Project 5

Cars Case Study - Predicting Mode of Transport to Commute

Tahmid Bari – McCombs School of Business (Great Learning)
PGP – Data Science & Business Analytics

Contents

Description

This project requires you to understand what mode of transport employees prefers to commute to their office. The dataset **"Cars-dataset"** includes employee information about their mode of transport as well as their personal and professional details like age, salary, work exp. We need to predict whether or not an employee will use Car as a mode of transport. Also, which variables are a significant predictor behind this decision.

Following is expected out of the candidate in this assessment.

EDA (15 Marks)

- Perform an EDA on the data - (7 marks)
- Illustrate the insights based on EDA (5 marks)
- What is the most challenging aspect of this problem? What method will you use to deal with this? Comment (3 marks)

Data Preparation (10 marks)

- Prepare the data for analysis

Modeling (30 Marks)

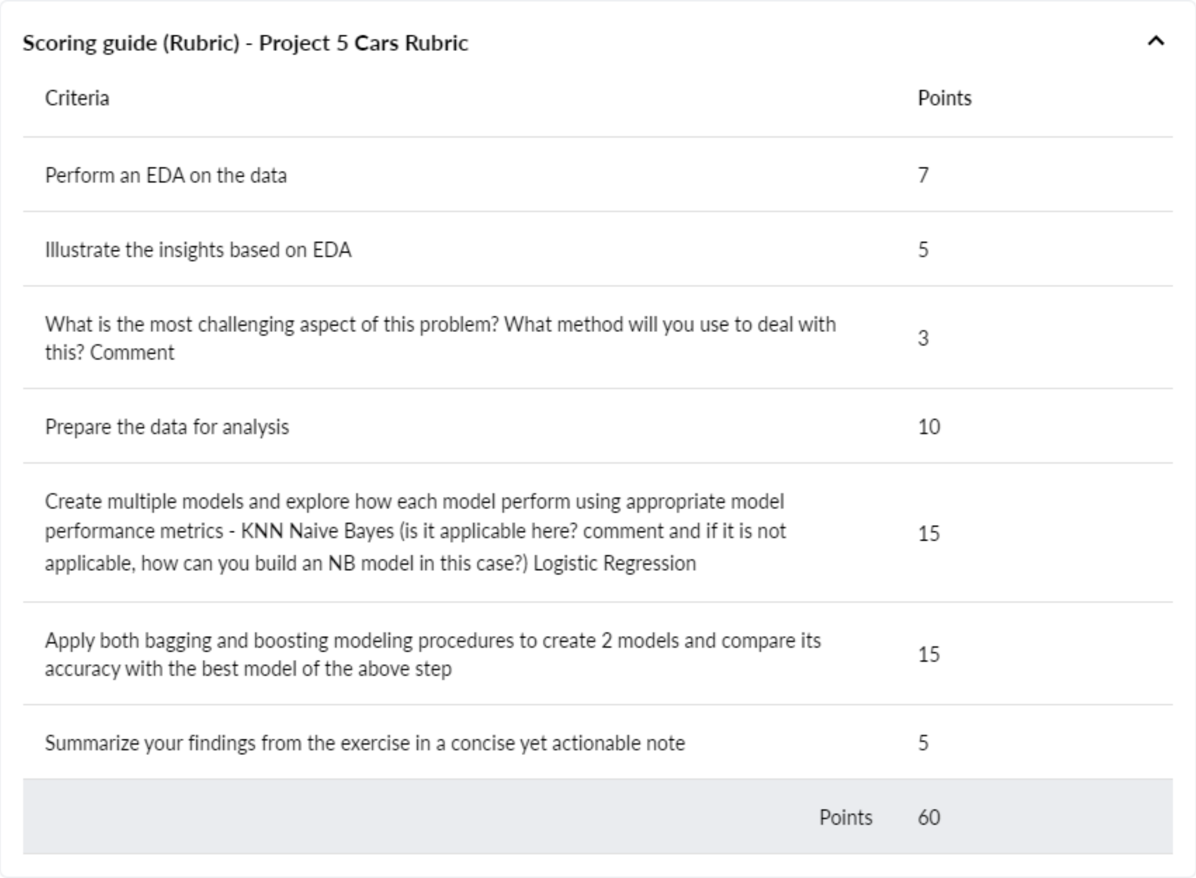
- Create multiple models and explore how each model perform using appropriate model performance metrics (15 marks)
    - KNN
    - Naive Bayes (is it applicable here? comment and if it is not applicable, how can you build an NB model in this case?)
    - Logistic Regression
- Apply both bagging and boosting modeling procedures to create 2 models and compare its accuracy with the best model of the above step. (15 marks)

Actionable Insights & Recommendations (5 Marks)

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

Please note the following:

1. There are two parts to the submission:
    1. The output/report in any file format - the key part of the output is the set of observations and insights from the exploration and analysis
    2. Commented R code in .R or .Rmd
2. Please don’t share your R code and/or outputs only, we expect some verbiage/story too - a meaningful output that you can share in a business environment
3. Any assignment found copied/ plagiarized with other groups will not be graded and awarded zero marks
4. Please ensure timely submission as post-deadline assignment will not be accepted



Scoring guide (Rubric) - Project 5 Cars Rubric

Criteria	Points
Perform an EDA on the data	7
Illustrate the insights based on EDA	5
What is the most challenging aspect of this problem? What method will you use to deal with this? Comment	3
Prepare the data for analysis	10
Create multiple models and explore how each model perform using appropriate model performance metrics - KNN Naive Bayes (is it applicable here? comment and if it is not applicable, how can you build an NB model in this case?) Logistic Regression	15
Apply both bagging and boosting modeling procedures to create 2 models and compare its accuracy with the best model of the above step	15
Summarize your findings from the exercise in a concise yet actionable note	5
Points	60

1. Project Objective

This project is to build a model which explains employees' preference to Car as mode of transport to commute to their office.

The data has employee information about their mode of transport as well as their personal and professional details like age, salary, work exp which would be analyzed to predict whether an employee will use Car as a mode of transport.

Also, identify variables which are a significant predictor behind this decision. We would be analyzing a data-set Cars.csv and performing techniques like logistic regression, KNN, Naïve Bayes and apply boosting and bagging modelling procedures to create 2 models then compare accuracy to come up with best model for our prediction.

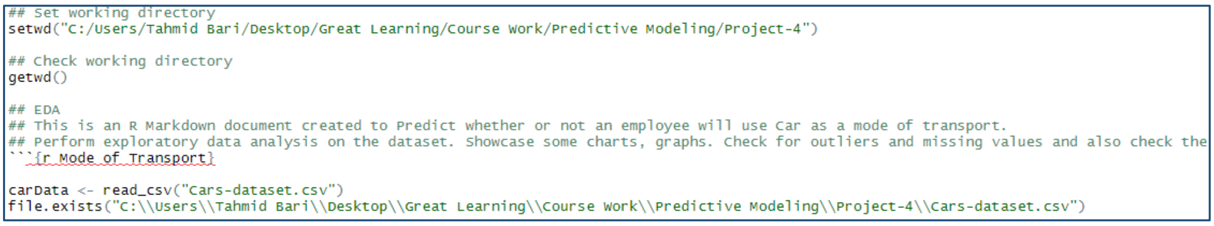
The Dataset looks like it's shown below:

Variable	**Description**
Age	Age of Employee
Gender	Gender of Employee (Male/ Female)
Engineer	1 states Employee is an Engineer, 0 states Employee has not performed Engineering related education
MBA	1 states Employee has an MBA, 0 states Employee has not performed MBA related education
Work Exp	Work Experience of Employee
Salary	Salary of Employee
Distance	Distance of Employee's home to office
License	1 state Employee possess license, 0 Employee do not
Transport	Mode of Transport generally used by the Employee to commute from home to office

2. Assumptions

There are a few assumptions considered:

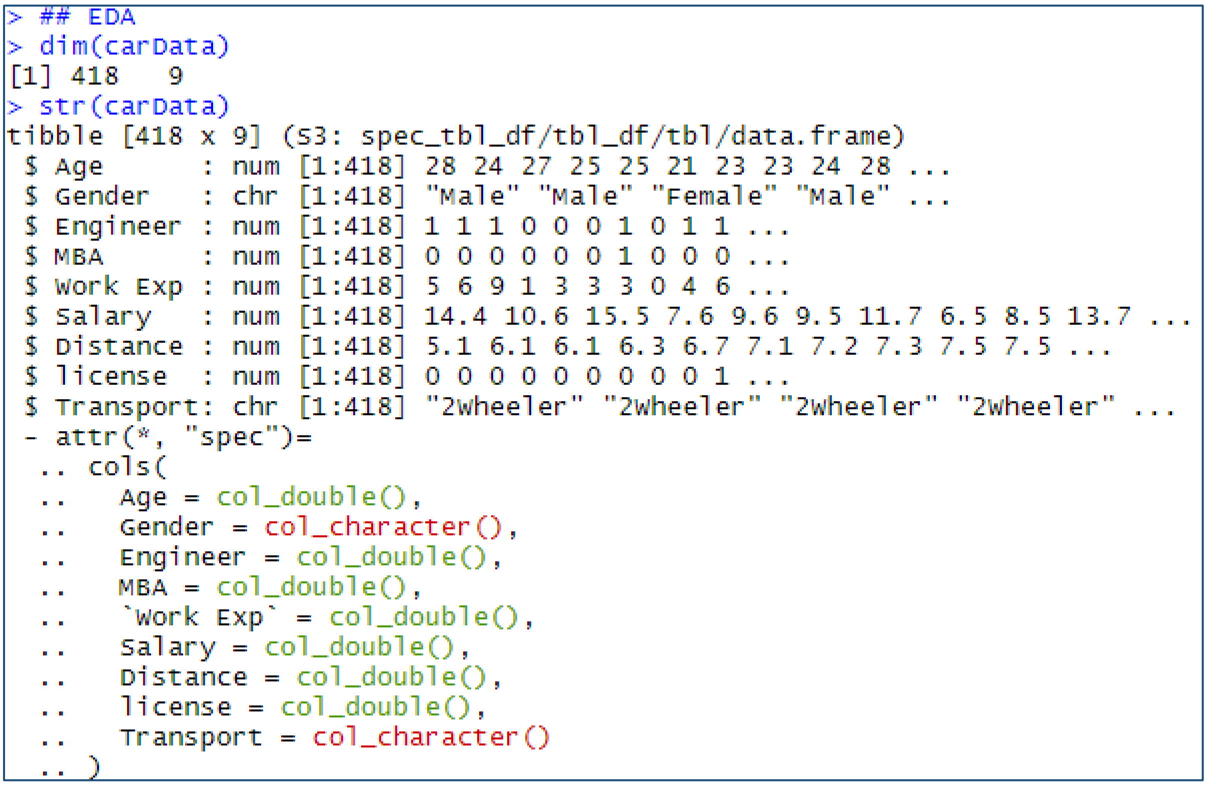
- The Sample size is adequate to perform techniques like logistic regression, KNN, Naïve Bayes.
- All the necessary packages are installed in R
- Working Directly is set to appropriate folder and file is in CSV format



~~~
## Set working directory
setwd("C:/Users/Tahmid Bari/Desktop/Great Learning/Course Work/Predictive Modeling/Project-4")

## Check working directory
getwd()

## EDA
## This is an R Markdown document created to Predict whether or not an employee will use Car as a mode of transport.
## Perform exploratory data analysis on the dataset. Showcase some charts, graphs. Check for outliers and missing values and also check the
```{r Mode of Transport}

carData <- read_csv("Cars-dataset.csv")
file.exists("C:\\Users\\Tahmid Bari\\Desktop\\Great Learning\\Course Work\\Predictive Modeling\\Project-4\\Cars-dataset.csv")
~~~

3. Exploratory Data Analysis

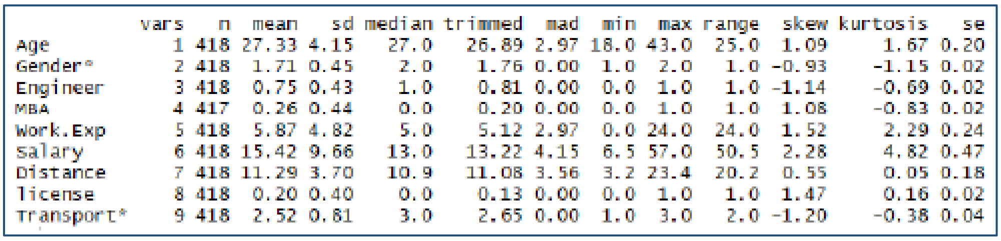
Structure of the data str(Transportation)and values analysis



```
> ## EDA
> dim(carData)
[1] 418   9
> str(carData)
tibble [418 x 9] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
 $ Age      : num [1:418] 28 24 27 25 25 21 23 23 24 28 ...
 $ Gender   : chr [1:418] "Male" "Male" "Female" "Male" ...
 $ Engineer : num [1:418] 1 1 1 0 0 0 1 0 1 1 ...
 $ MBA      : num [1:418] 0 0 0 0 0 0 1 0 0 0 ...
 $ Work Exp : num [1:418] 5 6 9 1 3 3 3 0 4 6 ...
 $ Salary   : num [1:418] 14.4 10.6 15.5 7.6 9.6 9.5 11.7 6.5 8.5 13.7 ...
 $ Distance : num [1:418] 5.1 6.1 6.1 6.3 6.7 7.1 7.2 7.3 7.5 7.5 ...
 $ license  : num [1:418] 0 0 0 0 0 0 0 0 0 1 ...
 $ Transport: chr [1:418] "2Wheeler" "2Wheeler" "2Wheeler" "2Wheeler" ...
 - 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()
  .. )
```

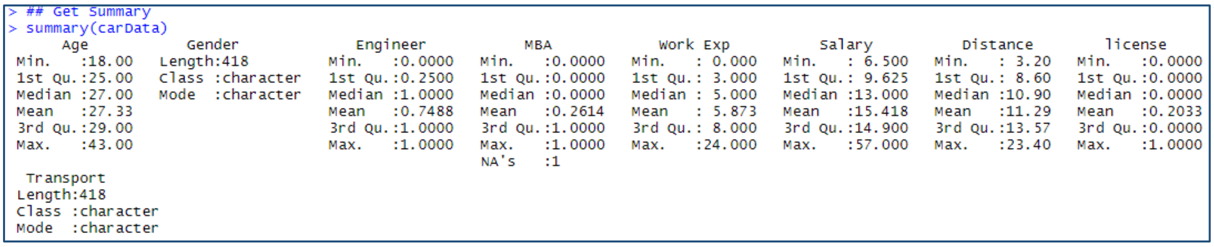
3.1 Number of Rows and Columns

- The number of rows in the dataset is 418
- The number of columns (Features) in the dataset is 9
- Target variable is “Transport”
- Description of data set

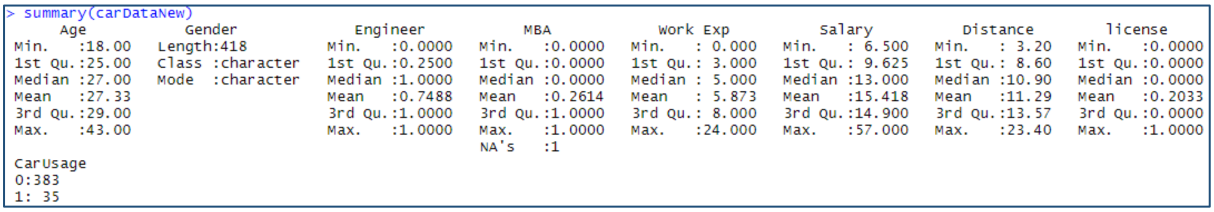


```
           vars   n  mean   sd median trimmed  mad  min  max range  skew kurtosis   se
Age           1 418 27.33 4.15   27.0   26.89 2.97 18.0 43.0  25.0  1.09     1.67 0.20
Gender*       2 418  1.71 0.45    2.0    1.76 0.00  1.0  2.0   1.0 -0.93    -1.15 0.02
Engineer      3 418  0.75 0.43    1.0    0.81 0.00  0.0  1.0   1.0 -1.14    -0.69 0.02
MBA           4 417  0.26 0.44    0.0    0.20 0.00  0.0  1.0   1.0  1.08    -0.83 0.02
Work.Exp      5 418  5.87 4.82    5.0    5.12 2.97  0.0 24.0  24.0  1.52     2.29 0.24
Salary        6 418 15.42 9.66   13.0   13.22 4.15  6.5 57.0  50.5  2.28     4.82 0.47
Distance      7 418 11.29 3.70   10.9   11.08 3.56  3.2 23.4  20.2  0.55     0.05 0.18
license       8 418  0.20 0.40    0.0    0.13 0.00  0.0  1.0   1.0  1.47     0.16 0.02
Transport*    9 418  2.52 0.81    3.0    2.65 0.00  1.0  3.0   2.0 -1.20    -0.38 0.04
```

3.1.1 Dataset Summary



```
> ## Get Summary
> summary(carData)
      Age           Gender             Engineer           MBA            Work Exp          Salary          Distance        license      
 Min.   :18.00   Length:418         Min.   :0.0000   Min.   :0.0000   Min.   : 0.000   Min.   : 6.500   Min.   : 3.20   Min.   :0.0000  
 1st Qu.:25.00   Class :character   1st Qu.:0.2500   1st Qu.:0.0000   1st Qu.: 3.000   1st Qu.: 9.625   1st Qu.: 8.60   1st Qu.:0.0000  
 Median :27.00   Mode  :character   Median :1.0000   Median :0.0000   Median : 5.000   Median :13.000   Median :10.90   Median :0.0000  
 Mean   :27.33                      Mean   :0.7488   Mean   :0.2614   Mean   : 5.873   Mean   :15.418   Mean   :11.29   Mean   :0.2033  
 3rd Qu.:29.00                      3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.: 8.000   3rd Qu.:14.900   3rd Qu.:13.57   3rd Qu.:0.0000  
 Max.   :43.00                      Max.   :1.0000   Max.   :1.0000   Max.   :24.000   Max.   :57.000   Max.   :23.40   Max.   :1.0000  
                                                     NA's   :1                                                                           
  Transport        
 Length:418        
 Class :character  
 Mode  :character  
```



