Final Project Report

TrafficTelligence: Advanced Traffic Volume Estimation with Machine Learning

1.Introduction

1.1 Project overviews

Traffic management is a major challenge in urban areas, where increasing populations and vehicle counts have led to frequent traffic jams and congestion. Traditional traffic management solutions rely on historical data or manual monitoring, which often results in delayed responses to changes in traffic flow. **TrafficTelligence** uses **machine learning (ML)** algorithms to analyze real-time data from multiple sources—traffic sensors, GPS, weather data, and social events—providing accurate traffic volume predictions. These predictions allow better planning for road usage, improved traffic control, and optimized transportation systems, resulting in shorter travel times and lower carbon emissions.

The **primary objective** of TrafficTelligence is to **develop an accurate and scalable machine learning model** for predicting traffic volume, empowering city planners, traffic authorities, and commuters to make data-driven decisions. This system aims to overcome the limitations of traditional traffic estimation methods by integrating **real-time and historical data** to offer **dynamic forecasts** of traffic conditions. Below is a breakdown of the key objectives:

1.2 Objectives

The main objective of **TrafficTelligence** is to develop a **machine learning-based traffic volume estimation system** that accurately predicts traffic patterns in real-time. The system aims to enhance urban transportation management, reduce congestion, and improve the commuter experience through dynamic and data-driven insights.

Specific Objectives:

1. **Accurate Traffic Prediction:**
   - Build a reliable ML model to predict traffic volume based on historical and real-time data.
   - Improve the precision of traffic forecasts compared to traditional estimation methods.
2. **Real-Time Insights:**
   - Integrate live data feeds from **traffic sensors, weather APIs, and public event schedules** to provide dynamic traffic predictions.
   - Continuously update the model to reflect current conditions for accurate forecasts.
3. **Optimized Traffic Management:**
   - Assist authorities in **synchronizing traffic signals**, planning road usage, and managing public transportation.
   - Enable **proactive responses** to potential congestion and disruptions.
4. **Reduced Environmental Impact:**
   - Promote **efficient routing** and minimize vehicle idle times to lower emissions and fuel consumption.
   - Support sustainable urban mobility through optimized traffic flows.
5. **Scalability and Adaptability:**
   - Design a flexible solution that can **scale across multiple cities** and adapt to various infrastructure and traffic patterns.
   - Ensure compatibility with new data sources and technologies such as **IoT devices**.
6. **Enhanced Commuter Experience:**
   - Provide actionable predictions through **web and mobile platforms**, allowing commuters to plan better routes.
   - Reduce travel time, frustration, and uncertainty for users by offering **real-time alerts**.
7. **Data-Driven Decision Support:**
   - Empower city planners and traffic managers with predictive insights to **improve long-term urban planning**.
   - Prepare for **unexpected events** such as accidents or weather disruptions, ensuring smooth traffic operations.

2. Project Initialization and Planning Phase

2.1 Define Problem Statements (Customer Problem Statement Template):

Managing traffic flow efficiently in urban areas remains a challenge due to population growth and increasing vehicle counts. Traditional data collection methods are limited in scope and accuracy, leading to poor traffic predictions. This results in congestion, increased pollution, and inefficient transportation planning.

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	A city traffic manager or planner	Forecast traffic volume	Current methods are manual & slow	Growing traffic levels & complex patterns	Frustrated with inefficiency and delays
PS-2	A public commuter or driver	Plan my daily commute	Traffic is unpredictable.	Lack of real-time traffic insights	Stressed and prone to delays

SMARTBRIDGE
Let's Bridge the Gap
a Veranda Enterprise

Smart Internz

I am	I'm trying to	But	Because	Which makes me feel
• A city traffic manager or planner	• Forecast traffic volume	• Current methods are manual & slow	• Growing traffic levels & complex patterns	• Frustrated with inefficiency and delays

I am	I'm trying to	But	Because	Which makes me feel
• A public commuter or driver	• Plan my daily commute	• Traffic is unpredictable	• Lack of real-time traffic insights	• Stressed and prone to delays

2.2 Project Proposal (Proposed Solution) template

This project proposal outlines a solution to address a specific problem. With a clear objective, defined scope, and a concise problem statement, the proposed solution details the approach, key features, and resource requirements, including hardware, software, and personnel.

Project Overview	
Objective	The primary objective of **TrafficTelligence** is to develop an **advanced, machine learning-based traffic volume estimation system** that provides **real-time, predictive insights** to improve urban traffic management. This solution aims to help city planners, commuters, and officials by optimizing traffic flow, reducing congestion, minimizing travel delays, and supporting sustainable urban development.
Scope	TrafficTelligence focuses on developing a machine learning-based system for **accurate traffic volume prediction** using real-time and historical data from sensors, weather APIs, and events. The project includes **model development, validation, and deployment** through a web or mobile interface, offering real-time alerts and route recommendations. It supports **traffic management optimization** by aiding city planners in reducing congestion and minimizing environmental impact through efficient routing. Designed to **scale across cities**, the solution is adaptable to local traffic patterns and integrates with **IoT devices** for enhanced monitoring. The scope is limited to traffic volume estimation, excluding vehicle-level management and toll systems
Problem Statement	
Description	Urban areas struggle with increasing traffic congestion due to outdated forecasting methods and the inability to capture real-time events like weather changes and accidents. This results in delays, commuter frustration, and higher emissions. **TrafficTelligence** aims to solve this by developing a machine learning-based traffic volume prediction system that integrates real-time data from sensors, weather APIs, and events. The solution will provide accurate forecasts to help city planners optimize traffic flow, reduce congestion, and improve the overall commuter experience.

