Book Recommender System Project

UNIVERSITY OF CAPE TOWN

Name: G Khuswana **Student No.: KHSGUD001**

Course Name: Data Science for Industry **Course Code: STA5073Z**

September 2024

Table of contents

1 Plagarism Declaration

I, Gudani Khuswana, declare that:

1. This work is my own and has not been copied from any other source.
2. All references and sources used have been properly cited.
3. I have not submitted this work elsewhere for credit.
4. I understand the consequences of academic dishonesty.

Signature: 

Date: 25/09/2025

2 Introduction

In a world inundated with choices, personalized recommendations are essential for guiding users toward relevant content. This project aims to develop an ensemble recommender system that suggests books to users based on their past evaluations, utilizing the Book-Crossing dataset, which includes over 278,000 users and more than 271,000 book ratings.

The recommender system will employ collaborative filtering techniques—both user-based and item-based—alongside matrix factorization. This approach will address the cold-start problem for new users, who may initially provide only a few ratings. The project will assess the accuracy of each method individually and then combine them in an ensemble model.

Through exploratory data analysis and thoughtful data reduction, this report seeks to evaluate the effectiveness of these recommendation strategies, ultimately enhancing the user experience in selecting books.

3 Data Preprocessing

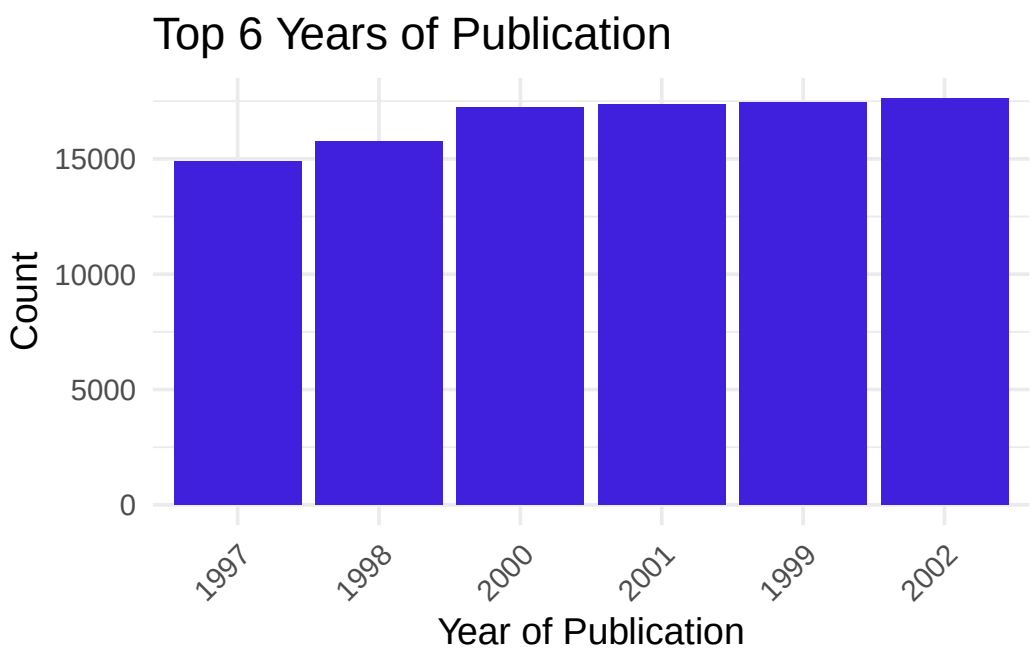
The dataset used for this analysis is considered reliable and substantial. It includes data on approximately 271,360 books and nearly 278,000 registered users, who have collectively provided around 1,149,780 ratings. This extensive dataset allows for a robust analysis and enhances the credibility of the findings presented in this report.

