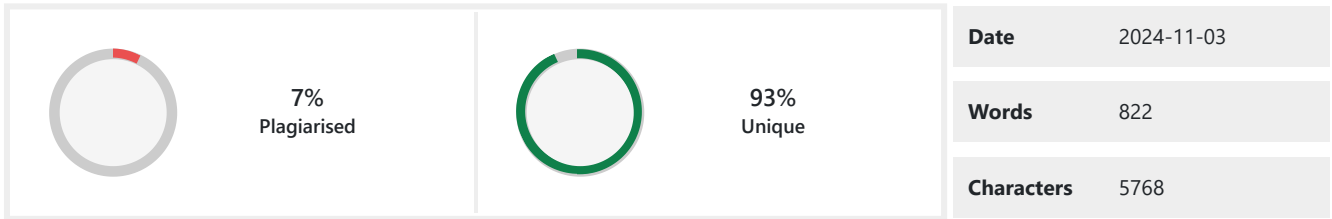


PLAGIARISM SCAN REPORT



Content Checked For Plagiarism

AIE425 Intelligent Recommender Systems, Fall Semester 24/25
Assignment #1: Neighborhood CF Models (User, Item-Based CF)
Student ID: A20000021, Full Name: Ahmed Ashraf Mohamed Ali

1. Core Idea

The goal of this Assignment is to explore the understanding of the recommender systems while focusing on Collaborative Filtering models. These models are used to make recommendations based to user-based and item-based. Through this assignment we will explore many stages as: data collection, data preparation and calculating similarities. With applying these techniques, we can make recommendations according to user's preferences.

2. Data Source Selection

We have many companies in different categories that use recommender systems such as:

1. Netflix
2. Spotify
3. Amazon
4. LinkedIn

I will choose Amazon as the data source for this assignment.

Amazon uses Explicit Ratings type which is 5-star scale rating type, with 1-star being the worst rating. Both feedback and reviews are displayed on Amazon as average ratings. It collects customer feedbacks by calculating product's star rating using machine learning models instead of the simple way of taking the average as it has a criteria or factors that it relies on such as:

1. Rating (review) date
2. Customer status (verified or not)
3. Purchase status (verified or not).

3. Data Collection & preprocessing

3.1. Data Collection

* Method: Web scraping using beautiful soup python library.

* How: By adding many URLs of Amazon products in text file and passing it to web beautiful soup code and looping on URLs and extract the information I need.

3.2. Column Naming

* Assigned column names:

* Product, Rating, Ratings Count, Availability.

3.3. Data Cleaning

- * Handling Missing Values:
- * Removed any rows containing any NaN values.
- * Whitespace removal:
- * Removed whitespace around text to make columns format the same in each column.
- * Ratings Count Conversion:
- * Removed words (ratings) from the Ratings Count column and convert it from a string to integer. (Such as: 63 ratings to 63)
- * Rating Conversion:
- * Take the numerical values from rating column (Such as: 4.1 instead of "4.1 out of 5 stars") and convert it to float number.

4. Dataset Description

4.1. Columns Description

- * Product:
- * Description: Contains the name, model, and key features of the product (Title) of each product.
- * Example: Amazon Essentials Men's Digital Chronograph Black Strap")
- * Purpose: Useful for recommending similar products based on content-based filtering.
- * Rating:
- * Description: Average user rating of a product.
- * Example: 4.1
- * Purpose: Identify highly rated products and calculate similarity scores for Collaborative Filtering.
- * Rating Count:
- * Description: The number of users who rated the product.
- * Example: 1000
- * Purpose: Identify highly bought products.
- * Availability:
- * Description: The stock status of each product.
- * Example: In Stock", "Only 3 left in stock - order soon
- * Purpose: Avoid recommendations of out-of-stock items.

4.2. Data Consistency:

- * All products have non-null entries across all fields, ensuring no gaps in critical information.

4.3. Data Processing:

- * Ratings are numerically formatted, and counts are integers.

5. Overview about user-based and item-based CF algorithms

5.1. User-Based Collaborative Filtering (CF):

It's a Collaborative Filtering (CF) Algorithm that relies on user preferences. For example, if we have two users, (User A and User B) User A has watched 4 movies and liked all of them, While User B has liked only three out of the four that User A liked. Therefore, it predicts that User B will also like the fourth movie.

