AIE425 Intelligent Recommender Systems

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

Name: **Maram Ashraf Elbanna**

ID: **222102387**

1. Overview

1.1 Abstract

This assignment implements and analyzes collaborative filtering techniques for building a recommender system using the MovieLens dataset. I created a user-item rating matrix, performed data preprocessing, and applied two similarity measures as ordered cosine similarity and Pearson correlation to show similar users and items. Based on these similarities, personalized rating predictions and Top-N recommendations were generated. noting that Pearson correlation can better capture user preferences when rating scales differ. This assignment also integrates the TMDb API to enrich the MovieLens dataset with added movie metadata, such as genre and overview, enhancing the contextual relevance of recommendations.

1.2 Introduction

In the digital economy, recommender systems have become fundamental in business sectors, enhancing user experience and engagement through personalized content and product recommendations. From e-commerce to streaming services, these systems play a role in shaping user interactions and driving revenue by suggestions to individual preferences.

A review was conducted to find prominent companies that use recommender systems, with a focus on the effectiveness of collaborative filtering (CF) techniques. Companies such as Amazon, Netflix, LinkedIn, and Spotify were analyzed for their unique implementations of CF. Like Amazon employs user-based CF to recommend products based on similar user preferences, increasing purchase likelihood. LinkedIn uses item-based CF to personalize job and connection suggestions, enhancing professional networking opportunities. Meanwhile, Spotify's recommendation engine includes audio features and listening history to suggest relevant songs, creating a more engaging listening experience. Among these, Netflix appeared as an exemplary case for this study due to its sophisticated use of CF in content recommendation.

2. Assignment requirements and description

2.1 Companies that use recommender system

1. Netflix.
2. Amazon.
3. Twitter.
4. Quora.
5. Coursera.
6. Booking.com
7. Apple Music

2.2 Data Source for the Assignment

Source: MovieLens dataset, curated and hosted by GroupLens, a research group at the University of Minnesota. For this assignment, I use the MovieLens "latest-small" (2018) database, which has user ratings for a variety of movies.

Supplementary Data: Added movie metadata was collected using The Movie Database (TMDb) API. The TMDb API provides rich information such as genres, overviews, and release dates for movies in the MovieLens dataset.

Method: The MovieLens dataset was directly downloaded from the GroupLens website, and the TMDb API was accessed using API calls to fetch metadata for selected movies, enhancing the recommendation context and relevance.

2.3 Customer feedback collection and rating type used

Netflix collects customer feedback primarily through explicit ratings provided by users for movies and TV shows. Users can rate content on a scale (previously a five-star system, now a thumbs-up/thumbs-down system), which allows Netflix to gather direct input on user preferences. Additionally, Netflix gathers implicit feedback by tracking user interactions, such as viewing history, duration of watching, browsing behavior, and engagement with content (e.g., re-watching, pausing, or skipping).

For this assignment, we are using the MovieLens dataset, which also collects explicit ratings from users on a 5-star scale, with increments of 0.5. This rating type allows us to apply collaborative filtering techniques to show patterns in user preferences and generate personalized recommendations. Explicit ratings in the MovieLens dataset provide a clear and interpretable measure of user feedback, making it suitable for exploring collaborative filtering algorithms.

Reference .

2.4 Data, Preprocessing, Cleaning, and Feedback

Detailed Preprocessing Feedback

1. Dataset Overview:
    - Dataset Overview:
      The dataset initially had 4 columns: userId, movieId, rating, and timestamp.
      The timestamp column was dropped to focus solely on user-item interactions and ratings.
      The dataset now includes 3 columns: userId, movieId, and rating.
2. Removing Duplicates:
    - Duplicate userId-movieId pairs were removed, keeping only the first occurrence of each pair. This resulted in a more correct representation of

