AIE425 Intelligent Recommender Systems.Full Semester 24/25

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

ID:B20000040 Name :youssef salah ahmed

**Agenda**

**· Introduction**

**· Customer Feedback Collection and Rating System**

**· Data Collection and Structure**

**· Collaborative Filtering Methods (User-Based and Item-Based)**

**· Prediction Results (Using Cosine Similarity and Pearson Correlation)**

**· Comparison of Results**

**· Assignment Results**

**· Implementation Process**

**· Future Recommendations**

**· Conclusion and Opinion**

**· References**

The core idea of the assignment is to explore and compare different collaborative filtering techniques user-based and item-based for making recommendations. Specifically, it involves applying cosine similarity and Pearson correlation to identify similar users or items and predict ratings for unrated items. The goal is to determine which approach and similarity measure yield the most accurate recommendations and to understand the advantages and limitations of each method when applied to a dataset derived from IMDb movie ratings.

Amazon: E-commerce giant leveraging personalized recommendations to suggest products based on user behavior and purchase history.

Netflix: Media streaming platform known for its highly sophisticated algorithm that suggests shows and movies based on viewing patterns.

Spotify: Music streaming service that curates playlists and song recommendations tailored to individual user preferences.

YouTube: Video-sharing platform employing algorithms to recommend videos based on watch history and engagement.

Good reads: Social cataloging website offering book recommendations based on user ratings and preferences.

IMDB: Online database that uses user ratings and feedback to recommend movies and shows.

TripAdvisor: Travel platform suggesting attractions, hotels, and restaurants based on user reviews and ratings.

Pandora: Internet radio service using music preferences to suggest songs.

1. Chosen Company for Data Source: IMDB
2. Customer Feedback Collection and Rating System

IMDB collects customer feedback primarily through user interactions with its platform. Users can rate movies and shows on a scale from 1 to 10, providing a granular view of their preferences. This numeric rating system enables detailed insights into user opinions. Additionally, IMDB allows users to write reviews, adding context to their ratings and enhancing the depth of feedback. By analyzing these ratings and reviews, IMDB can tailor recommendations and offer content that aligns closely with user interests, contributing to a more personalized browsing experience.

1. **Data Loading**: The first step involved importing the collected data into a data structure suitable for analysis, such as a DataFrame. This step is essential for organizing and visualizing the data.

**Data Type Standardization**: Ensuring that all data fields, especially rating columns, were in the correct format of integers was important to maintain consistency throughout the dataset. This helps in preventing errors during analysis.

1. **Data Collection:**

Source: Data was sourced from the IMDB website, leveraging web scraping techniques.

Method: We used requests to retrieve HTML content and BeautifulSoup to parse and extract relevant information such as movie titles and user ratings.

Data Structure: The extracted data was structured in a way that each movie's rating was collected, along with user-generated ratings, creating a dataset that included multiple rows representing user interactions.

Rating Type:

Numeric Scale: The ratings collected from IMDB were on a scale of 1 to 10. these ratings integer form ensured a consistent and simple representation of user feedback, which is essential for model training and analysis.

7.

1. **Source and Purpose:**

Source: The dataset was derived from scraping IMDb's Top 250 movies webpage to extract the titles and ratings of the top movies.

Purpose: The dataset serves as a simulated user-movie ratings matrix that could be used for recommender systems.

2. Structure of the Dataset

Format: CSV

Rows: 101 rows (1 header row + 100 user rows)

Columns: 21 columns (1 user column + 20 movie columns)

3. **Columns Description**

User: Represents the identifier for each simulated user, named sequentially from User\_1 to User\_100.

Movie Columns: Each of the next 20 columns corresponds to one of the top 20 movies extracted from IMDb, with the movie title serving as the column name.

Values: Each cell under the movie columns contains:

A numerical rating between 1 and 10, indicating the rating given by the user.

NaN (or None) indicating a missing value to simulate real-world situations where users may not have rated every movie.

4. **Characteristics of the Ratings**

Range: Ratings range between 1 and 10, consistent with IMDb’s rating system.

Distribution:

Initial Ratings: The original IMDb rating for each movie was extracted as the baseline.

Missing Values: Approximately 10% of the ratings were set to None to represent incomplete data typical in real-world datasets.

8.· The Shawshank Redemption: 8.79

· The Godfather: 8.76

· The Dark Knight: 8.48

· The Godfather Part II: 8.50

· Angry Men: 8.63

· The Lord of the Rings: The Return of the King: 8.61

· Schindler's List: 8.60

· Pulp Fiction: 8.38

· The Lord of the Rings: The Fellowship of the Ring: 8.49

· Il buono, il brutto, il cattivo: 8.29

9.1. **User-Based Collaborative Filtering**

Background and Overview:

User-Based Collaborative Filtering (User-CF) recommends items by identifying users who have shown similar behavior or preferences in the past. If two users have similar tastes or have rated items similarly, they’re likely to have comparable interests in other items as well. This approach leverages user similarity to make recommendations.

Analytical Solution:

The basic steps to implement User-CF are as follows:  
  
1. Calculate User Similarities:  
 - Using a similarity measure (such as cosine similarity, Pearson correlation coefficient, or Jaccard similarity), compute the similarity between pairs of users based on their interactions or ratings on various items.

- For instance, cosine similarity between two users u and v with vectors of ratings r\_u and r\_v can be calculated as:  
  
 sim(u, v) = (r\_u ⋅r\_v) / (||r\_u|| \* ||r\_v||)

Here, represents the dot product, and || ⋅|| denotes the Euclidean norm.  
  
2. Select Nearest Neighbors:  
 - For a target user, identify a set of similar users (neighbors) based on their similarity scores. The most similar users (top-k) are chosen to predict the ratings for the items that the target user hasn’t interacted with.  
  
