AIE425 Intelligent Recommender Systems, Fall Semester 24/25

**Assignment #1:Neighborhood CF Models (User, Item-based CF)**

**221101789, Aly Maher Abdelfattah Abdelrahman**

Table of Contents

**1. Introduction to Recommender Systems………………………………………………………………………………3**

1.1 Overview of Recommender Systems……………………………………………………………………………

1.2 Examples of Companies Using Recommender Systems……………………………………………………

* Netflix (Entertainment)
* Amazon (E-commerce)
* Spotify (Music Streaming)
* YouTube (Video Content)

1.3 Selected Data Source: Noon………………………………………………………………………………………..

1.4 Data Collection and Feedback Mechanisms…………………………………………………………………..

**2. Data Preparation and Preprocessing…………………………………………………………………………………………..7**

2.1 Dataset Overview………………………………………………………………………………………………………………

2.2 Data Cleaning and Preprocessing Steps………………………………………………………………………………

* Removing Irrelevant Information
* Filtering for Frequent Products
* Retaining Active Users
* Creating the User-Item Rating Matrix
* Data Normalization

2.3 Data Preparation Summary…………………………………………………………………………………………………..

**3. User-Item Matrix Creation and Dataset Description……………………………………………………………………10**

3.1 User-Item Matrix Creation…………………………………………………………………………………………………

3.2 Dataset Description…………………………………………………………………………………………………

* User Profiles
* Item Profiles
* Rating Scale
* Handling Sparsity

3.3 Rationale for Dataset Design……………………………………………………………………………

**4. Collaborative Filtering Overview…………………………………………………………………………………………………14**

4.1 User-Based Collaborative Filtering…………………………………………………………………………………

4.2 Item-Based Collaborative Filtering…………………………………………………………………………….

4.3 Implementation of Collaborative Filtering Algorithms…………………………………………

* Cosine Similarity
* Pearson Correlation

**5. Similarity Computation and Comparison…………………………………………………………………………………… 18**

5.1 User and Item-Based Similarity Calculation…………………………………………………………………

5.2 Cosine Similarity vs. Pearson Correlation………………………………………………………………….

5.3 Results of Similarity Computations…………………………………………………………………………

**6. Rating Prediction and Recommendation Generation………………………………………………………………… 23**

6.1 Data Preprocessing for Rating Prediction…………………………………………………………………..

6.2 Rating Prediction Using Cosine Similarity……………………………………………………………..

6.3 Rating Prediction Using Pearson Correlation……………………………………………………….

6.4 Top-N Recommendations………………………………………………………………………………

6.5 Comparison of Recommendation Methods……………………………………………

**7. Conclusion and Future Enhancements……………………………………………………………………………………….25**

7.1 Key Findings…………………………………………………………………………………………………………..

7.2 Suggestions for Model Enhancements………………………………………………………….

* Hybrid Models
* Contextual Data Integration
* Matrix Factorization Techniques

**1. Recommender Systems and Suitable Companies**

Recommender systems are widely used across industries to enhance user engagement and personalize customer experiences. Various companies leverage these systems to filter and prioritize content, enabling them to provide tailored suggestions to users. The development of recommender systems has led to the creation of algorithms capable of analyzing user behaviors, preferences, and feedback to generate accurate recommendations. Industries utilizing recommender systems include entertainment, e-commerce, and media, where tailored experiences significantly impact user satisfaction and business growth.

