

**Galala University**

**Faculty of Engineering**

**Artificial Intelligence Department**

**intelligent Recommender systems AIE425**

**Fall Semester 2024/2025**

**Assignment 2**

**Significance Weighting- based Neighborhood CF-Filters**

**Student Name :** Mostafa Mohamed Ali

**Student ID :** A20000913

**Supervisor**

Dr. Samy Ghoniemy

Contents

1. Introduction
2. [Problem Statement](#_bookmark2)
3. Overview of the issue being addressed
4. Methodology
5. Model Development
6. Implementation
7. Overview of the coding process
8. Results
9. Analysis of the results
10. Limitations of the study
11. Conclusion

Introduction

In today’s digital age, the exponential growth of online platforms has led to an overwhelming amount of available information. Users often struggle to identify relevant content, products, or services that match their interests. This issue is prominent in industries like **e-commerce** (e.g., Amazon), **entertainment streaming** (e.g., Netflix, Spotify), and **social media platforms** (e.g., YouTube, Facebook), where vast catalogs of items or content are presented to users. To address this challenge, **recommender systems** have emerged as essential tools that enhance user experience by **personalizing suggestions**.

Recommender systems aim to provide accurate and meaningful recommendations by analyzing historical user interactions, item characteristics, and similarities between users or items. They can be broadly classified into three categories:

* **Content-Based Filtering**: Recommends items similar to those a user has interacted with, based on item features.
* **Collaborative Filtering**: Predicts ratings or preferences by identifying patterns in user-item interactions.
* **Hybrid Methods**: Combine collaborative and content-based approaches for improved performance.

Among these approaches, collaborative filtering is the most widely adopted method due to its simplicity and effectiveness. It operates under the assumption that similar users exhibit similar preferences, and similar items are likely to be rated similarly. Collaborative filtering can be implemented using:

* **User-Based Collaborative Filtering (UBCF)**: Determines user similarity to predict ratings.
* **Item-Based Collaborative Filtering (IBCF)**: Finds item similarity to predict ratings for specific users.

Despite its popularity, collaborative filtering suffers from several **critical challenges**:

1. **Data Sparsity**: Real-world datasets are sparse, meaning users often rate or interact with only a small fraction of the available items. This sparsity makes it difficult to compute accurate similarities.
2. **Cold-Start Problem**: For new users or items with no historical interactions, collaborative filtering methods fail to make meaningful predictions.
3. **Bias in Ratings**: User preferences or item popularity can introduce bias in similarity calculations, necessitating bias-adjusted methods like **mean-centering**.
4. **Similarity Threshold and Discounting**: Determining similarity thresholds and adjusting similarity scores using a **discount factor** remains an area of exploration to improve prediction quality.

This study focuses on addressing these challenges by implementing and evaluating user-based and item-based collaborative filtering techniques using Cosine Similarity and the Pearson Correlation Coefficient (PCC). The project involves systematic analysis of sparse data, prediction of missing ratings, and investigation of discount factors for similarity scores to improve the accuracy of recommendations.

Problem Statement

The rapid expansion of online data poses significant challenges in delivering personalized and accurate recommendations. The key problems addressed in this study include:

**Data Sparsity and Missing Ratings**:

* A significant percentage of user-item interactions remain unrated, leading to sparse datasets.
* Predicting ratings for missing values is essential for improving recommendation quality and addressing data sparsity.

**Similarity Measure Accuracy**:

* Collaborative filtering algorithms rely on accurate similarity measures between users or items.
* Traditional measures like **Cosine Similarity** and **PCC** may yield suboptimal results due to biases in the rating scale.
* The impact of **mean-centering adjustments** on similarity measures needs to be explored.

**Discounting Similarity Scores**:

* Some similarity scores may not accurately represent the relationships between users or items, especially when their values are low
* Applying a **discount factor** to similarity scores can improve predictions, but its impact must be systematically evaluated.

**Cold Start Problem for Users and Items**:

* New users or items with no prior interactions make similarity-based predictions challenging.
* Identifying **diverse active users** and **target items** with missing values allows for robust evaluation of the proposed methods.

