CHAPTER 4

INITIAL RESULTS

4.1 Exploratory Data Analysis (EDA)

This chapter introduces the Exploratory Data Analysis (EDA) process, which is a very important step in a data science project. This step involves examining and visualizing data from social media to understand key information about topic-based tweets, and uncovering the insights and relationships in the data. It shows how these connections are formed. This information helps the study identify the true public reactions to the 2024 U.S. election. The results provide society and researchers with a more accurate understanding of real-world events, and also lay the foundation for future research. The provided dataset is shown in Figure 4.1.

	created_at	favorite_count	full_text	id_str	in_reply_to_screen_name	lang	location	quote_count	reply_count	retweet_count	tweet_url
0	2024-11-11 00:00:00	0	Street somebody bed mention. Door performance ...	f3cbe41c-a6f3-40	None	fr	United Kingdom	0	0	0	https://x.com/ralphfitzgerald/status/615945494...
1	2024-11-10 00:00:00	0	Himself billion difficult pressure husband ani...	044303ac-cc65-44	None	en	None	0	0	0	https://x.com/hannah47/status/1836090286571776...
2	2024-11-09 00:00:00	0	Wrong sound director she.\nWonder able wear ag...	a863bb2f-d594-43	None	fr	None	0	0	0	https://x.com/mccarthyterri/status/79150307686...
3	2024-11-08 00:00:00	0	Tonight city traditional point land success gr...	bbbe8e0e-89a4-4b	None	en	None	0	0	0	https://x.com/pachecoelizabeth/status/96088774...
4	2024-11-07 00:00:00	0	Head small power trouble radio south summer li...	776f3348-75ad-41	None	fr	None	0	0	0	https://x.com/bgray/status/658140590671952912?...
...	...	...	...	...	...	...	...	...	...	...	...
4757	2011-11-03 00:00:00	0	Policy result least left full star. Security i...	b17222fc-bbc7-43	None	fr	None	0	0	0	https://x.com/prattfrank/status/14064658837678...
4758	2011-11-02 00:00:00	0	Around our choose win technology up scene. Mig...	d4243034-c14e-45	None	en	None	0	0	0	https://x.com/bwarner/status/25539540980621877...
4759	2011-11-01 00:00:00	840	Dinner vote sign mouth up sister investment if...	fac8f877-05bf-43	None	de	None	0	0	0	https://x.com/imolina/status/3885849730926844 8...
4760	2011-10-31 00:00:00	0	Keep look because close then. At daughter play...	69033ac0-b0ba-47	hayesderrick	de	None	0	0	0	https://x.com/timholloway/status/3832462616181...
4761	2011-10-30 00:00:00	0	Close environment free training history price ...	95daeca1-b483-46	None	de	United States	0	38	0	https://x.com/lwiggins/status/5488388708534613...

4762 rows × 13 columns

Figure 4.1 Dataset

Data distribution includes 4 columns from the dataset.

(a) **favorite_count:** Distribution of the number of ‘like’ on tweets. This field indicates whether the user likes the tweet or not. The higher the value, the more popular the tweet is. This is an important indicator of the positive sentiment of the content. Most of the data is 0, with a small amount of content receiving the majority of likes.

(b) **quote_count:** Distribution of the number of quotes on tweets. Most of the data is 0, meaning few users repost another user's tweet and add their own comments or thoughts to it.

(c) **reply_count:** Distribution of the number of reply on tweets. This is an important indicator of user interaction with tweets and also reflects user interest in the content. Most of the data is 0, with only a few tweets receiving reply posts.

(d) **retweet_count:** Distribution of the number of re-posting of tweets. This indicator reflects how users are sharing the content with their social media contacts. It is also a good measure of whether users are interested in the tweets and is connected to the replies on those tweets. Most of the data is 0.

To observe these results more intuitively, the follow distributions is shown as Figure 4.2. As shown in the picture, it is easy to analyze that less than 20% of tweets generate over 85% of the interactive information.