To ensure compatibility when constructing matrices with ISBNs and User IDs as row or column names, a transformation is applied to the dataset. Specifically, the prefix "Id" is added to all ISBNs and User IDs. This adjustment is necessary because R automatically adds a prefix of 'X' to column or row names that start with a number. By adding "Isbn." to ISBNs and "User." to User IDs, this issue is avoided, resulting in cleaner and more manageable data structures.

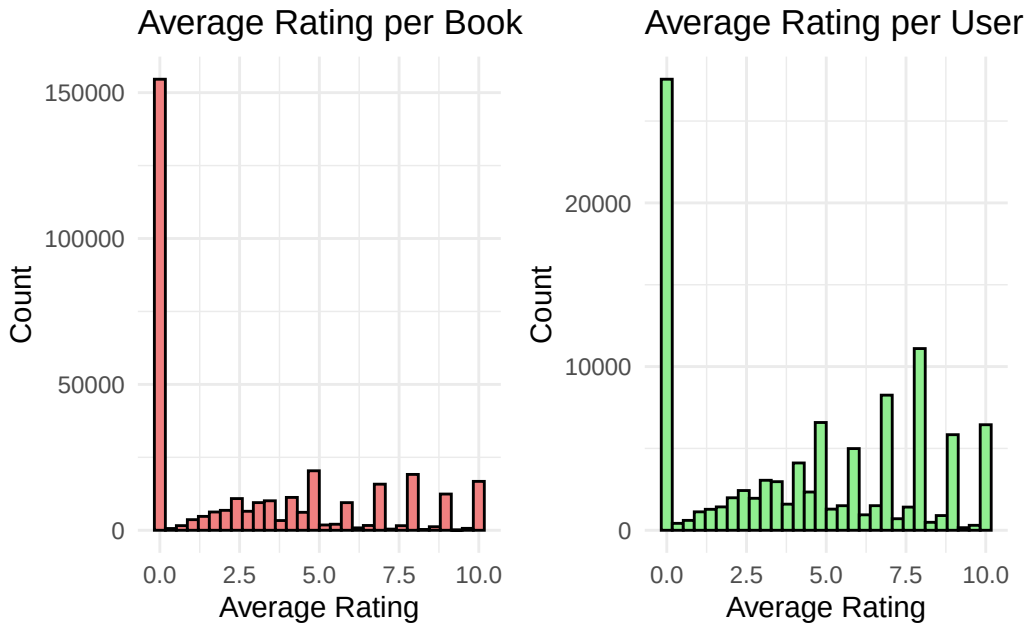
The data cleaning process for the 'Year-Of-Publication' column involved important steps to improve the dataset's integrity. Initially, invalid entries, such as non-numeric values and outliers like '0' and '1376', were filtered out. Valid year values were converted to integers, and a new datetime column was created to facilitate better handling of dates. The old 'Year-Of-Publication' column was dropped, and any remaining invalid years, including '2030' and '2050', were further removed.

