TrafficTelligence: Advanced Traffic Volume Estimation With Machine Learning

# 1. Project Overview

This project estimates the traffic volume based on weather conditions, date, time, and other relevant parameters. It uses a machine learning model trained on historical traffic data to predict the expected number of vehicles.

# 2. Technologies Used

• Python  
• Flask (for the web interface)  
• Scikit-learn (for model building and preprocessing)  
• Pandas & NumPy (for data handling)  
• HTML & CSS (for frontend)  
• Matplotlib & Seaborn (for visualization)  
• Pickle (for saving models and encoders)

# 3. Data Preprocessing

Steps Involved:  
• Load dataset traffic volume.csv  
• Handle missing values:  
 - Numerical columns ('temp', 'rain', 'snow') filled with mean  
 - Categorical columns ('weather', 'holiday') filled with most frequent/default values  
• Date column split into day, month, year  
• Time column split into hours, minutes, seconds  
• Encode 'holiday' and 'weather' using LabelEncoder  
• Feature scaling using StandardScaler  
• Features and target variable separated  
• Train-test split applied  
• Encoders and scaler are saved for use in the web app:  
 - encoder\_holiday.pkl  
 - encoder\_weather.pkl  
 - scaler.pkl

# 4. Model Building

• RandomForestRegressor is used for training  
• Model is evaluated using R² score for both training and test sets  
• Trained model saved as model.pkl

# 5. Flask Application Structure

Traffic Intelligence/  
├── app.py # Main Flask app  
├── templates/  
│ └── index.html # Web interface  
├── static/  
│ └── images.jpg # Background image  
├── model.pkl # Trained ML model  
├── scaler.pkl # Saved StandardScaler  
├── encoder\_holiday.pkl # Holiday LabelEncoder  
├── encoder\_weather.pkl # Weather LabelEncoder  
└── traffic volume.csv # Dataset

# 6. app.py Functionality

• Loads saved model, encoders, and scaler  
• Accepts user inputs through web form  
• Encodes categorical variables  
• Scales inputs  
• Predicts traffic volume  
• Displays result back on the webpage

# 7. index.html Interface

Form to input:  
• Holiday type  
• Weather condition  
• Temperature  
• Rain & Snow amounts  
• Date (Year, Month, Day)  
• Time (Hours, Minutes, Seconds)  
  
Displays prediction result below the form  
Styled with CSS and background image

# 8. Common Errors Handled

• Input validation (e.g., number formats)  
• Unseen labels in dropdown handled by matching training data options  
• Consistent feature order to avoid scaler mismatches

# 9. Future Enhancements

• Dynamic dropdown options populated from encoder classes  
• Model performance tuning  
• Interactive data visualization on frontend  
• Upload option for bulk predictions

# 10. Conclusion

This project demonstrates traffic volume estimation using real-world data, machine learning, and web integration. The system allows users to input relevant parameters and obtain quick predictions through an intuitive web interface.