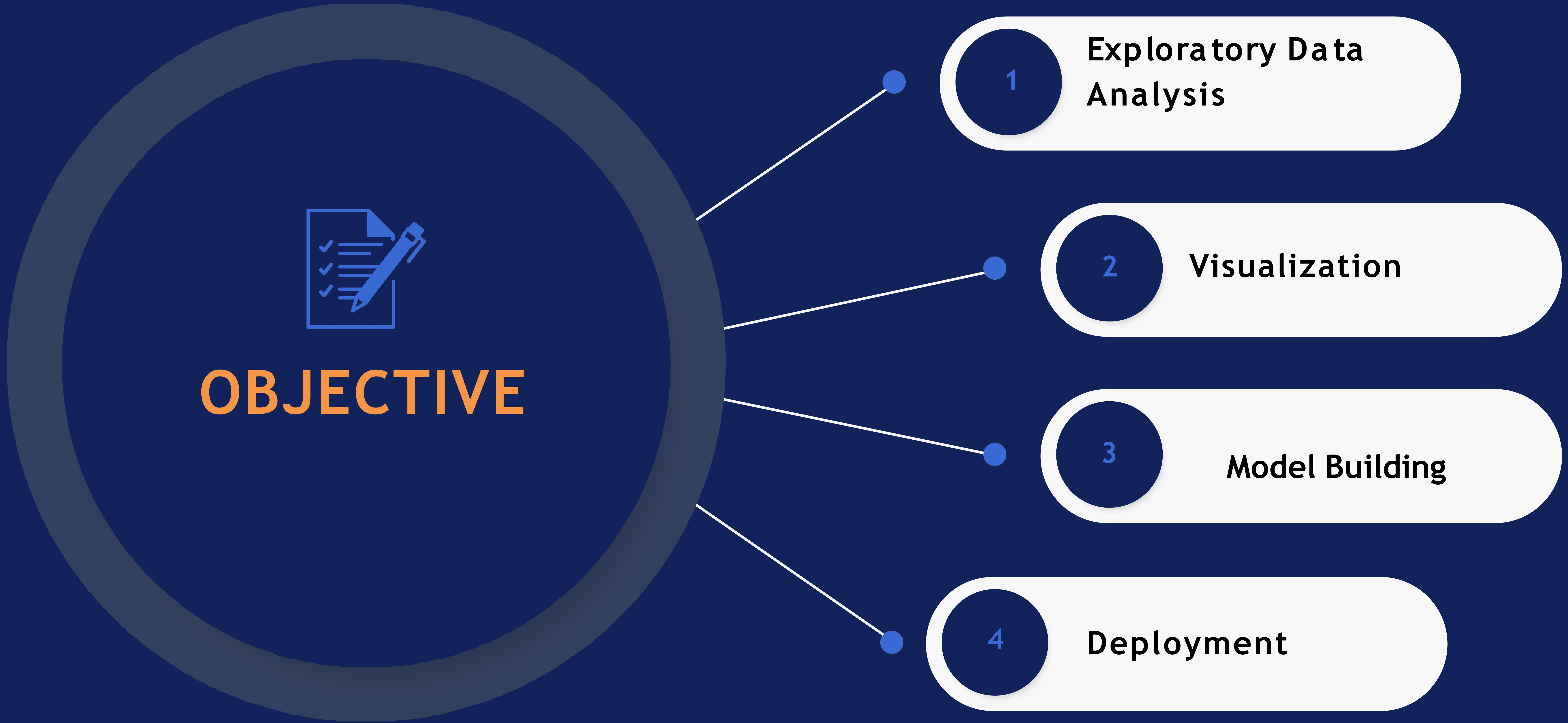
BOOK RECOMMENDATION SYSTEM

GROUP MEMBERS

- Kavya para
- Bhoomika B
- Drishya K
- Mr Kaif Shanawarali Quadri
- Amey Sandesh Shigwan



OBJECTIVE

1 Exploratory Data Analysis

2 Visualization

3 Model Building

4 Deployment

Objective

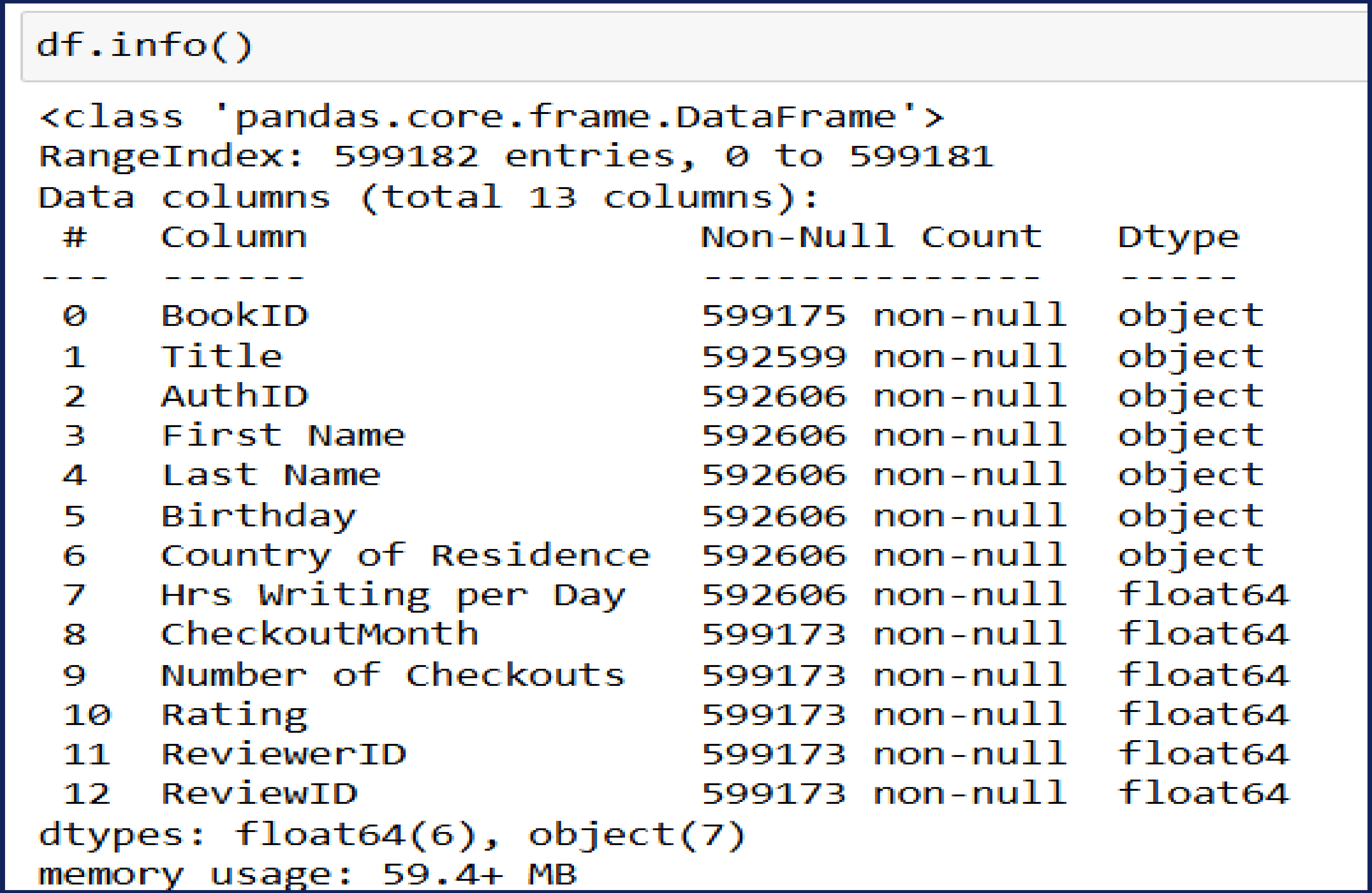
A Book Recommendation System aims to suggest books based on their preferences, reading history, and other relevant factors. The main objective of this dataset is to facilitate the development of a book recommendation system by analyzing and leveraging book metadata, ratings, and sales performance to suggest relevant and popular titles.

