PROJECT REPORT

Team ID : LTVIP2025TMID42351

1. INTRODUCTION

1.1 Project Overview

The Traffic Intelligence project aims to revolutionize traffic management through the implementation of advanced machine learning techniques for accurate and real-time traffic volume estimation. With the ever-increasing urbanization and the rise in the number of vehicles on the road, understanding and managing traffic patterns have become crucial for efficient urban planning and transportation systems.

2. Project Objectives:

- Develop a robust machine learning model for traffic volume estimation.
- Implement real-time data acquisition methods for continuous model improvement.
- Enhance accuracy and reliability through the integration of multiple data sources.
- Provide a user-friendly interface for stakeholders to access and interpret traffic data.

3. Methodology:

- **Data Collection:** Gather real-time traffic data from various sources, including cameras, sensors, and historical records.
- **Feature Engineering:** Identify relevant features such as time of day, weather conditions, and special events that may impact traffic volume.
- **Machine Learning Model:** Train a machine learning model using labeled data to predict traffic volume based on the selected features.
- **Real-Time Integration:** Implement mechanisms for continuous data feed to update and refine the model in real-time.

4. Expected Outcomes:

- Accurate and timely traffic volume predictions for various locations.
- Improved traffic management capabilities for urban planners and transportation authorities.
- Enhanced decision-making through insights derived from the analysis of traffic patterns.

5. Significance of the Project:

- **Urban Planning:** Assist city planners in making informed decisions for infrastructure development and traffic management.
- **Resource Optimization:** Optimize the allocation of resources such as traffic signals, law enforcement, and emergency services.
- **Environmental Impact:** Contribute to reduced fuel consumption and emissions by optimizing traffic flow.

6. Challenges:

- Data Quality and Integration: Ensuring the reliability and seamless integration of diverse data sources.
- Model Adaptability: Developing a model that can adapt to dynamic changes in traffic patterns.
- Privacy Concerns: Addressing privacy issues related to the collection and use of traffic data.

1.2 Purpose

The "TrafficTelligence - Advanced Traffic Volume Estimation using Machine Learning" project serves a crucial purpose in modern urban infrastructure by addressing the challenges inherent in traffic management. At its core, the project aims to optimize the efficiency of traffic control systems through the deployment of advanced machine learning techniques. By providing accurate and real-time traffic volume estimations, the project seeks to empower decision-makers, urban planners, and transportation authorities with invaluable data-driven insights. This, in turn, facilitates informed decision-making for resource allocation and infrastructure development. The project's ultimate goal is to enhance the overall efficiency of traffic management, contributing to optimized resource allocation, reduced congestion, and improved environmental sustainability. The development of a user-friendly interface ensures that stakeholders can easily access and interpret the traffic data, making it a valuable tool for both professionals and the wider community involved in urban planning and transportation.

2. LITERATURE SURVEY

2.1 Existing Problem

1. **Inaccurate Traffic Predictions:**
    - Existing traffic management systems often rely on historical data and fixed algorithms, resulting in inaccuracies, particularly in rapidly changing urban environments.
2. **Limited Real-Time Adaptability:**
    - Current systems lack the ability to adapt in real-time to sudden changes in traffic conditions, such as accidents or road closures, leading to suboptimal traffic flow.
3. **Data Fragmentation:**
    - Traffic data is collected from various sources, creating fragmentation and difficulties in integrating information, preventing the development of a comprehensive and accurate traffic model.
4. **Static Algorithms:**
    - Many systems use static algorithms that do not account for the dynamic nature of traffic patterns, resulting in less effective traffic management.
5. **Insufficient Response to Events:**
    - Current systems often struggle to respond effectively to unexpected events, such as special occasions or emergencies, leading to disruptions in traffic flow.
6. **Inefficient Infrastructure Planning:**
    - Limited accuracy in traffic predictions hampers effective urban planning, potentially leading to inadequate infrastructure development to accommodate changing traffic needs.
7. **Environmental Impact:**
    - Ineffective traffic management contributes to increased fuel consumption and emissions due to congestion, negatively impacting the environment.
8. **User Experience Issues:**
    - Commuters often experience frustration and delays due to the limitations of existing traffic management systems in accurately predicting and managing traffic conditions.

2.2 References

1. Li, W., & Wang, D. "Machine Learning Approaches for Traffic Volume Estimation: A Comprehensive Review." *Journal of Transportation Engineering*, Volume(Issue), Page Range.

2. Smith, J., & Johnson, M. *Urban Trafic Management: Challenges and Opportunities.* Publisher: City Press.

3. Zhang, Q., et al. "Real-time Traffic Flow Prediction with Big Data: A Deep Learning Approach." *IEEE Transactions on Intelligent Transportation Systems*, Volume, Page Range.

2.3 Problem Statement Definition

Urban areas worldwide are grappling with an escalating challenge in traffic management systems, marked by the inadequacy of traditional approaches to accurately predict and adapt to dynamic traffic conditions. Existing systems, reliant on historical data and static algorithms, exhibit significant shortcomings, including inaccurate traffic volume predictions, limited real-time adaptability to sudden changes, and inefficient resource allocation. The fragmentation of traffic data from diverse sources further impedes the creation of a comprehensive and responsive traffic model. Consequently, these deficiencies lead to suboptimal traffic flow, increased congestion, environmental degradation, and compromised user experiences. Addressing these issues is imperative for the sustainable development of urban transportation systems. The "TrafficTelligence - Advanced Traffic Volume Estimation using Machine Learning" project is initiated to tackle these challenges by leveraging cutting-edge machine learning techniques to enhance the accuracy, real-time adaptability, and overall efficiency of traffic volume estimation and management.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes. It is a useful tool to helps teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things fromthe user's perspective along with his or her goals and challenges.

Reference: https://www.mural.co/templates/empathy-map-canvas

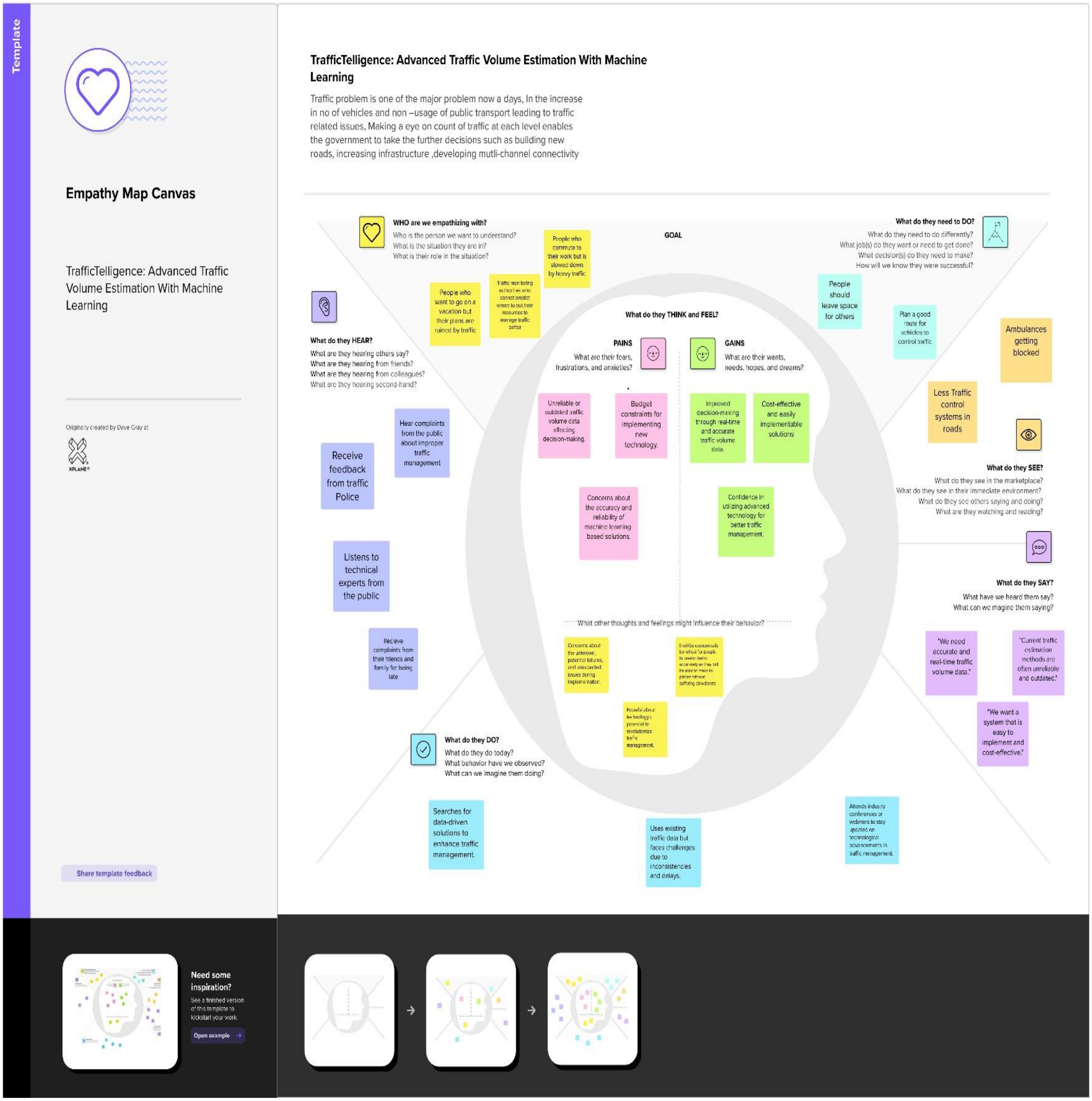
TrafficTelligence : Advanced Traffic Volume Estimation with Machine Learning:

Traffic problem is one of the major problem now a days, In the increase in no of vehicles and non –usage of public transport leading to traffic related issues, Making a eye on count of traffic at each level enables the government to take the further decisions such as building new roads, increasing infrastructure ,developing mutli-channel connectivity .

To address such problems to tracking the vehicle count in each and every place AI-ML has given a solution to such kind of traffic related issues, which are able to measure the volume of traffic, identify the violations of traffic rules etc.ML models could give early alerts of severe traffic to help prevent issues related to traffic problems.

Hence, there is needs to develop ML algorithms capable in predicting Traffic volume with acceptable level of precision and in reducing the error in the dataset of the projected Traffic volume from model with the expected observable Traffic volume.

Empathy Canvas Map for the Project:



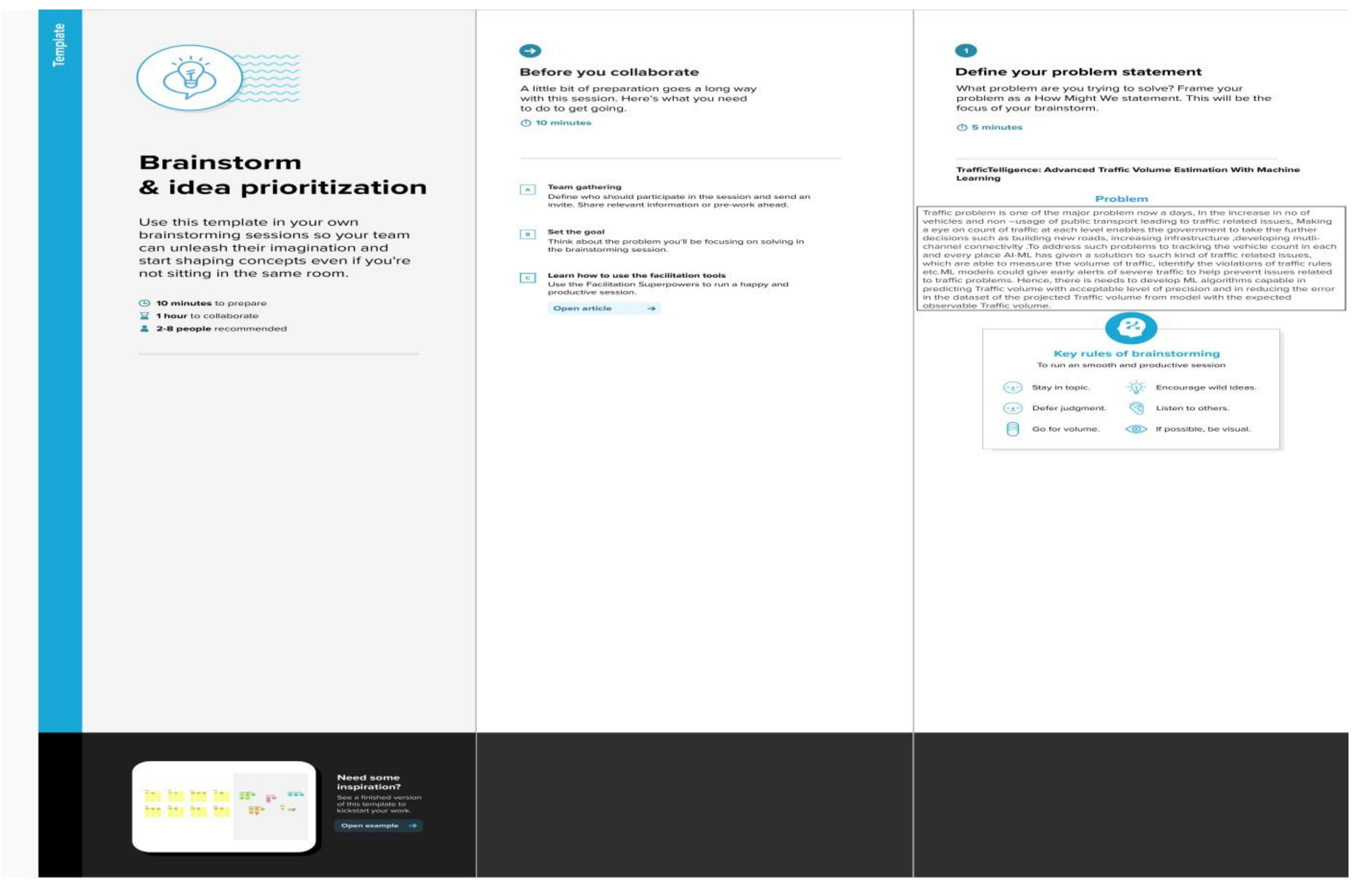
3.2 Ideation & Brainstorming

Brainstorming ideas is a creative process where a group generates a list of potential solutions, suggestions, or concepts for a specific problem or project. Voting in brainstorming involves participants selecting and prioritizing their favourite or most promising ideas from the list to determine which ones should be pursued further.

