Data Science 3rd Assignment

Dataset: Book Prices

Mahan Madani – 99222092

1. Overview

This dataset is comprised of over 5000 records of book sales, containing information in 9 different columns. The majority of the columns are text-based and include information on each individual book's title, author, price, and so on.

Due to this fact, NLP techniques can be used to extract useful data from this dataset, which will ultimately help us predict the price of a given book based on other attributes.

2. Dataset Exploration & Analysis

The dataset stores 9 different attributes. Each record in the dataset belongs to one specific edition of a book, with its own publishing date.

Below you can find a list of all the attributes:

- **Title:** The title of the book
- **Author:** The name of the author(s) of the book.
- **Edition:** The edition of the book, including its cover type, release date, and sometimes language (e.g., Paperback, – Import, 26 Apr 2018)
- **Reviews:** The number of customer reviews about the book
- **Ratings:** The customer rating of the book out of 5
- **Synopsis:** The synopsis of the book
- **Genre:** The genre the book belongs to
- **BookCategory:** The department the book is usually available at
- **Price:** The price of the book (Target variable). No currency was provided.

3. Data Preprocessing

3.1 Check for Duplicate Records:

To ensure the dataset contains no duplicate records, a combination of the 'Title', 'Author', 'Edition', 'Reviews', 'Ratings', 'Synopsis', 'Price' features can be used. If two or more records share the same value for these columns, that means they are potentially duplicates.

Even if the records aren't completely identical, they may still contradict each other and need to be handled. **After testing the dataset, it seems that it contains 525 duplicate records**. This is mostly a result of inconsistencies in the "Genre" column, creating multiple instances of the same book but with slightly different genres.

All instances of a duplicate record except the first one must be dropped. The goal is to avoid training our model with a skewed dataset.

3.2 Handle Null Values:

As seen in this table, out of the 9 features in the dataset none contain null values. No further action is required here.

	Null Count
Title	0
Author	0
Edition	0
Reviews	0
Ratings	0
Synopsis	0
Genre	0
BookCategory	0
Price	0

3.3 Detect Outlier Values:

Using the z-score method, outlier data can be identified. The only numerical column in the dataset is 'Price', so outliers are detected for that column. **164 records** were detected as outliers and should be dropped from the dataset.

3.4 Delete Unnecessary Text from Records:

Certain attributes ('Reviews' and 'Ratings') contain extra text for each record in addition to their numerical values. To simplify the database and prepare it to be use by a model, the extra text should be removed and only the numerical value should remain.

Following this step, all records contain a floating-point number between 0 and 5 in their 'Reviews' column, representing the average review score the book has received. Additionally, all records contain an integer greater than or equal to 0 in their 'Ratings' column, representing the amount of user ratings a book has received.

3.5 Improve Consistency:

A number of records share the same 'Author' or book 'Title', but sometimes the authors name or the book title is stored slightly differently (e.g., J. K. Rowling vs JK Rowling). To improve the dataset's consistency, these names should be stored similarly.

To achieve this, all author names and book titles were converted to lower-case and all characters other than alphanumeric ones were dropped (e.g., jkrowling is how the previous example is now stored).

3.6.1 Feature Generation:

Based on the available data, the following features can be added to further improve the dataset:

- **Year:** The year of the book's publication. Extracted from 'Edition'
- **CoverType:** The type of the book's cover (Paperback, Hardcover, etc.). This attribute is also extracted from the 'Edition' column.

Additional attributes such as Month can also be extracted from the dataset, however not all records have a month specified in 'Edition'. Some other attributes contain a varied collection of data that can't be handled manually, so a more powerful tool is needed to extract data from them.

3.6.2 Utilizing NLP for Feature Extraction:

The 'Synopsis' attribute contains a paragraph of text, providing a summary of the book in addition to other information about the author or the publication. This information is valuable but cannot be used in its current form.

Natural Language processing tools can be used to extract useful data from the 'Synopsis' column. First, I used Sentiment Analysis (from the TextBlob package) to extract numerical values relating to each record's 'Synopsis', these values are **Polarity** and **Subjectiveness**, and they were added as new features to the dataset.

Additionally, another usage of NLP is to extract the **Keywords** of a text. Utilizing the SpaCy package, keywords can be extracted from the 'Synopsis' and stored as a new feature. These keywords can later be vectorized and used for training the model.

