Predicting traffic volume based on correlating parameters

By:

Group 8-23

Josh Thoo Jen Sen Class 2018 4S1 (13)

Goh Koo Feng Class 2018 4S2 (05)

Chew Hong Jin Class 2018 4S2 (02)

Hwa Chong Institution

(High School)

Abstract

Data science is an interdisciplinary field to extract knowledge or insights from data in various forms. One of the uses of data science is in machine learning, a method to analyse data that automates analytical model building. In a metropolitan city, commuters must plan their commutes accordingly to fit into their tight schedule. This project aims to create an algorithm based on historical data to predict traffic volume in the Central Business District (CBD) of Singapore, allowing commuters to reach their destinations on time.

In the first research question, various conditions that affect traffic volume were investigated. It was found that the presence of ERP gantries affects traffic the most, followed by the time of the day. The correlation of rain and day of the week with traffic conditions were statistically insignificant. In the second research question, data from the LTA API was extracted for use in the created machine learning algorithm applying the Random Forest Algorithm that could predict traffic conditions with 43% accuracy. In the third research question, several optimisations of the dataset were conducted, improving algorithm accuracy to 63%. Alternative programs that yielded a higher accuracy rate were analysed, with the KNN algorithm (k=11) and Regression Algorithm yielding 90% and 68% accuracy respectively. The KNN algorithm was the most accurate machine learning algorithm and will be used to predict traffic volume in the CBD of Singapore.

(229 words)

1 Introduction

Data science is a concept “to unify statistics, data analysis, machine learning and their related methods" in order to "understand and analyse actual phenomena" with data (Simplystatistics.org. 2018). One of the uses in data science is in machine learning, a method to analyse data that automates analytical model building.

In a metropolitan city like Singapore, people have to get from point to point on time. Thus, it is important to know the conditions of the roads such as traffic volume ahead of time so that commuters can plan accordingly to fit into their tight schedule. One such method is to use an algorithm using supervised machine learning based on certain parameters such as weather conditions and time of the day through the collection of publicly available data. This is to ensure that commuters can reach their destinations on time to maximise efficiency of time usage. Hence, such an algorithm will bring much convenience to commuters and this is the aim of this project.

2 Objectives

1. To identify the conditions that affect traffic volume in a particular area.
2. To investigate on the different machine learning methods to predict traffic volume.
3. To identify applications of this algorithm in sectors such as transportation.

3 Research questions

1. How heavily do certain conditions affect traffic volume?
2. How to create an algorithm that can predict traffic volume based on given conditions using machine learning?
3. Are there other algorithms that would yield a more accurate result?

4 Methodology

1. To read up and investigate on the different machine learning algorithms that can be used to predict traffic volume.
2. To identify any correlation between parameters and traffic volume from public datasets using R and Python.
3. To model and tabulate the prediction of the machine learning algorithm into the Singapore map for public commuter usage through interactive maps like Tableau.

5 Results and Analysis

5.1 Weighted average of factors that affect traffic volume

First of all, the major factors that affect traffic volume were identified and are as follows: Time of day, Day of week, Weather condition and ERP gantries. The data for each of the variables were collected on the Application Program Interface (API) by LTA, known as DataMall. DataMall contains varying data that shows the current traffic speed in the roads of Singapore in real-time. Based on these 4 variables, the regression method was adapted to find the weighted average for

each independent variable which was used to determine the strength of relationship in predicting traffic band speed within the CBD region of Singapore. Using the regression model, we were able to:

1. Derive the best-fit line / best-fit curve for future predictions where the output is unknown.
2. Determine if a correlation relationship exists between 2 variables and find out whether this correlation relationship is statistically significant (to disprove the null hypothesis, $p<0.05$)

The gradient of the line of best fit and intercepts are given by the following equation that minimises the sum of the squared *x* distance of the points from the line, given a linear regression model $y = mx + c$:

$m = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sum_{i=1}^{n}(x_i-\bar{x})^2}$, where *m* is the slope, and *n* is the number of data points, and subsequently solve for y-intercept using

$c = \bar{y} - m\bar{x}$, where $\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$, the mean of the x-coordinate and $\bar{y} = \frac{\sum_{i=1}^{n} y_i}{n}$

The p-value determines the chance that the probability a detected relationship is only due to random variation, and is given by: $p = 2 \times P(\frac{m}{se(m)} > t_{n-2})$, where *se(m)* refers to the standard error of m, given by: $se(m) = \sqrt{\frac{\sum_{i=1}^{n} e_i^2}{(n-2)\sum_{i=1}^{n}(x_i-\bar{x})^2}}$, t_{n-2} refers to the normal distribution when the null hypothesis holds true.

Using the linear regression model, the value of *m* reflects on how heavily a certain variable affects traffic speeds. The larger the magnitude of the value of *m*, the more heavily that condition affects traffic speed. The p-value is used to determine the reliability of this correlating relationship.

It is found that the presence of ERP gantries affects traffic volume the most, while the time of the day affects traffic volume to a smaller extent. It is inconclusive whether the day of the week and the presence of rain affects traffic conditions at all as their relationships are statistically insignificant. These results are supported by the ordinal regression model (Appendix V).

5.2 Creation of algorithm to predict traffic volume

In the R program used to predict traffic band speeds (very slow: 0-20km/h, slow: 21-40km/h, fast: 41-60hm/h, very fast: 60+km/h) within the CBD area, the random forest algorithm was utilised (Fig 1). Random forest is an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. It encompasses the method of bootstrap aggregating (also known as bagging), which is the essence of improving the stability and accuracy of the predicted value (Appendix VI). In this algorithm, a total of 5, 10, 15, and 20 trees are created using 84 test points and 84 data points from the 9 segments of road forming the perimeter of the CBD region (Appendix IV). This yields us a total of 168 data points.

```
getwd()

setwd("D:/schoolstuff/datascraping")
d <- read.csv('cbd_connaughtdr.csv')
set.seed(16)
d <- d[sample(nrow(d)),]
train <- d[1:84,]
test <- d[85:168,]
library(randomForest)
my_forest <- randomForest(as.factor(TrafficSpeed) ~ TimeOfDay + DayOfWeek + Gantries,

                          data = train,

                          ntree = 10 )

varImPlot(my_forest)
rf_prediction <- predict(my_forest, test)
rf_prediction == test$TrafficSpeed
score= sum(rf_prediction == test$TrafficSpeed)
print(paste("Number of correct predictions= ", score, "/ 84"))
```

Fig 1: The random forest algorithm

For the algorithm with 5 and 10 trees, there were a total of 32/84 predictions that are correct. For 15 trees, there were a total of 34/84 correct predictions. For 20 trees, there were a total of 36/84 correct predictions (Appendix VII). This means that the random forest algorithm achieves the highest accuracy at 20 trees of 43%. The baseline accuracy of the dataset was 28.5% (mode: traffic condition 4), consisting of 48 of 168 incidences in the data, 13.5% lower than our results (Appendix IV). Although the machine learning algorithm is user friendly, results had proven it has a relatively low accuracy rate which might not be effective in predicting the traffic condition. This is due to insufficient variables and sizeable number of datasets.

5.3.1 Optimisation of Data

To counteract the problem of limited datasets, the dataset was optimised. Previously, only data collected from 1 segment of road, consisting only of 168 data points, was used. Since a larger dataset with more data points yields an algorithm with higher accuracy, data from all 9 points were combined together to form a combined dataset 9 times larger than the original dataset consisting of 1512 data points.

Next, modifications to inputs were made. An additional variable, the presence of rain, was added by collecting historical weather data from the Paya Lebar Weather Station, the weather station geographically closest to the CBD region. In addition, banding was done to create binary inputs that could be more easily interpreted by logistical algorithms (dealing with binary classifications). The following binary inputs are used:

→ Weekdays were given a value 1 while weekdays were given 0.

→ Peak hours are given value 1 while off-peak hours are given 0.

The dataset whose inputs were changed to binary classification will be named the banded dataset. Rather than an even split using a randomiser, data points are taken away from the test data into the training data so that the algorithm has more data points to analyse. The ratio of training data to test data was also increased from 1:1 to 13:1 so that a consistent size of test points is taken to analyse the different accuracies. When the algorithm is computed (Appendix VIII), the algorithm with 5 trees yield a total of 41/84 correct predictions. For 10 trees, there are a total of 51/84 correct predictions. For 15 and 20 trees, there are a total of 52/84 correct predictions, and accuracy rate of 63%. This proves that the optimisation of the dataset has increased the accuracy of the machine learning algorithm with 20 decision trees by 20%.