4 Exploratory Data Analysis (EDA)

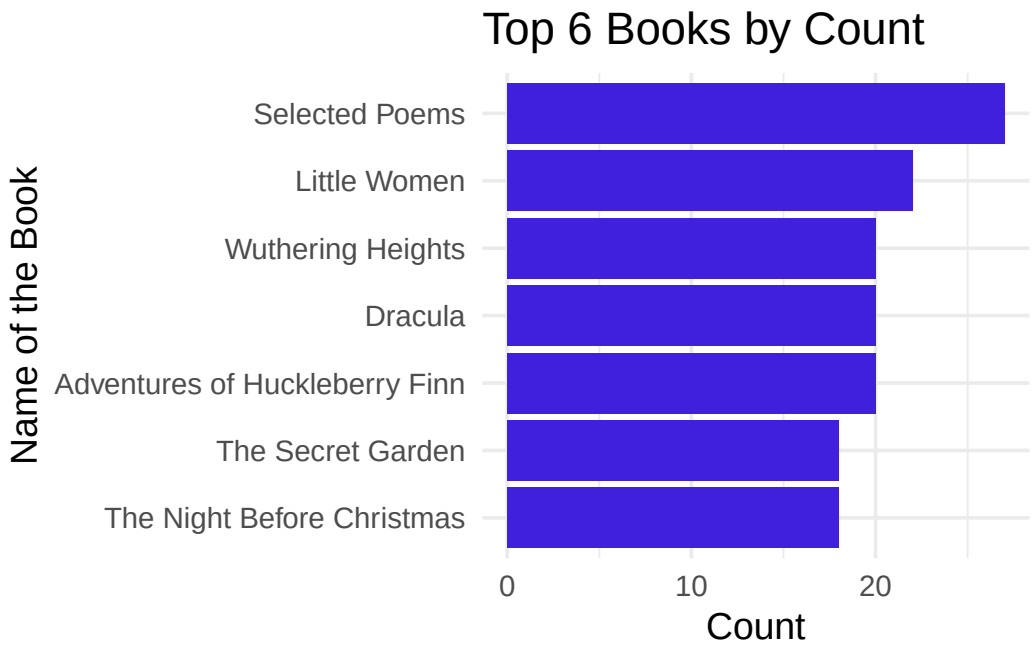
The bar chart displays the top 6 years of publication based on the number of books published. The years 1998, 1999, 2000, 2001, 2002, and 2003 are shown, with 2002 having the highest count of books published, nearing 10,000. The count for each year is close to 8,000–10,000 publications, indicating a consistent volume of books released across these years.



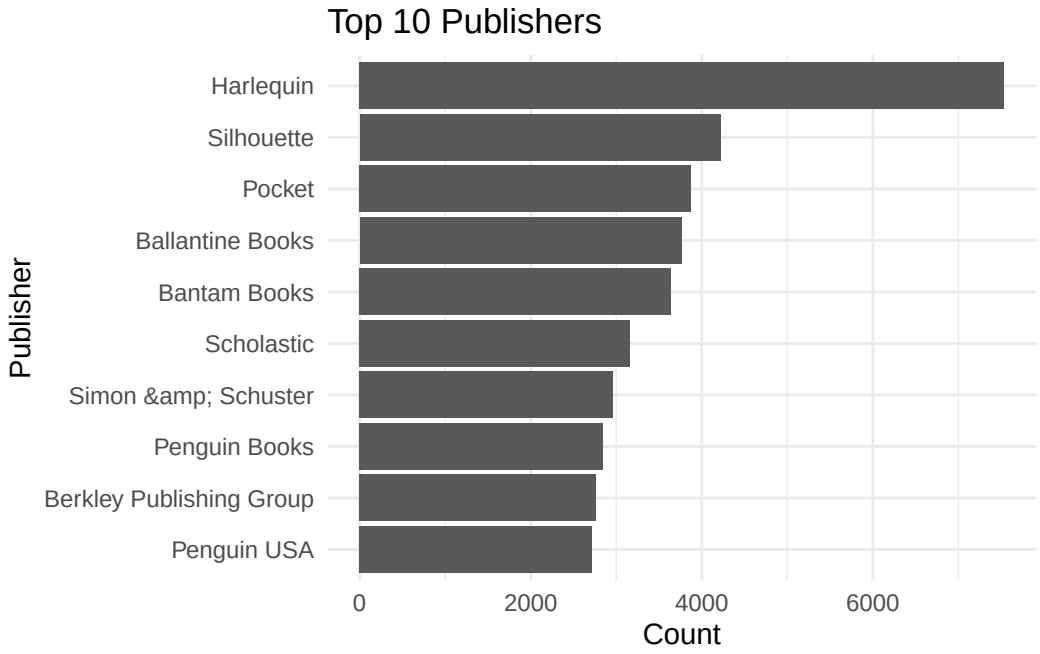
The Figure below display the distribution of average ratings. On the left, "Average Rating per Book" shows that most books receive an average rating of 0, with fewer books receiving higher average ratings. On the right, "Average Rating per User" shows a similar pattern, where the majority of users have an average rating of 0, but the distribution spreads more evenly across higher rating ranges. This indicates that a significant portion of books and users have low or no ratings, while a smaller group provides higher ratings more consistently.