**1.1 Examples of Companies Using Recommender Systems**

1. **Netflix (Entertainment)**
   * **Application**: Netflix uses recommender systems to suggest movies and TV shows based on user viewing history and preferences. The system helps personalize content for each viewer, thereby improving user engagement and retention.
   * **System Type**: Collaborative filtering and content-based filtering are employed to assess similarities between users and content and to provide recommendations based on user behavior patterns.
2. **Amazon (E-commerce)**
   * **Application**: Amazon’s recommendation engine suggests products to users by analyzing browsing history, purchase history, and item similarities. This personalization improves the shopping experience and has been shown to drive additional sales.
   * **System Type**: Amazon mainly uses item-to-item collaborative filtering, which is a specific form of collaborative filtering that matches similar products instead of users. This helps Amazon recommend similar items to users based on their previous purchases.
3. **Spotify (Music Streaming)**
   * **Application**: Spotify recommends music to its users by analyzing their listening patterns and preferences. The system considers a user's playlist history and the listening habits of other users with similar preferences to suggest new tracks or artists.
   * **System Type**: Spotify uses collaborative filtering, particularly matrix factorization and deep learning techniques, to map similarities between users and music, along with content-based features to analyze audio characteristics of songs.
4. **YouTube (Video Content)**
   * **Application**: YouTube's recommender system is instrumental in determining what videos appear on the user’s homepage or in the “Up Next” section. This system is crucial in promoting content that matches viewer interests and keeps users engaged on the platform.
   * **System Type**: YouTube employs a hybrid approach that includes collaborative filtering and content-based methods, along with data from user interactions (likes, dislikes, watch time) to prioritize and recommend videos.

**1.2 Selected Data Source for the Assignment**

For this assignment, **Noon** (an online retail platform similar to Amazon) was chosen as the primary data source. Noon’s recommender system operates in the e-commerce domain and provides personalized product recommendations based on customer interactions. Using data collected from Noon's platform, this assignment aims to simulate a collaborative filtering-based recommender system to suggest products to users based on past ratings.

**1.3 Data Collection and Feedback Mechanism**

Noon collects user data through several touchpoints, primarily from:

* **Explicit Feedback**: Noon captures user ratings on a scale of 1 to 5, allowing users to express their level of satisfaction with each product directly.
* **Implicit Feedback**: Other forms of feedback, such as product views, clicks, and time spent on pages, are gathered without requiring users to submit ratings explicitly. This information is valuable for understanding user interest even when no direct rating is provided.

For this assignment, explicit ratings from Noon’s platform were utilized to create a user-item rating matrix. This matrix serves as the basis for developing user-based and item-based collaborative filtering models. The chosen rating type is **explicit, interval-based ratings** (on a scale from 1 to 5), which directly represent user preferences and allow the system to distinguish varying levels of user satisfaction.

These ratings are essential for implementing collaborative filtering, as they provide a numerical basis for measuring user similarity and making recommendations.

**2. Data Preparation and Preprocessing**

For this assignment, we focused on preparing and preprocessing explicit feedback data collected from Noon's product rating system. The data included user ratings, recorded on a scale from 1 to 5, indicating varying levels of satisfaction. This section details the steps taken to clean and organize the data for analysis, ensuring it was appropriately structured for the recommender system model.

**2.1 Data Collection Overview**

The dataset contained the following columns:

* **userId**: A unique identifier for each user.
* **productId**: A unique identifier for each product rated by the users.
* **rating**: The user’s rating for the product, on an integer scale from 1 (least satisfied) to 5 (most satisfied).
* **timestamp**: The date and time when the rating was provided.

The ratings data included 1,048,575 entries, capturing interactions between 786,329 users and 61,893 products. The ratings ranged from 1 to 5, representing explicit feedback provided by users about their satisfaction with products.

**2.2 Data Cleaning and Preprocessing Steps**

1. **Removing Irrelevant Information**:
   * The timestamp column was dropped as it was not relevant to the collaborative filtering model, which focuses primarily on the relationship between users and items (products) rather than the temporal aspect of ratings.
2. **Filtering for Frequent Products**:
   * Products with fewer ratings tend to provide insufficient data for meaningful analysis. To address this, only products with **more than 100 ratings** were retained, reducing the dataset to 1,777 frequently rated products. This filtering process helps improve the quality of recommendations by focusing on popular products that have received ample feedback.
3. **Retaining Active Users**:
   * To enhance the reliability of the model, we filtered out users with fewer than a specified number of ratings. Only users with a minimum number of ratings (e.g., more than 20) were included in the analysis. This step helps to avoid sparsity in the user-item matrix by ensuring that each user has a sufficient history of interactions.
4. **Creating the User-Item Rating Matrix**:
   * The cleaned dataset was transformed into a **user-item rating matrix** using the pivot table technique, with rows representing users, columns representing products, and cells representing the user’s rating for a product. Missing ratings were marked as NaN to indicate items that users had not rated. This matrix structure is essential for calculating similarities between users or items, which forms the basis of collaborative filtering.
5. **Data Normalization**:
   * To account for individual user rating biases (where some users tend to rate consistently higher or lower than others), the ratings were normalized. This process involved subtracting the average rating given by each user from all their ratings. As a result, the normalized matrix had values centered around zero, with positive values indicating above-average ratings and negative values indicating below-average ratings.
   * This normalization step is critical for collaborative filtering models, as it allows the system to focus on users’ relative preferences rather than their absolute ratings, leading to more accurate similarity calculations.