3. Predict Ratings or Scores:  
 - For a given item i that a user u hasn’t rated, the rating can be predicted by aggregating ratings from similar users who have rated i.  
 - The weighted sum approach is often used, where the predicted rating r̂\_{u,i} is:  
  
 r̂\_{u,i} = r̅\_u + (∑\_(v ∈ N\_u) sim(u, v) ⋅(r\_{v,i} - r̅\_v)) / (∑\_(v ∈ N\_u) |sim(u, v)|)  
  
 where:  
 - N\_u is the set of similar users to u.  
 - r̅\_u and r̅\_v are the average ratings of users u and v, respectively.  
  
4. Generate Recommendations:  
 - Based on predicted ratings, the items with the highest scores are recommended to the target user.

2. **Item-Based Collaborative Filtering**

Background and Overview:

Item-Based Collaborative Filtering (Item-CF) recommends items by analyzing the similarity between items based on user interactions. Unlike User-CF, which relies on user similarities, Item-CF assumes that if two items receive similar ratings from multiple users, they are likely to be similar.

Analytical Solution:

The basic steps to implement Item-CF are as follows:  
  
1. Calculate Item Similarities:  
 - Use a similarity measure to compute the similarity between pairs of items based on user ratings.  
 - Adjusted cosine similarity is often used to account for user rating biases and is calculated as:  
  
 sim(i, j) = (∑\_(u ∈ U\_{i,j}) (r\_{u,i} - r̅\_u)(r\_{u,j} - r̅\_u)) / (√(∑\_(u ∈ U\_{i,j}) (r\_{u,i} - r̅\_u)^2) √(∑\_(u ∈ U\_{i,j}) (r\_{u,j} - r̅\_u)^2))  
  
 where:  
 - U\_{i,j} is the set of users who have rated both items i and j.  
 - r̅\_u is the average rating of user u.  
  
2. Select Nearest Neighbors:  
 - For a given item, find a set of similar items based on the computed similarity scores. These items are used to predict a target user’s rating for an unrated item.  
  
3. Predict Ratings or Scores:  
 - For an item i that a user u hasn’t rated, the predicted rating r̂\_{u,i} is based on the ratings that the user has given to similar items.  
 - A weighted average approach is commonly used:  
  
 r̂\_{u,i} = r̅\_u + (∑\_(j ∈ N\_i) sim(i, j) ⋅(r\_{u,j} - r̅\_u)) / (∑\_(j ∈ N\_i) |sim(i, j)|)  
  
 where N\_i is the set of items similar to i.  
  
4. Generate Recommendations:  
 - Based on predicted ratings, recommend the top-rated items to the user.

10.

User-based Cosine Similarity Calculation  
  
| User | Item\_1 | Item\_2 | Item\_3 | Item\_4 | Item\_5 | Item\_6 | Item\_7 | Item\_8 | Item\_9 | Item\_10 |  
  
| User\_1 | 8 | 9 | 9 | 9 | 9 | 9 | - | 7 | 8 | 8 |  
| User\_2 | 8 | 9 | 8 | 8 | 8 | 9 | - | 7 | - | - |  
| User\_3 | 8 | 9 | 9 | 8 | 9 | 8 | 8 | 7 | 7 | 8 |  
| User\_4 | 8 | 8 | 9 | 8 | 9 | 9 | 8 | 7 | 8 | 7 |  
| User\_5 | 8 | 8 | 8 | 8 | 9 | - | 8 | 7 | 8 | 7 |

Step 2: Calculate Cosine Similarity Between User\_1 and All Users

Similarity Between User\_1 and User\_2

Common Items: Item\_1, Item\_2, Item\_3, Item\_4, Item\_5, Item\_6, and Item\_8.  
- User\_1: [8, 9, 9, 9, 9, 9, 7]  
- User\_2: [8, 9, 8, 8, 8, 9, 7]  
  
Step 3.1: Calculate Dot Product  
(8 × 8) + (9 × 9) + (9 × 8) + (9 × 8) + (9 × 8) + (9 × 9) + (7 × 7) = 491  
  
Step 3.2: Calculate Magnitudes  
||User\_1|| = √(8^2 + 9^2 + 9^2 + 9^2 + 9^2 + 9^2 + 7^2) = √518 ≈ 22.76  
||User\_2|| = √(8^2 + 9^2 + 8^2 + 8^2 + 8^2 + 9^2 + 7^2) = √467 ≈ 21.62  
  
Step 3.3: Calculate Cosine Similarity  
Cosine Similarity = 491 / (22.76 × 21.62) ≈ 0.998

Similarity Between User\_1 and User\_3

Common Items: All items except Item\_7.  
- User\_1: [8, 9, 9, 9, 9, 9, 7, 8, 8]  
- User\_3: [8, 9, 9, 8, 9, 8, 7, 7, 8]

Step 3.1: Calculate Dot Product  
(8 × 8) + (9 × 9) + (9 × 9) + (9 × 8) + (9 × 9) + (9 × 8) + (7 × 7) + (8 × 7) + (8 × 8) = 620  
  
Step 3.2: Calculate Magnitudes  
||User\_1|| = √646 ≈ 25.42  
||User\_3|| = √597 ≈ 24.44  
  
Step 3.3: Calculate Cosine Similarity  
Cosine Similarity = 620 / (25.42 × 24.44) ≈ 0.998

Similarity Between User\_1 and User\_4

Common Items: All items except Item\_7.  
- User\_1: [8, 9, 9, 9, 9, 9, 7, 8, 8]  
- User\_4: [8, 8, 9, 8, 9, 9, 8, 7, 8]  
  
Step 3.1: Calculate Dot Product  
(8 × 8) + (9 × 8) + (9 × 9) + (9 × 8) + (9 × 9) + (9 × 9) + (7 × 8) + (8 × 7) + (8 × 8) = 627  
  
Step 3.2: Calculate Magnitudes  
||User\_1|| = √646 ≈ 25.42  
||User\_4|| = √612 ≈ 24.74  
Step 3.3: Calculate Cosine Similarity

Cosine Similarity = 627 / (25.42 × 24.74) ≈ 0.997

Similarity Between User\_1 and User\_5

Common Items: All items except Item\_6.  
- User\_1: [8, 9, 9, 9, 9, 7, 8, 8]  
- User\_5: [8, 8, 8, 8, 9, 8, 7, 8]  
  
