

Inteligentni sustav za preporuku za kupce auomobila

Jakov, Kristović

Master's thesis / Diplomski rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Organization and Informatics / Sveučilište u Zagrebu, Fakultet organizacije i informatike**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:211:570281>

Rights / Prava: [Attribution-NoDerivs 3.0 Unported](#) / [Imenovanje-Bez prerada 3.0](#)

Download date / Datum preuzimanja: **2024-07-29**



Repository / Repozitorij:

[Faculty of Organization and Informatics - Digital Repository](#)



**SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN**

Jakov Kristović

**INTELIGENTNI SUSTAV ZA PREPORUKU
ZA KUPCE AUTOMOBILA**

DIPLOMSKI RAD

Dubrovnik, 2020.

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ž D I N

Jakov Kristović

JMBAG: 0016115984

Diplomski studij: Baze podataka i baze znanja

INTELIGENTNI SUSTAV ZA PREPORUKU ZA KUPCE
AUTOMOBILA

DIPLOMSKI RAD

Mentor:

Prof. dr. sc. Božidar Kliček

Dubrovnik, travanj 2020.

Jakov Kristović

Izjava o izvornosti

Izjavljujem da je moj završni rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor potvrdio prihvaćanjem odredbi u sustavu FOI-radovi

Sažetak

U ovom diplomskom radu dat će se uvid u što su sustavi za preporuke te koje različite mogućnosti oni nude za korisnika takvog sustava kao i za samog pružatelja usluge. Obradit će se različiti izvori znanja koje takvi sustavi mogu koristiti kao i različiti pristupi kojima mogu raditi preporuke korisnicima. U nastavku se daje pregled tehnologija sustava za preporuku, od onih najjednostavnijih rješenja pa do inteligentnih sustava. Napravit će se analiza postojećih sustava za prodaju automobila. Nakon toga će se izraditi vlastiti prototip inteligentnog sustava za preporuku za kupce automobila, gdje će se prvo objasniti prikupljanje podataka te postupak kreiranja vlastitog testnog skupa podataka s kojim će prototip raditi. U nastavku se objašnjavaju koraci profiliranja korisnika s nekoliko različitih vrsta pitanja kako bi se dobio željeni korisnički profil za daljnju obradu nad skupom podataka, te naposljetku kreirati dvije različite komponente, odnosno algoritma za predlaganje automobila kupcima. Prethodno izradi pojedine komponente, dat će se pregled korištenih tehnologija, kao i ključne smjernice izrade same komponente.

Prva komponenta predviđanja je bazirana na jednostavnom algoritmu utemeljenom na bodovanju i faktorizaciji atributa automobila, u skladu s profiliranim preferencijama korisnika. Druga komponenta će uključivati treniranje podataka i izradu modela strojnog učenja putem ML .NET tehnologije. Nakon izrade prototipa inteligentnog sustava s dvije različite komponente, odnosno dvaju različitih algoritama inteligentnog rasuđivanja, rad tih komponenti će se prokomentirati kao i dobiveni rezultati. Na kraju rada slijedit će zaključak, popis literature te popis slika, videa i tablica korištenih prilikom izrade ovog rada.

Ključne riječi: inteligentni sustav; sustav za preporuku; automobil; prototip; ML .NET; strojno učenje; profiliranje korisnika.

Sadržaj

1. Uvod	5
2. Sustav za preporuku	6
2.1 Različite mogućnosti sustava za preporuku	7
2.1.1 Mogućnosti za pružatelja usluge	8
2.1.2 Mogućnosti za korisnika sustava.....	9
2.2 Izvori podataka i znanja	12
2.2.1 Predmeti	12
2.2.2 Korisnici	13
2.2.3 Transakcije - odnosi između korisnika i predmeta.....	15
2.3 Pristupi kod preporučivanja	18
2.3.1 Preporučivanje na temelju sadržaja.....	18
2.3.2 Kolaborativno filtriranje.....	20
2.3.3 Demografsko preporučivanje	21
2.3.4 Preporučivanje temeljeno na znanju.....	22
2.3.5 Preporučivanje temeljeno na zajednici.....	23
2.3.6 Hibridni sustavi za preporuke.....	25
3. Pregled tehnologija sustava za preporuku	26
3.1 Tehnologije utemeljene na memoriji (dostupnim podacima)	26
3.1.1 TF-IDF	27
3.1.2 Klasteriranje	28
3.1.3 KNN algoritam.....	29
3.1.4 Teorija grafova	30
3.2 Tehnologije utemeljene na modelima podataka	32
3.2.1 Matrična faktorizacija	33
3.2.2 Regresijska analiza.....	35
3.2.3 Stabla odlučivanja	36
3.2.4 Bayesovi klasifikatori.....	37
3.2.5 Bayesove mreže.....	39
3.2.6 Neuronske mreže.....	41
4. Analiza postojećih sustava za prodaju automobila.....	44
4.1 Provedena analiza.....	44
4.2 Rezultati analize	48
5. Izrada prototipa inteligentnog sustava za preporuku automobila.....	50

5.1 Prikupljanje podataka	50
5.2 Profiliranje korisnika.....	53
5.3 ML .NET	56
5.3.1 Izrada komponente strojnog učenja.....	57
5.3.2 Predikcija pomoću strojnog učenja	60
5.4 Algoritam bodovanja.....	61
5.5 Primjer rada prototipa sustava za preporuku automobila.....	64
6. Zaključak.....	67
7. Popis literature.....	69
8. Popis slika i video materijala	72
9. Popis tablica	73

1. Uvod

Pretraživanjem knjiga na Amazonu ili pregledom postova na Facebooku, mnogi korisnici se koriste i sustavom za preporuke a da za to možda nisu ni znali. Uz mogućnost kupnje putem interneta, potrošači imaju gotovo beskonačan izbor, a nitko od njih nema dovoljno vremena da isproba svaki proizvod koji je dostupan u prodaji. Sustavi preporuka igraju važnu ulogu u pomaganju korisnicima da pronađu proizvode ili dođu do sadržaja koji su im bitni, ali još važniju ulogu kod poduzeća koja ih koriste.

Sustavi za preporuku proizvoda danas dobivaju sve više na značaju kod online poslovanja poduzeća, gdje mogućnost predlaganja nove usluge, proizvoda ili sadržaja koji će se svidjeti korisniku može značajno povećati prodaju. Gotovo svaki online shop će korisniku prilikom pregleda nekog proizvoda, u nastavku prikazati i nekoliko sličnih proizvoda koji bi ga mogli zanimati. Odnosno, gotovo svaka od popularnijih web stranica gdje korisnici pregledavaju nekakve blog postove ili čitaju bilo kakve vrste članaka, prilikom pregleda jednog takvog sadržaja, u nastavku servisira korisnika s poveznicama na sadržaje slične tematike.

Iako je ovo još uvijek grebanje po samoj površini domena i opsega samih sustava za preporuku, može se vrlo lako zaključiti da su takvi sustavi široko rasprostranjeni te imaju različite zadaće. Od onih „jednostavnijih“ koji će na temelju pretrage jednog proizvoda na nekoj stranici ponuditi s nekoliko sličnih proizvoda, do onih naprednijih koji će prilikom korisničkog upisivanja pojma poput „nogometne tenisice“ u tražilicu, servirati ga i s povezanim reklamama o tenisicama na ostalim web stranicama koje ne moraju imati veze s njihovom prodajom, gdje će on možda čitati vijesti ili pretraživati novosti na društvenim mrežama.

U ovom diplomskom radu, u nastavku će se detaljnije objasniti što su sustavi za preporuku te koji (inteligentni) sustavi se nalaze u pozadini svega toga. Bit će dan pregled pristupa preporučivanja kojim se ti sustavi mogu koristiti, pregled tehnologija sustava za preporuku, od onih najjednostavnijih rješenja pa sve do inteligentnih sustava. Nakon toga će slijediti analiza postojećih sustava za prodaju automobila te izrada prototipa inteligentnog sustava za preporuku za kupce automobila kao glavnog dijela ovog diplomskog rada, gdje će se pritom naglasiti svi bitniji koraci izrade.

2. Sustav za preporuku

Prethodno daljnjem analiziranju sustava za preporuku, prvo je potrebno malo više pojasniti što taj pojam zapravo znači i predstavlja. Što su to **sustavi za preporuku** (eng. *Recommender Systems*, popularno još skraćeno *RSs*), ponajbolje objašnjavaju autori F. Ricci, L. Rokach i B. Shapira [1], gdje ih definiraju sljedećim riječima:

„Sustavi za preporuke, softverski su alati i tehnike koji daju prijedloge za stavke koje će biti korisne korisniku. Prijedlozi se odnose na različite procese donošenja odluke, poput stvari koje treba kupiti, koju glazbu slušati ili onoga što će se sljedeće čitati putem interneta.“

[1, str. 7]

Dok se kod definicije autori [1] zadržavaju na općenitoj svrsi takvog sustava, generalno govoreći s tehničkog stajališta, to je računalni program kojemu je zadaća pomoću korisniku da otkrije proizvode i sadržaje, prethodno predviđajući ocjenu korisnika za svaku od dostupnih stavki i prikazujući samo one za koje se vjeruje da bi korisniku bile najzanimljivije, sukladno nekim od metodologija za preporuku koje pri tome može koristiti.

Sustavi za preporuke se obično usredotočuju na određenu vrstu stavke, pri čemu je stavka zapravo općeniti pojam koji se koristi za označavanje onoga što sustav preporučuje (npr. knjiga ili vijesti). Prethodno spomenuti autori [1], dalje navode kako su sustavi za preporuku primarno usmjereni prema pojedincima kojima nedostaje dovoljno osobnog iskustva ili sposobnosti za procjenu potencijalno velikog broja alternativnih predmeta koje, primjerice, web stranica može ponuditi.

Već je prethodno spomenuto da sustavi za preporuku mogu koristiti različite metodologije i imati različite svrhe, o kojima nešto više u nastavku ovog rada, međutim osnovna apstrakcija rada sustava spomenutog poviše, može biti prikazana na sljedeći način.



Slika 1. Apstraktni prikaz rada sustava za preporuku [autorski rad]

Slika 1 ilustrira nekoliko osnovnih apstrahiranih dijelova koji sudjeluju u stvaranju preporuka, a to su **podaci o korisniku**, **stavka** koja se evaluira, **funkcija preporuke** i **predviđena ocjena** korisnika.

Podaci o korisniku – dobiveni putem profiliranja korisnika, bilo to tako da se analiziranju njegove prethodne kupnje ili pretraživanja, daju prijedlozi na osnovi sličnosti s ostalim korisnicima ili nekim od drugih metodologija.

Evaluirana stavka – stavka koja se nalazi, primjerice, u ponudi nekog web shopa.

Funkcija preporuke – ova funkcija predstavlja logiku koja stoji iza samog načina na koji se preporuka ostvaruje. To može varirati od doslovno neke funkcije s pripadajućim algoritmom kao najjednostavnijim rješenjem, pa sve do posebno razvijenih inteligentnih sustava ili modula strojnog učenja kao naprednijih rješenja.

Predviđena ocjena – predstavlja predviđenu mjeru „sviđanja“ neke stavke za nekog korisnika. Najbolje rangirane stavke će biti prikazane kao prijedlozi korisniku jer se za njih smatra da su to također stavke koji će korisnik najvjerojatnije kupiti.

2.1 Različite mogućnosti sustava za preporuku

U prethodnom poglavlju, sustavi za preporuku su definirani kao softverski alati i tehnike koje korisnicima pružaju prijedloge, dok se u nastavku fokus prebacuje na različite mogućnosti koje sustavi za preporuku mogu imati, odnosno različite uloge koje može igrati. Prije svega se tu moraju razlikovati dva nužna pogleda:

- a. ulogu koju sustav za preporuku igra kod **pružatelja takve usluge** i
- b. ulogu koju igra za **korisnika sustava**.

U nastavku će se navesti najbitnije i najprepoznatije mogućnosti koje sustav za preporuku može ponuditi pružatelju usluge te korisniku sustava, te koje će biti motiv za njegovo korištenje.

2.1.1 Mogućnosti za pružatelja usluge

Postoje različiti razlozi i prednosti koje sustavi za preporuku donose sa sobom, zbog kojih davatelji usluga možda žele iskoristiti tu tehnologiju, a pregled onih prema autorima Ricci, Rokach i Shapira [1] je izdvojen ispod.

Povećati broj prodanih predmeta – ovo je vjerojatno i najvažnija funkcija komercijalnog sustava za preporuku, a označava mogućnost prodaje dodatnog broja predmeta u odnosu na onaj koji se obično prodaje bez ikakvih preporuka. Taj se cilj postiže iz razloga što će preporučene stavke vjerojatno odgovarati potrebama i željama korisnika, a personaliziranu ponudu će vjerojatno prepoznati i korisnik nakon što je isprobao nekoliko preporuka. Nekomercijalne aplikacije imaju slične ciljeve, čak i ako korisnici nemaju troškove prilikom odabira neke stavke. Kao primjer se može navesti web stranica koja za cilj ima povećati broj vijesti koje se čitaju na njezinoj domeni. Općenito se može zaključiti da je jedan od osnovnih ciljeva uvođenja sustava za preporuku povećati stopu konverzije, odnosno broj posjetitelja koji na stranicama prihvaćaju preporuku i pregledavaju dodatne sadržaje ili predmete, u odnosu na mnoštvo jednokratnih posjetitelja koji samo pretražuju informacije.

Prodaja raznovrsnijih predmeta – druga glavna funkcija sustava za preporuku je omogućiti korisniku da odabere stavke koje bi bilo teško pronaći bez precizne preporuke. Za primjer se može uzeti sustav za preporuku filmova, kao što je to Netflix, gdje je ponuditelj usluge zainteresiran iznajmiti sve filmove koje ima u katalogu a ne samo one najpopularnije. Bez sustava za preporuku bi to bilo teško postići, jer je besmisleno reklamirati svaki film posebno i plaćati reklamu za njega, a on vjerojatno neće odgovarati ukusu određenih korisnika. Stoga sustav za preporuku, ako filmovi odgovaraju ukusu korisnika, predlaže i manje popularne filmove pravim korisnicima koje će to zanimati.

Povećanje zadovoljstva korisnika – dobro dizajniran sustav za preporuku može poboljšati iskustvo korisnika s određenom aplikacijom ili web stranicom. Korisniku će personalizirane preporuke biti zanimljive, relevantne i uz pravilno dizajniranu interakciju on će uživati u korištenju sustava. Takva kombinacija personaliziranih preporuka i korisnog sučelja povećava korisničku subjektivnu procjenu samog sustava, što će zauzvrat za posljedicu povećati korištenje sustava i vjerojatnost da buduće preporuke budu prihvaćene.

Povećanje vjernosti korisnika – ako određeno web mjesto, prilikom ponovne posjete, prepozna korisnika kao starog poznanika i tretira ga kao vrijednog posjetitelja, korisnik će u pravilu biti vjeran tom web mjestu i opet se vratiti. Ovo je normalna značajka nekog sustava za preporuku, s obzirom na to da mnogi sustavi za svoje preporuke koriste informacije dobivene od korisnika u prethodnim interakcijama (npr. ocjene prethodno pogledanih filmova). Sukladno tome, što duže korisnik stupa u interakciju s nekim web mjestom, sustav će dobiti i bolji profil korisnika te davati preciznije preporuke kao rezultat.

Bolje shvaćanje korisničkih potreba – još jedna važna funkcija sustava za preporuku, koja se može iskoristiti i u mnogim drugim aplikacijama, je shvaćanje korisničkih potreba. Bilo da je ono dano eksplicitno od strane korisnika ili prikupljeno od strane sustava, pružatelj usluge može iskoristiti ovo znanje za ostvarivanje drugih ciljeva, poput poboljšanja upravljanja zalihama i korigiranjem određene proizvodnje predmeta. Sukladno tome se mogu prilagoditi i reklame za određenu regiju, stvarati specifične vrste ponuda za specifične grupe korisnika i slično.

Prethodno opisana kategorija će posebno biti važna i u nastavku ovog diplomskog rada, gdje će se izraditi inteligentni sustav za preporuku za kupce automobila upravo na način da će se profilirati korisnici, te na temelju saznavanja njihovih potreba, predlagati relevantna vozila koja ih u najvećoj mjeri zadovoljavaju.

2.1.2 Mogućnosti za korisnika sustava

Prethodno je spomenuto nekoliko važnih mogućnosti sustava za preporuku, odnosno motiva zašto bi pružatelj usluge htio uvesti sustav za preporuku. Sada će se fokus prebaciti na stranu korisnika koji će možda htjeti sustav za preporuku, ukoliko će on učinkovito podržati njegove ciljeve ili zadatke koje želi ostvariti.

Herlocker i sur. [2], koji su na neki način postali klasična referenca za ovo polje, analizirali su motive korisnika za korištenjem ovakvih sustava, te definirali jedanaest popularnih zadataka s kojima im sustav za preporuku može pomoći u provedbi. Neki od tih zadataka se smatraju osnovnim zadaćama sustava za preporuku, poput *ponuditi prijedloge predmeta koji mogu biti korisni korisniku*, dok su drugi smatrani više „oportunističkim“ načinima iskorištavanja sustava. Ovi zadaci se također mogu usporediti s web tražilicom, gdje je njena primarna funkcija pronalaženje dokumenata koji su relevantni za potrebe korisnika, ali se također može provjeriti i važnost web stranice (njen položaj u popisu rezultata upita) ili pak

otkriti različiti načini upotrebe riječi u zbirci dokumenata [2]. Pregled istih kao i dodatno pojašnjenje, dano je u nastavku.

Pronaći neke dobre predmete – korisniku se preporučuju neke stavke kao (rangirani) popis u onom poretku u kojem bi ga mogle zanimati. Neki sustavi stavke rangiraju primjerice na skali od jedne do pet zvjezdica, drugi kao *top N* preporučenih stavki za korisnika dok neki sustavni uopće ne rangiraju stavke (ili ne prikazuju predviđenu ocjenu) te daju kao popis stavki u nastavku. To je ujedno i glavni zadatak preporuke koji sustavi obavljaju.

Pronaći sve dobre predmete – korisniku se preporučuju sve stavke koje mogu zadovoljiti neke od njegovih potreba. U takvim slučajevima nije dovoljno pronaći samo pojedine stavke koje najbolje odgovaraju upitu. To je posebno važno kada je broj predmeta relativno mali i kad je djelovanje sustava za preporuku od kritičnog značenja za odvijanje nekog postupka, primjerice u medicinskim ili financijskim aplikacijama. Kod takvih situacija, osim koristi koja se dobiva ispitivanjem svih mogućnosti, korisnik može imati korist i od sustavnog rangiranja ili davanja dodatnih neočekivanih objašnjenja koje on otvara.

Napomena u kontekstu – s obzirom na postojeći kontekst, poput popisa TV programa, korisniku se mogu naglasiti neki od programa ovisno o njegovim dugoročnim preferencijama. U takvom slučaju, sustav za preporuku TV emisija može istaknuti korisniku koje od nadolazećih emisija u rasporedu programa bi mogla biti zanimljive odnosno vrijedne pažnje korisnika.

Preporuka slijeda – umjesto fokusiranja na stvaranje pojedinačne preporuke, ideja je preporučiti slijed predmeta koji će biti ugodan kao cjelina za korisnika. Tipični primjeri bi uključivali preporuke za gledanje sljedeće TV serije, knjigu o sustavima za preporuku nakon već preporučene knjige o rudarenju podataka ili popis glazbe za reproduciranje [2].

Preporuka paketa – preporuka grupe predmeta koji se dobro uklapaju zajedno. Kao primjer se može navesti stvaranje plana putovanja koje se može sastojati od različitih atrakcija, odredišta i usluga smještaja koje se nalaze na nekom promatranom ograničenom području. Sa stajališta korisnika, ove različite mogućnosti mogu se smatrati i odabrati kao jedan plan putovanja na neko odredište [3].

Samo pregledavanje – kod ovog zadatka, korisnik pregledava katalog bez ikakve neposredne namjere za kupnju nekog predmeta. Zadatak sustava za preporuku je pomoći korisniku da pregledava stavke za koje je vjerojatnije da će biti unutar okvira korisničkih interesa za određenu sesiju pregledavanja.

Pronalazak vjerodostojnog preporučitelja – neki korisnici jednostavno ne vjeruju sustavima za preporuku, pa se igraju s njima da vide koliko su dobri u davanju preporuka. Stoga, neki sustavi nude i određene funkcije kako bi korisnici mogli testirati njihova ponašanja, uz samo one potrebne funkcije za dobivanje preporuka.

Poboljšavanje profila – odnosi se na sposobnost korisnika da sustavu za preporuke pruži, odnosno unese informacije o tome što mu se sviđa, a što ne. To je temeljni zadatak koji je strogo potreban za davanje personaliziranih preporuka. Ako sustav nema specifična znanja o aktivnom korisniku, on mu onda može pružiti samo one preporuke koje bi se isporučivale i prosječnom korisniku.

Mogućnost izražavanja – neki korisnici se možda uopće ne brinu o preporukama koje im sustav nudi. Umjesto toga, njima je važno da im sustav dopušta da daju svoje ocjene i izražavaju svoja mišljenja. Zadovoljstvo korisnika tom aktivnošću i dalje može poslužiti kao motiv za daljnje korištenje aplikacije.

Pomaganje drugima – neki korisnici rado doprinose informacijama, primjerice ocjenjivanjem predmeta, jer smatraju da zajednica ima koristi od njihovog doprinosa. Ovo može biti glavna motivacija za unošenje informacija u sustava preporuka koji se ne koristi rutinski. Na primjer, kod sustava za preporuku automobila, korisnik koji je već kupio svoj novi automobil svjestan je da ocjena unesena u sustav vjerojatnije će biti korisnija drugim korisnicima kod njihovih odluka, nego kod njegovih odluka kada će kupovati novi automobil.

Utjecaj na druge – u mrežnim sustavima za preporuku, postoje i korisnici čiji je glavni cilj utjecati na druge korisnike koji kupuju određene proizvode. Tu ponekad može doći i do pojave „zlomajernih“ korisnika koji će koristiti sustav samo kako bi promovirali ili kaznili (namjerno ostavili loše recenzije) određene proizvode.

Kao što ove različite točke ukazuju, uloga sustava za preporuku kod informacijskog sustava može biti vrlo raznolika. Sustav za preporuku također mora **uravnotežiti potrebe** pružatelja usluge kao i korisnika same usluge kako bi bio koristan za obje strane [1]. Raznolikost sustava za preporuku zahtijeva iskorištavanje različitih izvora podataka (kod naprednijih sustava mogu biti dostupne i baze znanja) kao i tehnika, odnosno metoda obrade tih podataka, o čemu će se nešto više reći u nastavku.

2.2 Izvori podataka i znanja

Sustavi za preporuku su sustavi koji aktivno prikupljaju različite vrste podataka kako bi mogli stvarati preporuke svojim korisnicima. Podaci koji se prikupljaju su prije svega o stavkama koje bi se mogle predložiti i o korisnicima koji će primiti te preporuke. Budući da izvori podataka, odnosno znanja koji su sustavima dostupni mogu biti vrlo raznoliki, ono što će se iskoristiti ili zanemariti u konkretnom slučaju ovisi i metodama preporuke koje sustav koristi. Više o metodama stvaranja preporuka će biti rečeno u sljedećem odlomku, ali općenito postoje metode koje donese odluka na temelju „jednostavnih podataka“ poput korisničkih ocjena za pojedine stavke, koje nemaju posebno razvijene baze znanja kao i one napredne koje donose preporuke upravo na temelju razvijenih baza znanja. To može uključivati ontologije opisa stavki koje je korisnik pregledavao, različita ograničenja, društvene odnose između korisnika, aktivnosti korisnika i slično.

Ricci, Rokach i Shapira [1, str. 26], navode opću klasifikaciju podataka koje koriste sustavi za preporuke se odnosi na tri vrste objekata: **predmeti**, **korisnici** i **transakcije**, tj. **odnosi između korisnika i predmeta**.

2.2.1 Predmeti

Predmeti su objekti koji se preporučuju korisnicima. Montaner, López i Rosa [4], govore o tome kako se predmeti mogu okarakterizirati svojom **složenošću** i **vrijednošću** ili **korisnošću**.

Vrijednost predmeta može biti pozitivna ako je predmet koristan za korisnika ili pak negativna ako taj predmet nije prikladan i korisnik je donio pogrešnu odluku prilikom odabira [4]. Kada korisnik nabavi predmet, on će uvijek sa sobom nositi trošak, koji uključuje kognitivni trošak pretraživanja i stvarni novčani trošak koji je na kraju plaćen. Primjer toga je da dizajner sustava za preporuku vijesti mora uzeti u obzir **složenost** vijesti, odnosno njezinu strukturu, tekstualnu reprezentaciju i vremenski ovisnu važnost [4]. Istovremeno, dizajner sustava mora shvatiti da čak i ako korisnik ne plaća čitanje vijesti, uvijek postoji kognitivna povezanost s pretraživanjem i čitanjem vijesti. Ako je odabrana vijest (stavka) relevantna za korisnika, tim troškom dominira korist od stjecanja korisnih informacija, dok ako vijest nije relevantna, vrijednost te vijesti za korisnika i njena preporuka je negativna. U ostalim domenama, primjerice kod automobila ili financijskih ulaganja, pravi monetarni trošak predmeta postaje važan element koji treba uzeti u obzir pri odabiru najprikladnijeg pristupa za stvaranje

preporuke. Predmeti s niskom složenošću i vrijednosti su: vijesti, web stranice, knjige, filmovi. Predmeti veće složenosti su: digitalni fotoaparati, mobilni telefoni, računala itd. Najkompleksniji predmeti koji su razmatrani su police osiguranja, financijska ulaganja, putovanja i poslovi [4].