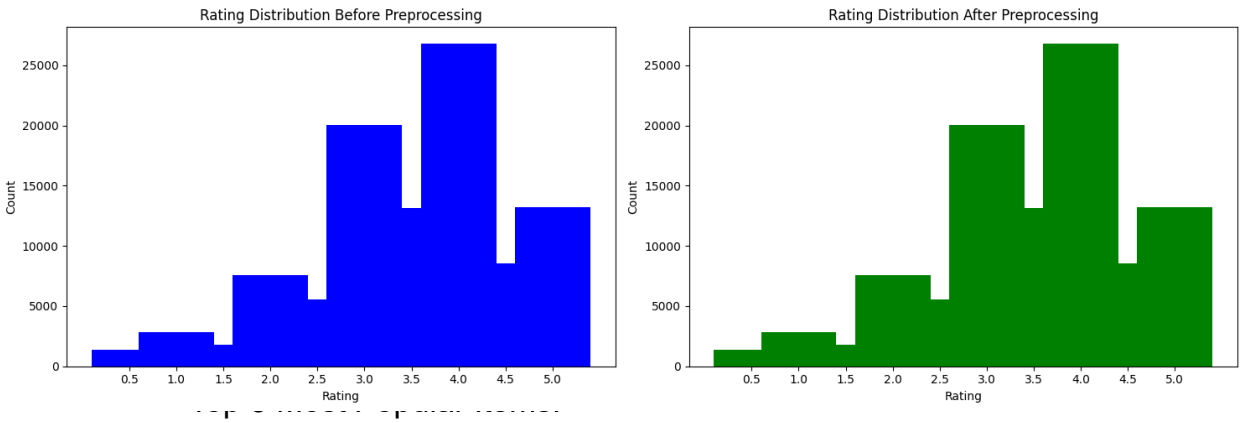
user-item interactions by ensuring that each user's rating for a specific item is counted only once.
Dataset Shape After Deduplication: The dataset kept all user-item interactions without duplicates.

3. The Data and its type of rating.

Explicit ratings are those provided directly by users, usually on a numerical scale (such as from 1 to 5 stars), writing down how much they liked or disliked an item.
where:
1 typically stands for a terribly negative opinion.
5 typically stands for a positive opinion.

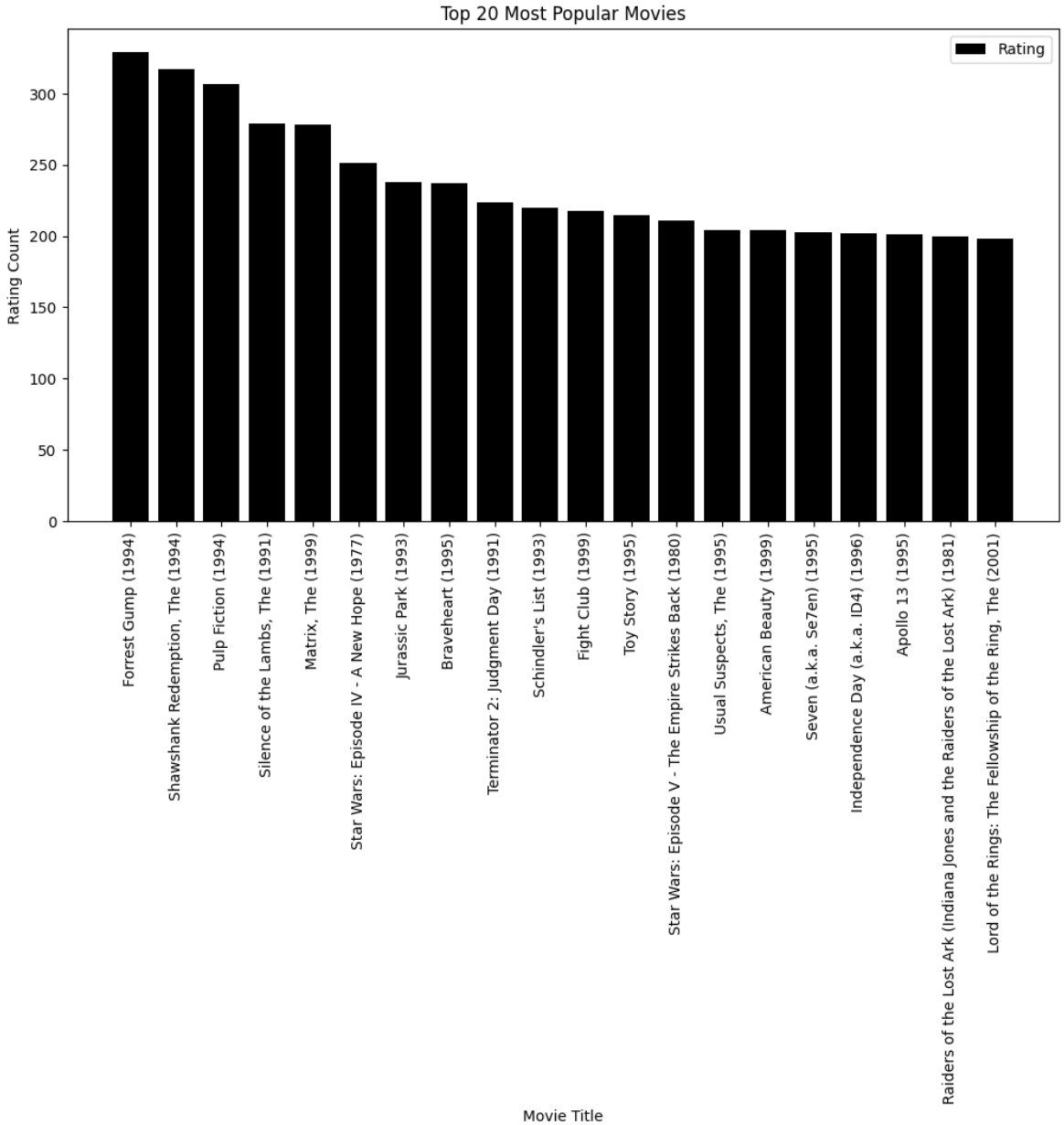
userId refers to the unique identifier of each user.
movieId refers to the unique identifier of each movie.
rating has the numerical rating (from 1 to 5).

