ES 114: Data Narrative 1

Jaidev Sanjay Khalane (22110103)

B.Tech 2022
Computer Science and Engineering
Indian Institute of Technology
Gandhinagar, India
jaidev.khalane@iitgn.ac.in

***Abstract*— This report is about the analysis of the data given in the website https://github.com/zygmuntz/goodbooks-10k about the various parameters of the books in a library/website, such as the ratings, publishing dates, etc. Various scientific questions and hypotheses were also created in order to effectively study the data. The software used is Python 3 with modules such as NumPy, Pandas and matplotlib.**

I. Overview Of The Dataset

This dataset gives data about a large number of books. It provides the raw data about the books in terms of various parameters such as the number of ratings, the average rating, the number of copies of that book, the year of publishing that book, the author of the book, the language of the book, the topic of the book, number of readers of that book, etc. The data of the book is given in the form of the unique ID of the book. In this Data Narrative, I have used various libraries of Python 3 to visualize the data and draw out certain trends that can be visually analysed. Plotting using the books ID is convenient for handling large datasets. We can later correlate the book ID with the book's actual title.

II. Scientific Questions about the Dataset

1. What is the readership of a particular book, and how can we compare it to other books (visually)? What are the ten most read books present on the website?
2. What is probability that a book was published in a given particular year? What trend can be visualised by taking the cumulative data up to that year? How would a smooth curve look like?
3. What analysis can be done based on the language of the books? How can we visualise the probability of getting a book in a particular language? What are the ten most common languages in the library?
4. What is the rating distribution, rating against the number of books with that rating? How can we analyse the general trend of the average ratings of the books against other books? Speaking about the user data, what is the analysis of the average rating given by the reader to all the books read by the reader, that is, the user v/s the average rating given by that user?
5. What are the top 100 most popular titles of the books (from the first word of tag names)? Who are the top 50 authors in terms of publishing more books?
6. What is the trend of the book count against the book id? What is the visualisation that can be provided based on that?
7. What analysis about the relative popularity of the books can be drawn from the data of ratings count, work ratings count and work text review count against the book ID?
8. If we choose a book, what is the probability that the book is being by a particular reader? Explain it by visualisation.

III. Details of Libraries used

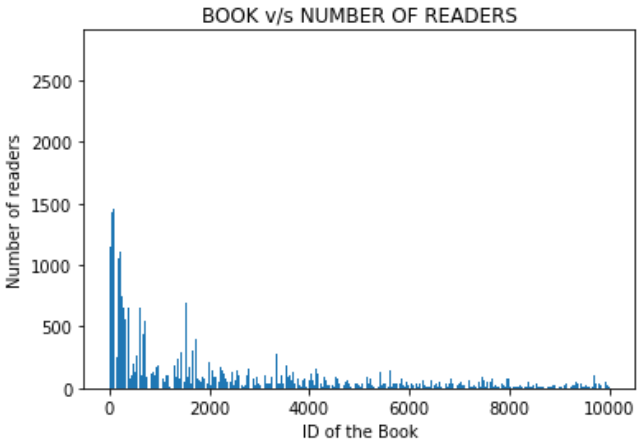
The libraries that are used include matplotlib, NumPy, and Pandas.

- matplotlib.pyplot: This library was used for applications such as plotting in the form of a line graph, bar graph, histogram etc. and other applications such as legend, the label of axes, title, grid, etc.
- pandas: pandas were used for reading the data from .csv files, creating dataframes, iterating through them, and deleting the wrongly formatted data points using the function df.drop.
- NumPy: I used the NumPy library for creating NumPy arrays for faster computing and storing data, for calculating mean, to get the unique elements out of the NumPy arrays, and by enabling the counts, I also got the number of times the unique element has occurred in the array and its index.

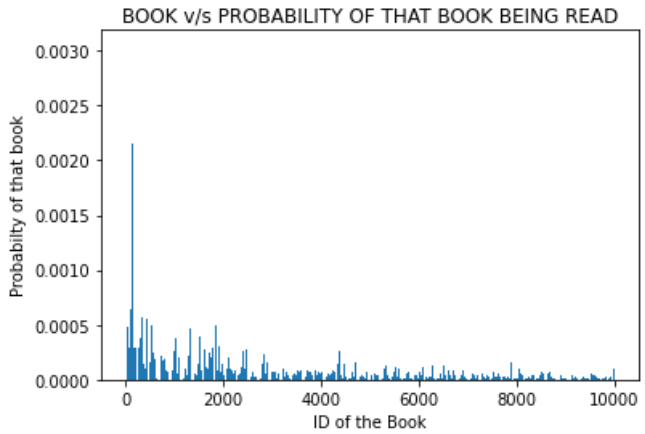
IV. Answers to the Questions and Summaries Drawn

1. What is the readership of a particular book, and how can we compare it to other books(visually)? What are the ten most read books present on the website?

A1. First, I read the .csv file https://raw.githubusercontent.com/zygmuntz/goodbooks-10k/master/to_read.csv' using the read_csv() of the pandas



module, and then I created a dictionary that would store the unique elements and their count. Then I plotted the book's readership that is given by the count using the matplotlib. This may be a useful parameter to find the impact of a book. The popularity of the book can be judged by this method. The probability was obtained by dividing the original count by the total.



