Book Recommender System

Programming Machine Learning Final Project

Na Young Choi

Final Project Presentation

March 19, 2023

Contents

1. Executive Summary

The book recommender system has the potential to enhance the user experience of library services and encourage reading habits.

The book recommender system project aimed to build an efficient and accurate book recommendation system using machine learning techniques. The project used the Goodreads dataset, which included data on book title, book ID, user ID, genres, and user ratings, among others. The project focused on the fantasy and paranormal genres and used a sample of 10,000 users to test the recommender system.

The exploratory data analysis revealed that users generally rated books positively, and the majority of ratings were distributed between 3.0 and 5.0 the collaborative filtering approach was used in the recommender system, and two methods, item-based filtering and user-based filtering, were employed. The item-based filtering approach recommended similar books based on the similarity of past borrowing history data, while user-based filtering recommended items based on the preference data of other users who had similar tendencies and preferences based on the books they borrowed in the past.

Both item-based filtering and user-based filtering approaches were evaluated using similarity, RMSE, and SVD measures. These results showed that the system could accurately recommend books based on the selection of one book or one user. The recommender system aimed to provide users with personalized book recommendations that matched their interests and preferences.

In conclusion, the book recommender system project built a book recommendation system that could help users find books that matched their interests and preferences.

2. Introduction

Recently the growth of digital libraries and online bookstores has made it easier for individuals to access a vast array of books. Although there are many options available, it can be challenging to find books that match an individual's interests and preferences. Most libraries and online bookstores are concerned with providing each customer or user with an engaging and individualized experience whether it is shopping for clothing on website, trying to find a move on a streaming services or OTT platforms which refers to the delivery of audio or video content over the internet, or scrolling through social media. To make it more convenient for users to use the library and easily find books that match their preferences, a recommender system is going to be needed. Recommender systems leverage data on an individual's past preferences to make personalized recommendations on what books they might enjoy reading.

In this project, it aims to build a book recommender system using machine learning techniques.

This system will analyze a user's reading history, book ratings, and other relevant data to recommend new books that align with their interests. To achieve this, it will first collect and preprocess data on book title, book id, user id, genres, user ratings, etc. According to various machine learning algorithms, it can build the recommender system. The main goal of this project is to provide an efficient and accurate book recommendation system to help users find books that match their interests and preferences.

3. Non-Technical Parts

3.1 Data Description

This project aims to build a book recommender system that accurately captures users' preferences using the Goodreads dataset. The Goodreads dataset is available on the UC Irvine Machine Learning Repository website and includes three files, namely the Book file, Interaction file, and Review file.

Source: https://sites.google.com/eng.ucsd.edu/ucsdbookgraph/home The

dataset includes three separate files:

1. Book file: A file for book ID mapping
2. Interaction file: A file all interactions for each book and user
3. Review file: A file for user ID mapping

The dataset contains vast amounts of data for all genres, making it difficult to handle within the project's scope. Therefore, this project will focus on the fantasy and paranormal genres, resulting in a manageable file size for developing the recommender system.

The selected genre contains 55,397,550 interactions between books, users, and review, 258,585 books data in the book file, and 3,424,641 reviews data in the review file. However, these data are still too large for testing within a reasonable amount of time. Then, the testing will be conducted using a sample of 10,000 users.

The data files contain several data types, including user_id, book_id, review_id, rating, num_votes, num_comments, title, average_rating, rating_count, text_review_count, and is_read. These data types provide essential information for building a recommender system that can predict a user's preferences based on their past behavior, such as books they have read and their ratings.

The dataset used in this project does not contain any missing values that require replacing with other values. To simplify the dataset, it differentiated between categorical and numerical variables. User_id, book_id, and review_id were converted into categorical variables to improve the readability of the dataset and prevent any issue when running the recommender system.

4. Technical Parts

4.1 Exploratory Data Analysis

The ratings in the dataset range from 5.0 to 0.0, divided into six categories. The majority of ratings fall in the 5.0 category, including in 46119 ratings and indicating that users generally rate books positively. Additionally, most of the ratings are distributed between 3.0 and 5.0, with fewer ratings as the lower end of the spectrum. To gain further insight into the distribution of ratings, a scatter graph was created (figure 1.), which revealed that the majority of ratings are clustered between 3.5 and 4.5. This distribution can be predicted as a left-skewed distribution, indicating that the dataset has a higher frequency of high ratings and a lower frequency of ratings.

