A Project Report On

**Bookify: The Ultimate Book Recommendation Application With Data-Driven Intelligence**

*Submitted by*

Divya Pondkule (PL)

(220940325029)

Dhiraj Mankar

(220940325026)

Rhutuja Wagh

(220940325054)

Twinkal Patil

(220940325084)

Deepam Gharsele

(220940325025)

*In partial fulfilment of*

*the requirements for the award of the degree of*

**Post Graduate Diploma**

**In**

**Big Data Analytics**

**(PG-DBDA)**

****

**Year 2023**

**Abstract**

The rapid growth of online bookstores has led to an increase in the number of books available for purchase, making it difficult for users to discover books that match their interests.

To address this challenge, we propose a Bookify : The Ultimate Book Recommendation Application With Data-Driven Intelligencethat leverages user behaviour and book metadata to provide tailored book recommendations.

**Table of Contents**

1. Introduction…………………………………………………………….. 04
2. Objectives……………………………………………………………….05
3. Research Motivation………………………………………………….... 06
4. Problem Statement………………………………………………………07
5. Project Development…………………………………………………….08
6. Methodology and Implementation………………………………………09
7. Future Scope…………………………………………………………….29
8. Conclusion ...……………………………………………………………30
9. **Introduction**

The growing availability of books online has made it challenging for readers to find books that match their interests. Bookstores and online platforms are inundated with numerous books, making it difficult for readers to navigate through them and choose books that fit their preferences. To help readers discover books they will enjoy, book recommendation systems have been developed to provide popularity recommendations based on user preferences and book metadata.

In this report, we propose a book recommendation system that utilizes user preferences and book metadata to provide personalized recommendations. Our system collects data on user reading history, ratings, and preferences to build user profiles. We use a collaborative-based filtering algorithm to analyse user profiles and recommend books that have similar content and features to those previously enjoyed by the user. Additionally, our system incorporates book metadata such as genre, author, and publisher to recommend books that match the user's interests.

1. **Objectives**

The objective of the book recommendation system project report is to design and implement an intelligent algorithm that can suggest books to users based on their reading preferences. The system should be able to analyse the user's reading history, preferences, and feedback to recommend books that are most likely to interest them.

The project aims to develop a system that can accurately predict the user's reading choices and provide personalized recommendations. The system should be user-friendly, scalable, and efficient in handling large amounts of data.

The project report will include an analysis of various recommendation algorithms, their strengths, and limitations. It will also include a detailed description of the system's architecture, data collection, and evaluation metrics.

The project aims to create a book recommendation system that provides an enhanced user experience and encourages users to explore new genres and authors. The report will also discuss potential future enhancements and developments of the system.

1. **Research Motivation**

The availability of online books has increased exponentially in recent years, making it difficult for readers to navigate through the vast selection of books available and find books that match their interests. Consequently, there is a need for book recommendation systems that can provide personalized recommendations to readers. These systems can help readers discover books that they will enjoy.

The project's motivation is to provide readers with book recommendations that match their interests and preferences. We are recommending books using collaborative filtering (k-NN), showing top 50 books in entire collection and books popular yearly (popularity based filtering).

1. **Problem Statement**

With the growing availability of books online, it has become increasingly difficult for readers to navigate through the vast selection of books and find books that match their interests. While traditional bookstores rely on staff recommendations, online bookstores have utilized recommendation systems to help users find books they will enjoy. However, many of these systems are limited in their ability to provide personalized recommendations based on user preferences and book metadata.

Therefore, the problem this project aims to address is to develop a book recommendation system that can provide tailored book recommendations to users based on book metadata. The system will utilize machine learning techniques to pre-process book data and extract relevant features from book descriptions and metadata.

1. **Project Development**

The book recommendation system project will involve several stages of development. First, the project team will collect and pre-process book data, including book metadata and descriptions. Next, they will develop machine learning models to analyse the data and extract relevant features from the book descriptions. The team will then develop a recommendation engine that utilizes collaborative filtering techniques to identify similar users and recommend books that those users have enjoyed. The system will also incorporate popularity based filtering algorithms to match the frequency of usage and popular items of the books to the users. The final product will be a book recommendation system that provides book recommendations based on user preferences and book metadata.

