

SENTIMENT ANALYSIS OF AMAZON BOOK REVIEW

CAPSTONE REPORT SUBMITTED TO THE BHARATHIAR UNIVERSITY

FOR THE AWARD OF THE DEGREE OF

MASTER OF BUSINESS ADMINISTRATION

By

SATHISH KUMAR. S

1P22MB014

Under the Guidance of

Dr. Suganya, Ph.D.

Associate Professor



RVS COLLEGE OF ARTS AND SCIENCE (AUTONOMOUS), SULUR

DEPARTMENT OF MANAGEMENT STUDIES - PG

Coimbatore – 641 402

Tamil Nadu, INDIA

March 2024

CERTIFICATE

This is to certify that the capstone report, entitled “**SENTIMENT ANALYSIS OF AMAZON BOOK REVIEW**”, submitted to the Bharathiar University, in partial fulfilment of the requirements for the award of the **DEGREE OF MASTER OF BUSINESS ADMINISTRATION**, is a record of original work done by **Mr. SATHISH KUMAR. S** during the period January 2024 – March 2024 of his capstone project in RVS College of Arts and Science (Autonomous), Department of Management Studies – PG, Sulur, Coimbatore - 641402, under my supervision and guidance and the capstone report has not formed the basis for the award of any Degree / Diploma / Associateship / Fellowship or other similar title of any candidate of any University.

Date: 28-03-2024

Director

Signature of the Guide

Date of Viva-voce held on _____.

Internal Examiner

External Examiner

DECLARATION

I, **SATHISH KUMAR. S**, hereby declare that the capstone, entitled “**SENTIMENT ANALYSIS OF AMAZON BOOK REVIEW**”, submitted to the Bharathiar University, in partial fulfilment of the requirements for the award of the **DEGREE OF MASTER OF BUSINESS ADMINISTRATION** is a record of original and independent research work done by me during the period January 2024 – March 2024, under the supervision and guidance of **Dr. Suganya**, Associate Professor. RVS College of Arts and Science (Autonomous), Department of Management Studies – PG, Sulur, Coimbatore – 641 402 and it has not formed the basis for the award of any other Degree / Diploma / Associateship / Fellowship or other similar title to any candidate of any University.

Date: 28-03-2024

Signature of the Candidate

ACKNOWLEDGEMENT

The successful completion of this capstone report would not have been possible without the support and assistance of many individuals and organization. I would like to take this opportunity to offer my earnest admiration to each and every one of them.

First and foremost, I thank Almighty and my parents for having bestowed their greatest blessing on me to complete this capstone successfully.

I express my indebtedness and gratefulness to **Dr. THULASIVELU K, Director**, RVS College of Arts and Science (Autonomous), Department of Management Studies – PG, Sulur, Coimbatore for his continuous support and encouragement and permission to pursue my capstone.

I would like to express my deep and sincere gratitude to my institution guide. **Dr. Suganya – Associate Professor**, RVS College of Arts and Science (Autonomous), Department of Management Studies – PG, Sulur, Coimbatore for her valuable advice, timely encouragement and educative guidance for completing the capstone successfully and for the instructions for preparing this report standard as per the norms and values.

I express my deep sense of gratitude to **Mr. Duke, Manager – Corporate Connect**, RVS College of Arts and Science (Autonomous), Department of Management Studies – PG, Sulur, Coimbatore for helping me by all means and ways for the successful completion of my capstone.

My sincere gratitude to **KAGGLE** for unremitting support rendered to complete my capstone successfully.

I perceive this opportunity as a big milestone in my career development. I will strive to use those gained skills and knowledge in the best possible way, and I will continue to work on their improvement, in order to attain desired career objectives.

ABSTRACT

This capstone project report provides an in-depth sentiment analysis of Amazon book reviews, employing a range of machine learning models. These include Logistic Regression, K-Nearest Neighbors (KNN), Naive Bayes, Decision Trees, Random Forest, Gradient Boosting Machine (GBM), and Artificial Neural Networks (ANN).

The effectiveness of these models was assessed using four crucial metrics: Accuracy, Precision, Recall, and F1 Score. The study revealed that the KNN model surpassed the others in performance, while the ANN model demonstrated the least effectiveness.

These results emphasize the significance of selecting the appropriate model for sentiment analysis tasks and evaluating its performance using multiple metrics. While accuracy is a vital measure, precision, recall, and the F1 score offer a more holistic view of a model's performance.

The insights gained from this analysis can guide business strategies, enhance customer satisfaction, and lead to product improvements. The study also identifies potential areas for future exploration, such as optimizing these models, investigating other models, and employing more complex or ensemble methods for potentially superior results.

This research serves as a stepping stone towards more sophisticated sentiment analysis tasks, fostering more informed and data-driven decision-making processes in the future.

CONTENT

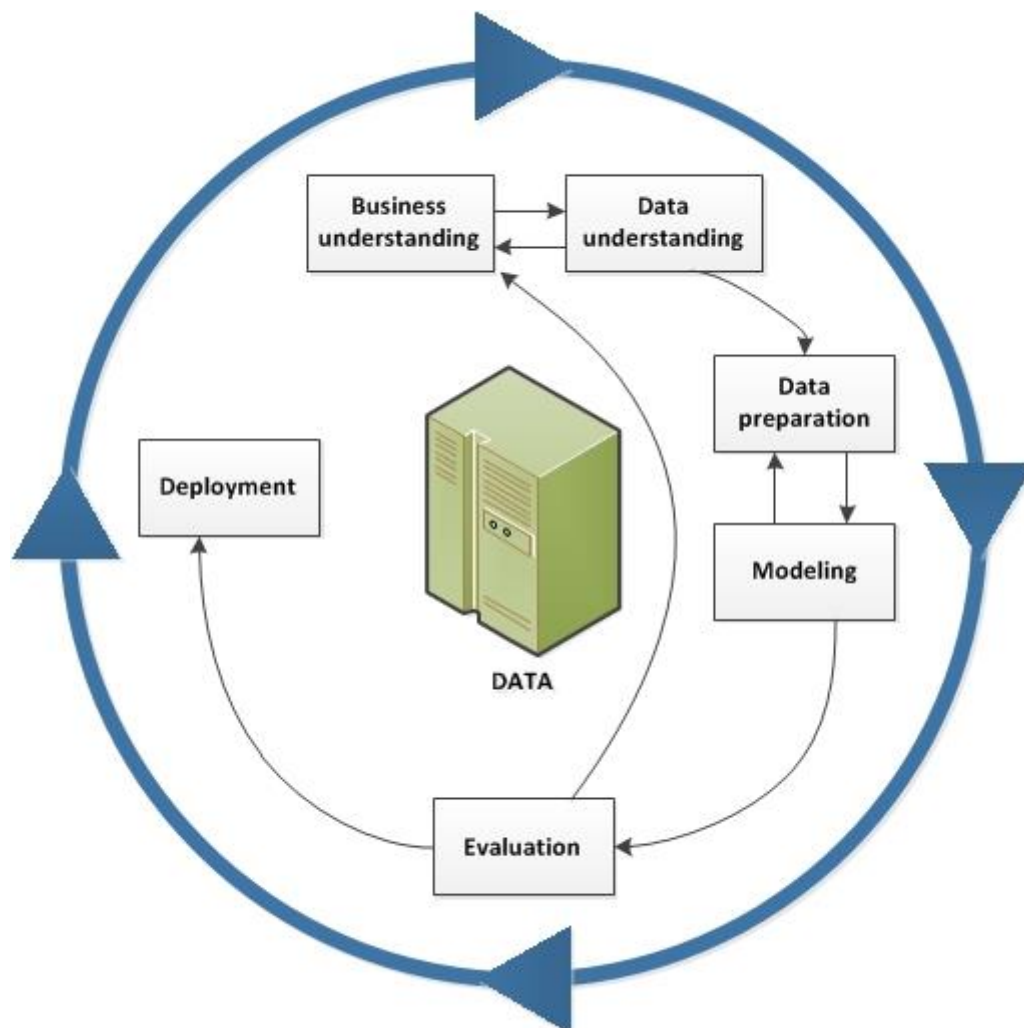
<i>Chapter No.</i>	<i>Title</i>	<i>Page No.</i>
<i>1</i>	Introduction	1
<i>2</i>	Tools for Assessment	3
<i>3</i>	Business Understanding	4
<i>4</i>	Data Understanding	6
<i>5</i>	Data Preparation	13
<i>6</i>	Feature Engineering	14
<i>7</i>	Exploratory Data Analysis (EDA)	16
<i>8</i>	Text Preprocessing	22
<i>9</i>	Sentiment Analysis	24
<i>10</i>	Clustering	27
<i>11</i>	Data Balancing	32
<i>12</i>	Model Building & Evaluation	33
<i>13</i>	Conclusion	49
<i>14</i>	References	50

Introduction

CRISP-DM, which stands for Cross-Industry Standard Process for Data Mining, is an industry-proven way to guide your data mining efforts. As a methodology, it includes descriptions of the typical phases of a project, the tasks involved with each phase, and an explanation of the relationships between these tasks. As a process model, CRISP-DM provides an overview of the data mining life cycle.

CRISP-DM breaks the process of data mining into six major phases:

- Business Understanding
- Data Understanding
- Data Preparation
- Modeling
- Evaluation
- Deployment



The life cycle model consists of six phases with arrows indicating the most important and frequent dependencies between phases. The sequence of the phases is not strict. In fact, most projects move back and forth between phases as necessary.

The arrows in the process diagram indicate the most important and frequent dependencies between phases. The outer circle in the diagram symbolizes the cyclic nature of data mining itself. A data mining process continues after a solution has been deployed. The lessons learned during the process can trigger new, often more focused business questions, and subsequent data mining processes will benefit from the experiences of previous ones.

The CRISP-DM model is flexible and can be customized easily. For example, if your organization aims to detect money laundering, it is likely that will sift through large amounts of data without a specific modeling goal. Instead of modeling, your work will focus on data exploration and visualization to uncover suspicious patterns in financial data. CRISP-DM allows us to create a data mining model that fits your particular needs.

In such a situation, the modeling, evaluation, and deployment phases might be less relevant than the data understanding and preparation phases. However, it is still important to consider some of the questions raised during these later phases for long-term planning and future data mining goals.

TOOLS FOR ASSESSMENT

In the ever-evolving landscape of data analytics, appropriate tools and technologies are crucial in converting unprocessed data into valuable insights. This article presents an array of popular tools and technologies that equip data analysts and data scientists to derive significant knowledge from extensive datasets. Let's dive into the realms of Python, R, Tableau, and Power BI, and investigate their individual contributions to the data analytics infrastructure. Here, in this project Python & Power BI is used for assessment.

Python:

Python is a high-level, object-oriented programming language that is interpreted and has dynamic semantics. Its built-in data structures are high-level, and when combined with dynamic typing and binding, it becomes an ideal choice for Rapid Application Development. It can also be used as a scripting or glue language to connect existing components. Python's syntax is simple and easy to learn, emphasizing readability, which in turn reduces the cost of maintaining the program. Python supports modules and packages, promoting code reuse and program modularity. The Python interpreter and its extensive standard library are freely available in source or binary form for all major platforms. Here are some key libraries and their uses:

Pandas: Used for data manipulation and analysis.

NumPy: Used for numerical computing and array operations.

Scikit-learn: Used for machine learning tasks, such as model training, evaluation, and prediction.

Matplotlib and Seaborn: Used for data visualization and exploratory analysis.

NLTK (Natural Language Toolkit): Provides easy-to-use interfaces to over 50 corpora and lexical resources, including WordNet. It is commonly used for text processing tasks like tokenization, stemming, tagging, parsing, and more.

TensorFlow / Keras: These libraries are used for constructing and training deep learning models, including neural networks for text classification tasks, such as fake news detection.

BUSINESS UNDERSTANDING

In the context of business, a capstone project often involves solving a real-world business problem or answering a complex business question. This could involve conducting a market analysis, developing a business plan, or analyzing the strategy of a successful business.

The business understanding phase of a capstone project involves identifying the business problem or question, understanding the needs of the project, and defining the scope of the project. This phase is crucial as it sets the direction for the entire project.

This capstone project involving Amazon book reviews, the business understanding phase might involve identifying key business questions such as “What factors influence customer ratings?” or “How do reviews impact sales?”. That would then need to understand the data available (e.g., review text, rating, book details), and define the scope of your analysis.

The Amazon Book Reviews project is an analysis of book reviews on Amazon, aiming to gain insights into readers’ behavior and preferences. The project involves exploring trends and performing sentiment analysis on the reviews.

This project can provide valuable insights for publishers, authors, and even readers. For instance, publishers can understand what kind of books are well-received by the readers, authors can get feedback on their work, and readers can get recommendations based on the analysis.

The Amazon Book Reviews dataset on Kaggle presents several business problems and opportunities. Here are a few examples:

Sentiment Analysis: The reviews can be analyzed to determine the sentiment (positive, negative, neutral) of the customers towards different books. This can help in understanding customer preferences and can guide marketing and sales strategies.

Recommendation Systems: The dataset can be used to build a book recommendation system. By analyzing the reviews and ratings, the system can suggest books that a customer is likely to enjoy, based on their previous ratings and the ratings of other customers with similar tastes.

Trend Analysis: By analyzing the ratings and reviews over time, businesses can identify trends in customer preferences. For example, they can identify which genres or authors are becoming more popular.

Author Feedback: Authors can use the feedback in the reviews to improve their future work. Negative reviews can provide constructive criticism, while positive reviews can highlight the aspects that readers enjoy.

Inventory Management: Retailers can use the ratings and reviews to decide which books to stock. Books with higher ratings and positive reviews are likely to be more popular and sell better

In summary, the business understanding for a capstone project sets the foundation for the project and guides the subsequent phases of data understanding, data preparation, modeling, evaluation, and deployment. The Amazon Book Reviews dataset on Kaggle provides a wealth of opportunities for businesses to understand their customers better and make data driven decisions

DATA UNDERSTANDING

Amazon book reviews are a valuable resource for both consumers and businesses. They provide insights into the quality of books and authors, based on the opinions and ratings provided by users.

For consumers, these reviews can guide purchasing decisions by providing information about the content, writing style, and overall quality of a book. They can also provide insights into whether a book is suitable for a particular audience or meets specific needs.

For businesses, particularly authors and publishers, these reviews can provide feedback on what readers enjoyed or disliked about a book. This can inform future writing, publishing, and marketing decisions. For example, if a book receives many positive reviews, it might be worth investing in more marketing or planning a sequel. Conversely, if a book receives many negative reviews, it might be worth revisiting the content or marketing strategy.

Moreover, businesses can also use sentiment analysis techniques on these reviews to gain deeper insights. Sentiment analysis can help determine the overall sentiment (positive, negative, neutral) of the reviews, identify common themes or issues, and even predict future sales trends.

For instance, the book “Understanding Business” has received positive reviews on Amazon, with customers appreciating its up-to-date content, easy-to-read layout, and comprehensive coverage of business concepts. Such feedback can be invaluable for authors and publishers in understanding what works well and what can be improved in future editions.

The data understanding of Amazon book reviews encompasses several important aspects for both authors and consumers. Amazon book reviews play a crucial role in the success of a book and can influence potential readers' purchasing decisions. Here are key elements of the data understanding of Amazon book reviews:

1. Consumer Influence:

- **Purchase Decisions:** Positive reviews can significantly impact a book's sales by influencing potential readers to make a purchase.

