Illinois Institute of Technology

CS-584 Machine Learning

Traffic Flow Analysis in Different Weather Conditions

Shiva Sankar Modala(A20517528)
smodala1@hawk.iit.edu

Keerthana Reddy Mucherla(A20517254)
kmucherla@hawk.iit.edu

Narthu Sailavanya(A20516764)
snarthu@hawk.iit.edu

Dr. Yan Yan

TABLE OF CONTENTS

LIST OF FIGURES

1. Introduction

Traffic flow information and the data related to the network of roads is of great significance to plan the transportation and other associated tasks. In the year 2017, commuters spent more than 82 hours on an average wait in traffic from the data collected from the fifteen most congested cities. For the purpose of determining the vehicles flow and rerouting traffic on other paths to overcome the increased traffic, the act of traffic flow estimation is of utmost importance. We can predict the traffic by using historical data on the volume of traffic and applying various machine learning techniques to it.

Fig.1.1. Traffic in Chicago Downtown

Recent studies have revealed that the mortality rate for drivers, travelers, and residents who live close to major highways and busy regions is notably high. Traffic congestion on the roads also shows that ambient air quality is reduced and that there is an increase in vehicle emissions. We now have a limited understanding of how air pollution affects traffic congestion on roads. Therefore, the study of traffic forecast technologies is crucial for resolving these issues. The traffic controllers can better manage the flow of traffic as a result. This in turn increases the amount of time needed for the trip and imposes an increase in the cost of using public transportation.

In order for the algorithms to assess traffic data, appropriate data collection is required. For the collecting of traffic data, numerous techniques are available. To count the number of vehicles that pass through a road within a specific time period, several sensors are installed at the traffic lights and signals on the streets.

In order to collect traffic data, various methods might be used. Some of them are manual road studies, test cars or floating car data (FCD), street-side finders, closed circuit television (CCTV), cameras, and photos, the most often utilized of which are FCD and CCTV. We have attempted to collect the necessary information needed for projecting the traffic flow through this research, while also taking into account the type of traffic that exists in the United States.

1.1. Summary of the problem

In current life there are a lot of pressing issues one of which is traffic congestion which needs to be addressed as it is getting more serious for quite a while. The high volume of vehicles, the lack of infrastructure and the uneven distribution of the development are the primary reasons of traffic congestion.

Traffic flow analysis and forecasting has always been a topic of enormous significance. This information can be used by the government and traffic police to better guide traffic and ensure its efficient and free flow. The knowledge gained can stop traffic jams in the future and spare commute time and resources effectively.

Since emergency services like the ambulance can get to hospitals more quickly, it can also save lives. To successfully address these issues, traffic flow prediction is crucial. The mean square error (MSE), root mean square error (RMSE) are used to evaluate the performance of the project's numerous regression machine learning methodologies with those of existing machine learning algorithms.

1.2. Previous works and Methods

Over the years, various approaches have been put forth to precisely predict the volume of traffic flow. Few of them are –

1.2.1 Traffic Flow Estimation using Multiple Regression Approach

Multiple regression methodology was applied in this study. Using traffic statistics from Hong Kong, they took into account several traffic characteristics. They collected the data and tried to estimate the traffic flow using Multiple Regression, Support Vector Regression and Artificial Neural Network. The data set collected does not consider the weather condition and fails to estimate the traffic flow accurately.

1.2.2 Traffic Flow Prediction combined with Period-Specific Traffic

In this paper, a model was proposed based on traffic flow combined with time. The city of Zhongshan in the Chinese province of Guangdong is where the data is collected. The necessary prediction is based on the data that has been gathered at route intersections. In this research, k-means clustering is employed for identifying the travel speed timeseries to split a day into multiple time slots in order to predict traffic flow.

1.2.3 Prediction of Traffic Flow using Gated Recurrent Unit

In this study, the researchers developed a model that combines gated recurrent units(GRU) and spatio-temporal analysis. Following that, the traffic is predicted using the spatio-temporal data using the GRU.

2. Problem Description

Whether the existing infrastructure can deal with the increased population is a cause of concern that needs to be addressed by each government. The public vehicles of transport like the trams or trains are not accessible to everyone, especially in the agricultural nations. In order to adapt to these troubles, the public authority should take measures to divert the traffic to less crowded lanes and the traffic should be evenly distributed in order to minimize congestion.

Thus, diverting traffic and managing it is of paramount importance in order to prevent fatalities on the roads and improve time utilization. If the citizens of a country are spending their time on the roads stuck in the traffic jams, it can have a harmful impact on the economy of the nation and lead to less development. Thus, it is in the interest of the nation to devise methods to analyze the traffic flow and predict the traffic flow in order to better manage it in the future.

Considering the existing systems, various researchers have proposed different models in order to tackle this problem. Few have worked on a model that has used the weather data, population, locality and holiday parameters altogether as these are the most fundamental factors to determine the flow of traffic and gives us a clear picture of the traffic that can be expected on the streets. Also most have primarily used Machine learning models for predicting the traffic flow pattern.

2.1. Data Description

Over the years, various approaches of predicting traffic flow volume have been put forth. We used the Metro Interstate Traffic Volume Data Set in our project. This dataset is considered because it includes a variety of elements, including weather information and is way clear than its predecessor. It contains hourly traffic data along the Minneapolis–St. Paul route.

