Book Recommendation System

smart sensor systems
LIT05808.08
Introduction to Microsystem Technol
LIT05807.08
Köhler · Fritzsche
Nanotechnology
3G Evolution
HSPA and LTE for Mobile Broadband
LIT05805.08
Meyn Control Techniques for Complex Net
LIT05804.08
Security and Quality of Service in Ad Hoc Wireless
LIT05803.08
Bing Emerging Technologies in Wireless LANs
LIT05802.08
NEXT GENERATION IPTV SERVICES AND TECH
LIT05801.08

Kanike Lakshmi Narayana

Data science Trainee at

AlmaBetter

Abstract

Generally, everyone has their own experience where a website makes some personalized recommendations. Google chrome tells us people also searched for this. Instagram tells us these are the people who are friends to your friends, mutual friends and followers.

Problem Statement

During the last few decades, with the rise of YouTube, Amazon, Netflix, and many other such web services, recommender systems have taken more and more place in our lives.

From e-commerce (suggest to buyers articles that could interest them) to online advertisement (suggest to users the right contents, matching their preferences), recommender systems are today unavoidable in our daily online journeys.

Introduction

Machine Learning is concerned with computer programs that automatically improve their performance through experience. In machine learning, we have supervised learning, unsupervised learning and reinforcement learning. Again the supervised learning is further divided into regression and classification. In this project, we are going to look after unsupervised learning. In unsupervised machine learning the algorithm is applied to the unlabelled data. The data is neither classified nor labelled. The unsupervised learning algorithm is used to find the hidden structures with the unlabelled data.

A recommendation system is one of the top applications of data science. It offers relevant suggestions to the customer. Every consumer internet company requires a recommendation system like Netflix, YouTube, a news feed, etc.

Objective

The main objective of this project is to build a model that can recommend similar books to the customer.

Dataset Peeping

The bookcrossing dataset comprises 3 files. Owing to the size of the dataset, extensive cleaning of the dataset is needed. The following steps are performed for the analysis purpose:

- The data set has many NaN values and different methods are used to clean.
- Wrong data entry in the year feature is dropped and changed the data type.
- Location feature values are made in a dictionary and find the frequency of words.
- At last, all the datasets are merged into one file and performed exploratory analysis.

Data Design

Users

- Contains the users. Note that user IDs (User-ID) have been anonymised and mapped to integers.
- Demographic data is provided (Location, Age) if available. Otherwise, these fields contain NULL values.

Books

- Books are identified by their respective ISBN. Invalid ISBNs have already been removed from the dataset. Moreover, some content-based information is given (Book-Title, Book-Author, Year-Of-Publication, Publisher), obtained from Amazon Web Services. Note that in the case of several authors, only the first is provided. URLs linking to cover images are also given, appearing in three different flavours (Image-URL-S, Image-URL-M, Image-URL-L), i.e., small, medium, large. These URLs point to the Amazon website.

Ratings

- Contains the book rating information. Ratings (Book-Rating) are either explicit, expressed on a scale from 1-10 (higher values denoting higher appreciation), or implicit, expressed by 0.

Challenges Faced

The following are the challenges faced in the data analysis:

- Cleaning null values.
- Splitting words into separate features and extracting some insights.
- Merging the datasets.

Approach

As the problem statement says the main objective is to build a model that can recommend similar books for the customer and the given dataset does not have any labelled data. I have chosen to build a recommendation system based on collaborative filtering from unsupervised machine learning.

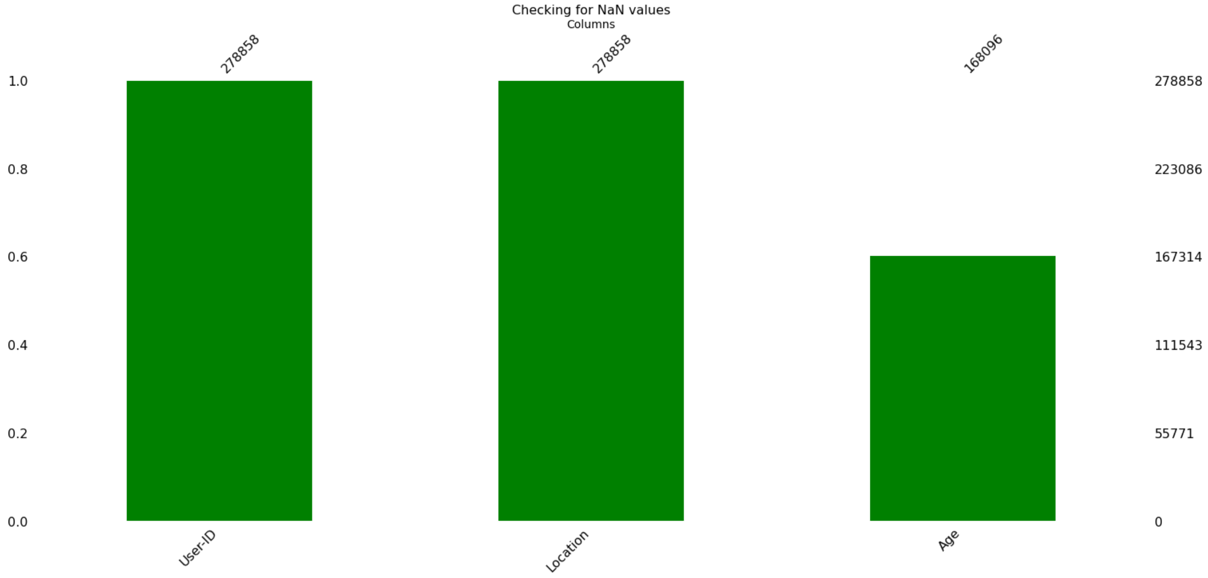
Tools Used

The whole project was done using python, in google collaboratory. Following libraries were used for analysing the data and visualizing:

- Pandas: Extensively used to load and wrangle with the dataset.
- Matplotlib: Used for visualization.
- Seaborn: Used for visualization.
- Plotly: Used for visualization.
- Missingno: For visualization.
- Warnings: For filtering and ignoring warnings.
- Numpy: For some math operations in predictions.
- Sklearn: For analysis and prediction.
- Statistics: For some math operations.

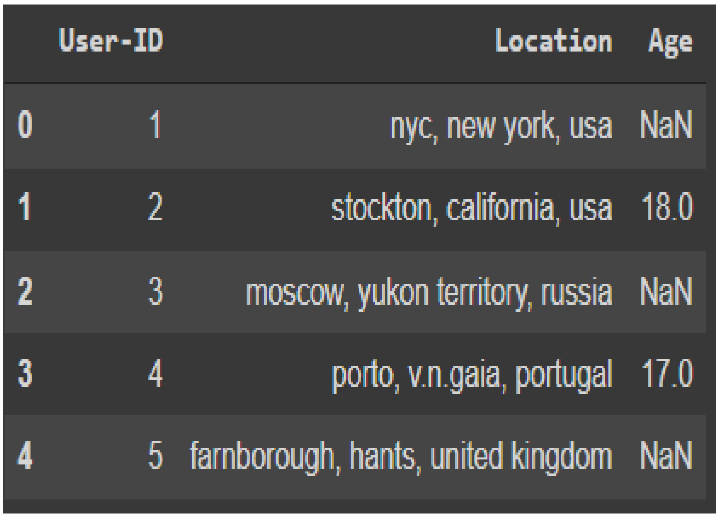
Users Dataset

Visualizing the presence of NaN values



The above bar plot shows that the Users dataset has almost 50 percent of the null values in the age feature.

