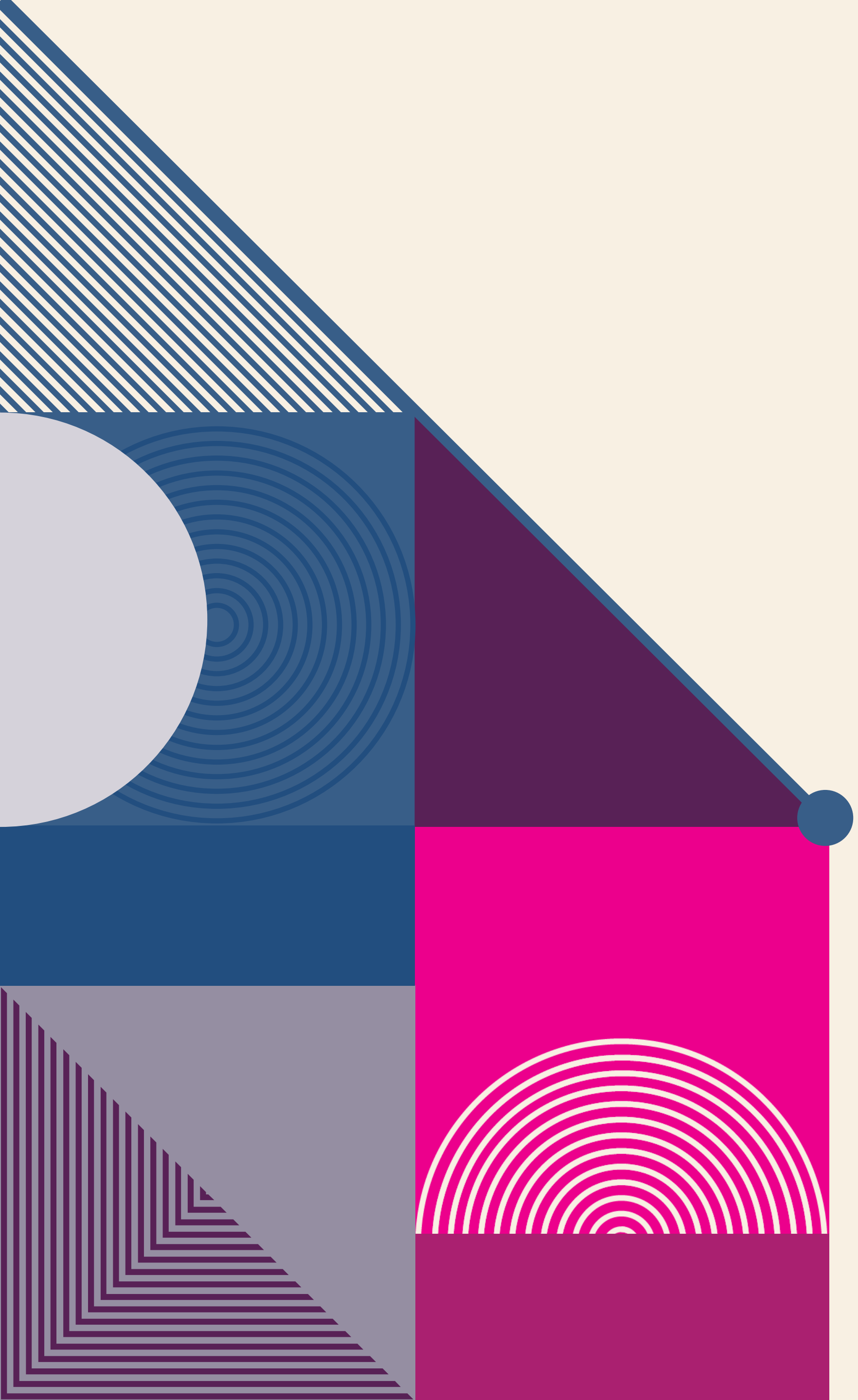
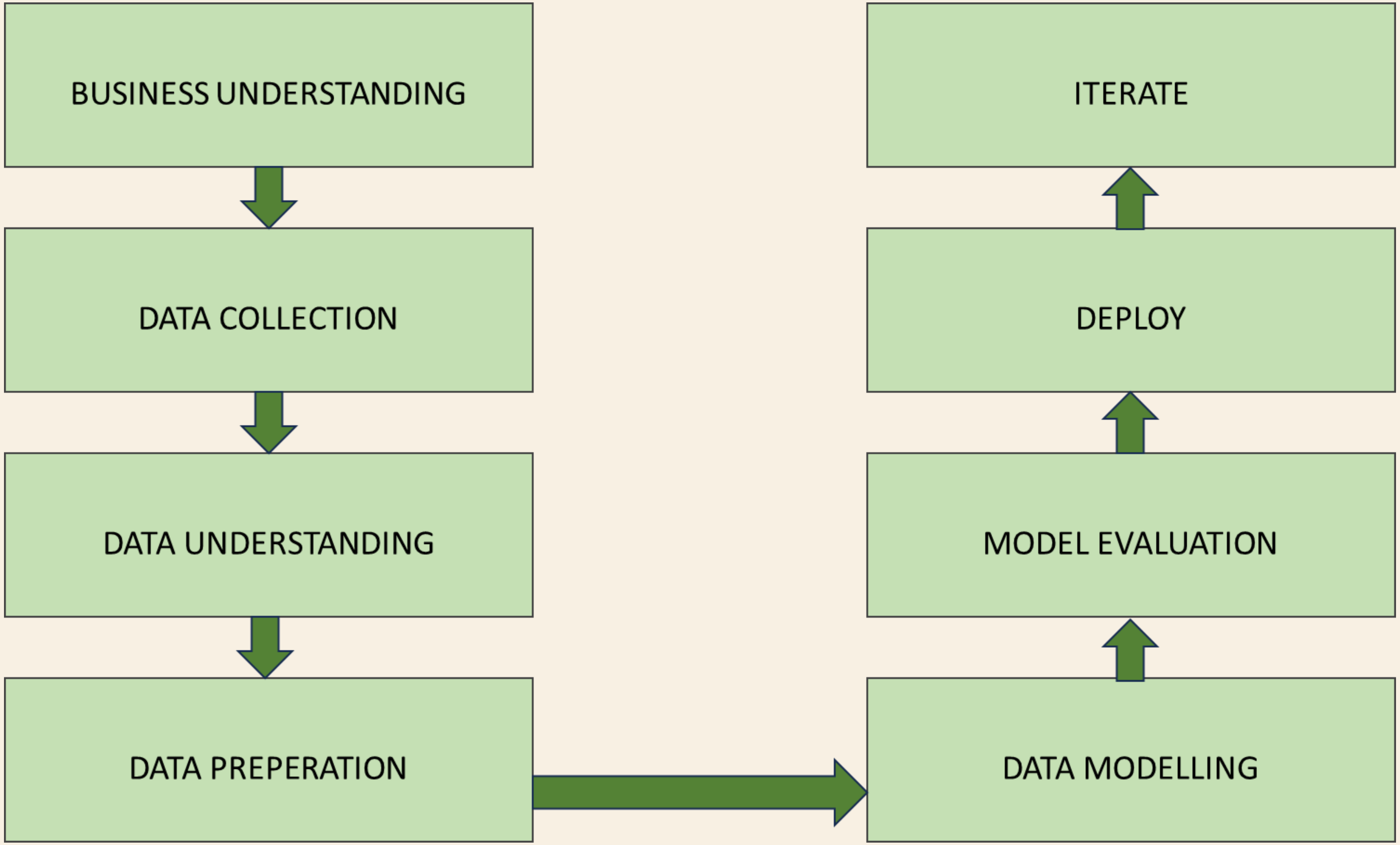
BOOKS RECCOMENDATION SYSTEM



BUSINESS OBJECTIVE

Generate the features from the dataset and use them to recommend the books accordingly to the users.

PROJECT FLOW



BUSINESS UNDERSTANDING

DATA COLLECTION

DATA UNDERSTANDING

DATA PREPERATION

DATA MODELLING

MODEL EVALUATION

DEPLOY

ITERATE

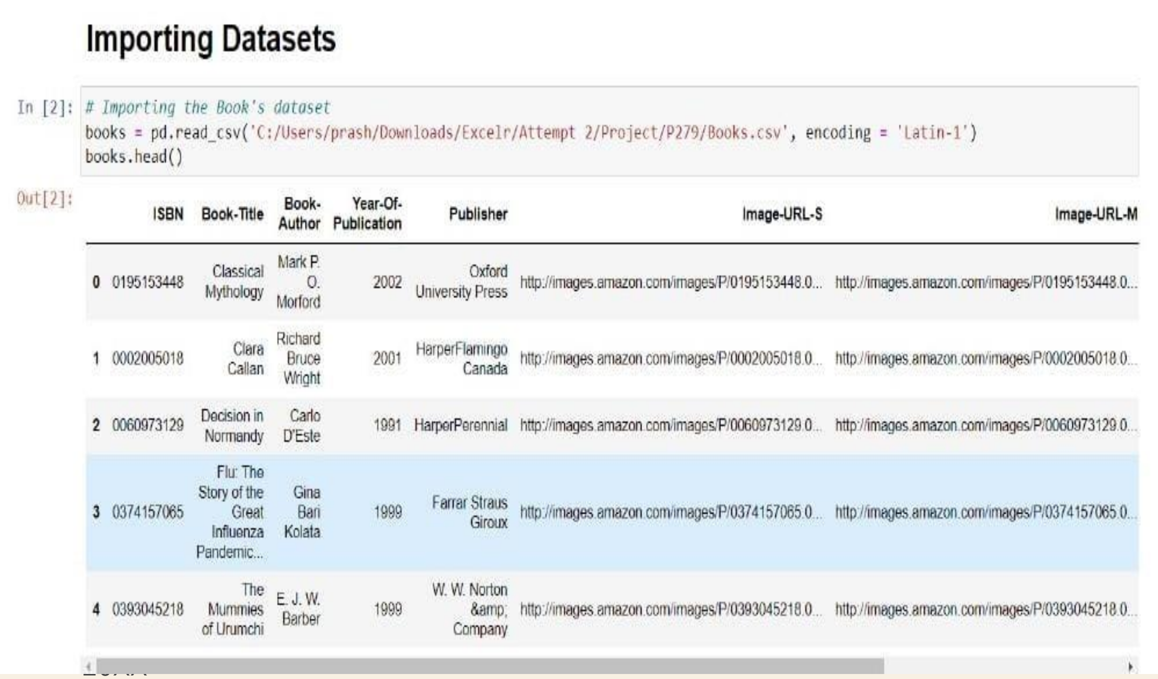
EXPLORATORY DATA ANALYSIS [EDA]

EXPLORATORY DATA ANALYSIS [EDA]

During our data exploration, we encountered three datasets: "books," "users," and "ratings."

The "books" dataset comprises 271,360 entries with eight columns.

Dataset [BOOKS] Details are as follows :

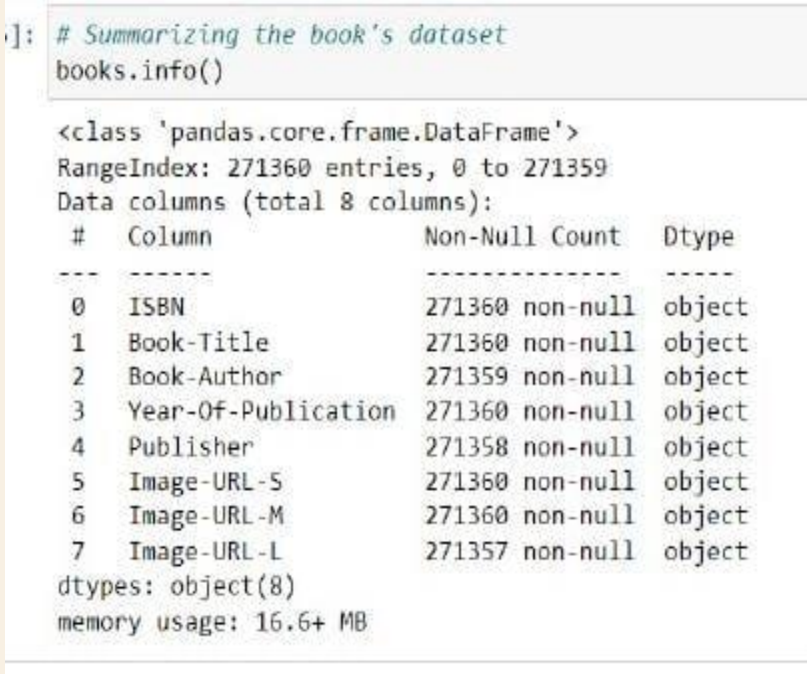


Importing Datasets

```
In [2]: # Importing the Book's dataset
        books = pd.read_csv('C:/Users/prash/Downloads/Excelr/Attempt 2/Project/P279/Books.csv', encoding = 'Latin-1')
        books.head()
```

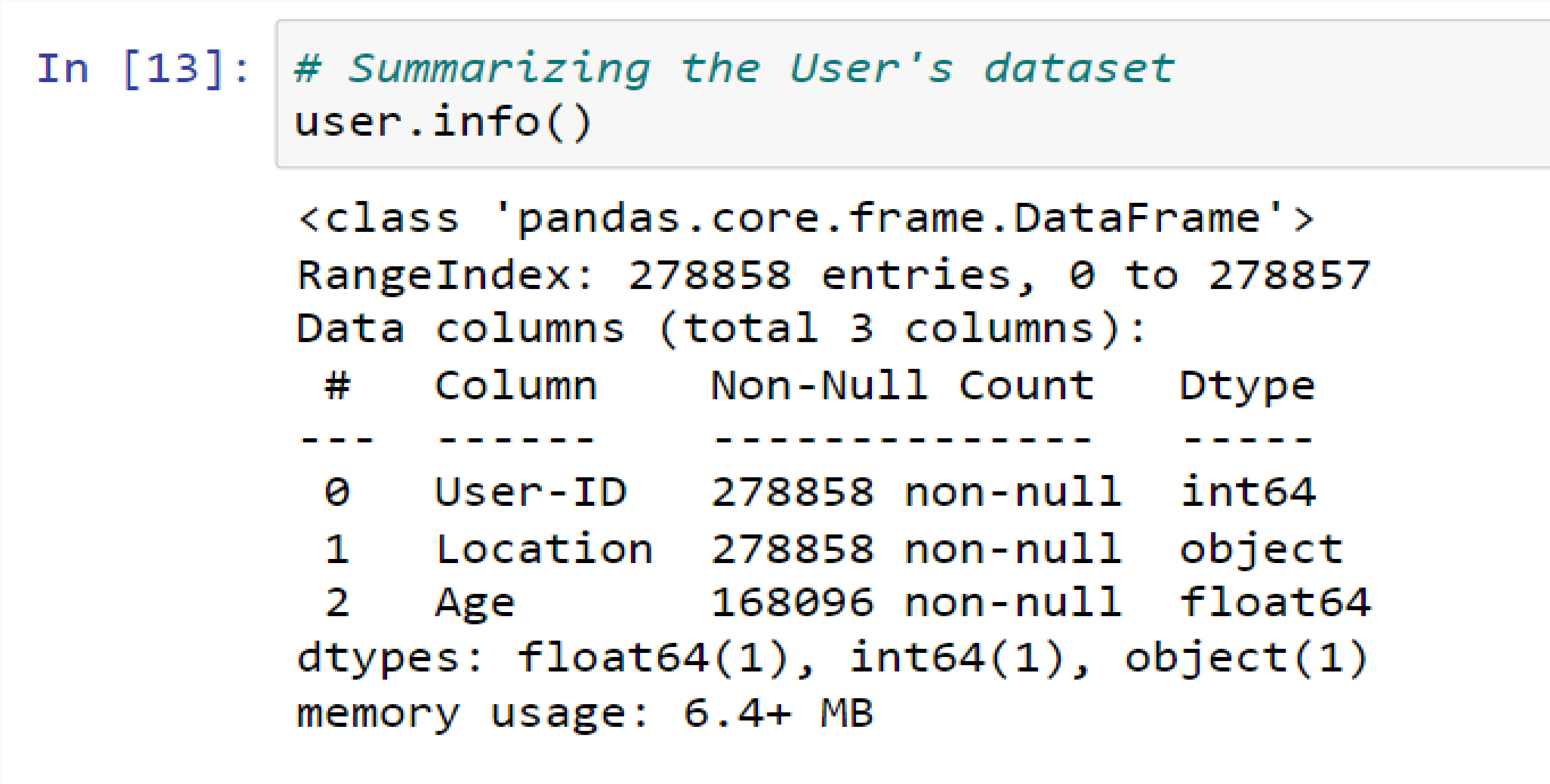
Out[2]:

	ISBN	Book-Title	Book-Author	Year-Of-Publication	Publisher	Image-URL-S	Image-URL-M
0	0195153448	Classical Mythology	Mark P. O. Morford	2002	Oxford University Press	http://images.amazon.com/images/P/0195153448.0...	http://images.amazon.com/images/P/0195153448.0...
1	0002005018	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	http://images.amazon.com/images/P/0002005018.0...	http://images.amazon.com/images/P/0002005018.0...
2	0060973129	Decision in Normandy	Carlo D'Este	1991	HarperPerennial	http://images.amazon.com/images/P/0060973129.0...	http://images.amazon.com/images/P/0060973129.0...
3	0374157065	Flu: The Story of the Great Influenza Pandemic...	Gina Bari Kolata	1999	Farrar Straus Giroux	http://images.amazon.com/images/P/0374157065.0...	http://images.amazon.com/images/P/0374157065.0...
4	0393045218	The Mummies of Urumchi	E. J. W. Barber	1999	W. W. Norton &amp; Company	http://images.amazon.com/images/P/0393045218.0...	http://images.amazon.com/images/P/0393045218.0...



```
]: # Summarizing the book's dataset
   books.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 271360 entries, 0 to 271359
Data columns (total 8 columns):
 #   Column               Non-Null Count   Dtype 
---  ------               --------------   ----- 
 0   ISBN                 271360 non-null  object
 1   Book-Title           271360 non-null  object
 2   Book-Author          271359 non-null  object
 3   Year-Of-Publication  271360 non-null  object
 4   Publisher            271358 non-null  object
 5   Image-URL-S          271360 non-null  object
 6   Image-URL-M          271360 non-null  object
 7   Image-URL-L          271357 non-null  object
dtypes: object(8)
memory usage: 16.6+ MB
```

