TRAFFIC FLOW ANALYSIS USING DIFFERENT MACHINE LEARNING ALGORITHMS

A PROJECT REPORT

Submitted by

IMAAD ZAFFAR KHAN (RA1811003010850)

MOHAMMAD MAAZ RASHID (RA1811003010866)

Under the Guidance of

Dr. T.K SIVAKUMAR

(Asst. Professor (Sr. Grade), Department of Computer Science and Engineering)

in partial fulfilment of the requirements for the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR- 603 203

MAY 2022

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR – 603 203

BONAFIDE CERTIFICATE

Certified that this B. Tech major project report titled "**Traffic Flow Analysis Using Different Machine Learning Algorithms**" is the bonafide work of Mr. IMAAD ZAFFAR KHAN and Mr. MOHAMMAD MAAZ RASHID who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion for this or any other candidate.

SIGNATURE GUIDE

Dr. T.K. SIVAKUMAR
Assistant Professor (Sr. Grade)
Dept. of Computing Technologies

SIGNATURE PANEL HEAD

Dr. A. JEYASEKAR
Associate Professor
Dept. of Computing Technologies

SIGNATURE

HEAD OF DEPARTMENT

Dr. M. PUSHPALATHA

Dept. of Computing Technologies

INTERNAL EXAMINER

EXTERNAL EXAMINER

Department of Computer Science and Engineering

SRM Institute of Science & Technology

Own Work Declaration Form

Degree/ Course: **B. Tech- Computer Science and Engineering**

Student Name: **IMAAD ZAFFAR KHAN, MOHAMMAD MAAZ RASHID**

Registration No: **RA1811003010850, RA1811003010866**

Title of Work: **TRAFFIC FLOW ANALYSIS USING DIFFERENT MACHINE LEARNING ALGORITHMS**

I / We hereby certify that this assessment compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism**, as listed in the University Website, Regulations, and the Education Committee guidelines.

I / We confirm that all the work contained in this assessment is my / our own except where indicated, and that I / We have met the following conditions:

- Clearly references / listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc)
- Given the sources of all pictures, data etc. that are not my own
- Not made any use of the report(s) or essay(s) of any other student(s) either past or present
- Acknowledged in appropriate places any help that I have received from others (e.g., fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course handbook / University website

DECLARATION:
I am aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except wherever indicated by referring, and that I have followed the good academic practices noted above.
If you are working in a group, please write your registration numbers and sign with the date for every student in your group.

I understand that any false claim for this work will be penalised in accordance with the University policies and regulations

(MOHAMMAD MAAZ RASHID)

(IMAAD ZAFFAR KHAN)

ACKNOWLEDGEMENT

We express our humble gratitude to **Dr. C. Muthamizhchelvan**, Vice Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support.

We extend our sincere thanks to **Dr. Revathi Venkataraman**, Professor & Chairperson, School of Computing, SRM Institute of Science and Technology, for his invaluable support.

We wish to thank **Dr. M. Pushpalatha,** Professor & Head of Department of Computer Science and Engineering, SRM Institute of Science and Technology, for her valuable suggestions and encouragement throughout the period of the project work.

We are extremely grateful to our Academic Advisor **Dr. G. Maragatham,** Professor, Department of Computer Science and Engineering, SRM Institute of Science and Technology, for their great support at all the stages of project work. We would like to convey our thanks to our Panel Head, **Dr. A. Jeyasekar ,** Associate Professor, Department of Computer Science and Engineering, SRM Institute of Science and Technology, for his / her inputs during the project reviews.

We register our immeasurable thanks to our Faculty Advisor**, Dr. V.V. Ramalingam,** Associate Professor, Department of Computer Science and Engineering, SRM Institute of Science and Technology, for leading and helping us to complete our course.

Our inexpressible respect and thanks to my guide, **Dr. T.K. Sivakumar,** Assistant Professor (Sr. Grade), Department of Computer Science and Engineering, SRM Institute of Science and Technology, for providing me an opportunity to pursue my project under his/her mentorship. He / She provided me the freedom and support to explore the research topics of my interest. Her / His passion for solving the real problems and making a difference in the world has always been inspiring.

We sincerely thank staff and students of the Computer Science and Engineering Department, SRM Institute of Science and Technology, for their help during my research. Finally, we would like to thank my parents, our family members and our friends for their unconditional love, constant support and encouragement.

(**IMAAD ZAFFAR KHAN)**

(MOHAMMAD MAAZ RASHID)

ABSTRACT

Prediction and analysis of the traffic flow is a subject of utmost importance. The Traffic Departments and the Governments could make use of the crucial data and key-points to interchange the vehicular routes and expedite the traffic movement in a smooth and effective manner. All the current and previous records and statistics pertaining to the vehicular traffic for a region can be utilized to identify the various road networks and the traffic patterns to predict the flow of the imminent and upcoming traffic.

This data can prevent congestions in the near future and help scale down the overall travelling time for an individual. Human lives can also be saved as the emergency facilities such as the ambulances, fire-fighting vehicles etc. would reach their respective emergency locations without any further delays. In order to control and eliminate the congestions and the gridlocks in a controlled and effective way, prediction and analysis of the traffic is paramount. This research paper will present various ML models and a comparison will be drawn between them and the currently in-use primitive models by utilizing different performance metrics. Toward the end, we will plot graphs showing the dependence of traffic on various attributes and an analysis of the outcomes and the graphs would be done to infer some useful understanding.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

ML	-	Machine Learning
RMSE	-	Root Mean Square Error
RF	-	Random Forest
NN	-	Neural Network
LSTM	-	Long Short-Term Memory
FVD	-	Floating Vehicle Data
CCTV	-	Closed Circuit Television
GRU	-	Gated Recurrent Unit
SVM	-	Support Vector Machine
ARIMA	-	Auto Regressive Integrated Moving Average

CHAPTER 1

INTRODUCTION

1.1 CURRENT SCENARIO

Vehicular routes statistics and the related data to the road network is of utmost value for designing transport activities and other associated research activities. According to a study, commuters ordinarily spent over 79 hours caught in the traffic in 2017 alone in some parts of the world. Analyzing the vehicular movements helps in ascertain the commute of vehicles through various roads and re-routing them to different lanes and roads to reduce the occurrences of congestions.

All the current and previous records and statistics pertaining to the vehicular traffic for a region can be utilized to identify the various road networks and the traffic patterns to predict the flow of the imminent and upcoming traffic.

Traffic congestion increases air pollution in the area as well as increased traffic pollution, and recent studies have shown that the death toll for motorists, commuters and people living near highways and traffic congested areas is very high.

Our current knowledge of air pollution and its impact on traffic congestion is insufficient. Therefore, the study of motor prediction methods is very important in alleviating this difficulty. This can help traffic controllers control traffic. This exchange increases the time required for the trip and thus forces the fare to rise.

The first step is to accumulate all the traffic-related data for analysis. There are numerous approaches for gathering the data. To collect the data, various detectors and equipment are instated throughout different road networks to estimate the volume of traffic on a road at a particular instance.

Equipment such as personal road courses, test vehicles or floating vehicle data (FVD), sidewalk detectors, closed-circuit television (CCTV), camera, photographs, are commonly used. In this study, we attempted to gather relevant data needed to predict traffic flow and consider the type of traffic that exists in India. The pre-existing records and statistics pertaining to the vehicular traffic for a region can be utilized to identify the various road networks and the traffic patterns to predict the flow of the imminent and upcoming traffic.

1.2 PROJECT PURPOSE

Over the few decades, various approaches have unfolded many algorithms and models so as to intercept this issue. In order to eliminate this problem accurately, certain crucial attributes have to be kept in mind. For instance, using of the climatic information, the overall population, national holidays, festival day-offs are some of the essential and paramount attributes that help in accurately forecasting the traffic flow.

But the existing systems haven't really taken these crucial attributes into consideration although they heavily influence the prediction outcomes.

Hence to address these shortcomings, our prediction model utilises the climatic information, the overall population, national holidays, festival day-offs and some other essential and paramount attributes that help in accurately forecasting the traffic flow by executing numerous machine-learning algorithms. Before coming up with our traffic prediction model, we need to examine different ML-algorithms.

Various metrics and constraints would be analysed to decide our final ML-algorithm on which we would frame our actual prediction model. This will eventually assist us in developing the most precise and accurate prediction model. This will be our first phase our project. The second phase would consist of prediction and analysis of the traffic flow patterns. Useful analysis and outcomes will be derived showing the dependency of the traffic flow on the various attributes.

CHAPTER 2

LITERATURE REVIEW

Over the few decades, various discrete approaches have unfolded many solutions to predict the traffic flux precisely. Initially, a regression perspective was put-forth by [1] “Mr. Liliian”. In this method, one was able to forecast the traffic flow precisely by taking into account various attributes from a traffic dataset of Southern China. Another prototype introduced by [2] “Mr. Fieng” suggested a representation using the overall summation of traffic flow. Data and useful insights were gathered at the road intersections using different devices. Yet another approach introduced by [3] “Mr. Shinmei” implied forecasting the traffic flux using the K-Means model. [4] “Mr. Guowandai” put-forth a framework that focused primarily on the temporal aspect along-with the “GRU”. The G-recurrent unit utilizes the Spatial-Temporal to forecast the overall traffic flux.

Another interesting approach devised by [5] “Mr. Rong Nyao” suggested applying the Markov Method Technique to forecast the flux accurately. [6] “Mr. Li-Chang’s” research utilized a hybrid-prediction approach that consisted of a SVM along-with a random forest model to only use the most important and useful information to ascertain the most ideal and peak predictive-attributes.

Other models consist of applying a flux approach on a car-orientation dataset where the spatial-likelihood dispersal of the vehicles is delineated. [7] “Mr. Ylxvan” prioritized using a LSTM framework to forecast the traffic flux along the different lanes of the roads.

[8] “Mr. Chinai” introduced another model consisting of radial Attributes to accurately forecast the velocities and the vehicular blockage. In order to remove the outliers and to even out all the noises, [9] “Mr. Xinqun” presented a framework that utilized discrete techniques and schemes and also consisted of a LSTM framework to better forecast the obstructions on the roads.

The already existing ITS's (Intelligent Transport System) can be further tuned for them to make better choices in terms of vehicular routing and blockage reduction. Keeping this in mind, two frameworks have been devised by [10] "Mr. Bukersche".

One is based on a statistical approach and the other is dependent on ML algorithms. LSTM frameworks can be further enhanced using Temporal convolutional context blocks. This was theorized by [11] "Mr. Hakaung" in which he used the loss switches in the whole architecture.

Frameworks integrating two models was first devised by [13] "Mr. Siaqumn". This model consisted of merging of the ARIMA and the LSTM frameworks. Obviously after the collaboration, the overall model gave improved outcomes.

Feature enhancements can make predicting the traffic flux for short intervals more accurate. [14] "Mr. Linjaingh" exploited this fact and came up with a G-Boosting RT model to make it possible. The outcomes were far more efficient and gave improved traffic flow forecastings.

CHAPTER 3

SYSTEM ANALYSIS

3.1 PROBLEM DEFINITION

Nowadays people face a lot of challenges in their day-to-day lives and one of the biggest obstacles that a man daily faces is the blockages and overcrowdings due to the traffic on the roads. Hence administering the traffic is the chief and the foremost task for a government to superintend to prevent fatalities on the roads and improve time utilization. The increasing number of vehicles is not the only culprit, in-fact other factors such as inadequate gridlock networks, poor planning, lack of urbanization and other factors also heavily contribute to daily bottlenecks on the roads.

All the above-mentioned factors are massive challenges for a Ministry to administer appropriately. Aiming to improve on the complications faced, a state agency or a governmental institution must ensure smooth flow of the traffic and take crucial and paramount steps to minimize road blockages. The overall concentration of the vehicles should be evenly distributed across different routes and grids to ensure no bottlenecks.