Pandas DataFrame



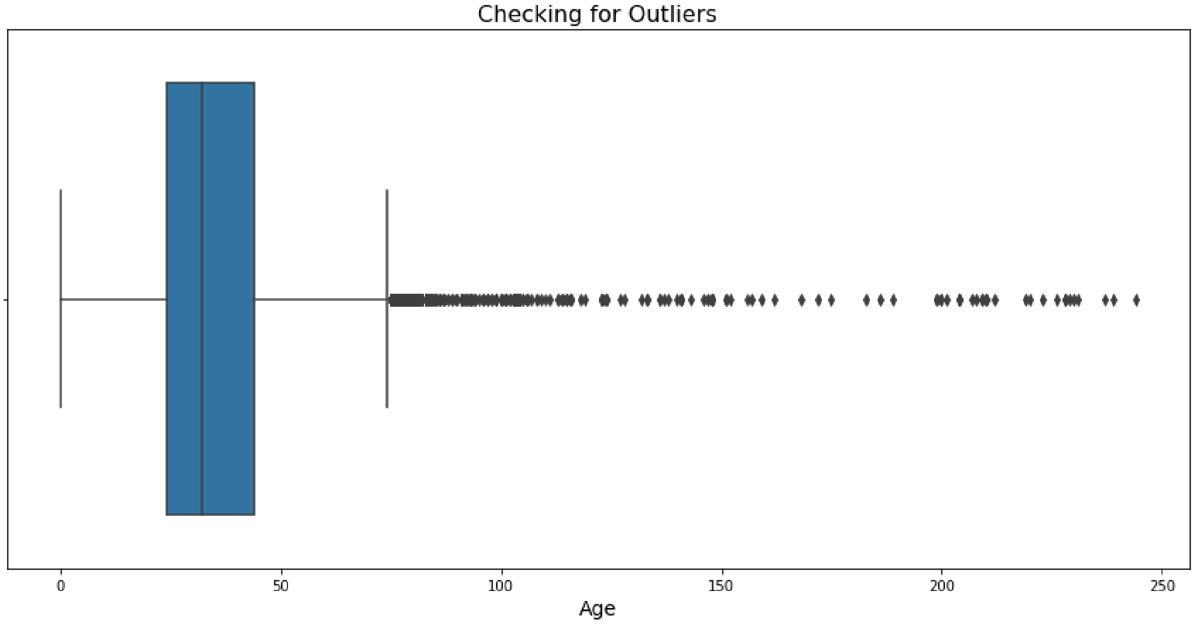
	User-ID	Location	Age
0	1	nyc, new york, usa	NaN
1	2	stockton, california, usa	18.0
2	3	moscow, yukon territory, russia	NaN
3	4	porto, v.n.gaia, portugal	17.0
4	5	farnborough, hants, united kingdom	NaN

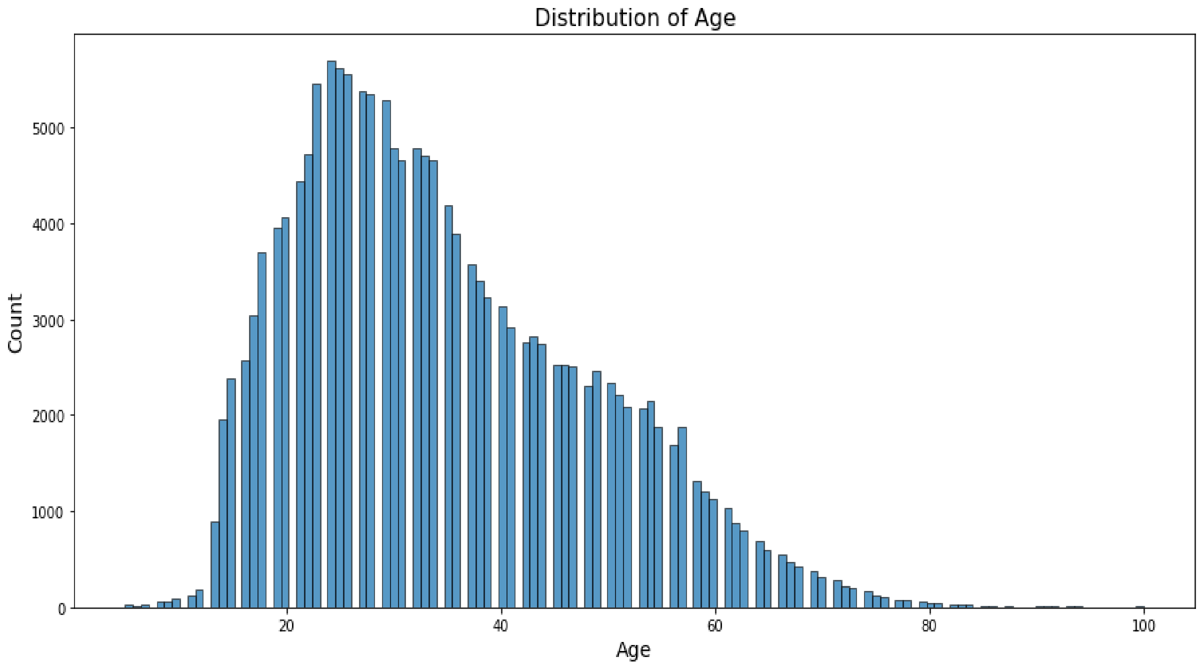
The table shows the User dataset in the form of Pandas DataFrame. The dataset has 278858 rows and 3 columns wholly the shape is (278858, 3).

It contains the following columns:

- User-Id
- Location
- Age

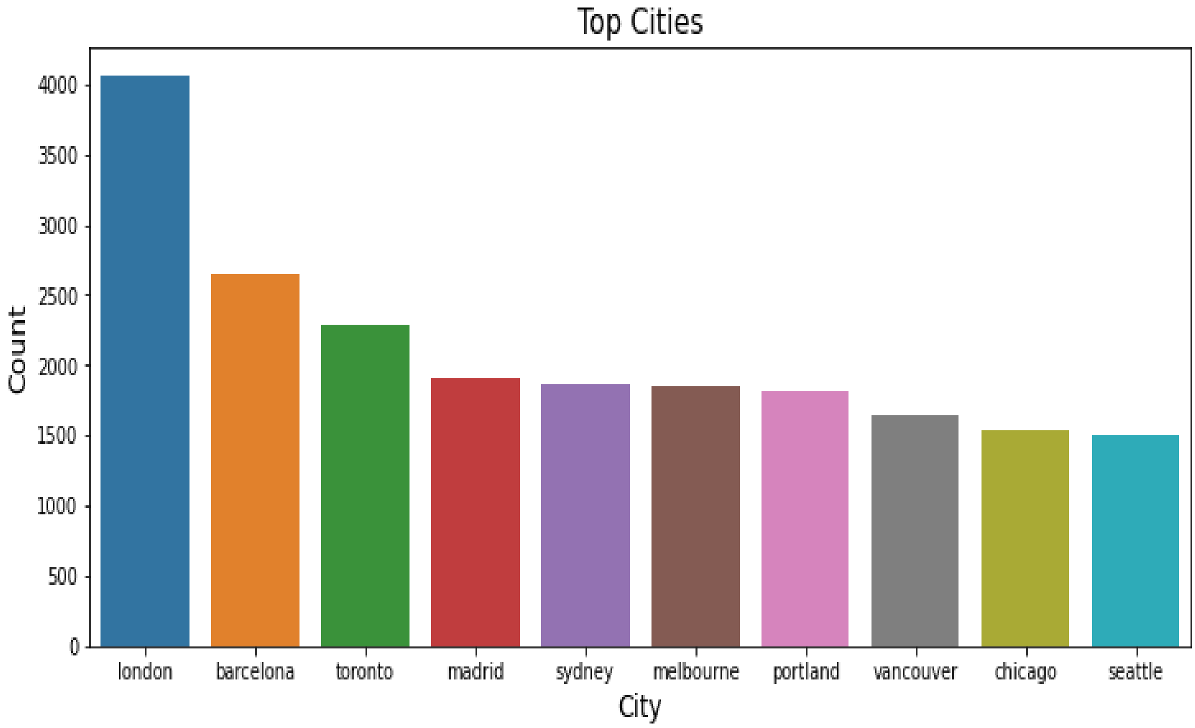
Outliers in Age

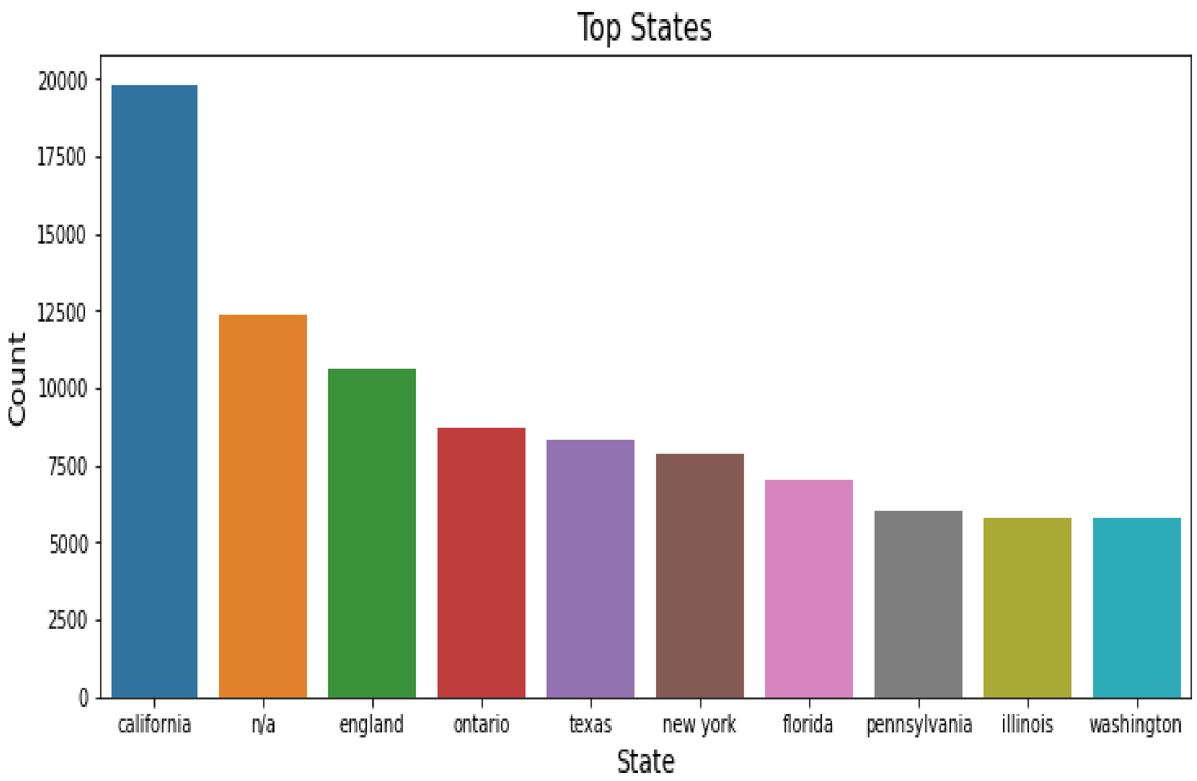


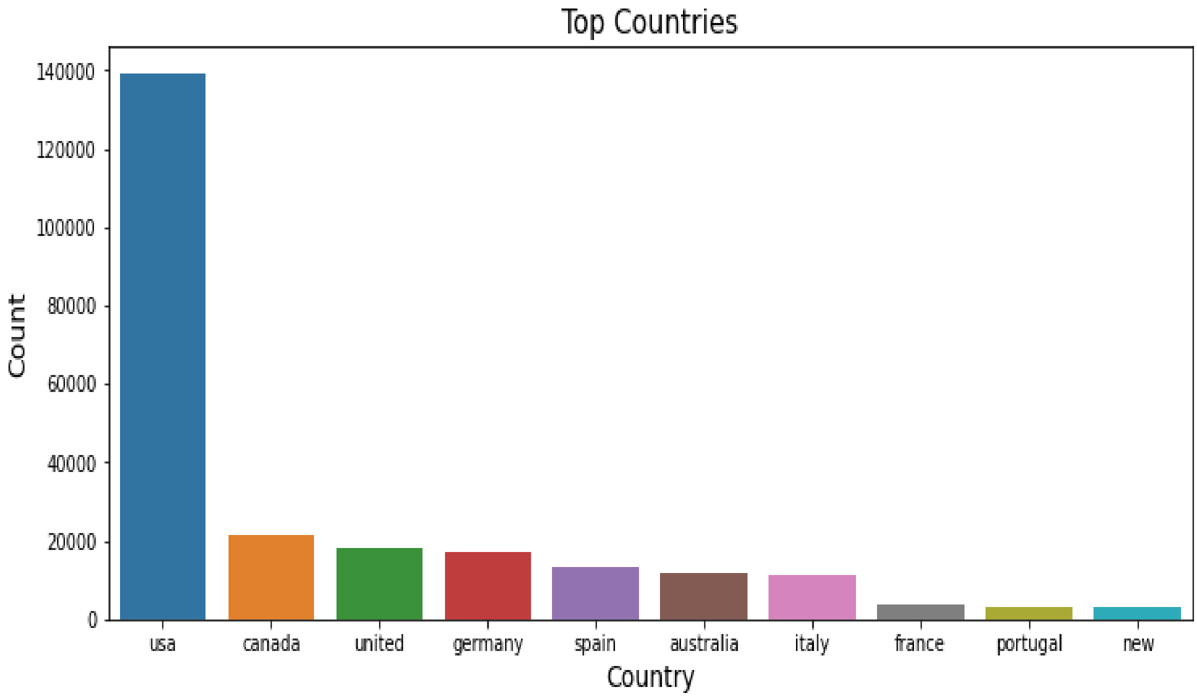


The age feature has the data of the age of the user from 0 to 250. It is assumed that the life span of users all over the dataset will be from 0 to 100. The data above 100 and below 5 are changed to NaN. At the Nan, values are around 50 percent that is replaced with a median of the feature

Analysis of Location



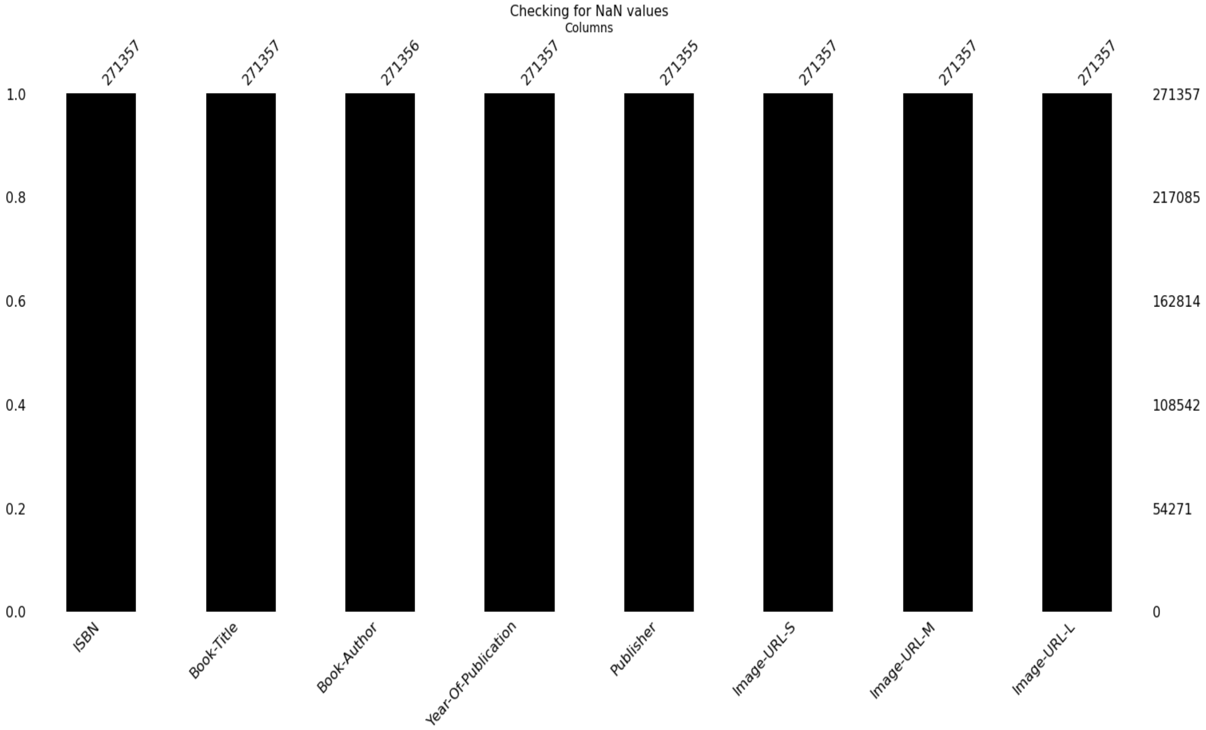




The location feature has a list of locations of users. It is assumed that it is a combination of city, state and country. As the same, these are extracted from the location feature. The figures show more customers are from the above geographical locations.

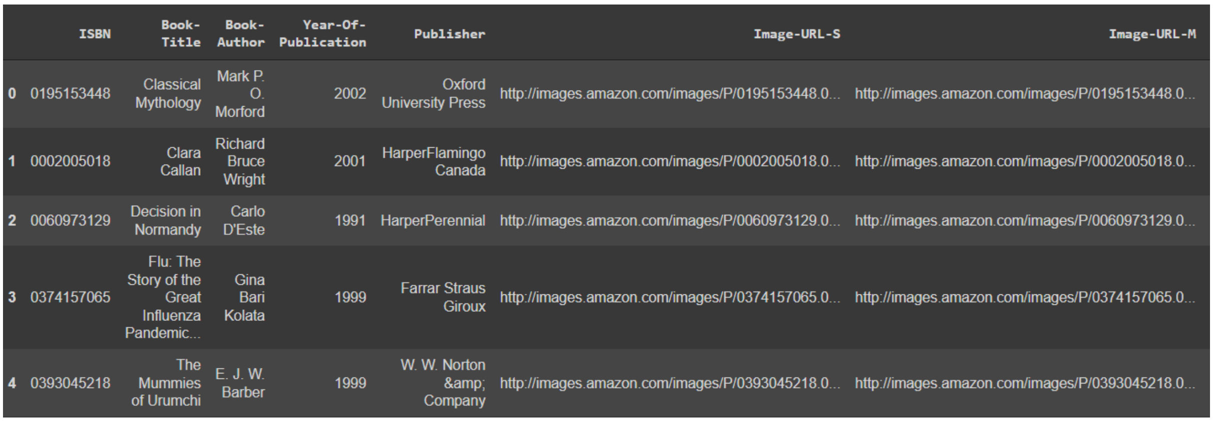
Books Dataset

Visualizing the presence of NaN values



The above figure shows that there are some null values in features and are cleaned by dropping.