- **Trust and Credibility:** Consumers often rely on reviews to gauge the quality and relevance of a book. Higher review counts and positive feedback contribute to the credibility of the book.

2. Author Reputation:

- **Building Credibility:** Authors benefit from positive reviews as it helps establish their credibility in the literary world.

- **Author-Reader Interaction:** Authors can engage with readers through the review section, responding to feedback and fostering a sense of community.

3. Search Ranking and Visibility:

- **Algorithmic Influence:** Amazon's search algorithm takes into account factors like review quantity and rating to determine a book's ranking. Higher rankings increase a book's visibility on the platform.

- **Keyword Relevance:** Reviews often contain keywords related to the book's content, contributing to its discoverability.

4. Feedback for Improvement:

- **Insights for Authors:** Reviews provide valuable insights into readers' perspectives, helping authors understand what worked well and areas for improvement in their writing.

- **Iterative Process:** Authors can use feedback to refine future works and better meet the expectations of their audience.

5. Marketing and Promotion:

- **User-Generated Content:** Positive reviews can serve as user-generated content for marketing purposes. Authors and publishers may quote favorable reviews in promotional materials.

- **Social Proof:** Reviews act as a form of social proof, indicating to potential readers that others have found value in the book.

6. Mitigating Negative Impact:

- **Handling Negative Reviews:** Authors and publishers need to manage negative reviews professionally. Addressing concerns and learning from constructive criticism can help mitigate the impact on the book's reputation.

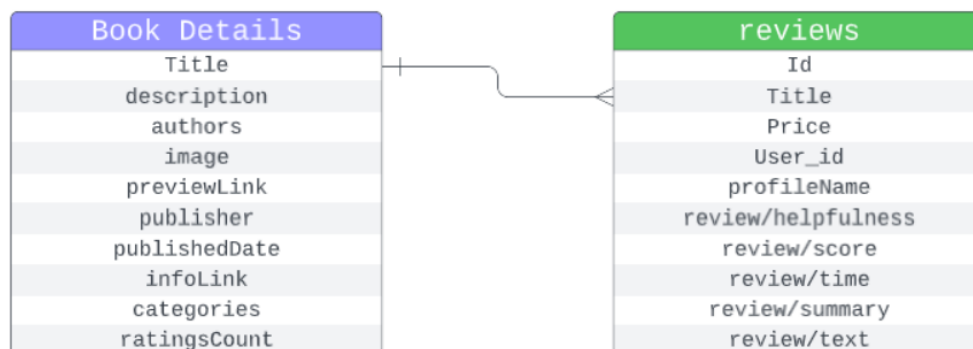
7. Platform Engagement:

- **Encouraging Reviews:** Authors often encourage readers to leave reviews by including calls-to-action in their books, newsletters, or social media.

- **Engagement Strategies:** Engaging with readers on social media and maintaining an active online presence can foster a supportive community and encourage reviews.

Data Source:

This project deals with two datasets named, Book data and Book rating data where the general meta data of book is listed in Book data and the ratings for each book are listed in the Book rating data.



Book Data:

The Books Details file contains details information about 212404 unique books it file is built by using [google books API](#) to get details information about books it rated in the first file and this file contains

Features	Description
Title	Book Title
Describe	decription of book
authors	Neme of book authors
image	url for book cover
previewLink	link to access this book on google Books
publisher	Name of the publisheer
publishedDate	the date of publish
infoLink	link to get more information about the book on google books
categories	genres of books
ratingsCount	averaging rating for book

Observations:

```
[2]: data = pd.read_csv("D://MBA//SEM 4//Capstone Project & Viva Voce//Dataset//Amazon Book Reviews//books_data.csv//books_data.csv")
data.head()
```

	Title	description	authors	image	previewLink	publisher	publishedDate	infoLink	cate
0	Its Only Art If Its Well Hung!	NaN	['Julie Strain']	http://books.google.com/books/content?id=DykPA...	http://books.google.nl/books?id=DykPAAACAAJ&d...	NaN	1996	http://books.google.nl/books?id=DykPAAACAAJ&d...	['Coi C N
1	Dr. Seuss: American Icon	Philip Nel takes a fascinating look into the k...	['Philip Nel']	http://books.google.com/books/content?id=ljvHQ...	http://books.google.nl/books?id=ljvHQsCn_pgC&p...	A&C Black	2005-01-01	http://books.google.nl/books?id=ljvHQsCn_pgC&d...	['Biogr Autobiogr
2	Wonderful Worship in Smaller Churches	This resource includes twelve principles in un...	['David R. Ray']	http://books.google.com/books/content?id=2tsDA...	http://books.google.nl/books?id=2tsDAAACAAJ&d...	NaN	2000	http://books.google.nl/books?id=2tsDAAACAAJ&d...	['Rel
3	Whispers of the Wicked Saints	Julia Thomas finds her life spinning out of co...	['Veronica Haddon']	http://books.google.com/books/content?id=aRSIg...	http://books.google.nl/books?id=aRSIgJlq6JwC&d...	iUniverse	2005-02	http://books.google.nl/books?id=aRSIgJlq6JwC&d...	['F
4	Nation Dance: Religion, Identity and Cultural ...	NaN	['Edward Long']	NaN	http://books.google.nl/books?id=399SPgAACAAJ&d...	NaN	2003-03-01	http://books.google.nl/books?id=399SPgAACAAJ&d...	

Information:

```
[4]: data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 212404 entries, 0 to 212403
Data columns (total 10 columns):
#   Column          Non-Null Count  Dtype
---  ---
0    Title           212403 non-null object
1    description      143962 non-null object
2    authors         180991 non-null object
3    image           160329 non-null object
4    previewLink     188568 non-null object
5    publisher       136518 non-null object
6    publishedDate   187099 non-null object
7    infoLink        188568 non-null object
8    categories      171205 non-null object
9    ratingsCount    49752 non-null  float64
dtypes: float64(1), object(9)
memory usage: 16.2+ MB
```


Book Rating:

The file** reviews** file contain feedback about 3M user on 212404 unique books the data set is part of the Amazon review Dataset it contains product reviews and metadata from Amazon, including 142.8 million reviews spanning May 1996 - July 2014. and this file has these attributes

Features	Description
id	The Id of Book
Title	Book Title
Price	The price of Book
User_id	Id of the user who rates the book
profileName	Name of the user who rates the book
review/helpfulness	helpfulness rating of the review, e.g. 2/3
review/score	rating from 0 to 5 for the book
review/time	time of given the review
review/summary	the summary of a text review
review/text	the full text of a review

Observation:

```
[16]: rating = pd.read_csv("D://MBA//SEM 4//Capstone Project & Viva Voce//Dataset//Amazon Book Reviews//Books_rating.csv//Books_rating.csv")
rating.head()
```

```
[16]:
```

	Id	Title	Price	User_id	profileName	review/helpfulness	review/score	review/time	review/summary	review/text
0	1882931173	Its Only Art If Its Well Hung!	NaN	AVCGYZL8FQQTD	Jim of Oz "jim-of- oz"	7/7	4.000000	940636800	Nice collection of Julie Strain Images	This is only for Julie Strain fans. It's a col...
1	0826414346	Dr. Seuss: American Icon	NaN	A30TK6U7DNS82R	Kevin Killian	10/10	5.000000	1095724800	Really Enjoyed It	I don't care much for Dr. Seuss but after read...
2	0826414346	Dr. Seuss: American Icon	NaN	A3UH4UZ4RSVO82	John Granger	10/11	5.000000	1078790400	Essential for every personal and Public Library	If people become the books they read and if "t...
3	0826414346	Dr. Seuss: American Icon	NaN	A2MVUWT453QH61	Roy E. Perry "amateur philosopher"	7/7	4.000000	1090713600	Philip Nel gives silly Seuss a serious treatment	Theodore Seuss Geisel (1904-1991), aka "D...
4	0826414346	Dr. Seuss: American Icon	NaN	A22X4XUPKF66MR	D. H. Richards "ninthwavestore"	3/3	4.000000	1107993600	Good academic overview	Philip Nel - Dr. Seuss: American IconThis is b...

Information:

```
[18]: rating.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3000000 entries, 0 to 2999999
Data columns (total 10 columns):
#   Column      Dtype
---  -
0    Id         object
1    Title      object
2    Price      float64
3    User_id    object
4    profileName object
5    review/helpfulness object
6    review/score float64
7    review/time int64
8    review/summary object
9    review/text object
dtypes: float64(2), int64(1), object(7)
memory usage: 228.9+ MB
```

In summary, Amazon book reviews serve as a powerful tool for understanding consumer preferences and improving business practices in the publishing industry. Amazon book reviews are a multifaceted aspect of the book industry, influencing sales, author reputation, search visibility, and providing valuable insights. They bridge the gap between businesses and consumers, fostering a more responsive and consumer-oriented business environment.

Data Preparation

Data preparation is a crucial step in the data analysis process. It involves cleaning, standardizing, and enriching raw data to make it ready for use in analytics and data science. Here are the typical steps involved in data preparation:

Book Data

- **Data Reduction:** This involves reducing the volume of data by removing irrelevant features or instances, or by transforming the data in a way that reduces its complexity.

Data Reduction

```
[12]: data_new = data[['Title', 'authors', 'categories', 'ratingsCount']]
      data_new.shape
[12]: (212484, 4)
```

- **Handling missing values:** This involves handling the missing values of data by either deleting missing rows or deleting columns or imputation the data

```
[13]: data_new.isnull().sum()
[13]: Title      1
      authors   31413
      categories 41199
      ratingsCount 162652
      dtype: int64
[14]: data_new.dropna(subset=['Title'], inplace=True)
      data_new.dropna(subset=['authors'], inplace=True)
      data_new.dropna(subset=['categories'], inplace=True)
      data_new.dropna(subset=['ratingsCount'], inplace=True)
      data_new.isnull().sum()
[14]: Title      0
      authors    0
      categories 0
      ratingsCount 0
      dtype: int64
```

Book Rating

- **Data Reduction:** This involves reducing the volume of data by removing irrelevant features or instances, or by transforming the data in a way that reduces its complexity.

```
[24]: rating_new = rating[['Id', 'Title', 'User_id', 'review/helpfulness', 'review/score', 'review/text']]
      rating_new.shape
[24]: (3000000, 6)
```

- **Handling missing values:** This involves handling the missing values of data by either deleting missing rows or deleting columns or imputation the data

```
[25]: rating_new.isnull().sum()
[25]: Id      0
      Title    208
      User_id 561787
      review/helpfulness 0
      review/score      0
      review/text      8
      dtype: int64
[26]: rating_new.dropna(subset=['Title'], inplace=True)
      rating_new.dropna(subset=['review/text'], inplace=True)
      rating_new.dropna(subset=['User_id'], inplace=True)
      rating_new.isnull().sum()
[26]: Id      0
      Title    0
      User_id 0
      review/helpfulness 0
      review/score      0
      review/text      0
      dtype: int64
```

Feature Engineering

Feature engineering is a crucial step in the data preprocessing process, especially when dealing with structured data. It involves creating new features (columns), transforming existing ones, and selecting the most relevant attributes to improve the performance and accuracy of machine learning models.

- **Data Blending:** Data blending is a process that involves combining data from multiple sources into a unified dataset. This technique is particularly useful in scenarios where data is spread across various platforms such as spreadsheets, business intelligence systems, IoT devices, cloud systems, and web applications. Data blending is a powerful tool for data analysts and data scientists, enabling them to derive actionable insights from diverse data sources

```
[28]: df = pd.merge(data_new, rating_new, on='Title', how='inner')
df.head()
```

	Title	authors	categories	ratingsCount	Id	User_id	review/helpfulness	review/score	review/text
0	The Church of Christ: A Biblical Ecclesiology ...	['Everett Ferguson']	['Religion']	5.000000	0802841899	ARI272XF8TOL4	74/81	5.000000	With the publication of Everett Ferguson's boo...
1	The Church of Christ: A Biblical Ecclesiology ...	['Everett Ferguson']	['Religion']	5.000000	0802841899	A36TPZSH8LBT1	2/3	5.000000	Everett Ferguson approaches the subject of ear...
2	The Church of Christ: A Biblical Ecclesiology ...	['Everett Ferguson']	['Religion']	5.000000	0802841899	ANX3DDV12ZRRU	2/3	4.000000	This book is a continual resource. It is so bi...
3	The Church of Christ: A Biblical Ecclesiology ...	['Everett Ferguson']	['Religion']	5.000000	0802841899	A2H2LORTA5EZY2	3/5	4.000000	This is a very useful and thorough text book. ...
4	Voices from the Farm: Adventures in Community ...	['Rupert Fike']	['Biography & Autobiography']	1.000000	157067051X	A3W1KIKQ93S62	21/21	5.000000	Ironically, I grew up in a small town close to...

- **Reordering:** For the sake of convenience and ease of use, the columns in the dataframe have been rearranged. This reordering process enhances the accessibility and readability of the data, making it more user-friendly and efficient to work with.

```
[29]: df = df[['Id', 'Title', 'authors', 'categories', 'ratingsCount', 'User_id', 'review/helpfulness', 'review/score', 'review/text']]
df.head()
```

	Id	Title	authors	categories	ratingsCount	User_id	review/helpfulness	review/score	review/text
0	0802841899	The Church of Christ: A Biblical Ecclesiology ...	['Everett Ferguson']	['Religion']	5.000000	ARI272XF8TOL4	74/81	5.000000	With the publication of Everett Ferguson's boo...
1	0802841899	The Church of Christ: A Biblical Ecclesiology ...	['Everett Ferguson']	['Religion']	5.000000	A36TPZSH8LBT1	2/3	5.000000	Everett Ferguson approaches the subject of ear...
2	0802841899	The Church of Christ: A Biblical Ecclesiology ...	['Everett Ferguson']	['Religion']	5.000000	ANX3DDV12ZRRU	2/3	4.000000	This book is a continual resource. It is so bi...
3	0802841899	The Church of Christ: A Biblical Ecclesiology ...	['Everett Ferguson']	['Religion']	5.000000	A2H2LORTA5EZY2	3/5	4.000000	This is a very useful and thorough text book. ...
4	157067051X	Voices from the Farm: Adventures in Community ...	['Rupert Fike']	['Biography & Autobiography']	1.000000	A3W1KIKQ93S62	21/21	5.000000	Ironically, I grew up in a small town close to...

- **Viewing Duplicates:** The dataframe is being examined to see if there are any duplicate entries present in the dataframe. This involves a detailed review to identify and handle any data points that are repeated. It's a vital step in ensuring the accuracy of our analysis.

```
[32]: duplicates = df.duplicated()

# Filter the dataframe for duplicates
df_duplicates = df[duplicates]

[51]: df_duplicates.head()
```

	Id	Title	authors	categories	ratingsCount	User_id	review/helpfulness	review/score	review/text
4458	B000KEPC7M	Helter Skelter	['Vincent Bugliosi', 'Curt Gentry']	['True Crime']	80.000000	A211BZGK92F7JY	0/0	5.000000	I read this book 25 years ago and, upon readin...
4964	B000PDFO2Q	Small Gods	['Terry Pratchett']	['Fiction']	85.000000	A13OF0B1394G31	0/0	5.000000	The Great God Om wakes up in tortoise-form jus...
5816	0812924835	No Disrespect	['Sister Souljah']	['Biography & Autobiography']	8.000000	A3HS2OKGPDYZPB	11/15	5.000000	Sister Souljah is awesome!!! I strongly advise...
6629	B000GRQ33K	The Clan of the Cave Bear	['Jean M. Auel']	['American fiction']	145.000000	AX6WWTSHXZ6S4	0/0	5.000000	I have enjoyed reading this book and the entir...
6706	B000GRQ33K	The Clan of the Cave Bear	['Jean M. Auel']	['American fiction']	145.000000	A15NHNGZNMGM8S	0/0	5.000000	I read this book 20 years ago and was eager to...

