TrafficTelligence – AI/ML-Based Traffic Volume Estimation System

Project Title: Traffic Telligence – AI/ML-Based Traffic Volume Estimation System

Team Members:
Team Leader: S. Varshitha
Team Member 01: M. Guna Sindu
Team Member 02: M. Sahithi
College Name: Sri Venkateswara College of Engineering

1. Introduction

Traffic Telligence addresses escalating urban traffic congestion using Artificial Intelligence (AI) and Machine Learning (ML). Its goal is to provide a smart, predictive solution for accurate traffic volume estimation. By transforming data into actionable insights, TrafficTelligence aims to improve urban mobility, reduce commute times, lower fuel consumption, and decrease carbon emissions, contributing to more sustainable cities.

2. Project Overview

Purpose:

To offer a smart AI/ML-based solution for estimating traffic volume, enabling data-driven decisions for city planners and commuters.

Features:

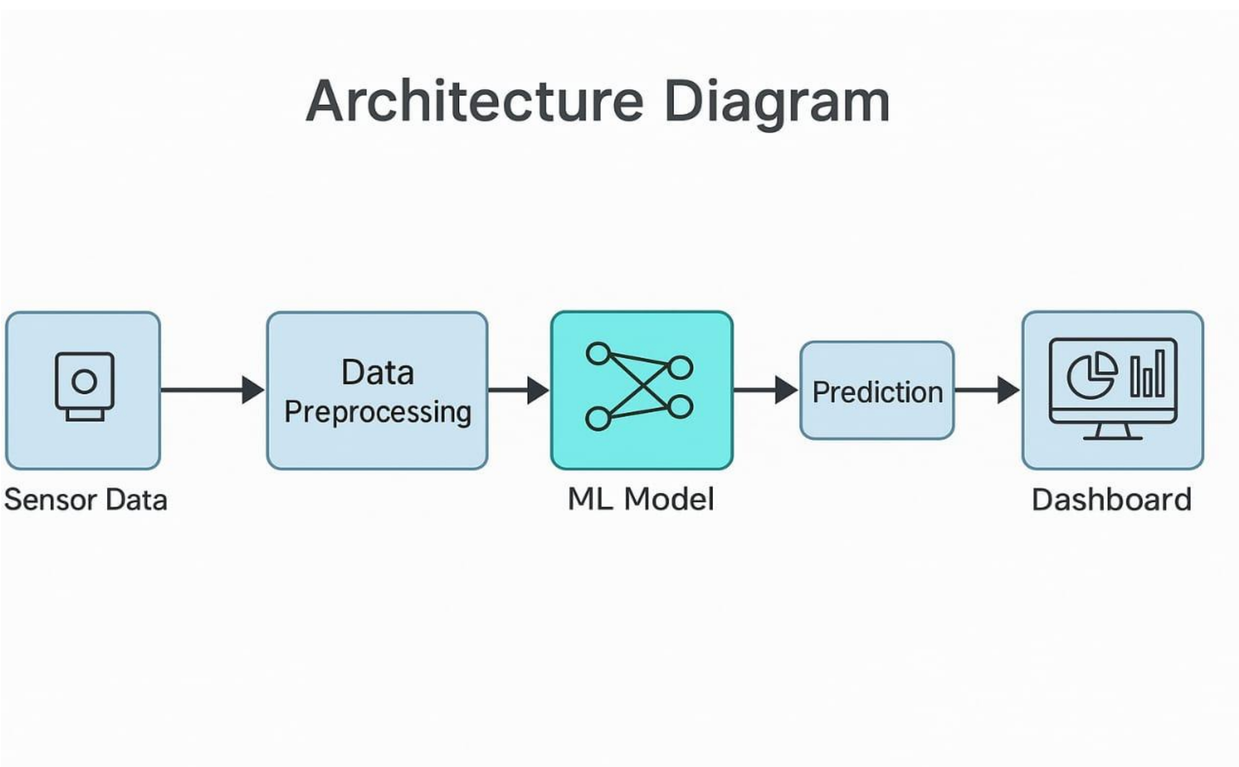
- **Real-time and Historical Data Analysis:** Processes both past and future live traffic data for comprehensive understanding.
- **Predictive Modeling using Random Forest:** Employs the robust Random Forest algorithm for accurate traffic volume forecasting.
- **Interactive Dashboards for Visualization:** Provides intuitive visual interfaces for understanding traffic patterns and predictions.
- **Feature Importance Analysis:** Identifies key factors influencing traffic volume, offering actionable insights for interventions.

