CSCI 57300: Data Mining Fall 2022

Final Project Report

The Oracle of Delphi: A recommendation system for the best books ever

(Recommendation system based on collaborative filtering)

Team members:

Arshia Joshi

Abhinav Asthana

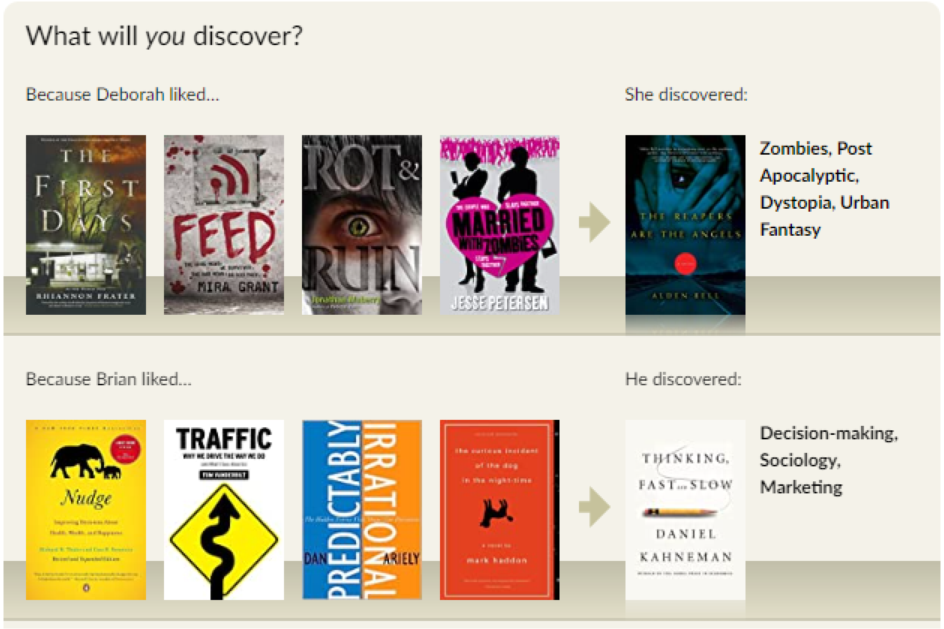
INTRODUCTION

For our data mining project, we wanted to analyze Goodreads book data set and based on the inference, we then wanted to make recommendations.

Why Goodreads?

Goodreads is a website and mobile app that allows users to discover, rate, and review books. It was founded in 2007 and is now owned by Amazon. Users of Goodreads can create a virtual bookshelf, add books to it, and rate and review books they have read. The website also has a recommendation system that suggests books to users based on their past ratings and reviews.

In addition to its review and recommendation features, Goodreads also has a number of other features for book lovers. Users can join groups and participate in discussions about books, set reading goals and track their progress, and participate in reading challenges. The website also has a "Want to Read" feature that allows users to create a list of books they want to read in the future. Goodreads is a popular resource for finding new books to read and connecting with other book lovers. It is also a useful tool for authors and publishers to promote their books and connect with readers.

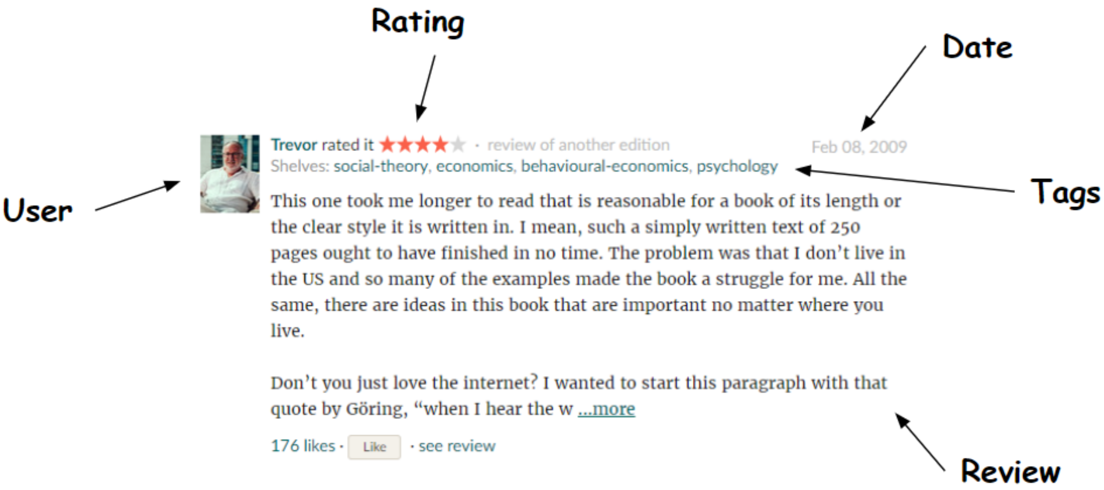


What will *you* discover?

Because Deborah liked... She discovered: Zombies, Post Apocalyptic, Dystopia, Urban Fantasy

Because Brian liked... He discovered: Decision-making, Sociology, Marketing

Anatomy of Goodreads review:



Trevor rated it ★★★★☆ · review of another edition Feb 08, 2009
Shelves: social-theory, economics, behavioural-economics, psychology

This one took me longer to read that is reasonable for a book of its length or the clear style it is written in. I mean, such a simply written text of 250 pages ought to have finished in no time. The problem was that I don't live in the US and so many of the examples made the book a struggle for me. All the same, there are ideas in this book that are important no matter where you live.

Don't you just love the internet? I wanted to start this paragraph with that quote by Göring, "when I hear the w ...more

176 likes · Like · see review

Why do we need recommendation systems?

Recommender systems are systems that help users discover items they may like. They help users in discovering new books, movies, products, and help companies by promoting relevant products to prospective customers.

The Goodreads book dataset is a collection of data about books and ratings from the Goodreads website. It can be used to build recommendation systems or to analyze trends and patterns in book ratings and reviews. The Goodreads book dataset includes information about the books themselves, such as the title, author, publisher, and genre, as well as ratings and reviews from users. It also includes information about the users themselves, such as their location and the books they have rated and reviewed.

The dataset can be used to build a recommendation system in a number of ways. For example, you could use collaborative filtering to make recommendations to users based on the ratings and reviews of similar users. You could also use the data to analyze trends in book ratings, such as which genres are most popular or which authors have the highest rated books.

Overall, the Goodreads book dataset is a rich resource for anyone interested in building recommendation systems or studying trends in book ratings and reviews.

In our project we used two different approaches: Popularity – based Recommendation as well as Collaborative filtering approach to analyze and make suitable recommendations.

A ***population-based recommendation system*** is a type of recommendation system that makes recommendations based on the overall preferences of a large group of users, rather than the individual preferences of a single user. This approach is often used in situations where there is a large dataset of user ratings or preferences, and the goal is to identify items that are popular or well-liked by a broad audience. In our project, we use a population-based recommendation system based on metrics like ratings and review count.

Collaborative filtering is a technique used to make recommendations based on the preferences of a group of users. It works by analyzing the past behavior of a group of users and identifying items that are highly rated by similar users.

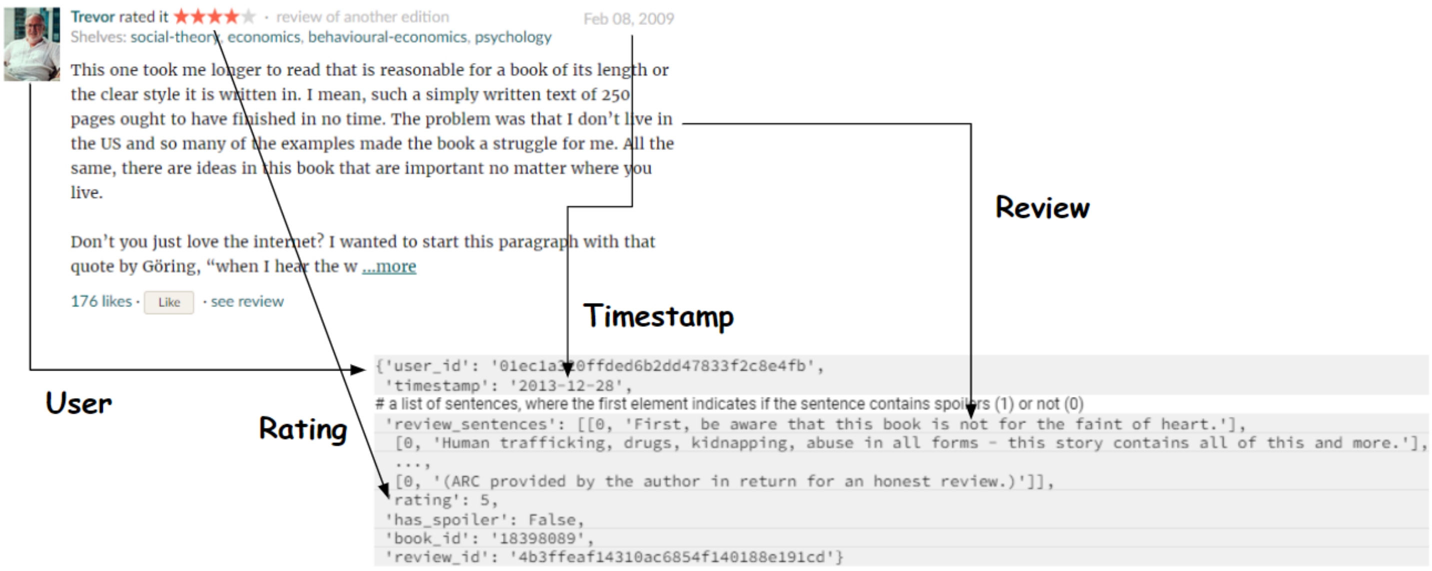
There are two main types of collaborative filtering: user-based and item-based. In user-based collaborative filtering, recommendations are made based on the preferences of similar users. For example, if Alice and Bob both rated a particular movie highly, and Alice also rated a second movie highly, the system might recommend the second movie to Bob.

In item-based collaborative filtering, recommendations are made based on the similarity between items. For example, if Alice and Bob both rated a particular movie highly, and the movie is similar to a second movie that Alice also rated highly, the system might recommend the second movie to Bob.

Collaborative filtering is a popular approach to recommendation systems because it does not require any information about the items being recommended, only the past behavior of users. This makes it easy to implement and scalable, as it does not require any upfront knowledge about the items being recommended.

Dataset Analysis

The dataset as shown above in introduction, is in fact a nested list of details about each book. In our project, we are considering the following two datasets:



Trevor rated it ★★★★☆ · review of another edition Feb 08, 2009
Shelves: social-theory, economics, behavioural-economics, psychology

This one took me longer to read that is reasonable for a book of its length or the clear style it is written in. I mean, such a simply written text of 250 pages ought to have finished in no time. The problem was that I don't live in the US and so many of the examples made the book a struggle for me. All the same, there are ideas in this book that are important no matter where you live.