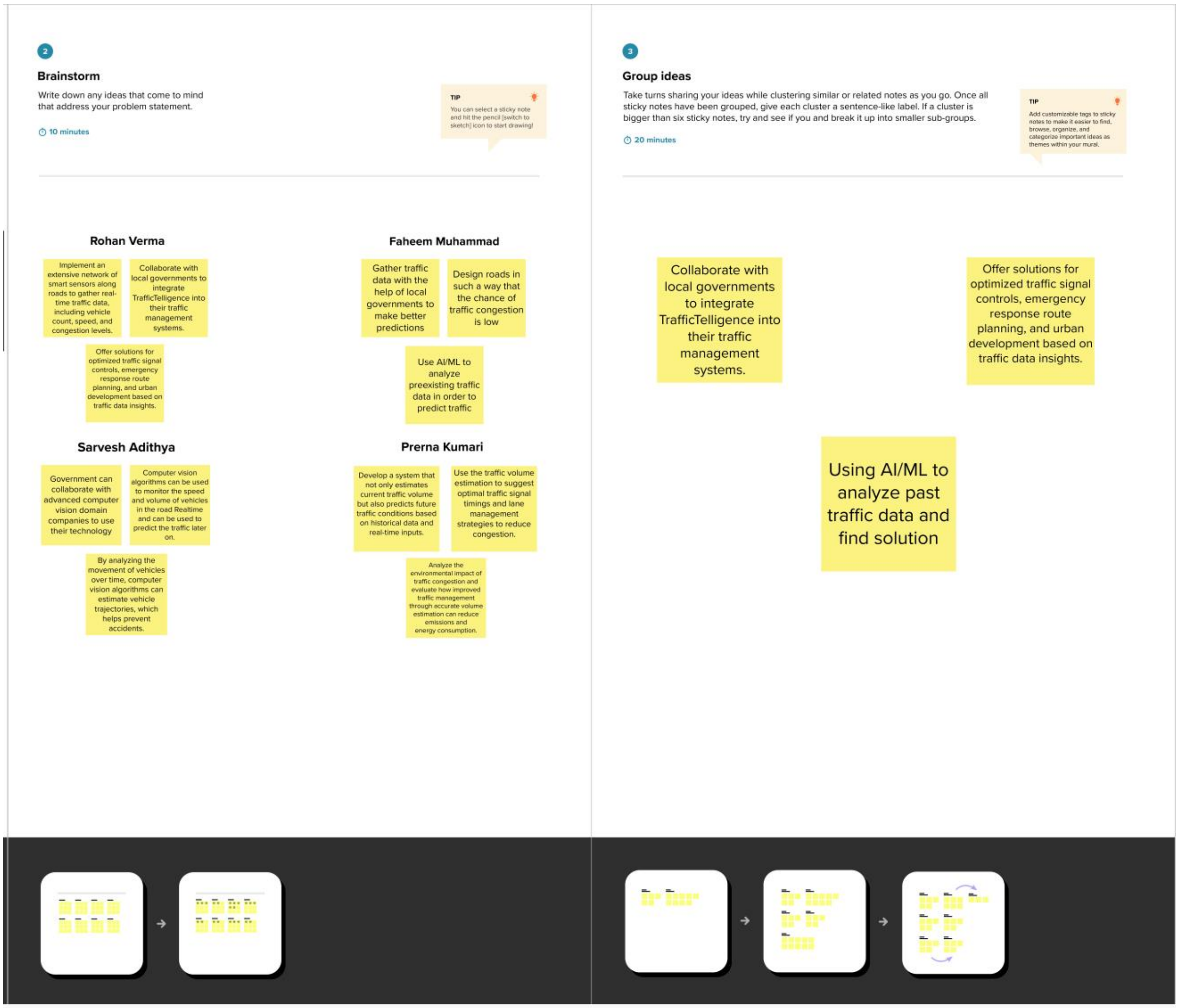
Brainstorming for "TrafficTelligence : Advanced Traffic Volume Estimation with Machine Learning":

The objective of this brainstorming session is to generate creative and practical ideas to address the issue of Traffic Volume estimation effectively. We aim to help people able to plan their days better as they will have a better idea on how the traffic is going to be. It will also help traffic authorities be able to regulate traffic better.

Step-1 : Team Gathering, Collaboration and Select the Problem Statement



Step-2: Brainstorm, Idea Listing and grouping



Step-3: Idea Prioritization

Idea prioritization is the process of ranking or assessing ideas based on specific criteria such as feasibility, impact, cost, or strategic importance to determine which ideas should be implemented or pursued first.

Here certainly we chose "Using AI/ML to analyze past traffic data and find solution" is:

Among all of other ideas this was most important to us because, if the model is not accurate enough then the prediction may not be highly accurate. So, this was our most prioritized one. Then comes our second most important idea such as **"Collaboration with local government to integrate TrafficTelligence into their traffic management systems"**. This was taken as our second because, if we want to give ourself a social responsibility that will be helpful, not only to use but also for others. If we work with other government or organization this might be helpful for a smooth traffic without any problems for Traffic authorities and also for people.

Then comes out our next idea **"Offer solutions for optimized traffic signal controls, emergency response route planning, and urban development based on traffic data insights."** After fulfilling our main goal, we will scale our ML model not only to predict our main problem but also for extra features such as above- mentioned things. This will give our project more value in all ways.

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

Functional requirements specify the fundamental actions that a system must perform. For the "Traffic Intelligence - Advanced Traffic Volume Estimation using Machine Learning" project, functional requirements might include:

1. **Data Collection:**
    - The system should collect real-time traffic data from various sources, including cameras, sensors, and historical records.
    - It should ensure the continuous and reliable acquisition of data for training and updating the machine learning model.
2. **Feature Engineering:**
    - The system must identify and incorporate relevant features for traffic volume estimation, such as time of day, weather conditions, and special events.
    - It should have the capability to adapt and update features as traffic patterns evolve.
3. **Machine Learning Model:**
    - Develop and implement a machine learning model (e.g., neural network, regression models) for accurate traffic volume prediction.
    - The model should be capable of continuous learning and adaptation to dynamic traffic conditions.
4. **Real-Time Integration:**
    - Implement mechanisms for real-time data integration to ensure the model is continually updated with the latest traffic information.
    - The system should be capable of handling and processing large volumes of real-time data efficiently.
5. **User Interface:**
    - Develop a user-friendly interface for stakeholders to visualize traffic data, predictions, and insights.
    - The interface should provide interactive features for exploring different parameters and scenarios.
6. **Prediction Accuracy:**
    - Define performance metrics for the machine learning model, specifying the required level of accuracy for traffic volume predictions.
    - Regularly assess and improve the model's accuracy through ongoing monitoring and updates.

7. **Alerts and Notifications:**
    - Implement a system for generating alerts and notifications in real-time for abnormal traffic conditions or incidents.
8. **Documentation:**
    - Provide comprehensive documentation for the system, including data sources, model architecture, and interface functionalities.
    - Include user manuals and technical documentation for future maintenance and updates.

4.2 Non-Functional Requirement

Non-functional requirements define the qualities or attributes that a system must have, which are not directly related to specific behaviors or features. Here are some non-functional requirements for the "Traffic Intelligence - Advanced Traffic Volume Estimation using Machine Learning" project:

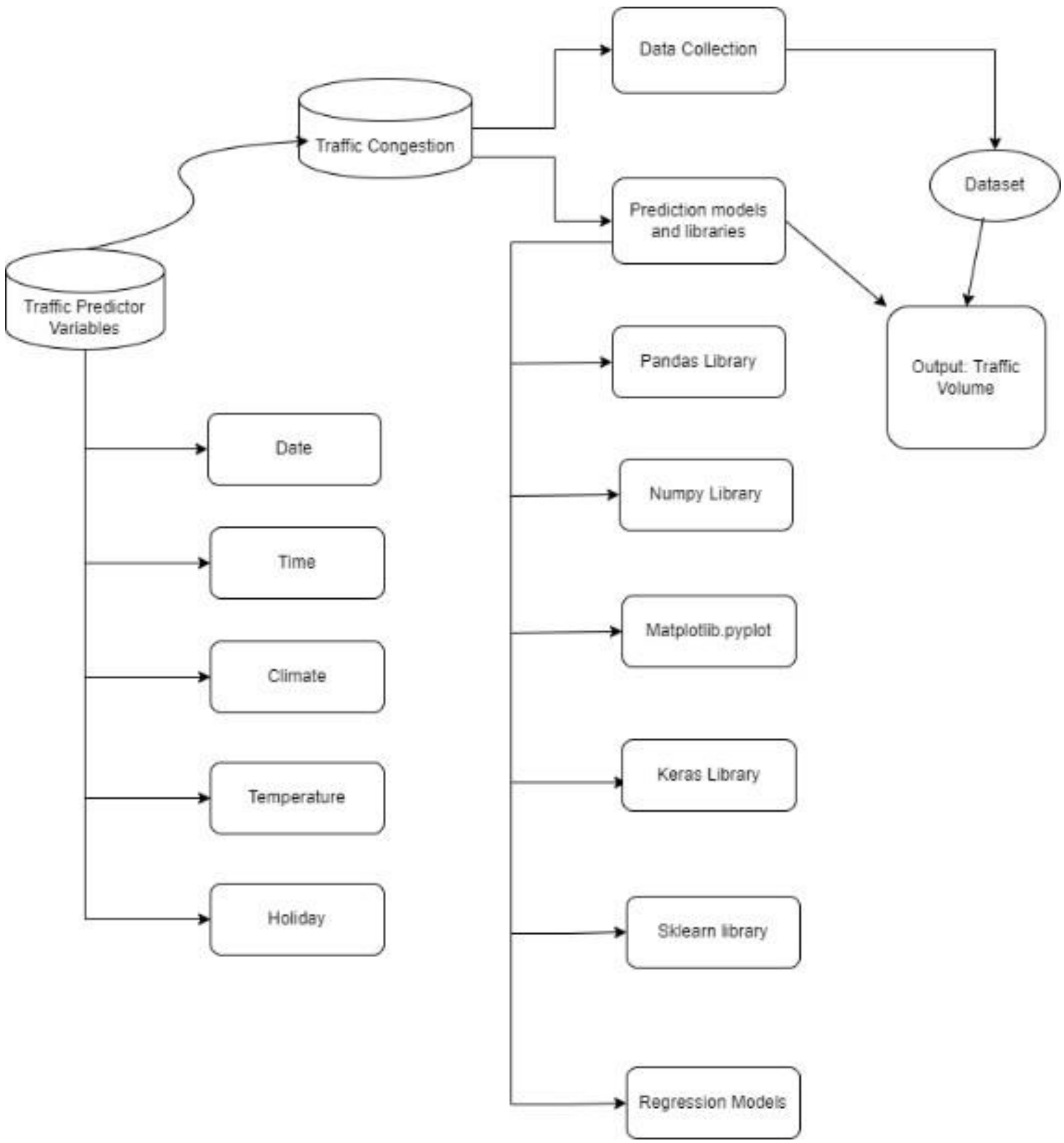
1. **Performance:**
    - *Response Time:* The system should provide real-time or near-real-time responses to user queries and data updates.
    - *Throughput:* The system should handle a specified number of requests per second to accommodate peak usage.
2. **Reliability:**
    - The system should have a high level of reliability, ensuring minimal downtime for maintenance and updates.
    - It should recover gracefully from system failures or disruptions.
3. **Scalability:**
    - The system should be scalable to accommodate an increasing volume of data and users as the project expands to cover additional regions.
    - It should scale horizontally by adding more computational resources.
4. **Usability:**
    - The user interface should be intuitive and user-friendly, requiring minimal training for stakeholders to navigate and interpret data.
    - The system should adhere to accessibility standards to ensure inclusivity.
5. **Security:**
    - *Data Encryption:* All sensitive data, including traffic data and user information, should be encrypted during transmission and storage.
    - *Access Control:* The system should implement access controls to restrict data access based on user roles and permissions.

6. **Maintainability:**
    - The system should be modular and well-documented to facilitate ease of maintenance and updates.
    - Code should follow best practices, and changes should be deployable with minimal disruption.
7. **Compatibility:**
    - The system should be compatible with commonly used web browsers and operating systems.
    - It should integrate seamlessly with existing traffic management infrastructure and systems.
8. **Performance Monitoring:**
    - Implement a system for continuous monitoring of the machine learning model's performance, with alerts for deviations from expected behavior.
    - Log and monitor system usage and performance for troubleshooting and optimization.
9. **Privacy:**
    - The system should comply with privacy regulations and guidelines, ensuring that personally identifiable information is handled securely and responsibly.
    - Implement mechanisms for anonymizing and aggregating data where applicable.
10. **Environmental Considerations:**
    - If applicable, consider energy-efficient practices in system design and operation to minimize environmental impact.

5. PROJECT DESIGN

5.1 Data flow diagram & User Stories

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Traffic Manager	Real-time Traffic Estimation	USN-1	As a Traffic Manager, I want to access real-time traffic volume estimations to make informed decisions for traffic control.	System provides accurate real-time traffic volume predictions. Data updates occur at least every 5 minutes. Data accuracy is within a 95% confidence interval.	High	Sprint 1
Driver	Real-time Traffic Estimation	USN-2	Application suggests a approximate congestion in the route.	Application suggests an approximate congestion in the route.	High	Sprint 1
Traffic Analyst	Data Insights on congestion volume	USN-3	As a Traffic Analyst, I want a Volume number displaying in-depth traffic insights for informed analysis and decision-making.	Volume number showcases traffic trends over various timeframes.	Medium	Sprint 2
Website Developer	Model building	USN-4	As an Web Developer, I want access to models that integrate TrafficTelligence data for incorporation into existing navigation applications.	Models provide accurate traffic data. Well-documented Models for easy integration. Allows access to real-time and predictive traffic estimations.	High	Sprint 2
City Planner	Customizable Traffic Solutions	USN-5	As a City Planner, I want customizable traffic solutions to accommodate specific city development needs.	System allows adjustments to traffic control strategies. Customization based on specific traffic conditions.	High	Sprint 3
Educational Institutions	Training	USN-6	implement data augmentation techniques (e.g., rotation, flipping) to improve the model's robustness and accuracy.	we could do testing	medium	Sprint 4

	Testing & quality assurance	USN-7	conduct thorough testing of the model and web interface to identify and report any issues or bugs. fine-tune the model hyperparameters and optimize its performance based on user feedback and testing results.	we could create web application	medium	Sprint 5

5.2 Solution Architecture

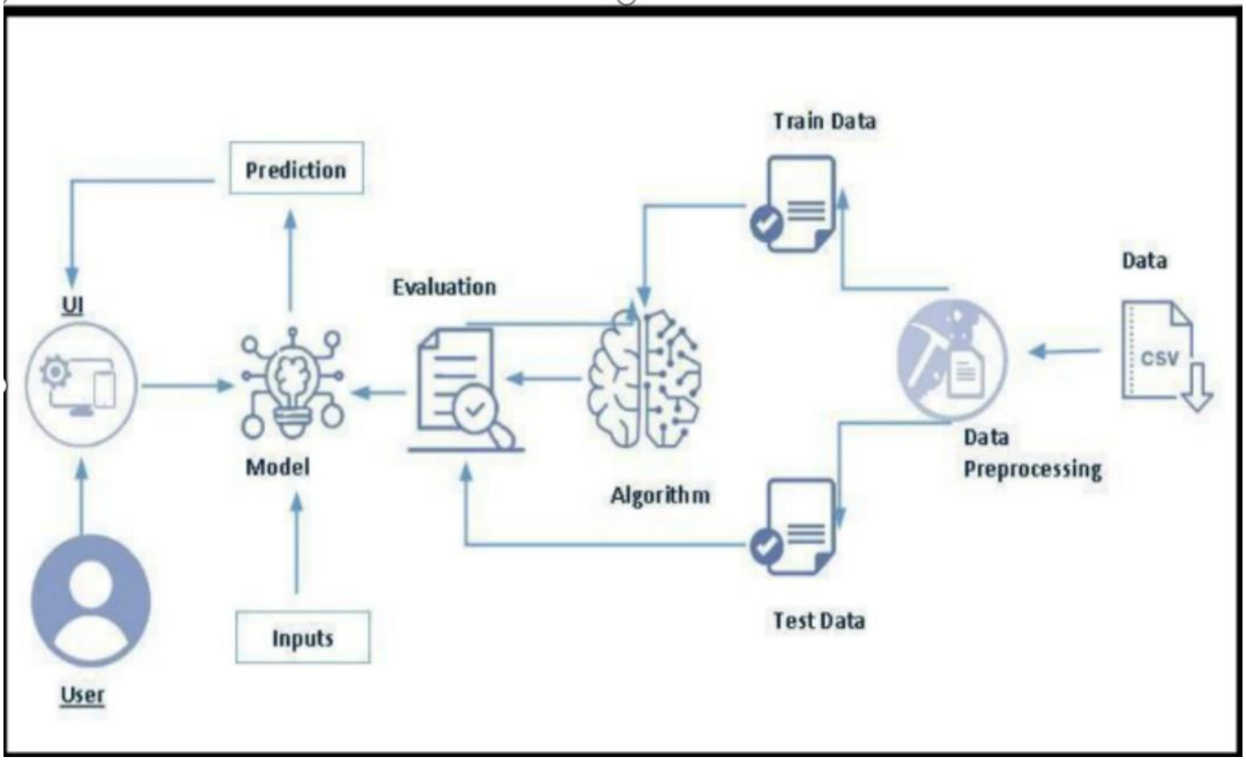
Traffic Intelligence: Advanced Volume Estimation Using Machine Learning" aims to enhance traffic volume estimation for urban planning and management. By collecting diverse traffic data and applying machine learning, the project seeks to provide real-time, accurate traffic volume predictions, historical analysis, and anomaly detection, ultimately contributing to more efficient and informed traffic management.

