Unsupervised ML Project

Book Recommendation System

By- Ranajay Biswas

Content :

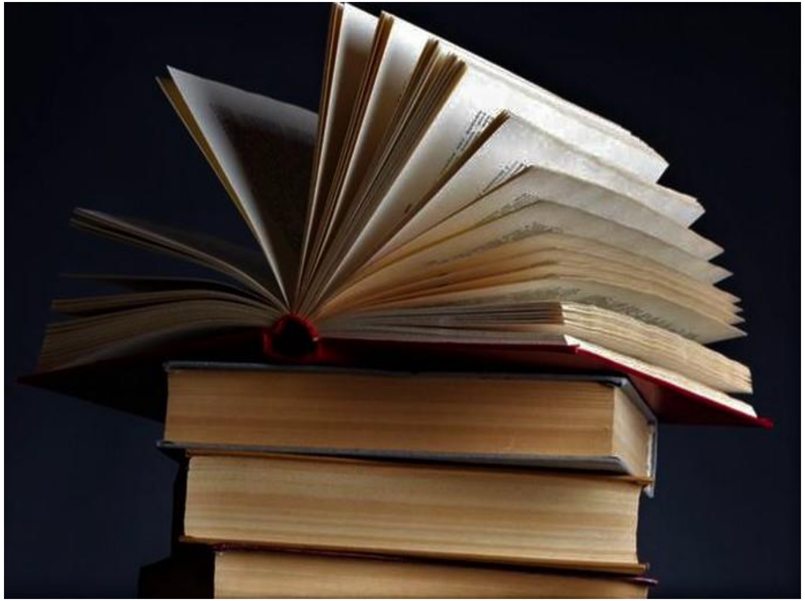
- Introduction
- Objectives & Problem Statement
- Data Summary and Attributes
- Exploratory Data Analysis
- Pre-Processing & Feature Engineering
- Modelling
- Conclusion

Introduction :

A recommendation system broadly recommends items to the user best suited to their tastes and traits. It uses the user's previous data and other user's data to give new recommendations.

Book Recommendation systems are popular recommendations system as most people have a very limited time that they spend on trying out and reading new books. So, when they visit an online bookstore or just simply search on the internet about some book, it becomes important to utilize this opportunity to make recommendations that are similar to what they would like.

Also it is important to consider books that are worth reading, meaning books that are popular for being good among other readers.



Problem Statement :

During the last few decades, with the rise of YouTube, Amazon, Netflix, and many other such web services, recommender systems have taken more and more place in our lives. From e-commerce (suggest to buyers articles that could interest them) to online advertisement (suggest to users the right contents, matching their preferences), recommender systems are today unavoidable in our daily online journeys.

In a very general way, recommender systems are algorithms aimed at suggesting relevant. items to users (items being movies to watch, text to read, products to buy, or anything else depending on industries). Recommender systems are really critical in some industries as they can generate a huge amount of income when they are efficient or also be a way to stand out significantly from competitors. The main objective for us, in this project is to create a book recommendation system for users.

Approach :

This project is heavily dependent upon analyzing the data and finding the most important Features that will help us to make better recommendations. We will need to pre-process the data well before using it to calculate the similarities between different books.

Our approach to solve this problem is going to be -

- Understanding the dataset, different rows and columns.
- During the EDA, we will try to find popular books and authors, where most of our readers resides. By calculating the number of votes and average ratings, we will find popular books in the data. Statistical methods and Visualizations are going to be very helpful in this EDA process.

- In the pre-processing step, we shall filter the most important features, make necessary transformations, create or omit features as needed. Depending on the features we choose, we will find best approaches for text pre-processing.
- Clustering techniques will be used to identify different groups that books can belong to based on rating and popularity.
- There is a choice to be made when it comes to which recommendation system to use, as the data have both user-interaction related features and also content-related features. So, we shall try both recommender techniques and see how it goes.
- Then we shall conclude the project with an overview and discussion about the observations that we make, about the performance of our models and how this project can prove to be useful from a business standpoint.

Data Description & Attributes :

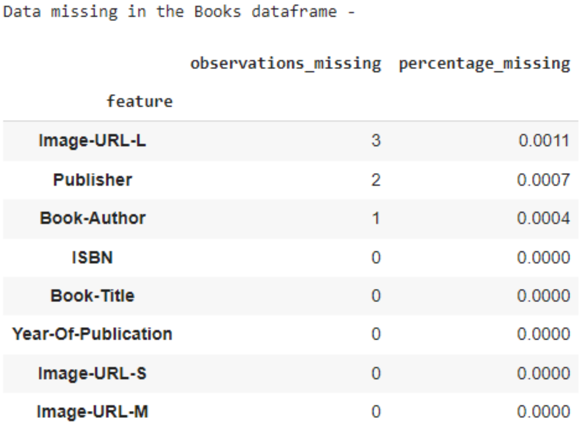
The Book-Crossing dataset comprises 3 files.
Users: Contains the users. Note that user IDs (User-ID) have been anonymized and map to integers. Demographic data is provided (Location, Age) if available. Otherwise, these fields contain NULL values.
Ratings: Contains the book rating information. Ratings (Book-Rating) are either explicit, expressed on a scale from 1-10 (higher values denoting higher appreciation), or implicit, expressed by O.

Books: Books are identified by their respective ISBN. Invalid ISBNs have already been removed from the dataset. Moreover, some content-based information is given (Book-Title, Book-Author, Year-Of-Publication, Publisher), obtained from Amazon Web Services. Note that in the case of several authors, only the first is provided. URLs linking to cover images are also given, appearing in three different flavors (Image-URL-S, Image-URL-M, Image-URL-L), i.e., small, medium, large. These URLs point to the Amazon website.

Exploratory Data Analysis :

Exploratory Data Analysis (EDA) is an approach to analyze the data using visual techniques. It is used to discover trends, patterns, or to check assumptions with the help of statistical summary and graphical representations.

