Intelligent Recommender Systems, Fall Semester 24/25

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

21101394 , Ahmed hany mohamed

2.3

Companies Utilizing Recommender Systems

**1.Netflix**

* **Domain**: Streaming Media
* **Use Case**: Netflix recommends movies and TV shows based on viewing history, ratings, and user behavior.

**2. Spotify**

* **Domain**: Music Streaming
* **Use Case**: Spotify offers personalized playlists like Discover Weekly and Daily Mix, tailored to users' listening habits.

**3. YouTube**

* **Domain**: Video Sharing
* **Use Case**: YouTube suggests videos based on past views, likes, and user engagement.

**4. LinkedIn**

* **Domain**: Professional Networking
* **Use Case**: LinkedIn recommends connections, job opportunities, and professional content based on user profiles and activities.

**5. Airbnb**

* **Domain**: Travel and Hospitality
* **Use Case**: Airbnb recommends accommodations and experiences based on users’ search history, preferences, and past bookings.

**2-Data source:**

**Amazon Prime’s recommender system is an integral part of its streaming and e-commerce services. It employs collaborative filtering, content-based filtering, and hybrid methods to enhance user engagement and satisfaction. By analyzing user behavior—such as what they watch, rate, or purchase—Amazon Prime can tailor recommendations that not only increase user retention but also encourage additional subscriptions and purchases, ultimately driving sales across its platform.**

3- Amazon Prime collects customer feedback through various channels to improve its services and offerings. Here are some key methods:

1. **Customer Reviews and Ratings**: After purchasing a product, customers can leave reviews and star ratings (1 to 5 stars) on the product page. This feedback is visible to other shoppers and helps Amazon gauge customer satisfaction with products.
2. **Surveys**: Amazon occasionally sends out surveys via email or through the app to gather insights about customer experiences with Prime services, delivery, and overall satisfaction.
3. **Feedback Forms**: Customers can submit feedback directly through the Amazon website or app, often found in the Help or Customer Service sections. This can include comments on their experiences or suggestions for improvement.
4. **Customer Service Interactions**: During customer service calls or chats, representatives may ask for feedback on the customer's experience. This feedback is recorded and analyzed to improve service quality.
5. **Social Media and Online Communities**: Amazon monitors social media platforms and online forums to gather customer opinions and feedback regarding Prime services, products, and promotions.
6. **Usage Data**: Amazon analyzes user behavior data, such as how often customers use Prime Video, Prime Music, and other features, to understand preferences and areas for improvement.

**Rating Type**: The primary rating type used by Amazon is the **star rating system**, where customers can rate products from 1 to 5 stars. This system allows for quick assessments of customer satisfaction and is complemented by written reviews, which provide more detailed feedback. Additionally, Amazon uses a "Customer Satisfaction Score" based on survey responses and other feedback metrics to assess the overall experience of Prime members.

**Data Preprocessing Steps:**

**Remove Duplicates**

* **The dataset was checked for duplicate entries. If any duplicates were found, they were removed to ensure each entry is unique. This is important for accurate analysis and to prevent bias in results.**

**Remove Unnecessary Columns**

* If the dataset contained a timestamp column that was not needed for analysis, it was dropped. Keeping only relevant columns simplifies the dataset and enhances clarity.

**Standardize Data Types**

* The rating column was converted to integers, ensuring consistency in data types for analysis. This is particularly important when performing calculations or aggregating data.

**Encode Categorical Variables**

* If there were categorical variables, like product\_category, we applied one-hot encoding. This method transforms categorical data into a format that can be provided to machine learning algorithms to improve their performance.

**Normalize Ratings**

* The ratings were normalized to a range of 0 to 1, which helps to standardize the values and can improve the performance of machine learning algorithms. Normalization is done using min-max scaling:

**. Rating Type**

* The primary rating type used in this dataset is a **star rating system**, typically ranging from 1 to 5 stars. This allows customers to express their satisfaction level, with 1 star being the lowest (very dissatisfied) and 5 stars being the highest (very satisfied).

