The Next Great Reading Adventure:
Goodreads Data Report

1. Introduction

Outside of visual mediums, such as television or movies, books are one of the highest forms of entertainment. They can be informative with historical exploits, as well as delightful diversions from reality, and everything in-between. It is because of this versatile nature that booksellers such as Barnes & Nobles or Amazon offer such a wide selection of books that appeal to a large customer base. However, from the customer's perspective, choosing the next book to read can be difficult.

Booksellers and online retailers often use Co-oping as a strategy in order to advertise new works or books that have recently been released. This means publishing companies personally ask bookstores for preferential treatment of their books or series. However, when it comes to tailoring to individual users, this kind of advertising is not very effective. Given the fact that online retailers can see a user's previous purchasing history and maintain a better record of it than brick-and-mortar bookstores, there now exists the option to tailor recommendations specifically to individuals. This type of system could also be used in other websites such as Goodreads.com.

Within this project, using a dataset from Goodreads.com, we intend to create a content-based recommendation system that can be used to recommend books to different users based partially on popularity and average ratings. This would also make a good base for a future collaborative recommendation system that would use user data along with popularity in order to further improve predictive accuracy.

2. Dataset

The dataset was downloaded from Kaggle.com, from a CSV file that was created using Goodreads.com's API. The dataset contains a list of book titles along with the authors, isbn numbers, average ratings, total number of ratings given, and total number of text reviews for those books. There are over 13714 different entries for this dataset, with a few book titles that repeated themselves. The link to the website and the location of the dataset is here.

3. Data Cleaning

To begin, we looked over the dataset to see if there were any outstanding issues that might make further analysis difficult. Fortunately, there was no need for much cleaning since the dataset was relatively neat to begin with. There were only a few minor corrections that were done for clarity in order to tidy up the data further.

3a. Misaligned columns:

When trying to first read the CSV file into the Jupyter Notebook, it produced an error. It was then discovered that there were a few rows that were problematic. These rows did not line up well with the rest of the dataframe because they contained an extra column within each row. It was decided to skip these rows, since they were only 6 points in the data.

3b. Multiple Authors:

Looking at the authors column, it became clear that some books had more than one name attached to them. While some of these names were actually the illustrators for the books (such as with the case of Harry Potter), others were translators, voice actors for audiobooks, or actually co-authors of the book itself. The book Good Omens was written both by Terry Pratchet and Neil Gaiman, and so both could be attributed as the authors. Since it was unclear to what extent the use of authors could be in the future analysis, it was decided to split the authors' column into two separate columns: the primary author and the secondary author. The first author in the column was designated the primary author since they were more often attributed to the book in that row, and the secondary author was whoever remained. This would make any comparison by authors in the future more accurate since readers may be more interested in the primary author rather than the secondary.

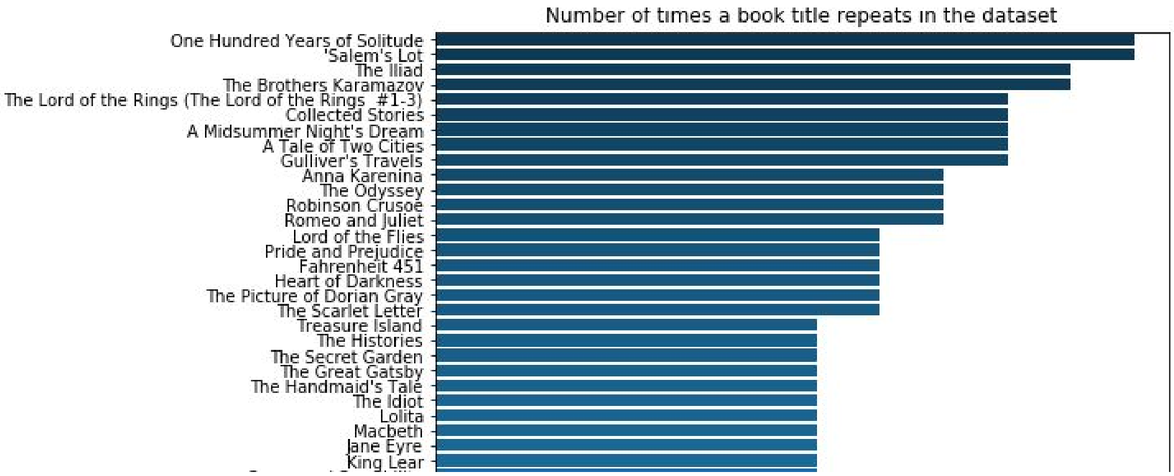
After these two problems were solved, the rest of the data was transferred over to a clean dataset and exported to a separate file in order to ensure accuracy.