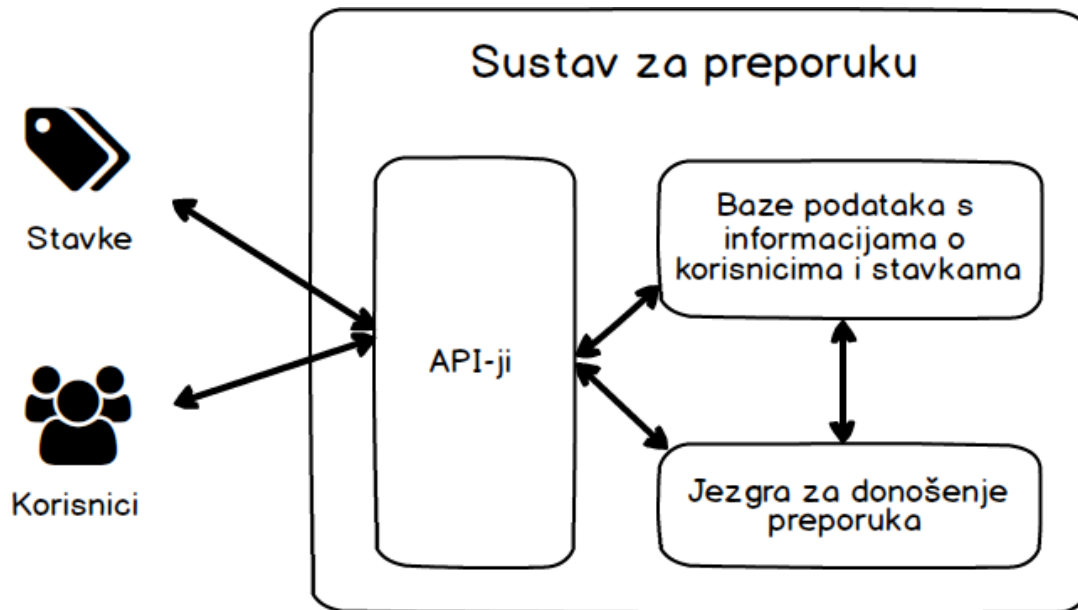
2.2.2 Korisnici

Korisnici sustava za preporuke, kao što je već nekoliko puta spomenuto poviše, mogu imati vrlo različite ciljeve korištenja sustava kao i karakteristike. Kako bi se što bolje personalizirale preporuke i interakcija između čovjeka i računala, sustavi za preporuke iskorištavaju što je više moguće informacija o korisnicima. Prikupljene informacije se mogu strukturirati na različite načine te model podataka ovisi o načinu, odnosno metodi stvaranje preporuke.

Kao primjer toga se može navesti pristup kolaborativnog filtriranja, o kojem više nešto kasnije, gdje je ulogu korisničkih podataka posebno promatrao Fischer [5] gdje navodi da se korisnici modeliraju kao jednostavan popis s ocjenama koje korisnik daje za neke predmete. Drugi primjer koji Fischer [5] navodi je demografsko filtriranje korisnika, gdje sustav za preporuke koristi sociodemografske atribute o korisnicima poput dobi, spola, profesije i obrazovanja. Zaključno, Fischer [5, str. 6] govori o tome kako **korisnički podaci** čine **korisnički model**. Korisnički model profilira korisnika, odnosno kodira njegove sklonosti i potrebe. U tom smjeru, sustav za donošenje preporuka se može gledati i kao alat koji generira preporuka izgradnjom i korištenjem korisničkih modela. Korisnici se također mogu opisati i podacima koji predstavljaju njihove obrasce ponašanja, primjerice obrasci pregledavanja nekog web mjesta ili obrasci pretraživanja putovanja. Tu se mogu pregledavati različiti faktori poput: **web sekcije** gdje korisnik **započne pregledavanje**, kojim **opcijama se najviše služi**, koje **filtre najčešće primjenjuje** ili na kojem **koraku pretraživanja odustaje** od pretrage [5]. Sve ovo, sustav za donošenje preporuka ne mora nužno raditi i prikupljati kod svakog korisnika, već na temelju njihovih atributa i sklonosti, preporučuje stavke korisnicima koje su preferirali slični korisnici s već dobro razvijenim modelom podataka.

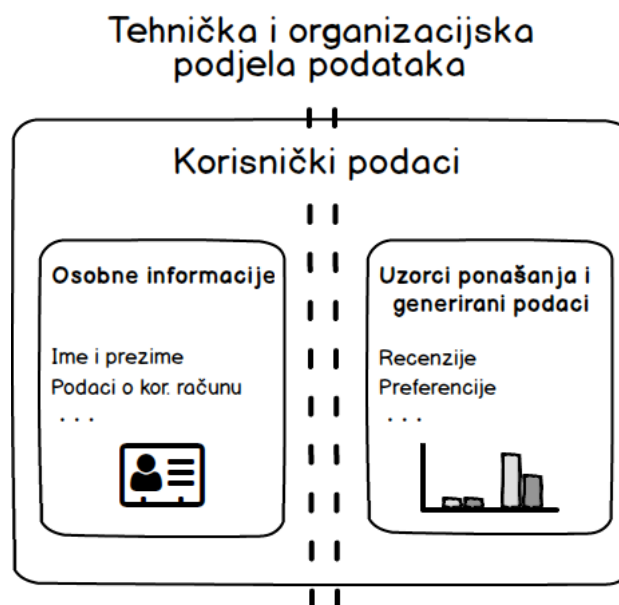
Govoreći o prikupljanju korisničkih podataka, treba spomenuti i relativno nove GDPR odredbe koje u nekim slučajevima značajno mijenjaju načine i količinu podataka koji se prikupljaju danas s obzirom na one u vremenu koje je prethodilo tim odredbama. Jedni od autora koji su se posvetili toj temi su Tejeda-Lorente i sur. [6], koji govore o novim regulacijama i preprekama koje one predstavljaju ovakvim sustavima, ali i načinima kako bi takav sustav na njih mogao odgovoriti. Pridodaje se važnost vođenju dodatnog računa o tome kako se

sastavljaju novi korisnički modeli, te predstavljaju i novi trend kod metodologija inteligentnih sustava [6].



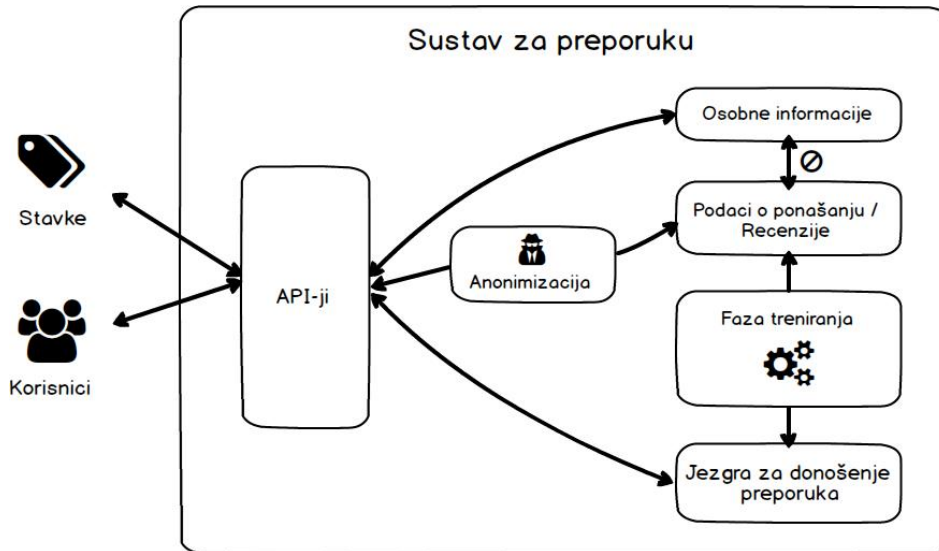
Slika 2. Glavne komponente arhitekture sustava za preporuke [autorski rad prema 6, str. 375]

Na slici 2 moguće je razlikovati tri glavne komponente tradicionalnih sustava za davanje preporuka: **baze podataka** koje sadrže informacije o korisnicima, korisničkim recenzijama i predmetima, **API-ji** koji razgovaraju i prikupljaju podatke o korisnicima i predmetima te **jezgra za donošenje preporuka**.



Slika 3. Princip razdvajanja znanja [autorski rad prema 6, str. 381]

Praćenjem smjernica novih odredbi, vrši se tehnička i organizacijska podjela korisničkih informacija koja se može vidjeti na slici 3. Uz tu podjelu, kao norma se provodi i **anonimizacija** i **pseudo-anonimizacija** kod svakog novog spremanja ili korištenja korisničkih podataka, kada to ne utječe na primarnu svrhu funkcije sustava. Na temelju toga, predlaže se i novu arhitekturu sustava za preporuke [6].



Slika 4. Arhitektura anonimiziranog sustava za preporuke [autorski rad prema 6, str. 381]

Osobni podaci se trebaju svesti na minimalno potrebne i spremati se odvojeno od podataka koji opisuju uzorke ponašanja i recenzije korisnika. Podaci koji opisuju uzorke ponašanja i recenzije korisnika trebaju biti anonimizirani prije spremanja. Dodatni korak koji se predlaže je također, ako je to moguće prethodno korištenju glavne jezgre, trenirati (strojno učiti) algoritam koji se koristi za davanje preporuka s učitanim korisničkim podacima koji se koriste u skladu s odredbama zaštite privatnosti [6].

2.2.3 Transakcije - odnosi između korisnika i predmeta

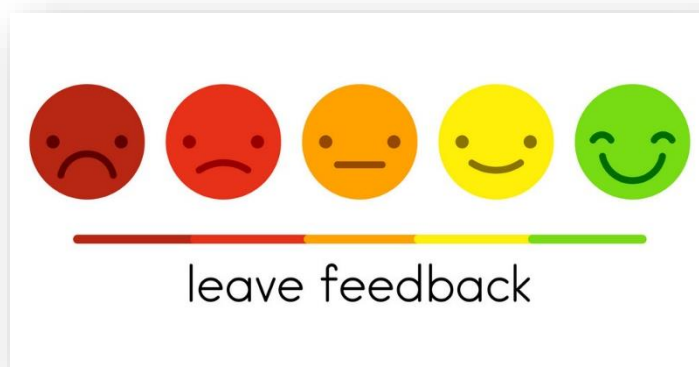
Transakcijama se općenito nazivaju sve **zabilježene interakcije** između korisnika i sustava za preporuku. Transakcije se mogu promatrati i kao dnevnik s važim podacima pohranjenim tijekom korisničke interakcije sa sustavom, koji je koristan za algoritam generiranja preporuka kojeg sustav koristi. Primjer takvih podataka može biti referenca na stavku koju je korisnik odabrao i polazni upit koji je koristio. Ukoliko je dostupno, ova transakcija može uključivati i povratne informacije koje je korisnik dao za odabranu stavku, poput recenzije. Schafer, Frankowski, Herlocker i Sen [7] navode kako su recenzije zapravo i

najpopularniji oblik podataka o transakcijama koje sustav za davanje preporuka prikuplja. Recenzije mogu biti implicitne i eksplicitne, dok su eksplicitne najpopularnije i traže od korisnika da pruži svoje mišljenje o predmetu na ljestvici ocjena.

Prema [7], ocjene mogu poprimiti različite oblike:

- **Numeričke ocjene** poput 1-5 zvjezdica kao kod preporuke knjiga na Amazon.com stranicama.
- **Skala zadovoljstva**, kao što su „*potpuno se slažem, slažem se, neutralan sam, ne slažem se, potpuno se ne slažem*“, pri čemu se od korisnika traži da odabere pojam koji najbolje opisuje njegovo mišljenje o predmetu (obično se ispituje putem upitnika).
- **Binarne ocjene** koje predstavljaju model ocjenjivanja gdje se od korisnika jednostavno traži da odluči je li određeni predmet *dobar* ili *loš*.
- **Unarne ocjene** koje mogu ukazivati na to da je korisnik kupio ili promatrao neki predmet, ili na drugi način pozitivno ocijenio predmet. U takvim slučajevima, nepostojanje ocjene ukazuje na nedostatak podataka kod odnosa korisnika s proizvodom (korisnik je stavku možda kupio negdje drugdje). [7]

Primjeri najčešćih načina prikupljanja recenzija tj. povratnih informacija kod korisnika će biti prikazani na slikama u nastavku.



Slika 5. Skala zadovoljstva

(Izvor: preuzeto 14.4.2020. sa <https://www.vectorstock.com/>)

How would you rate the support you received?

Good, I'm satisfied
 Bad, I'm unsatisfied

Add a comment about the quality of support you received (optional):

Slika 6. Binarne ocjene

(Izvor: preuzeto 14.4.2020. sa <https://www.zendesk.hk/>)

Leave Feedback

Please leave feedback and rate for the project **Project for**

	Negative	Positive
Clarity in Specification		☆☆☆☆☆
Communication		☆☆☆☆☆
Payment Promptness		☆☆☆☆☆
Professionalism		☆☆☆☆☆
Would you work for the employer again?		☆☆☆☆☆

Comment: (Maximum 200 characters. This will be public)

200 characters left

Leave a private note for Freelancer Staff.

Slika 7. Numeričke ocjene

(Izvor: preuzeto 14.4.2020. sa <https://www.br.freelancer.com/>)

Koje će se sve podaci između korisnika i predmeta prikupljati te kakva će se vrsta recenzija omogućiti korisniku, bilo to na način da korisnik daje ocjene na jedan od načina poviše ili ostavlja tekstualne komentare, vrlo često ovisi i o pristupu, odnosno tehnici preporučivanja predmeta kod određenog sustava.

2.3 Pristupi kod preporučivanja

Kako bi sustav za preporuku izvršavao svoju osnovnu funkcionalnost, prepoznavanje korisnih i zanimljivih predmeta za korisnika, sustav mora odgonetnuti koji predmet vrijedi preporučiti a koji ignorirati. Da bi to učinio, sustav mora biti u mogućnosti predvidjeti **korisnost** različitih predmeta, usporediti ju s ostalim te na kraju usporedbe odlučiti koje predmete preporučiti na temelju usporedbe.

Međutim, neki sustavi preporuka ne procjenjuju u potpunosti korisnost prije nego što daju preporuku, već primjenjuju neke heuristike da hipoteziraju da predmet koristi korisniku [1]. To je primjerice slučaj kod sustava utemeljenih na bazama znanja, gdje su predviđanja korisnosti napravljena specifičnim algoritmima koristeći različite vrste znanja o korisnicima, predmetima i znanja o samoj funkciji za izračunavanje korisnosti. Jedan od primjera može biti sustav čija funkcija za izračunavanje korisnosti je tipa *Boolean*, te će on stoga samo utvrditi da li je stavka korisna ili nije. Zatim, ukoliko su mu raspoloživa neka saznanja o korisniku koji traži preporuke, sustav može iskoristiti to znanje te s odgovarajućim algoritmom generirati različita predviđanja i prilagoditi preporuke.

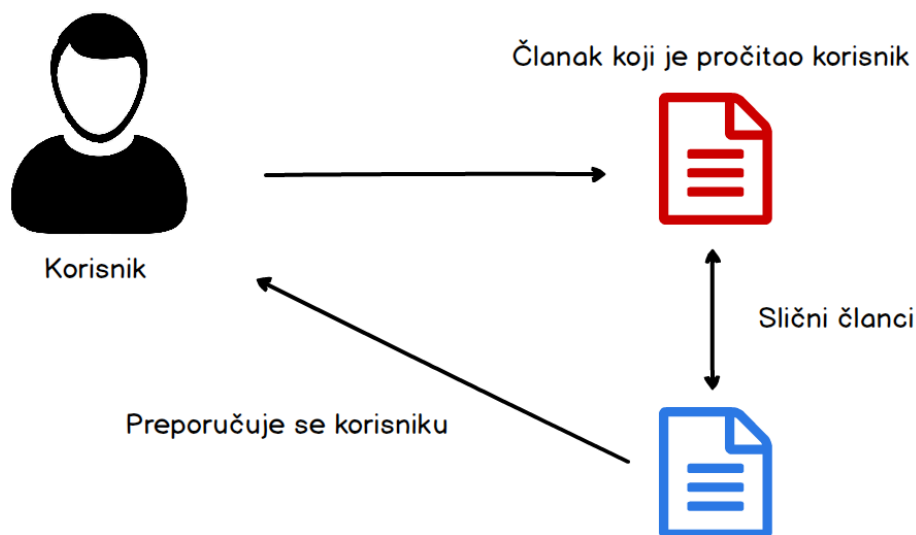
U nastavku će se dat pregled različitih pristupa odnosno metoda preporučivanja (a samim time i različitih vrsta sustava za preporuku) koji se ponajprije razlikuju po domeni djelovanja, znanju kojeg koriste ali i posebice algoritmu preporučivanja. Klasični način razlikovanja sustava za preporuke i samog referenciranja na njih, čini taksonomija kreirana od strane R. Burke [8]. Autor razlikuje šest različitih razreda (metoda) pristupa preporučivanja čiji je pregled dan u nastavku [8].

2.3.1 Preporučivanje na temelju sadržaja

Sustavi preporuka temeljeni na sadržaju (eng. *Content-based recommender systems*) uče kako preporučiti stavke korisniku slične onima koje je korisnik **označio pozitivno** (lajkao, stavio na listu želja, pretplatio se i slično), **kupio** ili **pročitao** u prošlosti. U tom postupku, sustavi se oslanjaju na **opise predmeta i korisnika** (što predstavlja *content*, odnosno sadržaj) za izradu reprezentativnih predmeta i korisničkih profila za predlaganje stavki ciljanom korisniku [1]. Postupak izrade preporuke sastoji se od usklađivanja atributa ciljanog korisničkog profila, u kojem su pohranjene preferencije i interesi, s atributima stavki. Rezultat tog postupka je predviđena korisnost koja predstavlja razinu interesa ciljanog korisnika s obrađenim stavkama.

Atributi za opis stavki su obično značajke izvađene iz **metapodataka** povezanih s tom stavkom ili **tekstualne značajke izvađene izravno iz opisa stavke**. Sadržaj izvađen iz metapodataka je često prekratak i sam po sebi nedovoljan da pravilno definira korisničke interese, dok upotreba tekstualnih značajki uključuje niz komplikacija prilikom stvaranja korisničkog profila zbog prirodne jezične dvosmislenosti [1]. To spadaju *polisemija* (višeznačnost), *sinonimija*, izrazi s više riječi, prepoznavanje imenovanih entiteta i slično, što sve zajedno predstavlja svojstvene probleme tradicionalnih profila temeljenih na ključnim riječima. Takvi profili vrlo često nisu u mogućnosti nadići upotrebu leksičkih i/ili sintaktičkih struktura da bi zaključili interes korisnika za pripadajuće teme.

Za primjer se može uzeti slučaj preporuke filmova, gdje ako je korisnik pozitivno ocijenio film koji pripada žanru komedije, tada sustav može naučiti preporučiti korisniku druge filmove iz ovog žanra, koji će možda biti i slične tematike ili glumačkog sastava, ukoliko su takvi podaci također bili uključeni u postupak stvaranja preporuke [1].

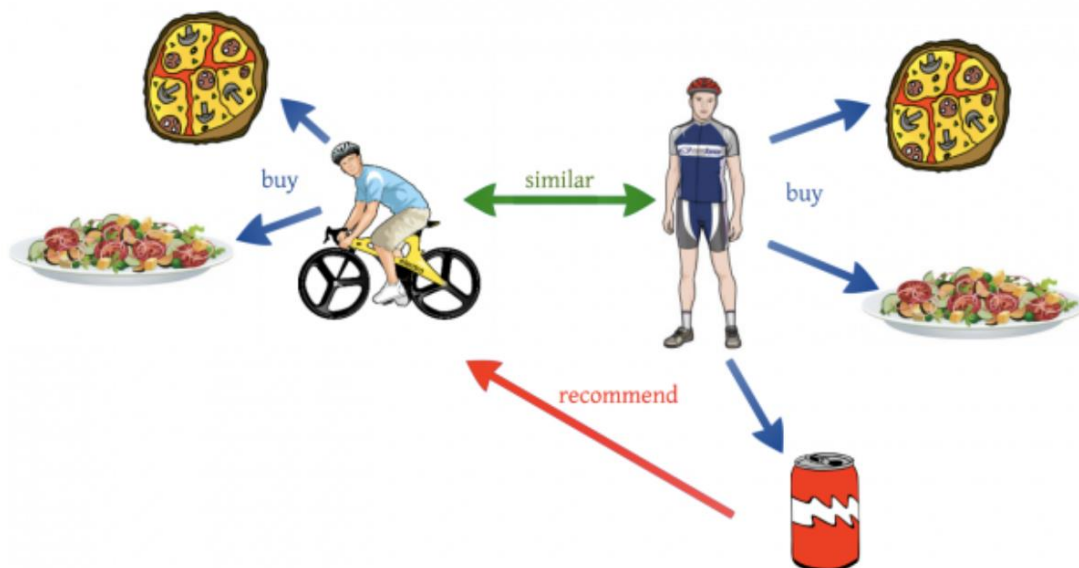


Slika 8: Preporučivanje na temelju sadržaja [autorski rad]

U posljednje vrijeme, sve veći interes za semantičke tehnologije i dostupnost nekoliko otvorenih izvora znanja, kao što su to Wikipedija, DBpedia, FreeBase i BabelNet, potaknuo je napredak u području sustava za davanje preporuka na temelju sadržaja. Uvodu se nove semantičke tehnike koje prelaze s reprezentacija na temelju ključnih riječi na reprezentacije konceptualnog prikaza stavki i korisničkih profila.

2.3.2 Kolaborativno filtriranje

Najjednostavnija i zapravo originalna primjena ovog pristupa je preporučiti aktivnom korisniku stavke koje su se u prošlosti sviđale drugim korisnicima koji imaju slične podatke, odnosno korisničke profile aktivnom korisniku. **Sličnost u interesima i preferencijama dvaju korisnika** izračunava se temeljem sličnosti u njihovim povijestima ocjenjivanja stavki. Ovo je također razlog zašto neki autori kolaborativno filtriranje nazivaju i „*korelacijom ljudi među ljudima*“ [9]. Kolaborativno filtriranje (eng. *Collaborative filtering*) se smatra najpopularnijom i najšire primijenjenom tehnikom kod sustava za davanje preporuka. Ilustracija osnovne ideje i načina na koji ovaj pristup radi je prikazana na slici 9.



Slika 9: Kolaborativno filtriranje

(Izvor: preuzeto 19.4.2020. sa <https://towardsdatascience.com/>)

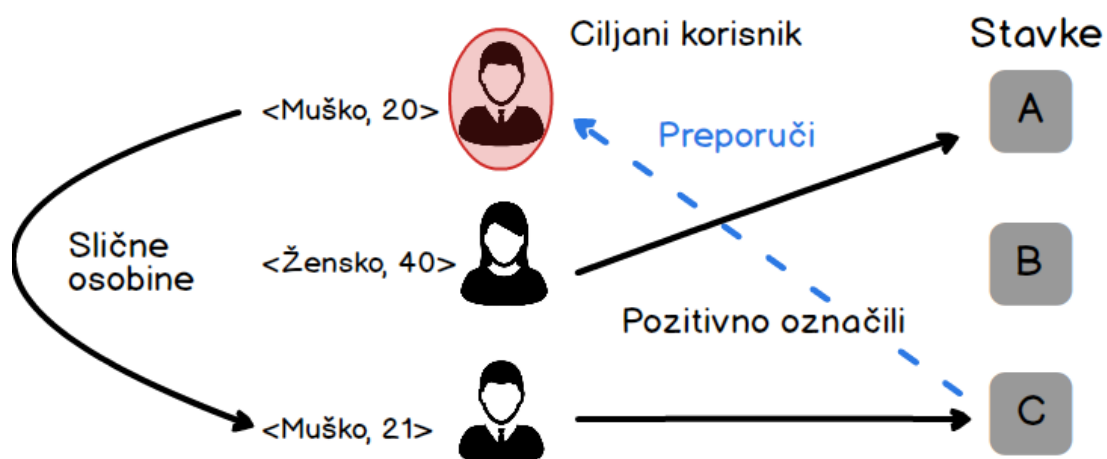
Razlog same popularnosti i rasprostranjenosti metoda sustava za preporuku koji rade na principu kolaborativnog filtriranja je davanje preporuka predmeta **na temelju obrazaca** u ocjenama ili upotrebi (npr. kupovina, dodavanje na listu želja, oznaka sviđa mi se) **bez potrebe za vanjski podacima** o stavkama ili korisnicima. Postoji više tipova pristupa kod izračunavanja preporuka unutar ove vrlo raširene metode, te se općenito uvažava njihova podjela na pristupe temeljene na memoriji, odnosno dostupnim podacima (eng. *memory based approach*) i pristupe zasnovane na modelima podataka (eng. *model based approach*) [7]. Prvi pristup je jednostavniji i predstavlja statistički pogled na pronalaženje sličnih korisnika, tipično na osnovi Pearson-ove korelacije i uzimanju ponderiranih prosječnih ocjena za određivanje preporuka. Unatoč vrlo jednostavnom ali efikasnom pristupu ukoliko postoji veći broj podataka, zbog

problema sa skalabilnosti ovih tehnika i rijetkim podacima, s vremenom se razvija i pristup temeljen na modelima podataka. Takav pristup koristi strojno učenje kako bi se predvidjele moguće korisničke ocjene za dosad neocijenjene predmete. Tipične tehnike koje se ovdje primjenjuju su PCA (Principal Component Analysis), dekompozicija pojedinačnih vrijednosti, neuronske mreže i matična faktorizacija [7].

2.3.3 Demografsko preporučivanje

Sustavi za demografsko preporučivanje (eng. *Demographic recommender systems*) temelje se na demografskim karakteristikama korisnika i preporučuju stavke koje imaju dobre povratne informacije od korisnika koji su demografski slični ciljanom korisniku. Ovakav pristup se zasniva na pretpostavci da bi trebalo stvoriti različite preporuke za različite demografske skupine te ovakvi sustavi imaju za cilj kategorizirati korisnike na temelju njihovih dostupnih atributa te dati demografski klasificirane preporuke [1].

Mnoge industrije se koriste ovakvim pristupom zbog njegove jednostavnosti za implementaciju. Prije svega, algoritmi trebaju pravilno provesti istraživanje tržišta u određenoj regiji kako bi prikupili podatke za kategorizaciju. Demografske tehnike stvaraju „*korelacije ljudi među ljudima*“ kao kod kolaborativnog filtriranja ali koriste druge podatke o korisnicima [9]. Prednost demografskog pristupa je u tome što on ne zahtijeva povijest korisničkih ocjena ili aktivnosti kao što je to slučaj kod prethodno navedenih pristupa preporučivanja na temelju sadržaja i kolaborativnog filtriranja.