5.3.2 Creation of an alternative algorithm

In the third research question, 2 algorithms were investigated, including the k-Nearest Neighbour algorithm (KNN). This algorithm allows the relationship between the dependent and independent variables to be captured and through this process, learn a function so that given its parameters (time of day, day of week etc.), the algorithm can confidently predict the corresponding output (Appendix IX). This algorithm stands out as it makes no explicit assumptions about the functional form of the data set, therefore avoiding the dangers of misfitting the underlying distribution of the data. The correlation of the data and its output is determined by training points with the lowest Euclidean distance to the test point. The test point then takes on the output which most training points have been assigned to. The Euclidean distance between points P with coordinates ($P_{1,}$ $P_{2...}P_n$) and points Q with coordinates ($Q_{1,}$ $Q_{2...}$ Q_n) is given by:

$$d(P,Q) = \sqrt{(P_1 - Q_1)^2 + (P_2 - Q_2)^2 + \cdots + (P_n - Q_n)^2}$$

The algorithm then runs through the whole dataset and find the number of points closest to the test point (Appendix X). The input is then assigned to the output with the highest conditional

probability. In the case of this algorithm, the data used in the 1st week will be used to model the results of 2nd week. Based on the algorithm, with *K=5* nearest neighbours, an accuracy of 560/756(74.07%) is achieved. Subsequently, for *K=7,* the accuracy is 552/756(73.02%), for *K=9,* the accuracy is 675/756 (89.29%), for *K=11,* the accuracy is 681/756 (90.01%), for *K=13,* the accuracy is 678/756 (89.68%), for *K=15,* the accuracy is 641/756 (84.79%). The use of odd *K* value is to ensure that a tied situation don't exist.

5.3.3 Linear Regression method

The underlying principle of this method is covered in Research Question 1 where the correlation between the independent variables (Time of Day, Day of Week, ERP gantries, Rain) and dependent variables (Traffic speed band) were investigated and tabulated. Using the algorithm (Appendix XI), the results is tabulated as follows, for $y = 0.5$ and set.seed(15), training size = 602 and test points = 910 :

Actual Value	Predicted Value	
	FALSE	TRUE
1	39	29
2	155	221
3	98	234
4	7	127

A "TRUE" value means that actual value matched predicted value, while a "FALSE" value means that actual value is different from predicted value. Thus, the total accuracy is as follows:

$$\text{Total accuracy} = \frac{29 + 221 + 234 + 127}{29 + 221 + 234 + 127 + 39 + 155 + 98 + 7} = \frac{611}{910} = (67.14\%)$$

Correspondingly, For $y = 0.7$ and set.seed(15), training points = 908 and test points = 604, the accuracy is $\frac{411}{604} = 68.05\%$, and for $y = 0.9$ and set.seed(15), training points = 1210 and test points = 302, the accuracy is $\frac{206}{302} = 68.21\%$ (Appendix XI).

5.3.4 Conclusion

Optimisation of the used dataset was done, increasing the accuracy of the Random Forest algorithm from 42.86% to 61.90%. Other algorithms, namely the k-Nearest Neighbours(KNN) algorithm and linear regression were also considered for comparison to explore which algorithm is best for predicting traffic volume. With optimised parameters, the KNN algorithm obtained 90.01% accuracy while the linear regression algorithm obtained 68.21% accuracy. Among the 3 algorithms, the optimised KNN algorithm was the most accurate and will thus be used to predict traffic volume.

6 Applications, Limitations and Further Extension

With the machine learning algorithm, an accurate predictor of traffic volume can be constructed for planning of the travel routes of commuters, providing them convenience through ease of travel. Transport agencies like the LTA can use the machine learning algorithm prediction tool to plan the timings of their transport services such as the frequency of trains, as well as the travel routes of buses. This can be used as navigation applications that take into consideration the prediction of slow traffic by the machine learning algorithm to find alternative routes, providing drivers more convenient ride and improve the quality of life of residents in their daily commute.

In comparison to our project's machine learning algorithm, existing methods only use real-time traffic conditions, and future possible traffic is generalised based on historical data by observing

the modal traffic condition on that particular road segment at that particular time. However, the existing methods fail to consider the influence of other correlating variables other than the time of day, such as day of week, ERP gantries, and rain forecast that will affect traffic conditions, making their predictions vague and inaccurate at times.

The accuracy of the algorithm used in this project can be improved with the increase in the size of the dataset. Due to computational limitations and difficulty in data collection, our dataset was limited to size 1512, making our machine learning algorithm relatively inaccurate as opposed to machine learning algorithms run by supercomputers that can have accuracies of nearly 98%. With greater computational availability with data collected over greater timespans, the machine learning algorithm will have more data to analyse and thus be more accurate.

Other traffic conditions such as traffic accidents were also not considered in this project due to their differing extents and influence, despite being a significant correlating variable to traffic conditions. An estimation and quantification of the severity of traffic conditions through analysis will provide another variable for comparison, which would increase accuracy of the algorithm.

A composite algorithm could also be work on which makes use of the underlying principle of bagging. The aim of the composite algorithm is to consider the outputs of all 3 different algorithms utilised in this project and display a more cogent outcome so that the weaknesses of the separate algorithms are supported by the strengths of others, which will improve their accuracies as opposed to a single machine learning algorithm used.

7 References

[1] Simplystatistics.org. (2018). The key word in "Data Science" is not Data, it is Science · Simply Statistics. [online] Available at: https://simplystatistics.org/2013/12/12/the-key-word-in-data-science-is-not-data-it-is-science/ [Accessed 8 Mar. 2018].

[2] Naur, P. (1974). Concise Survey of Computer Methods. 1st ed. Lund: Studentlitteratur, pp.Chapter 1.1. Retrieved 7 March 2018.

[3] National Science Board. "Long-Lived Digital Data Collections Enabling Research and Education in the 21st Century". National Science Foundation. Retrieved 6 March 2018.

[4] Cleveland, W. S. (2001). Data science: an action plan for expanding the technical areas of the field of statistics. International Statistical Review / Revue Internationale de Statistique, 21–26. Retrieved 7 March 2018.

[5] Khalifa et al. (2017). Machine Learning Approaches for Traffic Volume Forecasting:. Retrieved March 12, 2018, from https://www.thesisscientist.com/docs/Articles/a4f09b95-0265-4480-a47e-a8bd6f81cdac

[6] LaMorte, W. (2016, May). The Multiple Linear Regression Equation. Retrieved 11 August, 2018, from http://sphweb.bumc.bu.edu/otlt/MPH-Modules/BS/BS704-EP713_MultivariableMethods/BS704-EP713_MultivariableMethods2.html

[7] Zakka, K. (2016, July). A Complete Guide to K-Nearest-Neighbors with Applications in Python and R. Retrieved 12 August, 2018, from https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/

[8] Akhil et al. (2013). Classification of Heart Disease Using K- Nearest Neighbor and Genetic Algorithm. Retrieved 10 August 2018, from arxiv.org/ftp/arxiv/papers/1508/1508.02061.pdf

Appendix

Appendix I: Terminology

Term	Explanation
Banded Dataset	The dataset whose inputs are changed to binary classification.
Baseline Accuracy	The frequency of the mode in the dataset. A method that uses heuristics, simple summary statistics, randomness, or machine learning to create predictions for a dataset to measure the accuracy/performance of the result.
Conditional Probability	A measure of the probability of an event given that another event has occurred.
Confidence Interval	A type of interval estimate, computed from the statistics of the observed data, that might contain the true value of an unknown population parameter.
Decision Tree	An aggregator that compiles the provided sample data into a single final result.
Functional Form	It refers to the algebraic form of a relationship between a dependent variable and regressors or explanatory variables.
Hyperparameter	It is used to distinguish from parameters of the model for the underlying system under analysis.
K-Nearest-Neighbours	A classification algorithm that predicts an output of a test point based on the similarity of their inputs with training points with output already given.
Machine Learning	A field of computer science where algorithms are fed data to improve its accuracy on a specific task without explicit programming.
Null Hypothesis	A general statement that there is no relationship between two measured phenomena.

Random Forest	A method of averaging multiple deep decision trees, trained on different parts of the same training set, with the goal of reducing the variance.
Regression	A statistical method to identify the strength of relationship between a dependent variable.