```
> summary(carDataNew)
      Age            Gender             Engineer           MBA             Work Exp          Salary          Distance         license      
 Min.   :18.00   Length:418         Min.   :0.0000   Min.   :0.0000   Min.   : 0.000   Min.   : 6.500   Min.   : 3.20   Min.   :0.0000  
 1st Qu.:25.00   Class :character   1st Qu.:0.2500   1st Qu.:0.0000   1st Qu.: 3.000   1st Qu.: 9.625   1st Qu.: 8.60   1st Qu.:0.0000  
 Median :27.00   Mode  :character   Median :1.0000   Median :0.0000   Median : 5.000   Median :13.000   Median :10.90   Median :0.0000  
 Mean   :27.33                      Mean   :0.7488   Mean   :0.2614   Mean   : 5.873   Mean   :15.418   Mean   :11.29   Mean   :0.2033  
 3rd Qu.:29.00                      3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.: 8.000   3rd Qu.:14.900   3rd Qu.:13.57   3rd Qu.:0.0000  
 Max.   :43.00                      Max.   :1.0000   Max.   :1.0000   Max.   :24.000   Max.   :57.000   Max.   :23.40   Max.   :1.0000  
                                                     NA's   :1                                                                            
 CarUsage
 0:383   
 1: 35   
```

3.1.2 Preliminary Data Analysis

Cars data is carrying the Data Frame of the CSV file. Below are analysis points which should be considered for data correction.

- Transport is a target variable as it predicts whether an employee use car or other transport to reach office.
- “Car” Transport data is comparatively a very small percentage i.e. 8.4% so data treatment should be under imbalance small data principles.
- In order to predict whether employee will use Car as mode of transports, target variable data can be converted to more simpler form i.e. Car or other mode of transport (1 or 0).
- Gender consist of “Male” and “Female” so for better analysis converting it into simpler form like Male or not would be good approach .(1 /0)
- Null value or missing data exists in MBA column so it need to be treated .
- Data wrangling can be done for couple of fields so to allow for more convenient consumption and organization of the data..
- All the Flag fields need to be categorical so need to be converted from int to Factor or num when needed .
- Outliers noticed in analysis done on summary so far. It needs to be treated.

3.1.3 Zero variance/Near Zero variance check

Zero variance/ Near Zero variance check performed on data and data is good to continue for further operation.

3.1.4 Transport as Car Rate

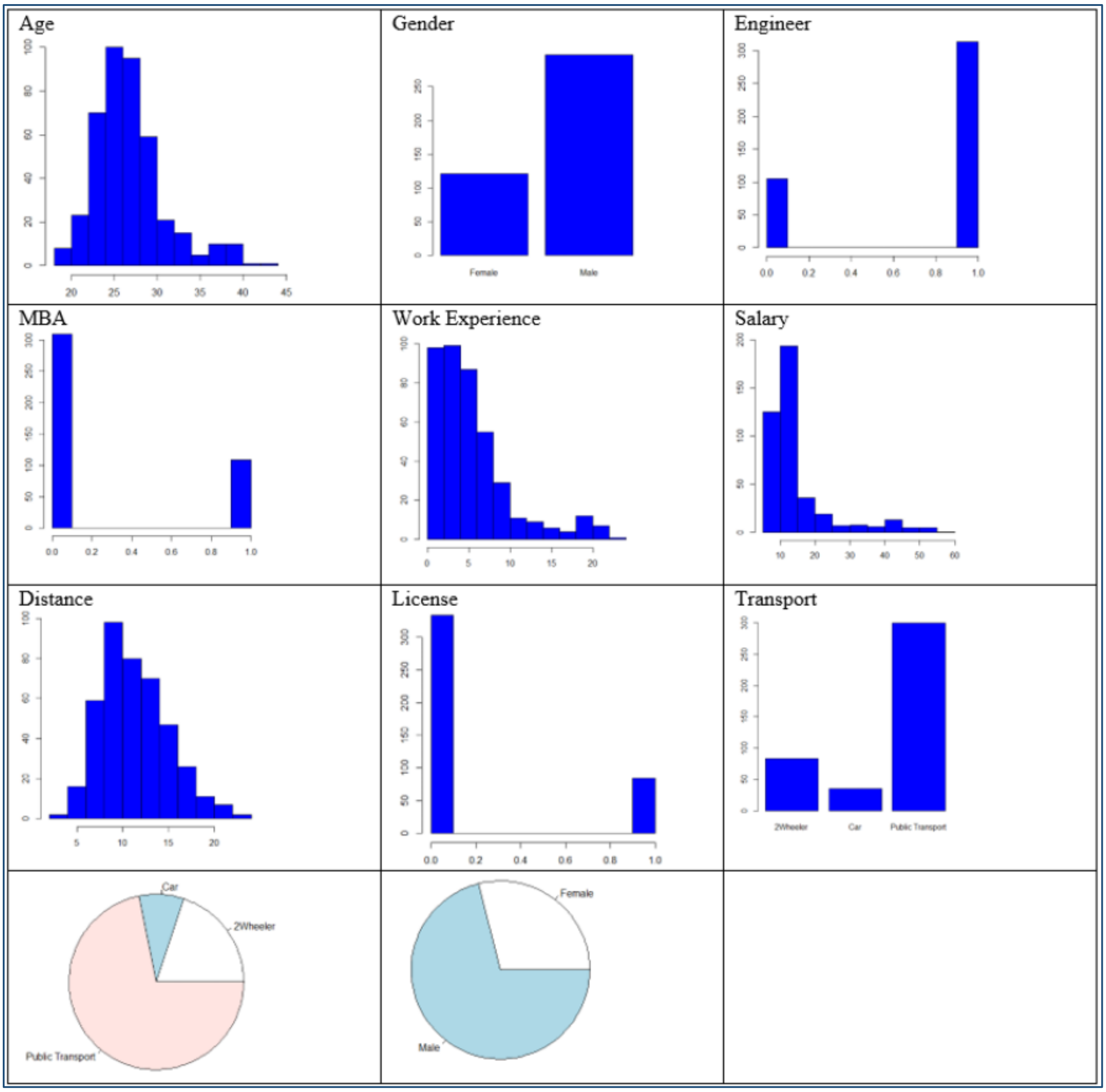
We calculated (carData) rate before splitting of data. 8.4% of Employees use Car as their transport to commute to office. So, let’s focus on this group to build various models to predict better.

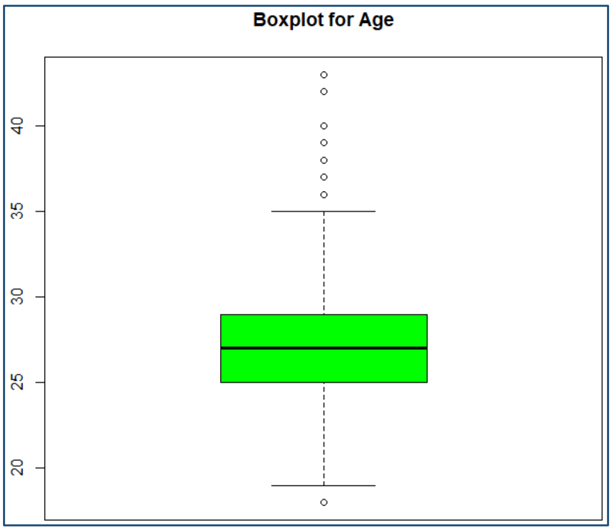
3.2 Data Visualization of the Variables (Plots and Charts)

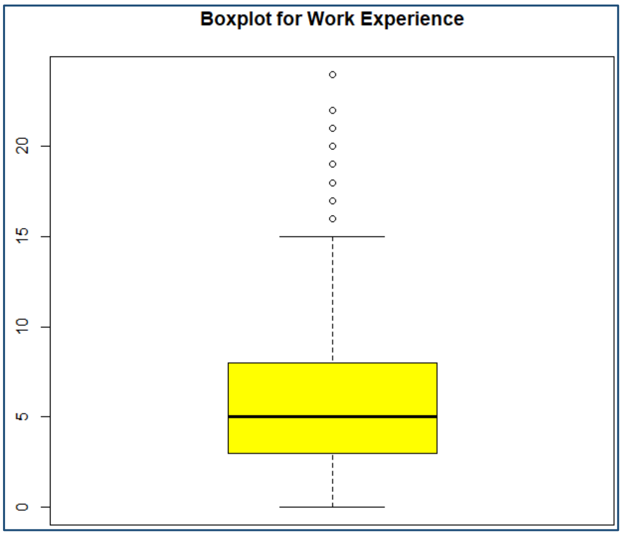
3.2.1 Univariate Analysis

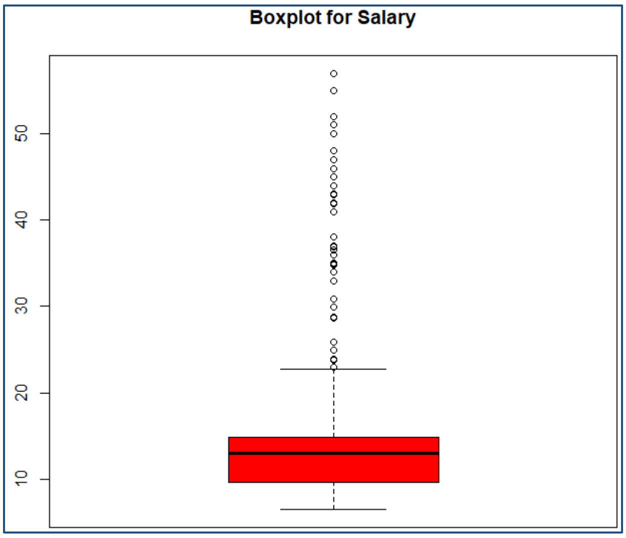
Data Analysis is done on each feature for better understanding:

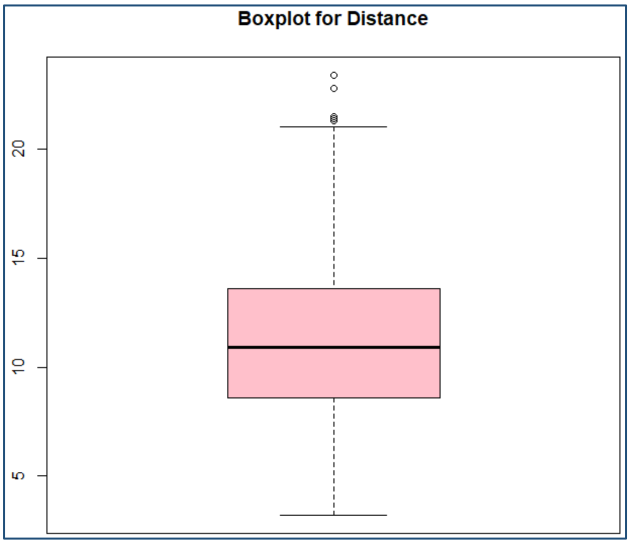
- Frequencies for categorical variable using table function. Also used Pie chart to get quick visual of categorical variable.
- Frequencies for continuous variable using Histogram function.

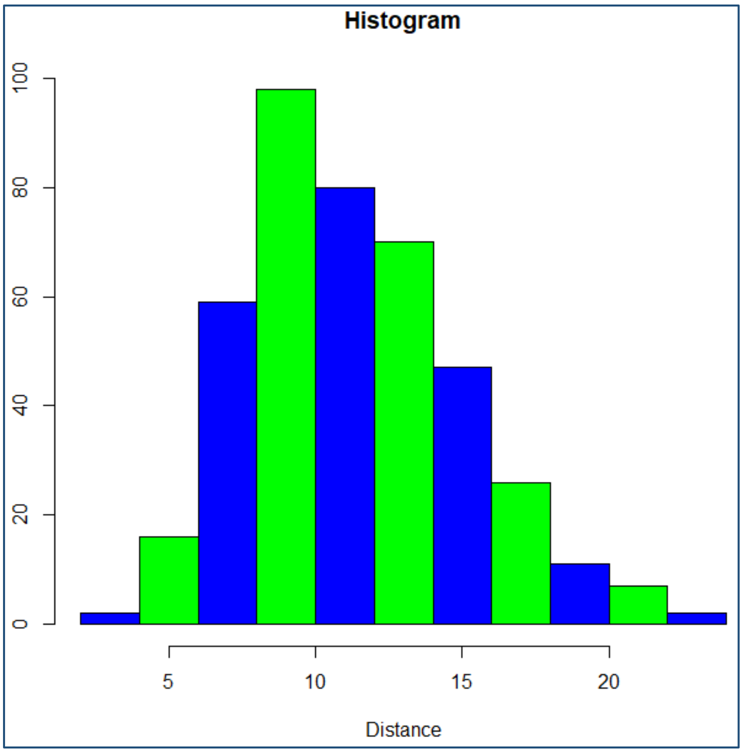

Age
0
20
40
60
80
100
20
25
30
35
40
45
Gender
0
50
100
150
200
250
Female
Male
Engineer
0
50
100
150
200
250
300
0.0
0.2
0.4
0.6
0.8
1.0
MBA
0
50
100
150
200
250
300
0.0
0.2
0.4
0.6
0.8
1.0
Work Experience
0
20
40
60
80
100
0
5
10
15
20
Salary
0
50
100
150
200
10
20
30
40
50
60
Distance
0
20
40
60
80
100
5
10
15
20
License
0
50
100
150
200
250
300
0.0
0.2
0.4
0.6
0.8
1.0
Transport
0
50
100
150
200
250
300
2Wheeler
Car
Public Transport
Car
2Wheeler
Public Transport
Female
Male

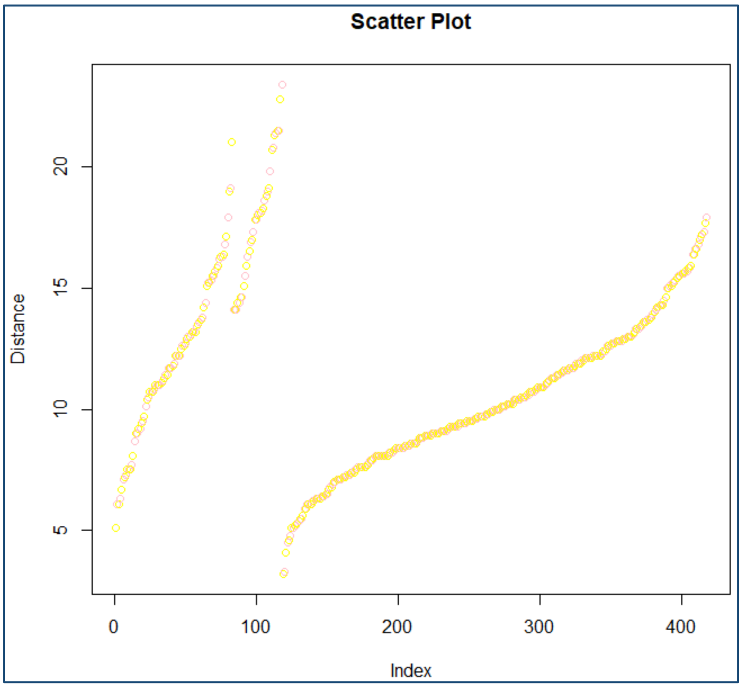

Boxplot for Age
20
25
30
35
40

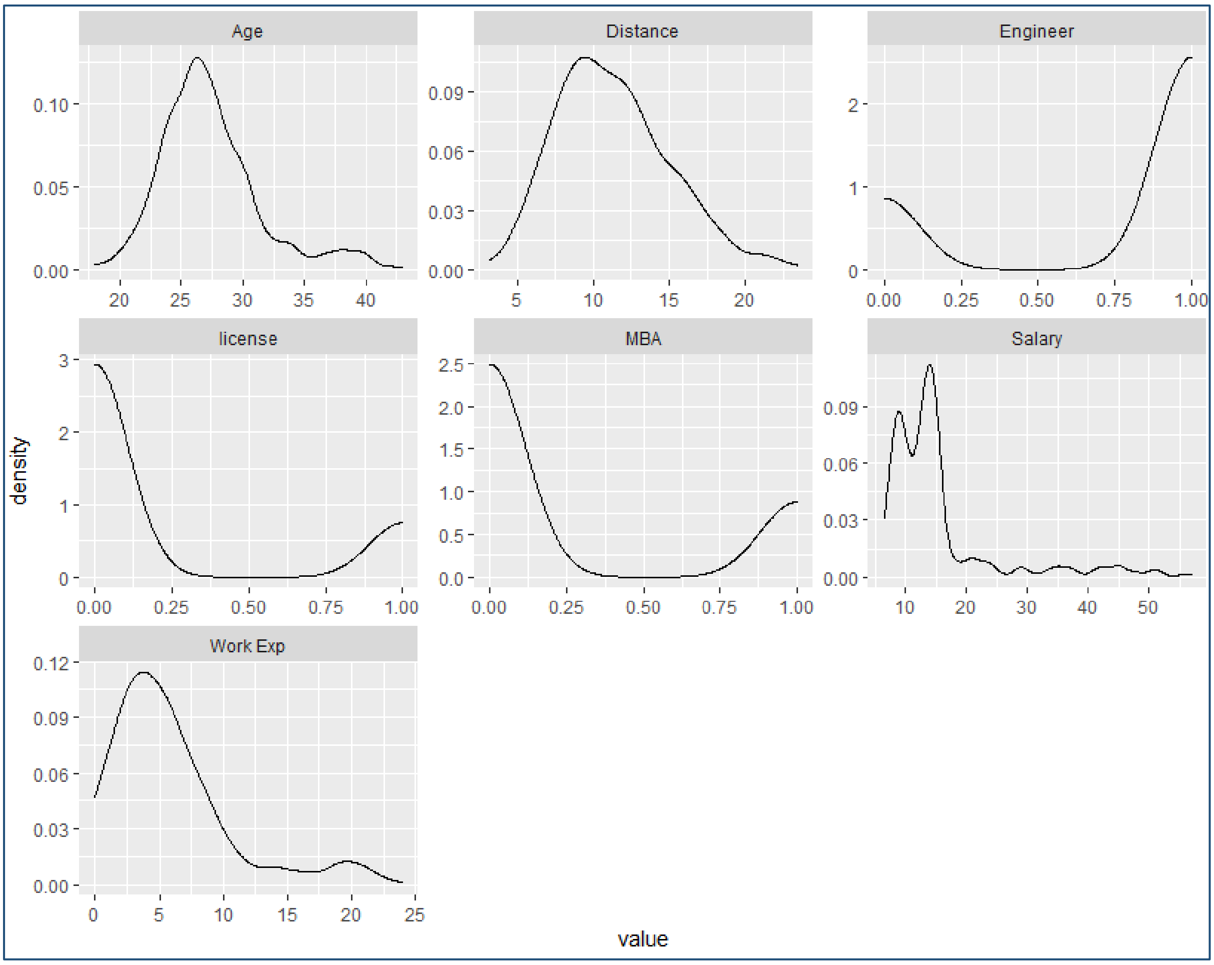

Boxplot for Work Experience
0
5
10
15
20


Boxplot for Salary
10
20
30
40
50


Boxplot for Distance
5
10
15
20

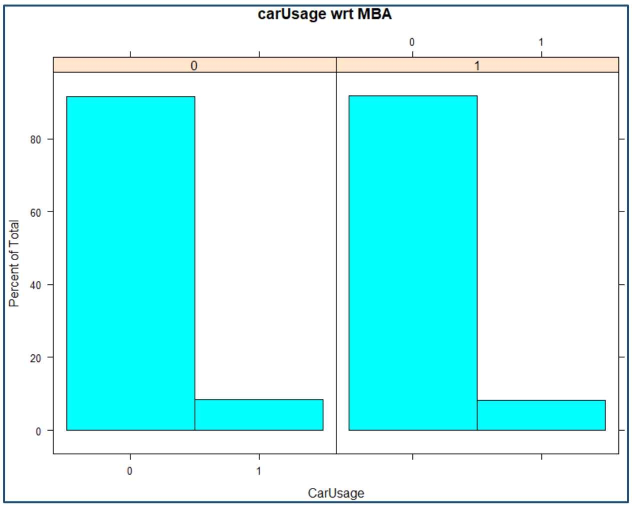

Histogram
0
20
40
60
80
100
5
10
15
20
Distance

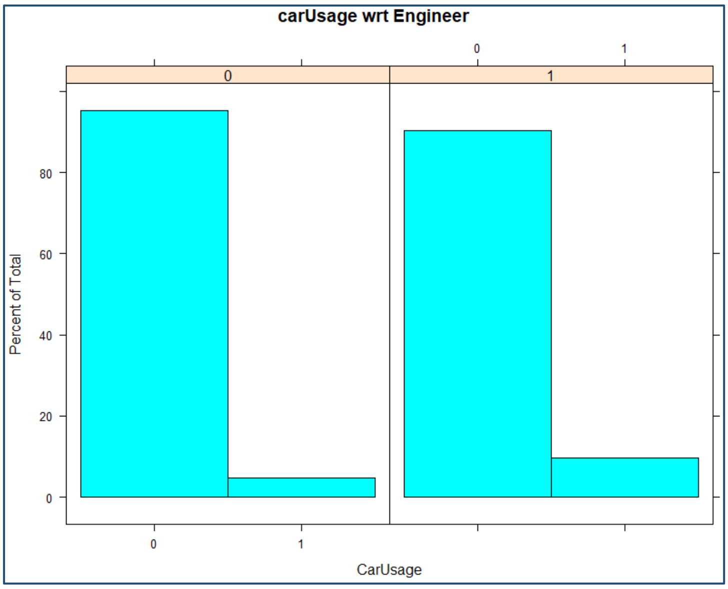

Scatter Plot
Distance
5
10
15
20
0
100
200
300
400
Index

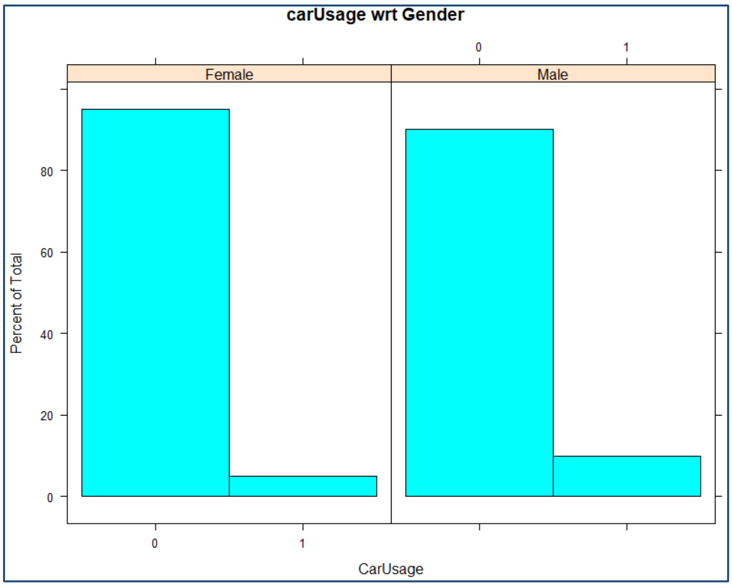


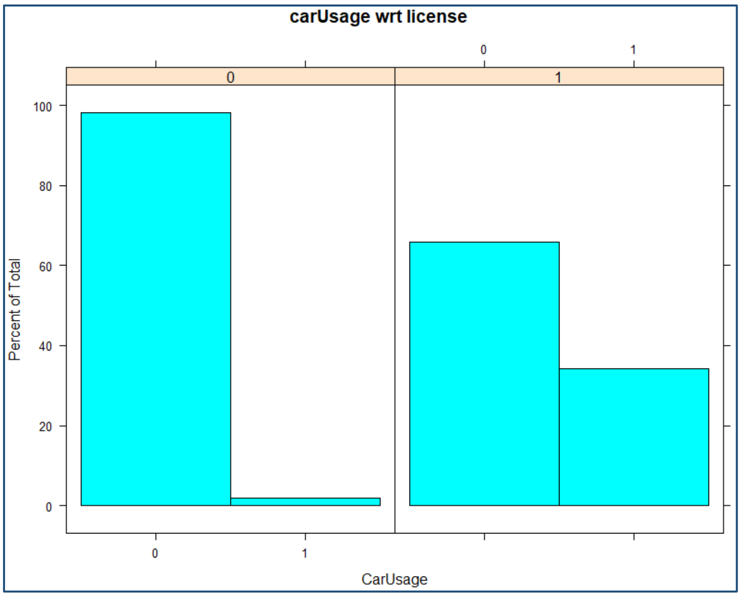
3.2.2 Bivariate Analysis

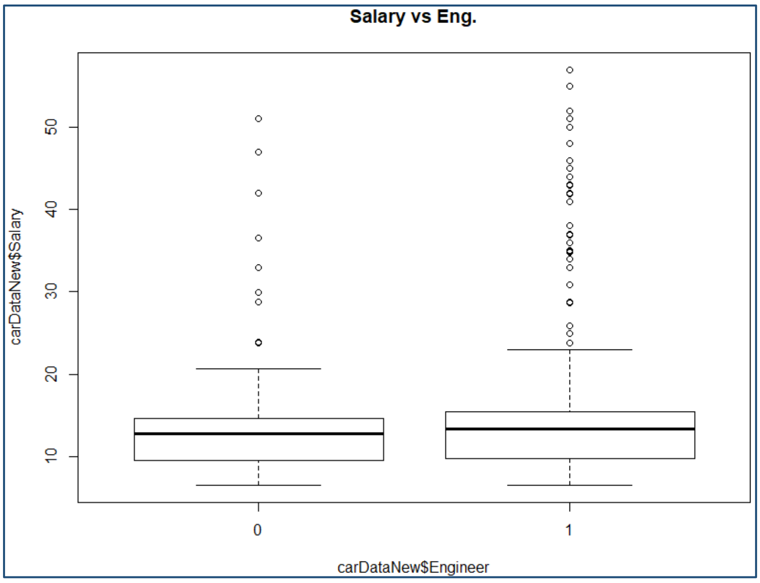
Below shows a scatter plot of matrices, with bivariate scatter plots below the diagonal, histograms on the diagonal, and the Pearson correlation (r)above the diagonal.

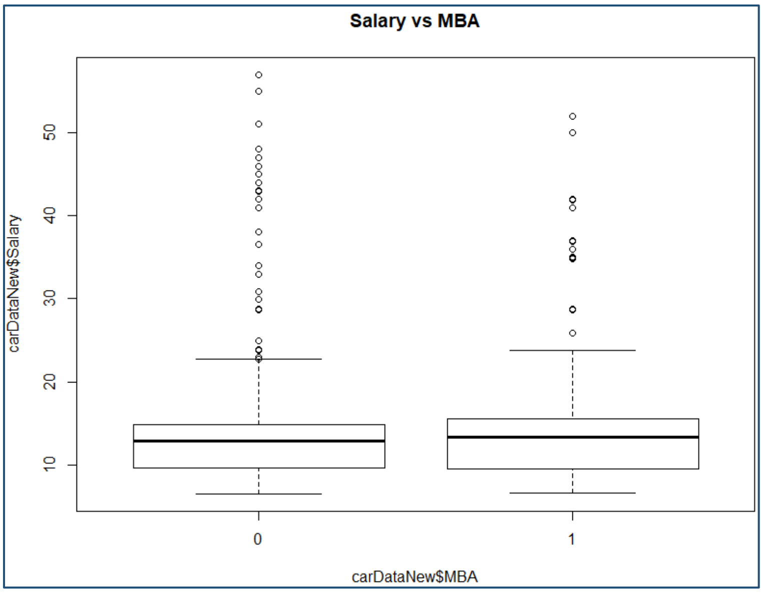


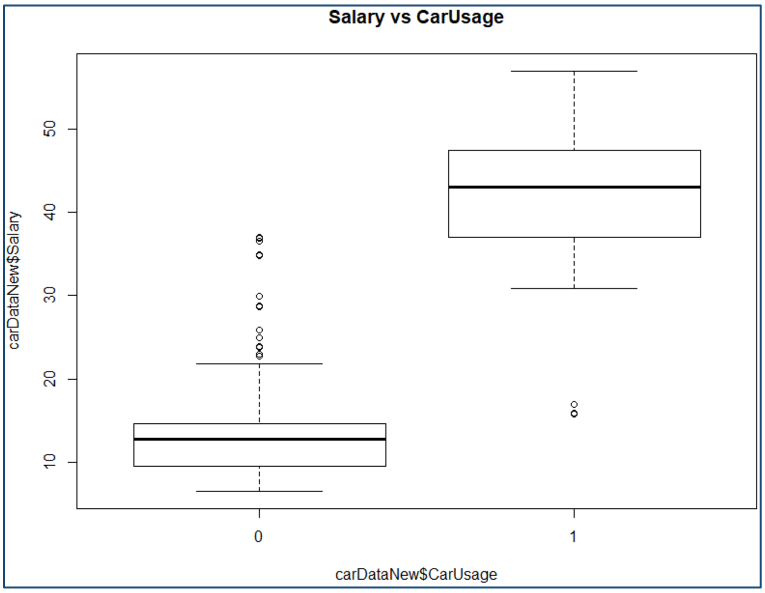

carUsage wrt Engineer
0
1
Percent of Total
0
20
40
60
80
0
1
0
1
CarUsage

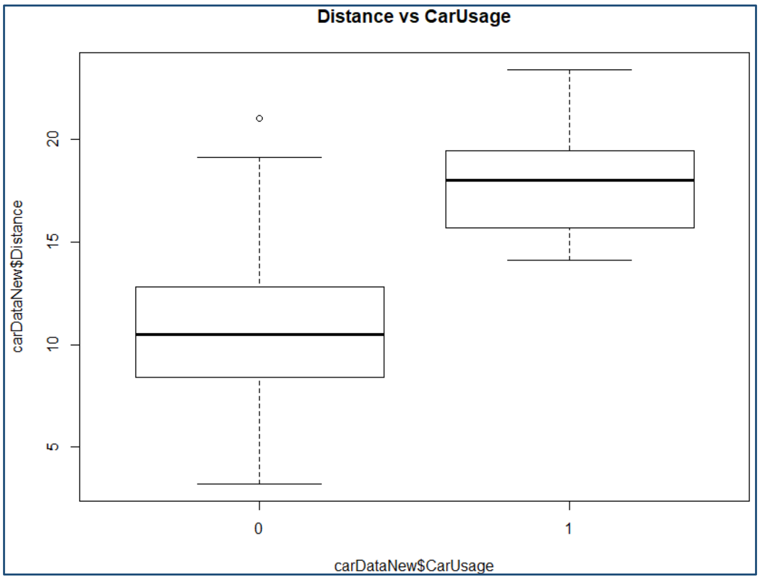

carUsage wrt Gender
Female
Male
Percent of Total
0
20
40
60
80
0
1
0
1
CarUsage

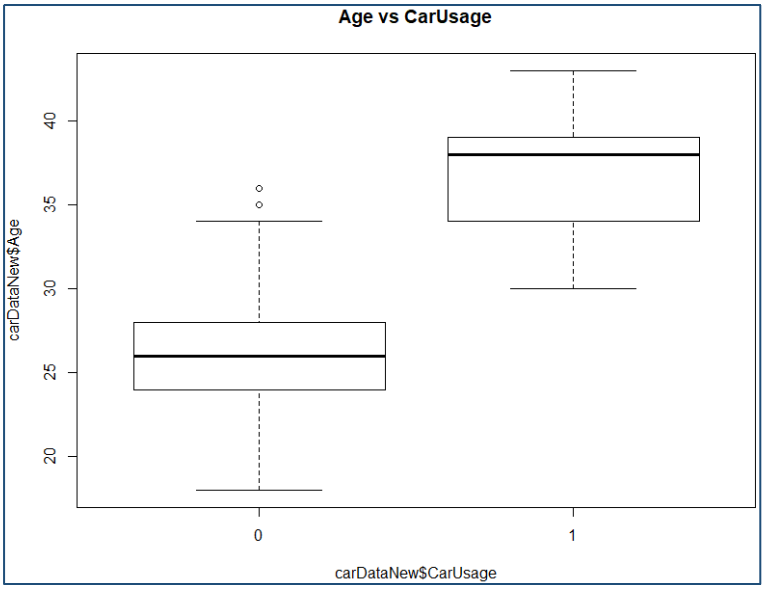

carUsage wrt license
0
1
Percent of Total
0
20
40
60
80
100
0
1
0
1
CarUsage


Salary vs Eng.


Salary vs MBA


Salary vs CarUsage





Relationship between CarUsage and Salary, Distance, Age, Gender, Engineer, MBA and license.
The 0 indicates that the mode of transport either public or two-wheeler while the 1 indicates that the mode of transport is a car. We notice a higher number of people who do not use car at all in every categorical variable.

Majority of the people actually are using car are Males or have an Engineering degree and most of them have li-cense. The number of women who drive a car are very negligible.

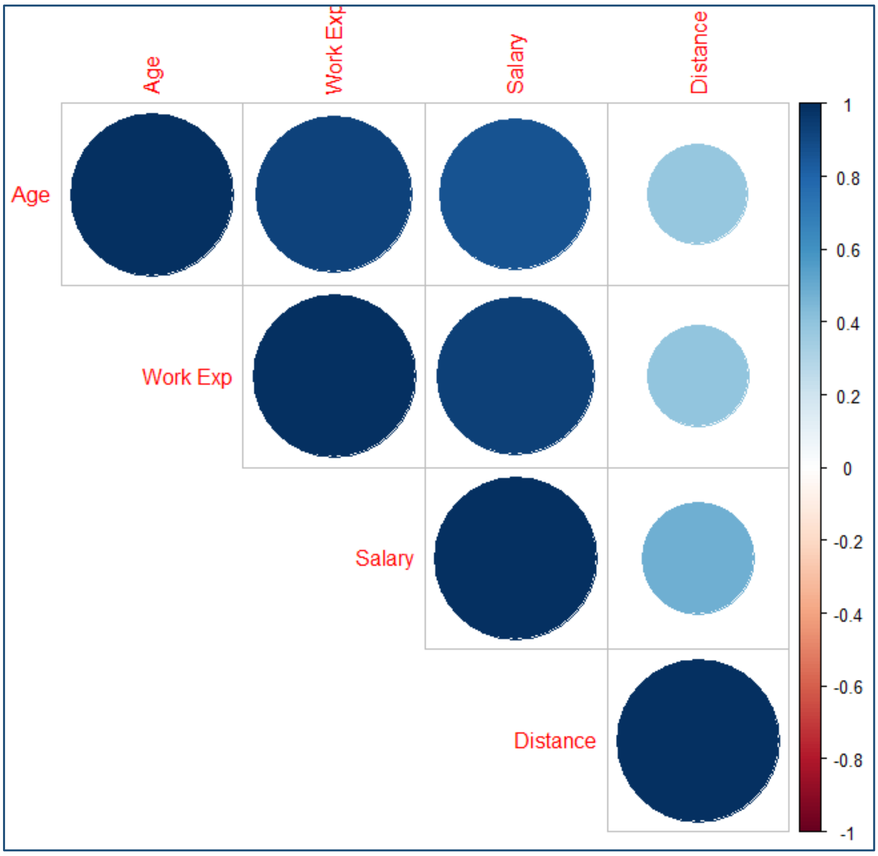
Also, we notice that the number of people who don't have an MBA degree drive car more than the people with an MBA degree.

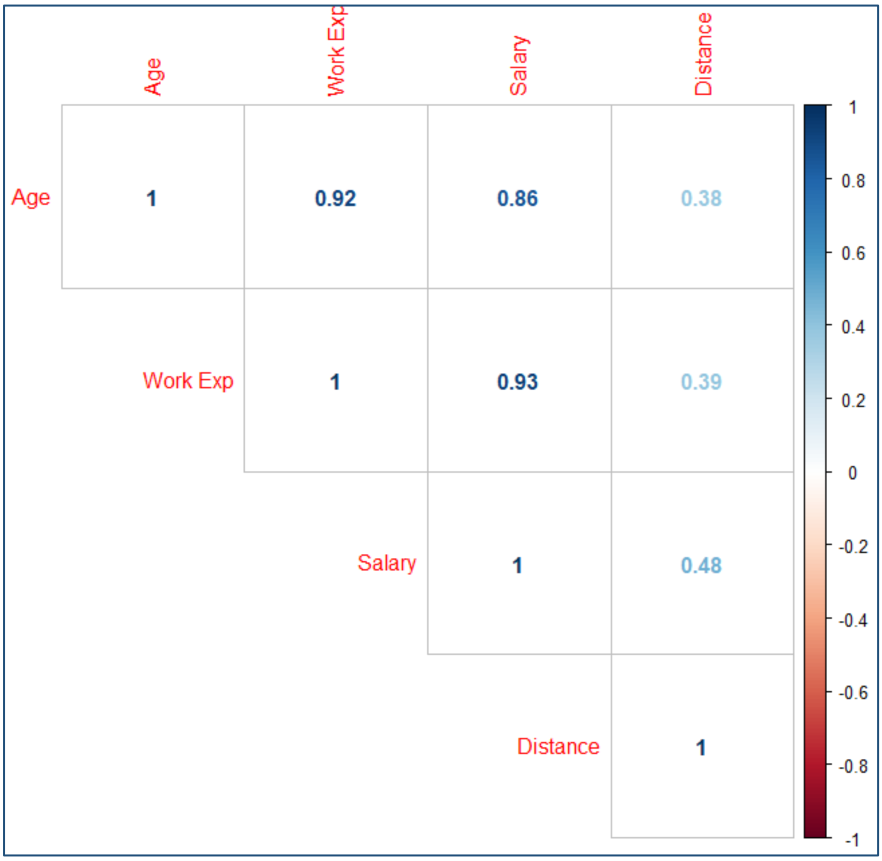
This is slightly surprising that some of the people who drive car do not have valid license. Every person who com-mutes via car should have a valid license (if they are driving).

We notice outliers in nearly every variable for especially when the transport involves a car for commute.

Correlation:

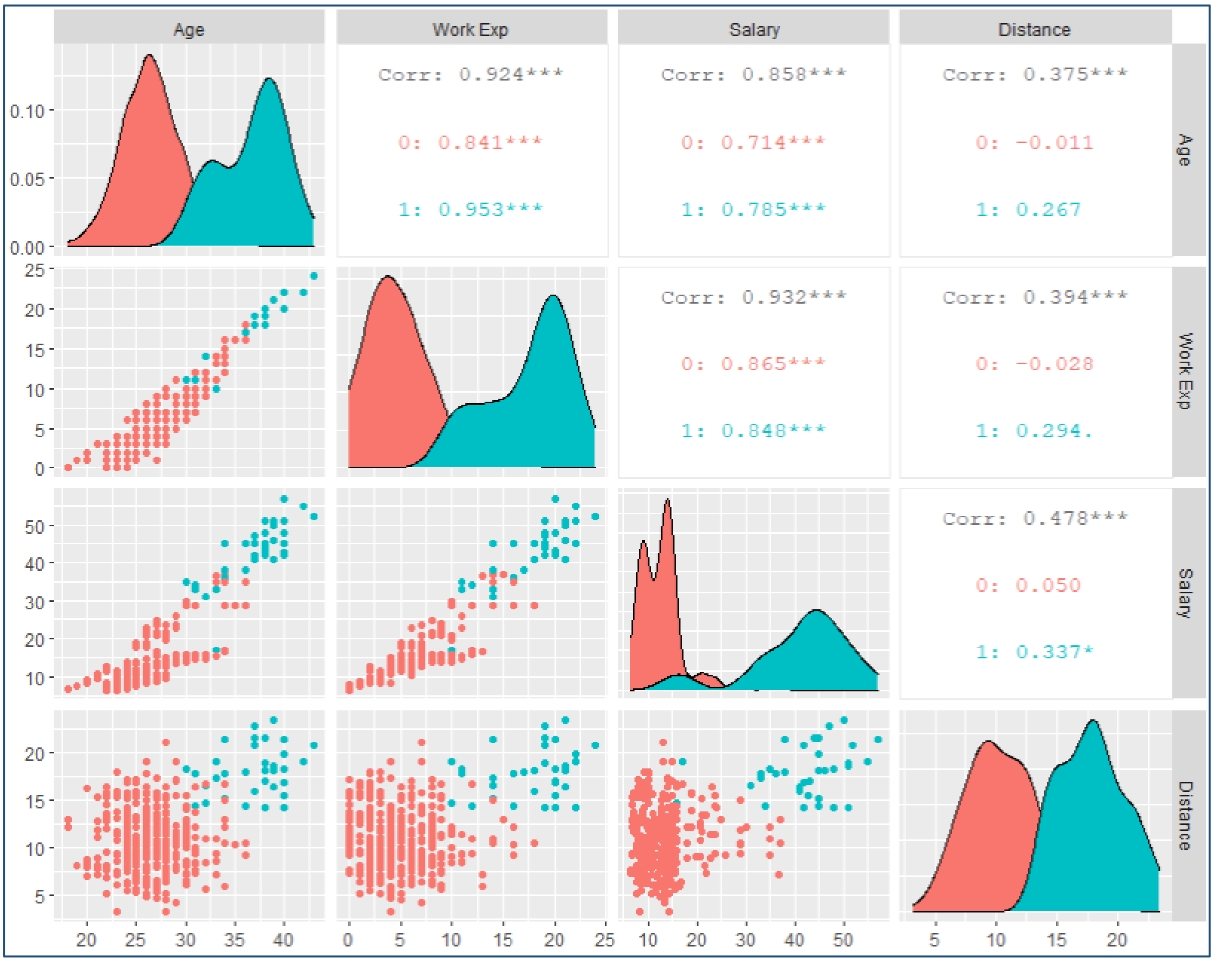
Plotting the correlation of every numeric variable [after skipping the factor variables]:





We notice a high correlation between [Work Experience & Age] and [Salary & Work Experience] and [Age & Salary].

This is because there is a direct relation between age and work experience and more the age, the more the person will have gained experience. Also, As the age increased, the work experience increased and this results in increase of Salary. Hence these three columns are related to one another. This will result in Multicollinearity and we may have to remove one or two variables or perform PCA/ PFA to counter the changes.



We notice a normal distribution for most of the numeric variables for the CarUsage of both 1 and 0 but not along the same curves. That is, the behavioural patterns of the people who use the car and who don't use a car are very distinct and different.

We notice that higher the age, the more likelihood they had of driving car to work. The same with respect to other numeric columns like Work Experience, Salary or Distance.

Young employees of the age group < 27 yrs did not drive a car at all and we can easily categorize the data into two different segments according to the age.

Also, people who had to travel more than 10km were more likely to drive a car rather than use public transport or 2wheeler. People who travelled less than 10km hardly ever used a car.

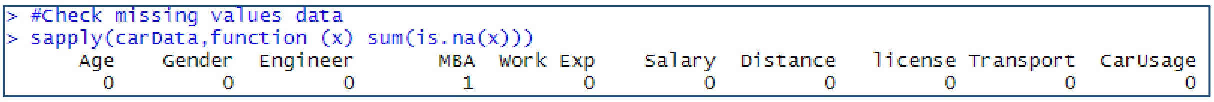
We also notice that young employees with less than 5yrs of experience did not drive a car at all while employees with around 15 yrs of experience seem to use car as their main mode of transportation.

We see different peaks with respect to the Salary and car usage. Some people with lesser salaries preferred cars while majority of the people who used car had Salary around 40 units. Also, majority of the people whose salaries are around 15 units preferred public transport or 2wheeler as their main mode of commute.

3.3 Outliers and Missing data in data set

3.3.1 Missing Value/s

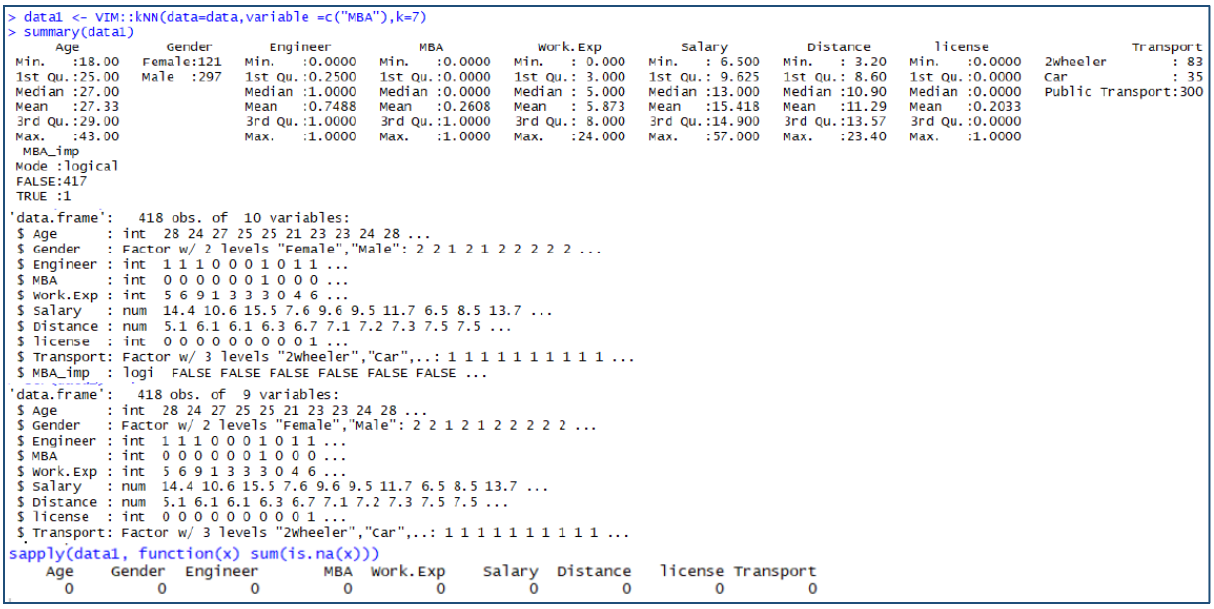
The data checked for Missing Values using R function sum(is.na(tget.data)). We use sapply to check the number if missing values in each column. There was missing value in MBA data column so it needs to be treated.