- * PROS:
- * Easy to implement and Context independent.
- * CONS:

* Sparsity: The percentage of people who rate items is really low.

- * Analytical Solution:

1- Similarity Calculation: Compute similarity scores between users using metrics like:

- * Pearson Correlation: Measures the linear correlation between two users' ratings.

* Cosine Similarity: Measures the cosine of the angle between two users' rating vectors.

2- Prediction: To predict the rating that user would give to item.

3- Recommendation Generation: After predicting ratings, recommend items with the highest predicted ratings that the user has not yet rated.

5.2. Item-Based Collaborative Filtering (CF):

It's a Collaborative Filtering (CF) Algorithm as the User-Based example but instead of relying on other users we rely on items. For example, if we have two items, Item X and Item Y, and many users have rated both, we can see that users who liked Item X also tended to like Item Y. If a user has liked Item X, the algorithm predicts that they will also enjoy Item Y based on the preferences of similar users.

* PROS:

* Stability and Scalability

* CONS:

* Cold Start: New items with no ratings struggle to receive accurate recommendations.

* Analytical Solution:

1- Similarity Calculation: Compute similarity scores between items using metrics like:

2- Adjusted Cosine Similarity:

* Prediction: To predict the rating that user would give to item:

* Recommendation Generation: After predicting ratings, recommend items with the highest predicted ratings that the user has not yet rated.

Matched Source

Similarity 13%

Title: 5 Stars: Behind the Scenes of Amazon's Rating System

Jun 26, 2023 — Both feedback and reviews are displayed on Amazon as average ratings. Buyers must provide a star rating and written text when leaving seller ...

<https://www.ecomengine.com/blog/amazon-rating-system>

Similarity 5%

Title: User-Based and Item-Based Collaborative Filtering - Medium

Sparsity: The percentage of people who rate items is really low ✓. Scalability: The more K neighbors we consider (under a certain threshold), the better my classification should be.

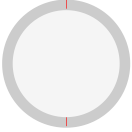

<https://medium.com/@cfpinela/recommender-systems-user-based-and-item-based-collaborative-filtering-5d5f375a127f>

Similarity 3%

Title: data science week 14 낱말 카드

<https://quizlet.com/kr/924289673/data-science-week-14-flash-cards>

PLAGIARISM SCAN REPORT

	0% Plagiarised		100% Unique	Date	2024-11-03
				Words	791
				Characters	5323

Content Checked For Plagiarism

6. Assignment Results

6.1. Average Rating:

- * The average rating for the dataset is 4.2124999999999995.
- * It is obtained after bringing all rating values into a number form and computing their average.
- * It is a representation of satisfaction level over all the products that have been reviewed within our proposed user-item matrix.

6.2 Similarity Measure techniques:

- * Cosine Similarity: It tells how similar two vectors' direction is by calculating cosine of the angle between the vectors.
- * Pearson Correlation Coefficient: This only supplies the measure of linear relationship by determining to what extent ratings of one user or item vary in the same way from the mean.

6.3. Peer Group Comparison

- * User-Based CF:
1- Cosine Similarity: In their approach, peer groups were described by looking for users who had similar ratings to one another in the direction. This is much efficient in identifying that user with related preference exist.

It shows high similarity scores among users, indicating strong preferences for similar products:

- * High Similarities:
* The similarity between Users 1 and 2 are 0.87 out of 1, and while Users 3 and 1 the score is relatively high which is 0.92.
* User 5 is very close to Users 2 and 3, with the distance coefficients of 0.96 and 0.88, respectively

- * Low Similarities:

- * User 4 is most similar with User 3 but with low similarity which is (0.73).
2- Pearson Correlation: This method was much finer and used the varying scales adopted by the users.

The correlation through Pearson shows varied results:

- * High Positive Correlation:
* User 2 directly related to user 5 with correlation score of 0.64 which shows that they both have some similarities in their preferences.
- * Negative Correlations:
* User 4 has the following negative scores where the preferences between the 2 users differ, User 1- (-0.13) and User 2 (-0.48).

Item-based CF:

- 1- Cosine Similarity: This involved clustering where one tries to assemble users who have given most similar observations regardless of the actual value of the observation.

