**MSBD 5002 - Knowledge Discovery and Data Mining - Spring Semester (2019-2020)**

**Group Project - Highway Tollgates Traffic Flow Prediction (Group 3)**

|  |  |  |
| --- | --- | --- |
| LING Kai Leong  20580290  kllingaa@connect.ust.hk | KWONG Ka Ming  20094899  kmkwongaa@connect.ust.hk | LO Kok Kit  20547224  kkloac@connect.ust.hk |

1. **Abstract**

This is a project from KDD Cup 2017 (one of the most famous data mining competitions). The major goals of the project is to analyze and predict the key bottlenecks in traffic networks - Highway tollgates.

1. **Introduction**

Highway tollgates are well known bottlenecks in traffic networks. During rush hours, long queues at toll gates can overwhelm traffic management authorities. Effective preemptive countermeasures are desired to solve this challenge. Such countermeasures include expediting the toll collection process and streamlining future traffic flow. The expedition of toll collection could be simply allocating temporary toll collectors to open more lanes. Future traffic flow could be streamlined by adaptively tweaking traffic signals at upstream intersections. Preemptive countermeasures will only work when the traffic management authorities receive reliable predictions for future traffic flow. For example, if heavy traffic in the next hour is predicted, then traffic regulators could immediately deploy additional toll collectors and/or divert traffic at upstream intersections.

In this project, we have selected task 2 to work on, which we need to predict average tollgate traffic volume. For travel volume prediction, the initial training set contains data gathered from Sep. 19th to Oct. 17th, where data structures have been shown in table 2. For every 20-minute time window, we try to predict the entry and exit traffic volumes at tollgates 1, 2 and 3.

1. **Dataset**

There are total 573,140 rows of record, with 543,699 (94.9%) are training data and 29,441 are testing data. Each record have the following attribute:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Data type** | **Meaning** |
| time | datatime | the time when a vehicle passes the tollgate |
| tollgate\_id | string | ID of the tollgate (ranges from 1 to 3) |
| direction | string | 0:entry, 1:exit |
| vehicle\_model | int | this number ranges from 0 to 7, which indicates the capacity of the vehicle(bigger the higher) |
| has\_etc | string | does the vehicle use ETC (Electronic Toll Collection) device?  0: No, 1: Yes |
| vehicle\_type | string | vehicle type: 0-passenger vehicle, 1-cargo vehicle |

1. **Objective**

We have 4 objectives throughout the project, including

* To acquire a better understanding of data mining techniques.
* To familiar how to complete a data mining and analytic project as a team
* To familiar the procedures of data engineering and data modeling
* To finish task 2 and estimate average tollgate traffic volume.

1. **Methodology**

When we work with the data mining and predictions, we have defined a three major step to produce the datasets, model and the results we need:

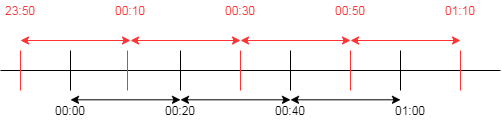
1. We have tried to determine the unit of analysis and identify the target field, such that we need to estimate the average travel time.
2. Next we need to work on data engineering to generate more data and find the relationships in between.
3. After that we use the dataset to build our model , and modify models with the use of accuracy on historical data

1. **Data Engineering**

**Data aggregation by a moving window**

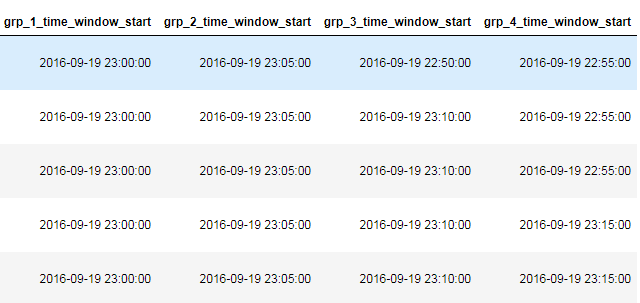
As required from the competition, we need to predict the volume for each tollgate and each direction in every 20 minute interval. However, in the provided data, the volume is provided in 1 minute interval. Therefore we need to aggregate the provided data to prepare the training data set. Here we use a moving window strategy to achieve this, with the benefit of having more training data to work with.

We tried to add some additional time windows in groping the traffic volume to allow more data for training. According to the requirement, we need to predict the entry and exit traffic volumes for every 20-minute time window.



