



Univerzitet u Nišu  
Elektronski fakultet



## **Sistem za preporuku restorana na osnovu web recenzija, lokacija i ocena upotrebom Web scraping-a**

Seminarski rad

Mentor: Doc. dr Miloš Bogdanović

Student: Iva Blagojević 2137

# Sadržaj

1. Uvod .....	3
2. Web Scraping .....	4
2.1 Proces Web scrapinga .....	4
2.2 Tipovi Web scrapera [2].....	5
2.3 Biblioteke za Web scraping.....	6
2.4 Izazovi u Web scraping-u.....	7
3. Sentiment analiza.....	8
3.1 Metode sentiment analize.....	9
3.2 Alati i biblioteke za analizu sentimenata.....	10
3.3 Izazovi analize sentimenata.....	10
4. Sistemi za preporuku .....	11
4.1 Tipovi sistema za preporuku [8].....	12
4.2 Eksplicitni i implicitni feedback .....	13
5. Implementacija sistema za preporuku restorana.....	15
5.1 Prikupljanje podataka.....	15
5.2 Preobrada podataka .....	16
5.3 Analiza sentimenata .....	16
5.4 Algoritam za preporuku .....	18
5.5 Korisnički interfejs .....	20
6. Zaključak .....	22
7. Reference .....	23

## 1. Uvod

Savremene digitalne platforme suočavaju korisnike sa ogromnom količinom informacija i velikim brojem dostupnih izbora. U takvom okruženju sistemi za preporuku imaju značajnu ulogu, jer pomažu korisnicima da brže pronađu sadržaj koji odgovara njihovim interesovanjima i potrebama. Ovi sistemi danas se široko primenjuju u različitim oblastima, kao što su elektronska trgovina, streaming servisi, društvene mreže i turističke platforme.

U oblasti ugostiteljstva, izbor restorana često zavisi od iskustava drugih korisnika, koja su najčešće dostupna u obliku tekstualnih recenzija na internet platformama. Ove recenzije sadrže vredne informacije o kvalitetu hrane, usluge, atmosferi i ukupnom utisku posetilaca. Analizom takvih podataka moguće je razviti sisteme koji korisnicima preporučuju restorane na osnovu sadržaja recenzija i njihovih preferencija.

Razvoj metoda obrade prirodnog jezika omogućio je efikasnu analizu tekstualnih podataka, uključujući izdvajanje značajnih termina, procenu značenja teksta i utvrđivanje emocionalnog tona komentara. Posebno važnu ulogu ima analiza sentimenta, koja omogućava procenu da li su korisnički utisci pozitivni, neutralni ili negativni. Kombinovanjem ovih tehnika sa pristupima sistema za preporuku moguće je dobiti preciznije i informativnije predloge za korisnike.

Cilj ovog rada je razvoj sistema za preporuku restorana zasnovanog na sadržaju tekstualnih recenzija. U radu se koriste tehnike web scrapinga za prikupljanje podataka, metoda TF-IDF za numeričku reprezentaciju teksta, kosinusna sličnost za pronalaženje sličnih restorana, kao i analiza sentimenta kako bi se u proces preporuke uključio i kvalitet korisničkih utisaka. Implementacija sistema realizovana je kroz interaktivnu aplikaciju koja omogućava korisnicima da na osnovu svojih preferencija dobiju personalizovane predloge restorana.

Rad je organizovan tako da se najpre daje pregled web scrapinga i načina prikupljanja podataka, zatim teorijski okvir analize sentimenta i sistema za preporuku, nakon čega se opisuje implementacija razvijenog sistema, analiza rezultata i zaključci o njegovoj primeni i mogućim unapređenjima.

## 2. Web Scraping

Web scraping je automatizovan proces prikupljanja velikih količina podataka sa web stranica. Smatra se jednim od najefikasnijih načina za ekstrakciju podataka sa interneta, jer omogućava brzo i sistematsko prikupljanje informacija iz različitih izvora. Zbog toga je postao važan alat kako za kompanije, tako i za pojedince. Posebno se koristi u istraživanju tržišta, generisanju potencijalnih klijenata u prodaji i marketingu, kao i za praćenje cena u konkurentskim granama poput maloprodaje i turizma.

Web scraping ima i značajnu ulogu u obezbeđivanju podataka za modele mašinskog učenja, čime direktno doprinosi razvoju sistema zasnovanih na veštačkoj inteligenciji. Na primer, prikupljene slike sa interneta mogu se koristiti za treniranje algoritama računarskog vida, tekstualni podaci za modele obrade prirodnog jezika, dok podaci o ponašanju korisnika mogu unaprediti sisteme za preporuku. Automatizacijom procesa prikupljanja i mogućnošću skaliranja na veliki broj izvora, web scraping omogućava formiranje obimnih i raznovrsnih skupova podataka, što je ključno za izgradnju pouzdanih i preciznih modela.

Ova tehnika je posebno korisna u situacijama kada javni sajt sa kog se prikupljaju podaci ne poseduje API ili pruža samo ograničen pristup podacima. U takvim slučajevima scraping predstavlja praktično rešenje za dobijanje potrebnih informacija direktno iz sadržaja stranice [1].

Web scraping se sastoji od dve glavne komponente:

- **Crawler (pretraživač):** algoritam zasnovan na veštačkoj inteligenciji koji „pretražuje“ internet i prati linkove kako bi pronašao potrebne podatke.
- **Scraper (ekstraktor):** alat koji je dizajniran da izdvoji identifikovane podatke sa web sajtova, pri čemu se njegova kompleksnost i način implementacije prilagođavaju obimu i zahtevima konkretnog projekta.

### 2.1 Proces Web scrapinga

Web scraperi mogu biti podešeni da prikupljaju sve podatke sa određene web stranice ili samo one informacije koje su korisniku zaista potrebne. U praksi je efikasnije jasno definisati koje podatke želimo da izdvojimo, jer se na taj način ubrzava proces prikupljanja i smanjuje količina nepotrebnih informacija. Ovakav pristup omogućava fokusirano i brže prikupljanje podataka, što je posebno značajno u projektima koji se oslanjaju na dalju obradu i analizu informacija.

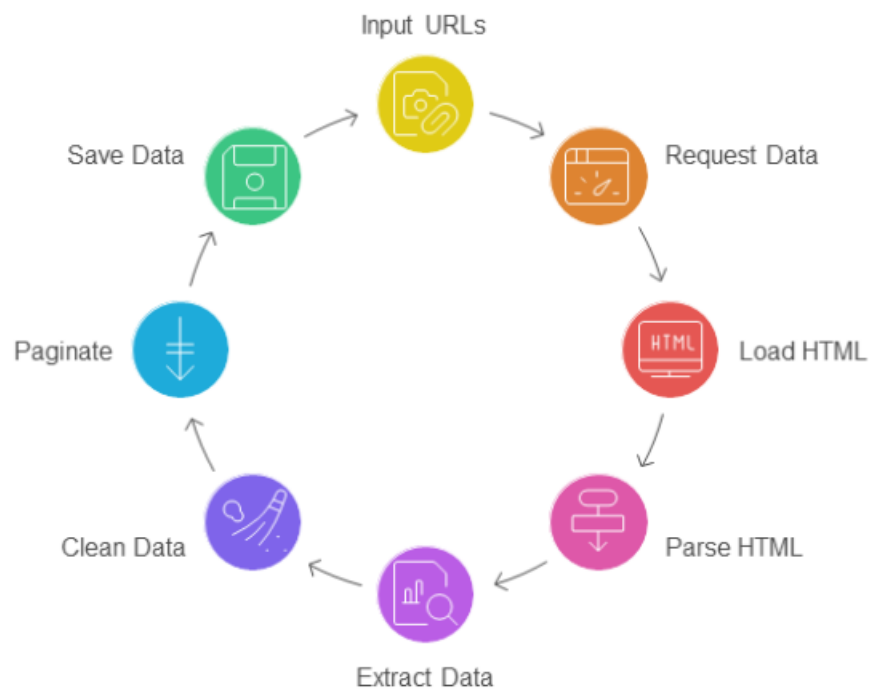
Proces Web scrapinga obuhvata sledeće korake:

1. **Slanje zahteva ka sajtu** - Prvi korak u web scrapingu je slanje HTTP zahteva ka željenoj web stranici. Ovo se obično radi pomoću biblioteka koje podržavaju HTTP protokol, kao što su *requests* ili *http.client* u Pythonu. Server zatim vraća sadržaj stranice, obično u HTML formatu.
2. **Preuzimanje i parsiranje HTML-a** - Nakon što je HTML sadržaj preuzet, potrebno ga je parsirati, tj. pretvoriti u strukturu koja se može lako analizirati. Biblioteke kao što su *BeautifulSoup*, *lxml* ili *html.parser* omogućavaju izdvajanje željenih elemenata iz HTML koda, kao što su naslovi, tekst, tabele ili linkovi.
3. **Identifikacija i ekstrakcija podataka** - Sledeći korak je identifikacija specifičnih podataka koji su relevantni za projekat. Na primer, to mogu biti recenzije korisnika, ocene,

cene proizvoda ili slike. Alati za ekstrakciju podataka (scraper-i) omogućavaju selektovanje i izdvajanje ovih elemenata pomoću CSS selektora, XPath izraza ili regularnih izraza.