Impact	**TrafficTelligence** will enhance urban mobility by providing **accurate traffic forecasts**, reducing congestion, and minimizing delays for commuters. It will help city planners **optimize signal timings and routes**, leading to smoother traffic flow and lower fuel consumption. The project will contribute to **reducing carbon emissions**, promoting sustainable transportation. Additionally, real-time predictions will improve the **efficiency of emergency responses** and reduce the frustration of daily travelers. The system's scalability ensures it can be **adapted to different cities and evolving traffic patterns**.
Proposed Solution	
Approach	The approach for **TrafficTelligence** involves a systematic process beginning with the **collection of real-time and historical data** from traffic sensors, weather APIs, and public events. This data undergoes **cleaning and preprocessing** to handle missing values and normalize features. Next, various machine learning models, including **Random Forest and XGBoost**, are developed and optimized through hyperparameter tuning. Finally, the solution is deployed via a user-friendly web or mobile interface, providing real-time traffic predictions and alerts to enhance urban mobility and planning.
Key Features	**TrafficTelligence** offers real-time traffic predictions by integrating live data from sensors, weather conditions, and local events. Its user-friendly web and mobile interface provides accurate forecasts, alternative routes, and congestion alerts. Utilizing advanced machine learning algorithms, the system is scalable and adaptable to various urban environments, supporting city planners in proactive traffic management and environmental impact monitoring.

Resource Requirements

Resource Type	**Description**	**Specification/Allocation**
Hardware		
Computing Resources	GPUs for model training	2 x NVIDIA V100 GPUs

Memory	RAM for processing large datasets	16 GB RAM
Storage	Disk space for models and logs	1 TB SSD
Software		
Frameworks	Python frameworks	Flask, FastAPI
Libraries	Additional machine learning tools	TensorFlow, PyTorch
Development Environment	IDE and version control tools	Jupyter Notebook, Git
Data		
Data	Data source, size, and format	Traffic sensors & GPS data, CSV format

2.3 Initial Project Planning Template

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members	Sprint Start Date	Sprint End Date (Planned)
Sprint-1	User Registration and Login	USN-1	Authentication & Account Setup	2	High	Jhansi, Nandini, Durga, Bharath	16 Aug 2024	18 aug 2024

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members	Sprint Start Date	Sprint End Date (Planned)
Sprint-1	User Registration and Login	USN-2	Email Verification	1	High	Jhansi, Nandini, Durga, Bharath	16 Aug 2024	18 aug 2024
Sprint-1	User Registration and Login	USN-3	User Authentication & Login	1	Low	Jhansi, Nandini, Durga, Bharath	16 Aug 2024	18 aug 2024
Sprint-2	Data Preprocessing and Model Selection	USN-4	Data Cleaning, Encoding	2	Medium	Jhansi, Nandini, Durga, Bharath	20 Aug 2024	25 Aug 2024
Sprint-2	Data Preprocessing and Model Selection	USN-5	Model Training & Tuning	5	High	Jhansi, Nandini, Durga, Bharath	20 Aug 2024	25 Aug 2024
Sprint-3	Model Evaluation and Web Deployment	USN-8	Performance Evaluation	3	Medium	Jhansi, Nandini, Durga, Bharath	01 Sep 2024	5 Sep 2024
Sprint-3	Model Evaluation and Web Deployment	USN-9	Backend Development	4	High	Jhansi, Nandini, Durga, Bharath	01 Sep 2024	5 Sep 2024
Sprint-4	Web Application Testing & Deployment	USN-10	Testing & Deployment	5	Medium	Jhansi, Nandini, Durga, Bharath	6 Sep 2024	10 Sep 2024

3. Data Collection and Preprocessing Phase

3.1 Data Collection Plan & Raw Data Sources Identification Template

Elevate your data strategy with the Data Collection plan and the Raw Data Sources report, ensuring meticulous data curation and integrity for informed decision-making in every analysis and decision-making endeavor.

Data Collection Plan Template

Section	Description
Project Overview	**TrafficTelligence** aims to develop a machine learning-based solution for real-time traffic volume estimation. By integrating data from traffic cameras, GPS, and IoT sensors, the project seeks to provide accurate predictions of traffic patterns and congestion. The solution will feature a user-friendly dashboard for city planners and commuters, enabling proactive traffic management and enhancing urban mobility. With a focus on continuous learning and scalability, **TrafficTelligence** will support sustainable urban development by reducing congestion and emissions.
Data Collection Plan	• Location data from vehicles to track movement and analyze traffic patterns. • Roadside sensors that measure vehicle speeds, counts, and environmental conditions.
Raw Data Sources Identified	Information sourced from weather stations and meteorological services, including temperature, precipitation, and wind conditions. This data helps analyze the impact of weather on traffic patterns.

	Location information collected from vehicles equipped with GPS devices. This data provides insights into vehicle movements, travel times, and congestion patterns across different routes.

Raw Data Sources Template

Source Name	Description	Location/URL	Format	Size	Access Permissions
Smart Internz Platform	This dataset includes location and speed data collected from vehicles equipped with GPS devices, tracking their movements over time.	https://drive.google.com/file/d/19GB6dD8TuX5riAk2PiWItRCz_jm1cx-j/view?usp=sharing	CSV	2.2 MB	Public

SMARTBRIDGE Let's Bridge the Gap a Veranda Enterprise

Smart Internz

Data Collection and Preprocessing Phase

3.2 Data Quality Report Template

The Data Quality Report Template will summarize data quality issues from the selected source, including severity levels and resolution plans. It will aid in systematically identifying and rectifying data discrepancies.

