

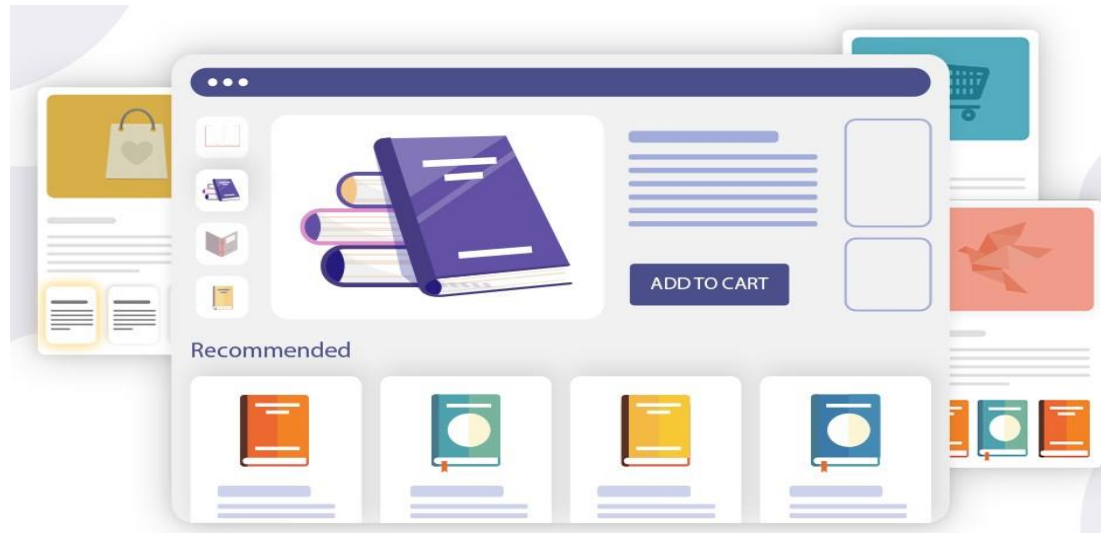
Unsupervised ML Classification

Capstone Project

Book Recommendation System

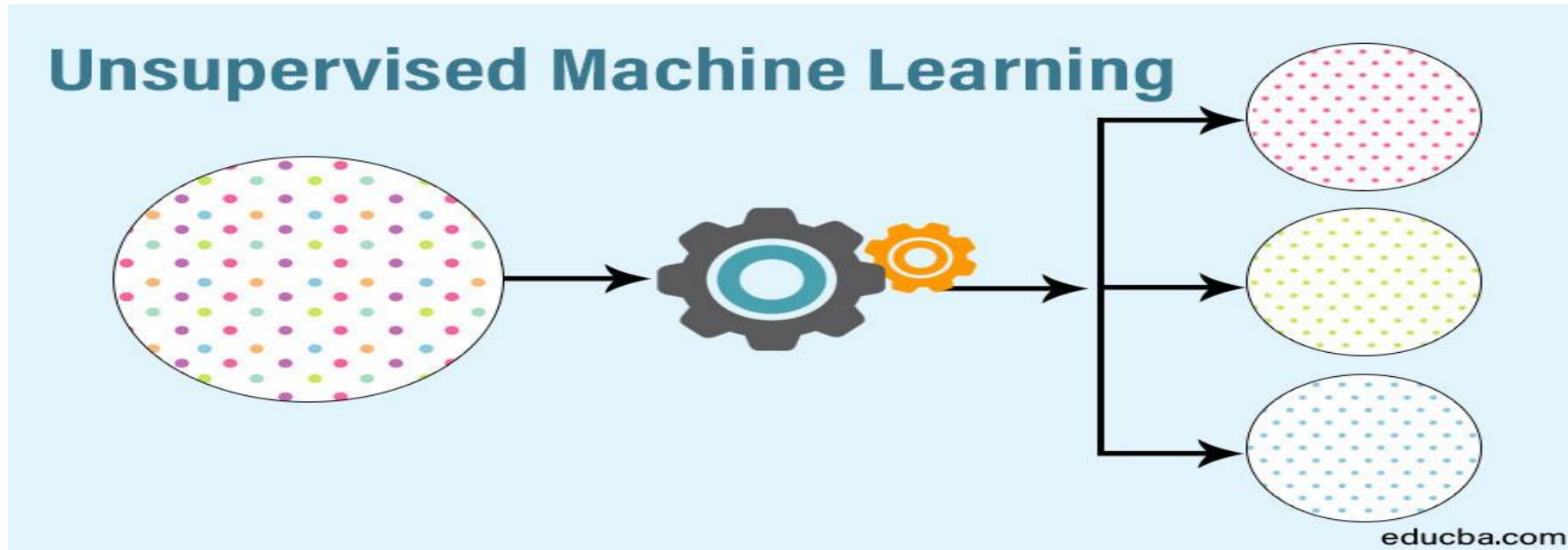
Tanvir Patel

(tanviripatel1998@gmail.com)



Unsupervised learning

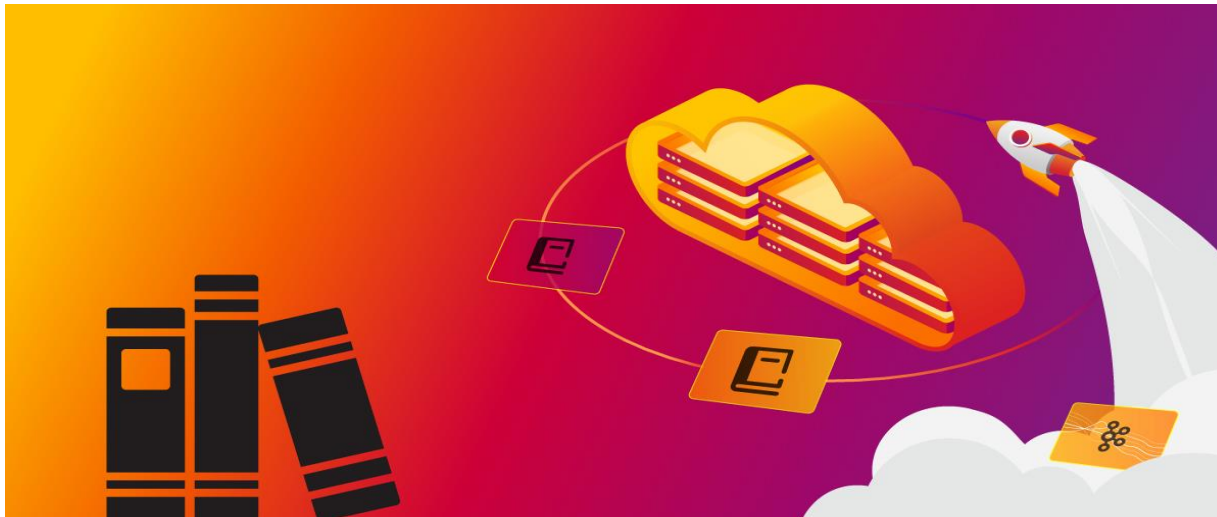
Unsupervised learning is a type of algorithm that learns patterns from untagged data. The hope is that through mimicry, which is an important mode of learning in people, the machine is forced to build a concise representation of its world and then generate imaginative content from it.



Problem Statement

During the last few decades, with the rise of YouTube, Amazon, Netflix, and many other such web services, recommender systems have become much more important in our lives in terms of providing highly personalized and relevant content.

The main objective is to create a recommendation system to recommend relevant books to users based on popularity and user interests.





Content

- Problem statement
- Data Summary
- Data Cleaning
- Imputing missing values
- Analysis of different datasets
- Different Recommendation Model
- Challenges
- Conclusion
- Future Scope

Data Summary

The dataset is comprised of three csv files:: User_df, Books_df, Ratings_df

Users_dataset.

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

Shape of Dataset - (278858, 3)

Books_dataset.

- ISBN (unique for each book)
- Book-Title
- Book-Author
- Year-Of-Publication
- Publisher

Shape of Dataset - (271360, 8)

Ratings_dataset.

- User-ID
- ISBN
- Image-URL-S
- Image-URL-M
- Image-URL-L
- Book-Rating


Shape of Dataset - (1149780, 3)

Data Preprocessing

Data Cleaning 1.

