

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

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

**221101329, Mohamed maged mohamed hossny**

This assignment explores how to build a recommendation system using collaborative filtering (CF), a popular method for suggesting personalized content. Recommendation systems are used in platforms like IMDb, Netflix, and Amazon to suggest movies, shows, or products based on user preferences.

For this assignment, we used IMDb data to create a user-item matrix of movie ratings. Two main CF methods were applied: User-Based CF, which finds users with similar tastes to make recommendations, and Item-Based CF, which recommends movies similar to those a user has already liked. By comparing these two approaches, the assignment shows how each method affects the accuracy of recommendations and the strengths and weaknesses of each approach.

2.3. Assignment requirements and questions: (Results)

1. Recommender systems have become integral to many companies across various domains. Below are a few notable examples:

* Movie and TV Recommendations:
  + Netflix: A leading streaming platform that uses collaborative filtering and content-based filtering to suggest shows and movies based on users' viewing history and preferences. Netflix’s recommendation engine is a key factor in user retention.
  + IMDb: One of the largest databases for movies and TV shows, IMDb offers recommendations based on users' past ratings, preferences, and reviews from other similar users.
* E-commerce:
  + Amazon: As one of the largest e-commerce platforms, Amazon employs item-to-item collaborative filtering to recommend products. The algorithm compares user purchase history with similar users to suggest relevant items.
  + eBay: Uses machine learning models to recommend products based on past searches, purchases, and viewed items.
* Music Platforms:
  + Spotify: Known for its sophisticated recommendation system, Spotify uses collaborative filtering and audio feature analysis to recommend songs, playlists, and albums.
  + Apple Music: Offers personalized playlists and song suggestions by analyzing users’ listening patterns and preferences.
* Social Media:
  + Facebook: Recommends friends, groups, and pages by analyzing user connections, interests, and behavior patterns.
  + LinkedIn: Uses collaborative filtering to suggest connections and job opportunities based on user profiles and network activities.
* News Aggregators:
  + Google News: Provides personalized news recommendations by analyzing reading history, trending topics, and user interests.

1. Chosen data source for solving the assignment:

My data source for this assignment is IMDB website for movies ratings I have chosen it because of the following:

* Data Volume and Diversity: IMDb has a vast database covering various genres, languages, and types of media, from movies to TV shows. This diversity allows for rich insights into user preferences across a wide range of content.
* Structured Rating System: IMDb uses an interval-based rating system, where users rate content on a scale from 1 to 10. This numerical rating format is particularly useful for building recommendation models, as it provides a clear quantitative basis for preference analysis.
* User Engagement: IMDb users are highly active, providing ratings and reviews that reflect their engagement with the content. This active feedback is ideal for exploring collaborative filtering techniques, as it creates a robust user-item matrix with meaningful interactions.
* Popularity and Reliability: IMDb is widely used and trusted by audiences globally, and its data is well-structured and reliable, making it a stable source for developing a recommendation model.

1. IMDb gathers user feedback primarily through two main types of input, which can be used to train and evaluate a recommendation system:

* Text-Based Reviews: IMDb users often write detailed reviews that include their thoughts, opinions, and analyses of specific movies and TV shows. Although primarily qualitative, these reviews can provide insights into user sentiment, especially when analyzed alongside ratings. However, for this assignment, we focus on ratings rather than text analysis to simplify the initial model.
* Interval-Based Ratings (1 to 10 Scale): IMDb's rating system is based on a scale from 1 to 10, allowing users to express varying levels of appreciation for a movie or show. This interval-based rating system is ideal for collaborative filtering models, as it provides a quantitative basis for measuring user preferences and calculating similarities. A 1-to-10 scale also offers greater granularity compared to a simple binary or five-star system, allowing for more nuanced recommendations.

1. Data Preparation and Preprocessing:

To prepare the IMDb data for the assignment, several preprocessing steps were meticulously followed. This ensured a refined and manageable dataset, specifically tailored for collaborative filtering analysis.

