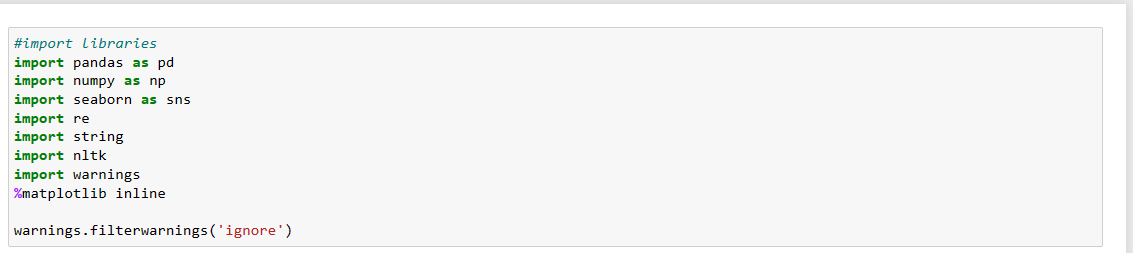
**TWITTER SENTIMENT ANALYSIS**

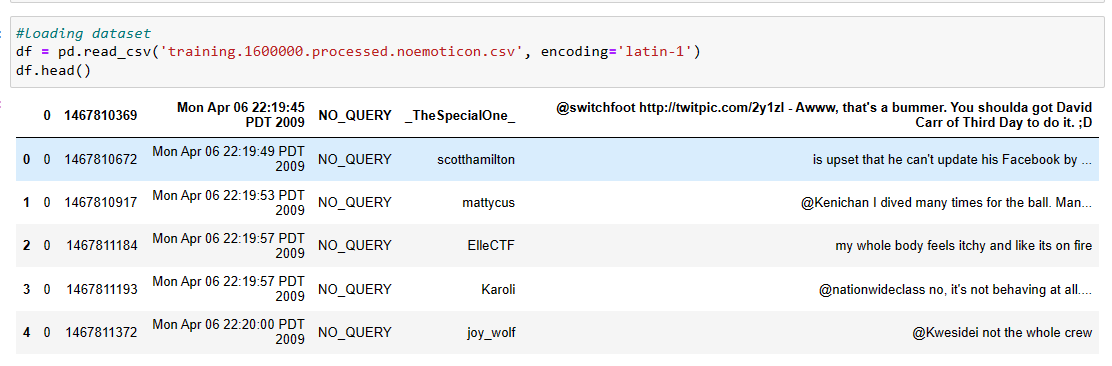
1. Data Loading and Exploration:

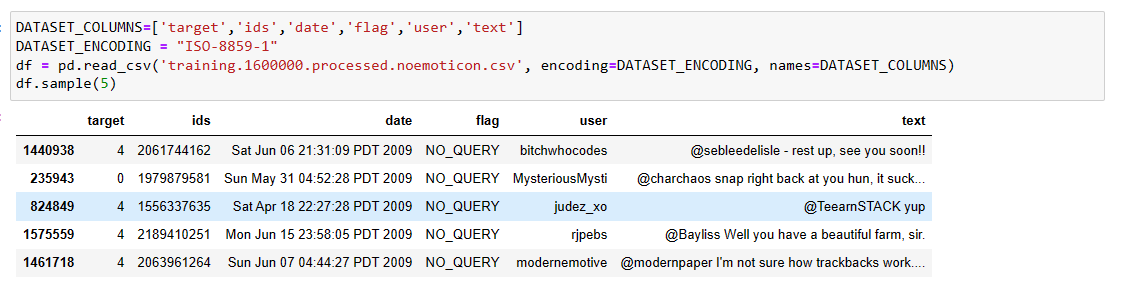
- Libraries Import: The necessary libraries are imported, including Pandas, NumPy, Seaborn, Regular Expressions, and NLTK. The `%matplotlib inline` magic command is used to display plots inline.

- Warnings Configuration: The code sets up a configuration to ignore warnings during execution.

