Saptarshi Ghosal

saptarshi.sg@gmail.com

Abstract

The objective of this problem is to clean and analyze the US Traffic 2015 dataset, publicly available in Kaggle and to present with the top 5 most obvious patterns. The entire repository of the project files is available here:

<https://github.com/ghosal-sg/US_Traffic_2015>

OCBC Hack it

Data and AI Track

Contents

[1. Data Exploration: 3](#_Toc82388939)

[1.1 Overview of Data Provided: 3](#_Toc82388940)

[1.2 Checking Existence of Duplicate Values: 3](#_Toc82388941)

[1.3 Checking Level at which Data is Unique: 3](#_Toc82388942)

[1.4 Merging the Two Datasets: 4](#_Toc82388943)

[1.5 Relationship between ‘previous\_station\_Id’ and ‘station\_id’: 4](#_Toc82388944)

[2. Univariate Analysis of Variables: 5](#_Toc82388945)

[2.1 Methodology of Univariate Analysis: 5](#_Toc82388946)

[2.2 Key Findings of Univariate Report: 5](#_Toc82388947)

[2.3 Derived Variables Creation: 5](#_Toc82388948)

[2.4 Missing Value Treatment: 7](#_Toc82388949)

[2.5 Categorical Variable Treatment: 7](#_Toc82388950)

[2.6 Removal of Perfectly Multicollinear Variables: 7](#_Toc82388951)

[2.7 Target Variable Treatment: 8](#_Toc82388952)

[3. Data Visualisation: 9](#_Toc82388953)

[3.1 Methodology of Bi-Variate Analysis: 9](#_Toc82388954)

[3.2 Observed Patterns: 10](#_Toc82388955)

[3.2.1 Relationship of Average Temperature and Traffic Volumes: 10](#_Toc82388956)

[3.2.2 Relationship of a particular Month with respect to Traffic Volumes: 11](#_Toc82388957)

[3.2.3 Relationship of Traffic Volume with respect to Weekend Flag: 12](#_Toc82388958)

[3.2.4 Effect of Traffic Volume with respect to Precipitation: 13](#_Toc82388959)

[3.2.5 Traffic Volumes with respect to Function Classification of Roads: 14](#_Toc82388960)

[4. Predictive Model Development and Validation: 14](#_Toc82388961)

[4.1 Choice of Model Training Algorithm: 14](#_Toc82388962)

[4.2 Model Formulation: 15](#_Toc82388963)

[4.2.1 Training and Validation Methodology: 15](#_Toc82388964)

[4.2.2 Hyperparameter Tuning: 15](#_Toc82388965)

[4.2.3 Best Model Selection: 16](#_Toc82388966)

[5. Appendix: 17](#_Toc82388967)

[5.1 Section A: Cross Frequency Table of ‘previous\_station\_id’ and ‘station\_id’: 17](#_Toc82388968)

[5.2 Section B: Univariate Report: 17](#_Toc82388969)

[5.3 Section C: String Indexing Logic: 17](#_Toc82388970)

[5.4 Section D: Bi-Variate Plots: 17](#_Toc82388971)

[5.5 Section E: XGBoost: Hyperparameter Tuning Results: 18](#_Toc82388972)

[5.6 Section F: XGBoost: Variable Importance: 18](#_Toc82388973)

[Figure 1: Example of Dual Traffic Volume Entry 4](#_Toc82388974)

[Figure 2: Missing Values in Weather Data 6](#_Toc82388975)

[Figure 3: Example of Identical Variables 7](#_Toc82388976)

[Figure 4: Identification of Identical Variables 8](#_Toc82388977)

[Figure 5: K-Means Clustering Output 9](#_Toc82388978)

[Figure 6: Scatter Plot of Average Temperature and Traffic Volumes 10](#_Toc82388979)

[Figure 7: Average Traffic Volume for different Months of Data 11](#_Toc82388980)

[Figure 8: Average Traffic Volume during Vacation Period 11](#_Toc82388981)

[Figure 9: Relationship of Traffic Volumes with Weekend Flag 12](#_Toc82388982)