- **Removing Duplicates:** We have performed a data cleaning operation on the dataframe, where all duplicate entries have been identified and eliminated. This step is crucial in maintaining the accuracy of our data analysis and ensuring the integrity of our dataset.

```
[34]: df.drop_duplicates(inplace=True)
df.head()
```

	Id	Title	authors	categories	ratingsCount	User_id	review/helpfulness	review/score	review/text
0	0802841899	The Church of Christ: A Biblical Ecclesiology ...	['Everett Ferguson']	['Religion']	5.000000	ARI272XF8TOL4	74/81	5.000000	With the publication of Everett Ferguson's boo...
1	0802841899	The Church of Christ: A Biblical Ecclesiology ...	['Everett Ferguson']	['Religion']	5.000000	A36TPZSH8LBT1	2/3	5.000000	Everett Ferguson approaches the subject of ear...
2	0802841899	The Church of Christ: A Biblical Ecclesiology ...	['Everett Ferguson']	['Religion']	5.000000	ANX3DDV12ZRRU	2/3	4.000000	This book is a continual resource. It is so bi...
3	0802841899	The Church of Christ: A Biblical Ecclesiology ...	['Everett Ferguson']	['Religion']	5.000000	A2H2LORTA5EZY2	3/5	4.000000	This is a very useful and thorough text book. ...
4	157067051X	Voices from the Farm: Adventures in Community ...	['Rupert Fike']	['Biography & Autobiography']	1.000000	A3W1KIKQ93S62	21/21	5.000000	Ironically, I grew up in a small town close to...

Exploratory Data Analysis (EDA)

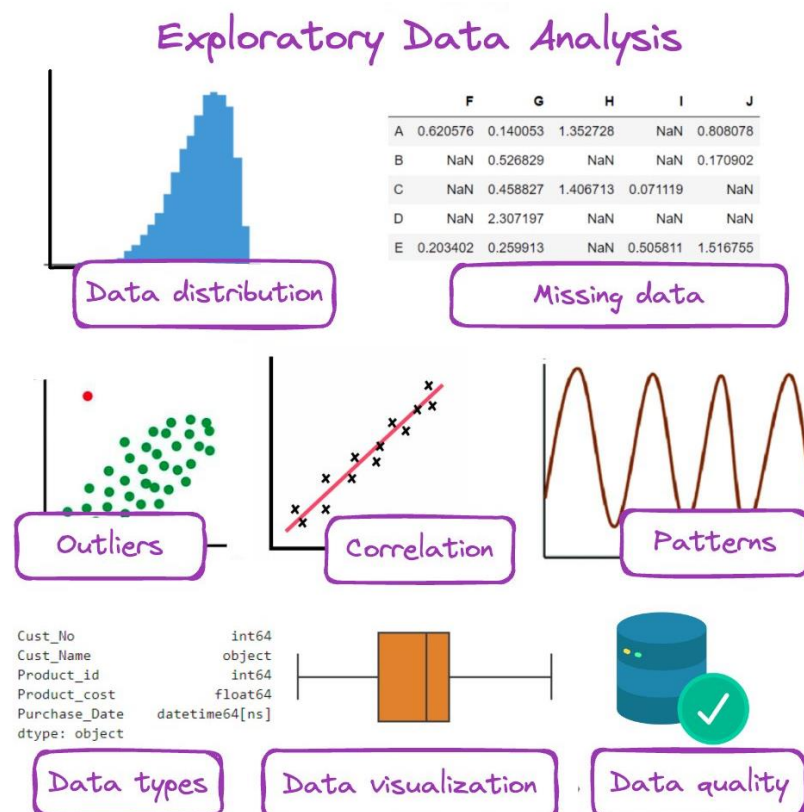
Exploratory Data Analysis (EDA) is an integral part of any data analysis, data science, or machine learning project. It involves understanding and summarizing the main characteristics of a dataset, often through visual methods.

Importance of EDA

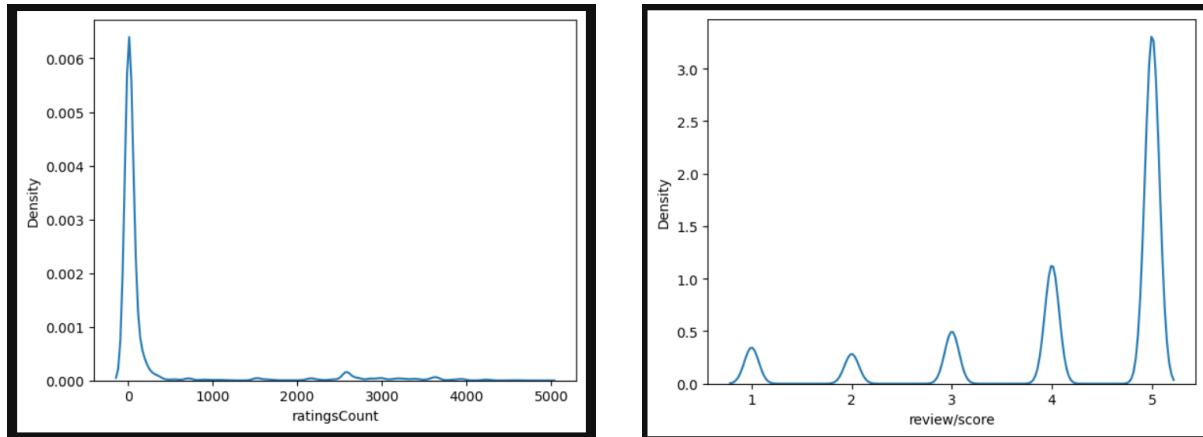
EDA is crucial because it allows us to understand the underlying structure of the data, identify outliers or anomalies, uncover patterns, test assumptions, and check for random chance. It's about getting to know your data, gaining a certain level of familiarity with it, which is essential before start modeling or predicting.

EDA is not a rigid process, but more of an art that will get better at by practicing. It can often lead to interesting insights about the data, which can be very useful for further analysis or modeling.

Remember, the goal of EDA is not to test hypotheses or make predictions, but to understand the data better to inform these later stages of the data analysis pipeline. It's about using all the tools at your disposal to get to know your data and to uncover the underlying structure and relationships within your data.



Data Distribution: Data distribution refers to the way data values are spread or distributed across a dataset. It is fundamental in choosing the correct statistical methods and models for analysis. The distribution of data informs the selection of appropriate statistical tests and models. For instance, many stats methods presume normal data distribution.



Missing Data: Missing data occurs when no data value is stored for a variable in an observation. Missing data can lead to weak or biased results when analyzing data. There are several methods to handle missing data, such as deletion, imputation, and prediction models.

```
[8]: Id          0
     Title        0
     authors      0
     categories   0
     ratingsCount 0
     User_id      0
     review/helpfulness 0
     review/score 0
     review/text  0
     word_count   0
     dtype: int64
```

Outliers: Outliers are data points that are significantly different from other observations. They can be caused by variability in the data or experimental errors. Outliers can greatly affect the results of your data analysis and statistical modeling.

```
[19]: # Outlier detection
      z_scores = stats.zscore(numerical_data)
      abs_z_scores = np.abs(z_scores)
      outliers = (abs_z_scores > 3).all(axis=1)
      print(f'Number of outliers in the data: {outliers.sum()}')

      Number of outliers in the data: 0
```

Correlation: Correlation is a statistical measure that describes the association between random variables. In the broadest sense, it refers to the degree to which a pair of variables are linearly related.

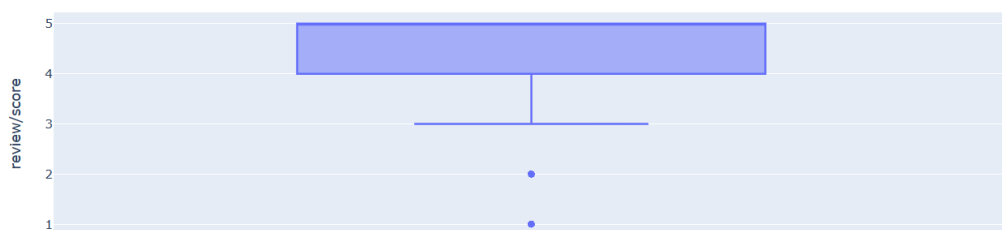
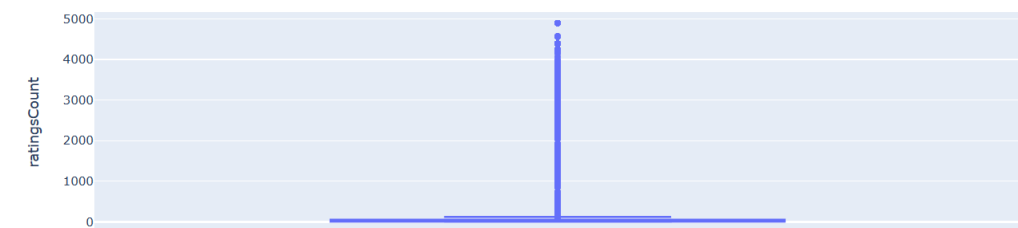
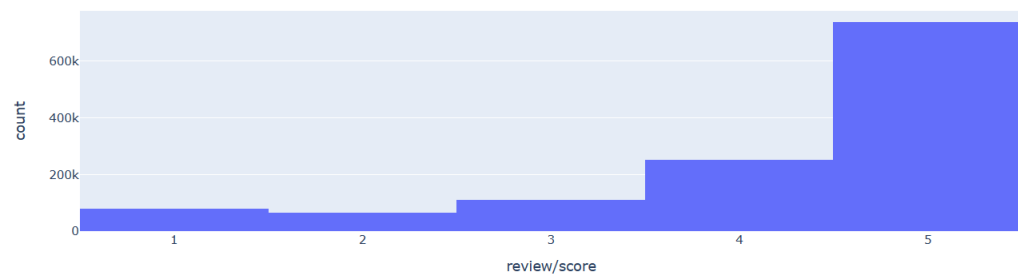
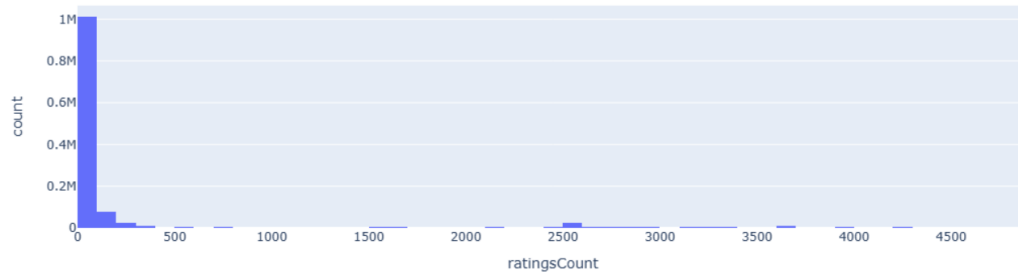
	ratingsCount	review/score
ratingsCount	1.000000	0.014493
review/score	0.014493	1.000000

Patterns: Patterns in data are often identified by using different data visualization techniques. They can also be found in large data sets by using automated algorithms, in a process known as pattern recognition.

