Mini Project - Mode of Transport

Name : Numer P

Table of Contents

1. Project Objective

The main objective of the report is to explore the Cars Dataset ("Cars_edited.csv") and in R and generate insights about the data set. This exploration report will consist of the following,

- Importing dataset in R
- Understanding the structure of Dataset
- Graphical exploration
- Descriptive Statistics
- Predictive Modelling

2. Assumptions

Transportation of employees are major source to the work. In the datasets, the employees are used to travel by Public transport, Bike and Car. The main approach is to find the employee using car as mode of transport to reach the office. In assuming, the employees are segregated by the salary and work experience shows that the employees are using car, bike and public transport facilities. Transport facilities are differed by their designations, salary and work experience of the individuals.

The datasets are classified by the numeric data and the categorical data variables. The category variables with the gender, designations, whether the employee is licensed driver or not. The numeric variables are classified with age, work experience, salary of the employees. Analysis is based on the numeric variables based on the mode of the transport. The transports are classified with the designations, gender and the license approvals.

3. Exploratory Data Analysis – Step by Step Approach

1. Environment Set up and Data Import
2. Variable Identification
3. Univariate Analysis
4. Bivariate Analysis
5. Variable Transformation
6. Feature Exploration

3.1 Environment Setup and Data Import

3.1.1 Install necessary packages and Import Libraries

This section is used to install packages and invoke the associated libraries. Having all packages at the same places increase code readability.

3.1.2 Setup Working Directory

Setting a working directory on starting of the R session makes importing and exporting data files and code files easier. Basically, working directory is the location/ folder on the PC where you have the data, codes etc. related to the project.

Please refer Appendix A for source code.

3.1.3 Import and read the dataset

The given dataset is in .csv format. Hence the command 'readr' is used for import the dataset.

Please refer Appendix A for source code.

3.2 Variable Identification

- setwd() used for setup working directory to export data and files from the folder or location in PC.
- getwd() used to identify the location was correctly entered or not.
- Library function is used to load the installed packages like ggplot2, dplyr, rpivotTable, readxl, readr, mice, car, cardata, psych, lattice, nfactors, data.table, tidyverse, broom, ggally, ROCR, pROC, caret, class, MASS, pscl, e1071, xgboost, DMwR.
- readr function is used to load the csv files in the path.
- attach function is used to reduce the reusability of variable name to enter each time.
- str function is used to check the category variables formats.
- summary function is used to get the summarised value like length, class and basic statistics values with quartile ranges.
- dim function is used to find the total observations and variables.
- Split function is used to split the data into train and test.
- Predict function is used to predict the train dataset and the validation dataset.
- glm is used for the predictive modelling for the logistic regression.
- Loglik is the function used to predict the likelihood of the predicted dataset.

3.2.1 Variable Identification – Inferences

#getwd()

It shows the working directory dataset

#library(mice)

(mice)takes out cbind and rbind the variables of two different datasets.

#library(readr)

(readr) helps in reading the rectangular datasets while the datasets are in table format.

#library(readxl)

(readxl) helps in reading the files in excel formats.

#library(dplyr)

(dplyr) helps in filter the datasets and intersect the datasets.

#library(ggplot2)

(ggplot2) helps is visualise the datasets in boxplot, histogram and graphical representation.

#read_csv

csv file was imported from the path and shows the variables

#str

str shows the variables along with class of the data. It shows some samples to understand the data.

#class

class function describes the full file in data.frame format. As the files includes category in season variables, it shows the values as character format.

#attach

Variable is attached to reduce the reusability in following syntax.

#dim

It shows the number observations and the variables associated in the file.

#summary

It produces the results as summarised format for each variable.

#names

It gives the column names from the dataset.

#boxplot

Boxplot graph is used for the dataset to find the outliers and the values are grouped.

#hist

This function is used to plot the histogram for the variables.

#glm

It is used for the generalized linear models and specifies the linear prediction for the variables and its error.

#vif

The variance inflation factor function is used for the calculation of variance in the linear models.

#lrtest

The generic function used for the prediction for the likelihood of the generalized linear models.

#loglik

The function is used for the likelihood prediction of the variables.

#predict

The predict function is used for the fitting the model function used for the datasets.

#table

The table is used for the cross classifying the variables in the contingency tables.

#cor

The correlation function is used to understand the correlation between the variables.

#naivebayes

This function is used for computing the posterior probability of the categorical variables with the predictor numeric variables.

#knn

The K Nearest Neighbour function is computed using the Euclidean distance of the variables and vectors in the train dataset using the test set values.

#confusionmatrix

Calculated for the cross validation in the observed and predicted values from the models.

#gbm

The generalized boosting method is used for the boosting technique using caret package.

#adaboost

Adaptive boosting is predicted with the function adaboost for the variables.

#xgboost

Extreme Gradient Boosting is used of the higher gradient techniques in the datasets.

#smote

Synthetic Minor Over-Sampling Techniques is used for the analysis of over sampling and under sampling methods

#bagging

Bagging techniques is used for the aggregating of the weak learners sequentially from the datasets.

#library(fastadaboost)

The library is loaded for the adaboost function to check the weak learners and weighting.

#library(DMwR)

Package is loaded for the smote analysis of the datasets.

#library(xgboost)

The xgboost is used for the analysis of the advance interface of the training the gradient boosting.

#library(bagging)

Bagging library is used for the ensemble methods to produce the combinations with to produce the datasets.

#as.matrix

Matrix function is used to create the matrices from the datasets and the model produced in the ensemble methods.

#sum

Sum function is used to identify the addition of the variables in the datasets.

#table

Table function is used for the creation of the datasets in the predicted variables for the better understanding of the variables.

#confusion matrix

Confusion matrix is used in the datasets for the analysis of the sensitivity, specificity and accuracy of the models in the variables.

#create data partition

The datasets are partitioned using the caret package for the analysis of gbm.

#predict

Predict function is used for the analysis in predicting the test datasets for the acquired model.

3.3 Univariate Analysis

Univariate analysis is the analysis of data of one variable at time and it involves whether the datasets are descriptive or inferential statistics.

1. Perform an EDA on the data

```
> transport=read_csv("Cars_edited.csv",col_names = TRUE)
```

The cars datasets is successfully imported to the r studio with true to its column names and the file can be used for the further analysis in the r studio.

```
> names(transport)
```

The names available in the datasets are read through this names command and the column names are followed by,

```
[1] "Age"       "Gender"    "Engineer"  "MBA"       "Work Exp"  "Salary"    "Distance"  "license"
[9] "Transport"
```

The column name “Work Exp” is separated by the space and the datasets can be errored as the variables can be separated as the names continues.

```
> names(transport)[5]="WorkExp"
```

The column name “Work Exp” is changed to non-space column name “WorkExp” with the help of above command.

```
> names(transport)
```

```
[1] "Age"       "Gender"    "Engineer"  "MBA"       "WorkExp"   "Salary"    "Distance"  "license"
[9] "Transport"
```

The column names are changed with non-space column names.

```
> transport
```

# A tibble: 444 x 9									
	Age	Gender	Engineer	MBA	WorkExp	Salary	Distance	license	Transport
	<dbl>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<chr>
1	28	Male	0	0	4	14.3	3.2	0	Public Transport
2	23	Female	1	0	4	8.3	3.3	0	Public Transport
3	29	Male	1	0	7	13.4	4.1	0	Public Transport
4	28	Female	1	1	5	13.4	4.5	0	Public Transport
5	27	Male	1	0	4	13.4	4.6	0	Public Transport
6	26	Male	1	0	4	12.3	4.8	1	Public Transport
7	28	Male	1	0	5	14.4	5.1	0	2Wheeler
8	26	Female	1	0	3	10.5	5.1	0	Public Transport
9	22	Male	1	0	1	7.5	5.1	0	Public Transport
10	27	Male	1	0	4	13.5	5.2	0	Public Transport
# … with 434 more rows									

The datasets are viewed for the better understanding on the variables and their factors with characters and numeric values as in the original dataset.

> str(transport)

```
tibble [444 × 9] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
 $ Age      : num [1:444] 28 23 29 28 27 26 28 26 22 27 ...
 $ Gender   : chr [1:444] "Male" "Female" "Male" "Female" ...
 $ Engineer : num [1:444] 0 1 1 1 1 1 1 1 1 1 ...
 $ MBA      : num [1:444] 0 0 0 1 0 0 0 0 0 0 ...
 $ WorkExp  : num [1:444] 4 4 7 5 4 4 5 3 1 4 ...
 $ Salary   : num [1:444] 14.3 8.3 13.4 13.4 13.4 12.3 14.4 10.5 7.5 13.5 ...
 $ Distance : num [1:444] 3.2 3.3 4.1 4.5 4.6 4.8 5.1 5.1 5.1 5.2 ...
 $ license  : num [1:444] 0 0 0 0 0 1 0 0 0 0 ...
 $ Transport: chr [1:444] "Public Transport" "Public Transport" "Public Transport" "Public
Transport" ...
 - attr(*, "spec")=
  .. cols(
  ..   Age = col_double(),
  ..   Gender = col_character(),
  ..   Engineer = col_double(),
  ..   MBA = col_double(),
  ..   `Work Exp` = col_double(),
  ..   Salary = col_double(),
  ..   Distance = col_double(),
  ..   license = col_double(),
  ..   Transport = col_character()
  .. )
```

The structure of the datasets are clearly identified with the factor variables character and the font size as per the character double, column double factors. The variables are identified with numeric and the character variables.

> summary(transport)

Age	Gender	Engineer	MBA	WorkExp	Salary
Min. :18.00	Length:444	Min. :0.0000	Min. :0.0000	Min. : 0.0	Min. : 6.50
1st Qu.:25.00	Class :character	1st Qu.:1.0000	1st Qu.:0.0000	1st Qu.: 3.0	1st Qu.: 9.80
Median :27.00	Mode :character	Median :1.0000	Median :0.0000	Median : 5.0	Median :13.60
Mean :27.75		Mean :0.7545	Mean :0.2528	Mean : 6.3	Mean :16.24
3rd Qu.:30.00		3rd Qu.:1.0000	3rd Qu.:1.0000	3rd Qu.: 8.0	3rd Qu.:15.72
Max. :43.00		Max. :1.0000	Max. :1.0000	Max. :24.0	Max. :57.00
			NA's :1		

Distance	license	Transport
Min. : 3.20	Min. :0.0000	Length:444
1st Qu.: 8.80	1st Qu.:0.0000	Class :character
Median :11.00	Median :0.0000	Mode :character
Mean :11.32	Mean :0.2342	
3rd Qu.:13.43	3rd Qu.:0.0000	
Max. :23.40	Max. :1.0000	

The summary of the datasets identified with the NA values and the values are treated with removing the NA. The dimension of the character variables are identified with the 444 observations and the datasets are provided with 9 independent variables.

```
> dim(transport)
[1] 444   9
```

The datasets are observed with 444 observations and 9 variables including the NA values for the datasets.

```
> any(is.na(transport))
[1] TRUE
```

The NA values are identified and the values are need to treated to minimize the error occurrence in the datasets.

```
> transport=na.omit(transport)
```

The NA are omitted form the datasets and treated for analyses with segregation of the variables as per the categorical function and the variables are identified with the observations.

```
> dim(transport)
[1] 443   9
```

After NAs are treated the dimension of the datasets are observed with 443 observations and with 9 variables.

```
> summary(transport)
```

Age	Gender	Engineer	MBA	WorkExp	Salary
Min. :18.00	Length:444	Min. :0.0000	Min. :0.0000	Min. : 0.0	Min. : 6.50
1st Qu.:25.00	Class :character	1st Qu.:1.0000	1st Qu.:0.0000	1st Qu.: 3.0	1st Qu.: 9.80
Median :27.00	Mode :character	Median :1.0000	Median :0.0000	Median : 5.0	Median :13.60
Mean :27.75		Mean :0.7545	Mean :0.2528	Mean : 6.3	Mean :16.24
3rd Qu.:30.00		3rd Qu.:1.0000	3rd Qu.:1.0000	3rd Qu.: 8.0	3rd Qu.:15.72
Max. :43.00		Max. :1.0000	Max. :1.0000	Max. :24.0	Max. :57.00

Distance	license	Transport
Min. : 3.20	Min. :0.0000	Length:444
1st Qu.: 8.80	1st Qu.:0.0000	Class :character
Median :11.00	Median :0.0000	Mode :character
Mean :11.32	Mean :0.2342	
3rd Qu.:13.43	3rd Qu.:0.0000	
Max. :23.40	Max. :1.0000	

The NA are removed from the variables and the values are treated with the observation for the MBA variables in the summary of the datasets.

```
> transport$Gender=as.factor(transport$Gender)
> transport$Engineer=as.factor(transport$Engineer)
> transport$MBA=as.factor(transport$MBA)
> transport$license=as.factor(transport$license)
> transport$Transport=as.factor(transport$Transport)
```

The variables (Gender, Engineer, MBA, License, Transport) are comes under category variables as the variables are treated with the values 0 and 1.

Here, the variables are organized by negative and positive outcomes of the variables used in the analysis. 0 mentions the variables are acquired with positive outcomes and 1 is mentioning for the negative outcomes of the factor variables. The 0 and 1 variables are associated with the Engineer, MBA, License variables. The Gender variables are treated with Male and Female characters and Transport variables are treated with Car, 2 wheeler and Public Transport characters.

```
> str(transport)
```

```
tibble [443 × 9] (S3: tbl_df/tbl/data.frame)
 $ Age      : num [1:443] 28 23 29 28 27 26 28 26 22 27 ...
 $ Gender   : Factor w/ 2 levels "Female","Male": 2 1 2 1 2 2 2 1 2 2 ...
 $ Engineer : Factor w/ 2 levels "0","1": 1 2 2 2 2 2 2 2 2 2 ...
 $ MBA      : Factor w/ 2 levels "0","1": 1 1 1 2 1 1 1 1 1 1 ...
 $ WorkExp  : num [1:443] 4 4 7 5 4 4 5 3 1 4 ...
 $ Salary   : num [1:443] 14.3 8.3 13.4 13.4 13.4 12.3 14.4 10.5 7.5 13.5 ...
 $ Distance : num [1:443] 3.2 3.3 4.1 4.5 4.6 4.8 5.1 5.1 5.1 5.2 ...
 $ license  : Factor w/ 2 levels "0","1": 1 1 1 1 1 2 1 1 1 1 ...
 $ Transport: Factor w/ 3 levels "2Wheeler","Car",..: 3 3 3 3 3 3 1 3 3 3 ...
 - attr(*, "na.action")= 'omit' Named int 145
  ..- attr(*, "names")= chr "145"
```

The transport datasets are identified with the factor level variables for Gender, Engineer, MBA, License, Transport. Transport variables are treated with three level factor variables and the other variables are created with two level factor variables.

```
> summary(transport)
```

Age	Gender	Engineer	MBA	WorkExp
Min. :18.00	Female:127	0: 108	0: 331	Min. : 0.0
1st Qu.:25.00	Male :316	1: 335	1: 112	1st Qu.: 3.0
Median :27.00				Median : 5.0
Mean :27.75				Mean : 6.3
3rd Qu.:30.00				3rd Qu.: 8.0
Max. :43.00				Max. :24.0

Salary	Distance	license	Transport
Min. : 6.50	Min. : 3.20	0: 339	2Wheeler : 83
1st Qu.: 9.80	1st Qu.: 8.80	1:104	Car : 61
Median :13.60	Median :11.00		Public Transport:299
Mean :16.24	Mean :11.33		
3rd Qu.:15.75	3rd Qu.:13.45		
Max. :57.00	Max. :23.40		

The variables are identified for the factor variables with transport facilities are classified by 2 wheeler, car and public transport. Gender variables are identified with two factor as Male and female.

```
> transport$Transport=as.character(transport$Transport)
> transport$Transport[transport$Transport %in% "2Wheeler"]="0"
> transport$Transport[transport$Transport %in% "Car"]="1"
> transport$Transport[transport$Transport %in% "Public Transport"]="0"
> str(transport$Transport)
```

```
chr [1:443] "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" ...
```

As the transport variables are treated with three level factors and the variables are converted to two level character for the future analysis and the prediction class. The variables are segregated as 0 for 2 wheeler, Public Transport and 1 for Car. Since, the character of the variables are segregated for the two level factors, the car usage of employee can be easily predicted.

```
> transport$Gender=as.character(transport$Gender)
> transport$Gender[transport$Gender %in% "Female"]="0"
> transport$Gender[transport$Gender %in% "Male"]="1"
```

> str(transport$Gender)

```
chr [1:443] "1" "0" "1" "0" "1" "1" "1" "0" "1" "1" "0" "1" "1" "1" "1" "1" "1" "0" "1" "1" "1" "1" ...
```

Since, the analysis requires all the factor variables should be in 0 and 1 values, the gender variable also classified with 0 as female and 1 male in the gender category variables.

