**PROJECT REPORT**

**Amazon Books Review**

Big Data Analytics

**Team Members**

Gireesh Bhavani Sankar Boddu (boddu2g)

Section 1: Overview

In this project, we analyzed a large dataset of Amazon book reviews and metadata, containing 3 million user reviews across 212,404 unique books. The objective was to clean, transform, and extract meaningful insights from the data. The following tasks were accomplished as part of this project:

We merged two separate datasets—one containing reviews and the other containing metadata—using the common Title column. This resulted in a combined dataset with information on both reviews and book details.We handled missing values by filling them in the review/text column with empty strings and removing rows with missing data in other critical columns.Duplicate rows were identified and removed, reducing the dataset size from 3 million to 225,968 rows.

We applied data transformations to clean up the authors and categories columns, removing unnecessary brackets and colons. A new word\_count column was added to count the words in the review/text column.We created a Spark session and used SQL queries to extract insights from the cleaned dataset.The queries were designed to find the most reviewed books, average review scores, book categories with the highest ratings, and more.

We also explored more complex queries for sentiment analysis and book recommendations based on categories and ratings.We used Matplotlib and Seaborn to create visualizations like bar charts, pie charts, and scatter plots to depict the distribution of books by genre, ratings count, and other key metrics.These visualizations helped to communicate the results of our analysis effectively.

Section 2: Tools Used

In this project, various tools were employed for data manipulation, analysis, and visualization.

* Pandas

Pandas is a popular data analysis library in Python, known for its ease of use in handling structured data like DataFrames. In this project, Pandas was used for To read the CSV files containing book reviews and metadata into Pandas DataFrames.The Pandas merge function was used to combine the two datasets on the Title column.Pandas was employed to identify and fill missing values, remove duplicate rows, and apply data transformations.

* PySpark

PySpark is the Python API for Apache Spark, designed for large-scale data processing and distributed computing. A Spark session was created to work with large datasets and perform SQL-based analysis.The cleaned Pandas DataFrame was converted to a PySpark DataFrame for efficient processing.PySpark allowed SQL queries to be run on the DataFrame to extract insights, such as identifying the most reviewed books, calculating average review scores, and finding the total ratings for each book.PySpark was used to save the cleaned and transformed data back to a CSV file for later use.

* Matplotlib and Seaborn

These are popular visualization libraries in Python, used for creating plots and charts. In this project, they were used for:

Bar Charts: To visualize the distribution of books by genre and other categorical data.

Pie Charts: To represent the proportion of highly rated books with over 4,000 ratings.

Scatter Plots: To explore the relationship between ratingsCount and review/score.

Histograms: To show the distribution of ratings count.

* Google Colab

Google Colab is a cloud-based platform for running Jupyter notebooks. It provides a flexible environment for coding, sharing, and collaboration. The project code was run in Google Colab, allowing for seamless integration with Google Drive and other cloud-based storage solutions.The datasets and cleaned data were stored in Google Drive, enabling easy access and sharing.Google Colab facilitated team collaboration, allowing multiple users to work on the same notebook.

Section 3: Data Cleaning

The data cleaning process is a critical step in preparing the dataset for analysis. Using the provided code, we merged, transformed, and cleaned the data to create a reliable dataset for further analysis. Here's what we observed during data cleaning and how it was addressed:

**Merging Datasets**

The rating and data datasets were merged on the **Title** column to combine book reviews and Books\_data. This merge operation provided a unified dataset, allowing us to analyze both reviews and book\_data details. After merging, we displayed the first few rows to verify the merge's success and check for discrepancies.

**Example: The first few rows of the merged dataset showed both review information (e.g., review/score, review/text) and book metadata (e.g., authors, categories), confirming that the merge was successful.**

**Handling Missing Values**

The dataset had several columns with missing values, requiring careful handling to maintain data integrity:

Filling Missing Values: Missing values in the **review/text** column were filled with empty strings to avoid issues during analysis.

Dropping Rows with Missing Data: Rows with missing values in other critical columns were dropped to avoid skewing results.

Observation: Before cleaning, the **ratingsCount** column had over 1.3 million missing values, and the Price column had over 2.5 million. This indicated significant gaps in the data that needed addressing.

Identifying and Removing Duplicates

Duplicate rows can lead to inaccurate analysis and inflated statistics. We identified duplicate rows in the dataset and removed them to ensure accurate results.