This information is crucial for building an effective recommender system as it provides insights into how users rate books and what types of ratings are most common. By understanding the distribution of ratings, it can design a more accurate recommendation algorithm that takes into account the skewness of the data. Analyzing the distribution of ratings is an essential step in building an effective book recommender system.

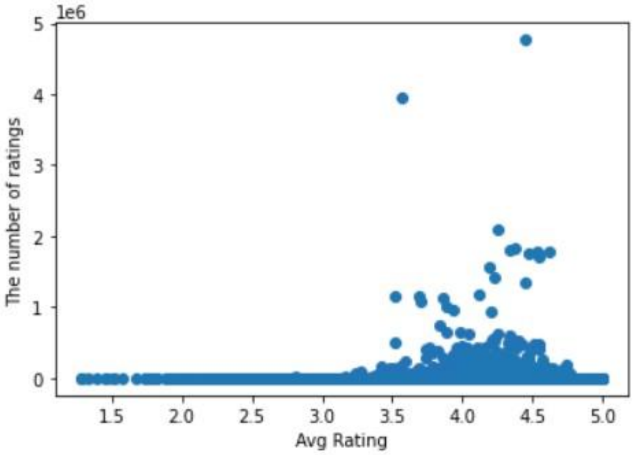


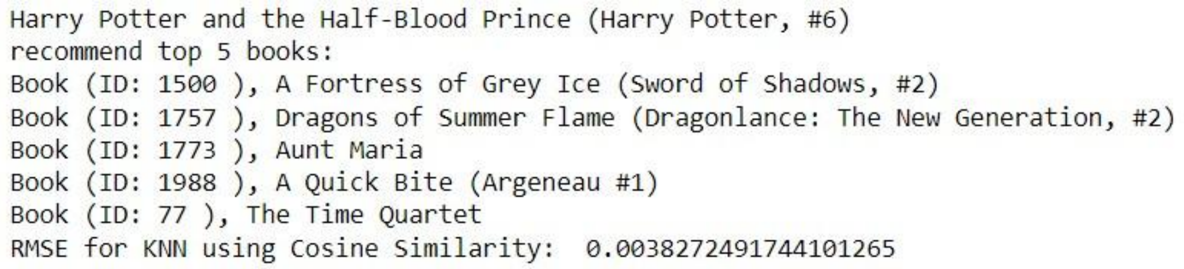
figure 1. Scatter graph

In this book recommender system, collaborative filtering is used to predict items that users are likely to prefer based on their past borrowing history. Two approaches, item-based filtering and user-based filtering were employed. Item-based filtering recommends similar books based on the similarity of past borrowing history data. User-based filtering recommends items based on the preference data of other users who have similar tendencies and preferences based on the books they borrowed in the past. By leveraging the power of collaborative filtering, this recommender system aims to provide users with personalized book recommendations that match their interests and preferences.

4.2 Item-Based Recommender

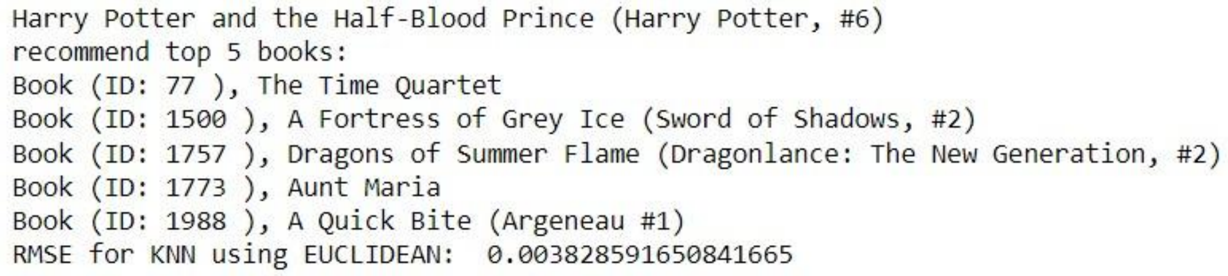
Item-based filtering involves assigning keywords to products, analyzing the user's preferences and ratings, searching those keywords in the databased, and recommending other items that share similar attributes. This approach is based on the idea that items with similar attributes are more likely to be preferred by user. By using item-based filtering, the system can recommend products that the user is more likely to be interested in based on their past interactions with similar products.