Step 3.1: Calculate Dot Product  
(8 × 8) + (9 × 8) + (9 × 8) + (9 × 8) + (9 × 9) + (7 × 8) + (8 × 7) + (8 × 8) = 537  
  
Step 3.2: Calculate Magnitudes  
||User\_1|| = √565 ≈ 23.77  
||User\_5|| = √514 ≈ 22.68

Step 3.3: Calculate Cosine Similarity

Cosine Similarity = 537 / (23.77 × 22.68) ≈ 0.996

Summary of Cosine Similarities:  
- Similarity between User\_1 and User\_2: 0.998  
- Similarity between User\_1 and User\_3: 0.998  
- Similarity between User\_1 and User\_4: 0.997  
- Similarity between User\_1 and User\_5: 0.996

**User-based Pearson Correlation Calculation:**

| User | Item\_1 | Item\_2 | Item\_3 | Item\_4 | Item\_5 | Item\_6 | Item\_7 | Item\_8 | Item\_9 | Item\_10 |  
  
| User\_1 | 8 | 9 | 9 | 9 | 9 | 9 | - | 7 | 8 | 8 |  
| User\_2 | 8 | 9 | 8 | 8 | 8 | 9 | - | 7 | - | - |  
| User\_3 | 8 | 9 | 9 | 8 | 9 | 8 | 8 | 7 | 7 | 8 |  
| User\_4 | 8 | 8 | 9 | 8 | 9 | 9 | 8 | 7 | 8 | 7 |  
| User\_5 | 8 | 8 | 8 | 8 | 9 | - | 8 | 7 | 8 | 7 |

Step 2: Calculate Pearson Correlation Between User\_1 and All Users

Pearson Correlation Between User\_1 and User\_2

Common Items: Item\_1, Item\_2, Item\_3, Item\_4, Item\_5, Item\_6, and Item\_8.  
- User\_1: [8, 9, 9, 9, 9, 9, 7]  
- User\_2: [8, 9, 8, 8, 8, 9, 7]  
  
Step 3.1: Calculate Means  
Mean(User\_1) = (8 + 9 + 9 + 9 + 9 + 9 + 7) / 7 = 8.57  
Mean(User\_2) = (8 + 9 + 8 + 8 + 8 + 9 + 7) / 7 = 8.14  
  
Step 3.2: Calculate Numerator  
Numerator = Σ((User\_1[i] - Mean(User\_1)) \* (User\_2[i] - Mean(User\_2)))  
= (8 - 8.57) \* (8 - 8.14) + (9 - 8.57) \* (9 - 8.14) + ... + (7 - 8.57) \* (7 - 8.14)  
= -0.57 \* -0.14 + 0.43 \* 0.86 + 0.43 \* -0.14 + 0.43 \* -0.14 + 0.43 \* -0.14 + 0.43 \* 0.86 + -1.57 \* -1.14  
= 0.0798 + 0.3698 - 0.0602 - 0.0602 - 0.0602 + 0.3698 + 1.7898 = 2.4286  
  
Step 3.3: Calculate Denominator  
Denominator = sqrt(Σ((User\_1[i] - Mean(User\_1))^2) \* Σ((User\_2[i] - Mean(User\_2))^2))  
= sqrt(((-0.57)^2 + (0.43)^2 + ... + (-1.57)^2) \* ((-0.14)^2 + (0.86)^2 + ... + (-1.14)^2))  
= sqrt(3.99 \* 3.78) = sqrt(15.0822) ≈ 3.88  
  
Step 3.4: Calculate Pearson Correlation  
Pearson Correlation = Numerator / Denominator ≈ 2.4286 / 3.88 ≈ 0.626

Pearson Correlation Between User\_1 and User\_3

Common Items: All items except Item\_7.  
- User\_1: [8, 9, 9, 9, 9, 9, 7, 8, 8]  
- User\_3: [8, 9, 9, 8, 9, 8, 7, 7, 8]  
  
Step 3.1: Calculate Means  
Mean(User\_1) = 8.22  
Mean(User\_3) = 8.11  
  
Step 3.2: Calculate Numerator and Denominator  
Follow similar steps as above.  
  
Result: Pearson Correlation ≈ 0.705

Pearson Correlation Between User\_1 and User\_4

Common Items: All items except Item\_7.  
- User\_1: [8, 9, 9, 9, 9, 9, 7, 8, 8]  
- User\_4: [8, 8, 9, 8, 9, 9, 8, 7, 8]  
  
Step 3.1: Calculate Means  
Mean(User\_1) = 8.22  
Mean(User\_4) = 8.00  
  
Step 3.2: Calculate Numerator and Denominator  
Follow similar steps as above.  
  
Result: Pearson Correlation ≈ 0.688

Pearson Correlation Between User\_1 and User\_5

Common Items: All items except Item\_6.  
- User\_1: [8, 9, 9, 9, 9, 7, 8, 8]  
- User\_5: [8, 8, 8, 8, 9, 8, 7, 8]  
  
Step 3.1: Calculate Means  
Mean(User\_1) = 8.38  
Mean(User\_5) = 8.13  
  
Step 3.2: Calculate Numerator and Denominator  
Follow similar steps as above.  
  
