CMP2003 (DATA STRUCTURES & ALGORITHMS) PROJECT REPORT

- **Emirhan Kılıç (2202372)**
- ***Mehmet Deniz Özbilgiç (2102353)***
- ***Barış Arda Kılıç (2202942*)**
- ***Alperen Göktuğ Alev (2019318)***

1.Introduction

This project aims at implementing collaborative filtering techniques a in given order data to set estimate of users' ratings for movies. Designed in a User-Based Collaborative Filtering (UBCF) model and applied it to make recommendations for ratings of movies that have not been rated by the users. The project applies proper data structures and algorithms to deal with big data and to enhance the accuracy and efficiency of the system.

2. Data Structures and Algorithms

Data Structures:

- unordered_map<int, User>: Maps user IDs to their respective rating data.
- unordered_map<int, vector<int>>: Maps movie IDs to a list of users who rated the movie.
- vector<pair<int, double>>: Stores user similarities and their associated similarity scores.

Algorithms:

- Preprocessing: Unordered map used to map movies to users.
- Similarity Calculation (Pearson Correlation): Calculates centered ratings by subtracting the average rating of each user.
- Rating Prediction: Predicts ratings using the weighted ratings of similar users combined with the target user's average rating.
- Sorting and Selection: Applies partial sort to efficiently identify the top-k users from the similarity list.

3. Methodology

3.1 Algorithms

In this project, we implemented the User-Based Collaborative Filtering (UBCF) method. The algorithm is based on identifying the most similar users to predict the target user's rating of an unseen movie. Below are the steps we followed:

i. Preprocessing:

User ratings were stored in an unordered map data structure for efficient access.

Each user's average score was pre-calculated to allow for quick adjustments when calculating centered ratings.

Time complexity: O(n)

ii. Similarity Calculation:

We used Pearson Correlation for calculate similarities. It compares two users' ratings for the same movies and computes the correlation based on deviations from their mean ratings.

To optimize performance, similarity scores were cached in an unordered map so that repeated calculations were not performed for the same pair of users.

m = total number of user pairs
Time complexity: O(m^2)

iii. Rating Prediction:

For each target user and movie, the k most similar users who rated the movie were selected.

Ratings were predicted using a weighted average of deviations from the average ratings of similar users.

k = the number of most similar users
Time complexity: O(kn)

iv. Efficiency Improvements:

We used partial sort to select top k users efficiently.

Similarity threshold allowed us to focus on more similar users to reduce rmse.

4. Results

Our implementation (with k=120, threshold=0.18) achieved the following results:

RMSE: 0.9239
Running Time: 0.1633 seconds

Total Time Complexity: O(n + m^2 + kn)

4.1 Experimentation:

k = 150, threshold = 0.15
- RMSE: 0.9278
- Runnig time: 0.1937 seconds

k = 120, threshold = 0.18
- RMSE: 0.9239
- Running time: 0.1633 seconds

k = 80, threshold = 0.2
- RMSE: 0.9287
- Running time: 0.2281 seconds

5. Conclusion

In this project, we achieved the UBCF method for movie rating prediction as the model for our system. Our model gave an RMSE of 0.9239 and had a running time of 0.1633 seconds which fulfills the project requirements in terms of accuracy and efficiency. Through the application of efficient data structures and algorithms, it was possible to show how neighborhood-based collaborative filtering can be used for recommendation of items that may interest specific users.