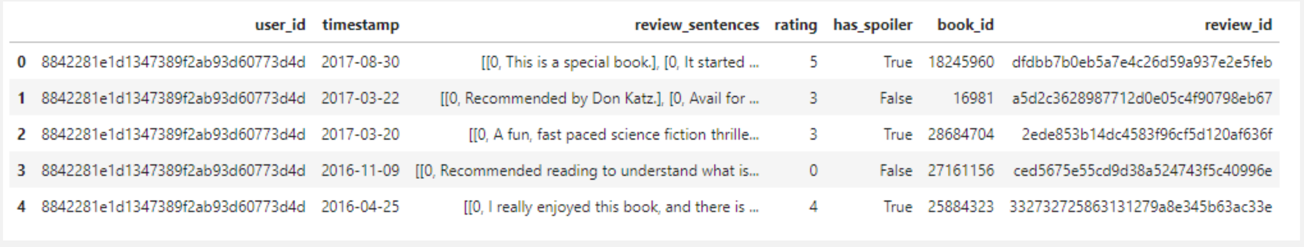
Don't you just love the internet? I wanted to start this paragraph with that quote by Göring, "when I hear the w ...more

176 likes · Like · see review

```
{'user_id': '01ec1a320ffded6b2dd47833f2c8e4fb',
 'timestamp': '2013-12-28',
# a list of sentences, where the first element indicates if the sentence contains spoilers (1) or not (0)
 'review_sentences': [[0, 'First, be aware that this book is not for the faint of heart.'],
  [0, 'Human trafficking, drugs, kidnapping, abuse in all forms - this story contains all of this and more.'],
  ...,
  [0, '(ARC provided by the author in return for an honest review.)']],
 'rating': 5,
 'has_spoiler': False,
 'book_id': '18398089',
 'review_id': '4b3ffeaf14310ac6854f140188e191cd'}
```

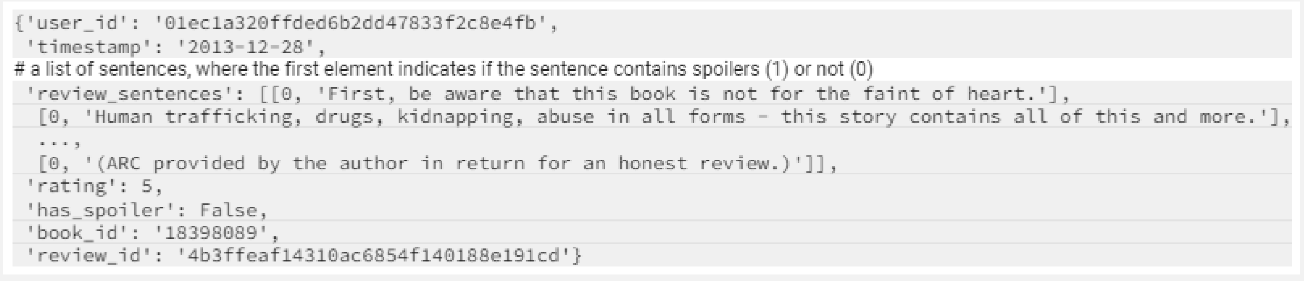
- ➢ Reviews dataset : https://drive.google.com/uc?id=196W2kDoZXRPjzbTjM6uvTidn6aTpsFnS
- ➢ Books dataset : https://drive.google.com/uc?id=1LXpK1UfqtP89H1tYy0pBGHjYk8IhigUK

For the **reviews dataset**, we have 7 columns and 1378033 rows.
Sample head of the data is:



	user_id	timestamp	review_sentences	rating	has_spoiler	book_id	review_id
0	8842281e1d1347389f2ab93d60773d4d	2017-08-30	[[0, This is a special book.], [0, It started ...	5	True	18245960	dfdbb7b0eb5a7e4c26d59a937e2e5feb
1	8842281e1d1347389f2ab93d60773d4d	2017-03-22	[[0, Recommended by Don Katz.], [0, Avail for ...	3	False	16981	a5d2c3628987712d0e05c4f90798eb67
2	8842281e1d1347389f2ab93d60773d4d	2017-03-20	[[0, A fun, fast paced science fiction thrille...	3	True	28684704	2ede853b14dc4583f96cf5d120af636f
3	8842281e1d1347389f2ab93d60773d4d	2016-11-09	[[0, Recommended reading to understand what is...	0	False	27161156	ced5675e55cd9d38a524743f5c40996e
4	8842281e1d1347389f2ab93d60773d4d	2016-04-25	[[0, I really enjoyed this book, and there is ...	4	True	25884323	332732725863131279a8e345b63ac33e

We extract this concentrated English review subset for spoiler detection, where each book/user has at least one associated spoiler review. This dataset contains more than 1.3M book reviews about 25,475 books and 18,892 users.



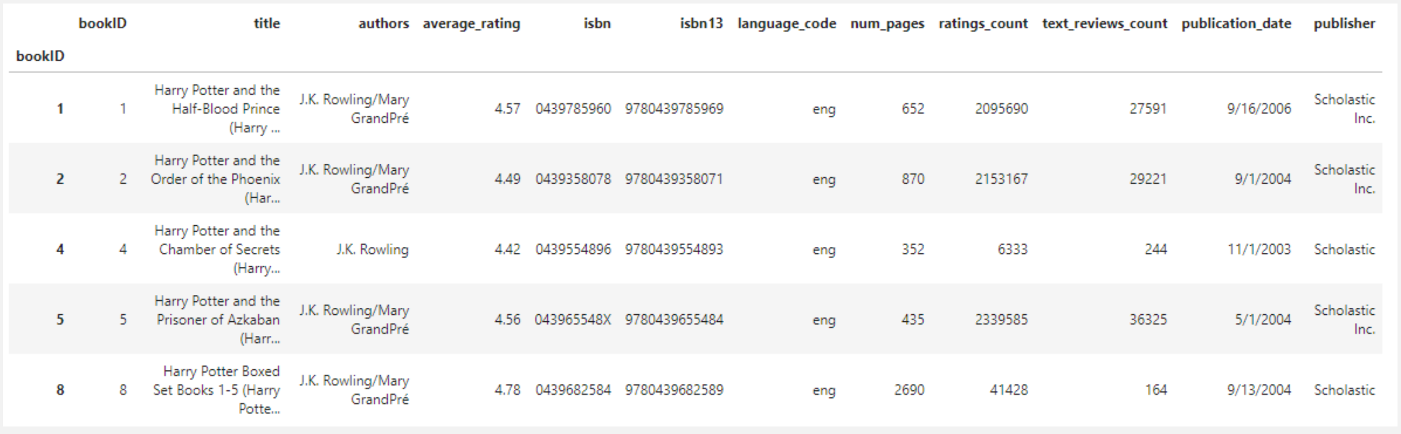
```
{'user_id': '01ec1a320ffded6b2dd47833f2c8e4fb',
 'timestamp': '2013-12-28',
# a list of sentences, where the first element indicates if the sentence contains spoilers (1) or not (0)
 'review_sentences': [[0, 'First, be aware that this book is not for the faint of heart.'],
  [0, 'Human trafficking, drugs, kidnapping, abuse in all forms - this story contains all of this and more.'],
  ...,
  [0, '(ARC provided by the author in return for an honest review.)']],
 'rating': 5,
 'has_spoiler': False,
 'book_id': '18398089',
 'review_id': '4b3ffeaf14310ac6854f140188e191cd'}
```

Details of the ***features*** for this dataset:

- ➢ **user_id** - Represents a unique id of each book user : ***categorical feature***
- ➢ **timestamp** - the timestamp at which user has posted ratings of a book : ***datetime feature***
- ➢ **review_sentences** - This is a list of multiple reviews. Each value represents a list of reviews for a particular book (unique book id) from one user (unique user id) : ***Textual feature***
- ➢ **rating** - this is a numerical value assigned by the user which can be between [0,5] based on their liking for a book : ***Numerical feature***

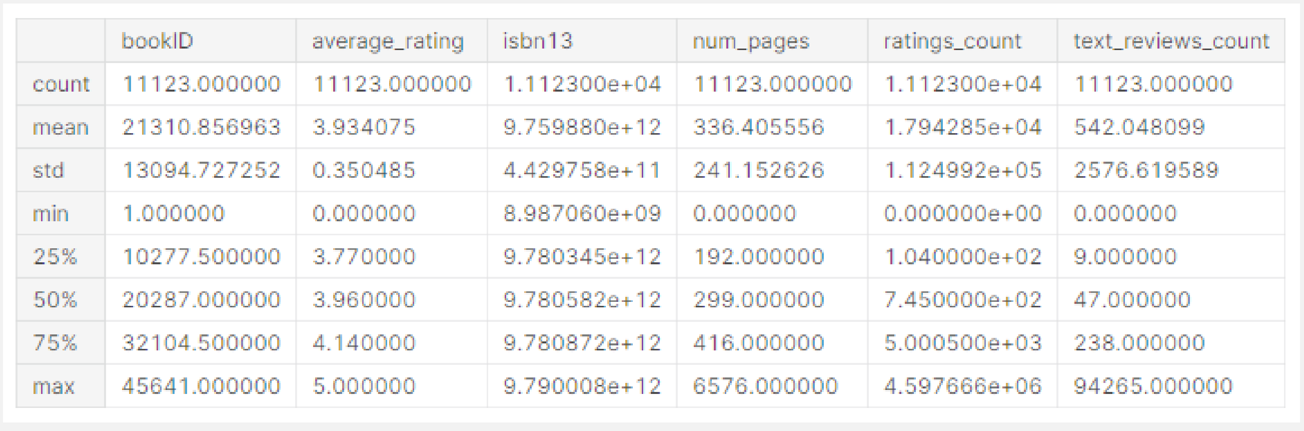
- **Has_spoiler** - This is a boolean value [True, False] that tells us whether or not the review gives out any spoiler for the book. We will convert it into categorical value [0,1] : ***Categorical feature***
- **book_id** - Represents a unique id for each book : ***Categorical feature***
- **Review_id** - Each user can review multiple books. For each review, there will a unique id given by review_id : ***Categorical feature***

For the **books dataset**, we have 12 columns and 11123 rows.
Sample head of the data is:



	bookID	title	authors	average_rating	isbn	isbn13	language_code	num_pages	ratings_count	text_reviews_count	publication_date	publisher
bookID												
1	1	Harry Potter and the Half-Blood Prince (Harry ...	J.K. Rowling/Mary GrandPré	4.57	0439785960	9780439785969	eng	652	2095690	27591	9/16/2006	Scholastic Inc.
2	2	Harry Potter and the Order of the Phoenix (Har...	J.K. Rowling/Mary GrandPré	4.49	0439358078	9780439358071	eng	870	2153167	29221	9/1/2004	Scholastic Inc.
4	4	Harry Potter and the Chamber of Secrets (Harry...	J.K. Rowling	4.42	0439554896	9780439554893	eng	352	6333	244	11/1/2003	Scholastic
5	5	Harry Potter and the Prisoner of Azkaban (Harr...	J.K. Rowling/Mary GrandPré	4.56	043965548X	9780439655484	eng	435	2339585	36325	5/1/2004	Scholastic Inc.
8	8	Harry Potter Boxed Set Books 1-5 (Harry Potte...	J.K. Rowling/Mary GrandPré	4.78	0439682584	9780439682589	eng	2690	41428	164	9/13/2004	Scholastic

We extract this data which consists of details for books with the features described below. This dataset contains about 2.3M books (~2gb data).



	bookID	average_rating	isbn13	num_pages	ratings_count	text_reviews_count
count	11123.000000	11123.000000	1.112300e+04	11123.000000	1.112300e+04	11123.000000
mean	21310.856963	3.934075	9.759880e+12	336.405556	1.794285e+04	542.048099
std	13094.727252	0.350485	4.429758e+11	241.152626	1.124992e+05	2576.619589
min	1.000000	0.000000	8.987060e+09	0.000000	0.000000e+00	0.000000
25%	10277.500000	3.770000	9.780345e+12	192.000000	1.040000e+02	9.000000
50%	20287.000000	3.960000	9.780582e+12	299.000000	7.450000e+02	47.000000
75%	32104.500000	4.140000	9.780872e+12	416.000000	5.000500e+03	238.000000
max	45641.000000	5.000000	9.790008e+12	6576.000000	4.597666e+06	94265.000000

Details of the ***features*** for this dataset:

- **bookID -** Contains the unique ID for each book/series : ***Categorical feature***
- **title** - contains the titles of the books : ***Textual feature***
- **Authors -** contains the author of the particular book : ***Textual feature***
- **average_rating** - the average rating of the books, as decided by the users : ***Numerical feature***
- **ISBN** - ISBN(10) number, tells the information about a book - such as edition and publisher : ***Categorical feature***
- **ISBN 13** - The new format for ISBN, implemented in 2007. 13 digits : ***Categorical feature***
- **Language_code -** Tells the language for the books: ***Categorical feature***
- **Num_pages -** Contains the number of pages for the book : ***Numerical feature***
- **Ratings_count -** Contains the number of ratings given for the book : ***Numerical feature***
- **Text_reviews_count -** Has the count of reviews left by users : ***Numerical feature***

Dataset Preprocessing and Feature selection

- The dataset contains has_spoiler (boolean feature) which we convert into a numerical feature for better analysis.
- Similarly, the review_sentences is a list of all the reviews given (categorical attribute) and for this we take the number of review_sentences given to a certain book for estimating impact.
- We also draw a T-SNE plot for all the numerical features projected into two components.