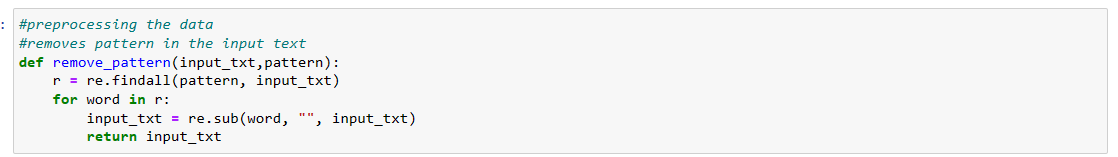


- Dataset Loading: The code loads the Twitter dataset from the file 'Twitter.csv' into a Pandas DataFrame (`df`). Two different methods are used for loading, first without specifying column names and then with specific column names ('target', 'ids', 'date', 'flag', 'user', 'text').

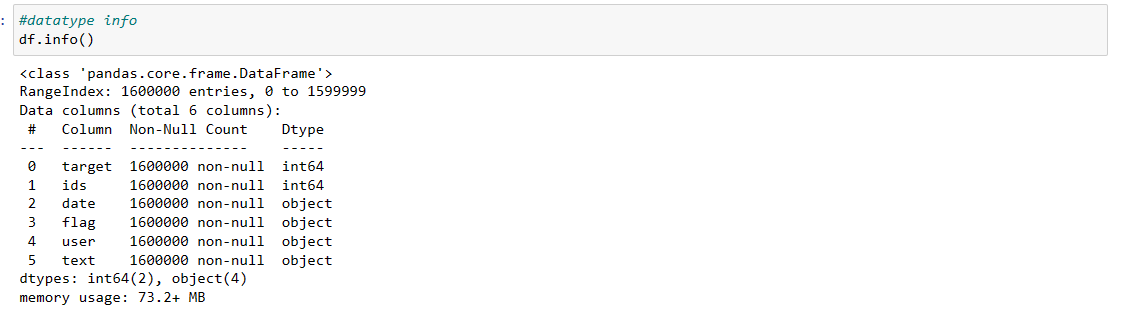




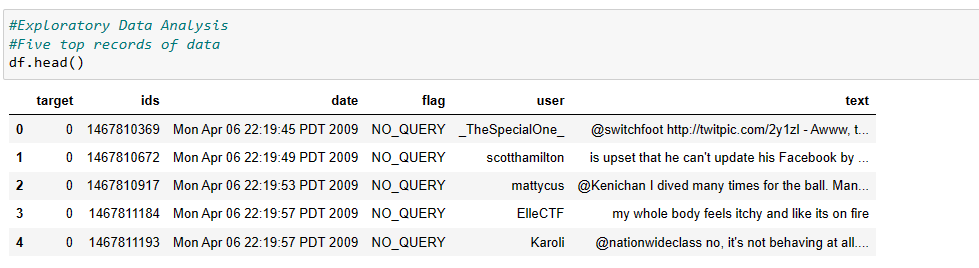
Function for Pattern Removal: The code defines a function `remove\_pattern` that takes an input text and a pattern, and removes instances of that pattern from the text.



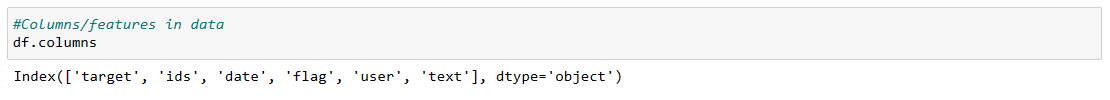
Dataset Information: The code displays information about the data types and non-null counts in each column of the DataFrame.



- Displaying First 5 Records: The first 5 records of the DataFrame are displayed using the `head()` function.



- Column Names: The column names of the DataFrame are displayed using the `columns` attribute.



- Length of Dataset: The length of the dataset (number of rows) is printed.

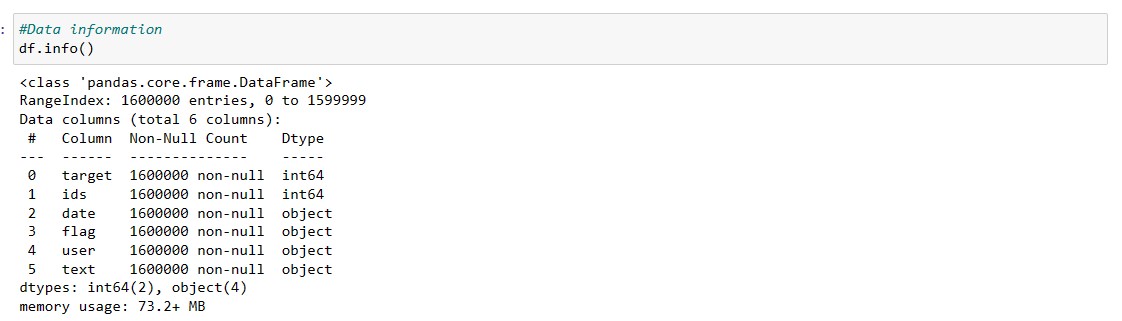


- Shape of Data: The shape of the DataFrame (number of rows and columns) is displayed