Pandas DataFrame



	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...

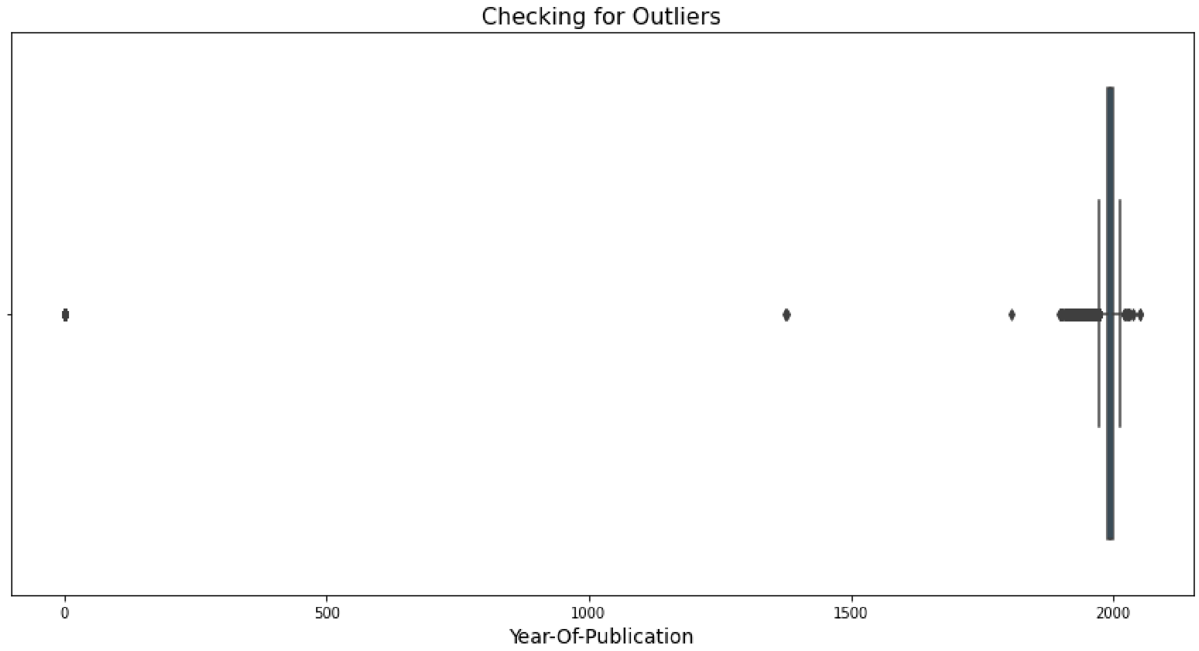
The table shows the Books dataset in the form of Pandas DataFrame. The dataset has 271354 rows and 8 columns; the shape is (271354, 8).

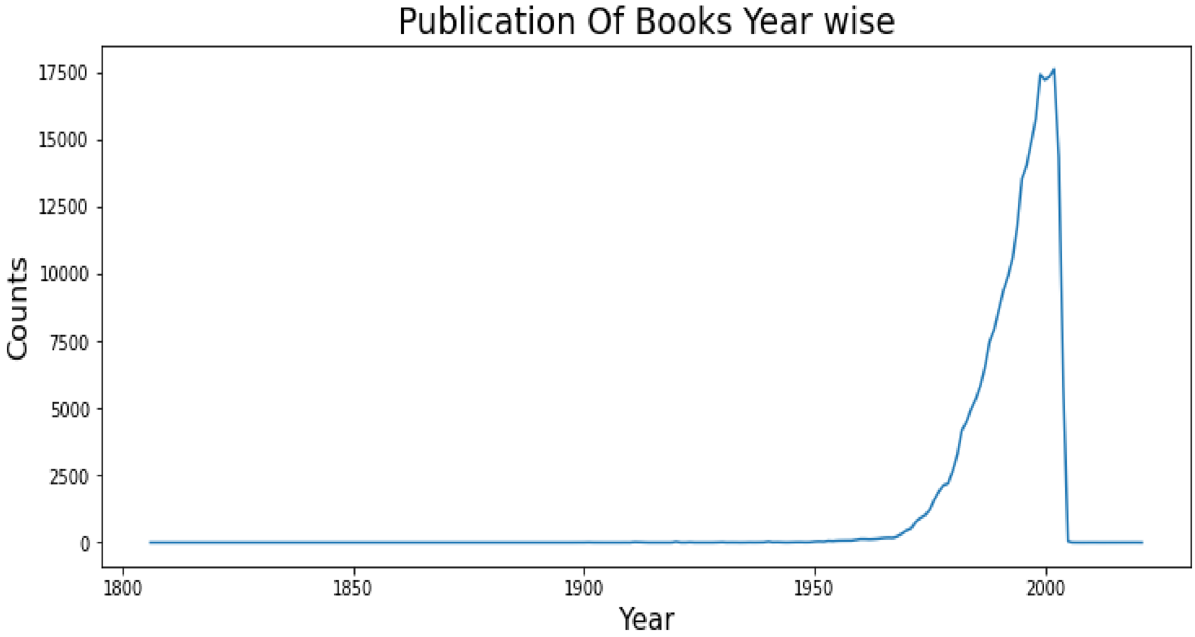
It contains the following columns:

- ISBN
- Book-Title

- Book-Author
- Year-Of-Publication
- Publisher
- Image-URL-S
- Image-URL-M
- Image-URL-L

