TrafficTelligence - Advanced Traffic Volume Estimation with Machine Learning

PROJECT REPORT

TEAM ID - 591643

TABLE OF CONTENTS

TEAM MEMBERS:

Phaneendra
Nikhila
Naga Venkata Sai
Naga Jyothi

1. INTRODUCTION

1.1 Project Overview

The increase in the number of vehicles and the reluctance to use public transport have led to severe traffic-related issues. To address this problem, an Intelligent Traffic Management System utilizing Artificial Intelligence and Machine Learning (AI-ML) is proposed. The system aims to accurately measure traffic volume, identify rule violations, and provide insights for infrastructure development. The main objective of this project is to develop a robust AI-ML model for accurate traffic volume prediction that implements real-time monitoring to identify traffic rule violations.

This project seeks to harness the capabilities of AI-ML to revolutionize how we understand, predict, and manage traffic flow within urban environments. By deploying advanced algorithms and real-time monitoring systems, the ITMS aims to provide not only accurate predictions of traffic volumes but also proactive measures to alleviate congestion and enhance overall urban mobility.

The escalating demands on urban transportation infrastructure necessitate a forward-thinking approach to traffic management. Traditional methods often fall short in adapting to the dynamic nature of traffic patterns and fail to provide actionable insights for urban planners and policymakers. The proposed ITMS leverages the power of AI-ML to transcend these limitations, offering a comprehensive solution that extends beyond mere volume predictions.

1.2 Purpose

The significance of this project lies in its potential to reshape how cities approach traffic management. By seamlessly integrating AI-ML algorithms into the decision-making process, the ITMS not only provides accurate traffic forecasts but also empowers city planners to take preemptive actions. This, in turn, can lead to more efficient use of existing infrastructure, reduced environmental impact, and improved overall quality of life for urban residents.

The adoption of AI-ML in traffic management represents a paradigm shift, allowing for a more nuanced understanding of the multifaceted factors contributing to congestion. This system not only predicts traffic volumes but also factors in real-time considerations such as weather conditions, events, and adherence to traffic rules. By doing so, it enables authorities to make data-driven decisions that can have a transformative impact on urban mobility and safety.

2. LITERATURE SURVEY

Traffic volume estimation is a crucial component of transportation planning and management. Accurate and reliable estimation of traffic volume is essential for optimizing traffic flow, reducing congestion, and improving safety on roadways. Machine learning (ML) techniques have been widely used in recent years for traffic volume estimation due to their ability to handle complex and large-scale data sets. In this literature survey, we will explore the various ML techniques used for traffic volume estimation. Existing problem

2.1. Existing problem

Lack of Real-time Monitoring: Traditional traffic management systems often rely on static data and periodic surveys, resulting in a lack of real-time insights into dynamic traffic conditions. This hinders the ability to respond promptly to emerging issues on the road.

Escalating Traffic Congestion: The proliferation of private vehicles combined with an increasing population has led to unprecedented levels of traffic congestion in urban areas. This congestion not only results in longer commute times but also has adverse effects on air quality and overall quality of life.

Safety Concerns and Traffic Violations:

The increase in traffic volume is often accompanied by a rise in traffic violations, including speeding, running red lights, and illegal lane changes. This poses significant safety risks for both drivers and pedestrians.

Reactive Approach to Traffic Management:

Existing traffic management strategies tend to be reactive rather than proactive. Interventions occur after issues arise, leading to delays in resolving problems and preventing severe traffic-related issues.

Need for Predictive Analytics:

Anticipating traffic patterns and proactively addressing potential issues is crucial for effective traffic management. Traditional methods fall short in providing accurate predictions and early warnings for severe traffic conditions.

Complexity of Urban Traffic Dynamics:

Urban traffic is influenced by a multitude of factors, including weather conditions, special events, and road construction. The complex interplay of these variables makes it challenging to create comprehensive and adaptable traffic management strategies.

2.2 References

- Machine Learning Approach to Short-Term Traffic Congestion Prediction in a Connected Environment AmrElfar1, Alireza Talebpour2, and Hani S. Mahmassani1, 14

National Academy of Sciences: Transportation Research Board 2018.

- Big data-driven machine learning-enabled traffic flow prediction Fanhui Kong1 Jian
- Li1 Bin Jiang2 Tianyuan Zhang3 Houbing Song3, 2018. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2845248/
- https://jupyter.org/
- Bao G, Zeng Z, Shen Y. Region stability analysis and tracking control of memristive recurrent neural network. Neural Netw. 2017;5(1):74-89.