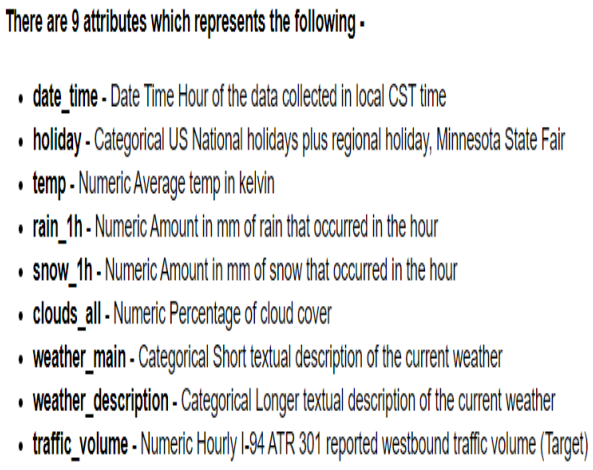

Fig.2.1. Attributes in Dataset

2.2. Algorithms used

2.2.1 Linear Regression

Linear Regression is simple machine learning algorithm used for solving regression problems. It shows the points on a 2-D axis and identifies the best-fit line by determining which line will pass through the most points. In order to determine the best fit line through the data and predict values, a linear model is used in this really simple approach.

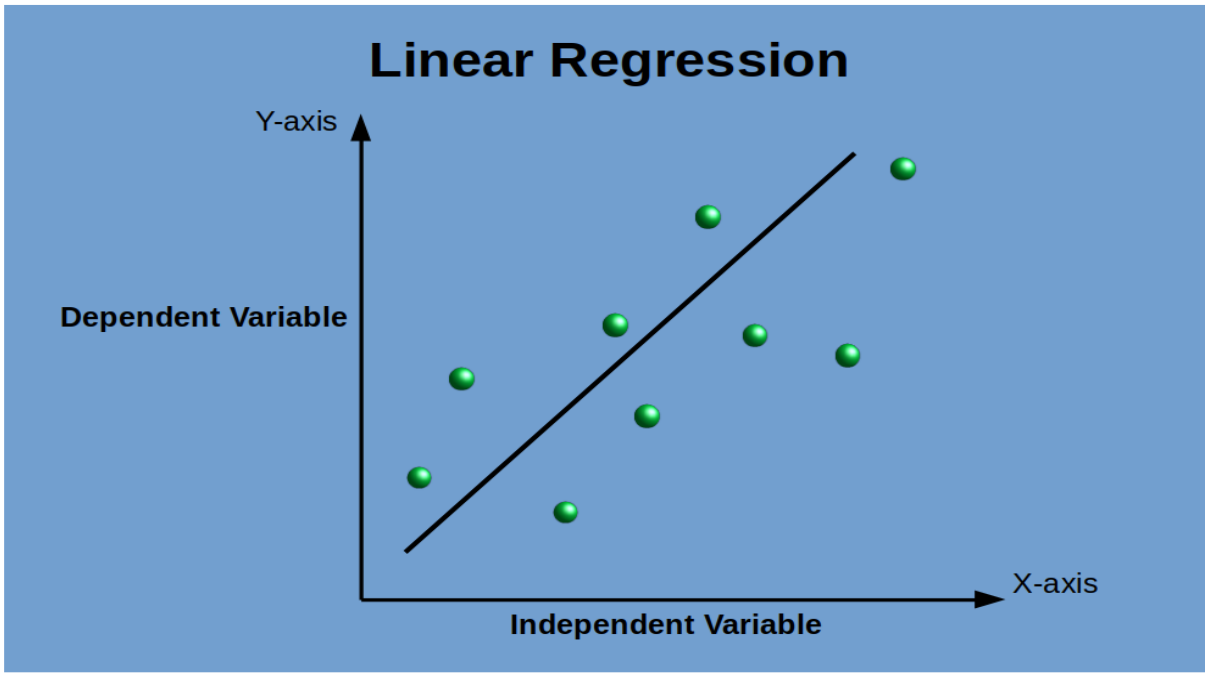

Fig.2.2. Linear Regression

2.2.2 Decision Tree

Decision Trees algorithm falls under the category of supervised learning algorithm. This method is applied to classification and prediction. A decision tree reaches its decision by performing a sequence of tests. Decision tree contains two types of nodes – Decision node and Leaf node. Decision node is the root node which will take all the decisions. Leaf node is the last node which does not have any other branches from there.