In most of the countries, we notice that most of the people waste a huge amount of time while being stuck in traffic jams for hours. This can lead to negative consequences for a country as man-hours are cut and eventually it results in reduced advancements and progress. Hence a governmental institution must take paramount steps to ensure trouble-free flow of the traffic.

3.2 PROJECT OBJECTIVES

Over the few decades, various approaches have unfolded many algorithms and models so as to intercept this issue. In order to eliminate this problem accurately, certain crucial attributes have to be kept in mind. For instance, using of the climatic information, the

overall population, national holidays, festival day-offs are some of the essential and paramount attributes that help in accurately forecasting the traffic flow.

But the existing systems haven't really taken these crucial attributes into consideration although they heavily influence the prediction outcomes.

Hence to address these shortcomings, our prediction model utilises the climatic information, the overall population, national holidays, festival day-offs and some other essential and paramount attributes that help in accurately forecasting the traffic flow by executing numerous machine-learning algorithms. Before coming up with our traffic prediction model, we need to examine different ML-algorithms.

Various metrics and constraints would be analysed to decide our final ML-algorithm on which we would frame our actual prediction model. This will eventually assist us in developing the most precise and accurate prediction model.

3.3 EXPERIMENTAL OBJECTIVE

The experimental objective will be to compare the below-mentioned machine learning algorithms based on their forecasting abilities. The systems will be tested and evaluated with the same data in a bid to obtain their prediction accuracy. Many versions of the predictive system will be implemented; each system will be moulded using a particular machine learning algorithm from the below list:

- **Linear Regression**
- **Decision Trees**
- **Random Forest**
- **Gradient Boosting**
- **Adaptive Boosting**

3.4 PROPOSED SYSTEM

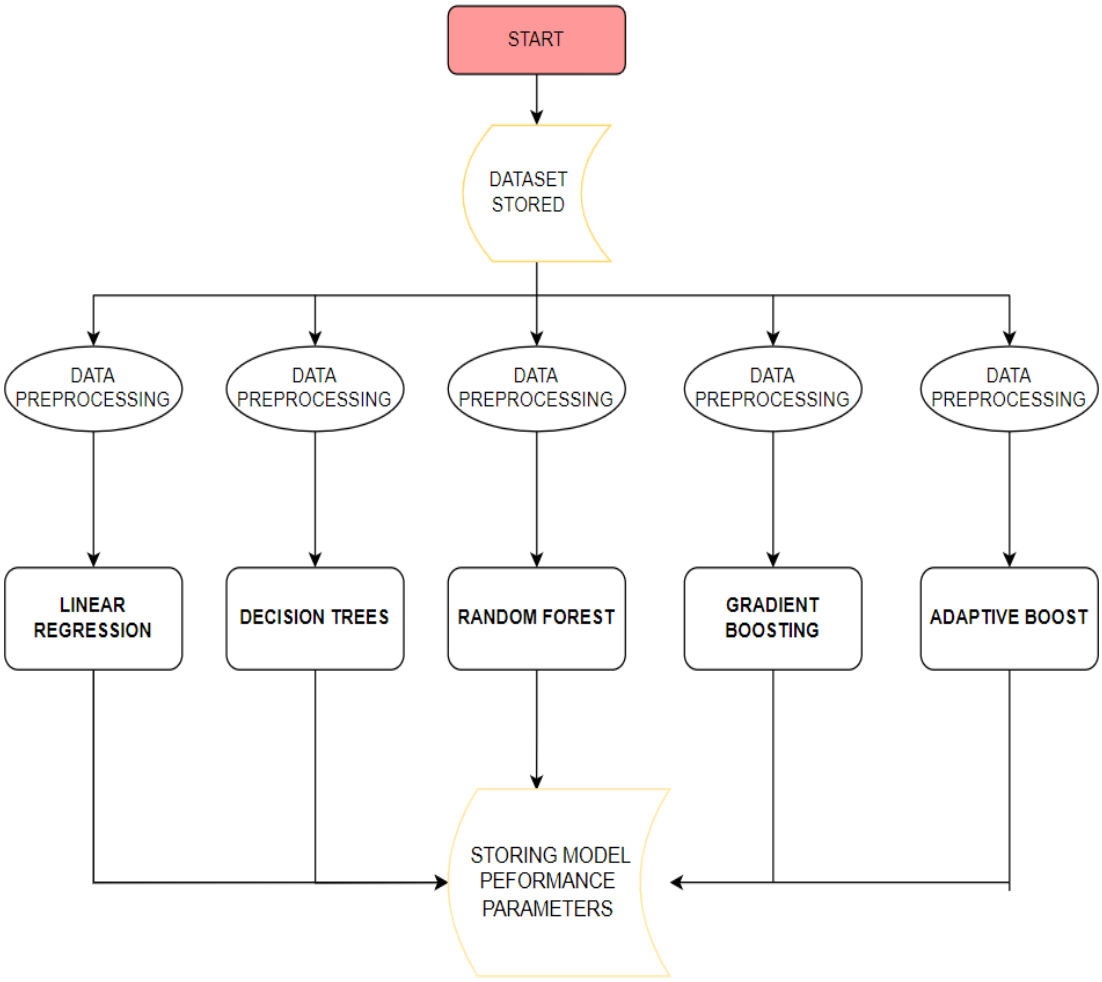


Figure 3.4: *Block Diagram for Model Analysis*

The above, figure 3.4, showcases the working of the proposed systems for analysis purposes. In this proposed system, we centre our system around analysing the different machine learning algorithms and try predicting traffic flow with each model so as to try to reach a conclusion wherein we will have obtained the better model.

CHAPTER 4
METHODOLOGIES

4.1 OBTAINING THE DATASET

We will be utilizing a dataset from *kaggle.com*. It contains the information regarding the Metro Interstate Traffic Volume. It contains a rich amount of significant information with various attributes pertaining to weather, day-offs, festivals, population etc. The raw data has been collected from the vehicles travelling from the area of Minneapolis towards Saint Paul. Around 48,000 instances and 9 attributes of the data have been specified in this dataset.

4.2 PROPOSED ALGORITHMS

4.2.1 LINEAR REGRESSION

It is the most basic algorithm of supervised learning technique. In this algorithm, predictions are made for continuous attributes. Basically, the relationship between an independent and a dependent attribute is plotted on a 2-dimensional plane as shown in Figure 4.2.1 and the optimum fit-line that passes through maximum points is found. This line is ultimately used to predict different values.

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n \qquad \text{(EQUATION 1)}$$

For the above Equation 1:

- y is the predicted value
- n is the number of features.
- x_1 is the i^{th} feature value.
- θ_j is the j^{th} model parameter

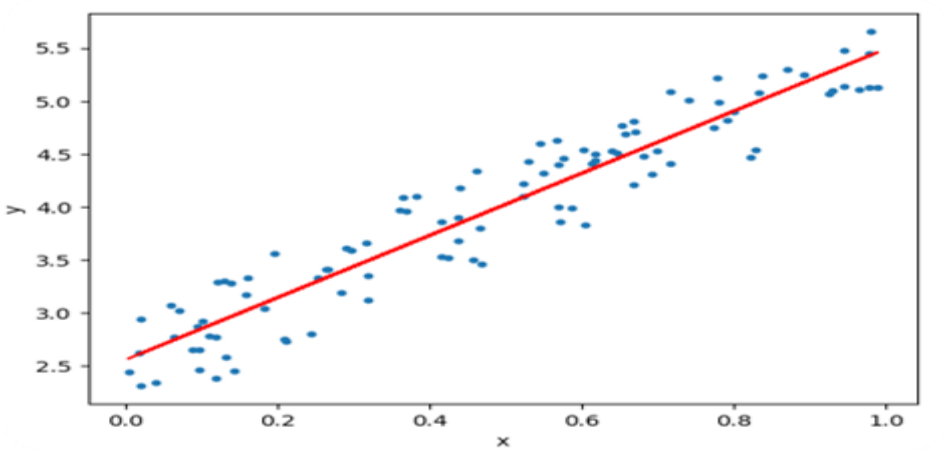


Figure 4.2.1: *Linear regression*

4.2.2 DECISION TREES

Another type of supervised-learning technique which uses the concept of tree-structured distributions. The tree structured distribution is shown in Figure 4.2.2. The nodes illustrate the characteristics of a dataset, branching illustrate the resolution rules and the leaf nodes illustrate an outcome. The tree is traversed from the root to the leaf nodes and all the nodes are organized simultaneously. This algorithm forecasts the final attribute values by implementing discrete decision rules.

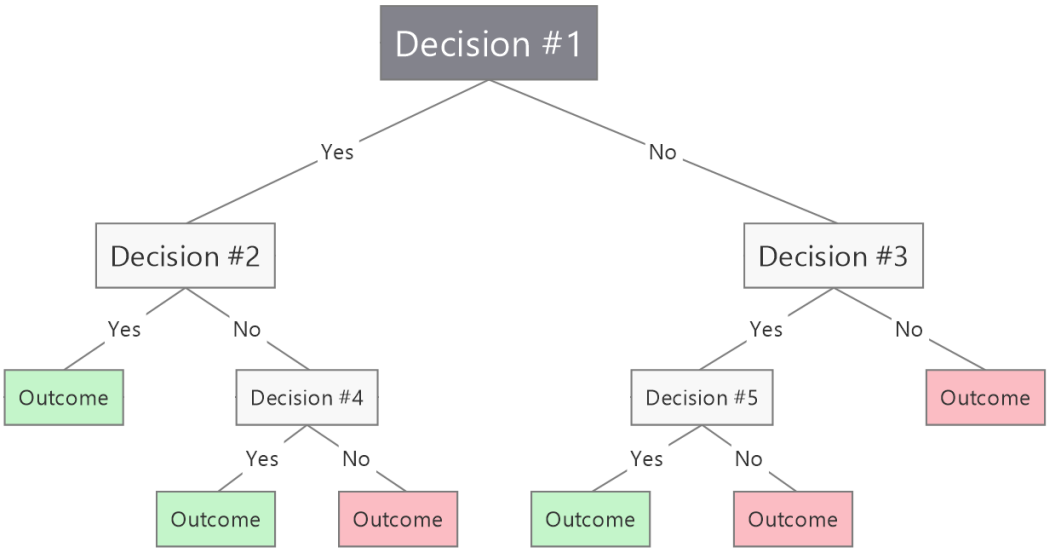


Figure 4.2.2: *Decision Tree*

4.2.3 RANDOM FOREST

This particular algorithm is established on the concept of ensemble learning. It is a type of supervised learning technique. It involves merging of all the individual decision trees and thus obtain the most optimum prediction model as illustrated in Figure 4.2.3. All the votes from different decision trees are accumulated to achieve the final output. The average of all the outcomes from individual decision trees is taken to improve the overall predictive precision for a particular dataset.