[Figure 10: Relationship of Traffic Volume with Public Holiday Flag 12](#_Toc82388983)

[Figure 11: Relationship of Traffic Volume with Long Weekend Flag 13](#_Toc82388984)

[Figure 12: Scatter Plot of Traffic Volume and Precipitation 13](#_Toc82388985)

[Figure 13: Traffic Volume with respect to different Functional Classification of Roads 14](#_Toc82388986)

[Figure 14: Hyperparameter Tuning Results 16](#_Toc82388987)

[Table 1: Shape of as received datasets 3](#_Toc82388988)

[Table 2: Distribution of Duplicate Entries with respect to Partitioning Variables in the dot\_traffic\_2015 dataset 3](#_Toc82388989)

[Table 3: Cross Frequency of 'station\_id' and 'previous\_station\_id' 4](#_Toc82388990)

[Table 4: Data Dictionary of Weather Data 6](#_Toc82388991)

[Table 5: Final Model Performance 16](#_Toc82388992)

[Table 6: Top 20 Most Significant Variables 17](#_Toc82388993)

# 1. Data Exploration:

## 1.1 Overview of Data Provided:

There were 2 datasets made available in Kaggle as part of the US Traffic 2015[[1]](#footnote-1) repository. One of the datasets (dot\_traffic\_2015) contained daily traffic volumes recorded hourly in 24 separate bins with respect to station id, geographical location, direction of travel and type of road, while the other (dot\_traffic\_stations\_2015) contained deeper location and historical data on individual observation stations with respect to different locations, direction of travel and type of road. The former as received dataset had 7140391 rows and 38 columns while the latter had 28466 rows and 55 columns.

## 1.2 Checking Existence of Duplicate Values:

The basics of solving any data science problem starts with data inspection and cleaning. As part of data cleaning first it was checked whether the as received datasets contain any duplicate entries or not. The dot\_traffic\_2015 dataset had 6396747 unique rows compared to the total of 7140391 rows as mentioned above. The duplicate entries were all dropped before proceeding forward with the analysis. However, there were no duplicate entry issue in the dot\_traffic\_stations\_2015 dataset.

Table : Shape of as received datasets

|  |  |  |  |
| --- | --- | --- | --- |
| **Dataset Name** | **Rows** | **Unique Rows** | **Columns** |
| **dot\_traffic\_2015** | 7140391 | 6396747 | 38 |
| **dot\_traffic\_stations\_2015** | 28466 | 28466 | 55 |

## 1.3 Checking Level at which Data is Unique:

Since there’s very limited prior information, so it was unknown that which combination of variables make each observation unique in both the datasets. From the very limited input provided in Kaggle it can be assumed that dot\_traffic\_2015 is unique with respect to ‘date’, ‘direction\_of\_travel’, ‘fips\_state\_code’, ‘functional\_classification’, ‘lane\_of\_travel’ and ‘station\_id’. To validate that, the total number of distinct rows with respect to the above variable combination was computed and found to be 5784441, compared to the total number of 6396747 distinct rows. Further analysis was performed to identify the missed-out variables, which on addition can make every observation unique. The distribution of the number of rows with respect to every unique combination of the above variables was created. Out of 5784441 unique combinations of the variables, there were 5172135 unique observations in the dataset, while there were 612306 cases having 2 entries each.

Table : Distribution of Duplicate Entries with respect to Partitioning Variables in the dot\_traffic\_2015 dataset

|  |  |  |
| --- | --- | --- |
| **dot\_traffic\_2015 dataset** | | |
| **Number of Unique Rows** | **Number of cases with Single Entries for Unique combinations of partitioning variables** | **Number of cases with Dual Entries for Unique combinations of partitioning variables** |
| 6396747 | 5172135 | 612306 |

A manual review was performed on few cases with dual entries, and it was observed that, for the given combination of partitioning variables, all other variables are having identical values in both the entries except for the traffic volume hourly bins. However, the recorded traffic volume in the two entries were marginally different, and an example has been presented in Figure 1.

