Predictive Traffic Analysis: Leveraging Machine Learning for Accurate Traffic Forecasting

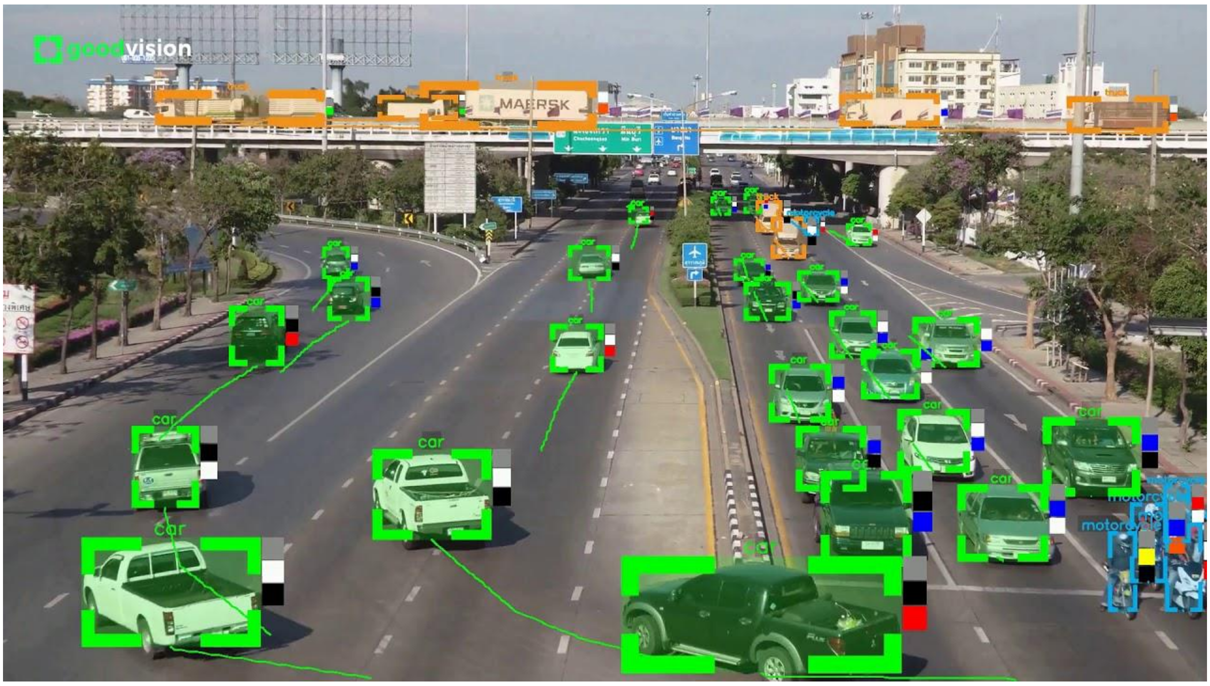

Table of Contents

Abstract

Data Science is a multidisciplinary field that involves extracting insights and knowledge from large volumes of structured and unstructured data. It combines aspects of statistics, mathematics, computer science, and domain knowledge to solve complex problems and make data-driven decisions. Data science and AI go hand in hand. The advancements in AI and machine learning algorithms will continue to drive the field of data science. AI-powered systems will become more sophisticated in analysing and interpreting complex data, leading to better prediction models, recommendation systems, and automation of various tasks. Machine learning algorithms are often employed to develop predictive models and uncover patterns and trends in the data. The future scope of data science is very promising, with a wide range of opportunities and advancements on the horizon. As technology continues to evolve, data is becoming increasingly important, and businesses are realizing the value of data-driven decision making. The applications of data science are vast and diverse. It is utilized across industries such as healthcare, finance, marketing, transportation, and many more. For example, data science can be used to predict customer behaviour, optimize business operations, detect fraud, develop recommendation systems, and even forecast future events such as traffic patterns or stock market trends.

1.1 The Great Gridlock-a concern

Traffic congestion is an extensive global phenomenon resulting from high population density, growth of motor vehicles and their infrastructure, and proliferation of rideshare and delivery services [1]. Researchers have defined congestion from different perspectives. The most common definition of congestion in the state of traffic flow is when the travel demand exceeds road capacity [2]. From the delay-travel time perspective, congestion occurs when the normal flow of traffic is interrupted by a high density of vehicles resulting in excess travel time [3]. Congestion can also be defined by the increment of the road user's cost due to the disruption of normal traffic flow [4]. A variety of reasons are responsible for creating congestion in most urban areas. Depending on these different reasons, congestion can be classified into recurring and nonrecurring congestion. Recurring congestion occurs regularly, mostly due to the excessive number of vehicles during peak hours [5]. On the other hand, unpredictable events—such as weather, work zones, incidents, and special events—are the causes of nonrecurring congestion [6,7,8]. According to the United States Department of Transportation Federal Highway Administration (DOT-FHWA), nonrecurring congestion contributes to more than

50% of all traffic congestion, where 40% of congestion is caused by recurring congestion [3]. The problem at hand is to increase the accuracy and reliability of traffic prediction using machine learning algorithms. The current state-of-the-art methods have limitations in providing precise forecasts due to the complex and dynamic nature of traffic patterns. Therefore, we aim to develop advanced machine learning models that can effectively leverage various data sources and features to improve the accuracy of traffic predictions.

1.2 The proposed solution

To address the problem, we propose to develop and deploy advanced machine learning models specifically designed for traffic prediction. The solution will involve several steps:

1. **Data Collection:** Comprehensive data collection will be performed to gather various types of data relevant to traffic prediction. This includes historical traffic data, real-time traffic data, weather conditions, road network data, public transportation data, and social media/news data. The data will be obtained from reliable sources and combined to create a comprehensive dataset.
2. **Feature Engineering:** Feature engineering will be employed to extract meaningful features from the collected data. This involves transforming and combining the data to create informative and relevant input features for the machine learning models. Features such as historical traffic patterns, real-time traffic conditions, weather conditions, road network characteristics, and relevant events will be considered.
3. **Model Selection:** Various machine learning algorithms will be explored and evaluated to select the most suitable models for traffic prediction. Algorithms such as linear regression, support vector machines (SVM), random forest, recurrent neural networks (RNN), long short-term memory (LSTM) networks, and gradient boosting will be considered based on their suitability for handling traffic data and capturing complex patterns.
4. **Model Training and Evaluation:** The selected models will be trained using the collected dataset, incorporating appropriate training and evaluation techniques. The dataset will be split into training and testing sets, with appropriate validation techniques used to ensure robust model performance. Evaluation metrics will be utilized to assess the performance of the models.

