Case Study 1

Predicting YourCab Taxicab Cancellations in Bangalore, India

By Khin Thu Zar Thant

Webster University George Herbert Walker School of Business and Technology

CSDA 6010 Analytics Practicum

16th December 2022

Executive Summary

This case study is aimed to create predicting models in order to identify why drivers cancel the bookings. Due to the imbalanced dataset, the case will contrast the difference between the models built from both balanced and imbalanced dataset. Align to this purpose, there will be many data frames created, with annotations assigned for their respective purpose, where data frames with b represents the balanced dataset and o represents the original imbalanced dataset. These will then be compared and contrasted to determine which dataset has better performance and elaboration on theses will be made throughout the paper, proceeding to the discussion on the final model selection to identify the best model to predict taxi cancellations.

The general breakdown for the paper will be as follows. First part is the introduction of the case identifying the business problem, project goals and the dataset briefing. This section will be proceeded by the data preparation and analysis section, where the data preparation, attribute manipulation, general attribute analysis and dimension reduction takes place. Data preparation is conducted by preprocessing the data dealing missing and null values, and attribute manipulation involves initially removing totally irrelevant data, altering the data format, adding necessary dummy variables and creating new columns which are deemed to be meaningful to the data and finally dealing with missing values. General Attribute Analysis is done by analyzing the patterns in the data using visualization and plotting meaning graphs against the target variables, then the dimension reduction is done using the logistic regression by only retaining the significant and relevant variables in the model. This data preparation and analysis section is proceeded by models' selection section, followed by model evaluation against the balanced and imbalanced dataset as well as the models against one another in order to select the most "predictive" model for YourCab.

Table of Contents

List of Figures

List of Tables

1. Introduction

In the past few decades, taxi service like Uber and Lyft has been dominating the ride sharing market which becomes a threat to the traditional taxi services like YourCab. The business models that have been adopted by the new services are said to be very profitable and efficient in terms of their functionality and the ease of access of their booking system, which threatens YourCab which systems are comparably not kept up with the current trend. They will need to produce a more effective and efficient system in order for them to remain profitable and not going out of business.

Being able to predict if the taxi bookings would be cancelled based on the different circumstances could be a great advantage for YourCab so that they could make certain adjustments and prepare better for cases of cancellations. Cancellations do cost firm money since not only the revenue from these cancelled trips, but also in most cases where the competitors have taken the cancelled orders, firm lost more than just financially – loyal customers. In fact, in 2013, the company noticed a significant problem with the taxi cancellations where drivers did not show up to rides for which the customers have scheduled.

1.1 Business Problem

YourCab, despite having their online booking platform, customers still could call in for bookings and this accounts for the majority of their bookings. For this reason, when drivers cancel the bookings, they do not know what has caused the cancellations and what could be done in order to minimize these situations. Drivers cancelling the bookings has not only make the firm earning less revenue, but also leaving the customers with no taxi when they have booked for one at this particular time or have scheduled them in advance in case of sudden unavailability.

1.2 Project Goal

In order to understand the situation better and assist in predicting whether or not a ride will be cancelled by the driver, a predictive classification model will be developed. This will be achieved using K Nearest Neighbors (KNN), Logistic Regression, Classification Tree and Neural Network models. In addition to that, the data is to be explored to possibly identify the potential causes for the drivers to cancel the rides and attempt to provide insights into this issue.

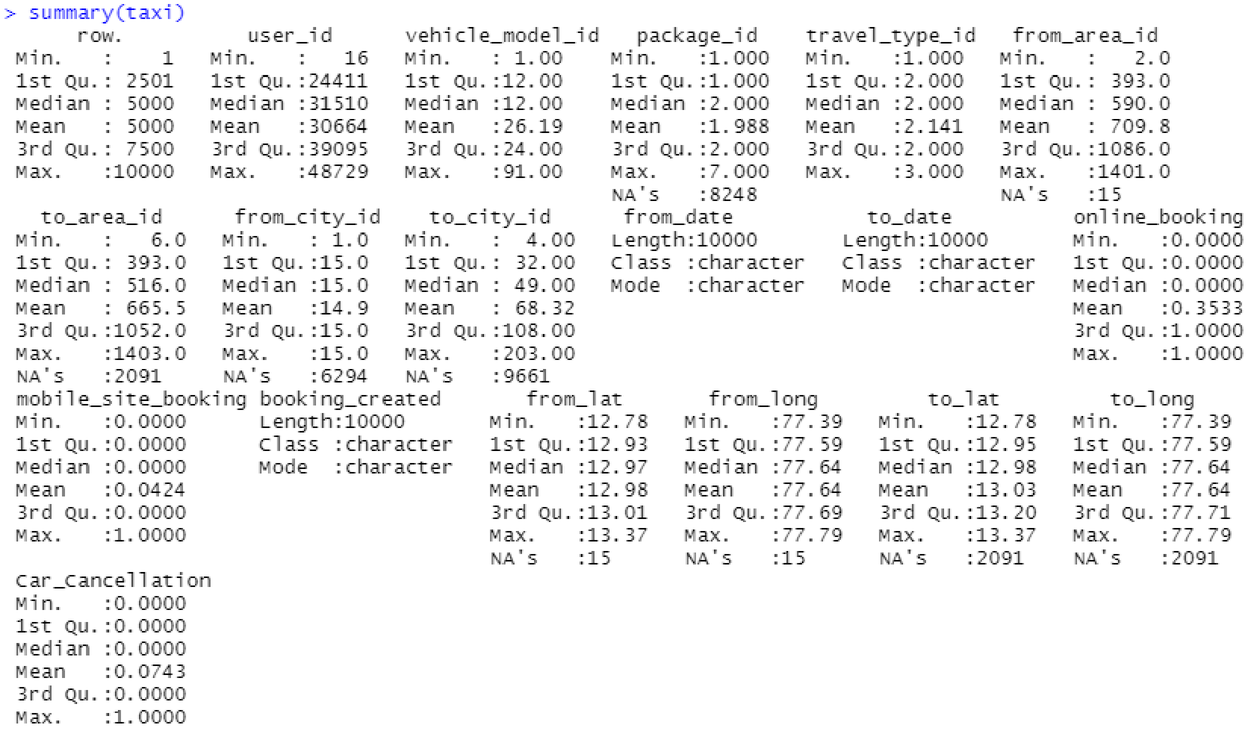
1.3 Data set briefings

The dataset contains a total of 10,000 observations with 19 variables. The variables description are as follows in Table 1. It is proceeded with the summary of the dataset.

Variable Names	Description
Row_ID	This is the unique identifier of this dataset. It is a number identifying each record.
User_ID	This is the identifier of each customer. There are many duplicates in this subset, meaning there are any customers who have called multiple rides.
Vehicle_Model_ID	This is an ID that represents the type of vehicle driven for each ride.
Package_ID	This is an ID that represents the type of travel package, with the following descriptions: 1=4hrs & 40kms, 2=8hrs & 80kms, 3=6hrs & 60kms, 4= 10hrs & 100kms, 5=5hrs & 50kms, 6=3hrs & 30kms, 7=12hrs & 120kms.
Travel_Type_ID	This is an ID that represents the type of travel (1= long distance, 2= point to point, 3= hourly rental).
From_Area_ID	This is an identifier of the starting area. Available only for point-to-point travel.

Variable Names	**Description**
To_Area_ID	This is an identifier of the ending area. Available only for point-to-point travel.
From_City_ID	Unique identifier of the starting city.
To_City_ID	Unique identifier of the ending city.
From_date	Date and time of the requested trip start.
To_date	Time stamp of trip end.
Online_booking	A binary (0,1) variable representing whether the booking was made online or not. 0 represents no, 1 represents yes.
Mobile_Site_Booking	A binary (0,1) variable representing whether the booking was made on their mobile site or not. 0 represents no, 1 represents yes.
Booking_Created	Date and time of booking created.
From_Lat	The latitude of the start area.
From_Long	The longtitude of the start area.
To_Lat	The latitude of the end area.
To_Long	The longtitude of the end area.
Car_Cancellation	The target variable. A binary (0,1) variable representing whether or not the ride was cancelled. 0 means no, 1 means yes.

Table 1 *Variable Description for the Dataset*



```
> summary(taxi)
      row.          user_id      vehicle_model_id   package_id    travel_type_id   from_area_id  
 Min.   :    1   Min.   :   16   Min.   : 1.00    Min.   :1.000   Min.   :1.000   Min.   :   2.0  
 1st Qu.: 2501   1st Qu.:24411   1st Qu.:12.00    1st Qu.:1.000   1st Qu.:2.000   1st Qu.: 393.0  
 Median : 5000   Median :31510   Median :12.00    Median :2.000   Median :2.000   Median : 590.0  
 Mean   : 5000   Mean   :30664   Mean   :26.19    Mean   :1.988   Mean   :2.141   Mean   : 709.8  
 3rd Qu.: 7500   3rd Qu.:39095   3rd Qu.:24.00    3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:1086.0  
 Max.   :10000   Max.   :48729   Max.   :91.00    Max.   :7.000   Max.   :3.000   Max.   :1401.0  
                                                  NA's   :8248                    NA's   :15      
   to_area_id      from_city_id     to_city_id      from_date            to_date          online_booking  
 Min.   :   6.0   Min.   : 1.0    Min.   :  4.00   Length:10000       Length:10000       Min.   :0.0000  
 1st Qu.: 393.0   1st Qu.:15.0    1st Qu.: 32.00   Class :character   Class :character   1st Qu.:0.0000  
 Median : 516.0   Median :15.0    Median : 49.00   Mode  :character   Mode  :character   Median :0.0000  
 Mean   : 665.5   Mean   :14.9    Mean   : 68.32                                         Mean   :0.3533  
 3rd Qu.:1052.0   3rd Qu.:15.0    3rd Qu.:108.00                                         3rd Qu.:1.0000  
 Max.   :1403.0   Max.   :15.0    Max.   :203.00                                         Max.   :1.0000  
 NA's   :2091     NA's   :6294    NA's   :9661                                                            
 mobile_site_booking booking_created       from_lat       from_long         to_lat          to_long     
 Min.   :0.0000      Length:10000       Min.   :12.78   Min.   :77.39   Min.   :12.78   Min.   :77.39  
 1st Qu.:0.0000      Class :character   1st Qu.:12.93   1st Qu.:77.59   1st Qu.:12.95   1st Qu.:77.59  
 Median :0.0000      Mode  :character   Median :12.97   Median :77.64   Median :12.98   Median :77.64  
 Mean   :0.0424                         Mean   :12.98   Mean   :77.64   Mean   :13.03   Mean   :77.64  
 3rd Qu.:0.0000                         3rd Qu.:13.01   3rd Qu.:77.69   3rd Qu.:13.20   3rd Qu.:77.71  
 Max.   :1.0000                         Max.   :13.37   Max.   :77.79   Max.   :13.37   Max.   :77.79  
                                        NA's   :15      NA's   :15      NA's   :2091    NA's   :2091    
 Car_Cancellation
 Min.   :0.0000  
 1st Qu.:0.0000  
 Median :0.0000  
 Mean   :0.0743  
 3rd Qu.:0.0000  
 Max.   :1.0000  
```

Figure 1 *Summary of the Dataset*

2. Data Preparation

2.1 Data Preprocessing

Firstly, the Car Cancellation dataset containing 10,000 observations and 19 variables is loaded into R. The data is then attached to ensure better reference in the latter section without calling on the data name repeatedly. After that, the internal structure of the data of each variable and the summary statistics (i.e., minimum, first quartile, median, mean, third quartile, maximum and NA values) are observed respectively. These functions could give a better overview of how the data looks like and brings an insight into how the data should be cleaned and wrangled.

Next, the data is preprocessed to check for missing values. Since this is a big dataset, extra caution has to be made in detecting missing values and hence, several additional functions have been undertaken. The missing values in each column are located, as well as visualizing the missing data. In large datasets, visualizations could be a better idea for analysts to visualize the structure of the data. Figure 1 below shows how the variables are sorted, starting with the to_city_id having the most NA values, followed by package_id, from_city_id, to_area_id, to_lat and to_long_id. Dealing with the missing values will be done later, if necessary, since the variables which have high missing values are dimensions that are irrelevant to the prediction model that we are creating. A breakdown table has also been illustrated below.

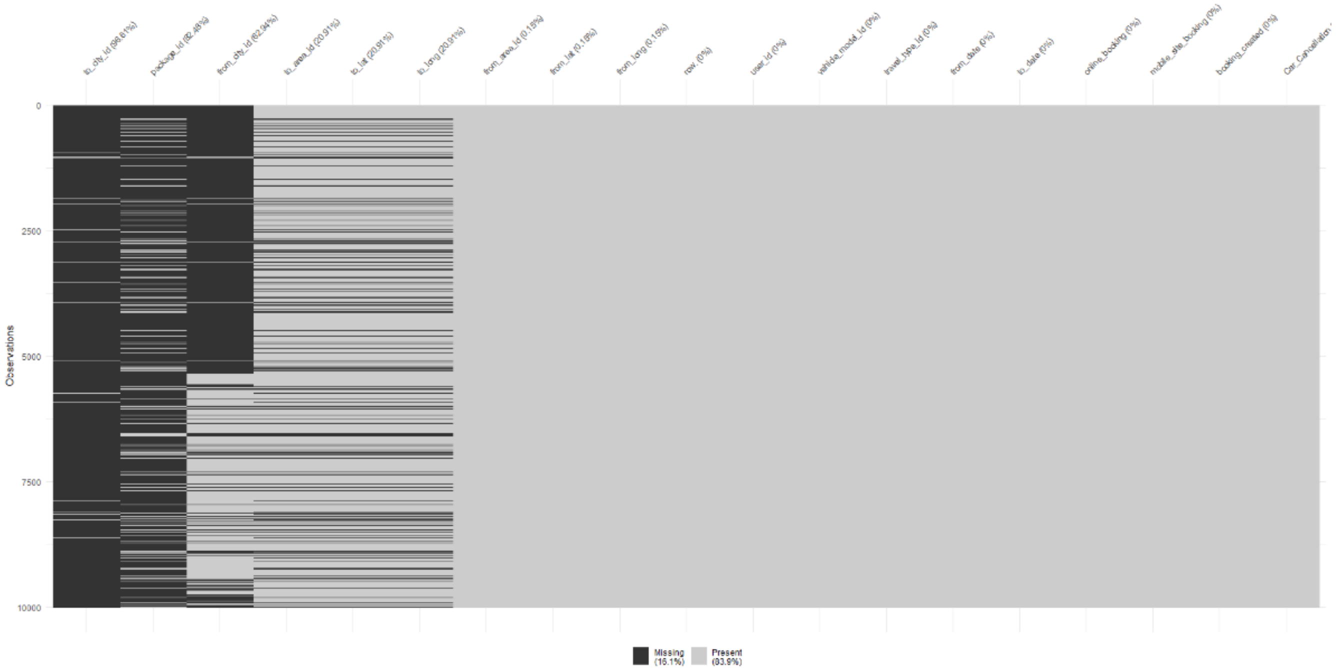


***Figure 2** Missing Data Visualization*

Variable	n_miss	pct_miss
to_city_id	9661	96.61
package_id	8248	82.48
from_city_id	6294	62.94
to_area_id	2091	20.91
to_lat	2091	20.91
to_long	2091	20.91
from_area_id	15	0.15
from_lat	15	0.15
from_long	15	0.15
row	0	0
user_id	0	0
vehicle_model_id	0	0
travel_type_id	0	0
from_date	0	0
to_date	0	0
online_booking	0	0
mobile_site_booking	0	0
booking_created	0	0
Car_Cancellation	0	0

Table 2 *Missing Value Breakdown in Each Variable*

After that, null values are checked, which returns a FALSE output, and this indicates that there are no null values in the dataset. For a safer measure, the function is run again to access the individual data to better make sure if the null values exist and all functions returned FALSE as well. Therefore, no measures are taken to deal with null value.

2.2 Attribute Manipulation

2.2.1 Removing irrelevant variables

Variable “Row” is removed at once since it is just stating the number of the row which is irrelevant to the prediction model developed. The City ID columns (From City_ID and To_City_ID) have also been removed due to its duplication of representation with the Area columns (From Area_ID and To_Area_ID). With that said, as a data analyst, it is important to make sure that there is little or no multicollinearity, which means to reduce the closely predictors from the model. In layperson terms, when two things mean the same to a target, then removing one of them would not affect the target outcome and in fact, reduce the redundancy.

2.2.2 Altering the Date columns

All the date-time format has to be changed to date and time format/%D%Y %H:%M" as a time-series object in order to contain what is in the column itself; the only thing is the change of format instead of the change of data value, so it has to made sure. It is then used to divide into two separate columns – “from_date_date” and “from_date_time.” The same thing is done to the booking_created column, where it is first converted to a time- series object and then separate the column. The same thing is done for the booking_created columns.

2.2.3 Additional Column Variables Created

Apart from the four new columns created for the date time separation above, there are three new columns created before wrapping up the variable interpretation section. The first column created is the “gap time” column which is defined by the difference between the time taxi is booked and the actual time when the requested trip started. This is to determine the time difference between these two actions since it could be a direct attribute for why the drivers cancel the trip and could potentially affect the cancellation probability. This is done based on the assumption that the longer the gap time, the higher the probability that the drivers cancel the order, which could be due do them not being able to make it back from the previous trip.

Another variable created is the “Call- In Booking.” The two booking methods that are in the dataset are online-booking and the mobile site booking, presuming that if both variables show 0, then the only booking method left is when the customer actually call in to book for their cab. This variable is added on as another predictor, which could be an important attribute that causes the cancellation and thus this shall be classified.

The last variable created is the “Trip_Distance” variable which is the distance calculated using the GPS data. A function is used to create this distance calculation where the latitude and longitude of two points are inserted to return a new variable “Trip_Distance”. This is to measure the distance for each trip since it could be a causing factor for drivers to cancel the trip due to long distances or incompatible distance range with their working hours, etc., and the results are in kilometers.

2.3 General Attribute Analysis Against the Target Variable and Visualizations

After the interpretation and the necessary alterations of the attributes, it is also important to have a summary of each identifier attribute before jumping to the data partitioning and developing the models.

2.3.1 General Histogram and Bar Graph & Box plot

Histogram and bar graphs are created to provide a brief overview on the distribution of the data in each category and range of the variable, where the histogram is used for numerical variables and the bar graph is used for the categorical variables. In this section, date and time variables are excluded since they are non-numeric and categorical, and it would not be relevant to plot them along with the other variables. These variables are to be explored using different methods. Figure 2 on the next page illustrates a combined plots for the variables.

In addition to that, Figure 3 illustrates the combination of the box plot for all the numeric data in order to check the scale of the mean and quartile ranges as well as giving an overview for the potential outliers.

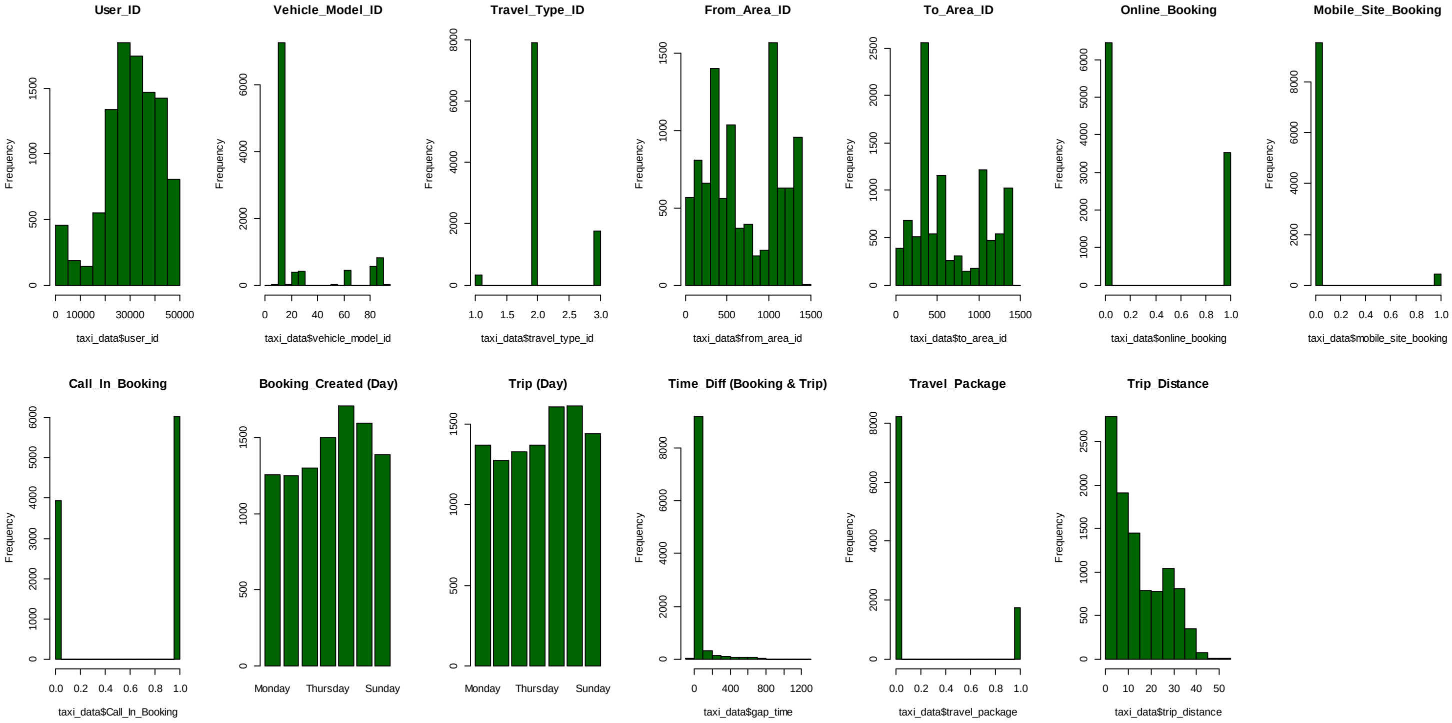


__Figure 3__ Combined histogram/bar plot for variables

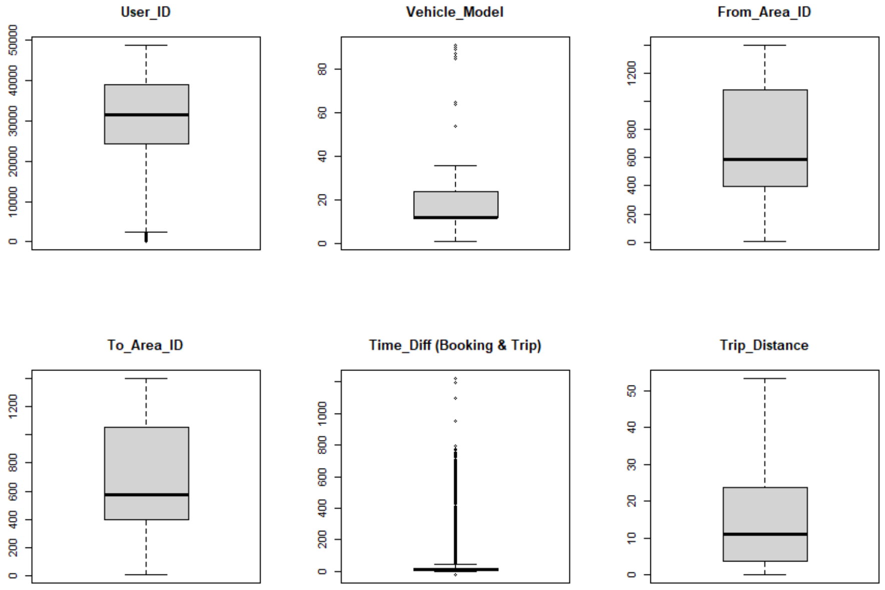


Figure 4 *Combined Box plot for numeric variables*

2.3.2 Class Distribution

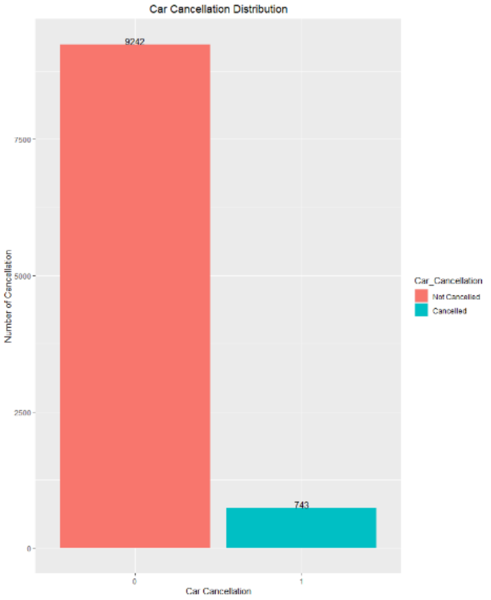


Figure 5 *Bar Graph for Class Distribution*

The outcome variable, Car Cancellation is first used to see the proportion of each class. It can be seen that it only has two outcome which are binary variable of 1 and 0, where 1 represents that the taxi was cancelled and 0 represents that the taxi was not cancelled (trip completed). Figure 2 on the right shows the breakdown of the two outcomes. From there, we can see that the 0 class outnumbered the 1 class, where it occupies approximately 90% of the whole dataset. In fact, out of the 10,000 observations, 9,257 This has to be carefully handled when it comes to data partitioning in the latter section to maintain equal proportions of the class differences is there for all partitions (i.e. approximately 90% of class 0 and approximately 10%

of class 1 in all data partitions). This will be discussed more in the *Dealing with Imbalance Dataset* section.

2.3.3 Cancellation Rate by User ID

Considering that the customer could be the factor related to the drivers cancelling the bookings, summarization of the statistics has been made to check how the data is describing such trend. For this analysis, the data has been subset and summarized with each user_id with their own respective cancellation counts. Since the number of users_id is huge, the data is then filtered to just those with more than 20 cancellations, which is 2% of the total observations in the data, leaving just 14 user IDs remaining with all of them having more than 20 cancellations. The user_id is also converted to categorical variable in order to produce a presentable bar graph to show the cancellations for each user. Figure 5 shows the bar graph representation of the case situation, and it can be seen that customer with user ID of 29648 standing out in the graph with 91 rides in total and the highest cancellation among all with 14 out of 91 (which is equivalent to 15% cancellation). Overall, it can be seen that the cancellation rates compared against their respective user IDs are not that high to a point to argue that it is indeed the users that led to drivers cancelling the rides. In fact, even the highest cancellation rate observed is just 15% indicating that this percentage of cancellation is led by some other factors, not the user ID and it will not make sense realistically to say that the drivers selectively choose the passengers.

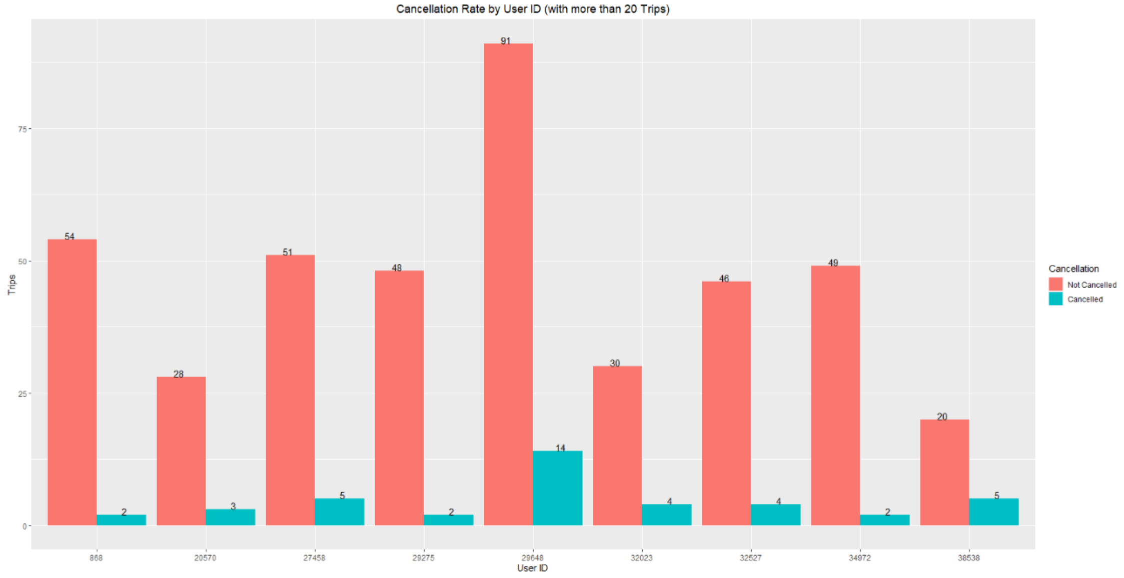


***Figure 6** Cancellation Rate by User IDs*

2.3.4 Cancellation Rate by Vehicle Model

Vehicle Model could be another attribute to why drivers cancel the rides, taking into the possibilities that the driver could not transport certain group of customers due to its vehicle capability or that certain vehicle could only hold certain number of trips per day, etc. As well as that, another important reason to present this scenario is to visualize which vehicles cancel the bookings and their respective number of cancellations and track the drivers afterwards. Therefore, this variable is also used for summarization of these possibility. Similarly, the data has been subset and summarized by the vehicle models with the cancellation of bookings in order to produce a cancellation rate per vehicle model. The result is also filtered with only vehicles with more than 100 rides in order to selectively present the problematic vehicles instead of presenting with all the vehicles, which might contain other insignificant instances. Figure 6 shows the representation of what the breakdown would be like. From the graph, it can be seen that drivers with 89 vehicle model ID has a cancellation rate of 12.86%, followed by ID number 12 and 28 with 8.47% and 5.43% respectively.

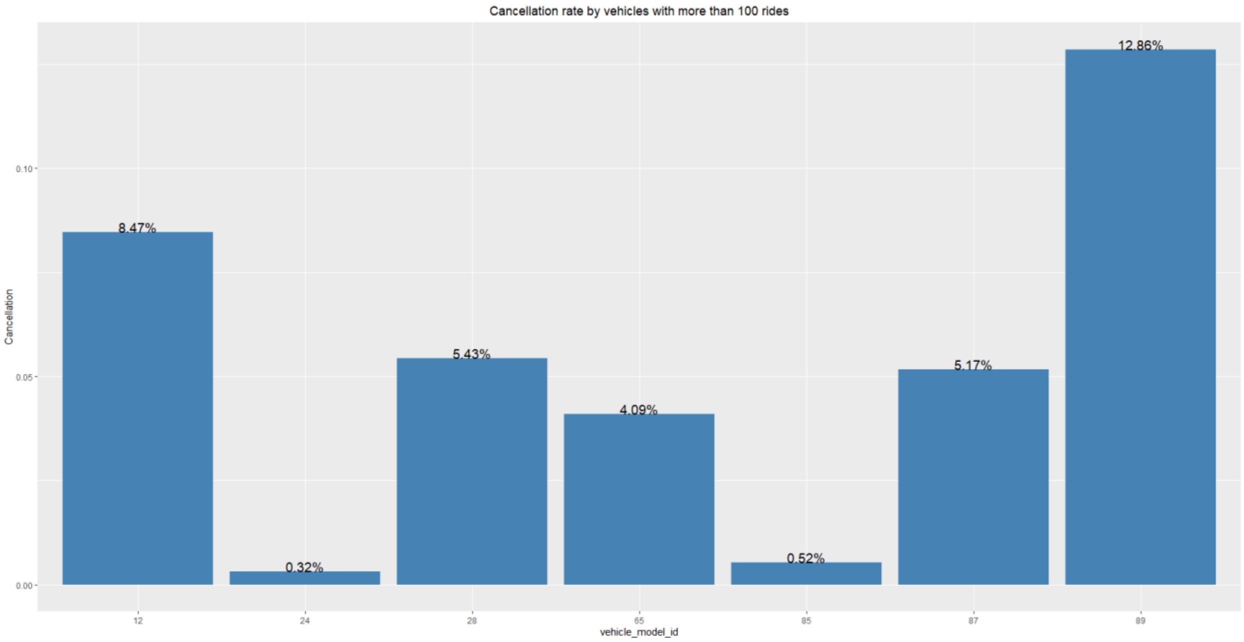


Figure 7 *Cancellation Rate by Vehicle Model ID with more than 100 rides*

2.3.5 *Cancellation Rate by other attributes: Travel Type, Methods of Booking and Day of the Week*

Similar assumptions are also made to other classifier variables- travel type, method of booking and the day of the week. Data has been subset with their respective variables and bar graph has been plotted. The figures below show the breakdown of each category cancellation rate.