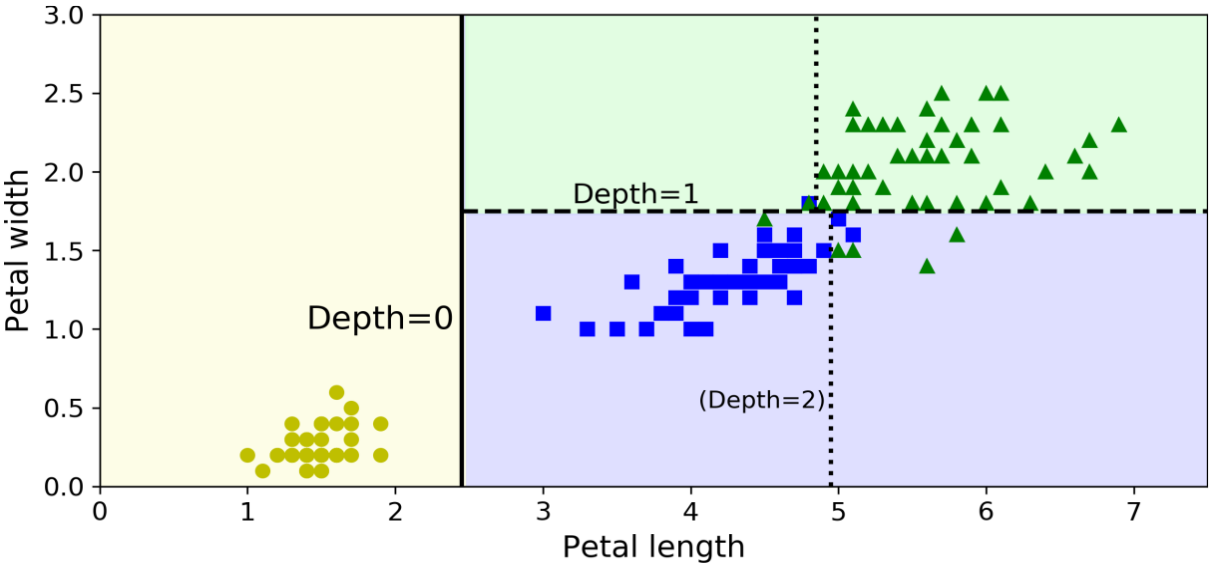

Fig.2.3. Decision Tree

2.2.3 Random Forest

In this approach, a group of decision trees are chosen at random from a trained set, and a vote is then generated at random using all the decision trees. The Random Forest Algorithm creates the final output by merging the results of the numerous

Decision Trees. Random Forest is combination of Decision trees which is trained to get the best prediction after combining all the individual trees.

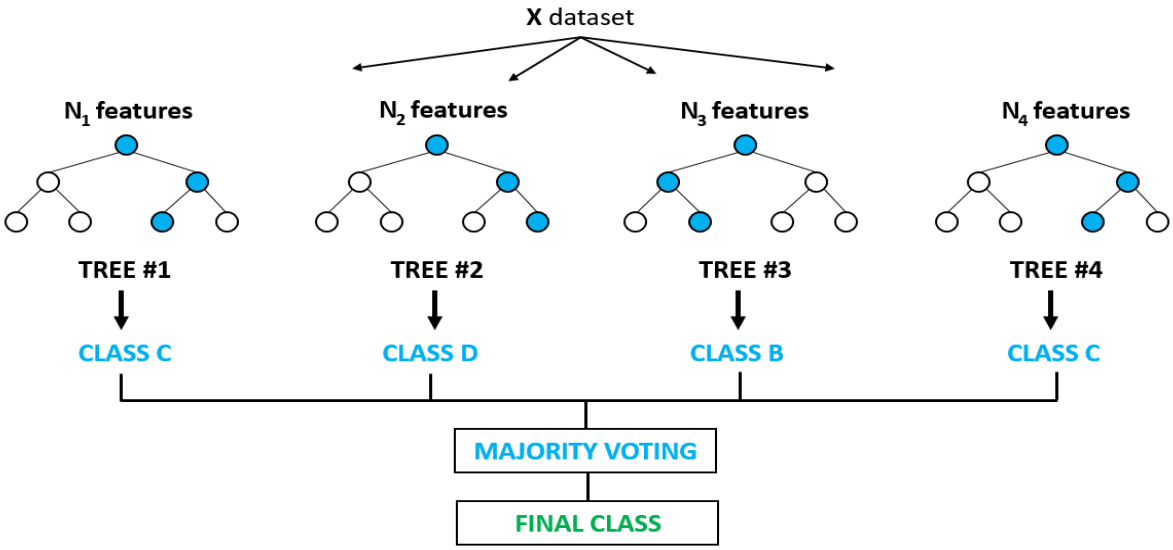

Fig.2.4. Random Forest

2.2.4 Gradient Boosting

Gradient boosting is a type of machine learning algorithm that uses boosting. It selects the best possible next model by combining it with the previous models and thus it reduces the error considerably. Gradient Boosting gradually, additively, and sequentially trains a large number of models. The loss function is a metric that shows how well the coefficients of the model fit the original data.

$$y_i^p = y_i^p + \alpha * \delta \sum (y_i - y_i^p)^2 / \delta y_i^p$$

which becomes, $y_i^p = y_i^p - \alpha * 2 * \sum (y_i - y_i^p)$

where, α is learning rate and $\sum (y_i - y_i^p)$ is sum of residuals

Fig.2.5. Gradient Boosting

2.2.5 AdaBoost

AdaBoost Algorithm is a boosting algorithm. It assigns equal importance to each of the observations in the beginning. After evaluating and checking the first tree, it increases the significance of other observation and later decreases the significance of these observations which are easier for classification. The other tree then uses this new set of weighted data in order to make a new tree. It can therefore increase the first tree's prediction accuracy. The first tree and the second tree are combined to create the new model. AdaBoost learning algorithm picks a few crucial traits from a wide pool in order to deliver convincing results.

