	REGRESIJA	ODLUČIVANJA	K SUSJEDA	
Sklearn.metrics	RMSE		0.1904	0.0991	0.0994	0.0754	0.0998	0.082
	MAE		0.1616	0.0201	0.0134	0.0139	0.0136	0.0131
	NDCG		0.0	0.6467	0.635	0.2667	0.2729	0.2558
	MAP		NAN	0.0378	0.0327	0.0227	0.0206	0.011
	PRECIZ	micro	0.9821	0.9699	0.9726	0.9861	0.9848	0.9622
		macro	0.5201	0.5314	0.5307	0.5055	0.5182	0.5028
	NOST	weighted	1.0	0.9823	0.982	0.9872	0.9878	0.987
	ODZIV	micro	0.9821	0.9699	0.9726	0.9861	0.9848	0.9622
		macro	0.5112	0.5726	0.5615	0.5111	0.5251	0.5109
		weighted	0.9821	0.9699	0.9726	0.9861	0.9848	0.9622
	F1	micro	0.9821	0.9699	0.9726	0.9861	0.9848	0.9622
		macro	0.5156	0.5513	0.5457	0.5083	0.5228	0.5068
		weighted	0.991	0.976	0.9773	0.9867	0.9863	0.9745

4.2.1. RMSE

Procesom unakrsne validacije dobiveni su rezultati evaluacije svih spomenutih mjera evaluacije. Što se tiče vrijednost RMSE u unakrsnoj validaciji za sklearn.metrics, može se primijetiti da je najmanja na stablu odlučivanja i logističkoj regresiji, a da svi ostali algoritmi preporučivanja imaju za nešto manje od 0.05 veće vrijednosti RMSE od ove dvije prije spomenute. Najveću vrijednost i najlošiji RMSE ima metoda TF-IDF. Po već spomenutim mjerilima kvalitete preporuke svaka se evaluacija RMSE u radu smatra dobrom, dok ostale primijenjene metode evaluacije to potvrđuju.

Kod TF-IDF-a su najveće razlike među rezultatima dobivenim različitim metodama. Rezultat dobiven unakrsnom validacijom u metodi sklearn.metrics je za 0.08 manji od rezultata dobivenog train-test metodom. Kod SVD-a i KNN-a se rezultati razlikuju između različitih metoda u rasponu od 0.0005. Kod logističke regresije su rezultati u rasponu od 0.0003 jedni od drugih, kod stabla odlučivanja u rasponu od 0.025 jedan od drugog, a kod klasifikatora K susjeda u rasponu od 0.01 jedni od drugih.

Prema ovim rezultatima se može zaključiti da su rezultati jako dobro naučeni na modelu. RMSE je poželjan za korištenje ukoliko u predviđanju nema većih grešaka, jer utječe na dvije njegove velike mane. Dvije mane RMSE-a su nemogućnost analize smjera greške i osjetljivost na velike greške (*engl. long tail problem*). Za nadomještanje ovih spomenutih nedostataka se koristi metoda MAE.

4.2.2 MAE

Kao što je napisano prije u radu, što je vrijednost MAE bliža nuli to je njena vrijednost bolja. U unakrsnoj validaciji za `sklearn.metrics` metodi najbolji se rezultati za MAE ostvaruju za stablo odlučivanja, KNN, klasifikator K susjeda, logističku regresiju te SVD, a TF-IDF ima najlošiji zabilježen rezultat. Svi se od ovih zabilježenih rezultata smatraju jako dobrim.

Kod MAE su također zabilježene razlike u rezultatima između korištenih metoda preporučivanja. Rezultat TF-IDF-a u unakrsnoj validaciji za `sklearn.metrics` je za 0.09 manji od rezultata zabilježenog u train-test metodi evaluacije. Razlika za SVD je puno manja (0.0001) između različitih korištenih metoda, za KNN je 0.0004, a za logističku regresiju, stablo odlučivanja i klasifikator K susjeda je razlika otprilike 0.01.

Može se uz to primijetiti kako MAE ima nešto manje vrijednosti od RMSE-a, a razlog tome je korijenovanje i kvadriranje koje se primijenjuje u RMSE-u. Osim toga, RMSE daleko više kažnjava greške velikih dimenzija od MAE, dok je MAE interpretabilniji jer prikazuje prosjek apsolutne pogreške.

4.2.3. NDCG

Kod NDCG-a, veća vrijednost prikazuje bolje performanse u evaluaciji rangiranja stvari. Po unakrsnoj validaciji za `sklearn.metrics` najveći su rezultati generirani za metode SVD i KNN te se njihovi rezultati smatraju prihvatljivima. Loše su vrijednosti zabilježene za stablo odlučivanja, logističku regresiju i klasifikator K susjeda. TF-IDF ima lošu vrijednost 0.

