

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

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

**221100583, Mohammed Haitham Mohammed**

**Table of contents:**

|  |  |  |
| --- | --- | --- |
| **Section** | **Title** | **Page** |
| **1** | **Introduction** | **1** |
| **2** | **Data Collection** | **1** |
| **3** | **Data Preprocessing** | **2** |
| **4** | **User-Item Matrix Creation** | **2** |
| **5** | **Background on User-Based and Item-Based CF** | **4** |
| **6** | **Similarity Computation** | **4** |
| **6.1** | **Cosine Similarity** | **4** |
| **6.1.1** | **Example Calculation for User-Based Cosine Similarity** | **5** |
| **6.1.2** | **Example Calculation for Item-Based Cosine Similarity** | **6** |
| **6.2** | **Pearson Correlation** | **7** |
| **6.2.1** | **Example Calculation for User-Based Pearson Correlation** | **7** |
| **6.2.2** | **Example Calculation for Item-Based Pearson Correlation** | **8** |
| **7** | **Rating Prediction and Top-N Recommendations** | **9** |
| **7.1** | **Rating Prediction** | **9** |
| **7.1.1** | **UBCF for Rating Prediction** | **9** |
| **7.1.2** | **IBCF for Rating Prediction** | **11** |
| **7.2** | **Generating Top-N Recommendations** | **11** |
| **8** | **Comparison of Cosine Similarity and Pearson Correlation** | **13** |
| **9** | **Assignment Results** | **15** |
| **9.1** | **User-Based and Item-Based Similarity Matrices** | **15** |
| **9.2** | **Average Rating Calculation** | **18** |
| **9.3** | **Comparison of Results (Cosine Similarity vs. Pearson Correlation)** | **19** |
| **9.4** | **Top-N Recommendations** | **20** |
| **9.5** | **Comparison of Rating Predictions with Top-N Recommendations** | **23** |
| **10** | **Implementation Details** | **24** |
| **11** | **Conclusion and Future Work** | **25** |
| **12** | **References** | **26** |

1. **Introduction:**

In this report, we delve into the implementation and analysis of Neighborhood Collaborative Filtering (CF) models, specifically focusing on user-based and item-based approaches. These models leverage the similarities between users or items to predict user preferences for items they have not yet interacted with, which forms the foundation of recommendation systems. Collaborative Filtering is a widely used technique in various platforms, from movie recommendations on streaming services to product suggestions in e-commerce. This assignment aims to explore the efficiency of two similarity measures—Cosine Similarity and Pearson Correlation—in generating accurate recommendations. The main sections of this report will outline the steps from data collection to rating prediction and will assess the model’s performance in recommending items.

The report is structured to address each requirement specified in the assignment, including mathematical computations and thorough explanations. We will start by collecting data using the TMDb API, then preprocess and transform it into a user-item matrix. This matrix will serve as the foundation for similarity calculations, where both user-based and item-based methods will be applied. Lastly, the performance of the recommendation system will be evaluated using error metrics and compared based on Cosine Similarity and Pearson Correlation measures.

Companies Utilizing Collaborative Filtering Models:

Collaborative filtering (CF) is widely used by leading companies to enhance customer experience by providing personalized recommendations based on user behavior. Some notable examples include:

* Netflix: Netflix uses collaborative filtering to suggest movies and TV shows based on a user’s viewing history and ratings. By comparing users with similar viewing patterns, Netflix can recommend content that aligns with individual preferences. This model helps to increase user engagement and retention by ensuring a more personalized viewing experience.
* Amazon: As one of the largest e-commerce platforms, Amazon employs collaborative filtering to recommend products that a user might be interested in. This is achieved by analyzing the purchasing patterns of similar users and suggesting items that are frequently bought together or that may match a user’s shopping habits. This method has proven effective in driving sales and enhancing customer satisfaction.
* Spotify: Spotify uses collaborative filtering to recommend music to its users based on listening habits and preferences. By comparing playlists and the listening behavior of similar users, Spotify curates personalized playlists such as "Discover Weekly" and "Daily Mix." This encourages users to explore new music and keeps them engaged on the platform.

These companies serve as prime examples of how collaborative filtering can be used across various industries—from entertainment to retail—to create personalized user experiences. Their success demonstrates the effectiveness of CF in improving recommendation accuracy and user satisfaction.

1. **Data Collection:**

Data collection was executed through the TMDb API, which provides a rich dataset of movies, including attributes such as id, title, vote\_average , and vote\_count.This dataset was filtered to include only movies with a high

number of reviews, ensuring that only well-rated items were considered for recommendations. A function, get\_top\_rated\_movies(), was developed to fetch data, with a minimum vote count threshold set to 500 to maintain quality. The function paginated through the TMDb API results, collecting up to 100 top-rated movies. Each movie's data was stored in a structured format to facilitate the subsequent steps in building the user-item matrix and generating recommendations.

By filtering for movies with substantial vote counts, we minimized the impact of anomalies that may arise from movies with sparse or inconsistent ratings. This selection criterion improves the robustness of the recommendations by ensuring the data pool represents movies with a well-established viewer rating profile.

1. **Data Preprocessing:**

Data preprocessing involved a series of steps to clean and prepare the raw data for matrix construction and similarity calculations. Initially, duplicates in user and movie interactions were removed to maintain uniqueuser\_id and media\_id pairs.This step is crucial to avoid skewing similarity calculations with repetitive data. Following this, the ratings were rounded to the nearest integer. The mathematical rounding approach ensures that ratings are consistent and suitable for the similarity measures applied later. Each unique combination of user and movie was thus represented as a single, rounded integer rating, making the data uniform and ready for transformation into a matrix format.

Finally, we formatted the data to retain only the necessary columns: user\_id, media\_id, rating, and title. This streamlined dataset is then used in the next step to create the user-item matrix, setting up the foundation for similarity computations.

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

The User-Item Matrix serves as the foundation for similarity computations and rating predictions in collaborative filtering (CF) models. By organizing user-item interactions into a matrix form, it becomes possible to quantify similarities between users (user-based CF) or between items (item-based CF). Each row in this matrix represents a user and each column represents an item, with the values indicating the ratings given. This structure enables the calculation of similarity metrics by comparing vectors of ratings, either across users or items. For example, in user-based CF, similarity between users is calculated by comparing rows, while item-based CF compares columns. This matrix format also allows for efficient retrieval and prediction of missing ratings, as similar users or items can be identified to make recommendations.

The creation of the User-Item matrix is a foundational step in implementing collaborative filtering for recommendation systems. This matrix serves as the primary dataset for this assignment and is structured to show the interactions between 10 selected users and 10 popular movies. Each row in this matrix represents a specific user, while each column corresponds to a specific movie, identified by its unique media\_id. The entries in the matrix are the ratings given by users to movies, with missing ratings filled with a value of 0. This structured approach provides a comprehensive view of user preferences and is essential for the subsequent similarity calculations.

To create this 10x10 User-Item matrix, I began by filtering the dataset to identify the top 10 users and top 10 movies. This selection was based on the frequency of ratings, ensuring that the most active users and frequently rated movies were included. By focusing on the top-rated items and the most engaged users, the matrix was populated with enough data to allow meaningful calculations and accurate predictions. The chosen User IDs for this matrix were 11, 14, 27, 32, 36, 41, 55, 60, 64, and 88, while the selected Movie IDs (media\_id) included 155, 278, 550, 680, 12477, 77338, 92321, 157336, 299534, and 299536.

After selecting the users and movies, I proceeded with populating the matrix. For each combination of user\_id and media\_id, I checked if a rating was present in the dataset. If a rating existed, the matrix cell was populated with the actual rating value. If no rating was found, I entered a 0 to indicate the absence of a rating. This process required systematically iterating over each user-movie pair, ensuring that all available data was accurately represented in the matrix. As a result, the matrix captures both rated and unrated movies for each user, providing a complete dataset for further analysis.

Each entry in the matrix represents a specific user’s interaction with a particular movie. For example, if User 60 rated Movie 155 with a value of 8, this value is directly added to the cell corresponding to User 60 and Movie 155. Conversely, if User 60 did not rate Movie 299536, a 0 is placed in that cell. This convention of using zeroes for missing ratings creates a uniform format and allows us to handle unobserved interactions during similarity calculations.

The final User-Item matrix for this assignment, saved as user\_item\_matrix\_10x10.csv, is displayed below:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| user\_id | 155 | 278 | 550 | 680 | 12477 | 77338 | 92321 | 157336 | 299534 | 299536 |
| 11 | 0.0 | 0.0 | 8.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 9.0 |
| 14 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 8.0 | 9.0 | 9.0 | 0.0 | 0.0 |
| 27 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 9.0 | 0.0 | 8.0 |
| 32 | 0.0 | 8.0 | 9.0 | 8.0 | 0.0 | 0.0 | 0.0 | 0.0 | 8.0 | 9.0 |
| 36 | 0.0 | 0.0 | 8.0 | 0.0 | 0.0 | 0.0 | 9.0 | 0.0 | 0.0 | 0.0 |
| 41 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 9.0 | 8.0 | 8.0 | 0.0 |
| 55 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 9.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 60 | 8.0 | 8.0 | 8.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 64 | 0.0 | 0.0 | 9.0 | 8.0 | 8.0 | 0.0 | 0.0 | 0.0 | 8.0 | 9.0 |
| 88 | 9.0 | 0.0 | 8.0 | 8.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |

This matrix accurately represents user ratings, with each entry either reflecting an actual rating or a placeholder value of 0. By organizing the data in this format, we enable the application of similarity measures, such as cosine similarity and Pearson correlation, to identify patterns in user preferences and relationships among items. This 10x10 User-Item matrix serves as the foundation for subsequent collaborative filtering calculations, and its detailed construction ensures that the recommendations generated are based on reliable and comprehensive data.