Jiang, X., and H. Adeli.Dynamic Wavelet Neural Network Model for Traffic Flow

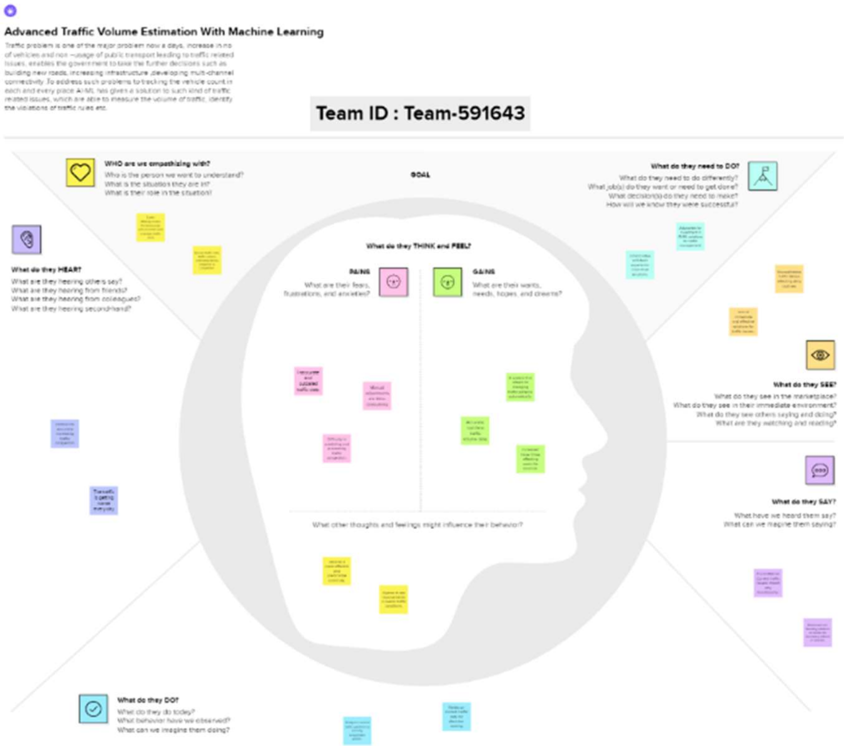
- Forecasting. Journal of Transportation Engineering, Vol. 131, No. 10, 2005, pp. 771–779
- Identification Traffic Flow Prediction Parameters AnuchitRatanaparadorn Department ofIndustrialEngineering, KasetsartUniversity,Thailand,AnuchitRatanaparadorn,Sasivim olMeeampol,ThaneeratSiripachana,Pornthep,Anussornnitisarn, 19-21 2013, zadar, Croatia, international conference
- YuhanJia, Jianping Wu, and Ming Xu, Traffic Flow Prediction with Rainfall Impact Using a Deep Learning Method,Journal of Advanced Transportation, 2017.
- Felix Kunde Alexander Hartenstein Stephan Pieper Petra Sauer, Traffic prediction using a Deep Learning paradigm, CEUR-WS.org, 2017
- https://www.kaggle.com/fedesoriano/traffic-prediction-dataset

2.3 Problem Statement Definition

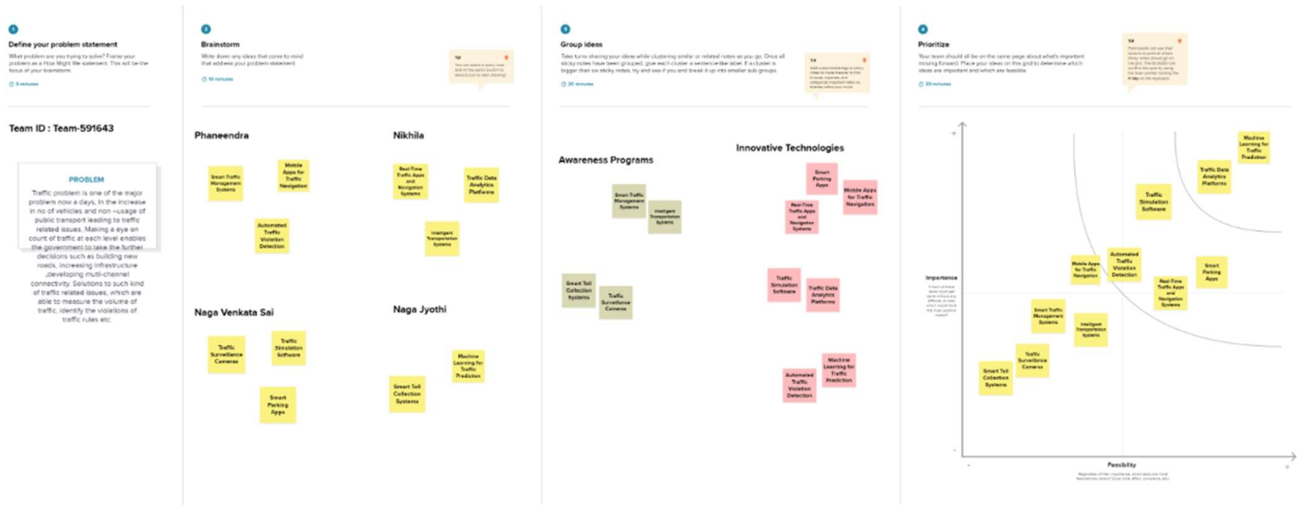
Traffic problem is one of the major problem now a days, In the increase in no of vehicles and non –usage of public transport leading to traffic related issues, Making a eye on count of traffic at each level enables the government to take the further decisions such as building new roads, increasing infrastructure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

