III. - Kolaborativní filtrování

Makarov Danil, Olexandr Burakov

2023 FIT CVUT

Popis projektu

Cílem našeho projektu bylo vytvořit full-stack aplikaci, která slouží jako příklad demonstrace algoritmu **kolaborativního filtrování**. Jako bonus jsme se rozhodli přidat také **systém doporučování filmů**, nejen předpověď hodnocení pro film.

Způsob řešení

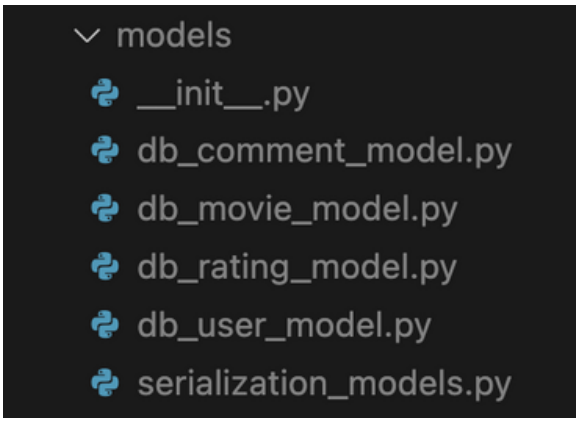
Byly použity následující algoritmy:

- **User-based Collaborative Filtering**
- **Spearmanův algoritmus pro výpočet korelace**

Implementace

Frontend naší aplikace je napsán v **ReactJS** s využitím **RTKQ** (Redux Toolkit Query se používá k vytváření API na frontendu). Backend naší aplikace je napsán v **Flask** (Python).

Pro ukládání dat jsme použili **SQLite**. Je dostatečně lehká a pro naše účely jsme nepotřebovali širší databáze. Celkem máme 4 modely: **Komentáře**, **Film**, **Hodnocení**, **Uživatel**. Existuje také další pomocný model **Serializace**. Používá se pro pohodlné vrácení dat na frontend.



Modely databáze

Struktura backendu je docela typická pro lehké backendové aplikace. Máme zde:

- **Kontrolery**: Funkce, které jsou volány na endpointech.
- **Složka exts:** Obsahuje konfiguraci pro databázi SQLite.
- **Models**: Jak bylo řečeno, obsahuje modely pro databázi.
- **Recsys**: Složka, která obsahuje hlavní soubory pro naše algoritmy výběru filmů.
- **Routes**: Inicializace endpoints.

Nyní bychom rádi hovořili podrobněji o třídách **SpearmanMechanism** a **RecMechanism**.

Třída **SpearmanMechanism** je určena pro výpočet **Spearmanova koeficientu** hodnocení mezi dvěma uživateli na základě jejich hodnocení filmů. Třída má následující **atributy**:

- **target_user**: Primární uživatel.
- **another_user**: Sekundární uživatel.
- **common_rated_movies**: Seznam filmů, které oba uživatelé hodnotili.
- **target_user_rated_movie_ranks**: Slovník s ranky pro cílového uživatele vzhledem k společně hodnoceným filmům s druhým uživatelem.
- **another_user_rated_movie_ranks**: Slovník s ranky pro druhého uživatele vzhledem k společně hodnoceným filmům s cílovým uživatelem.
- **squared_common_rated_movies_rank_diffs_sum**: Součet čtverců rozdílů hodnocení u filmů, které hodnotili oba uživatelé.
- **spearman_correlation_coefficient**: Spearmanův koeficient hodnocení mezi dvěma uživateli na základě jejich hodnocení filmů.

Metody:

- **__init__**: Inicializace třídy s dvěma uživatelskými objekty.
- **get_spearman_correlation_coefficient**: Getter pro Spearmanův koeficient korelace (byl vypočítán dříve).
- **_get_rated_movie_ranks**: Tato metoda vypočítá a vrátí hodnocení filmů, které uživatelé společně hodnotili.
- **_get_common_rated_movies**: Získává společné filmy, které hodnotil cílový uživatel a další uživatel.
- **_get_common_rated_movie_ranks_squared_diffs_sum**: Vypočítává součet čtverců rozdílů v hodnocení společných filmů mezi dvěma uživateli.
- **_calculate_spearman_correlation_coefficient**: Tato metoda vypočítává Spearmanův koeficient hodnocení na základě součtu čtverců rozdílů hodnocení a počtu společně hodnocených filmů.

Třída **RecMechanism** je určena pro doporučování filmů cílovému uživateli na základě Spearmanova koeficientu a výpočet předpokládaného hodnocení pro konkrétní film. Hlavní metody třídy jsou:

- **__init__**: Inicializuje třídu s cílovým uživatelem a seznamem všech ostatních uživatelů. Vypočítá Spearmanovy koeficienty hodnocení mezi cílovým uživatelem a všemi ostatními uživateli.
- **get_spearman_correlation_coefficients**: Vrací slovník Spearmanových koeficientů hodnocení, který byl dříve vypočítán.