4. **Čuvanje i obrada podataka** - Prikupljeni podaci se potom čuvaju u formatima pogodnim za dalju analizu, kao što su CSV, JSON ili baze podataka. Nakon toga se često vrši dodatna obrada, čišćenje podataka i normalizacija kako bi bili spremni za upotrebu u modelima mašinskog učenja ili u sistemima za preporuke.
5. **Automatizacija i skaliranje** - Za veće projekte ili kontinuirano prikupljanje podataka, web scraping se može automatizovati i skalirati, koristeći skripte koje periodično preuzimaju nove podatke, ili korišćenjem crawlera koji automatski prolazi kroz linkove na sajtu kako bi prikupio sve relevantne informacije.

Primena ovakvog procesa omogućava kreiranje bogatih i raznovrsnih dataset-a, koji se mogu koristiti za različite analize, uključujući analizu sentimenta, preporuke i istraživanje tržišta.



Slika 1

## 2.2 Tipovi Web scrapera [2]

U zavisnosti od načina implementacije, okruženja u kome rade i nivoa složenosti, web scraperi se mogu podeliti u više kategorija. Razumevanje različitih tipova scrapera važno je kako bi se izabrao odgovarajući pristup za konkretan projekat i potrebe prikupljanja podataka. Među najčešće tipove ubrajaju se sopstveno razvijeni scraperi (*Self-built*), unapred napravljeni alati (*Pre-built*), ekstenzije za pregledače, samostalni softverski scraperi, scraperi sa različitim korisničkim interfejsima, kao i cloud i lokalni scraperi.

## **Sopstveno razvijeni scraperi**

Uz odgovarajuće znanje programiranja, korisnik može samostalno razviti web scraper. Nivo potrebnog znanja zavisi od složenosti funkcionalnosti koje se žele postići. Nakon usvajanja potrebnih veština, scraper se može razviti korišćenjem programskih jezika kao što je Python. Sa druge strane, moguće je koristiti i unapred razvijene scrapere koji se jednostavno preuzmu i pokrenu. Ovi alati često nude dodatne funkcionalnosti, poput izvoza podataka u različite formate (npr. tabele ili JSON).

## **Ekstenzije za pregledače i softverski scraperi**

Ekstenzije za web scraping instaliraju se u internet pregledače poput Chrome-a ili Firefox-a. Iako su jednostavne za korišćenje, njihove mogućnosti su ograničene jer funkcionišu unutar pregledača, što otežava implementaciju složenijih funkcija.

Nasuprot tome, samostalni softver za scraping instalira se direktno na računar korisnika. Iako nije integrisan u pregledač, ovakav softver nudi veću fleksibilnost i naprednije opcije obrade podataka.

## **Korisnički interfejs web scrapera**

Web scraperi mogu imati različite tipove korisničkog interfejsa. Neki alati nude minimalan interfejs i oslanjaju se na komandnu liniju, što može otežati razumevanje procesa za manje iskusne korisnike.

Drugi alati imaju razvijen grafički interfejs koji omogućava korisniku da direktno označi elemente sa web stranice koje želi da izdvoji. Postoje i napredniji sistemi koji pružaju preporuke, objašnjenja i smernice, čineći proces prikupljanja podataka intuitivnijim.

## **Cloud i lokalni web scraperi**

Web scraper može raditi lokalno na računaru ili na udaljenom serveru u oblaku. Lokalni scraper koristi resurse korisnikovog računara — procesor, memoriju, internet konekciju — što može usporiti rad sistema i zauzeti značajne resurse. Takođe, veće količine preuzetih podataka mogu brzo dostići ograničenja internet saobraćaja. Kod cloud-baziranog scrapinga, obrada se izvršava na serverima u oblaku, koje obično obezbeđuje kompanija koja je razvila alat. Na taj način računar korisnika nije opterećen procesom, a scraping može da se odvija brže i efikasnije.

Iako se web scraperi mogu razlikovati po načinu rada i okruženju u kome funkcionišu, njihova realizacija u praksi najčešće se zasniva na upotrebi specijalizovanih programskih biblioteka i alata. U nastavku su predstavljene najčešće korišćene biblioteke za web scraping, kao i njihove osnovne karakteristike.

## **2.3 Biblioteke za Web scraping**

U praktičnoj implementaciji web scrapinga najčešće se koriste specijalizovane programske biblioteke koje omogućavaju slanje zahteva ka web stranicama, parsiranje HTML sadržaja i

izdvajanje željenih podataka. Izbor biblioteke zavisi od strukture stranice, prisustva dinamičkog sadržaja i složenosti zadatka.

Jedna od najčešće korišćenih biblioteka za parsiranje HTML dokumenata je **Beautiful Soup**. Ova biblioteka omogućava jednostavno pronalaženje elemenata u HTML strukturi koristeći oznake, klase ili hijerarhiju elemenata. Često se koristi u kombinaciji sa bibliotekom *requests* za slanje HTTP zahteva. Idealna je za manje projekte ili stranice sa statičkim sadržajem, gde se sadržaj učitava direktno iz HTML koda.

U slučajevima kada web stranica koristi JavaScript za dinamičko učitavanje sadržaja, potrebno je koristiti naprednije alate. Jedan od najpoznatijih je **Selenium**, koji omogućava automatizaciju rada u web pregledaču. Selenium može simulirati ponašanje korisnika, poput klikova, skrolovanja i popunjavanja formi, čime omogućava prikupljanje podataka i sa kompleksnih, dinamičkih stranica.

Još jedan značajan alat za web scraping je **Scrapy**, framework razvijen u programskom jeziku Python. Scrapy se posebno izdvaja po sposobnosti da efikasno podrži kompleksnije i obimnije scraping projekte. Za razliku od jednostavnijih biblioteka koje služe samo za preuzimanje i parsiranje stranica, Scrapy je osmišljen kao celovito rešenje koje omogućava automatsko praćenje linkova, obradu paginacije i poštovanje pravila definisanih u *robots.txt* fajlu. Jedna od ključnih prednosti ovog framework-a jeste njegova robusnost i fleksibilnost.

Ipak, u poređenju sa jednostavnijim alatima, Scrapy zahteva nešto složenije početno podešavanje i bolje razumevanje strukture projekta, zbog čega se češće koristi u naprednijim sistemima za prikupljanje podataka [3].

## 2.4 Izazovi u Web scraping-u

Savremeni izazovi u web scrapingu proističu iz činjenice da vlasnici web stranica imaju snažan interes da zaštite svoje podatke. Na primer, e-commerce platforme nastoje da spreče konkurenciju da automatski prikuplja informacije za potrebe tržišne analize, društvene mreže žele da monetizuju pristup podacima putem zvaničnih servisa, dok portali sa sadržajem nastoje da onemoguće neovlašćeno preuzimanje i ponovno objavljivanje tekstova.

Kako bi se zaštitile od automatizovanog prikupljanja podataka, web stranice koriste više slojeva zaštite. Jedan od njih je *fingerprinting* pregledača, pri čemu se prikuplja veliki broj karakteristika uređaja i softvera kako bi se formirao jedinstven digitalni otisak korisnika. Ovaj otisak može identifikovati automatizovane alate čak i kada koriste različite IP adrese. Pored toga, mnoge stranice primenjuju analizu ponašanja korisnika. Ovaj pristup prati način navigacije, brzinu klikanja, skrolovanje i druge obrasce interakcije kako bi se procenilo da li se radi o stvarnom korisniku ili automatizovanom programu.

Značajnu ulogu u zaštiti imaju i mreže za isporuku sadržaja, poput **Cloudflare**, koje mogu detektovati neuobičajene obrasce saobraćaja karakteristične za scraping i blokirati takve zahteve.

Dodatni problem predstavlja i česta promena strukture web stranica. Vlasnici sajtova redovno menjaju nazive klasa, identifikatore elemenata i strukturu DOM-a, zbog čega scraperi moraju stalno da se prilagođavaju kako bi ostali funkcionalni.