It indicates how closely related the products are based on user ratings:

- * High Similarities:

- * Xiaomi Redmi Note 12 and Samsung Galaxy S24 are very alike (0.95).

- * The correlation coefficients which are above 0.9 suggest that sample smartphones show high probability that users likely rate similar products.

- * Low Similarities:

- * The lowest coefficient here implies different consumers for the Shark Pet Cordless Stick Vacuum the Xiaomi Redmi A3 products with the results at 0.63. similarly in the direction.

2- Pearson Correlation: This is an efficient way to find users with shared preferences.

- * High Positive Correlations:

- * Samsung Galaxy S24 and Xiaomi Redmi Note 12 show strong correlations (0.81), indicating shared user interest.

- * Negative Correlations:

- * The item correlations reflect varied user preferences, with some items having low or negative correlations, suggesting they are less likely to be rated by the same users

6.4 Pros and Cons of Each Technique

- * Cosine Similarity:

- * Pros: Cheap to compute and easy; most useful when directionality of user ratings is more important than overall value.

- * Cons: does not take the rating scale for every individual and therefore it creates less number of specific peer groups.

- * Pearson Correlation Coefficient:

- * Pros: It adjusts for personal rating confounds and fluctuations; therefore, it is more precise for individual-recommendation systems.

- * Cons: It might produce little relevant results when there is low data density because it is influenced by the scales due to a few ratings on the correlation.

6.5. Rating Prediction

- * User based Collaborative Filtering (CF): Predict the user unknown rating for a product according to ratings given by similar users.

- * It shows more high ratings for premium smartphones than Chinese smartphones, as Samsung Galaxy S24 on the top of the list for most users.

- * It displays that, users have different preferences, because of their unique different ratings.

- * Item-based CF: Predict a user's rating for a product based on that user's ratings of similar items.

- * The vacuum products are on top of recommendations, indicating their popularity among users.

- * Recommendations vary significantly from those based on cosine similarity.

6.6. Top-N Recommendation

- * Product ranking for every user.

- * Recommend top-N items according to the highest predicted ratings.

- * Top Recommendations: User 4 favors Xiaomi Redmi Note 12, while others lean towards Samsung Galaxy S24.

- * The recommendations become more varied; the top rank for User 1 is the LUENX Aviator Sunglasses, and for User 2, it will be the Amazon Essentials Watch.

* The top recommendations come the vacuum products, therefore it's the most popular among users.

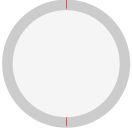

* The outstanding products according to the analysis on a correlation are SAMSUNG Galaxy S24 and LUENX Aviator Sunglasses.

Matched Source

No plagiarism found

Check By:  Dupli Checker

PLAGIARISM SCAN REPORT

	0% Plagiarised		100% Unique	Date	2024-11-03
				Words	992
				Characters	7122

Content Checked For Plagiarism

7. Implementation Process

* Introduction: I successfully created an intelligent recommender system using collaborative filtering techniques which are user based and item based with cosine similarity and Pearson correlation. I collected Amazon products dataset using web scraping method and finally, predicted user ratings and top recommendations for users.

7.1. Libraries

1. Beautiful Soup: Parse HTML document
2. Requests: Handle HTTP requests
3. Pandas: Data Cleaning
4. NumPy: Mathematical Functions
5. Matplotlib and Seaborn: Data Visualization
6. SciPy: Pearson correlation
7. Scikit learn: Cosine similarity

7.2. Web Scrapping Implementation

* I extracted products information from Amazon by passing a text file full of URLs of amazon products and let the code loop on it to extract the information I want, I created web scraper that collects (product tile, rating, ratings count and availability status).

* Code Reference: The web scraping code is based on a tutorial from an open-source website which is GeeksforGeeks: "Scraping Amazon Product Information using Beautiful Soup"

7.3. Intelligent Recommender Systems Implementation

* Data preparation

1. Assign columns: Product, Rating, Ratings count, and Availability.
 2. Data cleaning: Removed all rows with NaN values.
 3. Transformation: Change Ratings count data type into integers.
- Finally, saved the cleaned data into amazon_data_cleaned.csv for further operations.

7.4. Exploratory Data Analysis

- * Ratings Distribution: Visualized by using histogram and box plots.
- * Statistical summary: Statistical Insights on the dataset.

