TrafficTelligence Advanced Traffic Volume Es ma on with Machine Learning

1.Introduction

Project Title: TrafficTelligence Advanced Traffic Volume Es ma on with Machine Learning

Team Members:

1. S.Amarnath
2. A. Madhava datta
3. A. Harshita
4. M. Akash

2.Project Overview

2.1 Purpose: This project aims to develop a machine learning-based predic ve model to es mate traffic volume using historical traffic sensor data and real- me features like weather, me, and road type. It applies advanced algorithms and hyperparameter tuning to achieve accurate volume predic ons. The best-performing model is deployed via Flask, making it suitable for real- me integra on into smart city traffic systems.

2.23 Features: TrafficTelligence is a robust machine learning solu on that predicts traffic flow based on various contextual and sensor data. The project includes data preprocessing (handling missing values, encoding categorical variables, and normalizing data), training on regression algorithms like Linear Regression, Random Forest, Gradient Boos ng, and XGBoost. Evalua on metrics such as MAE, MSE, RMSE, and R2 score guide model selec on. The final model is deployed using a Flask web app for live input and predic on, contribu ng to be er urban traffic planning and conges on management.

3.Architecture

3.1 Frontend: Built with HTML and CSS, the interface allows users to input features like me of day, weather condi ons, and road details. Upon submission, this data is sent to the backend, and the predicted traffic volume is displayed for user interpreta on.

3.2 Backend: Implemented with Flask, the backend processes the frontend data, applies preprocessing, and feeds it to the trained ML model. It returns predic ons for display and can store the results in a database if required.

3.3Database: MongoDB stores the user inputs and their respec ve predicted traffic volume. Each record includes input features, predicted output, and a mestamp. Flask integrates with MongoDB using PyMongo.

4.Setup Instructions

4.1 Prerequisites:

- Python 3.x
- Pandas
- NumPy
- Scikit-learn
- XGBoost
- Flask
- Flask-PyMongo
- Joblib or Pickle
- MongoDB 4.2 Installa on: pip install flask pandas numpy scikit-learn xgboost flaskpymongo git clone <your-repo-url> cd https://github.com/Amarnath50/TrafficTelligence-ML/tree/main
- Visit: h p://127.0.0.1:5000
- Demo link: https://drive.google.com/file/d/1Pypn4o9GCnUxAeXZnWBw5hwH_UGjO121/view?usp=sharing

5.Folder Structure

5.1 Client (Frontend): Built with HTML/CSS. A form allows user entry for features like day, hour, temperature, humidity, and weather. On submit, data is sent to the Flask API and results are shown on the same or redirected page.

5.2 Server (Backend): The Flask server processes the input, applies preprocessing, loads the model, and returns the result. It also logs predic ons to MongoDB if enabled.

6.API Documentation.

POST /predict

Request: applica on/json

```
{
  "hour": 17,
  "day_of_week": "Monday",
  "weather": "Clear",
  "temp": 24.5,
  ...
}
```

Response:

```
{
  "predic on": 352,
  "result": "Es mated traffic volume: 352 vehicles/hour"
}
```

2. GET / Response:

```
{
  "message": "Welcome to the Traffic Volume Es ma on API"
}
```

3. POST /store-data

Op onal for storing input and predic on in DB.

8.Authentication

No autheon added; the tool is open for general use. 9.User Interface

Clean, responsive interface for data input and instant traffic volume predic ons. Designed for planners, researchers, and analysts.

10 Testing

Data Valida on: Cleaned missing values, normalized features.

Model Evalua on: MAE, RMSE, R2 score for regression models.

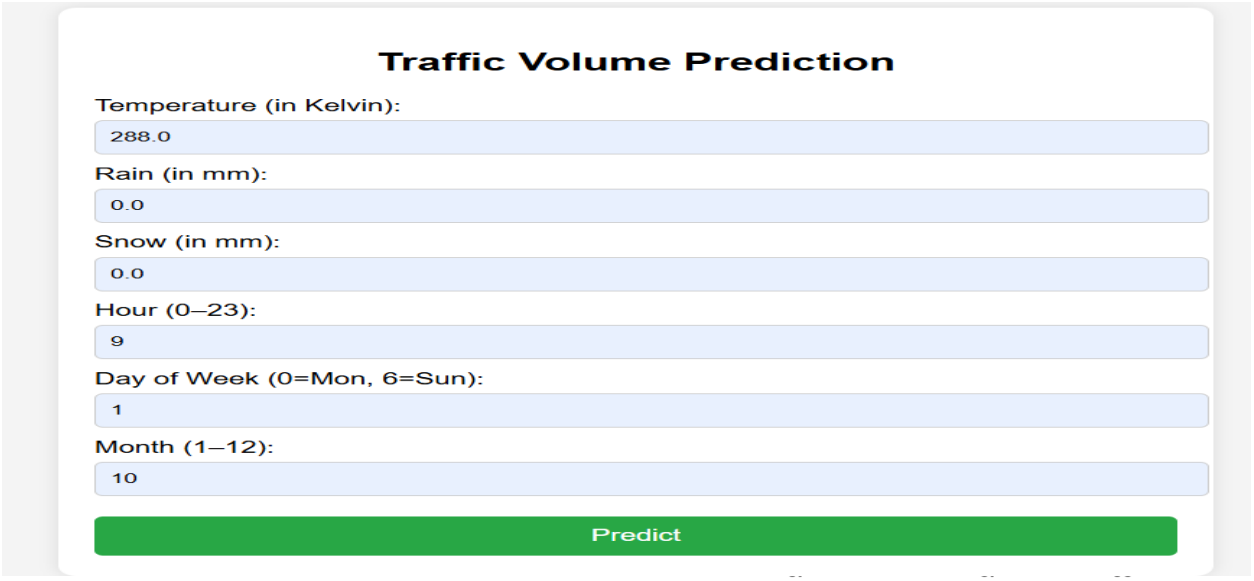
API Tes ng: Ensured endpoints func on with valid/invalid input.