> transport$Transport=as.factor(transport$Transport)

> transport$Gender=as.factor(transport$Gender)

The character variables are changed to factor variables and the analysis is easy to find the interpretation of the values for the datasets and the values are predicted for the various analysis.

> str(transport)

```
tibble [443 × 9] (S3: tbl_df/tbl/data.frame)
 $ Age      : num [1:443] 28 23 29 28 27 26 28 26 22 27 ...
 $ Gender   : Factor w/ 2 levels "0","1": 2 1 2 1 2 2 2 1 2 2 ...
 $ Engineer : Factor w/ 2 levels "0","1": 1 2 2 2 2 2 2 2 2 2 ...
 $ MBA      : Factor w/ 2 levels "0","1": 1 1 1 2 1 1 1 1 1 1 ...
 $ WorkExp  : num [1:443] 4 4 7 5 4 4 5 3 1 4 ...
 $ Salary   : num [1:443] 14.3 8.3 13.4 13.4 13.4 12.3 14.4 10.5 7.5 13.5 ...
 $ Distance : num [1:443] 3.2 3.3 4.1 4.5 4.6 4.8 5.1 5.1 5.1 5.2 ...
 $ license  : Factor w/ 2 levels "0","1": 1 1 1 1 1 2 1 1 1 1 ...
 $ Transport: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 - attr(*, "na.action")= 'omit' Named int 145
  ..- attr(*, "names")= chr "145"
```

As the category variables are classified with two level factors and the numeric variables are predicted with the numeric values for the higher prediction in the values.

> summary(transport)

Age	Gender	Engineer	MBA	WorkExp	Salary	Distance	license	Transport
Min. :18.00	0:127	0:108	0:331	Min. : 0.0	Min. : 6.50	Min. : 3.20	0:339	0:382
1st Qu.:25.00	1:316	1:335	1:112	1st Qu.: 3.0	1st Qu.: 9.80	1st Qu.: 8.80	1:104	1: 61
Median :27.00				Median : 5.0	Median :13.60	Median :11.00		
Mean :27.75				Mean : 6.3	Mean :16.24	Mean :11.33		
3rd Qu.:30.00				3rd Qu.: 8.0	3rd Qu.:15.75	3rd Qu.:13.45		
Max. :43.00				Max. :24.0	Max. :57.00	Max. :23.40		

The summary of the variables are predicted with the values and the factor level classification is clearly identified with the values 0 and 1. The values are organized with the variables as follows,

- Gender – 0 as Female and 1 as Male
- Engineer – 0 as Engineer Graduate and 1 as Non-Engineer Graduate
- MBA – 0 as MBA Graduate and 1 as Non-MBA Graduate

- License – 0 as Licensed Driver and 1 as Non-Licensed Driver
- Transport – 0 as 2 Wheeler and Public Transport while 1 as Car.

```
> head(transport,3)
```

```
# A tibble: 3 x 9
   Age Gender Engineer MBA   WorkExp Salary Distance license Transport
 <dbl> <fct>  <fct>    <fct>   <dbl>  <dbl>    <dbl> <fct>   <fct>
1    28 1      0        0           4   14.3     3.2 0       0
2    23 0      1        0           4    8.3     3.3 0       0
3    29 1      1        0           7   13.4     4.1 0       0
```

The header of the transport datasets are classified with the single character column and the values are organized by the various factor variables.

```
> tail(transport,4)
```

```
# A tibble: 4 x 9
   Age Gender Engineer MBA   WorkExp Salary Distance license Transport
 <dbl> <fct>  <fct>    <fct>   <dbl>  <dbl>    <dbl> <fct>   <fct>
1    38 1      1        0          19     44     21.5 1       1
2    37 1      1        0          19     45     21.5 1       1
3    37 1      0        0          19     47     22.8 1       1
4    39 1      1        1          21     50     23.4 1       1
```

The last four tail of the datasets are classified with the various factor is identified using the tail function.

```
> ct.data=subset(transport,select = c(Gender,Engineer,MBA,license))
> num.data=subset(transport,select = -c(Gender,Engineer,MBA,license,Transport))
```

The variables are classified with the category and numerical values. Category variables contains levels and the numeric variables are identified with the numeric values. The variables are associated with the independent and dependent characteristics of the variables.

```
> names(ct.data)
```

```
[1] "Gender"   "Engineer" "MBA"      "license"
```

The variables associated for the category variables subset are Gender, Engineer, MBA and license.

```
> names(num.data)
```

```
[1] "Age"      "WorkExp"  "Salary"   "Distance"
```

The variables associated for the numeric variables subset are Age, Work Experience, Salary and Distance.

> by(transport,INDICES = transport$Transport,FUN = summary)

transport$Transport: 0								
Age	Gender	Engineer	MBA	WorkExp	Salary	Distance	license	Transport
Min. :18.00	0:114	0: 99	0:282	Min. : 0.000	Min. : 6.50	Min. : 3.20	0:326	0:382
1st Qu.:24.00				1st Qu.: 2.000	1st Qu.: 9.50	1st Qu.: 8.40		
Median :26.00				Median : 4.000	Median :12.80	Median :10.50		
Mean :26.47	1:268	1:283	1:100	Mean : 4.806	Mean :13.05	Mean :10.69	1: 56	1: 0
3rd Qu.:28.00				3rd Qu.: 7.000	3rd Qu.:14.68	3rd Qu.:12.80		
Max. :36.00				Max. :18.000	Max. :37.00	Max. :21.00		
---								
transport$Transport: 1								
Age	Gender	Engineer	MBA	WorkExp	Salary	Distance	license	Transport
Min. :30.00	0:13	0: 9	0:49	Min. : 4.00	Min. :15.60	Min. : 9.0	0:13	0: 0
1st Qu.:33.00				1st Qu.:11.00	1st Qu.:17.00	1st Qu.:12.3		
Median :36.00				Median :17.00	Median :39.90	Median :14.4		
Mean :35.72	1:48	1:52	1:12	Mean :15.66	Mean :36.24	Mean :15.3	1:48	1:61
3rd Qu.:39.00				3rd Qu.:20.00	3rd Qu.:45.00	3rd Qu.:18.1		
Max. :43.00				Max. :24.00	Max. :57.00	Max. :23.4		

The by function is used to understand the variables association for the categories and the numeric values in the identified variables. Transport facilities used by the employees as 2 wheeler and public transport is maximum to the age 36 and the car mode of transport selected for the employee with maximum age of 23. Car transport facilities are acquired with higher work experience, salary when compared with 2 wheeler and public transport facility. The employees distance are nearly most similar and most employees are preferring public transport and 2 wheeler as the basic operations for the daily mode of transport. Whether the employees using car as the mode transport are without licensed driver as 1% and the employees preferring public transport and 2 wheeler are without licensed as nearly 80%.

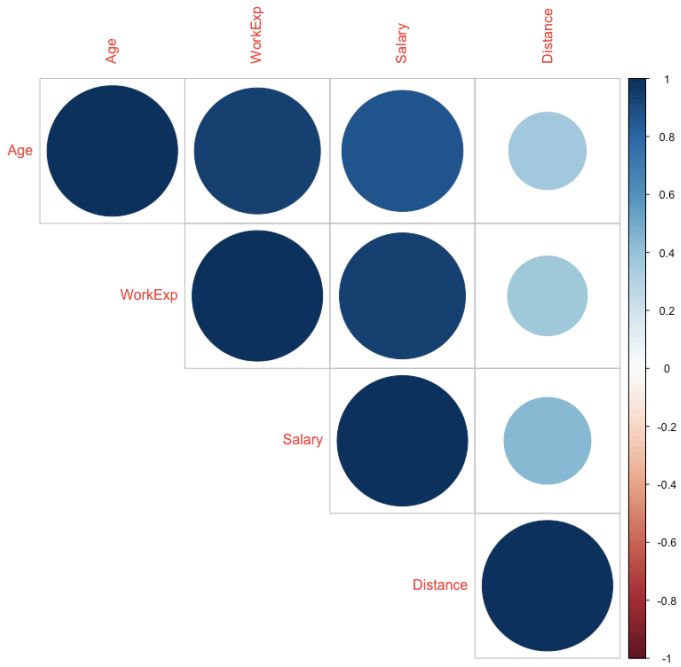
> correlation=cor(num.data)

> correlation

	Age	WorkExp	Salary	Distance
Age	1.0000000	0.9322510	0.8607652	0.3530563
WorkExp	0.9322510	1.0000000	0.9320081	0.3727857
Salary	0.8607652	0.9320081	1.0000000	0.4422379
Distance	0.3530563	0.3727857	0.4422379	1.0000000

The correlation is used to predicted the numeric variables are predicted for how the variables are maintaining relationship with the other variables for the various factors and the category variables are distinguished by the independent characters and the variables are associated for the higher correlation and the lower correlation. The variables with age and work experience are highly correlated and the age with distance are lightly correlated. While correlating the Work Experience with salary, the data are predicted with higher correlations.

```
> corrplot::corrplot(correlation,method = "circle",type = "upper")
```



The correlation plots are measured with the higher correlation as darker circles and the lesser correlation as light shaded circles and the plot shows the variables correlated rates from the numeric datasets.

```
> ggpairs(transport[,c("Age","WorkExp","Salary","Distance")],
+         ggplot2::aes(colour=as.factor(transport$Transport)))
```

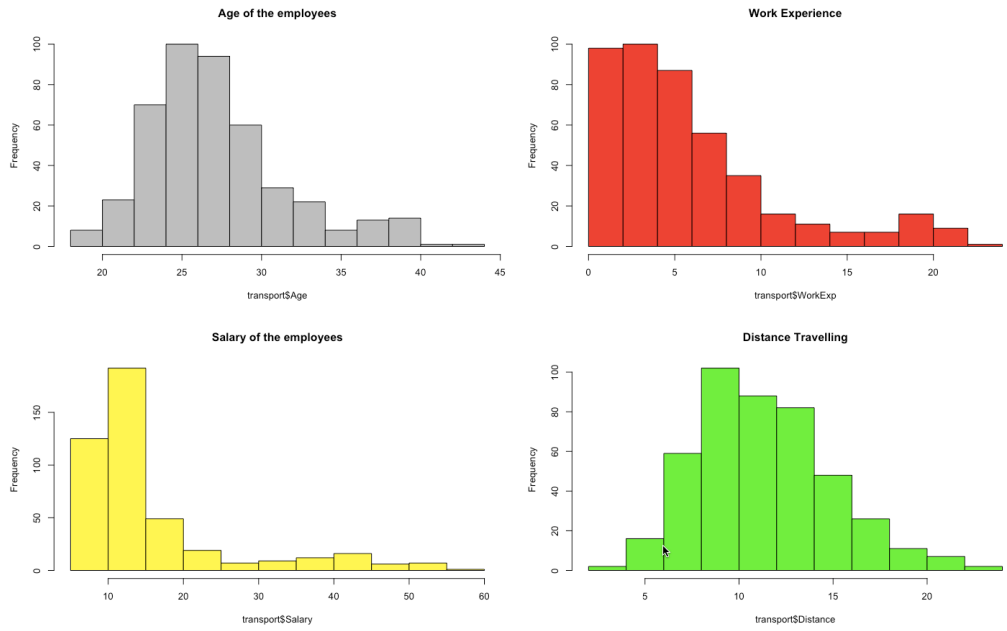
The plots are classified with the variables are highly correlated as per the plots for the two level classification for the numeric variables in the relations between the age, work experience, salary and distance of the employees. The correlation are classified with various predictions and the values are plotted with two different variables for the clustered groups. When plotting the transport variables the variables are classified with the plots are equally distributed in the distance and unequally distributed in salary. The employees are using the car transport facility is based on the salary correlations.

```
> outliers=boxplot(num.data,plot = FALSE)$out
> outliers
```

```
 [1] 39.0 39.0 39.0 38.0 40.0 38.0 38.0 38.0 38.0 40.0 40.0 39.0 40.0 38.0 39.0 38.0 40.0 39.0 38.0 42.0
[21] 40.0 43.0 40.0 38.0 39.0 19.0 16.0 21.0 17.0 16.0 18.0 19.0 18.0 21.0 16.0 19.0 19.0 18.0 19.0 20.0
[41] 22.0 16.0 20.0 18.0 21.0 20.0 20.0 16.0 17.0 21.0 18.0 20.0 21.0 19.0 22.0 22.0 19.0 24.0 20.0 19.0
[61] 19.0 19.0 21.0 36.6 38.9 25.9 34.8 28.8 39.9 39.0 28.7 36.9 28.7 34.9 47.0 28.8 36.9 54.0 29.9 34.9
[81] 36.0 44.0 37.0 24.9 43.0 37.0 54.0 44.0 34.0 48.0 42.0 51.0 45.0 34.0 28.8 45.0 42.9 41.0 40.9 30.9
[101] 41.9 43.0 33.0 36.0 33.0 38.0 46.0 45.0 48.0 35.0 51.0 51.0 55.0 45.0 42.0 52.0 38.0 57.0 44.0 45.0
[121] 47.0 50.0 20.7 20.8 21.0 21.3 21.4 21.5 21.5 22.8 23.4
```