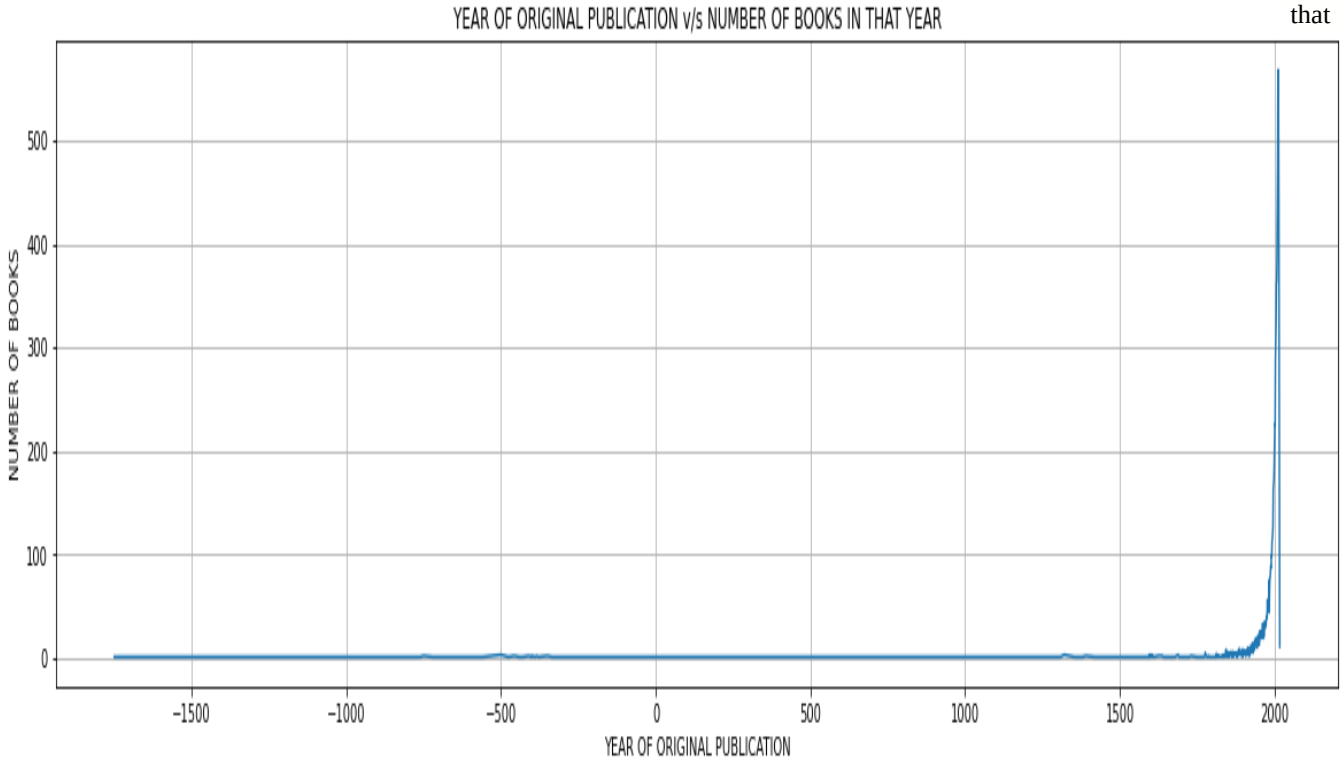
This is the obtained plot. We can also observe that the books with lower IDs generally have a larger readership .

The most common books were discovered by first making the NumPy arrays of the book IDs. Then the numpy.unique function was implemented on the array, which returned the arrays having the unique book IDs with their respective counts. Then the book IDs and their respective counts were formed into a pandas DataFrame, which was sorted by the column of "counts" in descending order. Then an array was made from the column of Book IDs which is arranged in the order of largest to smallest in terms of the readership base. Then the csv file 'https://raw.githubusercontent.com/zygmuntz/goodbooks-10k/master/books.csv' was opened as a DataFrame using the pandas. Then the column of 'original_title' was accessed and converted to an array. Then the array of the top 10 books ID was iterated, mapping to the sorted array of books ID in terms of number of readers to access the top 10 book titles in terms of readership. The top 10 most popular books were:

`Fahrenheit 451, Unbroken: A World War II Story of Survival, Resilience, and Redemption, Tuesdays with Morrie, Animal Farm: A Fairy Story, Divergent, Water for Elephants, Luftslottet som sprängdes, Eat, pray, love: one woman's search for everything across Italy, India and Indonesia, Gone with the Wind, The Giver`

2. What is probability that a book was published in a given particular year? What trend can be visualised by taking the cumulative data up to that year? How would a smooth curve look like?