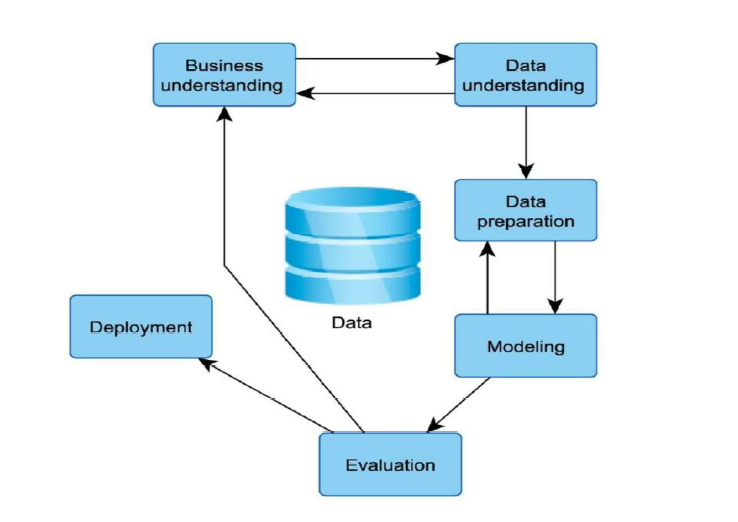


Figure 1: Project Development

1. **Methodology and Implementation**

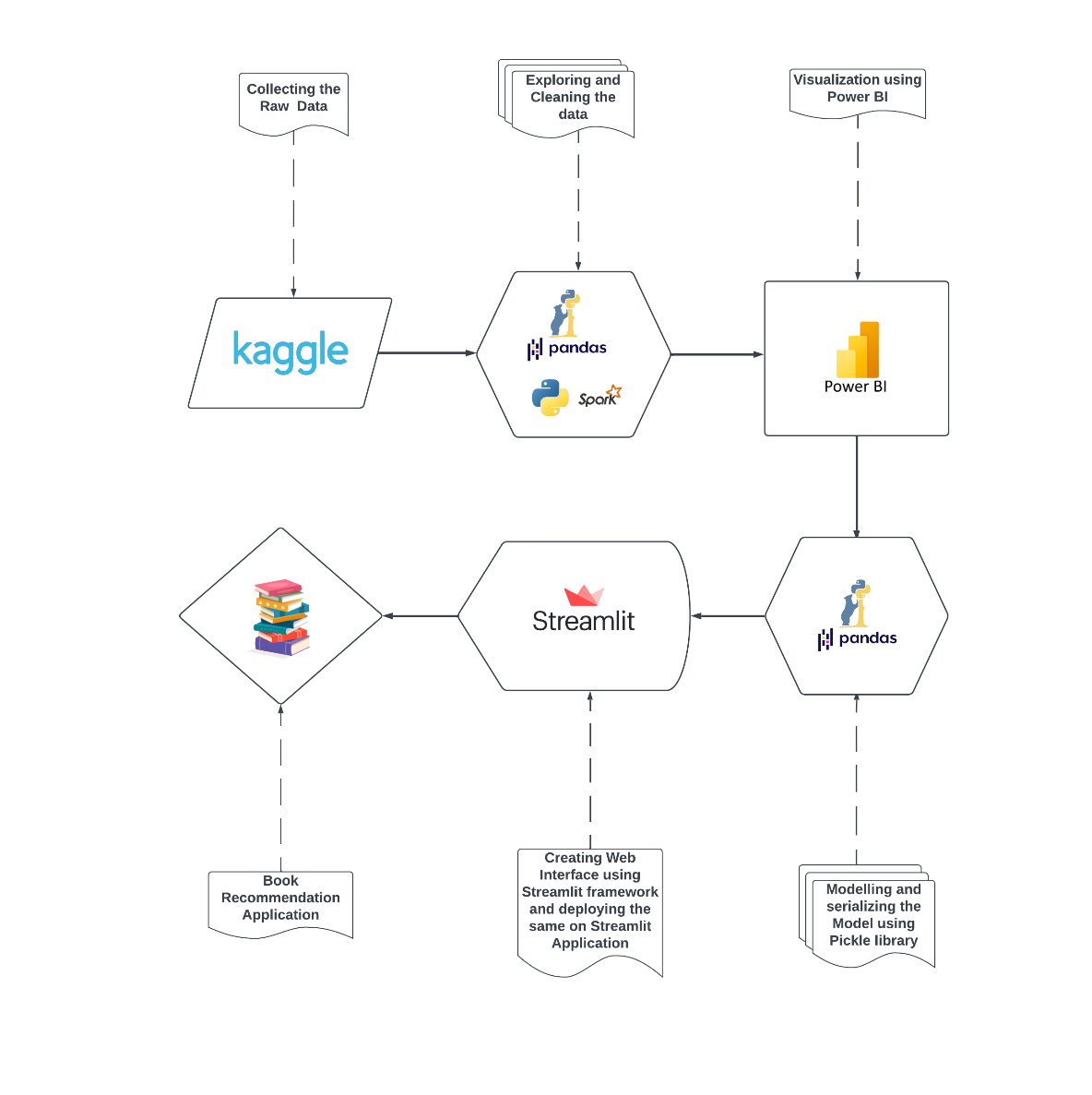
****

Figure 2: Project Architecture

**Technologies Used :**

1. Pyspark
2. Pandas
3. Machine Learning Algorithm
4. Streamlit
5. Power Bi
6. **Data Source:**

The data source that we will be using in the is analysis is a dataset

from Kaggle which contains the Book-Crossing dataset comprises 3 files.

* Users  
  Contains the users and having rows approximately more than 0.2 million. Note that user IDs (User-ID) have been anonymized and map to integers. Demographic data is provided (Location, Age) if available. Otherwise, these fields contain NULL-values.
* Books  
  Books are identified by their respective ISBN. Invalid ISBNs have already been removed from the dataset. The dataset contains rows approximately more than 0.2 million. Moreover, some content-based information is given (Book-Title, Book-Author, Year-Of-Publication, Publisher), obtained from Amazon Web Services. Note that in case of several authors, only the first is provided. URLs linking to cover images are also given, appearing in three different flavours (Image-URL-S, Image-URL-M, Image-URL-L), i.e., small, medium, large. These URLs point to the Amazon web site.
* Ratings  
  Contains the book rating information and have rows approximately more than 1 million. Ratings (Book-Rating) are either explicit, expressed on a scale from 1-10 (higher values denoting higher appreciation), or implicit, expressed by 0.

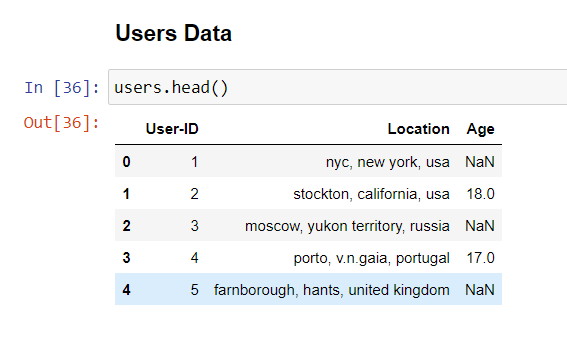


Figure :3

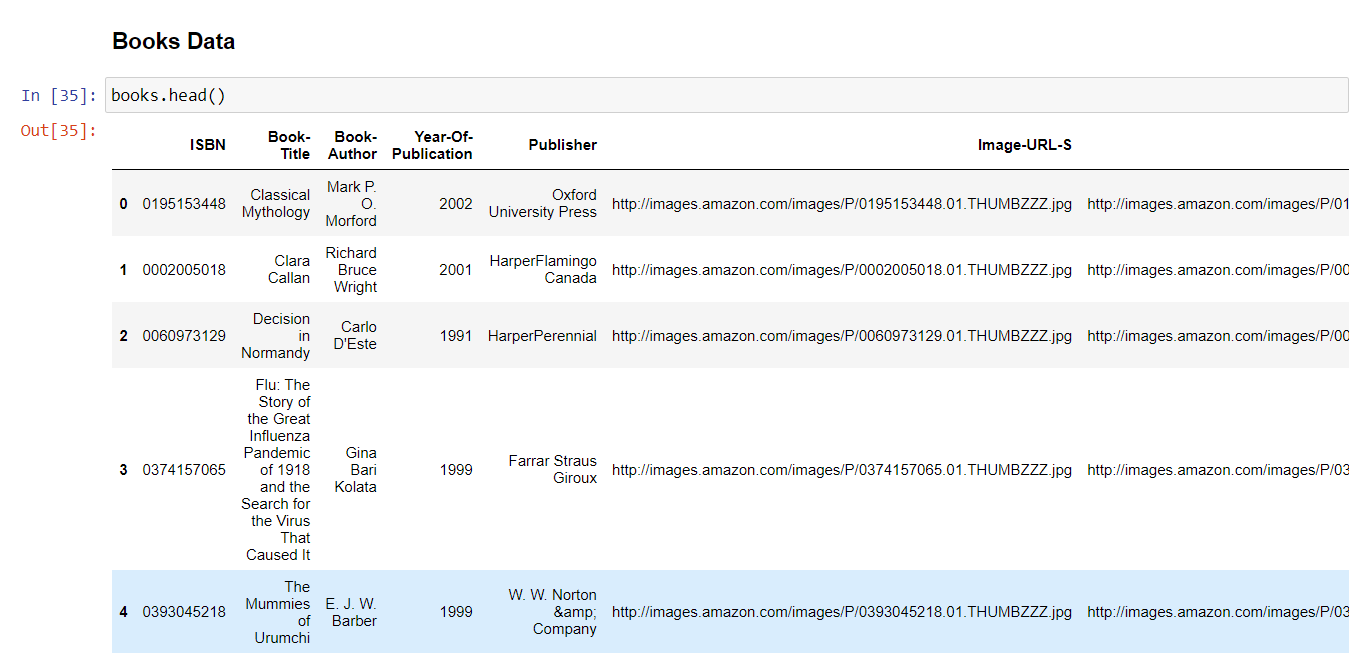


Figure :4

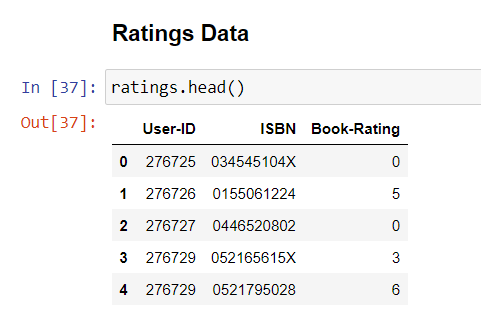
****

Figure :5

1. **Data Storage**

We have used local machine to store the data.

