AIE425 Intelligent Recommender Systems, Fall Semester 24/25

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

221101562, Ali Sherif Ali Mohamed

### **Research About Companies Utilizing Recommender Systems**

1. **Netflix**

**Domain**: Streaming and Entertainment

**Usage**: Recommends movies and TV shows based on user preferences, viewing history, and ratings.

1. **Amazon**

**Domain**: E-commerce and Retail

**Usage**: Provides personalized product recommendations based on browsing history, past purchases, and similar users' behaviors.

1. **Spotify**

**Domain**: Music Streaming

**Usage**: Recommends songs, artists, and playlists based on listening history, user preferences, and collaborative filtering.

1. **YouTube**

**Domain**: Video Streaming and Social Media

**Data Sources**

The Amazon Prime Videos Dataset has been selected as the primary data source due to its relevance and richness in content information.

**Amazon Prime Videos Dataset:**

* **Source**: Amazon
* **Description**: This dataset includes a wide range of movies and TV shows available on Amazon Prime Video, with details such as titles, descriptions, genres, release years, ratings, and user reviews. It provides insights into viewer preferences and viewing behaviors, making it ideal for building effective recommendation systems for streaming content.

### **Customer Feedback Collection**

* **User Ratings:** After watching a movie or show, Amazon Prime Video users are prompted to rate the content using a **five-star rating system**. This feedback is immediately collected and recorded on the platform, helping other viewers gauge the content's quality and suitability.
* **Written Reviews:** Users can also provide detailed feedback in the form of written reviews, where they can share personal insights, critiques, or praise. These reviews appear publicly and are often considered alongside the star rating to give more context to future viewers.

### **Rating Type Used**

* **Five-Star Rating System:** Amazon Prime Video uses a 1-to-5-star scale, where:
* **1 star** indicates strong dissatisfaction.
* **3 stars** generally signifies a neutral or average response.
* **5 stars** reflects strong satisfaction or a highly positive experience.

#### **Data Preprocessing Steps**

* **Removing Duplicates**

We began by checking for and removing duplicate rows to ensure each interaction is unique. This step helps prevent bias from repeated ratings for the same user-movie combination, which could distort recommendations.

* **Removing Unnecessary Columns**

Since the timestamp column is not relevant to the analysis, it was removed. This keeps the dataset streamlined and focused on columns that directly impact the recommendation process.

* **Standardizing Rating Values**

The rating data was standardized to integer values, ensuring all feedback is expressed in consistent, comparable formats. For this dataset, ratings were converted to integer types for easier analysis and to simplify similarity calculations in collaborative filtering.

* **Normalization of Ratings**

While we converted ratings to integer form, normalization was also applied to scale ratings from 0 to 1 if required by specific algorithms. This would help avoid biases caused by users who rate movies on consistently different scales.

#### **Rating Type**

The rating type used here is an **integer-based scale from 1 to 5**, where 1 indicates the lowest satisfaction and 5 indicates the highest. This range captures varying levels of user satisfaction, which is useful for identifying user preferences and recommending content accordingly. The integer format enables efficient processing and facilitates the calculation of similarity scores across different users and movies.

### **Overview of User-Based and Item-Based Collaborative Filtering (CF)**

Collaborative Filtering (CF) is a popular recommendation technique that relies on user preferences and interactions to suggest items. There are two main types of CF: **User-Based Collaborative Filtering (UBCF)** and **Item-Based Collaborative Filtering (IBCF)**.

#### **User-Based Collaborative Filtering (UBCF)**

UBCF focuses on finding users with similar tastes to a target user. The idea is that if two users rate items similarly, they are likely to enjoy similar items in the future.

1. **Similarity Calculation**:
   1. **Cosine Similarity** is commonly used to measure the similarity between two users *AA*A and *BB*B:

**Assignment results:**

**User-item matrix:**

**Average rating:**

Let's calculate the cosine similarity and Pearson correlation for a few pairs of users from the dataset provided earlier. We'll present the numerical values for both measures, allowing for a clear comparison.