Null Value Imputation:


Books_dataset

✓ 0s  `books.isnull().sum()`

ISBN	0
Book-Title	0
Book-Author	1
Year-Of-Publication	0
Publisher	2
Image-URL-S	0
Image-URL-M	0
Image-URL-L	3

As we can see in the output,
`books.isnull().sum()`
 Function has return Book-Author
 columns 1 NaN ,Publisher columns 2
 NaN and Image-URL-L columns 3NaN
 data are present

Users_dataset

✓ 0s  `users.isnull().sum()`

User-ID	0
Location	0
Age	110762
dtype: int64	

As we can see in the output,
`Users.isnull().sum()`
 Function has return Age columns
 110762 NaN data are present

Ratings_dataset

✓ 0s [94] `ratings.isnull().sum()`

user_id	0
ISBN	0
rating	0
dtype: int64	

As we can see in the output,
`ratings.isnull().sum()`
 Function has Not NaN data are
 present

Features Engineering :

Change the columns name for better reliability

Books_dataset

```
[84] books.rename(columns={'Book-Title':'title','Book-Author':'author','Year-Of-Publication':'year','Publisher':'publisher'},inplace=True)
```

```
books.head()
```

	ISBN	title	author	year	publisher
0	195153448	Classical Mythology	Mark P. O. Morford	2002	Oxford University Press
1	2005018	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada
2	60973129	Decision in Normandy	Carlo D'Este	1991	HarperPerennial
3	374157065	Flu: The Story of the Great Influenza Pandemic...	Gina Bari Kolata	1999	Farrar Straus Giroux
4	393045218	The Mummies of Urumchi	E. J. W. Barber	1999	W. W. Norton & Company

Ratings_dataset

```
[91] ratings.rename(columns={'User-ID':'user_id','Book-Rating':'rating'},inplace=True)
```

```
ratings.head()
```

	user_id	ISBN	rating
0	276725	034545104X	0
1	276726	155061224	5
2	276727	446520802	0
3	276729	052165615X	3
4	276729	521795028	6

Users_dataset

```
users.rename(columns={'User-ID':'user_id','Location':'location','Age':'age'},inplace=True)
```

```
users.head()
```

	user_id	location	age
0	1	nyc, new york, usa	NaN
1	2	stockton, california, usa	18.0
2	3	moscow, yukon territory, russia	NaN
3	4	porto, v.n.gaia, portugal	17.0
4	5	farnborough, hants, united kingdom	NaN

Merge Datasets:

```
ratings_with_books=ratings.merge(books,on='ISBN')
ratings_with_books.head()
```

	user_id	ISBN	rating	title	author	year	publisher
0	276847	446364193	0	Along Came a Spider (Alex Cross Novels)	James Patterson	1993	Warner Books
1	278418	446364193	0	Along Came a Spider (Alex Cross Novels)	James Patterson	1993	Warner Books
2	5483	446364193	0	Along Came a Spider (Alex Cross Novels)	James Patterson	1993	Warner Books
3	7346	446364193	0	Along Came a Spider (Alex Cross Novels)	James Patterson	1993	Warner Books
4	8362	446364193	0	Along Came a Spider (Alex Cross Novels)	James Patterson	1993	Warner Books

Create New Features:

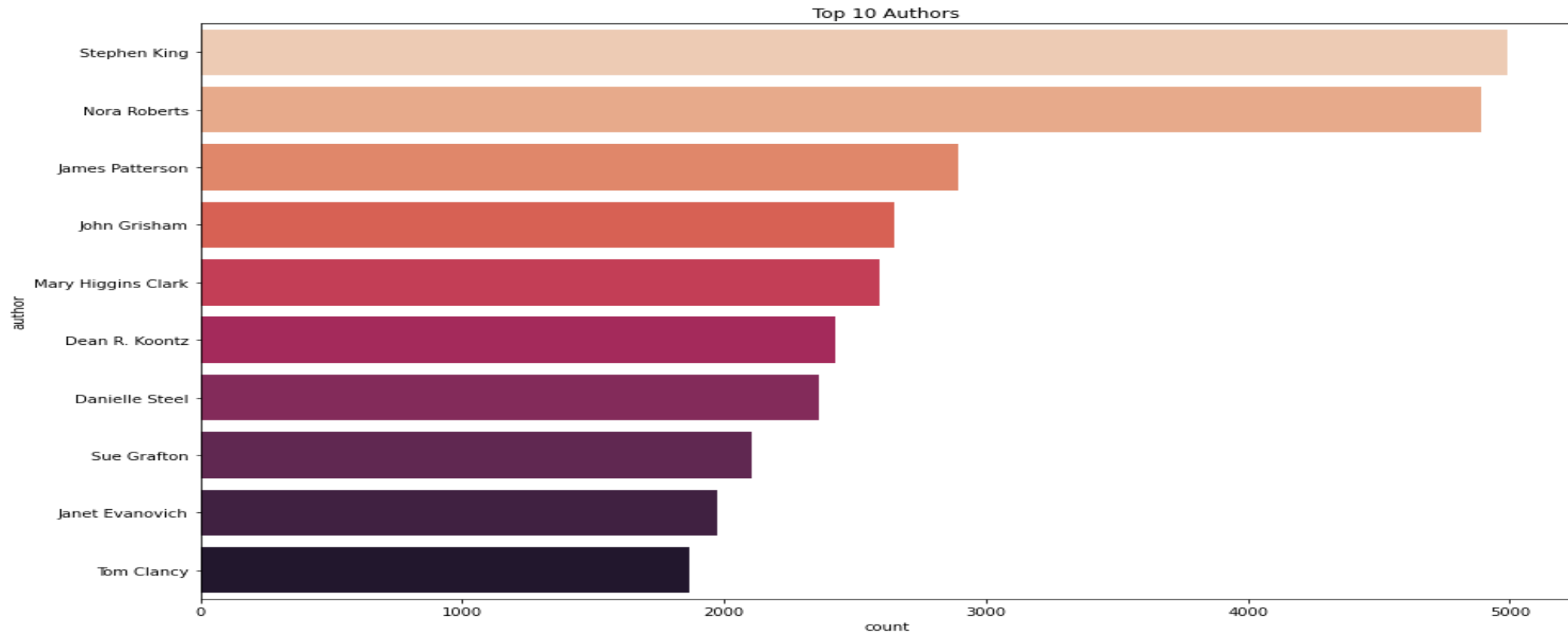
```
final_rating
```

	user_id	ISBN	rating	title	author	year	publisher	number of ratings
0	276847	446364193	0	Along Came a Spider (Alex Cross Novels)	James Patterson	1993	Warner Books	173
1	278418	446364193	0	Along Came a Spider (Alex Cross Novels)	James Patterson	1993	Warner Books	173
2	5483	446364193	0	Along Came a Spider (Alex Cross Novels)	James Patterson	1993	Warner Books	173
3	7346	446364193	0	Along Came a Spider (Alex Cross Novels)	James Patterson	1993	Warner Books	173
4	8362	446364193	0	Along Came a Spider (Alex Cross Novels)	James Patterson	1993	Warner Books	173
...
641491	250764	044023509X	0	Meltdown	James Powlik	2002	Dell Publishing Company	1
641492	250764	451157516	0	Cheyenne (Fortunes West, No 2)	A.R. Riefe	1988	New Amer Library (Mm)	1
641493	250764	048623715X	0	Glamorous Movie Stars of the Thirties: Paper D...	Tom Tierney	1982	Dover Publications	1
641494	250764	486256588	0	Schiaparelli Fashion Review: Paper Dolls in Fu...	Tom Tierney	1988	Dover Publications	1
641495	250764	515069434	0	Lady Laughing Eyes (To Have and to Hold)	Lee Damon	1984	Jove Books	1