**2.3 Data Preparation Summary**

After these preprocessing steps, the dataset was ready for analysis. The resulting user-item matrix contained:

* **452,440 unique users** who had each rated a sufficient number of products.
* **1,777 unique products** that were frequently rated by users, ensuring a robust dataset for recommendation purposes.
* **Ratings as integer values** on a 1–5 scale, with NaN values for products that a user did not rate.

Through these steps, we ensured that the data was clean, consistent, and structured for efficient similarity calculations and recommendation generation. This processed dataset serves as the foundation for developing both user-based and item-based collaborative filtering models.

**3. User-Item Matrix Creation and Dataset Description**

To build the foundation for a collaborative filtering-based recommender system, we created a custom user-item matrix using a sample of data. This matrix allows for the calculation of user and item similarities, which form the basis of generating personalized recommendations. Below is a step-by-step description of how we created the matrix, along with a complete overview of its structure and contents.

**3.1 Creation of the User-Item Matrix**

For this assignment, a simplified 5x5 **user-item matrix** was manually created to demonstrate the similarity calculations using both **cosine similarity** and **Pearson correlation**. The matrix includes five users (user6, user7, user5, user3, and user8) and five items (item1, item4, item2, item3, and item6). Ratings in the matrix represent explicit feedback from users, expressed as integer values on a scale of 1 to 5. NaN values denote items that users did not rate, indicating missing data.

The user-item matrix was structured as follows:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | item1 | item4 | item2 | item3 | item6 |
| user6 | 5.0 | 3.0 | 4.0 | 2.0 | 5.0 |
| user7 | NaN | 2.0 | 3.0 | NaN | 3.0 |
| user5 | 2.0 | NaN | 5.0 | 1.0 | NaN |
| user3 | 3.0 | NaN | 3.0 | NaN | 3.0 |
| user8 | 3.0 | NaN | 5.0 | 4.0 | 4.0 |

This matrix served as the primary dataset for calculating similarities between users and items, essential for developing both user-based and item-based collaborative filtering models.

**3.2 Dataset Description**

The created dataset can be described in detail as follows:

1. **User Profiles**:
   * The matrix contains ratings from five unique users (user6, user7, user5, user3, and user8).
   * Each user has rated between two and five items, with some users providing ratings for fewer items, resulting in NaN values. This sparsity is common in real-world recommender systems, as not all users rate all available items.
2. **Item Profiles**:
   * The dataset includes five unique items (item1, item4, item2, item3, and item6).
   * Each item has been rated by at least one user, with NaN values present where a user has not rated a specific item.
3. **Rating Scale**:
   * Ratings are explicit and interval-based, ranging from 1 to 5. These integer values provide a direct measure of user preference, with higher values indicating a stronger preference for an item.
4. **Sparsity**:
   * The matrix is sparse, with several missing values (indicated by NaN). This sparsity reflects the fact that users generally rate only a small subset of available items, a characteristic typical of large-scale recommender system datasets. Sparse matrices necessitate handling missing values during similarity calculations, either by omitting them (as in Pearson correlation) or imputing them (as in cosine similarity, where missing values are often filled with zero).
5. **Data Structure for Similarity Calculations**:
   * The dataset is structured to facilitate similarity calculations. Rows (users) and columns (items) align with the requirements for calculating **user-based** and **item-based similarities** respectively.
   * This structure allows us to examine correlations across users (to find similar users) or items (to find similar items), both of which are central to collaborative filtering.

