Project 4 – Cars Case Study

Hassaan Pasha
hassaanepasha@gmail.com
Date: 18-04-2020

Contents

Problem 1

Cars Case Study

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

Answer: Initial Steps

Importing Libraries

```
library(readr)
library(dplyr)
library(ggplot2)
library(gridExtra)
library(corrplot)
library(caTools)
library(DMwR)
library(caret)
library(car)
library(class)
library(e1071)
library(ipred)
library(rpart)
library(gbm)
```

Importing Data

```
## set working directory
setwd("D:/PGP-DSBA/4 - Predictive/Week5")
cars = read.csv("Cars-dataset.csv", header=TRUE)
dim(cars)

## [1] 418   9
```

The dataset has 418 row and 9 columns

Structure of Data

```
str(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 ...
```

Engineer, MBA and License needs to be converted to factor

```
cars$Engineer = as.factor(as.character(cars$Engineer))
cars$MBA = as.factor(as.character(cars$MBA))
cars$license = as.factor(as.character(cars$license))
str(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 : Factor w/ 2 levels "0","1": 2 2 2 1 1 1 2 1 2 2 ...
##  $ MBA      : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 2 1 1 1 ...
##  $ 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  : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 2 ...
## $ Transport: Factor w/ 3 levels "2Wheeler","Car",..: 1 1 1 1 1 1 1 1 1 1 ...
```

Summary of Data

```
summary(cars)
```

```
##       Age          Gender     Engineer   MBA         Work.Exp
## Min.   :18.00   Female:121   0:105    0   :308   Min.   : 0.000
## 1st Qu.:25.00   Male  :297   1:313    1   :109   1st Qu.: 3.000
## Median :27.00                         NA's:  1   Median : 5.000
## Mean   :27.33                                    Mean   : 5.873
## 3rd Qu.:29.00                                    3rd Qu.: 8.000
## Max.   :43.00                                    Max.   :24.000
##     Salary          Distance      license             Transport
## Min.   : 6.500   Min.   : 3.20   0:333   2Wheeler         : 83
## 1st Qu.: 9.625   1st Qu.: 8.60   1: 85   Car              : 35
## Median :13.000   Median :10.90           Public Transport:300
## Mean   :15.418   Mean   :11.29
## 3rd Qu.:14.900   3rd Qu.:13.57
## Max.   :57.000   Max.   :23.40
```

Labeling Engineer, MBA and Licenses to be boolean

```
cars$Engineer = factor(cars$Engineer, labels = c("Non-Engineer","Engineer"))
cars$MBA = factor(cars$MBA, labels = c("Non-MBA","MBA"))
cars$license = factor(cars$license, labels = c("No-License","License"))
summary(cars)
```

```
##       Age          Gender            Engineer         MBA          Work.Exp
## Min.   :18.00   Female:121   Non-Engineer:105   Non-MBA:308   Min.   : 0.000
## 1st Qu.:25.00   Male  :297   Engineer    :313   MBA    :109   1st Qu.: 3.000
## Median :27.00                                   NA's   :  1   Median : 5.000
## Mean   :27.33                                                 Mean   : 5.873
## 3rd Qu.:29.00                                                 3rd Qu.: 8.000
## Max.   :43.00                                                 Max.   :24.000
##     Salary          Distance          license               Transport
## Min.   : 6.500   Min.   : 3.20   No-License:333   2Wheeler         : 83
## 1st Qu.: 9.625   1st Qu.: 8.60   License   : 85   Car              : 35
## Median :13.000   Median :10.90                    Public Transport:300
## Mean   :15.418   Mean   :11.29
## 3rd Qu.:14.900   3rd Qu.:13.57
## Max.   :57.000   Max.   :23.40
```

Cleansing of data by treating the missing values

```
sum(is.na(cars))

## [1] 1

cars[is.na(cars)] = "Non-MBA"
summary(cars)

##       Age            Gender             Engineer          MBA          Work.Exp
##  Min.   :18.00   Female:121   Non-Engineer:105   Non-MBA:309   Min.   : 0.000
##  1st Qu.:25.00   Male  :297   Engineer    :313   MBA    :109   1st Qu.: 3.000
##  Median :27.00                                                 Median : 5.000
##  Mean   :27.33                                                 Mean   : 5.873
##  3rd Qu.:29.00                                                 3rd Qu.: 8.000
##  Max.   :43.00                                                 Max.   :24.000
##      Salary           Distance          license                  Transport
##  Min.   : 6.500   Min.   : 3.20   No-License:333   2Wheeler        : 83
##  1st Qu.: 9.625   1st Qu.: 8.60   License   : 85   Car             : 35
##  Median :13.000   Median :10.90                    Public Transport:300
##  Mean   :15.418   Mean   :11.29
##  3rd Qu.:14.900   3rd Qu.:13.57
##  Max.   :57.000   Max.   :23.40
```

I have observed that only one missing value in the MBA column. I have imputed the missing value by the majority class that is Non-MBA (0).

Value of Transport into Boolean

```
cars$Transport = ifelse(cars$Transport =="Car",1,0)
cars$Transport = as.factor(cars$Transport)
cars$Transport = factor(cars$Transport, labels = c("No","Yes"))
table(cars$Transport)

##
##  No Yes
## 383  35
```

The column transport has 3 categories i.e. 2Wheeler, Public Transport and Car. Since we need to focus on the prediction of Cars only, I will make the value for Car as 1 (Yes) and other as 0 (no).

Response Rate

```
response_rate <- prop.table(table(cars$Transport))
response_rate

##
##         No        Yes
## 0.91626794 0.08373206
```

Only 8.3% of the required is the Target Variable. This is a very low value.

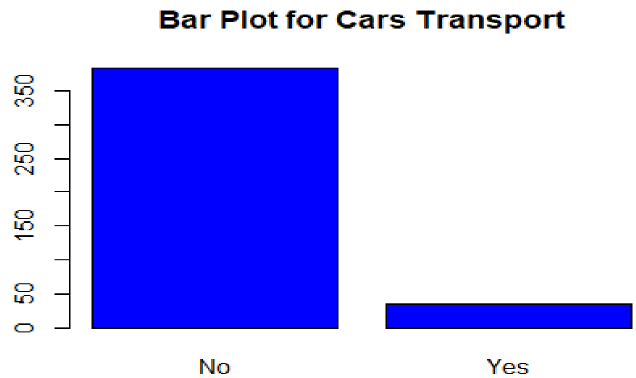
Univariate Analysis

Transport

```
summary(cars$Transport)

##   No Yes
## 383  35

plot(cars$Transport, col="blue", main = "Bar Plot for Cars Transport")
```



As the Bar plot suggests that majority of the dataset do not use car.

