Book Recommender System

Data Miners

Problem Definition:

A Recommender system is an information filtering software that suggests similar items to a User based on their earlier 'ratings' and 'preferences'. Book Recommendation System is being used by Amazon, Barnes and Noble, Flipkart, Goodreads, etc. to recommend books the user would be tempted to buy as they are matched with user choices. To offer products and services to consumers, three types of machine learning recommendation systems based on filtering are utilized.

- Content Filtering
- Collaborative Filtering
- Hybrid Filtering

We are building a book recommender system using a Collaborative based model. Collaborative filtering models can recommend an item to user A based on the interests of a similar user B.
A book recommendation system recommends similar books to the reader based on their interest. For instance, when a user visits the homepage, the system should recommend books based on both similarities to books the user has liked in the past and books that similar users liked. Every time a user searches for a new book, the input data is stored, and a user profile is built to get a better idea of what that user's common search terms and interests are. Once the user profile is set up, the user's past behavior is used, and implement a collaborative filtering system to recommend users with similar profiles, and similar books.

Project Motivation:

To recommend the top 5 books KNN model using Euclidean distance will be used. Planning to create an interactive web page using HTML, CSS, Bootstrap, and Flask, and deploy using Heroku.

We also built a chatbot based on Natural Language Processing (NLP) and use a recommendation engine to suggest similar books. The bot can also recommend various genres of books like Horror, Mystery, Thriller, and Fiction types - Science Fiction, Historical, Fantasy, and Realistic Fiction. It can even detect vague replies and reply accordingly.

This chatbot will be hosted on a web application. HTML, CSS, Bootstrap, and Flask will be used to take our python code and create a web application with it. Finally, it is deployed using Heroku.

Data/Dataset Description:

We are using the 3 datasets from Kaggle.

1. Books.csv

   The ISBN is used to identify books. Additionally, certain content-based information (Book-Title, Book-Author, Year-Of-Publication, Publisher) is provided, which was obtained from Amazon Web Services. URLs to cover photos are also provided.

2. Users.csv

   Contains the users and user IDs (User-ID) have been anonymized and mapped to integers. Demographic data like age and location are provided.

3. Ratings.csv

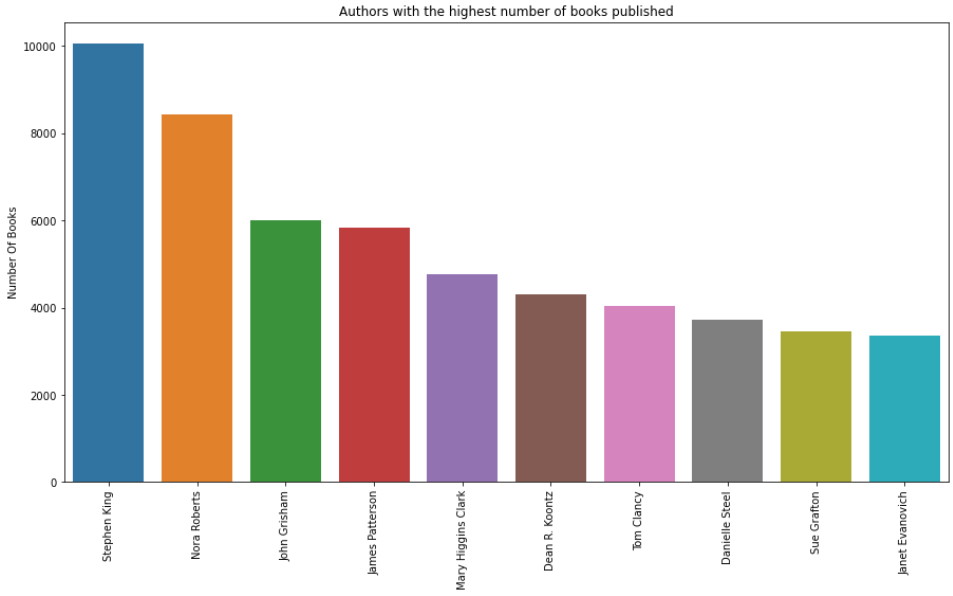
   Contains the book rating information. Ratings are either explicit, expressed on a scale from 1-10, or implicit, expressed by 0.
   Planning to create an interactive web page using HTML, CSS, Bootstrap, and Flask and deploy using Heroku.