Libraries

Which we have used in the whole project

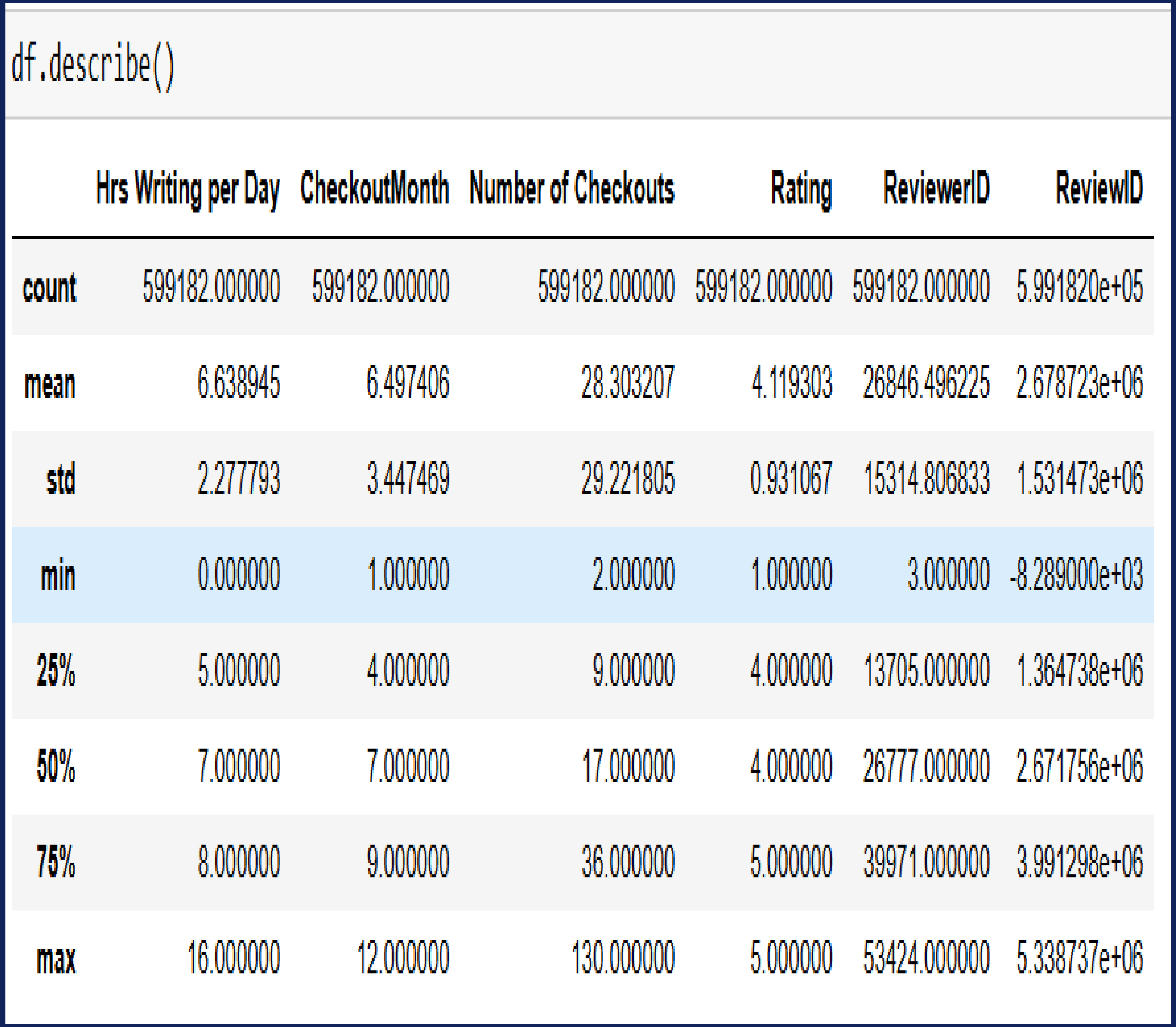
```
import pandas as pd
import  numpy as np
import matplot.pyplot as plt
import seaborn as sns
import warnings
warnings.fitterwarnings(‘ignore’)
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
from mlxtend.preprocessing import TransactionEncoder
from sklearn.cluster import Kmeans
from sklearn.metrics import silhouette_score
from scipy.sparse import csr_matrix
from sklearn.neighbors import NearestNeighbors
import pickle
```