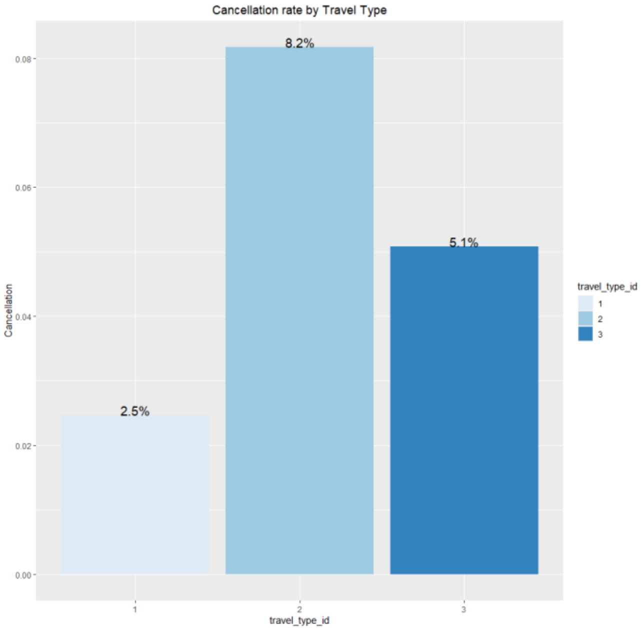


Figure 9 Cancellation Rate by Travel Type

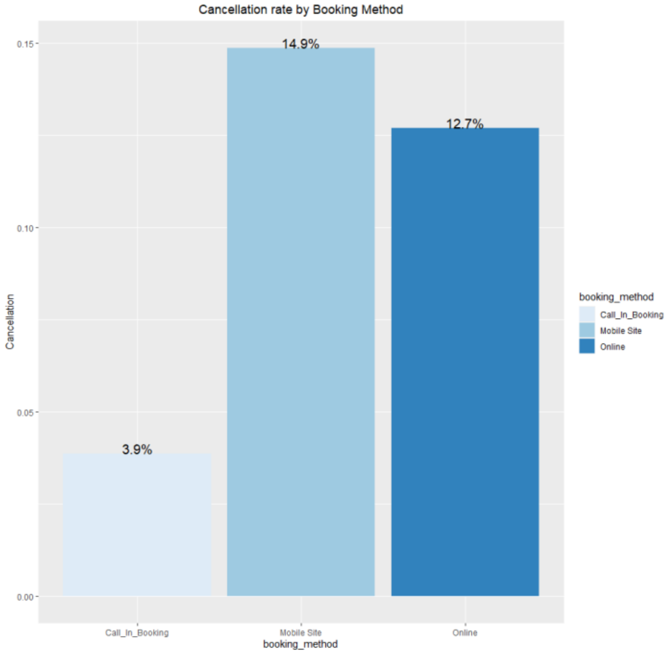


Figure 8 Cancellation Rate by Booking Method

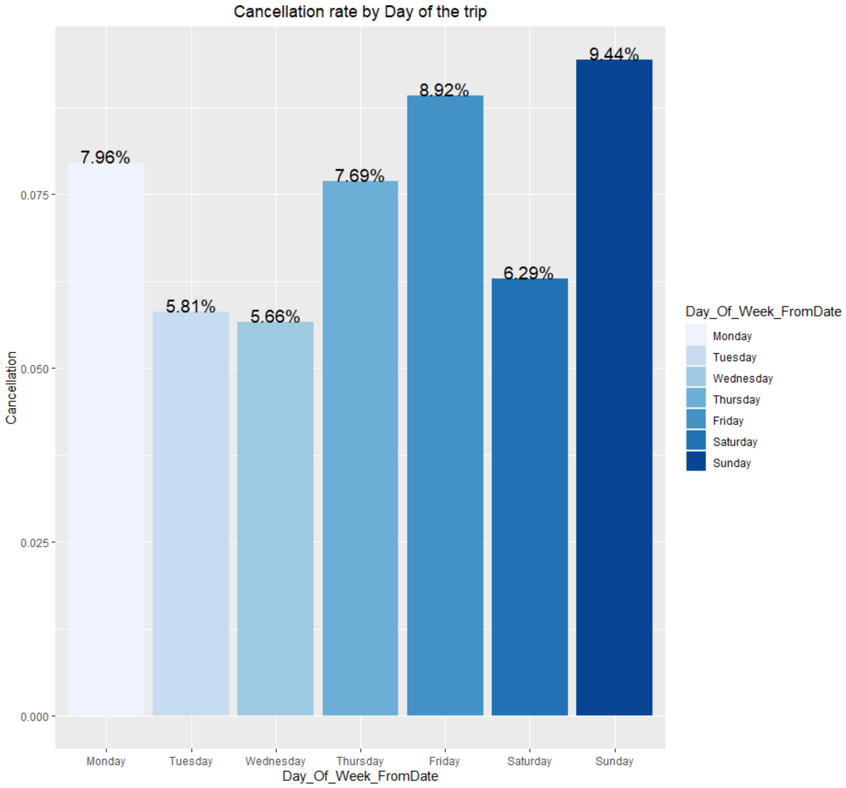


Figure 10 Cancellation Rate by Day of Trip

Based on the above illustrations, the following comments could be made. Firstly, looking at Figure 7 for cancellation rate by travel type, it can be seen that Type 2 (point to point travel) has the highest cancellation rate, followed by Type 3 which is the hourly rental and Type 1 which is the long distance. It looks reasonable since point-to-point travel is the most frequent travels among all, which makes it higher probability of cancellation. Next in Figure 8 for cancellation rate by booking method, surprisingly, mobile site booking has the highest cancellation rate, which is followed by online booking and call-in booking. Call- In booking accounts for most of the observations and yet it came in last for cancellations, so there might be more reasons behind the cancellations, which could be the online and mobile site ineffective site. As for the Figure 9 which is the cancellation rate by the weekday, it shows that Sunday, Friday, Monday and Thursday have the highest cancellation rate, with the rest of the week with similar rates.

2.4 Dealing with missing numbers

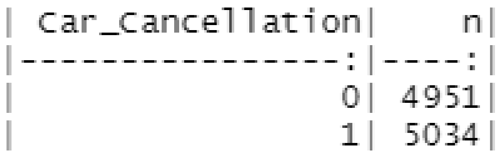
At some point with the original data, after removing the irrelevant variables, creating necessary columns (like date/time, gap time and distance columns) and finally explore the variables using graph visualization, it is now important to address the missing value before moving on to the next stage of balancing the data. The summary of the missing data has confirmed the missing values in these columns: to_area_id, to_lat, to_long, from_area_id, from_lat and from_long. Since the latter three has fewer missing values of 15 values each, it is dealt with first. With all three of them being the location related observations, it is assumed that in these observations, most of the other location related variables also have the missing values, and therefore, those rows with missing values in these four variables and eventually, these columns no longer have the missing value.

Next to deal with is the to_area_id which has over 2000 missing values, which accounts for more than one-fifth of the total observations. In this case, it is assumed that the missing value is resulted from being in the same Area ID from where the trip start, or in other words, they could be short trips with the same Area ID all along from where the trip starts until its destination. Hence, the missing values are replaced with their corresponding from_area_id to address this notion and to do this, the function is called to mutate these two variables, and this eliminates the missing values in this column. Similar denotation is made on the to_lat and to_long data since their nature of being the location related is similar so such denotation is also relevant to them.

The summary of the missing value is called again and there were no more missing values, and it is set to move on to the other step: Balancing the data.

2.5 Balancing the data

This balancing of the data does not intend to interfere with the partition of the data later, but it is to make the data balanced and unbiased before dimension reduction, which variables to be removed are determined by the logistic regression model which will be done at the latter section. This section will be about the balancing of the data, with the method used to balance this data is using the "BOTH," which is both under sampling and oversampling. Under sampling alone could make significant majority class (Class 1) observations to be removed and likewise oversampling alone could make too much manipulation on the unoriginal minority class (Class 0) observations to be added. Therefore, to stand a middle ground, the method "BOTH" is selected. The figure below shows the breakdown of the two classes after balancing the data. The difference between them is only less than 100 records and this time there are more Class 1 records.

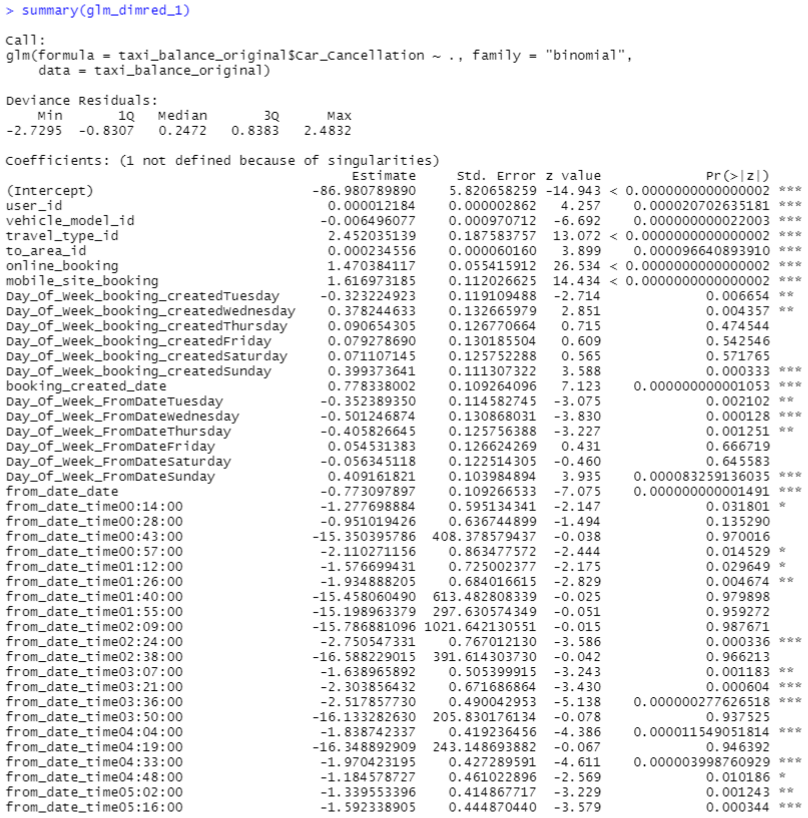


```
| Car_Cancellation|    n|
|----------------:|----:|
|                0| 4951|
|                1| 5034|
```

***Figure 11** Breakdown of the classes after balancing the data*

2.6 Dimension Reduction

The next step here will be reducing the number of variables, which is the Dimension Reduction. In this case, Principal Component Analysis (PCA) is not necessary given that there are only 19 variables in which there are weak relationship among one another. Instead, logistic regression will be done to identify the significant and insignificant variables, where the summary of the model is used to identify them. From the model summary, it can be seen that the from_area_id and booking_created_time are not significant to the target variables, so they will be removed, and the model was re-run. The figures below show a partial screenshot of the model summary (the whole summary will be attached in the **Appendix**). There were some from_date_time entries removed from the screenshot since there is a similar pattern of sometime values being significant and some were not or less significant, so this variable is ought to be kept, which is why the screenshot in between is not shown.



```
> summary(glm_dimred_1)

Call:
glm(formula = taxi_balance_original$Car_Cancellation ~ ., family = "binomial", 
    data = taxi_balance_original)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-2.7295  -0.8307   0.2472   0.8383   2.4832  

Coefficients: (1 not defined because of singularities)
                                          Estimate     Std. Error z value             Pr(>|z|)    
(Intercept)                          -86.980789890    5.820658259 -14.943 < 0.0000000000000002 ***
user_id                                0.000012184    0.000002862   4.257    0.000020702635181 ***
vehicle_model_id                      -0.006496077    0.000970712  -6.692    0.000000000022003 ***
travel_type_id                         2.452035139    0.187583757  13.072 < 0.0000000000000002 ***
to_area_id                             0.000234556    0.000060160   3.899    0.000096640893910 ***
online_booking                         1.470384117    0.055415912  26.534 < 0.0000000000000002 ***
mobile_site_booking                    1.616973185    0.112026625  14.434 < 0.0000000000000002 ***
Day_Of_Week_booking_createdTuesday    -0.323224923    0.119109488  -2.714             0.006654 ** 
Day_Of_Week_booking_createdWednesday   0.378244633    0.132665979   2.851             0.004357 ** 
Day_Of_Week_booking_createdThursday    0.090654305    0.126770664   0.715             0.474544    
Day_Of_Week_booking_createdFriday      0.079278690    0.130185504   0.609             0.542546    
Day_Of_Week_booking_createdSaturday    0.071107145    0.125752288   0.565             0.571765    
Day_Of_Week_booking_createdSunday      0.399373641    0.111307322   3.588             0.000333 ***
booking_created_date                   0.778338002    0.109264096   7.123    0.000000000001053 ***
Day_Of_Week_FromDateTuesday           -0.352389350    0.114582745  -3.075             0.002102 ** 
Day_Of_Week_FromDateWednesday         -0.501246874    0.130868031  -3.830             0.000128 ***
Day_Of_Week_FromDateThursday          -0.405826645    0.125756388  -3.227             0.001251 ** 
Day_Of_Week_FromDateFriday             0.054531383    0.126624269   0.431             0.666719    
Day_Of_Week_FromDateSaturday          -0.056345118    0.122514305  -0.460             0.645583    
Day_Of_Week_FromDateSunday             0.409161821    0.103984894   3.935    0.000083259136035 ***
from_date_date                        -0.773097897    0.109266533  -7.075    0.000000000001491 ***
from_date_time00:14:00                -1.277698884    0.595134341  -2.147             0.031801 *  
from_date_time00:28:00                -0.951019426    0.636744899  -1.494             0.135290    
from_date_time00:43:00               -15.350395786  408.378579437  -0.038             0.970016    
from_date_time00:57:00                -2.110271156    0.863477572  -2.444             0.014529 *  
from_date_time01:12:00                -1.576699431    0.725002377  -2.175             0.029649 *  
from_date_time01:26:00                -1.934888205    0.684016615  -2.829             0.004674 ** 
from_date_time01:40:00               -15.458060490  613.482808339  -0.025             0.979898    
from_date_time01:55:00               -15.198963379  297.630574349  -0.051             0.959272    
from_date_time02:09:00               -15.786881096 1021.642130551  -0.015             0.987671    
from_date_time02:24:00                -2.750547331    0.767012130  -3.586             0.000336 ***
from_date_time02:38:00               -16.588229015  391.614303730  -0.042             0.966213    
from_date_time03:07:00                -1.638965892    0.505399915  -3.243             0.001183 ** 
from_date_time03:21:00                -2.303856432    0.671686864  -3.430             0.000604 ***
from_date_time03:36:00                -2.517857730    0.490042953  -5.138    0.000000277626518 ***
from_date_time03:50:00               -16.133282630  205.830176134  -0.078             0.937525    
from_date_time04:04:00                -1.838742337    0.419236456  -4.386    0.000011549051814 ***
from_date_time04:19:00               -16.348892909  243.148693882  -0.067             0.946392    
from_date_time04:33:00                -1.970423195    0.427289591  -4.611    0.000003998760929 ***
from_date_time04:48:00                -1.184578727    0.461022896  -2.569             0.010186 *  
from_date_time05:02:00                -1.339553396    0.414867717  -3.229             0.001243 ** 
from_date_time05:16:00                -1.592338905    0.444870440  -3.579             0.000344 ***
```

Figure 12 *Screenshot of the Logistic Regression Model*

From the figure above, it can be seen that not only the from_date_time having different values with different significance, but there are also day columns (day_of_week_booking_created and day_of_week_from_date) which have the similar trend. Therefore, as mentioned above about the fact that there are some significance in some of the variable class, the whole column variable is kept. As well as that, it can be seen that the Call_In_Booking is insignificant, but for the sake of the classification model which the dummy variable is required, this variable will be kept.

After re-running the model and finalize which columns are not significant, the original data is called on to remove these variables from the data. The table below shows a summary of what variables are remaining from the data after the dimension reduction process.

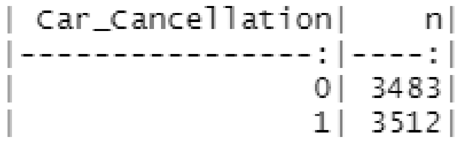
Variables	**Description**
User_ID	Refer to Table 1 for Variable Description
Vehicle_Model_ID	Refer to Table 1 for Variable Description
Travel_Type_ID	Refer to Table 1 for Variable Description
To_Area_ID	Refer to Table 1 for Variable Description
Online_Booking	Refer to Table 1 for Variable Description
Mobile_Site_Booking	Refer to Table 1 for Variable Description
Car_Cancellation	Refer to Table 1 for Variable Description
Day_Of_Week_booking created	Day of the week when the booking take place
Booking_Created_Date	Date when booking was first created
Day_Of_Week_FromDate	Day of the week when the actual trip occured
From_Date_Date	Date of the trip
From_Date_Time	Time of the trip
Gap_Time	Time difference between the Booking_Created and From_Date (in hours)
Call_In_Booking	One of the booking methods
Travel_Package	Adaptation from the Package ID columns, which is changed to binary values
Trip_Distance	Measures the distance of each trip

***Table 3** Summary of the remaining variables in the final dataset*

2.7 Data Partitioning

The resulting data, which is the taxi_data is to be partitioned with 70% for training data and 30% for testing data. [1] In this case, the partition method is static partition since the data is already physically categorized and group and ready to be added as a partition to a table and here, the taxi_data is partitioned using the sample () function where the number of samples and the proportion of each partition has been determined and input.

Since the training data is still not balanced, the balancing will be done using the same method as the previous sampling (which is the method "BOTH"). The number of instances is not specified and instead the number is to be set by the function itself to reduce manipulation and biases. The figure below shows the breakdown and probability of each class in the training data.



```
| Car_Cancellation|    n|
|----------------:|----:|
|                0| 3483|
|                1| 3512|
```

Figure 13 *Balanced Train Data Class Breakdown*

Table 4 below summarizes the list of data frame names allocated for balanced/ imbalanced or train/ test data and provides a summary of what has been mentioned all along in this section on the data used in R for its own specific purpose.

[1] To avoid confusion, the taxi_data here is not the balanced data. The balanced data is taxi_balanced_original which is used for dimension reduction purpose (running the logistic regression) only. Using the outcome from the model, it is applied to the original unbalanced data, this will then be used throughout the rest of the stages with certain modifications which varies.

Data frame names	What the data is for
taxi	The original data which is loaded into R with some initial removal of irrelevant variable
Taxi_balance_original	The balanced data to the original data which is solely used for the purpose of dimension reduction
Taxi_train_imbalanced	The unbalanced train data, which is the 70% of the total imbalanced data; used in models with imbalanced data
Taxi_test	The test data, which is 30% of the original imbalanced data; used in all models with or without balanced data
Taxi_train_balanced	The balanced train data, which is the balanced data from the unbalanced train data (taxi_train_imbalanced); used in models with balanced data

Table 4 *Summarization of the main data frames*

3. Classification Models Selected

Given the purpose of identifying the best model to classify the car cancellations by the drivers, classification models will be used. In this section, the preprocessed and cleaned data will be fed into four classifications machine learning algorithms: K- Nearest Neighbor, Classification Tree, Logistic Regression and Neural Network. Modifications to the train and test data has been made respectively to tailor to the effectiveness of each model, and all the models are evaluated using the confusion matrix (for model accuracy, sensitivity and specificity measures), where the important class is set to “1” in order to prioritize the class 1 which is the booking being cancelled and the Receiver Operating Characteristics (ROC) curve (for accessing the performance through classification thresholds). The randomization seed is set to 12345 in order to ensure consistent results for all models as well as their performance.

3.1 K- Nearest Neighbor

The first model selected is the most classic classification algorithm which assumes that the data points which distances are close to one another should belong to the same class and predictions to the new data are made based on this assumption. A series of action have been taken in order to fit a KNN model.

3.1.1 Model Mechanism

In order to perform KNN, first all the irrelevant variables are to be removed. In this case, non- numeric variables like the date, time and week of the day variables are removed. The remaining variables value are also normalized in order to put the scale between 0 and 1 to optimize the model performance. In the codes, the proportion class are checked from time to time in order to ensure that the data frame used are correct for each designated section. Both KNN models are set with an initial random value of 7, which are later tuned for better performance. All other alternative k is also used to build models with the desirable sensitivity rate.

3.2 Classification Tree

Classification tree which is sometimes referred to as decision tree is a structural mapping of the categorical decisions of the data based on the predictors' rules and splits based on the purity of the data. It is yet another popular machine learning algorithm which the target variable comes to take a set of values with all probability of class stated in each node and known for classifying the new data based on a set of decision rules which represents the relationship exist in the data.

3.2.1 Model Mechanism

Before fitting a tree, first all categorical variables are converted to categorical and using the respective data frame name (balanced or imbalanced/ train or test), the model is fitted using the rpart, which is then plotted. The tree is first used to predict the training data itself, and then used in testing data. In addition, it is also pruned using the best cp value with the least xerror vale. All predicted values are evaluated using the confusion matrix.

3.2.2 Variable Handling: Date and Time Variable

Apart from that, there are a few more adjustments made to the classification tree model, is the handling of the date and time data. First, speaking time, since there are varying data in the data, there would be a great mess when plotting the tree and especially when the tree model algorithm splits the time variable, it would become unmeaningful split. For instance, the split could be if time = 01:00:00, 02:00:00, 03:00:00 and so on, since it is too exact about the split, it is hard for analysts and the end users to actually interpret the tree. Therefore, instead, it is used to create the time period which is split into 5 period of the day: Midnight, Morning, Afternoon, Evening and Night. Since the time formats are in 24-hour clock format, it is easier to determine the division of the time period using the time difference from the midnight 12 a.m., which is 0. The diagram below shows the breakdown of the time along with the 12-hour clock format for better understanding. In the R code, the variable was renamed as “trip period,” and the alteration is done to the from_date_time variable which is the time when the trip actually takes place.

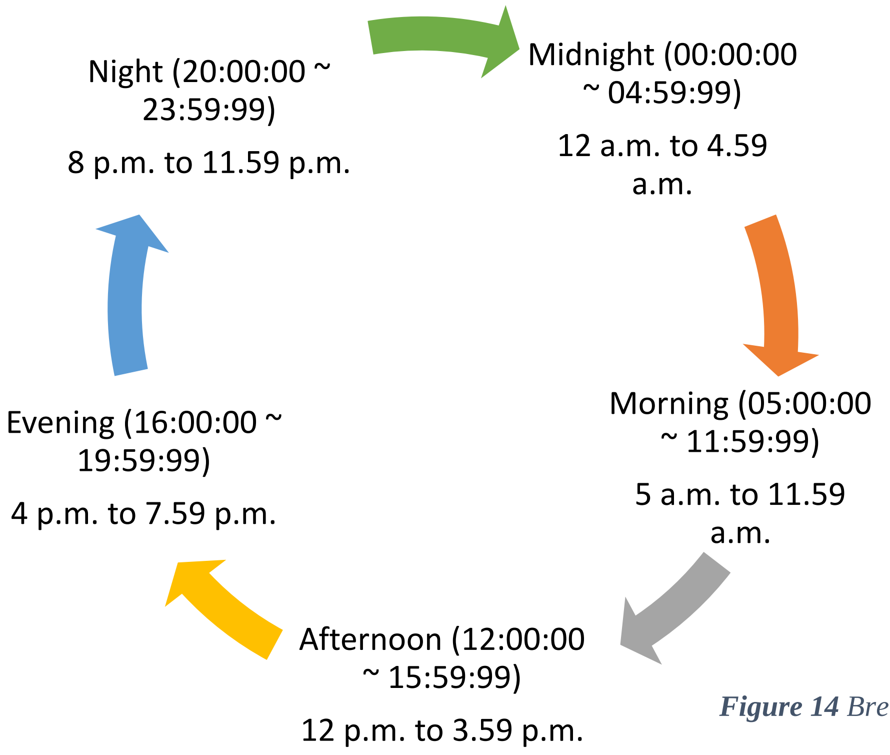

Night (20:00:00 ~ 23:59:99)
8 p.m. to 11.59 p.m.
Midnight (00:00:00 ~ 04:59:99)
12 a.m. to 4.59 a.m.
Morning (05:00:00 ~ 11:59:99)
5 a.m. to 11.59 a.m.
Afternoon (12:00:00 ~ 15:59:99)
12 p.m. to 3.59 p.m.
Evening (16:00:00 ~ 19:59:99)
4 p.m. to 7.59 p.m.

***Figure 14** Breakdown of Time Period for Time Variables*

Next thing is the date variable, which is in the format of YYYY/MM/DD. Similar problem to the time variables, the date variable could bring in unreasonable interpretation on the date that could not be understood by the end users. In fact, it will produce numeric values with scientific notation, for instance booking_created_date = 16e +3. Therefore, in order to prevent this, the date variable is to be divided into three new columns, specifically "Year," "Month," "Day", and among that only the latter two will be used. The "Year" variable will be removed since it is not significant for it to be there with reasons being that firstly, the data is all in 2013 and the next reason also the most important reason is that all new data will be used for the model which is not going to be 2013 since the model is intended to predict the future data.

3.3 Logistic Regression

The next model used for this case is the logistic regression model, which is another predictive algorithm which outcome variable is predicted based on the relationship between the predictors. The predicted outcome is in probability, which is then converted to the categorical class based on the cutoff value set where the value greater than one cutoff value is allocated as a certain class.

3.3.1 Model Mechanism

In order to fit the logistic regression model, first categorical variables are first converted to categorical, and the model is built upon that. The options (scipen = 999) are used to avoid all scientific notations forcing full display of the numbers for better interpretation to the model. Similar to the previous models, it is first tested on the training data itself, then on the test data, both using the cutoff value of 0.5, which indicates the predicted probability greater than 0.5 will be categorized into class 1 cancelled and those less than 0.5 will be in class 0 not cancelled.

3.4 Neural Network

Neural Network, yet another efficient and reliable algorithm, is a model that resembles the biological response to a stimulus using a network of interconnected neuron cells in order to develop a response. In this case, the predictors are the stimuli, and it is used to create a network of neurons until it reaches the output nodes of the classes. The distinctive feature to this sophisticated algorithm is its adjustments of weight in each iteration in order to produce a result with the least error rate.

3.4.1 Model Mechanism

The model uses the training and testing data from the tree data, since tree model has the most alteration of creating levels and factors to the variables which would be relevant to neural network algorithm which require creating dummy columns for the categorical variables. After dummy columns are created and compiled into a data frame, which has to be done in both training and testing data, the data is to be normalized in order to put them in a unified scale without the units. Then the model is run using the *neuralnet* function having run with two output nodes Car_Cancellation_0 and Car_Cancellation_1 since there are two classes. The model performance is then tested with both the training data itself and the testing data and their performance is evaluated using the confusion matrix.

Variables used in the models			
K-NN	**Classification Tree**	**Logistic Regression**	**Neural Network**
Car_Cancellation	**Car_Cancellation**	**Car_Cancellation**	**Car_Cancellation**
Online_booking	**User_id**	**User_id**	**User_id**
Mobile_site_booking	**Vehicle_model_id**	**Vehicle_model_id**	**Vehicle_model_id**
Gap_time	**Travel_type_id**	**Travel_type_id**	**Travel_type_id**
Call_In_Booking	**To_area_id**	**To_area_id**	**To_area_id**
Travel_Package	**Day_of_week_booking_created**	**Online_booking**	**Day_of_week_booking_created**
Trip_distance	**Book_month**	**Mobile_site_booking**	**Book_month**
	Book_day	**Day_of_week_booking_created**	**Book_day**
	Day_of_week_fromdate	**Booking_created_date**	**Day_of_week_fromdate**
	Trip_month	**Day_of_week_fromdate**	**Trip_month**
	Trip_day	**From_date_date**	**Trip_day**
	Gap_time	**From_date_time**	**Gap_time**
	Travel_package	**Gap_Time**	**Travel_package**
	Trip_distance	**Call_In_Booking**	**Trip_distance**
	Booking_method	**Travel_Package**	**Booking_method**
	Trip_period	**Trip_distance**	**Trip_period**

***Table 5** List of variables used in each model*

4. Classification Modelling (Original Imbalanced Data Versus Balanced Data)

All models for imbalanced data are made from the imbalanced data: taxi_train_imbalanced, whereas all models for balanced data are made from the balanced data: taxi_train_balanced. Green highlights are made for those with higher rates in their designated columns in order for a better comparison. In addition, training data has been tested itself to see the performance increment of the dataset for the model comparing its performance in the training data and the testing data. In reality, it would just be tested with the testing data to test for its measures in the unknown data, but this is a special case.

4.1 MODEL 1 - K- Nearest Neighbor

The table on the next page shows a comparison table for both balanced and imbalanced data using different k values. For both data, a random initial value for k is set as 7, and the models are also developed with the best k value that tuning suggested. The alternatives for k might differ in both data since k value could be determined around the tuning k value or based on their respective patterns of the confusion matrix measures.

For the original imbalanced data, the k value that the tuning suggested is 27, so the alternative k models are first intended to be made from k value of 1 to that value, but as it could be seen in the table, it is stopped at 14. The reason is that the accuracy rate of both k = 13 and 14 have the same accuracy rate as the tuning, and looking at the pattern of the sensitivity, it does not seem to increase since k = 2 and therefore, it is assumed that there will only be irrelevant number of models with little or no performance measure increases. On the other hand, for the balanced dataset, the alternative k values are tested from 1 to 10, and it is stopped at 10 since the sensitivity rate does not seem to increase significantly.

One thing to notice seeing the comparison table is that although the original dataset might have better accuracy rate with unreasonably low sensitivity rate, the balanced dataset has higher sensitivity rate but with lower accuracy rate. At this point, in this model, it can be assumed that the balanced data provides with a better model since not only is the accuracy important, but also the sensitivity rate (which is to detect the important class of 1 to identify the cancellation of rides by the drivers), so both measures have to be considered. Hence, balanced dataset shows a better performance of the models for this KNN algorithm.

Original Imbalanced Dataset				Balanced Dataset (*)			
K values	**Accuracy**	**Sensitivity**	**Specificity**	**K values**	**Accuracy**	**Sensitivity**	**Specificity**
7 (Initial K)	0.9247	0	1	7 (Initial K)	0.6328	0.46083	0.64623
27 (Tuning Suggest)	0.9274	0	1	1 (Tuning Suggest)	0.8378	0.15207	0.89145
Other Alternatives K (Both data might not have same alternative – alternatives could be set around the tuning k and their respective patterns)							
1	0.8813	0.064516	0.945186	2	0.8017	0.25806	0.84421
2	0.8826	0.69124	0.946268	3	0.7251	0.32258	0.75658
3	0.914	0.013825	0.984493	4	0.6957	0.36406	0.7216
4	0.9181	0.009216	0.989181	5	0.6746	0.41014	0.69528
5	0.9231	0.0092166	0.9945907	6	0.6445	0.39631	0.6639
6	0.9207	0.0046083	0.992427	8	0.6294	0.48848	0.64046
8	0.9261	0.0046083	0.9981969	9	0.6244	0.50230	0.63397
9	0.9261	0	0.998558	10	0.6301	0.46544	0.64299
10	0.9271	0.0046083	0.999279				
11	0.9271	0	0.999639				
12	0.9268	0	0.999278				
13	0.9274	0.0092166	0.9992788				
14	0.9274	0.009216	0.9992788				

Table 6 *Summary of K- Nearest Neighbor Performance Evaluation Using Confusion Matrix*

4.2 MODEL 2 - Classification Tree (Decision Tree)

The table shown later summarizes the performance of the classification tree on both data, where it is first used to test the training data itself, then the test data and finally prune the tree for better model performance. In this case, the performance used on the training data itself will not be compared; it is just used just to provide a basic ground of how the model is doing predicting its own data that is used to trained it. In this case, expectedly, both models have the overall highest performance measures in the training data, but it can be seen that the original imbalanced data has the higher measures with accuracy and specificity rate over 90%.

Original Imbalanced Dataset				Balanced Dataset (*)			
Data used	**Accuracy**	**Sensitivity**	**Specificity**	**Data used**	**Accuracy**	**Sensitivity**	**Specificity**
Training Data	0.9272	0.0798	0.9961	Training Data	0.7831	0.7847	0.7815
Testing Data	0.9284 **(+0.0012)**	0.0507 **(-0.0291)**	0.9971 **(+0.001)**	Testing Data	0.7348 **(-0.0483)**	0.6820 **(-0.1027)**	0.7389 **(-0.0426)**
Testing Data (Pruned)	0.9274 **(-0.001)**	0 **(-0.0507)**	1 **(+0.0029)**	Testing Data (Pruned)	0.7348 **(+/-0)**	0.6820 **(+/-0)**	0.7389 **(+/-0)**

Table 7 *Summary of Classification Tree Evaluation Using Confusion Matrix*

Another thing to be noted is the sensitivity rate again and it can be clearly seen that it is a problem in the original imbalanced data, where the sensitivity rate is unexceptionally low. Comparably in the balanced dataset, although the accuracy rate is not as high, it is at a reasonable rate of 70% and over, and the sensitivity rate is reasonable with the best pruned tree model having sensitivity of 68.203% which is more favorable compared to all other measures.