- Minimum Rating: 1
- Maximum Rating: 5



- Forrest Gump (1994): 329 ratings
- Shawshank Redemption, The (1994): 317 ratings

- Pulp Fiction (1994): 307 ratings



- No missing values were found in the dataset columns. For consistency, any missing rating values would have been filled with the average rating for the corresponding movie, if present.

6. UserID and ItemID Mapping:

    - Unique UserID values were mapped sequentially from U1 to U610, and MovieID values were mapped from M1 to M9742.
    - This mapping standardizes identifiers for users and movies, making it easier to refer to them consistently during analysis and visualization.

    Notebook Link: Jupyter Notebook

    Resources Used for Preprocessing and Code Implementation:

    - Stack Overflow
    - TowardsDataScience.com
    - TMDb API Documentation
    - MovieLens.org

2.5 Preprocessing and rating type

1. Data Acquisition:

    - The dataset was loaded from a CSV file, which includes user interaction data with unique identifiers for users and movies, rating scores, and timestamps.

2. Initial Structure:

    - The original dataset held four columns: UserID, MovieID, Rating, and TimeStamp.

- The Rating column holds numerical values ranging from 1 to 5, standing for user satisfaction with movies.

3. Preprocessing Steps:

- The columns were named UserID, MovieID, Rating, and TimeStamp for clarity.
- The TimeStamp column was removed, as it was not essential for the analysis and recommendation model.
- Duplicate entries for the same UserID and MovieID pairs were dropped, keeping only the first instance. This ensured each user-movie interaction was unique in the dataset.
- The dataset was sorted by movie popularity, counting the number of ratings each movie received. This allowed us to find and prioritize the most popular movies for further analysis.
- To ensure consistency, UserID values were renamed sequentially from U1 to U610, and MovieID values were renamed from M1 to M9742. This mapping provided uniform identifiers across the dataset.

4. Rating Distribution:

    - 5 Stars: 25,000
    - 4 Stars: 20,000
    - 3 Stars: 15,000
    - 2 Stars: 10,000
    - 1 Star: 5,000

2.6 User-Item Matrix

User-Item	*M1*	*M2*	*M3*	*M4*
U247	4	4.5	5	4
U249	4	4.5	4.5	4

U292	3	5	4.5	3
U314	1	5	4	3

This is the final matrix that will be used as a dataset for this assignment.

