## **📄 Internship Task 4: Sentiment Analysis Using Machine Learning**

### **Introduction**

This report presents a sentiment analysis project focused on classifying text data (specifically tweets) as either *Positive* or *Negative*. The objective was to apply Natural Language Processing (NLP) techniques and build a machine learning model that can predict the sentiment of a given text input.

### **Dataset Description**

The dataset used consists of thousands of labeled tweets:

* **Features**: tweet (text)
* **Target**: label (0 = Negative, 1 = Positive)

The dataset was sourced from a public GitHub repository, containing a balanced number of positive and negative tweets.

### **Preprocessing Steps**

To prepare the textual data for machine learning, the following preprocessing steps were applied:

* **Text Cleaning**: Removed non-alphabetic characters and converted text to lowercase.
* **Tokenization**: Split the text into individual words.
* **Stopword Removal**: Removed common, uninformative words (e.g., *the*, *is*, *and*).
* **Stemming**: Reduced words to their root forms using Porter Stemmer.

Each tweet was transformed into a cleaned version suitable for vectorization.

### **Feature Extraction**

We used the **TF-IDF (Term Frequency–Inverse Document Frequency)** technique to convert text into numeric vectors that the machine learning model can process. We limited the features to the **top 5000 words** for efficiency.

### **Model Building**

We chose **Logistic Regression**, a basic yet powerful algorithm for binary classification. The dataset was split as follows:

* **80% Training Data**
* **20% Testing Data**

The model was trained using the TF-IDF feature vectors and corresponding sentiment labels.

### **Model Evaluation**

The model was evaluated on the test set using several metrics:

* **Accuracy**: 83%
* **Precision, Recall, and F1-score**: Showed balanced performance for both classes.
* **Confusion Matrix**: Displayed a low number of false predictions.

These results show that the model performs well for a simple baseline setup.

### **Confusion Matrix (Visual)**

|  |  |  |
| --- | --- | --- |
|  | **Predicted Positive** | **Predicted Negative** |
| **Actual Positive** | TP (True Positives) | FN (False Negatives) |
| **Actual Negative** | FP (False Positives) | TN (True Negatives) |

*A heatmap was plotted to visualize the matrix using Seaborn.*

### **Conclusion**

* This task successfully demonstrates how to build a sentiment analysis model using basic NLP and machine learning techniques.
* With minimal preprocessing and a simple logistic regression model, we achieved good accuracy.
* Future improvements may include:
* Using **LSTM** or **BERT-based** deep learning models.
* Applying **hyperparameter tuning** and **cross-validation**.