Sentiment Analysis :Twitter Data

**1. Data Preprocessing**:

The initial steps involve importing necessary libraries and loading the dataset.

* Import Libraries

pandas for data manipulation.

numpy for numerical operations.

matplotlib.pyplot and seaborn for data visualization.

re for regular expression operations.

nltk for natural language processing tasks.

warnings to handle warnings during code execution.

* Load Dataset:

The dataset is loaded into a Pandas DataFrame (df) using pd.read\_csv().

* Data Inspection:

The head() method is used to display the first few rows of the dataset.

The info() method is used to get information about the dataset, including data types and missing values.

* Text Cleaning:

Define a function remove\_pattern() to remove specific patterns (e.g., Twitter handles) from the tweets.

Apply this function to remove Twitter handles from the tweet text.

Remove special characters, numbers, and punctuation marks using regular expressions.

Remove short words (less than 3 characters).

Tokenize the cleaned tweets to separate words.

Stemming is performed using the Porter Stemmer algorithm to reduce words to their root forms.

Combine the tokenized words into single sentences.

* Visualize Word Frequency:

Use WordCloud to visualize the most frequent words in the cleaned tweets.

Separate visualizations are created for all tweets, positive tweets, and negative tweets.

**2. Hashtag Extraction:**

Hashtags are extracted from the cleaned tweets to understand the topics or themes discussed in the tweets.

* Hashtag Extraction Function:

Define a function hashtag\_extract() to extract hashtags from tweets using regular expressions.

* Extract Hashtags:

Extract hashtags separately for positive and negative tweets.

* Visualize Top Hashtags:

Visualize the top 10 hashtags for both positive and negative tweets using bar plots.

**3. Feature Extraction:**

Features are extracted from the cleaned tweets to train the sentiment analysis model.

* CountVectorizer:

Use CountVectorizer from sklearn.feature\_extraction.text to convert text data into numerical features.

Parameters such as max\_df, min\_df, and max\_features are set to control the vocabulary size.

Stop words are removed using the built-in English stop word list.

* Split Dataset:

Split the dataset into training and testing sets using train\_test\_split from sklearn.model\_selection.

**4. Model Building and Evaluation:**

A logistic regression model is trained on the extracted features, and its performance is evaluated.

* Logistic Regression Model:

Train a logistic regression model using LogisticRegression from sklearn.linear\_model.

* Model Evaluation:

Evaluate the model's performance using F1 score and accuracy metrics.

* Threshold Adjustment:

Adjust the classification threshold using probability scores to optimize model performance.