NEXUS INFO

ROLE: DATA ANALYST

PROJECT 3: TWITTER SENTIMENT ANALYSIS

Deekshitha Dommati

INTRODUCTION

Project Overview:

Social media platforms like Twitter have become invaluable sources of real-time public opinion and sentiment. Understanding and analyzing the sentiments expressed in tweets can provide deep insights into public perception, trends, and reactions to events, products, or services. This project focuses on leveraging data analytics techniques to conduct sentiment analysis on a dataset of tweets, aiming to extract meaningful patterns and insights from the vast amount of unstructured text data.

Twitter, with its vast user base and diverse content, presents a unique opportunity to gauge public sentiment at scale. By analyzing tweets, we can uncover not only the prevalence of positive, negative, and neutral sentiments but also the specific topics, themes, and emotions that resonate most strongly with users. Such insights are invaluable for businesses, marketers, policymakers, and researchers alike, helping them make informed decisions, identify emerging trends, manage brand reputation, and understand customer sentiment

Objective:

- **Data Exploration and Cleaning:** Understanding the dataset's structure, handling missing values, and ensuring data quality are crucial initial steps.
- **Exploratory Data Analysis (EDA):** Visualizing sentiment distributions, word frequencies, and temporal patterns provides a foundational understanding of the dataset.
- **Text Preprocessing:** Cleaning and preparing tweet text through techniques like tokenization and lemmatization is essential for accurate sentiment analysis.
- **Sentiment Prediction Model:** Building and evaluating a machine learning model to predict sentiment from tweet text, providing quantitative insights into sentiment classification.
- **Feature Importance:** Analyzing which words or phrases contribute most significantly to sentiment predictions, offering qualitative insights into user sentiment and language usage.

METHODOLOGY

PHASE 1: DATA EXPLORATION AND CLEANING

Data Overview:

- The dataset was sourced from a CSV file containing a large volume of tweets.
- Initial exploration involved loading the dataset using pandas and examining its structure with methods like info () and head () to understand its columns and format.
- Column renaming was performed for clarity, with adjustments made to ensure uniformity across variables like sentiment labels and tweet content.

Data Cleaning:

- Addressed data quality issues such as missing values and duplicate entries to maintain dataset integrity.
- A key part of the cleaning process involved preprocessing tweet content. This included removing URLs, mentions, hashtags, and special characters using regular expressions to prepare the text for subsequent analysis.
- By standardizing the data format and eliminating noise from tweet content, the dataset became more suitable for effective sentiment analysis.

PHASE 2: EXPLORATORY DATA ANALYSIS (EDA)

Insights Gained:

- **Sentiment Distribution:** Visualized the distribution of sentiment labels ('positive', 'negative', 'neutral') using bar plots. This revealed that positive sentiments were predominant, reflecting a general trend observed in social media sentiment analysis.
- **Word Frequency Analysis:** Utilized word clouds to visually represent the most frequently occurring words in both positive and negative sentiment tweets. This qualitative approach highlighted common themes and emotional expressions prevalent among Twitter users.
- **Temporal Analysis:** Investigated how sentiment varied over time by analyzing tweet timestamps. This analysis revealed temporal trends and fluctuations in sentiment, which could be linked to specific events or periods of heightened activity on Twitter.

PHASE 3: TEXT PREPROCESSING

Text Processing:

- Implemented text preprocessing techniques to clean and prepare tweet text for sentiment analysis. This included tokenization, where tweets were split into individual words or tokens, and subsequent removal of stop words (commonly occurring words like 'and', 'the', 'is') to focus on meaningful content.
- Lemmatization was applied to reduce words to their base or root form, ensuring consistency in word representation across the dataset. These preprocessing steps enhanced the quality of text data and facilitated more accurate sentiment analysis.

PHASE 4: SENTIMENT PREDICTION MODEL

Model Development:

- Employed a supervised learning approach to build a sentiment prediction model using logistic regression, a commonly used algorithm for binary classification tasks like sentiment analysis.
- The dataset was divided into training and testing sets to evaluate model performance. Text data was vectorized using CountVectorizer, converting tweet text into numerical feature vectors suitable for machine learning algorithms.

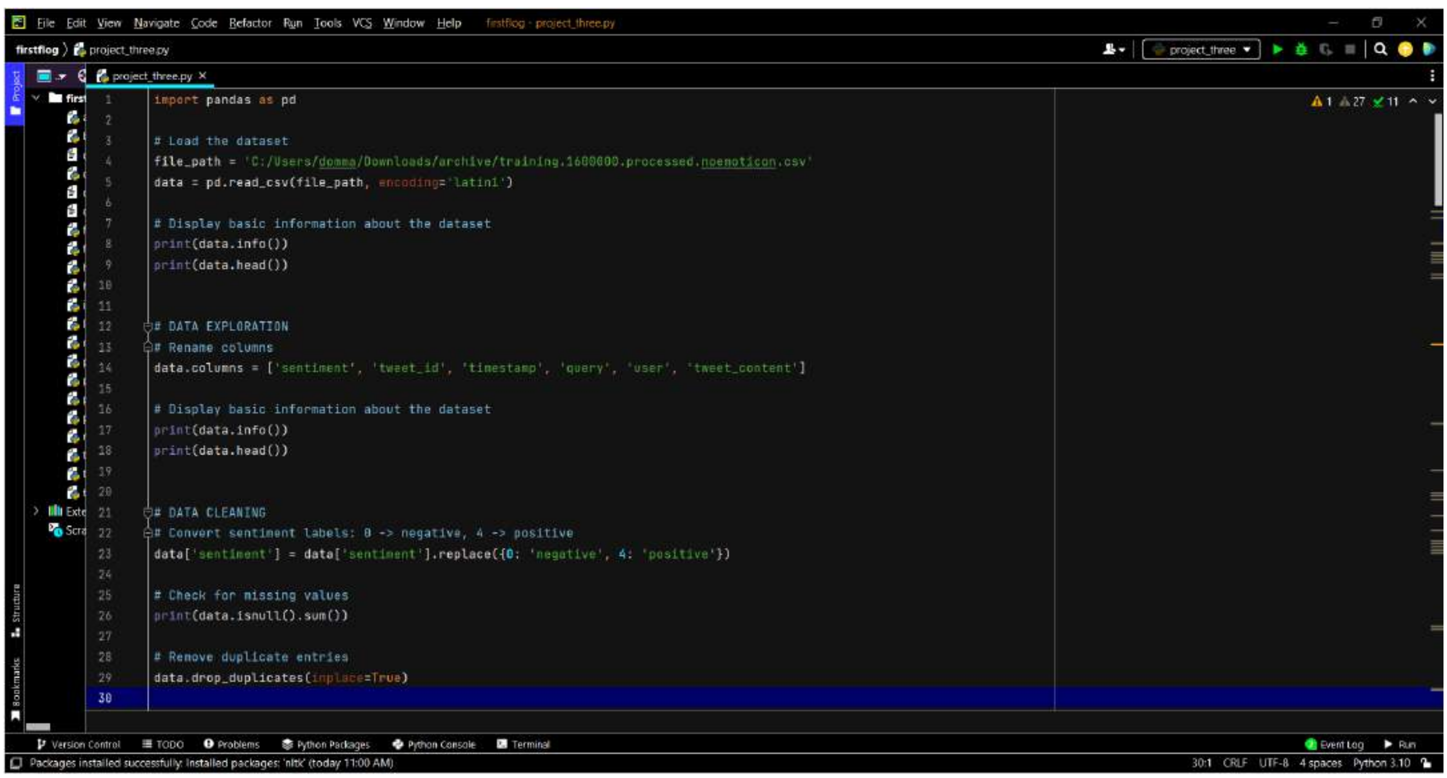
- Trained the logistic regression model on the training set and assessed its performance using metrics such as accuracy and F1 score. These metrics provided quantitative measures of how well the model predicted sentiment labels ('positive' or 'negative') based on tweet content.

