UNIVERZITET U NIŠU

ELEKTRONSKI FAKULTET

Katedra za računarstvo

IMPLEMENTACIJA SISTEMA ZA PREPORUKU AUTOMOBILA KORISTEĆI TWO TOWER ARHITEKTURU

SEMINARSKI RAD

Predmet: Web Mining

Student:

Nemanja Stojanović
2126

Mentor:

Doc. dr Miloš Bogdanović

Sadržaj

Uvod

Sistemi za preporuke

Zašto koristiti sisteme za preporuke?

Sistemi za preporuke predstavljaju jedan od najvažnijih podskupova inteligentnih informacionih sistema i danas su temelj personalizacije u gotovo svim modernim digitalnim platformama.

Njihov primarni cilj je da korisnicima pomognu da pronađu informacije, proizvode ili sadržaje koji su relevantni za njihove potrebe, preferencije i ponašanje. Razlog za njihov značaj leži u nekoliko ključnih faktora.

Njihov cilj je jednostavan: povezati korisnike sa stvarima koje će voleti.

Cilj je jednostavan, međutim mnoge današnje platforme imaju ogromnu količinu podataka i korisnika, kako onda da što bolje prikažemo svakom korisniku mali skup podataka koji će mu se svideti i koji će ga zadržati na našoj platformi?

Preopterećenost informacijama (information overload)

Savremeni korisnik se suočava sa ogromnim obimom dostupnih podataka. Na tržištu automobila situacija je ista, hiljade oglasa, različiti modeli, generacije, stanja, cene i konfiguracije.

Bez inteligentnog sistema koji filtrira relevantne informacije:

- korisnik mora ručno da pretraži veliki broj oglasa,
- proces izbora traje dugo,
- često se propuste najbolji izbori zbog obima ponude.

Sistem za preporuke rešava ovaj problem tako što prioritet daje sadržaju koji najverovatnije odgovara interesovanjima korisnika.

Personalizacija kao ključ poboljšanog korisničkog iskustva

Korisnici očekuju personalizovan pristup, sadržaj usklađen sa njihovim navikama, preferencama i ciljevima.

U domenu automobila to znači:

- preporuke u skladu sa budžetom,

- izbor goriva koji korisnik preferira (benzin/dizel/hibrid),
- tip automobila koji mu odgovara (gradski, porodični, sportski),
- godište i kilometraža koje su u skladu sa njegovim očekivanjima.

Ušteda vremena i smanjenje kompleksnosti procesa odlučivanja

Kupovina automobila je složen proces:

- potrebno je analizirati tehničke karakteristike,
- postoje subjektivne preferencije,
- ponuda je velika i raznovrsna,
- automobili se razlikuju u nijansama koje korisniku možda nisu odmah jasne.

Preporučivači pomažu:

- skraćivanjem vremena potrebnog za informisanje,
- automatskim filtriranjem nerelevantnih opcija,
- isticanjem najkvalitetnijih relevantnih izbora.

Zašto ću koristiti two-tower model?

Two-tower arhitektura predstavlja jednu od najrasprostranjenijih savremenih metoda za izgradnju efikasnih i skalabilnih sistema za preporuke. Ova arhitektura je posebno pogodna za domene u kojima postoje različiti tipovi ulaznih podataka za korisnike i stavke, kao što je slučaj sa tržištem automobila.

Veliki broj atributa, raznovrsni profili kupaca i potreba za personalizacijom čine two-tower pristup logičnim izborom u projektovanju preporučivačkih modela.

Suština two-tower pristupa jeste razdvajanje procesa reprezentacije korisnika i stavki u dva nezavisna, paralelna enkodera: **user tower** i **item tower**.

Umesto da se informacije o korisniku i automobilu „mešaju“ u jednom modelu, kao kod klasičnih dubokih mreža, two-tower arhitektura omogućava da se oba entiteta uče odvojeno, a da se njihova sličnost izračunava tek na kraju.

Koncept dvostrukih enkodera (user tower + item tower)

Two-tower model se sastoji od dva glavna dela, nazvana „tornjevi“, pri čemu svaki od njih ima specifičnu funkciju i strukturu:

User Tower

User tower je neuronska mreža koja prima ulazne podatke o korisniku. Ti podaci mogu uključivati:

- prethodno pregledane automobile,
- korisnikove preferencije (npr. maksimalna cena, omiljena marka),
- implicitne signale kao što su klikovi, zadržavanje na stranici ili istorija pretrage.

Cilj ovog tornja je da sve te informacije transformiše u niz brojeva, **user embedding,** koji predstavlja kompaktno, matematičko mapiranje korisnikovih interesovanja.

Item Tower

Item tower radi isto to, ali za stavke: u ovom slučaju automobile. Kao ulaz se koriste atributi vozila, kao što su:

- tehničke specifikacije (snaga, zapremina, menjač, gorivo),
- starost i kilometraža,
- tip karoserije,
- nivo opreme,
- eventualni tekstualni opisi.

Item tower uči **item embedding**, koji predstavlja poziciju određenog automobila u istom vektorskom prostoru u kojem se nalazi i korisnik.

Ujedinjavanje reprezentacija

Kada se embedding korisnika i embedding automobila izračunaju, model koristi meru sličnosti (najčešće dot product) kako bi utvrdio koliko dobro određeni automobil odgovara datom korisniku.

Visoka vrednost sličnosti znači da automobil ima karakteristike koje se poklapaju sa korisnikovim interesima. Na taj način model može brzo da pretražuje veliki skup automobila i rangira one koji najviše odgovaraju profilu korisnika.

Prednosti two-tower arhitekture

Ovaj pristup uvodi nekoliko ključnih prednosti u odnosu na tradicionalne modele preporuka:

- **Skalabilnost:** Item embedding može da se unapred izračuna i kešira, što omogućava brzu pretragu čak i nad stotinama hiljada oglasa.
- **Efikasnost:** Sličnost između korisnika i stavki izračunava se vrlo brzo, što omogućava preporuke u realnom vremenu.

- **Fleksibilnost:** Ulazi za svaki toranj mogu biti različiti i nezavisno poboljšavani bez izmene celog sistema.
- **Prilagodljivost velikim i heterogenim atributima**, što je naročito važno za automobile.

Two-tower modeli tako omogućavaju inteligentno uparivanje korisnika i automobila, oslanjajući se na reprezentacije naučene iz podataka, a ne na ručno definisane filtere. Rezultat je sistem koji bolje razume korisnikove potrebe i preciznije rangira ponudu vozila.

Struktura user tower-a

User tower predstavlja deo two-tower arhitekture koji je zadužen za modelovanje korisničkog profila. Njegova osnovna funkcija je da prikupi, transformiše i kondenzuje informacije o korisniku u kompaktnu numeričku reprezentaciju: **user embedding**. Ova reprezentacija služi kao apstraktan prikaz korisnikovih preferencija i ponašanja, koji se kasnije poredi sa embeddingom automobila radi generisanja preporuka.

User tower je ključan za preciznu personalizaciju, jer on određuje kako sistem razume potrebe, želje i navike korisnika u procesu kupovine automobila.

1. Uloga korisničkih osobina (preference, parametri kupovine)

Podaci o korisniku mogu dolaziti iz više izvora, a svaki od njih nosi specifičnu informaciju o tome kakav automobil korisnik traži ili vrednuje. U domenu preporuke automobila najvažnije kategorije korisničkih osobina su:

a) Implicitne preferencije

Ovo su signali koji proizilaze iz korisnikovog ponašanja na platformi:

- automobili koje je pregledao,
- modeli na kojima se najduže zadržao,
- vozila koja je označio kao omiljena ili uporedio,
- istorija pretrage (npr. „BMW dizel“, „karavan“, „do 10.000 €“).