```
> #Check missing values data
> sapply(carData,function (x) sum(is.na(x)))
      Age    Gender  Engineer         MBA  Work Exp    Salary  Distance   license Transport  CarUsage
        0         0         0           1         0         0         0         0         0         0
```

3.3.1.1 Null Value Treatment

KNN is an algorithm that is useful for matching a point with its closest k neighbors in a multi-dimensional space. It can be used for data that are continuous, discrete, ordinal and categorical which makes it particularly useful for dealing with all kind of missing data. The assumption behind using KNN for missing values is that a point value can be approximated by the values of the points that are closest to it, based on other variables. It is seen here that MBA_imp new logical column has been created and it has one value set as TRUE. This means one null value has been imputed.



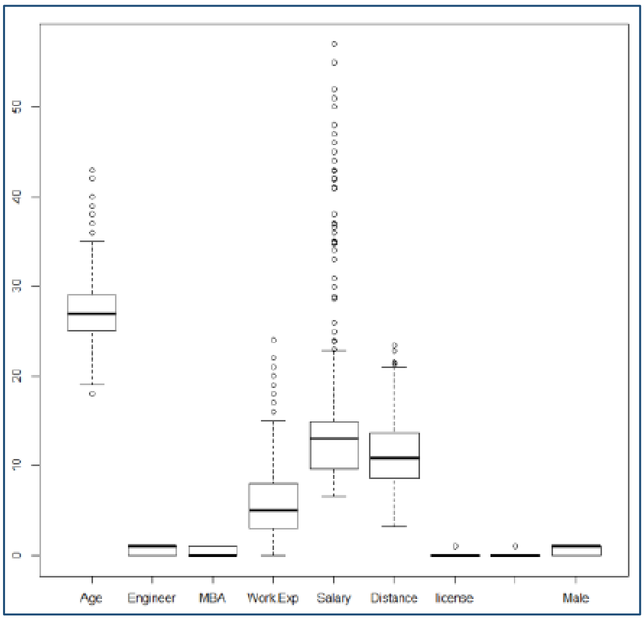
```
> data1 <- VIM::kNN(data=data,variable =c("MBA"),k=7)
> summary(data1)
      Age            Gender        Engineer           MBA            Work.Exp          Salary          Distance         license                  Transport  
 Min.   :18.00   Female:121   Min.   :0.0000   Min.   :0.0000   Min.   : 0.000   Min.   : 6.500   Min.   : 3.20   Min.   :0.0000   2Wheeler        : 83  
 1st Qu.:25.00   Male  :297   1st Qu.:0.2500   1st Qu.:0.0000   1st Qu.: 3.000   1st Qu.: 9.625   1st Qu.: 8.60   1st Qu.:0.0000   Car             : 35  
 Median :27.00                Median :1.0000   Median :0.0000   Median : 5.000   Median :13.000   Median :10.90   Median :0.0000   Public Transport:300  
 Mean   :27.33                Mean   :0.7488   Mean   :0.2608   Mean   : 5.873   Mean   :15.418   Mean   :11.29   Mean   :0.2033                         
 3rd Qu.:29.00                3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.: 8.000   3rd Qu.:14.900   3rd Qu.:13.57   3rd Qu.:0.0000                         
 Max.   :43.00                Max.   :1.0000   Max.   :1.0000   Max.   :24.000   Max.   :57.000   Max.   :23.40   Max.   :1.0000                         
  MBA_imp       
 Mode :logical  
 FALSE:417      
 TRUE :1        
'data.frame':	418 obs. of  10 variables:
 $ Age      : int  28 24 27 25 25 21 23 23 24 28 ...
 $ Gender   : Factor w/ 2 levels "Female","Male": 2 2 1 2 1 2 2 2 2 2 ...
 $ Engineer : int  1 1 1 0 0 0 1 0 1 1 ...
 $ MBA      : int  0 0 0 0 0 0 1 0 0 0 ...
 $ Work.Exp : int  5 6 9 1 3 3 3 0 4 6 ...
 $ Salary   : num  14.4 10.6 15.5 7.6 9.6 9.5 11.7 6.5 8.5 13.7 ...
 $ Distance : num  5.1 6.1 6.1 6.3 6.7 7.1 7.2 7.3 7.5 7.5 ...
 $ license  : int  0 0 0 0 0 0 0 0 0 1 ...
 $ Transport: Factor w/ 3 levels "2Wheeler","Car",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ MBA_imp  : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
'data.frame':	418 obs. of  9 variables:
 $ Age      : int  28 24 27 25 25 21 23 23 24 28 ...
 $ Gender   : Factor w/ 2 levels "Female","Male": 2 2 1 2 1 2 2 2 2 2 ...
 $ Engineer : int  1 1 1 0 0 0 1 0 1 1 ...
 $ MBA      : int  0 0 0 0 0 0 1 0 0 0 ...
 $ Work.Exp : int  5 6 9 1 3 3 3 0 4 6 ...
 $ Salary   : num  14.4 10.6 15.5 7.6 9.6 9.5 11.7 6.5 8.5 13.7 ...
 $ Distance : num  5.1 6.1 6.1 6.3 6.7 7.1 7.2 7.3 7.5 7.5 ...
 $ license  : int  0 0 0 0 0 0 0 0 0 1 ...
 $ Transport: Factor w/ 3 levels "2Wheeler","Car",..: 1 1 1 1 1 1 1 1 1 1 ...
sapply(data1, function(x) sum(is.na(x)))
      Age    Gender  Engineer       MBA  Work.Exp    Salary  Distance   license Transport 
        0         0         0         0         0         0         0         0         0 
```

Null value is missing now. Original data frame had 418 observations and new data frame also has same number of record.

3.3.2 Checking for Outlier

For all Features , plotted a Box plot to check for Outliers , below are the observation and analysis points.

- Age : Outlier : Exist at both end but majorly at right side .Data is evenly skewed which is reflected in plot and graph both.
- Salary : Outlier : Exist at right end .Data is slightly left skewed which is reflected in plot and graph both.Grouping of data can be done based on Histogram.
- Work Experience : Outlier : Exist at right end .Data is slightly right skewed which is reflected in plot and graph both.Grouping of data can be done based on Histogram
- Distance and License : Few Outlier exist at right end



3.3.2.1 Outlier Treatment

There are couple of methods to remove outliers:

1. Identifying Outlier values by Boxplot command. It can be used to get the actual values of the outliers.

    a. boxplot(data$Age)$out

    b. Or boxplot(data$Age,plot =FALSE)$out

2. Assign outlier values into vector

3. Check which rows has outliers : data[which(data$Age %in% outliers),]

4. Option 1-

    a. Removal of outliers is to remove the row .:data <- data [-which(data$Age %in% outliers),]

5. Option 2-

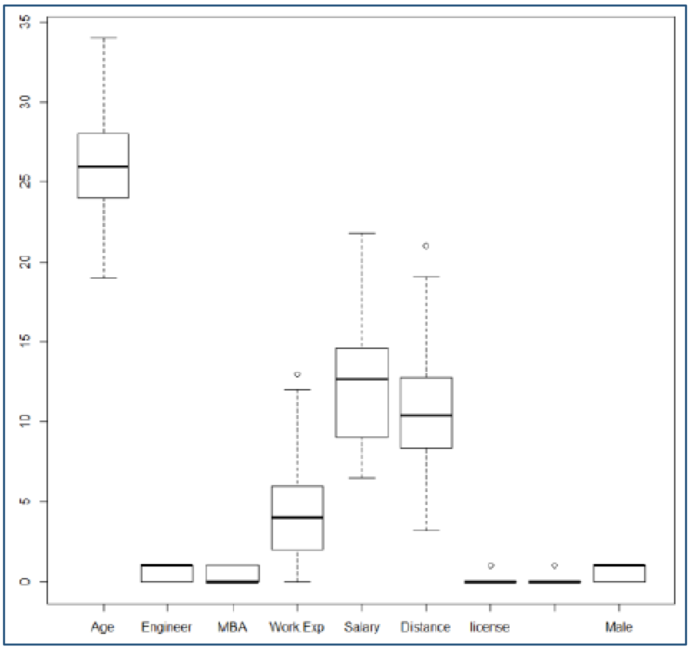
    a. Replace of outliers with mean, median or mode values.

6. Option 3

    a. Data binning/grouping by using dummy variables.

Option 3 : Data Binning / Grouping

Used Option1 for this exercise.

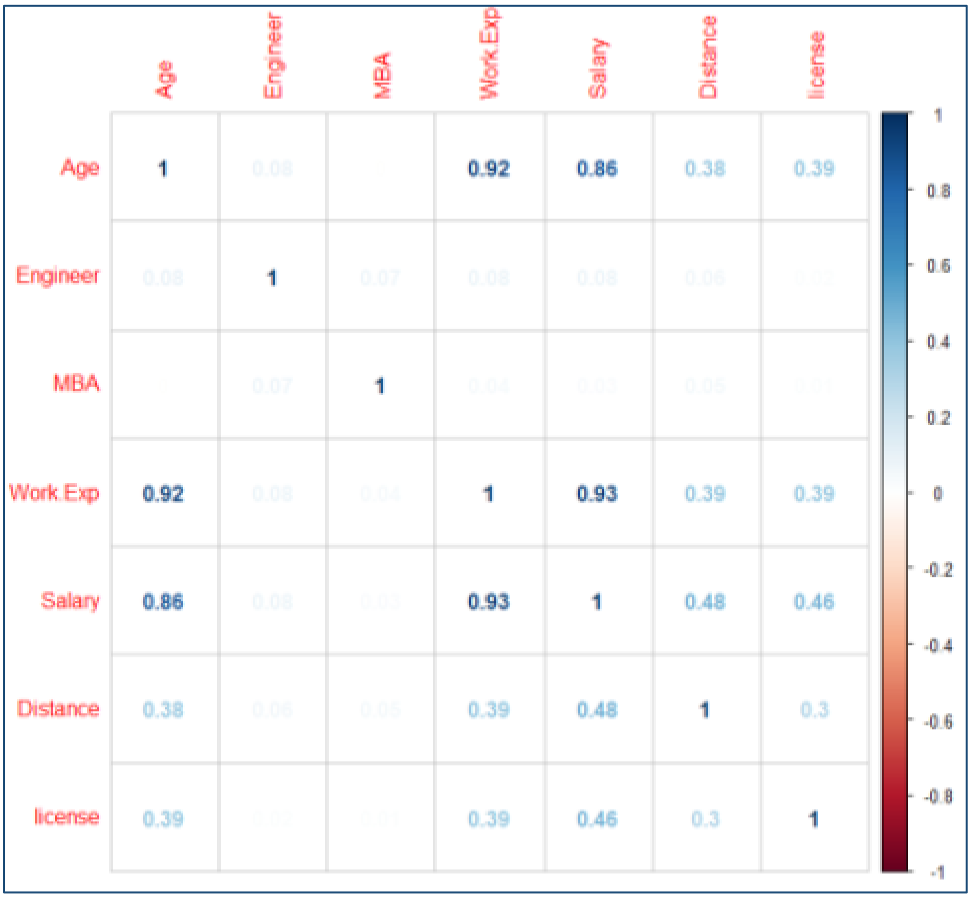


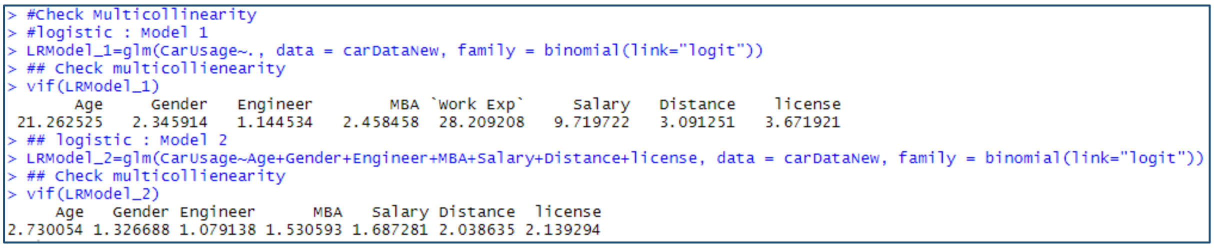
3.4 Multicollinearity check and treatment

3.4.1 Multicollinearity check

Following checks can be done on data to check multicollinearity

1. Check on signification of variables using LM model and decide on consideration of variable for model /its analysis.



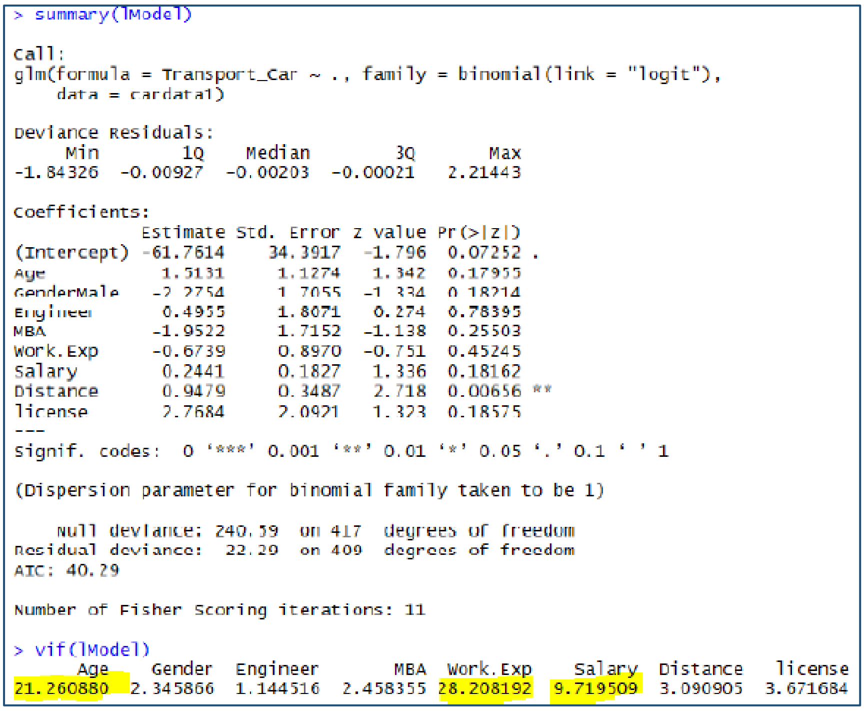