U ostalim su metodama zabilježeni slični rezultati. Što se tiče TF-IDF-a, kod njega je zabilježena vrijednost 0 i u train-test metodi. Kod SVD-a se rezultati razlikuju između metoda za otprilike 0.025, kod KNN-a za 0.02, kod logističke regresije su u rasponu 0.06, a kod klasifikatora K susjeda su u rasponu od 0.04. Kod stabla odlučivanja je najveći raspon u rezultatima, gdje se za

unakrsnu validaciju putem `sklearn.metrics`-a bilježi rezultat 0.4806, kod train-testa rezultat 0.2729, dok se u `cross_val_score`-u bilježi rezultat 0.2423.

NDCG je mjera koja prikazuje relevantnost rezultata prikazujući koliko će se relevantnih rezultata pojaviti u TOP-N rezultata. Osim ove mjere evaluacije rangiranja, provedeno je i računanje MAP-a. Razlika između ove dvije metode jest ista razlici predviđanja od 0-1 između regresije i klasifikacije.

4.2.4. Srednja prosječna preciznost (MAP)

U slučaju MAP-a također su najbolje vrijednosti bliže 1, dok su vrijednosti bliže nuli najlošije. Svi rezultati koji su izgenerirani za MAP u ovom radu mogu se smatrati lošima, dok se za TF-IDF uopće nije uspio izgenerirati rezultat.

Za metodu evaluacije SVD su vrijednosti dobivene različitim metodama unutar 0.001 jedna od druge, kod KNN to je 0.005, kod logističke regresije 0.003, kod stabla odlučivanja je unutar 0.2, dok je to kod klasifikatora K susjeda 0.0013.

MAP je mjera koja također prikazuje relevantnost preporuka, što poboljšava korisničko zadovoljstvo. MAP upućuje na konzistentnost ili točnost rangiranja te se koristi za ocjenjivanje kvalitete rangiranih listi. Kao što se može primijetiti, rezultati za NDCG su veći od rezultata za MAP. Razlog tome je što NDCG daje veću važnost podacima koji se nalaze na vrhu liste, dok MAP gleda na sve podatke jednako. Kako je u radu korišten nebalansirani skup podataka, ovo se ogleda i u rezultatima evaluacije rangiranja.

4.2.5. Preciznost

Što se evaluacije klasifikacije tiče, postoji gradacija i razlika u kvaliteti rezultata. Rezultati se kreću od nula do 1 te što je vrijednost veća to je rezultat preciznosti, odziva ili F1 bolji. Što se ovog rada tiče, najbolje je koristiti vrijednosti za makro rezultate, jer oni najbolje mogu prikazati rezultate kod nebalansiranih skupova podataka.

Unakrsnom validacijom preko `sklearn.metrics`-a najbolji je rezultat preciznosti dobiven za stablo odlučivanja te se klasificira kao dobra vrijednost. Za metode SVD, KNN, logističku regresiju i klasifikator K susjeda dobiveni su rezultati koji se klasificiraju kao loši, ali prihvatljiv rezultati. Dok je za TF-IDF dobiven rezultat koji se klasificira kao loš rezultat.

Što se tiče metode TF-IDF, za train-test je rezultat veći od unakrsne validacije preko `sklearn.metrics`-a za 0.08 te se taj rezultat klasificira kao loš, ali prihvatljiv. Kod SVD-a su rezultati bliski (unutar 0.001 jedan od drugog), kod KNN-a unutar 0.003 jedan od drugog, kod logističke regresije unutar 0.002, kod klasifikatora K susjeda unutar 0.001 jedan od drugog. Kod stabla odlučivanja se ističe rezultat za unakrsnu validaciju preko `sklearn.metrics`-a koji je za 0.12/0.13 veći od rezultata za train-test i *cross_val_score*.

Preciznost je mjera kojom se provjerava točnost predviđanja te se promatra koliko je predviđanja točno pozitivno predviđeno. Pogotovo je važna za minimizaciju lažnih predviđanja i specifičnost rezultata. Preciznost je posebno važna u slučajevima gdje lažne pozitivne vrijednosti mogu dovesti do većih posljedica (medicina).

4.2.6 Odziv

Što se odziva tiče, u unakrsnoj validaciji metodom `sklearn.metrics` najbolji su rezultati zabilježeni za stablo odlučivanja te se oni klasificiraju kao prihvatljivi. Rezultati za klasifikator K susjeda, SVD, KNN i logističku regresiju se mogu klasificirati kao loši, ali prihvatljivi. Dok se rezultati za TF-IDF klasificiraju kao loši.

Što se tiče TF-IDF-a, rezultat u metodi train-test je 0.5112 što ga čini za 0.11 većim od rezultata u metodi unakrsne validacije `sklearn.metrics`. Kod SVD-a razlika između dvaju rezultata je 0.005, kod KNN-a je ta razlika manja od 0.001, kod logističke regresije ta je razlika 0.005, kod klasifikatora K susjeda je ta razlika 0.003. U stablu se odlučivanja opet ističe unakrsna validacija metodom `sklearn.metrics` koja ima vrijednosti veće za 0.14 od *cross_val_score*-a i train-test metode.

