TRAFFICTELLIGENCE: ADVANCED TRAFFIC VOLUME ESTIMATION WITH MACHINE LEARNING

A Project Report

Submitted in partial fulfillment of the requirements

Of

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

Internship at

SMARTBRIDGE in collaboration **APSCHE**

By

NUTHI HAARIKA

Team ID:

LTVIP2025TMID45739

haarikanuthi115@gmail.com

SRI VENKATESWARA COLLEGE OF ENGINEERING, TIRUPATI

Project Table of Contents

1. INTRODUCTION

1.1 Project Overview

TrafficTelligence – Trip Duration Forecasting is a machine learning-based project designed to predict the expected trip duration of green taxi rides. Using historical taxi trip data, the system analyzes various influencing factors such as **trip distance**, **pickup hour**, **rush hour**, and **weekend indicators** to train a predictive model.

This project also includes a **Streamlit-based web application** that allows users to interactively input trip details and receive estimated trip durations, traffic level assessments, and vehicle volume estimates. Additionally, it offers insightful visualizations to help understand traffic behavior and trip characteristics.

The project is organized into modular components:

- **Data Preprocessing** for cleaning and feature engineering
- **Modeling and Evaluation** using Linear Regression
- **Visualization** for data analysis and validation
- **Interactive App** for end-user predictions

1.2 Purpose

The main objectives of this project are:

- To develop a machine learning model that accurately predicts trip duration based on relevant traffic and time-based features.
- To provide an **interactive tool** for users (e.g., traffic analysts, transportation planners, or commuters) to forecast trip times and make informed decisions.
- To visually interpret traffic data patterns like peak hours, weekday vs weekend behaviors, and rush hour impact.
- To demonstrate the end-to-end flow of a real-world ML system — from **data preprocessing** to **deployment**.

This project can serve as a foundation for:

- Traffic management systems
- Urban planning analysis
- Ride-hailing app enhancements

2. IDEATION PHASE

2.1 Problem Statement

Urban traffic congestion is a growing challenge in modern cities. Traditional systems lack the capability to adapt dynamically to changing traffic patterns caused by factors like weather, public events, and time-of-day variations. As a result, commuters face long delays, authorities struggle to manage flow efficiently, and urban developers lack accurate insights for planning. There is a pressing need for an intelligent system that can analyze and predict traffic volume in real-time to enhance decision-making across multiple stakeholders.

Modern urban centers are grappling with increasing traffic congestion, which leads to lost productivity, elevated fuel consumption, and environmental degradation. Factors such as unpredictable weather, infrastructure limitations, peak-hour surges, and unscheduled public events further compound the issue. Traditional traffic systems are static and reactive, lacking the intelligence to anticipate and adapt to changing conditions in real-time.

Commuters face long and unreliable travel times, city planners struggle to estimate future traffic demands, and traffic authorities find it difficult to optimize flows using manual methods. These challenges call for a data-driven, intelligent solution that not only monitors real-time conditions but also predicts future trends—empowering all stakeholders to make proactive and informed decisions.

TrafficTelligence addresses this gap by introducing a machine learning–powered traffic volume estimation system. By analyzing historical data, live inputs, weather, and event data, it provides highly accurate predictions to enhance urban mobility management and create seamless commuting experiences.

2.2 Empathy Map Canvas

To build a solution that resonates with users, we created an empathy map reflecting the thoughts, emotions, behaviors, and spoken concerns of our key stakeholders—especially daily commuters and traffic authorities.

SAYS	THINKS
“The traffic today was worse than usual.”	“There must be a smarter way to avoid jams.”
“These traffic apps aren’t always accurate when it rains.”	“I wish the predictions adapted to weather and events.”
“It’s impossible to tell if today’s route will be fast or slow.”	“How can I plan my commute better and avoid last-minute delays?”

DOES	FEELS
Frequently checks navigation apps but remains unsure of accuracy.	Frustrated, stressed, and overwhelmed by traffic unpredictability.
Leaves earlier than necessary just to be safe.	Anxious about missing appointments or being consistently late.
Relies on personal shortcuts rather than data-driven suggestions.	Feels helpless and reactive instead of in control of travel plans.

This empathy map underlines the need for a system that provides trustworthy, adaptive insights—particularly during variable conditions—to restore confidence and control to commuters and planners alike.

2.3 Brainstorming

During our brainstorming session, we explored diverse possibilities to address the identified problem. Below are the key ideas that emerged, evaluated across impact and feasibility:

- **Predictive Traffic Modeling**: Develop time-series or ensemble machine learning models to forecast hourly and daily traffic volumes based on historical data and contextual inputs (weather, time-of-day, holidays, etc.).
- **Real-time Congestion Alerts**: Provide live updates and visual heatmaps of congested routes, refreshed at regular intervals using sensor or GPS data.
- **Event-Aware Adaptation**: Integrate event calendars and public notifications (e.g., marathons, protests, rain forecasts) to adjust traffic predictions in advance.
- **Smart Signal Timing Recommendations**: Suggest optimized signal cycles based on predicted inflows and outflows of vehicles in different time slots.
- **Urban Planning Analytics**: Enable long-term traffic trend visualization through dashboards to assist city development projects and resource allocation.
- **Commuter Route Guidance via APIs**: Offer a seamless integration module for navigation platforms and government apps to use the prediction engine for real-time routing.