Data Cleaning:

There aren't many nulls in the Author, Publisher, and image URL columns of the Books data, according to our search for null values. We dealt with those by substituting "others" or replacing them with the appropriate values. There weren't any duplicates. We noticed that the data in three rows were not correctly aligned, for example, the publisher's name was beneath the year and the book author's year of publication was under the year. The values in those rows were replaced as our solution. The age column in the Users database has many null entries, as was discovered. Along with the ages >95 and 5, we changed those numbers to the mean. The three datasets were then combined into a single data frame**.**

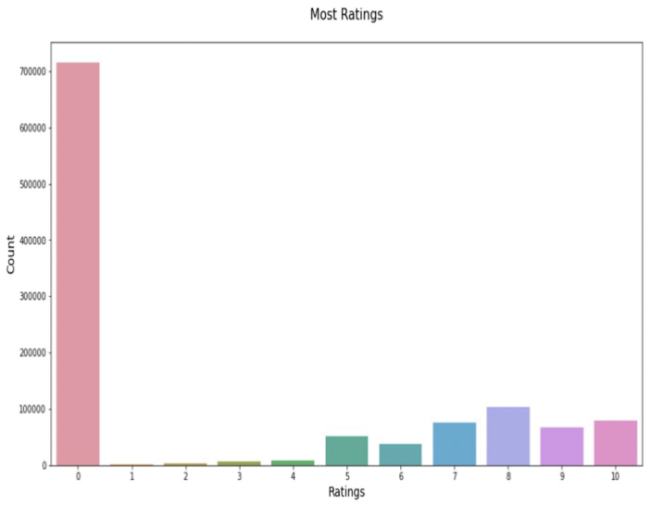
Exploratory Data Analysis:

1. Authors with the highest number of books published.



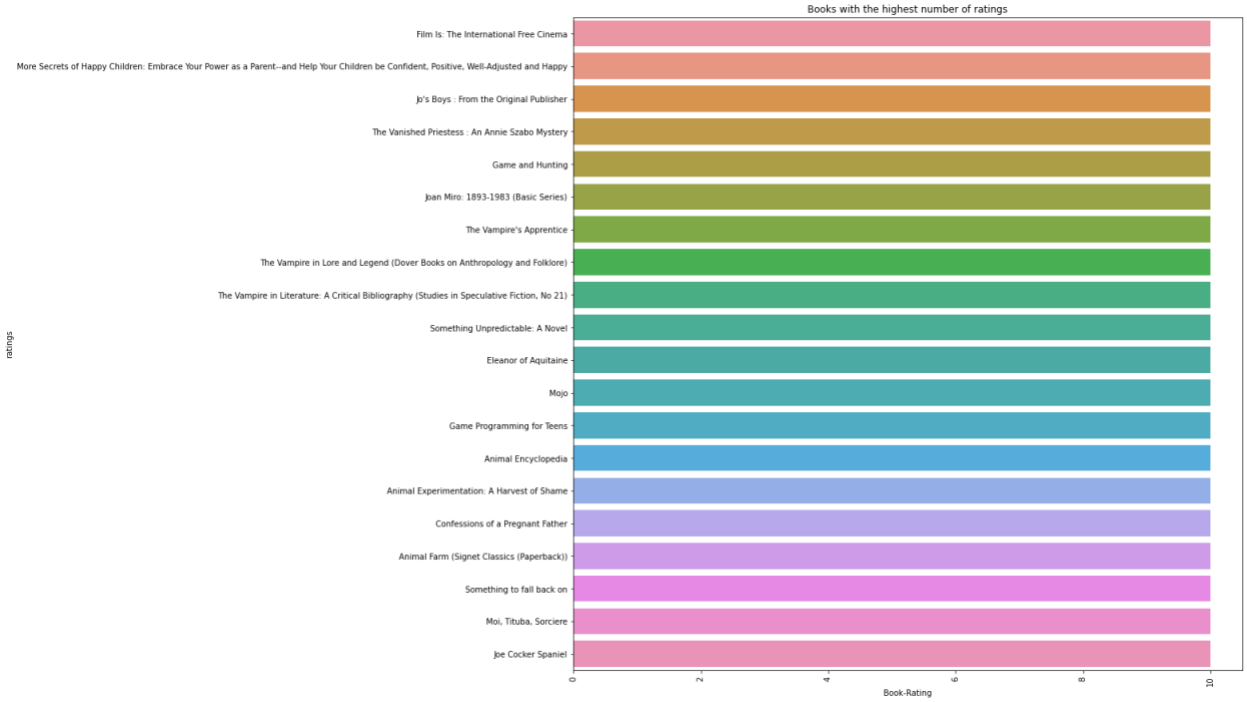
This Bar graph represents the Author who has published the highest number of books and we can observe that Agatha Christie published the highest number of books around 650 followed by William Shakespeare around 570 books. Agatha Christie and Stephen King wrote Crime genre novels maybe we can conclude that crime fiction novels were the highest published out of all genres.