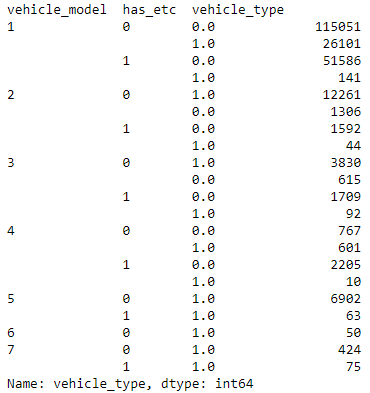
Instead of just aggregating the volume for minutes 0-19, 20-39 and 40-59, we are also aggregate them for minutes 05-24, 25-44 and 44-05(next hour), if the window is shifted by 5 minutes.

Hence, we have created 3 additional time windows (shifted by 5, 10 and 15 minutes from the original):



**Data Imputation - Missing vehicle type**

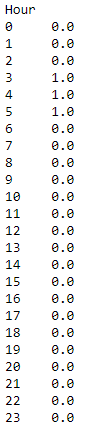
We observed that there are some missing values for the attribute “vehicle\_type”. For all “direction” = 0 (indicating an entry record), the vehicle\_type is missing. By observing the vehicle\_type across with other attribute, we have the following pattern:



By observing the pattern mentioned above, we decided to modify the “vehicle\_type” in the exit data for better training. We start with grouping the “vehicle\_model” and “has\_etc” as our independent variable, and vehicle\_type is our target variable.

We simply took the mode as the class label despite the condition for vehicle\_model = 1 and has\_etc = 0. It is because this class has a large sample size when compared to the other class.

For vehicle\_model = 1 and has\_etc = 0, we will further drill into the hour. We also take mode as the class label. Below is the rule specific for vehicle\_model = 1 and has\_etc = 0:



By combining these two sets of rules, we verify the prediction with our original data. The result shows that the accuracy is around **87.5%** which we believe is acceptable to proceed to the next step.

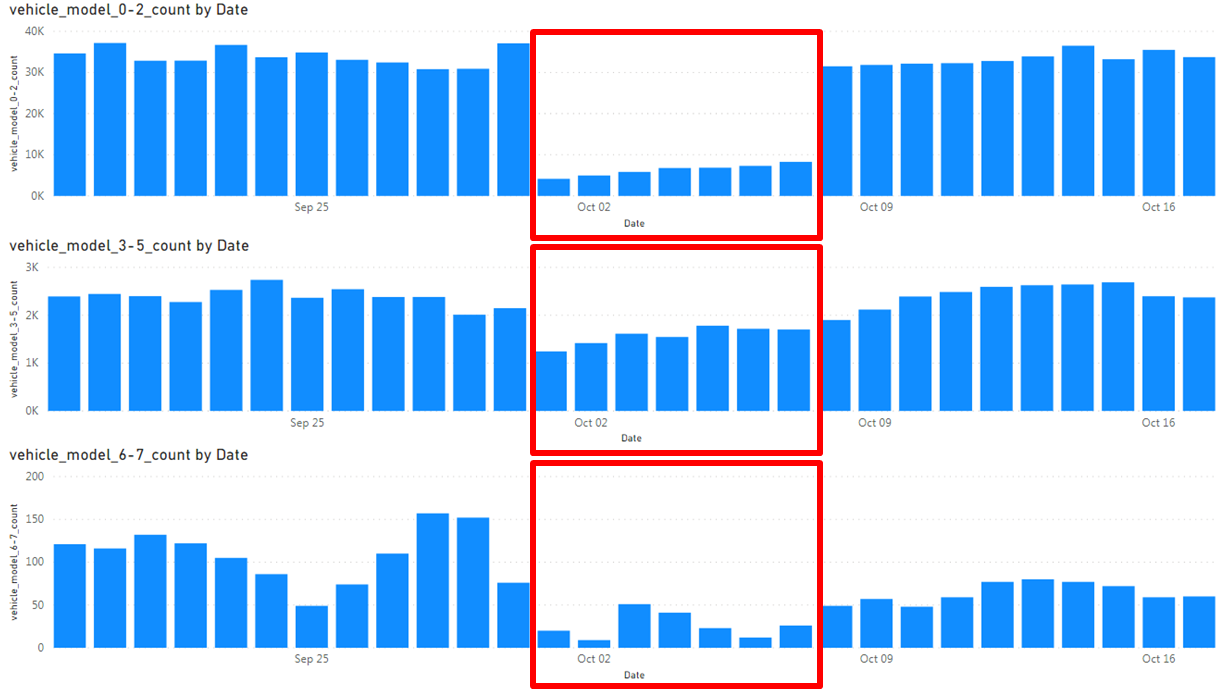
**Data Cleansing - dropping the Golden Week**

The data from 01-07 Oct is dropped because it is very different from the others. This is due to the Golden Week in China. The prediction period, 18-24 Oct, is normal so we believe that dropping these days can help the model to learn better.