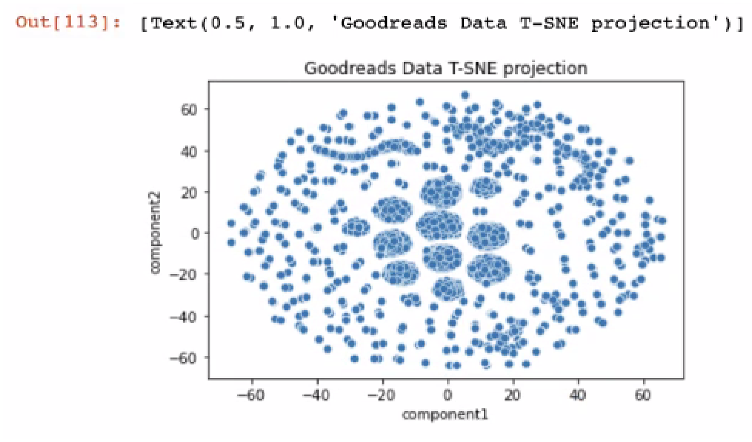
T-SNE (t-distributed stochastic neighbor embedding) is a dimensionality reduction technique that is often used for visualizing high-dimensional data. It is particularly useful for visualizing data in which there are many features (i.e. dimensions), as it can reduce the data down to just two or three dimensions for easy visualization.

In the context of feature selection, t-SNE can be used to identify the most important features in a dataset. By visualizing the data with t-SNE, you can see which features are most strongly correlated with one another and which are less important. This can help you identify the most relevant features for a particular task, such as building a machine learning model.

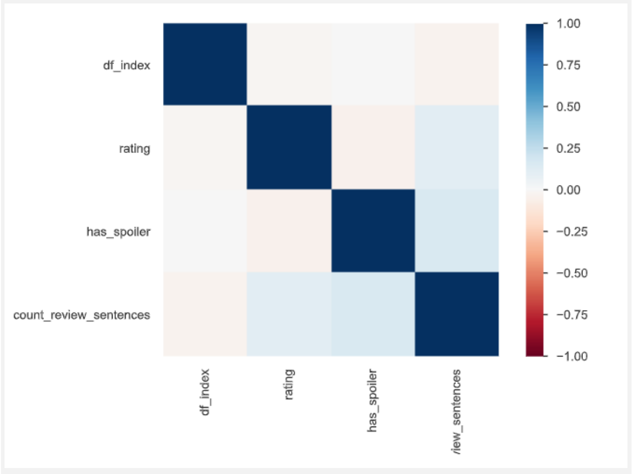
To create a t-SNE plot, you first need to select a set of features from your dataset and then apply t-SNE to reduce the data down to two or three dimensions. The resulting plot will show you how the features are related to one another, and you can use this information to identify the most important features for your task.

Overall, t-SNE is a useful tool for visualizing and understanding high-dimensional data and can be a helpful tool for feature selection in machine learning tasks.

Regenerate response



```
Out[113]: [Text(0.5, 1.0, 'Goodreads Data T-SNE projection')]
```



We find that the features (has_spoiler, count_review_sentence, rating) are not highly correlated so we can consider them ideal features for selection.

Exploratory Data Analysis

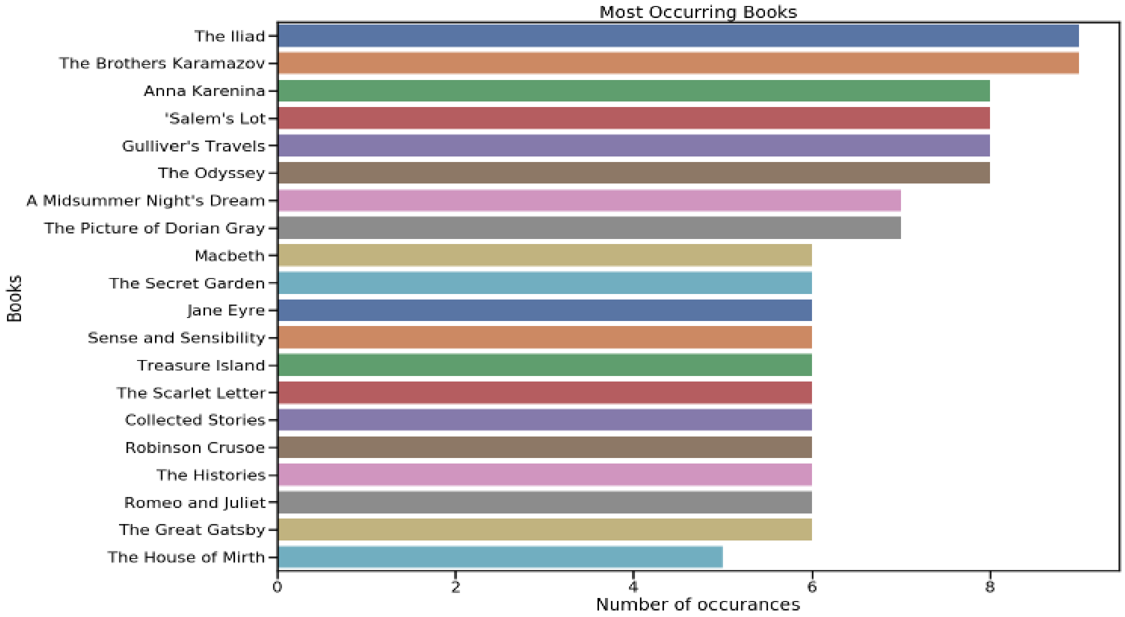
To identify patterns, trends, and anomalies in the dataset, we do exploratory data analyses which helps us gain a better understanding of the data.

<u>Questions we targeted:</u>

- Does any relationship lie between ratings and the total ratings given?
- Where do the majority of the books lie, in terms of ratings - Does reading a book really bring forth bias for the ratings?
- Do authors tend to perform the same over time, with all their newer books? Or do they just fizzle out.
- Do the number of pages make an impact on reading styles, ratings and popularity?
- Can books be recommended based on ratings? Is that a factor which can work?

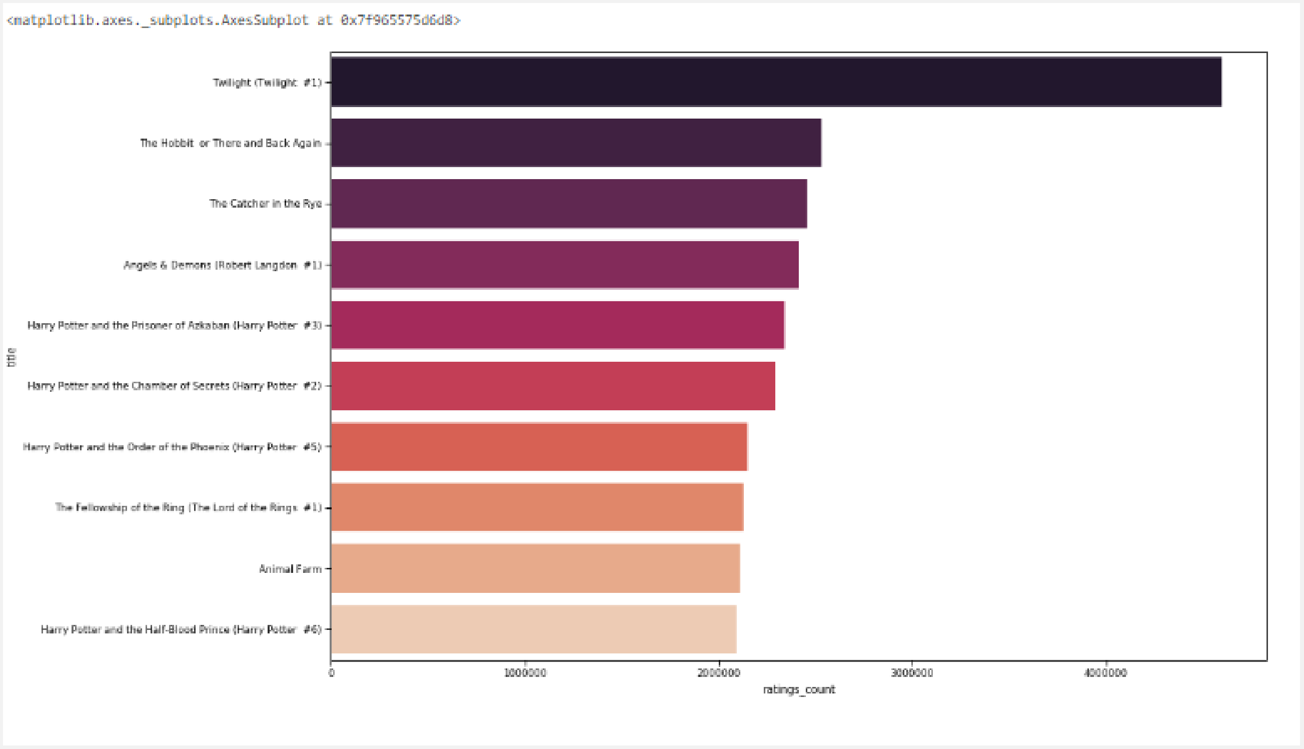
For EDA, we explored our dataset more, and tried to figure the distribution of the data and answer some of the basic following question:

1. **Which are the books with the most occurrences in the list?**



We can see that The Iliad and The Brothers Karamazov have the most number of occurrences with the same name in the data.