Jedan od najpoznatijih mehanizama zaštite jeste upotreba **CAPTCHA** testova. CAPTCHA sistemi zahtevaju od korisnika da potvrdi da nije robot izvršavanjem zadataka kao što su prepoznavanje objekata na slici, označavanje određenih polja ili rešavanje jednostavnih slagalica. Savremeni sistemi, poput **reCAPTCHA**, dodatno analiziraju ponašanje korisnika i procenjuju

verovatnoću da se radi o automatizovanom programu. Ovakve metode predstavljaju značajnu prepreku za web scraping [4].

Uprkos navedenim izazovima, postoje pristupi koji omogućavaju stabilnije i pouzdanije prikupljanje podataka. Jedan od njih je slanje zahteva kontrolisanom brzinom kako bi se izbeglo opterećenje servera i smanjila verovatnoća blokiranja. Takođe, u projektima se često koriste alati koji mogu da simuliraju ponašanje pregledača i time obezbede realističniju interakciju sa web stranicom.

Kada je moguće, preporučuje se korišćenje zvaničnih API servisa koje pojedini sajtovi nude, jer oni predstavljaju pouzdan i etički način pristupa podacima. Na taj način se obezbeđuje stabilnost sistema i poštuju pravila korišćenja web stranica.

Zbog svega navedenog, web scraping danas predstavlja složen proces koji zahteva pažljiv izbor alata, strategija i pristupa kako bi se obezbedilo pouzdano i odgovorno prikupljanje podataka.

### **3. Sentiment analiza**

Nakon prikupljanja podataka metodama web scrapinga, naredni korak u analizi jeste njihova obrada i interpretacija. Veliki deo prikupljenih informacija sa weba ima tekstualnu formu, poput korisničkih komentara, recenzija ili opisa usluga. Kako bi se iz takvih podataka izvukli korisni zaključci, primenjuju se tehnike obrade prirodnog jezika (NLP), među kojima značajno mesto zauzima sentiment analiza.

Analiza sentimenata predstavlja postupak automatskog određivanja emocionalnog tona teksta, odnosno procene da li je stav izražen u tekstu pozitivan, negativan ili neutralan. Ova metoda omogućava pretvaranje nestrukturiranih tekstualnih podataka u kvantitativne pokazatelje koji se dalje mogu koristiti u analizi korisničkog zadovoljstva, istraživanju tržišta ili razvoju sistema za preporuku.

Pre nego što se primene metode sentiment analize, prikupljeni tekstualni podaci se standardno pripremaju kroz osnovnu predobradu, koja uključuje čišćenje nepotrebnih karaktera, tokenizaciju i uklanjanje stop reči. Detalji ove predobrade biće predstavljeni u poglavlju koje opisuje implementaciju sistema i korišćene podatke.



Slika 2

Na Slici 2 prikazan je tipičan životni ciklus analize sentimenta koji je primenjen i u ovom radu. Proces započinje unosom sirovog teksta (*Text Input*), nakon čega sledi faza predobrade. Ova faza obuhvata tokenizaciju (razbijanje teksta na pojedinačne reči) i filtriranje stop-reči (*Stop Word Filtering*) kako bi se uklonili jezički elementi koji ne nose emocionalno značenje. Dalja obrada uključuje naprednije NLP tehnike poput stemming-a (svođenje reči na njihov koren) i obrade negacija (*Negation Handling*), što je ključno za ispravno prepoznavanje konteksta (npr. razlika između 'dobro' i 'nije dobro'). Finalni koraci obuhvataju klasifikaciju podataka, čime se svakoj recenziji dodeljuje pripadajuća klasa sentimenta (*Sentiment Class*) – pozitivna, negativna ili neutralna.

### 3.1 Metode sentiment analize

Sentiment analiza se može realizovati različitim metodama, koje se generalno dele na leksikonske pristupe i pristupe zasnovane na mašinskom učenju [5]:

**1. Leksikonski pristup** - koristi unapred definisane liste reči sa pozitivnim, negativnim ili neutralnim tonom. Algoritam prebrojava pojavljivanja ovih reči u tekstu i određuje ukupni sentiment. Ovaj pristup je jednostavan za implementaciju i često daje dobre rezultate na manjim skupovima podataka ili kada su tekstovi relativno kratki i direktni. Međutim, može biti ograničen u slučaju složenih izraza, ironije ili konteksta koji menja značenje reči.

**2. Mašinsko učenje** - Pristupi zasnovani na mašinskom učenju koriste modele koji se treniraju na označenim skupovima podataka. Tipični modeli uključuju logističku regresiju, Naive Bayes,

SVM (*Support Vector Machines*), pa sve do naprednih neuronskih mreža i transformera. Ovi modeli mogu prepoznati složenije obrasce i kontekstualne informacije, što ih čini pogodnim za analizu većih količina podataka i tekstova sa raznovrsnim stilom izražavanja.

**3. Kombinovani pristupi i moderni alati** - U praksi se često kombinuju leksikonski i mašinski pristupi kako bi se postigla veća preciznost. Takođe, savremeni alati i biblioteke, poput **NLTK**, **TextBlob** ili **Transformers** (*Hugging Face*), omogućavaju jednostavnu implementaciju i eksperimentisanje sa različitim metodama sentiment analize.

### 3.2 Alati i biblioteke za analizu sentimenata

Za analizu sentimenta u tekstualnim podacima koriste se različiti alati i biblioteke, koje omogućavaju kako osnovnu leksikonsku analizu, tako i sofisticirane modele dubokog učenja. U okviru ovog rada fokus je stavljen na Python biblioteke i unapred trenirane modele koji se često koriste u istraživačkim i praktičnim projektima [6]:

- **NLTK (Natural Language Toolkit)** - jedna od najpoznatijih Python biblioteka za obradu prirodnog jezika. Pruža alate za tokenizaciju, lematizaciju, uklanjanje stop reči i kreiranje leksikonskih modela sentimenta. NLTK je pogodan za istraživanje jezika i razvoj prototipova, a u kombinaciji sa sopstvenim rečnicima ili dodatnim modelima može služiti za osnovnu klasifikaciju sentimenta.
- **SpaCy** - brza i efikasna biblioteka fokusirana na proizvodne NLP sisteme. Pruža preciznu tokenizaciju, lematizaciju i prepoznavanje entiteta u tekstu. SpaCy je posebno koristan kada je potrebno raditi sa velikim datasetovima, jer je optimizovan za performanse i može se kombinovati sa unapred treniranim modelima za dodatne zadatke, uključujući klasifikaciju sentimenta.
- **TextBlob** - pojednostavljuje obradu teksta i analizu sentimenta. Omogućava procenu polariteta (od -1 do +1) i subjektivnosti (od 0 do 1) teksta, što ga čini pogodnim za brzo i jednostavno vrednovanje mišljenja korisnika. TextBlob je praktičan za prototipove i male datasetove, i često se koristi u kombinaciji sa NLTK za predobradu i tokenizaciju.
- **VADER (Valence Aware Dictionary and sEntiment Reasoner)** - leksikonski model optimizovan za kratke tekstove i recenzije, naročito u kontekstu društvenih mreža. Dobro prepoznaje intenzitet emocija, kolokvijalni jezik i emotikone. VADER je jednostavan za implementaciju i daje precizne rezultate za realne tekstualne podatke sa recenzija i komentara.
- **BERT i srodni transformeri** - za naprednu sentiment analizu koriste se modeli dubokog učenja, poput **BERT**, **RoBERTa** i **DistilBERT**. Ovi modeli omogućavaju razumevanje konteksta reči u tekstu i prepoznavanje složenih izraza ili ironije, što leksikonski modeli često ne mogu da obrade. U radu se mogu koristiti za fino podešavanje (fine-tuning) na dataset recenzija kako bi se dobila što preciznija klasifikacija sentimenta.

### 3.3 Izazovi analize sentimenata

Iako se tehnologije za analizu sentimenta brzo razvijaju, ovo polje je još uvek relativno novo. Postoji još mnogo izazova koje treba prevazići kako bi se poboljšala preciznost ovih metoda. Neki od najčešćih problema uključuju [7]:

- **Nedostatak konteksta** - kontekst je ključan za razumevanje emocija izraženih u tekstu i često izaziva greške kod alata za sentiment analizu. Na primer, na anketi korisnik može odgovoriti na pitanje: „Šta vam se svidelo u našoj aplikaciji?” sa odgovorima „funktionalnost” i „UX”. Ako bi pitanje glasilo „Šta vam se nije svidelo u našoj aplikaciji?”, značenje odgovora bi se promenilo bez promene reči. Da bi algoritam ispravno razumeo kontekst, potrebno je uključiti originalno pitanje ili situaciju, što je često složen i dugotrajan proces.
- **Upotreba ironije i sarkazma** - bez obzira na nivo obuke modela, softver teško prepoznaje ironiju i sarkazam u tekstu. Ovo je zato što se ironija ili sarkazam često prenose tonom glasa ili facijalnom ekspresijom, a reči same po sebi ne daju jasan signal. Na primer, analiza fraze „Super, još jedna kazna za parking od hiljadu dolara – baš mi je trebala” bi mogla biti pogrešno označena kao pozitivna zbog prisustva reči „super”.
- **Negacija** - negacija menja značenje rečenice korišćenjem negativnih reči. Na primer, rečenica „Ne bih rekao da su cipele jeftine” zapravo implicira da su cipele verovatno skupe ili bar srednje skupe, ali algoritmi za sentiment analizu često ne prepoznaju ovu nijansu.
- **Idiomatski izrazi** - idiomi i uobičajene fraze, kao što su „Ne okolišajmo” ili „Srećno na sceni” (eng. „*Break a leg*”), često zbunjuju algoritme. Kada se ovakve fraze koriste u recenzijama ili na društvenim mrežama, alati za analizu sentimenta mogu ih pogrešno protumačiti – na primer, „Break a leg” može biti shvaćeno doslovno kao nešto negativno – ili ih uopšte ne prepoznati.

