INTRODUCTION

Sentiment analysis using machine learning involves training models to decipher and understand the emotional tone and intent behind text, enabling them to classify content as positive, negative, or neutral.

This aids businesses in extracting insights from customer feedback and social media, enabling informed decisions and accurate understanding of public perception.

In this sentiment analysis project, we explore the application of machine learning algorithms to classify text sentiment. By employing Naive Bayes, Support Vector Machine (SVM), Decision Tree, and Logistic Regression, we aim to effectively categorize text data into positive, negative, or neutral sentiments. Through comparative analysis of these algorithms, we seek to identify the most suitable approach for accurate sentiment classification, enhancing our understanding of their strengths and limitations.

DATASET: https://www.kaggle.com/datasets/dunyajasim/twitter-dataset-for-sentiment-analysis

METHODOLOGY

Listed below are the steps for the implementation of the project:

Certainly, here's the workflow of the sentiment analysis project using the mentioned machine learning algorithms:

1. Data Collection and Preprocessing: Gather a diverse dataset containing text samples with labeled sentiments (positive, negative, neutral). Clean and preprocess the text data by removing stopwords, special characters, and converting text to lowercase.

2. Feature Extraction: Convert the preprocessed text data into numerical features that machine learning algorithms can work with.

3. Data Splitting: Divide the dataset into training and testing subsets for model evaluation.

4. Model Selection: Choose Naive Bayes, SVM, Decision Tree, and Logistic Regression as the machine learning algorithms for sentiment classification.

5. Model Training: Train each algorithm using the training dataset and the extracted features.

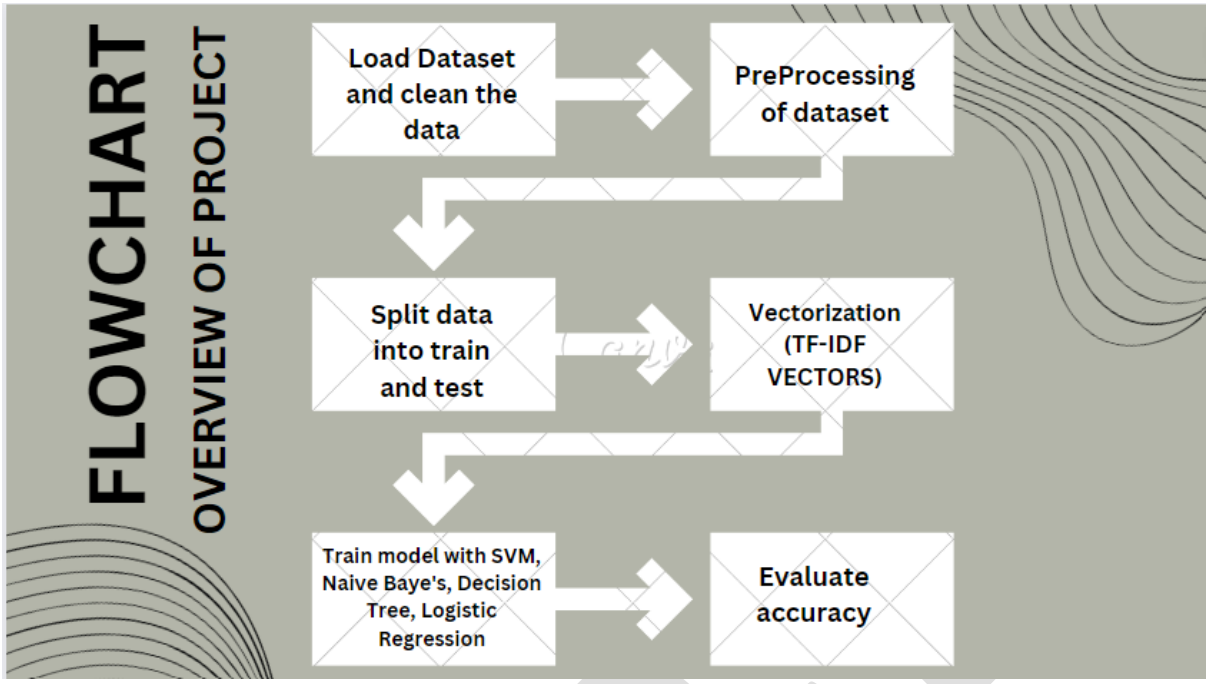
6. Model Evaluation: Assess the performance of each model using metrics such as accuracy, precision, recall, and F1-score on the testing dataset.

8. Comparative Analysis: Compare the performance of Naive Bayes, SVM, Decision Tree, and Logistic Regression models based on evaluation metrics.

