# CP421 Final Project

# Book Recommendation System

Mariam Lo

Melissa Pinto

Yvonne Itangishaka

Hoi Hin Ng

December 12, 2023

# Contents

# 1 Introduction

In recent decades, the influence of recommendation systems has been increasingly evident through platforms such as YouTube, Amazon, Netflix, and various other online services. These systems, designed to suggest items to users, play an integral role in our online experiences, spanning different sectors like online shopping, advertising, content streaming, and engagement. Recommendation systems also have a pivotal role in revenue generation and market differentiation. The ”Netflix prize,” a competition organized by Netflix a few years ago with a million-dollar prize, underscores the critical importance of efficient recommender systems in today’s digital landscape.

This project takes a comprehensive exploration of various recommender algorithm approaches, including collaborative and content-based filtering to create a hybrid recommendation system model. This is done by utilizing straightforward yet illustrative datasets. Each algorithmic approach will be explored, unraveling its inner workings, delving into the theoretical foundations, and assessing its strengths and weaknesses. Our objective is to gain insights into the evolution of recommender systems and their profound influence on user engagement and satisfaction.

# 2 Data

## 2.1 Data Set Overview

The data set utilized for this project is the Book Recommendation data set obtained from Kaggle and compiled by Cai-Nicolas Ziegler[1]. It was gathered through a 4-week crawl in August/September 2004 from the Book-Crossing community, containing anonymized information from 278,858 users providing 1,149,780 ratings on 271,379 books. Compromising three key files ( ’Users.csv’, ’Books.csv’, and ’Ratings.csv’), these datasets offer a comprehensive view of user demographics, book details, and user ratings respectively. The key variables in these files are as follows:

- Users - This section pertains to user information. It is important to note that ‘user-ID’s have been anonymized and mapped to integers. If available, demographic information such as ‘Location’, and ‘Age’ is provided; otherwise, these fields contain NULL values.

- Books - Books in this file are identified by their specific ISBNs, with invalid ISBNs removed for accuracy. It also contains content-based information such as ‘Book-Title’, ‘Book-Author’, ‘Year-Of-Publication’, and ‘Publisher’, sourced from Amazon Web Services. In cases with multiple authors, redundancy is avoided by retaining only one author. Additionally, URLs are provided for cover images in varied sizes (‘Image-URL-S’, ‘Image-URL-M’, ‘Image-URL-L’), linking to the Amazon Website.

- Ratings - This section contains information related to books. ‘Book-Rating’ has two distinct types: explicit and implicit. Explicit ratings are expressed on a scale from 1-10, with higher levels indicating greater appreciation. Implicit ratings are denoted by a value of 0, signifying an indirect or implicit form of feedback.

## 2.2 Dataset Preprocessing

To ensure data quality and relevance, various pre-processing steps were implemented across the three files. Initially, a new column named ‘Book-Data’ was created to concatenate relevant columns from the ‘Books’ file. Then we eliminated columns in the ‘Book’ file containing URLs for book covers (‘Image-URL-S’, ‘Image-URL-M’, ‘Image-URL-L’) since they were unnecessary for our future models. After dropping these columns, attention was turned to handling missing values and duplicates, recognizing their potential impact on data quality. Fortunately, no duplicates were identified in any of the files. However, a substantial number of 110762 ‘Age’ values, were missing, which was possibly connected to data not being collected or data loss. Additionally, there was 1 missing ‘Book-Author’, 2 missing ‘Publisher’, and 6 missing ‘Book-Data’ values. To avoid any issues in model training, it was decided to drop these values since this does not affect our data usage or system implementation.

## 2.3 Exploratory Data Analysis

Before implementing the Hybrid (Collaborative - Content filtering) model, we conducted an exploratory data analysis (EDA) to gain an understanding of our data demography in terms of the popularity of books and the distribution of users’ ratings. Descriptive statistics were calculated to understand the numeric characteristics of the data set, laying the groundwork for subsequent visualizations and threshold-setting based on average book ratings.