Hence, again for this model, the balanced dataset has better sensitivity and specificity rates which seems reasonable for the model, and overall, the balanced data model has a more realistic model performance than the imbalanced one with little or no sensitivity rate, which is

not applicable for real case situation. In addition, if the code in the R file is run, it can also be seen that the tree has only one node for its plot after pruning, and this is not right given that one class is lower than the other which has influenced and biased the tree algorithm.

4.3 MODEL 3 - Logistic Regression

Table 8 presents the comparison of the logistic regression model between the original imbalance dataset and balanced dataset. Similar pattern with the first two models can be observed where the accuracy of the imbalanced data is higher but with a low sensitivity rate, whereas in the balanced dataset, the accuracy is moderate, but the sensitivity is comparatively higher and more reasonable. Both models are formed from the data processing and the predicted values on both models are converted to target class with the cutoff value of 0.5.

From there we can see that the balanced dataset provides a better model putting aside the accuracy rate and in fact, this is more realistic as said above since it is not favorable to have a model which has little or no capability to predict one class (whether it is important or not).

Original Imbalanced Dataset				Balanced Dataset			
Data used	**Accuracy**	**Sensitivity**	**Specificity**	**Data used**	**Accuracy**	**Sensitivity**	**Specificity**
Training Data	0.9272	0.0837	0.9958	Training Data	0.7714	0.8035	0.7390
Testing Data	0.9268 **(+0.004)**	0.0553 **(-0.0284)**	0.9950 **(-0.0008)**	Testing Data	0.7 **(-0.0714)**	0.6221 **(-0.1814)**	0.7061 **(-0.0329)**

__Table 8__ Summary of Logistic Regression Evaluation Using Confusion Matrix

4.4 MODEL 4 – Neural Network

The table below shows the comparison of the neural network model between the original imbalance dataset and the balanced dataset. From there we can see that the imbalanced dataset has better performance with its training data whereas the balanced dataset has a better performance with the testing data, with the accuracy rate of 80.03%, sensitivity rate of 64.06% and specificity rate of 81.28%. In this model as well, we can see that it has a similar pattern with the other three models with the performance pattern of having high accuracy and specificity rate with low sensitivity rates, which is less realistic to be deployed as a model.

Original Imbalanced Dataset				Balanced Dataset			
Data used	**Accuracy**	**Sensitivity**	**Specificity**	**Data used**	**Accuracy**	**Sensitivity**	**Specificity**
Training Data	0.9269	0.1103	0.9934	Training Data	0.7704	0.7856	07551
Testing Data	0.9258 **(-0.0011)**	0.0645 **(-0.0458)**	0.9931 **(-0.0003)**	Testing Data	0.8003 **(+0.0299)**	0.6406 **(-0.145)**	0.8128 **(+0.0577)**

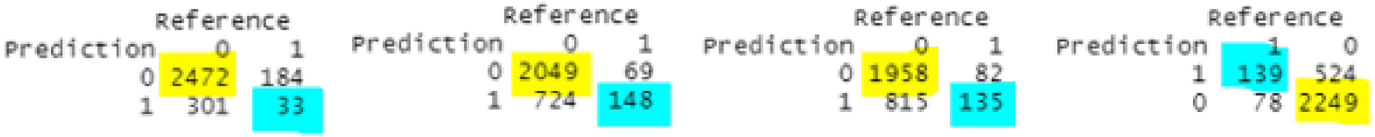
Table 9 *Summary of Neural Network Evaluation Using Confusion Matrix*

Therefore, after analyzing the model performances in both original imbalanced dataset and balanced dataset, it can be concluded that although the imbalanced dataset models have higher accuracy and specificity rate, their sensitivity rate are generally exceptionally low. Depending on the way the model is built, it can be either an issue or not however, in this case of taxi cancellation by the driver, it could more likely be an issue. Drawing back to the business problem where it is the problem of the drivers cancelling the ride, I would assume that the more important class for the model to identify is the class 1 of car being cancelled. Hence, in selecting the models, both the accuracy and sensitivity rate will be considered and either one having too high and the other one being too low will not work out, and the models from the balanced dataset will be selected.

5. Model Selection

5.1 Confusion Matrix Measures

Upon selecting which models to use, first it is important to analyze the performance metrics, and indeed the confusion matrix measures is a great measure for model performance evaluation. The confusion matrix for the models from the selected balanced dataset has been pasted below to illustrate the measure metrics, as well as a summary table for these measures from their respective confusion matrix. For Figure 13 Confusion Matrix the yellow highlight indicates the true negative whereas the turquoise highlight indicates the true positive. This highlight has been done to prevent confusion with different confusion matrix. For Table 10, the green highlight indicates the highest performance rating whereas the yellow highlight indicates the second highest performance rating.



```
          Reference
Prediction    0    1
         0 2472  184
         1  301   33

          Reference
Prediction    0    1
         0 2049   69
         1  724  148

          Reference
Prediction    0    1
         0 1958   82
         1  815  135

          Reference
Prediction    1    0
         1  139  524
         0   78 2249
```

Figure 15 *Confusion Matrix for all models (Left to Right: KNN, Classification Tree, Logistic Regression and Neural Network*

Models				
Performance Measures	K- Nearest Neighbor (tuning suggest k = 1)	Classification Tree	Logistic Regression	Neural Network
Accuracy	0.8378	0.7348	0.7000	0.8003
Sensitivity	0.15207	0.68203	0.62212	0.64055
Specificity	0.89145	0.73891	0.70609	0.81284
Precision$\frac{TP}{(FP+TP)}$	0.0988	0.16972	0.14211	0.20965
Type I Error (FP)	301 (10.07%)	724 (24.21%)	815 (27.26%)	524 (17.53%)
Type II Error (FN)	184 (6.15%)	69 (2.31%)	82 (2.74%)	78 (2.61%)

Table 10 *Summary of Performance Metrics Using Confusion Matrix*

Accuracy is not the only measure that needs to be taken into consideration. Often times, accuracy is the most straightforward method to access the general performance of the model since it measures the proportion of all correctly identified samples in the class. However, there is a major drawback to the accuracy, in case of the imbalanced dataset (although it is not applicable in this case since the models are built from the balanced dataset), the model can easily obtain high accuracy if it predicts most of the majority class correctly, but it becomes questionable of its predictivity of the minority class. My assumption here is that all important metrics of the model have to be analyzed instead of diving straight to the model with the highest accuracy. After said that, from Table 10 it can be seen that KNN has the highest accuracy rate of 83.76%, followed by the Neural Network model with 80.03%.

Sensitivity represents how well a model predicts the positive class, and indeed the higher sensitivity the better, whereas specificity represents how well a model predicts the negative class. Both measures could be equally important, but it varies from cases to cases and the higher these measures are, the better. The highest sensitivity can be seen in classification tree model with 68.2% and the second highest being 64.06%, and the highest specificity rate can be seen in KNN (with the highest accuracy) with 89.15% with the second highest being 81.28%. From here, we can see that the accuracy rate of the model is greatly affected by the specificity rate seeing how the model with highest specificity the highest accuracy and the model with the second highest having had the same pattern. Until this measure, it can be denoted that the top 2 best model is KNN and Neural Network, with KNN having the highest accuracy and specificity rate and the Neural Network being in second in those measures. Since the three major metrics are compared, the other 2 models will be excluded from the selection in order to finalize the final model to be selected.

Other important measures to be considered are the precision rate, type I error and type II error. Precision rate indicates when predicted 1, how often it is correctly predicted, type I error is wrongly predicted as positive class where it is actually a negative class, and the type II error is wrongly predicted as negative class where it is actually a positive class. Neural Network has the highest precision, KNN has the lowest type I error, and the classification tree has the lowest type II error. The precision rate and type II error would be more emphasized, assuming that the positive class is determined as the important class. Higher precision rate in the model indicates the accuracy rate among the predicted positive class, and the lower type II error indicates that it has more correctly predictive positive class with fewer wrongly classified instances. Aligned to this case, type II error could be a serious issue when there are situations where it is predicted not cancelled (class 0) but it is actually cancelled (class 1) which could create issue for YourCab since it is aiming to predict if drivers are going to cancel. There would be a greater cost to incorrectly predict the taxi as being not cancelled (while it is cancelled) than incorrectly predict the taxi as being cancelled (while it is not cancelled). Being able to not predictive taxi cancellation could bring issue to the business so having fewer type II error is preferred in selecting the model, so Neural Network model has way fewer errors than the KNN model.

To summarize so far, KNN model has a higher accuracy and specificity rate and lower type I error whereas the Neural Network model has a better sensitivity and precision rate and a lower type II error rate. Hence, in this case with similar accuracy rate, the model with better sensitivity and precision rate, with fewer type II error will be selected, and the selected model is the **Neural Network model** based on this confusion matrix measures.

5.2 ROC Curve

ROC, standing for Receiver Operating Characteristics, is often used to show the trade-off between the sensitivity and specificity of the model and the Area Under the Curve (AUC) is the measure of the usefulness of a test where greater percentage of AUC indicates a better model. The straight line in these graphs represents the performance of the random model classifier, and it is a straight line indicating that 50% correct predictions with AUC of 50% where True Positive rate is equal to the False Positive Rate. This is used to compared with the model since random model is at the middle so any model above that is better.

Figure 14 below illustrates the combined ROC Curve plot for all the models. It can be seen that the KNN model has the least Area Under Curve (AUC) of 52.5% where the best performing models in terms of the AUC is the Neural Network with 72.7% and Logistic Regression with 72.8%. Classification Tree model stands in between with 71.1%. In terms of the ROC, it can be concluded that all models excluding the KNN perform relatively close together with overall over 70%.

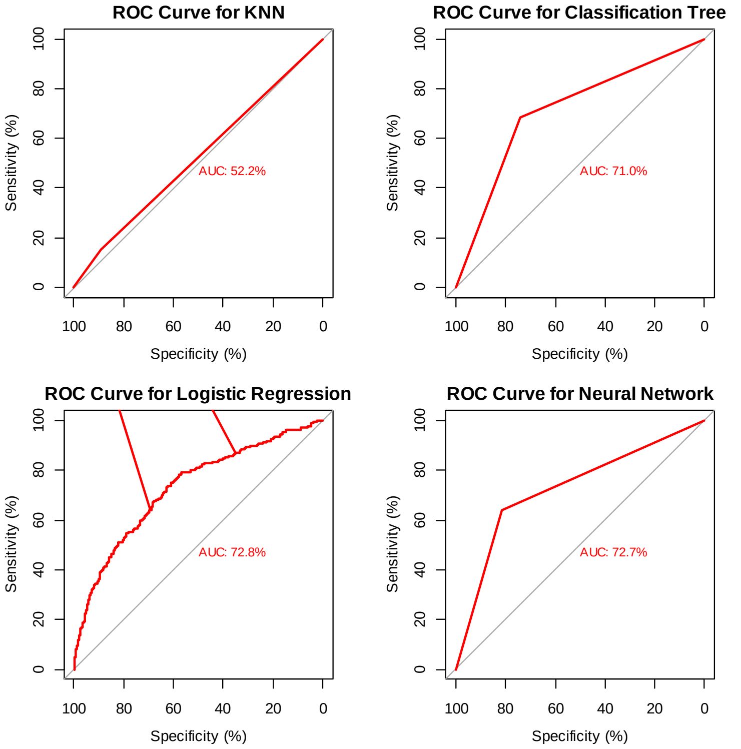


Figure 16 Combined ROC Plot for all models

To summarize the result from the performance evaluation from both the confusion matrix measures and the ROC curve performance, it can be concluded that the "**Neural Network**" is the best performing model from all four models. KNN draws in close with better accuracy and specificity rate, but considering the goal alignment with the business problem, better sensitivity and precision as well as the fewer Type II error prioritize the final selection. In addition to that, Neural Network has a better AUC than KNN, which makes it the better performing model and the best performing model among the other models.

6. Variable Importance

Although the final selected model Neural Network has the best performance, it is hard to interpret the variable contribution to the model and the importance of these variables. The only way to identify through the model will be through the weights of the model since they determine the ‘value’ of the outcome in more or less through the numeric values of the weights. Knowing this would not be sufficient for to affirm the variables, metrics from other relevant models have also been used to compare.

The variable importance from the classification ree has been analyzed as well as the significance values of the logistic regression have been accessed in order to provide for a better comparison and see if there are any overlap variables deemed as important in their respective models. The Table below outlines the top 5 most important variable from each model, excluding the knn model since this model does not have mechanisms to support the variable importance to my knowledge. The highlights have been done with green indicating as the most influential variable, yellow as the next influential and proceeds by the gray highlight.

From here it can be seen that the booking month could be an important variable for the business to analyze why drivers cancel followed by the booking method. Mentioned earlier in General Attribute Analysis section on how the cancellation rate having the most in other less frequent booking platforms of online and mobile booking, this could be something that the business could keep an eye on. Is it that the drivers are not used to the new booking platform? Are there any functions that could be altered or update in these platforms, or more training for the drivers on the functionality of the platforms? Another variable that could be important is probably the trip distance. Is further distance having more cancellation? Does it have anything to do with the vehicle itself (Vehicle model ID highlighted for this speculation). This could be an important discovery for the business to reflect on, not only just being able to predict the cancellation, but also taking actions to minimalize the possible causes.

Classification Tree	Logistic Regression	Neural Network
Book Month	Travel type 2, online booking, mobile booking, trip distance	Trip Month 10
Gap Time	Vehicle Model ID	Trip Month 10
Booking_Method	Travel Type 3	Book Month 10
Trip Distance	Trip_time 03:36:00	Trip Day
Book Day	Trip time 22:33:00	Book Month 6

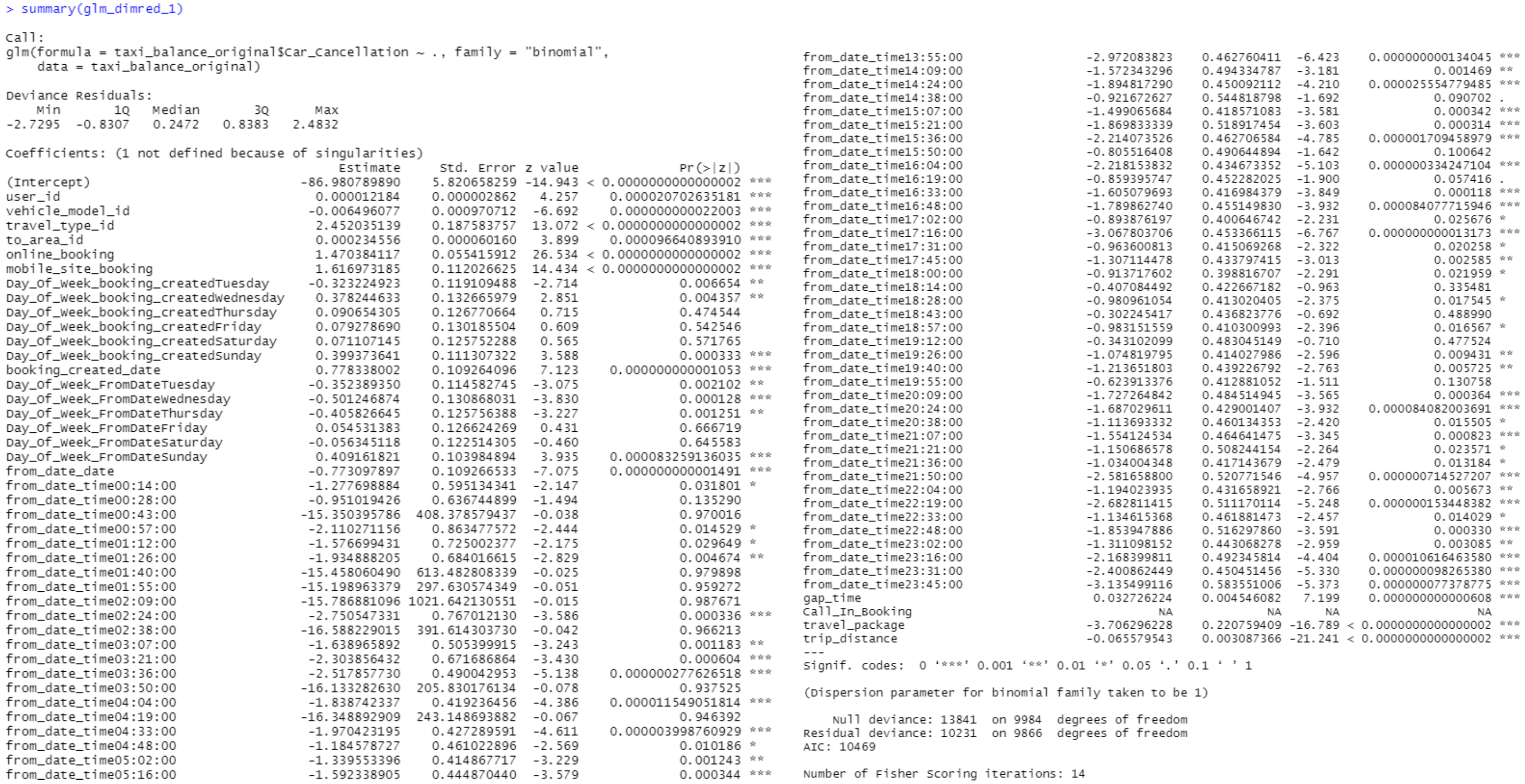
Table 11 *Top 5 Most Contributing Variables to the Model*

7. Conclusion

To conclude, Neural Network is selected as the final model to be deployed for YourCabs with its best performance in the performance evaluation process and its high predictivity of the taxi cancellation. The model using the balanced dataset has been selected in order to optimize the performance of the model while the model is validated using the imbalanced testing data in order to realistically predict the actual instances of the business for possible taxi cancellations. Although the model has the good predictivity, however due to its “black box nature” of its lack of transparency of the model, it may not satisfy the explanation of the variables importance when delivering to the business as a client. Therefore, it is recommended to deploy the model for predicting the instances for taxi cancellations, but in terms of identifying the causes of the cancellation, other methods have to be used. Aligning to the business goal which is to build a model to assist in predicting if the driver is cancelling the ride, the Neural Network model could be used.

With the limited time and understanding on the business mechanism, there are certain areas of improvement for further research. First is the feature engineering in order to create more effective variable models. This section has been done with the best of my knowledge and common sense, there could be certain features that have not been amplified. Next is the better data exploration. Effective data exploration certainly requires better understanding of the data in order to spot meaningful correlation and relationship between the variables and this area can definitely be improved. This also applies to better identifying the variable importance where there could be more effective and solid way of measuring the importance of each variable to the data and the model. Last but not least will be the choice and selection of the model. This means that the model selected might not be the most relevant model for other interpreter, or other analysts could have better interpretation on the selection of other models and algorithms. The best model could be varied determined by each analyst based on their understanding and interpretation of the business, and they might have different model algorithms better than the four models I have presented. These are the rooms of improvement that I have selected to encourage for future improvement to better predict the cancellations.

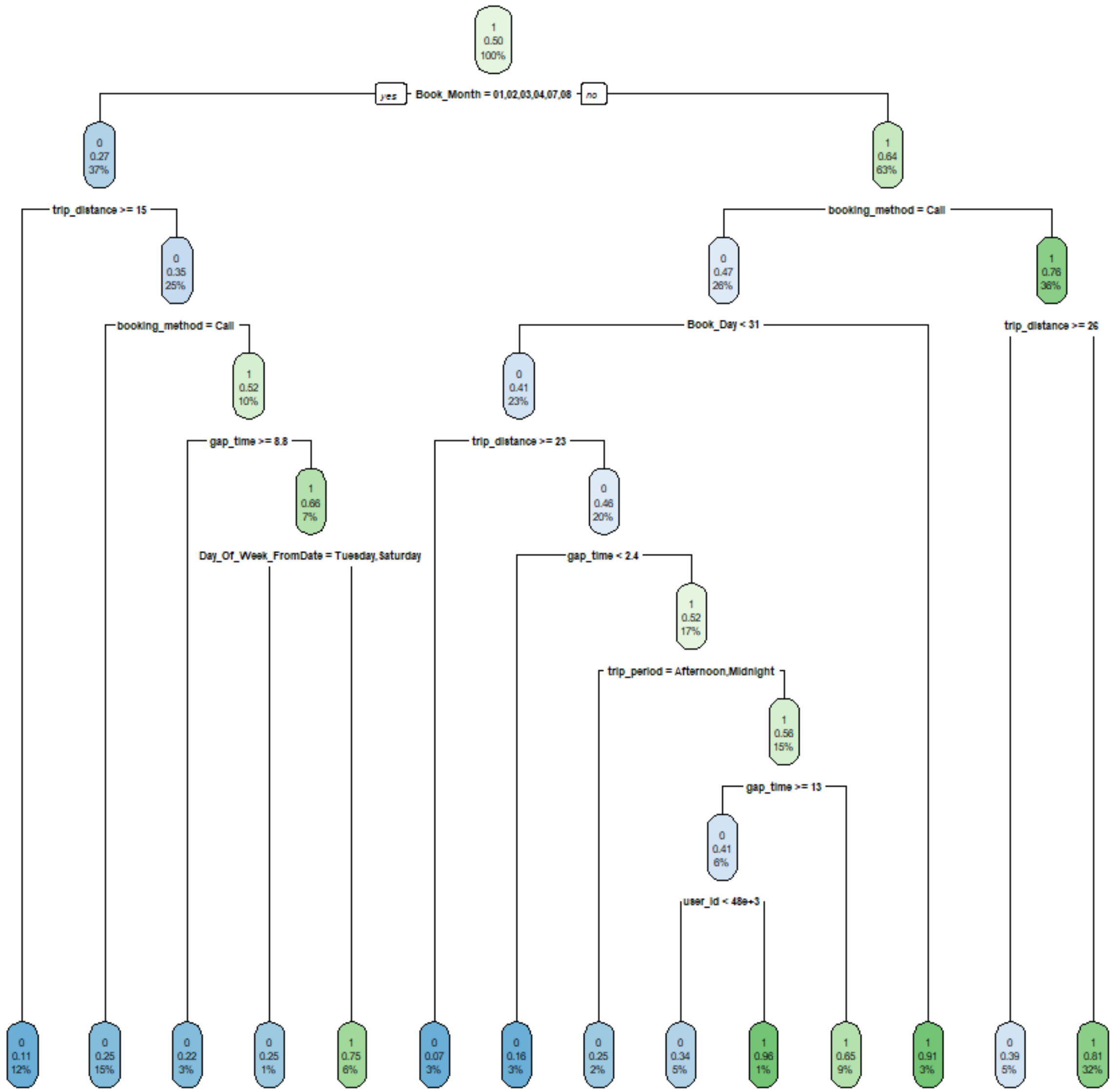
8. Appendix



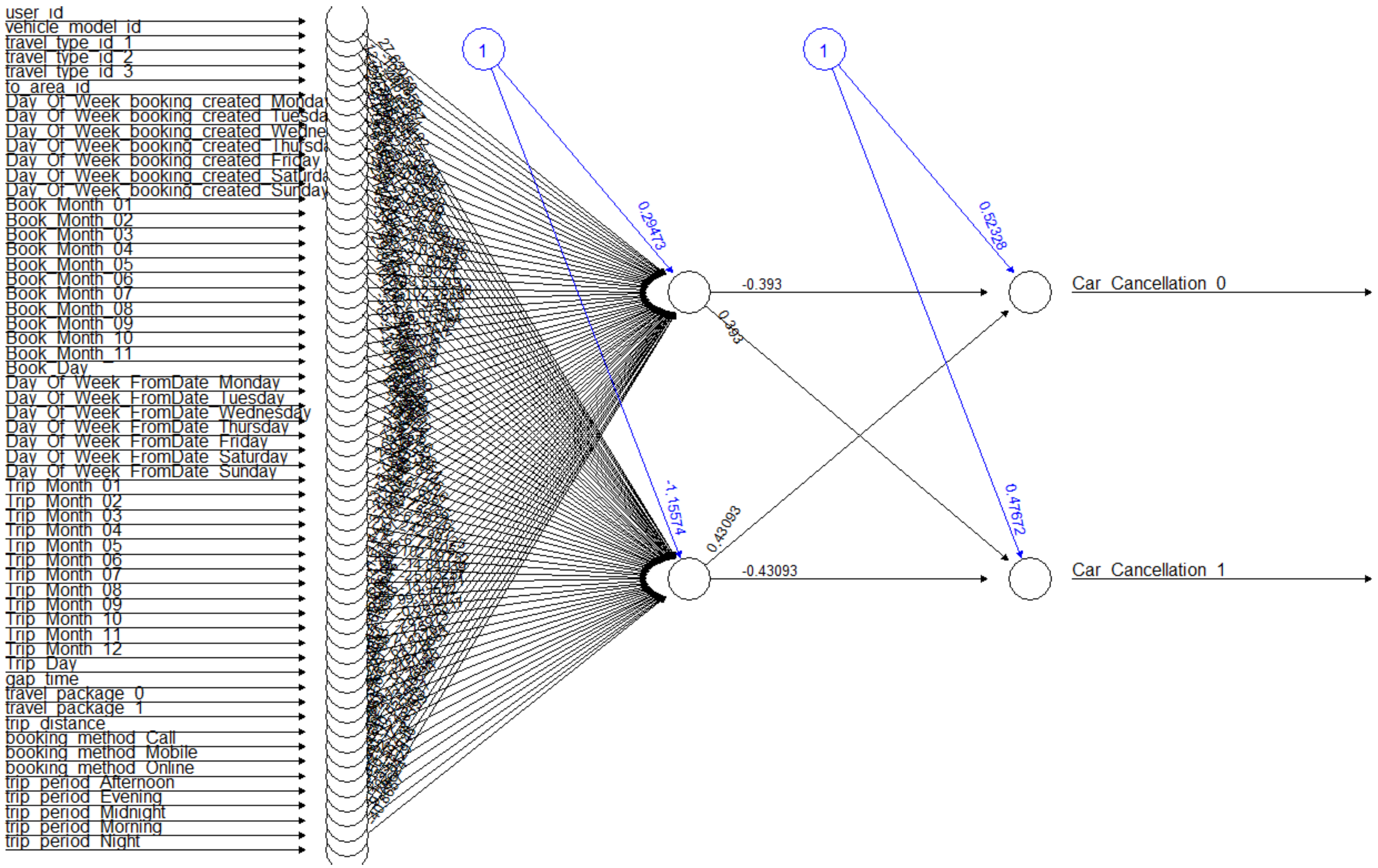
```
> summary(glm_dimred_1)

Call:
glm(formula = taxi_balance_original$Car_Cancellation ~ ., family = "binomial",
    data = taxi_balance_original)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.7295  -0.8307   0.2472   0.8383   2.4832

Coefficients: (1 not defined because of singularities)
                                     Estimate     Std. Error z value             Pr(>|z|)
(Intercept)                        -86.980789890    5.820658259 -14.943 < 0.0000000000000002 ***
user_id                              0.000012184    0.000002862   4.257    0.00002070263518 ***
vehicle_model_id                    -0.006496077    0.000970712  -6.692    0.00000000002200 3 ***
travel_type_id                       2.452035139    0.187583757  13.072 < 0.0000000000000002 ***
to_area_id                           0.000234556    0.000060160   3.899    0.00009664089391 0 ***
online_booking                       1.470384117    0.055415912  26.534 < 0.0000000000000002 ***
mobile_site_booking                  1.616973185    0.112026625  14.434 < 0.0000000000000002 ***
Day_Of_Week_booking_createdTuesday  -0.323224923    0.119109488  -2.714             0.006654 **
Day_Of_Week_booking_createdWednesday 0.378244633    0.132665979   2.851             0.004357 **
Day_Of_Week_booking_createdThursday  0.090654305    0.126770664   0.715             0.474544
Day_Of_Week_booking_createdFriday    0.079278690    0.130185504   0.609             0.542546
Day_Of_Week_booking_createdSaturday  0.071107145    0.125752288   0.565             0.571765
Day_Of_Week_booking_createdSunday    0.399373641    0.111307322   3.588             0.000333 ***
booking_created_date                 0.778338002    0.109264096   7.123    0.00000000001053 ***
Day_Of_Week_FromDateTuesday         -0.352389350    0.114582745  -3.075             0.002102 **
Day_Of_Week_FromDateWednesday       -0.501246874    0.130868031  -3.830             0.000128 ***
Day_Of_Week_FromDateThursday        -0.405826645    0.125756388  -3.227             0.001251 **
Day_Of_Week_FromDateFriday           0.054531383    0.126624269   0.431             0.666719
Day_Of_Week_FromDateSaturday        -0.056345118    0.122514305  -0.460             0.645583
Day_Of_Week_FromDateSunday           0.409161821    0.103984894   3.935    0.00008325913603 5 ***
from_date_date                      -0.773097897    0.109266533  -7.075    0.00000000001491 ***
from_date_time00:14:00              -1.277698884    0.595134341  -2.147             0.031801 *
from_date_time00:28:00              -0.951019426    0.636744899  -1.494             0.135290
from_date_time00:43:00             -15.350395786  408.378579437  -0.038             0.970016
from_date_time00:57:00              -2.110271156    0.863477572  -2.444             0.014529 *
from_date_time01:12:00              -1.576699431    0.725002377  -2.175             0.029649 *
from_date_time01:26:00              -1.934888205    0.684016615  -2.829             0.004674 **
from_date_time01:40:00             -15.458060490  613.482808339  -0.025             0.979898
from_date_time01:55:00             -15.198963379  297.630574349  -0.051             0.959272
from_date_time02:09:00             -15.786881096 1021.642130551  -0.015             0.987671
from_date_time02:24:00              -2.750547331    0.767012130  -3.586             0.000336 ***
from_date_time02:38:00             -16.588229015  391.614303730  -0.042             0.966213
from_date_time03:07:00              -1.638965892    0.505399915  -3.243             0.001183 **
from_date_time03:21:00              -2.303856432    0.671686864  -3.430             0.000604 ***
from_date_time03:36:00              -2.517857730    0.490042953  -5.138    0.00000027762651 8 ***
from_date_time03:50:00             -16.133282630  205.830176134  -0.078             0.937525
from_date_time04:04:00              -1.838742337    0.419236456  -4.386    0.00001154905181 4 ***
from_date_time04:19:00             -16.348892909  243.148693882  -0.067             0.946392
from_date_time04:33:00              -1.970423195    0.427289591  -4.611    0.00000399876092 9 ***
from_date_time04:48:00              -1.184578727    0.461022896  -2.569             0.010186 *
from_date_time05:02:00              -1.339553396    0.414867717  -3.229             0.001243 **
from_date_time05:16:00              -1.592338905    0.444870440  -3.579             0.000344 ***
```

```
from_date_time13:55:00              -2.972083823    0.462760411  -6.423    0.000000000134045 ***
from_date_time14:09:00              -1.572343296    0.494334787  -3.181             0.001469 **
from_date_time14:24:00              -1.894817290    0.450092112  -4.210    0.000025554779485 ***
from_date_time14:38:00              -0.921672627    0.544818798  -1.692             0.090702 .
from_date_time15:07:00              -1.499065684    0.418571083  -3.581             0.000342 ***
from_date_time15:21:00              -1.869833339    0.518917454  -3.603             0.000314 ***
from_date_time15:36:00              -2.214073526    0.462706584  -4.785    0.000001709458979 ***
from_date_time15:50:00              -0.805516408    0.490644894  -1.642             0.100642
from_date_time16:04:00              -2.218153832    0.434673352  -5.103    0.000000334247104 ***
from_date_time16:19:00              -0.859395747    0.452282025  -1.900             0.057416 .
from_date_time16:33:00              -1.605079693    0.416984379  -3.849             0.000118 ***
from_date_time16:48:00              -1.789862740    0.455149830  -3.932    0.000084077715946 ***
from_date_time17:02:00              -0.893876197    0.400646742  -2.231             0.025676 *
from_date_time17:16:00              -3.067803706    0.453366115  -6.767    0.000000000013173 ***
from_date_time17:31:00              -0.963600813    0.415069268  -2.322             0.020258 *
from_date_time17:45:00              -1.307114478    0.433797415  -3.013             0.002585 **
from_date_time18:00:00              -0.913717602    0.398816707  -2.291             0.021959 *
from_date_time18:14:00              -0.407084492    0.422667182  -0.963             0.335481
from_date_time18:28:00              -0.980961054    0.413020405  -2.375             0.017545 *
from_date_time18:43:00              -0.302245417    0.436823776  -0.692             0.488990
from_date_time18:57:00              -0.983151559    0.410300993  -2.396             0.016567 *
from_date_time19:12:00              -0.343102099    0.483045149  -0.710             0.477524
from_date_time19:26:00              -1.074819795    0.414027986  -2.596             0.009431 **
from_date_time19:40:00              -1.213651803    0.439226792  -2.763             0.005725 **
from_date_time19:55:00              -0.623913376    0.412881052  -1.511             0.130758
from_date_time20:09:00              -1.727264842    0.484514945  -3.565             0.000364 ***
from_date_time20:24:00              -1.687029611    0.429001407  -3.932    0.000084082003691 ***
from_date_time20:38:00              -1.113693332    0.460134353  -2.420             0.015505 *
from_date_time21:07:00              -1.554124534    0.464641475  -3.345             0.000823 ***
from_date_time21:21:00              -1.150686578    0.508244154  -2.264             0.023571 *
from_date_time21:36:00              -1.034004348    0.417143679  -2.479             0.013184 *
from_date_time21:50:00              -2.581658800    0.520771546  -4.957    0.000000714527207 ***
from_date_time22:04:00              -1.194023935    0.431658921  -2.766             0.005673 **
from_date_time22:19:00              -2.682811415    0.511170114  -5.248    0.000000153448382 ***
from_date_time22:33:00              -1.134615368    0.461881473  -2.457             0.014029 *
from_date_time22:48:00              -1.853947886    0.516297860  -3.591             0.000330 ***
from_date_time23:02:00              -1.311098152    0.443068278  -2.959             0.003085 **
from_date_time23:16:00              -2.168399811    0.492345814  -4.404    0.000010616463580 ***
from_date_time23:31:00              -2.400862449    0.450451456  -5.330    0.000000098265380 ***
from_date_time23:45:00              -3.135499116    0.583551006  -5.373    0.000000077378775 ***
gap_time                             0.032726224    0.004546082   7.199    0.000000000000608 ***
Call_In_Booking                               NA             NA      NA                   NA
travel_package                      -3.706296228    0.220759409 -16.789 < 0.0000000000000002 ***
trip_distance                       -0.065579543    0.003087366 -21.241 < 0.0000000000000002 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 13841  on 9984  degrees of freedom
Residual deviance: 10231  on 9866  degrees of freedom
AIC: 10469

Number of Fisher Scoring iterations: 14
```

Appendix 1 *Full Screenshot of the Logistic Regression from Dimension Reduction Section*



Appendix 3 *Classification Tree plot for the Balanced Dataset*



***Appendix 4** Neural Network (Final Selected) Model Plot*

END OF CASE 1

R Codes for Case Study 1

Case Study 2

Mortgage Payback Analysis

By Khin Thu Zar Thant

Webster University George Herbert Walker School of Business and Technology

CSDA 6010 Analytics Practicum

16th December 2022

Executive Summary

This case study aims to identify if the mortgage will be defaulted or paid off by assessing the borrowers' portfolio from the historical data. The dataset for this case study includes the transactions of the borrowers which loans have either defaulted or paid off, or not yet declared, at the time of collecting the data. Each borrower ID may have more than one transaction in the dataset before it is declared or not, which makes the dataset interesting to explore on how to alter the dataset to develop a model. Given the structure of the dataset, the borrowers whose loans have not been declared with then be used to predict on whether it has a higher chance of being defaulted or being paid off in the future at the loan maturity date.