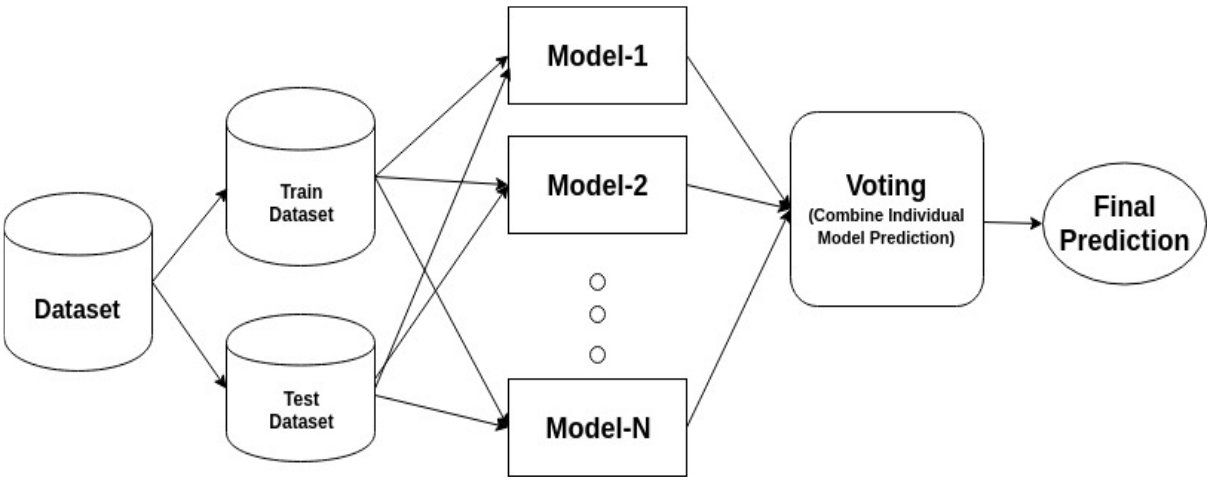

Fig.2.6. AdaBoost

3. Implementation

3.1. Data Visualization

The data will be visualized through different graphs and then correlation matrix is plotted to find correlation between the attributes.

3.1.1 Traffic Flow for different weather

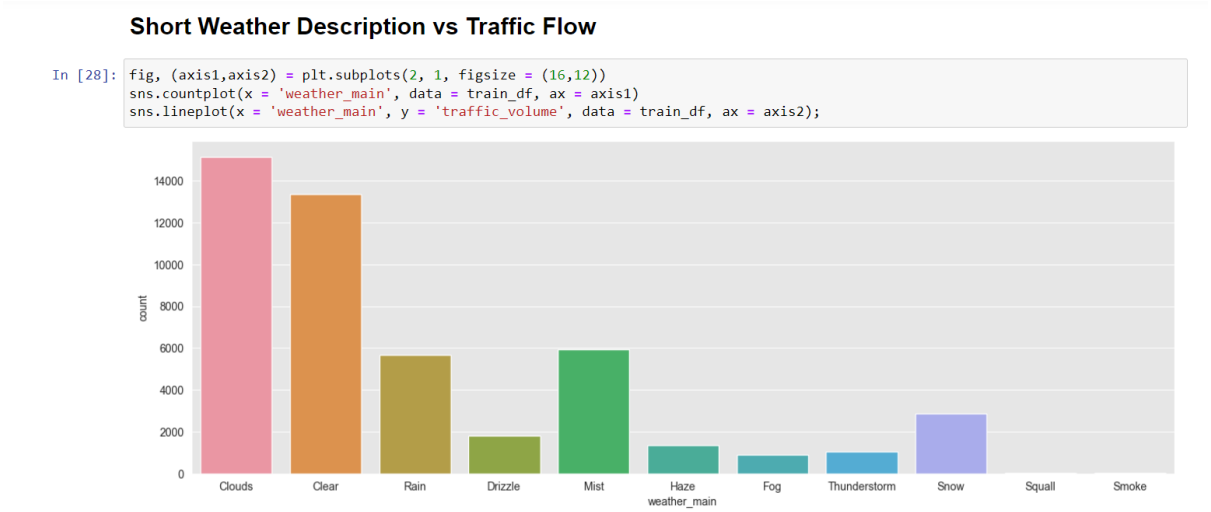

Fig.3.1. Traffic Flow for different weather

3.1.2 Day VS Traffic Flow

Day vs Traffic Flow

```
In [15]: train_df['day'] = train_df['date_time'].dt.day_name()

In [16]: 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);
```

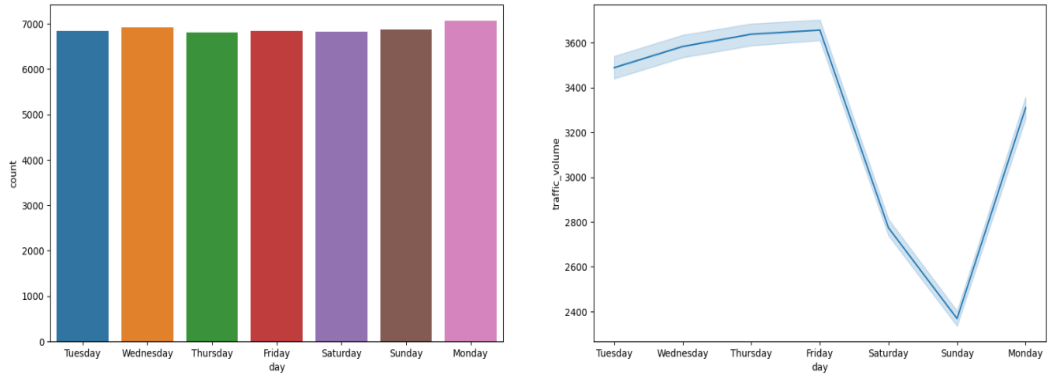

Fig.3.2. Day VS Traffic Flow

3.1.3 Time VS Traffic Flow

Time vs Traffic Flow

```
In [9]: train_df['time'] = train_df['date_time'].dt.hour

In [10]: 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);
```