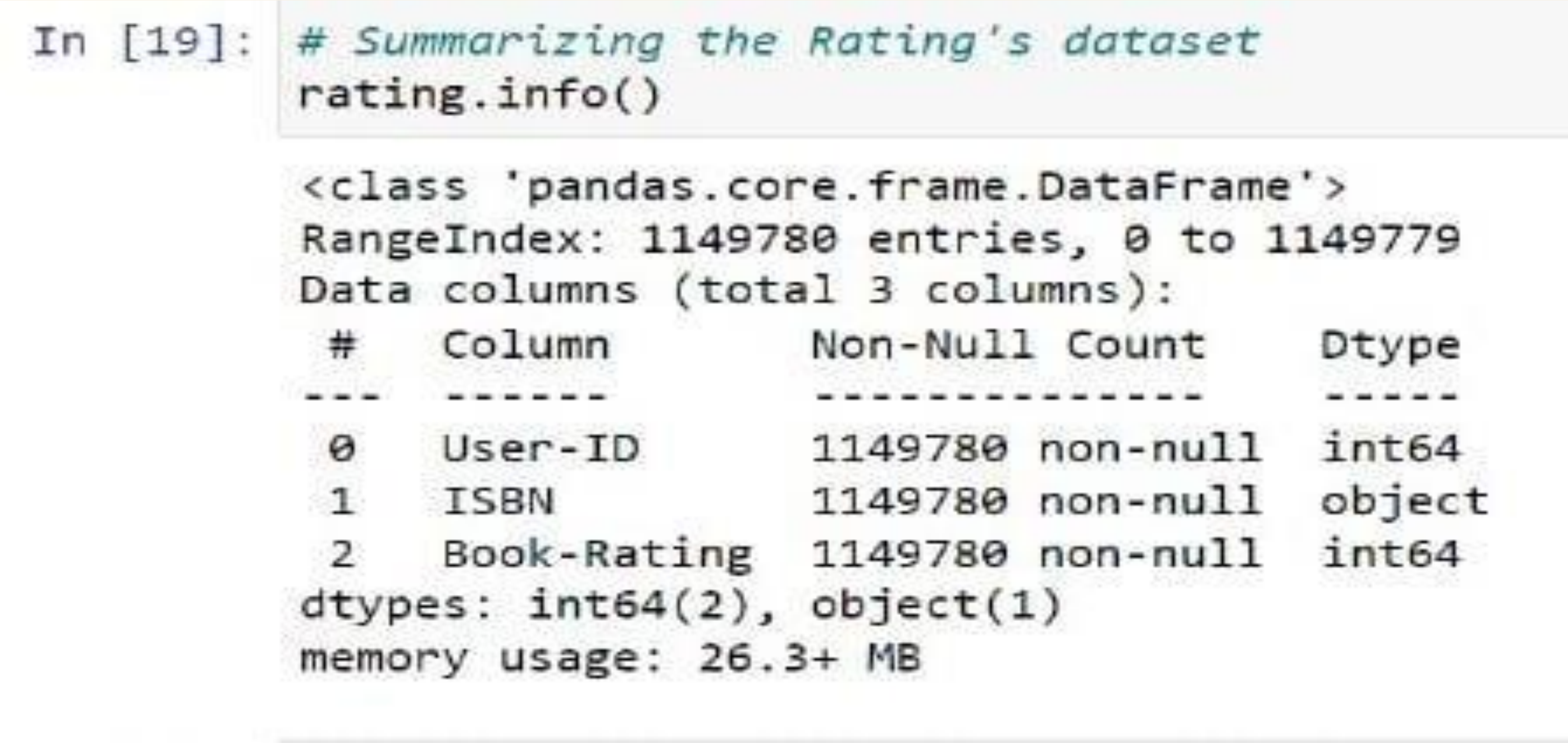
Dataset [USERS] Details are as follows :



```
In [13]: # Summarizing the User's dataset
         user.info()
```

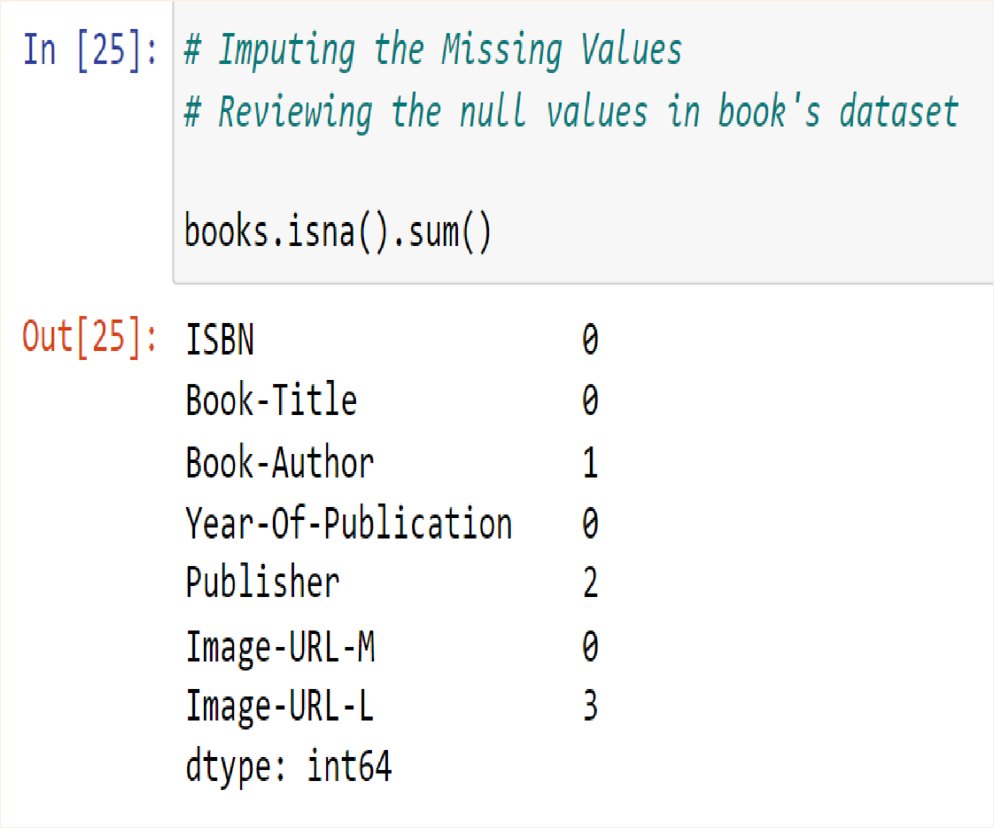
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 278858 entries, 0 to 278857
Data columns (total 3 columns):
 #   Column    Non-Null Count   Dtype  
---  ------    --------------   -----  
 0   User-ID   278858 non-null  int64  
 1   Location  278858 non-null  object 
 2   Age       168096 non-null  float64
dtypes: float64(1), int64(1), object(1)
memory usage: 6.4+ MB
```

Dataset [RATINGS] Details are as follows :

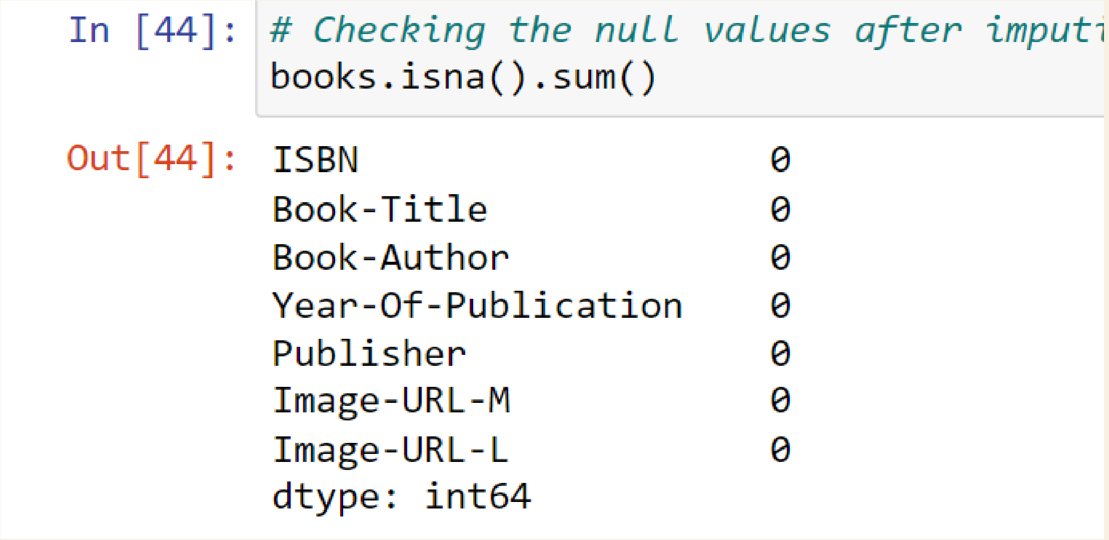


```
In [19]: # Summarizing the Rating's dataset
         rating.info()

         <class 'pandas.core.frame.DataFrame'>
         RangeIndex: 1149780 entries, 0 to 1149779
         Data columns (total 3 columns):
          #   Column       Non-Null Count    Dtype
         ---  ------       --------------    -----
          0   User-ID      1149780 non-null  int64
          1   ISBN         1149780 non-null  object
          2   Book-Rating  1149780 non-null  int64
         dtypes: int64(2), object(1)
         memory usage: 26.3+ MB
```

- During our analysis of the "books" dataset, we detected a number of missing values and incorrect entries. To resolve this issue, we conducted imputation by obtaining the necessary information from online sources, resulting in the dataset being corrected.

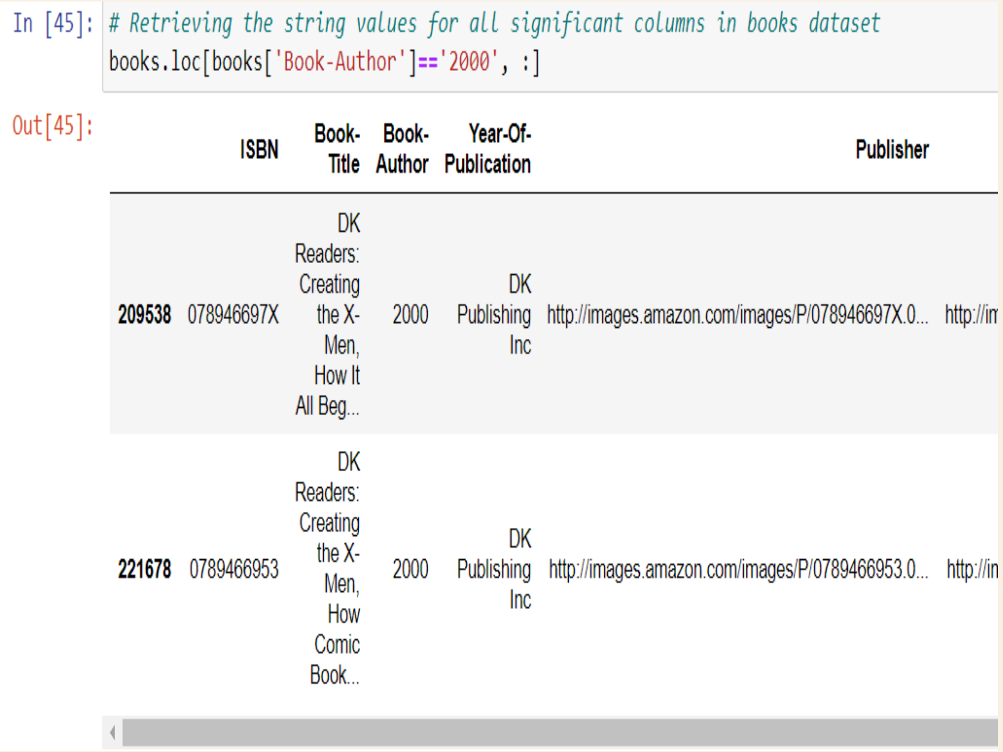


```
In [25]: # Imputing the Missing Values
         # Reviewing the null values in book's dataset

         books.isna().sum()

Out[25]: ISBN                 0
         Book-Title           0
         Book-Author          1
         Year-Of-Publication  0
         Publisher            2
         Image-URL-M          0
         Image-URL-L          3
         dtype: int64
```



```
In [44]: # Checking the null values after imputi
         books.isna().sum()

Out[44]: ISBN                   0
         Book-Title             0
         Book-Author            0
         Year-Of-Publication    0
         Publisher              0
         Image-URL-M            0
         Image-URL-L            0
         dtype: int64
```

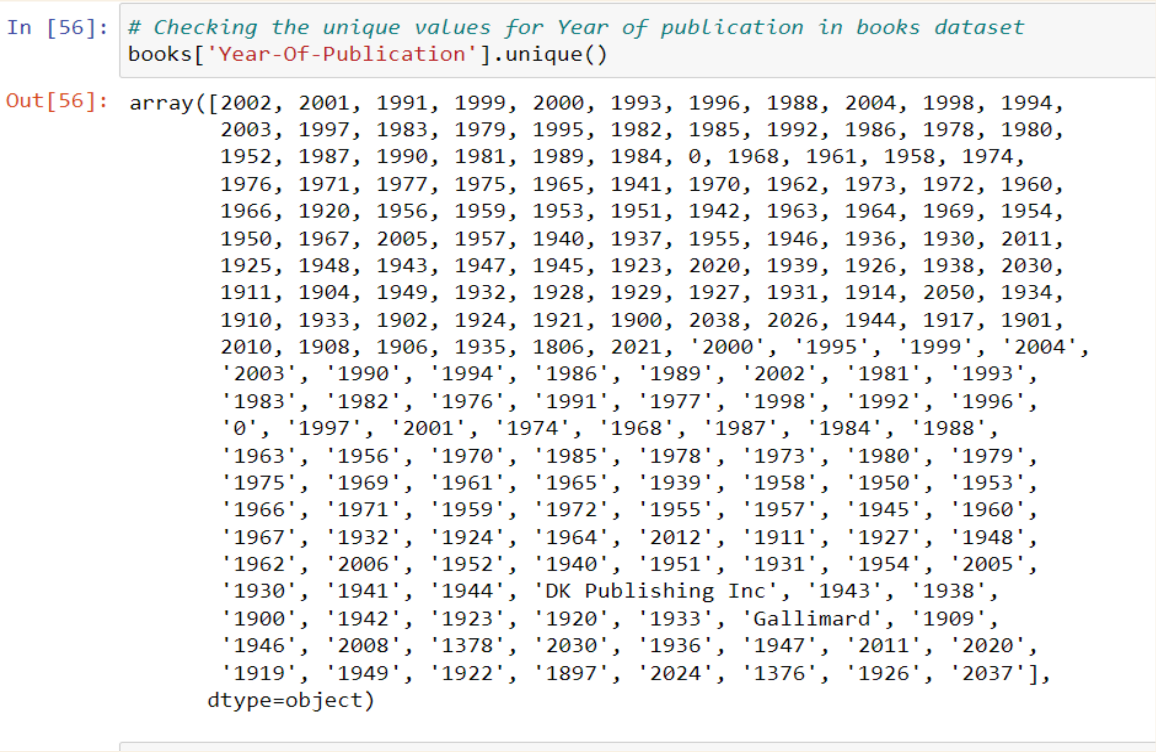
We found some mix-ups in the dataset's columns and fixed them. We also corrected any wrong information in all the columns. In the "Year of Publication" column, we noticed unusual values like zeros and dates from the future, and we fixed those too. Moreover, we removed any duplicate entries from the "books" dataset.



```
In [45]: # Retrieving the string values for all significant columns in books dataset
         books.loc[books['Book-Author']=='2000', :]
```

Out[45]:

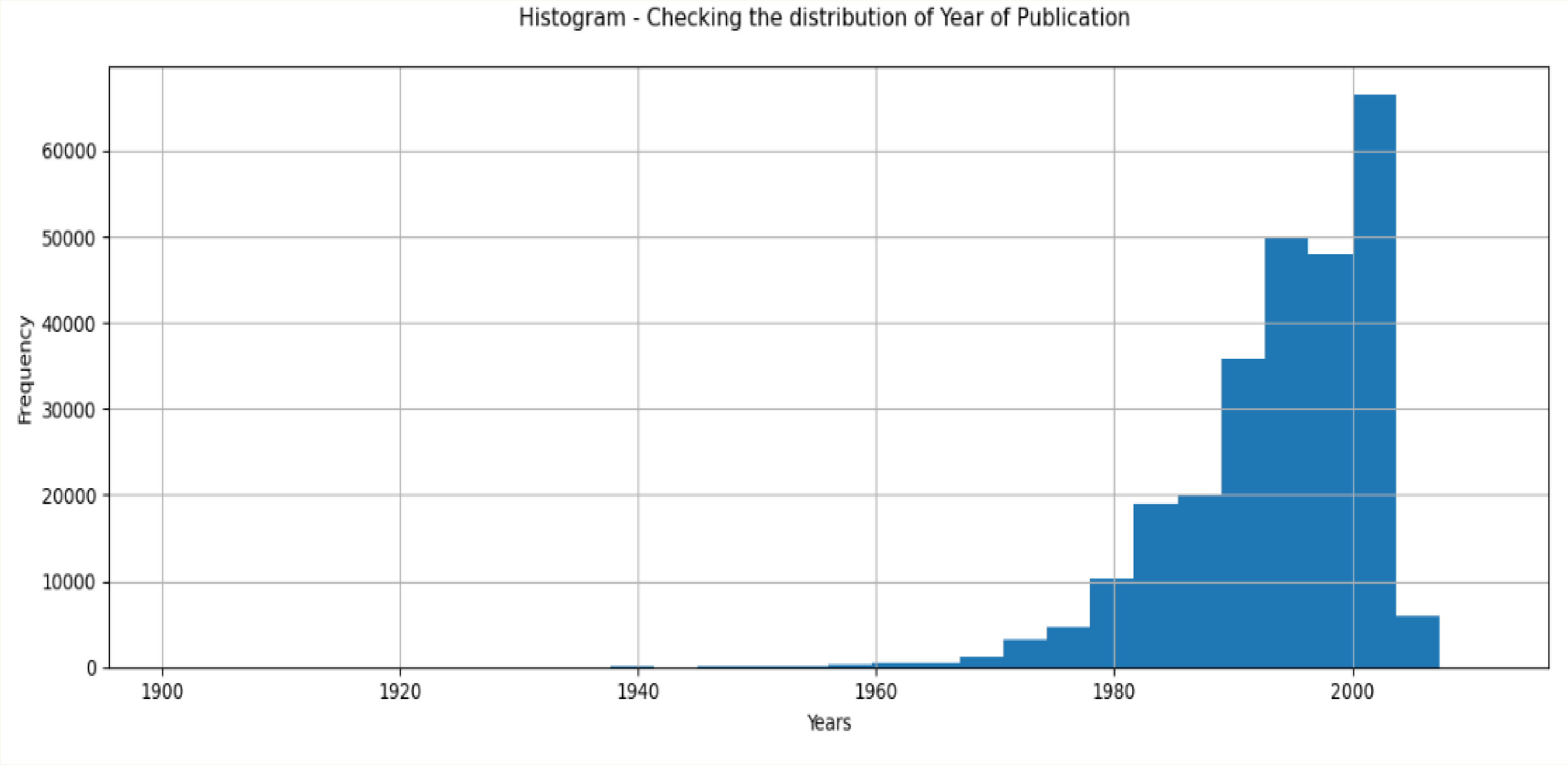
	ISBN	Book-Title	Book-Author	Year-Of-Publication	Publisher	
209538	078946697X	DK Readers: Creating the X-Men, How It All Beg...	2000	DK Publishing Inc	http://images.amazon.com/images/P/078946697X.0...	http://im
221678	0789466953	DK Readers: Creating the X-Men, How Comic Book...	2000	DK Publishing Inc	http://images.amazon.com/images/P/0789466953.0...	http://in



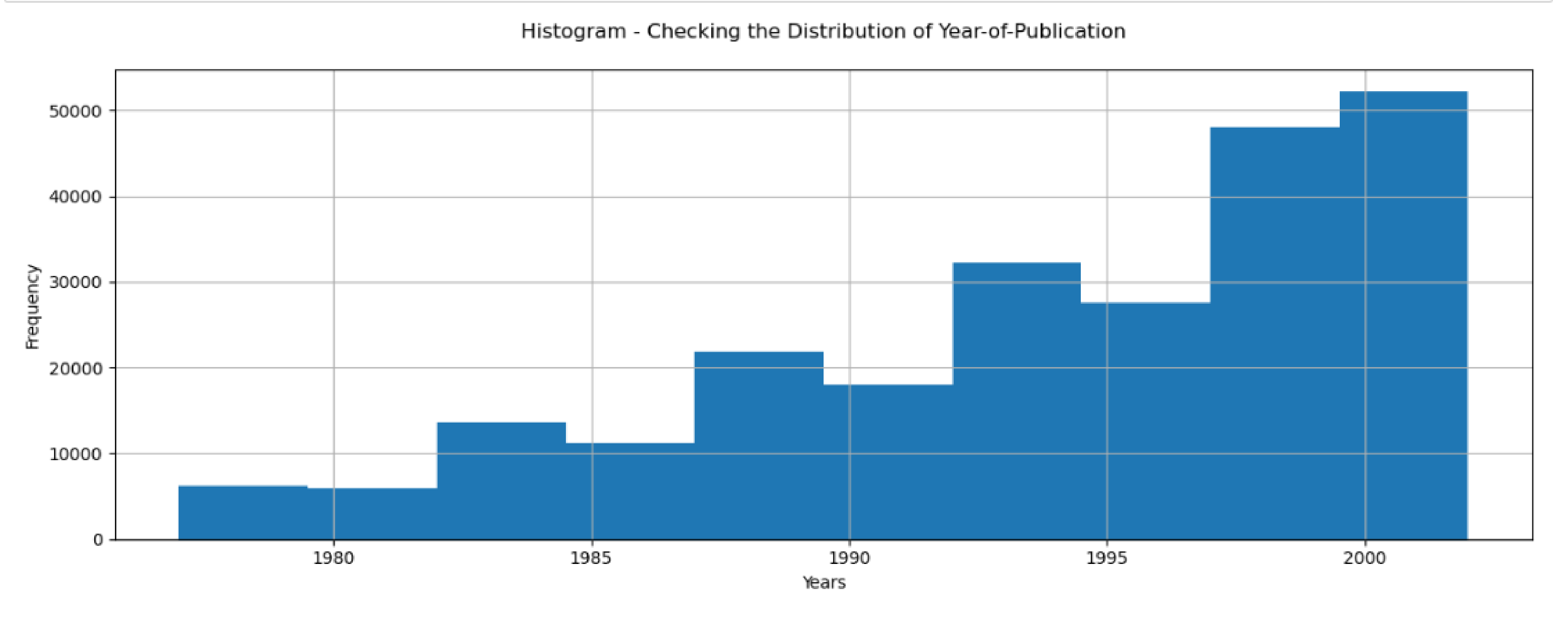
```
In [56]: # Checking the unique values for Year of publication in books dataset
         books['Year-Of-Publication'].unique()
```