Appendix II: Background on Data Science

The term “data science” was first used in 1974 in computer scientist Peter Naur’s book “Concise survey of computer methods” as an alternative term for computer science. Naur (1974) defined this as “The science of dealing with data, once they have been established, while the relation of the data to what they represent is delegated to other fields and sciences.” The National Science Board proposed an alternative definition in their 2005 paper "Long-lived Digital Data Collections: Enabling Research and Education in the 21st Century", defining data scientists as those that help with "the successful management of a digital data collection" with “creative inquiry and analysis" using the data (National Science Board, 2005).

Cleveland (2001) introduced the concept of data science being an interdisciplinary field of statistics that incorporates “advances in computing with data”. The article sets out six technical areas that data science involves, namely: multidisciplinary investigations, models and methods for data, computing with data, pedagogy, tool evaluation, and theory.

One of the uses in data science is in machine learning, one of the computer algorithm analyses data that automates analytical model building. These algorithms can outperform traditional programs and professionals - for example, in October 2015, Google’s AlphaGo, beat the 9-dan (highest Go rating) professional Go player Lee Sedol 4-1 without handicaps and in August 2017, Elon Musk’s

open AI that used unsupervised machine learning beat top professionals in the popular online game DOTA 2 without handicaps.

Appendix III: Used Methods in Predicting Traffic Volume

According to Scientist, T. (n.d.)., their first attempt to approach on traffic volume forecasting on the Moroccan Highway Network was mainly based on statistical models, but it comes to give a poor performance on most stations. This was due to the high non-seasonal variations in the data that these traditional models couldn't explain. Next, they set out to move on by adopting some machine learning techniques. After testing several algorithms, they concluded the ones with the best accuracy rates, which include:

Random Forest: It is a supervised classification algorithm. As the name suggest, this algorithm creates the forest with a number of trees. In general, the more trees in the forest the more robust the forest looks like. In the same way in the random forest classifier, the higher the number of trees in the forest gives the high accuracy results. Its main parameters are: the number of trees in the forest, the number of features to consider when looking for the best split, the minimum number of samples needed to split an internal node and the minimum number of samples required to be at a leaf.

Multi-layer Perceptron: An MLP is a network of simple neurons called perceptrons. The perceptron computes a single output from multiple real-valued inputs by forming a linear combination according to its input weights and then possibly putting the output through some nonlinear activation function. Multi-layer Perceptron is sensitive to feature scaling, so it is highly recommended to scale the data. Its parameters include: The regularization parameter, The number of hidden layers and number of neurons within each layer and maximum number of iterations.

Appendix IV: Data collection

First of all, the major factors that affect traffic volume are identified and are as follows: Time of day, Day of week, Weather condition and ERP gantries. The data for each of the variables are requested on the Application Program Interface (API) by LTA, known as DataMall. Access was granted as the LTA provided an Account Key that provided access to this API, through an external application called Postman. The API contains varying data that shows the current traffic speed in the roads of Singapore in the real time.

Using the Postman function Collections, 9 different API requests were requested and logged into a common Excel file. The 9 API requests correspond to the filter links for each of the 9 analysed roads:

1. Sheares Avenue
2. Bayfront Avenue
3. Marina Boulevard
4. Central Boulevard
5. Keppel Viaduct

6. Cantonment Road
7. Shenton Way
8. Esplanade Drive
9. New Bridge Road

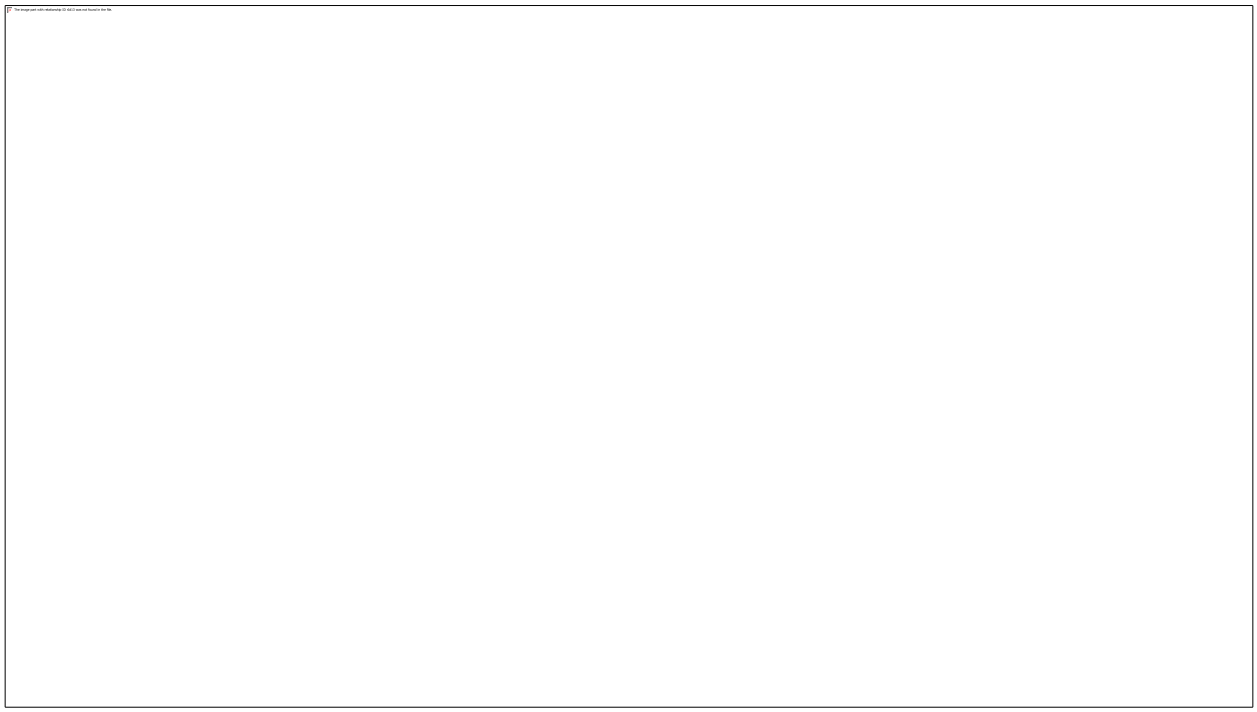


Fig 2: Perimeter of the CBD area of Singapore

For segments with multiple entries, only the 1st one is considered and added into the Excel file. The 9 segments surround the inner CBD and aim to accurately act as a representation of traffic in the CBD.

Procedure:

- Data was collected hourly from 8.00am to 8.00pm between 4 and 17 June, logging 168 rows of data points per segment of road, yielding 1512 data points.
- Column titles and the column values of Time of Day, Day of Week and ERP Gantries were added manually using Excel functionality.

- Using historical weather data collected by the Paya Lebar Weather Station, the presence of rain was added into the datapoint as an additional input (through online website https://www.wunderground.com)

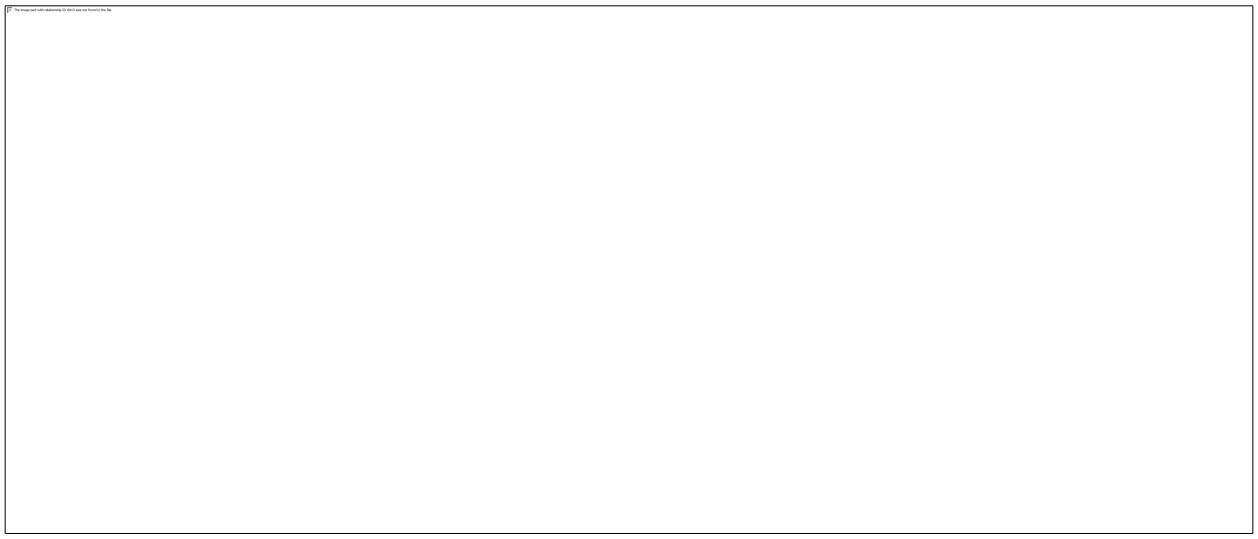


Fig 3: Weather data from Paya Lebar Weather Station

Appendix V: Statistical significance of each variable

Using the dataset collected in the data collection process, several bandings were used to simplify regression calculations:

1. Off-peak hours (10-4pm) were given a value 0 while peak hours (8-9am, 5-7pm) were given a value 1.
2. Weekdays were given a value 1 while weekends were given a value 0.