5. **Model Optimization:** Model tuning and optimization techniques will be applied to further enhance the accuracy of the traffic prediction models. This may involve fine-tuning hyperparameters, ensemble techniques, and applying advanced optimization algorithms to improve model performance.
6. **Deployment and Integration:** The optimized machine learning models will be deployed and integrated into a traffic prediction system. The system will utilize the trained models to generate real-time traffic predictions and provide insights and recommendations to traffic management authorities, drivers, and commuters.

1.3 The Result

- **Number of Cars (CarCount)** has the **most contribution** to Traffic
- **Thursday** and **Wednesday** are the most **busy days** for traffic
- **Peak hours** of traffic are between **8:00am-10:00am** and **3:00pm-6:00pm**
- **Normal traffic** situation **counts** the **most**
- **Heavy Traffic** mostly occurs **after 9:00pm**
- **Friday** sees the **minimum Traffic**

Chapter 1

Introduction

Imagine reaching late in an important meeting and getting your bonus cancelled by your boss! Or imagine reaching late in your own wedding! Horrible? Right.

Traffic congestion is indeed a significant concern in many cities around the world. It refers to the excessive number of vehicles on the road, resulting in slow, inefficient, and frustrating movement of traffic. The effects of traffic congestion are multi-faceted and can have serious implications for both individuals and society as a whole.

Firstly, traffic congestion leads to increased travel times. The time wasted in traffic can be quite significant, causing frustration and stress for commuters. It also has economic implications as it translates into productivity losses for businesses and increased fuel consumption. Additionally, the idling vehicles in traffic contribute to air pollution, affecting air quality and public health.

Traffic congestion also poses challenges for urban planning and infrastructure development. As populations grow and urban areas expand, the existing road networks often struggle to accommodate the increasing volume of vehicles. This leads to a demand for more roads and transportation infrastructure, which can be expensive and time-consuming to implement.

Traffic prediction is a critical aspect of transportation management, allowing for effective traffic flow optimization, route planning, and resource allocation. Accurate traffic prediction can help alleviate congestion, reduce travel times, and improve overall transportation efficiency. Machine learning algorithms have demonstrated great potential in analysing large amounts of traffic data and forecasting future traffic conditions. However, there is a need to further enhance the accuracy and reliability of traffic prediction models to maximize their effectiveness.

1.1 Problem Statement

Increasing traffic congestion leading to delay in reaching destination.

1.3 Objectives

Traffic congestion is a global issue that challenges the development of a resilient and sustainable transportation system. The long-term goal of this prediction model is to contribute to the development of a sustainable and resilient transportation management system that aims to minimize the negative socio-economic-environmental impact of congestion. Prior to the implementation stage, a multitude of road traffic analyses from different perspectives must be conducted. Monitoring the traffic flow in an area is one of the initial steps in establishing a proper traffic management system or mitigating congestion. Since there are various congestion measures available, considering multiple congestion measures can be complicated in a road traffic analysis. Thus, this paper reviews various traffic congestion measures by comparing each measure in a small-scale case study. Evaluating the available measures in order to find the appropriate congestion measures to be employed in road traffic analysis is crucial. In addition to exclusively listing various available congestion measures, this paper also aims to aid decision-makers with a preliminary evaluation of comparing each measure through data analysis.

Based on the challenges mentioned above, the motivation and objectives of this project are as follows:

(1) identification of the dataset
(2) performing data analysis
(3) normalizing the data and removing outliners
(4) finding best model
(5) making a prediction model with maximum possible accuracy

Chapter 2

Background

Understanding traffic patterns and analysing data can provide valuable insights for transportation planning, infrastructure development, and congestion management.

This dataset has been picked up from Kaggle. It is a valuable resource for studying traffic conditions as it contains information collected by a computer vision model. The dataset is stored in a CSV file and includes additional columns such as time in hours, date, days of the week, and counts for each vehicle type (CarCount, BikeCount, BusCount, TruckCount). The "Total" column represents the total count of all vehicle types detected within a 15-minute duration. The following attributes are considered

1. **Time:** it is recorded in a range of every 15 minutes.
2. **Date:** contains records of dates for which the data was collected.
3. **Day of the week:** contains the day of the week
4. **CarCount**: contains record for the no of cars passed in the specified time range.
5. **BikeCount:** contains record for the no of cars passed in the specified time range.
6. **BusCount:** contains record for the no of cars passed in the specified time range.
7. **TruckCount:** contains record for the no of cars passed in the specified time range.
8. **Total:** contains total number of vehicles passed for that duration. It includes the sum of car, bike, bus and trucks passed.
9. **Traffic Situation**: defines the type of traffic recorded in three categories i.e low, normal and heavy

Classification tree [1, 7]: It is a supervised machine learning approach to predict the output parameter (i.e., outcome). It consists of nodes (tests), edges (the outcome of a test) and leaf nodes (outcome). In this, the decision variable is discrete or categorical. It is mainly designed using binary recursive partitioning. This process uses iterations to split the data into partitions. The partitioning of the samples of each node is done utill all samples belong to the same class.

Support Vector Machine [47, 40, 23, 5, 38, 18]: SVM is a supervised learning approach to analyse the data and used it in the classification problem. SVM constructs a hyperplane or set of hyperplanes to classify the data into different classes.

K-Nearest Neighbour (k-NN) [8, 29, 33, 32, 4]: It is a classification algorithm that keeps all available data and classifies new data based on a similarity measure. The new data is classified based on the closest distance among the neighbours. The similarity measure is performed using Euclidean distance, Manhattan distance, Minkowski distance and hamming distance.

Naïve Bayes: This is based on bayes theorem, which is the collection of algorithms. It has independent assumptions for the features. It is a conditional probability model, which considers each feature to contribute separately, regardless of the correlation between the features. The main advantage of this algorithm is that it requires a small dataset for training to classify the categories.

Random Forest (RF): It is a classification approach that uses the average of multiple deep decision trees. The training algorithm used by RF is bootstrapping aggregation or bagging method.

Neural Network (NN) [48, 39, 13, 44, 19, 17]: It is a machine learning approach, which is used for classification purposes by modelling itself as a human brain. It consists of neurons that are arranged layer-wise, which converts the input vector to output. Each unit in NN takes input and applies a non-linear function to generate output, which is further passed to the next layer. Generally, ANN is a feed-forward network. Here, weights are applied to pass from one layer to another. In this way, learning is performed to get the desired output.

AdaBoost: It is an algorithm that is used for binary classification. This is mainly used with short decision trees. It is originally called as adaboost.m1. Each instance of the training set is assigned with a weight value. The initial weight is assigned as 1n, where n is the number of instances in the training set.

Logistic Regression: This is a well-known and popular classification algorithm that estimates the discrete values, such as yes or no, true or false and 0 or 1. It predicts the probability of an event by using the data in a logistic function.

