TrafficTelligence: Advanced Traffic Volume Estimation With Machine Learning

# 1. Project Overview

TrafficTelligence is a smart solution that leverages machine learning to estimate traffic volume based on historical traffic data. The system aims to predict traffic congestion trends and aid in intelligent traffic management and planning.

# 2. Pre Requisites

- Python 3.x installed

- Jupyter Notebook or Google Colab

- Libraries: pandas, numpy, matplotlib, seaborn, scikit-learn, tensorflow (optional)

- Internet connection for dataset download

- Basic knowledge of data science and ML pipelines

# 3. Prior Knowledge

To successfully complete this project, users should understand:

- Basic Python programming

- Data cleaning and visualization

- Machine learning concepts (Regression)

- Evaluation metrics (MAE, RMSE, R2 Score)

# 4. Project Objectives

- Analyze historical traffic data

- Preprocess and clean the dataset

- Build regression models for volume prediction

- Evaluate model accuracy and performance

- Deploy the model in a simple web app (optional)

# 5. Project Flow

1. Data Collection

2. Data Preprocessing

3. Exploratory Data Analysis (EDA)

4. Model Selection and Training

5. Model Evaluation

6. Application Deployment

# 6. Project Structure

traffic-telligence/

├── data/

│ └── traffic\_data.csv

├── notebooks/

│ └── eda.ipynb

│ └── model\_training.ipynb

├── model/

│ └── traffic\_model.pkl

├── app/

│ └── app.py

├── requirements.txt

└── README.md

# 7. Data Collection

Use the Metro Interstate Traffic Volume dataset from UCI or Kaggle.

Sample loading code:

```python

import pandas as pd

df = pd.read\_csv("Metro\_Interstate\_Traffic\_Volume.csv")

print(df.head())

```

# 8. Data Pre-Processing

Steps:

- Handle missing values

- Convert date-time to features

- Remove outliers

- Encode categorical variables

Example:

```python

df['date\_time'] = pd.to\_datetime(df['date\_time'])

df['hour'] = df['date\_time'].dt.hour

df = df.dropna()

```

# 9. Model Building

Use regression models such as:

- Linear Regression

- Decision Tree Regressor

- Random Forest Regressor

Example:

```python

from sklearn.ensemble import RandomForestRegressor

model = RandomForestRegressor()

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

predictions = model.predict(X\_test)

```

# 10. Application Building

Use Flask or Streamlit for web app interface.

Sample with Streamlit:

```python

import streamlit as st

import pickle

model = pickle.load(open('traffic\_model.pkl', 'rb'))

hour = st.slider("Hour of the day", 0, 23, 8)

prediction = model.predict([[hour]])

st.write(f"Estimated traffic volume: {int(prediction[0])}")

```