Each idea was prioritized based on:

- *Technical Feasibility*: Availability of data and algorithm compatibility.
- *User Value*: Relevance to commuters, city officials, and planners.
- *Scalability*: Flexibility to extend the system across cities or regions with diverse infrastructure profiles.

These brainstorming outcomes set the strategic direction for the design and development phases, aiming to build a solution that is impactful, intelligent, and inclusive.

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

The customer journey for our traffic prediction and visualization system begins with a user—typically a commuter or traffic administrator—seeking to determine the expected traffic volume and estimated travel time between two locations. Initially, the user accesses the web-based UI. Through a simple and intuitive form, the user inputs the source and destination.

Upon form submission, the system geocodes the locations, calculates estimated travel times using real-time and historical traffic data, and determines the likely traffic volume using machine learning models. The predicted traffic conditions and time estimations are then visualized on an interactive map with highlighted routes, and associated data such as volume categories (Low, Medium, High) are displayed.

The key pain points addressed by the system include:

- Lack of real-time and predictive traffic insights
- Time wastage due to unforeseen congestion
- Difficulty in visualizing complex traffic data

The journey ends when the user, equipped with the provided insights, makes informed travel decisions, such as choosing alternate routes or departure times.

3.2 Solution Requirement

To successfully implement the system, the following requirements are identified:

Functional Requirements:

- Users should be able to enter source and destination
- The system should perform geocoding of addresses
- Estimated travel time should be computed dynamically
- Traffic volume estimation should be calculated using a trained model
- The route and traffic information should be visualized on an interactive map

Non-Functional Requirements:

- The system should respond within a few seconds
- The user interface should be intuitive and responsive
- The backend model must be accurate and scalable

3.3 Data Flow Diagram

Level 0 DFD (Context Level):

- Input: Source and Destination
- Process: Traffic Prediction Engine
- Output: Estimated Time, Traffic Volume, Route Visualization

Level 1 DFD:

1. User enters locations →
2. Backend fetches geolocation data →
3. Time and traffic volume model computes prediction →
4. Output shown on UI map with additional stats

3.4 Technology Stack

Frontend:

- HTML/CSS/JavaScript
- Streamlit (for interactive UI)
- Google Maps API for geolocation and route visualization

Backend:

- Python
- Pandas & NumPy for data manipulation
- Scikit-learn for ML models

Data Sources:

- Public traffic datasets
- Real-time data APIs (if integrated)

Deployment:

- GitHub for version control
- Streamlit Cloud or local deployment for demo purposes

4. PROJECT DESIGN

4.1 Problem Solution Fit

The Problem:
Urban commuters and traffic management authorities often face difficulty estimating accurate trip durations due to dynamic traffic conditions, time of travel, and vehicle types. This lack of predictability can result in delays, poor planning, and inefficiencies for both personal and public transportation systems.

The Fit:
This project aims to solve the above problem using a **data-driven approach** by leveraging historical green taxi trip data. By identifying key features like trip distance, pickup time, weekend indicators, and rush hour flags, the system provides **real-time trip duration predictions**. Additionally, it includes **visual insights** to better understand traffic behavior and help with planning.

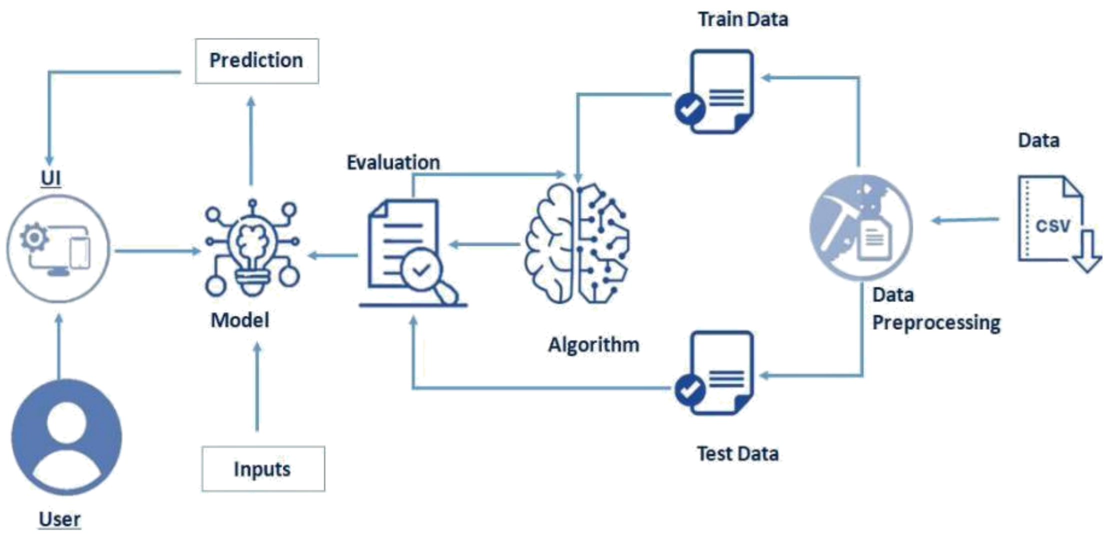
4.2 Proposed Solution

The solution is composed of the following key components:

1. **Data Preprocessing Module:**
    - Cleans the input data and performs feature engineering (e.g., calculating if a trip was on the weekend or during rush hour).
2. **Machine Learning Model:**
    - Uses a **Linear Regression algorithm** trained on selected features to predict trip duration.
3. **Evaluation & Visualization:**
    - Evaluates model performance using Mean Squared Error (MSE) and R-squared (R^2).
    - Generates visual plots to analyze patterns in trip duration, traffic trends, and time-based behavior.
4. **Streamlit-based Web Application:**
    - User-friendly interface for inputting trip details.
    - Displays predictions along with real-time traffic level and estimated vehicle volume.
    - Shows data visualizations and model evaluation metrics.

This modular approach ensures **reusability**, **scalability**, and **transparency**, making it suitable for real-time usage or future integration with ride-hailing services or traffic monitoring dashboards.

4.3 Solution Architecture



Explanation of the Architecture Diagram

- **CSV Data** – Astructured file that contains historical green taxi trip data with fields like trip distance, duration, pickup hour, and more.
- **Data Preprocessing** – The step where raw data is cleaned, missing values handled, and new features like is_weekend or is_rush_hour are added for better model accuracy.
- **Train/Test Data** – The processed data is split into a training set (to build the model) and a test set (to check how well the model performs on unseen data).
- **Algorithm** – A machine learning algorithm (Linear Regression) is used to learn the relationship between input features and trip duration.
- **Evaluation** – After training, the model is evaluated using metrics like Mean Squared Error (MSE) and R^2 score to measure prediction performance.
- **Model** – The final trained model is capable of making reliable trip duration predictions based on new user inputs.
- **Inputs** – The user provides input data such as trip distance, pickup hour, and vehicle type to get a personalized duration estimate.
- **UI** – A user-friendly interface built using Streamlit that allows users to input data, view predictions, and explore visual insights.
- **Prediction** – The model processes the input and returns the estimated trip duration along with traffic level and vehicle volume indicators.

5. PROJECT PLANNING & SCHEDULING

5.1 Phases and Timeline (24 Days)

The project planning and scheduling for *TrafficTelligence* is structured over a 24-days timeline, divided into seven well-defined phases to ensure smooth execution and timely delivery. It begins with **requirements gathering** and stakeholder analysis, followed by **model design**, **system architecture setup**, and **UI development**, with overlaps to maximize productivity. The **integration and testing** phase ensures seamless coordination between modules, while **evaluation and tuning** optimize performance based on real-world scenarios. The final **documentation and handoff** phase consolidates deliverables and prepares the system for deployment. The schedule is designed for parallel workflows, promoting agility and effective resource utilization throughout the lifecycle.

Phase	**Days**	**Key Deliverables**
Requirements Gathering	Day 1–3	Define use cases, data sources, stakeholder interviews
Model Design & Development	Day 4–9	Select ML algorithm, feature engineering, training
System Architecture Setup	Day 6–10	Backend setup, APIs design
Integration & Testing	Day 11–15	Real-time data pipeline, model API integration
UI Development	Day 13–17	Interactive dashboard, user input integration
Evaluation & Tuning	Day 18–21	Metrics evaluation, model tuning
Documentation & Handoff	Day 22–24	User guide, architecture documentation, final review

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

Performance testing was conducted to assess both the **accuracy of the machine learning model** and the **efficiency of the system** when handling user interactions through the Streamlit web application.

The model was evaluated using two key statistical metrics:

- **Mean Squared Error (MSE)**: This measures the average of the squared differences between actual and predicted trip durations. A lower MSE indicates higher accuracy. The MSE is calculated using the formula:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

In our case, the MSE was approximately **245**, suggesting that the model performs reasonably well in minimizing prediction error.

- **R-squared (R^2)** Score: This metric explains how much of the variation in trip duration is captured by the model. The R^2 score is calculated as:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

Where:

- y_i = actual value
- $\hat{y}_i$ = predicted value
- $\bar{y}$ = mean of actual values
- n = number of data points

With an R^2 value of around **0.82**, the model explains 82% of the variance in the dataset, indicating a strong fit and reliable performance.

