**Name Aoun Muhammad**

**Roll no 045**

**Project twitter sentiment analysis**

**Submit Sir Muhammad Rasikh Ali**

**Project**

**Twitter Sentiment Analysis**

* **Introduction**

The goal of this project is to create a web application that analyzes the feelings and emotions expressed in tweets related to specific topics or keywords. The application categorizes tweets as Positive, Negative, or Neutral. It also identifies emotions like happiness, sadness, anger, and more, based on the words used in the tweets.

* **2. Technologies Used**

The following tools and technologies were used in developing the project:

* **Python**: This is the main programming language used in the project.
* **Flask**: A simple web framework used to build the web application.
* **Scikit-learn**: A machine learning library that helps in creating a model to classify the sentiments of tweets.
* **Tweepy**: A Python tool that allows the program to fetch tweets from Twitter.
* **Pandas**: A library used for handling and storing the collected tweets.
* **TfidfVectorizer**: A tool used to convert tweet data into a form that can be processed by machine learning algorithms.
* **Pickle**: A Python tool for saving and loading machine learning models.
* **HTML/CSS**: Used to design the user interface of the web application.
* **3. Project Structure**

The project has several important parts:

* **Data Collection**: The system fetches tweets related to a specific keyword or topic from Twitter using the Tweepy library. Up to 1000 tweets can be fetched at once.
* **Data Preprocessing**: Once the tweets are fetched, unnecessary parts like mentions (@user), URLs, and retweets (RT) are removed. The remaining data is converted into a numerical format that can be used for machine learning.
* **Modeling**: A machine learning model is used to classify the sentiment of the tweets (Positive, Neutral, or Negative). The system also analyzes the emotions in the tweets by comparing the words to a predefined list of emotions.
* **Web Application**: The user can input a keyword or phrase, and the system fetches relevant tweets, analyzes their sentiment and emotions, and displays the results on a web page.
* **4. Data Collection**

Tweets are collected using the Tweepy API, which allows access to Twitter’s data.

**Steps for Data Collection:**

* **OAuth Authentication**: The program uses Twitter credentials to authenticate and connect to Twitter’s API.
* **Fetching Tweets**: A function is used to collect tweets that match a user-provided keyword. Up to 1000 tweets can be fetched in one go.
* **Cleaning Tweets**: After fetching, unnecessary information like retweets, mentions, and URLs are removed from the tweets to make the data cleaner for analysis.
* **5. Preprocess**

Before the tweets can be analyzed, they go through several cleaning steps:

* **Removing Retweets**: The "RT @user:" part of retweets is removed.
* **Removing Mentions**: User mentions, such as @user, are taken out.
* **Removing URLs**: Any web links (http://...) are removed from the tweets.
* **Normalization**: All text is converted to lowercase to ensure that the analysis treats all words consistently.
* **Text Vectorization**: The clean tweets are then transformed into a numerical format using the TfidfVectorizer, which prepares the data for machine learning.
* **6. Modeling**

The heart of the sentiment analysis is the machine learning model. Here's how it works:

* **TfidfVectorizer**: This tool turns the text into numbers that the machine learning model can understand. It does this by counting how often words appear in the tweets, ignoring common words that don't add much meaning (called stop words).
* **Sentiment Classification**: The machine learning model is trained to classify the sentiment of each tweet as Positive, Neutral, or Negative. It uses labeled data to learn how to predict the sentiment of new tweets.
* **Emotion Detection**: The system identifies emotions like happiness, sadness, anger, and others by comparing the words in the tweets with a list of words linked to different emotions.
* **7. Web Application (Flask Implementation)**

The web application is built using Flask. It has the following key features:

* **Routes**:
  + The homepage (/ or /home) lets users enter a keyword or phrase to search for related tweets.
  + The /predict route takes care of fetching tweets, analyzing their sentiment and emotions, and displaying the results.
* **Prediction Logic**:
  + **Sentiment Analysis**: The system then analyzes the sentiment of these tweets using the trained machine learning model.
  + **Emotion Analysis**: The emotions in the tweets are detected using the predefined emotion list.
* **Result Display**: The app shows the results of sentiment and emotion analysis on the results page. Users can see the sentiment of each tweet, as well as an overall summary of the emotions.
* **8. Key Features**

The application offers several useful features:

* **Live Tweet Fetching**: Users can enter any keyword or hashtag, and the system will fetch the latest tweets related to that keyword.
* **Sentiment Classification**: Each tweet is classified as Positive, Negative, or Neutral.
* **Emotion Detection**: The system also identifies emotions like happiness, sadness, and anger in the tweets.
* **Visual Output**: The results are shown in an easy-to-read format, including the counts of each sentiment and emotion.
* **9. Conclusion**

This Twitter Sentiment Analysis Web Application successfully demonstrates the power of machine learning and natural language processing to analyze the feelings and emotions in tweets. By using TfidfVectorizer for transforming text and a machine learning model for sentiment prediction, the system provides valuable insights into public opinions on various topics. The Flask-based web interface makes it easy and interactive for users.

This application can be especially useful in areas like brand monitoring, analyzing public opinions, and studying social media trends.

* **10. Future Work**

There are several ways the application could be improved in the future:

* **Improve Accuracy**: Try different machine learning models, like Support Vector Machines (SVM), Random Forest, or even deep learning, to improve the system’s accuracy.
* **Real-Time Data Processing**: Implement a system to fetch and analyze tweets continuously in real-time.
* **Expand Emotion Lexicon**: Add more emotions and words to the emotion detection list to improve its accuracy.
* **Multilingual Support**: Extend the application to handle tweets in multiple languages, making it usable worldwide.