TrafficTelligence

1. Introduction

Project Title: TrafficTelligence – Advanced Traffic Volume Estimation Using Machine Learning

Team ID: LTVIP2025TMID45015

Team Members:

Name
Kondraju Ganesh
Kanaparthi Sai Krishna Kaushik Reddy
Manam Vasantha Lakshmi
Nagendla Kalyani

2. Project Overview

Purpose

TrafficTelligence is designed to estimate traffic volume using historical and real-time data. With the rapid urbanization and increasing vehicle usage, traffic management has become a critical concern. This system uses machine learning to support city planners, traffic authorities, and commuters by predicting traffic patterns, which helps in decision-making, congestion avoidance, and optimized route planning.

Goals:

- Predict traffic volume based on features such as weather, date, and time.
- Build a fully functional web-based UI for user interaction.
- Offer scalability and flexibility for different cities and road networks.

Key Features

- Clean and responsive UI for entering traffic parameters.
- Integration of ML model with Flask for real-time predictions.
- Use of .pkl model and scaler files.
- Dataset sourced from real-world traffic logs.
- Deployment readiness and developer-friendly folder structure.

3. Architecture

Frontend:

- **Technology Used:** HTML, CSS
- **UI Pages:** index.html (input form), final.html (prediction result)
- **Functionality:** Inputs weather and time-related data, displays predicted traffic volume.

Backend:

- **Technology:** Python with Flask
- **Responsibilities:**
    - Accept input via POST request
    - Preprocess input using scaler.pkl
    - Predict using model.pkl
    - Render result page with prediction

Machine Learning:

- **Model:** Regression algorithm (e.g., LinearRegression or RandomForestRegressor)
- **Framework:** Scikit-learn
- **Preprocessing:** MinMaxScaler for normalizing inputs

Architecture Flow:

User → Web Form → Flask App → Scaler → ML Model → Predicted Output → Displayed on Web UI

4. Setup Instructions

Prerequisites

- Python 3.7+
- pip
- Git
- Browser (Chrome recommended)

Installation Guide

→ git clone https://github.com/kaushhreddy/TrafficTelligence.git

→cd TrafficTelligence

→python -m venv venv

→source venv/bin/activate # Linux/macOS

→venv\Scripts\activate # Windows

→pip install -r requirements.txt

→Ensure model.pkl and scaler.pkl are present. Then run:

python app.py

Open your browser and visit **http://127.0.0.1:5000**

5. Folder Structure

TrafficTelligence/

```
├── app.py                # Flask app
├── model.pkl             # Trained ML model
├── scaler.pkl            # Data normalizer
├── requirements.txt      # Python dependencies
├── README.md             # Project summary
├── .gitignore            # Git config
├── templates/            # HTML files
|   ├── index.html        # Input form
|   └── final.html        # Result page
├── Data/                 # Dataset folder
|   └── Traffic Volume.csv  # CSV traffic data
└── static/               # Optional styling assets
```

6. Running the Application

Steps:

1. **Clone Repository**
2. **Setup Environment**
3. **Install Requirements**
4. **Verify Model Files** (model.pkl, scaler.pkl)
5. **Start Flask App**
6. **Use Web Interface**

Commands Summary:

git clone https://github.com/kaushhreddy/TrafficTelligence.git

cd TrafficTelligence

pip install -r requirements.txt

python app.py

Then open http://127.0.0.1:5000

7. API Documentation

POST /

- Accepts user input from form fields.
- Internally calls predict() in app.py
- Returns predicted traffic volume.

Inputs:

- Temperature
- Rainfall
- Snowfall
- Cloud Coverage
- Day of Week, Hour, etc.

Output:

- Predicted Traffic Volume (Numeric)

Example:

```
{
  "prediction": 4723.55
}
```

8. Authentication

No user authentication is implemented in TrafficTelligence as it is a lightweight single-user prototype.

