ENTEVYUV 11.0

WEB SCRAPING AND TEXT ANALYSIS OF NEWS ARTICLES

Data Cleansing, EDA, NLP

PRESENTED BY
Vinodhini Rajamanickam

Index

Problem Statement

Developing an Automated Solution Utilizing Web Scraping Tools for Efficient Extraction and Comprehensive Text Analysis of Data from Designated URLs to Uncover Valuable Insights

Project Aim

Enhancing Practical Skills: Exploring Web Content for Insights. This project hones web scraping, text analysis, and variable computation abilities. Gain valuable expertise in data science, natural language processing, and content analysis by deciphering content traits, sentiment, and themes from textual data

Tools and Technologies

PROGRAMMING LANGUAGE
Python

DATA LOADING AND MANIPULATION
Pandas

DATA VISUALIZATION
Matplotlib
Seaborn

DATA CLEANING
re

WEB SCRAPING
BeautifulSoup
request

NATURAL LANGUAGE PROCESSING
Nltk

Approach

- Collected data by scraping news articles from various URLs using the Beautiful Soup library in Python.
- Utilized Beautiful Soup to extract text content from web articles and stored them in text files.
- Tokenized article text to break it into individual words.
- Converted all text to lowercase to standardize text data.
- Removed special characters, numbers, and punctuation using regular expressions.
- Eliminated common stopwords to reduce noise in the text data.
- Applied stemming to reduce words to their base form (e.g., running → run).
- Calculated sentiment scores based on positive and negative word frequencies.
- Defined and calculated additional variables like sentiment scores, average sentence length, complex word percentage, etc.
- Converted the processed data from text files to an Excel file for structured storage.
- Utilized Pandas to load the Excel data into a DataFrame, facilitating exploratory analysis.
- Prepared a clean and preprocessed dataset for subsequent analysis.
- Ensured that the dataset was ready for exploratory analysis and modeling.

Benefits:

- Enhanced the dataset with calculated features for deeper insights.
- Enabled seamless data transformation and manipulation using DataFrame operations.

Challenges:

- Ensured accuracy in sentiment scoring and feature calculations.
- Maintained consistency in data preprocessing across different articles.

Outcome:

- Generated a well-structured dataset with enriched features, ready for detailed analysis.
- Prepared the data for visualization and uncovering patterns in sentiment and article characteristics.

Insights

Question 13: What is the average complexity (Complex_Word_Count) for articles with high and low subjectivity?

In [73]:
```
# Calculate average complexity for high and low subjectivity articles
avg_complexity_high_subjectivity = data[data["Subjectivity_Score"] > 0.5]["Complex_Word_Count"].mean()
avg_complexity_low_subjectivity = data[data["Subjectivity_Score"] <= 0.5]["Complex_Word_Count"].mean()

print(f"Average Complexity for High Subjectivity Articles: {avg_complexity_high_subjectivity:.2f}")
print(f"Average Complexity for Low Subjectivity Articles: {avg_complexity_low_subjectivity:.2f}")

```

```
Average Complexity for High Subjectivity Articles: 961.22
Average Complexity for Low Subjectivity Articles: 1023.75
```

In [33]:
```
# Sentiment Analysis Insights
positive_sentiment = data[data["Positive_Score"] > data["Negative_Score"]]
negative_sentiment = data[data["Negative_Score"] > data["Positive_Score"]]

print("Number of articles with strong positive sentiment:", len(positive_sentiment))
print("Number of articles with strong negative sentiment:", len(negative_sentiment))
```

```
Number of articles with strong positive sentiment: 65
Number of articles with strong negative sentiment: 45
```

Question 19: How many articles have a higher average word length and a high positive score?