Implicitne preferencije imaju veliku težinu jer predstavljaju realne obrasce interesovanja, čak i kada korisnik to ne izrazi eksplicitno.

b) Eksplicitne preferencije

Korisnik direktno navodi svoje kriterijume, na primer:

- maksimalna cena,
- tip karoserije (SUV, limuzina, karavan),

- motor (dizel, benzin, hibrid),
- minimalna snaga motora,
- godište i maksimalna kilometraža.

Ove informacije mogu biti stabilne ili promenljive, ali predstavljaju jasne parametre kupovine koje sistem mora ugraditi u model korisnika.

c) Kontekstualne osobine
U određenim slučajevima korisnički profil može uključiti i dodatne podatke:

- lokacija korisnika,
- sezonske namere (npr. kupovina pre zime ili leta),
- uređaj sa kog pretražuje.

Ove informacije nisu direktno vezane za preferencije, ali poboljšavaju kvalitet personalizacije. Sve ove osobine user tower agregira u jedinstven skup karakteristika koji opisuje korisnikov stil i prioritete pri izboru automobila.

2. Kako se kreira embedding korisnika
Cilj user tower-a je da korisničke osobine, koje su različitih tipova (numeričke, kategorijske, tekstualne, implicitne), transformiše u niz brojeva fiksne dužine — **embedding vektor**.

Proces kreiranja embeddinga obično se sastoji iz nekoliko koraka:

a) Pretprocesiranje korisničkih ulaza

- Numeričke vrednosti se normalizuju (npr. cena ili starost vozila).
- Kategorijske vrednosti se pretvaraju u indeksirane reprezentacije (npr. marka, tip goriva).
- Višestruki signali (serija pogledanih automobila) se agregiraju statističkim funkcijama ili sekvencijalnim modelima.

Cilj je da svi ulazi budu pripremljeni za neuronsku mrežu.

b) Prolazak kroz neuronske slojeve
User tower je obično višeslojna perceptron mreža (MLP) koja uči kompleksne obrasce iz podataka. Tipična struktura:

- ulazni sloj u koji ulaze transformisane korisničke osobine,
- 2–4 skrivena sloja sa nelinearnim aktivacijama (ReLU, GELU),
- dropout ili batch normalization radi stabilnosti,
- izlazni sloj koji generiše embedding.

Mreža uči da mapira korisnike u vektorski prostor tako da korisnici sličnih interesovanja budu blizu jedni drugima.

c) Formiranje finalnog embeddinga
Izlaz user tower-a je embedding vektor, npr. veličine 32, 64 ili 128 elemenata. Ovaj vektor predstavlja matematički kod korisnikovih preferencija.

Embedding ima sledeće osobine:

- **sažet je**, jer smanjuje kompleksnost korisničkog profila na mali broj dimenzija,
- **semantički bogat**, jer sadrži informacije koje direktno utiču na to koje automobile sistem preporučuje,
- **kompatibilan**, jer se nalazi u istom prostoru kao embedding automobila, što omogućava neposredno poređenje.

d) Gubitak i treniranje
Tokom treniranja, sistem uči embedding tako što „gura“ korisnika bliže automobilima koje je gledao ili birao, a dalje od automobila koji nisu relevantni. Ovo se postiže tzv. kontrastivnim učenjem (contrastive learning), često uz negativno sample-ovanje. Time embedding postaje sve precizniji kako raste broj korisničkih interakcija.

User tower tako postaje centralni mehanizam koji pretvara raznolike informacije o korisniku u efikasnu reprezentaciju koja omogućava personalizovane preporuke automobila.

Struktura item tower-a

Item tower predstavlja drugi ključni deo two-tower arhitekture i zadužen je za modelovanje stavki: u ovom slučaju automobila. Za razliku od user tower-a, koji opisuje korisničke preferencije, item tower opisuje svojstva svakog automobila i pretvara ih u kompaktnu numeričku reprezentaciju, tzv. **item embedding**.

Na osnovu tog embeddinga sistem može da proceni koliko se određeni automobil „poklapa“ sa profilom korisnika.

Item tower je posebno važan u domenu preporuka automobila, jer automobili imaju veliki broj atributa koji direktno utiču na odluku kupca, pa precizno modelovanje stavki predstavlja osnovu efikasnih preporuka.

1. Atributi automobila

Automobili su kompleksne stavke sa velikim brojem kategorijskih, numeričkih i tekstualnih atributa. Najčešći atributi koji se koriste u preporučivačkim sistemima su:

a) Kategorijski atributi

Ovi atributi predstavljaju opisne, diskretne informacije o vozilu:

- marka i model (npr. BMW 3, Audi A4),
- tip goriva (benzin, dizel, hibrid),
- tip menjača (manuelni, automatski),
- tip karoserije (limuzina, karavan, SUV),
- pogon (prednji, zadnji, 4x4).

Kategorijski atributi su važni jer odražavaju osnovne preference kupaca: mnogi korisnici već na početku imaju jasno definisanu marku, segment ili tip motora koji preferiraju.

b) Numerički atributi

Ovi atributi predstavljaju kvantitativne karakteristike vozila:

- cena,
- godina proizvodnje,
- kilometraža,
- snaga motora,
- zapremina motora,
- potrošnja goriva,
- emisija CO_2.

Numerički podaci se obično normalizuju kako bi neuronske mreže mogle efikasno da rade sa njima.

c) Tekstualni atributi

Neke platforme uključuju:

- opis oglasa,
- informacije o održavanju,
- izlistane dodatne opcije (npr. „kožna sedišta“, „parking senzori“, „panorama“).

Tekst može biti modelovan jednostavnim metodama (bag-of-words) ili naprednim jezičkim embedding modelima.

d) Statični i dinamički atributi

- Statični atributi su oni koji se ne menjaju, kao marka i model.
- Dinamički atributi uključuju cenu i kilometražu koje se mogu ažurirati tokom vremena.

Item tower kombinuje sve ove informacije u jedinstvenu reprezentaciju koja opisuje vozilo na visokom nivou apstrakcije.

2. Kreiranje embeddinga automobila

Cilj item tower-a je da sve različite atribute automobila pretvori u embedding vektor — niz brojeva koji matematički predstavlja automobil u istom prostoru u kojem se nalazi embedding korisnika.

Proces kreiranja embeddinga obično uključuje sledeće korake:

a) Pretprocesiranje atributa

- Numerički atributi se skaliraju (npr. MinMax ili StandardScaler).
- Kategorijski atributi se pretvaraju u indeksirane vrednosti i prolaze kroz embedding slojeve.
- Tekstualni atributi se reprezentuju posebnim embedding metodama kada postoje.

Cilj pretprocesiranja je da svi atributi budu u formi pogodnoj za neuronsku mrežu.

b) Učenje reprezentacija iz kategorijskih atributa

Kategorijski podaci (marka, model, gorivo…) prolaze kroz **embedding slojeve**, koji svakoj kategoriji dodeljuju vektor (npr. dužine 8, 16, 32).
Ovakvi embedding vektori omogućavaju da se semantički slične vrednosti (npr. BMW i Audi) nađu bliže jedna drugoj u vektorskom prostoru.

c) Učenje reprezentacija iz numeričkih atributa

Numerički podaci obično ulaze direktno u mrežu ili se kombinuju sa kategorijskim embeddingima.
Neuronska mreža zatim uči kako da te vrednosti kombinuje u smislen prikaz automobila.

d) Agregacija svih atributa

Svi embeddingi i numerički podaci se konkateniraju (spajaju) u jedan veliki vektor koji sadrži sve informacije o automobilu.
Taj spojeni vektor se prosleđuje kroz više potpuno povezanih (MLP) slojeva koji uče:

- odnose između različitih atributa,
- hijerarhiju važnosti među karakteristikama automobila,
- apstraktne obrasce koji nisu vidljivi iz pojedinačnih atributa.

e) Formiranje finalnog embeddinga

Izlazni sloj item tower-a generiše **item embedding**, obično veličine 32, 64 ili 128 dimenzija.
Ovaj embedding predstavlja automobil na način koji omogućava upoređivanje sa embeddingom korisnika putem sličnosti (npr. dot product).