641496 rows × 8 columns

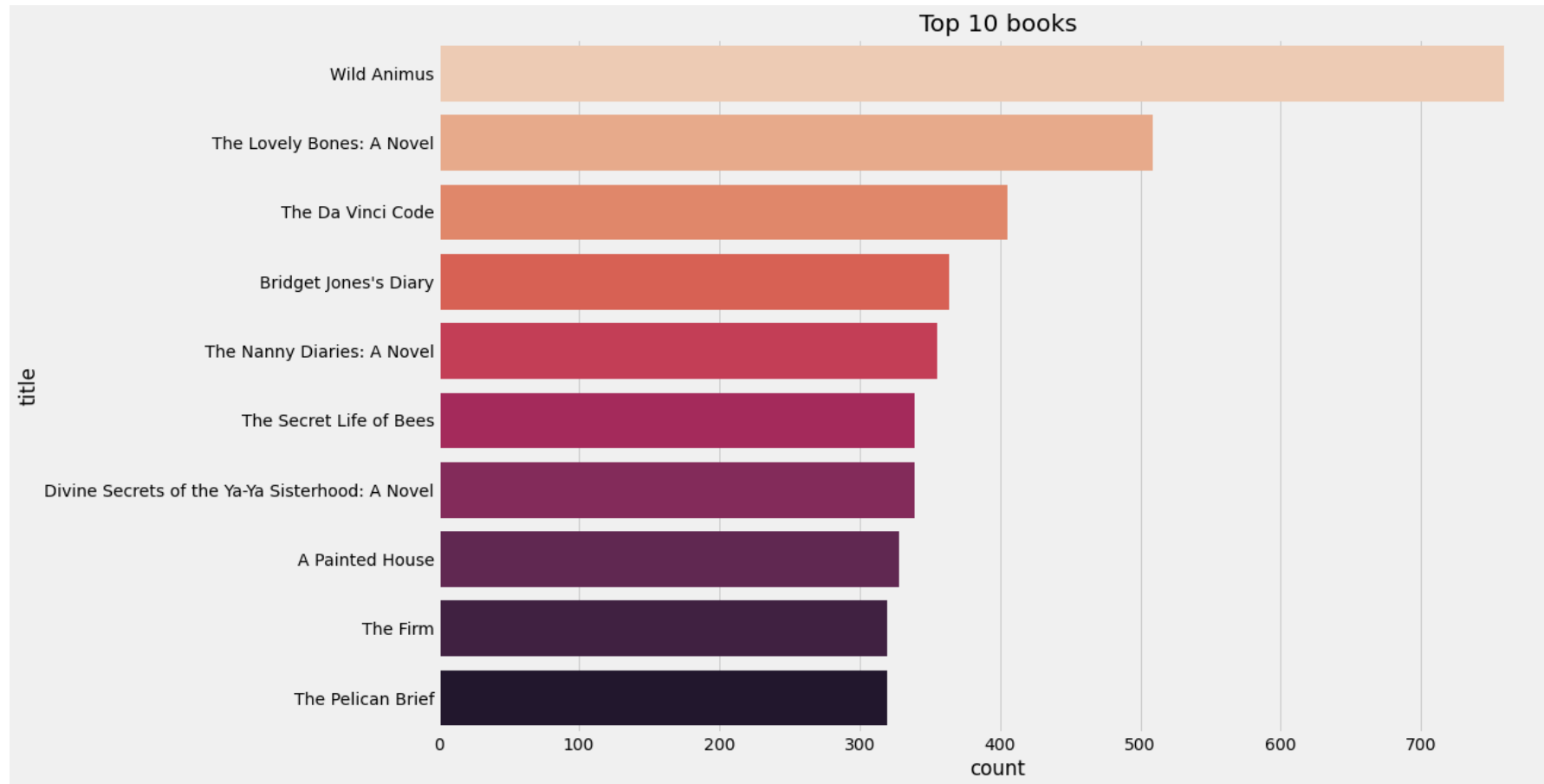
Exploratory Data Analysis (EDA)

- Observations from Top 10 Authors



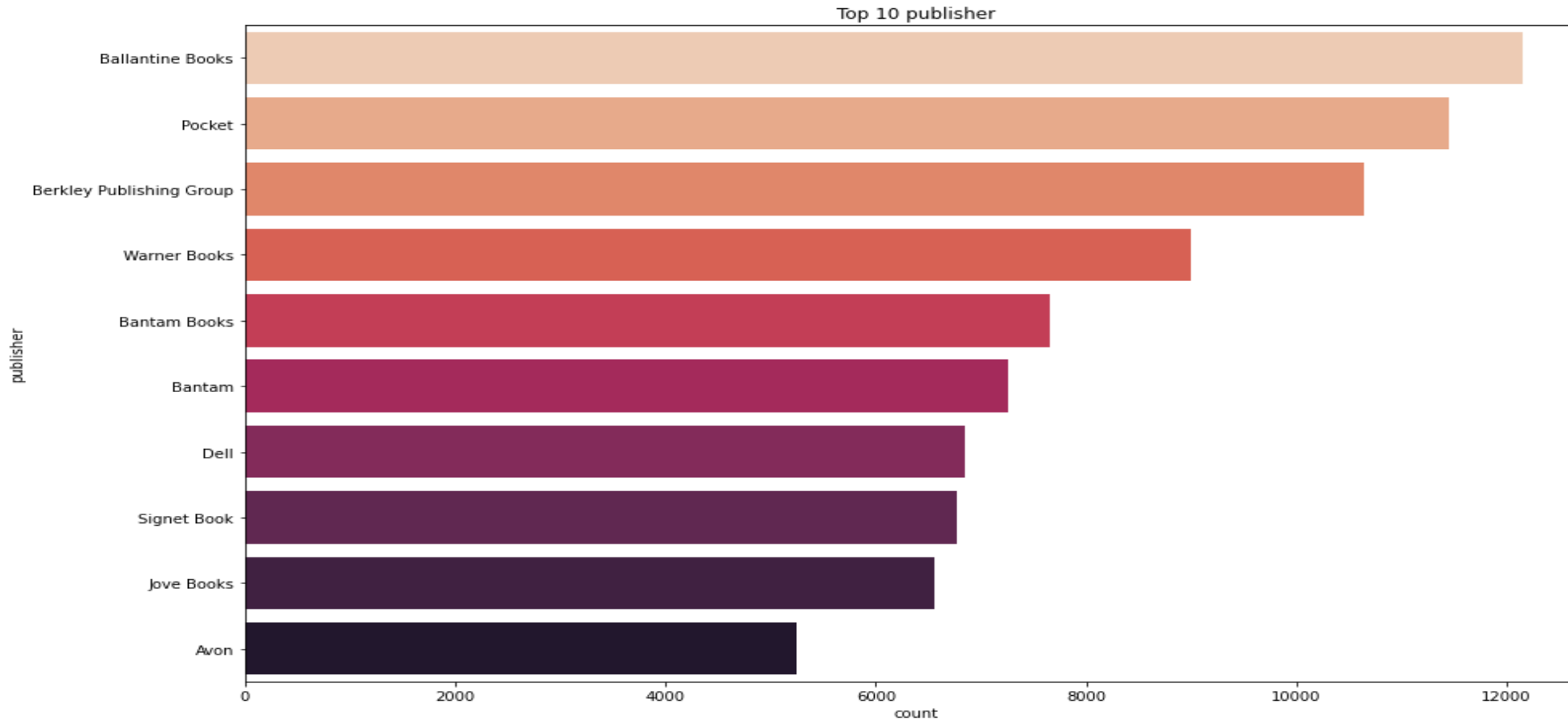
- Author Stephen King highest number of books in our given dataset

- Observations from Top 10 Books



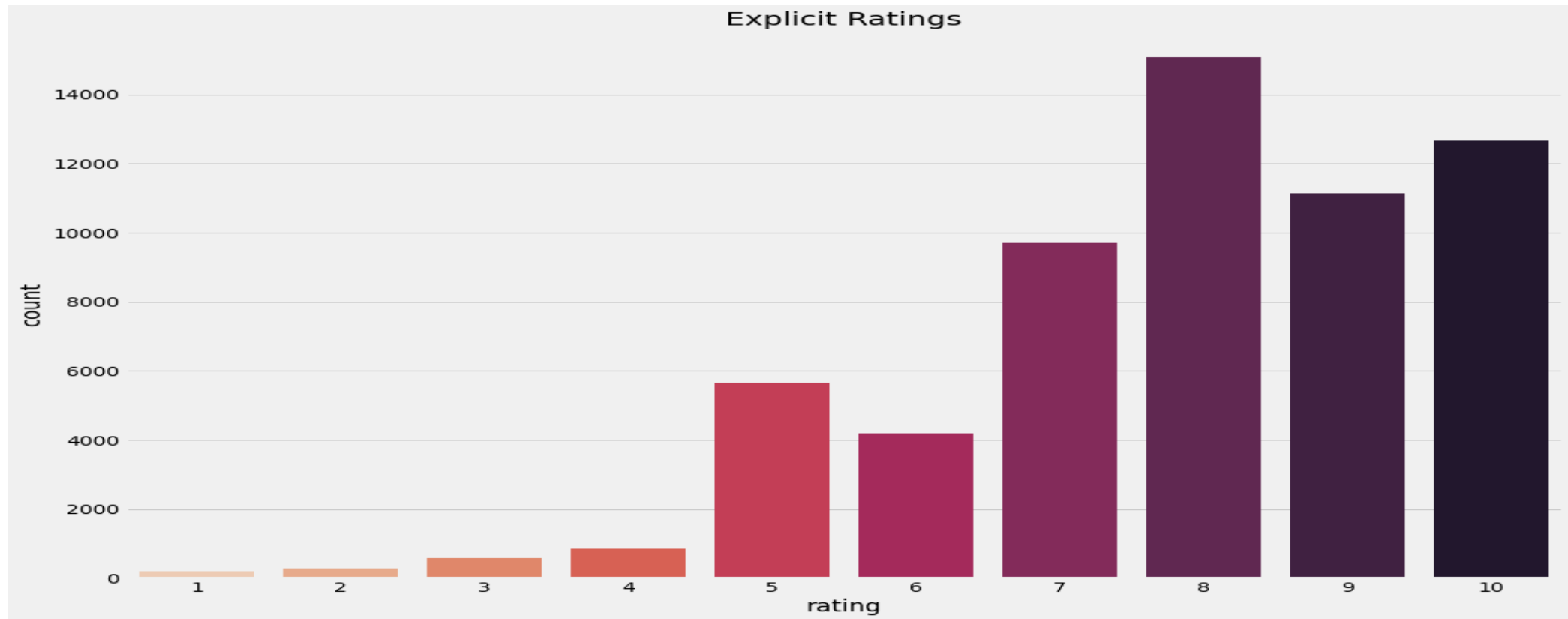
Wild Animal highest number of books in our given dataset