What exactly is this dataset and how was it created?
This dataset has been picked up from Kaggle.**[9]** It is a valuable resource for studying traffic conditions as it contains information collected by a computer vision model. The model detects four classes of vehicles: cars, bikes, buses, and trucks. The dataset is stored in a CSV file and includes additional columns such as time in hours, date, days of the week, and counts for each vehicle type (CarCount, BikeCount, BusCount, TruckCount). The "Total" column represents the total count of all vehicle types detected within a 15-minute duration.
The dataset is updated every 15 minutes on kaggle, providing a comprehensive view of traffic patterns over the course of one month. Additionally, the dataset includes a column indicating the traffic situation categorized into four classes: 1-Heavy, 2-High, 3-Normal, and 4-Low. This information can help assess the severity of congestion and monitor traffic conditions at different times and days of the week.

In what cases can it be useful?
The dataset is useful in transportation planning, congestion management, and traffic flow analysis. It helps understand vehicle demand, identify congested areas, and inform infrastructure improvements. The dataset enables targeted interventions like signal optimizations and lane adjustments. It allows researchers to study traffic patterns by hour, day, or specific dates and explore correlations with external factors. It supports transportation research on vehicle relationships and traffic behaviour. Urban planners can assess traffic impact for zoning and infrastructure decisions. Overall, the dataset empowers stakeholders to make data-driven decisions, enhance urban mobility, and create efficient and sustainable cities.

2.2 related work

Literature	Dataset Used	Algorithm Used	Accuracy
Traffic Prediction: GRU (KARNIKA KAPOOR)	Traffic Prediction Dataset	Gated Recurrent Unit (GRU)	
Learn python using Traffic Index data sets (OSAMA BARAKAT)	Worldwide Traffic Congestion Ranking		
	Traffic Prediction Dataset		
	Traffic Prediction Dataset		

Chapter 3

Proposed Framework

In consideration of the above objectives, a framework was followed to achieve the task. The suitable dataset was identified and picked up from Kaggle. The data was then analysed using different using different ML charts and correlation features. To attain the maximum accuracy, the data was pre-processed using normalization and feature selection. Also, all the outliners were removed. The model was then trained and validated.

Flow diagram explanation

Figure 1: flow chart of proposed framework

Pseudo code

The steps are used to implement the wine quality prediction model is depicted as pseudo code

Pseudo code: Wine quality rate computing system

Input: Traffic Prediction Dataset

Output: Quality score

Step 1: Load the datasets

Step 2: Summarize the data distribution range using Visualization tool

Step 3: Exploratory data analysis

Step 4: Feature engineering

Step 5: Data Preprocessing

Step 6: Normalizing Data

Step 7: Performing Feature Selection

Step 8: Removing Outliners

Step 9: Invoke PCA

Step 10: Model Configuration

Step 11: Summarize the performance in terms rating and strength of a models using metrics

Getting Data

```
!pip install pycaret
import pandas as pd
df = pd.read_csv("/content/Traffic.csv")
df.dtypes
```

```
import seaborn as sns
```

```
Time                  object
Date                   int64
Day of the week       object
CarCount               int64
BikeCount              int64
BusCount               int64
TruckCount             int64
Total                  int64
Traffic Situation     object
dtype: object
```

Figure 2:dtypes of csv

Famililarize with the Data

```
df.head()
```

	Time	Date	Day of the week	CarCount	BikeCount	BusCount	TruckCount	Total	Traffic Situation
0	12:00:00 AM	10	Tuesday	31	0	4	4	39	low
1	12:15:00 AM	10	Tuesday	49	0	3	3	55	low
2	12:30:00 AM	10	Tuesday	46	0	3	6	55	low
3	12:45:00 AM	10	Tuesday	51	0	2	5	58	low
4	1:00:00 AM	10	Tuesday	57	6	15	16	94	normal

Figure 3:df.head()

```
df.describe()
```

	Date	CarCount	BikeCount	BusCount	TruckCount	Total
count	2976.000000	2976.000000	2976.000000	2976.000000	2976.000000	2976.000000
mean	16.000000	68.696573	14.917339	15.279570	15.324933	114.218414
std	8.945775	45.850693	12.847518	14.341986	10.603833	60.190627
min	1.000000	6.000000	0.000000	0.000000	0.000000	21.000000
25%	8.000000	19.000000	5.000000	1.000000	6.000000	55.000000
50%	16.000000	64.000000	12.000000	12.000000	14.000000	109.000000
75%	24.000000	107.000000	22.000000	25.000000	23.000000	164.000000
max	31.000000	180.000000	70.000000	50.000000	40.000000	279.000000

Figure 4:df.describe()

```
df.isnull().sum()
```

```
Time                 0
Date                 0
Day of the week      0
CarCount             0
BikeCount            0
BusCount             0
TruckCount           0
Total                0
Traffic Situation    0
dtype: int64
```

Figure 5:df.isnull().sum()

```
df
```

	Time	Date	Day of the week	CarCount	BikeCount	BusCount	TruckCount	Total	Traffic Situation
0	12:00:00 AM	10	Tuesday	31	0	4	4	39	low
1	12:15:00 AM	10	Tuesday	49	0	3	3	55	low
2	12:30:00 AM	10	Tuesday	46	0	3	6	55	low
3	12:45:00 AM	10	Tuesday	51	0	2	5	58	low
4	1:00:00 AM	10	Tuesday	57	6	15	16	94	normal
...	...	...	...	...	...	...	...	...	...
2971	10:45:00 PM	9	Thursday	16	3	1	36	56	normal
2972	11:00:00 PM	9	Thursday	11	0	1	30	42	normal
2973	11:15:00 PM	9	Thursday	15	4	1	25	45	normal
2974	11:30:00 PM	9	Thursday	16	5	0	27	48	normal
2975	11:45:00 PM	9	Thursday	14	3	1	15	33	normal

2976 rows × 9 columns

Figure 6:df-the dataset

Exploratory Data Analysis

```python
import matplotlib.pyplot as plt
x = df['Time']
y = df['CarCount']
z = df['BikeCount']
w = df['BusCount']
plt.bar(x, y, label='Cars', color='r')
plt.bar(x, z,  label='Bikes', color='y')
plt.bar(x, w, label='Buses', color='orange')

plt.title('Vehicles V/S Time')
plt.ylabel('Vehicles')
plt.xlabel('Time')

plt.legend()

plt.show()
```