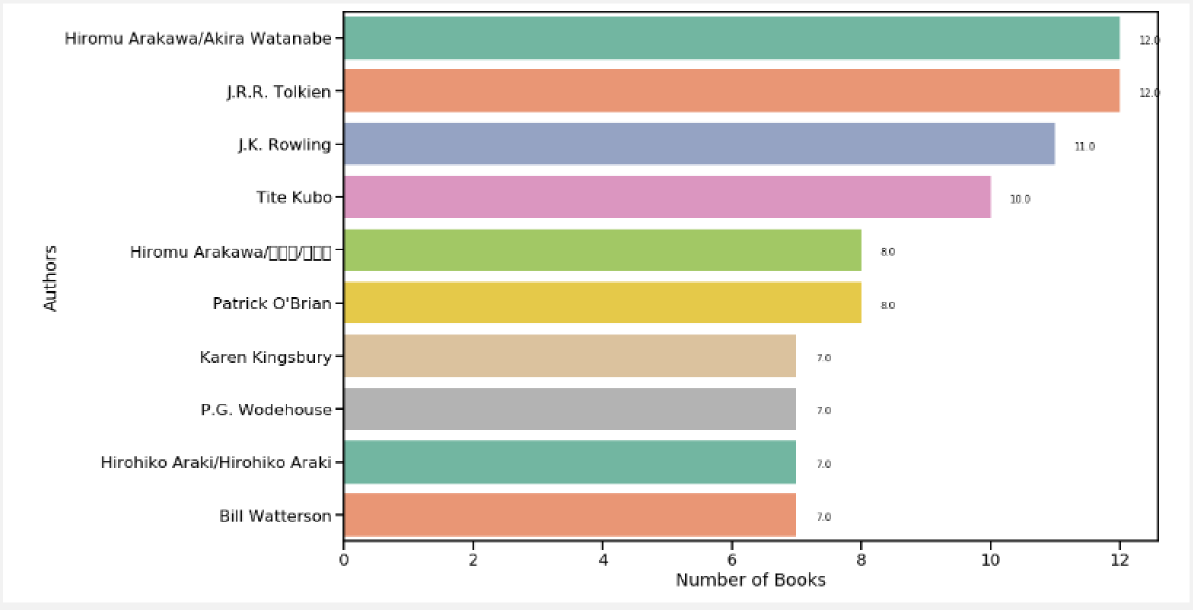
2. **Which are the top 10 most rated books?**



```
<matplotlib.axes._subplots.AxesSubplot at 0x7f965575d6d8>
```

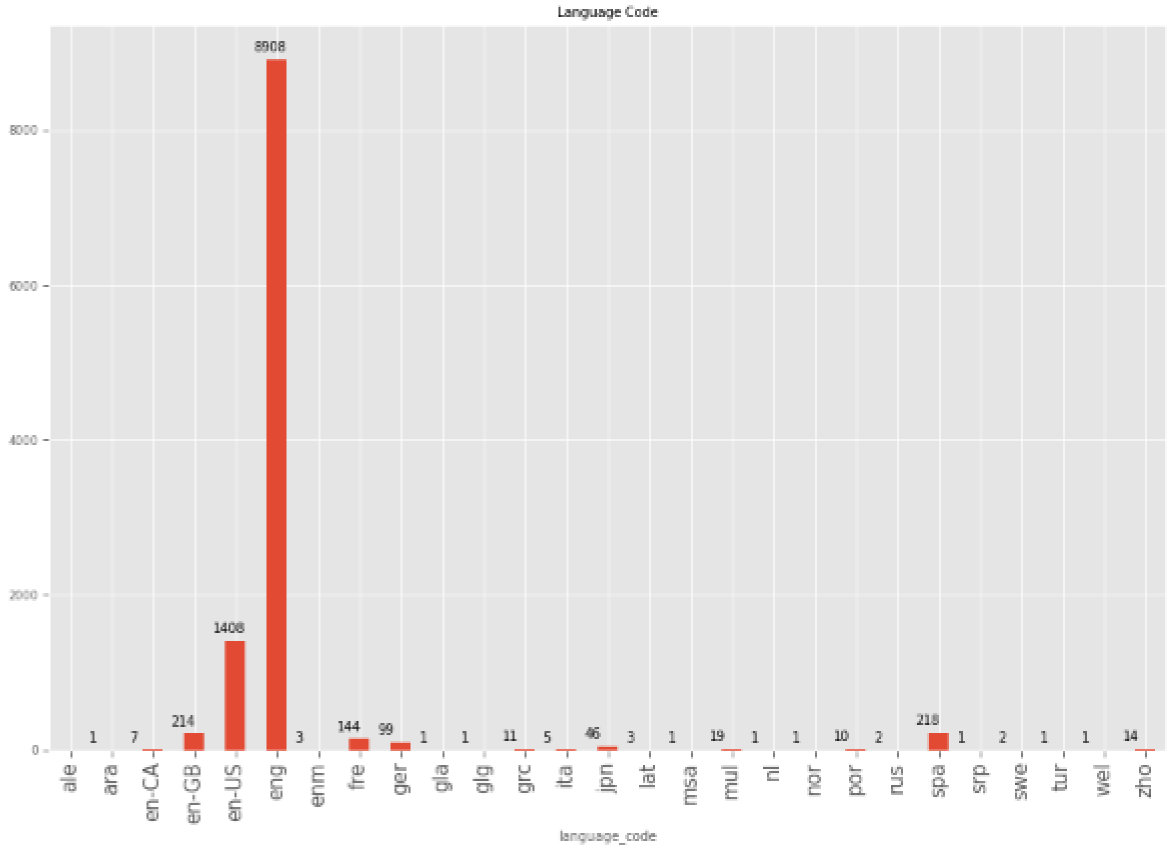
We can see that the beginning books of the series usually have most of the ratings, i.e, **Twilight #1, The Hobbit, Angels and demons #1**.

3. Which are the top 10 highly rated authors?



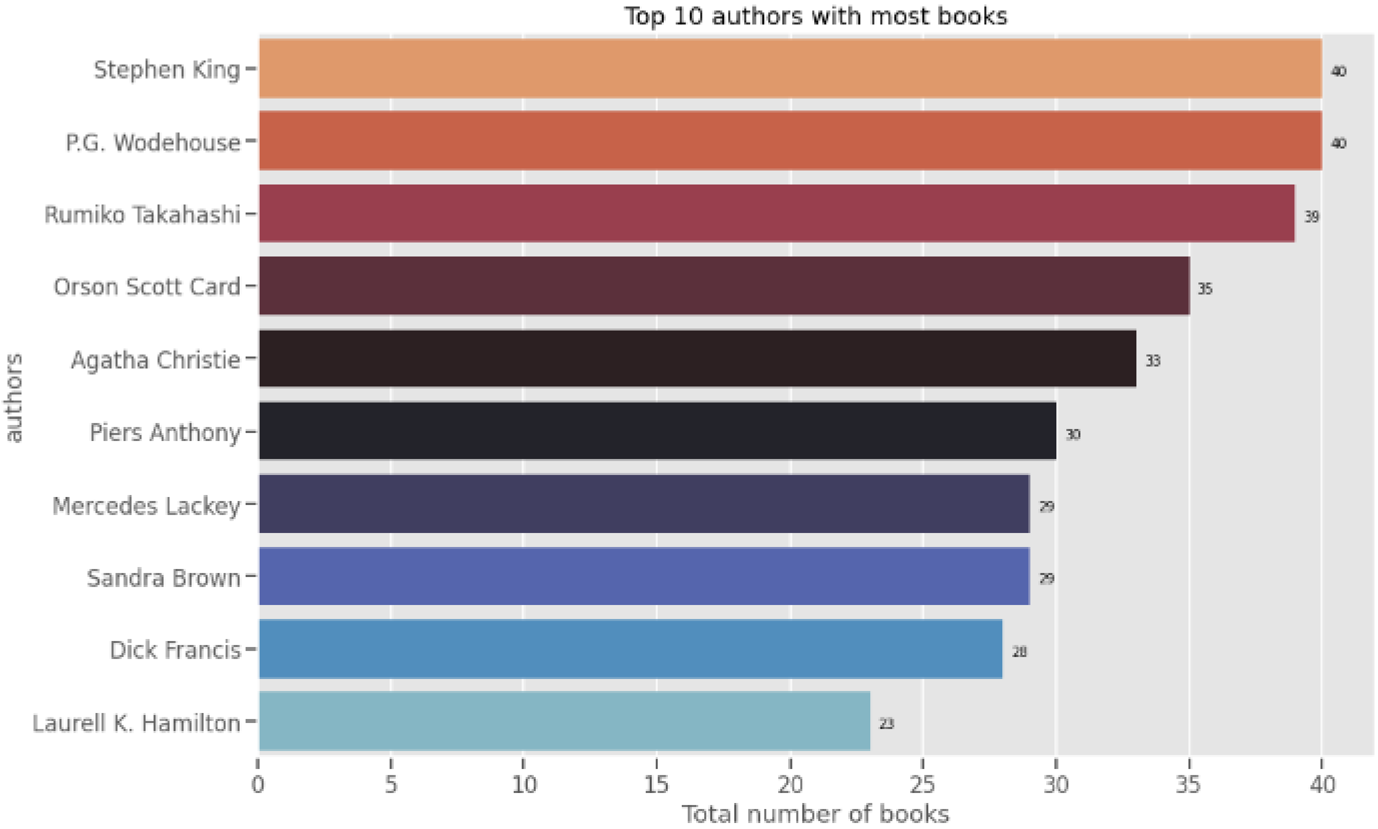
We can infer from the plot that J.R.R Tolkien has the highest ratings for his books, making his average consistency rather impressive.
It's also impressive to note the vast dominance Tolkien has over the competition, easily dethroning any other competition while being above 4.3 in rating.

3. What is the distribution of books for all languages?



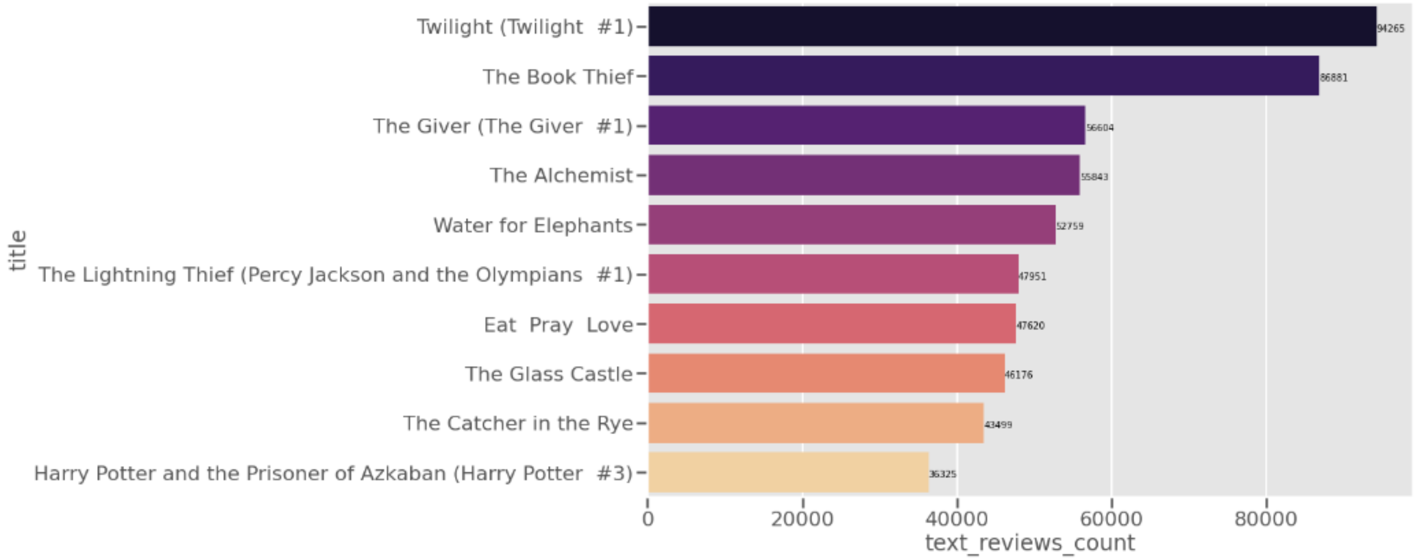
Most popular language is English, followed by Spanish and French.