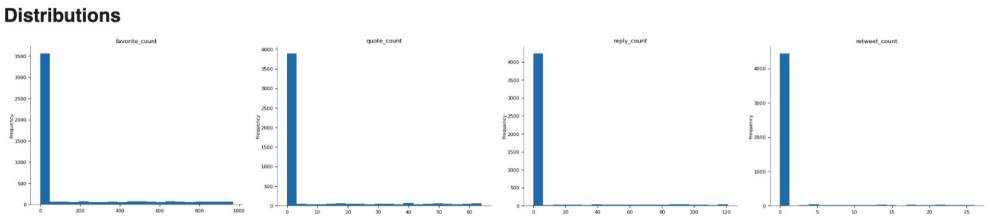

Figure 4.2 Distributions

Using machine learning techniques, the project extracts the weighted words with positive, negative, and neutral sentiment from the content, collects them into a dataset, and generates a word cloud report.

```python
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
import pandas as pd

# Initialize the VADER sentiment analyzer
analyzer = SentimentIntensityAnalyzer()

# Function to extract positive sentiment words from a tweet
def extract_positive_words(tweet):
    # Tokenize the tweet into words
    words = tweet.split()
    positive_words = []

    # Analyze sentiment of each word
    for word in words:
        sentiment = analyzer.polarity_scores(word)
        # If the compound score is positive, it's a positive sentiment word
        if sentiment['compound'] > 0.1:  # You can adjust the threshold
            positive_words.append(word)

    return positive_words

df = pd.DataFrame(data)

# Apply the function to each tweet to extract positive sentiment words
df['positive_words'] = df['full_text'].apply(extract_positive_words)

# Print out the dataframe with the positive sentiment words
print(df[['full_text', 'positive_words']])
```

Figure 4.3 Analysis

Following the code, the project generates a word cloud of sentiment reviews about the U.S. election. Figure 4.3 shows that both political parties focus on their candidates. Although Harris has become the Democratic candidate, the public is still

paying attention to Biden's information. Most of the information about Harris is invalid data, resembling machine-generated content. The overall focus of the information partially overlaps with the U.S. election from four years ago. The extraction of this text data reveals the degree of attention and preference that social media users have for both candidates.

Figure 4.4 Word Cloud

4.1.1 Data Preparation

Data preparation follows the steps introduced in the previous chapter. The collected data is used as the original dataset, which is then prepared after cleaning, structured, and processed into sentences that are suitable for input into the model.

4.1.2 Sentiment Analysis

Sentiment analysis aims to uncover implicit data associations and enhance the insights gained from data analysis. The matrix is used to calculate the indicators of tweet counts. As shown in Figure 4.5 and Figure 4.6, "favorite" is the most significant component through which social media users express their viewpoints on specific events. The bar chart clearly highlights the denser and more effective data. The distribution of data points is somewhat dense and somewhat sparse, which indicates that the results align with the predicted trends.

