FINAL REPORT : ADVANCED TRAFFIC VOLUME ESTIMATION WITH MACHINE LEARNING

1. INTRODUCTION

1.1 Project Overview:

This project focuses on predicting traffic volume using historical traffic data. By leveraging machine learning models, it helps forecast traffic congestion in a given area based on factors such as weather, holidays, and time-based features.

1.2 Purpose:

The purpose is to assist traffic authorities and city planners in making data-driven decisions by accurately estimating traffic volume in advance.

2. IDEATION PHASE

2.1 Problem Statement:

Increasing traffic congestion has become a major challenge in urban areas. Predicting traffic volume can aid in better traffic management and planning.

2.2 Empathy Map Canvas:

Stakeholders include commuters, traffic police, and urban planners who need a system to understand and act upon future traffic conditions.

2.3 Brainstorming

What data can help predict traffic?

How to handle holidays and weather?

What models perform best for regression-based traffic forecasting?

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map:

Users (e.g., traffic department officials) input weather and time conditions into the system and get an estimated traffic volume to take necessary action.

3.2 Solution Requirement:

- Clean and reliable traffic data
- Machine learning models
- Web-based user interface

3.3 Data Flow Diagram:

1. User inputs data →

2. Data passed to model →

3. Prediction generated →

4. Result shown in UI

3.4 Technology Stack:

- Frontend: HTML
- Backend: Python (Flask)
- Modeling: Scikit-learn, XGBoost
- IDE: VS Code
- Libraries: Pandas, NumPy, Matplotlib

4. PROJECT DESIGN

4.1 Problem Solution Fit:

The solution offers an accurate and user-friendly method for traffic volume prediction, addressing traffic management issues.

4.2 Proposed Solution:

A Flask-based web app where users enter parameters (date, time, weather, etc.) and receive real-time traffic volume predictions.

4.3 Solution Architecture:

1. Data preprocessing
2. Model training
3. Model deployment using Flask
4. Prediction interface using HTML form

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning:

The project was divided into 5 phases:

1. Data collection & cleaning
2. Model training & evaluation
3. Web development
4. Integration & testing
5. Documentation & deployment

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing:

The performance of the Traffic Volume Estimation model was tested using the R^2 score and RMSE (Root Mean Squared Error).

Random Forest Regressor gave the best performance with an R^2 score of approximately 0.91 on the training set and 0.86 on the testing set.

The RMSE value was calculated to evaluate the error rate, indicating good model reliability and prediction accuracy.

6.2 Functional Testing:

The web application was tested with multiple input combinations to verify that the form accepts values correctly and returns valid traffic volume predictions.

The app consistently provided predictions without failure, and the functionality was verified across multiple test cases.

7. RESULTS

7.1 Output Screenshots:

Include screenshots of:

HTML form UI

Prediction result

Terminal output (Estimated traffic volume)

Any charts or plots used in EDA

8. ADVANTAGES & DISADVANTAGES

Advantages:

Accurate traffic prediction

Easy to use interface

Helps reduce congestion proactively

Disadvantages:

Accuracy depends on dataset quality

May not handle sudden/unusual events (e.g., accidents)

9. CONCLUSION

The Traffic Volume Estimation project effectively uses machine learning to forecast traffic levels. It demonstrates the potential of data science in solving real-world urban planning problems.

10. FUTURE SCOPE

Add live traffic feeds (e.g., via APIs)

Build mobile app version

Include accident and event data for better predictions

11.APPENDIX

Source Code:

GitHub Link:

https://github.com/Manojnavittanala/Traffic-Volume-Estimation

Dataset Link:

https://drive.google.com/file/d/1iV5PfYAmI6YP0_0S4KYy1ZahHOqMgDbM/view

Project Demo Link:

Runs locally using Flask at: http://127.0.0.1:5000/