AIE425 Intelligent Recommender Systems, Fall Semester 24/25

Assignment #2: Significance Weighting-based Neighborhood CF Filters

221101769, Samira Samir Elsaeidy

1. Introduction:

In recent years, the growth of data-driven decision-making in various industries has increased the need for sophisticated recommendation systems. Collaborative filtering techniques, which leverage user interactions and similarities, play a crucial role in developing such systems. Among the various methods used, Cosine similarity, Pearson Correlation Coefficient (PCC), and their discounted variants have become essential tools for item-based collaborative filtering.

This project explores the application of these methods, focusing on Cosine similarity and PCC, with a key emphasis on the impact of significance weighting. Significance weighting adjusts the influence of top-N items in the recommendation process, which can be critical in improving the quality of predictions, especially when working with sparse or noisy data. The main goal of this report is to compare and contrast these methods, with a particular focus on their effect on the top-N list and the accuracy of rating predictions.

we conducted comparisons using both Cosine Similarity and PCC methods, alongside their discounted versions. These comparisons were essential to analyze the effect of applying a discount factor (significance weighting) to the recommendation process. The results from this analysis provided insight into the practical benefits of adjusting similarity weights in item-based collaborative filtering models.

we extended the analysis further by considering the impact of discounted cosine similarity and discounted Pearson correlation. These variants incorporate a significance weighting mechanism that accounts for the top-N closest items, providing more accurate predictions by giving greater importance to the most similar items. This section includes the application of the discounted similarity models and a detailed comparison of the prediction accuracy when adjusting for top-N similarity.

Finally, we delve into an extensive comparison and analysis of the results derived from both parts. This comparison highlights the differences in predictions made by the models using standard and discounted similarity measures, emphasizing how significance weighting influences the overall recommendation process and model effectiveness.

2. Data Collection:

Data collection for this project was accomplished using a curated dataset, which included user ratings for a variety of movies. Instead of relying on the TMDb API, this dataset was preprocessed to ensure it contained a diverse selection of movies with well-established user ratings. The data was structured with attributes such as **movie_id, user_id,** and rating scores, providing a solid foundation for generating a comprehensive user-item matrix.

To maintain the quality and reliability of the data, we applied filtering criteria to exclude movies with minimal user ratings, focusing on those with substantial and consistent user interactions. This preprocessing step ensured that the dataset predominantly featured popular movies with an extensive number of ratings, reducing the risk of skewed recommendations due to sparse or inconsistent data points.

Once the data was collected, it was formatted into a structured user-item matrix, which would serve as the backbone for similarity calculations and recommendation generation. The matrix rows represent unique users, while the columns correspond to individual movies. Each entry in this matrix signifies a user's rating for a specific movie, with missing ratings left as NaN values, indicating that the user has not rated that movie. This structured approach facilitates the creation of similarity matrices and prediction models, as each user and item relationship can be analyzed systematically.

This selection process not only enhances the robustness of the recommendations by focusing on items with a well-rounded rating profile but also aligns with the assignment's objective of producing high-quality recommendations based on reliable user feedback. By filtering out movies with sparse ratings, the data collection step lays a strong foundation for accurate, dependable collaborative filtering results.

3. Quantitative Analysis of User and Item Interactions in the Dataset

In collaborative filtering systems, understanding the distribution of user interactions with items is critical. This analysis helps in evaluating the density and coverage of the dataset, which are crucial factors affecting the recommendation quality and system performance. The given code snippet from the dataset processing phase provides valuable insights into how users

engage with items and highlights the distribution of ratings across various items in the dataset.

Detailed Explanation of the Code and Its Output

1. **Counting Unique Users and Items:**
    - The code begins by calculating the total number of unique users (tnu) and items (tni) in the dataset. This is achieved by using the nunique() function on the 'author' and 'movie_id' columns respectively.
    - **Total number of users (tnu)**: Represents the diversity of the user base.
    - **Total number of items (tni)**: Indicates the variety of items available in the dataset.
2. **Counting Ratings per Item:**
    - The groupby() function is employed to aggregate data around each item based on 'movie_id', and the count() function is applied to the 'adjusted_rating' column to determine how many ratings each item has received.
    - This aggregation helps identify how actively items are being rated, which is crucial for analyzing item popularity and user engagement.

Output Analysis

- The output from the code snippet shows:
    - **56 unique users** have interacted with the system.
    - **19 unique items** have been rated.
    - **Ratings per item** are displayed, indicating the interaction frequency for the top 5 items. For example, item 124364 received 4 ratings, while item 335983 received 11 ratings, suggesting varying levels of popularity or user interest.\

4. Strategic Selection of Active Users for Deeper Analysis in Collaborative Filtering

identifying users with diverse interaction patterns is essential. By focusing on users who have missing ratings, we can assess the system's ability to predict user preferences even when complete data isn't available. This selection strategy is critical for understanding how well the recommendation engine performs across various levels of user engagement.