- Observations from Top 10 publisher



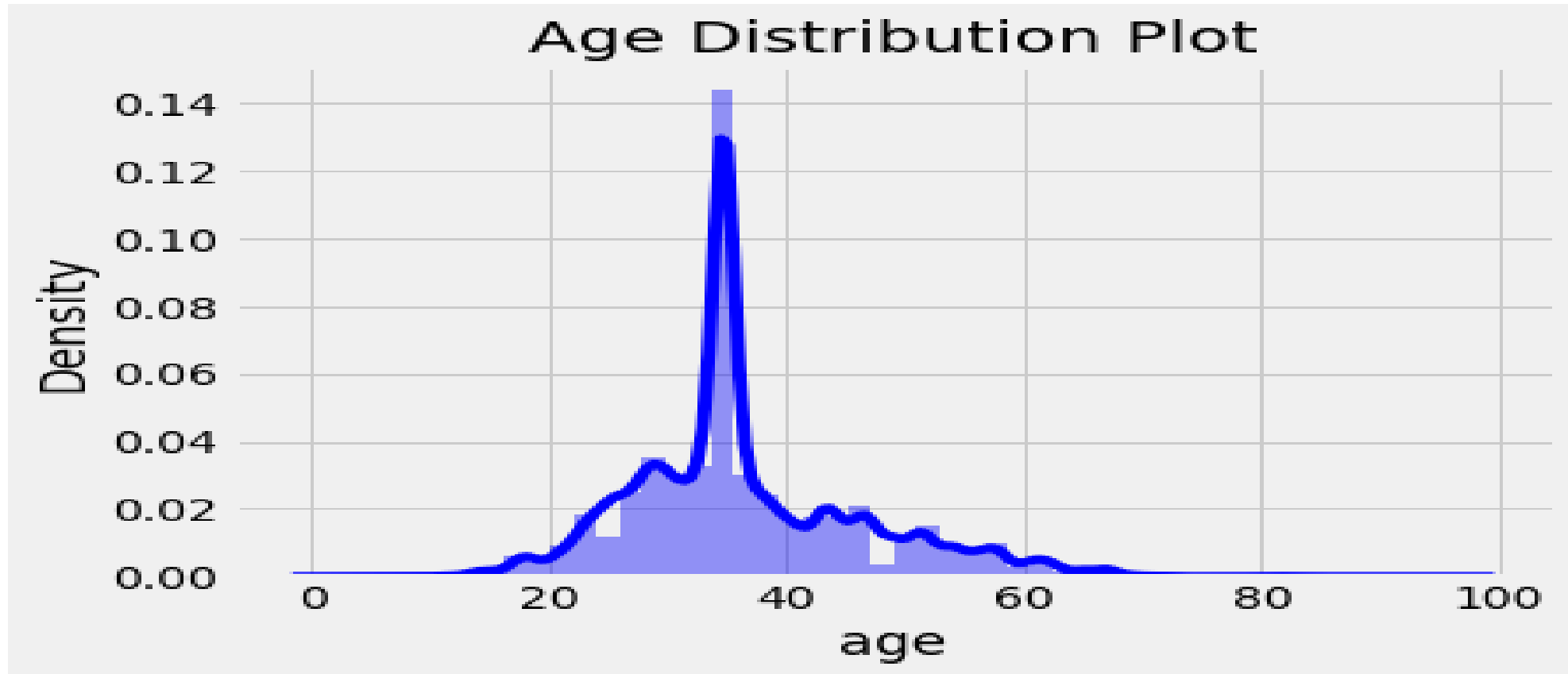
- Ballantine Books published highest number of books in our given dataset

- Observations from Book Ratings.



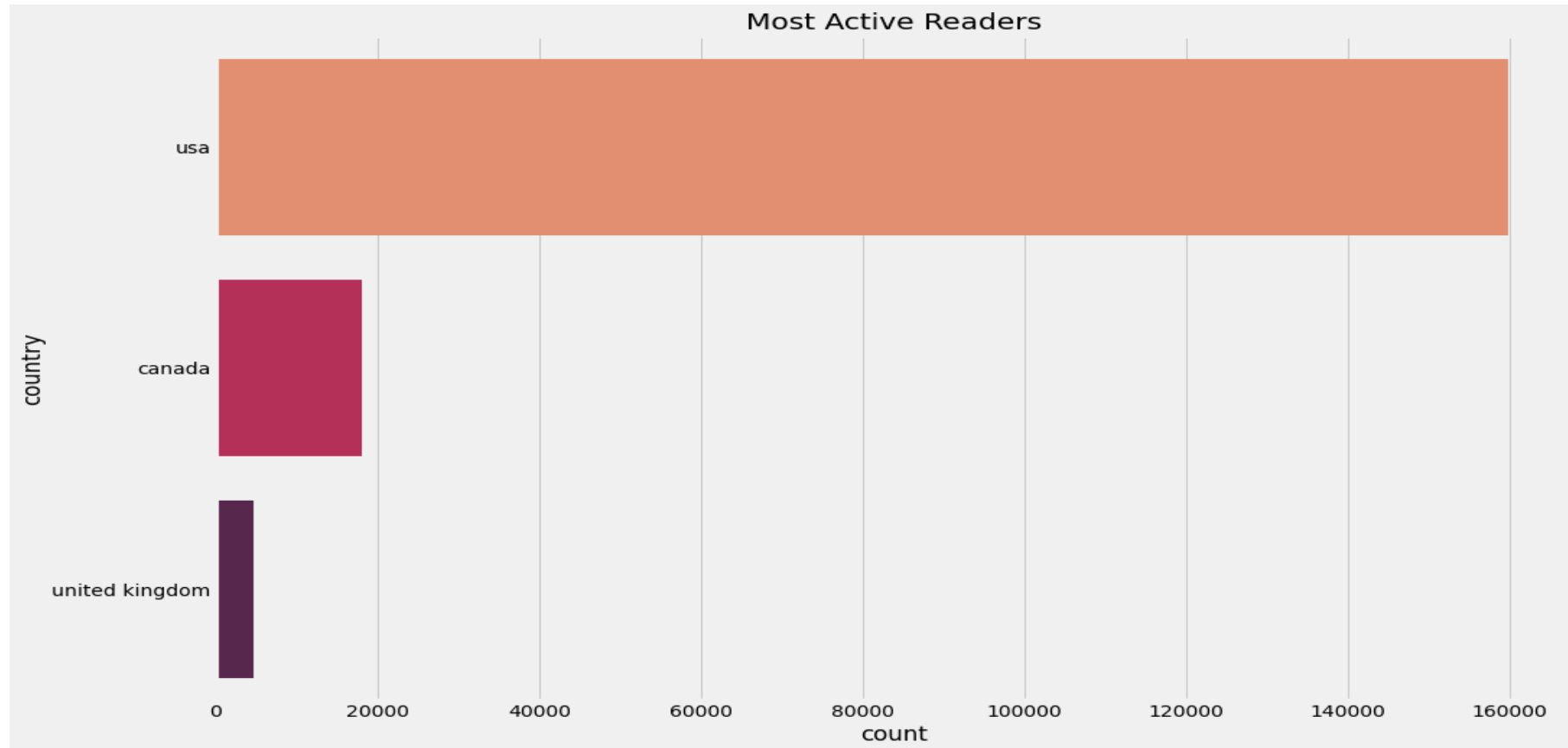
- Higher ratings are more common amongst users
- Rating 8 has been rated the highest number of times

- Age Distribution.