```
> #Check Multicollinearity
> #logistic : Model 1
> LRModel_1=glm(CarUsage~., data = carDataNew, family = binomial(link="logit"))
> ## Check multicollienearity
> vif(LRModel_1)
      Age     Gender   Engineer        MBA `Work Exp`     Salary   Distance    license 
21.262525   2.345914   1.144534   2.458458  28.209208   9.719722   3.091251   3.671921 
> ## logistic : Model 2
> LRModel_2=glm(CarUsage~Age+Gender+Engineer+MBA+Salary+Distance+license, data = carDataNew, family = binomial(link="logit"))
> ## Check multicollienearity
> vif(LRModel_2)
     Age   Gender Engineer      MBA   Salary Distance  license 
2.730054 1.326688 1.079138 1.530593 1.687281 2.038635 2.139294 
```

We are using Analysis of significance of features individually using LM model. Above matrix shows the correlation matrix between numerical variables. Highlighted are corelated and either can be removed. Usage of Variance inflation Factor to assess whether these correlations are really statistically significant or not.

2. Variance inflation Factor

Below is retrieval of VIF using LM model.



```
> summary(lModel)

Call:
glm(formula = Transport_Car ~ ., family = binomial(link = "logit"), 
    data = cardata1)

Deviance Residuals: 
     Min        1Q    Median        3Q       Max  
-1.84326  -0.00927  -0.00203  -0.00021   2.21443  

Coefficients:
            Estimate Std. Error z value Pr(>|z|)  
(Intercept) -61.7614    34.3917  -1.796  0.07252 .
Age           1.5131     1.1274   1.342  0.17955  
GenderMale   -2.2754     1.7055  -1.334  0.18214  
Engineer      0.4955     1.8071   0.274  0.78395  
MBA          -1.9522     1.7152  -1.138  0.25503  
Work.Exp     -0.6739     0.8970  -0.751  0.45245  
Salary        0.2441     0.1827   1.336  0.18162  
Distance      0.9479     0.3487   2.718  0.00656 **
license       2.7684     2.0921   1.323  0.18575  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 240.59  on 417  degrees of freedom
Residual deviance:  22.29  on 409  degrees of freedom
AIC: 40.29

Number of Fisher Scoring iterations: 11

> vif(lModel)
      Age    Gender  Engineer       MBA  Work.Exp    Salary  Distance   license 
21.260880  2.345866  1.144516  2.458355 28.208192  9.719509  3.090905  3.671684 
```

Key Observations:

- vif values indicate that Work experience , salary and age are correlated . As work experience is high VIF , removal of it would improve VIF of other variable and remove correlation
- VIF between 5 and 10 indicates high correlation that may be problematic. And if the VIF goes above 10, you can assume that the regression coefficients are poorly estimated due to multicollinearity.

Conclusion

Multicollinearity exists between variables due to high Variance Inflation Factor values.

3.4.2 Multicollinearity Treatment

Below are few methods applied on data for treatment of multicollinearity.

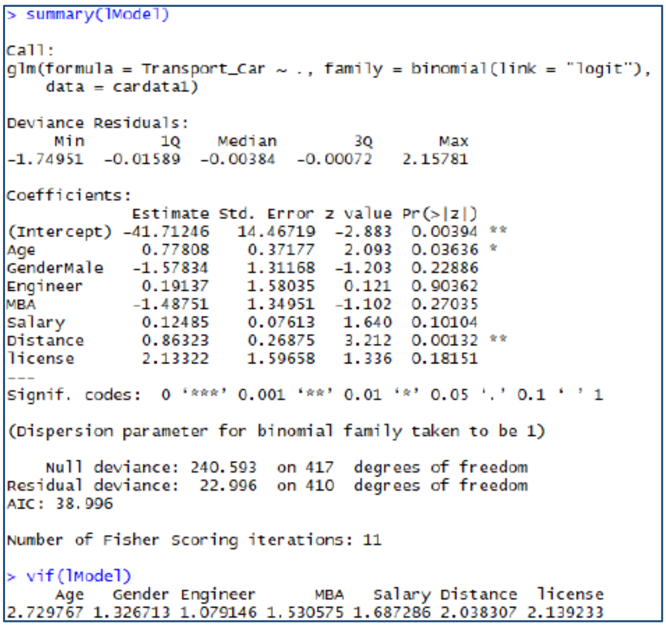
1. Remove highly correlated predictors from the model. If you have two or more factors with a high VIF, remove one from the model.

o Distance, Age and Salary are statistically significant variables. Remaining all variables are not statistically significant as far as prediction of Car usage is concerned.

o If other variables are not playing significant role the natural thing will be to delete these from further analysis. And recheck the VIF. Let's remove Work experience and check.

2. Use Partial Least Squares Regression (PLS) or Principal Components Analysis, regression methods that cut the number of predictors to a smaller set of uncorrelated components.

o Post removal, checked the multicollinearity then did data binning and rechecked.



```
> summary(lModel)

Call:
glm(formula = Transport_Car ~ ., family = binomial(link = "logit"), 
    data = cardata1)

Deviance Residuals: 
     Min        1Q    Median        3Q       Max  
-1.74951  -0.01589  -0.00384  -0.00072   2.15781  

Coefficients:
             Estimate Std. Error z value Pr(>|z|)   
(Intercept) -41.71246   14.46719  -2.883  0.00394 **
Age           0.77808    0.37177   2.093  0.03636 * 
GenderMale   -1.57834    1.31168  -1.203  0.22886   
Engineer      0.19137    1.58035   0.121  0.90362   
MBA          -1.48751    1.34951  -1.102  0.27035   
Salary        0.12485    0.07613   1.640  0.10104   
Distance      0.86323    0.26875   3.212  0.00132 **
license       2.13322    1.59658   1.336  0.18151   
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 240.593  on 417  degrees of freedom
Residual deviance:  22.996  on 410  degrees of freedom
AIC: 38.996

Number of Fisher Scoring iterations: 11

> vif(lModel)
     Age   Gender Engineer      MBA   Salary Distance  license 
2.729767 1.326713 1.079146 1.530575 1.687286 2.038307 2.139233 
```

Now post treatment i.e. removal of work experience, all variables have VIF value within 5

3.5 EDA summary

CAR usage data analysis points.

- No. of records: 418
- No. of predictor variables: 9
  1. 4 continuous numeric variables
  2. 5 ordinal variables with 2 levels [0,1]: Car, Engineer, MBA, License which have been converted from numeric to factors and one 3 levels: Transport. [We will be converting this to 2 level factors].
- Transport is a target variable as it predicts whether a customer use Car or not.
- Data wrangling can be done for couple of fields like Transport, Salary, Age etc. so to allow for more convenient consumption and organization of the data.
- Based on the model or algorithm used, Flag fields need to be converted from int to Factor
- Outliers noticed in data and treatment done as discussed in earlier sections.
- Multicollinearity exist in data and it is checked by couple of methods and treated by data deletion for less significant variable, data binning etc.
- The numeric data is normally distributed for most of the variables even though the spread of Gaussian curves is around different points for transport using car and otherwise.

- Outliers are present in every numeric column. Hence, we will be keeping the data as it is and will not be treating any of the outliers as it will introduce high bias if we treat the data at this scale.
  - Transport has 3 levels: Public Transport, 2wheeler and car. As we are only analyzing if a person uses car or not, we will group PublicTransport and 2-wheeler together as non-car in a new column called CarUsage. This is our target variable.
  - With 2 distinct classes: 1 for those who use car and 0 for those who don't use car.
  - 91.62% of the people did not use car while 8.37% did. This is terms of count is: 383 people did not use car while 35 people used car as their mode of commute.
  - The data is highly unbalanced and we need to treat it to build appropriate models.
  - In the sample data, 71.05% of the people are men/males and 28.95% of them are women/females. The number of females is significantly lesser than 50% in the company.
  - In the data sample, 74.88% of the people are Engineers while 25.12% of the people are not.
- We notice a normal distribution for most of the numeric variables for the CarUsage of both 1 and 0 but not along the same curves. That is, the behavioural patterns of the people who use the car and who don't use a car are very distinct and different even though they are uniforml y distributed.
- Young employees of the age group < 27 yrs did not drive a car at all which means they were used to public transport or 2wheeler which is cheaper/easier option for them.
- People who had to travel more than 10km were more likely to drive a car rather than use public transport or 2wheeler. This shows that the Public Transport system may not be easily accessible/widespread/well-connected or may be very time consuming for long distance. Hence, people prefer more comfortable mode of transportive., car. 2-wheeler may not be very comfortable as well.
- We also notice that young employees with less than 5yrs of experience did not drive a car at all while employees with around 15 yrs of experience and above that seem to use car as their main mode of transportation. This is an indirect indication that using a car isn't the cheapest mode of transport available to young employees [price of car and petrol/diesel] while older employees were capable of buying and regularly using a car.
- We see different peaks with respect to the Salary and car usage. Some people with lesser salaries preferred cars while majority of the people who used car had Salary around 40 units. Also, majority of the people whose salaries are around 15 units preferred public transport or 2wheeler as their main mode of commute.
- We will be removing the column: Work Experience going forth if the values of VIF is higher than 5. Also, if the VIF is still higher than 5 then we can remove License as well. This will not make a difference as we have the column Salary which is highly correlated with Work Experience while License is correlated with CarUsage. A person needs to have license in order to drive a car so most of the people who commute in a car have license.

❖ The Dependent and independent variables that we will be using throughout will be:

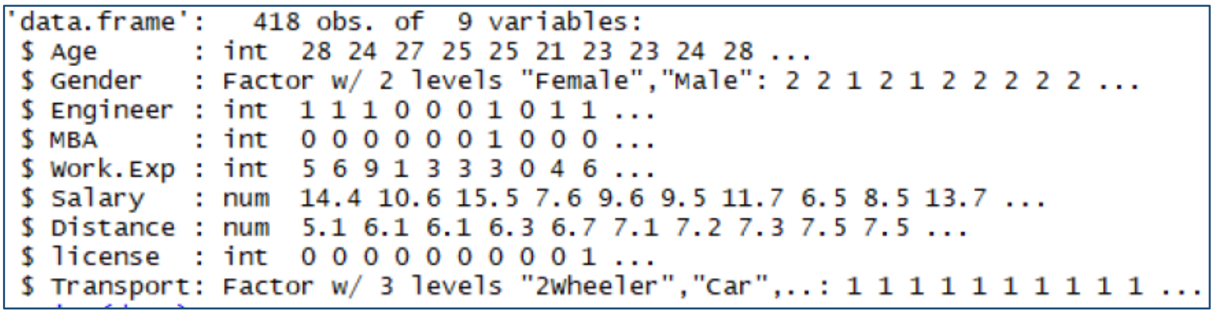
Dependent Variable	Independent Variable
CarUsage [based on Transport variable]	Age
	Gender
	Engineer
	MBA
	Work Exp
	Salary
	Distance
	License

4. Build Models and check for the best ones

4.1 Prepare data for analysis using SMOTE

Smote: Synthetic Minority Oversampling Technique to handle imbalance in binary classification. Post that we can use Logistic for predict probability.

Step 1: Let’s check structure of Data set Cars.



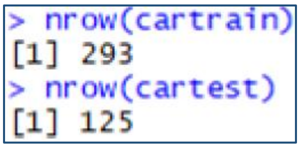
```
'data.frame':   418 obs. of  9 variables:
 $ Age      : int  28 24 27 25 25 21 23 23 24 28 ...
 $ Gender   : Factor w/ 2 levels "Female","Male": 2 2 1 2 1 2 2 2 2 2 ...
 $ Engineer : int  1 1 1 0 0 0 1 0 1 1 ...
 $ MBA      : int  0 0 0 0 0 0 1 0 0 0 ...
 $ Work.Exp : int  5 6 9 1 3 3 3 0 4 6 ...
 $ Salary   : num  14.4 10.6 15.5 7.6 9.6 9.5 11.7 6.5 8.5 13.7 ...
 $ Distance : num  5.1 6.1 6.1 6.3 6.7 7.1 7.2 7.3 7.5 7.5 ...
 $ license  : int  0 0 0 0 0 0 0 0 0 1 ...
 $ Transport: Factor w/ 3 levels "2Wheeler","Car",..: 1 1 1 1 1 1 1 1 1 1 ...
```

Step 2: To simplify Transport as Car usage, convert to Car 1 or 0 if others.

Step 3: Remove work experience column .

Step 4: Set the seed then partition the data with train and test as 70:30.

Step 5: Check the distribution

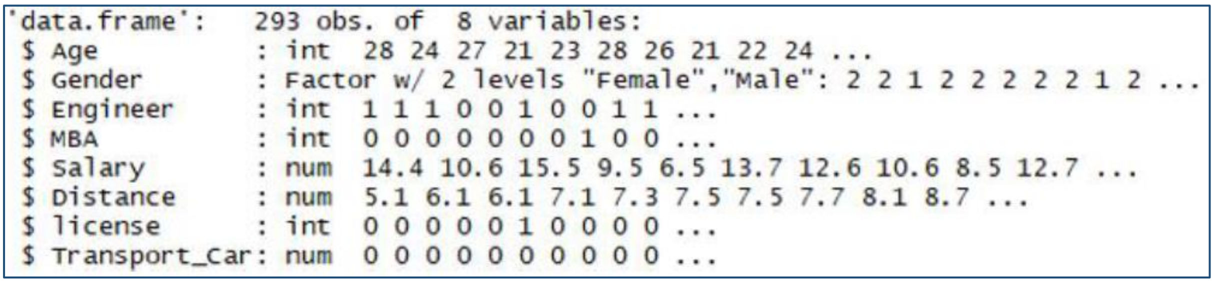


```
> nrow(cartrain)
[1] 293
> nrow(cartest)
[1] 125
```

Step 6: Let’s check Car usage rate in both dataset.]

Car usage rate is very less in both data set. Thus, it falls under minority imbalance data so SMOTE can be used to treat imbalance.

Step 7: Let check structure of Train data and convert target variable to Factor if needed as for SMOTE that is mandatory.



```
'data.frame':   293 obs. of  8 variables:
 $ Age          : int  28 24 27 21 23 28 26 21 22 24 ...
 $ Gender       : Factor w/ 2 levels "Female","Male": 2 2 1 2 2 2 2 2 1 2 ...
 $ Engineer     : int  1 1 1 0 0 1 0 0 1 1 ...
 $ MBA          : int  0 0 0 0 0 0 0 1 0 0 ...
 $ Salary       : num  14.4 10.6 15.5 9.5 6.5 13.7 12.6 10.6 8.5 12.7 ...
 $ Distance     : num  5.1 6.1 6.1 7.1 7.3 7.5 7.5 7.7 8.1 8.7 ...
 $ license      : int  0 0 0 0 0 1 0 0 0 0 ...
 $ Transport_Car: num  0 0 0 0 0 0 0 0 0 0 ...
```

Step 9: Now data is ready for SMOTE,

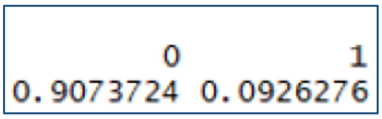
In SMOTE we have to define our equation

#perc.over means that minority class will be added for every value of perc.over

#now we have increased the minority class. We are adding 48 for every minority class sample. - perc.over

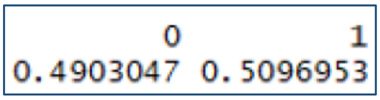
#We are subtracting 10 for every 100 - perc.under. We are taking out of the majority class as well.

With used equation minority class is not treated , let’s try another to treat imbalance.



```
        0         1 
0.9073724 0.0926276 
```

Step 10: Treat imbalance using another SMOTE equation, This equation provided better results and it balances data pretty well . Lets further improve our model or equation.