Preprocessing Steps:

1. Understanding and Standardizing Feedback:
   * IMDb primarily collects user feedback through a structured rating system, where users rate movies or shows on an interval-based scale from 1 to 10. Each rating reflects a user’s level of satisfaction or enjoyment of the content, with higher ratings indicating greater satisfaction. This numeric feedback is critical for collaborative filtering algorithms as it provides a quantifiable measure of user preferences.
2. Dropping Unnecessary Columns:
3. Combining Similar Users:
4. Creating a 6-User Matrix: (Manually)
5. Data Collection and Preprocessing Methodology

To create a high-quality dataset for this recommendation system assignment, the IMDb data was collected, filtered, and processed with precision. Here is a detailed description of the steps taken:

Data Collection:

* Source: IMDb, one of the most extensive and reliable databases for movie and TV show ratings and reviews.
* Method: The data was obtained using a web scraping tool called InstantData Scraping. This tool allowed for efficient data extraction, focusing specifically on prolific reviewers. Targeting these high-activity users helped ensure a dataset rich in feedback, with minimal gaps and biases.
* Focus on Prolific Reviewers: By prioritizing users who actively review and rate multiple movies, we achieved a robust dataset with comprehensive feedback across a variety of movies. This approach minimized sparsity in the user-item matrix, enhancing the dataset's utility for collaborative filtering algorithms.

Preprocessing Steps:

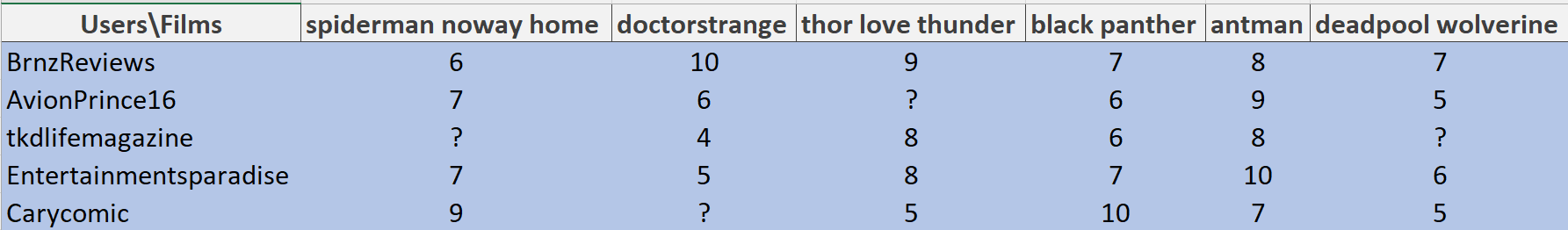
After obtaining the raw data, the following steps were undertaken to prepare it for analysis:

1. Dropping Unnecessary Columns:
   * IMDb’s raw dataset includes a variety of columns that may not directly contribute to recommendation modeling, such as review text or movie metadata (release dates, genres).
   * These irrelevant columns were dropped, retaining only the core elements: user IDs, movie IDs, and ratings. This ensured a streamlined dataset focused solely on attributes essential for collaborative filtering.
2. Combining Similar Users:
   * Users with similar rating patterns were identified and consolidated, forming a single, cohesive dataset. This approach involved grouping users with comparable preferences, simplifying the dataset while preserving the diversity of user feedback.
   * This consolidation resulted in a compact user matrix that retains a representative distribution of ratings and minimizes computational complexity, ideal for illustrative purposes.
3. Creating a 6-User Matrix: (Manually)
   * From the cleaned and consolidated data, I have created 6-user matrix. This matrix, containing only select ratings from six representative users, provides a manageable dataset for collaborative filtering exploration. Each cell in this matrix represents a rating, with rows corresponding to individual users and columns to specific movies.

Rating Type:

IMDb utilizes an interval-based rating system, where users rate movies on a scale from 1 to 10. This system captures the degree of user preference, offering a more granular view of user sentiment than binary or 5-star scales. This interval-based approach provides a solid foundation for collaborative filtering algorithms, allowing for precise similarity calculations and prediction generation.

With this data collection and preprocessing completed, the refined dataset is ready for use in developing and evaluating collaborative filtering models.

