



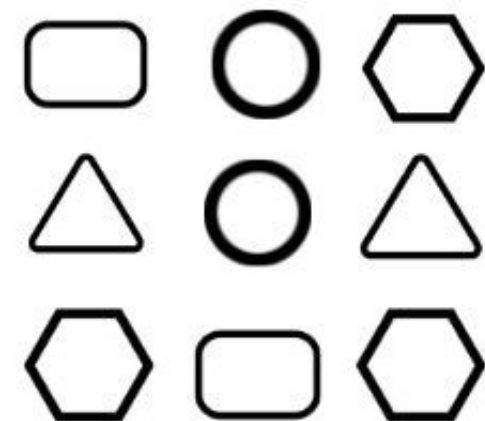
Unsupervised ML Classification Capstone Project

Book Recommendation System

Unsupervised learning

Unsupervised machine learning, uses machine learning algorithms to analyze and cluster unlabeled datasets. These algorithms discover hidden patterns or data groupings without the need for human intervention. Its ability to discover similarities and differences in information make it the ideal solution for exploratory data analysis, cross-selling strategies, customer segmentation, and image recognition.

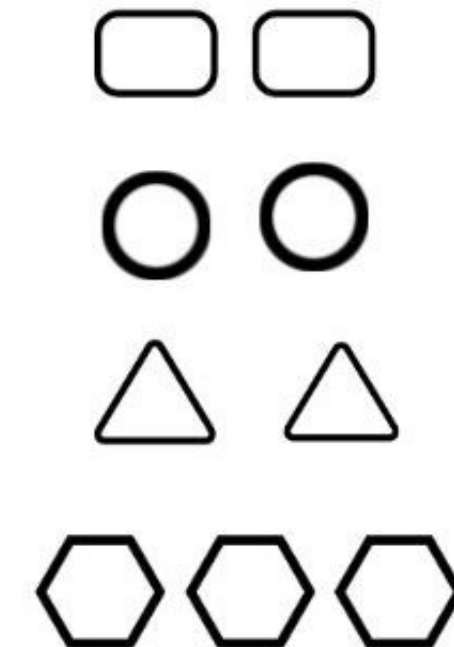
Unlabelled Data



Machine



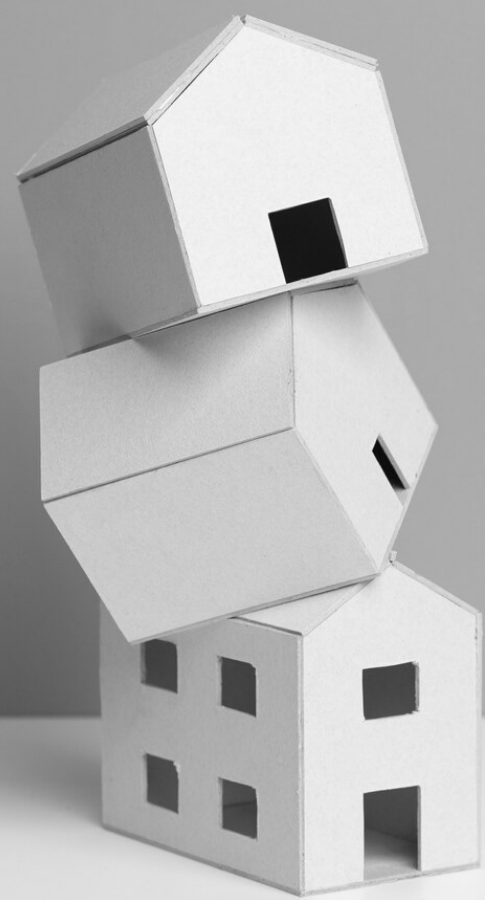
Results



Team Members:

- Kunal Gawande
- Chinma Rojatkar
- Deepali Mahajan
- Bipasha Zade
- Nikhil Aggarwal