Odziv je bitan u slučajevima gdje je model osjetljiv na identifikaciju pozitivnih instanci kao što je i u našem modelu te se pritom posebno ističe odnos preciznosti i odziva. Kako je u ovom modelu važniji odziv, onda bi trebalo usporediti odziv i preciznost. Kod metode TF-IDF je u svim metodama zabilježena veća preciznost od odziva, kod SVD-a, KNN-a, logističke regresije i klasifikatora K susjeda je zabilježen veći odziv u svim metodama, dok je kod stabla odlučivanja u unakrsnoj validaciji `sklearn.metrics` i train-test metodi zabilježen veći odziv, a u unakrsnoj validaciji *cross_val_score* je zabilježena veća preciznost (za 0.003).

4.2.7 F1

U unakrsnoj validaciji sklearn.metrics je najbolji rezultat zabilježen za stablo odlučivanja (prihvatljiv). Loši, ali prihvatljivi su rezultati zabilježeni za SVD, KNN, logističku regresiju i klasifikator K susjeda. Dok je loš rezultat zabilježen za TF-IDF.

Kod train-test metode za TF-IDF je opet zabilježen rezultat veći za 0.09 od rezultata u unakrsnoj validaciji metodom sklearn.metrics. Razlika između različitih rezultata za SVD je 0.003, za KNN 0.001, za logističku regresiju je 0.004, za klasifikator K susjeda je to 0.0008. Kod metode stabla odlučivanja se opet ponovio slučaj kao kod odziva i preciznosti gdje rezultat za unakrsnu validaciju metodom sklearn.metrics odskoče za 0.13 od ostala dva.

4.2.8 Evaluacijske mjere za performanse sustava preporučivanja

EVALUACIJA SUSTAVA PREPORUČIVANJA						
	TF-IDF	SVD SURPRISE	KNN SURPRISE	LOGISTIČKA REGRESIJA	STABLO ODLUČIVANJA	KLASIFIKATOR K SUSJEDA
NOVITET	0.0931	6.1318	6.8383	5.1032	5.9659	6.0772
POKRIVENOST KATALOGA	0.01	0.01	0.07	0.48	0.85	0.8
PERSONALIZACIJA	0.0551	0.0433	0.2329	0.9692	0.9806	0.9895
RAZLIČITOST	0.0152	0.0615	0.0134	0.0099	0.0039	0.0246

Najveći je novitet zabilježen kod KNN-a, kojeg prati SVD, klasifikator K susjeda, stablo odlučivanja te potom logistička regresija. Daleko manja vrijednost od ovih svih je zabilježena za TF-IDF. Toliko mala vrijednost prikazuje činjenicu da algoritam TF-IDF nije podoban u preporučivanju novih stvari i proizvoda. S druge strane ostali algoritmi su podobniji u toj svrhi i bolje bi poslužili u pretraživanju stvari koje se ne nalaze u stvarima s kojima je korisnik imao interakcije.

Pokrivenost kataloga se kreće između 0 i 1 te 1 prikazuje najveću pokrivenost kataloga stvari. Po rezultatima u tablici, najveća je pokrivenost kataloga zabilježena za stablo odlučivanja. Slijede ga klasifikator K susjeda i logistička regresija koji su imali vrijednosti 0.8 i 0.48 respektabilno. Najmanje su vrijednosti zabilježene za TF-IDF, SVD i KNN. Pokrivenost kataloga prikazuje koliko je stvari obuhvaćeno iz cijelog skupa stvari.

Što se tiče personalizacije, primijećeno je kako je personalizacija znatno veća za algoritme strojnog učenja (0.9-1) negoli za ostale algoritme. Kod SVD-a i TF-IDF-a su zabilježene male vrijednosti personalizacije, dok KNN ima vrijednost personalizacije 0.2329.

Za različitost je primijećeno kako sve metode preporučivanja imaju zabilježene jako male vrijednosti različitosti. Najveća je zabilježena za SVD i klasifikator K susjeda.

4.3 Opis rezultata po metodi preporučivanja

4.3.1 TF-IDF

Za metodu TF-IDF je zabilježena jako dobra evaluacija regresije što ukazuje na dobro preslikavanje modela TF-IDF na predviđanja. Može se također primijetiti da je TF-IDF manje osjetljiv na outliere od ostalih metoda. Razlog tome je u tome što TF-IDF ima najmanju razliku između RMSE i MAE što predstavlja uravnoteženiji raspored predviđanja i uravnoteženiji skup podataka. Za razliku od ostalih korištenih metoda TF-IDF preporučuje stvari isključivo na bazi izraza koji su korišteni bez uzimanja konteksta u obzir. Ovo pritom smanjuje broj outliera. S druge strane evaluacija regresije je najveća za TF-IDF naspram ostalih korištenih metoda preporuke. Ovome je vjerojatno razlog činjenica što korišteni TF-IDF nije dio ugrađenog modela iz postojeće biblioteke za razliku od ostalih korištenih sustava preporučivanja.

Sljedeće je provedena evaluacija rangiranja, gdje su zabilježeni rezultati 0.0 za NDCG i NAN za MAP. Razlog tome je činjenica da TF-IDF uzima isključivo u obzir težinu izraza bez obzira na kontekst. Kako se ne uzima kontekst u obzir, onda se ni ne može dobro preslikati preporučivanje (s obzirom na rangiranje i relevantnost) na TF-IDF-u.