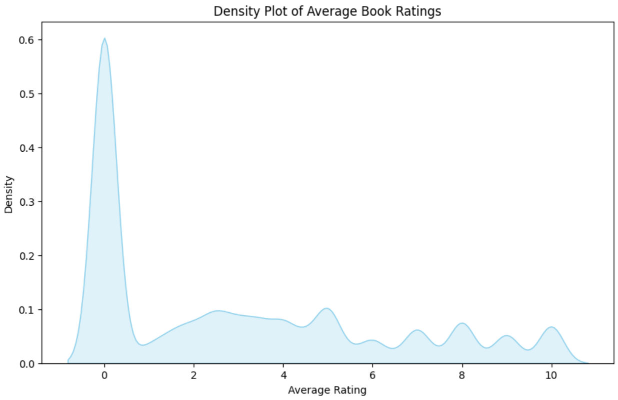


Figure 1: Density Plot of Average Book Ratings

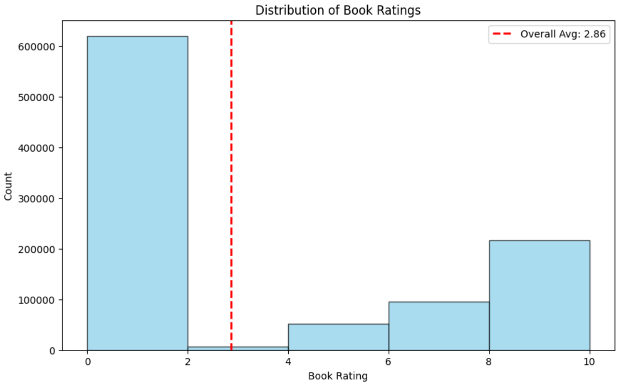


Figure 2: Distribution of Book Ratings

Notably, the Book-Rating distribution revealed a sizable portion of ratings at 0, indicating a possible skewness toward lower ratings and the presence of users who have not provided ratings. This is indicative of a cold start problem, where insufficient data hinders accurate recommendations.

The merging of tables facilitated the creation of a unified data set for analysis, combining the ’Books’, ’Ratings’, and ’Users’ data sets based on common identifiers. The core of our Exploratory Data Analysis involved creating key visualizations to comprehend variables and their relationships:

- Publication Year Trends - A density plot and count plot uncovered patterns in book publication years, highlighting significant periods in the year 2000’s and potential outliers at year 0, which may suggest missing or undefined values.

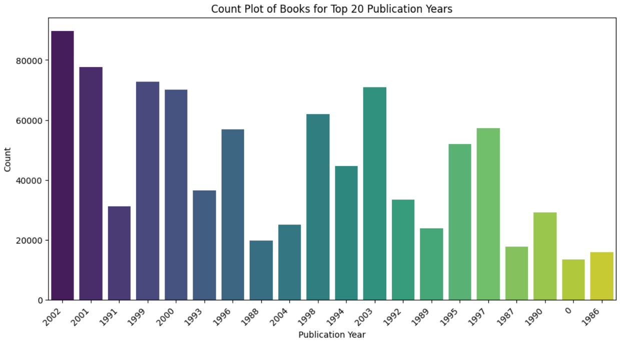


Figure 3: Count Plot of Books for Top 20 Publication Years

- Rating Sentiment - Density plots of average book ratings and the distribution of individual book ratings provided insights into the overall sentiment and the prevalence of different rating levels. Notably, many books had an average rating of 2.84, and a substantial number had no ratings (defined as 0).
- User Engagement - A histogram highlighted the distribution of user rating counts, offering a quick understanding of user engagement patterns, with most users located in the USA.

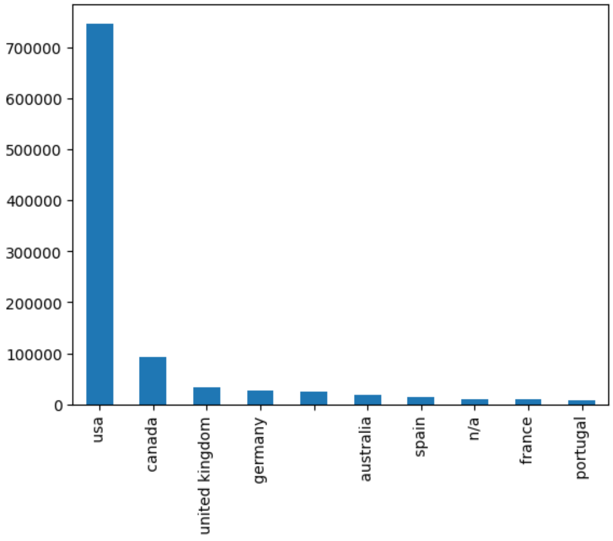


Figure 4: Users vs Countries

To enhance data set reliability, we applied a threshold based on rating counts for both books and users, retaining only those with enough ratings. Considering the data set size, visualizations, and calculated statistics, we chose to filter based on books with over 50 ratings and users who have rated over 50 books. This exploratory analysis, enhanced by visualizations, provided a deep understanding of the data set's information and trends. These insights will serve as a robust foundation for building our model, ensuring that our recommendations are founded on solid and reliable information.