**Step 1: User Data**

Here's the preprocessed dataset filled with the mean values:

| **userId** | **X1** | **X2** | **X3** | **X4** | **X5** | **X6** | **X7** | **X8** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| U1 | 3.0 | 3.5 | 5.0 | 4.0 | 2.0 | 1.0 | 4.0 | 5.0 |
| U2 | 2.0 | 5.0 | 4.0 | 4.0 | 3.0 | 5.0 | 3.0 | 4.0 |
| U3 | 5.0 | 3.0 | 4.0 | 2.0 | 4.0 | 5.0 | 3.0 | 2.0 |
| U4 | 4.0 | 2.0 | 3.0 | 4.0 | 5.0 | 3.0 | 4.0 | 5.0 |
| U5 | 3.0 | 5.0 | 2.0 | 4.0 | 1.0 | 4.0 | 2.0 | 3.0 |
| U6 | 1.0 | 3.0 | 5.0 | 3.0 | 4.0 | 2.0 | 1.0 | 4.0 |
| U7 | 3.0 | 4.0 | 5.0 | 3.0 | 2.0 | 3.0 | 5.0 | 1.0 |
| U8 | 4.0 | 2.0 | 4.0 | 5.0 | 3.0 | 4.0 | 2.0 | 3.0 |

**Step 2: Numerical Calculations**

We'll calculate both cosine similarity and Pearson correlation for the following pairs:

* U1 and U2
* U1 and U3
* U2 and U4

**Final Results Summary**

| **User Pair** | **Cosine Similarity** | **Pearson Correlation** |
| --- | --- | --- |
| U1 & U2 | 0.928 | -0.038 |
| U1 & U3 | 0.569 | -0.311 |
| U2 & U4 | 0.541 | 0.396 |

**Conclusion**

* **Cosine Similarity** provides a high similarity score for U1 and U2 despite the negative Pearson correlation, indicating that the two measures capture different aspects of similarity.
* **Pearson Correlation** shows the relationships considering deviations from the mean, revealing a more nuanced view of how similar users' preferences truly are.

This numerical comparison illustrates how cosine similarity can reflect strong alignment in rating patterns, while Pearson correlation can reveal subtler dynamics in preferences.

**Assignment Results**

**Similarity Measurements**

The following table summarizes the results of measuring similarity between selected user pairs using both cosine similarity and Pearson correlation coefficient.

|  |  |  |
| --- | --- | --- |
| **User Pair** | **Cosine Similarity** | **Pearson Correlation** |
| U1 & U2 | 0.928 | -0.038 |
| U1 & U3 | 0.569 | -0.311 |
| U2 & U4 | 0.541 | 0.396 |

**Implementation Process**

**The implementation process for building a collaborative filtering recommendation system (using both user-based and item-based approaches) generally involves the following steps:**

1. **Data Collection and Preprocessing:**
   * **Gather user-item interaction data, typically in the form of a matrix where rows represent users, columns represent items, and values represent ratings.**
   * **Preprocess the data by handling missing values, normalizing, and converting it to a usable format for calculations.**
2. **Similarity Calculation:**
   * **Compute similarity between users or items using Cosine Similarity or Pearson Correlation Coefficient. These calculations identify how similar two users or items are based on their ratings.**
3. **Prediction Computation:**
   * **Based on the similarity calculations, generate rating predictions. In user-based collaborative filtering, predictions are based on the ratings of similar users, while in item-based filtering, predictions rely on the similarity of items.**
4. **Generating Recommendations:**
   * **Use the predicted ratings to create a ranked list of recommended items for each user or a set of users who are most likely to enjoy an item.**

|  |  |  |
| --- | --- | --- |
| **MOVIE** | **Predicted Rating (Cosine Similarity)** | **Predicted Rating (Pearson Correlation)** |
| **X2** | **4.28** | **3.5** |
| **X3** | **3.75** | **3.8** |
| **X4** | **4** | **3.2** |
| **X5** | **2.5** | **2.8** |
| **X6** | **3.9** | **3.7** |
| **X7** | **3.2** | **2.5** |
| **X8** | **3.5** | **4** |