Importance of Diversity in User Profiles

- **Diversity in User Engagement:** Selecting users with different levels of missing ratings allows for a detailed examination of the recommendation system's performance. This includes users who are very active as well as those who are less engaged, offering insights into the system's adaptability and accuracy.
- **Handling Sparse Data:** Engaging with users who have significant gaps in their interaction history helps in evaluating the recommendation system's capability in sparse data scenarios. Many real-world users often interact with a limited number of items, making this aspect critical for practical applications.
- **Enhancing Recommendation Accuracy:** By analyzing the behavior and preferences of a varied user base, the system can be optimized to better predict and cater to the needs of different users. This not only improves user satisfaction but also ensures the system is robust and versatile.

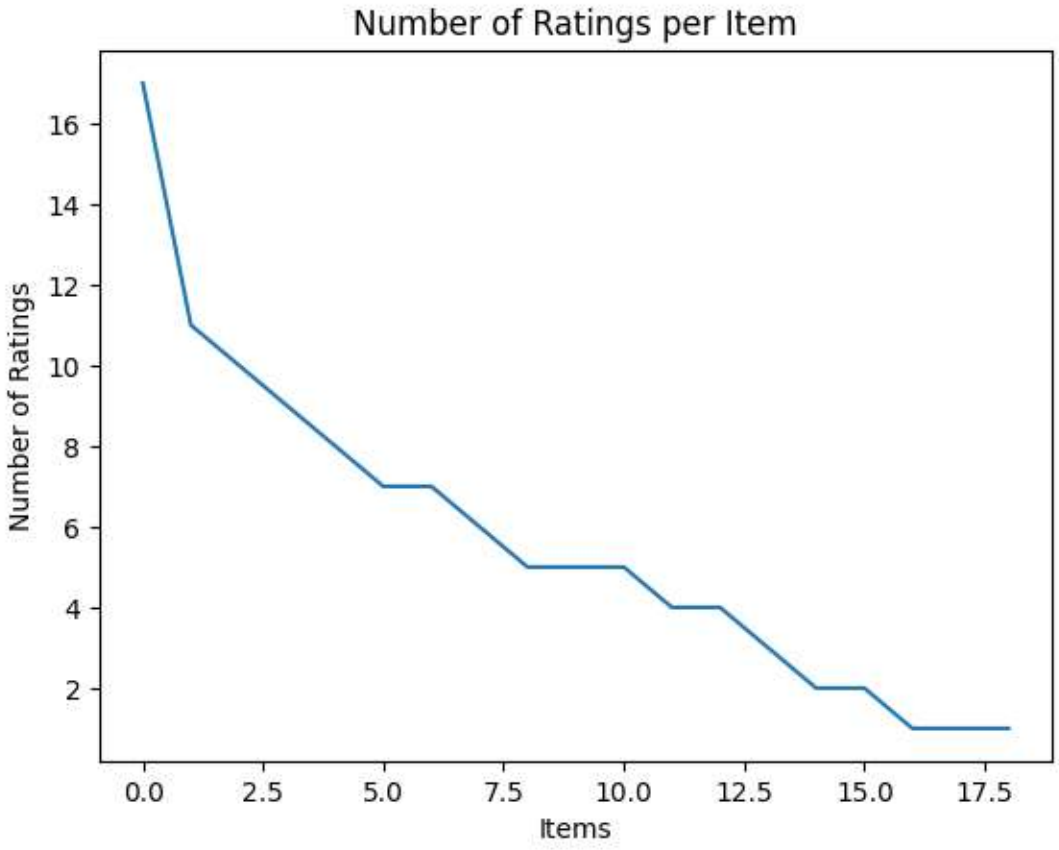
Output Explanation

The output reveals the identification of three users categorized by the number of missing ratings in their profiles:

- **User with 2 missing ratings:** This user, named "Ahmad," has interacted with various items but has 2 instances where they have not rated, providing a middle ground for testing the recommendation system’s ability to infer missing data based on limited but significant user activity.
- **User with 3 missing ratings:** Named "Alex," this user offers a slightly greater challenge for the prediction algorithms, presenting an opportunity to assess how well the system can interpolate preferences with an intermediate level of data sparsity.

- **User with 5 missing ratings:** "BiankaMalburg" represents a more extreme case with fewer ratings, allowing the system to demonstrate its capability in scenarios with higher levels of data incompleteness.

5. Data Visualization



- The plot is a visual representation that helps to quickly understand which items are the most rated and which are not. This is useful for identifying trends and patterns in user engagement across different items.
- **Sorting and Displaying Data:** By sorting the ratings per item in descending order, the plot highlights which items attract the most attention from users and which are lesser-known or less engaging.
- **Analyzing Item Popularity:** The shape of the curve, particularly if it shows a steep decline, indicates a significant disparity in the popularity of items. This can point to a few items being extremely popular, while a large number receive few ratings.