3. Architecture

TrafficTelligence features an efficient, scalable, and modular architecture focused on its machine learning pipeline.

- **Frontend:** Not applicable (user interaction is primarily via Jupyter Notebook/Google Colab outputs).
- **Backend:** Developed in Python using Jupyter Notebook/Google Colab, leveraging scikit-learn for ML, and pandas and numpy for data manipulation.
- **Database:** Utilizes static public traffic record datasets (CSV), including fields like date, time, weather, and holiday status.

Architecture Diagram:



4. Setup Instructions

Setting up TrafficTelligence involves installing Python and essential data science libraries.

Prerequisites:

- Python 3.x (3.7+ recommended)
- Jupyter Notebook / Google Colab
- Required Python Libraries: `pandas`, `numpy`, `matplotlib`, `seaborn`, `scikit-learn`

Installation:

```
pip install pandas numpy matplotlib seaborn
scikit-learn
```

(Anaconda is recommended for local setup)

- Clone the repository or open the notebook in Google Colab
- Upload the dataset and run cells sequentially

5. Folder Structure

```
TrafficTelligence/
├── data/                    # Dataset files
(e.g., traffic_data.csv)
├── notebooks/              # Jupyter notebooks
(e.g., main_pipeline.ipynb)
├── visualizations/          # Output
graphs/images
├── model/                   # Saved ML models
(e.g., random_forest_model.pkl)
├── scripts/                 # Optional utility
scripts
├── config/                  # Optional config
files
├── README.md                # Project
description
└── requirements.txt         # List of
dependencies
```

6. Running the Application

TrafficTelligence is a notebook-based analytical tool. Its execution involves running the cells within the main Jupyter/Google Colab notebook.

You can run:

```
!python final_code.py
```

Or just execute the cells sequentially. Output is shown inside the notebook — no separate frontend/backend required.

7. API Documentation

Note: No REST API is deployed currently. Future versions plan to include Flask-based APIs.

Planned Endpoints:

- `/predict (POST)`
  - Request: `{"date": "2025-07-01", "time_of_day": "08:00", ...}`
  - Response: `{"predicted_traffic_volume": 1250, "unit": "vehicles/hour"}`
- `/batch_predict (POST)`
- `/health (GET)`

8. Authentication

Currently, no authentication. Future plans include:

- Token-based Authentication (OAuth 2.0 / JWT)
- User Roles & Permissions
- API Key Management
- Integration with SSO/LDAP

9. User Interface

Visualizations inside notebook:

- **Wireframe Sketch:** *[Refer to the dashboard sketch]*
- **Feature Importance Chart:** Bar chart showing feature impact
- **Actual vs Predicted Traffic:** Time-series plot

Dashboard Includes:

- Traffic Prediction Graph
- Alerts Panel
- Day-wise Analysis

10. Testing

Model is tested using 80-20 train-test split.

Metrics Used:

- MAE (Mean Absolute Error): ~770
- MSE (Mean Squared Error): ~1,035,689
- R^2 Score: ~0.739 (74% variability explained)

