Project Instruction Document: Part 1

Project Title: Sentiment Analysis & Visualization on Sentiment140 Dataset

Objective:

To understand the nature of sentiments in the Sentiment140 dataset through comprehensive data analysis and visualization.

Link to the Dataset: http://help.sentiment140.com/for-students/

Tips and Recommendations:

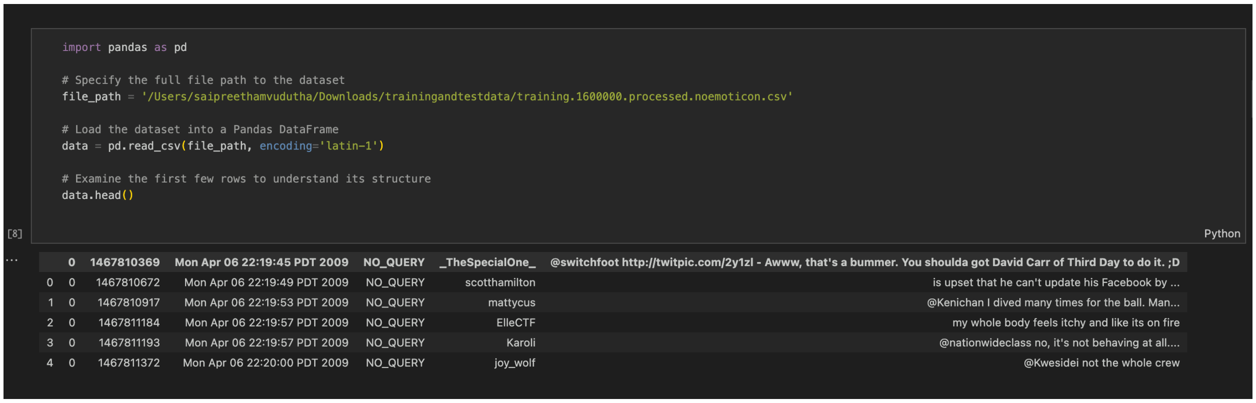
- Regularly save your work and the dataset after significant changes.
- Consider using Jupyter Notebook or Jupyter Lab for this project as they allow for interactive analysis and visualization in the same environment.
- Be careful while removing stopwords, as sometimes they can carry sentiment value. Adjust the stopwords list accordingly.
- While tokenizing and cleaning, consider using stemming or lemmatization to reduce words to their base form, thus reducing the overall number of unique words.
- Regularly discuss your findings with team members to gain different perspectives on the data.

Detailed Instructions:

1. Data Analysis and Exploration using Python:

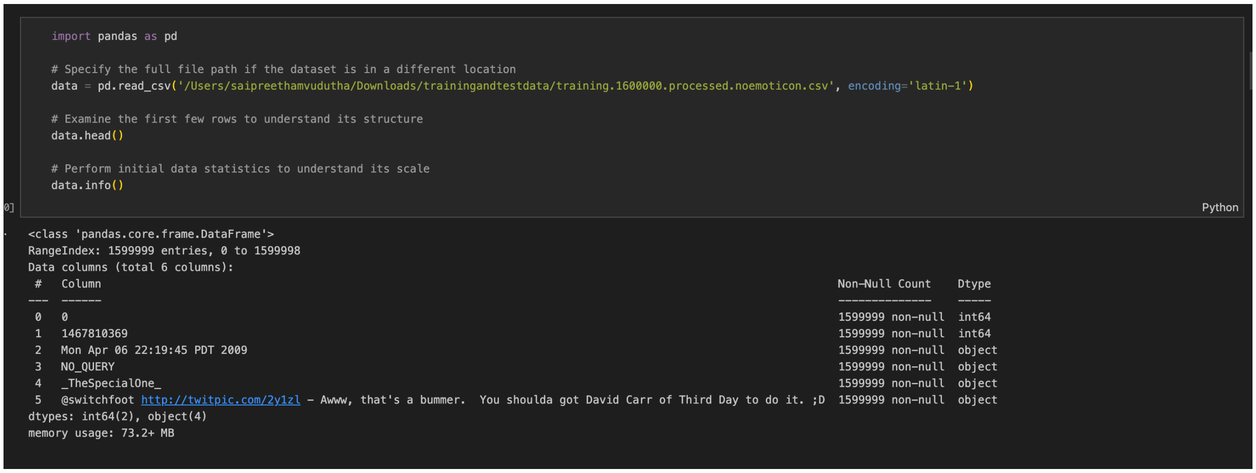
1.1. Loading the Dataset:

- Load the Sentiment140 dataset into a Pandas DataFrame.
- Examine the first few rows to understand its structure.
- Perform initial data statistics to understand its scale, e.g., `dataframe.info()`, `dataframe.describe()`.



```python
import pandas as pd

# Specify the full file path to the dataset
file_path = '/Users/saipreethamvudutha/Downloads/trainingandtestdata/training.1600000.processed.noemoticon.csv'

# Load the dataset into a Pandas DataFrame
data = pd.read_csv(file_path, encoding='latin-1')

# Examine the first few rows to understand its structure
data.head()
```

	0	1467810369	Mon Apr 06 22:19:45 PDT 2009	NO_QUERY	_TheSpecialOne_	@switchfoot http://twitpic.com/2y1zl - Awww, that's a bummer. You shoulda got David Carr of Third Day to do it. ;D
0	0	1467810672	Mon Apr 06 22:19:49 PDT 2009	NO_QUERY	scotthamilton	is upset that he can't update his Facebook by ...
1	0	1467810917	Mon Apr 06 22:19:53 PDT 2009	NO_QUERY	mattycus	@Kenichan I dived many times for the ball. Man...
2	0	1467811184	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	ElleCTF	my whole body feels itchy and like its on fire
3	0	1467811193	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	Karoli	@nationwideclass no, it's not behaving at all....
4	0	1467811372	Mon Apr 06 22:20:00 PDT 2009	NO_QUERY	joy_wolf	@Kwesidei not the whole crew

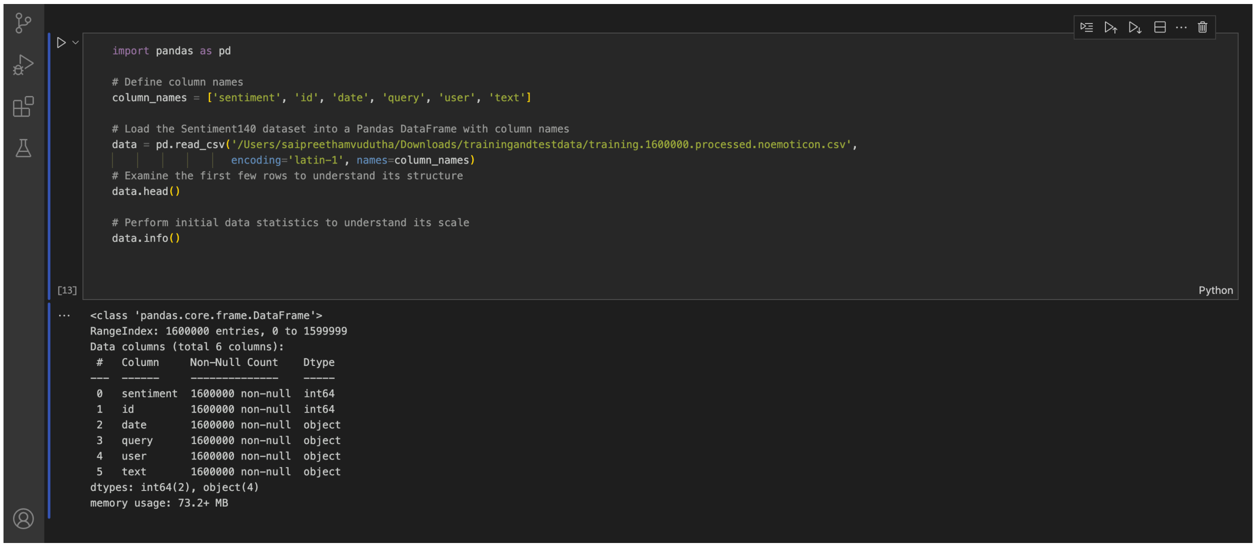


```python
import pandas as pd

# Specify the full file path if the dataset is in a different location
data = pd.read_csv('/Users/saipreethamvudutha/Downloads/trainingandtestdata/training.1600000.processed.noemoticon.csv', encoding='latin-1')

# Examine the first few rows to understand its structure
data.head()

# Perform initial data statistics to understand its scale
data.info()
```

```
<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
```