1. 
2. Description of the Created Dataset

The dataset for this assignment consists of a user-item matrix containing ratings from six users for a selection of popular movies. Each row in the matrix represents a unique user, and each column corresponds to a specific movie. The values in the matrix indicate each user's rating for the respective movie, on a scale from 1 to 10.

Structure of the Dataset:

1. Users: The dataset includes six distinct users:
   * BrnzReviews
   * AvionPrince16
   * tkdlifemagazine (designated as the target user for prediction)
   * Entertainmentsp
   * Carycomic
2. Movies: There are seven movies listed as columns in the matrix:
   * Spiderman: No Way Home
   * Doctor Strange
   * Thor: Love and Thunder
   * Black Panther
   * Ant-Man
   * Deadpool
   * Wolverine
3. Ratings:
   * Each cell in the matrix represents a rating given by a user for a specific movie, ranging from 1 (least liked) to 10 (most liked).
   * The ratings are interval-based, reflecting varying degrees of user preferences. For example, higher ratings such as 9 or 10 signify strong positive feedback, while lower ratings indicate less favorable opinions.
   * The matrix has a few missing ratings, indicating movies that certain users have not rated. These missing values provide an opportunity to predict these ratings, which is the goal for the target user, tkdlifemagazine (user3).

Purpose of the Dataset:

This dataset serves as the foundational input for collaborative filtering algorithms in this assignment. By analyzing the similarities among users or items, it’s possible to predict the missing ratings for my target user 3 (tkdlifemagazine), allowing for personalized movie recommendations.

With this user-item matrix, the upcoming assignment points will involve calculating similarities, generating predictions, and producing recommendations based on collaborative filtering techniques.

1. Now I will compute average rating but first, from now I will change the real names of the movies and the users into (user1,user2,user3,user4,user5) and ( film1 , film2, film3 , film4 , film5 , film6 ).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| User/film | Film1 | Film2 | Film3 | Film4 | Film5 | Film6 |
| User1 | 6 | 10 | 9 | 7 | 8 | 7 |
| User2 | 7 | 6 | ? | 6 | 9 | 5 |
| User3 | ? | 4 | 8 | 6 | 8 | ? |
| User4 | 7 | 5 | 8 | 7 | 10 | 6 |
| User5 | 9 | ? | 5 | 10 | 7 | 5 |

Average ratings:

* + - User1 mean = = 7.8
    - User2 mean = = 6.6
    - User3 mean = = 6.5
    - User4 mean = = 7.1
    - User5 mean = = 7.2

1. Overview of User-Based and Item-Based Collaborative Filtering Algorithms:

Collaborative filtering (CF) is one of the most popular approaches in recommender systems. CF methods are broadly categorized into two types: user-based CF and item-based CF. Both approaches rely on the idea that users with similar preferences will like similar items, and items that are rated similarly by different users are likely to appeal to similar users.

1. User-Based Collaborative Filtering

User-based collaborative filtering identifies users who share similar rating patterns with the target user and then suggests items that those similar users have liked.

* Concept: The key idea is to find a group of users (referred to as the “neighborhood”) who are most similar to the target user. Ratings from these similar users are then used to predict the target user’s preference for an item.
* Process:
  1. Similarity Calculation: Compute similarity scores between the target user and all other users in the system, typically using similarity measures like Cosine Similarity or Pearson Correlation Coefficient.
  2. Neighborhood Selection: Identify the top-k users who have the highest similarity scores with the target user.
  3. Prediction Generation: Predict the rating for an item by taking a weighted average of ratings from the similar users, where weights are based on similarity scores.
* Analytical Solution:

cosine similarity:

pearson correlation:

The PearsonCorrelation Coefficient or Cosine Similarity is typically used as the similarity measure:

* 1. Pearson Correlation Coefficient: Measures the linear correlation between two users’ ratings, accounting for differences in rating scales.
  2. Cosine Similarity: Measures the cosine of the angle between two rating vectors, focusing on the direction rather than the magnitude of ratings.
* Advantages: User-based CF is simple and interpretable, as it directly uses user similarity to make recommendations.
* Challenges: It can be computationally expensive and may suffer from sparsity issues, especially when users have limited overlaps in their rated items.