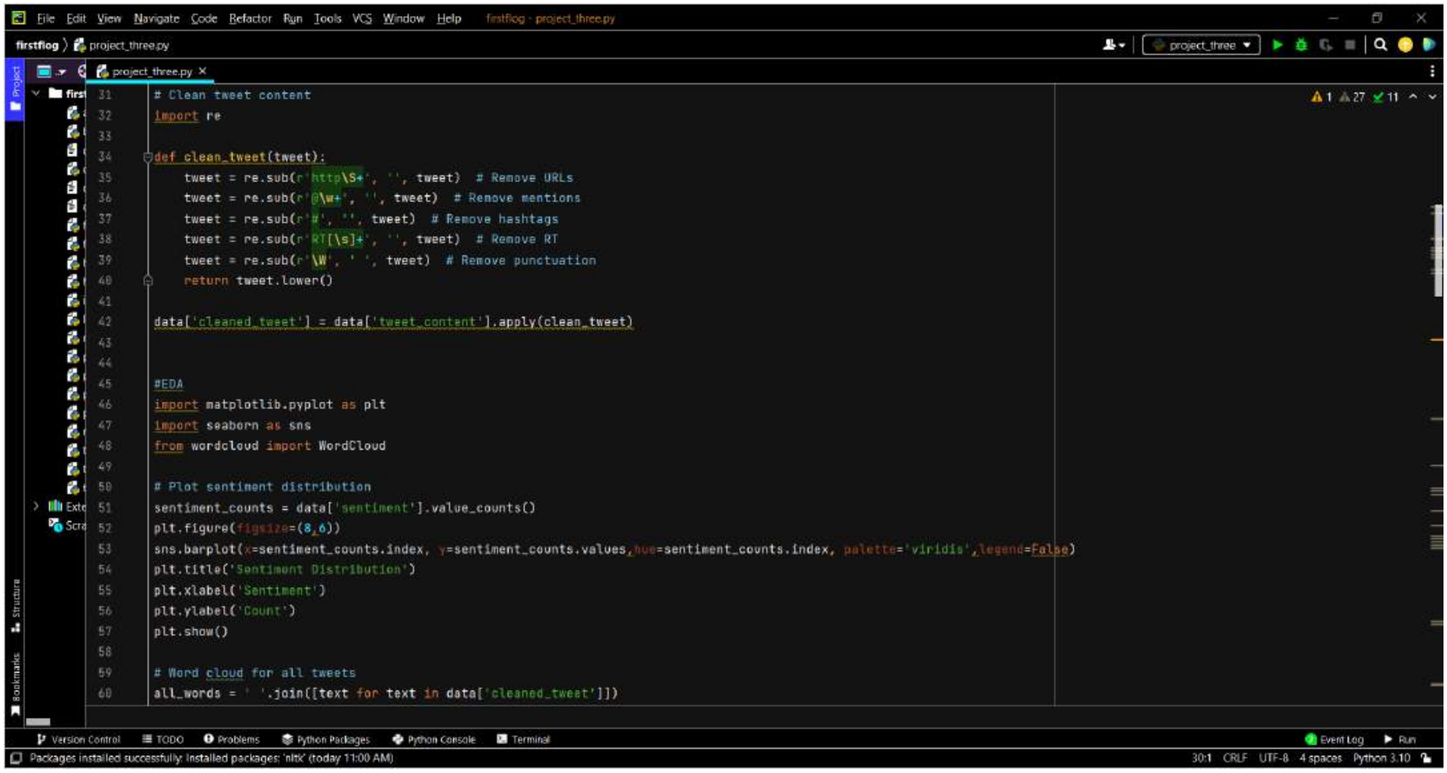
PHASE 5: FEATURE IMPORTANCE

Feature Analysis

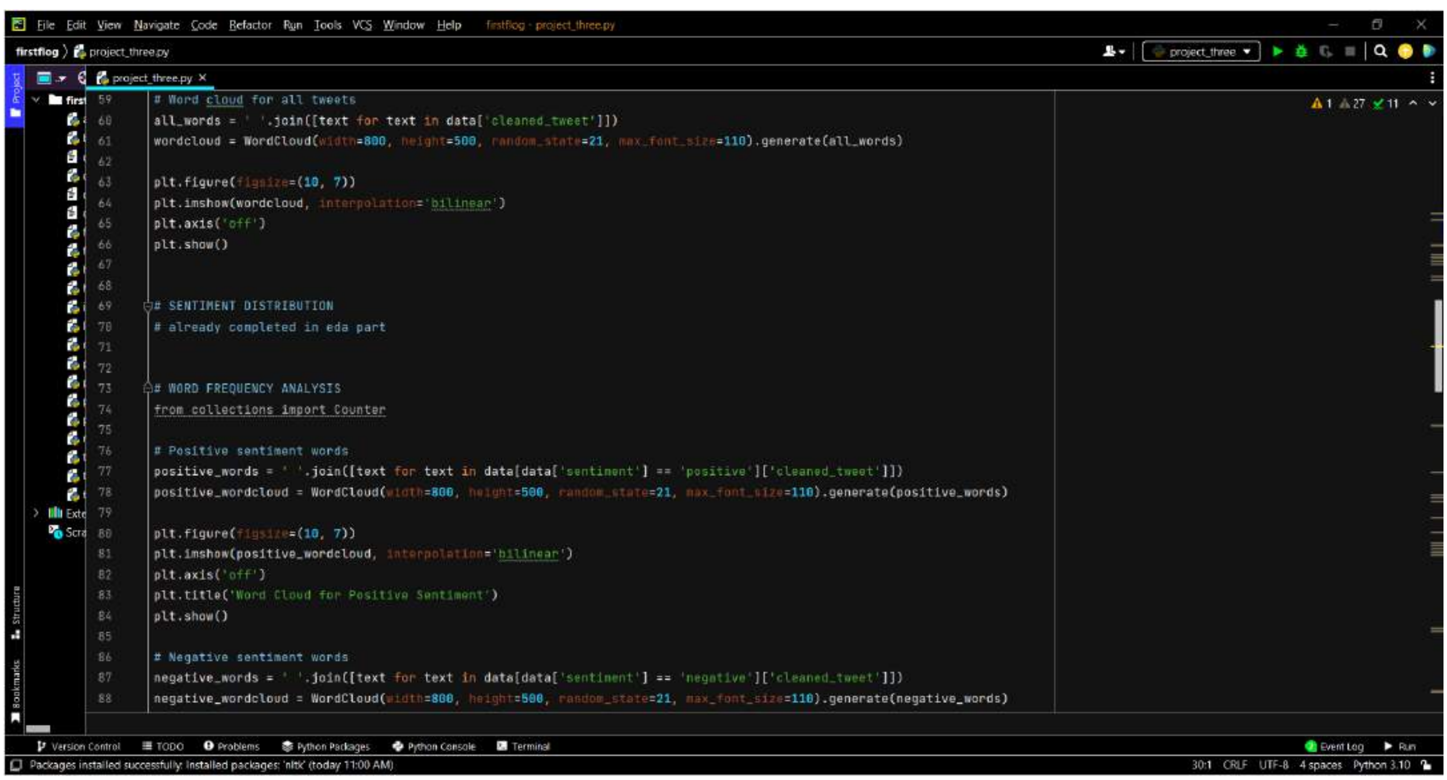
- Investigated feature importance within the sentiment prediction model to identify key words or terms contributing significantly to sentiment predictions.
- Visualized feature importance using techniques like bar plots, showcasing the top words that influenced sentiment classification decisions. This analysis provided insights into the linguistic cues and expressions that strongly correlated with positive or negative sentiment in tweets.

CODE:





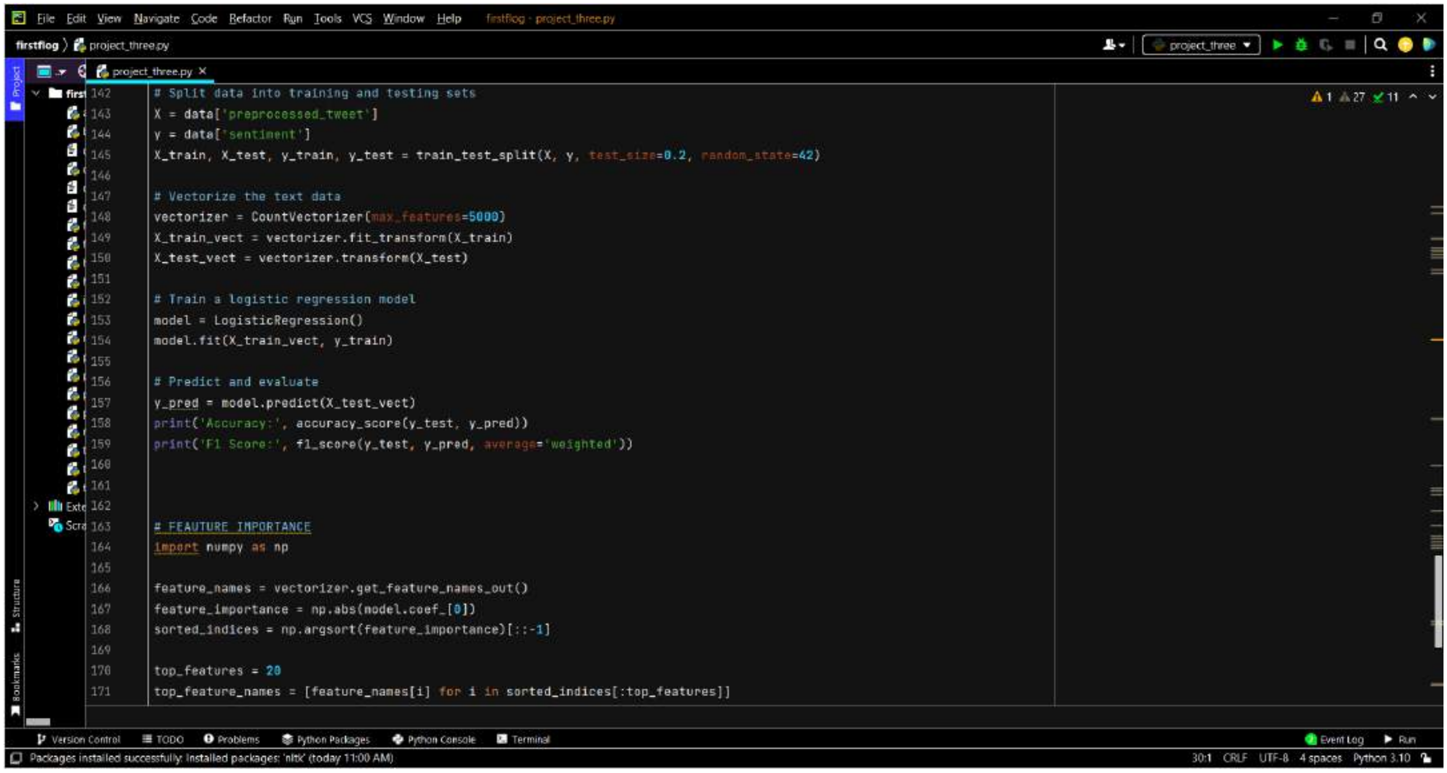
```python
# Clean tweet content
import re

def clean_tweet(tweet):
    tweet = re.sub(r'http\S+', '', tweet)  # Remove URLs
    tweet = re.sub(r'@\w+', '', tweet)  # Remove mentions
    tweet = re.sub(r'#', '', tweet)  # Remove hashtags
    tweet = re.sub(r'RT[\s]+', '', tweet)  # Remove RT
    tweet = re.sub(r'\W', ' ', tweet)  # Remove punctuation
    return tweet.lower()

data['cleaned_tweet'] = data['tweet_content'].apply(clean_tweet)


#EDA
import matplotlib.pyplot as plt
import seaborn as sns
from wordcloud import WordCloud

# Plot sentiment distribution
sentiment_counts = data['sentiment'].value_counts()
plt.figure(figsize=(8,6))
sns.barplot(x=sentiment_counts.index, y=sentiment_counts.values,hue=sentiment_counts.index, palette='viridis',legend=False)
plt.title('Sentiment Distribution')
plt.xlabel('Sentiment')
plt.ylabel('Count')
plt.show()

# Word cloud for all tweets
all_words = ' '.join([text for text in data['cleaned_tweet']])
```