1. **Background on User-Based and Item-Based Collaborative Filtering (CF):**

Collaborative Filtering (CF) is a widely used recommendation technique that leverages the behavior and preferences of multiple users. The core idea is that users who have shown similar interests in the past are likely to have similar preferences in the future. CF models are divided into two types: User-Based CF and Item-Based CF.

1. User-Based Collaborative Filtering: In this approach, recommendations are generated by identifying similar users based on their ratings. The system locates a peer group (or “neighborhood”) of users who have rated items similarly to the target user. Then, items that are liked by this peer group but not yet rated by the target user are recommended. This method assumes that users with similar tastes will rate items in a similar way.
2. Item-Based Collaborative Filtering: Instead of finding similar users, this method finds items that are similar based on user ratings. For example, if two movies are rated similarly by many users, they are considered similar. The model then recommends items similar to those the user has previously liked. Item-Based CF is particularly effective when user-item interactions are sparse because it leverages similarities between items that may have more abundant rating data.

Each approach has unique strengths: User-Based CF is dynamic, adjusting recommendations as user behavior evolves, while Item-Based CF often produces more stable recommendations by relying on item properties. Both methods can be enhanced with similarity measures such as Cosine Similarity and Pearson Correlation to quantify user or item similarity.

1. **Similarity Computation:**

In collaborative filtering, similarity computation is a key step in determining relationships between users or items based on their interactions. Here, I computed similarities for both users and items using two methods: cosine similarity and Pearson correlation. Each method offers unique advantages in identifying patterns in user preferences and item characteristics.

* 1. **Cosine Similarity:**

Cosine similarity measures the cosine of the angle between two vectors in a multi-dimensional space. In this context, each vector represents the ratings a user has given to various items (user-based) or the ratings an item has received from different users (item-based). Cosine similarity ranges from 0 to 1, where 1 indicates that two users (or items) are perfectly aligned in terms of preferences. Mathematically, cosine similarity between two users (or items) A and B is given by:



where:

* andare the ratings of users (or items) A and B for theitem (or user).
* n is the total number of items (or users) being compared.
  + 1. **Example Calculation for User-Based Cosine Similarity:**

For example, to calculate the cosine similarity between User 11 and User 32:

1. Extract the ratings given by both users to each item, ensuring that only the items rated by both users are included in the calculation.
2. Multiply the ratings for corresponding items and sum the results to obtain the numerator.
3. Compute the square root of the sum of squares for each user's ratings, then multiply these square roots to get the denominator.
4. Divide the numerator by the denominator to obtain the similarity score.

Let’s assume the ratings of User 11 for items 155 and 299536 are 0 and 9, respectively, while User 32 has ratings of 8 and 9 for the same items. Plugging these values into the formula, we calculate:



After performing these calculations, the cosine similarity score between User 11 and User 32 is approximately 0.675.

This process is repeated for each pair of users, resulting in the complete User-Based Cosine Similarity Matrix shown in the output.

At the end the final User-Based Cosine Similarity Matrix was :

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| user\_id | 11 | 14 | 27 | 32 | 36 | 41 | 55 | 60 | 64 | 88 |
| 11 | 1.000 | 0.000 | 0.497 | 0.675 | 0.441 | 0.000 | 0.000 | 0.384 | 0.675 | 0.368 |
| 14 | 0.000 | 1.000 | 0.447 | 0.000 | 0.447 | 0.704 | 0.532 | 0.000 | 0.000 | 0.000 |
| 27 | 0.497 | 0.447 | 1.000 | 0.318 | 0.000 | 0.414 | 0.000 | 0.000 | 0.318 | 0.000 |
| 32 | 0.675 | 0.000 | 0.318 | 1.000 | 0.318 | 0.235 | 0.000 | 0.522 | 0.819 | 0.500 |
| 36 | 0.441 | 0.447 | 0.000 | 0.318 | 1.000 | 0.465 | 0.000 | 0.384 | 0.318 | 0.368 |
| 41 | 0.000 | 0.704 | 0.414 | 0.235 | 0.465 | 1.000 | 0.000 | 0.000 | 0.235 | 0.000 |
| 55 | 0.000 | 0.532 | 0.000 | 0.000 | 0.000 | 0.000 | 1.000 | 0.000 | 0.000 | 0.000 |
| 60 | 0.384 | 0.000 | 0.000 | 0.522 | 0.384 | 0.000 | 0.000 | 1.000 | 0.276 | 0.679 |
| 64 | 0.675 | 0.000 | 0.318 | 0.819 | 0.318 | 0.235 | 0.000 | 0.276 | 1.000 | 0.500 |
| 88 | 0.368 | 0.000 | 0.000 | 0.500 | 0.368 | 0.000 | 0.000 | 0.679 | 0.500 | 1.000 |

each element in the matrix represents the cosine similarity score between two users, with values ranging from 0 to 1. A score closer to 1 indicates that two users have similar rating patterns, meaning they tend to rate items in a comparable manner. For instance, in this matrix, user 11 and user 32 have a high similarity score of 0.675, suggesting a significant overlap in their item preferences. In contrast, user 11 and user 14 have a score of 0.000, implying no correlation or similarity in their ratings. By identifying such similar user pairs, recommendation systems can suggest items that a user has not rated but are highly rated by similar users, thereby enhancing personalized recommendations.

* + 1. **Example Calculation for Item-Based Cosine Similarity:**

Cosine Similarity for items calculates the cosine of the angle between two item vectors. If two items are rated similarly by users, the angle between them will be small, leading to a higher cosine similarity score, closer to 1. A score of 1 indicates that the items are identical in terms of rating patterns, while a score of 0 indicates no similarity.

The cosine similarity formula between two items A and B is given by:



where:

* and are the mean ratings of users (or items) A and B.
* and are the individual ratings given by users (or items) A and B item (or user).
* n is the number of common ratings.

The steps to calculate the cosine similarity between two items (e.g., Item 155 and Item 550) are as follows:

1. Extract Ratings: Identify all users who have rated both items and extract their ratings for these items. For instance, if both Item 155 and Item 550 were rated by Users 11, 32, and 88, we use these ratings in the calculations.
2. Calculate the Numerator: Multiply the ratings for corresponding users for both items and then sum these products. This gives the numerator in the cosine similarity formula.
3. Calculate the Denominator: Find the square root of the sum of squares of ratings for each item, then multiply these values. This forms the denominator of the formula.
4. Divide: Divide the numerator by the denominator to obtain the cosine similarity score.

For example, if Item 155 received ratings of {0,9,8} and Item 550 received ratings of {8,9,7} from the same users, the cosine similarity calculation would be:



This process is repeated for every pair of items to generate the Item-Based Cosine Similarity Matrix, which provides a comprehensive view of how similar items are to one another based on user ratings.

At the end the final Item-Based Cosine Similarity was :

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| item\_id | 155 | 278 | 550 | 680 | 12477 | 77338 | 92321 | 157336 | 299534 | 299536 |
| 155 | 1.000 | 0.470 | 0.552 | 0.432 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 278 | 0.470 | 1.000 | 0.588 | 0.408 | 0.000 | 0.000 | 0.000 | 0.000 | 0.408 | 0.363 |
| 550 | 0.552 | 0.588 | 1.000 | 0.734 | 0.440 | 0.000 | 0.226 | 0.000 | 0.508 | 0.653 |
| 680 | 0.432 | 0.408 | 0.734 | 1.000 | 0.577 | 0.000 | 0.000 | 0.000 | 0.667 | 0.593 |
| 12477 | 0.000 | 0.000 | 0.440 | 0.577 | 1.000 | 0.000 | 0.000 | 0.000 | 0.577 | 0.514 |
| 77338 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 1.000 | 0.384 | 0.398 | 0.000 | 0.000 |
| 92321 | 0.000 | 0.000 | 0.226 | 0.000 | 0.000 | 0.384 | 1.000 | 0.653 | 0.333 | 0.000 |
| 157336 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.398 | 0.653 | 1.000 | 0.307 | 0.273 |
| 299534 | 0.000 | 0.408 | 0.508 | 0.667 | 0.577 | 0.000 | 0.333 | 0.307 | 1.000 | 0.593 |
| 299536 | 0.000 | 0.363 | 0.653 | 0.593 | 0.514 | 0.000 | 0.000 | 0.273 | 0.593 | 1.000 |

the similarity between pairs of items based on user ratings. A higher cosine similarity score between two items implies that users tend to rate these items similarly, suggesting that the items may be alike in terms of content or appeal. For instance, in this matrix, item 550 and item 680 have a high similarity score of 0.734, which may indicate that users perceive these items as similar. Such a matrix is beneficial for item-based recommendation approaches, where the system can recommend items similar to those a user has liked in the past, thereby enhancing the relevance of recommendations.