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

**Problem
statement**

01

Data Summary

Data Cleaning

02

**Imputing missing
values**

**Analysis of
different datasets**

03

**Different
Recommendation Model**

Challenges

04

Conclusion

**Future
Scope**

05

Content

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 Books_dataset

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

✓ 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

2 Users_dataset

As we can see in the output,

`Users.isnull().sum()`

Function has return Age columns
110762 NaN data are present

✓ 0s `users.isnull().sum()`

User-ID	0
Location	0
Age	110762
dtype:	int64

3 Ratings_dataset

As we can see in the output,

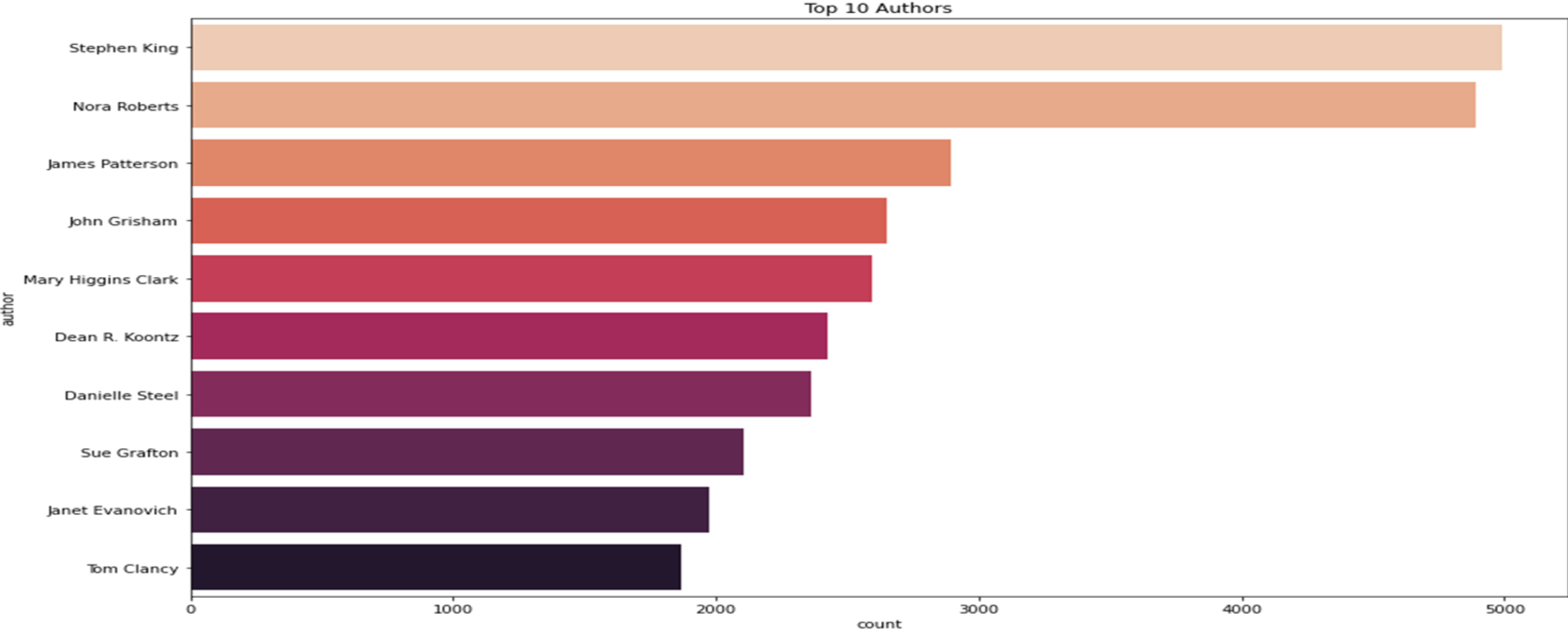
`ratings.isnull().sum()`

Function has Not NaN data are
present

✓ 0s [94] `ratings.isnull().sum()`

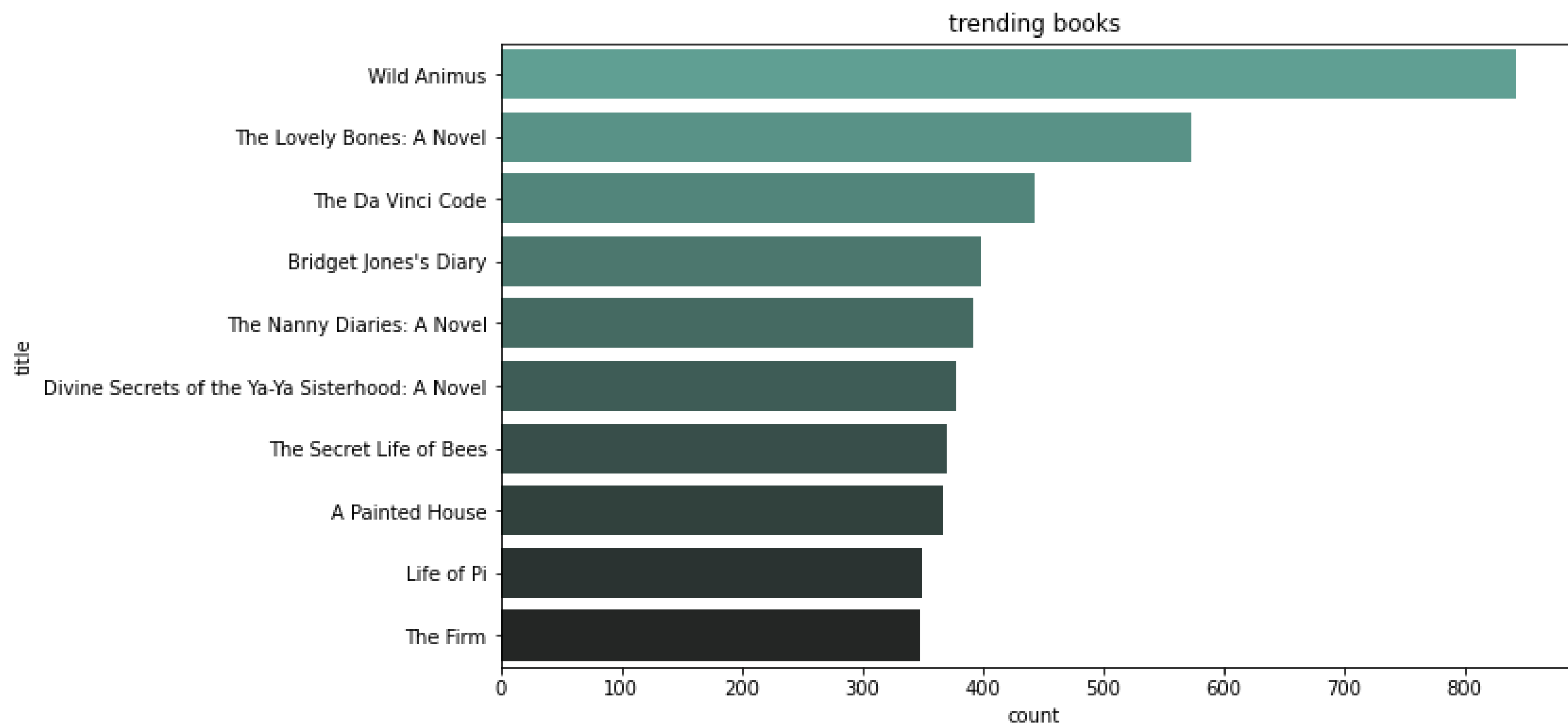
user_id	0
ISBN	0
rating	0
dtype:	int64

Exploratory Data Analysis (EDA)