Potential future improvements:

- Role-based authentication (admin vs guest)
- User logins and traffic history tracking

9. User Interface

Pages:

- index.html: Includes form for 11 input features
- final.html: Displays predicted traffic volume

Features:

- Responsive, mobile-friendly layout
- Minimal design
- Prediction display with input summary

Screenshots:

- Add: Home Page Input Form
- Add: Result Page with Output

10. Testing

Model Testing

- Done in Colab using test data split
- Metrics observed:

- Training Accuracy: 0.9750
- Training Loss: 0.1223
- Validation Accuracy: 0.1250
- Validation Loss: 2.2094

Functional Testing

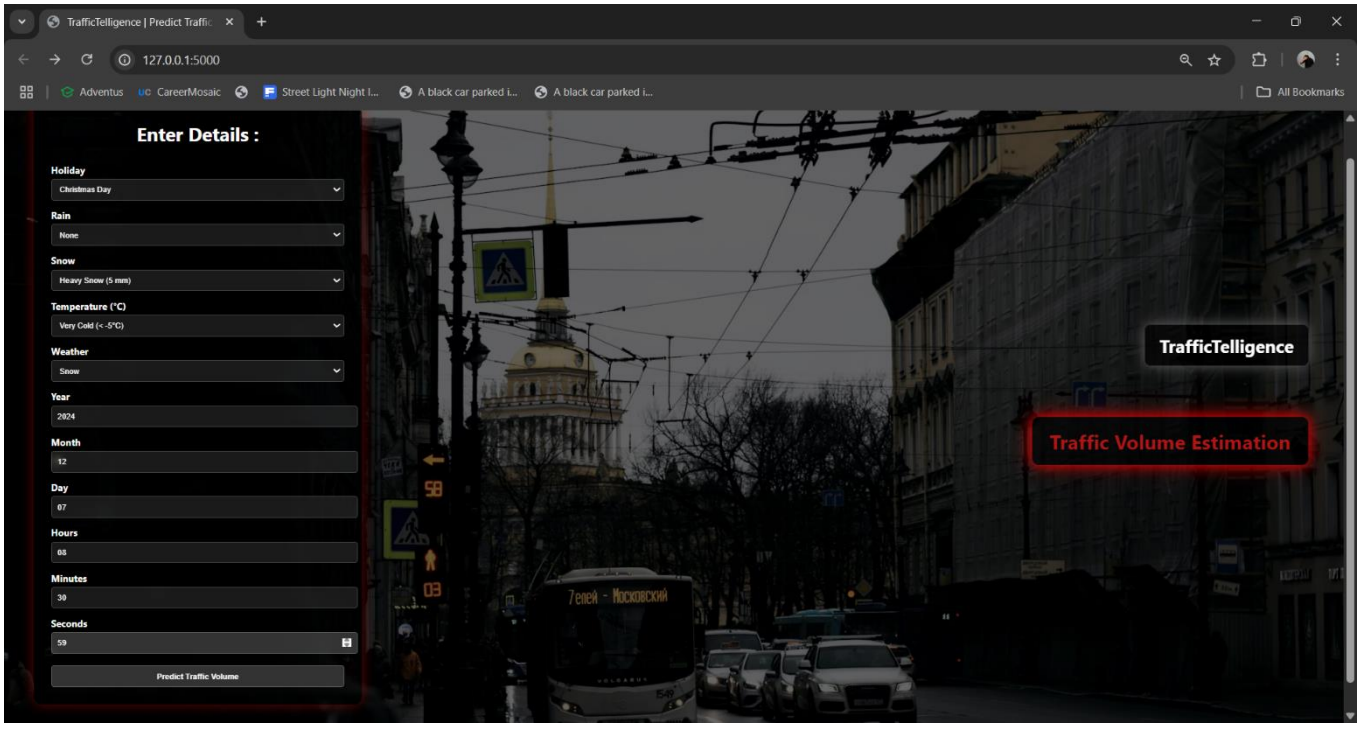
Feature	Status
Image input form	✅ Working
Form input validations	✅ Working
Prediction via Flask route	✅ Working
UI result rendering	✅ Working