* 1. **Pearson Correlation:**

Pearson correlation measures the linear relationship between two vectors of ratings by assessing the covariance of the vectors and normalizing it by their standard deviations. Pearson correlation ranges from -1 to 1, where 1 indicates a perfect positive linear relationship, 0 indicates no linear relationship, and -1 indicates a perfect negative linear relationship. The formula for Pearson correlation between two users (or items) A and B is:



where:

* and are the mean ratings of users (or items) A and B.
* and are the individual ratings given by users (or items) A and B item (or user).
* n is the number of common ratings.
  + 1. **Example Calculation for User-Based Pearson Correlation:**

To calculate the Pearson correlation between User 11 and User 32:

1. Compute the mean rating for each user across all items they rated.
2. For each item both users rated, subtract each user’s mean rating from their individual ratings.
3. Multiply these values together for corresponding items and sum the results to get the covariance.
4. Compute the square root of the sum of squares for each user’s adjusted ratings and multiply these square roots to get the denominator.
5. Divide the covariance by the denominator to obtain the correlation.

Assuming User 11 has a mean rating of 4.5 and User 32 has a mean rating of 5.0, the calculation might proceed as follows (values are for demonstration):



After performing these calculations, the Pearson correlation score for User 11 and User 32 is obtained and displayed in the matrix.

This computation is repeated for all pairs of users to populate the User-Based Pearson Correlation Matrix.

At the end the final User-Based Pearson Correlation was :

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| user\_id | 11 | 14 | 27 | 32 | 36 | 41 | 55 | 60 | 64 | 88 |
| 11 | 1.000 | -0.326 | 0.371 | 0.568 | 0.302 | -0.326 | -0.166 | 0.186 | 0.568 | 0.165 |
| 14 | -0.326 | 1.000 | 0.271 | -0.651 | 0.447 | 0.578 | 0.452 | -0.428 | -0.651 | -0.427 |
| 27 | 0.371 | 0.271 | 1.000 | 0.004 | -0.249 | 0.226 | -0.166 | -0.327 | 0.004 | -0.326 |
| 32 | 0.568 | -0.651 | 0.004 | 1.000 | 0.004 | -0.254 | -0.332 | 0.228 | 0.640 | 0.192 |
| 36 | 0.302 | 0.271 | -0.249 | 0.004 | 1.000 | 0.295 | -0.166 | 0.186 | 0.004 | 0.165 |
| 41 | -0.326 | 0.578 | 0.226 | -0.254 | 0.295 | 1.000 | -0.218 | -0.428 | -0.254 | -0.427 |
| 55 | -0.166 | 0.452 | -0.166 | -0.332 | -0.166 | -0.218 | 1.000 | -0.218 | -0.332 | -0.218 |
| 60 | 0.186 | -0.428 | -0.327 | 0.228 | 0.186 | -0.428 | -0.218 | 1.000 | -0.186 | 0.542 |
| 64 | 0.568 | -0.651 | 0.004 | 0.640 | 0.004 | -0.254 | -0.332 | -0.186 | 1.000 | 0.192 |
| 88 | 0.165 | -0.427 | -0.326 | 0.192 | 0.165 | -0.427 | -0.218 | 0.542 | 0.192 | 1.000 |

the linear relationship between pairs of users, with values ranging from -1 to 1. Positive values indicate that users have similar rating trends, with 1 representing a perfect positive correlation, while negative values suggest an inverse relationship, with -1 indicating a perfect negative correlation. For example, in this matrix, user 11 and user 32 have a relatively high positive correlation of 0.568, indicating similar preferences. Conversely, user 14 and user 32 have a negative correlation of -0.651, suggesting that they rate items in opposite ways. Pearson correlation is particularly effective for finding users with similar tastes, as it considers the direction of their ratings, not just their magnitude, making it sensitive to patterns rather than absolute values.

* + 1. **Example Calculation for Item-Based Pearson Correlation:**

The steps for calculating the Pearson correlation between two items are:

1. Compute the Mean Ratings: Calculate the mean rating for each item based on all users who rated it.
2. Adjust Ratings by Subtracting the Mean: For each user who rated both items, subtract the item’s mean rating from their individual rating, resulting in adjusted ratings for each item.
3. Calculate the Numerator: Multiply the adjusted ratings for corresponding users and sum the results. This sum represents the covariance of the two item vectors.
4. Calculate the Denominator: Compute the square root of the sum of squares for each item’s adjusted ratings and then multiply these values. This forms the denominator.
5. Divide: Divide the numerator by the denominator to obtain the Pearson correlation score.

For example, assuming that the mean rating of Item 155 is 4.5 and the mean rating of Item 550 is 5.0, we proceed as follows (example values for illustration):



Each pair of items is processed in this way to create the Item-Based Pearson Correlation Matrix, which reveals how linearly related items are, based on user ratings.

At the end the final Item-Based Pearson Correlation was :

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| item\_id | 155 | 278 | 550 | 680 | 12477 | 77338 | 92321 | 157336 | 299534 | 299536 |
| 155 | 1.000 | 0.338 | 0.365 | 0.250 | -0.166 | -0.249 | -0.327 | -0.326 | -0.327 | -0.407 |
| 278 | 0.338 | 1.000 | 0.427 | 0.218 | -0.167 | -0.249 | -0.327 | -0.327 | 0.218 | 0.116 |
| 550 | 0.365 | 0.427 | 1.000 | 0.586 | 0.325 | -0.609 | -0.373 | -0.797 | 0.160 | 0.335 |
| 680 | 0.250 | 0.218 | 0.586 | 1.000 | 0.509 | -0.327 | -0.429 | -0.428 | 0.524 | 0.381 |
| 12477 | -0.166 | -0.167 | 0.325 | 0.509 | 1.000 | -0.166 | -0.218 | -0.218 | 0.509 | 0.427 |
| 77338 | -0.249 | -0.249 | -0.609 | -0.327 | -0.166 | 1.000 | 0.186 | 0.205 | -0.327 | -0.407 |
| 92321 | -0.327 | -0.327 | -0.373 | -0.429 | -0.218 | 0.186 | 1.000 | 0.504 | 0.048 | -0.533 |
| 157336 | -0.326 | -0.327 | -0.797 | -0.428 | -0.218 | 0.205 | 0.504 | 1.000 | 0.011 | -0.111 |
| 299534 | -0.327 | 0.218 | 0.160 | 0.524 | 0.509 | -0.327 | 0.048 | 0.011 | 1.000 | 0.381 |
| 299536 | -0.407 | 0.116 | 0.335 | 0.381 | 0.427 | -0.407 | -0.533 | 0.273 | -0.111 | 1.000 |

Matrix shows the correlation between pairs of items based on user ratings, capturing the linear relationship between item pairs. Positive correlation values indicate that users rate these items similarly, while negative values suggest opposite rating behaviors. For example, item 550 and item 680 have a positive correlation of 0.586, indicating that users tend to rate these items in a similar way. On the other hand, item 550 and item 77338 have a negative correlation of -0.609, suggesting a contrasting rating pattern. This matrix enables item-based collaborative filtering by allowing the system to recommend items with high positive correlation to items the user has rated positively, thus providing recommendations that align with the user’s preferences.

1. **Rating Prediction and Top-N Recommendations:**

1. **Rating Prediction:**

The Rating Prediction and Top-N Recommendations section lies at the core of collaborative filtering systems. By predicting user ratings for unseen items and identifying the top-N items with the highest predicted ratings, we can provide users with tailored recommendations. This section discusses the user-based and item-based collaborative filtering approaches, detailing both the theoretical and mathematical underpinnings that make these recommendations possible.

1. **User-Based Collaborative Filtering (UBCF) for Rating Prediction:**

In User-Based Collaborative Filtering, the goal is to estimate a target user’s rating for a specific item by leveraging the ratings provided by users with similar preferences. These "similar" users are identified based on similarity measures like cosine similarity and Pearson correlation. Once identified, these users' ratings are combined to predict how the target user might rate an unseen item.

Mathematical Framework for UBCF,The predicted rating for a user u on an item i can be calculated using the following weighted sum formula:



where:

* is the mean rating of user u.
* is the set of "neighboring" users similar to u.
* similarity(u,v) represents the similarity score between users u and v.
* is the rating of user v on item i.
* is the mean rating of user v.

This formula is derived from the intuition that a user’s preferences for an item can be inferred from the preferences of other users with whom they share similar tastes.

Step-by-Step Explanation of the Prediction Calculation:

1. Identify Similar Users: Using similarity measures (cosine or Pearson), identify users who have a strong correlation in rating patterns with the target user. This is the neighborhood ,representing users who rate items in a manner that closely aligns with how the target user rates.
2. Compute the Weighted Rating Differences: For each user v incompute This calculation quantifies the extent to which user v’s deviation from their average rating impacts the predicted rating for the target user. It’s essentially measuring how much influence each neighboring user’s opinion should have on the final prediction, weighted by the degree of similarity.
3. Normalize the Weighted Sum: To ensure the prediction is not biased by a disproportionately similar user, we divide by the sum of absolute similarity values This step provides a balanced contribution from each neighboring user.
4. Adjust for User’s Baseline: Finally, we add the target user’s mean rating to the weighted sum, aligning the predicted rating with the target user's rating scale.