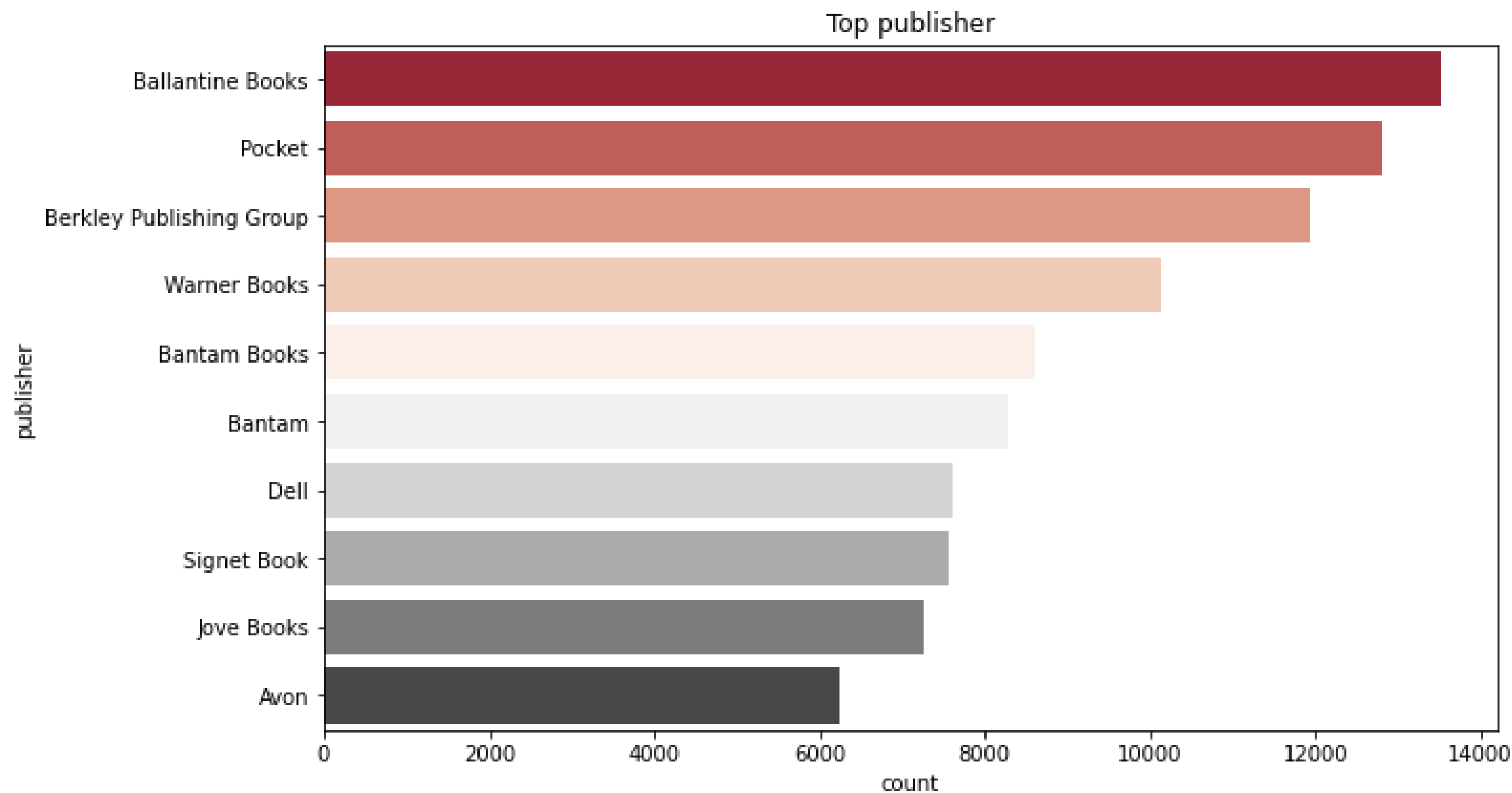
Author Stephen King highest number of books in our given dataset .