EXPLORATORY DATA ANALYSIS



```
df.info()
```

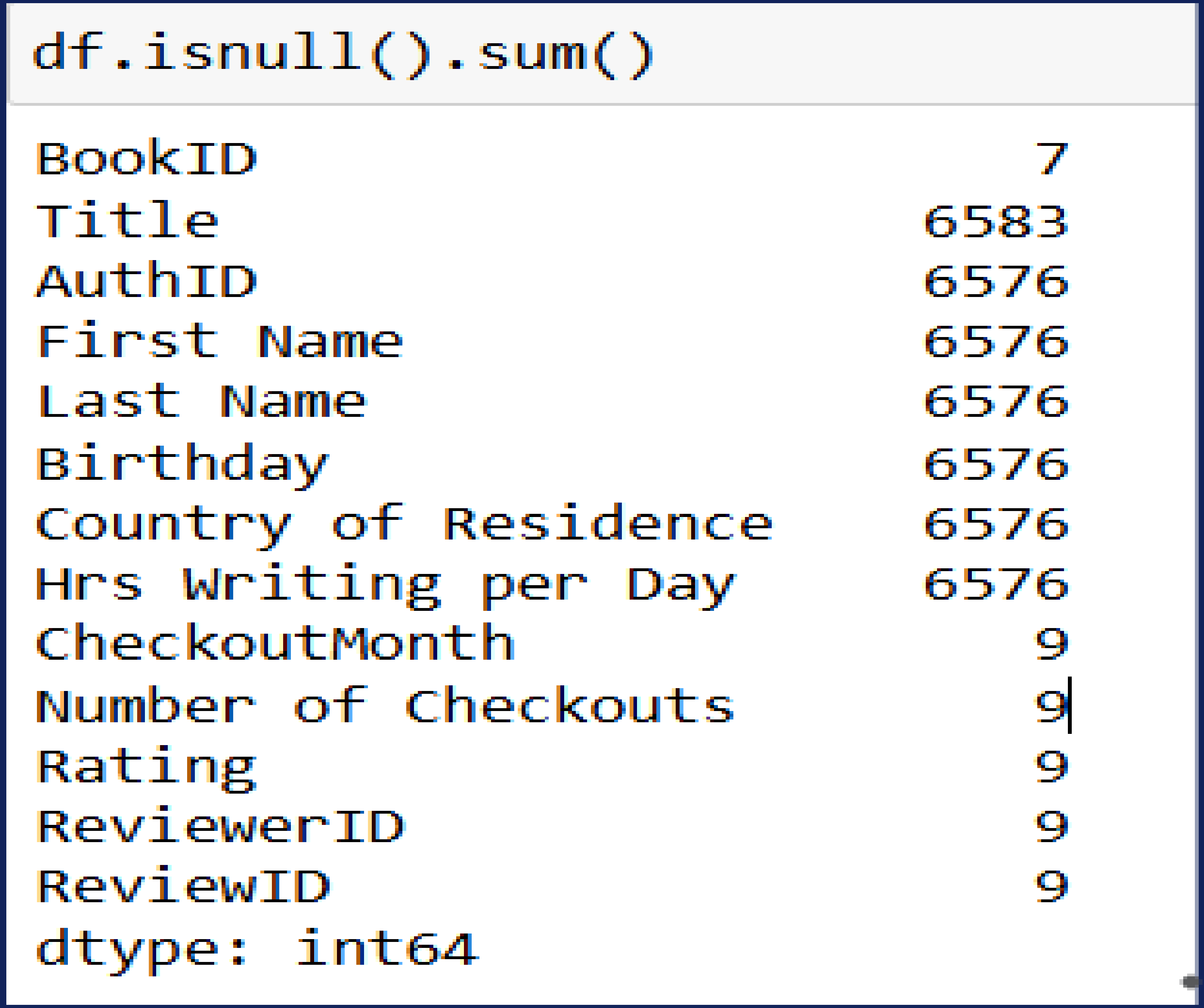
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 599182 entries, 0 to 599181
Data columns (total 13 columns):
 #   Column                  Non-Null Count   Dtype  
---  ------                  --------------   -----  
 0   BookID                  599175 non-null  object 
 1   Title                   592599 non-null  object 
 2   AuthID                  592606 non-null  object 
 3   First Name              592606 non-null  object 
 4   Last Name               592606 non-null  object 
 5   Birthday                592606 non-null  object 
 6   Country of Residence    592606 non-null  object 
 7   Hrs Writing per Day     592606 non-null  float64
 8   CheckoutMonth           599173 non-null  float64
 9   Number of Checkouts     599173 non-null  float64
 10  Rating                  599173 non-null  float64
 11  ReviewerID              599173 non-null  float64
 12  ReviewID                599173 non-null  float64
dtypes: float64(6), object(7)
memory usage: 59.4+ MB
```