2. Most Ratings.



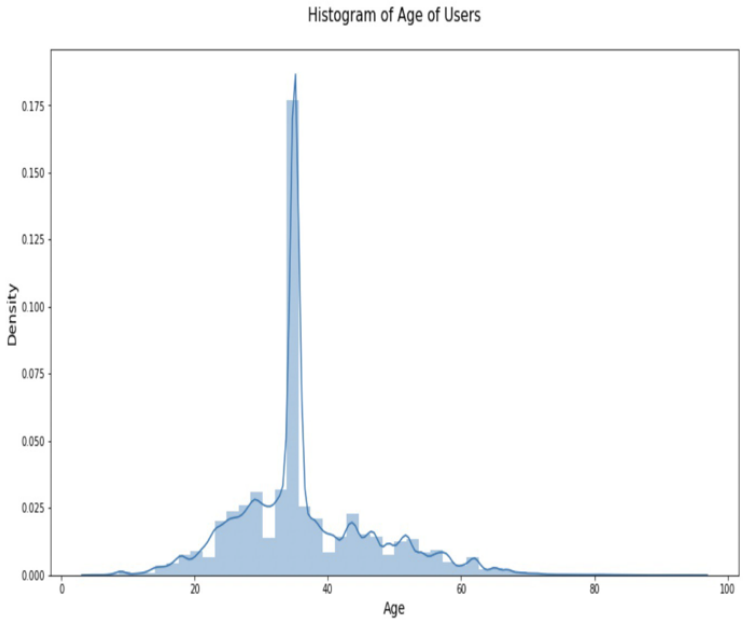
This graph represents the ratings given by the users. Most of the books were not rated by the users and ‘8’ was the most common rating given by more than 100k users. We can also observe that most of the rated books have a rating above 5 and the poor rated books were negligible

3. Books with the highest number of ratings



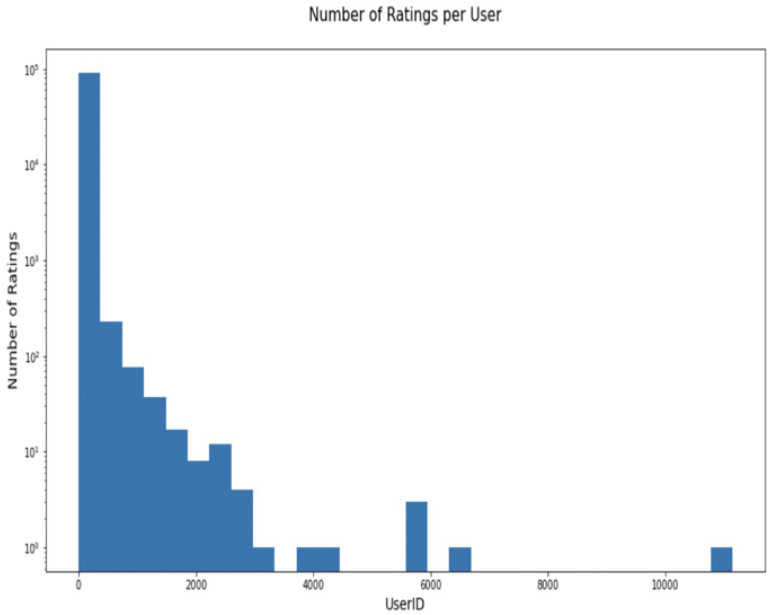
This represents the books with the highest number of ratings. These show a few of the top-rated books like Film Is: The International Free Cinema, More Secrets of happy children, Jo's boys.

