Data Narrative

Bhavik Patel, Btech Mechanical Engineering
Roll no.-22110047
Prof.- Shanmuga R , IIT Gandhinagar

I. OVERVIEW OF THE DATASETS

The Goodbooks-10k dataset is a collection of book ratings and metadata from the website Goodreads. The dataset contains information on approximately 10,000 books, including book ID's, titles, authors, descriptions, publication years and ratings on a scale of 1-5 stars.

NOTE: Each book may have many editions, goodreads_book _id and best_book_id generally point to the most popular edition of the given book, while goodreads work_id refers to the book in the abstract sense.

The dataset can be used for a variety of tasks, including book recommendations, text classification, and data analysis.

II. SCIENCTIFIC QUESTION/HYPOTHESESS

A. *Can we find the probability of different ratings from 1 to 5 for a newly launched book, given that the book is rated by same set of users in the given dataset?*

Approach:

1. First, I will take sum of each ratings from 1 to 5
2. Calculate total number of ratings given.
3. Then, I will find the probability for each rating by dividing the sums by total number of ratings.
4.

Equation: Probability=Total no. of favorable outcomes/Total no. of outcomes

Solution:

Fig. 1- Probability of ratings v/s Ratings

Fig. 2- Pie chart of % rating possibilities

Library used: Pandas, Matplotlib.pyplot and numpy(arange)

B. *What is the probability rated 4 or higher is a popular book (i.e. Rated by more than one lakh users)?*

Approach:

1. First, we will filter the dataframe only to include books rated 4.5 or higher with books.
2. We will filter again only to include books with more than 100,000 ratings.
3. Add a column named 'popular' having value 1 if the book is popular and 0 if not.
4. Then, we will calculate the mean of the 'popular' column, which will be nothing other than the required probability.

Solution:The probability comes out to 0.12

The below KDE(kernel density estimation) plot shows the distribution of average ratings for all books in the dataset.

Fig. 3- KDE of average rating

The KDE plot shows that the distribution of average ratings is approximately normal, with peak around 4 and flattens towards higher ratings.

Library used:Pandas, Matplotlib.pyplot→ KDE(kernel density estimation)

C. *Which is the most preferred/coomon language to write a book to reach a large audience?{This question is important for the authors who wants to reach large audience}*

Approach:

1. Make a series of language code.
2. Perform value count method on that series to get below series with number of books written in different languages.

Language	**No. of books**
eng	5557
en-US	1940
en-GB	215
en-CA	49
fre	22
spa	19
ger	13
ind	8
ara	6
jpn	6
por	5
pol	5
dan	3

ita	2
per	2
nor	1
vie	1
nl	1
tur	1
fil	1
swe	1
rum	1
rus	1

Table 1

The above table and two graphs below clearly depicts that **eng** is the most preferred language, with 5557 books written in that language.

Fig. 4- No. of books v/s languages plot

Fig. 5- Pie chart showing % of language

D. *What is the most common/famous tag by the users for the book they read? What is the probability that a randomly chosen book is given the most common tag?*

Approach:

1. We need to merge the book_tags csv and tags csv having tad_id as common column.
2. Make a series of tag counts by using value_counts method.
3. Then plot the bar graph of that series with top 15 Tag counts.

Fig. 6- Tag names vs Tag counts

From the above graph, we can state that '**to-read**' tag is most commonly used tag by majority of the users.

The probability that a randomly chosen book is given the most famous tag i.e. to-read tag is 0.01.

E. *Who are the top 10 Authors who wrote max no. of books? What is the probability that a randomly chosen book is written by one of those top 10 authors?*

Approach:

1. First, we will find no. of books written by each authors using value count methods.
2. Then, we will filter top 10 authors by slicing. The below Fig. shows who are the top 10 author and no. of books.

```
Nora Roberts          58
Stephen King          53
Terry Pratchett       40
Agatha Christie       38
Dean Koontz           34
James Patterson       33
J.D. Robb             33
Meg Cabot             32
David Baldacci        31
Laurell K. Hamilton   29
```

Fig. 7

3. The Fig. 7 shows that Nora Roberts has written most no. of books, followed by Stephen King
4. To find the probability that a randomly chosen book is written by one of the above top 10 authors, we used the below code.

```
authors=b['authors'].value_counts()
top_authors=authors[:10]
print(top_authors)
```

```
Prob=top_authors.sum()/authors.sum()
```

5. The probability we got was 0.05

F. *What are top 10 books suggested by most of the users to read ?*

Approach:

1. We need to use to_read.csv and book.csv in this question
2. First, merge to_read.csv and book.csv with common column book_id
3. Then perform value count method on 'original_title' column.
4. Then, filter out top 10 values.