Figure : Example of Dual Traffic Volume Entry

![Graphical user interface, application

Description automatically generated]()

Under such circumstances where there is no prior information on the data recording process, randomly one out of the two entries were deleted from each unique combination of the partitioning variables. This method of data deduping will provide a very decent approximation of the overall traffic volume records. After the final removal of duplicate entries, we ended up with 5784441 unique rows in the dot\_traffic\_2015 dataset.

A similar exercise was performed on the dot\_traffic\_stations\_2015 dataset, however no discrepancies in the data entries were observed. This dataset is a snapshot table of historical traffic station level information and is unique with respect to ‘direction\_of\_travel’, ‘fips\_state\_code’, ‘functional\_classification’, ‘lane\_of\_travel’ and ‘station\_id’. With respect to these variables there are 28466 unique rows, which is exactly same as the total number of unique rows of this dataset.

## 1.4 Merging the Two Datasets:

The dot\_traffic\_2015 dataset contains daily information, while dot\_traffic\_stations\_2015 is a snapshot table, hence merging these two tables require a many on one merge with respect to ‘direction\_of\_travel’, ‘fips\_state\_code’, ‘functional\_classification’, ‘lane\_of\_travel’ and ‘station\_id’. As expected, the merged dataset will be hereafter called as the US\_Traffic\_2015 dataset, and it had 5874441 observations just like the dot\_traffic\_2015 dataset.

## 1.5 Relationship between ‘previous\_station\_Id’ and ‘station\_id’:

A variable called ‘previous\_station\_id’ was observed in the dot\_traffic\_stations\_2015 dataset. Presence of such a variable raises questions of self-merging opportunities for creation of richer information. A manual review of the cross-frequency table of ‘station\_id’ and ‘previous\_station\_id’ was performed to understand such possibilities. Roughly ~85% of the ‘previous\_station\_id’ variable was either 0 or missing. After manual inspection of a handful of the remaining cases, it can be safely concluded that the ‘previous\_station\_id’ contains the older reference numbers of very old traffic stations. With passing time, due to introduction of newer stations in the vicinity, such renaming of stations became necessary. Thus, possibilities of self-merge were ruled out. An example of few such manually inspected cases which suggests ‘station\_id’ is basically a renamed version of ‘previous\_station\_id’ has been presented in Table 3. The detailed cross frequency table is present in [Appendix A](#_5.1_Section_A:).

Table : Cross Frequency of 'station\_id' and 'previous\_station\_id'

|  |  |  |
| --- | --- | --- |
| **previous\_station\_id** | **station\_id** | **Counts** |
| 000011 | 110102 | 275 |
| 000012 | 120304 | 299 |
| 000014 | 001401 | 159 |
| 000014 | 001402 | 128 |
| 000015 | 001503 | 251 |
| 000016 | 001602 | 263 |
| 000017 | 001701 | 331 |
| 000017 | 001702 | 331 |

# 2. Univariate Analysis of Variables:

## 2.1 Methodology of Univariate Analysis:

Many variables which are numeric in nature stay encoded as categorical datatype in any software which in this case is a pandas dataframe in python. So as a first step, all variables of the dataset with categorical datatype were algorithmically attempted for a conversion to numeric. With variables with alphanumeric entry, such attempts failed, and the original data type was retained.

Fill Rate of all the variables of both categorical and numeric data types were checked as a second step. Finally for numeric variables the mean, standard deviation, 25th, 50th, and 75th percentiles were computed, and on the contrary for categorical variables the one-way frequency table of all the distinct levels were computed. The detailed univariate report is present in [Appendix B](#_5.2_Section_B:).

## 2.2 Key Findings of Univariate Report:

The variable 'restrictions' was found to be 100% missing, while the variables 'record\_type' and 'year\_of\_data' has zero standard deviation (constant value). These 3 variables were dropped before proceeding forward with subsequent analysis. There were other variables with very low fill rate, but they were retained to evaluate potential patterns which might be present due to the presence or absence of data in such variables.

## 2.3 Derived Variables Creation:

Feature Engineering or Derived variable creation is a key part in any statistical analysis. Derived variables help in validating business hypothesis with real life data, and in a predictive modelling setup, derived variables are the key in having better model performance.