7.5. User item matrix

- * Generated user item matrix with 5 user IDs and product IDs from the cleaned dataset.
- * Random rating (from 1 to 5) assigned to users for each product and saved .

7.6. Similarity Calculation

- * User based with cosine and Pearson similarity: Measures similarity between users based on item ratings.
- * Item based with cosine and Pearson similarity: Measures similarity between items based on user ratings.

7.7 Prediction

Predict ratings using collaborative filtering techniques

- * User Based: Predicted ratings by weighting each user's similarity to others.
- * Item Based: Predicted ratings by weighting each item's similarity to others.
- * Top Recommendations: Generated lists of the top-N recommended items for each user

7.8. Results

* User-Based Collaborative Filtering (Cosine and Pearson): In this section, the list of users and their predicted ratings for airplanes appears, as well as top choices are given. The evaluation metrics provided above showed that the RMSE and the MAE were lowest in the case of the User-Based CF using Pearson correlation = 1.05, 0.84, respectively which emphasize its capability to capture the user preference correctly.

* Item-Based Collaborative Filtering (Cosine and Pearson): This section gives an output of the predicted ratings and the most recommended items. Performance metrics indicated that the proposed Item-Based CF using Pearson correlation had the highest RMSE of 1.74 and MAE of 1.46, proving that it is difficult to handle item similarities in sparse datasets than in user-based methods.

8 User Based vs. Item Based: Cosine Similarity & Pearson Correlation

User Based Collaborative Filtering (CF) emphasizes their users' similarities and provide recommendation to the users having this similarity. The cosine similarity measure in user-based CF, computes the angle between two vectors representing user preferences and focus on similarity. $\cos \theta$ Races are good when there is enough information on the user, but it can fail in situations where there is little data or user preferences change with time.

On the other hand, Item Based CF assumes similarity of items through end-user ratings on those items to ascertain similarities of items. The Pearson correlation coefficient is applied in measuring the different intensities of relations between item ratings that aims at establishing the linear relationship between the items. This makes this method more stable in general because it is less frequently that the characteristics of items are going to change compared to the preferences of a particular user.

Overall, user-based CF is effective at approaching user-oriented recommendations, while item-based CF focuses on the relationships between items, which seem to be generally smoother and more stable recommendations, especially if the base is a large number of different but not very popular items.

9. Conclusion

The evaluation of collaborative filtering strategies reveals distinct impacts on predicted accuracy:

1. User-Based Collaborative Filtering (Cosine Similarity):

This approach had an of RMSE = 1.11 and MAE = 0.93. However, depending on cosine similarity the algorithm could sometimes miss the minor difference in the users rating pattern and has moderate prediction accuracy.

2. User-Based Collaborative Filtering (Pearson Correlation):

This method turned out to be better than the cosine similarity version giving RMSE of 1.05 and MAE of 0.84. With a sensitivity of the relationship between users' ratings, Pearson correlation provided a finer grain prediction, that captured the direction and strength of the user's preference co-linearity.

3. Item-Based Collaborative Filtering (Cosine Similarity):

This strategy has provided **RMSE of 1.17 and MAE of 0.99** implying that item similarity would enhance extra variation of prediction. While regarding the relations between items, it also takes into consideration the user id but at times does not focus on personal preference thus giving a slightly lesser accurate prediction.

4. Item-Based Collaborative Filtering (Pearson Correlation):

It had the highest error metrics of all the approaches tried with An RMSE of 1.74 and MAE of 1.46. Despite this; it probably underperformed as a result of low density of user-item interactions which might have resulted to low reliability of the prediction model.

Collaborative filtering that used Pearson's coefficient provided the lowest values of prediction errors when using the three strategies. It better captured user preferences and witnessed a lower error rate than the other strategies suggesting that selection of an appropriate similarity measure would improve the recommendation exercise

10. Probable Enhancements

To enhance the recommendation system further, we may consider the following:

1. Hybrid Approaches: Combine collaborative filtering with content-based filtering to leverage both user-item interactions and item attributes. This can help address cold-start problems and improve recommendations for new users or items.

2. Advanced Similarity Metrics: Explore more sophisticated similarity metrics, such as adjusted cosine similarity or Jaccard similarity.

Matched Source

No plagiarism found

Check By:  Dupli Checker