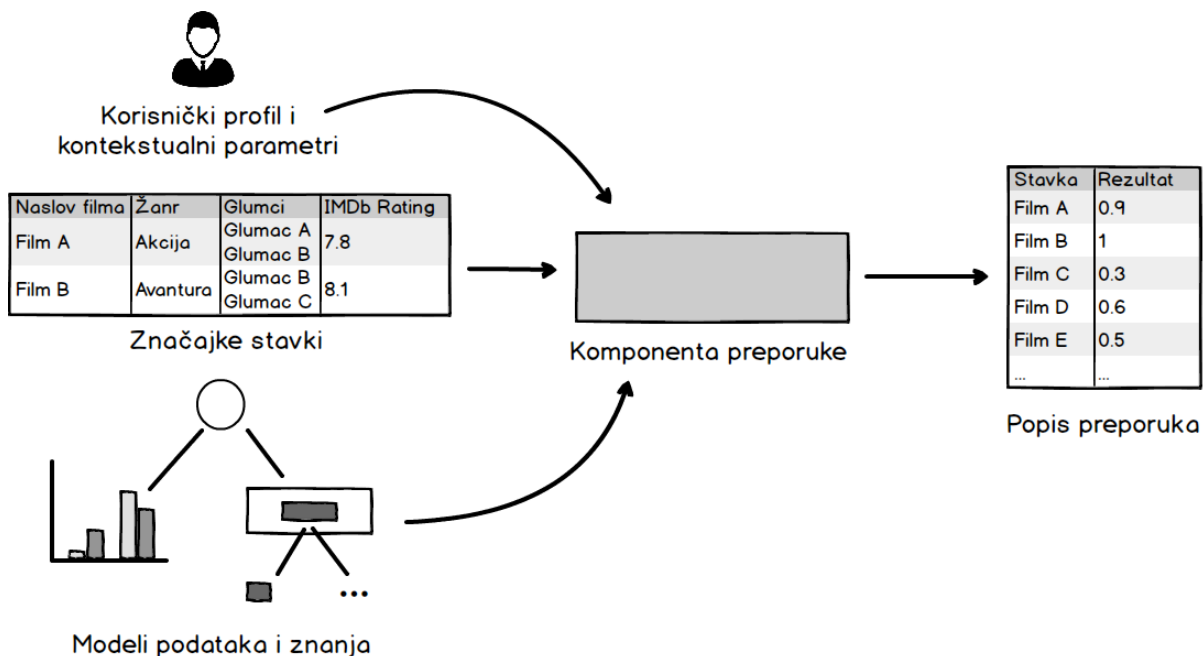


Slika 10: Demografsko preporučivanje [autorski rad]

Mnoge web stranice usvajaju jednostavna i učinkovita rješenja personalizacije temeljena na demografiji. Na primjer, korisnici se šalju na određene web stranice na temelju njihova jezika ili zemlje. Prijedlozi se također mogu prilagoditi prema dobi korisnika, kao što je to i prikazano na slici 10. Dok su ovakvi pristupi ustaljeni i poprilično popularni u marketingu, kod demografskih sustava za preporuku je do sada bilo relativno malo pravilnog istraživanja [1].

2.3.4 Preporučivanje temeljeno na znanju

Sustavi za preporuke temeljeni na znanju (eng. *Knowledge-based recommender systems*) preporučuju stavke temeljene na određenoj **domeni znanja** o tome kako **određene značajke kod predmeta** zadovoljavaju **potrebe i preferencije pojedinog korisnika**, te na samom kraju na temelju toga određuju u kojoj mjeri je stavka korisna za korisnika [10]. Ovakvi sustavi temeljeni na znanju obično djeluju bolje od ostalih sustava kada se gleda rad sustava neposredno nakon njegovog puštanja u produkciju. Međutim, ukoliko ti sustavi nisu opremljeni s komponentama za daljnje učenje na temelju novih prikupljenih informacija, takve sustave mogu nadmašiti ostali sustavi koji koriste „jednostavnije“ pristupe koji iskorištavaju podatke o interakcijama korisnika sa sustavom (kao što je to na primjer kolaborativno filtriranje) [10].



Slika 11: Preporučivanje temeljeno na znanju [autorski rad]

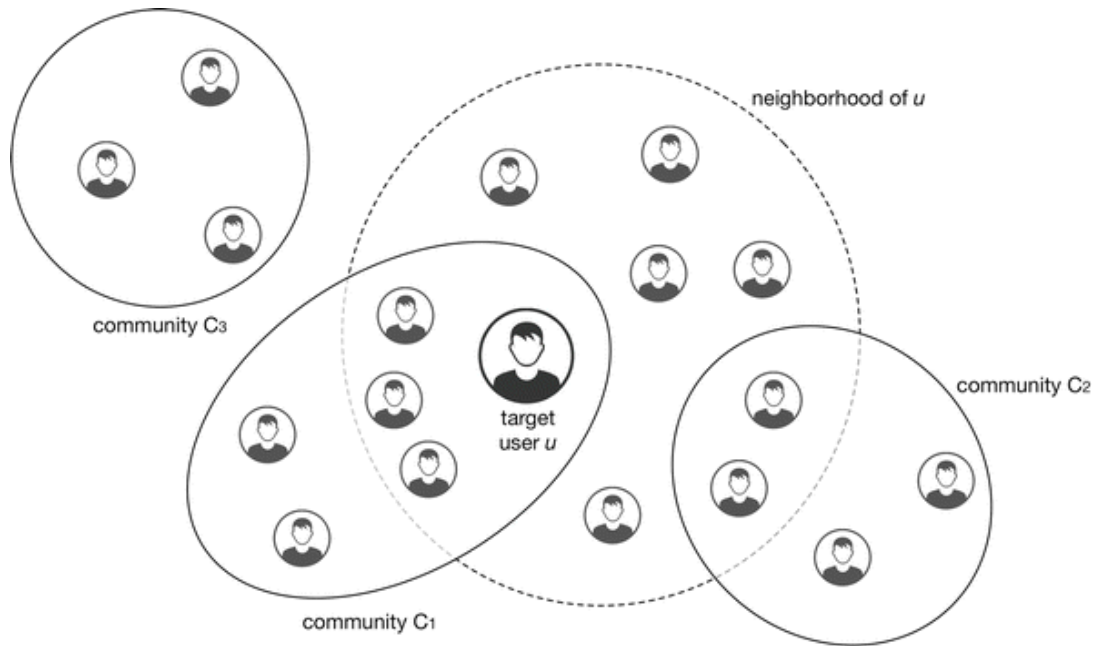
Bridge, Goker, McGinty i Smyrh [10] objašnjavaju kako značajan udio sustava za preporuke temeljenih na znanju se također temelji na slučajevima. Takva podskupina ovih sustava radi na principu „**pronađi predmet čije značajke odgovaraju mojim potrebama**“. U tim sustavima funkcija sličnosti procjenjuje koliko **korisnikove potrebe** (opis problema) odgovaraju **moogućim preporukama** (rješenjima problema) [10]. Ovdje se ta ocjena sličnosti može direktno interpretirati kao ocjena korisnosti preporuka za korisnika.

Druga podskupina sustava za preporuku temeljenih na znanju su sustavi temeljeni na ograničenjima. U smislu korištenja određene baze znanja, obje podskupine su slične: prikupljaju se zahtjevi korisnika, u situacijama kad se ne mogu pronaći rješenja automatski se predlažu popravci za nedosljedne zahtjeve, kao i prikazivanje objašnjena rezultata preporuka. Glavna razlika leži u načinu na koji se rješenja izračunavaju. Sustavi temeljeni na slučajevima određuju preporuke na temelju metrika sličnosti, dok sustavi temeljeni na ograničenjima pretežno koriste unaprijed definirane baze znanja koje sadrže izričita pravila o povezivanju zahtjeva korisnika sa značajkama predmeta [1].

2.3.5 Preporučivanje temeljeno na zajednici

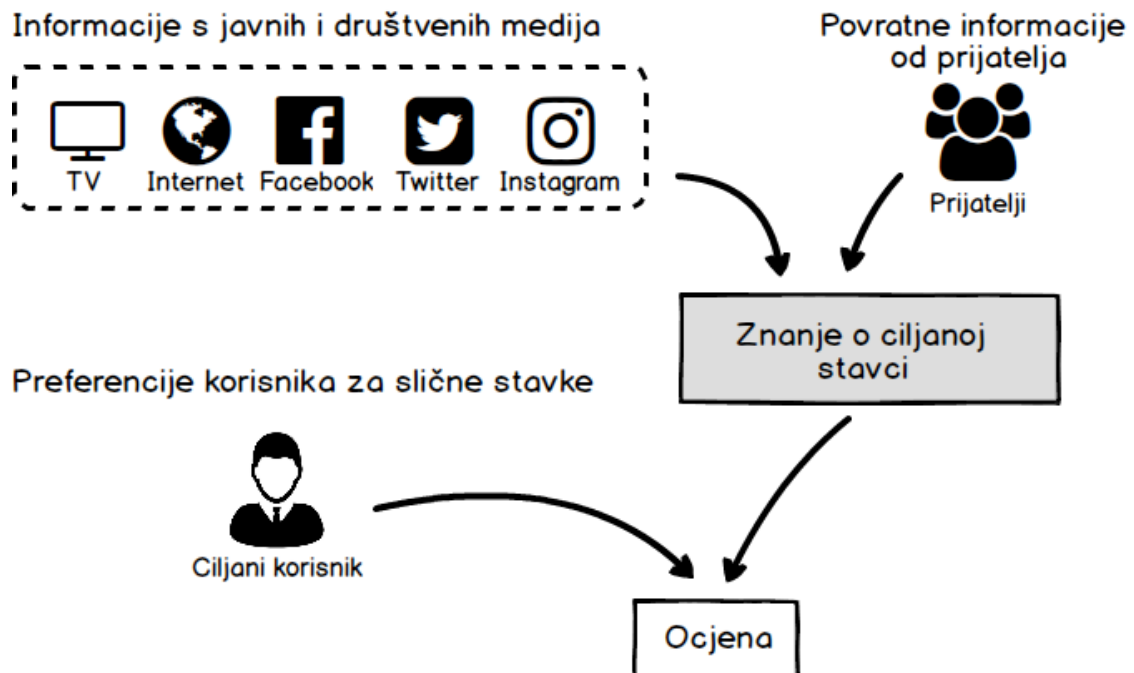
Preporučivanje temeljeno na zajednici (eng. *Community-based recommendation*) preporučuje stavke na temelju **preferencija prijatelja ciljanog korisnika**. Ovaj pristup preporučivanja slijedi epigram „*Recite mi tko su vam prijatelji i ja ću vam reći tko ste vi*“ [11]. Istraživanja pokazuju da se ljudi više oslanjaju na preporuke svojih prijatelja nego na preporuke sličnih ali anonimnih korisnika [11]. Ovo opažanje, u kombinaciji s velikom popularnošću društvenih aplikacija i mreža, generira veliki interes za sustave za preporučivanje temeljene na zajednici, ili još nazvane *društvene sustave za preporuku*.

Ova vrsta sustava za preporuku skuplja i modelira informacije o društvenim odnosima između korisnika kao i preferencijama prijatelja korisnika. Preporuka se temelji na recenzijama, odnosno ocjenama koje su ostavili prijatelji korisnika. Može se jednostavno reći da ovi sustavi prate aktivnosti na društvenim mrežama i omogućavaju jednostavno i sveobuhvatno prikupljanje podataka povezanih s društvenim odnosima između korisnika. Potrebno je još napomenuti da je vrlo bitan korak kod ove metode i sama **detekcija zajednice** u kojoj se ciljani korisnik nalazi [11]. Često ta zajednica ljudi sa sličnim interesima i preferencijama nisu korisnici koji su geografski *blizu* i spadaju u primjerice iste gradske zajednice, već mogu biti raznoliko raspoređeni pa je potrebno korisnika i njegove prijatelje ispravno klasificirati.



Slika 12: Detekcija zajednice

(Izvor: Gasparetti F., Micarelli A., Sansonetti G. (2017) Community Detection and Recommender Systems)



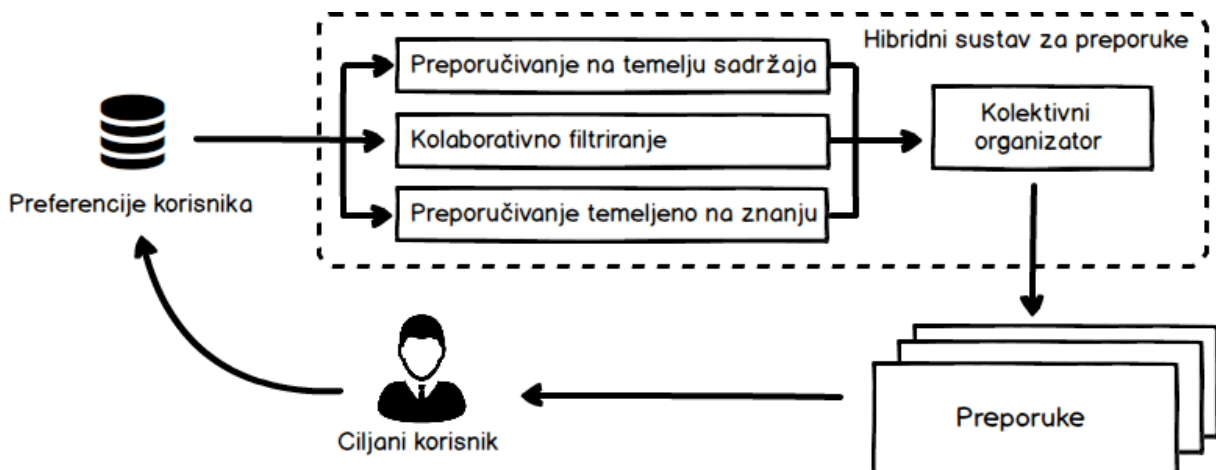
Slika 13: Preporučivanje temeljeno na zajednici [autorski rad]

Iako su ovi sustavi dosta popularni i dobro zastupljeni, postoje različita istraživanja koja donose i različite rezultate o njihovim performansama. Neka istraživanja govore o tome kako generalno preporuke temeljene na društvenim mrežama nisu točnije od preporuka izvedenih iz

tradicionalnih pristupa poput kolaborativnog filtriranja [1]. Tu u izuzetke spadaju posebni slučajevi kada su korisničke ocjene za određenu stavku vrlo različite (kontroverzne stavke) ili za slučajeve *hladnog pokretanja* (eng. *cold-start situations*) gdje korisnici nisu dali dovoljno ocjena za izračunavanje sličnosti s drugim korisnicima, što rezultira točnijim preporukama ovakvih sustava.[1] Druga istraživanja pak pokazuju da u nekim slučajevima podaci na društvenim mrežama daju bolje preporuke u odnosu na podatke o sličnosti korisnika, te se preporučuje dodavanje podataka o društvenim mrežama tradicionalnim pristupima poput kolaborativnog filtriranja kako bi se poboljšali rezultati preporuka [11, 1].

2.3.6 Hibridni sustavi za preporuke

Hibridni sustav za preporuke (eng. *Hybrid recommender system*) se temelji na **kombinaciji** prethodno navedenih pristupa. Hibridni sustav koji kombinira pristupe **A** i **B** pokušava **iskoristiti prednosti A** za **ispravljanje nedostataka B**. Na primjer, metode kod kolaborativnog filtriranja imaju probleme s novim predmetima, odnosno ne mogu izraditi preporuke za predmete koji nemaju ocjene. Taj slučaj ne ograničava sustave s pristupom temeljenim na sadržaju jer je predviđanje za nove predmete temeljeno na njihovim opisima (značajkama) koje su obično lako dostupne. Danas postoji mnogo predloženih i usvojenih načina preporučivanja koji kombiniraju dva ili tri osnovna pristupa sustava za preporuku, sve s ciljem davanja što preciznijih preporuka i izbjegavanja slučajeva gdje nije moguće ponuditi preporuku [1]. Takvi slučajevi u konačnici rezultiraju i propuštenim prilikama za ostvarivanje poslovnih ciljeva.



Slika 14: Hibridni sustav za preporučivanje [autorski rad]

3. Pregled tehnologija sustava za preporuku

Dok su dosad sustavi za preporuku bili promatrani na temelju pristupa kod preporučivanja, odnosno načinom **kako** donose preporuke, u nastavku ovog rada napraviti će se pregled tehnologija sustava za preporuku. Drugim riječima, ovog puta će se fokus staviti na **kojom tehnologijom** se sustav za preporuke služi kako bi donio preporuku.

Prethodno je analizirano 6 različitih pristupa kod stvaranja preporuka te su se u sklopu toga tekstualno i slikovno predstavili principi na kojima su oni zasnovani. Dok u sklopu ovog diplomskog rada nije moguće na ovakav način proći kroz sve tehnologije koje su se nalazile u pozadini svih vrsta sustava za davanje preporuka, u nastavku će se napraviti osvrt na uobičajene tehnologije dva najčešća pristupa. Daleko najčešći pristup koji se koristio kroz prošlost te ostao i najčešće korištenim danas je **kolaborativno filtriranje**, kojeg slijedi **preporučivanje na temelju sadržaja** [1]. Treba svakako napomenuti da je velika većina današnjih sustava za preporuke, gledajući sa strane pristupa preporučivanja, hibridna te kombinira nekoliko pristupa kako bi se maksimizirala učinkovitost i ispravili nedostaci [1]. Hibridni sustavi u velikoj većini koriste prethodno navedena dva pristupa (ili njihove varijacije) kao osnovu, kombinirajući ih s ostalima za postizanje željenih rezultata.

Pregled uobičajenih tehnologija u nastavku će slijediti od jednostavnijih algoritama prema složenijim tehnologijama, te će također biti podijeljen s obzirom na pristup rada s podacima. Razlikovat će se tehnologije utemeljene na **memoriji**, odnosno dostupnim podacima (eng. *memory based approach*, neki autori [1] još spominju i pridjev *heuristički*) i tehnologije zasnovane na **modelima podataka** (eng. *model based approach*).

3.1 Tehnologije utemeljene na memoriji (dostupnim podacima)

Prvi skup algoritama, odnosno tehnologija koje će biti obrađene u nastavku su tehnologije utemeljene na dostupnim podacima. Ove tehnologije koriste podatke poput lajkova, glasova, klikova i sličnih dostupnih kako bi uspostavile povezanosti (sličnosti) između korisnika (kolaborativno filtriranje) ili predmeta (preporučivanje na temelju sadržaja) te preporučile predmet korisniku koji ga nikad prije nije vidio. Ukratko, ovakve tehnologije se uvelike oslanjaju na *jednostavne* mjere sličnosti kako bi povezale slične korisnike ili predmete zajedno.

3.1.1 TF-IDF

Prvi uobičajeni algoritam u ovom pregledu tehnologija utemeljenim na memoriji je TF-IDF, skraćenica za pojam **Term Frequency – Inverse Document Frequency**, koji predstavlja numeričku statistiku kojoj je svrha odražavati koliko je pojedina riječ značajna za dokument u nekoj zbirci. TF-IDF je vrlo koristan u mnogim područjima kao što je filtriranje na temelju sadržaja, tehnike rudarenja tekstualnih podataka i sličnim kontekstima gdje je bitno pretraživanje informacija. Algoritam je također **ključni dio izgradnje profila predmeta** u sustavima koji rade preporuke na temelju sadržaja.

TF-IDF je rezultat kombiniranja dviju odvojenih statistika, frekvencije pojma (eng. *term frequency*) i inverzne frekvencije dokumenta (eng. *inverse document frequency*). Dok postoje različiti načini za utvrđivanje točnih vrijednosti kod obje statistike [25, 26], u nastavku će se ukratko objasniti osnovni matematički model za njihov izračun.

$$TF_{ij} = \frac{f_{ij}}{n_j} \quad (1)$$

Gdje je f_{ij} frekvencija pojma i unutar dokumenta j . n_j je ukupan broj riječi koje se nalaze u dokumentu j .

$$IDF_i = 1 + \log\left(\frac{N}{c_i}\right) \quad (2)$$

Gdje je N ukupan broj dokumenata u zbirci. c_i je broj dokumenata koji sadrže riječ i .

$$\omega_{ij} = TF_{ij} \times IDF_i \quad (3)$$

Gdje je ω_{ij} TF-IDF rezultat za pojam i unutar dokumenta j .

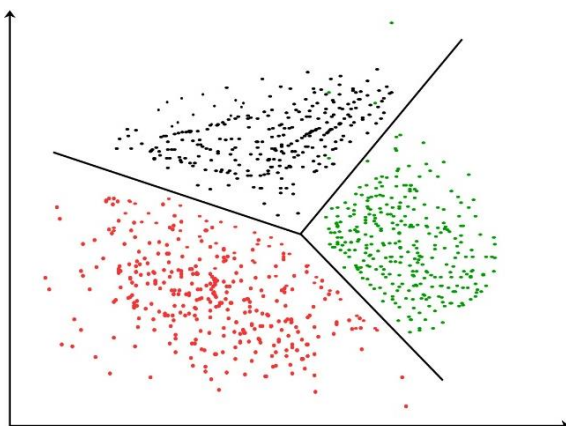
Moguće je primijetiti da ako se izračunava TF-IDF vrijednost za svaku riječ unutar svakog dokumenta, onda će krajnji rezultat oblikovati matricu oblika *broj riječi* \times *broj dokumenata*. Dakle, može se zaključiti da je rezultat obrade nekog pojma putem TF-IDF algoritma unutar jednog teksta zapravo jedna vrijednost (težinski rezultat, što je veća vrijednost to je riječ relevantnija), ali ako se obrađuje nekakav skup riječi (naslov članka ili riječi u opisu nekog predmeta) onda se formira matrica sa stupcem težinskih vrijednosti za svaki od promatranih dokumenata.

3.1.2 Klasteriranje

Klasteriranje (eng. *clustering* ili *cluster analysis*) je **operacija grupiranja skupa objekata** na način da su objekti u istoj grupi (zvanoj klaster) međusobno sličniji u promatranom smislu, nego onim objektima u drugim grupama, odnosno klasterima. Klasteriranje je jedna od glavnih operacija istraživačkog rudarenja podataka i uobičajena tehnika za statističke analize podataka koje se koriste u mnogim područjima, uključujući prepoznavanje uzoraka, analiza slike, pronalaženju podataka, biometrija, kompresija podataka, računala grafika kao i strojno učenje.

Sama analiza klastera nije jedan specifičan algoritam, već opći zadatak koji je potrebno riješiti. Analiza klastera se može postići raznim algoritmima koji se značajno razlikuju u razumijevanju onoga što čini klaster i kako ih učinkovito pronaći [12]. Popularne predodžbe klastera uključuju grupe objekata s malim razmacima između članova klastera, guste površne podatkovnih prostora ili određene statističke raspodjele. Koji će se algoritmi upotrijebiti, koja će biti gustoća i završni broj klastera ovisi o pojedinačnom skupu podataka kao i željenom rezultatu procesa klasteriranja [12].

S obzirom na brzi rast Interneta sa strane broja korisnika i predmeta koji se prikazuju i nude korisnicima, većina sustava za preporuke koja koristi pristup kolaborativnog filtriranja pati od problema kao što je rijetkost i skalabilnost podataka. Klasteri mogu pomoći kod tog problema, a primjer toga su autori Das, Mukherjee, Majumder i Gupta [13], koji su izradili sustav za preporuke isključivo temeljen na tehnikama klasteriranja podataka kako bi riješili problem skalabilnosti povezan s davanjem preporuka. Temeljna ideja je **podijeliti korisnike** sustava za preporuku koristeći algoritme klasteriranja i **primijeniti algoritam preporučivanja zasebno po svakom klasteru** [13]. Tako se ostvaruje preporučivanje novih predmeta korisniku unutar određenog klastera samo koristeći statistike ocjena ostalih korisnika unutar tog klastera.



Slika 15: Klasteriranje podataka

(Izvor: www.geeksforgeeks.org/clustering-in-machine-learning/)

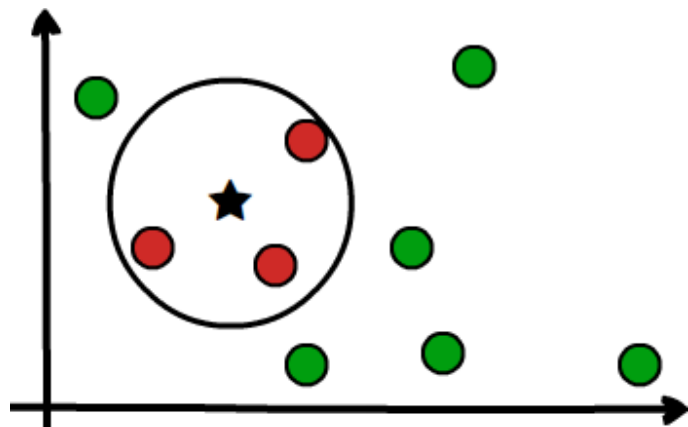
3.1.3 KNN algoritam

KNN je akronim za **algoritam k-najbližih susjeda** (eng. *k-nearest neighbors algorithm*) te je često korišten algoritam strojnog učenja kod pristupa kolaborativnog filtriranja, gdje se koristi za pronalaženje skupina sličnih (objekata) korisnika na temelju zajedničkih ocjena predmeta [14]. Zadatak algoritma je donošenje predviđanja sljedećih korisničkih ocjena koristeći prosječne ocjene k-najbližih susjeda.

Algoritam se može koristiti za klasifikacijske i regresijske probleme predviđanja, dok je u velikoj većini korišten upravo za klasifikacijske probleme u industriji sustava za preporuku. Algoritam odlikuju **jednostavnost razumijevanja izlaznih podataka, kratko vrijeme računanja i prediktivna snaga** [14]. KNN algoritam pronalazi k najbližih stavki određenog objekta na temelju zadane metrike udaljenosti, gdje su tipično korištene Euklidska udaljenost (pravocrtna udaljenost između dvije točke) ili Jaccard-ov koeficijent sličnosti [14]. Kod problema s klasifikacijom za predviđanje klase ili grupe određene instance, prvo se pronalaze k najbližih instanci stavki na temelju zadane metrike udaljenosti, te na temelju sheme većinskog glasanja ili glasovanja s ponderiranom većinom (gdje su bliži susjedi više uvaženi), predviđaju klase.

