Zpráva k projektu z predmetu BI-VWM.21

Názov projektu

Kolaborativní filtrování – Filmový portál

Zoznam členov

Kryštof Jelínek – jelinkry , Jakub Ďurkovič - durkojak

Popis projektu

Cieľom projektu bolo vytvoriť webovú aplikáciu odporúčajúcu produkty na základe preferencií prihlásených používateľov. Aplikácia, inšpirovaná portálom IMDB, umožňuje používateľom hodnotiť filmy, na základe čoho následne odporúča ďalšie filmy. Na vstupe sme mali databázu používateľov, produktov a hodnotení, a na výstupe zoznam odporúčaných produktov podľa budovaných preferencií používateľov.

Spôsob riešenia

Pre implementáciu tohto problému sme použili tri rôzne vzorce pre výpočet odporúčaných filmov. Použili sme kosínovu podobnosť, Spearmanov koeficient a Pearsonov koeficient. Následne výsledky z týchto jednotlivých vzorcov sme použili do Resnickovho prediktora, ktorého výsledok sme použili na odporúčané filmy.

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

Kosínova podobnosť

$$\rho = 1 - \frac{6 \sum d_i^2}{N(N^2 - 1)}$$

Spearmanov koeficient

$$p_{a,j} = \bar{v}_a + \kappa \sum_{i=1}^{n} w(a,i)(v_{i,j} - \bar{v}_i)$$

Resnickov prediktor

$$r = \frac{\sum (x - \bar{x})(y - \bar{y})}{\sqrt{\sum (x - \bar{x})^2 \sum (y - \bar{y})^2}}$$

Pearsonov koeficient

Implementace

Projekt sme rozdelili do troch častí : **backend**, **frontend** a **databáza**.

Databáza

Pre jednoduchosť a minimálne nároky na infraštruktúru sme sa rozhodli použiť SQLite. Dáta čerpáme z MovieLens 100K datasetu, ktorý obsahuje 100 000 hodnotení od tisíc používateľov a približne 1700 filmov, čo presne vyhovovalo rozsahu nášho semestrálneho projektu. Na načítanie,

transformáciu a vloženie týchto údajov do databázy sme vyvinuli v Pythone skripty, ktoré automaticky parsujú pôvodné súbory a ukladajú dáta do definovaných tabuliek.

Frontend

Frontend jsme implementovali v jazyce TypeScript s využitím knihovny React a nástroje Vite pro rychlou a efektivní kompilaci. Uživatelské rozhraní jsme navrhli minimalisticky, s důrazem na jednoduchost a přehlednost. Aplikace umožňuje uživatelům registraci a přihlášení, po kterém mají k dispozici rozhraní pro vyhledávání filmů – vyhledávací pole podporuje našeptávání podle zadaného textu. Přihlášený uživatel má také přístup ke svému profilu, kde se mu zobrazují již hodnocené filmy, aktuální trendy a filmy doporučené na míru. Samostatná administrátorská stránka umožňuje spouštět rekalkulace trendů i personalizovaných doporučení pro všechny uživatele. Celý frontend komunikuje s backendem pomocí REST API.

Backend

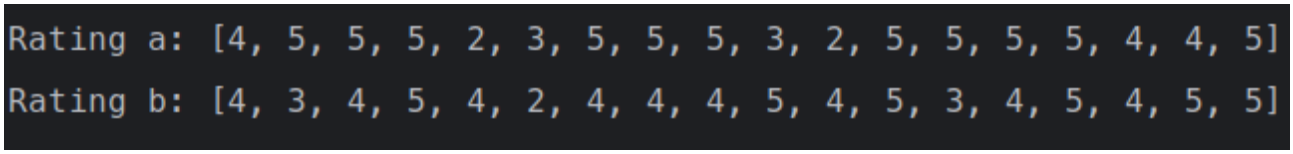
Backend sme postavili v Pythone s využitím minimalistického frameworku Flask, ktorý nám umožnil jednoduchú prácu s používateľskými reláciami aj SQL tabuľkami. Celú aplikáciu sme rozdelili do troch vrstiev – Controller na spracovanie požiadaviek, Service pre obchodnú logiku a Repository na prácu s databázou – pričom doménové entity máme definované v Models. Jadro celej funkcionality nájdete v súbore similarity_funcs, kde bežia výpočty Pearsonovho aj Spearmanovho koeficientu, kosínusovej podobnosti a samotného Resnickovho prediktora.