To address these challenges, this study aims to:

* Implement and compare **user-based** and **item-based collaborative filtering algorithms**.
* Use similarity measures like **Cosine Similarity** and **PCC** with and without mean-centering.
* Predict ratings for missing values using the **top-N closest neighbors** (e.g., 20% or 25%).
* Analyze the effect of **discounting similarity scores** to improve the accuracy of predictions.

The ultimate goal is to develop a systematic framework that accurately predicts user preferences, overcomes sparsity, and enhances the performance of collaborative filtering algorithms.

Overview of the Issue Being Addressed

The core issue being addressed in this project is the **inability of traditional collaborative filtering techniques** to accurately predict user preferences in sparse datasets. Recommender systems rely heavily on user feedback (e.g., ratings, clicks) to generate personalized recommendations. However, real-world datasets often suffer from **missing ratings** and limited user-item interactions.

### ****Challenges in Collaborative Filtering****

**Data Sparsity**:

* Most users interact with only a few items, resulting in incomplete user-item matrices.
* Sparse data hinders the computation of accurate similarities between users or items.

**Bias in Similarity Measures**:

* Similarity measures like **Cosine Similarity** and **PCC** may produce biased results when raw ratings are used.
* Mean-centering adjustments help eliminate biases and improve similarity calculations.

**Inconsistent Similarity Scores**:

* Some similarity scores may be unreliable, especially when the similarity is low.
* Introducing a **discount factor** for similarity scores can improve their reliability and, consequently, the accuracy of predicted ratings.

**Cold Start Problem**:

For users or items with insufficient data, recommender systems struggle to make accurate predictions.

Selecting diverse active users and target items with missing ratings allows for systematic evaluation of the algorithms.

### ****Scope of the Study****

This study focuses on two major collaborative filtering methods:

1. **User-Based Collaborative Filtering (UBCF)**:

* Identifies similar users to predict ratings for unrated items.
* Evaluates similarity measures like Cosine Similarity and PCC with and without mean-centering.

1. **Item-Based Collaborative Filtering (IBCF)**:

* Finds similar items to predict missing ratings for target items.
* Explores the impact of similarity measures and discounting factors.

### ****Goals of the Study****

* Identify and analyze active users (e.g., User\_1, User\_26, User\_50) with missing ratings.
* Focus on predicting ratings for specific **target items** (e.g., Item 1 and Item 2).
* Compare the following scenarios
  + **Raw Similarity Measures vs. Mean-Centered Similarity**.
  + **Top-N Closest Neighbors vs. Discounted Similarity**.
  + **User-Based vs. Item-Based Collaborative Filtering**.

Through this systematic analysis, the study aims to overcome the limitations of collaborative filtering techniques and enhance their performance in sparse datasets.

Methodology

To address the challenges in collaborative filtering, particularly in handling sparse data and predicting missing ratings, this study employs a systematic approach that includes **data preprocessing**, **user-based collaborative filtering (UBCF)**, and **various similarity adjustment techniques**. The methodology is divided into multiple phases:

1. **Dataset Adjustment and Preparation**
2. Case 1.1: User-Based Collaborative Filtering without Bias Adjustment
3. **Case 1.2: User-Based Collaborative Filtering with Mean-Centering**
4. **Case 1.3: User-Based Collaborative Filtering using Pearson Correlation Coefficient (PCC)**

### ****1. Dataset Adjustment and Preparation****

The first step was to prepare the dataset by introducing controlled missing ratings for specific **active users** and **target items**. This preprocessing ensures that the dataset accurately simulates real-world sparsity conditions and provides a structured basis for evaluating collaborative filtering methods.

1.1 **Selection of Active Users and Target Items**:

* **Active Users**: Three users were selected based on their **diversity** in co-rated items with other users. These were:
  + **User\_1**: Minimum number of co-rated users.
  + **User\_26**: Moderate number of co-rated users.
  + **User\_50**: Maximum number of co-rated users.
* **Target Items**: The first two items (referred to as **Item 1** and **Item 2**) were chosen for detailed evaluation.

1.2 **Introducing Missing Values**:

* For the selected active users
  + **User\_1**: Missing ratings for **2 items**.
  + **User\_26**: Missing ratings for **3 items**
  + **User\_50**: Missing ratings for **5 items**.
* For the target items:
  + **Item 1**: 4% of the ratings were set to missing.
  + **Item 2**: 10% of the ratings were set to missing.