Data Source	Data Quality Issue	Severity	Resolution Plan
Smart Internz Dataset	Categorical data in the dataset	Moderate	Encoding has to be done in the data

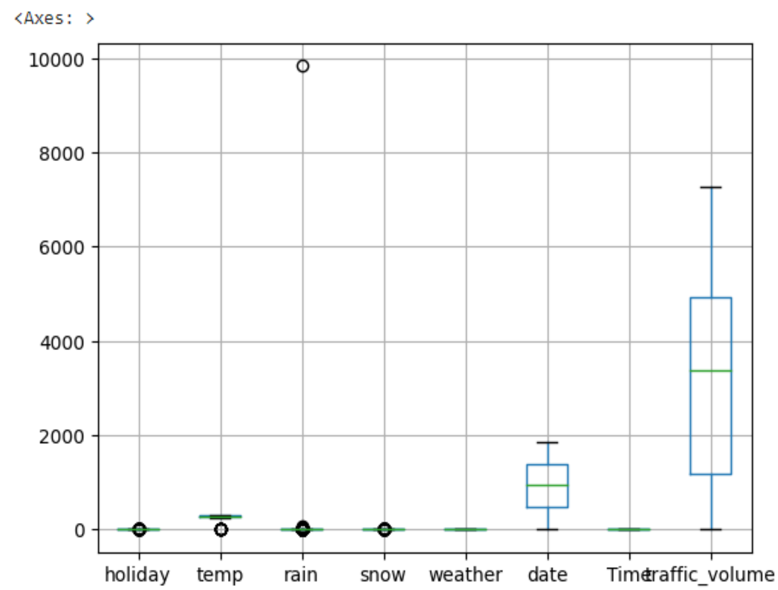
Data Collection and Preprocessing Phase

3.3 Data Exploration and Preprocessing Template

Identifies data sources, assesses quality issues like missing values and duplicates, and implements resolution plans to ensure accurate and reliable analysis.

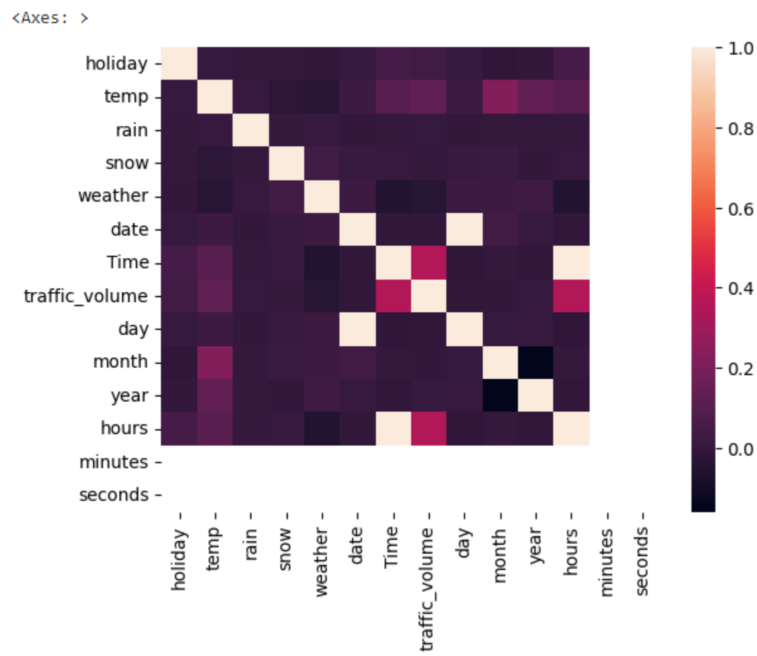
<table>
<tr><th>Section</th><th>Description</th></tr>
<tr><td>Data Overview</td><td>
<table>
<tr><th></th><th>temp</th><th>rain</th><th>snow</th><th>traffic_volume</th></tr>
<tr><td>count</td><td>48151.000000</td><td>48202.000000</td><td>48192.000000</td><td>48204.000000</td></tr>
<tr><td>mean</td><td>281.205351</td><td>0.334278</td><td>0.000222</td><td>3259.818355</td></tr>
<tr><td>std</td><td>13.343675</td><td>44.790062</td><td>0.008169</td><td>1986.860670</td></tr>
<tr><td>min</td><td>0.000000</td><td>0.000000</td><td>0.000000</td><td>0.000000</td></tr>
<tr><td>25%</td><td>272.160000</td><td>0.000000</td><td>0.000000</td><td>1193.000000</td></tr>
<tr><td>50%</td><td>282.460000</td><td>0.000000</td><td>0.000000</td><td>3380.000000</td></tr>
<tr><td>75%</td><td>291.810000</td><td>0.000000</td><td>0.000000</td><td>4933.000000</td></tr>
<tr><td>max</td><td>310.070000</td><td>9831.300000</td><td>0.510000</td><td>7280.000000</td></tr>
</table>
Dimension: 145460 rows × 23 columns
</td></tr>
</table>