**Example: Duplicate rows were found in various parts of the dataset. Removing these duplicates reduced the dataset from 3 million rows to 225,968 rows.**

**Data Transformation**

Cleaning Authors and Categories: **Brackets** and **colons** were removed from the authors and categories columns, making the data more consistent. Missing author names were filled with **"Unknown."**

**Adding a Word Count**: A new word\_count column was added, counting the number of words in each review/text. This helped in further analysis, such as calculating the average word count for different book categories.

Observation: The addition of word\_count allowed us to analyze the length of reviews, providing insights into review detail and verbosity.

**Final Cleaned Dataset**

After cleaning, the final dataset consisted of 225,968 rows and 8 columns: Title, review/score, review/text, authors, categories, ratingsCount, Price, and word\_count. This cleaned dataset served as the foundation for further SQL-based analysis and visualization.

**Converting to PySpark DataFrame**

Once the data was cleaned, it was converted to a PySpark DataFrame. This conversion enabled further large-scale data operations and SQL-based analysis, as Spark is designed for efficient distributed processing.

The Pandas DataFrame, containing 225,968 rows and 8 columns, was converted to a PySpark DataFrame for continued analysis.The newly developed dataset, named spark\_df, was stored as a CSV file to ensure it could be reloaded and accessed in later stages of the project.

**Storage Location**: The cleaned dataset was saved to the following location in Google **Drive: /content/drive/MyDrive/project ITC686/cleaned\_merged.csv.**

“File Format: The data was stored in CSV format, which is compatible with various data processing tools and easy to import into different environments.”

Section 4: Analysis Using SQL Queries

The following SQL queries were executed to derive insights from the cleaned dataset:

* **Simple Queries**

# Query 1: List the top 10 most reviewed books

**query1 = """**

**SELECT Title, COUNT(\*) AS review\_count**

**FROM data\_cleaned**

**GROUP BY Title**

**ORDER BY review\_count DESC**

**LIMIT 10**

**"""**

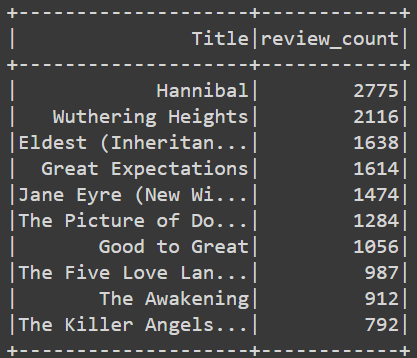
**result1 = spark.sql(query1)**

**result1.show()**

Purpose: This query aims to identify the ten books with the highest number of reviews.

Execution: The dataset is grouped by book titles, and the number of occurrences of each title is counted. The results are then sorted in descending order based on the count of reviews and limited to the top 10.

**Result**



The book **"Hannibal"** has the highest number of reviews with 2775 reviews, indicating it is highly discussed or read by users.

# Query 2: Find the average review score of books by each author

**query2\_with\_coalesce = """**

**SELECT authors, COALESCE(AVG(`review/score`), -1) AS avg\_review\_score**

**FROM data\_cleaned**

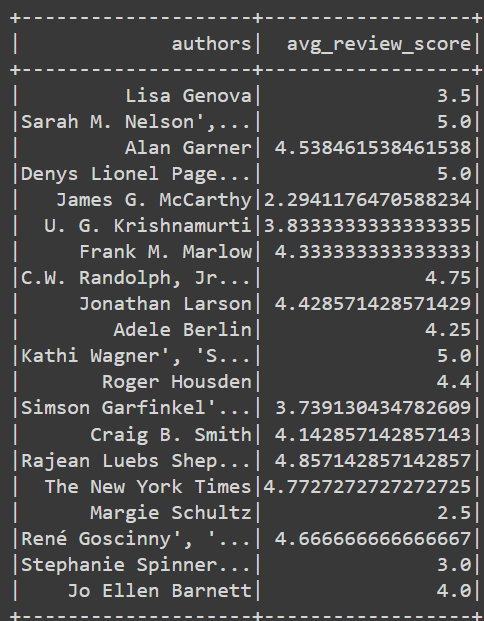
**GROUP BY authors**

**"""**

**result2\_with\_coalesce = spark.sql(query2\_with\_coalesce)**

**result2\_with\_coalesce.show()**

**Result**



For instance, books by "Denys Lionel" have an average review score of 5.0, indicating that they are generally well-received by readers