The library(ggplot2) for the R program to calculate regression models was utilised for ease of programming.

The following R program was run, substituting “Condition” for the analysed input at interest.

```
>library(ggplot2)
library(htmltab)

dataset <- read.csv("E:/data/cbd_dataset.csv")
names(dataset)
fit <- lm(TrafficConditions~Condition, d=dataset)
summary(fit)
```

Traffic Condition 1: Rain

A p-value of 0.271 is obtained, while its gradient, *m*, is 0.071. This indicates a weak and statistically insignificant correlation relationship.

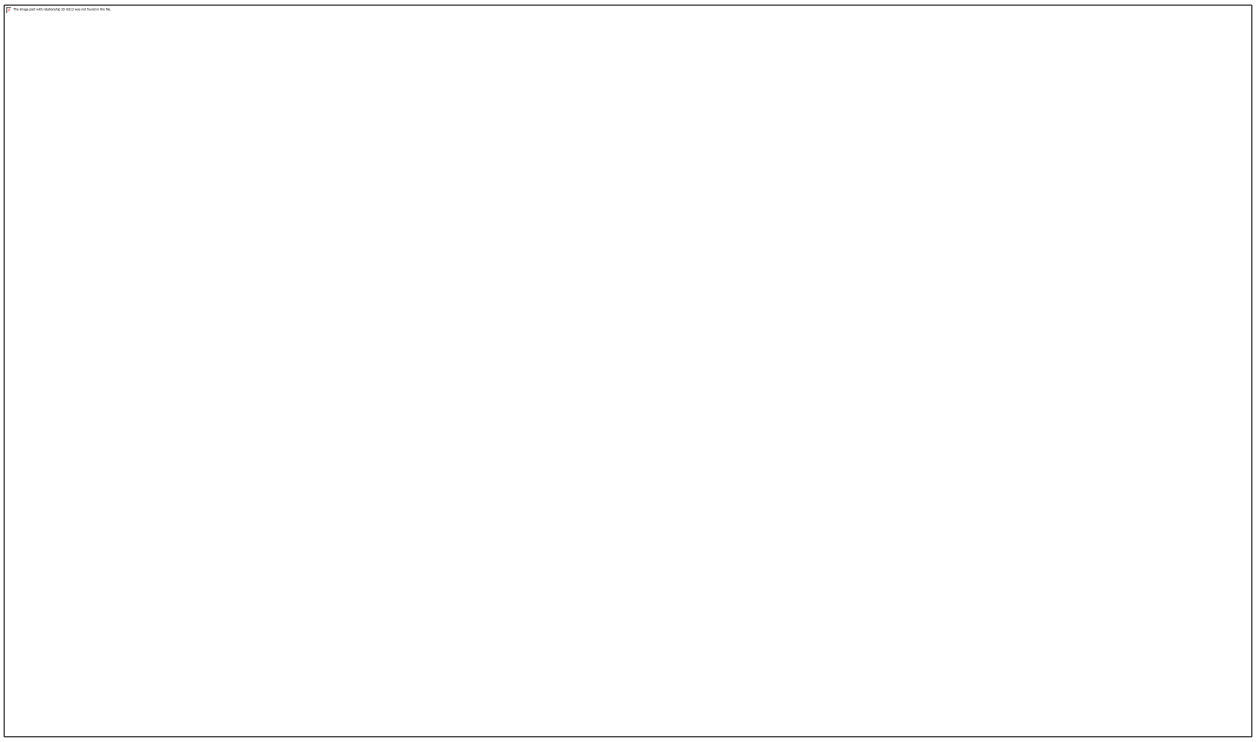


Fig 4: R program demonstrating the p-value of rain condition

Traffic Condition 2: Presence of ERP Gantries

A p-value of 2.2×10^{-16} is obtained with gradient 1.04. This indicates an extremely reliable and strong correlation relationship.

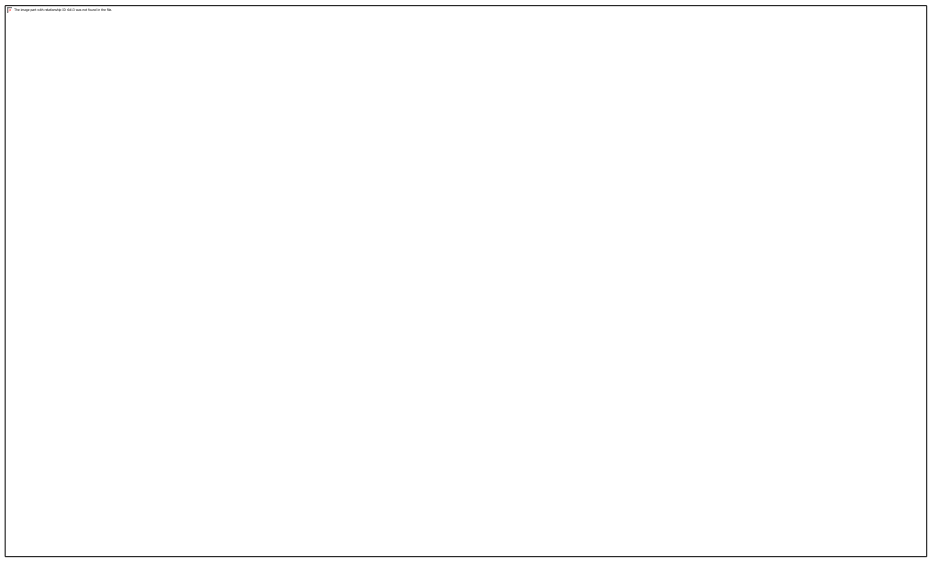


Fig 5: R program demonstrating the p-value of ERP gantry

Traffic Condition 3: Time of the day

A p-value of 0.000028 was obtained with gradient -0.181. This indicates an extremely reliable and strong coefficient relationship.

Fig 6: R program demonstrating the p-value of time of the day

Traffic Condition 4: Day of the week

A p-value of 0.498 is obtained while its gradient was -0.032. This indicates a statistically insignificant correlation relationship.