Outliers in Year of Publication

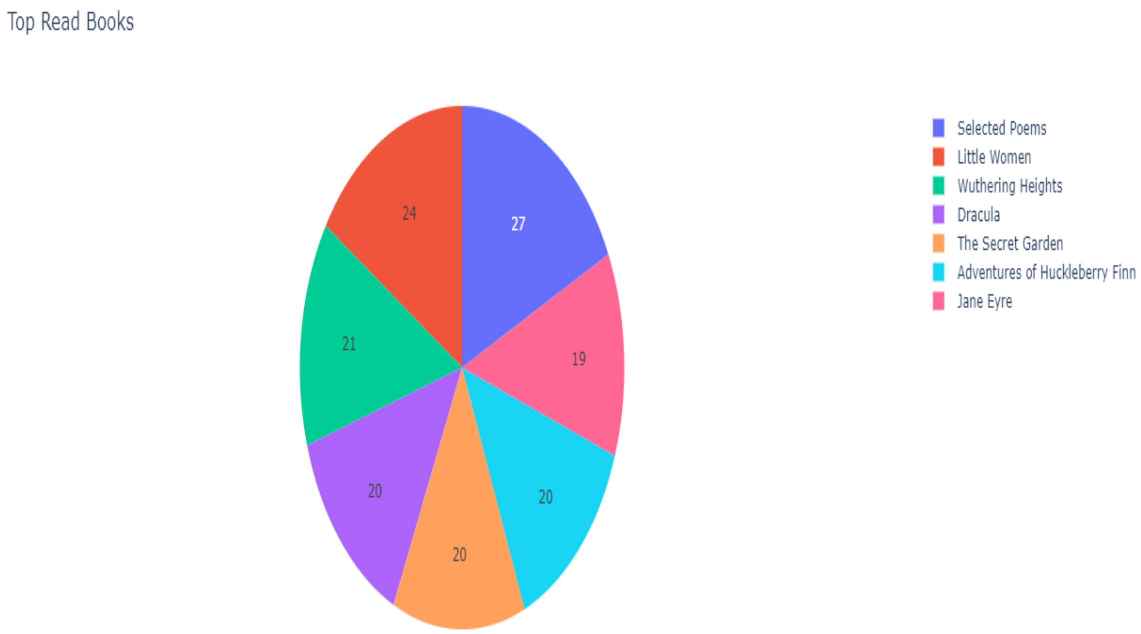


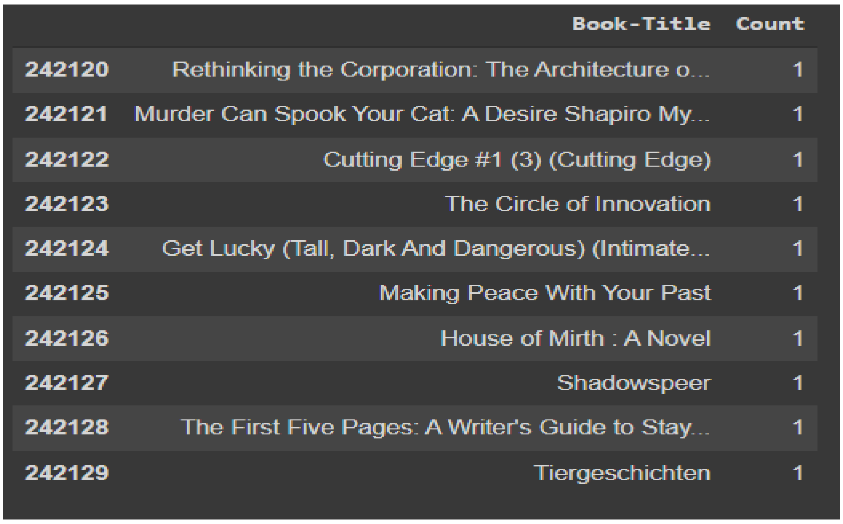


The year of publication data contains the books from 0 to 2050. As the data consists of outliers the data from 1800 to 2021 is taken into consideration and the remaining are at first replaced with Nan later with the median of the feature of the data.

The unwanted features are dropped.

Analysis on Book-Title

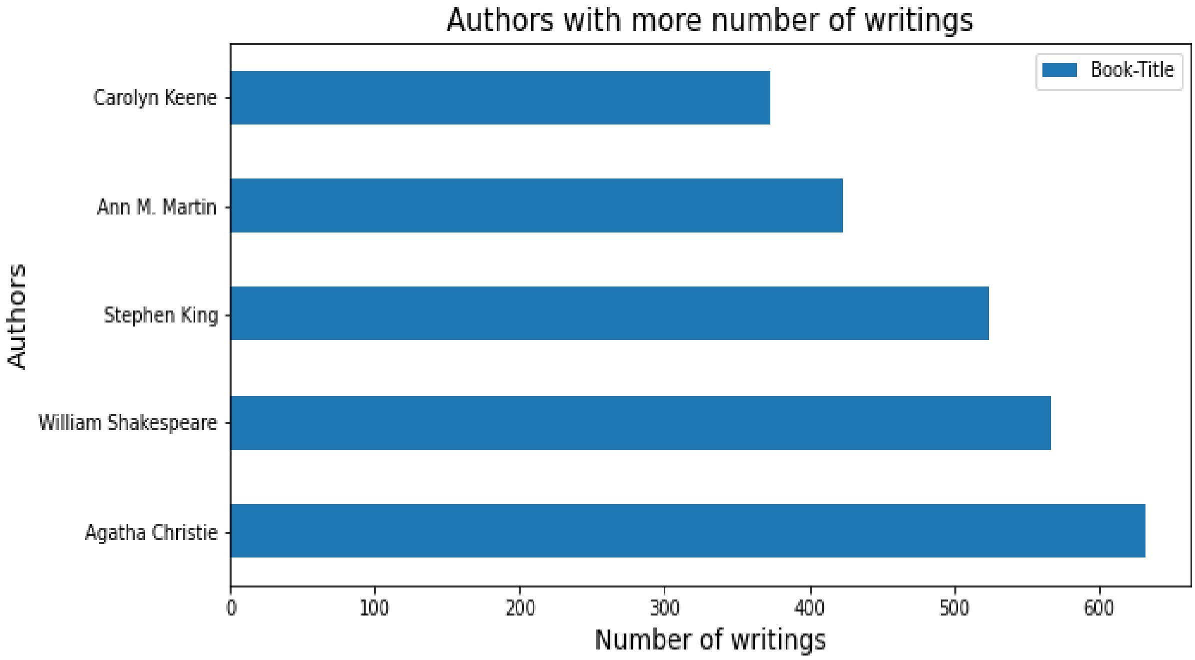


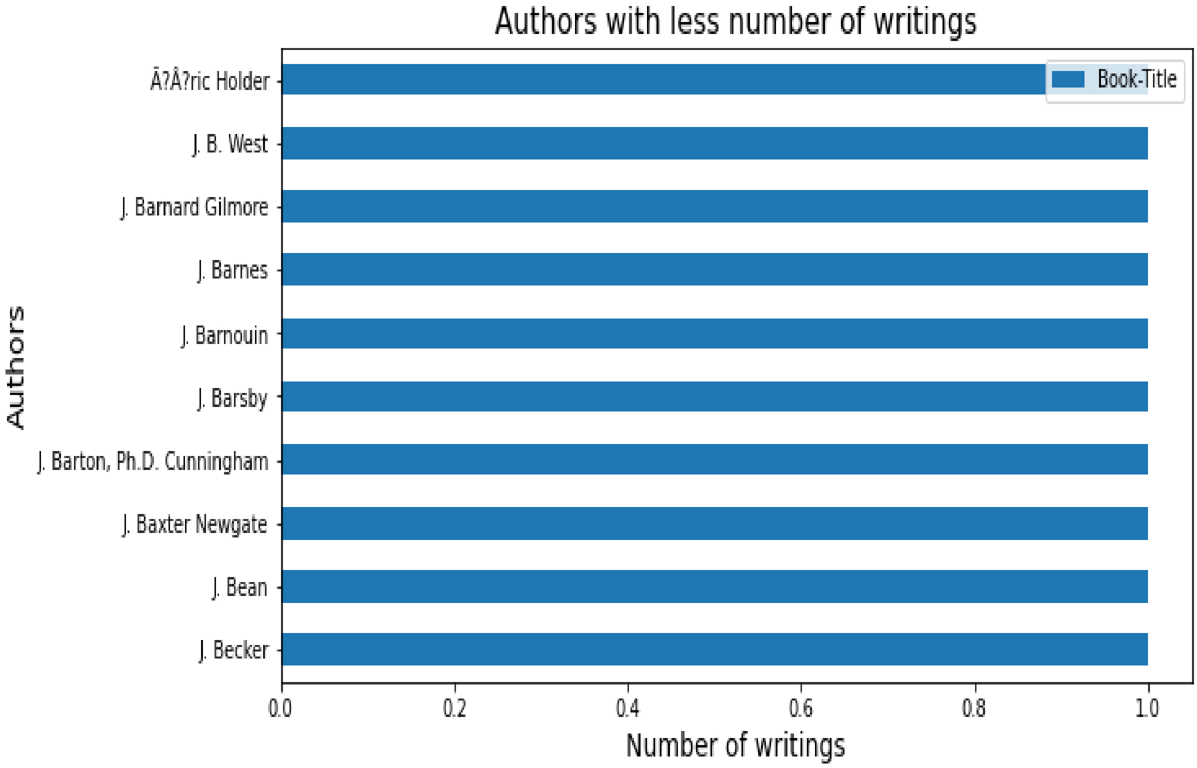


	Book-Title	Count
242120	Rethinking the Corporation: The Architecture o...	1
242121	Murder Can Spook Your Cat: A Desire Shapiro My...	1
242122	Cutting Edge #1 (3) (Cutting Edge)	1
242123	The Circle of Innovation	1
242124	Get Lucky (Tall, Dark And Dangerous) (Intimate...	1
242125	Making Peace With Your Past	1
242126	House of Mirth : A Novel	1
242127	Shadowspeer	1
242128	The First Five Pages: A Writer's Guide to Stay...	1
242129	Tiergeschichten	1

The above figures show the books which are mostly and rarely read by users.