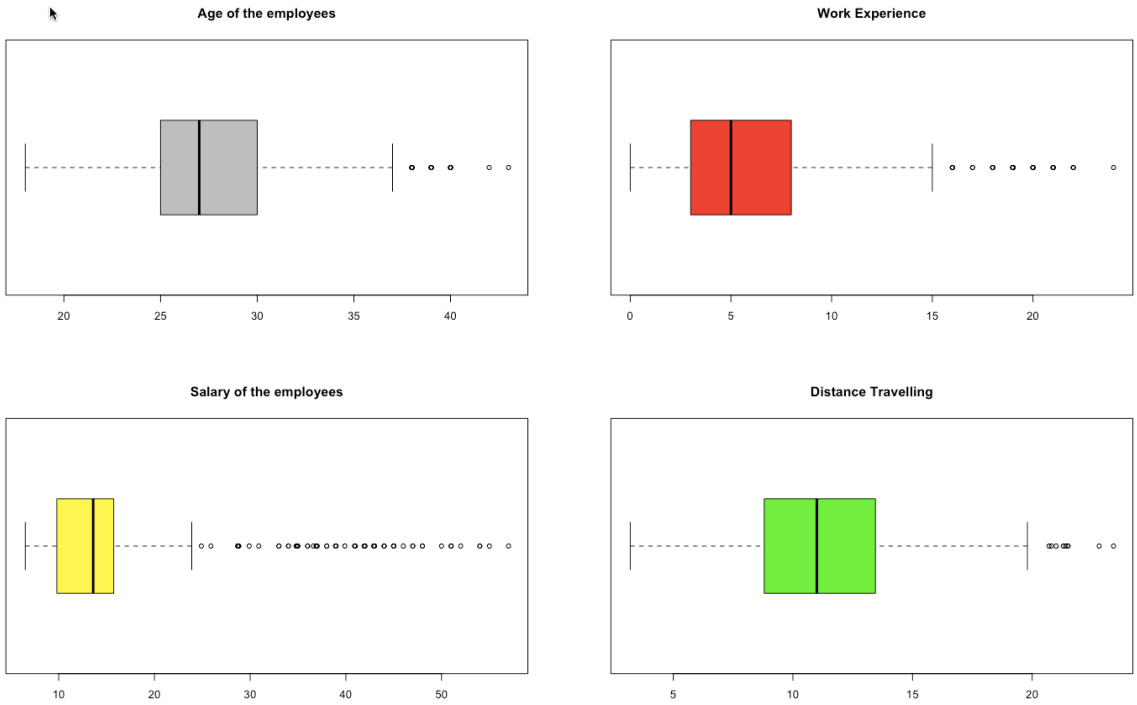
The outliers of numeric variables are classified with various numeric values and the values are grouped by the numeric category variables using boxplot values.

```
> par(mfrow=c(2,2))
> hist(transport$Age,main = "Age of the employees",col = "grey")
> hist(transport$WorkExp,main = "Work Experience",col = "red")
> hist(transport$Salary,main = "Salary of the employees",col = "yellow")
> hist(transport$Distance,main = "Distance Travelling",col = "green")
```



The histogram of the numeric variables are classified by age, work experience, salary and distance of the employees. The most the employees are based on the middle age category and more employees are segregated by the work experience below 10 years and the salary of the employees are certainly high at the region of the 10 to 15. The people travelling to the office are maintained with the 8 to 10 kilometres and the employees are classified with the salary and the work experience for the values are based on hierarchical structure of the organizations. The employees with higher work experience and age are classified with the higher position employees. The histogram clearly explains the employees basic details with their work experience, salary and distance travelled by them from the office.

```
> par(mfrow=c(2,2))
> boxplot(transport$Age,horizontal = T,main = "Age of the employees",col = "grey")
> boxplot(transport$WorkExp,horizontal = T,main = "Work Experience",col = "red")
> boxplot(transport$Salary,horizontal = T,main = "Salary of the employees",col = "yellow")
> boxplot(transport$Distance,horizontal = T,main = "Distance Travelling",col = "green")
```

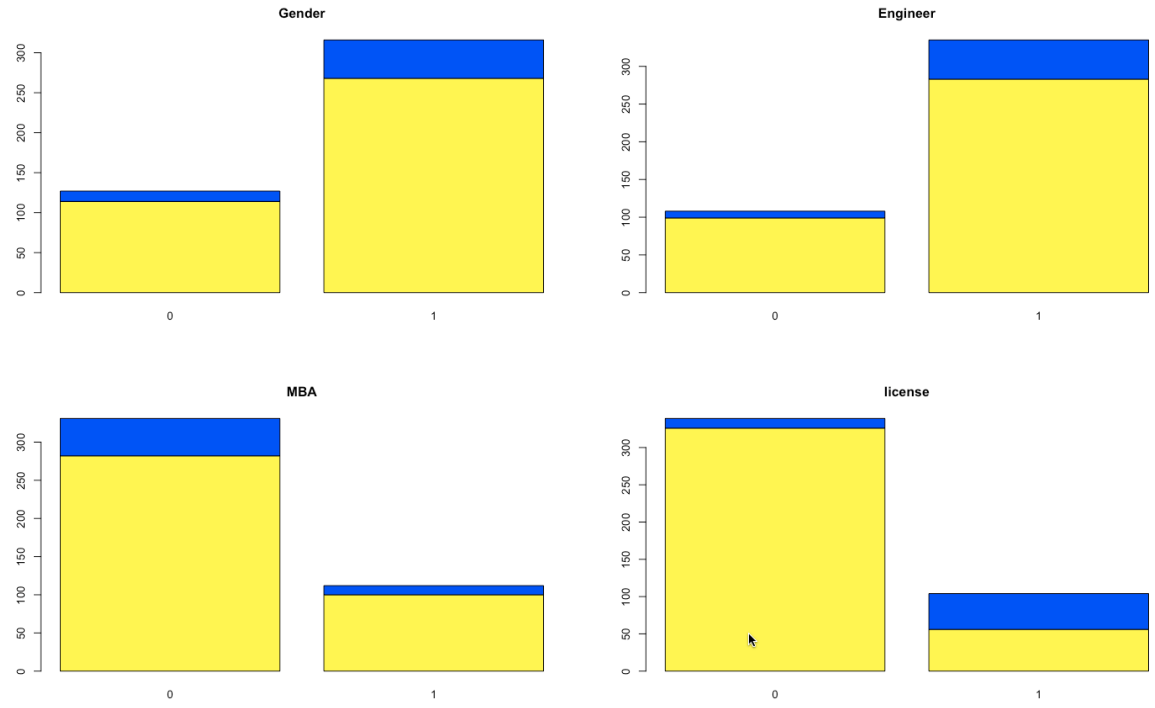


The boxplot is used for the analysis of the numeric variables outliers. The age category shows that 50% of the employees are between the age 25 to 30 and the 50% work experience of the employees are between 3 to 8 and the 50% of the salary of the employees are between 10 to 15 lacs. The maximum age of the employee is about 43 and work experience is about 23 and the salary is 57 lacs per annum. The boxplot is accrued with most parameters of the 50% and the employees classified with the age and work experience qualification.

```
> par(mfrow=c(2,2))
> for(i in names(ct.data)){
+   print(i)
+   print(round(prop.table(table(transport$Transport,ct.data[[i]])),digits = 3)*100)
+   barplot(table(transport$Transport,ct.data[[i]]),
+           col = c("yellow","blue"),
+           main = names(ct.data[i]))
+ }
```

[1] "Gender"			[1] "Engineer"		
	0	1		0	1
0	25.7	60.5	0	22.3	63.9
1	2.9	10.8	1	2	11.7
[1] "MBA"			[1] "license"		
	0	1		0	1
0	63.7	22.6	0	73.6	12.6
1	11.1	2.7	1	2.9	10.8

The category variables are classified with the transport facilities and the gender based classification provides the transport mode for the not using car is based on the values are maintained with 85% and 15% of female and male respectively. The engineer graduates are calculated by the values based on the 86% and 14% of engineer and non-engineers. MBA graduates are classified with 96% and 4% of MBA holders and Non-MBA holders.



The employees using car as mode of transport are little based on the values acquired. Gender classification shows that female employees are using the car as lesser than male employees. The Non-MBA graduates are using the car as mode of transport is higher than the MBA graduates and the Non-MBA graduates are indicating the employees can offer the car mode of transport easily for their daily transport. And the employees with the license is using the car as mode of transport.

2. Illustrate the insights based on EDA

- The mode of transport for the employees are classified as 2 wheeler, public transport and car. Multi-level factors can makes the datasets are more complex in predicting the employees usage of the car as mode of transport. Hence, the transport facilities are changed as two level factor like 2wheeler and public transport user as one category and the car users are another category.
- The car transport are majorly used by the higher work experience employees and higher salary brought employees. Since, the car transport is majorly influencing the employees with age factors as the employees with higher age is using the car facilities to reach the office.
- The distance factor is similar for the 2 wheeler, public transport and car transport facilities for the employees. The maximum distance covered by the car users are about 23 kilometres and the maximum distance travelled by non-car users are about 21 kilometres. Hence, the distance is not making much difference in employees transport and the transport facilities of using car is mainly of age, work experience and salary as ultimate factors.
- When comparing the employees basic details and the designation of the employees, the car facilities used by the employees are differed. The Non-MBA graduates are preferring car transport facility when compared with the MBA graduates which is highly significant of 10% increased factor. The Engineer graduates are using car as mode transport compared than MBA graduate employees. This inferencing that the employees with Engineering graduates are using car as mode of transport when compared others.
- Car mode of transport are mainly significant with the engineering graduates and the employees earning salary as higher when compared with non-engineer, non-MBA and employees getting salary below 5 Lacs. In this factor, age and gender is classifying that the male employees with higher age and female employees with higher age are using car as mode of transport facilities to reach office.

- The concluding factors of the employees using the car are explained with the employees in the maximum category of age, work experience, salary in the place of engineers and the employees experiencing the car mode of transport in Non-MBA graduates indicates that the employees may with higher work experience, age and higher salary when compared with MBA graduates. This insights provided that male are using car as mode of transport than females and making that employees using car as mode of transport in factor of engineer, non-MBA graduates and higher work experience, higher salary are preferred to be male.

3.4 Bivariate Analysis

Bivariate Analysis is used to analyse the two variables and find the relationship between them. This analysis will help in identifying the association and strength of the variables. The analysis is used find the regression, distributions and plots.

As the Cars Dataset is mostly focus on customer churn, the categorical variables are selected and the numeric variables are selected and multicollinearity is measured and the insights are computed.

3. Check for Multicollinearity - Plot the graph based on Multicollinearity & treat it.

```
> attach(transport)
> log.reg=glm(Transport~Age,data = transport,family = binomial(link = logit))
> summary(log.reg)
```

Logistics regression is used for the collinearity check and the collinearity is based on the variables on age and mode of transport.

Min	1Q	Median	3Q	Max
-2.24156	-0.18868	-0.08458	-0.02527	2.23336

The deviance residuals are calculated with the maximum value of 2.2% and the age is less error on the variables.

	Estimate	Std. Error	z value	Pr(>|z|)	
(Intercept)	-26.5852	3.331	-7.981	1.45E-15	***
Age	0.8059	0.1048	7.686	1.51E-14	***

The significance variables are higher and the comparing the collinearity between the variables age and positively correlated for the higher z value.

The AIC value is indicating with the 117.43 and the values are measured for the DOF is about 441.

```
> log.reg=glm(Transport~Gender,data = transport,family = binomial(link = logit))

> summary(log.reg)
```

The collinearity of the variables between transport and gender is lesser significant and the values are maintained with the less error.

Min	1Q	Median	3Q	Max
-0.574	-0.574	-0.574	-0.465	2.1351

The values are measured and the maximum values are increased by the maximum 2.1351 and the residuals are measured minimal error.

	Estimate	Std. Error	z value	Pr(>|z|)	
(Intercept)	-2.1712	0.2927	-7.418	1.19E-13	***
Gender1	0.4515	0.332	1.36	1.74E-01	

The coefficients are calculated with the DOF is 441and the z value is predicted for the variables is predicting the values are measured as 1.36 which is lesser significant in the transport variables. The values are calculate in the standard error for the measures in 0.33.

```
> log.reg=glm(Transport~Engineer,data = transport,family = binomial(link = logit))

> summary(log.reg)
```

Min	1Q	Median	3Q	Max
-0.5808	-0.5808	-0.5808	-0.417	2.2293

The deviance of the transport and the engineer variables are acquired with the maximum residual of 2.22 and the error factor is less when compared with the higher possibilities in the maximum values of the general methods.

	Estimate	Std. Error	z value	Pr(>|z|)	
(Intercept)	-2.3979	0.3482	-6.887	5.68E-12	***
Engineer1	0.7037	0.3794	1.855	6.37E-02	.

The values are measured with the z value prediction of 1.855 and the significance values are measured with the lesser significance values for the measured coefficients in the AIC value is 355.17 in the DOF of 441. The coefficients are measured with the very less significance of the predictions.

> log.reg=glm(Transport~MBA,data = transport,family = binomial(link = logit))

> summary(log.reg)

Min	1Q	Median	3Q	Max
-0.5661	-0.5661	-0.5661	-0.476	2.1136

The deviance residuals are measured for the maximum value is 2.11 and the error are measured with the minimum value is about 0.56 and the 25% data are lesser significance in the values for the higher correlation in the dataset.

	Estimate	Std. Error	z value	Pr(>|z|)	
(Intercept)	-1.7501	0.1548	-11.31	<2e-16	***
MBA1	-0.3702	0.3425	-1.081	2.80E-01	

The coefficients are measured for the values in the z value and the variables are measured form lesser significance in the values and the variables are predicted with the AIC is 357.84 and the DOF is 441. The variables are MBA and transport are lesser significance in the variable regressions.

> log.reg=glm(Transport~WorkExp,data = transport,family = binomial(link = logit))

> summary(log.reg)

Min	1Q	Median	3Q	Max
-2.31003	-0.245	-0.14823	-0.069	3.00553

The variable Work Experience is highly significant with the maximum value in the deviance residuals is 3.00 which is highly predictable for the transport variable to produce the collinear in the variables.

	Estimate	Std. Error	z value	Pr(>|z|)	
(Intercept)	-6.53473	0.64158	-10.19	<2e-16	***
WorkExp	0.50727	0.05873	8.637	<2e-16	***

The work experience and the values are interpreted with the z value of 8.637 is highly correlated for the higher experience in the values are measured with higher significance of the variables measured in the factors of the AIC is 144.2 with iterations rate of 7. Work Experience can significantly make regression in the transport.

```
> log.reg=glm(Transport~Salary,data = transport,family = binomial(link = logit))
```

```
> summary(log.reg)
```

Min	1Q	Median	3Q	Max
-1.7715	-0.288	-0.2162	-0.157	2.4543

The variables are measured form the deviance of the transport and salary which is highly predictable in the maximum residuals of 2.45 and the values are interpreted for the higher significance of the measured values.