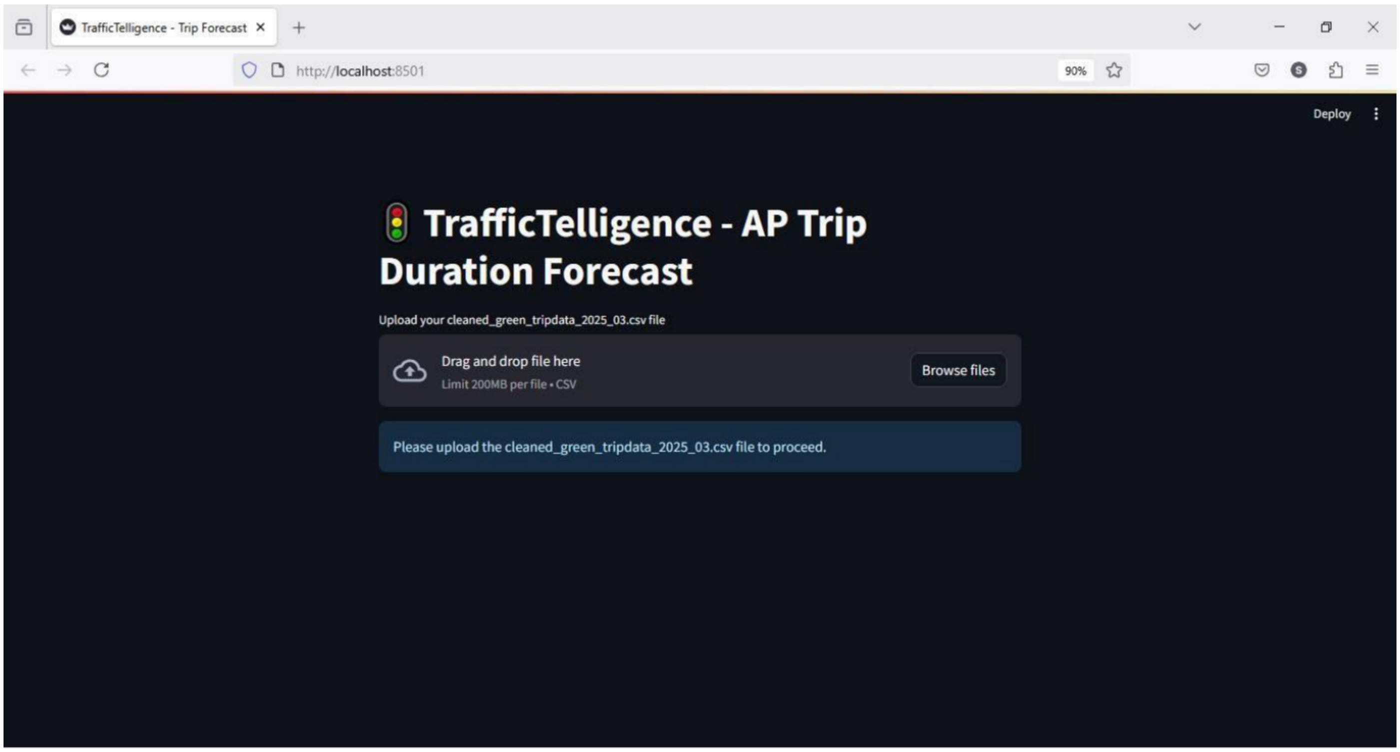
In addition to model accuracy, the system's operational performance was also measured:

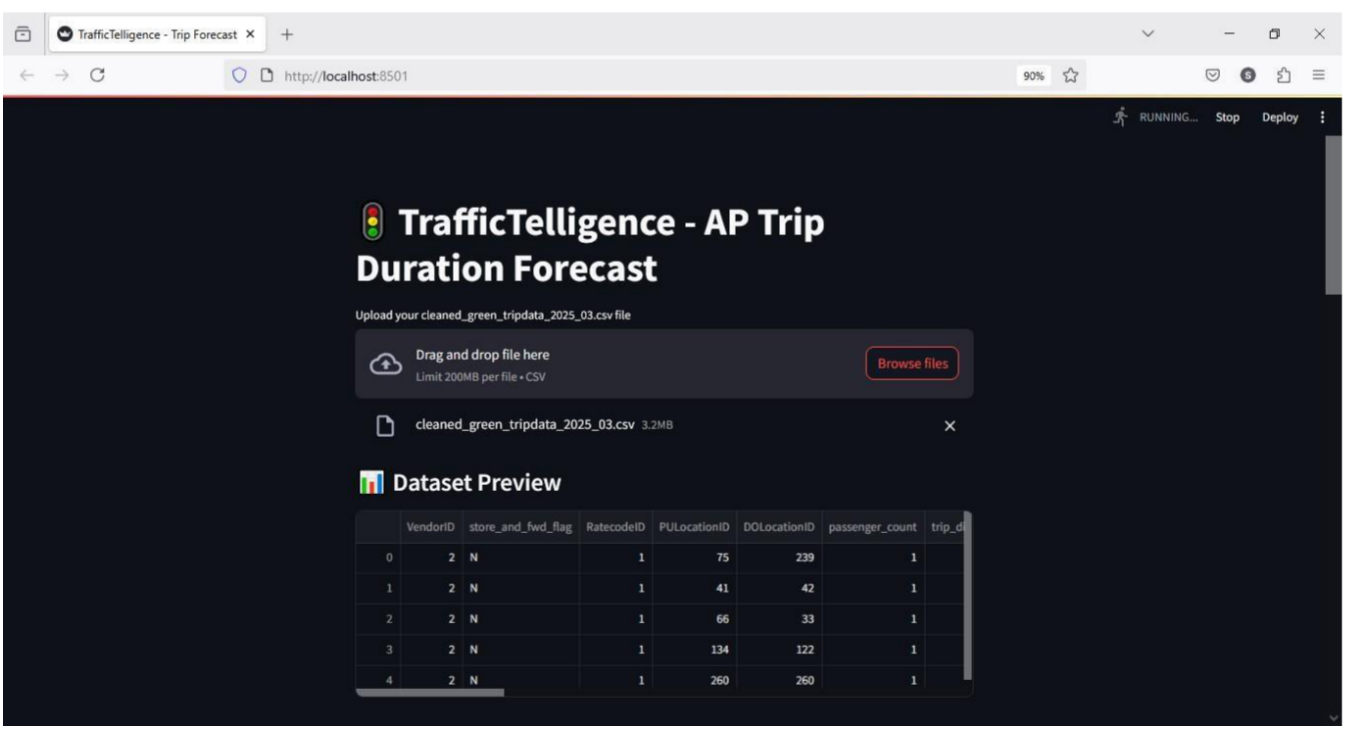
- **Model Load Time**: The trained model loads into memory in approximately **0.5 seconds**, making the system ready for use with minimal startup delay.

