Project Documentation: TrafficTelligence

---

1. Introduction

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

---

2. Project Overview

- **Purpose**: The project aims to develop an intelligent system that predicts traffic volume based on weather, time, and holiday conditions using machine learning. This supports better traffic planning and management.
- **Features**:
  - Real-time traffic volume prediction
  - User-friendly web interface
  - Input form for temperature, weather, hour, holiday
  - Intelligent backend model using Random Forest Regressor
  - Visual indication of predicted traffic intensity

---

3. Architecture

- **Frontend**: HTML and CSS-based web pages for user input and displaying output.
- **Backend**: Flask (Python) application with trained ML model served via API.
- **Model**: Random Forest Regressor trained on traffic data from traffic_volume.csv.
- **Data Flow**: User → Form Input → Preprocessing → ML Model → Output Prediction → Display

---

4. Setup Instructions

- **Prerequisites**:
  - Python 3.8+
  - pip (Python package manager)
  - Flask

  - Required libraries: pandas, numpy, scikit-learn, matplotlib, seaborn, joblib

- **Installation**:

1. Clone the repository: git clone https://github.com/yourusername/traffictelligence.git

2. Navigate to the project folder: cd TrafficTelligence

3. Create and activate virtual environment:

    - Windows: python -m venv venv && venv\Scripts\activate

    - Linux/Mac: python3 -m venv venv && source venv/bin/activate

4. Install dependencies: pip install -r Requirements.txt

5. Run the Flask app: python app.py

6. Access the app locally at: http://127.0.0.1:5000

---

5. Folder Structure

- /templates: HTML templates (home, result pages)
- /static: CSS files
- app.py: Main Flask application
- model.pkl, preprocessor.pkl: ML model and preprocessing pipeline
- traffic_volume.ipynb: Jupyter notebook for model training and evaluation
- traffic volume.csv: Dataset used
- Requirements.txt: Python dependencies

---

6. Running the Application

- Activate virtual environment
- Start backend Flask server: python app.py
- Open browser and navigate to http://127.0.0.1:5000
- Enter weather and time data to receive traffic prediction