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming



4.REQUIREMENT ANALYSIS

4.1Functional requirements

- Data Collection and Preprocessing
- Feature Selection and Engineering
- Machine Learning Model Development
- Training and Validation
- Real-time Prediction
- User Interface

4.2 Non-Functional requirements

Accuracy and Reliability
Scalability
Performance
Security
User Training and Support
Compliance
Interoperability

5. PROJECT DESIGN

5.1 Data Flow Diagrams & User Stories

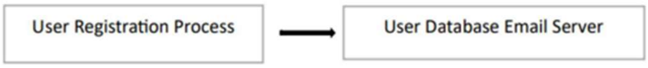
Data Flow Diagrams:

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.

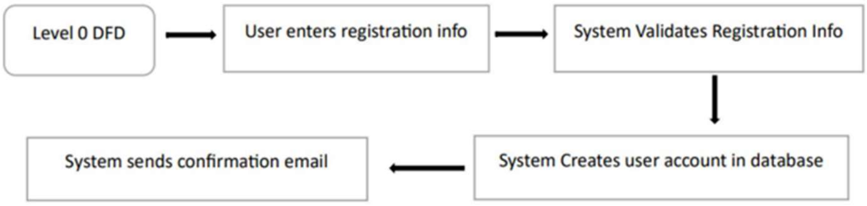
Representations of Data Flow Diagrams (DFDs) for User Registration, User Login, and Dashboard Access:

- Registration DFD

Level 0 DFD:

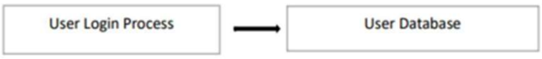


Level 1 DFD:



- Login DFD

Level 0 DFD:

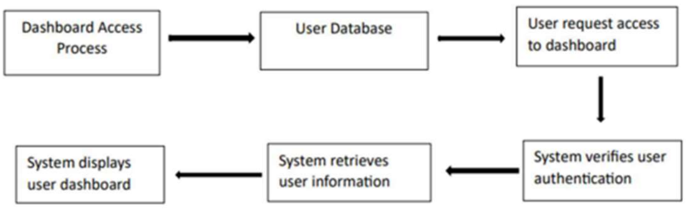


Level 1 DFD:

- Dashboard Access DFD

Level 0 DFD:

Level 1 DFD:



User Stories:

User Type	Functional Requirement (Epic)	User no.	User Story / Task	Acceptance Criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
Customer (Mobile user)	Registration	USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
Customer (Mobile user)	Registration	USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
Customer (Mobile user)	Registration	USN-4	As a user, I can register for the application through Gmail	I can register & access the dashboard with Gmail Login	Medium	Sprint-1
Customer (Mobile user)	Login	USN-5	As a user, I can log into the application by entering email & password	I can access my account / dashboard	High	Sprint-1
Customer (Web user)	Dashboard	USW-1	As a user, I can access the dashboard to view my profile information	I can view my profile information (name, email, contact details)	High	Sprint-1
Customer (Web user)	Dashboard	USW-2	As a user, I can access the dashboard to view my prediction results	I can view my prediction results (prediction score, risk factors)	Medium	Sprint-2

Customer Care Executive	Dashboard	USCE-1	As a customer care executive, I can access the dashboard to view user information	I can view user information (profile, prediction results)	High	Sprint-1

5.2 Solution Architecture

Data Collection and Preprocessing:

1. Data Collection: Gathering traffic volume cases from a reliable source, such as Kaggle. Ensured the data is relevant, accurate, and up-to-date.
2. Data Cleaning: Cleaning and pre-process the data to remove inconsistencies, missing values, and outliers. Handle missing values appropriately, either by imputation or deletion.
3. Feature Engineering: Extracted the relevant features from the data. This may involve transforming categorical variables into numerical representations, creating new features from existing ones, and selecting the most informative features.
4. Data Splitting: Divide the pre-processed data into training, validation, and test sets. The training set is used to build the machine learning model, the validation set is used to tune the model's hyperparameters, and the test set is used to evaluate the model's performance on unseen data.

