Artificial intelligence and Machine learning

Project Documentation

1. Introduction

- **Project Title:** TrafficTelligence: Advanced Traffic Volume Estimation with Machine Learning
- **Team Members:**

Name	Role
J.V.S.Sujan	Python Development, Testing
G.Raj Kumar	Front end development
P.Akshay Kumar	Front end development

2. Project Overview

- **Purpose:**

The purpose of *TrafficTelligence* is to develop an intelligent, machine learning-driven system for accurate and real-time traffic volume estimation. By integrating data from multiple sources—such as surveillance cameras, GPS, and environmental sensors—the system aims to overcome the limitations of traditional traffic monitoring methods. This innovation will support urban planners, traffic engineers, and policymakers in making informed decisions to reduce congestion, improve road safety, and enhance commuter experience. The project also seeks to promote efficient resource allocation, optimize traffic signal timing, and contribute to the development of smarter, more sustainable cities through advanced data analytics and predictive modelling.

- **Features:**
    - Estimates the traffic volume accurately
    - User friendly interface
    - Takes into account environment (temperature, rain, weather, snow etc) information and date while calculating the traffic volume.

3. Architecture

- **Frontend:**
    - Used HTML to display the data on the background on screen

    - Also used to take the inputs from user
- **Backend:**
    - Used machine learning algorithms using Python to train and test the model
    - Used flask framework to give the data to HTML
- **Database:**
    - Data used for training the model is obtained from - https://drive.google.com/file/d/1iV5PfYAmI6YP0_0S4KYy1ZahHOqMgDbM/view

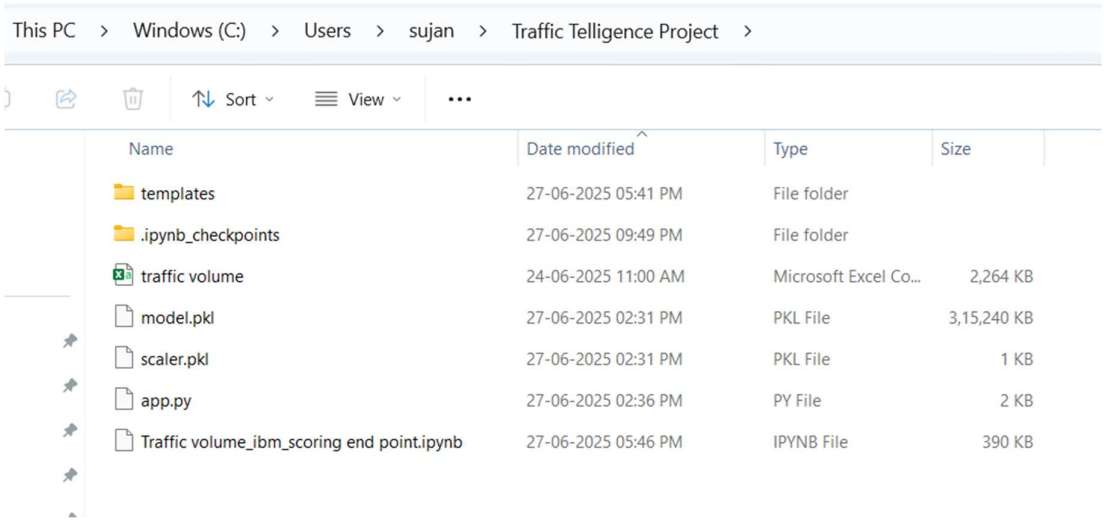
4. Setup Instructions

- **Prerequisites:**
  Anaconda Software, Anaconda Prompt, Jupyter Notebook, Spyder 3

- **Installation:**
    1. **Install Anaconda Software**
        1. Go to https://www.anaconda.com/products/distribution.
        2. Download the latest version for your OS (Windows/macOS/Linux).
        3. Run the installer → Click **Next** → Accept license → Install with default settings.
    2. **Open Anaconda Prompt**
        1. After installation, search **"Anaconda Prompt"** in your Start menu (Windows).
        2. Click to open — it's like Command Prompt with Python and Conda preloaded.
        3. You can run Python code or manage environments here.
    3. **Launch Jupyter Notebook**
        1. Open **Anaconda Prompt** and type: jupyter notebook → Press Enter.
        2. Your browser will open with the Jupyter interface.
        3. Click **New > Python 3** to start coding in a notebook.
    4. **Open Spyder IDE**
        1. From the Start menu, search **"Spyder"** (or open Anaconda Navigator and click Spyder).
        2. Spyder launches as a full Python IDE, great for data science projects.
        3. You can write and run scripts in the editor and view results in the console.