1. **Data Pre-processing**

Data pre-processing is an essential step in developing a book recommendation system as it involves cleaning, transforming, and preparing raw data into a format that can be used for training and testing machine learning models.

* 1. **Books Dataset:** In this dataset, we checked for null values, duplicates and unique values. After that we checked that some values have been placed into the wrong columns. So, we shift the values into the respective rows and placed them in the right columns. Another observation is that few field have wrong information so we have corrected with proper values. After that we replaced the null values of ‘Year of Publication’ by mode, ‘Book Author’ and ‘Publisher’ column by ‘Not Mentioned’.
  2. **Users Dataset:** Most of the users are from the age group 25-50

It is highly unlikely to have users under the age of 4 and above 100.The peaks near 0 and 100 in the kdeplot indicates that there are some outlier values in the 'Age' column. It is highly unlikely to have users of age above 100 and below in this case. so, we replace these values with np.nan.

* 1. **Ratings Dataset:** The dataset ratings doesn’t need any pre-processing.

we extracted the clean data in the form of csv file in-order to show the insights data by using pyspark.sql:

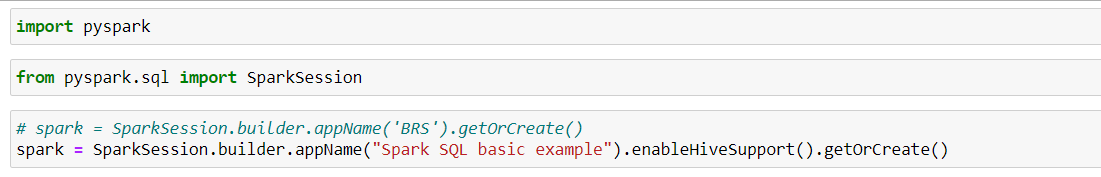


Figure :6



Figure :7

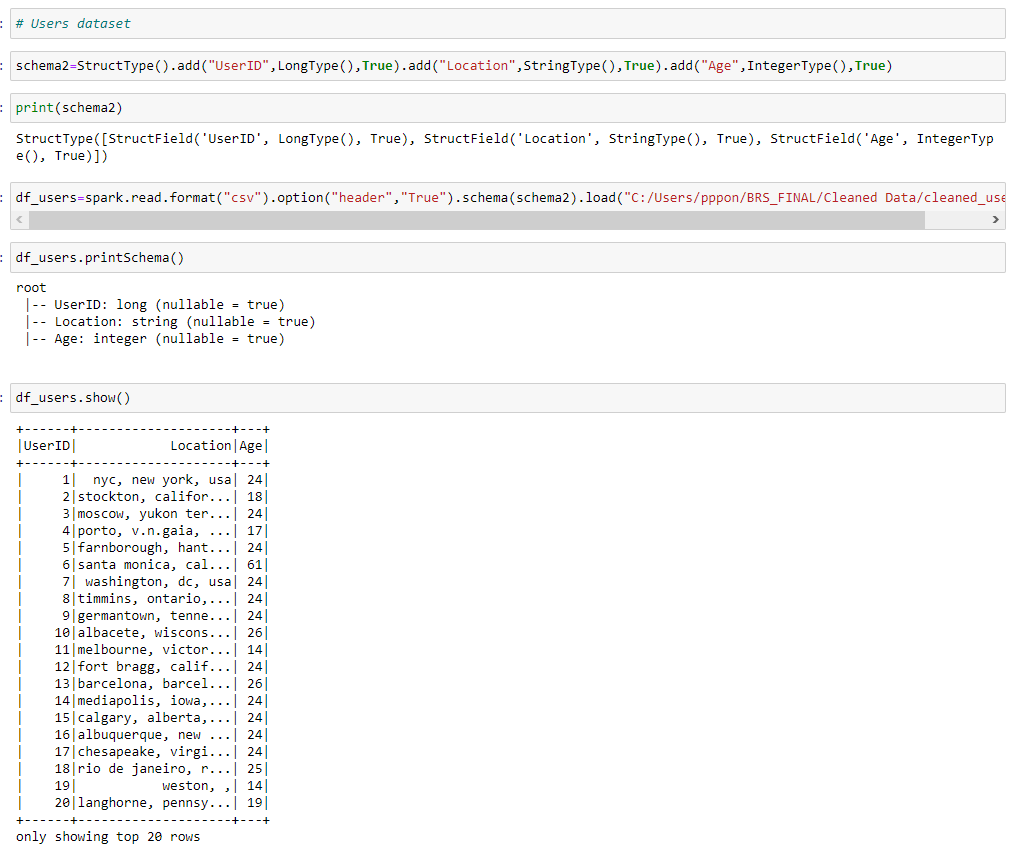


Figure :8

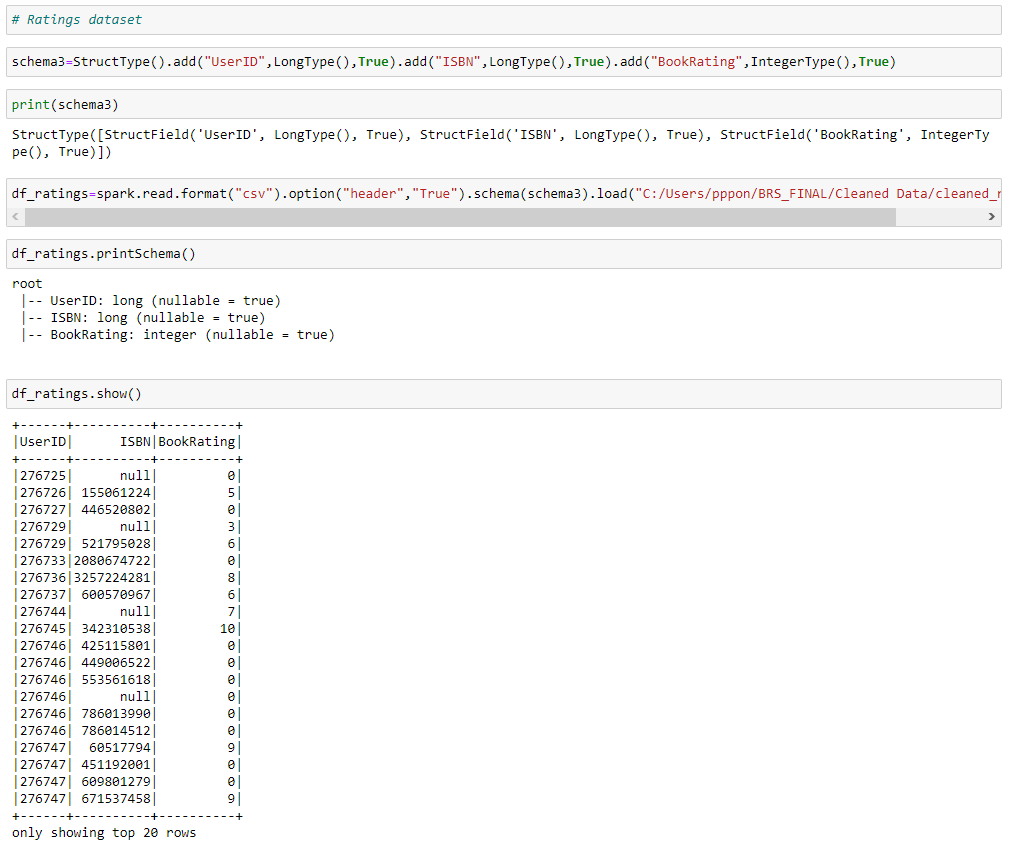


Figure :9

1. **Data Visualization:**

After our data is cleaned, we are ready for visualizing our data to gain some valuable insights as well as present our project.