- The Age range distribution is right skewed
- Most active readers lie in age group 20- 60

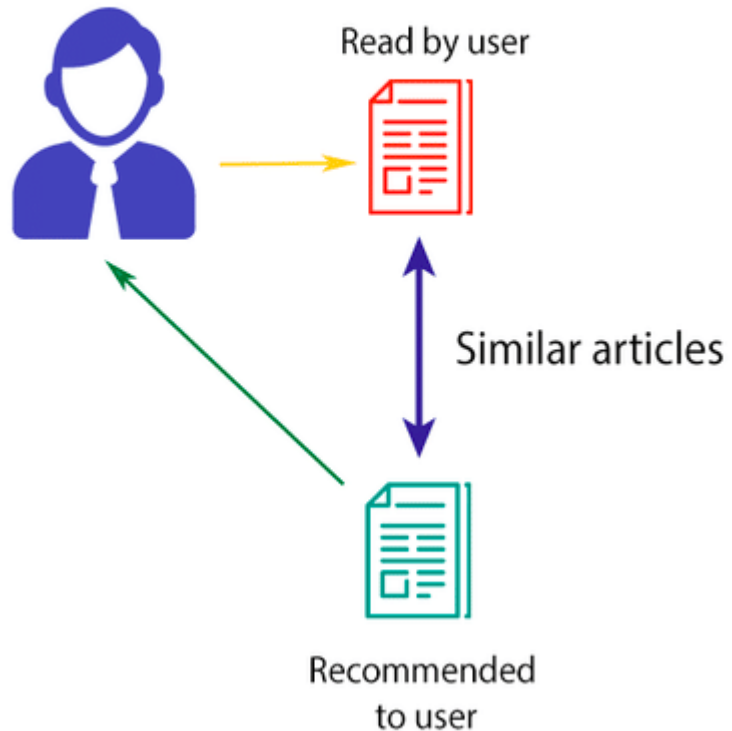
- Observations from Most Active Readers.



- Splitting Location column and analyzing country.
- Most active readers are from USA.

Recommendation Systems

CONTENT-BASED FILTERING



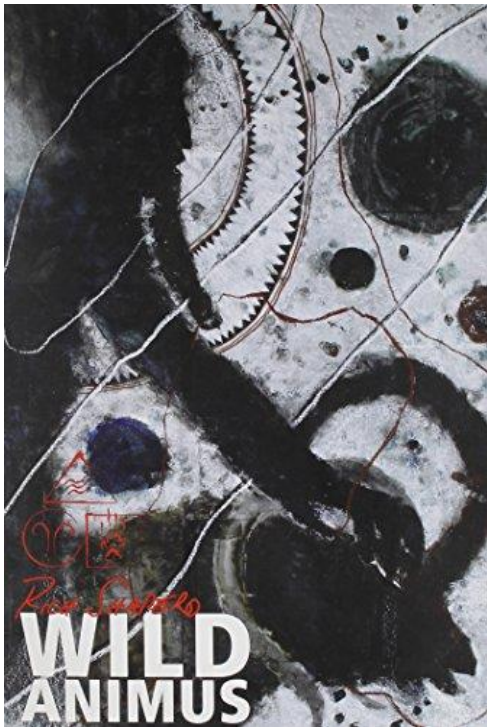
Content Based Filtering :

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

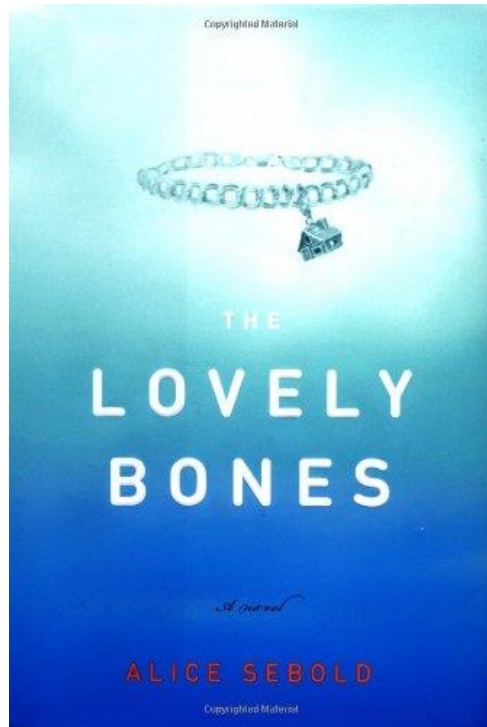
Different Models Test set:
Actual Popularity Based top rated books