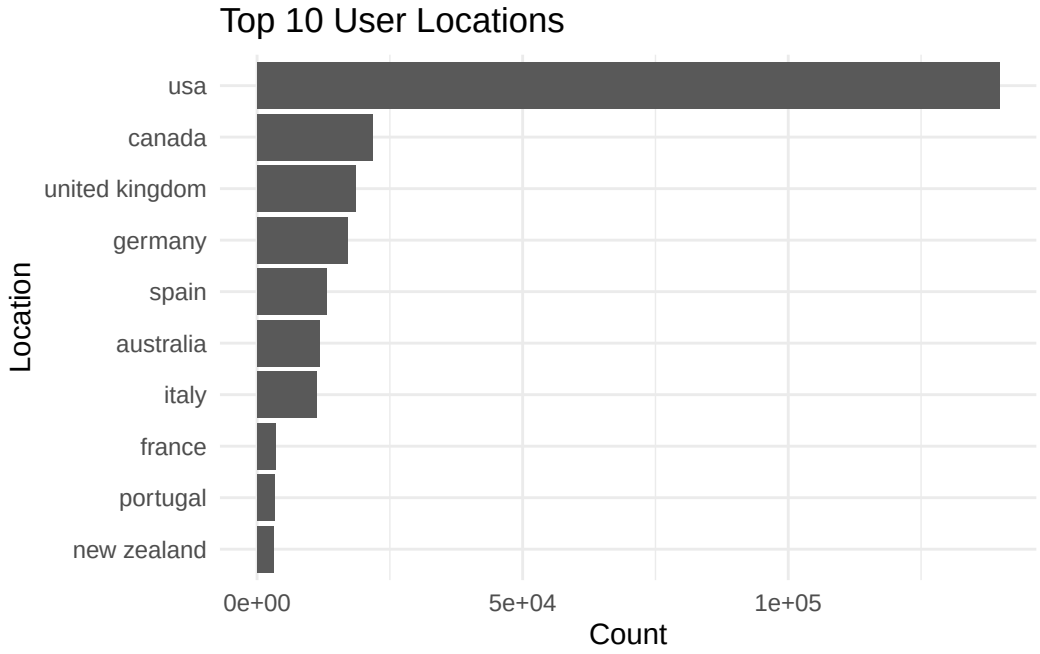
The bar chart shows the top six books based on their counts, reflecting their popularity. **Selected Poems** stands out as the most prominent, followed by **Little Women** and **Wuthering Heights**, which also have notable but slightly lower counts. The remaining books—**Dracula**, **Adventures of Huckleberry Finn**, **The Secret Garden**, and **The Night Before Christmas**—are still significant, though their counts are lower in comparison.



The Figure below displays the top 10 publishers from the book dataset, with **Harlequin** emerging as the most prolific publisher, having over 6,000 books, followed by **Silhouette** and **Pocket** with more than 3,000 books each. The chart illustrates that these three publishers have a significant dominance over the others, such as **Ballantine Books**, **Bantam Books**, and **Scholastic.**