The book recommendation system was developed to recommend five books based on the selection of one book, and the system was evaluated using similarity and RMSE measures. RMSE was chosen as the accuracy measurement method since it was difficult to predict ratings for the books. Cosine similarity, Euclidean distance, and Pearson similarity were used to measure the similarity between books, and all three methods produced similar scores. The RMSE values for Cosine similarity, Euclidean distance, and Pearson similarity were 0.0038272, 0.00382859, and 0.00382075, respectively, as you can see the figure 2, 3, & 4. Although Pearson similarity had the lowest RMSE value, when comparing the recommended books, it produced vastly different recommendations compared to the other methods. Therefore, Pearson similarity was excluded from the evaluation, and only Cosine similarity and Euclidean distance were used. Among these two, Cosine similarity produced a slightly better score, so it was considered as the optimal method for this book recommendation system.



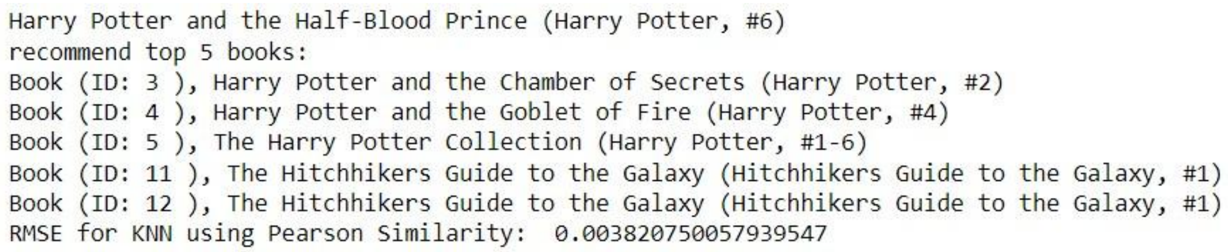
```
Harry Potter and the Half-Blood Prince (Harry Potter, #6)
recommend top 5 books:
Book (ID: 1500 ), A Fortress of Grey Ice (Sword of Shadows, #2)
Book (ID: 1757 ), Dragons of Summer Flame (Dragonlance: The New Generation, #2)
Book (ID: 1773 ), Aunt Maria
Book (ID: 1988 ), A Quick Bite (Argeneau #1)
Book (ID: 77 ), The Time Quartet
RMSE for KNN using Cosine Similarity:  0.0038272491744101265
```

figure 2. RMSE (Cosine Similarity)



```
Harry Potter and the Half-Blood Prince (Harry Potter, #6)
recommend top 5 books:
Book (ID: 77 ), The Time Quartet
Book (ID: 1500 ), A Fortress of Grey Ice (Sword of Shadows, #2)
Book (ID: 1757 ), Dragons of Summer Flame (Dragonlance: The New Generation, #2)
Book (ID: 1773 ), Aunt Maria
Book (ID: 1988 ), A Quick Bite (Argeneau #1)
RMSE for KNN using EUCLIDEAN:  0.003828591650841665
```

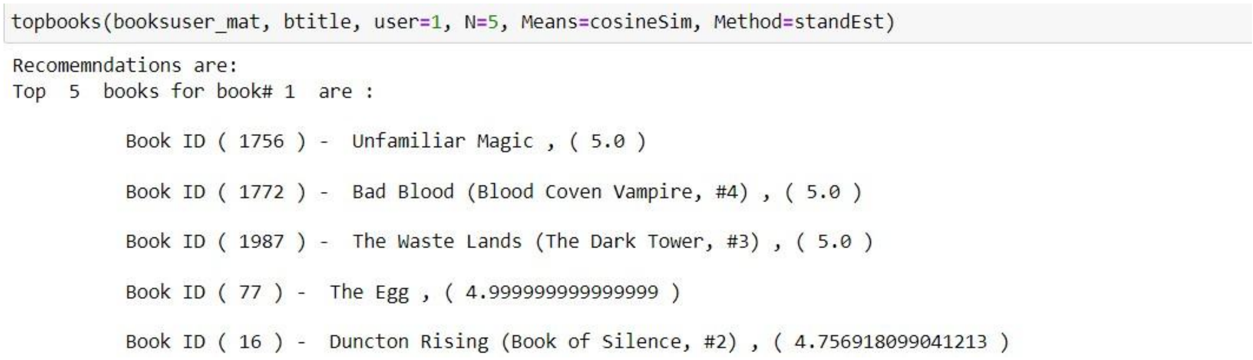
figure 3. RMSE (Euclidean Distance)