```
df.describe()
```

	Hrs Writing per Day	CheckoutMonth	Number of Checkouts	Rating	ReviewerID	ReviewID
count	599182.000000	599182.000000	599182.000000	599182.000000	599182.000000	5.991820e+05
mean	6.638945	6.497406	28.303207	4.119303	26846.496225	2.678723e+06
std	2.277793	3.447469	29.221805	0.931067	15314.806833	1.531473e+06
min	0.000000	1.000000	2.000000	1.000000	3.000000	-8.289000e+03
25%	5.000000	4.000000	9.000000	4.000000	13705.000000	1.364738e+06
50%	7.000000	7.000000	17.000000	4.000000	26777.000000	2.671756e+06
75%	8.000000	9.000000	36.000000	5.000000	39971.000000	3.991298e+06
max	16.000000	12.000000	130.000000	5.000000	53424.000000	5.338737e+06

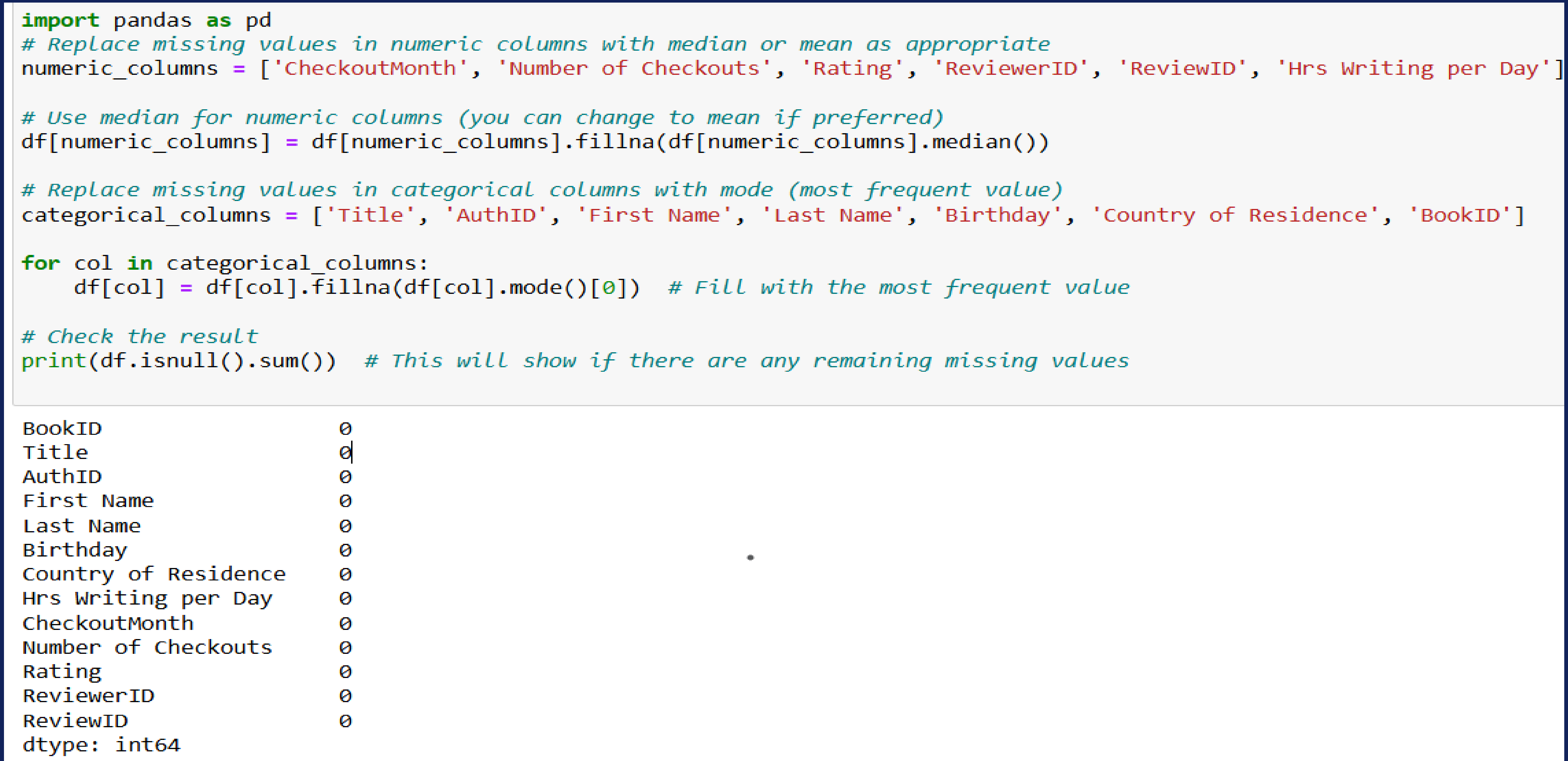
Missing values