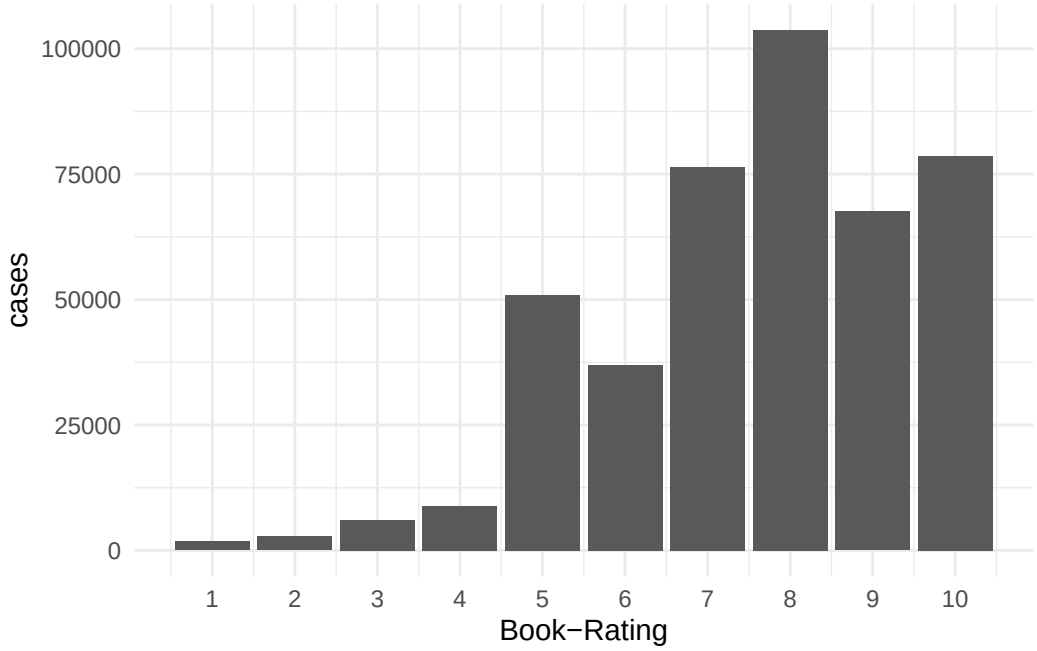


The bar chart below illustrates the top 10 user locations, with the **USA** having the largest number of users, far exceeding 100,000. **Canada** and the **United Kingdom** follow, each contributing a significant number of users to the dataset. Other countries like **Germany**, **Spain**, and **Australia** also feature prominently, while smaller contributions come from **Italy**, **France**, **Portugal**, and **New Zealand**. This distribution indicates that the user base is predominantly from English-speaking countries, with noticeable participation from several European nations.



To improve the reliability and relevance of the recommendation system, a filtering step was applied to the ratings dataset by removing entries where the `Book-Rating` was equal to 0. This action ensures that only meaningful ratings, where users actively rated the books, are considered in the analysis. A rating of 0 generally indicates no opinion or lack of feedback, which does not provide valuable information for predicting preferences. By excluding these non-informative ratings, the dataset becomes more focused on explicit user feedback, enhancing the accuracy of both item-based and user-based collaborative filtering models.

The bar chart below shows the distribution of ratings across books, following the exclusion of non-informative ratings (ratings of 0). The distribution indicates that users tend to give higher ratings, with a significant proportion of ratings clustered between 7 and 10. Specifically, the mode of the distribution is at a rating of 8, with over 100,000 cases, followed closely by ratings of 7 and 10. Lower ratings, particularly between 1 and 4, are relatively uncommon. This distribution reflects a positive skew, where users are more likely to rate books favorably, which may suggest a bias toward higher ratings in this dataset.



5 Data Reduction Strategy

In the dataset, users vary significantly in how many books they rate, which impacts the quality and usefulness of the data for building recommender systems. Initially, the number of ratings per user was summarized, showing a wide distribution. The summary statistics reveal that most users have rated a very small number of books, with a median of just one rating per user and an average of approximately 5.57 ratings. Some users, however, are highly active, with the maximum number of ratings reaching 8,524.

This process serves to reduce the noise in the dataset and make it more manageable for both exploratory data analysis and model training. It also mitigates the cold-start problem by focusing on users who have provided enough information to generate reasonable recommendations.

