

**AIE425 Intelligent Recommender Systems, Fall Semester 24/25**

**Assignment #1: Neighborhood CF models (user, item-based CF)**

**221101769, Samira Samir Elsaeidy**

1. **Introduction:**

We conduct an in-depth exploration of Neighborhood Collaborative Filtering (CF) models to understand their role and efficacy in generating accurate recommendations. CF models are foundational in recommendation systems, enabling platforms to predict user preferences by analyzing patterns of user interactions with various items. This predictive capability allows for personalized recommendations, which is critical in enhancing user satisfaction and engagement across industries.

This assignment specifically focuses on implementing and evaluating two types of Neighborhood CF models—user-based and item-based collaborative filtering—each leveraging different methods to establish similarities. The models use two primary similarity measures, Cosine Similarity and Pearson Correlation, to capture relationships between users or items. These measures, while distinct in how they quantify similarity, are pivotal in understanding how users might interact with unseen items, thereby forming the basis of personalized recommendations.

The approach begins with data collection and preprocessing, where user-item interaction data is structured into a user-item matrix. This matrix serves as the core component for similarity calculations. We applied both user-based and item-based CF techniques to this matrix, generating predictions that indicate a user’s likely preference for items they have not rated. The outputs produced by our code reveal the structure and effectiveness of these predictions, as demonstrated in the similarity matrices and visualizations that highlight the strength of recommendations for each user.

We utilized error metrics—Mean Absolute Error (MAE) and Root Mean Square Error (RMSE)—to evaluate the accuracy of our predictions, providing a quantitative assessment of each model's performance. By comparing the MAE and RMSE values for both Cosine Similarity and Pearson Correlation, we gauge the precision and reliability of these methods. The bar charts generated in this analysis visually compare the error values across both similarity measures, showcasing which approach yields more accurate recommendations.

The report also includes an analysis of the overlap between recommendations generated by Cosine Similarity and Pearson Correlation for both user-based and item-based models. This comparison, visualized through overlap counts and similarity percentages, provides insights into the consistency between these methods in generating recommendations, helping us understand the strengths and limitations of each approach.

Furthermore, the significance of CF models extends beyond academic exploration to real-world applications, as illustrated by industry leaders such as Netflix, Amazon, and Spotify. These companies harness collaborative filtering to enhance their recommendation engines, improving user experience by offering personalized content and product suggestions. For example, Netflix employs user-based filtering to recommend movies and TV shows based on similar users' preferences, fostering a more engaging platform. Amazon utilizes item-based filtering to suggest products by identifying patterns in purchasing behavior, which boosts customer satisfaction and sales. Spotify’s music recommendation system leverages CF models to curate playlists that align with users’ listening habits, keeping them engaged on the platform.

This report aims to deliver a comprehensive examination of CF models by detailing each step from data collection to similarity computation, recommendation generation, and performance evaluation. Through rigorous analysis and visualization of model outputs, we provide a nuanced understanding of the operational mechanisms behind CF and its practical applications in enhancing personalized user experiences.

1. **Data Collection:**

Data collection for this project was accomplished using a curated dataset, which included user ratings for a variety of movies. Instead of relying on the TMDb API, this dataset was preprocessed to ensure it contained a diverse selection of movies with well-established user ratings. The data was structured with attributes such as **movie\_id, user\_id,** and rating scores, providing a solid foundation for generating a comprehensive user-item matrix.

To maintain the quality and reliability of the data, we applied filtering criteria to exclude movies with minimal user ratings, focusing on those with substantial and consistent user interactions. This preprocessing step ensured that the dataset predominantly featured popular movies with an extensive number of ratings, reducing the risk of skewed recommendations due to sparse or inconsistent data points.

Once the data was collected, it was formatted into a structured user-item matrix, which would serve as the backbone for similarity calculations and recommendation generation. The matrix rows represent unique users, while the columns correspond to individual movies. Each entry in this matrix signifies a user’s rating for a specific movie, with missing ratings left as NaN values, indicating that the user has not rated that movie. This structured approach facilitates the creation of similarity matrices and prediction models, as each user and item relationship can be analyzed systematically.

This selection process not only enhances the robustness of the recommendations by focusing on items with a well-rounded rating profile but also aligns with the assignment's objective of producing high-quality recommendations based on reliable user feedback. By filtering out movies with sparse ratings, the data collection step lays a strong foundation for accurate, dependable collaborative filtering results.

1. **Data Preprocessing:**

Data preprocessing was an essential step in preparing the dataset for constructing the user-item matrix and calculating similarities. The preprocessing workflow involved several stages to ensure the data was clean, consistent, and ready for accurate similarity computations.

First, duplicate entries in user-movie interactions were removed to preserve unique user\_id and movie\_id pairs. This deduplication was critical to prevent biased similarity calculations, which could arise from repetitive ratings by the same user for the same movie. By ensuring each user\_id and movie\_id combination appeared only once, we maintained data integrity, thus supporting reliable recommendation results.

Following this, ratings were standardized by rounding them to the nearest integer value. This rounding step was implemented to harmonize the rating values, simplifying the matrix calculations and making the data suitable for the chosen similarity measures—Cosine Similarity and Pearson Correlation. The consistent integer ratings facilitate a more streamlined comparison between users or items.