Result: Pearson Correlation ≈ 0.543  
Summary of Pearson Correlations:

- Pearson Correlation between User\_1 and User\_2: 0.626  
- Pearson Correlation between User\_1 and User\_3: 0.705  
- Pearson Correlation between User\_1 and User\_4: 0.688  
- Pearson Correlation between User\_1 and User\_5: 0.543

**Item-based Cosine Similarity Calculation:**  
  
| User | Item\_1 | Item\_2 | Item\_3 | Item\_4 | Item\_5 | Item\_6 | Item\_7 | Item\_8 | Item\_9 | Item\_10 |  
  
| User\_1 | 8 | 9 | 9 | 9 | 9 | 9 | - | 7 | 8 | 8 |  
| User\_2 | 8 | 9 | 8 | 8 | 8 | 9 | - | 7 | - | - |  
| User\_3 | 8 | 9 | 9 | 8 | 9 | 8 | 8 | 7 | 7 | 8 |  
| User\_4 | 8 | 8 | 9 | 8 | 9 | 9 | 8 | 7 | 8 | 7 |  
| User\_5 | 8 | 8 | 8 | 8 | 9 | - | 8 | 7 | 8 | 7 |

Cosine Similarity Between Item\_6 and Item\_1

Step 3.1: Calculate Dot Product

Dot Product = Σ(Item\_6[i] \* Item\_1[i]) = 280

Step 3.2: Calculate Magnitudes

||Item\_6|| = 17.52

||Item\_1|| = 16.00

Step 3.3: Calculate Cosine Similarity

Cosine Similarity = Dot Product / (||Item\_6|| \* ||Item\_1||) = 0.999

Cosine Similarity Between Item\_6 and Item\_2

Step 3.1: Calculate Dot Product

Dot Product = Σ(Item\_6[i] \* Item\_2[i]) = 298

Step 3.2: Calculate Magnitudes

||Item\_6|| = 17.52

||Item\_2|| = 17.03

Step 3.3: Calculate Cosine Similarity

Cosine Similarity = Dot Product / (||Item\_6|| \* ||Item\_2||) = 0.999

Cosine Similarity Between Item\_6 and Item\_3

Step 3.1: Calculate Dot Product

Dot Product = Σ(Item\_6[i] \* Item\_3[i]) = 306

Step 3.2: Calculate Magnitudes

||Item\_6|| = 17.52

||Item\_3|| = 17.52

Step 3.3: Calculate Cosine Similarity

Cosine Similarity = Dot Product / (||Item\_6|| \* ||Item\_3||) = 0.997

Cosine Similarity Between Item\_6 and Item\_4

Step 3.1: Calculate Dot Product

Dot Product = Σ(Item\_6[i] \* Item\_4[i]) = 289

Step 3.2: Calculate Magnitudes

||Item\_6|| = 17.52

||Item\_4|| = 16.52

Step 3.3: Calculate Cosine Similarity

Cosine Similarity = Dot Product / (||Item\_6|| \* ||Item\_4||) = 0.998

Cosine Similarity Between Item\_6 and Item\_5

Step 3.1: Calculate Dot Product

Dot Product = Σ(Item\_6[i] \* Item\_5[i]) = 306

Step 3.2: Calculate Magnitudes

||Item\_6|| = 17.52

||Item\_5|| = 17.52

Step 3.3: Calculate Cosine Similarity

Cosine Similarity = Dot Product / (||Item\_6|| \* ||Item\_5||) = 0.997

Cosine Similarity Between Item\_6 and Item\_7

Step 3.1: Calculate Dot Product

Dot Product = Σ(Item\_6[i] \* Item\_7[i]) = 136

Step 3.2: Calculate Magnitudes

||Item\_6|| = 12.04

||Item\_7|| = 11.31

Step 3.3: Calculate Cosine Similarity

Cosine Similarity = Dot Product / (||Item\_6|| \* ||Item\_7||) = 0.998

Cosine Similarity Between Item\_6 and Item\_8

Step 3.1: Calculate Dot Product

Dot Product = Σ(Item\_6[i] \* Item\_8[i]) = 245

Step 3.2: Calculate Magnitudes

||Item\_6|| = 17.52

||Item\_8|| = 14.00

Step 3.3: Calculate Cosine Similarity

Cosine Similarity = Dot Product / (||Item\_6|| \* ||Item\_8||) = 0.999

Cosine Similarity Between Item\_6 and Item\_9

Step 3.1: Calculate Dot Product

Dot Product = Σ(Item\_6[i] \* Item\_9[i]) = 200

Step 3.2: Calculate Magnitudes

||Item\_6|| = 15.03

||Item\_9|| = 13.30

Step 3.3: Calculate Cosine Similarity

Cosine Similarity = Dot Product / (||Item\_6|| \* ||Item\_9||) = 1.000

Cosine Similarity Between Item\_6 and Item\_10

Step 3.1: Calculate Dot Product

Dot Product = Σ(Item\_6[i] \* Item\_10[i]) = 199

Step 3.2: Calculate Magnitudes

||Item\_6|| = 15.03

||Item\_10|| = 13.30

Step 3.3: Calculate Cosine Similarity

Cosine Similarity = Dot Product / (||Item\_6|| \* ||Item\_10||) = 0.995

**Item-based Pearson Correlation Calculation :**

| User | Item\_1 | Item\_2 | Item\_3 | Item\_4 | Item\_5 | Item\_6 | Item\_7 | Item\_8 | Item\_9 | Item\_10 |  
  
| User\_1 | 8 | 9 | 9 | 9 | 9 | 9 | - | 7 | 8 | 8 |  
| User\_2 | 8 | 9 | 8 | 8 | 8 | 9 | - | 7 | - | - |  
| User\_3 | 8 | 9 | 9 | 8 | 9 | 8 | 8 | 7 | 7 | 8 |  
| User\_4 | 8 | 8 | 9 | 8 | 9 | 9 | 8 | 7 | 8 | 7 |  
| User\_5 | 8 | 8 | 8 | 8 | 9 | - | 8 | 7 | 8 | 7 |

Pearson Correlation Between Item\_6 and Item\_1

Step 3.1: Calculate Means

Mean(Item\_6) = 8.75

Mean(Item\_1) = 8.00

Step 3.2: Calculate Numerator

Numerator = Σ((Item\_6[i] - Mean(Item\_6)) \* (Item\_1[i] - Mean(Item\_1))) = 0.00

Step 3.3: Calculate Denominator

Denominator = sqrt(Σ((Item\_6[i] - Mean(Item\_6))^2) \* Σ((Item\_1[i] - Mean(Item\_1))^2)) = 0.00