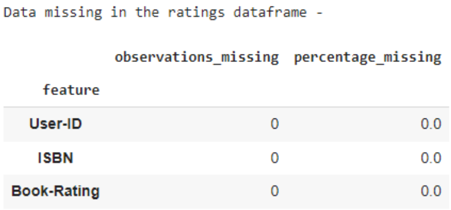
Checking Null Values :



Data missing in the Books dataframe -

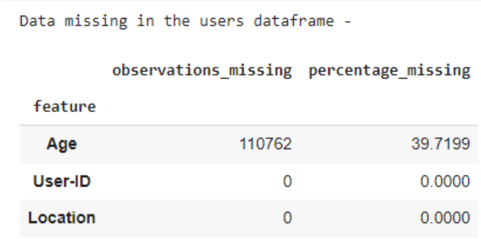
	observations_missing	percentage_missing
feature		
Image-URL-L	3	0.0011
Publisher	2	0.0007
Book-Author	1	0.0004
ISBN	0	0.0000
Book-Title	0	0.0000
Year-Of-Publication	0	0.0000
Image-URL-S	0	0.0000
Image-URL-M	0	0.0000

- The amount of null values are almost negligible in the books data.
- No missing values in the ratings data.
- For the users data, we see that Age column has many null values. almost 40 percent data is missing for this column.



Data missing in the ratings dataframe -

	observations_missing	percentage_missing
feature		
User-ID	0	0.0
ISBN	0	0.0
Book-Rating	0	0.0



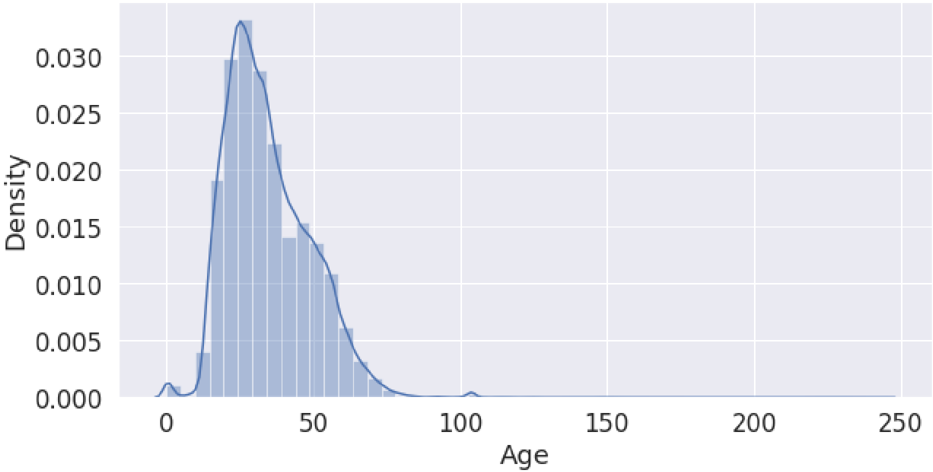
Data missing in the users dataframe -

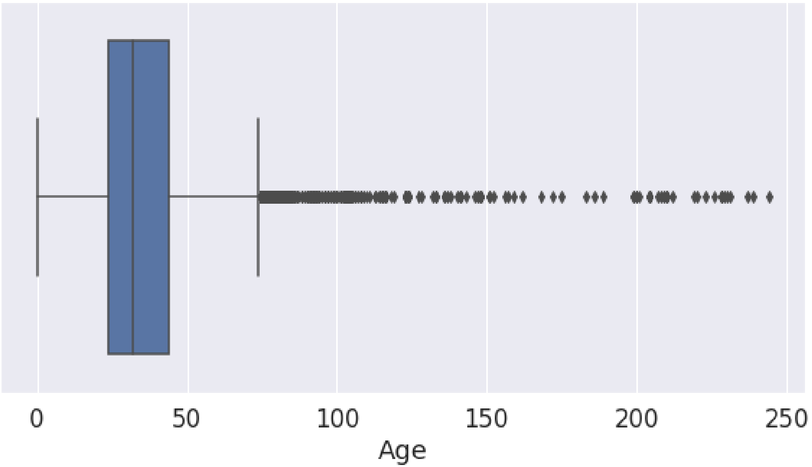
	observations_missing	percentage_missing
feature		
Age	110762	39.7199
User-ID	0	0.0000
Location	0	0.0000

EDA Continued...

Users' **age** will not be useful even if we impute the null values in that column. And we will take a model building approach where we do not need users' age. So, we will have to drop that column.

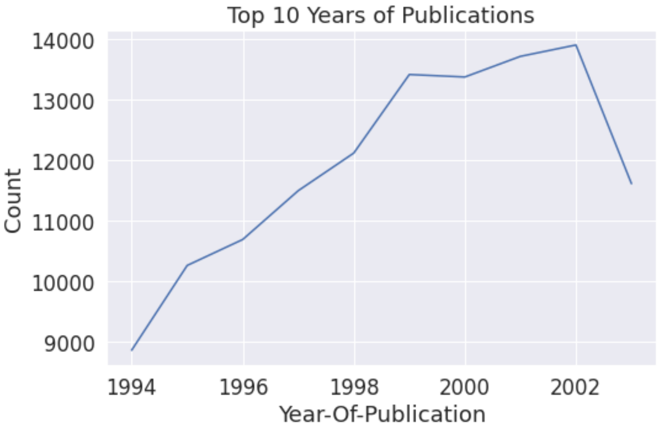
But before dropping, let's check the distribution the rest of the data that is available in this column.





Most of the readers are below 45 years of age. We can also see that there are many nonsensical values like all the values above 85 or 90. So, we can say that this column will not be really useful.

