Problem Statement

In recent decades, with the rise of platforms like YouTube, Amazon, Netflix, and other similar web services, recommender systems have steadily become a more prominent part of our daily lives, shaping the way we discover and interact with content and products.

With the #booktok blowing up, many people have been buying up books - either physical or digital copies- thanks to book-influencers. After reading a book, readers want to read something new and they want to know which book will satisfy them.

An online e-reader is hoping to profit off this book blowup. They want to incorporate a recommendation system based on user ratings to offer the top 10 books that they may enjoy reading.

Data Wrangling and Cleaning

The dataset was found on Kaggle, it was composed of 3 datasets: users, books, and ratings. The dataset was collected by Cai-Nicolas Ziegler from the Book-Crossing community.

Once the dataset was loaded in the original shape of the book dataframe was 271360 rows and 8 columns. The original users dataframe had the shape of 278858 rows and 3 columns and finally the ratings dataframe had 1149780 rows and 3 columns. Then 3 columns from the book Dataframe was deleted, Image_url_s, image_url_m and image_url_l. after deleting those columns, the three dataframes were merged together. Once a merge dataset, I checked for null values. Only 3 columns had missing values, Age = 277835, Book-Author = 2, Publisher = 2. Since age was more than 10% missing, the column was dropped. As a precaution, any duplicate rows were dropped.

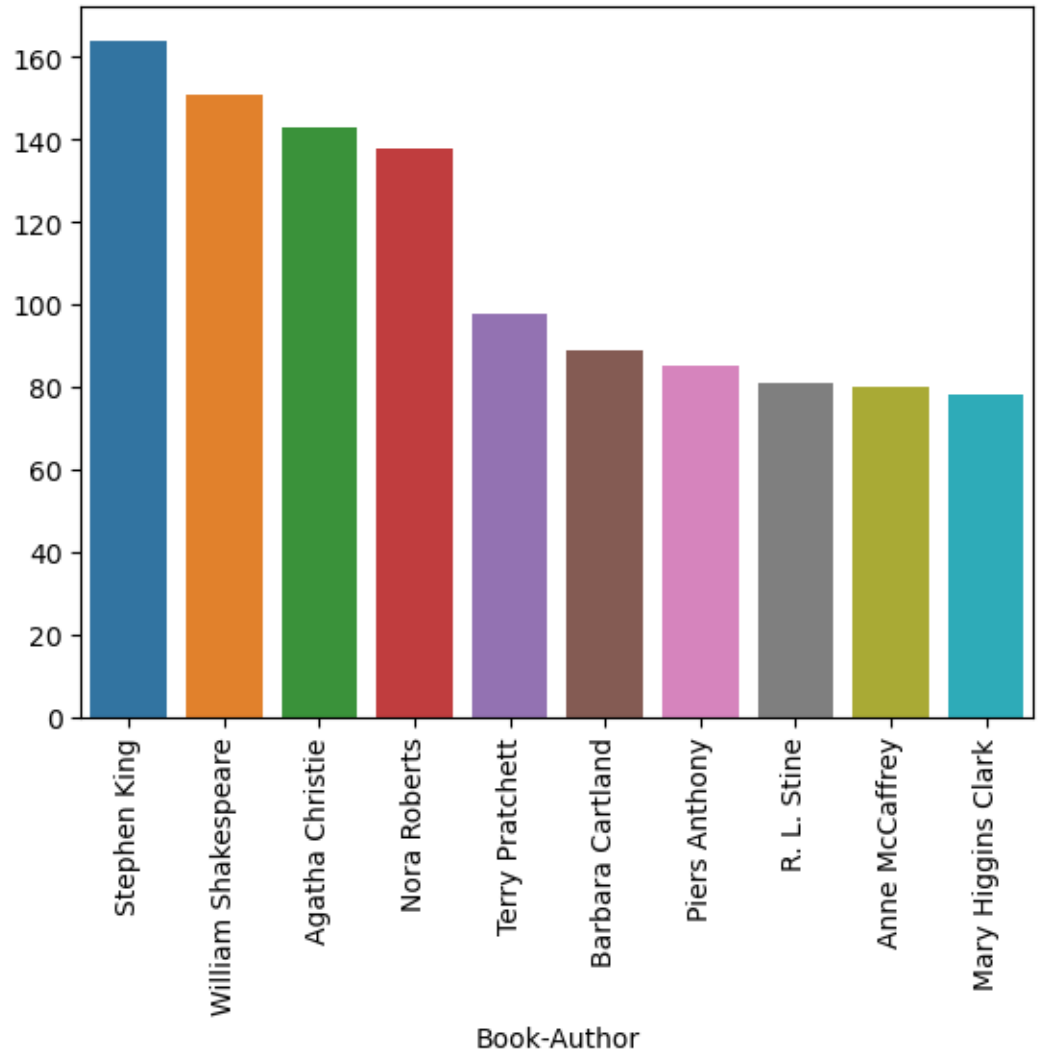
Then it was important to separate the explicit ratings from the implicit ratings. The implicit ratings were the rows that had a rating score of 0, the explicit were those that had a rating between 1-10. Since the dataset was quite large, a random sample of 100k was taken. The sampled Dataset was saved and we move on the the EDA.

