CGS616: Human Centered Computing

PROJECT REPORT

Project 2: Recommended Systems

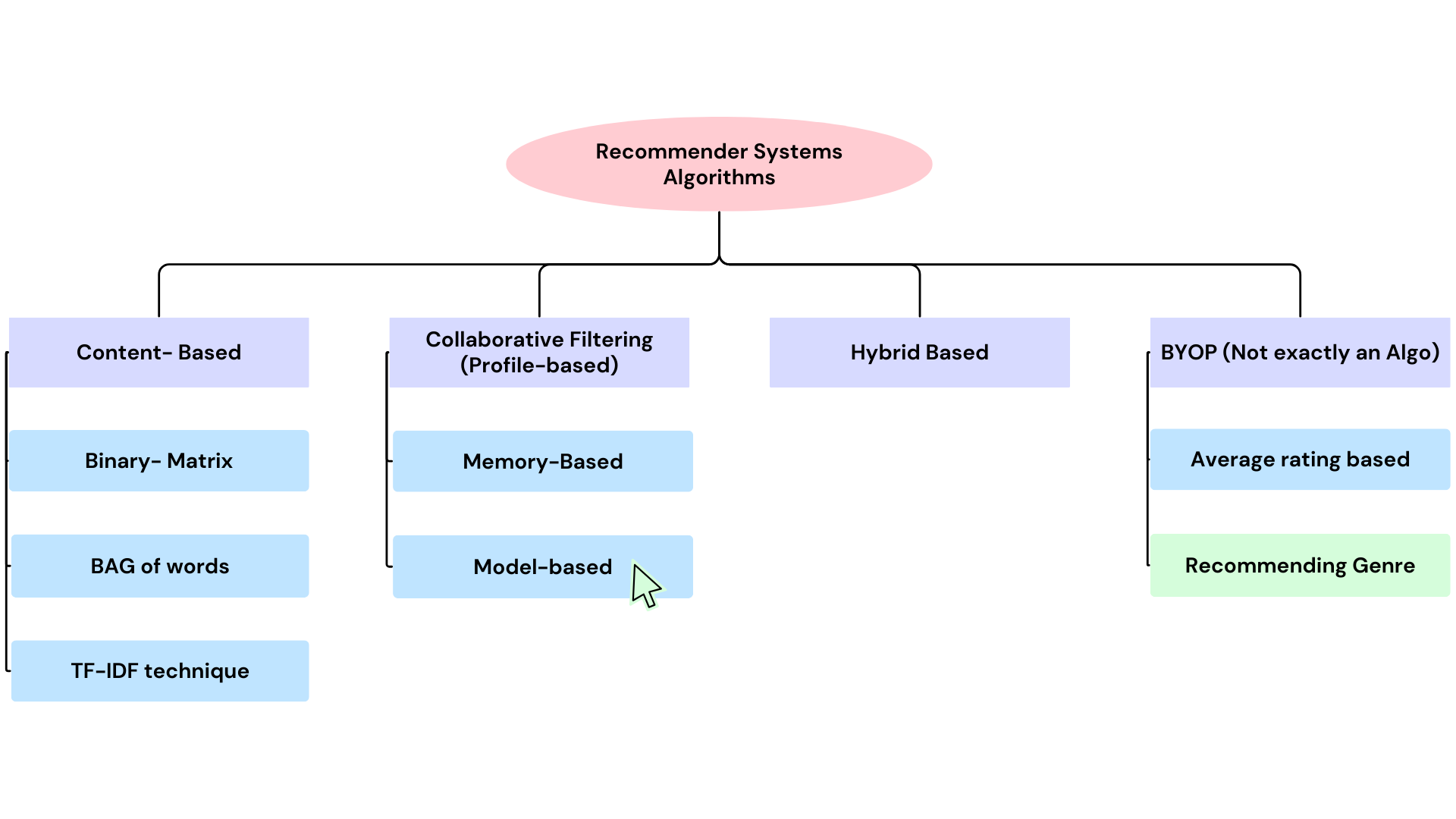
* Algo Alchemist

1. Adithya (220587)
2. Yash (221218)

**Aim:** To apply recommender system algorithms to predict movies for a user based on dataset that has movie ratings given by people.