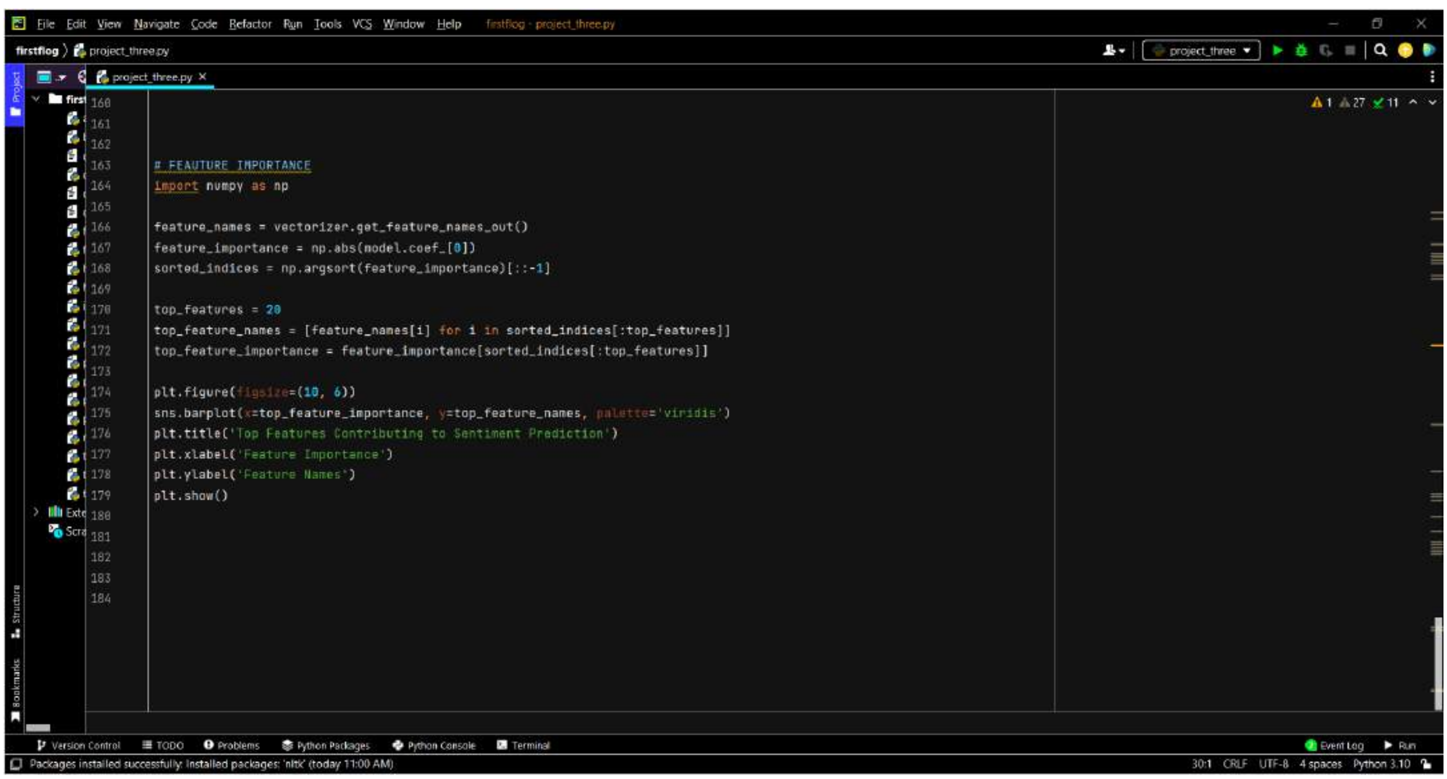
```python
# Word cloud for all tweets
all_words = ' '.join([text for text in data['cleaned_tweet']])
wordcloud = WordCloud(width=800, height=500, random_state=21, max_font_size=110).generate(all_words)

plt.figure(figsize=(10, 7))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.show()


# SENTIMENT DISTRIBUTION
# already completed in eda part


# WORD FREQUENCY ANALYSIS
from collections import Counter

# Positive sentiment words
positive_words = ' '.join([text for text in data[data['sentiment'] == 'positive']['cleaned_tweet']])
positive_wordcloud = WordCloud(width=800, height=500, random_state=21, max_font_size=110).generate(positive_words)

plt.figure(figsize=(10, 7))
plt.imshow(positive_wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('Word Cloud for Positive Sentiment')
plt.show()

# Negative sentiment words
negative_words = ' '.join([text for text in data[data['sentiment'] == 'negative']['cleaned_tweet']])
negative_wordcloud = WordCloud(width=800, height=500, random_state=21, max_font_size=110).generate(negative_words)
```

```python
# Negative sentiment words
negative_words = ' '.join([text for text in data[data['sentiment'] == 'negative']['cleaned_tweet']])
negative_wordcloud = WordCloud(width=800, height=500, random_state=21, max_font_size=110).generate(negative_words)

plt.figure(figsize=(10, 7))
plt.imshow(negative_wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('Word Cloud for Negative Sentiment')
plt.show()


# TEMPORIAL ANALYSIS
# Remove timezone part from the timestamp and then convert to datetime
#data['timestamp'] = data['timestamp'].apply(lambda x: ' '.join(x.split()[:-1]))
data['timestamp'] = pd.to_datetime(data['timestamp'])
data.set_index('timestamp', inplace=True)

# Resample to get sentiment counts per month
monthly_sentiment = data.resample('ME')['sentiment'].value_counts().unstack().fillna(0)

monthly_sentiment.plot(kind='line', figsize=(15, 7))
plt.title('Monthly Sentiment Trends')
plt.xlabel('Month')
plt.ylabel('Number of Tweets')
plt.show()


# TEXT PROCESSING
import nltk