4. **Which are the authors with most books?**

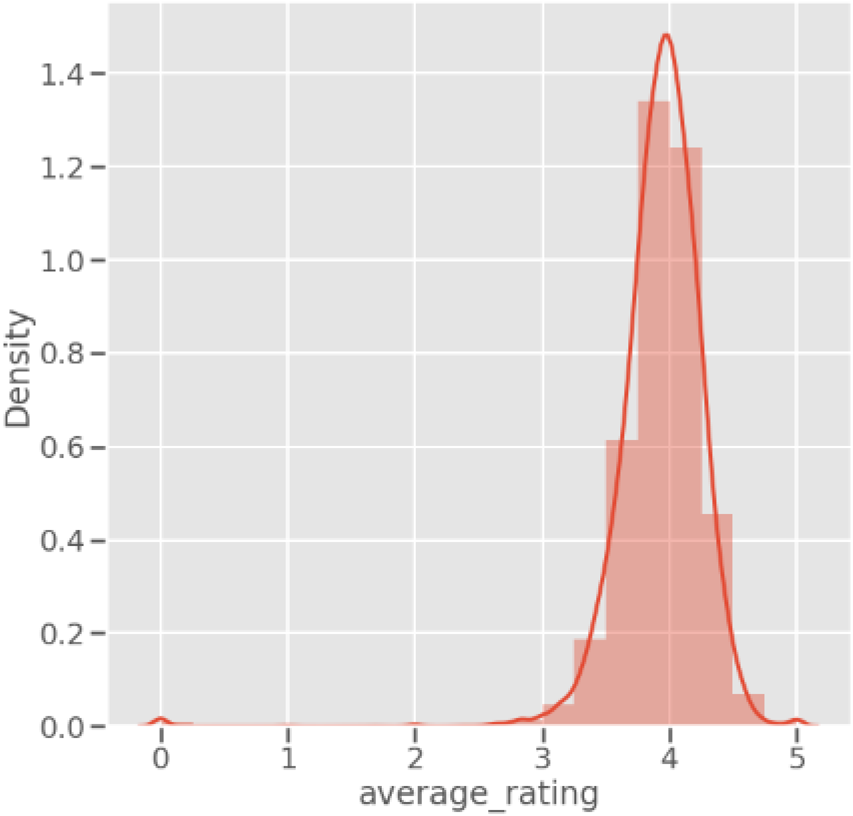


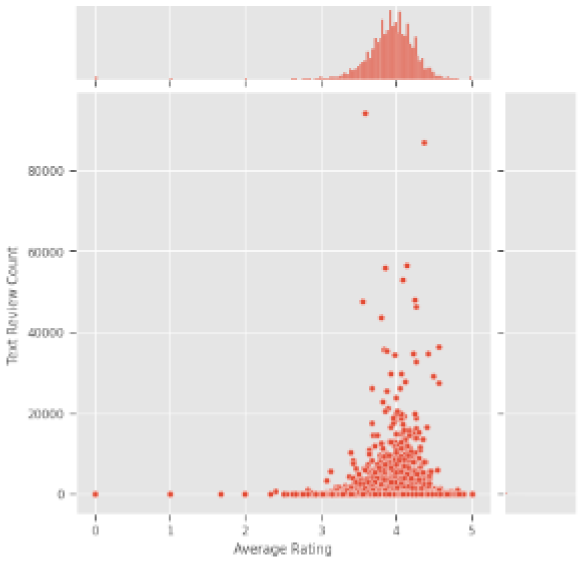
We can see from the above plot that Stephen King has the most number of books in the list. From the names in the list, we can again gather that most of the authors have either been writing for decades, churning numerous books from time to time, or are authors who are regarded as the 'classics' in our history.

5. Which are the books with the highest reviews?



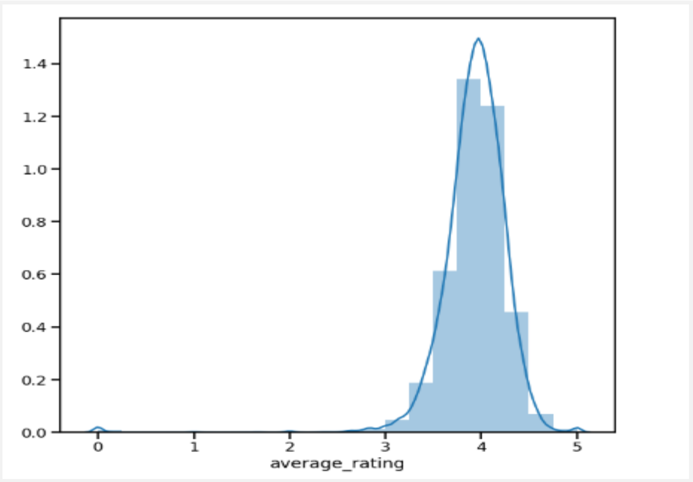
6. What is the rating distribution for the books?

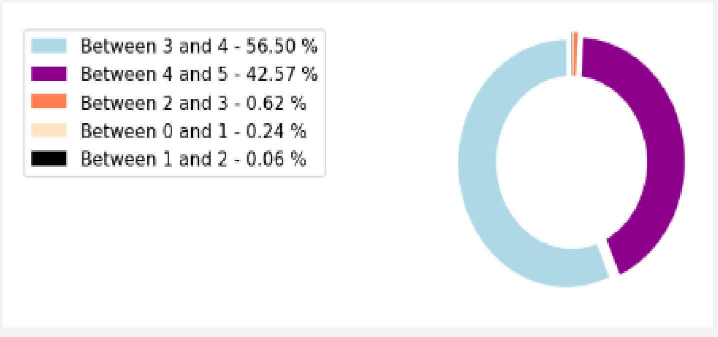




From the given plot, we can infer that:

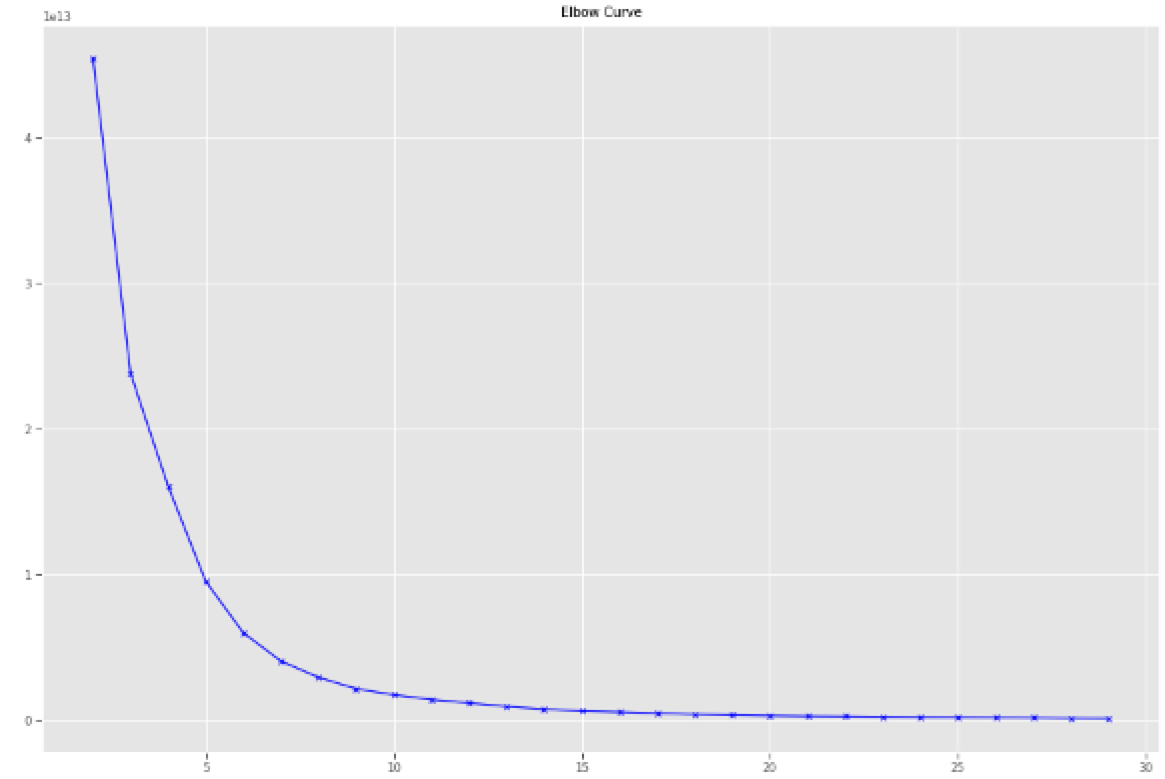
- Majority of the ratings lie near 3.7-4.3, approximately.
- Books having scores near 5 are extremely rare.



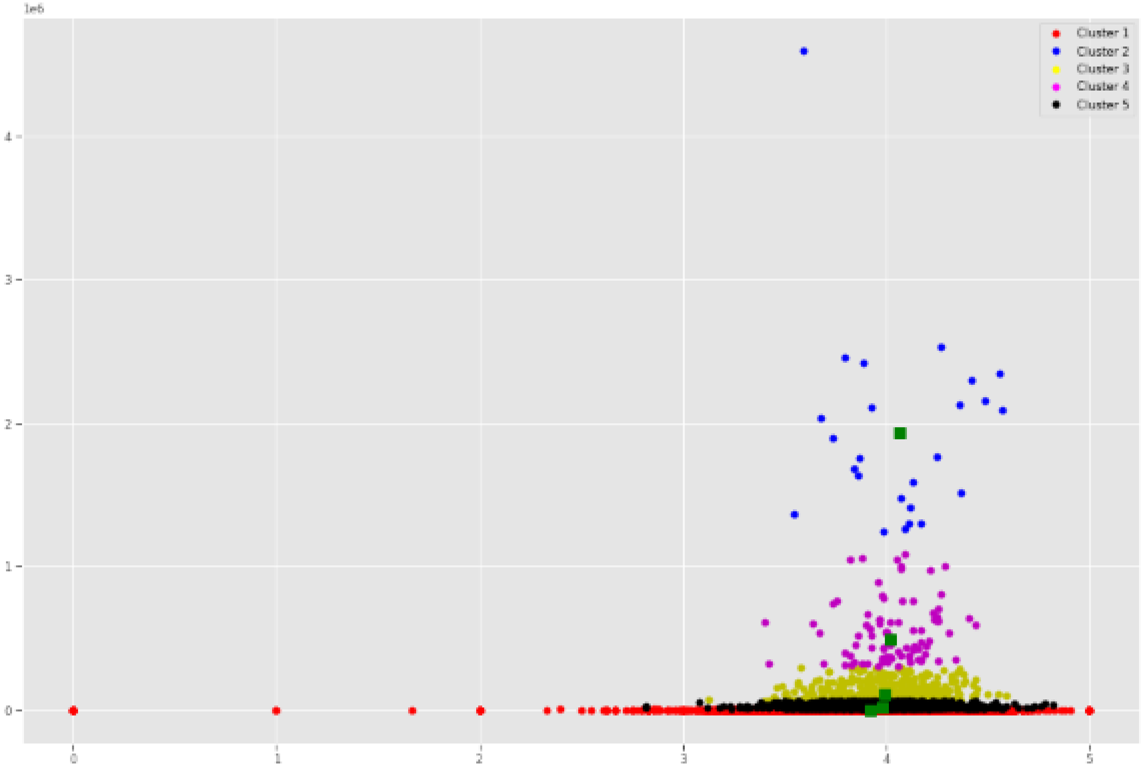


The average of the graph just seems to land between 3 and 4, signifying that for the effort it took to read, and the thrilling nature of the book, the majority of the ratings lie between 3 and 4.
We can infer from the plot that most of the ratings for the books seem to lie near 3-4, with a heavy amount of reviews lying barely near 5000, approximately.