**Recommended System**: A machine learning model that simply predicts and squeezes down the prospective output among an exponentially growing system.

**Data Available:** Movielens 100k dataset

Used Recommended Algorithm:

1. **Content Based**

This analyzes what we have preferred and watched before and consequently builds a profile based on our interests to recommend movies that we may like.

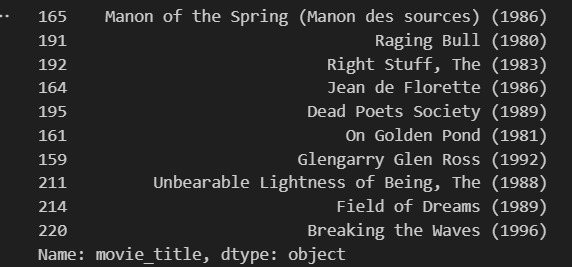
1.1) Binary-Matrix:

The matrix shows each movie's "genre profile." The system compares these profiles to find movies similar to ones we have liked before (according . to the dataset). By finding movies with matching genres (similar profiles), the system recommends movies we might enjoy.

Code Snippet:



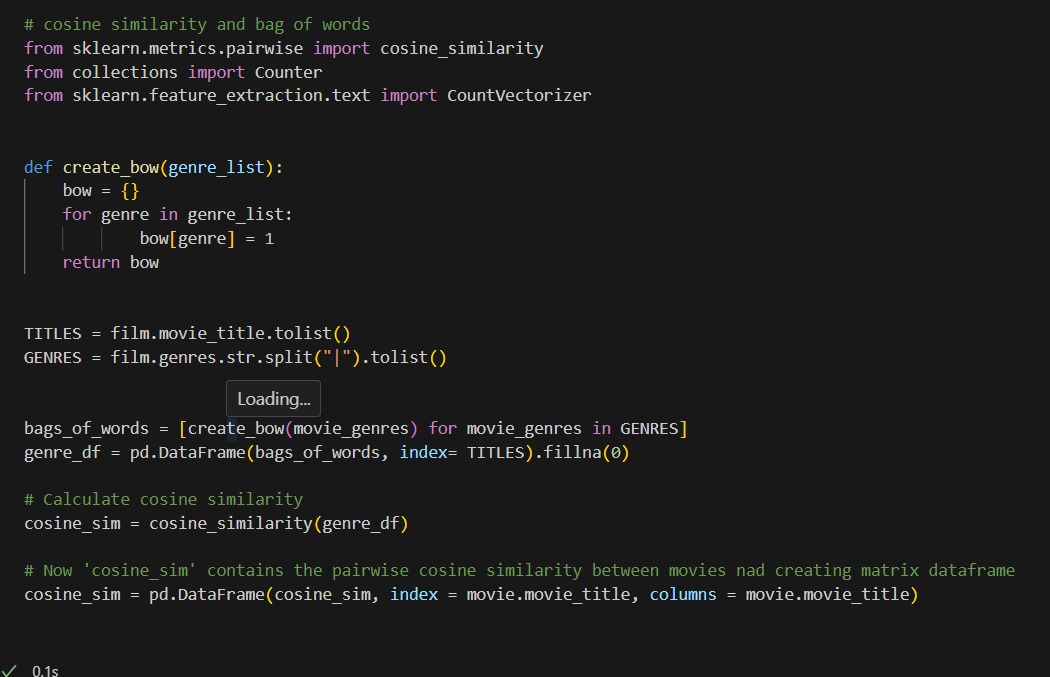
Output:



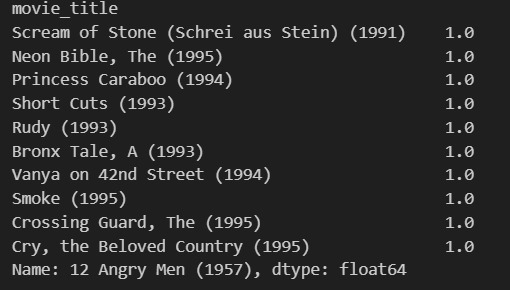
1.2) BAG of words

Uses a summary of words to describe and characterize a movie from its title and the system compares these "bags of words" to see how similar movies are based on their title. It then recommends movies that share many words with movies we have liked before.