Ovi izazovi pokazuju da, iako sentiment analiza pruža korisne uvide, njena preciznost i pouzdanost zavise od kvaliteta predobrade podataka, metode analize i konteksta u kome se tekst pojavljuje.

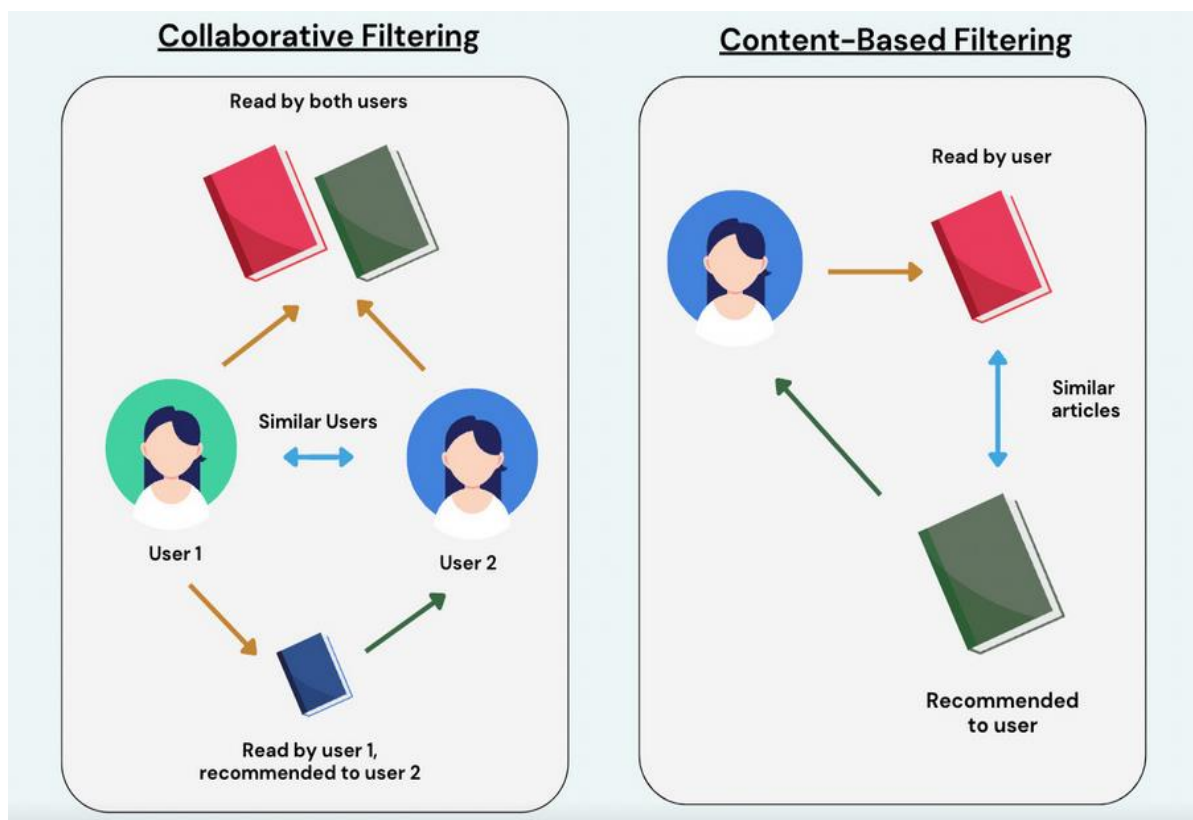
## 4. Sistemi za preporuku

Sistemi za preporuku predstavljaju jednu od ključnih tehnologija u savremenim digitalnim platformama, koji korisnicima omogućavaju da otkriju proizvode, usluge ili sadržaje koji su relevantni za njihove interese i potrebe. Sa sve većim obimom podataka dostupnih na internetu, korisnici se često suočavaju sa problemom preopterećenja informacijama, što otežava donošenje odluka. Sistemi za preporuku rešavaju ovaj problem tako što automatski filtriraju i prezentuju najrelevantnije opcije, čime poboljšavaju korisničko iskustvo i olakšavaju pronalaženje kvalitetnog sadržaja.

Osnovna funkcija preporučivača je da poveže korisnika sa stavkama koje bi mu mogle biti zanimljive, koristeći informacije o njegovim prethodnim interakcijama, preferencijama ili obrascima ponašanja sličnih korisnika. Ovi sistemi se koriste u širokom spektru primena, uključujući preporuke filmova i muzike, predloge članaka, restorana, turističkih destinacija, pa čak i personalizovane obrazovne sadržaje.

Razumevanje principa na kojima funkcionišu sistemi za preporuku je ključno za njihov razvoj i primenu. U literaturi se uglavnom izdvajaju dve osnovne kategorije preporučivača: **content-based sistemi**, koji koriste karakteristike stavki i preferencije korisnika, i **collaborative filtering sistemi**, koji analiziraju obrasce ponašanja korisnika i njihove međusobne sličnosti. Ova podela omogućava bolje razumevanje prednosti i ograničenja različitih pristupa, a u praksi se često kombinuju kako bi se obezbedila što relevantnija i preciznija personalizacija.

Sistemi za preporuku postali su neizostavan deo digitalnog okruženja i jedan od temelja modernih aplikacija zasnovanih na podacima, jer omogućavaju optimizaciju interakcije između korisnika i velikih datasetova, pružajući personalizovano iskustvo u realnom vremenu.



Slika 3

## 4.1 Tipovi sistema za preporuku [8]

Sistemi za preporuku se najčešće dele u tri glavne kategorije: **content-based pristup**, **collaborative filtering** i **hibridni pristup**. Svaka metoda ima svoje prednosti i nedostatke, a izbor metode zavisi od dostupnih podataka, ciljeva sistema i željenog nivoa personalizacije.

### 1. Content-based pristup

**Content-based (CB) sistemi** fokusiraju se na karakteristike stavki i preferencije korisnika. Algoritmi analiziraju attribute proizvoda ili usluge – kao što su tip kuhinje, lokacija, ocene ili opis – i kreiraju korisnički profil zasnovan na prethodnim interakcijama. Preporuke se zatim generišu pronalaženjem stavki koje su slične onima koje je korisnik ranije pozitivno ocenio.

Prednosti CB pristupa uključuju:

- Ne zavisi od drugih korisnika i može raditi sa malim brojem interakcija.
- Omogućava jasnu interpretaciju zašto je stavka preporučena.

Ograničenja uključuju:

- Teško se preporučuju potpuno nove stavke koje korisnik još nije ocenjivao.
- Sistem može postati previše fokusiran na slične stavke, što smanjuje raznovrsnost preporuka.

## 2. Collaborative filtering pristup

**Collaborative filtering (CF) sistemi** se zasnivaju na analizi obrazaca ponašanja korisnika i sličnosti između korisnika ili stavki. Dve osnovne vrste CF pristupa su:

- **User-based CF:** preporučuje stavke koje su slični korisnici već ocenili pozitivno.
- **Item-based CF:** preporučuje stavke koje su slične onima koje je korisnik prethodno ocenio.

Prednosti CF pristupa uključuju:

- Može otkriti stavke koje korisnik možda ne bi sam otkrio.
- Dobro funkcioniše za velike datasetove sa mnogo korisničkih ocena.

Ograničenja uključuju:

- Zahteva dovoljan broj korisničkih ocena da bi bio efikasan (problem *hladnog starta*).
- Ponekad može generisati preporuke koje su teško interpretirati.

## 3. Hibridni pristup

**Hibridni sistemi** kombinuju prednosti content-based i collaborative filtering metoda kako bi se smanjili njihovi nedostaci. Na primer, hibridni pristup može koristiti CB algoritam za rešavanje problema hladnog starta (novi korisnici ili stavke) i CF za poboljšanje raznovrsnosti i otkrivanje neočekivanih stavki.