After looking into the data, the initial hypotheses were that there will be different traffic volumes during weekends, public holidays, festive seasons, or long weekends. So, 4 new features were created as explained below:

* **Weekend\_Flag:** All Saturdays and Sundays were marked as 1, and the rest of the days were marked as 0.
* **Public\_Holiday\_Flag:** Using the python library ‘holidays’[[2]](#footnote-2), all the public holidays were identified in the year of 2015, and all those days were marked as 1, and the rest as 0.
* **Long\_Weekend\_Flag:** Sometimes we end up having public holidays on a Friday or on a Monday. Under such circumstances people make special plans, to go out of town for a long drive or to travel within the town for specific purposes. All weekends preceded or succeeded by a public holiday were marked as 1 along with the public holiday and the rest as 0.
* **Vacation\_Flag:** This was a subjective call. Usually, people take a big break in US in the last 2 weeks of December and the first week of January to celebrate Christmas and New Year. After observing the 2015 calendar, 1st to 4th of January and 19th to 31st of December were considered as vacation days and marked as 1, and the rest of the days were marked as 0.

The above variables will be used to check whether there is any impact on traffic volumes due to presence or absence of such events.

Weather has a key role in traffic volumes. People will seldom drive during high precipitation (snowfall or rainfall), or during extreme temperatures (extreme heat, or extreme cold). In the as received datasets there was no weather information available. However, latitude and longitude information were present. So, the python library ‘meteostat’[[3]](#footnote-3) was used to fetch daily weather information for all the distinct latitude and longitude combinations. This library collects data at different weather stations. The nearest weather station for a given latitude/ longitude combination was used.

After fetching the weather data using ‘meteostat’, a missing value rate of ~28% was observed in average daily temperature (‘tavg’), while a very high missing rate of (~81%) was observed in the precipitation variable (‘prcp’). The details of the meteostat variables with description has been listed below in Table 4 along with the % missing values in Figure 2.

Table : Data Dictionary of Weather Data[[4]](#footnote-4)

|  |  |  |
| --- | --- | --- |
| **Column** | **Description** | **Type** |
| **station** | The Meteostat ID of the weather station (only if query refers to multiple stations) | String |
| **time** | The date | Datetime64 |
| **tavg** | The average air temperature in *°C* | Float64 |
| **tmin** | The minimum air temperature in *°C* | Float64 |
| **tmax** | The maximum air temperature in *°C* | Float64 |
| **prcp** | The daily precipitation total in *mm* | Float64 |
| **snow** | The snow depth in *mm* | Float64 |
| **wdir** | The average wind direction in degrees (*°*) | Float64 |
| **wspd** | The average wind speed in *km/h* | Float64 |
| **wpgt** | The peak wind gust in *km/h* | Float64 |
| **pres** | The average sea-level air pressure in *hPa* | Float64 |
| **tsun** | The daily sunshine total in minutes (*m*) | Float64 |

Figure : Missing Values in Weather Data

![Table

Description automatically generated]()

Average Temperature is a better measure of weather in a day compared to maximum and minimum temperatures, and hence based on fill rates only the average temperature variable was retained (‘tavg’) as an indicator of temperature. The precipitation variable although had a low fill rate, yet it can be a very key driver in daily traffic volume. Hence the ‘prcp’ variable was retained for subsequent analysis.

## 2.4 Missing Value Treatment:

There are several sophisticated missing value treatment methods for different kind of missing values like MAR (Missing at Random), MCAR (Missing completely at Random) and NMAR (Not Missing at Random). However, for a dataset where the data collection process is unknown, and proper data dictionary is not available, it’s better to avoid complicated imputation techniques. Rather a special value of ‘-99999999’ was used to impute all missing values.

