Project: Cars Prediction

Predictive Modelling

By: Khaled Majzoub

Table of Contents

1- 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, and 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?**

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

2- EDA

```
3- > setwd("C:/Users/khaled Majzoub/Desktop/R/Predicitve Modeling/Project"
   )
4- > cars = read.csv("Cars-dataset.csv")
```

Libraries needed:

```
> library(car) # use for multicollinearity test (i.e. Variance Inflation Fact
or(VIF))
> library(MASS) # use for basic statistics
> library(dummies) # use for dummy variable transformation(i.e. One-Hot Encod
ing)
> library(ggplot2) # use for visualisation
> library(caret) # use for LM model training i.e Naive bayes (train() functio
n )
> library(Information) # use for calculating WOE and Information value
> library(caTools)
> library(ROCR) # use for ROC curve
> library(dplyr) # use for basic data wrangling
> library(tidyr) # Converting data shape- long to wide or wide to long format
> library(corrplot) # for correlation analysis
> library(ggplot2) # for visualization
> library(GGally) # for better visualization of multiple plots in one grid
> library(factoextra) # use for PCA techinque
> library(e1071) # using for machine learning models(i.e.Naive Bayes,KNN Mode
ls)
```

Checking Data:

```
> head(cars)
  Age Gender Engineer MBA Work.Exp Salary Distance license Transport
1  28   Male        1   0        5   14.4      5.1       0  2Wheeler
2  24   Male        1   0        6   10.6      6.1       0  2Wheeler
3  27 Female        1   0        9   15.5      6.1       0  2Wheeler
4  25   Male        0   0        1    7.6      6.3       0  2Wheeler
5  25 Female        0   0        3    9.6      6.7       0  2Wheeler
6  21   Male        0   0        3    9.5      7.1       0  2Wheeler
> table(cars$Transport)

        2Wheeler                Car Public Transport 
              83                 35              300 
> 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 ...
```

- **Transport is the dependat variable , it is a factor type of 3 levels**
- **Other variables are numeric , integer or factor as “Gender “**
- **Few of the variables have 0 or 1 range**

Any Nas:

```
> anyNA(cars)
[1] TRUE
> sum(is.na(cars))
```

```
[1] 1
> sum(is.na(cars$Age))
[1] 0
> sum(is.na(cars$Gender))
[1] 0
> sum(is.na(cars$Engineer))
[1] 0
> sum(is.na(cars$MBA))
[1] 1
```

1 Na under the MBA , I will delete it since it is only 1

```
> cars = na.omit(cars)
> anyNA(cars)
[1] FALSE
```

Now we have 417 obs from 418 obs

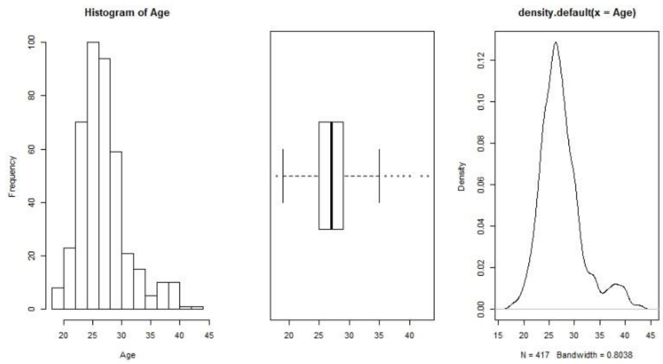
```
> summary(cars)
      Age          Gender        Engineer           MBA              Work.Exp    
 Min.   :18.00   Female:120   Min.   :0.0000   Min.   :0.0000   Min.   : 0.00
0
 1st Qu.:25.00   Male  :297   1st Qu.:1.0000   1st Qu.:0.0000   1st Qu.: 3.00
0
 Median :27.00                Median :1.0000   Median :0.0000   Median : 5.00
0
 Mean   :27.33                Mean   :0.7506   Mean   :0.2614   Mean   : 5.87
3
 3rd Qu.:29.00                3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.: 8.00
0
 Max.   :43.00                Max.   :1.0000   Max.   :1.0000   Max.   :24.00
0
     Salary          Distance        license                    Transport  
 Min.   : 6.50   Min.   : 3.2   Min.   :0.0000   2Wheeler        : 83  
 1st Qu.: 9.60   1st Qu.: 8.6   1st Qu.:0.0000   Car             : 35  
 Median :13.00   Median :10.9   Median :0.0000   Public Transport:299  
 Mean   :15.42   Mean   :11.3   Mean   :0.2038                         
 3rd Qu.:14.90   3rd Qu.:13.6   3rd Qu.:0.0000                         
 Max.   :57.00   Max.   :23.4   Max.   :1.0000                         
```

Changing the variables to correct format

```
> cars$Engineer = as.factor(cars$Engineer)
> cars$MBA = as.factor(cars$MBA)
> cars$license = as.factor(cars$license)
> str(cars)
'data.frame':   417 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 ...
 - attr(*, "na.action")= 'omit' Named int 243
  ..- attr(*, "names")= chr "243"
```

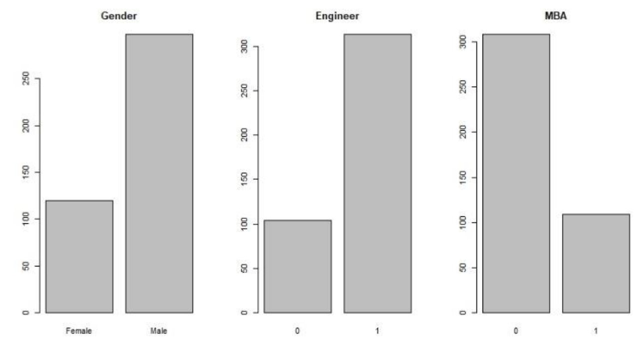
Now is ok to start with the univariate and multivariate

```
> attach(cars)
> par(mfrow=c(1,3))
> hist(Age)
> boxplot(Age, horizontal = TRUE)
> plot(density(Age))
```