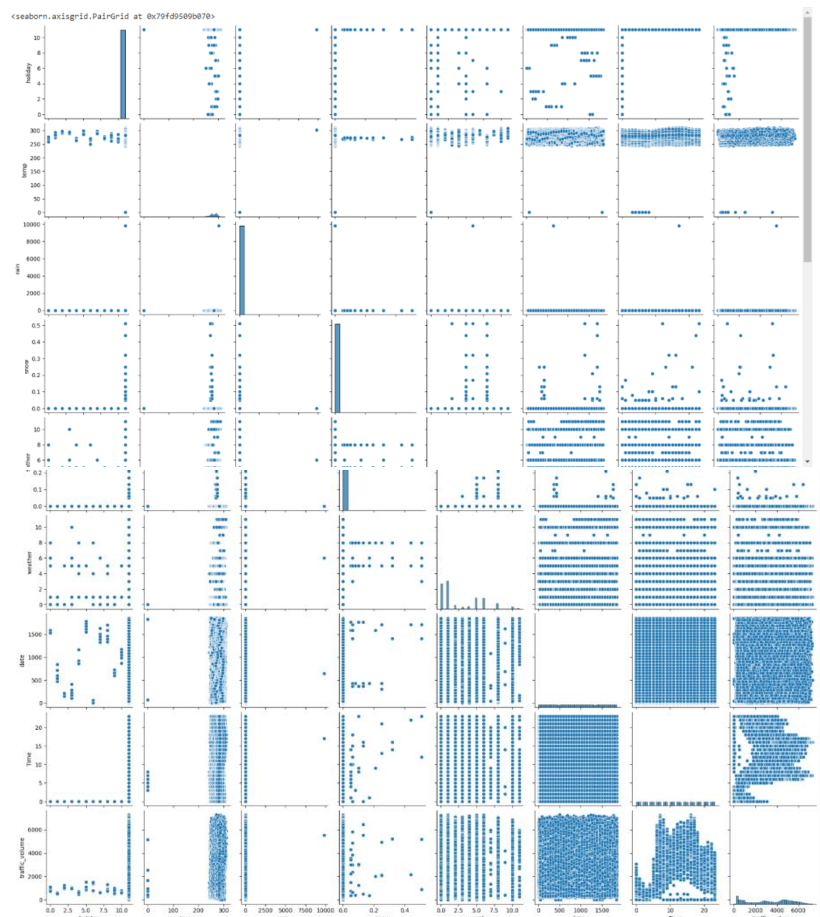
Univariate Analysis



	0
holiday	48143
temp	53
rain	2
snow	12
weather	49
date	0
Time	0
traffic_volume	0

dtype: int64

<table>
<tr>
<td>Bivariate Analysis</td>
<td>


</td>
</tr>
<tr>
<td>Multivariate Analysis</td>
<td>


</td>
</tr>
</table>

<table>
<tr><td>Outliers and Anomalies</td><td>----</td></tr>
<tr><td colspan="2">Data Preprocessing Code Screenshots</td></tr>
<tr><td>Loading Data</td><td>
LOADING DATASET
<pre>
data=pd.read_csv("/content/traffic volume.csv")

[4] data.head()
</pre>
<table>
<tr><th></th><th>holiday</th><th>temp</th><th>rain</th><th>snow</th><th>weather</th><th>date</th><th>Time</th><th>traffic_volume</th></tr>
<tr><td>0</td><td>NaN</td><td>288.28</td><td>0.0</td><td>0.0</td><td>Clouds</td><td>02-10-2012</td><td>09:00:00</td><td>5545</td></tr>
<tr><td>1</td><td>NaN</td><td>289.36</td><td>0.0</td><td>0.0</td><td>Clouds</td><td>02-10-2012</td><td>10:00:00</td><td>4516</td></tr>
<tr><td>2</td><td>NaN</td><td>289.58</td><td>0.0</td><td>0.0</td><td>Clouds</td><td>02-10-2012</td><td>11:00:00</td><td>4767</td></tr>
<tr><td>3</td><td>NaN</td><td>290.13</td><td>0.0</td><td>0.0</td><td>Clouds</td><td>02-10-2012</td><td>12:00:00</td><td>5026</td></tr>
<tr><td>4</td><td>NaN</td><td>291.14</td><td>0.0</td><td>0.0</td><td>Clouds</td><td>02-10-2012</td><td>13:00:00</td><td>4918</td></tr>
</table>
</td></tr>
<tr><td>Handling Missing Data</td><td>
<pre>
[7] data.isnull().sum()
</pre>
<table>
<tr><th></th><th>0</th></tr>
<tr><td>holiday</td><td>48143</td></tr>
<tr><td>temp</td><td>53</td></tr>
<tr><td>rain</td><td>2</td></tr>
<tr><td>snow</td><td>12</td></tr>
<tr><td>weather</td><td>49</td></tr>
<tr><td>date</td><td>0</td></tr>
<tr><td>Time</td><td>0</td></tr>
<tr><td>traffic_volume</td><td>0</td></tr>
</table>
dtype: int64
<pre>
data['temp'].fillna(data['temp'].mean(),inplace=True)
data['rain'].fillna(data['rain'].mean(),inplace=True)
data['snow'].fillna(data['snow'].mean(),inplace=True)

<ipython-input-8-fe7e5557f060>:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through ch

data['weather'].fillna('clouds',inplace=True)

<ipython-input-10-541126c44dcd>:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series througl
</pre>
✓ 1s completed at 12:10PM
</td></tr>
</table>

Data Transformation

```
[60] y=data['traffic_volume']
     x=data.drop(columns=['traffic_volume'],axis=1)

[61] x.shape
     y.shape

(48204,)

[62] names=x.columns

[63] from sklearn.preprocessing import scale

[64] data['holiday']=le.fit_transform(data['holiday'])

     x = scale(x)
```

```
[11] data[["day", "month", "year"]] =data["date"].str.split("-",expand = True)

[12] data[["hours", "minutes", "seconds"]] = data["Time"].str.split(":", expand = True)

[13] from sklearn.preprocessing import LabelEncoder
     le=LabelEncoder()
     data['weather']=le.fit_transform(data['weather'])

[14] data['holiday']=le.fit_transform(data['holiday'])

[15] data['date']=le.fit_transform(data['date'])

[16] data['Time']=le.fit_transform(data['Time'])
```

```
[25] data.drop(columns = ["date", "Time"],axis=1,inplace = True)

     data.head()
```

	holiday	temp	rain	snow	weather	traffic_volume
0	NaN	288.28	0.0	0.0	Clouds	5545
1	NaN	289.36	0.0	0.0	Clouds	4516
2	NaN	289.58	0.0	0.0	Clouds	4767
3	NaN	290.13	0.0	0.0	Clouds	5026
4	NaN	291.14	0.0	0.0	Clouds	4918

Feature Engineering	Attached the codes in final submission
Save Processed Data	-

4.Model Development Phase Template

4.1 **Feature Selection Report Template**

In the forthcoming update, each feature will be accompanied by a brief description. Users will indicate whether it's selected or not, providing reasoning for their decision. This process will streamline decision-making and enhance transparency in feature selection.

