# Semantic Analysis and recommendation system for Book

Aishwarya Sinhasane[1]* , Anuj Mahajan[2]*, Shashwati Diware[3]*,Shubham Jambhale[4]*

**Abstract**
For both users and publishers/authors, the issue of sentiment analysis of book reviews and recommendations for books utilizing Amazon dataset is crucial. The objective of book review sentiment analysis is to automatically categorize a book review's sentiment as positive, negative, or neutral. Machine learning algorithms that have been trained on labeled review datasets are frequently used to do this. The task of book recommendation is to make book recommendations to users based on their preferences and previous actions. Sentiment analysis of book reviews can be helpful for book recommendations in a number of ways, including helping readers discover books they are likely to enjoy and assisting publishers and authors in understanding reader reactions to their works. Book recommendation systems can give users a more personalized and relevant experience, boost user engagement and satisfaction, and help users discover books that they might not have found otherwise by assessing the sentiment of book reviews and using this information to create better recommendations. Moreover, book recommendation systems can assist publishers and authors in more efficiently promoting their books by recommending books to readers based on their tastes and behavior. This results in higher sales, more money, and the ability to reach new audiences. Also, by examining the tone of user reviews and other user data, book recommendation algorithms can better understand user preferences and behavior. As a result, publishers and authors can design more specialized marketing strategies and new consumer-focused goods and services. Improving personalization, customer satisfaction, competitive advantage, innovation, and social impact can all be facilitated by finding a solution to the book review sentiment analysis and book suggestion conundrum.

**Keywords**
sentiment analysis — recommendation — book reviews

[1] *Luddy School of Informatics, Computing, and Engineering, Indiana University, Bloomington, IN, USA*

## Contents



## 1. Problem and Data Description

**Problem Description**

- The problem of sentiment analysis of book reviews and recommendations for books using the Amazon dataset can be utilized to deliver more tailored and relevant book recommendations based on individual readers' likes and behavior.
- Book review sentiment analysis is the task of automatically classifying the sentiment of a book review as positive, negative, or neutral. This is often accomplished using machine learning algorithms that have been trained on labeled datasets of reviews. The algorithm's input is the review text, and its output is the sentiment label.
- Book recommendation is the task of suggesting books to users based on their preferences and past behavior. This can be done using a variety of techniques, including collaborative filtering, content-based filtering, and hybrid models.
- Book review sentiment analysis can be useful for book recommendation in several ways. For example, the sentiment of a user's review can be used as a feature in a content-based filtering model, helping the algorithm understand the user's preferences more accurately. Sentiment analysis can also be used to filter out reviews that are not relevant to the user's preferences, such as

reviews of books in genres that the user does not like.

**Data Description**

- The data we obtain has two csv which are linked with each other using book titles. The two csv are Book Details and Book Reviews.
- Book Details consist of attributes like Title, Description, Author, Publisher, Published Date, Categories, and Rating Count.
- Majority of data in the dataset is categorical in nature. This CSV gives us insight into the book we are reviewing. It tells us who published the book at what time and who are the authors of the book. It also conveys which category the book belongs to and how many ratings it has received.
- The other CSV which is Book Rating is mostly focused on reviews received by different books. It has short summary of the review to analyze the review in short phrases which gives us important information from the overall review. This data is the main source for our sentimental analysis. It also shows helpful review ratings.

## 2. Data Preprocessing & Exploratory Data Analysis

### 2.1 Handling Missing Values

**Handling Missing Values and Data Preprocessing**

```
1  books_rating.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3000000 entries, 0 to 2999999
Data columns (total 10 columns):
 #   Column              Dtype
---  ------              -----
 0   Id                  object
 1   Title               object
 2   Price               float64
 3   User_id             object
 4   profileName         object
 5   review/helpfulness  object
 6   review/score        float64
 7   review/time         int64
 8   review/summary      object
 9   review/text         object
```

**Figure 1.** Book Rating Information

- Book Rating csv consists of above columns and info
- It consists of only 3 numerical columns which are Id, review/score, and review/time. Other all the columns are categorical.

```
1 nan_cols = data_sem.columns[data_sem.isna().any()].tolist()
2 print(nan_cols)
3 data_sem.isna().sum()
```

```
['Title', 'Price', 'User_id', 'profileName', 'review/summary', 'review/text', 'description', 'authors', 'image', 'previewLink',
'publisher', 'publishedDate', 'infoLink', 'categories', 'ratingsCount']

Id                        0
Title                   208
Price                2510854
User_id               558559
profileName           558658
review/helpfulness         0
review/score               0
review/time                0
review/summary            38
review/text                8
description           638314
authors               389373
image                 538679
previewLink           329541
publisher             780240
publishedDate         353315
infoLink              329541
categories            549679
ratingsCount         1357238
```