```
        0         1
0.4903047 0.5096953
```

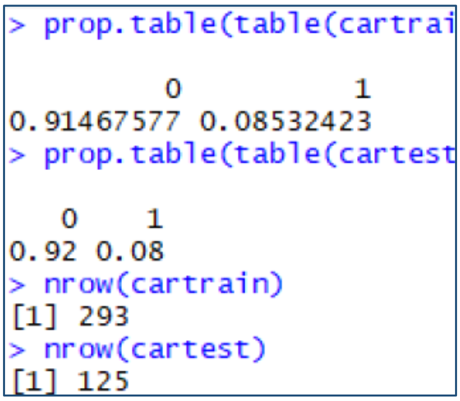
Step 11: Further treatment using another SMOTE equation,
#perc.over means that minority class will be added for every value of perc.over
#now we have increased the minority class. We are adding 25 for every minority class sample. -perc.over
#We are subtracting 15 for every 100 - perc.under. We are taking out of the majority class as well.
This is good distribution .Lets see the impact on accuracy and other performance measures using this SMOTE data.

4.2 Model using Logistic Regression Technique on SMOTE data

Logistic regression as a special case of linear regression when the outcome variable is categorical, where we are using log of odds as dependent variable. In simple words, it predicts the probability of occurrence of an event by fitting data to a logit function.

4.2.1 Building Logistic Regression Model

Build Model using Logistic Regression Technique on train and test dataset 70:30.
Check Car usage rate on dataset:



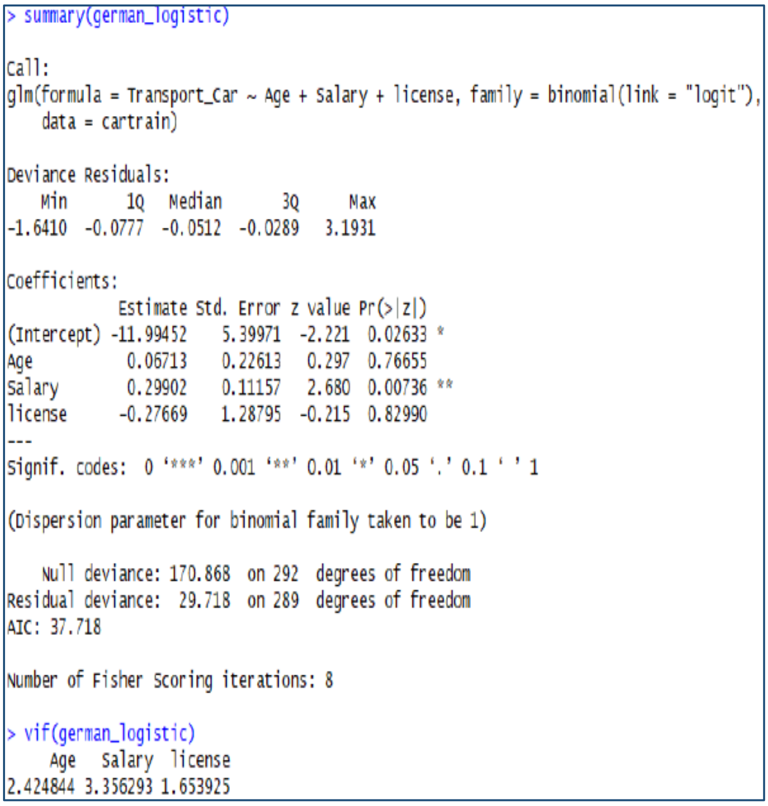
```
> prop.table(table(cartrai

         0          1
0.91467577 0.08532423
> prop.table(table(cartest

   0    1
0.92 0.08
> nrow(cartrain)
[1] 293
> nrow(cartest)
[1] 125
```

Start estimating a Logistic Regression Model using the glm (generalized linear model) function.
Install Package Caret .

- GLM does not assume a linear relationship between dependent and independent variables. However, it assumes a linear relationship between link function and independent variables in logit model.
- The dependent variable need not to be normally distributed.
- It does not uses OLS (Ordinary Least Square) for parameter estimation. Instead, it uses maximum likelihood estimation (MLE).
- Errors need to be independent but not normally distributed.

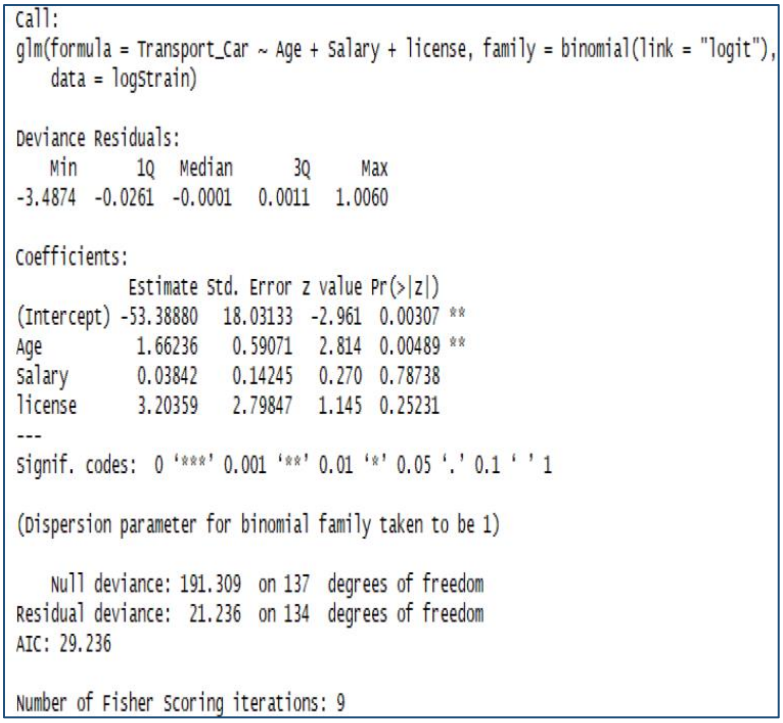
let’s use all independent variables and compare it with target value CarData. After executing couple of models and comparing AIC values least AIC value model is considered:

Logistic regression before applying SMOTE:



```
> summary(german_logistic)

Call:
glm(formula = Transport_Car ~ Age + Salary + license, family = binomial(link = "logit"),
    data = cartrain)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6410  -0.0777  -0.0512  -0.0289   3.1931

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -11.99452    5.39971  -2.221  0.02633 *
Age           0.06713    0.22613   0.297  0.76655
Salary        0.29902    0.11157   2.680  0.00736 **
license      -0.27669    1.28795  -0.215  0.82990
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 170.868  on 292  degrees of freedom
Residual deviance:  29.718  on 289  degrees of freedom
AIC: 37.718

Number of Fisher Scoring iterations: 8

> vif(german_logistic)
     Age   Salary  license
2.424844 3.356293 1.653925
```

Logistic regression on SMOTE Train data set:



```
Call:
glm(formula = Transport_Car ~ Age + Salary + license, family = binomial(link = "logit"),
    data = logStrain)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.4874  -0.0261  -0.0001   0.0011   1.0060

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -53.38880   18.03133  -2.961  0.00307 **
Age           1.66236    0.59071   2.814  0.00489 **
Salary        0.03842    0.14245   0.270  0.78738
license       3.20359    2.79847   1.145  0.25231
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 191.309  on 137  degrees of freedom
Residual deviance:  21.236  on 134  degrees of freedom
AIC: 29.236

Number of Fisher Scoring iterations: 9
```

Feature Analysis:

The top four most-relevant feature is Age, salary, license.

4.2.2 Logistic Regression Model Validity

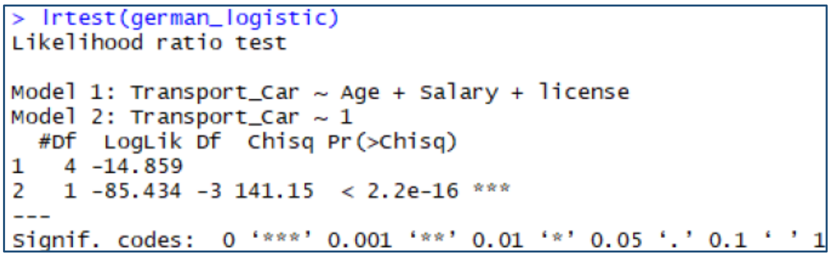
Step 1: Likelihood ratio test

Overall validaity of the model - Can I use Logistic regression? use lrtest . The full model vs Null model

Difference between these multiplied by 2(G2 statistics) is the ChiSq. Going by the results, we say that

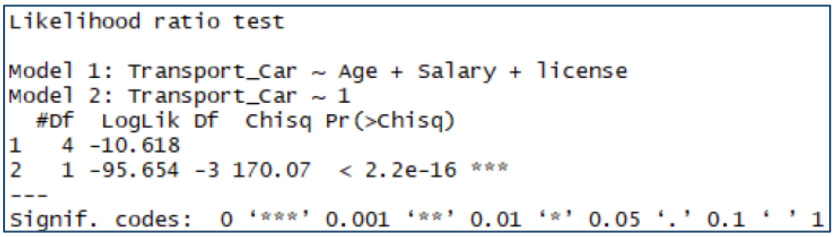
null hypothesis can be rejected and accept alternate hypothesis (at least one variable is siginificant)

Likelihood ratio test before applying SMOTE:



```
> lrtest(german_logistic)
Likelihood ratio test

Model 1: Transport_Car ~ Age + Salary + license
Model 2: Transport_Car ~ 1
  #Df  LogLik Df  Chisq Pr(>Chisq)    
1   4 -14.859                         
2   1 -85.434 -3 141.15  < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Likelihood ratio test on SMOTE Train data set:



```
Likelihood ratio test

Model 1: Transport_Car ~ Age + Salary + license
Model 2: Transport_Car ~ 1
  #Df  LogLik Df  Chisq Pr(>Chisq)    
1   4 -10.618                         
2   1 -95.654 -3 170.07  < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Step 2 Mcfadden R Square.

This also gives likelood with Null vs Full model along with G2.

Gives three R squared , R sq McFadden is the best and recommended, R sq Mamimum Likelihood,

R2 McFadden = Full - Null/Gain(i.eG2/2)

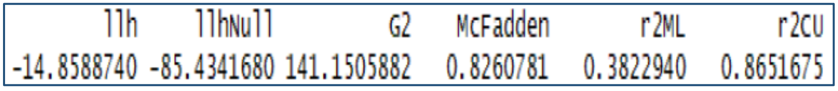
in Logistic, we will not get R2 closer to 1. Very rarely we get.

24.8%(i.e R2 McFadden) of the intercept only model has been explained/calibrated by the Full Model.

Any Pseudo r2 less than 10% is poor fit.

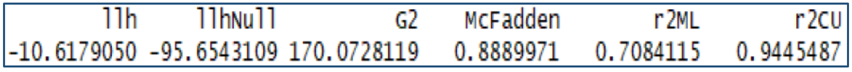
Between 10% to 20% is just satisfactory. 20 to 40% is good model. Above 40% excellent.

Mcfadden R Square before applying SMOTE:



```
        llh      llhNull           G2     McFadden         r2ML         r2CU 
-14.8588740  -85.4341680  141.1505882    0.8260781    0.3822940    0.8651675 
```

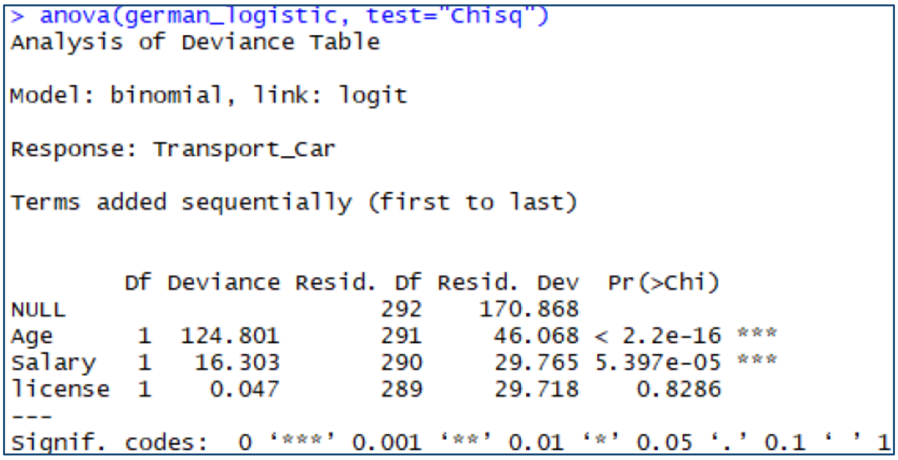
Mcfadden R Square on SMOTE Train data set:



```
        llh      llhNull           G2     McFadden         r2ML         r2CU 
-10.6179050  -95.6543109  170.0728119    0.8889971    0.7084115    0.9445487 
```

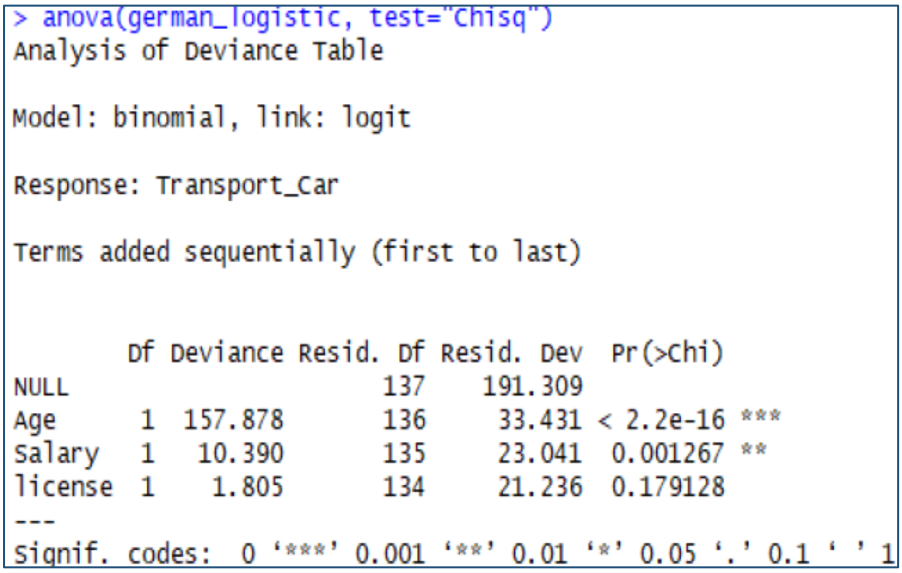
4.2.3 Deviance table Analysis

Deviance trend can be seen from below so to reveal drop of deviance by adding features anova before applying SMOTE:



```
> anova(german_logistic, test="Chisq")
Analysis of Deviance Table

Model: binomial, link: logit

Response: Transport_Car

Terms added sequentially (first to last)


        Df Deviance Resid. Df Resid. Dev  Pr(>Chi)    
NULL                      292    170.868              
Age      1  124.801       291     46.068 < 2.2e-16 ***
Salary   1   16.303       290     29.765 5.397e-05 ***
license  1    0.047       289     29.718    0.8286    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

anova on SMOTE Train data set:



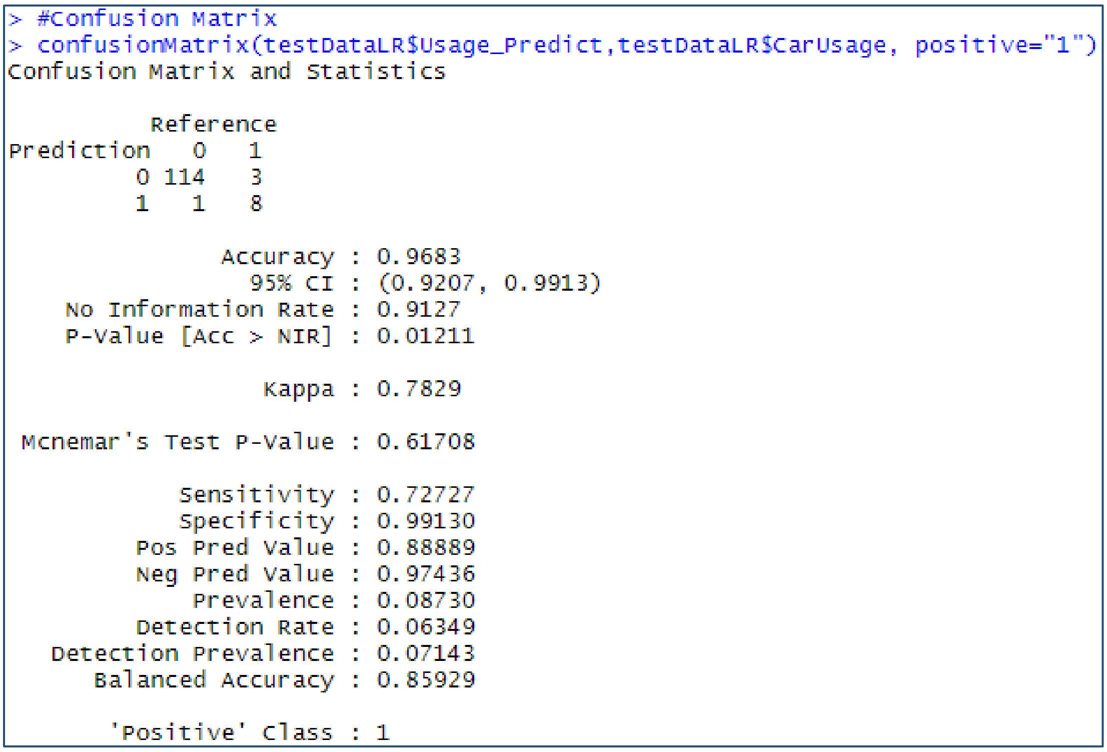
```
> anova(german_logistic, test="Chisq")
Analysis of Deviance Table

Model: binomial, link: logit

Response: Transport_Car

Terms added sequentially (first to last)


        Df Deviance Resid. Df Resid. Dev  Pr(>Chi)    
NULL                      137    191.309              
Age      1  157.878       136     33.431 < 2.2e-16 ***
Salary   1   10.390       135     23.041  0.001267 ** 
license  1    1.805       134     21.236  0.179128    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Analyzing the deviance table we can see the drop in deviance when adding each variable one at a time. Adding license significantly reduces the residual deviance

4.2.4 Performance Measures :

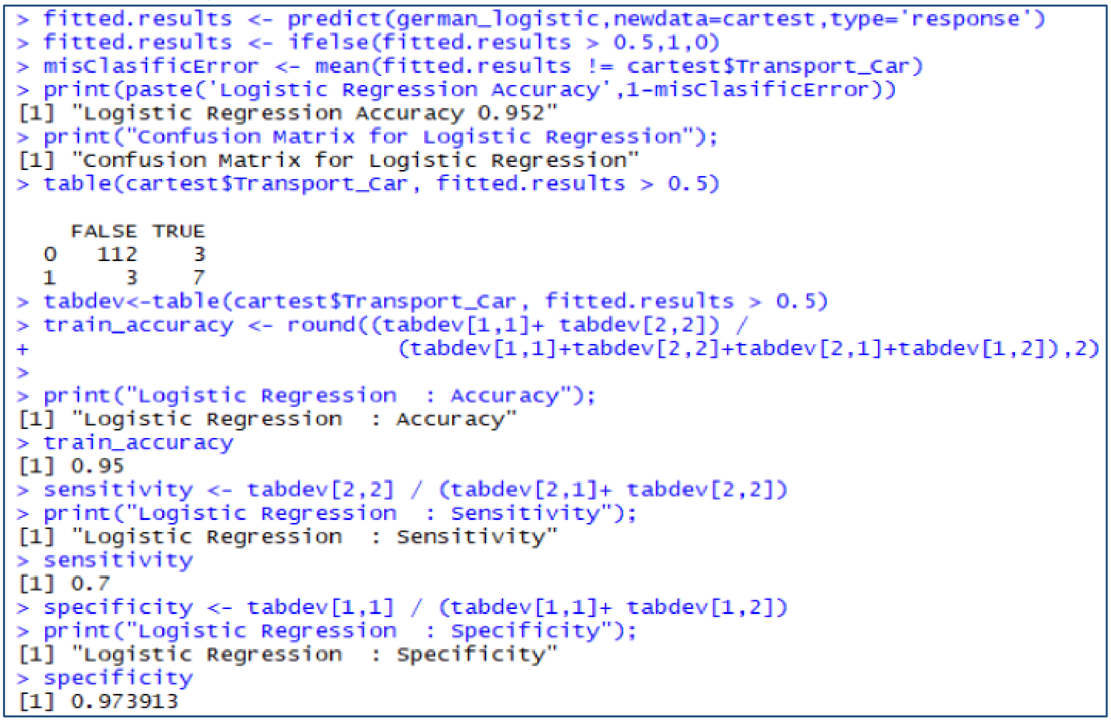
4.2.4.1 Prediction -Confusion Matrix

Assessing the predictive ability of the Logistic Regression model



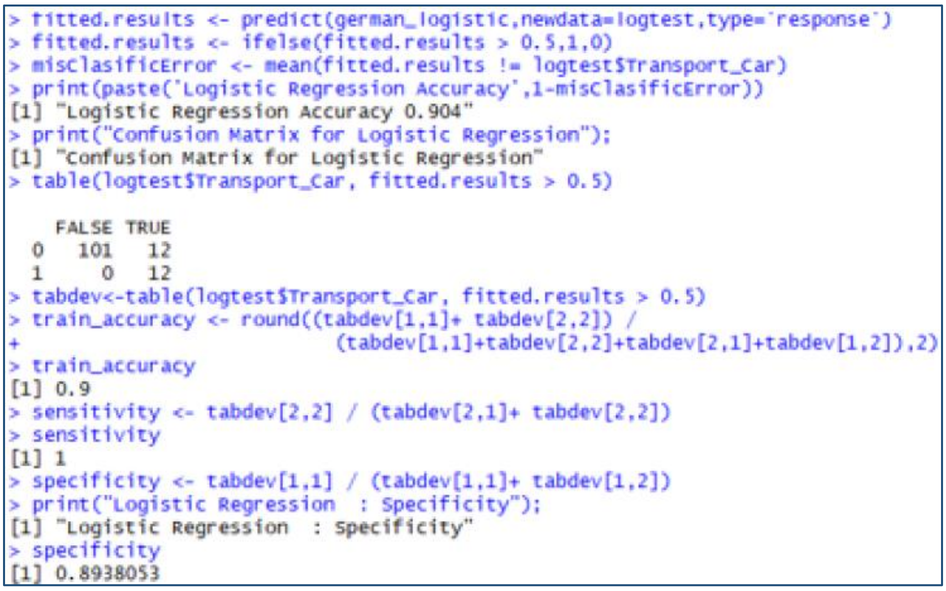
```
> #Confusion Matrix
> confusionMatrix(testDataLR$Usage_Predict,testDataLR$CarUsage, positive="1")
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 114   3
         1   1   8

               Accuracy : 0.9683
                 95% CI : (0.9207, 0.9913)
    No Information Rate : 0.9127
    P-Value [Acc > NIR] : 0.01211

                  Kappa : 0.7829

 Mcnemar's Test P-Value : 0.61708

            Sensitivity : 0.72727
            Specificity : 0.99130
         Pos Pred Value : 0.88889
         Neg Pred Value : 0.97436
             Prevalence : 0.08730
         Detection Rate : 0.06349
   Detection Prevalence : 0.07143
      Balanced Accuracy : 0.85929

       'Positive' Class : 1
```

Predictive ability -Test Data



```
> fitted.results <- predict(german_logistic,newdata=cartest,type='response')
> fitted.results <- ifelse(fitted.results > 0.5,1,0)
> misClasificError <- mean(fitted.results != cartest$Transport_Car)
> print(paste('Logistic Regression Accuracy',1-misClasificError))
[1] "Logistic Regression Accuracy 0.952"
> print("Confusion Matrix for Logistic Regression");
[1] "Confusion Matrix for Logistic Regression"
> table(cartest$Transport_Car, fitted.results > 0.5)