Exploratory Data Analysis

We load in the sampled data and start to ask our questions

Which Authors had written the most books?

Using a groupby and nunique and sorting values from highest to lowest, the authors with the most written books was revealed.

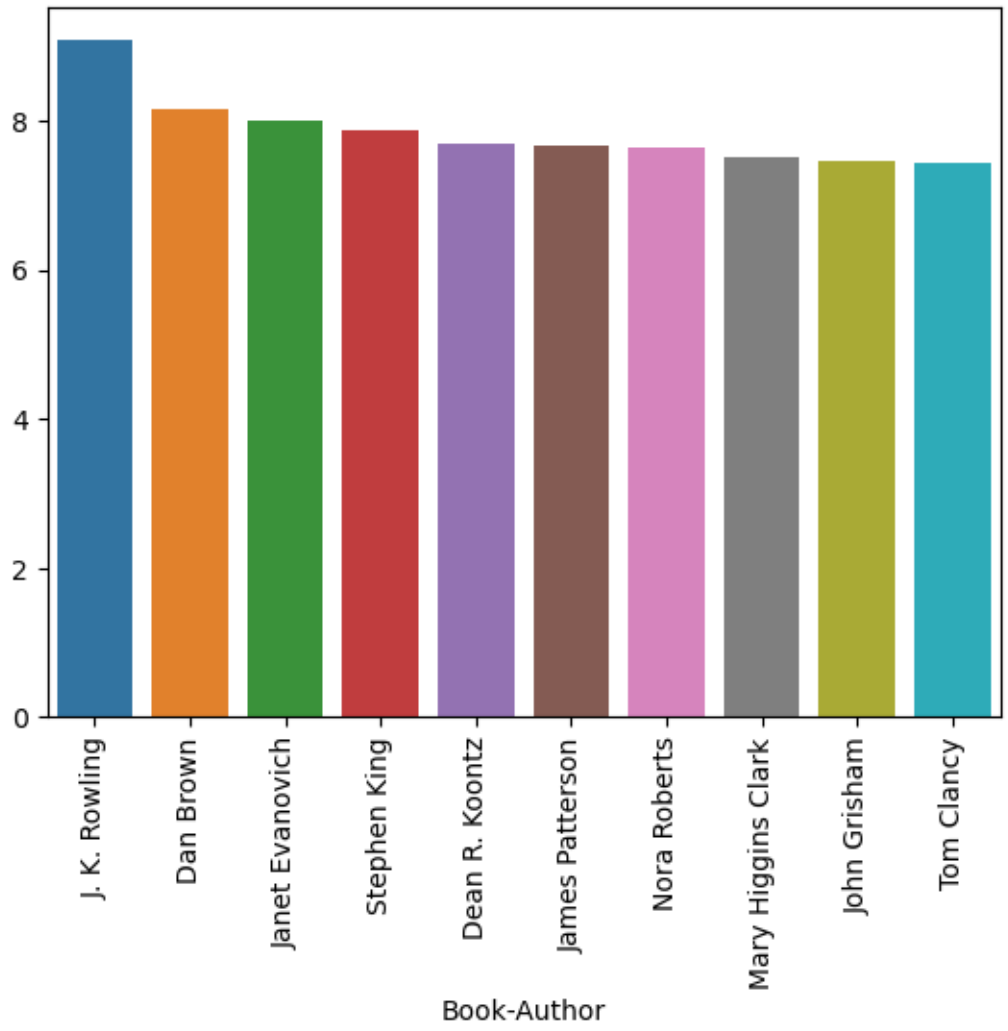


From our given dataset, Stephen King wrote 164 books, William Shakespeare wrote 151 books, and Agatha Christie wrote 143 books.

Which Authors are considered the most popular based on ratings?

To find the authors with the highest ratings ratings, the dataset was grouped by author and the book_ratings were summed, the values were sorted from highest to lowest sum. After that we take the names that were in the top ten and find them within the dataset and save it to pop_author_names. Finally the we group pop_author_names by the author and

aggregate the book rating mean.