Observations from Top 10 Books



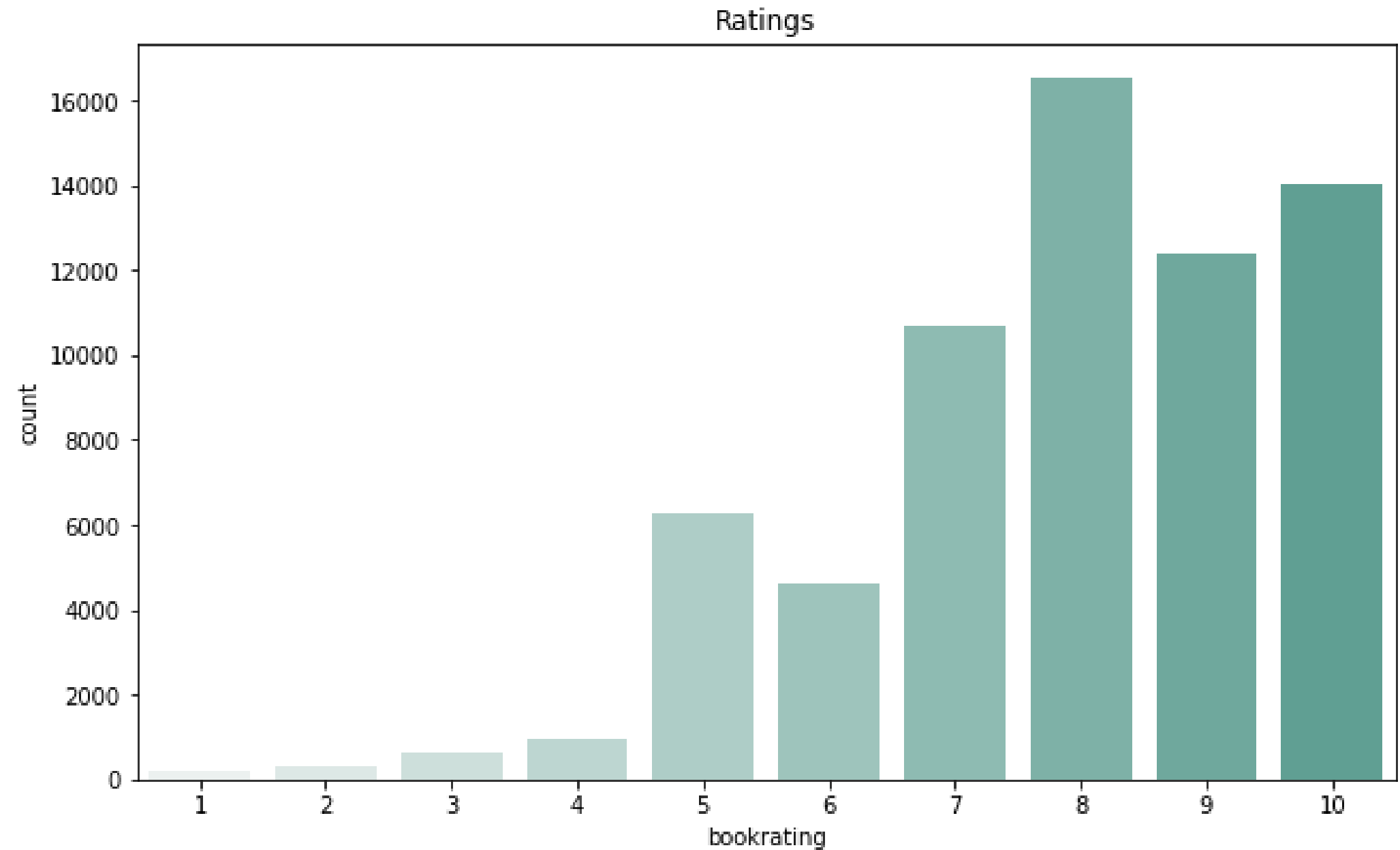
Wild Animal highest number of books in our given dataset .