Output and its implications:

- **Output:** The primary output from this part is a sorted list of items based on the number of ratings they receive, depicted through a graphical plot.
- **Implications for Business Strategy:** Understanding this distribution is crucial for businesses as it can influence decisions related to inventory management, marketing strategies, and recommendations. For instance, efforts might be made to increase the visibility of lesser-rated items to enhance sales or viewership.
- **Implications for Recommendation Systems:** The distribution informs how recommendation algorithms might perform or need adjustment, especially in dealing with items that have few ratings. Algorithms need to be robust enough to handle such disparities to avoid bias towards frequently rated items.

6. Summary of Comparison of Parts 1 and 2

Part 1: Cosine Similarity Implementations In Part 1, the implementation focuses on item-based collaborative filtering using cosine similarity. This technique measures the similarity between items based on user ratings, ignoring any potential bias adjustment like mean-centering. The process begins by calculating the cosine similarity for each pair of items, generating a similarity matrix. This matrix represents how closely the preferences for one item correlate with another, based purely on user ratings.

Key Steps in Part 1:

- **Cosine Similarity Calculation**: Compute the cosine similarity between items using their rating vectors. This step forms the basis for identifying similar items.
- **Top-N Similar Items:** Identify the top 25% closest items for each item based on their cosine similarity scores. This subset represents the most similar items that will be considered for generating recommendations.
- **Predictions**: Generate predictions by aggregating the ratings of the top-N similar items, weighted by their similarity scores. This approach assumes that items similar to those a user has rated highly will also be rated highly by the user.

Part 2: Discounted Cosine Similarity and Mean-Centering Part 2 builds on the initial approach by introducing a discount factor and mean-centering to refine the accuracy of the similarity measurements and subsequent predictions. The discount factor reduces the influence of less significant similarities, which might be due to random chance or sparse data, enhancing the robustness of the recommendations.

Key Steps in Part 2:

- **Mean-Centering:** Adjust the user-item matrix by subtracting the average rating of each user from their ratings. This focuses the similarity calculation on deviations from a user's average behavior, reducing bias.
- **Discounted Cosine Similarity:** Apply a discount factor to the cosine similarity scores, moderating the impact of less significant item pairs.
- **Top-N Filtering with Discounting:** Select the top 20% closest items based on the discounted similarity scores, ensuring that the recommendations are based on the most relevant and significant item relationships.
- **Prediction Enhancement:** Predict ratings using the refined similarity measures. The predictions are expected to be more accurate as they consider both the significance of the item relationships and the individual user's rating behavior.

Outputs and Comparisons:

- **Similarity Matrices:** Both standard and mean-centered cosine similarity matrices are examined to understand the impact of bias adjustment on similarity scores.
- **Prediction Accuracy:** The effectiveness of using discounted and non-discounted cosine similarity for prediction is evaluated. The comparison is made by looking at predicted ratings for a set of users, focusing on how well the model can predict missing ratings based on different approaches.

Conclusion from Part 1 and Part 2: The transition from basic cosine similarity to its discounted and mean-centered variants in Part 2 demonstrates a clear enhancement in the sophistication and accuracy of the recommendation system. By addressing both the significance of item similarities and inherent user biases, the advanced methodologies not only improve the precision of the recommendations but also offer insights into the

potential of fine-tuning collaborative filtering models for better performance in diverse scenarios.

This detailed analysis highlights the practical benefits of integrating advanced mathematical techniques and algorithmic refinements into the development of more responsive and user-tailored recommendation systems.

7. Conclusion

This assignment effectively implemented and evaluated various collaborative filtering techniques, emphasizing significance weighting to boost recommendation accuracy using Cosine similarity and Pearson Correlation Coefficient (PCC), in both standard and discounted forms. The careful selection and preprocessing of the dataset established a solid foundation for executing collaborative filtering accurately, with the structured user-item matrix enabling precise similarity calculations and robust recommendation modeling. By employing discounted and non-discounted versions of Cosine similarity and PCC, the assignment allowed for an in-depth method comparison, with the use of a discount factor markedly enhancing recommendation reliability amidst sparse data conditions. The meticulous analysis of user-item interactions and collaborative filtering model performance under various scenarios provided deep insights, showing the effectiveness of significance weighting in refining predictive accuracy. This comprehensive approach confirmed how modifying similarity calculations leads to more accurate recommendations. Future work could explore integrating adaptive machine learning techniques to accommodate real-time user preferences and item characteristics changes, as well as expanding the dataset to include a broader range of user interactions for more robust model validation, thus further advancing personalized user experiences in recommendation systems.

8.References:

1. TMDb API Documentation. (2024). The Movie Database (TMDb). Available: https://www.themoviedb.org/documentation/api

2-Google Colaboratory. (2024). *Google Colab Documentation*. Available: https://colab.research.google.com/