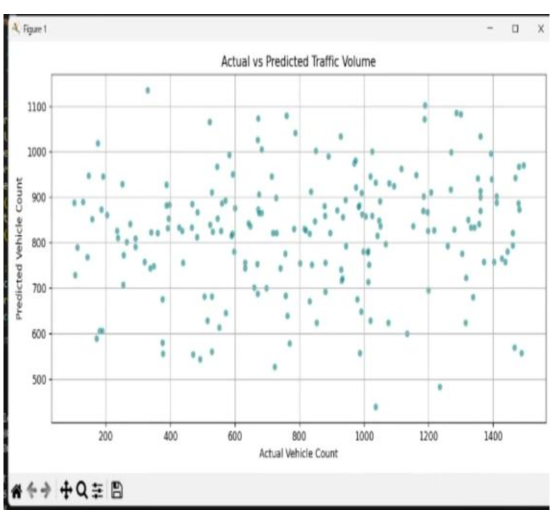
11. Screenshots or Demo

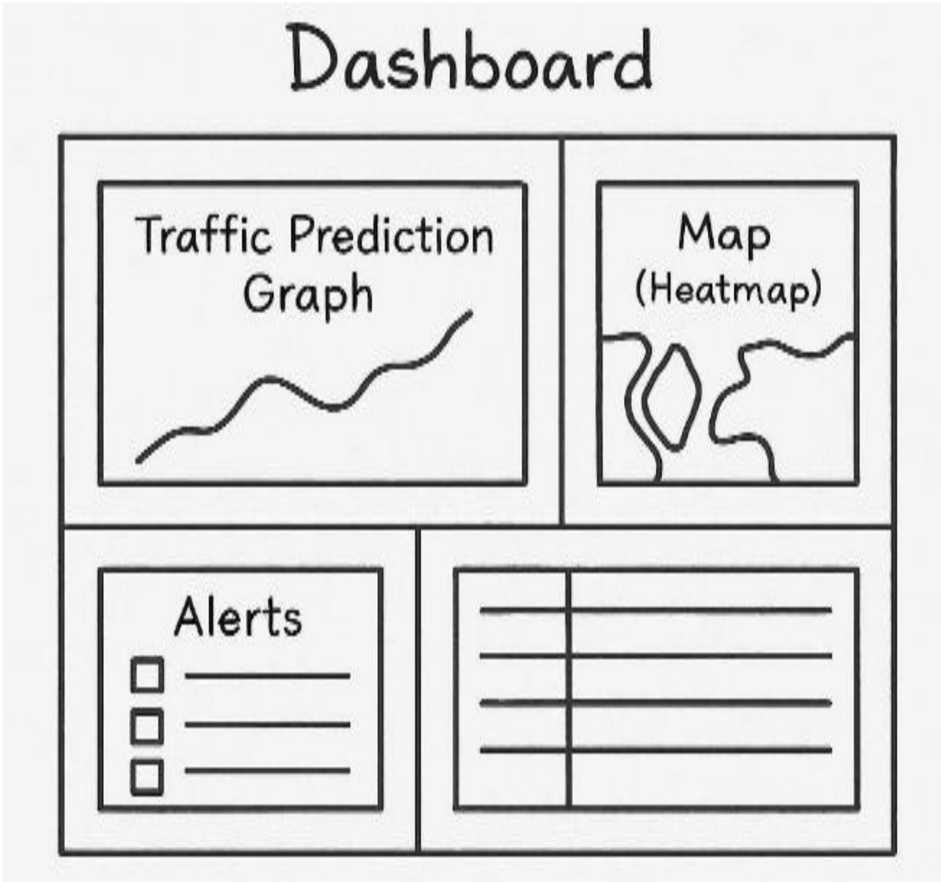
Model Output (Console):

```
Mean Absolute Error (MAE): 770.25
Mean Squared Error (MSE): 1035689.78
R-squared (R²): 0.7391
```

Dashboard Includes:

- Actual vs Predicted Graph




Dashboard
Traffic Prediction Graph
Map (Heatmap)
Alerts

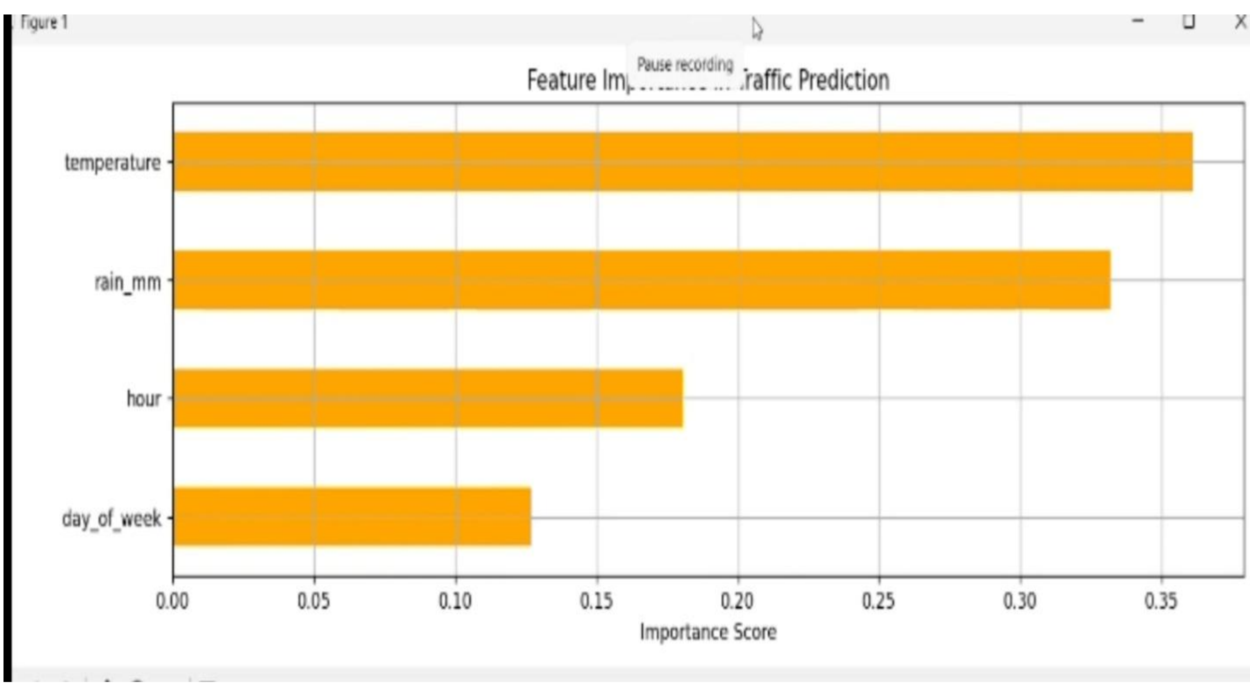


Figure 1
Pause recording
Feature Im Traffic Prediction
temperature
rain_mm
hour
day_of_week
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
Importance Score

12. Known Issues

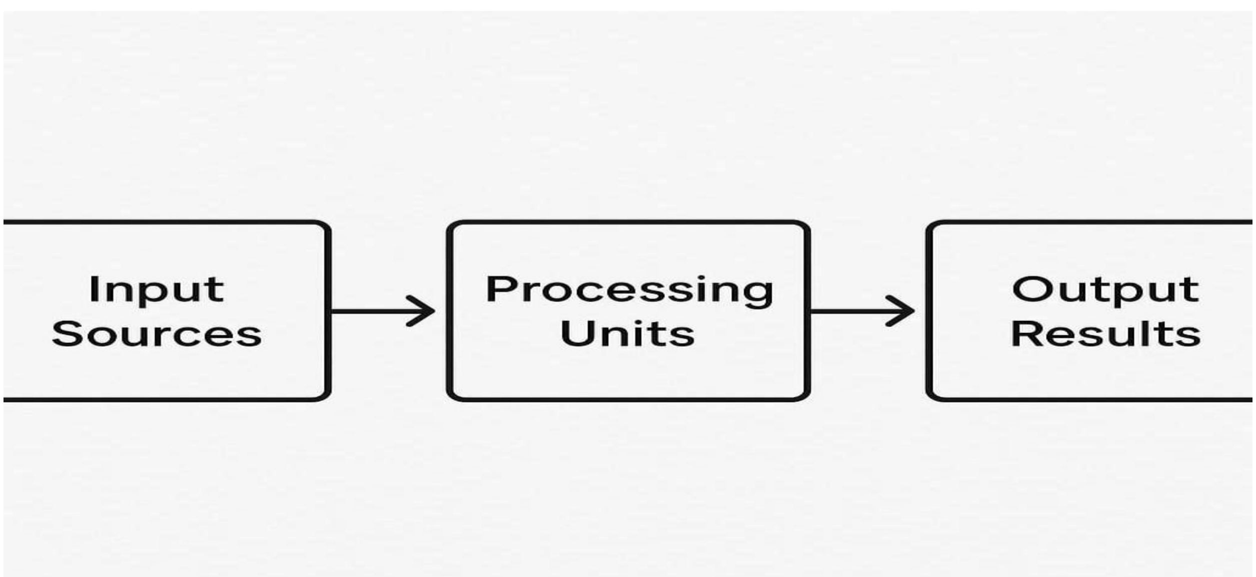
- Accuracy drops during extreme weather (insufficient training data)
- Data irregularities (missing values, outliers) may affect performance
- Real-time data not yet integrated

13. Future Enhancements

- Real-time data via sensors/CCTV feeds
- Mobile/Web app for users
- Deep learning models like LSTM
- Route planning integration

Additional Diagrams:

- **Block Diagram:**



- **System Flowchart:**