- **get_recommendations**: Vrací seznam doporučených filmů pro cílového uživatele. Tento seznam se vypočítává na základě Spearmanových koeficientů hodnocení.
- **get_predicted_rating_for_movie**: Vrací předpovězené hodnocení pro konkrétní film pro cílového uživatele. Předpovězené hodnocení se vypočítává na základě Spearmanových koeficientů hodnocení mezi cílovým uživatelem a nejbližšími sousedy.

Soukromé metody třídy jsou:

- **_calculate_spearman_correlation_coefficients**: Vypočítá Spearmanovy koeficienty hodnocení mezi cílovým uživatelem a všemi ostatními uživateli.
- **_calculate_recommended_movies**: Vypočítá doporučené filmy pro cílového uživatele na základě Spearmanových koeficientů.
- **_calculate_predicted_rating_for_movie**: Vypočítá předpovězené hodnocení pro konkrétní film pro cílového uživatele na základě Spearmanových koeficientů hodnocení mezi cílovým uživatelem a nejbližšími sousedy.

Výpočet Spearmanova koeficientu:

- Pro výpočet **Spearmanova koeficientu** mezi dvěma uživateli se porovnávají jejich hodnocení filmů.
- Nejprve se vybere množina filmů, kterou oba uživatelé hodnotili.
- Poté se pro společně prohlížené filmy tohoto uživatele vybere specifické pořadí pro určení ranků každému filmu. Filmy jsou seřazeny vzestupně podle udělených hodnocení.
- Dále je každému z těchto společně hodnocených filmů přiřazen rank pro následný výpočet korelace - speciální váhová jednotka filmu závislá na jeho hodnocení.
- Následně se spočítá součet čtverců rozdílů mezi ranky těchto filmů pro oba uživatele.
- Spearmanův koeficient se vypočítá na základě tohoto součtu a počtu filmů.

Předpověď hodnocení filmu:

- Pro předpověď hodnocení filmu pro cílového uživatele se používají vypočítané Spearmanovy koeficienty mezi cílovým uživatelem a ostatními uživateli.
- Nejbližších **MAX_NEIGHBORS** uživatelů s nejvyššími Spearmanovými koeficienty se vybere.
- Pro tyto uživatele se spočítá vážený průměr jejich hodnocení daného filmu na základě jejich Spearmanových koeficientů.

- Poté program využívá předchozí výpočty a matematickou formuli pro kolaborativní filtraci a vypočítává předpokládané hodnocení pro daného uživatele pro konkrétní film.
- Výsledkem je předpovězené hodnocení filmu pro cílového uživatele.

Doporučení filmů:

- Pro doporučení filmů pro cílového uživatele se opět využívají vypočítané Spearmanovy koeficienty.
- Uživatelé se seřadí podle jejich Spearmanových koeficientů sestupně.
- Postupně se program prochází těmito uživateli a jejich hodnoceními filmů.
- Pokud film nebyl cílovým uživatelem hodnocen a jeho hodnocení přesáhne minimální hranici hodnocení MIN_RATING a koeficientu korelace MIN_CORRELATION, je přidán do seznamu doporučených filmů.
- Seznam doporučených filmů je omezen na maximální počet doporučení MAX_RECOMMENDATIONS.

Spuštění aplikace

Pro spuštění projektu potřebujete **NodeJS** verzi **19** a **Python** verzi **3.11, pipenv** (**pip install pipenv** instalovat globálně). Budete potřebovat otevřít 2 okna terminálu - pro spuštění frontendu a backendu. Pro spuštění frontendu je třeba přejít do **složky frontend**, nainstalovat závislosti pomocí **npm install** a spustit projekt pomocí **npm start**. Pro spuštění backendu musíte přejít do **složky backend** a zadat **pipenv shell**, **pip install -r requirements.txt** a **python3 run.py**. Frontend běží na **localhost:3000**, backend běží na **localhost:5000**.

Uživatel pro testování má email: **1@gmail.com** a heslo: **111**.