	Estimate	Std. Error	z value	Pr(>|z|)	
(Intercept)	-6.09359	0.52373	-11.64	<2e-16	***
Salary	0.20079	0.02111	9.511	<2e-16	***

The higher significance is predicted for the z value is 9.511 to the higher correlated value in the each variable for the predicting the regression in the AIC value of 157.71 and the lower AIC promotes the iteration values are predicted for the 6.

```
> log.reg=glm(Transport~Distance,data = transport,family = binomial(link = logit))
```

```
> summary(log.reg)
```

Min	1Q	Median	3Q	Max
-1.7369	-0.4958	-0.3052	-0.191	2.5867

The values are highly correlated for the variables measuring the values in the regression of the maximum deviance values is 2.58 and the distance variable is making the significant changes in the transport.

	Estimate	Std. Error	z value	Pr(>|z|)	
(Intercept)	-6.73568	0.70161	-9.6	< 2e-16	***
Distance	0.38066	0.04896	7.775	7.54E-15	***

The standard error of coefficients are measured with the values of 0.04 and the z value is impacted with the higher significance of the transport with the value of 7.775 of AIC is 274.4 in the DOF is about 441.

```
> log.reg=glm(Transport~license,data = transport,family = binomial(link = logit))
```

```
> summary(log.reg)
```

Min	1Q	Median	3Q	Max
-1.1127	-0.2797	-0.2797	-0.28	2.5538

The variables are regretted with the maximum deviance residuals for the value is 2.5538 and the variables are predicting the higher significance in the standard error of the predictions.

	Estimate	Std. Error	z value	Pr(>|z|)	
(Intercept)	-3.2219	0.2828	-11.39	<2e-16	***
license1	3.0678	0.3445	8.905	<2e-16	***

The license category is measuring the values are regretted for the higher significance of the z values in the measuring values of the 8.905 and the AIC is about 257.84 for the iterations 6.

```
> model=glm(Transport~.,data = transport,family = binomial(link = "logit"))
```

```
> model
```

(Intercept)	Age	Gender1	Engineer1	MBA1	WorkExp	Salary	Distance	license1
-71.054	2.2605	-1.7066	0.8569	-1.9357	-1.199	0.1852	0.4906	2.7085

The coefficients are measured with the negative interception of the variables and the positive coefficients are measured with the binomial factor. Since, the factor variable of the transport variables is change to the two level factors. The value placed for the interception and the coefficients are associated with the car users.

> summary(model)

Min	1Q	Median	3Q	Max
-1.99436	-0.042	-0.0072	-0.0005	2.27142

The deviance of the model is predicted with the higher regression value of 2.27142 and the values are measured with the various methods in the structured datasets of the values and the variables are measuring the values in the higher significance.

	Estimate	Std. Error	z value	Pr(>|z|)	
(Intercept)	-71.054	15.6669	-4.535	5.75E-06	***
Age	2.2605	0.5264	4.294	1.75E-05	***
Gender1	-1.7066	0.8336	-2.047	0.040631	*
Engineer1	0.8569	0.9138	0.938	0.348396	
MBA1	-1.9357	0.9094	-2.129	0.033285	*
WorkExp	-1.1989	0.3617	-3.315	0.000917	***
Salary	0.1852	0.072	2.573	0.010086	*
Distance	0.4906	0.1409	3.482	0.000499	***
license1	2.7085	0.8635	3.137	0.001709	**

The significance values are measured for the logistics regression in the predicted variables for checking the residuals the variables Age, Gender, Work Experience, Distance are highly significance and the values are measured above the standard deviation of 0.5 and the prediction are regretted with the negative significance on Work Experience with value is -3.315 in the variables. License variable is regretted with the significant in predicting the transport mode and the values are predicting the nature for the AIC value of 81.262 which is lower than the predicted function and the values are likelihood function in the values for the DOF of 434.

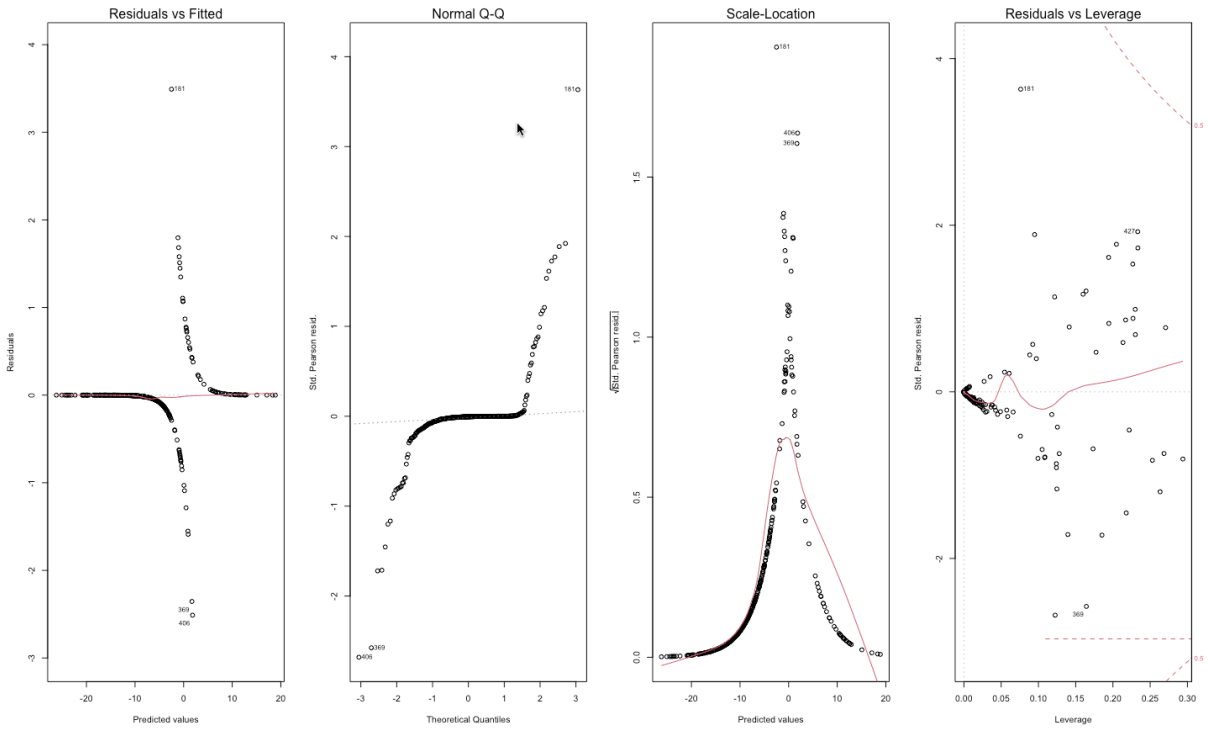
> car::vif(model)

Age	Gender	Engineer	MBA	WorkExp	Salary	Distance	license
11.903723	1.486422	1.112810	1.461142	16.950407	3.970501	1.714671	1.844857

The variance inflation factor is predicted the values are highly varied with the variables Age, Work Experience and the inflation rate are measured for the values with higher than 1 and the values are highly correlated for the measuring the variance of the inflating the total deviance and the residuals for the measured values. The inflation is low on Engineer and the values are varied in the standard error and Z value.

```
> par(mfrow=c(1,4))

> plot(model)
```



The plot shows residuals and fitted values are measured from the predicted variables in the measured values. The Predicted values are showing the fitted values are reaching the values in the highest prediction of the 181 and the residuals are measured to the negative trend of 406. The theoretical quantiles and the quantile function of the values are predicted for the measured values in between the 25% and the quantile can be achieved with the significance layer of the functions. The standard error value are coordinating at the point and the measures are influencing the values for the higher prediction in the slope of increasing and most of the values are influenced in the group cluster between 0 to 1. The error are identify the most prediction on the values are maintained in the value is 0.

```
> set.seed(50)

> splits=sample.split(transport$Transport,SplitRatio = 0.80)

> train=subset(transport,splits==TRUE)

> test=subset(transport,splits==FALSE)
```

The data are split into train and test dataset for the modelling and the performance matrices function using the validation of test datasets and the function on the train datasets for the prediction rate in the values.

```
> prop.table(table(test$Transport))
```

The proportion of the transport variables split in the test data is,

0	1
0.863636	0.13636

```
> prop.table(table(train$Transport))
```

The proportion of the train datasets is acquired in the function of the variables,

0	1
0.861972	0.13803

```
> trainmodel=glm(Transport~.,data = train,family = binomial(link = logit))
```

```
> summary(trainmodel)
```

Min	1Q	Median	3Q	Max
-2.14763	-0.03925	-0.00619	-0.00031	1.94353

The deviance residuals of the train model is predicted for the variables and the prediction is predicted for the various maximum value of 1.94353 and the values are predicted in the higher variance and the significance factor.

	Estimate	Std. Error	z value	Pr(>|z|)	
(Intercept)	-73.454	18.4868	-3.973	7.09E-05	***
Age	2.3452	0.62083	3.778	1.58E-04	***
Gender1	-2.28794	1.00648	-2.273	0.023014	*
Engineer1	0.51907	0.99153	0.524	0.600623	
MBA1	-1.84344	1.06597	-1.729	0.083746	.
WorkExp	-1.12545	0.43658	-2.578	0.00994	**
Salary	0.20044	0.08975	2.233	0.02553	*
Distance	0.4702	0.16469	2.855	0.004302	**
license1	2.63105	1.00263	2.624	0.008687	**

The higher significance of the train dataset is Age and the other variables are less significance in the train datasets. The predicted model and the train model are varied with the predictions of the significance values to maximise the error on the standard deviations.

The AIC value is calculated with 65.308 and the values are measured for the predicted variables in with DOF 346.

> car::vif(trainmodel)

Age	Gender	Engineer	MBA	WorkExp	Salary	Distance	license
9.109296	1.755454	1.095791	1.468506	12.328447	4.048327	1.896605	1.834372

The variance factors are measured for the variables in the various sources of the higher inflation factor on age and work experience. The higher inflation rate shows the variables are measured with the higher significance factor on the logit function.

> lrtest(trainmodel)

	#Df	LogLik	Df	Chisq	Pr(>Chisq)	
1	9	-23.654				
2	1	-142.486	-8	237.66	< 2.20E-16	***

The likelihood ratio has been performed for the prediction on the train model and the values are measured for the values and the predicting values in the various chi-squared value for 237.66.

> pR2(trainmodel)["McFadden"]

McFadden
0.8339917

The McFadden values are predicted for the variables and the value is higher r squared factor for the increased prediction. The data is stable for the various prediction and train dataset is significant to analyse the data in the model performance matrices and the ensemble method of predicting the independent variables.

> logLik(trainmodel)

'log Lik.' -23.65378 (df=9)

The log likelihood of the predicted variables are produced in the DOF is 9 and the log value is predicted with the values are generally negative reference of -23.65.

```
> trainmodel1=glm(Transport~1,data = train,family = binomial(link = logit))
```

```
> 1-(logLik(trainmodel)/logLik(trainmodel1))
```

'log Lik.' 0.8339917 (df=9)

The log likelihood of the trainmodel when the prediction are made on single variables for the higher significant values in the prediction techniques.

```
> testpredict=predict(trainmodel,newdata = test,type = "response")
```

```
> testpredict
```

The prediction is taken in the consideration for the validation of the train datasets and acquired the values are predicted with the rate above 0.5 and the values are converged with the transport mode.

```
> table(test$Transport,testpredict>0.5)
```

	FALSE	TRUE
0	72	4
1	1	11

Table shows that the prediction value of the transport is measured with the values 12 and the values are predicted for the major value in the datasets.

4. Prepare the data for analysis (SMOTE)

SMOTE is used to analyse the imbalance problems with the new minority classes with the over sampling and under sampling techniques.

```
> strain=subset(transport,splits==TRUE)
```

```
> stest=subset(transport,splits==FALSE)
```

The SMOTE analysis data is taken using the already split data and the subset is produced in the values and the variables are predicted for the various transport model.

> table(strain$Transport)

0	1
306	49

The table shows the transport mode of car and other mode count in the dataset.

> strain$Transport=as.factor(strain$Transport)

Rechecking the transport variables is predicted for the variables and the predicted values are measured for the transport variables and the values are measured in the numeric variables.

> balanced.transport=DMwR::SMOTE(Transport~.,as.data.frame(strain),perc.over = 200,k=5,perc.under = 200)

The balanced transport object is created for the SMOTE analysis with the prediction value i.e. the under fitting and over fitting value of the 200 and the K nearest neighbour is about 5 for the prediction over all variables in SMOTE.

> table(balanced.transport$Transport)

The balanced object of the transport variables is increased with the analysis of the imbalance dataset.

0	1
196	147

> sftrain=as.matrix(balanced.transport[,-c(2,3,4,8,9)])

> sltrain=as.matrix(balanced.transport$Transport)

The balanced SMOTE analysis is used in the prediction of the boosting techniques and the variables are removed with the factor levels. Since, the boosting techniques is implied with the numeric data and the model predicts the values are measured for the higher prediction in the transport model. The values are predicted in the matrix formulated data.

```
> smote.xgb=xgboost::xgboost(
+   data = sftrain,
+   label = sltrain,
+   eta = 0.7,
+   max_depth = 5,
+   nrounds = 50,
+   nfold = 5,
+   objective = "binary:logistic",
+   verbose = 0,
+   early_stopping_rounds = 10
+ )
> smote.xgb
```

iter	train_error
1	1.75E-02
2	0.005831

16	0
17	0.002915

The evaluation of the prediction and the values are predicted for the train error rate and the prediction for the value is 0.002915 as the ntrees are calculated with the prediction over the rate with the values and the measured variables are increased with the ntrees.

The parameters are organized and the values are increased for the general values in the various variables. The predicted values are measured for the all variables in the train datasets in the values and the values for the predicting subset are measured in values of ntrees.

```
> sftest=as.matrix(stest[,-c(2,3,4,8,9)])
```

```
> stest$pred.class.smote=predict(smote.xgb,sftest)

> stest$pred.class.smote=ifelse(stest$pred.class.smote<0.5,0,1)
```

The prediction of the SMOTE validation exercise is calculated for the confusion matrix in the values predicted for the generalized factors. The factors variables are removed and the variables are predicted with the confusion matrix for the better understanding on accuracy, sensitivity, specificity of the model in the SMOTE analysis.

```
> cm_smote=caret::confusionMatrix(data = factor(stest$pred.class.smote),
+                                 reference = factor(stest$Transport),
+                                 positive = "1")

> cm_smote
```

	Reference	
Prediction	0	1
0	69	0
1	7	12

The confusion matrix is predicted with the values that the interpretation shows the value of 19 car modes in the synthesis of the analysed data for the faster interpretation in the datasets.

```
Accuracy:0.9205
95%CI:(0.843,0.9674)
NoInformationRate:0.8636
P-Value[Acc>NIR]:0.07427
Kappa:0.7289
Mcnemar'sTestP-Value:0.02334
Sensitivity:1.0000
Specificity:0.9079
PosPredValue:0.6316
NegPredValue:1.0000
Prevalence:0.1364
DetectionRate:0.1364
DetectionPrevalence:0.2159
BalancedAccuracy:0.9539
'Positive'Class:1
```

The prediction values are measured for the reference and the prediction of the variables on the confusion matrix and the values are produced for the SMOTE analysis.