Age

```
summary(cars$Age)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   18.00   25.00   27.00   27.33   29.00   43.00

par(mfrow=c(1,2))
hist(cars$Age,col='red', main='Age of All Employees',xlab='Age')
boxplot(cars$Age, col = 'Pink', main = 'Box plot of Age')
```

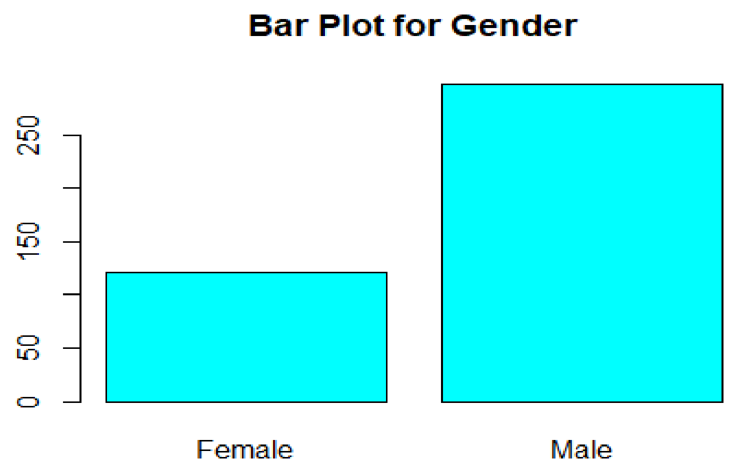
The dataset contains data for all the age groups in the company. The maximum age is 43 and so all the outliers seem to be real values and therefore need not be omitted.

Gender

```
summary(cars$Gender)

## Female   Male
##    121    297

plot(cars$Gender, col="cyan", main = "Bar Plot for Gender")
```



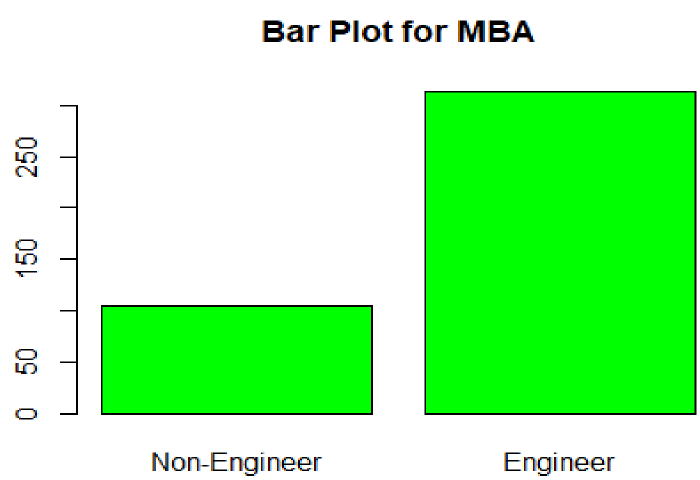
The male to female ratio is 2.45:1

Engineer

```
summary(cars$Engineer)

## Non-Engineer     Engineer
##          105          313

plot(cars$Engineer, col="green", main = "Bar Plot for MBA")
```



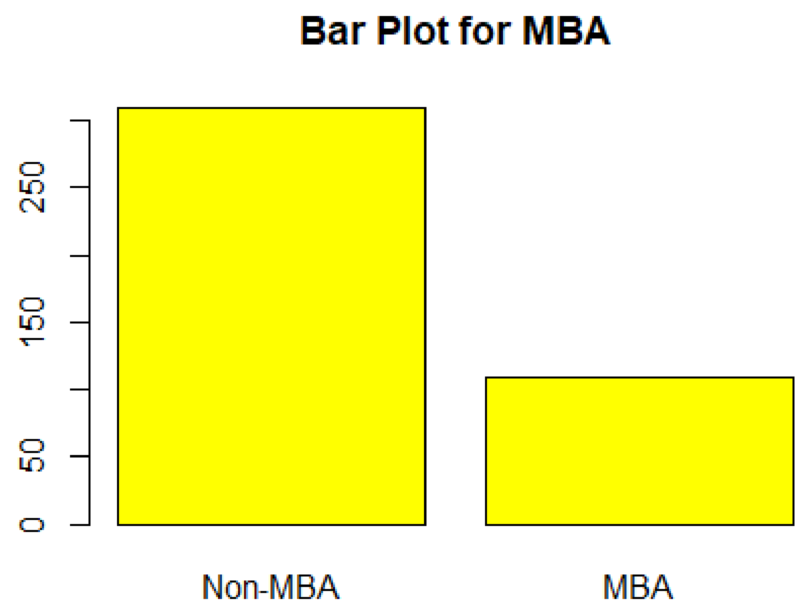
The ratio of Engineer to Non- Engineer is 2.98:1.

MBA

```r
summary(cars$MBA)

## Non-MBA     MBA
##     309     109

plot(cars$MBA, col="yellow", main = "Bar Plot for MBA")
```



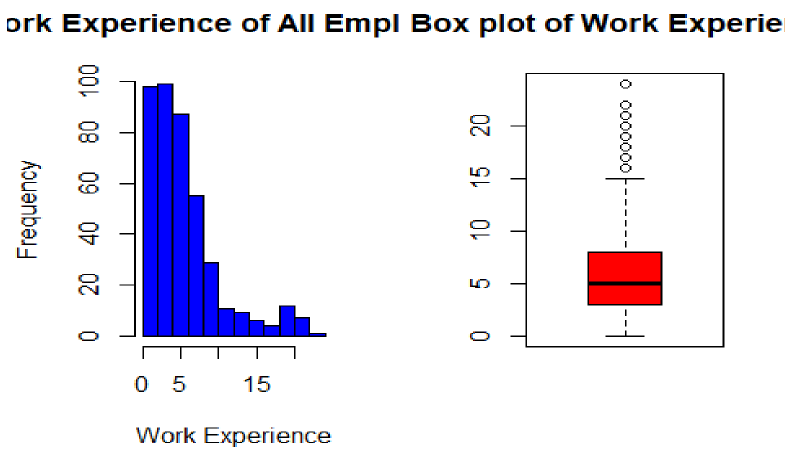
The MBA to Non MBA ratio is 1:2.83

Work Experience

```
summary(cars$Work.Exp)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   0.000   3.000   5.000   5.873   8.000  24.000

par(mfrow=c(1,2))
hist(cars$Work.Exp,col='Blue', main = 'Work Experience of All Employees',xlab='Work E
xperience')
boxplot(cars$Work.Exp, col = 'Red', main = 'Box plot of Work Experience')
```