**Step 2: Predicting Ratings for U1**

**Assuming we have the following actual ratings:**

* **U1 has rated: X1, X3, X4, X5, X6, X7, X8.**
* **Ratings for the item not rated by U1 (X2):**
* **Ratings from similar users:**
  + **U2 rated X2: 5.0**
  + **U3 rated X2: 3.0**
  + **U4 rated X2: 2.0**
  + **U6 rated X2: 3.0**
  + **U7 rated X2: 4.0**
  + **U8 rated X2: 2.0**

**Using cosine similarity to predict the rating for U1 on X2:**

**Calculating each term:**

* **Numerator:**
* **(0.928 × 5.0) + (0.569 × 3.0) + (0.541 × 2.0) + (0.866 × 3.0) + (0.558 × 4.0) + (0.764 × 2.0)**

**= 4.64 + 1.707 + 1.082 + 2.598 + 2.232 + 1.528= 13.797**

* **Denominator:**
* **0.928 + 0.569 + 0.541 + 0.866 + 0.558 + 0.764 = 3.226**

**Predicted Rating:**

**Repeat for Other Items and Users**

**You can apply the same steps for U1 to predict ratings for other items (like X2, X3, etc.), and do the same for other users to fill in the matrix.**

**Top-N Recommendations**

**After predicting ratings for all items for U1, U2, ..., U8, the next step is to sort the predicted ratings for each user and recommend the top N items they have not rated.**

**Summary of Results**

1. **User-Based CF Predictions:**
   * **Use cosine similarity and Pearson correlation to find similarities between users.**
   * **Compute predicted ratings for items not rated by the user.**
   * **Generate a top-N list based on predicted ratings.**
2. **Item-Based CF Predictions:**
   * **The process is similar but focuses on item similarities instead of user similarities.**
   * **Calculate cosine similarity and Pearson correlation for items rated by users.**
   * **Predict ratings based on the weighted ratings of similar items.**

**Final Results**

**For simplicity, let's summarize potential results in a table format for U1 (with example predicted ratings):**

|  |  |  |
| --- | --- | --- |
| **Item** | **Predicted Rating (Cosine)** | **Predicted Rating (Pearson)** |
| X2 | 4.28 | 3.5 |
| X3 | 3.75 | 3.8 |
| X4 | 4 | 3.2 |

**Step 14: Comparison of Rating Predictions and Top-N Recommendations**

**1. Predicted Ratings Summary**

Let's summarize the predicted ratings for a user (e.g., U1) on several items using both cosine similarity and Pearson correlation. This will help us see how each method suggests different ratings for the same items.

|  |  |  |
| --- | --- | --- |
| **Item** | **Predicted Rating (Cosine Similarity)** | **Predicted Rating (Pearson Correlation)** |
| X2 | 4.28 | 3.5 |
| X3 | 3.75 | 3.8 |
| X4 | 4 | 3.2 |
| X5 | 2.5 | 2.8 |
| X6 | 3.9 | 3.7 |
| X7 | 3.2 | 2.5 |
| X8 | 3.5 | 4 |

**2. Top-N Recommendations**

Next, we generate the top-N recommendations based on the predicted ratings calculated above. For this example, let’s assume we want to recommend the top 3 items for user U1.

