SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1982

SUSTAV ZA PREPORUČIVANJE VIJESTI TEMELJEN NA VELIKOM SKUPU PODATAKA

Barbara Bobeta

Zagreb, lipanj 2025.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1982

SUSTAV ZA PREPORUČIVANJE VIJESTI TEMELJEN NA VELIKOM SKUPU PODATAKA

Barbara Bobeta

Zagreb, lipanj 2025.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 3. ožujka 2025.

ZAVRŠNI ZADATAK br. 1982

Pristupnica:	**Barbara Bobeta (0036553203)**
Studij:	Elektrotehnika i informacijska tehnologija i Računarstvo
Modul:	Računarstvo
Mentor:	izv. prof. dr. sc. Alan Jović

Zadatak: **Sustav za preporučivanje vijesti temeljen na velikom skupu podataka**

Opis zadatka:

Sustavi za preporučivanje (engl. recommender systems) česti su kod bilo kojih tvrtki koje žele maksimizirati profit na temelju svoje baze korisnika i drugih uključenih dionika (npr. medijske kuće). Kod velikih skupova podataka, zadatak preporučivanja sadržaja postaje izazovan u smislu izbora najprikladnijeg algoritma u ovisnosti o domeni sadržaja. Cilj ovog završnog rada je teorijski istražiti i opisati metode preporučivanja sadržaja, uključujući suradničko filtriranje (engl. collaborative filtering), filtriranje temeljeno na sadržaju (engl. content-based filtering), preporučivanje zasnovano na dubokom učenju i hibridne pristupe. Nakon toga, potrebno je na temelju velikog skupa podataka namijenjenog preporučivanju pod nazivom MIND (https://msnews.github.io/) odabrati i implementirati najmanje jedan prikladan algoritam za preporučivanje vijesti. Model preporučivanja potrebno je vrednovati korištenjem odgovarajućih mjera vrednovanja. Implementaciju je potrebno napraviti u programskom jeziku po vlastitom izboru, a za izgradnju modela može se u slučaju nedostatka vlastitih sklopovskih resursa koristiti dostupna web rješenja (npr. Google Colab).

Rok za predaju rada: 23. lipnja 2025.

Zahvaljujem mentoru, izv. prof. dr. sc. Alan Joviću, na pomoći pri izradi završnog rada, kao i djedu i bakama na kontinuiranoj podršci kroz cijelo moje obrazovanje.

Sadržaj

Uvod

U današnjem ubrzanom svijetu, svakoga smo dana okruženi sve većim brojem informacija i podataka, stoga se stvara sve jača potreba za efikasnim i personaliziranim pretraživanjem istih. Kako bi povećale promet korisnika i krajnji profit, brojne tvrtke nastoje spoznati navike svojih kupaca i na račun njih im predlagati relevantne ponude i informacije. Ovo personalizirano pretraživanje i predlaganje sadržaja ostvaruje se upravo korištenjem sustava za preporučivanje. Od preporuka videa na društvenim mrežama, preko predložene odjeće u internetskim trgovinama, pa sve do personaliziranih vijesti na internetskim portalima, sustavi preporuka omogućuju korisnicima da brže i jednostavnije dođu do sadržaja koji ih zanima.

Zbog kratkog vijeka aktualnosti vijesti te brzorastućeg broja novih, dolazi do posebne problematike prilikom personaliziranog predlaganja vijesti čitateljima. Potrebno je ostvariti kvalitetan i precizan algoritam preporuka koji će moći raditi s velikim brojem podataka i istovremeno ostati relevantan u stvarnom vremenu.

Cilj je ovog završnog rada istražiti i opisati različite pristupe preporučivanja sadržaja, uključujući suradničko filtriranje, filtriranje temeljeno na sadržaju, metode temeljene na dubokom učenju i hibridne pristupe. Nakon pregleda teorijskog dijela, na temelju velikog skupa podataka pod nazivom MIND odabran je i implementiran najmanje jedan prikladan algoritam za preporučivanje vijesti te naposljetku taj je isti model vrednovan korištenjem odgovarajućih mjera vrednovanja.

Ovaj je rad podijeljen u pet dijelova, s pročitanim uvodom, u drugom dijelu prelazi se na teorijski pregled sustava za preporučivanje. Treće poglavlje daje pregled skupa podataka MIND. Četvrto poglavlje opisuje implementaciju odabranog modela, dok se u petom poglavlju prikazuju rezultati vrednovanja. Rad završava zaključkom i prijedlozima za budući rad.

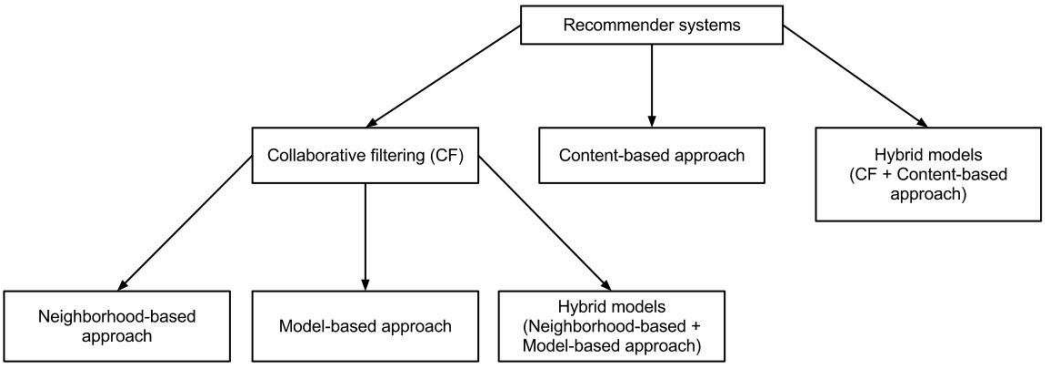
1. Teorijski pregled sustava za preporučivanje

1.1. Uvod u sustave za preporučivanje

Preporučiteljski sustavi su specijalizirani modeli koji analiziraju ponašanje korisnika s ciljem predlaganja relevantnog i zanimljivog sadržaja [1]. U današnje doba kada su korisnici svakodnevno izloženi velikoj količini informacija i podataka, ključno je omogućiti smanjenje preopterećenja te na taj način poboljšati korisničko iskustvo. Upravo su preporučiteljski sustavi ključ u stvaranju takvog personaliziranog iskustva temeljenog na prijašnjem ponašanju, interesu i preferencijama korisnika.

Danas su sustavi preporuke sveprisutni u različitim sferama digitalnog svijeta, od online trgovina, preko platformi za streaming glazbe i filmova, pa sve do društvenih mreža i vijesti. Tako na primjer, najveća platforma za streaming filmova i serija, Netflix, koristi sustave preporuke kako bi svojim korisnicima predložili relevantne filmove i serije na temelju njihovih prethodnih ocjena i pregledanog sadržaja. Spotify, vodeća glazbena platforma, generira personalizirane popise pjesama, dok Amazon, najveća internetska trgovina, preporučuje proizvode analizom korisničkih pretraga, recenzija i kupnji. Sustavi preporuka imaju značajnu ulogu i kod portala za vijesti poput Google Newsa i Microsoft Newsa, gdje korisnicima nude relevantne članke koji odgovaraju njihovim interesima.

S obzirom na način rada, sustavi preporuka mogu biti **nepersonalizirani** ili **personalizirani**. Nepersonalizirani sustavi generiraju jednake preporuke za sve korisnike, ne uzimajući u obzir njihove individualne interese ili ponašanje [2]. Prednosti takvih sustava je njihova jednostavna i brza implementacija jer se kriteriji preporuke uglavnom temelje na najnovijem, najbolje ocijenjenom, najpopularnijem ili najposjećenijem proizvodu. Nedostaci proizlaze iz činjenice da nema svaki korisnik preferencije za proizvod koji je u tom trenu najnoviji ili najpopularniji. Upravo se iz toga stvara potreba za razvojem personaliziranih sustava koji svakom korisniku pružaju personaliziranu i relevantnu preporuku na temelju prijašnjeg ponašanja, interesa i preferencija. U daljnjim dijelovima ovog rada fokus je upravo na personaliziranim sustavima preporuke prikazanim na slici (Sl. 1.1), a dan je i detaljan pregled glavnih metoda preporučivanja, uključujući suradničko i sadržajno filtriranje, duboko učenje te hibridne pristupe.

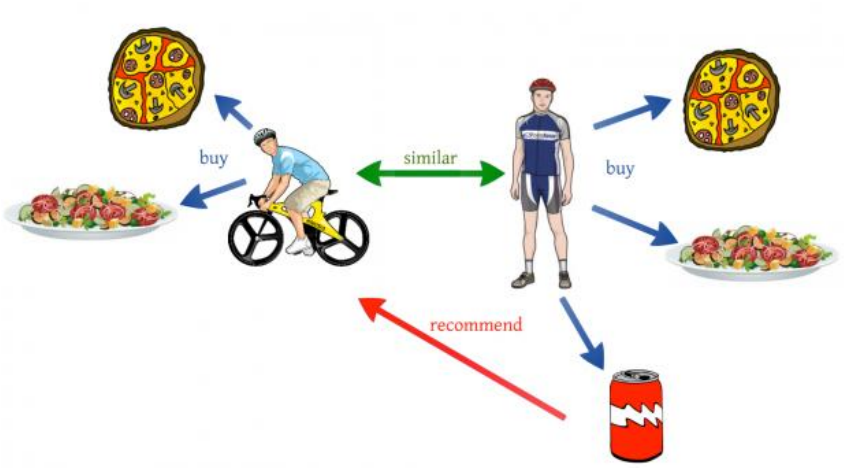


Sl. 1.1 Osnovna podjela preporučiteljskih modela [3]

1.2. Suradničko filtriranje

Suradničko filtriranje (engl. *Collaborative Filtering*) jedna je od najraširenijih i najčešće korištenih metoda u sustavima preporuka. Temelji se na ideji da se korisniku preporuče stavke koje su se svidjele drugim korisnicima sličnih interesa.

Na primjer, ako grupa korisnika ima slične preferencije kao ciljani korisnik, i ako se toj istoj grupi sviđa proizvod nepoznat ciljanom korisniku, sustav pomoću suradničkog filtriranja predlaže taj proizvod ciljanom korisniku, što je prikazano na sljedećem primjeru (Sl. 1.2).



Sl. 1.2 Primjer suradničkog filtriranja [4]

Upravo na taj način suradničko filtriranje koristi zajedničke obrasce ponašanja korisnika kako bi se predvidjela njihova sklonost prema novim stavkama [5]. Glavna prednost suradničkog filtriranja je činjenica da je neovisan o informacijama o sadržaju proizvoda jer se preporuke u potpunosti temelje na ponašanju korisnika. Danas je ta metoda sustava

preporuka široko korištena, pa se na primjer jako precizni i kvalitetni preporučiteljski sustavi Amazona i Netflixa, temelje upravo na suradničkom filtriranju.

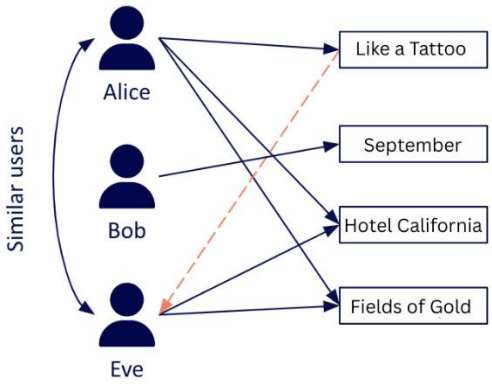
Postoje dvije temeljne vrste suradničkog filtriranja: sustavi temeljeni na memoriji (sustavi temeljeni na susjedima) i sustavi temeljeni na modelu.

1.2.1. Sustavi temeljeni na memoriji

Sustavi temeljeni na memoriji temelje se na analizi susjedstva jer pokušavaju predvidjeti odnos korisnika prema određenom proizvodu na temelju ponašanja sličnih korisnika ili sličnih proizvoda. Samo ime aludira na činjenicu da se sustav ne uči kao model, već koristi sirove podatke u stvarnom vremenu [6]. Takvi sustavi se dijele na dva podtipa: sustavi orijentirani prema korisniku (engl. *User-based*) i sustavi orijentirani prema proizvodu (engl. *Item-based*).

Sustavi orijentirani prema korisniku

Pretpostavimo da postoje dva korisnika koji imaju vrlo sličan glazbeni ukus, pod time podrazumijevamo da su korisnik A i korisnik B slušali veći broj istih pjesama i slično ih ocijenili. Pjesmu “Like a Tattoo - Sade” korisnik A nije poslušao, ali ju je korisnik B poslušao i pozitivno ocijenio. Logičan zaključak bi bio pretpostaviti da će se korisniku A također svidjeti ista pjesma i upravo će mu ju ovakav sustav i predložiti, što se vidi na slici (Sl. 1.3). Sustav preporučuje stavke ciljanom korisniku na temelju preferencija drugih korisnika, uspoređuje bivše ponašanje ciljanog korisnika s ponašanjem drugih, njemu sličnih korisnika [6].