Primjer

Uzme li se kontekst predviđanja sljedeće serije korisniku Netflix na temelju njegove posljednje pregledane serije, uspoređivanjem atributa poput žanra, broja epizoda, ocjene korisnika i glumačkih uloga pronalaze se top k najbližije serije. Ukoliko se k odredi, na primjer na vrijednost 3, algoritam izvršava pretragu s ciljem pronalaska 3 najbližije serije. Analogno tome, može se izvršiti i klasifikacija korisnika pronalazeći grupe od k najbližijih korisnika te davati im prilagođene preporuke. Što je **koeficijent k veći**, to je i **mogućnost pogreške veća** te se stoga preporučuje algoritam primjenjivati u prikladnom kontekstu [14].



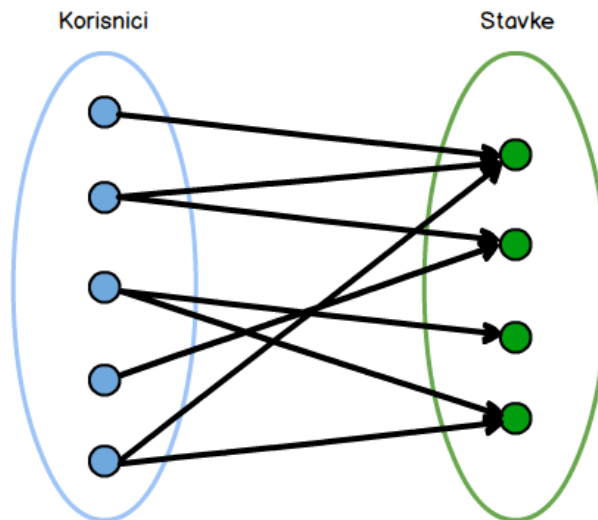
Slika 16: Pronalazak 3 najbližih susjeda
[autorski rad]

3.1.4 Teorija grafova

Generalni problem koji se pojavljuje kod algoritama temeljenih na dostupnim podacima je taj što implementacija takvih algoritama može biti dosta zamorna i komplicirana uzimajući u obzir jako velike brojeve korisnika sustava za preporuku [1]. Taj problem je netrivialan jer se sustavi za preporuke uglavnom i koriste na velikom broju korisnika, te se danas na web stranicama velikih društvenih mreža i online shopova nerijetko nalazi i preko milijardu korisnika, a sustav za preporuke bi ih trebao sve *pokriti*. Može se naslutiti da koristeći konvencionalne metode koje su bazirane na statistici i izračunavanju sličnosti postaju komplicirane za implementirati, odnosno skalirati na puno veći broj korisnika, te sama složenost programskog koda koji se nalazi u pozadini eksponencijalno raste s brojem korisnika koji su uzeti u obzir prilikom izrade algoritma. Stoga, kako bi se **pojednostavio rad** sustava za preporuku, ali i sama **složenost programskog koda** koji ga pokreće, koristi se pristup teorije grafova [15].

Grafovi su jedan od način modeliranja složenih odnosa između mnogih objekata. Jednostavnim rječnikom, graf se sastoji od zbirke čvorova (može se govoriti i vrhova) i bridova (poveznica, lukova) koji ih mogu povezivati. Neki čvor u grafu je samo apstraktna točka te može predstavljati bilo što - osobu, računalo, zgradu itd. Brid povezuje dva čvora i opcionalno može biti usmjeren s tim da se poštuje pravilo da informacije mogu teći samo u jednom smjeru za jedan brid.

Algoritmi za donošenje preporuka zasnovani na teoriji grafova se sastoje od dva koraka: **izgradnja grafa** koji predstavlja dostupne podatke o korisnicima i predmetima te **donošenje preporuka temeljem analize izgrađenog grafa** [15]. Sustavi koji su primjenjivali algoritme na temelju teorije grafova su **pretežno sustavi s pristupom kolaborativnog filtriranja**. Ovisno o domeni sustava za preporuke, podacima koje ima na raspolaganju i vrsti preporuke koju mora donijeti – algoritmi preporuka na temelju grafova su iskorištavali različite vrste grafova. Međutim, svima je zajedničko to da su im glavne komponente grafa odnosi između korisnika i onih stavki koje su ocijenili. Najčešći pristup koji se koristi je izgradnja bipartitne mreže gdje se veze protežu od jednog kraja mreže koji predstavlja korisnike do drugog kraja mreže koji predstavlja predmete [15]. Jednom kad se mreža izgradi, postoji nekoliko različitih načina koji se obično koriste kako bi se rangirali predmeti za preporuku, a obično su to pristupi poput zajedničkih susjeda dvaju čvorova (eng. *common neighbors*), Katz-ova mjera centralnosti te difuzijski rezultati čvorova (eng. *diffusion scores*) [15].



Slika 17: Bipartitni graf [autorski rad]

Na slici poviše moguće je vidjeti primjer bipartitnog grafa gdje jedan dio mreže predstavljaju čvorovi korisnika a drugi čvorovi s predmetima. U ovom slučaju graf se također sastoji i od usmjerenih bridova koji pobliže označavaju odnose između korisnika i predmeta. Primjer jednog takvog odnosa može biti:

Čvor[*Jakov*] : *Brid*[*je kupio*] → *Čvor*[*Call of Duty*]

gdje se jasno može ustanoviti zavisnost između čvorova, odnosno prikazati veza između korisnika i predmeta, što je u ovom slučaju informacija o kupnji video igre.

Sustavi koji rade s pristupom kolaborativnog filtriranja imaju koristi od korištenja algoritama teorije grafova jer oni otklanjaju dosad često spominjane probleme s oskudnosti podataka takvih sustava [15]. Najnoviji pomaci u industriji sustava za preporuku koji rade na temelju teorije grafova donose nove metode koje proširuju bipartitnu mrežu dodajući joj određene slojeve. Neki autori [15] su razmotrili korištenje sesijskog sloja kako bi uzeli u obzir dugoročne i kratkoročne sklonosti korisnika s ciljem bolje vremenski tempiranih davanja preporuka. Također se spominje korištenje različitih tipove čvorova u višeslojnoj strukturi kako bi dali nasumičnu preporuku nasumičnim hodom po grafu [15]. U nekim slučajevima korišten je troslojni graf za poboljšanje preporuka, gdje su uzete u obzir i oznake koje su korisnici dodijelili predmetima, koristeći ocjene temeljene na difuzijskim rezultatima. Neki autori [27, 28] su čak revidirali strukturu mreže, odnosno bipartitnog grafa te smatraju da heterogena mreža u obliku zvijezde gdje bi se mogle ostvariti različite poveznice između korisnika i predmeta bila pogodnija za preporuke. Svakako treba napomenuti da niti jedan od ovih algoritama na temelju teorije grafova nije dizajniran da obuhvati kontekst izbora i preferencija koje korisnik može

imati. Odnosno, velika većina njih ovisi o direktnim dobivenim podacima o predmetu (poput vremenu kupnje i sadržajnom opisu) koje ne moraju uvijek postojati, ili uopće biti dostupne unutar svih aplikacija dok također mogu biti skupe za prikupljanje.

3.2 Tehnologije utemeljene na modelima podataka

Već je prethodno bilo spomenuto kako se tehnologije utemeljene na memoriji uvelike oslanjaju na *jednostavne* mjere sličnosti kako bi zajedno povezale slične korisnike ili predmete. S druge strane, tehnologije utemeljene na modelima podataka idu korak dalje te se bave *pogađanjem* koliko će se korisniku svidjeti predmet s kojim se ranije nije susreo.

Sustavi za preporuke utemeljeni na memoriji nisu uvijek tako brzi i skalabilni kao što bi se to za njih željelo, pogotovo u kontekstu davanja preporuka u stvarnom vremenu na temelju jako velikih skupova podataka. Ako je to također jedan bitan aspekt sustava za preporuku, rješenje je korištenje sustava za preporuku koji koriste tehnologije utemeljene na modelima podataka. Sustavi utemeljeni na modelima podataka uključuju izgradnju modela na temelju skupa podataka o dostupnim ocjenama. Drugim riječima, vade se određeni podaci iz skupova podataka i ti se podaci koriste kao **model** za izradu preporuka, pri tome **eliminirajući potrebu da se svaki put mora prolaziti kroz cijeli skup podataka donošenjem nove preporuke**. Ovaj pristup stoga nudi **prednosti brzine** i **skalabilnosti** kod procesa donošenja preporuka. U kontekstu kvalitete preporuka, činjenica da sustavi utemeljeni na modelima podataka ne koriste sve podatke (cijeli skup podataka) koji su mu dostupni, moguće je da se nekad ne dobiju preporuke točne koliko i one dane od strane sustava utemeljenim na memoriji. Tu svakako treba napomenuti da kvaliteta preporuka koje će nastati ovisi o načinu na koji je model izgrađen, te danas općenito sustavi utemeljeni na modelima podataka postižu bolje rezultate od onih utemeljenih na memoriji. U nastavku će biti obrađene jedne od najčešće korištenih tehnologija utemeljenih na modelima podataka kod sustava koji imaju pristupe kolaborativnog filtriranja i preporučivanja na temelju sadržaja.

3.2.1 Matrična faktorizacija

Matrična faktorizacija je jedan od najpopularnijih algoritama koji se koristi kod sustava za davanje preporuka, tipično s pristupom kolaborativnog filtriranja. Danas postoji više različitih algoritama matrične faktorizacije gdje današnje usavršene verzije spadaju pod tehnologije utemelje na modelima te od tu dolazi i razlog njihovog svrstavanja u ovu skupinu, dok originalna verzija algoritma više nalikuje na tehnologije temeljene na memoriji. Osnovna ideja ovih algoritama je dekompozicija matrice interakcije između korisnika i predmeta na dvije pravokutne matrice manjih dimenzija. Ova obitelj algoritama je postala široko poznata tijekom izazova za Netflix-ovu nagradu zbog njihove učinkovitosti i boljih performansi naspram ostalih tehnologija, o čemu je prvi pisao Simon Funk [16] te svoja istraživanja podijelio s istraživačkom zajednicom.

Izvorni algoritam koji je S. Funk predložio, je **faktorizacija matrice ocijenjenih stavki** od strane korisnika kao **produkt dvije matrice nižih dimenzija**, gdje bi prva sadržavala redak za svakog korisnika a druga stupac za svaku stavku. Redovi ili stupci koji su povezani s određenim korisnikom ili stavkom nazivaju se **latentnim faktorima** [16].

Izvornim algoritmom, predviđene ocjene se mogu izračunati kao

$$\tilde{\mathcal{R}} = \mathbf{H}\mathbf{W},$$

gdje je $\tilde{\mathcal{R}} \in \mathbb{R}^{\text{korisnici} \times \text{stavke}}$ matrica ocjena određenih korisnika za dostupne stavke,

$\mathbf{H} \in \mathbb{R}^{\text{korisnici} \times \text{latentni faktori}}$ matrica koja sadrži korisničke latentne faktore te

$\mathbf{W} \in \mathbb{R}^{\text{latentni faktori} \times \text{stavke}}$ matrica koja sadrži latentne faktore stavke.

Pojedinačno, predviđena ocjena korisnika k za stavku s može biti izračunata kao:

$$\tilde{r}_{ks} = \sum_{f=0}^{n \text{ faktora}} H_{k,f} W_{f,s}$$

Snagu modela je moguće **prilagođavati promjenom broja latentnih faktora**. Pokazano je da matrična faktorizacija s jednim latentnim faktorom je jednaka najpopularnijoj preporuci (preporuka koja se obično daje na početnim stranicama bez ikakve personalizacije)[16]. Povećanje broja latentnih faktora kod korisnika poboljšava personalizaciju te samim time i kvalitetu preporuka, sve dok broj uključenih čimbenika ne postane previsok te dođe do *overfitting*-a kada počinje padati kvaliteta preporuke. Rezultati predviđanja se mogu poboljšati dodavanjem različitih težina latentnim faktorima na temelju popularnosti određenih predmeta ili aktivnosti korisnika [16]. Različita prilagođavanja algoritma i uklanjanje nedostataka metode

kombiniranjem s određenim tehnologijama donose nove metode matrične faktorizacije, od kojih su neke:

- **SVD++** - osim eksplicitnih podataka, odnosno ocjena korisne se i implicitni podaci o stavkama poput lajkova korisnika, podataka o kupnji, preskakanju određenog predmeta kao i dodavanja predmeta na listu želja.
- **Asimetrični SVD** - kombinira prednosti SVD++ te djeluje kao algoritam temeljen na modelu podataka. Na taj način može uzeti u obzir i nove korisnike bez ili s jako malo ocjena bez potrebe da se ponovo trenira model. Uzima u obzir i preferencije korisnika.
- **Hibridna MF** – daje mogućnosti kombiniranja implicitnih i eksplicitnih podataka o interakcijama između korisnika i stavki te može uključiti podatke kako kolaborativnog pristupa tako i podatke karakteristične za pristup temeljen na sadržaju.
- **Deep Learning MF** – kako se u posljednje vrijeme pojavljuje sve više tehnologija koje rade s neuronskim mrežama, ova tehnologija generalizira tradicionalni pristup matrične faktorizacije s nelinearnom neuronskom arhitekturom.

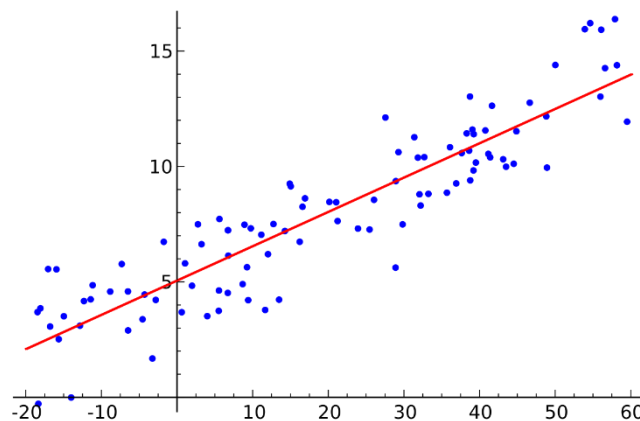


Slika 18: Matrična faktorizacija prema S. Funk [autorski rad prema 16]

3.2.2 Regresijska analiza

U statističkom modeliranju, regresijska analiza se definira kao skup statističkih postupaka za procjenu odnosa između zavisne varijable (često se još naziva i *ishodišna* varijabla) i jedne ili više neovisnih varijabli [17]. Postoji više oblika regresijske analize koji uključuju linearnu regresiju, nelinearnu regresiju, interpolaciju i ekstrapolaciju, dok je najčešće korišteni oblik linearna regresija. Regresijska analiza se prvenstveno koristi u dvije različite svrhe: **predviđanje i prognoziranje** kao prva i široko rasprostranjena, te **zaključivanje uzročno posljedičnih odnosa** između zavisnih i nezavisnih varijabli kao druga i nešto rjeđa.

Do sada se nekoliko puta spomenulo kako izbor pristupa i tehnologija kod stvaranja preporuka značajno utječe na kvalitetu preporuka. Istraživanja autora Mild i Natter [17] pokazuju da količine dostupnih podataka, gdje se fokus prebacio s analize performansi sustava u radu s rijetkim podacima na slučajeve gdje su jako velike količine podataka dostupne, ipak utječu na učinkovitost sustava i samu kvalitetu donesenih preporuka. Poznato je da kolaborativno filtriranje ima probleme kod donošenja preporuka za nove korisnike (problem hladnog starta), međutim autori [17] dolaze do zanimljivih rezultata istraživanja kako prilikom sve većeg broja korisnika, u jednom trenutku se postiže zasićenje te taj pristup ne postiže značajniji učinak daljnjim povećanjem broja korisnika. Suprotno tome, linearna regresija značajno povećava performanse s većim brojem korisnika. Treba svakako napomenuti da je ovaj bolji učinak metode linearne regresije vidljiv tek kod dovoljno velikog broja korisnika, dok točan broj nije definiran od strane autora [17]. Autori [17] također daju objašnjenje da će za različite kombinacije podataka koji se prikupljaju o korisnicima, u različitom trenutku doći i do samog zasićenje poboljšavanja učinka pristupa kolaborativnog filtriranja. Uzmu li se u obzir stalni napori za proširivanjem baze korisnika kod svih sustava za preporuku, regresijska analiza predstavlja vrlo učinkovitu i poželjnu metodu za kasnije faze sustava za preporuku s velikim bazama korisnika.



Slika 19: Linearna regresija

(Izvor: www.wikipedia.org/wiki/Linear_regression)

3.2.3 Stabla odlučivanja

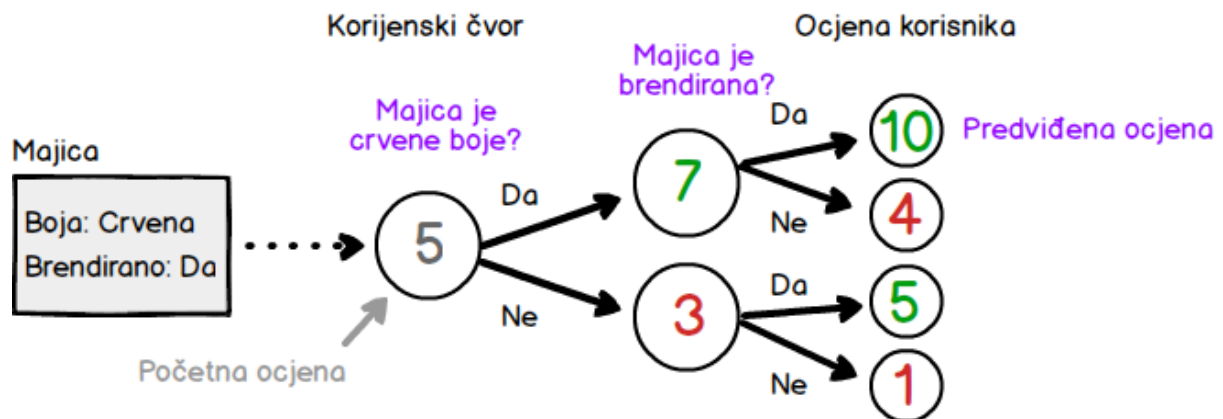
Stabla odlučivanja se već dugo vremena koriste kao tehnologija preporučivanja temeljena na modelu podataka. Korištenje stabla odlučivanja za izradu modela preporučivanja nudi nekoliko **prednosti** poput: **učinkovitosti** (ali ne i preciznosti), **interpretacije** i **fleksibilnosti u rukovanju različitim vrstama ulaznih podataka** (ocjene, demografske informacije, kontekstualni podaci).

Stablo odlučivanja formira prediktivni model koji **mapira ulazni podatak u predviđenu vrijednost** temeljenu na atributima ulaznog podatka [18]. Svakom unutarnjem čvoru na stablu odlučivanja odgovara neki atribut, a svaki luk od roditeljskog čvora do čvora djeteta predstavlja moguću vrijednost ili skup vrijednosti tog atributa. Izgradnja stabla započinje korijenskim čvorom i ulaznim skupom ulaza. Korijenu je dodijeljen atribut te su stvoreni lukovi i podčvorovi za svaki skup vrijednosti. Ulazni skup podataka je zatim podijeljen s vrijednostima tamo da svaki podređeni čvor prima samo onaj dio ulaznog skupa podatka koji odgovara vrijednosti atributa navedenoj u luku prema čvoru djeteta. Postupak se ponavlja za svako dijete dok cijepanje više nije izvedivo. Nad elementima podijeljenih skupova može se dalje primijeniti pojedinačna klasifikacija (koja predstavlja predviđenu vrijednost) ili pak odrediti određeni prag, ovisno o slučaju korištenja [18].

Glavna mana ili slabost korištenja stabala odlučivanja kao modela predviđanja u sustavu za preporuke je potreba za izgradnjom ogromnog broja stabala (bilo to da se gradilo za svaku stavku ili za svakog korisnika). Nadalje model može izračunati očekivanu ocjenu samo za jednu stavku u jednoj iteraciji algoritma (samo za jedan ulaz odjednom). Kako bi se korisniku pružile preporuke, mora se za svaki predmet prijeći stablo (ili više njih) od korijena do krajnjih listova kako bi izračunali njegovi predviđenu ocjenu za taj predmet. Tek nakon što se izračunaju predviđene ocjene svih predmeta, sustav može dati preporuke najviših predviđenih ocjena. Ovo je razlog zašto se stabla odlučivanja ne skaliraju dobro s obzirom na broj predmeta.

Iako su neki autori [18] predstavili model stabla odlučivanja koji bi bio skalabilan za veći broj predmeta, tako da umjesto predviđanja ocjene predmeta stablo odlučivanja vraća ponderiranu (težinsku) listu predmeta za preporuku. Na taj način se pomoću samo jednog prolaska kroz stablo odlučivanja mogu konstruirati preporuke za korisnika no postavlja se pitanje koliko to zapravo utječe na točnost same preporuke zbog variranja broja predmeta koji

će dobiti *najveće težine* u novonastaloj listi kao i moguće promjene težina kod ostalih predmeta prilikom dodavanja novih predmeta u kolekciju [18].



Slika 20: Primjer jednostavnog stabla odlučivanja prema preferencijama korisnika [autorski rad]

3.2.4 Bayesovi klasifikatori

Tehnologije utemeljene na modelu podataka danas generalno postižu bolje rezultate u pogledu točnosti preporuka te performansi i skalabilnosti sustava [19]. Jedna od najpopularnijih tehnologija temeljena na modelu podataka koja se implementira kod sustava za preporuku, tipično s pristupom kolaborativnog filtriranja, je zasigurno matrična faktorizacija zajedno s njenim varijantama. Međutim, glavni problem matrične faktorizacije u kontekstu sustava za preporuke je taj što dobiveni latentni vektori koji su nastali pod utjecajem različitih latentnih faktora (vidjeti poglavlje 3.2.1) nisu lagani niti očiti za protumačiti, te takvi modeli ne mogu direktno objasniti svoje dobivene rezultate [19].

Tu dolaze do izražaja Bayesove mreže, o kojima općenito nešto više u sljedećem poglavlju, dok se ovdje fokus stavlja na najjednostavniji oblik model Bayesovih mreža – *jednostavni* (pridjev koji se još koristi je i *naivni*) Bayesov klasifikator. Jednostavni Bayesov klasifikator jedan je od najuspješnijih algoritama strojnog učenja u mnogim domenama klasifikacije [19]. Unatoč svojoj jednostavnosti, jednostavni Bayesov klasifikator se pokazao i smatra konkurentnim i s mnogim drugim složenim pristupima, posebno kod zadataka kategorizacije teksta [20].

Jednostavni Bayesov klasifikator je nadzirani algoritam višeklasne klasifikacije koji se temelji na primjeni Bayesovog teorema s „*naivnom*“ pretpostavkom uvjetne neovisnosti svakog para

varijabli [20]. Predloženo je nekoliko različitih implementacija jednostavnog Bayesovog klasifikatora koje se uglavnom razlikuju po pretpostavkama koje daju u vezi s podjelom vjerojatnosti [20]. U nastavku će biti opisana osnovna implementacija klasifikatora.

Uz *naivnu* pretpostavku da su značajke neovisne s obzirom na oznaku klase, vjerojatnost da neka stavka pripada klasi j s obzirom na n vrijednosti značajki, $p(class_j | f_1, f_2, \dots, f_n)$ je proporcionalna izrazu:

$$p(class_j) \prod_i^n p(f_i | class_j),$$

gdje se i $p(class_j)$ i $p(f_i | class_j)$ mogu procijeniti iz podataka o treniranju [20]. Kako bi se utvrdila najvjerojatnija klasa neke stavke, izračunava se vjerojatnost svake klase te je stavka dodijeljena klasi s najvećom vjerojatnošću.

Ispod će se navesti primjer definiranja jednostavnog Bayesovog model za klasificiranje kod pristupa kolaborativnog filtriranja, danog od [20]. U ovom modelu, ostali korisnici odgovaraju značajkama, a unosi u matrice odgovaraju vrijednostima značajki. Kako bi se odredila najvjerojatnija klasa ciljanje stavke, koristi se sljedeća formula:

$$\begin{aligned} \text{Class} &= \arg \max_{class_j \in \{Like, Dislike\}} p(class_j | U_1 = Like, U_2 = Dislike, \dots, U_n = Like) \\ &= \arg \max_{class_j \in \{Like, Dislike\}} p(U_1 = Like, U_2 = Dislike, \dots, U_n = Like | class_j) p(class_j) \\ &= \arg \max_{class_j \in \{Like, Dislike\}} p(class_j) \prod_i p(U_i = class_k | class_j). \end{aligned}$$

Iako sama pretpostavka da su značajke neovisne, jednom kad se sazna oznaka klase stavke, nije realna u ovoj domeni, pokazalo se da je jednostavni Bayesov klasifikator optimalan u mnogim situacijama gdje ovakve pretpostavke ne vrijede [19]. Također je empirijski pokazano da je konkurentan mnogo složenijim pristupima i u mnogim drugim situacijama kroz razna istraživanja autora [19]. Još jedna od prednosti jednostavnog Bayesovog klasifikatora je njegova brzina, iz razloga što njegovo vrijeme učenja raste linearno povećanjem broja primjera u podacima o treniranju. Dok se u ovom poglavlju fokus prvenstveno stavio na jednostavni Bayesov klasifikator zbog njegove popularnosti i široke zastupljenosti, postoje i druge verzije Bayesovih klasifikatora poput djelomično naivnog Bayesovog klasifikatora, višedimenzionalnog Bayesovog klasifikatora, stablasto proširenog Bayesov klasifikatora kao i Bayesov klasifikator proširen Bayesovim mrežama. Većina ovih klasifikatora je nastala u nastojanju da se usavrši jednostavni Bayesov klasifikator u različitim specifičnim situacijama kao što je to na primjer rad s nepotpunim podacima.