```
df.isnull().sum()
```

```
BookID                     7
Title                   6583
AuthID                  6576
First Name              6576
Last Name               6576
Birthday                6576
Country of Residence    6576
Hrs Writing per Day     6576
CheckoutMonth              9
Number of Checkouts        9
Rating                     9
ReviewerID                 9
ReviewID                   9
dtype: int64
```

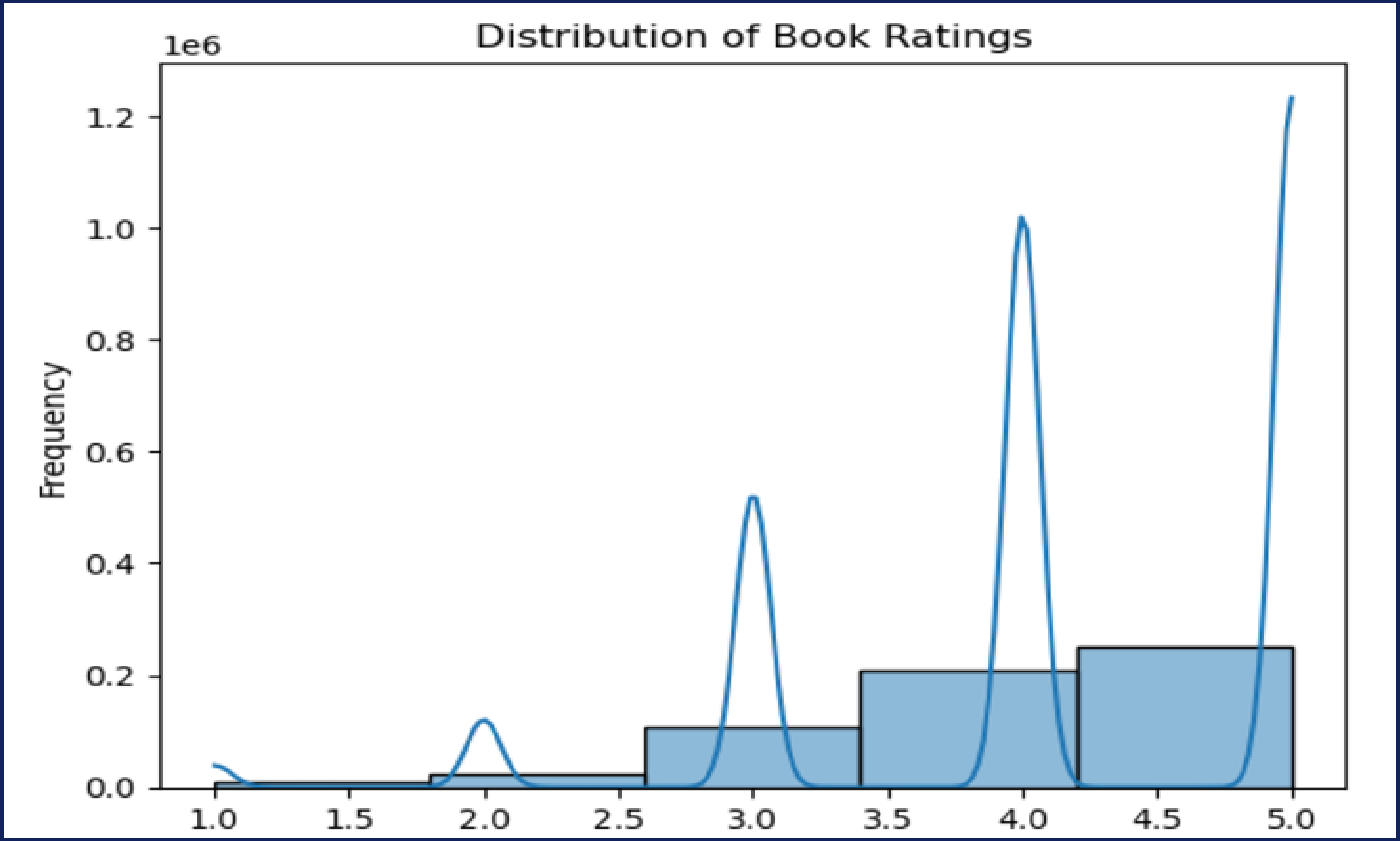
Handaling missing values

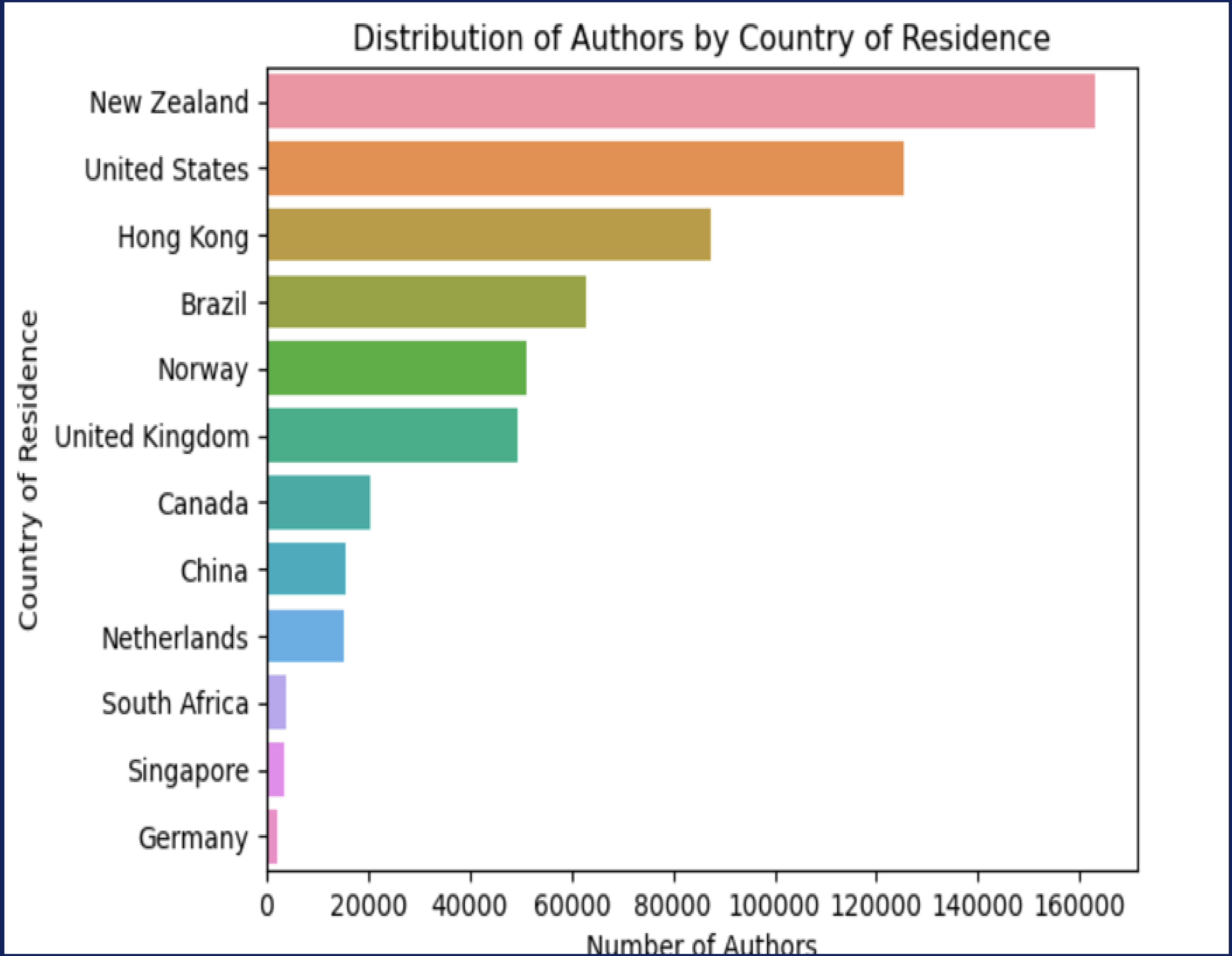


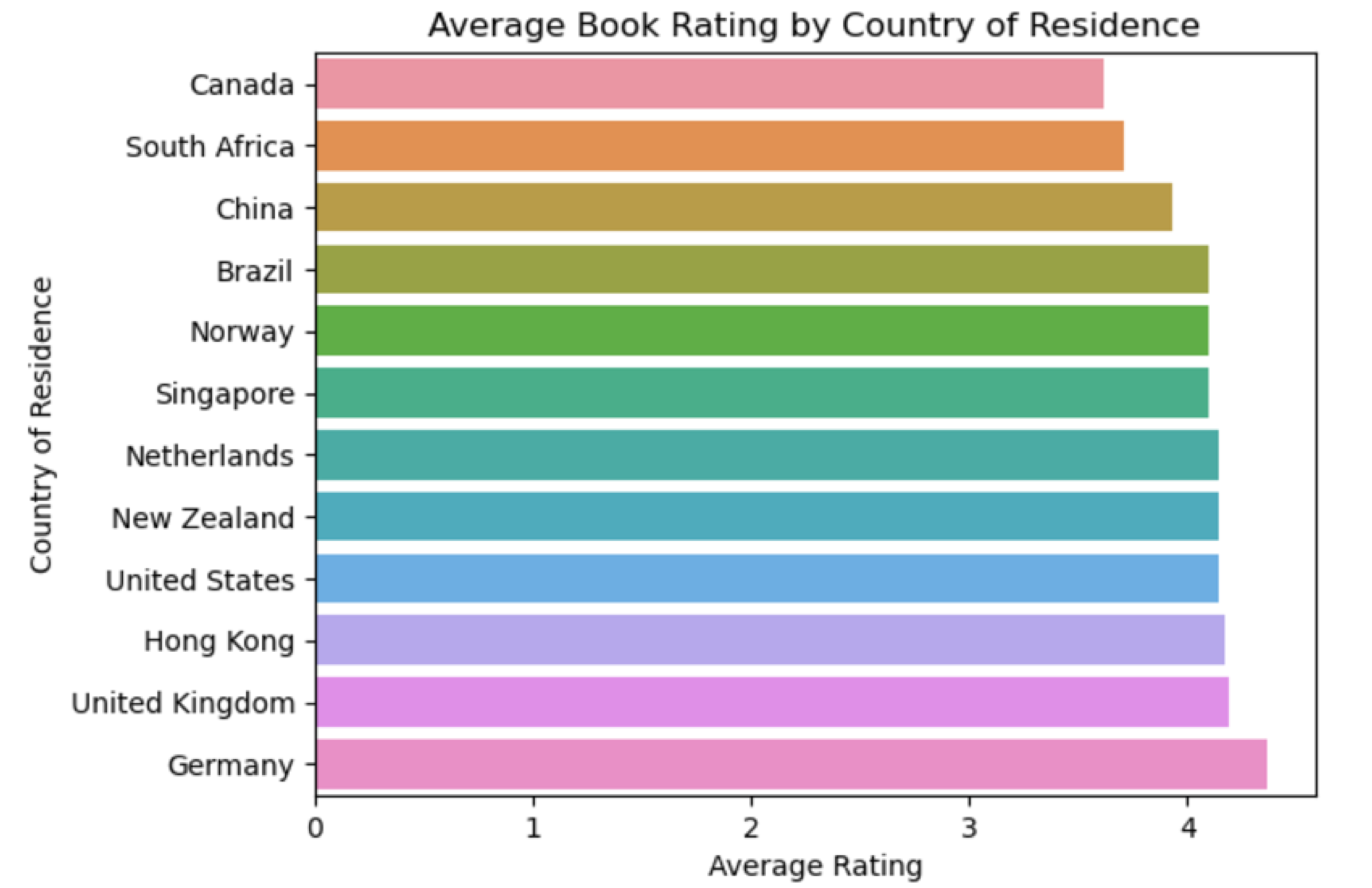
```python
import pandas as pd
# Replace missing values in numeric columns with median or mean as appropriate
numeric_columns = ['CheckoutMonth', 'Number of Checkouts', 'Rating', 'ReviewerID', 'ReviewID', 'Hrs Writing per Day']