```
Out[56]: array([2002, 2001, 1991, 1999, 2000, 1993, 1996, 1988, 2004, 1998, 1994,
                2003, 1997, 1983, 1979, 1995, 1982, 1985, 1992, 1986, 1978, 1980,
                1952, 1987, 1990, 1981, 1989, 1984, 0, 1968, 1961, 1958, 1974,
                1976, 1971, 1977, 1975, 1965, 1941, 1970, 1962, 1973, 1972, 1960,
                1966, 1920, 1956, 1959, 1953, 1951, 1942, 1963, 1964, 1969, 1954,
                1950, 1967, 2005, 1957, 1940, 1937, 1955, 1946, 1936, 1930, 2011,
                1925, 1948, 1943, 1947, 1945, 1923, 2020, 1939, 1926, 1938, 2030,
                1911, 1904, 1949, 1932, 1928, 1929, 1927, 1931, 1914, 2050, 1934,
                1910, 1933, 1902, 1924, 1921, 1900, 2038, 2026, 1944, 1917, 1901,
                2010, 1908, 1906, 1935, 1806, 2021, '2000', '1995', '1999', '2004',
                '2003', '1990', '1994', '1986', '1989', '2002', '1981', '1993',
                '1983', '1982', '1976', '1991', '1977', '1998', '1992', '1996',
                '0', '1997', '2001', '1974', '1968', '1987', '1984', '1988',
                '1963', '1956', '1970', '1985', '1978', '1973', '1980', '1979',
                '1975', '1969', '1961', '1965', '1939', '1958', '1950', '1953',
                '1966', '1971', '1959', '1972', '1955', '1957', '1945', '1960',
                '1967', '1932', '1924', '1964', '2012', '1911', '1927', '1948',
                '1962', '2006', '1952', '1940', '1951', '1931', '1954', '2005',
                '1930', '1941', '1944', 'DK Publishing Inc', '1943', '1938',
                '1900', '1942', '1923', '1920', '1933', 'Gallimard', '1909',
                '1946', '2008', '1378', '2030', '1936', '1947', '2011', '2020',
                '1919', '1949', '1922', '1897', '2024', '1376', '1926', '2037'],
               dtype=object)
```

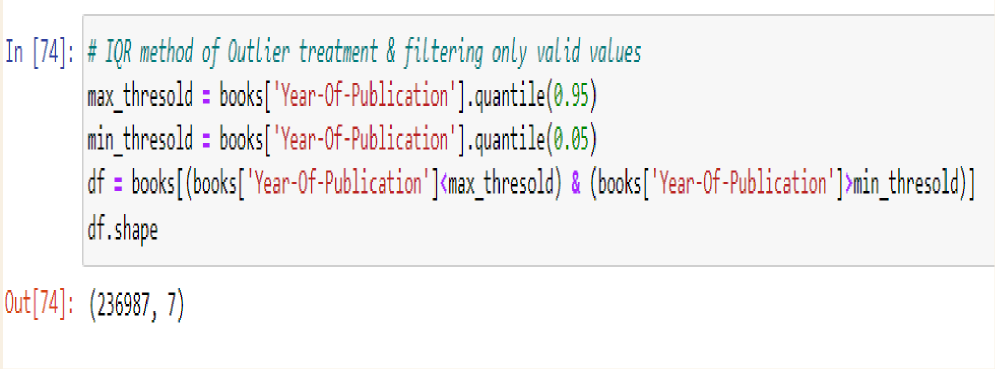
HISTOGRAM


Histogram - Checking the distribution of Year of Publication
Frequency
0
10000
20000
30000
40000
50000
60000
1900
1920
1940
1960
1980
2000
Years

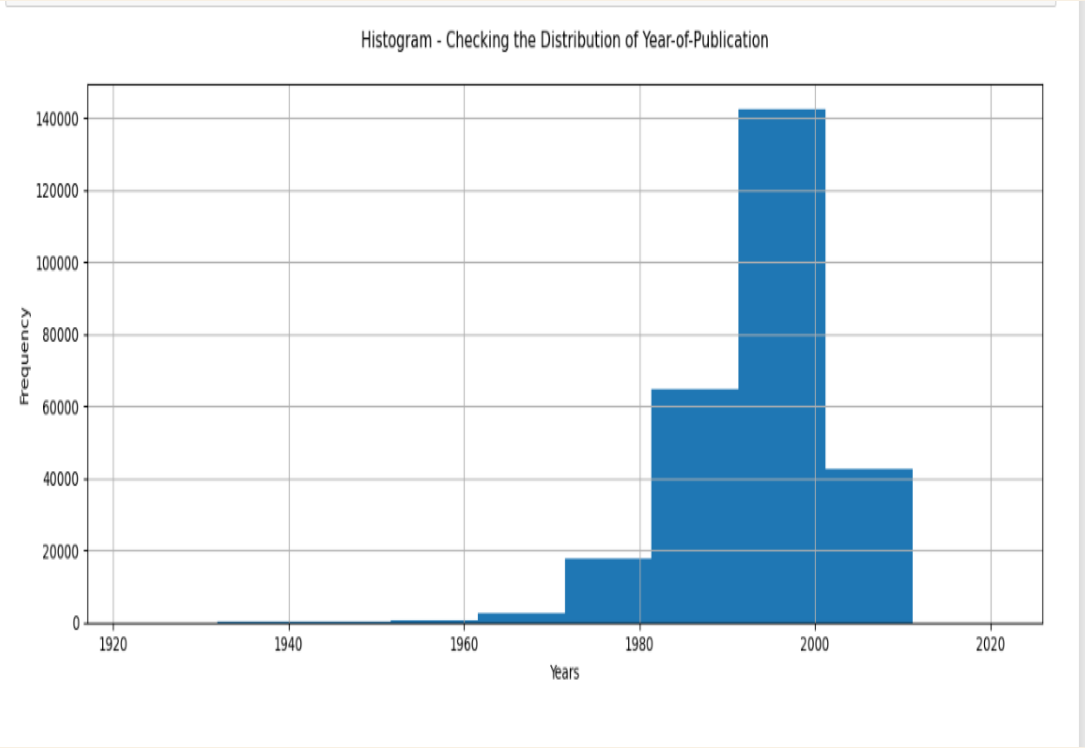
In the "Year of Publication" column, there are some values that stand out as unusual, and the data doesn't follow a normal distribution; it's skewed to the left. To handle this, we can't use the average (mean) because outliers can mess it up. Instead, we applied a method called Interquartile Range (IQR) to filter the data and remove extreme values, which improved our data's statistics and reduced skewness. Now, we can either use the middle value (median) or the most common value (mode) to replace the outliers.



After replacing the extreme outlier values with the most common year (2002), we've made the distribution more balanced. We've also effectively addressed incorrect zero values in the "Books" dataset. Now, let's apply similar fixes to the other datasets.

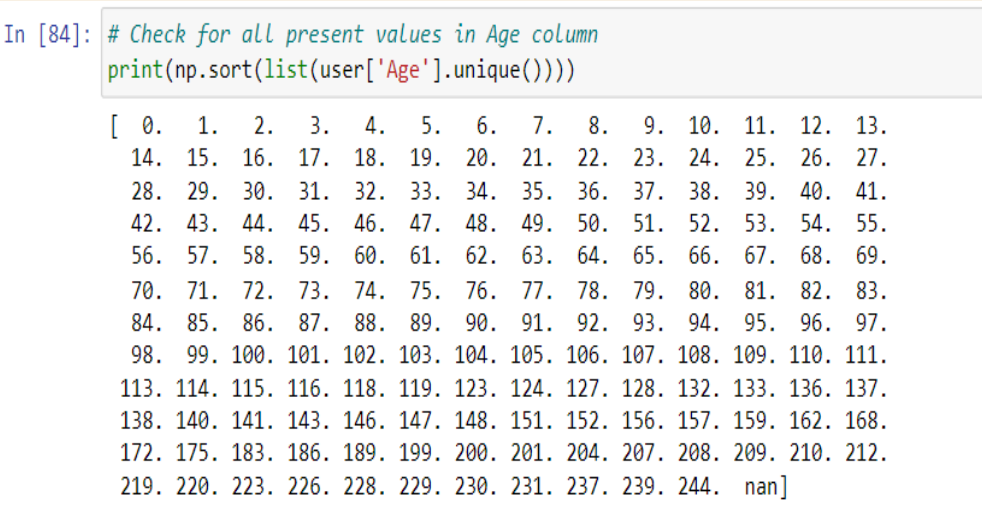


```
In [74]: # IQR method of Outlier treatment & filtering only valid values
         max_thresold = books['Year-Of-Publication'].quantile(0.95)
         min_thresold = books['Year-Of-Publication'].quantile(0.05)
         df = books[(books['Year-Of-Publication']<max_thresold) & (books['Year-Of-Publication']>min_thresold)]
         df.shape

Out[74]: (236987, 7)
```

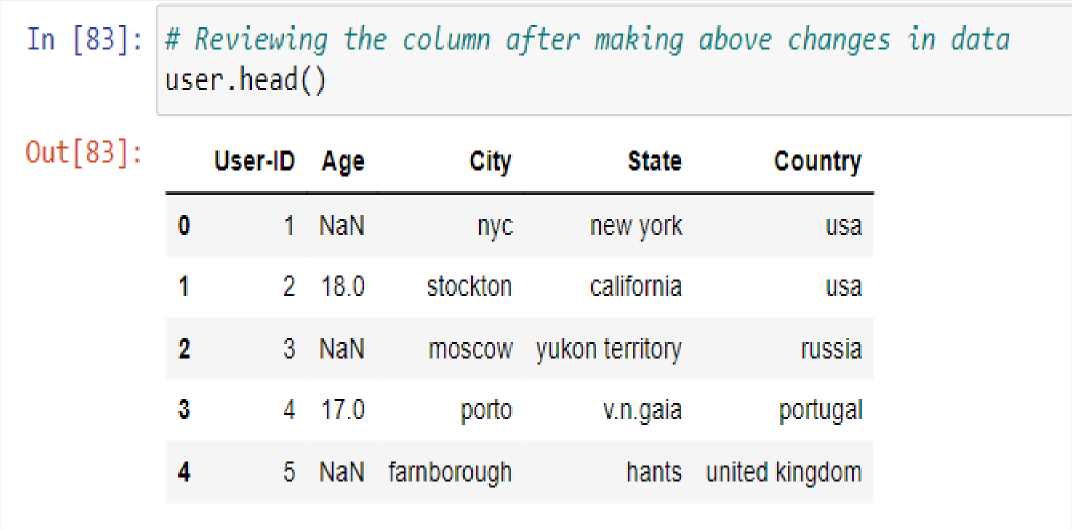


Histogram - Checking the Distribution of Year-of-Publication

In the "Users" dataset, all columns are important, so none will be removed. However, we found missing values in the "Age" column, which we will fill in. We also organized the "Location" column into separate state and country columns. Now, for the "Age" column, we'll handle both extremely high and missing values. We'll replace extreme ages with a range from 8 to 90 years because very young kids and individuals over 90 might not be independent readers. For the missing values, we'll use the median age, which is less affected by outliers and is around 32 years, close to the average age across states and countries.



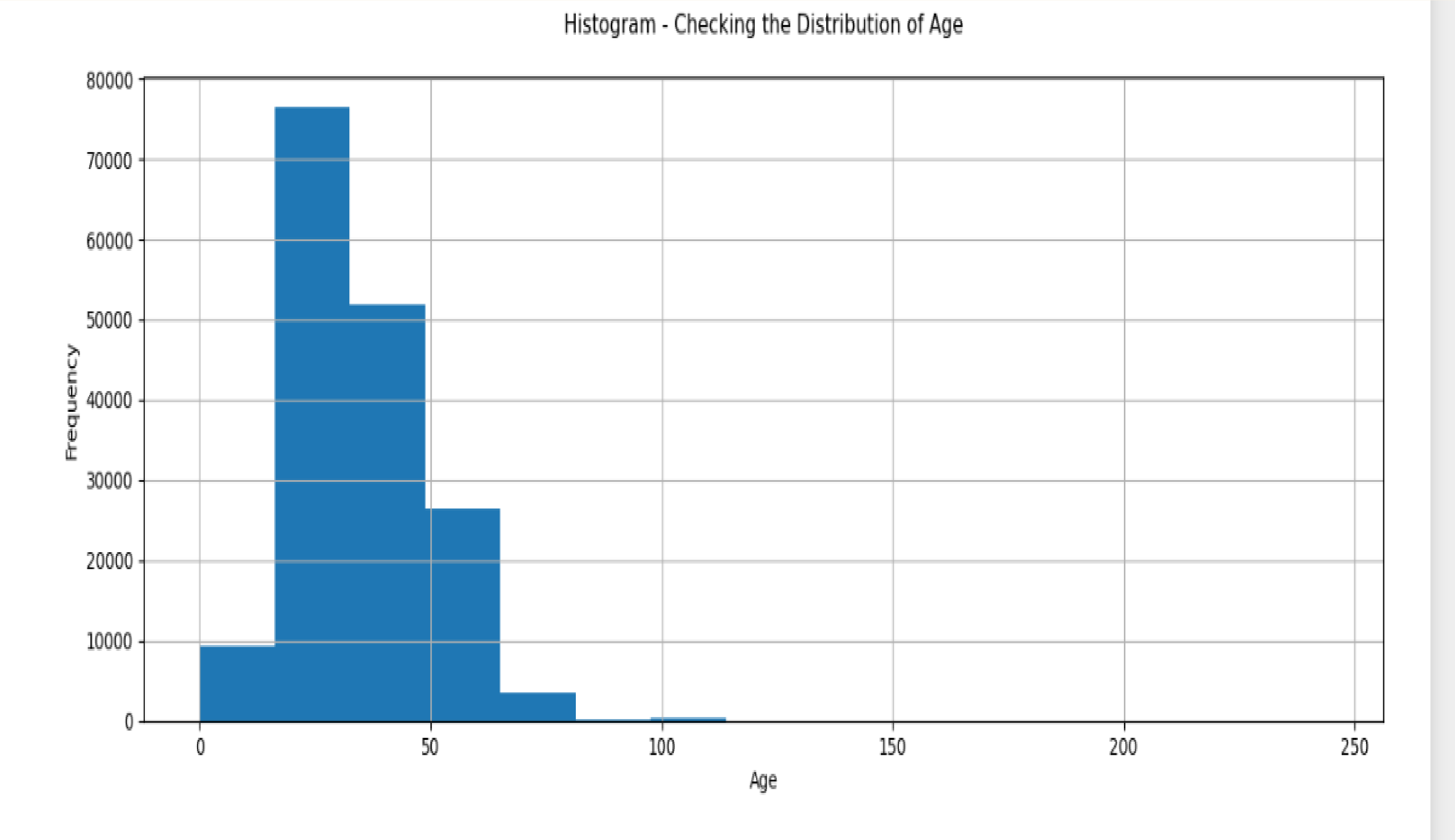
```
In [84]: # Check for all present values in Age column
         print(np.sort(list(user['Age'].unique())))

         [  0.   1.   2.   3.   4.   5.   6.   7.   8.   9.  10.  11.  12.  13.
           14.  15.  16.  17.  18.  19.  20.  21.  22.  23.  24.  25.  26.  27.
           28.  29.  30.  31.  32.  33.  34.  35.  36.  37.  38.  39.  40.  41.
           42.  43.  44.  45.  46.  47.  48.  49.  50.  51.  52.  53.  54.  55.
           56.  57.  58.  59.  60.  61.  62.  63.  64.  65.  66.  67.  68.  69.
           70.  71.  72.  73.  74.  75.  76.  77.  78.  79.  80.  81.  82.  83.
           84.  85.  86.  87.  88.  89.  90.  91.  92.  93.  94.  95.  96.  97.
           98.  99. 100. 101. 102. 103. 104. 105. 106. 107. 108. 109. 110. 111.
          113. 114. 115. 116. 118. 119. 123. 124. 127. 128. 132. 133. 136. 137.
          138. 140. 141. 143. 146. 147. 148. 151. 152. 156. 157. 159. 162. 168.
          172. 175. 183. 186. 189. 199. 200. 201. 204. 207. 208. 209. 210. 212.
          219. 220. 223. 226. 228. 229. 230. 231. 237. 239. 244.  nan]
```

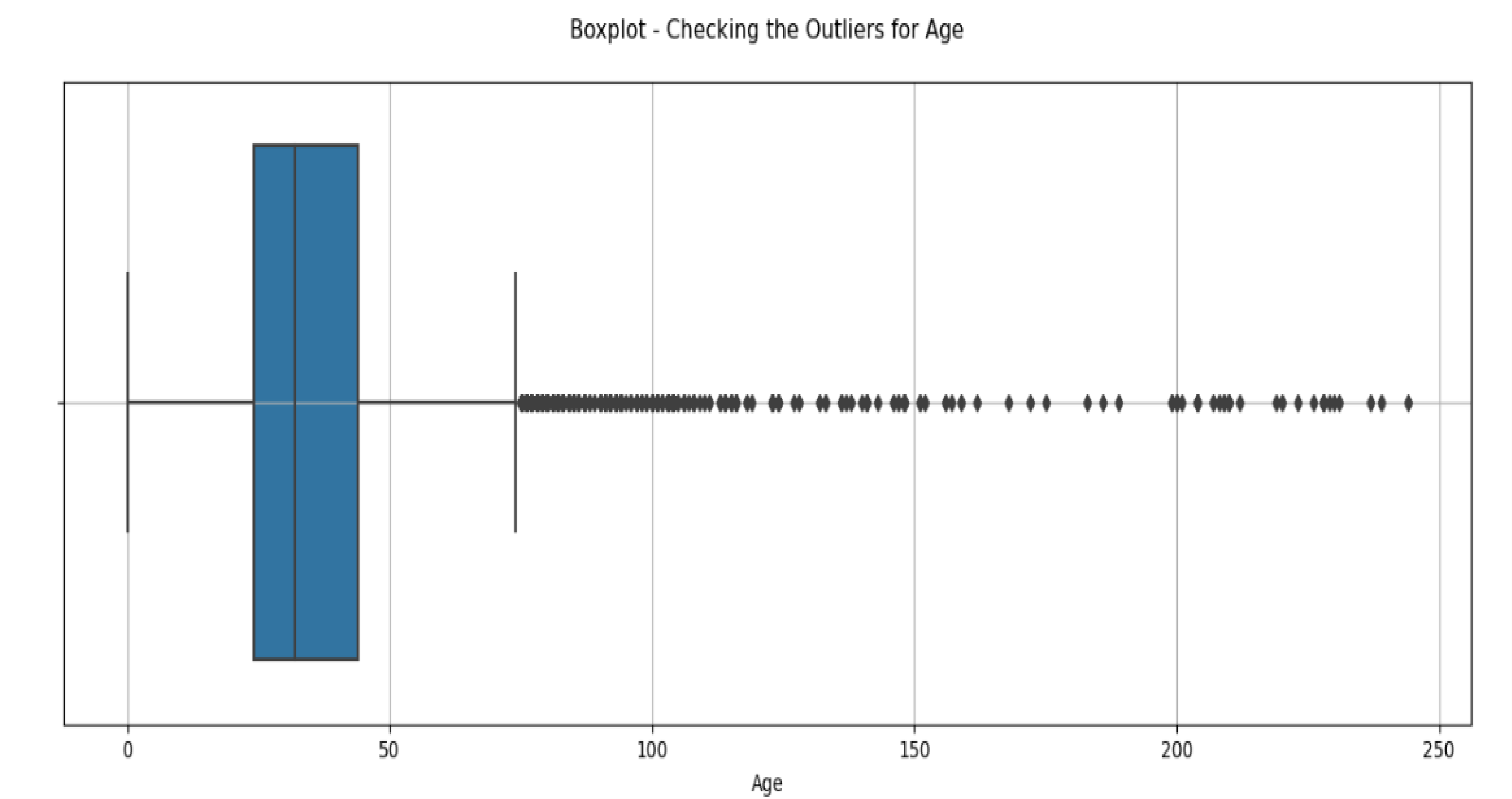