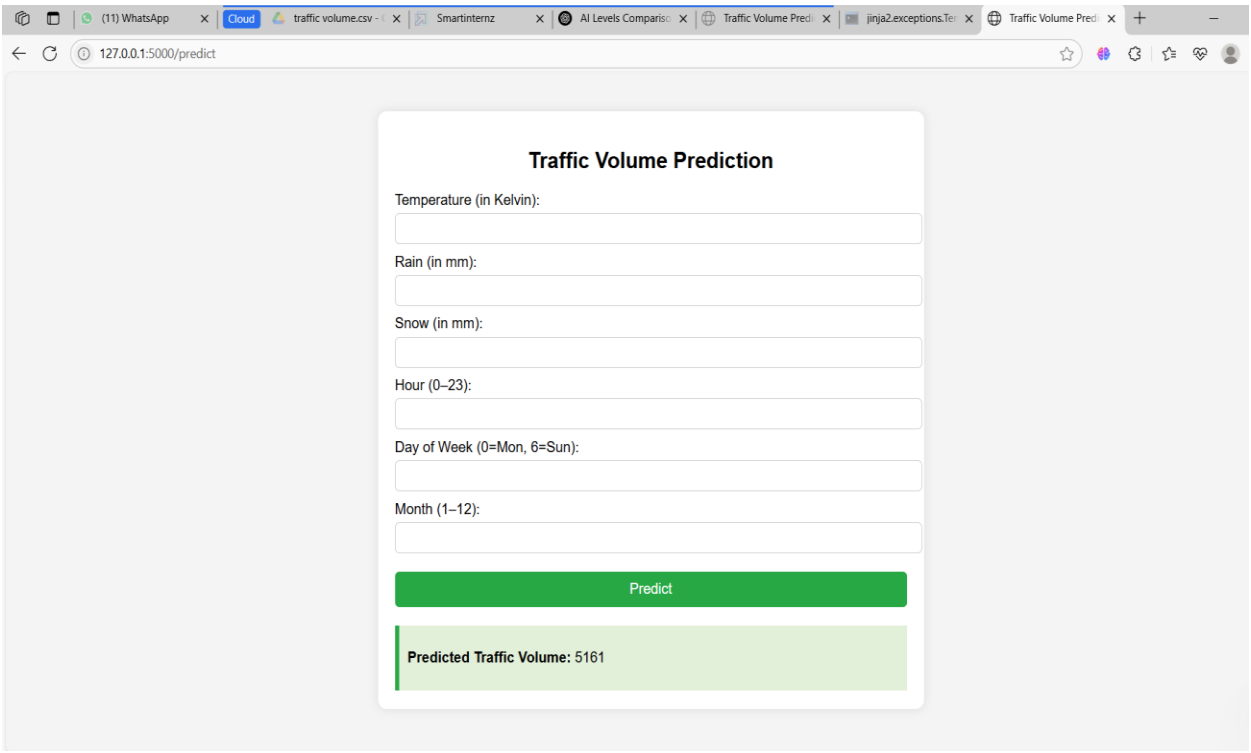
Integra on: Verified flow from form to predic on.

Edge Cases: Tested for unusual condi ons like weather or me.

Tools: scikit-learn, XGBoost, Pytest, Postman, Jupyter, DevTools.

11.Screenshots





12.Known Issues

1. Data Imbalance Low volume segments (e.g., night) affect predic ons.
2. Sensor Errors Raw data may have noise/missing values.
3. Deployment Limita ons Flask isn't ideal for large-scale use.
4. Input Valida on Minimal frontend checks.
5. No Role-Based Access Public tool.

13.Future Enhancements

SHAP/LIME for Explainability

Mobile UI Op miza on

Role-based Access (Doctor/Admin/Researcher)

Real- me Dashboard for Predic ons

Cloud Deployment with Docker + AWS/Heroku