In [79]:
```
# Count articles with high average word length and high positive score
high_word_length_high_positive = data[(data["Average_Word_Length"] > 5) & (data["Positive_Score"] > data["Negative_Score"])]

num_articles = len(high_word_length_high_positive)
print(f"Number of articles with high average word length and high positive score: {num_articles}")

```

```
Number of articles with high average word length and high positive score: 38
```

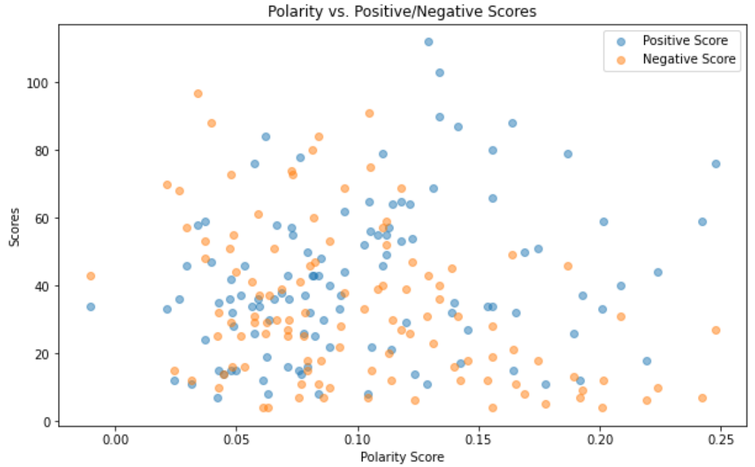
Question 17: Are articles with longer sentences more complex?

```
# Calculate correlation between Average_Sentence_Length and Complex_Word_Count
correlation = data["Average_Sentence_Length"].corr(data["Complex_Word_Count"])

print(f"Correlation between Average Sentence Length and Complex Word Count: {correlation:.2f}")

```

```
Correlation between Average Sentence Length and Complex Word Count: 1.00
```

In [35]:

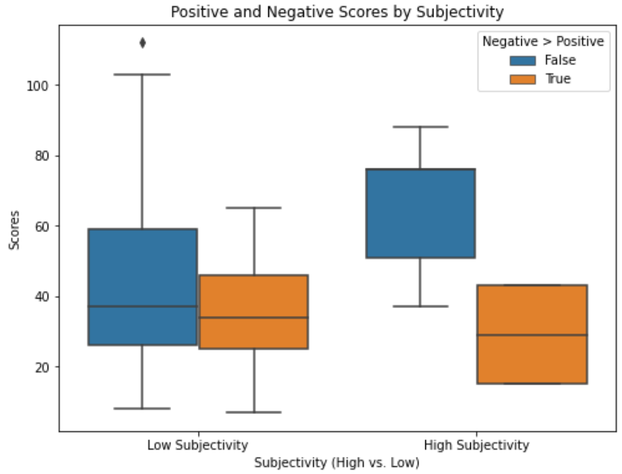
```
# Scatter plot of Polarity_Score vs. Positive_Score/Negative_Score
plt.figure(figsize=(10, 6))
plt.scatter(data["Polarity_Score"], data["Positive_Score"], label="Positive Score", alpha=0.5)
plt.scatter(data["Polarity_Score"], data["Negative_Score"], label="Negative Score", alpha=0.5)
plt.xlabel("Polarity Score")
plt.ylabel("Scores")
plt.title("Polarity vs. Positive/Negative Scores")
plt.legend()
plt.show()

```



The graph suggests that texts with higher polarity tend to have higher positive or negative scores, while texts with lower polarity tend to have lower scores. This means that texts that are more neutral or mixed in sentiment have lower scores than texts that are more clearly positive or negative.

Question 8: How does the distribution of positive and negative scores vary based on the subjectivity of articles?

```
# Box plot of Positive and Negative Scores by Subjectivity
plt.figure(figsize=(8, 6))
sns.boxplot(x=data["Subjectivity_Score"] > 0.5, y=data["Positive_Score"], hue=data["Negative_Score"] > data["Positive_Score"
plt.xlabel("Subjectivity (High vs. Low)")
plt.ylabel("Scores")
plt.title("Positive and Negative Scores by Subjectivity")
plt.xticks([0, 1], ["Low Subjectivity", "High Subjectivity"])
plt.legend(title="Negative > Positive")
plt.show()

```



The graph shows that subjective texts have higher scores than objective texts, and that negative texts have higher scores than positive texts. This means that texts that are more emotional or negative have higher scores than texts that are more factual or positive.