Vstupy a výstupy

Spearmanův koeficient

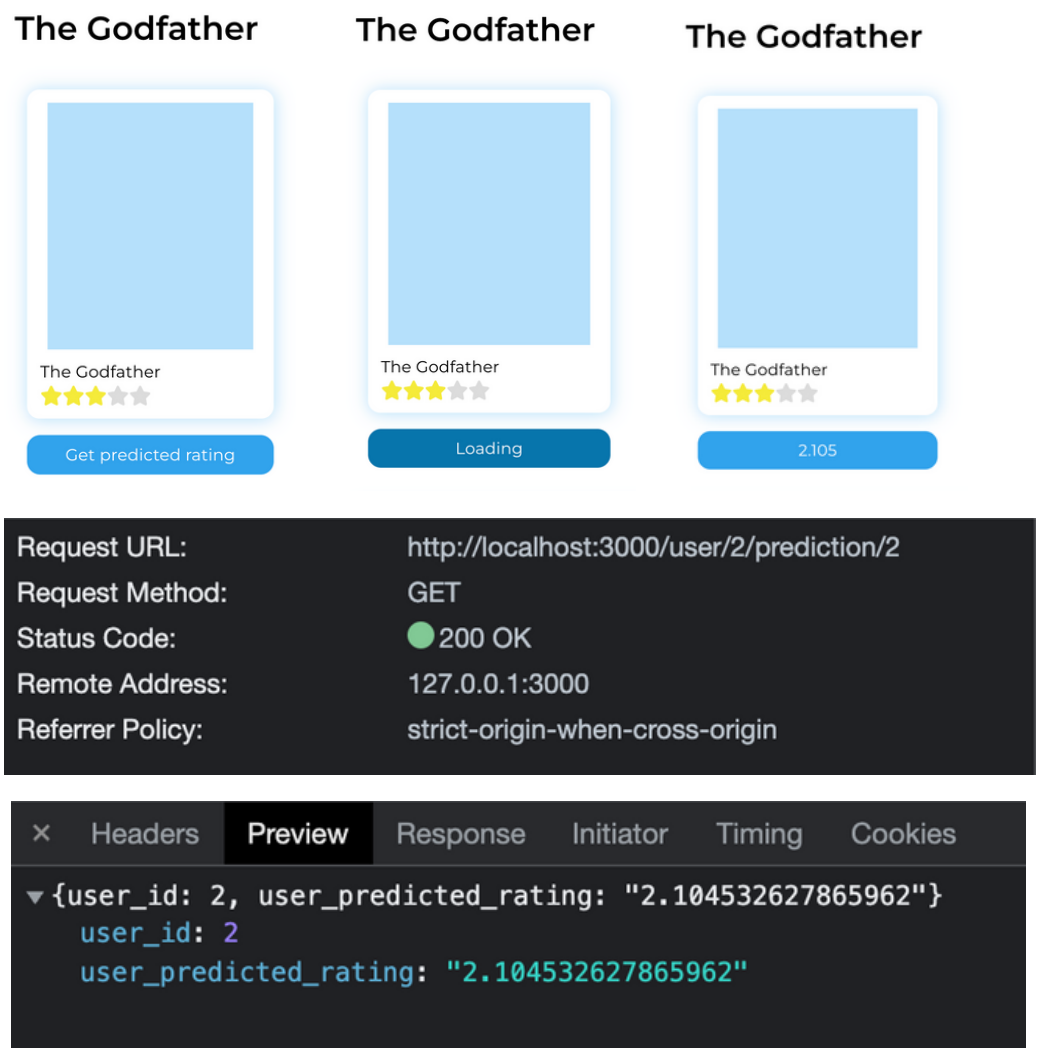
Vstup: objekt pro aktuálního uživatele a objekt pro druhého uživatele, se kterým chceme vypočítat korelaci. Objekt uživatele obsahuje pole s hodnocenými filmy (tj. id filmu a hodnocení jako číslo od 1 do 5 pro daného uživatele).

Výstup: je Spearmanův koeficient.

Předpověď hodnocení filmu

Vstup: objekt pro cílového uživatele, Spearmanovy koeficienty pro cílového uživatele a všechny ostatní uživatele, id uvažovaného filmu.

Výstup: předpovídané hodnocení jako float číslo od 1 do 5.