As we can see J.K Rowling had the highest mean ratings with a 9.07. Dan brown had a mean rating of 8.16, Janet Evanovich had a rating of 8.01 and the rest were somewhere in the 7.8 to 7.4 range.

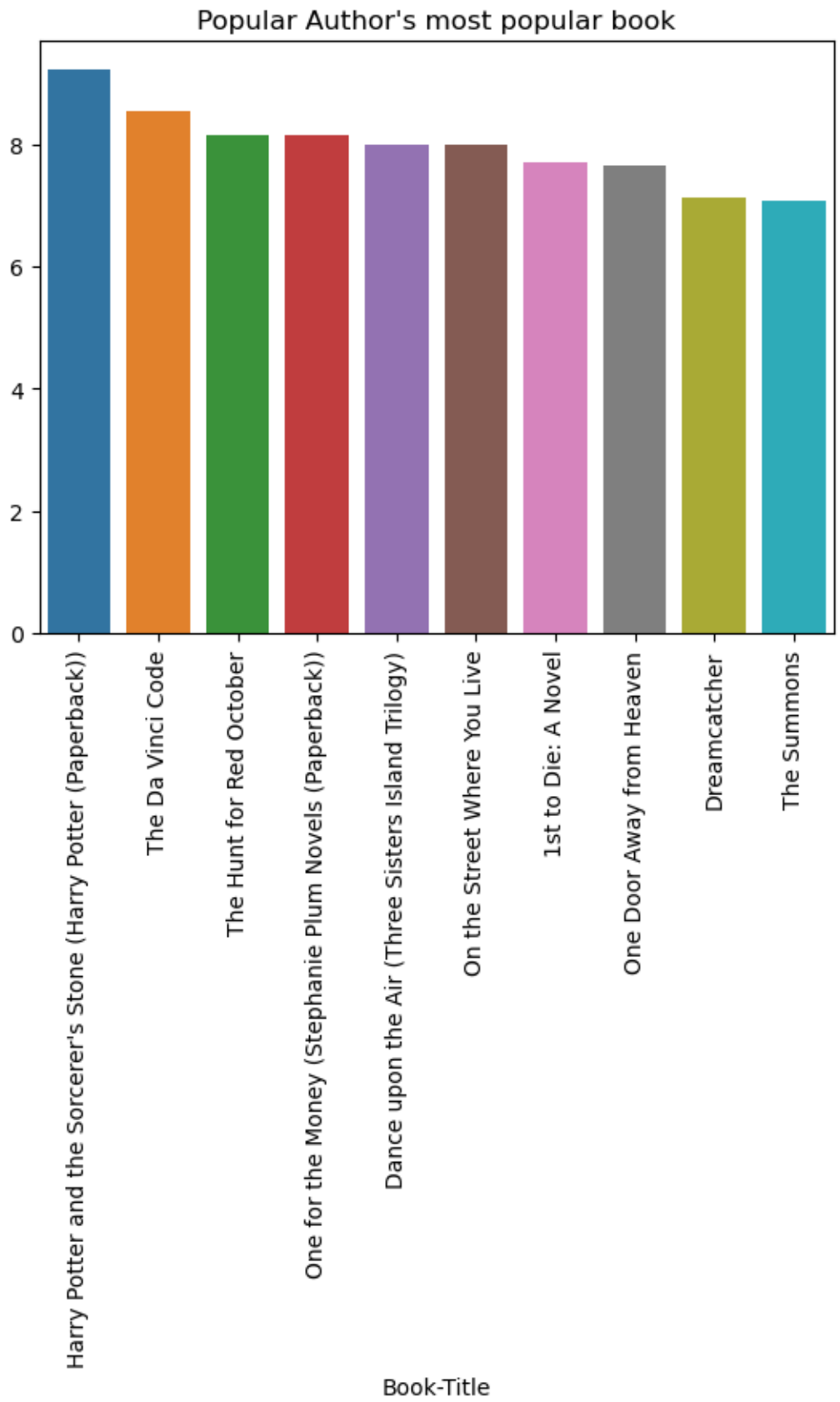
Based on the highest Rated authors, Which book is their most popular?

Finding the most popular author's most popular book was relatively easy, as all we needed was the pop_author_names. From that dataset, the data was grouped by the author and book title and we got the aggregate sum value of book_rating. We sort two values, one

descending and one ascending, Book-author and book rating respectively.