Feature	Description	Selected (Yes/No)	Reasoning
Holiday	Indicates if the day was a holiday	Yes	Traffic volume can differ significantly on holidays..
Temperature	Temperature in Kelvin.	Yes	Weather conditions such as temperature can affect traffic patterns
Rain	Amount of rain in millimeters	Yes	Rain affects driving conditions, which can influence traffic volume..
Snow	Amount of snow in millimeters	**Yes**	Snow impacts road conditions, hence affecting traffic..
Weather	Weather condition (e.g., clouds, clear)	**Yes**	Weather conditions affect traffic flow and driver behavior.
Date	The date of traffic data recording	**No**	Date by itself might not provide much insight without further aggregation..

Time	The time of day	**Yes**	Traffic volume varies significantly based on the time of day (e.g., rush hours).
Traffic Volume	The recorded traffic volume	**No**	This is the target variable we are predicting, not a feature to be selected

Model Development Phase Template

4.2 Model Selection Report

In the forthcoming Model Selection Report, various models will be outlined, detailing their descriptions, hyperparameters, and performance metrics, including Accuracy or F1 Score. This comprehensive report will provide insights into the chosen models and their effectiveness.

Model Selection Report:

Model	**Description**	**Hyperparameters**	**Performance Metric (e.g., Accuracy, F1 Score)**
Random Forest	Ensemble learning algorithm that builds multiple decision trees and combines their predictions to improve accuracy and prevent overfitting	-	Accuracy score=80%
XGBoost	Gradient boosting model used for better accuracy	-	Accuracy score=80%

Decision Tree	Decision tree for predicting traffic volume	-	Accuracy score=68%

Model Development Phase Template

4.3 Initial Model Training Code, Model Validation and Evaluation Report

The initial model training code will be showcased in the future through a screenshot. The model validation and evaluation report will include classification reports, accuracy, and confusion matrices for multiple models, presented through respective screenshots.

Initial Model Training Code:

Paste the screenshot of the model training code

```
[97] from sklearn import linear_model
     from sklearn import tree
     from sklearn import ensemble
     from sklearn import svm
     import xgboost
```

```
[98] lin_reg=linear_model.LinearRegression()
     Dtree=tree.DecisionTreeRegressor()
     Rand=ensemble.RandomForestRegressor()
     svr=svm.SVR()
     XGB=xgboost.XGBRegressor()
```

```
lin_reg.fit(x_train,y_train)
Dtree.fit(x_train,y_train)
Rand.fit(x_train,y_train)
svr.fit(x_train,y_train)
XGB.fit(x_train,y_train)
```

[100]
```python
p1=lin_reg.predict(x_train)
p2=Dtree.predict(x_train)
p3=Rand.predict(x_train)
p4=svr.predict(x_train)
p5=XGB.predict(x_train)
```

[101]
```python
from sklearn import metrics
```

[102]
```python
print(metrics.r2_score(p1,y_train))
print(metrics.r2_score(p2,y_train))
print(metrics.r2_score(p3,y_train))
print(metrics.r2_score(p4,y_train))
print(metrics.r2_score(p5,y_train))
```

[103]
```python
p1=lin_reg.predict(x_test)
p2=Dtree.predict(x_test)
p3=Rand.predict(x_test)
p4=svr.predict(x_test)
p5=XGB.predict(x_test)
```

```python
print(metrics.r2_score(p1,y_test))
print(metrics.r2_score(p2,y_test))
print(metrics.r2_score(p3,y_test))
print(metrics.r2_score(p4,y_test))
print(metrics.r2_score(p5,y_test))
```

[105]
```python
MSE=metrics.mean_squared_error(p3,y_test)
```

```python
np.sqrt(MSE)
```

```
797.8660187218496
```

[109]
```python
import pickle
```

```python
pickle.dump(Rand,open("model.pk1",'wb'))
pickle.dump(le,open("encoder.pk1",'wb'))
```

SMARTBRIDGE Let's Bridge the Gap a Veranda Enterprise

Smart Internz

Model Validation and Evaluation Report:

Model	Classification Report	Accuracy	Confusion Matrix
Random Forest	[104] print(metrics.r2_score(p1,y_test)) print(metrics.r2_score(p2,y_test)) print(metrics.r2_score(p3,y_test)) print(metrics.r2_score(p4,y_test)) print(metrics.r2_score(p5,y_test)) -5.365880084970513 0.6896537016306165 0.8034531412767689 -11.986624908714624 0.8092036247253418 [105] MSE=metrics.mean_squared_error(p3,y_test) [106] np.sqrt(MSE) 797.8660187218496	80%	---

5.Model Optimization and Tuning Phase Template

Model Optimization and Tuning Phase

The Model Optimization and Tuning Phase involves refining machine learning models for peak performance. It includes optimized model code, fine-tuning hyperparameters, comparing performance metrics, and justifying the final model selection for enhanced predictive accuracy and efficiency.

Hyperparameter Tuning Documentation (6 Marks):

Model	Tuned Hyperparameters	Optimal Values
----	----	

Performance Metrics Comparison Report (2 Marks):

