Capstone Project-4

Book Recommendation system

Team Members

1)Ranjith K

2)Rohan A G

3)Lohith D S

Contents

- Introduction
- Exploratory Data Analysis
- Data Processing
- Model To be Used in the project
- Recommendation Predictions
- Results & Conclusions
- Challenges Faced

Problem Statement

BUILD A BOOK RECOMMENDATION SYSTEM
USING PYTHON & MACHINE LEARNING
HARRY POTTER and the CURSED CHILD
J.K. ROWLING
HARRY POTTER and the GOBLET of FIRE
HARRY POTTER and the DEATHLY HALLOWS
HARRY POTTER and the SORCERER'S STONE
J.K. ROWLING
J.K. ROWLING
J.K. ROWLING

- 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 is to create a book recommendation system for users based on popularity and user interests

- 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).

Data Summary:

We are using Book-Crossing dataset to train and test our recommendation system. It contains 1.1 million ratings of 270,000 books by 90,000 users. The ratings are on a scale from 1 to 10. The Book-Crossing dataset comprises 3 files.

In the users dataset we have the following feature variables.

- User-ID (unique for each user)
- Location (contains city, state and country separated by commas)
- Age

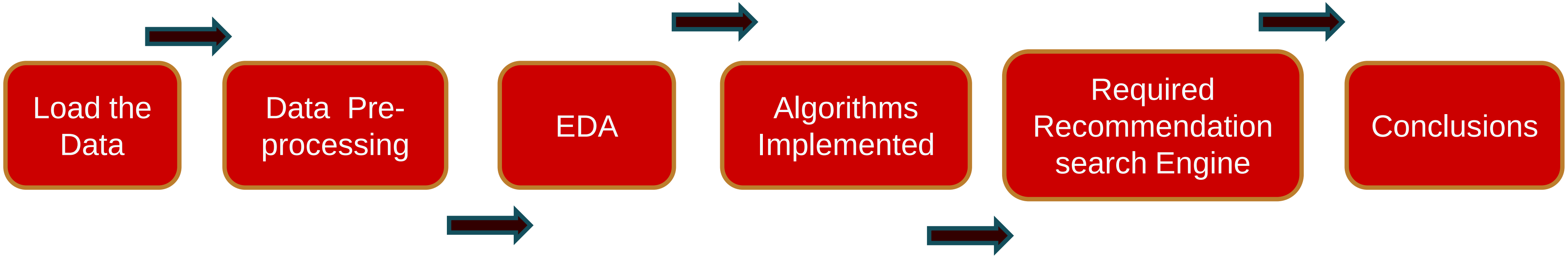
In the books dataset we have the following feature variables.

- ISBN (unique for each book)
- Book-Title
- Book-Author
- Year-Of-Publication
- Publisher
- Image-URL-S
- Image-URL-M
- Image-URL-L

In the ratings dataset we have the following feature variables.

- User-ID
- ISBN
- Book-Rating

Flow of the Project



Load the Data

Data Pre-processing

EDA

Algorithms Implemented

Required Recommendation search Engine

Conclusions

Data Pre-processing

The dataset consists of three tables; Books, Users, and Ratings. Data from all three tables are cleaned and pre-processed separately as defined below briefly:

For Books Dataset:

•Drop all three Image URL features.

•Check for the number of null values in each column. There comes only 3 null values in the table. Replace these three empty cells with ‘Other’.

•Check for the unique years of publications. Two values in the year column are publishers. Also, for three tuples name of the author of the book was merged with the title of the book. Manually set the values for these three above obtained tuples for each of their features using the ISBN of the book.

•Convert the type of the years of publications feature to the integer.

•By keeping the range of valid years as less than 2022 and not 0, replace all invalid years with the mode of the publications that is 2002.

•Upper-casing all the alphabets present in the ISBN column and removal of duplicate rows from the table.

Data Pre-processing

For Users Dataset:

•Check for null values in the table. The Age column has more than 1 lakh null values.

•Check for unique values present in the Age column. There are many invalid ages present like 0 or 244.

•By keeping the valid age range of readers as 10 to 80 replace null values and invalid ages in the Age column with the mean of valid ages.

•The location column has 3 values city, state, and country. These are split into 3 different columns named; City, State, and Country respectively. In the case of null value, ‘other’ has been assigned as the entity value.

•Removal of duplicate entries from the table.

Data Pre-processing

For Ratings Table:

•Check for null values in the table.

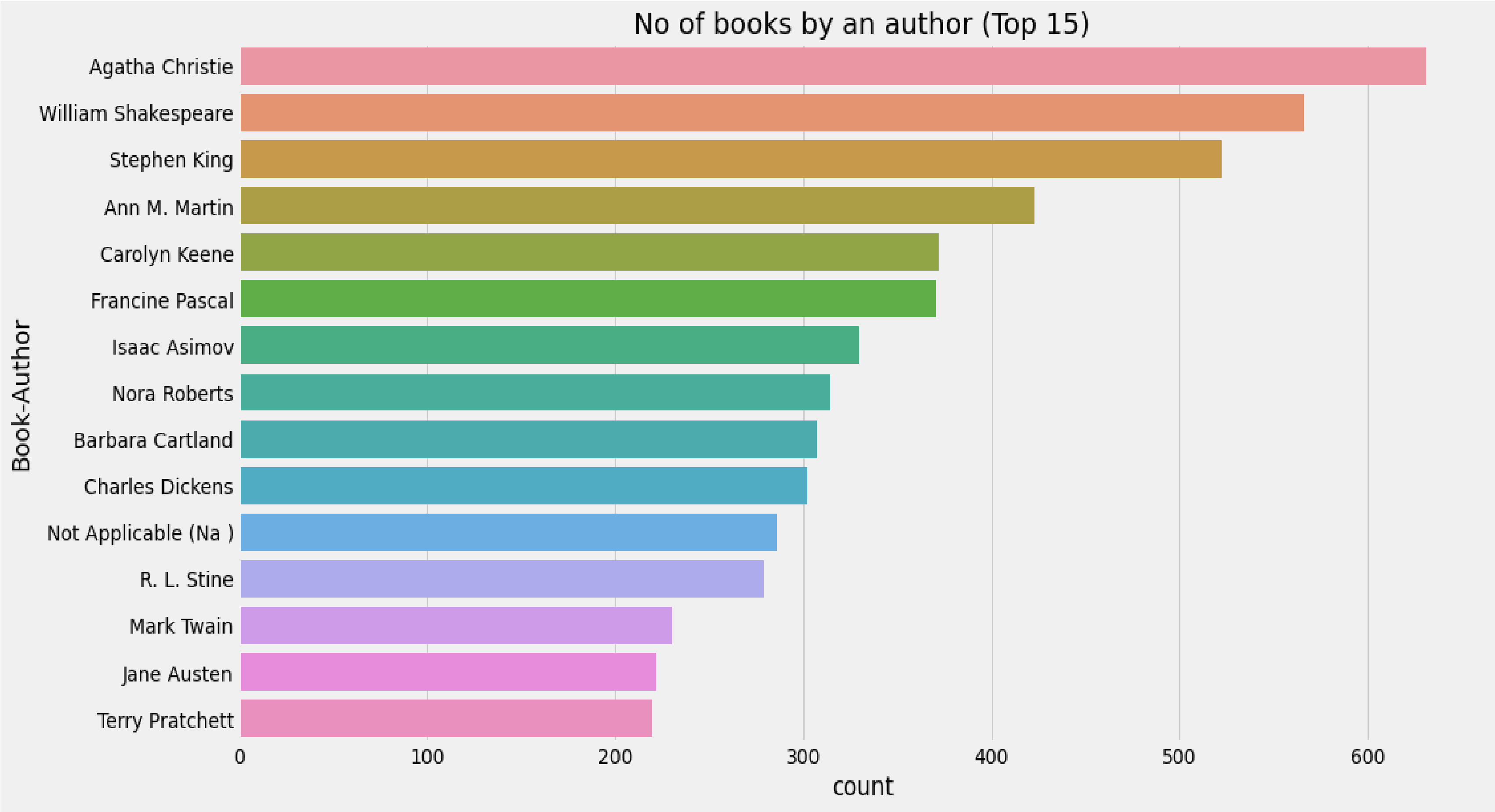
•Check for Rating column and User-ID column to be an integer.

•Removal of punctuation from ISBN column values and if that resulting ISBN is available in the book dataset only then considering else drop tha entity.

•Upper-casing all the alphabets present in the ISBN column.