Fig. 8

The above graph shows that 'The Book Thief' book is suggested by most of the users to read. So we conclude it to be a good book to read.

G. *Who are the top 10 users who have rated the maximum no. of books ?Which user ID has given the highest average rating among them ?*

Approach:

1. First, we will find to top 10 users using value count on user_id column in rating.csv file.
2. The top 10 users IDs giving maximum no. of ratings are *12874, 30944, 52036, 12381, 28158, 45554, 6630, 37834, 15604 and 7563*
3. To find average rating by those users, we will use a for loop and in that that loop, we will filter out the ratings given by each of them and then take the mean, which will give us the average rating of that user.

The above Fig. is of top 10 user IDs giving maximum no. of ratings v/s average rating given by those user. The graph clearly depicts that user ID *30944* has given highest average rating amongst the other top 10 user IDs.

III. DETAILS OF LIBRARIES

1. PANDAS : Pandas is a popular open-source Python library for data manipulation, analysis, and preparation. It provides fast and flexible data structures for structured data, such as tables, time series, and matrices, and offers tools for data cleaning, exploration, and visualization. The two primary data structures in pandas are Series and DataFrame.[2]

2. MATPLOTLIB: Matplotlib is a well-known open-source Python library for creating high-quality data visualizations. Line plots, scatter plots, bar charts, histograms, and other plot types are available. Matplotlib is built to work with NumPy arrays and can create static and interactive visualizations. Some major features of this library areeasy customization, supports multiple output format, etc.

3. NUMPY: NumPy is a widely used open-source Python library for numerical computation. It provides a quick and efficient way to work with numeric data arrays and matrices, as well as various tools for mathematical operations, random number generation, linear algebra, and more. Some key features are multidimensional array support, broadcasting, mathematical operations, and random number generation.

4. SEABORN: Seaborn is a popular open-source Python data visualization library. It is based on Matplotlib and offers a high-level interface for creating informative and appealing statistical graphics. Seaborn is built to work well with Pandas data structures and can easily handle large datasets.

IV. DETAILS OF FUNCTIONS

1. *read_csv*: The read_csv function is a method provided by the Pandas library in Python for reading and parsing data from a comma-separated values (CSV) file. It is a flexible function that can handle a wide range of input data formats and options.

2. *plt.plot:* The plt.plot function is a method provided by the Matplotlib library in Python for creating line plots. It is a simple and flexible function that can handle a wide range of input data formats and options. The plt.plot function takes one or more arrays of x and y values and creates a data line plot. It provides various options for customizing the appearance of the plot, such as setting the color, line style, marker style, and label.

3. *value_counts():*The value_counts() function is a Python method provided by the Pandas library for calculating the frequency of unique values in a Pandas Series. It returns a new Series object that contains the count of each unique value in the original Series.

4. *drop_duplicates():* The drop_duplicates() function is a method provided by the Pandas library in Python for removing duplicate rows from a data frame. It is a powerful tool for cleaning and preprocessing data and is commonly used in data analysis and exploration tasks.

5. *merge():* The merge() function is a method provided by the Pandas library in Python for combining two or more DataFrames based on a common set of columns. It is a powerful tool for merging and joining data from different sources and is commonly used in data analysis and exploration tasks.

V. REFERENCE LIST

1. Zygmuntz. “GitHub - Zygmuntz/Goodbooks-10k: Ten Thousand Books, Six Million Ratings.” GitHub, n.d. https://github.com/zygmuntz/goodbooks-10k

2. “Pandas Documentation — Pandas 1.5.3 Documentation,” n.d. https://pandas.pydata.org/docs/

3. “Matplotlib — Visualization with Python,” n.d. https://matplotlib.org/.

4. “NumPy Documentation — NumPy v1.24 Manual,” n.d. https://numpy.org/doc/stable/

5. “Seaborn: Statistical Data Visualization — Seaborn 0.12.2 Documentation,” n.d. https://seaborn.pydata.org