KMeans Clustering without outliers

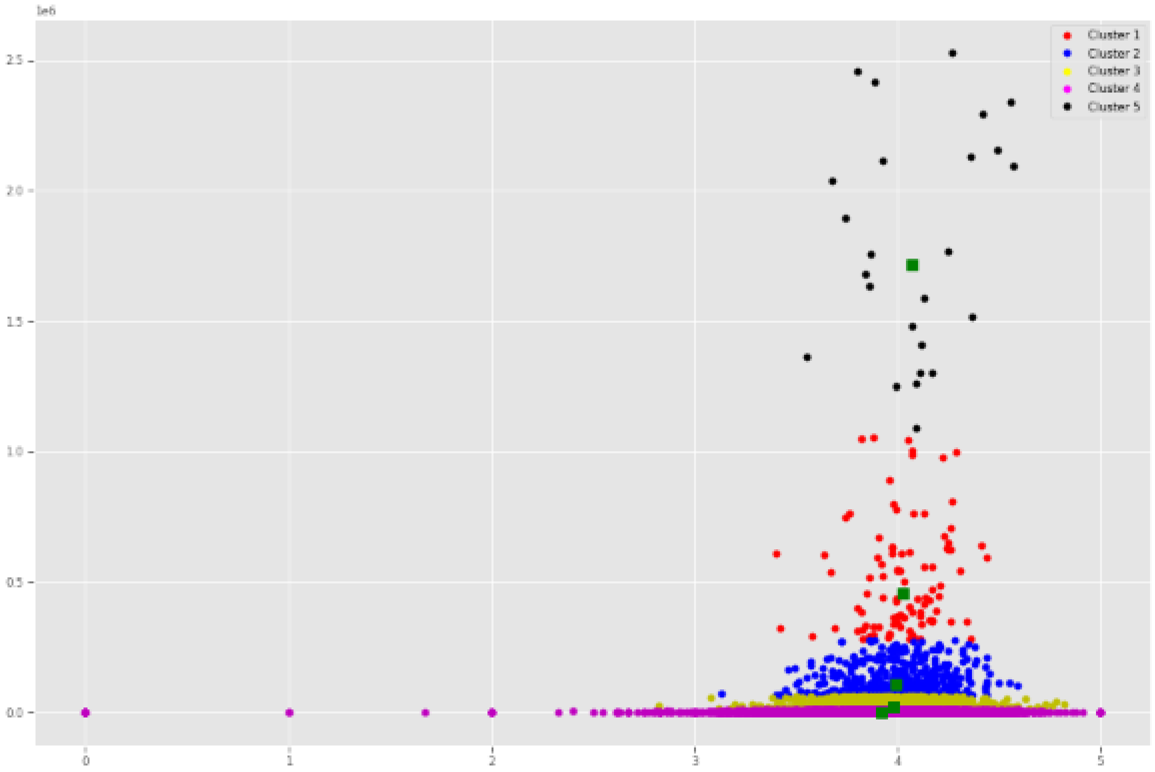


From the above plot, we can see that the elbow lies around the value K=5, so that's what we will attempt it with.



We can see from the above plot that because of one outlier, the whole clustering algorithm is skewed. Let's remove them and form inferences.

KMeans with optimisation



From the above plot, now we can see that once the whole system can be classified into clusters. As the count increases, the rating would end up near the cluster given above. The green squares are the centroids for the given clusters. As the rating count seems to decrease, the average rating seems to become sparser, with higher volatility and less accuracy.

Recommendation Engine

Having seen the clustering, we can infer that there can be some recommendations which can happen with the relation between Average Rating and Ratings Count.

Taking the Ratings_Distribution (A self created classifying trend), the recommendation system works with the algorithm of K Nearest Neighbors.

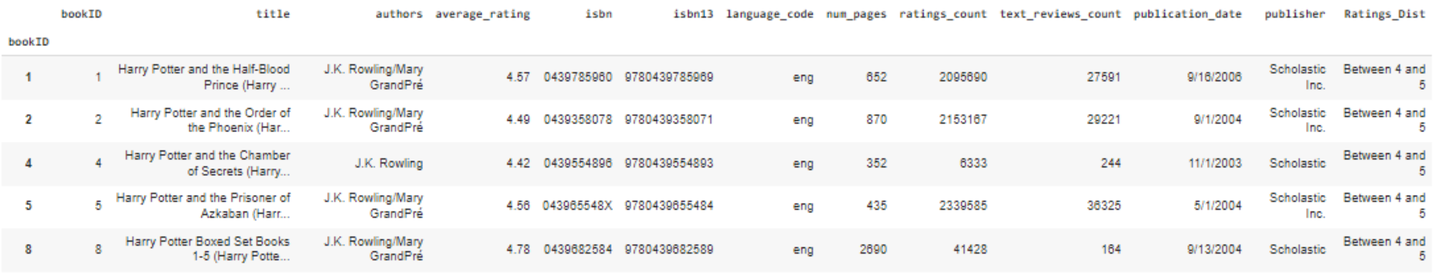
In a setting such as this, unsupervised learning takes place, with the similar neighbors being recommended. For the given list, if I ask for recommendations for "The Catcher in the Rye", five books related to it would appear.

Creating a books features table, based on the Ratings Distribution, which classifies the books into ratings scale such as:

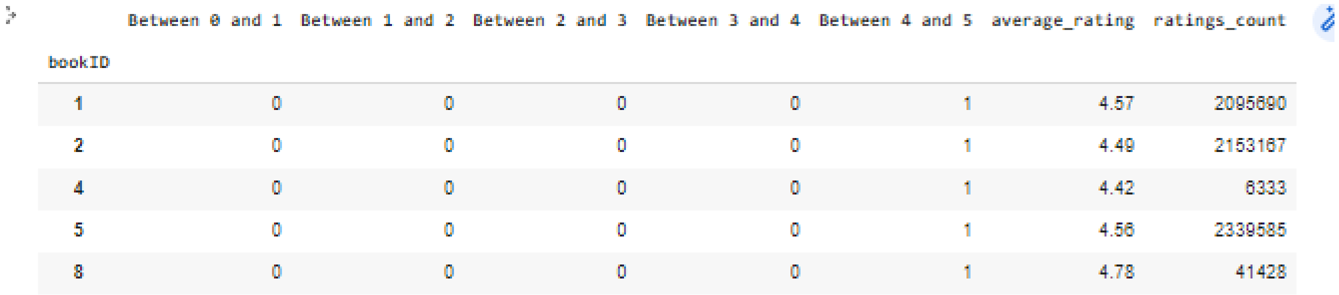
- Between 0 and 1
- Between 1 and 2
- Between 2 and 3
- Between 3 and 4
- Between 4 and 5

Broadly, the recommendations then consider the average ratings and ratings cout for the query entered.

After adding column “Rating_dist”

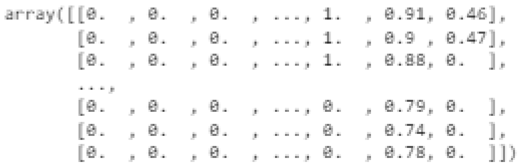


bookID	bookID	title	authors	average_rating	isbn	isbn13	language_code	num_pages	ratings_count	text_reviews_count	publication_date	publisher	Ratings_Dist
1	1	Harry Potter and the Half-Blood Prince (Harry ...	J.K. Rowling/Mary GrandPré	4.57	0439785960	9780439785969	eng	652	2095690	27591	9/16/2006	Scholastic Inc.	Between 4 and 5
2	2	Harry Potter and the Order of the Phoenix (Har...	J.K. Rowling/Mary GrandPré	4.49	0439358078	9780439358071	eng	870	2153167	29221	9/1/2004	Scholastic Inc.	Between 4 and 5
4	4	Harry Potter and the Chamber of Secrets (Harry...	J.K. Rowling	4.42	0439554896	9780439554893	eng	352	6333	244	11/1/2003	Scholastic	Between 4 and 5
5	5	Harry Potter and the Prisoner of Azkaban (Harr...	J.K. Rowling/Mary GrandPré	4.56	043965548X	9780439655484	eng	435	2339585	36325	5/1/2004	Scholastic Inc.	Between 4 and 5
8	8	Harry Potter Boxed Set Books 1-5 (Harry Potte...	J.K. Rowling/Mary GrandPré	4.78	0439682584	9780439682589	eng	2690	41428	164	9/13/2004	Scholastic	Between 4 and 5



bookID	Between 0 and 1	Between 1 and 2	Between 2 and 3	Between 3 and 4	Between 4 and 5	average_rating	ratings_count
1	0	0	0	0	1	4.57	2095690
2	0	0	0	0	1	4.49	2153167
4	0	0	0	0	1	4.42	6333
5	0	0	0	0	1	4.56	2339585
8	0	0	0	0	1	4.78	41428

The min-max scaler is used to reduce the bias which would have been present due to some books having a massive amount of features, yet the rest having less. Min-Max scaler would find the median for them all and equalize it.



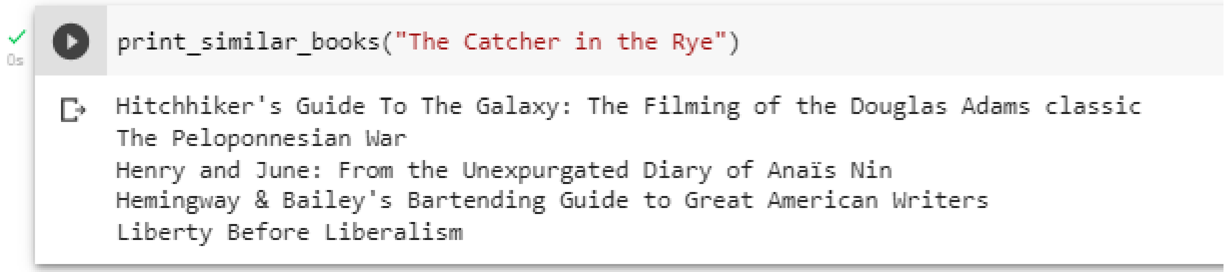
```
array([[0.  , 0.  , 0.  , ..., 1.  , 0.91, 0.46],
       [0.  , 0.  , 0.  , ..., 1.  , 0.9 , 0.47],
       [0.  , 0.  , 0.  , ..., 1.  , 0.88, 0.  ],
       ...,
       [0.  , 0.  , 0.  , ..., 0.  , 0.79, 0.  ],
       [0.  , 0.  , 0.  , ..., 0.  , 0.74, 0.  ],
       [0.  , 0.  , 0.  , ..., 0.  , 0.78, 0.  ]])
```

Ball tree is used for the Nearest Neighbour search. The Ball Tree and the KD Tree algorithm are tree algorithms used for spatial division of data points and their allocation into certain regions. In other words, they are used to structure data in a multidimensional space.

Creating specific functions to help in finding the book names:

- Get index from Title
- Get ID from partial name (Because not everyone can remember all the names)
- Print the similar books from the feature dataset. (This uses the Indices metric from the nearest neighbors to pick the books.)