Prednosti hibridnog pristupa:

- Veća preciznost preporuka zahvaljujući kombinaciji više izvora informacija.
- Smanjenje problema hladnog starta i povećanje raznovrsnosti preporuka.

Hibridni pristupi su danas standard u većini komercijalnih sistema za preporuke, jer omogućavaju balans između personalizacije, raznovrsnosti i skalabilnosti sistema.

## 4.2 Eksplicitni i implicitni feedback

Sistemi za preporuku prikupljaju informacije o korisnicima kako bi generisali personalizovane predloge. Ove informacije se mogu podeliti u dve glavne kategorije - **eksplicitni** i **implicitni feedback**.

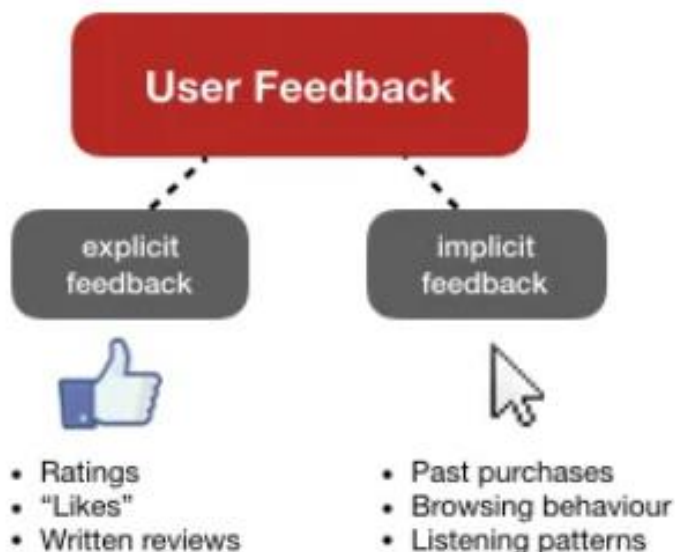
**Eksplicitni feedback** predstavlja ocene ili komentare koje korisnik direktno daje kako bi izrazio svoje zadovoljstvo određenom stavkom. Primeri uključuju:

- ocene u obliku zvezdica na skali od 1 do 5 nakon kupovine proizvoda,
- „thumb up/down“ posle gledanja videa ili slušanog sadržaja.

Prednost eksplicitnog feedback-a je što pruža precizne i direktne informacije o korisnikovim preferencijama. Njegov nedostatak je što je teško prikupiti, jer većina korisnika ne ostavlja ocene ili recenzije za svaku stavku sa kojom dolazi u interakciju.

**Implicitni feedback**, s druge strane, oslanja se na pretpostavku da interakcije korisnika sa stavkama ukazuju na njihove preferencije. Primeri uključuju: istoriju kupovina, preglede proizvoda, listu pesama koju je korisnik slušao ili klikove na određene sadržaje. Implicitni feedback je veoma obilno dostupan, ali je manje precizan i može biti „bučan“ – na primer, neko može kupiti proizvod kao poklon, pa sistem može pogrešno pretpostaviti da korisnik lično preferira tu stavku. Ipak, zbog ogromnog obima dostupnih podataka, većina modernih sistema za preporuku danas se više oslanja na implicitni feedback kako bi generisala relevantne predloge.

Razumevanje razlike između eksplicitnog i implicitnog feedback-a je važno za dizajn sistema za preporuku, jer direktno utiče na izbor algoritama, metode predobrade podataka i strategije za kreiranje korisničkog profila [9].



Slika 4

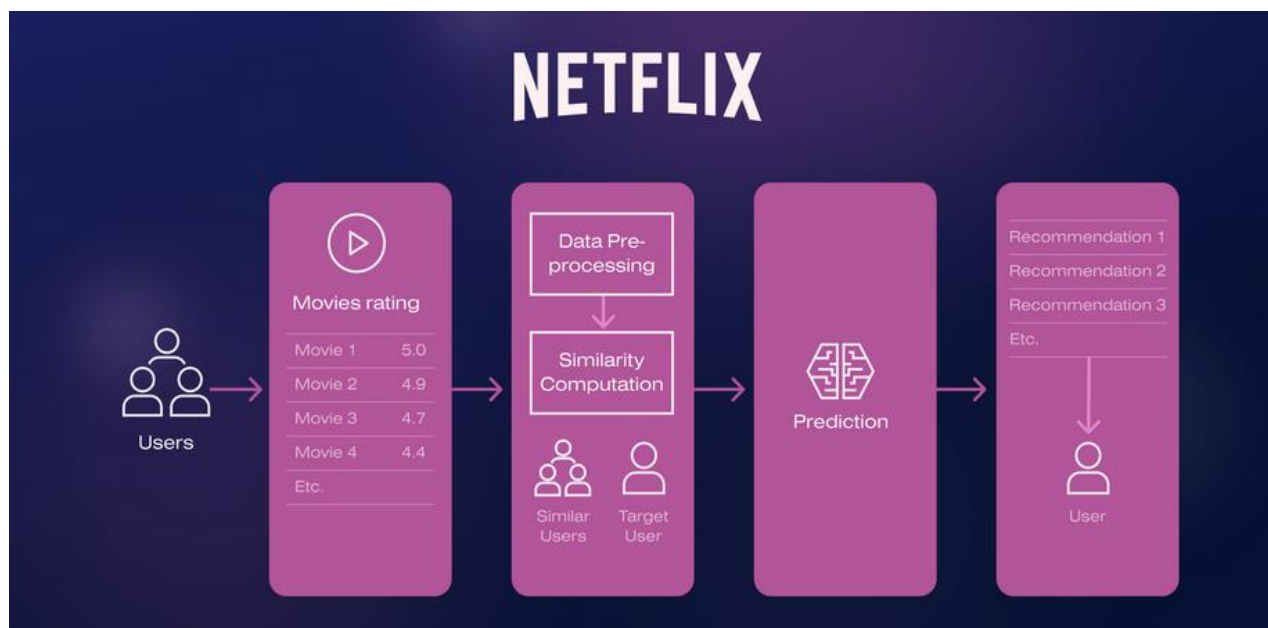
### Primer praktične primene: Netflix Cinematch [10]

Netflix-ov sistem preporuka, poznat kao **Cinematch**, jedan je od najsofisticiranijih u svetu digitalnih servisa. Sistem kombinuje **content-based** i **collaborative filtering** pristupe kako bi korisnicima pružio personalizovane preporuke.

Kada korisnik započne korišćenje platforme, unosi informacije o svojim omiljenim filmovima i serijama. Na osnovu tih podataka kreira se početni feed koji se dalje prilagođava korišćenjem **eksplicitnog feedback-a** (ocene, dodavanje sadržaja u listu omiljenih) i **implicitnog feedback-a** (vreme provedeno gledajući sadržaj, redosled gledanja, učestalost gledanja).

Sistem takođe koristi podatke o grupama korisnika sa sličnim interesovanjima i segmentira ih u više od 2000 segmenata. To znači da i aktivnosti drugih korisnika sa sličnim ukusima utiču na preporuke. Dodatno, sistem uzima u obzir demografske podatke, geografski položaj i vremenski kontekst (npr. sezonski sadržaj), kako bi preporuke bile što relevantnije.

Ovaj primer pokazuje kako veliki sistemi za preporuke koriste kombinaciju različitih metoda i izvora podataka da bi generisali precizne i personalizovane preporuke za svakog korisnika.



Slika 5

## 5. Implementacija sistema za preporuku restorana

Praktična realizacija sistema obuhvata proces od prikupljanja sirovih podataka sa weba do razvoja interaktivne aplikacije koja krajnjem korisniku pruža personalizovane preporuke restorana. Implementacija je organizovana kroz nekoliko ključnih faza: prikupljanje podataka putem web scrappinga, predobradu tekstualnih podataka, analizu sentimenta i razvoj algoritma za preporuku.

U nastavku rada svaka od ovih faza biće detaljno opisana, uz prikaz korišćenih alata, metoda i načina na koji su međusobno povezane u funkcionalan sistem.

### 5.1 Prikupljanje podataka

Razvijen je namenski web scraper u programskom jeziku Python, koristeći biblioteku **Selenium** i njen napredni modul **undetected-chromedriver**. Izbor ovih tehnologija omogućio je automatizovano prikupljanje podataka sa platforme *TripAdvisor*, uz uspešno zaobilazanje anti-bot sistema koji štite sajt.

Radi dodatnog prikrivanja automatizovanog ponašanja, implementirana je funkcija *human\_delay*, kojom su uvedene nasumične pauze između 12 i 22 sekunde, kao i simulacija skrolovanja stranice pomoću JavaScript komande *execute\_script*. Na ovaj način smanjena je mogućnost da platforma prepozna skriptu kao automatizovani alat.