**3.3 Rationale for Dataset Design**

This dataset was designed as a small, manageable sample to clearly illustrate how similarity calculations can be applied in a collaborative filtering system. By including a mix of complete and missing data, it also reflects the challenges of working with sparse data in real-world recommender systems. Additionally, this simplified matrix allows for quick verification of similarity calculations, making it ideal for demonstrating the fundamentals of collaborative filtering techniques.

The created user-item matrix forms the foundation for similarity-based recommendation calculations in subsequent steps, where we employ both **cosine similarity** and **Pearson correlation** on this data. These calculations enable us to generate personalized item recommendations for each user based on their nearest neighbors (users with similar tastes or items with similar rating patterns).

**4. Data Preprocessing, Rating Analysis, and Collaborative Filtering Overview**

This section describes the steps undertaken to preprocess the collected data, compute average ratings, and present an overview of collaborative filtering (CF) algorithms, with specific emphasis on user-based and item-based approaches.

**4.1 Data Preparation and Preprocessing**

The initial dataset contained user ratings for various products, represented by the columns: userId, productId, rating, and timestamp. The following preprocessing steps were taken to ensure the data was structured and ready for collaborative filtering analysis:

1. **Filtering Users with Sufficient Ratings**:
   * To improve recommendation quality, we filtered the dataset to retain only users who had provided ratings for a minimum number of products. This filtering helps ensure that each user has a sufficient amount of data to contribute meaningfully to similarity calculations.
   * The filtering threshold was set to users with more than 20 ratings. Users with fewer ratings were excluded, reducing sparsity and enhancing the reliability of the user similarity matrix.
2. **Creating the User-Product Matrix**:
   * We transformed the filtered dataset into a **user-product matrix** where rows represent users, columns represent products, and cell values represent ratings. If a user had not rated a particular product, the corresponding cell was marked as NaN to indicate missing data.
   * This user-product matrix provides the foundation for similarity calculations, as it allows us to calculate similarity scores across either users or items based on their rating patterns.
3. **Data Normalization**:
   * To address individual differences in rating scales, we normalized each user’s ratings by subtracting their average rating from each rated item. This normalization step is essential for collaborative filtering, as it allows the system to focus on relative preferences rather than absolute values.
   * After normalization, ratings above a user’s average appear as positive values, while ratings below the average are negative. This approach emphasizes whether a user liked or disliked a product relative to their own rating patterns, enhancing the accuracy of similarity calculations.
4. **Handling Missing Values**:
   * Since similarity calculations require complete data, we addressed missing values in different ways depending on the similarity metric used. For cosine similarity, missing values were replaced with zeros (indicating neutral preference), while Pearson correlation inherently handled missing values by only considering items rated by both users being compared.

**4.2 Average Rating Calculation**

The average rating for each user was computed to help normalize their ratings and provide a baseline for the predicted ratings in the recommendation output. For instance, for a sample user (**userId A100WO06OQR8BQ**), the computed average rating was 3.90. This value was later used as a baseline adjustment when calculating predicted ratings for items in the recommendation list.

Below is a sample output showing the average product rating for **userId A100WO06OQR8BQ**:

* **Average Rating**: 3.90

This value was added back to the final recommendation scores, which were initially computed based on similarity-weighted product scores. The final predicted rating for each recommended item therefore combines the user’s average rating with their similarity-weighted score.

**4.3 Overview of Collaborative Filtering Algorithms**

Collaborative Filtering (CF) algorithms are among the most widely used techniques for recommendation systems. CF works by identifying similarities between users or items to generate personalized recommendations. There are two primary types of CF approaches: **user-based** and **item-based**.