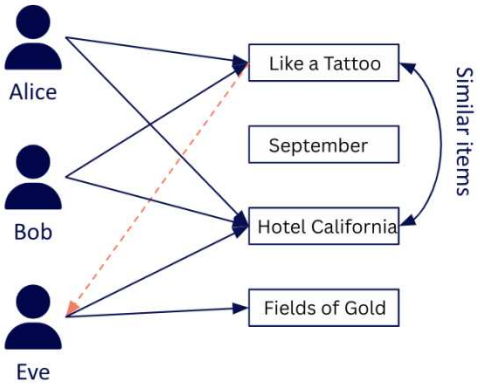
Sl. 1.3 Primjer suradničkog filtriranja orijentiranog prema korisniku

Sustav orijentiran prema korisniku koristi reprezentaciju udaljenosti sličnosti (kosinusna sličnost, Pearsonova koleracija i sl.) između ciljanog korisnika i ostalih aktivnih korisnika, te na temelju tog izračuna, koristeći "*K-nearest Neighbour Algorithm*" traži *K* najbližih susjeda ciljanom korisniku. Sustav zatim stvara predikciju ponašanja korisnika prema težinskom prosjeku ponašanja *K* odabranih susjeda [6]. Ovime se stvara **matrica *User-Item*** na temelju koje se predviđa interakcija korisnika (ocjene, klikovi, lajkovi i sl.) sa stavkama s kojima još nije došao u doticaj.

Sustavi preporuke temeljeni na memoriji imaju kvalitetne rezultate rada na manjim platformama, no kako broj korisnika raste, sve je zahtjevnije izračunati sličnosti između svih parova korisnika.

Sustavi orijentirani prema proizvodu

Ovoga puta, umjesto predlaganja nove pjesme korisniku A na temelju preferencija njemu sličnog korisnika B, korisniku A sada predlažemo nove pjesme koje su slične onima koje je već poslušao i pozitivno ocijenio, što je prikazano i na primjeru (Sl. 1.4). Ovakav sustav orijentiran prema proizvodu predlaže nove stavke ciljanom korisniku na temelju njegovog prijašnjeg ponašanja prema sličnim stavkama. Bitno je za naglasiti da se ovakav model sustava preporuke ne fokusira na sličnost između atributa stavki (filtriranje temeljeno na sadržaju), već isključivo na sličnost u korisničkoj interakciji prema stavkama.



Sl. 1.4 Primjer suradničkog filtriranja orijentiranog prema proizvodu

Za početak, sustav traži udaljenosti između svih parova stavki koristeći neku vrstu funkcije sličnosti (npr. kosinusna sličnost), potom, koristeći težinsku sumu ili prosjek, sustav stvara listu stavki najsličnijih prethodno ocijenjenim stavkama korisnika. Sustavi orijentirani prema proizvodu se u pravilu bolje skaliraju od sustava orijentiranih prema korisniku jer je

broj stavki uobičajeno manji od broja korisnika, pa se matrica sličnosti može izračunati unaprijed i koristiti za više korisnika.

1.2.2. Sustavi temeljeni na modelu

Sustavi preporuke temeljeni na modelu grade prediktivni model strojnog učenja koji se koristi za predviđanje preferencija korisnika prema do tad neviđenim stavkama. Modeli koriste vrijednosti matrice *User-Item* kao skup podataka za učenje i s pomoću njih stvaraju predikcije za nedostajuće vrijednosti. U stvaranju modela koriste se različite metode strojnog učenja kao što su grupiranje, matična faktorizacija i duboko učenje. Jedna od najšire korištenih tehnika je tehnika matrične faktorizacije [6].

Matrična faktorizacija prikazuje matricu *User-Item R* veličine *m* x *n* (*m* - broj korisnika, *n* - broj stavki) kao produkt dvije pravokutne matrice manjih dimenzija (1.1).

$$R \approx U\Sigma V \tag{1.1}$$

U je matrica latentnih faktora korisnika veličine *m* x *k*, dok je *V* matrica latentnih faktora stavki veličine *k* x *n* . Zbog toga se matrična faktorizacija često klasificira kao tip latentno faktornog modela. Kao takav model, matrična faktorizacija pretpostavlja da se sličnost između stavki ili korisnika mogu odrediti kroz određen broj drugih značajki koje nije potrebno definirati, već učiti iz podataka o korisničkim interakcijama. Najpoznatije metode matrične faktorizacije uključuju *Singular Value Decomposition*, *Alternating Least Squares* i optimizirane verzije poput *FunkSVD* koje su prilagođene za rad s rijetkim matricama [7].

Ovakvi sustavi dobro funkcioniraju s velikim i rijetkim skupovima podataka i omogućuju dobru skalabilnost na većem broju korisnika i stavki.

1.2.3. Prednosti i nedostaci suradničkog filtriranja

Prednosti suradničkog filtriranja

- **Neovisnost o sadržaju stavki:** Suradničko filtriranje se temelji isključivo na podacima o korisničkom ponašanju, što znači da je u potpunosti neovisno o atributima stavki i samim time te iste stavke ne treba razumjeti. Zbog toga je primjena

ovakvog sustava preporuka široka, može se koristiti u raznim sferama digitalnog svijeta, pa se tako koristi u stvaranju preporuka filmova, proizvoda, glazbe i slično.

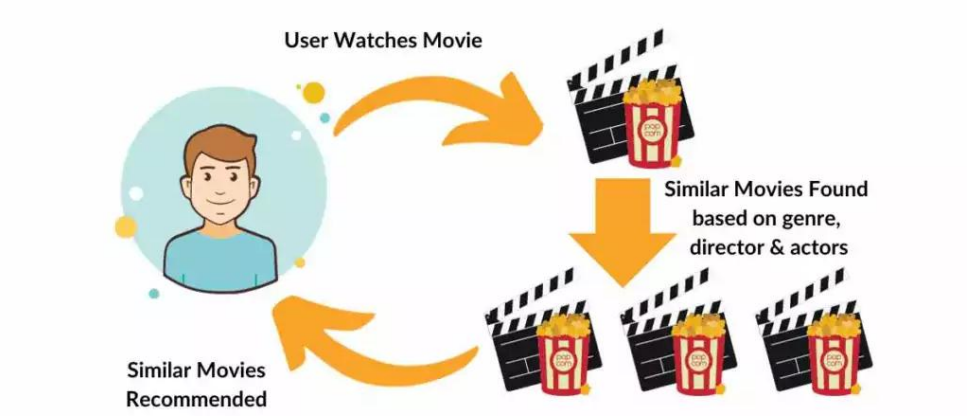
- **Neočekivane preporuke:** S obzirom na to da preporuke nisu zasnovane na atributima stavki, već isključivo na ponašanju korisnika sličnih ciljanom korisniku, može doći do potpuno neočekivanih i sadržajno nepovezanih preporuka. Takve preporuke mogu uključivati stavke koje su u potpunosti drugačije od prethodno preferiranih stavki korisnika.
- **Skalabilnost uz optimirane algoritme (Sustavi temeljeni na modelima):** Suradničko filtriranje, posebice varijante zasnovane na modelima poput matrične faktorizacije, vrlo su skalabilne kad se koriste uz optimirane algoritme. Takvi sustavi mogu učinkovito učiti model i generirati preporuke i kod vrlo velikih skupova podataka, s velikim brojem korisnika i stavki.

Nedostaci suradničkog filtriranja

- **Hladni start (engl. *cold start*):** Kada dođe do prijave novog korisnika ili dodavanja nove stavke, sustav preporuke temeljen na suradničkom filtriranju nema dovoljno podataka za usporedbu ponašanja i stvaranja preporuka [6]. To je najšire kritiziran nedostatak ovakvog sustava preporuke, posebice u generiranju preporuka vijesti gdje se kontinuirano pojavljuju novi članci.
- **Razrijeđenost podataka (engl. *sparsity*):** U stvarnim slučajevima većina korisnika ocjenjuje ili čita jako mali broj dostupnih stavki, zbog toga su matrice *User-Item* običajno poprilično prazne što, naravno, otežava stvaranje preporuka. Što je matrica rjeđa, to preporuke postaju sve manje točne. Zbog toga su često korišteni sustavi temeljeni na matričnoj faktorizaciji, kao što je npr. *Singular Value Decomposition*, koji koriste latentne faktore i pune praznine reduciranjem dimenzionalnosti.
- **Skalabilnost (Sustavi temeljeni na memoriji):** Suradničko filtriranje, posebice u sustavima temeljenim na memoriji, zahtijeva značajne vremenske i memorijske resurse prilikom stvaranja preporuka u sustavima s velikim brojem korisnika i stavki. Povećanjem broja korisnika ili stavki, postaje sve teže otkriti točne susjede i stvoriti preciznu preporuku.

1.3. Filtriranje temeljeno na sadržaju

Filtriranje temeljeno na sadržaju (engl. *Content-Based Filtering*) je preporučiteljski sustav koji koristi atribute stavki kako bi preporučio slične stavke korisniku [8]. Takvo se filtriranje razlikuje od ostalih preporučiteljskih sustava po tome što za generiranje preporuka nije potrebno znati preferencije drugih korisnika. Sve preporuke su građene isključivo na preferencijama ciljanog korisnika. U slučaju stvaranja sustava preporuka filmova, ako korisnik gleda filmove različitih žanrova, ali se u svim njima pojavljuje glumac Matthew McConaughey, sustav mu može preporučiti neki do tada nepogledani film s Matthewom u jednoj od uloga (Sl. 1.5).



Sl. 1.5 Primjer filtriranja temeljenog na sadržaju [9]

U sustavima filtriranja temeljenog na sadržaju, u suštini se uspoređuje profil korisnika i profil stavki kako bi se predvidjela interakcija *User-Item* te na taj način stvorile kvalitetne i relevantne preporuke.

Matrica korisnika je dvodimenzionalna matrica koja sadrži sve informacije vezane za korisnike. Svaki red ove matrice predstavlja jednog korisnika, a svaki stupac predstavlja brojčanu reprezentaciju korisnika. Ta reprezentacija čini profil korisnika, vektor kojem su izražene preferencije korisnika uobičajeno formiran agregacijom značajki svih stavki koje je korisnik pozitivno ocijenio [10].

Koristi se isti koncept i u gradnji **matrice stavki**, dvodimenzionalne matrice s informacijama o stavkama sustava. Svaki redak predstavlja jednu stavku, a stupac brojčanu reprezentaciju te stavke. Takva reprezentacija čini profil stavke, numerički vektor koji predstavlja skup svih značajki relevantnih za sadržaj stavke, kao što su u slučaju vijesti, kategorija članka, ključne

riječi i slično. Takve stavke su iz tekstnog oblika konvertirane u numerički oblik korištenjem metoda kao što su TF-IDF ili *one-hot* kodiranje [10].

Na taj su način korisnici i stavke smješteni unutar zajedničkog vektorskog prostora što nam olakšava računanje sličnosti i usporedbu njihovih profila.

Za računanje sličnosti koristimo različite metrike. Jedna od češće korištenih je već spomenuta **kosinusna sličnost**, koja predstavlja veličinu kuta između dva vektora, s vrijednostima između -1 i 1. Što je vrijednost izračunata izrazom (1.2) veća, to su dvije stavke predstavljene vektorima sličnije [6].

$$Cosine(x, y) = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} {x_i}^2} \sqrt{\sum_{i=1}^{n} {y_i}^2}} \tag{1.2}$$

Osim kosinusne sličnosti, može se koristiti i **Euklidsku udaljenost**, duljina pravca koji spaja dvije točke vektora, predstavljena izrazom (1.3). Najmanja vrijednost koju udaljenost može poprimiti je nula, a što je ta udaljenost manja, to su vektori sličniji [6].

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (y_{i-} x_i)^2} \tag{1.3}$$

Korištenjem takvih mjera sličnosti uspoređuje se profil korisnika s profilima stavki, čime se predviđaju stavke koje odgovaraju korisnikovim preferencijama te se stvara kvalitetna preporuka punjenjem matrice *User-Item*.

1.3.1. Prednosti i nedostatci filtriranja temeljenog na sadržaju

Prednosti filtriranja temeljenog na sadržaju

- **Neovisnost o drugim korisnicima:** S obzirom na to da se preporuke grade isključivo na temelju preferencija ciljanog korisnika i atributa stavki, nije potrebno imati informacije o preferencijama i ponašanju drugih korisnika. U trenutku kada sustav ima podatke o ciljanom korisniku, može odmah početi stvarati relevantne preporuke bez usporedbe s ponašanjem drugih korisnika.
- **Kvalitetne i personalizirane preporuke:** Zbog neovisnosti o ponašanju drugih korisnika prilikom stvaranja preporuka, svaki korisnik dobiva visoko personalizirane

i kvalitetne preporuke temeljene na njegovim preferencijama. Što korisnik ima više interakcija sa sustavom, preporuke su sve točnije i preciznije [10].