	Book-Author	Book-Title	Book-Rating
7	Dan Brown	The Da Vinci Code	1186
26	Dean R. Koontz	One Door Away from Heaven	298
70	J. K. Rowling	Harry Potter and the Sorcerer's Stone (Harry P...	793
77	James Patterson	1st to Die: A Novel	485
133	Janet Evanovich	One for the Money (Stephanie Plum Novels (Pape...	301
189	John Grisham	The Summons	537
238	Mary Higgins Clark	On the Street Where You Live	232
297	Nora Roberts	Dance upon the Air (Three Sisters Island Trilogy)	184
461	Stephen King	Dreamcatcher	378
618	Tom Clancy	The Hunt for Red October	236

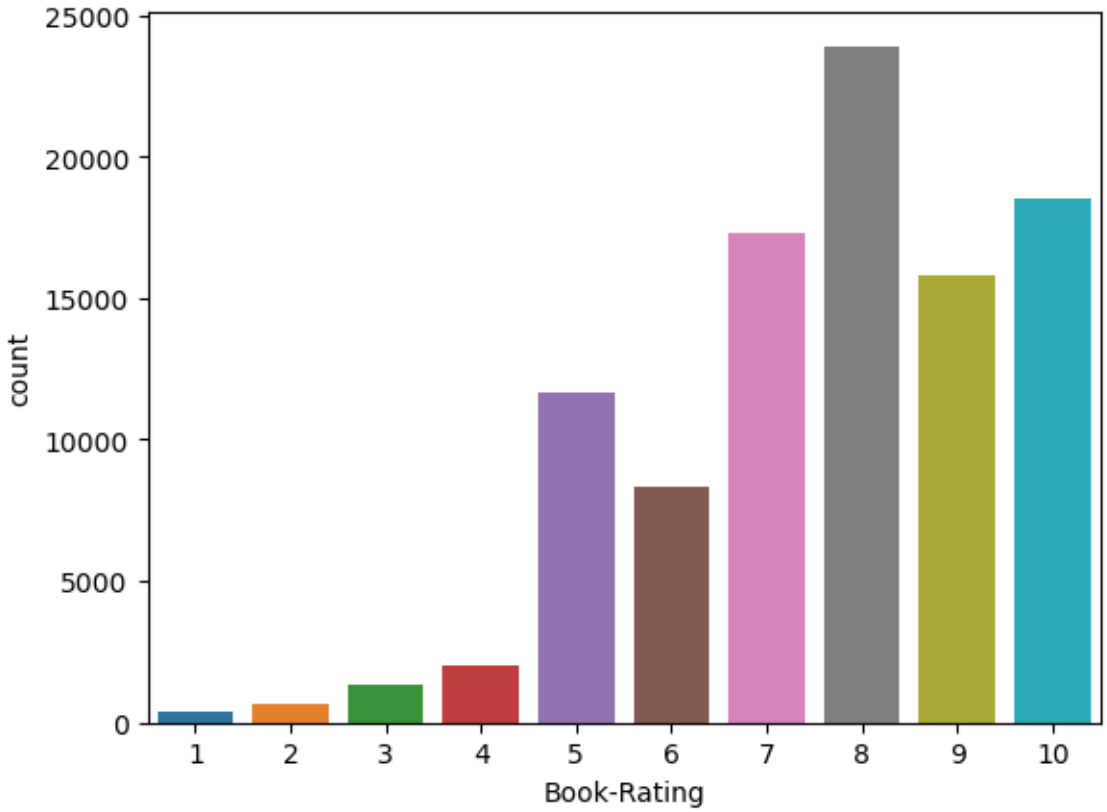
We then take the book titles and find the mean ratings from the users.


Popular Author's most popular book
0
2
4
6
8
Harry Potter and the Sorcerer's Stone (Harry Potter (Paperback))
The Da Vinci Code
The Hunt for Red October
One for the Money (Stephanie Plum Novels (Paperback))
Dance upon the Air (Three Sisters Island Trilogy)
On the Street Where You Live
1st to Die: A Novel
One Door Away from Heaven
Dreamcatcher
The Summons
Book-Title

Harry potter had a mean rating of 9.2. and The Summons (which was number 10) had a score of 7.06.

Most common Ratings for books?