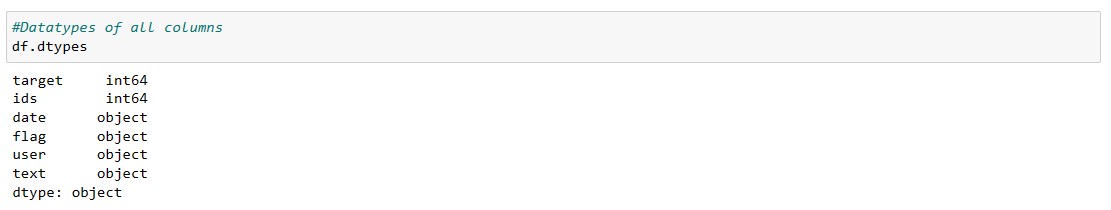


.

- Data Information: Similar to Section 15, this repeats the display of information about data types and non-null counts.



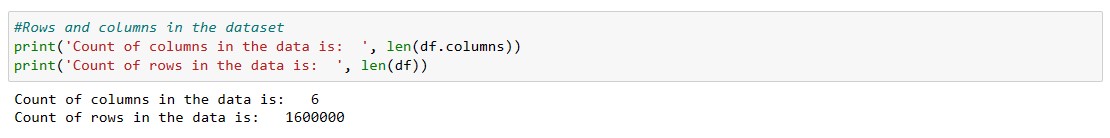
- Data Types of Columns: The data types of all columns are displayed using the `dtypes` attribute.



- Checking for Null Values: The code checks for null values in the DataFrame using the `isnull()` function and sums them up.



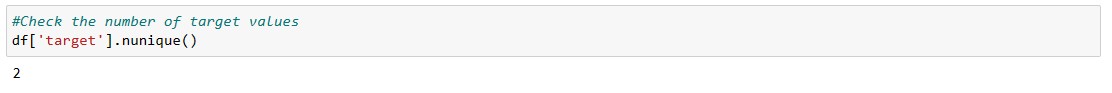
- Rows and Columns Count: The count of columns and rows in the dataset is printed.



- Unique Target Values: The unique values present in the 'target' column (sentiment) are displayed. In this case, it shows `[0, 4]`.



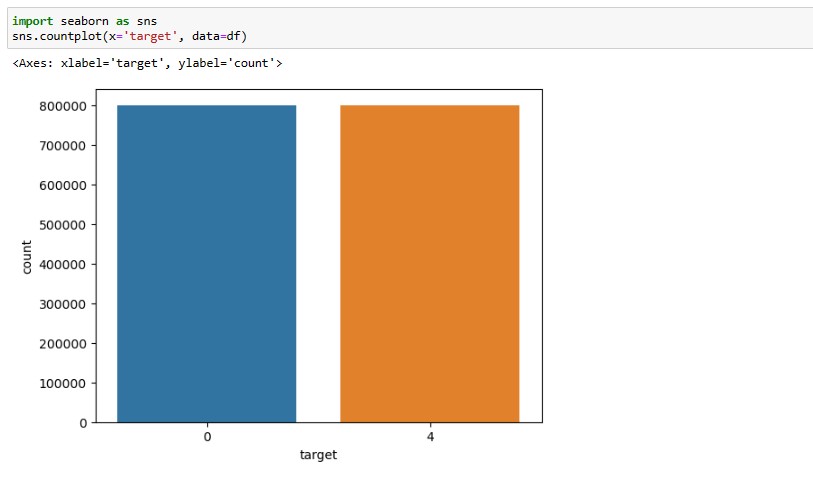
- Number of Unique Target Values: The number of unique values in the 'target' column is printed. In this case, it shows `2`.



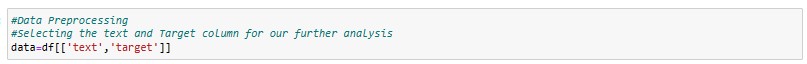
- Data Visualization - Bar Plot :- A bar plot is created to visualize the distribution of data points for each sentiment class ('target').



