Sentiment Pulse - Web application

CS595- TERMINATION PROJECT

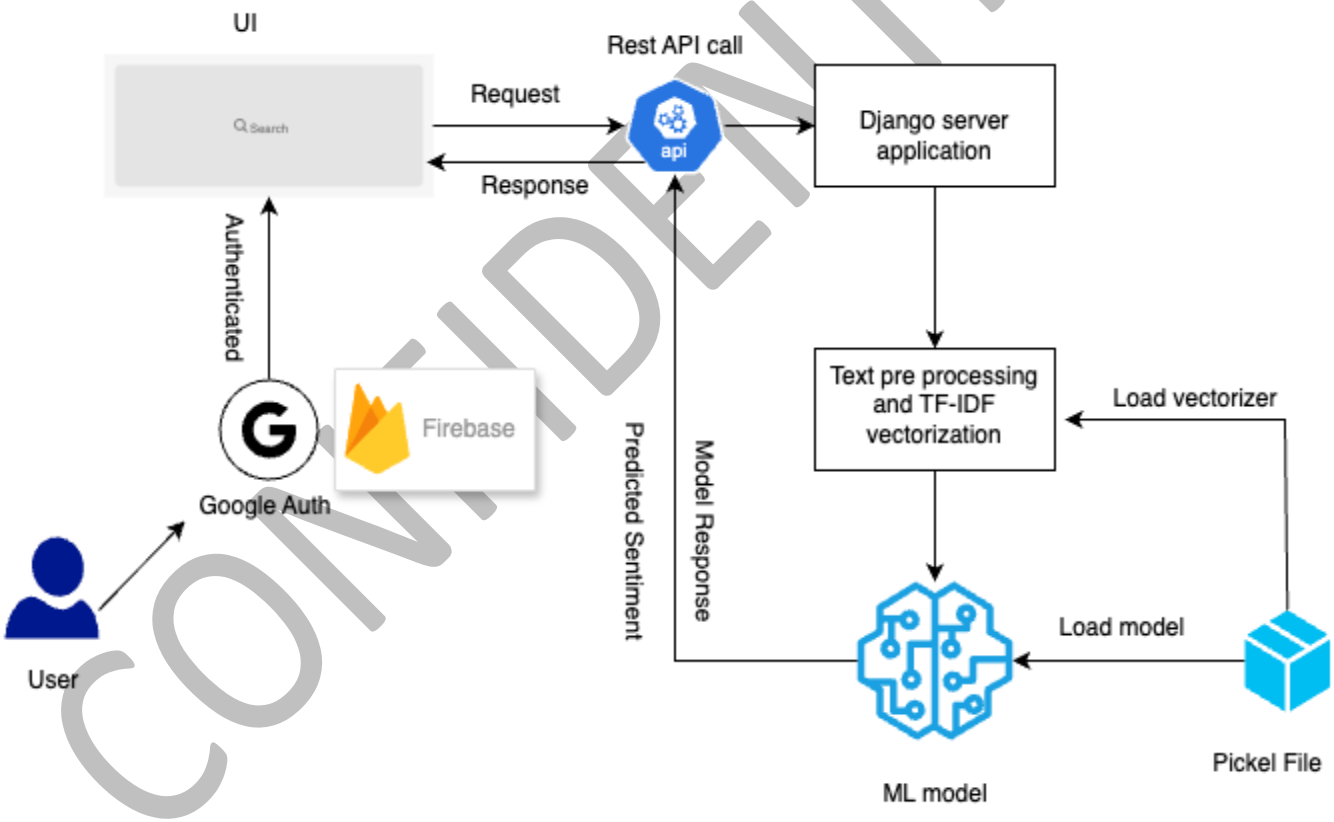
B00981964

Abstract

This project, Sentiment Pulse is a web application that predicts the sentiment of a single text or tweet. The project aims to decode the sentiment of the text using a trained machine learning model. The application provides valuable insights into public opinions, emotions and attitudes.

Sentiment analysis, also known as opinion mining, is a natural language processing (NLP) and machine learning (ML) technique used to determine the sentiment expressed in a piece of text. The goal of sentiment analysis is to identify and extract subjective information from the text and classify it as positive, negative, or neutral.