    FALSE TRUE
  0   112    3
  1     3    7
> tabdev<-table(cartest$Transport_Car, fitted.results > 0.5)
> train_accuracy <- round((tabdev[1,1]+ tabdev[2,2]) /
+                       (tabdev[1,1]+tabdev[2,2]+tabdev[2,1]+tabdev[1,2]),2)
>
> print("Logistic Regression  : Accuracy");
[1] "Logistic Regression  : Accuracy"
> train_accuracy
[1] 0.95
> sensitivity <- tabdev[2,2] / (tabdev[2,1]+ tabdev[2,2])
> print("Logistic Regression  : Sensitivity");
[1] "Logistic Regression  : Sensitivity"
> sensitivity
[1] 0.7
> specificity <- tabdev[1,1] / (tabdev[1,1]+ tabdev[1,2])
> print("Logistic Regression  : Specificity");
[1] "Logistic Regression  : Specificity"
> specificity
[1] 0.973913
```

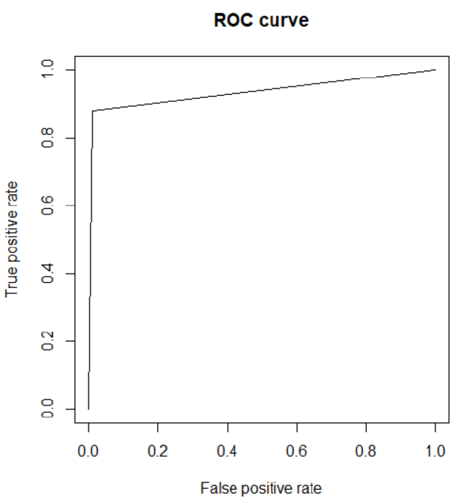
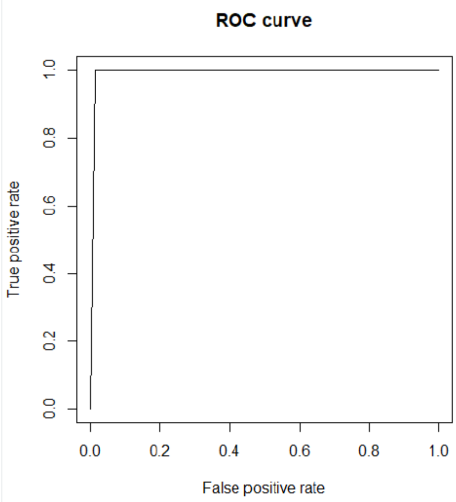
Predictive ability-Test Data

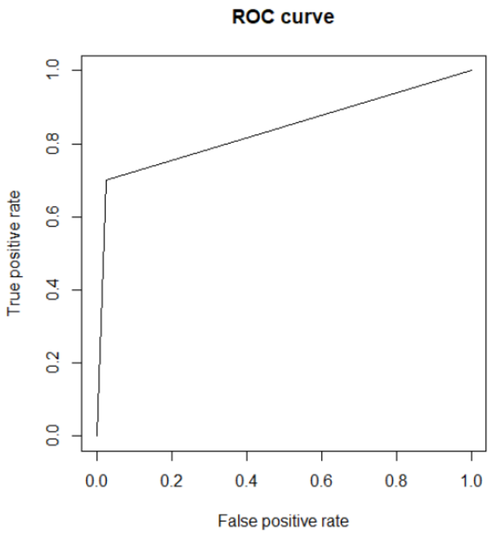
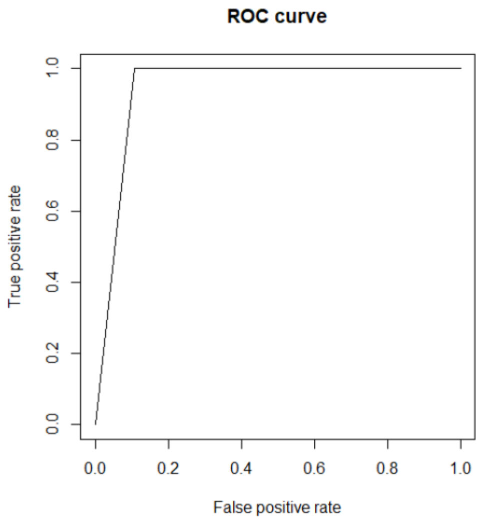


```
> fitted.results <- predict(german_logistic,newdata=logtest,type='response')
> fitted.results <- ifelse(fitted.results > 0.5,1,0)
> misClasificError <- mean(fitted.results != logtest$Transport_Car)
> print(paste('Logistic Regression Accuracy',1-misClasificError))
[1] "Logistic Regression Accuracy 0.904"
> print("Confusion Matrix for Logistic Regression");
[1] "Confusion Matrix for Logistic Regression"
> table(logtest$Transport_Car, fitted.results > 0.5)

    FALSE TRUE
  0   101   12
  1     0   12
> tabdev<-table(logtest$Transport_Car, fitted.results > 0.5)
> train_accuracy <- round((tabdev[1,1]+ tabdev[2,2]) /
+                         (tabdev[1,1]+tabdev[2,2]+tabdev[2,1]+tabdev[1,2]),2)
> train_accuracy
[1] 0.9
> sensitivity <- tabdev[2,2] / (tabdev[2,1]+ tabdev[2,2])
> sensitivity
[1] 1
> specificity <- tabdev[1,1] / (tabdev[1,1]+ tabdev[1,2])
> print("Logistic Regression  : Specificity");
[1] "Logistic Regression  : Specificity"
> specificity
[1] 0.8938053
```

The above model with an accuracy of 90% is indeed a good quality model.

4.2.4.2 AUC charts

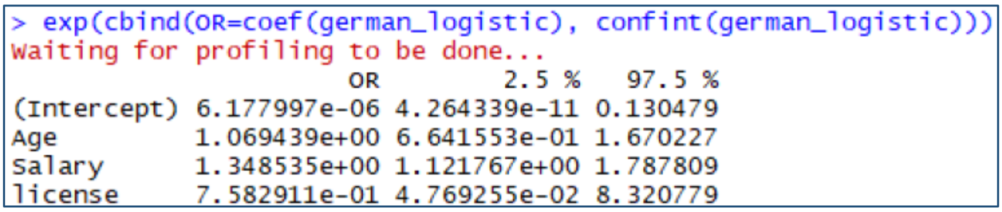
Predictive ability -Train	Predictive ability-Smote Train
	

Predictive ability -Test	Predictive ability-Smote Test
	

4.2.4.3 Odd Ratio

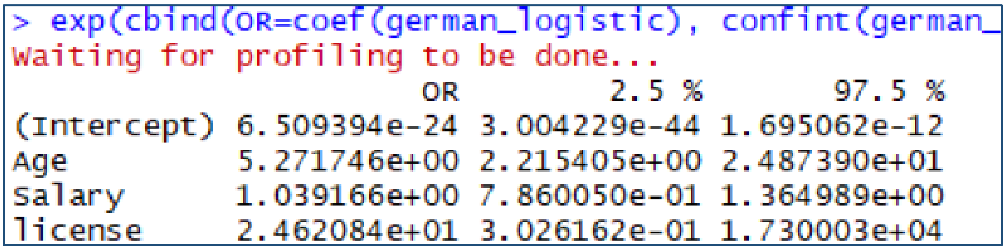
One of the interesting performance measurements in logistic regression is Odds Ratio. Basically, Odds ratio is what the odds of an event is happening.

Odd Ratio



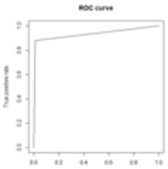
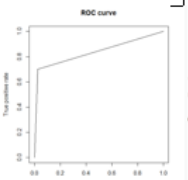
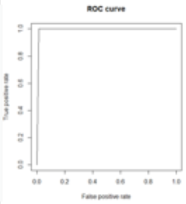
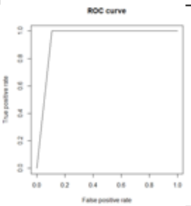
```
> exp(cbind(OR=coef(german_logistic), confint(german_logistic)))
Waiting for profiling to be done...
                      OR        2.5 %   97.5 %
(Intercept) 6.177997e-06 4.264339e-11 0.130479
Age         1.069439e+00 6.641553e-01 1.670227
Salary      1.348535e+00 1.121767e+00 1.787809
license     7.582911e-01 4.769255e-02 8.320779
```

Predictive ability-Smote Test



```
> exp(cbind(OR=coef(german_logistic), confint(german_
Waiting for profiling to be done...
                      OR        2.5 %       97.5 %
(Intercept) 6.509394e-24 3.004229e-44 1.695062e-12
Age         5.271746e+00 2.215405e+00 2.487390e+01
Salary      1.039166e+00 7.860050e-01 1.364989e+00
license     2.462084e+01 3.026162e-01 1.730003e+04
```

Performance Measures of Logistic Regression Model at a glance

Performance Measures	Logistic Regression		Logistic Regression with SMOTE	
	Train	Test	Train	Test
Confusion Matrix	FALSE TRUE 0 265 3 1 3 22	FALSE TRUE 0 112 3 1 3 7	FALSE TRUE 0 68 1 1 0 69	FALSE TRUE 0 101 12 1 0 12
Accuracy	0.98	0.95	0.99	0.9
Specificity	0.988	0.973	0.985	0.893
Sensitivity	0.88	0.7	1	1
KS	0.868	0.673	0.985	0.893
AUC	0.934	0.836	0.992	0.9469
Classification Error Rate	0.02	0.05	0.01	0.1
Gini	0.868	0.672	0.984	0.8938
AUC Curve				

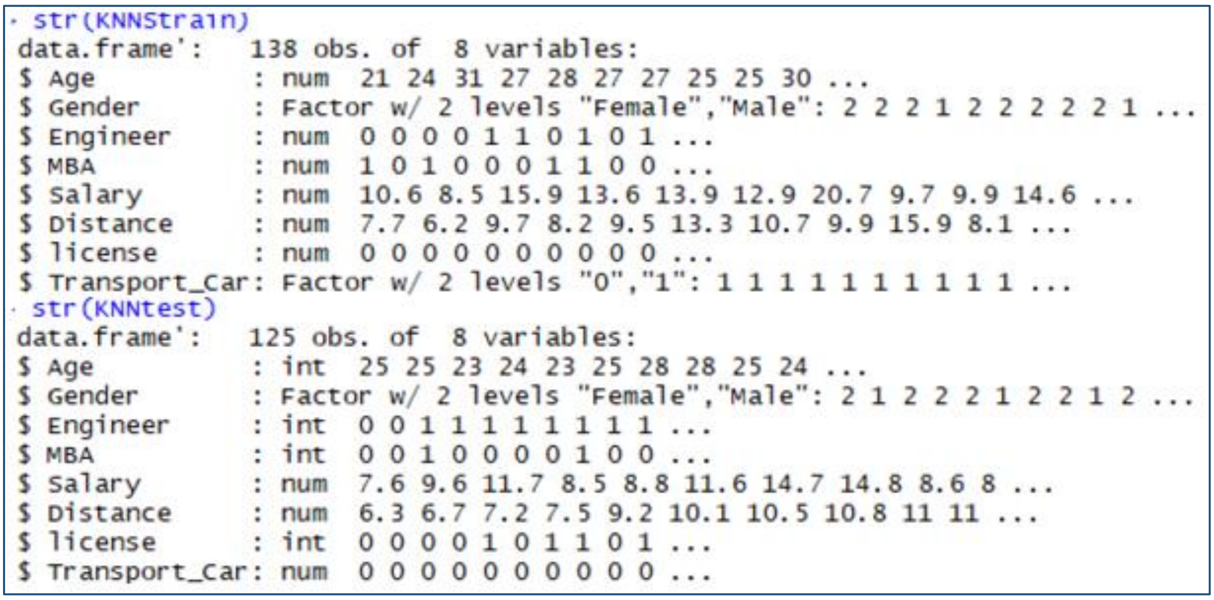
Interpretation:

Logistic regression on SMOTE training data is good as it is accurate True positive.

4.3 Model using KNN Algorithm

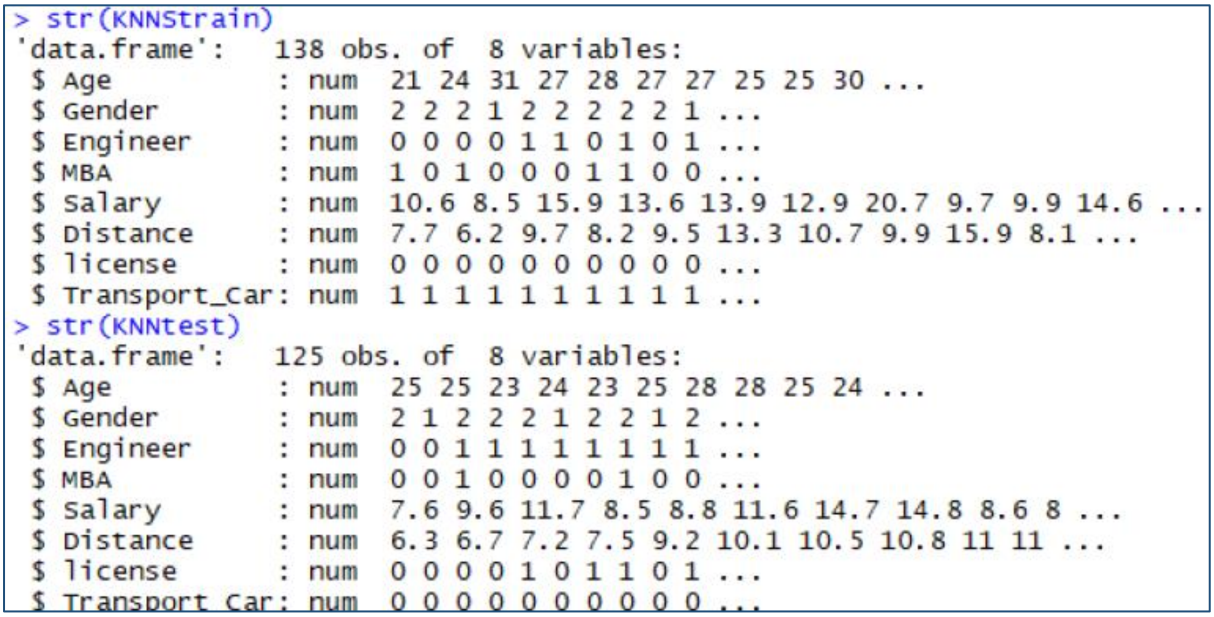
4.3.1 Building Model using KNN

KNN is executed on numerical variable so below the structure of data should be converted to Numeric



```
> str(KNNStrain)
data.frame':   138 obs. of  8 variables:
 $ Age          : num  21 24 31 27 28 27 27 25 25 30 ...
 $ Gender       : Factor w/ 2 levels "Female","Male": 2 2 2 1 2 2 2 2 2 1 ...
 $ Engineer     : num  0 0 0 0 1 1 0 1 0 1 ...
 $ MBA          : num  1 0 1 0 0 0 1 1 0 0 ...
 $ Salary       : num  10.6 8.5 15.9 13.6 13.9 12.9 20.7 9.7 9.9 14.6 ...
 $ Distance     : num  7.7 6.2 9.7 8.2 9.5 13.3 10.7 9.9 15.9 8.1 ...
 $ license      : num  0 0 0 0 0 0 0 0 0 0 ...
 $ Transport_Car: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
> str(KNNtest)
data.frame':   125 obs. of  8 variables:
 $ Age          : int  25 25 23 24 23 25 28 28 25 24 ...
 $ Gender       : Factor w/ 2 levels "Female","Male": 2 1 2 2 2 1 2 2 1 2 ...
 $ Engineer     : int  0 0 1 1 1 1 1 1 1 1 ...
 $ MBA          : int  0 0 1 0 0 0 0 1 0 0 ...
 $ Salary       : num  7.6 9.6 11.7 8.5 8.8 11.6 14.7 14.8 8.6 8 ...
 $ Distance     : num  6.3 6.7 7.2 7.5 9.2 10.1 10.5 10.8 11 11 ...
 $ license      : int  0 0 0 0 1 0 1 1 0 1 ...
 $ Transport_Car: num  0 0 0 0 0 0 0 0 0 0 ...
```

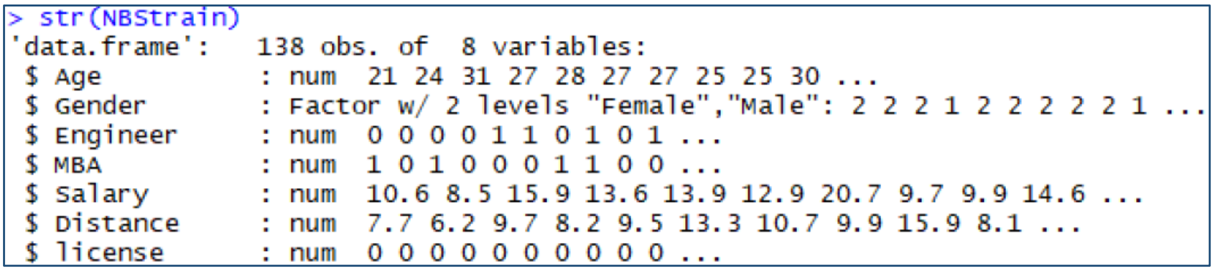
Data ready for KNN



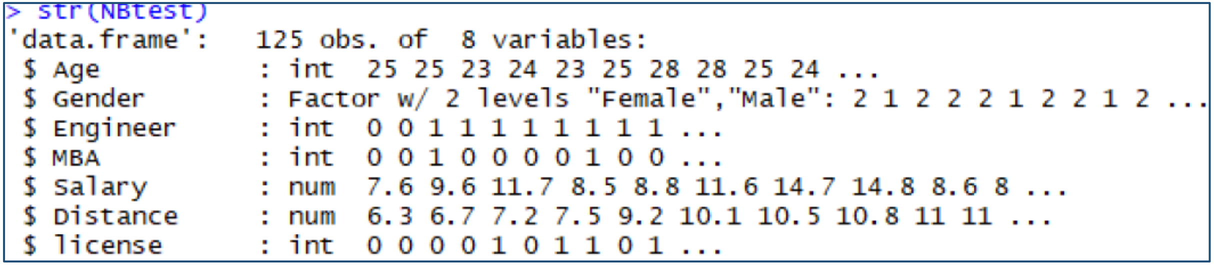
```
> str(KNNStrain)
'data.frame':	138 obs. of  8 variables:
 $ Age          : num  21 24 31 27 28 27 27 25 25 30 ...
 $ Gender       : num  2 2 2 1 2 2 2 2 2 1 ...
 $ Engineer     : num  0 0 0 0 1 1 0 1 0 1 ...
 $ MBA          : num  1 0 1 0 0 0 1 1 0 0 ...
 $ Salary       : num  10.6 8.5 15.9 13.6 13.9 12.9 20.7 9.7 9.9 14.6 ...
 $ Distance     : num  7.7 6.2 9.7 8.2 9.5 13.3 10.7 9.9 15.9 8.1 ...
 $ license      : num  0 0 0 0 0 0 0 0 0 0 ...
 $ Transport_Car: num  1 1 1 1 1 1 1 1 1 1 ...
> str(KNNtest)
'data.frame':	125 obs. of  8 variables:
 $ Age          : num  25 25 23 24 23 25 28 28 25 24 ...
 $ Gender       : num  2 1 2 2 2 1 2 2 1 2 ...
 $ Engineer     : num  0 0 1 1 1 1 1 1 1 1 ...
 $ MBA          : num  0 0 1 0 0 0 0 1 0 0 ...
 $ Salary       : num  7.6 9.6 11.7 8.5 8.8 11.6 14.7 14.8 8.6 8 ...
 $ Distance     : num  6.3 6.7 7.2 7.5 9.2 10.1 10.5 10.8 11 11 ...
 $ license      : num  0 0 0 0 1 0 1 1 0 1 ...
 $ Transport_Car: num  0 0 0 0 0 0 0 0 0 0 ...
```

4.3.2 Scaling of data using Normalization

4.4.3 Data Partitioning



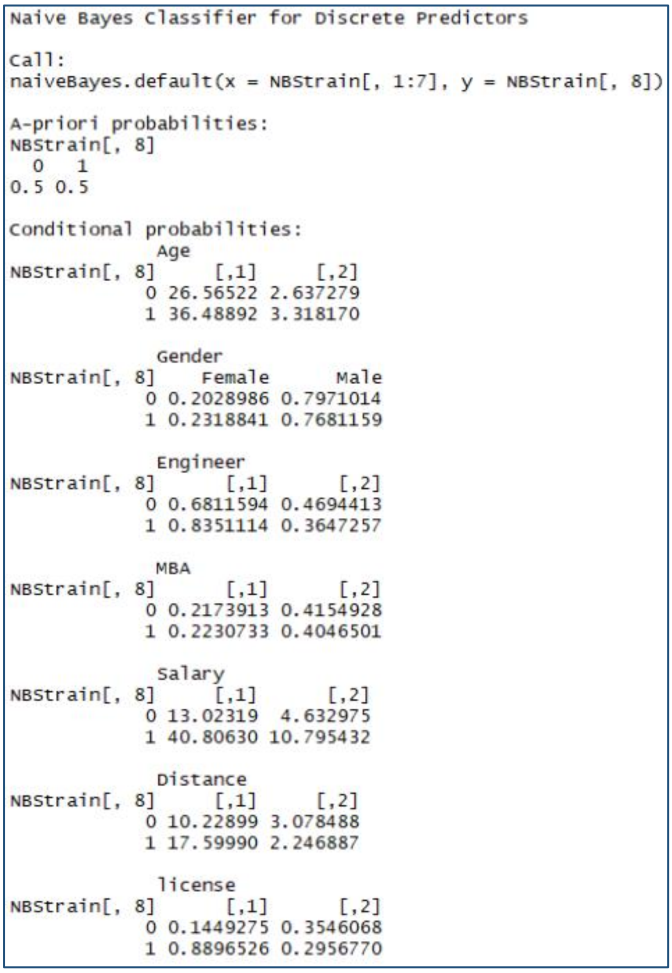
```
> str(NBStrain)
'data.frame':	138 obs. of  8 variables:
 $ Age          : num  21 24 31 27 28 27 27 25 25 30 ...
 $ Gender       : Factor w/ 2 levels "Female","Male": 2 2 2 1 2 2 2 2 2 1 ...
 $ Engineer     : num  0 0 0 0 1 1 0 1 0 1 ...
 $ MBA          : num  1 0 1 0 0 0 1 1 0 0 ...
 $ Salary       : num  10.6 8.5 15.9 13.6 13.9 12.9 20.7 9.7 9.9 14.6 ...
 $ Distance     : num  7.7 6.2 9.7 8.2 9.5 13.3 10.7 9.9 15.9 8.1 ...
 $ license      : num  0 0 0 0 0 0 0 0 0 0 ...
```



```
> str(NBtest)
'data.frame':	125 obs. of  8 variables:
 $ Age          : int  25 25 23 24 23 25 28 28 25 24 ...
 $ Gender       : Factor w/ 2 levels "Female","Male": 2 1 2 2 2 1 2 2 1 2 ...
 $ Engineer     : int  0 0 1 1 1 1 1 1 1 1 ...
 $ MBA          : int  0 0 1 0 0 0 0 1 0 0 ...
 $ Salary       : num  7.6 9.6 11.7 8.5 8.8 11.6 14.7 14.8 8.6 8 ...
 $ Distance     : num  6.3 6.7 7.2 7.5 9.2 10.1 10.5 10.8 11 11 ...
 $ license      : int  0 0 0 0 1 0 1 1 0 1 ...
```

4.4.4 Executing Naïve Bayes Algorithm on data

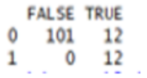
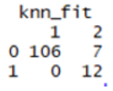
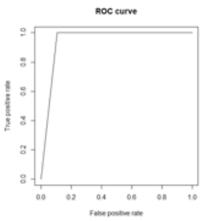
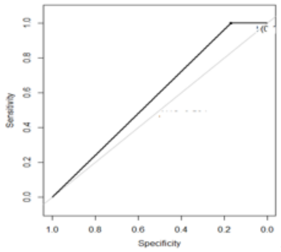
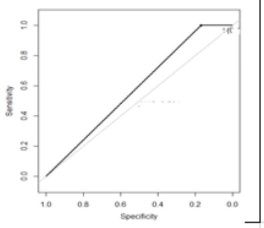
Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other. Calculate the prior probabilities from the count of the training data. So, it should follow the proportion of the parent dataset.