## 2.5 Categorical Variable Treatment:

One Hot Encoding is the most popular technique for categorical variable treatment. In this process for every unique level of a categorical variable a dummy variable (1/0) is created based on the presence of absence of that level, corresponding to the variable. One major concern of this technique is that it increases data dimensions significantly. Thus, such a technique has been avoided. String Indexing was used for categorical variable treatment. In String Indexing, the level having highest frequency is assigned an index of 0, while the level having second highest frequency is assigned a level of 1, so on and so forth. This technique effectively presents a categorical variable in a numeric form, without increasing data dimensions. A detailed mapping file of the string indexing logic is present in [Appendix C](#_5.3_Section_C:). After the indexing process all the original categorical variables were dropped, and indexed variables were named with the original name of the categorical variable with an additional suffix of ‘\_index’.

## 2.6 Removal of Perfectly Multicollinear Variables:

As part of the dataset some identical variables with different names were provided. For example, the variable ‘function\_classification’ and ‘functional\_classification\_name’ are identical, where the former contains the abbreviation of the latter, - an example of which has been presented in Figure 3.

Figure : Example of Identical Variables

![Table

Description automatically generated]()

The Univariate Report present in [Appendix B](#_5.2_Section_B:), can also be referred to evaluate these cases, where the One-Way frequency tables of such variables are identical.

It’s really a cumbersome task to identify such cases manually and drop one variable from the identical pair. Hence a correlation matrix of all possible variable combinations was computed. The variable combinations apart from identical variables which are having a perfect correlation coefficient of 1 were filtered out. One variable from each such pair was dropped. The snapshot of the correlation matrix has been presented in Figure 4.

Figure : Identification of Identical Variables

![Table

Description automatically generated]()

Based on the above table, the following variables were dropped henceforth:

* ‘method\_of\_data\_retrieval’,
* ‘functional\_classification\_name\_index’
* ‘type\_of\_sensor\_name\_index’

## 2.7 Target Variable Treatment:

Since the primary objective is to identify patterns in Traffic Volumes, hence the recorded traffic volume variables (example: traffic\_volume\_counted\_after\_0100\_to\_0200), will be henceforth called as Target Variables. There are 24 target variables of 24 respective hourly bins, recording the traffic volumes of a particular combination of ‘station\_id’, locational information (geographical place), traffic flow direction and type of road over a period of 1 year. Typically, traffic volumes vary throughout the day depending on multiple factors of business hours, non-business hours, night-time, etc. However, there's opportunity to club some of the hourly bins into a single variable based on the respective average traffic volumes. For example, traffic volumes can be similar between 10am to 11am and 11am to 12 noon, hence these 2 variables can be clubbed into a single variable representing traffic volume between 10am to 12 noon.

A K-Means clustering of the means of these 24 variables were performed to create 3 such groups, which will be henceforth named as Traffic Volume during Business Hours (High Traffic Volume), Traffic Volume during non-business hours (Medium Traffic Volume) and traffic volume during night-time (Low Traffic Volume). Based on the k-means clustering output, we observed that there's heavy traffic volume between 7 am to 7 pm (Business Hours), medium traffic volume in between 6am to 7am in the morning and between 7pm to 10pm at night (Non-Business Hours) and low traffic volume between 10pm to 6am (Night-Time). As per the above definitions the 24 Target Variables were converted to 3 variables named as:

* Traffic\_Volume\_Non\_Business\_Hours (7am to 7pm)
* Traffic\_Volume\_Night\_Time (6am to 7am or 7pm to 10pm)
* Traffic\_Volume\_Business\_Hours (10pm to 6am)

The K-means clustering output has been presented below in Figure 5.

Figure : K-Means Clustering Output

![Text

Description automatically generated with medium confidence]()

# 3. Data Visualisation:

All other variables apart from the 3 Target Variables will be hereafter called as the Independent Variables, and the objective of this analysis is to evaluate Bi-Variate relationship between the Target Variables and the Independent Variables.

## 3.1 Methodology of Bi-Variate Analysis:

The Target variables are traffic volume information and it's continuous in nature. Since all the categorical variables are already indexed, so all the independent variables which we are going to deal with will be numeric. However, to make the visualizations intuitive, all independent variables which are indexed (originally categorical) or variables having less than 50 unique levels or variables which portray nominal behaviour, will be used to create variable average bar plots.

**Variable Average Bar Plot:** For every unique level of each independent variable the average of the target variable will be computed, and bars will be stacked for each unique level.