Model	Baseline Metric	Optimized Metric
----	----	

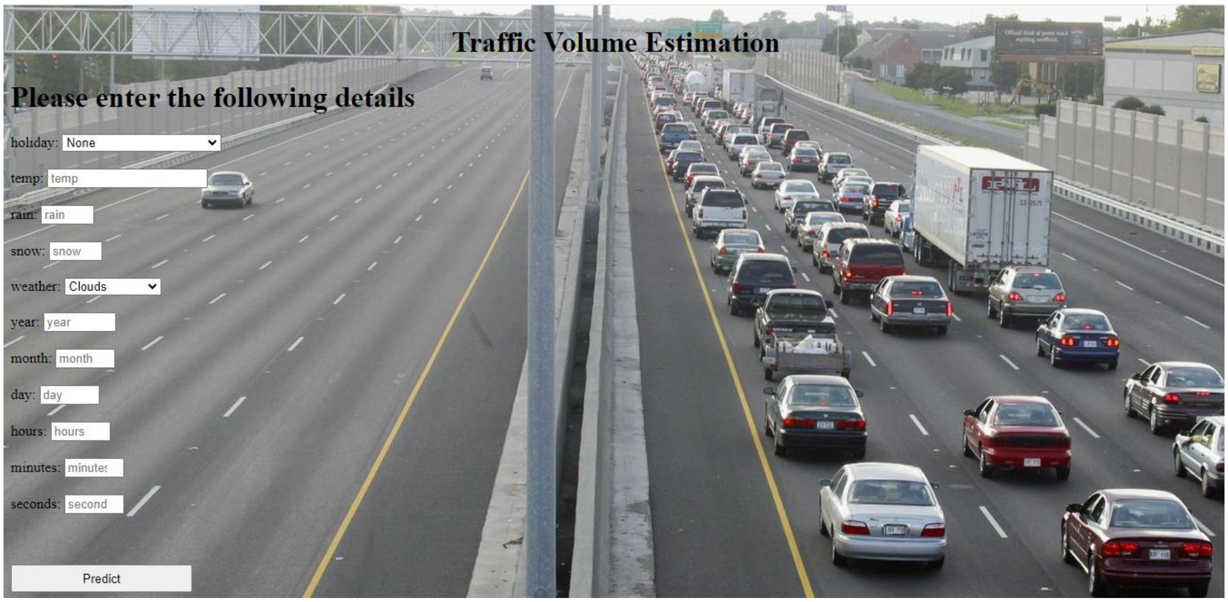
Final Model Selection Justification:

Final Model	Reasoning
Gradient Boosting	XG Boost was choosen as the final optimized model due to its high predictive accuracy and effieciency.Its built-in regularization helped prevent overfitting,while its ability to handle missing values simplified preprocessing.Additionally,XGBoost provides valuable insights into feature importance,enhancing interpretability and model refinement to align with project objectives justifying it as a final model.

6. Results

6.1 Outputs screenshots

Output for there will be rain tomorrow



Output for there will be no rain tomorrow

7. Advantages & Disadvantages

Advantages

1. **High Accuracy:**
   Utilizes advanced machine learning algorithms to provide precise traffic volume predictions, improving planning and decision-making.
2. **Real-Time Insights:**
   Offers dynamic updates based on live data, helping users avoid congestion and optimize travel routes.
3. **User-Friendly Interface:**
   The intuitive web and mobile platforms make it easy for commuters and planners to access critical information quickly.
4. **Scalability:**
   Designed to be adaptable across multiple cities and urban settings, making it suitable for diverse traffic conditions.
5. **Environmental Benefits:**
   Helps reduce fuel consumption and emissions by promoting efficient routing and minimizing congestion.

Disadvantages

1. **Data Dependency:**
   The accuracy of predictions relies heavily on the availability and quality of real-time data from various sources.
2. **Computationally Intensive:**
   Training and maintaining machine learning models can require significant computational resources and time.
3. **Complexity:**
   The system may face challenges in model interpretability, making it difficult for non-experts to understand predictions.
4. **Ongoing Maintenance:**
   Continuous updates and monitoring are necessary to ensure model accuracy and relevance as traffic patterns evolve.
5. **Initial Setup Costs:**
   Implementing the necessary infrastructure and technology can involve high upfront costs for cities or organizations.

8. Conclusion

The **TrafficTelligence** project stands as a testament to the significant advancements that machine learning can bring to urban traffic management. As cities worldwide grapple with the increasing complexity of transportation systems, characterized by rising populations and vehicle counts, the need for innovative solutions has never been more pressing. TrafficTelligence effectively addresses these challenges by providing a sophisticated framework for predicting traffic volume in real time, thereby enhancing the overall efficiency of urban mobility.

At the core of TrafficTelligence is the integration of diverse data sources—traffic sensors, weather information, and public event schedules—allowing the system to generate accurate predictions that reflect current conditions. This real-time responsiveness empowers commuters to make informed decisions about their travel routes, significantly reducing time spent in traffic and alleviating frustration. For city planners and traffic authorities, the insights derived from TrafficTelligence facilitate proactive traffic management. By optimizing traffic signals and planning detours in anticipation of congestion, urban planners can enhance the flow of traffic, reduce bottlenecks, and improve the overall travel experience.

Moreover, TrafficTelligence prioritizes sustainability, addressing one of the most pressing issues of our time. By minimizing vehicle idle times and promoting efficient routing, the system contributes to lower fuel consumption and reduced greenhouse gas emissions. This aligns with global efforts to create more sustainable urban environments, where transportation systems are designed not only for efficiency but also with a commitment to environmental stewardship.

The scalability and adaptability of TrafficTelligence further enhance its utility. The system is designed to be implemented across various urban settings, each with its unique traffic patterns and infrastructure. This adaptability ensures that the solution remains relevant and effective as cities evolve and face new challenges. Additionally, ongoing integration with IoT devices and continuous updates will keep the model attuned to real-time changes in traffic dynamics, ensuring sustained accuracy over time.

In conclusion, TrafficTelligence represents a forward-thinking approach to urban traffic management, marrying cutting-edge technology with practical applications to enhance the efficiency, sustainability, and user-friendliness of transportation systems. As cities continue to grow and evolve, solutions like TrafficTelligence will be instrumental in creating smart, connected urban environments that prioritize both mobility and the well-being of their residents. By paving the way for better traffic management and improved commuter experiences, TrafficTelligence not only addresses current challenges but also sets the stage for a more sustainable and efficient future in urban transportation..