EDA Continued…

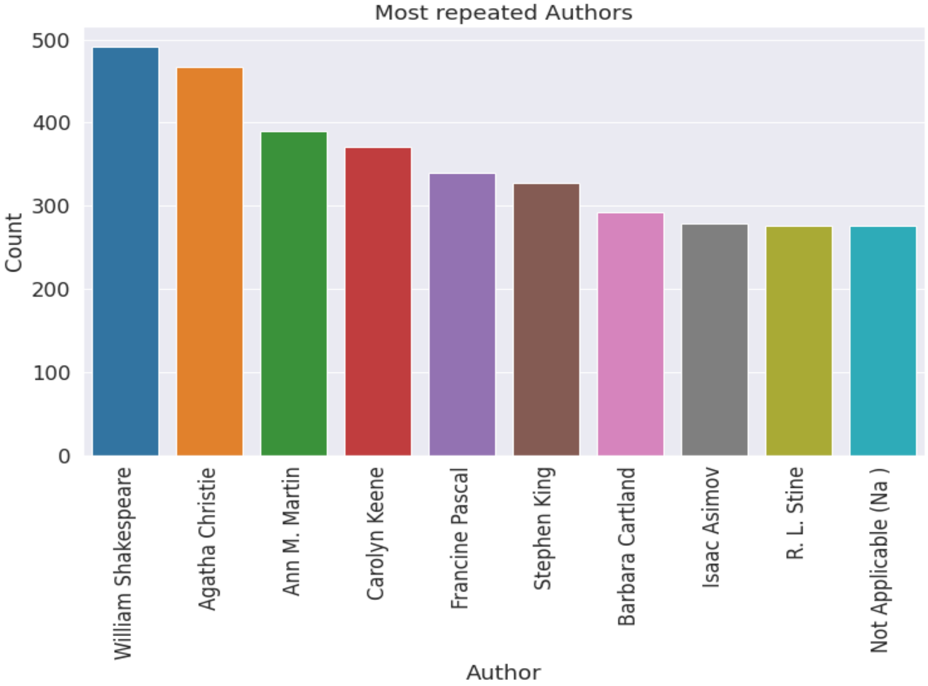


- Most of the books were published after 1994.
- In 2002, highest number of books were published

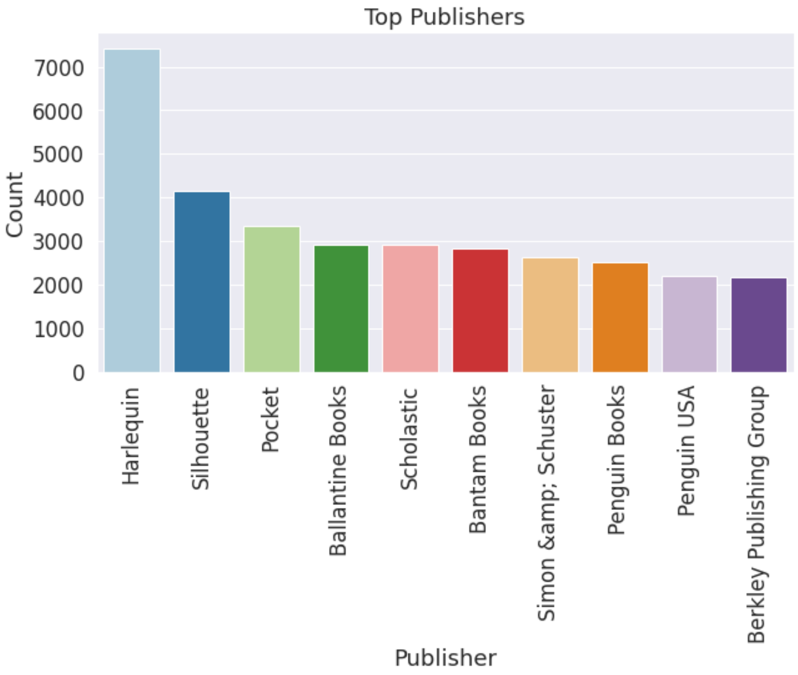
We have same book titles appearing more than once in the data and for those the author is also same.

So we needed to drop those duplicates to count the actual number of times that particular author has repeated.

Most books are written by Shakespeare. Then have Agatha Christie.