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   1.000   1.000   5.574   3.000 8524.000
```

In this project, a pivot table was created from a subset of the top 20,000 user ratings from the dataset. This involved transforming the user-item interaction data, where the `User-ID` was set as the row identifier and the unique `ISBN` values were spread across columns representing books. Missing ratings were filled with zeros to ensure all books were represented. The resulting pivot table was then converted into a matrix for further analysis. Upon examining the first 25 rows and columns, it was evident that many users have not rated a majority of the books, resulting in a sparse matrix where the majority of entries are zeros. This sparsity reflects a common challenge in collaborative filtering systems, where users rate only a few items, leading to difficulties in generating accurate recommendations due to insufficient data.

6 Modelling

This project will develop an ensemble recommender system to predict book ratings using three collaborative filtering techniques: User-Based Collaborative Filtering (UBCF), Item-Based Collaborative Filtering (IBCF), and Matrix Factorization (MF). The UBCF approach will identify users with similar rating patterns to generate recommendations, while the IBCF method will focus on the similarities between items to suggest books that are alike to those previously enjoyed by the user. For the Matrix Factorization approach, Singular Value Decomposition (SVD) will be employed to decompose the user-item rating matrix into latent factors that represent user preferences and item characteristics. The accuracy of the matrix factorization model will be assessed through a regularization approach, evaluating the impact of regularization on model performance. Finally, an ensemble model will be created to combine predictions from all three methods, enhancing overall recommendation accuracy. In this section it will besed on both existing users and new users to avoid the cold start problem.

6.1 Item-based recommendation system

Item-based Collaborative Filtering (ICF) is commonly used in recommender systems due to its effectiveness in modeling user preferences and its simplicity in providing personalized recommendations Xue et al. (2019). Similar products to those purchased by the user will be identified and recommended based on their resemblance to highly rated items.

6.1.1 Item-based CF on Existing Users

The system employs a cosine similarity function to measure the similarity between books while handling missing values by replacing them with zeros. By calculating similarities based on user ratings, the function recommends the top five books similar to a given target book by creating a function to calculate the similarity only on the product id that we choose. For example, when querying the book with ISBN `Isbn.0446677450`, the system returned several recommendations as show in the table below, all sharing a similarity score of 0.538. This means moderate similarity, indicating that while the recommendations are relevant, more refinement will improve precision.

```
# A tibble: 8 x 2
  ISBN            similarity
  <chr>                <dbl>
1 Isbn.0671621009      0.538
2 Isbn.006092943X      0.538
3 Isbn.0316748641      0.538
4 Isbn.0385235941      0.538
5 Isbn.0696214563      0.538
6 Isbn.0737003219      0.538
7 Isbn.0764504193      0.538
8 Isbn.1583331433      0.538
```

6.1.2 Item-based CF on New Users

This secton outlines the implementation of an item-based collaborative filtering recommendation system for new users, enabling personalized book suggestions based on user ratings. The results show a list of book recommendations for a new user based on their ratings of previously rated books as shown in the table below. Each entry includes the **ISBN** of a recommended book and its associated **similarity score**, which reflects how closely the book aligns with the user's preferences based on the provided ratings.

A higher similarity score (in this case, all entries have a score of approximately 0.615 as shown in the table below) indicates that these books are considered closely related to the user's interests, making them strong candidates for recommendation. The uniformity of the similarity scores suggests that these books share similar characteristics that resonate with the user's prior ratings

```
# A tibble: 17 x 2
   ISBN            similarity
   <chr>                <dbl>
 1 Isbn.0029087104      0.615
 2 Isbn.0030615321      0.615
 3 Isbn.0226726770      0.615
 4 Isbn.0345285549      0.615
```