- Data Visualization - Count Plot:- A count plot using Seaborn is created to visualize the distribution of data points for each sentiment class.



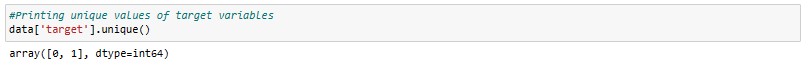
- Data Preprocessing - Selecting Columns:- The code creates a new DataFrame `data` by selecting only the 'text' and 'target' columns from the original DataFrame.



- Data Preprocessing - Replacing Values :- Values in the 'target' column are replaced to make it more interpretable (replacing 4 with 1).



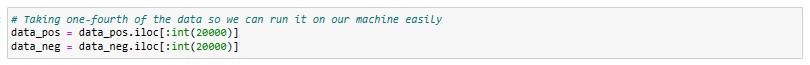
- Unique Values of Target Variable: The unique values present in the 'target' column after replacement are displayed. In this case, it shows `[0, 1]`



- Separating Positive and Negative Tweets: The code creates two DataFrames (`data\_pos` and `data\_neg`) by filtering positive and negative tweets based on the 'target' column.



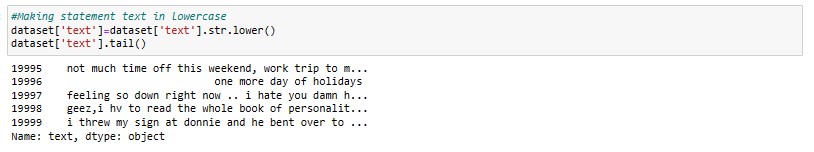
-Data Subset Size Reduction: One-fourth of the data for both positive and negative tweets is selected.



-Combining Positive and Negative Tweets: The positive and negative tweet DataFrames are concatenated into a new DataFrame named `dataset`.

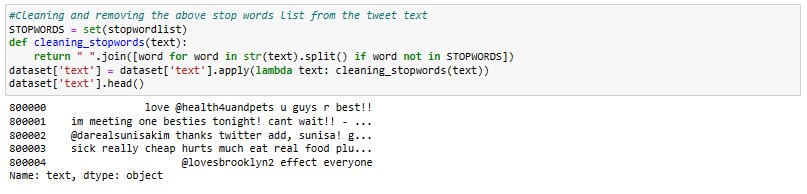


- Text Lowercasing: The text in the 'text' column of the dataset is converted to lowercase.

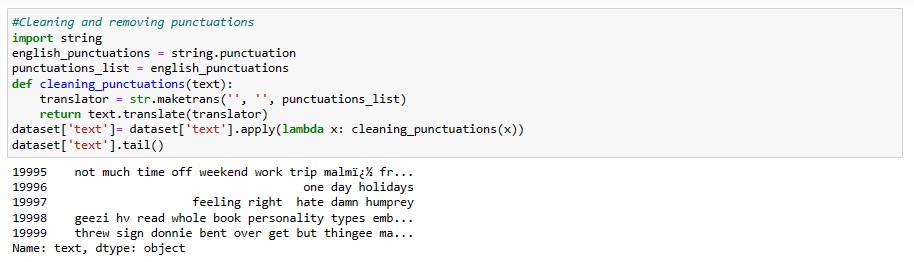


- Cleaning and Removing Stopwords: A list of English stopwords is defined, and stopwords are removed from the tweet text using the `cleaning\_stopwords` function.

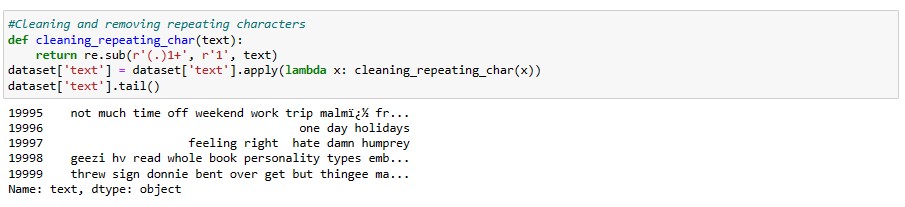




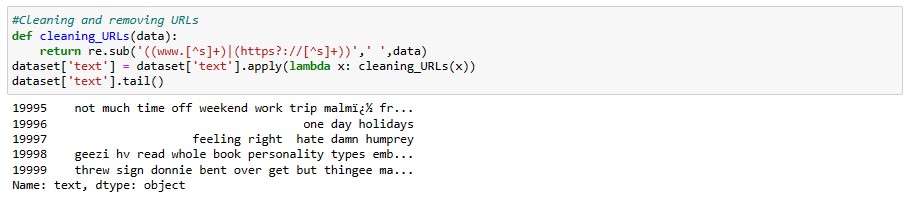
- Cleaning and Removing Punctuations: Punctuations are removed from the tweet text using the `cleaning\_punctuations` function.