Što se evaluacija klasifikacije tiče, u train-testu su zabilježene loše vrijednosti, dok je u unakrsnoj validacije preko sklearn.metrics-a zabilježen loš, ali prihvatljiv rezultat. Također se može primijetiti kako je preciznost veća od odziva, što je važno za modele gdje je ključna točnost predviđanja te gdje lažni pozitivni mogu imati ozbiljne posljedice (medicina). U ovom se radu s druge strane preferira veća vrijednost odziva od preciznosti.

Kod mjera evaluacije karakterističnih za sustave preporučivanja su zabilježene male vrijednosti svih mjera. Stoga TF-IDF nije metoda kojom bi se mogli pronaći novi sadržaji (novitet), kojom ne bi bio pokriven veći broj stvari (pokrivenost kataloga), koji ima manju sposobnost preporučivati po mjeri osobe (personalizacija) te preporuke nisu diverzificirane (različitost).

4.3.2 SVD

Što se metode SVD tiče, za evaluaciju regresije su zabilježeni jako dobri rezultati što ukazuje na dobro preslikavanje modela SVD na predviđanja. SVD je osjetljiviji na outliere nego TF-IDF, ali je vrijednost RMSE i MAE manja nego kod TF-IDF-a što upućuje na raznolikost predviđanja. Kod evaluacije rangiranja dobri su rezultati zabilježeni za NDCG, a dosta lošiji su rezultati zabilježeni za MAP. Razlog toga je nebalansiranost skupa podataka što utječe na dobre performanse za predviđanje TOP-N sadržaja (NDCG) i lošije performanse rangiranja na ostatku skupa podataka (MAP). SVD uz KNN ima najbolje performanse rangiranja naspram ostalih sustava te ga čini jako dobrim u rangiranju elemenata.

Kod evaluacije klasifikacije su zabilježeni mahom loši, ali prihvatljivi rezultati. Uz to je u svakoj od metoda evaluacije klasifikacije zabilježen veći odziv nego preciznost što je jako dobro za ovaj preporučivanje u ovom radu. Kod mjera evaluacije sustava preporučivanja zabilježena je dobra sposobnost pronalaska novog sadržaja (novitet), mala pokrivenost kataloga, manja sposobnost preporučivanja po mjeri korisnika (personalizacija) i mala diverzificiranost sadržaja (različitost).

4.3.3 KNN

U evaluaciji regresije za KNN su zabilježeni slični rezultati kao u SVD-u te se njegovi rezultati mogu klasificirati kao dobri. Zabilježen je nešto veći broj outliera nego u metodi SVD zbog manje vrijednosti MAE, dok istovremeno imaju isti RMSE. U evaluaciji rangiranja je zabilježen dobar rezultat za NDCG i jako loš rezultat za MAP, a razlog je nebalansirani skup podataka kao i kod SVD-a. Uz SVD ima najbolje ocjene rangiranja od svih korištenih algoritama.

Evaluacijom klasifikacije su zabilježeni loši, ali prihvatljivi rezultati te je odziv veći od preciznosti što se smatra dobrim za ovu vrstu preporuke. Kod mjera evaluacije karakterističnim za sustave preporučivanja za KNN je zabilježena najveća sposobnost pronalaska novog sadržaja (novitet), slabija pokrivenost kataloga, manja sposobnost pronalaska sadržaja po mjeri korisnika (personalizacija) te dosta mala različitost. Zabilježena personalizacija ima vrijednost 0.2329, što je dosta više od SVD-a i TF-IDF-a, ali dosta manje od modela koji će bit u idućim poglavljima opisani.

4.3.4 Logistička regresija

Kod logističke regresije su također zabilježene dobre vrijednosti evaluacije regresije, čak bolje nego kod metoda preporučivanja koje su prethodno opisane. Zabilježen je najmanji RMSE od svih mjera evaluacije, što upućuje na manji broj outliera. Osim toga, zabilježen je najmanji MAE u metodi *cross_val_score*. U evaluaciji rangiranja su zabilježeni loši rezultati za NDCG i MAP, što upućuje na lošu sposobnost preporuke relevantnosti preporučenih podataka. Također je zabilježen veći rezultat NDCG naspram MAP zbog nebalansiranog skupa podataka. Za mjere klasifikacije su zabilježeni mahom loši, ali prihvatljivi rezultati te je odziv veći od preciznosti. U mjerama evaluacije za sustave preporučivanja je zabilježena velika sposobnost pronalaska novog sadržaja (novitet), pokrivenost kataloga srednje veličine (0.48), jako velika sposobnost pronalaska sadržaja po mjeri korisnika (personalizacija) te mala različitost.