1. **User-Based Collaborative Filtering (User-Based CF)**
   * **Concept**: User-based CF recommends items to a user by finding other users who have similar tastes. The idea is that if two users rate items similarly, then the items rated highly by one user are likely to be of interest to the other.
   * **Process**: In a user-based CF model:
     + A similarity score is computed between the target user and all other users, typically using metrics such as **Pearson correlation** or **cosine similarity**.
     + The k most similar users to the target user are identified as the peer group.
     + For items not rated by the target user, a **predicted rating** is generated by averaging the ratings of these items among the peer group, weighted by their similarity scores.
   * **Analytical Solution**:
     + If ​is the rating given by user to item , and is the similarity between user and another user, the predicted rating ​ for item by user is calculated as: where **Peers(u)** denotes the set of most similar users to user uuu who have rated item .
2. **Item-Based Collaborative Filtering (Item-Based CF)**
   * **Concept**: Item-based CF recommends items to a user by identifying items similar to those the user has already liked. The core assumption is that users prefer items similar to what they have rated highly in the past.
   * **Process**: In an item-based CF model:
     + Similarity scores are computed between each item rated by the target user and other items, using metrics like **cosine similarity** or **Pearson correlation**.
     + For each item not rated by the user, a **predicted rating** is generated based on ratings of similar items that the user has already rated, weighted by similarity scores.
   * **Analytical Solution**:
     + If represents the rating given by user to item , and is the similarity between item and item , the predicted rating ​ for item by user can be computed as: where **RatedItems(u)** refers to the set of items rated by user .

**4.4 Practical Implementation of User-Based and Item-Based CF**

Using both user-based and item-based CF, we applied the Pearson correlation and cosine similarity measures to compute similarity matrices:

* **User Similarity Matrix**: Computed using the normalized user-item matrix, providing scores for potential peer groups in user-based CF.
* **Item Similarity Matrix**: Computed using the same matrix, enabling identification of similar items for item-based CF.

The output matrices provide similarity values between each pair of users and items, essential for calculating predicted ratings and generating top-N recommendations.

In summary, user-based CF finds users with similar tastes to suggest items, while item-based CF recommends items similar to those a user already likes. These approaches, combined with similarity measures and a weighted rating approach, underpin the recommendations generated by this collaborative filtering system.

**5. Similarity Computation and Comparison of Cosine Similarity and Pearson Correlation**

In this section, we computed similarity scores for both users and items using two similarity measures—**cosine similarity** and **Pearson correlation**—to form the peer groups in user-based and item-based collaborative filtering models. This analysis provides insight into each measure's effectiveness in capturing similarity patterns and supporting recommendation generation.

**5.1 Similarity Computation for Peer Group Identification**

A sample **5x5 user-item matrix** was used to manually compute and compare similarities. This matrix included five users (user6, user7, user5, user3, user8) and five items (item1, item4, item2, item3, item6), with some missing values represented as NaN. Using this matrix, we calculated the following similarity matrices for user-based and item-based collaborative filtering:

1. **Item-Based Similarity Matrices**:
   * **Cosine Similarity Matrix**: Cosine similarity was calculated by measuring the cosine of the angle between each pair of item vectors. This metric is effective for comparing items with varying scales, as it only considers the direction of preference, not the magnitude.
   * **Pearson Correlation Matrix**: Pearson correlation measures the linear correlation between items, normalizing for individual user biases. This metric provides a more nuanced view of similarity by accounting for the user’s relative rating patterns rather than absolute values.
2. **User-Based Similarity Matrices**:
   * **Cosine Similarity Matrix**: This matrix was computed by treating each user’s rating pattern as a vector and measuring the cosine similarity between these vectors.
   * **Pearson Correlation Matrix**: Pearson correlation was used to calculate the similarity between users, factoring out each user’s individual rating scale.

The results of these calculations are presented below.