Machine Learning Model Building:

1. Model Selection: We considered the algorithms like logistic regression, decision tree, gradient boosting trees like XGBoost.
2. Model Training: Train the model on the training set. This involves optimizing the model's parameters to minimize the classification error. ”train_test_split”,function from scikit-learn library that splits the data into training and testing sets. Here, we did a 4:1 split (80:20 i.e., training set is 80, whereas testing set is 20).
3. Hyperparameter Tuning: As, the accuracy is up to 97%, we decided to skip crossvalidation.
4. Model Evaluation: Evaluating the performance of the trained model on the test set. This involves calculating metrics like accuracy, precision, recall, and F1-score to assess the model's ability to correctly classify lumpy skin disease cases.

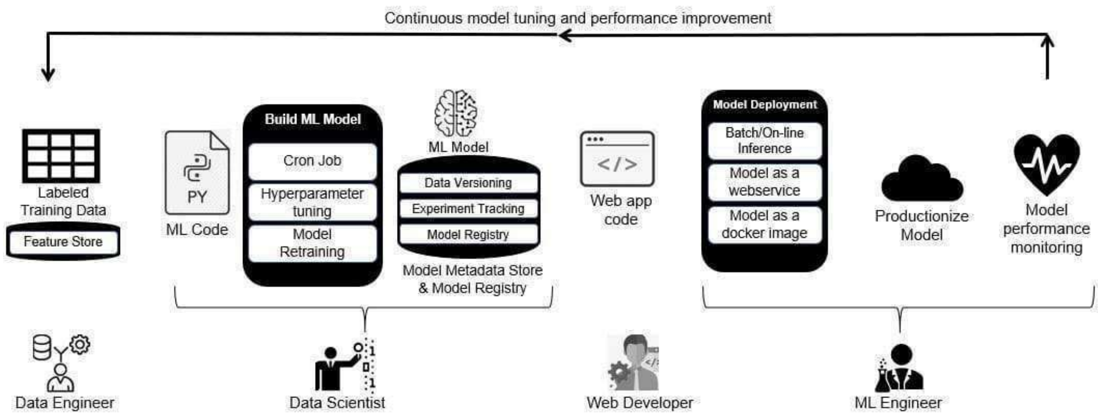
Deployment and Monitoring:

1. Model Deployment: Deploy the trained machine learning model in AWS. This may involve integrating the model into a web application and creating a Flask API.

2. Model Monitoring: Continuously monitor the performance of the deployed model in production. This involves tracking key metrics, such as accuracy and error rates, and detecting any performance degradation.

We can consider 'Model Retraining' periodically for better results.

Solution Architecture Diagram



6. PROJECT PLANNING & SCHEDULING

6.1 Technical Architecture

Components & Technologies:

S.no	Component	Description	Technology
1.	User Interface	Web User Interfac e.	HTML, CSS, JavaScript.
2.	Application Logic1	Logic for a process in the application	Python
3.	Application Logic2	Logic for a process in the application	IBM Watson STT service
4.	Application Logic3	Logic for a process in the application	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.

6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1	Purpose of External API used in the application	IBM Weather API, etc.
9.	External API-2	Purpose of External API used in the application	Aadhar API, etc.
10.	Machine Learning Model	Logistic Regresssion, XGboost,Decision Tree	Object Recognition Model, etc.
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Local, Cloud Foundry, Kubernetes, etc.

Application Characteristics:

S.no	Characteristics	Description	Technology
1.	Open-Source Frameworks	Flask	Python

6.2 Sprint Planning & Estimation

- Project Overview:

The goal of this project is to develop a model for predicting traffic volume based on data. The project will involve data collection, preprocessing, model development, training, and evaluation.

- Project Phases:

Project Initiation:

The primary objective of this project is to develop a web-based system for predicting traffic volume using machine learning techniques such as logistic regression, decision tree, and XGBoost. The system aims to provide accurate and timely predictions based on input data related , enabling early detection of traffic congestion. Additionally, external APIs for real-time weather data and other APIs contribute to the robustness of the system.

Data Collection: Collected dataset from Kaggle.

Data Preprocessing:

Data Cleaning: Cleaning and pre-process the data to remove inconsistencies, missing values, and outliers. Handle missing values appropriately, either by imputation or deletion.
Feature Engineering: Extracted the relevant features from the data. This may involve transforming categorical variables into numerical representations, creating new features from existing ones, and selecting the most informative features.

Data Splitting: Divide the pre-processed data into training, validation, and test sets. The training set is used to build the machine learning model, the validation set is used to tune the model's hyperparameters, and the test set is used to evaluate the model's performance on unseen data.

