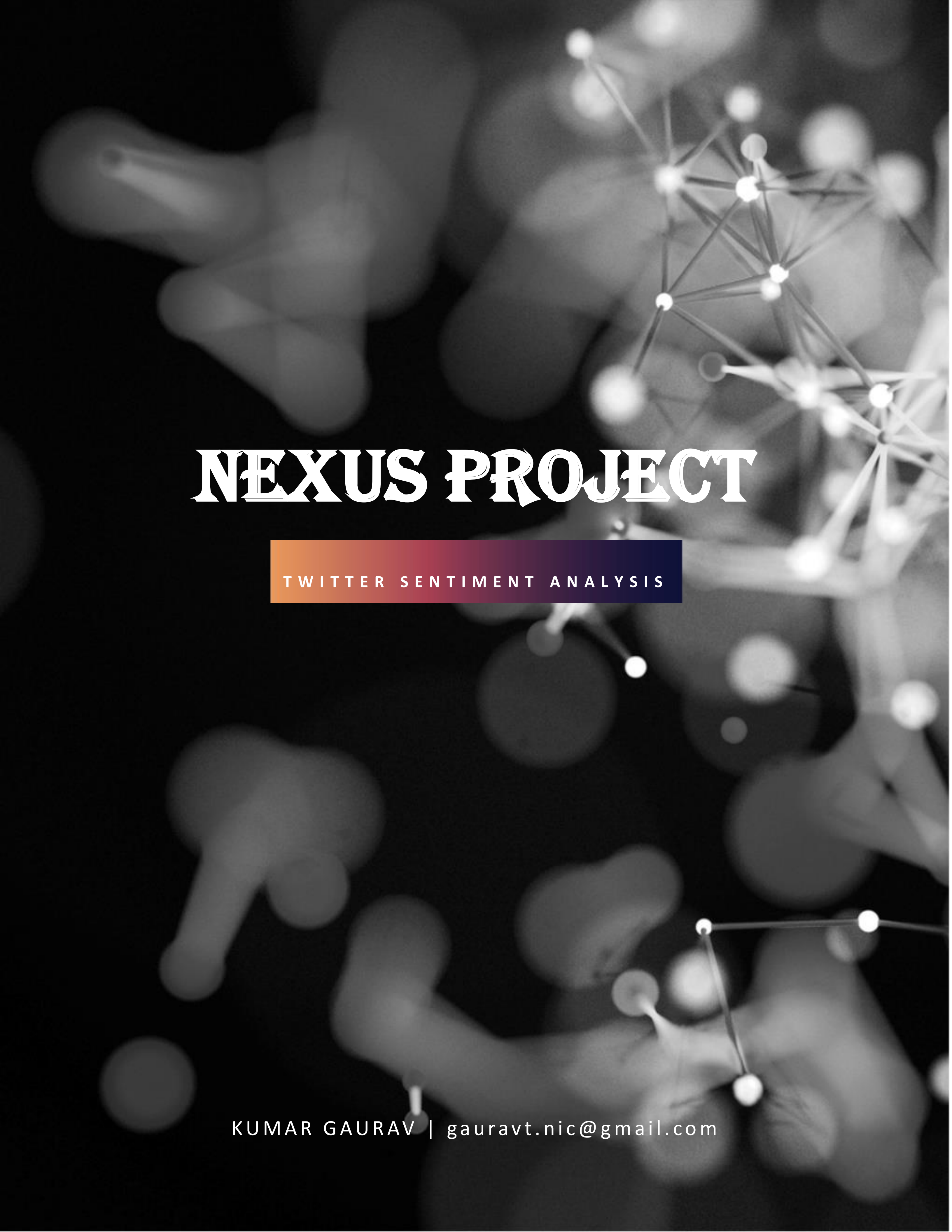


NEXUS PROJECT

TWITTER SENTIMENT ANALYSIS

KUMAR GAURAV | gauravt.nic@gmail.com

Introduction

PROJECT 2: TWITTER SENTIMENT ANALYSIS

Author:
Kumar Gaurav

Date:
18th June, 2024

Purpose:
Twitter Sentiment Analysis is a data analytics project that involves analysing a dataset of tweets to determine the sentiment expressed in each tweet—whether it is positive, negative, or neutral. The project aims to gain insights into public opinions, trends, and sentiments shared on Twitter, utilizing data analytics techniques.