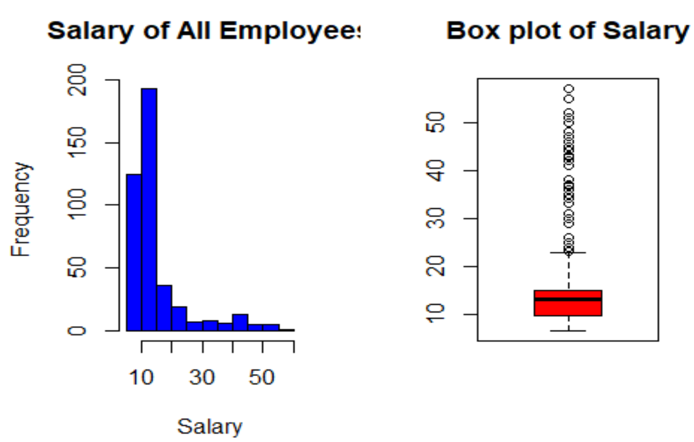
The maximum experience is 24 years which is acceptable. This is skewed towards right, there would be more juniors then seniors in any firm

Salary

```
summary(cars$Salary)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   6.500   9.625  13.000  15.418  14.900  57.000

par(mfrow=c(1,2))
hist(cars$Salary,col='blue', main='Salary of All Employees',xlab='Salary')
boxplot(cars$Salary, col = 'red', main = 'Box plot of Salary')
```



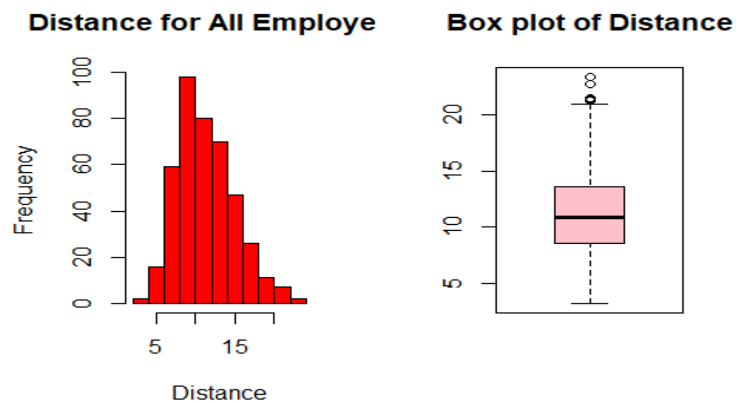
None of the values seem to be wrong or typos. Therefore the outliers do not need to be treated.

Distance

```
summary(cars$Distance)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    3.20    8.60   10.90   11.29   13.57   23.40

par(mfrow=c(1,2))
hist(cars$Distance,col='red', main='Distance for All Employees',xlab='Distance')
boxplot(cars$Distance, col = 'Pink', main = 'Box plot of Distance')
```



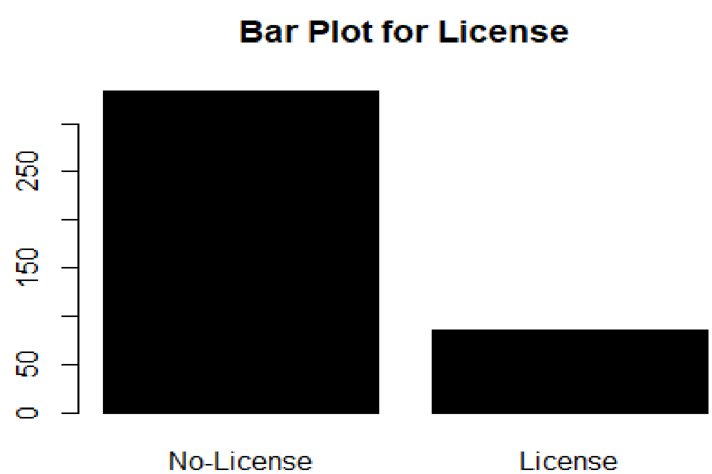
None of the values seem to be wrong or typos. Therefore the outliers do not need to be treated

License

```
summary(cars$license)

## No-License    License
##        333         85

plot(cars$license, col="black", main = "Bar Plot for License")
```



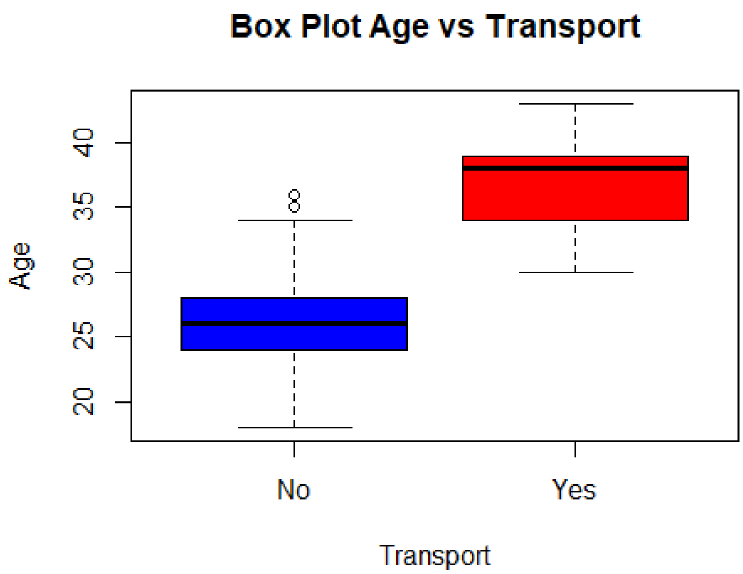
The licensed to no-license ratio is 1:3.97

Bivariate Analysis

Let us now do bivariate analysis with respect to the target variable Transport

Age vs Transport

```
boxplot(Age~Transport,data=cars,horizontal=FALSE,col=c('blue','red'),main='Box Plot A
ge vs Transport')
```



This can be observed that usually employees aged above 30 are using cars. This can be due to the fact that young employees cannot afford cars or are more health conscious and prefer transport other than cars.