# Use median for numeric columns (you can change to mean if preferred)
df[numeric_columns] = df[numeric_columns].fillna(df[numeric_columns].median())

# Replace missing values in categorical columns with mode (most frequent value)
categorical_columns = ['Title', 'AuthID', 'First Name', 'Last Name', 'Birthday', 'Country of Residence', 'BookID']

for col in categorical_columns:
    df[col] = df[col].fillna(df[col].mode()[0])  # Fill with the most frequent value

# Check the result
print(df.isnull().sum())  # This will show if there are any remaining missing values
```

```
BookID                  0
Title                   0
AuthID                  0
First Name              0
Last Name               0
Birthday                0
Country of Residence    0
Hrs Writing per Day     0
CheckoutMonth           0
Number of Checkouts     0
Rating                  0
ReviewerID              0
ReviewID                0
dtype: int64
```

VISUALIZATION



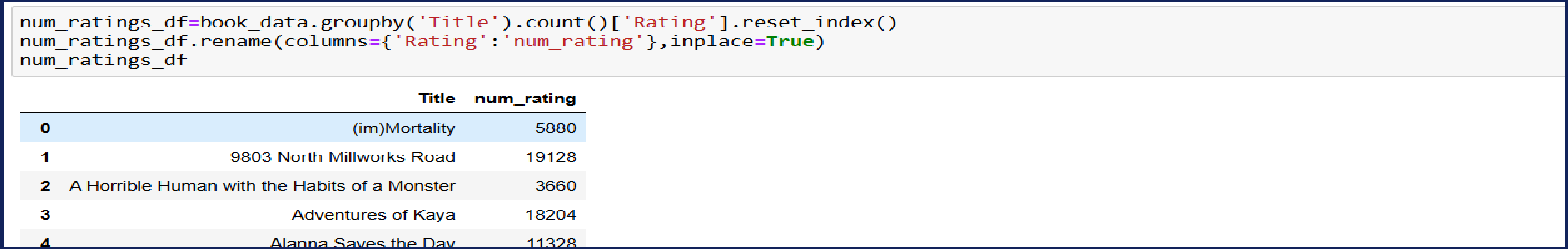

Distribution of Authors by Country of Residence
Country of Residence
New Zealand
United States
Hong Kong
Brazil
Norway
United Kingdom
Canada
China
Netherlands
South Africa
Singapore
Germany
0
20000
40000
60000
80000
100000
120000
140000
160000
Number of Authors


Average Book Rating by Country of Residence
Country of Residence
Canada
South Africa
China
Brazil
Norway
Singapore
Netherlands
New Zealand
United States
Hong Kong
United Kingdom
Germany
0
1
2
3
4
Average Rating

POPULARITY BASED RECOMMENDATION SYSTEM

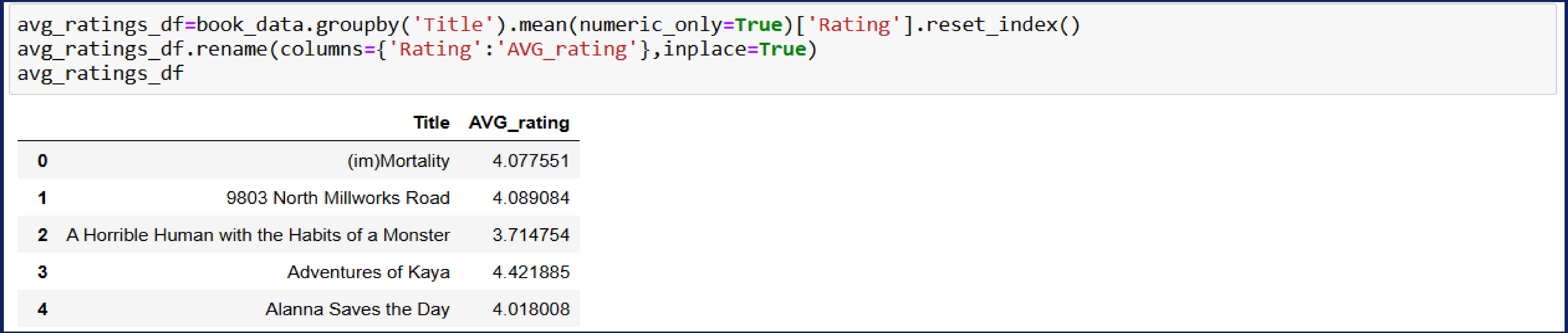
Popularity based recommendation systems are based on the rating of items

- Popularity based recommendation systems works with the trend.
- It basically uses the items which are in trend right now.