3.7 Feature Transformation:

Training and testing a model require numerical data, and considering how the majority of this dataset is text-based, some features must be transformed so that they can be used in machine-learning tasks.

Vectorization is the technique used here. The 'Author', 'CoverType', and 'BookCategory' features can be vectorized using one-hot encoding, creating a new column for each unique category in them.

The 'Keyword' column has too many unique variations so one-hot encoding isn't the best approach here. Instead, another method called **TF-IDF** vectorization can be used to transform this attribute to a collection of numerical features.

Note that if the amount of newly added columns is extremely large, training the model can take a long amount of time. To reduce the number of features (Dimensions), Principal Component Analysis (**PCA**) is used.

4. Visualization

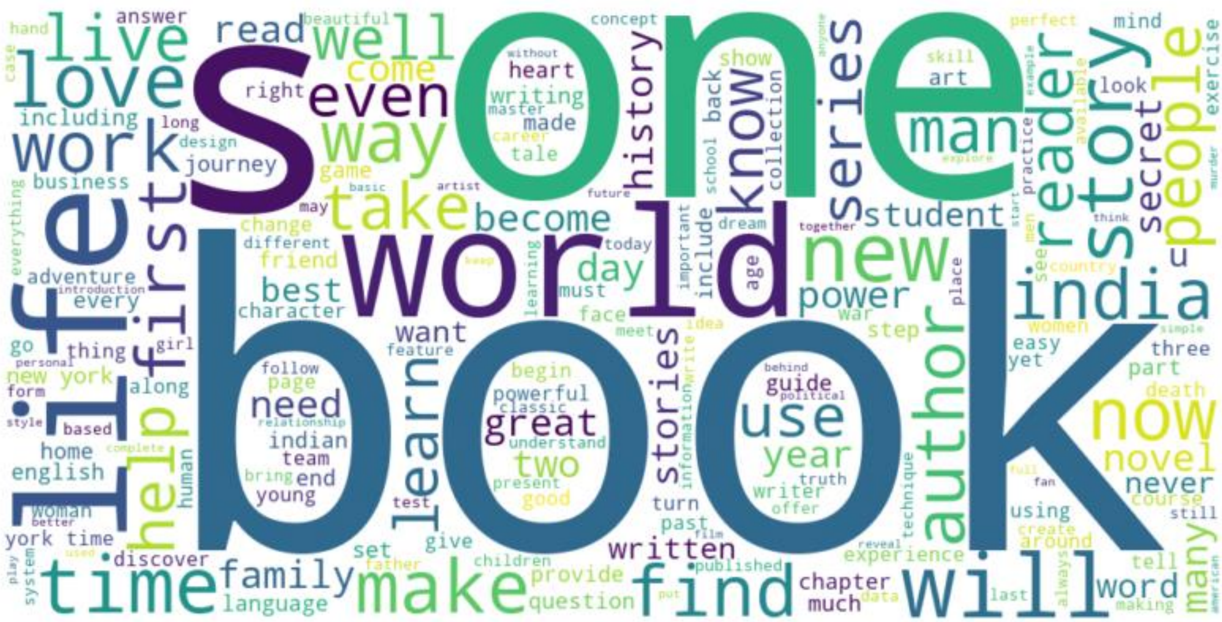


Figure 4.1: Wordcloud based on 'Synopsis'

This plot was generated based on all words in the 'Synopsis' column.