Observations from Top 10 publisher



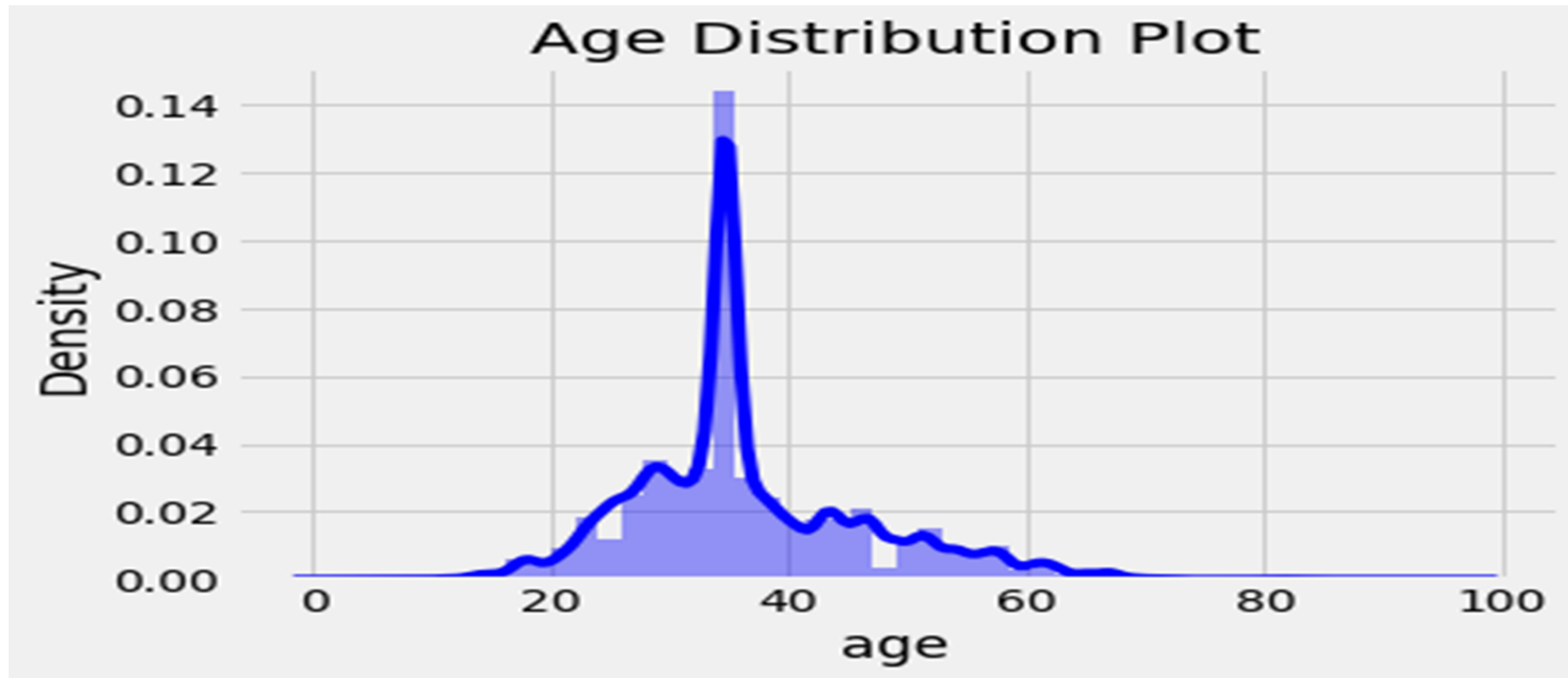
Ballantine Books published highest number of books in our given dataset

Observations from Book Ratings.