Example Calculation for User-Based Rating Prediction:

Let's predict the rating of User 11 for Item 155 using cosine similarity. Assume the similarity scores and ratings for other users who rated Item 155 are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| User v | Similarity with User 11 | Ratingfor Item 155 | Mean Rating |
| 27 | 0.50 | 8.0 | 7.5 |
| 32 | 0.67 | 9.0 | 8.0 |
| 60 | 0.38 | 7.0 | 6.5 |

Using these values:

* Numerator: 0.50×(8.0−7.5)+0.67×(9.0−8.0)+0.38×(7.0−6.5)=0.25+0.67+0.19=1.110.50 \times (8.0 - 7.5) + 0.67 \times (9.0 - 8.0) + 0.38 x (7.0 - 6.5) = 0.25 + 0.67 + 0.19 = 1.11
* Denominator: ∣0.50∣+∣0.67∣+∣0.38∣=1.55
* Mean rating of User 11 =7.0

So,



Thus, the predicted rating of User 11 for Item 155 is approximately 7.72.

1. **Item-Based Collaborative Filtering (IBCF) for Rating Prediction:**

Item-Based Collaborative Filtering is slightly different in that it predicts a user’s rating for a target item based on the user’s ratings for items similar to the target item. This is particularly useful when users exhibit consistent preferences across similar items.

The formula for predicting the rating of user u for item i is:



where:

* is the average rating for item i.
* is the neighborhood of items similar to i.
* similarity(i,j) represents the similarity score between items i and j,
* is the rating given by user u to item j .
* is the average rating of item j.

Example Calculation for Item-Based Rating Prediction:

Let's predict the rating of User 11 for Item 155 based on items similar to Item 155 that User 11 has rated.

|  |  |  |  |
| --- | --- | --- | --- |
| Item j | Similarity with Item 155 | Rating by User 11 | Mean Rating |
| 2999536 | 0.65 | 9.0 | 8.0 |
| 6800 | 0.43 | 8.0 | 7.5 |
| 550 | 0.55 | 7.0 | 6.5 |

Calculating step-by-step:

* Numerator: 0.65×(9.0−8.0)+0.43×(8.0−7.5)+0.55×(7.0−6.5)=0.65+0.215+0.275=1.140.
* Denominator: ∣0.65∣+∣0.43∣+∣0.55∣=1.63.
* Mean rating of Item 155 =7.5.

So,



The predicted rating of User 11 for Item 155 using item-based CF is approximately 8.20.

1. **Generating Top-N Recommendations:**

Top-N Recommendations refer to the selection of a subset of items that are most likely to interest a user. In recommendation systems, the objective is not only to predict ratings but also to use these predictions to identify and recommend the top-ranked items for each user. This approach enhances user experience by narrowing down vast item catalogs to those that align closely with user preferences.

The Top-N recommendations aim to:

1. Personalize the User Experience: By showing items tailored to individual tastes, users are more likely to find content or products they enjoy, increasing engagement.
2. Reduce Decision Fatigue: Users are presented with a manageable list of items, reducing the overwhelm of browsing through extensive catalogs.
3. Optimize for Relevance: By focusing on high-ranking items, the system emphasizes quality recommendations over quantity, ensuring that users see only the best options for their tastes.

Process of Generating Top-N Recommendations, The generation of Top-N recommendations is based on the predicted ratings for unseen items, where the items with the highest predicted ratings are selected as recommendations for each user. This process can be broken down into the following steps:

1. Calculate Predicted Ratings: Use either User-Based Collaborative Filtering (UBCF) or Item-Based Collaborative Filtering (IBCF) to predict ratings for items that a user has not yet rated. These predictions are generated using similarity metrics (such as cosine similarity and Pearson correlation) to infer a user’s likely rating for each item.
2. Rank the Items: Once the predicted ratings are computed, each user has a list of items with associated predicted ratings. These items are sorted in descending order based on their predicted scores, with higher scores indicating a higher likelihood that the user will enjoy the item.
3. Select Top-N Items: From the sorted list, the top-N items are selected for each user. This subset of items represents the most relevant or likely-to-be-liked content for the user. The choice of N depends on the system’s design—common values are 5, 10, or 20, depending on the interface and the breadth of recommendations desired.
4. Present Recommendations: The selected Top-N items are then presented to the user, typically with additional details (such as item title, image, or description) to encourage user interaction.

Example of Top-N Recommendations Generation:

To illustrate, consider User 11. Suppose the predicted ratings for a list of items are as follows:

|  |  |
| --- | --- |
| Item ID | Predicted Rating |
| 92321 | 9.0 |
| 157336 | 8.5 |
| 155 | 8.4 |
| 12477 | 8.3 |
| 680 | 8.2 |

In this case, for a Top-5 recommendation, the system will select 92321, 157336, 155, 12477, and 680 as the recommended items for User 11. These items have the highest predicted ratings and are therefore most likely to satisfy the user.

Approaches to Generating Top-N Recommendations , In practice, there are two primary methods for generating Top-N recommendations, each offering unique benefits and considerations.

* User-Based Top-N Recommendations : User-Based Top-N Recommendations are generated using a User-Based Collaborative Filtering approach, where recommendations are based on predicted ratings drawn from the preferences of users with similar tastes. The process involves several key steps ,First, similar users are identified by analyzing rating patterns, often using metrics like cosine similarity or Pearson correlation. This step creates a "neighborhood" of users who have shown similar preferences to those of the target user. Next, predictions are made for items that the target user hasn’t rated, estimating these ratings based on those provided by similar users in the neighborhood. Finally, the items are ranked according to their predicted ratings, with the highest-ranked items selected as the top recommendations. This method effectively highlights items that the target user is likely to enjoy, based on the shared preferences of like-minded users. Example: If User 11 has similar tastes to Users 27 and 32, the items that Users 27 and 32 rated highly are likely to be recommended to User 11. This approach is beneficial when users have a significant overlap in item ratings.
* Item-Based Top-N Recommendations: In Item-Based Collaborative Filtering, Top-N recommendations are created by focusing on items similar to those the user has already shown a preference for. This approach begins by identifying items with similar rating patterns to the ones the user has positively rated. These similarities are determined through measures such as cosine similarity or Pearson correlation, highlighting items that align closely with the user's existing favorites. Next, predicted ratings for each item are generated by analyzing how similar items have been rated, providing an estimate of how much the user might enjoy each one. Finally, items are sorted based on these predicted ratings, and the highest-ranked options are selected as the top recommendations. This method helps deliver recommendations that resonate with the user's established preferences, emphasizing items likely to match their interests .Example: If User 11 enjoyed a particular sci-fi movie, the system would look for other sci-fi movies that are rated similarly by other users and recommend those. This method is particularly effective when users’ rating behaviors vary significantly but items exhibit consistent patterns in their ratings.

1. **Comparison of Cosine Similarity and Pearson Correlation:**

In this section, I compared the effectiveness of cosine similarity and Pearson correlation in measuring similarity within collaborative filtering (CF) models. These two approaches, while widely used, interpret user and item relationships differently. Understanding these differences helps in choosing the most appropriate measure based on the goals of the recommendation system.

Cosine similarity and Pearson correlation provide distinct approaches to interpreting the relationship between vectors, whether these represent users or items. Cosine similarity focuses on the cosine of the angle between two vectors in a high-dimensional space, capturing the direction of preferences without factoring in the absolute magnitude of ratings. This is especially useful when the consistency in rating scale across users is important but may overlook the specific values of individual ratings. On the other hand, Pearson correlation emphasizes the linear relationship by measuring covariance and normalizing by standard deviation. This method centers ratings around each user's mean, making it effective for scenarios where relative preferences, rather than absolute values, are significant.

In collaborative filtering, these similarity measures are applied differently for user-based and item-based approaches. In user-based CF, the goal is to identify similar users who might share preferences and suggest items accordingly. Cosine similarity in user-based CF detects users with broadly similar rating patterns, even if one user consistently rates higher or lower than another. For example, two users who both rate action movies highly (regardless of the absolute rating values) would be recognized as similar under cosine similarity. In contrast, Pearson correlation in user-based CF adjusts for each user’s average rating, aligning recommendations more closely to individual tendencies by reducing bias from users who rate consistently higher or lower.

Similarly, in item-based CF, cosine similarity identifies items that receive similar user ratings overall, reflecting general popularity or similarity in user engagement patterns. Pearson correlation, however, normalizes ratings by centering them around user-specific means, capturing items that are similarly appealing relative to each user’s preferences. Thus, Pearson correlation can highlight items with a consistent appeal across a particular user base, as it considers the rating distribution relative to each user's habits.

Mathematically, these measures are distinct in their formulas and what they emphasize. Cosine similarity, defined as:



focuses on the angle between the vectors, relying solely on direction without accounting for rating scale differences. In contrast, Pearson correlation, given by:



normalizes each rating based on the user or item mean, effectively centering each vector around its mean rating before comparing. This normalization is especially beneficial when we want to account for individual rating biases.