## 3 Methodology

The analysis of this dataset was conducted using Python Programming language with prominent libraries such as Pandas, NumPy, Scikit-Learn, and others. The specific versions of these tools are detailed in the project environment setup.

The Hybrid (Collaborative-Content filtering) model was chosen due to its capacity to leverage both user-user item interactions (collaborative filtering) and item characteristics (content-based filtering). This model effectively handles dataset sparsity, offers personalized recommendations, and accommodates less-rated or new books, addressing the cold start issue identified during the data exploration stage. This model was created using a two-step approach: collaborative filtering and content-based filtering. Collaborative filtering uses user-item interactions to identify similar users and items, whereas content-based filtering focuses on the characteristics of books.

To refine the dataset, thresholds were applied to filter items with fewer than 50 ratings and users who rated fewer than 50 books. This strategic refinement aimed to enhance dataset reliability by retaining items and users with significant engagement.

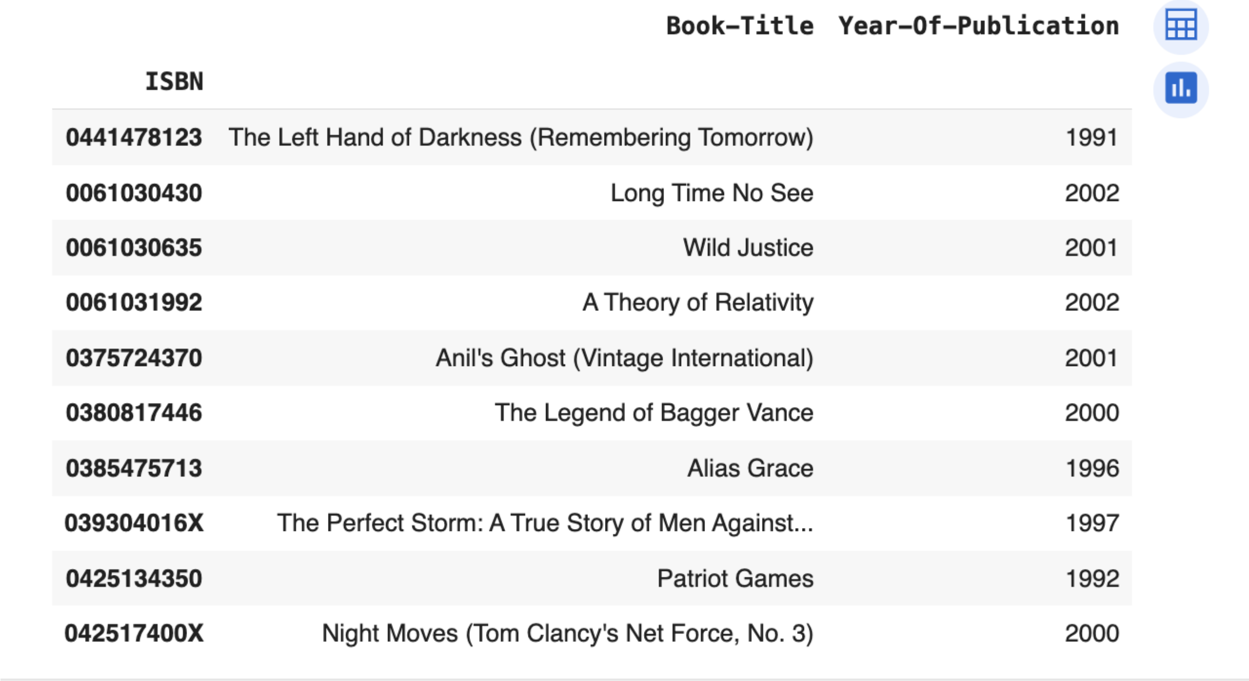
The recommendation system starts with collaborative filtering, utilizing user-based and item-based approaches to customize book recommendations based on user preferences. User-based collaborative filtering calculates Pearson similarity between users, predicting a target user's likes. Simultaneously, item-based collaborative filtering focuses on book similarities. In Content-Based Filtering, the process includes initialization, TF-IDF matrix creation, similar book recommendations, TF-IDF matrix exploration, and cosine similarity calculation. The primary aim is to provide personalized book recommendations based on book content, such as titles. A Book Recommender method was also created in both collaborative and Content-based filtering to generate the topN books to be recommended to the user. In our case, topN was equal to 10.

After model creation, accuracy assessment involved various metrics, including accuracy, precision, recall, Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE). These metrics collectively provide a comprehensive evaluation of model performance.