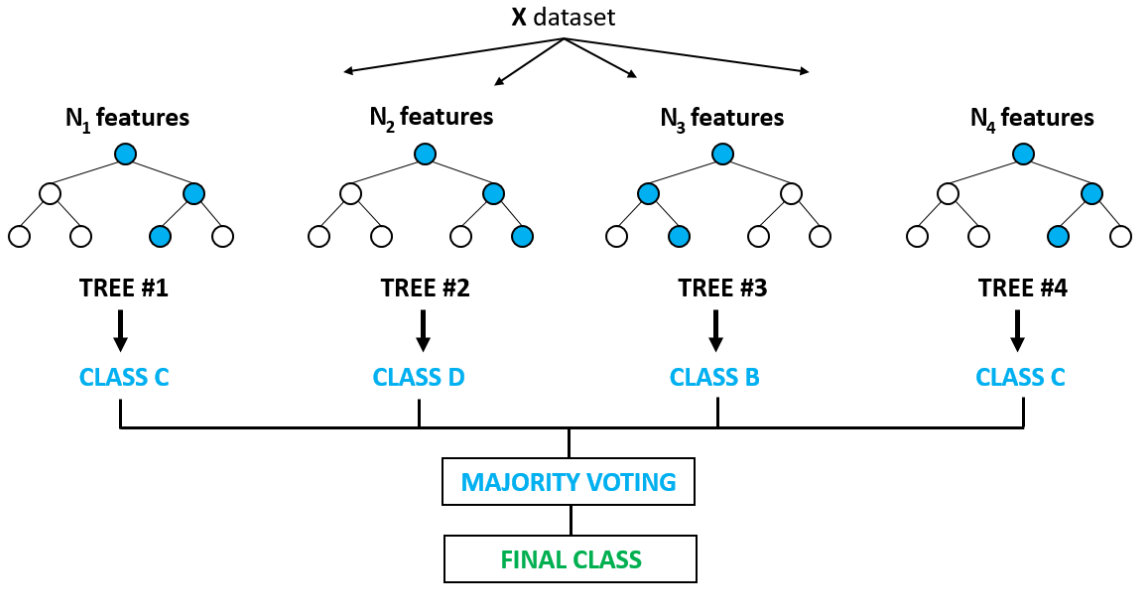


Figure 4.2.3: *Random Forest*

4.2.4 GRADIENT BOOSTING

Boosting is one of the most powerful algorithms in Machine-learning. Gradient Boosting, a type of boosting model works by minimizing the error in each and every step as illustrated in Figure 4.2.4. Each predictor combines itself with its predecessor to minimize the overall residual error. Gradient boosting is a sequential ensemble technique where the overall accuracy is optimized over consecutive iterations.

The gradient boosting algorithm consists of certain components like a loss function, weak learners and strong learners.

Loss function: The loss function is optimized to minimize the overall error. The outcomes from the weak learners are averaged in order to reduce the loss function.

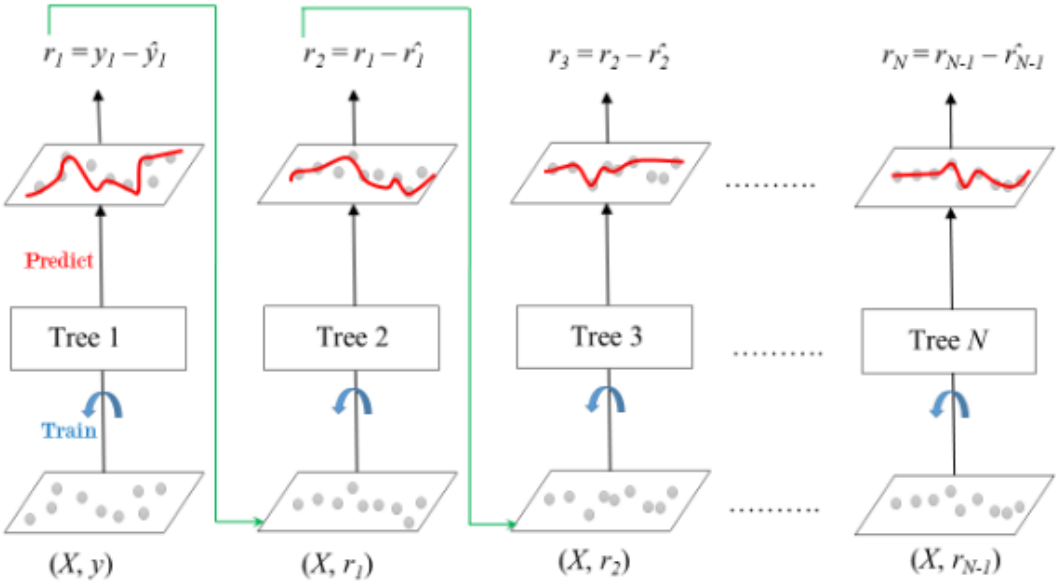


Figure 4.2.4: *Gradient boosting algorithm*

4.2.5 ADAPTIVE BOOST

AdaBoost is another type of boosting algorithm where uniform significance is assigned to each of the observations at the start. Shortly afterwards assessing the 1st tree, the importance of other observations is incremented and for the observations which are simply classified, their importance is decremented. Now these latest sets of weighted information are used in constructing of a redesigned tree. Subsequently, the newer framework becomes an aggregate of the 1st and the 2nd tree. Hence this newer model is able to enhance the overall precision of the predictions produced by the 1st tree. Only certain crucial and dominant attributes are selected in this algorithm to generate convincing outcomes.

Ada-boost functions mostly by transforming weak learners to strong learners. It makes ‘n’ number of decision trees during the data training period. As the first decision tree/model is made, the incorrectly classified record in the first model is given priority. Only these records are sent as input for the second model. The process goes on until we specify a number of base learners we want to create. Remember, repetition of records is allowed with all boosting techniques. The functioning and the different steps of the Ada Boost algorithm is explained in Figure 4.2.5.

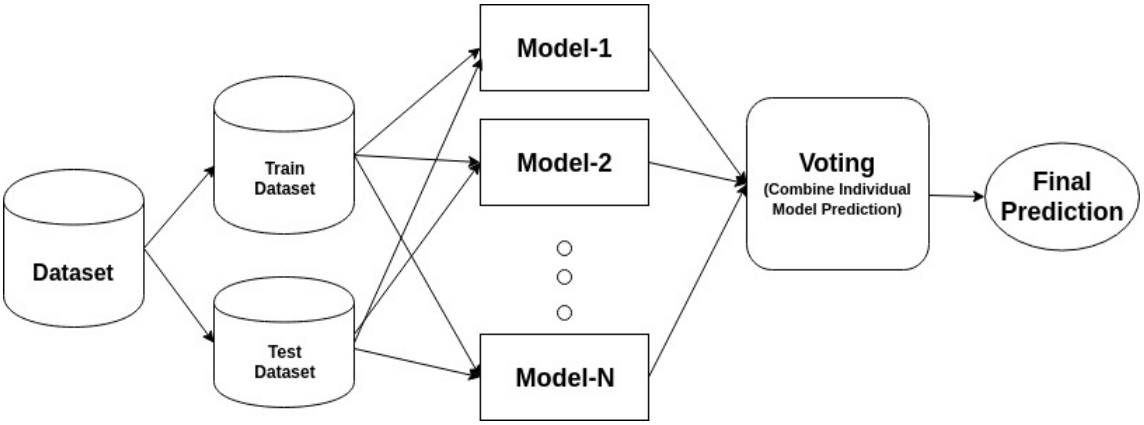


Figure 4.2.5: *Adaptive Boost algorithm*

CHAPTER 5

PROPOSED APPROACH

To accurately forecast the vehicular movements and predict the road blockages, many techniques have been proposed previously. We will be implementing certain Machine-learning algorithms in this project. For our model, we would utilize a dataset from "www.kaggle.com" with various significant attributes. We will only be implementing those algorithms that would be compatible with our dataset. The dataset that we have chosen and initialization and the implementation of different modules and libraries on the dataset would heavily influence the overall results of the prediction model.

Taking note of the points mentioned above, we would be implementing different ML-algorithms and compare and contrast their performances using different metrics. Eventually after comparing the algorithms, we will select the most optimum algorithm and continue to develop our prediction model.

5.1 DATASET

We will be utilizing a dataset from *kaggle.com*. It contains the information regarding the Metro Interstate Traffic Volume. It contains a rich amount of significant information with various attributes pertaining to weather, day-offs, festivals, population etc. The raw data has been collected from the vehicles travelling from the area of Minneapolis towards Saint Paul. Around 48,000 instances and 9 attributes of the data have been specified in this dataset. The 9 attributes are:

- date_time – DateTime Hour of the data collected in local CST time.
- Holiday – Categorical US holidays plus regional holiday-Minnesota State Fair.
- Temp - Numeric Average temp in Kelvin.
- Rain_1h – Numeric Amount in mm of rain that occurred in the hour.
- Snow_1h - Numeric Amount in mm of snow that occurred in the hour.
- Clouds_all – Numeric percentage of cloud cover.
- Weather_main – Categorical Short textual description of current weather.
- Weather_description - Categorical longer textual description of current weather.
- traffic_volume – Numeric Hourly I-94 ATR 301 reported Westbound Traffic volume.

```
In [10]: train_df.describe(include = 'all')
```

Out[10]:

	holiday	temp	rain_1h	snow_1h	clouds_all	weather_main	weather_description	date_time	traffic_volume
count	48204	48204.000000	48204.000000	48204.000000	48204.000000	48204	48204	48204	48204.000000
unique	12	NaN	NaN	NaN	NaN	11	38	40575	NaN
top	None	NaN	NaN	NaN	NaN	Clouds	sky is clear	2013-05-19 10:00:00	NaN
freq	48143	NaN	NaN	NaN	NaN	15164	11665	6	NaN
mean	NaN	281.205870	0.334264	0.000222	49.362231	NaN	NaN	NaN	3259.818355
std	NaN	13.338232	44.789133	0.008168	39.015750	NaN	NaN	NaN	1986.860670
min	NaN	0.000000	0.000000	0.000000	0.000000	NaN	NaN	NaN	0.000000
25%	NaN	272.160000	0.000000	0.000000	1.000000	NaN	NaN	NaN	1193.000000
50%	NaN	282.450000	0.000000	0.000000	64.000000	NaN	NaN	NaN	3380.000000
75%	NaN	291.806000	0.000000	0.000000	90.000000	NaN	NaN	NaN	4933.000000
max	NaN	310.070000	9831.300000	0.510000	100.000000	NaN	NaN	NaN	7280.000000

Figure 5.1: *Dataset Features*

5.2 PRE-PROCESSING

Consequent to choosing a dataset, the pre-processing and data cleaning begins. For this purpose, we chose python and some of its significant modules and libraries. For instance, Numpy implements different mathematical computations throughout the algorithm and offers the functionality of implementing multi-dimensional arrays, Pandas assists in transformation and scanning of the dataset along-with the ability of representing data in the form of data-frames, Matplotlib creates immersive visualisations of the data and many more. Pre-processing is spread out in numerous phases. No dataset is completely flawless and inconsistencies are always present, hence pre-processing of the dataset initially is of paramount significance. Some of the steps implemented while pre-processing the data involve dropping of redundant and non-essential columns, discovering the missing values and deleting the rows and columns containing null values, checking the correlations amongst different attributes and dropping the columns that are non-significant in the forecasting.

Pre-processing also involves changing the formatting of the values of certain columns so that the ML-algorithms can identify and read them. For instance, for our model, the data in the weather column first had to be converted into a numerical format in order for the algorithm to actually make use of it.

Another instance where the pre-processing is significant is when certain data fields of the rows contain a null or an infinity value. Usually, these values are substituted with different entries. Null values have to be substituted with mean values and the infinity values are removed in favour of priorly-calculated maximum values.

5.3 SOFTWARE COMPONENTS USED

Software components required for traffic flow prediction are Jupyter notebook and different libraries in python that are used for performing machine learning.

.

5.3.1 JUPYTER NOTEBOOK

Figure 5.3.1: *Jupyter Notebook Tool*

Jupyter Notebook is an open-source platform which facilitates computing across various programming languages, and it is a combination of software code, computational output, self-explanatory text and multi-media information in a single tool.

It is used to transform data and provide statistical answers and combine different code executions.

We can create multiple notebook documents using Jupyter. By using Jupyter Notebook, you can use Keras, TensorFlow and OpenCV simply by installing and importing in the notebook. It provides an environment that associates code execution, rich text, mathematics, plots and rich media.

5.4 ARCHITECTURE DIAGRAM

Figure 5.4, showcases the working of the proposed systems for analysis purposes. In this proposed system, we centre our system around analysing the different machine learning algorithms and try predicting traffic flow with each model so as to try to reach a conclusion wherein we will have obtained the better model and then build our final prediction model using that very ML algorithm.