9. Visualization: Create visualizations _confusion matrix) to illustrate the comparative performance of the models.

10. Model Selection and Deployment: Choose the algorithm with the best overall performance for sentiment classification.

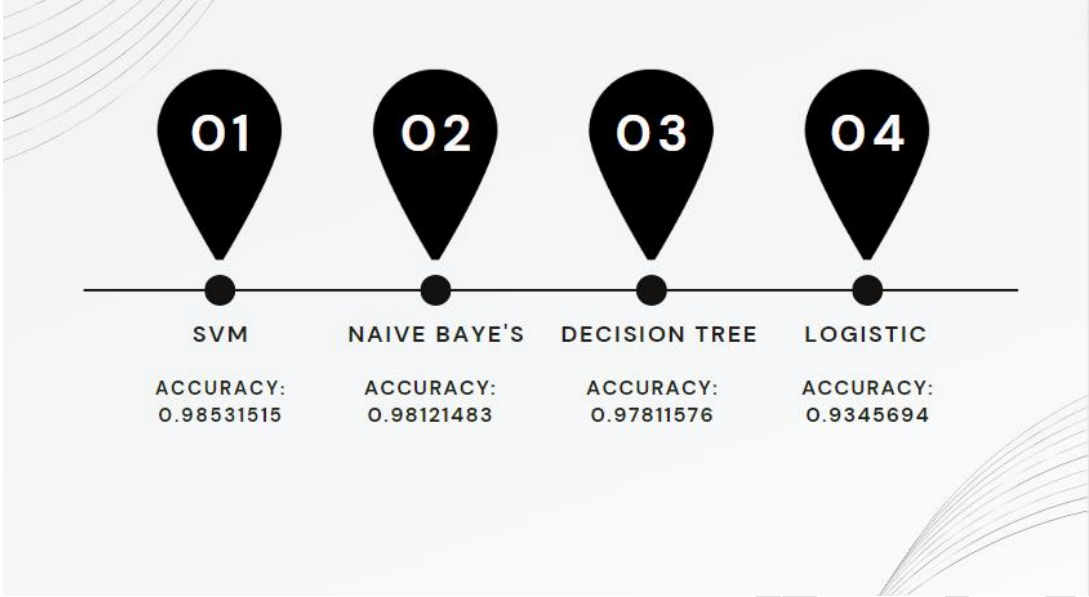
FLOWCHART



ALGORITHMS AND TECHNIQUES USED:

1. Data cleaning and preprocessing
    - Convert to lowercase and Remove special characters, punctuation, etc
    - Created functions to remove special characters, URL's, stop words and function.
    - Removed null values from the dataset
2. Vectorisation:
    - We have used TF-IDF from sci-kit learn library. TF-IDF (Term Frequency-Inverse Document Frequency) is a vectorization technique in natural language processing. It measures the importance of a word in a document within a collection.
    - High TF-IDF values indicate words that are unique to the document, helping capture its context while downplaying commonly used words.
    - Tokenization: The technique of tokenizing involves dividing text into separate tokens (words or sub words). It separates the text into manageable processing parts.

3. Model Selection:
   We have selected four models to work with:
    - SVM (Support Vector Machine)
    - Naïve Baye's
    - Decision Tree
    - Logistic Regression