Požadavky na spustenie aplikácie

Všetky knihovny použité a ich správne verzie sú definované v requirements.txt, štandardný postup u Flask projektov. Návod na spustenie databaze, frontendu a backendu je konkrétnejšie popísaný v README.md.

Príklady vstupu a výstupu

Spearmanova, Pearsonova a kosínusová funkcia nám vráti podobnosť medzi dvoma užívateľmi, prikladáme output testovacích funkcií, ktoré hodnotili podobnosť užívateľa 1 a užívateľa 2. Vstupom týchto funkcíi sú všetky hodnotenia užívateľa 1 a všetky hodnotenie užívateľa 2. Následne sa nájdu filmy, ktoré obaja hodnotili a tieto hodnotenie sa použiju na hodnotenie (viz obrázok)



Spearman Test

```
Spearman similarity between user 1 and user 2: 0.274
```

Pearson Test

```
Pearson similarity between user 1 and user 2: 0.161
```

Cosine Test

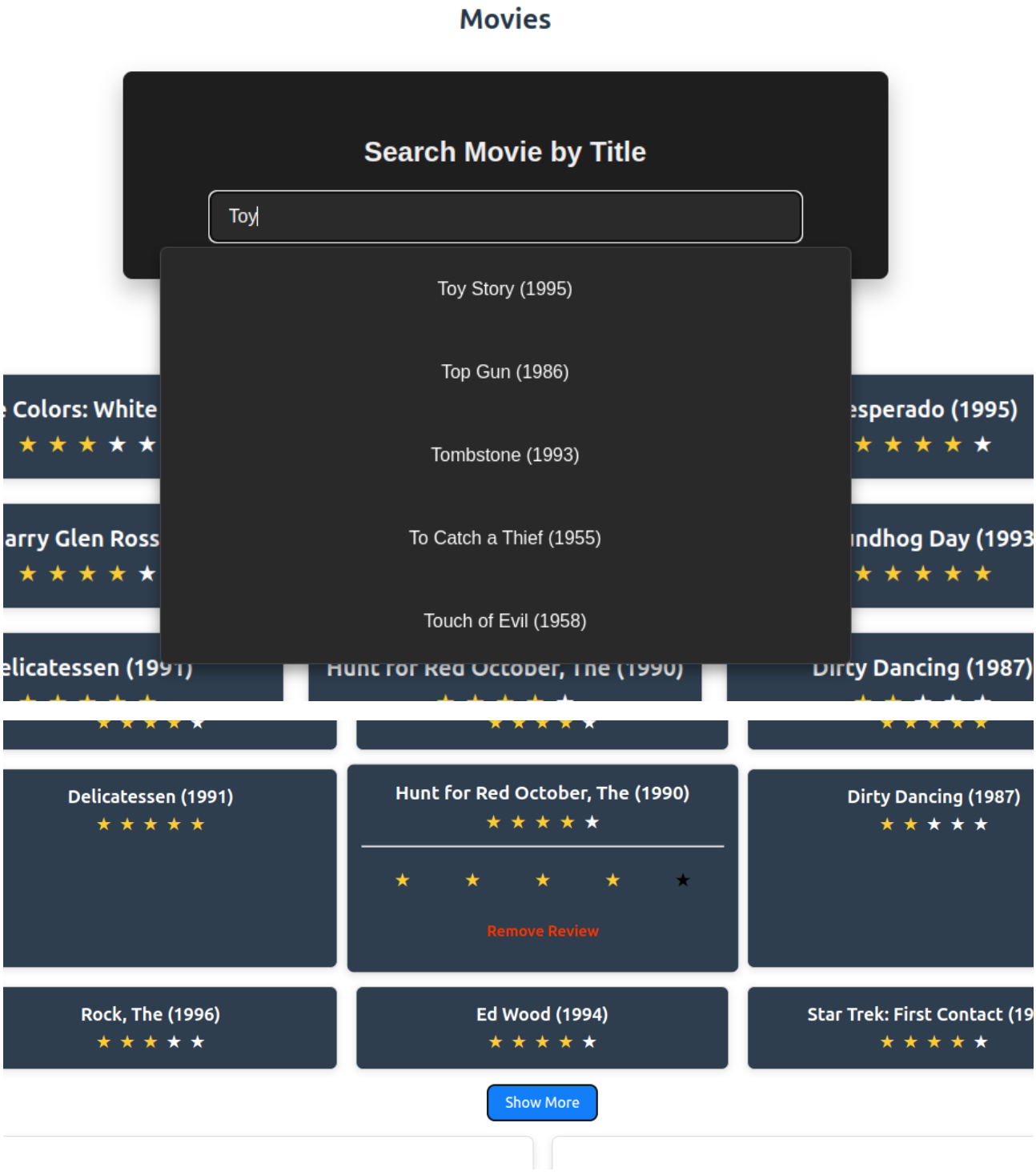
```
Cosine similarity between user 1 and user 2: 0.961
```