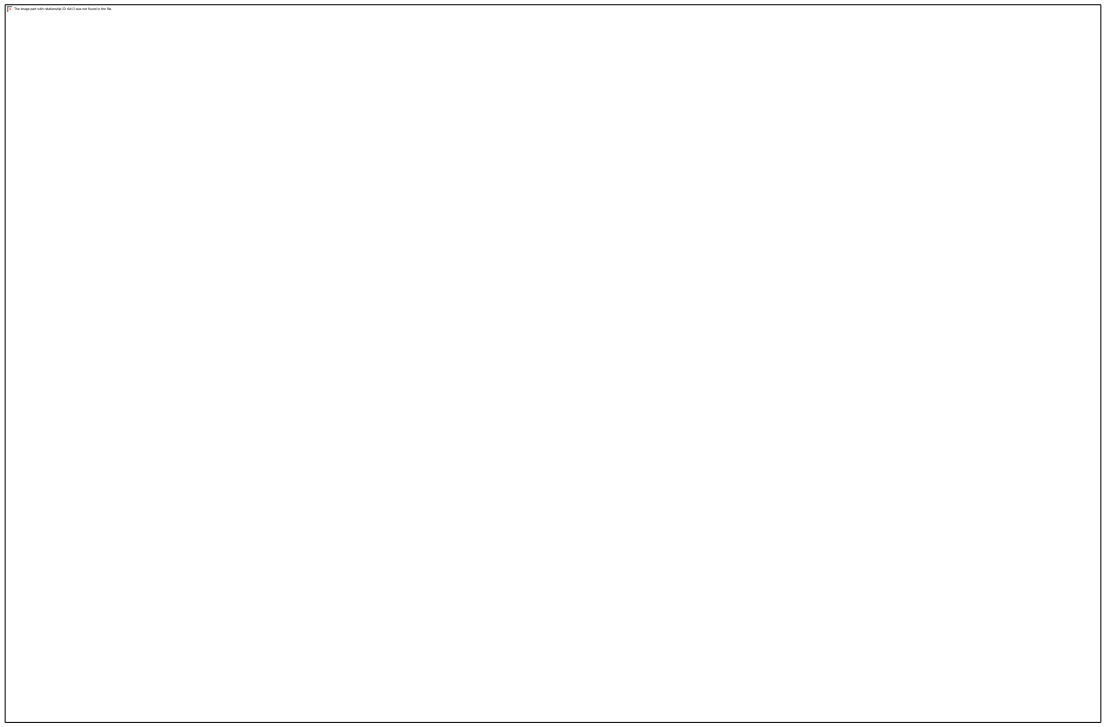


Fig 7: R program demonstrating the p-value of day of the week

Additional Test 1: Consideration of all factors with multi-variable regression

Multi-variable regression allows for the comparison of the strength of relationship between many inputs and a single output.

The multiple linear regression equation is as follows:

$\hat{y} = b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_p x_p$

where $\hat{y}$ is the predicted or expected value of the dependent variable, x_1 to x_p are p distinct independent variables, b_0 is the value of y when all of the independent variables (x_1 through x_p) are equal to zero, and b_1 through b_p are the estimated regression coefficients. Each regression coefficient represents the change in y relative to a one unit change in the respective independent variable. In the multiple regression situation, b_1, for example, is the change in y relative to a one unit change in x_1, holding all other independent variables constant (i.e., when the remaining independent variables are fixed). Again, statistical tests can be performed to assess whether each regression coefficient is significantly different from zero.

If it is found that an initial input has an insignificant relationship with the output after consideration of a second input, its p-value would increase due to the decrease in confidence intervals. This would signal that the correlation relationship of the former input is weaker than the correlation relationship of the latter input.

The following R program was used to find the regression of the 4 independent variables:

```
library(ggplot2)
library(htmltab)

dataset <- read.csv("E:/data/cbd_dataset.csv")
names(dataset)
fit <-
lm(TrafficConditions~factor(TimeOfDay)+factor(DayOfWeek)+factor(Gantri
es)+factor(Rain) , d=dataset)
summary(fit)
```

Fig 8: R program for the regression of 4 independent variables

Table of p-value for respective independent variables using multi-variable regression

Condition	Single-variable p-value	Multi-variable p-value	Single-variable Best-fit line gradient	Multi-variable Best-fit line gradient
Time of Day	0.00028	0.00000264	-0.181	-0.178
Day of Week	0.498	0.395	-0.032	-0.035
Gantries	2.2×10^{-16}	2×10^{-16}	1.04	1.03
Rain	0.271	0.686	0.071	0.023

Addition Test 2: Use of ordinal analysis

Ordinal analysis is a type of regression analysis used for predicting an ordinal variable, a variable whose value is based on arbitrary classifications. In ordinal analysis, the generalised linear model (GLM) is used. The GLM is a generalization of ordinary linear regression that allows for response variables that have error distribution models other than a normal distribution. In retrospect, ordinary linear regression only predicts the expected value of a given unknown quantity (the response variable, a random variable) as a linear combination of a set of observed values. This

implies that a constant change in a predictor leads to a constant change in the response variable (i.e. a linear-response model). This is appropriate when the response variable has a normal distribution. However, these assumptions are inappropriate for some types of response variables. On the other hand, GLM cover all these situations by allowing for response variables that have arbitrary distributions (rather than simply normal distributions), and for an arbitrary function of the response variable (the link function) to vary linearly with the predicted values. This is done by assuming a dependent variable is related to the independent variable through a simple exponential function (considering different possible distributions including normal, binomial, Gamma and Poisson) with a linear coefficient:

$$E(Y) = g^{-1}[X\beta]$$

where g is the identified most representative exponential function, X is a linear parameter, and β is the value of the independent variable. In this project, ordinal analysis is incorporated to find the p-value of the 4 traffic conditions and see whether the p-value of each independent traffic condition corresponds to the multi-variable regression method.

Tests 1 and 2: Ordinal analysis of all 4 factors with a banded and unbanded dataset

The following R program is used, where "cbd_dataset.csv" refers to the used dataset.

```
>library(foreign)
library(ggplot2)
library(MASS)
library(Hmisc)
library(reshape2)
dataset <- read.csv("E:/data/cbd_dataset.csv")
m <- polr(factor(TrafficConditions)~Gantries+TimeOfDay
                      +DayOfWeek+Rain, data=dataset, Hess=TRUE)
summary(m)
```

```
ctable <- coef(summary(m))

p <- pnorm(abs(ctable[,"t value"]), lower.tail=FALSE)*2
ctable <- cbind(ctable, "p value"=p)
```

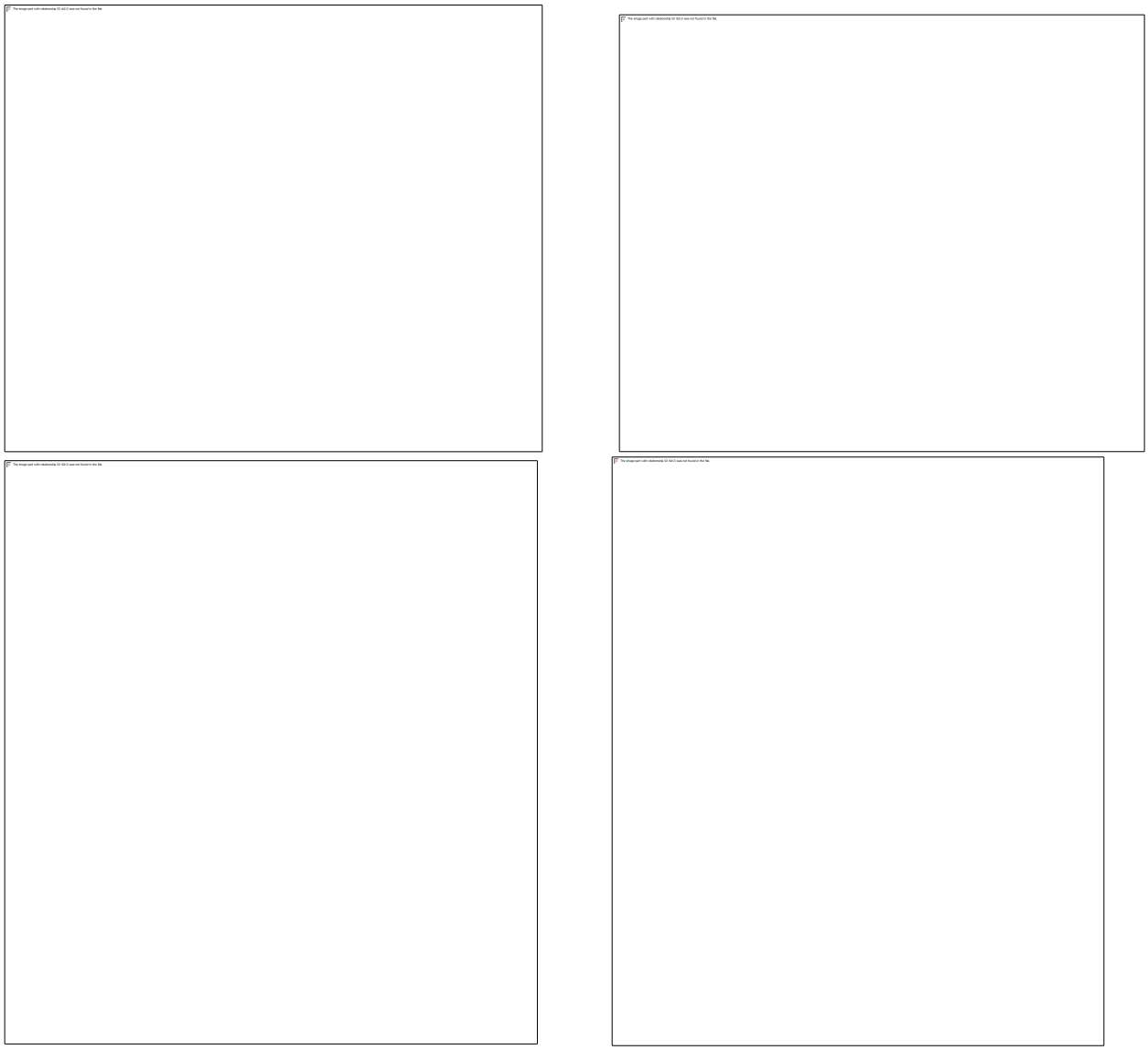


Fig 9: Compilation of R programs for the p-value of the independent variables using ordinal analysis