For entry tollgate:



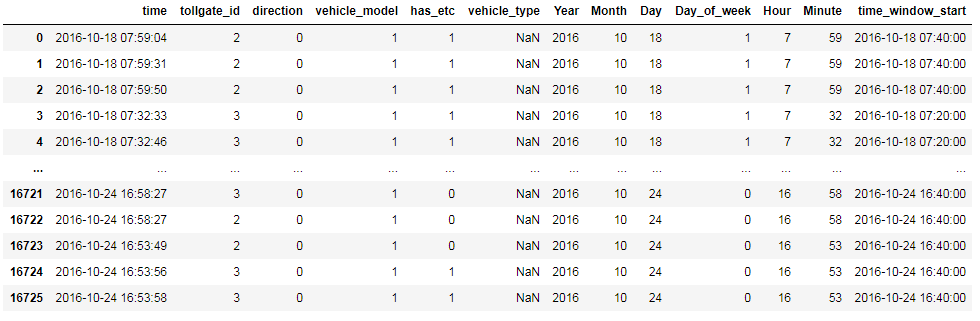
For exit tollgate:



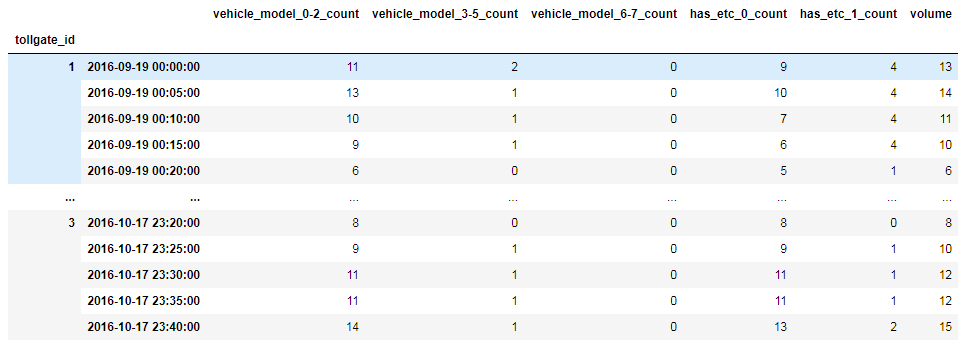
**Feature Engineering**

1. **Time based features**

Time-based feature columns are added to the original data. The time attribute is split into “Year”, “Month”, “Day”, “Day\_of\_week” (Monday as 0 , and Sunday as 6), “Hour”, “Minute” and added into the original dataframe.



1. **Derived features**

****

* 1. Grouping vehicle model into 3:

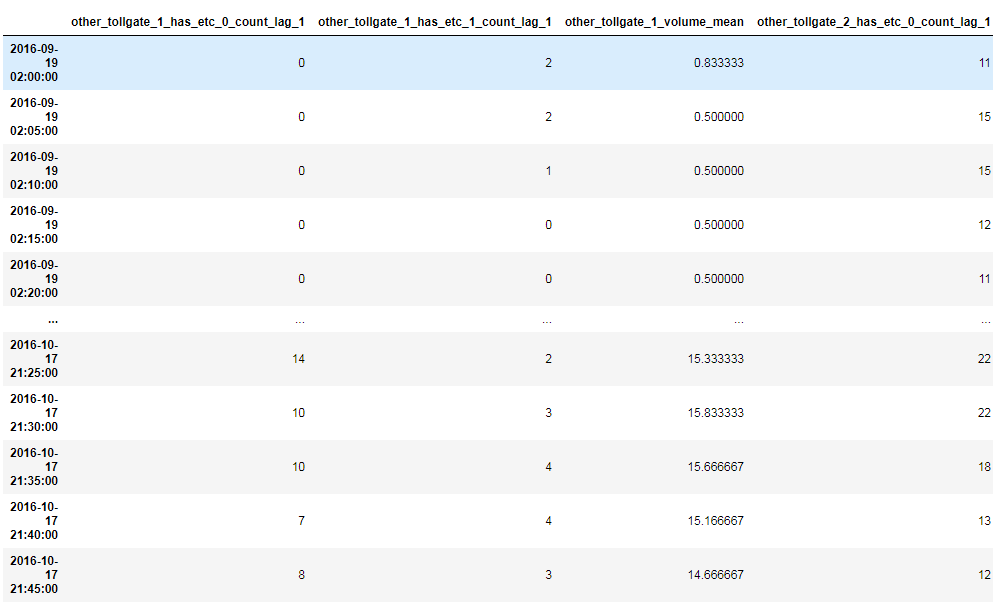
Since the “vehicle model” indicates the capacity of the vehicle, we then group the vehicle into three groups:

* 1st group: vehicle model = 1 and vehicle model = 2;
* 2nd group: vehicle model = 3, vehicle model = 4 and vehicle model = 5;
* 3rd group: vehicle model = 6, vehicle model = 7
  1. Selective counting:

Instead of just summing up the total volume of vehicles passed, we also count the volume of vehicles passed which satisfies different criteria, such as having “has\_etc” as 0/1 etc.

* 1. Information from other tollgates:

For each tollgate, we also add some features which capture the volume of vehicle passed from the other tollgates at the 2 latest time intervals. These are added in order to capture the correlation between different tollgates.

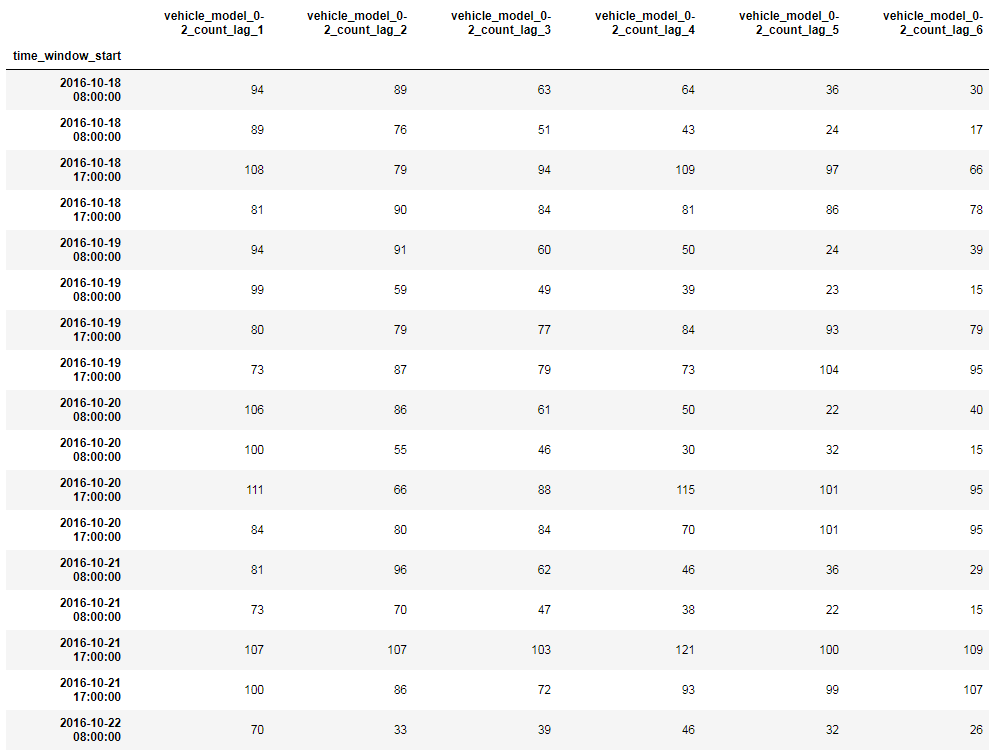


1. **Lagging features**

The lagging values are important for time series prediction. In our case, the lagging values are not just extracted from target columns, but also other feature columns.

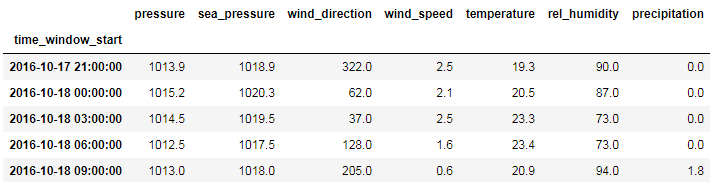
6 “lag” are generated into the data. We are expected to predict 2 hours ahead, which is 6 20-minute intervals, so having the previous 2 hours as the features is more natural in our belief. In short, we use the past 2 hours to predict the next 2 hours, applied with a 20 minutes interval.