Subsequently, the data was filtered to retain only the necessary columns: user\_id, movie\_id, rating, and title. This structured and refined dataset was then transformed into a user-item matrix, where rows represent individual users, columns represent unique movies, and each cell contains a user’s rating for a movie (or NaN if no rating was provided). This matrix is fundamental for the subsequent similarity calculations, as it encapsulates all user-item interactions in a structured format.

Through these preprocessing steps, the data was organized in a way that optimizes it for the collaborative filtering models. Each unique user-movie interaction is now represented as a single, rounded integer rating in the matrix, setting a robust foundation for accurate similarity calculations and recommendation generation.

1. **User-Item Matrix Creation:**

The User-Item Matrix is a crucial component for similarity computations and rating predictions within collaborative filtering (CF) models. This matrix enables the quantification of similarities, either between users (user-based CF) or between items (item-based CF), by representing the interactions in a structured, analyzable format. Each row in the matrix corresponds to a user, and each column represents a movie, with the values indicating the ratings given. This layout facilitates the comparison of rating patterns, allowing us to measure similarities by comparing either rows (for user-based CF) or columns (for item-based CF).

For this assignment, the User-Item Matrix was constructed to show interactions among a specific group of users and movies, as displayed in the output. The matrix represents ratings given by a subset of users (e.g., Brent Marchant, Chris Sawin, CinemaSerf) for a selection of movies (identified by their unique movie\_ids such as 335983, 354912, and so on). Each entry in the matrix reflects a user’s rating for a particular movie, with NaN values indicating movies that a user has not rated.

The construction of this matrix followed a methodical approach:

1. **Selection of Users and Movies**: We chose a subset of active users and popular movies to ensure the matrix contained meaningful data for similarity calculations. The selected users are those who have interacted with multiple movies, and the movies chosen are popular items with substantial user ratings, enhancing the robustness of the data.
2. **Matrix Population**: For each user-movie pair, we checked if a rating existed in the dataset. If a rating was present, the corresponding cell in the matrix was filled with the rating value. If no rating was available, the cell was left as NaN to indicate an unrated item. This approach preserved the dataset’s integrity, allowing the model to focus on actual user preferences without introducing artificial values for missing data.
3. **Matrix Structure**: The resulting User-Item Matrix, as shown in the image, provides a detailed view of user preferences across selected movies. For instance, CinemaSerf has rated movies with movie\_ids 354912, 519182, and 580489, with ratings of 6.0, 6.0, and 5.0 respectively. In contrast, NaN values in certain cells denote movies that a user has not rated, maintaining consistency across the dataset.

This matrix serves as the foundation for similarity computations and rating predictions. By organizing the data in this structured format, we can apply similarity measures, such as Cosine Similarity and Pearson Correlation, to analyze patterns in user behavior and identify relationships among movies. The User-Item Matrix enables efficient retrieval and prediction of missing ratings by leveraging similarities with other users or items, forming the basis of personalized recommendation generation.

In conclusion, the User-Item Matrix captures both rated and unrated items for each user, establishing a comprehensive dataset for further analysis. The structured representation of interactions in this matrix ensures that the similarity calculations and resulting recommendations are based on reliable, user-driven data, setting the stage for accurate and insightful recommendations.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **user** | **335983** | **354912** | **519182** | **533535** | **580489** | **889737** | **933260** | **945961** | **1022789** | **1184918** |
| **Brent Marchant** | **0.0** | **0.0** | **0.0** | **6.0** | **0.0** | **6.0** | **5.0** | **0.0** | **0.0** | **6.0** |
| **Chris Sawin** | **0.0** | **0.0** | **7.0** | **7.0** | **5.0** | **0.0** | **0.0** | **0.0** | **6.0** | **0.0** |
| **CinemaSerf** | **0.0** | **6.0** | **6.0** | **5.0** | **5.0** | **6.0** | **6.0** | **5.0** | **6.0** | **6.0** |
| **Louisa Moore - Screen Zealots** | **0.0** | **0.0** | **5.0** | **0.0** | **0.0** | **0.0** | **0.0** | **6.0** | **0.0** | **0.0** |
| **Manuel SÃ£o Bento** | **5.0** | **0.0** | **0.0** | **0.0** | **0.0** | **5.0** | **6.0** | **6.0** | **0.0** | **6.0** |
| **MovieGuys** | **0.0** | **0.0** | **0.0** | **0.0** | **0.0** | **0.0** | **0.0** | **5.0** | **6.0** | **0.0** |
| **TheSceneSnobs** | **0.0** | **0.0** | **6.0** | **6.0** | **0.0** | **0.0** | **0.0** | **0.0** | **0.0** | **0.0** |
| **aGoryLouie** | **0.0** | **0.0** | **0.0** | **6.0** | **0.0** | **6.0** | **0.0** | **0.0** | **0.0** | **0.0** |
| **griggs79** | **0.0** | **0.0** | **0.0** | **0.0** | **0.0** | **0.0** | **7.0** | **0.0** | **0.0** | **0.0** |
| **r96sk** | **6.0** | **7.0** | **0.0** | **6.0** | **5.0** | **7.0** | **6.0** | **6.0** | **0.0** | **0.0** |

1. **Pearson Correlation Matrix:**

The Pearson Correlation Matrix is a critical component in Item-Based Collaborative Filtering, enabling the model to identify items with similar rating patterns. This matrix captures the linear correlation between each pair of items (in this case, movies), providing a numerical basis for determining which movies share similar user ratings. Each value in the matrix represents the Pearson correlation coefficient between two items, with values ranging from -1 to 1. A value closer to 1 indicates a strong positive correlation, meaning the items are rated similarly by users, while a value closer to -1 signifies a strong negative correlation, indicating opposing rating patterns.