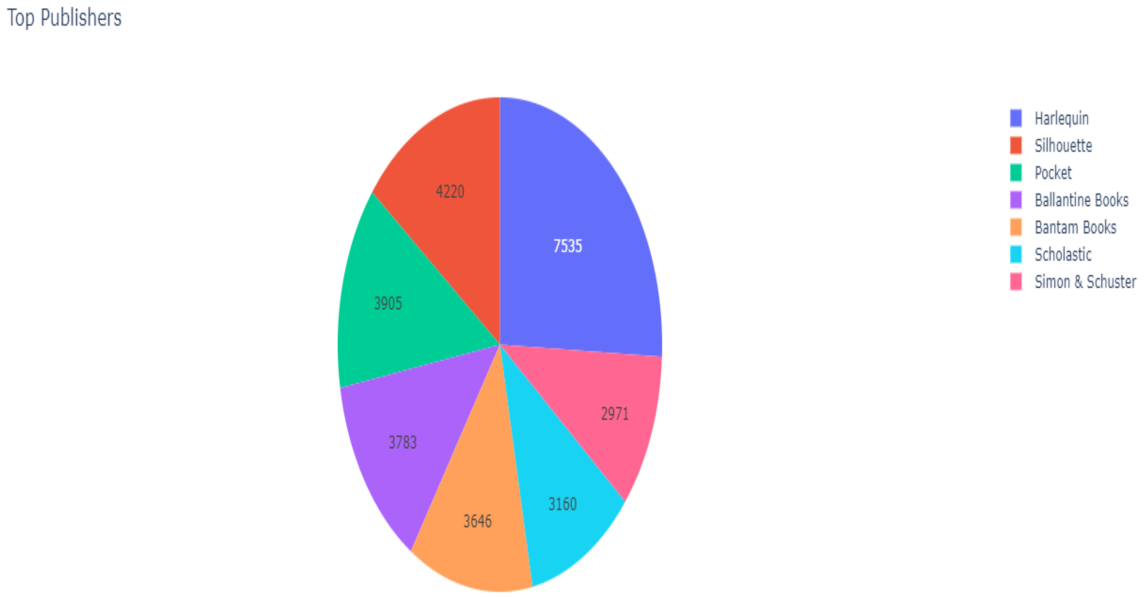
Analysis on Book-Author

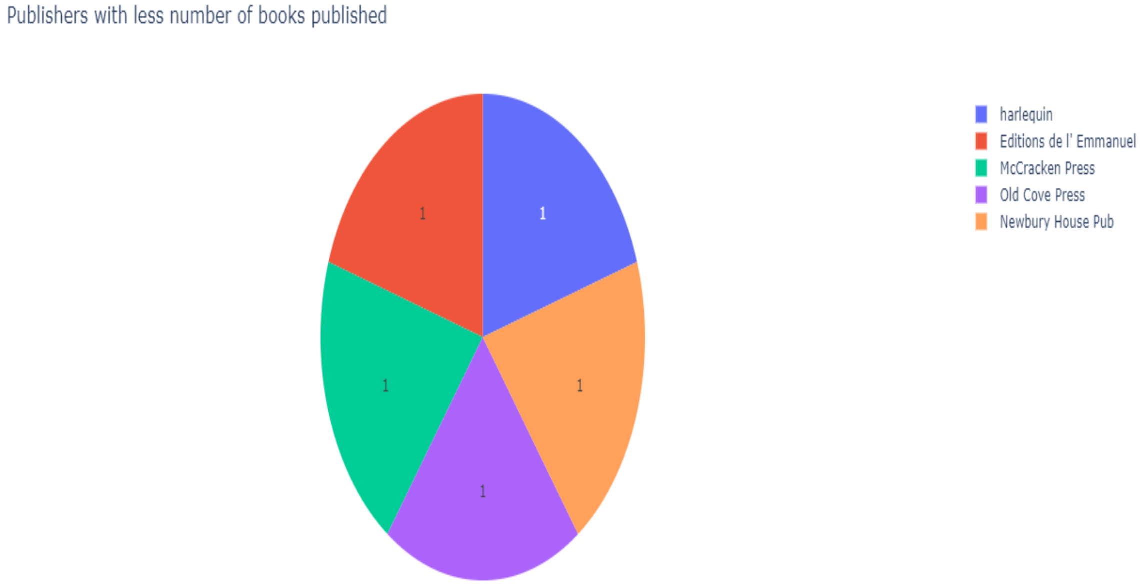




The above horizontal bar plots show the authors who are having a high and low number of writings. Among all authors, Agatha Christie has more writings.

Analysis on Publisher

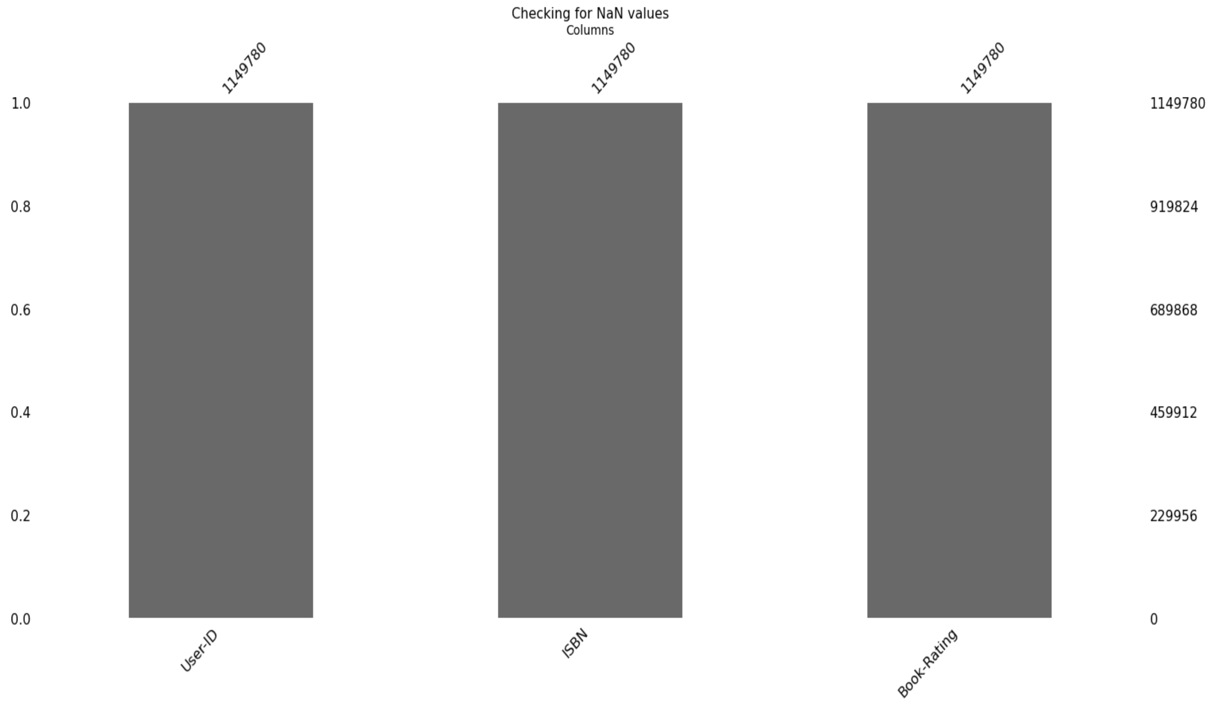




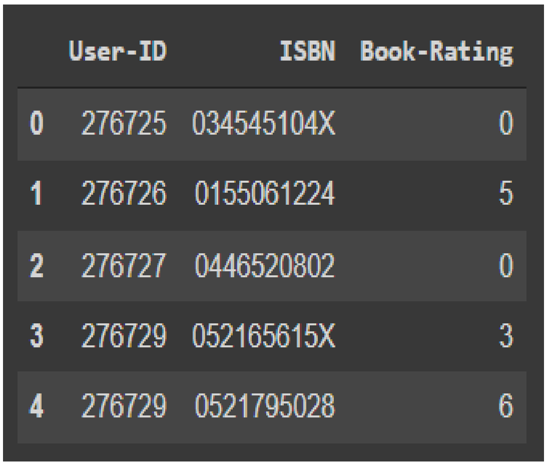
The above horizontal bar plots show the publishers who are having a high and low number of published books from their house. Among all Harlequin is having more books published from their house.

