Mevies: Movies for me documentation

Repozitorij se nalazi ovdje: Best-API

NAPOMENA: Dupli filmovi koji se pojavljuju u web sučelju su posljedice dupliciranja u originalnoj bazi podataka

Autori: Filip Jukić i Marko Čorokalo

1. Arhitektura sustava

Na backend strani, Mevies je izveden kao Django aplikacija koji se izvršava na Linux serveru i spaja se na bazu podaka PostgreSQL. Backend je zapravo manje-više REST API baziran na Django Rest Frameworku koji izlaže podatke o filmovima i preporukama.

Na frontend strani nalazi se AngularJS aplikacija koja komunicira s API-jem te prikazuje podatke u obliku uređene HTML stranice.

U `deploy` folderu nalaze se Ansible skripte i Vagrantfile za provizioniranje servera.

2. Django aplikacija

U `/opt/app` nalazi se git repozitorij koji sadrži aplikaciju. Sam Django projekt nalazi se u `/opt/app/best_api` , u istom folderu se nalazi i requirements.txt datoteka s popisom potrebnim python paketa. Sama organizacija projekta je standardna u Django svijetu, ali za neupućene:

- `best_api/movies` : aplikacija unutar Django projekta gdje se nalaze modeli, viewovi i serializeri za API
- `best_api/best_api` : direktorij gdje se nalaze mapiranje URL-ova na viewove, WSGI skripta te postavke aplikacije
- `best_api/static` : statičke datoteke potrebne za rad stranice (CSS, JS)
- `best_api/templates` : HTML templateovi potrebni za prikaz stranice
- `best_api/media` : datoteke uploadane "od strane korisnike", u ovom slučaju poster za svaki film u bazi

Lista API endpointova:

1. `/api/movies/` - prikaz filmova iz baze, dozvoljeni parametri za filtriranje podataka iz baze:
    - title, year, actor, director - filtriranje po nekom od ovih parametara (npr. `?year=2012`)
    - ordering: year, title, id - sortiranje po parametru, s minusom ispred se dohvaća u obrnutom smjeru (`?ordering=-year`)
    - page_size - broj filmova koji će se dohvatiti

    - page - stranica na kojoj se trenutno nalazimo (straničenje)
2. `/api/recommendations` : dobivanje 10 preporuka filmova na temelju već pogledanih filmova koji se pamte u sessionu

3. Angular aplikacija

Angular aplikacija nalazi se u `best_api/static/js/app.js`

4. Algoritam za preporuke

Budući da zbog obaveza na faksu i posla nismo imali vremena za implementaciju nekog od složenih algoritama baziranih na faktorizaciji matrica, odlučili smo se za "jednostavniji" item-based algoritam. U razmatranje je ušao i Slope One algoritam koji se u našem slučaju nije pokazao dovoljno preciznim.

Algoritam radi tako da za svaki film prvo računa koji su mu najsličniji filmovi što se radi tako da se za svaki par filmova potraži funkcija sličnosti, u našem slučaju Adjusted Cosine (poglavlje 3.1.3). Budući da svaki korisnik ocjenjuje po drugačijoj skali, ova funkcija u sebi ima normalizaciju tih ocjena, te se tako uvelike smanjuje negativni efekt različitih skala ocjenjivanja. U analizu je bila uključena i Pearsonova funkcija udaljenosti, ali se pokazala kao nešto lošija te stoga nije korištena.

Da bi se dobile preporuke filmova za korisnika, on prvo mora "pogledati" jedan ili više filmova za koje smatramo da su mu se svidjeli. Zatim se pomoću prije izračunatih sličnosti među filmova dobiva lista preporučenih filmova.

Sami algoritam ima nekoliko podesivih parametara:

- `similarity_function` : funkcija sličnosti, po defaultu Adjusted Cosine
- `min_num_of_ratings` : minimalan broj ljudi koji su ocjenili oba filma da bi se on uzeo u obzir, spriječava pojavu tzv. "autsajdera" (stroga granica)
- `similar_calculate` : koliko najsličnijih filmova od nekog filma će se spremiti, tj. uzeti u obzir prilikom računanja preporuka
- `damp_factor` : vrijednost koja smanjuje "vrijednost" filmova koji nemaju puno zajedničkih ocjena (ocjena više korisnika), npr. damp faktor 20 dijeli broj zajedničkih ocjena nekog para filmova s 20 ukoliko imaju manje od 20 zajedničkih ocjena
- `training_set_percent` : koliki postotak ukupnog dataseta s ocjenama će se koristiti za treniranje modela, a posljedicom toga i koliki će se postotak koristiti za testiranje

Evaluacija algoritma je izvršena na način da su se pokretale razne kombinacije gore navedenih parametara, te se mjerio MAE (Mean Average Error) između stvarih ocjena nekog korisnika za film dobivenog iz dataseta za testiranje, te ocjene koju je algoritam "predvidio" za neki par (osoba, film). Što je manji MAE, to je u pravilu algoritam točniji.

Nakon oko 150 iteracija, n``ajpreciznij po MAE-u te subjektivno najbolji algoritam imao je sljedeće parametre:

- `similarity_function` : Adjusted Cosine

- `min_num_of_ratings` : 30
- `similar_calculate` : 75
- `damp_factor` : 50
- `training_set_percent` : 90%

MAE za te parametre je bio **0.735322844747**.

Kako bi se izbjegla kalkulacija prilikom svakog učitavanja stranice, ranije izračunati model se učitava ili iz cachea (memcached), ili s diska.