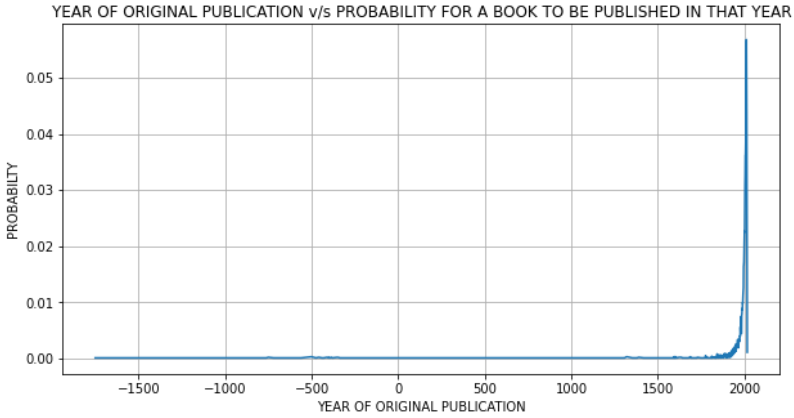
A2. First the csv file 'https://raw.githubusercontent.com/zygmuntz/goodbooks-10k/master/books.csv' was opened and converted to a pandas DataFrame using the pandas library. Then the column of the original year of publication was made into a NumPy array and using the numpy.unique function, the unique publishing years were returned with the number of books published in that year.

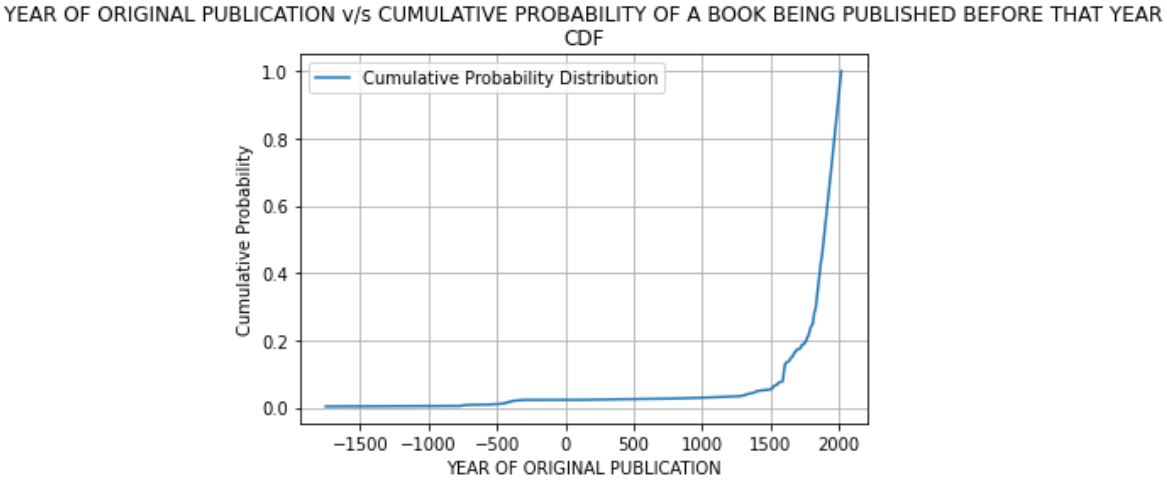


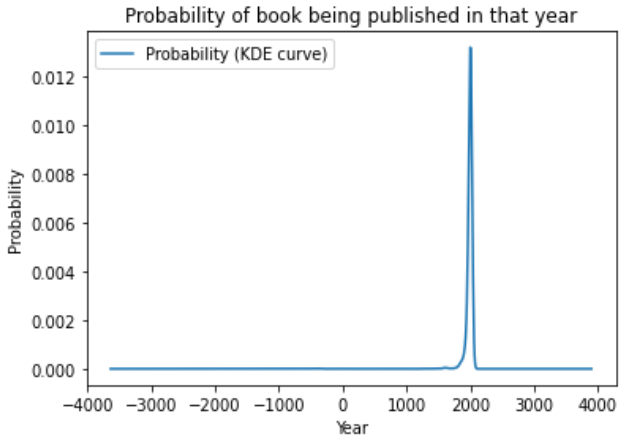
Finally, this data was plotted with the publishing year on the x axis and the number of books divided by the total number of books on the y axis to get the plot of the probability of getting a book published that year. In order to get the cumulative number of books published up to that year, the numpy.cumsum function was used and the plotting was done in the same way as given above. Finally, in order to plot the cumulative distribution function, the array of cumulative number of books was divided by the total number of books. This was plotted in the same way as given above. From this, I got the cumulative distribution of probability of getting a book published in a particular year.

Using the pandas df.plot.kde(), Kernel density estimation function, the dataframe was plotted into a smooth curve to give the probability of getting a book published in a particular year.

We can observe the probability distribution from the plots given below that many books are published during 19th and 20th centuries, but some are as old as from 17th century B.C.E.