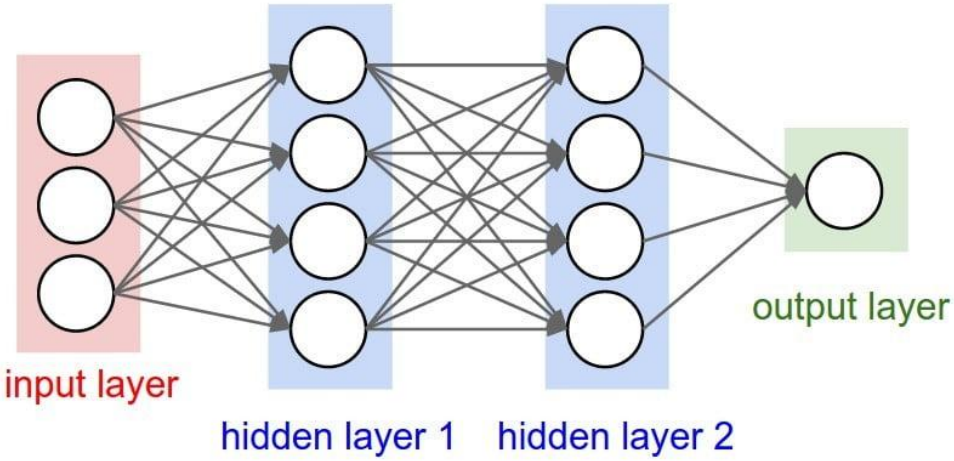
- **Otpornost na hladni start za stavke:** Najveći je nedostatak suradničkog filtriranja ležao u nemogućnosti stvaranja preporuka za tek dodane stavke ili korisnike. U filtriranju temeljenom na sadržaju, čak i kada postoji mali broj korisnika i njihovih podataka, mogu se stvoriti kvalitetne preporuke bez korištenja podataka o ponašanju drugih korisnika. Zbog toga mnoge platforme u ranim stadijima njihovog rasta koriste filtriranje temeljeno na sadržaju, koje kasnije, s rastom broja korisnika, zamjenjuju sa suradničkim filtriranjem.

Nedostaci filtriranja temeljenog na sadržaju

- **Ograničena raznolikost (engl. *overspecialization*):** Sustav ima tendenciju preporučivanja sličnih stavki onima s kojima je korisnik već interaktirao. Takav pristup dovodi do smanjenja raznolikosti preporuka [10]. U slučaju da je korisnik pogledao Spider-Man 1 i Spider-Man 2, velika je vjerojatnost da će sustav iduće predložiti filmove kao što su Spider-Man 3, The Amazing Spider-Man, Captain America i slične Marvelove filmove, zatvarajući korisnika u uzak prostor žanrova.
- **Ovisnost o kvaliteti značajki:** U svrhu stvaranja kvalitetne i precizne preporuke, potrebno je dobro definirati značajke stavki. Sustav u velikoj mjeri ovisi o kvaliteti informiranosti značajke koje opisuje stavke i zbog toga previše općenite ili nepovezane značajke dovode do nekvalitetnih i nerelevantnih preporuka.
- **Skalabilnost:** Usko vezano uz prethodni nedostatak, za svaku dodatnu stavku potrebno je precizno definirati njene značajke. Zbog toga je teško skalirati sustave kojima broj stavki raste eksponencijalno. Ručno unošenje atributa stavki oduzima mnogo vremena i može biti nekonzistentno ovisno o broju ljudi koji to obavlja.

1.4. Preporuke temeljene na dubokom učenju

Prije definiranja preporučiteljskih sustava temeljenih na dubokom učenju, važno je razumjeti pojam dubokog učenja. Riječ je o podskupu metoda strojnog učenja koji primjenjuje algoritme inspirirane strukturom i funkcijom mozga. Duboko učenje koristi višeslojne neuronske mreže, zvane duboke neuronske mreže prikazane na slici (Sl. 1.6), kako bi se simulirao rad mozga prilikom donošenja kompleksnih zaključaka. Dok se u drugim metodama strojnog učenja podaci analiziraju linearno, sustavom dubokog učenja omogućena je nelinearna analiza podataka i samim time povećavana učinkovitost u ekstrakciji značajnih informacija. Najveća prednost korištenja dubinskog učenja je njegova sposobnost izgradnje novih značajki, koje se u tradicionalnim pristupima ručno definiraju, na temelju različitih kombinacija iz početnog skupa značajki [11].

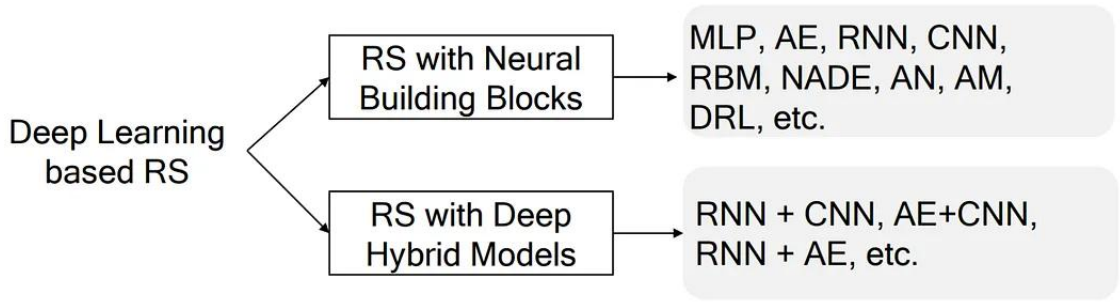


Sl. 1.6 Duboka višeslojna neuronska mreža [12]

U kontekstu sustava preporuka, duboko učenje se koristi zbog svoje sposobnosti obrade velikog broja različitih podataka, od kontekstnih, preko tekstnih, pa sve i do vizualnih tipova podataka. Njegova sposobnost povezivanja nelinearnih veza između korisnika i stavki, stvara izrazito precizne i personalizirane preporuke.

S obzirom na njihovu izgradnju, sustavi preporuke temeljeni na dubokom učenju su kategorizirani u dvije skupine, prikazane slikom (Sl. 1.7):

- **Preporuke temeljene na neuronskim građevnim blokovima** (jedan model dubokog učenja)
- **Duboki hibridni modeli** (korištenje više različitih modela dubokog učenja istovremeno)



Sl. 1.7 Kategorije sustava preporuke temeljenih na dubokom učenju [13]

U prvoj kategoriji, preporuka temeljenih na neuronskim građevnim blokovima, koristi se samo jedan tip dubokog modela u izgradnji sustava preporuka. Dvije najčešće korištene arhitekture ovakvog tipa sustava preporuka su **rekurentne neuronske mreže (RNN)** i **konvolucijske neuronske mreže (CNN)**.

- **Rekurentne neuronske mreže:** Rekurentne neuronske mreže su korisne u situacijama kada je redoslijed interakcija korisnika važan npr. u preporukama glazbe, vijesti ili videozapisa. One čuvaju informacije o prethodnim koracima korisnika i na temelju njih modeliraju ponašanje korisnika kroz vrijeme [14]. Ovo je posebice korisno u situacijama kada se korisnici ne prijavljuju u sustave, nego sustav za preporuke mora izgraditi predviđanja na temelju vrlo rijetkih podataka pohranjenih u kolačićima. Prednosti korištenja rekurentne neuronske mreže su još vidljivije s dugotrajnim odlukama korisnika prilikom njegovog prijavljenog korištenja sustava. Tada će model imati pristup cijeloj povijesti odluka i na taj način generirati preciznije i kvalitetnije preporuke.
- **Konvolucijske neuronske mreže**: Konvolucijske neuronske mreže su se prvotno koristile u računalnom vidu, ali su se pokazale izrazito korisnima u domeni preporuka na temelju analize vizualnog ili tekstnog sadržaja. Ovakve mreže su temeljene na životinjskoj percepciji vida, počevši od analize baznih atributa kao što su rubovi tijela, do prepoznavanja objekata višeg reda kao što su lica, automobili na ulici i

slično. U sustavima preporuke, ovakvi modeli mogu automatski naučiti reprezentaciju slike nekog proizvoda, odjeće ili vijesti i koristiti tu informaciju za stvaranje personaliziranih preporuka. Često se ovakva arhitektura koristi u sustavima za preporuku odjeće, u kojima konvolucijske neuronske mreže na temelju boje, stila i nekih drugih vizualnih značajki može usporediti komad odjeće s prethodno kupljenim ili pregledanim artiklima korisnika i na taj način izgraditi kvalitetnu preporuku [14].

U drugoj kategoriji, dubokih hibridnih modela, koriste se kombinacije više arhitektura kako bi se povećala učinkovitost i preciznost preporuka. Na primjer, može se koristiti CNN za ekstrakciju značajki iz stavki te zatim RNN za analizu vremenskog slijeda interakcija korisnika.

1.4.1. Prednosti i nedostatci preporuka temeljenih na dubokom učenju

Prednosti preporuka temeljenih na dubokom učenju

- **Učenje kompleksnih nelinearnih odnosa:** Za razliku od tradicionalnih sustava preporuka, korištenjem dubokih neuronskih mreža model može prepoznati i naučiti složene uzorke i nelinearne odnose između korisnika i stavki te na temelju njih izgraditi preciznije i personaliziranije preporuke.
- **Automatsko učenje značajki:** U mnogim klasičnim modelima, značajke svih novih stavki su se morale dugotrajnim procesom ručno definirati. Duboko učenje omogućava modelima da sami otkriju korisne značajke stavki te na taj način poboljšaju mogućnost skaliranja sustava na veliki broj novih stavki i podataka.
- **Skalabilnost:** Uz pravilno učenje i održavanje, duboki modeli se mogu koristiti u izgradnji sustava preporuka s velikim brojem korisnika i stavki.

Nedostaci preporuka temeljenih na dubokom učenju

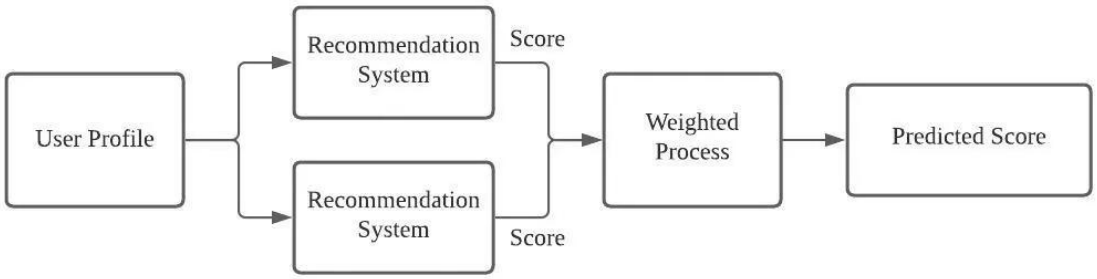
- **Velika količina podataka:** Za učenje dubokih modela potrebne su velike količine podataka kako ne bi došlo do preučenja ili slabih performansi.
- **Složenost razvoja, održavanja i računalnih zahtjeva:** Učenje i izvođenje dubokih neuronskih mreža zahtijeva veću količinu računalne snage, što može biti skupo i zahtjevno. Osim toga potrebna su i velika znanja iz područja strojnog i dubokog učenja, čime se povećava složenost održavanja takvih sustava.
- **Manjak interpretabilnosti:** Zbog funkcioniranja dubokih modela kao "crne kutije", teško je razumjeti zašto je nekom korisniku preporučena određena stavka. To postaje problem u sustavima u kojima je bitno pružiti korisnicima takvo objašnjenje [14].

1.5. Hibridni sustavi preporuka

S obzirom na sve veću raznolikost podataka i sve složenije korisničke potrebe, teško je pronaći jedan model sustava preporuka koji savršeno odgovara našim potrebama. Većinu vremena, kombinacijom više različitih modela dobijaju se kvalitetniji rezultati i bolje performanse. Upravo hibridni sustavi preporuka kombiniraju više različitih metoda preporuka, danas najčešće suradničko filtriranje i filtriranje temeljeno na sadržaju. Cilj korištenja hibridnih sustava preporuka je iskoristiti prednosti svakog od korištenih sustava i istovremeno smanjiti njihove slabosti i nedostatke, kao što su problem hladnog starta, ograničene raznolikosti, razrijeđenosti podataka i slično.