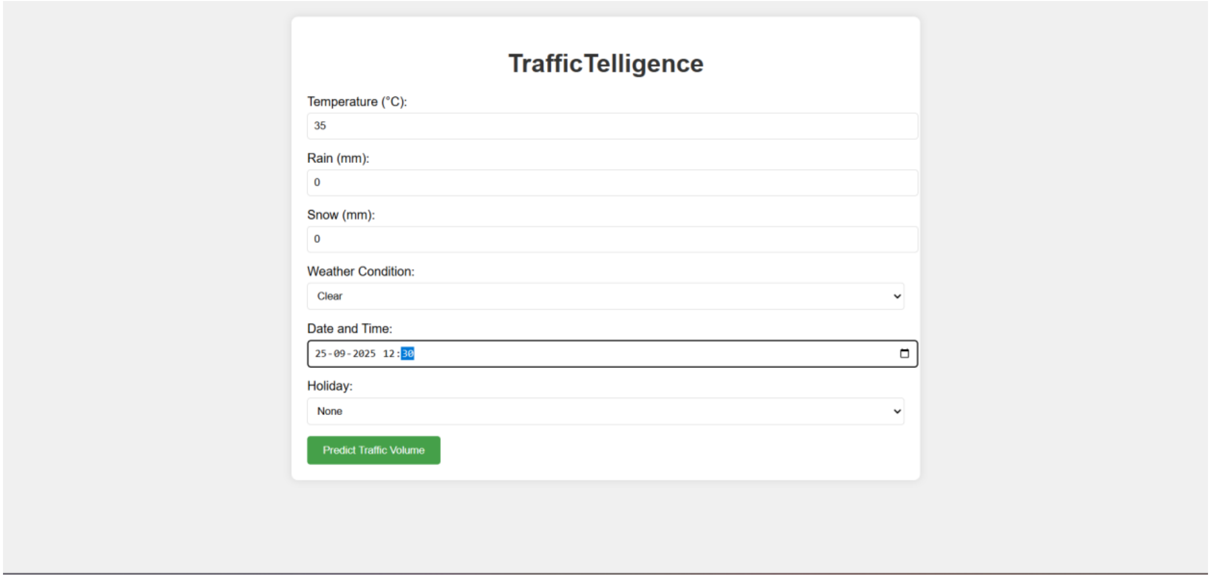
---

10. Testing

- **Testing Tools**: Jupyter Notebook for model evaluation, Flask test client for API testing
- **Testing Strategy**:
  - Validated model with 80-20 split
  - Metrics: R^2 Score, MAE, RMSE
  - Manual form submission tests in the web app interface

---

11. Screenshots or Demo



TrafficTelligence

Temperature (°C): 35

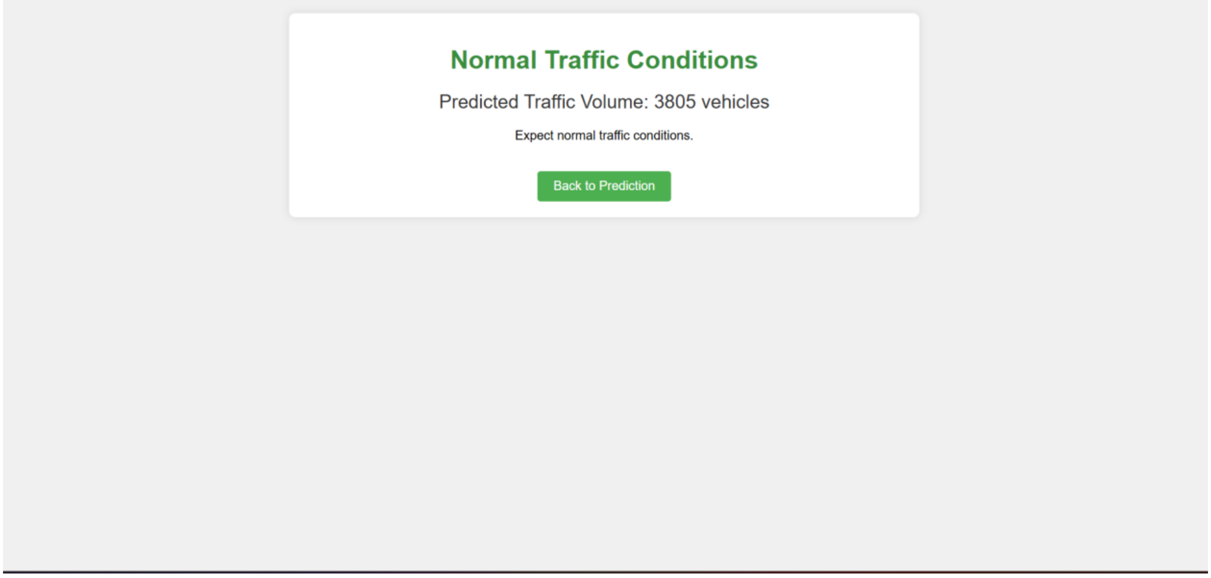
Rain (mm): 0

Snow (mm): 0

Weather Condition: Clear

Date and Time: 25-09-2025 12:30

Holiday: None



Normal Traffic Conditions

Predicted Traffic Volume: 3805 vehicles

Expect normal traffic conditions.

12. Known Issues

- Does not support live weather or time-based updates yet
- No authentication implemented
- UI is basic and can be improved for better UX

13. Future Enhancements

- Integrate real-time traffic and weather APIs
- Enhance UI with modern frontend frameworks (React/Angular)
- Store user queries and usage stats with MongoDB
- Add login and user profile features
- Extend prediction using deep learning models like LSTM or GRU