# Query 3: Identify the categories with the highest average ratings

**query3\_with\_coalesce = """**

**SELECT categories, COALESCE(AVG(`review/score`), -1) AS avg\_rating**

**FROM data\_cleaned**

**GROUP BY categories**

**ORDER BY avg\_rating DESC**

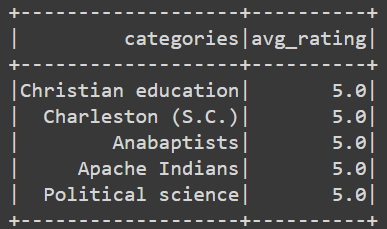
**LIMIT 5**

**"""**

**result3\_with\_coalesce = spark.sql(query3\_with\_coalesce)**

**result3\_with\_coalesce.show()**

**Result**



The query aimed to identify the book categories with the highest average ratings. The output shows the top five categories with an average rating of 5.0. This observation indicates that, within these categories, users generally rate the books highly.

* **Moderately Complex Queries**

# Query 4: Calculate the average word count of reviews for each category

**query4 = """**

**SELECT categories, AVG(word\_count) AS avg\_word\_count**

**FROM data\_cleaned**

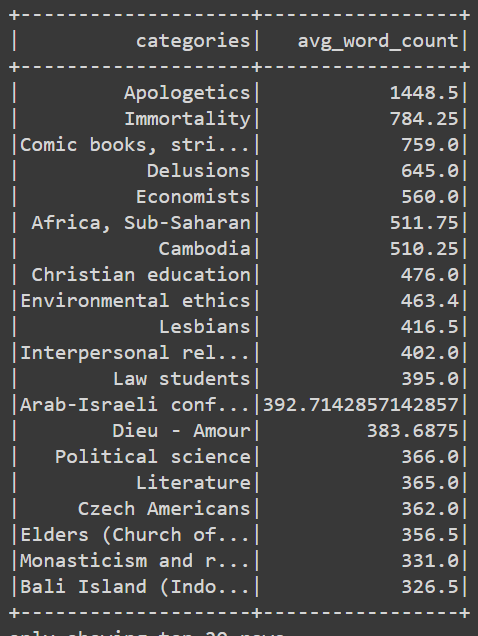
**GROUP BY categories**

**ORDER BY avg\_word\_count DESC**

**"""**

**result4 = spark.sql(query4)**

**result4.show()**

**Result**

For instance, books categorized under "Apologetics" have an average word count of 1448.5, indicating that reviews in this category tend to be more detailed or extensive.

# Query 5: Determine the total number of ratings for each book

**query5 = """**

**SELECT Title, SUM(ratingsCount) AS total\_ratings**

**FROM data\_cleaned**

**GROUP BY Title**

**ORDER BY total\_ratings DESC**

**"""**

**result5 = spark.sql(query5)**

**result5.show()**

**Result**

- For example, "Jane Eyre (New Wi..." has the highest total ratings count of 3,882,516, indicating it is widely read and rated by users.

Query to find the authors with the highest number of books, excluding "Unknown"

**query6\_corrected = """**

**SELECT authors, COUNT(\*) AS book\_count**

**FROM data\_cleaned**

**WHERE authors != 'Unknown'**

**GROUP BY authors**

**ORDER BY book\_count DESC**

**LIMIT 5**

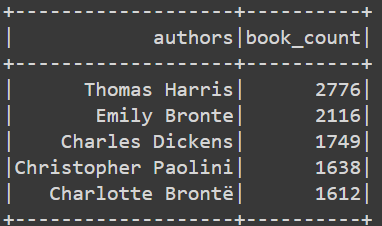
**"""**

**# Execute the corrected query**

**result6\_corrected = spark.sql(query6\_corrected)**

**result6\_corrected.show()**

**Result**



- For example, "Thomas Harris" has the highest number of books with 2776 titles attributed to them, indicating they have a significant presence in the dataset.