Ďalej prikladáme výsledok Resnickovho prediktoru, teda predpokladané hodnotenie filmov, ktoré ešte neboli hodnotené.

```
Recommendations for user 1:
Movie ID 292 → Predicted Rating: 5.00
Movie ID 306 → Predicted Rating: 5.00
Movie ID 881 → Predicted Rating: 5.00
Movie ID 286 → Predicted Rating: 4.87
Movie ID 887 → Predicted Rating: 4.80
Movie ID 1025 → Predicted Rating: 4.80
Movie ID 330 → Predicted Rating: 4.80
Movie ID 299 → Predicted Rating: 4.75
Movie ID 904 → Predicted Rating: 4.66
Movie ID 311 → Predicted Rating: 4.66
```

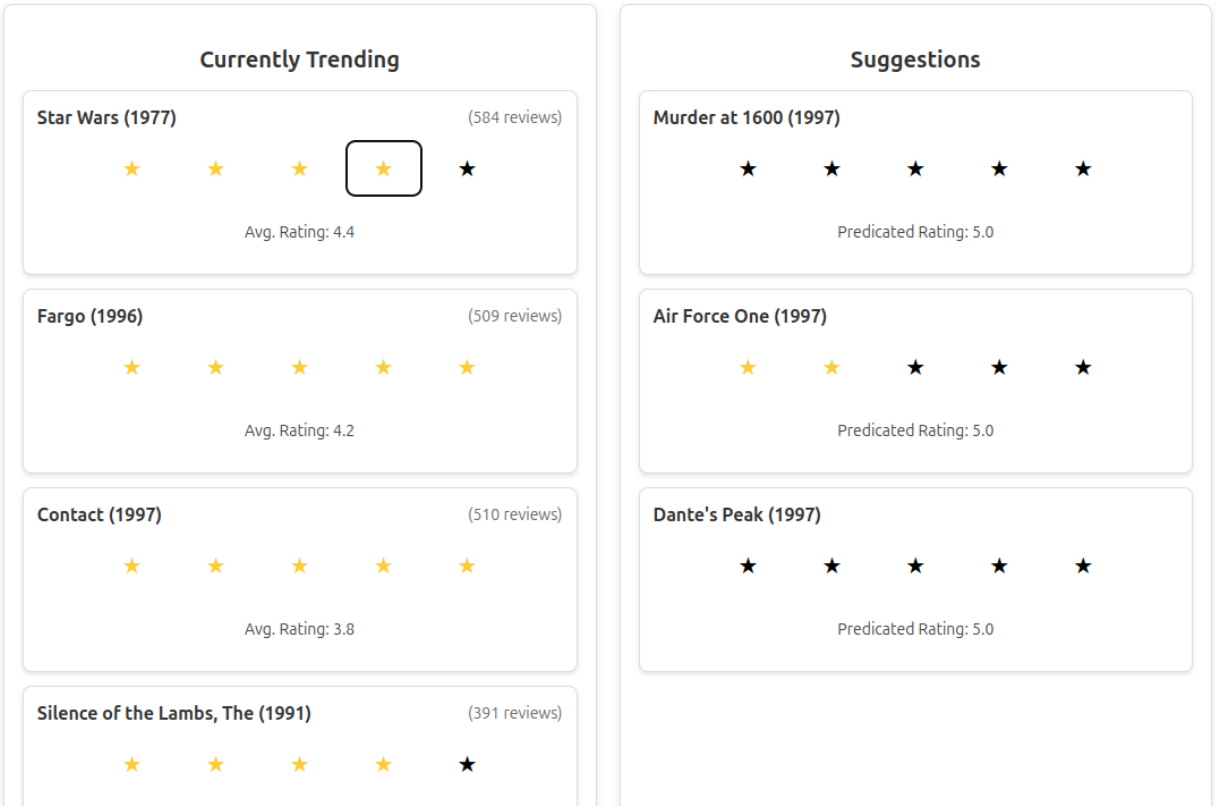
Vyhledávání filmů