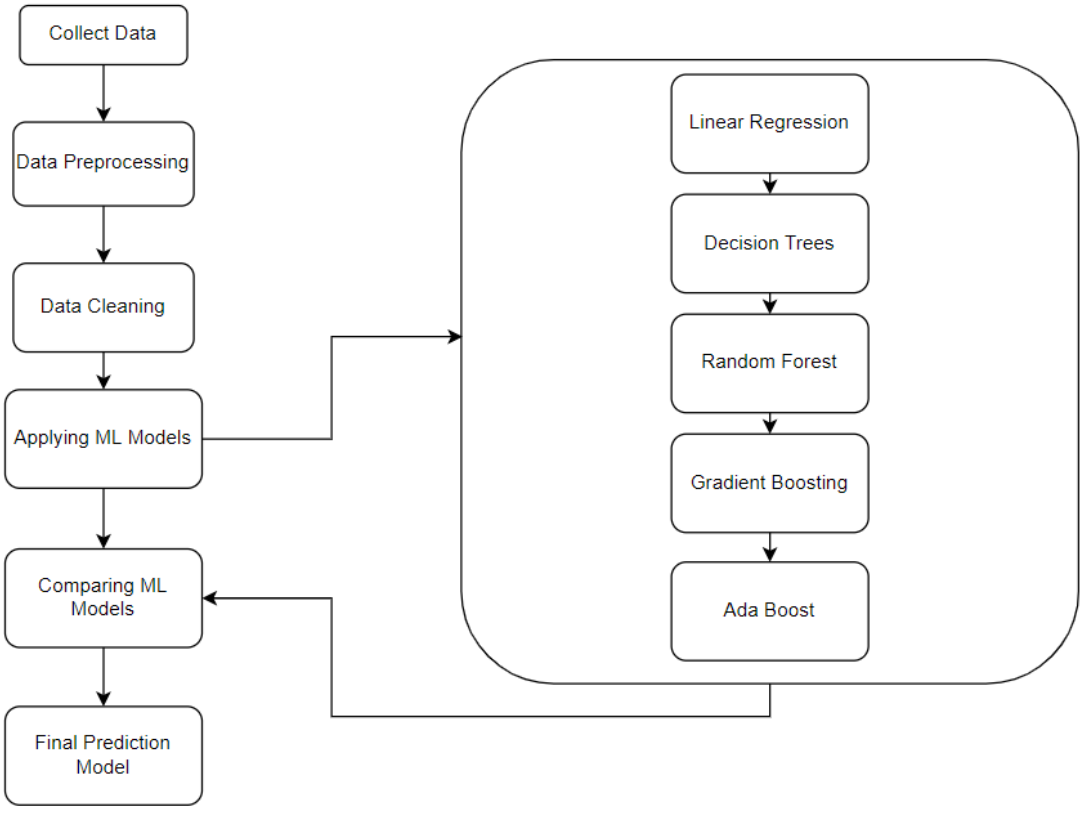


Figure 5.4: *Architecture Diagram*

5.5 EVALUATION PARAMETERS

The different parameters utilized for comparing and evaluating several algorithms:

- **Root Mean Square Error (RMSE)**: Root Mean Square Error (RMSE) is a standard way to measure the error of a model in predicting quantitative data. RMSE provides approximation up-to a certain value. RMSE tells you how concentrated the data is around the line of best fit. Root mean square error is commonly used in climatology, forecasting, and regression analysis to verify experimental results. Accuracy score equation is depicted below:

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

(EQUATION 2)

In the above Equation 2, (y_i) represents the true-label, ($y_i\hat{}$) represents the predicted label. In case of a multi-label classification, the RMSE value denotes the precision of the subset. The RMSE value is exactly 1.0 when the predicted values are equivalent to the true-value while as the value is 0 when they are not equivalent.

- **Coefficient of Determination**: Denoted by (R^2), it is a very essential statistical measurement utilized by various algorithms. It presents the relationship between the independent and the dependent variable by comparing their variances. This metric is a crucial measurement technique that informs of how effectively the figure fits the model and how effectively it equals with the real figures. The values it can accommodate varies from 0 and 1.

CHAPTER 6

EXPERIMENTAL ANALYSIS AND RESULTS

Various machine learning models were implemented and their outcomes were compared using some performance metrics to discover the most accurate working model/algorithm.

6.1 LINEAR REGRESSION

For the Linear Regression model, similar to any Machine Learning model, the initial step is to get the data and then pre-process it. A True vs Predicted values plot was created for linear regression as shown in Figure 6.1. Root mean square error value (RMSE) of 1843 is given by Linear regression. The coefficient of determination is calculated as 0.144 and the overall accuracy of the algorithm was less than 15%.

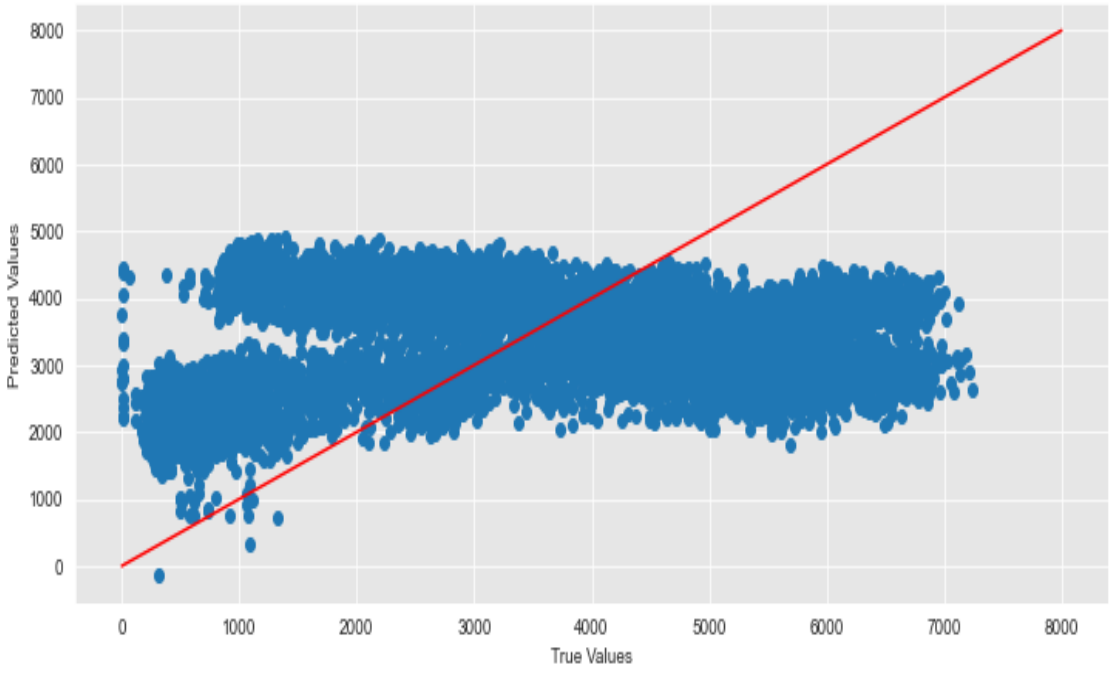


Figure 6.1: *True vs Predicted values (Linear Regression)*

6.2 DECISION TREES

For the Decision Trees model, again the initial step is to get the data and then pre-process it. A True vs Predicted values plot was created for decision trees as shown in Figure 6.2. Accuracy score of 93% is given by Decision tree regressor and Root mean square error value of 497. Coefficient of determination is calculated as 0.93.

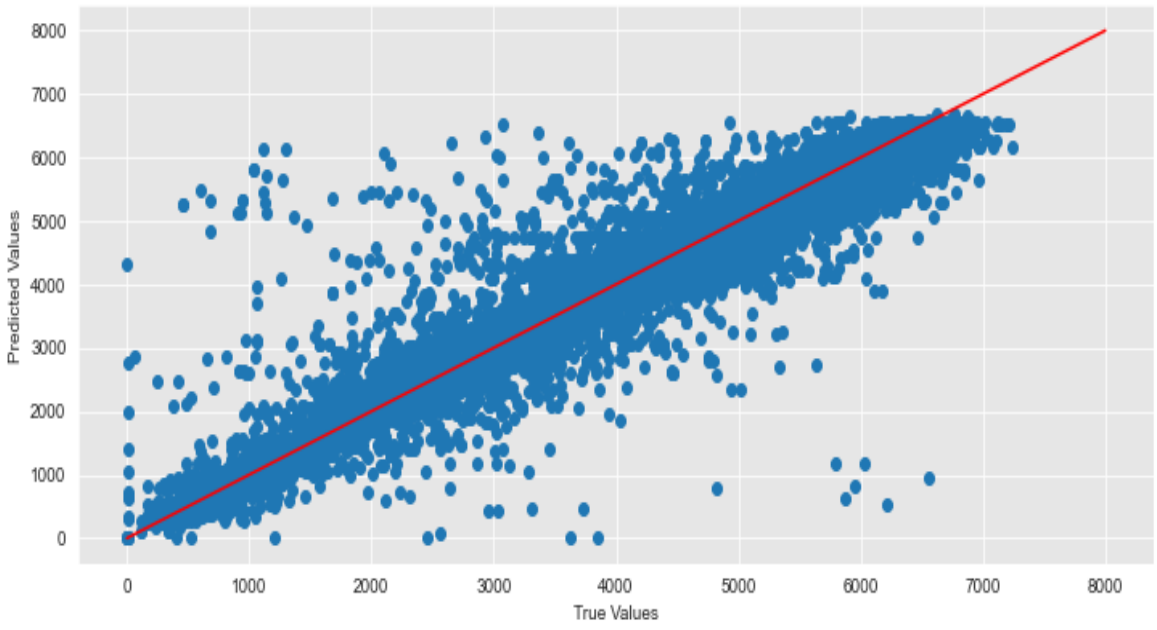


Figure 6.2: *True vs Predicted values (Decision Tree)*

6.3 RANDOM FOREST

A True vs Predicted values plot was created for random forest as shown in Figure 6.3. A Root mean square error value (RMSE) of 440 is given by random forest. The coefficient of determination is calculated as 0.95 and the overall accuracy of the algorithm was slightly less than 96%.

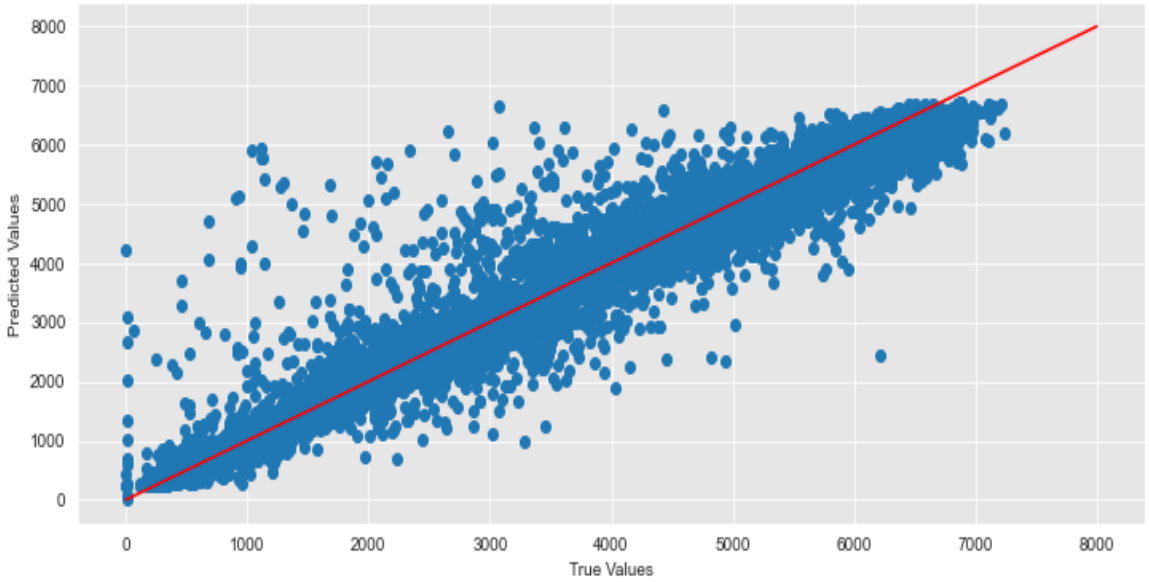


Figure 6.3: *True vs Predicted values (Random Forest)*

6.4 ADAPTIVE BOOST

Implementation of AdaBoost algorithm is done and the total estimators were around 60, thus producing a model with accuracy of 95% and RMSE value of 449. Coefficient of determination is calculated as 0.951.