4. Data Story

Certain columns of the data were unnecessary to explore because they would not typically be considered when searching for a book. These columns were the bookID and the isbn columns. Outside of those columns, the data was explored in depth.

4.1 Most Frequent Book Titles in the Dataset

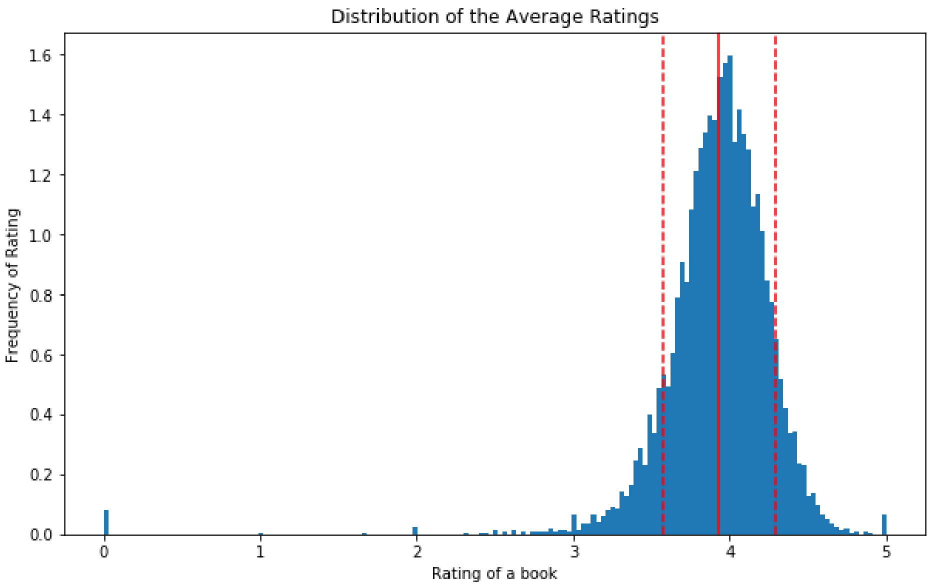
The first part of the data that was analyzed was the book titles. Since there were over 13700 book titles, we wanted to check if any repeated. As it turns out, there were 1292 repeated titles. Below is a snippet of the full a horizontal bar graph listing the most frequent titles and how often they repeated.



The most repeats for a title was 11 repeats, which is not too bad for the dataset. Furthermore, since these repeats were for potential different versions of a book, and since there were only 1292 out of 13 thousand samples, the titles were not removed. It did mean that there would have to be another way for the recommendation system to specify which book was being searched for in order to compare others to it.

4.2 Distribution of Average Ratings

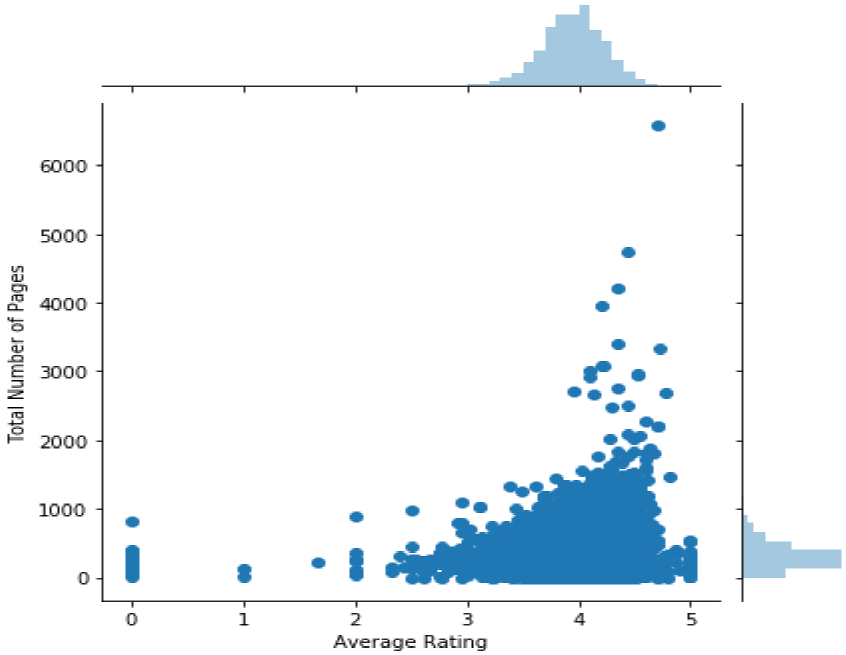
Next, we wanted to take a closer look at the distribution of the average ratings for all the books. Plotting this, we saw that most of the distribution fell pretty much between 3.6 and 4.2 for average rating. Determining the mean and standard deviation of the distribution was performed in the statistical analysis notebook. This is presented below with the solid line representing the mean and the dotted lines the standard deviation.