The KAPPA value is measured with 72.8% on the prediction of the SMOTE analysed data on the reference to the various factors. The sensitivity of the data is measured with the values are predicting the values for the 1 which is highly stable in the prediction. The specificity is increased for the 90% and the values with balanced accuracy for 95.3% which is highly predicted in the data validation.

```
> table.smote=table(stest$Transport,stest$pred.class.smote>=0.5)
```

```
> table.smote
```

	FALSE	TRUE
0	69	7
1	0	12

The table is predicted with the data validation and the total cars mode of transport using is 12.

```
> sum(stest$Transport==1 & stest$pred.class.smote>=0.5)
```

```
[1] 12
```

The predicted class also identify the same results in the values for the various table in predicting the nature of the general values in the control factor accuracy for the variables.

5. Create multiple models and explore how each model perform using appropriate model performance metrics (15 marks)

The model are created for the performance matrices in the values for the predicting of the variables.

KNN

KNN (K-Nearest Neighbour) is predicting the nearest neighbour from the K cluster. The variables are measuring the values are validating the classification of the majority vote by the calculation of Euclidean Distance.

```
> scale = preProcess(train, method = "range")
```

```
> train.norm = predict(scale, train)

> test.norm = predict(scale, test)
```

The train and test dataset for data validation is formulated for the analysis of the K nearest neighbour in the clustered group.

```
> knn = train(Transport ~., data = train.norm, method = "knn",

+             trControl = trainControl(method = "cv", number = 3),

+             tuneLength = 10)

> knn
```

The prediction is happens on 355 observations with class variables 0 and 1. 0 implies the employees without car mode of transport and 1 implies employees with car mode of transport.

k	Accuracy	Kappa
5	0.9408679	0.72156
7	0.9408441	0.70833
9	0.943669	0.71404
11	0.9464939	0.73053
13	0.9464939	0.73053
15	0.949295	0.74671
17	0.9464701	0.73529
19	0.938043	0.69769
21	0.938043	0.69769
23	0.9408679	0.70847

The table shows the Kappa and accuracy measures are predicted with the K cluster in the predicted variables for the KNN. The best fit for the KNN prediction is predicted as 15.

```
> knn$bestTune$k

[1] 15
```

The prediction value of the KNN is 15 and it can be used to predict the nearest variable of the transport.

```
>ktransport=knn(train.norm[,-c(2,3,4,8,9)],test.norm[,-c(2,3,4,8,9)],train.norm$Transport,k=15)
```

```
> class::knn(train.norm[,-c(2,3,4,8,9)],test.norm[,-c(2,3,4,8,9)],train.norm$Transport,k=15)
```

The KNN values are predicted for the levels on the variables in the prediction of the variables and the values are predicted for the measures of the independent variables.

```
[1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1
0 0
[52] 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 1 0 0 0 1 0 0 1 0 0 1 0 1 1 1 1
Levels: 0 1
```

```
> knnpred.test = predict(knn, newdata = test.norm[-9], type = "raw")
```

Prediction of KNN is predicted by the validation of the test dataset in the variables and the confusion matrix is produced with the help of the predicted variables in the dataset.

```
> cm_knn=confusionMatrix(knnpred.test,test.norm$Transport,positive="1")
```

```
> cm_knn
```

	Reference	
Prediction	0	1
0	76	5
1	0	7

The confusion matrix is used in analysed the factor variables are predicted with the car mode of transport is 7 and the values are mentioned in the values for the confusion matrix in the various models.

The confusion matrix predicted with the variables are predicted for the values measuring the values in accuracy of 94% and the Kappa value is measured as 70%. The dataset is stable in predicting the nature of the values are measured with the sensitivity as 58% and the specificity of the dataset is predicted for 1 and the values are measured in the basic of the balanced accuracy as 79%. The confusion matrix is used for the prediction in the values are measured on the independent variables classification.

Accuracy:0.9432
95%CI:(0.8724,0.9813)
NoInformationRate:0.8636
P-Value[Acc>NIR]:0.01456
Kappa:0.7074
Mcnemar'sTestP-Value:0.07364
Sensitivity:0.58333
Specificity:1.00000
PosPredValue:1.00000
NegPredValue:0.93827
Prevalence:0.13636
DetectionRate:0.07955
DetectionPrevalence:0.07955
BalancedAccuracy:0.79167
'Positive'Class:1

Naive Bayes

> NB = naiveBayes(x=train.norm[-c(2,3,4,8,9)], y=train.norm$Transport)

> NB

The Naïve Bayes is calculated for the measure of the transport variable in the factor variables and the predictors are implying in the measures of probabilities.

A-priori probabilities:	
train.norm$Transport	
0	1
0.8619718	0.1380282

The Naïve Bayes is predicted for the transport variables in the values for the 13% of the car variables are predicted in the values for the prediction of the variables.

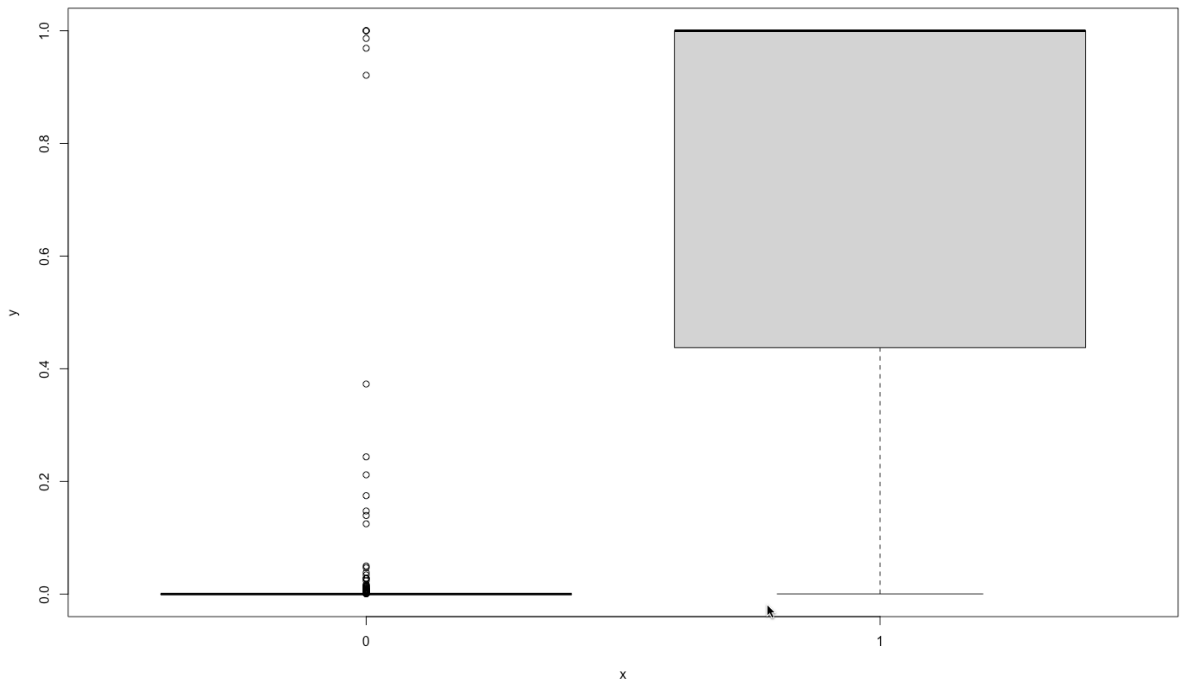
Conditional probabilities:		
Age		
train.norm$Transport	[,1]	[,2]
0	0.33869	0.1139726
1	0.70122	0.145167
WorkExp		
train.norm$Transport	[,1]	[,2]
0	0.19662	0.1237375
1	0.63776	0.2185356

Salary		
train.norm$Transport	[,1]	[,2]
0	0.12514	0.0906463
1	0.57232	0.2727289
Distance		
train.norm$Transport	[,1]	[,2]
0	0.36535	0.1602241
1	0.59285	0.1929781

The variables are separated for the analysis of the transport variable in the prediction of the datasets and the values are measuring in the other dependent variables. The ratio produced in the conditional probabilities are measured with the values for the age is predicted with the 21% of the car mode is used and the salary basis as reported with the values are predicted for the transport mode is predicted with the values in the 84% and the distance variable is predicted with 78%. The salary and work experience are measured same in the prediction of the variables for the prediction of variables with 84%.

```
> par(mfrow=c(1,1))
```

```
> plot(train.norm$Transport,NB.pred[,2])
```



The plot is used to understand the Naïve Bayes is predicted with the values are measured for the values based on the boxplot acquiring the 25% of the dataset in the train and validation is done on test datasets.

```
> NBpred.test = predict(NB, newdata = test.norm[-9])
```

```
> cm_NB=confusionMatrix(NBpred.test,test.norm$Transport,positive="1")
```

The prediction of the test dataset is validated with the confusion matrix for the best performance in accuracy, sensitivity, specificity and the Kappa values for the prediction values.

```
> cm_NB
```

	Reference	
Prediction	0	1
0	68	1
1	8	11

The statistics is showing the results are interpreted for the various models and the car mode is selected for 19.

```
Accuracy:0.8977
95%CI:(0.8147,0.9522)
NoInformationRate:0.8636
P-Value[Acc>NIR]:0.2232
Kappa:0.6514
Mcnemar'sTestP-Value:0.0455
Sensitivity:0.9167
Specificity:0.8947
PosPredValue:0.5789
NegPredValue:0.9855
Prevalence:0.1364
DetectionRate:0.1250
DetectionPrevalence:0.2159
BalancedAccuracy:0.9057
'Positive'Class:1
```

The confusion matrix is predicted with classes the transport variable is accuracy with 89% and the sensitivity is 91% and specificity is 89% which is the better model to predict the dependent

variables in the balanced accuracy of the dataset is 90% and the values are predicted for the values in the 90% and the Kappa value is predicted with 65% and the better model than KNN to predict the mode of the transport by car in the model.

Logistic Regression

```
> vif(glm(Transport~Age+WorkExp+Salary+Distance,data = train,family = binomial(link = logit)))
```

Age	WorkExp	Salary	Distance
4.383426	6.857621	2.652237	1.265631

The variance inflation are measured with the higher correlation on the work experience and the values are measured for the regression on the values are related with the logit function on the variables.

```
> summary(glm(Transport~Age+WorkExp+Salary+Distance,data = train,family = binomial(link = logit)))
```

Min	1Q	Median	3Q	Max
-2.59833	-0.10311	-0.02541	-0.00393	1.73786

The values are measured with the variables for the maximum variables are measured with 1.73 and the values are correlated for the higher significance in the datasets in the train dataset. The validation is based on the model on the test validation dataset.

	Estimate	Std. Error	z value	Pr(>|z|)	
(Intercept)	-56.4133	12.10376	-4.661	3.15E-06	***
Age	1.75281	0.41383	4.236	2.28E-05	***
WorkExp	-0.81684	0.3055	-2.674	0.0075	**
Salary	0.18043	0.06777	2.662	0.00776	**
Distance	0.37427	0.12438	3.009	0.00262	**

The variables with higher significance is based on the values on the z value and the standard deviation is based on the values in measuring the age and the values are based on the standard deviation on the higher significance for the values with 41% and 12% are based on the values in each category for the variables. The variables are highly significant values with the variables are age, work experience, salary and distance of the transport.

> logistics.car=glm(Transport~Age+WorkExp+Salary+Distance,data = train,family = binomial(link = logit))

Logistics model is created with the object based on the summary section of the variables are calculated.

> exp(coef(logistics.car))

(Intercept)	Age	WorkExp	Salary	Distance
3.16E-25	5.77E+00	4.42E-01	1.20E+00	1.45E+00

The exponential factors of the variables in the logistics model is acquired with the interception of the age with higher variable inflation factor and the variables (Work Experience, Salary and Distance) are measured for the lower values. The coefficients are based on the age factor for increasing the values in the independent variables.

> exp(coef(logistics.car))/(1+exp(coef(logistics.car)))

(Intercept)	Age	WorkExp	Salary	Distance
3.16E-25	8.52E-01	3.06E-01	5.45E-01	5.92E-01

The variables are measured for the higher coefficients with increasing factors for the age, work experience, salary and distance for the independent variables in the datasets.

> nrow(train[train$Transport==1,])/nrow(train)

[1] 0.1380282

The variables are measured with the mode of transport car is selected for the 13.8 in the train datasets.

> lrtest(logistics.car)

The likelihood test for the logistics model is accuracy in the likelihood function for the chi square values are predicted in with 224.33 in the values are examined in the values for the prediction in scores for the value predicted on the logistic model.

```
> pR2(logistics.car)["McFadden"]
```

McFadden
0.7871866

The McFadden values are examined in the variables and the value is acquired with 78% of the data are stable for the processing model of the logistics in the values. The McFadden values are predicted for the data stable in the validation on the model.

```
> logLik(logistics.car)
```

'log Lik.' -30.32284 (df=5)

The log likelihood function helps in understanding the data are experiencing the most DOF is 5 and the variables are predicted for the -30.32.

```
> logistics.pred=predict(logistics.car,data=train,type = "response")
```

```
> logistics.pred
```

The prediction is used in the validation of the datasets for the increased values in the prediction of the values for promoting the matrices in values for promoting the higher values.

```
> pred.num=ifelse(logistics.pred>0.5,1,0)
```

```
> pred.num
```

The prediction values are calculated in the values against 0.5, 0 and 1 as the prediction values exceeds the acquired value of the datasets.

```
> pred=factor(pred.num,levels = c(0,1),labels = c(0,1))
```

Converting the numeric predicted variables to the measured variables by factor levels for the transport facilities.

```
> pred.actual=train$Transport
```

The actual value are taken with the transport variables for the values are predicted with mode of transporting the values.

> cm_log_reg=confusionMatrix(pred,pred.actual,positive="1")

> cm_log_reg

	Reference	
Prediction	0	1
0	300	12
1	6	37

The confusion matrices helps in understanding the data in the transport mode of car is predicted with 43 then compared with other models for the prediction of transport uses among employees.

Accuracy:0.9493
95%CI:(0.921,0.9697)
NoInformationRate:0.862
P-Value[Acc>NIR]:7.209e-08
Kappa:0.7754
Mcnemar'sTestP-Value:0.2386
Sensitivity:0.7551
Specificity:0.9804
PosPredValue:0.8605
NegPredValue:0.9615
Prevalence:0.1380
DetectionRate:0.1042
DetectionPrevalence:0.1211
BalancedAccuracy:0.8677
'Positive'Class:1

The predicted confusion matrix allows them to predict the accuracy of the data model is 94% and the Kappa value is 77% on the datasets for the higher prediction of the value with sensitivity of the data is measured with 75% and the specificity of the variables are measured with 98% in the transport mode. The balanced accuracy for the dataset is predicted for 86% which the samples are validated in the train validated datasets for the variance in the factors.

The logistics model is created for the variables in the numeric factors and the variables values are measured from the confusion matrices on the models.

6. Apply both bagging and boosting modelling procedures to create 2 models and compare its accuracy with the best model of the above step.

Bagging

```
> bagtrain=train

> bagtest=test
```

The bagging validation and training datasets are same as the variables picked in the train and test datasets for the bagging analysis.