```
In [83]: # Reviewing the column after making above changes in data
         user.head()
```

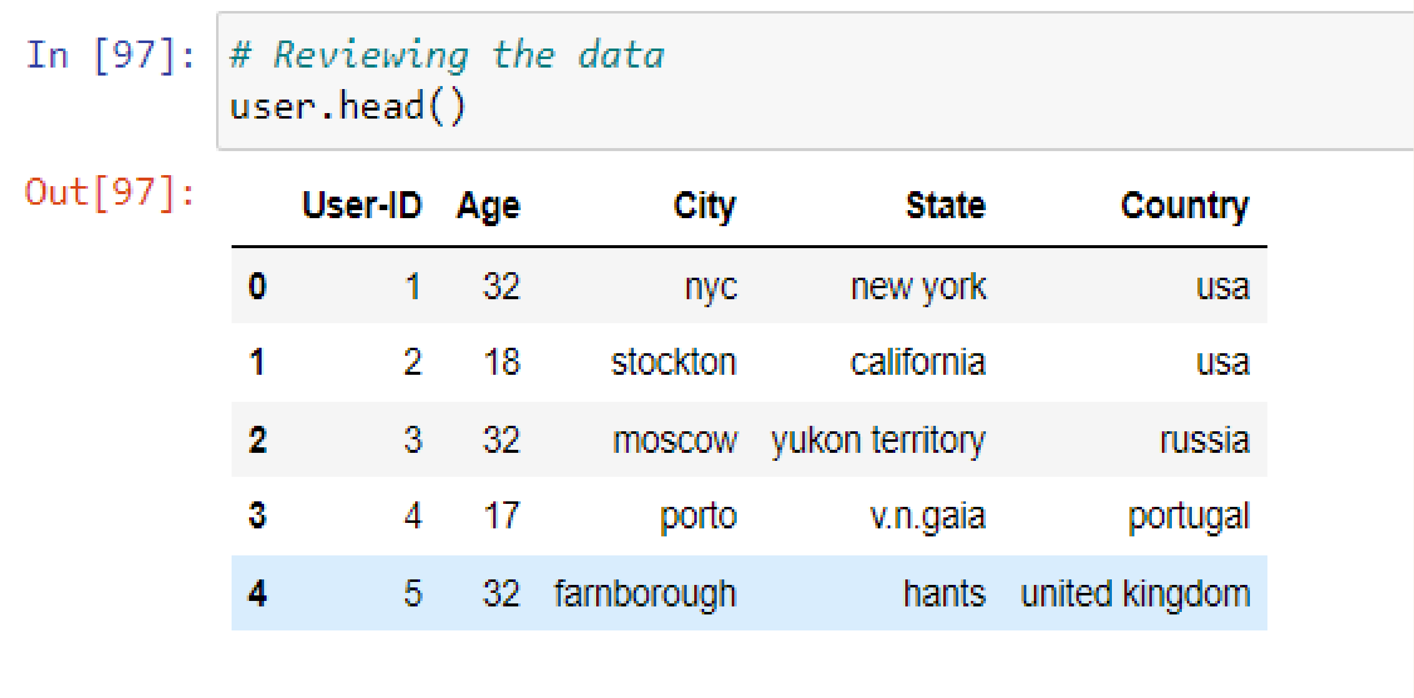
Out[83]:

	User-ID	Age	City	State	Country
0	1	NaN	nyc	new york	usa
1	2	18.0	stockton	california	usa
2	3	NaN	moscow	yukon territory	russia
3	4	17.0	porto	v.n.gaia	portugal
4	5	NaN	farnborough	hants	united kingdom


Histogram - Checking the Distribution of Age
Frequency
80000
70000
60000
50000
40000
30000
20000
10000
0
0
50
100
150
200
250
Age


Boxplot - Checking the Outliers for Age
0
50
100
150
200
250
Age

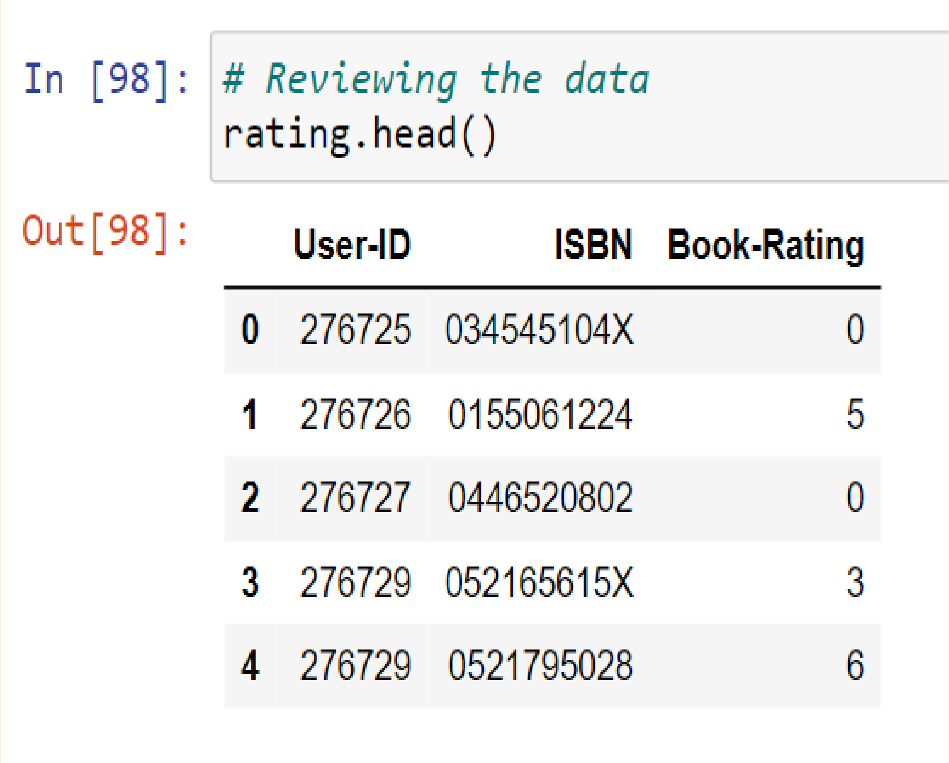
Final Users Dataset after the modification



```
# Reviewing the data
user.head()
```

	User-ID	Age	City	State	Country
0	1	32	nyc	new york	usa
1	2	18	stockton	california	usa
2	3	32	moscow	yukon territory	russia
3	4	17	porto	v.n.gaia	portugal
4	5	32	farnborough	hants	united kingdom

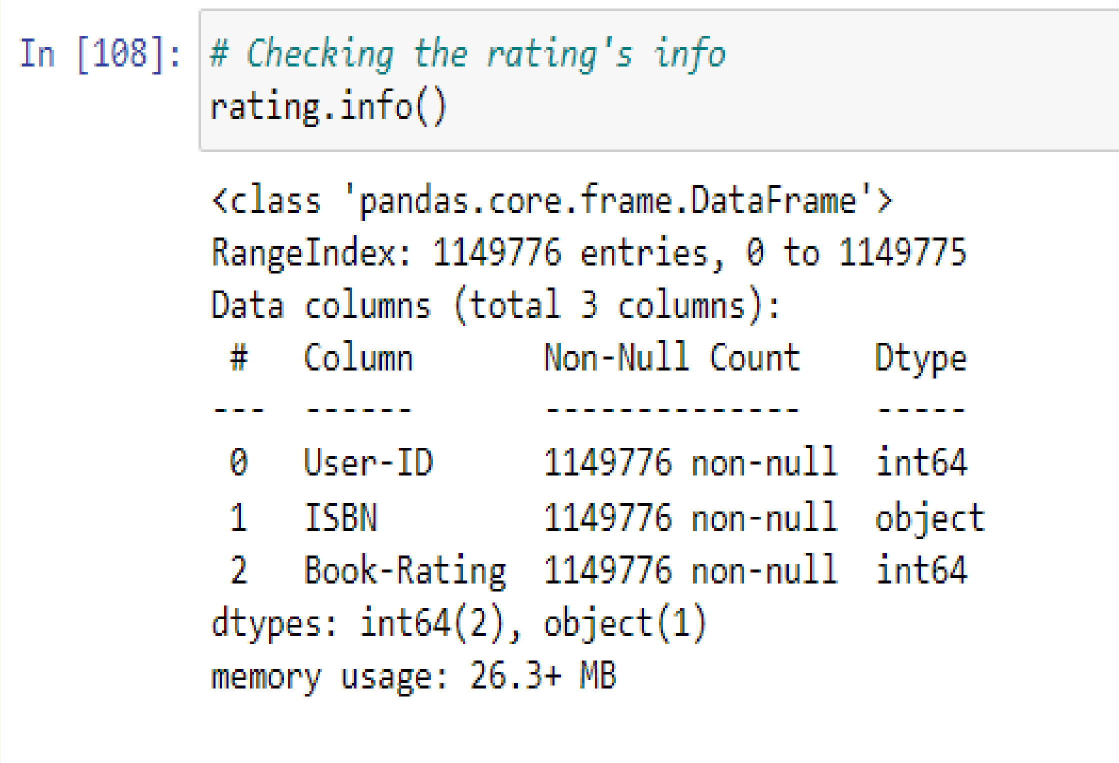
In the "Ratings" dataset, there are no empty or repeated values to fix, and there are no strange values. However, we noticed that there are quite a few zero ratings, which likely represent users who haven't rated any books yet.



```
In [98]: # Reviewing the data
         rating.head()
```

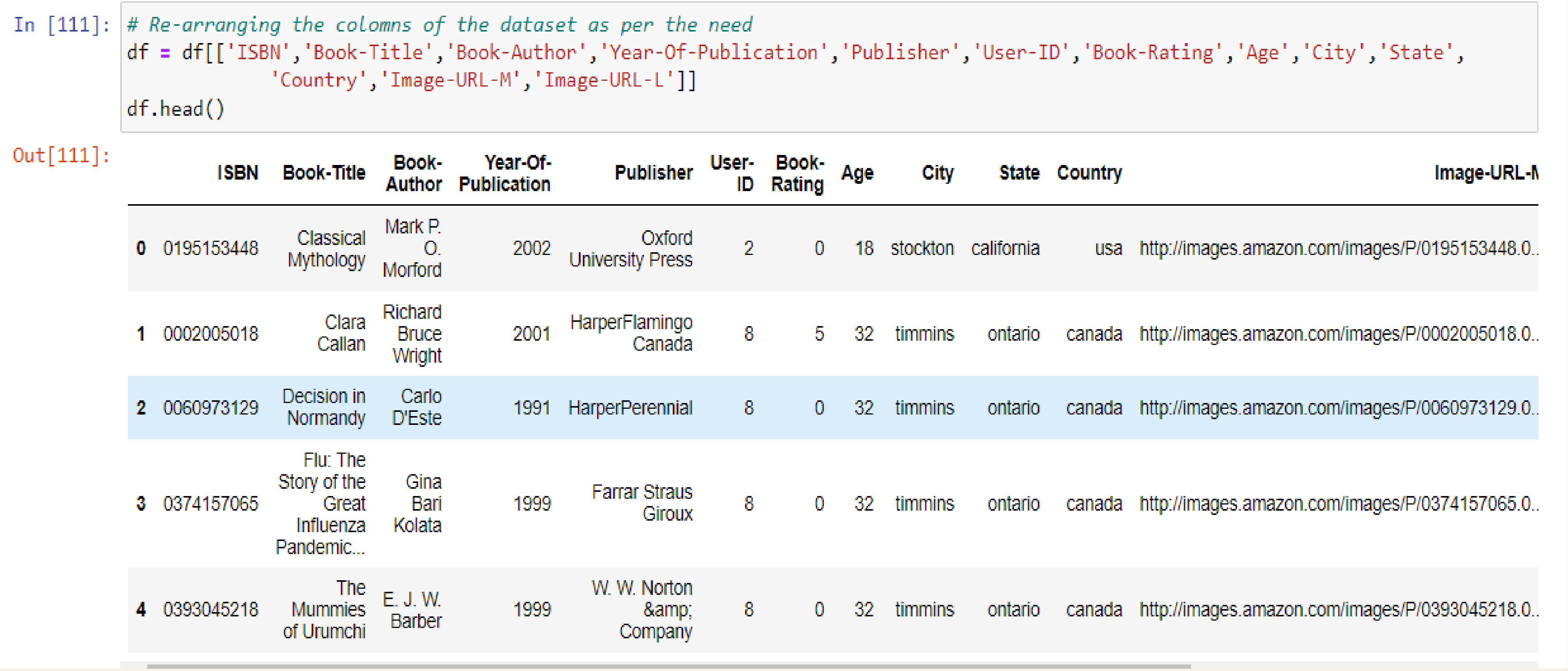
Out[98]:

	User-ID	ISBN	Book-Rating
0	276725	034545104X	0
1	276726	0155061224	5
2	276727	0446520802	0
3	276729	052165615X	3
4	276729	0521795028	6



```
In [108]: # Checking the rating's info
          rating.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1149776 entries, 0 to 1149775
Data columns (total 3 columns):
 #   Column       Non-Null Count    Dtype 
---  ------       --------------    ----- 
 0   User-ID      1149776 non-null  int64 
 1   ISBN         1149776 non-null  object
 2   Book-Rating  1149776 non-null  int64 
dtypes: int64(2), object(1)
memory usage: 26.3+ MB
```

To move forward, we plan to merge these three datasets into one, similar to how we combine data in SQL. This merged dataset will make it simpler for us to analyze, visualize, and build models. It'll help us draw meaningful insights and create better visualizations.

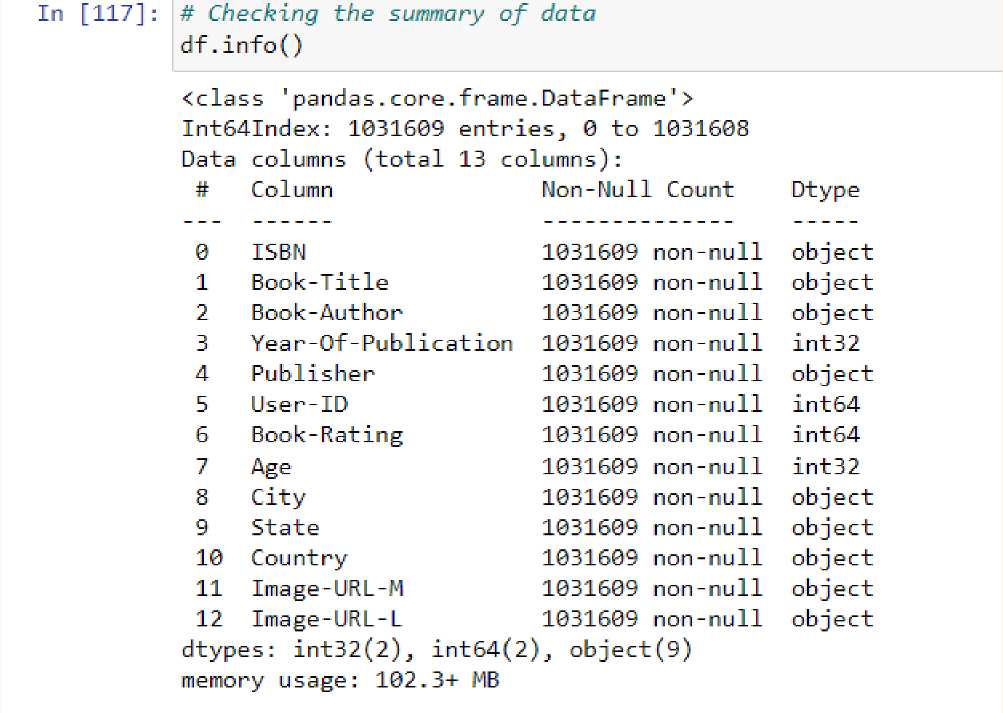