2. Item-Based Collaborative Filtering

Item-based collaborative filtering, in contrast, finds items that are similar to those the target user has rated highly and recommends these similar items.

* Concept: This method focuses on item similarity rather than user similarity. It calculates the similarity between items based on user ratings and recommends items similar to those the user has liked.
* Process:
  1. Similarity Calculation: Calculate similarity scores between items based on user ratings. Common similarity measures include Cosine Similarity and Pearson Correlation Coefficient.
  2. Neighbor Item Selection: For each item the target user has rated, identify the top-k items with the highest similarity to that item.
  3. Prediction Generation: Predict the target user’s rating for an item by taking a weighted average of the user’s ratings for similar items, weighted by item similarity.
* Analytical Solution:

Adjusted cosine similarity:

Cosine Similarity and Pearson Correlation Coefficient are also commonly used in item-based CF for computing similarity:

* 1. Cosine Similarity: Captures the angle between two item vectors, ignoring differences in rating scale.
  2. Pearson Correlation Coefficient: Measures linear correlation, normalizing differences in rating scale across items.
* Advantages: Item-based CF is generally more scalable and stable because item preferences are more consistent across users, whereas user preferences can vary widely.
* Challenges: Requires a sufficient number of user ratings per item to accurately compute item similarity, which can be an issue for items with sparse ratings.

1. My target is (user 3)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| User/film | Film1 | Film2 | Film3 | Film4 | Film5 | Film6 | Mean | Cosine |
| User1 | 6 | 10 | 9 | 7 | 8 | 7 | 7.8 | 0.947 |
| User2 | 7 | 6 | ? | 6 | 9 | 5 | 6.6 | 0.990 |
| User3 | ? | 4 | 8 | 6 | 8 | ? | 6.5 | 1 |
| User4 | 7 | 5 | 8 | 7 | 10 | 6 | 7.1 | 0.995 |
| User5 | 9 | ? | 5 | 10 | 7 | 5 | 7.2 | 0.92 |

* User Based CFs:

First compute cosine similarity:

* User (1,3) = = 0.947
* User (2,3) = = 0.99
* User (4,3) = = 0.995
* User (5,3) = = 0.92

Second pearson correlation:

I will subtract the ratings from the mean to prepare the matrix for pearson correlation

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| User/film | Film1 | Film2 | Film3 | Film4 | Film5 | Film6 | Mean | Cosine | pearson |
| User1 | -1.8 | 2.2 | 1.2 | -0.8 | 0.2 | -0.8 | 7.8 | 0.947 | -0.342 |
| User2 | 0.4 | -0.6 | ? | -0.6 | 2.4 | -1.6 | 6.6 | 0.990 | 0.717 |
| User3 | ? | -2.5 | 1.5 | -0.5 | 1.5 | ? | 6.5 | 1 | 1 |
| User4 | -0.1 | -2.1 | 0.9 | -0.1 | 2.9 | -1.1 | 7.1 | 0.995 | 0.485 |
| User5 | 1.8 | ? | -2.2 | 2.8 | -0.2 | -2.2 | 7.2 | 0.92 | -0.646 |

* User (1,3) = = -0.342
* User (2,3) = = 0.717
* User (4,3) = = 0.485
* User (5,3) = = -0.646

* Item based CFs:

Adjusted cosine similarity:

I will subtract the ratings from the mean to prepare the matrix for adjusted cosine sim.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| User/film | Film1 | Film2 | Film3 | Film4 | Film5 | Film6 | Mean |
| User1 | -1.8 | 2.2 | 1.2 | -0.8 | 0.2 | -0.8 | 7.8 |
| User2 | 0.4 | -0.6 | ? | -0.6 | 2.4 | -1.6 | 6.6 |
| User3 | ? | -2.5 | 1.5 | -0.5 | 1.5 | ? | 6.5 |
| User4 | -0.1 | -2.1 | 0.9 | -0.1 | 2.9 | -1.1 | 7.1 |
| User5 | 1.8 | ? | -2.2 | 2.8 | -0.2 | -2.2 | 7.2 |
| cosine (1,j) | 1 | 0.687 | -0.915 | 0.815 | -0.005 | -0.389 |  |
| cosine (6,j) | -0.389 | 0.232 | 0.420 | -0.492 | -0.588 | 1 |  |