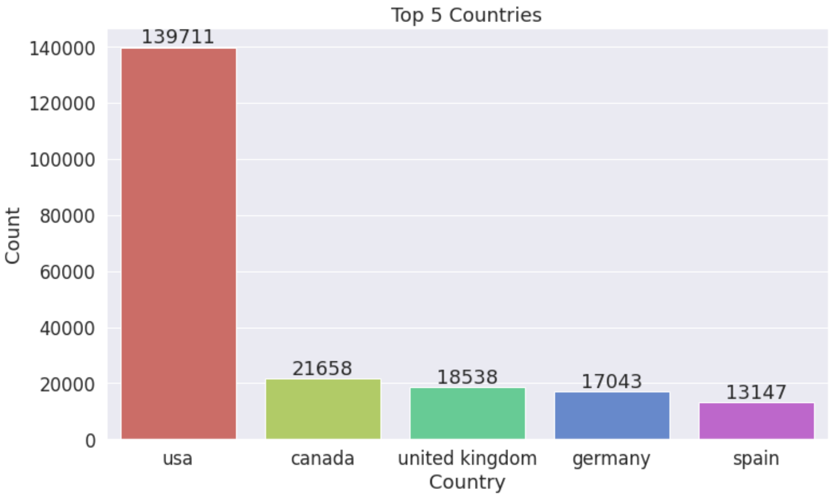


EDA Continued…

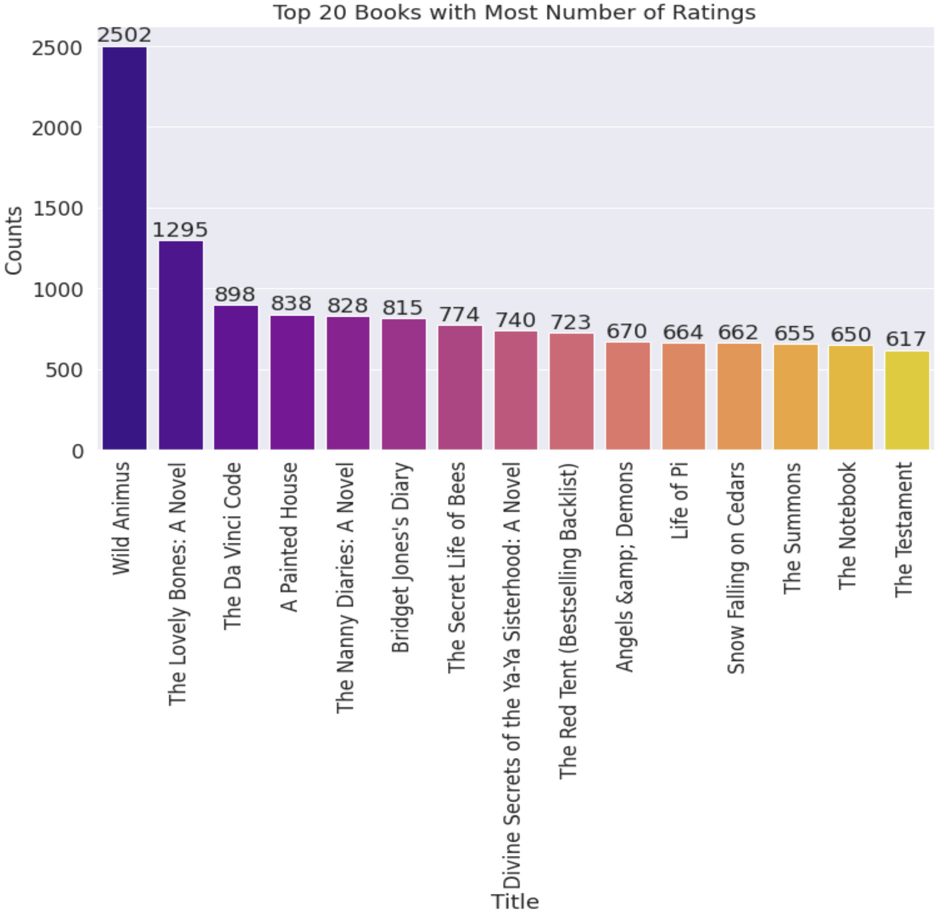


- Harlequinn, Silhouette and Pocket are the top 3 publishers of books in this data.
- The top 5 countries where most of our readers come from are -

1. USA
2. Canada
3. United Kingdom
4. Germany
5. Spain



EDA Continued...



The top 3 most rated books are -

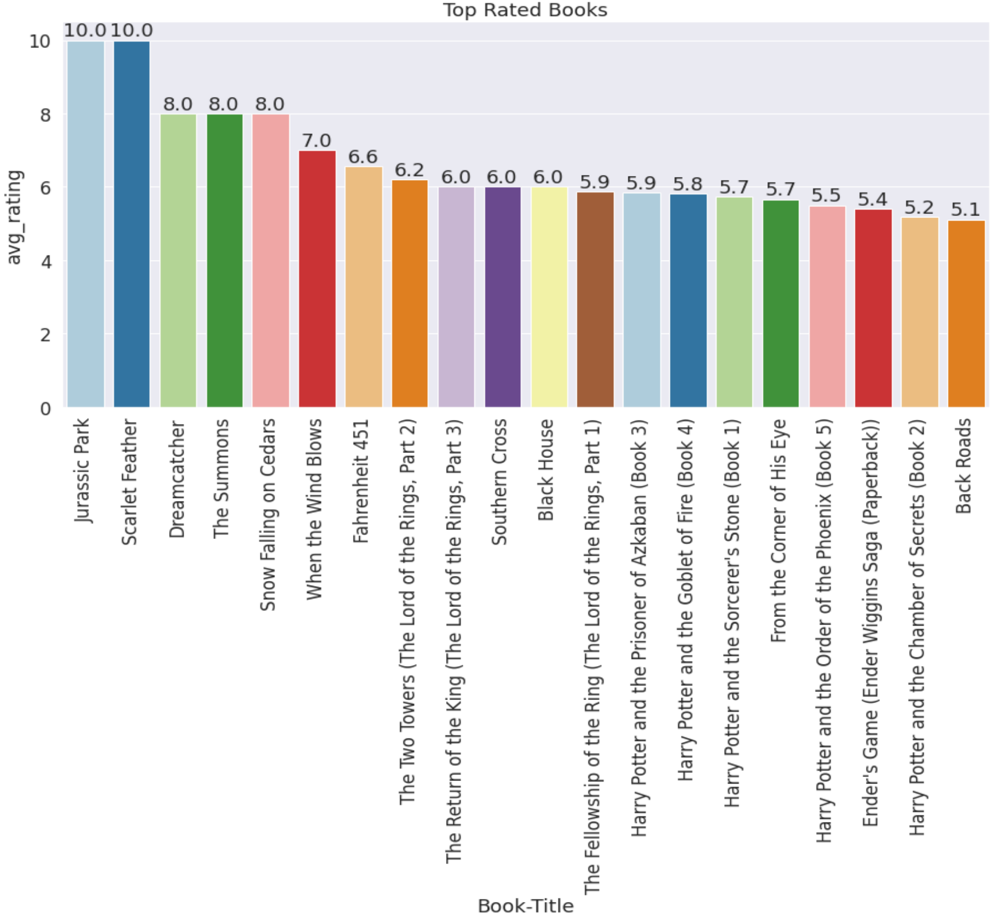
1. Wild Animus
2. The Lovely Bones: A /novel
3. The Da Vinci Code

EDA Continued…

We can see,

- Jurassic Park is the top rated book.
- Harry Potter books, The Hobbit and Lord of the Rings books are also very popular among the readers.

In the top rated books list, we have many great books. We can use these top 50 popular books as a catalogue in Book recommendation website.



Pre-Processing & Feature Engineering:

Feature Creation: Doing groupby and taking a mean of the ratings, we found the average ratings of each book and stored the values by creating a column **avg ratings.** We counted the occurrence of rating and calculated how many times a book has been rated. We stored these values in **num ratings** column.

We created another feature useful for clustering, by multiplying the avg rating and num rating columns.

Feature Selection: For doing the Content based and Collaborative filtering, we selected **Book-Title, Book-Author, Average Ratings, Number of ratings** columns.