```
In [111]: # Re-arranging the coloms of the dataset as per the need
          df = df[['ISBN','Book-Title','Book-Author','Year-Of-Publication','Publisher','User-ID','Book-Rating','Age','City','State',
                   'Country','Image-URL-M','Image-URL-L']]
          df.head()
```

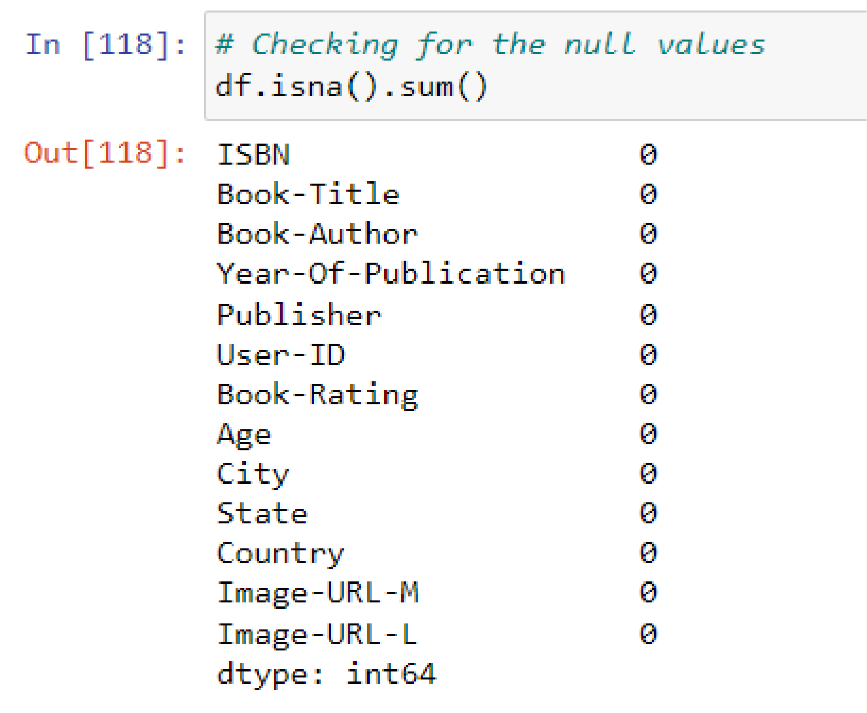
Out[111]:

	ISBN	Book-Title	Book-Author	Year-Of-Publication	Publisher	User-ID	Book-Rating	Age	City	State	Country	Image-URL-M
0	0195153448	Classical Mythology	Mark P. O. Morford	2002	Oxford University Press	2	0	18	stockton	california	usa	http://images.amazon.com/images/P/0195153448.0...
1	0002005018	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	8	5	32	timmins	ontario	canada	http://images.amazon.com/images/P/0002005018.0...
2	0060973129	Decision in Normandy	Carlo D'Este	1991	HarperPerennial	8	0	32	timmins	ontario	canada	http://images.amazon.com/images/P/0060973129.0...
3	0374157065	Flu: The Story of the Great Influenza Pandemic...	Gina Bari Kolata	1999	Farrar Straus Giroux	8	0	32	timmins	ontario	canada	http://images.amazon.com/images/P/0374157065.0...
4	0393045218	The Mummies of Urumchi	E. J. W. Barber	1999	W. W. Norton &amp; Company	8	0	32	timmins	ontario	canada	http://images.amazon.com/images/P/0393045218.0...

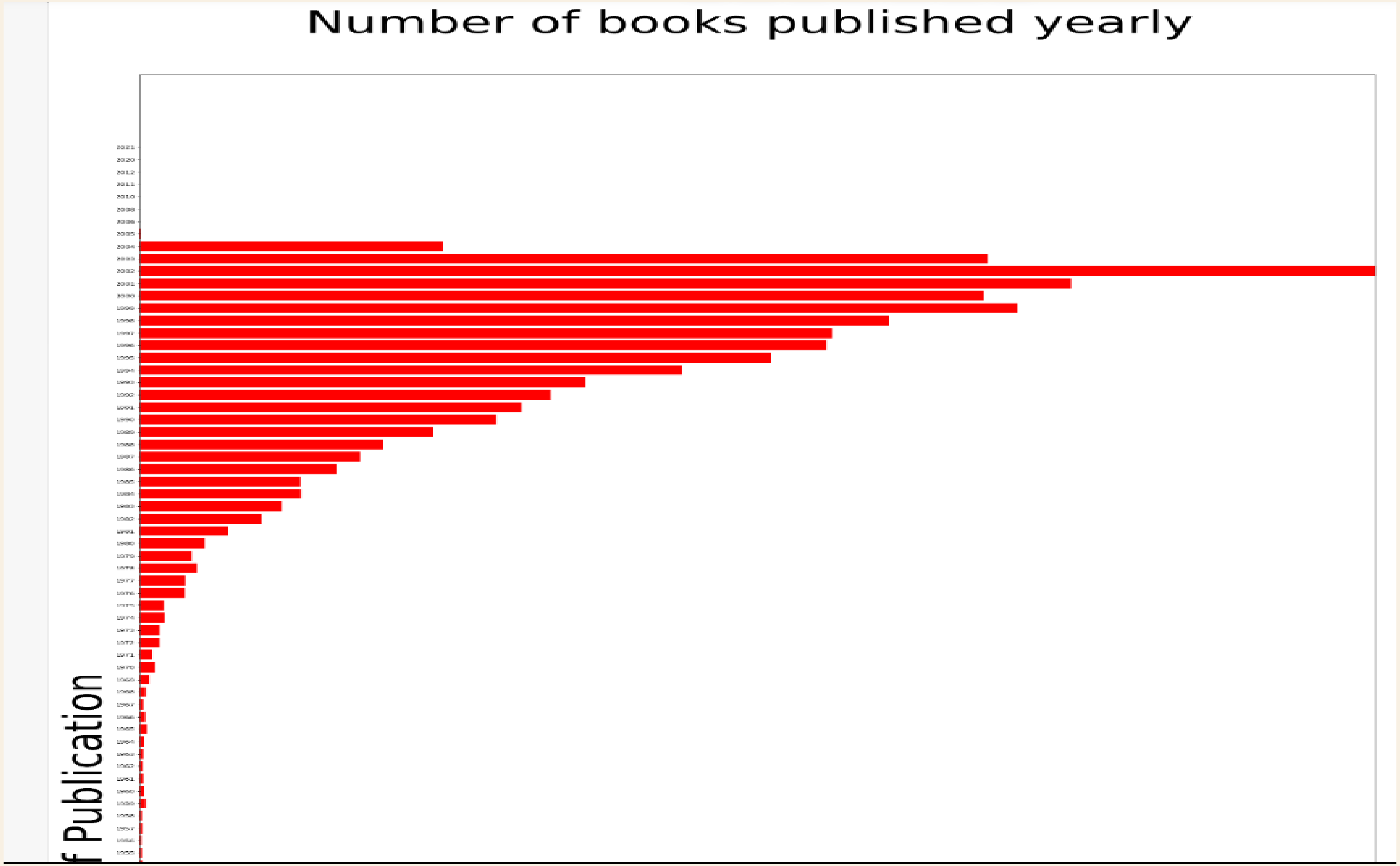
We examined the merged dataset's information, including details about any missing values.

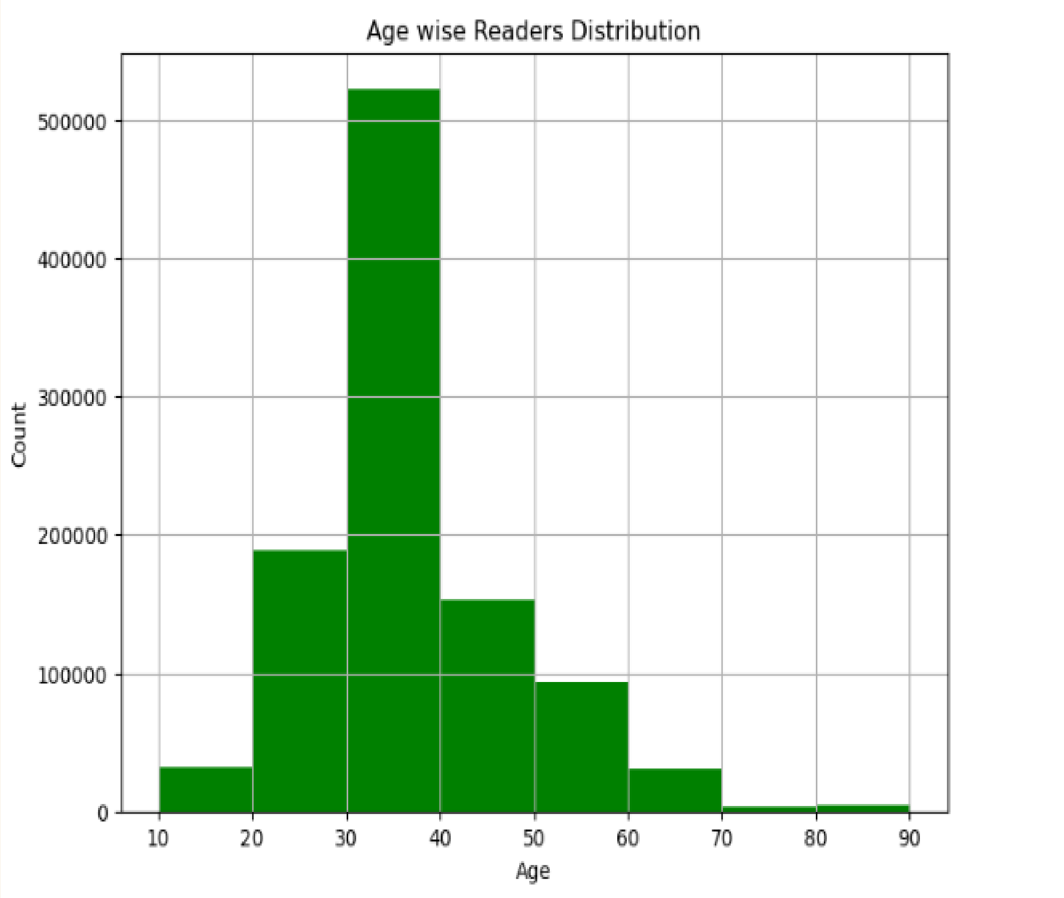


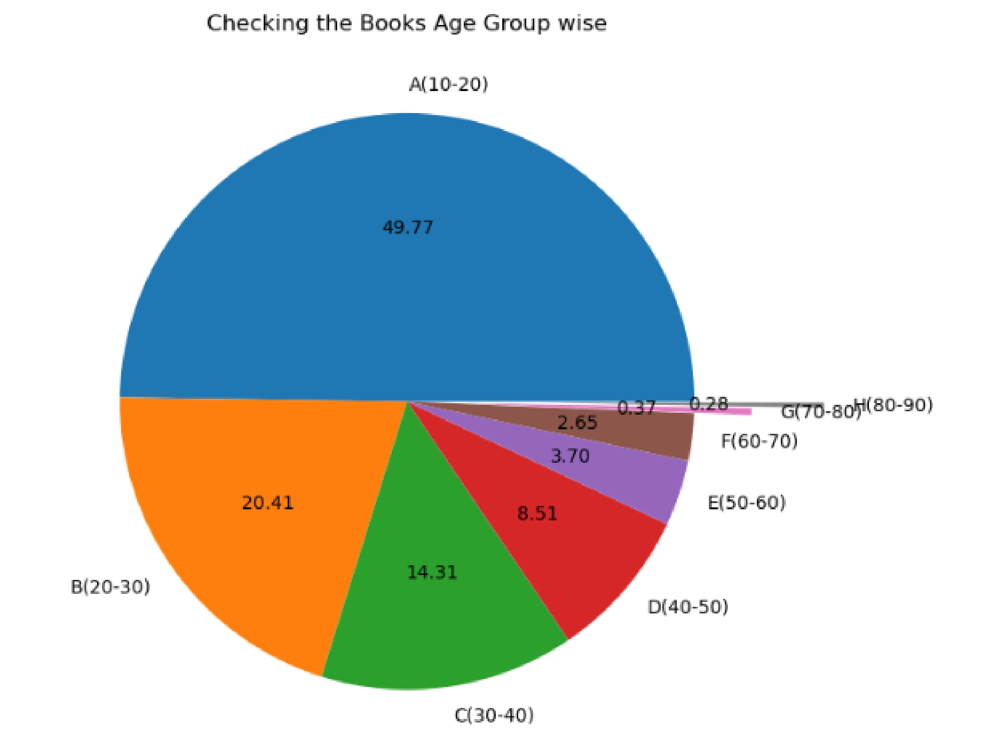
```
In [117]: # Checking the summary of data
          df.info()

          <class 'pandas.core.frame.DataFrame'>
          Int64Index: 1031609 entries, 0 to 1031608
          Data columns (total 13 columns):
           #   Column               Non-Null Count    Dtype 
          ---  ------               --------------    ----- 
           0   ISBN                 1031609 non-null  object
           1   Book-Title           1031609 non-null  object
           2   Book-Author          1031609 non-null  object
           3   Year-Of-Publication  1031609 non-null  int32 
           4   Publisher            1031609 non-null  object
           5   User-ID              1031609 non-null  int64 
           6   Book-Rating          1031609 non-null  int64 
           7   Age                  1031609 non-null  int32 
           8   City                 1031609 non-null  object
           9   State                1031609 non-null  object
           10  Country              1031609 non-null  object
           11  Image-URL-M          1031609 non-null  object
           12  Image-URL-L          1031609 non-null  object
          dtypes: int32(2), int64(2), object(9)
          memory usage: 102.3+ MB
```

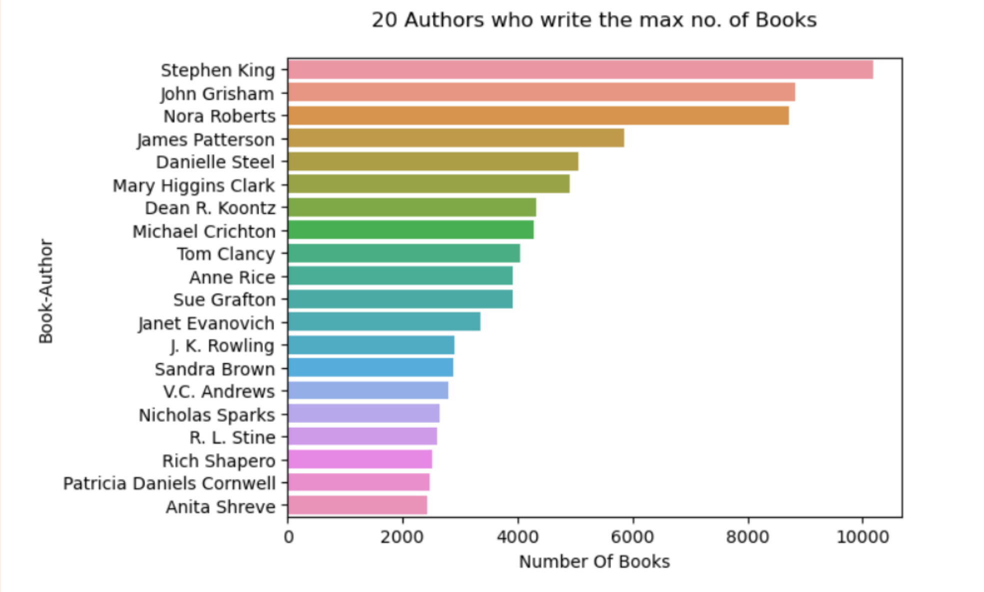


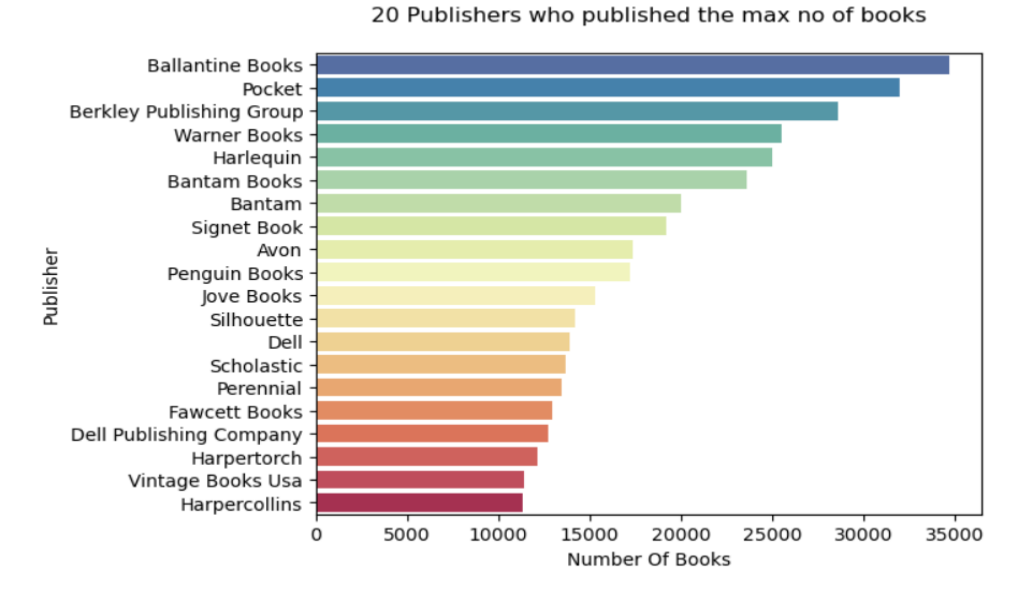
```
In [118]: # Checking for the null values
          df.isna().sum()