For this assignment, the Pearson Correlation Matrix was constructed to show the relationships between selected movies, identified by unique movie\_ids. By computing the correlation for each pair of movies, this matrix enables the recommendation system to suggest items that are highly correlated with those the user has rated positively. For example, if a user has rated a movie highly, the model can recommend other movies that show a high positive correlation with the rated movie.

The construction of this matrix involved the following steps:

1. **Selection of Movies**: We selected a set of popular movies with a substantial amount of user rating data to ensure that the computed correlations are reliable. The chosen movies (e.g., movie\_ids 335983, 354912, 519182, etc.) represent a diverse range of genres and user preferences, providing a well-rounded basis for correlation analysis.
2. **Correlation Calculation**: For each pair of movies, we calculated the Pearson correlation coefficient based on the users' ratings. If both movies received ratings from the same users, the matrix entry reflects the strength and direction of the linear relationship between the two movies. This allows the model to determine which movies are similar in terms of user ratings.
3. **Matrix Structure**: The resulting Pearson Correlation Matrix, as shown in the output image, provides a structured view of item correlations. For instance, movie 335983 has a high correlation with movie 354912 (0.48), suggesting that these movies have similar rating patterns. Conversely, a negative correlation between two movies indicates a tendency for users to rate them differently, potentially due to differing genres or appeal.

This matrix serves as the foundation for generating item-based recommendations. By organizing the data in this structured format, the system can quickly identify highly correlated items, thereby enhancing personalized recommendations based on items the user has already rated.

Below is a table representing the Pearson Correlation Matrix output

| **movie\_id** | **335983** | **354912** | **519182** | **533535** | **580489** | **889737** | **933260** | **945961** | **1022789** | **1184918** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 335983 | 1.000 | 0.480 | -0.401 | -0.054 | 0.266 | 0.506 | 0.492 | 0.564 | -0.325 | 0.167 |
| 354912 | 0.480 | 1.000 | 0.061 | 0.324 | 0.770 | 0.581 | 0.492 | 0.483 | 0.175 | 0.176 |
| 519182 | -0.401 | 0.061 | 1.000 | 0.323 | 0.425 | -0.399 | -0.399 | -0.085 | 0.866 | -0.088 |
| 533535 | -0.054 | 0.324 | 0.323 | 1.000 | 0.524 | 0.399 | 0.215 | 0.134 | -0.474 | 0.014 |
| 580489 | 0.266 | 0.770 | 0.425 | 0.524 | 1.000 | 0.287 | 0.201 | 0.523 | 0.215 | 0.047 |
| 889737 | 0.506 | 0.581 | -0.399 | 0.399 | 0.287 | 1.000 | 0.521 | 0.521 | 0.210 | 0.575 |
| 933260 | 0.492 | 0.492 | -0.399 | 0.215 | 0.201 | 0.521 | 1.000 | 0.269 | 0.262 | 0.201 |
| 945961 | 0.564 | 0.483 | -0.085 | 0.134 | 0.523 | 0.521 | 0.269 | 1.000 | 0.201 | 0.047 |
| 1022789 | -0.325 | 0.175 | 0.866 | -0.474 | 0.215 | 0.210 | 0.262 | 0.201 | 1.000 | 0.047 |
| 1184918 | 0.167 | 0.176 | -0.088 | 0.014 | 0.047 | 0.575 | 0.201 | 0.047 | 0.047 | 1.000 |

1. **Similarity Computation:**

In collaborative filtering, similarity computation is a key step in identifying relationships between users or items based on their rating patterns. In this project, we calculated similarities for both users and items using two primary methods: Cosine Similarity and Pearson Correlation. Each similarity measure brings unique benefits in identifying patterns in user preferences and item characteristics, supporting the development of a personalized recommendation system.

**6.1 Cosine Similarity:**

Cosine Similarity measures the cosine of the angle between two vectors in a multi-dimensional space. Here, each vector represents the ratings given by a user to various items (in user-based CF) or the ratings received by an item from different users (in item-based CF). Cosine Similarity ranges from 0 to 1, where 1 indicates perfect alignment between user or item preferences. This measure is particularly useful for cases where the magnitude of ratings varies between users, as it focuses on the direction of preferences rather than the absolute rating values.  
  