The bagging helps in reduce the variance on the prediction on the training datasets.

```
> Transport.bagging=bagging(Transport~.,data = bagtrain,

+                           control=rpart.control(maxdepth = 5,minsplit = 4))
```

The transport bagging object is created for the variables measured in the maximum splits and the minimum splits for the higher prediction in the training datasets in the variable sources for the measures.

The bagging methods takes the training dataset to provide the higher predictions on the 25 bootstrap aggregations on the variables produced in the variables.

```
> bagtest$pred.class.bag=predict(Transport.bagging,bagtest)
```

The prediction on the bagging data is evaluated with the validation data and the values are measured for the increased prediction in minimize the factors for the transportation mode of car.

```
> bagtest$pred.class.bag
```

```
 [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0
[52] 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 1 0 0 0 1 0 0 1 0 0 1 0 0 1 0 1 1 1 1
Levels: 0 1
```

The prediction values are varied with two level of variables for the bagging validation data.

```
> cm_bagging=confusionMatrix(data = factor(bagtest$pred.class.bag),
+                reference = factor(bagtest$Transport),
+                positive = "1")
> cm_bagging
```

	Reference	
Prediction	0	1
0	70	0
1	6	12

The confusion matrices produces the variables are selected by the transportation mode of variables in acquiring the values about 18.

```
Accuracy:0.9318
95%CI:(0.8575,0.9746)
NoInformationRate:0.8636
P-Value[Acc>NIR]:0.03552
Kappa:0.7609
Mcnemar'sTestP-Value:0.04123
Sensitivity:1.0000
Specificity:0.9211
PosPredValue:0.6667
NegPredValue:1.0000
Prevalence:0.1364
DetectionRate:0.1364
DetectionPrevalence:0.2045
BalancedAccuracy:0.9605
'Positive'Class:1
```

Accuracy of the variables are predicted with the confusion matrix for the values are increased for the 93% and the sensitivity of the data becomes 1. The variables are measuring the specificity of the variables for the 92% with Kappa values are 76%. The model predicted in the confusion matrices are mostly predicting the values are differed in the variables for the rate of the variables in the confusion matrices. The interpretation provides the values are highly increased in the variables factors of the 97% of the validation datasets.

```
> table.bagging=table(bagtest$Transport,bagtest$pred.class.bag==1)

> table.bagging
```

	FALSE	TRUE
0	70	6
1	0	12

The table identifies the transport mode of car is 12 on the predicted datasets of the variables.

Gradient Boosting

```
> sp=createDataPartition(transport$Transport,p=0.80,list = FALSE)

> gb.train=transport[sp,]

> gb.test=transport[-sp,]
```

The generalized boosting method is predicted using the package caret for the default ntrees, shrinkage, verbose values for the training datasets. Hence the data re split among the train and test using the caret package for the most predictable transport used by the customers.

```
> gbmfit=caret::train(Transport~.,
+                     data = gb.train,
+                     method = "gbm",
+                     trControl = trainControl(method = "repeatedcv",
+                                              number = 5,
+                                              repeats = 3,
+                                              verboseIter = FALSE),
+                     verbose = 0)
```

The GBM is measured using the package caret and the repeats value, cv number are based on the classification of the two levels segregation on the datasets.

> gbmfit

Stochastic Gradient Boosting
355 samples
8 predictor
2 classes: '0', '1'

The gradient boosting is predicted with the variables for 355 samples with 8 predictor and the class levels of 0 and 1.

The prediction is happens for the 5 times folds and repeated three time for the actual prediction in the assuming the sample values should be 284. The values are predicted with the rotation of the various datasets in the training model.

interaction.depth	n.trees	Accuracy	Kappa
1	50	0.9549937	0.8104138
1	100	0.9634317	0.8457799
1	150	0.9652836	0.8524081
2	50	0.9559196	0.813894
2	100	0.967122	0.8627766
2	150	0.9680744	0.8598913
3	50	0.9577979	0.8190862
3	100	0.9643707	0.847099
3	150	0.96531	0.8505851

The interaction depth model of the variables are repeated for three times. The interaction depth 1 shows the trees are predicted with three different layers with 50, 100 and 150 trees. The accuracy of the interaction depth 1 is highly predicted with the ntrees count of 150 and the similar results are appeared for the interaction depth 2 and 3 the average count shows the GBM model is 96% accuracy on the training datasets.

The n.trees with 150 are predicted in the shrinkage value of 0.1 and the interaction depth of the datasets is 2 and the minimum nodes generated is 10.

```
> cm_gb=caret::confusionMatrix(data = predict(gbmfit,gb.test),
+                              reference = gb.test$Transport)
> cm_gb
```

	Reference	
Prediction	0	1
0	76	0
1	0	12

The confusion matrices are measured with the values are generated for the values is about 12.

```
Accuracy:1
95%CI:(0.9589,1)
NoInformationRate:0.8636
P-Value[Acc>NIR]:2.495e-06
Kappa:1
Mcnemar'sTestP-Value:NA
Sensitivity:1.0000
Specificity:1.0000
PosPredValue:1.0000
NegPredValue:1.0000
Prevalence:0.8636
DetectionRate:0.8636
DetectionPrevalence:0.8636
BalancedAccuracy:1.0000
'Positive'Class:0
```

The confusion matrices are arranged with the variables for the prediction in the accuracy is 1 and the values with specificity, sensitivity and the kappa values are highly same in the predictions for the variables is 1 and the datasets are stable in same conditions for the values in the various predicting techniques for the model is reached with the higher prevalence for 86%.

Extreme Gradient Boosting

```
> xgbtrain=train
```

```
> xgbtest=test
```

The train and test validation datasets are taken form the splits of the original dataset in the variables for the predictions.

```
> xgbftrain=as.matrix(train[,-c(2,3,4,8,9)])
```

```
> xgbltrain=as.matrix(train[,9])
```

```
> xgbftest=as.matrix(test[,-c(2,3,4,8,9)])
```

The matrix vectors are created for the variables with features, labels for the training dataset. The labelled datasets are created form the transport mode of the employees and the features are taken from the numeric variables in the training dataset. The validation datasets are predicted with the values are differed from the category datasets.

```
> xgbfit=xgboost::xgboost(
+   data = xgbftrain,
+   label = xgbltrain,
+   eta = 0.001,
+   max_depth = 3,
+   min_child_weight = 3,
+   nrounds = 100,
+   nfold = 5,
+   objective = "binary:logistic",
+   verbose = 0,
+   early_stopping_rounds = 10
+ )
```

The Extreme Gradient Boosting object are created with the binary operations in the variables for the folds and the values are increased for the higher prediction list of the stopping rounds is 10, depth of the trees is fixed for 3 and the values are based on the training datasets.

```
> xgbfit
```

The iteration rates are measured with 0.03662 in all samples for the higher predictions on model.

```
> xgbtest$pred.class.xgb=predict(xgbfit,xgbftest)

> table.xgb=table(xgbtest$Transport,xgbtest$pred.class.xgb>0.5)

> table.xgb
```

The extreme gradient boosting techniques is based on the variables for the increased prediction in the value with 0.5 and the variables with different predictions in the transport variables is differed with validation datasets.

	FALSE	TRUE
0	73	3
1	1	11

The table shows the values are predicted for the datasets promoting the variables with 12.

```
> xgbtest$pred.class.xgb=ifelse(xgbtest$pred.class.xgb<0.5,0,1)

> cm_xgb=caret::confusionMatrix(data = factor(xgbtest$pred.class.xgb),

+                               reference = factor(xgbtest$Transport),

+                               positive = "1")
```

The prediction is taken in the validation datasets and the values are maintained in the prediction class below 0.5, 0 and 1 for the higher variance in the characters for the various predictions with more predictions in the independent variables. The confusion matrix is produced to check the sensitivity and accuracy of the datasets provided in the models.

```
> cm_xgb
```

	Reference	
Prediction	0	1
0	73	1
1	3	11

The variables are provided with the variables prediction of 14 observations in the datasets.

```
Accuracy:0.9545
95%CI:(0.8877,0.9875)
NoInformationRate:0.8636
P-Value[Acc>NIR]:0.00497
Kappa:0.8197
Mcnemar'sTestP-Value:0.61708
Sensitivity:0.9167
Specificity:0.9605
PosPredValue:0.7857
NegPredValue:0.9865
Prevalence:0.1364
DetectionRate:0.1250
DetectionPrevalence:0.1591
BalancedAccuracy:0.9386
'Positive'Class:1
```

The accuracy of the datasets are provided with the variables are predicted for the 95% and the sensitivity of the datasets are produced with the values for the higher prediction in the 91% and the specificity of the data variable is 96% of the Kappa value of 81% in the predicted datasets for the values in the higher prediction of the datasets in understanding the confusion matrix of the datasets in the values.

```
> sum(xgbtest$Transport==1 & xgbtest$pred.class.xgb>=0.5)
```

[1] 11

The sum of the variables is finally produced the 11 observations are factors in the major classification of the variables in the each category of the variables.

The parameters are changed in the category variables in the factors and the values are measured in the values for the higher prediction tuning parameters of the extreme gradient boosting.

```
> t.xgb=vector()

> l=c(0.001, 0.01, 0.1, 0.3, 0.5, 0.7, 1)

> m=c(1,3,5,7,9,15)

> n=c(2, 50, 100,1000,10000)
```

The tuning model is created with the dummy variables and the prediction for the understanding variables like values are measured with cluster variables for the predicted values are maintained with numeric values in the values.

```
> for (i in l) {
+   xgbfit=xgboost::xgboost(
+     data = xgbftrain,
+     label = xgbltrain,
+     eta = i,
+     max_depth = 5,
+     nrounds = 10,
+     nfold = 5,
+     objective = "binary:logistic",
+     verbose = 0,
+     early_stopping_rounds = 10
+   )
+   xgbtest$pred.class.xgb=predict(xgbfit,xgbftest)
+   t.xgb=cbind(t.xgb,sum(xgbtest$Transport==1 & xgbtest$pred.class.xgb>=0.5))
+ }
```

The variables for the tuning parameters are arranged in the values for the variables in measuring the values for the maintain the vectors are associated for the object created tuning extreme boosting. The parameters are controlled with same parameters in binary functions and the measures of the gradient boosting values are diversified in the variable factors.

```
> t.xgb
```

The tuning parameters are measured with the values in the variables with counts 12.

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]
[1,]	12	12	12	12	12	12	12

The parameters are measured with the value and the predictions are calculated for the series in 7 variables of the predicted values.

The tuning parameters and the vectors are understanding the features of the best fit in the variables for the higher predictions in the values.

```
> xgbfit=xgboost(
+   data = xgbftrain,
+   label = xgbltrain,
+   eta = 0.7,
+   max_depth = 5,
+   nrounds = 20,
+   nfold = 5,
+   objective = "binary:logistic",
+   verbose = 1,
+   early_stopping_rounds = 10
+ )
```

The extreme boosting methods enables the training parameters are not improved in the 10 levels and the values are measured for the basic train error parameters with 0.022 as the highest predictions in the datasets. The train datasets are measured for the values in the best iterations is 7 as the predicted variables is calculated as 0 and the parameters are tuned for the best fit.

```
> xgbtest$pred.class.xgb=predict(xgbfit,xgbftest)

> sum(xgbtest$Transport==1 & xgbtest$pred.class.xgb>=0.5)

[1] 12
```

The variables are predicted in the values and the prediction values are increased by the factors in the variables is 12. The methods are measuring in the prediction over 0.5.

```
> table.xgb=table(xgbtest$Transport,xgbtest$pred.class.xgb>=0.5)

> table.xgb
```

	FALSE	TRUE
0	71	5
1	0	12

The table is identified with the best fit in iterations for the variables is 12 in the transport model for the higher predictions in the variables.

Adaptive Boosting

The binding process of the weak learners are measured in the variables in the various factors in the values.

```
> adatrain=train

> adatest=test
```

The train and test datasets are split in the original dataset with ratio 80%.

```
> adaboost.fit=fastAdaboost::adaboost(Transport~.,data = as.data.frame(adatrain),nIter = 10)
```

The adaptive boosting is best in aggregating the variables in the various independent variables in each category of the ensemble methods. The variables are measured with maximum values are organized in the factors for the values in organized variables.

The object vector is created for the higher prediction in the variables and the variables are measured in the various performance matrices in the training datasets.

```
> adaboost.fit
```

The adaptive boosting values are predicted for the variables in the boosting techniques in the values measured with the n.trees function are developed in each variables.

```
No of trees:10
The weights of the trees are:1.7213731.8927441.2763671.2688331.0586691.2421780.9538021.0727781.0551540.870289
```

The weight of trees are measured with the variables in predicting is 10 and the values are decreasing with the weight of the trees.

```
> ada.pred=predict(adaboost.fit,newdata = adatest)

> cm_adaboost=caret::confusionMatrix(data = factor(ada.pred$class),

+                                    reference = factor(adatest$Transport),

+                                    positive = "1")
```

The prediction variables are predicted for the validation of the datasets and the values are measured in the variables for the prediction on the various factors for the higher increased factors for the transport variables.

```
> cm_adaboost
```

	Reference	
Prediction	0	1
0	71	0
1	5	12

The prediction variables are interpreted with the confusion matrix of the variables in the table with 17 observations.

The accuracy of the variables are measured from the variables with accuracy is 94% with the sensitivity of 1 and the specificity is measured with the values 93% in the variables for the prediction of the higher values. The balanced accuracy value is 96% and the values are highly interpreted for the Kappa values is 79% in the variables.

```
Accuracy:0.9432
95%CI:(0.8724,0.9813)
NoInformationRate:0.8636
P-Value[Acc>NIR]:0.01456
Kappa:0.7948
Mcnemar'sTestP-
Value:0.07364
Sensitivity:1.0000
Specificity:0.9342
PosPredValue:0.7059
NegPredValue:1.0000
Prevalence:0.1364
DetectionRate:0.1364
DetectionPrevalence:0.1932
BalancedAccuracy:0.9671
'Positive'Class:1
```

Model Comparison for Ensemble Methods

```
> modelcomparison=c("cm_bagging","cm_gb","cm_xgb","cm_smote","cm_adaboost")
```

The model comparison is used to identify the better in measuring the confusion matrix values for the higher possible variables for the class predictors with the ensemble methods techniques in achieving the maximum results in comparing the best model values.

The vector is created with all confusion matrix objects and the variables are treated as per the sensitivity, specificity, random recall and the accuracy of the models.

```
> modelcomparison
```

```
[1] "cm_bagging"  "cm_gb"      "cm_xgb"     "cm_smote"   "cm_adaboost"
```

The model comparison object is successfully created with the variables of the confusion matrix.

```
> table_ensemble=data.frame(Sensitivity = NA,
+                           Specificity = NA,
+                           Precision = NA,
```

```
+                 Recall = NA,
+                 F1 = NA)
```

The dummy table is created for the sensitivity, recall, specificity, precision and the F1 of the confusion matrix are predicted with the values for the best model comparison in the various values.

```
> for (i in seq_along(modelcomparison)) {
+   model=get(modelcomparison[i])
+   a=data.frame(Sensitivity = model$byClass["Sensitivity"],
+                Specificity = model$byClass["Specificity"],
+                Precision = model$byClass["Precision"],
+                Recall = model$byClass["Recall"],
+                F1 = model$byClass["F1"])
+   rownames(a)=NULL
+   table_ensemble=rbind(table_ensemble,a)
+ }
```

The variables are created with the values for the various structures on the class model prediction on the values for the various predicting vectors in the places for the sequence model predicted in the values are measured with the five main prediction in the variables.

```
> table_ensemble=table_ensemble[-1,]
> row.names(table_ensemble)=c("BAGGING","GBM","XGB","SMOTE","ADABOOST")
```

The table created for the variables are changed in the predictions for the various values in the row names inserting the values for the model compared in the datasets. The model is helpful in understanding the variables for the various methods in the interaction of depths.