Code Snippet:



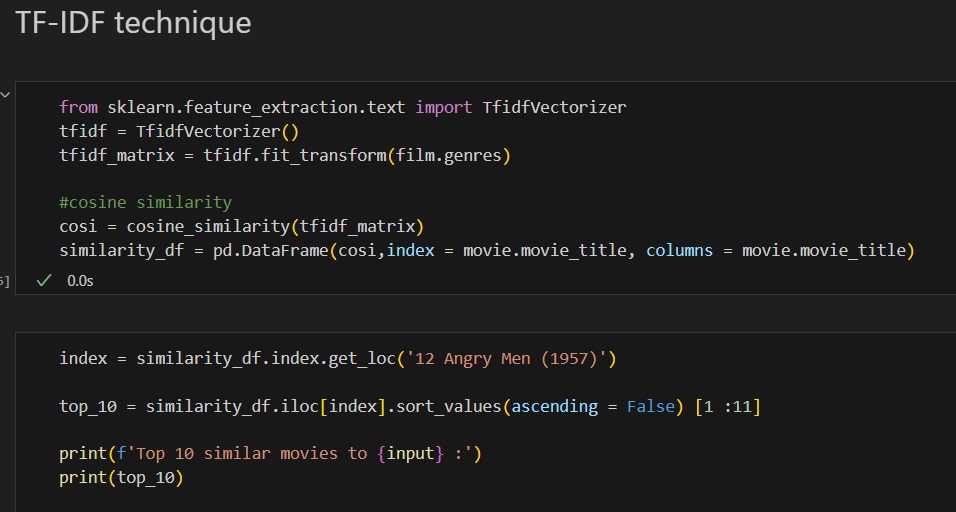
Output:



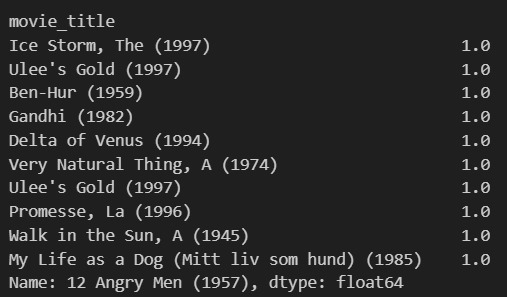
1.3) TF-IDF Technique

Genres are vectorized, and subsequently, the cosine angle between various genres is calculated, and the content-rich vectors are compared to find movies with similar genre.

Code Snippet:



Output:



1. **Collaborative Filtering**

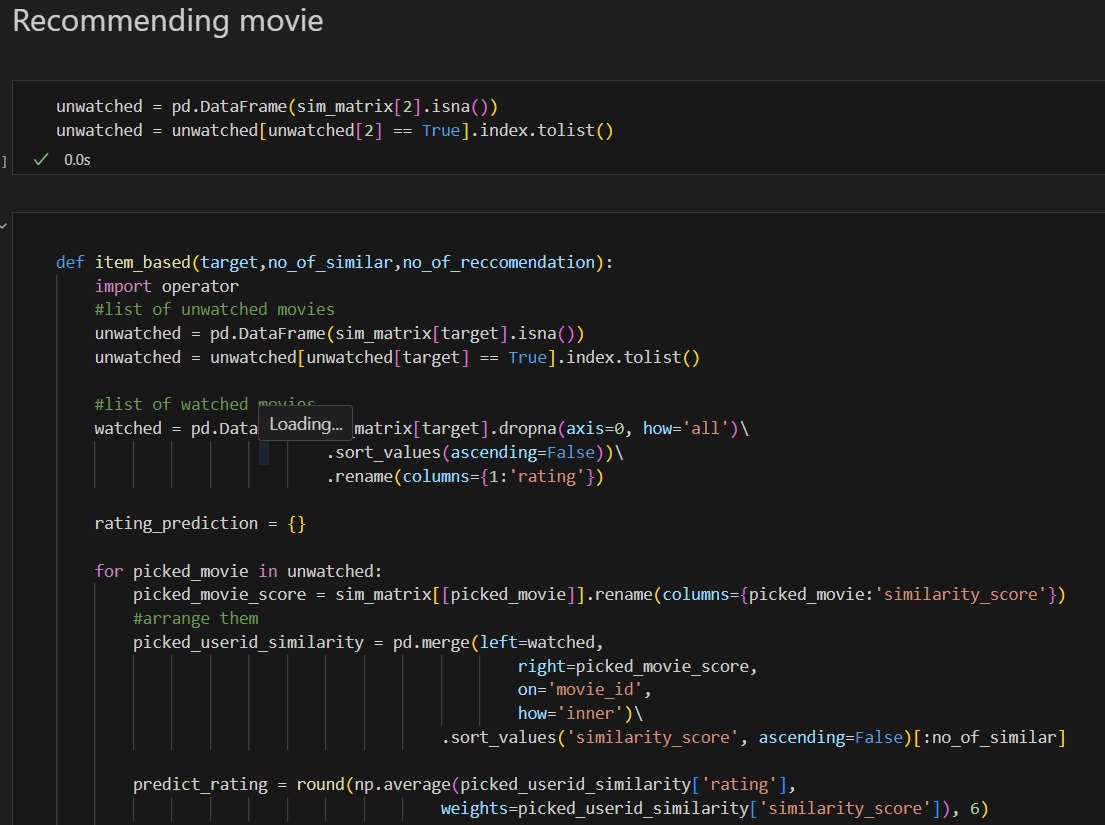
In collaborative filtering, recommender systems find people with similar tastes and recommend what those similar users liked. This is like getting suggestions from a friend with similar interests.