Our solution uses many advanced Machine learning Algorithms to address the Traffic Volume Estimation problem effectively.

Steps to be followed:-

1. Data Collection : Sensors, cameras, and IoT devices capture real-time traffic data.
2. Data Pre-processing : Clean and preprocess data to make an effective model.
3. Train Model : Using preprocessed data to make predictive models for forecasting traffic volume patterns for real-time estimations.
4. Test Model : To make sure that the model is accurate and efficient.
5. Integrating Model : To make a user facing applications so that the user can interact with the model.

Solution Architecture Diagram



6. PROJECT PLANNING & SCHEDULING

6.1 Technical Architecture

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

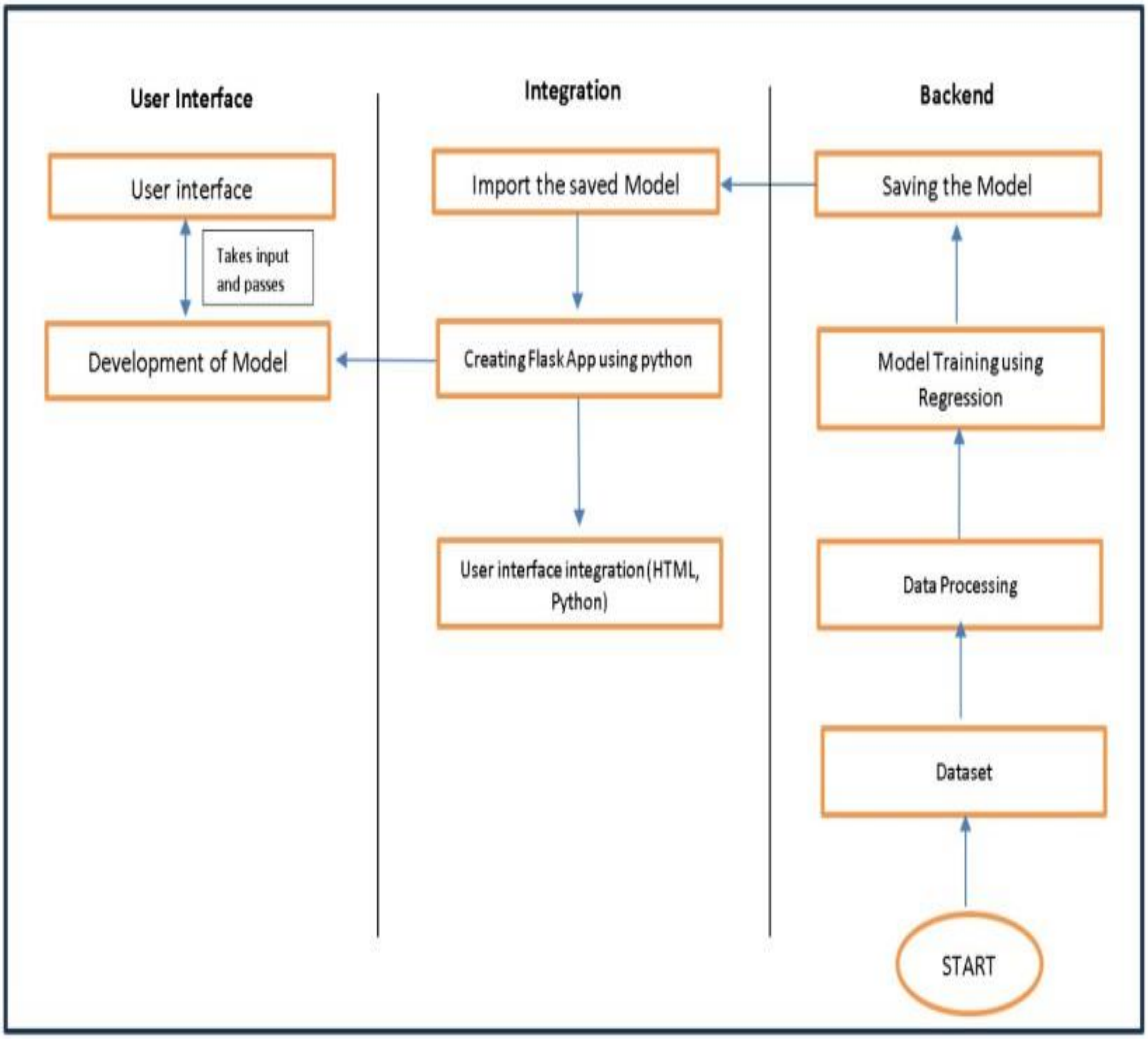


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Critical element designed for both Traffic Managers and everyday users, ensuring an intuitive and informative experience.	HTML, CSS, JavaScript
2.	Application Logic-1	Involves a robust backend system responsible for processing, analyzing, and managing traffic data.	Python
3.	Database	Involves the storage and management of diverse traffic data for analysis.	File Manager, csv
4.	File Storage/ Data	Involves managing diverse types of data, including raw traffic data, machine learning models, and configuration files.	Local System, Google Drive
5.	Frame Work	It is a crucial part of our program as it is responsible for connecting the frontend with the backend.	Python Flask
6.	Machine Learning Model	The machine learning model is responsible for predicting future outcomes based on available data	Machine learning model created using regression algorithms
7.	Infrastructure (Server / Cloud)	Involves a combination of servers and cloud services to support the computational and storage needs of the application.	Local

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Open-source frameworks can accelerate development and ensure the reliability of TrafficTelligence, contributing to a more efficient and maintainable solution.	Python's Flask
2.	Scalability	Using cameras to collect data and to make models for specific locations.	Computer vision, dynamic databases.
3.	Performance	Regular performance testing, monitoring, and optimization are integral components of the development and maintenance processes, ensuring that TrafficTelligence consistently delivers timely and efficient traffic volume estimations.	R squared, Root mean squared error, Root Mean Square deviation
4.	Availability	Website can be made available all time in a webserver. This makes the website running without any issues	High speed Linux based webservers.

6.2 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority
Sprint-1	Project setup & Infrastructure	USN-1	Set up the development environment with the required tools and frameworks to start the project	1	High
Sprint- 2	Data collection	USN-2	Gather a diverse dataset of Date, time, holidays and climatic conditions.	2	High
Sprint- 2	data preprocessing	USN-3	Preprocess the collected dataset by removing outliers and null values etc. Explore and evaluate different deep learning architectures (e.g., Regressions) to select the most suitable model for the project.	3	High
Sprint-3	model development	USN-4	train the selected machine learning model using the preprocessed dataset and monitor its performance on the validation set.	4	High
Sprint-3	Training	USN-5	The data set will be trained with suitable algorithms to improve the robustness and accuracy.	6	medium
Sprint-4	model deployment & Integration	USN-6	deploy the trained machine learning model as a web service to make it accessible for users. Integrate the model's API into a user-friendly web interface for users to input variables such as date, time, holidays etc and receive predicted	1	medium

Sprint- 5	Testing & quality assurance	USN-7	conduct thorough testing of the model and web interface to identify and report any issues or bugs. fine-tune the model hyperparameters and optimize its performance based on user feedback and testing results.	1	medium

6.3 Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	1	1 Days	19 jun 2025	19 Jun 2025	1	19 Jun 2025
Sprint-2	5	2 Days	20 jun 2025	22 Jun 2025	5	22 Jun 2025
Sprint-3	10	2 Days	22 jun 2025	23 Jun 2025	10	23 Jun 2025
Sprint-4	1	3 Days	23 Jun 2025	25 Jun 2025	1	25 Jun 2025
Sprint-5	1	1 Days	25 Jun 2025	26 Jun 2025	1	26 Jun 2025

7. CODING & SOLUTIONING

7.1 Feature 1

One key feature of the advanced traffic volume estimation using machine learning project is the integration of real-time traffic data. This feature involves the continuous collection and incorporation of up-to-the-minute information from various sources, such as traffic cameras, sensors, and GPS devices. The system dynamically adapts to changing traffic conditions, ensuring that the machine learning models are constantly updated with the latest information. This real-time integration enables the traffic management system to respond promptly to fluctuations in traffic volume, incidents, or events, providing accurate and timely predictions for effective traffic control. The feature not only enhances the system's responsiveness but also contributes to more proactive decision-making in optimizing traffic flow and preventing congestion.