Važno je da embedding:

- dobro opisuje karakter automobila,
- ističe osobine važne za korisnike,
- bude stabilan, skalabilan i efikasan za uklapanje u realne preporuke.

f) Treniranje item tower-a
Model se trenira tako što se embedding automobila približava embeddingu korisnika koji je pokazao interesovanje za taj automobil.
Istovremeno se udaljava od embeddinga nerelevantnih automobila.
Taj proces daje embeddinge koji postaju sve precizniji kako raste količina podataka.

Item tower, zajedno sa user tower-om, omogućava kreiranje sistema koji istovremeno razume i korisnika i automobile, i efikasno ih uparuje kroz zajednički embedding prostor. Ovaj deo sistema predstavlja temelj za kvalitetne, personalizovane i brze preporuke u domenu kupovine polovnih automobila.

Funkcija sličnosti

U two-tower arhitekturi, nakon što user tower i item tower generišu svoje embedding vektore, potrebno je izračunati meru sličnosti između korisnika i automobila. Ta mera određuje koliko dobro se dati automobil uklapa u korisnikove preferencije.

Funkcija sličnosti je zato centralni deo preporučivačkog procesa: ona omogućava rangiranje stavki i izbor onih koje su najrelevantnije.

Najčešće korišćene funkcije sličnosti u recommender sistemima su **dot product** i **cosine similarity**. Obe funkcije rade na vektorskim reprezentacijama, ali naglašavaju različite aspekte odnosa između korisnika i stavke.

1. Dot product (skalarni proizvod)

Dot product je najjednostavnija i najčešće korišćena funkcija u two-tower modelima. Za dva vektora, korisnički embedding *u* i embedding automobila *v*, skalarni proizvod se računa kao:

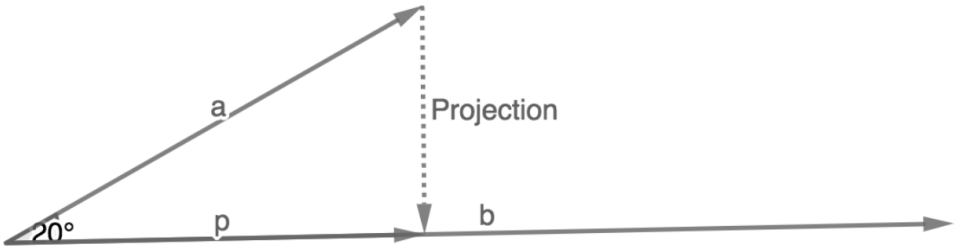
$$u \cdot v = \sum_{i=1}^{n} u_i v_i$$

Osobine:

- Uzimaju se u obzir i pravac i dužina vektora.

- Veći rezultat znači veću sličnost.
- Veoma je efikasna za izračunavanje i dobro se uklapa u proces treniranja.
- Omogućava korišćenje tehnika kao što je *approximate nearest neighbor search*, što ubrzava pretragu kroz veliki broj oglasa.

Dot product se u praksi koristi kada je model treniran tako da vektori već implicitno uče i smer i intenzitet korisničkih preferencija.



https://math.stackexchange.com/questions/805954/what-does-the-dot-product-of-two-vectors-represent

2. Cosine similarity (kosinusna sličnost)

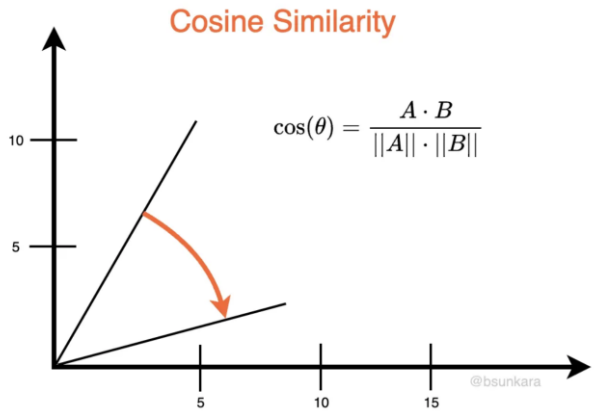
Cosine similarity meri ugaonu sličnost između dva vektora, bez obzira na njihovu dužinu. Definiše se formulom:

$$\cos(\theta) = \frac{u \cdot v}{\| u \| \| v \|}$$

Osobine:

- Normalizuje vektore, pa poredi samo njihov pravac.
- Pogodna je kada je važno da se ne favorizuju automobili čiji embedding ima veću dužinu (npr. zbog većeg broja atributa).
- Stabilnija je u sistemima gde vektori mogu značajno varirati u magnitudi.

Cosine similarity se često koristi u tekstualnim preporukama i embedding prostorima gde je smer važniji od intenziteta.



https://medium.com/@bhavanishankarsunkara/exploring-similarities-cosine-sine-and-tangent-6056fa6a2c61

3. Zašto se koriste u preporukama

Dot product i cosine similarity koriste se zato što embedding prostor omogućava da se složeni odnosi između korisnika i stavki prikažu kao geometrijski odnosi vektora. Prednosti ovih funkcija uključuju:

a) Brzo i efikasno izračunavanje
Sličnost se računa kao jednostavna algebraička operacija, što je ključno za sisteme sa desetinama hiljada automobila.

b) Kompatibilnost sa uvežbanim embedding prostorom
Tokom treniranja, model uči da:

- relevantni automobili budu blizu korisnika u embedding prostoru,
- nerelevantni automobili budu udaljeni.

Zbog toga dot product i cosine similarity prirodno izražavaju naučenu semantičku bliskost.

c) Omogućavaju rangiranje
Preporuke se generišu tako što se automobili sortiraju po vrednosti sličnosti.
Najveći rezultat je ujedno i najrelevantniji automobil.

d) Pogodni su za indeksiranje i pretragu
Posebno dot product omogućava upotrebu ANN algoritama (FAISS, ScaNN i dr.), što omogućava pretragu u realnom vremenu.

e) Jednostavni su za implementaciju i optimizaciju tokom treniranja
Ove funkcije uklapaju se u kontrastivne gubitke (npr. softmax loss, triplet loss), koji su standard u two-tower modelima.

U kontekstu preporuka automobila, funkcije sličnosti omogućavaju da sistem intuitivno i efikasno utiče na rangiranje vozila, povezujući korisnikove preferencije sa karakteristikama automobila. Rezultat je personalizovana lista automobila koja najbolje odgovara potrebama i stilu korisnika.

Prednosti two-tower pristupa

Two-tower arhitektura razvijena je kao odgovor na potrebe modernih preporučivačkih sistema, u kojima je neophodno kombinovati veliku količinu podataka o korisnicima i stavkama, uz brzu i efikasnu pretragu.

Ovaj pristup donosi niz prednosti koje ga čine standardom u industriji, posebno u domenima gde postoji veliki broj korisnika, veliki broj stavki i potreba za personalizacijom u realnom vremenu.

Tri ključne prednosti su **skalabilnost**, **brzina pretrage** i **dokazana primena u najzahtevnijim industrijskim okruženjima** poput Google-a, YouTube-a i Airbnb-ja.

1. Skalabilnost

Two-tower model je izuzetno skalabilan jer odvojeno uči embeddinge korisnika i stavki. To omogućava da se embedding vektori za automobile unapred izračunaju, sačuvaju i indeksiraju, tako da se ne moraju ponovo izračunavati pri svakom upitu korisnika.

Ova struktura donosi nekoliko važnih pogodnosti:

- **Nezavisna obrada podataka o korisnicima i automobilima:** dodavanje novih automobila ili ažuriranje podataka ne zahteva ponovno treniranje celog modela.
- **Efikasna manipulacija velikim setom stavki:** item embedding može da postoji u memoriji ili u optimizovanom indeksu, bez povećanja složenosti modela.
- **Lako proširivanje modela:** bez obzira da li platforma ima 10.000, 100.000 ili milion automobila, proces pretrage i preporuke ostaje stabilan i predvidljiv.