```
num_ratings_df=book_data.groupby('Title').count()['Rating'].reset_index()
num_ratings_df.rename(columns={'Rating':'num_rating'},inplace=True)
num_ratings_df
```

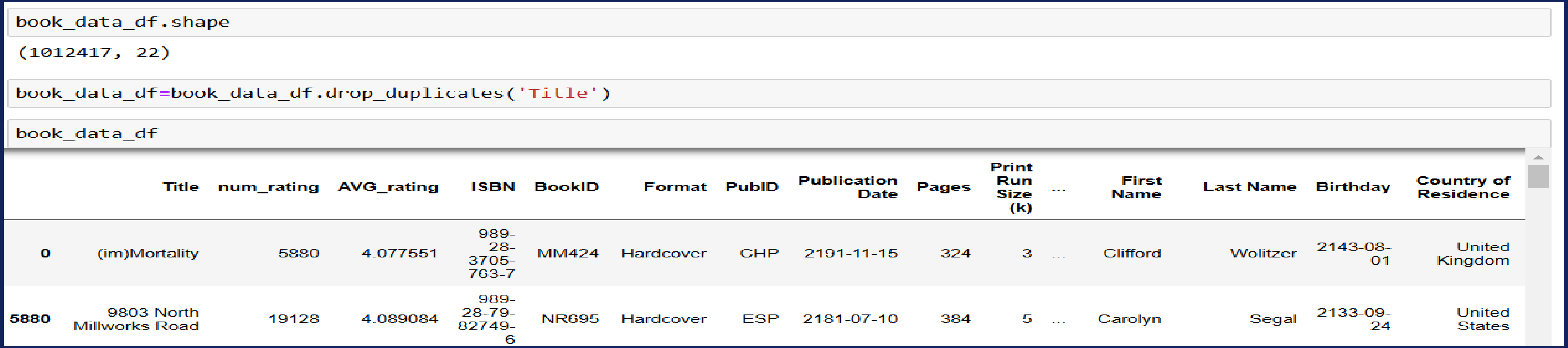
	Title	num_rating
0	(im)Mortality	5880
1	9803 North Millworks Road	19128
2	A Horrible Human with the Habits of a Monster	3660
3	Adventures of Kaya	18204
4	Alanna Saves the Day	11328



```
avg_ratings_df=book_data.groupby('Title').mean(numeric_only=True)['Rating'].reset_index()
avg_ratings_df.rename(columns={'Rating':'AVG_rating'},inplace=True)
avg_ratings_df
```

	Title	AVG_rating
0	(im)Mortality	4.077551
1	9803 North Millworks Road	4.089084
2	A Horrible Human with the Habits of a Monster	3.714754
3	Adventures of Kaya	4.421885
4	Alanna Saves the Day	4.018008

Collaborative Filtering Based Recommender System



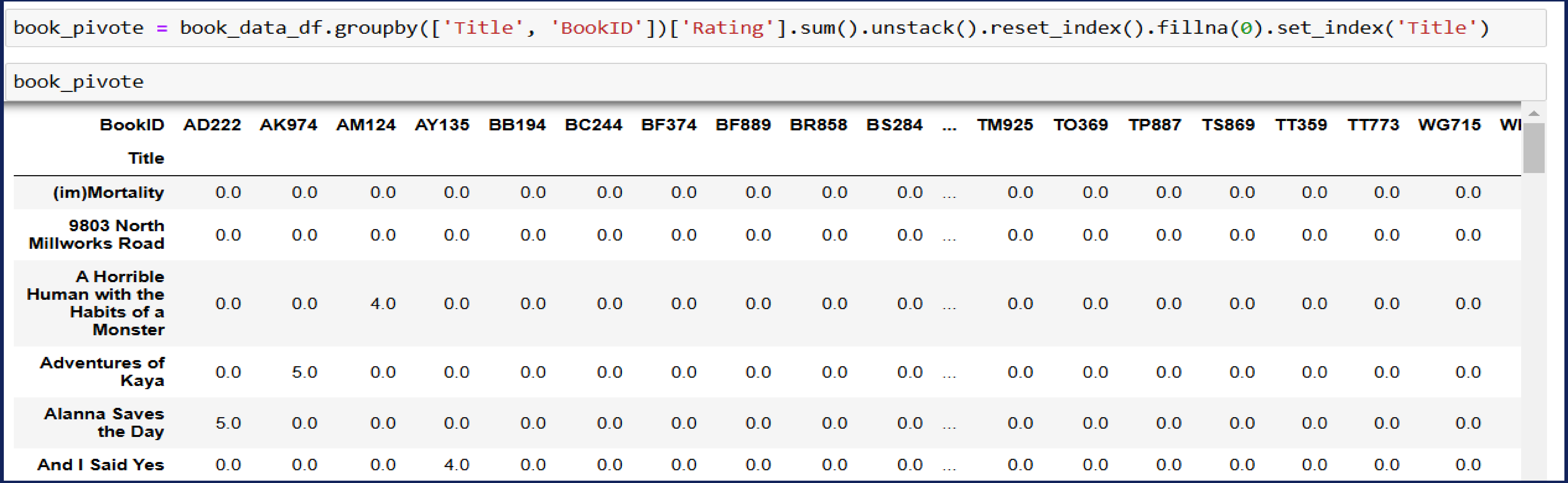
```
book_data_df.shape
```