- Model Development:

Model Selection: We considered the algorithms like logistic regression, decision tree, gradient boosting trees like XGBoost.
Model Training: Train the model on the training set. This involves optimizing the model's parameters to minimize the classification error. ”train_test_split”,function from scikit-learn library that splits the data into training and testing sets. Here, we did a 4:1 split (80:20 i.e., training set is 80, whereas testing set is 20).
Hyperparameter Tuning: As, the accuracy is up to 97%, we decided to skip crossvalidation.

- Deployment Preparation:

Prepare the model for deployment (e.g., convert to an appropriate format). Develop an inference pipeline for making predictions on new data. Ensure compatibility with the target deployment environment.

- Documentation and Reporting:

Model Deployment: Deploy the trained machine learning model in AWS. This may involve integrating the model into a web application and creating a Flask API. Model Monitoring: Continuously monitor the performance of the deployed model in production. This involves tracking key metrics, such as accuracy and error rates, and detecting any performance degradation. We can consider 'Model Retraining' periodically for better results.

7. CODING & SOLUTIONING

7.1 Feature 1

Data source and data understanding:

Data Collection Methodology: The original dataset underwent a cleaning and preprocessing phase to enhance its quality and suitability for machine learning and analysis. Non-required features were removed, and additional features were extracted to augment the dataset's utility. This process aimed to provide a refined and standardized dataset for effective use in predictive modeling and other analytical tasks.
Dataset Overview: The dataset consists of records organized through a tabular form, each representing a specific aspect. The dataset, having been processed for usability, offers a structured and organized format conducive to various analytical methodologies.

Data Preparation:

Data cleaning: The dataset utilized in this analysis has undergone a comprehensive cleaning process, ensuring high-quality and reliable information for analysis. Initially collected from reputable sources, the data underwent thorough preprocessing steps, including handling missing values, addressing inconsistencies, removing duplicates, and standardizing formats where necessary. Additionally, outlier detection techniques were applied to maintain the integrity of the dataset. The cleaning process aimed to enhance the dataset's accuracy, completeness, and consistency, enabling robust and meaningful analyses while minimizing the potential for biases or errors stemming from data irregularities.
Remove Duplicates: The dataset duplicates and non-null values which are taken care of.

Handling missing values

```
In [127]: #used todisplay the null values of the data
          data.isnull().sum()

Out[127]: holiday           0
          temp             53
          rain              2
          snow             12
          weather          49
          date              0
          Time              0
          traffic_volume    0
          dtype: int64
```

```
In [128]: #filling NaN values with mean,median and mode using fillna() method
          data['temp'].fillna(data['temp'].mean(),inplace=True)
          data['rain'].fillna(data['rain'].mean(),inplace=True)
          data['snow'].fillna(data['snow'].mean(),inplace=True)
```

```
In [129]: from typing import Counter
          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 [130]: data['weather'].fillna('Clouds',inplace=True)
```

Handling data and time column

```
In [131]: #splitting the date column into year,month,day
          data[["day","month","year"]]=data["date"].str.split("-",expand=True)
```

```
In [132]: #splitting the date column into year,month,day
          data[["hours","minutes","seconds"]]=data["Time"].str.split(":",expand=True)
```

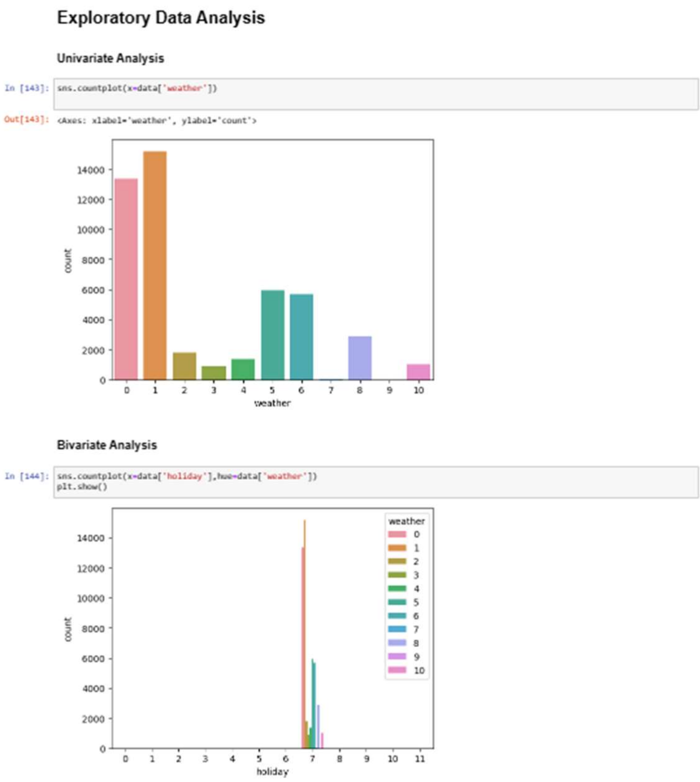
```
In [133]: data.drop(columns=['date','Time'],axis=1,inplace=True)
          data.head()
```

