

PROJECT DESIGN PHASE: ADVANCED TRAFFIC VOLUME ESTIMATION USING MACHINE LEARNING

1. System Architecture:

[User Input Form (HTML)] → [Flask Web Server (app.py)] → [ML Model (Random Forest)] → [Prediction Output]

Explanation:

User Input Form: HTML form collects user input (holiday, weather, temp, time, etc.).

Flask Web App: Receives form data, processes it, applies transformations.

ML Model: A trained RandomForestRegressor model is used to predict traffic volume.

Output:

Prediction result is returned and displayed on the same HTML page.

2. Modules Used:

Frontend:

- *HTML*
- *CSS (for styling)*

Backend:

- *Python*
- *Flask Framework*

Libraries:

- *NumPy*
- *Pandas*
- *Pickle*
- *Scikit-learn*

Model:

- *RandomForestRegressor*

3. Folder Structure:

TRAFFICTELLIGENCE_PROJECT/

├— app.py

├— model.pkl

├— encoder.pkl

├— traffic_volume.csv

├— templates/

├— index.html

├— images/

└— (background image)

4. Workflow Explanation:

- **Data Collection:**

Raw data (traffic_volume.csv) is read and inspected.

- **Preprocessing:**

Handled missing values

Encoded categorical columns like 'holiday' and 'weather'

Split date and time columns

Feature scaling applied

- **Model Training:**

Model initialized using RandomForestRegressor

Data split into training and testing

Model trained and evaluated using R2 Score and RMSE

- **Model Deployment:**

Model and encoders saved using Pickle

Flask app created (app.py) to serve predictions

HTML form developed (index.html) for user interaction

User inputs passed to app.py, processed, and prediction returned

6. Final Output:

The user receives a predicted value of traffic volume based on the entered parameters in a simple and user-friendly web interface.

Prepared By: [Manojna Vittanala]

Date: [27-6-2025]