4.3.5 Stablo odlučivanja

Za stablo odlučivanja i njegovu evaluaciju regresiju su zabilježeni jako dobri rezultati te treba primijetiti kako je u ovoj metodi zabilježen najveći broj outliera. Razlog tome je najveća razlika između RMSE i MAE među svim korištenim metodama preporučivanja. Unatoč tome, ovaj model jako dobro preslikava predviđanja na model. Što se evaluacije rangiranja tiče, zabilježene su loše vrijednosti te se nebalansirani skup podataka ogleda na velikoj razlici između NDCG i MAP-a. Ono što je zanimljivo kod stabla odlučivanja u ovom radu su znatno veće vrijednosti rezultata zabilježene metodom unakrsne validacije *sklearn.metrics* nego u drugim mjerama evaluacije. Posebno se ovo ističe kod NDCG-a i MAP-a, gdje su rezultati veći od train-testa i *cross_val_score*-a u NDCG 2 puta, a u MAP čak 10 puta. Razlog ovome je nasumičnost učenja stabla odlučivanja, gdje se u podjelama na 10 skupova podataka događa veća varijacija u podacima nego u ova dva ostala primjera.

Kod klasifikacije su opet zabilježeni loši, ali prihvatljivi rezultati u metodi train-test i *cross_val_score*. S druge strane u unakrsnoj validaciji *sklearn.metrics* zabilježeni su dobri rezultati. Odziv je veći od preciznosti u train-testu i unakrsnoj validaciji *sklearn.metrics*, dok je u *cross_val_score*-u vrijednost preciznosti malo veća (približno ista). Kod metoda evaluacije sustava preporučivanja zabilježena je velika sposobnost pronalaska novog sadržaja (novitet), najveća pokrivenost kataloga, dosta velika personalizacija i jako mala različitost.

4.3.6 Klasifikator K susjeda

I naposljetku kod klasifikatora K susjeda je opet zabilježena dobra evaluacija regresije te se predviđanja dobro preslikavaju na modelu. Što se rangiranja tiče, zabilježene su loše vrijednosti te je opet zabilježen dosta veći NDCG od MAP-a. Kod evaluacije klasifikacije zabilježeni su loši, ali prihvatljivi rezultati te je odziv veći od preciznosti. Zabilježeni novitet je velik, kao i pokrivenost kataloga i personalizacija, dok je zabilježena vrijednost različitosti mala.

4.4 Najbolja metoda preporuke

Prije nego se zaključi rad, trebalo bi analizirati dobivene rezultate i utvrditi koja bi od metoda preporučivanja bila najbolja u preporučivanju pjesama. Mjere evaluacije regresije sve imaju jako bliske vrijednosti te jedino TF-IDF ima malo veću vrijednost od ostalih. Također treba uzeti u obzir da druge metode imaju veće outliere od TF-IDF-a, ali svejedno su vrijednosti rezultata evaluacije manje na ostalim metodama nego na TF-IDF-u. Prema tome bi sve metode osim TF-IDF-a bile dobre za korištenje po evaluaciji regresije.

Prema evaluaciji rangiranja najveće su vrijednosti zabilježene za KNN i SVD te oni imaju najbolje sposobnosti prikazivanja relevantnih podataka u TOP-N rezultata. Ovo je posebno važno u situacijama kao u slučaju sustava preporučivanja kojeg želimo razvit, jer će korisnik uvijek prvo vidjeti stvari koje se nalaze u TOP-N elemenata što posredno utječe na kvalitetu interakcije korisnika i na financije. Razlog zašto su najveće vrijednosti evaluacije rangiranja zabilježene za SVD i KNN su: bolje prepoznavanje uzoraka u podacima i izbjegavanje klasifikacije podataka.

Što se vrijednosti evaluacije klasifikacije tiče, one su malo veće u prosjeku za metode SVD i KNN naspram ostalih algoritama. Također je u te dvije metode zabilježen veći odziv od preciznosti što povećava broj lažnih pozitivnih vrijednosti naspram istinitih negativnih vrijednosti. Osim toga, ovdje ne treba zaboraviti algoritam stabla odlučivanja gdje se ističe vrijednost dobivena unakrsnom validacijom sklearn.metrics. Sve u svemu, za evaluaciju klasifikacije najbolje metode preporučivanja bi bile KNN i SVD.

Što se tiče metoda evaluacije karakterističnih za sustave preporučivanja najbolje su vrijednosti zabilježene za stablo odlučivanja i klasifikator K susjeda. Oba algoritma imaju relativno visoki novitet, pokrivenost kataloga i personalizaciju, dok im je različitost niska kao u ostalim

algoritmima. Ostali algoritmi imaju barem jednu od ovih vrijednosti znatno nižu nego klasifikator K susjeda i stablo odlučivanja. Razlozi boljih performansi ovih spomenutih algoritama naspram ostalih su: bolje stvaranje kompleksnih putanja i bolje mogućnosti klasifikacije i regresije. Stoga klasifikator K susjeda i stablo odlučivanja mogu bolje otkriti nove stvari, bolje pokrивaju stvari te bolje preporučuju po mjeri korisnika.