```
 5 Isbn.0345307674      0.615
 6 Isbn.0345331605      0.615
 7 Isbn.0385314744      0.615
 8 Isbn.0394312066      0.615
 9 Isbn.0425050750      0.615
10 Isbn.0451169530      0.615
11 Isbn.0553801031      0.615
12 Isbn.0671461494      0.615
13 Isbn.067163884X      0.615
14 Isbn.0766607119      0.615
15 Isbn.0894803700      0.615
16 Isbn.0939766027      0.615
17 Isbn.1895383129      0.615
```

6.2 User-based Collaborative Recommendation

Collaborative Filtering relies on three key assumptions: individuals tend to share similar preferences and interests, these preferences remain consistent over time, and future choices can be predicted based on past behavior. The algorithm works by comparing a user's behavior with that of others to identify similar users, known as "nearest neighbors." It then predicts the user's preferences based on the interests or choices of these neighbors Zhao and Shang (2010).

6.2.1 User-based CF for Exisitng users

In this section, a cosine similarity function was defined to calculate the similarity between book ratings by users. Item recommendation function was created, which uses this similarity to identify books related to a specified book based on user ratings. The function was tested using the book "Isbn.0767912098," yielding a tibble of five recommended books along with their rating counts and average ratings as as shown below. A count of 1 for all recommended books means that each of these books has only been rated by one user in the dataset. The results indicate potential recommendations for users based on existing preferences, showing the effectiveness of this user–based cf.

```
# A tibble: 5 x 3
# Rowwise:
  ISBN            count avg_rating
  <chr>           <int>      <dbl>
1 Isbn.006251279X     1         10
2 Isbn.0062732757     1         10
3 Isbn.0141301201     1          5
4 Isbn.0345323211     1          5
5 Isbn.0380006340     1          8
```

6.2.2 User-based CF for New Users

A recommendation function was created for new users based on their ratings of a few existing books. The new user's ratings were added to the rating matrix, and their similarity to other users was calculated using cosine similarity. Unrated books were identified, and recommendations were generated based on similarity scores. Each recommended book included the number of ratings and the average rating from other users.

The results showed four recommended books. The first book, "Isbn.0312252617," had the most ratings (3) and a high average rating of 8.33, indicating it is popular. The last book, "Isbn.0385235941," had only 1 rating with a lower average of 6, suggesting it is less favored. Overall, these recommendations help the new user find popular and well-liked books to read

```
# A tibble: 4 x 3
# Rowwise:
  ISBN            count avg_rating
  <chr>           <int>      <dbl>
1 Isbn.0312252617     3       8.33
2 Isbn.0312261594     2       7.5
3 Isbn.0316748641     1       7
4 Isbn.0385235941     1       6
```

7 Matrix factorization

Matrix factorization is a powerful approach for reducing data dimensions, uncovering hidden features, and addressing sparsity issues. It is commonly applied in recommender systems due to these strengths. One popular matrix factorization technique used in recommenders is Singular Value Decomposition (SVD) Mehta and Rana (2017).

The recommendation system utilizes the **Recosystem** package, employing matrix factorization to predict book ratings for both existing and new users. The dataset underwent preparation, including filtering user ratings to a maximum of 50, an 80/20 train-test split, and converting ISBNs to numeric factors. The model was trained with 20 latent dimensions and a learning rate of 0.1, resulting in a significant reduction in RMSE from 5.9392 to 0.2966 over 20 iterations, indicating effective learning. Predictions were made on the test data to estimate user ratings, while a strategy to address the cold-start problem was implemented by allowing new users to provide up to five explicit ratings.

The table compares the actual user ratings with the predicted ratings, allowing for the evaluation of the model's accuracy in predicting book ratings. Higher alignment between actual ratings (`Book-Rating`) and predicted ratings (`Predicted-Rating`) indicates better model performance. **User-ID**: 276747 rated a book with ISBN **1891** with a **rating of 6**, and the model predicted a **rating of 7.10**. This suggests that the model has reasonably approximated the user's preference. This analysis will be further extended by calculating RMSE to quantify the accuracy of the model.

