
Movie Recommendation System (v. 0.1)

Team: Hello world
임정민, 현규환, 오동휘, 김기태

Function Recommend movies based on users taste

Description Recent on-demand video streaming services such as Netflix and Disney+ strongly utilize recommendation systems to make users stay in the services as long as possible. The system gathers a list of user preferred movies, a list of recently watched videos, and recently searched history. The result of the system would be a list of movies user might enjoy. The system can make use of gathered information provided by users to compute the relevant movies.

Input

- History of movie which user watched
- Keywords user searched
- User preferred movies

Source None

Output Ranked relevancy list of movies user might enjoy

Destination User browser client

Action

A user who is willing to use the recommendation system should access the client(webpage) and sign in or up. If a new user signs in, the system shows a list of movies and asks user to select his/her watched or preferred movies. Then, the client sends the list to the server and the server store the information in the user's account.

With a list of preferred movies, the server picks relevant movies based on the information and rank them by relevancy. Then, the server sends the ranked list to client.

The client shows provided ranked list to user with grid layout: showing thumbnail and some simple information. Details of a movie (title, genre, language, actors, thumbnail) are shown by clicking a grid of a movie. User can search a movie on client and can set/unset a movie as a preferred movie. Manager can add or edit movie info after signing in with a admin account.

Requirements

- There should be five pages called "Login Page", "Admin Page", "Bootstrap Page", "Result Page", "MovieDetailView Page".
- The "Login Page" will be shown to user initially. This page receives user's ID and password.
- There should be two buttons named "Login Button" and "Register Button" in the login page. When user hits login button, the page sends user's ID and password to server then the server sends back the result page if user successfully logs in. Log in fails if the given password dose not match with password of user searched using given ID in the database or user with the given ID dose not exists.
- When user hits register button, the page sends them to server and server sends back the bootstrap page only if the user account can be created successfully.

- The “Admin Page” have two buttons for search and add a movie. If a manager searches a movie, client moves to “MovieDetailView Page” of corresponding movie. If a manager press a button for adding a new movie, client moves to blank “MovieDetailView Page”.
- The “Bootstrap Page” will be shown after user successfully creates new account. This page takes new user’s ID, password, and initial list of preferred movies.
- The “Result Page” will be shown when user successfully logs in. This page contains list of recommended movies calculated based on the user’s list of preferred movies.
- “MovieDetailView Page” shows details of a movie. First, it checks if a user is admin or not and if a user is admin, user can modify the information or remove the movie. If not, the page only shows the information and the information cannot be edited.
- The system should initialize DB with data in csv files when new csv files are presented.
- The system should return a number of users when admin user requested it.
- User can rate movie in range of 1.0 to 5.0 in 0.5 scale.
- The recommendation system and DB should work on docker container using docker-compose.
- User can edit his/her ratings of a movie.
- All result should be returned in a JSON format.

Pre-condition

- There should exist a list of movies and a list of ratings for each users.

Post-condition None

Side-effects None