Table of p-value for respective independent variables using ordinal analysis

Variables	p-value of banded dataset	p-value of unbanded dataset	m, gradient of banded dataset	m, gradient of unbanded dataset
Time of Day	0.0000364	0.0463	-0.47	0.03
Day of Week	0.496	0.667	-0.07	0.01

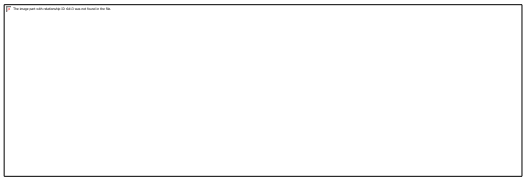
Gantries	1.41×10^{-84}	1.00×10^{-83}	2.96	2.94
Rain	0.817	0.248	0.04	0.17

Results

The results obtained from the ordinal regression analysis support the conclusion obtained from linear regression and multi-variable regression, indicating that Time of Day and ERP gantries are the major factors that influence traffic condition.

Appendix VI: Bootstrap aggregating (Bagging) used in random forest algorithm

The random forest algorithm is a supervised classification algorithm that creates a forest with a number of decision trees. The higher number of decision trees in the forest yields a higher accuracy. Bagging minimises the 'noise' variance (due to random chance or sporadic events) to create a model with lower variance, which would make the algorithm more accurate. This is also why more trees yield more accurate results, as the error through random chance is lowered. Given a set of data, S, a randomizer will split the data into a number of matrices, where f_n represents the *nth* data point, and letters *a, b and c* represent the variables used for the generation of the output *C*.



With the set, a decision tree is created such that the output from the decision tree always agrees with the data in its corresponding matrix. Else, the decision tree will find the permutation that yields the highest accuracy. This is done so by selecting a random sample with replacement of the training set and fits trees to these samples. For instance, given the dataset *S*, bagging repeatedly (B

times generate n training examples to form another set of data S_b. Then, the random forest algorithm will train a classification or regression tree f_b on S_b. After training, predictions for unseen samples x' can be made by averaging the predictions from all the individual regression trees on x', given by: $\hat{f} = \frac{1}{B}\sum_{b=1}^{B} f_b(x')$

This bootstrapping procedure leads to better model performance because it decreases the variance of the model, without increasing the bias. This means that while the predictions of a single tree are highly sensitive to noise in its training set, the average of many trees is not, as long as the trees are not correlated. Simply training many trees on a single training set would give strongly correlated trees; bagging is a way of de-correlating the trees by showing them different training sets.

Appendix VII: Random forest algorithm for varying trees (variables)