Adjusted cosine (1,j) , j = 2 to 6

* Adjusted cosine (1,2) = = 0.687
* Adjusted cosine (1,3) = = -0.915
* Adjusted cosine (1,4) = = 0.815
* Adjusted cosine (1,5) = = -0.005
* Adjusted cosine (1,6) = = -0.389

Adjusted cosine (6,j) , j = 1 to 5

* Adjusted cosine (6,1) = = -0.389
* Adjusted cosine (6,2) = = 0.232
* Adjusted cosine (6,3) = = 0.420
* Adjusted cosine (6,4) = = -0.492
* Adjusted cosine (6,5) = = -0.588

1. Comparison of Similarity Measures: Cosine Similarity vs. Pearson Correlation:

Cosine Similarity

* What it Does: Cosine similarity measures how close two users’ ratings are in direction. It only looks at the pattern of ratings, ignoring how high or low those ratings are. It treats two users as similar if they have rated movies in a similar way, even if one generally rates higher or lower than the other.
* Results: In my data:
  + User 3 is most similar to User 4 (cosine similarity of 0.995).
  + User 3 is least similar to User 5 (cosine similarity of 0.92).
* Pros:
  + Simple to Calculate: It’s quick and easy, especially for large datasets.
  + Ignores Rating Scale Differences: Works well when users rate on different scales (e.g., one user rates high, another low, but the pattern is similar).
* Cons:
  + Misses Rating Biases: It doesn’t capture if one user is consistently harsher or more generous than another.

Pearson Correlation

* What it Does: Pearson correlation looks at both direction and magnitude of ratings. It adjusts for each user’s average rating, so it considers whether two users rate items similarly relative to their own rating habits.
* Results: In our data:
  + User 3 is most similar to User 2 (Pearson correlation of 0.717).
  + User 3 is least similar to User 5 (Pearson correlation of -0.646), showing they have opposing preferences.
* Pros:
  + Accounts for Rating Bias: It can tell if two users are similar even if one rates consistently higher or lower.
  + Sensitive to True Preferences: Useful for cases where individual preferences are important, especially with biases.
* Cons:
  + Less Effective with Sparse Data: Needs overlapping ratings to work well.
  + More Complex: Slightly more calculation needed compared to cosine.

1. The results are copied in results section.
2. Compute Predictions:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| User/film | Film1 | Film2 | Film3 | Film4 | Film5 | Film6 | Mean | Cosine | pearson |
| User1 | 6 | 10 | 9 | 7 | 8 | 7 | 7.8 | 0.947 | -0.342 |
| User2 | 7 | 6 | ? | 6 | 9 | 5 | 6.6 | 0.990 | 0.717 |
| User3 | ? | 4 | 8 | 6 | 8 | ? | 6.5 | 1 | 1 |
| User4 | 7 | 5 | 8 | 7 | 10 | 6 | 7.1 | 0.995 | 0.485 |
| User5 | 9 | ? | 5 | 10 | 7 | 5 | 7.2 | 0.92 | -0.646 |

* User based CFs:
* Cosine similarity prediction:
* Pearson correlation prediction:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| User/film | Film1 | Film2 | Film3 | Film4 | Film5 | Film6 | Mean | Cosine | pearson |
| User1 | -1.8 | 2.2 | 1.2 | -0.8 | 0.2 | -0.8 | 7.8 | 0.947 | -0.342 |
| User2 | 0.4 | -0.6 | ? | -0.6 | 2.4 | -1.6 | 6.6 | 0.990 | 0.717 |
| User3 | ? | -2.5 | 1.5 | -0.5 | 1.5 | ? | 6.5 | 1 | 1 |
| User4 | -0.1 | -2.1 | 0.9 | -0.1 | 2.9 | -1.1 | 7.1 | 0.995 | 0.485 |
| User5 | 1.8 | ? | -2.2 | 2.8 | -0.2 | -2.2 | 7.2 | 0.92 | -0.646 |