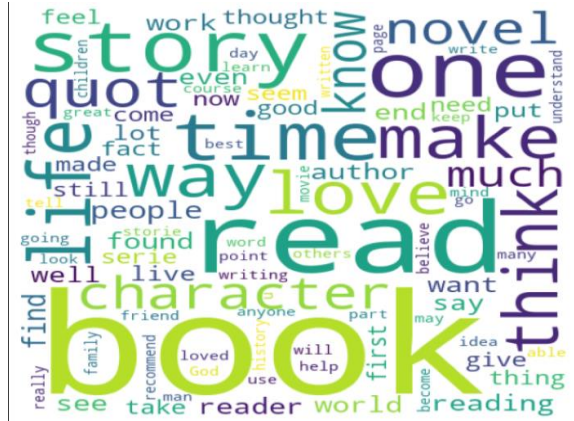
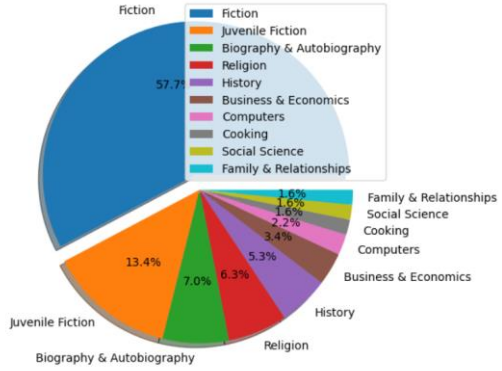
Data Types: In exploratory data analysis, it is important to know what type of data you are dealing with. The two main types of data are categorical (nominal and ordinal) and numerical (interval and ratio). Each data type requires different statistical techniques.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1240299 entries, 0 to 1240298
Data columns (total 10 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Id                    1240299 non-null object
1   Title                 1240299 non-null object
2   authors               1240299 non-null object
3   categories             1240299 non-null object
4   ratingsCount          1240299 non-null float64
5   User_id               1240299 non-null object
6   review/helpfulness    1240299 non-null object
7   review/score          1240299 non-null float64
8   review/text           1240299 non-null object
9   word_count            1240299 non-null int64
dtypes: float64(2), int64(1), object(7)
memory usage: 94.6+ MB
```

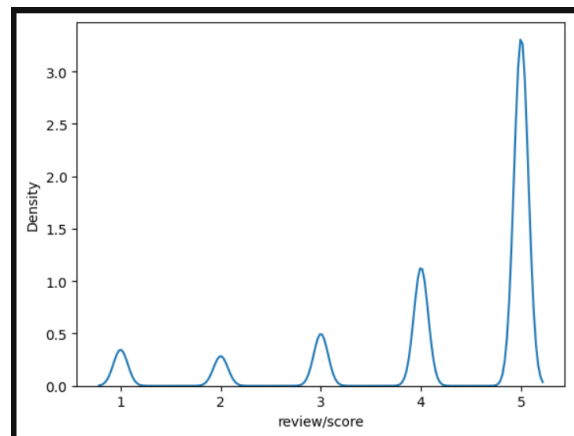
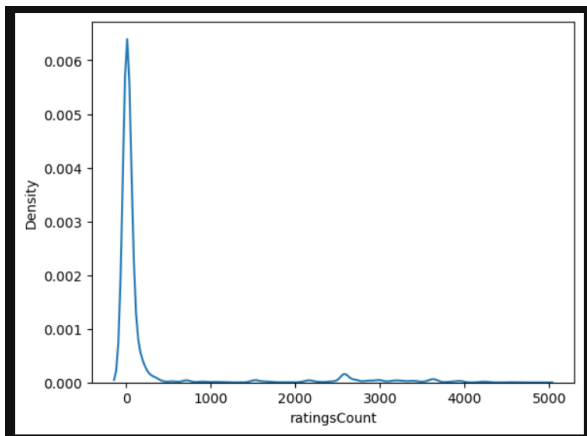
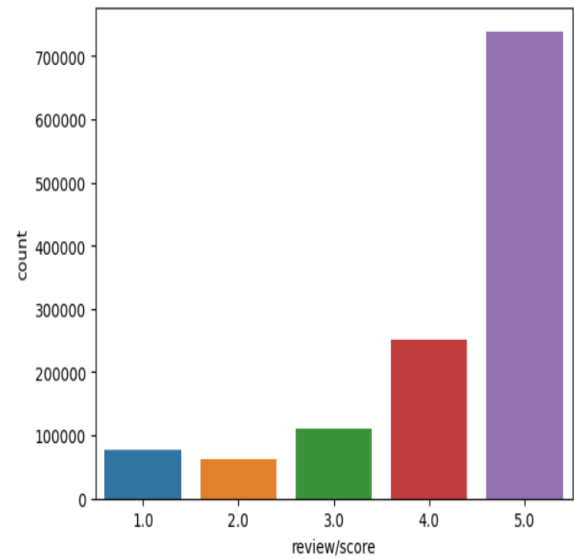
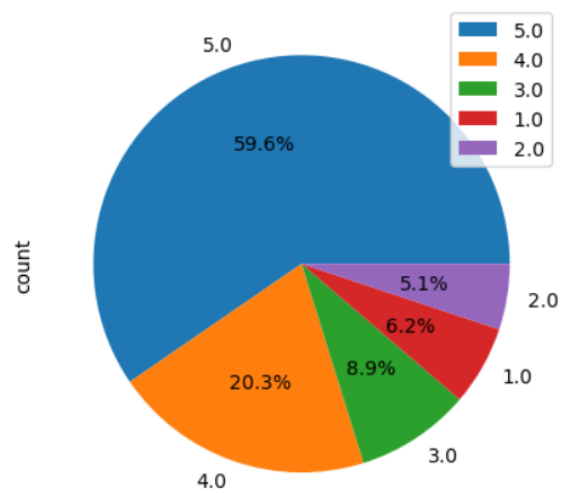

Data Visualization: Data visualization is the graphical representation of data and information. It uses visual elements like charts, graphs, and maps to provide an accessible way to see and understand trends, outliers, and patterns in data.

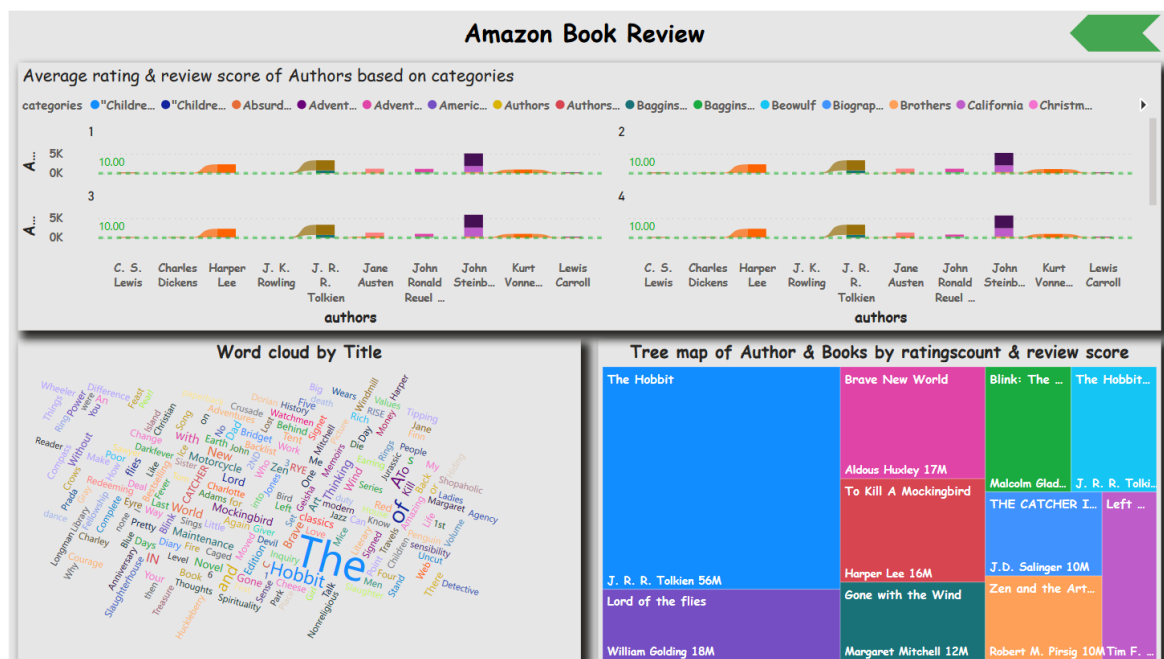
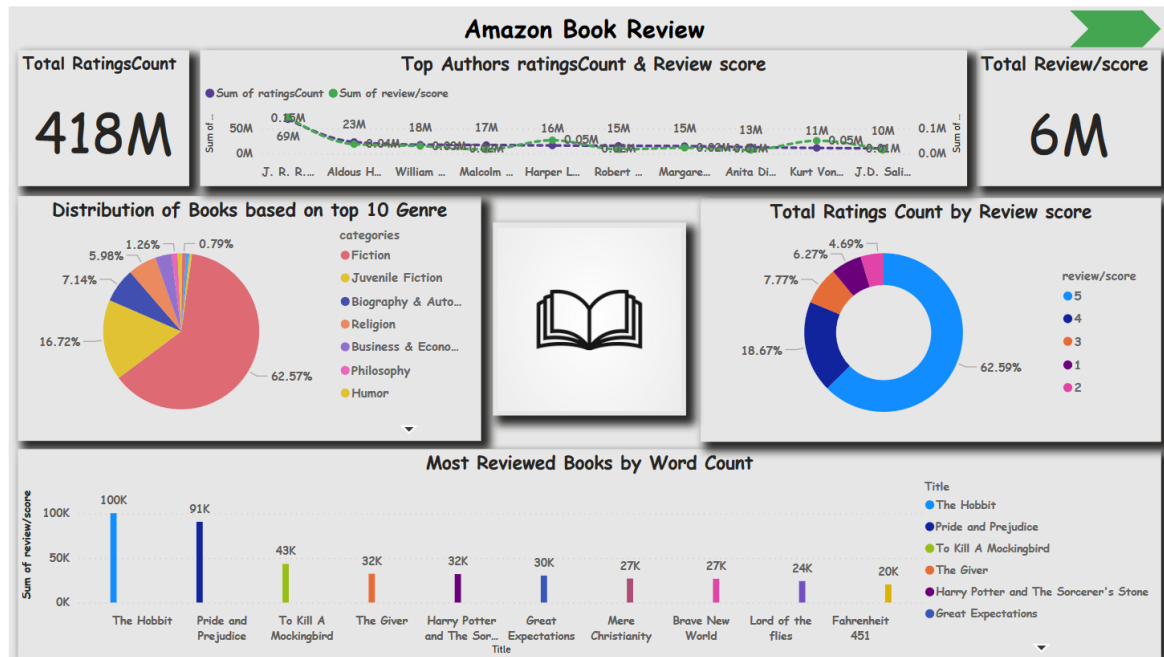


Distribution of Books Based on Top 10 Genre



Rating Distributions





Data Quality: Data quality refers to the condition of a set of values of qualitative or quantitative variables. High-quality data should be valid, accurate, complete, consistent, and relevant.

In summary, EDA provides a foundation for making informed decisions about the appropriate statistical methods to use and helps to interpret the results accurately.

Text Preprocessing

Text preprocessing is a crucial step in any natural language processing (NLP) task, including sentiment analysis. It involves cleaning and formatting the text data to improve the performance of machine learning models. Here are the key steps involved in text preprocessing for your capstone project:

Tokenization: This is the process of breaking down the text into individual words or tokens.

```
from nltk.tokenize import word_tokenize
```

Lowercasing: All the text is converted into lowercase to ensure that the algorithm does not treat the same words in different cases as different.

```
# Lower casing the reviews
df['review/text'] = df['review/text'].str.lower()
```

Stopwords Removal: Stopwords are common words that do not contribute much to the content or meaning of a document (e.g., “the”, “is”, “in”). These words are removed to reduce the dimensionality of the data and computational complexity.

```
import re # regular expression module
stemmer = nltk.SnowballStemmer("english") # for stemming
from nltk.corpus import stopwords
import string
stopword=set(stopwords.words('english')) # for stopword
```

Punctuation Removal: Punctuation can provide grammatical context to a sentence which supports our understanding. But for our vectorizer which counts the number of words and not the context, it does not add value, so it is removed.

```
authors_expressions = df['authors'].unique()
print(authors_expressions)
```

```
["['Everett Ferguson']" '['Rupert Fike']" '['Stefan Draminski']" ...
 '['Linda Barnes']" '['Patrick Balfour Baron Kinross']"
 '['Judy Halliday', 'Arthur Halliday']"]
```

```
df['authors'] = df['authors'].str.replace(r"\\[\\]|\\'", "", regex=True)
```

```
categories_expressions = df['categories'].unique()
print(categories_expressions)

["['Religion']" "['Biography & Autobiography']" "['History']" ...
"['Colombo (Sri Lanka)']" "['Mastodons']"
"['Tarzan (Fictitious character)']"]

df['categories'] = df['categories'].str.replace(r"\[|\]|\\'", "", regex=True)
```

Lemmatization: This is the process of reducing inflected (or sometimes derived) words to their base or root form—generally a written dictionary form or lemma.

```
from nltk.stem import WordNetLemmatizer
```

```
def lemmi(text):
    lemmatizer = WordNetLemmatizer()
    text=' '.join([lemmatizer.lemmatize(word) for word in text.split()])
    return text
```

Spell Check: Words are checked for spelling errors. Misspelled words can be corrected to improve the quality of the text data.

Removing HTML tags: If the data is scraped from the web, it might contain unwanted HTML tags.

```
def hapus_url(text):
    return re.sub(r'http\S+', '', text)
```

Removing Emojis and Emoticons: Depending on the nature of the task and the data, that might want to remove emojis and emoticons.

In summary, the goal of text preprocessing is to clean and simplify text data without losing valuable information for sentiment analysis. The steps to choose to include will depend on your specific use case. For instance, in some cases, punctuation, capitalization, and even emojis can carry sentiment, so that might decide to keep them for a sentiment analysis task.

Sentiment Analysis

Sentiment analysis, also known as opinion mining, is a subfield of Natural Language Processing (NLP) that involves determining the sentiment or emotion expressed in a piece of text. It can be used to analyze customer reviews, social media comments, and other user-generated content to understand public opinion about a product, service, or topic.

In this capstone project, we apply sentiment analysis to the Amazon Book Review Dataset. This dataset contains numerous book reviews from Amazon, each with a corresponding star rating. Our goal is to build a model that can accurately predict the sentiment of a review based on its text content.

Valence Aware Dictionary and Sentiment Reasoner (VADER):

The VADER (Valence Aware Dictionary and Sentiment Reasoner) sentiment analysis tool, which is specifically attuned to sentiments expressed in social media. VADER uses a combination of qualitative and quantitative means to determine the sentiment of a text. It not only talks about the Positivity and Negativity score but also tells us about how positive or negative a sentiment is.

```
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
vader = SentimentIntensityAnalyzer()
```

After initializing the SentimentIntensityAnalyzer, we can use it to score the sentiment of our review texts. The polarity_scores method of the SentimentIntensityAnalyzer object gives us the sentiment scores. This method returns a dictionary that contains positive, negative, neutral, and compound scores for the input text. The compound score is a metric that calculates the sum of all the lexicon ratings which have been normalized between -1 (most extreme negative) and +1 (most extreme positive).

```
# Lower casing the reviews
df['review/text'] = df['review/text'].str.lower()
```

This line of code is converting all the review texts in the 'review/text' column of the DataFrame (df) to lowercase. This is a common preprocessing step in text analysis to ensure that the algorithm does not treat the same words in different cases as different.

Swifter: Swifter is a Python package that efficiently applies any function to a pandas DataFrame or Series in the fastest available manner.

Vectorization: Swifter first tries to vectorize your function. Vectorization is the process of executing operations on entire arrays instead of individual elements, which can significantly speed up computations.

Estimation and Extrapolation: If vectorization is not possible, Swifter estimates the time it would take to process the function on a small subset of the data and extrapolates that to the full dataset.

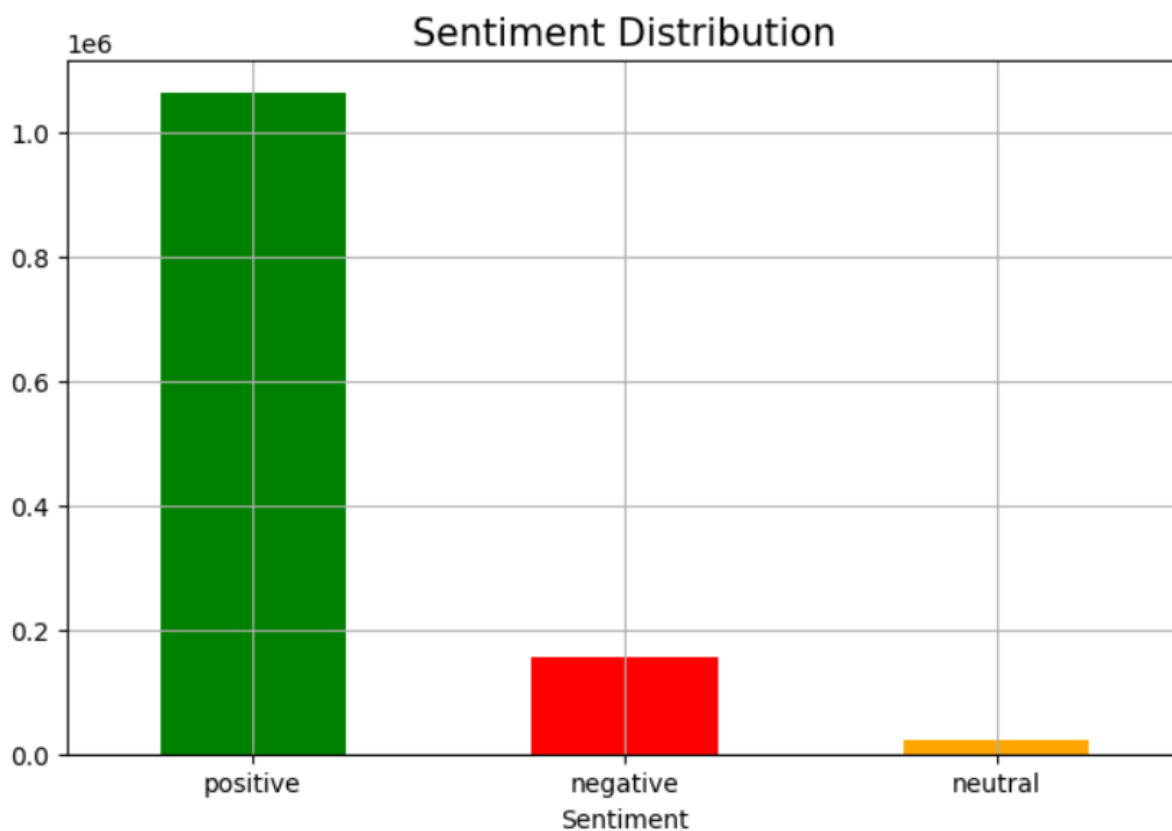
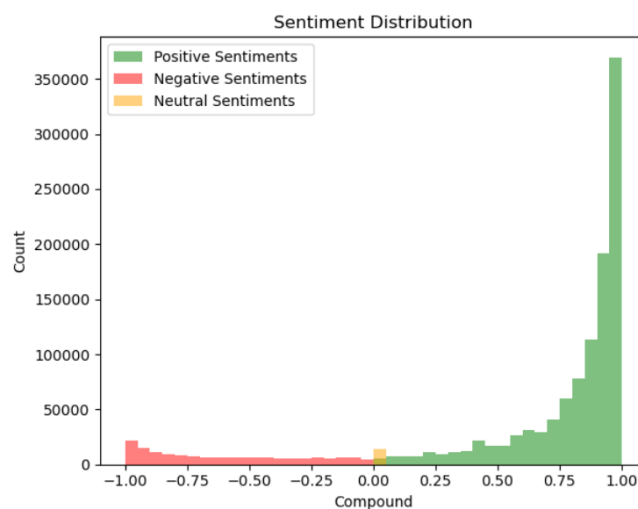
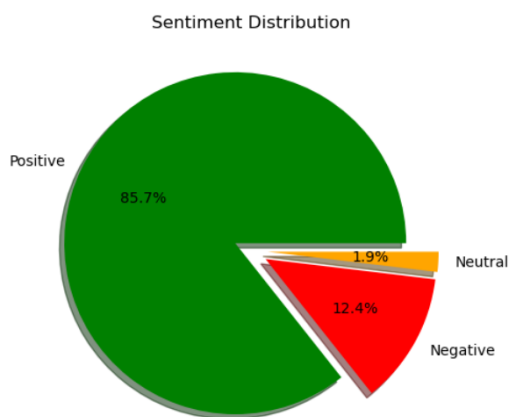
Dask Parallel Processing or Pandas Apply: If the extrapolated time is within an acceptable range, Swifter uses Dask, a flexible library for parallel computing in Python, to process the function. If not, it falls back to a simple pandas apply.

Validation of Output: Swifter validates the output to ensure that DataFrame applies are swift and accurate.