Moreover, we also added some statistical measures (“mean” in this case). Therefore, not just the 6 lagging values, but also the derived statistical measures for these lags are added to the data set.

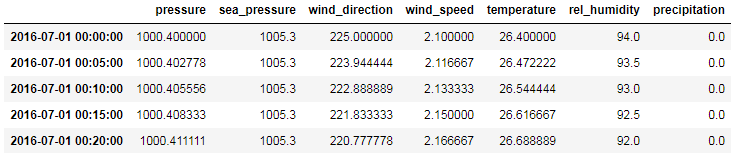


1. **Weather features**

Last but not least, we also added weather data into the dataset. Attributes included wind direction, wind speed, temperature, relative humidity and precipitation.



Since the weather data is in 3 hours interval, which is different from our training data set, linear interpolation is used before joining the weather data to our training and testing sets.



**Feature list**

The following are the features we used for the entry tollgate:

['tollgate\_id', 'vehicle\_model\_0-2\_count\_lag\_1',

'vehicle\_model\_3-5\_count\_lag\_1', 'vehicle\_model\_6-7\_count\_lag\_1',

'has\_etc\_0\_count\_lag\_1', 'has\_etc\_1\_count\_lag\_1', 'volume\_lag\_1',

'vehicle\_model\_0-2\_count\_lag\_2', 'vehicle\_model\_3-5\_count\_lag\_2',

'vehicle\_model\_6-7\_count\_lag\_2', 'has\_etc\_0\_count\_lag\_2',

'has\_etc\_1\_count\_lag\_2', 'volume\_lag\_2', 'vehicle\_model\_0-2\_count\_mean',

'vehicle\_model\_3-5\_count\_mean', 'vehicle\_model\_6-7\_count\_mean',

'has\_etc\_0\_count\_mean', 'has\_etc\_1\_count\_mean', 'volume\_mean',

'pressure', 'sea\_pressure', 'wind\_direction', 'wind\_speed',

'temperature', 'rel\_humidity', 'precipitation',

'other\_tollgate\_1\_has\_etc\_0\_count\_lag\_1',

'other\_tollgate\_1\_has\_etc\_1\_count\_lag\_1',

'other\_tollgate\_1\_volume\_mean',

'other\_tollgate\_2\_has\_etc\_0\_count\_lag\_1',

'other\_tollgate\_2\_has\_etc\_1\_count\_lag\_1',

'other\_tollgate\_2\_volume\_mean', 'Month', 'Day', 'Day\_of\_week',

'Is\_weekend', 'Hour', 'Minute', 'target\_1', 'target\_2', 'target\_3',

'target\_4', 'target\_5', 'target\_6']

The following are the features we used for the exit tollgate:

['tollgate\_id', 'vehicle\_model\_0-2\_count\_lag\_1',

'vehicle\_model\_3-5\_count\_lag\_1', 'vehicle\_model\_6-7\_count\_lag\_1',

'has\_etc\_0\_count\_lag\_1', 'has\_etc\_1\_count\_lag\_1',

'vehicle\_type\_0\_count\_lag\_1', 'vehicle\_type\_1\_count\_lag\_1',

'volume\_lag\_1', 'vehicle\_model\_0-2\_count\_lag\_2',

'vehicle\_model\_3-5\_count\_lag\_2', 'vehicle\_model\_6-7\_count\_lag\_2',

'has\_etc\_0\_count\_lag\_2', 'has\_etc\_1\_count\_lag\_2',

'vehicle\_type\_0\_count\_lag\_2', 'vehicle\_type\_1\_count\_lag\_2',

'volume\_lag\_2', 'vehicle\_model\_0-2\_count\_mean',

'vehicle\_model\_3-5\_count\_mean', 'vehicle\_model\_6-7\_count\_mean',

'has\_etc\_0\_count\_mean', 'has\_etc\_1\_count\_mean',

'vehicle\_type\_0\_count\_mean', 'vehicle\_type\_1\_count\_mean', 'volume\_mean',