AI

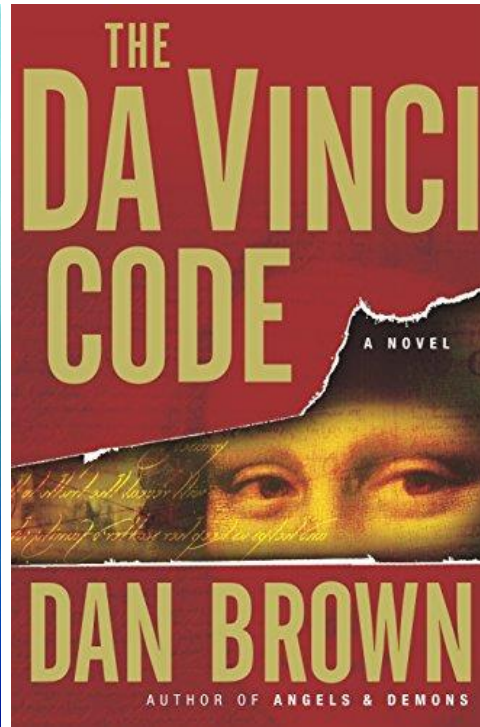
RECOMMENDATIONS RECOMMENDATIONS RECOMMENDATIONS RECOMMENDATIONS RECOMMENDATIONS



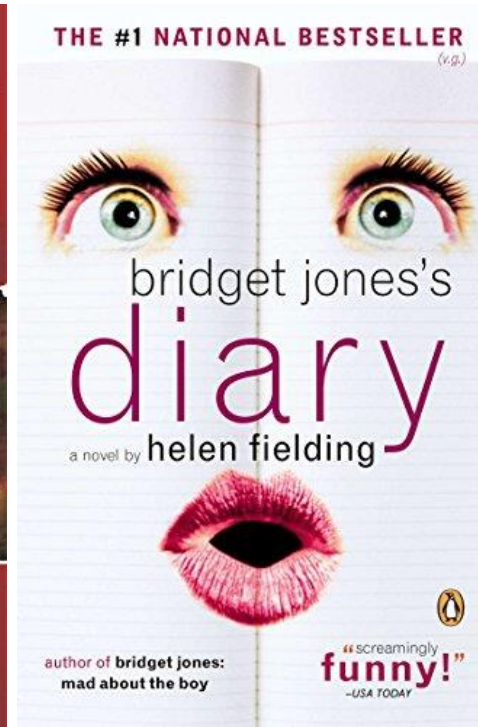
Wild Animus



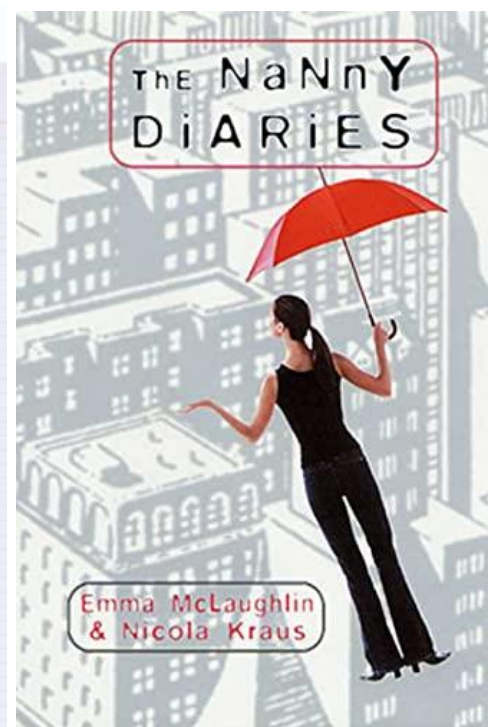
The Lovely
Bones: A Novel



The Da Vinci Code



Bridget Jones's Diary



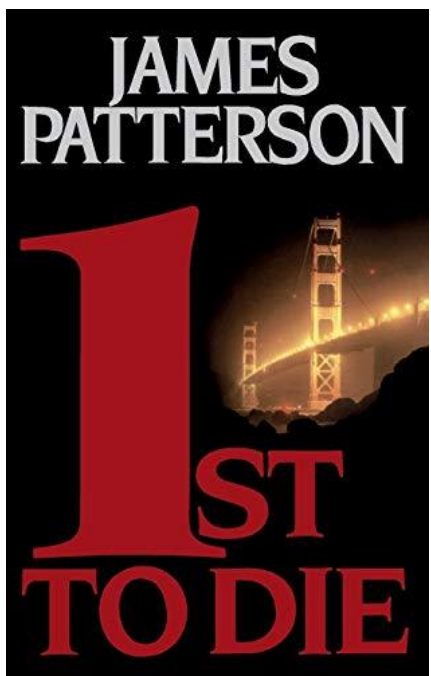
The Nanny
Diaries: A Novel

Different Models Test set:

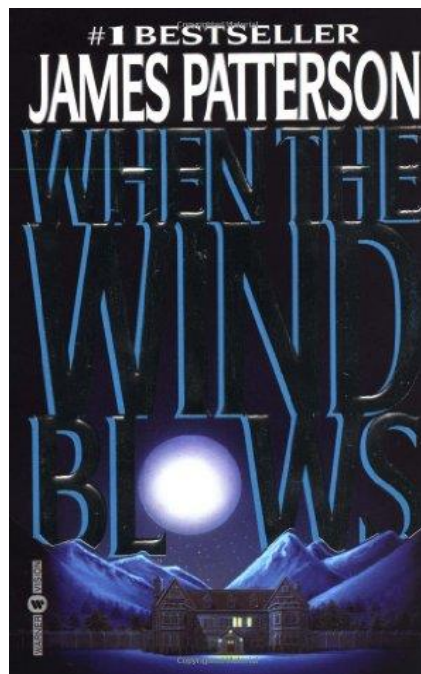
Author Based Recommendation Systems

Author Name: James Patterson

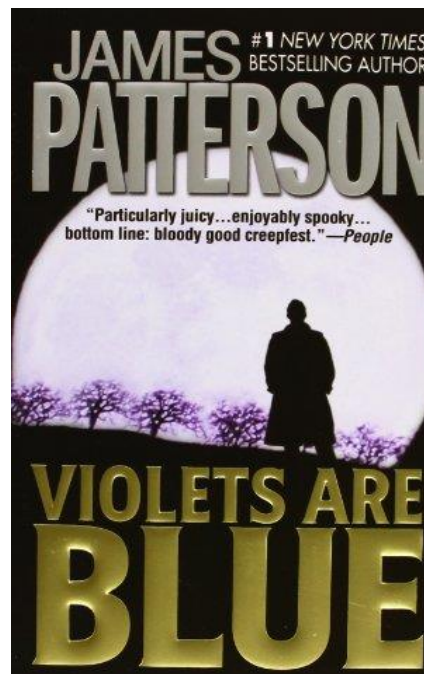
RECOMMENDATIONS RECOMMENDATIONS RECOMMENDATIONS RECOMMENDATIONS RECOMMENDATIONS



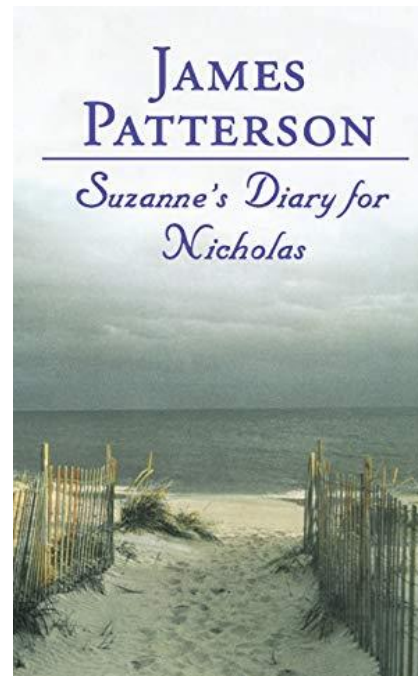
1st to Die: A Novel



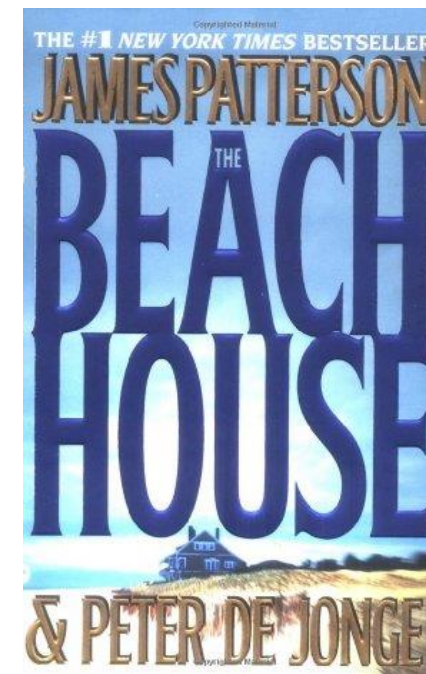
When the Wind
Blows



Violets Are Blue



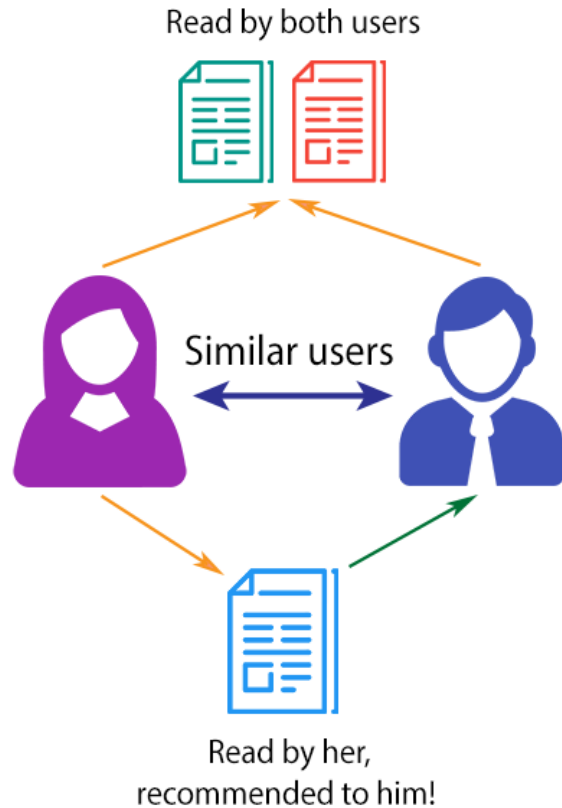
Suzanne's Diary for
Nicholas



The Beach House

Recommendation Systems

COLLABORATIVE FILTERING



Collaborative Filtering:

Collaborative filtering is a technique that can filter out items that a user might like on the basis of reactions by similar users

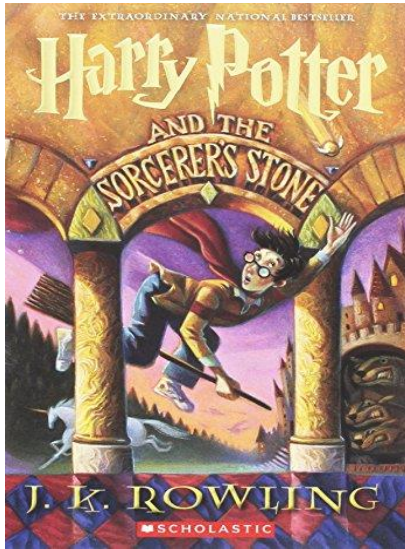
Model Implementation

- **Collaborative Filtering** (User-Item Filtering)
- **Collaborative Filtering** (Correlation Based)
- **Collaborative Filtering** (Nearest Neighbor's Based)

Collaborative Filtering (User-Item Filtering)