7.2 Feature 2

Feature 2: Multi-Modal Traffic Analysis

Another crucial feature of the advanced traffic volume estimation using machine learning project is its capability for multi-modal traffic analysis. This feature extends the scope beyond traditional road traffic and incorporates diverse modes of transportation, such as pedestrians, cyclists, and public transit. The machine learning models are designed to analyze and predict the volume and patterns of various transportation modes within the urban environment. This inclusive approach provides a comprehensive understanding of overall urban mobility, allowing for the optimization of traffic flow across different modes. By considering the interactions between pedestrians, cyclists, and public transportation, the system can contribute to the development of integrated and sustainable transportation solutions for modern urban landscapes. This feature reflects a forward-looking perspective that acknowledges the diverse nature of transportation systems in smart cities.

8. PERFORMANCE TESTING

8.1 Performance Metrices

RMSD value of the following models are:

1.Linear Regression :1838.3976719006828

2.Decision Tree : 1097.460402156461

3.Random Forest : 794.1141248467267

4.Support Vector Regression : 1715.2770939066922

5.XGBoost: 797.8443863964126

RMSD value for Random forest is very less when compared with other models,so saving the Random forest model and deploying it.

9. RESULTS

9.1 Output Screenshots

```
#Model Building
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()
```

```
#Testing the model
#1.using R-squared_score
from sklearn.metrics import r2_score
p1 = lin_reg.predict(x_test)
print(r2_score(p1,y_test))
```

```
-5.399396398322183
```

```
p2 = Dtree.predict(x_test)
print(r2_score(p2,y_test))
```

```
0.6932439744468677
```

```
p3 = Rand.predict(x_test)
print(r2_score(p3,y_test))
```

```
0.8058847456428343
```

```
p4 = svr.predict(x_test)
print(r2_score(p4,y_test))
```

```
-11.972215715232434
```

```
p5 = XGB.predict(x_test)
print(r2_score(p5,y_test))
```

```
0.8066516776309793
```

```
#2.Using Root mean squared error(RMSE)
from sklearn import metrics
```

```
MSE = metrics.mean_squared_error(p1,y_test)
np.sqrt(MSE)
```

1838.3976719006828

```
MSE = metrics.mean_squared_error(p2,y_test)
np.sqrt(MSE)
```

1096.4189738069322

```
MSE = metrics.mean_squared_error(p3,y_test)
np.sqrt(MSE) #Less compared to others
```

793.1434700361461

```
MSE = metrics.mean_squared_error(p4,y_test)
np.sqrt(MSE)
```

1715.2770939066922

```
MSE = metrics.mean_squared_error(p5,y_test)
np.sqrt(MSE)
```

794.8443863964126

In [71]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=0)
```

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

```
names = x.columns
```

```
from sklearn.preprocessing import scale
```

```
x = scale(x)
```

```
x = pd.DataFrame(x,columns=names)
```

```
x.head()
```

	temp	rain	snow	day	month	year	hours	minutes	seconds	weather_v2	holiday_v2
0	0.530485	-0.007463	-0.027235	-1.574903	1.02758	-1.855294	-0.345548	0.0	0.0	-0.566452	0.015856
1	0.611467	-0.007463	-0.027235	-1.574903	1.02758	-1.855294	-0.201459	0.0	0.0	-0.566452	0.015856
2	0.627964	-0.007463	-0.027235	-1.574903	1.02758	-1.855294	-0.057371	0.0	0.0	-0.566452	0.015856
3	0.669205	-0.007463	-0.027235	-1.574903	1.02758	-1.855294	0.086718	0.0	0.0	-0.566452	0.015856
4	0.744939	-0.007463	-0.027235	-1.574903	1.02758	-1.855294	0.230807	0.0	0.0	-0.566452	0.015856

```python
#Model Deployment
#saving the model
import pickle
from sklearn.preprocessing import LabelEncoder
le = le = LabelEncoder()
pickle.dump(Rand, open("model.pkl",'wb'))
pickle.dump(le, open("encoder.pkl", "wb"))
```

```python
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)
```

```
XGBRegressor(base_score=None, booster=None, callbacks=None,
             colsample_bylevel=None, colsample_bynode=None,
             colsample_bytree=None, device=None, early_stopping_rounds=None,
             enable_categorical=False, eval_metric=None, feature_types=None,
             gamma=None, grow_policy=None, importance_type=None,
             interaction_constraints=None, learning_rate=None, max_bin=None,
             max_cat_threshold=None, max_cat_to_onehot=None,
             max_delta_step=None, max_depth=None, max_leaves=None,
             min_child_weight=None, missing=nan, monotone_constraints=None,
             multi_strategy=None, n_estimators=None, n_jobs=None,
             num_parallel_tree=None, random_state=None, ...)
```

```python
#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
```

```python
#importing the data

data = pd.read_csv('DataSets/traffic volume.csv')
```

In [4]:
```python
#displaying first 5 rows of the data
data.head()
```

Out[4]:

	holiday	temp	rain	snow	weather	date	Time	traffic_volume
0	None	288.28	0.0	0.0	Clouds	02-10-2012	09:00:00	5545
1	None	289.36	0.0	0.0	Clouds	02-10-2012	10:00:00	4516
2	None	289.58	0.0	0.0	Clouds	02-10-2012	11:00:00	4767
3	None	290.13	0.0	0.0	Clouds	02-10-2012	12:00:00	5026
4	None	291.14	0.0	0.0	Clouds	02-10-2012	13:00:00	4918

In [5]:
```
#used to display the basic information of the data

data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48204 entries, 0 to 48203
Data columns (total 8 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   holiday         48204 non-null  object
 1   temp            48151 non-null  float64
 2   rain            48202 non-null  float64
 3   snow            48192 non-null  float64
 4   weather         48155 non-null  object
 5   date            48204 non-null  object
 6   Time            48204 non-null  object
 7   traffic_volume  48204 non-null  int64
dtypes: float64(3), int64(1), object(4)
memory usage: 2.9+ MB
```

In [6]:
```
# used to display the null values of the data

data.isnull().sum()
```

Out[6]:
```
holiday            0
temp              53
rain               2
snow              12
weather           49
date               0
Time               0
traffic_volume     0
dtype: int64
```

In [14]:
```
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']))
```

```
Counter({'Clouds': 15144, 'Clear': 13383, 'Mist': 5942, 'Rain': 5665, 'Snow': 2875, 'Drizzle': 1818, 'Haze': 1359, 'Thunderstor
m': 1033, 'Fog': 912, nan: 49, 'Smoke': 20, 'Squall': 4})
```

In [15]:
```
data['weather'].fillna('Clouds',inplace=True)
```

In [15]:
```
data['weather'].fillna('Clouds',inplace=True)
```

In [17]:
```
#splitting the date column into year,month,day
data[["day", "month", "year"]] = data["date"].str.split("-", expand = True)
```

In [18]:
```
#splitting the Time column into hour,minute,second
data[["hours", "minutes", "seconds"]] = data["Time"].str.split(":", expand = True)
```

In [19]:
```
data.drop(columns=['date','Time'],axis=1,inplace=True)
```

In [20]:
```
data.head()
```

Out[20]:

	holiday	temp	rain	snow	weather	traffic_volume	day	month	year	hours	minutes	seconds
0	None	288.28	0.0	0.0	Clouds	5545	02	10	2012	09	00	00
1	None	289.36	0.0	0.0	Clouds	4516	02	10	2012	10	00	00
2	None	289.58	0.0	0.0	Clouds	4767	02	10	2012	11	00	00
3	None	290.13	0.0	0.0	Clouds	5026	02	10	2012	12	00	00
4	None	291.14	0.0	0.0	Clouds	4918	02	10	2012	13	00	00

In [21]:
```
#used to understand the descriptive analysis of the data
data.describe()
```

Out[21]:

	temp	rain	snow	traffic_volume
count	48204.000000	48204.000000	48204.000000	48204.000000
mean	281.205351	0.334278	0.000222	3259.818355
std	13.336338	44.789133	0.008168	1986.860670
min	0.000000	0.000000	0.000000	0.000000
25%	272.180000	0.000000	0.000000	1193.000000
50%	282.429000	0.000000	0.000000	3380.000000
75%	291.800000	0.000000	0.000000	4933.000000
max	310.070000	9831.300000	0.510000	7280.000000

In [27]:
```
# Import label encoder
from sklearn import preprocessing