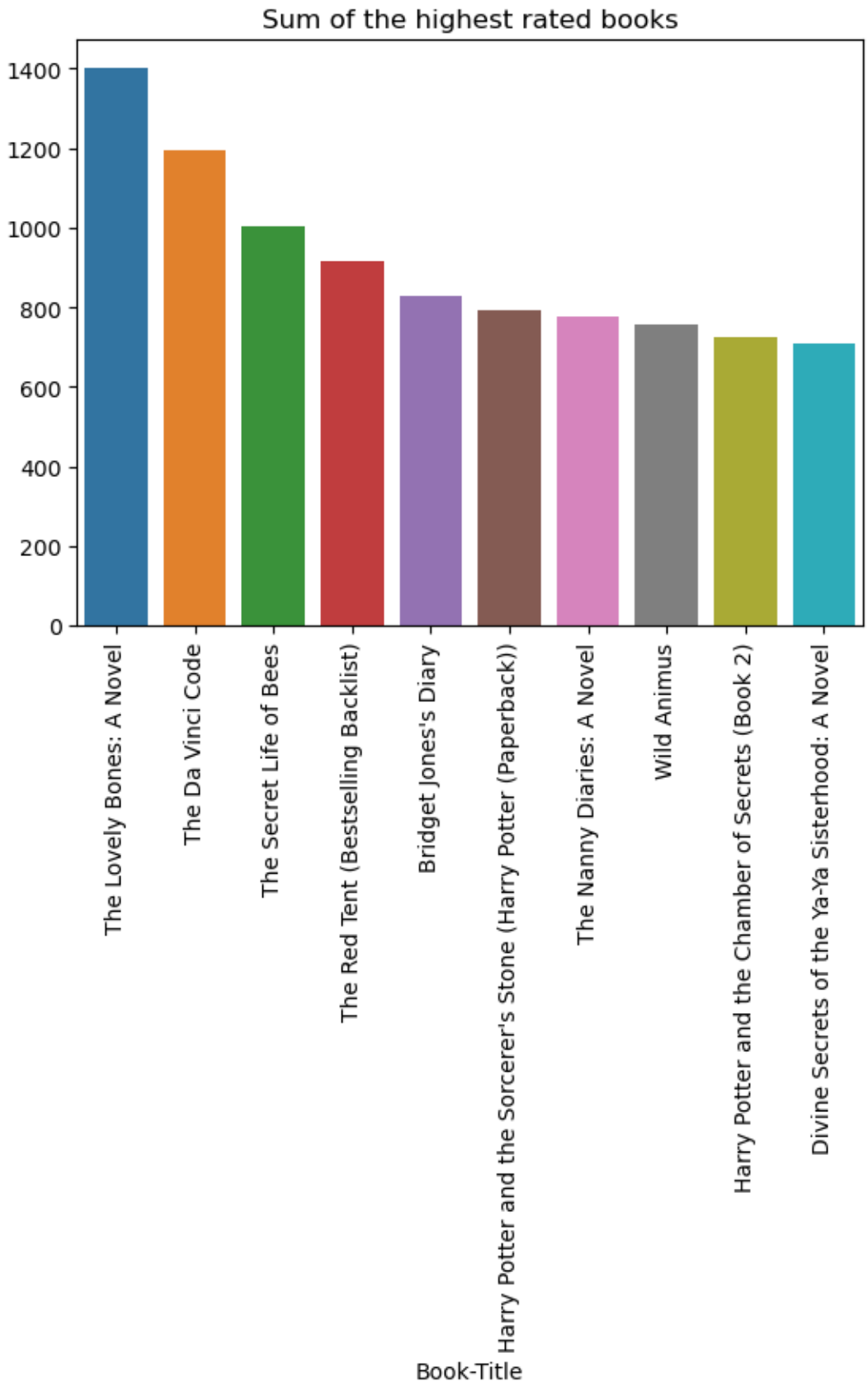
All that is needed to get the common ratings for books is getting the distribution of the Book_ratings column.