9. Future Scope

The future scope of **TrafficTelligence** is vast and multifaceted, with opportunities for significant enhancements and expansions that can further optimize urban traffic management and improve overall mobility. Key areas for future development include:

1. **Integration with IoT Devices:**
   Expanding the system to include Internet of Things (IoT) devices, such as smart traffic lights and connected vehicles, will enable real-time data exchange and more responsive traffic management.
2. **Geographic Expansion:**
   Adapting the TrafficTelligence system for implementation in various cities and regions will allow for the customization of models to account for local traffic patterns and infrastructure..
3. **Enhanced Predictive Analytics:**
   Incorporating advanced analytics capabilities, such as machine learning techniques for anomaly detection and trend forecasting, will provide deeper insights into traffic behaviors.
4. **User-Centric Applications:**
   Developing mobile applications that offer personalized traffic alerts, route suggestions, and estimated travel times based on user preferences will enhance the commuter experience.
5. **Climate Impact Studies:**
   Conducting studies on the effects of traffic patterns on urban air quality and climate change can help cities understand the broader implications of traffic management
6. **Collaboration with Urban Planning Initiatives:**
   Partnering with urban planners and local governments to inform infrastructure development will ensure that TrafficTelligence aligns with city growth strategies..
7. **Machine Learning Model Enhancement:**
   Continuously refining machine learning models with more extensive and diverse datasets will improve prediction accuracy. This includes incorporating feedback loops where user data and traffic outcomes inform model adjustments, resulting in a more resilient system.
8. **Automated Traffic Management Systems:**
   Future iterations of TrafficTelligence could evolve into fully automated traffic management systems that adjust traffic signals, reroute vehicles, and optimize public transit schedules without human intervention.

10. Appendix

10.1 Source Code

Trafficelligence.ipynb

#IMPORTING NECESSARY LIBRARIES

```
import pandas as pd
import numpy as np
import seaborn as sns
import sklearn as sk
from sklearn import linear_model
from sklearn import tree
from sklearn import ensemble
from sklearn import svm
import xgboost
from collections import Counter
import matplotlib.pyplot as plt

#CHANGE THE PATH TO YOUR FILE

data=pd.read_csv("traffic volume.csv")
```

Now the 'data' variable contains the content of the CSV file located at the provided path

```
data.head()
data.describe()
data.info()
data.isnull().sum()
data['temp'].fillna(data['temp'].mean(),inplace=True)
data['rain'].fillna(data['rain'].mean(),inplace=True)
data['snow'].fillna(data['snow'].mean(),inplace=True)
print(Counter(data['weather']))
data['weather'].fillna('clouds',inplace=True)
```

#splitting the date column into year,month,day

```
data[["day", "month", "year"]] =data["date"].str.split("-
",expand = True)
```

#splitting the date column into year,month,day

```
data[["hours", "minutes", "seconds"]] =
data["Time"].str.split(":", expand = True)
from sklearn.preprocessing import LabelEncoder
```

Assuming you have 'x' as your independent variables and 'y' as your dependent variable

```
le=LabelEncoder()
data['weather']=le.fit_transform(data['weather'])
data['holiday']=le.fit_transform(data['holiday'])
data['date']=le.fit_transform(data['date'])
data['Time']=le.fit_transform(data['Time'])

cor = data.corr()
sns.heatmap(cor)
sns.pairplot(data)
data.boxplot()

data.drop(columns = ["date", "Time"],axis=1,inplace = True)
data.head()
y=data['traffic_volume']
x=data.drop(columns=['traffic_volume'],axis=1)
x.shape
y.shape
names=x.columns
from sklearn.preprocessing import scale
data['holiday']=le.fit_transform(data['holiday'])
x = scale(x)
x=pd.DataFrame(x,columns=names)
x.head()
from sklearn.model_selection import train_test_split
```

Split the dataset into training and test sets

```
x_train,x_test,y_train,y_test =
train_test_split(x,y,test_size=0.2,random_state=0)
from sklearn import linear_model
from sklearn import tree
from sklearn import ensemble
from sklearn import svm
import xgboost
lin_reg=linear_model.LinearRegression()
Dtree=tree.DecisionTreeRegressor()
```

```
Rand=ensemble.RandomForestRegressor()
svr=svm.SVR()
XGB=xgboost.XGBRegressor()

lin_reg.fit(x_train,y_train)
Dtree.fit(x_train,y_train)
Rand.fit(x_train,y_train)
svr.fit(x_train,y_train)
XGB.fit(x_train,y_train)

p1=lin_reg.predict(x_train)
p2=Dtree.predict(x_train)
p3=Rand.predict(x_train)
p4=svr.predict(x_train)
p5=XGB.predict(x_train)

from sklearn import metrics
print(metrics.r2_score(p1,y_train))
print(metrics.r2_score(p2,y_train))
print(metrics.r2_score(p3,y_train))
print(metrics.r2_score(p4,y_train))
print(metrics.r2_score(p5,y_train))

p1=lin_reg.predict(x_test)
p2=Dtree.predict(x_test)
p3=Rand.predict(x_test)
p4=svr.predict(x_test)
p5=XGB.predict(x_test)

print(metrics.r2_score(p1,y_test))
print(metrics.r2_score(p2,y_test))
print(metrics.r2_score(p3,y_test))
print(metrics.r2_score(p4,y_test))
print(metrics.r2_score(p5,y_test))

MSE=metrics.mean_squared_error(p3,y_test)
np.sqrt(MSE)

import pickle
pickle.dump(Rand,open("model.pk1",'wb'))
pickle.dump(le,open("encoder.pk1",'wb'))
```

