Implementacija sistema preporuka zasnovanog na sadržaju

Putanja gdje je implementiran reccomender sistem u aplikaciji ProdajaNekretnina:

* github:

https://github.com/Lejlica/prodajaNekretnina/blob/main/ProdajaNekretnina.Services/N

ekretnineService.cs

* Putanja u aplikaciji:

ProdajaNekretnina.Services/NekretnineService.cs

* Putanja u Flutter projektu:

prodajanekretnina\_mobile\_novi\lib\screens\WishListaScreen.dart

Screenshot of a phone screen

AI-generated content may be incorrect.

Slika 1. Screenshot iz aplikacije gdje je implementiran recommender sistem

Vrsta sistema preporuke koju sam odabrala implementirati u seminarskom radu za predmet Razvoj softvera II je Content-Based sistem preporuke.

Funkcija *RecommendNekretnina(int userId)* implementira sistem preporuke nekretnina koji je zasnovan na listi nekretnina koje je korisnik dodao u svoju listu želja.

Za obradu podataka korišten je ML.NET a za računanje sličnosti je korišten cosine similarity.

Cosine similarity je mjera sličnosti između dva vektora u n-dimenzionalnom prostoru. U ovom slučaju, koristi se za upoređivanje vektora osobina nekretnina (broj soba, cijena, lokacija itd.) kako bi se utvrdilo koliko su dvije nekretnine međusobno slične.

U sistemu preporuke, nakon što se sve nekretnine transformišu u numeričke vektore (kroz one-hot encoding i normalizaciju), koristi se CosineSimilarity funkcija da izračuna koliko je neka nekretnina slična onoj koju je korisnik prethodno pregledao ili dodao u listu zelja.

Prvo da objasnim pojam one-hot encodinga.

Naime, one-hot encoding se koristi u slučaju ove aplikacije za lokaciju koja je kategorijska vrijednost (npr. Sarjevo, Mostar). Mašine odnosno modeli ne mogu direktno raditi s tekstom ili kategorijama, ono što njima treba su numeričke vrijednosti. Ako bismo direktno koristili gradove npr. Sarajevo=1, Mostar=2, model bi pogrešno pretpostavio da Sarajevo ima veću vrijednost od Mostara, ovdje koristimo one-hot encoding kako bi pretvorili kategorijske vrijednosti u **vektor binarnih (0/1) vrijednosti**. Za svaki jedinstveni grad se kreira posebna kolona u vektoru. Vrijednost je 1 samo za onu kategoriju kojoj podatak pripada, a 0 za sve ostale.

Cilj korištenja cosine similarity je da:

• Kvantitativno izmjeri koliko je neka nekretnina slična prethodno odabranoj (ciljnoj)

• Omogući **inteligentnu i personalizovanu preporuku** nekretnina korisniku

• Izbjegne preporučivanje već pregledanih nekretnina, a ponudi one koje su **najbliže po karakteristikama**

Prvi korak je iz baze dohvatiti nekretnine koje je korisnik prethodno dodao u svoju listu želja zajedno sa podacima o gradu i lokaciji.

A computer screen shot of white text

AI-generated content may be incorrect.

U slučaju da je lista želja prazna vraćamo praznu listu

A black background with white text

AI-generated content may be incorrect.

Posljednju nekretninu iz liste želja postavljamo kao cilj za upoređivanje sa ostalim

A black background with green and blue text

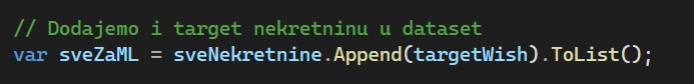
AI-generated content may be incorrect.

Idući korak je učitati sve nekretnine iz baze osim onih koje je korisnik označio kao željene

A computer screen shot of a black background

AI-generated content may be incorrect.

Dodajemo i targerWish u dataset kako bi ga mogli uporediti sa ostalima



Za svaku nekretninu pravi se objekt NekretninePodaci sa:

• Lokacijom (grad)

• Brojem soba

• Cijenom

A screen shot of a computer program

AI-generated content may be incorrect.

Pretvaramo u ML.NET IdataView



TRANSFORMACIJA PODATAKA

1. **One-hot encoding** za Lokacija jer je to kategorijska vrijednost. (prethodno je objasnjen pojam one-hot encodinga)
2. **Spajanje svih atributa** u Features vektor.

Atributi su karakteristike koje opisuju objekat (nekretninu), npr:

• Broj soba (numerički)

• Cijena (numerički)

• Lokacija (kategorijski → one-hot encoded vektor)

Model očekuje da dobije **jedan jedinstveni ulazni vektor sa svim relevantnim informacijama**. Spajanjem svih atributa u jedan vektor ("Features") pravimo numerički prikaz nekretnine koju model može procesirati.

**Kako izgleda?**

• Lokacija je pretvorena u one-hot vektor, npr. [1, 0, 0]

• Broj soba je, recimo, 3 → [3]

• Cijena je, recimo, 150000 → [150000]

**Krajnji izgled: Features = [1, 0, 0, 3, 150000]**

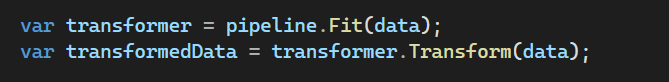
1. **Normalizacija** kako bi svi podaci bili u istom rasponu.

Normalizacija je proces **skaliranja numeričkih vrijednosti** tako da budu u istom rasponu, npr. između 0 i 1. Time osiguravamo da atributi s velikim vrijednostima (kao što je cijena nekretnine) ne dominiraju nad atributima s manjim vrijednostima (kao što je broj soba). Ako ne normalizujemo, model može pogrešno naučiti da je cijena mnogo važniji faktor samo zato što ima veće numeričke vrijednosti.

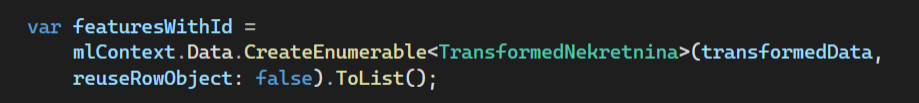
A screen shot of a computer

AI-generated content may be incorrect.

Izvršava se pipeline nad podacima



Dobija se lista TransformedNekretnina sa NekretninaId i Features (vektor osobina).



Računa se **cosine similarity** između ciljane i svake druge nekretnine i zadržavamo **top 3** najsličnije. Cosine similarity je sličnost između dva vektora (ugao između njih). Veća vrijednost znači veća sličnost.

A screenshot of a computer program

AI-generated content may be incorrect.

Iz baze se dohvaćaju informacije o preporučenim nekretninama. Mapper ih mapira na *Model.Nekretnina* da bi mogli biti poslani vani (npr. frontend).

A black screen with white text

AI-generated content may be incorrect.

COSINE SIMILARITY FUNKCIJA

Izračunava **cosine similarity** između dva n dimenzionalna vektora.

A computer screen shot of a black screen with white text

AI-generated content may be incorrect.