Skalabilnost je posebno važna za tržište automobila gde se stavke stalno menjaju, gde postoje sezonske oscilacije i gde se dodaje veliki broj novih oglasa svakog dana.

2. Brzina pretrage

Jedna od najvažnijih prednosti two-tower modela jeste mogućnost ekstremno brze pretrage relevantnih stavki.

Nakon što se user embedding izračuna, preporuka se svodi na:

1. poređenje embeddinga korisnika sa embeddingima svih automobila,
2. pronalaženje onih čija je sličnost najveća.

Zbog jednostavne i kontinuirane vektorske strukture, two-tower model omogućava:

- **preporuke u realnom vremenu**, čak i pri ogromnom broju stavki,
- **korišćenje ANN (Approximate Nearest Neighbor) algoritama** kao što su FAISS, ScaNN ili HNSW,
- **povratak liste najrelevantnijih automobila u milisekundama**.

Ovo je presudno za korisničko iskustvo, jer kupci očekuju da se preporuke menjaju i osvežavaju trenutno, odnosno prilikom filtriranja, pretrage ili pregleda konkretnih modela.

3. Industrijska primena (Google, YouTube, Airbnb)

Two-tower pristup nije teorijski koncept, već arhitektura koja se koristi u nekim od najvećih sistema na svetu. Njegova široka primena potvrđuje da model pouzdano funkcioniše u okruženjima sa milijardama korisnika i stavki.

Google Search & Google Ads

Google koristi varijante two-tower modela za uparivanje korisničkih upita sa relevantnim dokumentima i oglasima. Stavke se indeksiraju kroz item tower, dok user tower modeluje kontekst upita i profil korisnika. Rezultat su brze, precizne i personalizovane preporuke.

YouTube preporuke

YouTube primenjuje two-tower sisteme za generisanje liste kandidata (candidate generation stage). Korisnik se predstavlja embeddingom naučenim iz istorije gledanja, dok se video-sadržaj modeluje u item embedding prostor.

Ovim pristupom YouTube može da rangira milione videa za svakog korisnika u milisekundama.

Airbnb pretraga i rangiranje

Airbnb koristi two-tower modele da poveže korisničke preferencije (lokacija, budžet, stil putovanja) sa dostupnim objektima.

Two-tower arhitektura ovde omogućava efikasno rangiranje ogromnog broja oglasa i poboljšava personalizaciju pretrage.

Industrijska primena jasno pokazuje da two-tower pristup funkcioniše u veoma zahtevnim okruženjima i da predstavlja arhitekturu koja je istovremeno moćna, skalabilna i praktična.

Kombinacija ovih prednosti čini two-tower modele idealnim izborom za preporuke automobila. Tržište automobila generiše velike količine podataka, zahteva brzu pretragu i visoku preciznost preporuka, a upravo su to problemi koje two-tower arhitektura rešava na najbolji način.

Ograničenja two-tower modela

Iako two-tower arhitektura donosi brojne prednosti u skalabilnosti, brzini i industrijskoj primeni, ona ima i određena ograničenja. Razumevanje tih ograničenja važno je za pravilnu interpretaciju rezultata modela i za projektovanje celokupnog sistema za preporuke automobila. Dva ključna nedostatka su **nepovezanost korisničkih i stavkovnih reprezentacija tokom učenja** i **potreba za dodatnim modelima za re-ranking** kako bi se postigla visoka preciznost konačnih preporuka.

1. Nepovezanost reprezentacija

Two-tower modeli uče embeddinge korisnika i stavki odvojeno, u dva potpuno različita enkodera. Iako se tornjevi povezuju preko loss funkcije (najčešće dot product ili contrastive loss), njihova struktura i tok informacija i dalje su razdvojeni. Ova nepovezanost ima nekoliko posledica:

- **Ograničena interakcija atributa korisnika i automobila:** user tower vidi samo korisničke podatke, a item tower samo podatke o automobilima. Za razliku od jedinstvenih dubokih modela (npr. DLRM), two-tower pristup ne može da uči složene, višedimenzionalne interakcije između korisničkih osobina i karakteristika automobila.
- **Teško modelovanje specifičnih preferencija:** ako korisnik, na primer, vrednuje *samo dizel motore sa velikom snagom*, two-tower model može pogrešno generalizovati i davati preporuke prema pojedinačnim atributima, ali ne i prema kombinacijama.
- **Manja preciznost u finalnom rangiranju:** embedding prostor pokušava da približi korisnike relevantnim stavkama, ali bez eksplicitnog modelovanja složenih odnosa, kvalitativne nijanse preporuka mogu biti izgubljene.

Ovo ograničenje je posebno važno u domenu automobila, gde kupci često kombinuju više preciznih kriterijuma (godište + snaga + tip menjača + pogon), pa model mora razumeti interakcije između atributa, a ne samo njihove pojedinačne vrednosti.

2. Potreba za dodatnim modelima za re-ranking

Two-tower pristup je idealan za generisanje **kandidata** — prvog skupa vozila koji su potencijalno relevantni korisniku. Međutim, sam embedding prostor i funkcija sličnosti nisu dovoljni za donošenje završnog rangiranja.

Zbog toga se u industriji two-tower modeli gotovo uvek kombinuju sa **re-ranking modelima**, koji imaju sledeće karakteristike:

- **Rade na manjem broju stavki (npr. 50–200 kandidata).**
- **Modeluju detaljne interakcije između korisnika i automobila**, uključujući kombinacije atributa, istoriju ponašanja, kontekst i dodatne metapodatke.
- **Koriste kompleksnije modele**, kao što su:
    - duboke neuronske mreže (MLP sa interakcijskim slojevima),
    - DLRM (Deep Learning Recommendation Model),
    - attention mehanizmi,
    - GBDT modeli (XGBoost, LightGBM),
    - sekvencijalni modeli (Transformer, RNN).

Re-ranking faza je neophodna zato što:

- Two-tower embedding ne može sam da izrazi sve nijanse relevantnosti.
- Sigurnije preporuke zahtevaju detaljno razumevanje kombinacija atributa (cena + stanje + starost + oprema).
- Sistemi visokog kvaliteta moraju uzeti u obzir i poslovne metrike (engagement, CTR, konverzije, kvalitet prodavca).

Drugim rečima, two-tower arhitektura služi kao **brz i efikasan filter**, ali ne i kao završni donosilac odluke.

Ova ograničenja ne umanjuju upotrebljivost two-tower modela, već ukazuju na to da je on samo jedan od slojeva u celokupnom sistemu preporuka. U složenim domenima poput kupovine automobila, najbolje rezultate daje kombinacija two-tower arhitekture za generisanje kandidata i naprednih re-ranking modela za finalni odabir najkvalitetnijih preporuka.

Pregled korišćenog dataset-a

Dataset **„Vehicle Dataset from Cardekho“** predstavlja javno dostupnu kolekciju podataka o polovnim automobilima preuzetim sa indijskog tržišta. U okviru projekta za preporuke automobila, ovaj skup podataka služi kao osnova za treniranje two-tower modela, jer obuhvata relevantne atribute automobila koji utiču na korisničke preferencije i procenu vrednosti vozila. Dataset je dovoljno jednostavan za rad, ali istovremeno dovoljno bogat da omogući modelu učenje korisnih reprezentacija.