For the rest of the independent variables other than above (typically with more than 50 unique levels), scatter plots will be created with respect to the target variable.

After segregating the independent variables into two groups, the following 4 variables were chosen for scatter plot creation:

* 'prcp'
* 'lrs\_location\_point'
* 'year\_station\_established'
* 'tavg'

The rest of the independent variables were all used to create variable average bar plots with the target.

## 3.2 Observed Patterns:

All the Bi-Variate plots are present in [Appendix D](#_5.4_Section_D:), however few of them has been discussed below in detail. The subsequent validation of the following hypotheses have been done using the predictive model in [Section 4.2.3](#_4.2.3_Best_Model).

### 3.2.1 Relationship of Average Temperature and Traffic Volumes:

Typically, we can assume that people will avoid driving in extreme temperatures be it extreme cold or extreme hot. The scatter plot represented in Figure 6 tells us a similar story.

Figure : Scatter Plot of Average Temperature and Traffic Volumes

Chart, scatter chart

Description automatically generated

We observed very low traffic volumes in extreme low temperatures (< -200C) or extreme heat (>350C). Interestingly this pattern holds across traffic volumes in business hours, non-business hours or night-time.

### 3.2.2 Relationship of a particular Month with respect to Traffic Volumes:

In US, we can assume that there will be heavier traffic during business hours across the year except for the festive period where people take extended vacations and move back to their hometown. So, it’s a fair expectation to have significantly reduced traffic in December, and slightly lower traffic in January when compared with the other months. Night-time traffic on the other hand is expected to have similar patterns across all months, as during night mostly heavy vehicles travel inter-state for supply of goods. In Figure 7 we can see something similar as explained above.

Figure : Average Traffic Volume for different Months of Data

Chart, bar chart

Description automatically generated

Figure : Average Traffic Volume during Vacation Period

Chart, bar chart

Description automatically generated

The above hypothesis can be further supported by the average traffic volume with respect to the Vacation Flag as shown in Figure 8, where 1st to 4th January or 19th to 31st December has been flagged as vacation period.

### 3.2.3 Relationship of Traffic Volume with respect to Weekend Flag:

We can expect to have lower traffic volumes during the business hours of weekend compared to weekdays, as shown in Figure 9. A similar analysis was performed with Public Holiday Flag and Long Weekend Flag respectively presented in Figure 10 and Figure 11, where we see similar results.

Figure : Relationship of Traffic Volumes with Weekend Flag

Chart, bar chart

Description automatically generated

Figure : Relationship of Traffic Volume with Public Holiday Flag

Chart, bar chart

Description automatically generated

Figure : Relationship of Traffic Volume with Long Weekend Flag

Chart, bar chart

Description automatically generated

### 3.2.4 Effect of Traffic Volume with respect to Precipitation:

This is the most obvious hypothesis. Anyone who has been into driving, will avoid the activity during precipitation be it rainfall or snowfall for visibility and safety purposes. Interestingly the precipitation variable had very poor fill rate ~20%, however since we had a sufficiently large sample size, even with low fill rate the hypothesis can be validated just from simple observation of the graph. The scatter plot of precipitation and traffic volumes has been presented in Figure 12.

Figure : Scatter Plot of Traffic Volume and Precipitation

Chart, histogram

Description automatically generated

### 3.2.5 Traffic Volumes with respect to Function Classification of Roads:

We can expect that Urban or heavily developed areas with loads of freeways and expressways will have heavier traffic volumes compared to other areas. We observed as represented in Figure 13, that the index number 0 (1U: Urban: Principal Arterial - Interstate) and index number 4 (2U: Urban: Principal Arterial - Other Freeways or Expressways) have the heaviest traffic during all Business/ Non-Business Hours and Night-time. As per the DOT official website the following definitions of the above roads were obtained:

*“Serve the major centres of activity of a metropolitan area, the highest traffic volume corridors; carry a high proportion of the total urban area travel on a minimum mileage. The principal arterial system should carry the major portion of trips entering and leaving the urban area, as well as the majority of through movements desiring to bypass the central city. Almost all fully and partially controlled access facilities will be part of this functional system.” [[5]](#footnote-5)*