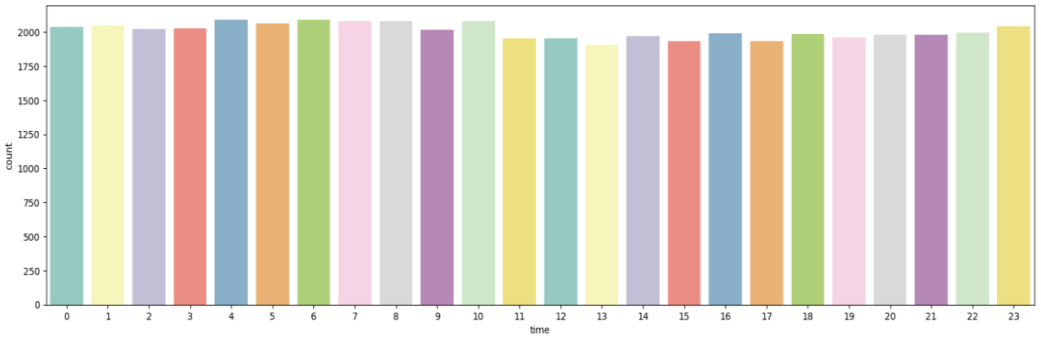

Fig.3.3. Time VS Traffic Flow

3.1.4 Month VS Traffic Flow

Month vs Traffic Flow

```
In [11]: train_df['month'] = train_df['date_time'].dt.month

In [12]: 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,)

Out[12]: <AxesSubplot:xlabel='month', ylabel='traffic_volume'>
```

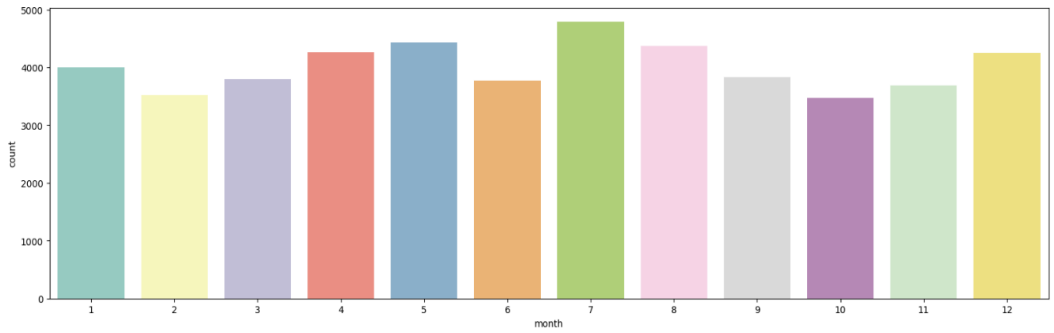

Fig.3.4. Month VS Traffic Flow

3.1.5 Year VS Traffic Flow

Year vs Traffic Flow

```
In [13]: train_df['year'] = train_df['date_time'].dt.year

In [14]: 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);
```