Fig 10: Random forest algorithm for 5 trees

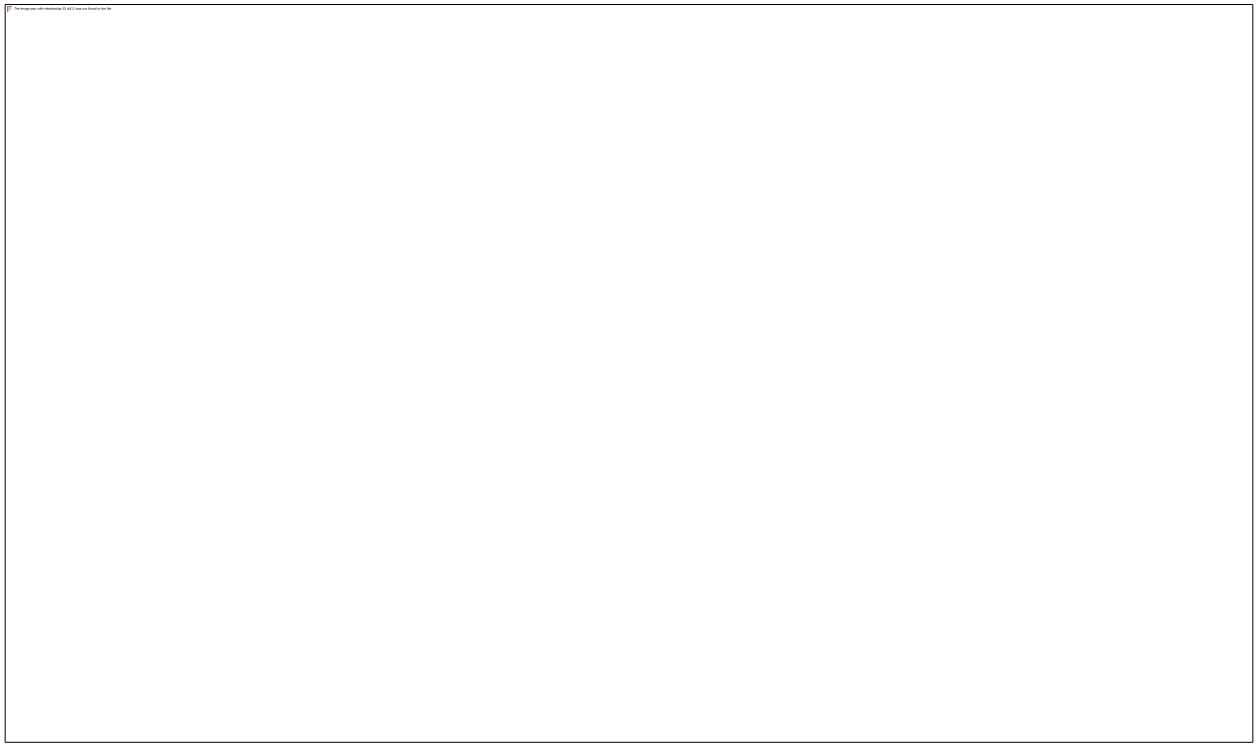


Fig 11: Random forest algorithm for 10 trees

Fig 12: Random forest algorithm for 15 trees

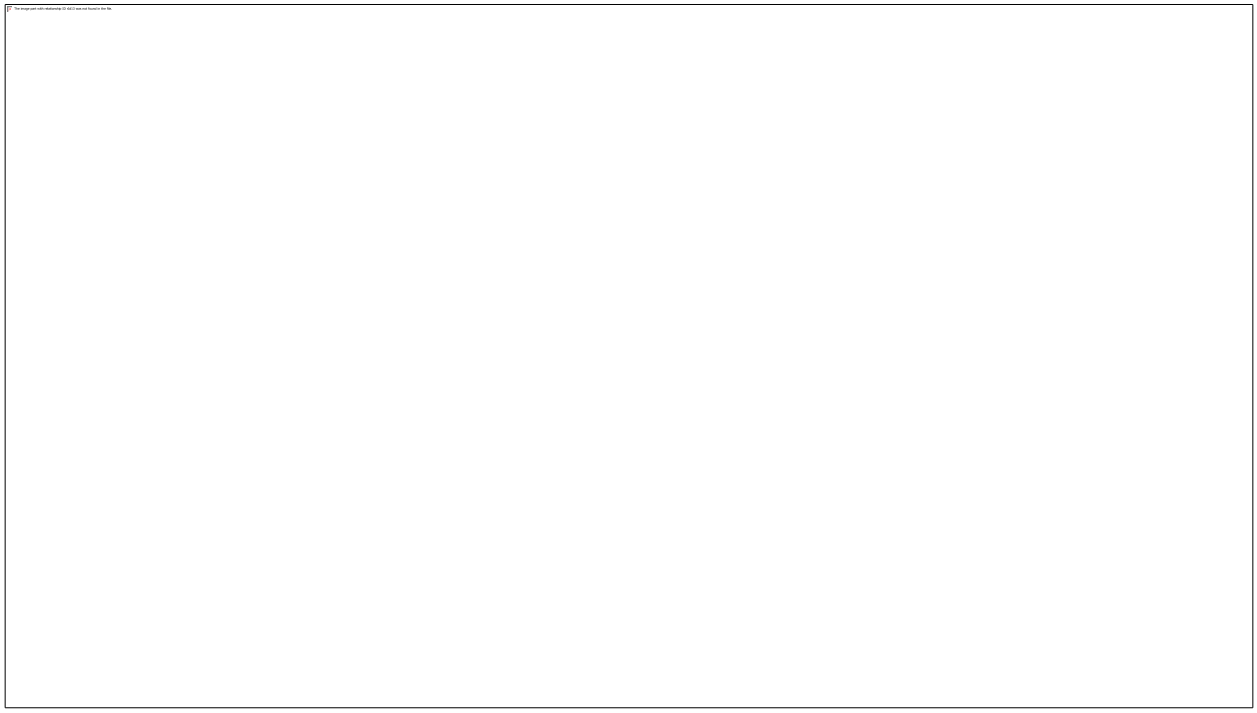


Fig 13: Random forest algorithm for 20 trees

Appendix VIII: Results after data optimisation

The following random forest algorithm with 1428 training points and 84 test points was ran:

```
>library(ggplot2)
d <- read.csv("E:/data/cbd_dataset.csv")
set.seed(1512)
train <- d[1:1428,]
test <- d[1429:1512,]
library(randomForest)

My_forest <- randomForest(as.factor(TrafficConditions) ~ TimeOfDay +
DayOfWeek +
                            Gantries + Rain,
                            data = train,
                            ntree= x,)
rf_prediction <- predict(My_forest, test)
rf_prediction == test$TrafficConditions
score = sum(rf_prediction == test$TrafficConditions)
```

```
print(paste("Number of Correct Predictions =" , score, "/84"))
```

When 5 trees are generated, it yields a 49% accuracy (41/84).

When 10 trees are generated, it yields a 61% accuracy (51/84).

When 15 trees are generated, it yields a 62% accuracy (52/84).

When 20 trees are generated, it yields a 62% accuracy (52/84).

Fig 14: Random forest algorithm after data optimisation for 5 trees

Fig 15: Random forest algorithm after data optimisation for 10 trees

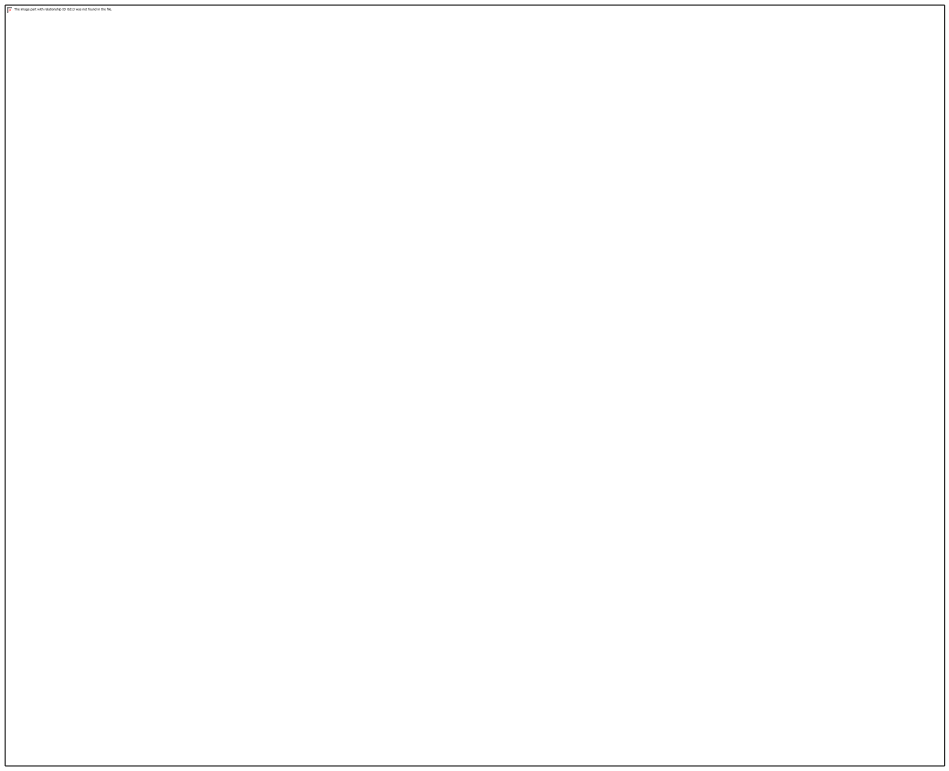


Fig 16: Random forest algorithm after data optimisation for 15 trees

Fig 17: Random forest algorithm after data optimisation for 20 trees

Appendix IX: K-NN algorithm

The algorithm searches the memorized training observations for the *K* instances that most closely resemble the new instance and assigns to it the most common class. In effect, the variable *K* act as a hyperparameter as the K-NN algorithm makes no assumptions to the dataset initially, as opposed to Nearest Neighbours Algorithm (assumption of simple exponential function) and Regression Algorithm (assumption of linearity of data/ overfitting with polynomial functions). The value of *K* can be varied based on the dataset applied to to find out which value of *K* yields the highest accuracy. When *K* is small, the region of a given prediction is restrained, forcing the classifier of the algorithm to less consider the overall distribution of the dataset. A small value for *K* provides the most flexible fit, which will have low bias but high variance. On the other hand, a higher *K*

averages more voters in each prediction and hence is more resilient to anomaly. Larger values of *K* will have smoother decision boundaries which means lower variance but increased bias (Fig 18).

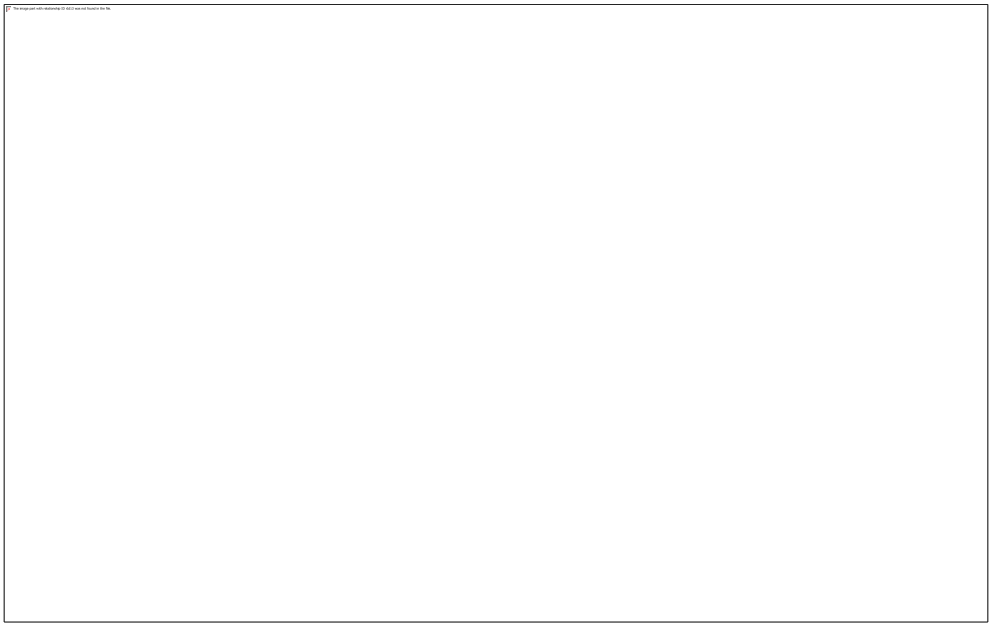


Fig 18: Graph of Error against Model Complexity

In order to maximise accuracy, the value of *K* must be varied and scaled upon to determine the value of *K* that will result in the lowest validation error. In context, inputs covering different magnitudes will decrease the accuracy of the program since the Euclidean distance formula will be largely calculated on the input with the greatest magnitude, downplaying the influence of other inputs with smaller magnitudes. Hence, the scaling function will be used (adjusts variables so that the mean is 0 and the standard deviation is 1) to optimise its accuracy as much as possible.