1. Opis atributa

Dataset sadrži više atributa koji se odnose na osnovne karakteristike polovnih automobila. Najvažniji među njima su:

- **year:** Godina proizvodnje vozila.
  Jedan od ključnih faktora pri kupovini automobila, jer utiče na cenu, stanje vozila i tehnološki nivo.
- **km_driven:** Ukupna kilometraža koju je vozilo prešlo. Velika kilometraža često implicira veću potrošenost vozila, dok manja povećava vrednost i privlačnost kupcima.
- **fuel:** Tip goriva (benzin, dizel, CNG, LPG, električno). Ovaj atribut snažno utiče na preferencije kupaca, troškove održavanja i performanse automobila.
- **seller_type:** Vrsta prodavca (individualni prodavac, diler, trust/kompanija). Može ukazivati na očekivano stanje automobila i pouzdanost oglasa.
- **transmission:** Tip menjača (manualni ili automatski). Jedan od najvažnijih preferencijskih atributa, jer kupci često unapred znaju šta žele.
- **owner:** Broj prethodnih vlasnika. Automobili sa više vlasnika često imaju manju cenu i manju privlačnost.
- **selling_price:** Cena po kojoj se vozilo prodaje. Ovo je ciljna vrednost za neke regresione zadatke, ali u sistemima preporuke služi kao važan numerički atribut koji utiče na relevantnost.
- **name:** Naziv automobila (marka i model), npr. "Maruti Swift Dzire". Ovaj atribut se koristi za identifikaciju modela i za kreiranje kategorijskih embeddinga.

Ovi atributi čine jezgro item tower-a, jer jasno opisuju automobil i omogućavaju modelu da nauči koje karakteristike su važne korisnicima.

2. Broj zapisa
Dataset iz Cardekho platforme sadrži **~8.000 zapisa**. Svaki zapis predstavlja jedan oglas za polovni automobil.
Ova veličina dataset-a je pogodna za istraživački projekat:

- dovoljno je velika da model nauči korisne obrasce,
- dovoljno mala da omogućava brzo treniranje two-tower modela,
- nema ekstremnu heterogenost koja bi otežala prvobitnu implementaciju.

Priprema dataseta

Nakon definisanja i izbora relevantnih atributa, sledeći korak predstavljao je pripremu dataseta za dalju primenu u modelu za preporuku. Ova faza obuhvata transformaciju podataka u numerički oblik pogodan za neuronske mreže, normalizaciju vrednosti, kodiranje kategoričkih promenljivih, kao i proširivanje skupa podataka dodatnim izvedenim karakteristikama.

Transformacija numeričkih vrednosti

Određene promenljive u datasetu bile su inicijalno predstavljene kao tekstualne vrednosti koje, pored numeričkog dela, sadrže i jedinice mere ili dodatne opise. Kako bi se omogućila njihova upotreba u numeričkim modelima, izvršeno je izdvajanje isključivo numeričkih vrednosti i njihova konverzija u realne brojeve. Ovim postupkom obezbeđena je konzistentnost numeričkih atributa i sprečene potencijalne greške tokom treniranja modela.

Nakon transformacije, svi zapisi koji sadrže nedostajuće ili nevalidne vrednosti uklonjeni su iz dataseta, čime je obezbeđen čist i konzistentan ulaz za dalju obradu.

Normalizacija numeričkih atributa

Numerički atributi normalizovani su primenom Min–Max skaliranja, pri čemu su sve vrednosti preslikane u interval *[0,1][0, 1]*[0,1]. Ovaj korak je od ključnog značaja za stabilno treniranje neuronske mreže, jer sprečava dominaciju atributa sa većim apsolutnim vrednostima i omogućava ravnomerniji doprinos svih numeričkih karakteristika prilikom učenja latentnih reprezentacija.

Normalizacija je posebno važna u Two-Tower arhitekturi, gde se korisnički i item vektori projektuju u zajednički latentni prostor i porede pomoću cosine similarity metrike.

Kodiranje kategoričkih promenljivih

Kategoričke promenljive transformisane su u diskretne celobrojne kodove. Ovi kodovi se koriste kao ulaz u embedding slojeve neuronske mreže, što omogućava modelu da nauči gustu, niskodimenzionalnu reprezentaciju svake kategorije. Na taj način izbegnuto je korišćenje one-hot enkodiranja, koje bi značajno povećalo dimenzionalnost ulaznog prostora.

Pored toga, sačuvane su originalne liste kategorija za svaku promenljivu, čime je omogućeno dosledno mapiranje novih korisničkih preferencija tokom faze inferencije i generisanja preporuka.

Proširivanje dataseta izvedenim karakteristikama

Kako bi se modelu obezbedile dodatne informacije koje nisu direktno prisutne u osnovnim atributima, uvedene su izvedene binarne karakteristike. Ove karakteristike dobijene su analizom naziva modela automobila i služe za implicitno kodiranje informacija o tipu vozila, kao što su pripadnost SUV/off-road kategoriji ili sportski karakter vozila.

Ovakav pristup omogućava modelu da prepozna obrasce vezane za stil i namenu vozila, čak i kada te informacije nisu eksplicitno navedene u strukturiranim atributima.

Generisanje sintetičkih korisničkih profila

S obzirom na to da dostupni dataset ne sadrži informacije o realnim korisnicima, uveden je postupak generisanja sintetičkih korisničkih profila. Definisano je više tipičnih korisničkih segmenata, pri čemu je za svaki segment generisan veći broj korisnika sa različitim, ali realističnim preferencijama.

Numeričke preferencije korisnika generisane su kao slučajne vrednosti unutar definisanih intervala, dok su kategoričke preferencije usklađene sa karakteristikama odgovarajućeg segmenta. Ovi sintetički profili koriste se za formiranje trening parova korisnik–automobil, čime je omogućeno treniranje modela u odsustvu stvarnih interakcija korisnika i sistema.

Rezultat pripreme podataka

Nakon sprovedenih transformacija, dobijen je finalni skup podataka u potpunosti prilagođen za rad sa neuronskim mrežama. Podaci su numerički stabilni, kategoričke promenljive enkodirane su na način kompatibilan sa embedding slojevima, a dodatne izvedene karakteristike omogućavaju bogatiju reprezentaciju automobila i korisničkih preferencija. Ovako pripremljen dataset predstavlja osnovu za dalju implementaciju i treniranje Two-Tower modela za preporuku automobila.

Implementacija i treniranje

Ovo poglavlje opisuje implementaciju Two-Tower sistema za preporuku, način formiranja trening skupa, arhitekturu neuronske mreže, proces treniranja i postupak generisanja preporuka.

Segmentno bodovanje kao nadzorni signal

Pošto ne postoje realne interakcije korisnika i automobila (klikovi, pregledi, kupovine), za treniranje je bilo potrebno obezbediti nadzorni signal. To je urađeno pomoću funkcije segmentnog bodovanja ***score_items_for_segment***, koja za svakog korisnika dodeljuje skor svakom automobilu u datasetu. Na osnovu tih skorova kasnije se formiraju pozitivni i negativni parovi.

Funkcija koristi dve komponente:

1. **Osnovna sličnost (base score)**
   Za numeričke atribute (koji su prethodno normalizovani u interval [0,1]) računa se apsolutna razlika između korisničkog vektora i vektora automobila:

$$d_i = |x_i - u_i|$$

Sličnost po atributu dobija se kao:

$$s_i = 1 - d_i$$

Vrednosti se potom ograničavaju na [0,1], a ukupni osnovni skor je zbir sličnosti po svim numeričkim atributima:

$$base\_score = \sum_{i=1}^{n} s_i$$

Ova komponenta osigurava da automobil bude ocenjen bolje ukoliko je "bliži" korisničkim preferencijama u prostoru normalizovanih numeričkih karakteristika.