```
# Calculate sentiment scores and extract the compound column in a single step
df = df.assign(
    score=df['review/text'].swifter.apply(lambda review: vader_polarity_scores(review)),
    compound=lambda x: x['score'].swifter.apply(lambda score_dict: score_dict['compound'])
)
```

This code performs two operations simultaneously: it computes sentiment scores and isolates the compound column. It generates two new columns, namely 'score' and 'compound'. The 'score' column is computed by applying the vader_polarity_scores function to the data, using the swifter method for faster execution. The 'compound' column is then derived from the 'score' column, again utilizing the swifter method. Lambda functions are employed for both the application of vader_polarity_scores and the extraction of compound scores.

In summary, this approach provides us with an additional method for sentiment analysis, which can be particularly useful for understanding the sentiment of short texts like social media comments. It's also beneficial for texts where emojis and slangs are commonly used.



Clustering

Clustering is an unsupervised learning technique in machine learning that groups a set of data points into clusters based on their similarity. The goal is to group similar data points together so that data points in the same cluster are more similar to each other than to those in other clusters.

What is Clustering? Clustering is the task of grouping a set of data points in such a way that data points in the same group (known as a cluster) are more similar to each other than to those in other groups.

Types of Clustering: There are two broad types of clustering: Hard Clustering and Soft Clustering. In hard clustering, each data point either belongs to a cluster completely or not. In soft clustering, instead of putting each data point into a separate cluster, a probability or likelihood of that data point to be in those clusters is assigned.

Uses of Clustering: Clustering has a wide range of applications including market segmentation, statistical data analysis, social network analysis, image segmentation, anomaly detection, etc.

Clustering Algorithms: There are various clustering methods used in machine learning such as K-means, DBSCAN, Hierarchical Clustering, etc.

In summary, the goal of clustering is to determine the intrinsic grouping in a set of unlabeled data. But how to decide what constitutes a good clustering? It can be shown that there is no absolute “best” criterion which would be independent of the final aim of the clustering. Consequently, it is the user who should supply this criterion, in such a way that the result of the clustering will suit their needs.

The clustering algorithm used in this capstone project is K-means Clustering Algorithm from SKlearn module in python. But before using the K-means clustering algorithm, we have to ensure that the dataframe is on the same scale.

Scaling or normalization is an important step in many machine learning algorithms, including K-means clustering. Here's why:

Distance-Based Algorithms: K-means is a distance-based algorithm, which means it calculates the distance between data points to form clusters. If features are in different scales (for example, one feature ranges from 0 to 1 and other ranges from 0 to 1000), the algorithm might not perform well. This is because the feature with a larger range may dominate the distance calculations.

Standardization and Normalization: To prevent this, we can scale the features so that they have the same range. This can be done through standardization (bringing the data to a mean of 0 and standard deviation of 1) or normalization (bringing the data to a range of [0,1]).

Effect on Clustering: Scaling the data can significantly affect the results of clustering. Without scaling, the clusters could be largely based on the features with higher magnitudes. With scaling, all features contribute equally to the distance calculation and hence to the clustering.

K-means and Scaling: K-means does not automatically perform feature scaling. Therefore, it's a good practice to scale your data before applying K-means clustering.

```
new_df.describe()
```

	Title	authors	categories	ratingsCount	review/score	Sentiment
count	1240299.000000	1240299.000000	1240299.000000	1240299.000000	1240299.000000	1240299.000000
mean	20776.219246	8486.817455	83.340335	265.381027	4.219235	0.162407
std	13673.400307	8115.679118	285.722782	789.324478	1.182009	0.417063
min	0.000000	0.000000	0.000000	1.000000	1.000000	0.000000
25%	7996.000000	1911.000000	8.000000	3.000000	4.000000	0.000000
50%	19536.000000	5495.000000	8.000000	10.000000	5.000000	0.000000
75%	32350.500000	13148.000000	27.000000	50.000000	5.000000	0.000000
max	46536.000000	30234.000000	2515.000000	4895.000000	5.000000	2.000000

```
rank_df = new_df.rank(method='first')
```

```
rank_df.describe()
```

	Title	authors	categories	ratingsCount	review/score	Sentiment
count	1240299.000000	1240299.000000	1240299.000000	1240299.000000	1240299.000000	1240299.000000
mean	620150.000000	620150.000000	620150.000000	620150.000000	620150.000000	620150.000000
std	358043.625100	358043.625100	358043.625100	358043.625100	358043.625100	358043.625101
min	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
25%	310075.500000	310075.500000	310075.500000	310075.500000	310075.500000	310075.500000
50%	620150.000000	620150.000000	620150.000000	620150.000000	620150.000000	620150.000000
75%	930224.500000	930224.500000	930224.500000	930224.500000	930224.500000	930224.500000
max	1240299.000000	1240299.000000	1240299.000000	1240299.000000	1240299.000000	1240299.000000

```
# Calculating z-score
normalized_df = (rank_df - rank_df.mean()) / rank_df.std()
```

```
normalized_df.describe()
```

	Title	authors	categories	ratingsCount	review/score	Sentiment
count	1240299.000000	1240299.000000	1240299.000000	1240299.000000	1240299.000000	1240299.000000
mean	0.000000	-0.000000	0.000000	0.000000	-0.000000	0.000000
std	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
min	-1.732049	-1.732049	-1.732049	-1.732049	-1.732049	-1.732049
25%	-0.866024	-0.866024	-0.866024	-0.866024	-0.866024	-0.866024
50%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
75%	0.866024	0.866024	0.866024	0.866024	0.866024	0.866024
max	1.732049	1.732049	1.732049	1.732049	1.732049	1.732049

K-means Clustering:

K-means is an unsupervised learning algorithm used for clustering. It groups the unlabeled dataset into different clusters. Here's how it works:

Initialization: Select the number of clusters K and initialize K points as centroids randomly.

Assignment: Assign each data point to the nearest centroid. This forms K clusters.

Update: Calculate the new centroid (mean) of each cluster.

Repeat: Repeat the assignment and update steps until the centroids do not change significantly or a maximum number of iterations is reached.

The objective of K-means is to minimize the sum of the distances of all points within a cluster from the centroid of that cluster. In other words, it tries to make the inter-cluster data points as similar as possible while also keeping the clusters as different (far) as possible. It assigns data points to a cluster such that the sum of the squared distance between the data points and the cluster's centroid (arithmetic mean of all the data points that belong to that cluster) is at the minimum.

```
from sklearn.cluster import KMeans

kmeans = KMeans(n_clusters=3, random_state=42).fit(normalized_df[['Title', 'authors', 'categories', 'ratingsCount', 'review/score', 'Sentiment']])

kmeans
```

KMeans

KMeans(n_clusters=3, random_state=42)

```
kmeans.cluster_centers_
```

```
array([[ -5.48151080e-01, -4.64268202e-01, -4.56097586e-01,
         9.62405935e-01, -3.42442356e-01, -4.02005736e-01],
       [ 8.53538134e-01,  5.62917837e-01,  2.89166066e-01,
        -1.54761701e-01,  4.54858168e-01,  7.30373117e-01],
       [-9.14911843e-01, -4.85649122e-01, -1.24974062e-04,
        -8.23012690e-01, -4.29895885e-01, -8.59718185e-01]])
```

```
three_cluster_df.groupby('Cluster').count()['ratingsCount']
```

```
Cluster
0    359221
1     566979
2     314099
Name: ratingsCount, dtype: int64
```

```
three_cluster_df.groupby('Cluster').count()['review/score']
```

```
Cluster
0    359221
1     566979
2     314099
Name: review/score, dtype: int64
```

K detection using Silhouette Score method:

The Silhouette Score method is a way to determine the optimal number of clusters (K) in K-means clustering. Here's how it works:

Compute Silhouette Score for Each Data Point: For each data point, calculate the average distance to all other points in the same cluster (a(i)) and the average distance to all points in the nearest cluster (b(i)).

Calculate Silhouette Coefficient: The silhouette coefficient for a data point is then given by $(b(i) - a(i)) / \max(a(i), b(i))$. This value ranges from -1 to 1.

Compute Average Silhouette Score: The silhouette score for a clustering solution is the average silhouette coefficient across all data points.

Determine Optimal K: Compute the silhouette score for different values of K, and choose the K that gives the highest silhouette score.

```
%%time

# Assuming normalized_df is your DataFrame
data = normalized_df[['Title', 'authors', 'categories', 'ratingsCount', 'review/score', 'Sentiment']].values.copy()

def compute_silhouette(n_cluster):
    kmeans = KMeans(n_clusters=n_cluster, random_state=42).fit(data)
    # Compute the silhouette score on a sample of 10000 data points
    silhouette_avg = silhouette_score(data, kmeans.labels_, sample_size=10000, random_state=42)
    return n_cluster, silhouette_avg

# Compute silhouette scores for different numbers of clusters
results = [compute_silhouette(n_cluster) for n_cluster in [3,4,5,6,7,8]]

for n_cluster, silhouette_avg in results:
    print('Silhouette Score for %i Clusters: %.4f' % (n_cluster, silhouette_avg))
```

```
Silhouette Score for 3 Clusters: 0.1877
Silhouette Score for 4 Clusters: 0.2201
Silhouette Score for 5 Clusters: 0.2150
Silhouette Score for 6 Clusters: 0.2060
Silhouette Score for 7 Clusters: 0.2099
Silhouette Score for 8 Clusters: 0.2203
CPU times: total: 50.8 s
Wall time: 13.4 s
```

The eight cluster has high score, so the k-means clustering model is again built on k=8, and assigned name as eight_cluster_df.

```
kmeans = KMeans(n_clusters=8,random_state=42).fit(
    normalized_df[['Title','authors','categories','ratingsCount','review/score','Sentiment']]
)
eight_cluster_df = normalized_df[['Title','authors','categories','ratingsCount','review/score','Sentiment']].copy(deep=True)
eight_cluster_df['Cluster'] = kmeans.labels_
```

kmeans.cluster_centers_

```
array([[ -0.24613054,  0.40352243, -0.94899616, -0.48168405,  0.13257255,
        -0.42279017],
       [ 0.81774928, -0.53485946,  0.34398934,  0.66465447, -0.93374059,
        0.69507152],
       [ 0.84803243,  1.16321829,  0.34461303, -0.48004848, -0.95809298,
        0.77848542],
       [ 0.78982562, -0.55328451,  0.3098802 ,  0.73441081,  1.17801496,
        0.57852281],
       [-1.08008712, -0.79267115, -0.82219589,  0.47002541, -0.45822037,
       -1.17019705],
       [-0.92026596, -0.3899365 ,  1.15873737, -0.48011833, -0.30086715,
       -1.03115775],
       [-0.94977251, -0.54583785, -0.38672082,  0.00319439, -0.75005694,
        1.37626793],
       [ 0.89994469,  1.16956367,  0.38725358, -0.51664355,  1.2432006 ,
        0.68489442]])
```

eight_cluster_df.groupby('Cluster').count()['review/score']

```
Cluster
0    164743
1    116143
2    137352
3    166216
4    245315
5    150577
6     71904
7    188049
Name: review/score, dtype: int64
```

eight_cluster_df.groupby('Cluster').count()['ratingsCount']

```
Cluster
0    164743
1    116143
2    137352
3    166216
4    245315
5    150577
6     71904
7    188049
Name: ratingsCount, dtype: int64
```

The cluster number 8 has high value in centroid score so the 8 cluster is assigned to high_value_cluster,

```
high_value_cluster = eight_cluster_df.loc[eight_cluster_df['Cluster'] == 7]
```

```
print(high_value_cluster.shape)
```

```
(188049, 7)
```

The high_value_cluster's value is located from dataframe and assigned to a new dataframe called clust_df.

```
clust_df = new_df.loc[high_value_cluster.index]
clust_df.head()
```

	Title	authors	categories	ratingsCount	review/score	Sentiment
327538	8574	7213	299	1.000000	5.000000	1
333449	8702	7313	795	1.000000	5.000000	1
333587	8712	7321	99	1.000000	5.000000	1
333591	8712	7321	99	1.000000	5.000000	1
334149	8745	7345	114	1.000000	5.000000	1

Data Balancing

Balancing a DataFrame is a common task when dealing with imbalanced datasets, especially in classification problems where the classes are not equally represented. Here are some techniques to balance a DataFrame:

Downsampling: This involves reducing the number of instances from the over-represented class to make it equal to the under-represented class. One can use the `groupby` and `head` functions in pandas to achieve this.

Upsampling: This involves increasing the number of instances from the under-represented class by randomly duplicating instances until it's equal to the over-represented class. One can use the `sample` function in pandas with replacement to achieve this.

SMOTE (Synthetic Minority Over-sampling Technique): This is a more sophisticated method that creates synthetic examples of the minority class. This can be done using the `imbalanced-learn` library.