Step 3.4: Calculate Pearson Correlation

Pearson Correlation = 0.000

Pearson Correlation Between Item\_6 and Item\_2

Step 3.1: Calculate Means

Mean(Item\_6) = 8.75

Mean(Item\_2) = 8.75

Step 3.2: Calculate Numerator

Numerator = Σ((Item\_6[i] - Mean(Item\_6)) \* (Item\_2[i] - Mean(Item\_2))) = -0.25

Step 3.3: Calculate Denominator

Denominator = sqrt(Σ((Item\_6[i] - Mean(Item\_6))^2) \* Σ((Item\_2[i] - Mean(Item\_2))^2)) = 0.75

Step 3.4: Calculate Pearson Correlation

Pearson Correlation = -0.333

Pearson Correlation Between Item\_6 and Item\_3

Step 3.1: Calculate Means

Mean(Item\_6) = 8.75

Mean(Item\_3) = 8.75

Step 3.2: Calculate Numerator

Numerator = Σ((Item\_6[i] - Mean(Item\_6)) \* (Item\_3[i] - Mean(Item\_3))) = -0.25

Step 3.3: Calculate Denominator

Denominator = sqrt(Σ((Item\_6[i] - Mean(Item\_6))^2) \* Σ((Item\_3[i] - Mean(Item\_3))^2)) = 0.75

Step 3.4: Calculate Pearson Correlation

Pearson Correlation = -0.333

Pearson Correlation Between Item\_6 and Item\_4

Step 3.1: Calculate Means

Mean(Item\_6) = 8.75

Mean(Item\_4) = 8.25

Step 3.2: Calculate Numerator

Numerator = Σ((Item\_6[i] - Mean(Item\_6)) \* (Item\_4[i] - Mean(Item\_4))) = 0.25

Step 3.3: Calculate Denominator

Denominator = sqrt(Σ((Item\_6[i] - Mean(Item\_6))^2) \* Σ((Item\_4[i] - Mean(Item\_4))^2)) = 0.75

Step 3.4: Calculate Pearson Correlation

Pearson Correlation = 0.333

Pearson Correlation Between Item\_6 and Item\_5

Step 3.1: Calculate Means

Mean(Item\_6) = 8.75

Mean(Item\_5) = 8.75

Step 3.2: Calculate Numerator

Numerator = Σ((Item\_6[i] - Mean(Item\_6)) \* (Item\_5[i] - Mean(Item\_5))) = -0.25

Step 3.3: Calculate Denominator

Denominator = sqrt(Σ((Item\_6[i] - Mean(Item\_6))^2) \* Σ((Item\_5[i] - Mean(Item\_5))^2)) = 0.75

Step 3.4: Calculate Pearson Correlation

Pearson Correlation = -0.333

Pearson Correlation Between Item\_6 and Item\_7

Step 3.1: Calculate Means

Mean(Item\_6) = 8.50

Mean(Item\_7) = 8.00

Step 3.2: Calculate Numerator

Numerator = Σ((Item\_6[i] - Mean(Item\_6)) \* (Item\_7[i] - Mean(Item\_7))) = 0.00

Step 3.3: Calculate Denominator

Denominator = sqrt(Σ((Item\_6[i] - Mean(Item\_6))^2) \* Σ((Item\_7[i] - Mean(Item\_7))^2)) = 0.00

Step 3.4: Calculate Pearson Correlation

Pearson Correlation = 0.000

Pearson Correlation Between Item\_6 and Item\_8

Step 3.1: Calculate Means

Mean(Item\_6) = 8.75

Mean(Item\_8) = 7.00

Step 3.2: Calculate Numerator

Numerator = Σ((Item\_6[i] - Mean(Item\_6)) \* (Item\_8[i] - Mean(Item\_8))) = 0.00

Step 3.3: Calculate Denominator

Denominator = sqrt(Σ((Item\_6[i] - Mean(Item\_6))^2) \* Σ((Item\_8[i] - Mean(Item\_8))^2)) = 0.00

Step 3.4: Calculate Pearson Correlation

Pearson Correlation = 0.000

Pearson Correlation Between Item\_6 and Item\_9

Step 3.1: Calculate Means

Mean(Item\_6) = 8.67

Mean(Item\_9) = 7.67

Step 3.2: Calculate Numerator

Numerator = Σ((Item\_6[i] - Mean(Item\_6)) \* (Item\_9[i] - Mean(Item\_9))) = 0.67

Step 3.3: Calculate Denominator

Denominator = sqrt(Σ((Item\_6[i] - Mean(Item\_6))^2) \* Σ((Item\_9[i] - Mean(Item\_9))^2)) = 0.67

Step 3.4: Calculate Pearson Correlation

Pearson Correlation = 1.000

Pearson Correlation Between Item\_6 and Item\_10

Step 3.1: Calculate Means

Mean(Item\_6) = 8.67

Mean(Item\_10) = 7.67

Step 3.2: Calculate Numerator

Numerator = Σ((Item\_6[i] - Mean(Item\_6)) \* (Item\_10[i] - Mean(Item\_10))) = -0.33

Step 3.3: Calculate Denominator

Denominator = sqrt(Σ((Item\_6[i] - Mean(Item\_6))^2) \* Σ((Item\_10[i] - Mean(Item\_10))^2)) = 0.67

Step 3.4: Calculate Pearson Correlation

Pearson Correlation = -0.500

**11.Comparison of Results**

**Pearson Correlation Coefficient Results:**

Item-based Pearson Correlation (between Item\_6 and other items) yielded both positive, negative, and zero correlations.

For example, the Pearson Correlation between Item\_6 and Item\_9 was 1.0, indicating a perfect positive relationship, while Item\_6 and Item\_10 had -0.5, indicating a moderate negative correlation.

In user-based calculations, Pearson correlation between User\_1 and other users resulted in correlations ranging from 0.543 to 0.705, indicating moderate positive correlations between users.

**Cosine Similarity Results:**

Item-based Cosine Similarity (between Item\_6 and other items) generally yielded high values ranging between 0.995 and 1.000, indicating that the items are very similar in terms of angle between vectors rather than their magnitudes.