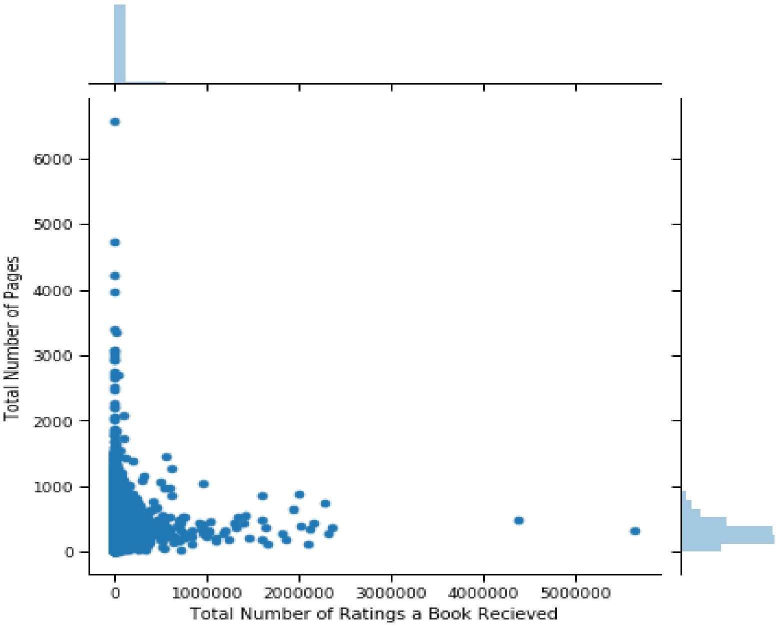
It was shown that the mean for the data was an average rating of 3.9, with a standard deviation of 0.3. From here, it was important to look at all the relationships between the rating of the books and other attributes, such as number of pages, total ratings given, and total text reviews.

4.3 Number of pages Vs. Rating

Initially, looking at the number of pages of a book and comparing it to its rating, there seemed to be a positive correlation: as the page number for a book increased, the ratings for it increased as well. The graph for this distribution is presented below.

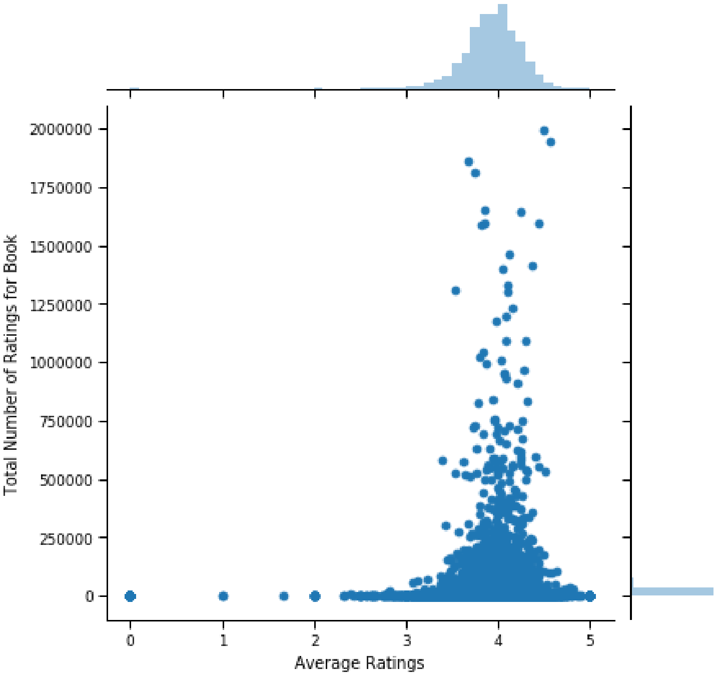


However, this correlation might have been because longer books are read less frequently than shorter ones, and so they actually might have had less total ratings. This would have weighted those ratings more heavily than for the shorter books. Following this logic, we checked a comparison of total number of pages against total number of ratings and discovered that as the page number increased, the total number of ratings was much lower.