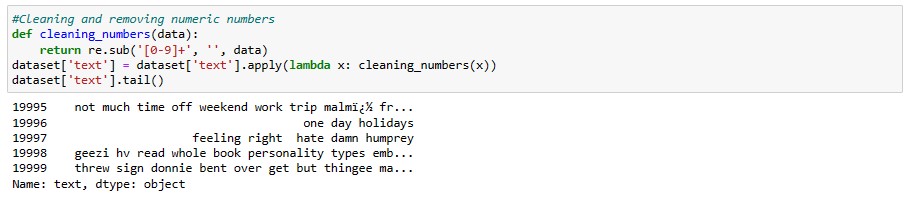
- Cleaning and Removing Repeating Characters: Repeating characters in the tweet text are removed using the `cleaning\_repeating\_char` function.



- Cleaning and Removing URLs: URLs are removed from the tweet text using the `cleaning\_URLs` function.



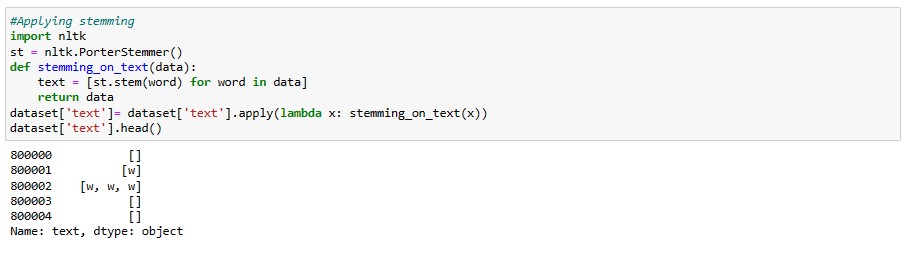
- Cleaning and Removing Numeric Numbers: Numeric numbers are removed from the tweet text using the `cleaning\_numbers` function.



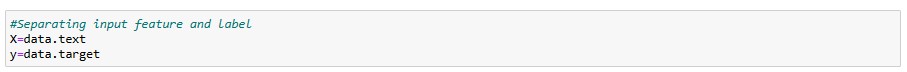
- Tokenization of Tweet Text: The tweet text is tokenized using the `RegexpTokenizer` from NLTK.



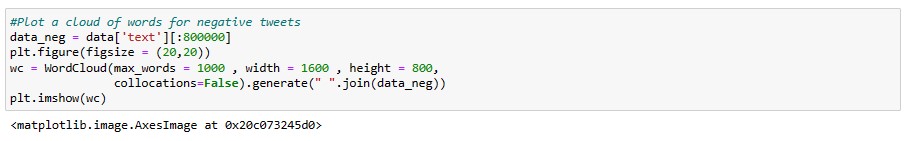
- Applying Stemming: Stemming is applied to the tokenized text using NLTK's PorterStemmer.

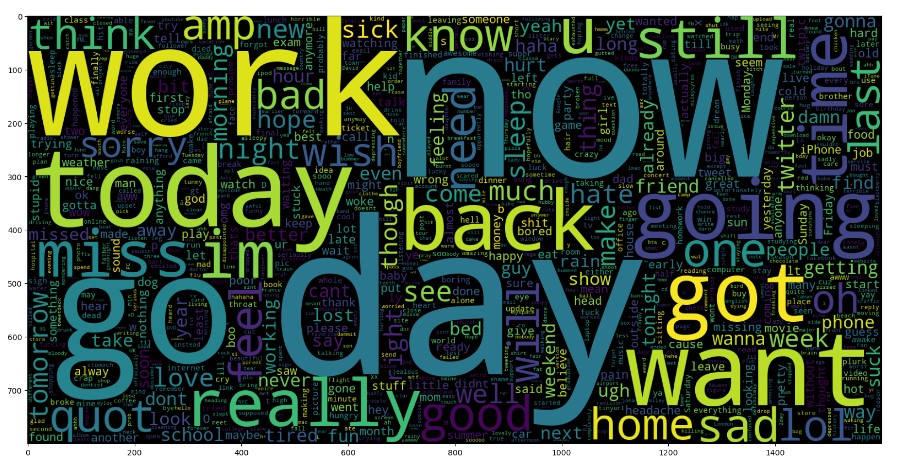


- Separating Input Feature and Label: The input feature (`X`) and label (`y`) are separated from the dataset.