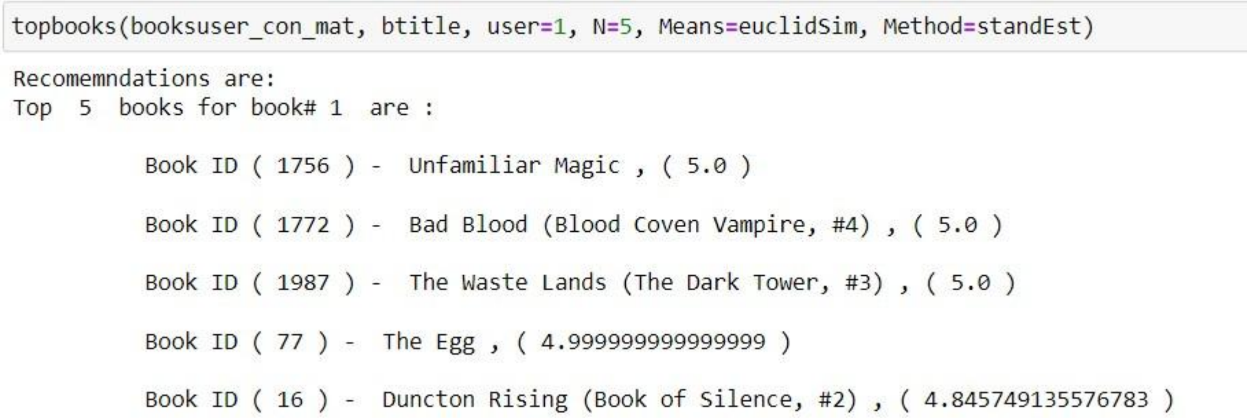
```
Harry Potter and the Half-Blood Prince (Harry Potter, #6)
recommend top 5 books:
Book (ID: 3 ), Harry Potter and the Chamber of Secrets (Harry Potter, #2)
Book (ID: 4 ), Harry Potter and the Goblet of Fire (Harry Potter, #4)
Book (ID: 5 ), The Harry Potter Collection (Harry Potter, #1-6)
Book (ID: 11 ), The Hitchhikers Guide to the Galaxy (Hitchhikers Guide to the Galaxy, #1)
Book (ID: 12 ), The Hitchhikers Guide to the Galaxy (Hitchhikers Guide to the Galaxy, #1)
RMSE for KNN using Pearson Similarity:  0.003820750057939547
```

figure 4. RMSE (Pearson Similarity)

Another evaluation method used for the book recommendation system was Singular Value Decomposition (SVD), a powerful matrix factorization technique. The SVD model can handle missing data and noisy input, which are common issues in recommendation system. There is no missing values in the dataset, SVD was chosen due to its ability to capture latent factors and provide accurate recommendations. In this evaluation, Cosine similarity produced a score of 4.3313, while Pearson similarity produced a score of 4.2415.



```
topbooks(booksuser_mat, btitle, user=1, N=5, Means=cosineSim, Method=standEst)

Recomemndations are:
Top  5  books for book# 1  are :

        Book ID ( 1756 ) -  Unfamiliar Magic , ( 5.0 )

        Book ID ( 1772 ) -  Bad Blood (Blood Coven Vampire, #4) , ( 5.0 )

        Book ID ( 1987 ) -  The Waste Lands (The Dark Tower, #3) , ( 5.0 )

        Book ID ( 77 ) -  The Egg , ( 4.999999999999999 )

        Book ID ( 16 ) -  Duncton Rising (Book of Silence, #2) , ( 4.756918099041213 )
```