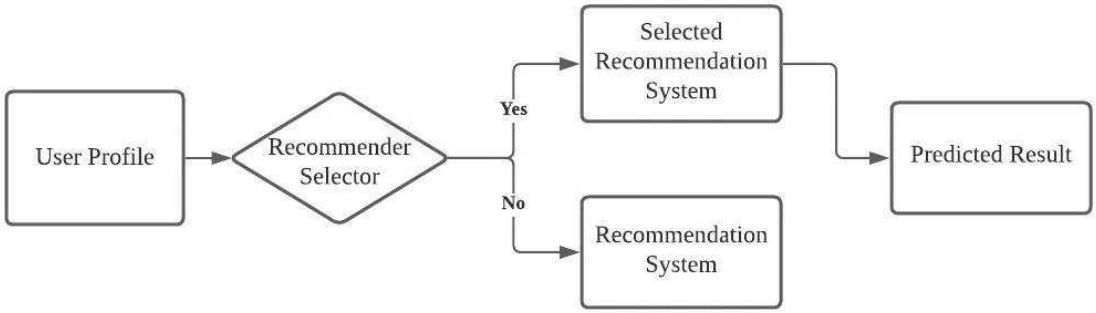
Postoje različiti pristupi gradnje hibridnih sustava preporuka. Burke u svome radu [15] dijeli hibridne sustave na sljedećih sedam tipova:

1. ***Weighted*:** Preporuke se generiraju odvojeno pomoću više metoda, a zatim se njihove predikcije kombiniraju u krajnji rezultat (Sl. 1.8).



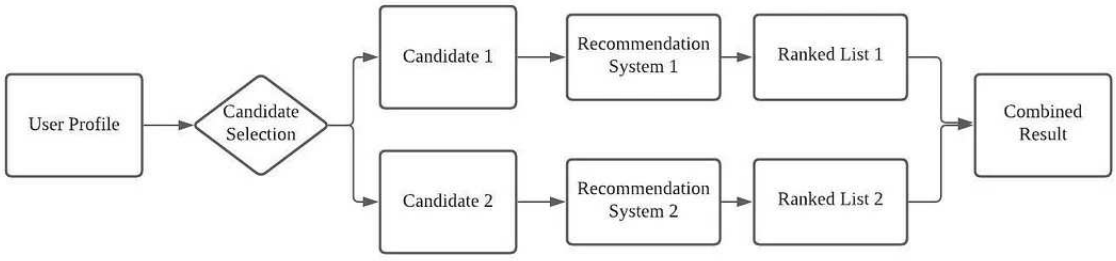
Sl. 1.8 *Weighted* hibridni sustav za preporučivanje [16]

2. ***Switching*:** Sustav odabire jednu metodu sustava preporuka ovisno o kontekstu. Ovakav pristup modelu dodaje dodatan sloj u kojemu se odabire prikladna tehnika (Sl. 1.9).



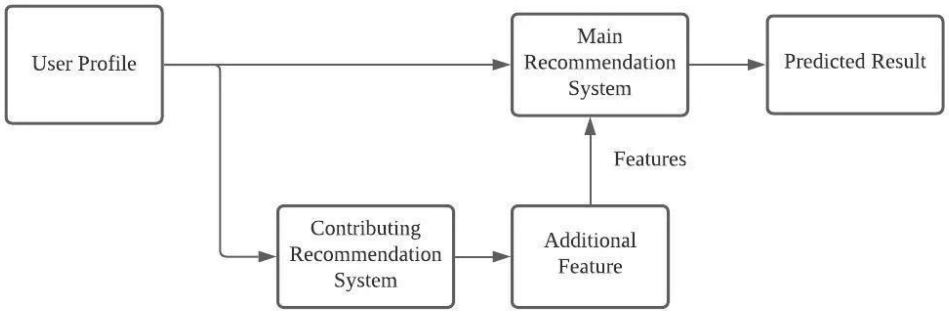
Sl. 1.9 *Switching* hibridni sustav za preporučivanje [16]

3. ***Mixed*:** Preporuke više različitih modela sustava preporuka prikazuju se zajedno (Sl. 1.10).



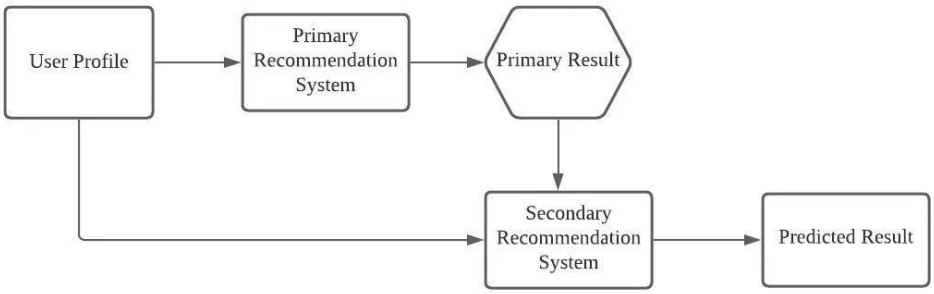
Sl. 1.10 *Mixed* hibridni sustav za preporučivanje [16]

4. ***Feature combination*:** Značajke iz različitih modela sustava preporuka se kombiniraju u jedan algoritam preporuka (Sl. 1.11).



Sl. 1.11 *Feature combination* hibridni sustav za preporučivanje [16]

5. ***Cascade*:** Prvi algoritam se koristi za generiranje početnog skupa preporuka, dok se drugi algoritam koristi za dodatnu obradu tog skupa (Sl. 1.12).



Sl. 1.12 *Cascade* hibridni sustav za preporučivanje [16]

6. ***Feature augmentation*:** Izlaz jednog algoritma preporuke je ulaz u drugi (Sl. 1.13).

Sl. 1.13 *Feature augmention* hibridni sustav za preporučivanje [16]

7. ***Meta-Level*:** Ovaj tip hibridnog sustava preporuke sličan je *Feature augmentation* pristupu, ali umjesto da se koristi samo izlaz prvog modela kao ulaz drugog, koristi se cijeli naučeni model kao ulaz.

1.5.1. Prednosti i nedostatci hibridnih sustava preporuka

Prednosti hibridnih sustava preporuka

- **Smanjenje slabosti pojedinačnih metoda:** Korištenjem kombinacija različitih pristupa mogu se ublažiti tipični problemi pojedinih sustava, poput problema hladnog starta kod novih korisnika ili stavki u suradničkom filtriranju ili problema ograničene raznolikosti u filtriranju temeljenog na sadržaju [17].
- **Povećana preciznost i točnost:** Zbog korištenja više različitih izvora podataka, vrsta informacija i tehnika stvaranja preporuka, hibridni sustavi često daju točnije i preciznije preporuke.
- **Fleksibilnost:** Hibridni sustavi se lako mogu prilagoditi specifičnim značajkama domene koju promatra kombiniranjem različitih metoda koje najbolje odgovaraju kontekstu u kojemu se nalazi.

Nedostaci hibridnih sustava preporuka

- **Povećana složenost razvoja i održavanja:** Zbog integracije više različitih metoda, razvoj i održavanje samog sustava postaje sve zahtjevnije i kompliciranije. Potrebno je više znanja i truda za razvoj, testiranje i održavanje sustava.
- **Veći računalni zahtjevi:** Kombiniranje više različitih algoritama i modela zahtijeva više računalnih resursa što rezultira višim troškom i većim vremenom obrade preporuka.
- **Manjak interpretabilnosti:** Zbog korištenja više različitih pristupa istovremeno, može biti teško objasniti korisnicima zašto je određena stavka preporučena, što postaje problem u sustavima u kojima je bitno pružiti korisnicima takvo objašnjenje.

1.6. Evaluacija sustava preporučivanja

Evaluacija je nužna kako bi se utvrdila učinkovitost sustava preporuka u generiranju relevantnih i točnih prijedloga korisnicima. Bez evaluacije se ne mogu uspoređivati različiti modeli niti odrediti koji pristup najbolje odgovara danom kontekstu. Evaluacija sustava preporuka je ključna za osiguranje relevantnosti i učinkovitosti modela te zadovoljstva korisnika sustava.

Tipičan slijed evaluacije sustava može se podijeliti na dvije glavne faze: **evaluaciju u kontroliranom okruženju** (engl. ***offline***) **i evaluaciju tijekom izvođenja** (engl. ***online***).

1.6.1. Offline evaluacija

Offline evaluacija koristi unaprijed prikupljene skupove podataka u kontroliranom okruženju za predviđanje interakcija korisnika i mjerenje različitih metrika. Tipično se podaci dijele na skup za učenje i skup za testiranje. Ovakva evaluacija je ključna za podešavanje i poboljšanje modela prije njegove implementacije u stvarno okruženje.

Najčešće korištene metrike u *offline* evaluaciji su:

- **Preciznost na *K*:** Mjeri udio relevantnih stavki među prvih *K* stavki prema izrazu (1.4). Govori koliko je relevantnih preporuka među ponuđenim te daje procjenu ispravnosti predviđanja [18]. Jednostavna je i intuitivna za korištenje, ali njena točnost ovisi o broju relevantnih stavki korisnika. Preciznost je korisna u situacijama kada postoji mnogo relevantnih stavki za svakog od korisnika, od kojih se mora izabrati i prikazati samo nekolicina njih.

$$Preciznost\ na\ K\ = \frac{Broj\ relevantnih\ preporuka}{Ukupan\ broj\ preporuka} \tag{1.4}$$

- **Odziv na *K*:** Mjeri pokrivenost relevantnih stavki u *top-K* stavki prema izrazu (1.5). Omjerom preporučenih relevantnih stavki u odnosu na ukupan broj svih relevantnih stavki lako se može odrediti pokrivenost cijelog sustava.

$$Odziv\ na\ K = \frac{Broj\ relevantnih\ preporuka}{Ukupan\ broj\ relevantnih\ preporuka} \tag{1.5}$$

- **Mjera F-beta:** Metrika koja uravnotežuje preciznost i odziv stavki prema izrazu (1.6). Kombiniranjem metrike preciznosti i odziva pruža jednu uravnoteženu procjenu vrijednosti. Parametar beta predstavlja omjer važnosti odziva naprama preciznosti. Više važnosti nosi odziv ukoliko je vrijednost parametra beta veća od jedan, suprotno, u slučaju kada je vrijednost parametra manja od jedan, metrika favorizira preciznost. Mjera F-beta je korisna kada je bitna točnost predviđanja i mogućnost pokrivanja što više relevantnih stavki [18] .

$$F_\beta = \frac{(1+\beta^2)^2 \times Preciznost \times Odziv}{(\beta^2 \times Preciznost)+Odziv} \tag{1.6}$$

- **Srednja apsolutna pogreška (MAE):** Mjeri razliku između stvarnih i predviđenih korisničkih ocjena prema izrazu (1.7).

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |Y_i - \overline{Y}_i| \tag{1.7}$$

- **Korijen srednje kvadratne pogreške (RMSE):** Također mjeri razliku između stvarnih i predviđenih korisničkih ocjena prema izrazu (1.8), ali mnogo manje kažnjava model kada je blizu stvarnog predviđanja i mnogo više kada je model daleko od stvarnog predviđanja u usporedbi s MAE [20].

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (Y_i - \overline{Y}_i)^2} \tag{1.8}$$

- **Stopa pogotka (engl. *hit rate*):** Mjeri udio korisnika koji su dobili barem jednu relevantnu preporuku prema izrazu (1.9). Za početak je potrebno generirati *top-K* preporuka za korisnika. Pogotkom se smatra situacija u kojoj se barem jedna relevantna preporuka pojavljuje među *top-K* preporukama. *Hit rate* se uobičajeno povećava s povećanjem parametra *K*, što je veća lista preporuka, veća je vjerojatnost da će se relevantna preporuka naći među njima.

$$Hit\ rate = \frac{Broj\ pogodaka}{Broj\ korisnika} \tag{1.9}$$

- **nDCG (*normalized Discounted Cumulative Gain*):** Mjeri koliko su relevantne stavke visoko rangirane na listi preporuka prema izrazu (1.10), uzimajući u obzir poziciju stavke na listi. Stavke koje su pozicionirane visoko na listi nose veću težinu.

$$NDCG = \frac{DCG_p}{IDCG_p} = \frac{\sum_{i=1}^{p} \frac{rel_i}{\log(i+1)}}{\sum_{i\prime=1}^{p} \frac{rel_{i\prime}}{\log(i\prime+1)}} \quad (1.10)$$

- **Pokrivenost (engl. *coverage*):** Procjenjuje koliko dio svih dostupnih stavki sustav može preporučiti korisnicima. Veća pokrivenost znači da sustav nije ograničen na preporučivanje samo najpopularnijih stavki, već ima sposobnost preporučivanja većeg broj stavki iz cijelog skupa podataka. Takvi sustavi imaju raznolikije preporuke što može doprinijeti boljem korisničkom iskustvu. S druge strane, niska pokrivenost odražava činjenicu da sustav preporuka često predlaže isti skup stavki, što rezultira smanjenoj raznolikosti preporuka.