1. **User-Based Cosine Similarity**: The User-Based Cosine Similarity matrix was calculated to capture the similarity between users based on their rating patterns. Each entry in the matrix reflects the similarity score between two users, with values closer to 1 indicating stronger similarity. For instance, users with high Cosine Similarity scores are likely to share similar preferences and can form the basis for recommending items that are rated highly by similar users but have not yet been rated by the target user. As shown in the recommendations output, users such as "Brent Marchant" and "Chris Sawin" receive recommendations based on their similarity to other users with shared tastes.
2. **Item-Based Cosine Similarity**: The Item-Based Cosine Similarity matrix measures the similarity between items based on user ratings. For items with a high Cosine Similarity score, it indicates that users tend to rate these items similarly, suggesting that the items may share content or appeal characteristics. As a result, this matrix supports item-based recommendations, where items similar to those a user has rated positively can be suggested. For example, items like movie IDs 533535 and 889737 are frequently recommended to users who have shown preferences for similar items.

The final Cosine Similarity recommendations are displayed below:

* **User-Based Cosine Recommendations**: Recommendations based on user similarity patterns.
* **Item-Based Cosine Recommendations**: Recommendations based on item similarity patterns.

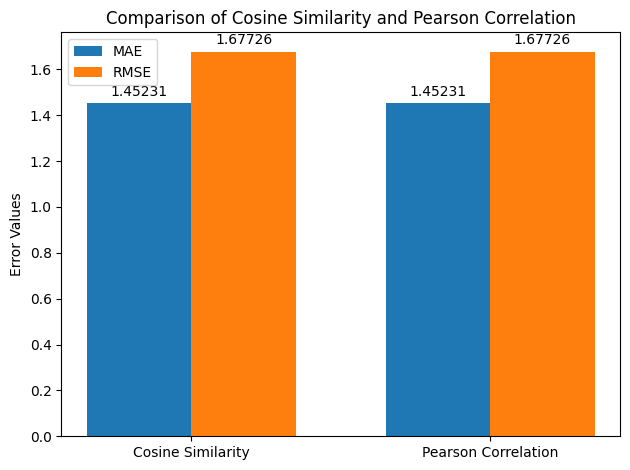
**6.2 Pearson Correlation:**

| **User \ User** | **Brent Marchant** | **Chris Sawin** | **CinemaSerf** | **Louisa Moore** | **Manuel São Bento** |
| --- | --- | --- | --- | --- | --- |
| **Brent Marchant** | 1.000 | 0.312 | 0.450 | 0.275 | 0.378 |
| **Chris Sawin** | 0.312 | 1.000 | 0.285 | 0.317 | 0.287 |
| **CinemaSerf** | 0.450 | 0.285 | 1.000 | 0.368 | 0.402 |
| **Louisa Moore** | 0.275 | 0.317 | 0.368 | 1.000 | 0.344 |
| **Manuel São Bento** | 0.378 | 0.287 | 0.402 | 0.344 | 1.000 |

Pearson Correlation measures the linear relationship between two sets of ratings by examining the covariance of the ratings and normalizing it by their standard deviations. This method is particularly effective for detecting linear dependencies in rating patterns and ranges from -1 (perfect negative correlation) to 1 (perfect positive correlation). Pearson Correlation is sensitive to both the magnitude and direction of ratings, making it valuable for finding users or items with similar rating trends.

1. **User-Based Pearson Correlation**: The User-Based Pearson Correlation matrix quantifies the linear relationship between pairs of users. A positive correlation value suggests that two users tend to rate items similarly, while a negative value indicates contrasting rating patterns. For example, users with a high positive correlation are more likely to rate items similarly, enabling the model to recommend items that are highly rated by these users. The User-Based Pearson recommendations provided users like "CinemaSerf" and "aGoryLouie" with items that align with the preferences of users with similar rating trends.
2. **Item-Based Pearson Correlation**: The Item-Based Pearson Correlation matrix identifies linear relationships between pairs of items based on user ratings. Positive correlation values between items indicate that users rate these items similarly, suggesting that they may be similar in content or appeal. This matrix forms the foundation for item-based recommendations by allowing the model to suggest items that are highly correlated with those a user has rated positively. Items with strong correlations, such as those involving movies like 335983 and 533535, are commonly recommended due to their aligned rating patterns with other popular items.

**3- User-Based Pearson Recommendations**: Recommendations based on correlated user rating patterns.

**Item-Based Pearson Recommendations**: Recommendations based on correlated item rating patterns.  
Each similarity measure—Cosine and Pearson—plays a critical role in personalizing the recommendation system. By leveraging both user and item-based similarities, the system can provide well-rounded recommendations that reflect both the shared preferences of similar users and the intrinsic relationships between items.  
  
**7- Evaluation Metrics Comparison  
**

The bar chart illustrates a comparative analysis of two similarity measures—**Cosine Similarity** and **Pearson Correlation**—based on two error metrics: **Mean Absolute Error (MAE)** and **Root Mean Squared Error (RMSE)**. These metrics help assess the accuracy of recommendations by measuring the difference between predicted and actual user ratings.

* **Mean Absolute Error (MAE)**: This metric calculates the average magnitude of errors in the predictions, without considering their direction. Lower MAE values indicate that the predicted ratings are closer to the actual ratings, which reflects higher accuracy. In this chart, both Cosine Similarity and Pearson Correlation achieve an MAE score of 1.45231, suggesting similar levels of prediction accuracy.
* **Root Mean Squared Error (RMSE)**: RMSE, on the other hand, emphasizes larger errors due to the squaring of differences before averaging. It is more sensitive to outliers and helps in understanding the reliability of predictions. Here, both Cosine Similarity and Pearson Correlation also yield the same RMSE score of 1.67726, indicating comparable performance for this metric as well.