The general procedure for this case study will be as follows. First, the missing data will be dealt with after identifying the business problem, project goals and the dataset briefing. The data preparation and analysis section will be proceeded by the variable and dataset manipulations, data exploration and dimension reduction. The variable and dataset manipulations will be conducted by altering the structure of the data by creating different data frames for each purpose. Defaulted observations will have its own data frames, as well as the payoff observations for model development purpose, whereas the unknown data frame which consists of borrowers which the outcome has not yet been declared will also be created to predict on its possibility on either outcome in the future. Data exploration will be done by exploring the trends of the overall dataset, as well as observing the trends in each class of defaulted and payoff on the state of each variable at the borrower's declaration timestamp. The next step would be the dimension reduction section where the general logistic regression is done to check the significance of each variable supported by the correlation plot and heat map to reassure the reduced variables. Finally, it is proceeded by the model selection, followed by the evaluation against the balanced and imbalanced dataset and lastly against the better performing models in order to select the final best performing model to predict the unknown data. In addition to that, an additional step is done to test the model performance (after testing with the testing data) which is to randomly select the first observations for each borrower ID and check with the actual outcome to assess if the model prediction is accurate or not. This gives a better understanding of the performance of the data to the actual data.

Table of Contents

List of Figures

List of Tables

1. Introduction

The dataset used in this case study is collected from the U.S residential mortgage- back securities (RMBS) securitization portfolios provided by the International Financial Research. It consists of report origination and performance observations for 50,000 residential borrowers of over sixty periods. Every loan has their own pattern, given its origination time, maturity time, outstanding balance and other factors given with their respective situations. For instance, loans may originate before the start of the observation period if loans are transferred between bank and investors as in securitization, and every loan repayment term could be different depending on the outstanding amount as well as the interest rate and other economic factors.

1.1 Business Problem

Default risk is the risk that a lender takes on in the chance that a borrower will be unable to make the required payments on their debt obligations. Whenever a lender extends credit to a borrower, there is a chance that the loan amount will not be paid back. Therefore, analytics have been putting effort in predicting if the loan is expected to be defaulted, and the risk of it being defaulted (likelihood of getting paid off) in order to reduce default risk to the lenders. This not only include the nature of the loan itself, but it is also determined by the economic factors (unemployment rate, GDP, etc.) and whenever an investor is evaluating an investment, they mainly access the default risk and put them into possible ratings as to borrow or not.

1.2 Project Goal

In this case study, a predictive model will be created to best classify given the portfolio specifications that it will be defaulted or paid off. This will be achieved using K Nearest Neighbor, Classification Tree, and Logistic Regression. Patterns among the instances of being default and paid off will be accessed individually in order to get a brief overview on how a typical loan will seem like, respectively. The primary goal of this project is to predict the remaining class 0 (which represents neither default nor paid off) into one of the two classes of either it being defaulted or will be paid off.

1.3 Dataset Briefings

The dataset contains a total of 622,489 instances with 23 variables, with a total of 50,000 borrowers' portfolio. Each borrower ID contributes to several instances in the dataset. The variable descriptions are as follows.

Variable Name	**Description**
ID	Borrower ID
Time	Time Stamp of observation
Orig_Time	Time stamp for origination
First_Time	Time stamp for first observation
Mat_Time	Time stamp for maturity
Balance_Time	Outstanding balance at observation time
LTV_Time	Loan To Value ratio at observation time (in %)
Interest_Rate_Time	Interest rate at observation time (in %)
HPI_Time	House price index at observation time (base year = 10)
GDP_Time	Gross domestic product (GDP) growth at observation time (in %)
UER_Time	Unemployment rate at observation time (in %)

Variable Name	**Description**
Retype_CO_Orig_Tiem	Real estate type (condominium = 1, otherwise = 0)
Retype_PU_Orig_Time	Real estate type planned (urban development = 1, otherwise = 0)
Reetype_SF_Orig_Time	Single-family home = 1, otherwise = 0
Investor_Orig_Time	Investor borrower = 1, otherwise = 0
Balance_Orig_Time	Outstanding balance at origination time
FICO_Orig_Time	FICO score at origination time, in %
LTV_Orig_Time	Loan-to-value ratio at origination time, in %
Interest_Rate_Orig_Time	Interest rate at origination time, in %
HPI_Orig_Time	House price index at origination time (base year = 100)
Default_Time	Default observation at observation time (Default = 1, otherwise = 0)
Payoff_Time	Payoff observation at observation time (Paid off = 1, otherwise = 0)
Status_Time	observation at observation time (Default = 1, payoff = 2, and non- default/non- payoff = 0)

Table 1 *– Variable Description for Mortgage Dataset*

The diagram below summarizes the summary of the overall procedure of the case study with a series of steps in order.

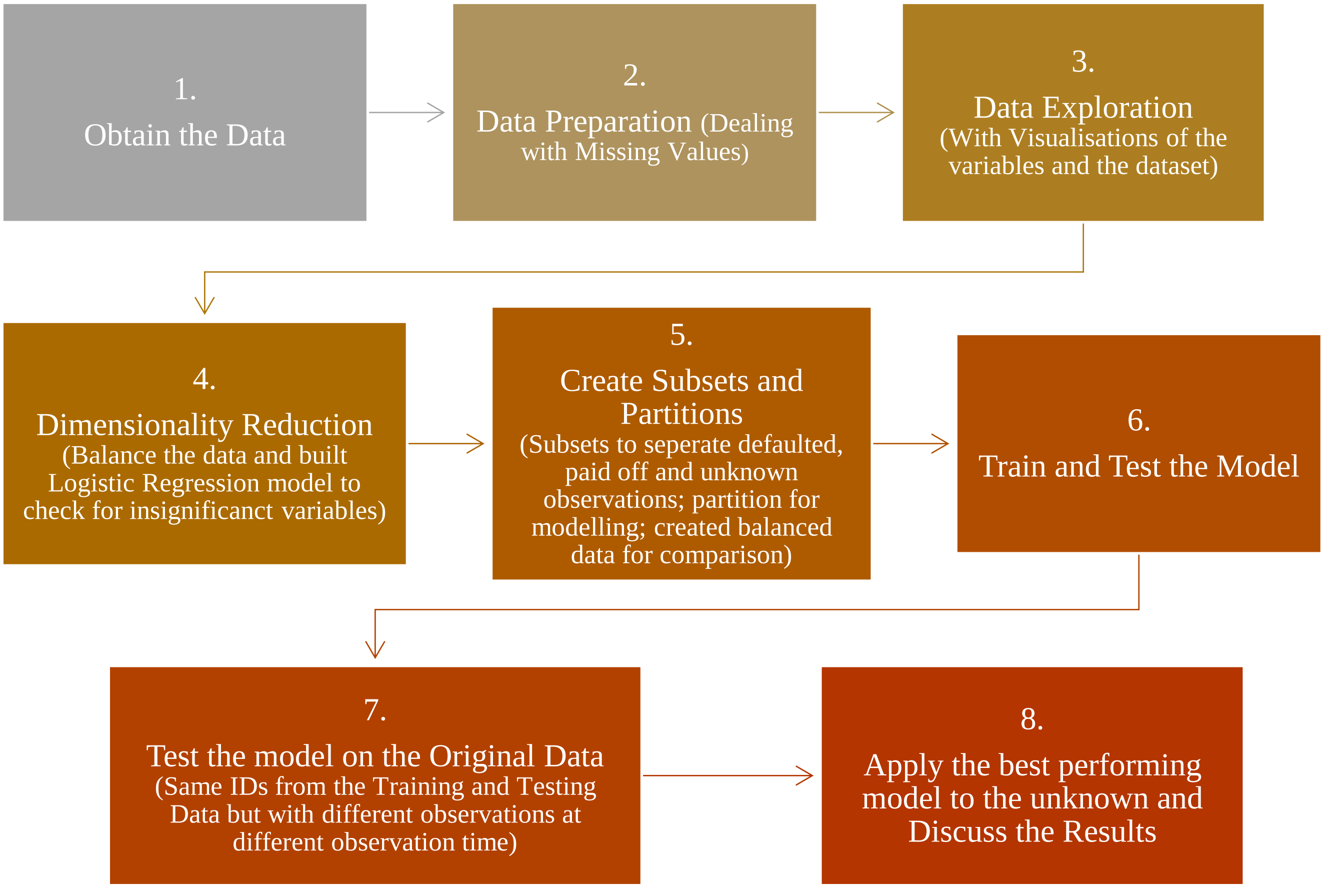
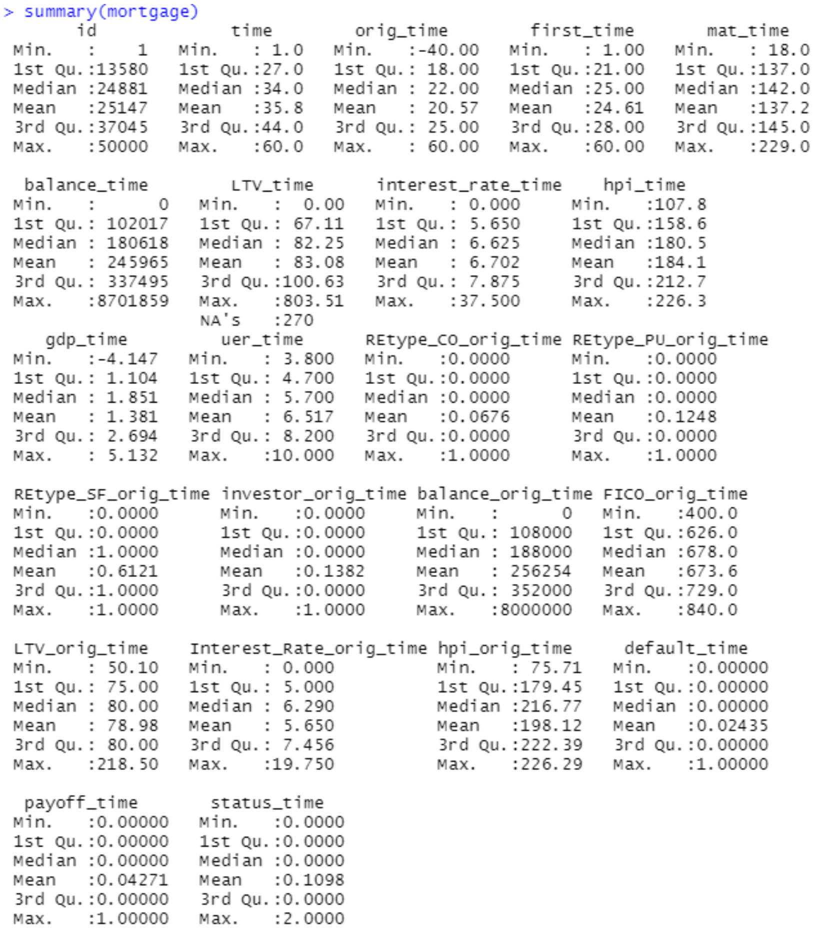

1.
Obtain the Data
2.
Data Preparation (Dealing with Missing Values)
3.
Data Exploration
(With Visualisations of the variables and the dataset)
4.
Dimensionality Reduction
(Balance the data and built Logistic Regression model to check for insignificanct variables)
5.
Create Subsets and Partitions
(Subsets to seperate defaulted, paid off and unknown observations; partition for modelling; created balanced data for comparison)
6.
Train and Test the Model
7.
Test the model on the Original Data
(Same IDs from the Training and Testing Data but with different observations at different observation time)
8.
Apply the best performing model to the unknown and Discuss the Results

Figure 1 *Diagram for the Data Analysis Process*

2. Data Preparation and Analysis

2.1 Data Preprocessing

The summary of the data is checked to analyze for required data preprocessing. The figure below attaches the summary of the whole dataset. It seems like all variables represent each aspect of the dataset, and no columns are visibly irrelevant to the context of the dataset. Even the ID column is necessary to represent each identity of the borrowers and each ID corresponds to several instances in the dataset depending on the number of observations made to this certain borrower. For instance, if it takes 8 observations for a borrower to get his/her mortgage loan defaulted, there will be 8 instances (rows) of that specific borrower ID with different timestamp of observations and outstanding amount.

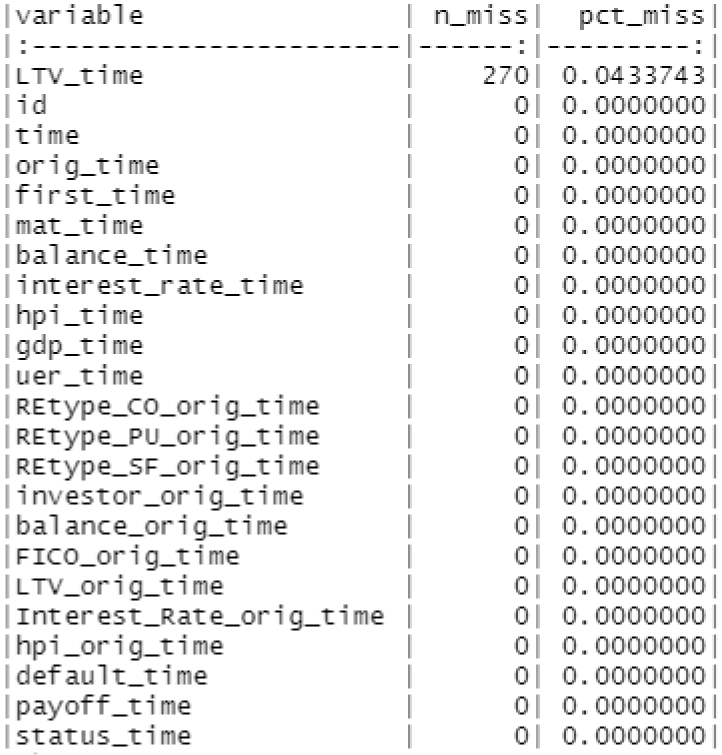


```
> summary(mortgage)
       id             time         orig_time        first_time        mat_time
 Min.   :    1   Min.   : 1.0   Min.   :-40.00   Min.   : 1.00   Min.   : 18.0
 1st Qu.:13580   1st Qu.:27.0   1st Qu.: 18.00   1st Qu.:21.00   1st Qu.:137.0
 Median :24881   Median :34.0   Median : 22.00   Median :25.00   Median :142.0
 Mean   :25147   Mean   :35.8   Mean   : 20.57   Mean   :24.61   Mean   :137.2
 3rd Qu.:37045   3rd Qu.:44.0   3rd Qu.: 25.00   3rd Qu.:28.00   3rd Qu.:145.0
 Max.   :50000   Max.   :60.0   Max.   : 60.00   Max.   :60.00   Max.   :229.0

  balance_time        LTV_time      interest_rate_time    hpi_time
 Min.   :      0   Min.   :  0.00   Min.   : 0.000     Min.   :107.8
 1st Qu.: 102017   1st Qu.: 67.11   1st Qu.: 5.650     1st Qu.:158.6
 Median : 180618   Median : 82.25   Median : 6.625     Median :180.5
 Mean   : 245965   Mean   : 83.08   Mean   : 6.702     Mean   :184.1
 3rd Qu.: 337495   3rd Qu.:100.63   3rd Qu.: 7.875     3rd Qu.:212.7
 Max.   :8701859   Max.   :803.51   Max.   :37.500     Max.   :226.3
                   NA's   :270
    gdp_time          uer_time      REtype_CO_orig_time REtype_PU_orig_time
 Min.   :-4.147   Min.   : 3.800   Min.   :0.0000      Min.   :0.0000
 1st Qu.: 1.104   1st Qu.: 4.700   1st Qu.:0.0000      1st Qu.:0.0000
 Median : 1.851   Median : 5.700   Median :0.0000      Median :0.0000
 Mean   : 1.381   Mean   : 6.517   Mean   :0.0676      Mean   :0.1248
 3rd Qu.: 2.694   3rd Qu.: 8.200   3rd Qu.:0.0000      3rd Qu.:0.0000
 Max.   : 5.132   Max.   :10.000   Max.   :1.0000      Max.   :1.0000

 REtype_SF_orig_time investor_orig_time balance_orig_time FICO_orig_time
 Min.   :0.0000      Min.   :0.0000     Min.   :      0   Min.   :400.0
 1st Qu.:0.0000      1st Qu.:0.0000     1st Qu.: 108000   1st Qu.:626.0
 Median :1.0000      Median :0.0000     Median : 188000   Median :678.0
 Mean   :0.6121      Mean   :0.1382     Mean   : 256254   Mean   :673.6
 3rd Qu.:1.0000      3rd Qu.:0.0000     3rd Qu.: 352000   3rd Qu.:729.0
 Max.   :1.0000      Max.   :1.0000     Max.   :8000000   Max.   :840.0

 LTV_orig_time    Interest_Rate_orig_time hpi_orig_time     default_time
 Min.   : 50.10   Min.   : 0.000          Min.   : 75.71   Min.   :0.00000
 1st Qu.: 75.00   1st Qu.: 5.000          1st Qu.:179.45   1st Qu.:0.00000
 Median : 80.00   Median : 6.290          Median :216.77   Median :0.00000
 Mean   : 78.98   Mean   : 5.650          Mean   :198.12   Mean   :0.02435
 3rd Qu.: 80.00   3rd Qu.: 7.456          3rd Qu.:222.39   3rd Qu.:0.00000
 Max.   :218.50   Max.   :19.750          Max.   :226.29   Max.   :1.00000

  payoff_time       status_time
 Min.   :0.00000   Min.   :0.0000
 1st Qu.:0.00000   1st Qu.:0.0000
 Median :0.00000   Median :0.0000
 Mean   :0.04271   Mean   :0.1098
 3rd Qu.:0.00000   3rd Qu.:0.0000
 Max.   :1.00000   Max.   :2.0000
```

***Figure 2** Summary of the Dataset*

2.2 Dealing with Missing and Null Values

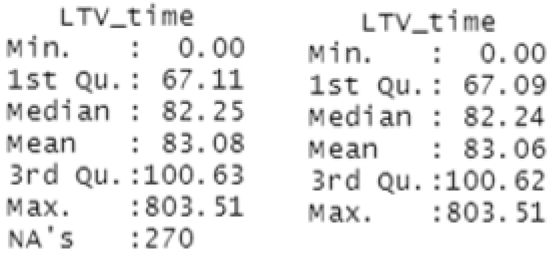
The missing values of the data is analyzed and there are a total of 270 observations with missing values which are found in the LTV_time variable (which is the Loan to Value Ratio at observation time). The diagram below shows the summary of the missing value observed using table format.



```
|variable                | n_miss|  pct_miss|
|:-----------------------|------:|---------:|
|LTV_time                |    270| 0.0433743|
|id                      |      0| 0.0000000|
|time                    |      0| 0.0000000|
|orig_time               |      0| 0.0000000|
|first_time              |      0| 0.0000000|
|mat_time                |      0| 0.0000000|
|balance_time            |      0| 0.0000000|
|interest_rate_time      |      0| 0.0000000|
|hpi_time                |      0| 0.0000000|
|gdp_time                |      0| 0.0000000|
|uer_time                |      0| 0.0000000|
|REtype_CO_orig_time     |      0| 0.0000000|
|REtype_PU_orig_time     |      0| 0.0000000|
|REtype_SF_orig_time     |      0| 0.0000000|
|investor_orig_time      |      0| 0.0000000|
|balance_orig_time       |      0| 0.0000000|
|FICO_orig_time          |      0| 0.0000000|
|LTV_orig_time           |      0| 0.0000000|
|Interest_Rate_orig_time |      0| 0.0000000|
|hpi_orig_time           |      0| 0.0000000|
|default_time            |      0| 0.0000000|
|payoff_time             |      0| 0.0000000|
|status_time             |      0| 0.0000000|
```

Figure 3 *Summary Table for Missing Values*

Instead of instantly removing these 270 records, the missing values are first analyzed, and it can be seen that these records signify 18 borrowers' ID portfolios, with a mixture of status of mortgage being default, paid off and neither. In order to prevent replacing the missing value with misinterpreted values, an alternative way of dealing with them is used which is using the "mice" package. The mice package consists of sophisticated functions which inspect the missing data pattern and impute the missing values "n" times which pools the results of the repeated analyses using custom imputation methods to replace the missing values. The summary of the new dataset created is also used to compare with the original dataset, and it can be seen that there is only roughly 0.02% difference in the mean value, indicating that the replacement of the missing value does not significantly affect the variable original values. The figure below illustrates the comparison between the new mortgage data created with the missing values replaced and the original data with the missing values.



```
   LTV_time           LTV_time
Min.   :  0.00     Min.   :  0.00
1st Qu.: 67.11     1st Qu.: 67.09
Median : 82.25     Median : 82.24
Mean   : 83.08     Mean   : 83.06
3rd Qu.:100.63     3rd Qu.:100.62
Max.   :803.51     Max.   :803.51
NA's   :270
```

Figure 4 *Original Data Versus New Data with Missing Values Replaced*

After dealing with the missing values, the null values are also checked in the data and results have shown that there are no null values in the dataset, which therefore no alterations are to be made regarding this issue.

2.3 Data Alternation and Variable Manipulation

Firstly, the count of the observations with the same ID variable is created named as "count." This gives an idea of how long a borrower takes in order to get either status: default (class 1) and payoff (class 2). The two columns of default_time and payoff_time are removed since it seems redundant since the status_time is already there to indicate the status of the loan. After this process, the goal is to create a data with only undeclared observations from the whole dataset (meaning just the borrowers with neither default nor paid off). Below are the series of steps that were taken to perform this.

First, the subset of the data with only defaulted observations is created (that is the data with only Class 1 in the status time), and the same is done for the payoff observations (which is the data with only Class 2). These are done with the filter function, so only the row with the status time of 1/2 will be shown and this left away the other observations with the same borrower ID which were later declared as default or payoff. For instance, while the ID 1 has 24 observations, the above execution will only give the 24th observation of the ID and filtered out the 23 remaining observations. For that reason, the %in% function was used in order to find matching ID values to bring out the full set of observations for each borrower ID. This process was done to both default and payoff dataset. Only after these two datasets are formed, it will be able to replace the duplicates from the original in order to get a pure dataset with only undeclared ID observations. The table below lists the data names along with their descriptions.

Data Names	Description
Mortgage_def	Data with final observation of the defaulted loans
Mortgage_payoff	Data with final observation of the paid off loans
Default_id_obs	Data with all observations for the defaulted loans
Payoff_id_obs	Data with all observations for the paid off loans
Unknown_obs	Data with undeclared status of the loan (neither defaulted nor payoff)

***Table 2** Description of the Data Frames created in the Dataset Alteration Section*

2.4 General Attribute Analysis with Visualizations

2.4.1 Mortgage Status Class Distribution

The presumed target variable for this data is the class for the mortgage status ranging from 0 to 2, where 0 represents that the mortgage is neither paid off nor defaulted, 1 represents that it is defaulted and 2 represents that it is paid off. Figure 4 has shown the bar graph visualization to visualize the distribution and we can see that class 0 has the most observations of over 90% of the total observations, where the other two occupies the remaining with class 2 slightly higher than the class 1. However, this is just a distribution of the whole dataset, where the class 0 contains some observations of the other classes before they are declared.

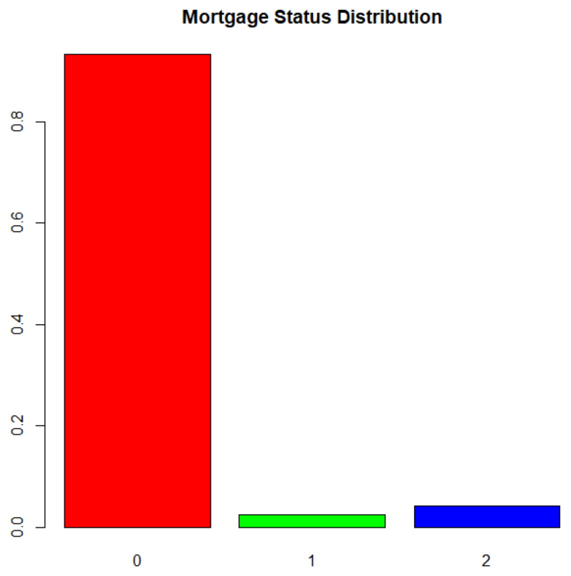


Figure 5 *Bar Graph for Status Distribution*

Therefore, the data is filtered in order to make sure that the Borrower ID which are later declared as defaulted, and payoff are removed from the 0 outcomes. Dealing with this data require detailed and mindful steps since it is always necessary to understand the data, understanding the data that this dataset includes all observations of the 50,000 borrower IDs, whether if the status (at the time of data collection) is defaulted, payoff or not yet declared. Therefore, in order to identify the actual groups of outcomes, it has to be from the 50,000 IDs, and the IDs which are later defaulted and paid off are to be removed since these observations will be considered as 'duplicate' and might not be able to observe the trend laying inside the data. For instance, there might be ID 'xx' which took the borrower 10 observations in order to be paid off, and the problem with the graph above is that this ID is included in the Class 2 (which represents payoff) as 1 count, while this contributes to the Class 0 (which is neither class) for 9 counts and this is duplicated. For this example, the count accounted for Class 2 is 1, but there should not be any count for the class 0. This concept is applied throughout the data in the further sections.

The graph below plots the real distribution of mortgage status in the whole dataset. It can be seen that the mortgages that have been paid off are weighing the most out of all with more than half of the whole data, with defaulted coming in second with around 30% and the unknown ones being the least with less than 20%. In such case, since the class 1 and 2 difference is significant, balancing the data might be required since the important class can be claimed as class 1, since the default risk is what the lender what to access so correctly classifying the class 1 might require balancing the data.

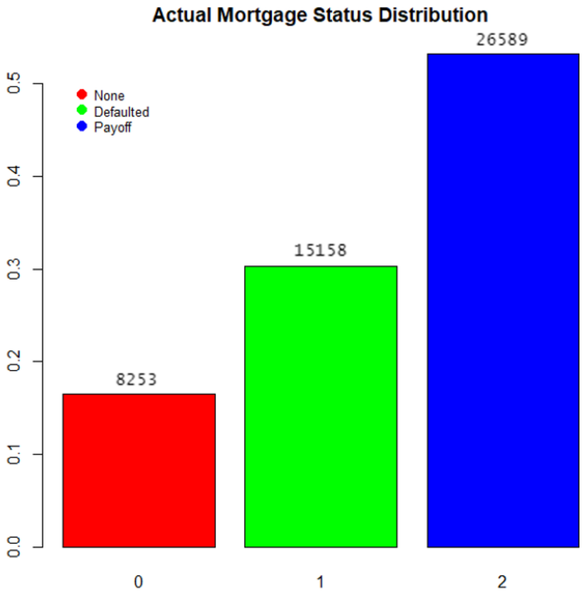


Figure 6 *Bar Graph for Actual Class Distribution*

2.4.2 General Box plot and Histogram/ bar graph for all variables

Boxplot has been plotted to all variables in order to spot for possible outliers and determine whether to deal with those specific variables or not. The categorical variables will be excluded from this boxplot since it will not visualize a good result when it is categorical variable. There is a total of 16 variables plotted which can be seen in Figure 7 as a combined plot.

