This project requires you to understand what mode of transport employees prefers to commute to their office. The attached data 'Cars.csv' 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

Modelling (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

- **1.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)

Rcode:

```
getwd()

head(Cars)

tail(Cars)

dim(Cars)

str(Cars)

summary(Cars)

Car_or_nocar <- ifelse(Cars$Transport == "Car",1,0)

View(Car_or_nocar)

actualcar <- cbind(Cars, Car_or_nocar)

actualcar$Transport <- NULL

actualcar$Gender <-  NULL

View(actualcar)

str(actualcar)
```

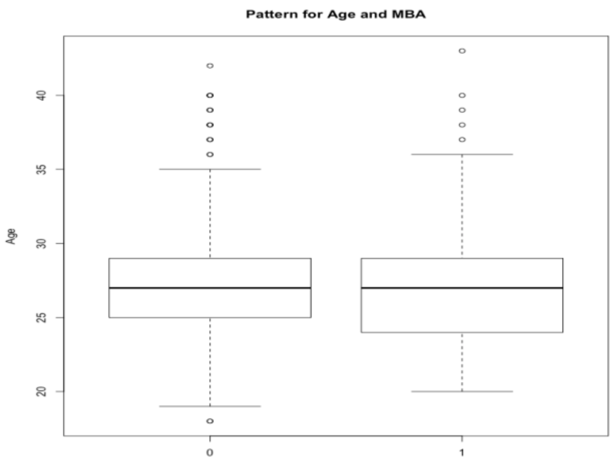
```
Cars$MBA = as.factor(Cars$MBA)

actualcar$Gender = as.numeric(actualcar$Gender)

Cars$license = as.factor(Cars$license)

actualcar$Car_or_nocar = as.factor(actualcar$Car_or_nocar)

actualcar$Car_or_nocar = as.numeric(actualcar$Car_or_nocar)


###########Exploratory Data
Analysis##############################

attach(Cars)

library(ggplot2)

boxplot(Age~MBA, main = "Pattern for Age and MBA", ylab ="Age")

boxplot(`Work Exp`~MBA, main ="Boxplot for Work Exp and MBA",
ylab="Work Exp")

boxplot(Salary~MBA, main ="Boxplot for Salary and MBA", ylab
="Salary")

boxplot(Age~Engineer, main = "Pattern for Age and Engineer", ylab
="Age")

boxplot(`Work Exp`~Engineer, main ="Boxplot for Work Exp and
Engineer", ylab="Work Exp")

boxplot(Salary~Engineer, main ="Boxplot for Salary and MBA", ylab
="Salary")

boxplot(Age~license, main = "Pattern for Age and license", ylab
="Age")
```

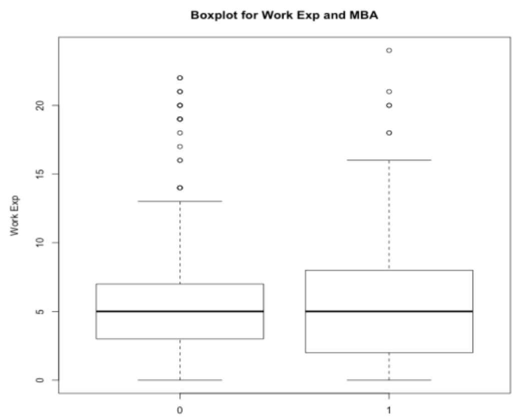
```
boxplot(`Work Exp`~license, main ="Boxplot for Work Exp and license", ylab="Work Exp")

boxplot(Salary~license, main ="Boxplot for Salary and license", ylab ="Salary")
```

