Es114 Data Narrative

Assignment 1

Pradeep Kumar Meena

Roll no: 22110196

IIT Gandhinagar

I. Overview of the Dataset

The Goodreads-10k dataset is a collection of files that contains large amounts of information about books, user ratings, and tags. The dataset comprises 10,000 books, each with details such as the author, title, and publication year. There is also a file that contains reader ratings for each book, which can be used to analyse reading preferences and trends. The dataset provides information on tags, allowing researchers to explore popular themes and topics among readers. In addition, there is a file containing a list of books that users have marked as "to read," providing that which books the readers want to read in the future.

The Goodreads-10k dataset is valuable for conducting research on reading habits, preferences, and trends. With this data, researchers and data scientists of companies like Google Books etc. can better analyse user ratings and tags to understand what people like to read and why. They can also use the data to build recommendation systems that suggest books to readers based on their behaviour and interests. This can be useful for book retailers, publishers, and online platforms to provide personalised recommendations to their readers.

II. Scientific Questions or Hypotheses

A. If we randomly choose a book, the probability that a book has an average rating of 4 is greater than the probability book has average rating is 3.

Approach: By using the file "books.csv" I counted the number of books and found that, there is more number of books whose average rating is four then the number of books whose rating is 3.5 .

B. Find those ten books which most people want to read. And also find how many people want to read the top one common book.

Approach: First I read the file "to_read.csv" I calculated the number of occurrencess of every book_id in the column book_id and then by using a function I print the 10 largest occurred values and plot the bar graph of the occurrences.

C. Find five authors whose books are maximum in the given data set and verify your answer?

Approach: First, I read the file "books.csv" and calculated the number of occurrences of every author in the column authors and then I take the top five data and plot the bar graph using matplotlib.

D. The reader wants to search books by the author's name. Provide the data of those books which were written by the specified author.

Approach: First, I read the file "books.csv" and take the authors name from the reader as an input and then check if in the columns "authors" the author name is available then print all those books data whose author is as entered by the reader. If there are no author of the name entered by the reader then the code print that the book is not available in the library.

E. How does the publication year of books related to their average rating or number of ratings, and could this be used to inform decisions about when to release new books?

Approach: First, filters the original dataframe "b" to only include books with an original publication year greater than 1900. The filtered data is stored in a new dataframe called "b". Next by using the "groupby" function to group the books in the b dataframe by their original publication year. And then by using the "mean()" function is used to calculate the average rating of books published in each year. The resulting grouped data is stored in a new pandas series called "avg_ratings". At last plot the scatter plot and a line plot to visualize the data.

F. Trends of how many books were published in a year after 2000?

Approach: First we will create a another dataframe which will only contain the data of those books which are published after the 2000. And then by using "groupby()" function to group the books dataframe in the b dataframe by their original publication year, and the size() function is

used to count the number of books published in each year. The resulting grouped data is stored in a new dataframe called “grouped_df”. Next by using the bar() plot the bar graph.

III. Details of libraries and functions

Some of the libraries I used to work with dataset

A. *Pandas:-*

- Pandas is a python library used for working with data. It is widely used for data analysis and manipulation. It is very easy to work with pandas to visualize data.
- Pandas provides two primary data structures: Series and DataFrame. A Series is a one-dimensional labelled array capable of holding any data type, while a DataFrame is a two-dimensional labelled data structure with columns of potentially different types.

B. *Matplotlib:-*

- Matplotlib is also a library in python which is used to visualize data by using various types of plots
- In matplotlib we can plot bar graphs, histogram , pie chart , kde plots etc.

Some of the widely used functions are:-

A) read_csv(): It is a function of pandas library which is used to read a csv file.

B) unique(): It is used to get unique values of Series object.

C) values_count(): It is used to get the occurrence of any element of a series object.

D) nlargest() and idxmax(): It is used to get the n largest values from a series.

E) to_frame(): It is used to convert a set of values to the pandas dataframe.

F) plot(): It is used to plot the graph this function is available in both pandas and matplotlib. In matplotlib it gives us a line plot.