2. **Segmentno specifična pravila (heuristike)**
   Na base_score se dodaju ili oduzimaju ponderisani termini koji reflektuju tipične preferencije segmenata:
   a. **Segment 1 (Budget Buyer):** veći značaj ceni, ekonomičnosti, kilometraži i godini, uz pogodovanje određenim kategorijama goriva/prodavca/vlasništva.
   b. **Segment 2 (Diesel Commuter):** favorizuje dizel, dobru ekonomičnost (mileage), manuelni menjač i "manje SUV" (penal za SUV).
   c. **Segment 3 (Family Buyer):** favorizuje veći broj sedišta (prag), automatski menjač, pouzdaniji kanal prodaje (dealer) i blagi plus za SUV.
   d. **Segment 4 (Sport Enthusiast):** favorizuje snagu, zapreminu, obrtni moment, sportske indikatore, penalizuje SUV; dodatno koristi binarne indikatore izvedene iz naziva (npr. coupe/sport/suv/sedan) kao jače signale.
   e. **Segment 5 (Off-road Utility):** snažno favorizuje SUV/off-road indikator i obrtni moment, kao i dizel i manuelni menjač.
   f. **Segment 6 (Luxury/Premium):** favorizuje novije, skuplje, automatske, "dealer" automobile, premium brendove i sedan profil; penalizuje ne-premium i SUV kada je cilj "urban luxury".

Ovaj hibridni skor predstavlja "pseudo-ground-truth" koji omogućava da model uči iz implicitno definisanih preferencija.

Generisanje trening parova (pozitivni i negativni primeri)

Funkcija generate_training_pairs_fast prolazi kroz sve sintetičke korisnike i za svakog korisnika:

1. Izračuna skorove za sve automobile:
$$scores = f_{segment}(u, cars)$$
2. Sortira automobile po skoru i bira:
    a. **pozitivne primere:** top npos automobila (najveći skorovi)
    b. **negativne primere:** bottom nneg automobila (najmanji skorovi)
3. Formira parove oblika:
$$(u,\ i,\ y)$$

gde je $y = 1$ za pozitivne i $y = 0$ za negativne primere.

Na kraju se svi parovi spakuju u tenzore za trening:

- korisnički numerički ulaz **u_num**
- korisničke kategorije **u_fuel**, **u_seller**, **u_trans**, **u_owner**
- item numerički ulaz **i_num**
- item kategorije **i_fuel**, **i_seller**, **i_trans**, **i_owner**
- oznake **y**

Ovakav trening skup ima smisao u scenariju bez realnih interakcija, jer model dobija stabilan signal: "ovo je relevantno za ovaj tip korisnika, a ovo nije".

Arhitektura Two-Tower modela

Model je implementiran kao dva odvojena podsistema (tornja), koja proizvode embedding vektore iste dimenzije *d* (u implementaciji **embedding_dim = 32**).

Korisnički toranj (User tower)

Ulazi:

- Numerički vektor dužine *p* (broj numeričkih atributa)
- 4 kategorička ulaza (fuel, seller_type, transmission, owner) kao celobrojni kodovi

Za svaku kategoričku promenljivu koristi se embedding sloj dimenzije 8. Embedding vektori se spljošte i konkateniraju sa numeričkim ulazom:

$h_u = [x_u; emb(fuel); emb(seller); emb(trans); emb(owner)]$

Zatim prolazi kroz Dense slojeve:

- Dense(128, ReLU)
- Dropout(0.2)
- Dense(64, ReLU)
- Dense(d) bez aktivacije (latentni vektor)

Izlaz korisničkog tornja je:

$$u \in \mathbb{R}^d$$

Item toranj (Item tower)

Ista logika i struktura kao user toranj, samo sa item ulazima. Izlaz je:

$$v \in \mathbb{R}^d$$

Skor kompatibilnosti (matching)

Na izlazu kompletne mreže računa se skalarni proizvod:

$$s(u, i) = u \cdot v$$

Ovaj skor model treba da poveća za pozitivne parove i smanji za negativne.

Model: "functional_11"

Layer (type)	Output Shape	Param #	Connected to
user_fuel (InputLayer)	(None)	0	–
user_seller (InputLayer)	(None)	0	–
user_trans (InputLayer)	(None)	0	–
user_owner (InputLayer)	(None)	0	–
item_fuel (InputLayer)	(None)	0	–
item_seller (InputLayer)	(None)	0	–
item_trans (InputLayer)	(None)	0	–
item_owner (InputLayer)	(None)	0	–
embedding_24 (Embedding)	(None, 8)	32	user_fuel[0][0]
embedding_25 (Embedding)	(None, 8)	24	user_seller[0][0]
embedding_26 (Embedding)	(None, 8)	16	user_trans[0][0]
embedding_27 (Embedding)	(None, 8)	40	user_owner[0][0]
embedding_28 (Embedding)	(None, 8)	32	item_fuel[0][0]
embedding_29 (Embedding)	(None, 8)	24	item_seller[0][0]
embedding_30 (Embedding)	(None, 8)	16	item_trans[0][0]
embedding_31 (Embedding)	(None, 8)	40	item_owner[0][0]
user_num (InputLayer)	(None, 8)	0	–
flatten_24 (Flatten)	(None, 8)	0	embedding_24[0][…
flatten_25 (Flatten)	(None, 8)	0	embedding_25[0][…
flatten_26 (Flatten)	(None, 8)	0	embedding_26[0][…
flatten_27 (Flatten)	(None, 8)	0	embedding_27[0][…
item_num (InputLayer)	(None, 8)	0	–
flatten_28 (Flatten)	(None, 8)	0	embedding_28[0][…
flatten_29 (Flatten)	(None, 8)	0	embedding_29[0][…
flatten_30 (Flatten)	(None, 8)	0	embedding_30[0][…
flatten_31 (Flatten)	(None, 8)	0	embedding_31[0][…

concatenate_6 (Concatenate)	(None, 40)	0	user_num[0][0], flatten_24[0][0], flatten_25[0][0], flatten_26[0][0], flatten_27[0][0]
concatenate_7 (Concatenate)	(None, 40)	0	item_num[0][0], flatten_28[0][0], flatten_29[0][0], flatten_30[0][0], flatten_31[0][0]
dense_18 (Dense)	(None, 128)	5,248	concatenate_6[0]…
dense_21 (Dense)	(None, 128)	5,248	concatenate_7[0]…
dropout_6 (Dropout)	(None, 128)	0	dense_18[0][0]
dropout_7 (Dropout)	(None, 128)	0	dense_21[0][0]
dense_19 (Dense)	(None, 64)	8,256	dropout_6[0][0]
dense_22 (Dense)	(None, 64)	8,256	dropout_7[0][0]
dense_20 (Dense)	(None, 32)	2,080	dense_19[0][0]
dense_23 (Dense)	(None, 32)	2,080	dense_22[0][0]
dot_3 (Dot)	(None, 1)	0	dense_20[0][0], dense_23[0][0]

Total params: 31,392 (122.62 KB)

Trainable params: 31,392 (122.62 KB)

Non-trainable params: 0 (0.00 B)

Izlaz model.summary() – prikaz slojeva, dimenzija i broja parametara Two-Tower arhitekture.

Treniranje modela

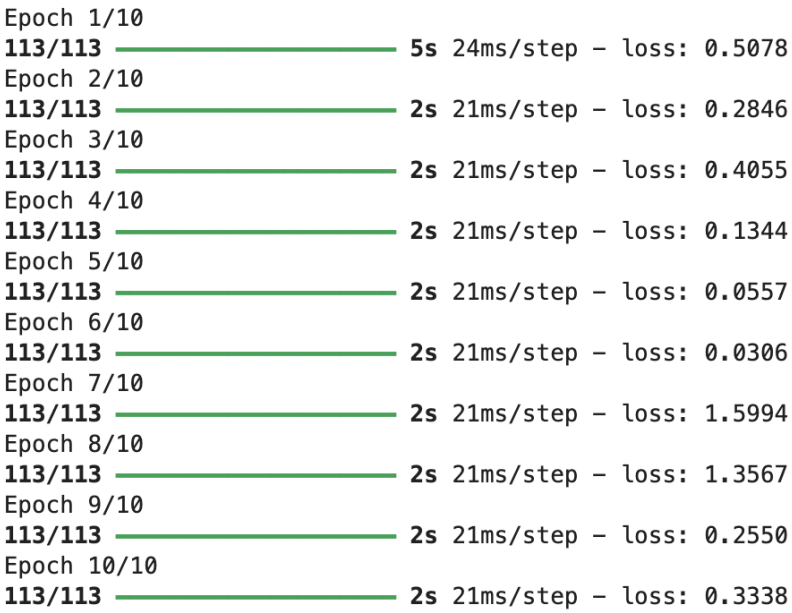
Model je treniran nad generisanim parovima koristeći:

- optimizator: Adam
- loss: binary cross-entropy
- epohe: 10
- batch size: 64

Cilj treniranja je da se nauče embedding reprezentacije tako da:

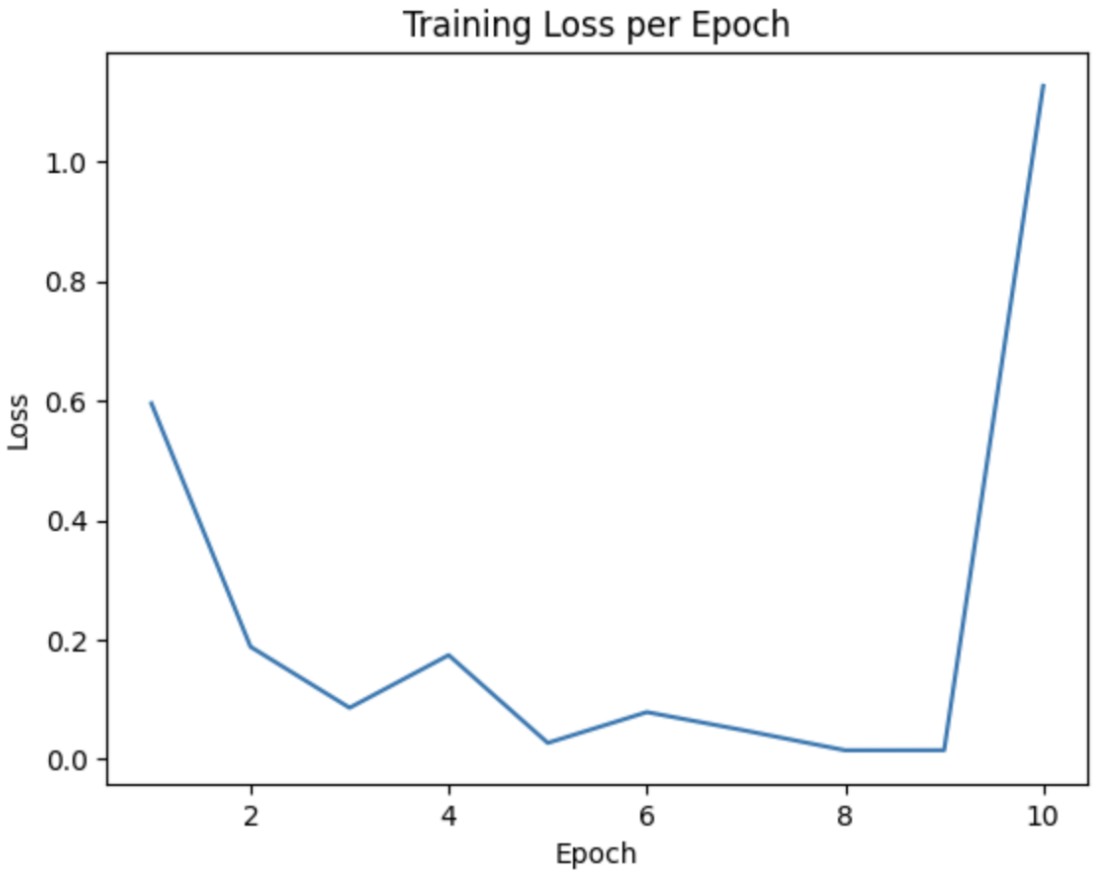
- pozitivni parovi imaju veći skor *s(u,i)*
- negativni parovi imaju manji skor

Tokom treniranja, u Jupyter-u se dobija log za svaku epohu (1–10) koji prikazuje vrednost loss-a (i eventualno druge metrike, ako se dodaju).



Log treninga po epohama (epoha 1–10) preuzet iz Jupyter okruženja

Pored tekstualnog loga epoha, prikazan je i grafički prikaz promene vrednosti funkcije gubitka tokom treniranja (*Training Loss per Epoch*). Ovakav prikaz omogućava bržu i intuitivniju procenu dinamike učenja i stabilnosti optimizacije kroz epohe.

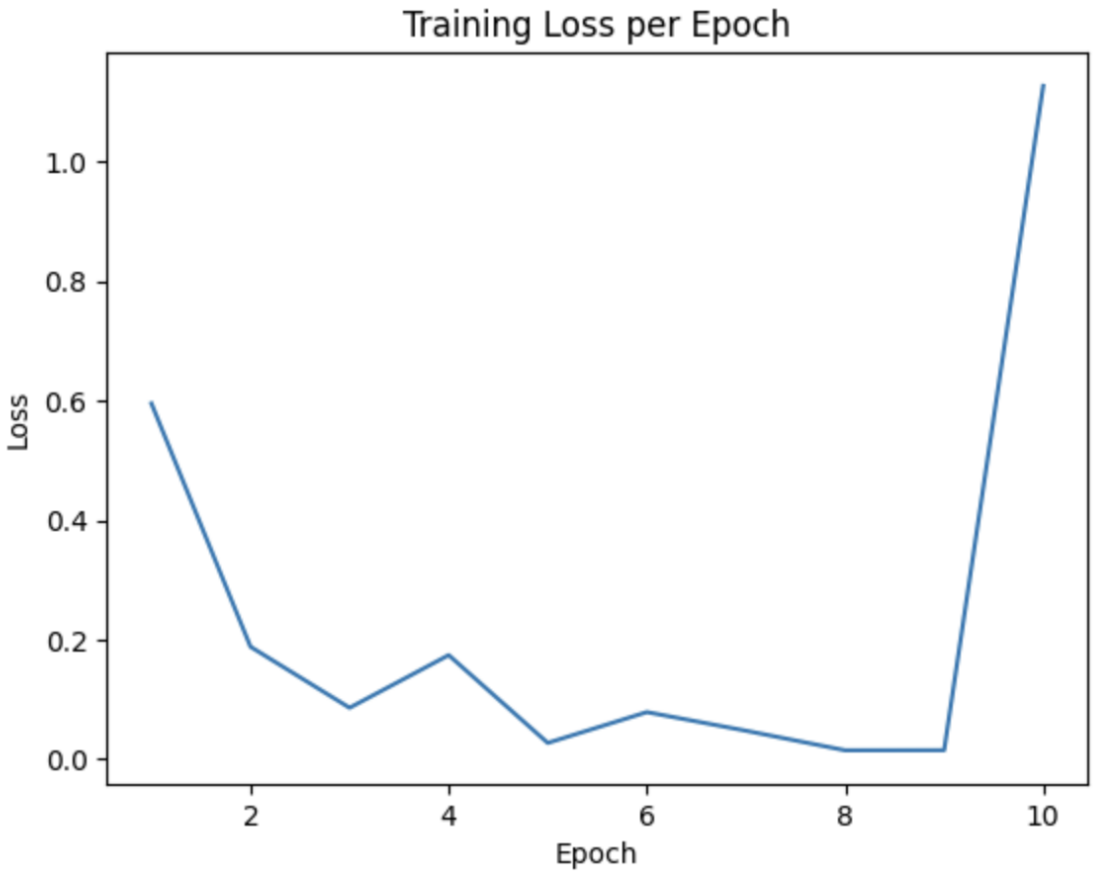


Training Loss per Epoch (inicijalna verzija modela)

Na osnovu prikazane krive može se uočiti da funkcija gubitka u prvim epohama brzo opada, što ukazuje da model efikasno uči osnovno razdvajanje pozitivnih i negativnih parova. Međutim, u kasnijoj fazi treniranja javlja se nagla promena (skok) vrednosti gubitka, što ukazuje na nestabilnost treninga i/ili pretreniranost u završnim epohama. Ovakvo ponašanje je tipično u scenarijima kada se koristi dot-product skor bez dodatne normalizacije latentnih vektora ili kada je brzina učenja previsoka za dati skup podataka.