Figure : Traffic Volume with respect to different Functional Classification of Roads

Chart, bar chart

Description automatically generated

# 4. Predictive Model Development and Validation:

## 4.1 Choice of Model Training Algorithm:

With the pool of independent variables available we can develop a predictive model to know apriori the volume of traffic in a particular day, in a particular area, in the jurisdiction of a particular station id, direction of travel and type of road. With the predictions available, Traffic Control can make sufficient planning for the ease of passage of vehicles. Since we are predicting a continuous variable (Volume of Traffic), hence we must use a Regressor. There are lots of options among Regressors, namely Linear Regressor, Decision Tree Regressor, Random Forest Regressor, Gradient Boosting Regressor or XGBoost Regressor. From the Bi-Variate analysis as presented in Appendix D, we observed that most of the independent variables are having a non-linear relationship with the target variables. Hence to develop a parametric model like Linear Regression, we must perform lots of transformations to satisfy the assumptions. Under such circumstances usage of Machine Learning is a better idea. Among the Machine Learning algorithms XGBoost[[6]](#footnote-6) is the most popular one across the industry for its options of regularisation (which controls overfit) and parallel processing (faster model trainings). XGBoost algorithm also supports GPU acceleration while training, and hence for the above advantages the XGBoost Regressor was chosen for development of the predictive model. Since we have created 3 target variables namely, Traffic volumes of Non-Business Hours, Night-Time, and Business Hours, we have chosen only one Target Variable for the predictive model development. Out of the 3, predicting traffic volumes during business hours (when there are a greater number of vehicles), will be more useful for practical usage.

## 4.2 Model Formulation:

### 4.2.1 Training and Validation Methodology:

The entire dataset was split into 2 parts: a ~70% random sample for training and the remaining ~30% for In-Time Validation, hereafter will be called as Test. Under circumstances where there is absence of Out of Time dataset for model performance evaluation usage of K-fold cross validation is an industry standard practice. Since we have chosen XGBoost Regressor, we must perform hyperparameter tuning to identify the most optimal model to attain stability and best performance in our Test dataset. For hyperparameter tuning there are multiple options, like Random Search, Manual Grid Search or Bayesian Optimisation to choose from. Bayesian Optimisation was chosen as the preferred method of hyperparameter tuning for its efficiency in moving towards the convergence with lesser number of trials. Hyperparameter tuning is itself an expensive process and using K-Fold cross validation on top of it makes it even more complex, - hence K-Fold cross validation was avoided for this experiment. Such an experiment requires cloud compute and can't be processed in a personal laptop.

### 4.2.2 Hyperparameter Tuning:

A custom objective function was formulated for the Bayesian approach of Hyperparameter tuning. In this objective we want to maximise the performance of the Test dataset (here we have used adjusted R Squared as our evaluation metric). We also want to minimise overfit (Here we have defined overfit as the relative difference of adjusted R Squared between the Train dataset and the Test dataset, if the Train adjusted R squared value is strictly greater than the Test dataset. Else we have defined the overfit value as 0). To achieve this 2-fold optimisation here we have introduced a concept called Ideal Model. We have defined Ideal model as the one having ideal adjusted R Squared which is 1, and ideal overfit, which is 0 (No drop of Test performance from Train). In a 2D Cartesian system where the Test Performance is represented in the X-axes and the Overfit is represented in the Y-axes, the ideal model can be marked with the point (1,0). For any given hyperparameter we can plot the respective Test Performance and Overfit in this same cartesian plane. The best model among all the hyperparameter combinations will be the one having minimum Euclidean Distance from the Ideal Model. The Bayesian Optimisation was performed using the python library ‘hyperopt’[[7]](#footnote-7). In Manual Grid Search we typically try a fixed number of hyperparameter combinations to find the optimal combination. This process is heavily expensive, yet there’s no guarantee of obtaining the best model. However, in Bayesian Optimisation, the algorithm suggests the next best possible hyperparameter combination based on the prior results, and thus we can gradually walk towards the zone of optimality with significantly lesser number of trials. For all practical purposes we have limited the total number of HyperOpt trials to 100, and utilised GPU acceleration to achieve maximum parallelism while training. The following hyperparameters were tuned using the Bayesian Optimisation process:

* max\_depth
* learning\_rate
* gamma
* lambda
* alpha
* subsample
* colsample\_bytree

We have used the early stopping criteria to tune the total number of trees or boosting rounds also known as ‘num\_boost\_round’. If for 10 consecutive iterations there’s no reduction in RMSE (Root Mean Squared Error) of the Test dataset, training was stopped.