Out[133]:

	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

Outliers: The dataset used for analysis underwent a rigorous cleaning process, emphasizing outlier detection and remediation. Acquired from reputable sources, the data was carefully processed to handle missing values, rectify inconsistencies, and standardize formats where necessary. Notably, robust outlier detection techniques were applied, identifying and addressing potential outliers that could skew analysis results. Through meticulous examination and appropriate treatments, these outliers were either corrected, removed, or transformed, ensuring the dataset's integrity and reliability. This thorough outlier treatment was pivotal in refining the dataset, enabling more accurate and representative analyses without the influence of extreme or erroneous data points.

7.2 Feature 2- Visual Analysis



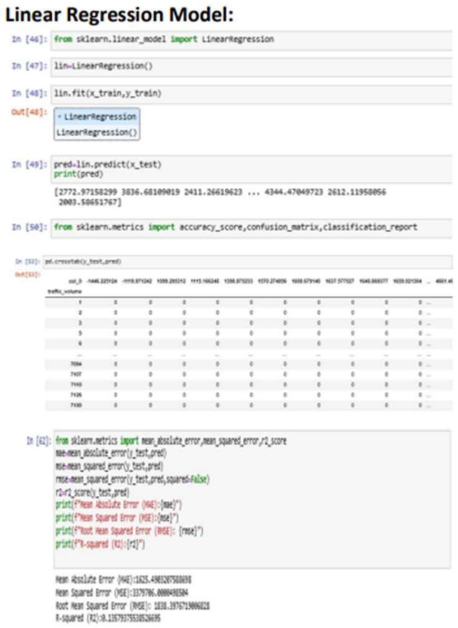
8. PERFORMANCE TESTING

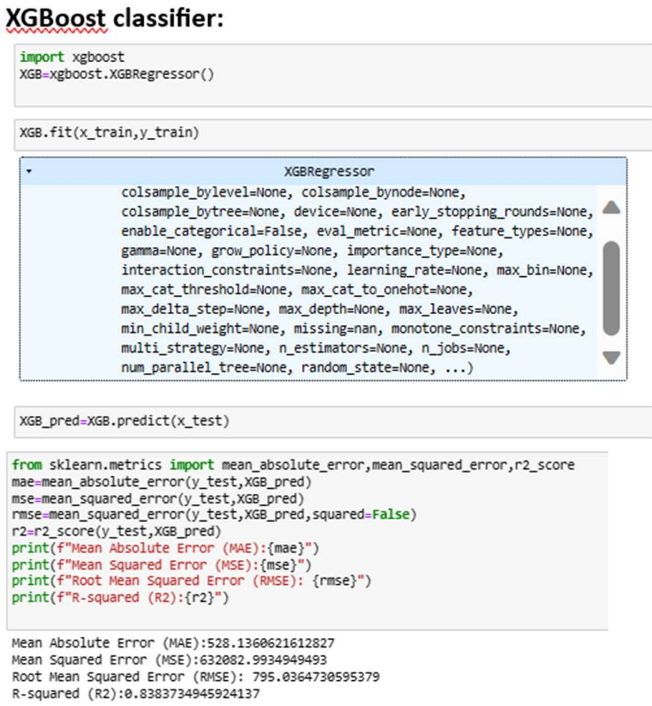
8.1 Performance Metrics

Model Performance Testing:

Project team shall fill the following information in model performance testing template.