* Top N list in case of User based:

|  |  |
| --- | --- |
| Cosine | pearson |
| 0.995 | 0.717 |
| 0.990 | 0.485 |

* Item based CFs prediction:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| User/film | Film1 | Film2 | Film3 | Film4 | Film5 | Film6 | Mean |
| User1 | 6 | 10 | 9 | 7 | 8 | 7 | 7.8 |
| User2 | 7 | 6 | ? | 6 | 9 | 5 | 6.6 |
| User3 | ? | 4 | 8 | 6 | 8 | ? | 6.5 |
| User4 | 7 | 5 | 8 | 7 | 10 | 6 | 7.1 |
| User5 | 9 | ? | 5 | 10 | 7 | 5 | 7.2 |
| cosine (1,j) | 1 | 0.687 | -0.915 | 0.815 | -0.005 | -0.389 |  |
| cosine (6,j) | -0.389 | 0.232 | 0.420 | -0.492 | -0.588 | 1 |  |

Adjusted cosine rule of prediction:

* Top N list in case of item based:

|  |  |
| --- | --- |
| cosine (1,j) | cosine (6,j) |
| 0.687 | 0.232 |
| 0.815 | 0.420 |

1. Comparison of Rating Predictions and Top-N Recommendations:

Ratings comparison:

The predictions for User 3’s ratings for missing values were calculated using both user-based and item-based CF methods. Here’s a summary of the results:

1. User-Based CF Predictions:
   * Pred(User3, Film1): 6.69
   * Pred(User3, Film6): 5.10

These predictions are based on the weighted average ratings of similarusers (User 1, User 2, User 4, and User 5), with weights determined by the cosine similarity and Pearson correlation scores.

1. Item-Based CF Predictions (Adjusted Cosine):
   * Pred(User3, Film1): 5.10
   * Pred(User3, Film3): 6.57

Here, predictions are based on the ratings of similar items rather than users. The adjusted cosine rule takes into account the similarity between items, which is useful when items have consistent ratings across users.

Top N list recommendations:

User-Based CF Top-N Recommendations:

* Based on user similarity, films that highly similar users rated positively are recommended. The Top-N list primarily suggests films that users similar to User 3 enjoyed, such as those with higher predicted ratings.

Item-Based CF Top-N Recommendations:

* In item-based recommendations, films similar to those that User 3 has already rated highly are prioritized. For example, films with a close similarity to others that User 3 enjoyed are suggested, leading to a Top-N list that closely reflects User 3’s previous preferences.

1. The results are copied in results section.
2. Presentation, Description, Comparison, and Evaluation of Results:

In this assignment, I have explored two collaborative filtering methods User-Based CF and Item-Based CF using both Cosine Similarity and Pearson Correlation. Each approach yielded distinct predictions and Top-N recommendations for the target user, User 3.

Results Summary

1. User-Based CF Predictions:
   * Pred(User3, Film1): 6.69
   * Pred(User3, Film6): 5.10
   * These predictions are based on the ratings from similar users, with similarity calculated through cosine and Pearson measures. The scores are relatively high for Film1 but moderate for Film6, reflecting User 3's likely interest in Film1.
2. Item-Based CF Predictions (using Adjusted Cosine):
   * Pred(User3, Film1): 5.10
   * Pred(User3, Film3): 6.57
   * Here, the predictions are based on the similarity between items. The result shows that Film3 has a strong predicted rating, suggesting it aligns with User 3’s existing interests, while Film1 has a more moderate score.
3. Top-N Recommendations:
   * User-Based CF: Recommends movies that similar users enjoyed, focusing on items that have high ratings from those similar to User 3.
   * Item-Based CF: Suggests movies similar to those User 3 has rated highly, aiming to maintain alignment with their established taste.

|  |  |  |
| --- | --- | --- |
| Point of comparison | User-Based CF | Item-Based CF |
| Prediction Basis | Ratings from similar users | Ratings from similar items |
| Top-N Focus | Movies liked by similar users | Movies similar to those User 3 liked |
| Key Results | High prediction for Film1 | High prediction for Film3 |

