**Project Title** : Twitter Sentiment analysis using machine learning

# **Team No : 2**

* 20KP1A4404 - Yamini
* 20KP1A4437 - Sowmya
* 20KP1A4456 - Vishnu
* 20KP1A4447 - Irfan

System architecture:

1. Data Collection:

Objective: Gather a diverse set of tweets for sentiment analysis.

Implementation: Utilize the Twitter API to access real-time tweet streams based on relevant search queries, hashtags, or user timelines.

Data Storage: Save the collected tweets in a dedicated database for subsequent processing and analysis.

2. Preprocessing:

Objective: Prepare the raw tweet data for analysis by removing noise and irrelevant information.

Cleaning: Eliminate non-essential elements such as URLs, mentions, and special characters to focus on the core text content.

Tokenization: Break down the text into individual words for further analysis.

Stop Word Removal: Exclude common words that do not contribute much to sentiment analysis results.

Stemming/Lemmatization: Reduce words to their base form to standardize the text.

3. Feature Extraction:

Objective: Transform the processed text into numerical features suitable for machine learning models.

Techniques: Apply methods like TF-IDF or use pre-trained word embeddings such as Word2Vec, GloVe, or BERT embeddings.

4. Sentiment Analysis Model:

Objective: Develop a model capable of discerning sentiment from textual data.

Machine Learning Options:

Naive Bayes: Suitable for its simplicity and efficiency.

Support Vector Machines: Effective for binary classification tasks.

Random Forests: Useful for handling non-linear relationships.

Deep Learning Models:

RNNs: Capture sequential information in tweets.

LSTMs: Address the vanishing gradient problem in RNNs.

Transformers: Efficiently process contextual information in text.

5. Model Training and Evaluation:

Process:

Split the dataset into training and testing sets.

Train the sentiment analysis model on the training set.

Evaluation Metrics: Assess the model's performance using metrics such as accuracy, precision, recall, and F1 score on the testing set.

6. Deployment:

Integration: Embed the trained model into a web application or API for real-time sentiment analysis.

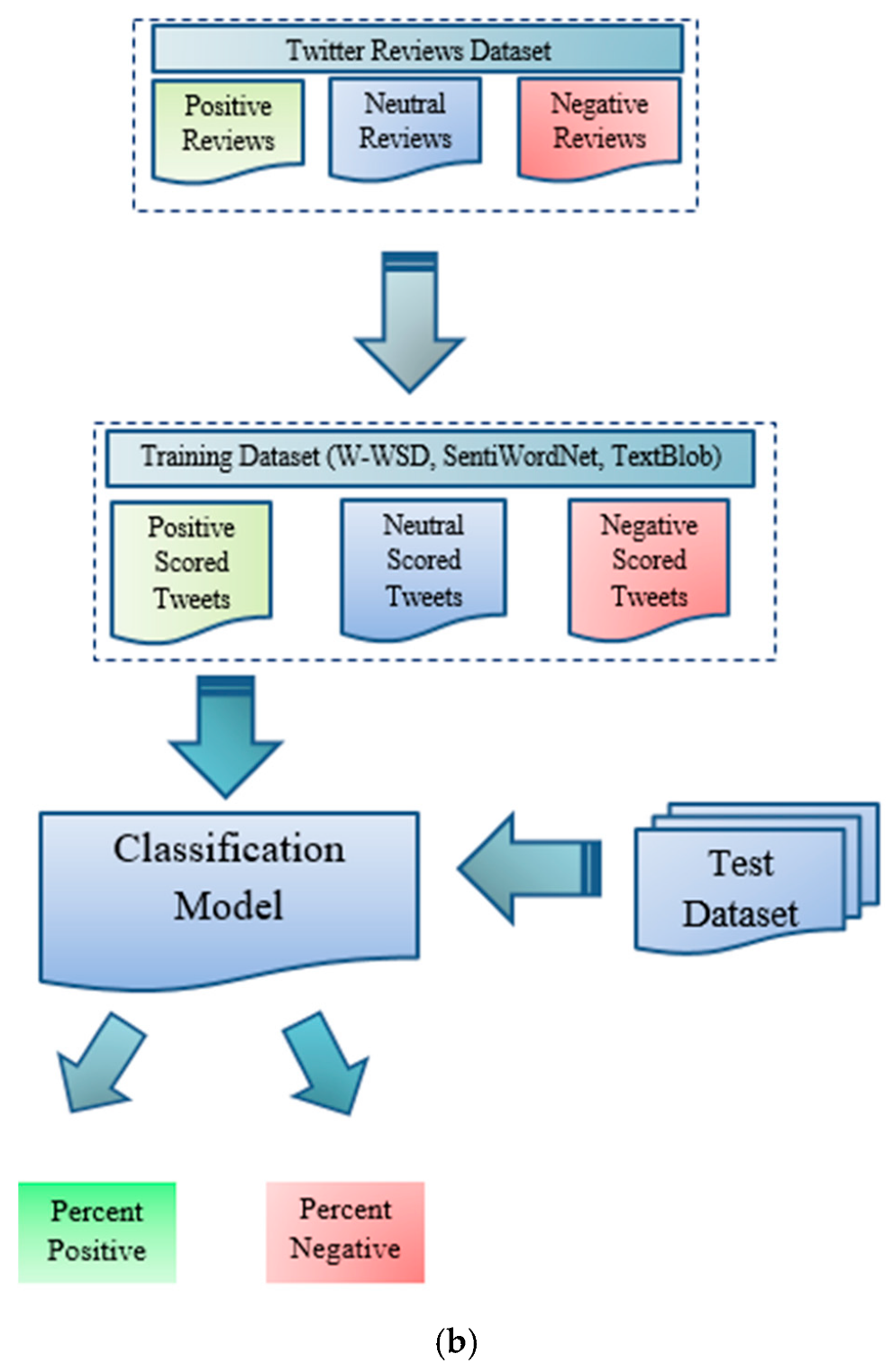
Scalability: Ensure the system can handle a large volume of tweets and user requests.