```
(1012417, 22)
```

```
book_data_df=book_data_df.drop_duplicates('Title')
```

```
book_data_df
```

	Title	num_rating	AVG_rating	ISBN	BookID	Format	PubID	Publication Date	Pages	Print Run Size (k)	...	First Name	Last Name	Birthday	Country of Residence
0	(im)Mortality	5880	4.077551	989-28-3705-763-7	MM424	Hardcover	CHP	2191-11-15	324	3	...	Clifford	Wolitzer	2143-08-01	United Kingdom
5880	9803 North Millworks Road	19128	4.089084	989-28-79-82749-6	NR695	Hardcover	ESP	2181-07-10	384	5	...	Carolyn	Segal	2133-09-24	United States

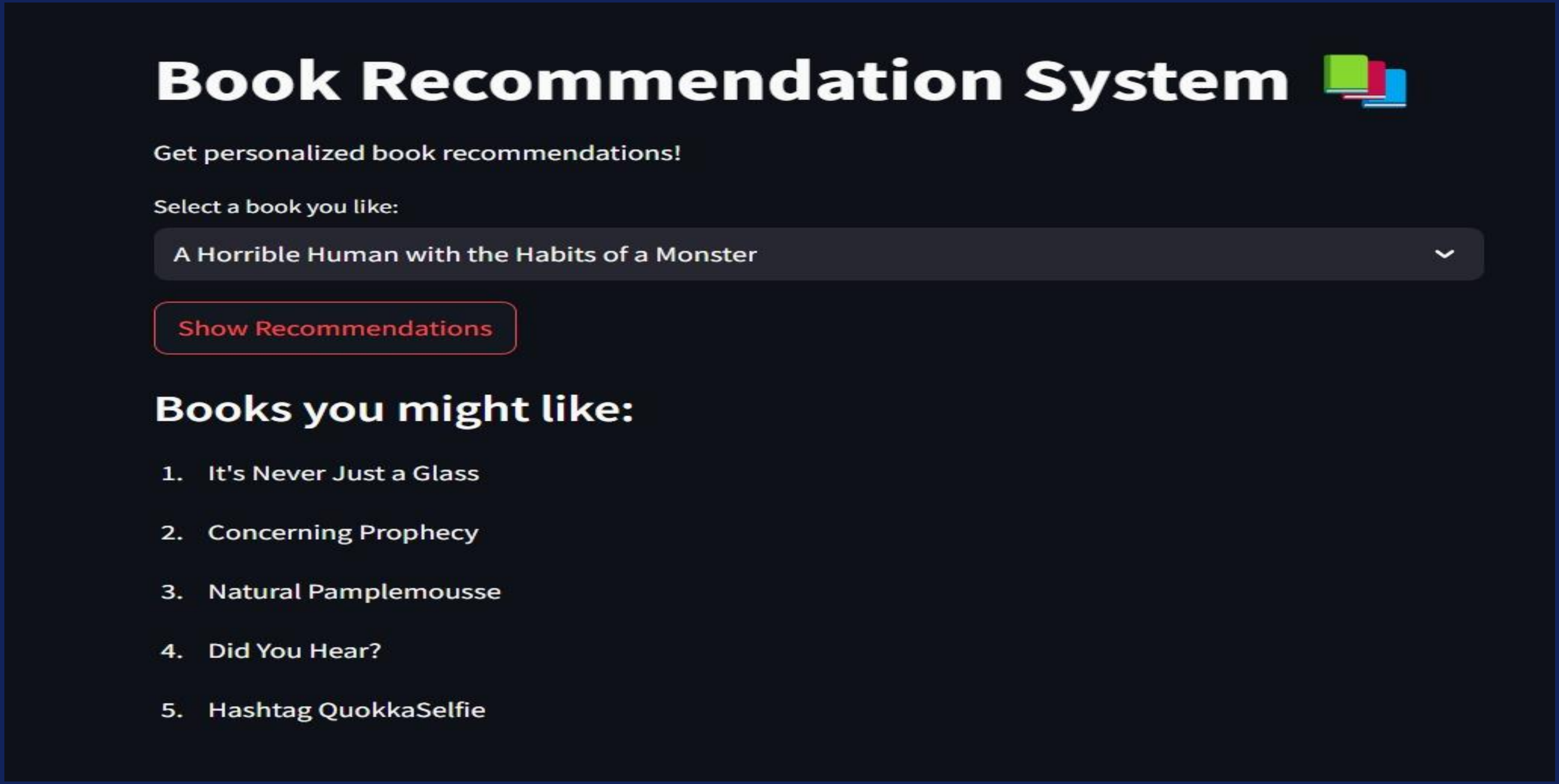


```
book_pivote = book_data_df.groupby(['Title', 'BookID'])['Rating'].sum().unstack().reset_index().fillna(0).set_index('Title')
```

```
book_pivote
```

BookID	AD222	AK974	AM124	AY135	BB194	BC244	BF374	BF889	BR858	BS284	...	TM925	TO369	TP887	TS869	TT359	TT773	WG715	WI
Title																			
(im)Mortality	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
9803 North Millworks Road	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
A Horrible Human with the Habits of a Monster	0.0	0.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
Adventures of Kaya	0.0	5.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
Alanna Saves the Day	5.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
And I Said Yes	0.0	0.0	0.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

DEPLOYMENT



Book Recommendation System 📚

Get personalized book recommendations!

Select a book you like:

A Horrible Human with the Habits of a Monster

Books you might like:

1. It's Never Just a Glass
2. Concerning Prophecy
3. Natural Pamplemousse
4. Did You Hear?
5. Hashtag QuokkaSelfie

Conclusion

The development and implementation of a book recommendation system provide significant value to users by simplifying the process of discovering new books tailored to their interests. This system leverages various recommendation techniques, such as collaborative filtering, content-based filtering, and hybrid approaches, to generate personalized suggestions that enhance user engagement and satisfaction.

THANK YOU