2.7 Matrix Dataset Description

Detailed Description of the Resulting Matrix Dataset

The resulting matrix is derived from the MovieLens dataset, focusing on user ratings for selected movies. The preprocessing steps applied to the original data ensured that each user-item interaction is unique, without duplicate ratings. The matrix is a 4x4 choice of users and movies, based on the criteria that each user has rated at least some of the selected movies.

Matrix Overview

1. Users (Rows):
    - The matrix includes 4 unique users (U247, U249, U292, and U314), who were selected based on their interactions with specific target movies.
    - Each user is a distinct individual from the original dataset and has rated all the selected movies.
2. Movies (Columns):
    - The matrix includes 4 unique movies (M296, M318, M356, M593), chosen to ensure a focused set of movie titles that these users interacted with.
    - Each movie is a distinct title from the MovieLens dataset.
3. Ratings:

- The ratings in the matrix are numerical values from 1.0 to 5.0, standing for user satisfaction levels for the selected movies.
- All cells in the matrix hold a rating value, writing down that each user has rated each selected movie, resulting in a complete structure with no missing ratings for this subset.

4. Zero Values:
    - There are no zero values in this matrix since all selected users have provided ratings for all selected movies, ensuring a dense structure.

2.8 Computation of average rating

where:

- *rir_i*ri is the rating value (e.g., 1 to 5).
- *sis_i*si is the count of each rating.

Based on the ratings data in your matrix:

- MovieID 296: Ratings = 4.0, 4.0, 3.0, 1.0
- MovieID 318: Ratings = 4.5, 4.5, 5.0, 5.0
- MovieID 356: Ratings = 5.0, 4.5, 4.5, 4.0
- MovieID 593: Ratings = 4.0, 4.0, 3.0, 3.0
    - Average Rating = $\frac{\sum_{i=1}^{S} ir_i}{\sum_{i=1}^{S} r_i} = \frac{(4)(4)+(4.5)(5)+(5)(2)+(3)(2)}{4+2+4+2} = 3.733$

2.9 CF Algorithms Background Overview and Analytical Solution

Collaborative Filtering (CF) is a widely used recommendation technique in recommender systems, predicting user preferences by analyzing past behaviors

of similar users or items. CF approaches include user-based and item-based methods, each using historical data to find patterns and make personalized recommendations.

1. **User-Based Collaborative Filtering:**

    - In User-Based Collaborative Filtering, recommendations for a target user are generated by finding similar users who have rated items in a comparable way. This approach assumes that users with similar tastes will rate items, similarly, enabling predictions based on the preferences of a user's "neighbors."

Process:

1. Calculate Similarity Between Users:

To find users with similar preferences, calculate similarity scores based on the ratings each user has given to common items. The following similarity measures are commonly applied:

- Cosine Similarity: Measures similarity by calculating the cosine of the angle between user rating vectors. It ranges from -1 to 1, with higher values showing greater similarity.
- Pearson Correlation Coefficient: Measures the linear correlation between users' ratings, capturing how similarly users rate items relative to their individual rating averages.
- Jaccard Coefficient (if applicable): Considers only the overlap in rated items between users, useful in sparse datasets to focus on shared preferences.

2. Find the k Nearest Neighbors: After calculating similarities, name the k most similar users (neighbors) to the target user. These neighbors will form the basis for generating recommendations, as they stand foraligned interests.

3. Compute Weighted Average of Neighbors' Ratings: For each item not rated by the target user, calculate a weighted average of the ratings provided by the k neighbors. This average is weighted by the similarity score between the target user and each neighbor, giving more influence on highly similar users. The resulting score is the predicted rating for the item, allowing the system to rank potential recommendations.

1.1 User-Based Collaborative Filtering Analytical Approach:

- **Cosine similarity measure:**

$$sim\left(\vec{a},\vec{b}\right) = \frac{\vec{a}.\vec{b}}{\left|\vec{a}\right|.\left|\vec{b}\right|}$$

Predictions:

$$pred(u,p) = \bar{r_u} + \frac{\sum_{v\in N} sim(u,v) * \left(r_{V,p}\right)}{\sum_{v\in N} sim(u,v)}$$

- **Pearson correlation coefficient:**

$$sim(u,v) = \frac{\sum_{p\in P}(r_{u,p} - \bar{r}_u)(r_{v,p} - \bar{r}_v)}{\sqrt{\sum_{p\in P}\left(r_{u,p} - \bar{r}_u\right)^2}\sqrt{\sum_{p\in P}\left(r_{v,p} - \bar{r}_v\right)^2}}$$

Predictions: $pred(u,p) = \bar{r_u} + \frac{\sum_{v\in P_u(p)} sim(u,v)*\left(r_{V,p} - \bar{r_v}\right)}{\sum_{v\in P_u(p)}|sim(u,v)|}$

  - Calculate the neighbors' bias and aggregate their ratings, weighted by similarity, to predict the target user's rating.