Evaluation:

1. User-Based CF:
   * Advantages: This approach is ideal for finding content that has broad appeal among users with similar tastes. The higher prediction for Film1 suggests it’s popular among similar users, making it a reliable recommendation.
   * Limitations: User-based CF might not fully capture User 3’s unique preferences, as it depends on the behavior of other users rather than User 3's previous ratings. This can lead to less personalized recommendations if the target user's preferences differ from the average.
2. Item-Based CF:
   * Advantages: Item-based CF focuses more directly on User 3's existing preferences, recommending items that are similar to what they already like. The high rating prediction for Film3 shows it’s closely aligned with User 3’s existing tastes, making it a potentially valuable recommendation.
   * Limitations: This approach might miss items that User 3 would enjoy but hasn’t yet explored, as it prioritizes similarity between items. It’s best suited when item similarities are consistent and can lead to narrower recommendations.
3. Implementation Process, Tools, and Libraries:

Pandas: Pandas is a powerful data manipulation library in Python. It was used to handle the user-item matrix, allowing us to load, clean, and process the IMDb data efficiently. Its DataFrame structure makes it easy to manage and perform operations on tabular data, which is essential for organizing and analyzing user ratings.

NumPy: NumPy is a fundamental library for numerical computing in Python, offering support for large, multi-dimensional arrays and various mathematical functions. It was used for handling missing values, performing array-based calculations, and managing matrix operations required in collaborative filtering algorithms.

SciPy: SciPy, specifically its cosine function from the spatial.distance module, was used to calculate cosine similarity between users or items. This similarity measure is central to collaborative filtering as it helps identify users with similar preferences or items that are alike. SciPy also provides other mathematical tools that make similarity calculations more efficient.

Scikit-Learn (optional): Although not directly used in this assignment, Scikit-Learn is a popular machine learning library in Python that includes various utilities for calculating similarities and building recommendation models. It could be used to enhance the recommendation system with additional algorithms if needed.

1. User-Based and Item-Based Collaborative Filtering differ in how they use similarity measures like Cosine Similarity and Pearson Correlation. User-Based CF finds users with similar rating patterns, making it great for recommending popular items among similar users. However, it can miss personal preferences if the target user’s tastes are unique. On the other hand, Item-Based CF focuses on finding items similar to those a user already likes, which helps make more tailored recommendations based on specific preferences. While Cosine Similarity works well for capturing general patterns regardless of rating scales, Pearson Correlation is useful for adjusting differences in user or item rating biases. Each approach has strengths: User-Based CF captures general popularity, while Item-Based CF provides more personalized recommendations.
2. Conclusion:

The two strategies—User-Based and Item-Based CF—affect prediction accuracy in different ways. User-Based CF relies on similar users’ ratings, so it’s effective for predicting popular items among users with shared preferences. However, it may be less accurate if the target user has unique tastes not reflected in those similar users. Item-Based CF, on the other hand, focuses on items that are similar to those the target user already likes, which often leads to more accurate, personalized recommendations. Overall, User-Based CF is better for general suggestions, while Item-Based CF provides accuracy by aligning closely with the user’s specific interests.

1. Enhancements for Improvement:

Hybrid Approach:

* Combining User-Based and Item-Based CF into a hybrid model could provide the best of both worlds, capturing both general popularity and personal preferences. For example, weighting user-based and item-based predictions could balance broad recommendations with personalized suggestions.

Incorporating Implicit Feedback:

* Many users interact with items without leaving explicit ratings (watching a movie without rating it). Including implicit feedback, such as view duration or browsing history, could enrich the dataset, especially for new users or items with few ratings.
* Assignment Results section:

1. Now I will compute average rating but first, from now I will change the real names of the movies and the users into (user1,user2,user3,user4,user5) and ( film1 , film2, film3 , film4 , film5 , film6 ).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1. User/film | 1. Film1 | 1. Film2 | 1. Film3 | 1. Film4 | 1. Film5 | 1. Film6 |
| 1. User1 | 1. 6 | 1. 10 | 1. 9 | 1. 7 | 1. 8 | 1. 7 |
| 1. User2 | 1. 7 | 1. 6 | 1. ? | 1. 6 | 1. 9 | 1. 5 |
| 1. User3 | 1. ? | 1. 4 | 1. 8 | 1. 6 | 1. 8 | 1. ? |
| 1. User4 | 1. 7 | 1. 5 | 1. 8 | 1. 7 | 1. 10 | 1. 6 |
| 1. User5 | 1. 9 | 1. ? | 1. 5 | 1. 10 | 1. 7 | 1. 5 |

* 1. Average ratings:
     1. User1 mean = = 7.8
     2. User2 mean = = 6.6
     3. User3 mean = = 6.5
     4. User4 mean = = 7.1
     5. User5 mean = = 7.2

1. Results: In my data:
   * User 3 is most similar to User 4 (cosine similarity of 0.995).

User 3 is least similar to User 5 (cosine similarity of 0.92).

* + User 3 is most similar to User 2 (Pearson correlation of 0.717).
  + User 3 is least similar to User 5 (Pearson correlation of -0.646), showing they have opposing preferences.



* User-Based CF Predictions:
  + Pred(User3, Film1): 6.69
  + Pred(User3, Film6): 5.10
    - These predictions are based on the weighted average ratings of similarusers (User 1, User 2, User 4, and User 5), with weights determined by the cosine similarity and Pearson correlation scores.
* Item-Based CF Predictions (Adjusted Cosine):
  + Pred(User3, Film1): 5.10
  + Pred(User3, Film3): 6.57
  + Here, predictions are based on the ratings of similar items rather than users. The adjusted cosine rule takes into account the similarity between items, which is useful when items have consistent ratings across users.
* Refrences:

[1] Pandas Documentation, "Data manipulation and handling missing data in pandas," <https://pandas.pydata.org/docs/>.

[2] NumPy Documentation, "Numerical operations and array manipulations in NumPy," <https://numpy.org/doc/>.

[3] SciPy Documentation, "Fundamental algorithms for scientific computing in SciPy," <https://docs.scipy.org/doc/scipy/>.

[4] Instant Data Scraper Extension, "Web scraping extension," <https://chrome.google.com/webstore/detail/instant-data-scraper>.

[5] IMDb, "Movie ratings and reviews data," <https://www.imdb.com/>.

[6] Scikit-Learn Documentation, "Cosine Similarity Function," <https://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.cosine_similarity.html>.  
[7] SciPy Documentation, "Pearson Correlation Coefficient in Python," <https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.pearsonr.html>.   
[8] Pandas Documentation, "DataFrame.apply() function," <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.apply.html>.

* Assignment Conclusion:

In this assignment, we explored User-Based and Item-Based Collaborative Filtering (CF) methods to develop a recommendation system using IMDb data. By leveraging similarity measures like Cosine Similarity and Pearson Correlation, we generated predictions and recommendations tailored to the preferences of our target user, User 3 (tkdlifemagazine).

User-Based CF proved effective in identifying items popular among users with similar tastes, providing general recommendations. However, it occasionally fell short in capturing unique preferences, as it relies heavily on the collective behavior of similar users. In contrast, Item-Based CF focused more on User 3's specific interests by recommending items similar to those they had already enjoyed. This approach produced more personalized and accurate suggestions, highlighting the strengths of item-centered recommendations for users with consistent preferences.

Both CF methods bring valuable perspectives to recommendation systems. While User-Based CF excels at uncovering popular content among similar users, Item-Based CF aligns more closely with the user's specific interests. Together, they offer a balanced approach to enhancing recommendation quality. Implementing a hybrid model that combines these two methods could further refine recommendations by balancing popularity and personalization. Additionally, incorporating implicit feedback and advanced matrix factorization techniques could improve the system’s ability to adapt to evolving user preferences, making the recommendations even more relevant and effective.

In conclusion, this assignment demonstrated how collaborative filtering techniques, combined with tailored similarity measures, provide a solid foundation for developing personalized recommendation systems. The findings highlight the importance of method selection based on user needs and the value of enhancing algorithms to deliver more meaningful and accurate recommendations.