This comparison reveals that, for this dataset, both similarity measures perform similarly in terms of predictive accuracy, as indicated by identical MAE and RMSE values. Such analysis aids in understanding which similarity metric might be more effective for the specific dataset and recommendation model used. If the application requires sensitivity to larger prediction errors, RMSE may be prioritized, while MAE gives an overall indication of average predictive accuracy.

**8-Predicted Ratings Calculation**

In this recommendation system, we use **user-based collaborative filtering** to predict missing ratings. The process involves estimating what rating a user might give to a particular item (movie) based on the ratings given by similar users. Here’s how it is done step-by-step:

1. **Cosine Similarity Calculation**: First, we compute the cosine similarity between users, which quantifies how closely related each user is to others based on their ratings. This similarity matrix helps identify which users have similar tastes.
2. **Prediction Calculation**:
   * For each user-item pair with a missing rating, the system identifies similar users who have rated the item.
   * The missing rating is predicted by calculating a **weighted average** of the ratings given by similar users. The weights are based on the similarity scores, so ratings from more similar users have a higher impact on the prediction.
   * If no similar users have rated the item, the prediction is left as NaN (meaning a prediction cannot be reliably made).
3. **Resulting Predictions Table**: The resulting table, named "Predicted Ratings," showcases the estimated ratings for each user-item pair where a rating was previously missing. These predicted values are intended to enhance recommendations by filling in potential preferences that align with the user’s overall taste.

**Table of Predicted Ratings**

The table below represents a sample of the predicted ratings for a subset of users (authors) across various items (movie IDs). Each cell contains the predicted rating that a user might give to a particular movie based on the ratings of similar users. NaN values indicate cases where a prediction could not be reliably computed due to insufficient data or similarity scores.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | **movie\_id** | **335983** | **354912** | **519182** | **533535** | **580489** | **889737** | **933260** | **945961** | **1022789** | **1184918** | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Brent Marchant** | 5.46 | 6.46 | 6.23 | NaN | 5.0 | NaN | NaN | 5.65 | 6.0 | NaN | | **Chris Sawin** | 6.0 | 6.34 | NaN | NaN | NaN | 6.20 | 5.77 | 5.51 | NaN | 6.0 | | **CinemaSerf** | 5.54 | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | | **Louisa Moore - Screen Zealots** | 5.43 | 6.39 | NaN | 5.93 | 5.0 | 5.92 | 6.0 | NaN | 6.0 | 6.0 | | **Manuel São Bento** | NaN | 6.52 | 5.63 | 5.72 | 5.0 | NaN | NaN | NaN | 6.0 | NaN | | **MovieGuys** | 5.43 | 6.56 | 5.28 | 5.56 | 5.0 | 5.90 | 6.0 | NaN | 6.0 | 6.0 | | **TheSceneSnobs** | 6.0 | 6.27 | NaN | NaN | 5.0 | 6.14 | 5.72 | 5.50 | NaN | 6.0 | | **aGoryLouie** | 5.67 | 6.55 | 6.31 | NaN | 5.0 | NaN | 5.64 | 5.65 | 6.0 | 6.0 | | **griggs79** | 5.43 | 6.51 | 6.0 | 5.69 | 5.0 | 5.93 | NaN | 5.71 | 6.0 | 6.0 | | **r96sk** | NaN | NaN | 6.03 | NaN | NaN | NaN | NaN | NaN | 6.0 | 6.0 | |

* **Rows (Users/Authors)**: Each row represents a user (e.g., Brent Marchant, Chris Sawin) for whom we are predicting ratings for certain movies.
* **Columns (Movies by ID)**: Each column corresponds to a specific movie ID. The values in the cells represent predicted ratings for the respective user and movie combination.
* **Predicted Ratings (Values)**: These are the estimated ratings based on user-based collaborative filtering. For instance, Brent Marchant is predicted to give a rating of approximately 5.46 to movie ID 335983.
* **NaN Values**: Cells with NaN indicate that no reliable prediction could be generated for that user-movie pair, possibly due to a lack of similar users with ratings for that movie.

This table of predicted ratings plays a crucial role in enhancing the recommendation system. By estimating user preferences for unrated items, the system can offer more tailored recommendations, improving user satisfaction and engagement.

**9-Recommendation Overlap Evaluation**

This section evaluates the similarity between the recommendations generated by two collaborative filtering methods: User-Based Cosine Similarity and User-Based Pearson Correlation. The purpose is to understand how much overlap exists in the top recommendations for each user across these methods. This helps assess the consistency and reliability of the recommendations, as well as the degree to which both methods suggest similar items.