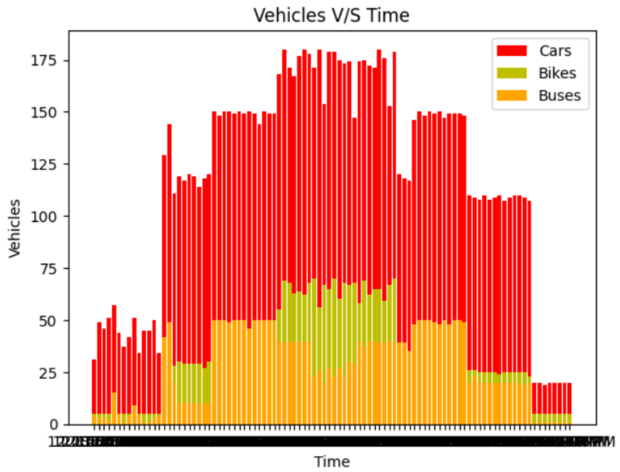

Figure 7:vehicles v/s time

The above graph represents the information between cars, bikes and buses with time. It shows that cars have the maximum contribution in increasing the traffic congestion followed by bikes and buses.

```
plt.figure(figsize=(10,5))
sns.histplot(data=df, x="Day of the week", hue="Traffic Situation")
```

```
<Axes: xlabel='Day of the week', ylabel='Count'>
```

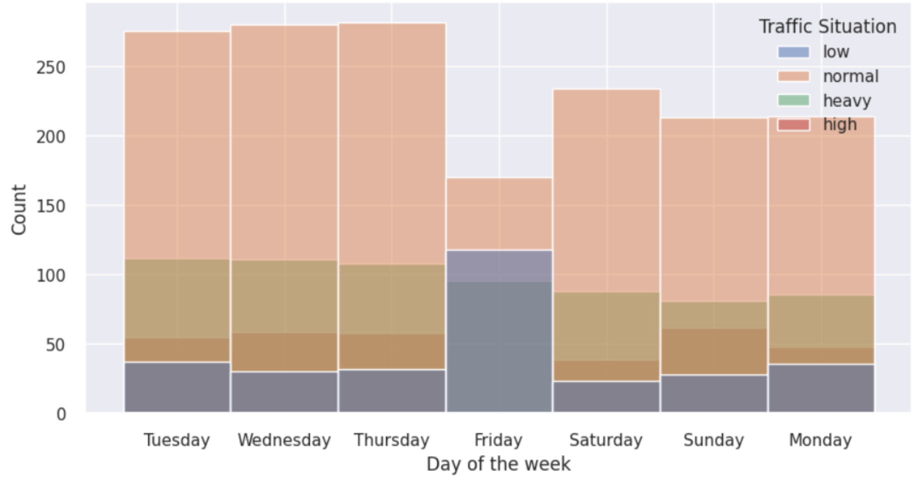

Figure 8:count v/s week day

The above graph represents the relationship of week days with the traffic situation. It is done with the help of seaborn library. It gives an insight about the time period when the highest traffic is recorded.

```
sns.histplot(df, x="Date", hue="Traffic Situation", element="poly")
```

```
<Axes: xlabel='Date', ylabel='Count'>
```

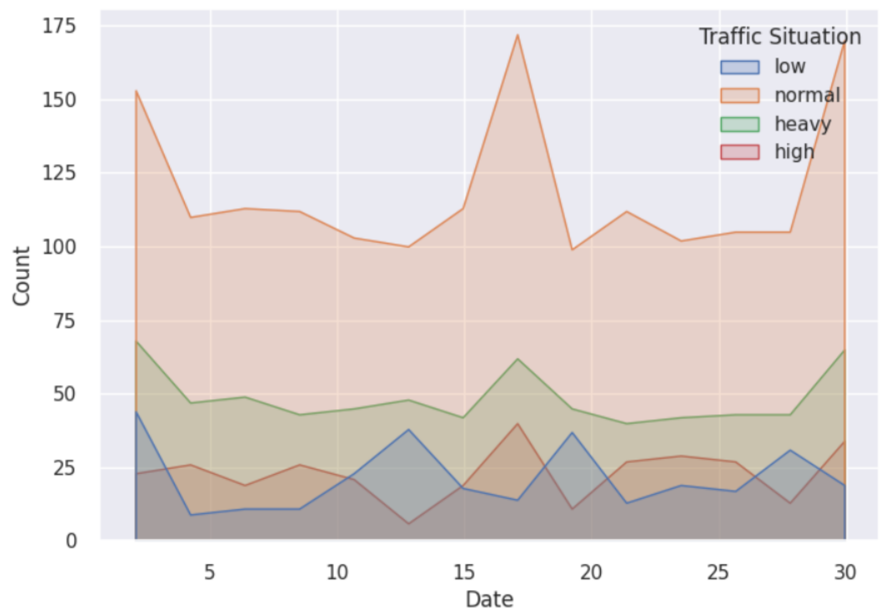

Figure 9:count v/s date

The above graph represents the relationship of week days with the traffic situation. It is done with the help of seaborn library. It gives an insight about the time period when the highest traffic is recorded.

```
sns.histplot(data=df, x="Total", hue="Traffic Situation", multiple="stack")
```

```
<Axes: xlabel='Total', ylabel='Count'>
```

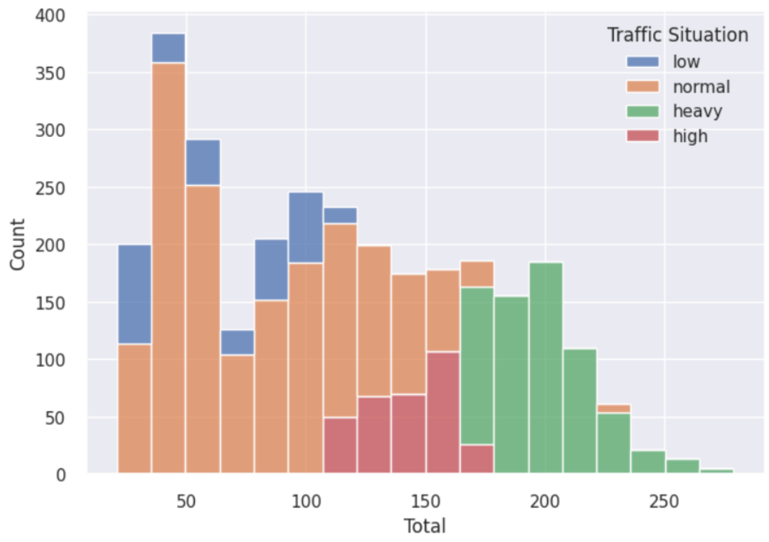

Figure 10:count v/s total

The above is represents Total V/S Traffic Situation.