2.1) Memory-based: Cosine similarity or Pearson similarity is used. In the case of Cosine similarity, we truncate the N/A values to 0 (which implies disliking), while Pearson similarity omits N/A values and doesn’t account for these null values to avoid ambiguity.

(Item-Item) Code Snippet 1:

Extracts similarity from item-item matrix and recommends unwatched movies by the target user and predicts their rating.

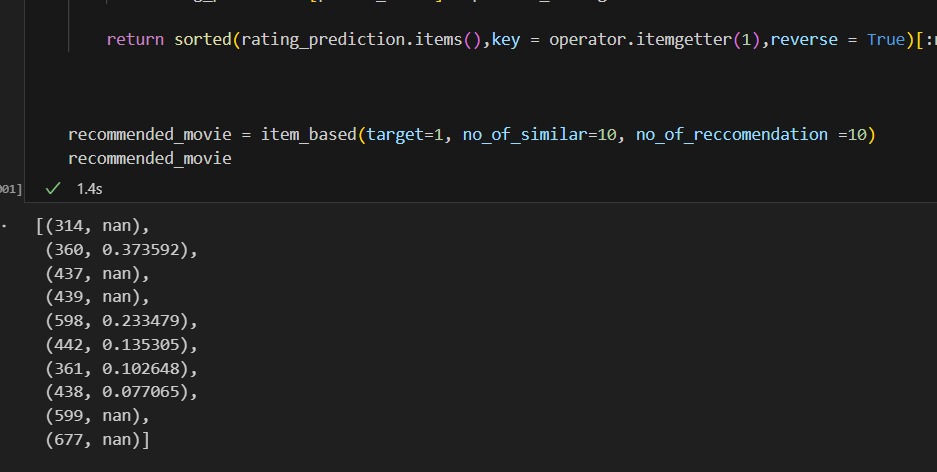
(Item-Item) Code Snippet 2:

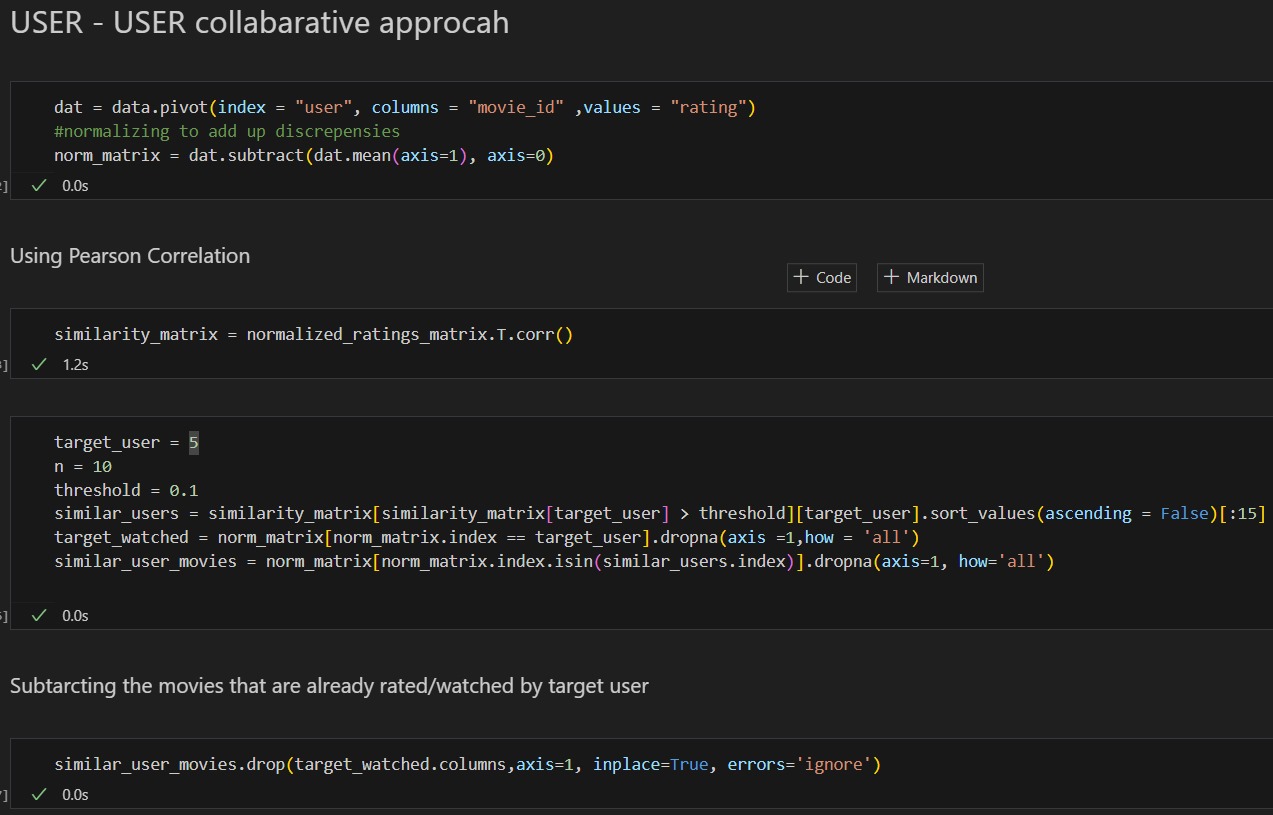


Output 1:

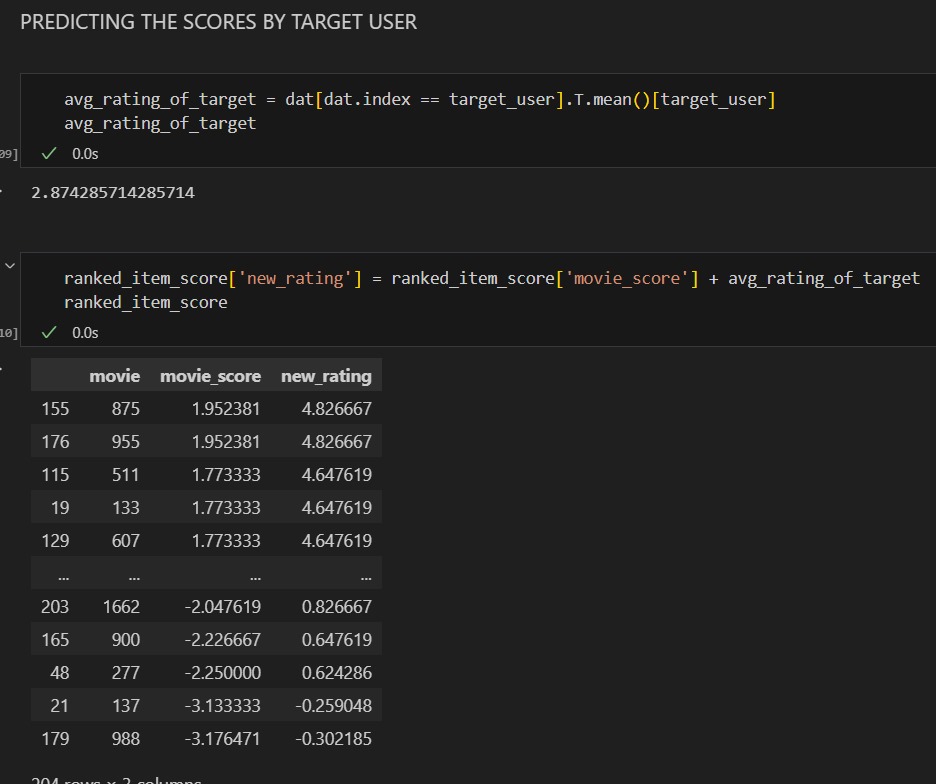


Output 2:



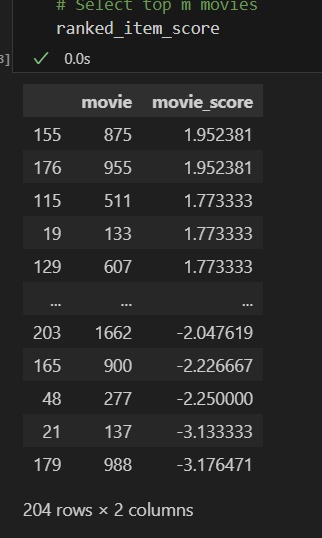
(User-User) Code Snippet :

Output 1:



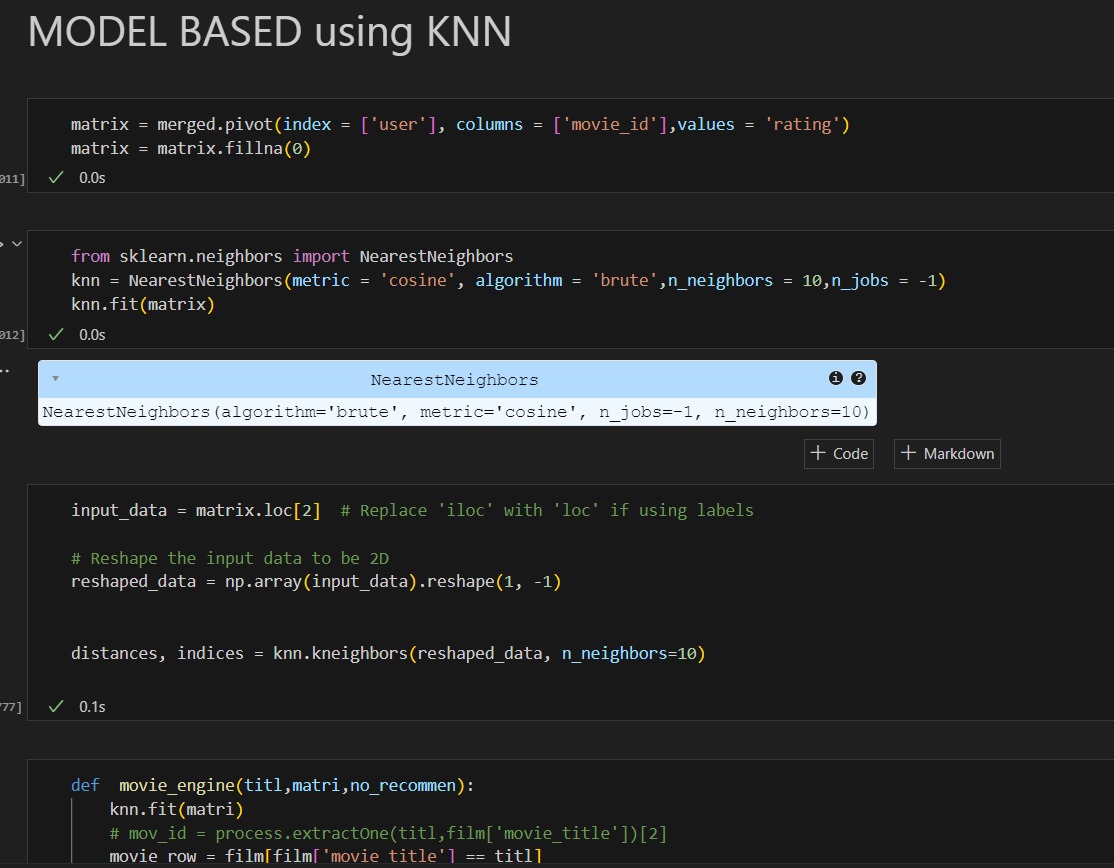
Extracts similarity from user-user matrix and recommends unwatched movies by the target user and predicts their rating.