Architecture



Tweet to Insight: Application Flow

User is authenticated using Google Auth backed by Google Firebase database and once the authentication is successful, he/she enters into the application search page. User enters the tweet data or text in the text area provided and the request payload is sent to the back-end Django server via a REST API call. The input text is preprocessed to remove irrelevant information,

such as stop words, punctuation, and special characters or emojis. It may also involve stemming or lemmatization to reduce words to their base form. The cleaned text is then transformed using a TF-IDF vectorizer for the data to be fed to the pre trained machine learning model. The vectorizer and the model are loaded from a pickle file. The predicted model response is sent back to the client and the predict sentiment class is displayed on the UI screen.

Technology Stack

Next.js, Typescript – front end (UI), Django - Python (back end), Google firebase (for Google Authentication)

Application Setup and Execution

UI

sentiment-front is the working directory for the UI screen. Execute **npm install** to install the required dependencies. Later execute **npm run dev** to run the application. The application is hosted on port 3000 by default.

Back end – Django application

In the analysis-core directory execute **source venv/bin/activate** to activate python virtual environment and then execute **python manage.py runserver**, the server will be up and running on port 8000 by default.

Model

The machine learning model is pre trained following the principles of natural language processing using a large dataset, model's performance is evaluated, and the model is stored in a pickle file. This eliminates the need to train the model each time a request (single tweet text) is sent to the server. Details about model processing is documented here. Sentiment Analysis

Source Code

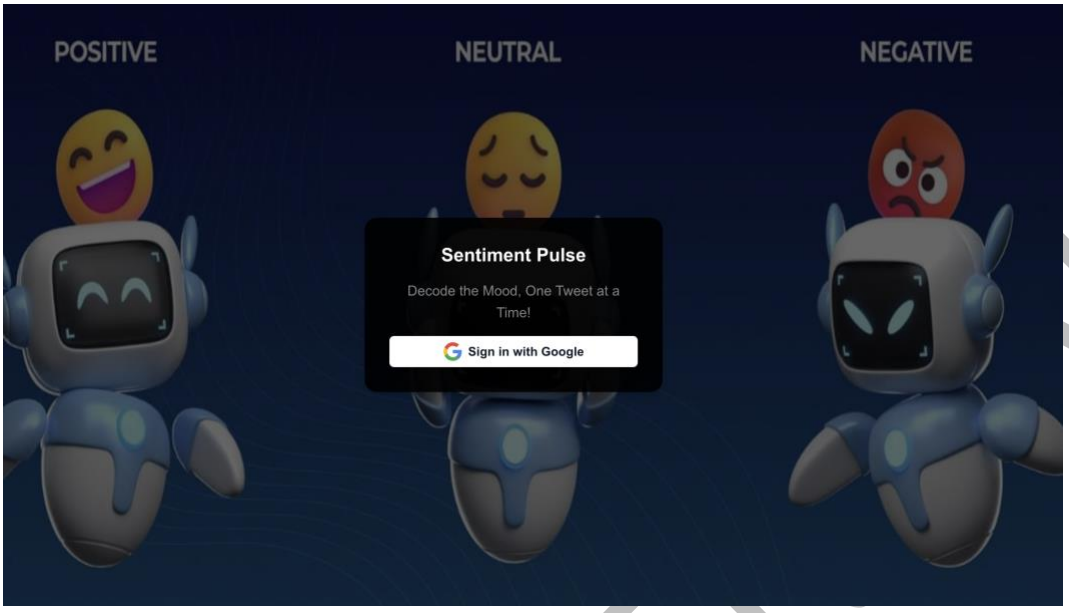
The code is available on github.

Future Scope

The project can be extended to fine tune the model to accept multilingual data or instead of the pre trained model, we can use any of the existing Large language models (LLM) like Open AI or Google Gemini.

UI Screen

The UI has 2 screens, the landing page (login page) and the home page for prediction.



Once the user is authenticated, we have a text box to enter tweet data and the Predicted Sentiment class is displayed on the screen.