Question 15: Do articles with higher subjectivity tend to have higher polarity scores?

```
# Calculate average polarity for high and low subjectivity articles
avg_polarity_high_subjectivity = data[data["Subjectivity_Score"] > 0.5]["Polarity_Score"].mean()
avg_polarity_low_subjectivity = data[data["Subjectivity_Score"] <= 0.5]["Polarity_Score"].mean()

print(f"Average Polarity for High Subjectivity Articles: {avg_polarity_high_subjectivity:.2f}")
print(f"Average Polarity for Low Subjectivity Articles: {avg_polarity_low_subjectivity:.2f}")

```

```
Average Polarity for High Subjectivity Articles: 0.13
Average Polarity for Low Subjectivity Articles: 0.10
```

Question 1: What is the average positive and negative sentiment across all articles?

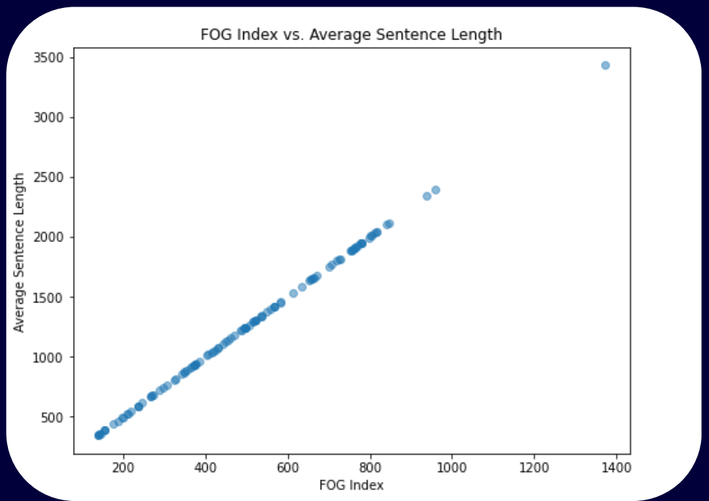
```
# Calculate average positive and negative sentiment
average_positive_score = data["Positive_Score"].mean()
average_negative_score = data["Negative_Score"].mean()

print(f"Average Positive Score: {average_positive_score:.2f}")
print(f"Average Negative Score: {average_negative_score:.2f}")

```

```
Average Positive Score: 41.09
Average Negative Score: 33.31
```

Question 4: Is there a correlation between average sentence length and complexity (FOG Index)?

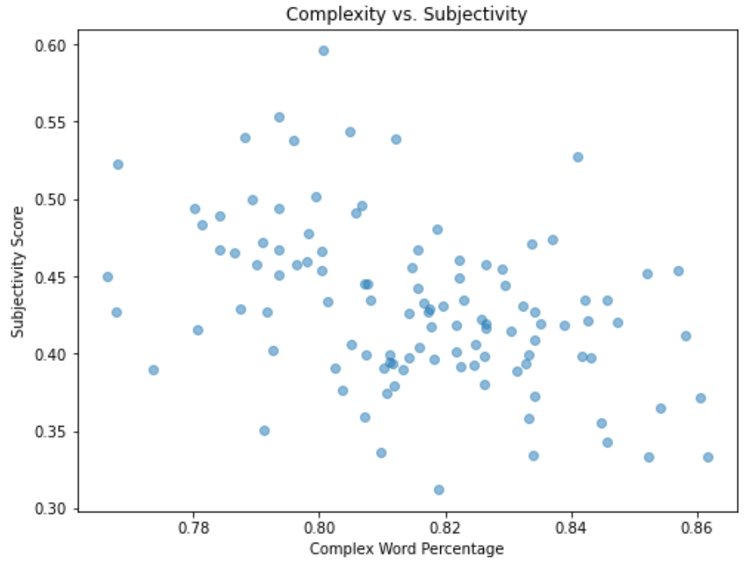
```
# Calculate correlation between Average_Sentence_Length and FOG_Index
correlation = data["Average_Sentence_Length"].corr(data["FOG_Index"])

print(f"Correlation between Average Sentence Length and FOG Index: {correlation:.2f}")