* 1. **Book Dataset:**

**Analysis no. 1 - Author with highest no. of books**

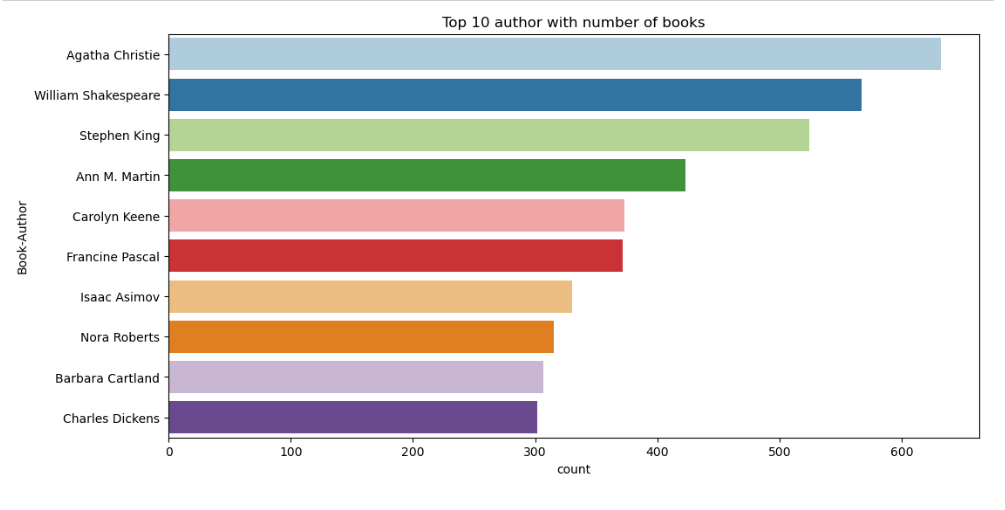


Figure :10

Agatha Christie is leading at top with more than 600 counts as she has the greatest number of books written as compared to others, followed by William Shakespeare.

**Analysis no. 2 - Top publishers**

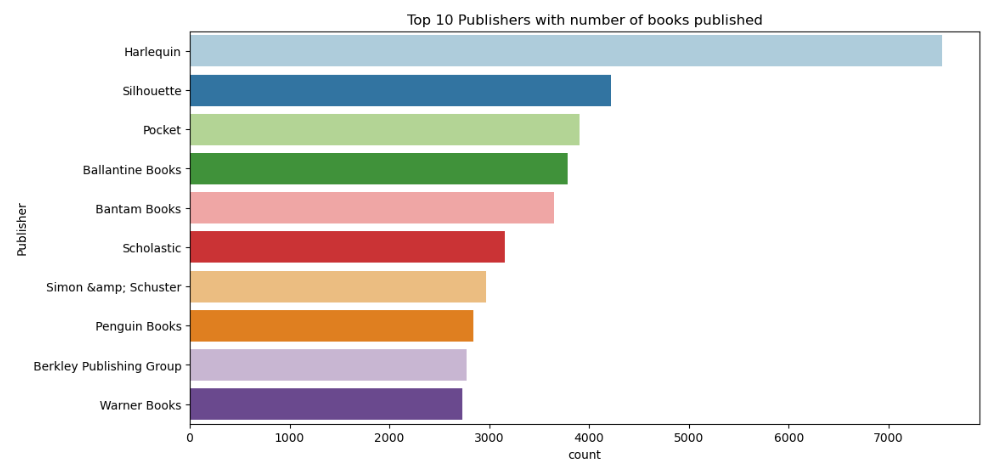


Figure :11

Harlequin has the greatest number of books published, followed by Silhouette.

**Analysis no.3 - Number of Books published on yearly basis**

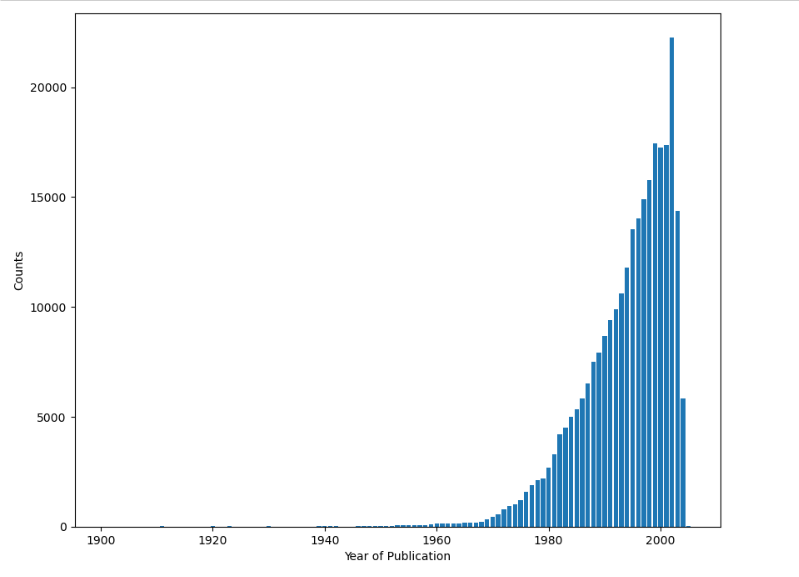


Figure :12

Most no. of books were published in the year 2002. The book industry boomed from 1980's and it was at its peak around the year 2000, it might because people started to understand the importance of books and gradually started to implement productive habits in their life.

* 1. **Users Dataset:**

**Analysis no.4 - Age wise distributions of users**

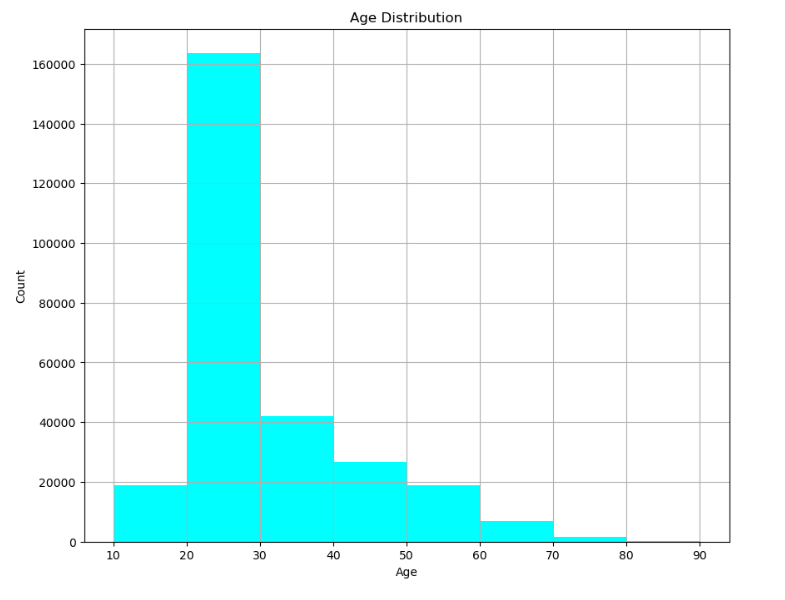
****

Figure :13

Most of the users are 20 - 30 years old which represents the youth.