- **Raznolikost (engl. *diversity*):** Ocjenjuje raznolikost artikala preporučenih korisnicima, procjenjuje koliko su različite preporučene stavke za svakog korisnika prema izrazu (1.11). Ukoliko korisnik gleda crtani film Ledeno doba 1. Sustav niske raznolikosti bi mu preporučio samo iduće dijelove Lednog doba, dok bi sustav visoke raznolikosti preporučio i ostale filmove sličnog žanra ili drugih značajki. Visoka raznolikost nije uvijek dobra, potpuno nasumične stavke nam također daju visoku raznolikost, ali loše preporuke [19].

$$Raznolikost = (1 - S),\ S = avg\ sličnost\ između\ preporučenih\ parova \quad (1.11)$$

- **Novitet (engl. *novelty*):** Procjenjuje koliko su preporučene stavke jedinstvene korisnicima, to jest, u kojoj se mjeri predložene stavke razlikuju od popularnih. Visoki rezultati novosti nisu uvijek dobri, kao i u slučaju raznolikosti, korištenjem nasumičnih stavki mogu se dobiti visoki rezultat novosti, ali opet i loše preporuke.

Glavne prednosti *offline* evaluacije su njena jednostavnost i brzina, jako brzo se mogu testirati različiti modeli i parametri te provjeriti performanse modela. Međutim, zbog

korištenja povijesnih podataka, rezultati evaluacije mogu biti optimistični i često nisu odraz stvarnih interakcija korisnika.

1.6.2. *Online* evaluacija

Online evaluacija se izvodi tijekom stvarnog korištenja sustava, koristeći aktualne korisničke interakcije. Za razliku od *offline* evaluacije koja koristi povijesne podatke za simulaciju stvaranja preporuke, *online* evaluacija mjeri stvarne reakcije korisnika na preporuke. Ovakav tip evaluacije daje najtočniji uvid u učinkovitost sustava i korisničko iskustvo.

Najčešće metode *online* evaluacije su:

- **A/B testiranje:** Metoda za usporedbu dvije verzije algoritma preporuke kako bi se utvrdilo koja metoda ima bolju izvedbu na temelju unaprijed definiranih metrika. Potrebno je korisnike podijeliti u dvije grupe, prva grupa (grupa A) dobiva preporuke generirane prvim modelom, dok grupa B dobiva preporuke generirane drugim [20]. Naposljetku se usporedbom metrika kao što su klikovni postotak, vrijeme provedeno na stranici i slično, procjenjuje koji model ima bolje rezultate.

- **Multivarijantno testiranje (MVT):** Oblik A/B testiranja koji uključuje testiranje više od dvije varijante sustave istovremeno. Ideja je da se istovremeno mogu testirati kombinacije više različitih varijabli (npr. model preporuke + boja teksta) kako bi se uvidjelo koja od njih ima najbolje performanse.

- **Ispreplitanje (engl. *interleaving*):** Tehnika u kojoj se korisniku zajedno predstavljaju preporuke generirane iz različitih modela te se na temelju interakcije korisnika s ovim isprepletenim preporukama određuje koje modele preporuka korisnici preferiraju. Daje izravnu usporedbu između različitih modela pri identičnim uvjetima što može pomoći u donošenju informiranih odluka o odabiru modela [20].

- **Interaktivne metrike korisničkog ponašanja:**
  - ***Click-Through Rate* (CTR):** Izravna metrika koja mjeri koliko često korisnici klikaju na preporuke, daje omjer klikova na preporučene stavke u odnosu na

ukupan broj prikazanih stavki. Ovakva metrika se uobičajeno koristi u aplikacijama koje koriste online oglašavanje. Što je CTR veći, to su preporuke privlačnije i relevantnije korisnicima [18].

- ***Conversion Rate*:** Metrika za praćenje udjela klikova koji rezultiraju željenom radnjom, kao što je kupnja proizvoda, gledanje filma ili čitanje članka [19].

- ***Dwell Time*:** Prosječno vrijeme koje korisnik provodi u interakciji sa specifičnom stavkom/preporukom. Dulje vrijeme zadržavanja često označava veću zainteresiranost i zadovoljstvo korisnika. Ovakva metrika se koristi u poboljšanju relevantnosti i kvaliteta budućih preporuka, posebice u sustavima u kojima je bitna duljina interakcije korisnika, a ne samo klik.

Glavna prednost *online* evaluacije je što precizno mjeri ponašanje stvarnih korisnika u realnim uvjetima. Takva evaluacija donosi mnogo korisnih informacija koje se kasnije koriste za poboljšanje ili promjenu trenutačnog modela sustava preporuka. Provedba ovakvog tipa evaluacije zahtijeva znatno više vremena i resursa od *offline* evaluacije. Potrebno je neko vrijeme za postavljanje samog okvira, zatim prikupljanja dovoljno podataka te naposljetku izračuna metrike.

Kombinacijom *offline* i *online* evaluacijskih metoda može se dobiti puna slika o efikasnosti i performancama modela. Nužno je kvalitetno evaluirati korišteni sustav preporuka kako bi se u model unijele promjene koje će rezultirati poboljšanjem korisničkog iskustva i njihove interakcije sa sustavom.

2. Pregled skupa podataka MIND

MIND (Microsoft News Dataset) [21] je skup podataka velikih razmjera koji se koristi za potrebe istraživanja i evaluacije preporučiteljskih sustava temeljenih na vijestima. Podaci su prikupljeni s platforme Microsoft News i sadrže anonimne zapise ponašanja korisnika. Nasumično je uzorkovano milijun korisnika koji su imali barem pet klikova na vijesti u razdoblju od šest tjedana od 12. rujna do 22. studenog 2019. godine. U svrhu zaštite privatnosti, svi su korisnici sigurno hashirani u obliku anonimnog ID-a.

Skup podataka dijeli se na tri skupa, prvi podskup koji se koristi za učenje modela, a obuhvaća ponašanja korisnika iz petog tjedna, s poviješću temeljenom na prva dva tjedna. Drugi podskup korišten za validaciju modela sadrži uzorke prikupljene zadnji dan petog tjedna, dok treći podskup, korišten za njegovo testiranje, sadrži zapise ponašanja korisnika u šestom tjednu prikupljanja podataka.

Pored punog skupa MIND, postoji i MIND-small, koji predstavlja umanjenu verziju skupa. Podaci su temeljeni na uzorkovanju 50 tisuća nasumičnih korisnika, a naposljetku su podijeljeni samo na podskupove za učenje i validaciju modela.

2.1. Struktura skupa podataka

Skup podataka MIND je strukturiran kako bi podržao stvaranje preporuka temeljenih na sadržaju vijesti i ponašanju korisnika. Skup podataka se sastoji od četiri različite datoteke spremljene u zip komprimiranoj mapi čiji su opisi prikazani u tablici (Tablica 2.1):

Tablica 2.1 Skup podataka MIND

Naziv datoteke	**Opis**
`behaviors.tsv`	Zapisi korisničke interakcije s vijestima
`news.tsv`	Detaljne informacije vezane uz vijesti
`entity_embedding.vec`	Vektorske reprezentacije entiteta vijesti izdvojene iz grafa znanja

relation_embedding.vec	Vektorske reprezentacije entiteta odnosa izdvojenih iz grafa znanja

Datoteka behaviors.tsv

Datoteka behaviors.tsv sadrži interakcije korisnika s vijestima. Svaki redak opisuje jedan zapis interakcije, a sadrži sljedeće informacije prikazane u tablici (Tablica 2.2):

Tablica 2.2 Struktura behaviors.tsv

Stupac	Opis	Primjer zapisa
Impression ID	Jedinstveni identifikator prikaza/interakcije vijesti	18
User ID	Anonimizirani identifikator korisnika	U67590
Time	Vrijeme interakcije u formatu "MM/DD/YYYY HH:MM:SS AM/PM"	11/15/2019 2:48:12 PM
History	Vremenski poredana lista identifikatora vijesti na koje je korisnik kliknuo prije klika na trenutni članak	N29177 N54540 N54827
Impressions	Lista vijesti koje su prikazane u ovoj interakciji i korisnikova interakcija s njima (1 za klik na vijesti i 0 za suprotno)	N52850-0 N24802-0 N51008-1 N53717-0 N4331-0 N38324-0 N17647-0 N32708-0

Datoteka news.tsv

Datoteka news.tsv sadrži detaljne informacije o vijestima koje su uključene u interakcijama u datoteci beahvior.tsv. Ima sedam stupaca koji sadrže sljedeće informacije prikazane u tablici (Tablica 2.3):

Tablica 2.3 Struktura `news.tsv`

Stupac	**Opis**	**Primjer zapisa**
`News ID`	Jedinstveni identifikator vijesti	`N22486`
`Category`	Glavna tema vijesti, npr "*lifestyle*", "*health*", "*news*"...	`health`
`SubCategory`	Podtema unutar glavne kategorije, npr. "*football_nfl*" u kategoriji "*sports*"	`wellness`
`Title`	Naslov vijesti koji ukratko sažima sadržaj članka	`How CrossFit injuries affect the shoulder`
`Abstract`	Kratak opis vijesti	`Question: Does CrossFit put me at risk for injury to my shoulder? Answer: The popularity of CrossFit has grown …`
`URL`	Veza na izvorni članak (uglavnom neaktivni)	`https://assets.msn.com/labs/mind/AAJ43sB.html`
`Title Entities`	Detektirani entiteti (osobe, mjesta, organizacije) u naslovu koji su povezani s Wikidata identifikatorima	`[{"Label": "CrossFit", "Type": "O", "WikidataId": "Q2072840", "Confidence": 1.0, "OccurrenceOffsets": [4], "SurfaceForms": ["CrossFit"]}]`
`Abstract Entities`	Detektirani entiteti (osobe, mjesta, organizacije) u tekstu sažetka koji su povezani s Wikidata identifikatorima	`[{"Label": "CrossFit", "Type": "O", "WikidataId": "Q2072840", "Confidence": 1.0, "OccurrenceOffsets": [15, 92, 205], "SurfaceForms": ["CrossFit", "CrossFit", "CrossFit"]}]`

Opisi ključeva rječnika koji se pojavljuju unutar entitetima u stupcima `Title Entities` i `Abstract Entities` su prikazani u tablici (Tablica 2.4):

Tablica 2.4 Opis ključeva u stupcima `Title` i `Abstract Entities`

Ključ	**Opis**
`Label`	Naziv entiteta kako je prikazan u tekstu te u Wikidata grafa znanja
`Type`	Vrsta entiteta prema Wikidata klasifikaciji
`WikidataId`	Jedinstveni identifikator entiteta u Wikidata grafa znanja
`Confidence`	Vrijednost između 0 i 1 koja predstavlja razinu sigurnosti da je entitet točno prepoznat i povezan
`OccurrenceOffsets`	Pozicija entiteta u tekstu predstavljena brojem znakova od početka
`SurfaceForms`	Izvorni prikaz entiteta onako kako se pojavio u članku

Datoteke `entity_embedding.vec` i `relation_embedding.vec`

Datoteke `entity_embedding.vec` i `relation_embedding.vec` sadrže 100-dimenzionalne vektore entiteta i odnosa naučene iz podgrafa WikiData grafa znanja koristeći metodu *TransE*. U obje datoteke prvi stupac predstavlja identifikator entiteta/relacije, a ostali stupci vrijednosti vektora, primjer jednog retka se može vidjeti u tablici (Tablica 2.5).

Tablica 2.5 Struktura `entitety_embedding.vec` i `relation_embedding.vec`

`ID`	`Embedding Values`
`Q42306013`	`0.014516 -0.106958 0.024590 ... -0.080382`

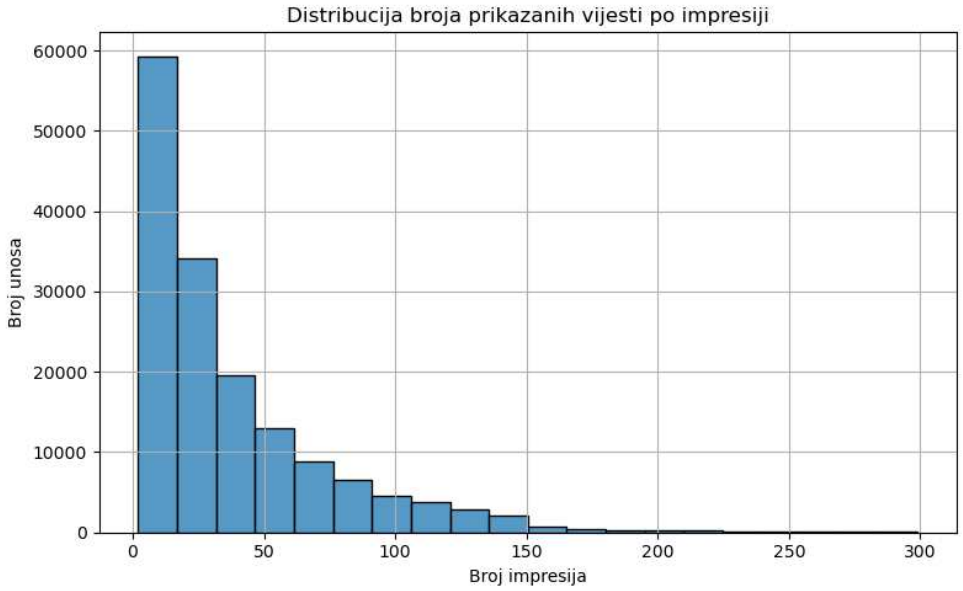
2.2. Analiza skupa podataka

U svrhu stvaranja bolje predodžbe o danom skupu podataka, potrebno ih je analizirati na način koji pruža kvalitetne i korisne informacije za izgradnju preporučiteljskog modela.