Out[118]: ISBN                   0
          Book-Title             0
          Book-Author            0
          Year-Of-Publication    0
          Publisher              0
          User-ID                0
          Book-Rating            0
          Age                    0
          City                   0
          State                  0
          Country                0
          Image-URL-M            0
          Image-URL-L            0
          dtype: int64
```

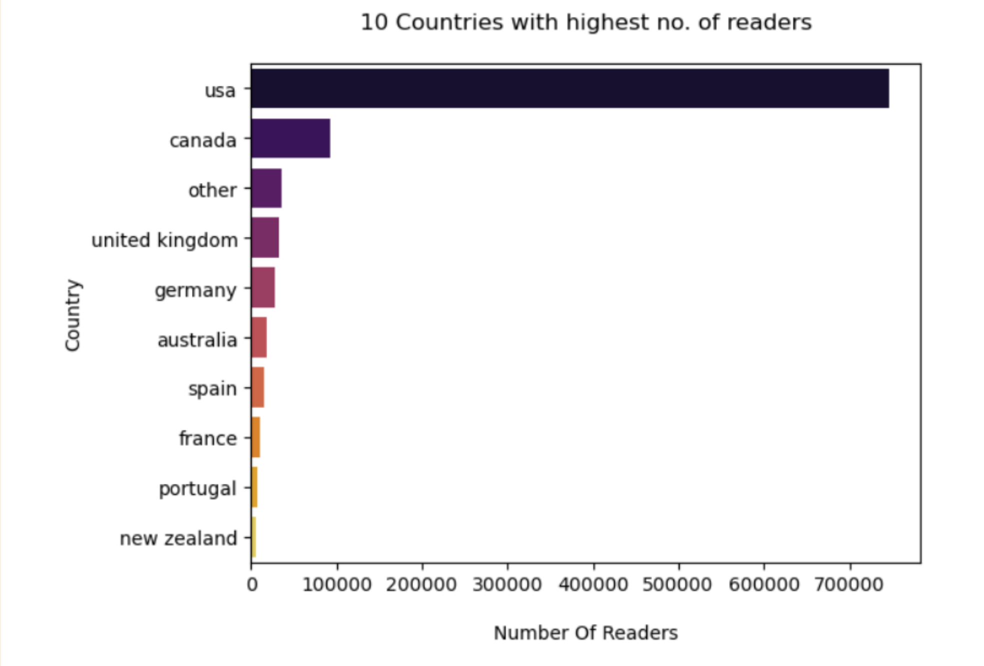

Number of books published yearly
f Publication
2021
2020
2012
2011
2010
2008
2006
2005
2004
2003
2002
2001
2000
1999
1998
1997
1996
1995
1994
1993
1992
1991
1990
1989
1988
1987
1986
1985
1984
1983
1982
1981
1980
1979
1978
1977
1976
1975
1974
1973
1972
1971
1970
1969
1968
1967
1966
1965
1964
1963
1962
1961
1960
1959
1958
1957
1956
1955

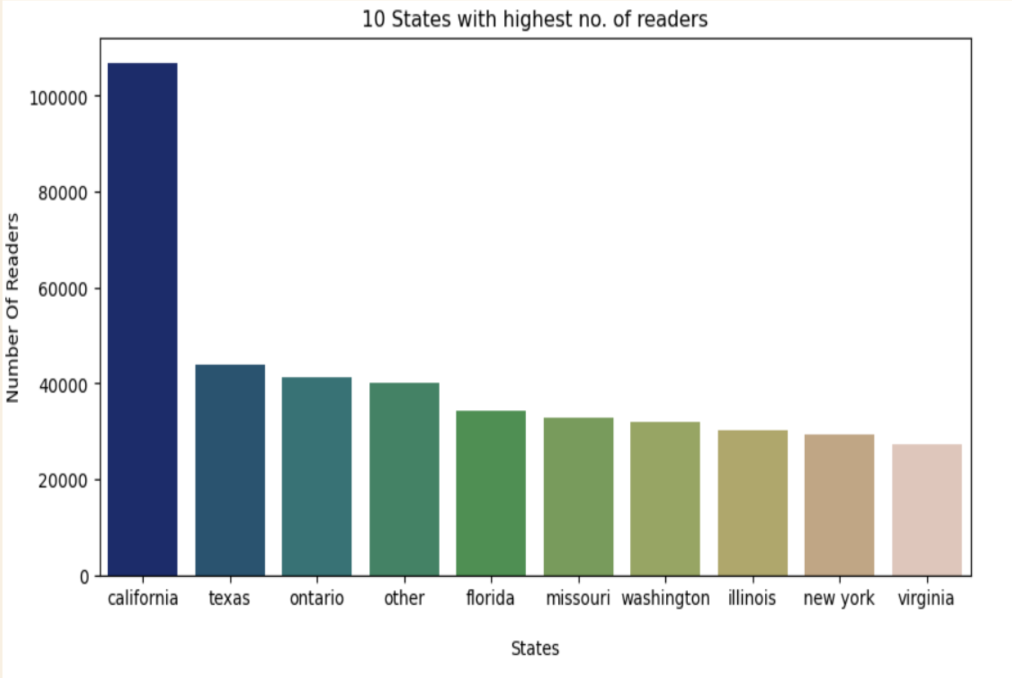

Age wise Readers Distribution
Count
500000
400000
300000
200000
100000
0
10
20
30
40
50
60
70
80
90
Age

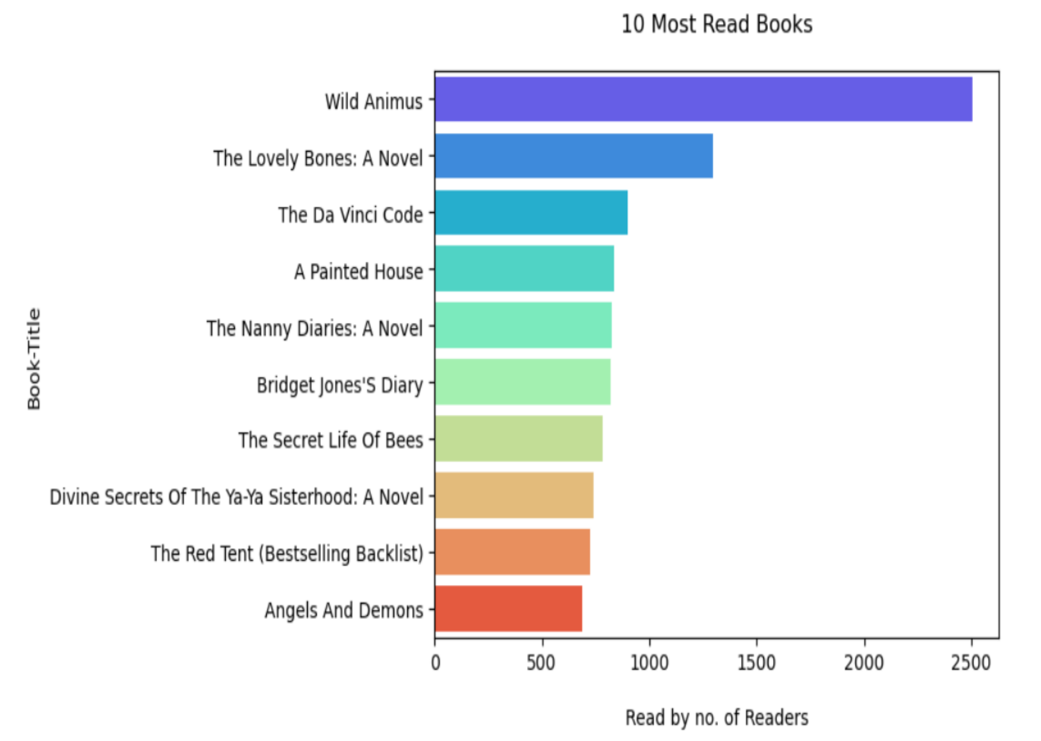

Checking the Books Age Group wise
A(10-20)
49.77
B(20-30)
20.41
C(30-40)
14.31
D(40-50)
8.51
E(50-60)
3.70
F(60-70)
2.65
G(70-80)
0.37
H(80-90)
0.28

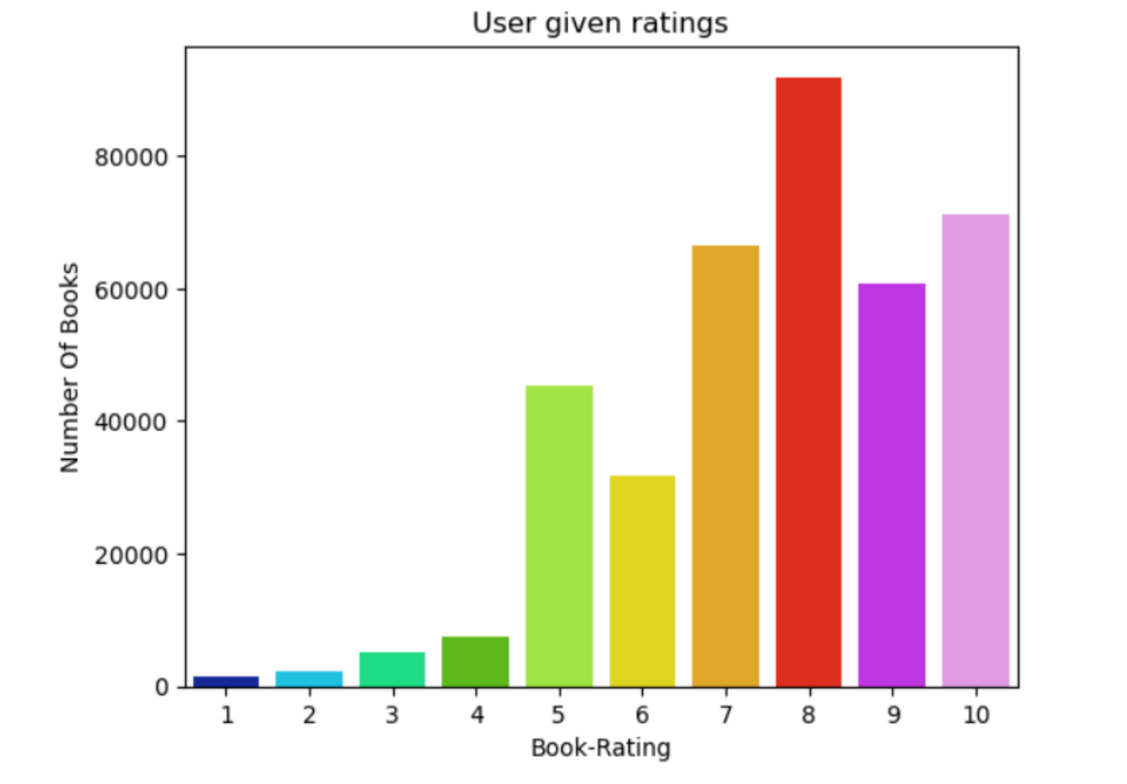

20 Authors who write the max no. of Books
Book-Author
Stephen King
John Grisham
Nora Roberts
James Patterson
Danielle Steel
Mary Higgins Clark
Dean R. Koontz
Michael Crichton
Tom Clancy
Anne Rice
Sue Grafton
Janet Evanovich
J. K. Rowling
Sandra Brown
V.C. Andrews
Nicholas Sparks
R. L. Stine
Rich Shapero
Patricia Daniels Cornwell
Anita Shreve
0
2000
4000
6000
8000
10000
Number Of Books

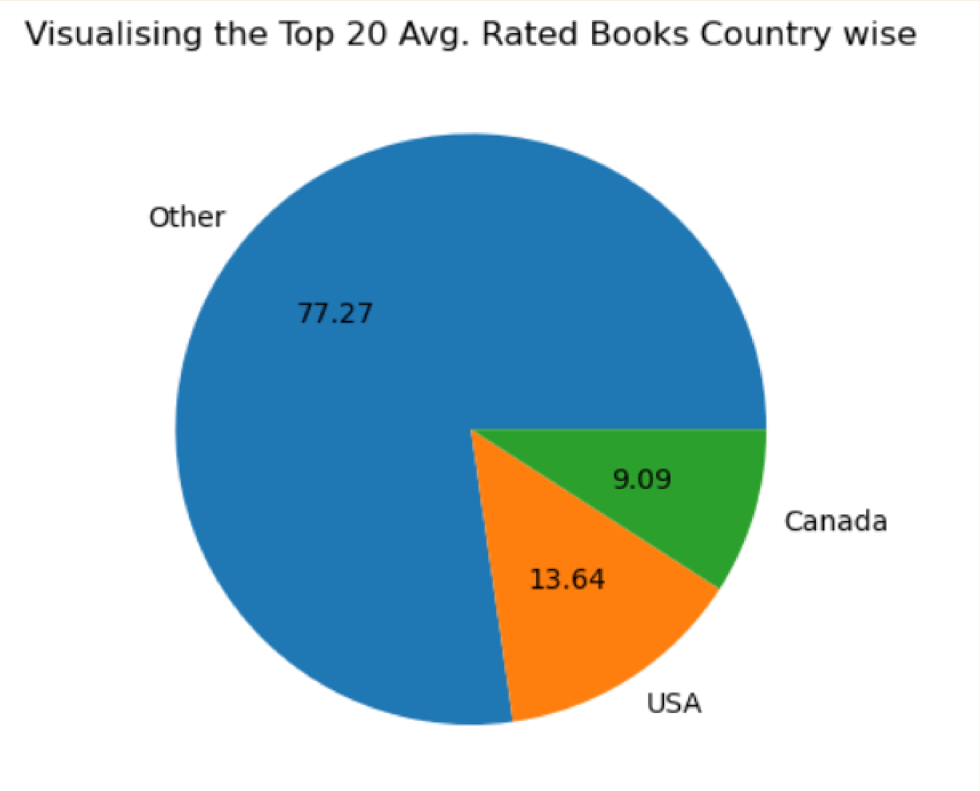

20 Publishers who published the max no of books
Publisher
Ballantine Books
Pocket
Berkley Publishing Group
Warner Books
Harlequin
Bantam Books
Bantam
Signet Book
Avon
Penguin Books
Jove Books
Silhouette
Dell
Scholastic
Perennial
Fawcett Books
Dell Publishing Company
Harpertorch
Vintage Books Usa
Harpercollins
0
5000
10000
15000
20000
25000
30000
35000
Number Of Books

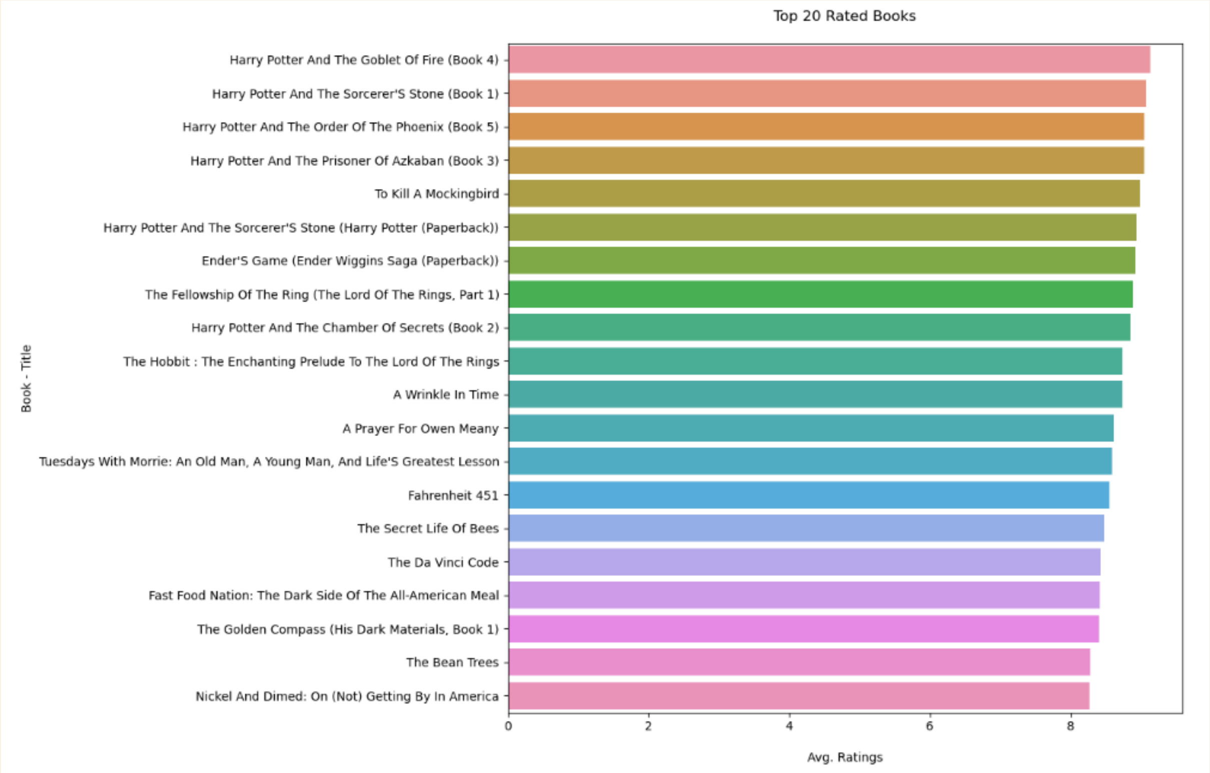

10 Countries with highest no. of readers
Country
usa
canada
other
united kingdom
germany
australia
spain
france
portugal
new zealand
0
100000
200000
300000
400000
500000
600000
700000
Number Of Readers


10 States with highest no. of readers
Number Of Readers
100000
80000
60000
40000
20000
0
california
texas
ontario
other
florida
missouri
washington
illinois
new york
virginia
States

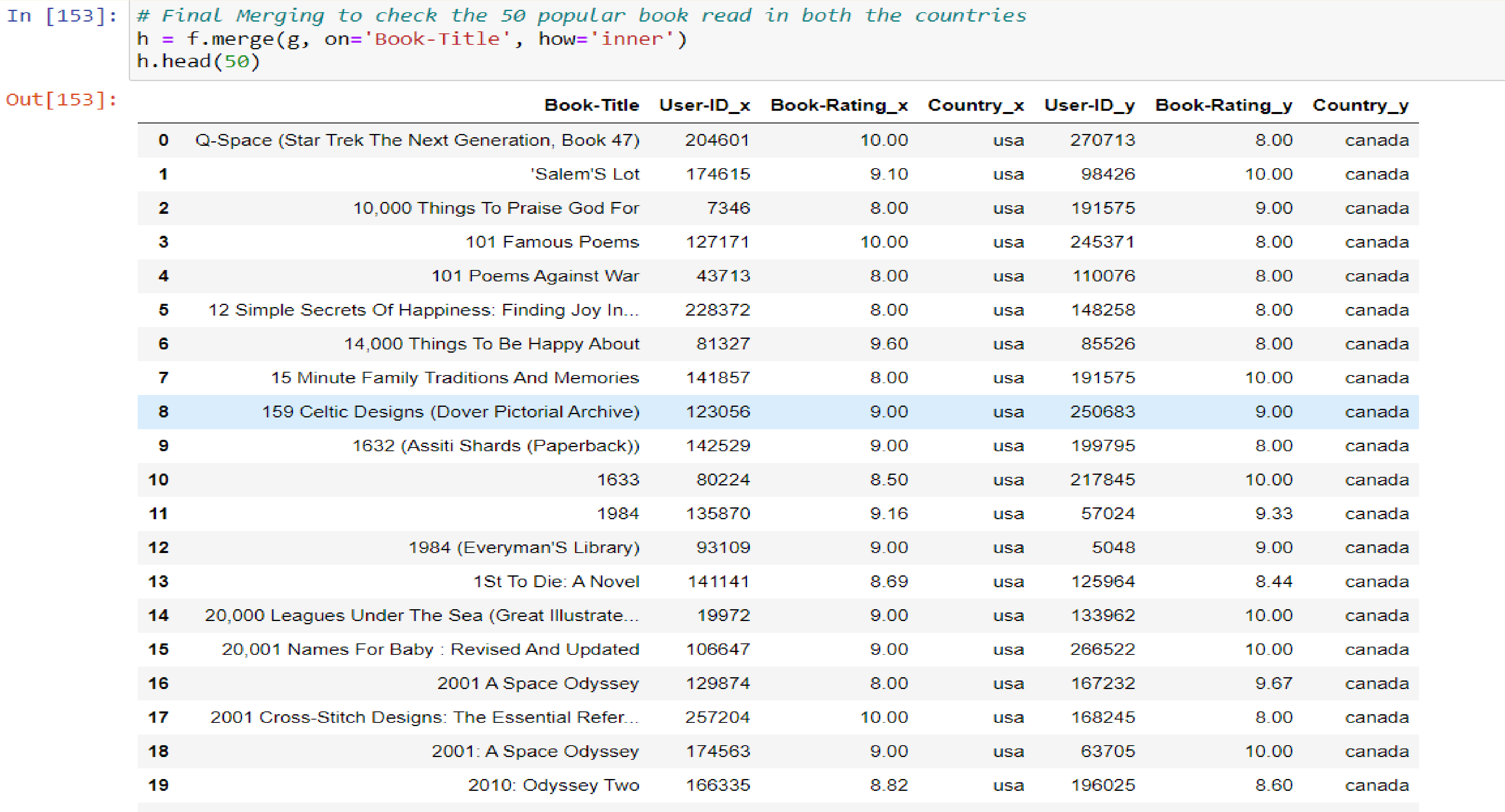

10 Most Read Books
Book-Title
Wild Animus
The Lovely Bones: A Novel
The Da Vinci Code
A Painted House
The Nanny Diaries: A Novel
Bridget Jones'S Diary
The Secret Life Of Bees
Divine Secrets Of The Ya-Ya Sisterhood: A Novel
The Red Tent (Bestselling Backlist)
Angels And Demons
0
500
1000
1500
2000
2500
Read by no. of Readers


User given ratings
Number Of Books
0
20000
40000
60000
80000
1
2
3
4
5
6
7
8
9
10
Book-Rating


Visualising the Top 20 Avg. Rated Books Country wise
Other
77.27
9.09
Canada
13.64
USA


Top 20 Rated Books
Harry Potter And The Goblet Of Fire (Book 4)
Harry Potter And The Sorcerer'S Stone (Book 1)
Harry Potter And The Order Of The Phoenix (Book 5)
Harry Potter And The Prisoner Of Azkaban (Book 3)
To Kill A Mockingbird
Harry Potter And The Sorcerer'S Stone (Harry Potter (Paperback))
Ender'S Game (Ender Wiggins Saga (Paperback))
The Fellowship Of The Ring (The Lord Of The Rings, Part 1)
Harry Potter And The Chamber Of Secrets (Book 2)
The Hobbit : The Enchanting Prelude To The Lord Of The Rings
A Wrinkle In Time
A Prayer For Owen Meany
Tuesdays With Morrie: An Old Man, A Young Man, And Life'S Greatest Lesson
Fahrenheit 451
The Secret Life Of Bees
The Da Vinci Code
Fast Food Nation: The Dark Side Of The All-American Meal
The Golden Compass (His Dark Materials, Book 1)
The Bean Trees
Nickel And Dimed: On (Not) Getting By In America
Book - Title
0
2
4
6
8
Avg. Ratings

The final dataset would be :