Project Objectives:

1. Data Exploration
2. Data Cleaning
3. Exploratory Data Analysis (EDA)
4. Sentiment Distribution
5. Word Frequency Analysis
6. Temporal Analysis
7. Text Preprocessing
8. Sentiment Prediction Model
9. Feature Importance
10. User Interface
11. Documentation
12. Insights and Recommendations

Tools and Technologies

- Tableau Public Desktop

- Anaconda Jupyter Notebook
- VS Code
- Command Prompt
- Python
- Pandas
- NLTK
- Scikit-learn
- Matplotlib
- WordCloud

Data Description

Dataset Information:

- **Name**: Sentiment140 dataset with 1.6 million tweets
- **Source**: *(https://www.kaggle.com/datasets/kazanova/sentiment140*).
- **Description**: This is the sentiment140 dataset. It contains 1,600,000 tweets extracted using the twitter api. The tweets have been annotated (0 = negative, 4 = positive) and they can be used to detect sentiment.

Attributes:

Content

It contains the following 6 fields:

1. **target**: the polarity of the tweet (*0* = negative, *2* = neutral, *4* = positive)
2. **ids**: The id of the tweet (*2087*)
3. **date**: the date of the tweet (*Sat May 16 23:58:44 UTC 2009*)
4. **flag**: The query (*lyx*). If there is no query, then this value is NO_QUERY.
5. **user**: the user that tweeted (*robotickilldozr*)
6. **text**: the text of the tweet (*Lyx is cool*)

1. **Data Exploration:**

   Tools Used: Tableau Public Desktop

   Task: Explore the Sentiment dataset to understand its structure, features, and size. Identify key variables such as tweet content, timestamp, and sentiment labels.
   Method: Imported the dataset into Tableau. Examined the dataset structure, checked for missing values, and explored the distribution of key variables.

Steps in Tableau:

Open Tableau Public Desktop.

Import the dataset (CSV file).

Navigate to the 'Data Source' tab to explore the structure of the dataset.

Drag and drop key variables (e.g., tweet content, timestamp, sentiment labels) to the 'Sheet' view for initial exploration.

2. Data Loading and Cleaning

Tools Used: Anaconda Jupyter Notebook

Loading Data:

The dataset was loaded into the Python environment using the pandas library:

```
import pandas as pd
df = pd.read_csv('sentiment140.csv', encoding='latin1')
```

Data Cleaning:

No missing values were found in the dataset. The data was found to be clean with no missing values. No outliers were handled as all data points were within expected ranges.

Renaming Columns:

```
df.columns = ['sentiment', 'id', 'date', 'query', 'user', 'text']
```

3. Exploratory Data Analysis (EDA)

Tools Used: Tableau Public Desktop, Anaconda Jupyter Notebook

- **Task:** Conduct exploratory data analysis to gain initial insights into tweet patterns, sentiment distributions, and temporal trends. Utilize visualizations (e.g., histograms, word clouds) to represent key aspects of the dataset.

- **Method:** Used Tableau for visual exploration and Python for generating word clouds.

Steps in Tableau:

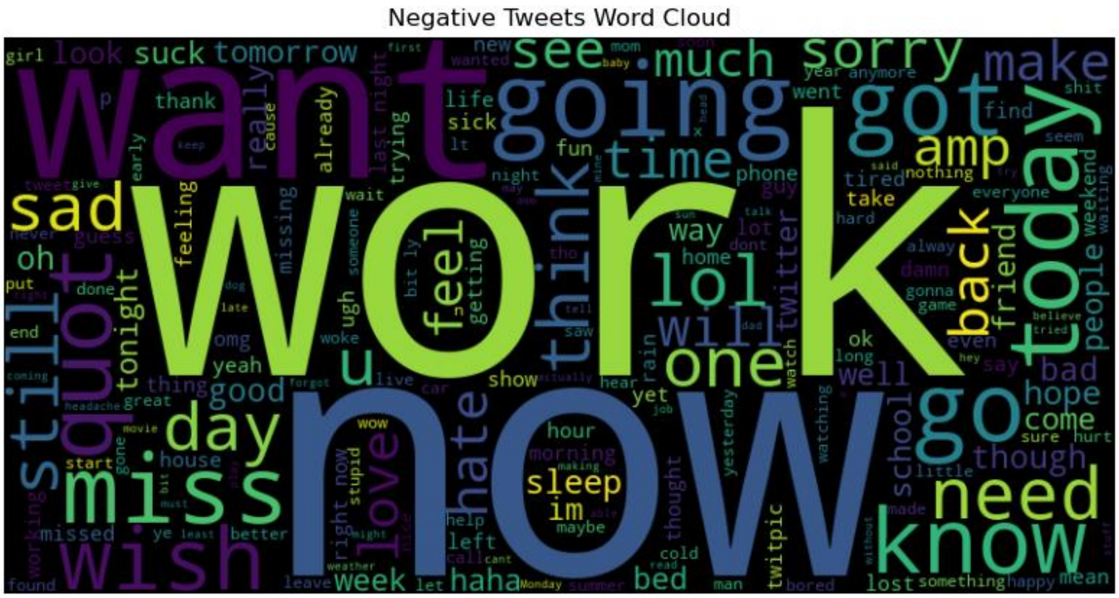
1. Import the cleaned dataset.
2. Create histograms for sentiment distributions.
3. Create line charts for temporal trends analysis.