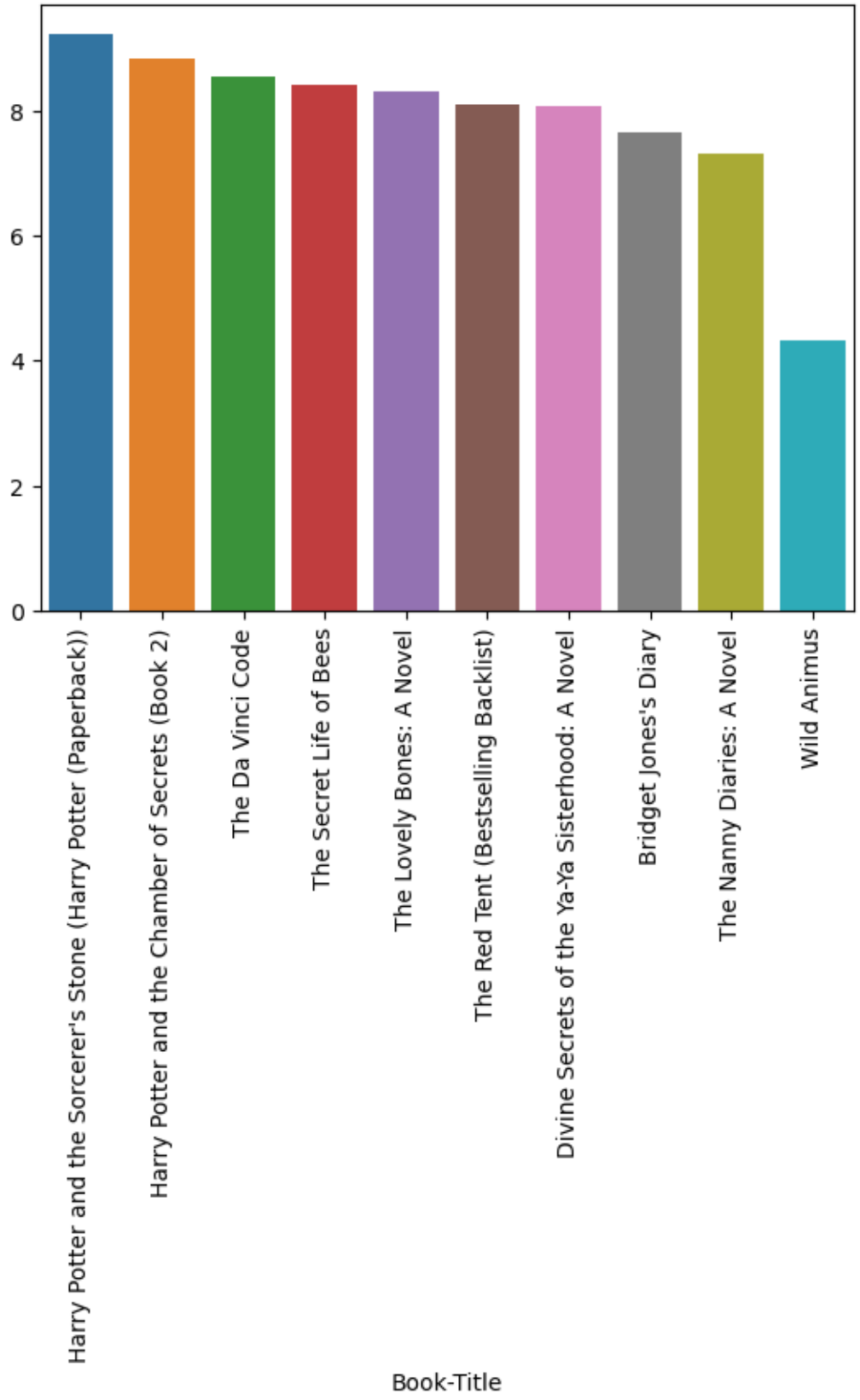
Most ratings were 7 and above.

Which books are considered popular by Rating

From the original dataframe, we group the data by book title and we get the sum of the book_ratings.


Sum of the highest rated books
1400
1200
1000
800
600
400
200
0
The Lovely Bones: A Novel
The Da Vinci Code
The Secret Life of Bees
The Red Tent (Bestselling Backlist)
Bridget Jones's Diary
Harry Potter and the Sorcerer's Stone (Harry Potter (Paperback))
The Nanny Diaries: A Novel
Wild Animus
Harry Potter and the Chamber of Secrets (Book 2)
Divine Secrets of the Ya-Ya Sisterhood: A Novel
Book-Title

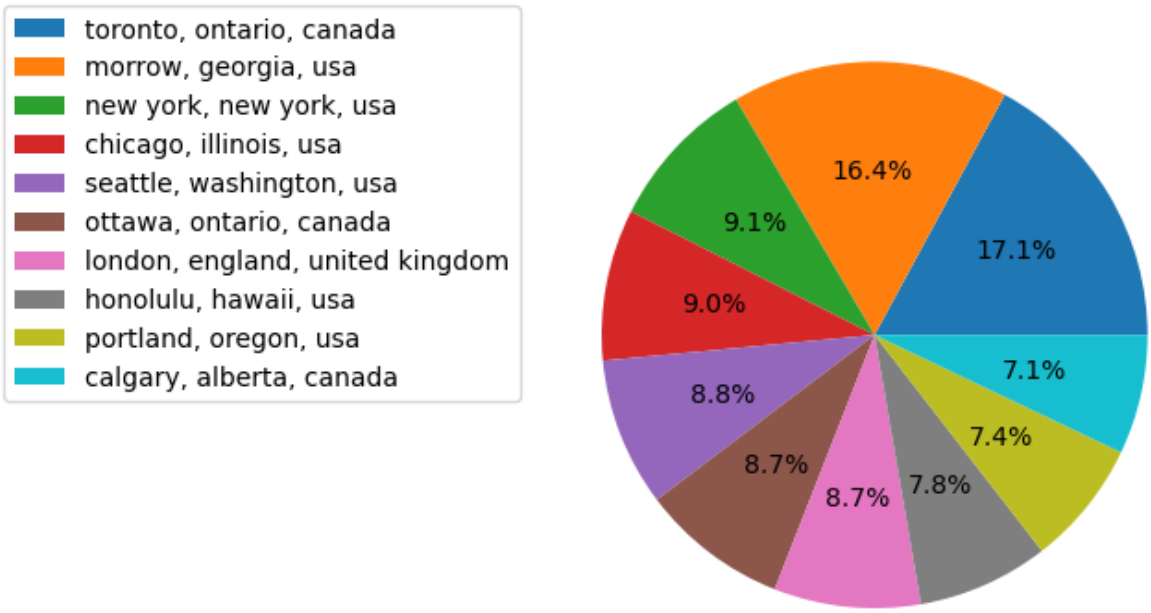
Then we get the mean of the ratings per book.