* 1. **Ratings Dataset:**

**Analysis No. 5 - Rating distribution**

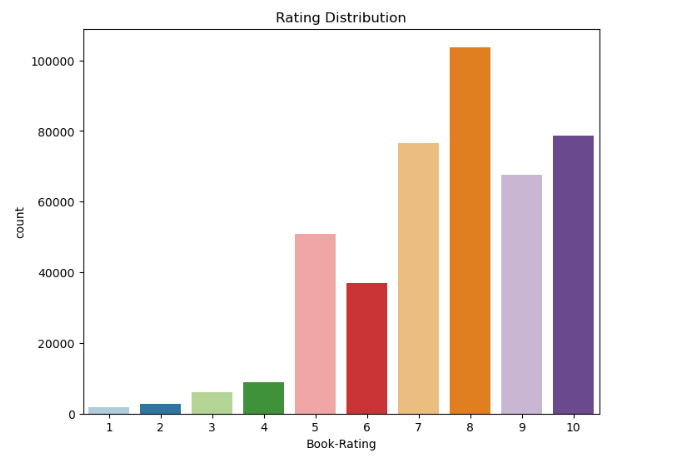
****

Figure :14

Above count plot indicates that higher ratings are more common amongst users and rating 8 has been rated highest number of times.

**Analysis No. 6 - Top 10 highest rated books**

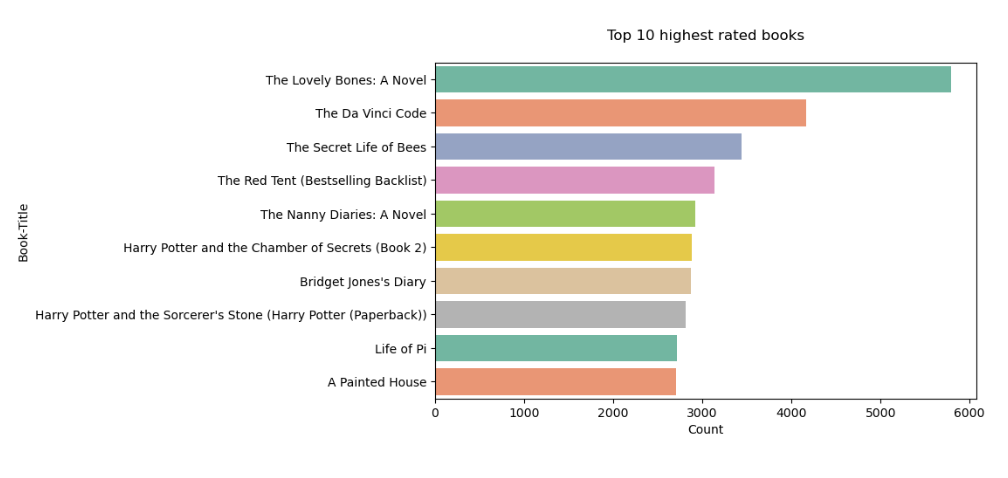
****

Figure :15

The book which has been rated by most number of users is 'The Lovely Bones'.

**Analysis No. 7 - Top 10 highest rated authors**

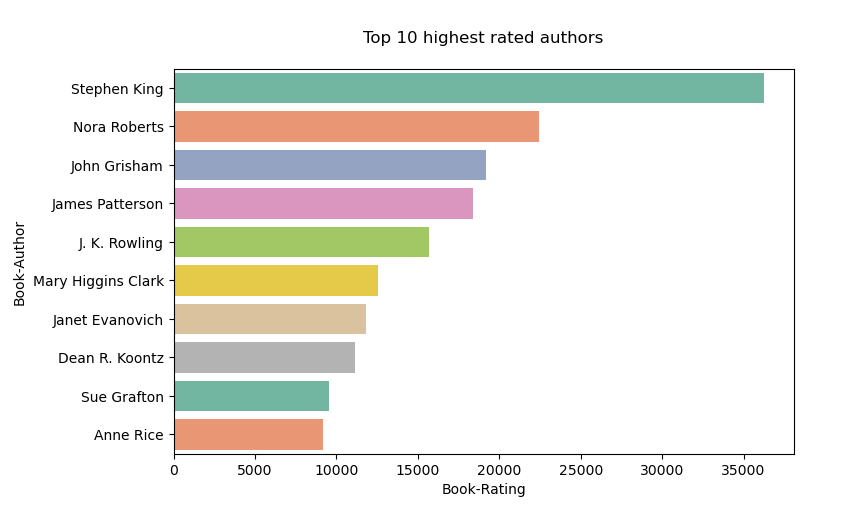
****

Figure :16

Top book author with respect to the number of ratings is Stephen King.

1. **Methodology:**

To build this recommendation system, we have taken three approaches as follows:

1. **Popularity Based Recommendation System.**

With its simplicity, this is the most basic recommendation system which offers generalized recommendation to every user based on their popularity. In a bookstore, if a certain book is popular among its customers & is also critically acclaimed, in the scenario that a new customer walk in & asks for the best, they would be suggested to try that book too. The same is true for movies, shows, music, etc. Whatever is more popular among the general public, is more likely to be recommended to new customers too.

This type of recommendation system makes generalized recommendation not personalized, meaning that this system will not take into account the personal preferences or choices, rather it would tell that this particular thing is liked by most of the users.

* 1. **Using average rating - Top 50 books in whole collection**

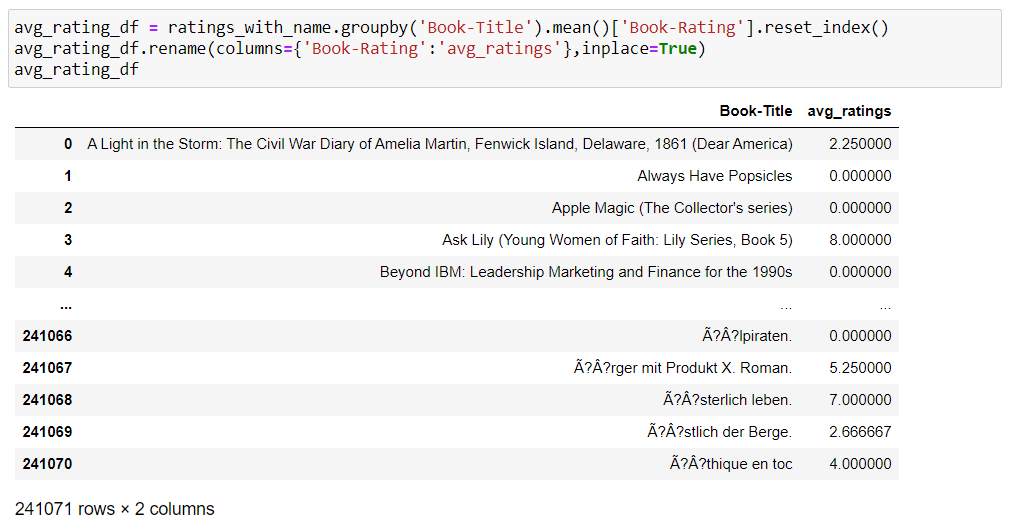
****The average rate of books is displayed below:

Figure :17

The top 50 recommended books are:

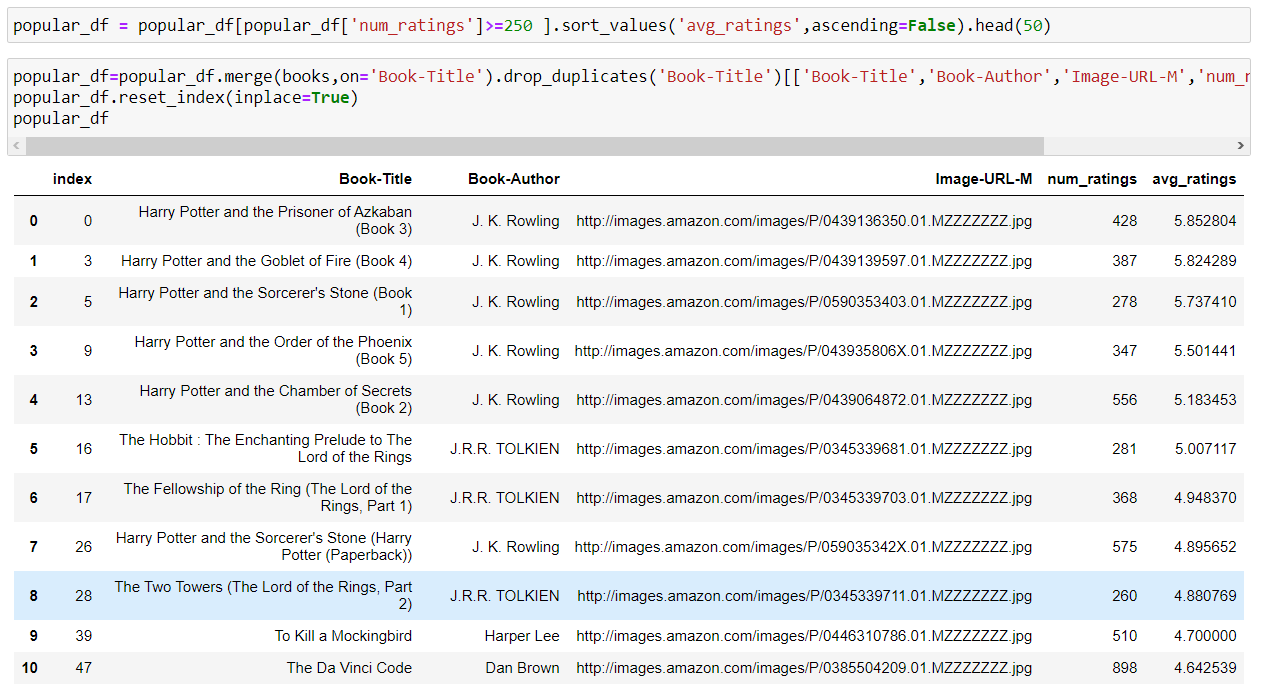


Figure :18

* 1. **Books popular yearly :**

Here, we have shown the books which are popular by yearly.

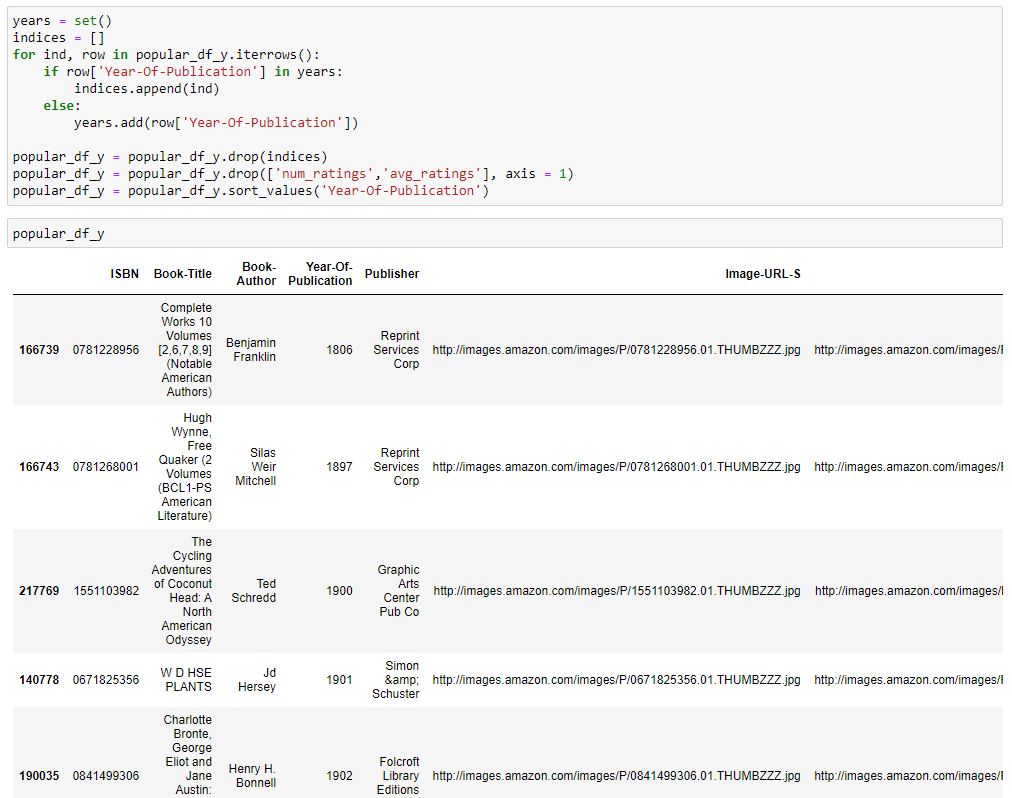
****

Figure :19

1. **Correlation Based Recommendation System**

Correlation coefficients are used to measure how strong a relationship is between two variables. There are several types of correlation coefficient, but the most popular is Pearson’s. A Pearson correlation is a number between -1 and +1 that indicates to which extent 2 variables are linearly related. So in this case, it is the rating for two books.

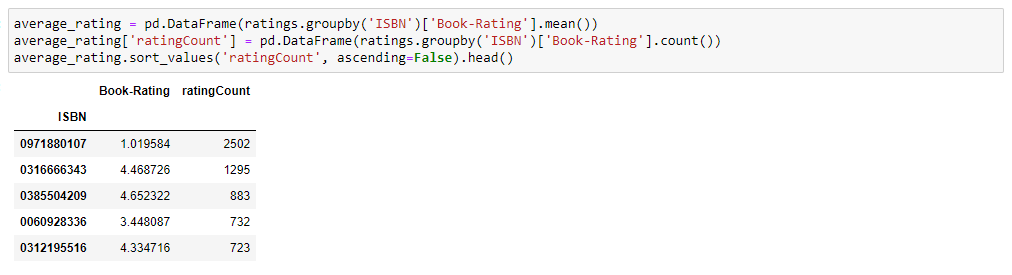
First, the average rating & the number of ratings each book received were found and then sorted based on the rating count in descending order.

Figure :20

**Observations:** The book with the most rating counts isn't necessarily a highly rated book. As seen from the table above, the book with the most rating counts of '2502' only had a rating of '1.019584'. As a result, if recommendations were made solely based on rating counts, it is evident that mistakes would be made.

The ‘ratings’ table is then converted into a 2D matrix. The matrix is sparse since not every user rated every book.