4.4 Average Ratings Vs. Total Ratings

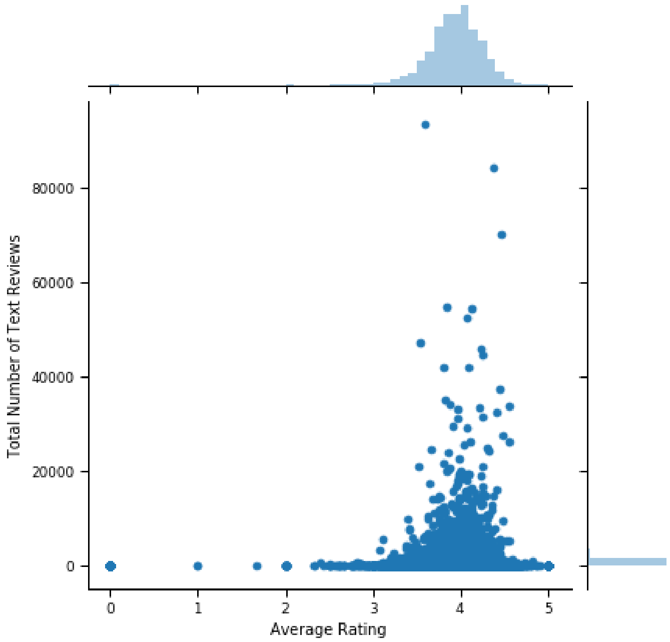
The next comparison made to the average ratings of the books were the count of how many ratings that book received. This was meant to check to see if books that received more ratings had better or worse ratings than others.



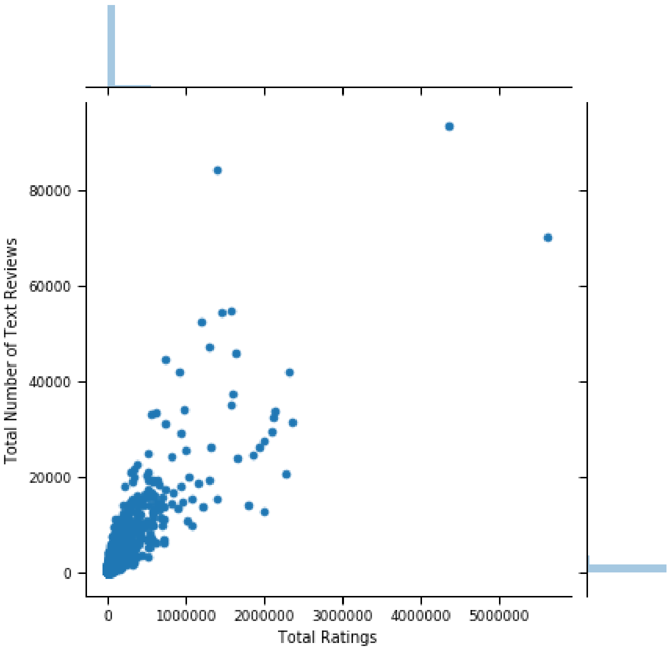
Looking above, one can see that there does seem to be a clumping of data around where the mean of the average ratings is located. This might mean that as the number of ratings for a book goes up, the rating for the book may be higher. However, we also see that books with very good ratings (those closest to 5) have less total number of ranking for them. Perhaps then there is a curve to the total number of rankings that indicate whether a book is of high quality or not based on how many have read it and ranked it.

4.5 Average Ratings Vs. Total Text Reviews

Following the total number of rankings for a book, we looked at the total number of text reviews. This showed the same distribution as for the total rankings given for a book, which makes sense that those who would rank the book may take time to write a text review for it.