'pressure', 'sea\_pressure', 'wind\_direction', 'wind\_speed',

'temperature', 'rel\_humidity', 'precipitation',

'other\_tollgate\_1\_has\_etc\_0\_count\_lag\_1',

'other\_tollgate\_1\_has\_etc\_1\_count\_lag\_1',

'other\_tollgate\_1\_volume\_mean', 'Month', 'Day', 'Day\_of\_week',

'Is\_weekend', 'Hour', 'Minute']

1. **Data Model & Performance**

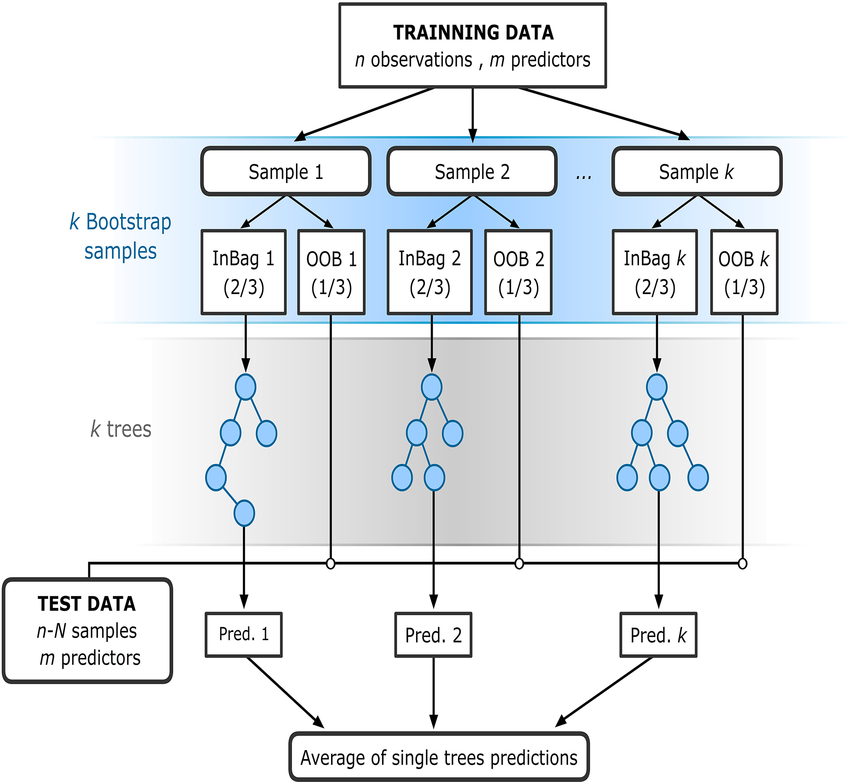
With the data engineering process mentioned above, we have discovered a lot of data correlation and generated a bigger dataset for model training by shifting the time windows. Based on our understanding of the project, we have designed 4 senorio master dataset for the project, and below are the master datasets we have used throughout the project:

* Whole set of data (including data under default time windows, and 3 additional data under moved time windows)
* Whole set of data (including data under default time windows, and 3 additional data under moved time windows) with cutting the data of non-busy hours
* Dataset under default time windows without cutting the data of non-busy hours
* Dataset under default time windows with cutting the data of non-busy hours

We have then done a quick performance checking for four senorio mentioned above. And we discovered that the senorio of “dataset under default time windows without cutting the data of non-busy hours” gives the best result. The senorio of “using additional data under moved time windows” gives worse results, which we guess is due to the reason of too much noise. While when we cut the “non-busy hours”, the results are not as good as we thought as well. Thus, we have selected the dataset under default time windows without cutting the data of non-busy hours.

After the preparation of the master dataset of data, we have split our data into 2 subsets from the master set, one set is the training set which includes the first 75% data and another subset is the testing set which includes the next 25% of data.

Next we have to choose our model. Based on our exploratory data analysis and the work in data engineering, we discovered that random forest regressor is one the best options in this project. Random forests is an [ensemble learning](https://en.wikipedia.org/wiki/Ensemble_learning) method for [regression](https://en.wikipedia.org/wiki/Regression_analysis) by constructing a multitude of [decision trees](https://en.wikipedia.org/wiki/Decision_tree_learning) at training time and outputting the class that is the [mode](https://en.wikipedia.org/wiki/Mode_(statistics)) of the prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of [overfitting](https://en.wikipedia.org/wiki/Overfitting) to their [training set](https://en.wikipedia.org/wiki/Test_set).



**Evaluation Metric**

In this project, mean absolute percentage error (MAPE) is the major calculation score.