Korišteni skup podataka MINDsmall sadrži zapise 50.000 korisnika i 51.282 različitih vijesti. Ukupno je u datoteci behaviors.tsv sadržano 156.965 impresija koje nam daju informacije o ponašanju korisnika, kao što su njihove povijesti čitanja i interakcije s vijestima.

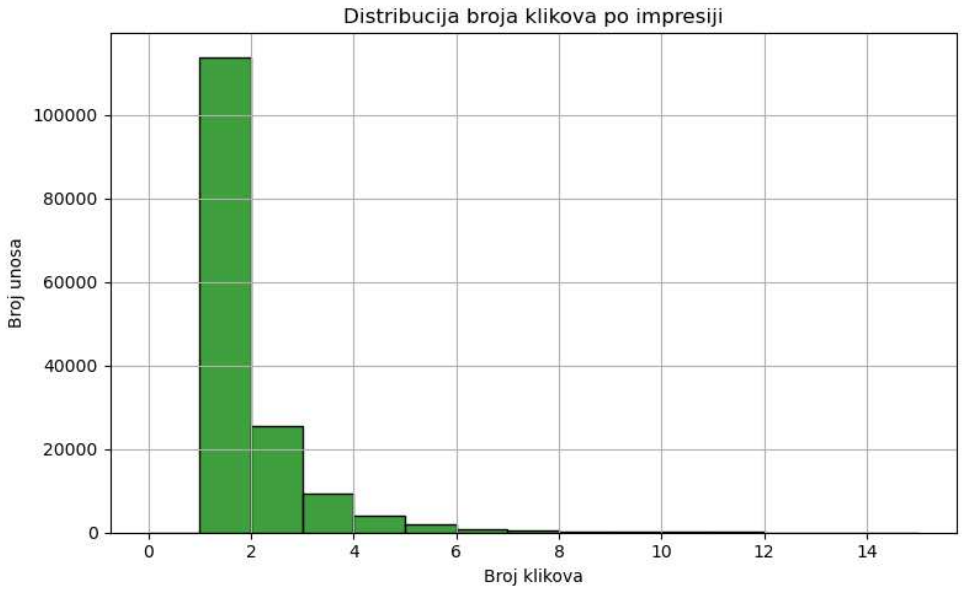
Svaka impresija sadrži popis prikazanih vijesti, od kojih su neke kliknute, a druge ne. Ukupno je zabilježeno 236.344 kliknutih vijesti te 5.607.100 preskočenih. Analizom je utvrđeno da je **prosječan broj prikazanih vijesti po impresiji 37,23**, dok je **prosječan broj klikova po impresiji samo 1,51**. Ovakav nerazmjer između broja prikazanih i kliknutih vijesti može znatno otežati stvaranje kvalitetnih preporuka zbog izrazito rijetke matrice *User-Item*.

Histogramski prikaz distribucije broja prikazanih vijesti po impresiji je prikazan na slici (Sl. 2.1). Na prikazu se vidi da vrijednosti idu sve do skoro 300 prikazanih vijesti po impresiji, dok ih najveći broj ima prikazanih oko 15.



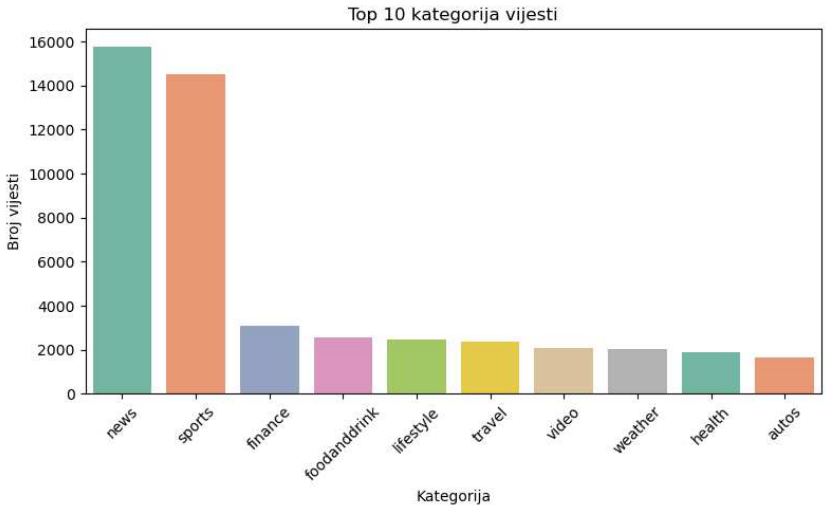
Sl. 2.1 Distribucija broja prikazanih vijesti po impresiji

Na sljedećem histogramu (Sl. 2.2) je prikazana distribucija broja kliknutih vijesti po impresiji. S najvećim brojem impresija sa samo jednom kliknutom vijesti, jasno se vidi mala količina interakcija korisnika sa sustavom. Ovako „škrta“ količina informacija o ponašanju korisnika može negativno utjecati na stvaranje kvalitetnih personaliziranih preporuka.



Sl. 2.2 Distribucija broja klikova po impresiji

Osim distribucije unutar impresija, zanimljivi su i podaci o pripadnosti vijesti različitim tematskim kategorijama i potkategorijama. Ukupno je **17 različitih kategorija i 264 različitih potkategorija.** Veliki broj različitih tematskih pripadnosti može otežati precizno modeliranje korisničkih interesa, pogotovo u slučajevima kad su pojedine kategorije ili potkategorije manje zastupljene. Vizualna analiza (Sl. 2.3) pokazuje da su pojedine kategorije vijesti, kao što su news i sports, znatno zastupljenije od drugih. Ovakva **neravnoteža među kategorijama** može uzrokovati pristranost modela prema češćim kategorijama, čime se smanjuje kvaliteta i raznolikost preporuka.



Sl. 2.3 Najzastupljenijih deset kategorija vijesti

Analiza je pokazala da su naslovi u prosjeku dugi 11 riječi, dok sažeci imaju prosječnu duljinu od 34 riječi. Ova informacija nam može koristiti prilikom izgradnje tekstnih reprezentacija vijesti.

Zanimljiva je i informacija da je korisniku u vremenskom razdoblju prikupljanja podataka **prikazano 116,87 vijesti**, od kojih je **prosječan broj klikova po korisniku samo 4,73**. Razlika u broju vijesti s kojima je korisnik u interakciji i ukupnih **50.000 različitih vijesti** predstavlja **najveći izazov** u gradnji ovoga modela. Model mora naučiti prepoznati korisnikove interese na temelju vrlo malog broja interakcija s vijestima te pokušati izgraditi relevantne preporuke iz nesrazmjerno većeg skupa svih vijesti.

3. Implementacija modela preporučivanja vijesti

3.1. Odabir pristupa

Kao što je opisano u prethodnim cjelinama ovoga rada, sustavi za preporuke mogu se izgraditi na mnogo različitih načina. Važno je kvalitetno proučiti dani skup podataka i zahtjeve sustava kako bi se odabrao zadovoljavajući pristup.

Za izgradnju sustava preporučivanja vijesti odabran je pristup temeljen na sadržaju. Ovaj je pristup odabran zbog mogućnosti generiranja preporuka isključivo na temelju atributa stavki, bez potrebe za analizom ponašanja drugih korisnika. Razmatrani su i drugi pristupi, poput suradničkog filtriranja i hibridnih modela, ali je zbog njihove veće složenosti i ograničenog vremena za izradu ovog rada odlučeno da će se koristiti jednostavniji pristup temeljen na sadržaju.

3.2. Opis arhitekture sustava

Arhitektura odabranog sustava za preporučivanje vijesti temelji se na filtriranju temeljenom na sadržaju. U tom okviru, sustav treba omogućiti stvaranje personaliziranih preporuka vijesti na temelju karakteristika već pročitanih članaka.

3.2.1. Stvaranje profila vijesti

Kako bi se atributi svake vijesti prikazali uniformno, potrebno je kvalitetno obraditi ulazne podatke i stvoriti profil za svaku vijest iz danog skupa `news.tsv`.

Tako će svaka vijest iz ulaznog skupa u ovom modelu biti opisana sljedećim atributima:

- `news_id` - jedinstveni identifikator vijesti
- `title` - naslov vijesti koji ukratko sažima sadržaj članka
- `abstract` - kratak opis vijesti
- `category` - glavna tema vijesti
- `subcategory` - podtema unutar glavne kategorije
- `time` – vrijeme čitanja vijesti

Prilikom predobrade tekstnih atributa vijesti (`title, abstract`) sva su slova pretvorena u mala te su uklonjene interpunkcije i posebni znakovi.

Nakon njihove predobrade, tekst se vektorizira korištenjem **metode TF-IDF (*Term Frequency-Inverse Document Frequency*)**. TF-IDF je statistička mjera koja se koristi u obradi prirodnog jezika i pronalaženju informacija za procjenu važnosti riječi u dokumentu u odnosu na zbirku dokumenata [22]. TF-IDF kombinira dvije komponente, učestalost riječi i inverznu učestalost dokumenata. Učestalost riječi mjeri koliko je često pojavljivanje neka riječi u dokumentu, dok inverzna učestalost dokumenta smanjuje težinu riječi koje se pojavljuje više puta u više različitih dokumenata, a radi obrnuto s rijetko viđenim riječima.

Ova metoda omogućuje predstavljanje tekstnih atributa vijesti kao numeričke vektore s odrazom važnosti pojedinih riječi u skupu podataka.

```
text = news["title"].fillna("") + " " + news["abstract"].fillna("")
tfid = TfidfVectorizer(max_features=6000)
text_vector = tfid.fit_transform(text)
```

Obrada atributa kategorije i potkategorije ostvarena je pretvorbom kategorijskih vrijednosti u numeričke pomoću metode *one-hot* kodiranja, prikazane na slici (Sl. 3.1).

***One-hot* kodiranje** je metoda za pretvaranje kategoričkih varijabli u binarni format. Stvara nove stupce za svaku kategoriju gdje 1 znači da je kategorija prisutna, a 0 znači da nije [23].

```
category_list = list(news["category"].dropna().unique())
subcategory_list = list(news["subcategory"].dropna().unique())

category_subcategory = []

for i in range(len(news)):
    line = news.iloc[i]

    category = [0] * len(category_list)
    subcategory = [0] * len(subcategory_list)

    if line["category"] in category_list:
        category[category_list.index(line["category"])] = 1

    if line["subcategory"] in subcategory_list:
        subcategory[subcategory_list.index(line["subcategory"])] = 1

    category_subcategory.append(category + subcategory)
```

Sl. 3.1 One-hot kodiranje kategorija i potkategorija

Zbog nedostatka raznolikosti u kategorijama i potkategorijama kasnije preporučenih vijesti, svim kodiranim vektorima kategorija i potkategorija je pridodana težina od 0,5.

```
category_subcategory = np.array(category_subcategory) * 0.5
```

Na samome kraju obrade vijesti se spajanjem TF-IDF vektora naslova i sažetka te *one-hot* kodiranih vektora kategorija i potkategorija stvara konačni vektor profila vijesti. Na ovaj način je osigurano da vektor vijesti sadrži potpunu informaciju o sadržaju i tematskoj pripadnosti vijesti.

3.2.2. Stvaranje profila korisnika

Druga komponenta ove arhitekture je odgovorna za izgradnju korisničkog profila. Profil korisnika se gradi na temelju vijesti koje je korisnik u prošlosti pročitao ili na koje je kliknuo u danoj impresiji.

Za svakog korisnika iz datoteke `behaviors.tsv` se stvaraju dva različita zapisa:

- `user_news_time`**:** mapa svih vijesti koje je korisnik pročitao u povijesti i vijesti na koje je korisnik kliknuo kada su mu bile prikazane (N33885-1), te vrijeme njihovog čitanja
- `unclicked_news`**:** lista svih vijesti koje su korisniku bile prikazane, ali na koje nije kliknuo (N42977-0)

Za svakog se korisnika potom dohvaćaju vektori svih pročitanih vijesti (`user_news_time`) te se ti vektori zbrajaju u jedan agregirani korisnički vektor.