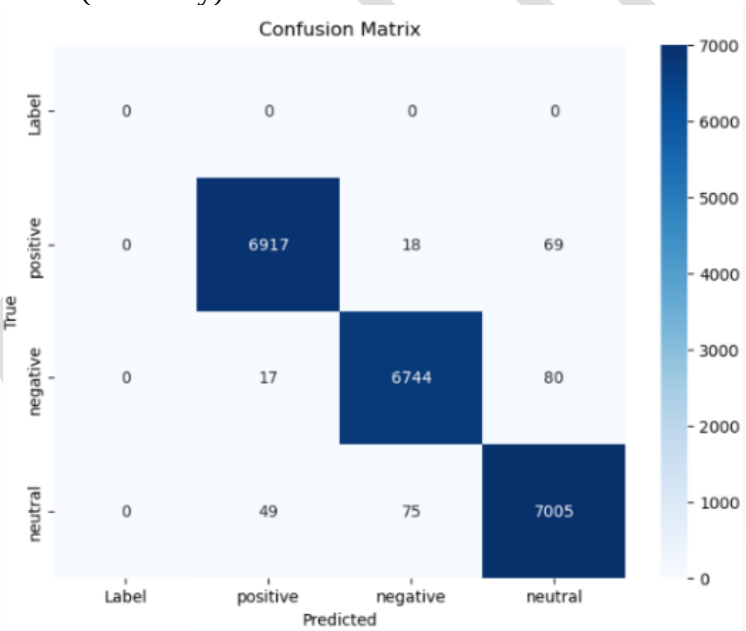


4. EVALUATION:

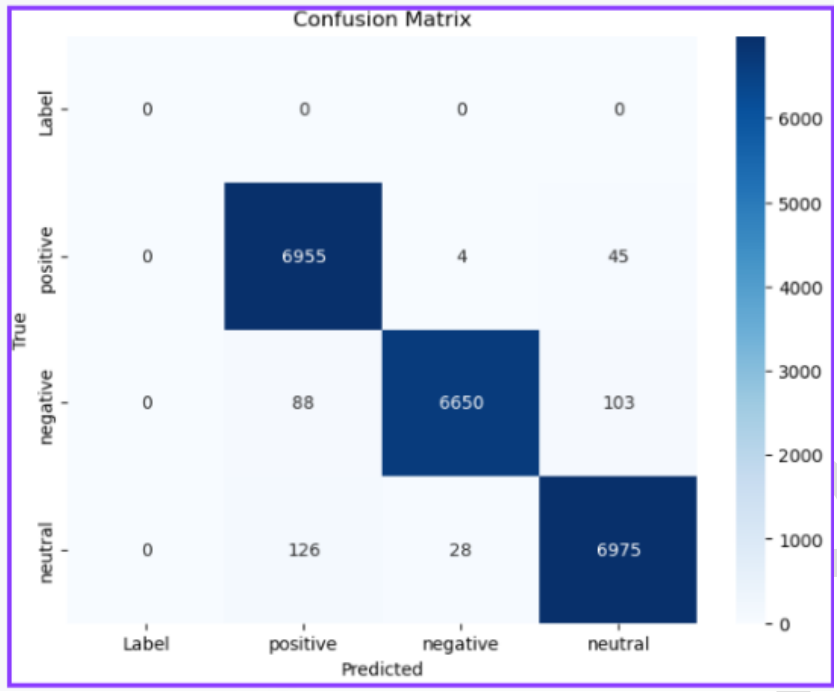
CONFUSION MATRIX

A confusion matrix is a tabular representation in machine learning that shows the true and predicted classes, enabling evaluation of classification model performance through metrics like accuracy and precision.