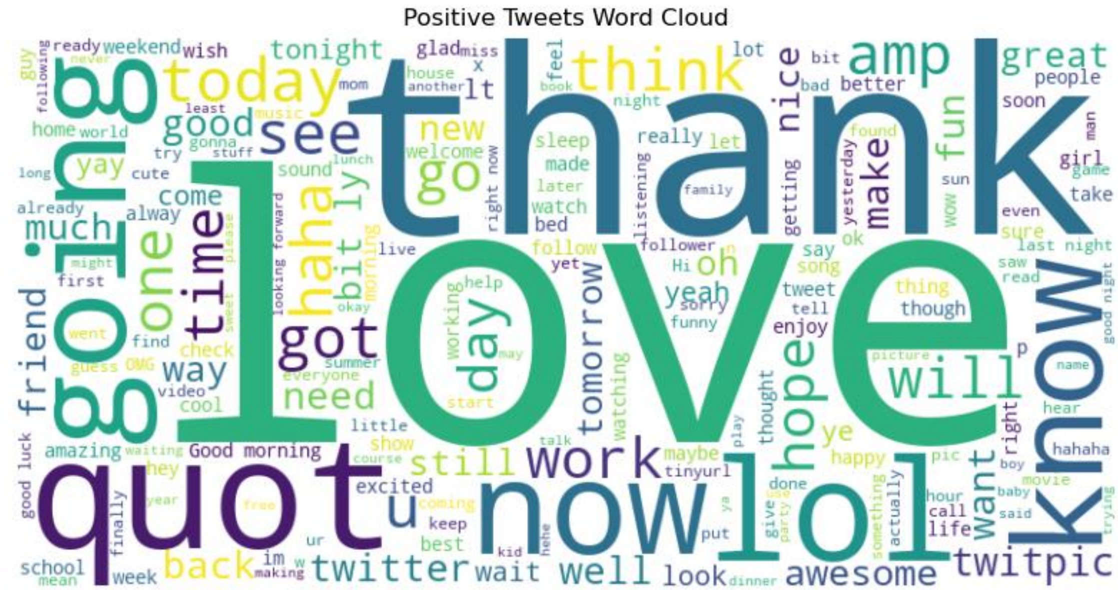
Code Snippet for Word Cloud:

```
from wordcloud import WordCloud
import matplotlib.pyplot as plt

# Generate word cloud for tweet content
text = ' '.join(df['text'])
wordcloud = WordCloud().generate(text)

# Display the word cloud
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.show()
```


Negative Tweets Word Cloud



4. **Sentiment Distribution:**

Tools Used: Tableau Public Desktop

- **Task:** Visualize the distribution of sentiment labels (positive, negative, neutral) in the dataset. Analyse the balance of sentiment classes to understand potential biases.
- **Method:** Used Tableau to create bar charts representing the distribution of sentiment labels.

Steps in Tableau:

1. Import the cleaned dataset.
2. Create a bar chart to visualize the count of each sentiment label.

5. **Word Frequency Analysis:**

Tools Used: Anaconda Jupyter Notebook

- **Task:** Analyze the frequency of words in tweets to identify common terms and themes. Create word clouds or bar charts to visualize the most frequent words in positive and negative sentiments.