```
iter      tr_rmse          obj
   0       5.9392   6.5445e+06
   1       3.3213   2.0467e+06
   2       1.9689   7.1922e+05
   3       1.4105   3.6912e+05
   4       1.1571   2.4841e+05
   5       1.0214   1.9357e+05
   6       0.9248   1.5868e+05
   7       0.8425   1.3168e+05
   8       0.7673   1.0924e+05
   9       0.6975   9.0273e+04
  10       0.6337   7.4510e+04
  11       0.5764   6.1643e+04
  12       0.5252   5.1180e+04
  13       0.4796   4.2684e+04
  14       0.4391   3.5776e+04
  15       0.4031   3.0145e+04
  16       0.3718   2.5641e+04
  17       0.3436   2.1908e+04
  18       0.3187   1.8845e+04
  19       0.2966   1.6319e+04
```

```
  User-ID  ISBN Book-Rating Predicted-Rating
1  276747 18916           9         7.100761
2  276747 27255           7         8.556341
3  276813 30289           8         7.774769
4  276822  4000           9         3.777657
5  276822  4093          10         8.885351
6  276822 22937          10         5.792573
```

7.1 Assess Matrix Factorization

In this section, the accuracy of the matrix factorization recommender system was evaluated with and without regularization. The models were trained using the same dataset, with the regularization parameters set to zero for the first model and to 0.1 for the second. The training process showed a decrease in training RMSE for both models, indicating effective learning. The RMSE on the test set showed that the model with regularization achieved a lower error rate of 2.317 compared to 2.368 for the model without regularization. This suggests that incorporating regularization improved the model's ability to generalize to unseen data, reducing overfitting and improving its predictive accuracy for book ratings.

```
RMSE without regularization: 2.366356
```

```
RMSE with regularization: 2.318515
```

8 Model Ensemble

In this section, the process involves combining predictions from three recommendation methods—item-based collaborative filtering, user-based collaborative filtering, and matrix factorization—into a single data frame (`ensemble_predictions`) using ISBN as a common identifier. By calculating the average predicted ratings for each book, the ensemble approach leverages the strengths of individual models, improving prediction accuracy and reducing bias and variance. The ensemble predictions are then compared to actual ratings from the test data, which is important for assessing performance. Root Mean Squared Error and Mean Absolute Error are computed to quantify the accuracy of the ensemble predictions, providing a comprehensive evaluation of the model's performance.

The accuracy metrics obtained included a Root Mean Squared Error (RMSE) of 2.6535 and a Mean Absolute Error (MAE) of 2.1310. The RMSE value was higher compared to the matrix factorization model with regularization, which achieved an RMSE of 2.32. This indicates a moderate level of prediction error, suggesting that while the ensemble model captures some trends in the data, there remains a significant discrepancy between the predicted and actual ratings. The performance of the combined model is not as effective as that of the matrix factorization approach with regularization.

```
Root Mean Squared Error (RMSE) for Ensemble Predictions: 2.653546
```

```
Mean Absolute Error (MAE) for Ensemble Predictions: 2.13104
```

Reference

Mehta, Rachana, and Keyur Rana. 2017. "A Review on Matrix Factorization Techniques in Recommender Systems." In *2017 2nd International Conference on Communication Systems, Computing and IT Applications (CSCITA)*, 269–74. IEEE.

Xue, Feng, Xiangnan He, Xiang Wang, Jiandong Xu, Kai Liu, and Richang Hong. 2019. "Deep Item-Based Collaborative Filtering for Top-n Recommendation." *ACM Transactions on Information Systems (TOIS)* 37 (3): 1–25.

Zhao, Zhi-Dan, and Ming-Sheng Shang. 2010. "User-Based Collaborative-Filtering Recommendation Algorithms on Hadoop." In *2010 Third International Conference on Knowledge Discovery and Data Mining*, 478–81. IEEE.