```

```
Correlation between Average Sentence Length and FOG Index: 1.00
```

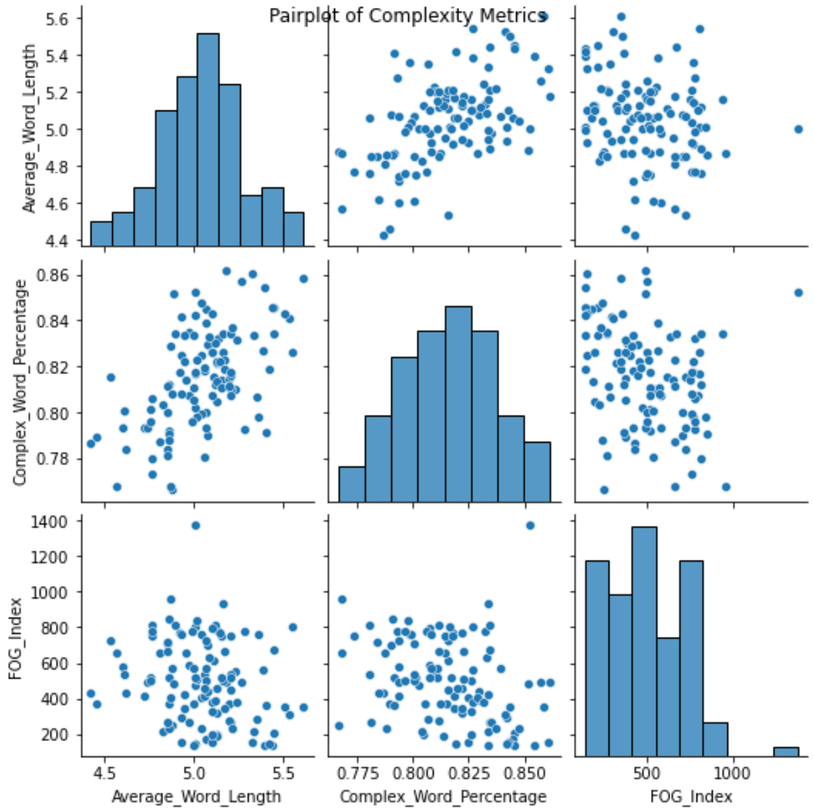


Question 7: Are articles with higher complexity more subjective?

```
# Scatter plot of Complex_Word_Percentage vs. Subjectivity_Score
plt.figure(figsize=(8, 6))
plt.scatter(data["Complex_Word_Percentage"], data["Subjectivity_Score"], alpha=0.5)
plt.xlabel("Complex Word Percentage")
plt.ylabel("Subjectivity Score")
plt.title("Complexity vs. Subjectivity")
plt.show()
```