Teško je reći koja bi od ovih spomenutih metoda preporučivanja bila najbolja za korištenje za potrebe sustava preporučivanja za Spotify, no mogu se istaknuti neke prednosti svake od metoda preporučivanja. Jako je istaknuta prednost koju SVD i KNN imaju u evaluaciji rangiranja i malo manje u evaluaciji klasifikacije, dok s druge strane istu takvu dojmljivu prednost imaju klasifikator K susjeda i stablo odlučivanja za personalizaciju i pokrivenost kataloga. Kako su ove sve navedene metode jako važne u izgradnji sustava preporučivanja, bilo bi ih korisno upariti za izgradnju što boljeg preporučitelja. Kombinacijom različitih metoda bi se mogla postići i dobra personalizacija, pokrivenost kataloga, kvaliteta klasifikacije i rangiranja. Stoga bi za dobivanje svih tih prednosti bilo najbolje koristiti hibrid ovih spomenutih modela. Hibrid može kombinirati različite metode, to jest može biti kombinacija SVD-stablo odlučivanja ili KNN- klasifikator K susjeda ili neka treća kombinacija ovih metoda.

5. Zaključak

Kroz rad su predstavljeni različite metode preporučivanja i evaluacije sustava preporučivanja. Međusobno su uspoređeni različite metode preporučivanja i evaluacijom su se istraživali odnosi između njih. Na kraju je zaključeno koji bi sustav preporučivanja bio najkompatibilniji za korištenje na platformi Spotify. Kroz provedene se evaluacije također može primijetiti kako i koja od metoda preporučivanja oblikuje svoje preporuke i što uzimaju u obzir pri preporučivanju.

Primijećeno je kako rezultati za evaluaciju regresije imaju relativno bliske vrijednosti jedan drugom. S druge su strane kod evaluacije klasifikacije i rangiranja, za SVD i KNN su zabilježene nešto veće vrijednosti nego za ostale algoritme zato što bolje prepoznaju uzorke i izbjegavaju klasificirati podatke. Kod evaluacije karakteristične za sustave preporučivanja zabilježeni su natprosječno dobri rezultati za klasifikator K susjeda i za stablo odlučivanja. Razlog tome su veće i bolje izgrađene putanje te bolje mogućnosti klasifikacije i regresije.

Također su zabilježena razna ograničenja i nedostaci svake od metoda preporučivanja. Logistička je regresija imala lošije rezultate za pokrivenost kataloga zato što je ona manje robusna i ima slabiju moć klasifikacije nego stablo odlučivanja. Što se NDCG-a tiče, logistička regresija lošije razumije uzorke, lošije razumije latentne faktore te više stavlja fokus na binarnu klasifikaciju nego rangiranje. Kod TF-IDF-a su zabilježene jako niske vrijednosti za mjere evaluacije karakteristične za sustave preporučivanja primarno zato što TF-IDF isključivo uzima tekstualne podatke u obzir i više pažnje daje težine riječi naspram relevantnosti riječi. Za SVD i KNN su primijećeni nedostaci u personalizaciji i pokrivenosti kataloga naspram ostalih sustava preporučivanja. Razlozi tome su razni, ali neki od njih su to da spomenuti algoritmi dobro prepoznaju dominantne uzorke u matrici korisnik-stvar što ih sprječava u pretraživanju van matrice. Također su osjetljivi na šum podataka u matrici

Evaluacija sustava preporučivanja je iznimno bitna za uspjeh pojedinih metoda sustava preporučivanja te se konstantnim ispitivanjem i poboljšavanjem osigurava da određen sustav preporučivanja bude dugo upotrebljiv i isplativ. Posebno su sustavi preporučivanja važni u kontekstu vremena gdje se ljudske interakcije sve više smanjuju i virtualno preuzima primat. Stoga je za očekivat da će se u budućnosti sustavi preporučivanja imati sve veću važnost. Stoga je važno razumijeti kako se pojedini sustavi preporučivanja koriste i evaluiraju.