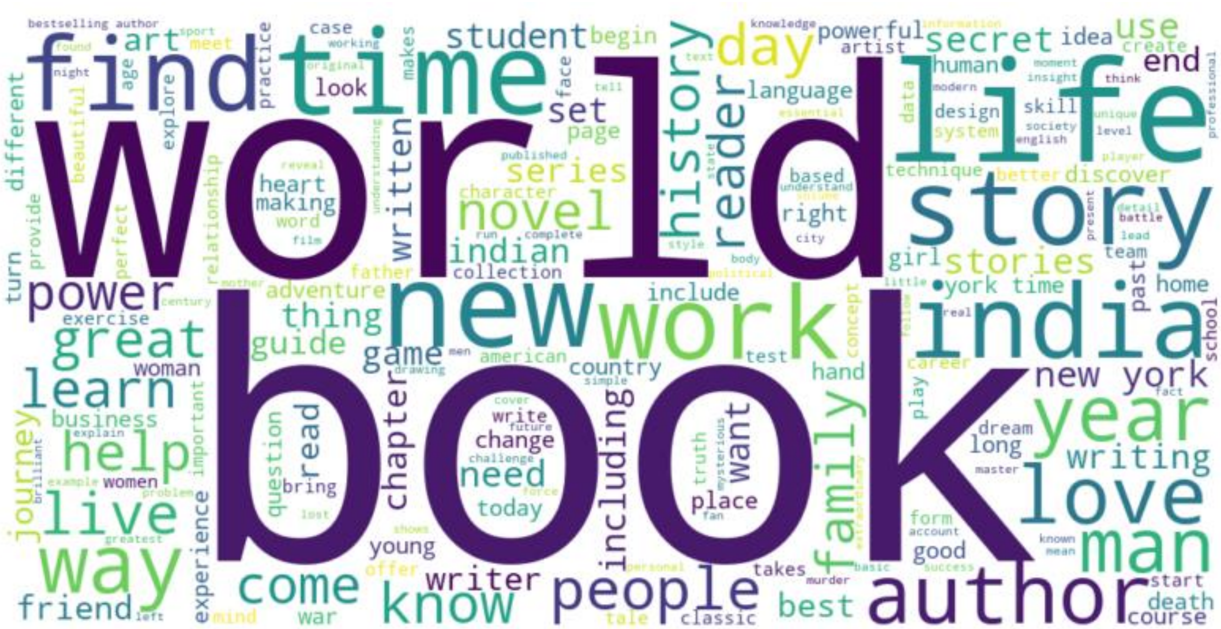


Figure 4.2: Wordcloud based on extracted Keywords

This plot was generated based on the Keywords extracted from 'Synopsis'. Interestingly, the word book is the most repeated in both wordclouds.

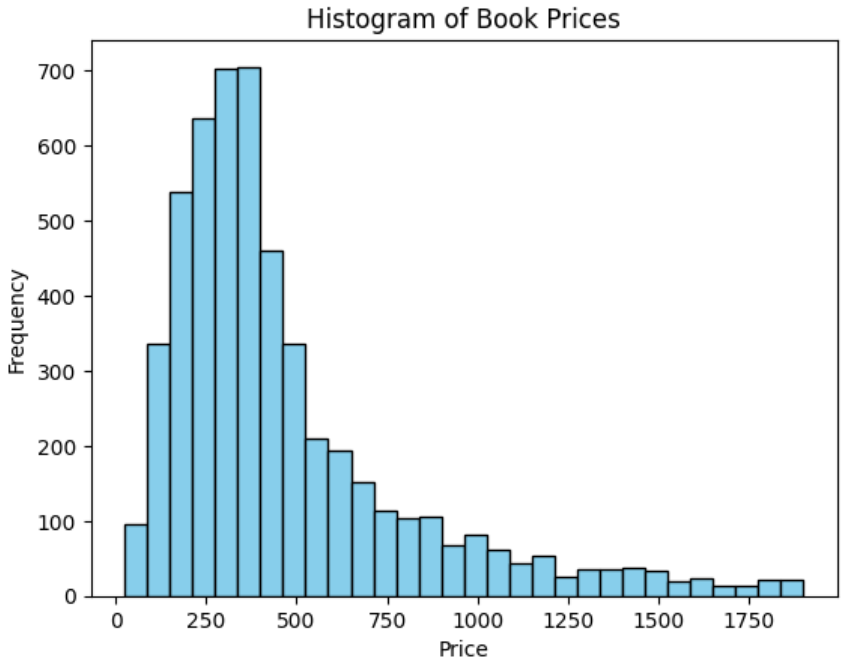


Figure 4.3: Histogram of Book Prices

It is easy to see that book prices are right-skewed and not normalized.