> table_ensemble

	Sensitivity	Specificity	Precision	Recall	F1
BAGGING	1.0000000	0.9210526	0.6666667	1.0000000	0.8000000
GBM	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000
XGB	0.9166667	0.9605263	0.7857143	0.9166667	0.8461538
SMOTE	1.0000000	0.9078947	0.6315789	1.0000000	0.7741935
ADABOOST	1.0000000	0.9342105	0.7058824	1.0000000	0.8275862

The table is showing the models compared in the various variables and the factors of the independent variables in creating the model for the prediction on the values for the higher predicted values.

The table identifies the Recall and Sensitivity of the Bagging, GBM, SMOTE and ADBOOST are same in predicting the conditions of the variables in the predicting features of the variables as the models are predicted in the same predictions of the values.

The GBM is predicted with the same in the model comparison as it helps in trying the fit of next residuals in the models predicted for XGB in the ensemble methods. The best fit of the model comparison is identified with GBM and the next fit of the model is predicted with the values are making the nature of the values in the Adaptive Boosting of the ensemble techniques.

Model Comparison for Model Performance Matrices

> modelcomp=c("cm_knn","cm_NB")

The model performance matrices are making the best comparison when compared with the cluster and the reduced factors for the various variables in the measuring features of the variables.

The modelcomp is the vector created for the model comparison parameter in the comparing the model of the confusion matrix

> modelcomp

```
[1] "cm_knn" "cm_NB"
```

The model comparison is taken in KNN and Naïve Bayes confusion matrix.

```
> table_modelcomp=data.frame(Sensitivity = NA,
+                            Specificity = NA,
+                            Precision = NA,
+                            Recall = NA,
+                            F1 = NA)
```

The another table with dummy variables are created for the model comparison with sensitivity, specificity, recall, precision and F1 of the variables in the models.

```
> for (i in seq_along(modelcomp)) {
+   model1=get(modelcomp[i])
+   b=data.frame(Sensitivity = model1$byClass["Sensitivity"],
+                Specificity = model1$byClass["Specificity"],
+                Precision = model1$byClass["Precision"],
+                Recall = model1$byClass["Recall"],
+                F1 = model1$byClass["F1"])
+   rownames(b)=NULL
+   table_modelcomp=rbind(table_modelcomp,b)
+ }
```

The model comparison is based on the various factors in the confusion matrix analysis of the factor variables in the measures of the sequence order in the various variations on the row names of the table constructed as the dummy variables.

The variables are treated with the confusion matrix model comparison in the values of the higher prediction values.

> table_modelcomp=table_modelcomp[-1,]

> row.names(table_modelcomp)=c("KNN","NAIVE BAYES")

The row names are created for the model comparison in the table comparison of the variables in the values.

> table_modelcomp

	Sensitivity	Specificity	Precision	Recall	F1
KNN	0.5833333	1.0000000	1.0000000	0.5833333	0.7368421
NAIVE BAYES	0.9166667	0.8947368	0.5789474	0.9166667	0.7096774

The table helps in identifying the variables are created in the confusion matrix with sensitivity and the recall values are measuring the same values in the model comparison of the confusion matrix.

The maximum predicted variables for the best acquired model in performance matrix is predicted with best model as Naïve Bayes in understanding the variables to produce the variations in the values are organized in the sensitivity, specificity and the recall values of the various model functions in the confusion matrix.

7. Summarize your findings from the exercise in a concise yet actionable note

- Exploratory Data Analysis helps in making the prediction on the mode of the transport that employees are preferred to use to reach office. The datasets are created for the car transport by the employees to reach the office. The variables are predicted with the numeric variables and category variables to distinguish the inferences on the car usage as transport medium. The Car transport variable is predicted with employee basic details and their designations on the organizations.
- Transport medium through 2 wheeler and Public Transport is the major employees using more than car, this makes the transport medium to create the two level factors and the insights provided that the employees using car as a medium of transport, the employee might be an Engineer with 5 to 8 years' work experience and earning about 10 to 20 lacs per annum as the predicted variables are ensure that the cars are used by those employees rather than a MBA graduate and the higher experience employee in the organizations.

- Logistics Regression is executed for the analysis to predict the transport variable are highly correlated and significant with the variables that are making differences in the employee disturbance of using car. The Regression techniques makes the model in predicting that car medium of transport is highly significant in the affecting Age of the employees. The Logistics regression predicted the variables age, work experience, salary are highly significant in making the employees to move on the transport facilities as car. The regression is achieved with the accuracy of 70% employees preferring car as the important factor of age.
- The logistics implies on the factor which helps in identify the employees are highly significant modules on the age and salary in the likelihood function for the best acquired model. The likelihood of the regression techniques is analysed with the binomial distribution of factors and the values are changed with the predictions over their function of employees are better in using car transport in basic need of age factor and the salary factor. This makes an interesting fact that the models created are highly significant in EDA for assuming the car mode of transport is likely in prediction of 70%.
- The model performance are analysed the variables by techniques like KNN and Naïve Bayes which gives the insights in the employees transport facility is correlated with other cluster group of variables in predicting the clustering techniques. The K cluster shows the near factors of the transport variables are predicted as 15 cluster nearer for the predictions in the car medium of transport. Accuracy on the confusion matrix is produced with the prediction of 90% employees are making the car transport as the medium of travel.
- Model performance of the predicted variables are mostly significant in the numeric variables and measures taken in transport mode are predicted with the sensitivity of 94% and the specificity of 92% in comparing the models, the predictors helps in the future prediction of the employees are use the car mode of transport with the calculation acquired in the regression technique for the models in the predicted variables.
- The Synthetic Minority Techniques is used in the predictions of the variables are based on the oversampling methods of the factor variables in transport facility. The SMOTE analysis is provide that the employees using the car medium of transport are nearly 30% in the accuracy of the predicted variables. The employees underfitting the nature of the usage of car is more than predicted values and the employees using the values in the assumptions of transport facility is increased with the values in the predicted variables for the transport mode and the employees without preferring the values is about 70%.

- Ensemble techniques are used in the acquired values for decreases the error and the Gradient Boosting is making the parallel decision on the variables. The Transport variable is measured in the parallel decision with the iterations for the employees choosing the car mode as transport facility, as the Gradient Boosting is making the transport variables to decreases deviance in the variables for the prediction in the car mode of transport. The Gradient Boosting is stable in predicting the variables are calculated are 100% in accuracy for the employees using car in transport.
- Deviance is reduced again in the extreme boosting techniques, where transport variables are tuned as per our convenience of the prediction variables in the models. The Extreme Gradient Boosting is helps in the identification of the variables with higher prediction rates shows that employees are preferring 94% of the car transport with their basic details like designation and salary of the employees. The tuning parameters are classified in the best approach to showcase the values are again returned with 90% accuracy in the model matrices.
- The Adaptive Boosting helps in identifying the variables are measured the correct fit of the variables in the transport mode. The transport variable is predicted with function on the minimize the error on the unbalanced structures of the variables for the predicting feature in higher variance of the 82% in the produced transport mode. The minimized unbalanced transport variable is predicted with employees are using the car with lesser usage than the prediction on the public transport usage of the factors.
- Bagging is the group of cluster in which the under fitting model, over fitting model of the variables are easily identified in the occurrence in the measures of the employees in preferred towards car mode. The training data of the employees transport facility is highly predict in the various variables with the weak learners. The weak predictions on the variables are measured in the people technique towards the values are monitored with accuracy for about 92% in achieving the major usage of weak learning variables.
- The Model Comparison of the confusion matrix of ensemble methods and model performance matrices are measured in the employees identified with the usage of car in various techniques for the selecting the transport facility as car. This ensures the model comparison insights the employees using car averagely is based on age, salary and the less significance cases of work experience in the organization. The Model matrices are performed in reducing the errors and the deviance of the variables of the transport variable predicting the employees variable in the datasets.

3.5 Outlier Identification

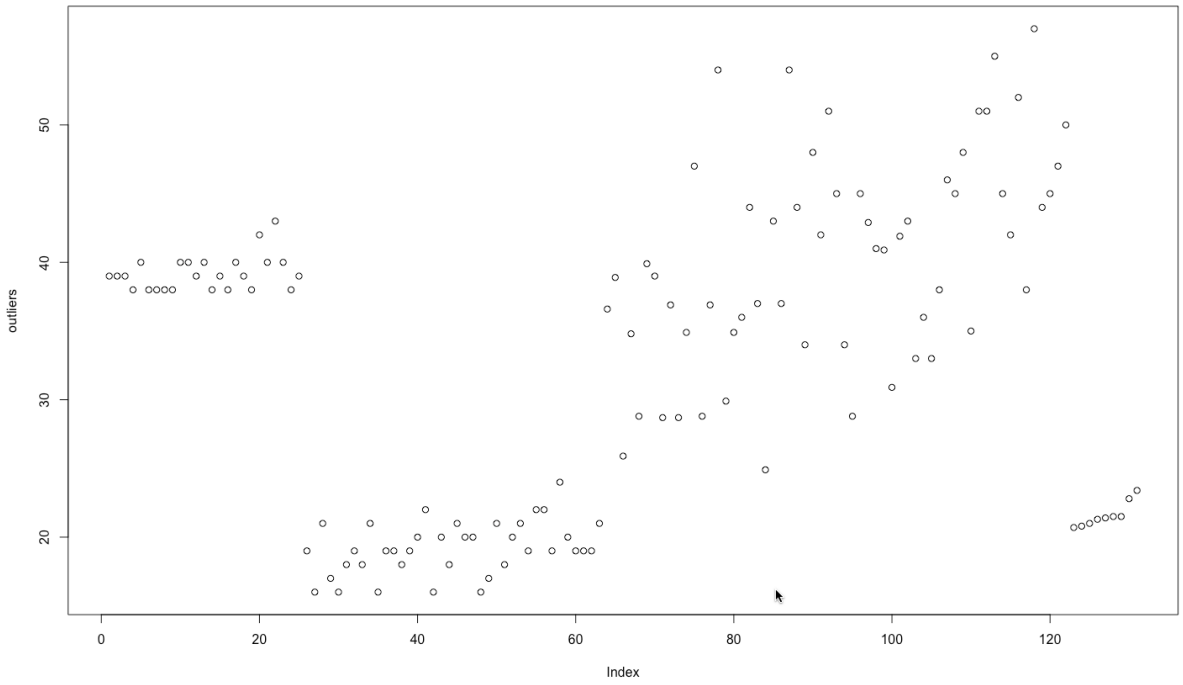
The outlier identification for the temperature is found as the maximum values in the variables.

```
> outliers=boxplot(num.data,plot = FALSE)$out
> outliers
```

```
 [1] 39.0 39.0 39.0 38.0 40.0 38.0 38.0 38.0 38.0 40.0 40.0 39.0 40.0 38.0 39.0 38.0 40.0 39.0 38.0 42.0
[21] 40.0 43.0 40.0 38.0 39.0 19.0 16.0 21.0 17.0 16.0 18.0 19.0 18.0 21.0 16.0 19.0 19.0 18.0 19.0 20.0
[41] 22.0 16.0 20.0 18.0 21.0 20.0 20.0 16.0 17.0 21.0 18.0 20.0 21.0 19.0 22.0 22.0 19.0 24.0 20.0 19.0
[61] 19.0 19.0 21.0 36.6 38.9 25.9 34.8 28.8 39.9 39.0 28.7 36.9 28.7 34.9 47.0 28.8 36.9 54.0 29.9 34.9
[81] 36.0 44.0 37.0 24.9 43.0 37.0 54.0 44.0 34.0 48.0 42.0 51.0 45.0 34.0 28.8 45.0 42.9 41.0 40.9 30.9
[101] 41.9 43.0 33.0 36.0 33.0 38.0 46.0 45.0 48.0 35.0 51.0 51.0 55.0 45.0 42.0 52.0 38.0 57.0 44.0 45.0
[121] 47.0 50.0 20.7 20.8 21.0 21.3 21.4 21.5 21.5 22.8 23.4
```

The outliers are plotted in the charts as per the groups and the variables are identified with the outliers present in the datasets.

The maximum outliers are found in Age and Salary variable. The clusters are making the outliers in the datasets.



3.6 Variable Transformation/ Feature creation

```
> transport$Transport=as.character(transport$Transport)
> transport$Transport[transport$Transport %in% "2Wheeler"]="0"
> transport$Transport[transport$Transport %in% "Car"]="1"
> transport$Transport[transport$Transport %in% "Public Transport"]="0"
> transport$Gender=as.character(transport$Gender)
> transport$Gender[transport$Gender %in% "Female"]="0"
> transport$Gender[transport$Gender %in% "Male"]="1"
```

The variables (Transport and Gender) are transformed as per the binary values like 0 and 1. The main transformation for the variable transport into two level factor is helping the whole datasets to predicts the nature of the logistics regression in binomial function.

As the transport variables are treated with three level factors and the variables are converted to two level character for the future analysis and the prediction class. The variables are segregated as 0 for 2 wheeler, Public Transport and 1 for Car. Since, the character of the variables are segregated for the two level factors, the car usage of employee can be easily predicted.

Since, the analysis requires all the factor variables should be in 0 and 1 values, the gender variable also classified with 0 as female and 1 male in the gender category variables.

```
> summary(transport$Transport)
```

```
  0   1
382  61
```

```
> summary(transport$Gender)
```

```
  0   1
127 316
```

4. Conclusion

Employees are reaching the office by 2 wheeler, public transport and car as mode of transport. In this analysis, employees are predicted with usage of car as mode of transport as per their convenience. The conclusion may predict the employees finished Engineering and earned MBA graduation with 5 to 8 years' work experience and their designation salary will look for the transport facility as car and prediction makes that nearly 75% employees will use car as their mode of transport to reach their office. This prediction may fall over on the decision of the fresher employees in the organization. While predicting this car mode of transport employees of older age with higher work experience and higher salary will use the car as mode of transport.