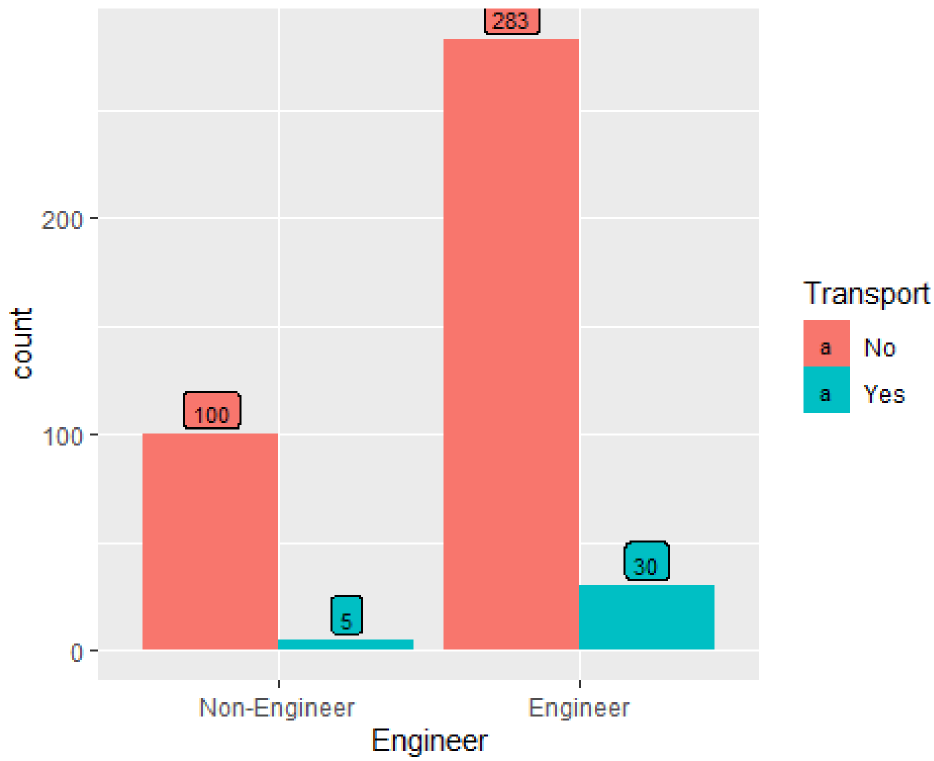
Gender vs Transport

```
p1 = ggplot(cars, aes(Gender,fill= Transport)) + geom_bar(stat = "count", position =
"dodge") + geom_label(stat = "count", aes(label= ..count..),
             size = 3, position = position_dodge(width = 0.9), vjust=-0.15)
grid.arrange(p1)
```

The proportion of Males and Females using cars is significantly less than those who do not travel by car. Also the Males using cars are more the female using cars.

Engineer vs Transport

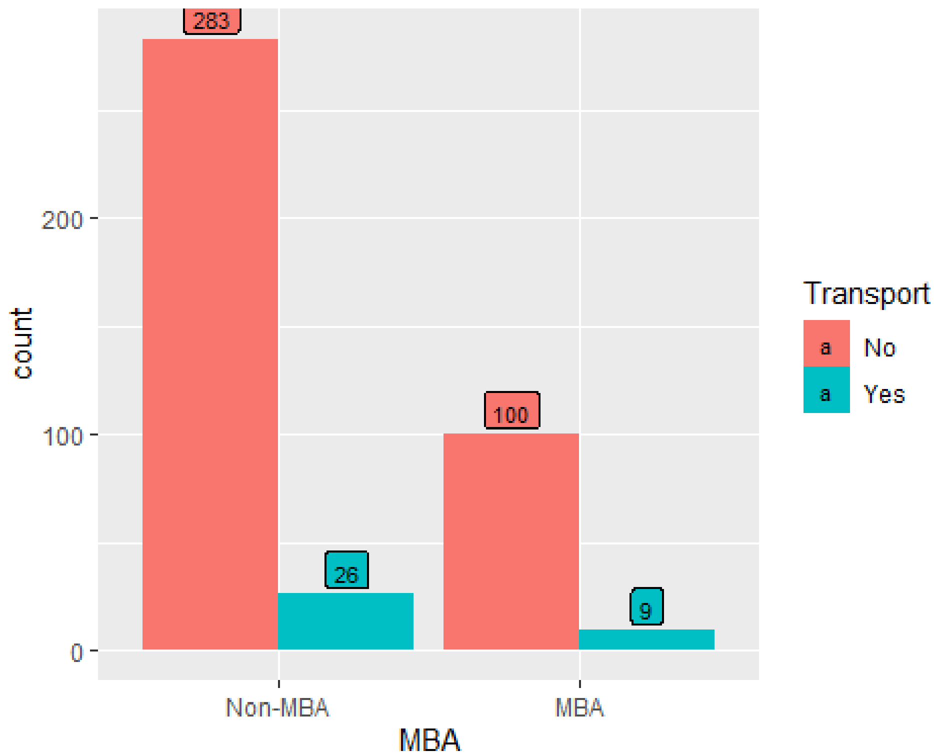
```
p2 = ggplot(cars, aes(Engineer,fill= Transport)) + geom_bar(stat = "count", position
= "dodge") + geom_label(stat = "count", aes(label= ..count..),
              size = 3, position = position_dodge(width = 0.9), vjust=-0.15)
grid.arrange(p2)
```



Majority of the engineers and non- engineer are not travelling by Car.

MBA vs Transport

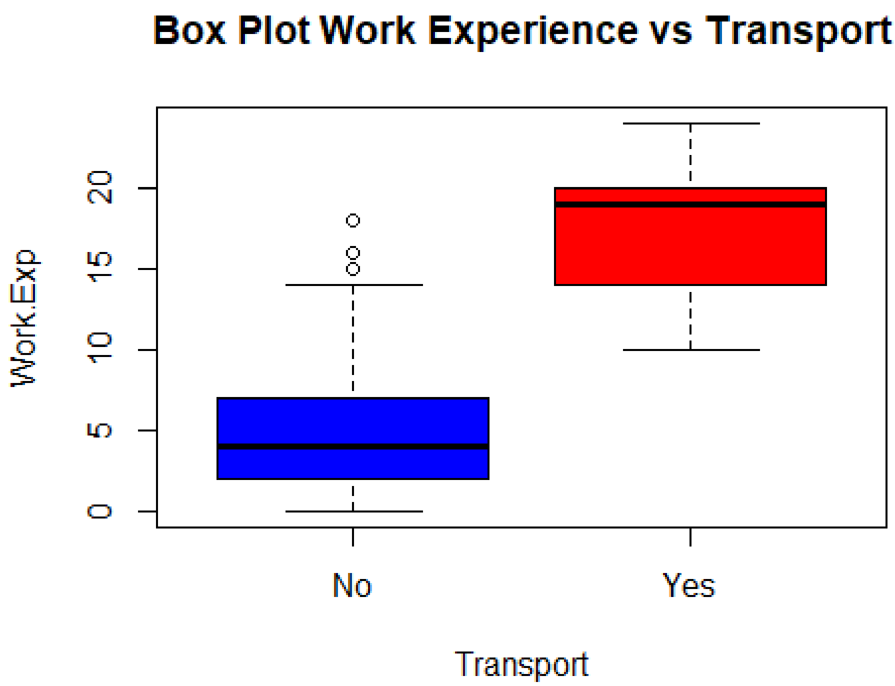
```
p3 = ggplot(cars, aes(MBA,fill= Transport)) + geom_bar(stat = "count", position = "do
dge") + geom_label(stat = "count", aes(label= ..count..),
                size = 3, position = position_dodge(width = 0.9), vjust=-0.15)
grid.arrange(p3)
```



Majority of MBA and Non-MBA do not travel by car. Also, Non-MBA that travel by car are more than who are MBAs.