However, by default, the random forest regressor in sklearn does not support MAPE. Thus, instead of MAPE, we have tried to use mean absolute error (MAE), mean squared error (MSE) as the metric to be minimized when training the random forest model.

For evaluation, we use both MAE and sMAPE (Symmetric MAPE) as the metric. sMAPE instead of MAPE is used because there are 0 volume for some entries in the data set, and this will cause the MAPE to be undefined. Therefore sMAPE is used as an alternative to MAPE.

|  |  |
| --- | --- |
| **MAPE** | **MAE** |
| **MSE** | **sMAPE** |

In this project, we have also tried to discover the impact of the error with different setting

* Usage of “Log-Transform” - which is used to check if minimize the impact from different scaling of the target values will help the model performance
* Global Model vs Local Model - which is used to check if the model performance will increase when we use a meaningful subset data group to train the model instead of using the whole dataset.

**With Log-Transform vs Without Log-Transform**

We have also explored the use of log transform as it is theoretically helpful to minimize the impact from different scaling of the target values. And below is the showcase of entry direction and exit direction error result in the cases of log transform and without any log transform:

Exit direction:

|  |  |
| --- | --- |
| Without log transform | With log transform |

Entry direction:

|  |  |
| --- | --- |
| Without log transform | With log transform |

From above, we can see that not using log transform yields better results in terms of both MAE and sMAPE.

**Global model vs local model**

Furthermore, as the dataset comes from different groups of data, we have made an assumption that the difference in between local models and global models may affect the error calculation. In general, local models use all of the available data for training, but internally separate the data into local groups and train a different model for each group, which is in contrast to global models that train a single model based on all training data.

And below is the showcase of entry direction and exit direction error result in the cases of global and local model:

Exit direction:

|  |  |
| --- | --- |
| Global Model (Using Global Model, All Tollgate testing set) | Global Models - Tollgate 1 (Using Global Model , Tollgate 1 testing set)  Local Models - Tollgate 1 (Using Local Model , Tollgate 1 testing set)  Global Models - Tollgate 3 (Using Global Model , Tollgate 3 testing set)  Local Models - Tollgate 3 (Using Local Model , Tollgate 3 testing set) |

Entry direction:

|  |  |
| --- | --- |
| Global Model (Using Global Model, All Tollgate testing set)    Global Models - Tollgate 1 (Using Global Model , Tollgate 1 testing set)  Local Models - Tollgate 1 (Using Local Model , Tollgate 1 testing set) | Global Models - Tollgate 2 (Using Global Model , Tollgate 2 testing set)  Local Models - Tollgate 2 (Using Local Model , Tollgate 2 testing set)Global Models - Tollgate 3 (Using Global Model , Tollgate 3 testing set)    Local Models - Tollgate 3 (Using Local Model , Tollgate 3 testing set) |

From above, we can see that

* For exit direction: Train global model is better overall
* For entry direction: Similar performance

**Final setting**

**Not using every lag**

Although we have prepared lagged values for 6 x 20 minutes, at the end not all of them are used. We dropped the lag 3-6 because they are insignificant in terms of feature importance.

**Not using log-transform**

As discussed above, not using log-transform yields better performance, so we decided not to apply log transform eventually.

**Model averaging**

From above, we can train both global and local models. To achieve higher stability in predictions, we adopt a simple model averaging.

* For each tollgate in exit direction: 0.5 \* prediction from global model + 0.5 \* prediction from local model
* For each tollgate in entry direction: 0.5 \* prediction from global model + 0.5 \* prediction from local model

Please note that for simplicity we just do a simple averaging here, the weights are actually trainable based on the model performance using the technique of stacking.