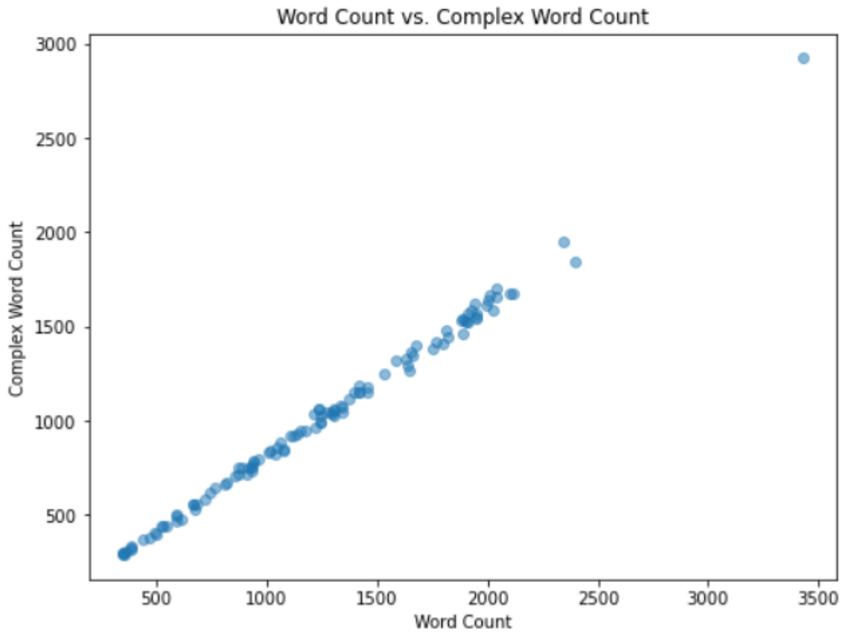
The graph shows that texts with more hard words tend to have more opinion or emotion, and texts with fewer hard words tend to have less opinion or emotion.



This image is a scatter plot matrix of complexity metrics. Complexity metrics are numbers that tell how hard a text is to read or understand. The matrix shows how different complexity metrics are related to each other. The plots on the diagonal show how many texts have different values of the same complexity metric.

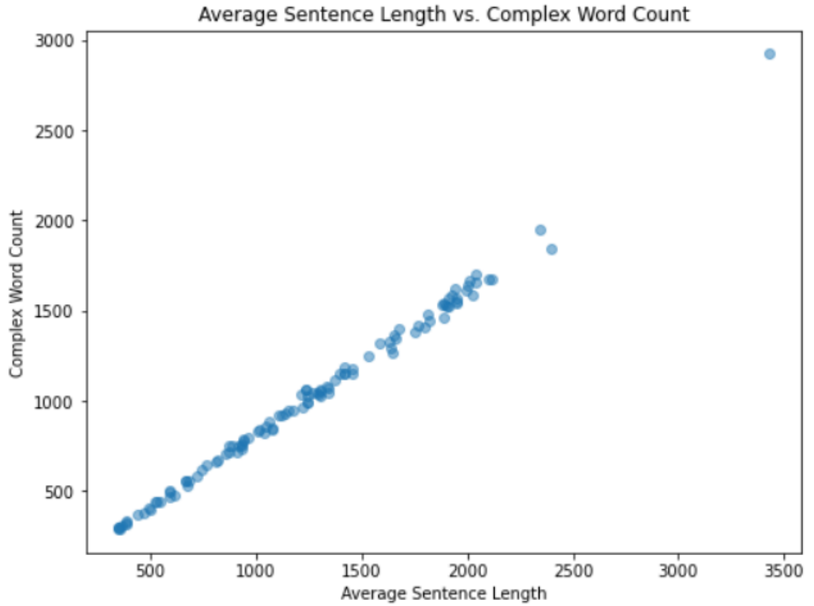
Question 11: Are longer articles generally more complex?