Scraper je konfigurisan da prolazi kroz listu od šest evropskih metropola (London, Pariz, Rim, Beograd, Madrid, Berlin), pri čemu je za svaki grad obrađeno po petnaest najrelevantnijih restorana. Tokom ovog procesa prikupljeni su podaci o nazivima restorana, gradovima, korisničkim ocenama i tekstualnim recenzijama. Poseban fokus bio je na prikupljanju tekstualnih komentara korisnika, jer oni predstavljaju ključni izvor informacija za kasniju analizu sentimenta i izgradnju sistema preporuke.

Za ekstrakciju podataka korišćeni su **XPATH** i **CSS** selektori, čime je omogućeno precizno pronalaženje naziva restorana, ocena i sadržaja recenzija. Kako bi se obezbedio kvalitetniji skup podataka za dalju obradu, zadržane su samo recenzije duže od 70 karaktera. Svi prikupljeni podaci (ime restorana, grad, ocena i tekst recenzije) su u realnom vremenu čuvani u .csv fajl, čime je formirana baza podataka spremna za dalju obradu u *Pandas* biblioteci.

## 5.2 Preobrada podataka

Nakon prikupljanja podataka, sledeća faza implementacije odnosila se na pripremu tekstualnih recenzija za dalju analizu. S obzirom na to da su podaci preuzeti sa web stranica nestrukturirani i često sadrže šum, bilo je neophodno sprovesti postupak predobrade kako bi se obezbedio kvalitetan ulaz za modele analize sentimenta i algoritam za preporuku.

U okviru predobrade izvršeno je uklanjanje duplikata, nedostajućih vrednosti, kao i osnovno čišćenje tekstualnih podataka. Recenzije su normalizovane kako bi se uklonile nepravilnosti koje mogu negativno uticati na rezultate analize, poput suvišnih razmaka ili praznih zapisa. Iz recenzija su uklonjeni specijalni karakteri (poput emodžija) i nepotrebni prelazi u novi red. Tekst je enkodovan u ASCII format kako bi se standardizovali karakteri i olakšala dalja obrada.

Dodatno, u okviru predobrade sprovedene su standardne tehnike obrade prirodnog jezika kako bi se unapredio kvalitet tekstualnih podataka. Definisan je skup "zabranjenih fraza" (npr. *Sign in*, *Sponsored*, *Thank you for sharing*) kako bi se iz baze uklonili generički odgovori menadžera restorana, reklame i sistemske poruke TripAdvisora koje ne sadrže stvarni opis iskustva korisnika.

Posebna pažnja posvećena je uklanjanju stop reči, odnosno čestih reči poput „the“, „and“, „is“, koje ne doprinose značenju recenzije, ali mogu negativno uticati na model. Ovaj postupak realizovan je korišćenjem unapred definisanih lista stop reči dostupnih u NLP bibliotekama.

Zadržani su samo komentari duži od 50 karaktera, čime su eliminisani kratki i neinformativni unosi (poput "Super", "Great place"), koji nemaju dovoljno težine za kvalitetnu TF-IDF analizu.

Ovaj korak predstavlja ključnu pripremu za naredne faze sistema, jer kvalitet predobrade direktno utiče na tačnost sentiment analize i relevantnost preporuka koje sistem generiše.

## 5.3 Analiza sentimenta

Nakon predobrade, tekstualne recenzije su analizirane radi određivanja sentimenta, odnosno emocionalnog tona izraženog u korisničkim komentarima. Za ovu fazu korišćen je Python biblioteka **TextBlob**, koja omogućava automatsko izračunavanje sentimenta na osnovu polariteta rečenica.

Svaka recenzija je transformisana u numerički skor (*polarity*), gde pozitivne vrednosti označavaju pozitivno iskustvo korisnika, negativne vrednosti označavaju nezadovoljstvo, a vrednosti blizu nule predstavljaju neutralan ton. Na osnovu dobijenog skor-a, recenzije su svrstane u tri kategorije: **Pozitivna**, **Neutralna** i **Negativna**.

Za potrebe analize, izvršena je agregacija prosečnog sentimenta po gradovima, što omogućava identifikaciju gradova sa najpozitivnijim i najnegativnijim recenzijama. Konačni skup podataka, koji uključuje originalne recenzije, ocene, grad i dobijeni *Sentiment\_Score*, sačuvan je u CSV formatu, spreman za dalju obradu u algoritmu za preporuku.

```
def analiziraj_tekst(tekst):  
    tekst = str(tekst)
```

```

blob = TextBlob(tekst)

return blob.sentiment.polarity

df['Sentiment_Score'] = df['Recenzija'].apply(analiziraj_tekst)

def kategorija(skor):
    if skor > 0.1:
        return 'Pozitivna'
    elif skor < -0.1:
        return 'Negativna'
    else:
        return 'Neutralna'

df['Emocija'] = df['Sentiment_Score'].apply(kategorija)

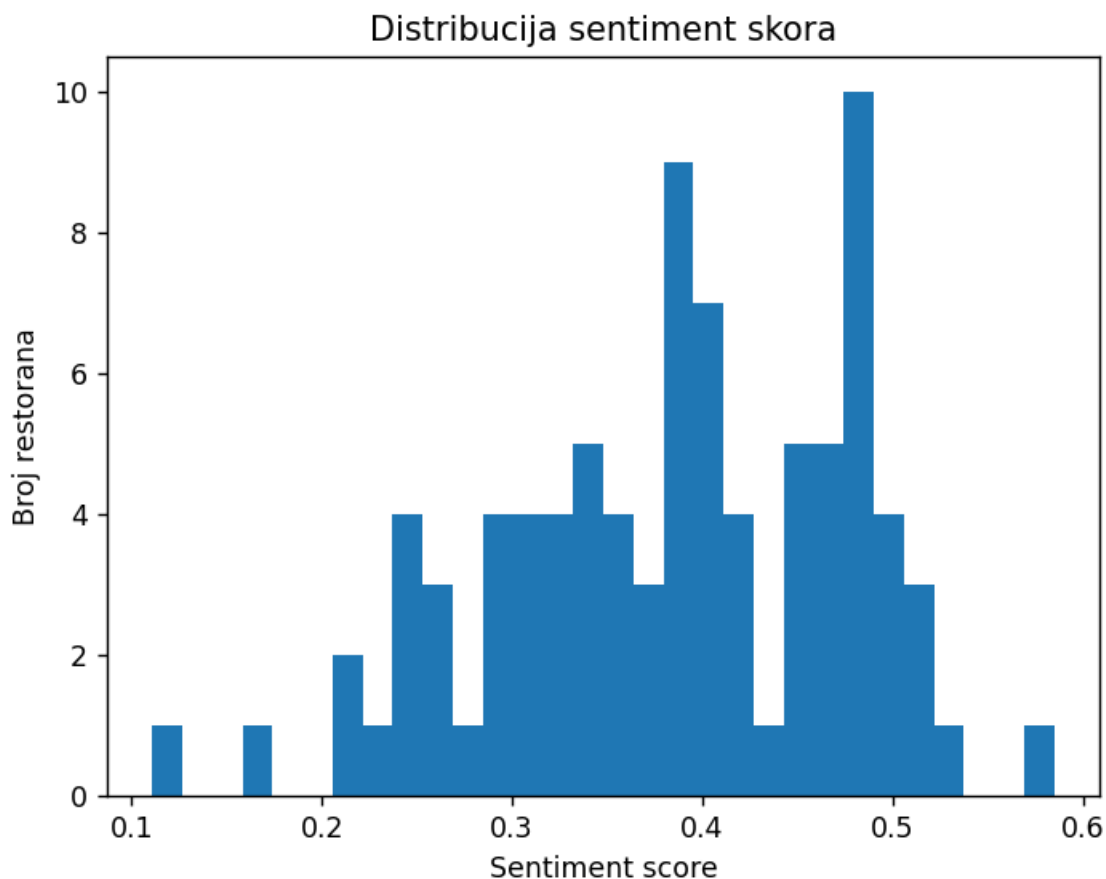
```

### Analiza distribucije sentimenta

Nakon proračuna sentimenta za sve restorane, kreiran je histogram koji prikazuje distribuciju dobijenih rezultata. Ovaj grafik je važan iz nekoliko razloga:

- **Dominacija pozitivnog sentimenta:** Većina restorana ima sentiment skor u rasponu od 0.3 do 0.5, što ukazuje na to da su recenzije u bazi pretežno pozitivne.
- **Identifikacija ekstrema:** Jasno se vide restorani sa veoma visokim sentimentom (blizu 0.6), koji će sistemu biti prioritet pri preporuci, kao i oni sa nižim skorom (oko 0.1), koji će biti niže rangirani.
- **Gustina podataka:** Najveći broj restorana (njih 10) ima vrednost oko 0.48, što predstavlja standard kvaliteta u prikupljenom skupu podataka.