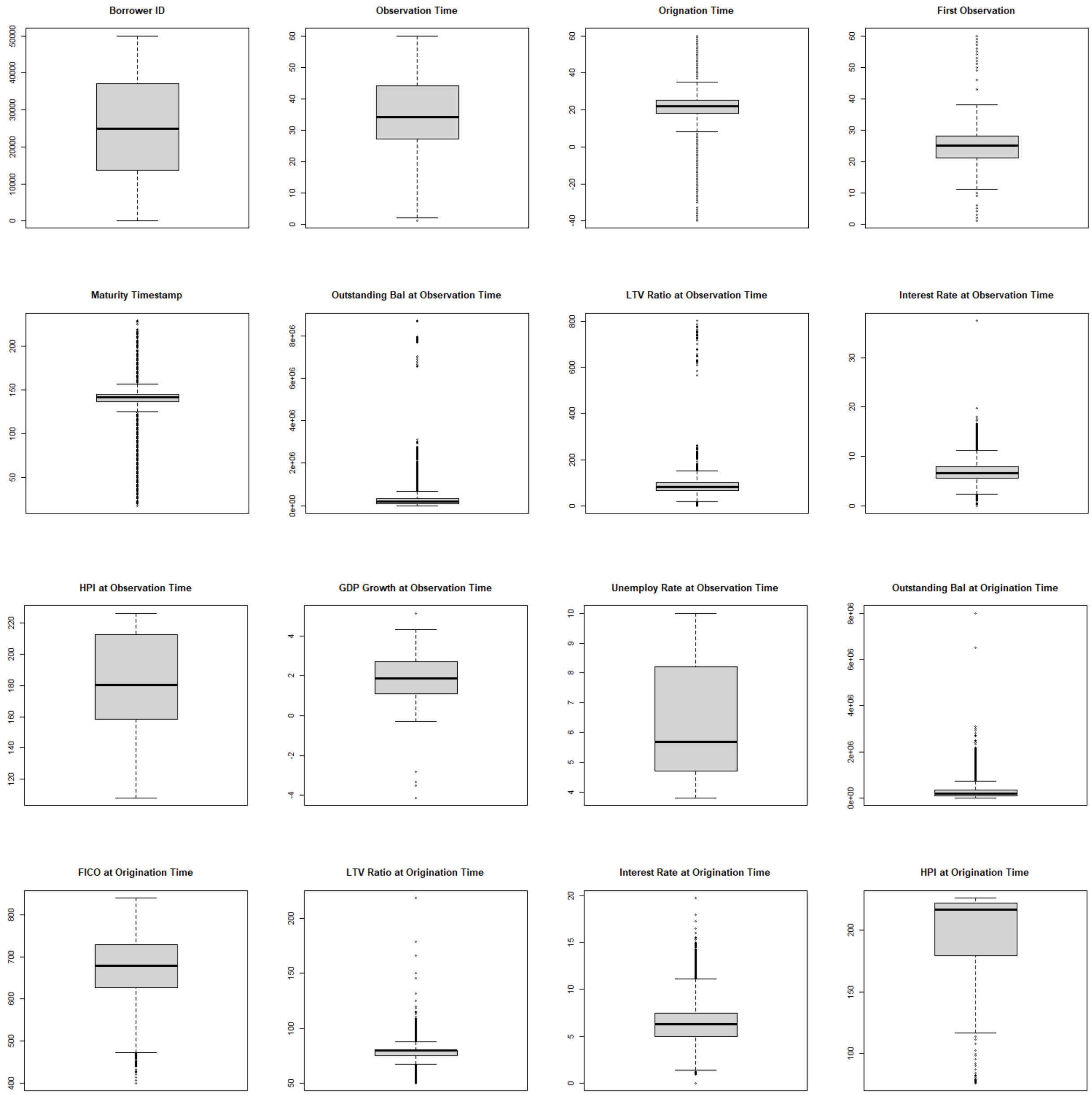


Figure 7 *Combined Box Plot for all Numeric Variables*

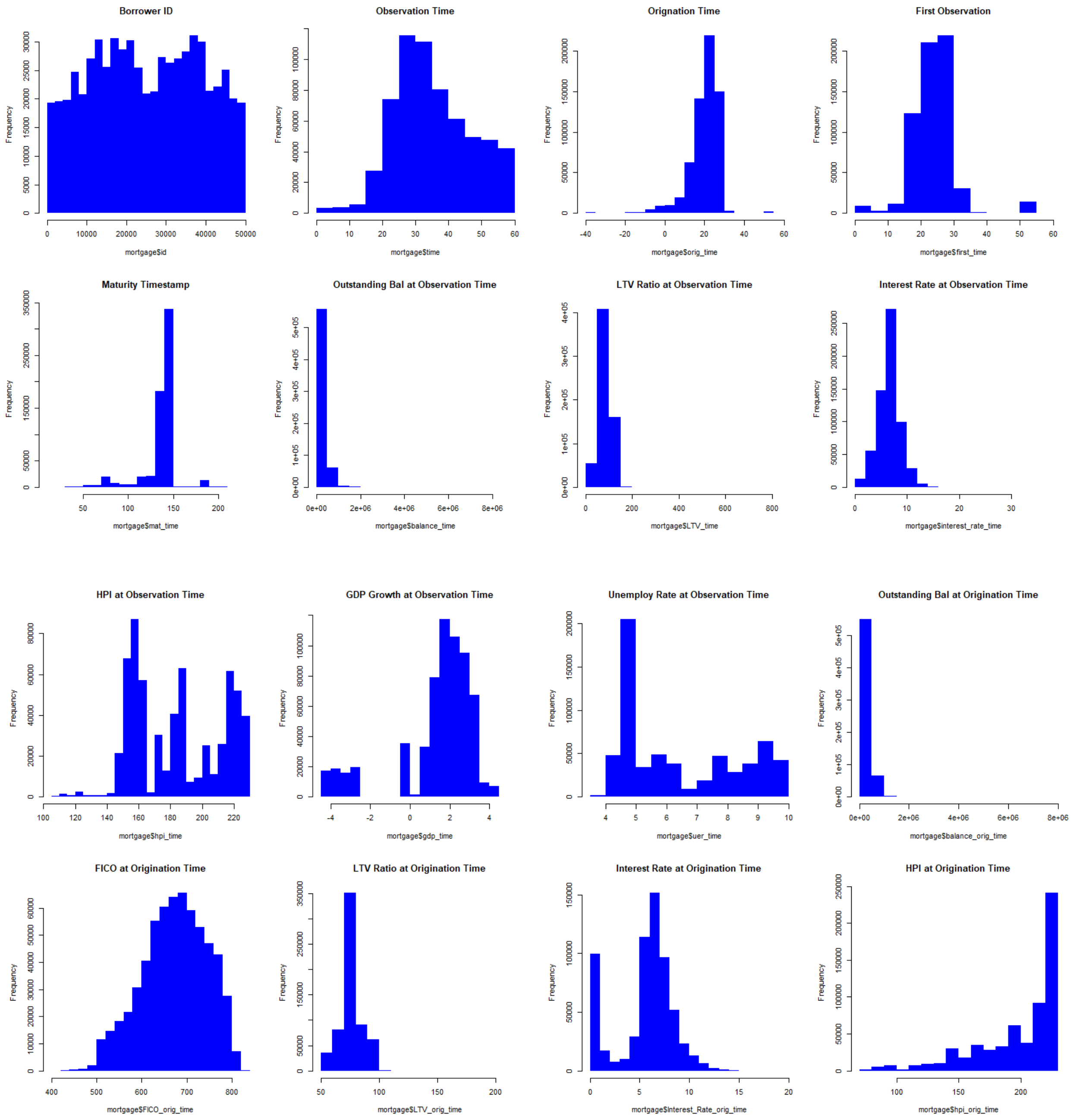


***Figure 8** Combined Histogram for data*

In addition to that, Figure 6 illustrates the combination plot for the histogram and Figure 7 illustrates the combination box plot to provide a brief overview of the distribution of the data in the range of each variable. While the histogram checks on for numeric variables, the bar graph is used to illustrate the categorical variables.

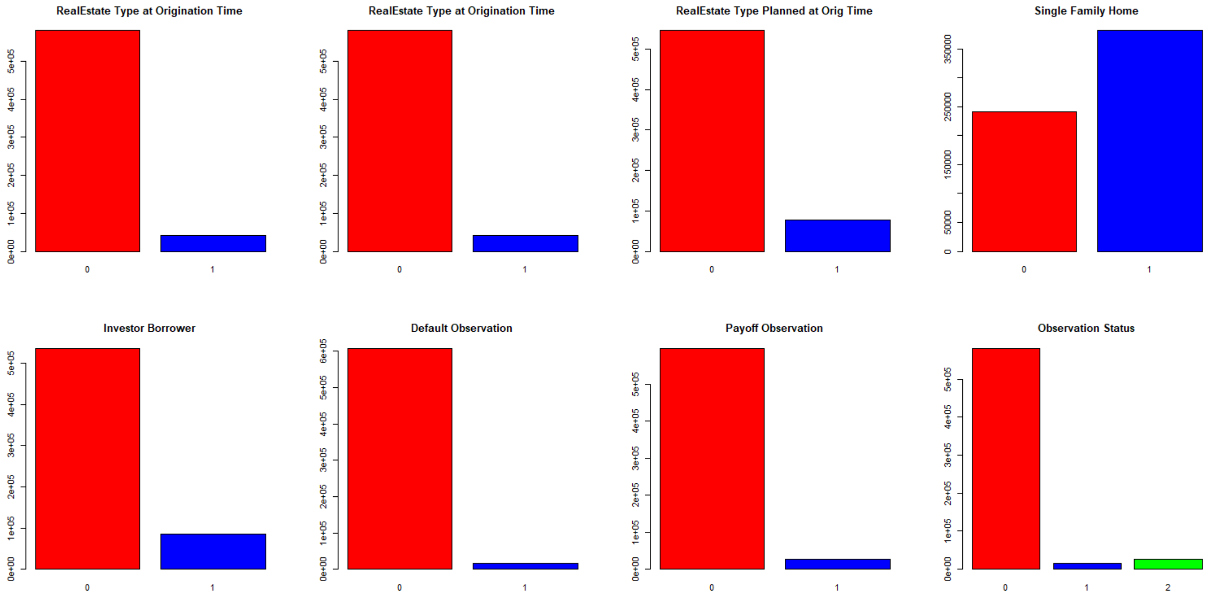


***Figure 9** Combined Bar Graph for Remaining Categorical Variables*

2.4.3 Analysis on Variables pattern for "Default" and "Payoff"

In this section, the dataset is subset into two parts, each for the mortgage status of "default" and "payoff" in order to analyze the variable patterns for each status. Here the third class of class 0 which is neither of the two classes, have been excluded since they are yet to be determined as one of these classes and it is not possible to filter the results with 0 classes as it will include the transaction observations of those which have not yet been defaulted or paid off but later been declared as one of these classes, which makes it unrealistic to extract to study the pattern for class 0 alone. In the R file, plots and graphs have been done to visualize the pattern and these plots will be attached in ***Appendix***. The summary table for the pattern observed from this exploration will be shown below.

Since there are some variables which have origination and observation values, columns will also be separated to provide better comparison at the time point of being in one of the two statuses. Another thing to note is that since the data comprises of borrowers with more than one observation, the subsets of the data will only include the observations at the time stamp of being declared as these classes. For instance, assuming that it takes 10 observations for ID 7 to be declared as class 1 of being defaulted, the subset of class default would only take the 10th observation of the point of being defaulted. This could give an idea of the state of each variable at the point of being defaulted or payoff.

	DEFAULT		**PAYOFF**	
Variables	At Origination Time	At Default Time	At Origination Time	At Payoff Time
Origination Time	Around 25 to 30		Around 20	
First Observation	Between 20 and 30		Between 20 and 30	
Maturity Time		150		Around 140 to 150
Outstanding Balance	Range below 400,000	Range below 500,000	Range below 400,000	Range below 500,000 (100,000 to 200,000)

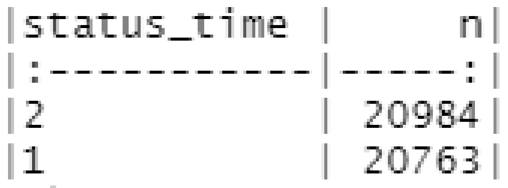
	DEFAULT		**PAYOFF**	
LTV ratio	Around 70 to 80 percent	Around 80 to 120 percent	Around 70 to 80 percent	Around 60 to 80 percent
Interest rate	Between 5 to 10 percent	Between 6 to 9 percent	Between 6 to 8 percent	Between 4 to 8 percent
HPI	Around 210 to 220	Around 150 to 170	Around 210 to 230	Around 230 to 250
GDP		Around 1 to 3		Around 2 to 3.5
Unemployment Rate		Between 4.5 to 5.5		Between 4.5 to 5
Real Estate Type	0 (not condominiums)		0 (not condominiums)	
Real Estate Type Planned	0 (not for urban development)		0 (not for urban development)	
Investor Borrower	1 (Investor Borrowers)		1 (Investor Borrowers)	
FICO Score	Around 600 to 700		Around 650 to 700	

***Table 3** Comparison Table for Exploration of Default Versus Payoff*

In addition to these explorations, the dataset is also explored to check if there are any ID duplicates. For instance, ID ‘xx’ paid off the mortgage loan at timestamp 25, but later at timestamp 50 for example, it could reissue the loan again for a different purpose. This could be a key factor to explore in order to dig deep on the borrower's history in determining the chances of that borrower being defaulted or paid off. Assume the previous example given with the ID 'xx' but this time the borrower defaulted the loan at timestamp 25. In this case, at timestamp 50 when that borrower came to issue another loan again, based on the history he could have more chance of getting a defaulted loan compared to a new borrower (for example the ID 'yy'). With this data exploration, while the model indicates that in the future ID 'xx' and 'yy' could pay off the loan (based on its algorithm), the bank/ borrowee could take different measures for 'xx' ('yy' in this case will have less to consider since there is no history about this ID before) to access this situation, which minimize the risk of not knowing that this ID 'xx' has defaulted the loan in the past. However, in this dataset, there is no duplicate ID numbers so this series of actions will not be needed.

2.5 Balancing the dataset

As seen in Figure 5 on the bar graph of the class distribution, it can be seen that the Class 2 which is the paid off has almost half more than the other class of class 1 which is the default. As mentioned above, although it is not heavily imbalanced, this dataset indefinitely needs to be balanced for the purpose of Class 1 predictability. The figure below shows the distribution after balancing.

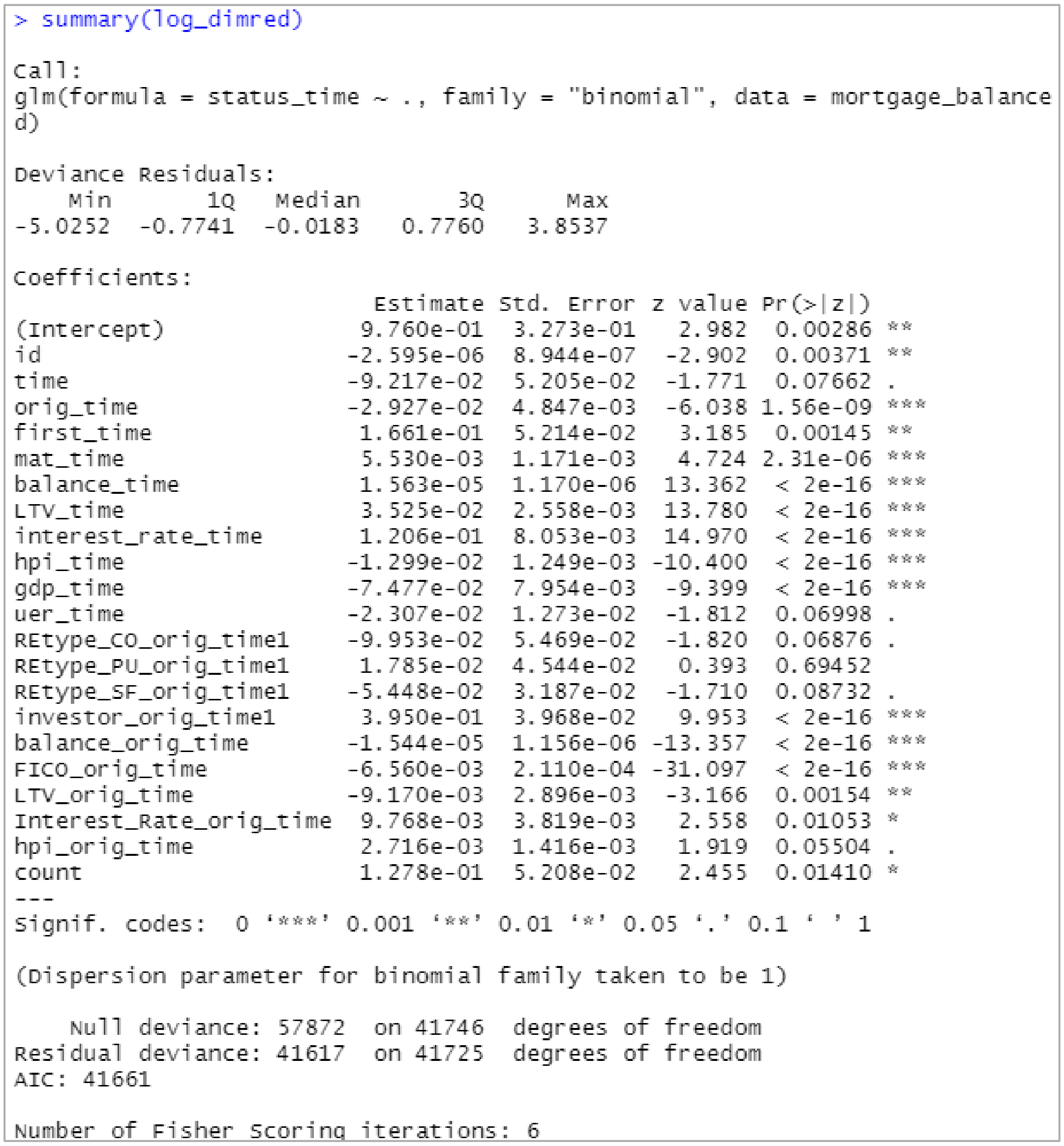


```
|status_time |     n|
|:-----------|-----:|
|2           | 20984|
|1           | 20763|
```

Figure 10 *Balanced Data Class Distribution*

2.6 Dimension Reduction

The next section is the dimension reduction. First, a general multinomial logistic regression is run to check for the variable’s coefficients and weights towards the target variable. Figure 9 below shows the summary of the model. Among that, it can be seen that Orig_time, Uer_time, retype_CO_orig_time and hpi_orig_time are not significant, with no stars in the model significance.



```
> summary(log_dimred)

Call:
glm(formula = status_time ~ ., family = "binomial", data = mortgage_balance
d)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-5.0252  -0.7741  -0.0183   0.7760   3.8537  

Coefficients:
                          Estimate Std. Error z value Pr(>|z|)    
(Intercept)              9.760e-01  3.273e-01   2.982  0.00286 ** 
id                      -2.595e-06  8.944e-07  -2.902  0.00371 ** 
time                    -9.217e-02  5.205e-02  -1.771  0.07662 .  
orig_time               -2.927e-02  4.847e-03  -6.038 1.56e-09 ***
first_time               1.661e-01  5.214e-02   3.185  0.00145 ** 
mat_time                 5.530e-03  1.171e-03   4.724 2.31e-06 ***
balance_time             1.563e-05  1.170e-06  13.362  < 2e-16 ***
LTV_time                 3.525e-02  2.558e-03  13.780  < 2e-16 ***
interest_rate_time       1.206e-01  8.053e-03  14.970  < 2e-16 ***
hpi_time                -1.299e-02  1.249e-03 -10.400  < 2e-16 ***
gdp_time                -7.477e-02  7.954e-03  -9.399  < 2e-16 ***
uer_time                -2.307e-02  1.273e-02  -1.812  0.06998 .  
REtype_CO_orig_time1    -9.953e-02  5.469e-02  -1.820  0.06876 .  
REtype_PU_orig_time1     1.785e-02  4.544e-02   0.393  0.69452    
REtype_SF_orig_time1    -5.448e-02  3.187e-02  -1.710  0.08732 .  
investor_orig_time1      3.950e-01  3.968e-02   9.953  < 2e-16 ***
balance_orig_time       -1.544e-05  1.156e-06 -13.357  < 2e-16 ***
FICO_orig_time          -6.560e-03  2.110e-04 -31.097  < 2e-16 ***
LTV_orig_time           -9.170e-03  2.896e-03  -3.166  0.00154 ** 
Interest_Rate_orig_time  9.768e-03  3.819e-03   2.558  0.01053 *  
hpi_orig_time            2.716e-03  1.416e-03   1.919  0.05504 .  
count                    1.278e-01  5.208e-02   2.455  0.01410 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 57872  on 41746  degrees of freedom
Residual deviance: 41617  on 41725  degrees of freedom
AIC: 41661

Number of Fisher Scoring iterations: 6
```

Figure 11
Summary of Multinomial Logistic Regression for Dimension Reduction

Then, another attempt is to plot simple correlation matrix to access the correlation between every variable, as well as a heatmap to better visualize the correlation result. Dimension reduction aims to reduce the uncorrelated and insignificant variables, so the focus is not only in identifying the correlation between the variables, but also the variables correlation to the target variable – status_time.

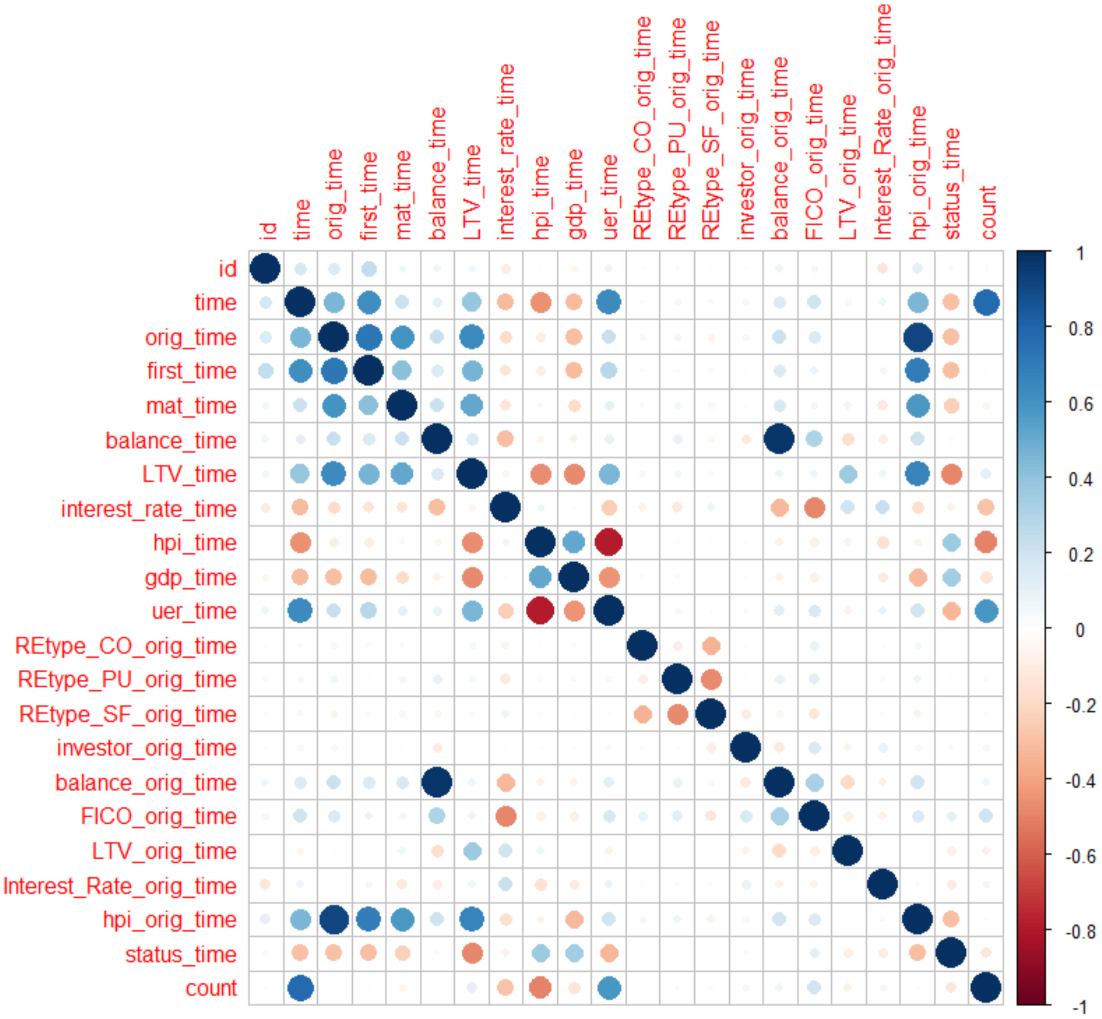


***Figure 12** Correlation Matrix for Dimension Reduction*

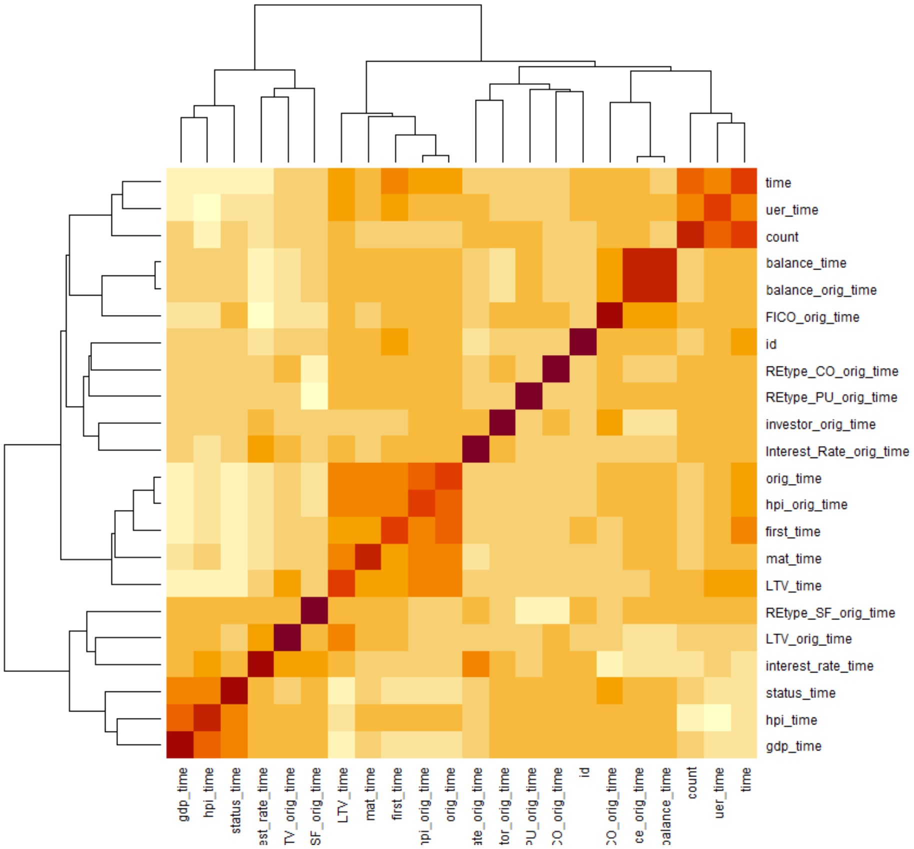


Figure 13 *Heat Map for Dimension Reduction*

From these two plots, there is no heavily correlated variables to status_time except for the hpi_time and gdp_time. Most heavily correlated variables are those against themselves, so it cannot be deluded of any significant or insignificant variables. From these three analyses, it has been decided to remove the following variables: Retype_CO_orig_time, Retype_PU_orig_time and Retype_SF_orig_time, since they are both insignificant and less correlated to the target variable. Below is the table of the remaining variables to be used in fitting the models.

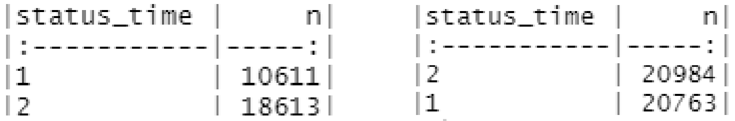
VARIABLE NAME	DESCRIPTION
ID	Refer to Table 1 for Variable Description
TIME	Refer to Table 1 for Variable Description

VARIABLE NAME	DESCRIPTION
ORIG_TIME	Refer to Table 1 for Variable Description
FIRST_TIME	Refer to Table 1 for Variable Description
MAT_TIME	Refer to Table 1 for Variable Description
BALANCE_TIME	Refer to Table 1 for Variable Description
LTV_TIME	Refer to Table 1 for Variable Description
INTEREST_RATE_TIME	Refer to Table 1 for Variable Description
HPI_TIME	Refer to Table 1 for Variable Description
GDP_TIME	Refer to Table 1 for Variable Description
UER_TIME	Refer to Table 1 for Variable Description
INVESTOR_ORIG_TIME	Refer to Table 1 for Variable Description
BALANCE_ORIG_TIME	Refer to Table 1 for Variable Description
FICO_ORIG_TIME	Refer to Table 1 for Variable Description
LTV_ORIG_TIME	Refer to Table 1 for Variable Description
INTEREST_RATE_ORIG_TIME	Refer to Table 1 for Variable Description
HPI_ORIG_TIME	Refer to Table 1 for Variable Description
STATUS_TIME	Refer to Table 1 for Variable Description
COUNT	The count of observation until it is declared as either class of 1 (default), 2 (payoff) and 0 (none/ not yet)

Table 4 *Summary of the Remaining Variables in the Final Dataset*

2.7 Data Partitioning

The original data (unbalanced) has been partitioned into 70% for training and 30% for testing since the whole balanced data is specifically for dimension reduction for logistic regression to provide an unbiased result. This partitioning will produce an imbalanced train data and a test data. The balancing of data might be required to the train data only (in order to make a comparison for the model performance between the balanced and imbalanced data). The test data might not require balancing to stick it to being realistic since there are seldom situations where the unknown to be predicted are always balanced and using the model to predict this imbalanced data could better access the performance of the models. The figure below shows the class distribution of the balanced train data.



```
|status_time |     n|     |status_time |     n|
|:-----------|-----:|     |:-----------|-----:|
|1           | 10611|     |2           | 20984|
|2           | 18613|     |1           | 20763|
```

Figure 14 *Class Distribution of the Imbalanced Train Data (left) and the Balanced Train Data (right)*

The table below summarizes the list of data frames allocated for balanced/ imbalanced and their respective training and testing data in order to avoid confusion of the data used for different purposes.

Data frame names	What the data is for
Mortgage	The original uncleaned and non-pre-processed data
Mortgage_new	The original data which is cleaned and pre-processed after dealing with missing values
Mortgage_balanced	The original balanced dataset which is solely used for the purpose of dimension reduction section

Mortgage_train_imbalanced	The imbalanced train data which is partitioned from the original data
Mortgage_test	The test data to be tested with the models from both balanced and imbalanced train data
Mortgage_train_balanced	The balanced train data balanced from the imbalanced train data

Table 5 *Summarization of the Data Frames Used*

3. Classification Models Selected

Given the purpose of identifying the best model to classify the both the testing data and the unknown class new data, classification models will be used. In this section, the preprocessed and cleaned data will be fed into four classifications machine learning algorithms: K- Nearest Neighbor, Classification Tree, and Logistic Regression. Modifications to the train and test data has been made respectively to tailor to the effectiveness of each model, and all the models are evaluated using the confusion matrix (for model accuracy, sensitivity and specificity measures), where the important class is set to "1" in order to prioritize the class 1 which is for the mortgage to be classified as "defaulted" and the Receiver Operating Characteristics (ROC) curve (for accessing the performance through classification thresholds). The randomization seed is set to 12345 in order to ensure consistent results for all models as well as their performance.

3.1 K- Nearest Neighbor

The first model selected is the most classic classification algorithm which assumes that the data points which distances are close to one another should belong to the same class and predictions to the new data are made based on this assumption. A series of action have been taken in order to fit a KNN model.

3.1.1 Model Mechanism

In order to perform KNN, first the irrelevant variables are to be removed. In this case, as far as the data looks, only the id column will be removed since it is irrelevant to the prediction of the future data, which ID will be indefinitely greater than those in the current data. All the remaining variables value are normalized in order to put the scale between 0 and 1 to optimize the model performance. While all values in the data are converted ranging from 0 to 1, this might not work for the target variable of status_time, which output has to be 1 for default and 2, and therefore, it is replaced with the original imbalanced data In the codes, the proportion class are checked from time to time in order to ensure that the data frame used are correct for each designated sections. The KNN models are trained with controls of "repeatedcv" (which is the repeated cross validation), number of iterations of five and the tune length of 10, and it works the same way for both balanced and imbalanced dataset

3.2 Classification Tree

Classification tree which is sometimes referred to as decision tree is a structural mapping of the categorical decisions of the data based on the predictors' rules and splits based on the purity of the data. It is yet another popular machine learning algorithm which the target variable comes to take a set of values with all probability of class stated in each node and known for classifying the new data based on a set of decision rules which represents the relationship exist in the data.

3.2.1 Model Mechanism

Before fitting a tree, first all categorical variables are converted to categorical and using the respective data frame name (balanced or imbalanced/ train or test), the model is fitted using the rpart function and it is then plot. Plot package from same package. The tree is then evaluated to the testing data, which is then pruned using the best cp value with the least xerror vale. All predicted values are evaluated using the confusion matrix.

3.3 Logistic Regression

The next model used for this case is the logistic regression model, which is another predictive algorithm which outcome variable is predicted based on the relationship between the predictors. The predicted outcome is in probability, which is then converted to the categorical class based on the cutoff value set where the value greater than one cutoff value is allocated as a certain class.

3.3.1 Model Mechanism

In order to fit the logistic regression model, again the categorical variables are first converted to categorical, and the model is built upon that. The option (scipen = 999) is used to avoid all scientific notations forcing full display of the numbers for better interpretation to the model. Similar to the previous models, it is first evaluated on the test data, both using the cutoff value of 0.5, which indicates the predicted probability greater than 0.5 will be categorized into class 2 payoff and those less than 0.5 will be in class 1 of being defaulted.

4. Classification Modelling (Original Imbalanced Data Versus Balanced Data)

All models for imbalanced data are made from the imbalanced data: mortgage_train_imbalanced, whereas all models for balanced data are made from the balanced data: mortgage_train_balanced.

4.1 MODEL 1 - K- Nearest Neighbor

The table on the next page shows a comparison table for both balanced and imbalanced data using different k values as suggested by the tuning of the model. The models have been tested to their respective train data in order to test if the model has been overfitted to the training data itself by comparing the metrics with that of the testing data's.

From the table below, it can be seen that the balanced data model (Model 2) originally has a higher accuracy and sensitivity

From the table, it can be seen that the Model 1 has a higher accuracy and specificity, but the Model 2 has a significant higher sensitivity rate of 80.4%. Generally speaking, both models are performing well on its own in terms of their own better performing measures, but in this case, given with similar accuracy rates, model 2 which has a better sensitivity rate would be chosen. The cost for not being able to predict if this borrower's mortgage will be defaulted is greater than not predicting the paid off ones.

Original Imbalanced Dataset **(Model 1)**				**Balanced Dataset** **(Model 2) ***			
K Value **(Tuning Suggested)**	**Accuracy**	**Sensitivity**	**Specificity**	**K Value** **(Tuning Suggested)**	**Accuracy**	**Sensitivity**	**Specificity**
23	0.8021	0.7155	0.8515	9	0.8283	0.8543	0.8025
23	0.7652 **(-0.0369)**	0.7207 **(+0.052)**	0.7905 **(-0.061)**	9	0.7275 **(-0.1008)**	0.8040 **(-0.0503)**	0.6838 **(-0.1187)**

Table 6 *Summary of K- Nearest Neighbor Performance Evaluation Using Confusion Matrix*

4.2 MODEL 2 - Classification Tree (Decision Tree)

The figures below show the plot for the two models and the table shown later summarizes the performance of the classification tree on both data then prune the tree for better model performance. According to the tree plots, both trees look similar with having LTV_time and time as the variables used to split but only with different values. The evaluation is similar to the KNN model where the accuracy and sensitivity rate is slightly better in Imbalanced Model (Model 3) and the sensitivity rate being higher in the Balanced Model (Model 4). The sensitivity difference here is larger with Model 4 having around 8% more than the Model 3. Again, for this algorithm, since the accuracy and sensitivity rate (especially the accuracy rate) are somewhat closer comparing the two, it would be recommended to select the model with the better sensitivity rate.