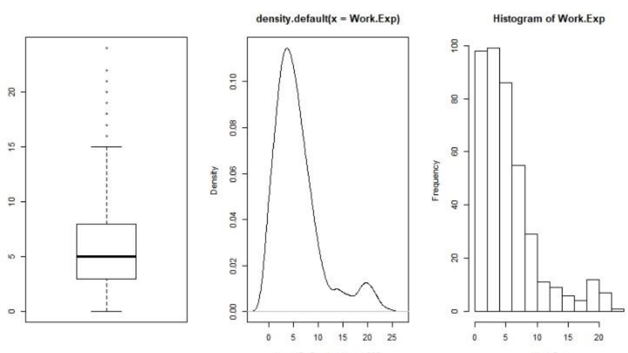
normal distribution with outliers

```
> plot(Gender, main = "Gender")
> plot(Engineer, main = "Engineer")
> plot(MBA, main = "MBA")
```



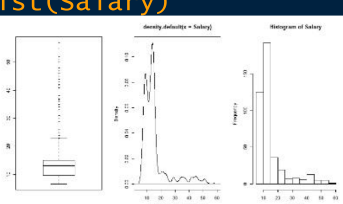
male, Engineers, non MBA's are more

```
> boxplot(Work.Exp)
> plot(density(Work.Exp))
> hist(Work.Exp)
```



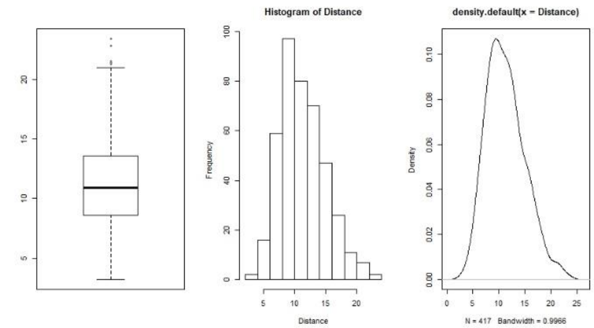
Work Exp normal dist skewed with outliers , it will be fixed

```
> boxplot(Salary)
> plot(density(Salary))
> hist(Salary)
```



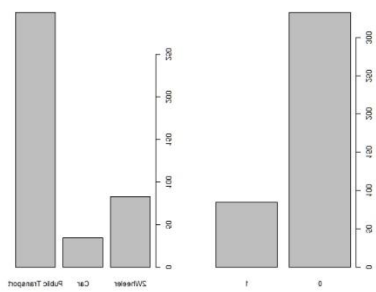
salary normal , skewed with outliers

```
> boxplot(Distance)
> hist(Distance)
> plot(density(Distance))
```



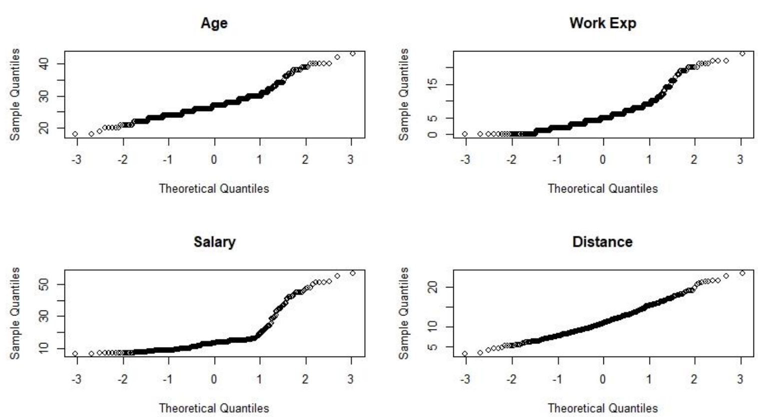
distance normal , little outliers

```
> plot(license)
> plot(Transport)
```



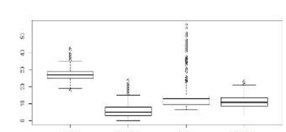
license, I think this variable affects the dependent , most of them don't have license – Transport The Y variable most of them use public – less 2 wheelers and least have cars.