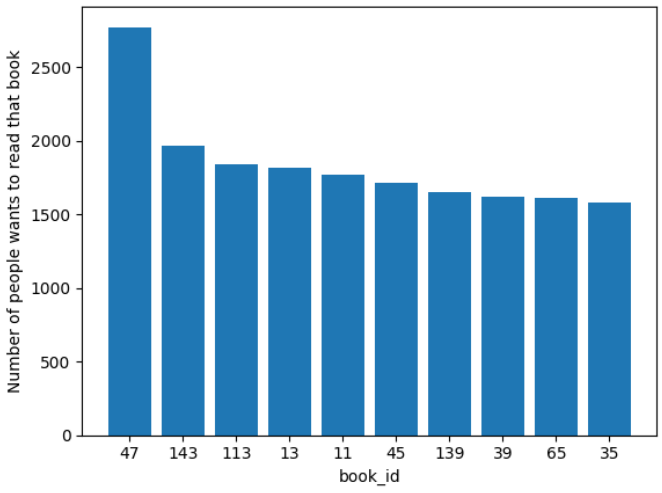
G) bar(): It is the function available in matplotlib to plot bar graph.

IV. Answers to the questions

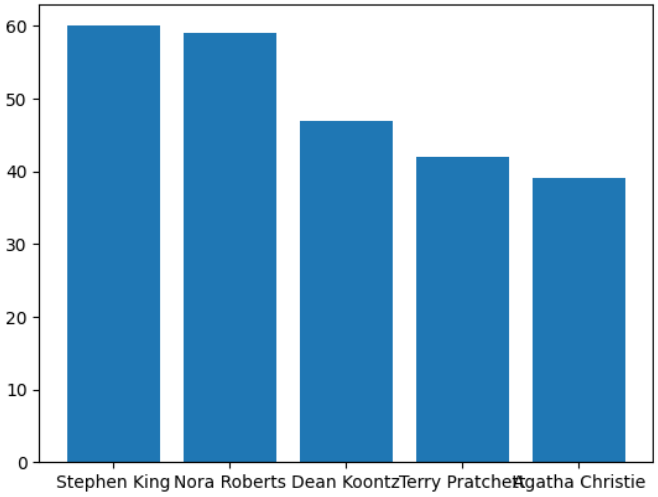
P(A) = favourable outcomes / Total outcome

A) Based on the given data from the “books.csv” file, we can test the hypothesis that the probability of selecting a book with an average rating of 4 is greater than the probability of selecting a book with an average rating of 3. To do this, we can calculate the probabilities of selecting a book with an average rating of 4 and a book with an average rating of 3, respectively. Then we can compare the two probabilities to determine which one is greater. According to the data, there are a total of “Total_books” unique books. If we count the number of books with an average rating of 4, we get books_avg_rating_4, and if we count the number of books with an average rating of 3, we get books_avg_rating_3. We can then calculate the probability of selecting a book with an average rating of 4 as probability_of_avg_rating_4 = books_avg_rating_4 / “Total_books”, and the probability of selecting a book with an average rating of 3 as probability_of_avg_rating_3 = books_avg_rating_3 / “Total_books”. If the probability of selecting a book with an average rating of 4 is greater than the probability of selecting a book with an average rating of 3, we can conclude that the hypothesis is true. Otherwise, the hypothesis is false. Therefore, by calculating the probabilities and comparing them, we can test the hypothesis and determine whether the probability of selecting a book with an average rating of 4 is greater than the probability of selecting a book with an average rating of 3 in the given dataset.

B) The first line of the code creates a Pandas Series object that counts the number of times each book_id appears in the tr DataFrame using the value_counts() function. The resulting counts are then sorted in descending order using the nlargest() function, which returns the top 10 most frequently occurring book IDs in the DataFrame. These top 10 book IDs, along with their corresponding counts, are then stored in a new DataFrame called h. Next, the X variable is created as a list of strings that contains the book IDs for the top 10 most frequently occurring books. The Y variable is a list that contains the corresponding number of people who want to read each of these books. These two lists are then used as inputs for the plt.bar() function, which generates a bar chart displaying the number of people who want to read each of the top 10 books. The plt.xlabel() and plt.ylabel() functions are used to label the x- and y-axes of the chart, respectively. Finally, the plt.show() function is used to display the chart.