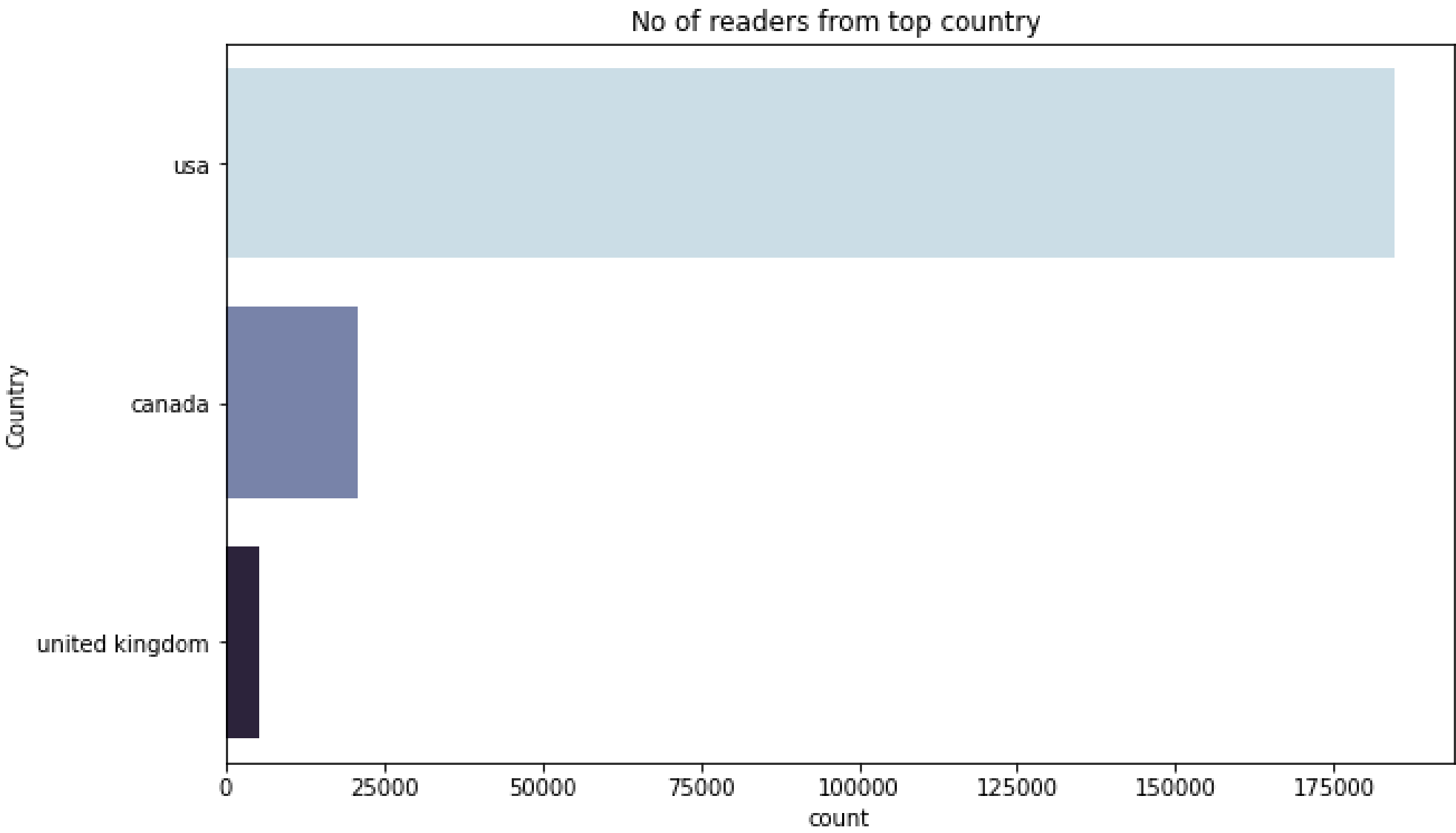
Higher ratings are more common amongst users
Rating 8 has been rated the highest number of times

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.