Work Experience vs Transport

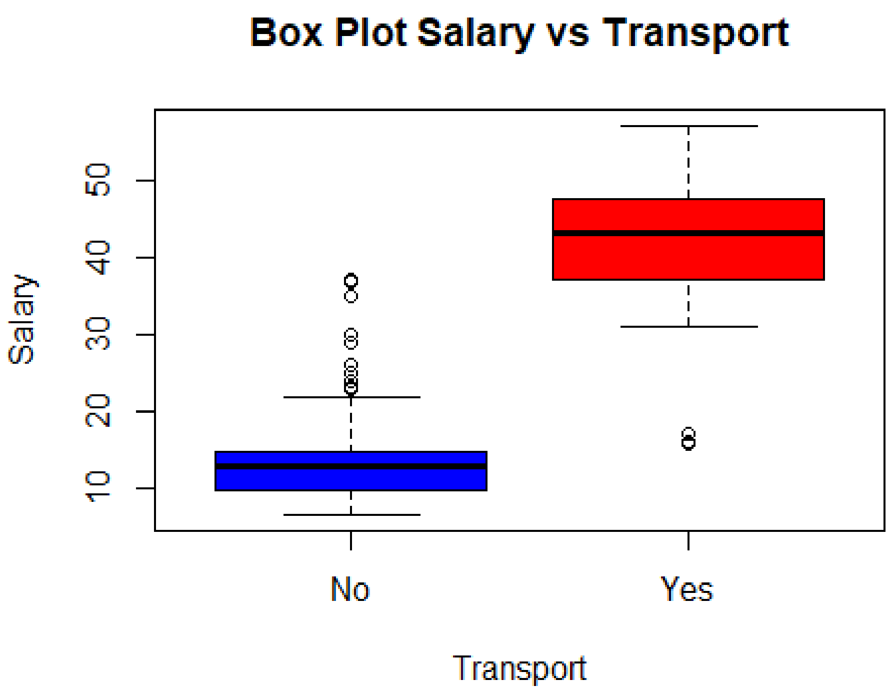
```
boxplot(Work.Exp~Transport,data=cars,horizontal=FALSE,col=c('blue','red'),main='Box P
lot Work Experience vs Transport')
```



It is observed that people with high experience are the ones who use cars. This also indicates that these people with High salaries as well as it is a similar box plot and can afford a car.

Salary vs Transport

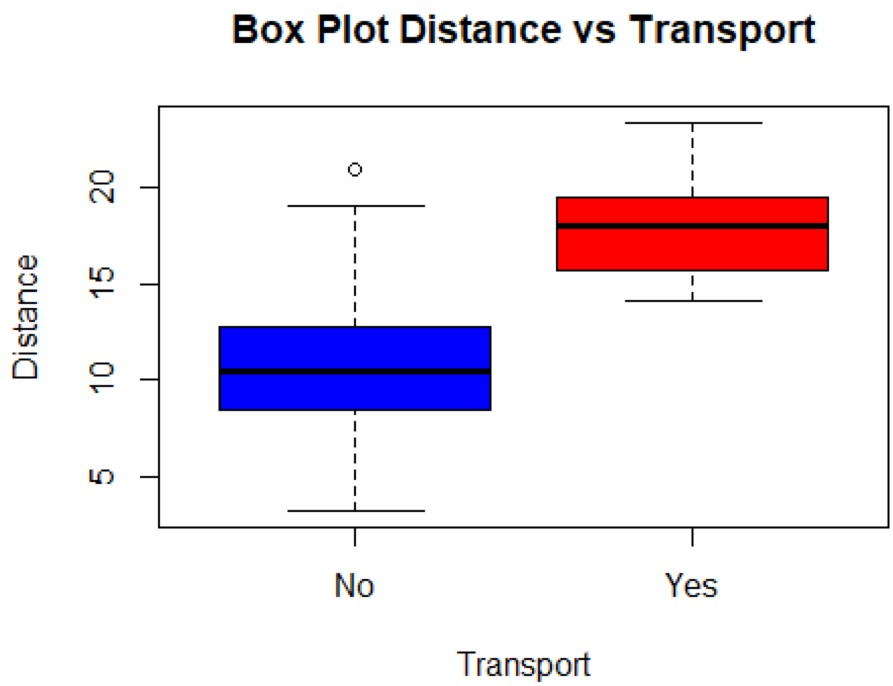
```
boxplot(Salary~Transport,data=cars,horizontal=FALSE,col=c('blue','red'),main='Box Plo
t Salary vs Transport')
```



This indicates that people with high salaries travel by car, and maybe can afford a car. It is a similar box plot to work experience.

Distance vs Transport

```
boxplot(Distance~Transport,data=cars,horizontal=FALSE,col=c('blue','red'),main='Box Plot Distance vs Transport')
```



It can be observed that people who live far from the office opt for car to use.