3.2.5 Bayesove mreže

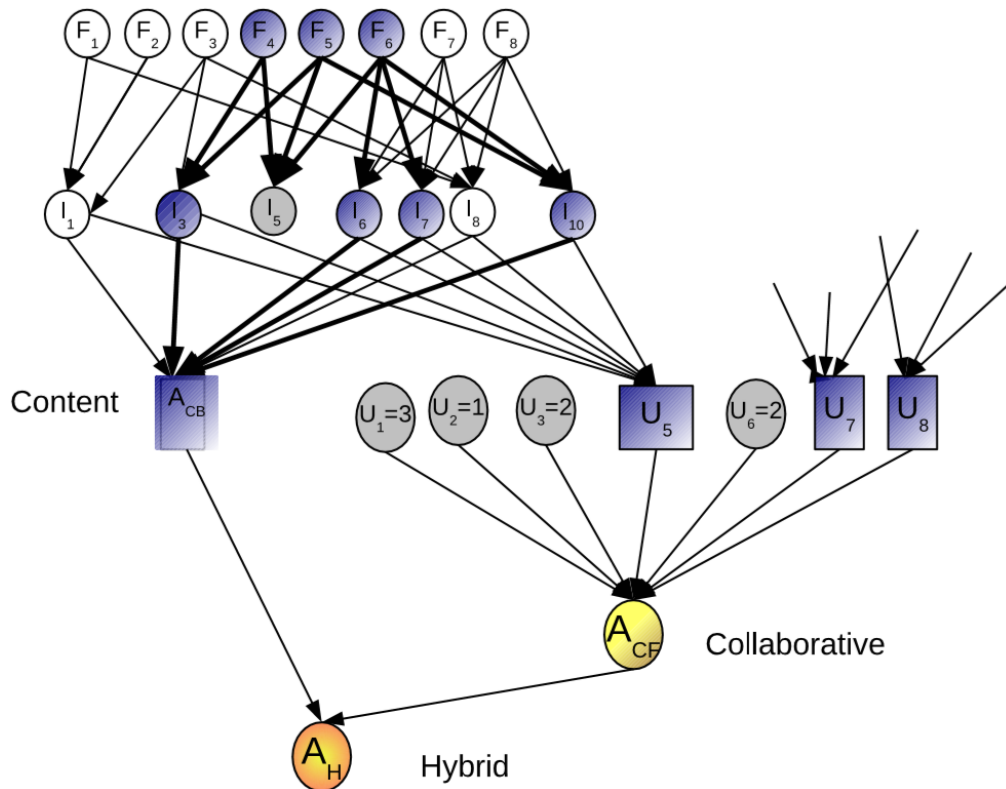
U prethodnom poglavlju su se spominjali Bayesovi klasifikatori kao najjednostavniji oblici Bayesovih mreža. Bayesove mreže (eng. *Bayesian networks*, *Bayes networks*, *belief networks*) su nekakvim pojednostavljenim rječnikom dijagrami koji organiziraju znanje u određenom području mapiranjem između uzroka i posljedica. Bayesove mreže su modeli vjerojatnosti predstavljeni usmjerenim acikličkim grafom gdje **čvorovi** (vrhovi, koji odgovaraju slučajnim varijablama) i **rubovi** (lukovi, strjelice koje povezuju parove čvorova) predstavljaju međusobne utjecaje slučajnih varijabli [21]. Sustavi temeljeni na Bayesovim mrežama su u stanju automatski generirati predviđanja, odnosno donositi odluke čak i ako određene informacije ne postoje [21].

Bayesove mreže su široko rasprostranjena tehnologija koja pronalazi svoju primjenu i u puno različitih domena nevezanih za sustave za preporuku. Razlog tome je mogućnost njihovih korištenja za širok raspon zadataka koji uključuju predviđanje, otkrivanje anomalija, dijagnostiku, zaključivanje, predviđanje vremena te donošenje odluka u slučajevima kada postoje određene nesigurnosti. Razlog spominjanja svega ovoga je kako bi se dao uvid u to koliko su Bayesove mreže zapravo bitna tehnologija o kojoj bi se vjerojatno jedan čitav diplomski rad mogao napisati fokusirajući samo na njih.

Bayesove mreže su vrlo često korištene kod hibridnih sustava za preporuku koji kombiniraju više različitih pristupa, te gotovo uvijek kombiniraju kolaborativno filtriranje te preporučivanje na temelju sadržaja uz moguće pojavljivanje još nekih. Razlog je taj što Bayesove mreže **omogućuju kombiniranje kvalitativnog prikaza problema** (eksplicitni prikaz ovisnosti i sličnosti između predmeta i korisničkih ocjena, kao i ovisnosti i sličnosti između pojedinih korisnika) **s kvantitativnim reprezentacijama** pomoću skupa distribucija vjerojatnosti, mjereći snagu tih odnosa [21]. Korištenje Bayesovih mreža može biti razlog i iz pogleda samog izračuna preporuke kojeg sustav mora obaviti u stvarnom vremenu unatoč mogućem postojanju rijetkih podataka, gdje one daju dobru podlogu za hvatanje u koštac s ta dva izazovna problema.

Na slici 21. se može vidjeti primjer korištenja Bayesove mreže kod hibridnog sustava za preporuku. Već je prije spomenuto kako je Bayesova mreža model vjerojatnosti predstavljen usmjerenim acikličkim grafom koji predstavlja međusobne utjecaje slučajnih varijabli. U ovom slučaju slučajne varijable su i korisnici i stavke kao i značajke koje stavke mogu imati. Sustav

tako prilikom izračunavanja uzima u obzir puno više faktora od onih koje bi uzeo da koristi samo pristup kolaborativnog filtriranja ili preporučivanja na temelju sadržaja [21].



Slika 21: Bayesova mreža kod hibridnog sustava za preporuku [21]

Aktivnom korisniku se izračunava preporuka na način da se analiziranjem ovisnosti značajki kod predmeta s kojima je bio u interakciji ili napravio transakciju, pronalaze najbolje ocijenjeni predmeti koji sadrže te značajke, u isto vrijeme uspoređujući s predmetima koji također sadrže barem većinu tih značajki te su pozitivno ocijenjeni od sličnih korisnika. Ovakav način pronalaska optimalnih predmeta za preporučiti korisniku uvelike povećava vjerojatnost da će korisnik dobro razmotriti preporuku. Također, ovakav sustav otklanja moguće probleme hladnog starta oba pristupa, bilo da se u sustav dodaje novi predmet ili tek registrira novi korisnik, zbog postojanja alternativnih načina izgradnje kvalitetnih preporuka [21]. Prednost je i što se prilikom izračuna preporuke ne koristi cijela mreža (cijeli model podataka) već se analiziraju samo zavisni čvorovi (samo stavke koje sadrže iste značajke i samo slični korisnici)[21].

3.2.6 Neuronske mreže

U prethodnim poglavljima je rečeno nešto više o različitim tehnologijama utemeljenim na modelima podataka te kako se one bave problemima skalabilnosti i hladnog starta kod sustava za preporuke. Moguće pitanje koje se nameće je odabir tehnologije koja rješava sve probleme i koja će biti optimalna za odabir prilikom izgradnje sustava za preporuke.

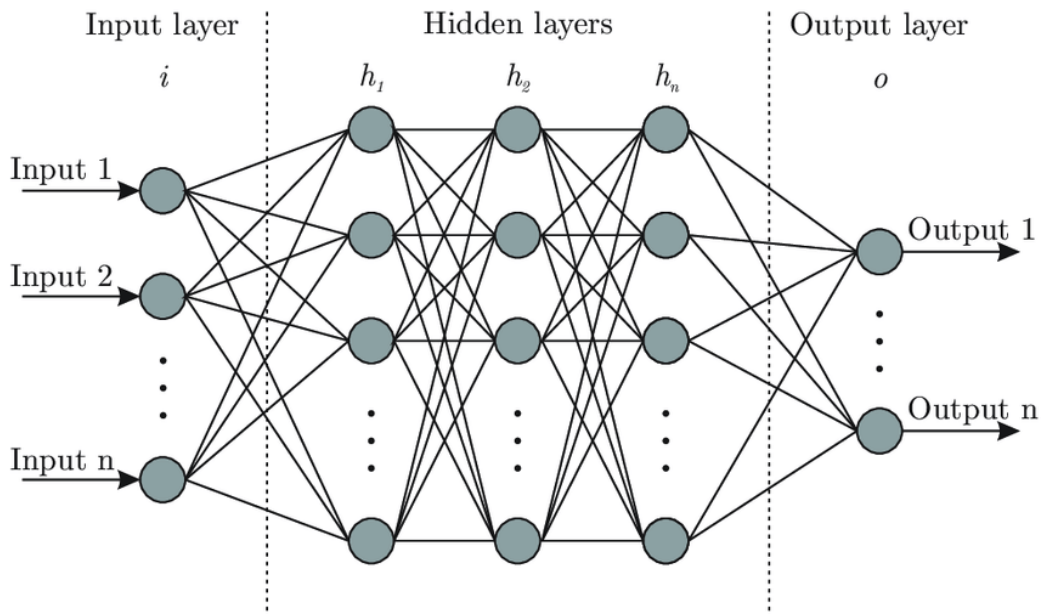
Nažalost, za prethodno spomenute tehnologije se još uvijek ne može reći da su optimalne, uzimajući u obzir njihovu skalabilnost, učinkovitost i točnost kod donošenja preporuka. Autori Ahirwadkar i Deshmukh [22], komentiraju postignute rezultate takvih tehnologija do 2019. godine, te govore kako su takvi modeli linearni te ne uspijevaju naučiti (sve) važne odnose između sličnih predmeta, sličnih korisnika, ali i njih međusobno. Autori navode kako su upravo duboke neuronske mreže pokazale **obećavajuće rezultate** u području sustava za preporuke, u kojima se koriste nelinearne funkcije za učenje složenih odnosa između podataka [22].

Umjetne neuronske mreže (eng. *Artificial neural networks*, skraćena ANN), ili kraće neuronske mreže su računalni sustavi nejasno inspirirani biološkim neuronskim mrežama koje čine životinjski mozak. Takvi sustavi tipično „uče“ obavljati pojedine zadatke razmatrajući primjere, uglavnom bez ikakvog programiranja pravila specifičnih za taj zadatak [23]. Često dan primjer je u prepoznavanju slika, gdje mogu naučiti prepoznavati slike koje sadrže mačke samo analizirajući dane primjere slika na kojima je ručno označeno „*ima mačka*“ i „*nema mačke*“. Zabilježene rezultate analiziranja koriste dalje za prepoznavanje mački na drugim slikama. Originalna namjera pristupa neuronskih mreža je bila rješavanje problema na isti način kako bi to napravio i ljudski mozak, ali s vremenom se fokus pomaknuo na rješavanje specifičnih zadataka te su nastala određena odstupanja od biologije [23].

Neuronska mreža se sastoji od 3 temeljne komponente:

- **neuroni** – umjetne neuronske **mreže su sastavljene od umjetnih neurona**, predstavljajući u određenoj mjeri biološki pojam neurona, koji prima određeni ulaz, daljega kombinira sa svojim unutarnjim stanjem (funkcija aktivacije) te naposljetku stvaraju određeni izlaz, koristeći izlaznu funkciju.
- **težinske veze** – umjetna neuronska **mreža se sastoji od veza**, gdje svaka veza spaja izlaz jednog neurona s ulazom drugog neurona. Svakoj vezi je također dodijeljena određena težina koja predstavlja njenu relativnu važnost. Jedan neuron može imati više ulaznih i više izlaznih veza.

- **funkcija propagacije** – funkcija propagacije ili funkcija širenja **izračunava ulaz u neuron** na temelju vrijednosti izlaza prethodnog neurona i težine veze konekcije koja ih spaja kao ponderiranu sumu. Na taj način težinske veze između neurona, ovisno o njihovoj težini tj. relevantnosti, sljedećem neuronu stvaraju određeno pristrano odstupanje od one vrijednosti koja mu je poslana od njegovog prethodnika.



Slika 22: Arhitektura umjetne neuronske mreže
 (Izvor: www.researchgate.net/publication/321259051)

Ranije je spomenuto kako su duboke neuronske mreže pokazale obećavajuće rezultate u području sustava za preporuke, te se postavlja pitanje koja je razlika između *običnih* neuronskih mreža i onih dubokih. Što se tiče samih komponenti, nema nikakve razlike, međutim duboke neuronske mreže (eng. *deep neural networks*) se tako nazivaju iz razloga što one imaju veći broj skrivenih slojeva (pogledati sliku 22.) te tako omogućuju da algoritam neuronske mreže dublje prouči i analizira ulazne podatke. Taj proces se još naziva *učenjem* (eng. *learning*) te predstavlja prilagođavanje mreže radi boljeg rješavanja zadatka uzimanjem u obzir opažanja odnosno analizu uzoraka na ulazu. Učenje uključuje prilagođavanje težina na vezama između neurona (i stvaranja opcionalnih pragova na ulazima u neuron koji moraju biti zadovoljeni) kako bi se poboljšala točnost izlaznih rezultata [23]. To se postiže minimiziranjem učenih grešaka. Kako dodavanje više skrivenih slojeva direktno utječe i na snagu algoritma neuronske mreže, odnosno kvalitetu izlaznih podataka, tih slojeva može biti jako puno te se proces ovog učenja kod neuronskih mreža popularno naziva *duboko učenje* (eng. *deep learning*).

Dok se svaki proces računalnog učenja popularno naziva strojnim učenjem, u praksi se pojmovi stalno miješaju te se često ne zna gdje završava klasično strojno učenje te ono počinje biti duboko učenje. Za klasično strojno učenje se uzima da algoritmi **obrađuju podatke**, uče iz njih te mogu donositi **odluke na temelju stečenog znanja**. Algoritmi dubokog učenja su **strukturirani u slojevima** i kreiraju *neuronske mreže* koje mogu **samostalno učiti** i donositi inteligentne odluke.

Različitim kombinacijama arhitekture neuronske mreže, kao i različitim načinima na koje će mreža učiti podatke ostvaruju se i određene specifične vrste neuronskih mreža s različito izraženim karakteristikama u koje se neće detaljno ulaziti u sklopu ovog rada. Međutim to je korisno za spomenuti jer se kod korištenja neuronskih mreža vrlo često daje na izbor primjerice broj skrivenih slojeva mreže te odabir različitih algoritama strojnog učenja, čiji izbor utječe i na krajnju robusnost neuronske mreže.

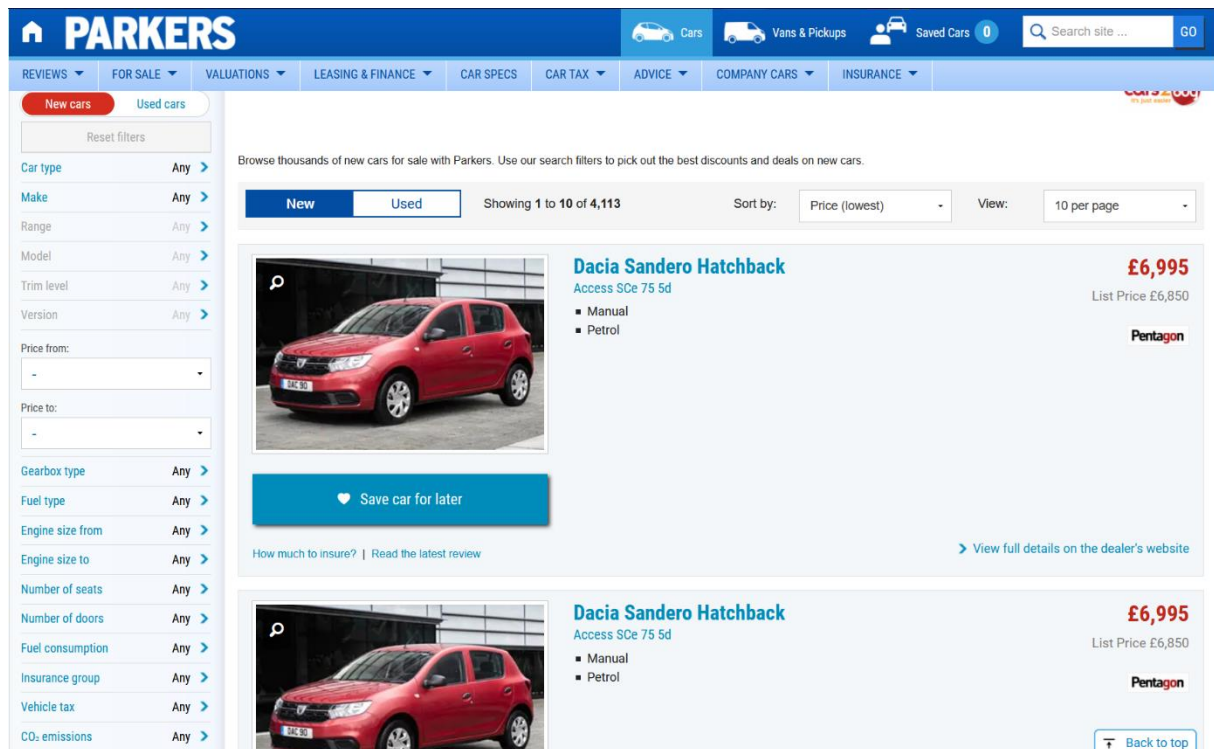
Općenito je dostupna velika količina informacija o predmetima i korisnicima na temelju njihovog korištenja današnjih aplikacija, što se prvenstveno odnosi na mobilne uređaje koji mogu otkriti jako puno o našim navikama. Korištenje što većeg broja dostupnih informacija u procesu donošenja preporuke, donosi sa sobom i bolje razumijevanje preferencija korisnika [22]. Postoji više razloga zašto primijeniti neuronske mreže, a jedan od glavnih bi svakako bi taj što one smanjuju količinu napora koji se mora uložiti kako bi se dizajnirala značajka preporuke jer ih se može istrenirati za obavljanje pojedinih zadataka bez pisanja programskih pravila, već samo korištenjem skupova podataka za treniranje. (*Uče se na testnim skupovima podataka o korisnicima i predmetima gdje im se eksplicitno daje do znanja za koje ulaze se očekuju koji izlazi.*) Drugi dobar razlog zašto ih koristiti je svakako taj što se u model preporučivanja mogu uključiti različite oblike informacija o korisnicima i predmetima, koji mogu uključivati tekst, slike, zvučne zapise pa čak i video zapise [22].

4. Analiza postojećih sustava za prodaju automobila

U sljedećem poglavlju će se analizirati postojeći sustavi za prodaju automobila. Danas postoji jako puno internetskih stranica za prodaju automobila te različite stranice imaju i različite pristupe kod definiranja zahtjeva kupca, odnosno personalizacije rezultata pretraživanja. Polazni kriteriji koji su bili prisutni kod analize postojećih sustava za prodaju automobila, uključivali su različite mogućnosti izražavanja preferencija korisnika kako bi sustav mogao pružiti što prikladnije automobile.

4.1 Provedena analiza

Velika većina postojećih sustava za prodaju automobila funkcionira na način da ulaskom korisnika na stranicu, prikaže listu novih i najpregledavanijih automobila, nudeći korisniku mogućnost izražavanja svojih zahtjeva, odnosno preferencija putem jednostavnih **alata za filtriranje** i uobičajene tražilice po ključnoj riječi. Pozitivni primjer takvog sustava je i **Parkers.co.uk** koji će biti analiziran u nastavku.



The screenshot shows the Parkers.co.uk website interface. At the top, there is a navigation bar with the Parkers logo and various menu items like 'CARS', 'VANS & PICKUPS', and 'SAVED CARS'. Below the navigation bar, there is a search bar and a list of filters. The main content area displays a list of cars for sale. The first car listed is a Dacia Sandero Hatchback, Access SSe 75 5d, priced at £6,995. The listing includes a photo of the car, a 'Save car for later' button, and a link to view full details on the dealer's website. The second listing is identical to the first.

Slika 23: Primjer uobičajenog sustava za prodaju automobila – Parkers.co.uk
(Izvor: snimka zaslona 15.05.2020. sa www.parkers.co.uk/cars-for-sale/new/)

Kako što je već spomenuto, ovo je i najčešći oblik sustava za prodaju automobila te djeluje tako da filtrira automobile u svojoj ponudi onako kako korisnik zadaje nova ograničenja. Na ovaj način korisnik nema dojam da se radi o nekakvom inteligentnom sustavu koji mu predlaže najbolje rezultate, već o jednostavnom filteru koji će mu prikazati one rezultate koji odgovaraju zadanim kriterijima.

Jedan od razloga zašto je odabrana upravo ova stranica kao prvi primjer ovakvih sustava je i nešto bogatiji izbor filtriranja tipičnih kriterija kod kupnje automobila prikazanih tablicom ispod.

Tipični kriteriji za filtriranje automobila	
Kriterij	Opis
Tip/kategorija	Odabir tipa/kategorije automobila klasificira automobile po tipičnim grupama koje uključuju obiteljska vozila, sportska, gradska, <i>coupe</i> , SUV, električna vozila i slično.
Proizvođač	Podatak o proizvođaču automobila.
Seriya	Podatak o seriji automobila određenog proizvođača.
Model	Podatak o modelu automobila određene serije nekog proizvođača.
Verzija	Modeli automobila često imaju i više verzija koje se razlikuju po godini proizvodnje, dodatnoj opremi i slično.
Cijena	Novčana vrijednost automobila.
Vrsta mjenjača	Vrsta mjenjača brzine automobila (ručni i automatski)
Tip goriva	Benzin, diesel, električni, autoplín...
Motor	Vrsta motora (dvotaktni, četverotaktni), broj cilindara i sl.
Broj sjedala	Broj sjedećih mjesta u automobilu.
Broj vrata	Broj vrata na automobilu.
Potrošnja goriva	Potrošnja goriva obično iskazana u potrošnji litara na 100 km za europsko tržište.
Cijena osiguranja	Cijena obaveznog auto osiguranja koja ovisi o više čimbenika
Godišnji porez / Cijena registracije	Godišnji porez na cestovna vozila koji ovisi o snazi motora
Emisija CO2	Količina emisije CO2 koju proizvodi određeni automobil

Tablica 1: Tipični kriteriji za filtriranje automobila
(Izvor: Podaci izvučeni sa www.parkers.co.uk, 2020.)

Dok sustav za prodaju automobila od Parkers nudi pregršt kriterija prema kojima se može pronaći auto po želji korisnika, daje korisniku na mogućnost i korištenje **Car choosing tool** koji radi profiliranje korisničkih potreba, doduše na vrlo oskudan način, te prikazuje listu najbolje ocijenjenih automobila koji odgovaraju prikazanim potrebama korisnika. Tu se može primijetiti određenu ulogu koji ima svojevrсни sustav za preporuku automobila, koji predlaže korisnicima automobile koji će odgovarati njihovim potrebama ali možda neće direktno izaći kao rezultat filtriranja na uobičajen način, za ovo ipak specifično tržište gdje nije pogodno raditi preporuke na temelju prethodnih kupnji.

Which car? The car choosing tool by Parkers

Parkers introduces the "Which car?" car-choosing tool, a brand new car finder tool to help you with your car buying process. With so many models available, choosing a car can be a daunting process. Make your search easier with Parkers' "Which car?" car chooser, and find a vehicle that will perfectly suit your needs. Narrow your search down by body type, price and lifestyle, and the car finder tool will return the models that best fit your requirements.

What is your price range?

Price type: Purchase price Monthly price

£0

£0

What do you want to do with your car?

New drivers
 Company car
 Long journey
 All seasons
 Big boot
 Easy access
 City driving
 Make a statement


What type of car are you looking for?

Convertible
 Coupé
 Electric/Hybrid
 Estate
 Family
 Sports Car
 Hatchback
 People Carrier
 Saloon
 Small/City Car
 Crossover/SUV
 4x4

New cars Used cars

Showing 1 to 10 of 27

Volkswagen Golf GTI (2013 onwards) Review



Parkers rating: ★★★★★ 4.8
New price: £26,715 - £33,995

PROS

- Iconic hot hatch
- VW build quality
- Everyday usability
- Performance


CONS

- Evolutionary looks
- Some rivals sharper

[Read full review](#)

Ford Focus Hatchback (2018 onwards) Review

Very little to fault – a huge step up from accomplished predecessor



Parkers rating: ★★★★★ 4.6
New price: £20,515 - £31,640

PROS

CONS

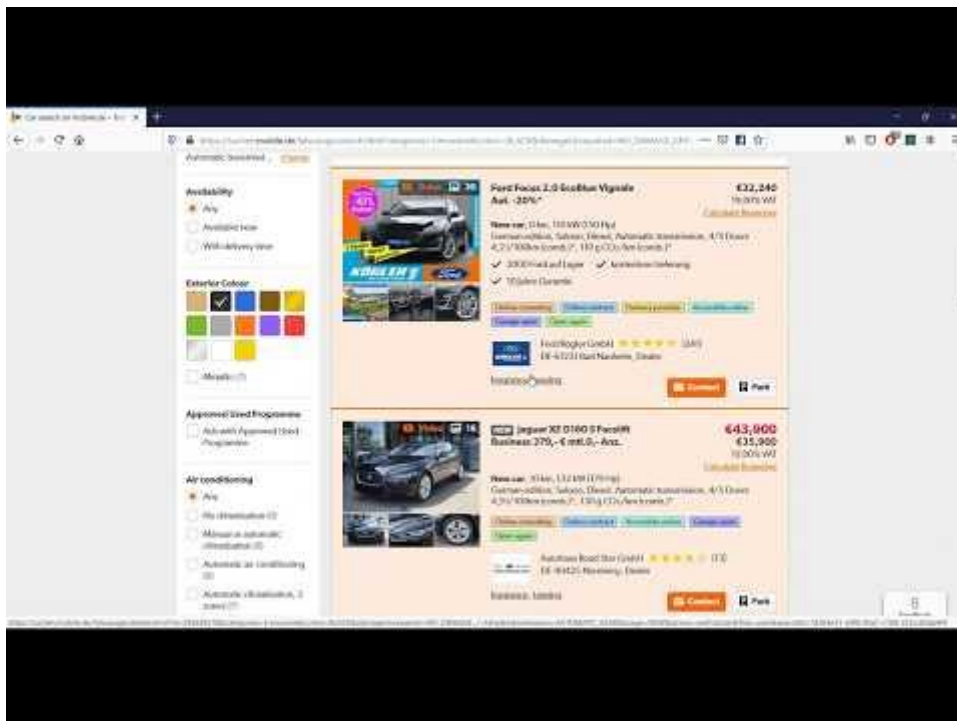
[Back to top](#)

Slika 24: Car choosing tool – Parkers.co.uk

(Izvor: snimka zaslona 15.05.2020. sa www.parkers.co.uk/which-car/)

U nastavku će također napraviti osvrt na jedan od naprednijih sustava za prodaju automobila koji se ipak nalaze na nekoliko stepenica poviše s obzirom na ostale, prvenstveno u pogledu profiliranja korisničkih zahtjeva, odnosno broja uključenih kriterija kod pretraživanja. Sustav koji će se analizirati je **mobile.de**, koji za sebe tvrdi da je najveće njemačko mrežno tržište automobila.