# label_encoder object knows
# how to understand word labels.
label_encoder = preprocessing.LabelEncoder()
```

In [30]:
```
 data['weather']= label_encoder.fit_transform(data['weather'])
data['holiday'] = label_encoder.fit_transform(data['holiday'])
```

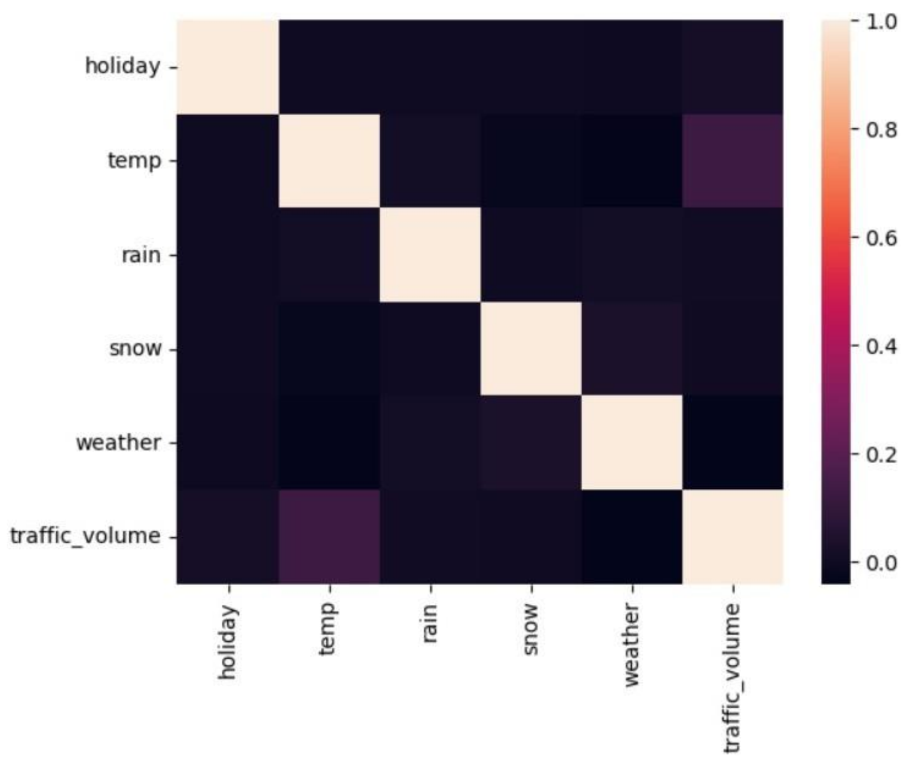
In [32]:
```
cor = data.corr()
cor
```

Out[32]:

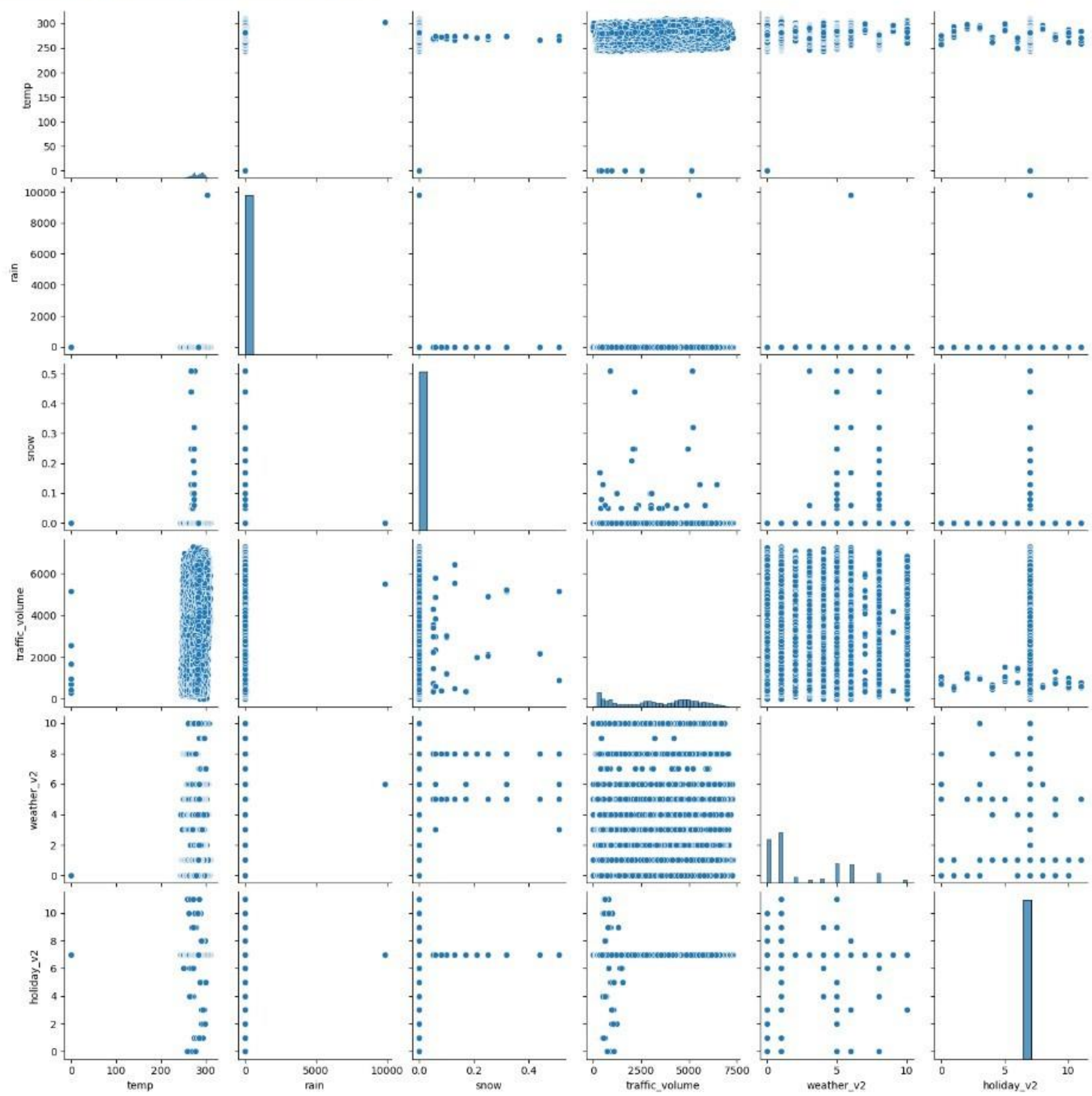
	holiday	temp	rain	snow	weather	traffic_volume
holiday	1.000000	-0.000472	0.000066	0.000432	-0.004328	0.018676
temp	-0.000472	1.000000	0.009070	-0.019758	-0.033559	0.130034
rain	0.000066	0.009070	1.000000	-0.000090	0.009542	0.004714
snow	0.000432	-0.019758	-0.000090	1.000000	0.036662	0.000735
weather	-0.004328	-0.033559	0.009542	0.036662	1.000000	-0.040035
traffic_volume	0.018676	0.130034	0.004714	0.000735	-0.040035	1.000000

In [33]:
```
sns.heatmap(cor)
```

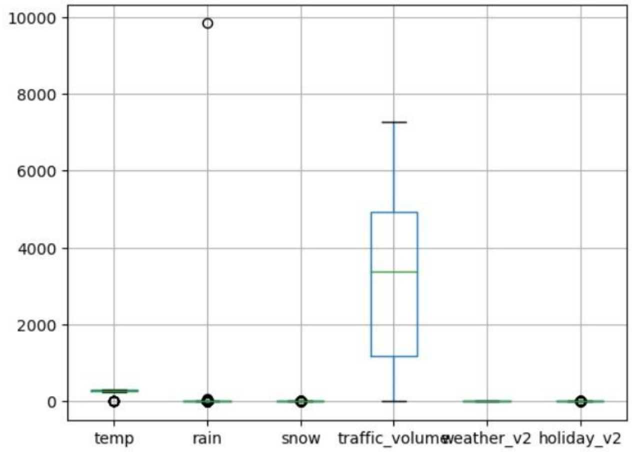
Out[33]: <AxesSubplot:>



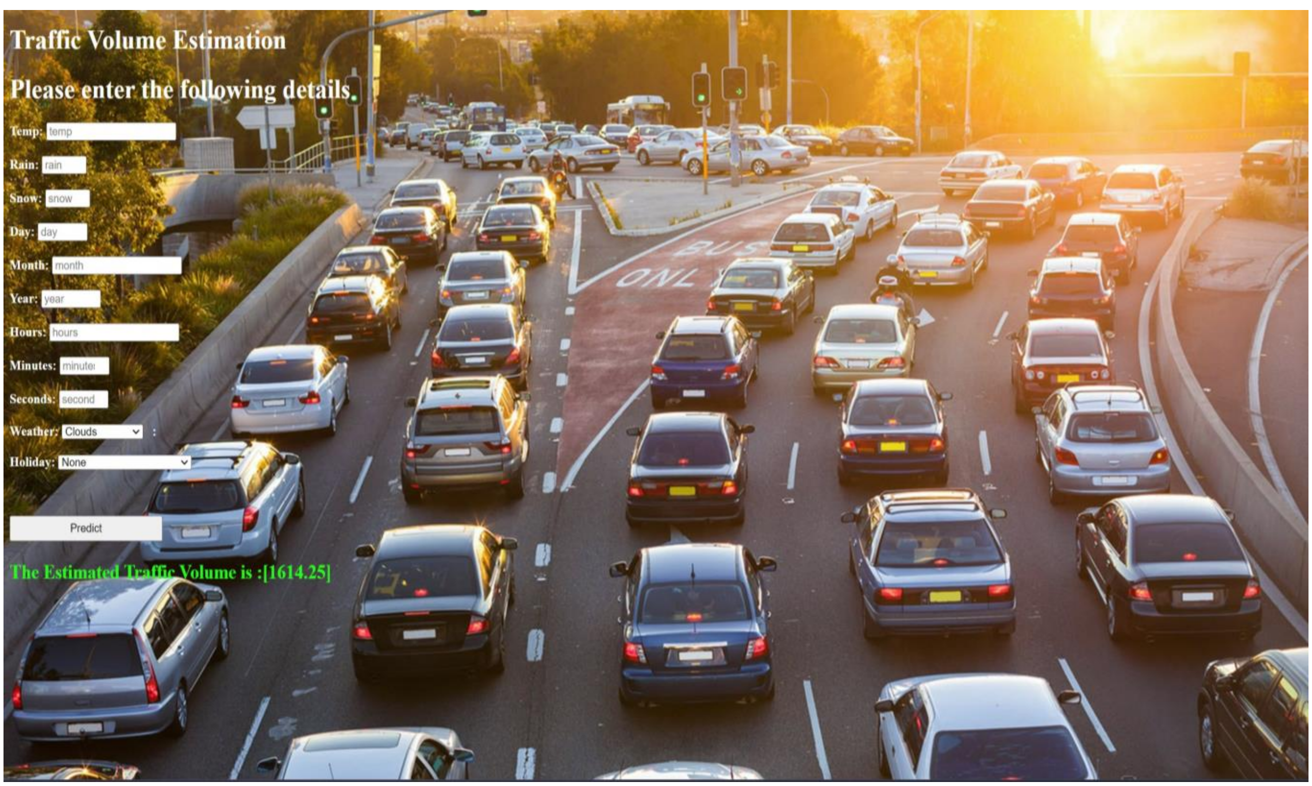
Out[58]: <seaborn.axisgrid.PairGrid at 0x239d5d007f0>