Following the creation of collaborative and content-based models, a hybrid recommendation system was implemented. By employing a weighted combination of collaborative and content-based recommendations, the hybrid model delivers robust and accurate suggestions for diverse scenarios. This enhances recommendation accuracy by addressing limitations found in standalone models. While collaborative filtering captures user preferences based on historical interactions, content-based filtering considers book attributes. This combination enhances recommendation accuracy, addressing limitations present in standalone models.

## 4 Results

Collaborative filtering utilizes user-based approach and item-based approaches to customize book recommendations based on user preferences and engagement. It leverages the behaviors and preferences by using the rating interaction of the user with the books. User-based approach collaborative filtering suggests items to a user based on similarities between items the user has interacted with and other similar items. The model calculated Pearson similarity between users' ratings, predicting a target user's likes. person similarity finds the correlation between user-item objects to calculate the results. The BookUserRecommender method puts together user-item results and generates the top recommended books for the user. In this project we went with the top 10 books to be recommended to the user as the is is a reasonable number of books.

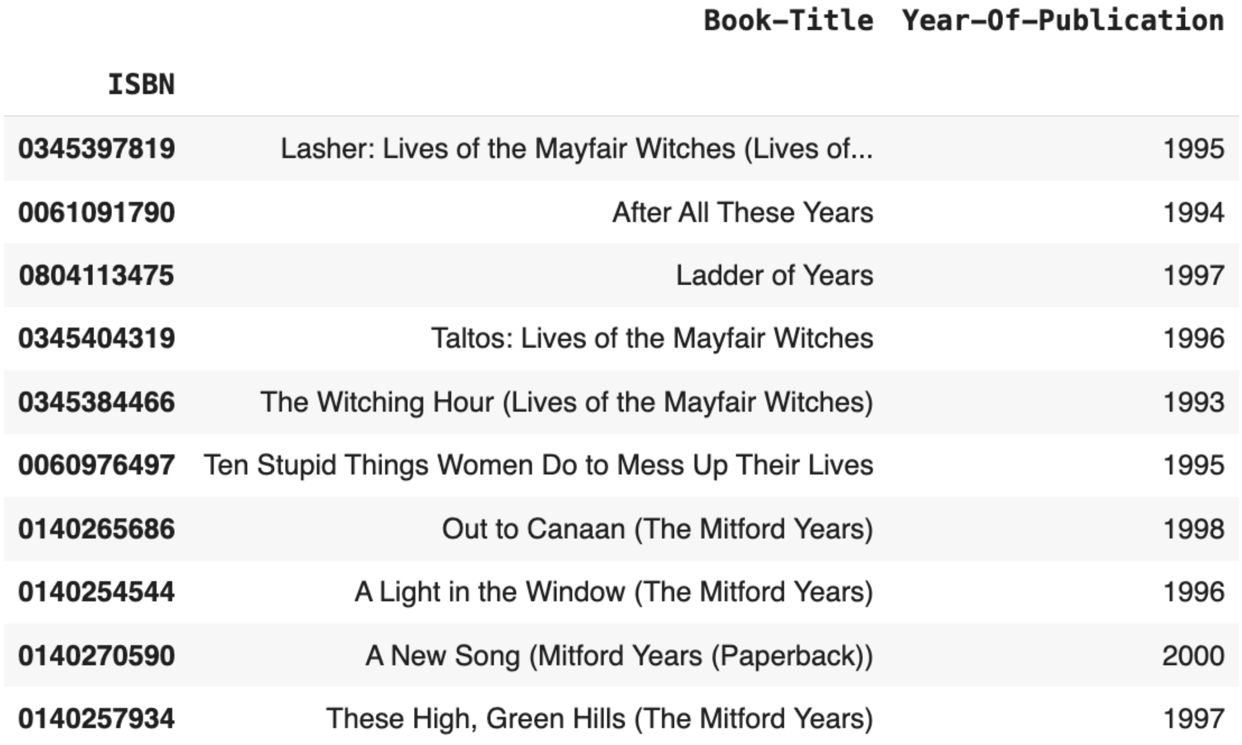


| | Book-Title | Year-Of-Publication |
|---|---|---|
| ISBN | | |
| 0441478123 | The Left Hand of Darkness (Remembering Tomorrow) | 1991 |
| 0061030430 | Long Time No See | 2002 |
| 0061030635 | Wild Justice | 2001 |
| 0061031992 | A Theory of Relativity | 2002 |
| 0375724370 | Anil's Ghost (Vintage International) | 2001 |
| 0380817446 | The Legend of Bagger Vance | 2000 |
| 0385475713 | Alias Grace | 1996 |
| 039304016X | The Perfect Storm: A True Story of Men Against... | 1997 |
| 0425134350 | Patriot Games | 1992 |
| 042517400X | Night Moves (Tom Clancy's Net Force, No. 3) | 2000 |