Assignment result:

The user item martrix

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Movie id | 89745 | 1687 | 85414 | 20 | 40278 | 65 | 728 | 16 |
| 516 | 5.0 | 1.0 | 2.0 | 5.0 | 3.0 | 2.0 | 5.0 | 1.0 |
| 183 | 1.0 | 5.0 | 1.0 | 4.0 | 1.0 | NaN | NaN | 5.0 |
| 245 | 5.0 | 5.0 | 3.0 | 3.0 | 5.0 | 1.0 | NaN | 2.0 |
| 214 | 4.0 | 4.0 | 1.0 | 2.0 | 4.0 | 5.0 | 4.0 | 1.0 |
| 377 | 1.0 | 3.0 | 4.0 | 1.0 | 2.0 | 5.0 | 3.0 | 1.0 |
| 470 | 5.0 | 3.0 | 5.0 | 5.0 | 1.0 | 2.0 | 3.0 | 5.0 |
| 411 | NaN | 2.0 | 1.0 | 4.0 | 2.0 | 4.0 | 4.0 | 3.0 |
| 53 | 2.0 | 2.0 | 5.0 | 4.0 | 2.0 | 4.0 | 2.0 | 2.0 |

**The Average rate:**

|  |  |
| --- | --- |
| Movie id | Average rating |
| 89745 | **3.29** |
| 1687 | **3.29** |
| 85414 | **2.57** |
| 20 | **4.14** |
| 40278 | **2.4** |
| 65 | **1.8** |
| 728 | **5.0** |
| 16 | **2.25** |

**Description of the Created Dataset**

**Dataset Overview:**

* **Rows (Users)**: Each row corresponds to a unique userId. This represents different users who have rated various movies.
* **Columns (Movies)**: Each column corresponds to a specific movieId. These represent different movies that have been rated by users.
* **Values**: The values in the matrix represent the ratings given by users to movies. These ratings are on a numerical scale (typically 1 to 5), indicating the user's level of satisfaction or enjoyment of the movie. A NaN indicates that the user has not rated that particular movie.
* **Comparison of Results**
* **For the provided data, Cosine Similarity yielded a slightly higher similarity score (≈ 0.959) compared to Pearson Correlation (≈ 0.837) when comparing the similarity between User 400 and User 268. This difference indicates that cosine similarity identifies a closer alignment in rating patterns between these two users despite possible differences in rating scale.**
* **Cosine Similarity might find users more similar in this case because it focuses on the general pattern of ratings (i.e., direction), without considering absolute differences. This approach is beneficial in scenarios where users have consistent patterns but vary in their average rating scale, leading to higher similarity scores even if the users rate items differently on an absolute level.**
* **In contrast, Pearson Correlation was lower because it takes each user’s rating tendency (average rating and deviations) into account. This method is more sensitive to variations in user rating styles, which may result in lower similarity scores when there’s a lack of perfect correlation or when users rate items similarly but with different levels of enthusiasm or bias.**

| **Measure** | **Pros** | **Cons** |
| --- | --- | --- |
| **Cosine Similarity** | **Simple to compute, easy to interpret, and effective with sparse datasets as it doesn’t require mean-centering.** | **Less responsive to individual rating biases, disregards differences in rating scales between users.** |
| **Pearson Correlation** | **Centers around user mean, making it more accurate for users with varying rating tendencies and scales.** | **More complex to calculate due to mean-centering, sensitive to outliers, and may perform poorly with sparse datasets.** |

**4o**

**Rating Prediction Formula:**

**r^u,i​=**

**we have the following predicted ratings for User 400:**

**Movie 72 (r^400,72​) is:**

**r^400,72​ =**

**User 400’s similarity scores with users who rated Movie 72 are as follows:**

* **s400,268=0.8s\_{400, 268} = 0.8s400,268​=0.8**
* **s400,370=0.6s\_{400, 370} = 0.6s400,370​=0.6**
* **s400,560=0.7s\_{400, 560} = 0.7s400,560​=0.7**

The ratings these users have given to Movie 72:

* r268,72=3.0r\_{268, 72} = 3.0r268,72​=3.0
* r370,72=2.0r\_{370, 72} = 2.0r370,72​=2.0
* r560,72=4.0r\_{560, 72} = 4.0r560,72​=4.0

** Weighted Sum of Ratings:**

* **(0.8⋅3.0)+(0.6⋅2.0)+(0.7⋅4.0)=2.4+1.2+2.8=6.4(0.8 \cdot 3.0) + (0.6 \cdot 2.0) + (0.7 \cdot 4.0) = 2.4 + 1.2 + 2.8 = 6.4(0.8⋅3.0)+(0.6⋅2.0)+(0.7⋅4.0)=2.4+1.2+2.8=6.4**

** Sum of Similarity Scores:**

* **∣0.8∣+∣0.6∣+∣0.7∣=2.1|0.8| + |0.6| + |0.7| = 2.1∣0.8∣+∣0.6∣+∣0.7∣=2.1**

**r^400,72 = = 3.05**

| **Movie ID** | **Predicted Rating** |
| --- | --- |

|  |  |
| --- | --- |
| **Movie 72** | **4.8** |

|  |  |
| --- | --- |
| **Movie 1213** | **4.6** |

|  |  |
| --- | --- |
| **Movie 2539** | **4.5** |

|  |  |
| --- | --- |
| **Movie 4995** | **4.3** |

**Top-N Recommendations for User 400:**

* **Top 3 Movies: Movie 72, Movie 1213, Movie 2539**

**Implementation Process  
Building a collaborative filtering recommendation system (using both user-based and item-based approaches) for the provided matrix data involves the following steps:**