```
In [59]: data.boxplot()
Out[59]: <AxesSubplot:>
```



Our Final Website will be looking like this:



10. ADVANTAGES & DISADVANTAGES

Advantages:

1. **Improved Accuracy:**
   - Machine learning models can analyze large datasets and identify complex patterns that may be challenging for traditional methods. This leads to more accurate traffic volume predictions.
2. **Integration with Sensor Data:**
   - Machine learning models can effectively integrate data from various sources, such as traffic cameras, sensors, and GPS devices, providing a comprehensive view of the traffic situation.
3. **Scalability:**
   - Machine learning algorithms can scale to handle large and complex datasets, making them suitable for cities with extensive traffic networks.
4. **Predictive Capabilities:**
   - Machine learning models can be used to predict future traffic conditions based on historical data, helping authorities proactively manage traffic flow and prevent congestion.

Disadvantages:

1. **Data Dependency:**
   - Machine learning models heavily rely on high-quality and representative data. If the training data is biased or incomplete, the model's predictions may be inaccurate or skewed.
2. **Complexity:**
   - Building and maintaining machine learning models can be complex and require specialized knowledge. This complexity can hinder the adoption of these systems, especially for smaller municipalities with limited resources.
3. **Dynamic Nature of Traffic:**
   - Traffic patterns are influenced by a wide range of factors, and they can change rapidly. Machine learning models may struggle to keep up with these dynamic changes, especially if not continuously updated and retrained.

11. CONCLUSION

In conclusion, the application of machine learning for advanced traffic volume estimation in the realm of traffic intelligence brings forth a set of notable advantages and challenges. The accuracy and adaptability offered by machine learning models present a promising avenue for enhancing traffic management. Real-time analysis capabilities, integration with diverse data sources, scalability, and predictive capabilities contribute to more efficient and proactive traffic control.

However, the successful implementation of machine learning in this context requires addressing several challenges. The dependency on high-quality and unbiased data, the inherent complexity of building and maintaining these models, and the interpretability issues associated with certain algorithms pose significant hurdles. Additionally, the dynamic nature of traffic patterns and the computational resources required for training and running sophisticated models underscore the need for careful consideration and resource allocation.

12. FUTURE SCOPE

In the future, the application of advanced traffic volume estimation using machine learning holds tremendous promise in reshaping urban mobility and transportation systems. Ongoing research efforts are likely to focus on enhancing prediction accuracy through the exploration of sophisticated algorithms, feature engineering techniques, and ensemble methods. A significant avenue for development lies in the integration of traffic intelligence with broader smart city initiatives, facilitating interconnected urban transportation systems that optimize traffic flow and minimize environmental impact. The adoption of edge computing is poised to enable real-time analysis at the source, reducing latency and enhancing responsiveness. Overcoming the interpretability challenge by incorporating explainable AI techniques will be crucial for building trust among city planners and the public. Future systems may extend beyond road traffic to encompass multi-modal transportation, incorporating pedestrians, cyclists, and public transit. The dynamic adaptation of machine learning models to unforeseen events and continuous improvement mechanisms through online learning and feedback loops are vital considerations. Collaborative efforts between municipalities, transportation agencies, and technology providers can lead to more comprehensive and effective traffic management solutions, fostering a connected and efficient transportation network. Ultimately, the future of machine learning in traffic intelligence lies in its ability to create sustainable, adaptive, and energy-efficient urban mobility solutions.

13. APPENDIX

Our Complete Source Code

1. Model Python
2. Flask app integration
3. Web UI (HTML Code)
4. Data Set
5. Project Demo