Feature Engineering

```
df.columns
```

```
Index(['Time', 'Date', 'Day of the week', 'CarCount', 'BikeCount', 'BusCount',
       'TruckCount', 'Total', 'Traffic Situation'],
      dtype='object')
```

Figure 11:df.columns

The df.columns attribute in python is used to get the column labels of a Pandas DataFrame. It returns a list containing all the column names in the DataFrame.

```
df['Traffic Situation'].value_counts()
```

```
normal    1669
heavy      682
high       321
low        304
Name: Traffic Situation, dtype: int64
```

```
df.isnull().sum()
```

```
Time                 0
Date                 0
Day of the week      0
CarCount             0
BikeCount            0
BusCount             0
TruckCount           0
Total                0
Traffic Situation    0
dtype: int64
```

Data Pre-Processing

```
from pycaret.classification import*
s=setup(data=df,target='Traffic Situation')
compare_models()
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
gbc	Gradient Boosting Classifier	0.9986	1.0000	0.9986	0.9986	0.9986	0.9976	0.9977	1.6590
xgboost	Extreme Gradient Boosting	0.9986	1.0000	0.9986	0.9986	0.9986	0.9976	0.9977	0.3690
lightgbm	Light Gradient Boosting Machine	0.9986	1.0000	0.9986	0.9986	0.9986	0.9976	0.9977	0.8280
dt	Decision Tree Classifier	0.9947	0.9950	0.9947	0.9949	0.9947	0.9913	0.9914	0.1870
rf	Random Forest Classifier	0.9923	0.9993	0.9923	0.9925	0.9923	0.9874	0.9875	0.3850
knn	K Neighbors Classifier	0.9438	0.9879	0.9438	0.9446	0.9432	0.9076	0.9080	0.1270
et	Extra Trees Classifier	0.9371	0.9926	0.9371	0.9380	0.9351	0.8950	0.8965	0.5300
lr	Logistic Regression	0.8714	0.9709	0.8714	0.8696	0.8671	0.7852	0.7872	1.3190
lda	Linear Discriminant Analysis	0.8555	0.9673	0.8555	0.8572	0.8544	0.7653	0.7666	0.1110
ridge	Ridge Classifier	0.7681	0.0000	0.7681	0.6795	0.7021	0.5727	0.6026	0.2060
svm	SVM - Linear Kernel	0.7365	0.0000	0.7365	0.7170	0.6954	0.5315	0.5676	0.2450
nb	Naive Bayes	0.7091	0.9338	0.7091	0.8185	0.7310	0.5784	0.6117	0.1450
ada	Ada Boost Classifier	0.5607	0.5756	0.5607	0.3144	0.4029	0.0000	0.0000	0.2490
dummy	Dummy Classifier	0.5607	0.5000	0.5607	0.3144	0.4029	0.0000	0.0000	0.1050
qda	Quadratic Discriminant Analysis	0.5440	0.7381	0.5440	0.6392	0.5323	0.2762	0.3014	0.1080

Figure 12:model comparison

From the above we conclude that gbc, xgboost and lightgbm shows highest accuracy of about 0.9986, AUC of 1.0000, Recall, F1 and Prec. of about 0.9986, Kappa with 0.9976 accuracy, and MCC of about 0.9977 accuracy.

Feature Selection

```
s = setup(data=df, target='Traffic Situation', normalize = True, normalize_method = 'zscore')
cm = compare_models()
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
gbc	Gradient Boosting Classifier	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.6550
xgboost	Extreme Gradient Boosting	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.3570
lightgbm	Light Gradient Boosting Machine	0.9995	1.0000	0.9995	0.9995	0.9995	0.9992	0.9992	1.0150
dt	Decision Tree Classifier	0.9986	0.9984	0.9986	0.9986	0.9986	0.9976	0.9977	0.1980
rf	Random Forest Classifier	0.9909	0.9993	0.9909	0.9910	0.9908	0.9850	0.9851	0.3780
et	Extra Trees Classifier	0.9419	0.9930	0.9419	0.9426	0.9405	0.9032	0.9043	0.3970
lr	Logistic Regression	0.8901	0.9744	0.8901	0.8872	0.8862	0.8164	0.8179	0.1590
knn	K Neighbors Classifier	0.8541	0.9495	0.8541	0.8511	0.8464	0.7541	0.7577	0.1390
lda	Linear Discriminant Analysis	0.8526	0.9644	0.8526	0.8546	0.8513	0.7609	0.7625	0.1160
svm	SVM - Linear Kernel	0.8272	0.0000	0.8272	0.8212	0.8061	0.7005	0.7128	0.2340
ridge	Ridge Classifier	0.7643	0.0000	0.7643	0.6929	0.6960	0.5657	0.5960	0.2210
nb	Naive Bayes	0.6020	0.9182	0.6020	0.8288	0.6507	0.4550	0.5196	0.1490
ada	Ada Boost Classifier	0.5607	0.5755	0.5607	0.3144	0.4029	0.0000	0.0000	0.2510
dummy	Dummy Classifier	0.5607	0.5000	0.5607	0.3144	0.4029	0.0000	0.0000	0.1100
qda	Quadratic Discriminant Analysis	0.3624	0.8261	0.3624	0.8180	0.3870	0.2197	0.3236	0.1150

Figure 13: normalization

Normalization refers to rescaling real-valued numeric attributes into a 0-1 range. Data normalization is used in machine learning to make model training less sensitive to the scale of features. This allows our model to converge to better weights and, in turn, leads to a more accurate model.

1m

```
s = setup(data=df, target='Traffic Situation', feature_selection = True)
cm = compare_models()
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lr	Logistic Regression	0.8041	0.8687	0.8041	0.7240	0.7544	0.6440	0.6702	0.7960
gbc	Gradient Boosting Classifier	0.8041	0.8971	0.8041	0.7993	0.7693	0.6488	0.6680	1.4680
xgboost	Extreme Gradient Boosting	0.7974	0.8905	0.7974	0.7811	0.7662	0.6403	0.6557	0.4190
lightgbm	Light Gradient Boosting Machine	0.7960	0.8903	0.7960	0.7755	0.7676	0.6397	0.6532	1.0680
rf	Random Forest Classifier	0.7936	0.8853	0.7936	0.7762	0.7660	0.6372	0.6493	1.0400
dt	Decision Tree Classifier	0.7840	0.8841	0.7840	0.7609	0.7631	0.6267	0.6341	0.3040
et	Extra Trees Classifier	0.7840	0.8841	0.7840	0.7609	0.7631	0.6267	0.6341	0.8790
qda	Quadratic Discriminant Analysis	0.7811	0.8494	0.7811	0.7010	0.7380	0.6184	0.6309	0.3680
nb	Naive Bayes	0.7806	0.8493	0.7806	0.7007	0.7376	0.6178	0.6302	0.6690
knn	K Neighbors Classifier	0.7801	0.8693	0.7801	0.7620	0.7621	0.6222	0.6293	0.4520
lda	Linear Discriminant Analysis	0.7638	0.8692	0.7638	0.6685	0.7049	0.5770	0.5968	0.3010
ridge	Ridge Classifier	0.7494	0.0000	0.7494	0.5940	0.6616	0.5412	0.5733	0.3320
svm	SVM - Linear Kernel	0.6710	0.0000	0.6710	0.5759	0.6116	0.4435	0.4782	0.3370
ada	Ada Boost Classifier	0.5607	0.5755	0.5607	0.3144	0.4029	0.0000	0.0000	0.4320
dummy	Dummy Classifier	0.5607	0.5000	0.5607	0.3144	0.4029	0.0000	0.0000	0.4280

Figure 14:feature selection

Feature selection is a method of filtering out the important features as all the features present in the dataset are not equally important.