Što se tiče aspekta korisničkog iskustva prilikom dolaska na stranicu, posjetitelj nije odmah serviran s popisom automobila već se prije samog prikaza automobila nudi korisniku **brzi filteri** vrste kao i svrhe vozila, te i napredno pretraživanje putem bogate liste kriterija i opcionalnih značajki dostupnih kod automobila koje će kupac možda želi da svakako budu uključene. To svakako daje jedan poseban **osjećaj** same **personalizacije pretraživanja** i uvelike **sužava** mogući broj vozila u **rezultatima pretraživanja**. Osim osnovnih kriterija koji su već prethodno navedeni u tablici 1., spomenuti sustav uključuje i kriterije poput *broja konjskih snaga automobila* ili *snage u kW*, *boje eksterijera*, *vrste klimatizacije*, opsežnog skupa *značajki dostupnih unutar automobila*, *značajki sigurnosti*, *vrste svjetala*, *vrste guma*, *vrste kočnica*, *dostupnih senzora*, *dodatne opreme* kao i samog *materijala i boje unutrašnjosti*. Kako je ovakav sustav teško predočiti pomoću nekoliko slika, u nastavku će se priložiti i autorski video kako primjer korištenja jednog takvog sustava te pregleda značajki koje nudi.



Video 1: Pregled mobile.de sustava za prodaju automobila

(Izvor: YouTube, Jakov Kristović 2020.)

Ovakav sustav predstavlja manjinu naprednih i sličnih sustava za prodaju automobila koji se mogu pronaći na internetu. Iako ovakav sustav nudi nevjerojatno puno atributa, odnosno kriterija koji se mogu uključiti u korisnička pretraživanja putem označavanja značajki za koje se želi da automobil u listi rezultata svakako sadrži, ovakvi sustavi se ne približavaju nekom kontekstu sustava za preporuke. Sustavi uzimaju u obzir kriterije koje zadaje korisnik ali ne predviđaju automobile koje će preporučiti korisniku s određenom vjerojatnošću, već samo radi filtriranje.

4.2 Rezultati analize

Nastojanje da se pronađe sustav za prodaju automobila koji će ponudi nekakav alat, značajku putem koje bi se korisnika takvog sustava profiliralo pomoću određenog skupa pitanja i saznale njegove preferencije te kao rezultat dala preporuka s nekoliko sustavno predviđenih optimalnih automobila, a ne putem zadavanja preciznih tehničkih specifikacija, nije urodilo plodom. Najbliži primjer toga, iako vrlo oskudan faktorima koji će profilirati preferencije korisnika, je bio *car choosing tool* sa stranica sustava *Parkers.co.uk* koji je opisan prethodno u tekstu.

Ne postojanje adekvatnog sustava za preporuke kod kupaca automobila, koji bi uzeo u obzir potrebe korisnika te pomogao s donošenjem odluke oko kupnje jednog od najskupljih i najkompleksnijih proizvoda koje čovjek kupuje u svom životu, ostavlja velik prazan prostor za napredak mnogim velikim . Kupovina i odabir pravog automobila za pojedinog korisnika nije lagana stvar te vrlo često predstavlja stresno razdoblje i za potrošače koji vole odgovorno raspolagati sa svojim novcem. Isto tako, potrošači se često vode neracionalnim pristupima kod donošenja odluke koji automobil izabrati, te je to danas vrlo česta tema kod mnogih popularnih automobilskih članaka [30, 31]. Kupci često odabiru automobile koji dolaze sa dodatnom opremom u odnosu na one koji ju nemaju, iako možda baš ti automobili bolje odgovaraju potrebama korisnika jer su ekonomičniji, s boljim omjerom cijene i kvalitete ili pak većim prtljažnikom koji je kod nekih kupaca od presudne važnosti. Druge odluke vezane uz tehnički aspekt automobila također nisu jednostavne za kupca, odabrati automobil sa ručnim ili automatskim mjenjačem, pogonom na benzin, dizel ili se odlučiti za električni automobil. Koliko je određeni faktor zapravo važan pojedinom korisniku, koja svojstva automobila su mu manje važna a koja neophodna, kao i problem kako to sve zajedno vrednovati su događaji koji se često pojavljuju, te se sami slučajevi razlikuju od kupca do kupca.

Sljedeće će poglavlje predstavljati glavni dio i temeljni doprinos ovog diplomskog rada, gdje će se postepeno prikazati izrada i krajnji proizvod rane verzije prototipa sustava za preporuku za kupce automobila, kakav trenutno još uvijek nije moguće pronaći. Nastojat će se napraviti kompleksni prototip inteligentnog sustava, koji će izvoditi profiliranje korisnika putem postavljanja jednostavnih pitanja te će na temelju preferencija korisnika preporučiti nekoliko optimalnih automobila.

5. Izrada prototipa inteligentnog sustava za preporuku automobila

Kroz prethodna poglavlja mogao se dobiti uvid u što su zapravo to sustavi za preporuku, koje uloge imaju za vlasnika takvog sustava kao i one koje imaju za samog korisnika sustava. Dao se pregled različitih pristupa preporučivanja kao i različitih tehnologija kojima se sustavi za preporuke mogu koristiti. Također, napravila se i analiza sustava za prodaju automobila koji su dostupni danas te mogao dobiti uvid na koji način oni nude automobile uvidjeti problem nedostatka adekvatnih sustava za preporuke unutar te domene.

Dok je primjena inteligentnih sustava prisutna u mnogo različitih područja danas, te oni dolaze u različitim oblicima poput automatiziranog usisavača Roomba ili pak programa za prepoznavanje lica na mobilnim uređajima. Jednostavnim rječnikom rečeno, inteligentni sustav je **informacijski sustav koji posjeduje određeno znanje (bazu znanja) koje mu omogućuje pokazivanje inteligentnog ponašanja**. Slijedno tome, svaki sustav koji ima određenu bazu znanja te može svoje akcije poduzimati uzimajući u obzir znanje, odnosno podatke na raspolaganju, je zapravo inteligentni sustav. Kako bi sustav koji će generirati preporuke imao odlike inteligentnog sustava, između ostalog, potrebno je imati i bazu znanja.

5.1 Prikupljanje podataka

Za bazu znanja koja će se koristiti prilikom izrade sustava za preporuke u sklopu ovog rada, prikupit će se podaci o **osamnaest marki automobila srednje klase** s nekoliko njihovih varijacijama gdje ukupan broj dolazi do **97 automobila**, koji će služiti kao pojednostavljeni skup testnih podataka. Kako je kod ovog rada fokus zapravo na izradi rane verzije novog softverskog proizvoda, konkretno inteligentnog sustava, a ne na izradi savršenog sustava koji je spreman za produkciju, pretpostavlja se ako sustav ispravno radi s ovim skupom podataka da će onda dobro raditi i sa svim ostalim podacima.

Kako je ideja bila fokusirati se na preferencije korisnika a ne primarno na tehničke attribute vozila, bilo je potrebno kreirati skup podataka, odnosno apstraktni model vozila koji će biti u mogućnosti ponuditi dovoljan broj varijabilnih faktora kod automobila ali i zaobići detaljne tehničke specifikacije. Automobil je vrlo kompleksan objekt kod kojeg postoji jako puno različitih faktora koji razlikuju jedan automobil od drugog. Glavni izvor podataka je bila stranica za recenzije automobila **Edmunds.com**, koja sadrži veliki broj stručnih i neovisnih recenzija, pritom dajući i obilje detaljnih informacija o samom vozilu.

Svaki automobil koji je dospio u finalni skup podataka je promatran na sljedeći način, odnosno sadrži podatke za sljedeće elemente:

- **Name** – jedinstveni naziv za svaki automobil, uključujući proizvođača, model te godinu izdavanja na tržište, prikupljenih s web stranice Edmunds.com i provjerenih na službenim stranicama proizvođača konkretnog automobila.
- **Year** – predstavlja numerički podatak o godini kada je automobil izdan na tržište. Podatak prikupljen s web stranice Edmunds.com.
- **Price** – predstavlja cijenu automobila pohranjenu u numeričkom cjelobrojnom obliku, izvorno prikupljene američkim dolarima (USD) sa stranice Edmunds.com. U sustavu se tijekom rada cijena pretvara u HRK u skladu sa srednjim tečajem.
- **Overall Score** - predstavlja ukupnu ocjenu automobila uzimajući u obzir sve njegove značajke, danu od strane stručnjaka na stranicama Edmunds.com. Sve ocjene dostupne na dotičnoj stranici su u rangu 1.0 – 10.0, dok su u ovom skupu podataka skalabilno prenesene u rang 10 – 100. Vrijedi za sve elemente koji imaju ključnu riječ *Score* u nazivu.
- **Driving Score** – predstavlja ocjenu performansi vožnje automobila uzimajući u obzir faktore poput upravljanja automobilom, suspenzije, držanja upravljača, kočenja i ubrzanja.
- **Comfort Score** – predstavlja ocjenu udobnosti automobila uzimajući u obzir faktore poput kvalitete sjedala, sustava klimatizacije, količine buke koja dopire iz okolnog prometa kao i količinu buke koja se stvara prilikom rada motora.
- **Interior Score** – predstavlja ocjenu izgleda i kvalitete interijera automobila uzimajući u obzir faktor poput dostupnog *infotainment* sustava, mogućnosti i rasporeda različitih kontrola, izgled upravljača i nadzorne ploče te same kvalitete svih komponenti.
- **Technology Score** – predstavlja ocjenu različitih integriranih tehnologija koje su dostupne unutar automobila te ga obogaćuju. Uzima u obzir faktore poput veličine i kvalitete zaslona za prikaz informacija, funkcionalnosti dodirnog zaslona, integracije pametnih uređaja, ozvučenja, glasovnih naredbi, različitih dostupnih senzora i ostalih mogućih jedinstvenih značajki kojima je automobil opskrbljen.
- **Storage Score** – predstavlja ocjenu raspoloživosti prostora kod prtljažnika automobila, uzima u obzir faktore poput dubine i širine prtljažnika te također kvalitete i izvedbe sustava za njegovo ručno/daljinsko otvaranje.

- **Economical Score** - predstavlja ocjenu ekonomičnosti automobila prvenstveno gledajući učinkovitost njegove potrošnje goriva s obzirom na performanse koje automobil nudi.
- **Good Value Score** – predstavlja ocjenu koja odražava odnos cijene i kvalitete konkretne izvedbe automobila, na taj način dajući određenu povratnu informaciju koliko samo podrijetlo proizvođača automobila utječe na cijenu.
- **Fuel Efficiency** – predstavlja kombiniranu efikasnost (potrošnju) goriva automobila izraženu u litrama po 100 prijeđenih kilometara. Prikupljena je numerička cjelobrojna informacija prosječne kombinirane potrošnje kod gradske vožnje i vožnje na autocesti.
- **Seats** – predstavlja numerički podatak o broju dostupnih sjedala u automobilu. Iako velika većina vozila srednje klase su automobili s 5 sjedala, ovaj element služi kako bi korisnik izrazio svoje preferencije kod automobila i u tom pogledu.
- **Transmission** – predstavlja tekstualni podatak o vrsti prijenosa kod automobila. Različite vrste prijenosa koje su dostupne u skupu podataka su *manual*, *automated manual*, *shiftable automatic*, *continuous speed-variable automatic* i *1-speed direct drive* koji je karakterističan za vozila na električni pogon.
- **Horsepower** – predstavlja numerički podatak koji označava broj konjskih snaga automobila.
- **Fuel Type** – predstavlja tekstualni podatak o tipu goriva koje automobil koristi. Tipične vrijednosti dostupne u skupu podataka su *petrol (gas)*, *diesel*, *LPG*, *biofuels*, *hybrid* i *electric*.

Prikaz dijela vlastito izrađenog predloška po kojem su prikupljeni prethodno navedeni faktori je moguće vidjeti na slici 25.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	
car_id	name	year	price	overall_score	driving_score	comfort_score	interior_score	tech_score	storage_score	economical_score	good_value_score	litres_on_100km	seats	transmissi
ID - given in insertion order	display name	release year	MSRP (list price) in USD - \$	1.0-10 edmunds scale to 10-100 (integer)	1.0-10 edmunds scale to 10-100 (integer)	1.0-10 edmunds scale to 10-100 (integer)	1.0-10 edmunds scale to 10-100 (integer)	1.0-10 edmunds scale to 10-100 (integer)	1.0-10 edmunds scale to 10-100 (integer)	1.0-10 edmunds scale to 10-100 (integer)	1.0-10 edmunds scale to 10-100 (integer)	combined consumption city/hwy integer	integer	1) manu 2) auton 3) shiftable 4) 1-spe
All prices, ratings, horsepower and estimated fuel consumption are based on publicly available information from edmunds.com expert reviews.														

Slika 25: Dio vlastitog napravljeno predloška za objekt automobila

(Izvor: snimka zaslona 27.05.2020., alat *MS Excel* 2019)

Na slici 26. se može vidjeti izgled skupa podataka nakon što je samo prikupljanje završeno, gdje su podaci prikupljeni prema prethodno opisanim pravilima, te predloškom.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
car_id	name	year	price	overall_sc	driving_sc	comfort_sc	interior_sc	tech_sc	storage_sc	economic	good_valu	litres_on	seats	transmissi	horsepow	fuel
1	Mercedes-Benz C-Class C 300 S	2020	41400	76	80	70	80	85	70	70	70	10	5	shiftable	255	petrol
2	Mercedes-Benz C-Class AMG C	2020	55950	76	80	70	80	85	70	70	70	13	5	shiftable	385	petrol
3	Mercedes-Benz C-Class C 300 C	2019	51850	76	80	70	80	85	65	65	70	12	4	shiftable	255	petrol
4	Mercedes-Benz C-Class AMG C	2019	67000	76	80	70	80	85	70	65	70	14	5	shiftable	469	petrol
5	Mercedes-Benz C-Class AMG C	2020	77500	76	80	70	80	85	70	65	65	14	4	shiftable	503	petrol
6	BMW 3 Series 330i Sedan (2020	2020	40750	76	80	70	75	70	80	80	75	9	5	shiftable	255	petrol
7	BMW 3 Series M340i Sedan (20	2020	54000	76	80	70	75	70	80	78	75	11	5	shiftable	382	petrol
8	BMW 3 Series 320i Sedan (2018	2018	34900	82	90	90	80	80	75	79	80	10	5	shiftable	180	petrol
9	BMW 3 Series Diesel 328d Wag	2018	46150	80	80	90	82	80	80	82	75	8	5	shiftable	180	diesel
10	BMW 3 Series Hybrid 330e Sedi	2017	44100	80	70	80	85	85	80	80	80	9	5	shiftable	180	hybrid
11	Audi A4 Premium Plus 45 TFSI S	2020	44000	83	85	80	85	90	80	80	80	11	5	shiftable	248	petrol
12	Audi A4 2.0 TFSI Premium Plus	2019	42900	82	85	80	85	90	80	81	80	9	5	shiftable	188	petrol
13	Audi A4 2.0 TFSI Titanium Seda	2019	39200	82	85	80	85	90	80	81	83	9	5	shiftable	188	petrol
14	Audi A4 2.0 TFSI ultra Premium	2017	34900	85	85	85	85	90	80	81	84	9	5	shiftable	190	petrol
15	Volkswagen Passat Wolfsburg I	2019	25295	70	60	65	80	70	85	80	85	10	5	shiftable	174	petrol
16	Volkswagen Passat SE R-Line St	2019	29995	70	60	65	80	70	85	80	85	10	5	shiftable	174	petrol
17	Volkswagen Passat R-Line Seda	2018	24995	70	60	65	80	70	85	80	85	10	5	shiftable	174	petrol
18	Volkswagen Passat SEL Premiu	2018	31650	70	60	65	80	70	85	80	81	10	5	shiftable	174	petrol
19	Volkswagen Passat S Sedan (20	2018	22995	70	60	65	80	70	85	80	87	10	5	shiftable	174	petrol
20	Toyota Avalon Limited Sedan (;	2020	42175	78	75	80	75	80	80	80	80	11	5	shiftable	301	petrol
21	Toyota Avalon XLE Sedan (2020	2020	35875	78	75	80	75	80	80	80	80	11	5	shiftable	301	petrol
22	Toyota Avalon XSE Sedan (2020	2020	38375	78	75	80	75	80	80	80	80	11	5	shiftable	301	petrol
23	Toyota Avalon XLE Sedan (2018	2018	33500	76	70	85	85	75	75	70	70	12	5	shiftable	268	petrol
24	Toyota Camry SE Sedan (2020)	2020	26170	76	75	75	80	75	80	80	75	9	5	shiftable	203	petrol
25	Toyota Camry LE Sedan (2019)	2019	24600	76	75	75	80	65	90	80	76	9	5	shiftable	203	petrol
26	Jaguar XE S Sedan (2020)	2020	39900	70	75	80	65	65	65	75	70	10	5	continuou	247	petrol
27	Jaguar XE R-Dynamic S Sedan (2	2020	46295	70	75	80	65	65	65	75	70	11	5	shiftable	296	petrol
28	Jaguar XE 30t R-Sport Sedan (2C	2019	51635	70	75	80	65	65	75	75	65	11	5	shiftable	296	petrol
29	Jaguar XE 20d Sedan (2019)	2019	38515	70	75	80	65	65	75	80	70	8	5	continuou	180	diesel
30	Jaguar XE 25t Landmark Sedan (2019	48990	70	75	80	65	65	75	75	70	10	5	continuou	247	petrol
31	Jaguar XE 20d Turbodiesel Seda	2018	37225	70	75	80	65	65	75	80	71	8	5	continuou	180	diesel
32	Alfa Romeo Giulia Ti Sedan (20	2020	41400	72	80	75	75	65	60	70	60	11	5	shiftable	280	petrol
33	Alfa Romeo Giulia Sedan (2020	2020	39400	72	80	75	75	65	60	70	61	11	5	shiftable	280	petrol
34	Alfa Romeo Giulia Ti Sport Sedi	2020	46400	72	80	75	75	65	60	70	58	11	5	shiftable	280	petrol
35	Alfa Romeo Giulia Quadrifoglio	2020	74500	72	80	75	75	65	60	65	65	14	5	shiftable	505	petrol
36	Alfa Romeo Giulia Ti Lusso Sed	2019	42995	69	80	75	75	50	60	70	60	11	5	shiftable	280	petrol
37	Ford Fusion SE Sedan (2020)	2020	24500	79	75	80	80	85	85	75	70	11	5	continuou	181	petrol
38	Ford Fusion SEL Sedan (2020)	2020	28690	79	75	80	80	85	85	75	68	11	5	continuou	181	petrol
39	Ford Fusion Titanium Sedan (2f	2020	34450	79	75	80	80	85	85	75	71	11	5	shiftable	245	petrol
40	Ford Fusion Sport Sedan (2019)	2019	40015	79	75	80	80	85	90	65	70	14	5	shiftable	325	petrol

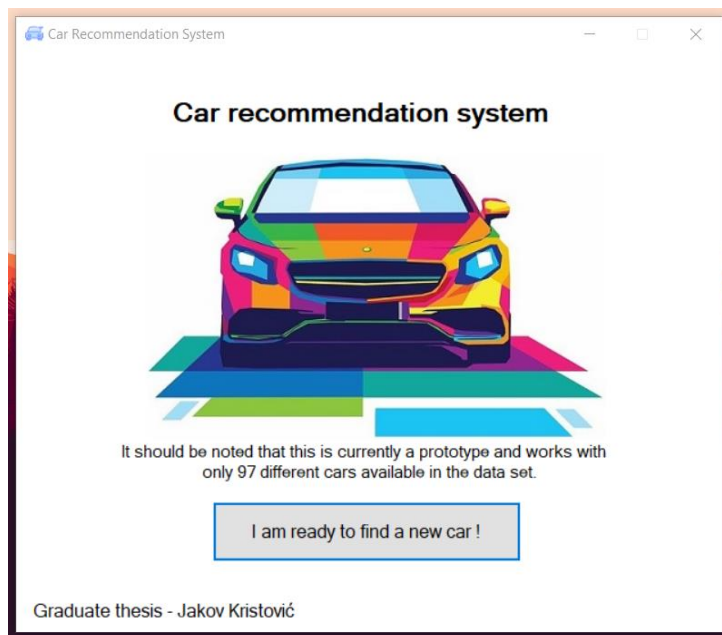
Slika 26: Dio dokumenta završnog skupa podataka

(Izvor: snimka zaslona 27.05.2020., alat MS Excel 2019)

5.2 Profiliranje korisnika

U prethodnom odlomku opisan je način na koji su se prikupljani podaci, te sami podaci, koji će činiti bazu znanja sustava za preporuku automobila koji se izrađuje. Prije samog prelaska na izradu dviju različitih komponenti koje će raditi predviđanje na temelju korisničkih preferencija, potrebne su upravo **preferencije korisnika**. Izrada modela preferencija korisnika će se izvesti putem prikupljanja odgovora na prethodno definirana pitanja, pomoću kojih će biti moguće dobiti uvid koliko su određeni faktori kod automobila zapravo bitni korisniku.

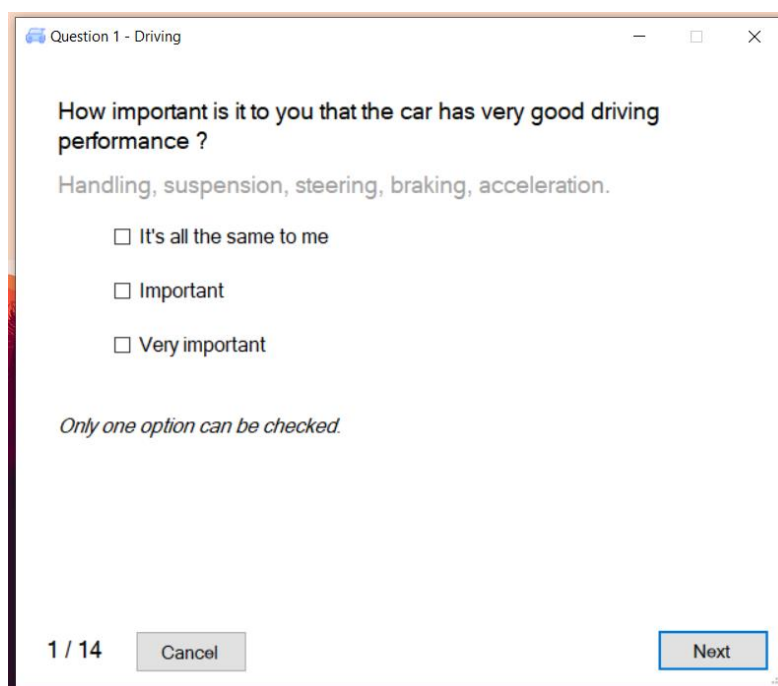
Prvi prozor s kojim se korisnik susreće je prikazan na slici ispod, te predstavlja uvodnu formu aplikacije s kojom se započinje postupak stvaranja korisničkog profila. Ta forma je ujedno zadužena i za inicijalizaciju statičke klase koja će prikupljati preferencija korisnika tijekom korisnikova prolaska kroz aplikaciju, što će biti prikazano malo kasnije u nastavku ovog odlomka. Slike u nastavku su snimke zaslona autorski izrađene aplikacije putem formi unutar .NET Framework-a.



Slika 27: Početni prozor aplikacije

Nakon što korisnik započne profiliranje svojih preferencija kod automobila napuštanjem uvodne forme, korisnik se susreće s **14 različitih pitanja** na koje je potrebno dati odgovor. U nastavku će se ukratko prikazati samo različiti oblici pitanja koji se pojavljuju tijekom profiliranja, dok će prikaz cijelog profiliranja kao i rada same aplikacije biti moguće pregledati na kraju ove cjeline putem video zapisa.

Prvi oblik pitanja s kojim se korisnik susreće je pitanje s tri moguća odabira, gdje korisnik mora dati odgovor na pitanje koliko mu je konkretni faktor kod automobila važan. Tu korisnik odabire između opcija „Svejedno“, „Važno“ i „Jako važno“, gdje smije odabrati samo jednu opciju.



Slika 28: Pitanje važnosti pojedine komponente

Drugi oblik pitanja s kojim se susreće korisnik je pitanje s izražavanjem preferencija kod određenih komponenti. Primjer toga je i pitanje *preferirane vrste prijenosa* na slici ispod, gdje korisnik može izraziti svoje osobne preferencije kod određene komponente automobila, ili pak označiti da nema preferencija za tu komponentu.

Question 9 - Transmission system

Preferred transmission system in the car ?

- Manual
- Automated manual
- Continuously speed-variable automatic
- Shiftable automatic
- No preferences

Only one option can be checked.

9 / 14 Cancel Next

Slika 29: Izražavanje osobnih preferencija

Treći oblik pitanja s kojim će se korisnik imati prilike susreti je pitanje s određivanjem nekih parametara važnih kod stvaranja preporuke, kao što je predviđeni budžet ili primjerice željeni broj konjskih snaga kod automobila.

Question 12 - Budget

The ideal budget you have envisioned for buying a new car amounts to ?

Enter amount: kn (HRK)
(Integer value, e.g. 320000, 250000)

NOTICE:
Dataset currently has 97 records of middle class sedans, where lowest car price available in dataset is 154 600 kn, and highest is 542 500 kn.
Average dataset car price is 260 000 kn.

12 / 14 Cancel Next

Slika 30: Određivanje različitih parametara - budžet

Na slici ispod se može vidjeti klasa *EvaluationModel.cs* koja predstavlja model preferencija korisnika te popunjava na način kako korisnik daje odgovore na konkretna pitanja. Pitanja za profiliranje prikupljaju preferencije koje pokrivaju potpuno cijeli apstrahirani objekt automobila kakav se nalazi u skupu podataka te se na temelju obrade preferencija mogu donositi jako personalizirane preporuke.

```
static class EvaluationModel
{
    // Not considered in any model making predictions
    public static int ID = 0;

    public static int? Year;

    public static int? Price;

    public static int OverallScoreCoef = 1;
    public static int? DrivingScoreCoef;
    public static int? ComfortScoreCoef;
    public static int? InteriorScoreCoef;
    public static int? TechnologyScoreCoef;
    public static int? StorageScoreCoef;
    public static int? EconomicalScoreCoef;
    public static int? GoodValueScoreCoef;

    public static string FuelEfficiency;

    public static int? Seats;

    public static string Transmission;

    public static int? Horsepower;

    public static string FuelType;
}
```

Slika 31: Model preferencija korisnika

5.3 ML .NET

Kod izrade prototipa sustava za preporuku koriste se dva načina izrade preporuka. Prvi način za koji će se dati pregled je izrada algoritma strojnog učenja pomoću ML .NET razvojnog okvira (eng. *framework*) za strojno učenje kod .NET tehnologija.