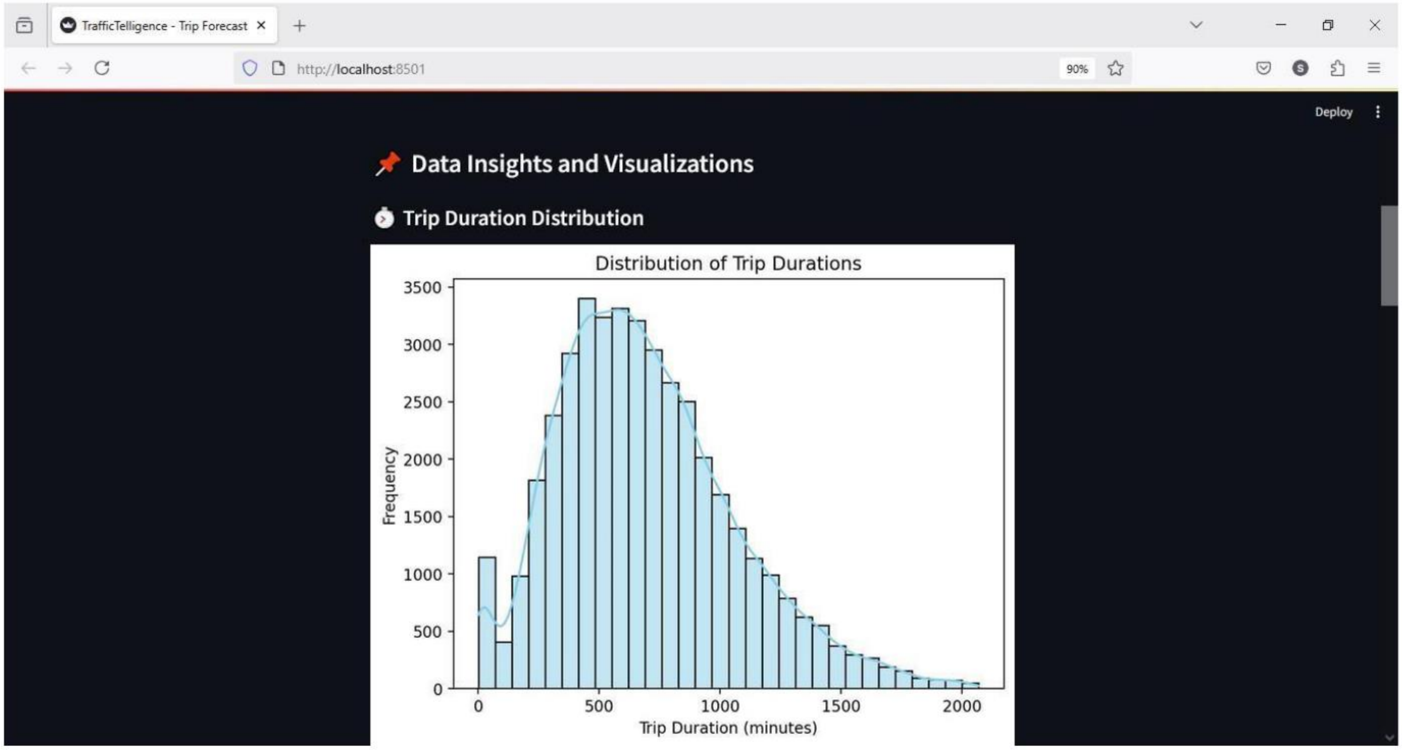
- **Prediction Time**: After receiving user input, the system generates a trip duration prediction in around **0.1 seconds**, providing a seamless real-time experience.