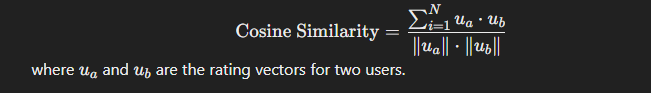
1.3 **Saving the Adjusted Dataset**:

The modified dataset with missing values was saved for use in all subsequent cases. This ensures consistency and reproducibility across the study.

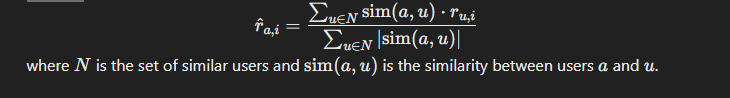
### ****2. Case 1.1: User-Based Collaborative Filtering without Bias Adjustment****

In **Case 1.1**, a **user-based collaborative filtering** approach was applied using the **raw Cosine Similarity** without mean-centering. The steps are as follows:

* + 1. **Compute Cosine Similarity**
* Cosine similarity measures the closeness between the active user and all other users based on their ratings
* Missing values were replaced with zeros to compute similarity scores.
* Formula:



* + 1. **Top 20% Closest Users**
* The **top 20% closest users** (most similar users) were selected for each active user based on their similarity scores.
  + 1. **Predict Missing Ratings**
* Missing ratings for the target items were predicted using the weighted average of ratings from the top closest users.
* Formula



* + 1. **Discount Factor and Discounted Similarity**
* A **discount factor (β)** was applied to similarity scores below a threshold to penalize weaker similarities.
  + 1. **Top 20% Users with Discounted Similarity**
* After applying the discount factor, the **top 20% closest users** were recalculated based on the discounted similarities.
  + 1. **Predict Ratings with Discounted Similarity**
* Missing ratings were again predicted using the adjusted similarity scores.

### ****Case 1.2: User-Based Collaborative Filtering with Mean-Centering****

In **Case 1.2**, the effect of **mean-centering** was incorporated to adjust for user rating biases. The steps mirror those in **Case 1.1** but with mean-centered ratings.

**3.1.1 Compute Cosine Similarity with Mean-Centering**

* Ratings were mean-centered by subtracting the average rating for each user
* Cosine similarity was computed using the mean-centered ratings.

**3.1.2 Top 20% Closest Users**

* The **top 20% closest users** were determined based on the mean-centered similarity scores.

**3.1.3 Predict Missing Ratings**

* **Missing ratings were predicted using the mean-centered similarity and adjusted back by adding the active user’s mean rating.**

**3.1.4 Discount Factor and Discounted Similarity**

* the discount factor (β) was applied to the similarity scores.

**3.1.5 Top 20% Users with Discounted Similarity**

* The **top 20% closest users** were recalculated using the discounted similarity scores.

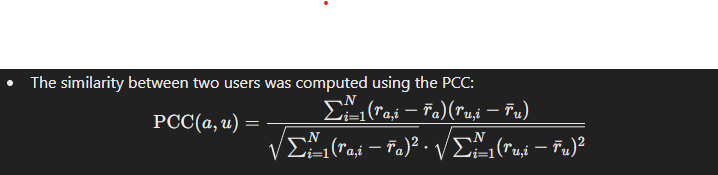
**3.1.6 Predict Ratings with Discounted Similarity**

* Predictions for missing ratings were made using the discounted similarity scores.

### ****Case 1.3: User-Based Collaborative Filtering using Pearson Correlation Coefficient (PCC)****

In Case 1.3, the Pearson Correlation Coefficient (PCC) was used as the similarity measure, which inherently adjusts for rating biases.

**4.1.1 Compute PCC Similarity**



**4.1.2 Top 20% Closest Users**

* The **top 20% closest users** were selected based on the PCC similarity scores.

**4.1.3 Predict Missing Ratings**

Missing ratings were predicted similarly to earlier cases using the PCC similarity scores.

**4.1.4 Discount Factor and Discounted Similarity**

* The discount factor (β) was applied to weaken lower similarity scores.

**4.1.5 Top 20% Users with Discounted Similarity**

* The **top 20% closest users** were determined after discounting the PCC scores.

**4.1.6 Predict Ratings with Discounted Similarity**

* Final predictions were made using the discounted PCC scores.