INPUT BOOK



```
k = list(final_rating['title'])  
m = list(final_rating['ISBN'])
```

```
collaborative = getTopRecommendations(m[k.index(bookName)])
```



Input Book:

Harry Potter and the Sorcerer's Stone (Harry Potter (Paperback))

RECOMMENDATIONS:

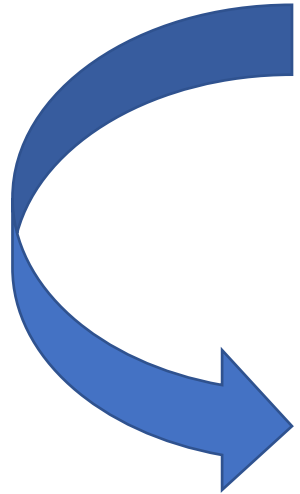
Harry Potter and the Prisoner of Azkaban (Book 3)

Harry Potter and the Goblet of Fire (Book 4)

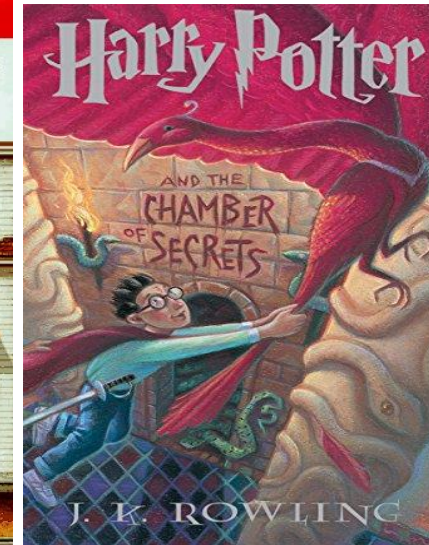
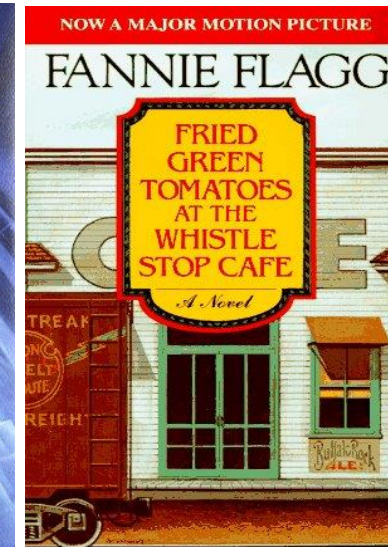
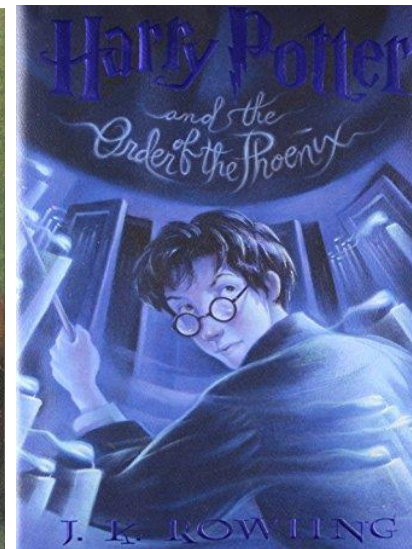
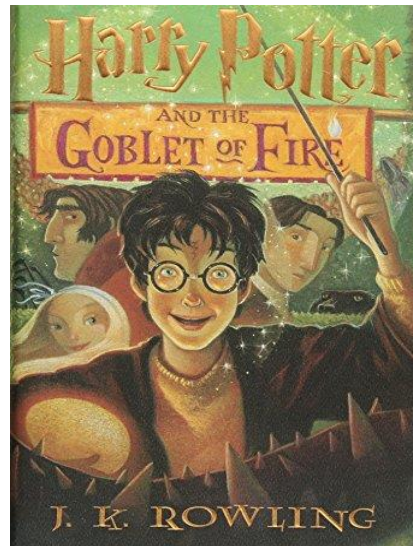
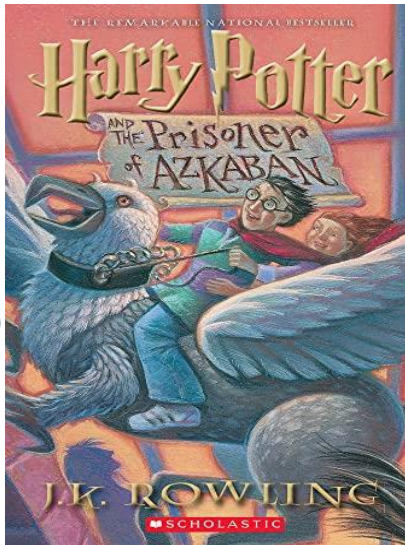
Harry Potter and the Order of the Phoenix (Book 5)

Fried Green Tomatoes at the Whistle Stop Cafe

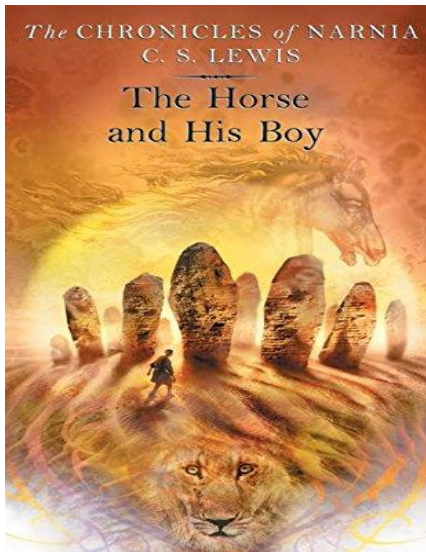
Harry Potter and the Chamber of Secrets (Book 2)



RECOMMENDATION



Collaborative Filtering (Correlation Based)



INPUT BOOK

```
#example= Harry Potter and the Sorcerer's Stone (Harry Potter (Paperback))

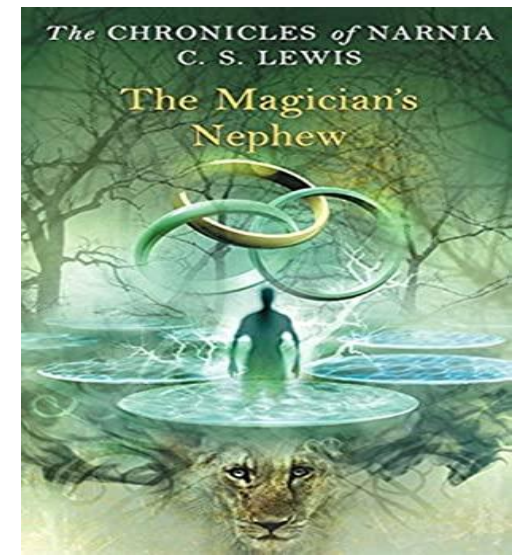
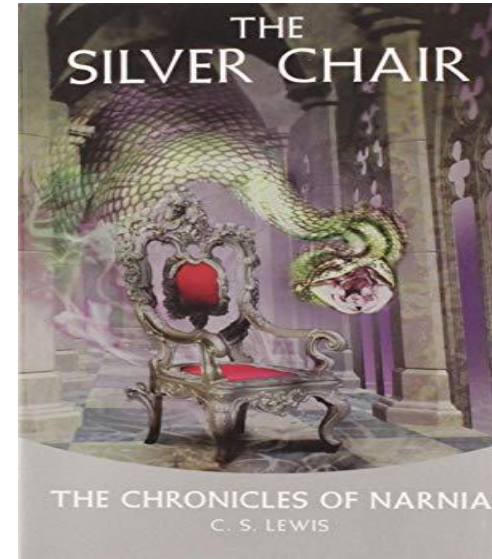
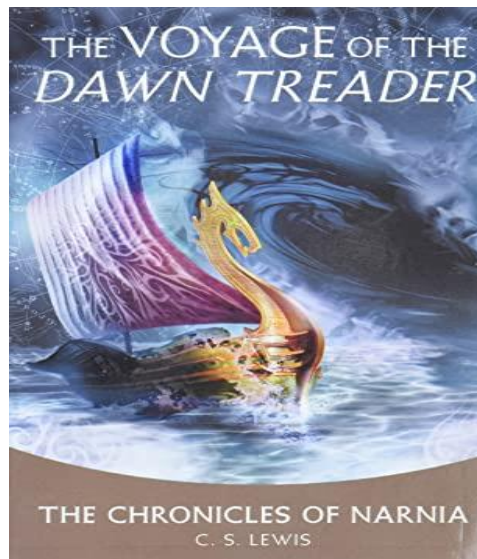
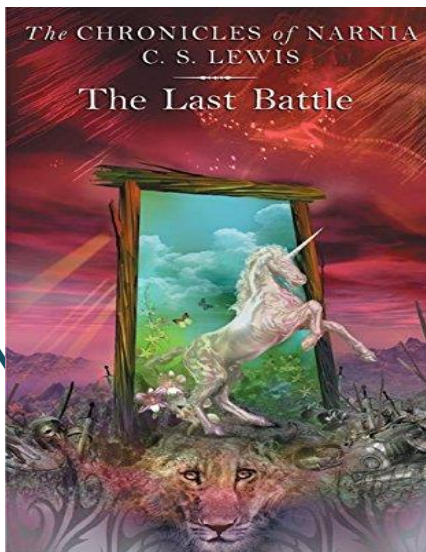
isbn = books.loc[books['title'] == bookName].reset_index(drop = True).iloc[0]['ISBN']
row = matrix[isbn]
correlation = pd.DataFrame(matrix.corrwith(row), columns = ['Pearson Corr'])
corr = correlation.join(average_rating['ratingCount'])

res = corr.sort_values('Pearson Corr', ascending=False).head(number+1)[1:].index
corr_books = pd.merge(pd.DataFrame(res, columns = ['ISBN']), books, on='ISBN')
print("\n Recommended Books: \n")
corr_books['title']
```