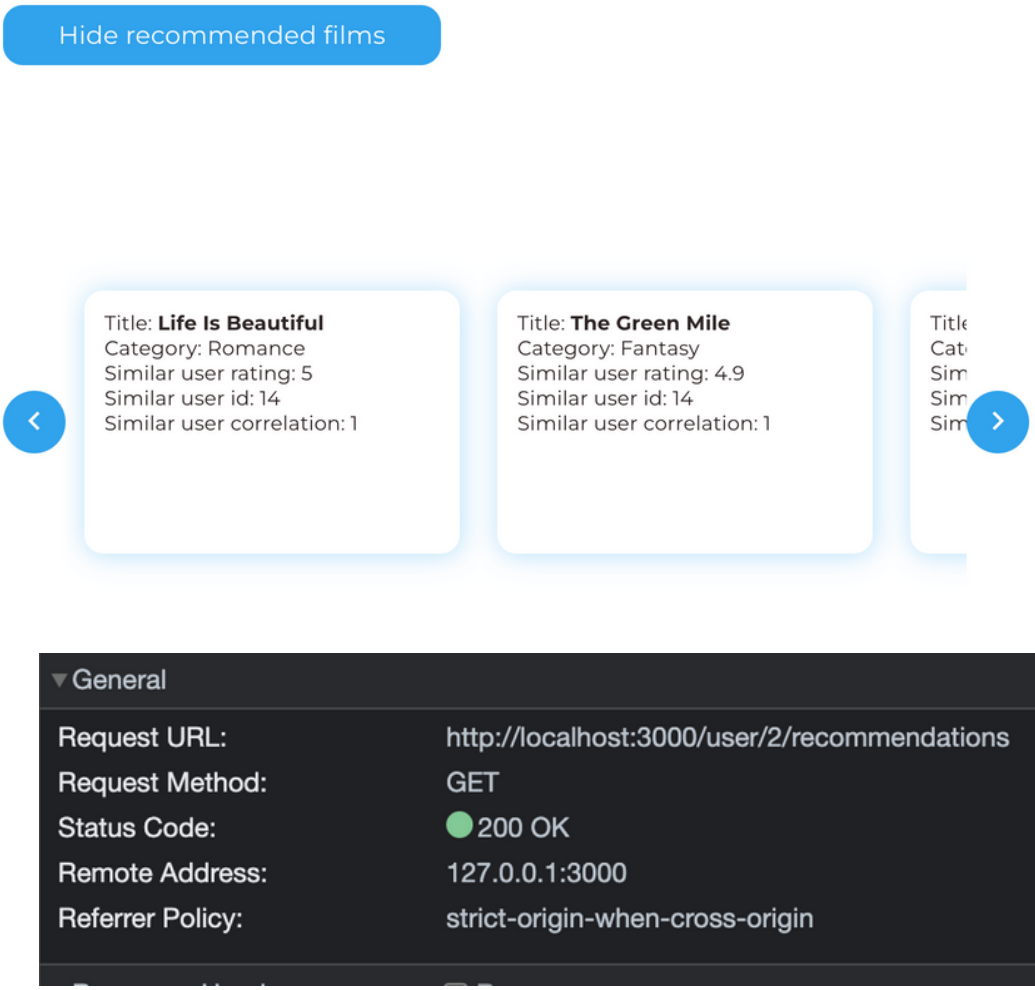


Předpověď hodnocení filmu UI

Doporučení filmů

Vstup: objekt pro cílového uživatele, Spearmanovy koeficienty pro cílového uživatele a všechny ostatní uživately.

Výstup: pole objektů obsahujících id filmu, category, title, **similar_user_id** ID podobného uživatele, který hodnotil film, **similar_user_rating** hodnocení filmu podobného uživatele, **similar_user_correlation** Spearmanův koeficient korelace mezi uživatelem a podobným uživatelem.



Doporučení filmů UI

Experimentální sekce

Výpočet Spearmanova koeficientu závisí na počtu společně hodnocených filmů mezi dvěma uživateli. Průměrná rychlost výpočtu je přibližně **18 milisekund** pro **20-30 různých filmů.**

Výpočet doporučení filmů pro našeho uživatele trvá přibližně **42 milisekund**, když v databázi existuje kolem **50** uživatelů a každý uživatel hodnotil přibližně **20-30 různých filmů**.

Výpočet doporučení filmů pro našeho uživatele trvá přibližně **74 milisekund**, když v databázi existuje kolem **50** uživatelů a každý uživatel hodnotil přibližně **20-30 různých filmů**.

Také máme parametry, které umožňují upravit chování algoritmu. V souboru **rec_mechanism.py** máme následující proměnné:

- MIN_CORRELATION: Parameter je zodvodedny za vraceni filmu, které mají korelaci vyšší než tato hodnota.
- MIN_RATING: Vrací filmy, které mají hodnocení vyšší než tato hodnota.
- MAX_RECOMMENDATIONS: Maximální počet doporučených filmů.
- MAX_NEIGHBORS: Vybere tolik nejlepších uživatelů na základě korelace pro výpočet předpovědi hodnocení filmu.

Závěr

V této semestrální práci jsme implementovali systém doporučování filmů a předpověď hodnocení filmů. Měli jsme však potíže s měřením a testováním, protože jsme ručně přidávali filmy, uživatele a jejich hodnocení filmů. Vytvoření takového testeru je samostatný velký blok. Samozřejmě pro přesnější měření bychom potřebovali tisíce uživatelů. Celkově se naše předpoklady o přesnosti doporučení filmů shodovaly. Algoritmus pracuje dostatečně přesně při dostatečném počtu potřebných pro vypočet dat a může se lišit o tisíciny (0,001). Pro srovnání výsledků je možně vyzkoušet použít jiné algoritmy pro výpočet korelace.