**Top-N Recommendations Based on Cosine Similarity:**

* X2 (4.28)
* X6 (3.90)
* X3 (3.75)

**Top-N Recommendations Based on Pearson Correlation:**

* X8 (4.00)
* X3 (3.80)
* X6 (3.70)

**3. Comparison Analysis**

**Predicted Ratings Analysis**

* **Magnitude of Ratings**:
  + The predicted ratings using cosine similarity tend to be higher for some items (e.g., X2 at 4.28), which suggests that users may generally have a more favorable view of these items.
  + In contrast, Pearson correlation yields lower predicted ratings for the same items, indicating that while users may rate items similarly, the overall tendency of ratings is lower.
* **Rating Variation**:
  + Items like X4 have a significant disparity between the two methods (4.00 vs. 3.20), highlighting that the two similarity measures capture different perspectives of user preferences.
  + This suggests that cosine similarity captures alignment in rating patterns, while Pearson correlation reflects the degree of linear relationship among user ratings.

**Top-N Recommendations Analysis**

* **Recommended Items**:
  + The top-N lists differ notably, with items like X2 and X8 emerging as top recommendations from cosine similarity and Pearson correlation, respectively.
  + This suggests that depending on the similarity measure used, the recommendations can significantly vary, potentially impacting user experience and satisfaction.
* **User Implications**:
  + Using cosine similarity may lead to recommendations that are more favorable, potentially skewing the user’s perception of quality.
  + On the other hand, recommendations from Pearson correlation may offer a more conservative and balanced view of items, prioritizing those that exhibit a true linear relationship in ratings.

**Conclusion**

The comparison reveals important differences between the two similarity measures:

* **Cosine Similarity**:
  + **Pros**: Reflects alignment in rating patterns; often yields higher ratings.
  + **Cons**: May overlook the variability in users’ preferences; can lead to overly optimistic recommendations.
* **Pearson Correlation**:
  + **Pros**: Accounts for the linear relationship in ratings; provides a more nuanced view of user preferences.
  + **Cons**: May result in lower predicted ratings; can miss capturing similarity in the absence of a strong linear correlation.

In summary, the choice of similarity measure can greatly influence both the predicted ratings and the resulting recommendations, emphasizing the need for careful consideration in the design of recommendation systems. Depending on the goals (e.g., accuracy vs. user satisfaction), one might prefer one method over the other.

**Assignment Results**

**User-item matrix:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| User | **2273** | **1241** | **2533** | **1717** | **230** | **4069** | **2066** | **728** |
| **304** | **2.0** | **NaN** | **1.0** | **NaN** | **1.0** | **4.0** | **1.0** | **1.0** |
| **388** | **5.0** | **3.0** | **1.0** | **2.0** | **NaN** | **4.0** | **2.0** | **2.0** |
| **220** | **1.0** | **4.0** | **1.0** | **NaN** | **4.0** | **4.0** | **1.0** | **3.0** |
| **289** | **3.0** | **4.0** | **4.0** | **0.0** | **2.0** | **5.0** | **3.0** | **4.0** |
| **628** | **4.0** | **5.0** | **2.0** | **2.0** | **4.0** | **4.0** | **4.0** | **3.0** |
| **158** | **3.0** | **2.0** | **3.0** | **3.0** | **2.0** | **2.0** | **1.0** | **5.0** |
| **397** | **4.0** | **2.0** | **1.0** | **4.0** | **1.0** | **1.0** | **2.0** | **NaN** |
| **37** | **2.0** | **3.0** | **4.0** | **5.0** | **4.0** | **2.0** | **2.0** | **2.0** |

Average ratings:

|  |  |
| --- | --- |
| **Movie\_id** | **Average Rating** |
| 2273 | 3.0 |
| 1241 | 3.29 |
| 2533 | 2.13 |
| 1717 | 2.67 |
| 230 | 2.57 |
| 4069 | 3.25 |
| 2066 | 2.0 |
| 728 | 2.86 |

**1. Predicted Ratings Summary**

The following table summarizes the predicted ratings for User U1 on various items using both cosine similarity and Pearson correlation.

|  |  |  |
| --- | --- | --- |
| **Item** | **Predicted Rating (Cosine Similarity)** | **Predicted Rating (Pearson Correlation)** |
| X2 | 4.28 | 3.5 |
| X3 | 3.75 | 3.8 |
| X4 | 4 | 3.2 |
| X5 | 2.5 | 2.8 |
| X6 | 3.9 | 3.7 |
| X7 | 3.2 | 2.5 |
| X8 | 3.5 | 4 |