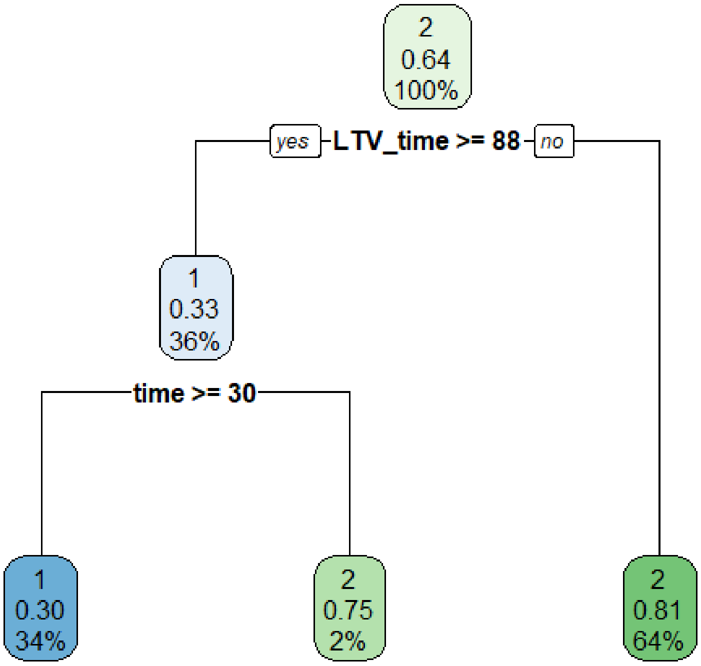


Figure 16 Classification Tree for Imbalanced Model

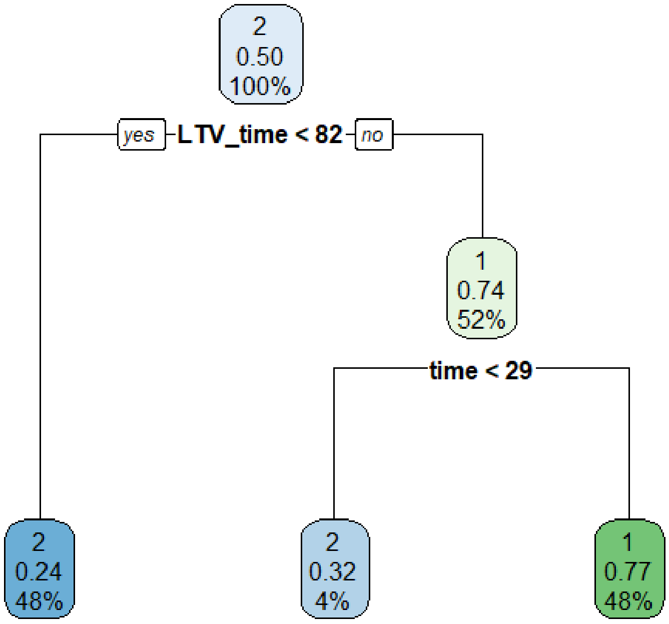


Figure 15 Classification Tree for Balanced Model

Original Imbalanced Dataset **(Model 3)**				**Balanced Dataset** **(Model 4) ***			
	Accuracy	**Sensitivity**	**Specificity**		**Accuracy**	**Sensitivity**	**Specificity**
Training Data	0.7740	0.6601	0.8389	Training Data	0.7643	0.7493	0.7792
Testing Data	0.7733 **(-0.0007)**	0.6721 **(+0.120)**	0.831 **(-0.007)**	Testing Data	0.764 **(-0.0003)**	0.7528 **(+0.0035)**	0.7704 **(-0.0088)**

Table 7 *Summary of Classification Tree Evaluation Using Confusion Matrix*

4.3 MODEL 3 - Logistic Regression

The table below (Table 8) summarizes the evaluation of the two models using the Logistic Regression. From the table, it can be seen that surprisingly the imbalanced model is performing better than the balanced model, with the reason being that the balanced model (Model 6) is performing exceptionally low with all measures around 20%. Without any doubt, Model 5 will be selected.

Based on what have been evaluated, the models from each algorithm which have better performance are: **Model 2 (Balanced Model For KNN)**, **Model 4 (Balanced Model for Classification Tree)** and **Model 5 (Imbalanced Model for Logistic Regression)**. These three models will then be proceeded for final evaluation to select which model to be deployed for this case.

Original Imbalanced Dataset **(Model 5)**				**Balanced Dataset** **(Model 6) ***			
	Accuracy	**Sensitivity**	**Specificity**		**Accuracy**	**Sensitivity**	**Specificity**
Training Data	0.7800	0.6270	0.6872	Training Data	0.2296	0.2459	0.2134
Testing Data	0.7751 **(-0.0049)**	0.6722 **(+0.0452)**	0.8595 **(+0.1723)**	Testing Data	0.2283 **(-0.0013)**	0.243 **(-0.0016)**	0.2199 **(+0.0065)**

Table 8 *Summary of Logistic Regression Evaluation Using Confusion Matrix*

5. Final Evaluation and Model Selection

These chosen models which are selected above are then tested with the original data, but this time not from the partitions that were created. These models are tested using the “rand” data frame that was created, which consists of the first observations of the borrower’s ID which was later declared as defaulted and payoff, or generally speaking, the first observations of the borrower ID which later become either class 1 or 2 (by removing all those undeclared borrower ID observations). This process gives another step of reassure the performance of the model before predicting the unknown outcome observations. The results performance evaluation using the confusion matrix is shown in the Table 9 below. (The green highlight indicates the highest measure among each category of measurements). Same alterations and changes have been done to the new data in order to fit the model and compare them on an equal scale.

	Model 2 KNN	Model 4 Tree	Model 5 Logistic Regression
Accuracy	0.6873	0.683	0.6671
Sensitivity	0.4175	0.22707	0.7335
Specificity	0.8410	0.94283	0.6292

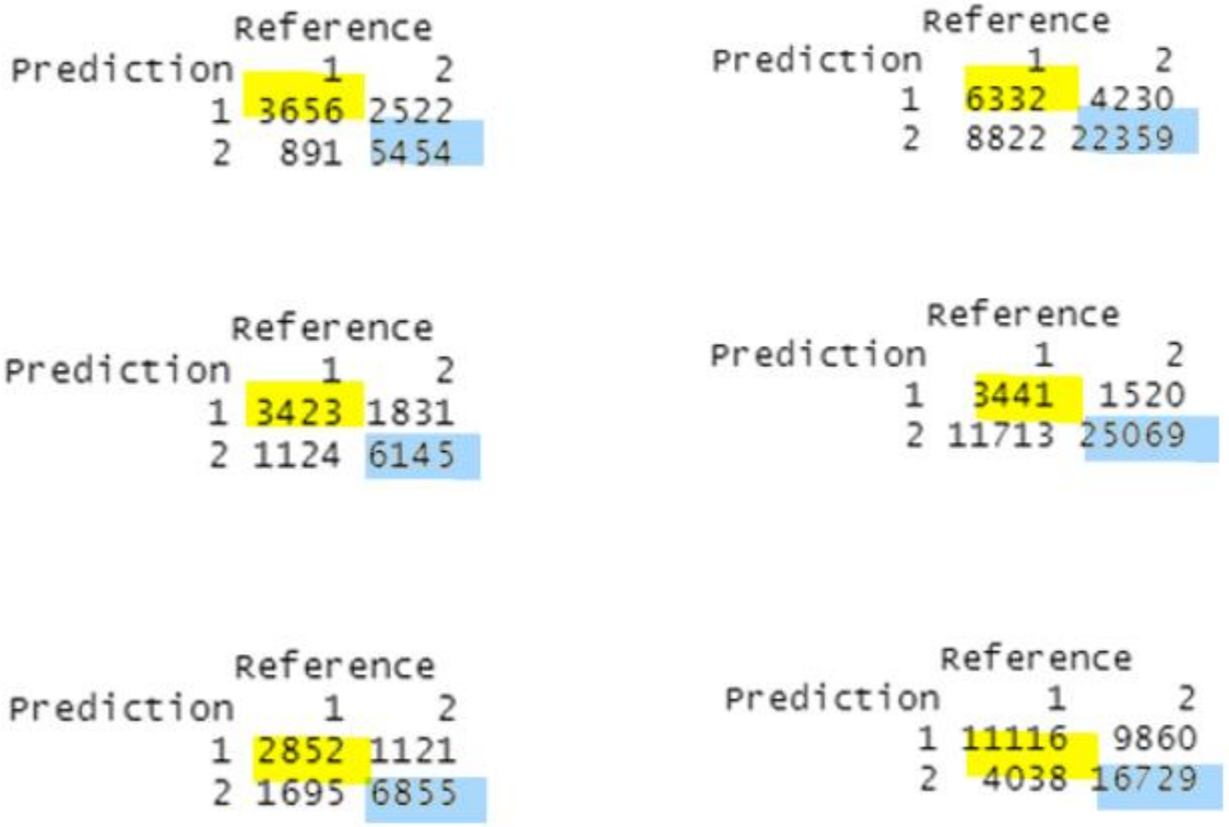
Table 9 *Performance Evaluation Summary (Tested with the first observation of the Original Declared Borrower ID)*

5.1 Evaluation Metrics – Confusion Matrix Measures

The Table 10 on the next page summarizes the performance evaluation of the models tested with the testing data and the first observations data in order to form a comparison. Along with that, the confusion matrices have been attached below for better reference to the measures listed in Table 10.

	Model 2 KNN		Model 4 Classification Tree		Model 5 Logistic Regression	
Performance Measures	Testing Data	First Obs Data	Testing Data	First Obs Data	Testing Data	First Obs Data
Accuracy	0.728	0.687 **(-0.040)**	0.764	0.683 **(-0.081)**	0.775	0.667 **(-0.108)**
Sensitivity	0.804	0.418 **(-0.386)**	0.753	0.227 **(-0.526)**	0.627	0.734 **(+0.107)**
Specificity	0.683	0.841 **(+0.158)**	0.770	0.942 **(+0.172)**	0.860	0.629 **(-0.231)**
Precision$\frac{TP}{(FP+TP)}$	0.592	0.221 **(-0.371)**	0.652	0.694 **(+0.042)**	0.718	0.530 **(-0.188)**
Type I Error (FP)	891 (7.115%)	8822 (21.134%)	1124 (8.985%)	11713 (28.060%)	1695 (13.54%)	4038 (9.673%)
Type II Error (FN)	2522 (20.139%)	4230 (10.133%)	1831 (14.621%)	1520 (3.641%)	1121 (8.95%)	9860 (23.621%)

***Table 10** Whole Summary for Model Performance in Both Tests (All Rounded to 3 decimal places)*



```
          Reference
Prediction    1    2
         1 3656 2522
         2  891 5454
```

```
          Reference
Prediction     1     2
         1  6332  4230
         2  8822 22359
```

```
          Reference
Prediction    1    2
         1 3423 1831
         2 1124 6145
```

```
          Reference
Prediction     1     2
         1  3441  1520
         2 11713 25069
```

```
          Reference
Prediction    1    2
         1 2852 1121
         2 1695 6855
```

```
          Reference
Prediction     1     2
         1 11116  9860
         2  4038 16729
```

Figure 17 *Confusion Matrices for All Model Performance Tested*

<u>Note for Figure 17</u>

First pair of Confusion matrix → KNN model (Model 2): Testing Data (left) Versus First Observations Only Data (right)

Second pair of Confusion matrix → Classification Model (Model 4): Testing Data (left) Versus First Observations Only Data (right)'

Third pair of Confusion matrix → Logistic Regression (Model 5): Testing Data (left) Versus First Observations Only Data (right)

Accuracy is not the only measure that has been taken into consideration when selecting which model has the best performance. While accuracy is often the most straightforward method to access the overall performance of the model since it indicates the proportion of the classes being correctly classified, however, it does not consider of which class has more accurate figures and whether that class is of the important class deemed for the business purpose or not. Therefore, the analysis will be based on a more variety of all other metrics in the classification model in the Confusion Matrix. With that being said, although the Classification Tree Model 4 has the highest accuracy among all, it might not be selected as the final model which has the best performance. Its accuracy decreased when tested with the original data with the first observations by 8.1%, which is relatively fine compared to Model 5 Logistic Regression models which has a greater percentage of 10.8%, but it can be seen that the Model 2 KNN has the least percentage drop of 4.2% So at the case of accuracy, Model 2 KNN could have an upper hand.

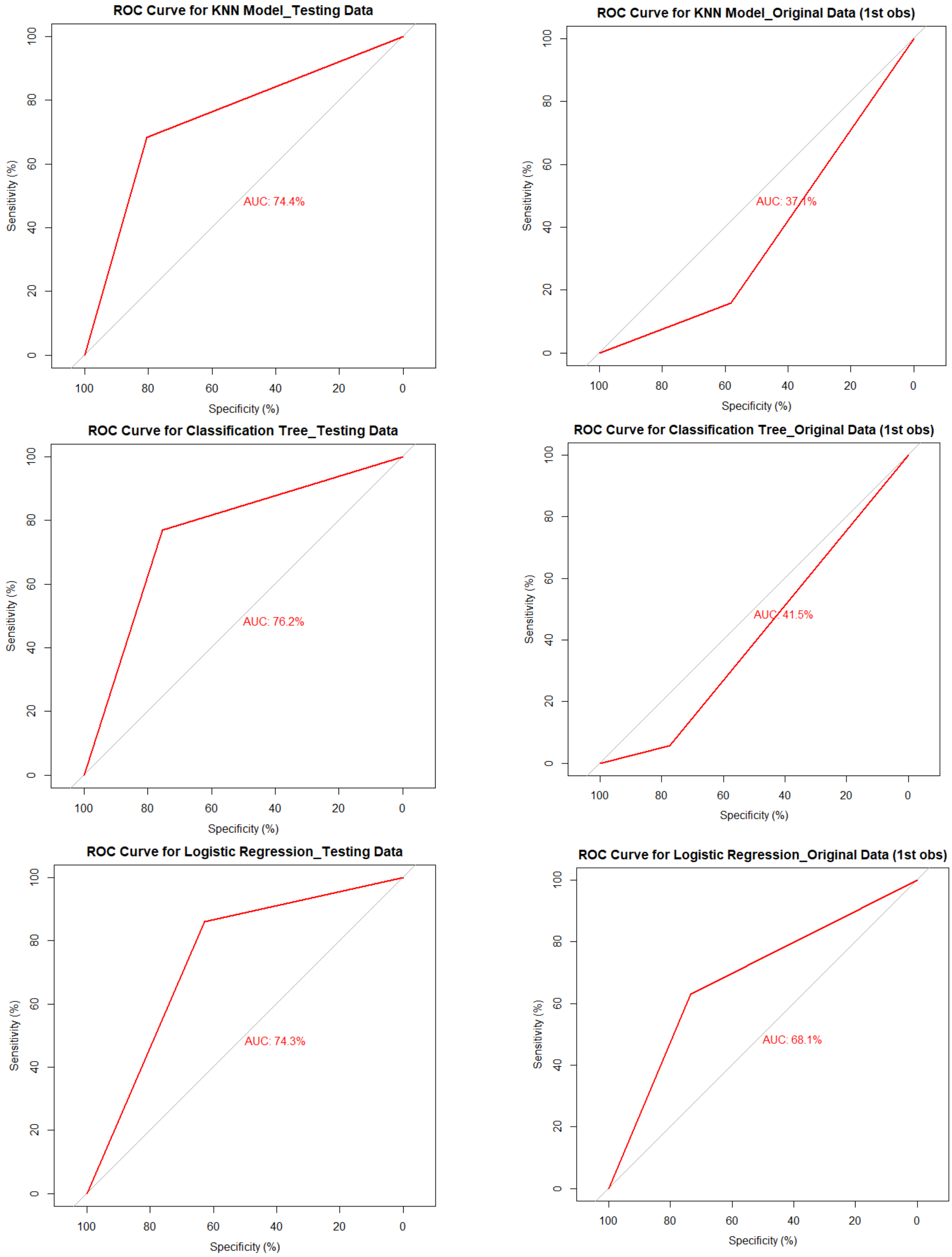
Sensitivity represetns how well the model is doing in terms of accuracy of the more important class, in this case, it would deemed to be the Class 1 (which is the Defaulted Cases), since borrowees might want to ensure that the default risk is low to prevent borrowers not paying what is requried to be paid for their mortgage payments. In this case, it can be seen that Model 2 has the highest sensitivity among all for the testing data, but when tested with the original data with their respective first observations, its sensitivity drastically drop to 41.8% (which is 38.6% decrease from the testing data performance). Similar situation can be seen in Model 4 Classification Tree model where there is a decrease of 52.6%, left with sensitivity rate of as low as 22.7%. Notable chanages can be observed in Model 5 Logisitc Regression, where the sensitivity rate increases from 62.7% in the testing to 73.4%, which is a 10.8% increase. At this point with the accuracy measure, Model 5 Logisitc Regression could have an upper hand.

Specificity represents how well the model is performing in terms of the accuracy of the other class, which is the Class 2 (which is the Class of the mortgages being paid off). Since the other class was considered as the more important class, the importance of the specificity is not that emphasized. It is not to mention that among the three models, Model 4 Classification Tree has the highest rate with 94.2% when tested in the original data with first observations (which has an increase of 17.2%) surpassing the Model 5 Logistic Regression which has a highest rate in the testing data, but was reduced to 62.9% when tested on the original data.

Other measures that were considered are the precision rate, Type I error, and Type II error. Precision rate indicates when predicted 1, how often it is correctly predicted, Type I error is wrongly predicted as positive class where it is actually a negative class, and the Type II error is wrongly predicted as negative class where it is actually a positive class. At the case of Precision, it can be seen that Model 4 Classification Tree has the best measures with an increase of 4.2% where all the other models have decreased rate with Model 2 KNN having 37.1% decrease and Model 5 Logistic Regression having 18.8%. At this point with the Type I and Type II error, more emphasis will be placed on the Type II error based on the similar assumption with the sensitivity rate. There would incur a greater cost for the borrowee to incorrectly predict the borrower to be paid off then it was defaulted then it is to incorrectly predict a default borrower which is later paid off. The Model 5 Logistic Regression definitely has the lowest rate of the Type I error rate which is worth to mention. Speaking of the Type II error rate, it can be seen that the Model 4 Classification Tree has the highest measure with lower than 10% decrease of error rate when testing the model on the original data. Model 5 Logistic Regression in this case, has an alarming result which increased its type II error rate from 8.95% to 23.621%.

5.2 Evaluation Metric – ROC Curve

The ROC Curve has also been used in order to analyze the Area Under Curve (AUC). ROC, standing for Receiver Operating Characteristics, is often used to show the trade-off between the sensitivity and specificity of the model and the Area Under the Curve (AUC) is the measure of the usefulness of a test where greater percentage of AUC indicates a better model. The straight line in these graphs represent the performance of the random model classifier, and it is a straight line indicating that 50% correct predictions with AUC of 50% where True Positive rate is equal to the False Positive Rate. This is used to compared with the model since random model is at the middle so any model above that is better.



ROC Curve for KNN Model_Testing Data

ROC Curve for KNN Model_Original Data (1st obs)

ROC Curve for Classification Tree_Testing Data

ROC Curve for Classification Tree_Original Data (1st obs)

ROC Curve for Logistic Regression_Testing Data

ROC Curve for Logistic Regression_Original Data (1st obs)

Figure 18 ROC Curve for All Models Tested

From the ROC Curve, it can be seen that originally, Model 4 Classification Tree has the highest AUC of 76.2%, but when tested with the original data, it has decreased to 41.5% which indeed is not what a good model will produce. Similar patter can be seen for the Model 2 KNN which has the second highest AUC of 74.4% but it was decreased to 37.1% when tested with the original data. However, different from these two cases, the Model 5 Logistic Regression which also has a decrease in the AUC from 74.3% to 68.1%, but the decrease is merely 6.2% decrease compared to the decrease of the other two models which has more than 30% decreases. At this case, the Model 5 Logistic Regression has the best measures in terms of the Area Under Curve (AUC) measures.

To conclude the analysis section, it can be assumed that the **Model 5 Logistic Regression** produces an overall best performing model. It has the best Area Under Curve (AUC) and better sensitivity rate when testing the model to the data with only the first observations. Since the more important class is the class 1 defaulted with the reasonings from above, sensitivity and the percentage of the type II error will be prioritized and although Model 5 has a greatest sensitivity rate being the one model with the sensitivity increase, its type II error is the highest among three. This could be the only big drawback of this model and it has to be considered after predicting the unknown data, especially it could come to a need of assistance from another model.

6. Results

Model 5 which have been selected in the previous section has been used to predict the outcome of the unknown borrower ID, those which has not been identified at the time of the dataset collection. The unknown observations data has been subset to only the first observations for each borrower ID. The graph below shows the distribution of the predicted outcome of these unknown data, and it can be seen that the majority of these borrowers' mortgage are predicted to be defaulted.

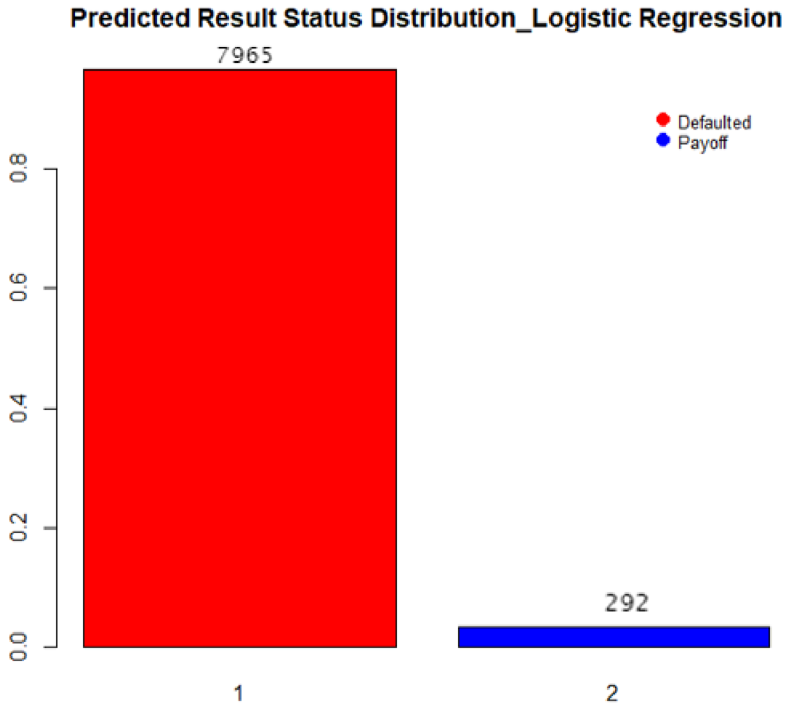


Figure 19 *Predicted Result Status Distribution Using Model 5 (Logistic Regression)*

While Model 5 sounds like the best performing model upon checking on the performance measure metrics, when it is used to predict the outcome for these unknown data, the results seem highly unrealistic with the defaulted mortgages outnumbering the paid off ones which in real life is not that case. The delinquency rate of the United States in 2022 is only as high as around 5% of the total mortgage loan issued, but the model predicted the opposite way. Therefore, in order to present another scenario to ensure if that is the case with the given dataset, the second-best performing model which is the Model 4 Classification Tree. The figure illustrates the distribution of the outcomes. From there, it can be seen that there is a more reasonable distribution of the outcomes where the mortgages paid off occupies around 75% of the total unknown mortgages, where the defaulted mortgages occupy the remaining approximately 25%. At this point, Model 4 has a more reasonable predicted outcome compared to the Model 5 which have higher performance measures but unreasonable predicted outcomes.

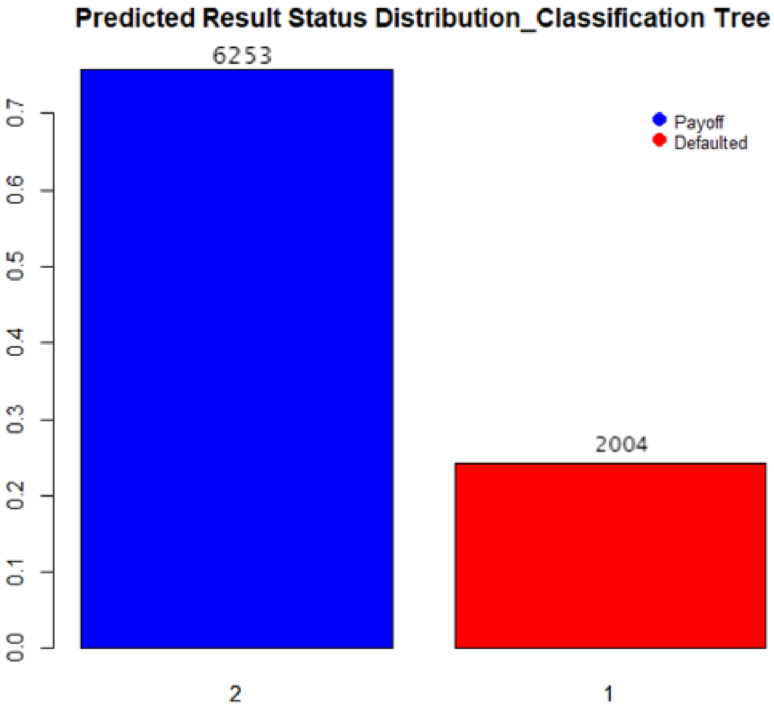
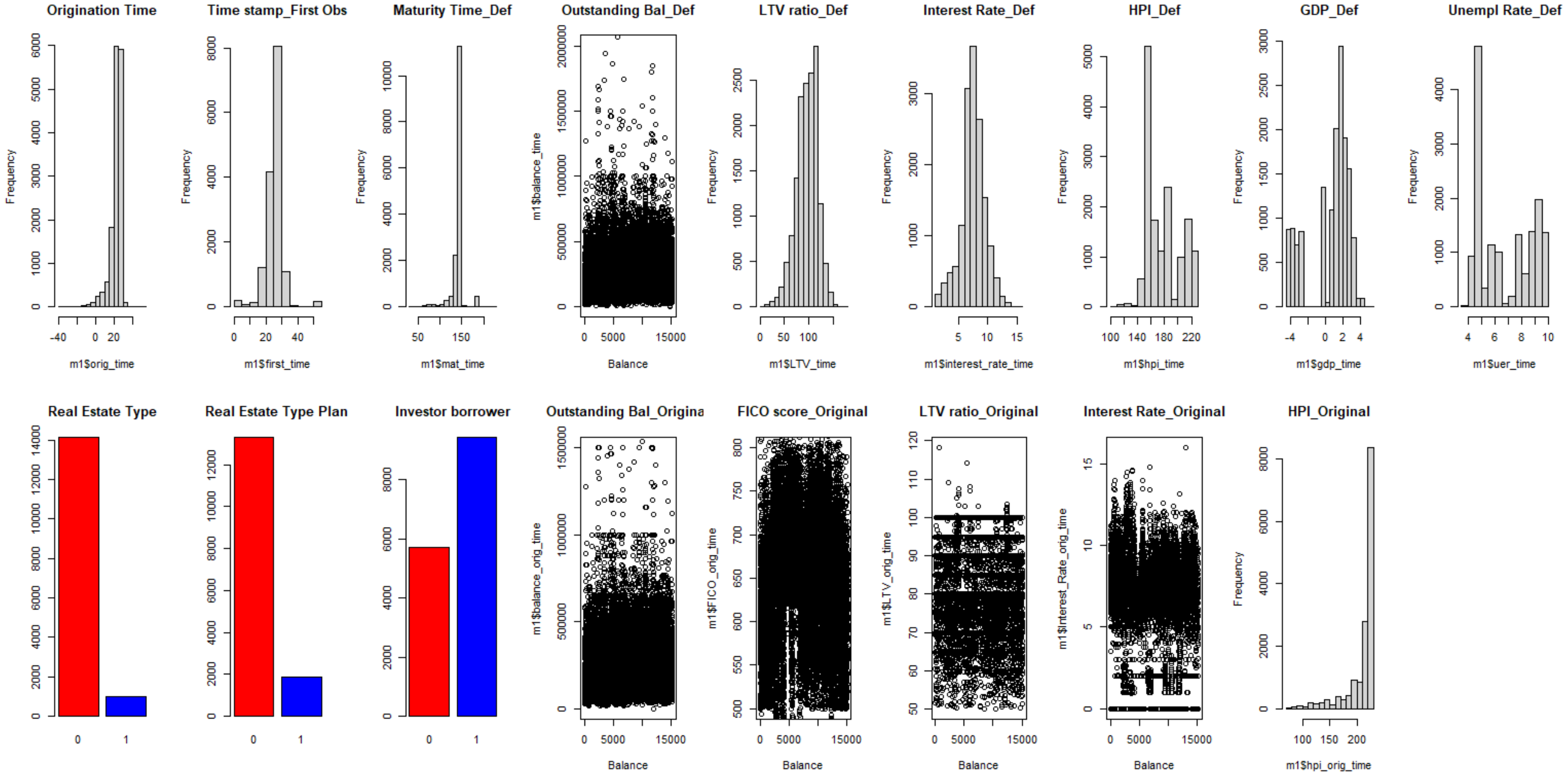


Figure 20 *Predicted Result Status Distribution Using Model 4 (Classification Tree)*

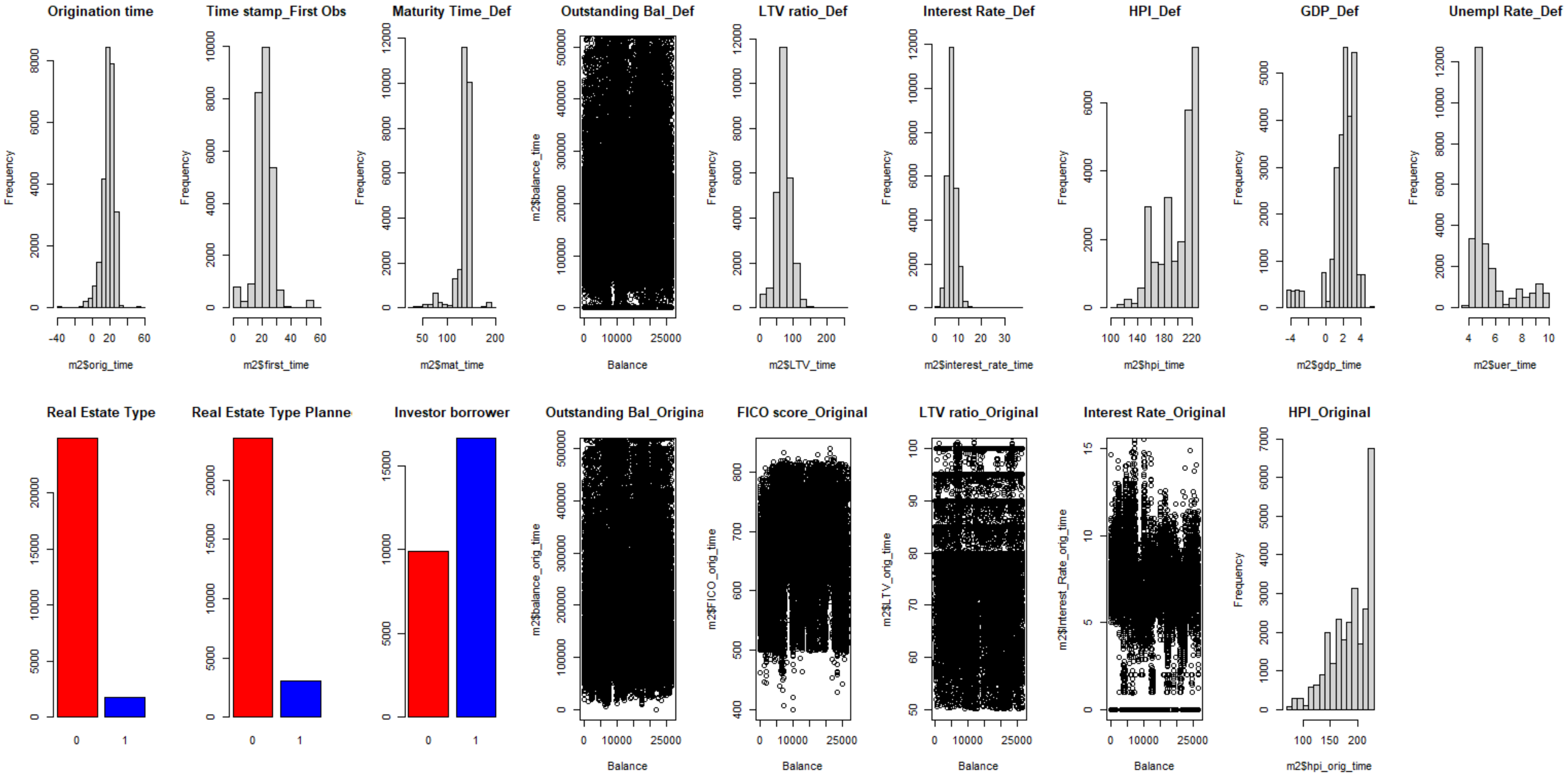
7. Conclusion

To conclude, although **Model 5 Logistic Regression** has the best cumulative performance measure metrics, when it is used to predict the unknown data, it has an unreasonable result which makes it seem not as reliable as the performance metrics shows. Surprisingly, when the Model 4 Classification Tree is used to predict the same unknown data, the results seem more realistic and reasonable. Therefore, at this point, it is hard to determine which model will give the most accurate outcome of these mortgage loans unless the actual outcome of these mortgage loans can be compared upon. This will give another validation to the performance of these models.