Fig.3.5. Year VS Traffic Flow

3.1.5 Correlation Matrix

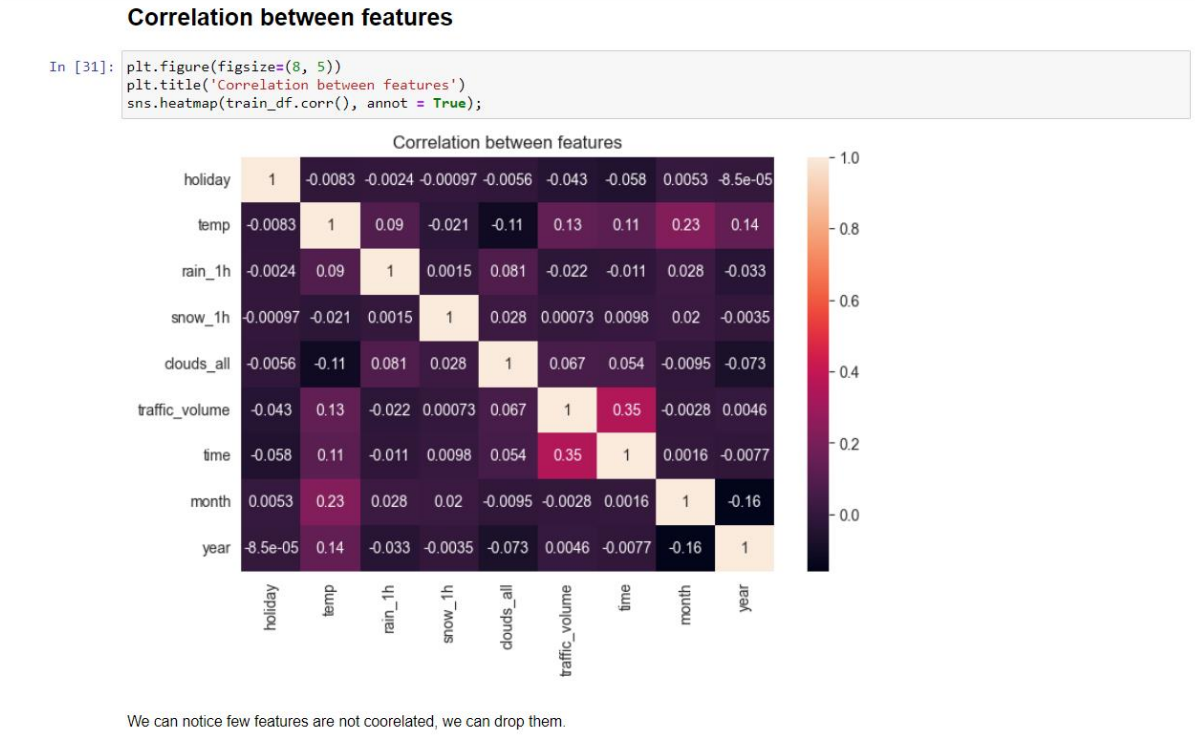

Fig.3.6. Correlation Matrix

3.2. Data Pre-processing

After selecting the dataset, important Python libraries are imported, including Numpy, which is used to add large multidimensional arrays and insert mathematical operations into the code, Pandas, which is used to manipulate and analyze the data, Matplotlib, which is used to plot a 2D representation of the model, and many others. Identifying the missing values and dealing with them, pre-processing is carried out in steps. The dataset must be correctly pre-processed in order to use the machine learning models, and any inconsistent data must be fixed. To do this, the following actions are taken:

- Remove unneeded columns from the dataset that won't help in prediction. These are removed after examining the correlation between features.
- Convert values to numerical format so that the algorithms can utilize them to make predictions, in order to correctly use the machine learning methods.
- The data contains a number of rows with these values. The very big values are replaced by the maximum value that has been established, and the missing values are replaced by the mean values.

4. Results and Observations

Linear Regression Algorithm

```
In [43]: # evaluate the Regressor
         evaluate_model(lireg, 'Linear Regression')

         Linear Regression Train score:  0.14
         Linear RegressionTest score:  0.14
         Root Mean Squared error:  1844.3593672909979
         Coefficient of determination:  0.1376927539352245
```

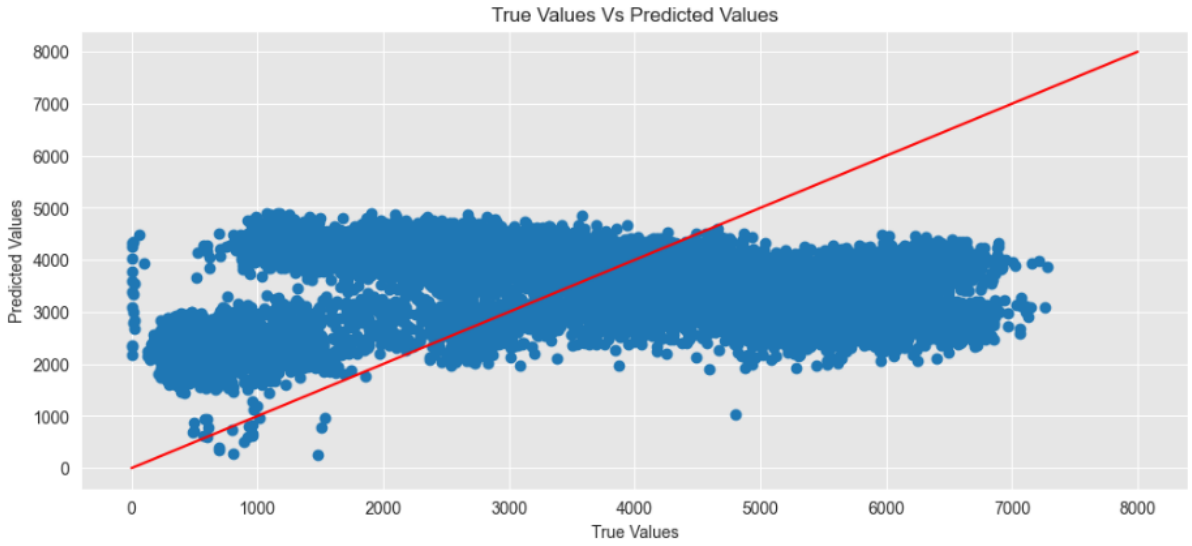

Fig.4.1. Linear Regression Algorithm

Decision Tree Algorithm

```
In [46]: # evaluate the Regressor
         evaluate_model(dtreg, 'Decision Tree')

         Decision Tree Train score:  0.96
         Decision TreeTest score:  0.94
         Root Mean Squared error:  479.64739853701013
         Coefficient of determination:  0.9416803202335742
```