4. Histogram of Age of Users



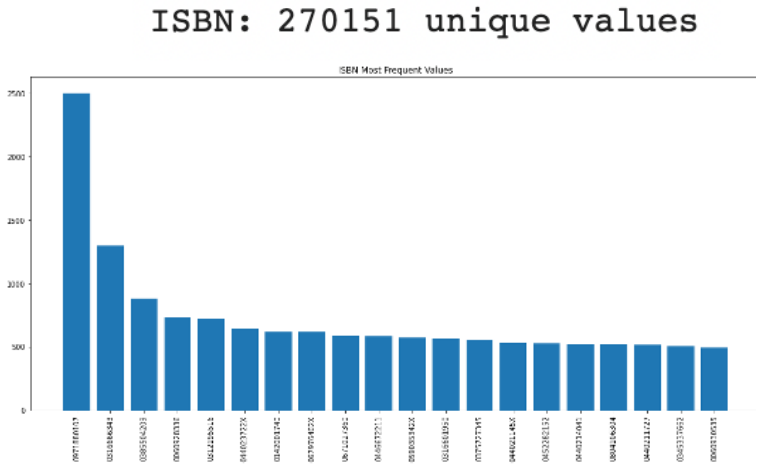
This histogram represents the age of users. We can observe that most users' age was in the range of 20 to 60. We can also observe that users around age 30 show interest in reading books compared to other age groups.

5. Number of Ratings per User



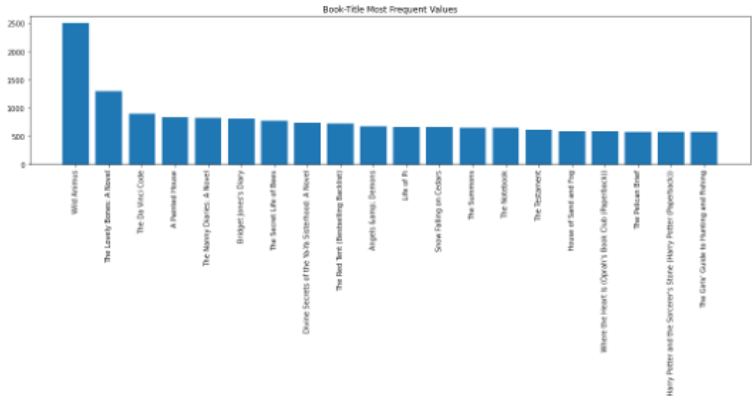
The highest number of ratings was given by the User ID between 0 and 3000. User ID's between 6400 and 10400 didn't rate any of the books or negligible. We can conclude that around 4000 users were giving ratings for the books they read.

6. ISBN Most Frequent Values.



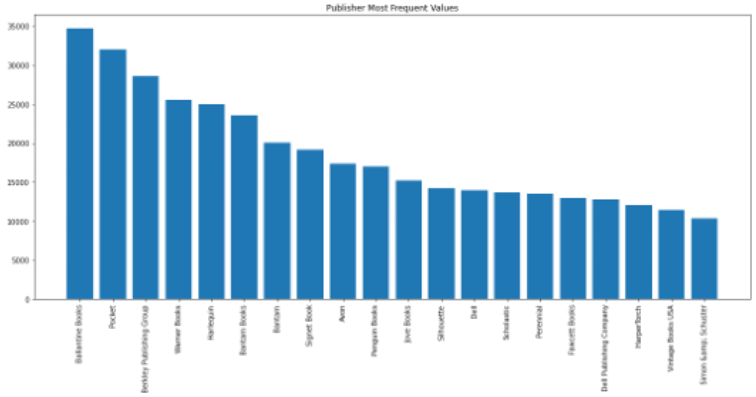
7. Book-Author Most Frequent Values.

```
Book-Author: 101587 unique values
```