**Figure 2.** Missing values in dataset

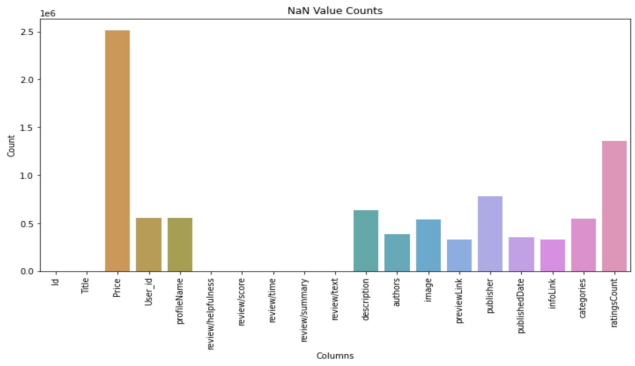


**Figure 3.** Missing values in dataset using histogram

- We have merged both the datasets (books_data and books_rating) based on Title and above are all the columns we have obtained.
- Fig 3 shows all the missing values in the merged dataset.
- It consists of multiple columns for which we have missing values. Price has the most missing values followed by RatingsCount.
- Fig 4 shows the missing values count using a histogram.
- As part of data pre-processing we have also removed the columns which doesn't have much impact on semantic analysis and which are irrelevant along with a huge amount of null values. Those columns are : [Price, Description, and publishedDate]
- One common approach to handling missing values issue is to replace any missing or invalid values with a standard value, such as "unknown". This approach can be particularly useful when dealing with categorical data, such as categories, authors, image, and publisher which are essential for recommendation tasks.

**Categories**

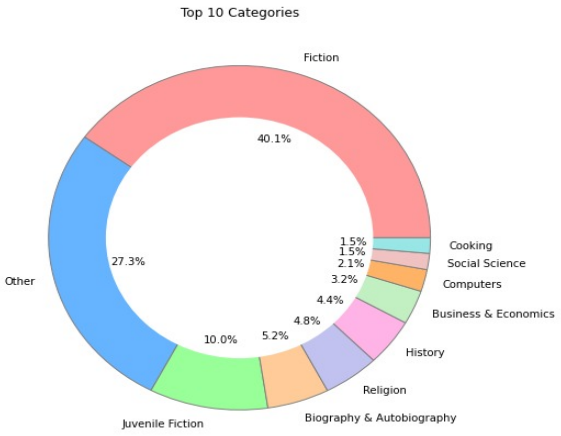


**Figure 4.** Different Categories distribution

- Data has a lot of empty values in categories. As part of data preprocessing, we have replaced the empty values in the categories column using 'Other' to make it more meaningful
- Fig 5 shows the top 10 different categories of books along with 'others'.
- From the data we can see that the 'Fiction' category has the maximum frequency followed by 'Other'.

**Authors**

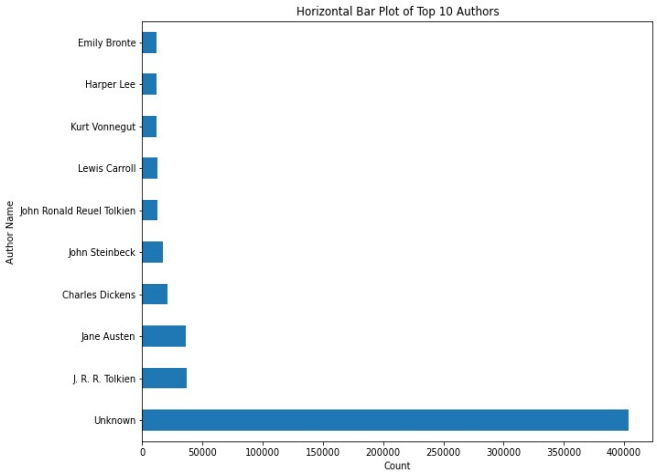


**Figure 5.** Different Categories distribution

- Data has a lot of empty values in Author. As part of data preprocessing, we have replaced the empty values in the categories column using 'Unknown' to make it more meaningful.
- Fig 6 shows the top 10 different Authors.
- From the data we can see that most of the data consist of an 'Unknown' author.