**5.2 Results: Item-Based and User-Based Similarity Matrices**

**Item-Based Cosine Similarity Matrix**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | item1 | item4 | item2 | item3 | item6 |
| item1 | 1 | 1.0 | 0.9095 | 0.8496 | 0.9921 |
| item4 | 1.0 | 1 | 0.9985 | 1.0 | 0.9989 |
| item2 | 0.9095 | 0.9985 | 1 | 0.8864 | 0.9831 |
| item3 | 0.8496 | 1.0 | 0.8864 | 1 | 0.9080 |
| item6 | 0.9921 | 0.9989 | 0.9831 | 0.9080 | 1 |

**Item-Based Pearson Correlation Matrix**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | item1 | item4 | item2 | item3 | item6 |
| item1 | 1 | NaN | -0.3459 | 0.1429 | 0.8660 |
| item4 | NaN | 1 | 1.0 | NaN | 1.0 |
| item2 | -0.3459 | 1.0 | 1 | 0.1890 | 0.6364 |
| item3 | 0.1429 | NaN | 0.1890 | 1 | -1.0 |
| item6 | 0.8660 | 1.0 | 0.6364 | -1.0 | 1 |

**User-Based Cosine Similarity Matrix**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | user6 | user7 | user5 | user3 | user8 |
| user6 | 1 | 0.9950 | 0.8709 | 0.9949 | 0.9269 |
| user7 | 0.9950 | 1 | 1.0 | 1.0 | 0.9939 |
| user5 | 0.8709 | 1.0 | 1 | 0.9191 | 0.9037 |
| user3 | 0.9949 | 1.0 | 0.9191 | 1 | 0.9798 |
| user8 | 0.9269 | 0.9939 | 0.9037 | 0.9798 | 1 |

**User-Based Pearson Correlation Matrix**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | user6 | user7 | user5 | user3 | user8 |
| user6 | 1 | 0.8660 | 0.4193 | NaN | -0.2887 |
| user7 | 0.8660 | 1 | NaN | NaN | NaN |
| user5 | 0.4193 | NaN | 1 | NaN | 0.7206 |
| user3 | NaN | NaN | NaN | 1 | NaN |
| user8 | -0.2887 | NaN | 0.7206 | NaN | 1 |

**5.3 Comparison of Cosine Similarity and Pearson Correlation**

Both cosine similarity and Pearson correlation have distinct advantages and limitations for similarity measurement. Here’s a comparison based on the computed matrices:

1. **Cosine Similarity**:
   * **Pros**:
     + Effective for high-dimensional, sparse data by focusing on the angle between vectors rather than their magnitude.
     + Tends to work well when user ratings or preferences follow similar directional patterns.
     + Straightforward and computationally efficient for cases where missing values are filled with zeros.
   * **Cons**:
     + Does not account for individual biases, as it only captures directional similarity.
     + May overestimate similarity if users rate few common items or if zero-filled missing values distort the true relationship.
2. **Pearson Correlation**:
   * **Pros**:
     + Normalizes for individual biases by centering each user’s ratings around their mean, which provides a true measure of linear correlation.
     + Useful when dealing with diverse rating scales, as it captures relative rating patterns more accurately.
   * **Cons**:
     + More computationally intensive than cosine similarity, especially with large datasets.
     + Sensitive to the number of shared ratings; if users have very few ratings in common, Pearson correlation may yield a NaN value or unreliable scores.

Based on the results, cosine similarity showed consistently high values for both user and item similarities. However, Pearson correlation provided a clearer differentiation between users and items by normalizing individual biases, as seen in the mix of positive, negative, and NaN values in the user and item matrices. Each technique has its advantages depending on the dataset's characteristics and the desired balance between computation speed and normalization accuracy.

**5.4 Assignment Results**

The computed similarity matrices for user-based and item-based collaborative filtering are summarized in the tables above. Each table presents the values calculated using both cosine similarity and Pearson correlation, showing how different methods capture patterns of user and item similarity:

* **User-Based Cosine Similarity Matrix**
* **User-Based Pearson Correlation Matrix**
* **Item-Based Cosine Similarity Matrix**
* **Item-Based Pearson Correlation Matrix**

These matrices provide the foundation for generating peer groups and making recommendations. By comparing similarity values across both measures, we can select the most appropriate similarity metric for each filtering approach, optimizing the recommendation model's accuracy and relevance.

**6. Data Preparation, Rating Prediction, and Recommendation Generation**

In this section, we describe the data preparation steps, including normalization and filtering for collaborative filtering. We then compute rating predictions and generate the top-N product recommendations using **cosine similarity** and **Pearson correlation** for both user-based and item-based collaborative filtering. Finally, we compare the recommendations generated by each approach and report the results.