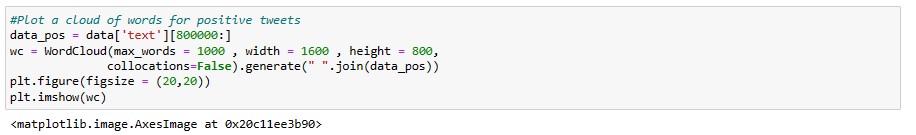


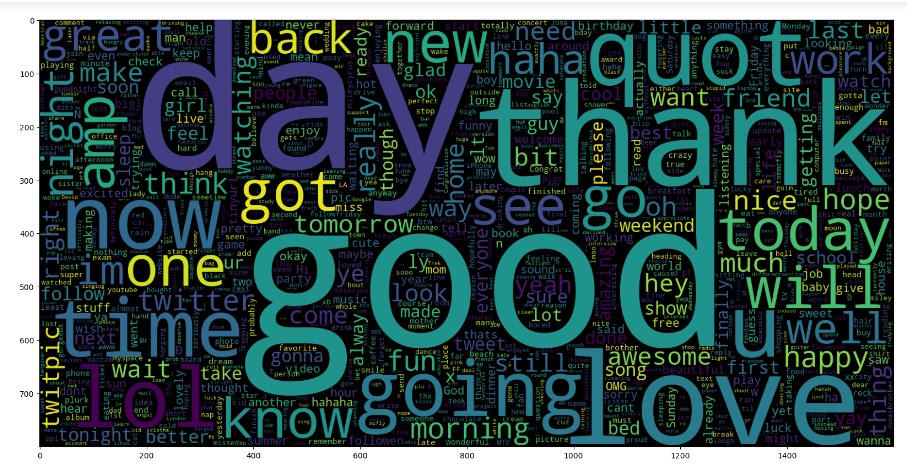
- Word Cloud for Negative Tweets: A word cloud is generated for the negative tweets.



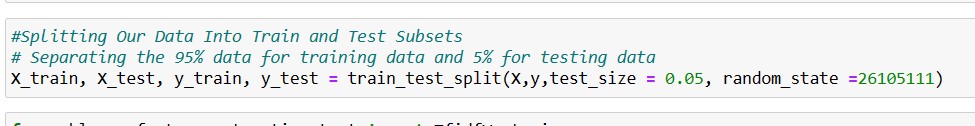


- Word Cloud for Positive Tweets: A word cloud is generated for the positive tweets.

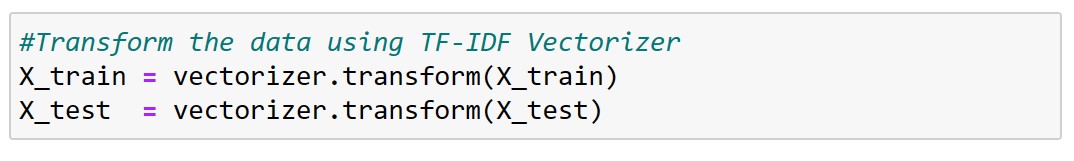




- Data Splitting for Training and Testing: The dataset is split into training and testing subsets using the `train\_test\_split` function from sklearn. The training set comprises 95% of the data, and the testing set comprises 5%.



- TF-IDF Vectorization: The text data in both the training and testing sets is transformed into numerical features using TF-IDF Vectorizer.



**# Insights:**

- The dataset contains 1.6 million records with six columns: 'target', 'ids', 'date', 'flag', 'user', and 'text'.

- The 'target' column represents sentiment with values 0 and 4, possibly indicating negative and positive sentiments, respectively.

- Extensive preprocessing steps are applied to clean and prepare the text data for sentiment analysis.

- Data exploration involves visualizations of sentiment distribution and word clouds for both positive and negative tweets.

- The dataset is split into training and testing sets for further analysis, likely sentiment classification using machine learning models.