* **Complex Queries**

# Query 7: Perform sentiment analysis on review text to categorize reviews as positive, neutral, or negative

**query7 = """**

**SELECT Title,**

**CASE**

**WHEN LOWER(`review/text`) LIKE '%good%' OR LOWER(`review/text`) LIKE '%excellent%' THEN 'Positive'**

**WHEN LOWER(`review/text`) LIKE '%bad%' OR LOWER(`review/text`) LIKE '%poor%' THEN 'Negative'**

**ELSE 'Neutral'**

**END AS sentiment**

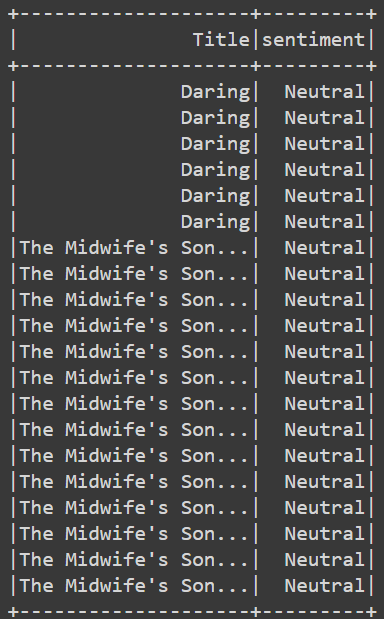
**FROM data\_cleaned**

**"""**

**result7 = spark.sql(query7)**

**result7.show()**

**Result**



- For instance, the sentiment analysis categorizes reviews for the book "The MidWife’s son " as mostly neutral, with a few categorized as positive.

# Query 8: Build a recommendation system to suggest similar books based on categories and ratings

**query8 = """**

**SELECT a.Title AS book\_title, b.Title AS recommended\_book**

**FROM data\_cleaned a**

**JOIN (**

**SELECT Title, categories**

**FROM data\_cleaned**

**GROUP BY Title, categories**

**) b**

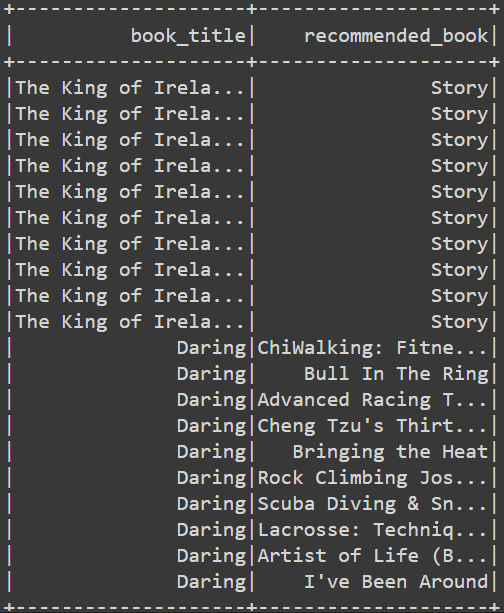
**ON a.categories = b.categories AND a.Title != b.Title**

**"""**

**result8 = spark.sql(query8)**

**result8.show()**

**Result**



For example, the book "Daring" is recommended alongside other titles based on similar categories and ratings criteria.