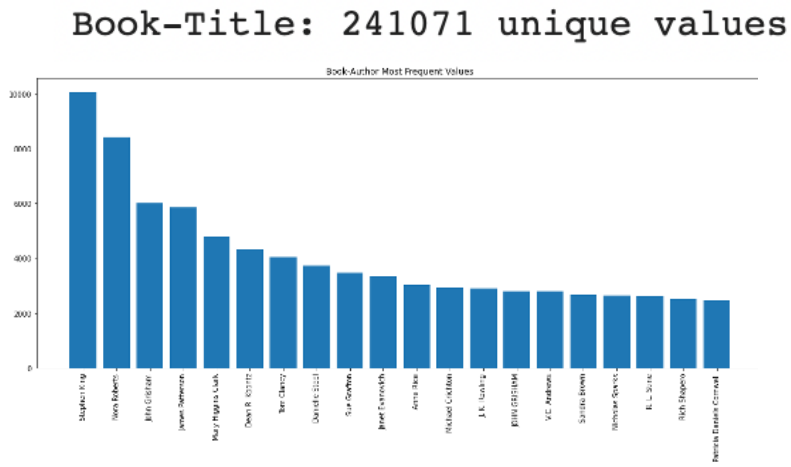


8. Publisher Most Frequent Values.

```
Publisher: 16727 unique values
```



9. Book-Title Most Frequent Values.



From this graph, we can see the topmost 20 frequent values of ISBN, Book-Author, Publisher and book title. The most frequent book was with ISBN 0971880107 followed by 0316666343 and the most popular author was Stephen King followed by Nora Roberts. The most frequent publisher was Ballantine Books which published 34724 books. Wild Animus was the topmost book followed by The Lovely Bones.

Recommender System:

We built our Recommender system using the Collaborative filtering method. Collaborative filtering is a popular method used for RS systems that are based on information and references obtained from the users. User rating is the most important variable for our project because it is widely used for reference in establishing our Book recommender System. The Collaborative filtering method focuses on finding users who have given similar ratings to the same books, thus creating a link between the users, to whom we will be suggesting books for future purposes. This explains that the collaborative filtering method relies on user behavior to make recommendations.

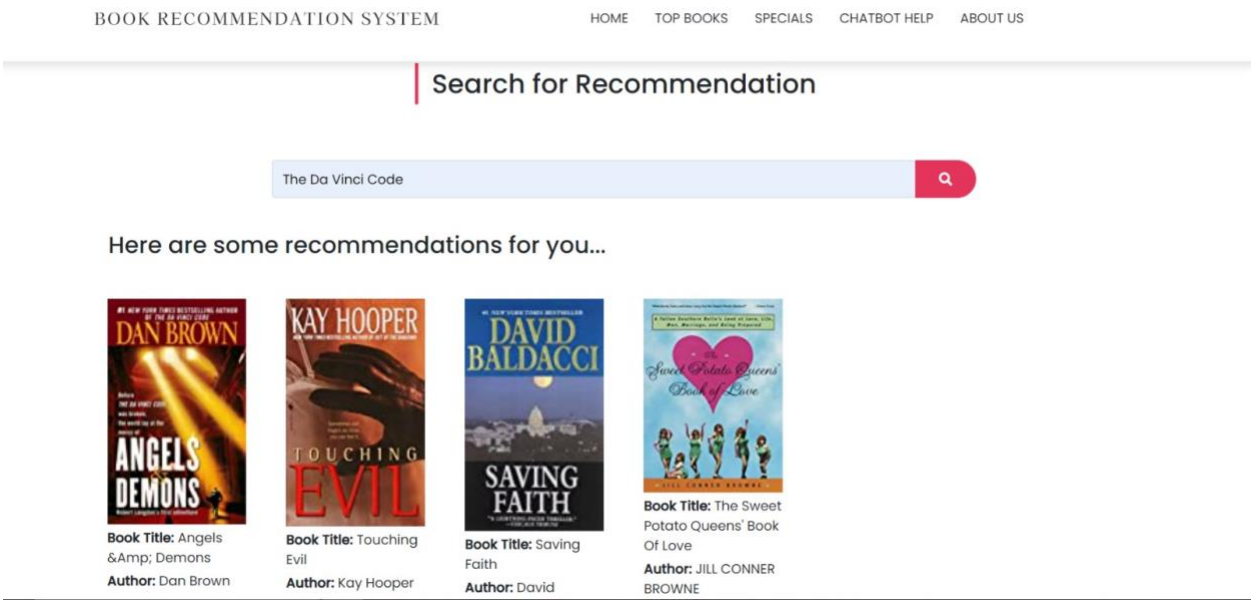
Cosine similarity measures the similarity between two vectors of an inner product space. It is measured by the cosine of the angle between two vectors and determines whether two vectors are pointing in roughly the same direction. We used cosine similarity to measure the nearest neighbor (similar users). We used KNN algorithm to find the similar users. KNN or K-Nearest Neighbors is a simple machine learning algorithm, which tries to assign labels to new data points based on the metrics/variables we chose. The assumption we make while choosing this algorithm is that similar data points always have similar metrics. In other words, when two data points are similar, then their position in the hyperplane is also very close to each other. In our model, we are going to find users/books which are similar, so that we can recommend them to the user based on the input we are going to feed into the recommender system.