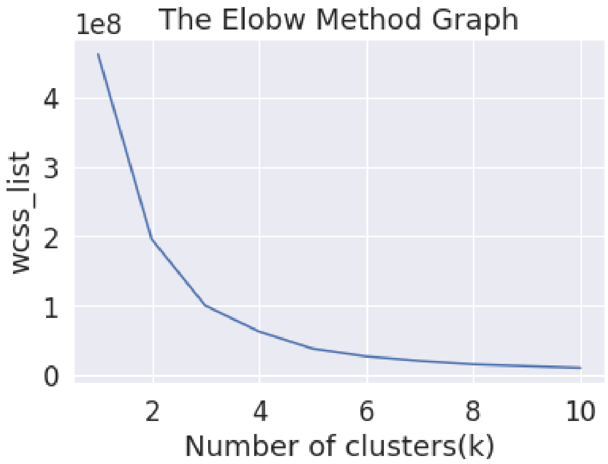
Feature Transformation: We used lowercasing, English stopwords removal and **TFIDF** vectorization on top of the Book-Title and Book-Author columns to get their vector forms that can be used for modelling. TF-IDF stands for Term Frequency Inverse Document Frequency of records. It can be defined as the calculation of how relevant a word in a corpus is to a text. The meaning increases proportionally to the number of times in the text a word appears but is compensated by the word frequency in the corpus (data-set).

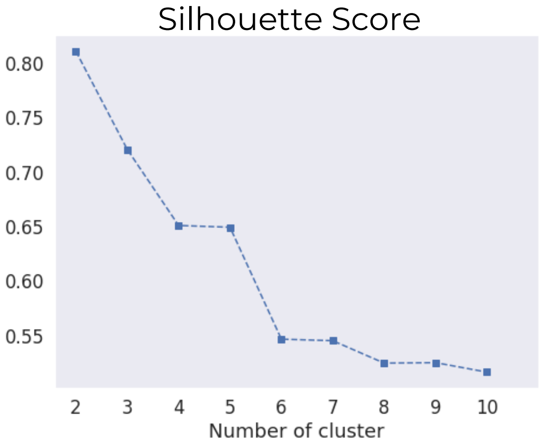
Modelling : Clustering

K-means Clustering :

K-Means is an iterative algorithm that divides the unlabeled dataset into k different clusters in such a way that each dataset belongs only to one group that has similar properties.

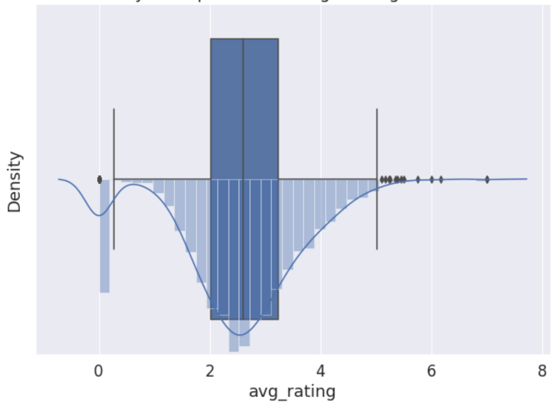
It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.

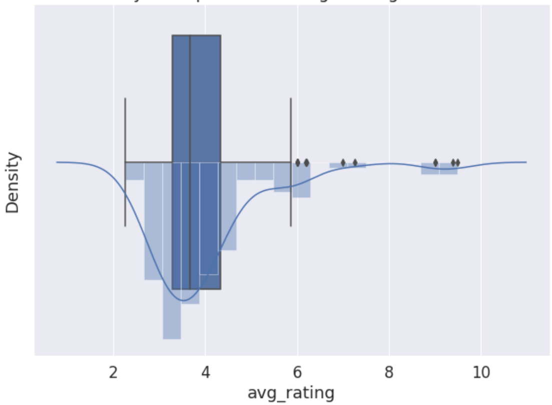


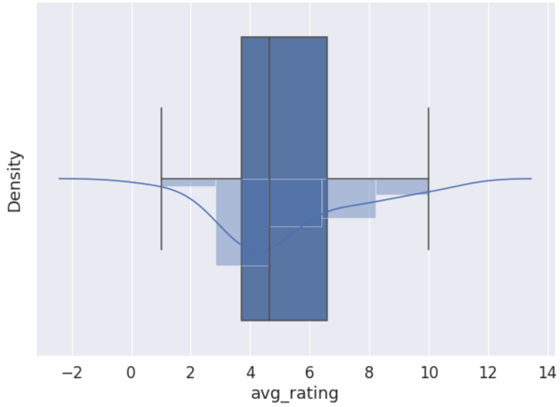


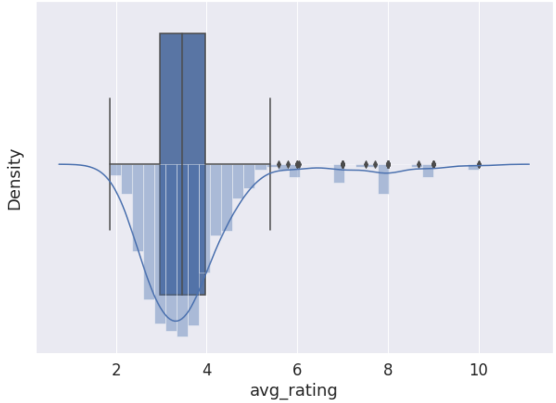
Referring the elbow curve, and evaluating the Silhouette scores, we decided to choose **4** to be the optimal number of clusters.

Clustering Analysis :