Enter a book name: The Horse and His Boy

Recommended Books:

```
0          The Last Battle
1    The Voyage of the Dawn Treader (rack) (Narnia)
2          The Silver Chair
3    Prince Caspian (rack) : The Return to Narnia (...)
4    The Magician's Nephew (rack) (Narnia)
Name: title, dtype: object
```

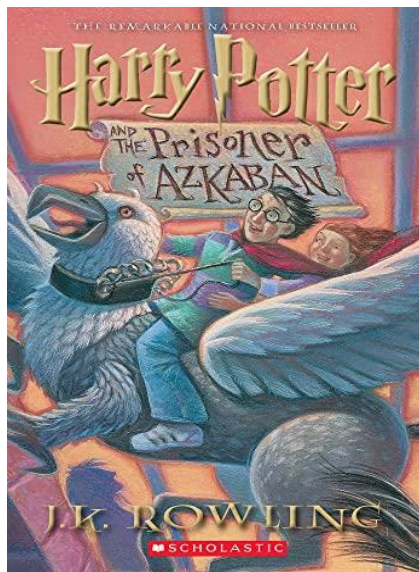


RECOMMENDATION

Collaborative Filtering (Nearest Neighbor's Based)



INPUT BOOK



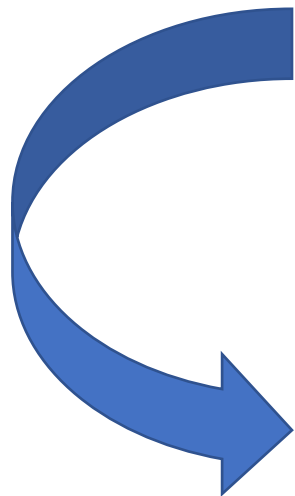
```
Books_Name=str(input("Pls Entre the Book Name \n"))

def recommend_book(book_name):
    book_id=np.where(book_pivot.index ==book_name)[0][0]
    distances,suggestions= model.kneighbors(book_pivot.iloc[book_id, :].values.reshape(1,-1))

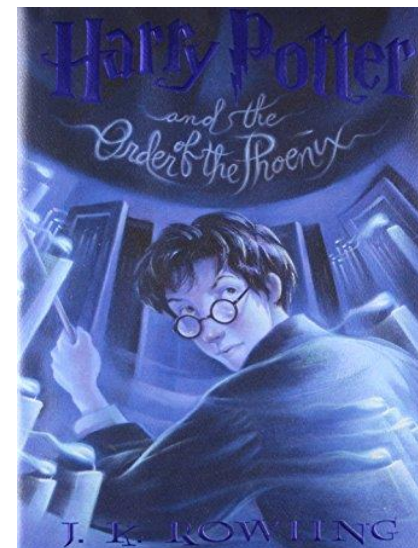
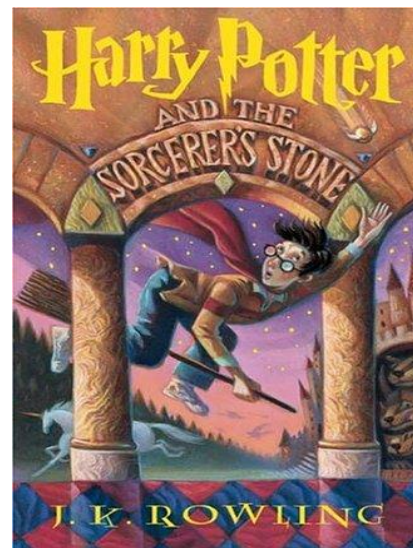
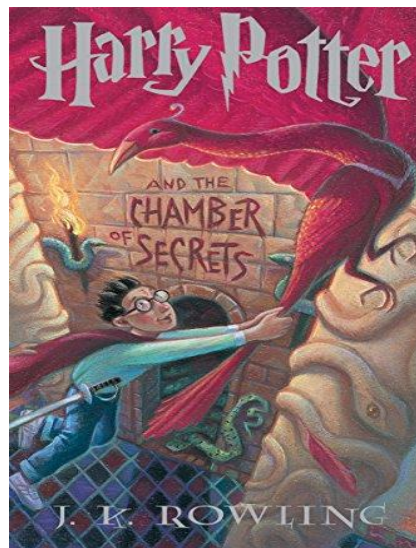
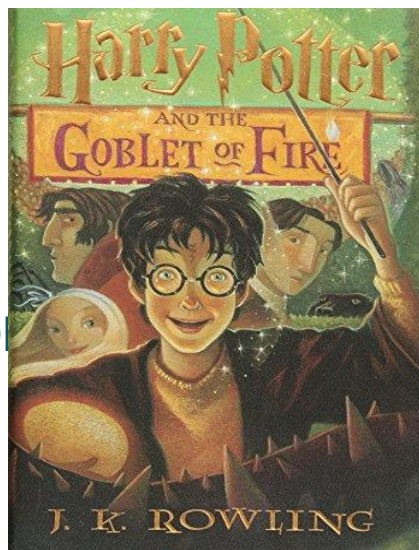
    for i in range(len(suggestions)):
        if i==0:
            print("The Suggestions for",book_name,'are :\n\n')
            if not i:
                print(book_pivot.index[suggestions[i]])
    recommend_book(Books_Name)
#Example-Books Name - "Harry Potter and the Prisoner of Azkaban (Book 3)"
```

```
Pls Entre the Book Name
Harry Potter and the Prisoner of Azkaban (Book 3)
The Suggestions for Harry Potter and the Prisoner of Azkaban (Book 3) are :
```

```
Index(['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 Sorcerer's Stone (Book 1)',
      'Harry Potter and the Order of the Phoenix (Book 5)'],
      dtype='object', name='title')
```



RECOMMENDATION



Conclusion

In EDA, the Top-10 most rated books were essentially novels. Books like The Wild Animus and The Lovely Bones: A Novel .

Majority of the readers were of the age bracket 20-50 and most of them came from North American and European countries namely USA, Canada, UK.

If we look at the ratings distribution, most of the books have high ratings with maximum books being rated 8. Ratings below 5 are few in number.

Author with the most books was Stephen King, Nora Roberts and James Patterson.

A recommendation system helps an organization to create loyal customers. The recommendation system today are very powerful that they can handle the new customer too who has visited the site for the first time. They recommend the products which are currently trending or highly rated and they can also recommend the products which bring maximum profit to the company.

Challenges

- Handling of sparsity was a major challenge as well since the user interactions were not present for the majority of the books.
- Understanding the metric for evaluation was a challenge as well.
- Since the data consisted of text data, data cleaning was a major challenge in features like Location etc..
- Decision making on missing value imputations and outlier treatment was quite challenging as well.

Future Scope

- Given more information regarding the books dataset, namely features like Genre, Description etc., we could implement a content-filtering based recommendation system and compare the results with the existing collaborative-filtering based system.
- We would like to explore various clustering approaches for clustering the users based on Age, Location etc., and then implement voting algorithms to recommend items to the user depending on the cluster into which it belongs.

Thank You