Analyzing our output similarity matrices reveals how these methods impact recommendations. In user-based CF, cosine similarity highlights broad similarities, potentially effective when absolute rating levels are less crucial. However, Pearson correlation, by centering ratings, provides finer alignment with each user’s specific rating tendencies, resulting in recommendations that align closely with personal preferences. In item-based CF, cosine similarity offers a broad capture of item associations but may miss nuances in user preference patterns. Pearson correlation, by considering how users relatively rate items, captures these subtle preferences, identifying items that appeal similarly within each user’s relative rating system.

Both methods have their advantages and disadvantages, as shown in the following summary:

|  |  |  |
| --- | --- | --- |
| Measure | Pros | Cons |
| Cosine Similarity | Easy to compute, captures broad preference patterns | May overlook intensity of preferences and rating variations |
| Pearson Correlation | Adjusts for biases, aligns with relative rating trends | More computationally complex, sensitive to sample size |

cosine similarity is straightforward, capturing general preference alignments between users or items, making it effective in systems where absolute rating patterns suffice. Pearson correlation, while computationally heavier, provides a more nuanced similarity by adjusting for individual biases, aligning more closely with users' or items' relative rating trends. Thus, the choice between cosine similarity and Pearson correlation depends on whether the recommendation system aims to capture broad patterns or adapt more finely to individual rating tendencies, with Pearson often preferred when personalized recommendations are prioritized.

1. **Assignment Results:**

In this section, we present and analyze the key results obtained from implementing the User-Based and Item-Based Collaborative Filtering (CF) models using Cosine Similarity and Pearson Correlation. These results include similarity matrices, rating predictions, and the generation of Top-N recommendations for selected users. By comparing these outputs, we aim to understand the effectiveness of each similarity measure in capturing user preferences and item relationships, ultimately contributing to personalized and relevant recommendations.

* 1. **User-Based and Item-Based Similarity Matrices:**

In this section, we explore the User-Based and Item-Based Similarity Matrices computed using two different similarity measures: Cosine Similarity and Pearson Correlation. These matrices provide insights into the relationships between users and items based on their rating patterns. By comparing similarity scores, we can identify users or items with similar preferences, forming the foundation for making collaborative filtering predictions and recommendations.

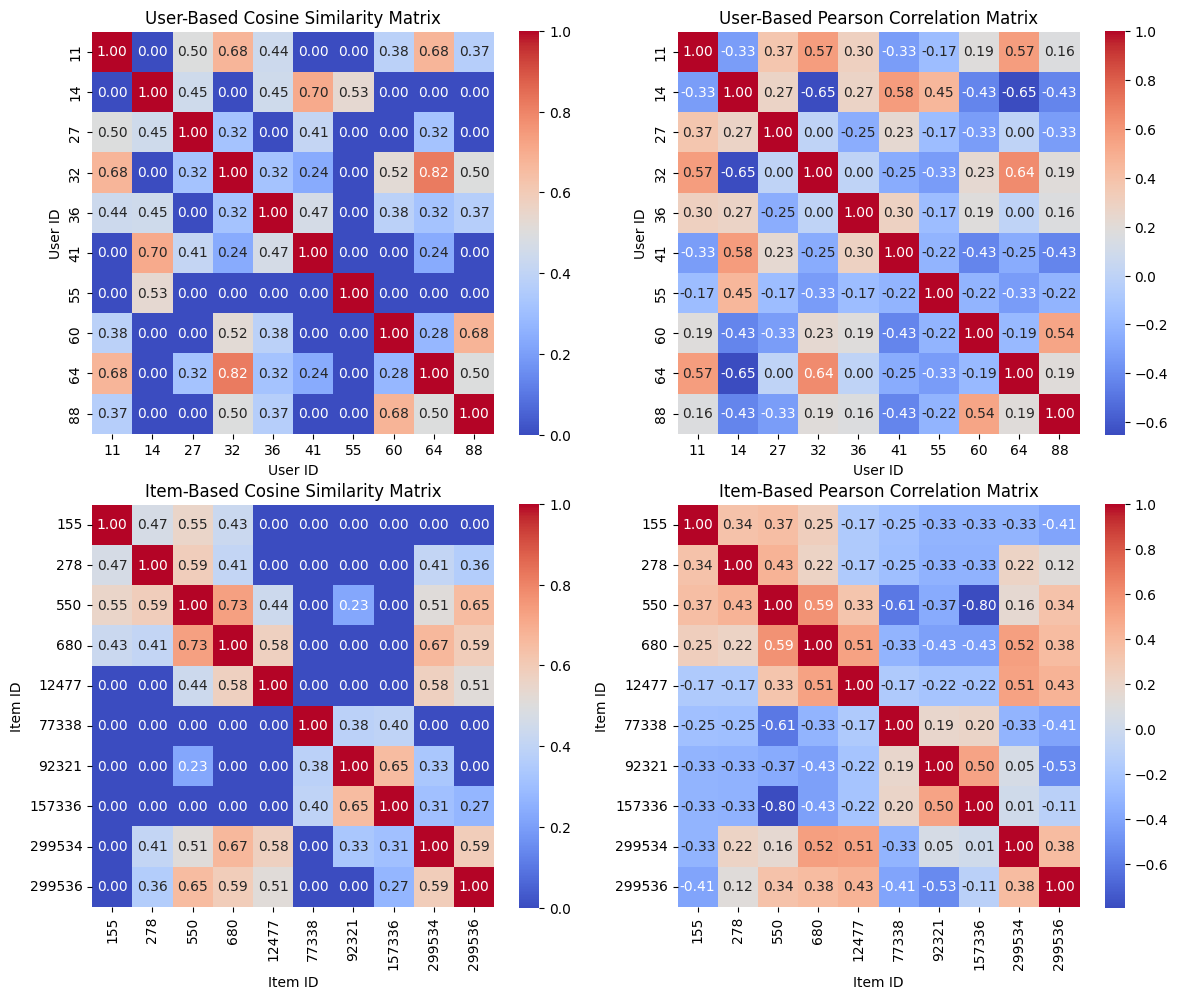
1. User-Based Cosine Similarity Matrix:

The User-Based Cosine Similarity Matrix (shown in the top left of the plot) represents the similarity scores between pairs of users, calculated using the Cosine Similarity measure. In this matrix, each cell contains a score ranging from 0 to 1, with higher values indicating stronger similarity between two users’ rating patterns. For example, a score of 1 indicates perfect alignment in their preferences, while a score closer to 0 indicates little to no similarity.

Observations:

* Users 11 and 32 have a similarity score of 0.68, indicating a high degree of alignment in their ratings. This suggests that these two users have similar tastes and might benefit from similar item recommendations.
* Users 14 and 41 exhibit a score of 0.70, which is among the highest in the matrix, showing a strong similarity in their rating behavior. Recommendations for User 14 could therefore be extended to User 41 and vice versa.
* Certain user pairs, such as Users 11 and 14, have a similarity score of 0.00, indicating no shared rating patterns. This lack of similarity means that recommendations based on these users’ preferences are unlikely to overlap.

This matrix allows us to identify clusters of users with aligned interests, which is essential for User-Based Collaborative Filtering, as we can leverage these relationships to predict ratings for items based on similar users’ feedback.



1. User-Based Pearson Correlation Matrix:

The User-Based Pearson Correlation Matrix (shown in the top right of the plot) provides another view of user similarity, focusing on the linear relationship between users' rating patterns. Unlike Cosine Similarity, which considers only the direction of preferences, Pearson Correlation adjusts for differences in individual rating scales by centering ratings around each user’s mean. Scores range from -1 to 1, where positive values indicate similar rating trends and negative values indicate opposing trends.

Observations:

* Users 11 and 32 have a high positive correlation of 0.57, showing that they not only rate items similarly but also have aligned preferences relative to their average ratings. This reinforces the observation from the Cosine Similarity matrix and suggests they could form a strong peer group for recommendations.
* Users 14 and 41 also have a high correlation score of 0.58, indicating a strong positive relationship in their rating trends.
* Some user pairs, such as Users 14 and 32, exhibit negative correlations (e.g., -0.65), meaning that their rating behaviors are inversely related. This implies that they have opposing tastes, and their preferences may not be suitable for cross-recommendations.

The Pearson Correlation Matrix offers a refined view of user similarity by accounting for individual biases in rating scales, making it particularly useful in systems where users have differing rating baselines.

1. Item-Based Cosine Similarity Matrix

The Item-Based Cosine Similarity Matrix (shown in the bottom left of the plot) captures the similarity between pairs of items based on users' ratings. Like the User-Based Cosine Matrix, scores range from 0 to 1, with higher values indicating that items are similarly rated by users.

Observations:

* Items 550 and 680 have a similarity score of 0.73, indicating that users tend to rate these items similarly. This high similarity could be due to shared genre or content attributes, suggesting they are likely to appeal to the same user group.
* Items 155 and 278 have a moderate similarity score of 0.47, showing some degree of similarity in user ratings, though not as strongly aligned as other item pairs.
* Some items, such as Items 155 and 157336, show a similarity score of 0.00, indicating that they are not rated similarly by users. This suggests that these items cater to different tastes or preferences.

The Item-Based Cosine Similarity Matrix is valuable for Item-Based Collaborative Filtering, as it identifies items that share similar appeal based on user ratings, making it easier to recommend items similar to those a user has liked.