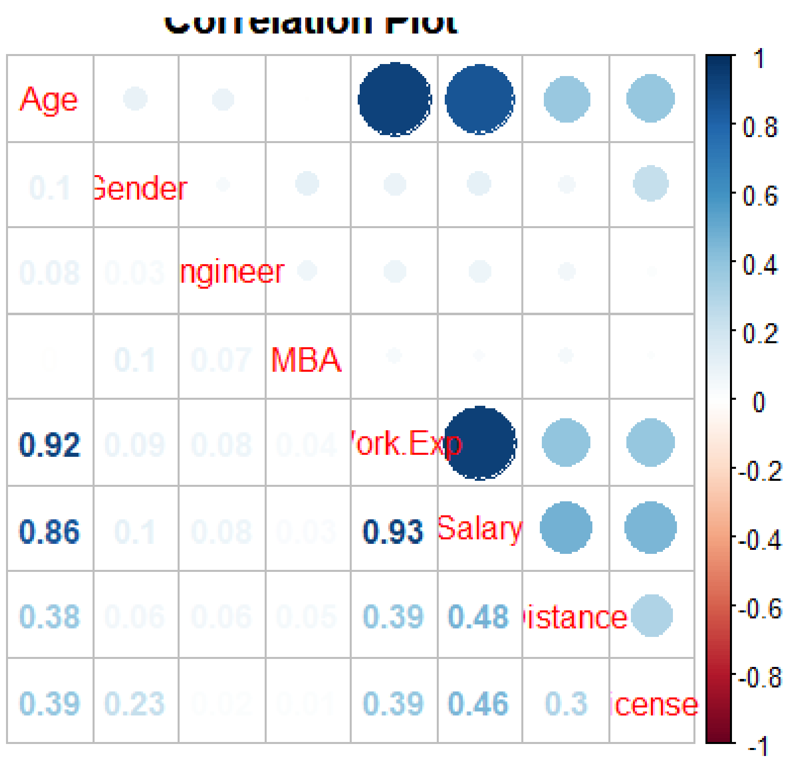
License vs Transport

```
p4 = ggplot(cars, aes(license,fill= Transport)) + geom_bar(stat = "count", position =
 "dodge") + geom_label(stat = "count", aes(label= ..count..),
             size = 3, position = position_dodge(width = 0.9), vjust=-0.15)
grid.arrange(p4)
```

It can be observe that Majority of the people are not licensed and therefore cannot drive. However, there are 6 people who do not have license and still travel by car, this means that either their family members pick and drop them in car or maybe they are licensed to drive 2 wheelers. One more thing to note here is that 56 licensed people still do not use cars.

Multicollinearity

```
cars.num = cars[-9]
cars.num = sapply(cars.num, as.numeric)
cars.cor = cor(cars.num)
corrplot.mixed(cars.cor, main = "Correlation Plot")
```



- It can be seen that Age, Work Experience and Salary have very high correlations.
- Distance and license have weaker correlations.
- Gender, Engineer and MBA have no correlations.

Data Preparation

Splitting into train and test data

```
set.seed(1000) #Input any random number
x = sample.split(cars$Transport, SplitRatio = 0.7)
cars_train = subset(cars, x == TRUE)
dim(cars_train)

## [1] 292   9

cars_test = subset(cars, x == FALSE)
dim(cars_test)

## [1] 126   9

rr.train = sum(cars_train$Transport == "Yes")/nrow(cars_train)
rr.train

## [1] 0.08219178

rr.test = sum(cars_test$Transport == "Yes")/nrow(cars_test)
rr.test

## [1] 0.08730159
```

I am splitting the data using a 70:30 split ratio. Since the response rate is the same as the main dataset we need to balance out the data for further analysis.

SMOTE for balancing data

```
table(cars_train$Transport)

##
##  No Yes
## 268  24

smote.cars_train = SMOTE(Transport ~., data = cars_train, perc.over = 3500, perc.unde
r = 500)
nrow(smote.cars_train)

## [1] 5064

table(smote.cars_train$Transport)

##
##   No  Yes
## 4200  864

prop.table(table(smote.cars_train$Transport))

##
##        No       Yes
## 0.8293839 0.1706161
```

After balancing the dataset with SMOTE we have at least a better response rate that we can analyze.

Logistic Regression

Let us now apply Logistic regression and train the model considering all the variables.

Model 1

```
lg.train = smote.cars_train
lg.test = cars_test
lgmodel1 = glm(Transport ~ ., data = lg.train, family = binomial(link="logit"))

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

summary(lgmodel1)

##
## Call:
## glm(formula = Transport ~ ., family = binomial(link = "logit"),
##     data = lg.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.5991   0.0000   0.0000   0.0000   2.8817
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -358.2972   157.0179  -2.282  0.02250 *
## Age                 -10.6132     4.6436  -2.286  0.02228 *
## GenderMale          -11.8646     4.0546  -2.926  0.00343 **
## EngineerEngineer     -0.4802     1.2312  -0.390  0.69653
## MBAMBA               -8.3070     3.0500  -2.724  0.00646 **
## Work.Exp             44.2513    19.1767   2.308  0.02102 *
## Salary               -4.6020     2.0607  -2.233  0.02554 *
## Distance             22.8910     9.9375   2.303  0.02125 *
## licenseLicense        8.2206     2.6978   3.047  0.00231 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4627.096  on 5063  degrees of freedom
## Residual deviance:   63.894  on 5055  degrees of freedom
## AIC: 81.894
##
## Number of Fisher Scoring iterations: 19
```

We can identify the significant variables to be Gender, Age, MBA, and License in the dataset. AIC is 81. It's good but not up to the mark.

```
vif(lgmodel1)

##          Age       Gender     Engineer          MBA     Work.Exp       Salary
##   828.536295    32.936202     2.629495    14.406054 26603.270461  1932.339151
##     Distance      license
## 10635.246583    12.944519
```

The vif of Age, Work, Salary and Distance is high. We shall remove one variable at a time to and run the model again. We shall remove Work Experience as it has the highest VIF.

Model 2

```
lg2.train = lg.train[,-5]

lgmodel2 = glm(Transport ~ ., data = lg2.train, family = binomial(link="logit"))

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

summary(lgmodel2)

##
## Call:
## glm(formula = Transport ~ ., family = binomial(link = "logit"),
##     data = lg2.train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.18352  -0.00008   0.00000   0.00000   3.09043
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -118.90688   16.11370  -7.379 1.59e-13 ***
## Age                 2.69455    0.38734   6.957 3.49e-12 ***
## GenderMale         -2.37250    0.51374  -4.618 3.87e-06 ***
## EngineerEngineer   -0.33540    0.57205  -0.586    0.558
## MBAMBA             -4.22897    0.71669  -5.901 3.62e-09 ***
## Salary              0.24092    0.04074   5.914 3.34e-09 ***
## Distance            2.01383    0.26347   7.644 2.11e-14 ***
## licenseLicense      4.03985    0.65837   6.136 8.46e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4627.10  on 5063  degrees of freedom
## Residual deviance:  182.23  on 5056  degrees of freedom
## AIC: 198.23
##
## Number of Fisher Scoring iterations: 12
```

Model 2 describes all the variables to be highly significant.

```
vif(lgmodel2)

##       Age    Gender  Engineer       MBA    Salary  Distance   license
## 14.622031  1.670764  1.452381  2.921871  3.455746 15.762365  3.108438
```

Age and Distance is VIF is higher than 5 which means we have to remove one more variable. We shall remove distance now as it has the highest VIF value.