Vyhledávací pole dynamicky načítá výsledky z backendu na základě zadaného textu. Doporučení jsou generována s využitím textové podobnosti (např. Levenshteinovy vzdálenosti), což umožňuje návrhy i při drobných překlepech nebo neúplném zadání názvu filmu.



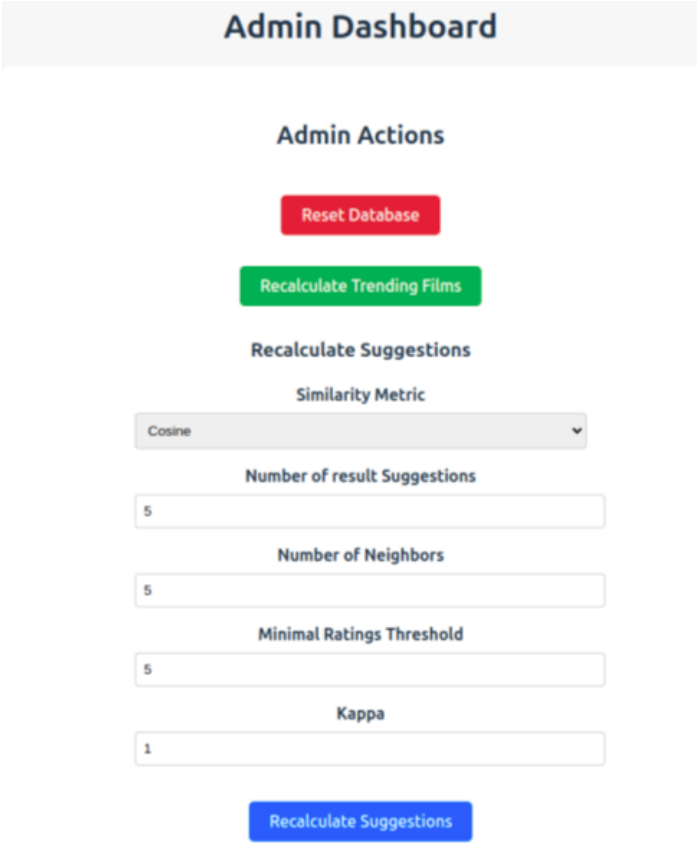
Aktuálně trendující filmy a personalizovaná doporučení

Sekce zobrazuje seznam filmů, které jsou aktuálně populární na základě počtu a kvality hodnocení v nedávném časovém období. Data jsou aktualizována po spuštění přepočtu administrátorem.Uživateli jsou zobrazena doporučení založená na kolaborativním filtrování. Výsledky jsou generovány při přepočtu na základě hodnocení ostatních uživatelů s podobným vkusem a jsou načítány z databáze po spuštění aktualizace adminem.