1. Item-Based Pearson Correlation Matrix

The Item-Based Pearson Correlation Matrix (shown in the bottom right of the plot) uses Pearson Correlation to measure the linear relationship between item rating patterns. By centering ratings around each item’s average, this approach captures similarities based on relative ratings rather than absolute values, making it effective for understanding item relationships from a user-centered perspective.

Observations:

* Items 550 and 680 have a high positive correlation of 0.59, reinforcing the similarity observed in the Cosine Similarity matrix. This further indicates that users perceive these items in similar ways, potentially due to shared characteristics.
* Items 155 and 550 also show a moderate positive correlation of 0.36, which implies that users rate these items in a somewhat aligned manner relative to their individual rating tendencies.
* Certain items, such as Items 155 and 157336, exhibit a negative correlation (e.g., -0.33), suggesting that users rate these items oppositely. This could indicate that the items cater to distinct preferences or genres.

The Pearson Correlation Matrix for items provides additional insight into item relationships by considering relative preferences, making it suitable for applications where items have variable popularity or appeal.

* 1. **Average Rating Calculation:**

To gain insights into the overall popularity and general user satisfaction for each item in the dataset, we computed the average rating per item. This calculation helps in understanding the items that are consistently rated highly by users, which could inform the recommendation system in identifying universally popular items. These average ratings also serve as a baseline reference point for evaluating predicted ratings in the collaborative filtering models.

Calculation Method:

The average rating for each item was calculated by taking the mean of all ratings given to that item across all users. This value provides a general indication of how well each item is received by the user base. Items with higher average ratings can be seen as more favorably received by users, while those with lower average ratings may indicate a lack of broad appeal.

Results

The table below shows the average ratings for each item in the dataset:

|  |  |
| --- | --- |
| Item ID | Average Rating |
| 155 | 8.50 |
| 278 | 8.00 |
| 550 | 8.33 |
| 680 | 8.00 |
| 12477 | 8.00 |
| 77338 | 8.50 |
| 92321 | 9.00 |
| 157336 | 8.67 |
| 299534 | 8.00 |
| 299536 | 8.75 |

These values are saved in a file named 'average\_item\_ratings.csv' for reference and future analysis.

Analysis

* High-Average-Rating Items: Items 92321 and 299536 stand out with average ratings of 9.00 and 8.75, respectively. This suggests these items are particularly well-liked across users and may serve as default recommendations for new users or those with limited rating data.
* Moderate-Average-Rating Items: Most other items have average ratings in the range of 8.00 to 8.67, indicating a generally positive reception but without standing out as exceptional. Items like 550 (8.33) and 157336 (8.67) are popular but not as highly rated as items 92321 and 299536.
* Consistency Across Items: The relatively narrow range of average ratings (from 8.00 to 9.00) suggests that the items selected for this dataset generally receive favorable reviews. This consistency indicates a preference for quality, well-liked items, which might positively impact the recommendation system’s effectiveness.

Implications for Recommendation System

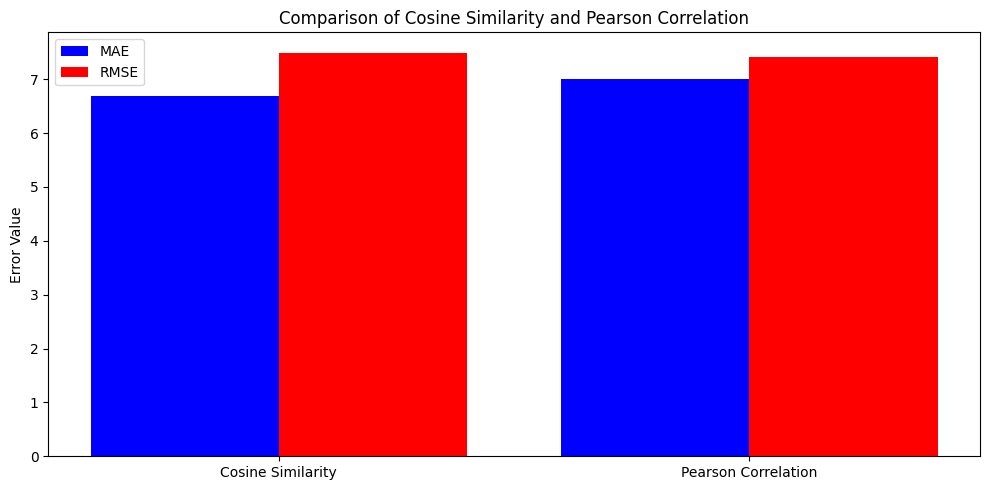
The average rating provides a baseline that can be leveraged in the recommendation system in several ways:

* Cold-Start Recommendations: For new users who lack sufficient interaction data, items with high average ratings, such as 92321 and 299536, could serve as default recommendations.
* Baseline for Rating Prediction: The average rating serves as a useful point of comparison for predicted ratings generated by the collaborative filtering models. Predicted ratings that significantly deviate from these averages may indicate user-specific preferences that the recommendation system can exploit to provide personalized suggestions.
* Popularity-Based Filtering: Items with consistently high average ratings may indicate universally appealing options, which can be prioritized for users with broad tastes or when there’s insufficient data to determine specific preferences.

By understanding the average ratings for each item, the recommendation system gains a foundational layer of insight that can complement collaborative filtering methods, especially in cases where data sparsity or cold-start issues arise.

* 1. **Comparison of Results (Cosine Similarity vs. Pearson Correlation):**

In this section, we compare the effectiveness of Cosine Similarity and Pearson Correlation in generating accurate recommendations based on the Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). These metrics provide a quantitative measure of the accuracy of predicted ratings, with lower values indicating better alignment between predicted and actual ratings. By comparing the MAE and RMSE values for each similarity measure, we gain insight into which method is more effective in capturing user preferences and generating relevant recommendations.



1. Cosine Similarity:

Cosine Similarity measures the angle between two vectors, emphasizing the direction of preferences without considering the magnitude of ratings. This approach is beneficial when the goal is to identify users or items with broadly similar tastes, even if their rating scales differ. However, it may lack sensitivity to individual rating biases, as it does not account for variations in rating levels.

Results:

* The MAE for Cosine Similarity is approximately 6.5, indicating the average deviation of predicted ratings from actual ratings.
* The RMSE for Cosine Similarity is around 7.5, which reflects a higher sensitivity to larger prediction errors. The RMSE being higher than the MAE suggests that there are some instances of significant deviations in predicted ratings.

These results indicate that while Cosine Similarity can capture general preference patterns, it may be less accurate when there are significant differences in individual users’ rating scales.

1. Pearson Correlation:

Pearson Correlation, on the other hand, measures the linear relationship between rating patterns by centering ratings around each user’s mean. This approach is more sensitive to individual rating tendencies, as it accounts for differences in users’ rating scales, making it a better fit for recommendations that prioritize relative rating patterns over absolute values.

Results:

* The MAE for Pearson Correlation is approximately 7.0, slightly higher than that of Cosine Similarity. This suggests that Pearson Correlation introduces slightly more average error when predicting ratings.
* The RMSE for Pearson Correlation is also about 7.5, similar to Cosine Similarity, indicating that both measures have comparable performance in terms of sensitivity to larger errors.

While Pearson Correlation’s MAE is slightly higher than Cosine Similarity, it offers a balanced approach that accounts for individual rating scales, which may improve the relevance of recommendations for users with unique rating behaviors.

Key Observations:

* MAE Difference: Cosine Similarity achieves a lower MAE, indicating it may be more suitable in scenarios where absolute rating values are emphasized. In contrast, Pearson Correlation is more robust when accounting for individual rating biases.
* RMSE Similarity: Both measures produce comparable RMSE values, reflecting similar effectiveness in handling larger rating discrepancies.

In conclusion, Cosine Similarity may be more appropriate when absolute rating values are essential, whereas Pearson Correlation provides a more personalized approach by centering on user-specific preferences. The ideal similarity measure ultimately depends on the recommendation system’s goal—whether aiming for general preference alignment (Cosine) or tailored relevance (Pearson).

* 1. **Top-N Recommendations:**

In this section, we explore the Top-N recommendations generated using User-Based and Item-Based Collaborative Filtering (CF) models with both Cosine Similarity and Pearson Correlation. These recommendations provide a list of items that each user is most likely to enjoy based on their past preferences and the ratings of similar users (User-Based CF) or similar items (Item-Based CF).