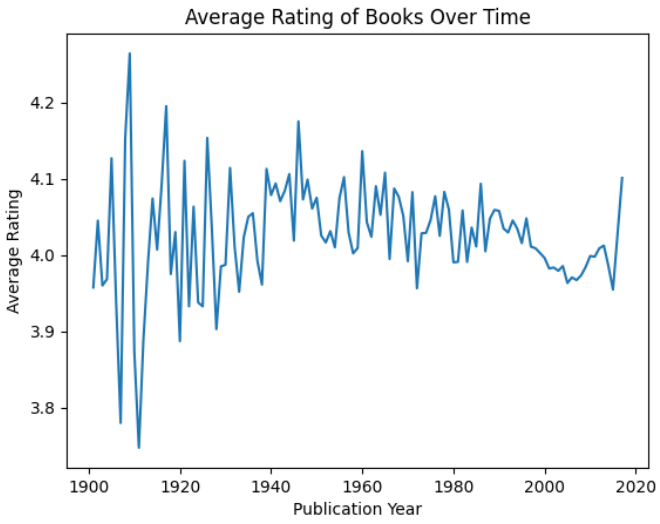
C) First create a Pandas Series object called author_occurances that counts the number of books written by each author in the b DataFrame using the value_counts() function. The resulting counts are then sorted in descending order using the nlargest() function, which returns the top 5 most frequently occurring authors in the DataFrame. Next, the df DataFrame is created to store the top 5 author counts along with their corresponding author names. The df.index function is used to display the author names as a list of strings. The occ list is then created to store the number of books written by each of the top 5 authors. Finally, the plt.bar() function is used to generate a bar chart displaying the number of books written by each of the top 5 authors. The x-axis of the chart displays the author names, while the y-axis displays the number of books written by each author. The plt.show() function is used to display the chart.

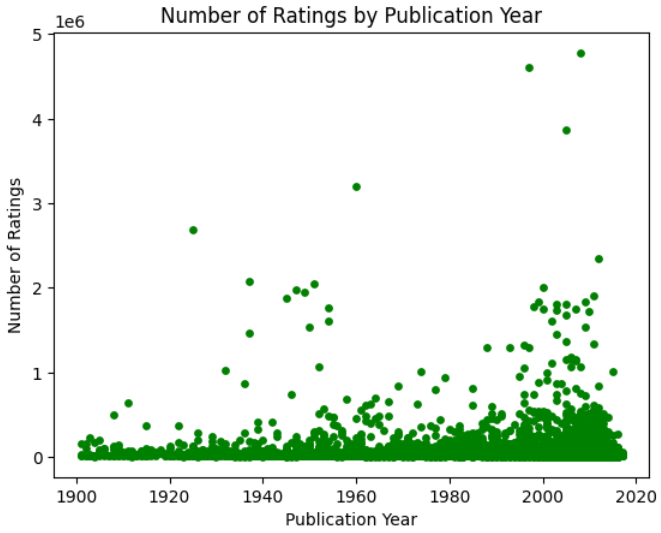


D) If you run the code and enter an author name when prompted, the resulting Available_books_data DataFrame will only include rows for books whose author name matches the input author name. For example, if you enter "j.k. rowling" as the author name, the resulting Available_books_data DataFrame will include all books written by J.K. Rowling that are in the original dataset. The DataFrame will have the same columns as the original dataset, and the number of rows will depend on how many books J.K. Rowling has in the dataset. If no books are found for the input author name, the message "This author's books is not available." will be printed. If the Available_books_data DataFrame is not empty after filtering by author name, it will be saved to a CSV file called 'my_data.csv' in the current working directory with index = False to avoid including the row index in the output file. And in the and the code will print the available data of those books.

E) First, the code filters the dataset to only include books that were published after 1900. Next, the groupby() method is used to group the filtered dataset by the 'original_publication_year' column, and the mean() method is called on the 'average_rating' column to calculate the average rating for each publication year. The resulting avg_ratings DataFrame is then plotted using matplotlib.pyplot.plot() to show the trend in average ratings over time. The second visualization plots the number of ratings by publication year using matplotlib.pyplot.scatter(). The 'original_publication_year' column is used for the x-axis, and the 'ratings_count' column is used for the y-axis. The resulting scatter plot shows the distribution of ratings counts over time, with larger points representing books with more ratings. Overall, these visualizations give insight into how average ratings and ratings counts have evolved over time in the Goodreads dataset.

Based on the visualizations created by the code, we can see that there is a general trend in the Goodreads dataset where books published more recently tend to have a higher average rating than books published in earlier years. The first visualization shows a gradual increase in average rating over time, with a steep rise in the late 1990s and early 2000s. The second visualization shows a similar trend in ratings counts over time, with a large increase in the number of ratings for books published in the last couple of decades.




Number of Ratings by Publication Year
1e6
5
4
3
2
1
0
Number of Ratings
1900
1920
1940
1960
1980
2000
2020
Publication Year