U narednoj iteraciji modela planirano je unapređenje trening procedure (npr. stabilizacija optimizacije i/ili dodatna regularizacija), nakon čega će biti prikazana i analizirana nova kriva gubitka koja predstavlja unapređenu verziju modela.

TODO: dodati unapredjenu verziju i dati kratku analizu



Generisanje item embeddinga i priprema za inferenciju

Nakon treninga, item toranj se koristi da izračuna embedding vektore za sve automobile u datasetu:

$$\mathbf{V} = [\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_N}]$$

Ovi embedding vektori se računaju jednom i čuvaju (u memoriji ili na disku), čime se ubrzava faza preporučivanja: umesto prolaska kroz celu mrežu za svaki automobil, koristi se već izračunata matrica embeddinga.

Generisanje preporuka za novog korisnika

Za ulazni profil korisnika user_pref:

1. Formira se korisnički ulaz (numeričke + kategoričke vrednosti).

2. Korisnički toranj generiše embedding:

$$\mathbf{u} = g(\text{user\_pref})$$

3. Izračuna se sličnost sa svim automobilima primenom cosine similarity:

$$\cos(\theta i) = \frac{u \cdot v_i}{||u\ ||\,||v_i||}$$

(u implementaciji preko cosine_similarity(u_emb, item_embeddings)).

4. Automobili se sortiraju po skoru i bira se Top-N.

Uklanjanje duplikata po nazivu

Pošto dataset može sadržati više zapisa za isti model (isti name, različita godina/cena/km), uvedena je deduplikacija: pri iteraciji kroz rang listu zadržava se samo prvi (najbolje rangirani) automobil za svako jedinstveno ime. Time se korisniku ne prikazuje više varijanti istog modela u top preporukama, što poboljšava raznovrsnost rezultata.

Diskusija rezultata

Implementirani sistem za preporuku automobila pokazuje da je moguće izgraditi funkcionalan i smislen recommender sistem čak i u odsustvu realnih korisničkih interakcija, korišćenjem kombinacije heurističkog bodovanja i neuronskog modela zasnovanog na Two-Tower arhitekturi.

Dobijeni rezultati ukazuju da model uspešno uči latentne reprezentacije korisnika i automobila koje reflektuju definisane preferencije korisničkih segmenata.

Analiza preporuka generisanih za različite tipove korisnika (npr. sportski entuzijasta, porodični korisnik) pokazuje da sistem u velikoj meri favorizuje automobile čije karakteristike odgovaraju očekivanjima datog segmenta. Na primer, kod sportskog korisnika preporuke su dominantno orijentisane ka modelima sa većom snagom, manjim brojem sedišta i sportskim oznakama, dok su kod porodičnog korisnika favorizovani automobili sa većim brojem sedišta, automatskim menjačem i boljim odnosom pouzdanosti.

Grafički prikaz funkcije gubitka (*Training Loss per Epoch*) ukazuje na brzo konvergiranje modela u ranim epohama treniranja. Takvo ponašanje je očekivano, s obzirom na to da su trening parovi generisani na osnovu konzistentnih heurističkih pravila, što modelu obezbeđuje jasan nadzorni signal. U kasnijoj fazi treniranja primećena je nestabilnost vrednosti funkcije gubitka,

što dodatno potvrđuje da model relativno brzo iscrpljuje dostupnu informaciju iz trening podataka.

Ovakvi rezultati ukazuju da implementirani sistem uspešno demonstrira osnovne principe modernih sistema za preporuku, ali istovremeno otvara prostor za dodatna unapređenja i stabilizaciju treninga.

Ograničenja implementacije

Iako sistem daje intuitivno smislene preporuke, važno je naglasiti nekoliko ključnih ograničenja.

Prvo, trening podaci ne potiču iz stvarnih korisničkih interakcija, već su generisani korišćenjem heurističkog segmentnog bodovanja. Iako ovaj pristup omogućava inicijalno treniranje modela, on ne može u potpunosti da zameni realne podatke o ponašanju korisnika, kao što su klikovi, pregledi ili kupovine. Zbog toga se dobijeni model ne može smatrati u potpunosti personalizovanim u realnom smislu, već pre predstavlja simulaciju ponašanja tipičnih korisničkih profila.

Drugo, korišćenje skalarnog proizvoda embedding vektora tokom treninga, bez eksplicitne normalizacije, može dovesti do nestabilnosti optimizacije, što se manifestuje naglim promenama funkcije gubitka u kasnijim epohama. Ovaj problem je uočen u eksperimentalnim rezultatima i predstavlja poznato ograničenje dot-product pristupa.

Treće, evaluacija kvaliteta preporuka ograničena je na kvalitativnu analizu (posmatranje smislenosti preporuka), jer ne postoje realne oznake relevantnosti koje bi omogućile izračunavanje standardnih metrika kao što su Precision@K ili Recall@K.

Na kraju, deduplikacija po nazivu automobila poboljšava raznovrsnost preporuka, ali istovremeno uklanja mogućnost da se korisniku ponudi više varijanti istog modela koje bi mogle biti relevantne u zavisnosti od cene ili kilometraže.

Moguća unapređenja sistema

Postoji više pravaca u kojima bi se implementirani sistem mogao unaprediti.

Jedno od osnovnih unapređenja jeste stabilizacija treninga modela, na primer primenom normalizacije embedding vektora ili korišćenjem cosine similarity metrike direktno tokom faze

treniranja. Takođe, podešavanje brzine učenja i broja epoha moglo bi doprineti stabilnijem konvergiranju modela.

Dalje unapređenje podrazumevalo bi uvođenje realnih korisničkih interakcija. Čak i jednostavni implicitni signali, poput pregleda oglasa ili zadržavanja na stranici, omogućili bi značajno realističniji trening i evaluaciju modela. U tom slučaju, heurističko bodovanje moglo bi se koristiti samo kao inicijalni korak (*cold start* faza).

Takođe, sistem bi se mogao proširiti dodatnim atributima automobila, kao što su lokacija, potrošnja u realnim uslovima ili ocene pouzdanosti, što bi dodatno obogatilo embedding reprezentacije. Moguće je i uvođenje hibridnog pristupa koji kombinuje content-based i collaborative filtering metode.

Na kraju, evaluacija sistema mogla bi se proširiti formalnim eksperimentima, upoređivanjem više varijanti modela (inicijalni i unapređeni model), kao i analizom uticaja pojedinih komponenti arhitekture na kvalitet preporuka.

Zaključak

U ovom radu predstavljen je sistem za preporuku polovnih automobila zasnovan na Two-Tower neuronskoj arhitekturi. Prikazan je kompletan proces, od pripreme podataka i generisanja sintetičkih korisničkih profila, preko implementacije heurističkog bodovanja i generisanja trening parova, do treniranja modela i generisanja personalizovanih preporuka.

Iako sistem koristi simulirane korisničke podatke, rezultati pokazuju da je moguće izgraditi funkcionalan i interpretabilan recommender sistem koji generiše smislene preporuke u skladu sa definisanim korisničkim preferencijama. Uvedena deduplikacija po nazivu automobila dodatno doprinosi kvalitetu preporuka povećanjem njihove raznovrsnosti.

Rad pruža solidnu osnovu za dalje istraživanje i razvoj sistema za preporuku, posebno u kontekstu realnih aplikacija gde su dostupni podaci o ponašanju korisnika. Predložena unapređenja ukazuju na potencijal daljeg razvoja sistema ka robusnijem i preciznijem rešenju za personalizovane preporuke automobila.