ML .NET je besplatan software koji dolazi u obliku opcionalnog dodatka koji se može integrirati u Visual Studio alatu. ML .NET se može koristiti s C# i F# jezicima te omogućava provedbu strojnog učenja. Radi se o vrlo modularnom, odnosno proširivom okviru koji podržava različite druge popularne okvire i biblioteke za strojno učenje poput TensorFlow, ONNX, Infer.NET i sličnih. Iako je sami okvir relativno nov (prva dostupna testna verzija se pojavila sredinom 2018. godine) već postoji kao software u stabilnom izdanju koji ima za cilj

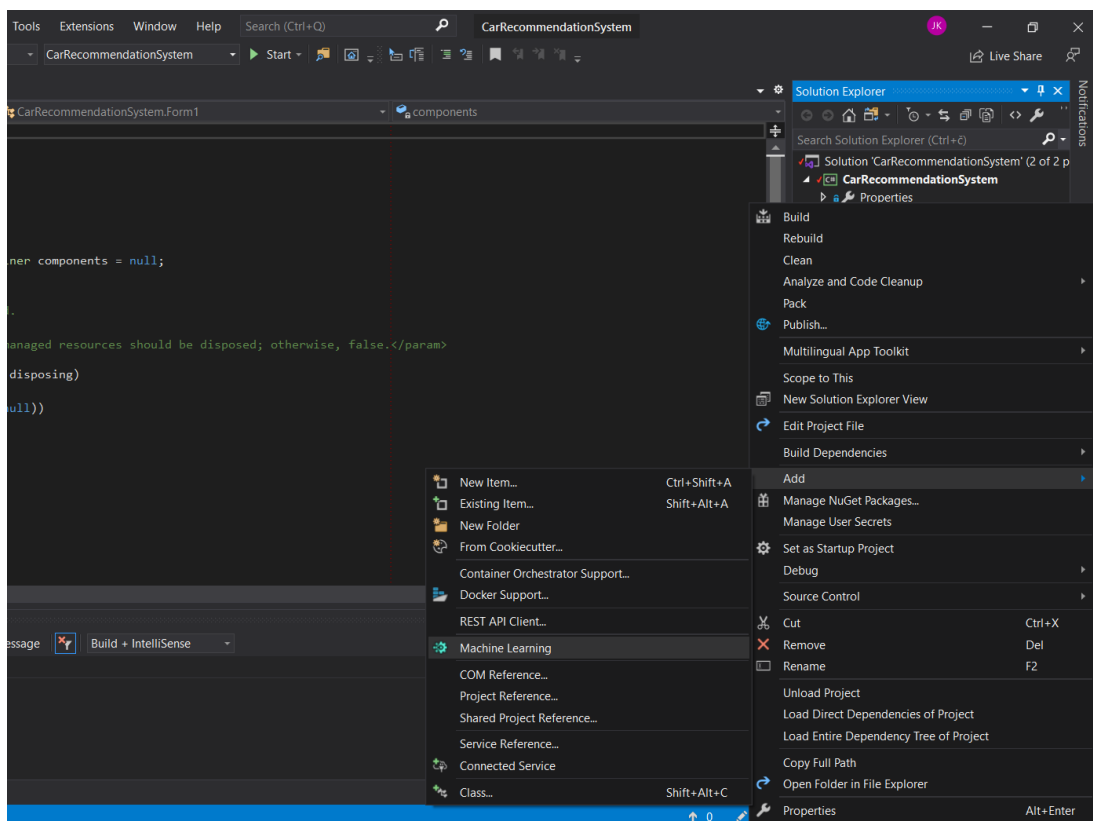
pojednostaviti način kreiranja modela strojnog učenja na proces koji se sastoji od samo nekoliko klikova, za razliku od također jako popularne Python biblioteke TensorFlow koja ipak zahtijeva minimalno poznavanje pisanja naredbi u Pythonu, kao i određenih statističkim metoda (ovisi o konkretnom slučaju). Iako se ovaj okvir još uvijek razvija i stalno unaprjeđuje dodavanjem novih mogućnosti i poboljšavanje performansi, njegove mogućnosti su bile sasvim dovoljne za izradu ovog prototipa.

Što se tiče samih performansi, za ML .NET je dokazano da postiže brze i sjajne rezultate prilikom treniranja modela s velikim skupovima podataka, kao što je to primjerice popularni Amazon-ov skup podataka s recenzijama velik 9 GB, koji se i inače koristi za takva testiranja, gdje je ML .NET postigao razinu pouzdanosti od 95% [24].

5.3.1 Izrada komponente strojnog učenja

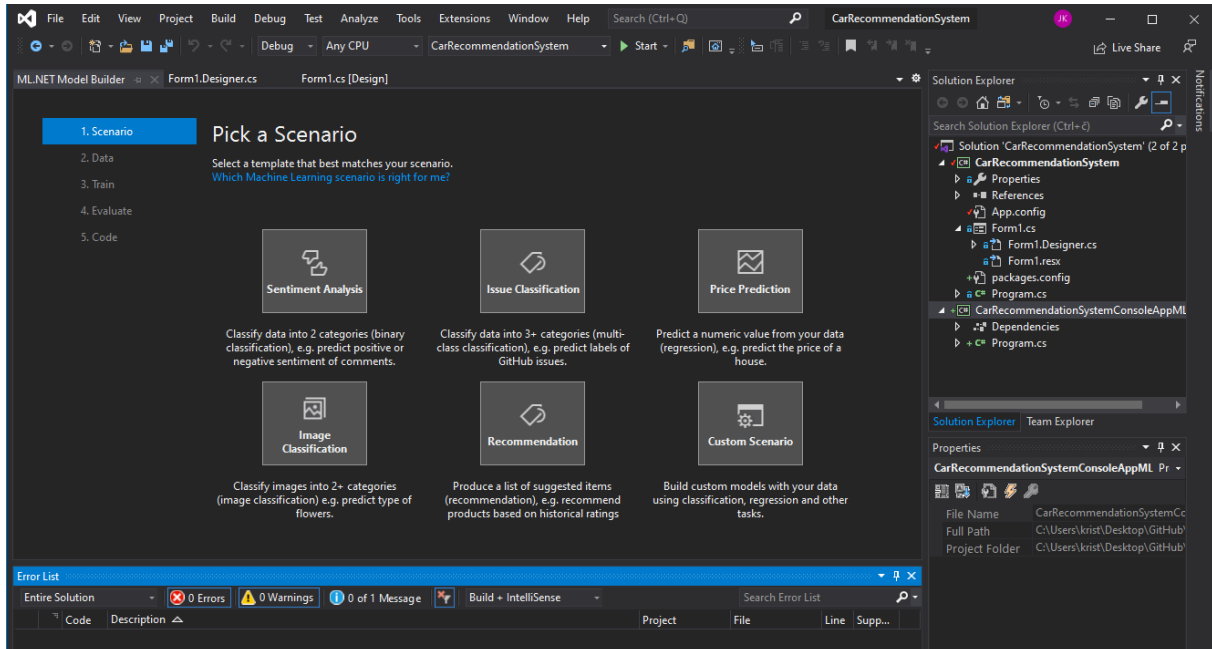
Kod izrade komponente strojnog učenja se koristio prethodno opisani skup podataka. U nastavku će biti prikazani osnovni koraci vezani za postupak izrade modela strojnog učenja kroz jednostavno sučelje koji nudi ML .NET razvojni okvir.

Prvi korak je pokretanje *ML .NET Model Builder*-a unutar *Solution*-a koji sadrži projekt sustava kojeg izrađujemo. Desnim klikom na *Solution* se odabire opcija *Add* te nakon toga *Machine Learning*.



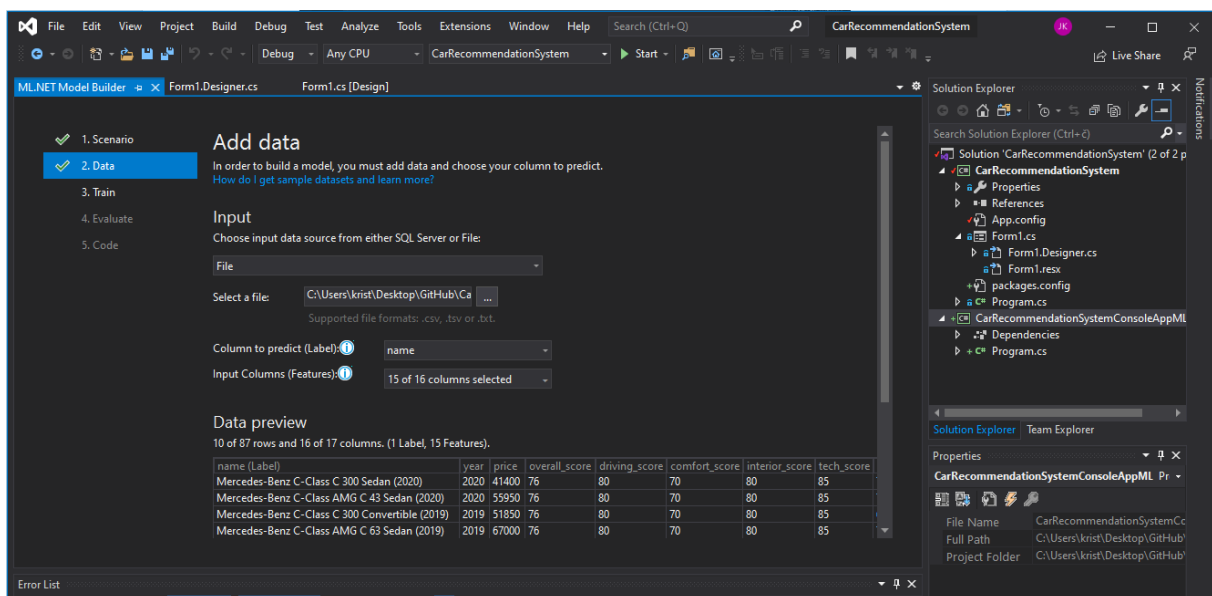
Slika 32: Model Pokretanje ML .NET Model Builder-a

Klikom na *Machine Learning* otvara se ML .NET Model Builder prozor vidljiv na slici ispod teksta. Builder nudi odabir jednog od predviđenih scenarija dok u ovom konkretnom slučaju niti jedan ne odgovara potrebama sustava koji se izrađuje te se odabire *Custom Scenario* opcija.



Slika 33: ML .NET – odabir scenarija

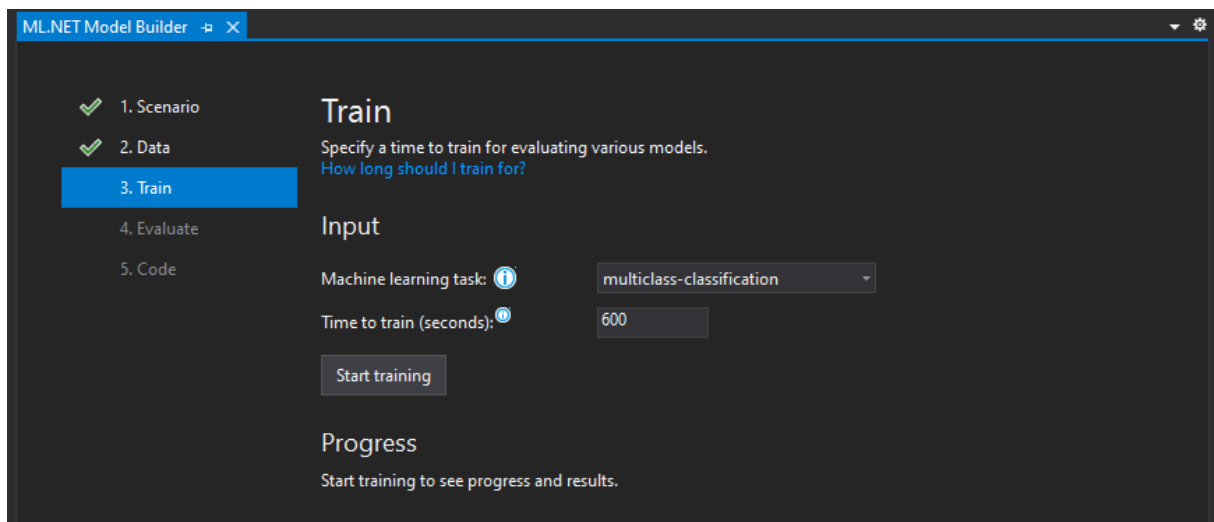
Sljedeći korak je odabir datoteke ulaznih podataka pomoću kojih se trenira model. Također je potrebno odabrati atribut za koji će se raditi predviđanje, što je u ovom slučaju **naziv automobila**, te odabir atributa koji ulaze u model za predviđanje. U ovom slučaju se odabiru svi atributi koji se prethodno opisali nešto poviše u tekstu, osim atributa ID-a kojeg sadrži svaki zapis (on se u aplikaciji koristi za prikazivanje slika automobila).



Slika 34: ML .NET – dodavanje skupa podataka za treniranje

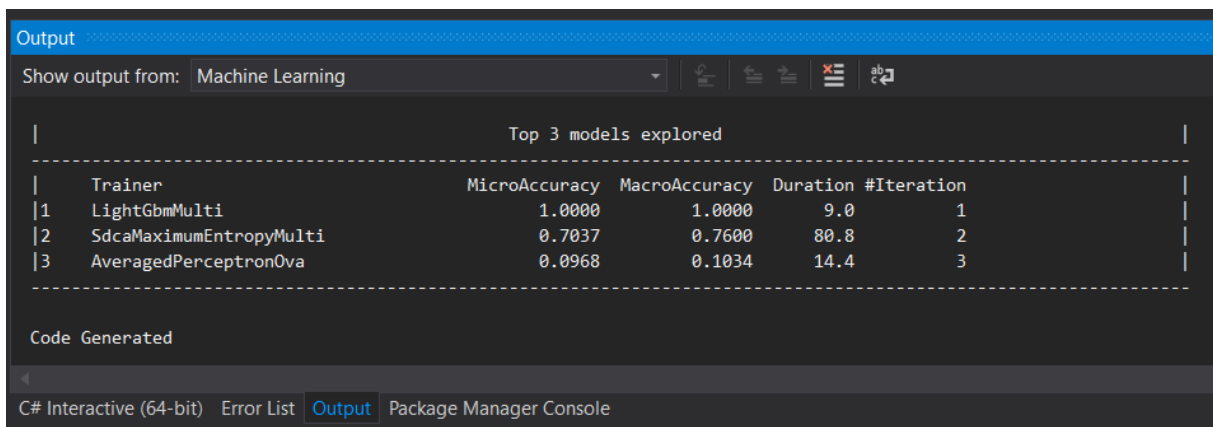
Nakon što se dodao skup podataka za treniranje i kreiranje modela strojnog učenja, potrebno je izvršiti trening. ML .NET ovdje ima zanimljiv pristup, gdje se korisniku nudi na odabir zadatak koji treba napraviti, dok će on provesti trening za skup podataka sa svim karakterističnim algoritmima strojnog učenja za rješavanje tog zadatka, te prikazati koji od njih nudi najbolju pouzdanost za konkretni model.

Kako je skup podataka s kojim se radi sastavljen tek od 97 različitih zapisa, ne može se očekivati da će se treningom nad tim skupom podataka stvoriti jako kvalitetan model, ali bit će dovoljno dobar za potrebu kreiranja prototipa ovog sustava. Vrijeme treniranja se postavlja na 10 minuta zbog malog broja zapisa u skupu podataka, što će biti više nego dovoljno za treniranje na nekoliko desetaka različitih algoritama. Zadatak koji je potrebno obaviti je višeklasno klasificiranje, jer se želi da model može odrediti klasu (naziv automobila), odnosno najslabiju klasu s obzirom na faktore koje će mu se proslijediti nakon profiliranja korisnika.



Slika 35: ML .NET – treniranje modela podataka

Rezultat treniranja različitih modela podataka korištenjem različitih algoritama treniranja je vidljiv na slici 36. Komentari rezultata se nalaze ispod slike.



Slika 36: ML .NET – 3 najbolja modela

Zbog malog broja zapisa tek 3 algoritma ulaze u popis najbolje treniranih modela strojnog učenja za donošenje predviđanja. Zanimljiv podatak koji je moguće pročitati iz rezultata treniranja je taj kako je jedan algoritam uspio postići preciznost od 100%.

Razlog tome je vrlo mali broj zapisa (97) te veliki broj atributa svakog zapisa (15). Algoritam je uspio kreirati model koji može jednoznačno odrediti klasu automobila s obzirom na njegove atribute. Iako je to anomalija te se gotovo nikada u praksi neće dogoditi, to ne znači da će preciznost, odnosno pouzdanost kod predviđanja biti 100% kod kreiranog sustava. Naime, kako će korisničke preferencija varirati, tako će varirati i kreirani profil automobila kojeg će se pokušati predvidjeti koristeći ovaj model. Testiranje stvarnih performansi slijedi u sljedećem poglavlju.

5.3.2 Predikcija pomoću strojnog učenja

U nastavku će se demonstrirati predikcija pomoću prethodno kreiranog modela strojnog učenja. Kod testiranja predikcije korištene su nasumično odabrane vrijednosti kod atributa automobila, gdje se posebno vodilo računa da vrijednosti ne odgovaraju u potpunosti niti jednom dostupnom zapisu u skupu podataka. Rezultati su vidljivi na slici ispod.



```
C:\Program Files\dotnet\dotnet.exe
Using model to make single prediction -- Comparing actual Name with predicted Name from sample data...

year: 2020
price: 28500
overall_score: 75
driving_score: 75
comfort_score: 80
interior_score: 75
tech_score: 70
storage_score: 85
economical_score: 80
good_value_score: 75
litres_on_100km: 10
seats: 5
transmission: shiftable automatic
horsepower: 250
fuel: petrol

Actual Name: This are custom made user preferences for the car with budget
Predicted Name value: Mazda 6 Grand Touring Sedan (2020)
Prediction accuracy: 46.37841%
Overall prediction accuracy: 39.85425%
Total number of records: 97
Predicted Name scores: [0.004559736,0.003794426,0.002189931,0.001840669,0.001351977,0.01983378,0.01486225,0.002705049,0.001783047,0.001221346,0.001498073,0.001803244,0.001803362,0.002631414,0.002792556,0.004229635,0.002200023,0.001879883,0.0008269773,0.01327935,0.02442214,0.01517387,0.001665099,0.006655615,0.003936886,0.005454918,0.002866219,0.002940412,0.001692706,0.002038393,0.002126592,0.003950809,0.005277415,0.001727827,0.002338104,0.0006531371,0.003106207,0.001819634,0.0354642,0.002221123,0.000838994,0.00125823,0.007895811,0.000989935,0.001191269,0.003009976,0.001409873,0.007280777,0.4637841,0.002005538,0.003519193,0.002428946,0.001843592,0.002245309,0.001257012,0.0009592145,0.001224147,0.001986846,0.001713829,0.00265583,0.004881953,0.01299419,0.01624516,0.01122866,0.00230256,0.001884116,0.003671929,0.003085326,0.002676004,0.002102643,0.004411115,0.001496112,0.002982728,0.001097219,0.002835306,0.001674413,0.0009422867,0.001178471,0.001445842,0.03425523,0.001188594,0.001729624,0.001857177,0.002015202,0.03663875,0.004874677,0.07107948,0.002072181,0.003458522,0.003010162,0.001561389,0.003607631,0.01048704,0.004304912,0.000759338,0.009965799,0.005881763]

===== End of process, hit any key to finish =====
```

Slika 37: ML .NET – predikcija pomoću modela strojnog učenja

Model koji je prethodno kreiran je postigao pouzdanost nešto veću od 46.37% kod konkretnog primjera ulaznih podataka na slici, dok je ukupna pouzdanost nakon 5 izvedenih predikcija iznosila nešto više od 39.85%.

Pouzdanost kreiranog modela je oko 40%, što nije zadovoljavajuće ukoliko bi se taj model trebao primijeniti u nekom od stvarnih sustava za preporuke, gdje ta pouzdanost obično mora minimalno biti čak i preko 85-90%. Ponovno se napominje, s obzirom na izrazito mali broj zapisa u skupu podataka s kojim se radi, nije realno niti očekivati veliku pouzdanost.

Detalj koji se namjerno preskočio u ovom odlomku, a potrebno ga je spomenuti je taj da nakon završetka treniranja, ako je developer zadovoljan, može dodati model u *Solution* aplikacije koju kreira. Jednostavnim klikom na „*Add to solution*“ se dodaju projekti koji sadrže logiku iza kreiranog modela strojnog učenja.

Novonastala klasa koju je potrebno poznavati je *ModelOutput*, te ona predstavlja objekt rezultata predviđanja na temelju ulaznih podataka. Ulazne podatke predstavlja *ModelInput* klasa, čiji objekt će se popuniti vrijednostima preferencija korisnika prilagođavanjem vrijednosti dobivenih putem profiliranja korisnika. Klasa *ConsumeModel* predstavlja klasu pomoću koje se kreirani model zapravo koristi za predikciju, i to putem metode *Predict* u koju se prosljeđuje jedna instanca *ModelInput* klase.

Naredba za predikciju glasi:

```
ModelOutput output = ConsumeModel.Predict(input);
```

U instanci klase *ModelOutput* je sadržan rezultat predikcije s pripadajućim postotcima predviđanja. U atributu *output.Score* će biti sadržana pouzdanost predviđanja, a *output.Prediction* će sadržati naziv automobila.

5.4 Algoritam bodovanja

U prethodnom poglavlju je opisana izrada i predikcija pomoću algoritma strojnog učenja dok će unutar ovog poglavlja biti opisan algoritam bodovanja koji je osmišljen specifično za prethodno navedeni oblik apstrahiranog modela automobila unutar skupa podataka kako bi davao što bolje rezultate.

Iako će ovaj algoritam dati puno preciznije i pouzdanije preporuke od onoga temeljnog na strojnom učenju, u stvarnom svijetu neće predstavljati praktičan način rada s velikim skupovima podataka koji se sastoje od nekoliko milijuna zapisa zbog vremena i resursa koje će morati potrošiti kako bi izračunao ukupne bodove za svaki zapis u skupu podataka posebno. U takvim situacijama dolazi do izražaja model strojnog učenja koji će se, nakon što se obavi trening i kreira vjerodostojan model na temelju svih tih podataka, poslije koristiti za brzo

generiranje predikcija na temelju ulaznih podataka jer će moći koristiti kreirani model te izbjeći potrebu rada sa svim zapisima unutar skupa podataka.