Density and Spread of Average Rating in CLuster 0

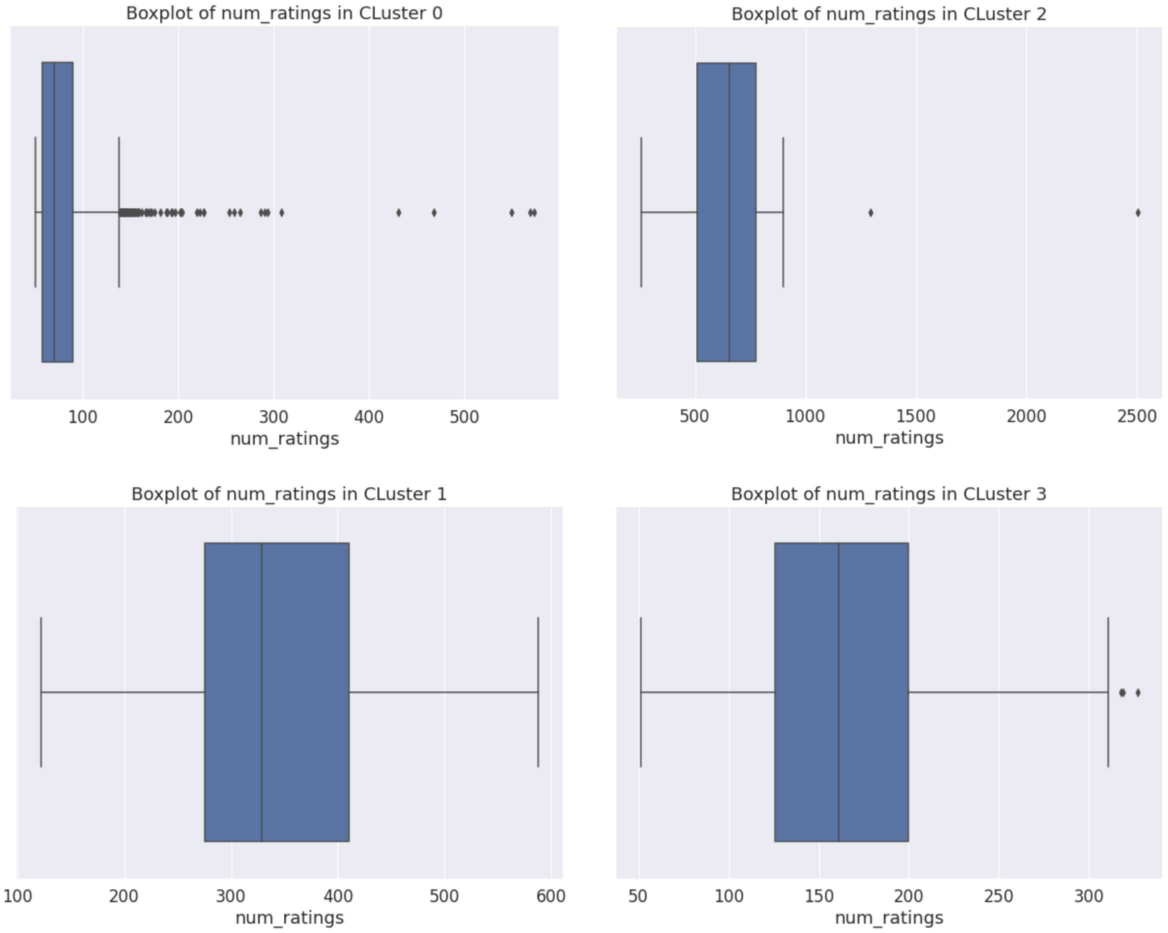

Density and Spread of Average Rating in CLuster 1


Density and Spread of Average Rating in CLuster 2


Density and Spread of Average Rating in CLuster 3

- Books in Cluster 2 have the highest average rating, followed by cluster 1, cluster 3 and then cluster 0.
- Cluster 2 average ratings seems to have no outliers.

Clustering Analysis :



- ➢ Books in Cluster 2 have the highest number of ratings, followed by cluster 1, cluster 3, cluster 0.
- ➢ Cluster 2 books are most popular among readers.
- ➢ We also found during analysis, that **J. K. Rowling** appears in cluster 2 and cluster 1 both, quite prominently.

Modelling : Recommender Systems

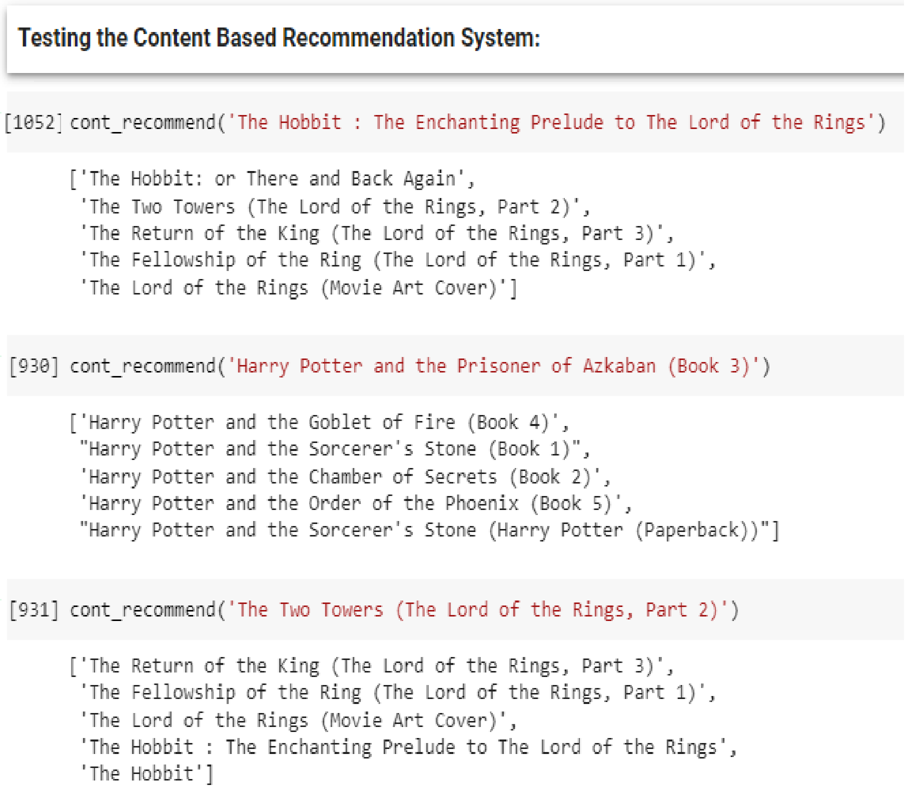
Technique 1 : Content Based Filtering

Content-based filtering uses item features to recommend other items similar to what the user likes, based on their previous actions or explicit feedback.

Content-based filtering uses similarities in products, services, or content features, as well as information accumulated about the user to make recommendations.

We selected **Book-Title, Book-Author** features, then made TFIDF transformation on these columns. Finally, we applied <u>Cosine Similarities</u> to calculate the similarities or distances between each book.