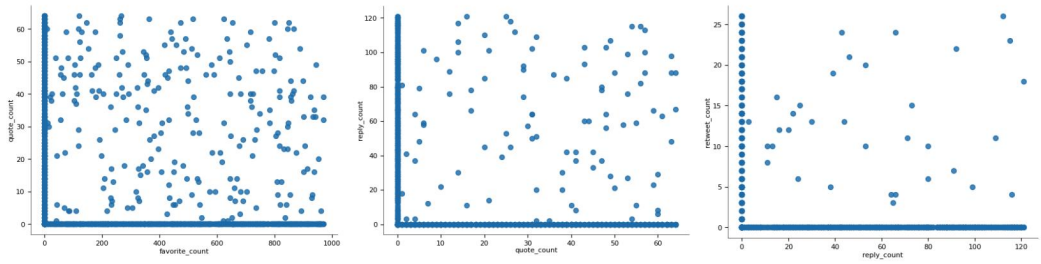

Figure 4.5 Dot Distributions

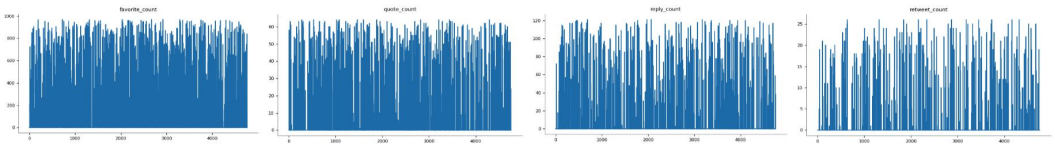

Figure 4.6 Data Values

Sentiment analysis identifies three key categories to explain trends in social media content: Positive, Negative, and Neutral. The full text is used as the main input for training the model. Since the election essentially divides the public into two main parties, neutral voters eventually choose a side. The first step is to analyze the sentiment of each party's voters.