6. LITERATURA

3. [https://monkeylearn.com/blog/what-is-tf-idf/#:~:text=TF%2DIDF%20\(term%20frequency%2D,across%20a%20set%20of%20documents.](https://monkeylearn.com/blog/what-is-tf-idf/#:~:text=TF%2DIDF%20(term%20frequency%2D,across%20a%20set%20of%20documents.)
5. https://users.ics.forth.gr/~potamias/mlnia/paper_6.pdf
7. <https://analyticsindiamag.com/singular-value-decomposition-svd-application-recommender-system/>
8. https://dl.acm.org/doi/pdf/10.1145/1148170.1148257?casa_token=2J-4qudJqOkAAAAA:nz2lCCrtWY9Xeu5rvv7aXU6SW_v7neS8QFgrrgUrWsqDcVOIIA85lua7AvM_IK74Wr6loME26EQ
15. https://www.researchgate.net/profile/Loc-Nguyen-101/publication/321753015_Model-based_approach_for_Collaborative_Filtering/links/5a4f80894585151ee284dfb3/Model-based-approach-for-Collaborative-Filtering.pdf
16. https://www.researchgate.net/profile/Loc-Nguyen-101/publication/321753015_Model-based_approach_for_Collaborative_Filtering/links/5a4f80894585151ee284dfb3/Model-based-approach-for-Collaborative-Filtering.pdf
17. https://www.sciencedirect.com/science/article/pii/S1877050915007462?ref=pdf_download&fr=RR-2&rr=844f04863a93248a
20. <https://web.stanford.edu/class/cs168/l19.pdf>
21. https://www.sciencedirect.com/science/article/pii/S0020025507001223?casa_token=e9nkNVtRUH8AAAAA:FbG9mGloDCsmB2bsHr2CU1MRP4-pAaEX4UrhPeoPLQP7jEvQaS52tz7qsVKpLyXK_eXeuDAS
22. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10119255&casa_token=rdrvPyLJ7jsAAAAA:m_UpJomwKrwBbzgWLnjL4jJWtgTuTGp9uROsL5PWwJF10_Jy3_snFSNL2aJ3E5Rk305UX1dU&tag=1
24. https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1541-0420.2010.01392.x?casa_token=zljktWxqo2IAAAAA:TF18zKANYypS9LzTjAtSbP9dofdRT7-SqNGbrNUutwe5ZvmXaV1qJfhiK3VYLYyX3ouQcU2DIVh8w
26. https://www.researchgate.net/profile/David-Bell-32/publication/221549123_KNN_Model-Based_Approach_in_Classification/links/55927ef908ae15962d8e937d/KNN-Model-Based-Approach-in-Classification.pdf
27. https://books.google.hr/books?id=b1CqCAAQAQBAJ&dq=lazy+learning+in+machine+learning&pg=PA1&redir_esc=y#v=onepage&q=lazy%20learning%20in%20machine%20learning&f=false, Aha, David (29 June 2013). *Lazy Learning* (illustrated ed.). Springer Science & Business Media, 2013. p. 424. ISBN 978-9401720533. Retrieved 30 September 2021.
28. <https://medium.com/@luigi.fiori.1f0303/distance-metrics-and-k-nearest-neighbor-knn-1b840969c0f4#:~:text=The%20formula%20to%20calculate%20Euclidean,total%20is%20our%20Euclidean%20distance.>
29. https://www.itm-conferences.org/articles/itmconf/pdf/2017/04/itmconf_ita2017_04008.pdf
30. <https://www.ibm.com/docs/en/ias?topic=knn-usage>
38. <https://www.capitalone.com/tech/machine-learning/understanding-tf-idf/>
39. https://spotintelligence.com/2022/11/28/tf-idf/#Why_is_TF-IDF_used_in_machine_learning
40. https://www.contentpowered.com/blog/tfidf-algorithm-content-seo/#how_does_google_use_tfidf
42. <https://ojs.aaai.org/aimagazine/index.php/aimagazine/article/view/18139>
44. <https://www.nvidia.com/en-us/glossary/data-science/recommendation-system/>

45. <https://link.springer.com/article/10.1007/s10257-014-0269-9>
46. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8275856&casa_token=Trg2jdMA9lQAAAAA:lyu_Vds_z_XtPKSRBhyZf4giD7aVQndWrFdPkksqbs4rWRaJcgm8aVNboopWm3DY2oejvP2s9&tag=1
47. <https://www.argoid.ai/blog/5-use-case-scenarios-for-recommendation-systems-and-how-they-help>
48. <https://www.mordorintelligence.com/industry-reports/recommendation-engine-market>
49. <https://research.aimultiple.com/recommendation-system/>
50. <https://link.springer.com/article/10.1007/s00146-020-00950-y>
51. <https://analyticsindiamag.com/how-to-measure-the-success-of-a-recommendation-system/>
52. https://www.researchgate.net/profile/Douglas-Mahoney/publication/5402488_Linear_Mixed_Effects_Models/links/57e560bf08ae9227da964db4/Linear-Mixed-Effects-Models.pdf#page=278
53. <https://www.ibm.com/topics/logistic-regression>
54. <https://www.analyticsvidhya.com/blog/2021/08/conceptual-understanding-of-logistic-regression-for-data-science-beginners/#h-how-does-logistic-regression-work>
55. <https://aws.amazon.com/what-is/logistic-regression/#:~:text=Logistic%20regression%20is%20a%20data,outcomes%2C%20like%20yes%20or%20no.>
56. <https://www.analyticsvidhya.com/blog/2021/08/decision-tree-algorithm/>
57. <https://medium.com/@shmuel.aka/recommendation-systems-based-decision-tree-20197b118f22>
58. <https://medium.com/analytics-vidhya/the-epsilon-greedy-algorithm-for-reinforcement-learning-5fe6f96dc870>
59. <https://ts2.space/en/the-epsilon-greedy-algorithm-unlocking-the-potential-of-reinforcement-learning/>
61. <https://medium.datadriveninvestor.com/evaluation-metrics-101-7c8b4c3421c2>
62. https://www.fer.unizg.hr/_download/repository/SU1-2022-P21-VrednovanjeModela.pdf
63. <https://www.iguazio.com/glossary/model-evaluation/>
64. <https://medium.com/mlearning-ai/automatic-evaluation-of-recommendation-systems-coverage-novelty-and-diversity-cc140330d3e7>
65. <https://statisticsbyjim.com/regression/root-mean-square-error-rmse/>
66. <https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d>
68. <https://www.analyticsvidhya.com/blog/2021/07/metrics-to-evaluate-your-classification-model-to-take-the-right-decisions/>
69. <https://towardsdatascience.com/comprehensive-guide-to-ranking-evaluation-metrics-7d10382c1025>
70. <https://medium.com/@linduri03/evaluation-metrics-for-search-ranking-and-recommendation-systems-14c8c9f2804d>
71. <https://medium.com/@readsumant/understanding-ndcg-as-a-metric-for-your-recomendation-system-5cd012fb3397#:~:text=Normalized%20Discounted%20Cumulative%20Gain%20or,relevant%20products%20are%20ranked%20higher.>
72. <https://www.aporia.com/learn/a-practical-guide-to-normalized-discounted-cumulative-gain-ndcg/>
73. <https://sdsawtelle.github.io/blog/output/mean-average-precision-MAP-for-recommender-systems.html>
74. <https://medium.com/mlearning-ai/automatic-evaluation-of-recommendation-systems-coverage-novelty-and-diversity-cc140330d3e7>