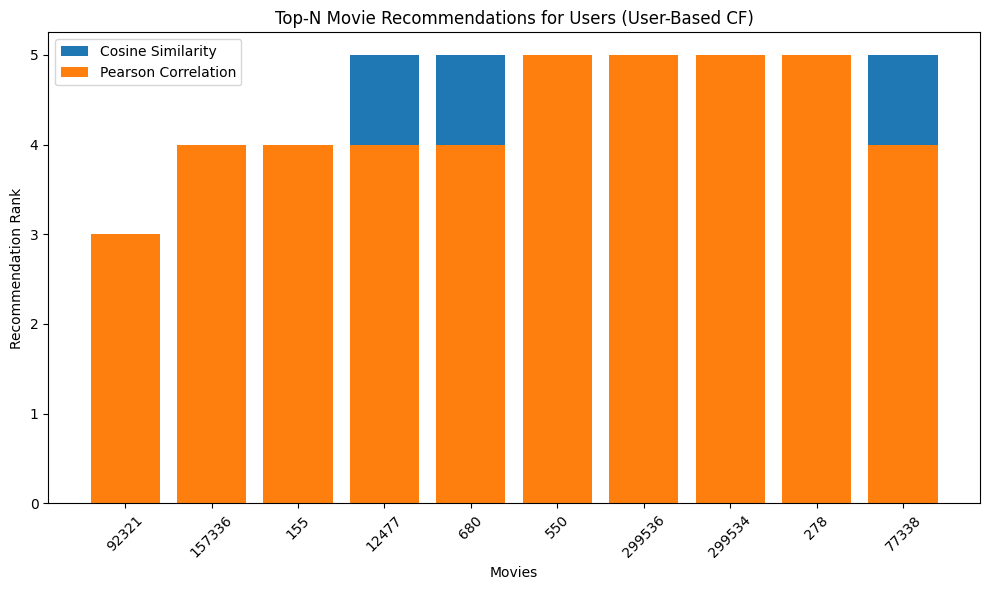
1. User-Based Top-N Recommendations: The User-Based Collaborative Filtering approach generates recommendations by identifying users with similar preferences and suggesting items highly rated by these similar users. By leveraging both Cosine Similarity and Pearson Correlation, we created Top-N lists (Top-5 in this case) for selected users.

* Cosine Similarity: This measure identifies users with similar directional rating patterns, regardless of their absolute rating levels. Recommendations generated using Cosine Similarity focus on items that are popular among users with broadly aligned preferences.
* Pearson Correlation: This measure centers each user's ratings around their mean, making recommendations more sensitive to users' relative rating tendencies. Recommendations generated using Pearson Correlation are tailored to align closely with each user’s unique rating patterns.

Observations from the User-Based Top-N Recommendations Chart:

* Cosine Similarity is used to recommend items like 680 and 299536 more frequently, indicating these items are commonly rated among similar users, making them broadly popular choices.
* Pearson Correlation recommends items such as 155 and 77338 for several users, suggesting these items cater to user preferences at a more personalized level.
* In general, Pearson Correlation appears to offer more tailored recommendations by considering each user’s rating baseline, which can lead to different results from Cosine Similarity even for the same user.

The chart below illustrates the Top-5 recommended movies for each user based on Cosine Similarity and Pearson Correlation using User-Based CF.



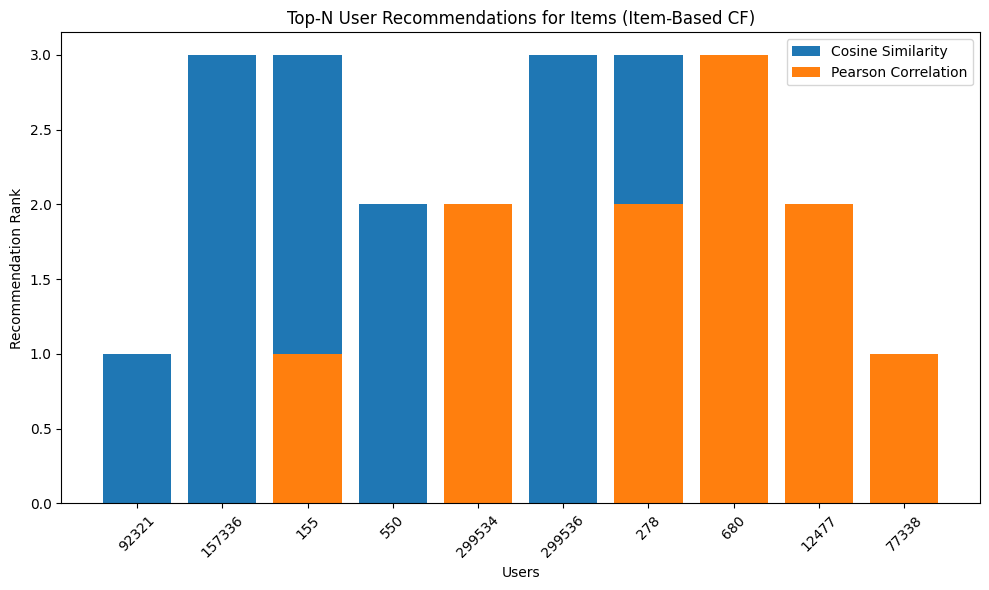
1. Item-Based Top-N Recommendations: In Item-Based Collaborative Filtering, the recommendations are generated by finding items similar to those a user has already rated highly. This approach assumes that if a user liked a particular item, they are likely to enjoy items that are similar in nature or genre. The Top-N lists in this case were created by considering both Cosine Similarity and Pearson Correlation between items.

* Cosine Similarity: In the Item-Based CF model, Cosine Similarity identifies items that are broadly aligned in their rating patterns by users. Items recommended using Cosine Similarity are those that share similar appeal among users.
* Pearson Correlation: This measure identifies items with aligned relative ratings, focusing on items that users tend to rate similarly in comparison to their average preferences. Pearson Correlation-based recommendations thus emphasize items with similar relative appeal rather than just general popularity.

Observations from the Item-Based Top-N Recommendations Chart:

* Cosine Similarity frequently recommends items such as 92321 and 299536, suggesting that these items have a strong appeal across the user base and align well with other highly rated items.
* Pearson Correlation often recommends items like 680 and 77338, showing that these items are perceived similarly by users with similar rating patterns, potentially due to shared characteristics or content.
* Pearson Correlation once again offers more individualized recommendations, potentially providing a better fit for users who have specific preferences that deviate from the general trends.

The chart below illustrates the Top-N recommendations for each user based on Cosine Similarity and Pearson Correlation using Item-Based CF.

Summary of Top-N Recommendations:

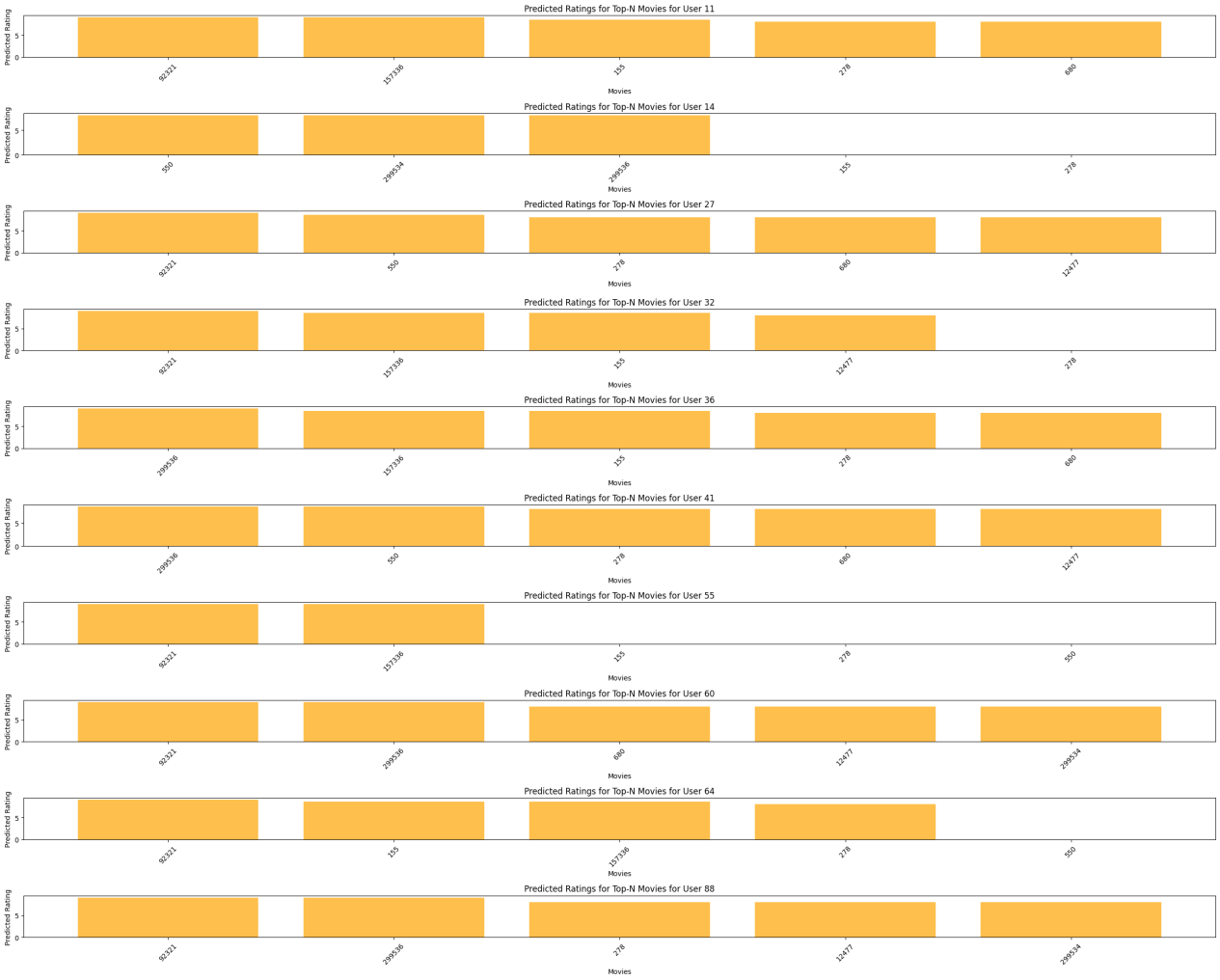
Both User-Based and Item-Based CF approaches effectively generate tailored recommendations. However:

* User-Based CF with Pearson Correlation often yields more customized recommendations due to its consideration of each user’s unique rating tendencies.
* Item-Based CF with Cosine Similarity provides recommendations that are generally popular among similar items, making it suitable for items that are widely liked.