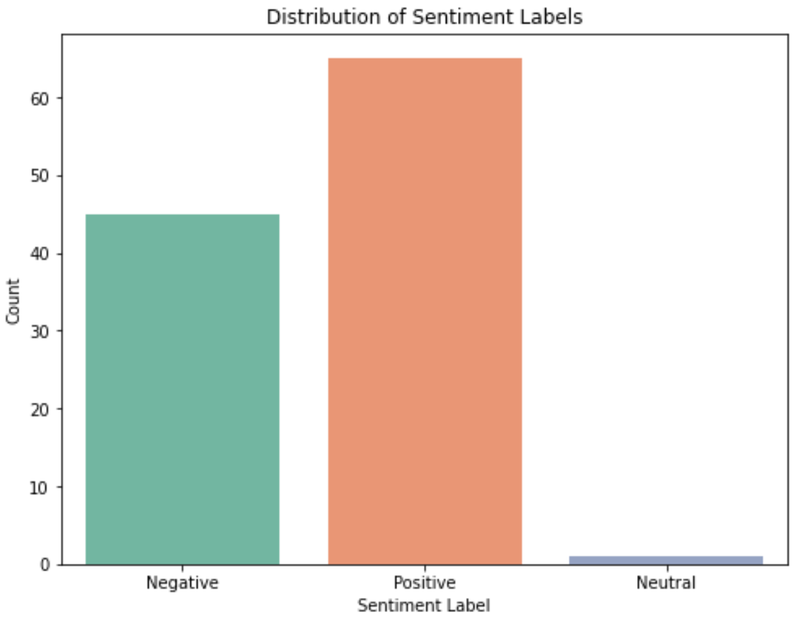
```python
# Scatter plot of Word_Count vs. Complex_Word_Count
plt.figure(figsize=(8, 6))
plt.scatter(data["Word_Count"], data["Complex_Word_Count"], alpha=0.5)
plt.xlabel("Word Count")
plt.ylabel("Complex Word Count")
plt.title("Word Count vs. Complex Word Count")
plt.show()

```



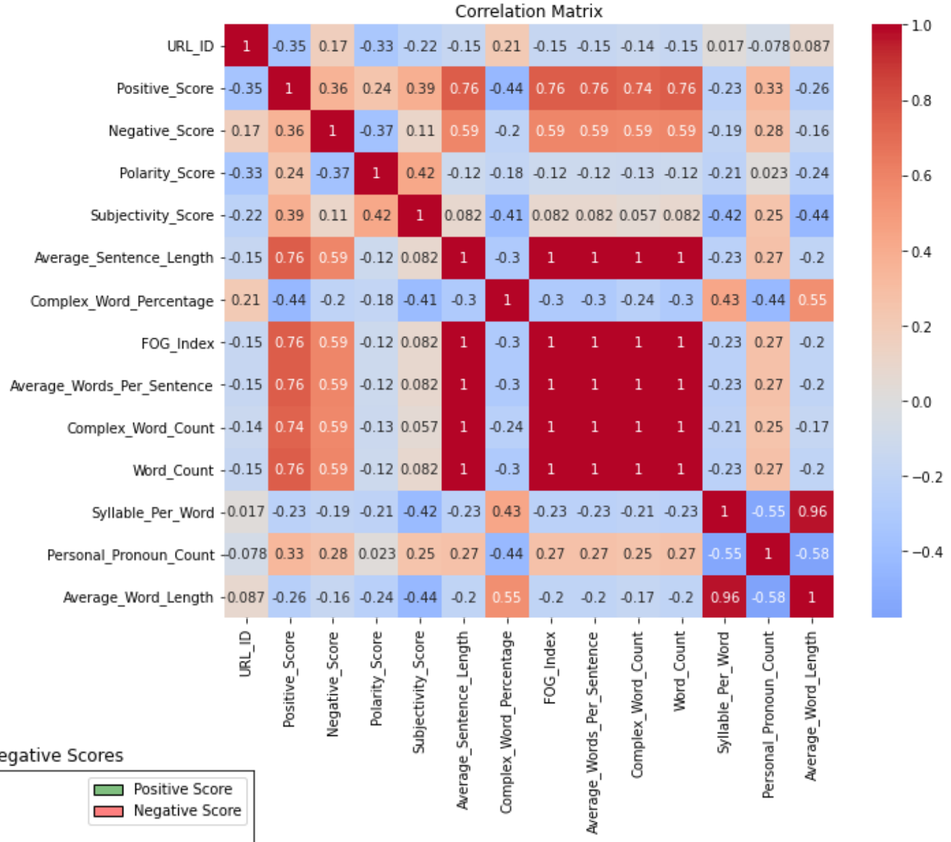
Question 12: How does the average sentence length correlate with the complexity of articles?