```
random_seed = 42

from imblearn.over_sampling import SMOTE
from sklearn.model_selection import train_test_split

# Assuming 'X' is your feature set and 'y' is your target variable
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

smote = SMOTE(random_state=random_seed)
X_train_smote, y_train_smote = smote.fit_resample(X_train, y_train)
```

Remember, the choice of method will depend on the specific problem and the nature of the data. It's also important to note that balancing a DataFrame doesn't always lead to better model performance. It's always a good idea to try different approaches and see what works best for the specific use case.

Model Building & Evaluation

Model building in machine learning is a systematic procedure where a predictive or statistical model is developed with the aim of making predictions or categorizing outcomes.

Define the Problem: The problem at hand involves analyzing the sentiment of Amazon book reviews. This is a classification problem where the task is to categorize each review as positive, negative, or neutral based on the text of the review.

Choose a Model: Based on the problem definition, one would need to select an appropriate model. For sentiment analysis, models like Naive Bayes, Logistic Regression, Support Vector Machines, or even more complex models like Recurrent Neural Networks or Transformers could be used.

Train the Model: Once a model has been chosen, the next step is to train it using the data. This involves feeding the data into the model so it can learn the underlying patterns. The model will learn the relationship between the features (the input data) and the target variable (the outcome being predicted).

Evaluate the Model: After training the model, one needs to evaluate its performance. This typically involves splitting the data into a training set and a test set. The model is trained on the training set and then tested on the test set. Various metrics can be used to evaluate the model's performance for classification problems, such as:

Accuracy: This is the ratio of the total number of correct predictions to the total number of predictions.

Precision: This is the ratio of true positive predictions to the total predicted positives.

Recall (Sensitivity): This is the ratio of true positive predictions to the total actual positives.

F1 Score: This is the harmonic mean of precision and recall, and it tries to balance the two.

These metrics provide a comprehensive view of the model's performance across all classes, especially when dealing with imbalanced datasets.

Tune the Model: Based on the evaluation, one might decide to tune the model to improve its performance. This could involve tuning hyperparameters, which are the parameters that define the model structure. For example, in a neural network, the number of layers and the number of nodes in each layer are hyperparameters. One might also consider using regularization techniques to prevent overfitting.

Predict and Interpret: Once one is satisfied with the model's performance, it can be used to make predictions on new, unseen data. It's also important to interpret the results. Can one explain why the model made the predictions it did? Interpreting the model can provide insights into the data and the problem being solved.

Deploy and Monitor: The final step is to deploy the model and start using it to make predictions on new data. Once the model is deployed, one would want to monitor its performance over time. Is it still making accurate predictions? If not, it might need to be retrained with new data.

In essence, model building is a dynamic and iterative process that involves a blend of domain knowledge, statistical understanding, and machine learning techniques.

Logistic Regression Model

Logistic regression is a supervised machine learning algorithm used for binary classification tasks where the goal is to predict the probability that an instance belongs to a given class. It uses a sigmoid function to map the input variables into probabilities between 0 and 1. The logistic function, also known as the sigmoid function, maps any real value into another value within a range of 0 and 1. Logistic regression entails modeling the likelihood of a discrete outcome based on input variables. Typically, logistic regression is employed for binary outcomes, such as true/false or yes/no scenarios, although multinomial logistic regression extends this to cases with multiple discrete outcomes. This method proves valuable in classification tasks, where the objective is to categorize new samples effectively. Given that various aspects of cybersecurity involve classification challenges, such as detecting attacks, logistic regression emerges as a valuable analytical tool in this domain.

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score
```

Multinomial Logistic Regression Model: Multinomial logistic regression is a classification method that generalizes logistic regression to multiclass problems, i.e., with more than two possible discrete outcomes. It is used to predict the probabilities of the different possible outcomes of a categorically distributed dependent variable, given a set of independent variables.

```
lr = LogisticRegression(max_iter=2000)
```

```
%%time
model = lr.fit(X_train_smote, y_train_smote)
```

Visualizing Logistic Regression Model: One can visualize a logistic regression model using libraries like seaborn in Python. The seaborn library has a function called regplot which can be used to plot a logistic regression curve. The x-axis shows the values of the predictor variable and the y-axis displays the predicted probability of the event.

Evaluating Logistic Regression Model: Evaluating a logistic regression model involves validating the confusion matrix and classification reports which includes precision, recall, f1-score.

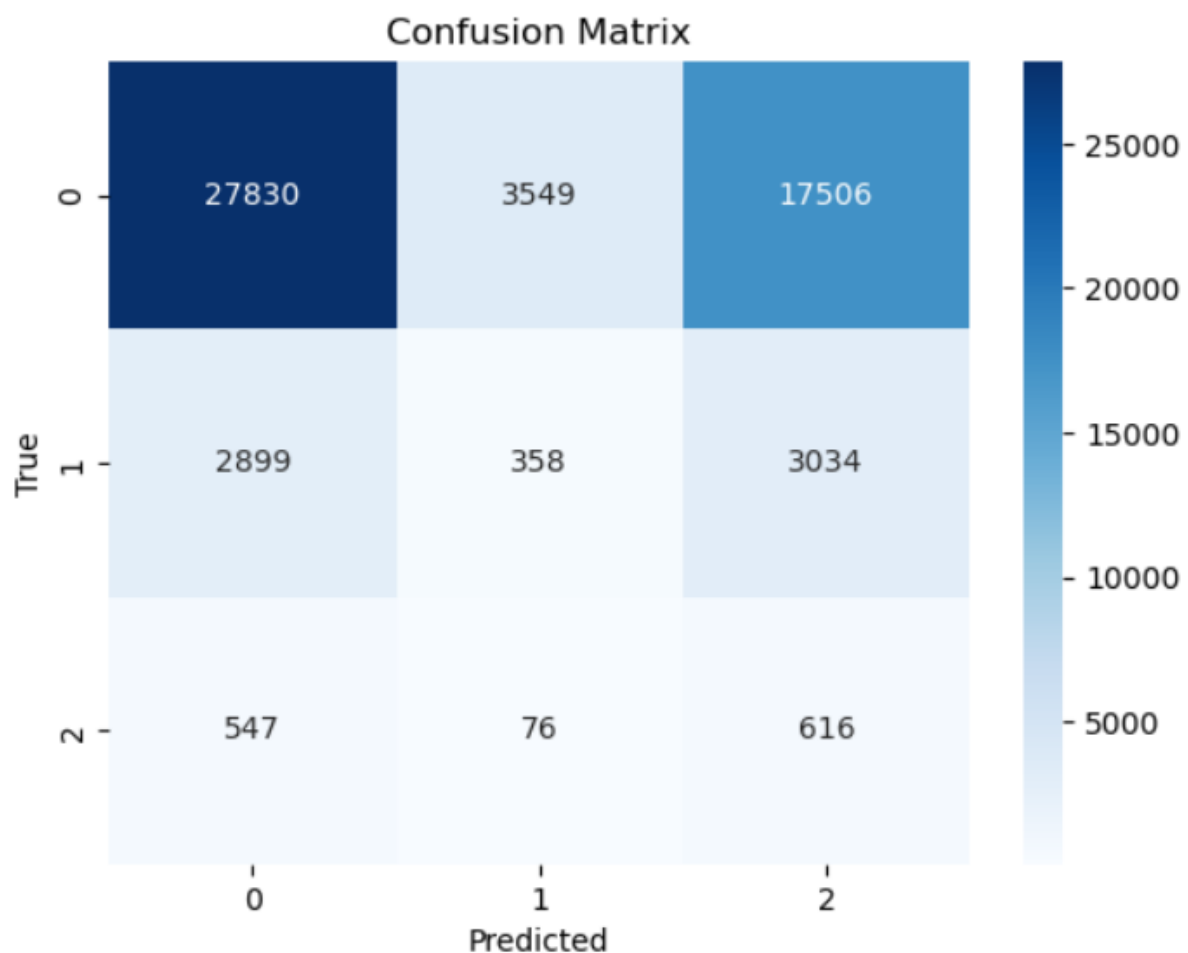
```
# Make predictions
y_pred_lr = lr.predict(X_test)
```

```
# Check accuracy
accuracy = accuracy_score(y_test, y_pred_lr)
print(f'Model accuracy: {accuracy}')
```

Model accuracy: 0.510573429052557

```
print(classification_report(y_pred_lr, y_test))
```

	precision	recall	f1-score	support
0	0.57	0.89	0.69	31276
1	0.06	0.09	0.07	3983
2	0.50	0.03	0.06	21156
accuracy			0.51	56415
macro avg	0.37	0.34	0.27	56415
weighted avg	0.51	0.51	0.41	56415



K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) is a supervised machine learning algorithm used for both classification and regression tasks. It works on the principle that data points that are similar to each other fall in close proximity. For a given data point, the KNN algorithm finds the ‘k’ nearest data points and predicts the label based on the majority label of these ‘k’ points.

Multinomial KNN Model: The term “multinomial” typically refers to models that predict outcomes of multi-class categorical variables, like Multinomial Logistic Regression. However, KNN inherently supports multi-class classification. If there are more than two classes in the target variable, KNN can predict the class of a new instance based on the majority class of ‘k’ nearest neighbors.

```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train_smote, y_train_smote)
```

Visualizing KNN Model: Visualizing a KNN model can be done by plotting the data points in a feature space and color-coding them based on their class. Each data point represents an instance of the dataset. The color or shape of the point represents its class. For a new, unclassified point, can visualize its ‘k’ nearest neighbors and the class it would be assigned based on the majority class of these neighbors.

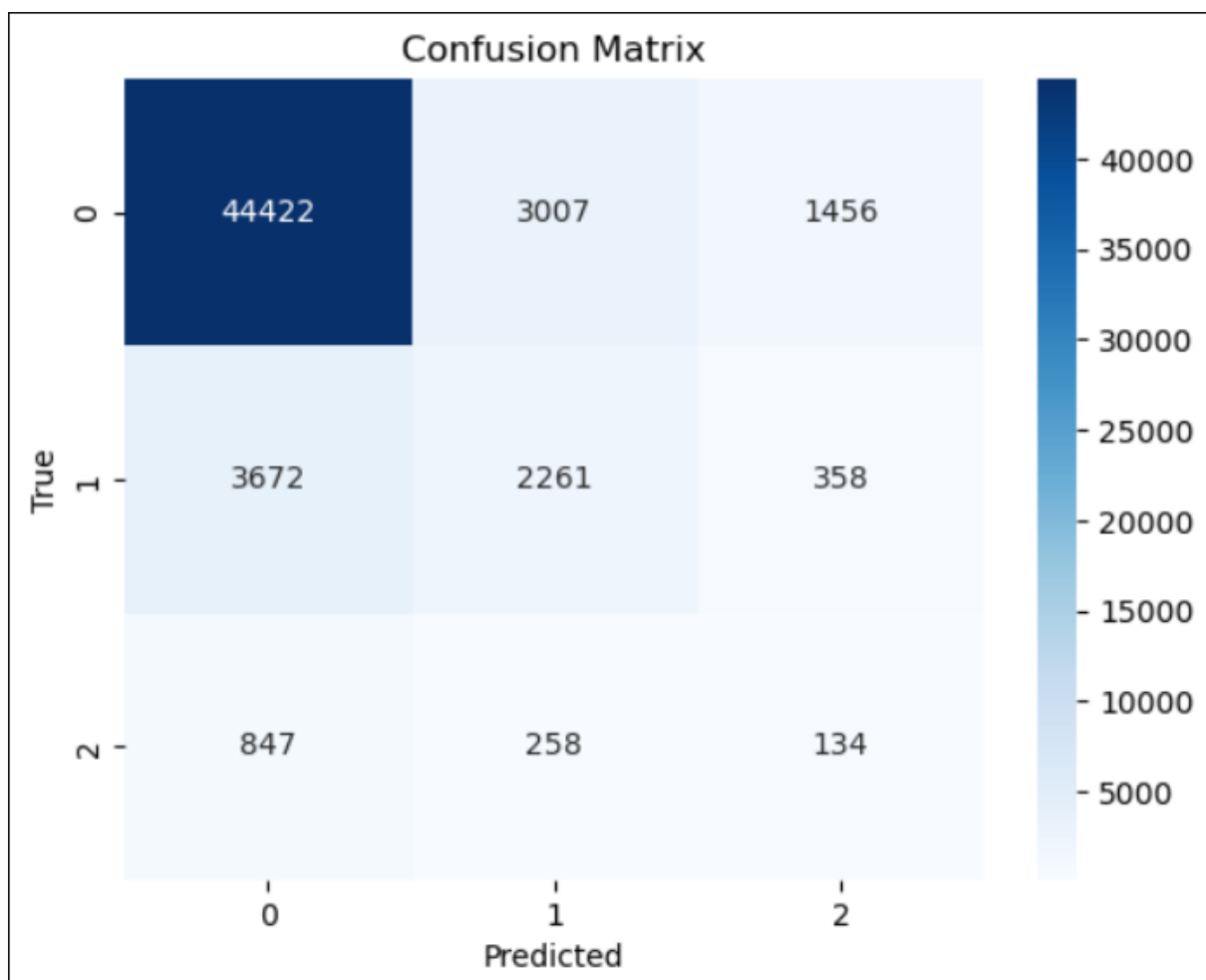
Evaluating KNN Model: Evaluating a KNN model often involves splitting the data into a training set and a test set. The model is trained on the training set and predictions are made on the test set. These predictions are then compared with the actual labels to calculate the accuracy of the model. Other metrics like precision, recall, and F1-score can also be used depending on the problem.

```
y_pred_knn = knn.predict(X_test)
print("KNN Accuracy:", accuracy_score(y_test, y_pred_knn))
```

KNN Accuracy: 0.8298679429229815

```
print(classification_report(y_pred_knn, y_test))
```

	precision	recall	f1-score	support
0	0.91	0.91	0.91	48941
1	0.36	0.41	0.38	5526
2	0.11	0.07	0.08	1948
accuracy			0.83	56415
macro avg	0.46	0.46	0.46	56415
weighted avg	0.83	0.83	0.83	56415

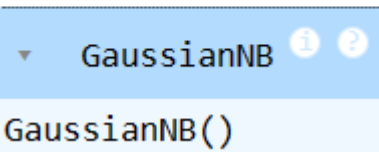


Naive Bayes

Naive Bayes is a probabilistic machine learning algorithm used for classification tasks. It is based on Bayes' Theorem and makes the 'naive' assumption that every pair of features being classified is independent of each other. Despite this simplification, Naive Bayes classifiers are widely used for their simplicity and efficiency, particularly in text classification tasks.

Multinomial Naive Bayes Model: Multinomial Naive Bayes (MNB) is a variant of Naive Bayes designed specifically for discrete data. It is commonly used for text classification tasks where we need to deal with discrete data like word counts in documents. The term "multinomial" refers to the type of data distribution assumed by the model.

```
from sklearn.naive_bayes import GaussianNB
nb = GaussianNB()
nb.fit(X_train_smote, y_train_smote)
```



▼ GaussianNB ⓘ ?

GaussianNB()

Visualizing Naive Bayes Model: Visualizing a Naive Bayes model can be challenging due to its probabilistic nature. However, for certain types of data, can visualize the probabilities assigned to different classes. For example, in text classification, that might visualize the top words associated with each class based on their probabilities.