Model 3

```
lg3.train = lg2.train[,-6]

lgmodel3 = glm(Transport ~ ., data = lg3.train, family = binomial(link="logit"))
summary(lgmodel3)

## 
## Call:
## glm(formula = Transport ~ ., family = binomial(link = "logit"), 
##     data = lg3.train)
## 
## Deviance Residuals: 
##      Min        1Q    Median        3Q       Max  
## -2.75628  -0.10463  -0.03237  -0.00640   2.25053  
## 
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)    
## (Intercept)      -29.99358    1.71060 -17.534  < 2e-16 ***
## Age                0.84772    0.05729  14.797  < 2e-16 ***
## GenderMale        -1.69166    0.22057  -7.670 1.73e-14 ***
## EngineerEngineer   0.01425    0.24121   0.059    0.953    
## MBAMBA            -0.07199    0.20460  -0.352    0.725    
## Salary             0.12988    0.01371   9.475  < 2e-16 ***
## licenseLicense     1.90392    0.21254   8.958  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 4627.10  on 5063  degrees of freedom
## Residual deviance:  765.16  on 5057  degrees of freedom
## AIC: 779.16
## 
## Number of Fisher Scoring iterations: 8

vif(lgmodel3)

##      Age   Gender Engineer      MBA   Salary  license 
## 1.208652 1.196677 1.030478 1.033076 1.256641 1.185462
```

Since all the VIF values are under 5 now. We can use this model to predict on test data.

Prediction on Test Data

```
lg.test$prd_lg3 = predict(lgmodel3, lg.test[1:8],type="response")
lg.test$class_lg3 = floor(lg.test$prd_lg3 +0.5)

# convert to factor
lg.test$class_lg3 = factor(lg.test$class_lg3, labels = c("No","Yes"))
confusionMatrix(lg.test$class_lg3, lg.test$Transport)

## Confusion Matrix and Statistics
## 
##           Reference
## Prediction  No Yes
##        No  109   0
```

```
##        Yes   6  11
##
##                Accuracy : 0.9524
##                  95% CI : (0.8992, 0.9823)
##     No Information Rate : 0.9127
##     P-Value [Acc > NIR] : 0.06955
##
##                   Kappa : 0.7603
##
##  Mcnemar's Test P-Value : 0.04123
##
##             Sensitivity : 0.9478
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.6471
##              Prevalence : 0.9127
##          Detection Rate : 0.8651
##    Detection Prevalence : 0.8651
##       Balanced Accuracy : 0.9739
##
##        'Positive' Class : No
##
```

Accuracy of 95.2% with Sensitivity of 94.8% and Specificity of 100%. This seems to be a good model.

KNN Model

Let us now apply KNN Model and train the model considering all the variables.

```
### Creating the Train and Test Data
knn.traindata = smote.cars_train
knn.testdata = cars_test

### Training Data
knn.traindata$Age = as.numeric(knn.traindata$Age)
knn.traindata$Gender = as.numeric(knn.traindata$Gender)
knn.traindata$Engineer = as.numeric(knn.traindata$Engineer)
knn.traindata$MBA = as.numeric(knn.traindata$MBA)
knn.traindata$Work.Exp = as.numeric(knn.traindata$Work.Exp)
knn.traindata$license = as.numeric(knn.traindata$license)
knn.traindata$Transport = as.numeric(knn.traindata$Transport)
str(knn.traindata)
```

```
## 'data.frame':    5064 obs. of  9 variables:
##  $ Age      : num  23 30 27 24 28 24 26 25 32 28 ...
##  $ Gender   : num  2 2 1 2 1 2 2 2 1 2 ...
##  $ Engineer : num  1 2 2 2 2 2 2 2 2 2 ...
##  $ MBA      : num  1 2 1 2 1 2 1 1 2 1 ...
##  $ Work.Exp : num  0 8 8 0 5 6 2 3 9 5 ...
##  $ Salary   : num  6.5 14.6 24.9 7.7 14.6 11.6 10 10.6 15.9 14.4 ...
##  $ Distance : num  7.3 10.6 13 11.3 9 11.3 16.4 8.1 16.6 5.1 ...
##  $ license  : num  1 1 1 2 1 2 2 1 1 1 ...
##  $ Transport: num  1 1 1 1 1 1 1 1 1 1 ...
```

```
### Test Data
knn.testdata$Age = as.numeric(knn.testdata$Age)
knn.testdata$Gender = as.numeric(knn.testdata$Gender)
knn.testdata$Engineer = as.numeric(knn.testdata$Engineer)
knn.testdata$MBA = as.numeric(knn.testdata$MBA)
knn.testdata$Work.Exp = as.numeric(knn.testdata$Work.Exp)
knn.testdata$license = as.numeric(knn.testdata$license)
knn.testdata$Transport = as.numeric(knn.testdata$Transport)
str(knn.testdata)

## 'data.frame':    126 obs. of  9 variables:
##  $ Age      : num  24 23 21 24 27 25 28 25 23 27 ...
##  $ Gender   : num  2 2 2 2 2 1 2 1 2 1 ...
##  $ Engineer : num  2 2 1 2 1 2 2 2 2 2 ...
##  $ MBA      : num  1 2 2 1 2 1 2 2 1 1 ...
##  $ Work.Exp : num  6 3 3 6 8 6 5 1 0 5 ...
##  $ Salary   : num  10.6 11.7 10.6 12.7 15.6 11.6 14.8 8.6 6.9 12.8 ...
##  $ Distance : num  6.1 7.2 7.7 8.7 9 10.1 10.8 11.2 11.7 11.8 ...
##  $ license  : num  1 1 1 1 1 1 2 1 1 1 ...
##  $ Transport: num  1 1 1 1 1 1 1 1 1 1 ...
```

Train KNN Model

```
cars.trainlabel = knn.traindata[,9]
cars.testlabel = knn.testdata[,9]

knn.traindata= knn.traindata[,-9]
knn.testdata= knn.testdata[,-9]

cars.testlabel.pred = knn(train = knn.traindata, test = knn.testdata, cl = cars.train
label, k =3)
```

Evaluate KNN Model with Confusion Matrix

```
# Convert to Factor Type
cars.testlabel = as.factor(cars.testlabel)
cars.testlabel = factor(cars.testlabel, labels = c ("No","Yes"))
cars.testlabel.pred = factor(cars.testlabel.pred, labels = c ("No","Yes"))
# Confusion Matrix
confusionMatrix(cars.testlabel.pred,cars.testlabel)

## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  113   0
##        Yes   2  11
##
##                Accuracy : 0.9841
##                  95% CI : (0.9438, 0.9981)
##     No Information Rate : 0.9127
##     P-Value [Acc > NIR] : 0.0008535
##
##                   Kappa : 0.908
##
##  Mcnemar's Test P-Value : 0.4795001
##
```

```
##            Sensitivity : 0.9826
##            Specificity : 1.0000
##         Pos Pred Value : 1.0000
##         Neg Pred Value : 0.8462
##             Prevalence : 0.9127
##         Detection Rate : 0.8968
##   Detection Prevalence : 0.8968
##      Balanced Accuracy : 0.9913
##
##       'Positive' Class : No
##
```

Accuracy of 98.4% with Sensitivity of 98.3% and Specificity of 100%. This seems to be a better model than logistic regression.