76. <https://github.com/statisticianinstilettos/recmetrics/tree/master>
77. <https://www.comet.com/site/blog/understanding-hold-out-methods-for-training-machine-learning-models/#:~:text=What%20is%20the%20Hold%2DOut,both%20model%20evaluation%20and%20selection>.
78. <https://www.fer.unizg.hr/download/repository/SU1-2023-P01-Uvod.pdf>
79. <https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761>
80. <https://www.ibm.com/topics/knn#:~:text=Next%20steps-,K%2DNearest%20Neighbors%20Algorithm,of%20an%20individual%20data%20point>.
81. <https://www.nature.com/articles/s41598-022-10358-x>
82. <https://www.turing.com/kb/collaborative-filtering-in-recommender-system#collaborative-filtering-in-recommender-system>
83. <https://www.clerk.io/blog/collaborative-filtering>
84. <https://www.analyticssteps.com/blogs/what-collaborative-filtering-types-working-and-case-study>
86. [https://datajobs.com/data-science-repo/Recommender-Systems-\[Netflix\].pdf](https://datajobs.com/data-science-repo/Recommender-Systems-[Netflix].pdf)
87. <https://towardsdatascience.com/how-does-collaborative-filtering-work-da56ea94e331>
88. <https://towardsdatascience.com/uncovering-how-the-spotify-algorithm-works-4d3c021ebc0>
89. <https://medium.com/analytics-vidhya/7-types-of-hybrid-recommendation-system-3e4f78266ad8>
90. Recommender Systems: Charu C. Aggarwal, 2016.
91. <https://jupyter-notebook.readthedocs.io/en/stable/notebook.html>
92. <https://www.analyticsvidhya.com/blog/2015/01/scikit-learn-python-machine-learning-tool/>
93. <https://surpriselib.com/>
94. <https://www.askpython.com/python/examples/pipelining-in-python>
95. https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html
96. <https://www.analyticsvidhya.com/blog/2021/10/evaluation-metric-for-regression-models/#:~:text=RMSE%20value%20with%20zero%20indicates,residual%20to%20the%20ground%20truth>.
97. <https://towardsdatascience.com/choosing-the-right-metric-is-a-huge-issue-99ccbe73de61>
98. <https://support.madkudu.com/hc/en-us/articles/4407471867405-Recall-and-Precision-how-is-the-performance-of-a-prediction-model-measured-#:~:text=Decent%20model%3A%20Recall%20%3D%2060%2D,Excellent%20model%3A%20Recall%20%3E%2080%25>
99. <https://neptune.ai/blog/recommender-systems-metrics>
100. <https://www.sciencedirect.com/science/article/pii/S0957417423011612>
101. <https://iamirmasoud.com/2022/06/19/understanding-micro-macro-and-weighted-averages-for-scikit-learn-metrics-in-multi-class-classification-with-example/>
103. <https://scikit-learn.org/stable/modules/neighbors.html#classification>
104. <https://utsavdesai26.medium.com/mastering-dimensionality-reduction-exploring-pca-and-svd-methods-f7d55c9ca3c9#:~:text=Dimensionality%20reduction%3A%20SVD%20can%20reduce,language%20processing%2C%20and%20recommendation%20systems>.
105. <https://medium.com/@corymaklin/memory-based-collaborative-filtering-user-based-42b2679c6fb5>

106. <https://medium.com/@beepabose/content-based-filtering-for-book-recommendation-using-pyspark-4369c4cbe006>
107. https://www.researchgate.net/publication/369503320_Feature_selection_and_importance_of_predictors_of_non-communicable_diseases_medication_adherence_from_machine_learning_research_perspectives/figures?lo=1
108. <https://medium.com/geekculture/decision-tree-construction-in-ml-5ede70eb0596>
109. <https://www.geeksforgeeks.org/epsilon-greedy-algorithm-in-reinforcement-learning/>
110. <https://www.semanticscholar.org/paper/Performance-Evaluation-of-Recommender-Systems-Chen-Liu/7d6090ef21f15f9f1210b6f96664e6a3a0e6b507>
111. https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html