Also, we only selected books that have been rated at least 50 times for this method of modelling, since we want to recommend books that have some popularity among the readers.



Testing the Content Based Recommendation System:

```
[1052] cont_recommend('The Hobbit : The Enchanting Prelude to The Lord of the Rings')

    ['The Hobbit: or There and Back Again',
     'The Two Towers (The Lord of the Rings, Part 2)',
     'The Return of the King (The Lord of the Rings, Part 3)',
     'The Fellowship of the Ring (The Lord of the Rings, Part 1)',
     'The Lord of the Rings (Movie Art Cover)']
```

```
[930] cont_recommend('Harry Potter and the Prisoner of Azkaban (Book 3)')

    ['Harry Potter and the Goblet of Fire (Book 4)',
     "Harry Potter and the Sorcerer's Stone (Book 1)",
     'Harry Potter and the Chamber of Secrets (Book 2)',
     'Harry Potter and the Order of the Phoenix (Book 5)',
     "Harry Potter and the Sorcerer's Stone (Harry Potter (Paperback))"]
```

```
[931] cont_recommend('The Two Towers (The Lord of the Rings, Part 2)')

    ['The Return of the King (The Lord of the Rings, Part 3)',
     'The Fellowship of the Ring (The Lord of the Rings, Part 1)',
     'The Lord of the Rings (Movie Art Cover)',
     'The Hobbit : The Enchanting Prelude to The Lord of the Rings',
     'The Hobbit']
```

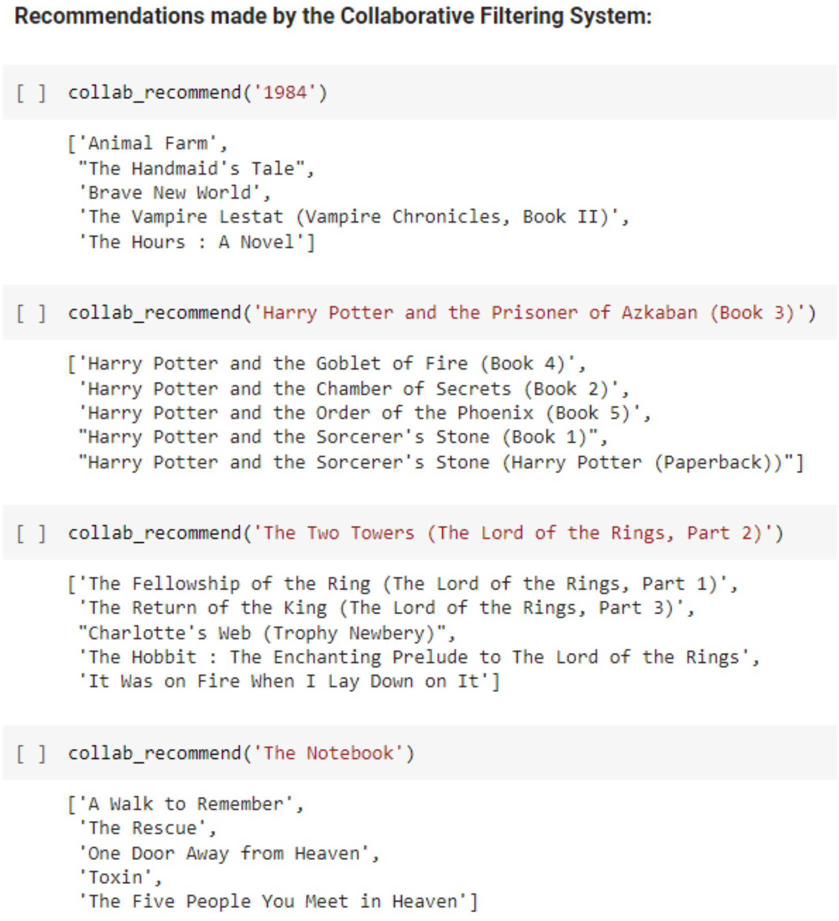
Modelling Continued :

Technique 2 : Collaborative filtering

Collaborative filtering (CF) is the process of filtering or evaluating items through the opinions of other people. CF technology brings together the opinions of large interconnected communities on the web, supporting filtering of substantial quantities of data.

To put it simply, **collaborative filtering** technique is a **recommendation system** that creates a prediction based on a user's previous behaviours.

We selected Avg ratings and num ratings columns to create a user-interaction matrix. We selected only those users who have rated at least 200 books and only those books that have at least 50 votes. That way our collaborative recommender system will be intelligent.



Recommendations made by the Collaborative Filtering System:

```
[ ]  collab_recommend('1984')

     ['Animal Farm',
      "The Handmaid's Tale",
      'Brave New World',
      'The Vampire Lestat (Vampire Chronicles, Book II)',
      'The Hours : A Novel']

[ ]  collab_recommend('Harry Potter and the Prisoner of Azkaban (Book 3)')

     ['Harry Potter and the Goblet of Fire (Book 4)',
      'Harry Potter and the Chamber of Secrets (Book 2)',
      'Harry Potter and the Order of the Phoenix (Book 5)',
      "Harry Potter and the Sorcerer's Stone (Book 1)",
      "Harry Potter and the Sorcerer's Stone (Harry Potter (Paperback))"]

[ ]  collab_recommend('The Two Towers (The Lord of the Rings, Part 2)')

     ['The Fellowship of the Ring (The Lord of the Rings, Part 1)',
      'The Return of the King (The Lord of the Rings, Part 3)',
      "Charlotte's Web (Trophy Newbery)",
      'The Hobbit : The Enchanting Prelude to The Lord of the Rings',
      'It Was on Fire When I Lay Down on It']

[ ]  collab_recommend('The Notebook')

     ['A Walk to Remember',
      'The Rescue',
      'One Door Away from Heaven',
      'Toxin',
      'The Five People You Meet in Heaven']
```