**Model Parameter Setting**

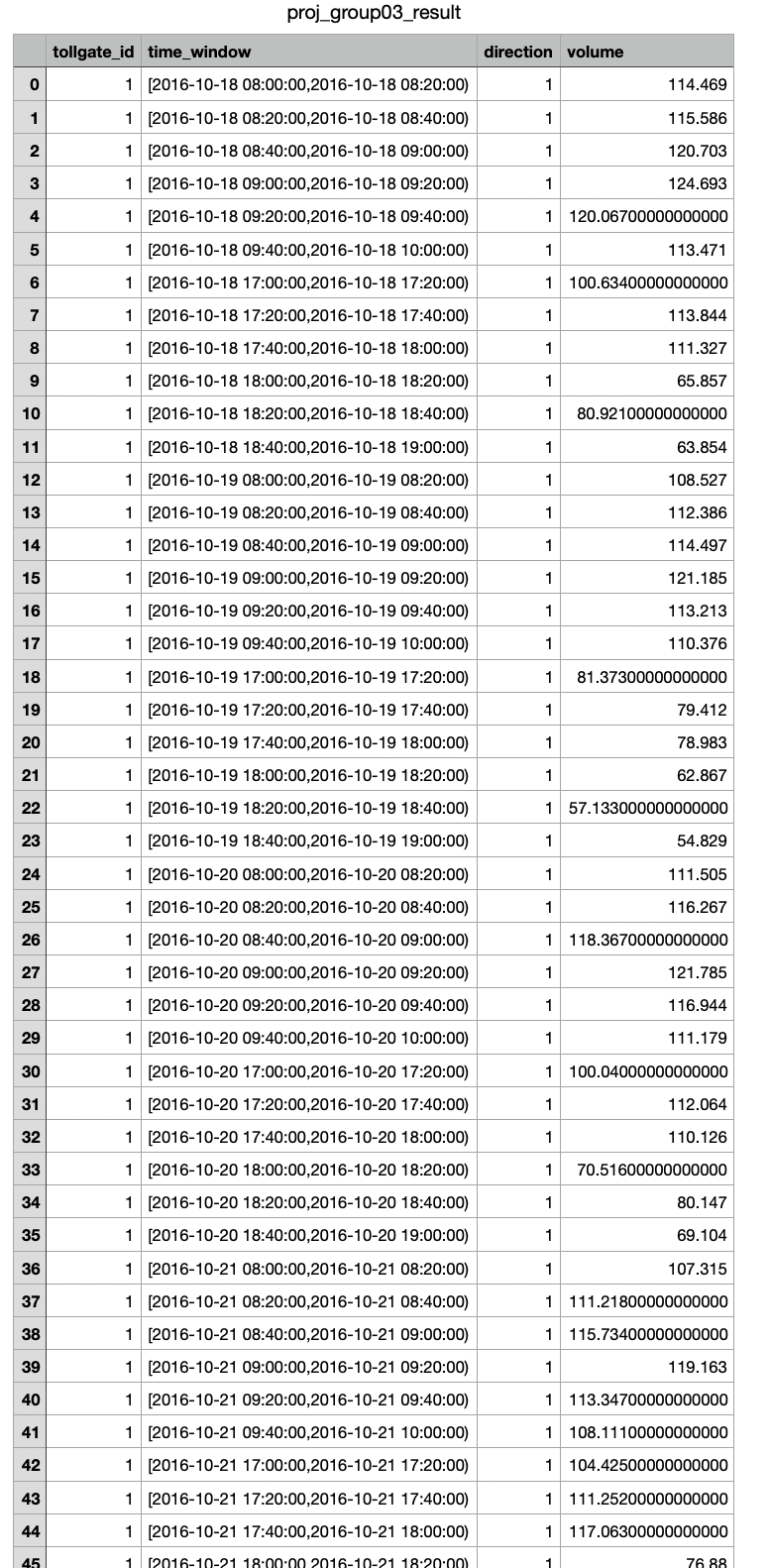
A model parameter is a configuration variable that is internal to the model and whose value can be estimated from data. They are required by the model when making predictions. And below are the model parameter setting we have used:

|  |  |  |
| --- | --- | --- |
| n\_estimators=500  criterion=mse  max\_depth=none  min\_samples\_split=2  min\_samples\_leaf=1  min\_weight\_fraction\_leaf=0 | max\_features=auto  max\_leaf\_nodes=None  min\_impurity\_decrease=0  min\_impurity\_split=None  bootstrap=True  oob\_score=False | n\_jobs=None  random\_state=0  verbose=0  warm\_start=False  ccp\_alpha=0  max\_samples=None |

**Prediction results**

In this project, as we don’t have the final verification set of data, we cannot further score our model. Below are the result sample that we have predicted:

(Details please refer the file “proj\_group03\_result.csv”)



1. **Conclusion**

In this project, we have selected task 2 and we successfully predict the average tollgate traffic volume as. Throughout the process, we discovered a lot of ways to massage the datasets to make it more meaningful for our training process. After the data engineering part, we have also tried to use different error functions and features combination to train and test our model, which is one of the excellent hand-on experiences throughout our life.