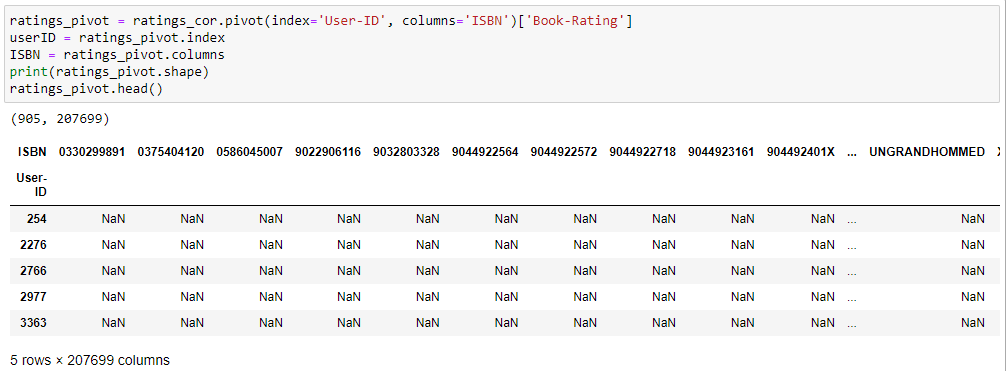


Figure :21

To test the system, the book **' The Lovely Bones: A Novel'** by American writer Alice Sebold.

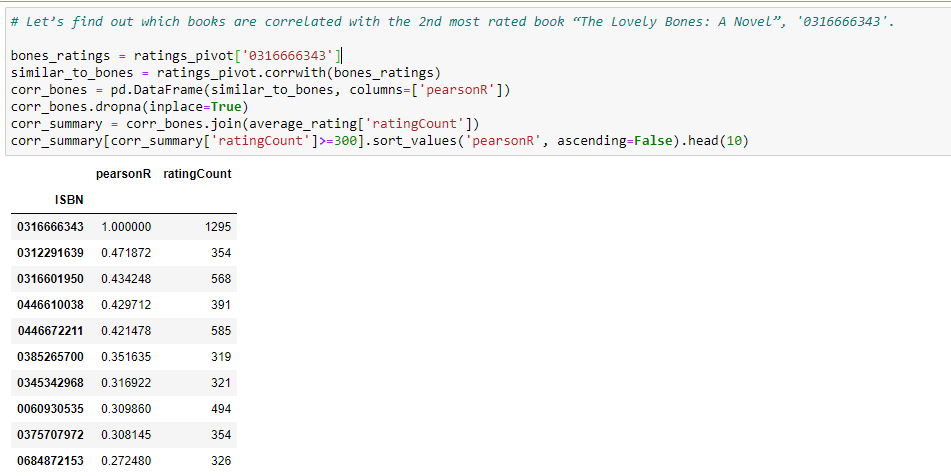
The following are the books correlated with the above-mentioned book: 

Figure :22

The recommended books are:

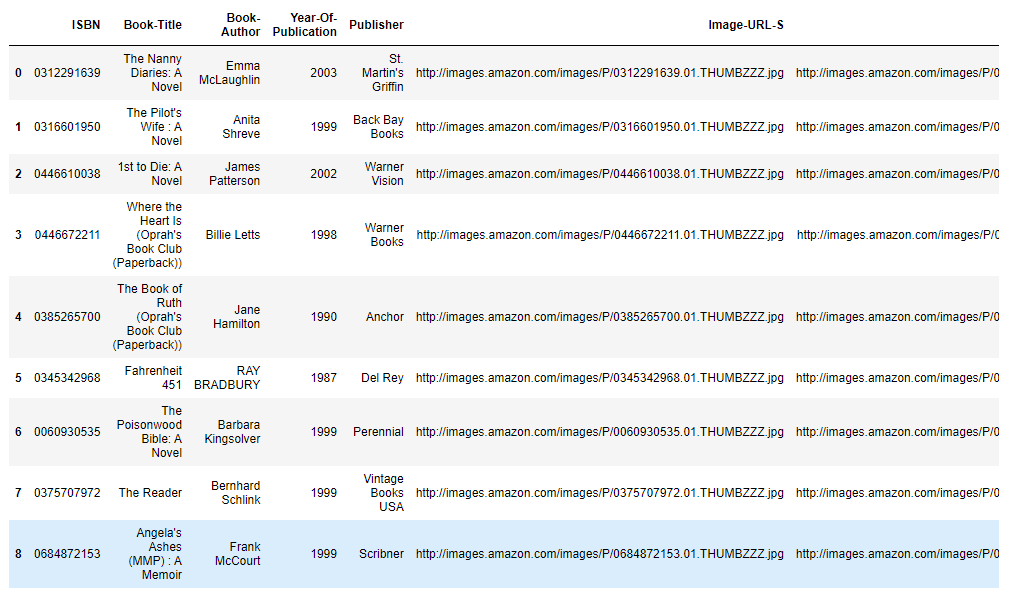


Figure :23

From this we can conclude that the all the books mentioned in Figure :19 does not show the similarity from the book **' The Lovely Bones: A Novel'.** It’s not showing the proper resulting recommended books are correlated with the book which we have tested, we can imply that our correlation-based book recommendation system is not working.

1. **Collaborative Filtering Based Recommendation System:**

In Collaborative Filtering, we tend to find similar users and recommend what similar users like. In this type of recommendation system, we don’t use the features of the item to recommend it, rather we classify the users into the clusters of similar types, and recommend each user according to the preference of its cluster.

1. **Cosine Similarity Based Approach:**

Cosine similarity is a metric used to measure how similar two items are. Mathematically, it measures the cosine of the angle between two vectors projected in a multi-dimensional space. The output value ranges from 0–1. 0 means no similarity, whereas 1 means that both the items are 100% similar.

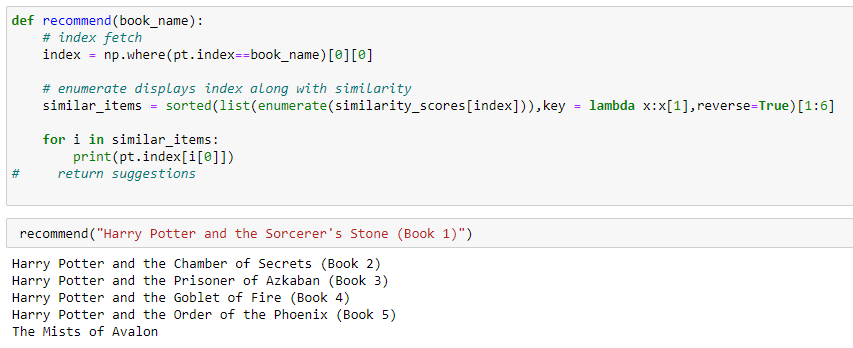


Figure :24

Here, in this code, we have recommended the book ‘Harry Potter and the Sorcerer’s Stone (Book 1)’ and according to this book, we are getting the similar outputs which shows the cosine similarity among the books.

1. **Matrix Factorization:**

Matrix factorization is a class of collaborative filtering algorithms used in recommender systems. Matrix factorization algorithms work by decomposing the user-item interaction matrix into the product of two lower dimensionality rectangular matrices.