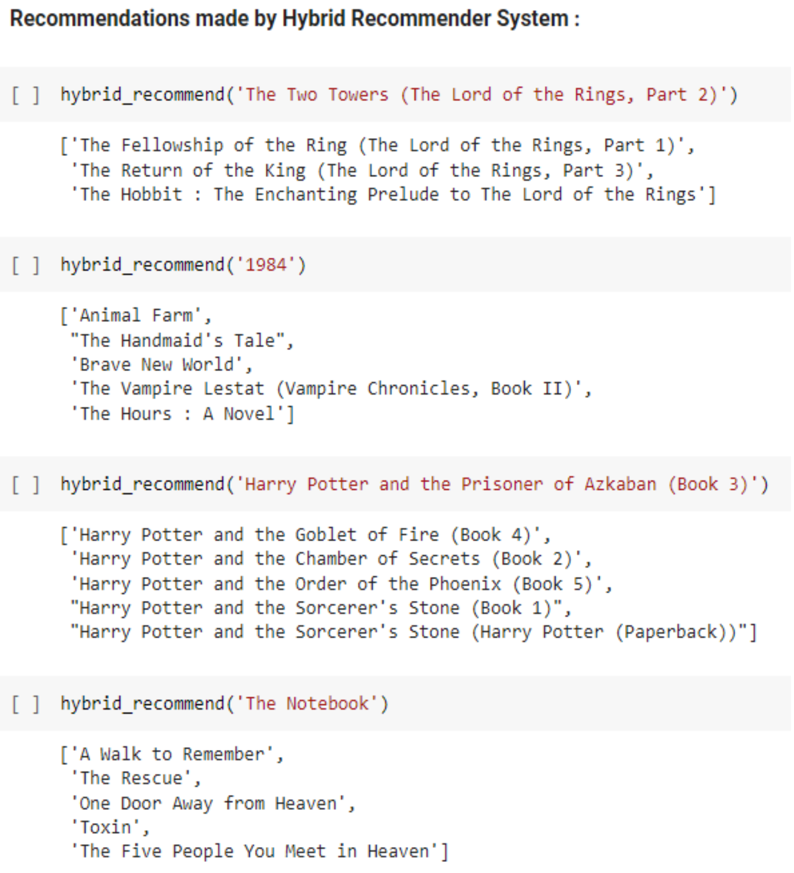
Modelling Continued :

Technique 3 : Hybrid Recommendation System

A hybrid recommendation system is a special type of recommendation system which can be considered as the combination of the content and collaborative filtering method. Combining collaborative and content-based filtering together may help in overcoming the shortcoming we are facing at using them separately and also can be more effective in some cases.

Hybrid recommender system approaches can be implemented in various ways like by using content and collaborative-based methods to generate predictions separately and then combining the prediction or we can just add the capabilities of collaborative-based methods to a content-based approach.

In this project, we took the predictions from both of our previous recommender systems and found the common recommendations.



Recommendations made by Hybrid Recommender System :

```
[ ]  hybrid_recommend('The Two Towers (The Lord of the Rings, Part 2)')

     ['The Fellowship of the Ring (The Lord of the Rings, Part 1)',
      'The Return of the King (The Lord of the Rings, Part 3)',
      'The Hobbit : The Enchanting Prelude to The Lord of the Rings']

[ ]  hybrid_recommend('1984')

     ['Animal Farm',
      "The Handmaid's Tale",
      'Brave New World',
      'The Vampire Lestat (Vampire Chronicles, Book II)',
      'The Hours : A Novel']

[ ]  hybrid_recommend('Harry Potter and the Prisoner of Azkaban (Book 3)')

     ['Harry Potter and the Goblet of Fire (Book 4)',
      'Harry Potter and the Chamber of Secrets (Book 2)',
      'Harry Potter and the Order of the Phoenix (Book 5)',
      "Harry Potter and the Sorcerer's Stone (Book 1)",
      "Harry Potter and the Sorcerer's Stone (Harry Potter (Paperback))"]

[ ]  hybrid_recommend('The Notebook')

     ['A Walk to Remember',
      'The Rescue',
      'One Door Away from Heaven',
      'Toxin',
      'The Five People You Meet in Heaven']
```

Conclusion :

To summarize what we have done,

Steps Performed-

- We performed EDA on the 3 different datasets. Made visualizations and found insights from the data.
- Performed different pre-processing techniques to prepare the data for send it to the recommendation systems and clustering algorithms.
- We made clusters for similar books based on their goodness and popularity among the readers.
- Implemented Content based, Collaborative and Hybrid Recommendation systems that are being able to efficiently recommend similar books.

Observations :

- In the EDA, we found most of the readers are from USA, Canada, UK, Germany and Spain. We found the top authors and publishers.
- Discovered the top rated books and most number of rated books.
- Many records were missing for users' age. Many of the existing information was invalid for this particular feature as well.

Conclusion…

- We noticed that for many books, very small number of ratings were present in the data which does not really give us an idea if those books are actually good or not. Also not all the users should be considered when building a recommendation system. We want genuine, credible and unbiased users. So, we declared thresholds in terms of selecting users and books that will be considered for recommendations.
- We had user-item data and content-feature, both kinds of data available. So, we decided to implement collaborative and content based filtering both.

Suggestions:

- More focus can be given to the users who come from the top 10 countries. These are the users that read and rate the books. So, these are more important users.
- The data regarding the users' age needs to be collected correctly. Then it can help us with performing many different analysis and also to recommend similar content to similar age group people.
- Doing the clustering analysis, we found books in Cluster 2 are both popular and also have better average rating. So, this books can be used for advertising or showcasing as the best collections in the online book-store.

Conclusion…

- Cluster 1 books are little less popular. But these can be used as recommendations. These books have a good chance of gaining popularity among the readers.
- Both Collaborative and Content based filtering are performing efficiently. But if we want to be more sure about the recommendations that are being made, then the Hybrid approach can be used.
- We found the best results for user and book-ratings threshold, that can be found in the final notebook. But as the data grows, we can tweak with these thresholds to consider and recommend newer books as well.

With that, we have reached the end of this project. We hope the analysis that we performed helped to understand people's reading habits. We found areas where more care should be given and explained how the recommendation systems can be used efficiently to provide books recommendations to readers.

Happy Reading…

Thank You