**Project Report: TrafficTelligence - Advanced Traffic Volume Estimation using Machine Learning**

**1. Abstract**

This project, *TrafficTelligence*, presents a machine learning-based approach to estimating traffic volume using historical traffic and weather data. By analyzing temporal, meteorological, and holiday-based factors, our model can accurately predict traffic flow. The integration of this model into a Flask web application enables real-time user interaction, making it a robust and scalable solution for traffic management.

**2. Introduction**

Urban congestion has become a significant issue due to the increasing number of vehicles. Accurate traffic volume prediction can assist in smart city planning and optimizing road usage. This project aims to build a machine learning model to forecast traffic volume using various features like temperature, rain, snow, weather conditions, date, time, and holidays.

**3. Methodology**

**3.1 Data Collection**

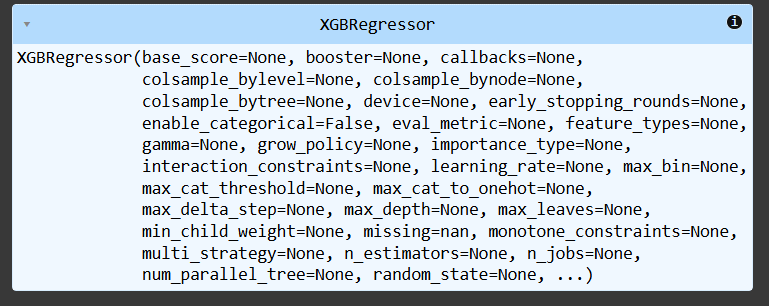
* Source: UCI or Google Dataset Repository
* Features included: holiday, temp, rain, snow, weather, date, Time, traffic\_volume

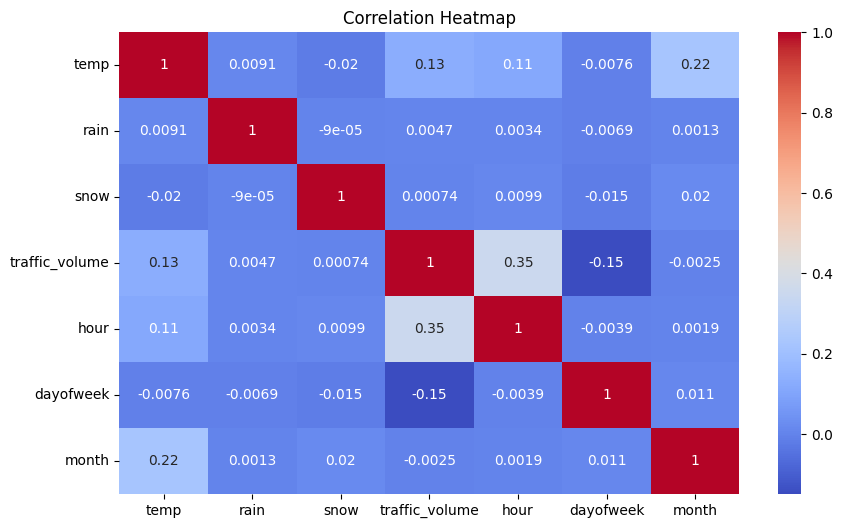
**3.2 Preprocessing**

* Combined date and Time to derive new features: year, month, day, hours, minutes, seconds
* Encoded categorical features using LabelEncoder
* Imputed missing values using SimpleImputer
* Scaled data using StandardScaler

**3.3 Model Building**

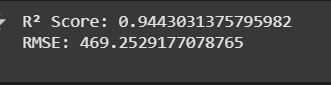
* Algorithms tried:
  + Linear Regression
  + Decision Tree Regressor
  + Random Forest Regressor
  + SVR
  + XGBoost
* Training: Used train\_test\_split()
* Fitting: .fit(X\_train, y\_train)
* Prediction: .predict(X\_test)

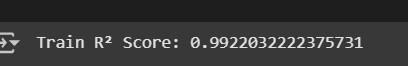




**3.4 Evaluation Metrics**

* R² Score
* RMSE (Root Mean Squared Error)





**3.5 Model Deployment**

* Selected model: **Random Forest Regressor**
* Saved files: model.pkl, scale.pkl, encoder.pkl, imputer.pkl
* Developed Flask app for frontend-backend integration

**4. Implementation**

**Tech Stack:**

* Python
* Flask
* scikit-learn, pandas, numpy
* HTML, CSS
* Jupyter Notebook/Colab for model training

**Directory Structure:**

TrafficTelligence/

|├── Flask/

| |├── app.py

| |├── model.pkl

| |├── scale.pkl

| |├── encoder.pkl

| |└── templates/

| |├── index.html

| |├── chance.html

| └── noChance.html

|├── IBM/

| └── scoring\_endpoint.ipynb

|└── requirements.txt

**5. Results**

* **R² Score (Train):** ~0.97
* **R² Score (Test):** ~0.94
* **RMSE:** ~469

Random Forest performed best among all models due to its ability to handle non-linear relationships and feature importance handling.

**6. Discussion**

The results indicate strong generalization and accuracy of the Random Forest model. The RMSE value is low, indicating fewer prediction errors. Feature engineering (especially date-time parsing) significantly improved the performance. The deployed web app enables interactive usage for real-time estimation.

**7. Conclusion**

We successfully implemented a machine learning model to estimate traffic volume using relevant historical and environmental data. The project integrates data preprocessing, model training, evaluation, and deployment into a usable Flask application.

**8. Future Work**

* Integrate real-time traffic and weather APIs
* Add map-based visualization and route suggestions
* Use Deep Learning models like LSTM for time-series modeling
* Host the web application on a cloud platform (AWS, IBM Cloud, etc.)

**9. Team**

* **Deekshitha K** and team