Nakon davanja odgovora na sva pitanja te popunjavanja svih atributa prethodno spomenute statičke klase *EvaluationModel.cs*, algoritam za bodovanje će dodijeliti bodove svakom zapisu unutar skupa podataka te prikazati 3 najbolja korisniku. Prije samog objašnjavanja na koji način algoritam bodovanja zapravo radi, slika ispod će prikazati kako se zapravo dodjeljuje vrijednost atributima klase za evaluaciju. U ovom konkretnom slučaju se provjerava odgovor na pitanje u vezi važnosti faktora performansi kod vožnje automobila.

```
1 reference
private void AssignEvaluationModelValue()
{
    if (cbVeryImportant.Checked)
        EvaluationModel.DrivingScoreCoef = EvaluationHelper.VeryImportantCoef;
    if (cbImportant.Checked)
        EvaluationModel.DrivingScoreCoef = EvaluationHelper.ImportantCoef;
    if (cbNeutral.Checked)
        EvaluationModel.DrivingScoreCoef = EvaluationHelper.NeutralCoef;
}
```

Slika 38: Dodjeljivanje vrijednosti atributu unutar klase evaluacije

Ovisno o važnosti pojedinog atributa koji se ispituje, modelu evaluacije se dodjeljuje i veći koeficijent koji će napraviti težinsku korekciju tog dijela bodova. Slike u nastavku prikazuju različite načine evaluacije različitih atributa kod zapisa automobila.

```
1 reference
55 public static void EvaluateList(List<CSVCarModel> carList)
56 {
57     foreach (var car in carList)
58     {
59         car.Score = 0;
60         EvaluateCoefficients(car);
61         EvaluateSeats(car);
62         EvaluateFuelEfficiency(car);
63         EvaluateTransmission(car);
64         EvaluateHorsepower(car);
65         EvaluateFuelType(car);
66         EvaluateBudget(car);
67         EvaluateReleaseYear(car);
68     }
69 }
70
71 1 reference
72 private static void EvaluateCoefficients(CSVCarModel car)
73 {
74     int drivingEval = car.DrivingScore * (int)EvaluationModel.DrivingScoreCoef;
75     int comfortEval = car.ComfortScore * (int)EvaluationModel.ComfortScoreCoef;
76     int interiorEval = car.InteriorScore * (int)EvaluationModel.InteriorScoreCoef;
77     int technologyEval = car.TechnologyScore * (int)EvaluationModel.TechnologyScoreCoef;
78     int storageEval = car.StorageScore * (int)EvaluationModel.StorageScoreCoef;
79     int economyEval = car.EconomicalScore * (int)EvaluationModel.EconomicalScoreCoef;
80     int goodValueEval = car.GoodValueScore * (int)EvaluationModel.GoodValueScoreCoef;
81     int overallEval = car.OverallScore * EvaluationModel.OverallScoreCoef;
82
83     car.Score += drivingEval + comfortEval + interiorEval + technologyEval + storageEval
84                 + economyEval + goodValueEval + overallEval;
85 }
```

Slika 39: Evaluacija početnih ocjena kod pripadajućih atributa automobila

Slika 39. pokazuje način na koji se ukupnom broju bodova kod automobila dodaju korigirane težinske vrijednosti ocjena za početne vrijednosti bodova.

```
115 | 1 reference  
    | private static void EvaluateTransmission(CSVCarModel car)  
116 | | {  
117 | |     if (EvaluationModel.Transmission == "any")  
118 | |         car.Score += 25;  
119 | |     else  
120 | |     {  
121 | |         if (EvaluationModel.Transmission == car.Transmission)  
122 | |             car.Score += 25;  
123 | |     }  
124 | | }  
125 | |  
126 | 1 reference  
    | private static void EvaluateHorsepower(CSVCarModel car)  
127 | | {  
128 | |     // If car has more horsepower, it gets some score points  
129 | |     if ((int)EvaluationModel.Horsepower <= car.Horsepower)  
130 | |         car.Score += 25;  
131 | |     // If car horsepower is in radius of 50 HP, scores additional score  
132 | |     if (Math.Abs((int)EvaluationModel.Horsepower - car.Horsepower) <= 50)  
133 | |         car.Score += 25;  
134 | | }
```

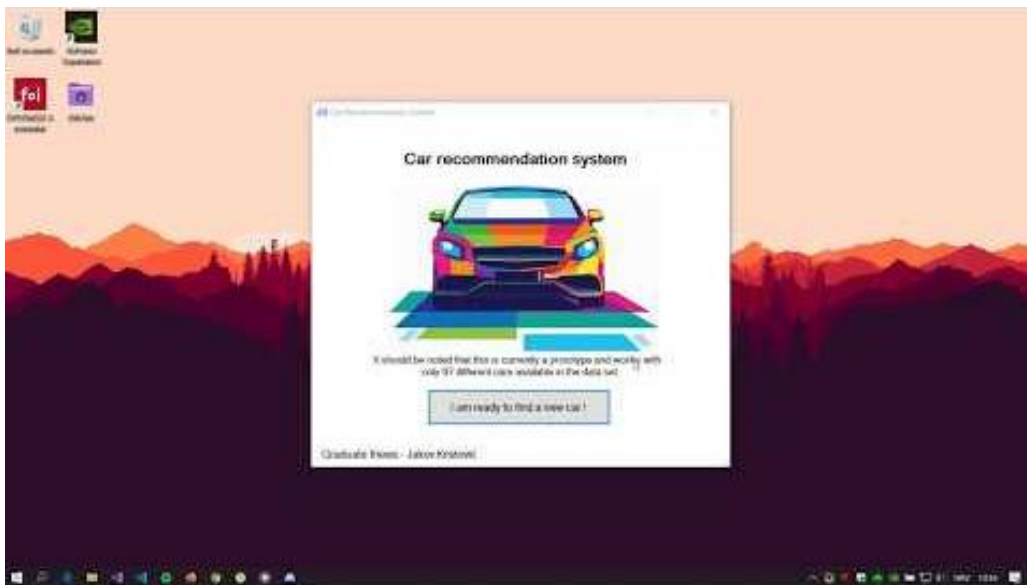
Slika 40: Evaluacija početnih ocjena kod pripadajućih atributa automobila

S druge strane, slika 40. prikazuje evaluaciju tekstualnih atributa kod automobila, gdje su prvom slučaju dodjeljuju bodovi za svaki automobil ukoliko je korisnik označio da nema preferirani način prijenosa kod automobila, odnosno dodjeljuju bodovi određenom automobilu samo ako njegov tip prijenosa odgovara onome kojeg korisnik preferira. U drugom slučaju gdje se radi evaluacija preferencija korisnika za konjske snage u automobilu, dodjeljuju se bodovi svakom automobilu koji ima više konjskih snaga nego što je korisnik htio, kao i dodatni/mogući bodovi svim onima koji imaju približan broj konjskih snaga onima koje je unio korisnik (u ovom slučaju +/- 50 konjskih snaga).

Evaluacijom svih atributa kod svih dostupnih zapisa u skupu podataka, dolazi se do završnog stanja bodova te se automobili sortiraju prema padajućim vrijednostima bodova kako bi se saznala tri optimalna automobila. Primjer rada aplikacije, odnosno sustava za preporuke te komentiranje završnog zaslona s rezultatima slijedi u sljedećem poglavlju.

5.5 Primjer rada prototipa sustava za preporuku automobila

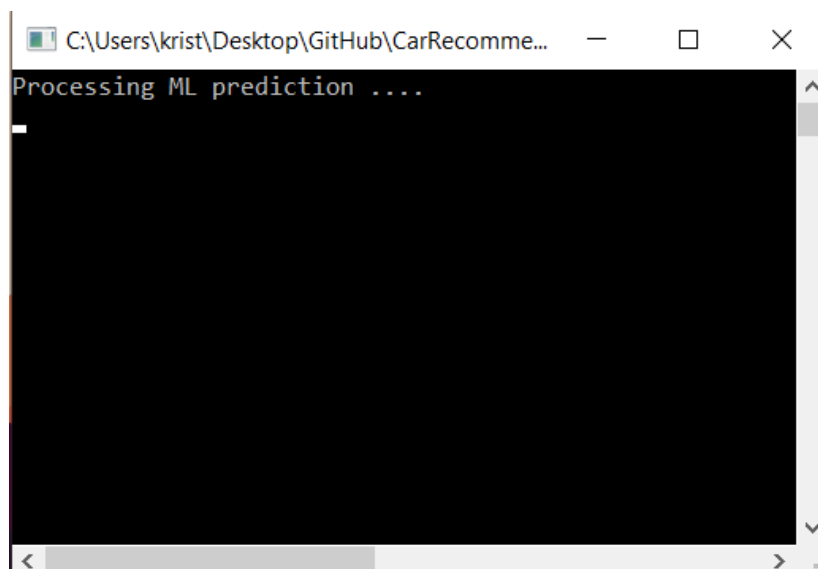
Nakon izrade prototipa sustava za preporuku automobila, vrijeme je prikazati njegov rad na djelu te se uvjeriti da ispravno radi. Kako nije baš praktično prikazati slikom svaki od zaslona s pitanjima za profiliranje korisnika pa tako ni ostalog dijela koji opisuje rad sustava, u nastavku će se priložiti i autorski video s primjerom korištenja izrađenog sustava kako bi se transparentno prikazao njegov rad.



Video 2: Primjer rada prototipa sustava za preporuku automobila

(Izvor: *YouTube*, Jakov Kristović 2020.)

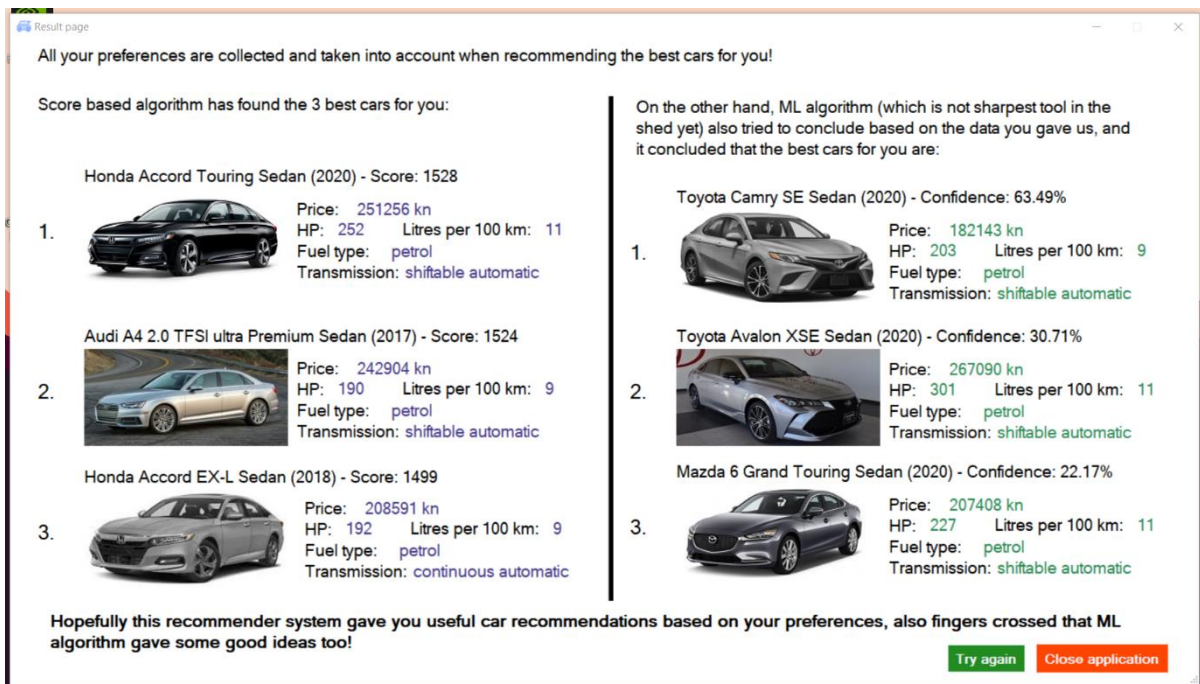
Prilikom pregleda rada sustava, nakon završetka procesa profiliranja korisnika moglo se primijetiti kako se pokraj aplikacije nakratko pojavi prozor vidljiv na slici ispod.



Slika 41: Izrada predviđanja pomoću modela strojnog učenja

Razlog pojavljivanja tog prozora je nužnost korištenja dva različita razvojnih okvira u sklopu jedne aplikacije kako bi se omogućilo predviđanje pomoću modela strojnog učenja. Preciznije rečeno, aplikacija je tipa *Windows Forms* te je napravljena unutar *.NET Framework* razvojnog okvira dok je ML *.NET* podržan samo od strane *.NET Core* razvojnog okvira. Iz tog razloga se mora pokrenuti kao odvojen proces od aplikacije, te je bilo potrebno sinkronizirati njihov rad i omogućiti komunikaciju između ta dva okvira kako bi koristili iste informacije. Naravno, ovaj prozor koji se pojavljuje se mogao i sakriti te korisniku prikazati nekakva animacija za vrijeme čekanja, međutim u ovom slučaju je namjerno izvedeno na ovaj način kako bi se naglasile prethodno spomenute informacije o odvojenim razvojnim okvirima ukoliko se netko okuša u izradi sličnog sustava pomoću ovih tehnologija.

Nakon toga je slijedio prikaz forme s rezultatima, odnosno kreiranim preporukama za prikupljene preferencije korisnika.



Slika 42: Završna forma s prikazom preporuka

S lijeve strane se mogu primijetiti preporuke koje su napravljene pomoću algoritma bodovanja, dok s desne strane su preporuke kreirane pomoću modela strojnog učenja. Ukoliko su se popratile preferencije korisnika u priloženom video zapisu, može se vidjeti kako je sustav dao preporuke za automobile koji najviše odgovaraju potrebama korisnika, pri tom uzimajući u obzir sve faktore. Najjednostavnije je za uočiti kako svi preporučeni automobili ulaze u budžet korisnika (265 000 kn, s opcijom da se plati malo više za kvalitetnijeg proizvođača), najbolje rangirani automobil s lijeve strane zadovoljava i željeni broj konjskih snaga koji je iznosio 250. Svi automobili imaju tip goriva *petrol* (benzin) te imaju srednju učinkovitost kod potrošnje goriva

kako je i naznačeno tijekom procesa profiliranja. Ostali faktori poput ocjena u pojedinim kategorijama koje uključuju performanse vožnje, udobnost, izgled interijera i ostale se također mogu provjeriti unutar skupa podataka te utvrditi visoka točnost preporuke.

Iako kreirani sustav za preporuke automobila predstavlja bolje rješenje od onih koja trenutno postoje u praksi, ovaj sustav nije savršen te toga treba biti svjestan kod njegovog korištenja. Izvorni kod je javno dostupan te se nalazi na javnom GitHub repozitoriju autora navedenom ispod ovog odlomka teksta. Na taj način se pokušava ohrabriti daljnje testiranje, proširivanje skupa podataka kao i nadogradnja sustava.

GitHub repozitorij: <https://github.com/TheJakov/CarRecommendationSystem>

6. Zaključak

U ovom diplomskom radu dan je dublji uvod u samo domenu sustava za preporuke te su se objasnile različite korisne mogućnosti koje ti sustavi nude, kako za korisnika sustava tako i za pružatelja usluge. Obradeni su različiti oblici tipičnih izvora znanja koji se mogu koristiti za personalizaciju korisničkih preporuka, kao i različiti pristupi preporučivanja kojima se razvojni inženjer takvog sustava može koristiti. U nastavku je dan i pregled tehnologija najčešćih vrsta sustava za preporuku, gdje su se obradili popularne tehnologije korištene za izgradnju sustava, od najjednostavnijih algoritama pa do inteligentnih sustava. Nakon toga napravljena je i analiza postojećih sustava te je slijedila izrada vlastitog sustava za preporuke za kupce automobila gdje su se detaljno prikazali svi osnovni koraci izrade takvog sustava kao i prokomentiran postignuti rezultat.

Sustav za preporuke je izrađen unutar .NET Framework razvojnog okvira u obliku Windows Forms aplikacije, te je sadržao dvije različite komponente za donošenje preporuka. Prva komponenta je bio autorski algoritam bodovanja koji je na temelju preferencija korisnika radio faktorizaciju i dodatno korigiranje osnovnih bodova, dok je druga komponenta izrađena pomoću ML .NET okvira za strojno učenje, razvijenog od strane Microsoft-a.

U ovom konkretnom slučaju skup podataka se samostalno izrađivao te zbog velikog broja atributa kojih je bilo potrebno prikupiti, sastojao se od samo 97 različitih zapisa o automobilima. U ovakvim uvjetima je autorski algoritam temeljen na bodovanju ostvario bolje i pouzdanije preporuke od onoga temeljenog na strojnom učenju. Razlog tome je nedostatak poželjno jako velikog broj zapisa dostupnih za treniranje modela kako bi sama kvaliteta modela bila što bolja te omogućila što preciznije preporuke.

Povezivanjem prethodno spomenutih cjelina, kreirani prototip sustava za preporuke će se staviti u kontekst s postojećim vrstama takvih sustava. S aspekta korisnih mogućnosti za pružatelja usluge koje sustav za preporuke nudi, izrađeni prototip omogućuje **bolje shvaćanje korisničkih potreba** kako svoje preporuke temelji isključivo na preferencijama korisnika. S aspekta korisnih mogućnosti za korisnika usluge, prototip omogućuje **pronalazak dobrih predmeta**, konkretno optimalnih automobila koji najbolje odgovaraju korisničkim potrebama. Izvori znanja kod izrađenog sustava su **predmeti** (automobili) i **korisnici**, kako sustav raspolaže skupom podataka koji bolje opisuju predmete, ali i prikuplja preferencije korisnika kako bi pronašao predmet koji u najvećoj mjeri zadovoljava korisničke potrebe. Pristup kod izrade preporuka kojeg izrađeni sustav koristi je **preporučivanje temeljeno na znanju**, kako sustav izrađuje preporuke utemeljene na znanju koje ima o značajkama automobila, za koje izračunava korisnost u onoj mjeri u kojoj one zadovoljavaju prikupljene preferencije korisnika.

Glede samih tehnologija koje su korištene za realizaciju sustava, sustav koristi **jednostavan programski algoritam** kreiran od strane autora kao prvu komponentu za izradu preporuka, dok se druga komponenta zasniva na **modelu strojnog učenja**. U ovom slučaju, najpouzdaniji algoritam koji se koristio za treniranje modela podataka je bio *LightGbmMulti*, koji izvršava pojačanu klasifikaciju putem algoritama utemeljenih na histogramima, dalje klasificirajući kontinuirane vrijednosti u manje skupine koje se koriste za učenje **stabla odlučivanja** [29].

Pažljivi pristup pregledu ovog rada također može dati jako dobre temelje o sustavima za preporuku te tipičnim tehnologijama koje se nalaze u pozadini takvih sustava. Također, rad daje priliku praćenja izrade jednog takvog sustava korak po korak te upoznavanja s najnovijom Microsoft-ovom tehnologijom koja uzima maha kod strojnog učenja – ML .NET.

7. Popis literature

- 1) Ricci, Francesco & Rokach, Lior & Shapira, Bracha (2010). *Recommender Systems Handbook*
- 2) Herlocker, Jonathan L., i suradnici (2000). "Explaining Collaborative Filtering Recommendations." *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work*, Association for Computing Machinery
- 3) Ricci, Francesco & Cavada, D. & Mirzadeh, N. & Venturini, A. (2006). *Case-based travel recommendations*
- 4) Montaner, Miquel & López, Beatriz & Rosa, Josep (2003). *A Taxonomy of Recommender Agents on the Internet*. Članak iz časopisa *Artificial Intelligence Review*, 19. izdanje
- 5) Fischer, Gerhard (2001). *User Modeling in Human-Computer Interaction*. Članak iz časopisa *User Modeling and User-Adapted Interaction*, 11. izdanje
- 6) Tejeda-Lorente, A. i suradnici (2018). *Adapting Recommender Systems to the New Data Privacy Regulations*. Poglavlje iz knjige *New Trends in Intelligent Software Methodologies, Tools and Techniques: Proceedings of the 17th International Conference SoMeT_18*, IOS Press.
- 7) Schafer J.B. & Frankowski D. & Herlocker J. & Sen S. (2007). *Collaborative Filtering Recommender Systems*. Poglavlje iz knjige Brusilovsky P., Kobsa A. & Nejdl W., *The Adaptive Web*, Springer, Berlin
- 8) Burke, R. (2007) *Hybrid web recommender systems*. . Poglavlje iz knjige Brusilovsky P., Kobsa A. & Nejdl W., *The Adaptive Web*, Springer, Berlin
- 9) Schafer, J.B., Konstan, J.A. & Riedl, J. (2001). *E-commerce recommendation applications*. Članak iz časopisa Goos, G. i suradnici, *Principles of Data Mining and Knowledge Discovery: 5th European Conference, Freiburg, Germany*
- 10) Bridge, D., Goker, M., McGinty, L. & Smyth, B. (2006). *Case-based recommender systems*. Članak iz časopisa *The Knowledge Engineering Review*, 20. izdanje
- 11) Arazy, O., Kumar, N. & Shapira, B. (2009). *Improving social recommender systems*. Članak iz časopisa *IT Professional*, 11. izdanje
- 12) Everitt, B. (2011). *Cluster analysis*, West Sussex, U.K: Wiley

- 13) Das, J., Mukherjee, P., Majumder, S. & Gupta, P. (2014). *Clustering-Based Recommender System Using Principles of Voting Theory*. Istraživači rad objavljen u sklopu *Proceedings of 2014 International Conference on Contemporary Computing and Informatics*.
- 14) Altman, Naomi S. (1992). *An introduction to kernel and nearest-neighbor nonparametric regression*. Članak iz časopisa *The American Statistician*, 46. izdanje.
- 15) Shams, B. & Haratizadeh, S. (2017). *Graph-based collaborative ranking*. ArXiv, abs/1604.030147.
- 16) Funk, Simon (2006). *Netflix Update: Try This at Home*. Dostupno 04.05.2020. na <https://sifter.org/~simon/journal/20061211.html>
- 17) Mild, A. & Natter, M. (2002). *Collaborative Filtering or Regression Models for Internet Recommendation Systems?*. Članak iz časopisa *Journal of Targeting*, 10. izdanje.
- 18) Gershman, A., Meisels, A., Lüke, K., Rokach, L., Schclar, A., & Sturm, A. (2010). *A Decision Tree Based Recommender System*. IICS.
- 19) Valdiviezo-Diaz, P., Ortega, F., Cobos, E. & Lara-Cabrera, R. (2019). *A Collaborative Filtering Approach Based on Naïve Bayes Classifier*. Članak iz časopisa *IEEE Access*, 7. izdanje.
- 20) Miyahara K. & Pazzani M.J. (2000). *Collaborative Filtering with the Simple Bayesian Classifier*. Poglavlje iz knjige Mizoguchi R. & Slaney J., *PRICAI 2000 Topics in Artificial Intelligence*, Springer, Berlin
- 21) De Campos, L.M., Fernandez-Luna, J.M., Huete, J.F., & Rueda-Morales, M.A. (2010). *Combining Content-Bases and Collaborative Recommendations: A Hybrid Approach Based on Bayesian Networks*. Članak iz časopisa *International Journal of Approximate Reasoning*, 51. izdanje.
- 22) Ahirwadkar, B. & Deshmukh, S.N. (2019). *Deep Neural Networks for Recommender Systems*. Članak iz časopisa *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, 8. svezak, 12. izdanje.
- 23) Chen, Lin, Kung, Chung & Yen (2019). *Design and Implementation of Cloud Analytics-Assisted Smart Power Meters Considering Advanced Artificial Intelligence as Edge Analytics in Demand-Side Management for Smart Homes*.

- 24) Ahmed Z. i suradnici, Microsoft i Yandex, (2019). *Machine Learning at Microsoft with ML.NET*. Znanstveni članak sa konferencije: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.
- 25) Croft, W. Bruce & Harper, David J. (1979). *Using probabilistic models of document retrieval without relevance information*. *Journal of Documentation*, 35 izdanje, str. 285-295.
- 26) Greiff, Warren R. (1998). *A theory of term weighting based on exploratory data analysis*. Znanstveni članak sa konferencije: *Proceedings of the 21st ACM SIGIR International Conference on Research and Development in Information Retrieval*.
- 27) Lee, S., Park, S., Kahng, M., & Lee, S. (2013). *PathRank : Ranking nodes on a heterogeneous graph for flexible hybrid recommender systems*. *Expert Systems with Applications*, 40, str. 684–697.
- 28) Yu, X., Ren, X., Sun, Y., & Gu, Q. (2014). *Personalized Entity Recommendation : A Heterogeneous Information Network Approach*. Znanstveni članak sa konferencije: *7th ACM International Conference on Web Search and Data Mining*, str. 283–292.
- 29) *Welcome to LightGBM's Documentation!* — LightGBM 2.3.2 dokumentacija, 2020. [Na internetu]. Dostupno: <https://lightgbm.readthedocs.io/en/latest/index.html>. [pristupano 24.06.2020.].
- 30) *“How to Find the Right Car for You.”* Članak sa Consumer Reports stranice, 2017. [Na internetu]. Dostupno: <https://www.consumerreports.org/buying-a-car/how-to-find-the-right-car-for-you/> [pristupano 24.06.2020.].
- 31) *“10 Common Mistakes People Make When Buying A Car.”* Članak sa Autoversed stranice, 2019. [Na internetu] Dostupno: <https://autoversed.com/10-common-mistakes-people-make-when-buying-a-car/> [pristupano 24.06.2020.].

8. Popis slika i video materijala

Popis slika treba biti izrađen po uzoru na indeksirani sadržaj, te upućivati na broj stranice na kojoj se slika može pronaći.

Slika 1. Apstraktni prikaz rada sustava za preporuku	6
Slika 2. Glavne komponente arhitekture sustava za preporuke	14
Slika 3. Princip razdvajanja znanja	14
Slika 4. Arhitektura anonimiziranog sustava za preporuke.....	15
Slika 5. Skala zadovoljstva.....	16
Slika 6. Binarne ocjene.....	17
Slika 7. Numeričke ocjene.....	17
Slika 8: Preporučivanje na temelju sadržaja	19
Slika 9: Kolaborativno filtriranje	20
Slika 10: Demografsko preporučivanje.....	21
Slika 11: Preporučivanje temeljeno na znanju	22
Slika 12: Detekcija zajednice	24
Slika 13: Preporučivanje temeljeno na zajednici	24
Slika 14: Hibridni sustav za preporučivanje	25
Slika 15: Klasteriranje podataka	28
Slika 16: Pronalazak 3 najbliža susjeda	29
Slika 17: Bipartitni graf.....	31
Slika 18: Matrična faktorizacija prema S. Funk.....	34
Slika 19: Linearna regresija.....	35
Slika 20: Primjer jednostavnog stabla odlučivanja prema preferencijama korisnika	37
Slika 21: Bayesova mreža kod hibridnog sustava za preporuku	40
Slika 22: Arhitektura umjetne neuronske mreže	42
Slika 23: Primjer uobičajenog sustava za prodaju automobila – Parkers.co.uk.....	44
Slika 24: Car choosing tool – Parkers.co.uk	46
Video 1: Pregled mobile.de sustava za prodaju automobila	47
Slika 25: Dio vlastitog napravljenog predloška za objekt automobila.....	52
Slika 26: Dio dokumenta završnog skupa podataka.....	53
Slika 27: Početni prozor aplikacije.....	54
Slika 28: Pitanje važnosti pojedine komponente.....	54
Slika 29: Izražavanje osobnih preferencija	55
Slika 30: Određivanje različitih parametara - budžet.....	55
Slika 31: Model preferencija korisnika	56
Slika 32: Model Pokretanje ML .NET Model Builder-a.....	57
Slika 33: ML .NET – odabir scenarija	58
Slika 34: ML .NET – dodavanje skupa podataka za treniranje.....	58
Slika 35: ML .NET – treniranje modela podataka	59
Slika 36: ML .NET – 3 najbolja modela	59
Slika 37: ML .NET – predikcija pomoću modela strojnog učenja.....	60
Slika 38: Dodjeljivanje vrijednosti atributu unutar klase evaluacije.....	62
Slika 39: Evaluacija početnih ocjena kod pripadajućih atributa automobila	62
Slika 40: Evaluacija početnih ocjena kod pripadajućih atributa automobila	63
Video 2: Primjer rada prototipa sustava za preporuku automobila	64
Slika 41: Izrada predviđanja pomoću modela strojnog učenja	64
Slika 42: Završna forma s prikazom preporuka	65

9. Popis tablica

Popis tablica treba biti izrađen po uzoru na indeksirani sadržaj, te upućivati na broj stranice na kojoj se tablica može pronaći.

Tablica 1: Tipični kriteriji za filtriranje automobila..... 45

Korišteni alati:

- 1) Balsamiq Studios, LLC (2018). *Balsamiq Mockups 3* [alat za izradu skica]
<https://balsamiq.com/>
- 2) ShareX Team (2020). *ShareX* [alat za izradu snimki zaslona]
<https://getsharex.com/>
- 3) Microsoft (2020). *Visual Studio 2019* [integrirano razvojno okruženje]
<https://visualstudio.microsoft.com/>
- 4) Microsoft (2020). *ML .NET* [razvojni okvir za strojno učenje]
<https://dotnet.microsoft.com/apps/machinelearning-ai/ml-dotnet>