  Open anaconda prompt as administrator.
    - Type “pip install numpy” and click enter.
    - Type “pip install pandas” and click enter.
    - Type “pip install matplotlib” and click enter.
    - Type “pip install scikit-learn” and click enter.
    - Type “pip install Flask” and click enter.
    - Type “pip install xgboost” and click enter.

5. Folder Structure



6. Running the Application

- Open Anaconda prompt and change the directory where the app.py file is located
- Type the command – python app.py
- This will give link - http://127.0.0.1:5000 which needs to be opened on browser.
- This will display the application where the user can select all the required values
- After selecting all the values, click on predict.
- This will display the output i.e. estimated traffic volume.

7. API Documentation

1. Libraries Used

Library	Purpose
numpy	Handles numerical arrays and data (np.array)
pandas	For creating and manipulating DataFrames (pd.DataFrame)
pickle	Loads the pre-trained model and scaler (model.pkl, scaler.pkl)
joblib	Imported but **not used** in this script
matplotlib.pyplot	Imported but **not used** in this script
os	Used for getting the port number from environment variables
Flask	Main framework for building the web application
render_template, request	Flask functions for handling HTML forms and rendering pages

2. **Flask Endpoints (API)**

Endpoint	Method(s)	Purpose
/	GET	Loads the homepage (index.html)
/predict	POST, GET	Takes user input, predicts volume, shows result

3. **Machine Learning Components**

Component	Description
model.pkl	Pre-trained ML model loaded using pickle
scaler.pkl	Pre-trained StandardScaler (or similar) used to scale input features
model.predict()	Used to predict traffic volume after scaling the input

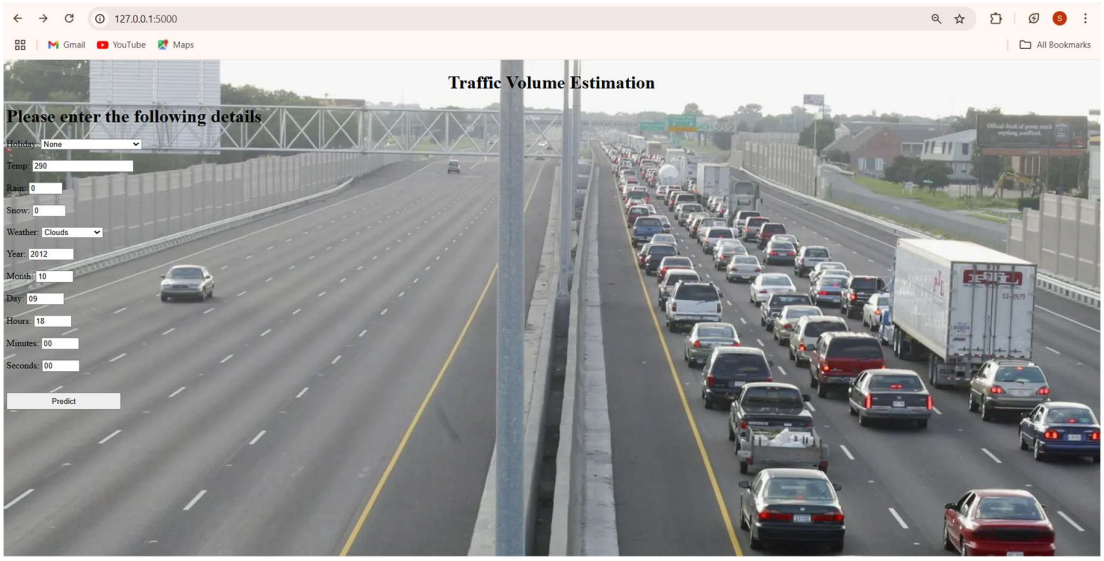
4. **Input/Features Used**
   'holiday', 'temp', 'rain', 'snow', 'weather', 'year', 'month', 'day', 'hours', 'minutes', 'seconds'

8. Authentication

Form Submission - Data is collected from an HTML form using request.form.values().

Model Prediction - The only backend logic is loading the model and returning predictions.