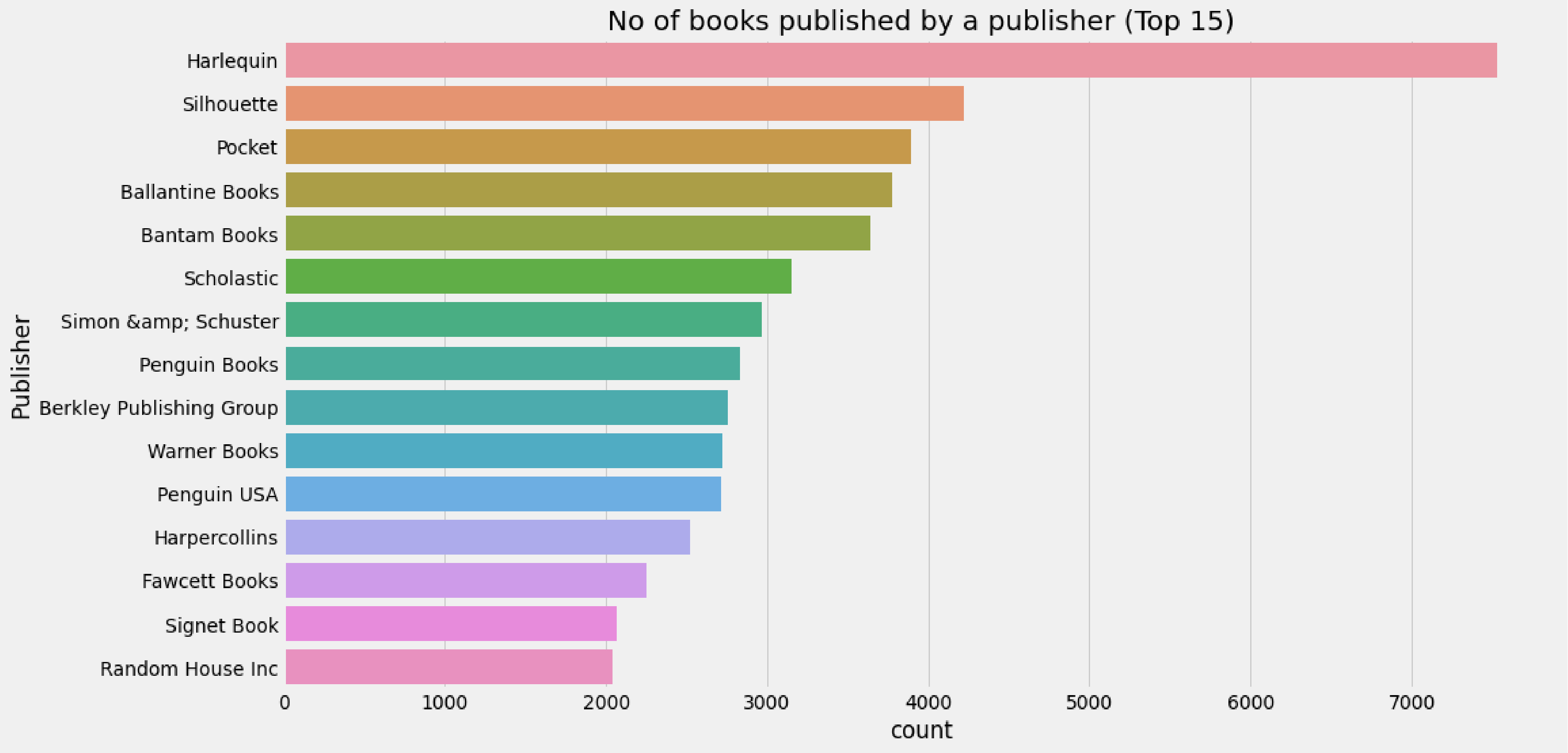
•Removal of duplicate entries from the table.

EDA

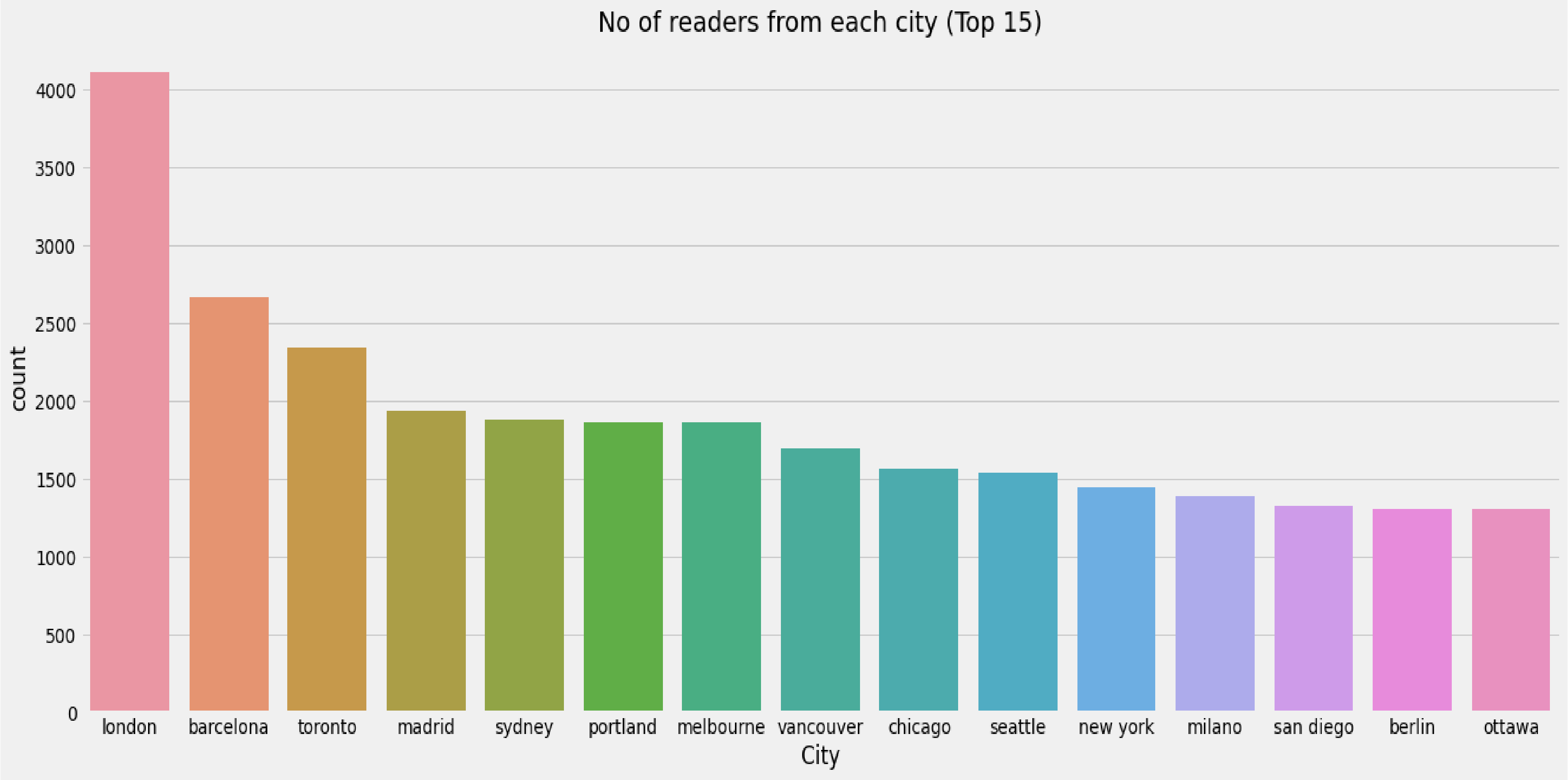


No of books by an author (Top 15)
Book-Author
Agatha Christie
William Shakespeare
Stephen King
Ann M. Martin
Carolyn Keene
Francine Pascal
Isaac Asimov
Nora Roberts
Barbara Cartland
Charles Dickens
Not Applicable (Na)
R. L. Stine
Mark Twain
Jane Austen
Terry Pratchett
0
100
200
300
400
500
600
count