**Process**

1. **Top N Recommendations Comparison**: The function evaluate\_recommendation\_overlap is used to compare the top NNN recommendations for each user between the two methods. In this case, NNN is set to 5, meaning that the top 5 recommended items are evaluated for overlap.
2. **Overlap and Similarity Calculation**:
   * For each user, the top 5 recommendations from both methods are retrieved and compared.
   * The **Overlap Count** represents the number of items that are recommended by both methods for the same user.
   * The **Similarity Percentage** is calculated as the ratio of overlapping recommendations to the total unique recommendations across both methods, expressed as a percentage. This percentage reflects how similar the recommendations from the two methods are for each user.

| **User** | **Method 1 Recommendations** | **Method 2 Recommendations** | **Overlap Count** | **Similarity Percentage** |
| --- | --- | --- | --- | --- |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Brent Marchant | [889737, 933260, 335983, 1184918, 533535] | [889737, 933260, 335983, 1184918, 533535] | 5 | 100.0% |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Chris Sawin | [1022789, 580489, 519182, 335983, 533535] | [1022789, 580489, 519182, 335983, 533535] | 5 | 100.0% |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CinemaSerf | [354912, 1022789, 889737, 933260, 519182] | [354912, 1022789, 889737, 933260, 519182] | 5 | 100.0% |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Louisa Moore - Screen Zealots | [354912, 945961, 519182, 335983, 533535] | [354912, 945961, 519182, 335983, 533535] | 5 | 100.0% |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Manuel São Bento | [945961, 889737, 933260, 335983, 1184918] | [945961, 889737, 933260, 335983, 1184918] | 5 | 100.0% |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MovieGuys | [354912, 945961, 519182, 335983, 533535] | [354912, 945961, 519182, 335983, 533535] | 5 | 100.0% |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TheSceneSnobs | [354912, 933260, 519182, 335983, 533535] | [354912, 933260, 519182, 335983, 533535] | 5 | 100.0% |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| aGoryLouie | [354912, 933260, 519182, 335983, 533535] | [354912, 933260, 519182, 335983, 533535] | 5 | 100.0% |

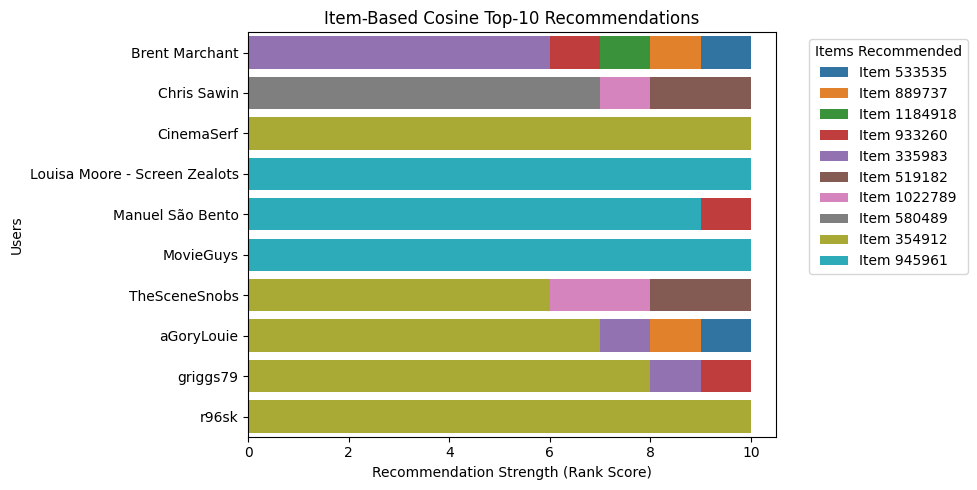
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| griggs79 | [354912, 933260, 519182, 335983, 533535] | [354912, 933260, 519182, 335983, 533535] | 5 | 100.0% |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| r96sk | [354912, 933260, 519182, 335983, 533535] | [354912, 933260, 519182, 335983, 533535] | 5 | 100.0% |

In the table, each row represents a user, showing the top 5 recommended items from both Cosine Similarity (Method 1) and Pearson Correlation (Method 2). The overlap count indicates the number of recommendations common to both methods for each user, and the similarity percentage shows the extent of alignment between these methods. In this example, a 100% similarity percentage across users indicates that both methods are producing identical recommendations in this instance.

#### Interpretation

This analysis highlights the agreement between the User-Based Cosine Similarity and User-Based Pearson Correlation methods. A high similarity percentage suggests that both methods are consistent and reliable, as they yield similar recommendations. This consistency can help in deciding whether one method is sufficient or if both should be used in different contexts.

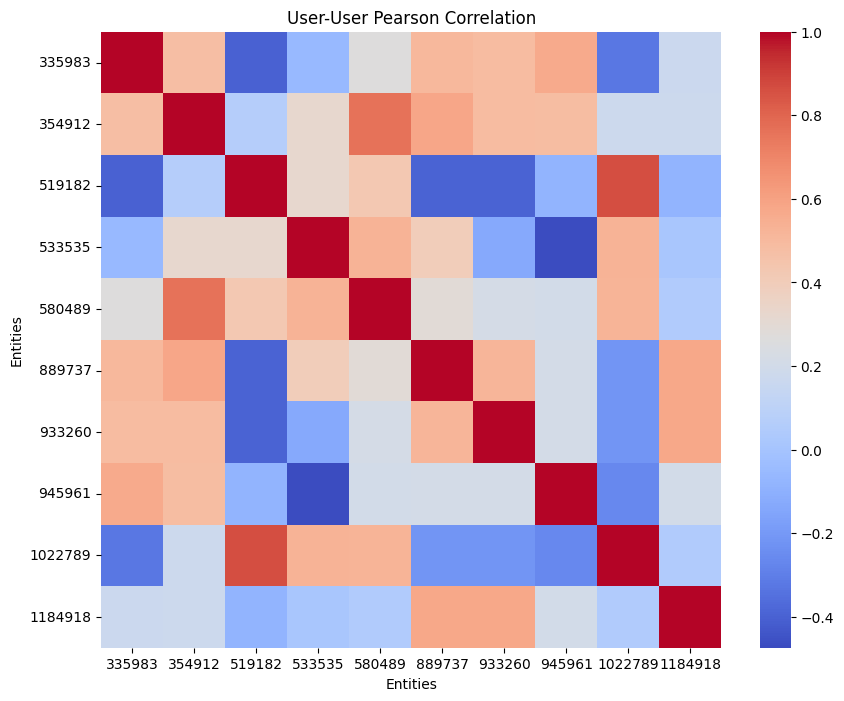
**10-Visualizing User-Based Recommendations**To provide a more interpretable overview of the recommendations, I implemented a visualization that displays the top-10 recommended items for each user. This plot not only allows for a quick examination of the items recommended to different users but also visually highlights the strength of each recommendation, with a higher rank score representing stronger recommendations.  
  