```
[34] s = setup(data=df, target='Traffic Situation', remove_outliers = True, outliers_threshold = 0.05)
     cm = compare_models()
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
dt	Decision Tree Classifier	0.9976	0.9973	0.9976	0.9976	0.9976	0.9961	0.9961	0.5500
gbc	Gradient Boosting Classifier	0.9966	0.9999	0.9966	0.9967	0.9966	0.9945	0.9945	1.9110
xgboost	Extreme Gradient Boosting	0.9933	1.0000	0.9933	0.9937	0.9931	0.9888	0.9892	0.5160
lightgbm	Light Gradient Boosting Machine	0.9923	0.9999	0.9923	0.9928	0.9921	0.9872	0.9876	1.5940
rf	Random Forest Classifier	0.9866	0.9991	0.9866	0.9868	0.9864	0.9779	0.9780	0.8980
et	Extra Trees Classifier	0.9404	0.9914	0.9404	0.9421	0.9387	0.9006	0.9021	0.7690
knn	K Neighbors Classifier	0.9361	0.9863	0.9361	0.9363	0.9354	0.8945	0.8950	0.5500
lr	Logistic Regression	0.8737	0.9676	0.8737	0.8711	0.8696	0.7889	0.7907	1.1900
lda	Linear Discriminant Analysis	0.8358	0.9515	0.8358	0.8384	0.8356	0.7323	0.7335	0.7070
ridge	Ridge Classifier	0.7614	0.0000	0.7614	0.6968	0.7028	0.5599	0.5887	0.4090
svm	SVM - Linear Kernel	0.7172	0.0000	0.7172	0.7453	0.6848	0.5241	0.5645	0.7040
nb	Naive Bayes	0.6903	0.9125	0.6903	0.8008	0.7141	0.5502	0.5819	0.4090
ada	Ada Boost Classifier	0.5607	0.5745	0.5607	0.3144	0.4029	0.0000	0.0000	0.5520
dummy	Dummy Classifier	0.5607	0.5000	0.5607	0.3144	0.4029	0.0000	0.0000	0.5890
qda	Quadratic Discriminant Analysis	0.4728	0.6475	0.4728	0.5710	0.4563	0.2109	0.2339	0.5800

Figure 15:outliner removal

```
s = setup(data=df, target='Traffic Situation', pca = True, pca_method = 'linear')
cm = compare_models()
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lightgbm	Light Gradient Boosting Machine	0.9554	0.9944	0.9554	0.9560	0.9548	0.9262	0.9269	3.3650
xgboost	Extreme Gradient Boosting	0.9539	0.9932	0.9539	0.9545	0.9533	0.9240	0.9246	0.4830
gbc	Gradient Boosting Classifier	0.9438	0.9906	0.9438	0.9445	0.9424	0.9065	0.9077	3.8160
knn	K Neighbors Classifier	0.9304	0.9864	0.9304	0.9308	0.9289	0.8843	0.8853	0.1410
rf	Random Forest Classifier	0.9203	0.9849	0.9203	0.9212	0.9170	0.8662	0.8685	0.7340
dt	Decision Tree Classifier	0.9121	0.9240	0.9121	0.9136	0.9116	0.8551	0.8560	0.1300
lr	Logistic Regression	0.9059	0.9757	0.9059	0.9051	0.9038	0.8439	0.8449	0.6040
et	Extra Trees Classifier	0.8958	0.9795	0.8958	0.8953	0.8909	0.8248	0.8277	0.6750
svm	SVM - Linear Kernel	0.8536	0.0000	0.8536	0.8538	0.8383	0.7463	0.7559	0.2540
lda	Linear Discriminant Analysis	0.8507	0.9640	0.8507	0.8534	0.8492	0.7580	0.7598	0.1890
nb	Naive Bayes	0.8363	0.9532	0.8363	0.8341	0.8328	0.7309	0.7324	0.1190
ada	Ada Boost Classifier	0.7998	0.8432	0.7998	0.8142	0.7847	0.6531	0.6695	0.3580
ridge	Ridge Classifier	0.7691	0.0000	0.7691	0.7001	0.6998	0.5731	0.6051	0.1870
dummy	Dummy Classifier	0.5607	0.5000	0.5607	0.3144	0.4029	0.0000	0.0000	0.1150
qda	Quadratic Discriminant Analysis	0.3557	0.7953	0.3557	0.8088	0.3856	0.2137	0.3237	0.1170

Figure 16:pca

Model Configuration

```
from pycaret.classification import *
s = setup(data=df, target='Traffic Situation')
rfModel = create_model('rf')
```

Fold	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.9856	1.0000	0.9856	0.9860	0.9855	0.9763	0.9766
1	0.9952	0.9995	0.9952	0.9953	0.9952	0.9922	0.9922
2	0.9856	0.9953	0.9856	0.9860	0.9857	0.9766	0.9767
3	0.9952	1.0000	0.9952	0.9952	0.9951	0.9921	0.9921
4	0.9952	0.9987	0.9952	0.9953	0.9952	0.9921	0.9922
5	0.9904	0.9995	0.9904	0.9905	0.9904	0.9842	0.9843
6	0.9904	0.9977	0.9904	0.9907	0.9904	0.9843	0.9845
7	0.9952	0.9999	0.9952	0.9952	0.9951	0.9921	0.9921
8	0.9952	1.0000	0.9952	0.9952	0.9951	0.9921	0.9922
9	0.9904	1.0000	0.9904	0.9905	0.9903	0.9842	0.9843
Mean	0.9918	0.9991	0.9918	0.9920	0.9918	0.9866	0.9867
Std	0.0037	0.0014	0.0037	0.0036	0.0038	0.0061	0.0061

Figure 17:creating rf model

Chapter 4

Results

```
newPredictions = predict_model(rfModel, data = df)
newPredictions
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	Random Forest Classifier	0.9976	0.9999	0.9976	0.9976	0.9976	0.9961	0.9962

Figure 18:predicted rf model

As per the predictions made, the random forest model shows an accuracy of about 0.9976, AUC of 0.9999, Recall and Prec. value as 0.9976, a F1 of 0.9976, kappa of 0.9961 and MCC of about 0.9962.

```
[59] plt.figure(figsize = (19,10))
     sns.heatmap(df[['Date', 'Day of the week', 'CarCount', 'BikeCount', 'BusCount',
                 'TruckCount', 'Total',  'Traffic Situation']].corr(),
                  cmap="YlGnBu",annot=True)
```