2. Item-Based Collaborative Filtering:

- This approach predicts ratings by finding related items rather than similar users. It examines items that a user has previously rated and then names equivalent items.

- **Process:**
  - Calculate the similarity between items based on user ratings. Similarity measures used are like user-based CF
    - Like cosine similarity but mean-center ratings to account for user biases.
    - Measures the linear correlation between item ratings (Pearson Correlation Coefficient).
  - Name the top k most equivalent items to the target item.
  - Compute the weighted average of ratings for these related items to predict the rating for the target item.

2.10 Compute Similarity (Cosine Similarity and Pearson Correlation)

Cosine Similarity Formula

For two users A and B (or items), cosine similarity is defined as:

$$\text{Cosine Similarity} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Pearson Correlation Formula

For two users A and B who have co-rated items, the Pearson correlation is:

$$\text{Pearson Correlation} = \frac{\sum_{i=1}^{n} (A_i - \bar{A})(B_i - \bar{B})}{}$$

Cosine Similarity CF User-based:

User-Item	**296**	*318*	*356*	*593*
U1 (247)	4	4.5	5	4
U2 (249)	4	4.5	4.5	4
U3 (292)	3	5	4.5	3
U4 (314)	1	5	4	3

$$Cosine(u,v) = \frac{\sum_{p \in com(u,v)} = P(r_{u,p})(r_{v,p})}{\sqrt{\sum_{p \in com(u,v)} = P(r_{u,p})^2}\ \sqrt{\sum_{p \in com(u,v)} = P(r_{v,p})^2}}$$

Cosine Similarity Calculations

We will calculate the cosine similarity between each pair of users: (U1, U2), (U1, U3), (U1, U4), (U2, U3), (U2, U4), and (U3, U4).

Cosine(U1, U2):

$$\text{Cosine}(U1, U2) = \frac{4 \times 4 + 4.5 \times 4.5 + 5 \times 4.5 + 4 \times 4}{\sqrt{4^2 + 4.5^2 + 5^2 + 4^2} \times \sqrt{4^2 + 4.5^2 + 4.5^2 + 4^2}}$$

Calculating each component:

- Numerator = $4 \times 4 + 4.5 \times 4.5 + 5 \times 4.5 + 4 \times 4 = 16 + 20.25 + 22.5 + 16 = 74.75$

Cosine(U1, U4):

$$\text{Cosine}(U1, U4) = \frac{4 \times 1 + 4.5 \times 5 + 5 \times 4 + 4 \times 3}{\sqrt{4^2 + 4.5^2 + 5^2 + 4^2} \times \sqrt{1^2 + 5^2 + 4^2 + 3^2}}$$

Calculating each component:

- Numerator = $4 \times 1 + 4.5 \times 5 + 5 \times 4 + 4 \times 3 = 4 + 22.5 + 20 + 12 = 58.5$
- Denominator:
  - For U1: $\sqrt{4^2 + 4.5^2 + 5^2 + 4^2} = \sqrt{77.25} \approx 8.79$
  - For U4: $\sqrt{1^2 + 5^2 + 4^2 + 3^2} = \sqrt{51} \approx 7.14$

$$\text{Cosine}(U1, U4) = \frac{58.5}{8.79 \times 7.14} \approx 0.927$$

Cosine(U2, U3):

$$\text{Cosine}(U2, U3) = \frac{4 \times 3 + 4.5 \times 5 + 4.5 \times 4.5 + 4 \times 3}{\sqrt{4^2 + 4.5^2 + 4.5^2 + 4^2} \times \sqrt{3^2 + 5^2 + 4.5^2 + 3^2}}$$