Final Output:

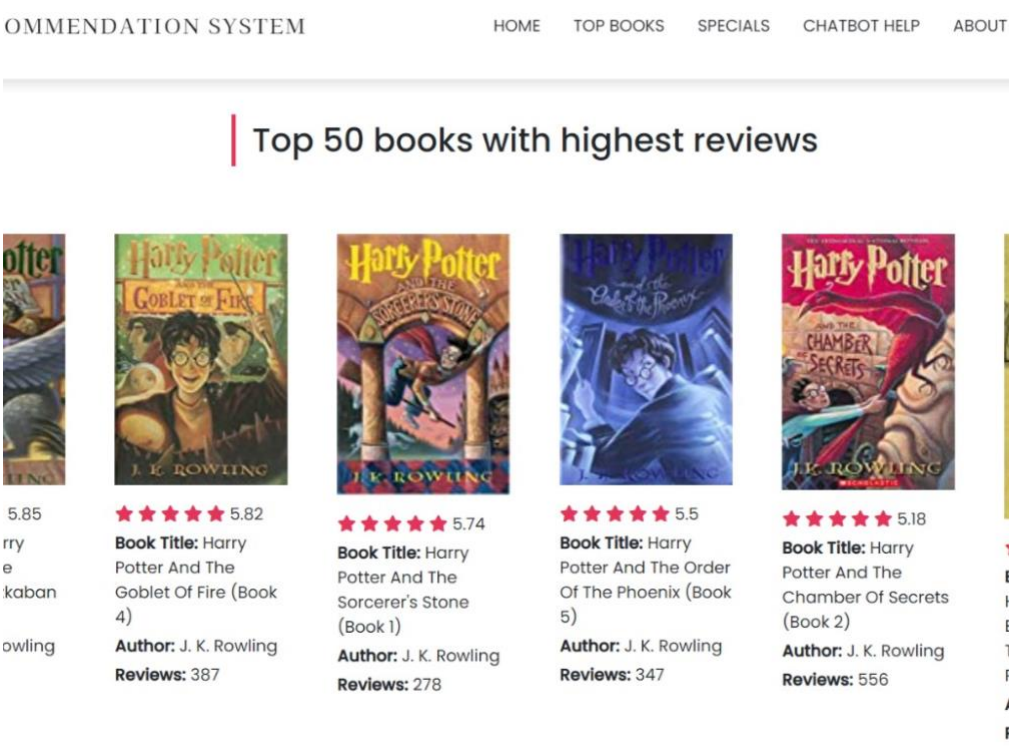
The output of the recommender system is provided. When a book title is entered, the recommender system suggests 5 books that are similar out of 243K books. The recommendation system will suggest only those books which have at least 50 ratings and by those users only who has rated at least 200 times.

```
n [45]: recommender_system('1984')

Animal Farm
The Handmaid's Tale
Brave New World
The Vampire Lestat (Vampire Chronicles, Book II)
The Hours : A Novel

n [46]: recommender_system('The Handmaid\'s Tale')

Bastard Out of Carolina
A Civil Action
1984
The Kitchen God's Wife
The Red Tent (Bestselling Backlist)
```