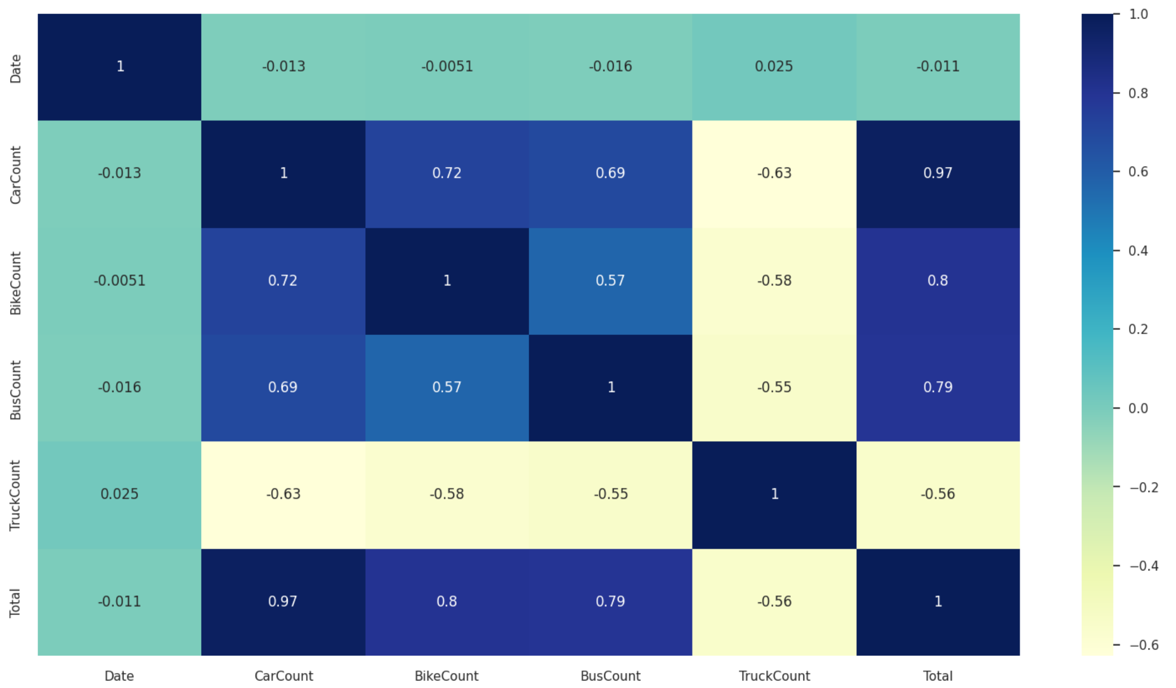

Figure 19: Heat map

The above map shows the 2-D representation of data in form of matrix where individual values are represented as colours.

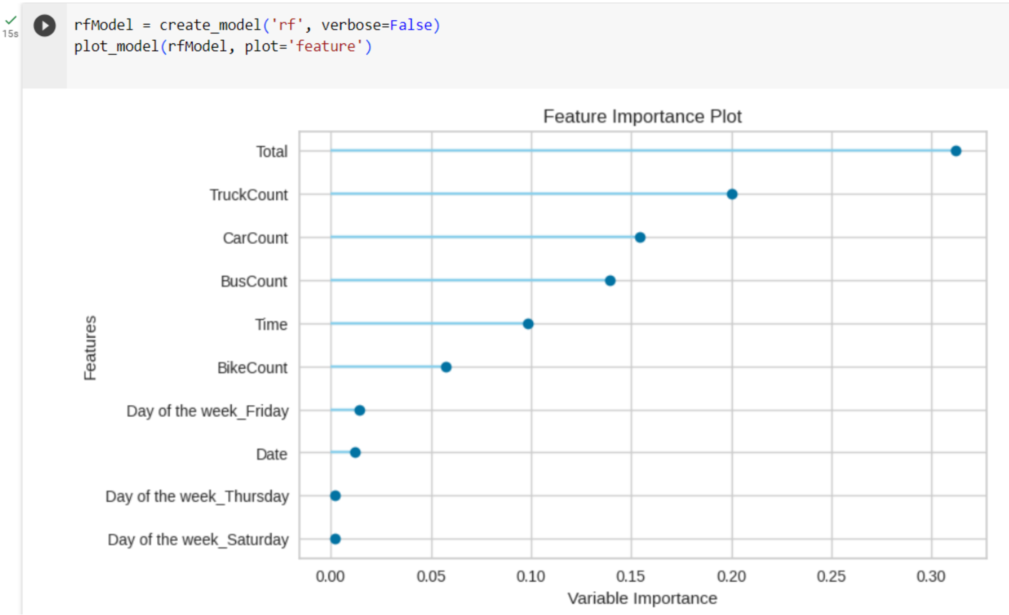

Figure 20:feature selection plot

The above graphs shows that the TOTAL attribute shows maximum variable importance followed by TRUCKCOUNT while the day of week attribute specifically Saturday shows least variable importance.

Figure 21:confusion matrix

The above matrix summarizes the performance of a machine learning model on the set of test data. The matrix is created between the True class and the predicted class.