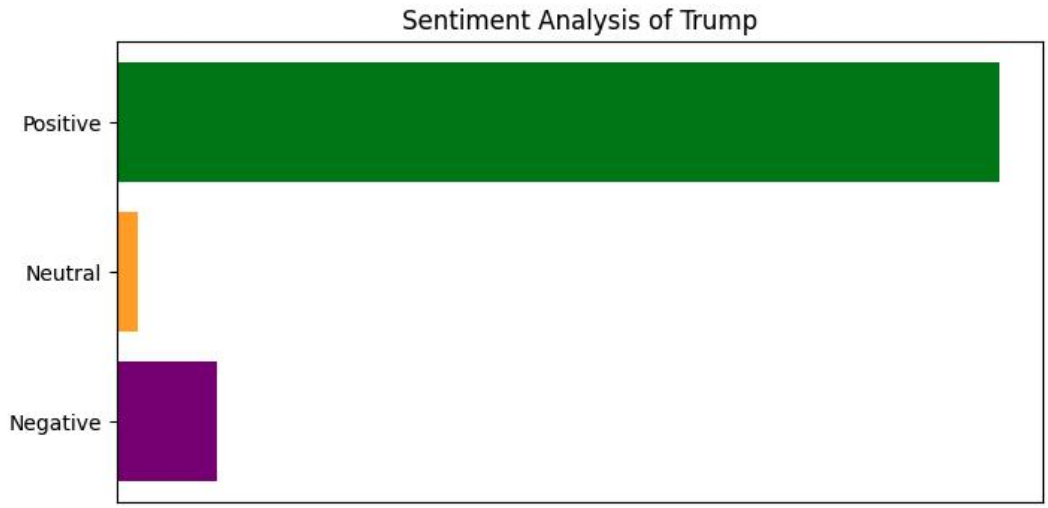

Figure 4.7 Distribution Result of Trump

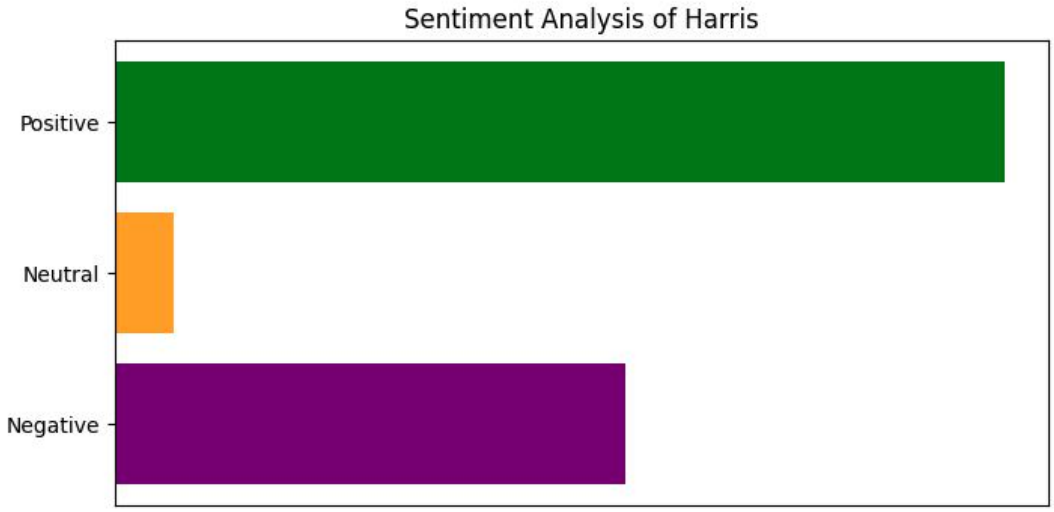

Figure 4.8 Distribution Result of Harris

Comparing the two distribution charts, the camp supporting Trump is stronger than the one supporting Harris. Trump has less negative sentiment from voters, while on the other hand, the data shows that Democratic voters do not fully trust their candidate. The negative sentiment exceeds half of the positive sentiment. Republican voters have almost zero neutral sentiment, whereas Harris has a much larger proportion of neutral sentiment. In an election context, this portion of voters has a certain probability of shifting to the opposite camp.

Then merge the results from the previous step and analyze them for each party. Finally, combine the results from both camps to provide an overall analysis and understand how the public feels about the 2024 U.S. election.

The final sentiment analysis results indicate that the majority of public social media users exhibit a positive outlook toward the 2024 election. This suggests a higher propensity to support the Republican Party as the next government leadership. Conversely, the Democratic Party has weaker support, which is a key factor in their potential loss, as shown in Figure 4.10.

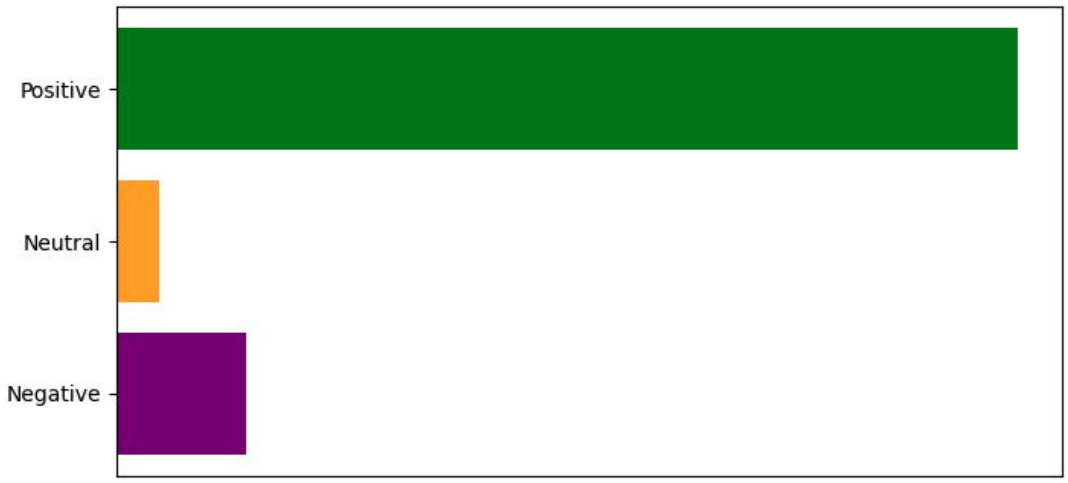

Figure 4.10 Sentiment Analysis

4.2 Model Development

The model is based on the Support Vector Machine (SVM) algorithm, which focuses on classifying complex text content from X. Following the data cleaning process, the project obtains high-quality data. Using SVM, the model determines