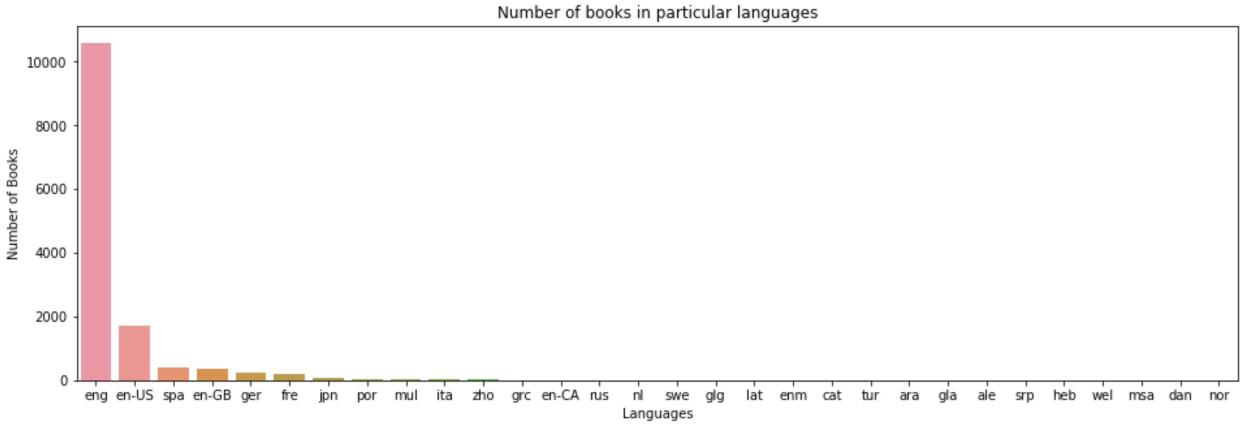


Another graph comparing both the ranking counts and text reviews below shows a clear positive between the two aspects of the books.