# Query to find the average review score and word count for each author, sorted by avg\_word\_count in descending order

**query9\_corrected = """**

**SELECT authors,**

**AVG(`review/score`) AS avg\_review\_score,**

**AVG(`word\_count`) AS avg\_word\_count**

**FROM data\_cleaned**

**GROUP BY authors**

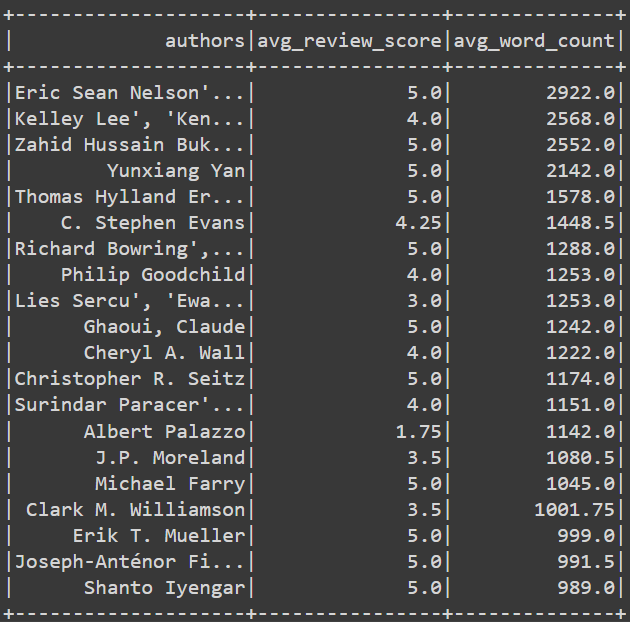
**ORDER BY avg\_word\_count DESC**

**"""**

**result9\_corrected = spark.sql(query9\_corrected)**

**result9\_corrected.show()**

**Result**

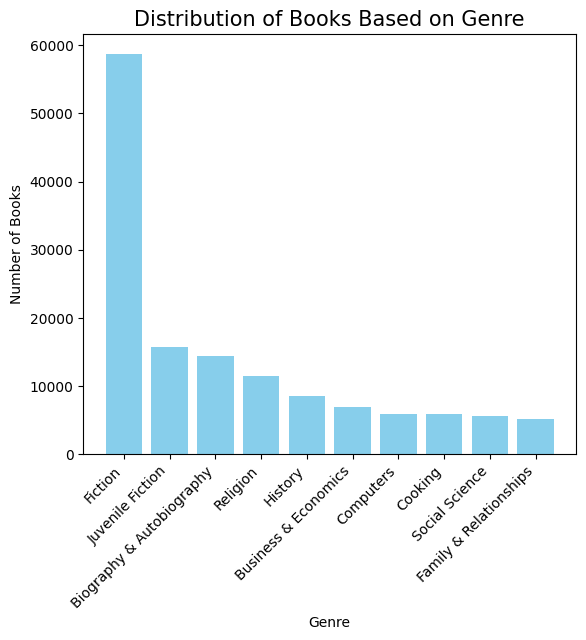


For example, books authored by "Eric Sean Nelson" have an average review score of 5.0 and an average word count of 2922.0

**Section 5: Exploratory Data Analysis (EDA)**

Exploratory Data Analysis refers to performing initial investigations on data so as to discover patterns, to test hypothesis and to check assumptions with the help of summary statistics and graphical representations.

1. **Distribution of Books Based on Genre (Bar Graph)**

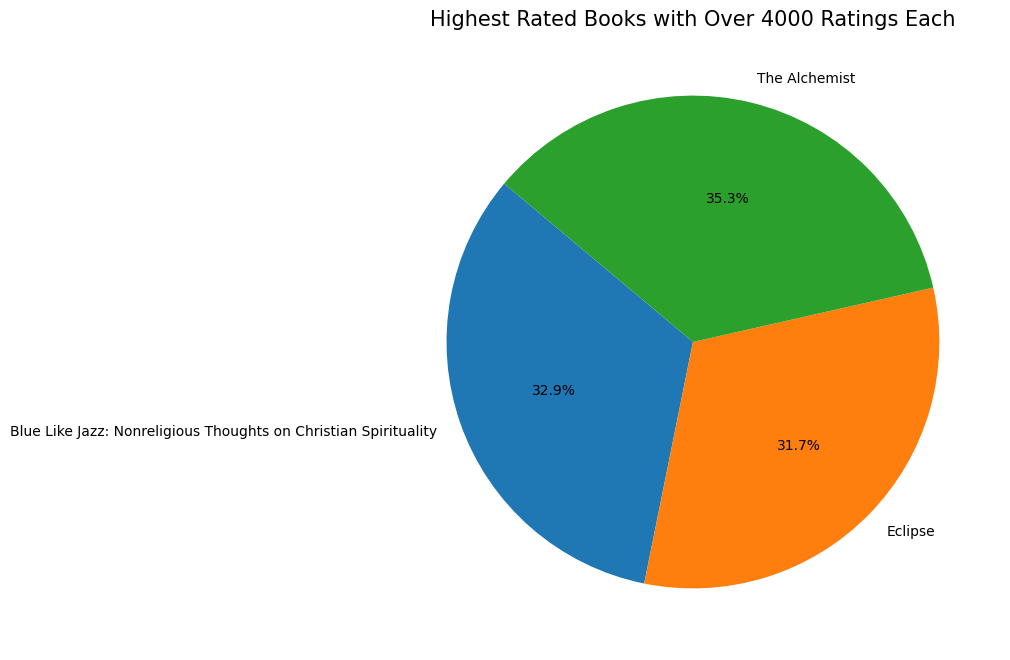


This bar graph illustrates the distribution of books across different genres, such as Fiction, Biography & Autobiography, Religion, etc.

- Example: The bar graph displays the number of books in the top 10 genres, with Fiction having the highest count.

- For example, the bar graph shows that the genre "Fiction" has the highest count of books, with a total of more than 50000 books. This indicates that Fiction is the most prevalent genre among the dataset.