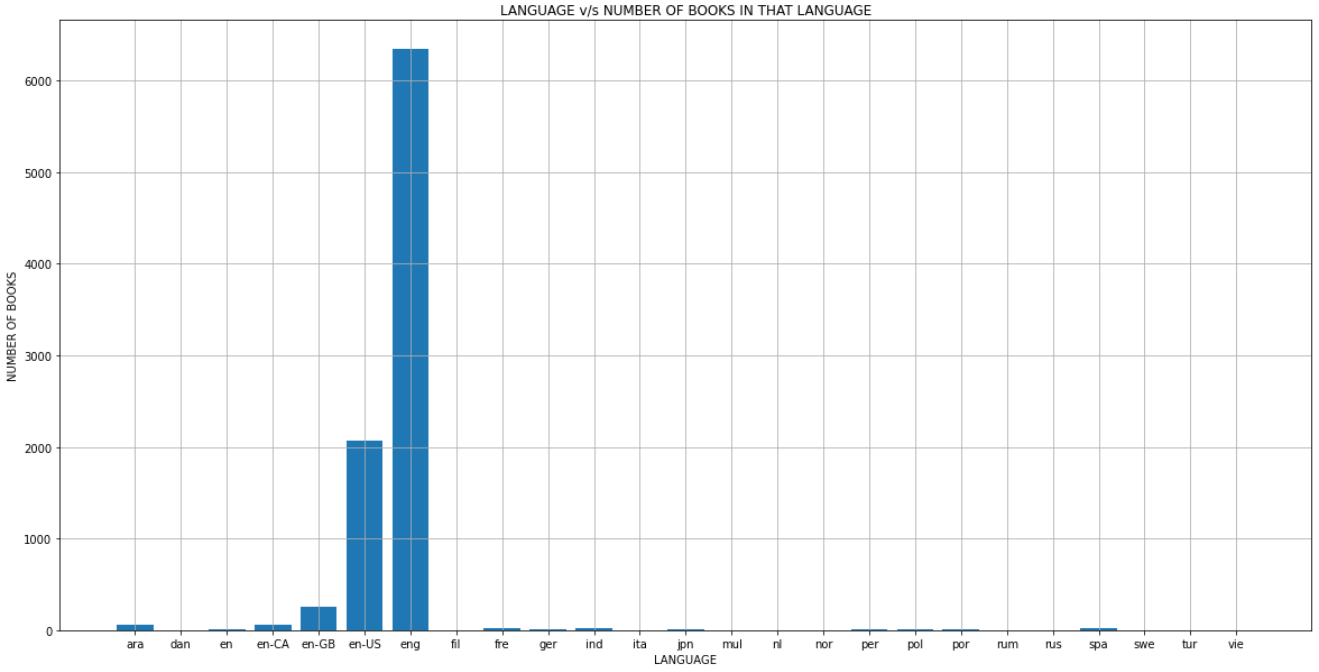


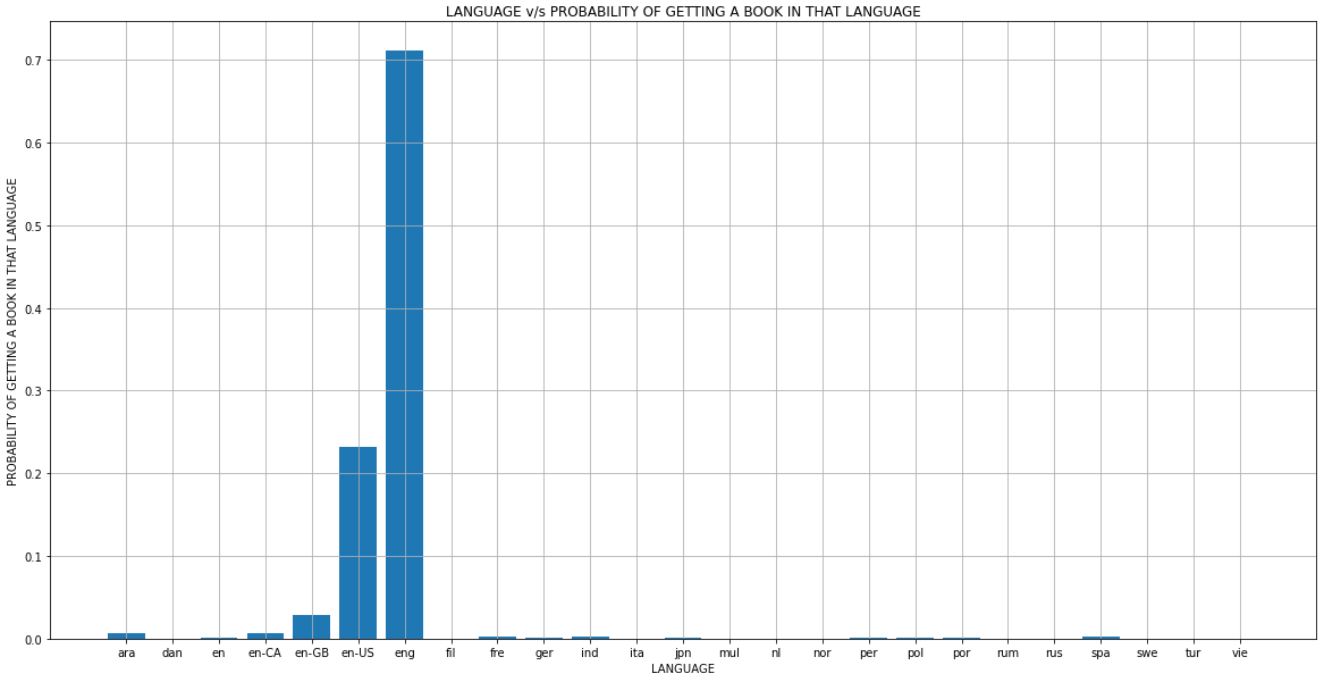


3. What can be analysed based on the language of the books? How can we visualise the probability of getting a book in a particular language? What are the 10 most popular languages of the books?

