Report on the Project: Movie Recommendation Using Pearson Correlation

Alessio Marinucci, 546778 → repo GitHub: https://github.com/alemari7/AdvancedTopics

Introduction

This report describes the development process of a movie recommendation system based on Pearson correlation between users. The system was implemented using the Python programming language and key libraries such as Pandas and Math for data manipulation and statistical calculations.

Methodology

1. Data Preparation

Movie ratings were loaded from a CSV file using the Pandas library. After loading, unnecessary columns such as the timestamp of ratings were removed to simplify data analysis.

2. Calculation of Pearson Correlation

Using Pandas' data aggregation and manipulation capabilities, Pearson correlation between the user of interest (USER_1) and all other users in the dataset was calculated. This was done by iterating over all users and using Pandas' `corr()` method to compute the correlation between movie ratings.

3. Prediction of Movie Ratings

A prediction function was implemented to calculate the predicted ratings of movies for the user of interest. This function utilizes the Pearson correlation calculated in the previous phase and the ratings of other users to estimate the rating of a movie for the user of interest.

4. Movie Recommendation

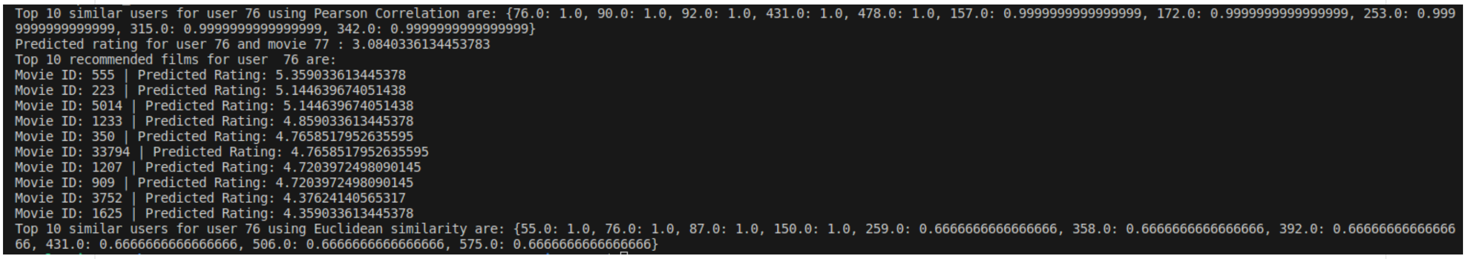
Finally, a function was developed to recommend movies to the user of interest. Using the predicted ratings calculated in the previous phase, the function recommends movies that have not yet been rated by the user of interest, sorting them based on the predicted rating.

5. Calculation of Euclidean Similarity

This function effectively finds users who have rated similar movies similarly to the target user. The Euclidean distance metric is used to quantify the dissimilarity between ratings, and the inverse of this distance serves as a measure of similarity between users.

Results

The recommendation system was able to successfully identify users similar to the user of interest and recommend movies to them based on the preferences of similar users. The recommendations were sorted based on the predicted rating, allowing users to view the most relevant movies at the top of the list.



```
Top 10 similar users for user 76 using Pearson Correlation are: {76.0: 1.0, 90.0: 1.0, 92.0: 1.0, 431.0: 1.0, 478.0: 1.0, 157.0: 0.9999999999999999, 172.0: 0.9999999999999999, 253.0: 0.999
9999999999999, 315.0: 0.9999999999999999, 342.0: 0.9999999999999999}
Predicted rating for user 76 and movie 77 : 3.0840336134453783
Top 10 recommended films for user  76 are:
Movie ID: 555 | Predicted Rating: 5.359033613445378
Movie ID: 223 | Predicted Rating: 5.144639674051438
Movie ID: 5014 | Predicted Rating: 5.144639674051438
Movie ID: 1233 | Predicted Rating: 4.859033613445378
Movie ID: 350 | Predicted Rating: 4.7658517952635595
Movie ID: 33794 | Predicted Rating: 4.7658517952635595
Movie ID: 1207 | Predicted Rating: 4.7203972498090145
Movie ID: 909 | Predicted Rating: 4.7203972498090145
Movie ID: 3752 | Predicted Rating: 4.37624140565317
Movie ID: 1625 | Predicted Rating: 4.359033613445378
Top 10 similar users for user 76 using Euclidean similarity are: {55.0: 1.0, 76.0: 1.0, 87.0: 1.0, 150.0: 1.0, 259.0: 0.6666666666666666, 358.0: 0.6666666666666666, 392.0: 0.66666666666666
66, 431.0: 0.6666666666666666, 506.0: 0.6666666666666666, 575.0: 0.6666666666666666}
```

Conclusions

In conclusion, the recommendation system implemented using Pearson correlation proved effective in providing personalized movie recommendations to users. The Pearson correlation function calculates the correlation coefficient between the user of interest and all other users in the dataset. This metric measures the linear relationship between two variables, in this case, the movie ratings of different users. Pandas' built-in corr() method facilitates the computation of Pearson correlation efficiently. By iterating over all users and computing their correlation with the user of interest, we obtain a similarity score that indicates how closely their movie preferences align. The recommendation function suggests movies to the user of interest based on their predicted ratings. It iterates over all movies in the dataset and identifies those that the user has not yet rated. For each unrated movie, the function calculates its predicted rating using the prediction function described above. The recommended movies are then sorted in descending order based on their predicted ratings, ensuring that the most relevant and potentially enjoyable movies appear at the top of the list. At the end, we calculate the top ten similar users using the Euclidean Similarity. However, further improvements could be made by exploring more advanced techniques and recommendation algorithms.