8. Appendix



__Appendix 1-__ Summary Graphs for Defaulted Observations (At the Timestamp of being Defaulted)



Appendix 2- Summary Graphs for Paid Off Observations (At the Timestamp of being Paid Off)

END OF CASE 2

R Codes for Case Study 2

Case Study 3

Human Resources Analysis

By Khin Thu Zar Thant

Webster University George Herbert Walker School of Business and Technology

CSDA 6010 Analytics Practicum

16th December 2022

Executive Summary

The case study aims to prove the hypotheses related to the data set provided which is the Human Resources dataset. The dataset includes the individual employees' variables which will be used to assess both the hypotheses and fit the models. This paper will use a substantial number of visualizations in order to plot the trend for better descriptive analysis to explore why employees are leaving the company and identify the potential reasons in the human resources of the company in hope to possibly recommend for a better strategy for the company to retain its employees and reduce the turnover rate.

The general procedure for this case study will be as follow. First, the missing data will be dealt with after identifying the business problem, project goals and the dataset briefing. The data preparation and analysis section will be proceeded with variable and dataset manipulations and data exploration take place. Exploratory Data Analysis (EDA) will occupy most space in this paper to test the hypotheses supported by their respective analysis through visualizations. Fitting the models with the data partitions will be followed and finally selecting the best model among all the models tested through evaluations using the performance metric measures.

Table of Contents

List of Figures

List of Tables

1. Introduction

Employees are the company's best assets. Companies have been constantly trying to find ways to retain talented employees and minimize the employee turnover rate. At this time, analytics holds a vital tool in order to analyze both the shortcomings and the strengths of the companies' regulation and human resources style and a basis on the possible alteration that are required to be initiated for potential human resource functionalities. Analytics can not only help identify the trend from the employees that have left but also provide answers regarding to the HR problems, which for instance which department has the most employee turnover rate and who is most likely to leave next. Being equipped with this aspect of HR analytics, employers can take steps to prevent 'high risk' employee groups from leaving the company and seek for working elsewhere especially in the competitors' company.

1.1 Business Problem

Companies, more than ever, have been more engaged in retaining talented and productive employees, which they believe are the key to surviving the highly competitive market in their respective industries. For these reasons, there is more analytic work done to understand their employees more and analyze the reasons for those employees who have left in order to implement preventive measures in the future. Retaining their best employees will be extended as a long-term strategy which can impact the company's growth and reduce risks for them in case employees left for the competitors' companies.

1.2 Project Goal

In this case study, the goal will be to identify the trends underlying in the dataset with a purpose of outlining the problems and gain insights into these issues which affect the human resources and create predictive models to help companies to predict if the employees are likely to leave or stay. In addition to that, based on the insights drawn from this data exploration, HR strategies recommendations will also be made, as well as testing out a few hypotheses which are as follows.

(i) Hypothesis 1: Salary is the reason the employees left the company.

(ii) Hypothesis 2: Employees leave the company because work is not safe.

(iii) Hypothesis 3: Is this company a good place to grow professionally?

1.3 Dataset Briefings

The dataset contains a total of 14,999 instances where each row represents an employee, and ten variables. The variable descriptions are as follows.

VARIABLE NAME	DESCRIPTION
SATISFACTION LEVEL	Satisfaction level of the employees (ranging from 0 to 1)
LAST_EVALUTION	Latest evaluation (evaluations conducted yearly)
NUMBER_PROJECT	Number of projects worked on
AVERAGE_MONTHLU_HOURS	Average monthly hours
TIME_SPEND_COMPANY	Time spent at the company (in years)

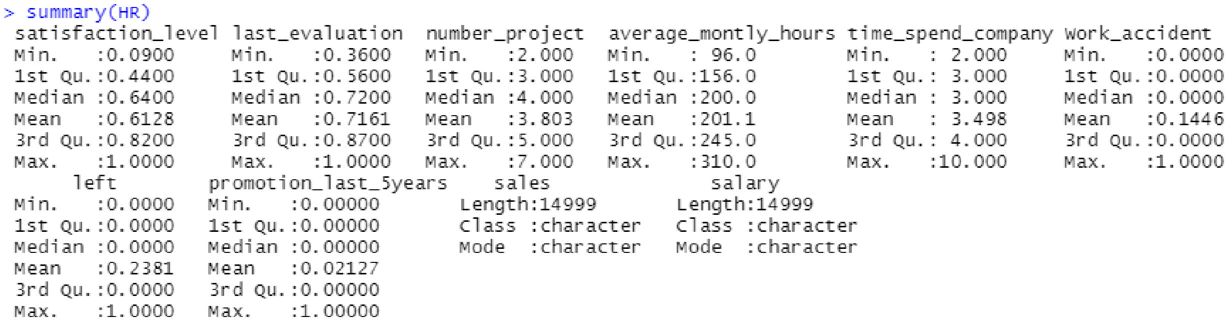
VARIABLE NAME	DESCRIPTION
WORK ACCIDENT	Whether the employees have any work accidents (within the last 2 years)
LEFT	Whether the employees have left the job
PROMOTION_LAST_5YERS	Whether the employees have had promotions in the last 5 years
SALES	Department names
SALARY	The salary for each employee

***Table 1** Variables Description for Human Resource Dataset*

2. Data Preparation

2.1 Data Preprocessing

The summary of the data is checked to analyze for required data preprocessing. The figure below attaches the summary of the whole data set with all the variables. Since there are only ten variables, all variables appear to be relevant to the data, so no initial removal of the variables seems necessary.

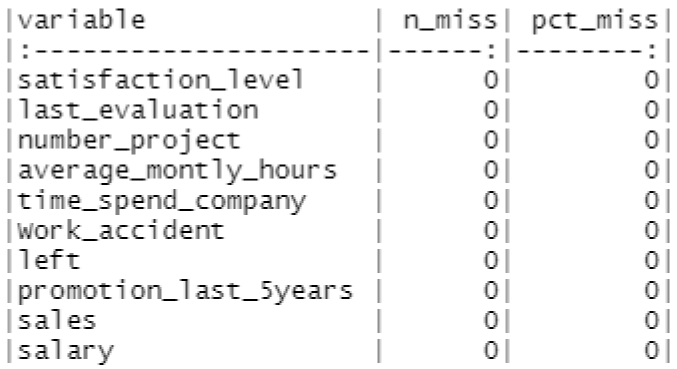


```
> summary(HR)
 satisfaction_level last_evaluation  number_project  average_montly_hours time_spend_company Work_accident
 Min.   :0.0900     Min.   :0.3600   Min.   :2.000   Min.   : 96.0        Min.   : 2.000     Min.   :0.0000
 1st Qu.:0.4400     1st Qu.:0.5600   1st Qu.:3.000   1st Qu.:156.0        1st Qu.: 3.000     1st Qu.:0.0000
 Median :0.6400     Median :0.7200   Median :4.000   Median :200.0        Median : 3.000     Median :0.0000
 Mean   :0.6128     Mean   :0.7161   Mean   :3.803   Mean   :201.1        Mean   : 3.498     Mean   :0.1446
 3rd Qu.:0.8200     3rd Qu.:0.8700   3rd Qu.:5.000   3rd Qu.:245.0        3rd Qu.: 4.000     3rd Qu.:0.0000
 Max.   :1.0000     Max.   :1.0000   Max.   :7.000   Max.   :310.0        Max.   :10.000     Max.   :1.0000
      left         promotion_last_5years    sales              salary
 Min.   :0.0000   Min.   :0.00000       Length:14999       Length:14999
 1st Qu.:0.0000   1st Qu.:0.00000       Class :character   Class :character
 Median :0.0000   Median :0.00000       Mode  :character   Mode  :character
 Mean   :0.2381   Mean   :0.02127
 3rd Qu.:0.0000   3rd Qu.:0.00000
 Max.   :1.0000   Max.   :1.00000
```

***Figure 1** Summary of the Variables in the Dataset*

2.2 Dealing with Missing and Null Values

The missing values and the null values are checked before moving on to the data alteration sections. There are not any missing values in the dataset as well as for the null values. The figure below shows the table for the missing values for all the variables, and it can be seen that there are zeros in all columns. This means that there will be nothing to be dealt with in this section.



```
|variable              | n_miss| pct_miss|
|:---------------------|------:|--------:|
|satisfaction_level    |      0|        0|
|last_evaluation       |      0|        0|
|number_project        |      0|        0|
|average_montly_hours  |      0|        0|
|time_spend_company    |      0|        0|
|work_accident         |      0|        0|
|left                  |      0|        0|
|promotion_last_5years |      0|        0|
|sales                 |      0|        0|
|salary                |      0|        0|
```

Figure 2 *Summary Table for the Missing Values*

2.3 Data Alteration and Variable Manipulation

Upon checking on the first and last few records of the dataset, it can be seen that in the "Sales" variable columns, there are other department names such as accounting, HR and technical so the variable name is not relevant to its content. Therefore, it has been renamed to "Department" instead.

3. Exploratory Data Analysis (EDA)

3.1 General Attribute Analysis

To have an overall idea of the data, general attributes analysis is conducted analyzing the individual variable on its own. There are different methods to get to this with the numeric and non-numeric variables, where the numeric variables such as the satisfaction level and last evaluation are assessed using boxplots and histograms whereas the non-numeric variables (also the categorical variables) such as the work accident and department are assessed using the bar graphs.

Frist, the distribution of the target variable "left" have been analyzed and a bar graph is used to visualize. From the graph, the dataset contains more class 0 which is the "not left" group represented by the red column, which the class 1 of class "left" has more than three times of it. Since this project heavily reflects on the originality of the dataset, imbalance dataset is not a problem as it sticks to the real- life situation of the company which the dataset represents. This might later be balanced when modelling using the classification predictive models.

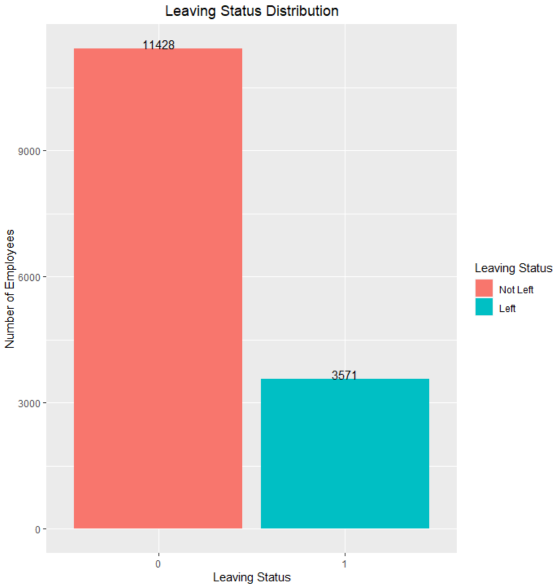


Figure 3 *Leaving Status Distribution*

Boxplots are then used to detect potential outliers lying in the data and it is clearly seen that there are not any visible outliers except for the "Years in Company," which is relevant given that in a workplace, there could be a few seniors who have worked long enough for the company.

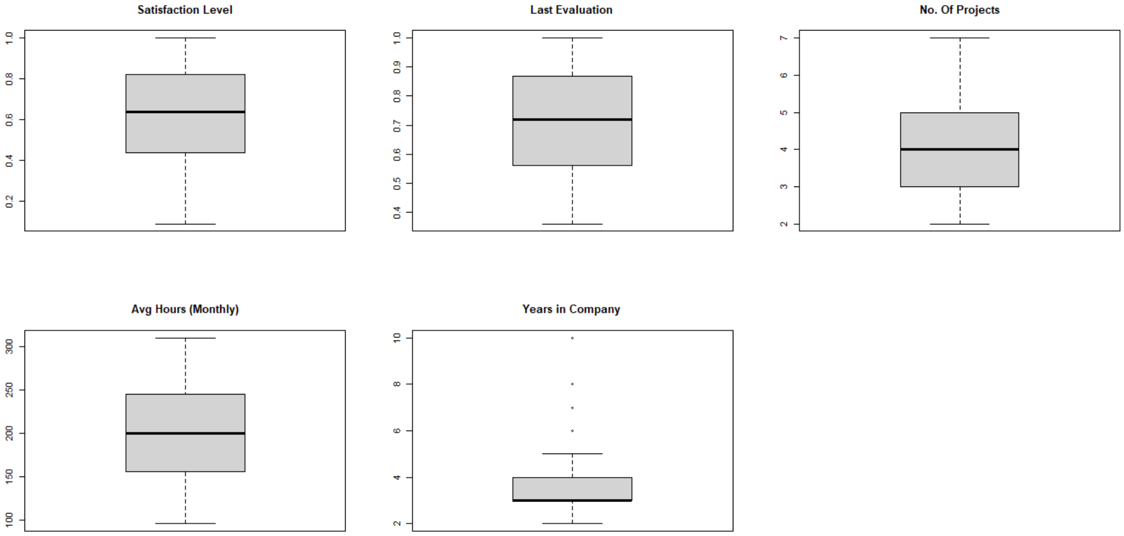


Figure 4 *Boxplot for Numeric Variables*

The figure below shows the histograms for these numeric variables. It can be seen that overall; most employees have satisfaction higher than 0.5 (from the scale of 0 to 1). As for the remaining variables, most employees lie around the lower values especially with the number of projects and the years in company. From these alone, these points can be denoted. First, most employees are satisfied with their job with lower evaluation scores, fewer projects and hours worked. Another interesting point here is that among these 14,999 employees, it mostly contains newer employees who only have joined the company for 2 years or less. These points could be more interesting to explore when compared against the employees who have left and not left.

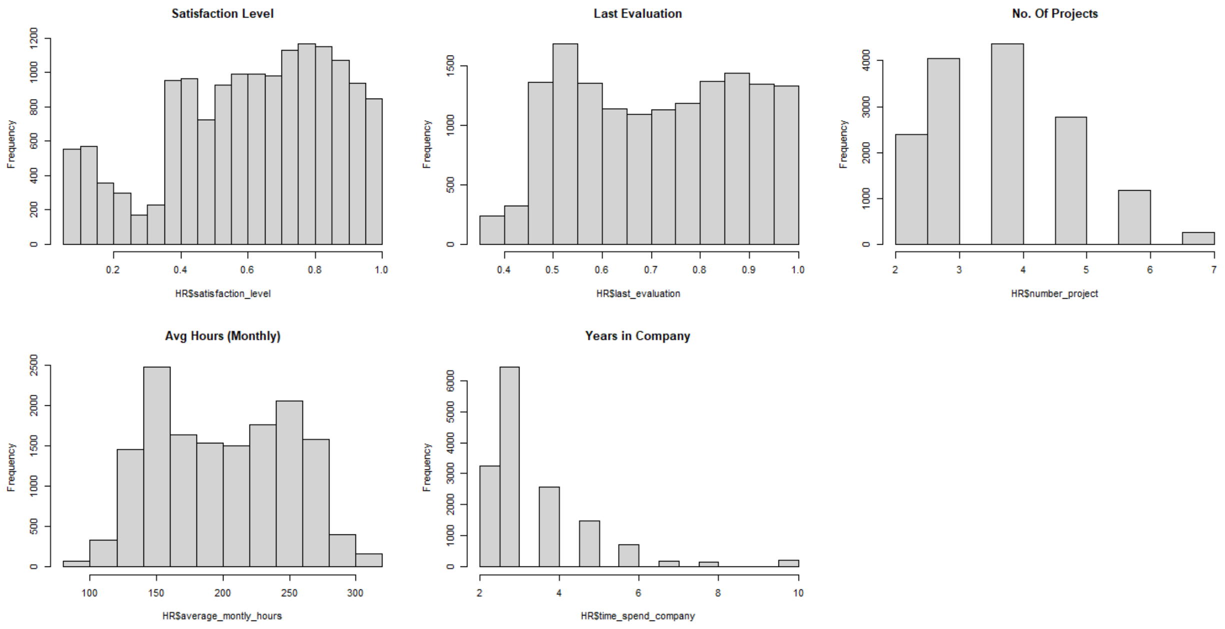


Figure 5 *Histograms for Numeric Variables*

For the non-numeric variables, compiled bar graphs are used to visualize the trend underlying. From the graph shown later, the sales department has the most employees, followed by the technical and support employees. It can also be seen that there are few or almost no work accidents and promotions happening in the company. Another point to denote is that the company contains employees with mostly low and medium pay employees. These factors would also be interesting to explore moving forward to compare the trends in left versus those who have not left.

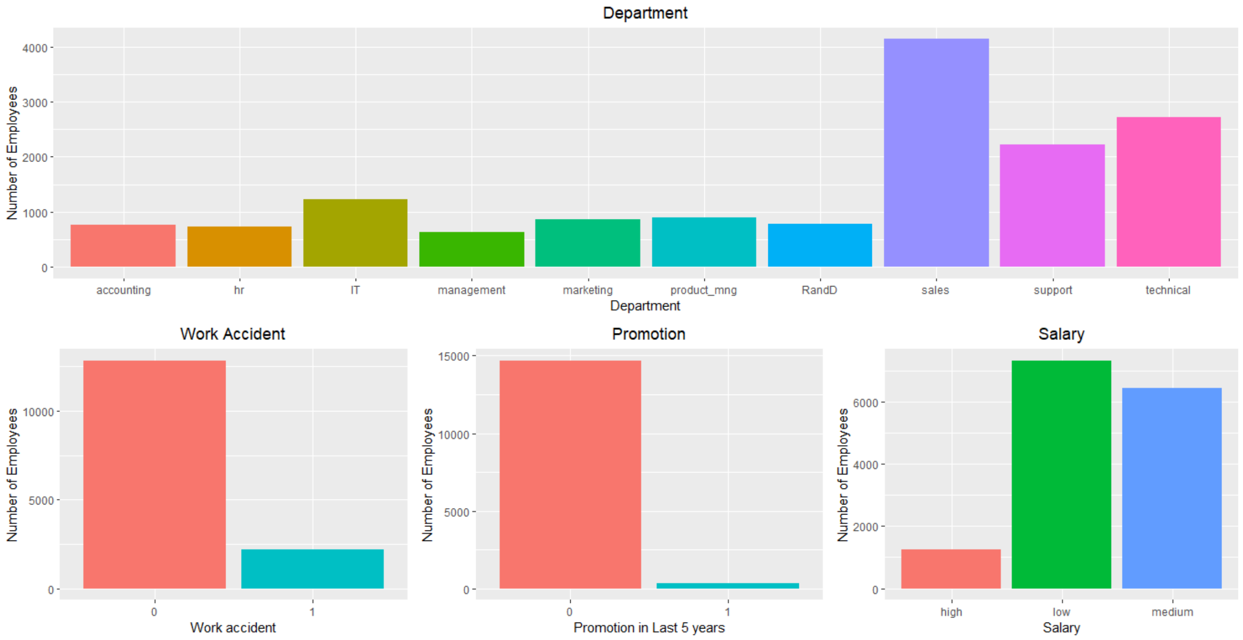


Figure 6 *Bar Graphs for Non- Numeric Variables*

3.2 Attribute Analysis with Target Variables

This section emphasizes the comparison between the two classes (i.e., left, and not left) in all the variables by dividing the numeric and non-numeric variables for plotting purposes.

3.2.1. *Numeric Variables*

The plot below summarizes the comparisons of the two classes for the numeric variables: satisfaction level, last evaluation, number of projects, average hours worked monthly, and the time spent in the company. The red bars represent the "left" class, the blue represents the "not left" class whereas the overlapped color is represented with their overlapped color, purple color.

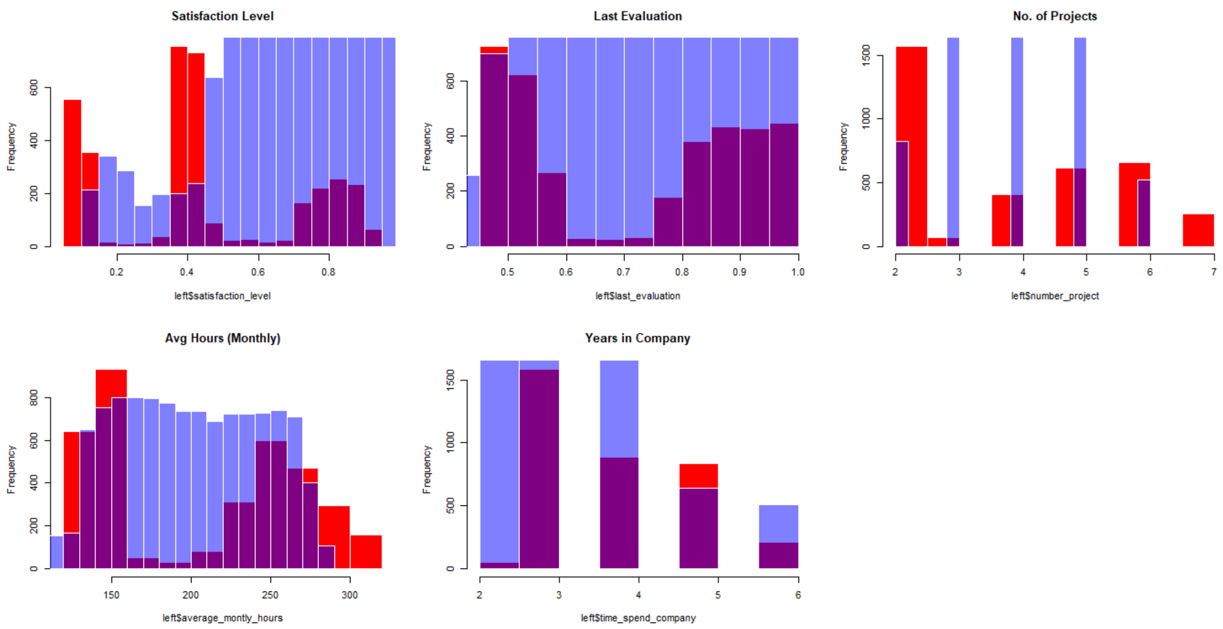


Figure 7 *Histogram for Numeric Variables with Class Comparisons*

By seeing these histograms, it can be seen that there are many overlapping areas in most of the variables, except for the number of projects variables which only had some overlapping areas in project numbers of 2, 3, 4, 5 and 6. For this variable, it can be seen that left class (with the red bars) is mostly populated in the fewer project numbers whereas the “not left” class is significantly populated in the project numbers of 3, 4 and 5, and most interestingly, there are a few employees who left having done 7 projects. At this point, it is hard to conclude if the number of projects really mattered when employees leave the company.

Speaking of the variables which have more overlaps, for the satisfaction level, most employees who left the company have lower satisfaction levels (mostly populated around 0.2 to 0.4) whereas the remaining employees in the company have higher satisfaction levels (from somewhere around 0.5 to 1). However, although this is the case with the majority, there are some employees who left with high satisfaction (looking at the purple section of the histogram) and likewise to the employees who have not yet left the company (looking at some blue areas in the lower satisfaction levels). Similar case with the last evaluation variable where we can see the employees who have left the company having varying evaluation scores, having fairly distributed score at the two edges (0.5 to 0.6 and 0.8 to 1) with the employees who have not left occupying fairly equally across the scores from 0.5 to 1. Likewise, it is hard to say whether it is those with lower scores who leave the company or those with higher scores. This variable should be explored in a greater depth for the company to assure that they are or are not retaining the better performing employees since this is the core objectives in every company to reduce turnover of the better performing staffs to reduce the risk of them going for their competitors.'

The next two variables have slightly different trends. For the average hours worked on a monthly basis, it can be seen that among those who left the company, most of them are populated in the higher hours' region of around 210 to 300 hours with a few of them (around nine hundreds of them) peaking at the 140 to 150 hours monthly. Interesting denotes could be made in what type of employees are leaving the company, to identify if the peak in the 140 to 150 hours indicate that more part time employees are leaving or other speculations like having most of the employees leaving at higher numbers of hours meaning they have more overtime. Being said that, however, it has remaining employees occupying across the hours of 150 to around 270 which could also mean that there are still employees who remains in the company after being worked overtime. Generally speaking, the common working hours per week would be 160 and any numbers higher than that could be considered as overtime and a distinction could be seen in the histogram that their visible red blocks (indicating those employees who left) in the hours of 290 and 310 which could be considered as excessive overtime hours. This could play a huge role in determining if the amount of overtime is scaring the employees away.

Last but not least, for the years variable, it can be seen that among those who left, the majority of them left after 2.5 to 3 years in the company followed by 4, 5 and 6 years. For those remaining, it can be seen that they are populated in 2 to 3 years and 3.5 to 4 years. A general trend in this attribute is that there are fewer old employees in the company as they stayed longer in the company, and most employees (either left or stay) are populated in the fewer number of years. This could indicate that the company might not be a good place to settle in and most employees left after staying for 3 to 4 years or less. This point could be further elaborated in future sections.

3.2.2. *Non- numeric Variables*

The plot below summarizes the comparisons of the two classes for the non- numeric variables: work accident, promotion last 5 years, department and salary. The red bar represents the “not left” class (Class 0) whereas the blue bar represents the “left” class (Class 1).

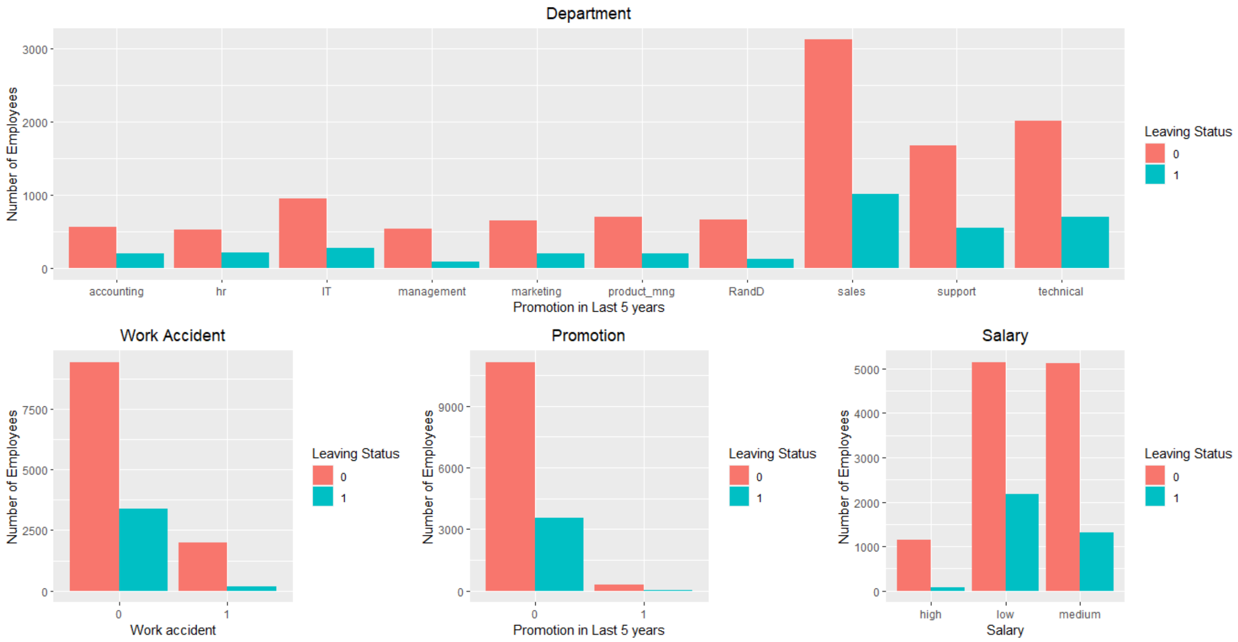


Figure 8 *Bar Graphs for Non- Numeric Variables with Class Comparisons*

First of all, for the department attribute, it can be seen that there are greater numbers of employees' turnover in departments with more employees which are the sales, technical and support departments, with the turnover rate more or less low than half of those remaining. Overall, the sales department has the highest turnover, and the management department has the lowest turnover. Speaking of the next variable in the plot, which is the work accident, among the employees who left, there are fewer employees who had work accidents at work so it might not really be determinant to why employees left. In the promotion variable, it is significantly visible that all the employees who left the company do not have promotion in the past 5 years, with only a very few proportions of them leaving after promotion in the recent 5 years. Lastly the salary variable, there are higher turnover in the low and medium salary group (looking at the blue bars in each category) with the low salary group being the highest.

From this section, it can be summarized that the following variables could be proportional to the leaving of the employees, which are the "department," "promotion" and "salary" attributes. An in-depth analysis of these will be conducted in the future hypothesis testing sections.

3.3 Simple Correlations between the variables

A simple correlation plot has also been plotted to have an overview of not only the variables to the target variable, but also to analyze the relationship between each variable and explore interesting trends. The department and salary variables are removed from the plot since they are character categorical variables which is of non- numeric values. From the plot attached on the right, apart from the variables against to themselves, average hours have moderate correlations to the last evaluation and number of projects which also works the same from either way. Another interesting point observed from the plot is that the target variable "left" has an inverse correlation to the satisfaction level indicating that employees with lower satisfaction level tends to leave the company.

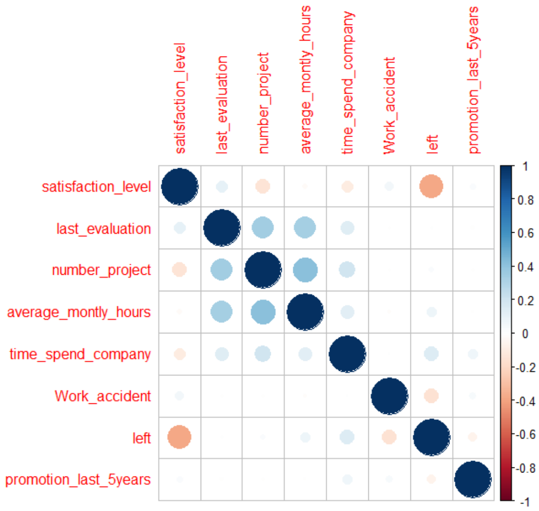


Figure 9 Correlation Matrix for variables

In addition to the previous correlation plot which is the total employees (with left and not left combined), another correlation plot with only the left employees will also be used to compare the differences between these two correlations and examine which variables have the highest correlations. Interestingly, the time spent in company has correlation with satisfaction level, last evaluation, number of projects and the average monthly hours, which sounds relevant. Since the time spend in company (in years) mean how long the employees have stayed in the company, the lower it is, the higher the employee turnover is and it is affected by how satisfied the employees are, their evaluation scores, the number of projects they worked on and their working hours. Especially the monthly hours makes so much sense since no employees enjoy working many overtime hours so this could be a great attribute for them leaving.

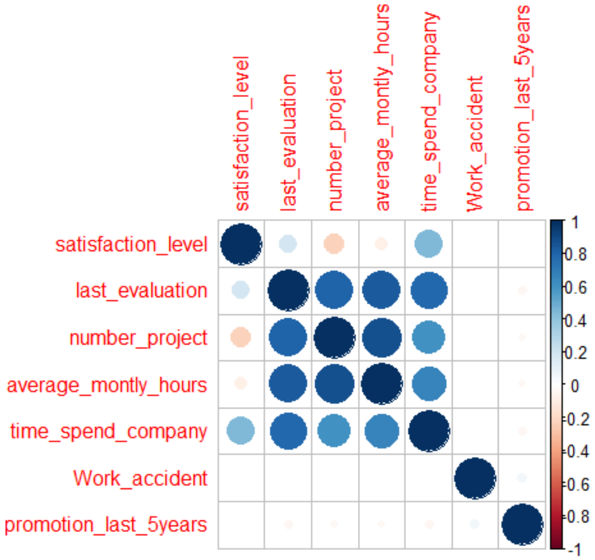


Figure 10 *Correlation Plot for Employees who left*

4. Hypothesis Analysis

As mentioned in the first section on the project goal on the hypothesis testing, this section will emphasize in testing those three hypotheses on salary, workplace safety and overall employee self- growth in the company of the dataset being given based on the understanding of the dataset. The visualizations will also be provided to prove the hypothesis results statistically and graphically.

