**AIE425 Intellegince Recommender System Fall semester 2024/2025**

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

**Student Name: Sondos Ahmed Ghoniem, Student ID: 221101426**

Week6

**Core idea**

The assignment aims to analyze, design, and implement an intelligent recommender system, understanding the mechanisms of recommendation algorithms, developing a model to predict user preferences, and evaluating its performance in providing personalized recommendations. The goal is to apply theoretical knowledge in a practical context, enhancing understanding of these systems' real-world applications.

**2.3 Assignment requirements and questions**

1.E-commerce: Amazon, eBay, Esty

Entertainment: Netflix, IMDb

2. Netflix: provides personalized movie and tv show recommendations based on user viewing history and preferences.

Amazon: recommends products based on user browsing purchasing behavior as well as popular items purchased by other users.

3. Netflix: collect feedback through viewing history, user ratings. Interactions with recommended content and likes dislikes, it uses these inputs to improve recommendations.

Amazon: allowing users to rate products from 1 to 5 stars, or written reviews. This rating data helps amazon recommend with higher ratings and relevance.

4.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Movie Id/user  Id | 53000 | 2539 | 2455 | 215 | 49013 | 1033 | 786 | 833 |
| 533 | 5.0 | 4.0 | 2.0 | NaN | 1.0 | NaN | 3.0 | 3.0 |
| 244 | 3.0 | 1.0 | 3.0 | 1.0 | 2.0 | 3.0 | 3.0 | 3.0 |
| 651 | 3.0 | 2.0 | 4.0 | 2.0 | 2.0 | 3.0 | 3.0 | 3.0 |
| 653 | 5.0 | 5.0 | 2.0 | 1.0 | 2.0 | 5.0 | 1.0 | 1.0 |
| 210 | 4.0 | 4.0 | 5.0 | 2.0 | 2.0 | 2.0 | 3.0 | 5.0 |
| 410 | 1.0 | 5.0 | 2.0 | 4.0 | 4.0 | 1.0 | 1.0 | 4.0 |
| 320 | 5.0 | 5.0 | 5.0 | 3.0 | NaN | 5.0 | 1.0 | 2.0 |
| 292 | 4.0 | 4.0 | NaN | 1.0 | 5.0 | 4.0 | 1.0 | 4.0 |

The dataset consists of rows (Users) and columns (Movies), representing different users who have rated various movies. The values in the matrix represent the ratings given by users to movies, with a NaN indicating that the user has not rated that particular movie.  
  
The Cosine Similarity method yielded a slightly higher similarity score (≈ 0.959) compared to Pearson Correlation (≈ 0.837) when comparing the similarity between User 400 and User 268. This difference suggests that cosine similarity identifies a closer alignment in rating patterns between the two users despite possible differences in rating scale. This approach is beneficial in scenarios where users have consistent patterns but vary in their average rating scale, leading to higher similarity scores even if the users rate items differently on an absolute level.  
  
On the other hand, Pearson Correlation is lower because it takes into account each user's rating tendency, which may result in lower similarity scores when there's a lack of perfect correlation or when users rate items similarly but with different levels of enthusiasm or bias.

The dataset consists of rows (Users) and columns (Movies), representing different users who have rated various movies. The values in the matrix represent the ratings given by users to movies, with a NaN indicating that the user has not rated that particular movie.  
  
The Cosine Similarity method yielded a slightly higher similarity score (≈ 0.959) compared to Pearson Correlation (≈ 0.837) when comparing the similarity between User 400 and User 268. This difference suggests that cosine similarity identifies a closer alignment in rating patterns between the two users despite possible differences in rating scale. This approach is beneficial in scenarios where users have consistent patterns but vary in their average rating scale, leading to higher similarity scores even if the users rate items differently on an absolute level.  
  
On the other hand, Pearson Correlation is lower because it takes into account each user's rating tendency, which may result in lower similarity scores when there's a lack of perfect correlation or when users rate items similarly but with different levels of enthusiasm or bias.

**Cosine similarity, Pearson correlation**

The cosine similarity test, which measures the angle between two users' rating vectors, The result of 0.916 indicates a high level of similarity between User A and User B. This method does not consider individual rating scales, indicating that even if one user rates higher or lower than the other, their ratings on common items align directionally.