In user-based calculations, Cosine Similarity between User\_1 and other users produced consistently high similarities, mostly around 0.996 to 0.998, indicating strong relationships.

**Pros and Cons of Each Technique :Pearson Correlation Coefficient**

Pros:

Captures Linear Relationships: Pearson correlation measures the strength and direction of a linear relationship between two variables. It helps identify positive, negative, or no correlation.

Centered Comparison: It involves centering the ratings by subtracting the mean, which is useful when accounting for differences in scale or biases among users or items.

Cons:

Negative Values: Pearson correlation can yield negative values, which may make interpretation harder in similarity contexts. For instance, negative values indicate inverse relationships, which might not be meaningful when calculating item or user similarity.

Affected by Magnitude Differences: If items or users have low variability in their ratings, Pearson correlation can yield zero or near-zero values, implying no relationship. This was observed with several items, such as Item\_6 and Item\_1, where the correlation was zero.

**Cosine Similarity**

Pros:Magnitude Independence: Cosine similarity considers only the angle between vectors, which means it is not affected by the magnitudes of the ratings. This makes it effective in cases where ratings are scaled differently.

Always Positive: Cosine similarity ranges from 0 to 1, avoiding negative values, making it easier to interpret in similarity contexts. In the given results, cosine similarities were consistently high, suggesting strong similarity even when Pearson correlation had negative or zero values.

Cons:

Lack of Centering: Unlike Pearson correlation, Cosine similarity does not center the ratings by subtracting the mean. This means it does not account for user or item biases

Cannot Capture Inverse Relationships: Since Cosine similarity is always positive, it cannot capture negative correlations. This might be a limitation when inverse relationships are meaningful in the dataset.

**12.Assignment Results:**

1. Pearson Correlation Coefficient:  
 - Item-based Pearson Correlation: The correlations between Item\_6 and other items ranged from -0.5 to 1.0, with some items showing positive relationships and others negative.

- User-based Pearson Correlation: The correlations between User\_1 and other users were moderate, ranging from 0.543 to 0.705.  
  
2. Cosine Similarity:  
 - Item-based Cosine Similarity: The similarity values between Item\_6 and other items were consistently high, ranging from 0.995 to 1.000, indicating strong similarity between items.

- User-based Cosine Similarity: The similarities between User\_1 and other users were consistently strong, mostly around 0.996 to 0.998.

**13.Prediction of user based cosine similarity :**

Similar Users and Their Ratings for Item\_7:

User\_3: Rating = 8

User\_4: Rating = 8

User\_5: Rating = 8

Cosine Similarities with User\_1:

User\_3: 0.998

User\_4: 0.997

User\_5: 0.996

Predicted Rating Calculation

Predicted Rating (User\_1, Item\_7) = (0.998 \* 8 + 0.997 \* 8 + 0.996 \* 8) / (0.998 + 0.997 + 0.996)

Step 4.2.1: Calculate the Numerator

(0.998 \* 8) + (0.997 \* 8) + (0.996 \* 8) = 7.984 + 7.976 + 7.968 = 23.928

Step 4.2.2: Calculate the Denominator

0.998 + 0.997 + 0.996 = 2.991

Step 4.2.3: Calculate Predicted Rating

Predicted Rating (User\_1, Item\_7) = 23.928 / 2.991 ≈ 8.00

**Prediction of user based Pearson correlation coefficient:**

Similar Users and Their Ratings for Item\_7:

User\_3: Rating = 8

User\_4: Rating = 8

User\_5: Rating = 8

Pearson Correlations with User\_1:

User\_3: 0.705

User\_4: 0.688

User\_5: 0.543

Predicted Rating Calculation

Predicted Rating (User\_1, Item\_7) = (0.705 \* 8 + 0.688 \* 8 + 0.543 \* 8) / (0.705 + 0.688 + 0.543)

Step 4.2.1: Calculate the Numerator

(0.705 \* 8) + (0.688 \* 8) + (0.543 \* 8) = 5.64 + 5.504 + 4.344 = 15.488

Step 4.2.2: Calculate the Denominator

0.705 + 0.688 + 0.543 = 1.936

Step 4.2.3: Calculate Predicted Rating

Predicted Rating (User\_1, Item\_7) = 15.488 / 1.936 ≈ 8.00

**Prediction of item based cosine similarity :**

The ratings by User\_1 for the items that are similar to Item\_6 are:

Item\_1: Rated by User\_1 = 8

Item\_2: Rated by User\_1 = 9

Item\_3: Rated by User\_1 = 9

Item\_4: Rated by User\_1 = 9

Item\_5: Rated by User\_1 = 9

Item\_8: Rated by User\_1 = 7

Item\_9: Rated by User\_1 = 8

Item\_10: Rated by User\_1 = 8

Step 2: Cosine Similarities with Item\_6

Cosine Similarity between Item\_6 and Item\_1: 0.999

Cosine Similarity between Item\_6 and Item\_2: 0.999

Cosine Similarity between Item\_6 and Item\_3: 0.997

Cosine Similarity between Item\_6 and Item\_4: 0.998

Cosine Similarity between Item\_6 and Item\_5: 0.997

Cosine Similarity between Item\_6 and Item\_8: 0.999

Cosine Similarity between Item\_6 and Item\_9: 1.000

Cosine Similarity between Item\_6 and Item\_10: 0.995

Step 3: Predicted Rating Calculation

Step 4.2.1: Calculate the Numerator

(0.999 \* 8) = 7.992  
(0.999 \* 9) = 8.991  
(0.997 \* 9) = 8.973  
(0.998 \* 9) = 8.982  
(0.997 \* 9) = 8.973  
(0.999 \* 7) = 6.993  
(1.000 \* 8) = 8.000  
(0.995 \* 8) = 7.960

Sum of these products: 7.992 + 8.991 + 8.973 + 8.982 + 8.973 + 6.993 + 8.000 + 7.960 = 66.864