**Pre-Processing for review/text:**

- We have performed preprocessing on text data, specifically reviews of the books, to prepare it for analysis and recommendation tasks. We are using the NLTK a popular library for text processing and analysis in Python.
- The first step in text preprocessing is to remove any URLs that may be present in the input text. This is done using regular expressions (re library) to match and replace any string starting with "http" or "www". URLs are usually irrelevant for the task of book recommendation, so removing them helps to simplify the data.
- The next step is to remove any unwanted characters such as punctuation marks, which are also irrelevant to the analysis. After that, we are removing any Twitter-specific symbols such as the @ and # symbols. Again, these symbols are not important for the analysis and can be safely removed.
- The most important step in this preprocessing is to remove stop words and lemmatize the remaining words. Stop words are commonly used words in a language (such as "the", "a", "an", "and", etc.) that are not meant for analysis. Removing them helps to focus on the more relevant words in the text. NLTK library provides a set of predefined stop words for the English language, which are loaded into the stop words object.
- We are using the word tokenize function from NLTK to split the text into individual words and then the lemmatizer object to lemmatize each word. The output is a cleaned and standardized version of the review text, which can be used for sentiment analysis and book recommendation.

## 2.2 Exploratory Data Analysis

- Calculating the average score rating for a book from a reviews dataset is a useful way to get an overall idea of how well-received a book is among readers. Breaking down the distribution of average ratings into categories such as "bad," "okay," "good," and "excellent" provides an even clearer picture of how readers are responding to the book.
- We are using a pie chart to visualize the distribution of average rating categories, as it allows for a quick and easy comparison of the number of reviews falling into each category. By displaying the data in this way, patterns and trends can become more apparent and insights can be gained into the success of the book.

```
labels = ['Bad', 'Okay', 'Good', 'Excellent']

# create a new column 'Bucket' based on the bins and labels
df_stat['rating_label'] = pd.cut(df_stat['review/score'], bins=bins, labels=labels)
```

In [43]:
```
df_stat
```

Out[43]:

| | Title | review/score | rating_label |
|---|---|---|---|
| 0 | Moving Forward | 5.0 | Excellent |
| 1 | Teaching Hand Papermaking: A Classroom Guide | 5.0 | Excellent |
| 2 | Teaching Developmental Reading: Historical, Th... | 5.0 | Excellent |
| 3 | Teaching Children About Life and Earth Science... | 5.0 | Excellent |
| 4 | Teaching Black Girls (Counterpoints) | 5.0 | Excellent |
| ... | ... | ... | ... |
| 212396 | City in the Sky | 1.0 | Bad |
| 212397 | The History of Presidential Elections | 1.0 | Bad |
| 212398 | All about OSHA (SuDoc L 35.2:OC 1/2/994) | 1.0 | Bad |
| 212399 | Laurens & Newberry Counties South Carolina: Sa... | 1.0 | Bad |
| 212400 | Political Change in the Metropolis (6th Edition) | 1.0 | Bad |

212401 rows × 3 columns

**Figure 6.** Distribution of Average Review Score

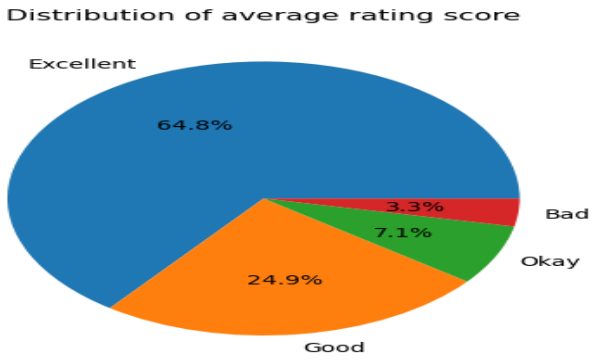


**Figure 7.** Distribution of Average Review Score

- When analyzing book reviews, it is important to consider the category of the book being reviewed. Reviews for books in different categories can have varying language, tone, and content.
- The category of a book can also help in determining the target audience for a particular book. For instance, young adult novels typically target teenagers, while children's books are aimed at young children. Therefore, knowing the category of a book can help in recommending it to the appropriate audience.

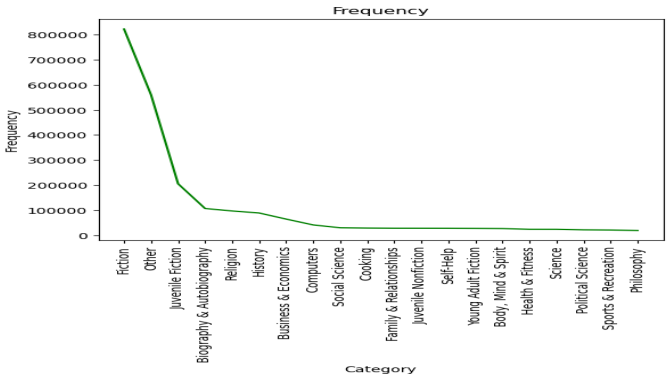


**Figure 8.** Categories Distribution

- The line chart above depicts which category has the maximum count of reviews. This information can be used to gain insights into the reading preferences of a particular population or to identify gaps in a library or bookstore's collection.
- We can also observe that there are many null values for categories that are represented by other category.

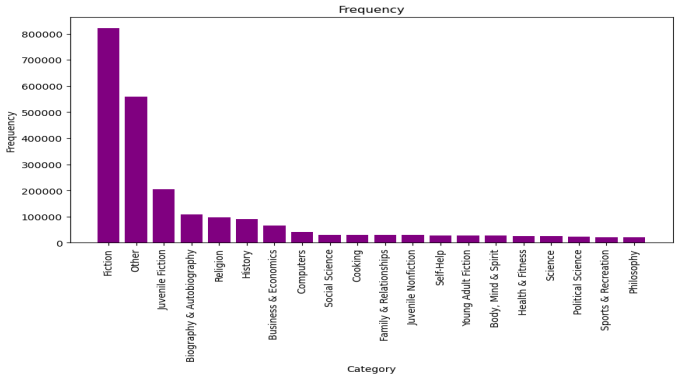


**Figure 9.** Categories Distribution

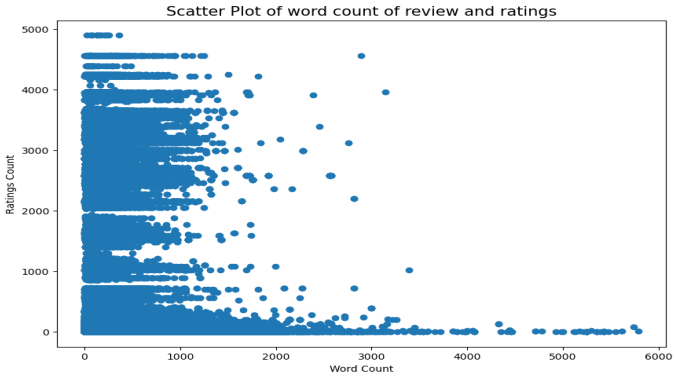


**Figure 10.** Scatter plot of Ratings Count vs Word Count

- This can help to understand the relationship between the length of a review (in terms of word count) and the number of ratings it has received. Additionally, it can be useful for sentiment analysis as longer reviews may have more in-depth analysis and opinions which can provide a more nuanced understanding of a book's reception. Furthermore, the plot can reveal any outliers or trends in the data.
- From the scatter plot, we can observe that there are some outliers in the data, which means there are some reviews that are very long and have a high number of ratings. These reviews could be either very positive or very negative, and they may have contributed to the overall popularity of the book.

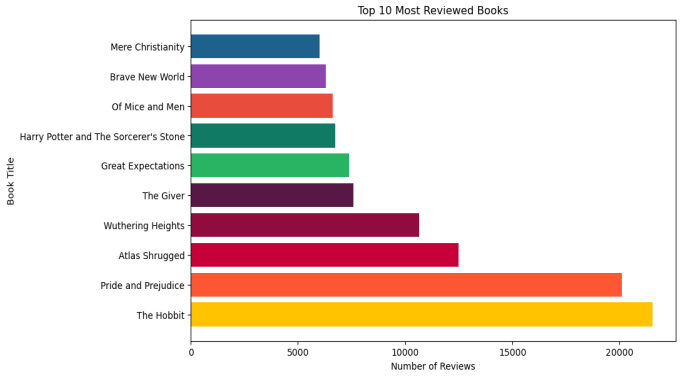


**Figure 11.** Top 10 Most Reviewed Books

- The above bar chart shows the top 10 most reviewed books that can be useful in making recommendations to readers who are looking for popular or highly-rated books. These books have already gained a lot of attention and interest, which makes them more likely to be enjoyed by a wider audience.
- Also, by analyzing the reviews of these books, we can identify the common themes, topics, and characters that readers enjoy or dislike, and use this information to recommend similar books to readers who have enjoyed these popular books.

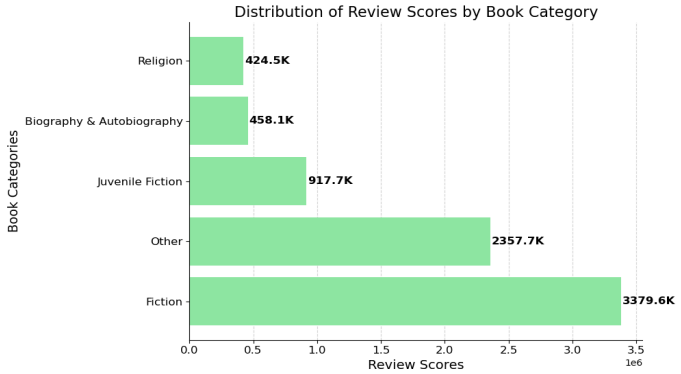


**Figure 12.** Categories and its average review score

- By analyzing the distribution of review scores for different book categories, we can identify which categories are more likely to receive positive or negative reviews. For example, if we observe that books in the "Fiction" category have a higher average review score compared to books in the "Biography & Autobiography" category, this can suggest that readers generally prefer fiction over non-fiction or biographical books.
- If a reader enjoys books in a particular category that tends to have high review scores, we can recommend other books in that category with similar characteristics.

# 3. Algorithm and Methodology

## 3.1 Sentiment Analysis:

**Support Vector Machine - Sentiment Analysis**

- Support Vector Machine algorithm trains multiple binary classifiers, each designed to distinguish between one class and the rest of the classes. For instance, if we have three classes (positive, negative, and neutral), we would train three binary classifiers: positive vs. (negative and neutral), negative vs. (positive and neutral), and neutral vs. (positive and negative).
- To classify a new review, we use each of the trained binary classifiers to make a prediction. The predicted class is the one associated with the classifier that returns the highest confidence score. The confidence score is the distance between the hyperplane and the input review's features. The higher the confidence score, the more confident we are in the predicted class.[1]
- Despite its effectiveness in sentiment analysis, we have found through experimentation with our dataset that SVM with the one-vs-all strategy does not perform well.

**Rule-Based Sentiment Analyzer:**

- The rule-based sentiment analyzer in NLTK works by assigning positive or negative scores to each word in a given text based on a predefined set of rules. These rules can be based on various factors such as the context in which the word appears, its part of speech, and its relationship to other words in the text.
- NLTK sentiment analyzer is a rule-based approach that relies on predefined rules and lexicons to identify the sentiment of a text. When given a text, the analyzer assigns a positive or negative score to each word based on the lexicon and then calculates an overall sentiment score for the text by aggregating the individual scores.
- For example, if the text contains words such as "happy", "joyful", and "excited", the analyzer will assign positive scores to each of these words, and the overall sentiment score for the text will be positive. [4]

**Result of Sentiment Analysis:**

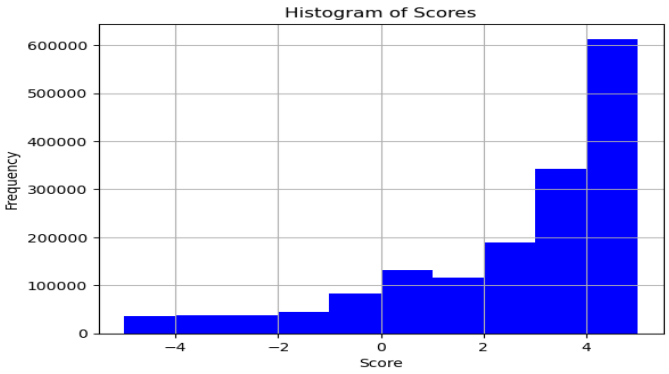


**Figure 13.** Sentiment Score Distribution

### 3.2 Recommendation System:

**Random Forest Algorithm:**

- The main idea behind Random Forest is to create a large number of decision trees (known as an ensemble) and then aggregate their predictions to make a final prediction. Each decision tree is trained on a random subset of the training data and a random subset of the features. This helps to reduce over-fitting and increase the model's robustness.
- Random Forest creates a large number of decision trees (usually hundreds or thousands) using a subset of the training data and a subset of the features.
- Once the decision trees have been created, their predictions are aggregated to make a final prediction. For book recommendation, this typically involves calculating the average rating predicted by each decision tree for a given user-item pair.
- According to one of the research papers "A random forest approach for rating-based recommender system" Comparing Results with other algorithms[2]:

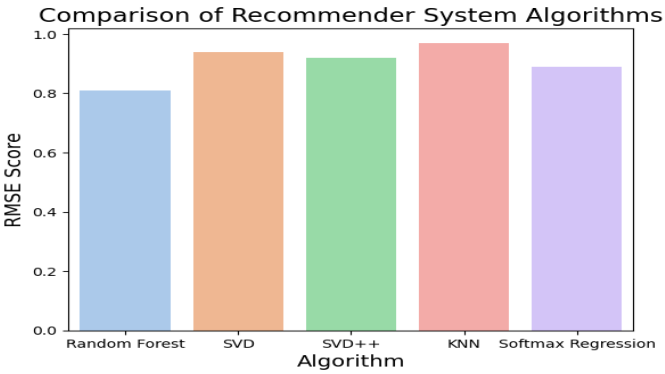


**Figure 14.** Comparison of rmse of random forest algorithm with other models

- Random Forest is a model-based algorithm that requires training on a large dataset to make accurate predictions. The system needs to handle sparsity in the data and provide accurate recommendations for users with few ratings, Random Forest algorithm is not a better choice for this use case.

**KNN with Means:**

- KNNWithMeans is a collaborative filtering algorithm used in recommender systems. It works by finding the k nearest neighbors of a user or an item based on their similarity in terms of ratings given by users[3]. The similarity is calculated using various distance metrics such as cosine similarity or Pearson correlation coefficient.
- The "Means" in KNNWithMeans refers to the fact that the algorithm takes into account the mean ratings of each user or item when making predictions. This helps to account for differences in rating scales between users or items.
- We have trained the algorithm on a dataset containing user-item ratings using the KNNWithMeans class from the Surprise library. The dataset is split into a training set and a test set, with 25% of the data being used for testing.
- The algorithm is initialized with k=50, meaning that it will consider the 50 most similar users or items when making predictions. The similarity measure used is Pearson baseline similarity, which is a variation of the Pearson correlation coefficient that takes into account the global mean rating and biases of each user and item.
- The top 40 items are then selected based on the predicted ratings and sorted in descending order. Finally, the top 5 items are recommended to the user based on their predicted ratings[3].

## 4. Experiments and Results

### 4.1 Experiments for Sentiment Analysis using SVM and Rule-Based Analyzer:

- The below table represents the results of experiments performed using the support vector machine algorithm and the Rule Based Sentiment Analyzer.
- The SVM sentiment score of 0.52 indicates that the "I really liked hearing from my old friends" in the second record is slightly positive, but not strongly positive. On the other hand, the NLTK rule-based sentiment analyzer gives a higher sentiment score of 0.74, which suggests that the sentence is more strongly positive.
- The SVM model is less sensitive to the specific words and phrases used in the sentence. The NLTK rule-based analyzer, on the other hand, takes into account the specific words and their polarity to determine the sentiment of the sentence.

| Text | SVM | Rule Based Analyzer |
|---|---|---|
| "Researchers looked at NGC 346, a highly active star-forming region in a galaxy near the Milky Way called the Small Magellanic Cloud (SMC). They chose this place because it has a very low concentration of metals — which astronomers define as any element heavier than hydrogen and helium." | +0.10 | +0.12 |
| "I really liked hearing from my old friends" | +0.52 | +0.74 |
| "Data mining is one of the best subjects I took." | +0.34 | +0.64 |
| "Well written chronicles of The Farm from the people who lived it. Excellent photos as well. I kept looking for a photo of me and my then-wife, but I wasn't there long enough" | +0.55 | +0.72 |
| "This book isn't that good." | -0.82 | -0.34 |

**Figure 15.** Sentiment Analysis Results

## 4.2 Results of Sentiment Analysis:


**Figure 16.** Sentiment Analysis Results

- We can observe that the sentence provided here is "This book isn't that good." which gives results a negative sentiment. Our model classifies this sentence as a negative review and gives a sentiment score of -0.3412.

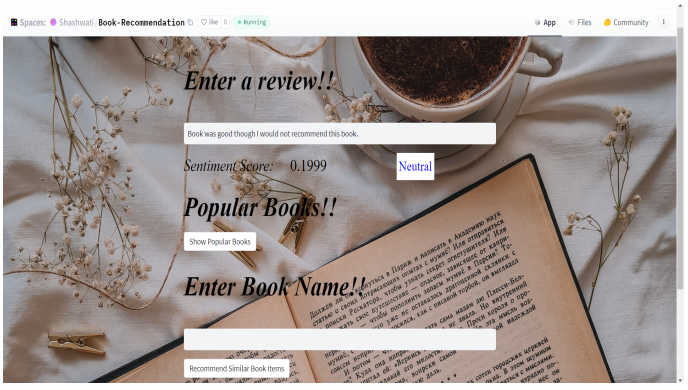


**Figure 17.** Sentiment Analysis Results

- In the above image we can see that the sentence "Book is good though I would not recommend" is classified as a Neutral review. Our model trained on the dataset which contains reviews that has a sentiment score ranging from -5 to +5 from which we consider the range for a neutral sentiment score is -0.3 to + 0.3.

## 4.3 Experiments for Recommendation System:


**Figure 18.** Similar Books Based on Given Book Name

- For the recommendation system we have developed three modules. Whenever the user opens the system they can see trending or popular books. When the user enters and book name it will recommend similar books. If there is no book to recommend it will show a message that there is no book to recommend and can give you some book names which you can try.
- In Fig 19 we can see that the user has provided one book named "Dissent in Dangerous Times" and as a result, we get that there is no recommendation for this book.

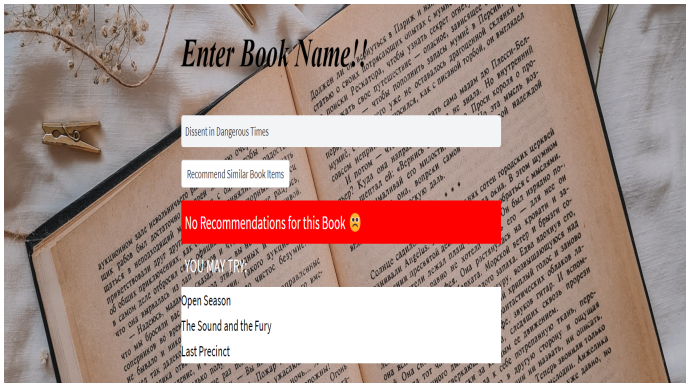


**Figure 19.** No Book to Recommend


**Figure 20.** Similar Books for Given User ID

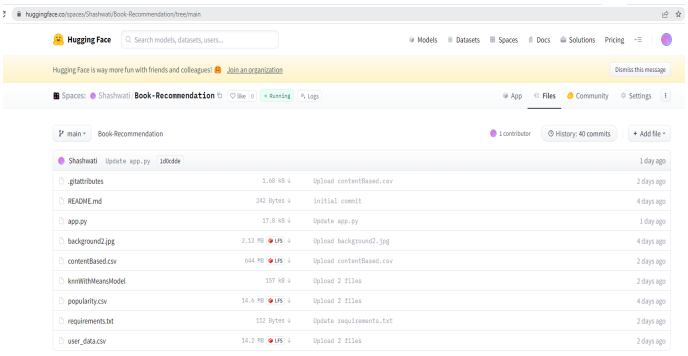


**Figure 21.** Github Repository

- When the user provides his user ID system will recommend similar books based on the previous rating given by similar users. We have tested with some of the user IDs, we can observe in the above Fig that system has recommended some books to the user ID - A20EEWWSFMZ1PN this user gave reviews for books under the fiction category and our system also recommends books to this user id falls under the fiction category.

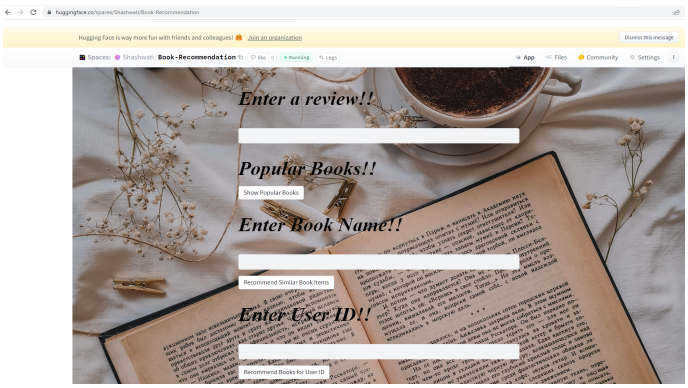


**Figure 22.** Deposed Application

## 5. Deployment and Maintenance

We have deployed our sentiment analysis and book recommendation application on Hugging Face Spaces, using the following steps:

- Our complete repository, including data files, models, and app.py, was included in the deployment in Hugging Faces github.
- app.py is the main file of the application, and any changes made to the repository will trigger a new build.
- Once the build is completed successfully, we can see our deployed application under the ”App” section on Hugging Face Spaces.
- To maintain version control and support for bugs, we are using Hugging Face’s Git system.
- Any new changes committed to the repository will trigger a new build and the changes will go live once successful.
- By using Hugging Face Spaces and Git, we can easily deploy, maintain, and update our application with version control and support for bugs.
- Hugging Face Github Repo (DM Project)
- Deployed Application (Sentiment Analysis and Recommendation for Books)

## 6. Summary and Conclusions

In today’s digital age, where books are being bought and reviewed online, sentiment analysis of book reviews has become crucial for publishers and authors to understand how their books are being received by readers. By automatically categorizing a book review’s sentiment as positive, negative, or neutral, sentiment analysis can help publishers and authors identify their strengths and weaknesses, which can guide them in improving their future works. In the proposed system we have used rule-based sentiment analysis which gives an accuracy of 86.50%.

Moreover, book recommendation systems are becoming increasingly important in helping readers discover books that match their interests and preferences. By analyzing the tone of user reviews and other user data, book recommendation algorithms can better understand user preferences and behavior, resulting in higher sales and more efficient promotion of books. The proposed system has been developed with three modules, including trending or popular books, recommending similar books when the user enters a book name, and recommending books based on the previous ratings given by similar users. The system has been tested with different user IDs, and we can see that it recommends books falling under the same category as the books previously rated by the users, which shows how to book recommendation systems can provide a personalized and relevant experience to the users.

In conclusion, sentiment analysis of book reviews and book recommendation systems are crucial for both readers and publishers/authors. By providing insightful information about the sentiment of book reviews, sentiment analysis can help publishers and authors identify their strengths and weaknesses, which can guide them in improving their future works. On the other hand, book recommendation systems can help readers discover books that match their interests and preferences, resulting in higher sales and more efficient promotion of books for publishers and authors.

## Acknowledgments

We would like to acknowledge the unwavering support and guidance of Drs. We would like to express our sincere thanks to Mr. Hasan Kurban and his teaching assistants. Their expertise and constructive feedback helped us succeed in our project and achieve our goals. We also thank all the faculty in attendance who provided valuable insights and comments during this project

We would also like to thank the organizations and individuals who provided us with the materials and support needed to carry out this project. We also thank our colleagues and friends who gave us moral support and encouragement throughout this project.

Finally, we would like to thank our team members for their hard work and dedication in completing this project. We worked together as a team and each member played an important role in the success of this project. We are grateful for the opportunity to work together and learn from each other's experiences.

## References

[1] A. Mounika, 2Dr. S. Saraswathi, "CLASSIFICATION OF BOOK REVIEWS BASED ON SENTIMENT ANALYSIS: A SURVEY", IJRAR, June 2019, Volume 6, Issue 2.

[2] Ajesh A1, Jayashree Nair1, Jijin PS1, "A Random Forest Approach for Rating–based Recommender System", Conference on Advances in Computing, Communications, and Informatics (ICACCI), Sept. 21-24, 2016.

[3] Wang, H., Zhang, F., & Chen, G. (2017). Collaborative filtering recommendation algorithm based on KNN and user interests. In Proceedings of the 2nd International Conference on Communication and Information Processing (ICCIP 2017) (pp. 292-297). Atlantis Press.

[4] Singh, S., Kumar, S., & Kumar, V. (2019). Comparative analysis of rule-based and machine learning approaches for sentiment analysis. International Journal of Computational Intelligence and Applications, 18(03), 1950009.