A3. First of all, the csv file 'https://raw.githubusercontent.com/zygmuntz/goodbooks-10k/master/books.csv' was opened and converted to DataFrame using the pandas module. Then the rows which had invalid type of data entry (not string type) in the column of languages were deleted using the pandas drop function.

Then a NumPy array was created out of the DataFrame column of 'language_code'. Then the numpy.unique function was used to return the unique names of languages and the counts of the books in those languages. Then this data of the language and its count was plotted using the matplotlib with appropriate labelling. In order to plot the probability of getting a book of a particular language, the count of the books of all languages were divided by the total count of all the books and then plotted using matplotlib. In order to get the 10 most popular languages of the books, the data of the language names and their count was stored in the form of dataframe. Then this dataframe was sorted in the descending order from the by the column of "Count". Then the names column of the dataframe was extracted and converted into a NumPy array of which the first 10 values were printed using the join function after converting to a list.





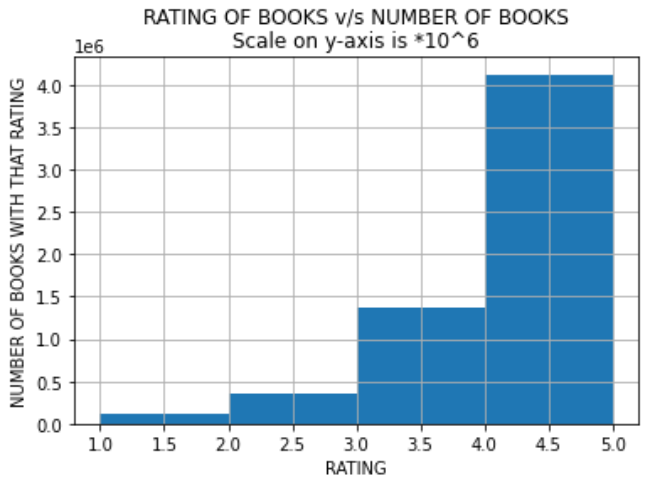
Most common languages: `eng, en-US, en-GB, ara, en-CA, fre, ind, spa, ger, per`

Summary from the observations is that the languages such as English, Arabic, French, etc. are extremely common in the books present in the website.

4. **What is the rating distribution, rating against the number of books with that rating? How can we analyse the general trend of the average ratings of the books against other books? Speaking about the user data, what is the analysis of the average**

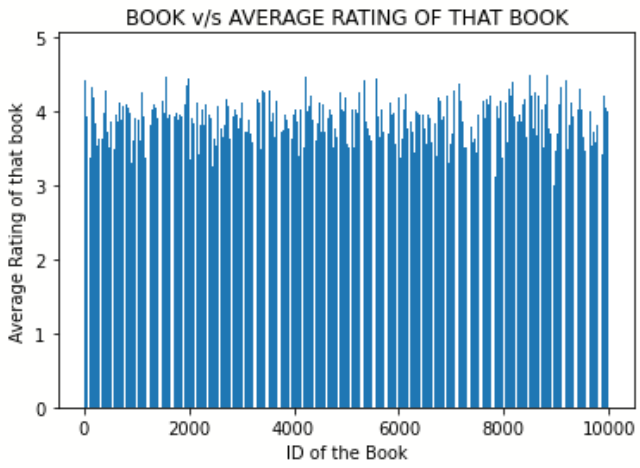
rating given by the reader to all the books read by the reader, that is, the user v/s the average rating given by that user?

A4. First, I read the file 'https://raw.githubusercontent.com/zygmuntz/goodbooks-10k/master/ratings.csv' using the pandas function of read_csv and obtained a dataframe using that. The dataframe contains columns of book_ID and ratings. So, I accessed the column of 'ratings' and formed it into a NumPy array. Then I plotted a histogram with five bins in order to get the idea of the general trend of ratings of the books available in our website or library.



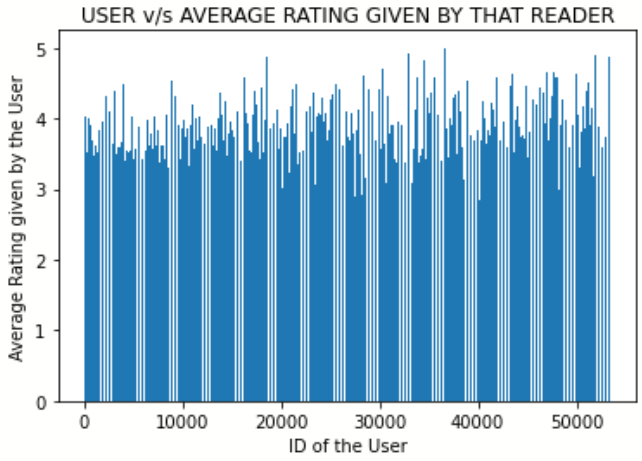
As we can observe from the histogram, majority of the books have the ratings between 4-5. Very few books have extremely low rating of 1-2. So, we can conclude that the books in this website are quite well rated.