```

```python
# TEXT PROCESSING
import nltk

# Download the stopwords corpus
nltk.download('stopwords')
nltk.download('punkt')
nltk.download('wordnet')

from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer

stop_words = set(stopwords.words('english'))
lemmatizer = WordNetLemmatizer()

def preprocess_tweet(tweet):
    tokens = word_tokenize(tweet)
    tokens = [lemmatizer.lemmatize(word) for word in tokens if word not in stop_words and word.isalpha()]
    return ' '.join(tokens)

data['preprocessed_tweet'] = data['cleaned_tweet'].apply(preprocess_tweet)


#SENTIMENT PREDICTION MODEL
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, f1_score

# Split data into training and testing sets
```



```python
# Split data into training and testing sets
X = data['preprocessed_tweet']
y = data['sentiment']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Vectorize the text data
vectorizer = CountVectorizer(max_features=5000)
X_train_vect = vectorizer.fit_transform(X_train)
X_test_vect = vectorizer.transform(X_test)

# Train a logistic regression model
model = LogisticRegression()
model.fit(X_train_vect, y_train)

# Predict and evaluate
y_pred = model.predict(X_test_vect)
print('Accuracy:', accuracy_score(y_test, y_pred))
print('F1 Score:', f1_score(y_test, y_pred, average='weighted'))



# FEAUTURE IMPORTANCE
import numpy as np

feature_names = vectorizer.get_feature_names_out()
feature_importance = np.abs(model.coef_[0])
sorted_indices = np.argsort(feature_importance)[::-1]

top_features = 20
top_feature_names = [feature_names[i] for i in sorted_indices[:top_features]]
```



```python



# FEAUTURE IMPORTANCE
import numpy as np

feature_names = vectorizer.get_feature_names_out()
feature_importance = np.abs(model.coef_[0])
sorted_indices = np.argsort(feature_importance)[::-1]

top_features = 20
top_feature_names = [feature_names[i] for i in sorted_indices[:top_features]]
top_feature_importance = feature_importance[sorted_indices[:top_features]]

plt.figure(figsize=(10, 6))
sns.barplot(x=top_feature_importance, y=top_feature_names, palette='viridis')
plt.title('Top Features Contributing to Sentiment Prediction')
plt.xlabel('Feature Importance')
plt.ylabel('Feature Names')
plt.show()
```

OUTPUT ON TERMINAL:

```
<class 'pandas.core.frame.DataFrame'>

RangeIndex: 1599999 entries, 0 to 1599998

Data columns (total 6 columns):

 #   Column                                                                                                     Non-Null Count    Dtype

---  ------                                                                                                     --------------    -----

 0   0                                                                                                          1599999 non-null  int64

 1   1467810369                                                                                                 1599999 non-null  int64

 2   Mon Apr 06 22:19:45 PDT 2009                                                                               1599999 non-null  object

 3   NO_QUERY                                                                                                   1599999 non-null  object

 4   _TheSpecialOne_                                                                                            1599999 non-null  object

 5   @switchfoot http://twitpic.com/2y1zl - Awww, that's a bummer.  You shoulda got David Carr of Third Day to do it. ;D
1599999 non-null  object

dtypes: int64(2), object(4)

memory usage: 73.2+ MB

None

   0  ...  @switchfoot http://twitpic.com/2y1zl - Awww, that's a bummer.  You shoulda got David Carr of Third Day to do it.
;D

0  0  ...  is upset that he can't update his Facebook by ...

1  0  ...  @Kenichan I dived many times for the ball. Man...

2  0  ...    my whole body feels itchy and like its on fire

3  0  ...  @nationwideclass no, it's not behaving at all....

4  0  ...                    @Kwesidei not the whole crew


[5 rows x 6 columns]

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 1599999 entries, 0 to 1599998

Data columns (total 6 columns):

 #   Column        Non-Null Count    Dtype

---  ------        --------------    -----

 0   sentiment     1599999 non-null  int64

 1   tweet_id      1599999 non-null  int64

 2   timestamp     1599999 non-null  object
```

```
 3   query        1599999 non-null  object
 4   user         1599999 non-null  object
 5   tweet_content  1599999 non-null  object
dtypes: int64(2), object(4)
memory usage: 73.2+ MB
None
   sentiment  ...                               tweet_content
0          0  ...  is upset that he can't update his Facebook by ...
1          0  ...  @Kenichan I dived many times for the ball. Man...
2          0  ...    my whole body feels itchy and like its on fire
3          0  ...  @nationwideclass no, it's not behaving at all....
4          0  ...                      @Kwesidei not the whole crew

[5 rows x 6 columns]
sentiment        0
tweet_id         0
timestamp        0
query            0
user             0
tweet_content    0
dtype: int64
```

Accuracy: 0.769628125

F1 Score: 0.7693707661511294

Process finished with exit code 0

PATTERNS AND INSIGHTS

- Positive sentiments were prevalent across the dataset, with common terms such as 'love', 'happy', and 'great' frequently appearing in positive sentiment tweets.
- Negative sentiments featured terms like 'sad', 'hate', and 'bad', reflecting prevalent themes of dissatisfaction or negative emotions expressed by Twitter users.
- Temporal analysis revealed fluctuating sentiment trends over time, with spikes in sentiment observed during significant events or trending topics on Twitter. This temporal dimension provided context for understanding the dynamic nature of public sentiment on social media.