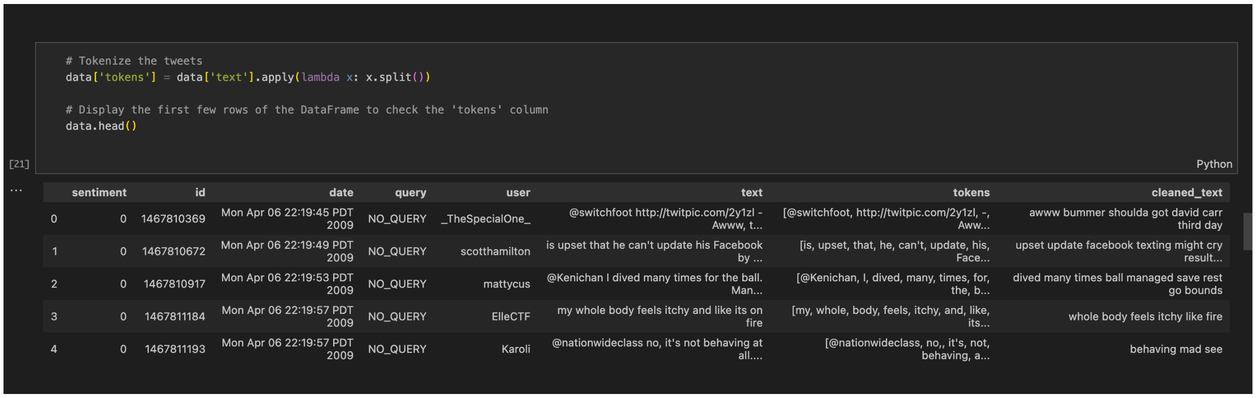
1.2. Data Cleaning:

- Identify and handle missing values.
- Check for duplicate tweets and decide on their removal based on the context.
- Tokenize the tweets: split the tweets into individual words for further analysis.
- Clean the tweets:
  - Remove URLs, mentions, numbers, and non-alphanumeric characters.
- Remove stop words using the NLTK library.
- Convert all words to lowercase to maintain consistency.



```
import pandas as pd

# Define column names
column_names = ['sentiment', 'id', 'date', 'query', 'user', 'text']

# Load the Sentiment140 dataset into a Pandas DataFrame with column names
data = pd.read_csv('/Users/saipreethamvudutha/Downloads/trainingandtestdata/training.1600000.processed.noemoticon.csv',
                   encoding='latin-1', names=column_names)
# Examine the first few rows to understand its structure
data.head()

# Perform initial data statistics to understand its scale
data.info()
```