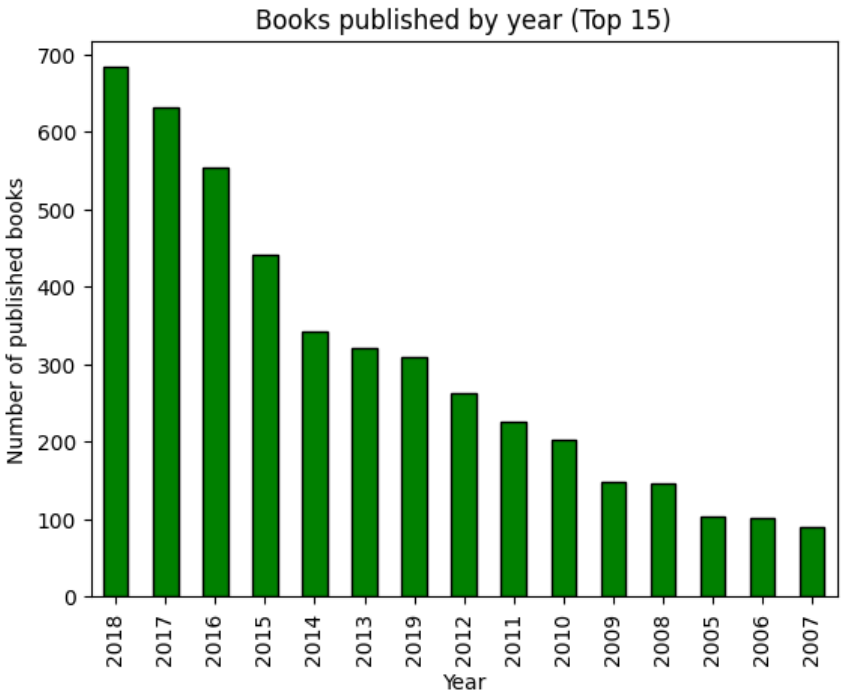


Figure 4.4: Bar Plot of Books Published by Year

This plot displays the top 15 years with the highest number of publications.