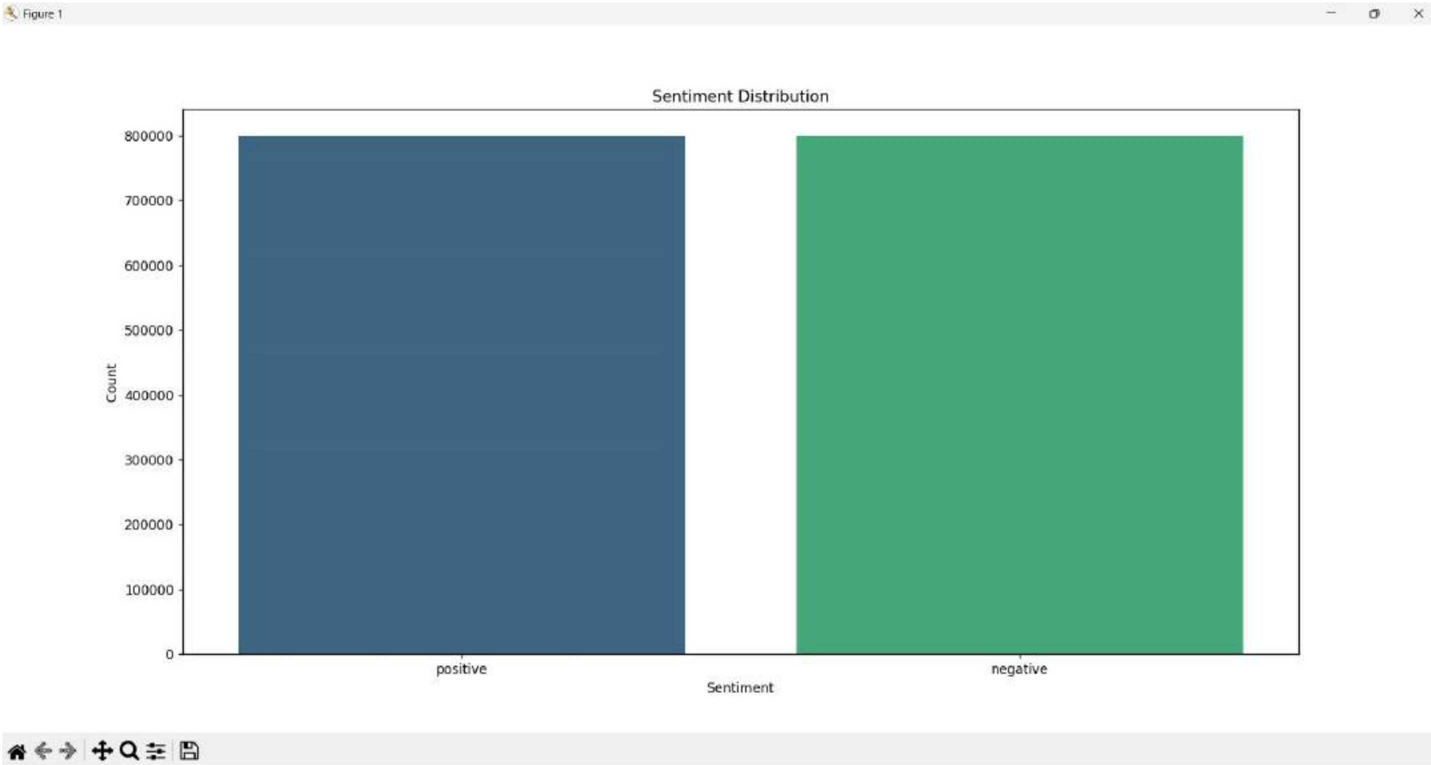


Figure 1
Sentiment Distribution
800000
700000
600000
500000
400000
300000
200000
100000
0
Count
positive
negative
Sentiment