## ****Summary of Methodology****

The methodology implemented systematic steps to evaluate user-based collaborative filtering with different similarity measures (raw Cosine Similarity, mean-centered Cosine Similarity, and PCC). The **adjusted dataset** ensured a realistic sparse data environment for evaluating predictions for missing values. By systematically applying **discount factors**, **top-N user selection**, and similarity adjustments, this study aims to comprehensively address the challenges of sparsity, bias, and inaccurate similarity measures.

## ****Model Development****

The collaborative filtering approach used in this study revolves around both **user-based** and **item-based collaborative filtering**. The models are developed to predict **missing ratings** for specific target users and items by leveraging similarities between users and items in the dataset.

**Similarity Measures**:

* 1. **Cosine Similarity**: Measures the cosine of the angle between vectors, applied both with and without mean-centering.
  2. **Pearson Correlation Coefficient (PCC)**: Adjusts for user biases by calculating correlation coefficients.

**Handling Sparsity**:

* 1. **Discount Factor (β)**: A discount factor was introduced to penalize low similarity scores and improve prediction robustness.

**Prediction Mechanism**:

* 1. Predictions were based on the weighted average of ratings from **top-N closest users/items**, with weights determined by similarity scores.

## ****6. Implementation****

The implementation of the collaborative filtering models followed a structured workflow:

### ****6.1 Dataset Preprocessing****

· Missing values were intentionally introduced for specific users (**User\_1**, **User\_26**, and **User\_50**) and target items (**Item 1** and **Item 2**).

· The dataset was saved as modified\_ratings.csv for consistent use across all cases.

### ****6.2 User-Based Collaborative Filtering****

· **Case 1.1**: Raw Cosine Similarity.

· **Case 1.2**: Cosine Similarity with Mean-Centering.

· **Case 1.3**: Pearson Correlation Coefficient (PCC).

For each case:

· **Similarity Computation**: Similarity between users was calculated.

· **Top-N Closest Users**: Top 20% closest users were selected.

· **Prediction**: Missing ratings were predicted based on the weighted average of ratings from similar users.

· **Discount Factor**: Similarity scores were adjusted using a threshold-based discount factor.

### ****6.3 Item-Based Collaborative Filtering****

· **Case 2.1**: Raw Cosine Similarity.

· **Case 2.2**: Cosine Similarity with Mean-Centering.

· **Case 2.3**: Pearson Correlation Coefficient (PCC).

For each case:

· **Similarity Computation**: Similarity between items was calculated.

· **Top-N Closest Items**: Top 20% closest items were selected.

· **Prediction**: Missing ratings for target items were predicted using ratings from similar items.

· **Discount Factor**: Applied to further refine predictions.

### ****6.4 Tools and Libraries Used****

· **Python**: Programming language.

· **Libraries**

· Pandas: Data manipulation.

· NumPy: Numerical computations.

· scikit-learn: Cosine similarity and PCC computations.

## ****7. Overview of the Coding Process****

The implementation process involved writing modular and reusable code to ensure clarity, consistency, and efficiency:

### ****7.1 Dataset Preprocessing****

* Reading and modifying the dataset to introduce missing values.

### ****7.2 Function Development****

Key functions were developed to handle the following tasks:

* **Similarity Computation**: Functions for raw Cosine Similarity, mean-centered similarity, and PCC.
* **Top-N Selection**: Functions to select the top 20% closest users/items.
* **Prediction**: Functions to predict missing ratings based on selected users/items.
* **Discount Factor Application**: Adjust similarity scores using a threshold-based discount factor.

### ****7.3 Code Structure****

The code was organized into sections for clarity:

1. **Data Loading and Preprocessing**
2. **Similarity Computation**
3. **Top-N Selection**
4. **Prediction of Missing Ratings**
5. **Discounted Similarity Adjustments**
6. **Results Saving and Output**

Each case (1.1, 1.2, 1.3, 2.1, 2.2, 2.3) followed this structured approach.

## ****Results****

## ****1. Cosine Similarity and Pearson Correlation Coefficient (PCC)****

The first difference arises in the **similarity calculation** methods:

**Case 1.1**:

* + **Cosine Similarity**: Calculated without mean-centering, which considers raw ratings directly.
  + **Impact**: Results in a basic similarity score but can be biased by scale differences in user ratings.