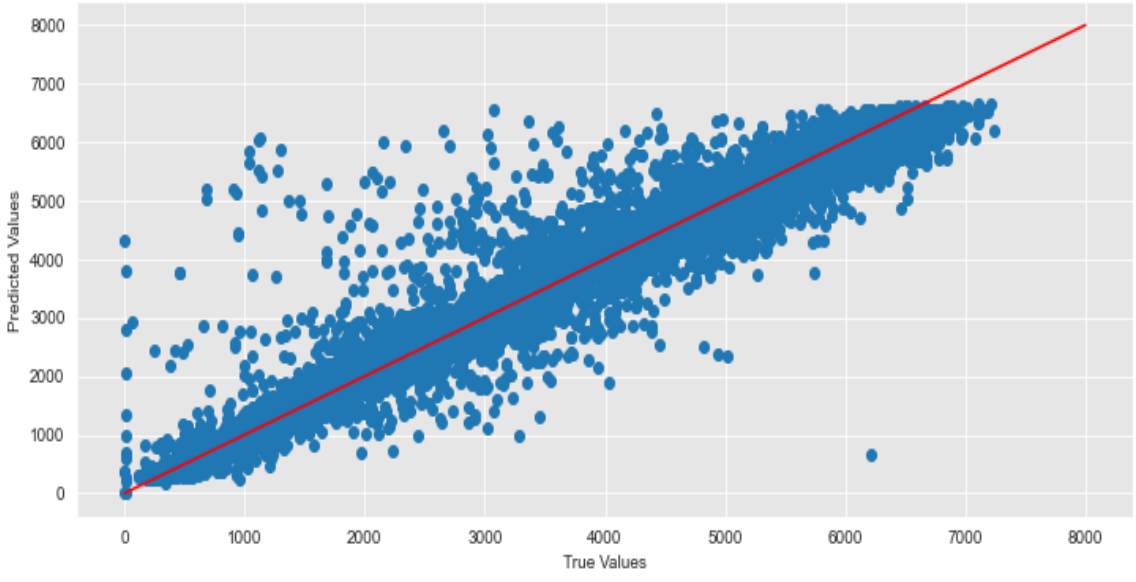


Figure 6.4: *True vs Predicted values (Ada Boost)*

6.5 GRADIENT BOOSTING

Gradient boosting was implemented initially and generated the lowest root mean square error in comparison with the other models. Figure 6.5 represents the true vs predicted values graph for Gradient boosting. From Figure6.5, we can clearly conclude that the majority of the points lie in the vicinity of the diagonal axis and hence generates an inflating accuracy score.

In gradient boosting, the total estimators were set around 595 and attained a 96 % accuracy score and RMSE value of 382. Coefficient of determination is calculated as 0.965 being highest in comparison with all the models.

Figure 6.5 clearly demonstrates how the straight line is in proximity to the points and fits most of the information and therefore has a less RMSE value. For other algorithms, the points are usually not in proximity of the line and hence have a higher RMSE score.

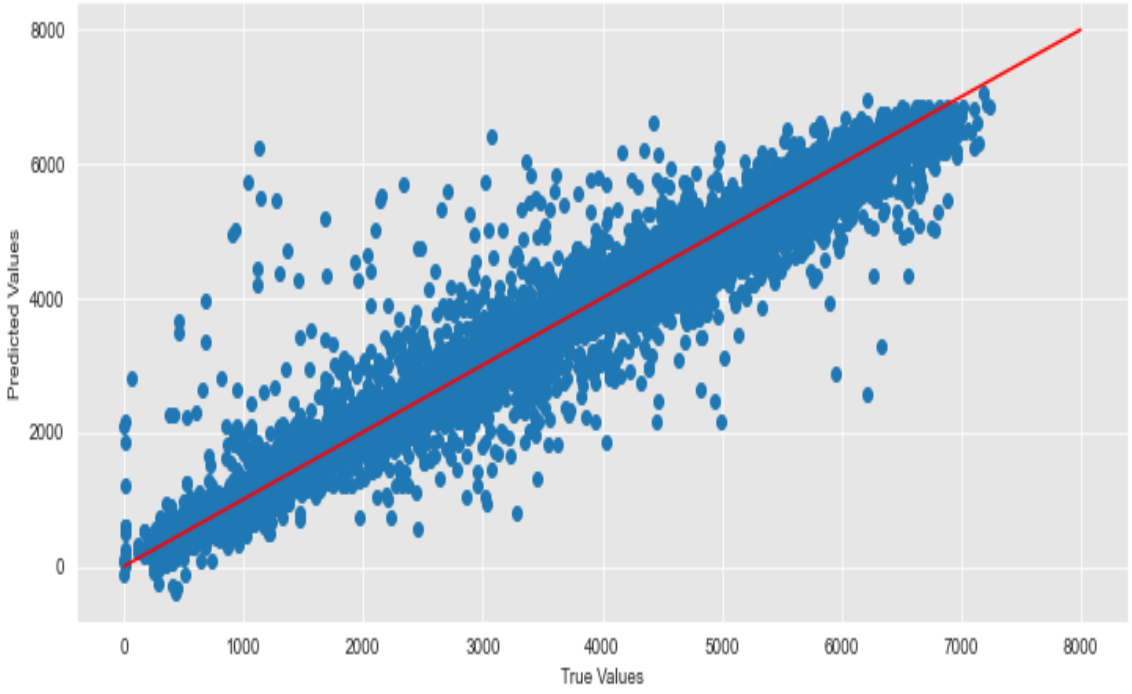


Figure 6.5: *True vs Predicted Values (Gradient boosting)*

6.6 COMPARISON

On the basis of the above model analysis, the following table showcases the comprehensive performances of the different machine learning algorithms on the basis of the performance parameter root mean square error. RMSE tells you how concentrated the data is around the line of best fit. **)**: Root Mean Square Error (RMSE) is a standard way to measure the error of a model in predicting quantitative data.The RMSE scores of different ML models is shown below in Table 6.1.

Table 6.1: RMSE Score for different models

RMSE SCORES	
Linear Regression	1843
Decision Trees	497
Random Forest	440
Gradient Boosting	382
Ada Boost	449

The below Table 6.2 showcases the comprehensive performances of the different machine learning algorithms on the basis of the performance parameter “Coefficient of Determination”. It helps in examining how differences in one variable can be explained by the difference in a second variable, when predicting the outcome of a given event. In other words, this coefficient, which is more commonly known as R-squared (or R^2), assesses how strong the linear relationship is between two variables.

Table 6.2: Coefficient of Determination Score for different models

COEFFICIENT OF DETERMINATION	
LINEAR REGRESSION	0.143
DECISION TREES	0.937
RANDOM FOREST	0.951
GRADIENT BOOST	0.963
ADA BOOST	0.949

Gradient boosting was implemented initially and generated the lowest root mean square error in comparison with the other models. From Figure 6.5, we can clearly conclude that the majority of the points lie in the vicinity of the diagonal axis and hence generates an inflating accuracy score.

For other algorithms, the points are usually not in proximity of the line and hence have a higher RMSE score. Hence, gradient boosting is the most accurate model and can be used in the final prediction model.

6.7 ANALYSIS OF TRAFFIC FLOW DEPENDANCE ON VARIOUS ATTRIBUTES

Using Jupyter notebook, we plot different graphs showing the dependence of traffic on different weather conditions, time of the year, holidays, population of an area, etc. Analysis of the outcomes and the graphs were done to infer some understanding regarding the dependance of the traffic on certain attributes.

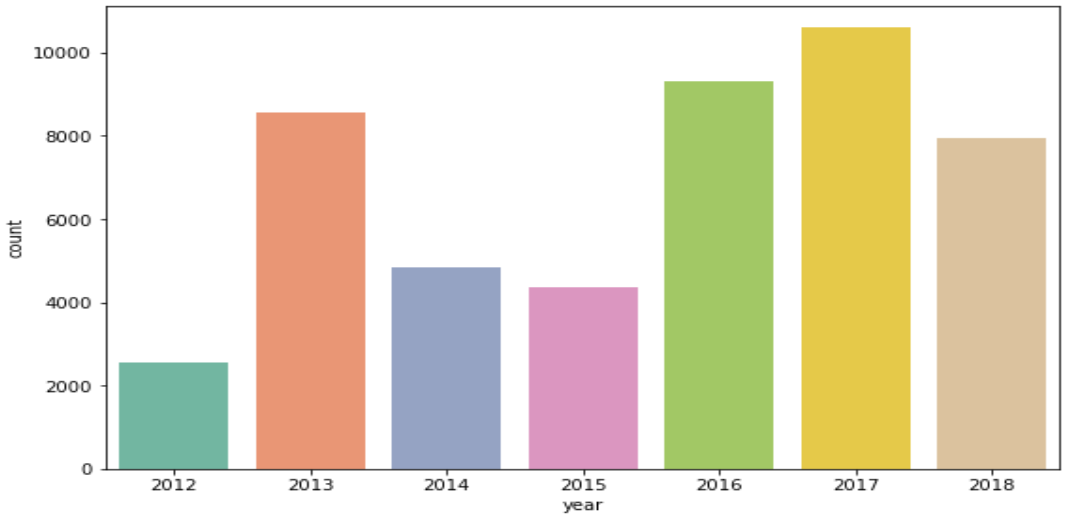


Figure 6.7.1: Traffic flow vs years

Figure 6.7.1 shows the dependence of the traffic flow across various years from the period of 2012-2018.

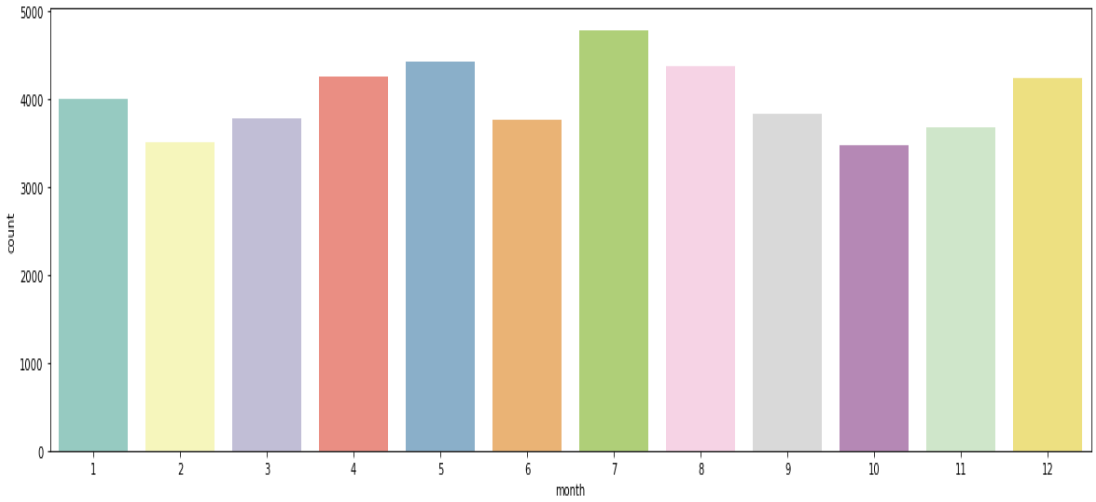


Figure 6.7.2: *Traffic flow vs months*

Figure 6.7.2 depicts the dependence of the traffic flow across the various months of a particular year. The X-axis represents the months 1-12 of a year.