1. **Highest Rated Books with Over 4000 Ratings Each (Pie Chart)**

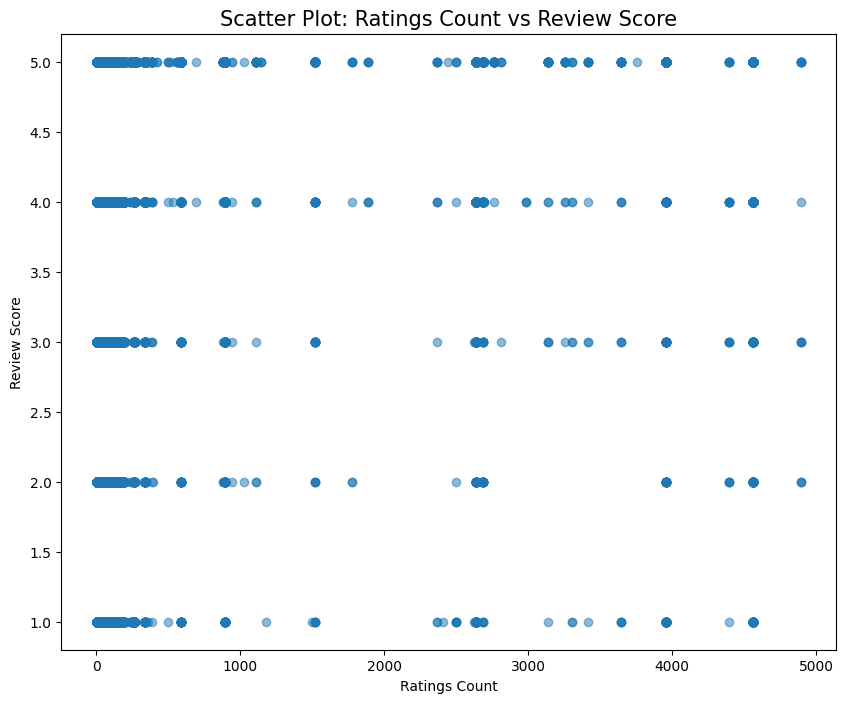


This pie chart visualizes the proportion of highly rated books, each with over 4000 ratings.

Example: The pie chart segments represent different highly rated books, with each segment showing its proportion relative to the total.

In the pie chart, one segment represents a highly rated book. For instance, the book “The Alchemist" occupies 35.3% of the chart, signifying that it is one of the highly rated books with over 4000 ratings out of the total.

