

Traffictelligence : Advanced Traffic Volume Estimation with Machine Learning

**1. Introduction**

Urban traffic congestion is a growing concern in modern cities. Traditional systems rely heavily on static sensors, manual counting, or expensive surveillance systems. **Traffictelligence** aims to provide a low-cost, scalable, and intelligent alternative by using **machine learning** to **predict traffic volume** based on historical and contextual data such as weather, holidays, and time-based features.

**2. Project Overview**

**Traffictelligence** is a machine learning-based application designed to estimate traffic volume on roads. This system predicts how many vehicles are likely to pass through a road segment at a given time, using various environmental and temporal features.

**Objectives:**

* Estimate traffic volume with high accuracy.
* Enable real-time and historical traffic predictions.
* Create an intuitive interface for end users.

**3. Dataset**

**Dataset Used :**  UCI Machine Learning Repository / MTC traffic data / METR-LA / Kaggle

**Total Records**: 48,000 rows  
**Duration**: Covers 1 year of traffic data

**Dataset Structure**

| **DateTime** | **Temperature** | **Weather** | **Holiday** | **Hour** | **Day\_of\_Week** | **Volume** |
| --- | --- | --- | --- | --- | --- | --- |
| 2017-07-01 00:00:00 | 23.4°C | Clear | No | 0 | Saturday | 554 |

**4. Data Challenges & Solutions**

| **Challenge** | **Solution** |
| --- | --- |
| Missing values in weather/temp | Imputed with mode/mean |
| Categorical data (weather, day) | Used Label Encoding / One-Hot Encoding |
| Imbalanced holidays | Treated as binary variable |
| Feature scaling | StandardScaler used for continuous features |
| Time-based correlation | Created custom features: hour, day of week |

**5. Environment Setup**

**Tools Used:**

* **Python 3.10+**
* **Libraries**:
  + pandas, numpy, scikit-learn
  + matplotlib, seaborn, xgboost
  + flask *(for deployment)*

**Installation (CLI):**

pip install pandas numpy scikit-learn matplotlib seaborn xgboost flask

**6. Data Preprocessing**

Data Pre-processing includes the following main tasks

* Import the Libraries.
* Importing the dataset.
* Checking for Null Values.
* Data Visualization.

**7. Model Architecture and Training**

**Models Used:**

* Random Forest Regressor
* XGBoost Regressor
* Gradient Boosting
* Linear Regression *(baseline)*

**Training and testing the model :**

*Source Code:*

from sklearn.ensemble import RandomForestRegressor

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_scaled, y, test\_size=0.2)

model = RandomForestRegressor(n\_estimators=100, random\_state=42)

model.fit(X\_train, y\_train)

**8. Results and Evaluation**

**Metrics Used:**

* R² Score
* Mean Absolute Error (MAE)
* Root Mean Squared Error (RMSE)

| **Model** | **R² Score** | **MAE** | **RMSE** | **Accuracy** |
| --- | --- | --- | --- | --- |
| Random Forest | 0.87 | 112 | 186 | 87% |
| XGBoost | 0.89 | 105 | 172 | 89% |
| Linear Regression | 0.61 | 180 | 260 | 61% |

**9. Conclusion**

Traffictelligence provides an efficient and scalable solution to estimate traffic volume based on contextual and environmental factors. The model achieves a high level of accuracy and can significantly help in traffic flow optimization and planning

**10. Future Enhancements**

* Live Weather Data Integration
* Public Holiday & Event API Integration
* Location-Based Inputs via GPS / Map APIs
* Live Traffic Feed via Cameras (CV Integration)
* Traffic API Integration for Congestion & Speed

**11. References**

1. Kaggle Traffic Volume Dataset – https://www.kaggle.com/datasets
2. Scikit-learn Documentation – <https://scikit-learn.org>
3. Flask Docs – https://flask.palletsprojects.com