Ratings Dataset

Visualizing the presence of NaN values



Pandas DataFrame



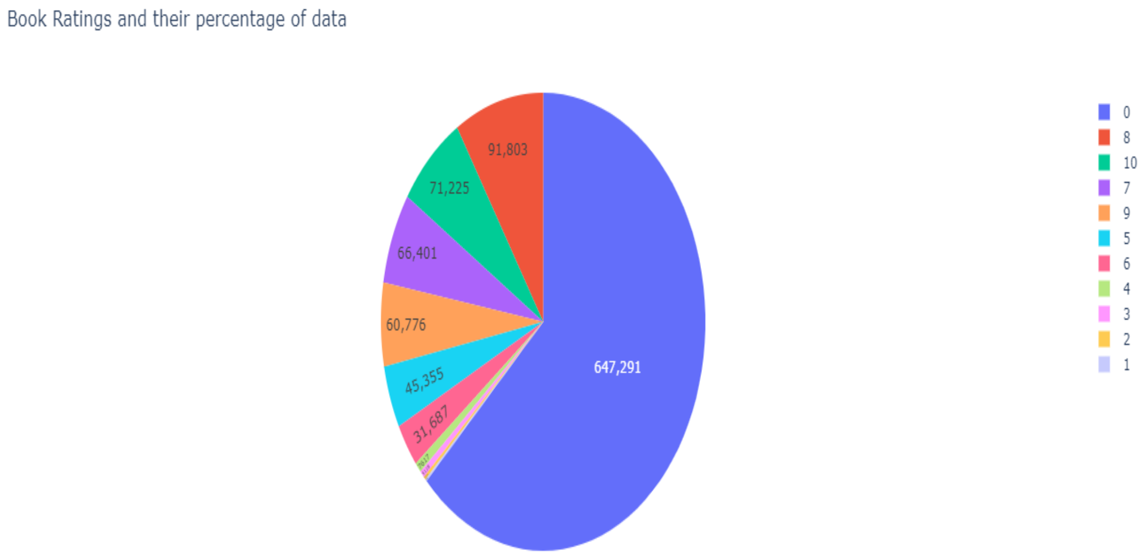
	User-ID	ISBN	Book-Rating
0	276725	034545104X	0
1	276726	0155061224	5
2	276727	0446520802	0
3	276729	052165615X	3
4	276729	0521795028	6

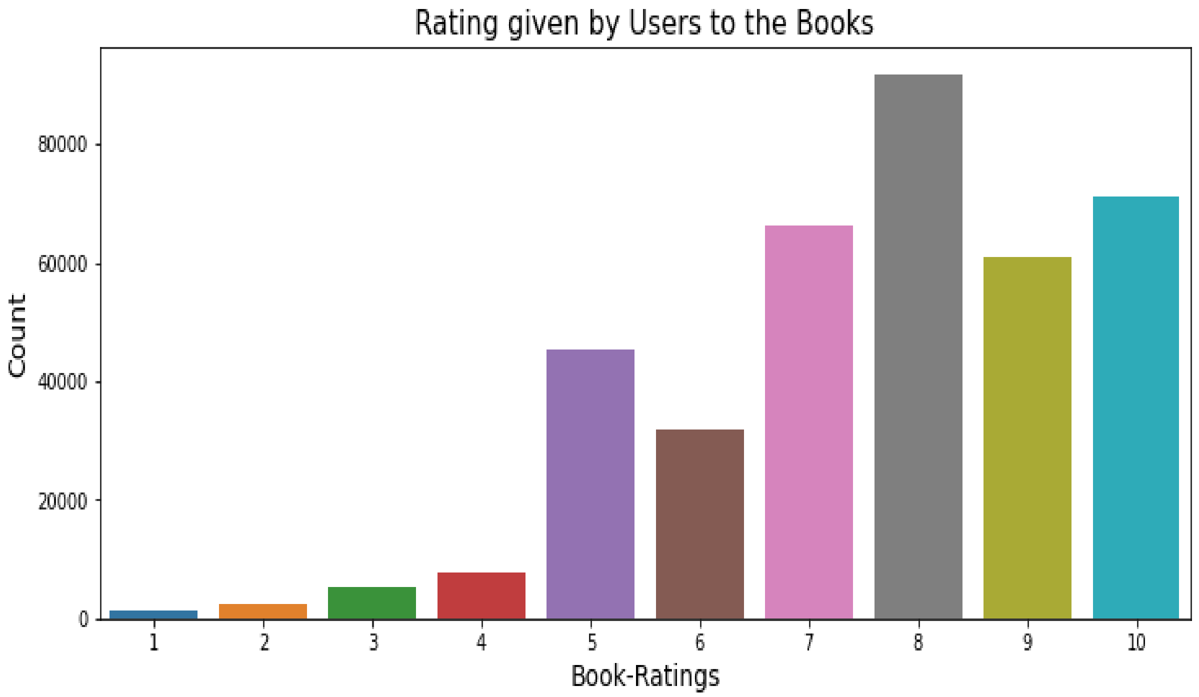
The table shows the Ratings dataset in the form of Pandas DataFrame. The dataset has 1149780 rows and 3 columns wholly the shape is (1149780, 3).

It contains the following columns:

- User-ID
- ISBN
- Book-Rating

Analysis on Book-Rating





The above pie chart shows the book rating from 0 to 10 given by the users. Almost 62 percent of the data is covered by 0 ratings. However, we are not going to suggest a 0 rated book to the user. So the data having 0 rated is dropped. 8 was frequently rated by users.

Data Preparation

After all the data was processed and exploratory analysis was done. And three datasets are merged into one DataFrame on common columns contained in it respectively and named as data.

Recommendation Engine

A recommendation system is one of the top applications of data science. It offers relevant suggestions to the customer and improves the user experience. Every consumer internet company requires a recommendation system like Netflix, YouTube, a news feed, etc. There are mainly three essential types of recommendation engines.

Content-based recommendation system

A content-based recommendation system works with data that the user provides, either explicitly (rating) or implicitly (clicking on a link). Based on the data, a user profile is generated and which is then used to make recommendations to the user.

Collaborative Filtering

The collaborative filtering method is based on collecting and analysing information based on behaviours, activities or user preferences and predicting what they will like based on the similarity with other users. In collaborative filtering, we have the following types.

- Model-based filtering recommendation system.
- Memory-based filtering recommendation system.

The selection of building recommendation models will be based on the sparsity of the data. Sparsity is defined as the number of elements or zero in a vector or matrix divided by the total number of entries in that vector or matrix. Feature sparsity refers to the sparsity of a feature vector whereas model sparsity refers to the sparsity of the model weights.

If the sparsity of the model is above 0.5 then we have to go with the model-based filtering recommendation model else we have to go with the memory-based filtering recommendation model. Again in the memory-based filtering recommendation model, we have the following types.

- User-Based recommendation system.

- Item-Based recommendation system.

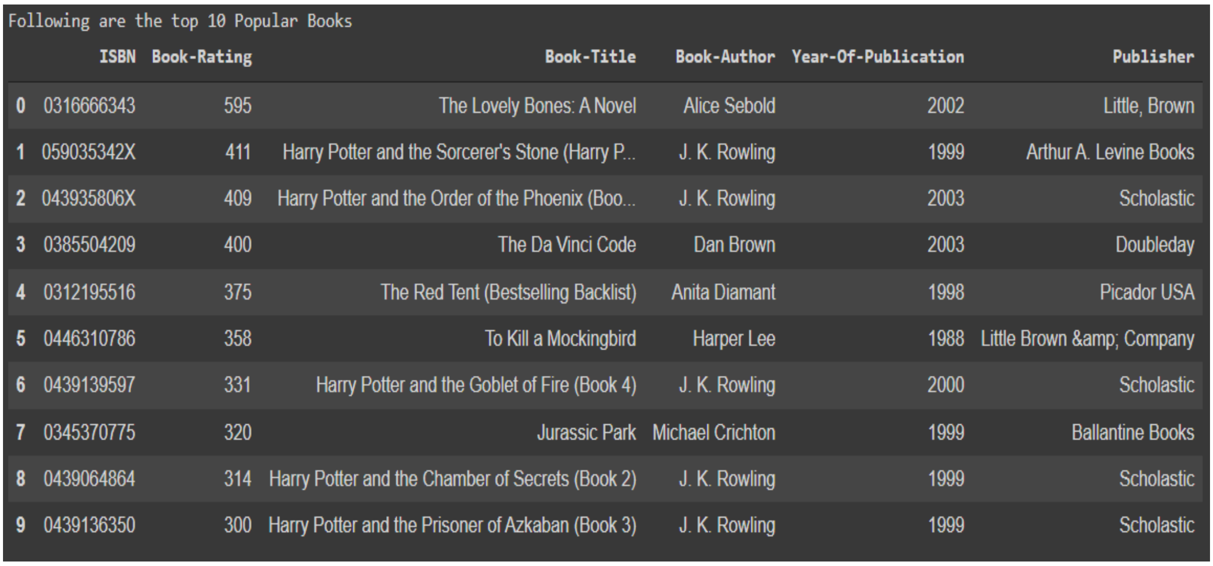
In this project, we are going to build a recommendation system based on collaborative filtering based on all the above models.

Cosine Similarity

Cosine similarity is a metric used to determine how similar two entities or documents are irrespective of their size. This could be used in building a recommendation system to recommend similar products, movies, shows, books and restaurants.

Popularity Based Recommendation System

The popularity based recommendation system is one of the types of recommendation systems based on collaborative filtering. This model suggests products to the customer based on the popularity of the product or item.