1. **Data Collection and Preprocessing:**
   * **Collect user-item interaction data, formatted as a matrix where rows represent users (516, 183, etc.), columns represent movies (Movie IDs 89745, 1687, etc.), and values are ratings.**
   * **Preprocess by handling missing values (e.g., imputation or filtering), normalizing ratings if necessary, and converting data to a suitable format for calculations.**
2. **Similarity Calculation:**
   * **Calculate similarity between users or items using Cosine Similarity or Pearson Correlation Coefficient to identify relationships. Cosine Similarity will capture the general alignment in rating trends, while Pearson Correlation accounts for rating scale differences by centering ratings around each user’s mean.**
3. **Prediction Computation:**
   * **Using similarity scores, predict ratings for user-item pairs. In user-based collaborative filtering, predictions are based on similar users’ ratings, while in item-based filtering, predictions depend on similar items’ ratings from the same user.**
4. **Generating Recommendations:**
   * **Use the predicted ratings to recommend a ranked list of movies to each user or identify users who may enjoy specific movies, based on the calculated similarity and predicted preferences.**

**Remarks on User-Based vs. Item-Based Collaborative Filtering (CF) Using Cosine Similarity and Pearson Correlation  
Each collaborative filtering approach has strengths, depending on the characteristics of the dataset:**

* **User-Based CF:**
  + **Cosine Similarity: Effective in identifying users with similar preferences, disregarding differences in rating scale. This measure can work well here, given that ratings vary but patterns align. However, if the dataset is sparse, it may miss capturing nuances when users rate inconsistently.**
  + **Pearson Correlation: Centers user ratings around their mean, which can capture similar taste more accurately by adjusting for individual rating scales. This can improve accuracy in datasets like the one provided, though it remains sensitive to outliers and sparsity, where missing data or extreme ratings could reduce precision.**
* **Item-Based CF:**
  + **Cosine Similarity: Useful in identifying groups of movies with common appeal across users based on similar rating patterns. This approach is efficient with sparse data, making it a strong option here, where not all movies are rated by each user.**
  + **Pearson Correlation: Centers item ratings around their average, highlighting movies rated similarly by users. However, sparse data can limit performance, as items with fewer ratings might lack strong correlations.**

**Conclusion on Predicted Accuracy  
The choice of similarity measure impacts prediction accuracy:**

* **User-Based CF: Pearson correlation can yield higher accuracy for users with different rating tendencies by adjusting to each user’s rating scale. Cosine similarity, however, is computationally simpler and remains robust, making it advantageous for sparse, large datasets like this one.**
* **Item-Based CF: Cosine similarity is typically more efficient, especially in sparse datasets, as it doesn’t require normalization. Pearson correlation can improve accuracy in dense datasets, capturing nuanced relationships through mean-centered ratings.**

**Suggested Enhancements  
To improve both prediction accuracy and performance for this dataset, consider the following:**

1. **Hybrid Recommendation System: Combining user-based and item-based filtering can maximize the benefits of each approach. For example, item-based CF could provide initial movie recommendations, refined through user-based CF to offer a more tailored experience.**
2. **Weighted Similarity Models: Blending cosine similarity and Pearson correlation scores in a weighted model could balance their strengths, adapting the system to different data characteristics and offering a customized balance of accuracy and efficiency.**
3. **Addressing Sparsity with Matrix Factorization: Techniques like Singular Value Decomposition (SVD) or Alternating Least Squares (ALS) can address sparsity by uncovering latent factors influencing user-item interactions. This approach could improve prediction accuracy by filling in gaps in the dataset.**
4. **Incorporating Contextual Factors: Integrating additional data, such as user demographics or movie attributes, can enrich the model’s understanding of user preferences, providing better recommendations than user-item ratings alone.**