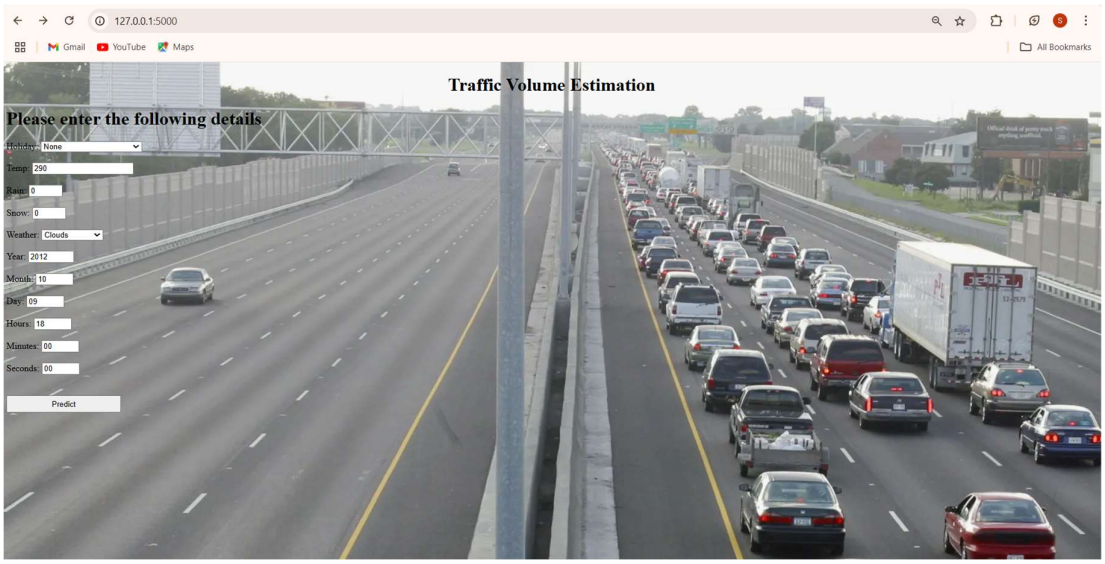
9. User Interface

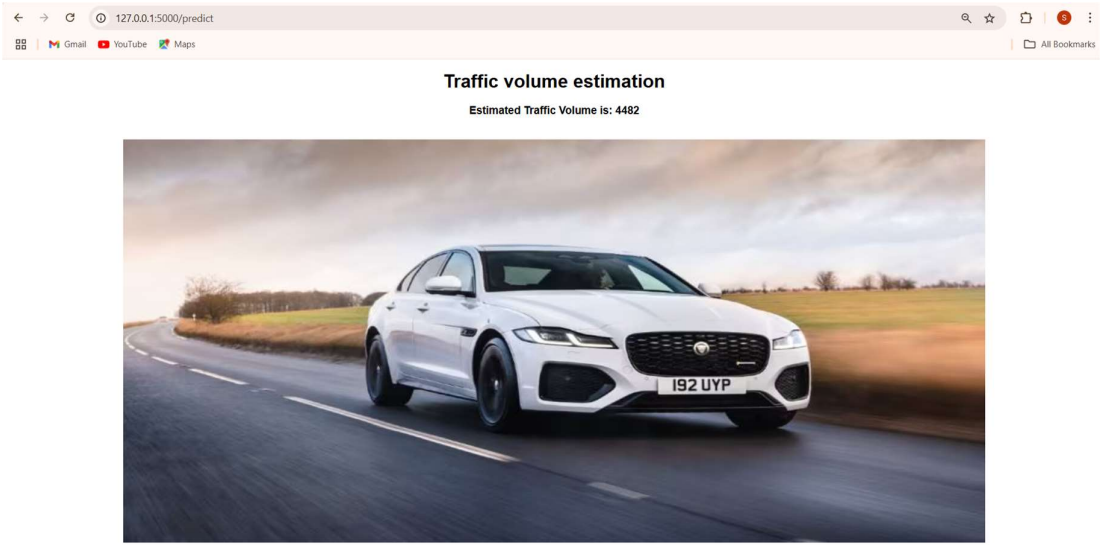


10. Testing

LinearRegression()
DecisionTreeRegressor()
RandomForestRegressor()
SVR()
XGBRegressor()
All these Algorithms are used to train the data and later test the data.

11. Screenshots or Demo





12. Future Enhancements

TrafficTelligence can be used with integrating advanced data sources such as live video feeds, GPS data from vehicles, and IoT-based traffic sensors for more granular predictions. Incorporating deep learning models like CNNs and LSTMs could further enhance accuracy by capturing spatial and temporal patterns. The system can be expanded to support traffic incident detection, congestion forecasting, and adaptive signal control. Integration with smart city platforms and mobile applications could provide real-time traffic insights to commuters. Additionally, the model can be trained for region-specific behavior, making it applicable in diverse urban settings worldwide, contributing to smarter and greener cities.