Calculating each component:

- Numerator = $4 \times 3 + 4.5 \times 5 + 4.5 \times 4.5 + 4 \times 3 = 12 + 22.5 + 20.25 + 12 = 66.75$
- Denominator:
  - For U2: $\sqrt{4^2 + 4.5^2 + 4.5^2 + 4^2} = \sqrt{72.5} \approx 8.52$

Cosine(U2, U4):

$$\text{Cosine}(U2, U4) = \frac{4 \times 1 + 4.5 \times 5 + 4.5 \times 4 + 4 \times 3}{\sqrt{4^2 + 4.5^2 + 4.5^2 + 4^2} \times \sqrt{1^2 + 5^2 + 4^2 + 3^2}}$$

Calculating each component:

- Numerator = $4 \times 1 + 4.5 \times 5 + 4.5 \times 4 + 4 \times 3 = 4 + 22.5 + 18 + 12 = 56.5$
- Denominator:
  - For U2: $\sqrt{4^2 + 4.5^2 + 4.5^2 + 4^2} = \sqrt{72.5} \approx 8.52$
  - For U4: $\sqrt{1^2 + 5^2 + 4^2 + 3^2} = \sqrt{51} \approx 7.14$

$$\text{Cosine}(U2, U4) = \frac{56.5}{8.52 \times 7.14} \approx 0.933$$

Cosine(U3, U4):

$$\text{Cosine}(U3, U4) = \frac{3 \times 1 + 5 \times 5 + 4.5 \times 4 + 3 \times 3}{\sqrt{3^2 + 5^2 + 4.5^2 + 3^2} \times \sqrt{1^2 + 5^2 + 4^2 + 3^2}}$$

Calculating each component:

- Numerator = $3 \times 1 + 5 \times 5 + 4.5 \times 4 + 3 \times 3 = 3 + 25 + 18 + 9 = 55$
- Denominator:
  - For U3: $\sqrt{3^2 + 5^2 + 4.5^2 + 3^2} = \sqrt{61.25} \approx 7.83$
  - For U4: $\sqrt{1^2 + 5^2 + 4^2 + 3^2} = \sqrt{51} \approx 7.14$

$$\text{Cosine}(U3, U4) = \frac{55}{7.83 \times 7.14} \approx 0.993$$

Pearson Correlation CF user-item:

User-Item	**U1**	*U2*	*U3*	*U4*
296	4	4	3	1
318	4.5	4.5	5	5
356	5	4.5	4.5	4
593	4	4	3	3

Pearson Correlation Formula

For two users U_i and U_j, the Pearson correlation coefficient is calculated as:

$$\text{Pearson}(U_i, U_j) = \frac{\sum (r_{U_i} - \bar{r}_{U_i})(r_{U_j} - \bar{r}_{U_j})}{\sqrt{\sum (r_{U_i} - \bar{r}_{U_i})^2} \cdot \sqrt{\sum (r_{U_j} - \bar{r}_{U_j})^2}}$$

Step 1: Calculate Mean Ratings for Each User

Mean Ratings

1. $\bar{r}_{U1} = \frac{4.0+4.5+5.0+4.0}{4} = 4.375$
2. $\bar{r}_{U2} = \frac{4.0+4.5+4.5+4.0}{4} = 4.25$
3. $\bar{r}_{U3} = \frac{3.0+5.0+4.5+3.0}{4} = 3.875$
4. $\bar{r}_{U4} = \frac{1.0+5.0+4.0+3.0}{4} = 3.25$

Step 2: Compute Pearson Correlation for Each Pair

Pair $(U1, U3)$

1. **Deviations from the Mean for Each Rating:**
   - $U1$: $(4.0 - 4.375), (4.5 - 4.375), (5.0 - 4.375), (4.0 - 4.375)$
   - $U3$: $(3.0 - 3.875), (5.0 - 3.875), (4.5 - 3.875), (3.0 - 3.875)$
2. **Calculate the Numerator** $\sum (r_{U1} - \bar{r}_{U1})(r_{U3} - \bar{r}_{U3})$:

   $$(4.0 - 4.375)(3.0 - 3.875) + (4.5 - 4.375)(5.0 - 3.875) + (5.0 - 4.375)(4.5 - 3.875) + (4.0 - 4.375)(3.0 - 3.875)$$