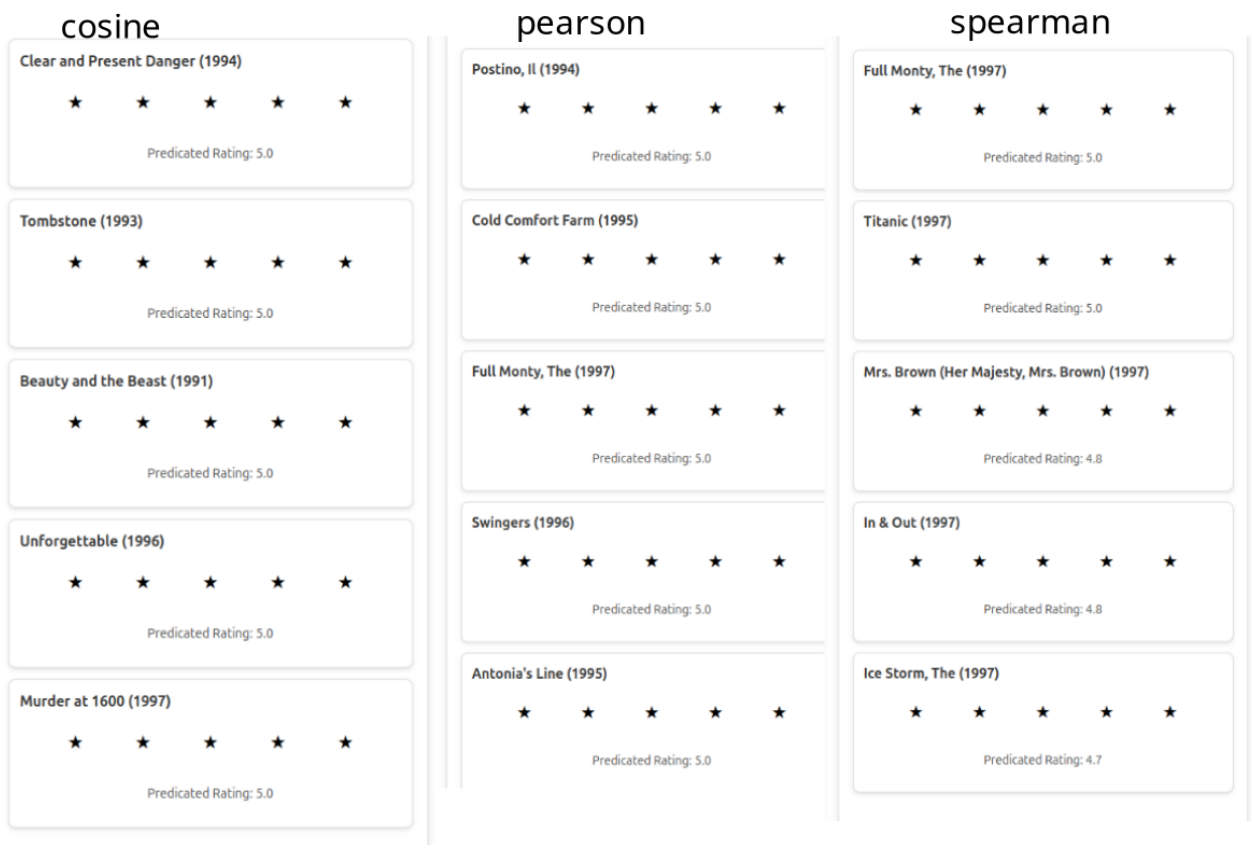


Admin Dashboard

Administrátorské rozhraní umožňuje resetovat databázi, spustit přepočet aktuálně trendujících filmů a přepočet doporučení pro jednotlivé uživatele s konkrétními parametry. Slouží primárně k otestování funkčnosti systému a aktualizaci dat.



Experimentálni sekce



Pro experimentální ověření jsme porovnali výstupy tří různých metrik podobnosti (vždy se stejnými vstupními parametry), které slouží jako základ pro doporučování filmů konkrétnímu uživateli: **kosínova podobnosť**, **Spearmanova korelace** a **Pearsonova korelace**. Na obrázcích níže jsou zobrazeny výsledky pro stejného uživatele a stejný stav databáze.Ačkoli všechny metody predikují podobné hodnocení (většinou 5.0), seznamy doporučených filmů se liší. To je způsobeno tím, že každá metrika hodnotí podobnost mezi uživateli jiným způsobem : **Kosínova podobnosť** bere v úvahu úhel mezi vektory hodnocení, a tedy je více citlivá na směr preferencí než jejich absolutní hodnotu.**Spearmanova korelace** je založena na pořadí hodnocení a ignoruje rozdíly mezi konkrétními hodnotami, což z ní dělá robustní metodu vůči odlehlým hodnocením.**Pearsonova korelace** měří lineární závislost mezi hodnoceními a zohledňuje i rozdíly v průměrném hodnocení uživatelů.Rozdíly v doporučených filmech jsou tedy důsledkem toho, jak jednotlivé metriky interpretují „podobnost“ mezi uživateli. Například Spearman může více zvýhodnit uživatele s podobným pořadím filmů, zatímco Pearson se zaměří na ty, kteří dávají hodnocení srovnatelně „vysoko nebo nízko“ jako cílový uživatel.