Kako bi preporuke bile što preciznije, prilikom zbrajanja vektora, vijestima koje su nedavno pročitane se pridodaje veća težina od onih koje su pročitane ranije u prošlosti. U svrhu generiranja odgovarajućih težina, koristi se eksponencijalna opadajuća funkcija koja uzima u obzir proteklo vrijeme izraženo u danima s težinom od 1,5.

```
days = (max_train_time - time).day
weight = np.exp(-days * 1.5)
user_vector.append(news_vec * weight)
```

Zbog izrazito velikog broja različitih vijesti, a relativno malog broja pročitanih i kliknutih vijesti, matrica profila korisnika je rijetka. Potrebno ju je normalizirati kako bi se prilikom računanja sličnosti dobili što precizniji rezultati na kojima će se zasnivati preporuke. Normalizacija se provodi tako da se suma vektora svih pročitanih vijesti podijeli s njegovom euklidskom (L2) normom. Na taj način se stvara jedinični vektor koji fokus stavlja na smjer vektora tj. interes korisnika.

```
user_sum = sum(user_vector)
norm = np.linalg.norm(user_sum.toarray())
if norm > 0:
      user_profile[user_id] = user_sum / norm
else:
      user_profile[user_id] = user_sum
```

Profil svakog korisnika na ovaj način predstavlja njegove interese i navike čitanja, sadrži sve

teme, kategorije i riječi koje dominiraju u vijestima koje je dosad čitao u svrhu stvaranja personaliziranih i zadovoljavajućih preporuka.

3.2.3. Računanje sličnosti i generiranje preporuka

Nakon izgradnje profila korisnika i vijesti, potrebno je pronaći vijesti koje su najsličnije korisničkom profilu. Za svaki se par korisnika i vijesti računa kosinusna sličnost.

Već spomenuta kosinusna sličnost, izračunata izrazom (3.1), daje mjeru sličnosti između dva vektora predstavljenu kao veličina kuta između njih. Što je ta vrijednost veća (bliže 1), to su dvije stavke predstavljene vektorima sličnije.

$$kosinusna\ sličnost = \frac{A \cdot B}{||A||||B||} \tag{3.1}$$

U ovom slučaju, vektor A predstavlja vektor korisnika, dok vektor B predstavlja vektor vijesti.

```
user_vector = user_profile[user_id]
similarity = cosine_similarity(user_vector, news_profile).flatten()
```

Izračunom kosinusne sličnosti za sve korisnike i vijesti nastao je temelj izgradnje personaliziranih preporuka.

Kako bi se osigurala što veća preciznost preporuka, ključno je pobrinuti se da se ne preporučuju vijesti koje je korisnik već pročitao ili kliknuo. Zbog toga se sličnosti svih pregledanih vijesti korisnika postavljaju na nulu.

```
read = user_read.get(user_id, set())
for news_id in read:
      if news_id in news_dict:
            similarity[news_dict[news_id]] = 0
```

Polazeći iz pretpostavke da korisnik ne interagira s vijestima koje su mu manje interesantne, mogu se sa sigurnošću smanjiti sličnosti vijesti koje su prikazane korisniku, ali na koje nije

kliknuo. Tako se svim vijestima iz liste `unclicked_news`, smanjuje vrijednost sličnosti za pola.

```
unclicked = unclicked_news[user_id]
for news_unclicked in unclicked:
        if news_unclicked in news_dict:
            similarity[news_dict[news_unclicked]] *= 0.5
```

Konačno, sustav odabire *K* vijesti s najvišom vrijednosti kosinusne sličnosti i predlaže ih korisniku. U eksperimentima provedenim za ovaj rad najčešće se koristila vrijednost $K = 10$ stvarajući najboljih 10 preporuka.

Zbog **problema hladnog starta** kao poznatog nedostatka sustava za preporučivanje, svim se korisnicima koji nemaju povijest pregledanih vijesti preporučuje *K* najpopularnijih vijesti, što se može vidjeti i na slici (Sl. 3.2). Izračun vrijednosti popularnosti je dobiven pribrajanjem broja interakcija s pojedinim vijestima. One vijesti s najvišem broja klikova i pregleda se smatraju najpopularnijima.

```
if user_id not in user_profile:
    popular_list = []

    for news_id in popularity_dict:
        count = popularity_dict[news_id]
        popular_list.append((news_id, count))

    popular_list.sort(reverse=True, key=lambda x: x[1])

    top_popular = []
    for pair in popular_list[:top_K]:
        top_popular.append(pair[0])

    preporuke = news[news["news_id"].isin(top_popular)][["news_id", "title", "category", "subcategory"]]
    return preporuke
```

Sl. 3.2 Rješenje problema hladnog starta

Na ovaj način sustav omogućuje personalizirano preporučivanje vijesti prilagođeno interesima korisnika na temelju njihovih prethodnih interakcija s vijestima. Sustav je trenutnačno relativno jednostavan s obzirom na opsežnost i složenost MIND dataseta te ga je moguće nadograditi dodatnim metodama, kao što je uvođenje suradničkog filtriranja, dodatnih značajki i slično.

3.3. Implementacija

Za implementaciju ovog sustava korišten je programski jezik Python zbog njegove širine i fleksibilnosti, posebice u radu s podacima i strojnim učenjem. Tijekom razvijanja sustava korištene su sljedeće biblioteke i alati:

- `pandas` - učitavanje i obrada ulaznih podataka
- `numpy` - izvođenje numeričkih operacija i rad s vektorima
- `scikit-learn (sklearn)` - vektorizacija teksta metodom TF-IDF, izračun kosinusne sličnosti i normalizacija podataka
- `scipy.sparse` - rad s rijetkim matricama
- `datetime` - obrada i računanje vremenskih podataka u svrhu dodjeljivanja težina pročitanim vijestima

Sustav je osmišljen tako da koristi efikasne metode za pohranu i računanje podataka tijekom rada s velikim brojem ulaznih podataka i iteracija programa. Cjelokupna implementacija je razvijena u okruženju Jupyter Notebook, što je omogućilo lakšu analizu i vizualizaciju rezultata tijekom izrade sustava.

4. Evaluacija modela

Nakon izgradnje modela za preporuke, isti je potrebno evaluirati kako bi se procijenila njegova kvalitetu na zadanom skupu podataka.

U ovom slučaju, koristit će se isključivo *offline* evaluacija zbog jednostavnosti i brzine prilikom testiranja različitih parametara te njihovih performansi. *Online* evaluacija daje uvid u stvarno ponašanje korisnika u realnom vremenu, ali ju je zbog ograničenih resursa i vremena, u ovom trenutku, nemoguće ostvariti.

4.1. *Offline* evaluacija

Evaluacija je provedena na testnom skupu podataka koji čini **20% vremenski najnovijih zapisa** iz originalnog skupa `behaviors.tsv`.

Korištene mjere vrednovanja su:

- **Preciznost na *K*:** udio relevantnih vijesti među prvih *K* preporučenih vijesti

```
intersect = actual_news_ids.intersection(recommended_ids_set)

precision = len(intersect) / K
```

- **Odziv na *K*:** omjer preporučenih relevantnih vijesti u odnosu na ukupan broj relevantnih vijesti

```
if actual_news_ids:
    recall = len(intersect) / len(actual_news_ids)
else:
    recall = 0
```

- **Mjera F1:** uravnotežena sredina između preciznosti i odziva koja balansira te dvije mjere

```
if (precision + recall) > 0:
    f1 = 2 * (precision * recall) / (precision + recall)
else:
    f1 = 0
```

- **nDCG:** mjera koja vrednuje redoslijed preporuka tj. poziciju relevantnih vijesti na listi preporuka

Dobiveni rezultati evaluacije modela na testnom skupu s $K = 10$ prikazani na sljedećoj tablici (Tablica 4.1):

Tablica 4.1 Rezultati evaluacije modela

Metrika	**Rezultat**
Preciznost na 10	0,0013
Odziv na 10	0,0061
Mjera F1	0,0020
nDCG	0,0065

Ove vrijednosti ukazuju na činjenicu da sustav ima određenu sposobnost preporučivanja relevantnih vijesti, ali je ona, nažalost, očekivano niska. Posebno je niska vrijednost preciznosti, što je posljedica velikog broja dostupnih vijesti, a ograničenih informacija o povijesti korisnika. Naime, analiza podatkovnog skupa `behaviors.tsv` je pokazala da u pojedinačnoj impresiji korisnici u prosjeku kliknu na **samo 1,51 vijest** od njih u prosjeku **prikazanih 37,23**. Takav omjer predstavlja značajnu prepreku u stvaranju kvalitetnih i personaliziranih preporuka.

U okviru preporučivanja vijesti, izazovna je i količina novih vijesti svakoga dana te njihova aktualnost. Korisnici često čitaju vijesti kakve prethodno nisu čitali ili o kojima nemaju prethodnu povijest interesa, što predstavlja dodatnu prepreku u stvaranju preporuka, posebice sustavima koji koriste filtriranje temeljeno na sadržaju. Osim aktualnosti vijesti, velika raznolikost tema, naslova, kategorija i potkategorija vijesti u skupu MIND povećava kompleksnost zadatka. S obzirom na to da većina korisnika čita vrlo mali broj vijesti u usporedbi s njihovim ukupnim brojem, matrica interakcija korisnika i vijesti je izrazito rijetka, što opet smanjuje kvalitetu preporuka.

Važno je istaknuti da sustav postiže bolje performanse od nasumičnog odabira, kao što se da vidjeti iz tablice (Tablica 4.2),. Ovakvi rezultati potvrđuju korisnost filtriranja temeljenog

na sadržaju na skupu podataka MIND-small, čak i u izazovnim uvjetima s ograničenim povijesnim podacima i velikim brojem vijesti.

Tablica 4.2 Rezultati evaluacije u slučaju nasumičnog odabira *K* preporuka

Metrika	**Rezultat**
Preciznost na 10	0,0001
Odziv na 10	0,0002
Mjera F1	0,0001
nDCG	0,0004

Iako je u ovom radu korišten skup podataka MIND-small zbog ograničenih računalnih resursa, važno je napomenuti da kompletan skup podataka MIND sadrži znatno više podataka. Kompletan skup za učenje sadrži približno sto tisuća vijesti i sedamsto tisuća korisnika, što predstavlja dvostruko veći broj vijesti i čak četrnaest puta više korisnika u odnosu na manji skup. Prosječan broj prikazanih vijesti po korisniku u većem skupu iznosi oko 117, dok je prosječan broj klikova po korisniku približno 4,8, što je vrlo slično vrijednostima u manjem skupu. Ipak, veći broj korisnika i veća raznolikost povijesnih podataka bi vrlo vjerojatno doprinijeli poboljšanju rezultata i kvaliteti preporuka.

Upravo je u radu Wu i sur. [24] predstavljen kompletni MIND skup podataka koji se koristi za evaluaciju različitih modela sustava preporuka. Tako je evaluiran i **model NRMS** (*Neural News Recommendation with Multi-Head Self-Attention*) s najboljim rezultatima, **nDCG@10 = 0,4163**, ***Mean Reciprocal Rank* (MRR) = 0,3305** i ***Area Under the ROC Curve* (AUC) = 0,6776**. MRR mjeri koliko visoko je na listi preporuka prvi relevantni rezultat, dok AUC ocjenjuje sposobnost modela da razlikuje relevantne i nerelevantne vijesti. Model koristi višeslojni mehanizam pozornosti kako bi modelirao interese korisnika te koristi neuronsku reprezentaciju vijesti.

U ovom radu ostvarena je vrijednost **nDCG = 0,0065**, što pokazuje jasnu razliku u preciznosti preporuka između složenijih modela i jednostavnijih pristupa.

Usporedba rezultata našeg modela s naprednijim pristupima jasno pokazuje da postoji još mnogo mjesta za napredak. Ovaj model bi se svakako mogao poboljšati korištenjem naprednijih tehnika, poput hibridnih modela koji kombiniraju pristupe filtriranja temeljenog na sadržaju i suradničkog filtriranja, čime bi se umanjili problemi hladnog starta i sparsnosti podataka. Osim toga, korištenjem modela zasnovanih na dubokom učenju bi se omogućilo bolje razumijevanje složenijih uzoraka u podacima te učinkovitija konstrukcija profila korisnika. Takvi bi pristupi mogli značajno poboljšati preciznost preporuka i povećati zadovoljstvo korisnika.

4.2. Analiza preporuka za primjer korisnika

Radi kvalitetnijeg razumijevanja preporuka, u nastavku je analiziran konkretan primjer korisnika i njegovih preporuka.

Korisnik: U67221

Pregledom podataka o impresijama korisnika u skupu podataka za učenje, prikupljene su sljedeće informacije (Tablica 4.3) o vijestima s kojima je korisnik imao interakciju:

Tablica 4.3 Pregledane vijesti korisnika U67221

news_id	title	abstract	category	subcategory
N56753	Fox News contributor: 'Most likely' outcome is Trump doesn't run in 2020	Fox News contributor Christopher Hahn predicted that President...	news	newspolitics
N8952	Chance the Rapper on 'SNL': 3 Sketches You Have to See	If Saturday Night Live episodes were named like installments of Friends...	tv	humor

N48122	'Friends' Creator Reveals Why Brad Pitt Was 'Hesitant' To Guest Star Opposite Jennifer Aniston	Brad Pitt received an Emmy nomination for his guest appearance on 'Friends' …	movies	movienews
N61661	The most surprising confessions made by the Queen's dressmaker in her new tell-all book about life at Buckingham Palace	Angela Kelly began her post in 1994 as the Queen's...	lifestyle	lifestyleroyals
N61972	New details on why Meg Ryan, John Mellencamp split, plus more news	Meg Ryan and John Mellencamp's marriage plans may have prompted...	entertain ment	entertainment-celebrity
N52694	Disney is betting everything on its Disney+ streaming service	On Nov. 12, the company will launch Disney+, hoping...	finance	finance-companies
N33440	Moeller football coach resigns		sports	football_ncaa

Korisnik ima relativno široke interese, od sporta, preko financija i vijesti, pa sve do zabave, televizije i filmova. Ovakav širok spektar interesa može stvoriti raznolike i neočekivane preporuke.