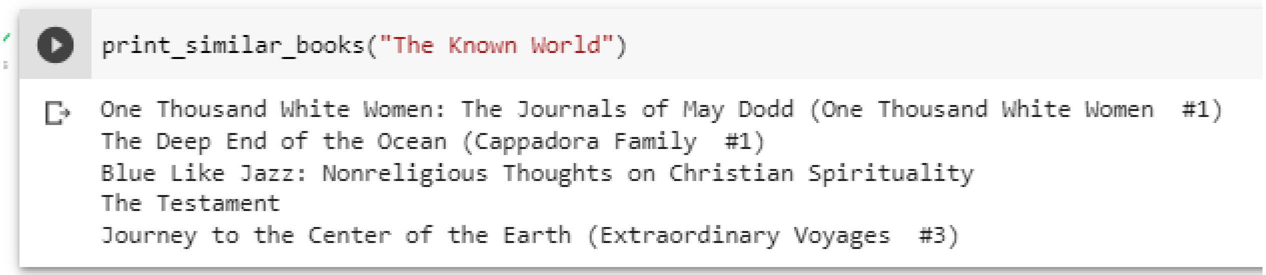
The Catcher in the Rye:



```
print_similar_books("The Catcher in the Rye")
```

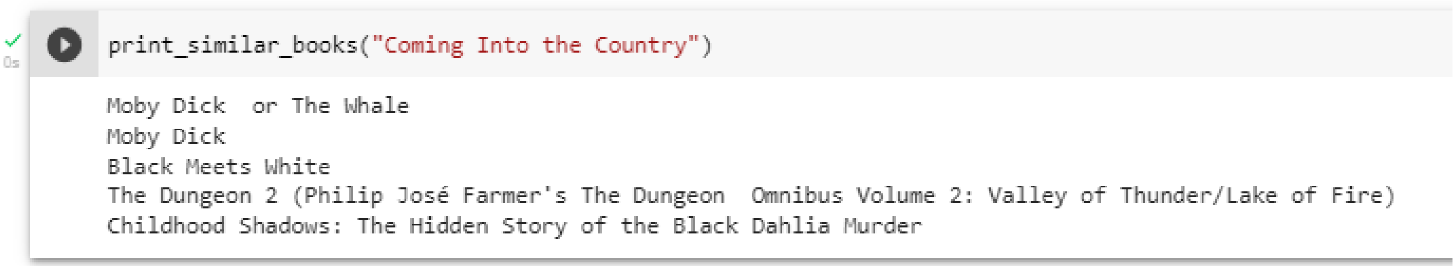
```
Hitchhiker's Guide To The Galaxy: The Filming of the Douglas Adams classic
The Peloponnesian War
Henry and June: From the Unexpurgated Diary of Anaïs Nin
Hemingway & Bailey's Bartending Guide to Great American Writers
Liberty Before Liberalism
```

More Recommendations:



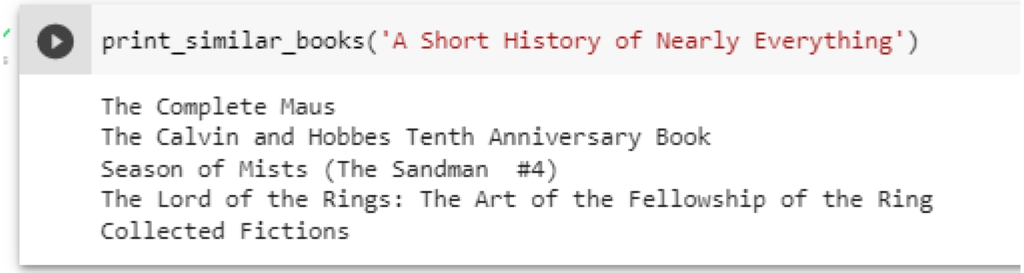
```
print_similar_books("The Known World")
```

```
One Thousand White Women: The Journals of May Dodd (One Thousand White Women  #1)
The Deep End of the Ocean (Cappadora Family  #1)
Blue Like Jazz: Nonreligious Thoughts on Christian Spirituality
The Testament
Journey to the Center of the Earth (Extraordinary Voyages  #3)
```



```
print_similar_books("Coming Into the Country")
```

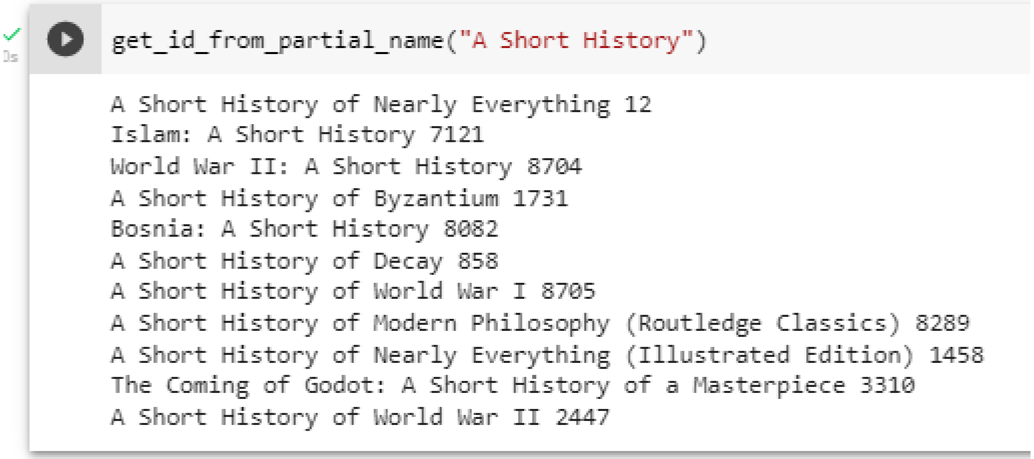
```
Moby Dick  or The Whale
Moby Dick
Black Meets White
The Dungeon 2 (Philip José Farmer's The Dungeon  Omnibus Volume 2: Valley of Thunder/Lake of Fire)
Childhood Shadows: The Hidden Story of the Black Dahlia Murder
```



```
print_similar_books('A Short History of Nearly Everything')
```

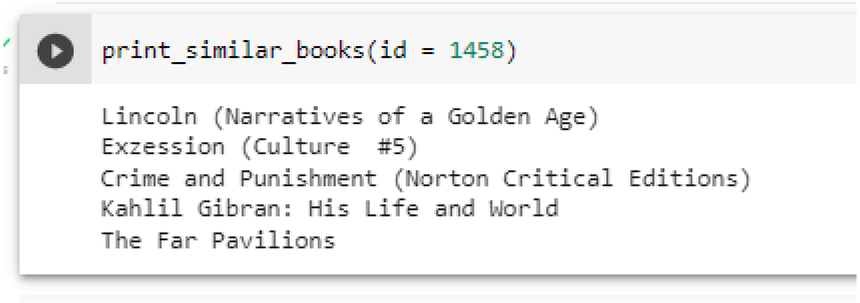
```
The Complete Maus
The Calvin and Hobbes Tenth Anniversary Book
Season of Mists (The Sandman  #4)
The Lord of the Rings: The Art of the Fellowship of the Ring
Collected Fictions
```

Since most users won't remember the name for the entire book (Especially how it has been entered in the books database), the function to get ID from the partial names helps to choose to ID of the book the user is looking for.



```
get_id_from_partial_name("A Short History")
```

```
A Short History of Nearly Everything 12
Islam: A Short History 7121
World War II: A Short History 8704
A Short History of Byzantium 1731
Bosnia: A Short History 8082
A Short History of Decay 858
A Short History of World War I 8705
A Short History of Modern Philosophy (Routledge Classics) 8289
A Short History of Nearly Everything (Illustrated Edition) 1458
The Coming of Godot: A Short History of a Masterpiece 3310
A Short History of World War II 2447
```



```
print_similar_books(id = 1458)

Lincoln (Narratives of a Golden Age)
Exzession (Culture  #5)
Crime and Punishment (Norton Critical Editions)
Kahlil Gibran: His Life and World
The Far Pavilions
```

Formulation of Machine Learning Task

Problem Statement: Given the goodreads book rating dataset, we wish to recommend books to different users based on the similarity between the users and also for a given book recommend similar books to users.

Procedure Applied:

- We will use the user-based Collaborative Filtering model to predict and recommend books to users
- This model will determine similar interests and patterns from other users, for generating recommendations.
- The approach used is an Unsupervised Machine learning Algorithm.

Feature Selection:

- Collaborative Filtering: (from goodreads review dataset)
    - User_id
    - Book_id
    - ratings
- Clustering :(from goodreads book dataset)
    - Book_id
    - Authors
    - title

What is Collaborative filtering?

Collaborative filtering is a popular technique for recommendation systems that uses the past behavior of users to make recommendations for new items. It works by identifying users who have similar tastes or preferences, and then using their past ratings or interactions with items to recommend new items to a given user.

There are two main approaches to collaborative filtering: user-based and item-based.
In user-based collaborative filtering, the system compares a given user's past ratings or interactions with items to those of other users. It then identifies users who have similar tastes or preferences and uses their ratings or interactions to recommend new items to the given user. This approach is useful when the number of items is large, as it reduces the computational burden of comparing the given user's ratings with all items. However, it can be less effective when there are few users with similar tastes, as there may not be enough data to make reliable recommendations.

In item-based collaborative filtering, the system compares the ratings or interactions of a given user with those of other users for a particular item. It then identifies other items that have been rated highly by the same users and recommends those items to the given user. This approach is useful when the number of users is large, as it reduces the computational burden of comparing the given user's ratings with all users. However, it can be less effective when there are few items with similar ratings patterns, as there may not be enough data to make reliable recommendations.

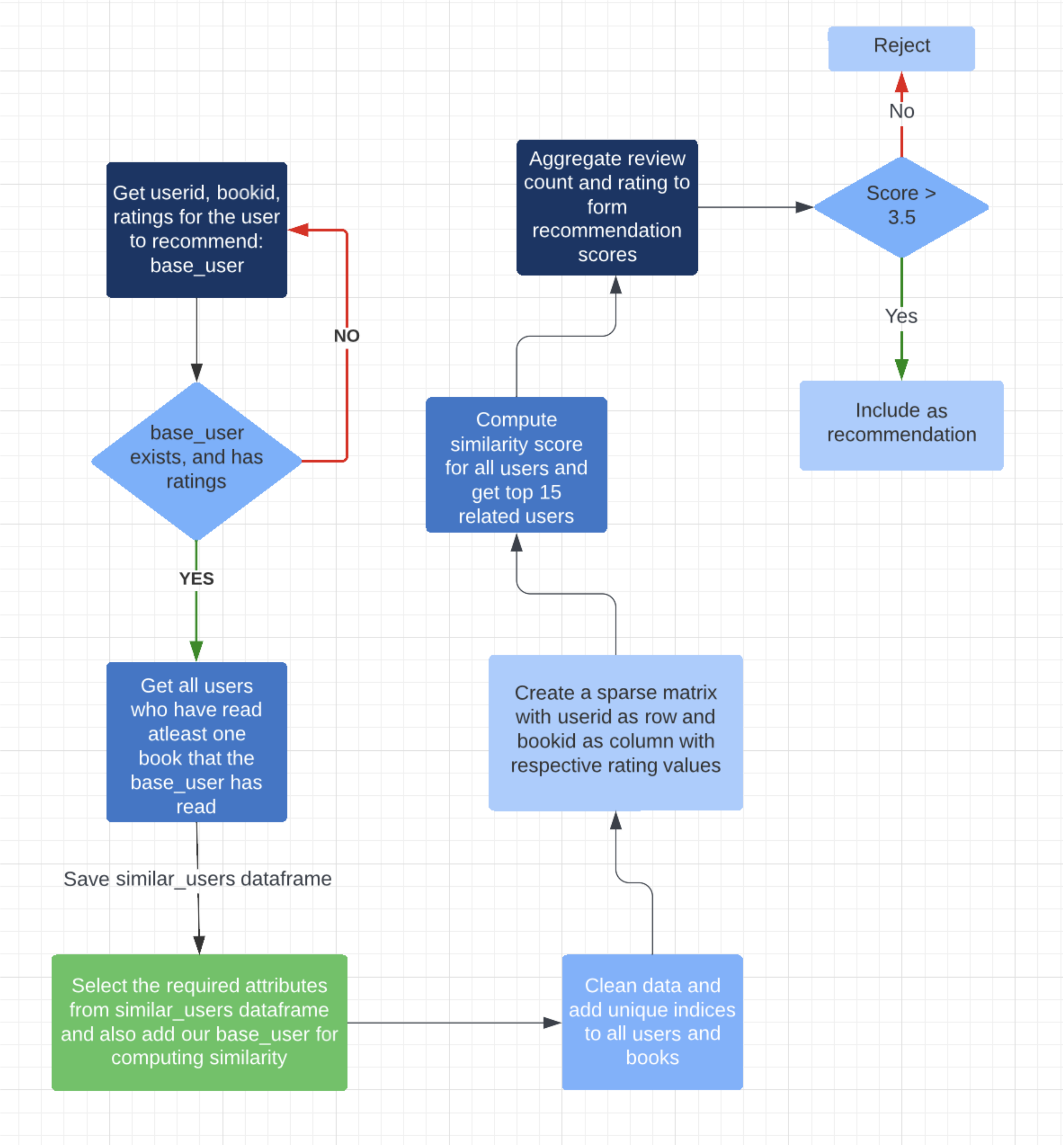
Both user-based and item-based collaborative filtering have their own strengths and weaknesses, and which approach is used may depend on the specific requirements and characteristics of the recommendation system.