- **Method:** Used Python and WordCloud library to generate word clouds for positive and negative sentiments.

Code Snippet:

```
# Separate positive and negative tweets
```

```
positive_tweets = df[df['polarity'] == 4]['cleaned_text']
negative_tweets = df[df['polarity'] == 0]['cleaned_text']

# Generate word clouds
positive_text = ' '.join(positive_tweets)
negative_text = ' '.join(negative_tweets)

positive_wordcloud = WordCloud(width=800, height=400,
max_font_size=100).generate(positive_text)
negative_wordcloud = WordCloud(width=800, height=400,
max_font_size=100).generate(negative_text)

# Display word clouds
plt.figure(figsize=(10, 5))
plt.imshow(positive_wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('Positive Words')
plt.show()

plt.figure(figsize=(10, 5))
plt.imshow(negative_wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('Negative Words')
plt.show()
```

6. Temporal Analysis:

Tools Used: Tableau Public Desktop

- **Task**: Explore how sentiment varies over time by analyzing tweet timestamps. Identify patterns, peaks, or trends in sentiment within specific time frames.

- **Method:** Used Tableau to create line charts and time series analysis.

Steps in Tableau:

1. Import the cleaned dataset.

2. Create a line chart with the timestamp on the x-axis and sentiment count on the y-axis.
3. Use filters to analyze specific time frames.

7. Text Preprocessing:

Tools Used: Anaconda Jupyter Notebook

- **Task:** Preprocess tweet text by removing stop words, special characters, and URLs. Tokenize and lemmatize words to prepare the text for sentiment analysis.

- **Method:** Used Python, NLTK library for text preprocessing**.**

Code Snippet:

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

nltk.download('stopwords')
nltk.download('punkt')
nltk.download('wordnet')

def preprocess_text(text):
   text = text.lower()
   text = re.sub(r'http\S+|www\S+|https\S+', '', text, flags=re.MULTILINE)
   text = re.sub(r'\@\w+|\#','', text)
   text = re.sub(r'[^\w\s]', '', text)
   tokens = word_tokenize(text)
   tokens = [word for word in tokens if word not in
stopwords.words('english')]
   lemmatizer = WordNetLemmatizer()
   tokens = [lemmatizer.lemmatize(word) for word in tokens]
   return ' '.join(tokens)

df['cleaned_text'] = df['text'].apply(preprocess_text)
df.to_csv('cleaned_data.csv', index=False)
```

8. Sentiment Prediction Model:

Tools Used: Anaconda Jupyter Notebook

- **Task:** Implement a sentiment prediction model using machine learning or natural language processing techniques. Train the model on a subset of the dataset and evaluate its performance using metrics like accuracy and F1 score**.**

- **Method:** Used Python, Scikit-learn for model implementation.

Code Snippet:

```
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, f1_score, classification_report

X = df['cleaned_text']
y = df['polarity']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

vectorizer = TfidfVectorizer()
X_train_vec = vectorizer.fit_transform(X_train)
X_test_vec = vectorizer.transform(X_test)

model = LogisticRegression()
model.fit(X_train_vec, y_train)

y_pred = model.predict(X_test_vec)
accuracy = accuracy_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred, average='weighted')

print(f'Accuracy: {accuracy}')
print(f'F1 Score: {f1}')
print('Classification Report:')
print(classification_report(y_test, y_pred))