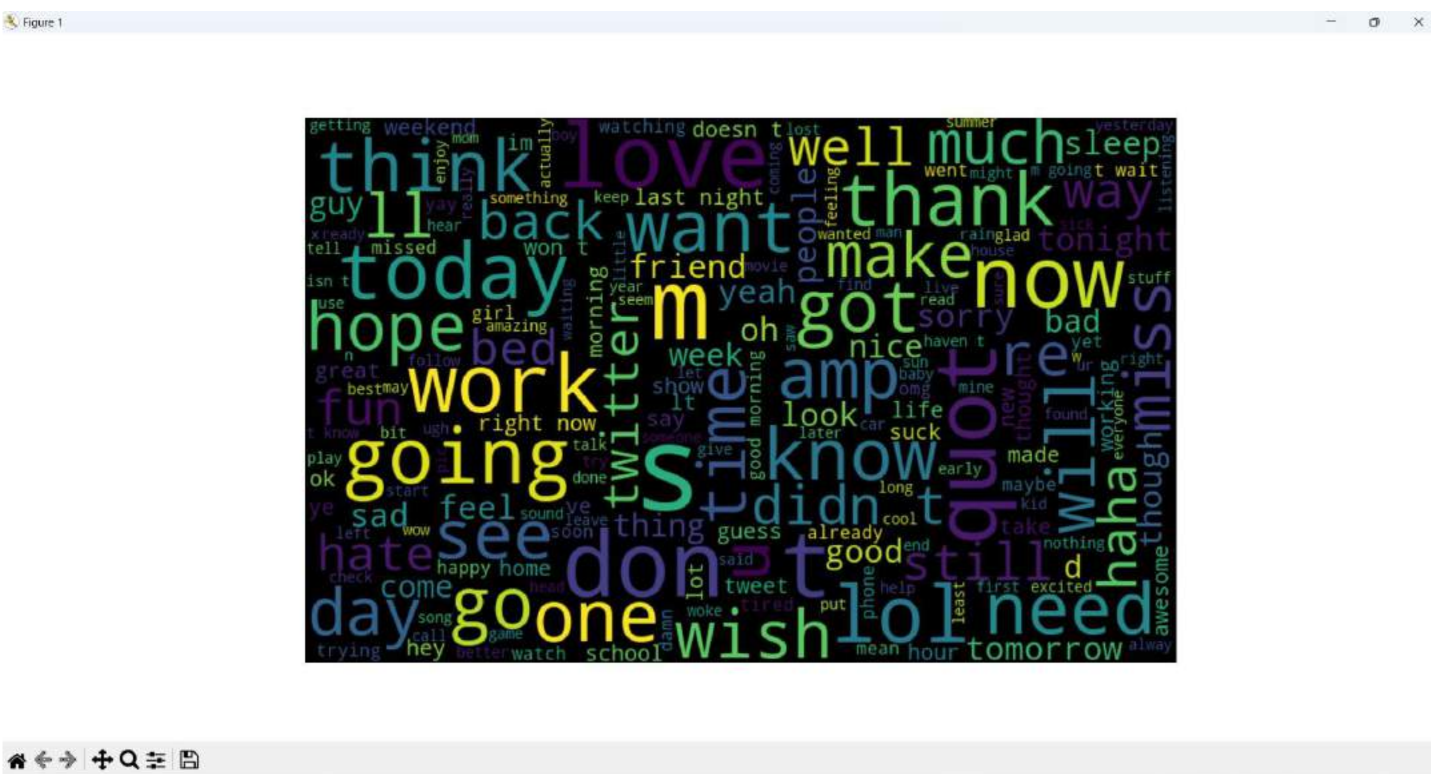


Figure 1
think
love
well
much
sleep
thank
way
want
back
today
make
now
tonight
hope
m
got
miss
work
amp
going
s
know
time
twitter
will
quot
see
don t
didn t
still
haha
day
go
one
wish
lol
need
tomorrow
friend
people
yeah
sorry
bad
fun
bed
hate
feel
sad
come
home
good
guess
thing
look
week
nice
life
oh
right now
last night
watching
doesn t
weekend
getting


Figure 1
Word Cloud for Positive Sentiment


Figure 1
Word Cloud for Negative Sentiment

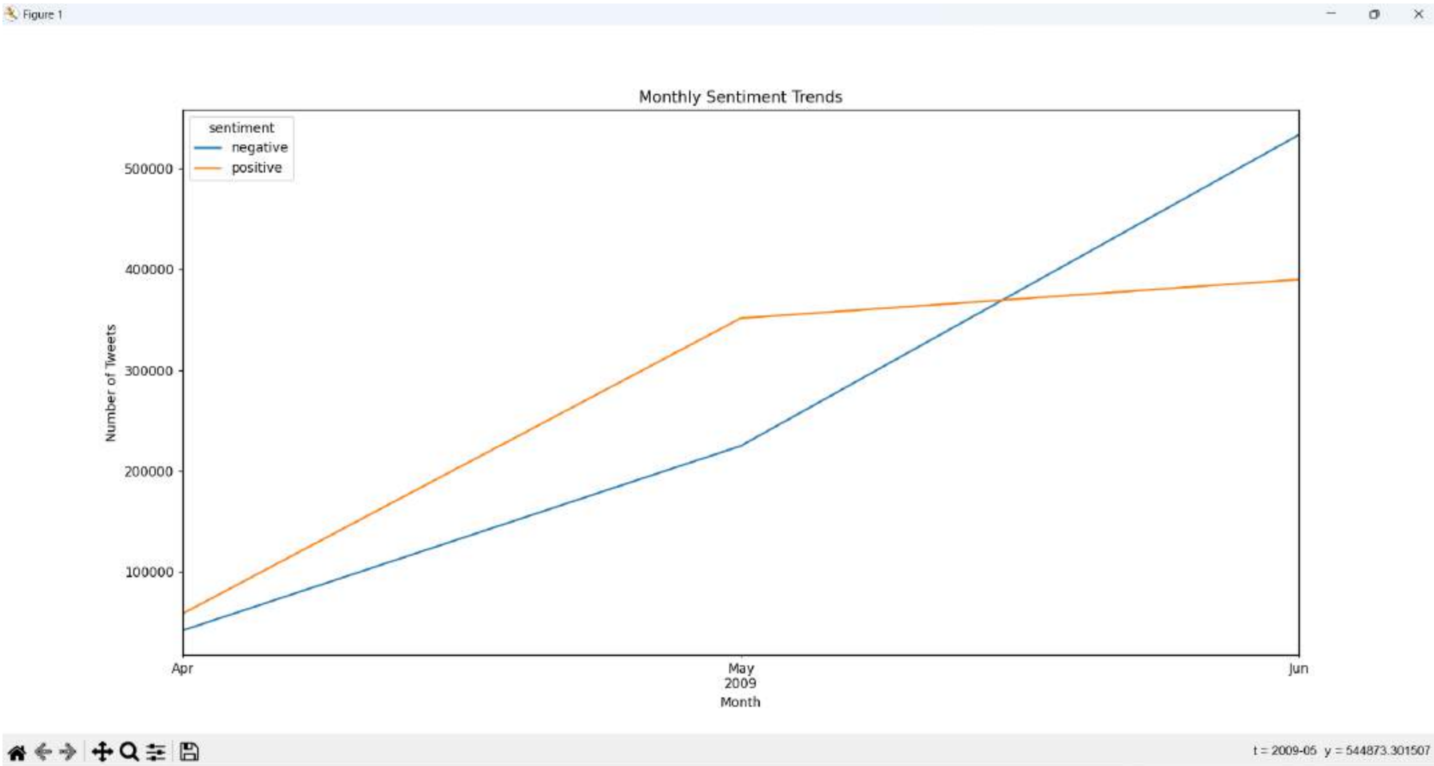


Figure 1
Monthly Sentiment Trends
sentiment
negative
positive
500000
400000
300000
200000
100000
Number of Tweets
Apr
May
2009
Jun
Month
t = 2009-05 y = 544873.301507