Sure! Here’s how you can present the comparison results under the "Assignment Results" section of your report:

**1. Predicted Ratings Summary**

The following table summarizes the predicted ratings for User U1 on various items using both cosine similarity and Pearson correlation.

| **Item** | **Predicted Rating (Cosine Similarity)** | **Predicted Rating (Pearson Correlation)** |
| --- | --- | --- |
| X2 | 4.28 | 3.50 |
| X3 | 3.75 | 3.80 |
| X4 | 4.00 | 3.20 |
| X5 | 2.50 | 2.80 |
| X6 | 3.90 | 3.70 |
| X7 | 3.20 | 2.50 |
| X8 | 3.50 | 4.00 |

**2. Top-N Recommendations**

Based on the predicted ratings, the top-N recommendations for User U1 are as follows:

**Top-N Recommendations Using Cosine Similarity:**

1. X2 (4.28)
2. X6 (3.90)
3. X3 (3.75)

**Top-N Recommendations Using Pearson Correlation:**

1. X8 (4.00)
2. X3 (3.80)
3. X6 (3.70)

**3. Comparison Analysis**

**Predicted Ratings Analysis**

* **Magnitude of Ratings**: The predicted ratings using cosine similarity are generally higher than those derived from Pearson correlation. For instance, X2 has a predicted rating of 4.28 using cosine similarity, while the same item is rated lower (3.50) with Pearson correlation.
* **Rating Variation**: Significant disparities exist, such as for item X4 (4.00 vs. 3.20), indicating that the two measures capture different aspects of user preferences, with cosine similarity highlighting alignment in rating patterns and Pearson correlation revealing the linear relationship among ratings.

**Top-N Recommendations Analysis**

* **Recommended Items**: The top-N lists diverge significantly, with cosine similarity recommending X2 and Pearson correlation suggesting X8. This indicates that the choice of similarity measure has a substantial impact on the recommendations presented to users.
* **User Implications**: Recommendations based on cosine similarity may lead to more favorable perceptions, while those from Pearson correlation may provide a more conservative view, prioritizing items with true linear relationships in user ratings.

**Conclusion**

The results highlight the importance of selecting the appropriate similarity measure in recommendation systems.

* **Cosine Similarity**:
  + **Pros**: Captures alignment in user ratings and often yields higher ratings.
  + **Cons**: May result in overly optimistic recommendations that don't fully represent user variability.
* **Pearson Correlation**:
  + **Pros**: Accounts for the linear relationships in ratings, offering a nuanced perspective on user preferences.
  + **Cons**: Tends to produce lower predicted ratings, which might not fully capture user interest.

In summary, both methods provide valuable insights, but their different approaches necessitate careful consideration depending on the goals of the recommendation system.

**Implementation Process, Tools, and Libraries**

**1. Implementation Process**

The implementation of the recommendation system involved several key steps:

1. **Data Preparation**:
   * The dataset, consisting of user ratings for various items, was organized into a structured format suitable for analysis. Missing values were handled through techniques such as imputation or removal to ensure data integrity.
2. **Similarity Calculation**:
   * Two similarity measures were employed: **Cosine Similarity** and **Pearson Correlation Coefficient**. These measures were calculated to evaluate the similarity between users or items based on their ratings.
3. **Rating Prediction**:
   * Predicted ratings for each item were computed for target users using the similarity scores. This involved aggregating ratings from similar users or items, weighted by their respective similarity measures.
4. **Recommendation Generation**:
   * Based on the predicted ratings, top-N recommendations were generated for users, allowing them to discover items they are likely to enjoy.
5. **Comparison and Evaluation**:
   * The results from both similarity measures were compared in terms of predicted ratings and recommended items, followed by a qualitative evaluation of each approach's strengths and weaknesses.