The analysis of the book's average rating against the book was done in the following way. First, the .csv file 'https://raw.githubusercontent.com/zygmuntz/goodbooks-10k/master/ratings.csv' was read using the read_csv function of pandas and saved as 'df'. Then a dictionary was formed with books ID and the array for the rating of the book. As the df was being traversed, if the book was new, a new key was created in the dictionary with a list containing the rating of that book. If the book was already present in the dictionary, the rating was just appended to the already existing list of the book. Finally, the NumPy mean of the array was taken, which denotes the average rating of the book. Using matplotlib, the following Bar plot was obtained by plotting the average rating against the book ID.



As we can observe from the graph that the average rating of the books is almost always around 4. This confirms the observation from the previous plot of rating of book.

The question regarding the users' data about the average ratings given 'by the user' against the user ID is as given in the following plot. The method is similar to the previous part. We can get the response of any particular user using this plot.



The average rating per user has a tremendous importance in terms of individual feedback of the user towards the books in the website/ library. As we can observe from the plot is that the feedback is almost positive.

5. **What are the top 100 most popular titles of the books (from the first word of tag names)? Who are the top 50 authors in terms of publishing more books (most popular authors)?**

A5. First of all, the csv file https://raw.githubusercontent.com/zygmuntz/goodbooks-10k/master/tags.csv was opened using pandas and converted into a dataframe. Then a NumPy array was made using the names of the tags (tag names).Then the array was iterated, and each item was split by '–' and the first part of it was taken. Then using the numpy.unique, I got the unique first names of

the tags with the count of the number of occurrences of those names. Then the names and their counts were converted into a dataframe. Then the dataframe was sorted according to the number of counts in the descending order, and then the dataframe's column of "title" was again iterated to remove the words such as "a", "the", "", etc. Finally top hundred tag first names were printed.

The most common titles were :

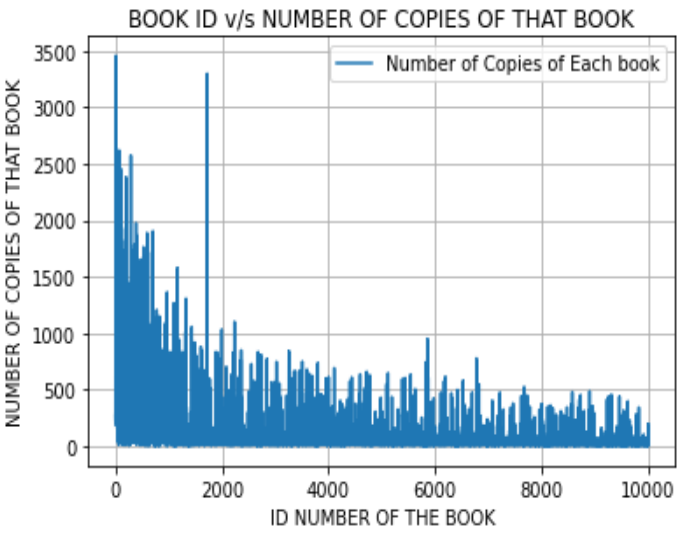
art book tbr children manga john mystery christian cookbooks comics genre food science author read black series kindle kids ya favorite history owned library 1 graphic not own james books i american best business historical new 2017 my fiction fantasy david dark adult cooking want self sci horror 2016 روايات ز philosophy first no أدب religion reading life health 1001 british world child personal middle alex crime modern all richard have كتب michael poetry young romance childrens robert spiritual shelfari comic women classic summer animal nonfiction 100 social pulitzer family love short

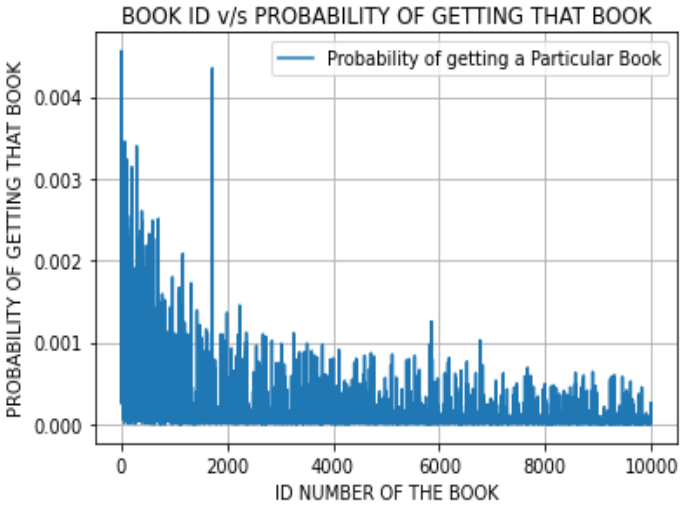
To get the data about the authors who published the most books, first of all, the csv file https://raw.githubusercontent.com/zygmuntz/goodbooks-10k/master/books.csv was opened using pandas and converted into a dataframe. Then a NumPy array was made using the column of names of the authors. Then using the numpy.unique, I got the names of the authors with the count of the number of occurrences of their names. Then the names and their counts were converted into a dataframe. Then the dataframe was sorted according to the number of counts in the descending order, and then the top fifty authors' names were printed by using the join function.