I think license and distance are the most important variables for car prediction

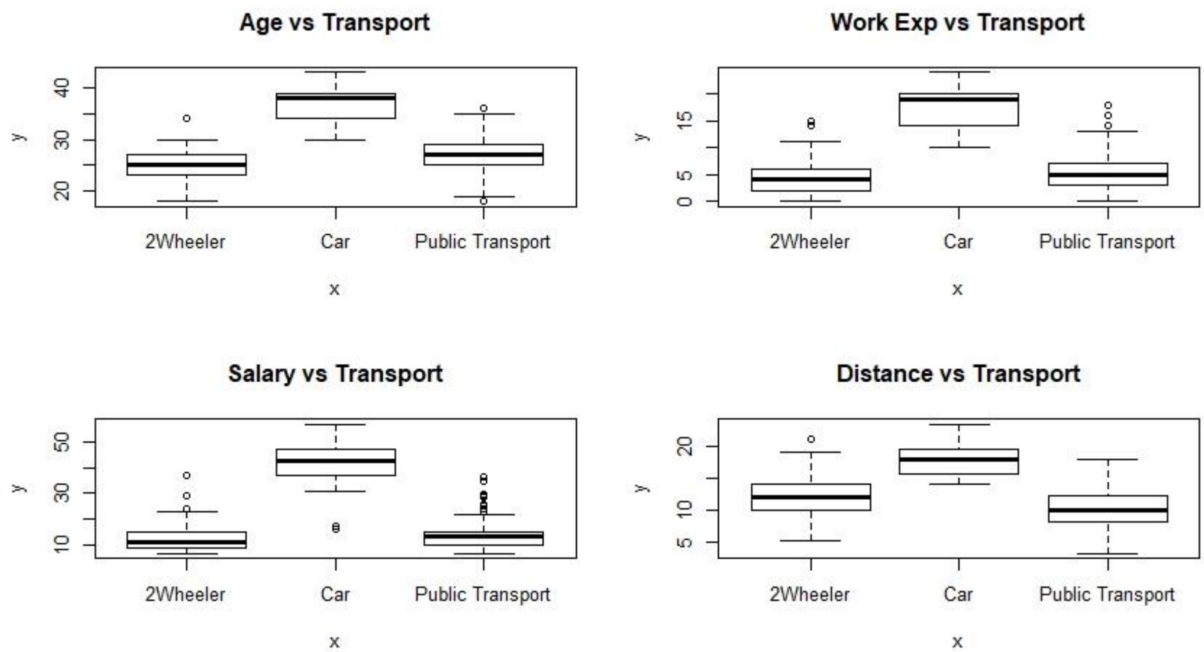


All of them have normality

```
boxplot(Age,Work.Exp,Salary,Distance, names = c( "Age" , "Work Exp", "Salary"
, "Distance") )
```



Multivariate



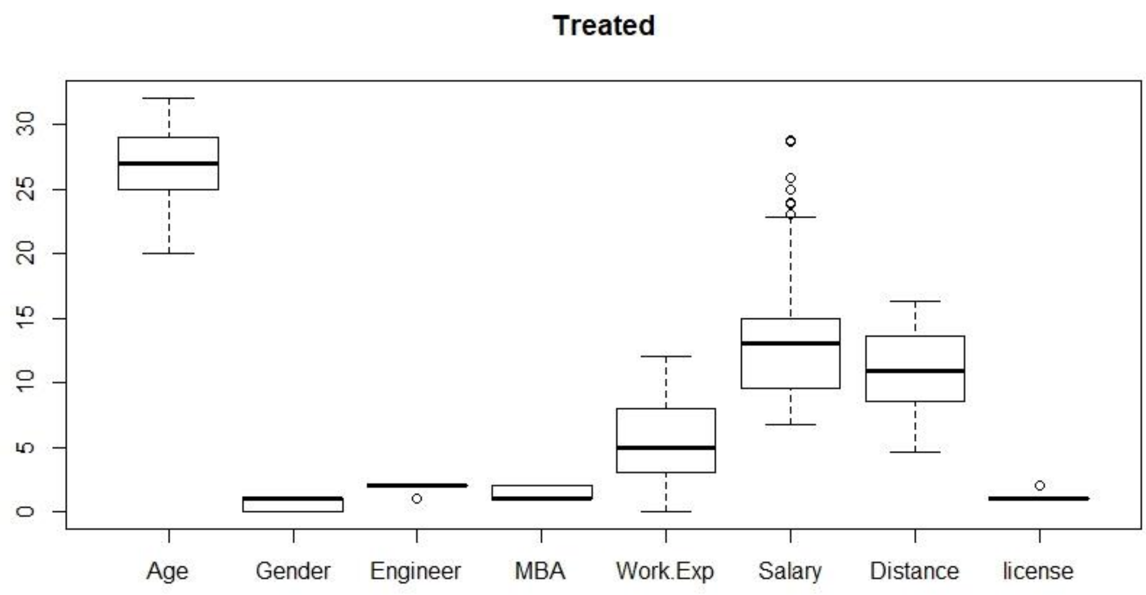
1- **Age Vs Trans : bigger ages have cars > above 30**
2- **Work Exp Vs Trans : the more Exp have cars > 10 years**
3- **Salary Vs Trans : Higher salaries have cars > 15 k**
4- **Distance Vs Trans: longer distance have cars> 15km , I think this is the most important variable**

```
• > plot(Transport,license, main = "License vs Trans")
• > plot(Transport,Gender, main = "Gender vs Trans")
• > plot(Transport,Engineer , main = "Engineer vs Trans")
• > plot(Transport,MBA, main = "MBA vs Trans")
```

Treating outliers

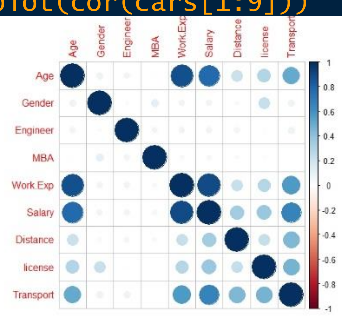
```
> new_vars <- c("Age","Gender","Engineer","MBA","Work.Exp","Salary","Distance
","license","Transport")
> cars$Gender = ifelse(cars$Gender=='Male',1,0)
> cars$Gender = as.integer(cars$Gender)
> cars$Engineer = as.integer(cars$Engineer)
> cars$MBA = as.integer(cars$MBA)
> cars$license = as.integer(cars$license)
> str(cars)
'data.frame':   417 obs. of  9 variables:
 $ Age      : num  28 24 27 25 25 21 23 23 24 28 ...
 $ Gender   : int  1 1 0 1 0 1 1 1 1 1 ...
 $ Engineer : int  2 2 2 1 1 1 2 1 2 2 ...
 $ MBA      : int  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   : int  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 ...
 - attr(*, "na.action")= 'omit' Named int 243
  ..- attr(*, "names")= chr "243"
> outlier_treatment_fun = function(data,var_name){
+   capping = as.vector(quantile(data[,var_name],0.9))
+   flooring = as.vector(quantile(data[,var_name],0.01))
+   data[,var_name][which(data[,var_name]<flooring)]<- flooring
+   data[,var_name][which(data[,var_name]>capping)]<- capping
+   #print('done',var_name)
+   return(data)
+ }
> for(i in new_vars[1:8]){
+   cars =  outlier_treatment_fun(cars,i)
+ }
boxplot(Age,Gender,Engineer, MBA,Work.Exp,Salary,Distance,license)
```



we took out the outliers

```
> cars$Transport = ifelse(cars$Transport == "Car","1","0")
> cars$Transport = as.integer(cars$Transport)
> corrplot(cor(cars[1:9]))
```



Related to Transport: Age – work exp – salary – distance – license

```
> 35/417 ## cars employees rate
[1] 0.08393285
```

What is the most challenging aspect of this problem? What method will you use to deal with this?