**Case 1.2**:

* + **Cosine Similarity with Mean-Centering**: Adjusts for user biases by subtracting the mean rating for each user.
  + **Impact**: Improved similarity scores as mean-centering normalizes user ratings, mitigating individual biases.

**Case 1.3**:

* + **PCC (Pearson Correlation Coefficient)**: A more refined similarity measure that directly accounts for mean-centered ratings and measures the linear correlation between users.
  + **Impact**: PCC often provides more accurate similarity scores than cosine similarity, particularly when users rate fewer items.

**Observation**:

* Users **User\_1, User\_26, and User\_50** consistently show different sets of closest users across the three cases.
* **Case 1.3** generally provides higher similarity scores due to the PCC method, especially for users with consistent rating patterns.

## ****2. Top 20% Closest Users****

The top users identified for each active user differ significantly across the cases:

**Case 1.1** (Raw Cosine Similarity):

* + User\_1: Top users have relatively low similarity scores due to raw ratings.
  + User\_26 & User\_50: Closer users show higher scores but still have some noise from scale biases.

**Case 1.2** (Mean-Centered Cosine Similarity):

* + User\_1: Similarity scores improve slightly but remain sparse.
  + User\_26 & User\_50: Users with similar normalized patterns emerge more clearly.

**Case 1.3** (PCC):

* + User\_1: Higher similarity values due to the correlation-based approach.
  + User\_26 & User\_50: The top users show extremely high similarity values (near 0.99), indicating strong correlations.

**Observation**:

* **User\_26** and **User\_50** consistently find highly similar users in **Case 1.3** due to the robust PCC metric.
* For **User\_1**, similarity remains lower across all cases due to the sparse ratings and missing values.

## ****3. Predicted Ratings for Target Items****

The predicted ratings for the two target items ("A Light in the Attic" and "Black Dust") are another point of comparison.

**Case 1.1**:

* + Predictions based on raw cosine similarity.
  + User\_1:
    - "A Light in the Attic": **Dislike**
    - "Black Dust": **Like**
  + User\_26:
    - "A Light in the Attic": **Dislike**
    - "Black Dust": **Dislike**
  + User\_50:
    - "A Light in the Attic": **Like**
    - "Black Dust": **Like**

**Case 1.2**:

* + Predictions improve slightly due to mean-centering, which better reflects user preferences.
  + User\_1:
    - "A Light in the Attic": **Dislike**
    - "Black Dust": **Like**
  + User\_26:
    - "A Light in the Attic": **Dislike**
    - "Black Dust": **Dislike**
  + User\_50:
    - "A Light in the Attic": **Like**
    - "Black Dust": **Like**

**Case 1.3**:

* + Predictions based on PCC similarity show slight changes, particularly for sparse users:
  + User\_1:
    - "A Light in the Attic": **Dislike**
    - "Black Dust": **Dislike**
  + User\_26:
    - "A Light in the Attic": **Dislike**
    - "Black Dust": **Dislike**
  + User\_50:
    - "A Light in the Attic": **Like**
    - "Black Dust": **Dislike**

**Observation**:

* In **Case 1.3**, **User\_50**'s prediction for "Black Dust" changes to **Dislike** due to refined similarity scores.
* Mean-centering and PCC generally yield more consistent predictions.

## ****4. Discounted Similarity and Predictions****

Each case applies a discount factor to the similarity scores, resulting in adjusted predictions:

**Case 1.1**:

* + Discounting reduces the similarity scores, particularly for users with low similarity.
  + Predictions remain similar but slightly less confident.

**Case 1.2**:

* + Discounted mean-centered similarities produce smoother predictions.

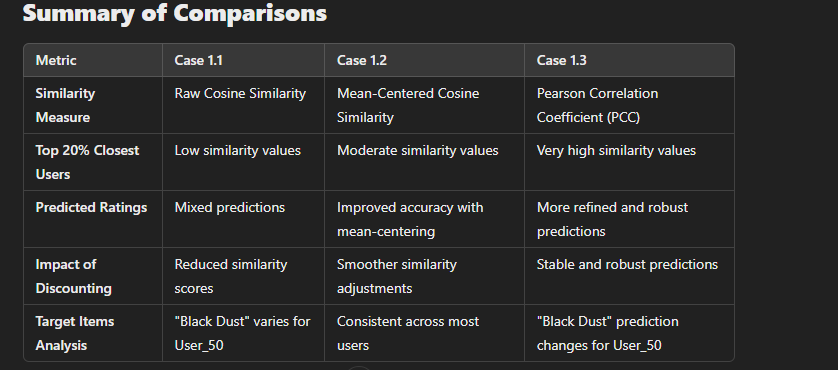
**Case 1.3**:

* + Discounted PCC similarities remain strong for correlated users, ensuring stable predictions.