Calculate the Denominator

Sum of similarity values: 0.999 + 0.999 + 0.997 + 0.998 + 0.997 + 0.999 + 1.000 + 0.995 = 7.984

Calculate Predicted Rating

Predicted Rating (User\_1, Item\_6) = 66.864 / 7.984 ≈ 8.38

**Prediction of item based Pearson correlation coefficient:**

The ratings for User\_5 are:  
- Item\_1: 8  
- Item\_2: 8  
- Item\_3: 8  
- Item\_4: 8  
- Item\_5: 9  
- Item\_7: 8  
- Item\_8: 7  
- Item\_9: 8  
- Item\_10: 7

The Pearson correlations for Item\_6 with other items are:  
- Item\_1: 0.000  
- Item\_2: -0.333  
- Item\_3: -0.333  
- Item\_4: 0.333  
- Item\_5: -0.333  
- Item\_7: 0.000  
- Item\_8: 0.000  
- Item\_9: 1.000  
- Item\_10: -0.500

Calculate the Numerator

(-0.333 \* 8) = -2.664

(-0.333 \* 8) = -2.664

(0.333 \* 8) = 2.664

(-0.333 \* 9) = -2.997

(1.000 \* 8) = 8.000

(-0.500 \* 7) = -3.500

Sum of these products: -2.664 + (-2.664) + 2.664 + (-2.997) + 8.000 + (-3.500) = 1.161

Calculate the Denominator

Sum of similarity values: |-0.333| + |-0.333| + |0.333| + |-0.333| + |1.000| + |-0.500| = 2.832

Step 3: Calculate the Predicted Rating

Predicted Rating (User\_5, Item\_6) = 1.161 / 2.832 ≈ 0.41

14.

User-Based Cosine Similarity vs Pearson Correlation for Item\_7

Cosine Similarity Prediction:

Predicted Rating for User\_1 on Item\_7 is 8.00.

Similar users (User\_3, User\_4, User\_5) have very high cosine similarities (0.998, 0.997, 0.996), meaning they have similar preferences.

Pearson Correlation Prediction:

Predicted Rating for User\_1 on Item\_7 is also 8.00.

Similar users (User\_3, User\_4, User\_5) have lower correlations (0.705, 0.688, 0.543). This means the correlation captures a more nuanced relationship compared to cosine similarity.

Implications:

Both similarity measures predict the same rating, but cosine similarity gives much higher similarity values between users compared to Pearson correlation.

For the top-N recommendation list, cosine similarity would prioritize User\_3, User\_4, and User\_5 as highly similar users more confidently compared to Pearson correlation, which suggests a less strong similarity.

Item-Based Cosine Similarity vs Pearson Correlation for Item\_6

Cosine Similarity Prediction:

Predicted Rating for User\_1 on Item\_6 is 8.38.

The similarities with other items are quite high, with values ranging from 0.995 to 1.000, indicating that Item\_6 is very similar to the other items rated by User\_1.

Pearson Correlation Prediction:

Predicted Rating for User\_5 on Item\_6 is 0.41.

The correlations vary significantly, with negative and low values for most items, except Item\_9 which has a perfect correlation (1.000). This means that the relationship between Item\_6 and other items is not strong or consistent.

Implications:

Cosine Similarity results in a much higher predicted rating (8.38), while Pearson Correlation predicts a very low rating (0.41). This indicates that, based on cosine similarity, User\_1 would be interested in Item\_6, whereas Pearson correlation suggests that User\_5 may not like Item\_6.

For the top-N recommendation list, cosine similarity would rank Item\_6 highly for User\_1, while Pearson correlation would not recommend Item\_6 to User\_5 due to the low predicted rating.

**15.Assignment Results:**

User-Based Cosine Similarity: Predicted Rating for User\_1 on Item\_7 is 8.00.

User-Based Pearson Correlation: Predicted Rating for User\_1 on Item\_7 is 8.00.

Item-Based Cosine Similarity: Predicted Rating for User\_1 on Item\_6 is 8.38.

Item-Based Pearson Correlation: Predicted Rating for User\_5 on Item\_6 is 0.41.

**16.1. User-based CF using Cosine Similarity**

In the user-based approach with cosine similarity, the predicted ratings matrix contains values indicating the expected ratings for items a user hasn't yet rated based on other users with similar rating behaviors. The cosine similarity for users ranged between 0.75 to 1, indicating that most users have high similarity, which likely results in more accurate predictions for those users with numerous rating overlaps. However, the predictions in this approach were sometimes zero, implying the lack of sufficient similar users for certain items, leading to uninformative recommendations.

**2. Item-based CF using Cosine Similarity**

The item-based CF using cosine similarity evaluates the similarity between items instead of users. This approach predicted ratings by examining other items that the user has rated similarly. The results showed more consistent ratings across different users, suggesting that items are often rated in similar ways by various users. The cosine similarity values were generally high, indicating strong item associations. However, similar to user-based CF, there were instances where zero ratings were predicted, particularly for items with fewer overlapping ratings.

**3. User-based CF using Pearson Correlation**

In the user-based CF method using Pearson correlation, the similarity between users was evaluated by assessing their rating correlations. This approach allowed us to focus on the strength of the linear relationships between ratings rather than merely angle-based similarity. The predicted ratings using Pearson correlation exhibited slightly more variability compared to cosine similarity. There were also some values in the prediction matrix, indicating scenarios where the linear correlation couldn't be properly calculated, usually due to lack of overlapping ratings.

**4. Item-based CF using Pearson Correlation**

The item-based CF with Pearson correlation calculates the linear correlation between items based on users' ratings. Similar to the user-based method, this approach exhibited some variability, but it also resulted in higher predictive quality for certain items. Since items with strong correlations often share similar rating patterns, this method is more suited for users who prefer specific genres or themes. However, as observed in the predictions, Pearson correlation sometimes leads to null values, which could make this approach less reliable for sparsely rated datasets.