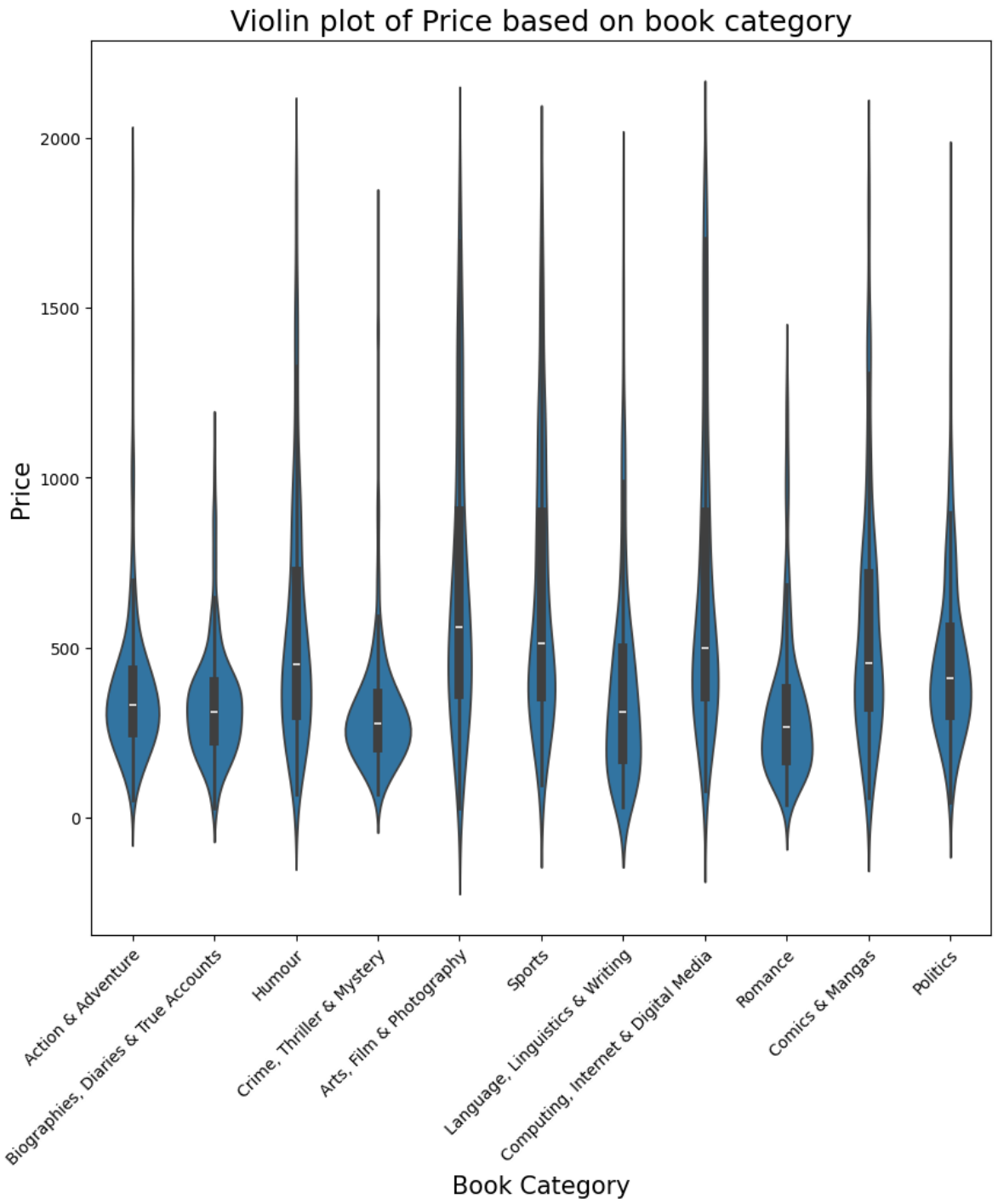


Figure 4.5: Violin Plot of Price based on Category

This plot visualizes the difference between book price distribution based on the book's category. Certain book categories (like Arts) have a broader price distribution with a higher average price, while others like romance are relatively normal.