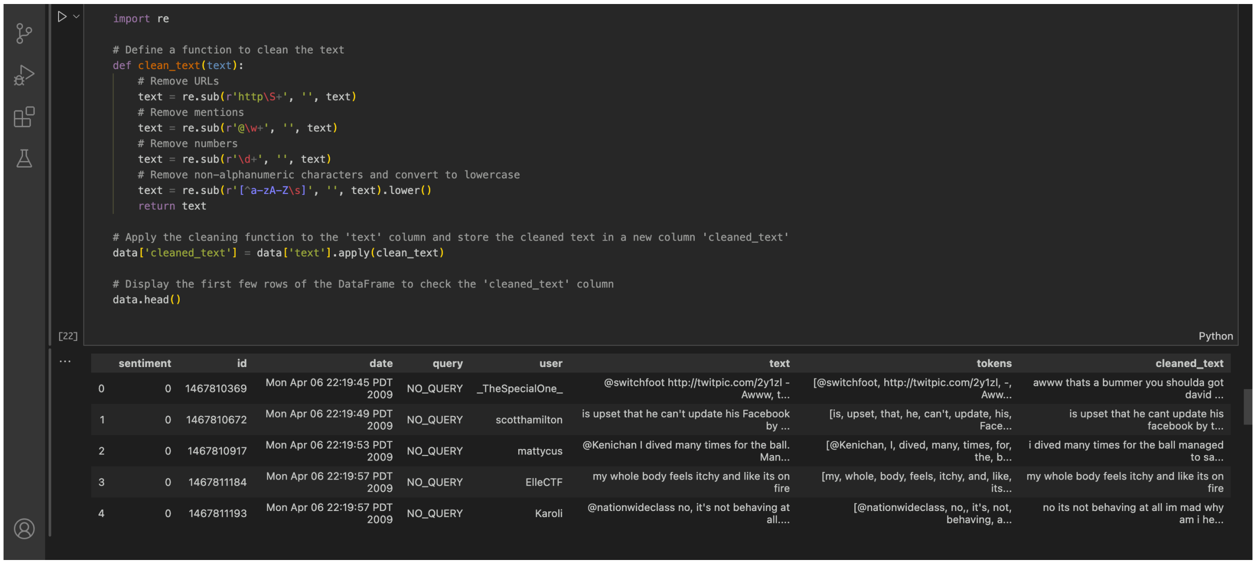
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1600000 entries, 0 to 1599999
Data columns (total 6 columns):
 #   Column     Non-Null Count    Dtype 
---  ------     --------------    ----- 
 0   sentiment  1600000 non-null  int64 
 1   id         1600000 non-null  int64 
 2   date       1600000 non-null  object
 3   query      1600000 non-null  object
 4   user       1600000 non-null  object
 5   text       1600000 non-null  object
dtypes: int64(2), object(4)
memory usage: 73.2+ MB
```



```
# Tokenize the tweets
data['tokens'] = data['text'].apply(lambda x: x.split())

# Display the first few rows of the DataFrame to check the 'tokens' column
data.head()
```

	sentiment	id	date	query	user	text	tokens	cleaned_text
0	0	1467810369	Mon Apr 06 22:19:45 PDT 2009	NO_QUERY	_TheSpecialOne_	@switchfoot http://twitpic.com/2y1zl - Awww, t...	[@switchfoot, http://twitpic.com/2y1zl, -, Aww...	awww bummer shoulda got david carr third day
1	0	1467810672	Mon Apr 06 22:19:49 PDT 2009	NO_QUERY	scotthamilton	is upset that he can't update his Facebook by ...	[is, upset, that, he, can't, update, his, Face...	upset update facebook texting might cry result...
2	0	1467810917	Mon Apr 06 22:19:53 PDT 2009	NO_QUERY	mattycus	@Kenichan I dived many times for the ball. Man...	[@Kenichan, I, dived, many, times, for, the, b...	dived many times ball managed save rest go bounds
3	0	1467811184	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	ElleCTF	my whole body feels itchy and like its on fire	[my, whole, body, feels, itchy, and, like, its...	whole body feels itchy like fire
4	0	1467811193	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	Karoli	@nationwideclass no, it's not behaving at all....	[@nationwideclass, no,, it's, not, behaving, a...	behaving mad see

Clean the text:

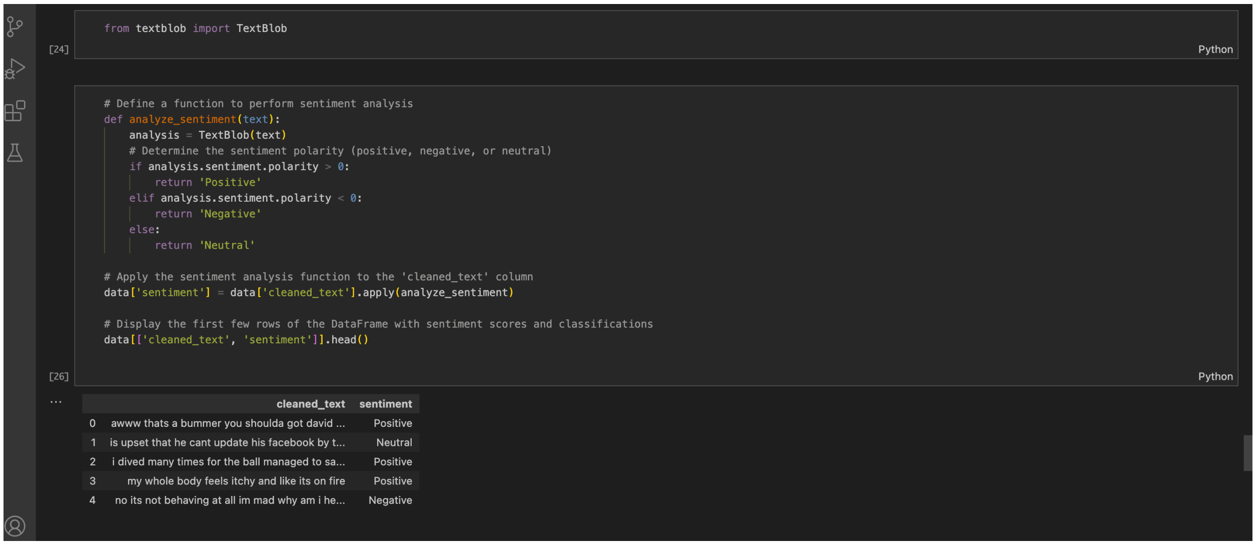