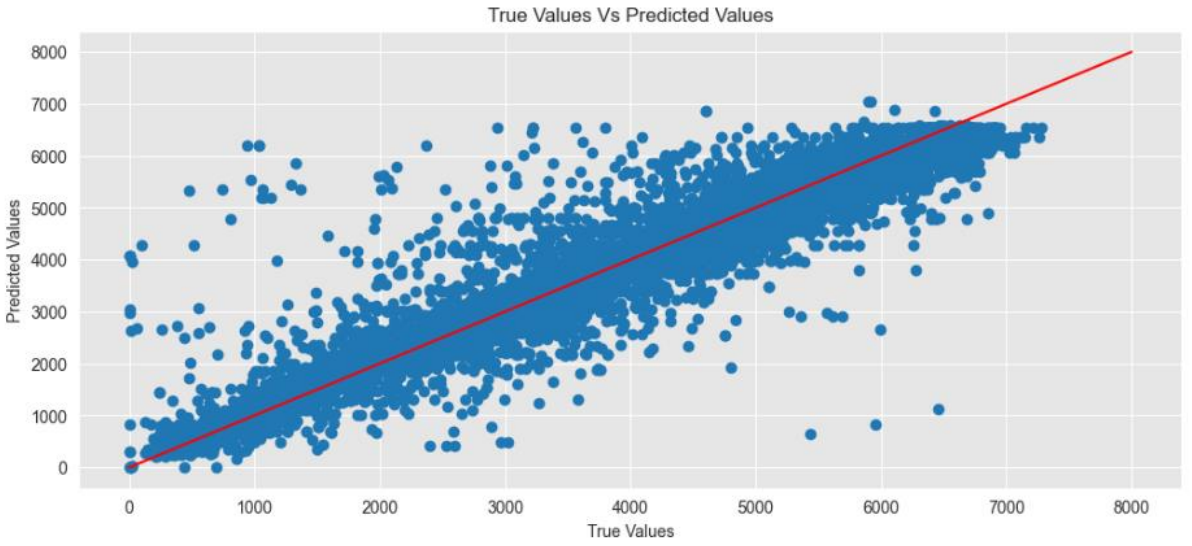

Fig.4.2. Decision Tree Algorithm

Random Forest Algorithm

```
In [49]: # evaluate the Regressor
         evaluate_model(rfreg, 'Random Forest')
```

```
Random Forest Train score:  0.97
Random ForestTest score:  0.95
Root Mean Squared error:  429.3829923369134
Coefficient of determination:  0.9532630227735078
```

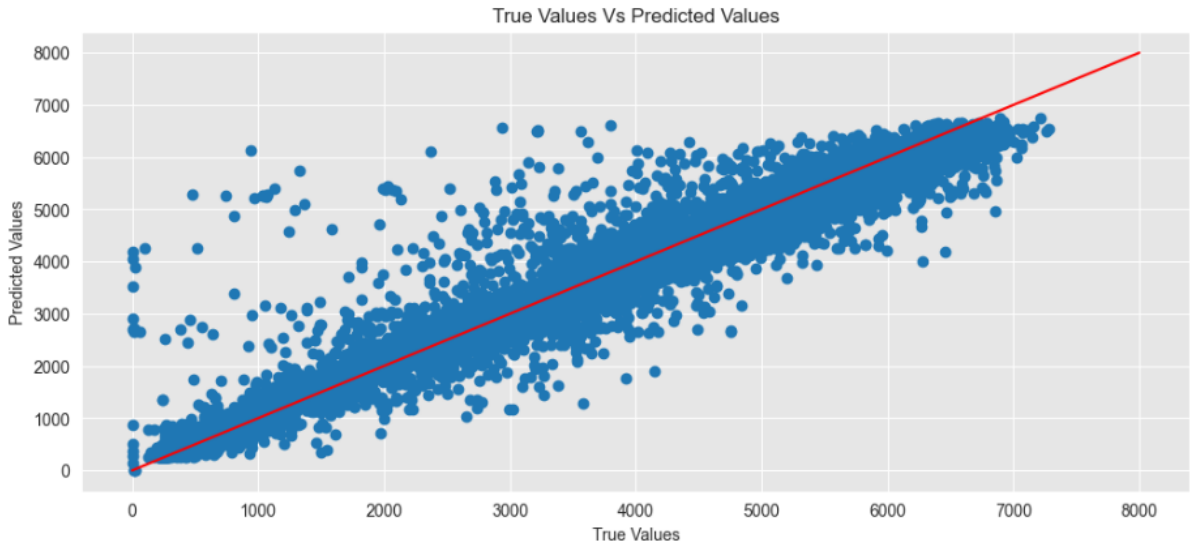

Fig.4.3. Random Forest Algorithm

Gradient Boosting Algorithm

```
In [52]: # evaluate the Regressor
         evaluate_model(gbreg, 'Gradient Boosting')
```

```
Gradient Boosting Train score:  1.0
Gradient BoostingTest score:  0.97
Root Mean Squared error:  362.4862455344718
Coefficient of determination:  0.9666915818613415
```

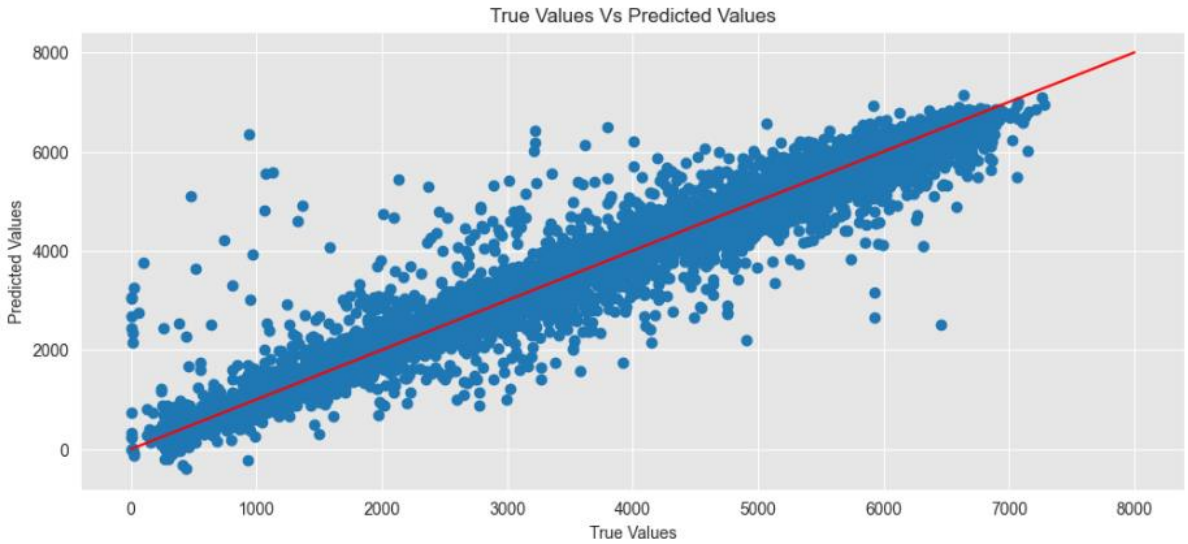

Fig.4.4. Gradient Boosting Algorithm

AdaBoost Algorithm

```
In [55]: # evaluate the Regressor
         evaluate_model(adareg, 'Ada Boost Tree')

Ada Boost Tree Train score:  0.96
Ada Boost TreeTest score:  0.95
Root Mean Squared error:  435.9998057684958
Coefficient of determination:  0.9518114858373186
```