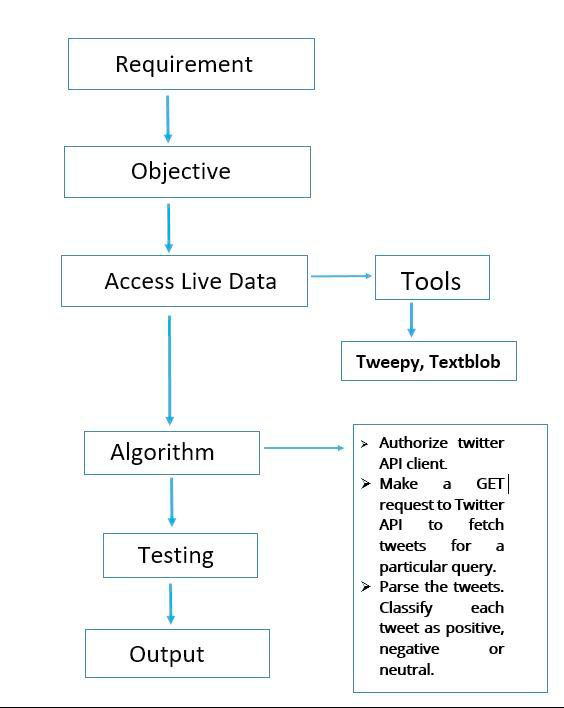
7. User Interface (Optional):

Objective: Provide an intuitive interface for users to interact with the sentiment analysis tool.

Visualization: Display results, trends, or sentiment scores in a user-friendly manner.

Data flow diagrams:





Algorithms used :

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import warnings

train = pd.read\_csv('drive/My Drive/Projects/Twitter Sentiment/train\_tweet.csv')

test = pd.read\_csv('drive/My Drive/Projects/Twitter Sentiment/test\_tweets.csv')

print(train.shape)

print(test.shape)

train.head()

test.head()

train.isnull().any()

test.isnull().any()

train[train['label'] == 0].head(10)

# checking out the postive comments from the train set

train[train['label'] == 1].head(10)

train['label'].value\_counts().plot.bar(color = 'pink', figsize = (6, 4))

length\_train = train['tweet'].str.len().plot.hist(color = 'pink', figsize = (6, 4))

length\_test = test['tweet'].str.len().plot.hist(color = 'orange', figsize = (6, 4))

out puts :

A screenshot of a computer

Description automatically generated

A screenshot of a black screen

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a black and white screen

Description automatically generated

A screenshot of a graph

Description automatically generated

A graph with different colored bars

Description automatically generated