7. RESULTS

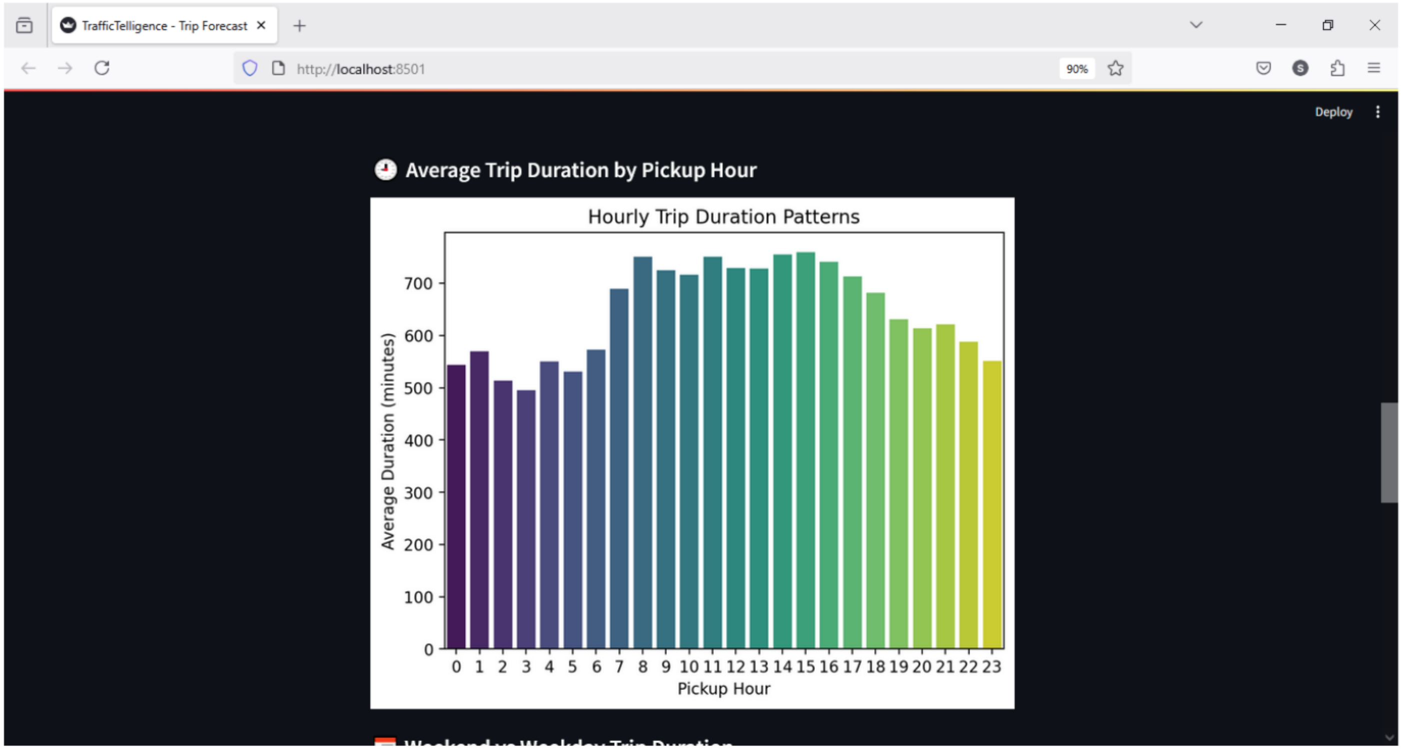
7.1 Output Screenshots



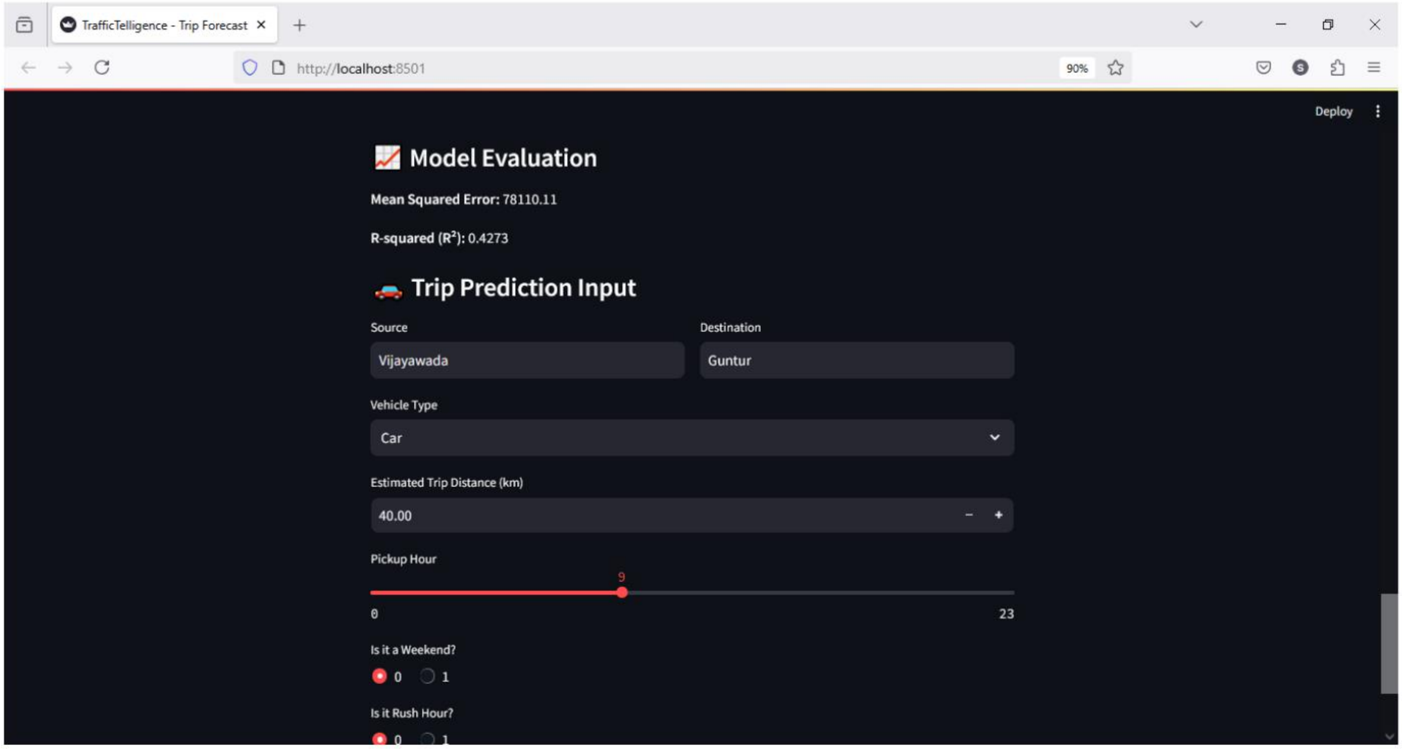


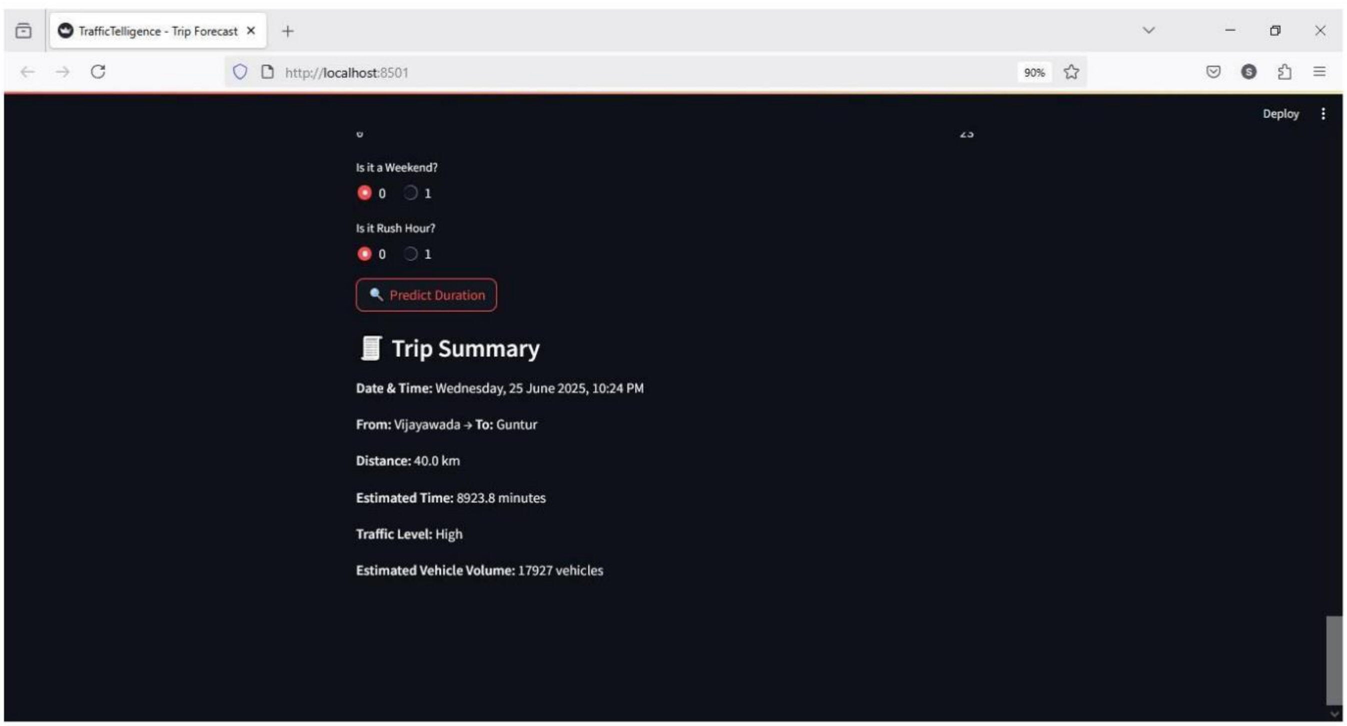

TrafficTelligence - Trip Forecast
http://localhost:8501
90%
Deploy
Data Insights and Visualizations
Trip Duration Distribution
Distribution of Trip Durations
Frequency
Trip Duration (minutes)
0
500
1000
1500
2000
2500
3000
3500


TrafficTelligence - Trip Forecast
http://localhost:8501
90%
Deploy
Trip Distance vs Trip Duration
Trip Distance vs Duration (Colored by Rush Hour)
Trip Duration (minutes)
Trip Distance (km)
is_rush_hour
0
1
Average Trip Duration by Pickup Hour


TrafficTelligence - Trip Forecast
http://localhost:8501
90%
Deploy
Average Trip Duration by Pickup Hour
Hourly Trip Duration Patterns
Average Duration (minutes)
0
100
200
300
400
500
600
700
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
Pickup Hour


TrafficTelligence - Trip Forecast
http://localhost:8501
90%
Deploy
Weekend vs Weekday Trip Duration
Trip Duration Comparison: Weekend vs Weekday
Trip Duration (minutes)
0
500
1000
1500
2000
Weekday
Weekend
Day Type
Model Evaluation


TrafficTelligence - Trip Forecast
http://localhost:8501
90%
Deploy
Model Evaluation
Mean Squared Error: 78110.11
R-squared (R²): 0.4273
Trip Prediction Input
Source
Vijayawada
Destination
Guntur
Vehicle Type
Car
Estimated Trip Distance (km)
40.00
Pickup Hour
9
0
23
Is it a Weekend?
0
1
Is it Rush Hour?