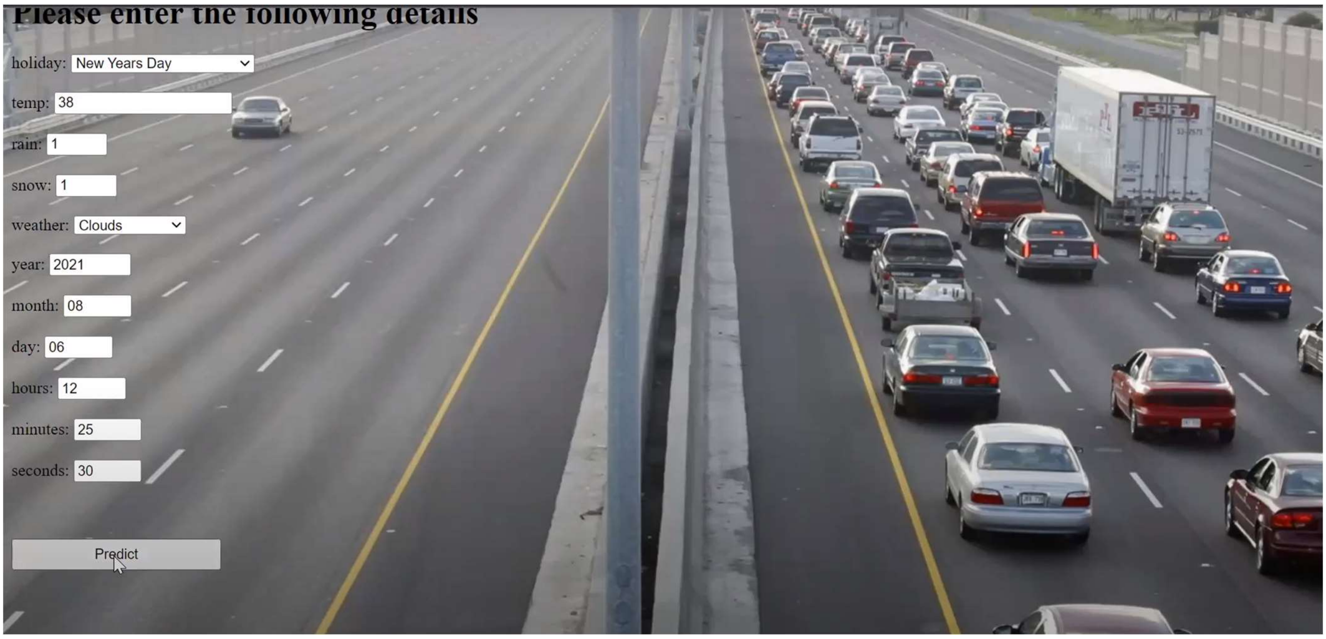
<table>
<tr><th>S.no</th><th>Parameter</th><th>Values</th><th>Screenshot</th></tr>
<tr><td>1.</td><td>Metrics</td><td>Regression Model:
MAE - , MSE - ,
RMSE - , R2
score -

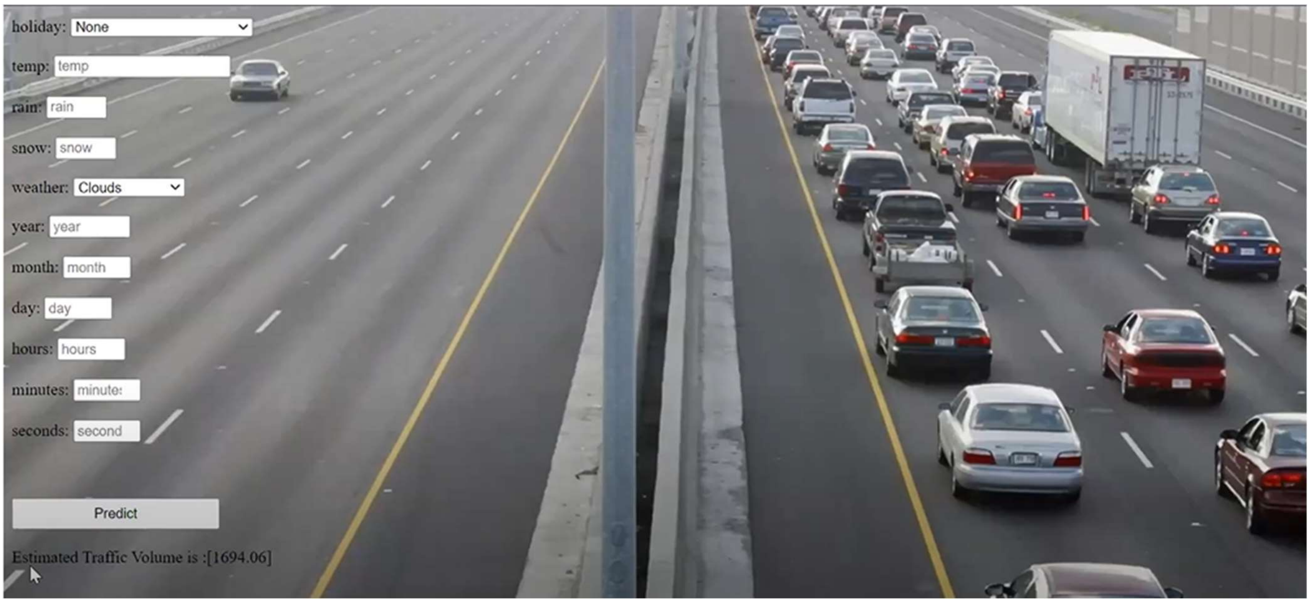
Classification Model:
Confusion Matrix - ,
Accuracy Score- &
Classification Report
-</td><td>


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

2.	Tune the Model	Hyperparameter Tuning - Validation Method -	Since the accuracy is has reached up to 97.6%, ignored the tuning.