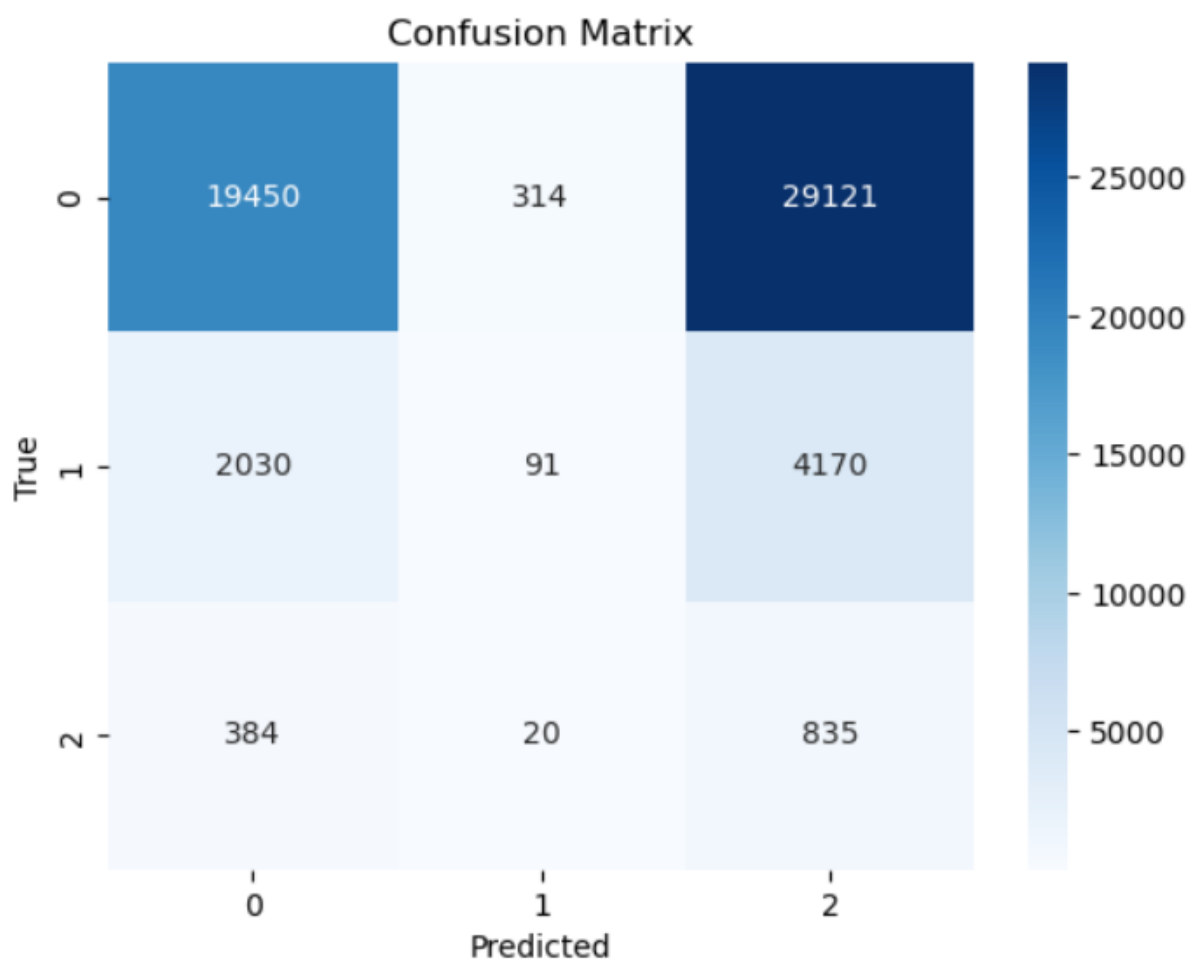
Evaluating Naive Bayes Model: Evaluating a Naive Bayes model often involves splitting the data into a training set and a test set. The model is trained on the training set and predictions are made on the test set. These predictions are then compared with the actual labels to calculate the accuracy of the model. Other metrics like precision, recall, and F1-score can also be used depending on the problem.

```
y_pred_nb = nb.predict(X_test)
print("Naive Bayes Accuracy:", accuracy_score(y_test, y_pred_nb))
```

Naive Bayes Accuracy: 0.3611805370911991

```
print(classification_report(y_pred_nb, y_test))
```

	precision	recall	f1-score	support
0	0.40	0.89	0.55	21864
1	0.01	0.21	0.03	425
2	0.67	0.02	0.05	34126
accuracy			0.36	56415
macro avg	0.36	0.38	0.21	56415
weighted avg	0.56	0.36	0.24	56415



Decision Tree

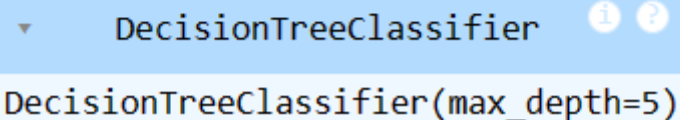
A decision tree is a supervised machine learning algorithm used for both classification and regression tasks. It builds a flowchart-like tree structure where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label. The goal is to find the attribute that maximizes the information gain or the reduction in impurity after the split.

Multinomial Decision Tree Classifier Model: In the context of decision trees, the term “multinomial” is not typically used. However, decision trees inherently support multi-class classification. If there are more than two classes in the target variable, a decision tree can predict the class of a new instance based on the majority class of the leaf node it falls into.

```
from sklearn import tree
```

```
dt_model = tree.DecisionTreeClassifier(  
    max_depth=5  
)
```

```
dt_model.fit(X_train_smote, y_train_smote)
```



```
DecisionTreeClassifier(max_depth=5)
```

Visualizing Decision Tree Model: To visualize a decision tree model in several ways:

Text Representation: To print a text representation of the tree using the `export_text` method from the `sklearn.tree` module.

Plot Tree: To plot the tree using the `plot_tree` method from the `sklearn.tree` module.

Graphviz: To use the `export_graphviz` method from the `sklearn.tree` module to visualize the decision tree with Graphviz.

Dtreviz Package: To use the `dtreeviz` package to create more detailed and informative visualizations.

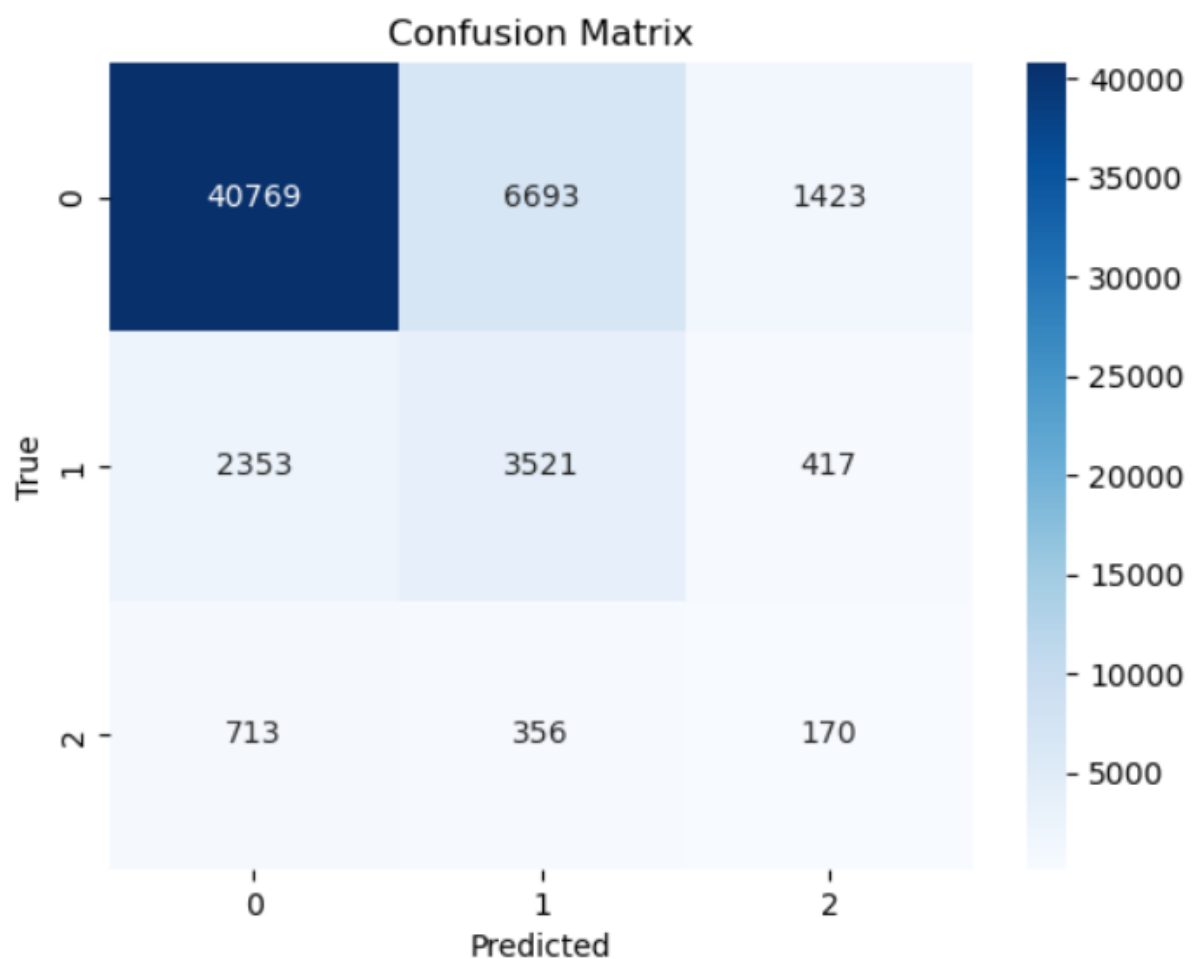
Evaluating Decision Tree Model: Evaluating a decision tree model often involves splitting the data into a training set and a test set. The model is trained on the training set and predictions are made on the test set. These predictions are then compared with the actual labels to calculate the accuracy of the model. Other metrics like precision, recall, and F1-score can also be used depending on the problem.

```
# Make predictions
y_pred_dt = dt_model.predict(X_test)
print("Decision Tree Accuracy:", accuracy_score(y_test, y_pred_dt))
```

Decision Tree Accuracy: 0.8591154834707081

```
print(classification_report(y_pred_dt, y_test))
```

	precision	recall	f1-score	support
0	0.98	0.88	0.93	53954
1	0.11	0.44	0.17	1510
2	0.09	0.12	0.10	951
accuracy			0.86	56415
macro avg	0.39	0.48	0.40	56415
weighted avg	0.94	0.86	0.89	56415



Random Forest

Random Forest is a powerful ensemble machine learning algorithm used for both classification and regression tasks. It works by creating a multitude of decision trees during the training phase. Each tree is constructed using a random subset of the dataset to measure a random subset of features in each partition. This randomness introduces variability among individual trees, reducing the risk of overfitting and improving overall prediction performance. In prediction, the algorithm aggregates the results of all trees, either by voting (for classification tasks) or by averaging (for regression tasks).

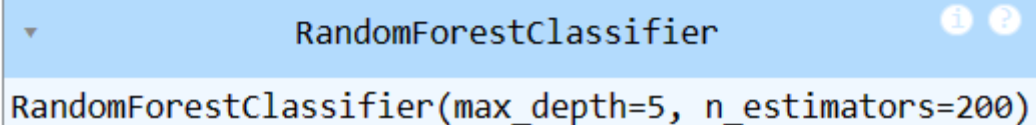
Multinomial Random Forest Classifier Model: Random Forest inherently supports multi-class classification. If there are more than two classes in the target variable, a Random Forest can predict the class of a new instance based on the majority class of the leaf node it falls into.

```
from sklearn.ensemble import RandomForestClassifier
```

```
rf_model = RandomForestClassifier(  
    n_estimators=200,  
    max_depth=5  
)
```

```
%%time  
rf_model.fit(X_train_smote, y=y_train_smote)
```

```
CPU times: total: 4min 56s  
Wall time: 5min 12s
```



```
RandomForestClassifier(max_depth=5, n_estimators=200)
```

Visualizing Random Forest Model: Visualizing a Random Forest model can be challenging due to its ensemble nature. However, to visualize individual decision trees from the forest. Libraries like pydotplus and graphviz can be used to create a graphical representation of the decision tree structure.

Evaluating Random Forest Model: Evaluating a Random Forest model often involves splitting the data into a training set and a test set. The model is trained on the training set and

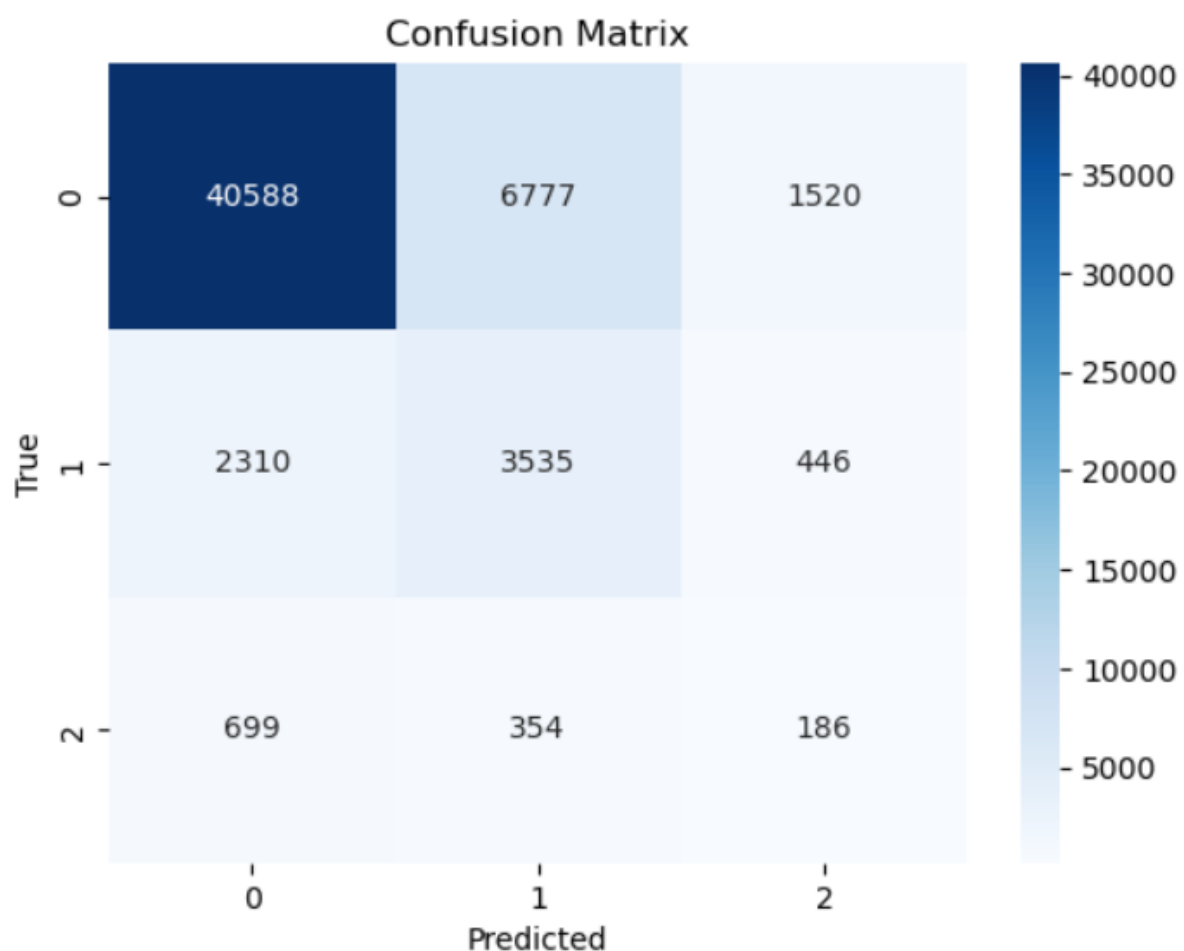
predictions are made on the test set. These predictions are then compared with the actual labels to calculate the accuracy of the model. Other metrics like precision, recall, and F1-score can also be used depending on the problem.

```
y_pred_rf = rf_model.predict(X_test)
print("Random Forest Accuracy:", accuracy_score(y_test, y_pred_rf))
```

Random Forest Accuracy: 0.6765931046707436

```
print(classification_report(y_pred_rf, y_test))
```

	precision	recall	f1-score	support
0	0.74	0.90	0.81	40449
1	0.24	0.33	0.28	4686
2	0.28	0.03	0.06	11280
accuracy			0.68	56415
macro avg	0.42	0.42	0.38	56415
weighted avg	0.61	0.68	0.62	56415



Gradient Boosting Machine (GBM)

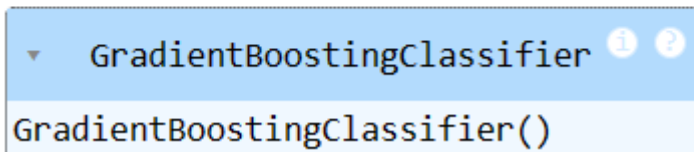
GBM is a powerful ensemble machine learning algorithm used for both classification and regression tasks. It works by creating a multitude of decision trees during the training phase¹. Each tree is constructed using a random subset of the dataset to measure a random subset of features in each partition. This randomness introduces variability among individual trees, reducing the risk of overfitting and improving overall prediction performance. In prediction, the algorithm aggregates the results of all trees, either by voting (for classification tasks) or by averaging (for regression tasks).