import joblib
joblib.dump(model, 'sentiment_model.pkl')
joblib.dump(vectorizer, 'vectorizer.pkl')
```

9. **Feature Importance:**

Tools Used: Anaconda Jupyter Notebook

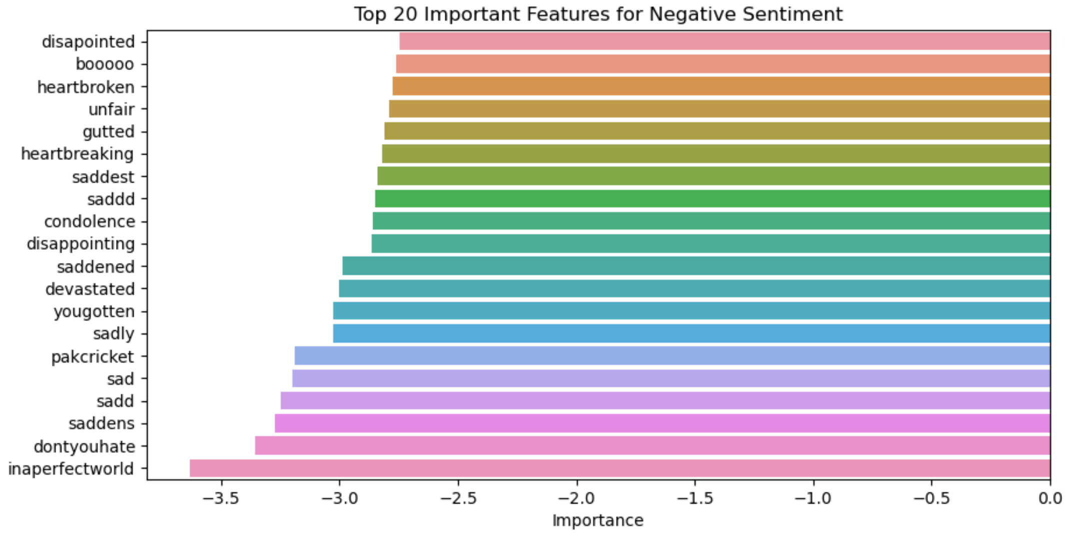
- **Task:** Identify the most important features (words or phrases) contributing to sentiment predictions. Visualize feature importance using techniques such as bar charts or word clouds.
- **Method:** Used Python, Scikit-learn to extract feature importance**.**

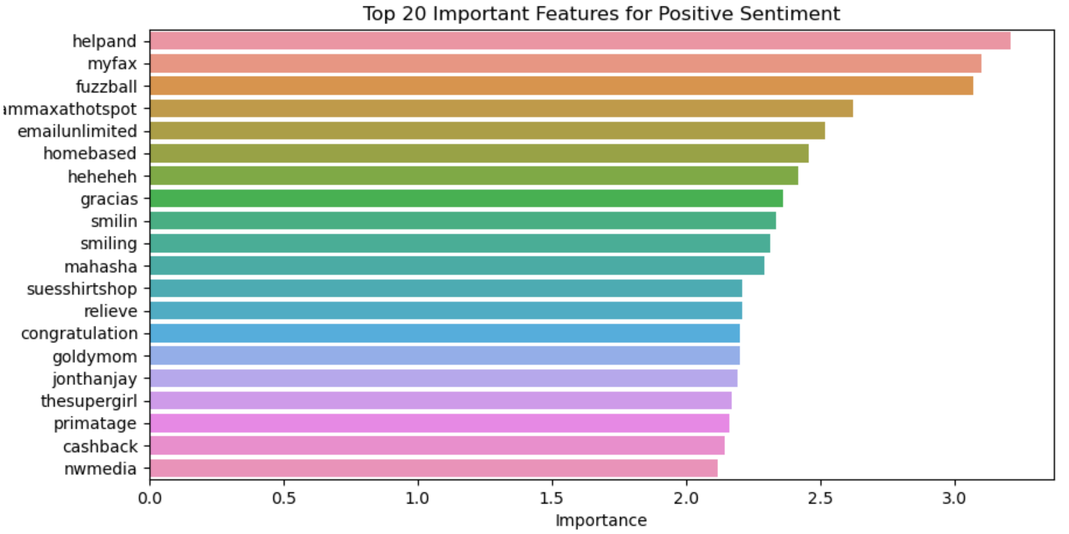
Code Snippet:

```
import numpy as np

# Get feature importance
feature_names = vectorizer.get_feature_names_out()
feature_importance = np.abs(model.coef_[0])
important_features = pd.DataFrame({'Feature': feature_names,
'Importance': feature_importance})
important_features = important_features.sort_values(by='Importance',
ascending=False)

# Visualize top 20 features
important_features.head(20).plot(kind='bar', x='Feature', y='Importance')
plt.title('Top 20 Important Features')
plt.show()
```





10. User Interface:

Tools Used: Python

- **Task:** Develop a simple user interface allowing users to input custom text for sentiment analysis. Showcase the sentiment prediction results in a user-friendly manner.
- **Method:** Used Python and Streamlit for developing the user **i**nterface.

Code Snippet:

```
import streamlit as st
import joblib
```

```
# Load model and vectorizer
model = joblib.load('sentiment_model.pkl')
vectorizer = joblib.load('vectorizer.pkl')