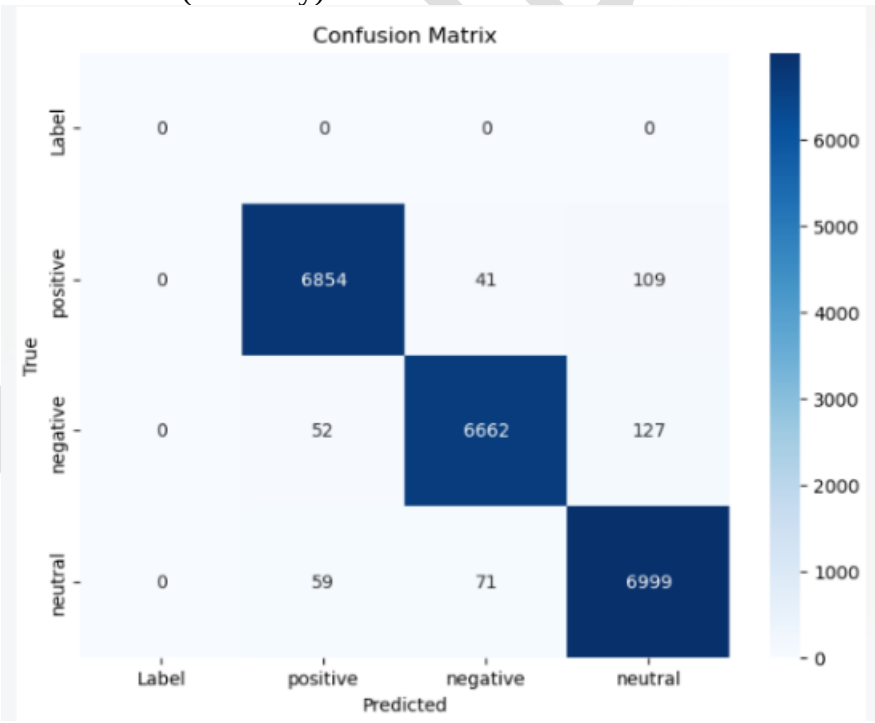
1. SVM (Accuracy): 0.98531515



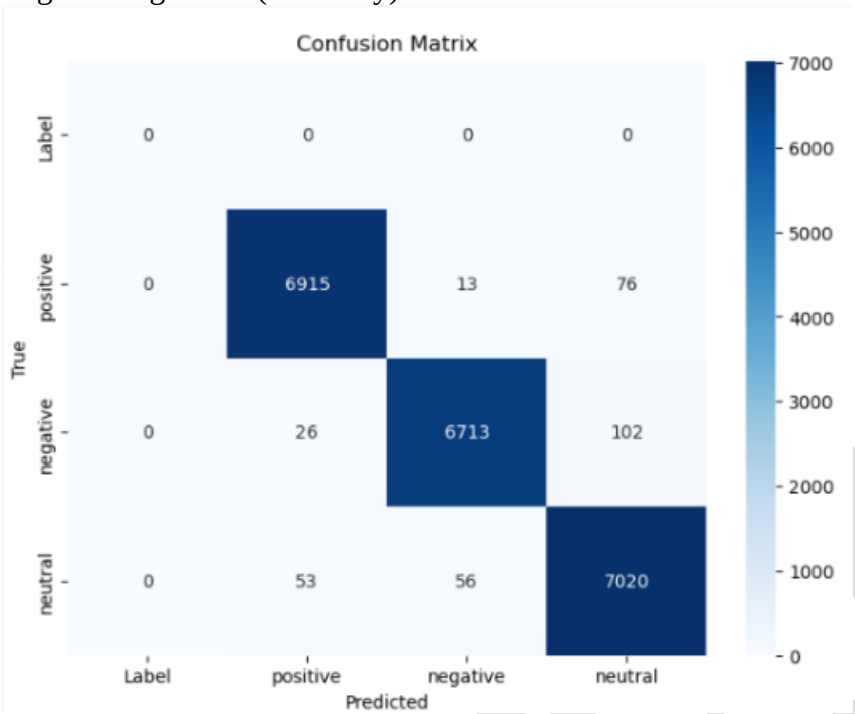
2. Naïve Baye's (Accuracy): 0.98121483



3. Decision Tree(Accuracy): 0.97811576

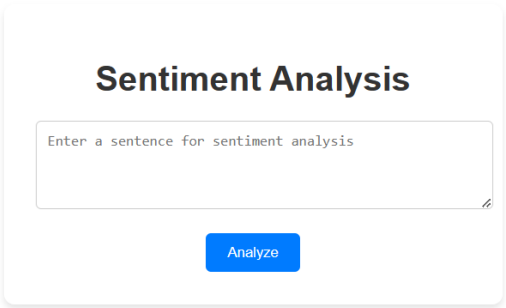


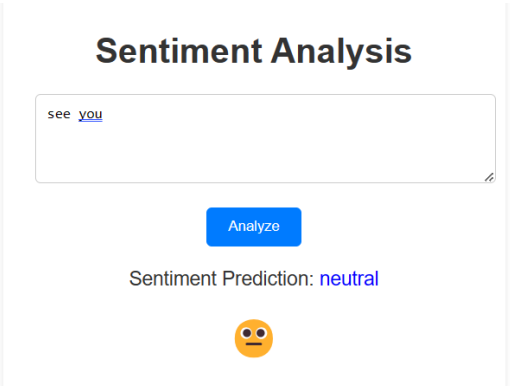
4. Logistic Regression(Accuracy): 0.9345694

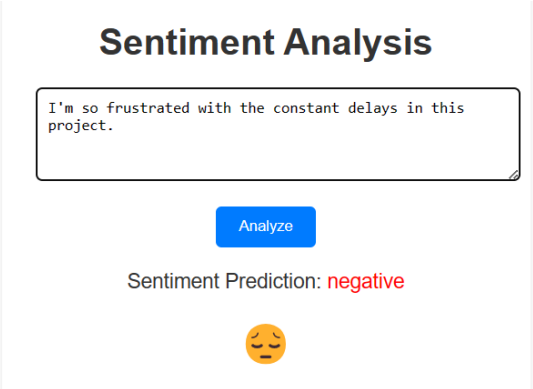


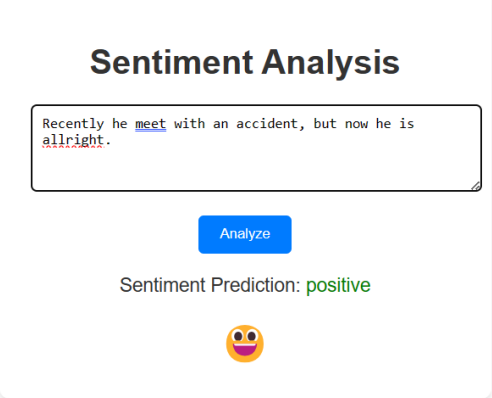
5. WEB APP: We have deployed our project with SVM model with Flask python framework which enables creating an interactive web app. Firstly install all the required libraries and frameworks by simply using pip install requirement.txt file. Then you can run app.py file easily. After that you will be provided with an local host post address which will redirect you to our web App.

Snippets of web-app:









CONCLUSION: In conclusion, this sentiment analysis project employed Naive Bayes, SVM, Decision Tree, and Logistic Regression algorithms to classify text sentiments. Through thorough evaluation and comparison, we identified the most effective algorithm (i.e. SVM), enhancing our understanding of their distinct strengths. This project underscores the significance of algorithm selection in achieving accurate sentiment classification in text data.