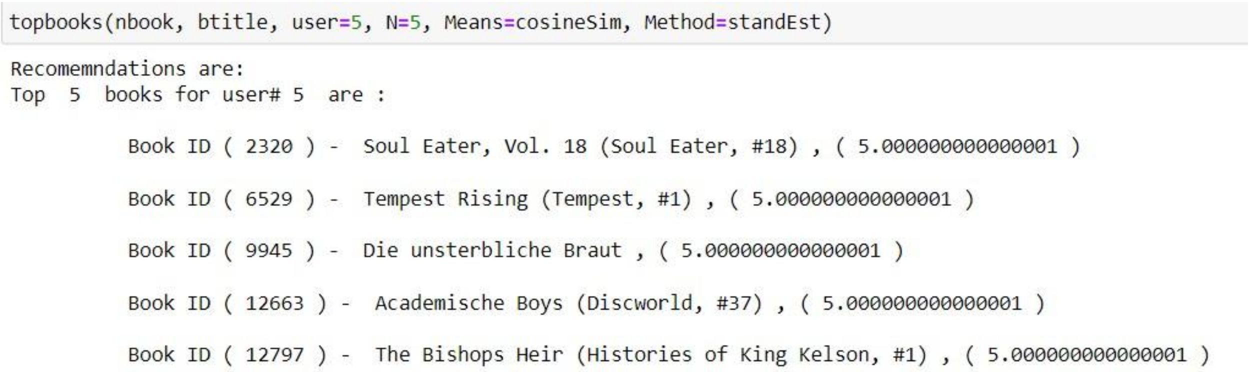


```
topbooks(booksuser_con_mat, btitle, user=1, N=5, Means=euclidSim, Method=standEst)

Recomemndations are:
Top  5  books for book# 1  are :

        Book ID ( 1756 ) -  Unfamiliar Magic , ( 5.0 )

        Book ID ( 1772 ) -  Bad Blood (Blood Coven Vampire, #4) , ( 5.0 )

        Book ID ( 1987 ) -  The Waste Lands (The Dark Tower, #3) , ( 5.0 )

        Book ID ( 77 ) -  The Egg , ( 4.999999999999999 )

        Book ID ( 16 ) -  Duncton Rising (Book of Silence, #2) , ( 4.845749135576783 )
```

4.3 User-Based Recommender

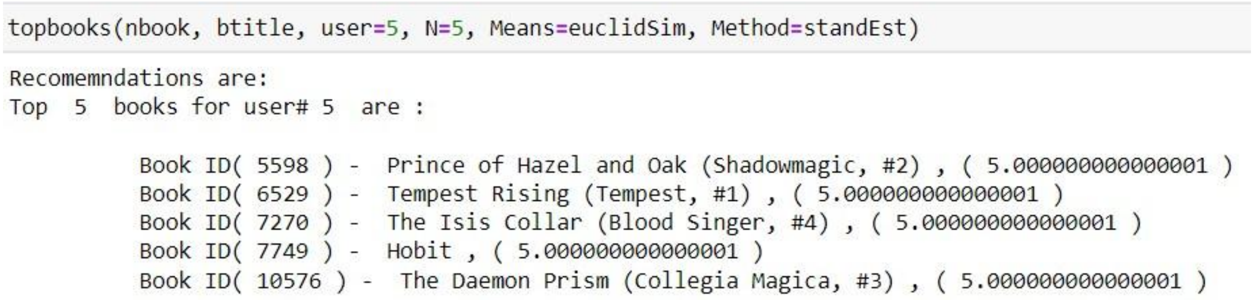
In user-based filtering, the system analyzed the preferences and interactions of similar users to recommend items that the current user may be interested in. The system first identifies users who have similar preferences to the current user based on their past interactions with books. It then recommends books that were highly rated by these similar users but have not yet been interacted with by the current user. By leveraging the preferences of similar users, user-based filtering can recommend books that are more likely to match the current user's interests and preferences.