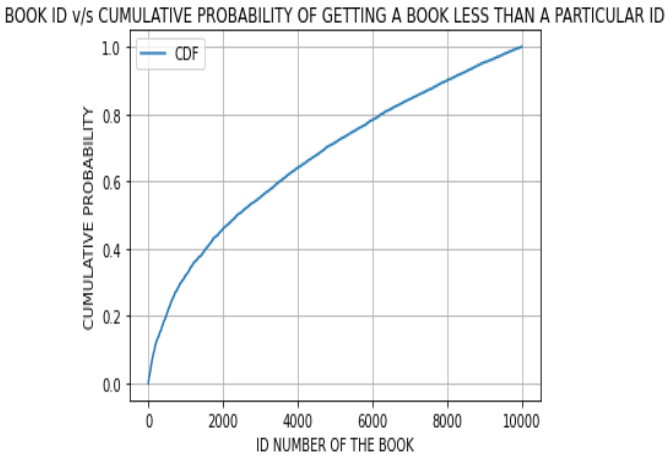
Stephen King, Nora Roberts, Dean Koontz, Terry Pratchett, Agatha Christie, Meg Cabot, James Patterson, David Baldacci, John Grisham, J.D. Robb, Laurell K. Hamilton, Janet Evanovich, Michael Connelly, John Sandford, Kristen Ashley, Tamora Pierce, Harlan Coben, Sherrilyn Kenyon, Patricia Cornwell, Sue Grafton, Jim Butcher, Anne Rice, Jodi Picoult, Charlaine Harris, Rick Riordan, Abbi Glines, R.A. Salvatore, Brandon Sanderson, J.R. Ward, Richelle Mead, Lee Child, Orson Scott Card, Terry Brooks, Douglas Preston, Lincoln Child, Jeffrey Archer, Nicholas Sparks, Mary Higgins Clark, C.S. Lewis, Isaac Asimov, Kurt Vonnegut Jr., Robin Hobb, Jennifer L. Armentrout, Dr. Seuss, Lisa Gardner, Sidney Sheldon, David Eddings, Carl Hiaasen, Anne McCaffrey, Daniel Silva, Ken Follett

6. **What is the trend of the book count against the book id? What is the visualisation that can provided based on that? What is the probability of picking a particular book when picking one at random from the library/store?**

A6. First, the csv file 'https://raw.githubusercontent.com/zygmuntz/goodbooks-10k/master/books.csv' was opened and converted to a dataframe using the pandas module. Then the columns of the dataframe, like books_ID and books_count, were converted into NumPy arrays. The total number of books was stored in a variable given by the sum(books_count). Then the first plot was plotted with the parameters of books_count against book_id, which gives the visualisation of the number of books against the books ID. For the probability part of the question, the number of copies of each book was divided by the total sum and plotted in the same way to give the required plot of probability. Then the number of books up to that ID which was obtained by numpy.cumsum was plotted against the book ID in order to give the cumulative distribution Function of picking a book from the library having the ID less than a particular number.

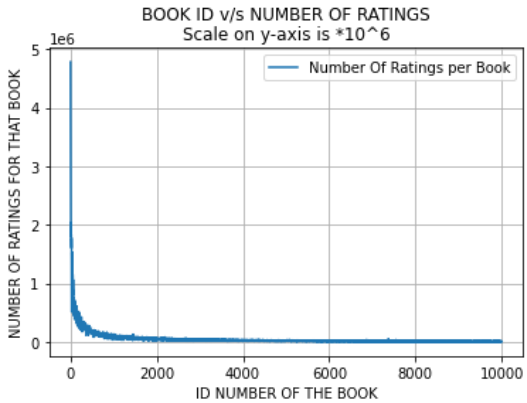




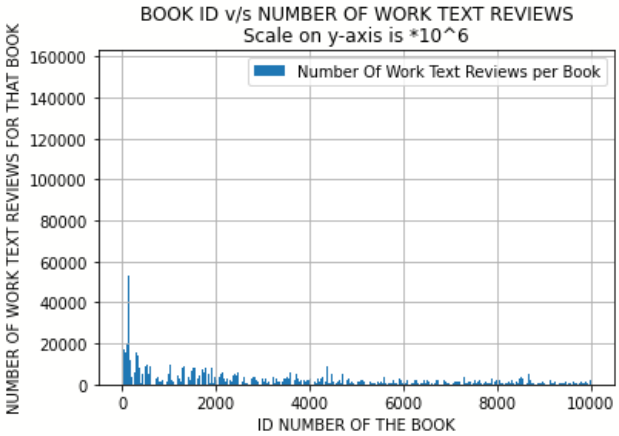


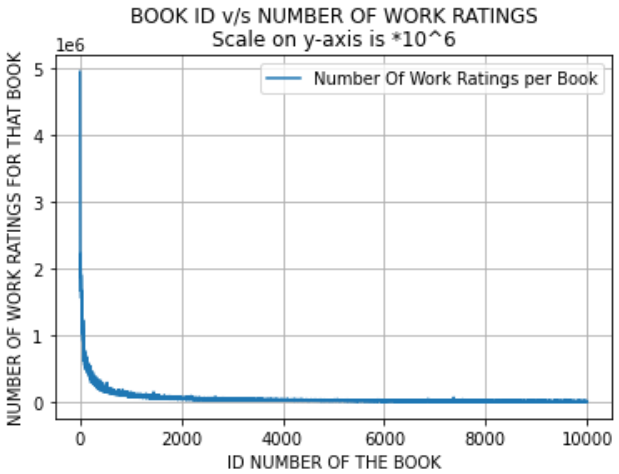
7. What analysis about the relative popularity of the books can be drawn from the data of ratings count, work ratings count and work text review count against the book ID?