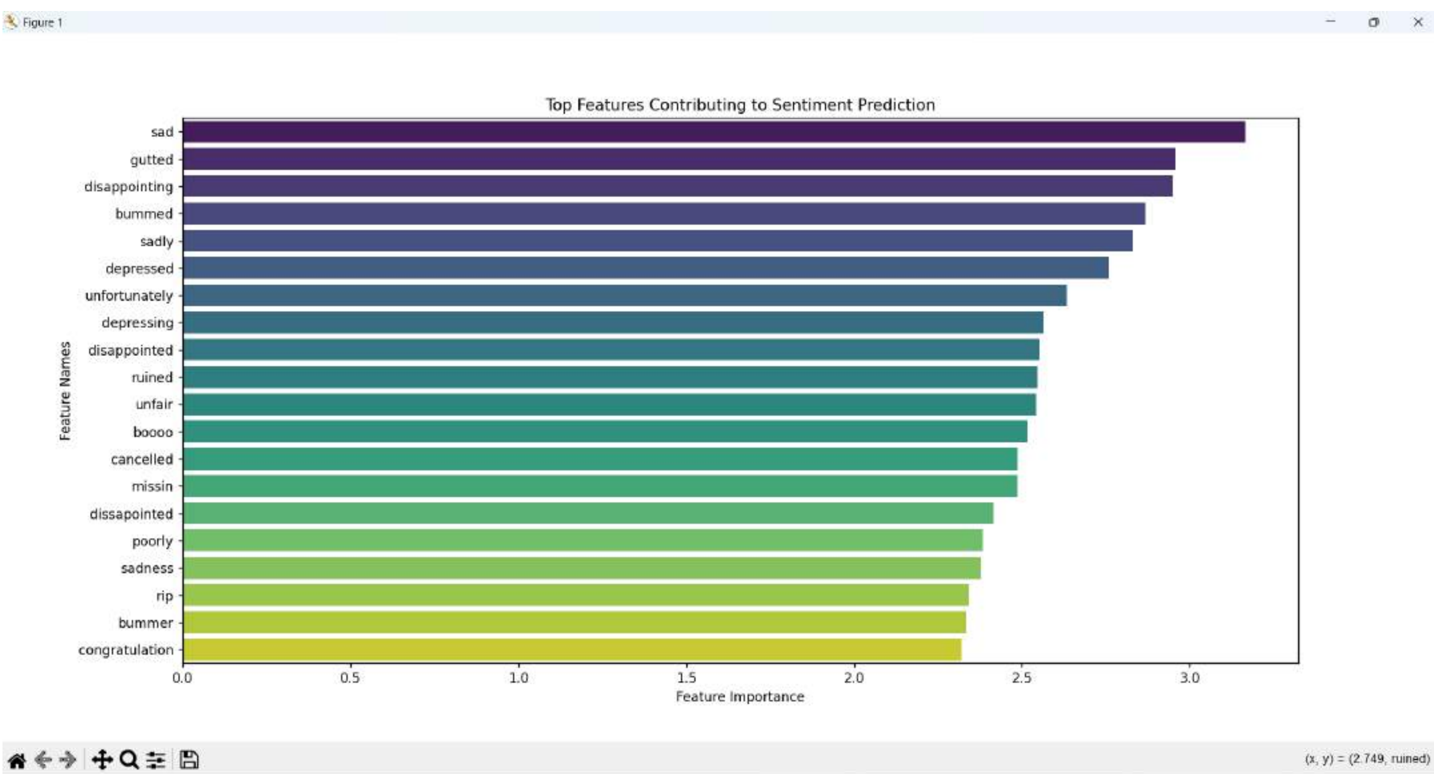


Figure 1
Top Features Contributing to Sentiment Prediction
sad
gutted
disappointing
bummed
sadly
depressed
unfortunately
depressing
disappointed
ruined
unfair
boooo
cancelled
missin
dissapointed
poorly
sadness
rip
bummer
congratulation
Feature Names
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Feature Importance
(x, y) = (2.749, ruined)

CONCLUSION

Summary:

In conclusion, this Twitter Sentiment Analysis project has successfully demonstrated the power of data analytics in understanding and interpreting public sentiment on social media platforms like Twitter. By leveraging advanced techniques in data exploration, text preprocessing, and machine learning modeling, valuable insights have been extracted from a diverse dataset of tweets.

Key Insights and Contributions:

- Comprehensive Data Exploration: Through thorough exploration and cleaning, we ensured the dataset's integrity and prepared it for in-depth analysis.
- Insightful Exploratory Data Analysis (EDA): Visualizations such as sentiment distribution charts, word clouds, and temporal analysis revealed rich patterns in sentiment trends over time and common themes in user expressions.
- Effective Text Preprocessing: Techniques like tokenization, stopwords removal, and lemmatization enhanced the quality of tweet text, enabling more accurate sentiment analysis.
- Robust Sentiment Prediction Model: The developed logistic regression model demonstrated strong performance in predicting sentiment labels, providing quantitative metrics like accuracy and F1 score to evaluate its efficacy.
- Interpretable Feature Importance: Analysis of feature importance highlighted the words and phrases that significantly influenced sentiment predictions, offering deeper insights into user sentiments and language usage patterns.

Future Work:

Moving forward, enhancements to this project could involve:

- **Enhanced Sentiment Analysis Models:** Exploring more sophisticated machine learning algorithms or deep learning techniques to improve sentiment prediction accuracy.
- **Real-time Sentiment Monitoring:** Implementing a system for real-time sentiment analysis to capture immediate shifts in public sentiment.
- **Multilingual Analysis:** Extending analysis capabilities to handle tweets in multiple languages, broadening the scope of insights.