Ovaj grafik dokazuje da TextBlob uspešno pravi razliku između restorana, što je neophodno da bi sistem preporuke mogao da funkcioniše precizno.



Slika 6

## 5.4 Algoritam za preporuku

Sistem za preporuku zasnovan je na **content-based pristupu**, kombinovanom sa analizom korisničkog profila. Ideja je da se korisniku predlože restorani koji su slični onima koje je ranije izabrao, uz dodatno filtriranje prema preferencijama u vezi sa vrstom hrane. Takođe, korisnik može da bira željeni grad.

Za kreiranje korisničkog profila korišćen je **TF-IDF (Term Frequency–Inverse Document Frequency)** model, koji omogućava kvantifikaciju značaja svake reči u tekstualnim recenzijama. Na osnovu odabranih omiljenih restorana i izabranih tipova hrane, sistem računa prosečni TF-IDF vektor, koji predstavlja preferencije korisnika. Izbor specifične kuhinje ima za 50% veći uticaj (\* 1.5) na finalni rezultat, kako bi preporuka bila što relevantnija trenutnom upitu.

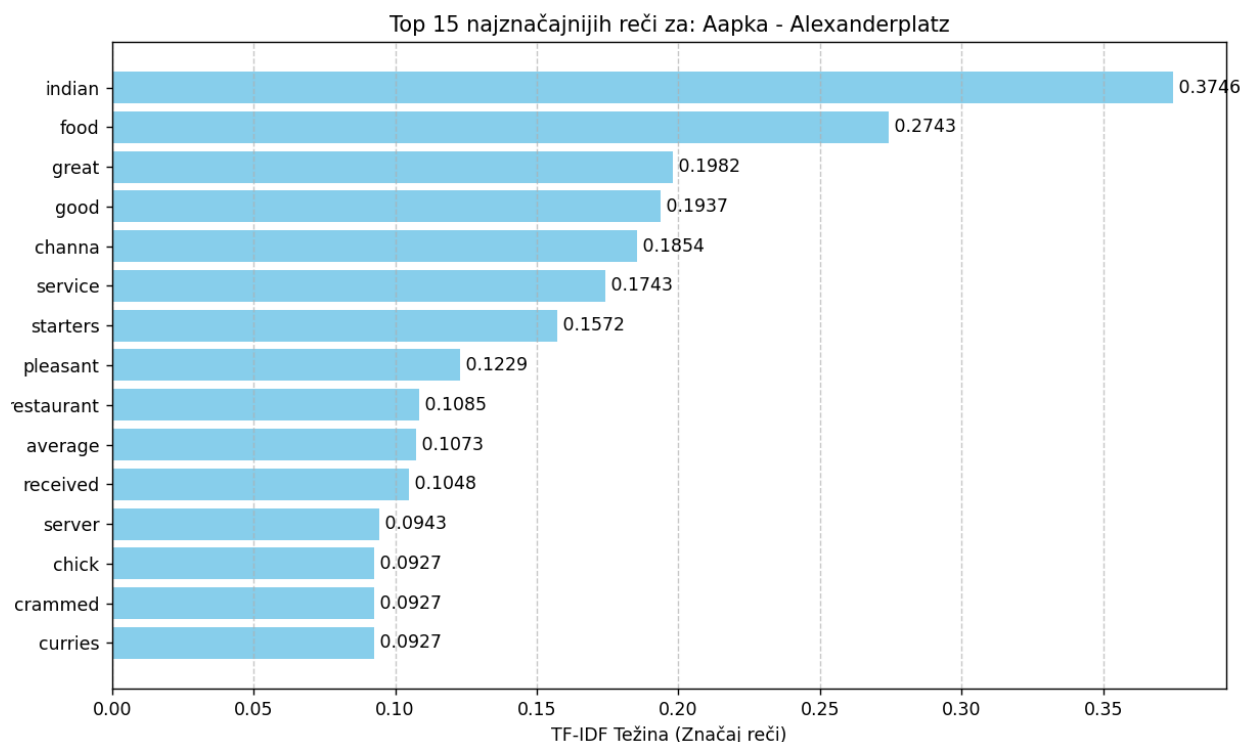
Sličnost između korisničkog profila i svih restorana u bazi izračunava se pomoću **cosine similarity**, koja meri stepen podudarnosti između vektora. Restorani se zatim rangiraju prema kombinovanom kriterijumu **sličnosti i prosečne ocene sentimenta recenzija**, čime se osigurava da korisniku budu prikazani restorani koji su i sadržajno relevantni i visoko ocenjeni od strane prethodnih posetilaca.

Sistem ne koristi samo sličnost, već vrši **sekundarno rangiranje** pomoću ranije izračunatog Sentiment\_Score-a. To znači da će, između dva podjednako slična restorana, sistem uvek na prvo mesto staviti onaj koji ima pozitivnije recenzije.

```
# Kreiranje TF-IDF matrice
tfidf = TfidfVectorizer(stop_words='english')
tfidf_matrix = tfidf.fit_transform(df['opis_zaj_algoritam'])

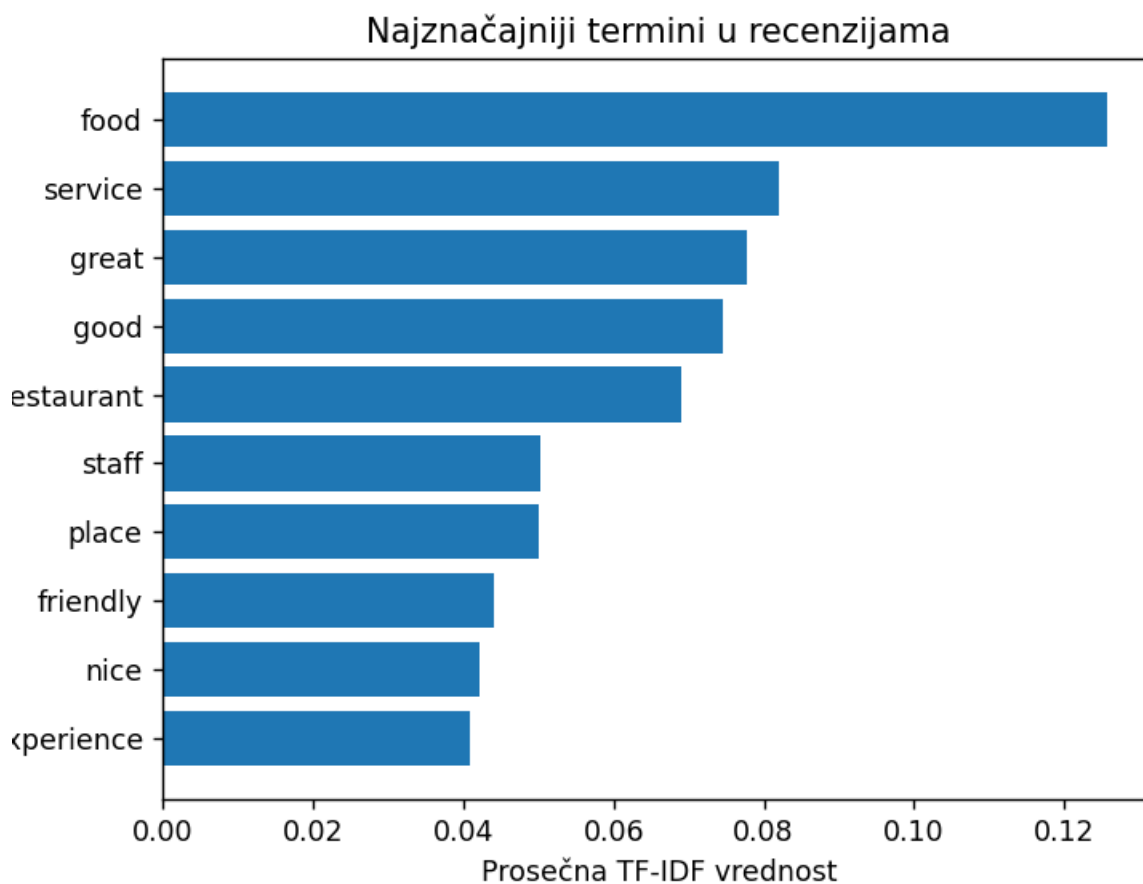
# Generisanje korisničkog profila
vektor_restorana = tfidf_matrix[omiljeni_indeksi].mean(axis=0)
vektor_kuhinje = tfidf.transform([tekst_kuhinje]).toarray()
korisnicki_profil = np.asarray(vektor_restorana) +
np.asarray(vektor_kuhinje) * 1.5

# Računanje sličnosti i rangiranje restorana
slicnost_skorovi = cosine_similarity(korisnicki_profil,
tfidf_matrix).flatten()
df['Slicnost'] = slicnost_skorovi
```



Slika7

Grafik prikazuje 15 najznačajnijih reči iz recenzija restorana *Aapka - Alexanderplatz* prema TF-IDF analizi. Na X-osi je prikazana TF-IDF težina koja označava značaj svake reči u kontekstu recenzija, dok Y-osa prikazuje same reči. Reči poput *indian*, *food*, *great* i *good* imaju najveću težinu, što ukazuje na to da se često pojavljuju u recenzijama i nose značajnu informaciju o sadržaju i kvalitetu restorana.