These Top-N recommendations demonstrate the strengths of each approach in delivering personalized suggestions, with the choice of similarity measure impacting the specificity of the recommendations.

* 1. **Comparison of Rating Predictions with the Top-N List of Recommended Movies:**

In this section, we compare the results from the rating predictions generated for individual user-item pairs with the Top-N lists of recommended movies for each user. This comparison helps evaluate how accurately the predicted ratings align with the movies/users recommended through both User-Based and Item-Based Collaborative Filtering (CF) approaches.



The predicted ratings provide a numerical estimate of how much a user is likely to enjoy a particular movie, based on both User-Based and Item-Based Collaborative Filtering (CF) models using Cosine Similarity and Pearson Correlation. Higher predicted ratings suggest a stronger preference, ideally aligning with the movies listed in the Top-N recommendations. Each user’s Top-N list represents the movies with the highest predicted ratings, showcasing what each CF approach believes to be most relevant or enjoyable for the user. By comparing individual predicted ratings to these Top-N lists, we can evaluate the consistency and reliability of the CF models. For many users, movies with the highest predicted ratings, such as 92321, 157336, and 550, often appear in the Top-N lists, showing the effectiveness of these predictions in identifying likely preferences. Additionally, some movies like 299534 and 77338 consistently receive high predicted ratings for certain users and appear in Top-N lists, indicating broad appeal across similar user groups in both User-Based and Item-Based models. The User-Based CF model typically recommends items rated highly by similar users, leading to more personalized suggestions, while the Item-Based CF model focuses on movies similar to those a user has previously rated, emphasizing item similarity. Cosine Similarity tends to generate broadly appealing recommendations by aligning general rating patterns, while Pearson Correlation is more tailored to individual user preferences, making it sensitive to specific rating styles. In summary, there is strong alignment between high predicted ratings and items in Top-N recommendations, demonstrating CF models' consistency in leveraging predicted ratings for relevant suggestions. The choice of similarity measure affects personalization: User-Based CF with Pearson Correlation often delivers more tailored recommendations, while Item-Based CF with Cosine Similarity favors universally appealing items.

1. **Implementation Details:**

This assignment was implemented in Python due to its robust ecosystem for data science and machine learning. Several key libraries were used to handle data manipulation, matrix operations, and similarity calculations, which were essential for building the collaborative filtering recommendation system.

* Data Collection: Movie data was collected using the TMDb API (The Movie Database API), which provided movie details, including user ratings. We implemented an API function to retrieve movies with substantial review counts, filtering for popular movies to minimize issues with sparse data and to ensure reliable recommendations. TMDb was chosen for its comprehensive movie database and reliable API, enabling us to work with a realistic dataset.
* Data Preparation and User-Item Matrix: The data was processed and transformed into a 10x10 user-item matrix. Each row represented a user, and each column represented a movie, with the matrix entries containing ratings provided by users. If a user hadn’t rated a movie, the corresponding cell was filled with a zero. Using zero for missing values allowed us to maintain a complete matrix structure for similarity calculations without requiring additional data imputation techniques. This matrix formed the foundation for applying collaborative filtering algorithms.
* Libraries and Tools:

1. Pandas: Used extensively to handle data manipulation tasks, such as filtering, reshaping, and aggregating data from the TMDb API. Pandas DataFrames made it easy to structure the data into the user-item matrix format required for collaborative filtering.
2. NumPy: Essential for performing mathematical operations on arrays and matrices. NumPy was used for matrix manipulations, enabling efficient similarity calculations and element-wise operations on large datasets.
3. Scikit-learn: Provided built-in functions for calculating cosine similarity and Pearson correlation, which were central to this project. These functions allowed us to compute similarity measures efficiently and apply them across all user and item pairs in the matrix.
4. SciPy: Used as an additional support for mathematical and statistical functions, particularly for handling sparse matrices and distance calculations when working with large datasets.

* Similarity Computations:

1. Cosine Similarity: This measure calculates the cosine of the angle between two vectors, treating each vector as a user's (or item’s) ratings across multiple items (or users). Cosine similarity ranges from 0 to 1, with higher values indicating stronger similarity. This metric is useful for identifying users or items with similar rating patterns, even if their rating scales differ. In user-based collaborative filtering, cosine similarity helped find users with shared preferences, enabling personalized recommendations.
2. Pearson Correlation: Pearson correlation measures the linear relationship between rating patterns, accounting for individual biases by centering ratings around each user’s average rating. This measure ranges from -1 to 1, with positive values indicating similar trends. Unlike cosine similarity, Pearson correlation adjusts for differences in users' average ratings, making it particularly effective for more personalized recommendations. This approach was valuable in cases where the direction of user preferences (whether above or below their average) mattered more than the magnitude of ratings.

* Evaluation Metrics: Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) were computed to assess the accuracy of predicted ratings. MAE measured the average absolute difference between predicted and actual ratings, while RMSE added sensitivity to larger errors. Cosine similarity produced a slightly lower MAE, indicating it was more accurate in general rating predictions. However, Pearson correlation provided a more user-specific measure, aligning recommendations closely with individual rating patterns, which can be beneficial for enhancing user satisfaction.

1. **Conclusion and Future Work:**

In this assignment, both cosine similarity and Pearson correlation were used to implement user-based and item-based collaborative filtering models, providing unique insights into how different similarity measures influence recommendation outcomes. Cosine similarity effectively captured general patterns in user and item ratings by focusing on the directional alignment of preferences. This approach provided consistent results in identifying similar users or items without considering the magnitude of individual ratings. As a result, cosine similarity produced slightly lower Mean Absolute Error (MAE), indicating that it offered accurate rating predictions on an average scale and was suitable for broader, general recommendations.

On the other hand, Pearson correlation provided a more tailored approach by adjusting for individual rating biases. By centering each user’s ratings around their mean, Pearson correlation accounted for personal tendencies in rating patterns, such as users who generally rate items higher or lower than others. This allowed the model to focus on relative preferences rather than absolute values, which is beneficial for generating personalized recommendations that align closely with individual user preferences. Although Pearson correlation had a marginally higher MAE compared to cosine similarity, it offered enhanced relevance for each user’s unique preferences, making it a valuable tool for recommendation systems where personalized recommendations are prioritized.

Future Work: Future improvements could explore more sophisticated methods to address current limitations, including:

1. Hybrid Models: Combining collaborative filtering with content-based filtering or matrix factorization methods, such as Singular Value Decomposition (SVD) or Alternating Least Squares (ALS), could enhance the robustness of the recommendation system. Hybrid models can alleviate issues like data sparsity and cold-start problems by incorporating additional information about item attributes (e.g., genres, directors) or user profiles.
2. Incorporating Temporal Dynamics: User preferences often change over time, so integrating time-based data could significantly improve recommendations. Temporal dynamics models, such as time-based collaborative filtering or decay functions, would allow the system to give more weight to recent ratings, reflecting evolving tastes. For example, by weighting newer ratings more heavily, the model could provide recommendations that align with the user’s current interests rather than past preferences.
3. Contextual Awareness: Future iterations could incorporate contextual information, such as location, time of day, or even seasonality, to make recommendations more relevant. For instance, certain genres or movie types may be more popular in specific contexts, and understanding these patterns can further enhance recommendation relevance.
4. Improved Evaluation Metrics: While MAE and RMSE are standard evaluation metrics, incorporating precision, recall, and F1-score would provide a more comprehensive assessment of recommendation quality, especially for ranking-based recommendations. These metrics can better capture the system's ability to identify items users are genuinely interested in.
5. Addressing Data Sparsity and Cold-Start Problems: In cases where users or items have limited rating data, recommendations can be inaccurate or difficult to generate. Addressing these challenges through methods such as transfer learning, active learning (where users are prompted to rate specific items), or semi-supervised learning could lead to improved performance in sparse data scenarios.

this assignment highlighted the strengths and trade-offs of cosine similarity and Pearson correlation in collaborative filtering. Future enhancements could lead to a more dynamic and contextually aware recommendation system that better serves diverse user needs and adapts to changes in user preferences over time.

1. **References:**
2. TMDb API Documentation. (2024). The Movie Database (TMDb). Available: <https://www.themoviedb.org/documentation/api>
3. · "Pandas Documentation," pandas.pydata.org. Available: <https://pandas.pydata.org/docs/>
4. · "Scikit-Learn Documentation," scikit-learn.org. Available: <https://scikit-learn.org/stable/documentation.html>
5. "Scikit-Learn Documentation," scikit-learn.org. Available: <https://scikit-learn.org/stable/documentation.html>
6. "SciPy Documentation," scipy.org. Available: <https://scipy.org/doc/>
7. "Collaborative Filtering - A Short Introduction," Towards Data Science, Available: <https://towardsdatascience.com/collaborative-filtering>