Hypothesis 1: Salary is the reason the employees left the company

The first hypothesis is initiated based on the assumption that most of the employees in today’s era prioritize their pay over most of the other factors. From this assumption, the hypothesis will be correct if there are higher proportion of the employees left spotted in the low to medium salary. The graph beside plots the differences.

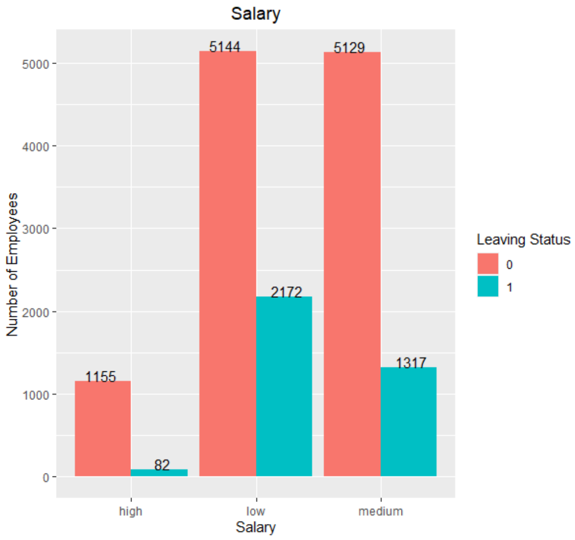


Figure 11 Distribution of Salary

Out of the 3,571 employees who left the company, more than half of them belong to the low salary group with one third of them being in the medium salary group. Moreover, comparing the class proportion within each salary category, “Low” salary is accounted for 42.22% of the total low salary group, where “Medium” accounts for 25.68% and “High” accounts for as low as 7.1%. From there, Low-Medium (especially Low group) has many employees turnover. Therefore, based on this result, the hypothesis of stating salary is the reason employees left the company is ACCEPTED.

Hypothesis 2: Employees leave the company because work is not safe

Work safety is also another issue that the employees prioritize when deciding whether to leave or stay. By saying that the work is not safe, the variable that would be investigated will be the work accident. The higher the work accident appears in the graph of the employees who left, the more it is likely to be a factor on why employees left. The graph below shows the difference.

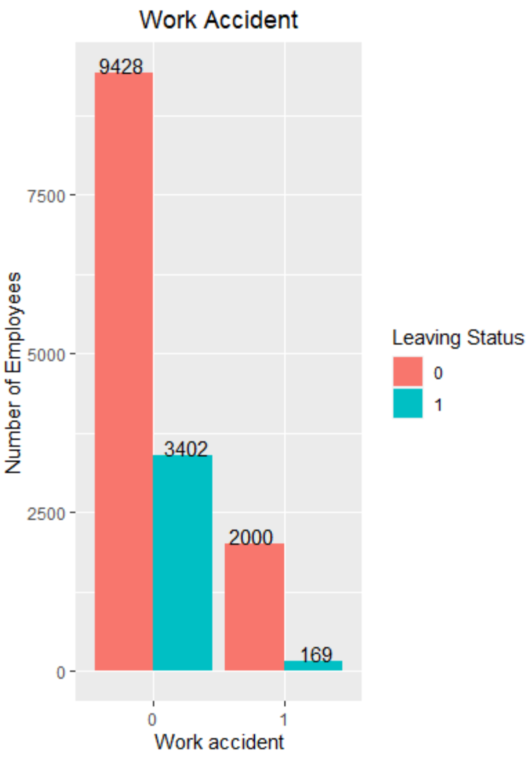


Figure 12 Distribution for Work Accident

From the graph, out of the 3,571 employees who left the company, over 3,402 of them have not left the company and the employees who left are only 169. This proportion is not that strong enough to argue that the workplace is unsafe and in fact the work accident is not the best metric to measure if the workplace is safe or not. Work accident could be not only the workplace unsafe, but it could also be the personal accident of the employees due to their own negligence. However, based off this dataset, this variable is the closest, it is used to compare and the result show that work accident is not a major factor to leading employees to leave the company so the hypothesis of stating the workplace is unsafe is REJECTED.

Hypothesis 3: Is this company a good place to grow professionally?

This section will emphasize in capturing the pattern of the employees who have left as well as the more capable employees who ended up leaving in order to analyze the reasons of why those employees left and based on this to assume the conditions of the workplace and determine if this is a suitable place for the employees to stay. Here, the work accident variable is excluded since it has been proven that it is not a factor for the employee turnover.

First, the capability of the employees based on this dataset is determined using the “last evaluation” variable. Since this variable is the evaluation of the performance of the employees, the better performing employees would reasonably have higher scores and vice versa to the opposites. The is left with around 1500 employees which is almost half of the employees who have left the company. This analysis is done because as mentioned above, businesses and companies generally intend to lower the employee turnover rate in order to retain the capable and valuable employees in the company, and by analyzing the pattern of those who left especially those who are deemed to be capable, it will be beneficial to the company on long term strategy to either provide better incentives or implement other long- term human resource strategies.

The first figure (with red graphs) shows the overall pattern of the employees who have left and the second figure coming below it (with the blue graphs) shows the pattern of the capable employees among the employees who have left.

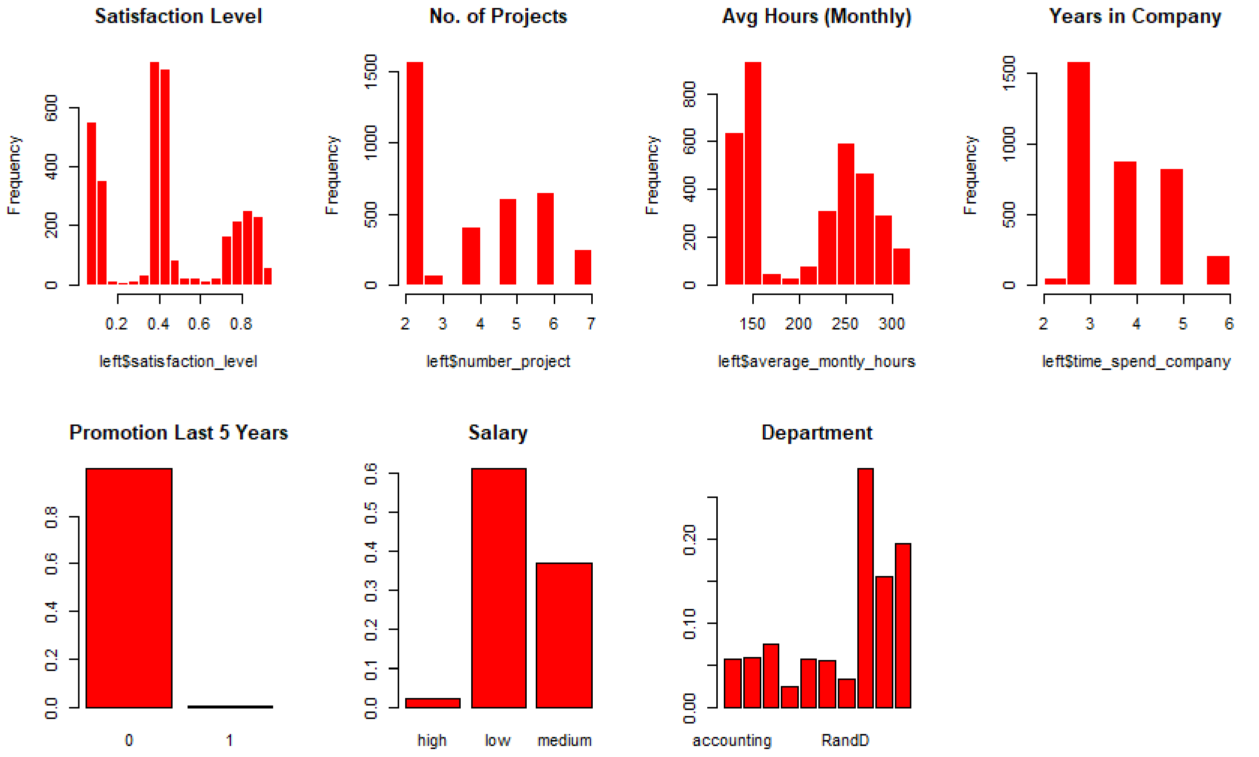


Figure 13 Visualizing Each Variable (All Employees who Left)

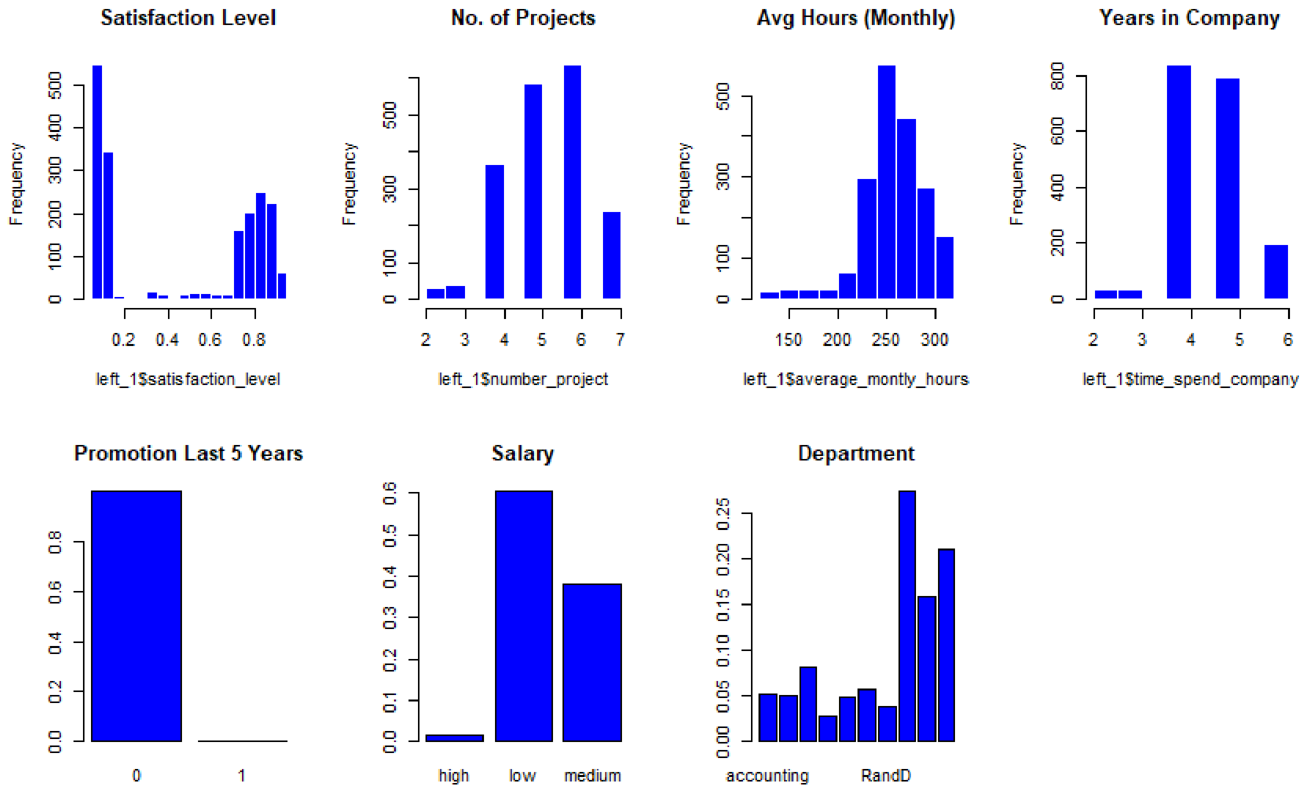


***Figure 14** Visualizing Each Variable (Valuable Employees)*

First, looking at the **"satisfaction level"** graphs, while the satisfaction level peaked at the middle for all employees' plots, it peaked at the lowest range of around 0.2. From there most employees leaving the company, especially those having higher evaluation scores, left the company because they are not or not quite satisfied with their work. Secondly, looking at the **"number of projects,"** one notable change from these two is the number of projects of one where all employees graph has the highest number of employees with one where the higher evaluation score employees have the highest amount in the 6 and 5 projects. Overall, there is no correlation to the reason the employees left with the number of projects so this will not be considered.

Next, looking at the **“average monthly hours”** graph, a surge in the higher hours could be observed comparing the two graphs. This clearly indicates that employees who left the company (especially the more capable ones) left because of the high number of working hours. There is a high chance from this seeing how visible the changes are. Furthermore, looking at the **“years in company,”** the employees who left overall have been with the company for 3 years mostly then followed by the 4 and 5 years, but the other graph shows that there are more employees left after 4 and 5 years, which to summarize is circling around the 3, 4 and 5 years. There would not be any correlation to the reasons on why the employees left the company with the years in the company and how long have the employees have been in the company, so this variable could be overlooked.

The other three variables: **“Promotion,” “Salary”** and **“Department”** seems to have the same trend. First, clearly, employees who left the company do not have a promotion in their work for the last 5 years, which evitable lead them to leave the company. Them, the salary, as mentioned in the Hypothesis 1, could also be another evident reason on why the employees left by seeing how the numbers are high in the low and medium salary group. Finally, for the department, it can be seen that there are a high number of employees left in the last three blocks, which is not that visible in these two plots. Therefore, it will be amplified below with the numbers of employees attached to it.

It can be seen below that there are higher numbers in the sales department with almost one third of the total employees who left, followed by the technical department with almost seven hundred employees and support with 555 employees who left the company. It seems like the department also matters with the reason on why they left, for instance, it could be that there is too much workload or pressure in the sales department which could explain why this department has so many employees' turnover. This variable could be another factor that affects the employee's turnover.

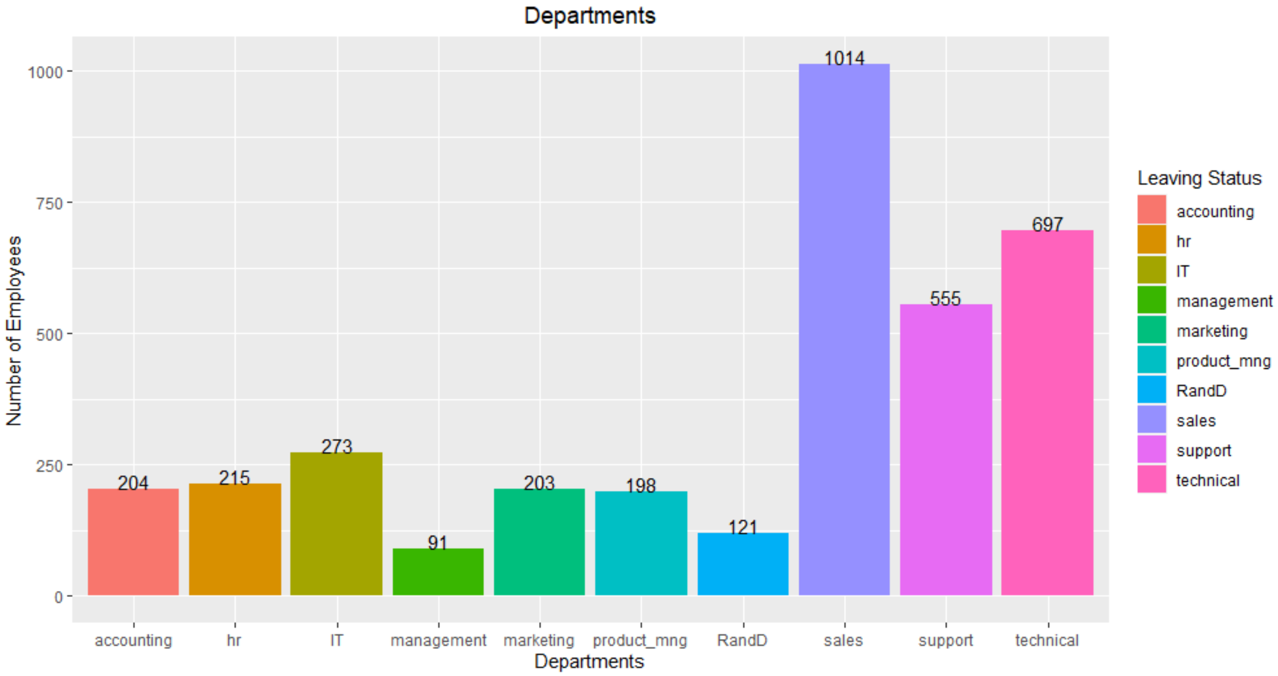


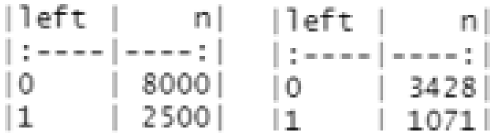
***Figure 15** Distribution for Department*

Overall, from seeing the trend in the employees who have left the company, the "satisfaction level," "average monthly hours," "promotion last 5 years", "salary" and "department" have high relations to why the employees left the company. However, taking account of the overall percentage of the employees who left the company which is only around 23% of the total employees, it is hard to prove that this company is a bad place for the employee's self- growth. Hence, while this stays true, the company have to implement long-term human resources strategy in the attributes which have been mentioned above in order to control and reduce their employee turnover.

5. Data Partitioning

Before the data is partitioned into training and testing data, the variables are first converted to the right data type where some variables are converted to categorical (work accident, left, promotion last 5 years, department and salary). The data is partitioned with 70% in training dataset and the remaining 30% in the testing data. The proportion of the two classes in the target is not that imbalanced with around 2:8 ratio which is not that heavily imbalanced so balancing the data is not necessary in this case since after partition, their ratio difference will be fewer, and this can be checked again after data partition.

Seeing the proportion of the training and testing dataset, it indeed does not seem like a heavily imbalanced dataset, so this has assured for not performing the balancing of the dataset. Where the left table in the figure represents the training dataset, the one on the right represents the testing dataset.



```
|left  |    n|   |left  |    n|
|:-----|----:|   |:-----|----:|
|0     | 8000|   |0     | 3428|
|1     | 2500|   |1     | 1071|
```

***Figure 16** Proportions of the Partitions (left: training, right: testing)*

6. Classification Models: Model Fitting

The models selected for this dataset would be (i) Classification Tree and (ii) Logistic Regression, where the Random Forest model will also be used to compare against the first model in terms of their performance metrics. For all models, the same training controls will be used as well as the same function of fitting the model, which is using the train () function to ensure same controls for each model to be compared upon. In addition, the models will be evaluated using the confusion matrix measures as well as the ROC Curves to determine which model has the best performance to deploy for this dataset in predicting whether the employees are leaving or not. All the models are used to predict the training data and the testing data to consider of the increase of the metrics from predicting the data that was fed into the model to the new data.

First, the classification tree model is fitted with the same controls as the other models with repeated cross validation method, with ten folds repeated for three times. The plot for the tree diagram and the CP graph is attached later. The figure on the right is the CP value graph. As the CP value increases, the accuracy decreases, and this shows that as the larger the split, the larger the accuracy (since the split increases as the CP value decreases). Pruning is necessary to avoid overfitting of the train data which will affect the accuracy of the testing data and eventually the accuracy of the new data and therefore in this case, the tree will be pruned to cp value of 0.07 in compromise to having favorable accuracy and at the same time to produce a decent plot. In addition, cross validation used in setting the controls for the model so overdoing the pruning might really affect the model performance. The tree model is plotted as shown in Figure 18.

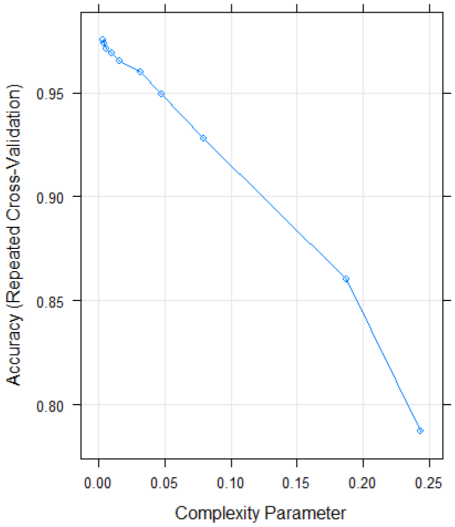


Figure 17 *CP Graph for Classification Tree*

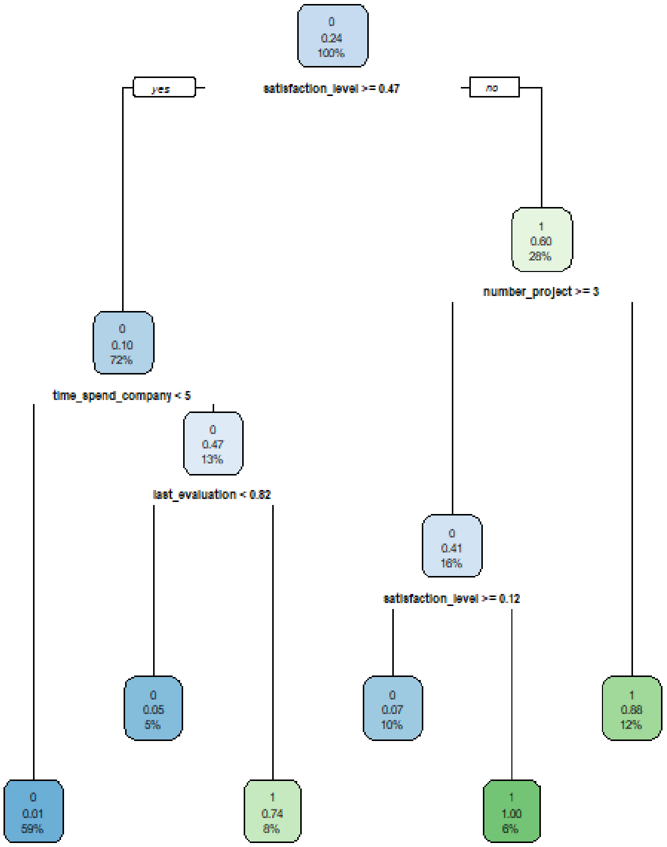


Figure 18 *Classification Tree Plot*

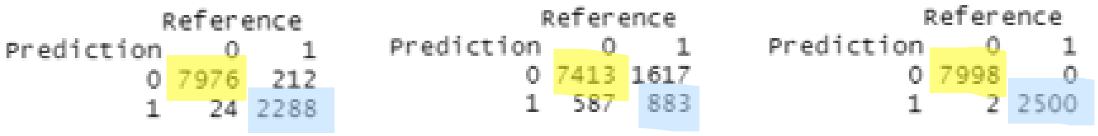
Then, the logistic regression and the random forest model is fitted, and their model performance evaluation will be discussed in the next section.

7. Model Performance Evaluation

The final model selection will be determined in this section where the confusion matrix and the ROC curve will be used to compare against the models to see which model has the highest metrics.

7.1 Confusion Matrix Measures

Confusion Matrices are plotted after predicting the outcomes for every model and the table below summarizes the metrics of the models evaluated with the training data and the testing data in order to consider of the overfitting by analyzing the differences between them. Along with that, confusion matrices for the testing data have also been attached below for better reference to the measures listed on the table.



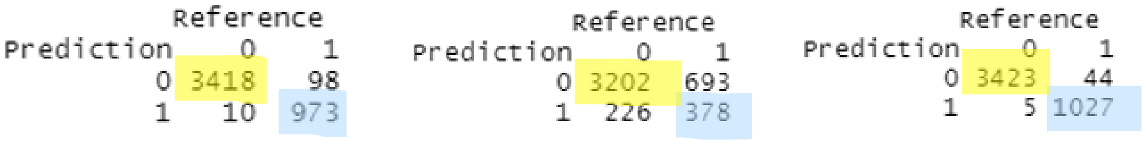
```
          Reference
Prediction    0    1
         0 7976  212
         1   24 2288
```

```
          Reference
Prediction    0    1
         0 7413 1617
         1  587  883
```

```
          Reference
Prediction    0    1
         0 7998    0
         1    2 2500
```

Figure 19 *Confusion Matrices for the models- Training Data*

(Classification Tree -> Logistic Regression -> Random Forest)



```
          Reference
Prediction    0    1
         0 3418   98
         1   10  973
```

```
          Reference
Prediction    0    1
         0 3202  693
         1  226  378
```

```
          Reference
Prediction    0    1
         0 3423   44
         1    5 1027
```

Figure 20 *Confusion Matrices for the models- Testing Data*

(Classification Tree -> Logistic Regression -> Random Forest)

	Model 1: Classification Tree		**Model 2: Logistic Regression**		**Model 3: Random Forest**	
Performance Measures	Training Data	Testing Data	Training Data	Testing Data	Training Data	Testing Data
Accuracy	0.9775	0.9760 **(-0.0015)**	0.7901	0.7957 **(+0.0056)**	0.9998	0.9891 **(-0.011)**
Sensitivity	0.9152	0.9085 **(-0.0067)**	0.3532	0.35294 **(-0.0003)**	1.0000	0.9589 **(-0.0411)**
Specificity	0.9970	0.9971 **(+0.0001)**	0.9266	0.93407 **(+0.0075)**	0.9998	0.9985 **(-0.0013)**
Precision$\frac{TP}{(FP+TP)}$	0.9896	0.9898 **(-0.002)**	0.6007	0.6258 **(+0.0251)**	0.9992	0.9952 **(-0.004)**
Type I Error (FP)	24 (0.229%)	10 (0.22%)	587 (5.59%)	226 (5.02%)	2 (0.019%)	5 (0.11%)
Type II Error (FN)	212 (2.02%)	98 (2.17%)	1617 (15.4%)	693 (15.4%)	0 (0%)	44 (0.978%)

Table 2 *Summary for all the model performance (In Both Training and Testing Data)*

From the table, the Random Forest model has a significant high number in the performance specifications with all metrics having perfect score, especially with the overall accuracy rate, specificity, and precision rates. It can also be seen that it has the lowest error rate (both Type I and Type II error) with not even 1% of the total observation in the respective datasets. Both training and testing data appears this way and since this is the result, it is almost sure to choose the "Random Forest" model for this part of the performance evaluation.

7.2 ROC Curve

The ROC Curve has also been used in order to analyze the Area Under Curve (AUC). ROC, standing for Receiver Operating Characteristics, is often used to show the trade-off between the sensitivity and specificity of the model and the Area Under the Curve (AUC) is the measure of the usefulness of a test where greater percentage of AUC indicates a better model. The straight line in these graphs represents the performance of the random model classifier, and it is a straight line indicating that 50% correct predictions with AUC of 50% where True Positive rate is equal to the False Positive Rate. This is used to compared with the model since random model is at the middle so any model above that is better.

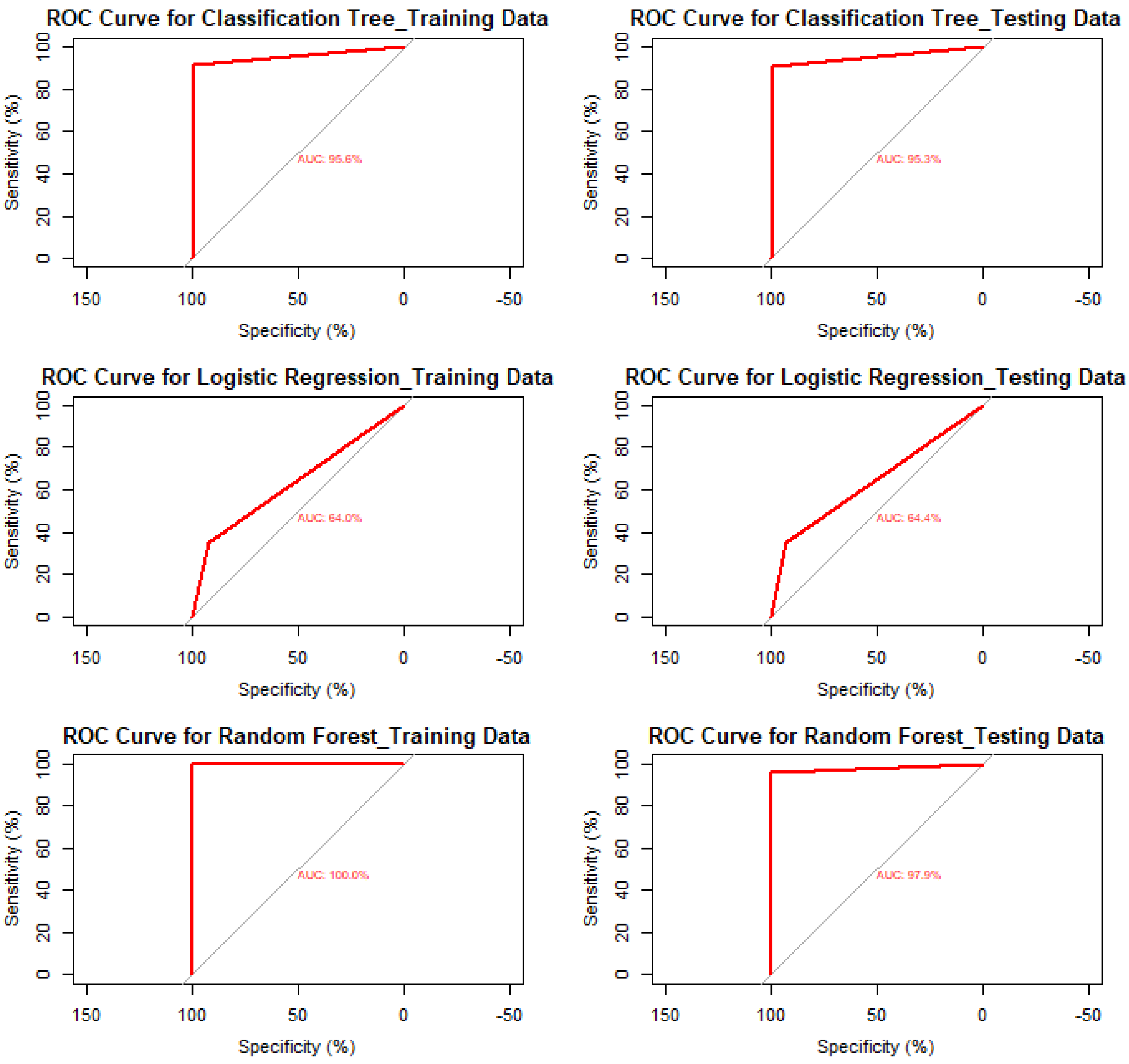


Figure 21 *ROC Curves for all the models (Left: Training Data; Right: Testing Data)*

Apparently, Random Forest model gives the best ROC Curves with 97.9% on the testing data and 100% on the training data which explicitly is the best model among all. Followed by the random forest model is the classification tree model which gives 95.6% on the training data and 95.3% on the testing data which are both high performance specifications. Comparatively, the Logistic Regression model performed poor with both AUC of around 64%.

Therefore, the conclusion drawn on this section is that the “Random Forest” model performed the best out of the three models that are evaluated. Not only are the metrics from the Confusion matrix are better, but also the ROC Curve is the best among all and gives almost perfect predictions on the testing data which are not included in the data that was fed to the model initially for building the model.

8. Conclusion

To conclude, this paper has proven three hypotheses from the Human Resources dataset as well as fitting models to see which model gives the best performance in predicting whether the employee is going to leave or not. First, it has been proven that “Salary is most likely to be one of the reasons why the employees left the company,” by analyzing the trends for salary variable through visualizations. It has also been proven that “The workplace is not unsafe for the employees,” by analyzing the work accident variable and see that the work accident rate is low. The third hypothesis, which is also the most interesting one has also been evaluated on if the company is a good place to grow professionally, and it has been proven that “It is relatively a fair place for employees to grow professionally.” For that hypothesis, all variables (except the work accident variable which was proven wrong in the second hypothesis) are used to analyzed and although it is hard to prove that the workplace is not a good place to grow, it is not entirely a great place for employees to grow professionally accounting that 23% of the employees who left the company have their common reasons to leave.

Finally, when selecting the best model to predict the employee’s turnover, “Random Forest” model has been chosen as the model with the best performance in terms of its higher scores in the confusion matrix measures as well as the ROC curve by making an almost perfect prediction. However, Random Forest could be a problem due to its black box nature since it is not a great descriptive tool to describe the relationships between the data as the model requires building of more trees to produce accurate predictions. Another practical drawback to this model is the longer run time of the model and especially with more trees in the model can make the model process slower. In most real work applications, the random forest model work fine fast enough but there could be certain circumstances when the run time is long and other models will be replaced for that reason. Nonetheless, this model is fairly the most flexible and better model tool to use but not without some practical limitations.

Last but not least, it would be recommended to the company to investigate the potential overtime work in all departments (especially the top 3 highest turnover departments which are Sales, Technical and Support). It is also recommended to adjust the promotional system or provide more incentives for the employees with superior performance since it seems like employees left as they do not feel their work is acknowledged and there is no raise in their pay. Therefore, these adjustments are highly recommended in order to reduce the employee turnover rate and make the company a place for their employees to grow professionally and grow together with the company.

END OF CASE 3

R Codes for Case Study 3