```
In [153]: # Final Merging to check the 50 popular book read in both the countries
          h = f.merge(g, on='Book-Title', how='inner')
          h.head(50)
```

Out[153]:

	Book-Title	User-ID_x	Book-Rating_x	Country_x	User-ID_y	Book-Rating_y	Country_y
0	Q-Space (Star Trek The Next Generation, Book 47)	204601	10.00	usa	270713	8.00	canada
1	'Salem'S Lot	174615	9.10	usa	98426	10.00	canada
2	10,000 Things To Praise God For	7346	8.00	usa	191575	9.00	canada
3	101 Famous Poems	127171	10.00	usa	245371	8.00	canada
4	101 Poems Against War	43713	8.00	usa	110076	8.00	canada
5	12 Simple Secrets Of Happiness: Finding Joy In...	228372	8.00	usa	148258	8.00	canada
6	14,000 Things To Be Happy About	81327	9.60	usa	85526	8.00	canada
7	15 Minute Family Traditions And Memories	141857	8.00	usa	191575	10.00	canada
8	159 Celtic Designs (Dover Pictorial Archive)	123056	9.00	usa	250683	9.00	canada
9	1632 (Assiti Shards (Paperback))	142529	9.00	usa	199795	8.00	canada
10	1633	80224	8.50	usa	217845	10.00	canada
11	1984	135870	9.16	usa	57024	9.33	canada
12	1984 (Everyman'S Library)	93109	9.00	usa	5048	9.00	canada
13	1St To Die: A Novel	141141	8.69	usa	125964	8.44	canada
14	20,000 Leagues Under The Sea (Great Illustrate...	19972	9.00	usa	133962	10.00	canada
15	20,001 Names For Baby : Revised And Updated	106647	9.00	usa	266522	10.00	canada
16	2001 A Space Odyssey	129874	8.00	usa	167232	9.67	canada
17	2001 Cross-Stitch Designs: The Essential Refer...	257204	10.00	usa	168245	8.00	canada
18	2001: A Space Odyssey	174563	9.00	usa	63705	10.00	canada
19	2010: Odyssey Two	166335	8.82	usa	196025	8.60	canada

BUILDING RECOMMENDATION MODELS

BUILDING RECOMMENDATION MODEL

We used

01. Popularity based

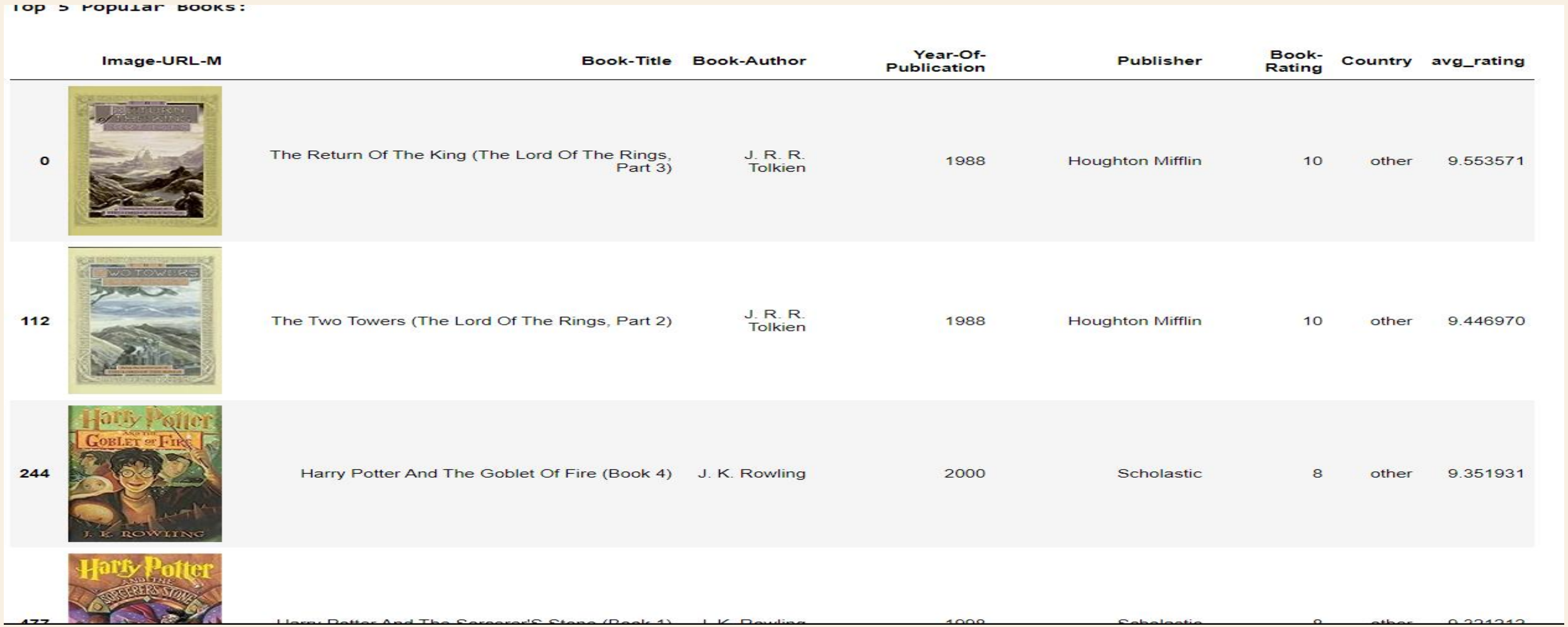
02. Collaborative Filtering based

POPULARITY BASED

A popularity-based recommendation system relies on current trends. It suggests items that are currently popular, such as products that many new users are buying. For instance, if a product is frequently purchased by new users, the system is likely to recommend it to a newly registered user.

TOP POPULAR BOOKS

We're creating a recommendation system that relies on the highest average ratings given by users. This system considers readers who have rated at least 100 books and ensures that a book is rated by at least 50 different users to be considered for recommendations.

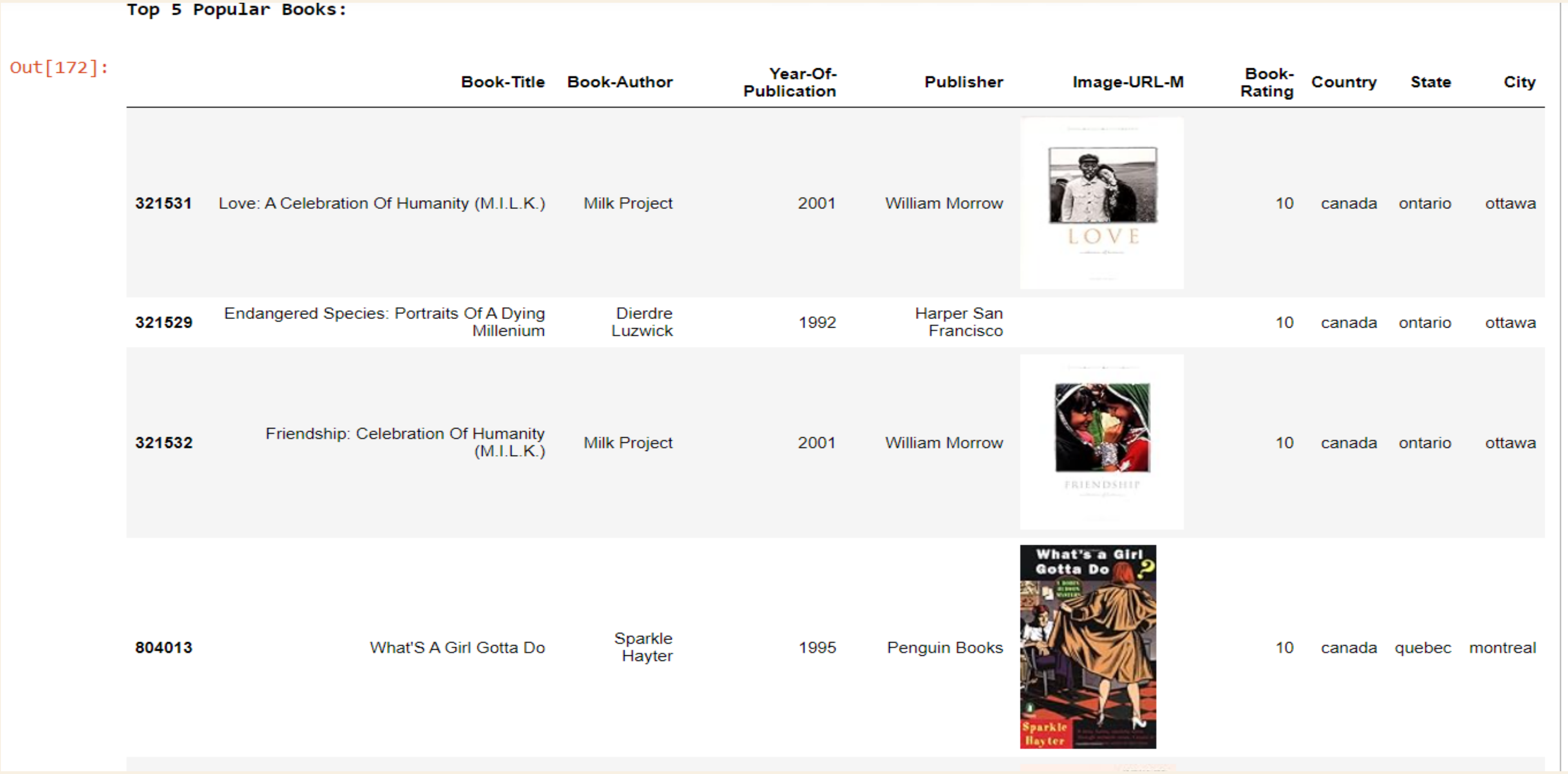


Top 5 Popular Books:

	Image-URL-M	Book-Title	Book-Author	Year-Of-Publication	Publisher	Book-Rating	Country	avg_rating
0		The Return Of The King (The Lord Of The Rings, Part 3)	J. R. R. Tolkien	1988	Houghton Mifflin	10	other	9.553571
112		The Two Towers (The Lord Of The Rings, Part 2)	J. R. R. Tolkien	1988	Houghton Mifflin	10	other	9.446970
244		Harry Potter And The Goblet Of Fire (Book 4)	J. K. Rowling	2000	Scholastic	8	other	9.351931
477		Harry Potter And The Sorcerer'S Stone (Book 1)	J. K. Rowling	1998	Scholastic	8	other	9.331313

TOP POPULAR BOOKS BASED ON LOCATION

We're creating a recommendation system that focuses on the top-rated books read by users in a specific location. This recommendation system takes into account readers who have read and given a score higher than 7 to a book within that particular location..



```
Top 5 Popular Books:
Out[172]:
```

	Book-Title	Book-Author	Year-Of-Publication	Publisher	Image-URL-M	Book-Rating	Country	State	City
321531	Love: A Celebration Of Humanity (M.I.L.K.)	Milk Project	2001	William Morrow	LOVE	10	canada	ontario	ottawa
321529	Endangered Species: Portraits Of A Dying Millenium	Dierdre Luzwick	1992	Harper San Francisco		10	canada	ontario	ottawa
321532	Friendship: Celebration Of Humanity (M.I.L.K.)	Milk Project	2001	William Morrow	FRIENDSHIP	10	canada	ontario	ottawa
804013	What'S A Girl Gotta Do	Sparkle Hayter	1995	Penguin Books	What's a Girl Gotta Do? Sparkle Hayter	10	canada	quebec	montreal

COLLABORATIVE FILTERING

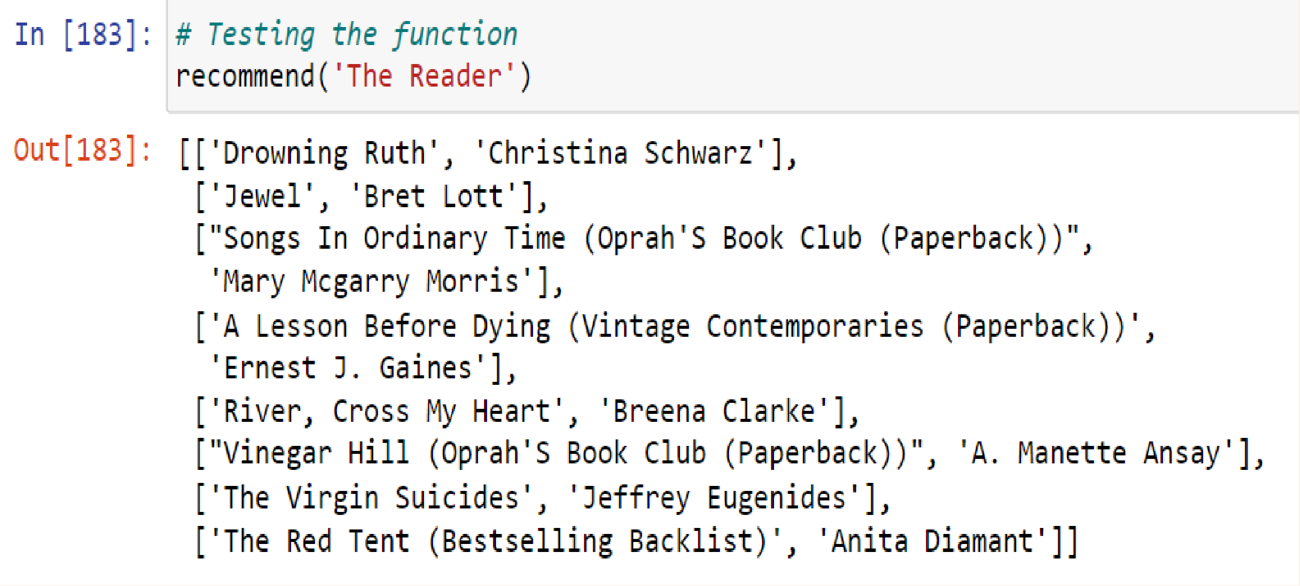
Collaborative filtering is a technique commonly used to build personalized recommendations. It is based on gathering and analysing data on user’s behaviour.

In Collaborative filtering, we tend to find similar users /items and recommend what similar users like. This includes the user’s online activities and predicting what they will like based on the similarity with other users.

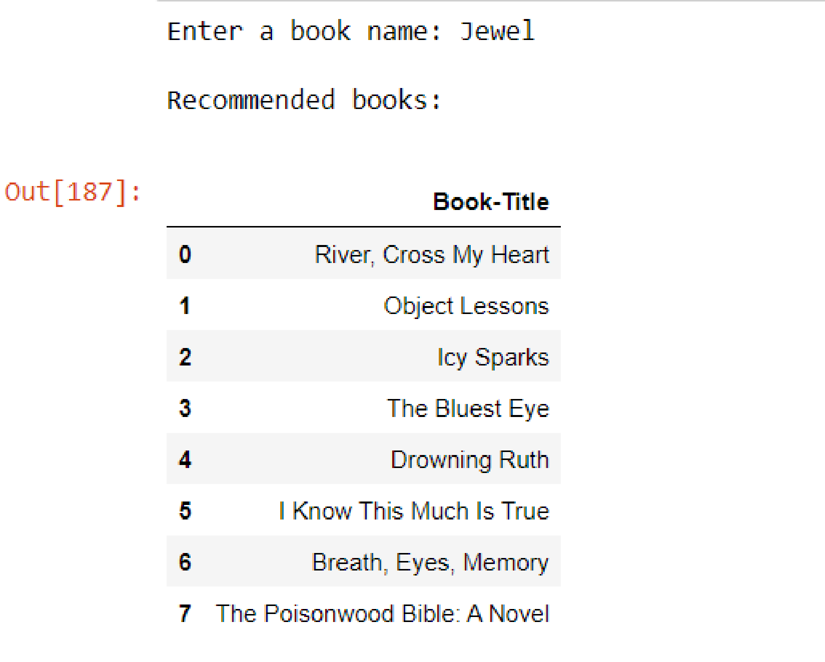
ITEM-BASED FILTERING

We've created a recommendation system using a cosine similarity matrix to measure item similarity. This system focuses on top users who've read at least 150 books and books that have been rated by at least 50 different users. Another model employs nearest neighbors and considers books with more than 50 ratings.

Out of the analysis, we identified just two books that were common in the search results. The Cosine Similarity Matrix demonstrated higher accuracy in contrast to the Nearest Neighbors model's item-based filtering. After reviewing both models for book recommendations based on genres, we've decided to proceed with constructing a user-user collaborative filtering model for our dataset.



```
In [183]: # Testing the function
          recommend('The Reader')

Out[183]: [['Drowning Ruth', 'Christina Schwarz'],
           ['Jewel', 'Bret Lott'],
           ["Songs In Ordinary Time (Oprah'S Book Club (Paperback))",
            'Mary Mcgarry Morris'],
           ['A Lesson Before Dying (Vintage Contemporaries (Paperback))',
            'Ernest J. Gaines'],
           ['River, Cross My Heart', 'Breena Clarke'],
           ["Vinegar Hill (Oprah'S Book Club (Paperback))", 'A. Manette Ansay'],
           ['The Virgin Suicides', 'Jeffrey Eugenides'],
           ['The Red Tent (Bestselling Backlist)', 'Anita Diamant']]
```



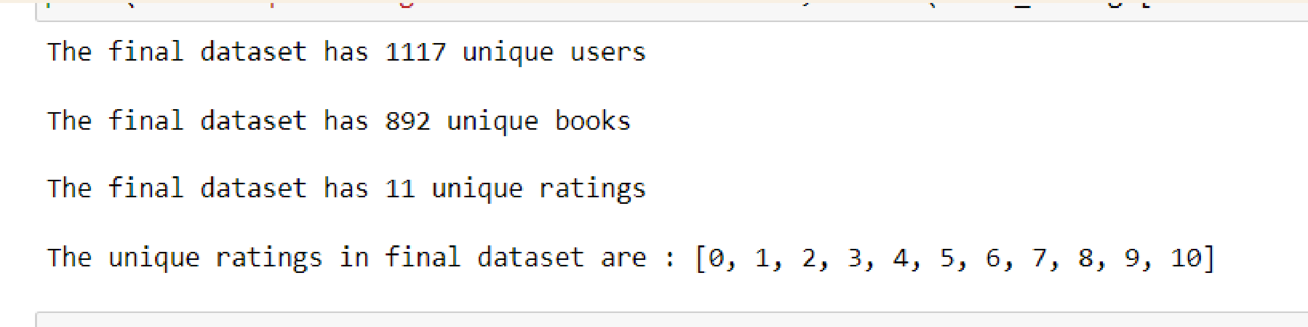
```
          Enter a book name: Jewel

          Recommended books:

Out[187]:
```

	Book-Title
0	River, Cross My Heart
1	Object Lessons
2	Icy Sparks
3	The Bluest Eye
4	Drowning Ruth
5	I Know This Much Is True
6	Breath, Eyes, Memory
7	The Poisonwood Bible: A Novel

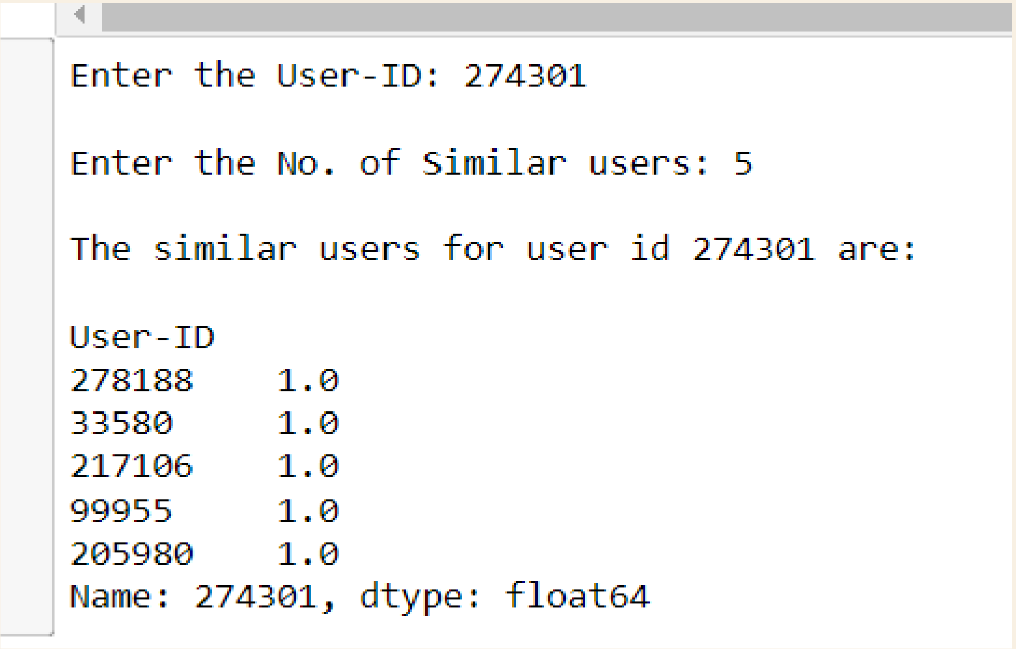
USER – USER BASED FILTERING

User-User collaborative filtering is a way to recommend things by finding people who are similar to you in terms of what they like or enjoy. We used Pearson's Correlation Similarity Matrix to do this.

In the user similarity matrix, values range from -1 to 1, indicating how similar users are in their book preferences. We've set a threshold of 0.3, meaning users need a correlation of at least 0.3 to be considered similar. After sorting the values, we find the most similar users to user ID 254. To do this, we exclude user ID 254 from the list and decide how many similar users we want. We then identify books that these similar users have read, excluding the ones user 254 has already read. We only keep the books similar users have read and remove the ones user 254 has already read from the user-item matrix.