A7. First, the csv file 'https://raw.githubusercontent.com/zygmuntz/goodbooks-10k/master/books.csv' was opened using the pandas module, and then it was converted to a dataframe. Then the column of ratings_count and the column of bookID were converted into NumPy arrays. Then these NumPy arrays were plotted against each other with appropriate axes to get



the plot of number of ratings against the book ID. Similar technique was also used for work_ratings and work_text_reviews.





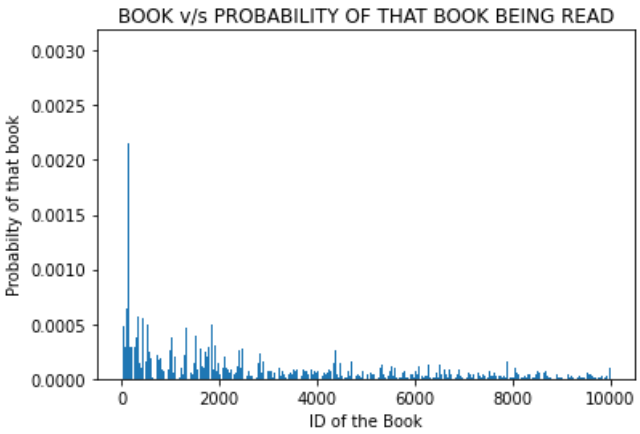
The observations that we can draw from the first graph is that the books with lower values of ID have the larger number of ratings; that is, they are very popularly read as the number of ratings increases as more people read the books. Number of work text reviews in the second graph and the number of work ratings in the third graph are also similar parameters that can be used to judge the popularity of books, but they are more focused toward the content of the book.

8. If we choose a book, what is the probability that the book is being by a particular reader? Explain it by visualisation.

A8. The method used in this is very similar to the first question and can be treated as a continuation of the first one. Only the difference was during the final plotting, the number of readers of a particular book was divided by the total number of readers in order to get the probability of a particular book being read. This may also be useful

parameter to find the impact of a book amongst the total subscriber community.

The general observation that can be drawn from this may be the fact that the readership of the books with smaller ID number is larger than those later. We can correlate the book ID with the actual title of the book later.



UNIQUENESS OF APPROACHES USED

The approaches used are unique in the sense that the libraries of Pandas, NumPy and matplotlib, along with the prior knowledge of Python, were used in a complementary manner in order to analyze the various parameters effectively. Various concepts of the course, such as basics of probability, probability distribution, cumulative distribution functions, smoothening the curve using kernel density estimation, etc., were also used by invoking various functions of the modules. On the whole, this analysis gives the various unseen perspectives of the dataset, including the ratings per user, languages, probability of getting a book from the book count, etc., which ensures the optimum use of the concepts of the course and the modules to be used in a synchronized way to effectively mine the data. *The most fascinating part that I feel is the analysis of user data (the rating given by the user against the user ID).*

SUMMARY AND CONCLUSIONS

The data has therefore been analyzed from various aspects such ratings from the perspective of the book as well as of the user, the average rating, the number of readers per book, the most famous titles, the analysis of the books against the year of publishing, analysis against the language of the book, most popular authors and key words in the title of the books, books count (the probability of a particular book being read by a given reader), analysis of the number of reviews of various types, etc. analyzed by taking into account the various concepts learnt in this course so far. The results are as given after the solution to the questions.

REFERENCES

[1] “NumPy User Guide.” NumPy user guide - NumPy v1.24 Manual. Accessed February 21, 2023. https://numpy.org/doc/stable/user/index.html

[2] “Matplotlib - Visualization with Python v3.7.0.” Matplotlib. Accessed February 21, 2023. https://matplotlib.org/

[3] “Pandas User Guide.” User Guide - pandas 1.5.3 documentation. Accessed February 21, 2023. https://pandas.pydata.org/docs/user_guide/index.html#user-guide

[4] “How to Calculate and Plot a Cumulative Distribution Function with Matplotlib in Python ?” How to calculate and plot a Cumulative Distribution function with Matplotlib in Python ? GeeksforGeeks, January 24, 2021. https://www.geeksforgeeks.org/how-to-calculate-and-plot-a-cumulative-distribution-function-with-matplotlib-in-python/

[5] “Pandas - Fixing Wrong Data.” Pandas - Cleaning Data. Accessed February 22, 2023. https://www.w3schools.com/python/pandas/pandas_cleaning_wrong_data.asp