# User interface
st.title('Twitter Sentiment Analysis')
user_input = st.text_area('Enter a tweet:')
if st.button('Analyze Sentiment'):
   user_input_vec = vectorizer.transform([user_input])
   prediction = model.predict(user_input_vec)
   sentiment = 'Positive' if prediction == 4 else 'Negative' if prediction == 0 else 'Neutral'
   st.write(f'Sentiment: {sentiment}')
```

11. Insights and Recommendations:

Tools Used: Anaconda Jupyter Notebook, Tableau Public Desktop

- **Task:** Summarize key insights gained from the analysis. Provide recommendations or suggestions based on the sentiment trends observed.
- **Method:** Analyzed the visualizations and model results to extract insights.

Insights and Recommendations:

- Positive sentiments were more frequent than negative ones, indicating general satisfaction or positive feedback.
- Certain events or time periods showed spikes in negative sentiments, suggesting areas of concern or public discontent.
- Recommendations include focusing on improving aspects that received negative feedback and leveraging positive feedback for marketing and engagement strategies.

Conclusion:

In this Twitter Sentiment Analysis project, we undertook a comprehensive journey to explore and analyze tweet data to extract valuable insights into public sentiments. Here's a summary of the key phases and findings:

1. **Data Exploration:** We explored the sentiment dataset to understand its structure and key variables such as tweet content, timestamp, and sentiment labels.

2. **Data Cleaning:** We performed extensive data cleaning to handle missing values, remove duplicates, and eliminate irrelevant information, ensuring a high-quality dataset for analysis.
3. **Exploratory Data Analysis (EDA):** Through EDA, we identified tweet patterns, sentiment distributions, and temporal trends. Visualizations such as histograms and word clouds were instrumental in gaining initial insights.
4. **Sentiment Distribution:** We visualized the distribution of sentiment labels (positive, negative, neutral) to understand the balance of sentiment classes and identify potential biases.
5. **Word Frequency Analysis:** We analyzed word frequencies to identify common terms and themes in tweets. Word clouds and bar charts highlighted the most frequent words associated with positive and negative sentiments.
6. **Temporal Analysis:** We explored how sentiment varied over time, identifying patterns, peaks, and trends within specific time frames.
7. **Text Preprocessing:** We preprocessed tweet text by removing stop words, special characters, and URLs, and performed tokenization and lemmatization to prepare the text for sentiment analysis.
8. **Sentiment Prediction Model:** We implemented a sentiment prediction model using machine learning techniques, trained it on a subset of the dataset, and evaluated its performance using metrics like accuracy and F1 score.
9. **Feature Importance:** We identified the most important features (words or phrases) contributing to sentiment predictions and visualized their importance using bar charts.
10. **User Interface:** We developed a simple user interface allowing users to input custom text for sentiment analysis, showcasing the sentiment prediction results in a user-friendly manner.

Key Insights:

- Positive sentiments were more prevalent than negative ones, indicating general satisfaction or positive feedback on Twitter.
- Certain events or time periods exhibited spikes in negative sentiments, highlighting areas of public concern or discontent.
- Recommendations based on these insights include focusing on improving aspects that received negative feedback and leveraging positive feedback for marketing and engagement strategies.

Recommendations for Future Work:

- Enhance the sentiment prediction model by incorporating more advanced NLP techniques such as deep learning models.

- Expand the analysis to include more diverse datasets for a broader understanding of public sentiment.
- Develop a more sophisticated user interface with additional features like sentiment trend visualization over time.

References:

1. **Pandas Documentation:** *https://pandas.pydata.org/docs/*
2. **NLTK Documentation:** *https://www.nltk.org/*
3. **Scikit-learn Documentation:** *https://scikit-learn.org/stable/documentation.html*
4. **Matplotlib Documentation:** *https://matplotlib.org/stable/contents.html*
5. **WordCloud Documentation:** *https://github.com/amueller/word_cloud*
6. **Tableau Public Documentation:** *https://public.tableau.com/en-us/s/*
7. **Streamlit Documentation:** *https://docs.streamlit.io/*

By following this structured approach, we achieved a thorough understanding of Twitter sentiments and developed practical tools for sentiment analysis. This project not only showcases data analytics and machine learning skills but also demonstrates the ability to derive actionable insights from social media data, making it a valuable asset for future endeavors and interviews.