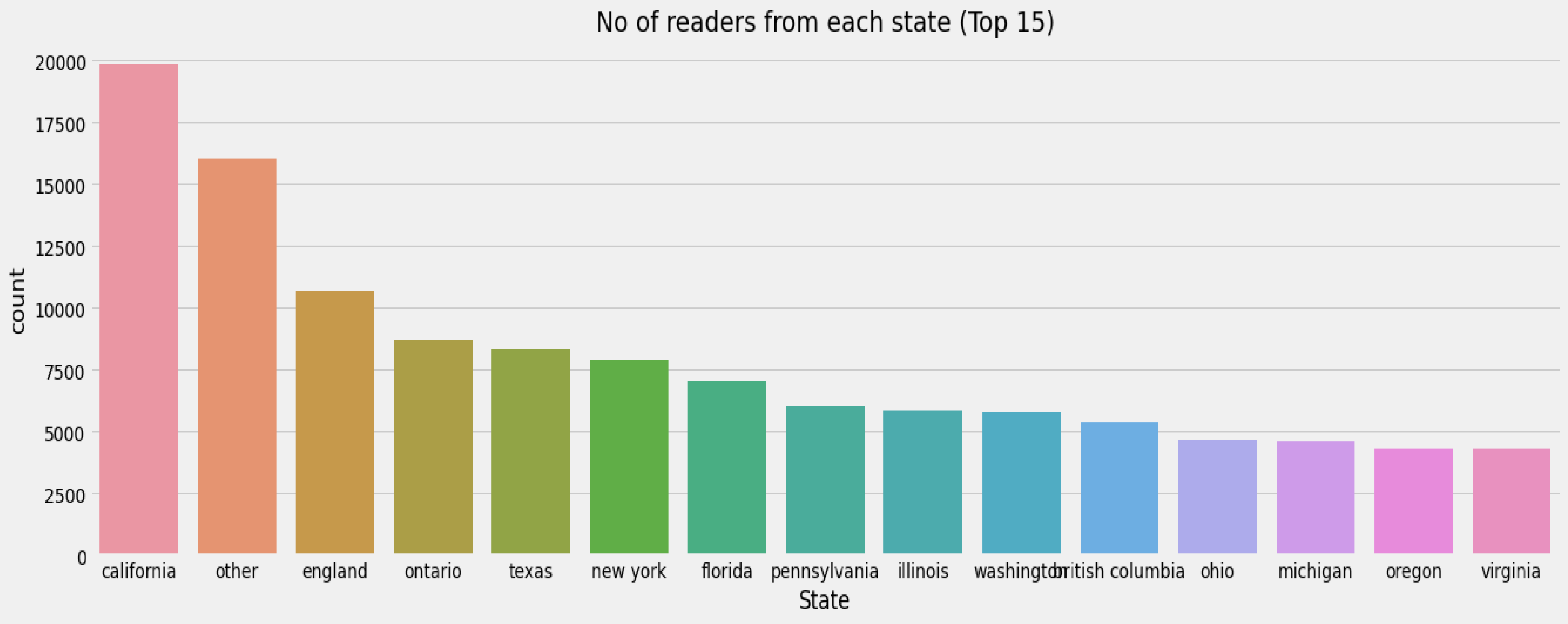
EDA


No of books published by a publisher (Top 15)
Publisher
Harlequin
Silhouette
Pocket
Ballantine Books
Bantam Books
Scholastic
Simon &amp; Schuster
Penguin Books
Berkley Publishing Group
Warner Books
Penguin USA
Harpercollins
Fawcett Books
Signet Book
Random House Inc
0
1000
2000
3000
4000
5000
6000
7000
count

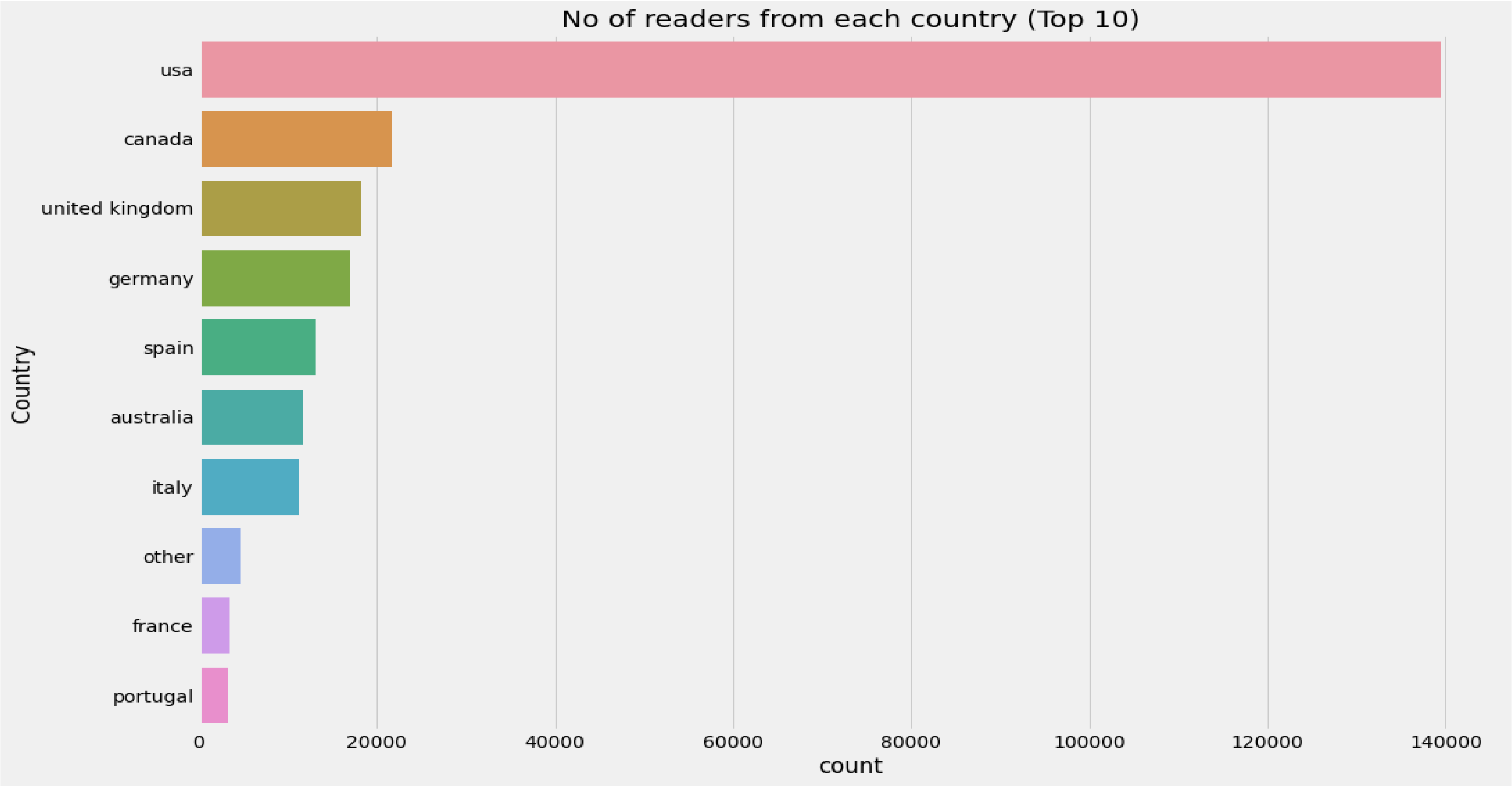
EDA


No of readers from each city (Top 15)
count
4000
3500
3000
2500
2000
1500
1000
500
0
london
barcelona
toronto
madrid
sydney
portland
melbourne
vancouver
chicago
seattle
new york
milano
san diego
berlin
ottawa
City

EDA


No of readers from each state (Top 15)
count
20000
17500
15000
12500
10000
7500
5000
2500
0
california
other
england
ontario
texas
new york
florida
pennsylvania
illinois
washington
british columbia
ohio
michigan
oregon
virginia
State

EDA

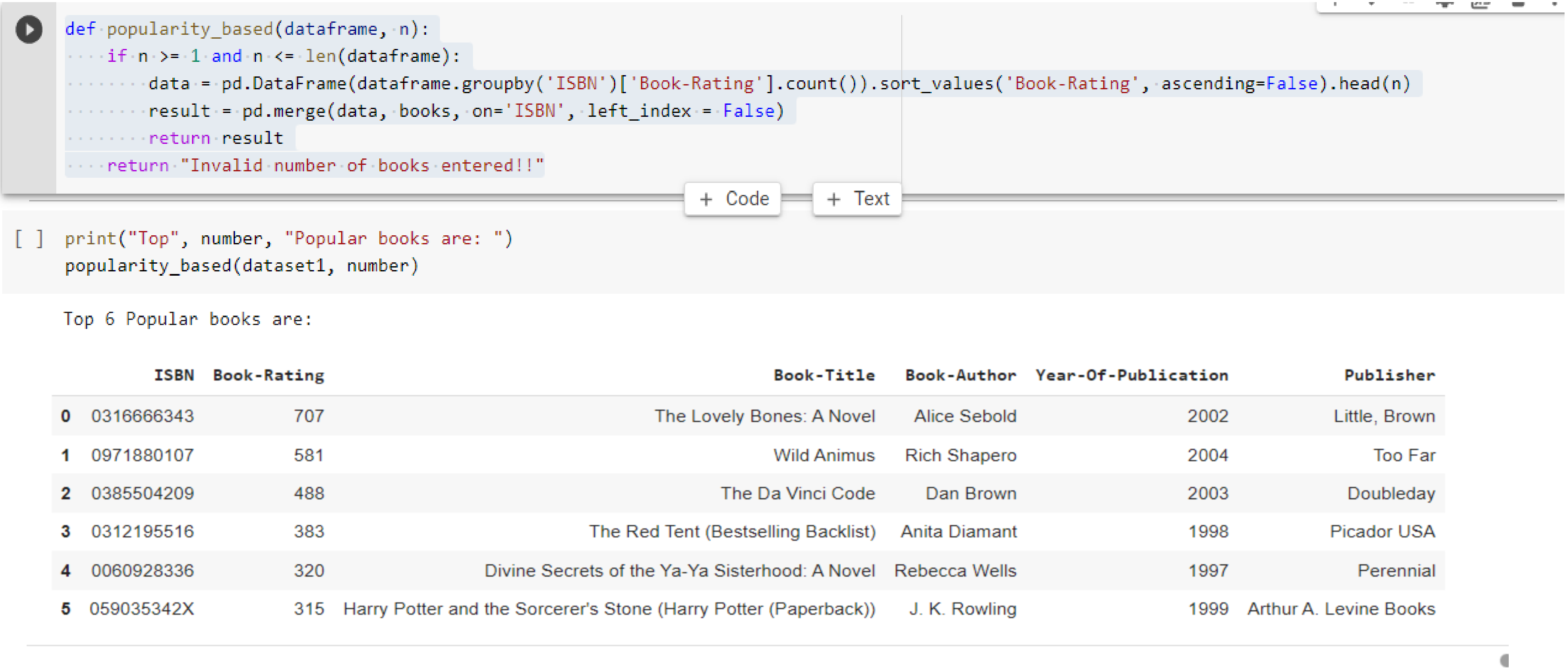

No of readers from each country (Top 10)
usa
canada
united kingdom
germany
spain
australia
italy
other
france
portugal
Country
0
20000
40000
60000
80000
100000
120000
140000
count