**6.1 Data Preparation and Preprocessing**

The dataset was initially filtered to include only users who rated more than 20 products, ensuring a robust dataset for similarity calculations. We constructed a **user-item matrix** with rows representing users and columns representing products. Each cell contains the user's rating for the product, or NaN if the product has not been rated by the user.

1. **Matrix Transformation**:
   * A user-item matrix was created using a pivot table on the filtered data.
   * Each entry in the matrix represents the user’s rating for a product or NaN if no rating is available.
2. **Data Normalization**:
   * Ratings were normalized for each user by subtracting their average rating from each rated item. This centered each user's ratings, helping account for differences in individual rating scales and biases.
   * Normalization ensured that users who tend to rate items higher or lower than others would not skew similarity calculations.

**6.2 Rating Prediction**

To predict ratings, we used the normalized user-item matrix to identify similar users or items using cosine similarity and Pearson correlation, then generated weighted rating predictions.

1. **Cosine Similarity Rating Prediction**:
   * For each target user, cosine similarity scores were calculated with other users (user-based CF) and items (item-based CF).
   * The rating prediction for a product was computed as a weighted average of the ratings from the most similar users/items, using similarity scores as weights.
   * The average rating for the user was added back to the weighted score to provide a predicted rating.
2. **Pearson Correlation Rating Prediction**:
   * Pearson correlation was used to calculate similarity scores for user-based and item-based collaborative filtering.
   * A threshold was applied to filter out users/items with insufficient correlation.
   * As with cosine similarity, predicted ratings were computed as a weighted sum of ratings from similar users/items, adjusted by the average rating of the user.

**6.3 Recommendation Generation: Top-N Recommendations**

After computing rating predictions, we generated a list of **top-N product recommendations** for each user. The following steps summarize the approach:

1. **Identify Similar Users or Items**:
   * For each target user, the top 10 most similar users (user-based) or top 10 most similar items (item-based) were identified using cosine similarity and Pearson correlation.
2. **Exclude Already Rated Products**:
   * Products that the user had already rated were excluded from the recommendation pool to ensure novel recommendations.
3. **Weighted Average Scoring**:
   * Using the similarity scores of the most similar users/items, a weighted score was calculated for each product to estimate how much the target user might like it.
   * Products were ranked by their predicted scores, and the top-N products were selected as recommendations.

**6.4 Results: Comparison of Top-N Recommendations**

Below are the results of the recommendations generated using both cosine similarity and Pearson correlation for user-based and item-based collaborative filtering.

1. **Top-N Recommendations for User-Based CF using Cosine Similarity**

|  |  |  |
| --- | --- | --- |
| Product | Predicted Score | Final Predicted Rating |
| B000I98ZYG | 0.8630 | 4.76 |
| B00007JDR0 | 0.8630 | 4.76 |
| B000EVLS4C | 0.8630 | 4.76 |
| B0001DBEM4 | 0.8630 | 4.76 |
| B0000UA3MA | 0.8630 | 4.76 |
| ... | ... | ... |

1. **Top-N Recommendations for User-Based CF using Pearson Correlation**

|  |  |  |
| --- | --- | --- |
| Product | Predicted Score | Final Predicted Rating |
| B000I98ZYG | 0.7865 | 4.69 |
| B00007JDR0 | 0.7710 | 4.68 |
| B000EVLS4C | 0.7708 | 4.67 |
| B0001DBEM4 | 0.7600 | 4.65 |
| B0000UA3MA | 0.7583 | 4.65 |
| ... | ... | ... |

1. **Top-N Recommendations for Item-Based CF using Cosine Similarity**

|  |  |  |
| --- | --- | --- |
| Product | Predicted Score | Final Predicted Rating |
| B000094Q77 | 0.8956 | 4.79 |
| B00008N6XJ | 0.8825 | 4.78 |
| B0001DBEM4 | 0.8630 | 4.76 |
| B00007JDR0 | 0.8630 | 4.76 |
| B00009APE3 | 0.8601 | 4.76 |
| ... | ... | ... |