Appendix X: Results of the K-NN algorithm

The following R program was ran:

```
>library(class)

d <- read.csv("E:/data/cbd_dataset.csv")
set.seed(n)
test <- d[1:756,]
train <- d[757:1512,]


train_knn <- train

test_knn <- test


train_knn <- scale(train_knn)

test_knn <- scale(test_knn)


knn_prediction <- knn(train = train_knn,

                      test = test_knn,

                      cl = train$TrafficConditions,

                      k = x)

knn_prediction == test$TrafficConditions
score = sum(knn_prediction == test$TrafficConditions)
print(paste("Number of correct predictions =", score, "/ 756"))
```

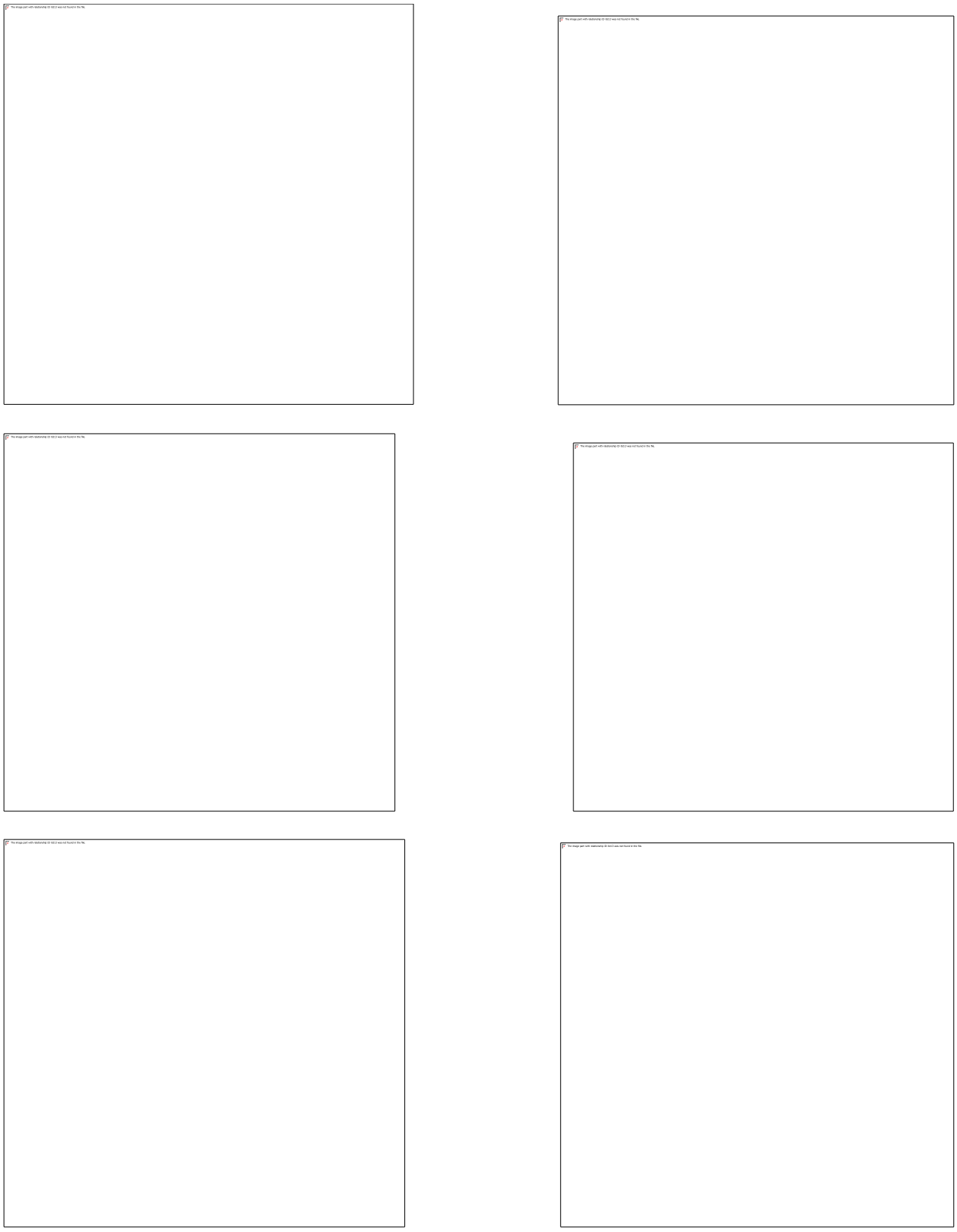


Fig 19: Results of the K-NN algorithm for *k=5, 7, 9, 11, 13* respectively

Appendix XI: Linear Regression algorithm

The following R program was ran:

```
>library(caTools)
library(ggplot2)
data <- read.csv("E:/data/cbd_dataset.csv")
set.seed(15)
split <- sample.split(data,SplitRatio = y)
training <- subset(data,split =="TRUE")
testing <- subset(data,split =="FALSE")

model <- lm(TrafficConditions~.,data)
summary(model)
lm_prediction <- predict(model,testing,type="response")
table(testing$TrafficConditions, PredictedValue=lm_prediction>x)
```

where *x* represents the intercept obtained from the regression analysis, which tells us the threshold where it is considered to be a different traffic condition.

In the algorithm, the calculated value of *x* is 2.39 and *y* represents the data split. Based on the random string of numbers obtained from set.seed(), a random number generator (RNG) takes some data and labels it as training points, while the remaining points are test points, and *y* represents the amount of time the random number generator will run. This RNG allows the data to be extracted randomly to ensure that the linear regression algorithm learns the correlation amongst variables and the set.seed function is used to ensure reproducibility of results.

For $y = 0.5$ and set.seed(15), training size = 602 and test points = 910.

Fig 20: Linear Regression for $y = 0.5$ and set.seed(15) and its corresponding results

For $y = 0.7$ and set.seed(15), training points = 908 and test points = 604,

Fig 21: Linear Regression for $y = 0.7$ and set.seed(15) and its corresponding results

Tabulation of accuracy of results for *y = 0.7* and set.seed(15)

Actual Value	Predicted Value	
	FALSE	TRUE
1	21	20
2	97	148
3	70	150
4	5	93

For *y = 0.9* and set.seed(15), training points = 1210 and test points = 302,

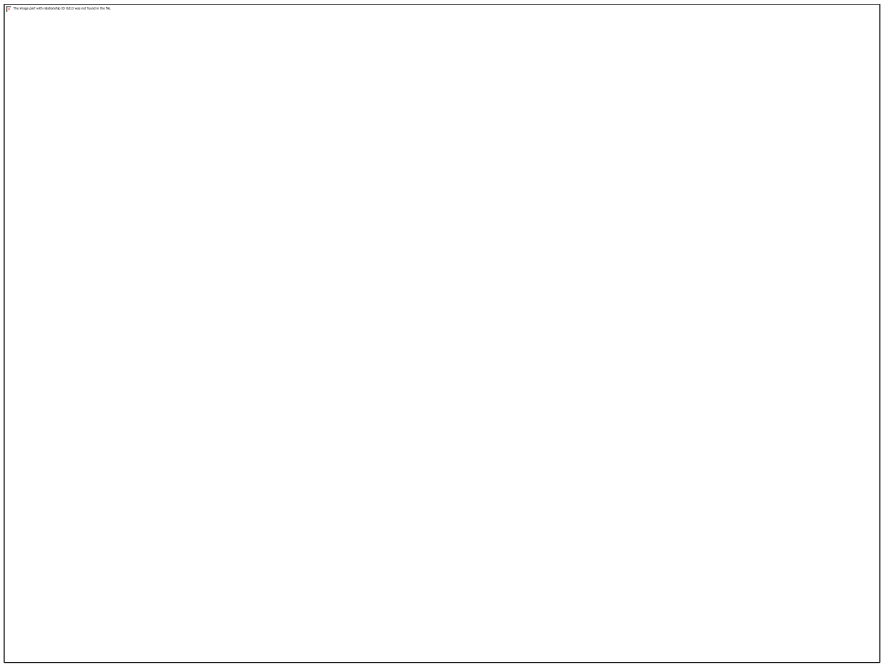


Fig 22: Linear Regression for *y = 0.9* and set.seed(15) and its corresponding results

Tabulation of accuracy of results for *y = 0.9* and set.seed(15)

Actual Value	Predicted Value	
	FALSE	TRUE
1	13	10
2	49	81
3	32	65
4	2	50

Further observations made:

Of all traffic conditions, the algorithm is most accurate at predicting traffic condition 4 (>60kmh), at 94.78%, 94.90% and 96.15% for *y = 0.5, 0.7* and *0.9* respectively. The algorithm is also equally adept at predicting traffic conditions 2 and 3. It is the least accurate at predicting traffic condition 1.