Algorithms Implemented:

1. **Popularity Based (Top In whole collection)**
2. **Books by same author, publisher of given book name**
3. **Average Weighted Ratings**
4. **User - Item Collaborative Filtering**
5. **Correlation Based Recommendation System**
6. **Nearest Neighbours Based recommendation System**
7. **SVD(Singular Value Decomposition) Based recommendation System**
8. **Content Based recommendation System**
9. **Hybrid Approach (Content + Collaborative) Using percentile**

Popularity Based (Top In whole collection)

Now let's try to build our first recommendation system based on popularity. These systems check about the product or movie which are in trend or are most popular among the users and directly recommend those.



```
def popularity_based(dataframe, n):
    if n >= 1 and n <= len(dataframe):
        data = pd.DataFrame(dataframe.groupby('ISBN')['Book-Rating'].count()).sort_values('Book-Rating', ascending=False).head(n)
        result = pd.merge(data, books, on='ISBN', left_index = False)
        return result
    return "Invalid number of books entered!!"
```

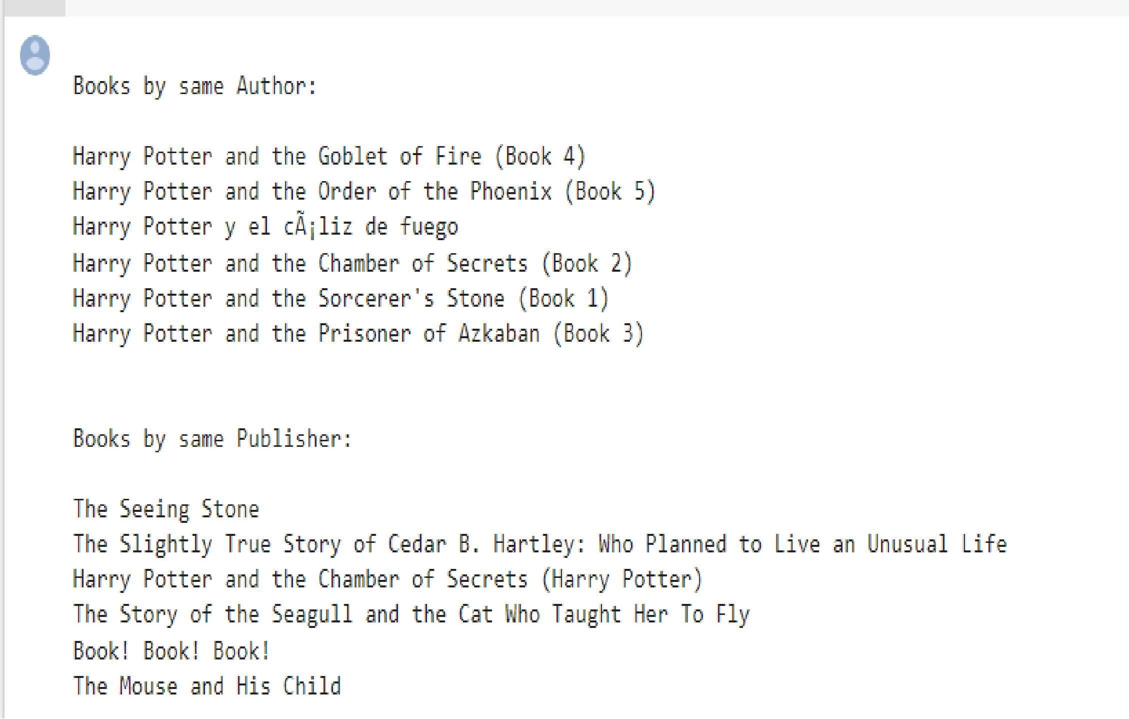
```
print("Top", number, "Popular books are: ")
popularity_based(dataset1, number)
```

```
Top 6 Popular books are:
```

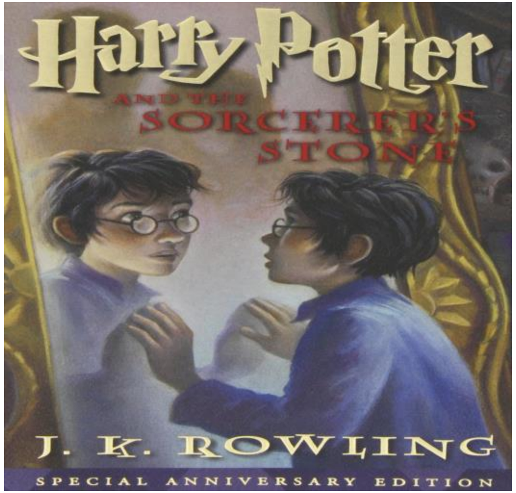
	ISBN	Book-Rating	Book-Title	Book-Author	Year-Of-Publication	Publisher
0	0316666343	707	The Lovely Bones: A Novel	Alice Sebold	2002	Little, Brown
1	0971880107	581	Wild Animus	Rich Shapero	2004	Too Far
2	0385504209	488	The Da Vinci Code	Dan Brown	2003	Doubleday
3	0312195516	383	The Red Tent (Bestselling Backlist)	Anita Diamant	1998	Picador USA
4	0060928336	320	Divine Secrets of the Ya-Ya Sisterhood: A Novel	Rebecca Wells	1997	Perennial
5	059035342X	315	Harry Potter and the Sorcerer's Stone (Harry Potter (Paperback))	J. K. Rowling	1999	Arthur A. Levine Books

Books by same author, publisher of given book name

Recommendation

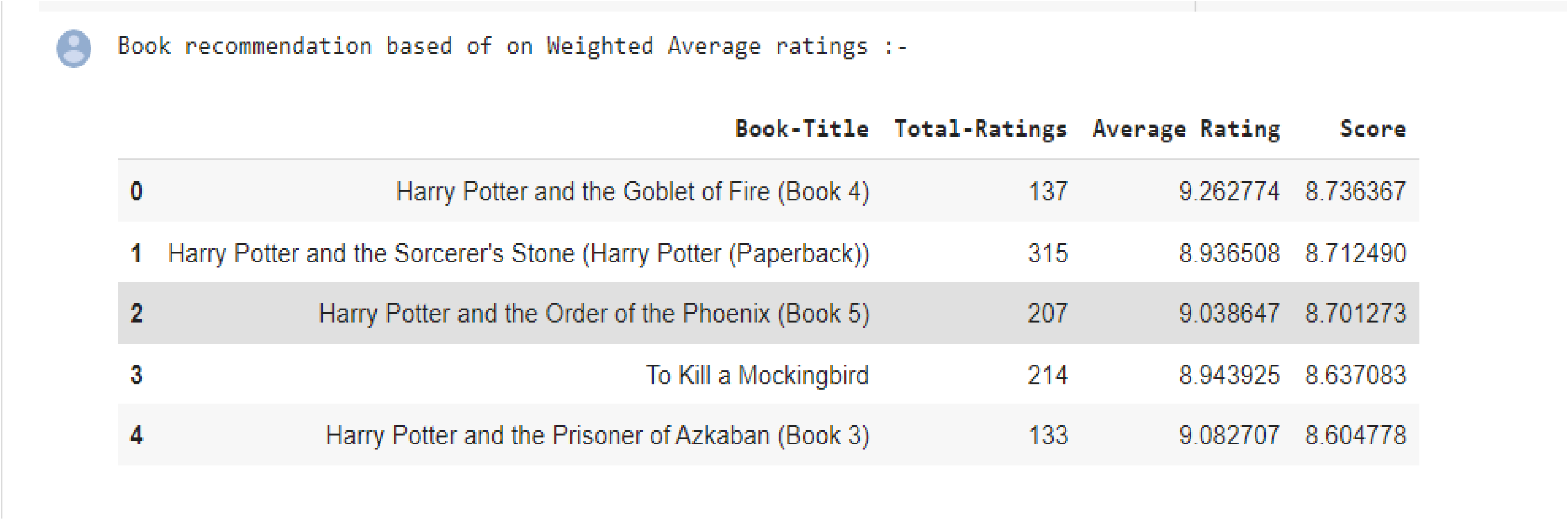


```
Books by same Author:

Harry Potter and the Goblet of Fire (Book 4)
Harry Potter and the Order of the Phoenix (Book 5)
Harry Potter y el cÃ¡liz de fuego
Harry Potter and the Chamber of Secrets (Book 2)
Harry Potter and the Sorcerer's Stone (Book 1)
Harry Potter and the Prisoner of Azkaban (Book 3)



Books by same Publisher:

The Seeing Stone
The Slightly True Story of Cedar B. Hartley: Who Planned to Live an Unusual Life
Harry Potter and the Chamber of Secrets (Harry Potter)
The Story of the Seagull and the Cat Who Taught Her To Fly
Book! Book! Book!
The Mouse and His Child
```