app.py

app.py

```
import numpy as np
import pickle
import matplotlib.pyplot as plt
import time
import pandas
import os
from flask import Flask, request, jsonify, render_template

app = Flask(__name__)

# Load model and encoder

with open(r"C:\Users\DELL\Downloads\Bootslander
(1)\Bootslander\model.pk1", 'rb') as file:
    model = pickle.load(file)

with open(r"C:\Users\DELL\Downloads\Bootslander
(1)\Bootslander\encoder.pk1", 'rb') as file:
    scale = pickle.load(file)
print("Model's feature names:", model.feature_names_in_)

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/predict', methods=["POST", "GET"])
def predict():

  #  reading the inputs given by the user

    input_feature=[float(x) for x in request.form.values()
]
    features_values=[np.array(input_feature)]
    names = [['holiday','temp', 'rain', 'snow', 'weather',
'year', 'month', 'day','hours', 'minutes', 'seconds']]
    data = pandas.DataFrame(features_values,columns=names)

   # predictions using the loaded model file

    prediction=model.predict(data)
```

```
    print(prediction)
    text = "Estimated Traffic Volume is :"
    return render_template("output.html",result = text +
str(prediction) + "units")

  # showing the prediction results in a UI

if __name__ == "__main__":
    print("Model's feature names:", model.feature_names_in_)
    port = int(os.environ.get('PORT', 5000))
    app.run(port=port, debug=True, use_reloader=False)
```

Index.html

```
<!DOCTYPE html>
<html >
<head>
  <meta charset="UTF-8">
  <title>Traffic Volume Estimation</title>
</head>


<body background="https://cdn.vox-
cdn.com/thumbor/voARJfEKvTp6iMSzW3ExPn06TDM=/0x78:3000x1766/1
600x900/cdn.vox-
cdn.com/uploads/chorus_image/image/44219366/72499026.0.0.jpg"
text="black">

 <div class="login">
      <center><h1>Traffic Volume Estimation</h1></center>

     <!-- Main Input For Receiving Query to our ML -->
    <form action="{{ url_for('predict')}}"method="post">
<h1>Please enter the following details</h1>


</style></head>


  <label for="holiday">holiday:</label>
        <select id="holiday" name="holiday">
            <option value=7>None</option>
            <option value=1>Columbus Day</option>
            <option value=10>Veterans Day</option>
```

```
            <option value=9>Thanksgiving Day</option>
            <option value=0>Christmas Day</option>
            <option value=6>New Years Day</option>
            <option value=11>Washingtons Birthday</option>
            <option value=5>Memorial Day</option>
            <option value=2>Independence Day</option>
            <option value=8>State Fair</option>
            <option value=3>Labor Day</option>
            <option value=4>Martin Luther King Jr
Day</option>


        </select>   <br>


<br>   <label>temp:</label>
      <input type="number"   name="temp" placeholder="temp  "
required="required" /><br>

 <br>
       <label>rain:</label>
      <input type="number"  min="0" max="1"  name="rain     "
placeholder="rain" required="required" /><br>

<br>
       <label>snow:</label>
      <input type="number" min="0" max="1"   name="snow     "
placeholder="snow     " required="required" /><br>


<br>


      <label for="weather">weather:</label>
        <select id="weather" name="weather">
            <option value=1>Clouds</option>
            <option value=0>Clear</option>
            <option value=6>Rain</option>
            <option value=2>Drizzle</option>
            <option value=5>Mist</option>
            <option value=4>Haze</option>
            <option value=3>Fog</option>
            <option value=10>Thunderstorm</option>
            <option value=8>Snow</option>
            <option value=9>Squall</option>
```

```
            <option value=7>Smoke</option><


        </select>   <br>
<br>
       <label>year:</label>
      <input type="number" min="2012"
max="2022"   name="year     " placeholder="year     "
required="required" /><br>

<br>
            <label>month:</label>
      <input type="number" min="1" max="12"   name="month   "
placeholder="month    " required="required" /><br>

<br>
           <label>day:</label>
      <input type="number" min="1" max="31"   name="day     "
placeholder="day      " required="required" /><br>

<br>
       <label>hours:</label>
      <input type="number" min="0" max="24"   name="hours   "
placeholder="hours    " required="required" /><br>

<br>
            <label>minutes:</label>
      <input type="number" min="0" max="60"   name="minutes "
placeholder="minutes  " required="required" /><br>

<br>
       <label>seconds:</label>
      <input type="number" min="0" max="60"   name="seconds "
placeholder="seconds  " required="required" /><br>

<br>

<br><br>


<button type="submit" class="btn btn-primary btn-block btn-
large" style="height:30px;width:200px">Predict</button>

    </form>
<br>
```

```
   {{ prediction_text }}


  <br>
  <br>
  <img src="data:image/png;base64,{{url_3}}" alt="Submit
Form" height="180" width="233"
onerror="this.style.display='none'"/>
  <img src="data:image/png;base64,{{url_1}}" alt="Submit
Form" height="180" width="233"
onerror="this.style.display='none'"/>
  <img src="data:image/png;base64,{{url_4}}" alt="Submit
Form" height="180" width="233"
onerror="this.style.display='none'"/>
  <br>
  <br>
  <img src="data:image/png;base64,{{url_2}}" alt="Submit
Form" height="150" width="711"
onerror="this.style.display='none'"/>

 </div>


</body>
</html>
```

OUTPUT.HTML

```
<!DOCTYPE html>
<html>
<head>
<title>Home</title>
<style>
body
{
    background-image: url("https://stat.overdrive.in/wp-
content/uploads/2021/10/2021-jaguar-xf-facelift-india-
01.jpg");
    background-size: cover;
}
.pd{
padding-bottom:45%;}
}
```

```html
</style>
</head>
<body>

<br>
<center><b class="pd"><font color="black" size="15" font-
family="Comic Sans MS" >Traffic volume
estimation</font></b></center><br><br>
<div>
<br>
<center>
<p><font color="black"> {{result}} </p>
</center>
</div>
</body>
</html>
```

1.1 GitHub & Project Demo Link

https://github.com/Jhansiiiii/TrafficTelligence-Advanced-Traffic-Volume-Estimation-with-Machine-Learning