Naive Bayes Model

Let us now apply Naïve Bayes Model and train the model considering all the variables.

```
### Creating the Train and Test Data
nb.cars.train = smote.cars_train
nb.cars.test = cars_test
```

Train the Naive Bayes Model

```
nb_cars<-naiveBayes(x=nb.cars.train[,1:8], y=as.factor(nb.cars.train[,9]))

pred_nb<-predict(nb_cars,newdata = nb.cars.test[,1:8])
```

Evaluate Naive Bayes Model with Confusion Matrix

```
confusionMatrix(pred_nb,nb.cars.test[,9])

## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  108   0
##        Yes   7  11
##
##                Accuracy : 0.9444
##                  95% CI : (0.8889, 0.9774)
##     No Information Rate : 0.9127
##     P-Value [Acc > NIR] : 0.13158
##
##                   Kappa : 0.7293
##
##  Mcnemar's Test P-Value : 0.02334
##
##             Sensitivity : 0.9391
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.6111
##              Prevalence : 0.9127
##          Detection Rate : 0.8571
##    Detection Prevalence : 0.8571
```

```
##       Balanced Accuracy : 0.9696
##
##        'Positive' Class : No
##
```

Accuracy of 94.4% with Sensitivity of 93.9% and Specificity of 100%. This seems to be a good model but not as good as KNN model

Bagging Model

Let us now apply Bagging Techniques and train the model considering all the variables.

```
### Creating the Train and Test Data
bag.cars.train = smote.cars_train
bag.cars.test = cars_test
```

Train the Bagging Model

```
cars.bagging = bagging(Transport ~ ., data = bag.cars.train, control = rpart.control(
maxdepth=5, minsplit=4), coob =TRUE)
cars.bagging

##
## Bagging classification trees with 25 bootstrap replications
##
## Call: bagging.data.frame(formula = Transport ~ ., data = bag.cars.train,
##     control = rpart.control(maxdepth = 5, minsplit = 4), coob = TRUE)
##
## Out-of-bag estimate of misclassification error:  0.0018
```

Evaluate Bagging Model with Confusion Matrix

```
bag.cars.test$pred_bag = predict(cars.bagging,bag.cars.test)

confusionMatrix(bag.cars.test$pred_bag,bag.cars.test$Transport)

## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  112   0
##        Yes   3  11
##
##                Accuracy : 0.9762
##                  95% CI : (0.932, 0.9951)
##     No Information Rate : 0.9127
##     P-Value [Acc > NIR] : 0.00371
##
##                   Kappa : 0.867
##
##  Mcnemar's Test P-Value : 0.24821
##
##             Sensitivity : 0.9739
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.7857
##              Prevalence : 0.9127
```

```
##          Detection Rate : 0.8889
##    Detection Prevalence : 0.8889
##       Balanced Accuracy : 0.9870
##
##        'Positive' Class : No
##
```

Accuracy of 97.6% with Sensitivity of 97.4% and Specificity of 100%. This seems to be a good model.

Boosting Model – GBM

Let us now apply Boosting Techniques and train the model considering all the variables.

```
### Creating the Train and Test Data
boost.cars.train = smote.cars_train
boost.cars.train$Transport = ifelse(boost.cars.train$Transport == "Yes",1,0)

boost.cars.test = cars_test
boost.cars.test$Transport = ifelse(boost.cars.test$Transport == "Yes",1,0)
```

Train the Boosting Model

```
cars.gbm = gbm(
  formula = Transport ~ .,
  distribution = "bernoulli",
  data = boost.cars.train,
  n.trees = 10000,
  interaction.depth = 1,
  shrinkage = 0.001,
  cv.folds = 5,
  n.cores = NULL,
  verbose = FALSE
)
```

Evaluate Boosting Model with Confusion Matrix

```
boost.cars.test$pred_boost = predict(cars.gbm, boost.cars.test, type="response")

## Using 10000 trees...

boost.cars.test$pred_boost = floor(boost.cars.test$pred_boost+0.5)

boost.cars.test$Transport = as.factor(boost.cars.test$Transport)
boost.cars.test$Transport = factor(boost.cars.test$Transport, labels = c ("No","Yes")
)

boost.cars.test$pred_boost = as.factor(boost.cars.test$pred_boost)
boost.cars.test$pred_boost = factor(boost.cars.test$pred_boost, labels = c ("No","Yes
"))

confusionMatrix(boost.cars.test$pred_boost, boost.cars.test$Transport)

## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  113   0
```

```
##        Yes   2  11
##
##                Accuracy : 0.9841
##                  95% CI : (0.9438, 0.9981)
##     No Information Rate : 0.9127
##     P-Value [Acc > NIR] : 0.0008535
##
##                   Kappa : 0.908
##
##  Mcnemar's Test P-Value : 0.4795001
##
##             Sensitivity : 0.9826
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.8462
##              Prevalence : 0.9127
##          Detection Rate : 0.8968
##    Detection Prevalence : 0.8968
##       Balanced Accuracy : 0.9913
##
##        'Positive' Class : No
##
```

Accuracy of 98.4% with Sensitivity of 97.4% and Specificity of 100%. This seems to be a good model.

Overall Best Model

Model/Measure	Logistic (Model3)	KNN	Naïve Bayes	Bagging	Boosting
Accuracy (%)	95.2	98.4	94.4	97.6	98.4
Sensitivity (%)	94.8	98.3	93.9	97.4	98.3
Specificity (%)	100	100	100	100	100

Looking at the table it seems that KNN and Boosting method are the best methods to model for the set dataset. However, we should also acknowledge the fact that SMOTE helped us in reaching this otherwise the results could have been different.