**Twitter Sentiment Analysis using Machine Learning**

**1. Project Overview**

This project focuses on performing sentiment analysis on tweets to classify them as either positive or negative. We use the Sentiment140 dataset containing 1.6 million labeled tweets and apply Natural Language Processing (NLP) and Machine Learning techniques to build a classification model.

**2. Dataset Description**

* **Source**: Sentiment140 from Kaggle
* **Size**: 1.6 million tweets
* **Columns**:
  + target: Sentiment label (0 = negative, 4 = positive)
  + id: Tweet ID
  + date: Timestamp of the tweet
  + flag: Query (not used)
  + user: Username
  + text: The actual tweet content

We map the target label 4 to 1 to simplify binary classification:

* 0 → Negative tweet
* 1 → Positive tweet

**3. Data Preprocessing**

**Steps Performed:**

* Loaded data using pandas
* Renamed columns for readability
* Checked and handled missing values
* Mapped sentiment label 4 to 1

**Text Preprocessing:**

* Removed non-alphabetic characters
* Converted text to lowercase
* Tokenized and removed stopwords
* Applied stemming using **PorterStemmer**

A new column stemmed\_content was created to store the cleaned and stemmed text.

**4. Feature Engineering**

We used **TF-IDF (Term Frequency-Inverse Document Frequency)** to convert textual data into numerical vectors. This helps weigh important words more and common words less, making the model more effective.

**5. Model Training**

We trained a **Logistic Regression** model using scikit-learn with max\_iter=1000.

**Performance:**

* **Training Accuracy**: ~77.8%
* **Test Accuracy**: ~77.8%

The model shows good generalization and handles unseen data well.

**6. Model Evaluation**

The dataset was split using an 80-20 ratio:

* **Training Set**: 80%
* **Test Set**: 20%

We evaluated performance using the accuracy\_score metric.

**7. Saving and Loading the Model**

We used Python's pickle module to:

* Save the trained model as trained\_model.sav
* Load it later for making predictions on new or unseen data

**8. Prediction Example**

A tweet was selected from the test dataset and its sentiment was predicted using the loaded model.  
The model outputs either:

* 0 → Negative
* 1 → Positive

The result is printed and interpreted accordingly.

**9. Conclusion**

This project successfully demonstrates the application of Natural Language Processing and Machine Learning for binary sentiment classification on Twitter data.

**Key Achievements:**

* Achieved ~77.8% accuracy on both training and testing datasets
* Efficiently preprocessed and stemmed text data
* Used TF-IDF for feature extraction
* Deployed a Logistic Regression model for prediction

The model is effective for basic sentiment analysis tasks and can serve as a strong foundation for more complex improvements.