Figure 5: Collaborative Filtering Recommendations

The Content-based book recommendation system leveraging the TF-IDF and cosine similarity will provide personalized suggestions based on the content of items such as the book title or book author. The Content-based filtering class constructor is initialized with the training data that is split from the “merged_data” dataset, it contains the processed book information from the previous data preprocessing stage. For the TF-IDF vectorizer configuration, the constructor configures the “TfidfVectorizer” to exclude common English stop words, and the TF-IDF matrix is generated based on the “Book-Data” column from “merged_data” that highlights the importance of each word in each document. The TF-IDF is applied to the textual content of books, capturing the significance of words within each book title and this matrix serves as the foundation for personalized content-based recommendation.

The Content-based recommender method takes parameters generated by the TF-IDF process, and then uses the cosine similarity calculation to find the index of the specified book in the dataset, it then calculates cosine similarity scores between this book and all others using the TF-IDF matrix. Finally, the recommender will select the top 10 similar books for the user based on the similarity scores. In the return information, this method will return the information such as “Book-Title”, “Book-Author”, and “Year-Of-Publication” for the selected top 10 similar books. For example, if the user provided the book title “Wild Justice” as an input, then this method will return the top 10 similar books and their detailed information to the users. In this case, it suggested the top 10 similar books such as “after all these years”, “A new song”, etc.



| ISBN | Book-Title | Year-Of-Publication |
|---|---|---|
| 0345397819 | Lasher: Lives of the Mayfair Witches (Lives of... | 1995 |
| 0061091790 | After All These Years | 1994 |
| 0804113475 | Ladder of Years | 1997 |
| 0345404319 | Taltos: Lives of the Mayfair Witches | 1996 |
| 0345384466 | The Witching Hour (Lives of the Mayfair Witches) | 1993 |
| 0060976497 | Ten Stupid Things Women Do to Mess Up Their Lives | 1995 |
| 0140265686 | Out to Canaan (The Mitford Years) | 1998 |
| 0140254544 | A Light in the Window (The Mitford Years) | 1996 |
| 0140270590 | A New Song (Mitford Years (Paperback)) | 2000 |
| 0140257934 | These High, Green Hills (The Mitford Years) | 1997 |

Figure 6: Content-based Filtering Recommendations

| Approach | RMSE | Precision | Recall | F1-score |
|---|---|---|---|---|
| Collaborative Filtering (KNN) | 3.51 | 30% | 1 | 0.46 |
| Content-Based Filtering (TF-IDF) | 3.79 | 40% | 0.002 | 0.005 |
| Hybrid Model | 4.64 | 80% | 1 | 0.88 |

Figure 7: RMSE Summary

**Interpretation:**

- **Collaborative Filtering (KNN):** Shows a moderate RMSE, low precision, and high recall and F1-score, indicating that while it might recommend many relevant items, it also suggests many irrelevant ones.

- **Content-Based Filtering (TF-IDF):** Presents a slightly lower RMSE, but very low precision, recall, and F1-score, indicating that the recommendations are imprecise and miss many relevant items.
- **Hybrid Model:** Has the highest RMSE but significantly improved precision, recall, and F1-score, suggesting that combining collaborative and content-based approaches resulted in more accurate and comprehensive recommendations.

## 5 Conclusion

In conclusion, this project explored various recommendation systems, leveraging Python and powerful libraries including Pandas, NumPy, and Scikit-Learn to derive insights from the diverse Book Recommendation dataset. The final Hybrid recommendation model successfully addressed challenges such as data sparsity and the cold start problem. It leveraged user-user interactions and item characteristics to provide personalized recommendations. Collaborative filtering utilized historical interactions, while content-based filtering considered specific book attributes like titles and authors. By performing evaluations through diverse metrics, we were able to confirm the hybrid model's reliability in offering meaningful book suggestions.

Despite successfully addressing challenges such as data sparsity and the cold start problem, it's noteworthy that the Root Mean Squared Error (RMSE) remains relatively high. To further improve the model's accuracy and performance, future projects can explore different strategies such as fine-tuning model parameters or incorporating additional features like book genres. This can help to provide more complex information for making recommendations and contribute to the ongoing improvement of the recommendation system.

## References

[1] Book Recommendation Dataset. https://www.kaggle.com/datasets/arashnic/book-recommendation-dataset, 2004.