**3)Scatter Plot: Ratings Count vs Review Score**

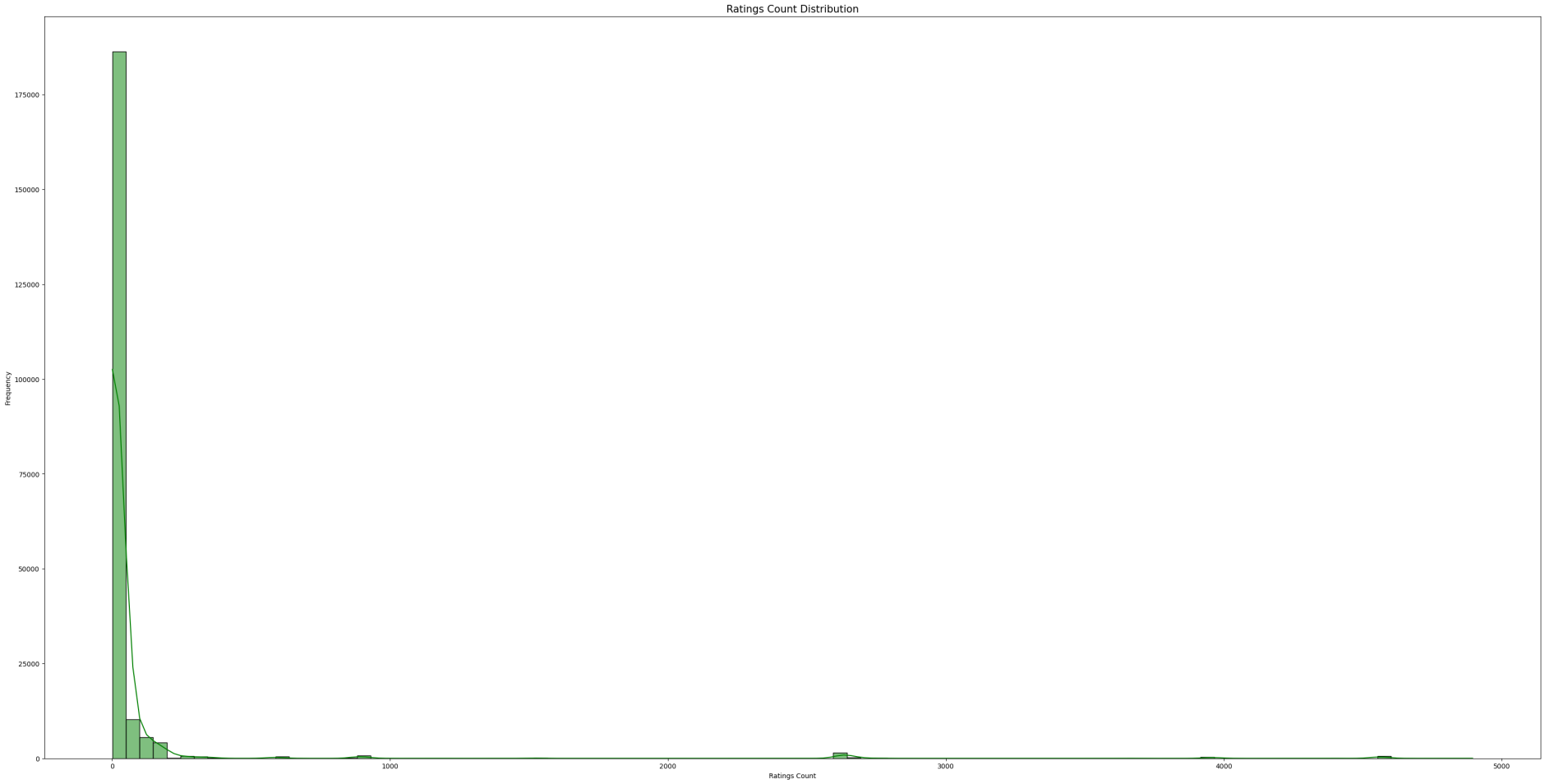


This scatter plot showcases the relationship between the number of ratings a book receives and its review score.

Example: Each point on the scatter plot represents a book, with its position indicating both the number of ratings it has received and its average review score.

Let's say a point on the scatter plot represents a book with 5000 ratings and a review score of 5.0. This point indicates that the book received a high number of ratings and a relatively positive review score.

**4)Ratings Count Distribution (Histogram)**



This histogram displays the distribution of ratings counts across all books.

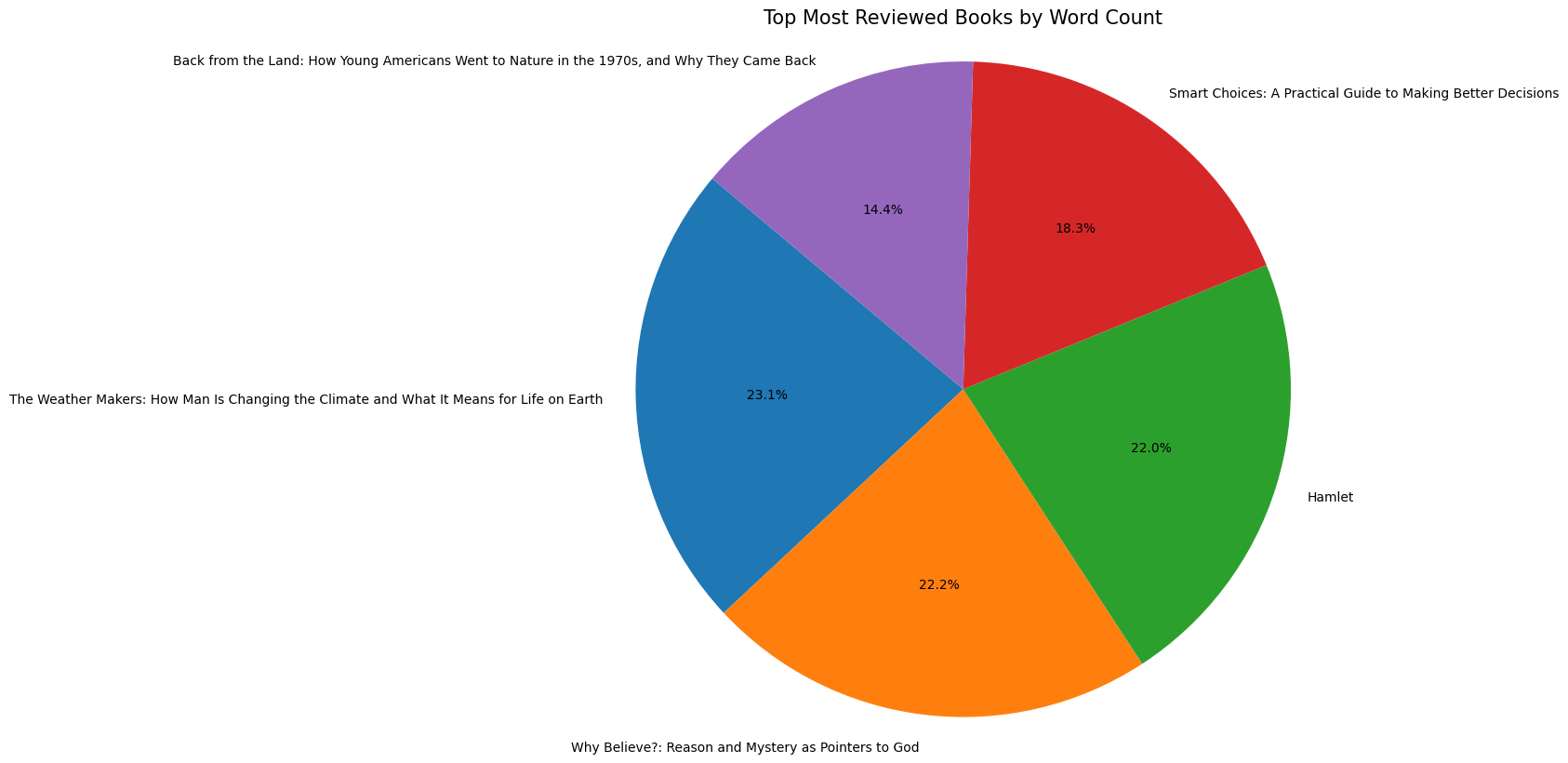
Example: The histogram depicts the frequency of books based on the number of ratings they have received, providing insights into the distribution pattern.