Approach :

- The dataset does not contain the detailed feature of the books to use content based recommendation and hence we decided to use the collaborative based recommendations.
- For collaborative filtering we plan to use Cosine similarity and Correlation to find similarity between all the users(to determine the list of all other users who are similar to a particular user).

$$similarity(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

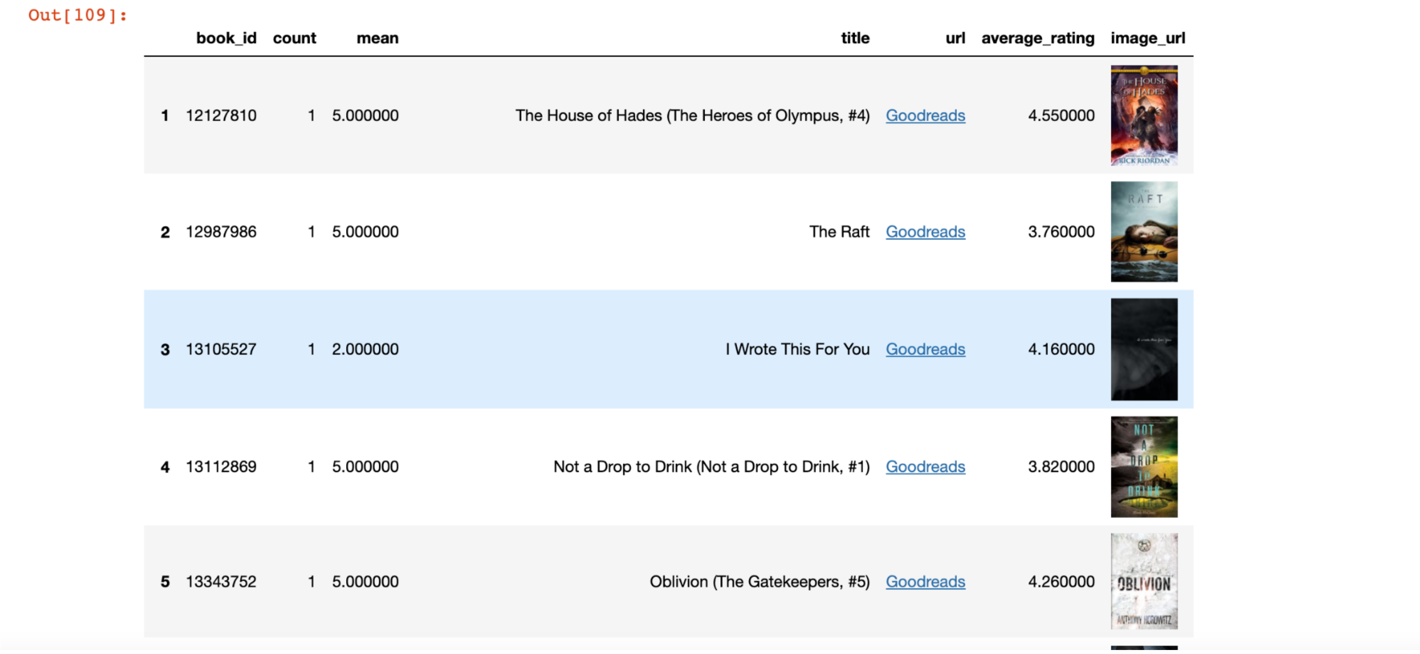
- After finding similarity, between all the users, we can recommend the most similar users to their recommended books.
- We also plan to cluster the books using K-means Clustering, to check if we can recommend better books by applying collaborative filtering on each cluster.


Reject
No
Aggregate review count and rating to form recommendation scores
Score > 3.5
Yes
Include as recommendation
Get userid, bookid, ratings for the user to recommend: base_user
NO
base_user exists, and has ratings
YES
Compute similarity score for all users and get top 15 related users
Get all users who have read atleast one book that the base_user has read
Create a sparse matrix with userid as row and bookid as column with respective rating values
Save similar_users dataframe
Select the required attributes from similar_users dataframe and also add our base_user for computing similarity
Clean data and add unique indices to all users and books

Expected Result :

- For a given user, we should be able to predict the books for most similar users.
- For a given book, we should also be able to find similar books for recommendation.

Sample book recommendations:



Out[109]:

	book_id	count	mean	title	url	average_rating	image_url
1	12127810	1	5.000000	The House of Hades (The Heroes of Olympus, #4)	Goodreads	4.550000	
2	12987986	1	5.000000	The Raft	Goodreads	3.760000	
3	13105527	1	2.000000	I Wrote This For You	Goodreads	4.160000	
4	13112869	1	5.000000	Not a Drop to Drink (Not a Drop to Drink, #1)	Goodreads	3.820000	
5	13343752	1	5.000000	Oblivion (The Gatekeepers, #5)	Goodreads	4.260000	

Model evaluation

Model evaluation of collaborative filtering-based recommendation systems is important because it helps to determine the effectiveness and accuracy of the recommendations being made by the system. A well-evaluated model can provide more accurate and relevant recommendations to users, leading to a better user experience and increased engagement.

Predictive accuracy: The primary goal of a recommendation system is to make accurate recommendations to users. Therefore, it is important to measure the accuracy of the model's predictions. One way to do this is by using the Root Mean Squared Error (RMSE) metric.

Procedure followed to calculate RMSE for our model:

- Split the dataset into a training set and a test set (70:30 ratio).
- Train the model on the training set.
- Make predictions on the test set using the trained model.
- Calculate the difference between the predicted ratings and the actual ratings for each user-item pair in the test set.
- Calculate the square root of the mean of the squared differences.

Output:

RMSE value for our model : 0.28

The RMSE (Root Mean Squared Error) value for the model is a measure of the model's performance in predicting the ratings of items for users. A lower RMSE value indicates that the model is making more accurate predictions, while a higher RMSE value indicates that the model is making less accurate predictions.

Conclusion:

In this case, an RMSE value of 0.28 for the model is relatively low, indicating that the model is making relatively accurate predictions. This suggests that the model is well-tuned and performing well, or that the data being used to train the model is of high quality.

Issues with collaborative filtering:

1. **Cold start problem:**

   Collaborative filtering relies on having a sufficient amount of data about users and their interactions with items. If there is not enough data available, the system may not be able to make accurate recommendations. This is known as the cold start problem.

2. **Sparsity:**

   If there are many items and relatively few ratings or interactions for each item, the data may be too sparse to generate accurate recommendations. This is because there may not be enough information about a given user's preferences to make reliable recommendations.

3. **Shilling attacks:**

   Collaborative filtering systems may be vulnerable to shilling attacks, where an individual or group of individuals attempt to manipulate the system by artificially inflating the ratings of certain items. This can lead to inaccurate or biased recommendations.

4. **Privacy concerns:**

   Collaborative filtering systems may raise privacy concerns, as they rely on collecting and analyzing data about users' interactions with items. This data may be sensitive or personal in nature, and there may be concerns about how it is used and protected.

5. **Limited personalization:**

   Collaborative filtering systems may not be able to take into account individual differences or preferences, leading to recommendations that are not tailored to a specific user.

References

1. Castellano G, Fanelli AM, Torsello MA. NEWER: *A system for neuro-fuzzy web recommendation*. Appl Soft Comput. 2011;11:793–806.
2. Crespo RG, Martínez OS, Lovelle JMC, García-Bustelo BCP, Gayo JEL, Pablos PO. *Recommendation system based on user interaction data applied to intelligent electronic books.* Computers Hum Behavior. 2011;27:1445–9.
3. Lin FC, Yu HW, Hsu CH, Weng TC. *Recommendation system for localized products in vending machines*. Expert Syst Appl. 2011;38:9129–38.
4. Wang SL, Wu CY. *Application of context-aware and personalized recommendation to implement an adaptive ubiq- uitous learning system*. Expert Syst Appl. 2011;38:10831–8.
5. García-Crespo Á, López-Cuadrado JL, Colomo-Palacios R, González-Carrasco I, Ruiz-Mezcua B. Sem-Fit: *A semantic based expert system to provide recommendations in the tourism domain.* Expert Syst Appl. 2011;38:13310–9.
6. Dong H, Hussain FK, Chang E. *A service concept recommendation system for enhancing the dependability of semantic service matchmakers in the service ecosystem environment.* J Netw Comput Appl. 2011;34:619–31.
7. Li M, Liu L, Li CB. *An approach to expert recommendation based on fuzzy linguistic method and fuzzy text classification in knowledge management systems.* Expert Syst Appl. 2011;38:8586–96.
8. Lorenzi F, Bazzan ALC, Abel M, Ricci F. *Improving recommendations through an assumption-based multiagent approach: An application in the tourism domain.* Expert Syst Appl. 2011;38:14703–14.
9. Huang Z, Lu X, Duan H. *Context-aware recommendation using rough set model and collaborative filtering.* Artif Intell Rev. 2011;35:85–99.
10. Chen RC, Huang YH, Bau CT, Chen SM. *A recommendation system based on domain ontology and SWRL for anti-diabetic drugs selection*. Expert Syst Appl. 2012;39:3995–4006.
11. Mohanraj V, Chandrasekaran M, Senthilkumar J, Arumugam S, Suresh Y. *Ontology driven bee's foraging approach based self-adaptive online recommendation system.* J Syst Softw. 2012;85:2439–50.
12. Hsu CC, Chen HC, Huang KK, Huang YM. *A personalized auxiliary material recommendation system based on learning style on facebook applying an artificial bee colony algorithm*. Comput Math Appl. 2012;64:1506–13.
13. Gemmell J, Schimoler T, Mobasher B, Burke R. *Resource recommendation in social annotation systems: A linear-weighted hybrid approach*. J Comput Syst Sci. 2012;78:1160–74.
14. Choi K, Yoo D, Kim G, Suh Y. *A hybrid online-product recommendation system: Combining implicit rating-based collaborative filtering and sequential pattern analysis.* Electron Commer Res Appl. 2012;11:309–17.
15. Garibaldi JM, Zhou SM, Wang XY, John RI, Ellis IO. I*ncorporation of expert variability into breast cancer treatment recommendation in designing clinical protocol guided fuzzy rule system models.* J Biomed Inform. 2012;45:447–59.