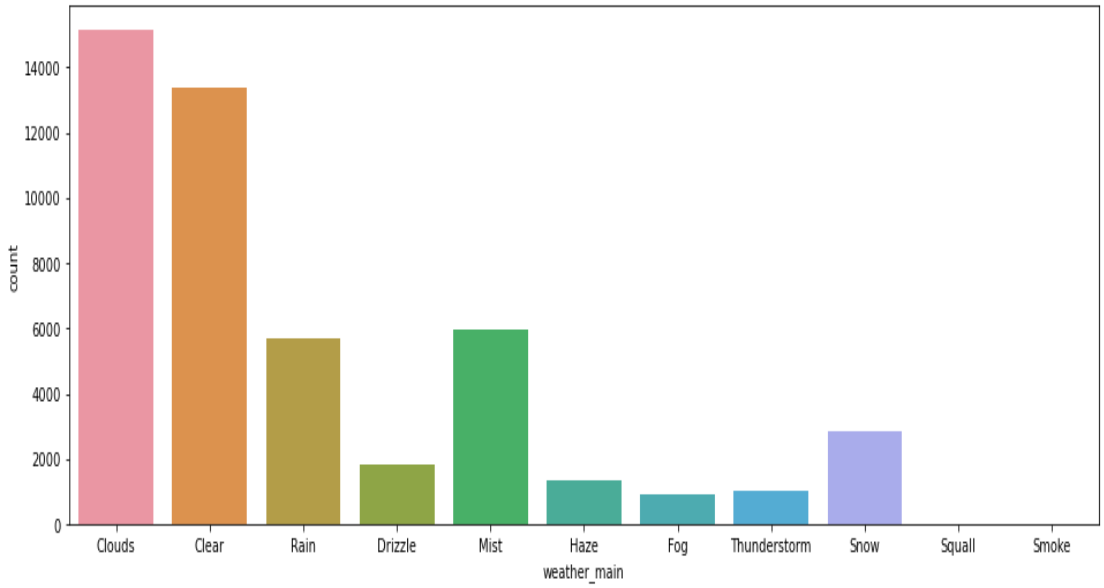


Figure 6.7.3: *Different weather conditions traffic count*

Figure 6.7.3 depicts the dependence of the traffic on different weather conditions. The X-axis consists of various weather conditions. Here we can see that the traffic flow is usually at its peak during clear or cloudy weather while-as during foggy and hazy weather, the traffic is quite minimal.

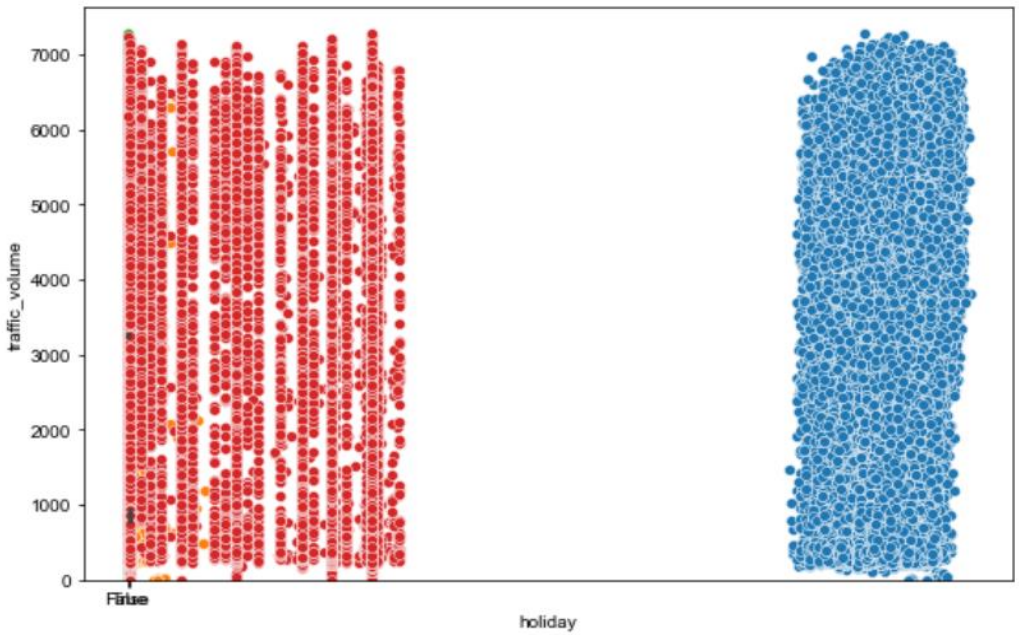


Figure 6.7.4: *Scatterplot of traffic vs holidays.*

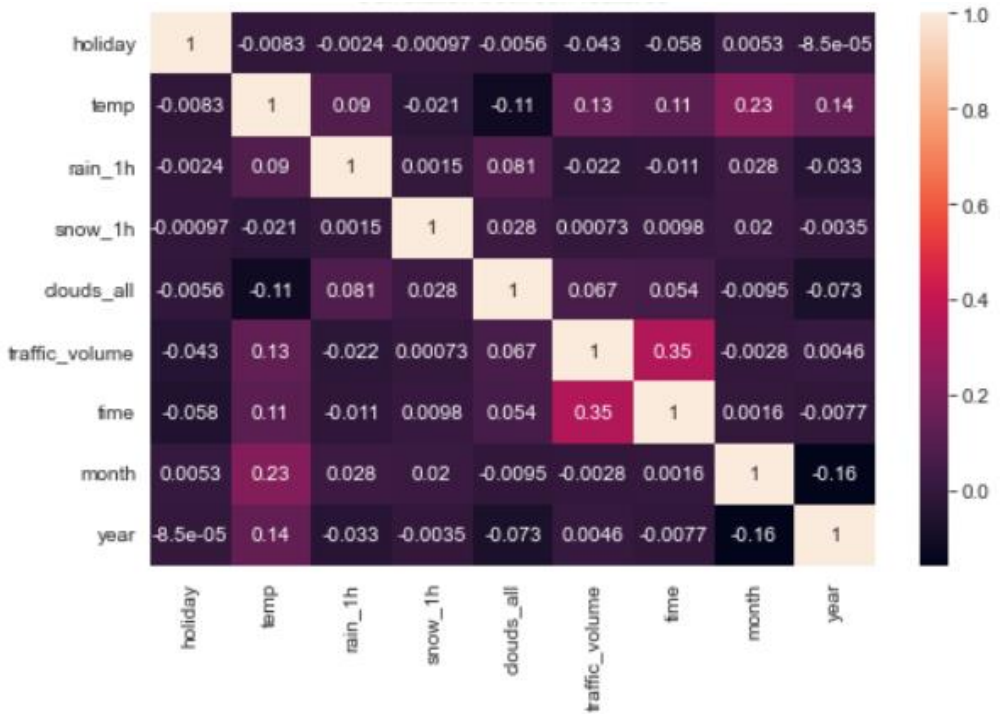


Figure 6.7.5: *Correlation between features*

Figure 6.7.4 depicts the scatterplot of dependence of traffic flow on the holidays across a particular year while-as Figure 6.7.5 shows the correlation between different features.

CHAPTER 7

CONCLUSION

7.1 CONCLUSION

Analysis and prediction of the flow of traffic is done using various ML models and several models were collated formed on the accuracy score and the RMSE value. Based on the inferences, the best performing algorithm was gradient boosting algorithm with 96.3 % accuracy and RMSE value of 382. Analysis of the outcomes and the graphs was done to infer some understanding regarding the dependance of the traffic on certain attributes.

Following the research-work and after analysis of the coefficient of determination and accuracy score of the various ML models, it can be concluded that for our interest the gradient-boosting algorithm produced superior results among all other algorithms used.

Post selection of our ML model, prediction and analysis of the traffic flow patterns utilizing this dataset was successfully implemented. Useful analysis and outcomes were derived from the plots showing the dependency of the traffic flow on the various attributes.

With the help of the plots derived earlier, we can conclude that the traffic flow is usually at its peak during clear or cloudy weather while-as during foggy and hazy weather, the traffic is quite minimal. Also attributes like holidays heavily influence the traffic conditions. We notice that the traffic is usually high during the holidays and the fuel-prices also sky-rocket. Also, we observe that the traffic is gradually increasing over the course of time maybe because due to the fact that overall number of cars have increased.

Apart from the technical knowledge, a huge importance was given toward the development of soft skills. time management and most importantly coordination between the partners and our guide.

The knowledge in this area was critical to understand the root cause of the problem which was eventually tackled by the different ML models.

7.2 FUTURE ENHANCEMENTS

For future learning, one can deduce more insights by taking into consideration more traffic situations, making use of different sensors and vehicle monitoring devices. One can implement much more advanced and higher performing machine-learning algorithm by obtaining a dynamic dataset with instantaneous statistics.

During the future, a complex web application can be developed wherein it might have an increased number of features.

REFERENCES

[1] LILIAN PUN1, PENGXIANG ZHAO 2, AND XINTAO LIU1, "A Multiple Regression Approach for Traffic Flow Estimation "

[2] BIN FENG 1, JIANMIN XU 1, YONGJIE LIN 1, (Member, IEEE), AND PENGHAO LIA Period-Specific Combined Traffic Flow Prediction Based on Travel Speed Clustering."

[3] SHANMEI LI, CHAO WANG, AND JING WANG "Exploring Dynamic Characteristics of Multi-State Air Traffic Flow a Time Series Approach "

[4] GUOWEN DAI1, CHANGXI MA 1, AND XUECAI XU "Short-Term Traffic Flow Prediction Method for Urban Road Sections Based on Space Time Analysis and GRU Air Traffic Flow: A Time Series Approach Prediction Based on Travel Speed Clustering.

[5] JIYAO AN, (Member, IEEE), LI FU, MENG HU, WEIHONG CHEN, AND JIAWEI ZHAN, "A Novel Fuzzy-Based Convolutional Neural Network Method to Traffic Flow Prediction with Uncertain Traffic Accident Information"

[6] RONGHAN YAO, WENSONG ZHANG, AND DONG ZHANG, "Period Division-Based Markov Models for Short-Term Traffic Flow Prediction."

[7] Lizong Zhang, Nawaf R Alharbe, Guangchun Luo, Zhiyuan Yao, and Ying Li "A Hybrid Forecasting Framework Based on Support Vector Regression with a Modified Genetic Algorithm and a Random Forest for Traffic Flow Prediction"

[8] SAIF EDDIN G. JABARI 1,2, DEEPTHI MARY DILIP DIANCHAO LIN AND BILAL THONNAM THODI2 s, "Learning Traffic Flow Dynamics Using Random Fields."

[9] YIXUAN MA, ZHENJI ZHANG AND ALEXANDER IHLER, "Multi-Lane Short-Term Traffic Forecasting with Convolutional LSTM Network."

[10] CHUN AI, LIJUN JIA, MEI HONG, AND CHAO ZHANG "Short-Term Road Speed Forecasting Based on Hybrid RBF Neural Network with the Aid of Fuzzy System-Based Techniques in Urban Traffic Flow."

[11] XinqiangChen ,Huixing Chen , Yongsheng Yang , HuafengWuc, Wenhui Zhang , Jiansen Zhao , Yong Xiong “Traffic flow prediction by an ensemble framework with data denoising and deep learning model.”

[12] AzzedineBoukerche, Yanjie Tao, Peng Sun, “Artificial intelligence-based vehicular traffic flow prediction methods for supporting intelligent transportation systems.”