5. Appendix

```
setwd("/Users/numerp/Documents/PGP-BABI/Module 6   Machine Learning/Project 5")
getwd()
library(readr)
library(dplyr)
library(psych)
library(car)
library(carData)
library(ggplot2)
library(mice)
library(lattice)
library(nFactors)
library(scatterplot3d)
library(data.table)
library(tidyverse)
library(broom)
library(GGally)
transport=read_csv("Cars_edited.csv",col_names = TRUE)
names(transport)
names(transport)[5]="WorkExp"
names(transport)
transport
str(transport)
summary(transport)
dim(transport)
any(is.na(transport))
transport=na.omit(transport)
dim(transport)
summary(transport)
transport$Gender=as.factor(transport$Gender)
transport$Engineer=as.factor(transport$Engineer)
transport$MBA=as.factor(transport$MBA)
transport$license=as.factor(transport$license)
transport$Transport=as.factor(transport$Transport)
str(transport)
summary(transport)
transport$Transport=as.character(transport$Transport)
transport$Transport[transport$Transport %in% "2Wheeler"]="0"
transport$Transport[transport$Transport %in% "Car"]="1"
transport$Transport[transport$Transport %in% "Public Transport"]="0"
str(transport$Transport)
transport$Gender=as.character(transport$Gender)
transport$Gender[transport$Gender %in% "Female"]="0"
transport$Gender[transport$Gender %in% "Male"]="1"
str(transport$Gender)
transport$Transport=as.factor(transport$Transport)
transport$Gender=as.factor(transport$Gender)
str(transport)
```

```
summary(transport)
summary(transport$Transport)
summary(transport$Gender)
head(transport,3)
tail(transport,4)
ct.data=subset(transport,select = c(Gender,Engineer,MBA,license))
num.data=subset(transport,select = -c(Gender,Engineer,MBA,license,Transport))
names(ct.data)
names(num.data)
by(transport,INDICES = transport$Transport,FUN = summary)
ggpairs(transport[,c("Age","WorkExp","Salary","Distance")],
        ggplot2::aes(colour=as.factor(transport$Transport)))
outliers=boxplot(num.data,plot = FALSE)$out
outliers
plot(outliers)
correlation=cor(num.data)
correlation
corrplot::corrplot(correlation,method = "circle",type = "upper")
par(mfrow=c(2,2))
hist(transport$Age,main = "Age of the employees",col = "grey")
hist(transport$WorkExp,main = "Work Experience",col = "red")
hist(transport$Salary,main = "Salary of the employees",col = "yellow")
hist(transport$Distance,main = "Distance Travelling",col = "green")
par(mfrow=c(2,2))
boxplot(transport$Age,horizontal = T,main = "Age of the employees",col = "grey")
boxplot(transport$WorkExp,horizontal = T,main = "Work Experience",col = "red")
boxplot(transport$Salary,horizontal = T,main = "Salary of the employees",col = "yellow")
boxplot(transport$Distance,horizontal = T,main = "Distance Travelling",col = "green")
par(mfrow=c(2,2))
for(i in names(ct.data)){
  print(i)
  print(round(prop.table(table(transport$Transport,ct.data[[i]])),digits = 3)*100)
  barplot(table(transport$Transport,ct.data[[i]]),
          col = c("yellow","blue"),
          main = names(ct.data[i]))
}
attach(transport)
log.reg=glm(Transport~Age,data = transport,family = binomial(link = logit))
summary(log.reg)
log.reg=glm(Transport~Gender,data = transport,family = binomial(link = logit))
summary(log.reg)
log.reg=glm(Transport~Engineer,data = transport,family = binomial(link = logit))
summary(log.reg)
log.reg=glm(Transport~MBA,data = transport,family = binomial(link = logit))
summary(log.reg)
log.reg=glm(Transport~WorkExp,data = transport,family = binomial(link = logit))
summary(log.reg)
log.reg=glm(Transport~Salary,data = transport,family = binomial(link = logit))
summary(log.reg)
log.reg=glm(Transport~Distance,data = transport,family = binomial(link = logit))
```

```
summary(log.reg)
log.reg=glm(Transport~license,data = transport,family = binomial(link = logit))
summary(log.reg)
model=glm(Transport~.,data = transport,family = binomial(link = "logit"))
model
summary(model)
car::vif(model)
par(mfrow=c(1,4))
plot(model)
#splitting data
library(caTools)
set.seed(50)
splits=sample.split(transport$Transport,SplitRatio = 0.80)
train=subset(transport,splits==TRUE)
test=subset(transport,splits==FALSE)
prop.table(table(transport$Transport))
prop.table(table(test$Transport))
prop.table(table(train$Transport))
trainmodel=glm(Transport~.,data = train,family = binomial(link = logit))
summary(trainmodel)
car::vif(trainmodel)
library(lmtest)
lrtest(trainmodel)
library(pscl)
pR2(trainmodel)["McFadden"]
logLik(trainmodel)
trainmodel1=glm(Transport~1,data = train,family = binomial(link = logit))
1-(logLik(trainmodel)/logLik(trainmodel1))
logLik(trainmodel1)
testpredict=predict(trainmodel,newdata = test,type = "response")
testpredict
table(test$Transport,testpredict>0.5)
#**************************************************************************
**********************
#logistics regression
library(caret)
vif(glm(Transport~Age+WorkExp+Salary+Distance,data = train,family = binomial(link =
logit)))
summary(glm(Transport~Age+WorkExp+Salary+Distance,data = train,family =
binomial(link = logit)))
logistics.car=glm(Transport~Age+WorkExp+Salary+Distance,data = train,family =
binomial(link = logit))
logistics.car
exp(coef(logistics.car))
exp(coef(logistics.car))/(1+exp(coef(logistics.car)))
nrow(train[train$Transport==1,])/nrow(train)
lrtest(logistics.car)
pR2(logistics.car)["McFadden"]
logLik(logistics.car)
logistics.pred=predict(logistics.car,data=train,type = "response")
```

```
logistics.pred
pred.num=ifelse(logistics.pred>0.5,1,0)
pred.num
pred=factor(pred.num,levels = c(0,1),labels = c(0,1))
pred
pred.actual=train$Transport
pred.actual
cm_log_reg=confusionMatrix(pred,pred.actual,positive="1")
cm_log_reg
#**************************************************************************
************************
#knn
library(class)
scale = preProcess(train, method = "range")
train.norm = predict(scale, train)
test.norm = predict(scale, test)
knn = train(Transport ~., data = train.norm, method = "knn",
            trControl = trainControl(method = "cv", number = 3),
            tuneLength = 10)
knn
knn$bestTune$k
class::knn(train.norm[,-c(2,3,4,8,9)],test.norm[,-c(2,3,4,8,9)],train.norm$Transport,k=15)
ktransport=knn(train.norm[,-c(2,3,4,8,9)],test.norm[,-
c(2,3,4,8,9)],train.norm$Transport,k=15)
table(test.norm$Transport,ktransport)
knnpred.train = predict(knn, data = train.norm[-9], type = "raw")
confusionMatrix(knnpred.train,train.norm$Transport,positive="1")
knnpred.test = predict(knn, newdata = test.norm[-9], type = "raw")
knnpred.test
cm_knn=confusionMatrix(knnpred.test,test.norm$Transport,positive="1")
cm_knn
#**************************************************************************
************************
#Naive Bayes
library(e1071)
NB = naiveBayes(x=train.norm[-c(2,3,4,8,9)], y=train.norm$Transport)
NB
NB.pred=predict(NB,type = "raw",newdata = train.norm)
NB.pred
par(mfrow=c(1,1))
plot(train.norm$Transport,NB.pred[,2])
NBpred.train = predict(NB, newdata = train.norm[-9])
confusionMatrix(NBpred.train, train.norm$Transport,positive="1")
NBpred.test = predict(NB, newdata = test.norm[-9])
cm_NB=confusionMatrix(NBpred.test,test.norm$Transport,positive="1")
cm_NB
#**************************************************************************
************************
#Modelling - Bagging, Boosting, SMOTE
#Bagging
```

```
library(xgboost)
library(ipred)
library(rpart)
library(caret)
bagtrain=train
bagtest=test
Transport.bagging=bagging(Transport~.,data = bagtrain,
                          control=rpart.control(maxdepth = 5,minsplit = 4))
bagtest$pred.class.bag=predict(Transport.bagging,bagtest)
bagtest$pred.class.bag
cm_bagging=confusionMatrix(data = factor(bagtest$pred.class.bag),
                reference = factor(bagtest$Transport),
                positive = "1")
cm_bagging
table.bagging=table(bagtest$Transport,bagtest$pred.class.bag==1)
table.bagging
#**************************************************************************
************************
#Gradient Boosting using CARET Package. Since GBM package is not working properly.
library(caret)
sp=createDataPartition(transport$Transport,p=0.80,list = FALSE)
gb.train=transport[sp,]
gb.test=transport[-sp,]
gbmfit=caret::train(Transport~.,
                    data = gb.train,
                    method = "gbm",
                    trControl = trainControl(method = "repeatedcv",
                                             number = 5,
                                             repeats = 3,
                                             verboseIter = FALSE),
                    verbose = 0)
gbmfit
cm_gb=caret::confusionMatrix(data = predict(gbmfit,gb.test),
                             reference = gb.test$Transport)
cm_gb
#**************************************************************************
************************
#Extreme Gradient Boosting
library(xgboost)
xgbtrain=train
xgbtest=test
xgbftrain=as.matrix(train[,-c(2,3,4,8,9)])
xgbltrain=as.matrix(train[,9])
xgbftest=as.matrix(test[,-c(2,3,4,8,9)])
xgbfit=xgboost::xgboost(
  data = xgbftrain,
  label = xgbltrain,
  eta = 0.001,
  max_depth = 3,
  min_child_weight = 3,
```

```
  nrounds = 100,
  nfold = 5,
  objective = "binary:logistic",
  verbose = 0,
  early_stopping_rounds = 10
)
xgbfit
xgbtest$pred.class.xgb=predict(xgbfit,xgbftest)
table.xgb=table(xgbtest$Transport,xgbtest$pred.class.xgb>0.5)
table.xgb
xgbtest$pred.class.xgb=ifelse(xgbtest$pred.class.xgb<0.5,0,1)
cm_xgb=caret::confusionMatrix(data = factor(xgbtest$pred.class.xgb),
                              reference = factor(xgbtest$Transport),
                              positive = "1")
cm_xgb
sum(xgbtest$Transport==1 & xgbtest$pred.class.xgb>=0.5)
#Extreme Gradient Boosting Tuning
t.xgb=vector()
l=c(0.001, 0.01, 0.1, 0.3, 0.5, 0.7, 1)
m=c(1,3,5,7,9,15)
n=c(2, 50, 100,1000,10000)
for (i in l) {
  xgbfit=xgboost::xgboost(
    data = xgbftrain,
    label = xgbltrain,
    eta = i,
    max_depth = 5,
    nrounds = 10,
    nfold = 5,
    objective = "binary:logistic",
    verbose = 0,
    early_stopping_rounds = 10
  )
  xgbtest$pred.class.xgb=predict(xgbfit,xgbftest)
  t.xgb=cbind(t.xgb,sum(xgbtest$Transport==1 & xgbtest$pred.class.xgb>=0.5))
}
t.xgb
#Best Fit
xgbfit=xgboost(
  data = xgbftrain,
  label = xgbltrain,
  eta = 0.7,
  max_depth = 5,
  nrounds = 20,
  nfold = 5,
  objective = "binary:logistic",
  verbose = 1,
  early_stopping_rounds = 10
)
xgbtest$pred.class.xgb=predict(xgbfit,xgbftest)
```

```
sum(xgbtest$Transport==1 & xgbtest$pred.class.xgb>=0.5)
table.xgb=table(xgbtest$Transport,xgbtest$pred.class.xgb>=0.5)
table.xgb
#**************************************************************************
**********************
#Adaptive Boosting
library(fastAdaboost)
adatrain=train
adatest=test
str(adatrain$Transport)
adaboost.fit=fastAdaboost::adaboost(Transport~.,data = as.data.frame(adatrain),nIter = 10)
adaboost.fit
ada.pred=predict(adaboost.fit,newdata = adatest)
cm_adaboost=caret::confusionMatrix(data = factor(ada.pred$class),
                           reference = factor(adatest$Transport),
                           positive = "1")
cm_adaboost
#**************************************************************************
************************
#SMOTE
library(DMwR)
table(transport$Transport)
strain=subset(transport,splits==TRUE)
stest=subset(transport,splits==FALSE)
table(strain$Transport)
strain$Transport=as.factor(strain$Transport)
balanced.transport=DMwR::SMOTE(Transport~.,as.data.frame(strain),perc.over =
200,k=5,perc.under = 200)
table(balanced.transport$Transport)
sftrain=as.matrix(balanced.transport[,-c(2,3,4,8,9)])
sltrain=as.matrix(balanced.transport$Transport)
smote.xgb=xgboost::xgboost(
  data = sftrain,
  label = sltrain,
  eta = 0.7,
  max_depth = 5,
  nrounds = 50,
  nfold = 5,
  objective = "binary:logistic",
  verbose = 0,
  early_stopping_rounds = 10
)
smote.xgb
sftest=as.matrix(stest[,-c(2,3,4,8,9)])
stest$pred.class.smote=predict(smote.xgb,sftest)
stest$pred.class.smote=ifelse(stest$pred.class.smote<0.5,0,1)
cm_smote=caret::confusionMatrix(data = factor(stest$pred.class.smote),
                        reference = factor(stest$Transport),
                        positive = "1")
cm_smote
```

```
table.smote=table(stest$Transport,stest$pred.class.smote>=0.5)
table.smote
sum(stest$Transport==1 & stest$pred.class.smote>=0.5)
#*************************************************************************
************************
#Model Comparison for Ensemble Methods
modelcomparison=c("cm_bagging","cm_gb","cm_xgb","cm_smote","cm_adaboost")
modelcomparison
table_ensemble=data.frame(Sensitivity = NA,
              Specificity = NA,
              Precision = NA,
              Recall = NA,
              F1 = NA)
for (i in seq_along(modelcomparison)) {
  model=get(modelcomparison[i])
  a=data.frame(Sensitivity = model$byClass["Sensitivity"],
            Specificity = model$byClass["Specificity"],
            Precision = model$byClass["Precision"],
            Recall = model$byClass["Recall"],
            F1 = model$byClass["F1"])
  rownames(a)=NULL
  table_ensemble=rbind(table_ensemble,a)
}
table_ensemble=table_ensemble[-1,]
row.names(table_ensemble)=c("BAGGING","GBM","XGB","SMOTE","ADABOOST")
table_ensemble
#*************************************************************************
************************
#Model Comparison for Model Performance Matrices
modelcomp=c("cm_knn","cm_NB")
modelcomp
table_modelcomp=data.frame(Sensitivity = NA,
                      Specificity = NA,
                      Precision = NA,
                      Recall = NA,
                      F1 = NA)
table_modelcomp
for (i in seq_along(modelcomp)) {
  model1=get(modelcomp[i])
  b=data.frame(Sensitivity = model1$byClass["Sensitivity"],
            Specificity = model1$byClass["Specificity"],
            Precision = model1$byClass["Precision"],
            Recall = model1$byClass["Recall"],
            F1 = model1$byClass["F1"])
  rownames(b)=NULL
  table_modelcomp=rbind(table_modelcomp,b)
}
table_modelcomp=table_modelcomp[-1,]
row.names(table_modelcomp)=c("KNN","NAIVE BAYES")
table_modelcomp
```

```
#**************************************************************************
**********************
```