While lovely bones had the highest sum ratings, the actual mean ratings show that both Harry Potter books, The Davinci Code and The Secret Life of Bees had higher mean ratings, with ratings of 9.2, 8.8, 8.5, and 8.4.

Location where most users are from?

Simply use value counts on the Location column on the sampled datset.



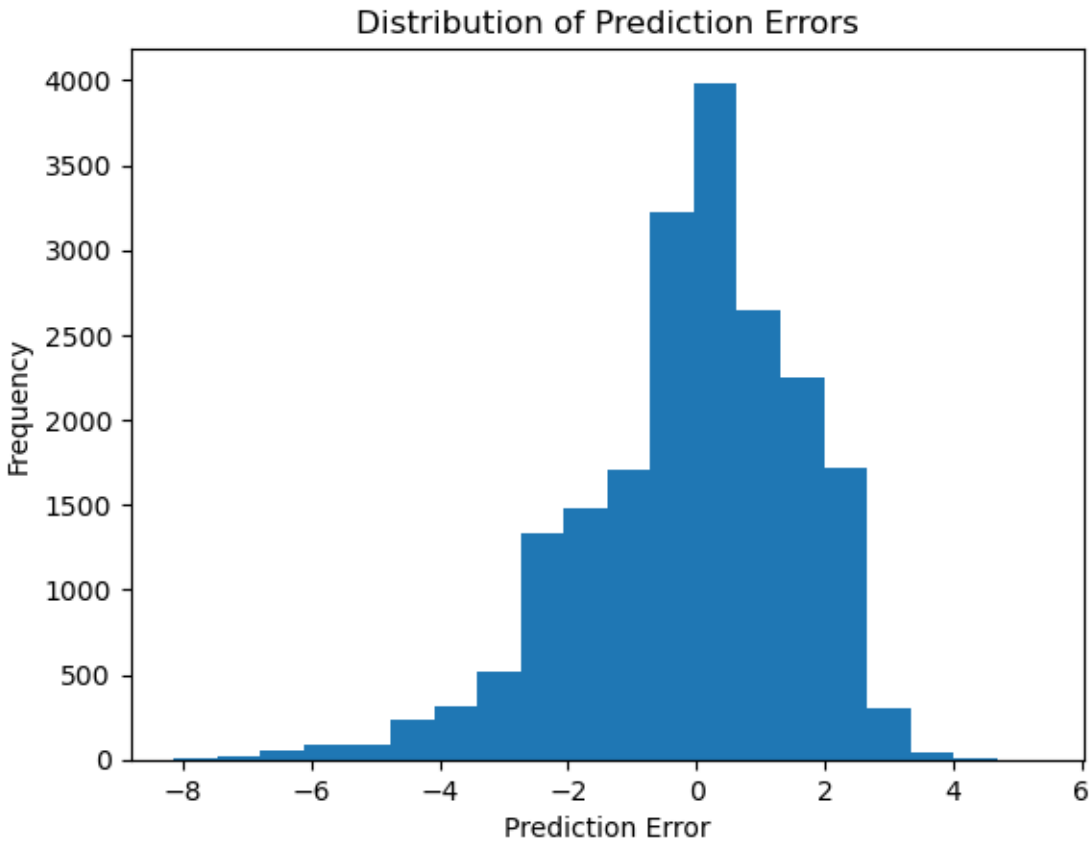
Most readers are from the USA, with a 58.5% the most readers in the USA being from Morrow, Georgia. Canada had 32.9% with Toronto having the most readers.

Preprocessing and Modelling

To begin the preprocessing process, we create a scikit-surprise reader to create a surprise object. The only columns in the surprise object is user_ID, ISBN and Book-rating. The data is then split using a 80/20 split to use in a svd algorithm. I used the metrics RMSE and MAE to check performance. RMSE = 1.7034 and MAE = 1.3206, these metrics didn't seem great so it's time to hypertune the model. Using a RandomizedSearchCV , parameters such as lr_all, n_factors, reg_all and n_epochs were tested and returned the best parameters.