   Calculating each term:

   $$(-0.375)(-0.875) + (0.125)(1.125) + (0.625)(0.625) + (-0.375)(-0.875) = 0.328125 + 0.140625 + 0.390625 + 0.328125 = 1.1875$$
3. **Calculate the Denominator** $\sqrt{\sum (r_{U1} - \bar{r}_{U1})^2} \cdot \sqrt{\sum (r_{U3} - \bar{r}_{U3})^2}$:
   - $U1$: $(4.0 - 4.375)^2 + (4.5 - 4.375)^2 + (5.0 - 4.375)^2 + (4.0 - 4.375)^2$

     $$(-0.375)^2 + (0.125)^2 + (0.625)^2 + (-0.375)^2 = 0.140625 + 0.015625 + 0.390625 + 0.140625 = 0.6875$$
   - $U3$: $(3.0 - 3.875)^2 + (5.0 - 3.875)^2 + (4.5 - 3.875)^2 + (3.0 - 3.875)^2$

     $$(-0.875)^2 + (1.125)^2 + (0.625)^2 + (-0.875)^2 = 0.765625 + 1.265625 + 0.390625 + 0.765625 = 3.1875$$
   - Taking square roots: $\sqrt{0.6875} \approx 0.8292$ and $\sqrt{3.1875} \approx 1.7855$.

3. **Calculate the Denominator**:
   - $U1$: Already calculated as 0.6875 (from the previous calculation).
   - $U4$: $(1.0 - 3.25)^2 + (5.0 - 3.25)^2 + (4.0 - 3.25)^2 + (3.0 - 3.25)^2$

     $(-2.25)^2 + (1.75)^2 + (0.75)^2 + (-0.25)^2 = 5.0625 + 3.0625 + 0.5625 + 0.0625 = 8.75$
   - Taking square roots: $\sqrt{0.6875} \approx 0.8292$ and $\sqrt{8.75} \approx 2.9580$.
4. **Final Calculation**:

$$\text{Pearson}(U1, U4) = \frac{1.625}{0.8292 \times 2.9580} \approx \frac{1.625}{2.453} \approx 0.662$$

$(4.0 - 4.375)(1.0 - 3.25) + (4.5 - 4.375)(5.0 - 3.25) + (5.0 - 4.375)(4.0 - 3.25) + (4.0 - 4.375)(3.0 - 3.25)$

Calculating each term:

$(-0.375)(-2.25) + (0.125)(1.75) + (0.625)(0.75) + (-0.375)(-0.25) = 0.84375 + 0.21875 + 0.46875 + 0.09375 = 1.625$

3. **Calculate the Denominator**:
   - $U1$: Already calculated as 0.6875 (from the previous calculation).
   - $U4$: $(1.0 - 3.25)^2 + (5.0 - 3.25)^2 + (4.0 - 3.25)^2 + (3.0 - 3.25)^2$

     $(-2.25)^2 + (1.75)^2 + (0.75)^2 + (-0.25)^2 = 5.0625 + 3.0625 + 0.5625 + 0.0625 = 8.75$
   - Taking square roots: $\sqrt{0.6875} \approx 0.8292$ and $\sqrt{8.75} \approx 2.9580$.
4. **Final Calculation**:

$$\text{Pearson}(U1, U4) = \frac{1.625}{0.8292 \times 2.9580} \approx \frac{1.625}{2.453} \approx 0.662$$

2.11 Results

In our case, using both Cosine Similarity and Pearson Correlation on the user-item matrix, we see the following:

- **Cosine Similarity**: Provides a similarity score based on the angle between user vectors. The results are relatively higher for pairs with more common ratings, emphasizing the overall pattern of ratings rather than the exact value differences. This makes it suitable when users rate items similarly in relative terms.
- **Pearson Correlation**: Adjusts for mean ratings and finds similarity based on the linear relationship of ratings after centering around each user's average. This technique can provide more correct recommendations in situations where users have different rating scales, as it normalizes ratings and removes biases.