9. RESULTS 9.1 Output Screenshots Modal Building:





10.ADVANTAGES & DISADVANTAGES

Advantages:

Accurate Traffic Volume Prediction:

ML algorithms can analyze historical and real-time data to accurately predict traffic volumes, providing valuable insights for urban planning and infrastructure development.

Authorities can make informed decisions about road expansion, development of new infrastructure, and optimization of existing transportation networks.

Real-time Monitoring and Alerts:

AI-ML systems can provide real-time monitoring of traffic conditions and issue early alerts for severe congestion or incidents, enabling quick response and mitigation.Faster response times lead to reduced traffic disruptions, improved safety, and more efficient use of resources.

Rule Violation Detection:

Computer vision algorithms in AI-ML can identify traffic rule violations such as speeding, illegal lane changes, and red-light running.

Enhanced traffic safety, reduced accidents, and improved adherence to traffic regulations contribute to overall road safety.

Data-Driven Decision Making:

ML algorithms enable data-driven decision-making by providing actionable insights into traffic patterns, allowing authorities to address issues proactively.

Optimized resource allocation, improved traffic flow, and better urban planning contribute to sustainable and efficient transportation systems.

Disadvantages:

Privacy Concerns:

Collecting and analyzing individual vehicle data for traffic management raises privacy concerns.

Striking a balance between effective traffic management and safeguarding individual privacy becomes a critical challenge.

Data Quality and Reliability:

ML algorithms heavily depend on the quality and reliability of data, and inaccurate or incomplete data can lead to flawed predictions.

Inaccurate predictions may result in suboptimal decision-making and resource allocation.

Implementation Costs:

Implementing AI-ML solutions requires significant upfront costs for infrastructure, technology, and skilled personnel.

Initial financial barriers may limit widespread adoption, particularly in less economically developed areas.

Technological Dependence:

Relying heavily on AI-ML systems may lead to dependence on technology, making transportation systems vulnerable to disruptions or cyber threats.

Potential system failures or security breaches could compromise the effectiveness of traffic management.

Balancing the advantages and addressing these challenges will be crucial for the successful implementation and widespread acceptance of AI-ML solutions in addressing contemporary traffic-related issues.

11.CONCLUSION

The Intelligent Traffic Management System using AI-ML aims to address the growing challenges associated with traffic problems. By predicting traffic volume, identifying rule violations, and providing real-time insights, the system contributes to effective traffic management and infrastructure development. Continuous refinement and adaptation will ensure the system remains effective in addressing evolving traffic patterns and challenges.

As the project concludes, the path forward involves continuous refinement of AI-ML algorithms, integration with smart city initiatives, and collaboration with governmental bodies to implement recommendations. The goal is to create a seamlessly integrated Intelligent Traffic Management System that adapts to changing urban dynamics, contributing to the creation of smarter, safer, and more sustainable cities.

In conclusion, the application of AI-ML in traffic management is not merely a technological advancement but a visionary step towards building more resilient and adaptive urban environments. The success of such initiatives lies not only in technological prowess but in the commitment to fostering positive changes in the way we perceive, manage, and navigate the complex dynamics of urban mobility.

12. FUTURE SCOPE

Some potential future directions for the project of predicting traffic volume using machine learning:

Integration with Smart City Initiatives:

The project can evolve to align with broader smart city initiatives. Integration with other smart systems, such as energy management, waste management, and public safety, can create a holistic approach to urban development.

Enhanced Predictive Analytics:

Continual improvement in predictive analytics is crucial. Future iterations of the project could focus on refining algorithms, incorporating more variables, and exploring advanced machine learning techniques to achieve even greater accuracy in traffic volume predictions.

Edge Computing for Real-Time Processing:

Embracing edge computing technologies can enhance the real-time processing capabilities of the system. This allows for faster decision-making, reduced latency, and improved responsiveness in handling dynamic traffic conditions.

Adaptive Traffic Signal Control:

Implementing adaptive traffic signal control systems that dynamically adjust signal timings based on real-time traffic conditions can further optimize traffic flow, reduce congestion, and improve overall transportation efficiency.

13. APPENDIX

Source Code: https://drive.google.com/drive/folders/1kqyEE0oLpy-

blcZseEhiBrsuxtVrIqZc?usp=drive_link

Github Link: https://github.com/smartinternz02/SI-GuidedProject-613813-1701254258