Singular value decomposition (SVD) is used here. The users' rating table is converted into a utility matrix & the missing values are filled with zeros,

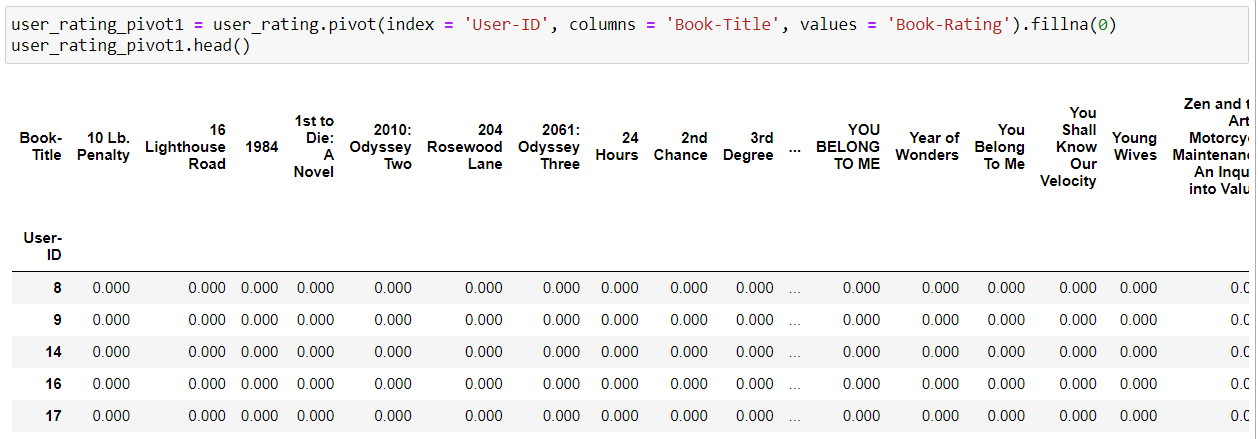


Figure :25

This utility matrix is transposed in order for the 'book title' & 'userIDs' to become rows & columns respectively. After using TruncatedSVD to decompose it, it is fitted into the model for further dimensionality reduction.

Pearson's R correlation coefficient is calculated for every book pair in the final matrix. :-



Figure :26

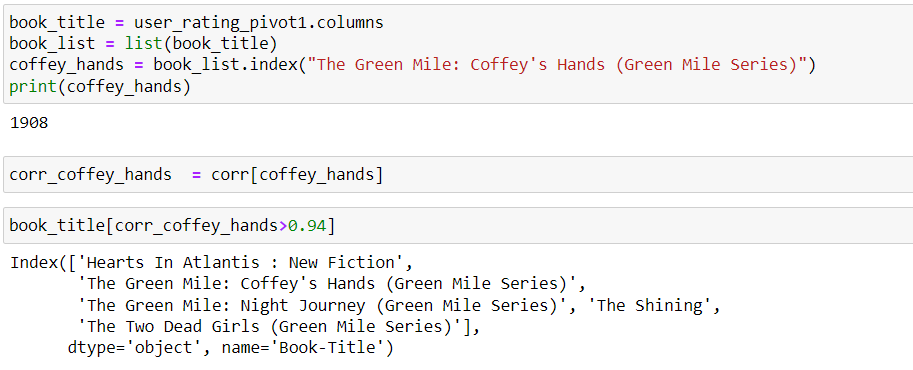


Figure :27

Clearly, as seen from the Figure :16 codes, the recommendation system using matrix factorization also recommended the same books as the system using cosine similarity-based approach.

1. **K-Nearest Neighbours (k-NN) :**

K-Nearest Neighbours (KNN) algorithm is a type of supervised ML algorithm which can be used for both classification as well as regression predictive problems. KNN algorithm assumes the similarity between the new case/data and available cases and puts the new case into the category that is most similar to the available categories.

Using this algorithm, clusters of similar users based on common book ratings can be found and predictions can be made using the average rating of the top-k nearest neighbours.

The table is converted into a 2D matrix & the missing values are filled with zeroes since the distances between rating vectors will be calculated. The values of the matrix data frame are then transformed into a scipy sparse matrix for more efficiency calculations

The algorithm used to compute the nearest neighbours is 'brute' & the metric is 'cosine' so that the algorithm will calculate the cosine similarity between rating vectors.



Figure :28

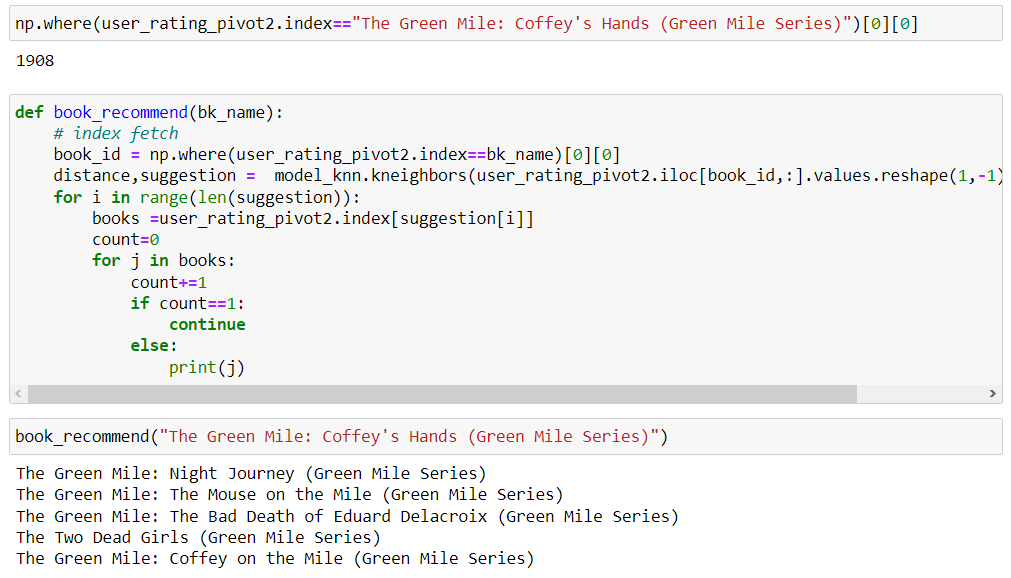
****

Figure :29

Clearly, as seen from the Figure :18 codes, the recommendation system using k-Nearest Neighbour (k-NN) algorithm also recommended the same books as the system using cosine similarity-based approach and matrix factorization method.

**7. Future Scope**

The book recommendation system project has immense potential for future development and enhancements. One possible avenue for improvement is incorporating more sophisticated machine learning algorithms that can analyse user behaviour and preferences more accurately. Another area for expansion is integrating the system with social media platforms to provide users with recommendations based on their social network's reading habits. Additionally, the system could be adapted to recommend other types of media, such as movies, music, and podcasts, increasing its reach and usefulness.

**8**. **Conclusion**

In conclusion, the Bookify: The Ultimate Book Recommendation Application With Data-Driven Intelligence project aimed to develop a collaborative-based and popularity-based recommendation system that suggests books to users based on their reading history and preferences. The project used a dataset of books, ratings, and users, which was pre-processed and transformed using data cleaning, encoding, and feature extraction techniques.

The collaborative-based recommendation system used user-based and item-based collaborative filtering techniques to generate recommendations, while the popularity-based recommendation system used book popularity and rating averages to suggest popular books.

In conclusion, the Bookify project demonstrates the importance of using collaborative-based and popularity-based recommendation systems to provide more accurate and personalized recommendations to users. The project contributes to the field of recommender systems and provides insights into the design and evaluation of book recommendation systems. The limitations and future work of the project include improving the scalability and robustness of the system and incorporating more advanced techniques such as deep learning and reinforcement learning for recommendation.