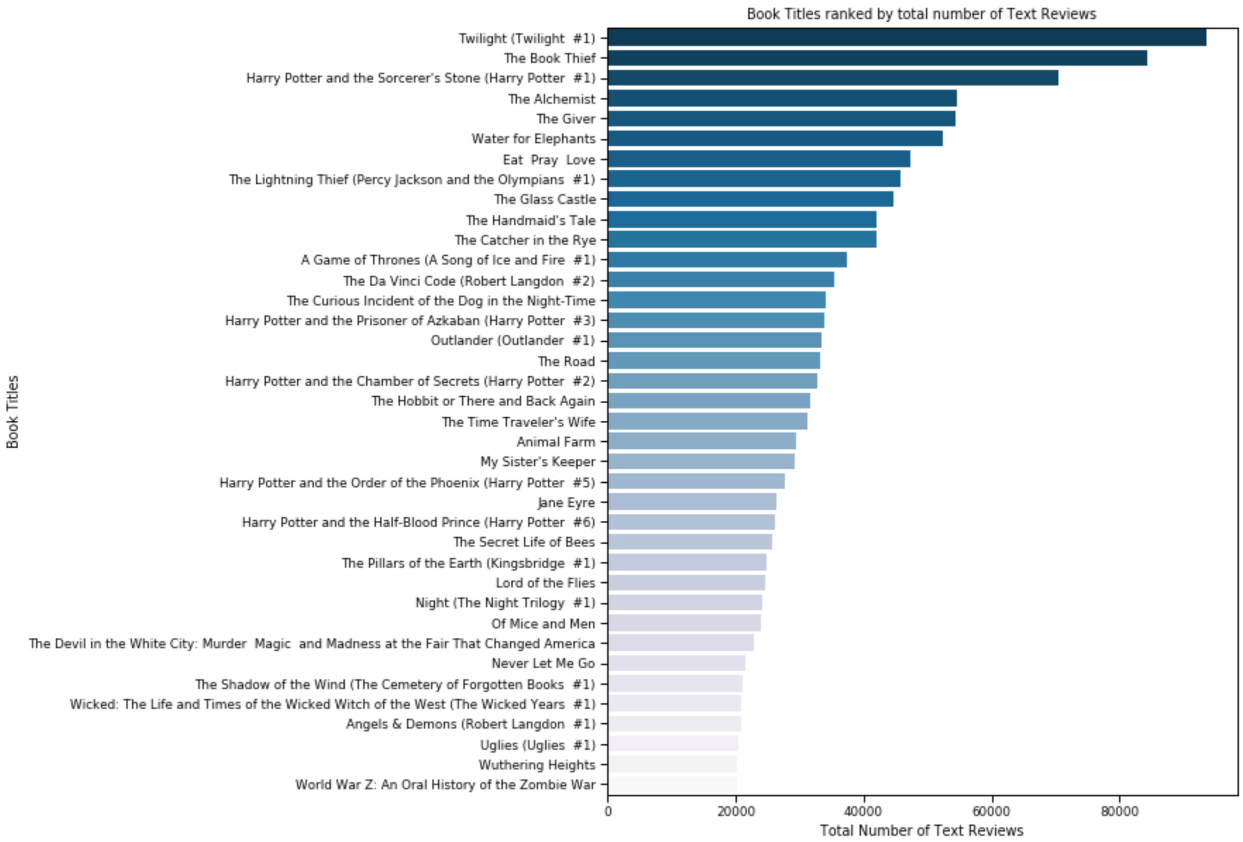


4.6 Final Exploratory Visualizations of Languages and Popularity

Finally, we explored the number of different languages that the books were written in, which you can see below is almost astoundingly english. This works very well with a recommendation system based predominantly for english reading audiences.



Also observed was book titles ranked by total number of text reviews.



Just from looking above we can see that many of the most popular books have been made into movies or television series, indicating that they have wide appeal to audiences.

5. EDA Conclusions

The conclusions drawn from the data story are noted below:

- The mean of the ratings falls around 3.9, meaning most of the rankings for this dataset fall around that area, with few books significantly lower, despite the option to vote for lower rankings.
- The ratings for the books have no strong correlation with any one characteristic of the data. The closest to being the most significant relationship for the data is with ratings and the number of rankings given.
- There is a positive correlation between total rankings and total text reviews.
- Almost all of the dataset are books written in english, and the most popular books are tied in with the popular culture of the time

Following these conclusions, it was determined that the first step in creating a recommendation system was to group the data into labeled groups based on their similarity in terms of ratings and popularity. After attaining their labeled groups, a recommendation system based on using the

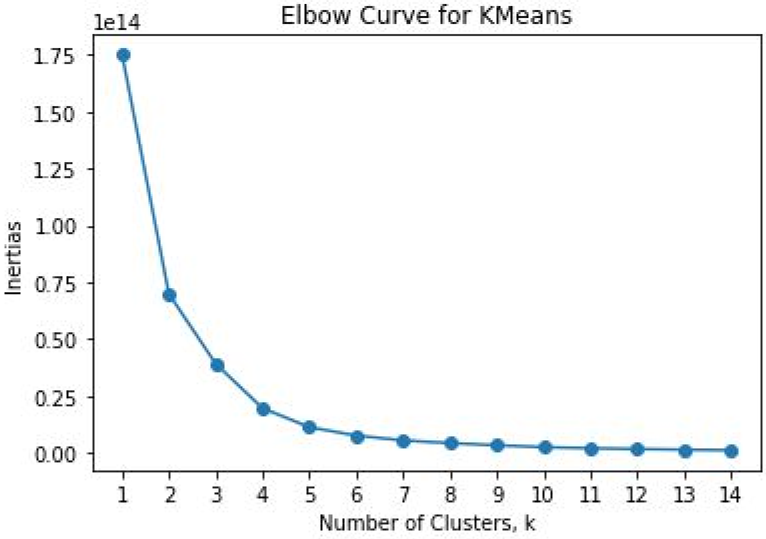
K-NearestNeighbor algorithm could be used to generate the top book titles that were similar to an input title.

4. In-depth Analysis with Machine Learning Algorithms

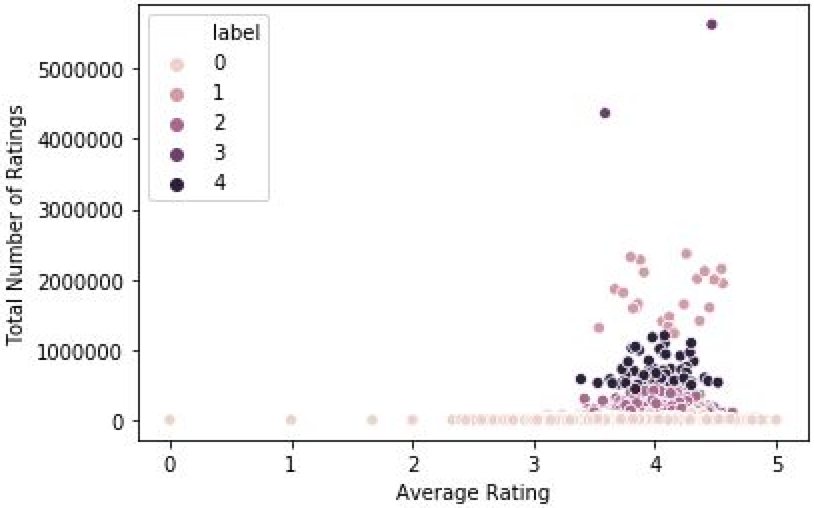
Based on the EDA and the statistical analysis, it was determined that in order to group the data into labeled categories, the most significant comparison was the average rating of a book and the total number of ratings it was given. Using these two categories as a guide we were able to cluster the data using the learning algorithm KMeans clustering. With KMeans, we could determine how many categories the dataset could be grouped into, and use these labels further down the line to determine similarity.