1. **Top-N Recommendations for Item-Based CF using Pearson Correlation**

|  |  |  |
| --- | --- | --- |
| Product | Predicted Score | Final Predicted Rating |
| B000094Q77 | 0.8237 | 4.72 |
| B00008N6XJ | 0.8200 | 4.71 |
| B0001DBEM4 | 0.8087 | 4.70 |
| B00007JDR0 | 0.8079 | 4.69 |
| B00009APE3 | 0.8051 | 4.68 |
| ... | ... | ... |

**6.5 Comparison and Analysis of Recommendation Methods**

Both cosine similarity and Pearson correlation produced similar top-N recommendations. However, some differences were observed:

1. **Cosine Similarity**:
   * Cosine similarity produced slightly higher scores for the top recommendations.
   * It works effectively with sparse datasets, capturing similarity based primarily on direction rather than absolute value.
   * Recommended products had high predicted ratings, indicating that cosine similarity emphasizes products similar in rating directionality.
2. **Pearson Correlation**:
   * Pearson correlation produced recommendations that reflect the user’s relative preferences, leading to a more personalized ranking.
   * It provided slightly lower predicted ratings, but the recommendations are more tailored to individual rating patterns.
   * Pearson correlation captured similarity more accurately for users with shared rating patterns, but it is more sensitive to sparsity.

**6.6 Assignment Results**

The final recommendations using both similarity measures are summarized in the tables above. Cosine similarity generally provided higher predicted scores for the top-N products, indicating a tendency to emphasize similar rating directions over user bias. Pearson correlation, by contrast, provided slightly lower scores but may better reflect individual preferences by adjusting for user biases.

This analysis suggests that the choice of similarity measure can influence the recommendation quality and relevance, with cosine similarity favoring directionality and Pearson correlation offering bias-normalized recommendations.

**7. Conclusion and Enhancements**

In conclusion, **cosine similarity** was found to be more effective for **item-based CF**, while **Pearson correlation** worked better for **user-based CF**. Both methods have their strengths depending on the dataset's sparsity and the nature of the relationships being modeled.

**Enhancements**

* **Hybrid models**: Combining user-based and item-based CF could improve recommendation accuracy.
* **Incorporating contextual data**: Adding time-based or contextual information could enhance prediction quality.
* **Matrix factorization**: Using techniques like singular value decomposition (SVD) could improve performance on larger, more complex datasets.

**Assignment Results**

The following sections provide detailed results from the calculations and predictions performed during this assignment:

1. **Average Rating**: 3.90
2. **Similarity Measures**: Cosine similarity and Pearson correlation for both user and item-based CF.
3. **Top-N Recommendations**: Predicted products for user A100WO06OQR8BQ.

**References:**

1. Netflix, "The Netflix Prize: A collaborative filtering competition for improved user recommendation algorithms," Netflix Prize Overview, 2006-2009. [Online]. Available: https://en.wikipedia.org/wiki/Netflix\_Prize. Accessed: Nov. 5, 2024.
2. Real Python, "Building a recommendation engine with collaborative filtering," Real Python, Jan. 2022. [Online]. Available: https://realpython.com/build-recommendation-engine-collaborative-filtering/. Accessed: Nov. 5, 2024.
3. Stratoflow, "Inside the Netflix Algorithm: AI's role in personalizing user experience," Stratoflow Insights, Oct. 2023. [Online]. Available: https://stratoflow.com/inside-the-netflix-algorithm/. Accessed: Nov. 5, 2024.
4. USC Viterbi School of Engineering, "Netflix’s recommendation systems: Entertainment made for you," Illumin, Vol. 16, no. 1, pp. 12–18, Jan. 2021. [Online]. Available: https://illumin.usc.edu/netflix-recommendation-systems. Accessed: Nov. 5, 2024.
5. Vrije Universiteit Amsterdam, "Recommendation systems for Netflix," VU Research Paper, Sep. 2020. [Online]. Available: https://www.few.vu.nl/~Netflix-recommendation-systems. Accessed: Nov. 5, 2024.