Suppose the histogram shows that the majority of books have received ratings counts between 1 and 1000. This suggests that most books in the dataset have garnered moderate to high levels of user engagement.

X-axis is frequency

Y-axis is ratings count

1. **Topmost Reviewed Books by Word Count (Pie Chart)**



This pie chart highlights the topmost reviewed books based on their word count.

Example: Each segment of the pie chart represents a book, with the size of the segment corresponding to the word count of the book.

For example, if The” weather makers: how man is changing …”occupies the largest segment in the pie chart, it implies that this book has the highest word count among the topmost reviewed books. The size of the segment indicates the magnitude of its word count relative to others.

**Section 6 Instructions how to run the source code.**To run the source code provided in this notebook, follow these steps:

1. Ensure you have access to the necessary datasets:

- 'Books\_rating.csv': Contains rating data for books.

- 'books\_data.csv': Contains additional data about books.

2. Upload the datasets to your Google Drive in a directory named 'project ITC686'.

3. Open this notebook in Google Colab by clicking on the provided link.

4. Install the required libraries if they are not already installed. You can install them by executing the following command in a code cell:

!pip install pandas pyspark matplotlib seaborn

5. Run each code cell in the notebook sequentially by clicking on the cell and either pressing Shift+Enter or clicking the Play button in the top-left corner of the cell.

6. The code will load the datasets, perform data cleaning and preprocessing, execute SQL queries using PySpark, and generate visualizations using matplotlib and seaborn.

7. Review the results displayed in the notebook, including tables and visualizations.

8. To reproduce or modify any part of the analysis, you can edit the code as needed and rerun the relevant code cells.

9. After running all the code cells, you can download the modified datasets or any generated visualizations if needed.

10. If you encounter any errors during execution, make sure to check the file paths specified in the code for reading and writing data, and ensure that the datasets are correctly uploaded to Google Drive.

**Conclusion**

* The combined insights from EDA and SQL queries provide a comprehensive understanding of the dataset.
* Fiction remains the dominant genre, while highly rated books like "The Alchemist" enjoy enduring popularity.
* The positive correlation between ratings count and review score suggests that well-reviewed books tend to attract more ratings.
* Authors' reception varies, as seen from the average review scores, and certain categories receive higher average ratings than others.
* Detailed feedback or discussions are evident for books with higher word counts in reviews, reflecting readers' engagement.
* The total number of ratings indicates the popularity of certain books, influencing their overall reception and standing within the dataset.

**References:**

* McKinney, W. (2017). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython (2nd ed.). O'Reilly Media.
* Wickham, H., & Grolemund, G. (2017): Import, Tidy, Transform, Visualize, and Model Data. O'Reilly Media.
* VanderPlas, J. (2016). Python Data Science Handbook: Essential Tools for Working with Data. O'Reilly Media.