```
import re

# Define a function to clean the text
def clean_text(text):
    # Remove URLs
    text = re.sub(r'http\S+', '', text)
    # Remove mentions
    text = re.sub(r'@\w+', '', text)
    # Remove numbers
    text = re.sub(r'\d+', '', text)
    # Remove non-alphanumeric characters and convert to lowercase
    text = re.sub(r'[^a-zA-Z\s]', '', text).lower()
    return text

# Apply the cleaning function to the 'text' column and store the cleaned text in a new column 'cleaned_text'
data['cleaned_text'] = data['text'].apply(clean_text)

# Display the first few rows of the DataFrame to check the 'cleaned_text' column
data.head()
```

	sentiment	id	date	query	user	text	tokens	cleaned_text
0	0	1467810369	Mon Apr 06 22:19:45 PDT 2009	NO_QUERY	_TheSpecialOne_	@switchfoot http://twitpic.com/2y1zl - Awww, t...	[@switchfoot, http://twitpic.com/2y1zl, -, Aww...	awww thats a bummer you shoulda got david ...
1	0	1467810672	Mon Apr 06 22:19:49 PDT 2009	NO_QUERY	scotthamilton	is upset that he can't update his Facebook by ...	[is, upset, that, he, can't, update, his, Face...	is upset that he cant update his facebook by t...
2	0	1467810917	Mon Apr 06 22:19:53 PDT 2009	NO_QUERY	mattycus	@Kenichan I dived many times for the ball. Man...	[@Kenichan, I, dived, many, times, for, the, b...	i dived many times for the ball managed to sa...
3	0	1467811184	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	ElleCTF	my whole body feels itchy and like its on fire	[my, whole, body, feels, itchy, and, like, its...	my whole body feels itchy and like its on fire
4	0	1467811193	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	Karoli	@nationwideclass no, it's not behaving at all....	[@nationwideclass, no,, it's, not, behaving, a...	no its not behaving at all im mad why am i he...

2. Sentiment Analysis using TextBlob or VADER:

2.1. Analyzing Sentiments:
- For each cleaned tweet, determine its sentiment score using the chosen library.
- Classify tweets as positive, negative, or neutral based on the sentiment score.
- Add the sentiment classification as a new column in the DataFrame.

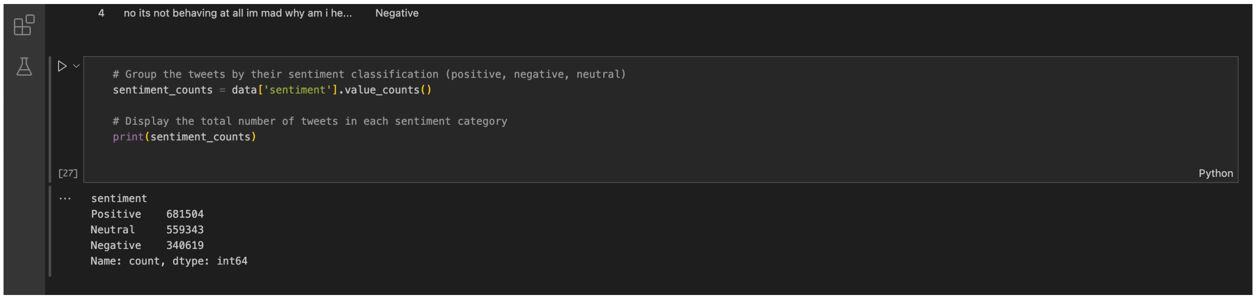