Similar to item-based filtering, it used a method that recommends five books. However, instead of inputting one book, it inputted one user and recommended five books based on that user's preferences. Based on the analysis of five recommended books using Cosine similarity and Euclidean distance, only one book, "The Tempest Rising", is the same recommendation between the two methods (Figure 5 & Figure 6). The last analysis using Pearson similarity had totally different book recommendations (Figure 7). In contrast to item-based filtering, user-based filtering using Cosine similarity and Euclidean distance provided two different recommendations. Additionally, when considering the Root Mean Square Error (RMSE), the user-based filtering using Cosine similarity has an RMSE of 0.16455 (figure 8) and the Euclidean distance has an RMSE of 0.164574 (figure 9). That means that cosine similarity has better results.



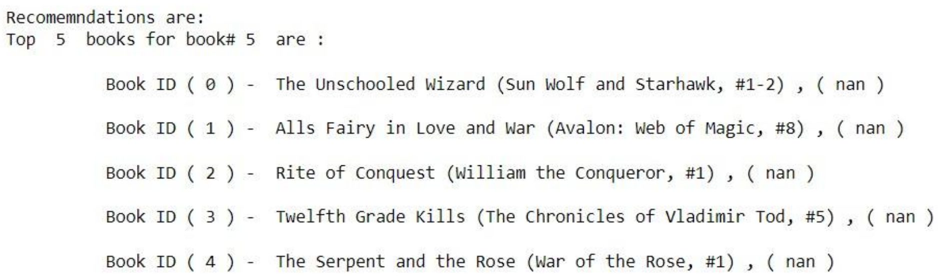
```
topbooks(nbook, btitle, user=5, N=5, Means=cosineSim, Method=standEst)

Recomemndations are:
Top  5  books for user# 5  are :

          Book ID ( 2320 ) -  Soul Eater, Vol. 18 (Soul Eater, #18) , ( 5.000000000000001 )

          Book ID ( 6529 ) -  Tempest Rising (Tempest, #1) , ( 5.000000000000001 )

          Book ID ( 9945 ) -  Die unsterbliche Braut , ( 5.000000000000001 )

          Book ID ( 12663 ) -  Academische Boys (Discworld, #37) , ( 5.000000000000001 )

          Book ID ( 12797 ) -  The Bishops Heir (Histories of King Kelson, #1) , ( 5.000000000000001 )
```

Figure 5. Five Recommendations using Cosine Similarity



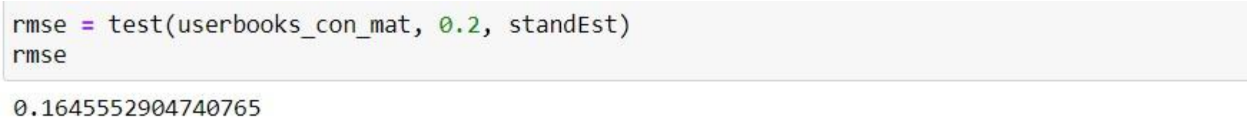
```
topbooks(nbook, btitle, user=5, N=5, Means=euclidSim, Method=standEst)

Recomemndations are:
Top  5  books for user# 5  are :

          Book ID( 5598 ) -  Prince of Hazel and Oak (Shadowmagic, #2) , ( 5.000000000000001 )
          Book ID( 6529 ) -  Tempest Rising (Tempest, #1) , ( 5.000000000000001 )
          Book ID( 7270 ) -  The Isis Collar (Blood Singer, #4) , ( 5.000000000000001 )
          Book ID( 7749 ) -  Hobit , ( 5.000000000000001 )
          Book ID( 10576 ) -  The Daemon Prism (Collegia Magica, #3) , ( 5.000000000000001 )
```

Figure 6. Five Recommendations using Euclidean Distance



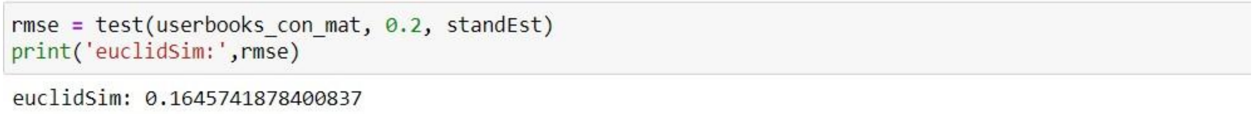
```
Recomemndations are:
Top  5  books for book# 5  are :

        Book ID ( 0 ) -  The Unschooled Wizard (Sun Wolf and Starhawk, #1-2) , ( nan )

        Book ID ( 1 ) -  Alls Fairy in Love and War (Avalon: Web of Magic, #8) , ( nan )

        Book ID ( 2 ) -  Rite of Conquest (William the Conqueror, #1) , ( nan )

        Book ID ( 3 ) -  Twelfth Grade Kills (The Chronicles of Vladimir Tod, #5) , ( nan )

        Book ID ( 4 ) -  The Serpent and the Rose (War of the Rose, #1) , ( nan )
```