Pri pevne nastavenej kosínovej podobnosti a kappa = 1 sme skúmali vplyv počtu susedov k pre používateľa 1: pri k = 1 je odporúčací výstup riadený jediným najpodobnejším používateľom, čo vedie k úzkemu výberu titulov a veľkej variabilite predikovaných hodnotení (od 5,00 po 3,97), pri k = 5 a k = 10 sa s rastúcim počtom susedov predikované hodnotenia zhlukujú k hornému limitu (mnoho 5,00) a zoznam sa rozširuje o filmy obľúbené viacerými používateľmi, a pri k = 50 sa zoznam top 10 a ich skóre úplne stabilizuje – pridanie ďalších susedov už nemá žiadny efekt; tieto výsledky naznačujú, že veľmi malá susedská množina (k = 1) spôsobuje vysokú variabilitu a úzky výber filmov, zatiaľ čo pri k ≈ 10 je konsenzuálny signál nasýtený, a preto použitie k v rozsahu 5–10 prináša prakticky identické odporúčania ako oveľa vyššie k, pričom šetrí výpočtový čas.

```
Recommendations for user 1:
Movie ID 306 → Predicted Rating: 5.00
Movie ID 321 → Predicted Rating: 4.97
Movie ID 286 → Predicted Rating: 4.97
Movie ID 294 → Predicted Rating: 3.97
Movie ID 319 → Predicted Rating: 3.97
Movie ID 292 → Predicted Rating: 3.97
Movie ID 990 → Predicted Rating: 3.97
Movie ID 288 → Predicted Rating: 3.97
Movie ID 331 → Predicted Rating: 3.97
Movie ID 872 → Predicted Rating: 3.97
```

k = 1

```
Recommendations for user 1:
Movie ID 286 → Predicted Rating: 5.00
Movie ID 292 → Predicted Rating: 5.00
Movie ID 331 → Predicted Rating: 5.00
Movie ID 302 → Predicted Rating: 5.00
Movie ID 881 → Predicted Rating: 5.00
Movie ID 315 → Predicted Rating: 5.00
Movie ID 904 → Predicted Rating: 5.00
Movie ID 313 → Predicted Rating: 5.00
Movie ID 898 → Predicted Rating: 5.00
Movie ID 354 → Predicted Rating: 5.00
```

k = 50

Diskuze

Jedným z technických nedostatkov tohto projektu je, že aktuálne tato aplikácia beží iba lokálne a ako databázu využíva Sqlite. Pre malý semestrálny projekt je toto vystačujúce, avšak keby sme z toho chceli spraviť niečo väčšie, určite by bolo dobré premigrovať na postgres a hostovať to na cloude. V aktuálnej verzii projektu prepočítavame odporúčania tak, že pre každého užívateľa to samostatne prepočítame úplne odznova, čo pri mnoho užívateloch môže trvať desiatky až stovky minút a zaťažuje server. Pre hladšie prevádzkovanie by bolo efektívne prejsť na inkrementálne updaty len nových ratingov alebo rozdeliť výpočty do menších blokov počas dňa, prípadne využiť predpočítané faktorizácie či aproximované vyhľadávanie susedov na skrátenie doby batch jobu. Existuje mnoho python knihoven, ktoré by tento problém zjednodušili, ako napríklad Annoy, ktorý využíva aj Spotify, aplikácia na streamovanie hudby, ktorá vám taktiež na základe trendov odporúča nové pesničky.

Záver

Cieľom projektu bolo vytvoriť webovú aplikáciu, ktorá na základe historických hodnotení používateľov a aktuálnych trendov generuje odporúčania filmov. Implementovali sme jednoduché používateľské rozhranie, spoľahlivý backend pre správu používateľov a výpočet hodnotení a

databázu na ukladanie všetkých dát. Keďže išlo o semestrálnu prácu, zamerali sme sa na základnú funkcionalitu odporúčacieho systému a ponechali priestor na ďalšie vylepšenia, optimalizácie a rozšírenie možností personalizácie.