```
from textblob import TextBlob
```
[24] Python

```
# Define a function to perform sentiment analysis
def analyze_sentiment(text):
    analysis = TextBlob(text)
    # Determine the sentiment polarity (positive, negative, or neutral)
    if analysis.sentiment.polarity > 0:
        return 'Positive'
    elif analysis.sentiment.polarity < 0:
        return 'Negative'
    else:
        return 'Neutral'

# Apply the sentiment analysis function to the 'cleaned_text' column
data['sentiment'] = data['cleaned_text'].apply(analyze_sentiment)

# Display the first few rows of the DataFrame with sentiment scores and classifications
data[['cleaned_text', 'sentiment']].head()
```
[26] Python

	cleaned_text	sentiment
0	awww thats a bummer you shoulda got david ...	Positive
1	is upset that he cant update his facebook by t...	Neutral
2	i dived many times for the ball managed to sa...	Positive
3	my whole body feels itchy and like its on fire	Positive
4	no its not behaving at all im mad why am i he...	Negative

2.2. Aggregation of Sentiments:
- Group tweets by their sentiment classification.
- Calculate the total number of tweets in each sentiment category.



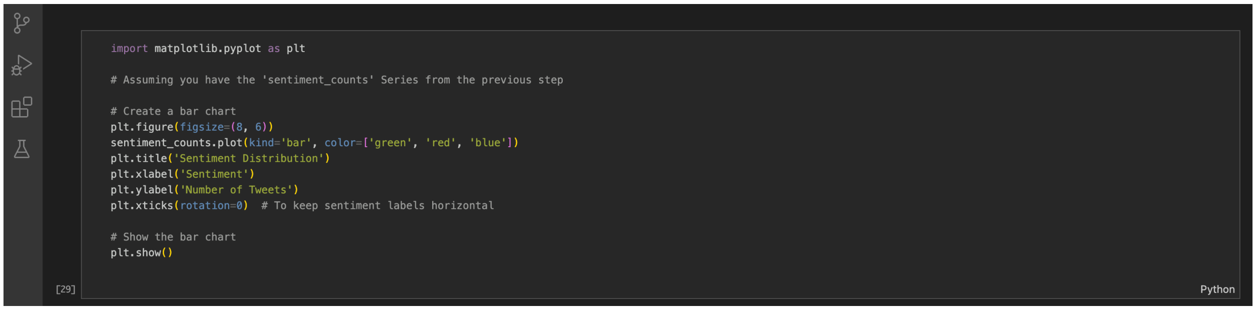
```
4    no its not behaving at all im mad why am i he...    Negative
```

```
# Group the tweets by their sentiment classification (positive, negative, neutral)
sentiment_counts = data['sentiment'].value_counts()