Enter a book name:
Harry Potter and the Sorcerer's Stone (Harry Potter (Paperback))
Enter number of books to recommend: 6

Average Weighted Rating Based Recommendations:



```
Book recommendation based of on Weighted Average ratings :-
```

	Book-Title	Total-Ratings	Average Rating	Score
0	Harry Potter and the Goblet of Fire (Book 4)	137	9.262774	8.736367
1	Harry Potter and the Sorcerer's Stone (Harry Potter (Paperback))	315	8.936508	8.712490
2	Harry Potter and the Order of the Phoenix (Book 5)	207	9.038647	8.701273
3	To Kill a Mockingbird	214	8.943925	8.637083
4	Harry Potter and the Prisoner of Azkaban (Book 3)	133	9.082707	8.604778

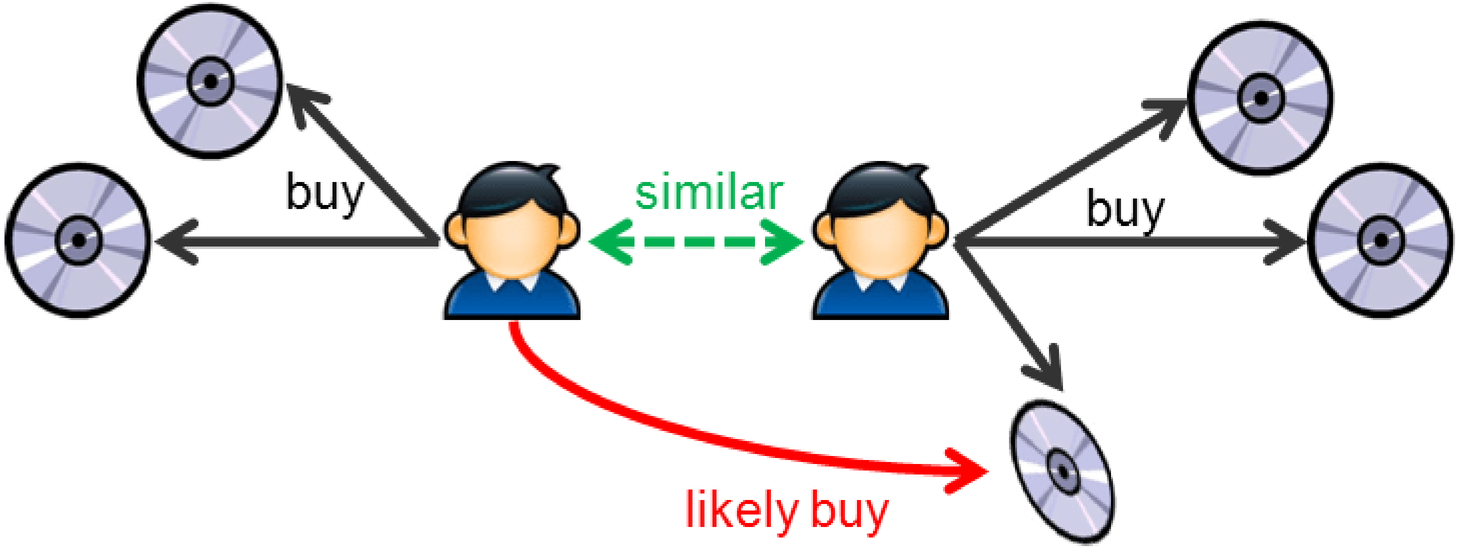
We have calculated the weighted score using the below formula for all the books and recommended the books with the highest score.

$$\text{score} = t/(t+m)*a + m/(m+t)*c$$

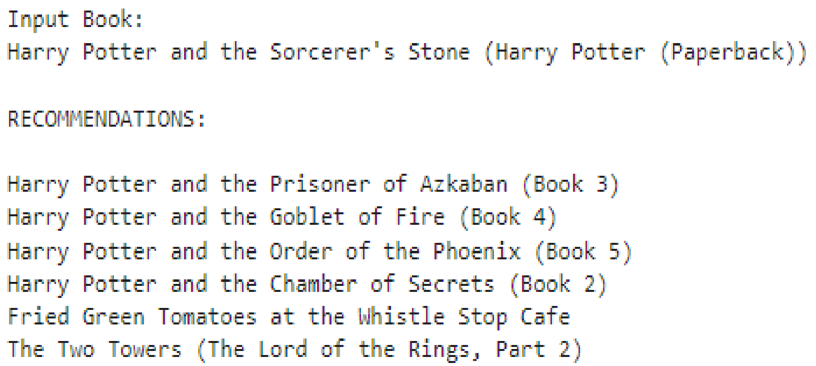
where,
t represents the total number of ratings received by the book,
m represents the minimum number of total ratings considered to be included
a represents the average rating of the book and,
c represents the mean rating of all the books.

Collaborative Filtering

Collaborative Filtering Recommendation System works by considering user ratings and finds cosine similarities in ratings by several users to recommend books. To implement this, we took only those books' data that have at least 50 ratings in all.

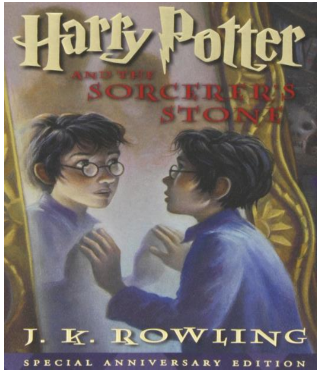


User - Item Collaborative Filtering

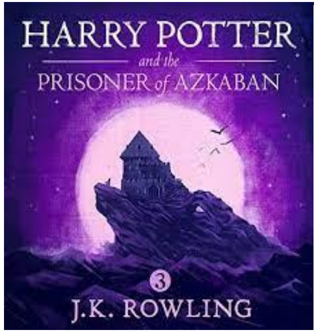


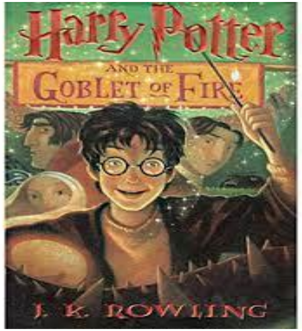
```
Input Book:
Harry Potter and the Sorcerer's Stone (Harry Potter (Paperback))

RECOMMENDATIONS:

Harry Potter and the Prisoner of Azkaban (Book 3)
Harry Potter and the Goblet of Fire (Book 4)
Harry Potter and the Order of the Phoenix (Book 5)
Harry Potter and the Chamber of Secrets (Book 2)
Fried Green Tomatoes at the Whistle Stop Cafe
The Two Towers (The Lord of the Rings, Part 2)
```

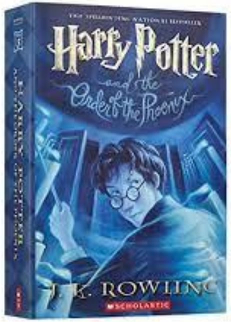
Your Input book is…..

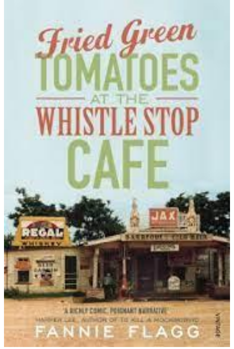


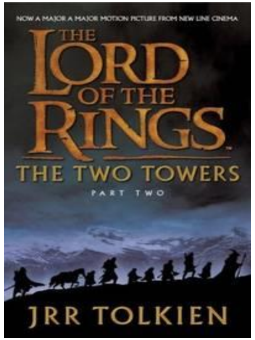
Your Recommendations are……..