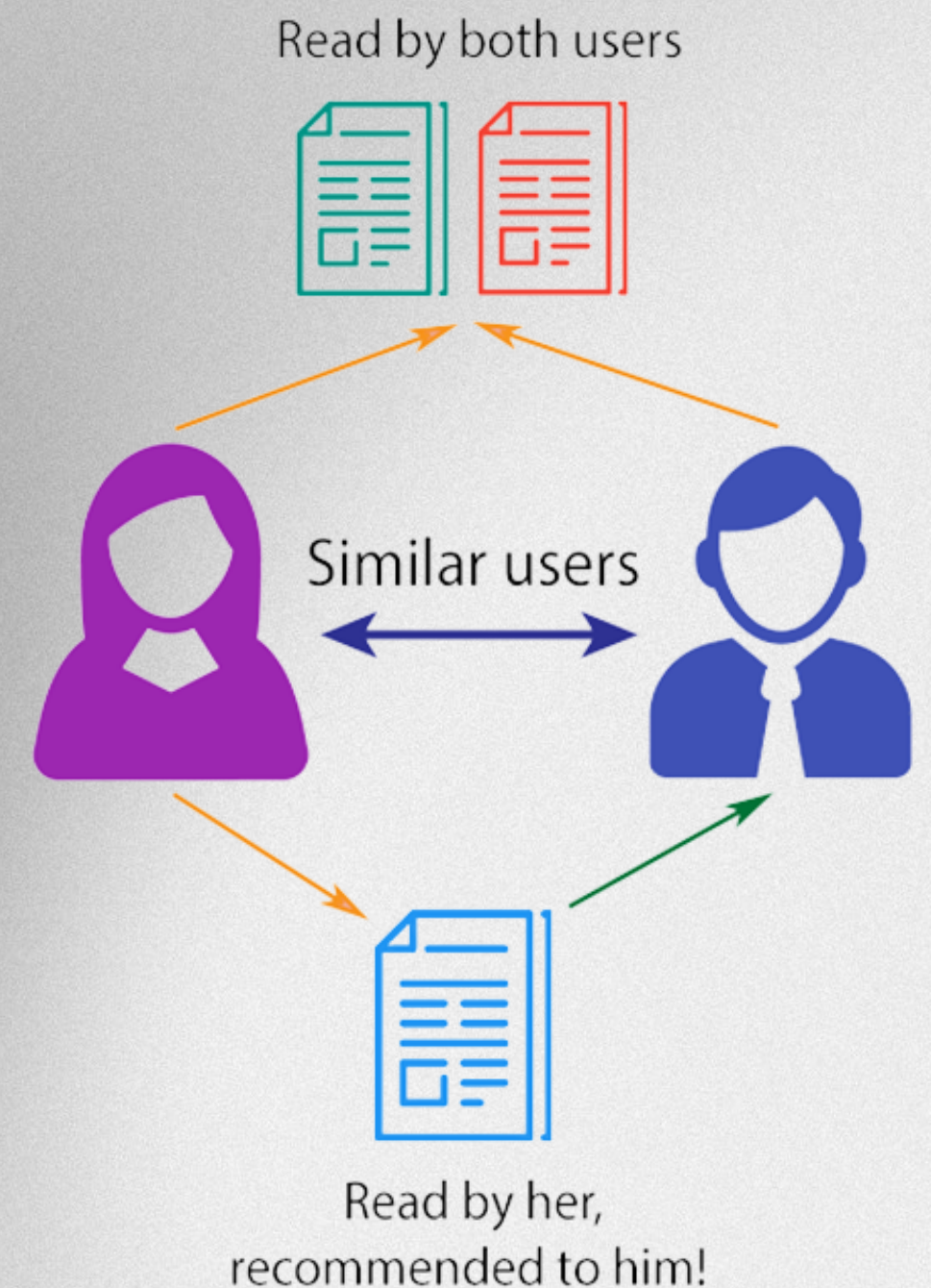
Recommendation Systems

Collaborative Filtering

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

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

COLLABORATIVE FILTERING



Collaborative Filtering (User-Item Filtering)

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

Collaborative Filtering (Correlation Based)

```
#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
```


Collaborative Filtering (Nearest Neighbor's Based)

```
▶ 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')
```

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.

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.

Thank you!