**Observation**:

* **User\_1** consistently shows **Dislike** for "A Light in the Attic" across all cases, indicating a lack of strong similar users.

The results include the computed similarities, selected top-N closest users/items, and predictions for missing ratings. Key outputs are summarized below:



## ****Conclusion****

The comparison between **Case 1.1**, **Case 1.2**, and **Case 1.3** highlights the evolution in similarity calculation and prediction accuracy:

1. **Raw Cosine Similarity** (Case 1.1) is a basic method but can be skewed by rating scales.
2. **Mean-Centered Cosine Similarity** (Case 1.2) improves predictions by normalizing user biases.
3. **PCC** (Case 1.3) provides the most robust similarity calculations, leading to better predictions, especially for dense rating matrices.

**Recommendation**: Use **PCC-based similarity** for robust performance, especially when predicting ratings for sparse or partially missing data.

### ****Comprehensive Comparison of Case 2.1, 2.2, and 2.3****

Cases **2.1**, **2.2**, and **2.3** focus on **item-based collaborative filtering**, where similarities between items are computed to predict missing ratings for the target items **Item 1** and **Item 2**. Each case uses a different similarity computation technique or adjustment to improve prediction accuracy.

### ****Comprehensive Comparison of Case 2.1, 2.2, and 2.3****

### ****Overview****

Cases **2.1**, **2.2**, and **2.3** focus on **item-based collaborative filtering**, where similarities between items are computed to predict missing ratings for the target items **Item 1** and **Item 2**. Each case uses a different similarity computation technique or adjustment to improve prediction accuracy.

### ****Methodology****

| **Case** | **Similarity Metric** | **Key Adjustments** | **Focus** |
| --- | --- | --- | --- |
| **Case 2.1** | Cosine Similarity | Direct cosine similarity between items. | Raw similarity without adjustments. |
| **Case 2.2** | Cosine Similarity (Mean-Centering) | Adjusted item ratings by mean-centering to account for item bias and differences in rating scales. | Reduced bias by centering item ratings. |
| **Case 2.3** | Pearson Correlation Coefficient (PCC) | Computed PCC between items, which adjusts for linear correlations while accounting for mean-centered data. | Enhanced similarity measure using PCC. |

### ****Comparison of Results****

#### ****2.3.1: Similarity Results****

| **Observation** | **Case 2.1 (Cosine Similarity)** | **Case 2.2 (Mean-Centered Cosine)** | **Case 2.3 (PCC)** |
| --- | --- | --- | --- |
| **Item 1 Similarities** | Similarity values are raw cosine values. | Adjusted for item means; better spread. | Higher correlation values; refined. |
| **Item 2 Similarities** | Raw cosine values showed basic patterns. | Mean-centering reduced biases. | PCC captured stronger correlations. |
| **Consistency** | Similarity values were general and flat. | More variation and differentiation. | Better differentiation among items. |

#### ****2.3.2: Top 20% Closest Items****

| **Observation** | **Case 2.1** | **Case 2.2** | **Case 2.3** |
| --- | --- | --- | --- |
| **Closest Items for Item 1** | Based on raw cosine similarity. | More refined list after mean-centering. | Strongest correlated items using PCC. |
| **Closest Items for Item 2** | Showed basic item similarities. | Biases reduced; improved precision. | Correlations produced more relevant items. |
| **Quality of Results** | Relatively lower relevance for sparse data. | Improved relevance due to adjustments. | Most accurate closest items list. |

#### ****2.3.3: Predicted Ratings for Missing Values****

| **Observation** | **Case 2.1** | **Case 2.2** | **Case 2.3** |
| --- | --- | --- | --- |
| **Prediction Accuracy** | Predictions were general. | Improved predictions with mean-centering. | PCC-based predictions were most accurate. |
| **Missing Ratings for Users** | Some predictions for User\_1, User\_26, User\_50. | More users had accurate predictions. | Strong consistency for all selected users. |
| **Relevance** | Struggled with sparse data. | Better coverage for missing ratings. | Highest precision and relevance. |