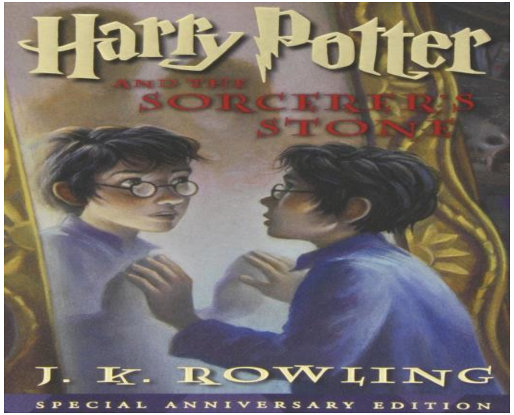




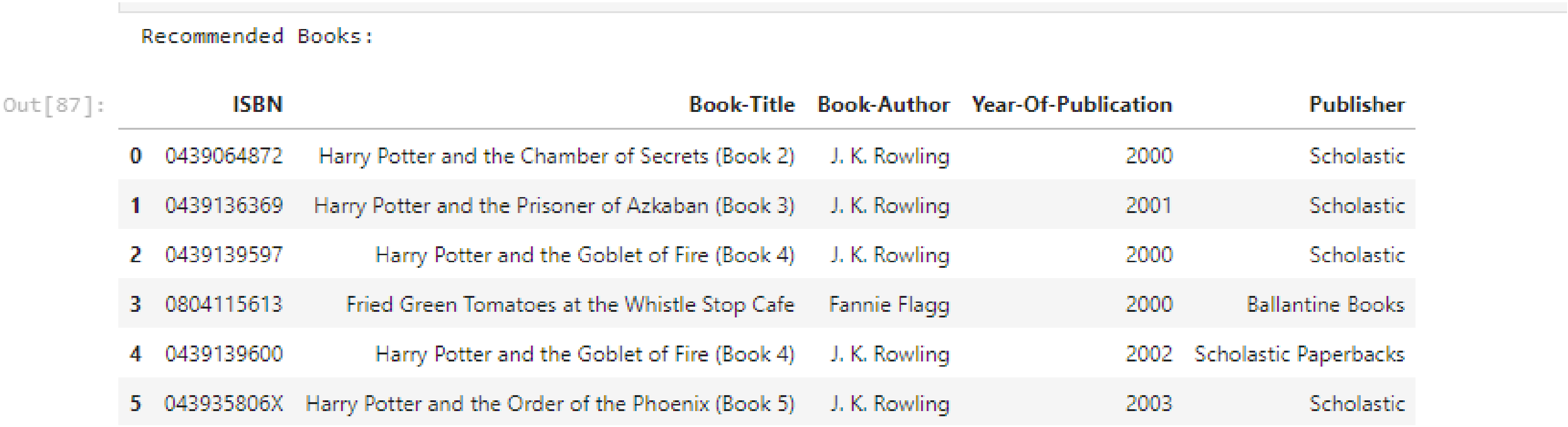


Correlation Based Recommendation System

Your Input book is.....



- ✓ For this model, we have created the correlation matrix considering only those books which have total ratings of more than 50. Then a user-book rating matrix is created. For the input book using the correlation matrix, top books are recommended.



```
Recommended Books:
```

Out[87]:

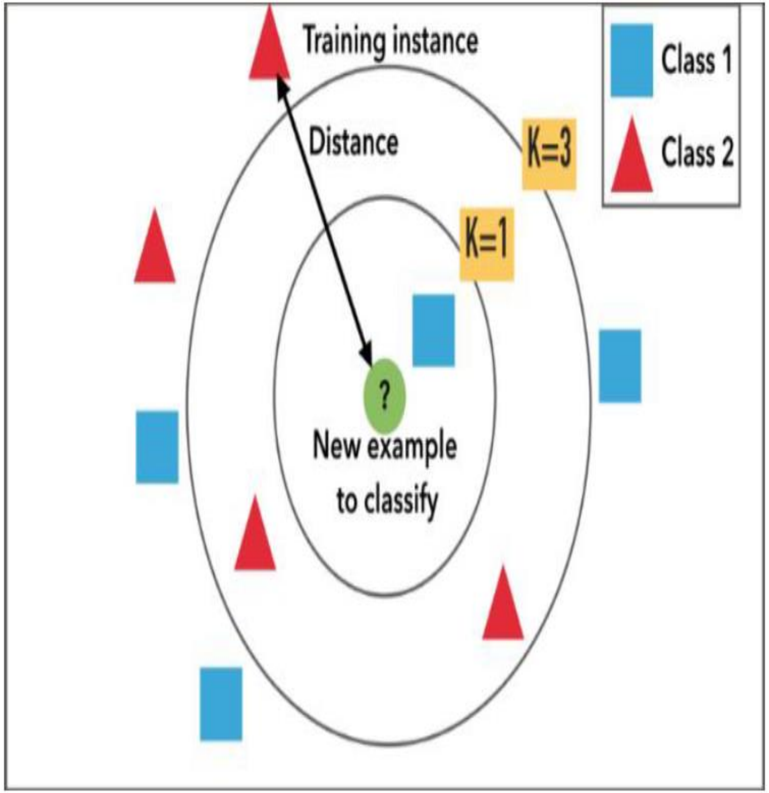
	ISBN	Book-Title	Book-Author	Year-Of-Publication	Publisher
0	0439064872	Harry Potter and the Chamber of Secrets (Book 2)	J. K. Rowling	2000	Scholastic
1	0439136369	Harry Potter and the Prisoner of Azkaban (Book 3)	J. K. Rowling	2001	Scholastic
2	0439139597	Harry Potter and the Goblet of Fire (Book 4)	J. K. Rowling	2000	Scholastic
3	0804115613	Fried Green Tomatoes at the Whistle Stop Cafe	Fannie Flagg	2000	Ballantine Books
4	0439139600	Harry Potter and the Goblet of Fire (Book 4)	J. K. Rowling	2002	Scholastic Paperbacks
5	043935806X	Harry Potter and the Order of the Phoenix (Book 5)	J. K. Rowling	2003	Scholastic

Nearest Neighbours Based recommendation System

KNN (k-Nearest Neighbours) as an algorithm seems to be inspired from real life. The full k-Nearest Neighbours algorithm works much in the way some of us ask for recommendations from our friends.

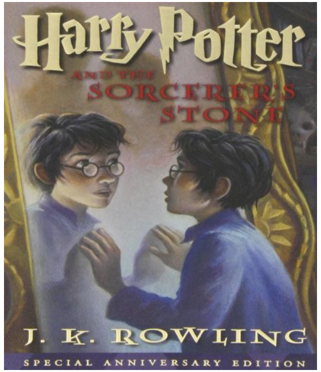
First, we start with people whose taste we feel we share, and then we ask a bunch of them to recommend something to us. If many of them recommend the same thing, we deduce that we'll like it as well.

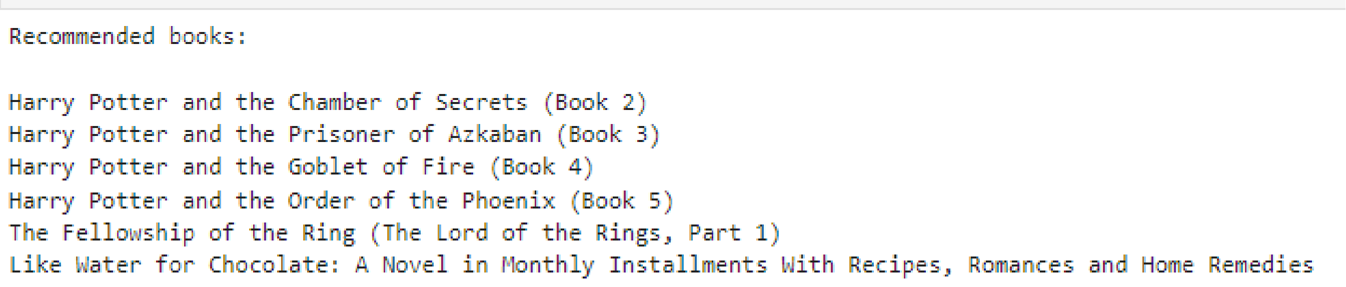
When KNN makes inference about a movie, KNN will calculate the "distance" between the target book and every other book in its database, then it ranks its distances and returns the top K nearest neighbour movies as the most similar book recommendations.



Nearest Neighbours Based recommendation System

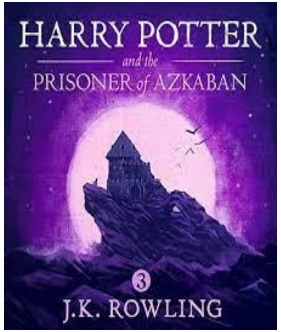
Your Input book is…..

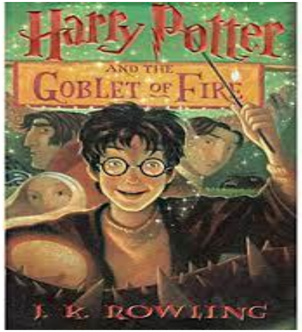


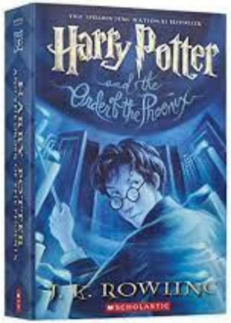


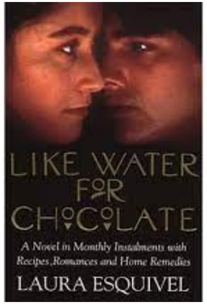
```
Recommended books:

Harry Potter and the Chamber of Secrets (Book 2)
Harry Potter and the Prisoner of Azkaban (Book 3)
Harry Potter and the Goblet of Fire (Book 4)
Harry Potter and the Order of the Phoenix (Book 5)
The Fellowship of the Ring (The Lord of the Rings, Part 1)
Like Water for Chocolate: A Novel in Monthly Installments With Recipes, Romances and Home Remedies
```

Your Recommendations are……..