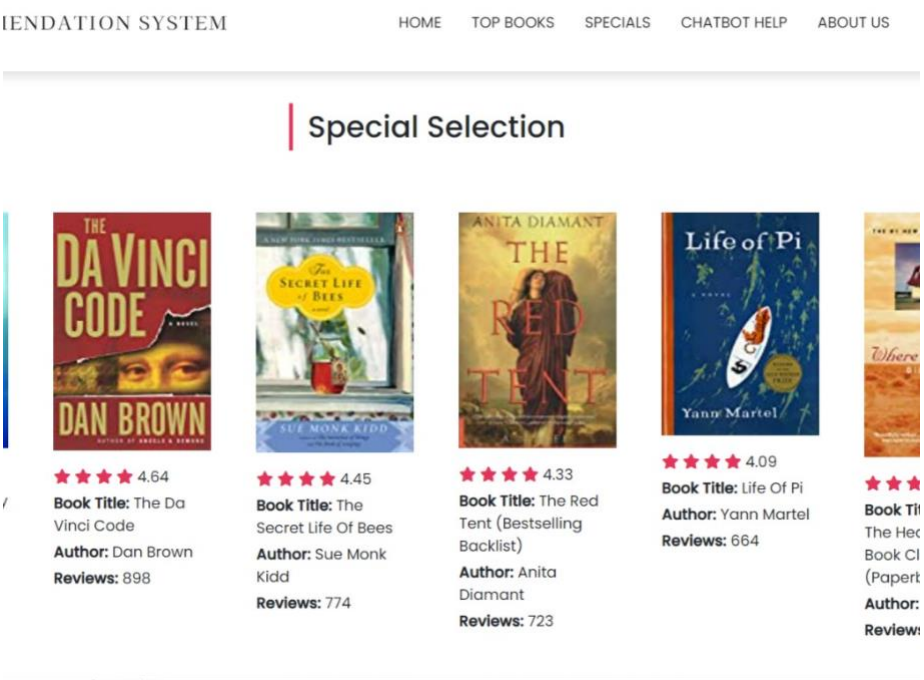
We developed a web application by using HTML, CSS, JavaScript, Bootstrap, and flask as front-end and back-end development components. Around 243K books were encountered in the data set.



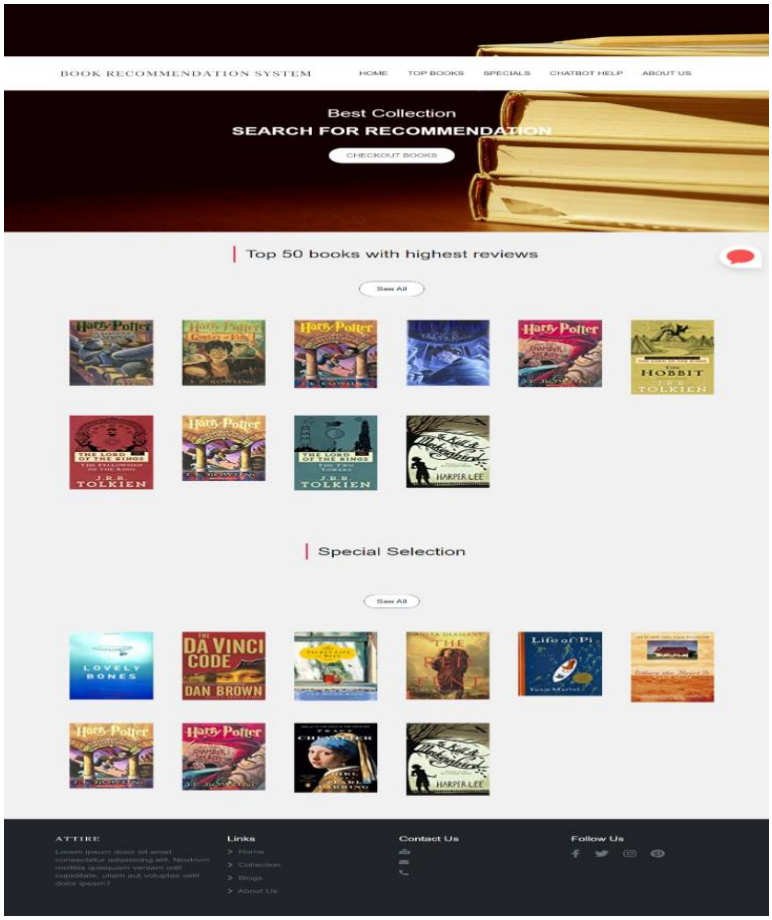
This image represents the Search Recommendation where the User can type a title/genre/author and the books similar to the keyword will pop up.



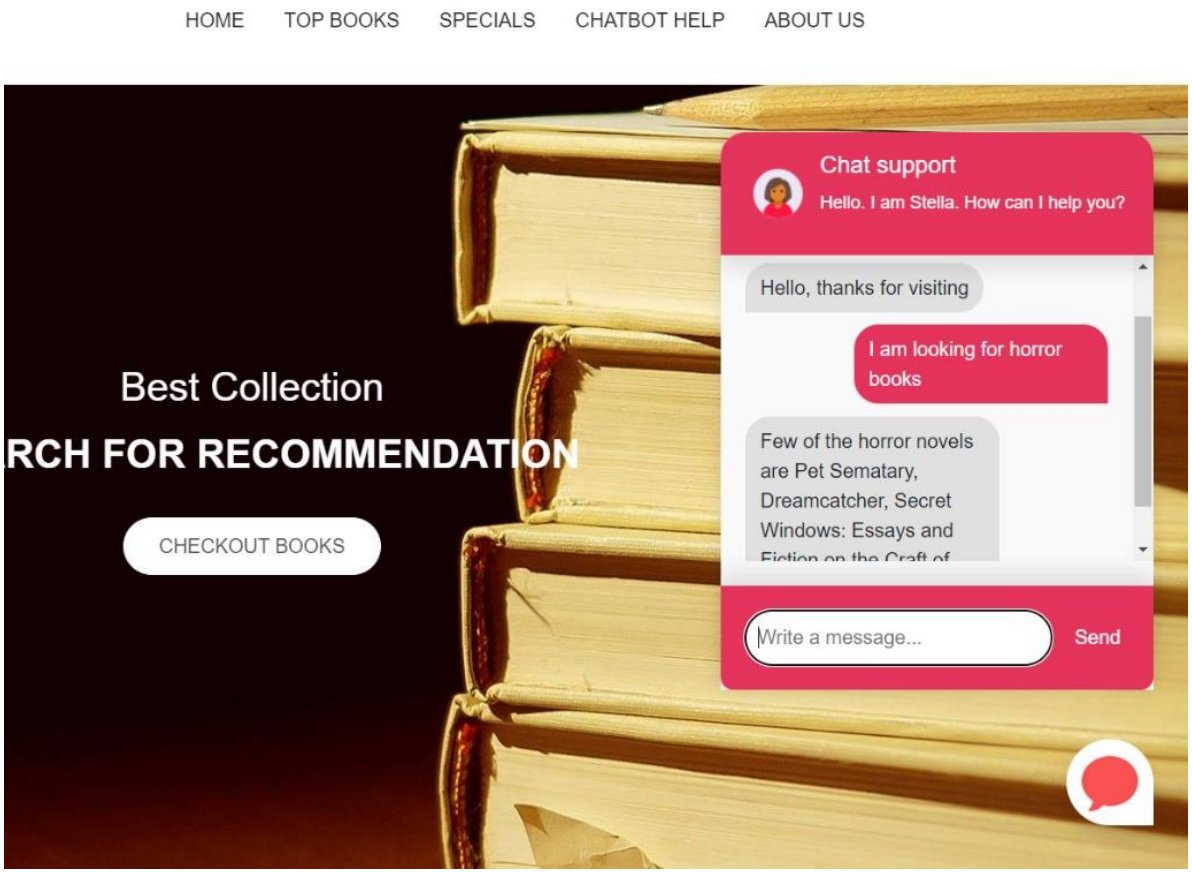
This image represents the Top 50 books recommended based on the ratings.



This section represents Top 50 books recommended on the basis of reviews.



This is how our Book recommender system web application looks like.



We have developed a user interactive chatbot that will assist users for searching books he/she is looking for.

References:

1. https://www.kaggle.com/datasets/arashnic/book-recommendation-dataset
2. https://towardsdatascience.com/my-journey-to-building-book-recommendation-system-5ec959c41847
3. https://www.analyticsvidhya.com/blog/2021/06/build-book-recommendation-system-unsupervised-learning-project/#:~:text=A%20book%20recommendation%20system%20is,library%2C%20good%20Read's%2C%20etc