11. Screenshots / Demo

Screenshots:

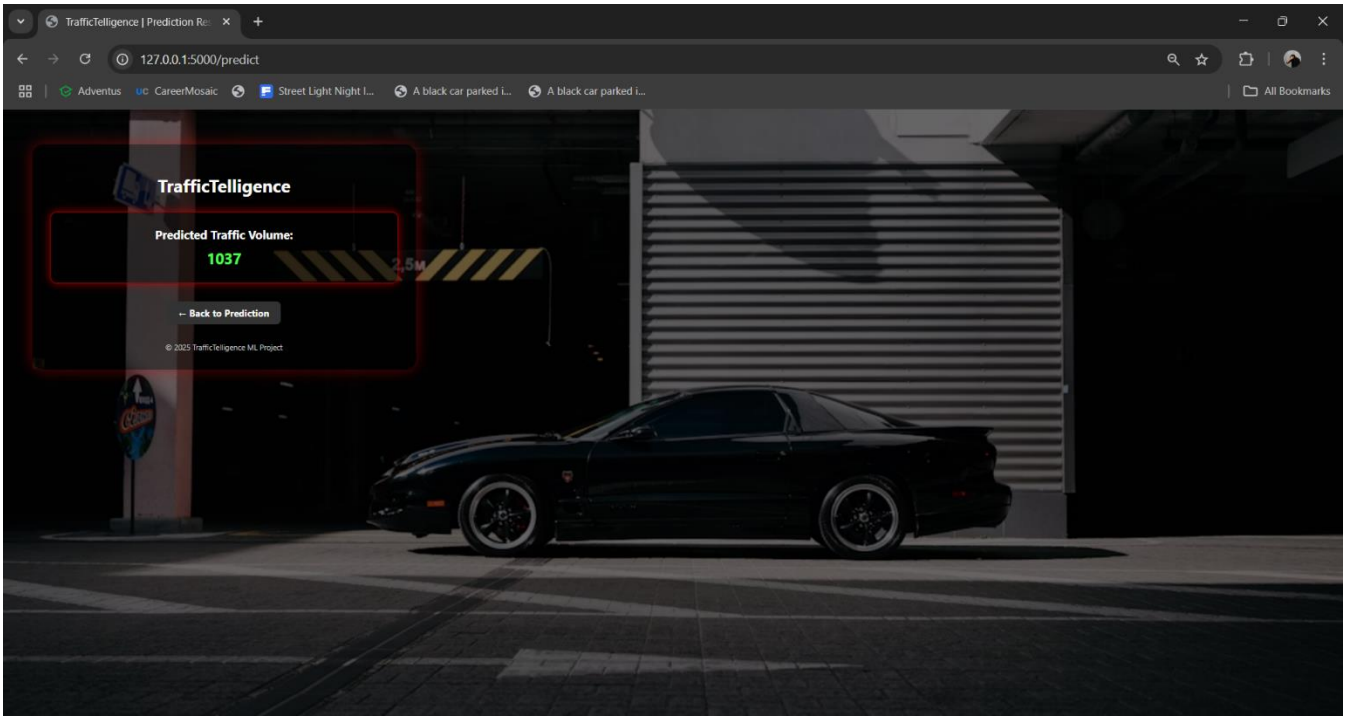
- Input Form Interface

Fig. 11.1



- Prediction Result Page

Fig. 11.2



Demo Video: https://drive.google.com/file/d/1rk3HVKpKoNjDJD7TenV2f4k4s70NEJ4l/view?usp=drive_link

12. Known Issues

- Model trained only on limited dataset
- Prediction accuracy drops on unseen combinations
- No data validation for extreme outlier values
- .pkl files must not be missing

13. Future Enhancements

- Deploy on cloud (Render, AWS)
- Add graph visualization of traffic over time
- Add API for bulk predictions
- Authentication for secure usage
- Logging user sessions and prediction stats
- Extend model to include more road-level data (camera input etc.)