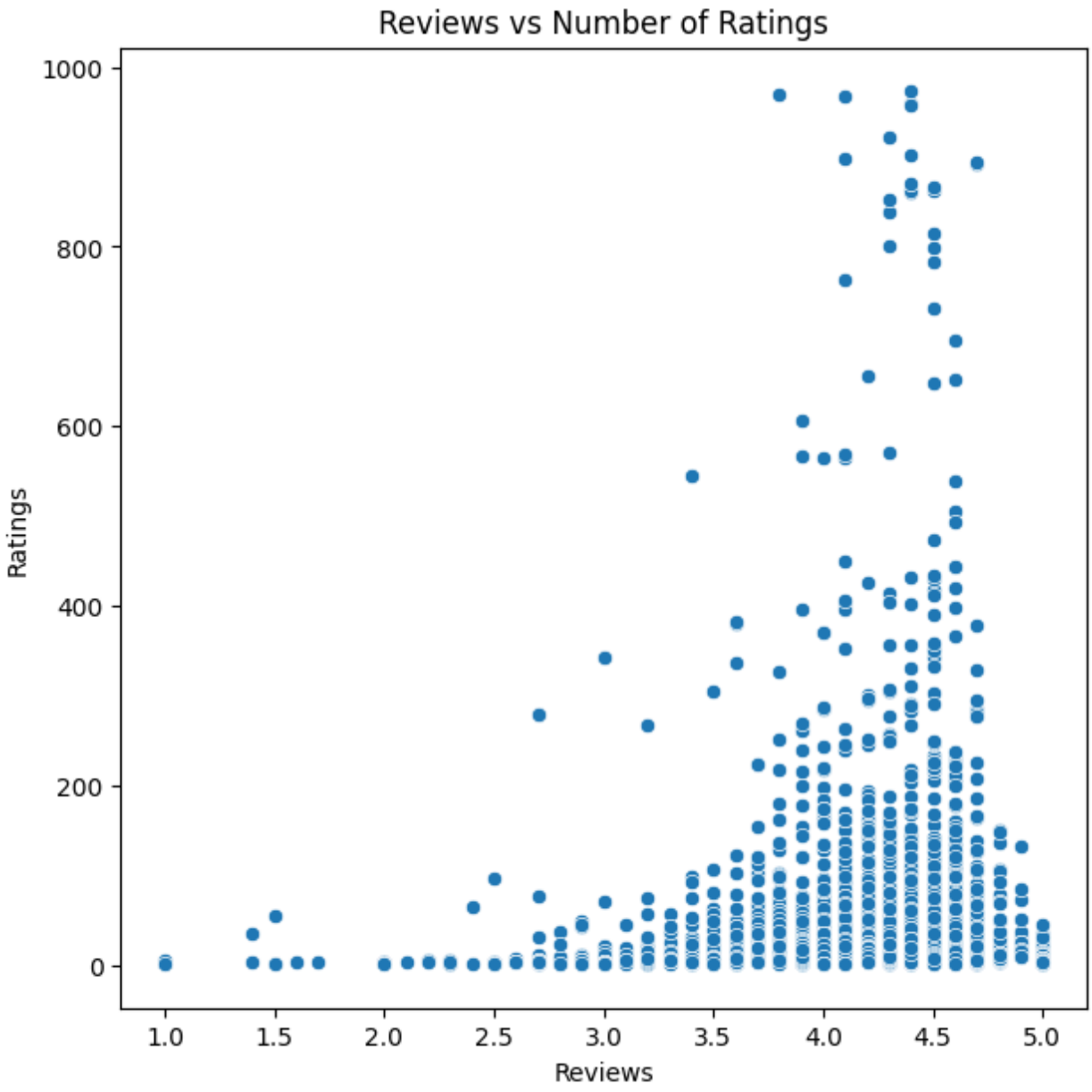


Figure 4.6: Scatter Plot of Reviews vs Number of Ratings

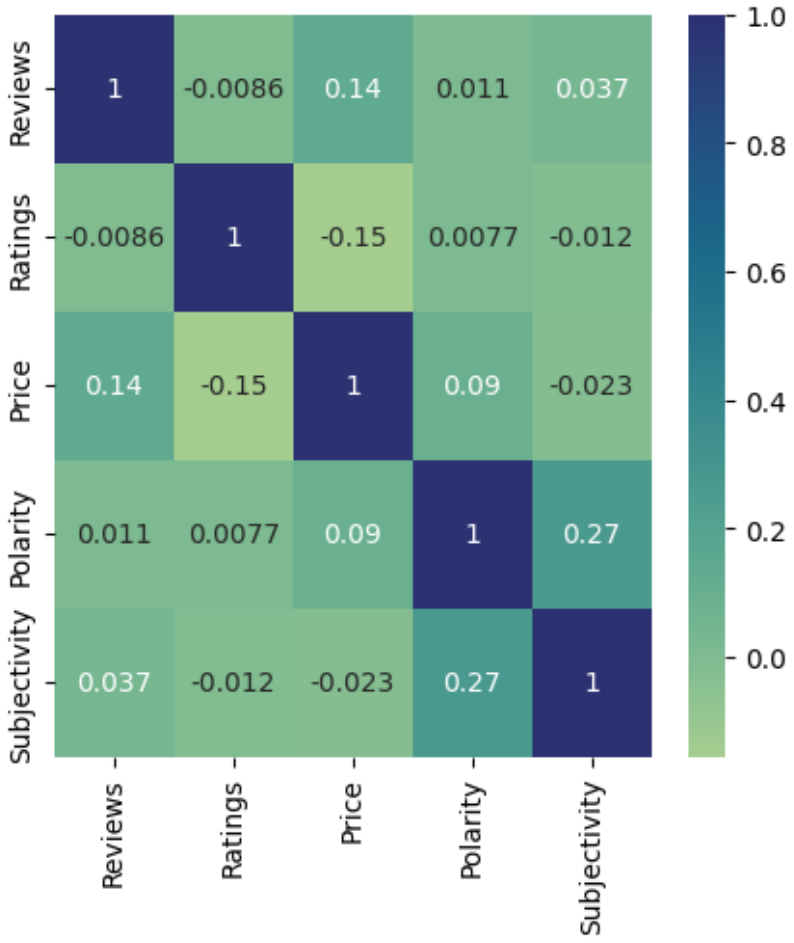
This scatter plot displays the relationship between the average review score of a book and the number of ratings it received. Interestingly, it appears that extremely high rating counts mostly belong to books with a +4 review score.

Figure 4.7: Heatmap of the Correlation Matrix

This heatmap visualizes the correlation between the various attributes of the dataset.

No strong correlation is displayed here, however the correlation between ‘Polarity’ and ‘Subjectivity’ was expected.

The negative correlation between ‘Ratings’ and ‘Price’ could potentially mean that the higher the price of a book is, the less ratings it receives because less people buy that book.



5. Model Results

The Model used in this research is a Random Forest Regressor. The input dataset is based on the following features:

'Reviews' - 'Ratings' - 'Year', 'Polarity' - 'Subjectivity' - ‘Keywords (Encoded)’ - ‘Author (Encoded)’ - CoverType (Encoded) - ‘BookCategory (Encoded)

The model will try to predict the book price based on these features.

Note: My model is prone to overfitting and so far, I have not been able to fix this issue. But increasing the data complexity even further might solve this problem.