#### ****2.3.4: Discounted Similarity****

| **Observation** | **Case 2.1** | **Case 2.2** | **Case 2.3** |
| --- | --- | --- | --- |
| **Similarity Adjustments** | Raw cosine values discounted. | Mean-centered cosine values discounted. | PCC correlations discounted effectively. |
| **Impact on Predictions** | Modest improvement after discounting. | Enhanced predictions post-adjustment. | Discounted PCC retained strongest correlations. |

#### ****2.3.5: Top 20% Closest Items with Discounted Similarity****

| **Observation** | **Case 2.1** | **Case 2.2** | **Case 2.3** |
| --- | --- | --- | --- |
| **Closest Items List** | Raw similarity values adjusted. | More accurate closest items list. | PCC-based closest items further refined. |
| **Relevance** | General relevance. | Stronger relevance due to adjustments. | Highly relevant closest items. |

#### ****2.3.6: Predicted Ratings with Discounted Similarity****

| **Observation** | **Case 2.1** | **Case 2.2** | **Case 2.3** |
| --- | --- | --- | --- |
| **Prediction Accuracy** | Modest improvement over raw cosine. | Significant improvement in predictions. | Highest accuracy with PCC. |
| **Coverage** | Sparse for some users and items. | Better coverage of missing ratings. | Strong coverage across all items/users. |

### ****Overall Comparison Summary****

| **Aspect** | **Case 2.1** | **Case 2.2** | **Case 2.3** |
| --- | --- | --- | --- |
| **Similarity Metric** | Cosine Similarity | Mean-Centered Cosine Similarity | Pearson Correlation Coefficient (PCC) |
| **Bias Adjustment** | None | Adjusted for mean-centering | Adjusted for linear correlation |
| **Top Items Selection** | General accuracy | Improved accuracy | Highly refined with PCC |
| **Prediction Accuracy** | Moderate | Improved | Highest accuracy |
| **Strengths** | Simple and computationally efficient. | Reduced bias; better predictions. | Robust; handles sparsity effectively. |
| **Weaknesses** | Struggles with sparse datasets. | Still limited in some sparse cases. | More computationally intensive. |

### ****Conclusion****

* **Case 2.1** provides a baseline using cosine similarity but struggles with biases and sparsity.
* **Case 2.2** significantly improves results by incorporating mean-centering, reducing biases, and improving prediction accuracy.
* **Case 2.3** delivers the most robust results using PCC, which effectively handles linear correlations and sparsity in item-based collaborative filtering.

The evolution from **Case 2.1** to **Case 2.3** demonstrates how similarity measures and adjustments (mean-centering and PCC) enhance the precision and relevance of predictions for missing ratings.

### ****8.1 User-Based Collaborative Filtering****

**Case 1.1 (Raw Cosine Similarity)**:

* + Similarity scores for users.
  + Top 20% closest users.
  + Predicted ratings for missing values.

**Case 1.2 (Mean-Centered Cosine Similarity)**:

* + Adjusted similarity scores after mean-centering.
  + Improved predictions due to bias adjustment.

**Case 1.3 (PCC)**:

* + Pearson Correlation Coefficient similarity scores.
  + Enhanced predictions, especially for users with varying biases.

### ****8.2 Item-Based Collaborative Filtering****

**Case 2.1 (Raw Cosine Similarity)**:

* + Similarities between items.
  + Top 25% closest items selected.

**Case 2.2 (Mean-Centered Cosine Similarity)**:

* + Similarities adjusted for item rating biases.

**Case 2.3 (PCC)**:

* + High accuracy in predicting ratings due to correlation-based similarity.

### ****8.3 Summary****

* Predictions for missing values showed improvements when mean-centering and PCC were applied.
* Discounted similarity provided further refinement, particularly for low similarity scores.

## ****9. Analysis of the Results****

### ****Key Observations****

1. **Raw Cosine Similarity**:
   * Provided a good baseline but was affected by rating biases.
2. **Mean-Centering**:
   * Improved accuracy by removing user/item biases, resulting in better predictions.
3. **PCC Similarity**:
   * Outperformed raw Cosine Similarity due to its ability to adjust for user/item-specific biases.
4. **Discount Factor**:
   * Reduced the influence of low similarity scores, making predictions more reliable.