The first step was taking the entire dataset and just filtering the columns of average ratings, total ratings, and total review ratings. Since there was already a correlation between total ratings and total review ratings, this would only serve to increase the accuracy of the clustering. SVD (singular value decomposition) was used to reduce the dimensionality of the three features being looked at to filter only the significant ones.

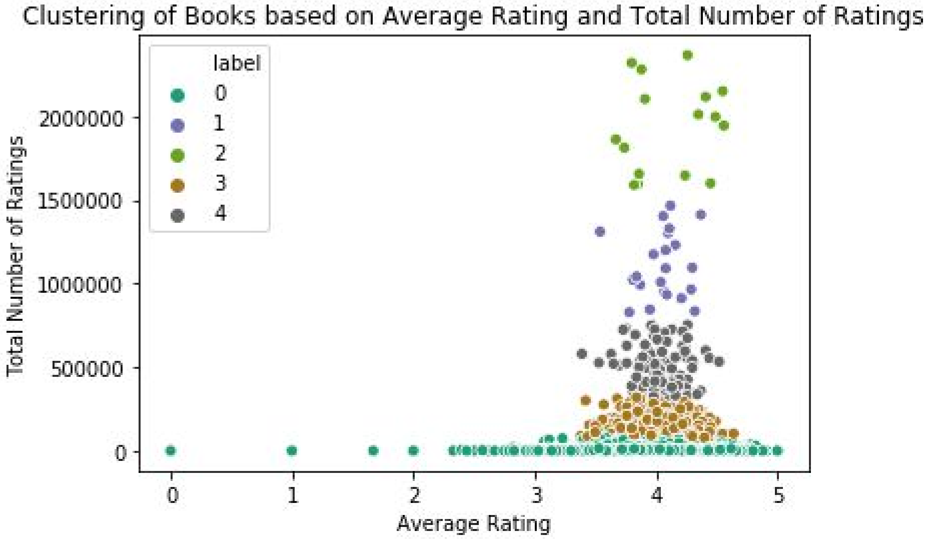
After this, an Elbow Curve was created to determine the value of clusters (k) to use with KMeans to figure out how many groups to best cluster the data.



Looking at the curve, it appeared that a k value of 5 seemed to be the best for clustering. Following this, a KMeans model with a k value of 5 was created for the entire dataset.



However, graphing the newly labeled data revealed that 2 of the books in the dataset were in their own labeled group. The outliers were then located and removed and clustering was repeated again.



The second attempt at clustering looked much cleaner with the groups having more books distributed among them.

With labeled data now, a K-NearestNeighbor model was created with the data and its labels in order to determine which titles were most similar to each other. A k value of 5 was used for this model since it was shown to consistently perform well with both training and testing data scores. From this model, the nearest neighbors could be acquired from one of the model's methods, and from the list of indices given, a function called Books_Recommended was created.

5. Results

The function Books_Recommended() worked by taking a title given, and returning the top 5 similar books to that book title. Even if a partial title was given, if there were at most 1 copy of the book within the dataframe, it returned a recommended list. A difficulty arose when multiple titles matched a given input. However, this was resolved by giving the user an error message with a list of possible titles and book IDs. The user could then use a book ID to match the book of interest, and the return value would print the title of the book matching the book ID and the top 5 books recommended.

6. Next Steps

For future systems, I would try to find a way to incorporate user data, using a collection of Goodreads accounts to determine what books they should read next. Furthermore, I would like to find a way to incorporate authors into recommendations as well, since people tend to read multiple works written by the same author. Lastly, I think a more updated dataset from Goodreads.com's API would be more relevant to recommending the latest popular books being read. There may even be a way to incorporate current popularity based on time series data of when and how often a rating for a book is received.