AI&ML Project Documentation

Introduction

Project Title:

TrafficTelligence: Advanced Traffic Volume Estimation with Machine Learning

Team Members:

• G Pranavi – Data Collection and Cleaning
• S Sarfaraz– Model Design and Training
• V Sharath– Evaluation and Optimization
• A Ramya Sree– Deployment and Integration

Project Overview

Purpose:

The purpose of this project is to develop a machine learning-based traffic volume estimation system to enhance smart city planning and commuter experiences. The system leverages historical traffic data, weather conditions, and external event data to make accurate traffic predictions.

Goals:

- Estimate real-time and future traffic volumes.
- Aid in traffic congestion reduction through intelligent traffic systems.
- Support urban planners and commuters with actionable insights.

Key Features:

- Accurate traffic volume prediction using ML algorithms.
- Real-time and historical data integration.
- Visualization of traffic patterns.
- Modular architecture for integration with traffic systems or navigation apps.

Architecture

Frontend:

Framework Used: Streamlit / React

Functionality:
- Display traffic volume predictions and trends.
- Map visualization of high/low congestion areas.
- User interface for selecting routes or areas of interest.

Backend:

Technology Used: Python with Flask or FastAPI

Responsibilities:
- Handle data input and preprocessing.
- Run trained machine learning models for prediction.
- Send results to frontend with prediction metrics.
- API endpoints for real-time estimation services.

Model Integration:

Models like Random Forest, XGBoost, or LSTM are trained and loaded for runtime inference.

Setup Instructions

Prerequisites:

Python 3.x
Libraries: pandas, numpy, scikit-learn, xgboost, flask, matplotlib, streamlit

STEP 1: Create Virtual Environment

```
conda create -n traffic_ml python=3.9
conda activate traffic_ml
```

STEP 2: Install Required Packages

```
pip install pandas numpy scikit-learn xgboost flask streamlit matplotlib
```

STEP 3: Run the Web Application

```
cd /d D:\TrafficTelligence\
python app.py
```

Open browser: http://127.0.0.1:5000/

How to Use

Steps:

1. Launch the application.
2. Select your route, region, or timeframe.
3. View predicted traffic volume and suggested travel times.
4. Receive suggestions for alternate routes.

Offline Usage

Details:

App runs locally with pre-downloaded model files.
No external data dependencies required at runtime.
Uses local datasets (CSV) for demonstrations.

Folder Structure

Structure:

```
TrafficTelligence/
├── app.py
├── model/
│   └── traffic_model.pkl
├── data/
│   ├── historical_traffic.csv
│   └── weather_events.csv
├── static/
├── templates/
├── utils/
│   └── preprocess.py
├── README.txt
```

Authentication

Details:

Optional: Add session-based or token-based user authentication in future.

User Interface

Features:

Dropdown menus for selecting location/time.
Line charts and bar graphs for volume trends.
Map overlays for congestion hotspots.

Testing

Methods:

Unit tests using PyTest for backend logic.
Real-world validation with test dataset.
Accuracy, MAE, RMSE used for evaluation.

Known Issues

Challenges:

Prediction accuracy may drop in case of sudden weather or event changes.
Limited generalization to unseen regions without retraining.

Future Enhancements

Plans:

Integrate live traffic APIs (e.g., Google Maps, TomTom).
Cloud deployment with RESTful APIs.
Incorporate deep learning models for spatiotemporal accuracy.
Add commuter mobile app interface with alert system.