[13] Huakang Lu, Zuhao Ge, Youyi Song, Dazhi Jiang, Teng Zhou, Jing Qin, “A Temporal-aware LSTM Enhanced by Loss-switch Mechanism for Traffic Flow Forecasting”

[14] SaiqunLu ,Qiyan Zhang , Guangsen Chen , Dewen Seng "A combined method for short-term traffic flow prediction based on recurrent neuralnetwork"

[15] Linjiang Zheng,Jie Yang , Li Chen , Dihua Sun , Weining Liu "Dynamic spatial-temporal feature optimization with ERI big data for Short-term traffic flow prediction".

APPENDIX-A

CODE

```
# import all required libraries for reading, analysing and visualizing data
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from math import sqrt
train_df = pd.read_csv('input/Metro_Interstate_Traffic_Volume.csv')
print('Dataset shape: ', train_df.shape)
train_df.info()
train_df.head()
train_df.describe(include = 'all')
train_df.isnull().sum()

# convert the date_time column to datetime type
train_df['date_time'] = pd.to_datetime(train_df['date_time'])
train_df['time'] = train_df['date_time'].dt.hour
fig, (axis1,axis2) = plt.subplots(2, 1, figsize = (20,12))
sns.countplot(x = 'time', data = train_df, ax = axis1, palette="Set3" )
sns.lineplot(x = 'time', y = 'traffic_volume', data = train_df, ax = axis2);

### Month vs Traffic Volume

train_df['month'] = train_df['date_time'].dt.month
```

```
fig, (axis1,axis2) = plt.subplots(2, 1, figsize = (20,12))
sns.countplot(x = 'month', data = train_df, ax = axis1, palette="Set3")
sns.lineplot(x = 'month', y = 'traffic_volume', data = train_df, ax = axis2,)
### Year vs Traffic Volume
train_df['year'] = train_df['date_time'].dt.year
fig, (axis1,axis2) = plt.subplots(1, 2, figsize = (20,6))
sns.countplot(x = 'year', data = train_df, ax = axis1, palette="Set2")
sns.lineplot(x = 'year', y = 'traffic_volume', data = train_df, ax = axis2);

###Day vs Traffic Volume
train_df['day'] = train_df['date_time'].dt.day_name()
fig, (axis1,axis2) = plt.subplots(1, 2, figsize = (20,6))
sns.countplot(x = 'day', data = train_df, ax = axis1)
sns.lineplot(x = 'day', y = 'traffic_volume', data = train_df, ax = axis2);
train_df['holiday'].value_counts()
z = lambda x: False if x == 'None' else True
train_df['holiday'] = train_df['holiday'].apply(z)
fig, (axis1,axis2) = plt.subplots(1, 2, figsize = (20,6))
sns.countplot(x = 'holiday', data = train_df, ax = axis1)
sns.barplot(x = 'holiday', y = 'traffic_volume', data = train_df, ax = axis2);
(train_df['temp'] == 0).sum()
train_df = train_df[train_df['temp'] != 0]
sns.scatterplot(x = 'temp', y = 'traffic_volume', data = train_df);
(train_df['rain_1h'] > 100).sum()
train_df = train_df[train_df.rain_1h < 100]

sns.set_style("darkgrid", {"axes.facecolor": ".9"})
```

```
sns.scatterplot(x = 'rain_1h', y = 'traffic_volume', data = train_df,);
sns.scatterplot(x = 'snow_1h', y = 'traffic_volume', data = train_df);
sns.scatterplot(x = 'clouds_all', y = 'traffic_volume', data = train_df);
```

###Short Weather Description vs Traffic Volume

```
fig, (axis1,axis2) = plt.subplots(2, 1, figsize = (16,12))
sns.countplot(x = 'weather_main', data = train_df, ax = axis1)
sns.lineplot(x = 'weather_main', y = 'traffic_volume', data = train_df, ax = axis2);
train_df['weather_description'].value_counts()
plt.figure(figsize = (20,6))
sns.lineplot(x = 'weather_description', y = 'traffic_volume', data = train_df);
correlation
plt.figure(figsize=(8, 5))
plt.title('Correlation between features')
sns.heatmap(train_df.corr(), annot = True);
```

Preprocessing of data

```
from sklearn.preprocessing import LabelEncoder

# drop the unrequired columns
train_df.drop(['date_time', 'weather_description'], axis = 1, inplace = True)

# convert values of day column to numerical format

encoder = LabelEncoder()
```

```
train_df['day'] = encoder.fit_transform(train_df['day'])

# subtract 242 from the temp column as there is no temperature below it
train_df['temp'] = train_df['temp'] – 242

# convert the values of weather_main column to numerical format
encoder = LabelEncoder()
train_df['weather_main'] = encoder.fit_transform(train_df['weather_main'])

# import the required modules
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2_score, mean_squared_error
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.ensemble import AdaBoostRegressor
(X, Y) = (train_df.drop(['traffic_volume'], axis = 1).values,
train_df['traffic_volume'].values)

# Scale the values
scaler = StandardScaler()
X = scaler.fit_transform(X)
(X_train, X_val, Y_train, Y_val) = train_test_split(X, Y)

print("X_train shape:" + str(X_train.shape))
```

```
print("Y_train shape:" + str(Y_train.shape))
print("X_val shape:" + str(X_val.shape))
print("Y_val shape:" + str(Y_val.shape))

# DataFrame to store the RMSE scores of various algorithms
results = pd.DataFrame(columns = ['RMSE'])
# helper function to evaluate a model
def evaluate_model(regressor, name):

 # train and test scores
    train_score = round(regressor.score(X_train, Y_train), 2)
    val_score = round(regressor.score(X_val, Y_val), 2)

# predicted output
    Y_pred = regressor.predict(X_val)
    print(name + ' Train score: ', train_score)
    print(name + 'Test score: ', val_score)
    print('Root Mean Squared error: ', sqrt(mean_squared_error(Y_val, Y_pred)))
    print('Coefficient of determination: ', r2_score(Y_val, Y_pred))

# add the current RMSE to the scores list
    results.loc[name] = sqrt(mean_squared_error(Y_val, Y_pred))

# plot predicted vs true values
    x_points=np.linspace(0,8e3)

    plt.figure(figsize=(12,5))
```

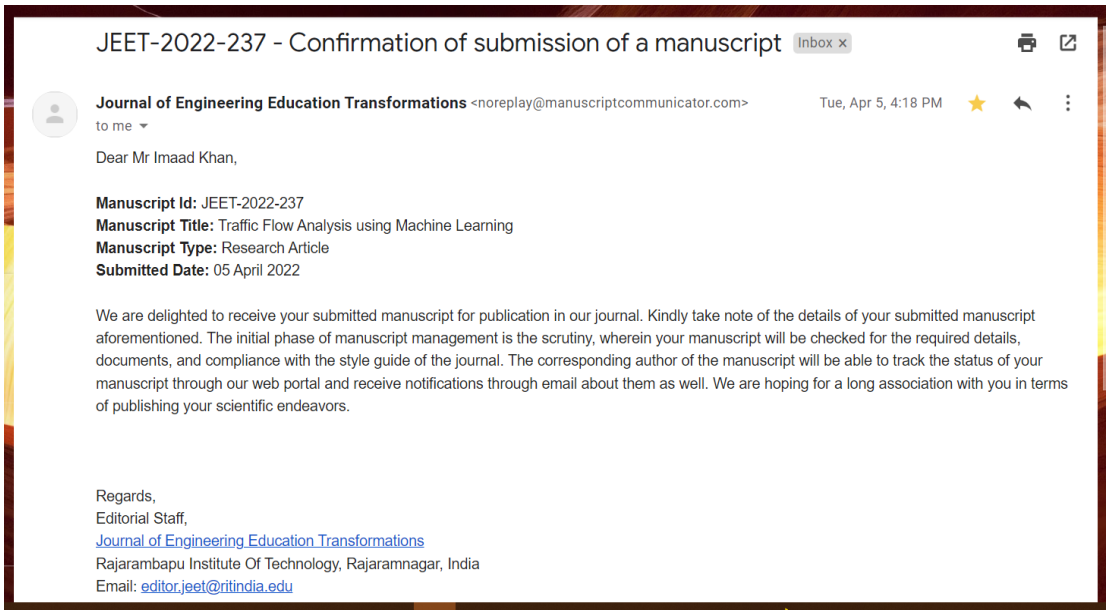
```
    plt.plot(x_points, x_points, color='r')
    plt.scatter(Y_val, Y_pred)
    plt.xlabel('True Values')
    plt.ylabel('Predicted Values')
    plt.title('True Values Vs Predicted Values');
lireg = LinearRegression()
lireg.fit(X_train, Y_train)
# evaluate the Regressor
evaluate_model(lireg, 'Linear Regression')
dtreg = DecisionTreeRegressor(max_depth = 12)
dtreg.fit(X_train, Y_train)

# Evaluate the Regressor
evaluate_model(dtreg, 'Decision Tree')
# n_estimators - The number of trees in the forest.
# min_samples_split - The minimum number of samples required to split an internal node
rfreg = RandomForestRegressor(n_estimators = 60, max_depth = 13,
min_samples_split = 5)
rfreg.fit(X_train, Y_train)
# evaluate the Regressor
evaluate_model(rfreg, 'Random Forest')
# n_estimators - The number of boosting stages to perform.
# max_depth - maximum depth of the individual regression estimators.
gbreg = GradientBoostingRegressor(n_estimators=497, max_depth=10)
gbreg.fit(X_train, Y_train)

# evaluate the Regressor
```

```
evaluate_model(gbreg, 'Gradient Boosting')

# n_estimators - The number of trees in the forest.

# learning_rate - Learning rate shrinks the contribution of each classifier by learning_rate.

adareg = AdaBoostRegressor(base_estimator=dtreg, n_estimators=60, learning_rate=0.005)

adareg.fit(X_train, Y_train)

# evaluate the Regressor

evaluate_model(adareg, 'Ada Boost')

results

plt.plot(gbreg.feature_importances_)

from keras.models import Sequential

from keras.layers import Dense, Dropout

from keras.wrappers.scikit_learn import KerasRegressor

from sklearn.model_selection import cross_val_score, KFold

from sklearn.pipeline import Pipeline

def nn_model ():

    model = Sequential()

    model.add(Dense(128, input_dim=10, kernel_initializer='normal', activation='relu'))

    model.add(Dense(256, kernel_initializer='normal', activation='relu'))

    model.add(Dense(256, kernel_initializer='normal', activation='relu'))

    model.add(Dense(1, kernel_initializer='normal'))

    model.compile(loss='mean_squared_error', optimizer='adam')

    return model

estimator = KerasRegressor(build_fn=nn_model, epochs=10, batch_size=5, verbose=0)

kfold = KFold(n_splits=10)
```

```
estimator.fit(X_train, Y_train)

# predicted output

Y_pred_nn = estimator.predict(X_val)

print('Root Mean Squared error: ', sqrt(mean_squared_error(Y_val, Y_pred_nn)))

print('Coefficient of determination: ', r2_score(Y_val, Y_pred_nn))
```

APPENDIX-B
PUBLICATION DETAILS

We submitted our research paper for publication in the journal, **Journal of Engineering Education Transformations**. We got the confirmation of our submission notification from the JEET stating our paper has been submitted for review and is currently with the Editor in Chief. Proof of publication sent email is attached in figure B.1.


JEET-2022-237 - Confirmation of submission of a manuscript Inbox ×

Journal of Engineering Education Transformations <noreplay@manuscriptcommunicator.com> Tue, Apr 5, 4:18 PM

to me

Dear Mr Imaad Khan,

Manuscript Id: JEET-2022-237
Manuscript Title: Traffic Flow Analysis using Machine Learning
Manuscript Type: Research Article
Submitted Date: 05 April 2022

We are delighted to receive your submitted manuscript for publication in our journal. Kindly take note of the details of your submitted manuscript aforementioned. The initial phase of manuscript management is the scrutiny, wherein your manuscript will be checked for the required details, documents, and compliance with the style guide of the journal. The corresponding author of the manuscript will be able to track the status of your manuscript through our web portal and receive notifications through email about them as well. We are hoping for a long association with you in terms of publishing your scientific endeavors.