**Comparison:**

Cosine Similarity vs. Pearson Correlation: Cosine similarity ensures that users/items are compared irrespective of their average rating values, focusing purely on directionality. In contrast, Pearson correlation also takes into account the users’ or items' rating biases. This makes Pearson correlation effective in distinguishing users with similar rating behaviors beyond directional similarity. However, it tends to yield NaN values when there are insufficient overlapping ratings, which was evident in both user and item-based methods.

User-based vs. Item-based CF: User-based CF tends to perform better in environments with active user involvement and abundant ratings, as it relies on identifying peer users. However, item-based CF is more stable when users exhibit clear preferences for particular items. In this analysis, item-based CF was observed to provide more stable predictions, with fewer instances of zero or NaN values, especially with cosine similarity, suggesting more comprehensive item relationships.

**Evaluation:**

Accuracy: Item-based CF using cosine similarity appears to provide more consistent predictions compared to user-based approaches, indicating higher prediction accuracy for items that are generally similarly rated by most users.

Sparsity Handling: Both cosine and Pearson methods faced issues dealing with sparse data, as indicated by the presence of zero or NaN predictions. Item-based CF handles sparsity slightly better, as items tend to have more consistent rating histories across a larger user base.

**17.Implementation Process:**

The implementation involves collecting movie ratings, computing similarity matrices, predicting ratings, and generating recommendations. Specifically:

Data Collection: We fetched movie data from IMDB’s Top 250 list using a web scraping approach with requests and BeautifulSoup. We extracted titles, ratings, and simulated user ratings, storing this data in a CSV file.

Data Preparation: We converted the collected ratings into a matrix suitable for analysis and replaced missing values with zeros for compatibility with similarity calculations.

Similarity Computation: We computed user and item similarities using two methods—cosine similarity and Pearson correlation.

Rating Prediction: Ratings were predicted using collaborative filtering based on similarity measures, employing both user-based and item-based approaches.

Generating Recommendations: Finally, top-N recommendations were generated for a specific user using each collaborative filtering technique.

**Tools and Libraries:**

Python: The primary programming language for the entire implementation.

Requests: Used for web scraping to fetch IMDB data.

BeautifulSoup: For parsing HTML content and extracting the necessary information from the IMDB page.

CSV: To handle the storage and organization of user and movie ratings data.

NumPy: Utilized for numerical operations, including matrix manipulations and computing similarities.

Pandas: For data handling and converting CSV files into structured data.

Scikit-learn (cosine\_similarity): Used to compute cosine similarity between users/items.

SciPy (pearsonr): For calculating Pearson correlation between users/items to evaluate linear relationships.

**18.**

User-based CF looks for people who have similar tastes and recommends movies they liked to others with similar preferences. It’s great for personalization, but it struggles when there isn't enough data like when a new movie comes out and hasn't been rated much yet. On the other hand, item-based CF works by finding similarities between movies. It predicts ratings based on how a user rated similar movies, and it tends to be more stable since relationships between items are more consistent over time.

As for the similarity measures, cosine similarity and Pearson correlation have their strengths and weaknesses. Cosine similarity is simple and effective, focusing purely on the direction of ratings rather than the scale. It worked well for user-based CF, giving consistent results. Pearson correlation, on the other hand, tries to normalize ratings to account for individual preferences, but it didn’t perform as well when data was sparse it ended up with gaps in the predictions.

1. **conclusion:**

each collaborative filtering (CF) strategy, whether user-based or item-based, produced varying levels of accuracy in predicting ratings. The user-based CF with cosine similarity showed a strong ability to group users with similar rating patterns, while Pearson correlation tended to struggle with missing values and led to some inconsistencies in the predictions. On the other hand, item-based CF with cosine similarity was consistent but generally yielded less variability in predictions compared to user-based approaches. Overall, cosine similarity was found to be more reliable than Pearson correlation, especially in handling the sparse rating matrix and maintaining stable prediction values.

​

**20.**

Incorporate Hybrid Filtering: Combining user-based and item-based collaborative filtering techniques could provide more robust recommendations. By leveraging the strengths of both approaches, we can address weaknesses like data sparsity and better accommodate diverse user preferences.

Add More Data Sources: Including additional contextual data such as user demographics, movie genres, or viewing history would help refine the similarity measures. This could enhance personalization and create better user-specific recommendations.

Use Matrix Factorization: Implementing advanced techniques like matrix factorization, for example, Singular Value Decomposition (SVD), could help reduce the dimensionality of the data and better address the sparsity issue, resulting in more accurate rating predictions.

Apply Deep Learning Techniques: Neural collaborative filtering could be another potential enhancement. Deep learning models are known for their capability to capture complex, non-linear relationships, and applying them to user-item interactions might yield improved prediction accuracy.

Optimize Similarity Measures: Instead of using cosine similarity or Pearson correlation alone, exploring other similarity measures such as adjusted cosine similarity or Jaccard similarity might better capture relationships and improve predictions.

**Conclusion**

this assignment helped us explore different collaborative filtering techniques—both user-based and item-based—and how they can be used to make recommendations. Each method has its strengths and weaknesses. User-based filtering is great for personalization, as it identifies users with similar tastes. However, it can face challenges with sparse data, like when there are new movies or users who haven’t rated much. On the other hand, item-based filtering is more stable, especially when items have consistent rating patterns, making it reliable in many scenarios.

From what I’ve seen, cosine similarity works well for identifying similarities without worrying too much about rating scales. It’s consistent when we have a lot of data. Pearson correlation, on the other hand, is better at adjusting for different user preferences by normalizing ratings, but it can struggle when data is sparse, leading to some unpredictable results.

**My opinion**

I think the best approach is a hybrid one—combining both user-based and item-based methods. By using cosine similarity to capture overall patterns and Pearson correlation to adjust for user preferences when possible, we can create a more balanced and effective recommendation system. This way, we take advantage of each method’s strengths while minimizing their individual shortcomings, offering better and more reliable recommendations for users.

References

1. <https://ieeexplore.ieee.org/document/5166005>
2. <https://ieeexplore.ieee.org/document/963770>
3. <https://ieeexplore.ieee.org/document/8187128>