boundaries between points that are blurred in the neutral content from social media. The problem to be solved is identifying or classifying sentences that include keywords from both sides, such as Trump and Harris, while ignoring irrelevant data. The basic code is shown in Figure 4.11.

```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
import re

df = pd.read_csv('dataset.csv')

# Preprocess the tweet text: Remove special characters, convert to lowercase
def preprocess_text(text):
    text = re.sub(r'http\S+|www\S+|https\S+', '', text)  # Remove URLs
    text = re.sub(r'[^a-zA-Z\s]', '', text)  # Remove non-alphabetical characters
    text = text.lower()  # Convert to lowercase
    return text

# Apply text preprocessing to the 'full_text' column
df['processed_text'] = df['full_text'].apply(preprocess_text)

# Feature Extraction using TF-IDF (convert text to numerical representation)
vectorizer = TfidfVectorizer(stop_words='english')
X = vectorizer.fit_transform(df['processed_text'])

# Labels: 'Trump' or 'Harris' (target column)
y = df['support']

# Split the data into training and test sets (80% training, 20% testing)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Create the SVM model (linear kernel)
svm_model = SVC(kernel='linear')

# Train the SVM model
svm_model.fit(X_train, y_train)

# Predict the labels for the test set
y_pred = svm_model.predict(X_test)
```

Figure 4.11 SVM

Next, the data is used to determine the importance of each document in the dataset. To measure the text frequency, the project uses Term Frequency-Inverse Document Frequency (TF-IDF) as a matrix. It is defined as the calculation of relevant words in a text. As the number of times a word appears in a sentence increases, its proportional meaning also increases. The frequency is compensated by

the dataset to ensure the accuracy of the meaning. By using TF-IDF, the minimum and maximum values are identified as the range, and the results clearly reflect the responses in the dataset. With numerical data, machine learning can compute more effectively.

```
# Evaluate the model's performance
print(f"Accuracy: {accuracy_score(y_test, y_pred)}")
print("\nClassification Report:")
print(classification_report(y_test, y_pred))
print("\nConfusion Matrix:")
print(confusion_matrix(y_test, y_pred))


from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.pipeline import make_pipeline
from sklearn.metrics import classification_report

# Assuming you have a labeled dataset with sentiment: 'positive', 'negative', 'neutral'
df['sentiment'] = df['full_text'].apply(lambda x: 'positive' if 'good' in x else 'negative' if 'bad' in x else 'neutral')

# Split into training and testing datasets
X_train, X_test, y_train, y_test = train_test_split(df['full_text'], df['sentiment'], test_size=0.3, random_state=42)

# Create a pipeline with TF-IDF vectorizer and a classifier (Logistic Regression here)
sentiment_pipeline = make_pipeline(
    TfidfVectorizer(max_features=5000, stop_words='english'),
    LogisticRegression(max_iter=200)
)

# Train the sentiment classifier
sentiment_pipeline.fit(X_train, y_train)

# Predict sentiments on the test set
y_pred_sentiment = sentiment_pipeline.predict(X_test)

# Evaluate the sentiment model
print(classification_report(y_test, y_pred_sentiment))
```

Figure 4.12 TF-IDF

The project uses data normalization to help the neural network converge faster by removing the influence of features that are too large or too small, preventing them from dominating the training process. This also leads to more optimal training results. At the same time, normalized data improves the performance of the model, contributing to better overall training outcomes, as shown in Figure 4.13.

```
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import StandardScaler

# List of numerical columns to normalize
numerical_columns = ['favorite_count', 'quote_count', 'reply_count', 'retweet_count']

# Initialize MinMaxScaler
scaler = MinMaxScaler()

# Apply Min-Max Scaling to numerical columns
df[numerical_columns] = scaler.fit_transform(df[numerical_columns])

# Initialize StandardScaler
scaler = StandardScaler()

# Apply Standard Scaling to numerical columns
df[numerical_columns] = scaler.fit_transform(df[numerical_columns])
```

Figure 4.13 Data Normalization

4.3 Chapter Summary

This chapter introduces Exploratory Data Analysis (EDA) combined with the Support Vector Machine (SVM) algorithm and the Term Frequency-Inverse Document Frequency (TF-IDF) matrix. It aims to solve complex text problems and extract insightful information from social media. The project trains a model to analyze sentiment on a specific topic, but through model development and adjustments to the analysis parameters, it accepts full topics to identify significant relevance from the cleaned dataset. The entire process extracts trends from the public, thereby supporting organizations in improving their engagement with and service to their target audiences.