Slika 8

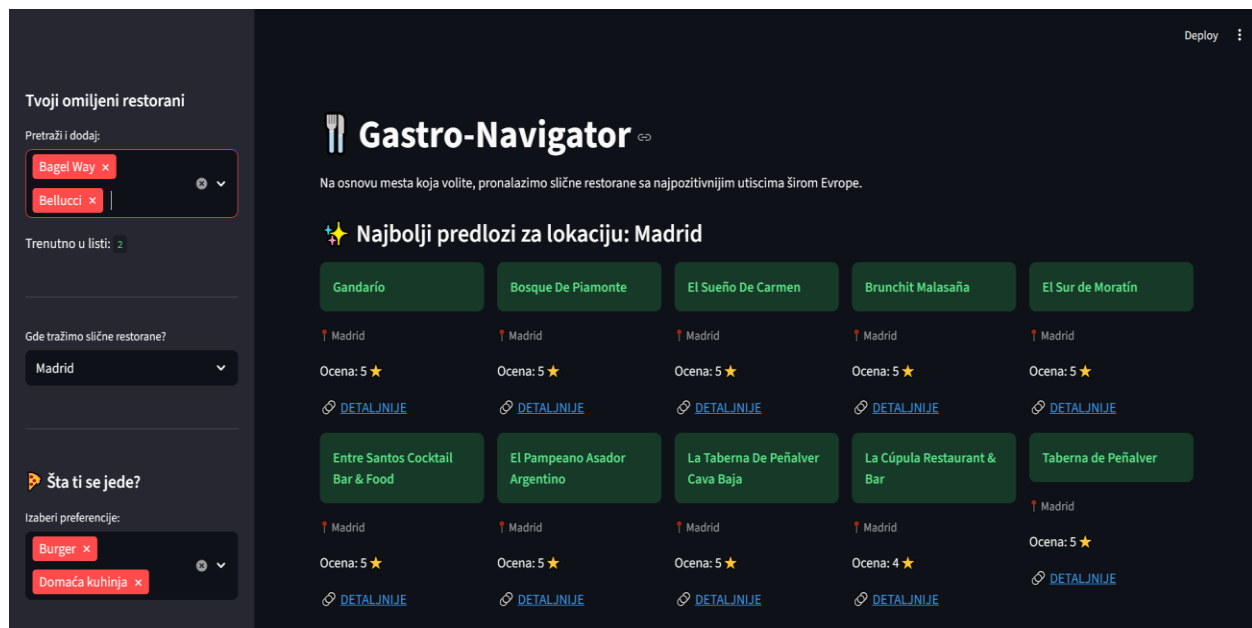
Grafik prikazuje najznačajnije termine iz svih recenzija restorana prema prosečnoj TF-IDF vrednosti. Na X-osi je prikazana prosečna TF-IDF vrednost, dok Y-osa prikazuje najčešće i najznačajnije reči u recenzijama. Reči poput *food*, *service*, *great*, *good* i *restaurant* dominiraju, što ukazuje na ključne aspekte koje korisnici najviše komentarišu i koji najviše doprinose karakterizaciji iskustva u restoranima.,

## 5.5 Korisnički interfejs

Pored algoritamskog dela, razvijen je interaktivni korisnički interfejs koristeći biblioteku **Streamlit**. Korisniku je omogućeno da:

- izabere omiljene restorane i tipove hrane,
- filtrira preporuke po gradu,
- dobije interaktivne kartice sa nazivom restorana, ocenom i direktnim linkom ka Google pretrazi za više detalja..

Vizuelni prikaz omogućava intuitivno korišćenje aplikacije, dok algoritam u pozadini osigurava personalizovane i relevantne preporuke.



Slika 9

## 6. Zaključak

Istraživanje je imalo za cilj da pokaže koliko tekstualne recenzije korisnika mogu doprineti razumevanju kvaliteta restorana i koliko su pogodne za izgradnju personalizovanih sistema preporuke.

Rezultati rada ukazuju da primena metoda obrade prirodnog jezika, zajedno sa TF-IDF reprezentacijom teksta i merom kosinusne sličnosti, omogućava efikasno prepoznavanje sličnosti između restorana. Uvođenje sentiment analize dodatno poboljšava kvalitet preporuka, jer sistem ne procenjuje samo tematsku povezanost restorana, već i opšti ton korisničkih iskustava. Razvijena aplikacija potvrđuje da se i uz pristupačne metode i relativno jednostavne modele može realizovati sistem koji pruža smislen i upotrebljiv skup preporuka.

Iako ostvareni rezultati pokazuju funkcionalnost i primenljivost sistema, postoje brojne mogućnosti za njegovo dalje unapređenje. U budućem radu preporučuje se proširenje baze podataka većim brojem gradova, restorana i recenzija, čime bi se povećala robusnost modela i smanjio uticaj pojedinačnih odstupanja u podacima. Takođe, sentiment analiza mogla bi se unaprediti primenom savremenijih modela zasnovanih na dubokom učenju, koji bolje prepoznaju kontekst, ironiju i nijanse u jeziku.

Dalji razvoj sistema mogao bi uključiti i kombinovanje postojećeg content-based pristupa sa kolaborativnim filtriranjem, čime bi se omogućilo uzimanje u obzir ponašanja većeg broja korisnika i njihovih međusobnih sličnosti. Pored toga, uvođenje dodatnih faktora, kao što su tip kuhinje, cenovni rang, lokacija ili sezonske preferencije korisnika, moglo bi doprineti preciznijim i realističnijim preporukama.

Na osnovu sprovedenog istraživanja može se zaključiti da analiza tekstualnih recenzija predstavlja vredan izvor informacija za izgradnju sistema preporuke, kao i da ovakav pristup ima značajan potencijal primene u savremenim digitalnim servisima, naročito u oblastima turizma i ugostiteljstva, gde personalizacija sadržaja ima sve veći značaj.

## 7. Reference

- [1] "What Is Web Scraping And How Does It Work? | Zyte.com," [Online]. Available: <https://www.zyte.com/learn/what-is-web-scraping/>.
- [2] "What Is Web Scraping & How Is It Used?," [Online]. Available: <https://www.fortinet.com/resources/cyberglossary/web-scraping>.
- [3] A. Yadav, "Scrapy vs BeautifulSoup vs Selenium," 10 8 2024. [Online]. Available: <https://medium.com/@amit25173/scrapy-vs-beautifulsoup-vs-selenium-579bce149262>.
- [4] K. Sahin, "Top Web Scraping Challenges in 2026," 5 1 2026. [Online]. Available: <https://www.scrapingbee.com/blog/web-scraping-challenges/>.
- [5] C. Dilmegani, "Sentiment Analysis Methods Overview, Pros & Cons," 4 4 2024. [Online]. Available: <https://research.aimultiple.com/sentiment-analysis-methods/>.
- [6] C. Dilmegani, "Top 7 Open Source Sentiment Analysis Tools in 2025," 10 6 2025. [Online]. Available: <https://research.aimultiple.com/open-source-sentiment-analysis/>.
- [7] IBM, "Sentiment Analysis," 24 8 2023. [Online]. Available: <https://www.ibm.com/think/topics/sentiment-analysis>.
- [8] "Collaborative Filtering Vs Content-Based Filtering," 2025. [Online]. Available: [https://www.meegle.com/en\\_us/topics/recommendation-algorithms/collaborative-filtering-vs-content-based-filtering](https://www.meegle.com/en_us/topics/recommendation-algorithms/collaborative-filtering-vs-content-based-filtering).
- [9] F. Casalegno, "Recommender Systems — A Complete Guide to Machine Learning Models," 25 11 2022. [Online]. Available: <https://medium.com/data-science/recommender-systems-a-complete-guide-to-machine-learning-models-96d3f94ea748>.
- [10] <https://www.facebook.com/andriy.khomyn.50>, "Product Recommendation System in E-Commerce: How Does It Work? - SoloWay Tech," 30 7 2023. [Online]. Available: <https://soloway.tech/blog/implementing-machine-learning-in-e-commerce-personalization-and-recommendation-systems/>.