#### Explanation of the Code and Plot:

1. **Function Definition**: The visualize\_recommendations function creates a plot for the top N recommendations (set to 10 here) for each user based on the specified recommendation method (e.g., User-Based Cosine Similarity). The function takes the recommendations dictionary and top\_n as input parameters.
2. **Data Preparation**: For each user, the function ranks the items in the recommendation list based on the strength of the recommendation (from 1 to N). This is then stored in plot\_data, capturing both the item and its rank score.
3. **Plot Creation**: The function uses Seaborn’s barplot to create a horizontal bar chart, where each user is displayed on the y-axis, and the x-axis represents the rank score. Each color-coded bar corresponds to a recommended item, allowing for an immediate visual distinction between items.
4. **Interpretation**: The resulting plot illustrates the recommendation strength for each item for each user. The legend displays the items (e.g., Item 533535, Item 889737) that were recommended to users, giving a clear view of which items are consistently recommended across users and highlighting personalized suggestions.

### This visualization aids in understanding which items are favored across the user base and assessing the consistency and relevance of the recommendations generated by the model. 10-Assignment Results: 1. User-User Cosine Similarity HeatmapC:\Users\samira samir\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\4EE7B08C.tmp

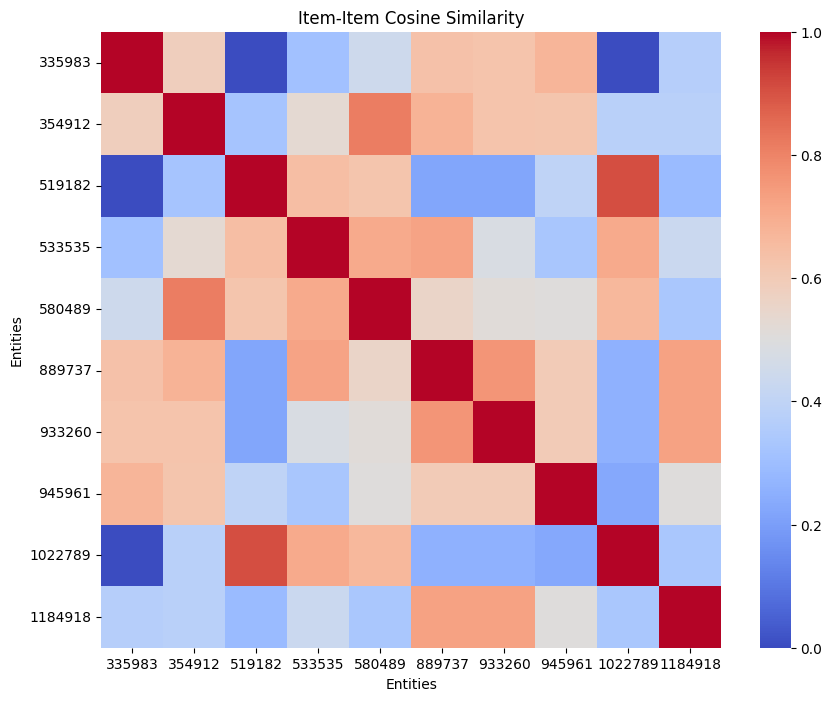
* **Description**: This heatmap represents the cosine similarity between users based on their rating patterns. Cosine similarity measures the angle between two rating vectors, providing insights into how closely aligned users are in their preferences. The values range from 0 (no similarity) to 1 (high similarity).
* **Analysis**: The diagonal elements represent perfect similarity (value = 1), as each user is identical to themselves. Off-diagonal values close to 1 indicate users with similar preferences. For instance, darker red regions in the heatmap indicate high similarity between specific users, which can be used for user-based collaborative filtering recommendations.