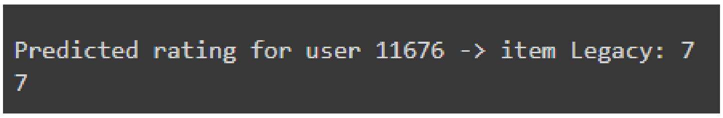
```
Following are the top 10 Popular Books
```

	ISBN	Book-Rating	Book-Title	Book-Author	Year-Of-Publication	Publisher
0	0316666343	595	The Lovely Bones: A Novel	Alice Sebold	2002	Little, Brown
1	059035342X	411	Harry Potter and the Sorcerer's Stone (Harry P...	J. K. Rowling	1999	Arthur A. Levine Books
2	043935806X	409	Harry Potter and the Order of the Phoenix (Boo...	J. K. Rowling	2003	Scholastic
3	0385504209	400	The Da Vinci Code	Dan Brown	2003	Doubleday
4	0312195516	375	The Red Tent (Bestselling Backlist)	Anita Diamant	1998	Picador USA
5	0446310786	358	To Kill a Mockingbird	Harper Lee	1988	Little Brown &amp; Company
6	0439139597	331	Harry Potter and the Goblet of Fire (Book 4)	J. K. Rowling	2000	Scholastic
7	0345370775	320	Jurassic Park	Michael Crichton	1999	Ballantine Books
8	0439064864	314	Harry Potter and the Chamber of Secrets (Book 2)	J. K. Rowling	1999	Scholastic
9	0439136350	300	Harry Potter and the Prisoner of Azkaban (Book 3)	J. K. Rowling	1999	Scholastic

The above are the books recommended by the model based on popularity.

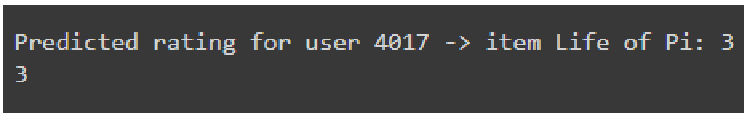
User-Based Recommendation System

The user-based recommendation system involves building a model based on the dataset of ratings. In other words, we extract the information from the dataset and use that as a model to make recommendations without having the use of a complete dataset every time. It assumes users are similar if they like similar items. In this model, we have used the k nearest neighbour model to find the similar user nearest to that user. 10 is taken as the k value and cosine similarity as a metric for the model evaluation.



```
Predicted rating for user 11676 -> item Legacy: 7
7
```

In the above example, we can see that the model has predicted a rating of 7 to the Legacy book and can be recommended to the user of having the userid as 11676 to read the book.

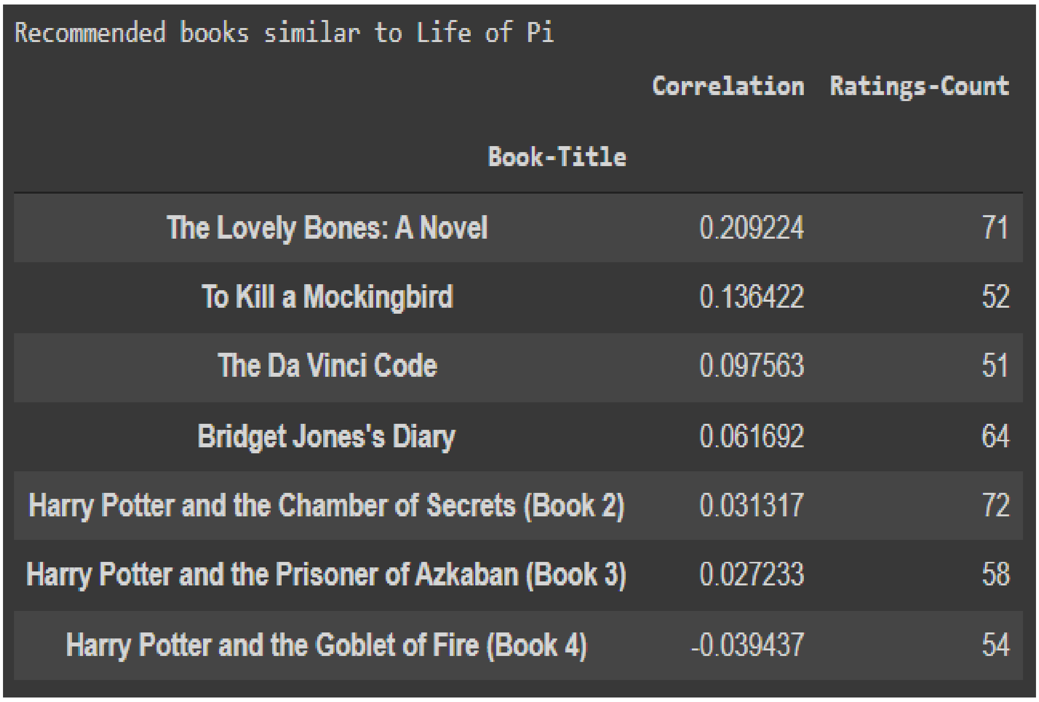


```
Predicted rating for user 4017 -> item Life of Pi: 3
3
```

In the above example, we can see that the model has predicted a rating of 3 to the Life of Pi book and can be not recommended to the user of having the userid as 4017 to read the book.

Item Based Recommendation System

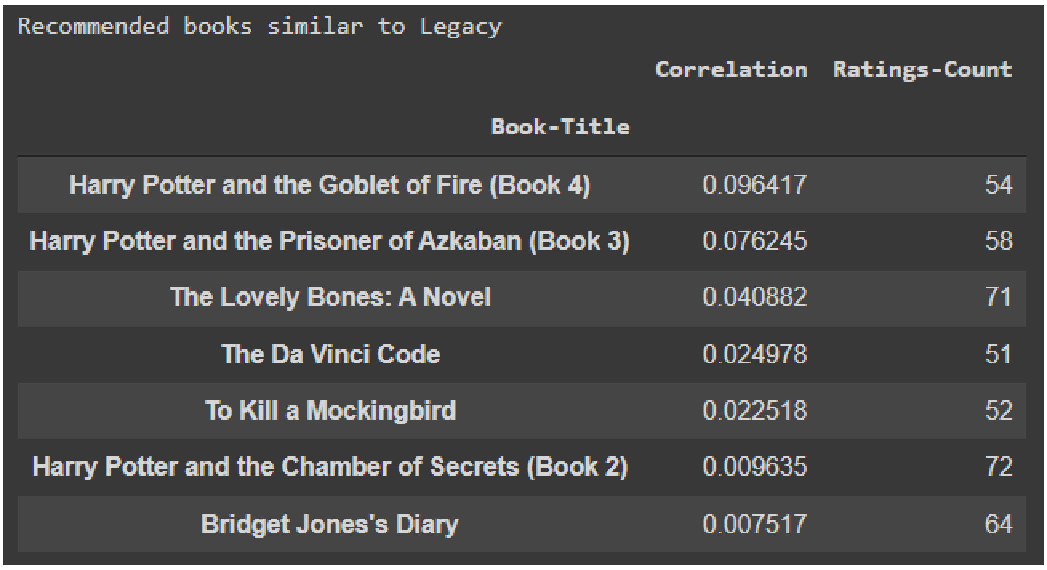
The item-based recommendation system suggests the item based on the similarity between items. It is calculated using the rating given by the users to such items. It helps to solve the issues that user-based collaborative filters suffer from such as when the system has many items with fewer items rated.



Recommended books similar to Life of Pi

	Correlation	Ratings-Count
Book-Title		
The Lovely Bones: A Novel	0.209224	71
To Kill a Mockingbird	0.136422	52
The Da Vinci Code	0.097563	51
Bridget Jones's Diary	0.061692	64
Harry Potter and the Chamber of Secrets (Book 2)	0.031317	72
Harry Potter and the Prisoner of Azkaban (Book 3)	0.027233	58
Harry Potter and the Goblet of Fire (Book 4)	-0.039437	54

The above table shows that the model has suggested the following books to the user who has read the Life of Pi book.



Recommended books similar to Legacy

Book-Title	Correlation	Ratings-Count
Harry Potter and the Goblet of Fire (Book 4)	0.096417	54
Harry Potter and the Prisoner of Azkaban (Book 3)	0.076245	58
The Lovely Bones: A Novel	0.040882	71
The Da Vinci Code	0.024978	51
To Kill a Mockingbird	0.022518	52
Harry Potter and the Chamber of Secrets (Book 2)	0.009635	72
Bridget Jones's Diary	0.007517	64

The above table shows that the model has suggested the following books to the user who has read the Legacy.

Conclusion

- Before replacing the outliers in age most of the users are from the 20 to 40 age group.
- After separating the city, state and country in the location column we came to know that most of the users are from California, USA.
- The outliers in the year feature are replaced with a median value.
- Selected poems were read more by users.
- Among all authors, Agatha Christie has more writings.
- Harlequin is having more books published from their house.
- 8 was frequently rated by users.