Sustav ovome korisniku preporučuje sljedećih deset vijesti (Tablica 4.4):

Tablica 4.4 Deset preporučenih vijesti za korisnika U67221

news_id	title	abstract	category	subcategory
N31801	Joe Biden reportedly denied Communion at a South Carolina church because of his stance on abortion	Joe Biden has a complicated history with the Catholic Church.	news	newspolitics
N55189	'Wheel Of Fortune' Guest Delivers Hilarious, Off The Rails Introduction	We'd like to solve the puzzle, Pat: Blair Davis' loveless marriage…	tv	tvnews
N306	Kevin Spacey Won't Be Charged in Sexual Assault Case After Accuser Dies…		movies	movies-celebrity
N42620	Heidi Klum's 2019 Halloween Costume Transformation Is Mind-Blowing	But, Like, What Is It? You might say she's scary good at playing dress-up, because Heidi Klum's…	lifestyle	lifestylebuzz
N43142	Former NBA first-round pick Jim Farmer arrested in sex sting operation	Farmer, 55, was booked for trafficking a person for a commercial sex act.	sports	basketball_nba
N45794	Four flight attendants were arrested in Miami's airport …		news	newscrime
N55689	Charles Rogers, former Michigan	Charles Rogers, the former Michigan State football star...	sports	football_nfl

	State football, Detroit Lions star, dead at 38			
N33619	College gymnast dies following training accident in Connecticut	Melanie Coleman, 20, of Milford, was practicing on the bars when she suffered a spinal cord injury.	news	newsus
N53585	Rip Taylor's Cause of Death Revealed, Memorial Service Scheduled for Later This Month	The comedian died at the age of 84 last month.	tv	tvnews
N35729	Porsche launches into second story of New Jersey building, killing 2	The Porsche went airborne off a median in Toms River, causing it to crash into a red brick building.	news	newsus

Kao što se vidi iz priložene tablice, preporuke za korisnika U67221 u velikoj mjeri odražavaju njegove prethodne interese. Među preporučenim vijestima prevladavaju teme vezane uz sportove, predsjednike SAD-a, poznate osobe i razne nesreće, što je u skladu s njegovom poviješću čitanja. Ovakva usklađenost između korisničke povijesti i predloženog sadržaja upućuje na to da sustav temeljen na sadržaju uspješno prepoznaje obrasce interesa korisnika.

Iako preporuke za korisnika U67221 pokazuju određenu razinu sklada s njegovim prethodnim interesima, što potvrđuje korisnost filtriranja temeljenog na sadržaju, treba opet istaknuti da su kvantitativni rezultati evaluacije modela i dalje relativno skromni. Niske vrijednosti mjera mogu se velikim dijelom pripisati **neravnoteži između broja dostupnih vijesti i ograničenog broja pozitivnih korisničkih interakcija**.

Zaključak

U ovom radu izgrađen je sustav za preporučivanje vijesti temeljen na sadržaju, korištenjem skupa podataka MIND, i to njegove smanjenje varijante (MIND-small). Kroz teorijski pregled predstavljeni su najvažniji pristupi u području preporučiteljskih sustava, uključujući suradničko filtriranje, filtriranje temeljeno na sadržaju, preporuke temeljene na dubokom učenju te hibridni modeli. Osim pristupa, predstavljeni su *online* i *offline* tipovi evaluacije te njihove metrike. Na temelju analize strukture i sadržaja skupa podataka, kao i vremenskih ograničenja, odabrana je metoda filtriranja temeljena na sadržaju.

Analizom skupa podataka utvrđeni su izazovi poput izrazite neravnoteže između broja prikazanih i kliknutih vijesti, velikog broja unikatnih vijesti te ograničene korisničke povijesti. Ovi faktori negativno utječu na performanse modela, što je potvrđeno rezultatima (npr. niska preciznost i recall) u *offline* tipu evaluacije, ali istovremeno ukazuju na smjer budućih poboljšanja.

Unatoč ograničenjima, sustav je pokazao sposobnost prepoznavanja korisničkih interesa i generiranja preporuka koje u mnogim slučajevima odgovaraju stvarnim interesima korisnika. Analiza konkretnih preporuka za jednog korisnika nam je potvrdila usklađenost između preporuka i korisničkih preferencija.

Za budući rad, preporučuje se istražiti naprednije metode, uključujući hibridne modele i pristupe temeljene na dubokom učenju, kao i uvođenje evaluacije u stvarnom vremenu (*online*), kako bi se dodatno povećala točnost i relevantnost preporuka.

Literatura

[1] NVIDIA. Recommendation System. NVIDIA Glossary. https://www.nvidia.com/en-us/glossary/recommendation-system/ Pristupljeno 10. svibnja 2025.

[2] ThingSolver. Introduction to Recommender Systems. ThingSolver Blog. 13. siječnja 2021. https://thingsolver.com/blog/introduction-to-recommender-systems/ Pristupljeno 10. svibnja 2025.

[3] Wikipedia. Collaborative Filtering in Recommender Systems. Wikimedia Commons. https://upload.wikimedia.org/wikipedia/commons/2/2c/Collaborative_Filtering_in_Recommender_Systems.jpg Pristupljeno 10. svibnja 2025.

[4] D4 Data Science. Recommender Systems 101. D4 Data Science Blog. 22. srpnja 2016. https://d4datascience.com/2016/07/22/recommender-systems-101/ Pristupljeno 10. svibnja 2025.

[5] Koren, Y., Bell, R., Volinsky, C. Matrix Factorization Techniques for Recommender Systems. IEEE Computer, 42(8), 2009., str. 42-49.

[6] Murel, J., Kavlakoglu, E. Collaborative Filtering. IBM Think Topics. 21. ožujka 2024. https://www.ibm.com/think/topics/collaborative-filtering Pristupljeno 10. svibnja 2025.

[7] Wikipedia. Matrix Factorization (Recommender Systems). Wikipedia. https://en.wikipedia.org/wiki/Matrix_factorization_(recommender_systems) Pristupljeno 15. svibnja 2025.

[8] Murel, J., Kavlakoglu, E. Content-Based Filtering. IBM Think Topics. 21. ožujka 2024. https://www.ibm.com/think/topics/content-based-filtering Pristupljeno 15. svibnja 2025.

[9] Van Otten, N. Content-Based Recommendation System. Spot Intelligence Blog. 15. studenog 2023. https://spotintelligence.com/2023/11/15/content-based-recommendation-system/ Pristupljeno 15. svibnja 2025.

[10] Rosidi, N. Step-by-Step Guide to Building Content-Based Filtering. StrataScratch Blog. 16. veljače 2023. https://www.stratascratch.com/blog/step-by-step-guide-to-building-content-based-filtering/ Pristupljeno 15. svibnja 2025.

[11] Rozhavsky, V. Deep Learning Recommendations. Dynamic Yield. https://www.dynamicyield.com/lesson/deep-learning-recommendations/ Pristupljeno 20. svibnja 2025.

[12] Johnson, J. Deep Neural Network. BMC Blogs. 27. srpnja 2020. https://www.bmc.com/blogs/deep-neural-network/ Pristupljeno 20. svibnja 2025.

[13] Abacus AI. Deep Learning-Based Recommendation Systems. Medium. 31. ožujka 2020. https://medium.com/abacus-ai/deep-learning-based-recommendation-systems-learning-ai-cbf1fded3b7e Pristupljeno 20. svibnja 2025.

[14] SciForce. Deep Learning-Based Recommender Systems. Medium. 30. travnja 2021. https://medium.com/sciforce/deep-learning-based-recommender-systems-b61a5ddd5456 Pristupljeno 20. svibnja 2025.

[15] Burke, R. Hybrid Recommender Systems: Survey and Experiments. User Modeling and User-Adapted Interaction. 12, 4 (2002), str. 331-370.

[16] Chiang, J. 7 Types of Hybrid Recommendation System. Analytics Vidhya. 26. lipnja 2021. https://medium.com/analytics-vidhya/7-types-of-hybrid-recommendation-system-3e4f78266ad8 Pristupljeno 2. lipnja 2025

[17] Milvus. What Defines a Hybrid Recommender System and What Are Its Benefits? Milvus. https://milvus.io/ai-quick-reference/what-defines-a-hybrid-recommender-system-and-what-are-its-benefits Pristupljeno 2. lipnja 2025.

[18] Evidently AI. Evaluating Recommender Systems. Evidently AI. 14. veljače 2025. https://www.evidentlyai.com/ranking-metrics/evaluating-recommender-systems Pristupljeno 6. lipnja 2025.

[19] Chokhra, P. Evaluating Recommender Systems. Medium. 27. siječnja 2021. https://medium.com/nerd-for-tech/evaluating-recommender-systems-590a7b87afa5 Pristupljeno 6. lipnja 2025.

[20] Scheltema, N. Evaluating Recommender Models: Offline vs Online Evaluation. 9. srpnja 2024. Shaped AI Blog. https://www.shaped.ai/blog/evaluating-recommender-models-offline-vs-online-evaluation Pristupljeno 6. lipnja 2025.

[21] MS News. Introduction to Recommender Systems. GitHub. https://github.com/msnews/msnews.github.io/blob/master/assets/doc/introduction.md Pristupljeno 7. lipnja 2025.

[22] GeeksforGeeks. Understanding TF-IDF (Term Frequency-Inverse Document Frequency). GeeksforGeeks. 7. veljače 2025. https://www.geeksforgeeks.org/understanding-tf-idf-term-frequency-inverse-document-frequency/ Pristupljeno 10. lipnja 2025.

[23] GeeksforGeeks. One-Hot Encoding in Machine Learning. GeeksforGeeks. 7. veljače 2025. https://www.geeksforgeeks.org/machine-learning/ml-one-hot-encoding/ Pristupljeno 10 lipnja 2025.

[24] Wu, F., Qiao, Y., Chen, J.-H., Wu, C., Qi, T., Lian, J., Liu, D., Xie, X., Gao, J., Wu, W., Zhou, M. *MIND: A Large-scale Dataset for News Recommendation*, 2020, *Proceedings of the ACL 2020 System Demonstrations*, https://aclanthology.org/2020.acl-demos.37 Pristupljeno 19. 6. 2025.

Sažetak

Sustav za preporučivanje vijesti temeljen na velikom skupu podataka

Barbara Bobeta

Preporučiteljski sustavi dio su mnogih digitalnih platformi, ključni za personalizaciju sadržaja i poboljšanja korisničkog iskustva. U prvom dijelu rada obrađena je teorijska podloga ovih sustava i njihove evaluacije. U praktičnom dijelu razvijen je sustav za preporučivanje vijesti temeljen na sadržaju koristeći stvarni skup podataka MIND koji sadrži informacije o korisnicima i vijestima. Analizom podataka pojavili su se izazovi poput velike raznolikosti vijesti i malog broja pozitivnih korisničkih interakcija. Model koristi tekstne i kategorijske značajke vijesti za izgradnju korisničkih profila na temelju povijesti čitanja. Iako su rezultati evaluacije skromni zbog strukture skupa podataka, preporuke su u velikom broju slučajeva bile u skladu s interesima korisnika. Predložena su moguća poboljšanja korištenjem hibridnih modela i metoda dubokog učenja.

Ključne riječi: preporučiteljski sustavi, filtriranje temeljeno na sadržaju, skup podataka MIND, vijesti, kosinusna sličnost, evaluacija, preporuke, personalizacija

Summary

News Recommender System Based on a Large Dataset

Barbara Bobeta

Recommender systems are an integral part of many digital platforms, playing a key role in content personalization and enhancing the user experience. The first part of this thesis covers the theoretical foundation of these systems and their evaluation methods. In the practical part, a content-based news recommendation system was developed using the real-world MIND dataset, which contains information about users and news articles. Data analysis revealed challenges such as the high diversity of news content and the low number of positive user interactions. The model utilizes textual and categorical features of news articles to build user profiles based on reading history. Although evaluation results were modest due to the dataset structure, the recommendations were in many cases aligned with user interests. Possible improvements include the use of hybrid models and deep learning methods.

Keywords: recommender systems, content-based filtering, MIND dataset, news, cosine similarity, evaluation, recommendations, personalization