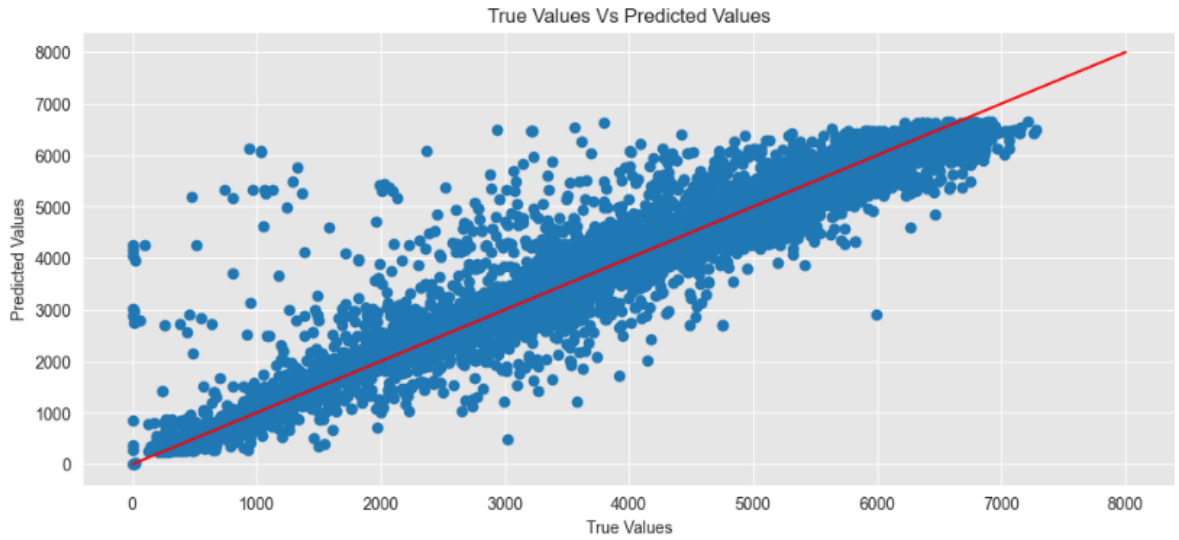

Fig.4.5. AdaBoost Algorithm

Comparison of Algorithms

Comparision between all the above algorithms

```
In [56]: results
Out[56]:
```

	RMSE
Linear Regression	1844.359367
Decision Tree	479.647399
Random Forest	429.382992
Gradient Boosting	362.486246
Ada Boost Tree	435.999806

Thus we can see that **Gradient Boosting** provide the least RSME, therefore we can use it to analyse the traffic flow effectively.

Fig.4.6. Comparison of Algorithms

The several ML models were put into practice, and the outcomes were compared to determine which model performed the best. The root mean square error for linear regression was 1844.35.

The decision tree regressor outperformed the linear regression model with an accuracy rating of 94% and RMSE value of 479.64.

The Random Forest produced a 95 percent accuracy rating with an RMSE value of 429.38, which is still an improvement over the previous model.

Additionally, the AdaBoost method was used, with 60 estimators and a learning rate of 0.005, to create a model with a 95 percent accuracy rate and a 435.99 root mean square error.

We have used the gradient boosting approach by setting the estimators count to 600, which results in an accuracy rating of 97 percent and root mean square error value of 362.48.

The gradient boost approach, thus, provides the better accuracy value and the lowest root mean square error value when all the models are compared. As a result, it can also be used to forecast traffic flow for hypothetical future use cases.

5. Conclusions and future scope

So, in order to estimate the traffic flow, we have deployed various machine learning algorithms. Depending on the accuracy rating and RMSE, we contrasted the various algorithms. Our research showed that the gradient boosting technique had the highest accuracy, at 97 percent, and the lowest root mean square error, at 362.48. In order to determine the busiest seasons and months, we also examined the traffic data.

We therefore conclude that the gradient boosting approach outperformed the other algorithms for our purposes after conducting the necessary research and calculating the accuracy of various machine learning algorithms.

The primary goal, the analyzing and forecasting of traffic using the provided data, was accomplished. By using the sensors and traffic monitoring system to account for more traffic circumstances, we can gain more knowledge for future work. If real-time traffic information can be obtained, a more accurate model can be used to forecast traffic congestion.

6. 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] 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.”

7. Repository Links

Shiva Sankar Modala Repository: https://github.com/SFA22SCM28M/ML_PROJECT_FALL22

Keerthana Reddy Mucherla Repository: https://github.com/KFA22SCM54M/ML_PROJECT_FALL22

Narthu Sailavanya Repository: https://github.com/SaiLavanya1/ML_PROJECT_FALL22