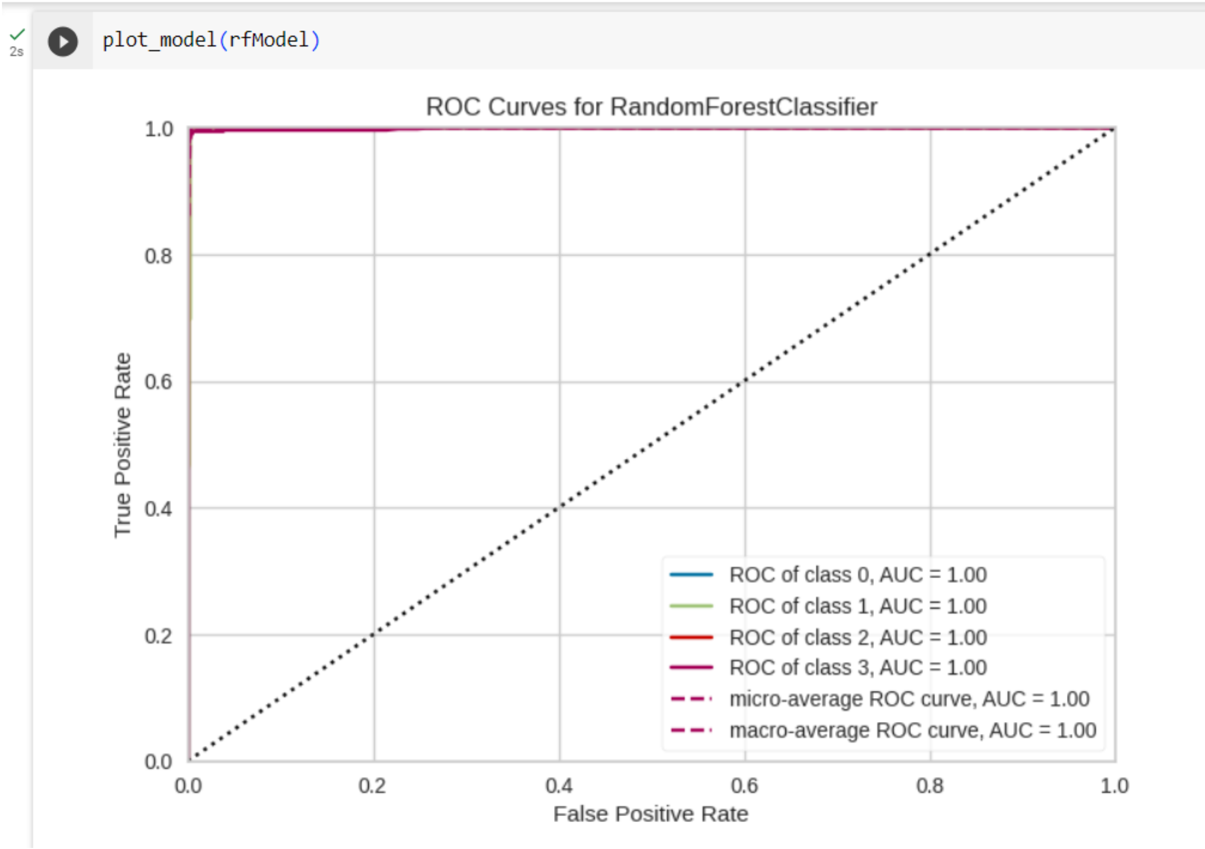

Figure 22:ROC Curves

Chapter 5

Conclusion and Future Scope

The proposed solution is expected to significantly increase the accuracy and reliability of traffic prediction using machine learning algorithms. By leveraging comprehensive data sources, advanced feature engineering techniques, and robust model training and optimization, we aim to provide highly accurate and timely traffic forecasts. The improved traffic prediction models will facilitate efficient traffic management, reduce congestion, optimize travel times, and enhance overall transportation planning and operations.

Enhancing traffic prediction using machine learning is a crucial requirement in optimizing transportation systems and improving the overall commuting experience. By applying advanced machine learning algorithms, comprehensive data collection, feature engineering, model training and optimization, and deployment, we aim to increase the accuracy and reliability of traffic predictions. This will contribute to more efficient traffic management, reduced congestion, and improved transportation planning, ultimately benefitting both traffic authorities and commuters alike.

5.1 Short-term solutions

Encourage sustainable transportation through public transportation, and incentivize people to use public transportation with economical fares that are socially equitable and accessible for all. Public transport improvements would also include automatic vehicle licensing and real time arrival information. Implement congestion pricing; price according to the number of people in a car and the time of day (toll for people who travel to town during high peak times; high parking fares can discourage people from using their cars downtown)

5.2 Mid-term solutions

Facilitate travel demand management by: Stuttering travel times Encouraging businesses to adopt telecommuting (working from home) Encouraging car-free zones, pedestrians, bicycle use and better pedestrian/bicycle connections Improving land use through smart growth policies (non-dense settlements and exclusive zoning) Designing transit strategies that encourage people to use high occupancy vehicles and public transportation. Use technology (such as GPS, digital maps) to help educate citizens and help them make better transportation choices. Digital platforms (apps) can also help to better integrate the transportation system so that citizens can plan their trips in real time. Transform culture, attitudes, and behaviours with regard to transportation. Pedestrianize the inner city to transform the human experience in downtown Istanbul and improve quality of life. Incorporate intelligent route finding to free up urban space for such activities as strolling around and communication. Add electronic or hybrid cars to the fleet of dolmuş to help alleviate greenhouse gas emissions.

5.3 Long-term solutions

Link rail, road and water transport on the one hand and public and private means of transport on the other. Create a sea dolmuş. Improve roadway security design; barriers on shoulders, curbs, roundabouts, advanced signal systems, lane restrictions for high occupancy vehicles (ex: bus lanes) and changeable lane allocation can help calm and manage traffic. Involve designers in the management and planning of an integrated transportation system. Designers have a unique mindset for solving problems that is distinct from traditional methods of urban planning, industrial design places the needs and experiences of human beings first when designing out traffic congestion.

References

1. Reed, T.; Kidd, J. *Global Traffic Scorecard*; INRIX Research: Altrincham, UK, 2019. [Google Scholar]
2. Aftabuzzaman, M. Measuring traffic congestion—A critical review. In Proceedings of the 30th Australasian Transport Research Forum (ATRF), Melbourne, Australia, 25–27 September 2007. [Google Scholar]
3. Systematics, C. *Traffic Congestion and Reliability: Trends and Advanced Strategies for Congestion Mitigation*; Cambridge Systematics Inc.: Cambridge, MA, USA, 2005. [Google Scholar]
4. Litman, T. *Congestion Reduction Strategies: Identifying and Evaluating Strategies to Reduce Traffic Congestion*; Victoria Transport Policy Institute: Victoria, BC, Canada, 2007. [Google Scholar]
5. FHWA. Operations–Reducing Recurring Congestion. Available online: https://ops.fhwa.dot.gov/program_areas/reduce-recur-cong.htm (accessed on 10 December 2019).
6. Falcocchio, J.C.; Levinson, H.S. Managing nonrecurring congestion. In *Road Traffic Congestion: A Concise Guide*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 197–211. [Google Scholar]
7. Ghosh, B. Predicting the Duration and Impact of the Nonrecurring Road Incidents on the Transportation Network. Ph.D. Thesis, Nanyang Technological University, Singapore, May 2019. [Google Scholar]
8. Fonseca, D.J.; Moynihan, G.P.; Fernandes, H. The role of nonrecurring congestion in massive hurricane evacuation events. In *Recent Hurricane Research—Climate, Dynamics, and Societal Impacts*; InTech: London, UK, 2011; pp. 441–458. [Google Scholar]
9. Traffic.csv (https://www.kaggle.com/datasets/hasibullahaman/traffic-prediction-dataset)

Cost Analysis

S. No.	Component / Material	Price (in Rs.)
1.		
2.		
3.		
Total		

ECE ARCHIVES PROJECT SUBMISSION FORM

Project Code: **CU/ECE/20____/Sem____/UID______________ (To be filled by Office)**

Project Name: ____

Team Members:

S. No.	Name	UID	Semester	Contact No.
1.				
2.				
3.				
4.				
5.				

Section to be filled by Project Mentor

Status (Please tick, whichever applicable)

Working		Not Working	
Marks Awarded		**60**	

Project Mentor Details:

Name _______________________________ Employee ID _______________

Sign ________________________________ Date ______________________

Section to be filled by Project Examiner(s)

Status (Please tick, whichever applicable)

Working		Not Working	

Project Examiner Signatures:

Internal _______________________________ Employee ID ______________

External _______________________________ Employee ID ______________

Date __________________________________