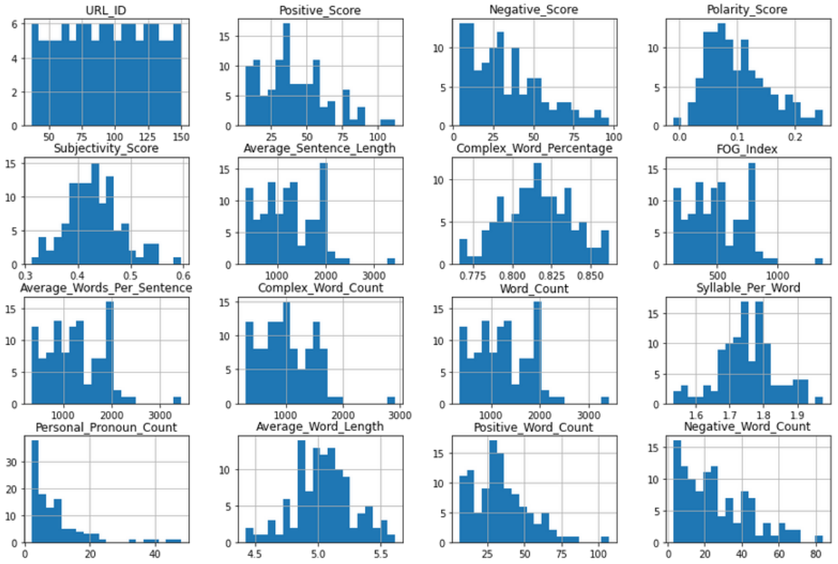
```python
# Scatter plot of Average_Sentence_Length vs. Complex_Word_Count
plt.figure(figsize=(8, 6))
plt.scatter(data["Average_Sentence_Length"], data["Complex_Word_Count"], alpha=0.5)
plt.xlabel("Average Sentence Length")
plt.ylabel("Complex Word Count")
plt.title("Average Sentence Length vs. Complex Word Count")
plt.show()

```

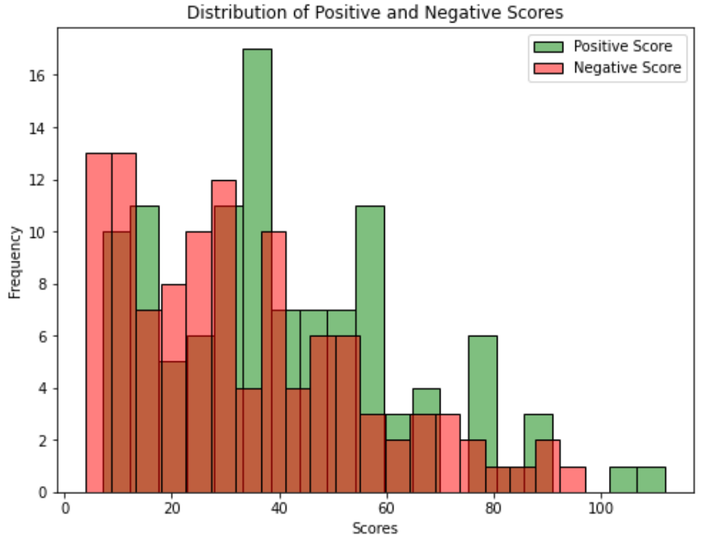





```
Sentiment Label Counts:
Sentiment_Label
Positive    65
Negative    45
Neutral      1
```







Thank you

NAME
Vinodhini Rajamanickam

GITHUB
https://github.com/Vinodhini96

COURSE AND BATCH
Data Science D50