```
Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = NBStrain[, 1:7], y = NBStrain[, 8])

A-priori probabilities:
NBStrain[, 8]
  0   1
0.5 0.5

Conditional probabilities:
             Age
NBStrain[, 8]     [,1]     [,2]
            0 26.56522 2.637279
            1 36.48892 3.318170

             Gender
NBStrain[, 8]    Female      Male
            0 0.2028986 0.7971014
            1 0.2318841 0.7681159

             Engineer
NBStrain[, 8]      [,1]      [,2]
            0 0.6811594 0.4694413
            1 0.8351114 0.3647257

             MBA
NBStrain[, 8]      [,1]      [,2]
            0 0.2173913 0.4154928
            1 0.2230733 0.4046501

             Salary
NBStrain[, 8]     [,1]      [,2]
            0 13.02319  4.632975
            1 40.80630 10.795432

             Distance
NBStrain[, 8]     [,1]     [,2]
            0 10.22899 3.078488
            1 17.59990 2.246887

             license
NBStrain[, 8]      [,1]      [,2]
            0 0.1449275 0.3546068
            1 0.8896526 0.2956770
```

4.5 Compare 3 Model Performance: Logistic Regression Vs KNN Vs Naïve Bayes

To check the performance, there are following performance measures and its parameters which has been considered and compared for evaluation

Performance Measures	Logistic Regression	KNN	Naïve Bayes
Confusion Matrix	 FALSE TRUE 0 101 12 1 0 12	 knn_fit 1 2 0 106 7 1 0 12	0 1 0 109 4 1 0 12
Accuracy	90%	**94%**	97%
Specificity	89%	**100%**	100%
Sensitivity	100%	**94%**	96%
KS	89%		
AUC	95%	**58%**	56%
Classification Error Rate	10%	**6%**	3%
Gini	89%	**17%**	**12%**
AUC Curve			

Area Under the ROC curve (AUC – ROC)

The ROC curve is the plot between sensitivity and (1- specificity). (1- specificity) is also known as false positive ate and sensitivity is also known as True Positive rate. This coordinate becomes on point in our ROC curve. To bring this curve down to a single number, we find the area under this curve (AUC).

Hence AUC itself is the ratio under the curve and the total area. Following are a few thumb rules:

- .90-1 = excellent (A)
- .80-.90 = good (B)
- .70-.80 = fair (C)
- .60-.70 = poor (D)
- .50-.60 = fail (F)

Considering our problem and models, KNN figures are not in good range. Logistic Regression stand out for comparison purpose.

K-S or Kolmogorov-Smirnov

K-S or Kolmogorov-Smirnov chart measures performance of classification models. More accurately, K-S is a measure of the degree of separation between the positive and negative distributions. The K-S is 100, if the

scores partition the population into two separate groups in which one group contains all the positives and the other all the negatives.

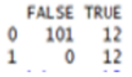
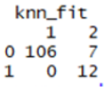
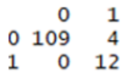
Gini coefficient

Gini coefficient is sometimes used in classification problems. Gini coefficient can be straight away derived from the AUC ROC number. Gini is nothing but ratio between area between the ROC curve and the diagonal line & the area of the above triangle. Following is the formulae used:
Gini above 60% is a good model so for our problem it is good for all models.

Classification Error Rate

The lower the classification error rate, higher the model accuracy, resulting in a better model. The classification error rate can be reduced if there were more independent variables were present for modeling.
For our problem, CeR is good for all models.

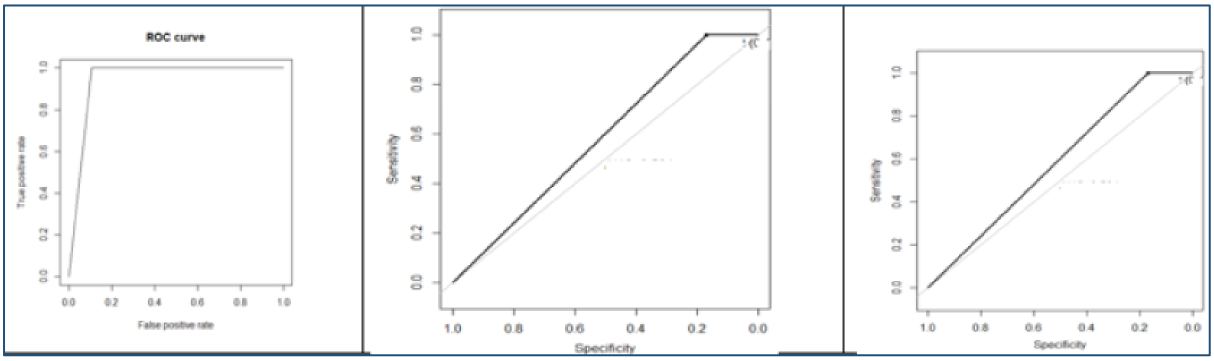
Confusion Matrix Interpretation

Performance Measures	Logistic Regression	KNN	Naïve Bayes
Confusion Matrix	 FALSE TRUE 0 101 12 1 0 12	 knn_fit 1 2 0 106 7 1 0 12	 0 1 0 109 4 1 0 12

Confusion Matrix compares actual outcome to predicted outcome.It descrive performance of model on set of test data for which true value are known. It interpret whether employee use Car to commute or not .

• Two possible predicted outputs whether employee would use car as transport or not. True /1 means they would.
• Classfier made a total of 125 predictions ie 125 employee were studied
• Out of 125 cases, classifier predicted “Yes” 24 for logistic Regression ,19 for KNN , 16 for Naïve Bayes
• And ” No” 101 for Logistic , 106 for KNN , 109 for NB.
• In reality , 12 use car for transport for all models in our case .
So based on above as Yes for Logistic is more .Logistic moel stand out .
AUC charts.

Logistic Regression AUC is nearing True positive rate so it is better as compared to other 2 models –

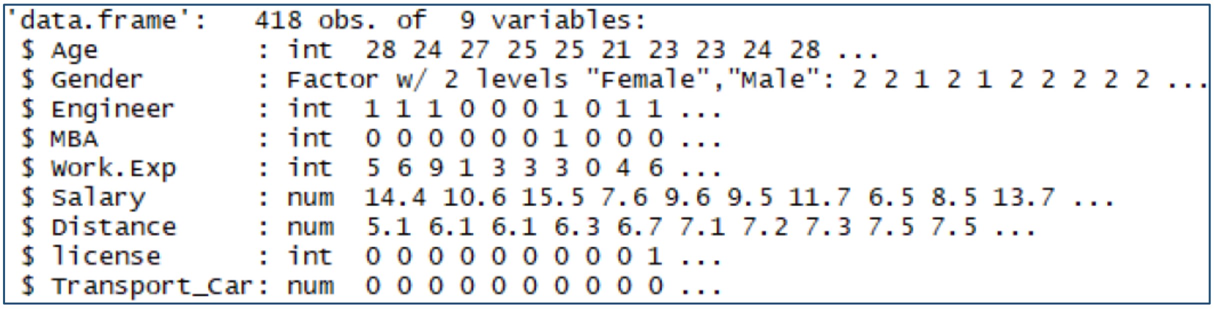


Remarks on Model Validation “Which Model is best”
As all 3 models performance measures are quite close but considering accuracy and AUC, Logistic Regression are providing better performance as compared to Naïve Bayes and KNN considering above 4 to 5 performance measures considered.

5.1 Model using Bagging Technique

5.1.1 Building Model using Bagging

For Bagging, make sure that your dependent variable is a numeric so let’s check the structure of data



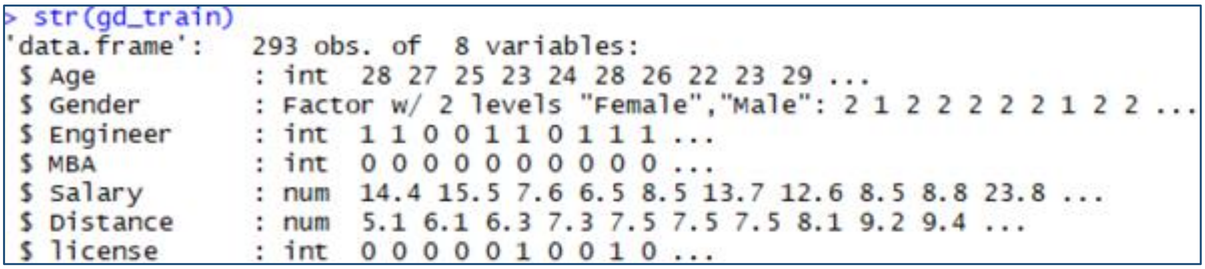
```
'data.frame':	418 obs. of  9 variables:
 $ Age          : int  28 24 27 25 25 21 23 23 24 28 ...
 $ Gender       : Factor w/ 2 levels "Female","Male": 2 2 1 2 1 2 2 2 2 2 ...
 $ Engineer     : int  1 1 1 0 0 0 1 0 1 1 ...
 $ MBA          : int  0 0 0 0 0 0 1 0 0 0 ...
 $ Work.Exp     : int  5 6 9 1 3 3 3 0 4 6 ...
 $ Salary       : num  14.4 10.6 15.5 7.6 9.6 9.5 11.7 6.5 8.5 13.7 ...
 $ Distance     : num  5.1 6.1 6.1 6.3 6.7 7.1 7.2 7.3 7.5 7.5 ...
 $ license      : int  0 0 0 0 0 0 0 0 0 1 ...
 $ Transport_Car: num  0 0 0 0 0 0 0 0 0 0 ...
```

5.1.2 Data preparation for Bagging method

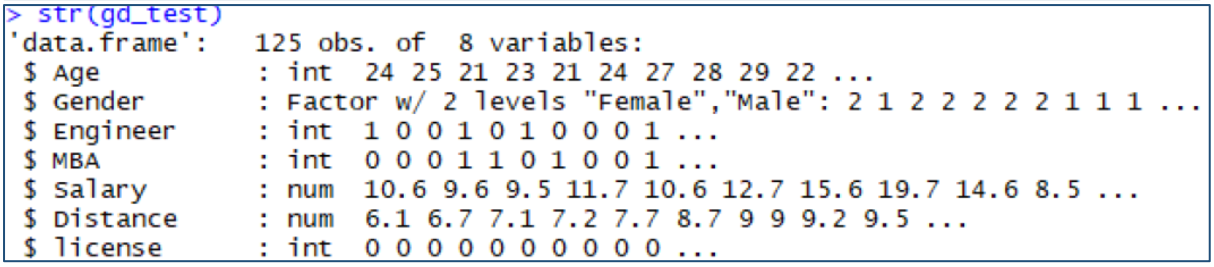
Bagging: each bootstrap has the same size as original. Accomplished by sampling with replacement. Converting Target variables as numerical if needed.

5.1.3 Data Partitioning (Train 70% and Test 30%)

Distribution of data between Train and test data set is 70:30. Also percentage of Car usage in both dataset which is comparatively very less.



```
> str(gd_train)
'data.frame':	293 obs. of  8 variables:
 $ Age         : int  28 27 25 23 24 28 26 22 23 29 ...
 $ Gender      : Factor w/ 2 levels "Female","Male": 2 1 2 2 2 2 2 1 2 2 ...
 $ Engineer    : int  1 1 0 0 1 1 0 1 1 1 ...
 $ MBA         : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Salary      : num  14.4 15.5 7.6 6.5 8.5 13.7 12.6 8.5 8.8 23.8 ...
 $ Distance    : num  5.1 6.1 6.3 7.3 7.5 7.5 7.5 8.1 9.2 9.4 ...
 $ license     : int  0 0 0 0 0 1 0 0 1 0 ...
```



```
> str(gd_test)
'data.frame':	125 obs. of  8 variables:
 $ Age         : int  24 25 21 23 21 24 27 28 29 22 ...
 $ Gender      : Factor w/ 2 levels "Female","Male": 2 1 2 2 2 2 2 1 1 1 ...
 $ Engineer    : int  1 0 0 1 0 1 0 0 0 1 ...
 $ MBA         : int  0 0 0 1 1 0 1 0 0 1 ...
 $ Salary      : num  10.6 9.6 9.5 11.7 10.6 12.7 15.6 19.7 14.6 8.5 ...
 $ Distance    : num  6.1 6.7 7.1 7.2 7.7 8.7 9 9 9.2 9.5 ...
 $ license     : int  0 0 0 0 0 0 0 0 0 0 ...
```

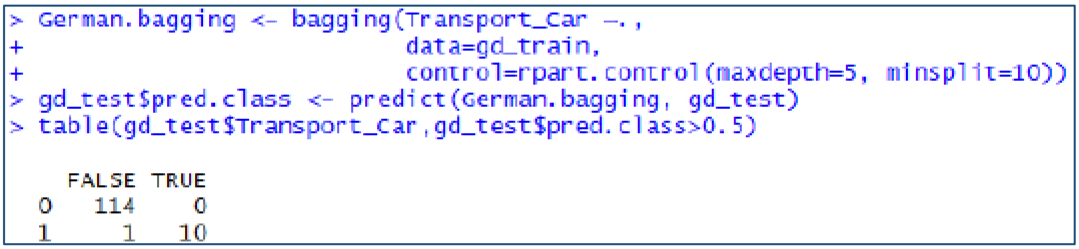
5.1.4 Executing Bagging Algorithm on data

Bagging (aka Bootstrap Aggregating): is a way to decrease the variance of your prediction by generating additional data for training from your original dataset using combinations with repetitions to produce multisets of the same cardinality/size as your original data.

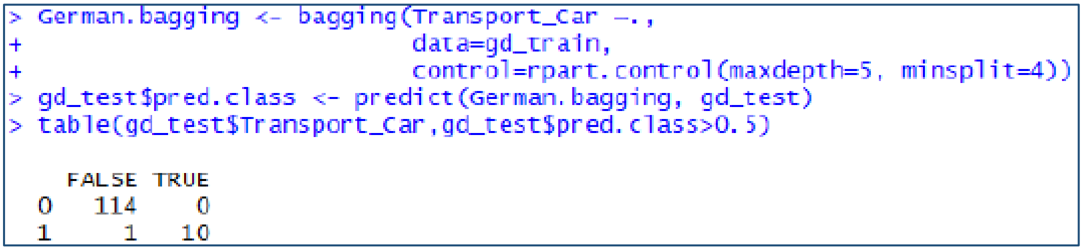
we can modify the maxdepth and minsplit if needed to get couple of equation with best accuracy.

Below are many of few models tried with

maxdepth 5, and minsplit 10,15,4- Best was 5,15

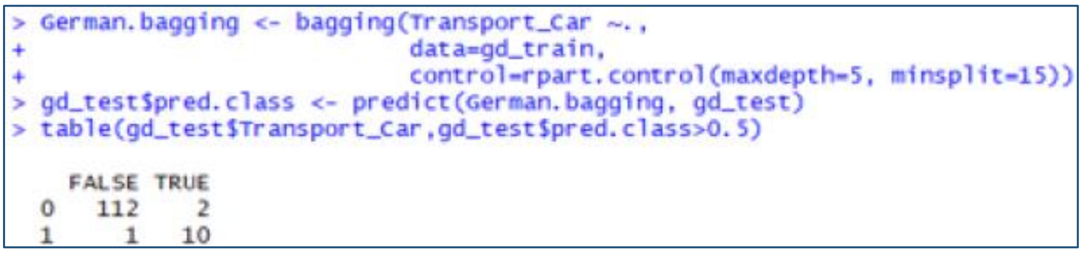


```
> German.bagging <- bagging(Transport_Car ~.,
+                           data=gd_train,
+                           control=rpart.control(maxdepth=5, minsplit=10))
> gd_test$pred.class <- predict(German.bagging, gd_test)
> table(gd_test$Transport_Car,gd_test$pred.class>0.5)

    FALSE TRUE
  0   114    0
  1     1   10
```



```
> German.bagging <- bagging(Transport_Car ~.,
+                           data=gd_train,
+                           control=rpart.control(maxdepth=5, minsplit=4))
> gd_test$pred.class <- predict(German.bagging, gd_test)
> table(gd_test$Transport_Car,gd_test$pred.class>0.5)

    FALSE TRUE
  0   114    0
  1     1   10
```

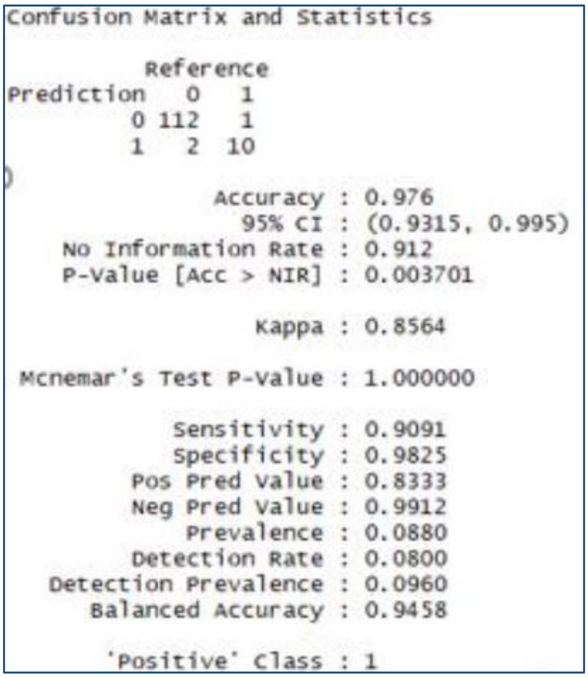
Chosen one is:



```
> German.bagging <- bagging(Transport_Car ~.,
+                           data=gd_train,
+                           control=rpart.control(maxdepth=5, minsplit=15))
> gd_test$pred.class <- predict(German.bagging, gd_test)
> table(gd_test$Transport_Car,gd_test$pred.class>0.5)

    FALSE TRUE
  0   112    2
  1     1   10
```

5.1.5 Performance Measures

5.1.5.1 Prediction -Confusion Matrix



```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 112   1
         1   2  10

               Accuracy : 0.976
                 95% CI : (0.9315, 0.995)
    No Information Rate : 0.912
    P-Value [Acc > NIR] : 0.003701

                  Kappa : 0.8564

 Mcnemar's Test P-Value : 1.000000

            Sensitivity : 0.9091
            Specificity : 0.9825
         Pos Pred Value : 0.8333
         Neg Pred Value : 0.9912
             Prevalence : 0.0880
         Detection Rate : 0.0800
   Detection Prevalence : 0.0960
      Balanced Accuracy : 0.9458

       'Positive' Class : 1
```

The above model with an accuracy of 99.2% is indeed a good quality model.

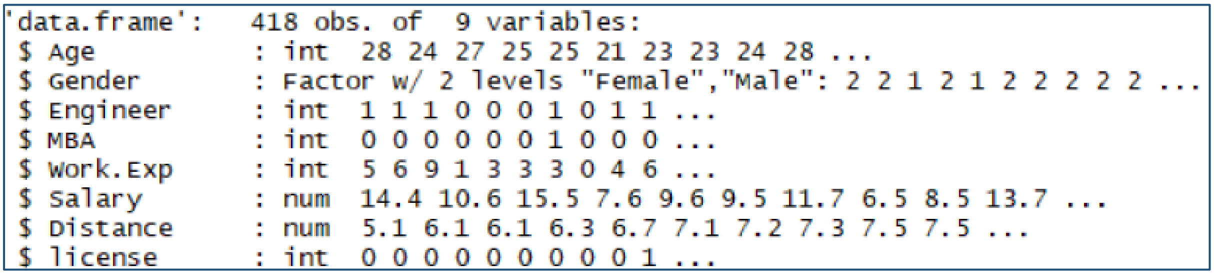
6.1 Model using Boosting Technique

6.1.1 Building Model using Boosting various technique

Boosting is a class of ensemble learning techniques for regression and classification problems. Boosting aims to build a set of weak learners to create one 'strong' learner (i.e. a predictive model that predicts the response variable with a high degree of accuracy).

We would use couple of boosting methods.

For Boosting, make sure that all data is numeric so let's check the structure of data and do conversion

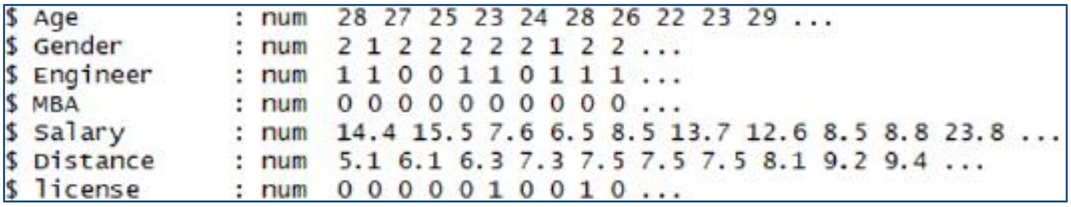


```
'data.frame':   418 obs. of  9 variables:
 $ Age          : int  28 24 27 25 25 21 23 23 24 28 ...
 $ Gender       : Factor w/ 2 levels "Female","Male": 2 2 1 2 1 2 2 2 2 2 ...
 $ Engineer     : int  1 1 1 0 0 0 1 0 1 1 ...
 $ MBA          : int  0 0 0 0 0 0 1 0 0 0 ...
 $ Work.Exp     : int  5 6 9 1 3 3 3 0 4 6 ...
 $ Salary       : num  14.4 10.6 15.5 7.6 9.6 9.5 11.7 6.5 8.5 13.7 ...
 $ Distance     : num  5.1 6.1 6.1 6.3 6.7 7.1 7.2 7.3 7.5 7.5 ...
 $ license      : int  0 0 0 0 0 0 0 0 0 1 ...
```

6.1.2 Data preparation for GBM (Gradient Boosting method)

Gradient Boosting is a boosting method which aims to optimize an arbitrary (differentiable) cost function

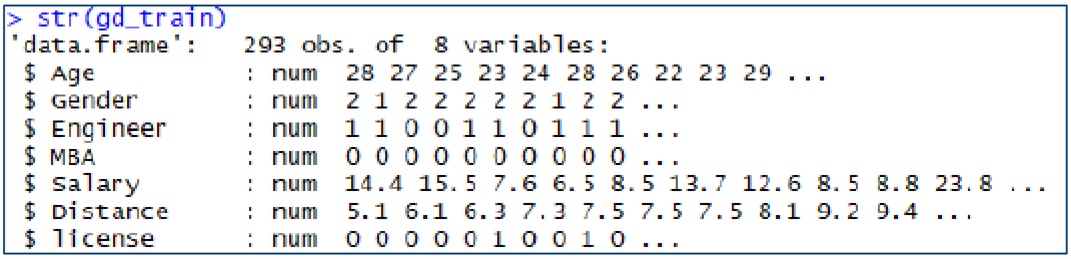
Converting all variables including target variable as numerical.



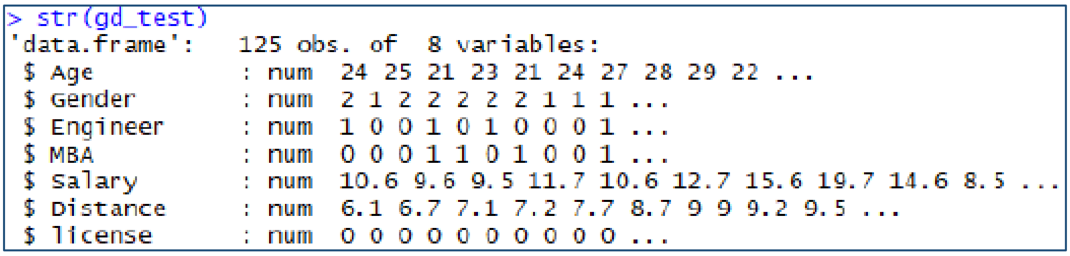
```
$ Age          : num  28 27 25 23 24 28 26 22 23 29 ...
$ Gender       : num  2 1 2 2 2 2 2 1 2 2 ...
$ Engineer     : num  1 1 0 0 1 1 0 1 1 1 ...
$ MBA          : num  0 0 0 0 0 0 0 0 0 0 ...
$ Salary       : num  14.4 15.5 7.6 6.5 8.5 13.7 12.6 8.5 8.8 23.8 ...
$ Distance     : num  5.1 6.1 6.3 7.3 7.5 7.5 7.5 8.1 9.2 9.4 ...
$ license      : num  0 0 0 0 0 1 0 0 1 0 ...
```

6.1.3 Data Partitioning (Train 70% and Test 30%)

Distribution of data between Train and test data set is 70:30. Also percentage of Car usage in both dataset which is comparatively very less.



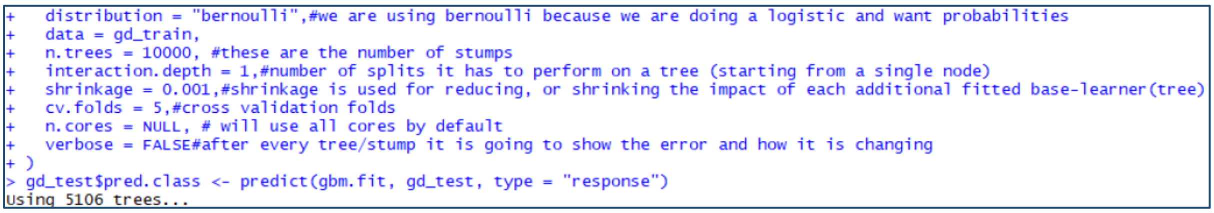
```
> str(gd_train)
'data.frame':   293 obs. of  8 variables:
 $ Age         : num  28 27 25 23 24 28 26 22 23 29 ...
 $ Gender      : num  2 1 2 2 2 2 2 1 2 2 ...
 $ Engineer    : num  1 1 0 0 1 1 0 1 1 1 ...
 $ MBA         : num  0 0 0 0 0 0 0 0 0 0 ...
 $ Salary      : num  14.4 15.5 7.6 6.5 8.5 13.7 12.6 8.5 8.8 23.8 ...
 $ Distance    : num  5.1 6.1 6.3 7.3 7.5 7.5 7.5 8.1 9.2 9.4 ...
 $ license     : num  0 0 0 0 0 1 0 0 1 0 ...
```



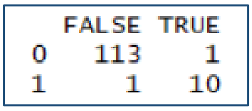
```
> str(gd_test)
'data.frame':   125 obs. of  8 variables:
 $ Age         : num  24 25 21 23 21 24 27 28 29 22 ...
 $ Gender      : num  2 1 2 2 2 2 2 1 1 1 ...
 $ Engineer    : num  1 0 0 1 0 1 0 0 0 1 ...
 $ MBA         : num  0 0 0 1 1 0 1 0 0 1 ...
 $ Salary      : num  10.6 9.6 9.5 11.7 10.6 12.7 15.6 19.7 14.6 8.5 ...
 $ Distance    : num  6.1 6.7 7.1 7.2 7.7 8.7 9 9 9.2 9.5 ...
 $ license     : num  0 0 0 0 0 0 0 0 0 0 ...
```

6.1.4 Executing GBM Algorithm on data

Using Bernoulli distribution as we are doing logistic and want probabilities . Below is best model post few trials of values which uses around 5000 trees.