The Pearson results of 0.416 indicates a moderate positive correlation between User A and User B, suggesting that despite sharing some common preferences, their overall rating tendencies slightly differ. This is because Pearson correlation considers individual rating tendencies, making it more sensitive to differences in rating patternxs.

|  |  |  |
| --- | --- | --- |
| Measure | Pros | Cons |
| Cosine similarity | This method is easy to compute and interpret, does not require mean-centering, and is effective with sparse data. | The system is less sensitive to individual rating biases and ignores differences in users' rating scales. |
| Pearson correlation | The system adjusts user ratings by focusing on the mean and offers a more accurate measure for users using different rating scales. | Mean-centering makes computation more complex, making it sensitive to outliers and extreme ratings. It may not perform well with sparse data. |

**Rating prediction**

The Top-N recommendation list for User 533 is generated using user-based CF, cosine similarity, and Pearson correlation, completing the task of generating recommendations based on predicted ratings.

**Implementation Process**

The process of implementing a collaborative filtering recommendation system involves several steps. These include data collection and preprocessing, which involves obtaining user-item interaction data in the form of a matrix, and preprocessing it to handle missing values, normalize it, and convert it to a usable format for calculations. Similarity calculations are performed using Cosine Similarity or Pearson Correlation Coefficient to identify the similarity between users or items based on their ratings. Prediction computation is then used to generate rating predictions, which are based on the ratings of similar users or items. Finally, the predicted ratings are used to create a ranked list of recommended items for each user or set of users.

**Suggested Enhancements**

The text suggests several enhancements to improve prediction accuracy and system performance. These include a hybrid recommendation system that combines user-based and item-based collaborative filtering with similarity measures, allowing for better personalized results. Weighted similarity models, which balance the strengths of each approach, can make the system more adaptable to different datasets. Techniques like Singular Value Decomposition (SVD) or Alternating Least Squares (ALS) can reduce sparsity issues in user-based collaborative filtering, improving accuracy. Additionally, incorporating contextual factors like item attributes or user demographics can enhance the model's understanding of user preferences and recommendation quality.

**results**

**average rating**

|  |
| --- |
| 533 3.000000 |
| 244 2.375000 |
| 651 2.750000 |
| 653 2.750000 |
| 210 3.375000 |
| 410 2.750000 |
| 320 3.714286 |
| 292 3.285714 |

**Cosine similarity:**

|  |  |  |
| --- | --- | --- |
| item | User A rating | User B rating |
| 1 | 3 | 4 |
| 2 | 5 | 3 |
| 3 | 4 | 5 |
| 4 | 2 | 2 |
| 5 | 1 | 3 |

∑(ai​⋅bi​) = (3×4) +(5×3) +(4×5) +(2×2) +(1×3) = 54

Denominator= 7.42

Cos similarity =

**Pearson correlation:**

|  |  |  |
| --- | --- | --- |
| Item | User A rating (Ai) | User B rating (Bi) |
| 1 | 3 | 4 |
| 2 | 5 | 3 |
| 3 | 4 | 5 |
| 4 | 2 | 2 |
| 5 | 1 | 3 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item | User A rating (Ai) | User B rating (Bi) | Ai | Bi |
| 1 | 3-3=0 | 4-3.4=0.6 | 3 | 4 |
| 2 | 5-3=2 | 3-3.4=-0.4 | 5 | 3 |
| 3 | 4-3=1 | 5-3.4=1.6 | 4 | 5 |
| 4 | 2-3=-1 | 2-3.4=-1.4 | 2 | 2 |
| 5 | 1-3=-2 | 3-3.4=-0.4 | 1 | 3 |

0×0.6) +(2×−0.4) +(1×1.6) +(−1×−1.4) +(−2×−0.4) = 3.0

Denominator= 3.16

Pearson correlation =

**Rating prediction:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Movie Id/user  Id | 53000 | 2539 | 2455 | 215 | 49013 | 1033 | 786 | 833 |
| 533 | 5.0 | 4.0 | 2.0 | NaN | 1.0 | NaN | 3.0 | 3.0 |
| 244 | 3.0 | 1.0 | 3.0 | 1.0 | 2.0 | 3.0 | 3.0 | 3.0 |
| 651 | 3.0 | 2.0 | 4.0 | 2.0 | 2.0 | 3.0 | 3.0 | 3.0 |
| 653 | 5.0 | 5.0 | 2.0 | 1.0 | 2.0 | 5.0 | 1.0 | 1.0 |
| 210 | 4.0 | 4.0 | 5.0 | 2.0 | 2.0 | 2.0 | 3.0 | 5.0 |
| 410 | 1.0 | 5.0 | 2.0 | 4.0 | 4.0 | 1.0 | 1.0 | 4.0 |
| 320 | 5.0 | 5.0 | 5.0 | 3.0 | NaN | 5.0 | 1.0 | 2.0 |
| 292 | 4.0 | 4.0 | NaN | 1.0 | 5.0 | 4.0 | 1.0 | 4.0 |

R244,215 =1

R651,215 =2

R653,215 =1

pred533,215​ =