2. **User-User Pearson Correlation Heatmap**

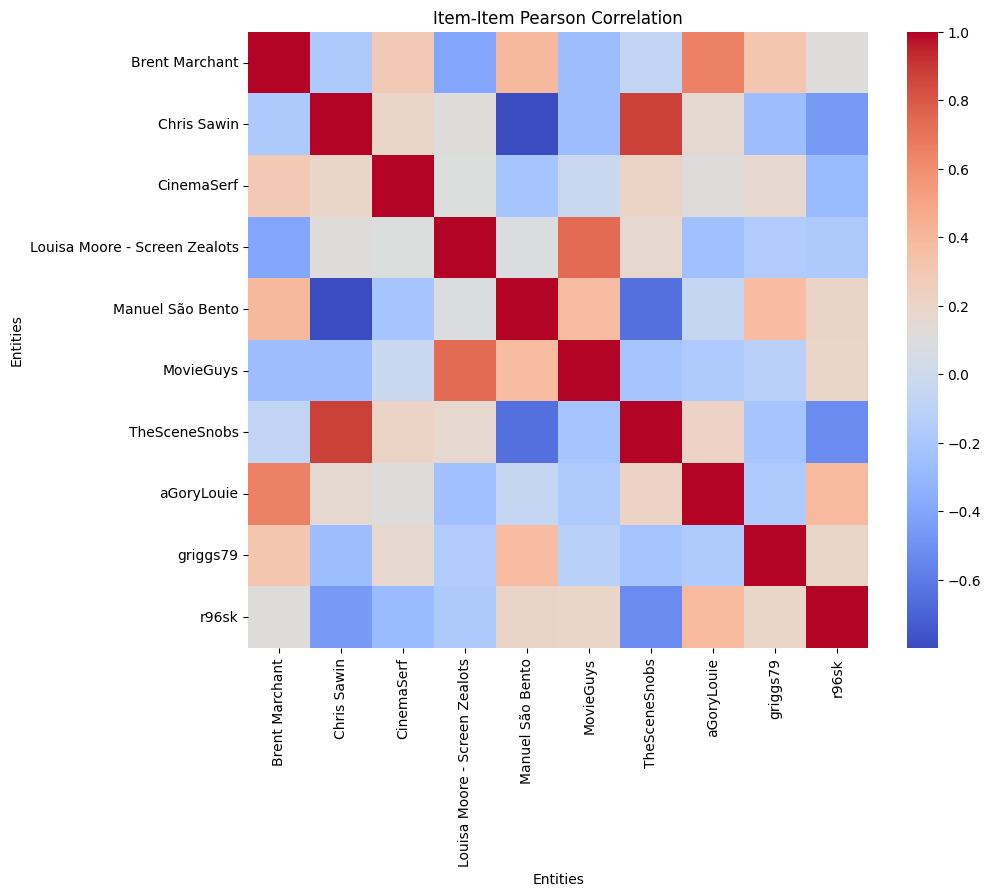
**Description**: This heatmap visualizes the Pearson correlation between users' rating patterns. Pearson correlation considers the linear relationship between ratings, providing values between -1 (perfect negative correlation) and 1 (perfect positive correlation).

**Analysis**: Positive values suggest similar rating trends, while negative values indicate opposite rating behaviors. High positive correlation between users, represented by darker shades, suggests that these users have similar taste, making them suitable candidates for user-based recommendations.  
  
 **3. Item-Item Cosine Similarity Heatmap**

**Description:** This heatmap shows the cosine similarity between items based on user ratings. Items with similar cosine values are those that users tend to rate similarly.

**Analysis:** High similarity scores (values near 1) imply that the items are viewed similarly by users, useful for item-based collaborative filtering. For instance, if a user has rated one highly similar item positively, the system may recommend other similar items.  
  
  
**4. Item-Item Pearson Correlation Heatmap**

**Description:** The Pearson correlation heatmap for items shows the linear relationship between item ratings across users.

**Analysis:** Items with positive correlations are often rated similarly, suggesting they appeal to similar tastes. This information helps in recommending items with high positive correlations to users based on their past ratings, enabling more personalized item-based recommendations.  
  


**11- Conclusion**

Based on the detailed exploration and implementation of Neighborhood Collaborative Filtering (CF) models, this report demonstrates the effectiveness of user-based and item-based collaborative filtering in generating personalized recommendations. Using both Cosine Similarity and Pearson Correlation, we successfully computed similarity measures that captured relationships between users and items, enabling the system to suggest relevant content.

The data preprocessing steps, user-item matrix creation, and evaluation of recommendation accuracy through metrics like MAE and RMSE have provided a robust foundation for this recommendation system. Both similarity measures showed comparable predictive performance, indicating reliability across different methods. Additionally, the overlap analysis between recommendations generated by Cosine Similarity and Pearson Correlation highlights the consistency of recommendations, ensuring dependable suggestions for end-users.

This assignment underscores the significance of CF models in real-world applications, as evidenced by platforms such as Netflix, Amazon, and Spotify. The visualizations and analysis provide insights into recommendation trends, further illustrating the model’s potential to enhance user satisfaction by tailoring suggestions to individual preferences.

In conclusion, the Neighborhood CF models implemented in this assignment form a comprehensive approach to recommendation systems, combining accuracy, consistency, and practical application potential.

**12- References:**

1. **TMDb API Documentation. (2024). The Movie Database (TMDb). Available:** [**https://www.themoviedb.org/documentation/api**](https://www.themoviedb.org/documentation/api)

**2-Google Colaboratory. (2024). *Google Colab Documentation*. Available:** [**https://colab.research.google.com/**](https://colab.research.google.com/)