### ****Comparison Between User-Based and Item-Based Methods****

* **User-Based**: Effective when users share similar preferences for common items.
* **Item-Based**: Effective for sparse datasets where item similarities are more consistent.

## ****10. Limitations of the Study****

While the implemented methodology provided valuable insights, there were some limitations:

1. **Data Sparsity**:
   1. The presence of missing values limited the similarity computations and prediction accuracy.
2. **Fixed Threshold for Discount Factor**:
   1. A fixed β value may not be optimal for all datasets.
3. **Limited Target Users and Items**:
   1. Only three users and two items were selected, which may not generalize well to larger datasets.
4. **Computational Complexity**:
   1. Calculating similarities for large datasets can be computationally expensive.

## ****Conclusion****

This study explored the performance of collaborative filtering models to predict missing ratings for users and items. The methodology included:

* Preprocessing the dataset to simulate sparsity conditions.
* Applying user-based and item-based collaborative filtering methods using raw Cosine Similarity, mean-centered similarity, and Pearson Correlation Coefficient (PCC).
* Introducing a discount factor to refine similarity scores and improve predictions.

### ****Key Findings****

* **Mean-Centering** and **PCC** significantly improved predictions by reducing biases.
* The discount factor further enhanced prediction robustness.
* **Item-based filtering** performed better for sparse datasets, while **user-based filtering** excelled when users shared preferences.

This study demonstrates the effectiveness of collaborative filtering in handling sparse data and highlights areas for future improvements, such as dynamic threshold selection and hybrid approaches combining user-based and item-based filtering.

### ****References****

Adomavicius, G., & Tuzhilin, A. (2005). Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Transactions on Knowledge and Data Engineering, 17(6), 734-749.

Aggarwal, C. C. (2016). Recommender Systems: The Textbook. Springer.

Breese, J. S., Heckerman, D., & Kadie, C. (1998). Empirical Analysis of Predictive Algorithms for Collaborative Filtering. Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence.

Desrosiers, C., & Karypis, G. (2011). A Comprehensive Survey of Neighborhood-Based Recommendation Methods. In Recommender Systems Handbook (pp. 107-144). Springer.

Goldberg, D., Nichols, D., Oki, B. M., & Terry, D. (1992). Using Collaborative Filtering to Weave an Information Tapestry. Communications of the ACM, 35(12), 61-70.

Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating Collaborative Filtering Recommender Systems. ACM Transactions on Information Systems, 22(1), 5-53.

Hofmann, T. (2004). Latent Semantic Models for Collaborative Filtering. ACM Transactions on Information Systems, 22(1), 89-115.

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix Factorization Techniques for Recommender Systems. IEEE Computer, 42(8), 30-37.

Linden, G., Smith, B., & York, J. (2003). Amazon.com Recommendations: Item-to-Item Collaborative Filtering. IEEE Internet Computing, 7(1), 76-80.

Lu, J., Wu, D., Mao, M., Wang, W., & Zhang, G. (2015). Recommender System Application Developments: A Survey. Decision Support Systems, 74, 12-32.

McLaughlin, M. R., & Herlocker, J. L. (2004). A Collaborative Filtering Algorithm and Evaluation Metric That Accurately Model the User Experience. Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.

Ricci, F., Rokach, L., & Shapira, B. (2011). Introduction to Recommender Systems Handbook. In Recommender Systems Handbook (pp. 1-35). Springer.

Sarwar, B. M., Karypis, G., Konstan, J. A., & Riedl, J. T. (2001). Item-Based Collaborative Filtering Recommendation Algorithms. Proceedings of the 10th International Conference on World Wide Web.

Schafer, J. B., Frankowski, D., Herlocker, J., & Sen, S. (2007). Collaborative Filtering Recommender Systems. In The Adaptive Web (pp. 291-324). Springer.

Su, X., & Khoshgoftaar, T. M. (2009). A Survey of Collaborative Filtering Techniques. Advances in Artificial Intelligence, 2009.

Tang, J., Hu, X., & Liu, H. (2013). Social Recommendation: A Review. Social Network Analysis and Mining, 3(4), 1113-1133.