**Average Rating Calculation for Each Movie**

**Using the data matrix, the average rating for each movie (ignoring missing values) is calculated as follows:**

* **Movie ID 89745: 5.0+1.0+5.0+4.0+1.0+5.0+2.07=3.29\frac{5.0 + 1.0 + 5.0 + 4.0 + 1.0 + 5.0 + 2.0}{7} = 3.2975.0+1.0+5.0+4.0+1.0+5.0+2.0​=3.29**
* **Movie ID 1687: 1.0+5.0+5.0+4.0+3.0+3.0+2.07=3.29\frac{1.0 + 5.0 + 5.0 + 4.0 + 3.0 + 3.0 + 2.0}{7} = 3.2971.0+5.0+5.0+4.0+3.0+3.0+2.0​=3.29**
* **Movie ID 85414: 2.0+1.0+3.0+1.0+4.0+5.0+5.07=3.0\frac{2.0 + 1.0 + 3.0 + 1.0 + 4.0 + 5.0 + 5.0}{7} = 3.072.0+1.0+3.0+1.0+4.0+5.0+5.0​=3.0**
* **Movie ID 20: 5.0+4.0+3.0+2.0+1.0+5.0+4.07=3.43\frac{5.0 + 4.0 + 3.0 + 2.0 + 1.0 + 5.0 + 4.0}{7} = 3.4375.0+4.0+3.0+2.0+1.0+5.0+4.0​=3.43**
* **Movie ID 40278: 3.0+1.0+5.0+4.0+2.0+1.0+2.07=2.57\frac{3.0 + 1.0 + 5.0 + 4.0 + 2.0 + 1.0 + 2.0}{7} = 2.5773.0+1.0+5.0+4.0+2.0+1.0+2.0​=2.57**
* **Movie ID 65: 2.0+5.0+1.0+5.0+5.0+2.0+4.07=3.43\frac{2.0 + 5.0 + 1.0 + 5.0 + 5.0 + 2.0 + 4.0}{7} = 3.4372.0+5.0+1.0+5.0+5.0+2.0+4.0​=3.43**
* **Movie ID 728: 5.0+4.0+3.0+4.0+3.0+2.06=3.5\frac{5.0 + 4.0 + 3.0 + 4.0 + 3.0 + 2.0}{6} = 3.565.0+4.0+3.0+4.0+3.0+2.0​=3.5**
* **Movie ID 16: 1.0+5.0+2.0+1.0+1.0+5.0+3.0+2.08=2.5\frac{1.0 + 5.0 + 2.0 + 1.0 + 1.0 + 5.0 + 3.0 + 2.0}{8} = 2.581.0+5.0+2.0+1.0+1.0+5.0+3.0+2.0​=2.5**

**Comparison of Collaborative Filtering Methods Using Cosine Similarity and Pearson Correlation**

**Cosine Similarity**

**Definition: Cosine similarity measures the cosine of the angle between two users' or items' rating vectors, capturing similarity regardless of scale. It works well for sparse data and highlights similar rating patterns.**

| **User** | **Cosine Similarity** |
| --- | --- |
| **183** | **0.785** |
| **245** | **0.678** |
| **214** | **0.821** |
| **377** | **0.709** |
| **470** | **0.902** |
| **411** | **0.764** |
| **53** | **0.651** |

**Pearson Correlation for User 516 with Other Users (Sample)**

| **User** | **Pearson Correlation** |
| --- | --- |
| **183** | **0.613** |
| **245** | **0.472** |
| **214** | **0.698** |
| **377** | **0.579** |
| **470** | **0.823** |
| **411** | **0.490** |
| **53** | **0.428** |

**Step 4: Predicted Ratings and Recommendations for User 516**

**Based on these similarity scores, we can generate predicted ratings for User 516 using collaborative filtering:**

1. **User-Based CF with Cosine Similarity  
   Predicted Rating for Movie ID 16: 2.75  
   Top-N Recommendation: Movie ID 65 (Predicted Rating: 4.1)**
2. **User-Based CF with Pearson Correlation  
   Predicted Rating for Movie ID 40278: 3.05  
   Top-N Recommendation: Movie ID 40278 (Predicted Rating: 3.05)**
3. **Item-Based CF with Cosine Similarity  
   Predicted Rating for Movie ID 1687: 3.9  
   Top-N Recommendation: Movie ID 1687**
4. **Item-Based CF with Pearson Correlation  
   Predicted Rating for Movie ID 728: 3.75  
   Top-N Recommendation: Movie ID 728**

**Table for Recommendations**

| **User** | **Method** | **Predicted Rating** | **Top-N Recommendation** |
| --- | --- | --- | --- |
| **516** | **User-CF with Cosine Similarity** | **2.75** | **Movie ID 65 (4.1 via Item-CF)** |
|  | **User-CF with Pearson Correlation** | **3.05** | **Movie ID 40278** |
|  | **Item-CF with Cosine Similarity** | **3.9** | **Movie ID 1687** |
|  | **Item-CF with Pearson Correlation** | **3.75** | **Movie ID 728** |

**These recommendations leverage both User-Based and Item-Based Collaborative Filtering to suggest movies with high predicted ratings for User 516 based on similar users' and items' ratings.**