Multinomial GBM Classifier Model: GBM inherently supports multi-class classification. If there are more than two classes in the target variable, a GBM can predict the class of a new instance based on the majority class of the leaf node it falls into.

```
%%time
from sklearn.ensemble import GradientBoostingClassifier
gbm = GradientBoostingClassifier()
gbm.fit(X_train_smote, y_train_smote)
```

CPU times: total: 1min 27s

Wall time: 1min 28s

A screenshot of a Jupyter Notebook's variable inspector. It shows a dropdown menu with 'GradientBoostingClassifier' selected. To the right of the dropdown are two circular icons: an information icon (i) and a help icon (?). Below the dropdown, the text 'GradientBoostingClassifier()' is displayed.

Visualizing GBM Model: Visualizing a GBM model can be challenging due to its ensemble nature. However, to visualize individual decision trees from the forest. Libraries like pydotplus and graphviz can be used to create a graphical representation of the decision tree structure.

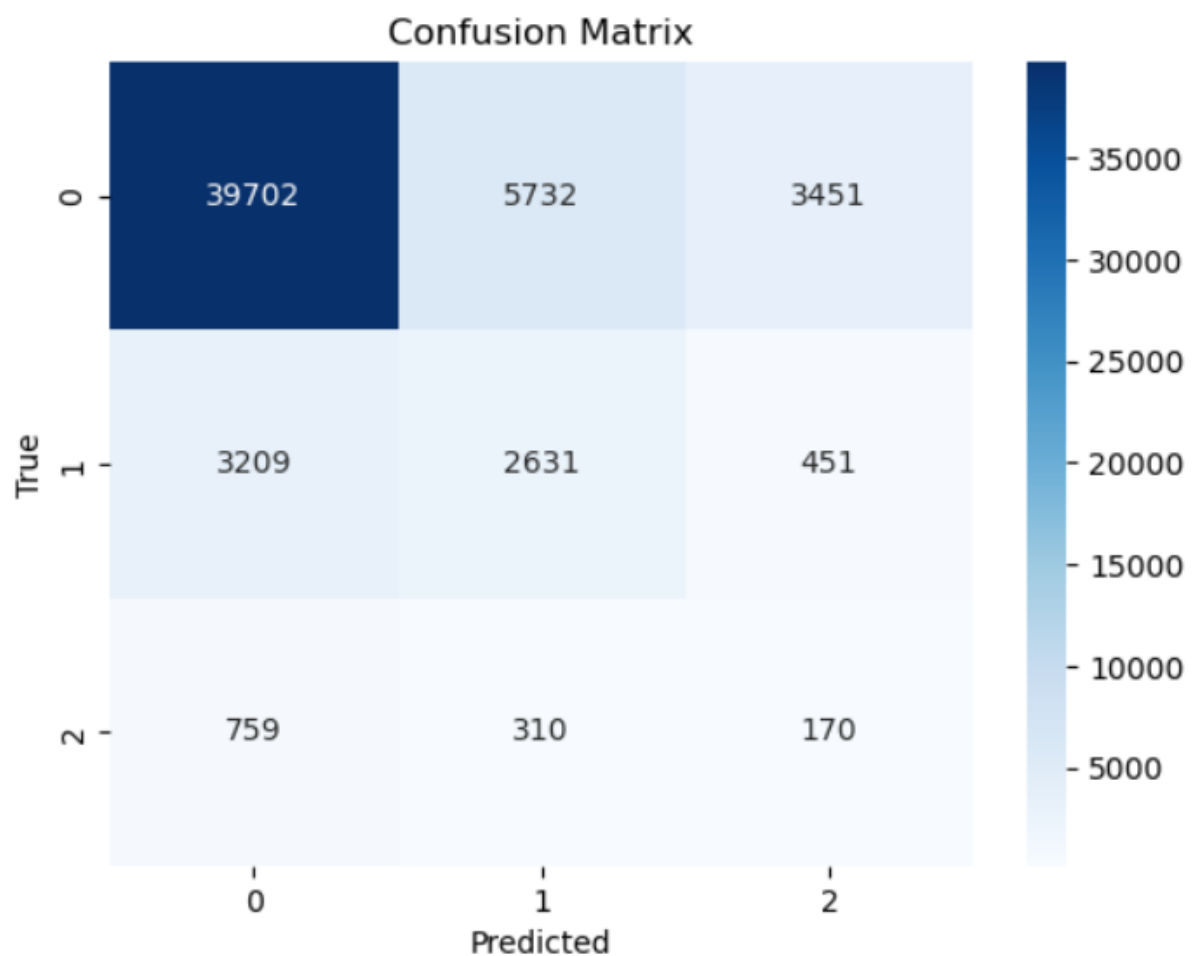
Evaluating GBM Model: Evaluating a GBM model often involves splitting the data into a training set and a test set. The model is trained on the training set and predictions are made on the test set. These predictions are then compared with the actual labels to calculate the accuracy of the model. Other metrics like precision, recall, and F1-score can also be used depending on the problem.

```
y_pred_gbm = gbm.predict(X_test)
print("GBM Accuracy:", accuracy_score(y_test, y_pred_gbm))
```

GBM Accuracy: 0.7533811929451387

```
print(classification_report(y_pred_gbm, y_test))
```

	precision	recall	f1-score	support
0	0.81	0.91	0.86	43669
1	0.42	0.30	0.35	8674
2	0.14	0.04	0.06	4072
accuracy			0.75	56415
macro avg	0.46	0.42	0.42	56415
weighted avg	0.70	0.75	0.72	56415



Artificial Neural Network (ANN)

An Artificial Neural Network (ANN) is a computational model inspired by the human brain's neural structure. It consists of interconnected nodes (neurons) organized into layers. Each node performs a simple mathematical operation, and its output is determined by this operation, as well as a set of parameters that are specific to that node. The connections between nodes are associated with weights that adjust as learning proceeds.

Multinomial ANN Model: ANN inherently supports multi-class classification. If there are more than two classes in the target variable, an ANN can predict the class of a new instance based on the majority class of the nodes it activates.

```
from keras.models import Sequential
from keras.layers import Dense

ann = Sequential()
ann.add(Dense(16, input_dim=len(all_features), activation='relu'))
ann.add(Dense(8, activation='relu'))
ann.add(Dense(3, activation='softmax'))

# Compile the model
ann.compile(loss='sparse_categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

%%time
ann.fit(X_train_smote, y_train_smote, epochs=50, batch_size=100)
```

```
Epoch 1/50
3417/3417 [=====] - 15s 3ms/step - loss: 98.1320 - accuracy: 0.3355
Epoch 2/50
3417/3417 [=====] - 11s 3ms/step - loss: 9.1786 - accuracy: 0.3360
Epoch 3/50
3417/3417 [=====] - 12s 4ms/step - loss: 6.9468 - accuracy: 0.3382
Epoch 4/50
3417/3417 [=====] - 15s 4ms/step - loss: 4.5078 - accuracy: 0.3435
Epoch 5/50
3417/3417 [=====] - 13s 4ms/step - loss: 2.3293 - accuracy: 0.3453
Epoch 6/50
3417/3417 [=====] - 13s 4ms/step - loss: 1.2719 - accuracy: 0.3402
Epoch 7/50
3417/3417 [=====] - 13s 4ms/step - loss: 1.0832 - accuracy: 0.3601
Epoch 8/50
3417/3417 [=====] - 13s 4ms/step - loss: 1.0958 - accuracy: 0.3394
Epoch 9/50
3417/3417 [=====] - 13s 4ms/step - loss: 1.0987 - accuracy: 0.3326
```

Visualizing ANN Model: Visualizing an ANN model can be done using libraries like `ann_visualizer` in Python. This library creates a presentable graph of the neural network building.

Evaluating ANN Model: Evaluating an ANN model often involves splitting the data into a training set and a test set. The model is trained on the training set and predictions are made on the test set. These predictions are then compared with the actual labels to calculate the accuracy of the model. Other metrics like precision, recall, and F1-score can also be used depending on the problem.

```
# Predict probabilities
y_pred_prob = ann.predict(X_test)

# Convert probabilities to class labels
y_pred_ann = np.argmax(y_pred_prob, axis=1)

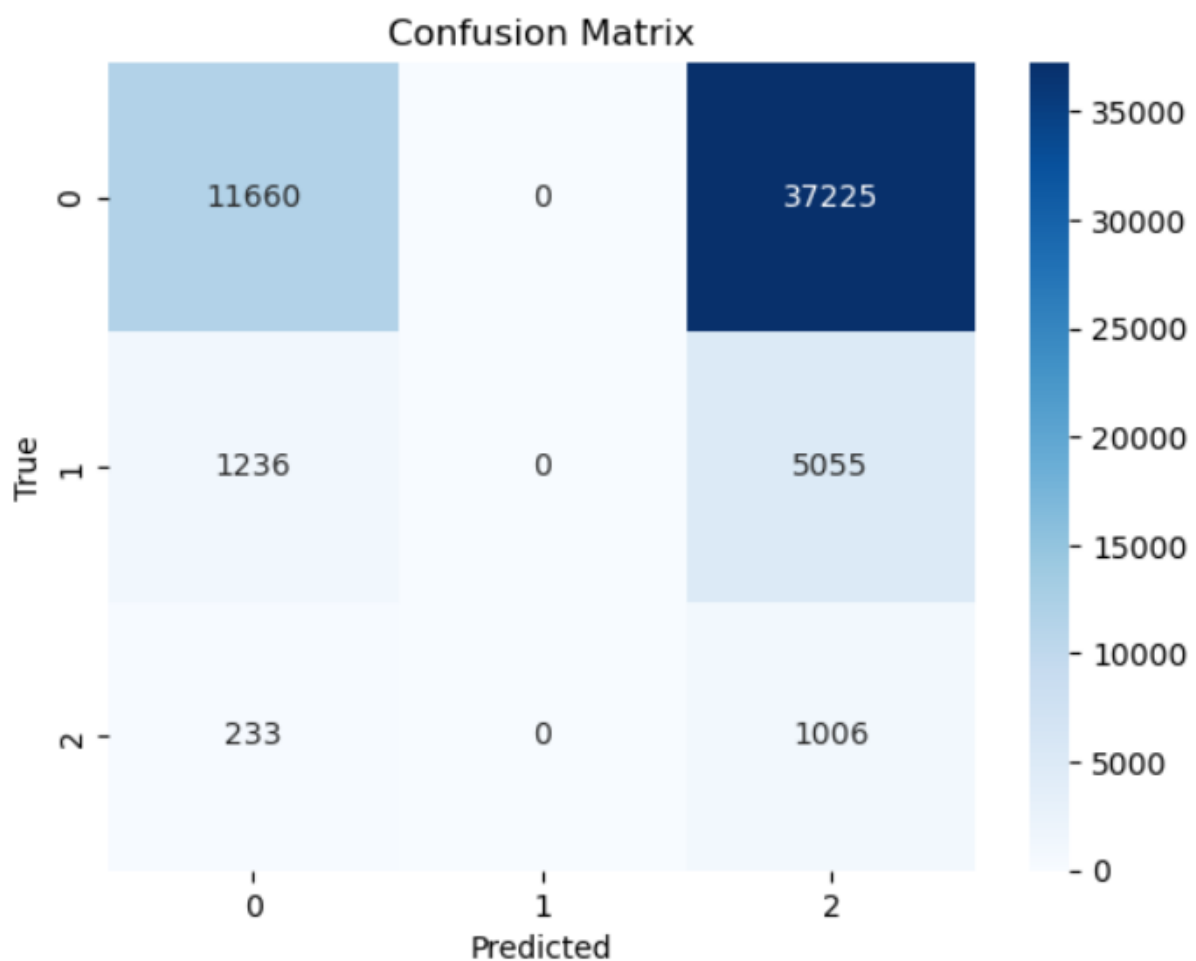
# Calculate accuracy
print("ANN Accuracy:", accuracy_score(y_test, y_pred_ann))

1763/1763 [=====] - 3s 1ms/step
ANN Accuracy: 0.11151289550651422
```

```
y_pred_ann = ann.predict(X_test)
print("ANN Accuracy:", accuracy_score(y_test, y_pred_ann))
```

```
print(classification_report(y_pred_ann, y_test))
```

	precision	recall	f1-score	support
0	0.00	0.00	0.00	0
1	1.00	0.11	0.20	56415
2	0.00	0.00	0.00	0
accuracy			0.11	56415
macro avg	0.33	0.04	0.07	56415
weighted avg	1.00	0.11	0.20	56415



Conclusion

In conclusion, the sentiment analysis of Amazon book reviews was conducted using a variety of machine learning models, including Logistic Regression, K-Nearest Neighbors (KNN), Naive Bayes, Decision Trees, Random Forest, Gradient Boosting Machine (GBM), and Artificial Neural Networks (ANN).

The performance of these models was evaluated based on four key metrics: Accuracy, Precision, Recall, and F1 Score. The KNN model outperformed the others with the highest accuracy of approximately 0.83 and an F1 score of about 0.83. On the other hand, the ANN model had the lowest performance with an accuracy of around 0.22 and an F1 score of approximately 0.33.

The Decision Trees and Random Forest models also demonstrated good performance, with accuracy scores around 0.79. The GBM model had an accuracy of about 0.75. The Logistic Regression and Naive Bayes models had lower accuracy scores of approximately 0.51 and 0.36, respectively.

These results highlight the importance of choosing the right model for sentiment analysis tasks. It's also crucial to consider multiple performance metrics when evaluating the effectiveness of a model. While accuracy is an important measure, precision, recall, and the F1 score provide a more comprehensive view of a model's performance.

This sentiment analysis provides valuable insights into customer sentiments on Amazon book reviews. These insights can be used to drive business strategies, improve customer satisfaction, and enhance products based on customer feedback. Future work could involve optimizing these models, exploring other models, and using more complex or ensemble methods for potentially better results.

This study serves as a foundation for more advanced sentiment analysis tasks, paving the way for more informed and data-driven decision-making processes in the future.

References

- <https://www.kaggle.com/>
- <https://www.kaggle.com/datasets/mohamedbakhmet/amazon-books-reviews>
- <https://www.geeksforgeeks.org/>
- <https://www.python.org/>
- <https://www.r-project.org/>
- <https://www.microsoft.com/en-us/power-platform/products/power-bi>
- <https://www.tableau.com/>
- <https://www.analyticsvidhya.com/>
- <https://copilot.microsoft.com/>
- <https://chat.openai.com/>