# Display the total number of tweets in each sentiment category
print(sentiment_counts)
```
[27] Python

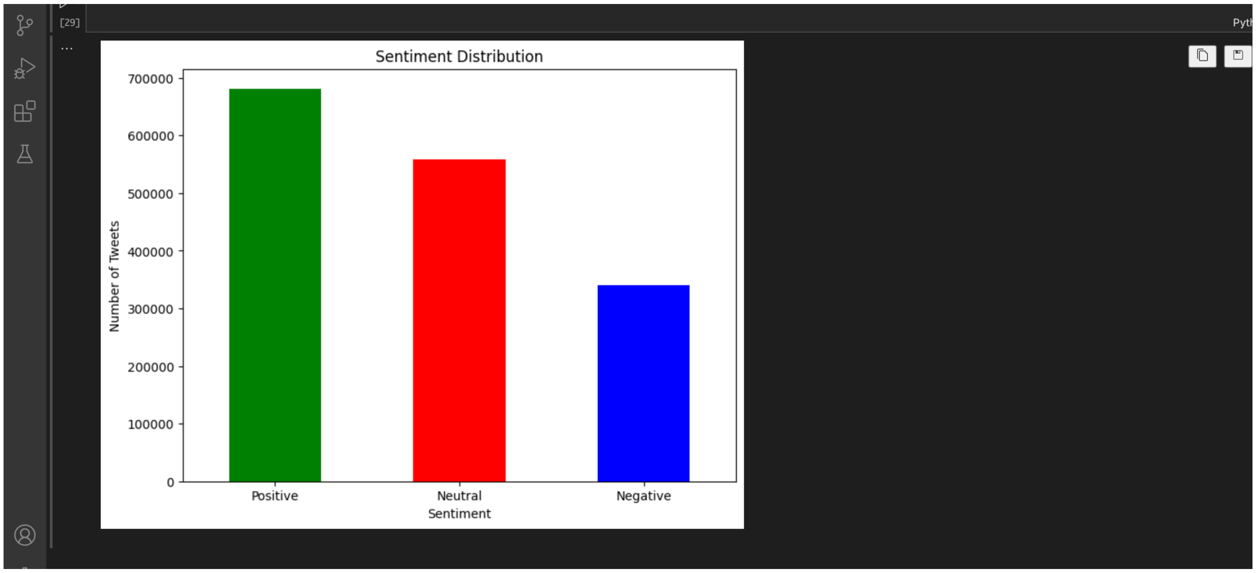
```
sentiment
Positive    681504
Neutral     559343
Negative    340619
Name: count, dtype: int64
```

3. Data Visualization using Matplotlib and Seaborn:

3.1. Distribution of Sentiments:
- Create a bar chart or pie chart to visualize the distribution of tweets among positive, negative, and neutral sentiments.



```python
import matplotlib.pyplot as plt

# Assuming you have the 'sentiment_counts' Series from the previous step

# Create a bar chart
plt.figure(figsize=(8, 6))
sentiment_counts.plot(kind='bar', color=['green', 'red', 'blue'])
plt.title('Sentiment Distribution')
plt.xlabel('Sentiment')
plt.ylabel('Number of Tweets')
plt.xticks(rotation=0)  # To keep sentiment labels horizontal

# Show the bar chart
plt.show()
```



3.2. Word Cloud:
- Generate a word cloud for each sentiment category using the `WordCloud` library. This provides a visual representation of the most frequent words associated with each sentiment.

Expected Outcomes:

1. Cleaned Dataset: The original Sentiment140 dataset but processed and cleaned with additional columns for tokenized and cleaned tweets and their sentiment classifications.
2. Analysis Report: A comprehensive report detailing the findings from the sentiment analysis.

3. Visualization Dashboard: A set of interactive visualizations showcasing sentiment distribution, word clouds, and sentiment trends over time.

Conclusion:

Part 1 of this project will give you a hands-on experience with a real-world dataset and will familiarize them with the steps involved in sentiment analysis. The visualizations will make it easier for you to communicate your findings. This foundational knowledge will set the stage for Part 2, where you will work with real-time data from Twitter.