Figure 7. Five Recommendations using Pearson Similarity



```
rmse = test(userbooks_con_mat, 0.2, standEst)
rmse

0.1645552904740765
```

Figure 8. RMSE (Cosine Similarity)



```
rmse = test(userbooks_con_mat, 0.2, standEst)
print('euclidSim:',rmse)

euclidSim: 0.1645741878400837
```

Figure 9. RMSE (Euclidean Distance)

Another evaluation method used for the book recommendation system was Singular Value Decomposition (SVD) that is used in the user-based filtering. The main idea behind SVD is to decompose a large user-item ratings matrix into three smaller matrices: a user-feature matrix, a feature-feature matrix, and an item-feature matrix. By doing so, SVD can capture latent features and relationships between users and items, even when there are missing values in the ratings matrix. It tried to obtain SVD values using Cosine similarity and Euclidean distance for a long time, but was ultimately unable to get an answer.

4.4 Discussion

This book recommender system project involved implementing two filtering approaches, itembased and user-based filtering, and evaluating their accuracy using RMSE and SVD. Typically, item-based collaborative filtering is considered more accurate than user-based collaborative filtering. However, this project's results showed that user-based filtering resulted in a lower RMSE. Both filtering methods produced lower RMSE scores using cosine similarity, with userbased filtering outperforming item-based filtering in order to the user-based filtering method get lower score than the item-based filtering method. However, interestingly, item-based filtering produced nearly identical recommendations when recommending five books using Cosine similarity and Euclidean distance, while user-based filtering recommended only one same book and different rest of books.

Another evaluation method that was used was SVD. However, there were several issues encountered during the derivation of this method. Firstly, it took too much time to obtain results, making it very difficult to get the desired output. The dataset used for this method was about 2GB each, and it took almost 2 hours to obtain a single result. Secondly, there were computer-related problems, resulting in the loss of all previously derived results. It seemed that this program was very demanding to execute due to the computer's performance, resulting in many errors occurring during the execution. As a result, it was impossible to obtain all SVD values, and only item-based filtering results were obtained. Despite running user-based filtering for an entire day, the answer could not be obtained. However, the SVD results obtained from item-based filtering were quite good, which scores of 4.33 and 4.24.

5. Conclusion

In conclusion, this project has developed a book recommender system using two different filtering methods: item-based and user-based filtering. Item-based filtering is a system that recommends similar items based on past item preference data. It calculates the similarity between items to generate a recommendation list. This system is advantageous when updating new books since if does not require knowledge of which users preferred the new items. User-based filtering, on the other hand, recommends books based on data from users with similar tastes and interests in the past. Since user-based filtering uses data from multiple users, it seems to be better a providing new recommendations. Both filtering methods provided good recommendation quality, but they come with different advantages and disadvantages. Previous mentioned that item-based filtering is suitable for updating new books since it does not require user preference data for the new items, while user-based filtering is better at providing new recommendations based on data from multiple users. However, a disadvantage of both filtering is that as the amount of data increases, computing the similarity becomes more costly for both methods.

Overall, the book recommender system project provides a useful tool for readers who want to discover new books based on their past preferences. The system could be further improved by incorporating more advanced techniques, such as collaborative filtering or deep learning algorithms. With more data and advanced techniques, the system could provide even better recommendations for a wider range of readers.

6. References

https://sites.google.com/eng.ucsd.edu/ucsdbookgraph/home

https://towardsdatascience.com/comprehensive-guide-on-item-based-recommendation-systemsd67e40e2b75d

https://datascienceschool.net/03%20machine%20learning/07.01%20%EC%B6%94%EC%B2%9C%20%EC%8B%9C%EC%8A%A4%ED%85%9C.html