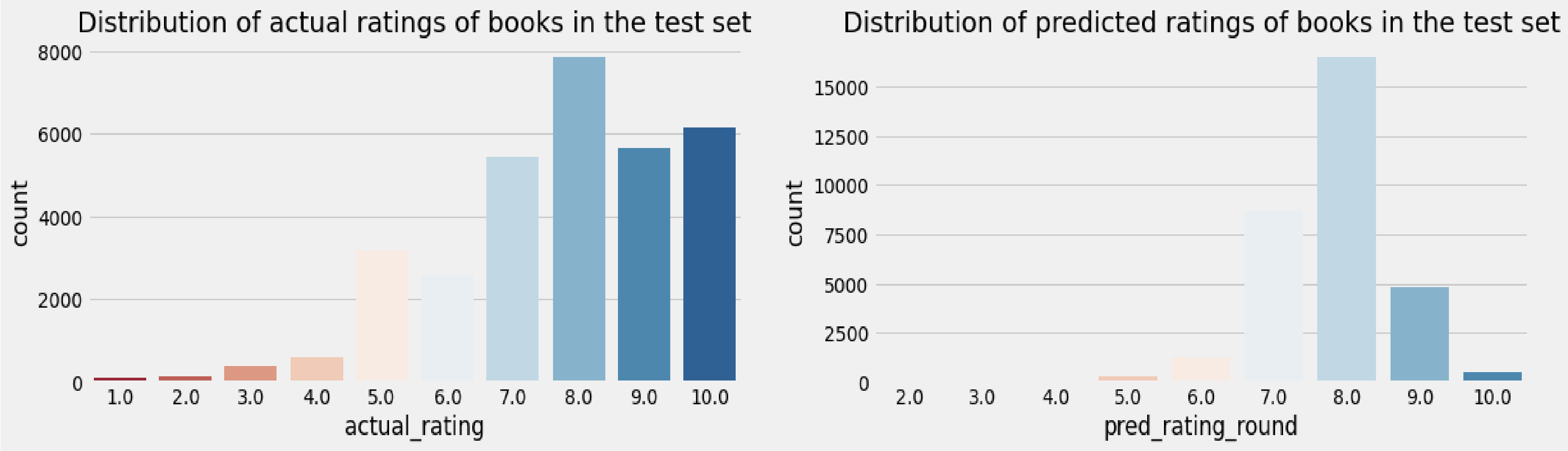




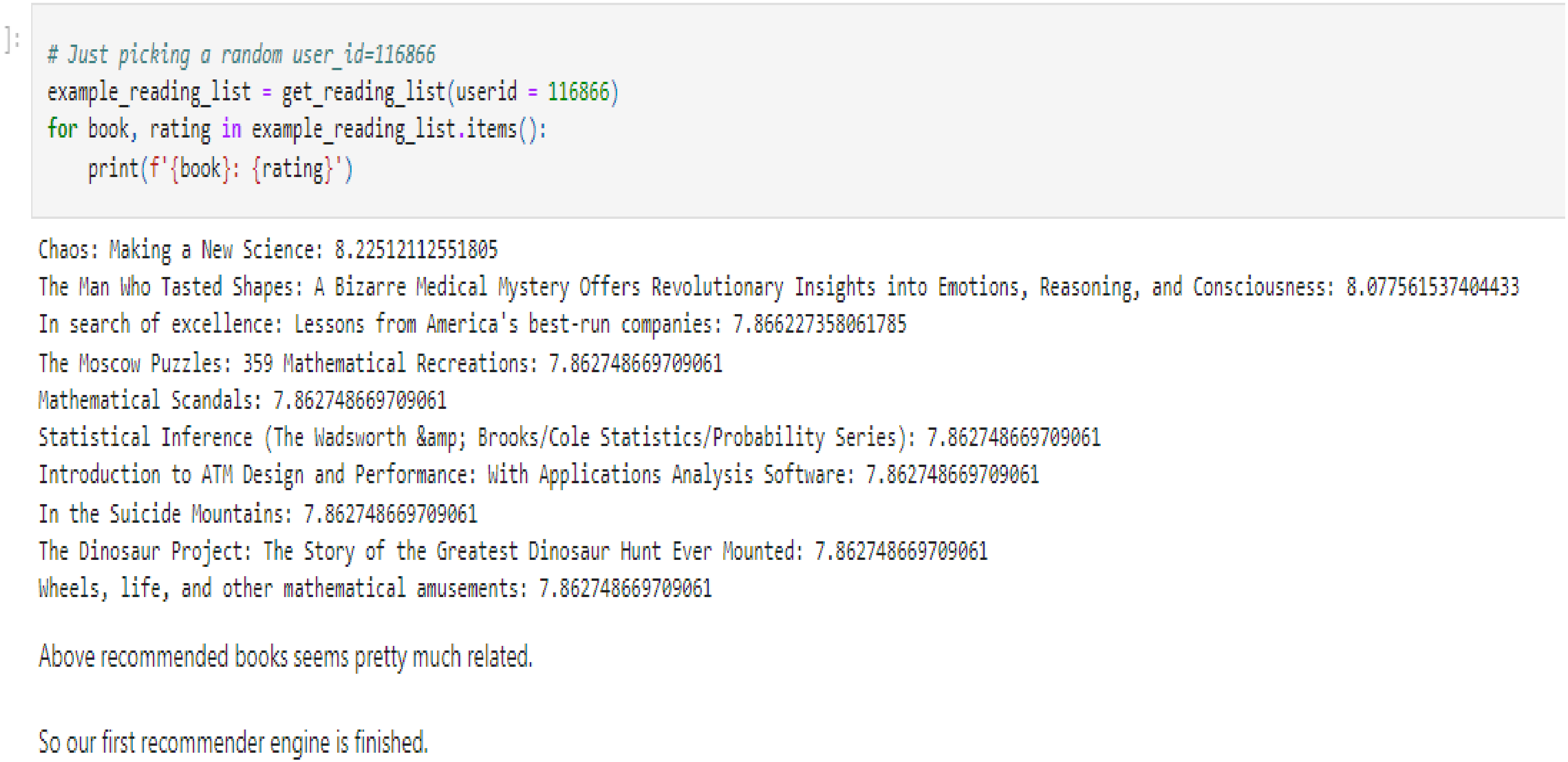

LIKE WATER FOR CHOCOLATE
A Novel in Monthly Instalments with Recipes, Romances and Home Remedies
LAURA ESQUIVEL

SVD Based recommendation System

The SVD(Singular Value Decomposition) is used as a collaborative filtering technique. It uses a matrix structure where each row represents a user, and each column represents an item. The elements of this matrix are the ratings that are given to items by users.



SVD Based recommendation System



```
# Just picking a random user_id=116866
example_reading_list = get_reading_list(userid = 116866)
for book, rating in example_reading_list.items():
    print(f'{book}: {rating}')
```

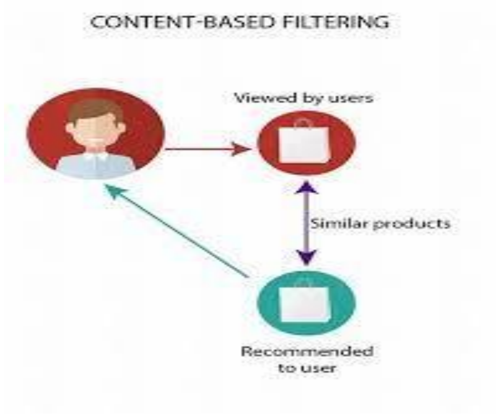
```
Chaos: Making a New Science: 8.22512112551805
The Man Who Tasted Shapes: A Bizarre Medical Mystery Offers Revolutionary Insights into Emotions, Reasoning, and Consciousness: 8.077561537404433
In search of excellence: Lessons from America's best-run companies: 7.866227358061785
The Moscow Puzzles: 359 Mathematical Recreations: 7.862748669709061
Mathematical Scandals: 7.862748669709061
Statistical Inference (The Wadsworth &amp; Brooks/Cole Statistics/Probability Series): 7.862748669709061
Introduction to ATM Design and Performance: With Applications Analysis Software: 7.862748669709061
In the Suicide Mountains: 7.862748669709061
The Dinosaur Project: The Story of the Greatest Dinosaur Hunt Ever Mounted: 7.862748669709061
Wheels, life, and other mathematical amusements: 7.862748669709061
```

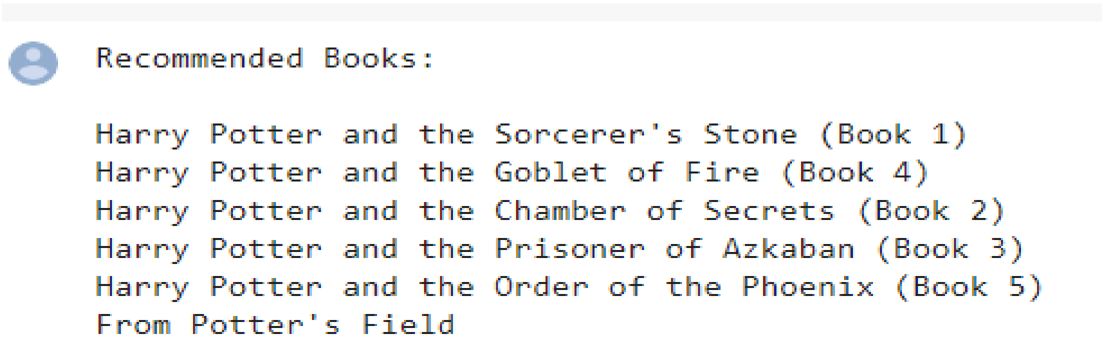
Above recommended books seems pretty much related.