Regards,
Editorial Staff,
Journal of Engineering Education Transformations
Rajarambapu Institute Of Technology, Rajaramnagar, India
Email: editor.jeet@ritindia.edu

Figure B.1: Confirmation of Submission

After the submission of our research paper in the journal, we got the confirmation and the status of our submission. Thereafter passing through the initial checking and the mentioned scrutiny, the status on their journal website mentions ‘under review by the EIC (Editor-in-Chief)’. The status is shown below in Figure B.2.

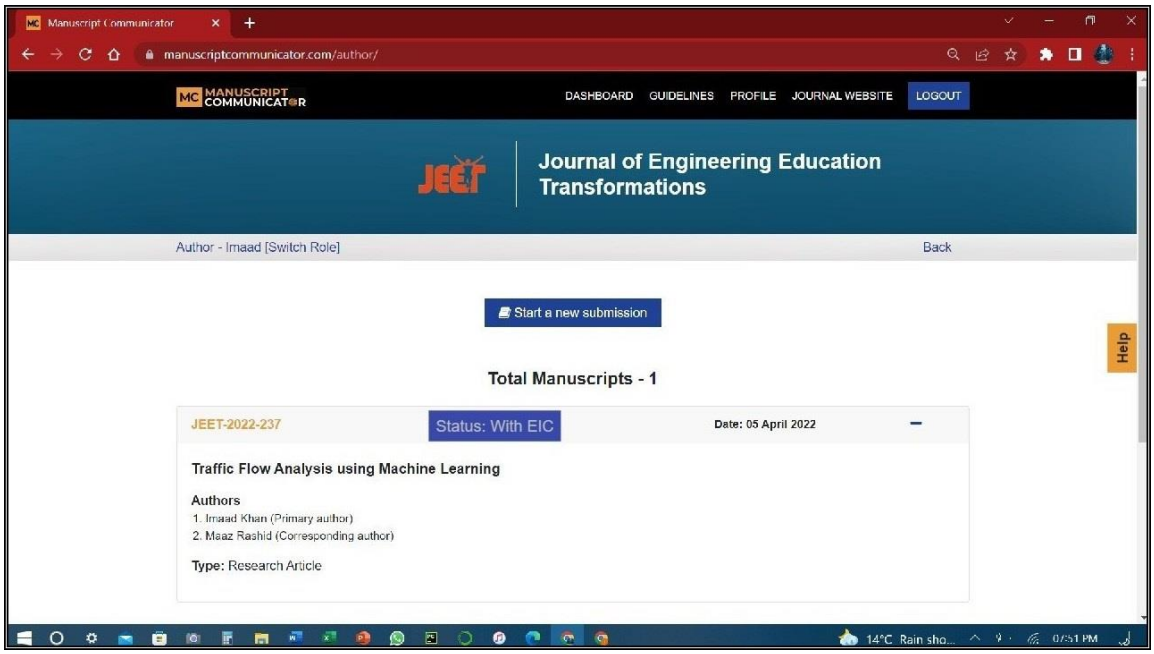


Figure B.2: Status of Submission

The cover page of our research paper has been attached below:

Traffic Flow Analysis Using Machine Learning

Mohammad Maaz Rashid
School of Computing
SRM Institute of Science and Technology
Chennai, India
mr8775@srmist.edu.in

Imaad Khan
School of Computing
SRM Institute of Science and Technology
Chennai, India
ik3746@srmist.edu.in

Dr T.K Sivakumar
Assistant Professor
School of Computing
SRM Institute of Science and Technology
Chennai, India
sivakumt2@srmist.edu.in

***Abstract*— Prediction and analysis of the traffic flow is a subject of utmost importance. The traffic department and Governments could make use of the crucial data and key-points to interchange the vehicular routes and expedite the traffic movement in a smooth and effective manner. This data can prevent congestions in the near future and help scale down the overall travelling time for an individual. Human lives can also be saved as the emergency facilities such as the ambulances, fire-fighting vehicles etc. would reach their respective emergency locations without any further delays. In order to control and eliminate the congestions and the gridlocks in a controlled and effective way, prediction and analysis of the traffic is paramount. This research paper will present various ML models such as Gradient boosting, Decision trees and a comparison will be drawn between them and the currently in-use primitive models like Linear Regression by utilizing different performance benchmarks such as the RMSE values.**

Keywords*— *Traffic-flow | Decision Tree| Linear regression | Gradient Boosting |Root mean square error (RMSE).

I. INTRODUCTION

Vehicular routes statistics and the related data to the road network is of utmost value for designing transport activities and other associated research activities. According to a study, commuters ordinarily spent over 79 hours caught in the traffic in 2017 alone in some parts of the world. Analyzing the vehicular movements helps in ascertain the commute of vehicles through various roads and re-routing them to different lanes and roads to reduce the occurrences of congestions. All the current and previous records and statistics pertaining to the vehicular traffic for a region can be utilized to identify the various road networks and the traffic patterns to predict the flow of the imminent and upcoming traffic.

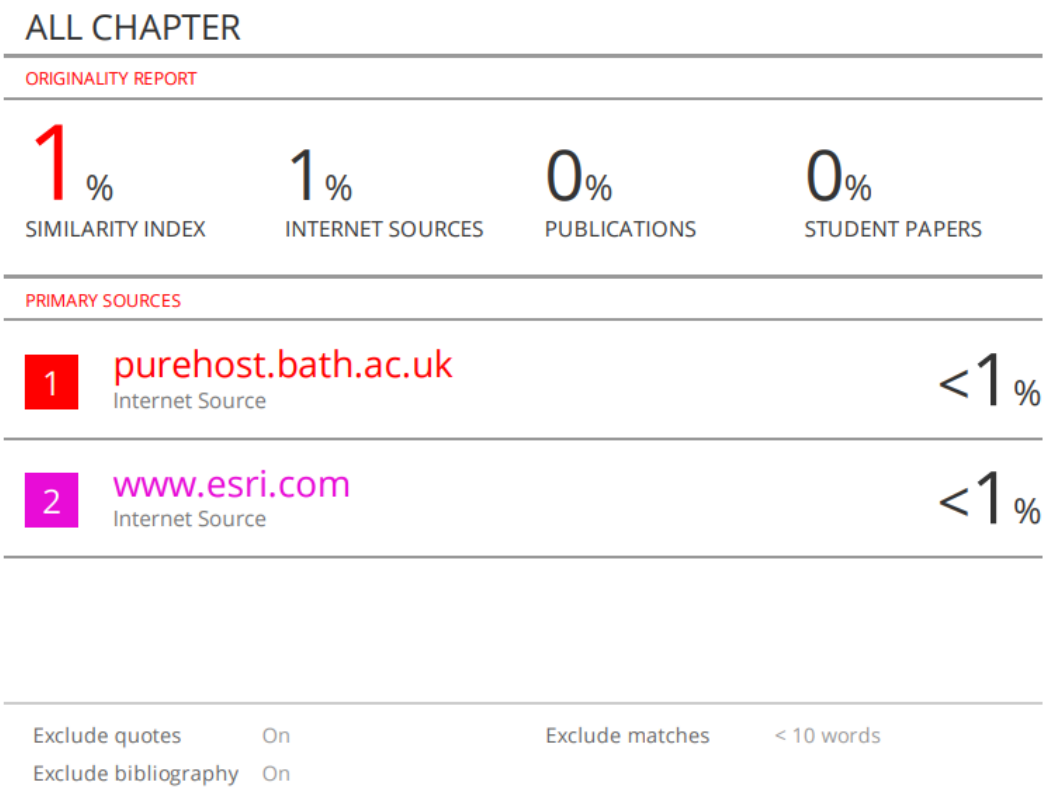
Traffic congestion reduces air pollution in the area as well as increased traffic pollution, and recent studies have shown that the death toll for motorists, commuters and people living near highways and traffic congested areas is very high. Our current knowledge of air pollution and its impact on traffic congestion is insufficient. Therefore, the study of motor prediction methods is very important in alleviating this difficulty. This can help traffic controllers control traffic. This exchange increases the time required for the trip and thus forces the fare to rise.

The first step is to accumulate all the traffic-related data for analysis. There are numerous approaches for gathering the data. To collect the data, various detectors and equipments are instated throughout different road networks to estimate the volume of traffic on a road at a particular instance. Equipments such as personal road courses, test vehicles or floating vehicle data (FCD), sidewalk detectors, closed-circuit television (CCTV), camera, photographs, are commonly used. In this study, we attempted to gather relevant data needed to predict traffic flow and consider the type of traffic that exists in India.

II. LITERATURE SURVEY

A. Over the few decades, various discrete approaches have unfolded many solutions to predict the traffic flux precisely. Initially, a regression perspective was put-forth by Mr. Liliian. In this method, one was able to forecast

Plagiarism report for our project report is shown below.



ALL CHAPTER

ORIGINALITY REPORT

1%	1%	0%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	purehost.bath.ac.uk Internet Source	<1%
2	www.esri.com Internet Source	<1%

Exclude quotes	On	Exclude matches	< 10 words
Exclude bibliography	On		

PLAGIARISM REPORT

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY **(Deemed to be University u/s 3 of UGC Act, 1956)**		
Office of Controller of Examinations		
REPORT FOR PLAGARISM CHECK ON THE DISSERTATION/PROJECT REPORTS FOR UG/PG PROGRAMMES (To be attached in the dissertation/project report)		
1	Name of the Candidate **(IN BLOCK LETTERS)**	**IMAAD ZAFFAR KHAN, MOHAMMAD MAAZ RASHID**
2	Address of the Candidate	SRM Hostels, SRM IST, 603203 Tamil Nadu **Mobile Number: 9790718068,8800355660**
3	Registration Number	RA1811003010850 RA1811003010866
4	Date of Birth	25/07/1999 13/04/1999
5	Department	Computer Science and Engineering
6	Faculty	
7	Title of the Dissertation/Project	Traffic Flow Analysis Using Different Machine Learning Algorithms.
8	Whether the above project/dissertation is done by	~~Individual~~ or group: (Strike whichever is not applicable) a) If the project/dissertation is done in group, then how many students together completed the project:2 b) Mention the Name & Register number of other candidates:

9	Name and address of the Supervisor/Guide	**Dr. T.K. SIVAKUMAR** Mail ID**: sivakumt2@srmist.edu.in** Mobile Number**: 9444202864**
10	Name and address of the Co-Supervisor/Co-Guide (if any)	**NA**
11	Software Used	Turnitin
12	Date of Verification	25th April 2022
13	**Plagiarism Details:(to attach the final report from the software)**	

Chapter	**Title of the Chapter**	**Percentage of similarity index (including self-citation)**	**Percentage of similarity index (Excluding self-citation)**	**% Of plagiarism after excluding Quotes, Bibliography, etc.**
1	Introduction	0%	0%	0%
2	Literature Review	0%	0%	0%
3	System analysis	0%	0%	0%
4	Methodologies	2%	2%	2%
5	Proposed Approach	3%	3%	3%
6	Experimental Analysis and Results	0%	0%	0%
7	Conclusions	0%	0%	0%
8				
9				
10				
Appendices		2%	2%	2%

<table>
<tr><td colspan="2">I / We declare that the above information has been verified and found true to the best of my / our knowledge</td></tr>
<tr><td>Signature of the Candidate</td><td>Name & Signature of the Staff
(Who uses the plagiarism check software)</td></tr>
<tr><td>Name & Signature of the Supervisor/Guide</td><td>Name & Signature of the Co-Supervisor/Co-Guide</td></tr>
<tr><td colspan="2">Dr. M. PUSHPALATHA
Name & Signature of the HOD</td></tr>
</table>