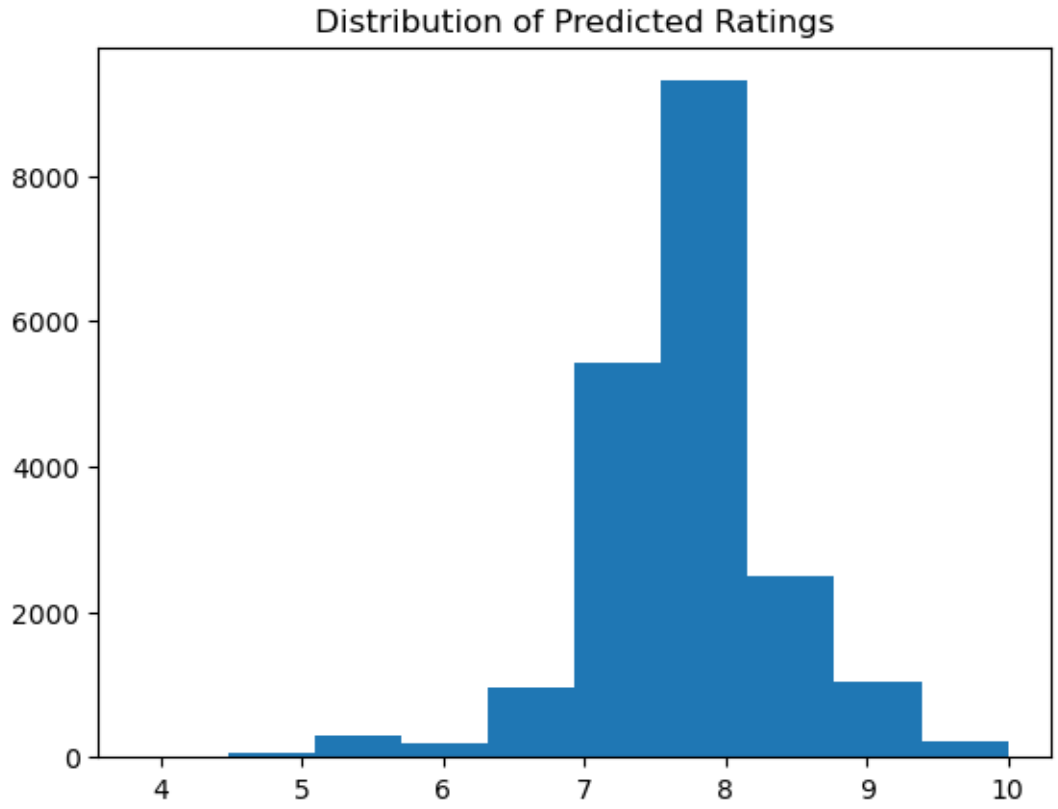
The best parameters were lr_all = .002, n_factors = 50, reg_all = 0.03 and n_epochs = 50.

The model was updated and refit. The metrics this time were RMSE =1.3549 and MAE = 1.0401. Let's check the range of predicted errors and predicted ratings



The X range from -8 to 6 means that there are some cases where the predictions were significantly off either overestimating or underestimating, however given that there's a bell centered around 0 suggests that most predicted errors are small which means the model

generally accurate.



It appears that most predictions have a score of 7 and 8, which based on what we've seen for common book ratings makes sense.

Moving on, it's time to get the recommendations. First we get all the unique ISBN, then the predicted rating for each user is calculated, giving estimated ratings for each ISBN by the specified user. Then the predictions are sorted in descending order giving us the highest predicted rating first.

Here's an example user and their recommended books

```
Top 10 Recommendations for user 154811:
Harry Potter and the Prisoner of Azkaban (Book 3): Predicted Rating: 8.93
The Lion, the Witch, and the Wardrobe (The Chronicles of Narnia, Book 2): Predicted Rating: 8.84
Harry Potter and the Sorcerer's Stone (Harry Potter (Paperback)): Predicted Rating: 8.84
Ender's Game (Ender Wiggins Saga (Paperback)): Predicted Rating: 8.83
The Perks of Being a Wallflower: Predicted Rating: 8.71
To Kill a Mockingbird: Predicted Rating: 8.66
Harry Potter and the Sorcerer's Stone (Book 1): Predicted Rating: 8.64
Griffin &amp;amp; Sabine: An Extraordinary Correspondence: Predicted Rating: 8.62
Harry Potter and the Chamber of Secrets (Book 2): Predicted Rating: 8.61
Lonesome Dove: Predicted Rating: 8.59
```