So our first recommender engine is finished.

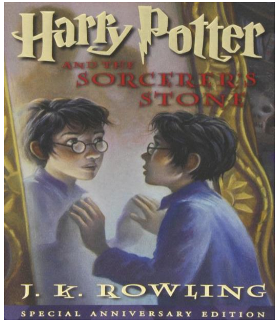
Content Based recommendation System

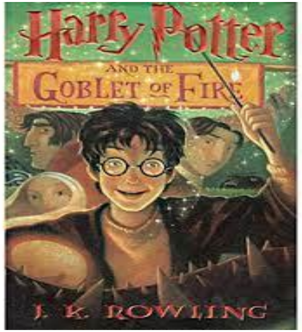
This system recommends books by calculating similarities in Book Titles. For this, TF-IDF feature vectors were created for unigrams and bigrams of Book-Titles; only those books' data has been considered which are having at least 80 ratings.

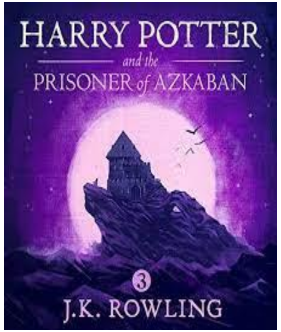


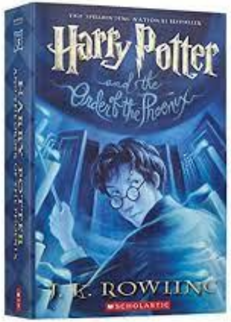


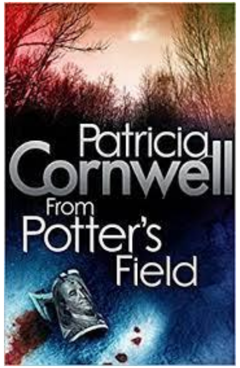
```
Recommended Books:

Harry Potter and the Sorcerer's Stone (Book 1)
Harry Potter and the Goblet of Fire (Book 4)
Harry Potter and the Chamber of Secrets (Book 2)
Harry Potter and the Prisoner of Azkaban (Book 3)
Harry Potter and the Order of the Phoenix (Book 5)
From Potter's Field
```



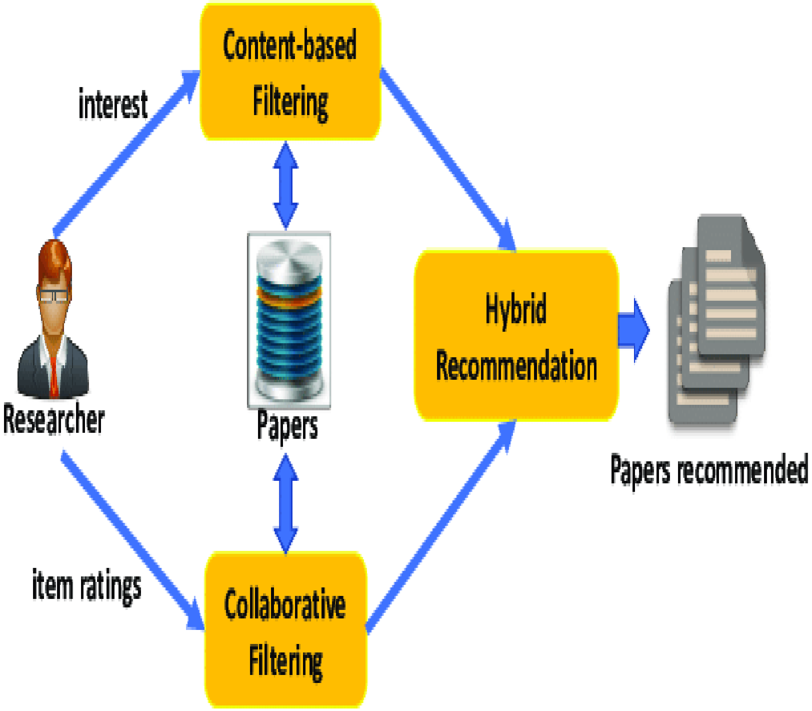








Hybrid Approach (Content + Collaborative)



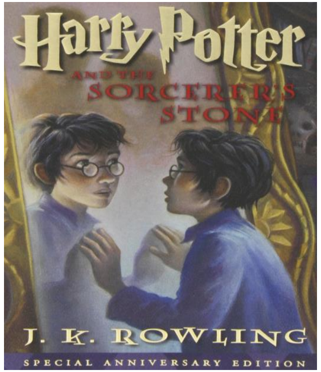
A hybrid recommendation system is **a special type of recommendation system which can be considered as the combination of the content and collaborative filtering method**.

A percentile score is given to the results obtained from both content and collaborative filtering models and is combined to recommend top n books.

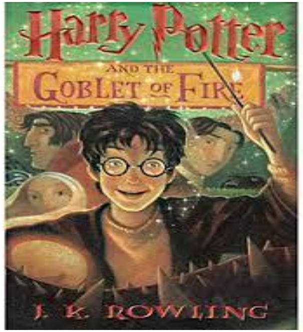
Netflix is a good example of the use of hybrid recommender systems.

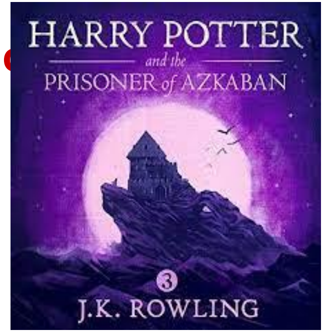
Hybrid Approach (Content + Collaborative)

Your Input book is…

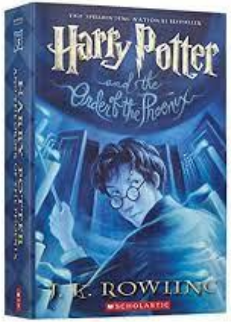


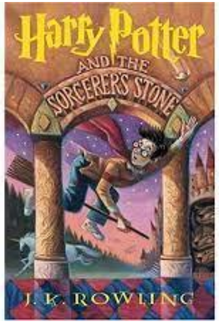
Your Recommendations are…

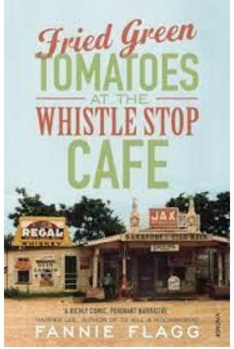




J.K. ROWLING
HARRY POTTER
and the
Chamber of Secrets







Results & Conclusions

•Recommendation system is unturned to exist in the e-commerce businesses with the help of collaborative or content-based filtering to predict different items and yes, users are most satisfied with the products recommended to them.

•While performing Exploratory Data Analysis we observed that almost **42%** of readers with **age-34** read more books compared to other age group of readers.

•Books with publication years are somewhat between **1950 - 2005**.

•Also the readers mostly give 8 ratings(on scale 1-10) to books followed by 10 and 7.

•There are more readers from locations London,england,united kingdom,toronto,ontario,canda compare to other locations.

•KNN model gives good recommendation for books.

Results & Conclusions

- **SVD(Singular value decomposition)** with best accuracy on test data which give stronger recommendations. These results show that our proposed system can remove boring books from the recommendation list more efficiently.

- Popularity based recommendation systems helpful to new users. we don't have data about new user so here popularity based recommendations are more useful

- Content based recommendation systems also performing well , they are given more accurate predictions.

- A hybrid recommendation system was built using the combination of both content-based filtering and collaborative filtering systems. A percentile score is given to the results obtained from both content and collaborative filtering models and is combined

- Most of the companies like Netflix , Amazon are using Hybrid recommendation search engines ,because they are more efficient..

- In Our case also Hybrid approach gives best recommendations...

- Finally this was nice Project to Work .I think i build a most efficient recommendation system

Challenges Faced

- Understanding the metric for evaluation was a challenge as well.
- Decision making on missing value imputations quite challenging.
- Handling of sparsity was a major challenge.

Refences and links

References:

- Almabetter
- Analytics Vidhya
- Kaggle
- Quora
- Stack over flow

Link for the code:

Git hub link:
https://github.com/RanjithK2608/Book-Recommendation-System