**2. Tools and Libraries**

The following tools and libraries were utilized in the implementation of the recommendation system:

* **Python**: The primary programming language for the project, selected for its simplicity and extensive libraries for data analysis and manipulation.
* **Pandas**: A powerful library used for data manipulation and analysis. It provided data structures like DataFrames to efficiently handle the dataset and perform operations such as filtering, grouping, and aggregation.
* **NumPy**: This library was used for numerical computations and handling arrays, especially in the calculation of similarity measures and predictions.
* **SciPy**: Utilized for additional statistical functions, including the calculation of Pearson correlation coefficients.
* **Scikit-learn**: Although not explicitly mentioned in the previous steps, this library can be used for additional machine learning techniques, if needed in a more advanced version of the project.
* **Jupyter Notebook**: An interactive development environment that allowed for the incremental development and testing of code. It facilitated the visualization of results through inline graphs and tables.

**Conclusion**

This implementation process, supported by robust tools and libraries, provided a structured approach to building and evaluating a recommendation system. The combination of Python and its ecosystem facilitated efficient data handling and analysis, leading to meaningful insights and recommendations for users.

**Remarks on User-Based vs. Item-Based Collaborative Filtering**

**1. Fundamental Approach**

* **User-Based Collaborative Filtering**:
  + This approach focuses on finding similarities between users based on their rating patterns. It identifies users who have rated items similarly in the past and recommends items that those similar users have enjoyed.
  + The underlying assumption is that if two users agree on the ratings for some items, they are likely to agree on the ratings for other items as well.
* **Item-Based Collaborative Filtering**:
  + In contrast, item-based filtering looks at the relationships between items based on user ratings. It recommends items that are similar to those the user has already liked or rated highly.
  + This method operates on the assumption that if users rate items similarly, those items share characteristics that can be leveraged for recommendations.

**2. Computational Efficiency**

* **User-Based**:
  + User-based filtering can become computationally intensive as the number of users increases. Calculating similarities between every pair of users requires considerable resources, especially in large datasets.
  + It can also suffer from the "cold start" problem, where new users without existing ratings make it difficult to find similar users for recommendations.
* **Item-Based**:
  + Item-based filtering generally performs better with large datasets, as the number of items is often less than the number of users. Once the item-item similarity matrix is computed, generating recommendations can be done quickly and efficiently.
  + Additionally, item-based methods can mitigate the cold start problem better, as new items can still be recommended based on the existing item similarity metrics.

**3. Recommendation Quality and User Experience**

* **User-Based**:
  + Recommendations generated through user-based methods can sometimes lead to a more personalized experience, as they take into account the preferences of similar users.
  + However, the recommendations may be less diverse, as they tend to reinforce existing preferences and might overlook novel items that haven't been rated by similar users.
* **Item-Based**:
  + Item-based recommendations often provide a broader scope, as they can introduce users to a wider array of items based on their past behavior. This can enhance user discovery and satisfaction.
  + The focus on items also allows for more stable recommendations, as item similarities are less likely to change frequently compared to user preferences.

**4. Context and Use Cases**

* **User-Based**:
  + This method can be particularly effective in environments where social influences are strong, such as in social networks or platforms where user interactions significantly impact preferences (e.g., music streaming services).
  + It excels in scenarios where understanding the nuances of user interactions is crucial.
* **Item-Based**:
  + Item-based filtering is often preferred in e-commerce and content platforms (like Netflix or Amazon), where users may benefit from discovering items that share characteristics with those they already enjoy.
  + It is particularly useful for products that have a longer lifespan, as item similarities tend to remain stable over time.

**Conclusion**

Both user-based and item-based collaborative filtering have their unique advantages and disadvantages. The choice between the two should be guided by the specific context of the application, the nature of the data, and the goals of the recommendation system. A hybrid approach, combining elements of both methods, can often yield the best results, balancing personalization with the breadth of recommendations to enhance user experience effectively.