```
+   distribution = "bernoulli",#we are using bernoulli because we are doing a logistic and want probabilities
+   data = gd_train,
+   n.trees = 10000, #these are the number of stumps
+   interaction.depth = 1,#number of splits it has to perform on a tree (starting from a single node)
+   shrinkage = 0.001,#shrinkage is used for reducing, or shrinking the impact of each additional fitted base-learner(tree)
+   cv.folds = 5,#cross validation folds
+   n.cores = NULL, # will use all cores by default
+   verbose = FALSE#after every tree/stump it is going to show the error and how it is changing
+ )
> gd_test$pred.class <- predict(gbm.fit, gd_test, type = "response")
Using 5106 trees...
```

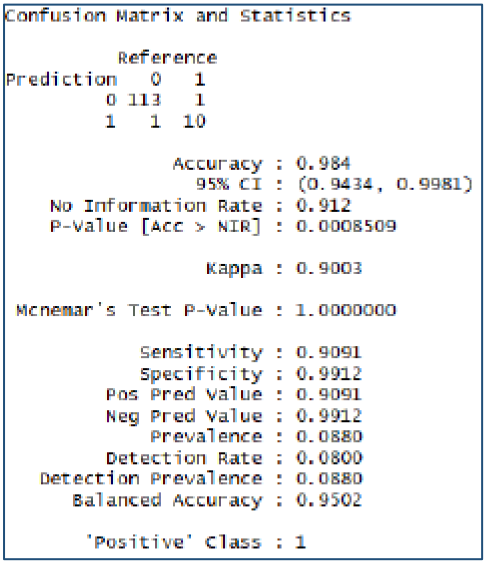


```
    FALSE TRUE
  0   113    1
  1     1   10
```

6.1.5 Performance Measures

6.1.5.1 Prediction -Confusion Matrix

Assessing the predictive ability of the Gradient Boosting model

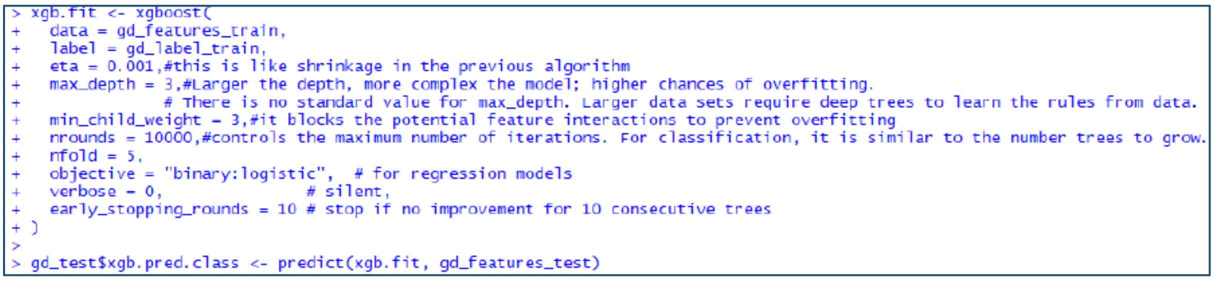


```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 113   1
         1   1  10
                                          
               Accuracy : 0.984           
                 95% CI : (0.9434, 0.9981)
    No Information Rate : 0.912           
    P-Value [Acc > NIR] : 0.0008509       
                                          
                  Kappa : 0.9003          
                                          
 Mcnemar's Test P-Value : 1.0000000       
                                          
            Sensitivity : 0.9091          
            Specificity : 0.9912          
         Pos Pred Value : 0.9091          
         Neg Pred Value : 0.9912          
             Prevalence : 0.0880          
         Detection Rate : 0.0800          
   Detection Prevalence : 0.0880          
      Balanced Accuracy : 0.9502          
                                          
       'Positive' Class : 1               
```

The above model with an accuracy of 98.4% is indeed a good quality model but lets check more method like Xboost for improvement of model.

6.1.6 Executing XGBoost Algorithm on data

Using binary logistic for regression models. It has various control parameter which restrict the iterations, overfitting chances, trees count etc.

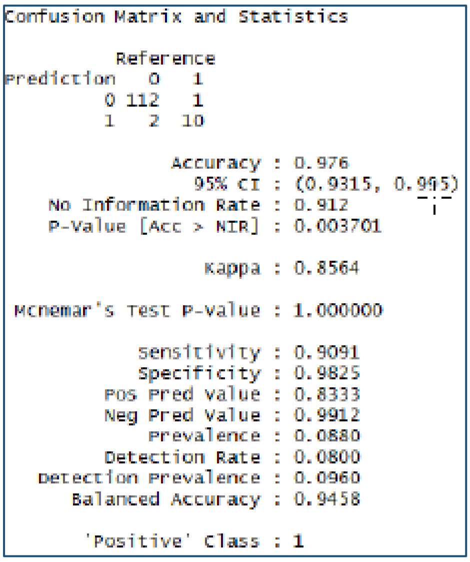


```
> xgb.fit <- xgboost(
+   data = gd_features_train,
+   label = gd_label_train,
+   eta = 0.001,#this is like shrinkage in the previous algorithm
+   max_depth = 3,#Larger the depth, more complex the model; higher chances of overfitting.
+             # There is no standard value for max_depth. Larger data sets require deep trees to learn the rules from data.
+   min_child_weight - 3,#it blocks the potential feature interactions to prevent overfitting
+   nrounds = 10000,#controls the maximum number of iterations. For classification, it is similar to the number trees to grow.
+   nfold = 5,
+   objective = "binary:logistic",  # for regression models
+   verbose - 0,              # silent,
+   early_stopping_rounds = 10 # stop if no improvement for 10 consecutive trees
+ )
>
> gd_test$xgb.pred.class <- predict(xgb.fit, gd_features_test)
```

6.1.7 Performance Measures on Boosting

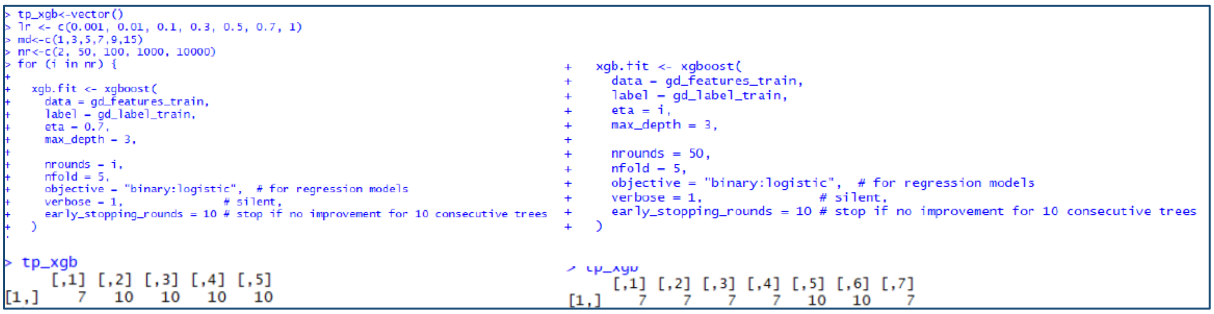
6.1.7..1 Prediction -Confusion Matrix

Assessing the predictive ability of the XGBoost model



```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 112   1
         1   2  10

               Accuracy : 0.976
                 95% CI : (0.9315, 0.995)
    No Information Rate : 0.912
    P-Value [Acc > NIR] : 0.003701

                  Kappa : 0.8564

 Mcnemar's Test P-Value : 1.000000

            Sensitivity : 0.9091
            Specificity : 0.9825
         Pos Pred Value : 0.8333
         Neg Pred Value : 0.9912
             Prevalence : 0.0880
         Detection Rate : 0.0800
   Detection Prevalence : 0.0960
      Balanced Accuracy : 0.9458

       'Positive' Class : 1
```

The above model with an accuracy of 97.6% is less than GBM lets check more method like Xboost for improvement of model In iterative steps , checking various value combination until we find the best fit Below is few example where values of shrinkage, eta , rounds varies. if employee using car and our prediction=0.5, we are going to display it with the next line compare the same algorithm for different values.

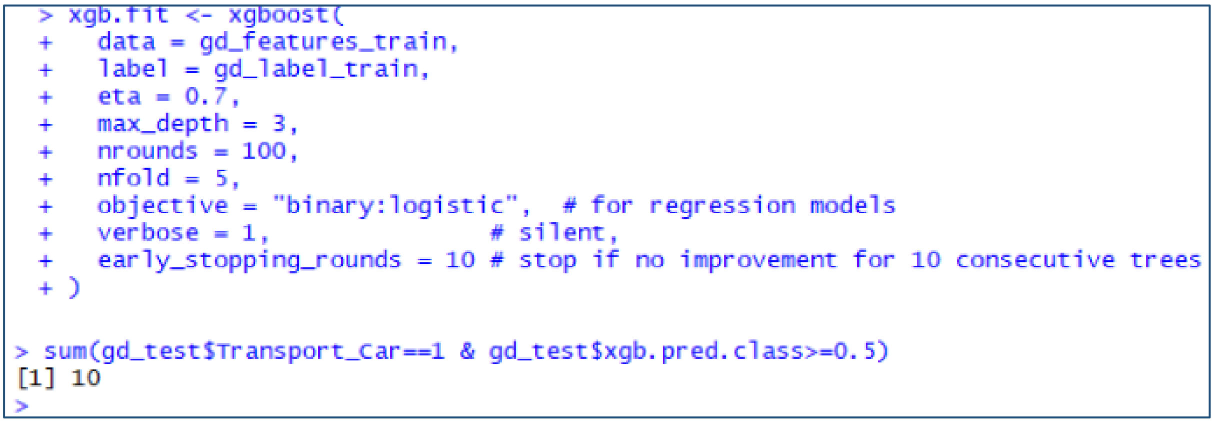


```
> tp_xgb<-vector()
> lr <- c(0.001, 0.01, 0.1, 0.3, 0.5, 0.7, 1)
> md<-c(1,3,5,7,9,15)
> nr<-c(2, 50, 100, 1000, 10000)
> for (i in nr) {
+
+   xgb.fit <- xgboost(
+     data = gd_features_train,
+     label - gd_label_train,
+     eta - 0.7,
+     max_depth - 3,
+
+     nrounds - i,
+     nfold = 5,
+     objective - "binary:logistic",  # for regression models
+     verbose - 1,               # silent,
+     early_stopping_rounds = 10 # stop if no improvement for 10 consecutive trees
+   )

> tp_xgb
     [,1] [,2] [,3] [,4] [,5]
[1,]    7   10   10   10   10
```

```
+   xgb.fit <- xgboost(
+     data - gd_features_train,
+     label - gd_label_train,
+     eta = i,
+     max_depth = 3,
+
+     nrounds = 50,
+     nfold - 5,
+     objective = "binary:logistic",  # for regression models
+     verbose = 1,               # silent,
+     early_stopping_rounds = 10 # stop if no improvement for 10 consecutive trees
+   )

> tp_xgb
     [,1] [,2] [,3] [,4] [,5] [,6] [,7]
[1,]    7    7    7    7   10   10    7
```

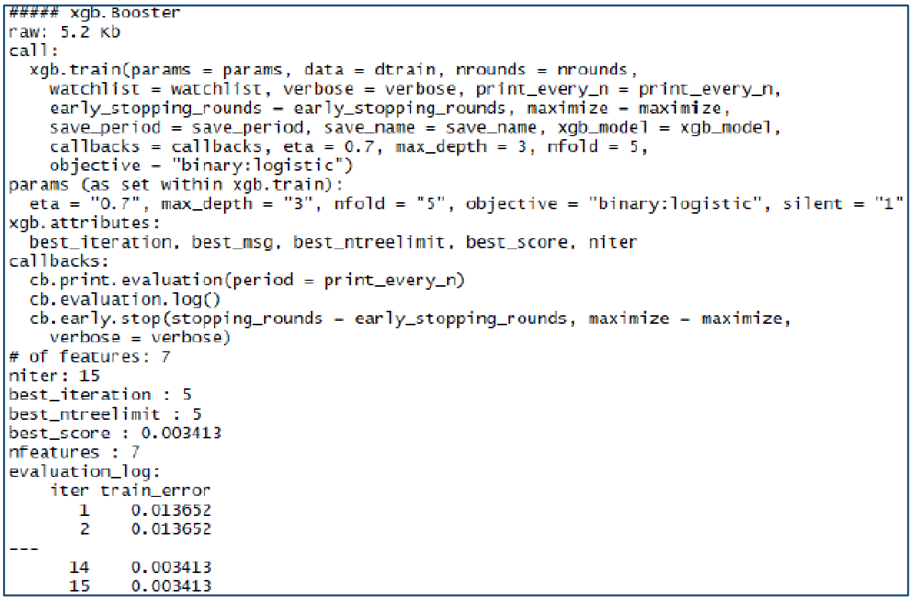
6.1.8 Best XGBoost Model and performance measures

Using binary logistic for regression models and best control parameter depicting best model

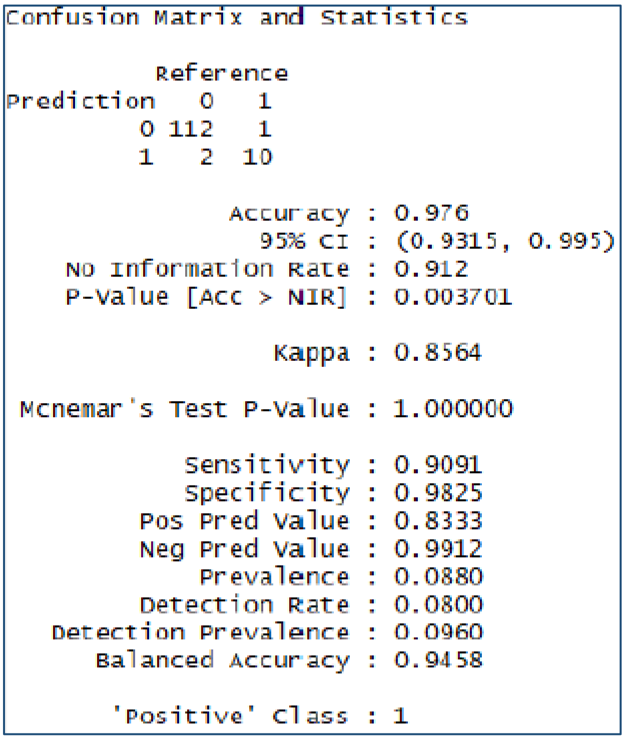


```
 > xgb.fit <- xgboost(
 +   data = gd_features_train,
 +   label = gd_label_train,
 +   eta = 0.7,
 +   max_depth = 3,
 +   nrounds = 100,
 +   nfold = 5,
 +   objective = "binary:logistic",  # for regression models
 +   verbose = 1,               # silent,
 +   early_stopping_rounds = 10 # stop if no improvement for 10 consecutive trees
 + )

> sum(gd_test$Transport_Car==1 & gd_test$xgb.pred.class>=0.5)
[1] 10
>
```

Assessing the predictive ability of the Best XGBoost model



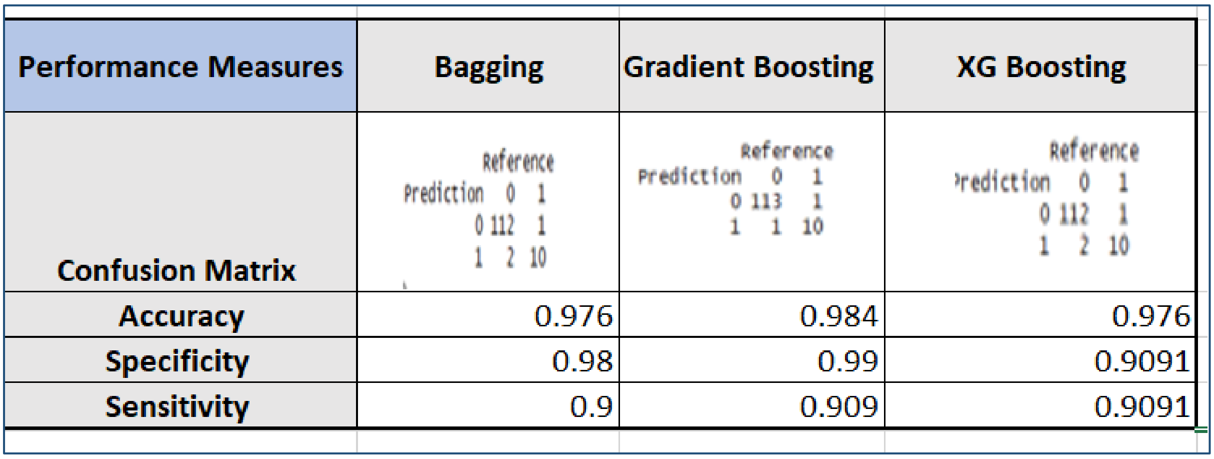
```
##### xgb.Booster
raw: 5.2 Kb
call:
  xgb.train(params = params, data = dtrain, nrounds = nrounds,
    watchlist = watchlist, verbose = verbose, print_every_n = print_every_n,
    early_stopping_rounds = early_stopping_rounds, maximize = maximize,
    save_period = save_period, save_name = save_name, xgb_model = xgb_model,
    callbacks = callbacks, eta = 0.7, max_depth = 3, nfold = 5,
    objective = "binary:logistic")
params (as set within xgb.train):
  eta = "0.7", max_depth = "3", nfold = "5", objective = "binary:logistic", silent = "1"
xgb.attributes:
  best_iteration, best_msg, best_ntreelimit, best_score, niter
callbacks:
  cb.print.evaluation(period = print_every_n)
  cb.evaluation.log()
  cb.early.stop(stopping_rounds = early_stopping_rounds, maximize = maximize,
    verbose = verbose)
# of features: 7
niter: 15
best_iteration : 5
best_ntreelimit : 5
best_score : 0.003413
nfeatures : 7
evaluation_log:
    iter train_error
       1    0.013652
       2    0.013652
---
      14    0.003413
      15    0.003413
```



```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 112   1
         1   2  10

               Accuracy : 0.976
                 95% CI : (0.9315, 0.995)
    No Information Rate : 0.912
    P-Value [Acc > NIR] : 0.003701

                  Kappa : 0.8564

 Mcnemar's Test P-Value : 1.000000

            Sensitivity : 0.9091
            Specificity : 0.9825
         Pos Pred Value : 0.8333
         Neg Pred Value : 0.9912
             Prevalence : 0.0880
         Detection Rate : 0.0800
   Detection Prevalence : 0.0960
      Balanced Accuracy : 0.9458

       'Positive' Class : 1
```

6.2 Compare Model Performance: Boosting and Bagging

To check the performance, there are following performance measures and its parameters which has been considered and compared for evaluation
Below are few considered for our problem –



Performance Measures	Bagging	Gradient Boosting	XG Boosting
Confusion Matrix	Reference Prediction 0 1 0 112 1 1 2 10	Reference Prediction 0 1 0 113 1 1 1 10	Reference rediction 0 1 0 112 1 1 2 10
Accuracy	0.976	0.984	0.976
Specificity	0.98	0.99	0.9091
Sensitivity	0.9	0.909	0.9091

Accuracy describes how often classifier are coorect so more the accuracy more often correct prediction done.

Sensitivity interprets the Yes ie how often does it predict Yes which is slightly more for Boosting.

Specificity interprets the No ie how often does it predict No which is more for Boosting ie Gradient Boosting.

Conclusion - As all 3 models performance measures are quite close but considering accuracy, Gradient boosting or Boosting stand out considering above performance measures considered.

7. Actionable Insight and recommendation

- The most important variables for predicting mode of transport from all 3 models are:
  - Age
  - Salary
  - Distance
- Company want to predict identify potential customer who are likely to use car as mode of transport ot commute office so they can focus on this group. This would help them to introduce policy on Car loan or car related reimbursement in company.
- Such policies are often used as retention policies.
- Model Performance values for Train and test are within the maximum tolerance deviation of +/- 10%. Hence, the all models are not over-fitting.
- Logistic Regression are providing better performance as compared to other models considering above 4 to 5 performance measures considered so either can be used.
- Company should introduce more features in data for better analysis and also suggest electric car and car pooling suggestions to focus group. It would increase the focus group volume.

- Electric car is good of environment but expensive currently so company can introduce various loan options along with some subsidiary as perk in order to be beneficial for environment and affordability.
- Focus on group of customers by sending mails/calls on new car policies, carpooling, car taxi options, to provide best results.
- Higher the age, higher the salary and higher the distance, the more probability of using a car.
- Majority of the people using car did not have an MBA degree.
- Many of the Engineers used car as their mode of commute.
- Young employees of the age group < 30 yrs and those with <6yrs experience and Salary range of 15L can be given loans so that they can afford a car.
- A reimbursement scheme with the partial costs of diesel/petrol should be implemented so that employees use car more from office to home and vice versa.
- The Female population who use cars is very low. Additional incentives can be provided for the women of the company in order to boost their morale and special loans can be given only to women.
- The traffic during peak hours may also be an issue due to which people either prefer 2-wheeler or public transport. Carpooling along with colleagues should be encouraged so that people around the vicinity use car instead of public transport.
- Also, strict action should be taken against individuals who drive without license as there is a small population who is doing that.
- The Public Transport system may not be easily accessible/widespread/well-connected or may be very time consuming for long distance. Hence, roads should be well maintained even if public transport is not very widespread/accessible to the people so that they prefer driving over public transport or 2wheeler.