- **The dependent Variable Y (Transport) is made of 3 levels**
- **Different models needs different data types, like logistic, with only factors type and NB with only numeric, so results will vary.**

3- Modeling

A- Logistic Regression:

```
B- log.cars = cars
```

Changing the transport into binomial & data treatment

```
> str(log.cars)
'data.frame':	417 obs. of  9 variables:
 $ Age      : num  28 24 27 25 25 21 23 23 24 28 ...
 $ Gender   : num  1 1 0 1 0 1 1 1 1 1 ...
 $ Engineer : num  2 2 2 1 1 1 2 1 2 2 ...
 $ MBA      : num  1 1 1 1 1 1 2 1 1 1 ...
 $ Work.Exp : num  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 6.8 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  : num  1 1 1 1 1 1 1 1 1 2 ...
 $ Transport: int  0 0 0 0 0 0 0 0 0 0 ...
 - attr(*, "na.action")= 'omit' Named int 243
  ..- attr(*, "names")= chr "243"
> log.cars$Transport = as.factor(log.cars$Transport)
> log.cars$Engineer = as.factor(log.cars$Engineer)
> log.cars$MBA = as.factor(log.cars$MBA)
> log.cars$Gender = as.factor(log.cars$Gender)
> log.cars$license = as.factor(log.cars$license)
> str(log.cars)
'data.frame':	417 obs. of  9 variables:
 $ Age      : num  28 24 27 25 25 21 23 23 24 28 ...
 $ Gender   : Factor w/ 2 levels "0","1": 2 2 1 2 1 2 2 2 2 2 ...
 $ Engineer : Factor w/ 2 levels "1","2": 2 2 2 1 1 1 2 1 2 2 ...
 $ MBA      : Factor w/ 2 levels "1","2": 1 1 1 1 1 1 2 1 1 1 ...
 $ Work.Exp : num  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 6.8 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 "1","2": 1 1 1 1 1 1 1 1 1 2 ...
 $ Transport: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 - attr(*, "na.action")= 'omit' Named int 243
  ..- attr(*, "names")= chr "243"
```

The model

```
> set.seed(123)
> split.indices = sample.split(log.cars$Transport,SplitRatio = .7)
> logistic.train.cars = log.cars[split.indices,]
> logistic.test.cars = log.cars[!split.indices,]
> print(nrow(logistic.test.cars)/nrow(log.cars))
[1] 0.3021583
> print(nrow(logistic.train.cars)/nrow(log.cars))
[1] 0.6978417
y = "binomial")
```

```
> logistic.test.model = glm(Transport~Gender+Engineer+MBA+license ,data = log
istic.test.cars,family = "binomial")
> logistic.train.model = glm(Transport~Gender+Engineer+MBA+license,data = log
istic.train.cars,family = "binomial")
> logistic.test.model

Call:  glm(formula = Transport ~ Gender + Engineer + MBA + license, 
    family = "binomial", data = logistic.test.cars)

Coefficients:
(Intercept)      Gender1    Engineer2         MBA2     license2  
   -19.3486      -1.7889      17.0633      -0.5524       3.8101  

Degrees of Freedom: 125 Total (i.e. Null);  121 Residual
Null Deviance:      74.65 
Residual Deviance: 48.28        AIC: 58.28
```

-19 the log odds ---- - 1.7 Gender ---- 17 Engineer --- -0.552 MBA – 3.2 license

```
> logistic.train.model = glm(Transport~Gender+Engineer+MBA+license,data = log
istic.train.cars,family = "binomial")
> logistic.train.model

Call:  glm(formula = Transport ~ Gender + Engineer + MBA + license, 
    family = "binomial", data = logistic.train.cars)

Coefficients:
(Intercept)      Gender1    Engineer2         MBA2     license2  
   -4.79353      0.09198      0.54117     -0.01733      3.59906  

Degrees of Freedom: 290 Total (i.e. Null);  286 Residual
Null Deviance:      165.7 
Residual Deviance: 112        AIC: 122
```

In the train it is different numbers and not matching

```
> summary(logistic.test.model)

Call:
glm(formula = Transport ~ Gender + Engineer + MBA + license, 
    family = "binomial", data = logistic.test.cars)

Deviance Residuals: 
     Min        1Q    Median        3Q       Max  
-1.06748  -0.33736  -0.18364  -0.00009   2.86046  

Coefficients:
             Estimate Std. Error z value Pr(>|z|)    
(Intercept)  -19.3486  1906.0771  -0.010 0.991901    
Gender1       -1.7889     1.1823  -1.513 0.130254    
Engineer2     17.0633  1906.0770   0.009 0.992857    
MBA2          -0.5524     0.9205  -0.600 0.548408    
license2       3.8101     1.1076   3.440 0.000582 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 74.655  on 125  degrees of freedom
Residual deviance: 48.279  on 121  degrees of freedom
AIC: 58.279

Number of Fisher Scoring iterations: 18
```

P value : License only is significant

```
> summary(logistic.train.model)

Call:
glm(formula = Transport ~ Gender + Engineer + MBA + license,
    family = "binomial", data = logistic.train.cars)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.9501  -0.1760  -0.1744  -0.1345   3.0990

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.79353    0.94492  -5.073 3.92e-07 ***
Gender1      0.09198    0.73993   0.124    0.901
Engineer2    0.54117    0.59425   0.911    0.362
MBA2        -0.01733    0.54819  -0.032    0.975
license2     3.59906    0.66127   5.443 5.25e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 165.74  on 290  degrees of freedom
Residual deviance: 112.03  on 286  degrees of freedom
AIC: 122.03

Number of Fisher Scoring iterations: 7
```

Same!

```
> vif(logistic.test.model)
  Gender Engineer      MBA  license
2.100517 1.000000 1.011506 2.099162
> vif(logistic.train.model)
  Gender Engineer      MBA  license
1.085095 1.055768 1.038965 1.067344
```

Below 5 is good

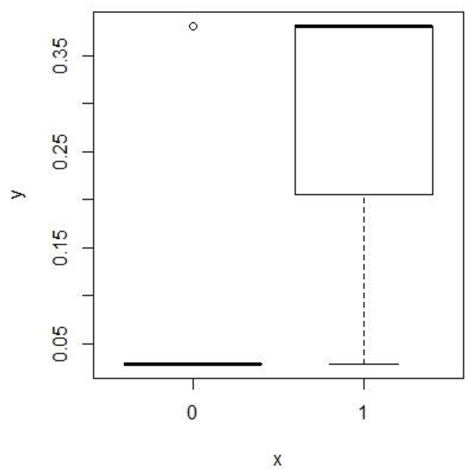
We will build the mode on license

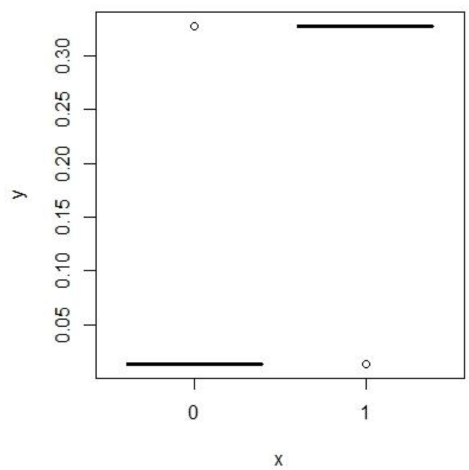
```
> summary(logistic.test.model)

Call:
glm(formula = Transport ~ license, family = "binomial", data = logistic.test.
cars)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.9794  -0.2408  -0.2408  -0.2408   2.6666

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -3.5264     0.5858  -6.020 1.74e-09 ***
license2      3.0409     0.7383   4.119 3.81e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 74.655  on 125  degrees of freedom
Residual deviance: 55.156  on 124  degrees of freedom
AIC: 59.156

Number of Fisher Scoring iterations: 6
```

```
 > logistic.train.model = glm(Transport~license,data = logistic.train.cars,fa
mily = "binomial")
> summary(logistic.train.model)

Call:
glm(formula = Transport ~ license, family = "binomial", data = logistic.train
.cars)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.8918  -0.1631  -0.1631  -0.1631   2.9415

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -4.3130     0.5812  -7.421 1.16e-13 ***
license2      3.5964     0.6393   5.626 1.85e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 165.74  on 290  degrees of freedom
Residual deviance: 112.92  on 289  degrees of freedom
AIC: 116.92

Number of Fisher Scoring iterations: 7
```

This model is good

```
> logistic.test.model$fitted.values
> logistic.train.model$fitted.values
> plot(logistic.test.cars$Transport,logistic.test.model$fitted.values)
> plot(logistic.train.cars$Transport,logistic.train.model$fitted.values)
```





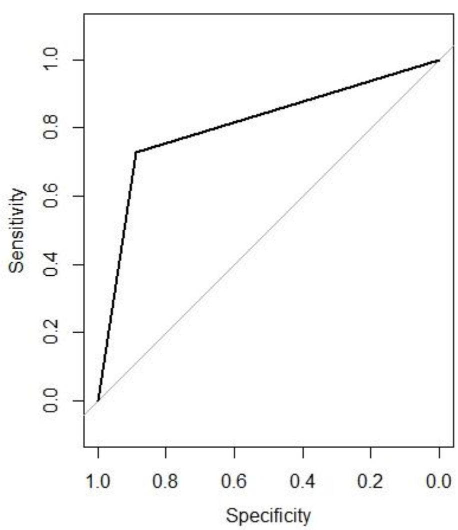
I have no idea what is this

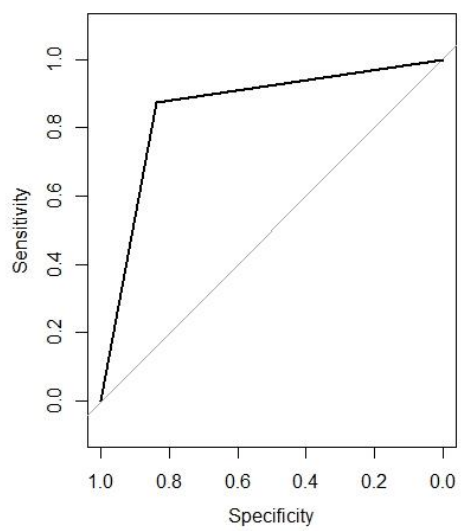
```
> logistic.test.Predict = ifelse(logistic.test.model$fitted.values<.9,"no car
" , "yes car")
> logistic.train.Predict = ifelse(logistic.train.model$fitted.values<.9,"no c
ar" , "yes car")
> table(logistic.test.cars$Transport,logistic.test.Predict)
   logistic.test.Predict
    no car
```

```
  0     115
  1      11
> table(logistic.train.cars$Transport,logistic.train.Predict)
   logistic.train.Predict
    no car
  0     267
  1      24
```

I don't know why there is no yes car

```
> library(pROC)
> roc(logistic.test.cars$Transport,logistic.test.model$fitted.values)
Setting levels: control = 0, case = 1
Setting direction: controls < cases

Call:
roc.default(response = logistic.test.cars$Transport, predictor = logistic.tes
t.model$fitted.values)

Data: logistic.test.model$fitted.values in 115 controls (logistic.test.cars$T
ransport 0) < 11 cases (logistic.test.cars$Transport 1).
Area under the curve: 0.8071
> plot(roc(logistic.test.cars$Transport,logistic.test.model$fitted.values))
Setting levels: control = 0, case = 1
Setting direction: controls < cases
> roc(logistic.train.cars$Transport,logistic.train.model$fitted.values)
Setting levels: control = 0, case = 1
Setting direction: controls < cases

Call:
roc.default(response = logistic.train.cars$Transport, predictor = logistic.tr
ain.model$fitted.values)

Data: logistic.train.model$fitted.values in 267 controls (logistic.train.cars
$Transport 0) < 24 cases (logistic.train.cars$Transport 1).
Area under the curve: 0.857
> plot(roc(logistic.train.cars$Transport,logistic.train.model$fitted.values))
Setting levels: control = 0, case = 1
Setting direction: controls < cases
```





.8 - .85 which is a good model

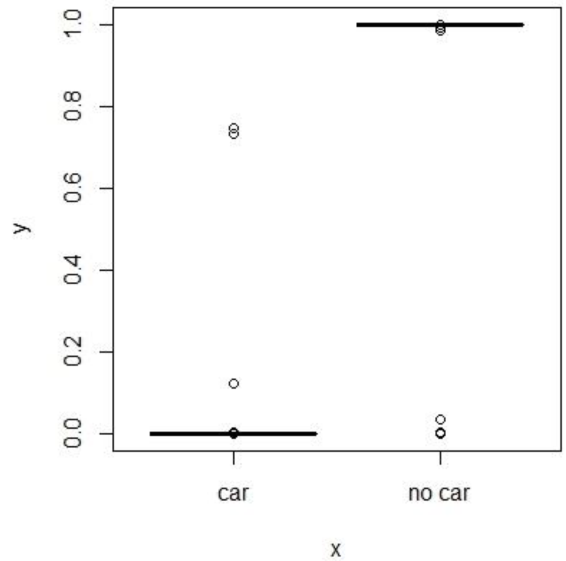
B – Naïve Bayes

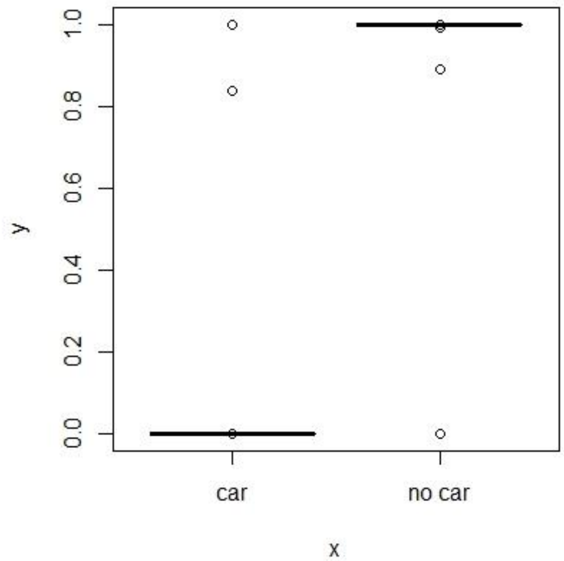
```
> nb.cars = cars
> nb.cars$Engineer = as.factor(nb.cars$Engineer)
> nb.cars$MBA = as.factor(nb.cars$MBA)
> nb.cars$Gender = as.factor(nb.cars$Gender)
> nb.cars$license = as.factor(nb.cars$license)
> nb.cars$Transport = ifelse(nb.cars$Transport == "1","car","no car")
> nb.cars$Transport = as.factor(nb.cars$Transport)
> str(nb.cars)
'data.frame':   417 obs. of  9 variables:
 $ Age      : num  28 24 27 25 25 21 23 23 24 28 ...
 $ Gender   : Factor w/ 2 levels "0","1": 2 2 1 2 1 2 2 2 2 2 ...
 $ Engineer : Factor w/ 2 levels "1","2": 2 2 2 1 1 1 2 1 2 2 ...
 $ MBA      : Factor w/ 2 levels "1","2": 1 1 1 1 1 1 2 1 1 1 ...
 $ Work.Exp : num  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 6.8 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 "1","2": 1 1 1 1 1 1 1 1 1 2 ...
 $ Transport: Factor w/ 2 levels "car","no car": 2 2 2 2 2 2 2 2 2 2 ...
 - attr(*, "na.action")= 'omit' Named int 243
  ..- attr(*, "names")= chr "243"
> summary(nb.cars)
      Age           Gender   Engineer MBA        Work.Exp          Salary      
Distance  
 Min.   :20.00   0:120    1:104    1:308    Min.   : 0.000   Min.   : 6.80   Mi
n.   : 4.632  
 1st Qu.:25.00   1:297    2:313    2:109    1st Qu.: 3.000   1st Qu.: 9.60   1s
t Qu.: 8.600  
 Median :27.00                              Median : 5.000   Median :13.00   Me
dian :10.900  
 Mean   :26.92                              Mean   : 5.321   Mean   :14.18   Me
an   :11.087  
 3rd Qu.:29.00                              3rd Qu.: 8.000   3rd Qu.:14.90   3r
d Qu.:13.600  
 Max.   :32.40                              Max.   :12.000   Max.   :28.74   Ma
x.   :16.340  
 license  Transport  
 1:332    car   : 35  
 2: 85    no car:382  
> ## train - test
> spilt.indices = sample.split(nb.cars$Transport,SplitRatio = .7)
> NB.train.cars = nb.cars[split.indices,]
> NB.test.cars = nb.cars[!split.indices,]
> print(nrow(NB.test.cars)/nrow(nb.cars))
[1] 0.3021583
> print(nrow(NB.train.cars)/nrow(nb.cars))
[1] 0.6978417
```

Model

```
> set.seed(123)
> NB.model.train = naiveBayes(Transport~Age+Work.Exp+Salary+Distance , data = 
NB.train.cars)
> NB.model.test = naiveBayes(Transport~Age+Work.Exp+Salary+Distance , data = 
NB.test.cars)
> NB.model.train

Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)
```

```
A-priori probabilities:
Y
       car     no car 
0.08247423 0.91752577 

Conditional probabilities:
        Age
Y             [,1]      [,2]
  car    32.15000 0.6079188
  no car 26.43221 2.8175624

        Work.Exp
Y              [,1]      [,2]
  car    11.666667 0.7019641
  no car  4.756554 2.9335857

        Salary
Y             [,1]     [,2]
  car    27.17667 4.229711
  no car 12.95086 4.604337

        Distance
Y             [,1]      [,2]
  car    15.85750 0.8340016
  no car 10.66174 3.0204909
> NB.model.test

Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
       car     no car 
0.08730159 0.91269841 

Conditional probabilities:
        Age
Y             [,1]      [,2]
  car    32.36364 0.1206045
  no car 26.42435 2.8794673

        Work.Exp
Y              [,1]    [,2]
  car    12.000000 0.00000
  no car  4.669565 3.15873

        Salary
Y             [,1]     [,2]
  car    28.74000 0.000000
  no car 12.91235 4.759763

        Distance
Y             [,1]      [,2]
  car    15.88000 0.8462151
  no car 10.62083 3.0369063

> predict.NB.model.train = predict(NB.model.train,type = "raw",newdata = nb.c
ars)
> plot(nb.cars$Transport,predict.NB.model.train[,2])
> predict.NB.model.test = predict(NB.model.test,type = "raw",newdata = nb.car
s)
> plot(nb.cars$Transport,predict.NB.model.test[,2])
```





I have no idea why it looks like this

```
> summary(predict.NB.model.test)
      car               no car
 Min.   :0.00000   Min.   :0.0000
 1st Qu.:0.00000   1st Qu.:1.0000
 Median :0.00000   Median :1.0000
 Mean   :0.08461   Mean   :0.9154
 3rd Qu.:0.00000   3rd Qu.:1.0000
 Max.   :1.00000   Max.   :1.0000
> summary(predict.NB.model.train)
      car               no car
 Min.   :0.00000   Min.   :0.0000005
 1st Qu.:0.00000   1st Qu.:1.0000000
 Median :0.00000   Median :1.0000000
 Mean   :0.08964   Mean   :0.9103596
 3rd Qu.:0.00000   3rd Qu.:1.0000000
 Max.   :1.00000   Max.   :1.0000000
> pred.NB.test = predict(NB.model.test,NB.test.cars,type = "raw")
> pred.NB.train = predict(NB.model.train,NB.train.cars,type = "raw")
> summary(pred.NB.test)
      car              no car
 Min.   :0.0000   Min.   :0.0000
 1st Qu.:0.0000   1st Qu.:1.0000
 Median :0.0000   Median :1.0000
 Mean   :0.0953   Mean   :0.9047
 3rd Qu.:0.0000   3rd Qu.:1.0000
 Max.   :1.0000   Max.   :1.0000
> summary(pred.NB.train)
      car               no car
 Min.   :0.00000   Min.   :0.0000005
 1st Qu.:0.00000   1st Qu.:1.0000000
 Median :0.00000   Median :1.0000000
 Mean   :0.09062   Mean   :0.9093750
 3rd Qu.:0.00000   3rd Qu.:1.0000000
 Max.   :1.00000   Max.   :1.0000000
```

```
> NB.predict.response.train= factor(ifelse(pred.NB.train >= 0.9, "car","no ca
r"))
> NB.predict.response.test= factor(ifelse(pred.NB.test >= 0.9, "car","no car"
))
> NB.test.matrix = confusionMatrix(NB.predict.response.test, NB.test.cars$Tra
nsport , positive = "car")
> summary(NB.predict.response.test)
   car no car
   126    126
> summary(NB.predict.response.train)
   car no car
   288    294
```

C– KNN

```
> knn.cars = cars
> knn.cars$Gender = as.factor(knn.cars$Gender)
> knn.cars$Transport = ifelse(knn.cars$Transport == "1","car","not car")
> knn.cars$Transport = as.factor(knn.cars$Transport)
> knn.cars$Engineer = as.factor(knn.cars$Engineer)
> knn.cars$MBA = as.factor(knn.cars$MBA)
> knn.cars$license = as.factor(knn.cars$license)
> str(knn.cars)
'data.frame':	417 obs. of  9 variables:
 $ Age      : num  28 24 27 25 25 21 23 23 24 28 ...
 $ Gender   : Factor w/ 2 levels "0","1": 2 2 1 2 1 2 2 2 2 2 ...
 $ Engineer : Factor w/ 2 levels "1","2": 2 2 2 1 1 1 2 1 2 2 ...
 $ MBA      : Factor w/ 2 levels "1","2": 1 1 1 1 1 1 2 1 1 1 ...
 $ Work.Exp : num  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 6.8 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 "1","2": 1 1 1 1 1 1 1 1 1 2 ...
 $ Transport: Factor w/ 2 levels "car","not car": 2 2 2 2 2 2 2 2 2 2 ...
 - attr(*, "na.action")= 'omit' Named int 243
  ..- attr(*, "names")= chr "243"
> summary(knn.cars)
      Age         Gender   Engineer MBA        Work.Exp          Salary       
Distance    
 Min.   :20.00   0:120   1:104    1:308   Min.   : 0.000   Min.   : 6.80   Mi
n.   : 4.632  
 1st Qu.:25.00   1:297   2:313    2:109   1st Qu.: 3.000   1st Qu.: 9.60   1s
t Qu.: 8.600  
 Median :27.00                            Median : 5.000   Median :13.00   Me
dian :10.900  
 Mean   :26.92                            Mean   : 5.321   Mean   :14.18   Me
an   :11.087  
 3rd Qu.:29.00                            3rd Qu.: 8.000   3rd Qu.:14.90   3r
d Qu.:13.600  
 Max.   :32.40                            Max.   :12.000   Max.   :28.74   Ma
x.   :16.340  
 license   Transport  
 1:332   car    : 35  
 2: 85   not car:382  
> set.seed(1)
> dim(knn.cars)
[1] 417   9
> index=sample(417,317)
> KNN.train.cars = knn.cars[index,]
> KNN.test.cars = knn.cars[-index,]
> dim(KNN.test.cars)
[1] 100   9
```

```
> dim(KNN.train.cars)
[1] 317   9
> library(class)
> knn.model = knn (KNN.train.cars[,c(6,7)],KNN.test.cars[,c(6,7)], KNN.train.
cars$Transport ,k=5)
> table(KNN.test.cars$Transport,knn.model)
         knn.model
          car not car
  car       5       1
  not car   0      94
> summary(knn.model)
    car not car
      5      95
```

- **Apply both boosting and bagging**

```
•  > library(gbm)
•  > library(xgboost)
•  > library(caret)
•  > library(ipred)
•  > library(rpart)
```

Bagging

```
> bag.cars = cars
> bag.cars$Transport= as.factor(bag.cars$Transport)
> str(bag.cars)
'data.frame':   417 obs. of  9 variables:
 $ Age      : num  28 24 27 25 25 21 23 23 24 28 ...
 $ Gender   : num  1 1 0 1 0 1 1 1 1 1 ...
 $ Engineer : num  2 2 2 1 1 1 2 1 2 2 ...
 $ MBA      : num  1 1 1 1 1 1 2 1 1 1 ...
 $ Work.Exp : num  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 6.8 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  : num  1 1 1 1 1 1 1 1 1 2 ...
 $ Transport: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 - attr(*, "na.action")= 'omit' Named int 243
  ..- attr(*, "names")= chr "243"
> split = sample.split(bag.cars$Transport, SplitRatio = .75)
> bag.cars.train = subset(bag.cars, split == FALSE)
> bag.cars.test = subset(bag.cars, split == TRUE)
> table(bag.cars.test$Transport)

  0   1
286  26
> table(bag.cars.train$Transport)

 0  1
96  9
> cars.bagging.train = bagging(Transport~. , data = bag.cars.train, rpart.con
trol(maxdepth = 5 , minsplit = 15))
Error in `[.default`(xj, i) : invalid subscript type 'list'
> cars.bagging.test = bagging(Transport~.,data = bag.cars.test,rpart.control(
maxdepth = 5
+
, minsplit = 15))
Error in `[.default`(xj, i) : invalid subscript type 'list'
```

Thanks to the expert, they never answered for this error , they told me to change all variables to numeric except for the Y variable

```
 > table(bagging.cars.test$Transport,bagging.cars.test$pred.class)
Error in table(bagging.cars.test$Transport, bagging.cars.test$pred.class) :
  object 'bagging.cars.test' not found
> table(bagging.cars.train$Transport,bagging.cars.train$pred.class)
Error in table(bagging.cars.train$Transport, bagging.cars.train$pred.class) :
  object 'bagging.cars.train' not found
```

Boosting

```
> gbm.fit = gbm(formula = Transport~. , distribution = "gaussian",
+                data = bag.cars.train , n.trees = 1000, interaction.depth=1
+                , shrinkage = .001 , cv.folds = 5 , n.cores = NULL , verbose
= FALSE )
> bag.cars.test$pred.class = predict(gbm.fit,bag.cars.test, type = "response"
)
Using 1000 trees...
> table(bag.cars.test$Transport, bag.cars.test$pred.class>.5)

    TRUE
  0  286
  1   26
```

I have no idea why only True….

```
> ### XG Boost
> ?xgboost
> feature.train = as.matrix(bag.cars.train[,1:8])
> label_train = as.matrix(bag.cars.train[,9])
> feature.test = as.matrix(bag.cars.test[,1:8])
> xgb.fit = xgboost(data = feature.train, label = label_train, eta = .001,
+                    max_depth = 3, min_child_weight = 3 , nrounds = 10000, nf
old = 5,
+                    objective = "binary:logistic" , verbose = 0 , early_stopp
ing_rounds =10)
> bag.cars.test$xgb.pred.class = predict(xgb.fit,bag.cars.test)
Error in xgb.DMatrix(newdata, missing = missing) :
  xgb.DMatrix does not support construction from list
> table(bag.cars.test$Transport, bag.cars.test$pred.class>.5)

    TRUE
  0  286
  1   26
```

4 – Conclusion

I don't have any conclusion. Sorry