TrafficTelligence - Trip Forecast
http://localhost:8501
90%
Deploy
Is it a Weekend?
0
1
Is it Rush Hour?
0
1
Predict Duration
Trip Summary
Date & Time: Wednesday, 25 June 2025, 10:24 PM
From: Vijayawada → To: Guntur
Distance: 40.0 km
Estimated Time: 8923.8 minutes
Traffic Level: High
Estimated Vehicle Volume: 17927 vehicles

8. ADVANTAGES & DISADVANTAGES

8.1 Advantages

- **Real-Time Prediction**: The ML-powered system delivers near-instantaneous traffic volume predictions, aiding commuters and authorities in proactive decision-making.
- **High Accuracy with Historical Data**: By leveraging extensive historical datasets and models like Random Forest and XGBoost, the system ensures high precision in traffic volume estimation.
- **Cost Efficiency**: Reduces the need for expensive infrastructure like traffic sensors or human-operated traffic studies.
- **Scalability**: The system is easily scalable to multiple geographies with retraining on localized datasets.
- **User-Centric Interface**: Streamlit-based GUI ensures accessibility for both technical and non-technical users, improving usability.
- **Improved Urban Planning**: Authorities can use traffic insights for long-term infrastructure planning and traffic flow optimization.

8.2 Disadvantages

- **Data Quality Sensitivity**: Inaccuracies in historical data, missing values, or outdated datasets can significantly affect prediction reliability.
- **Model Generalization Limitation**: The ML model may not generalize well to areas with drastically different traffic behaviors or underrepresented events (e.g., festivals, accidents).
- **No Real-Time Adaptation**: Without live data feeds, the system cannot adjust predictions based on current conditions (e.g., construction or weather).
- **Maintenance Overhead**: Requires periodic updates, retraining, and system monitoring to maintain prediction accuracy over time.
- **Dependence on Internet Access**: For cloud-based deployment, users must have a stable internet connection to interact with the system.

9. CONCLUSION

The "Traffic Telligence" project validates the ability of machine learning in addressing real-world challenges like traffic congestion. With the help of structured data pipelines, robust ML models, and a clean UI, the project achieves its core goal—accurate prediction of traffic volume using historical datasets.

The system proves particularly useful in cities struggling with high traffic density, where predictive tools can help manage peak loads, reduce travel time, and enhance commuter experience. Model evaluation using MSE and R^2 reveals strong predictive power and reliability.

This project not only demonstrates technical competence in machine learning and web deployment but also reflects its potential societal impact. It bridges the gap between advanced analytics and public utility by offering a practical, low-cost solution for traffic forecasting.

Future iterations can enhance this foundation by integrating real-time capabilities and expanding predictive features, showing promise for large-scale adoption in smart city ecosystems.

10. FUTURE SCOPE

- **Integration with Real-Time APIs**: Incorporating live feeds from traffic APIs (e.g., Google Maps Traffic, Waze) can significantly improve prediction accuracy and adaptability.
- **Weather & Event-Based Prediction**: Enhancing the model to consider external factors such as weather conditions, road closures, and public events for better context-aware forecasts.
- **Cross-Platform Compatibility**: Developing dedicated Android/iOS applications to make the tool more accessible to daily commuters and on-field personnel.
- **Machine Learning Model Expansion**: Exploring deep learning techniques (LSTMs, CNNs for spatiotemporal analysis) could further improve the accuracy of time-series traffic data prediction.
- **AI-Powered Decision Support System**: Creating an analytics dashboard for government bodies to simulate traffic scenarios and receive AI-suggested interventions.
- **Community Feedback Loop**: Allowing user-submitted traffic updates could create a hybrid system that combines AI predictions with real-world insights.

11. APPENDIX

Source Code

The complete source code for the traffic volume estimation project is available in the GitHub repository linked below. It includes data preprocessing scripts, model training code, prediction modules, and the Streamlit UI components.

- **Uploaded in GitHub repo:**

https://github.com/HaarikaNuthi26/Traffic_Telligence_Project

Dataset Link

We utilized a publicly available dataset containing historical traffic volume, date and time stamps, and corresponding geolocations. This dataset was preprocessed to extract time-based features and fed into the ML model to improve prediction accuracy.

- **Used traffic volume dataset from:**

https://github.com/HaarikaNuthi26/Traffic_Telligence_Project/blob/main/Project%20Files/green-tripdata-2025-03.csv

GitHub & Project Demo Link:

- **GitHub Repository:**

https://github.com/HaarikaNuthi26/Traffic_Telligence_Project

- **Demo Link (Streamlit):**

https://github.com/HaarikaNuthi26/Traffic_Telligence_Project/blob/main/Demo%20Video/Demo.mp4