Output 2:

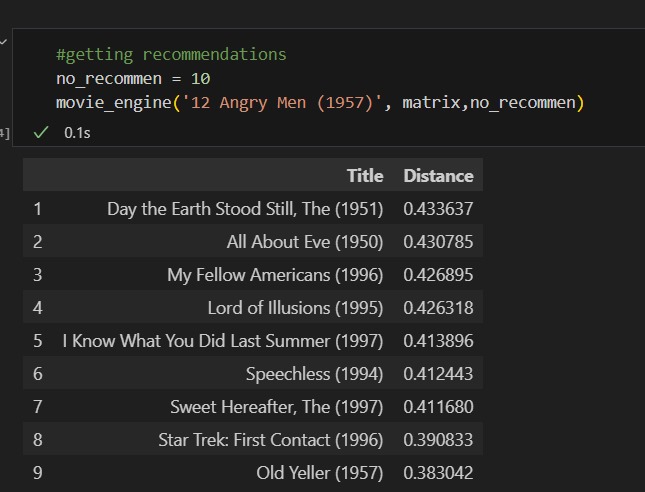


2.2) Model- based

Code Snippet:



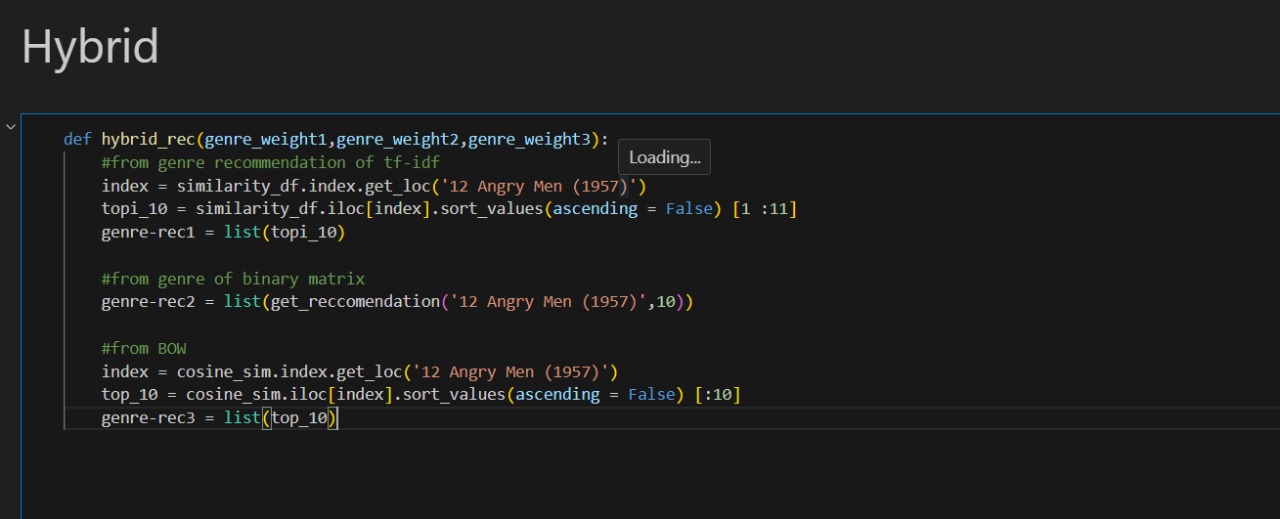
Output:



1. **Hybrid Based**

Providing weights of BOW, TF-IDF, binary matrix and calculating the weighted average and recommending the top-most movie from the content-based recommender algorithm system.

Code Snippet:



1. **BYOP (Our creative approach)**

Analyzes the rating given by the user and gives a statistical showing (matplotlib) their gender, profession, and age limit with number of rating rated and average value of the ratings.

Code Snippet:

