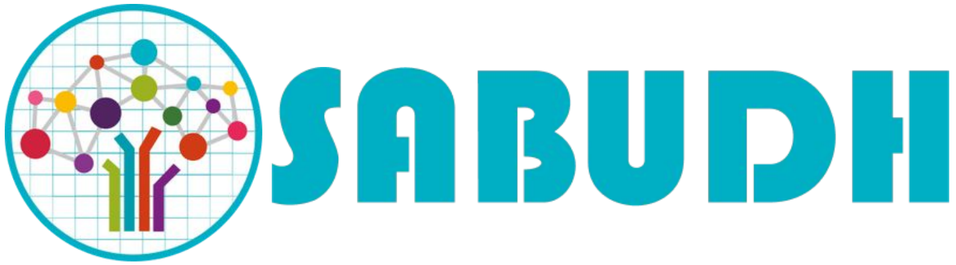
AI for Book Analysis



Midway Progress

06-11-2024

Project Mentor(s): Apoorav Mittal
Aashish Rana
Jaybrata Chakraborty

Presented by: Rohan Singh
Ankita Saha
Sanghamitra Goswami
Shreya Chatterjee
Jaykishan Padia
Thaidu Sai Vamshi
Prabhnoor Kaur

Problem Statement

- Readers face difficulty in discovering new books due to overwhelming choices and limited personalization in traditional recommendation systems.
- Existing systems often rely on either content-based or collaborative filtering, which fail to account for evolving user preferences and detailed book characteristics like themes and genres.
- This project aims to create a hybrid AI-driven book recommendation system that combines content-based and collaborative filtering with a genre classifier.
- A user-friendly interface will enable seamless interactions, with deployment on AWS for scalability and maintenance.

Project objectives

01

To review existing research and methodologies, the techniques used.

02

To introduce the concept of recommendation systems(Collaborative and Content Based Filtering).

03

To develop a scalable system and implement an user interface for seamless user interaction.

DATA FETCHING

- We fetched datasets from github repository.
- It was a goodreads book datasets.
- The dataset was loaded in chunks.
- Then we combined the chunk datasets and performed EDA on it.

Datasets

Meta-Data of Books

- Detailed book graph (~2gb, about 2.3m books): goodreads_books.json.gz
- Detailed information of authors: goodreads_book_authors.json.gz
- Detailed information of works (i.e., the abstract version of a book regardless any particular editions): goodreads_book_works.json.gz
- Detailed information of book series (Note: Unfortunately, the series id included here cannot be used for URL hack): goodreads_book_series.json.gz
- Extracted fuzzy book genres (genre tags are extracted from users' popular shelves by a simple keyword matching process): goodreads_book_genres_initial.json.gz

Book Shelves

- Complete user-book interactions in 'csv' format (~4.1gb): goodreads_interactions.csv
  User Ids and Book Ids in this file can be reconstructed by joining on the following two files: book_id_map.csv, user_id_map.csv.
- Detailed information of the complete user-book interactions (~11gb, ~229m records): goodreads_interactions_dedup.json.gz
- User-Book Club mapping information: book_clubs.json

Data Description

- The dataset contains 647,961 **rows** and 29 **columns** with a mixture of categorical and numerical columns.
- The **columns** are isbn, text_reviews_count, series, country_code, language_code, popular_shelves, asin, is_ebook, average_rating, kindle_asin, similar_books, description, format, link, authors, publisher, num_pages, publication_day, isbn13, publication_month, edition_information, publication_year, url,image_url, book_id.
- The data types of the columns are int64,float64,object and bool.
- There were missing values in some columns with the highest missing values in the language_code column.
- There were no duplicate rows in the dataset.

Data Exploration

Dataset Overview:

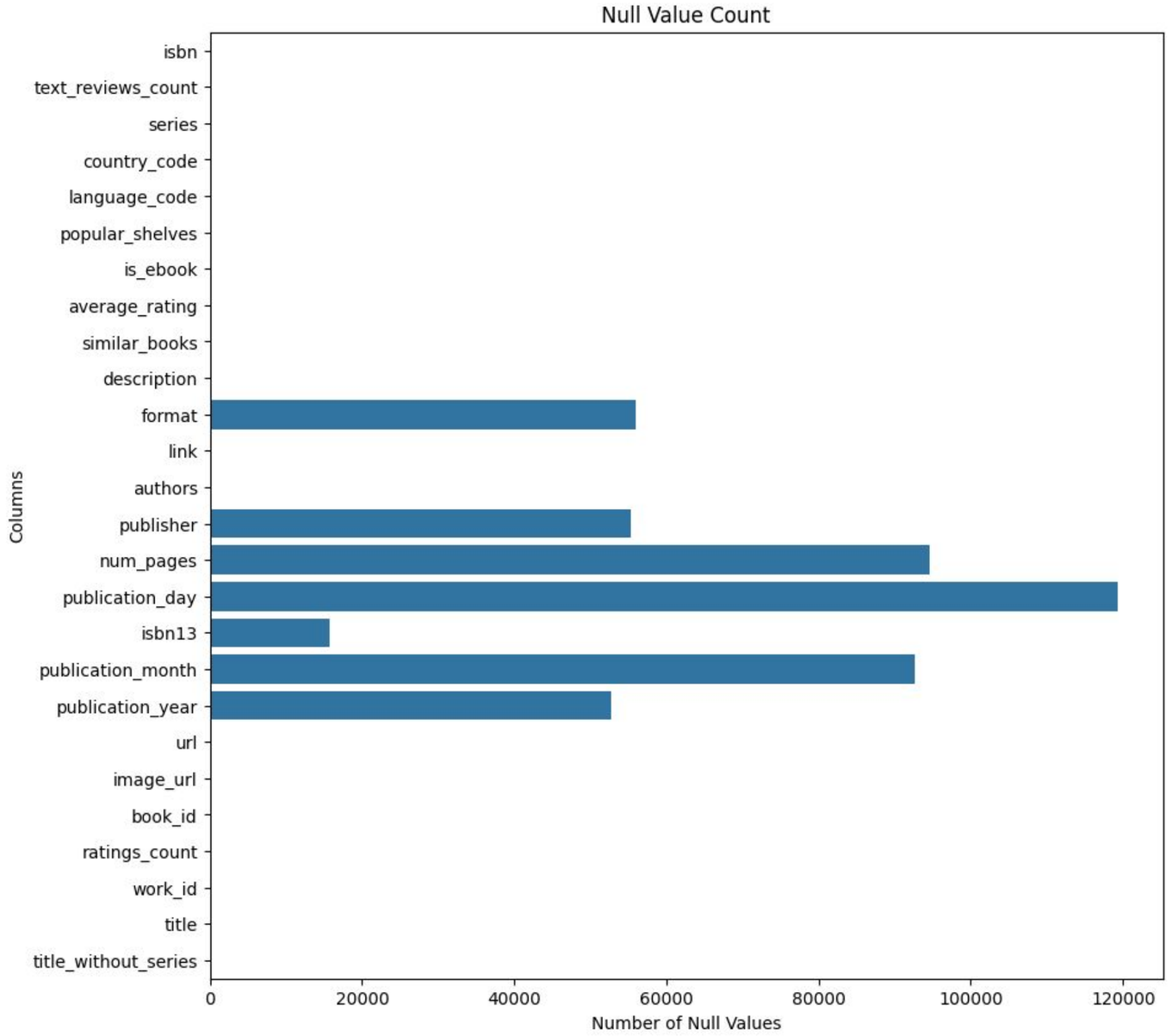
- Rows: 647,961 entries
- Columns: 29 attributes covering book reviews, ratings, publication details, etc.

Statistical Summary:

- Text Reviews Count: Mean of 57.38; maximum of 142,645 reviews, indicating a significant variance in user engagement across titles.
- Average Rating: 3.88, with a maximum of 5, suggesting overall favorable ratings.
- Number of Pages: Average of 273 pages, but some titles reach as high as 82,000 pages, possibly due to unique formats like compilations or data entry errors.
- Publication Date:
  - Day: Average of 11.57, indicating publications often occur around mid-month.
  - Month: Evenly distributed across the year.
  - Year: Average of 2005, though the maximum of 20091 suggests data entry errors.
- Ratings Count: Mean of 909, with a maximum of 48,00,000 ratings, indicating high variability in popularity.

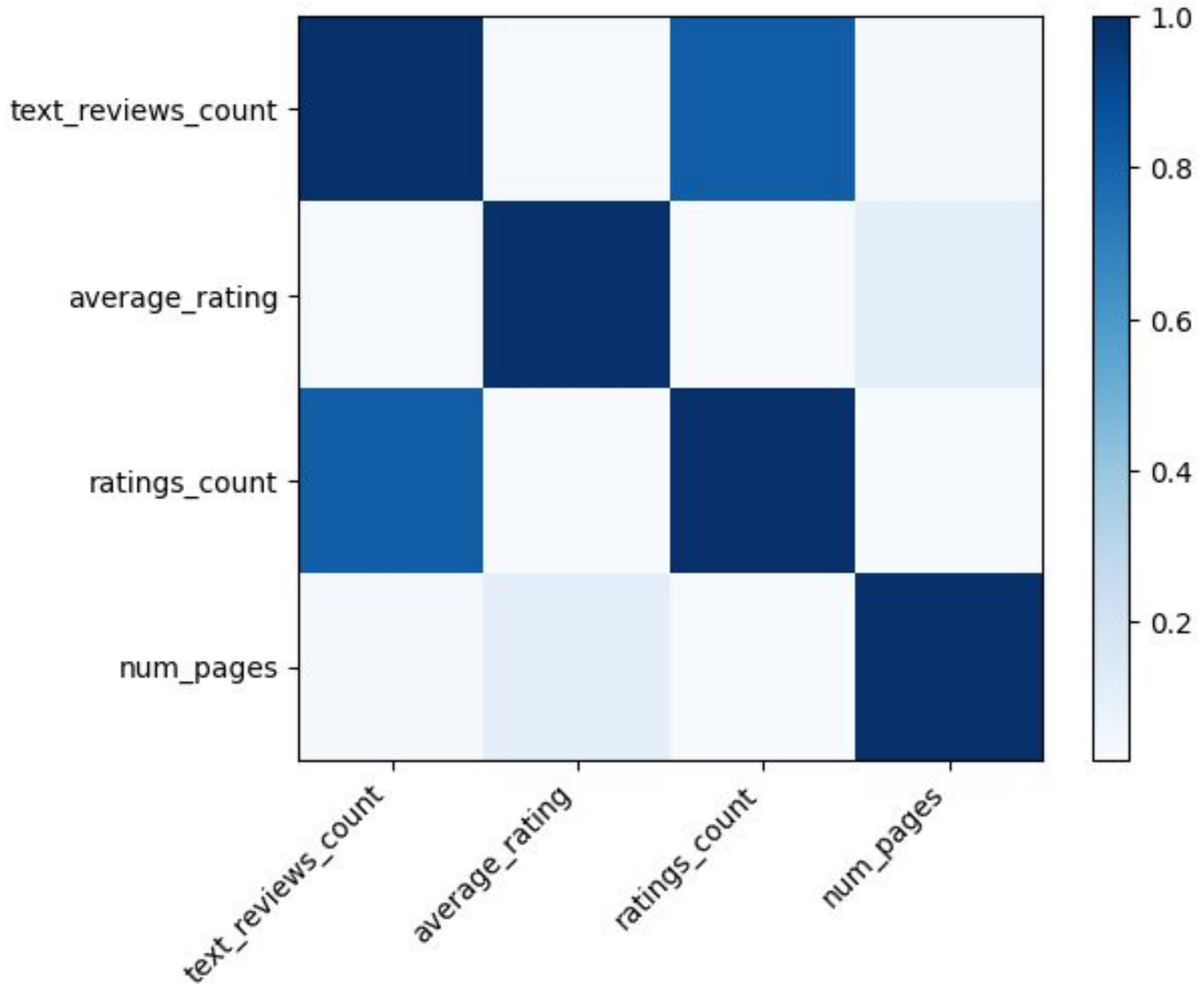
Data Exploration

Missing Values:



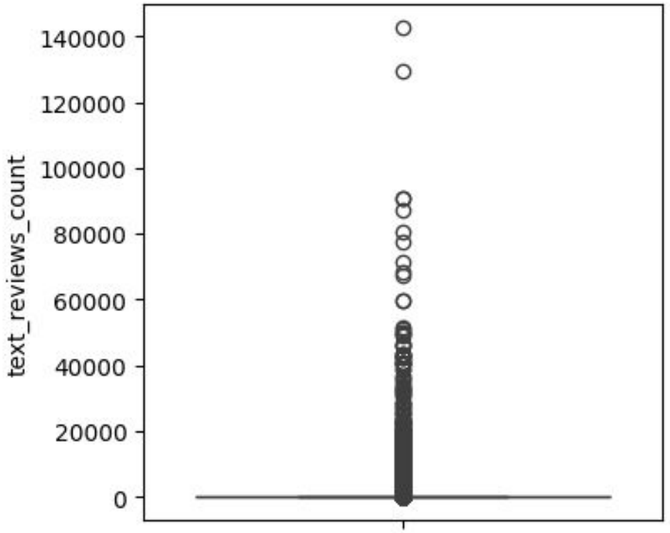
Data Exploration

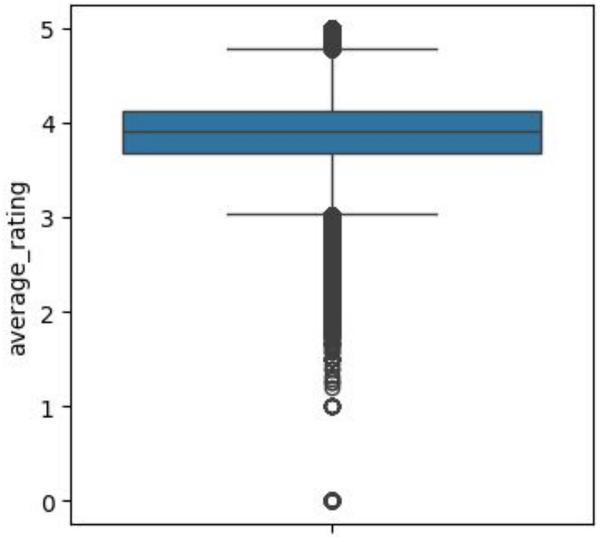
Correlation Among Numerical Features:

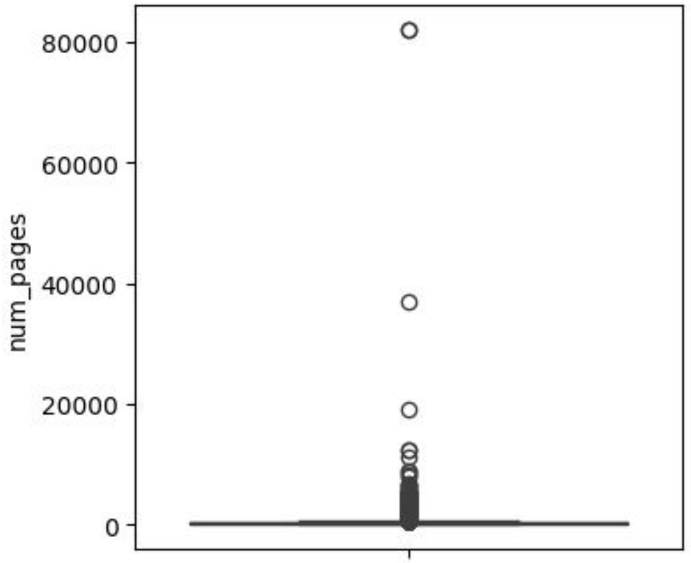


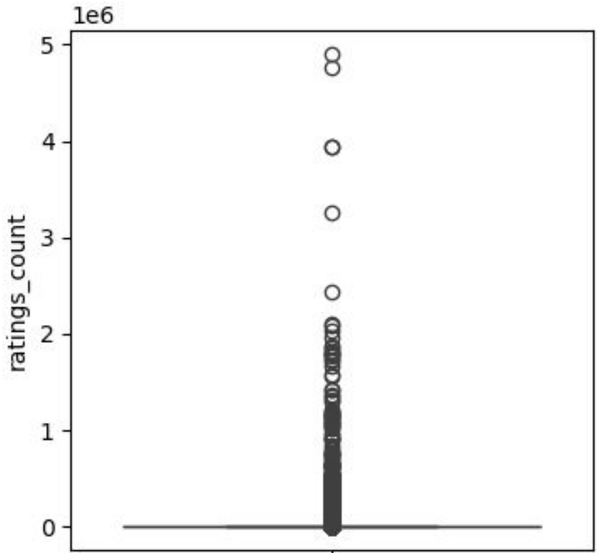
Data Exploration

Distribution of Key Numerical Features:

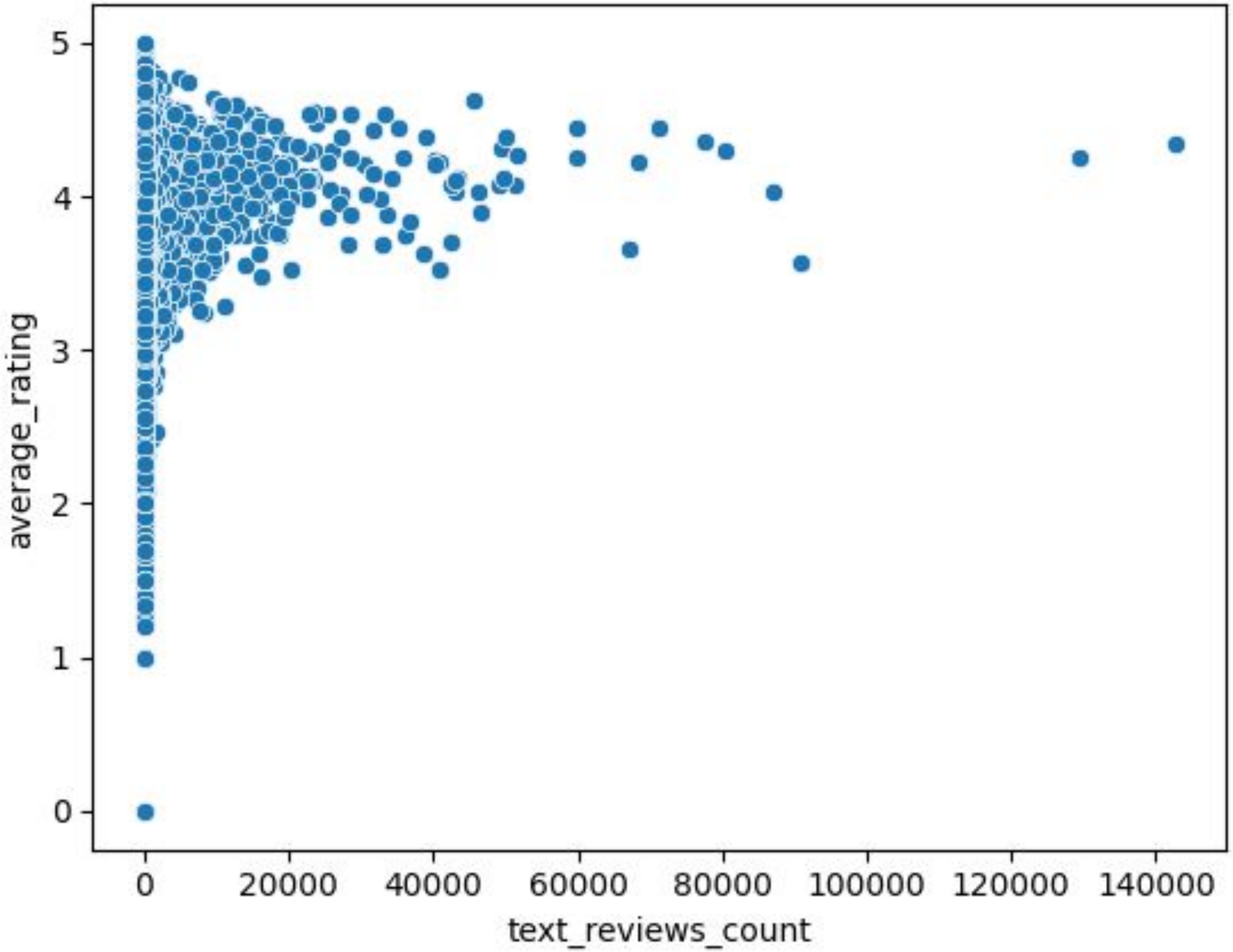




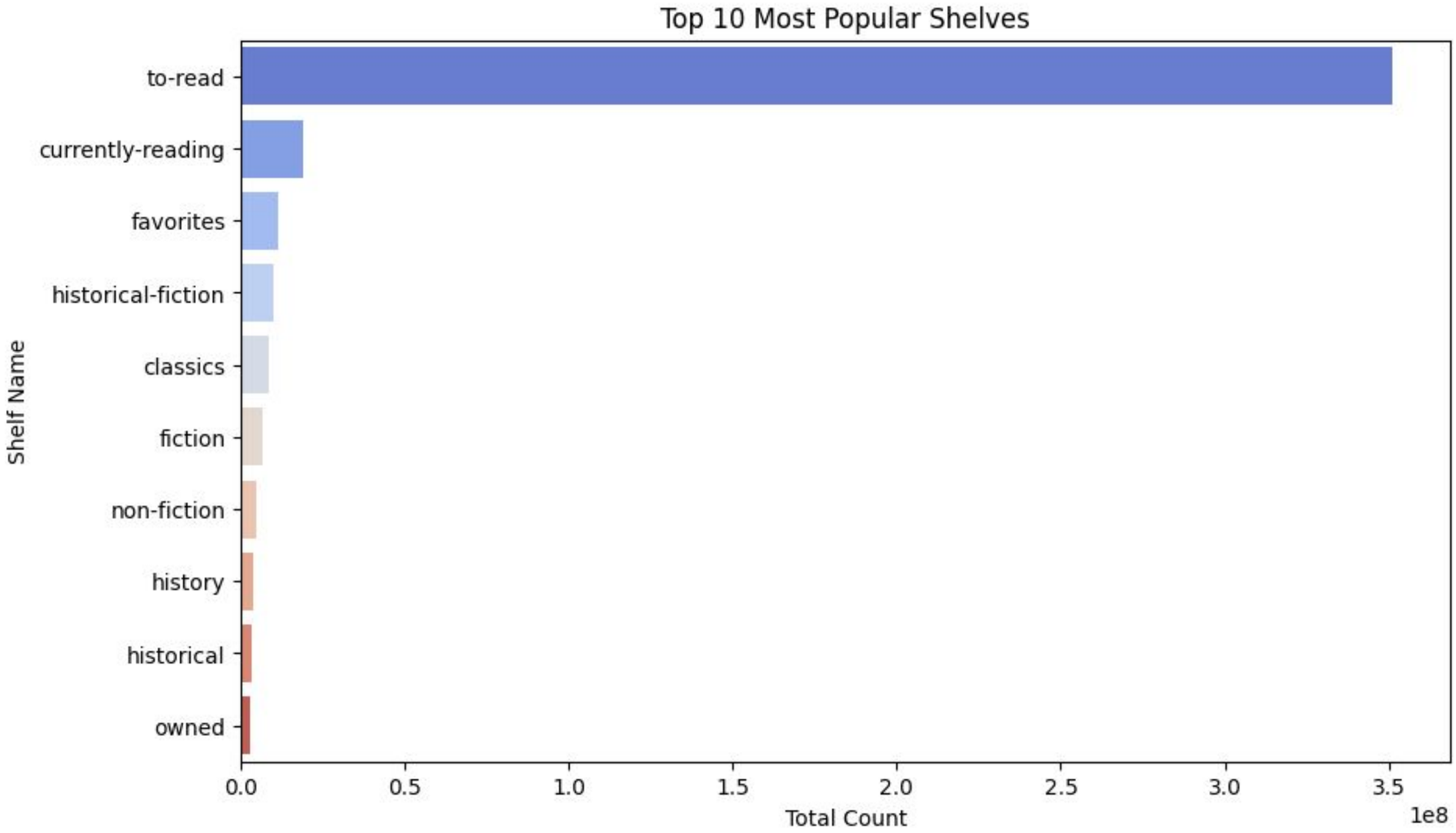




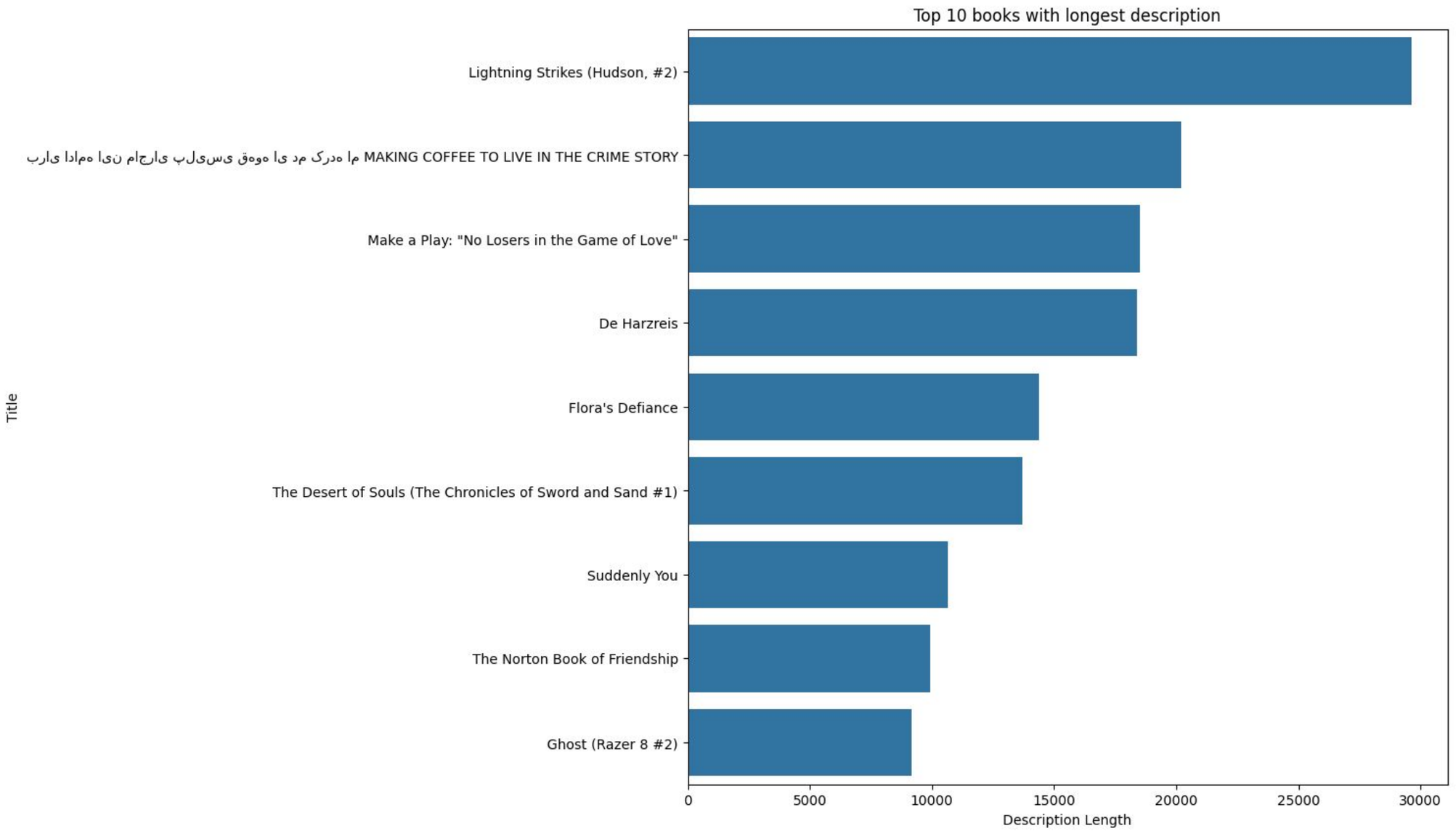
Data Exploration



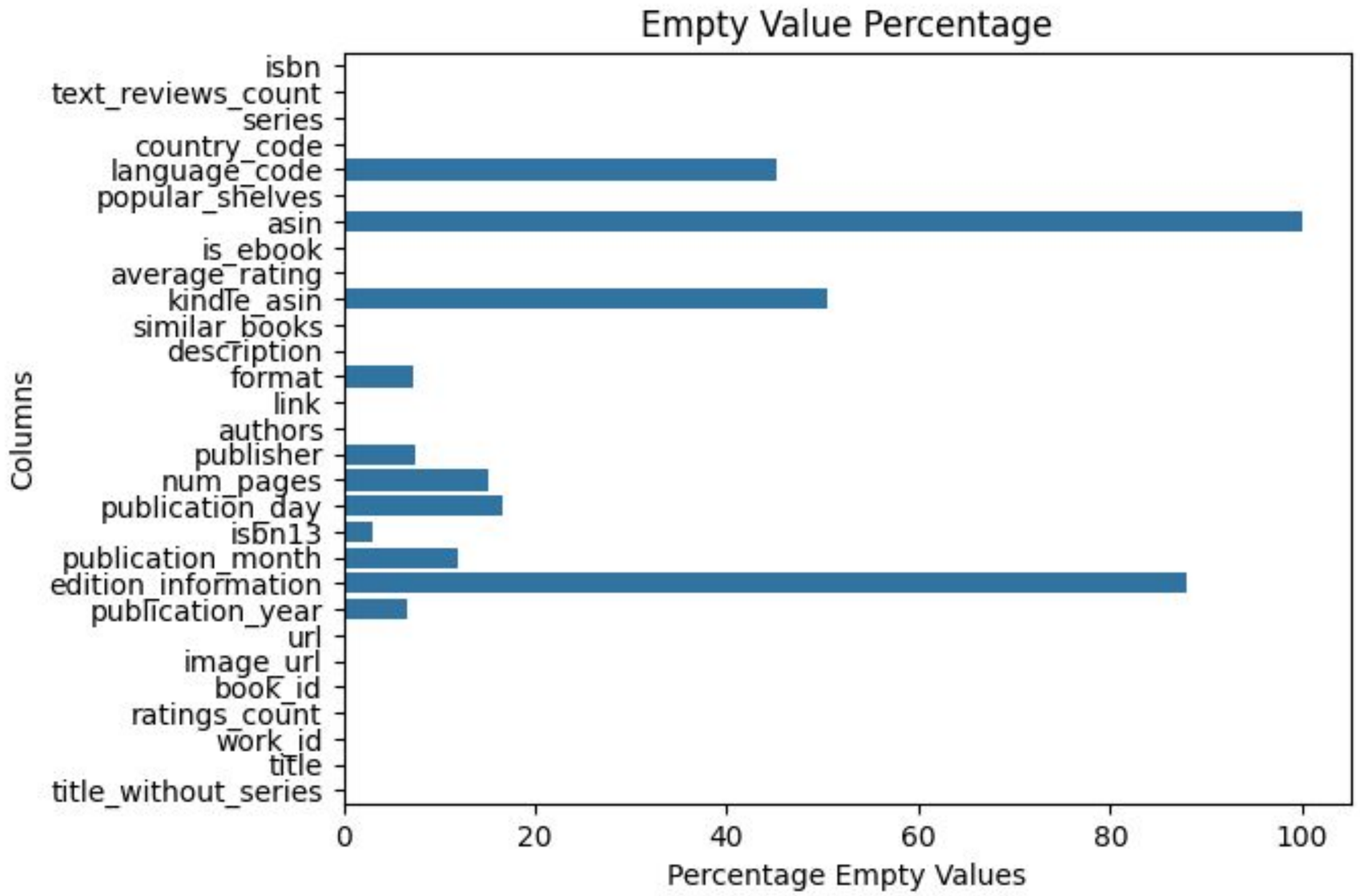
Data Exploration



Data Exploration



Data Cleaning



Columns with more than 50% missing values were removed to ensure data quality and relevance.

Columns Dropped:

- **ASIN:** Amazon Standard Identification Number, often unique to certain editions.
- **Kindle ASIN:** Identifier specific to Kindle editions, which may not be available for all books.
- **Edition Information:** Optional detail about book editions, which had high missing values.

Data Transformation

Objective: Ensure all book descriptions are in English for consistency in analysis.

Method:

- **Language Detection:** Used langdetect to identify the language of each description. If the detected language was not English, the text was flagged for translation.
- **Translation:** Leveraged googletrans to translate flagged descriptions into English, standardizing all text data.

Tools/Libraries:

- **langdetect:** To detect language codes of descriptions.
- **googletrans:** To translate non-English descriptions into English.

Error Handling:

- Managed LangDetectException errors from langdetect for cases with insufficient text, ensuring the process ran smoothly.

Data Storing and Retrieval

- As we know MongoDB is an open-source document-oriented database that is designed to store a large scale of data and also allows you to work with that data very efficiently so we preferred MongoDB for final data storage and retrieval.
- We downloaded MongoDB first, then the shell and we updated the environment variables.
- We imported the ‘MongoClient’ from ‘pymongo’ and declared the database name and collection name in jupyter notebook.
- Then as usual we imported the database (present in csv) in jupyter notebook using pandas.
- As MongoDB is Schema-less database management system we needed to store the imported data into ‘records’ into ‘dictionaries’.
- After inserting the datas into the collection we could use the ‘print()’ command to retrieve the data from the collection.

Proof of Concept (POC)

```
       book_id                                                title  \
1936  16718170          The Third Wheel (Diary of a Wimpy Kid, #7)
3196  14070444  Viaje al Bosque: un maletín lleno de Historias...
5790  18984670                       How to Steal a Dragon's Sword
8168   3116884                  Curious George Learns the Alphabet
9518   2775591                            The Teddy Bears' Picnic

                                                 authors  average_rating
1936                [{'author_id': '221559', 'role': ''}]            4.20
3196  [{'author_id': '288388', 'role': ''}, {'author...            4.83
5790  [{'author_id': '23894', 'role': ''}, {'author_...            4.43
8168                [{'author_id': '967839', 'role': ''}]            4.23
9518  [{'author_id': '60143', 'role': ''}, {'author_...            4.22
```

Proof of Concept (POC)

- Model Training: The code uses the SVD algorithm from the Surprise library to train the recommender system based on the collaborative approach.
- Recommendations: The function get_recommendations generates book recommendations based on the predicted ratings for books that the user has not yet rated
- The output received is a list of recommended books for the synthetic user . Each entry in the list provides two pieces of information:
- Book ID: The first element in each tuple is the ID of the recommended book. This ID corresponds to a specific book in dataset.
- Estimated Rating: The second element is the estimated rating that the model predicts the synthetic user would give to that book. This rating is based on the user's past interactions and the characteristics of the book.
- **RMSE: 0.3619**

Conclusion and Future Scope

- After performing EDA on our final dataset we got to know about the anomalies, the pattern & data quality of dataframe. From the EDA we also got to know about the missing values and took our modeling decisions.
- We also cleaned the dataset by removing maximum number of null values and also by taking actions on duplicate data to ensure data quality.
- We have prepared a POC(Proof of Concept)- based on which we would proceed making our recommender system.
- In our future endeavors, we are also trying to make this a content based recommender system.
- We would then compare the models and observe which is a better approach.

Thank you