### 4.2.3 Best Model Selection:

The entire hyperparameter tuning results based on Bayesian Optimisation has been presented in [Appendix E](#_5.5_Section_E:). Figure 14 shows the scatter plot of the Test dataset performance and Overfit. The Best model is the point which is nearest to the ideal model.

Figure : Hyperparameter Tuning Results

**Ideal Model**

**(1,0)**

The Adjusted R Squared of the Train and Test datasets along with the Normalised Root Mean Squared Errors have been presented below in Table 5:

Table : Final Model Performance

|  |  |  |  |
| --- | --- | --- | --- |
| **Hyperparameter:** | | | |
| **Train Adjusted R Squared** | **Train NRMSE** | **Test Adjusted R Squared** | **Test NRMSE** |
| 0.94 | 0.36 | 0.95 | 0.34 |

The most significant variables (Top 20) of the model are also presented in Table 6, and we see that significant value has been generated by means of the feature engineering. The Weather data along with the Holiday information were utilised multiple times for creating the tree splits. The entire Variable Importance sheet has been presented in Appendix F. The Top 5 patterns shown in [Section 3.2](#_3.2_Observed_Patterns:) are highlighted in Table 6. All the hypotheses related to the most obvious patterns have been validated using the predictive model.

Table : Top 20 Most Significant Variables

|  |  |
| --- | --- |
| **Variable** | **Weight** |
| tavg | 60316 |
| day\_of\_week | 59485 |
| month\_of\_data | 56036 |
| day\_of\_data | 54591 |
| station\_location\_index | 49466 |
| lrs\_location\_point | 38475 |
| posted\_signed\_route\_number\_index | 36913 |
| fips\_county\_code | 36597 |
| lane\_of\_travel | 33319 |
| direction\_of\_travel | 32968 |
| year\_station\_established | 29943 |
| lrs\_identification\_index | 27378 |
| Weekend\_Flag | 25282 |
| direction\_of\_travel\_name\_index | 23718 |
| hpms\_sample\_identifier\_index | 21672 |
| functional\_classification\_index | 18043 |
| prcp | 17243 |
| fips\_state\_code | 17233 |
| number\_of\_lanes\_monitored\_for\_traffic\_volume | 16565 |
| number\_of\_lanes\_monitored\_for\_vehicle\_class | 15117 |

# 5. Appendix:

## 5.1 Section A: Cross Frequency Table of ‘previous\_station\_id’ and ‘station\_id’:



## 5.2 Section B: Univariate Report:



## 5.3 Section C: String Indexing Logic:



## 5.4 Section D: Bi-Variate Plots:

(Please check GitHub Repo: *“./Output/Bi\_Variate\_Plots/”*)

## 5.5 Section E: XGBoost: Hyperparameter Tuning Results:



## 5.6 Section F: XGBoost: Variable Importance:



1. https://www.kaggle.com/jboysen/us-traffic-2015 [↑](#footnote-ref-1)
2. https://pypi.org/project/holidays/ [↑](#footnote-ref-2)
3. https://github.com/meteostat/meteostat-python [↑](#footnote-ref-3)
4. https://dev.meteostat.net/python/daily.html#example [↑](#footnote-ref-4)
5. https://www.dot.ny.gov/divisions/engineering/applications/traffic-data-viewer/tdv-definitions/Functional-Classifications.htm [↑](#footnote-ref-5)
6. https://xgboost.readthedocs.io/en/latest/ [↑](#footnote-ref-6)
7. http://hyperopt.github.io/hyperopt/ [↑](#footnote-ref-7)