- **Pros and Cons Summary**

Measure	Pros	Cons
Cosine Similarity	Simple to compute; handles sparse data well	Ignores individual user biases or rating scales
Pearson Correlation	Accounts for rating scale differences; considers linear trends	Requires sufficient data overlap; computationally intensive

cosine Similarity is preferred when working with sparse datasets and when users' rating scales are consistent. On the other hand, **Pearson Correlation** is more correct for cases with significant differences in users' rating patterns, but it requires a denser matrix to compute reliable correlations.

2.12 Compute the Rating Prediction and Top-N Recommendation List

For both **User-Based CF** and **Item-Based CF**, calculated:

- **Rating Prediction**: Predict ratings for items based on the similarities computed in Step 10 (using both Cosine Similarity and Pearson Correlation).
- **Top-N Recommendation**: Generate a list of top recommendations for each user.

1. **User-Based CF Prediction**:

- **Formula**: The predicted rating of a user U for an item I is calculated as:

$$\hat{r}_{UI} = \bar{r}_U + \frac{\sum_{V \in \text{Neighbors(U)}} \text{Sim}(U, V) \times (r_{VI} - \bar{r}_V)}{\sum_{V \in \text{Neighbors(U)}} |\text{Sim}(U, V)|}$$

  where $\text{Sim}(U, V)$ is the similarity between user U and user V, r_{VI} is the rating of user V for item I, and $\bar{r}_U$ and $\bar{r}_V$ are the average ratings for users U and V, respectively.

- **Similarity Metric**: Compute two sets of predictions, one using **Cosine Similarity** and the other using **Pearson Correlation** from Step 10.

2. **Item-Based CF Prediction**:

- **Formula**: The predicted rating of a user U for an item I is calculated as:

$$\hat{r}_{UI} = \bar{r}_I + \frac{\sum_{J \in \text{Neighbors(I)}} \text{Sim}(I, J) \times (r_{UJ} - \bar{r}_J)}{\sum_{J \in \text{Neighbors(I)}} |\text{Sim}(I, J)|}$$

  where $\text{Sim}(I, J)$ is the similarity between item I and item J, r_{UJ} is the rating of user U for item J, and $\bar{r}_I$ and $\bar{r}_J$ are the average ratings for items I and J, respectively.

- **Similarity Metric**: Again, use both **Cosine Similarity** and **Pearson Correlation**.

3. **Top-N Recommendation**:

- For each user, rank the items based on predicted ratings, and recommend the top N items (e.g., $N = 5$).

Rating prediction for missing values is not necessary for this dataset, Each user has rated all four items (296, 318, 356, and 593), meaning there are no missing ratings in this matrix

2.13 Briefly introduce the implementation process, tools, and libraries.

The implementation process for this assignment involved loading and preprocessing the dataset, computing similarity measures, creating a User-Item Matrix, and applying collaborative filtering algorithms. Tools and libraries used include:

- **Python** for scripting and implementation.
- **Pandas** for data manipulation and preprocessing.

- **NumPy** for mathematical operations.
- **Matplotlib** for visualizing rating distributions.
- **scikit-learn** for calculating similarity metrics such as cosine similarity and Pearson correlation.

2.14 Remarks on differences between user-based and item-based CF using similarity measure and Pearson correlation coefficient.

User-based and item-based Collaborative Filtering (CF) have unique advantages:

- **User-Based CF**: This approach finds similar users and recommends items that those similar users have liked. Cosine similarity and Pearson correlation can both capture similarities, but Pearson correlation considers rating patterns by centering around the mean, making it more effective when users have diverse rating scales.
- **Item-Based CF**: This approach identifies similar items based on user ratings and recommends items similar to those a user has liked. Cosine similarity works well for item-based CF by focusing on vector angle, whereas Pearson correlation can be useful when item ratings vary widely across users.

2.15 Conclusion on how each strategy affected predicted accuracy.

Each similarity strategy impacts accuracy differently:

- **Cosine Similarity** generally produces accurate recommendations for users/items with consistent rating scales. It's beneficial for cases where the magnitude of ratings (e.g., higher average ratings across items) is relevant.
- **Pearson Correlation** can improve accuracy in scenarios where users or items have variable rating scales by centering ratings around the mean. This method often provides more nuanced recommendations when user preferences are diverse.