```
The final dataset has 1117 unique users

The final dataset has 892 unique books

The final dataset has 11 unique ratings

The unique ratings in final dataset are : [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

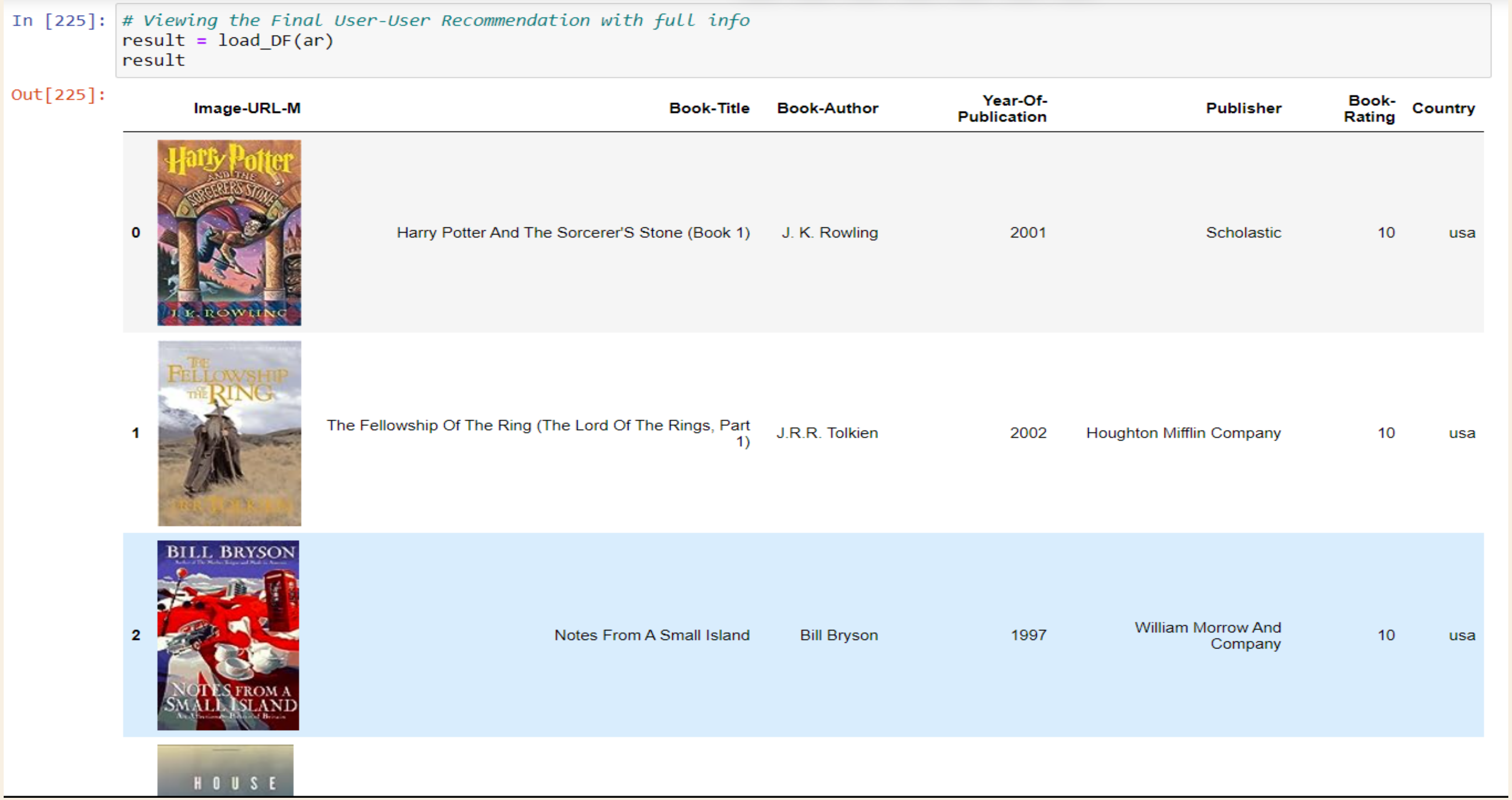


```
Enter the User-ID: 274301

Enter the No. of Similar users: 5

The similar users for user id 274301 are:

User-ID
278188    1.0
33580     1.0
217106    1.0
99955     1.0
205980    1.0
Name: 274301, dtype: float64
```

From analysis, we can say that there were at least 2 books found to be common among similar users and target users.

Now, we will remove the books already read by target user from similar user books dataset.

Final user-user Recommendation

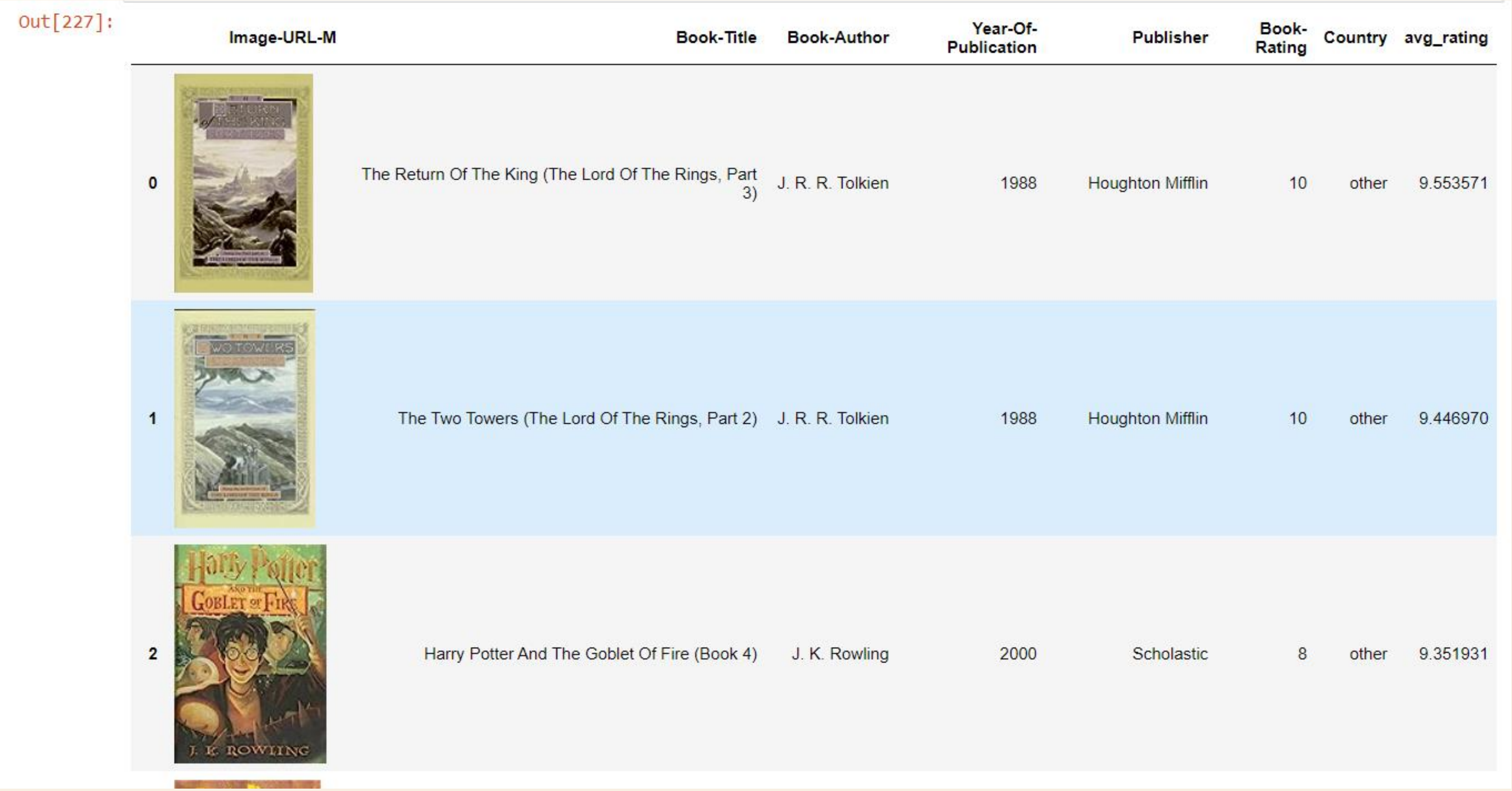


In [225]:
```
# Viewing the Final User-User Recommendation with full info
result = load_DF(ar)
result
```

Out[225]:

	Image-URL-M	Book-Title	Book-Author	Year-Of-Publication	Publisher	Book-Rating	Country
0		Harry Potter And The Sorcerer'S Stone (Book 1)	J. K. Rowling	2001	Scholastic	10	usa
1		The Fellowship Of The Ring (The Lord Of The Rings, Part 1)	J.R.R. Tolkien	2002	Houghton Mifflin Company	10	usa
2		Notes From A Small Island	Bill Bryson	1997	William Morrow And Company	10	usa

Recommending the popular books to user who are beginner readers

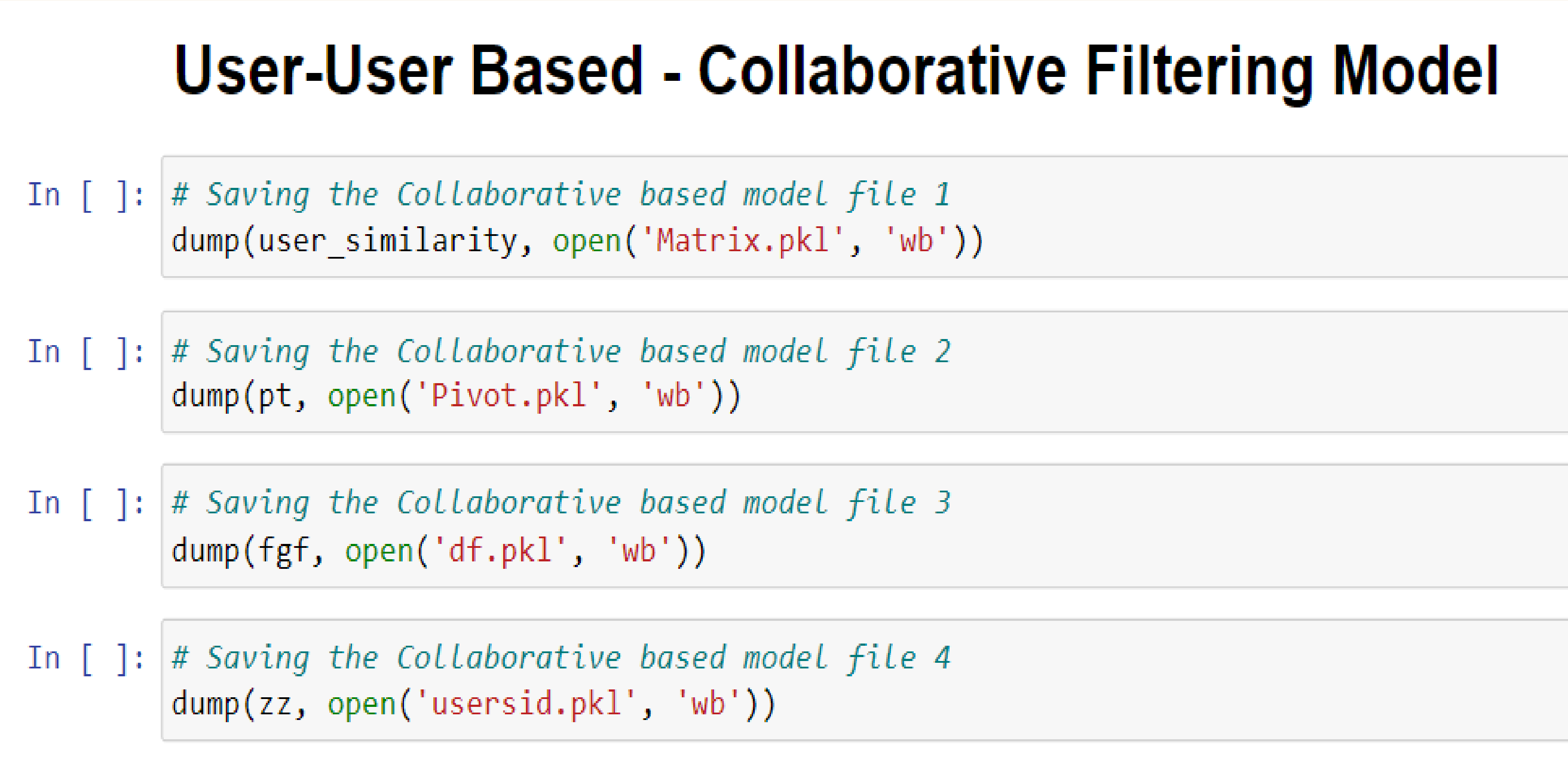


Out[227]:

	Image-URL-M	Book-Title	Book-Author	Year-Of-Publication	Publisher	Book-Rating	Country	avg_rating
0		The Return Of The King (The Lord Of The Rings, Part 3)	J. R. R. Tolkien	1988	Houghton Mifflin	10	other	9.553571
1		The Two Towers (The Lord Of The Rings, Part 2)	J. R. R. Tolkien	1988	Houghton Mifflin	10	other	9.446970
2		Harry Potter And The Goblet Of Fire (Book 4)	J. K. Rowling	2000	Scholastic	8	other	9.351931

DEPLOYMENT

DEPLOYMENT



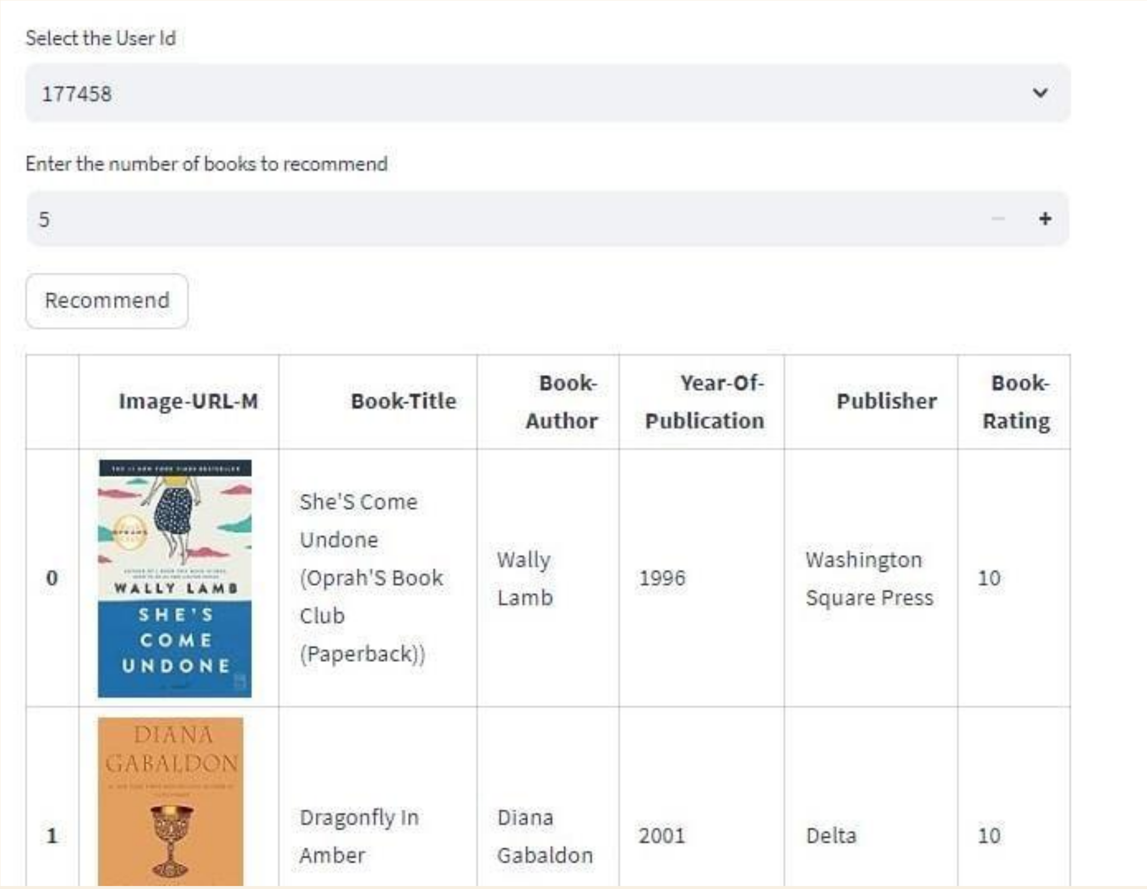
User-User Based - Collaborative Filtering Model

```
In [ ]: # Saving the Collaborative based model file 1
        dump(user_similarity, open('Matrix.pkl', 'wb'))
```

```
In [ ]: # Saving the Collaborative based model file 2
        dump(pt, open('Pivot.pkl', 'wb'))
```

```
In [ ]: # Saving the Collaborative based model file 3
        dump(fgf, open('df.pkl', 'wb'))
```

```
In [ ]: # Saving the Collaborative based model file 4
        dump(zz, open('usersid.pkl', 'wb'))
```



Select the User Id

177458

Enter the number of books to recommend

5

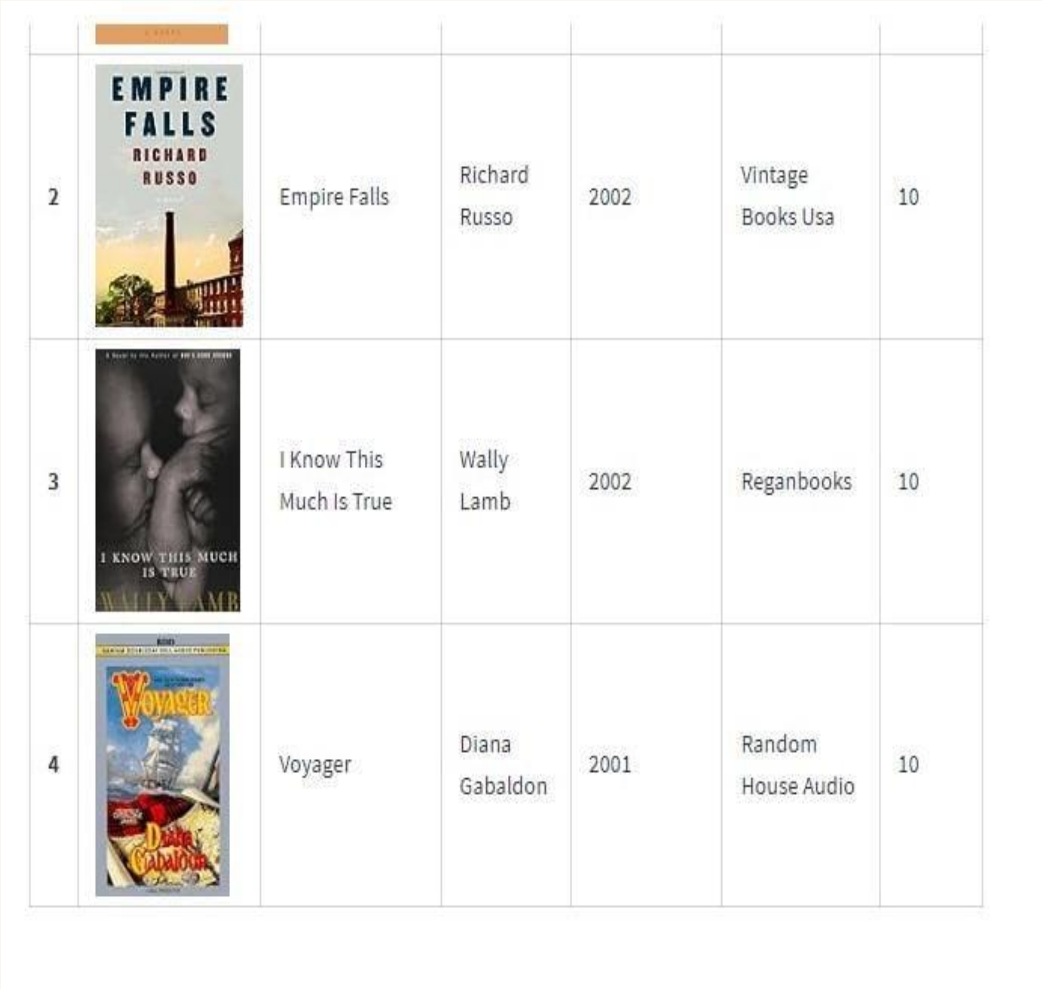


	Image-URL-M	Book-Title	Book-Author	Year-Of-Publication	Publisher	Book-Rating
0		She'S Come Undone (Oprah'S Book Club (Paperback))	Wally Lamb	1996	Washington Square Press	10
1		Dragonfly In Amber	Diana Gabaldon	2001	Delta	10
2		Empire Falls	Richard Russo	2002	Vintage Books Usa	10
3		I Know This Much Is True	Wally Lamb	2002	Reganbooks	10
4		Voyager	Diana Gabaldon	2001	Random House Audio	10

THANK YOU