# **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()`.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

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.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Clean the text:

A screenshot of a computer

Description automatically generated

##### 

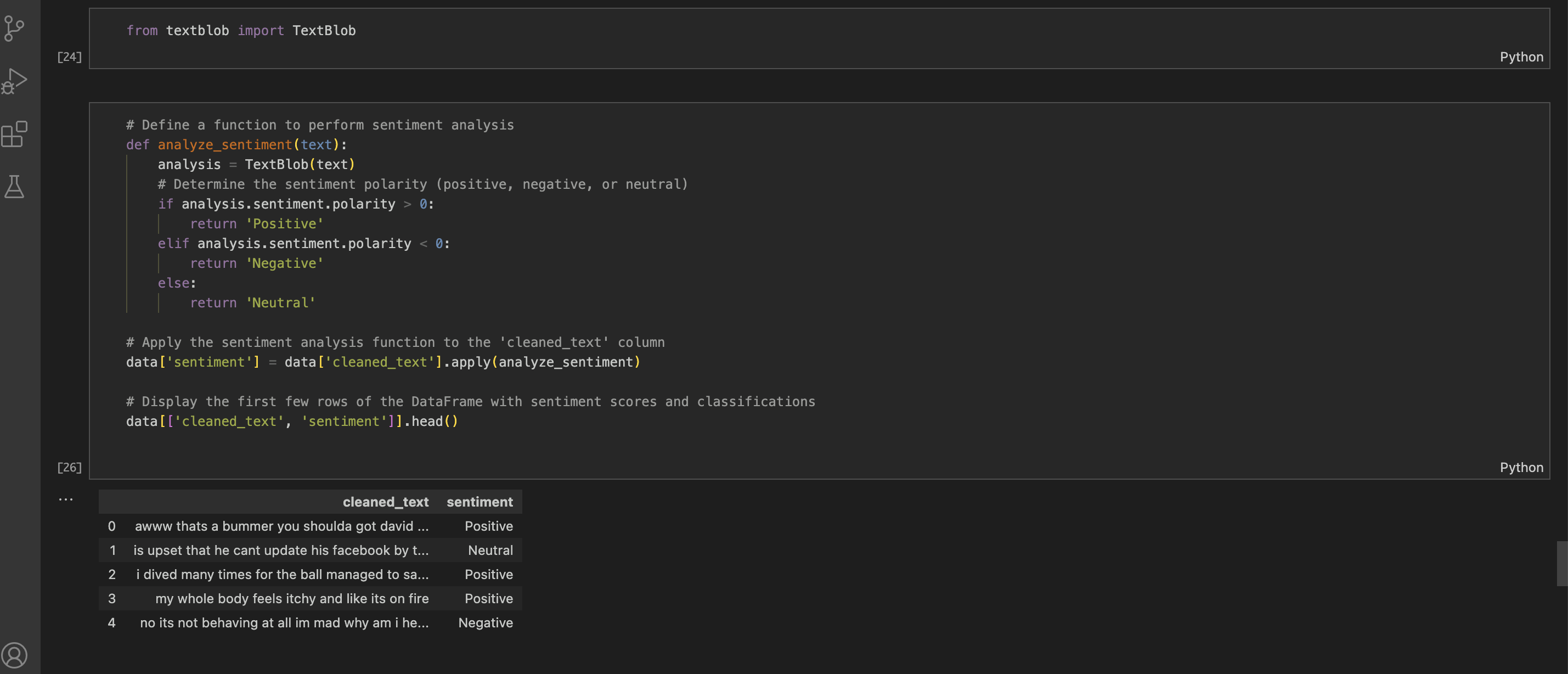
##### **2. Sentiment Analysis using TextBlob or VADER:**

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.



2.2. Aggregation of Sentiments:

- Group tweets by their sentiment classification.

- Calculate the total number of tweets in each sentiment category.

A screenshot of a computer

Description automatically generated

##### **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.

A screenshot of a computer

Description automatically generated

A screenshot of a graph

Description automatically generated

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.