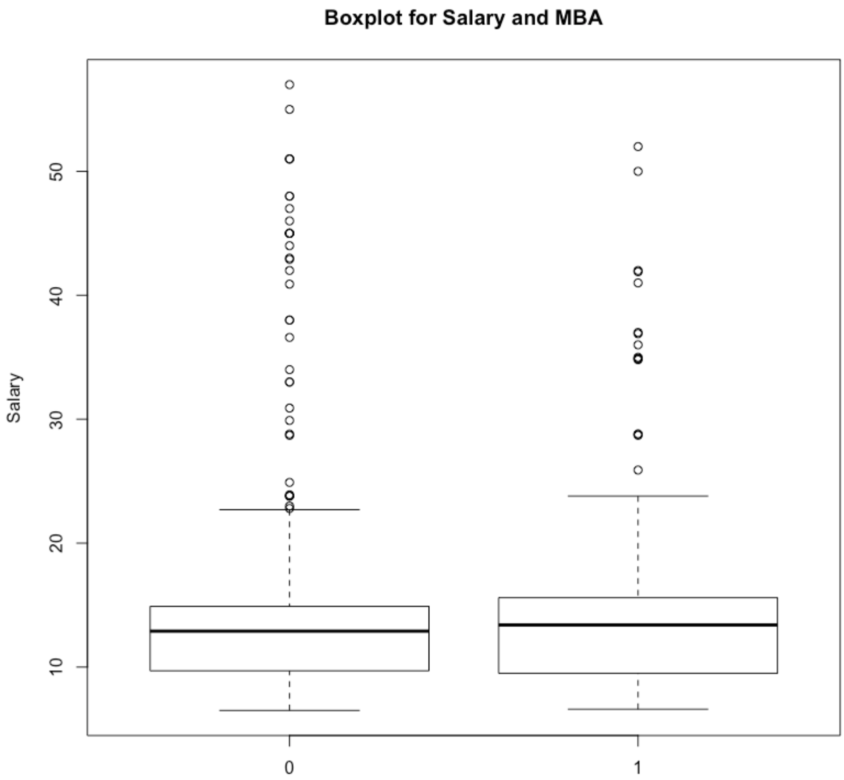
```
####For Categorical Variables##################################

ggplot(Cars, aes(x = Age, fill = MBA))+
  geom_bar(width = 0.25 , alpha=0.5)+
  scale_fill_manual(values = c('red', 'green'))
prop.table(table(Age,MBA),1)*100
ggplot(Cars, aes(x = `Work Exp`, fill = MBA))+
  geom_bar(width = 0.25 , alpha=0.5)+
  scale_fill_manual(values = c('black', 'blue'))
prop.table(table(`Work Exp`,MBA),1)*100
ggplot(Cars, aes(x = Salary, fill = MBA))+
```

```
  geom_bar(width = 0.25 , alpha=0.5)+
  scale_fill_manual(values = c('orange', 'yellow'))
prop.table(table(Salary,MBA),1)*100.
```

Based on the above Exploratory Data Analysis we can see that there are outliers between the variables. Boxplot also shows the various relations between the categorical and continuous variables.







MBA		
Salary	0	1
6.5	100.00000	0.00000
6.6	0.00000	100.00000
6.7	100.00000	0.00000
6.8	100.00000	0.00000
6.9	66.66667	33.33333
7	100.00000	0.00000
7.5	75.00000	25.00000
7.6	75.00000	25.00000

7.7	60.00000	40.00000
7.8	50.00000	50.00000
7.9	60.00000	40.00000
8	100.00000	0.00000
8.3	100.00000	0.00000
8.4	50.00000	50.00000
8.5	69.23077	30.76923
8.6	81.81818	18.18182
8.7	60.00000	40.00000
8.8	55.55556	44.44444
8.9	70.00000	30.00000
9	75.00000	25.00000
9.5	85.71429	14.28571
9.6	100.00000	0.00000
9.7	0.00000	100.00000
9.8	90.00000	10.00000
9.9	62.50000	37.50000
10	100.00000	0.00000
10.5	100.00000	0.00000
10.6	75.00000	25.00000
10.7	100.00000	0.00000

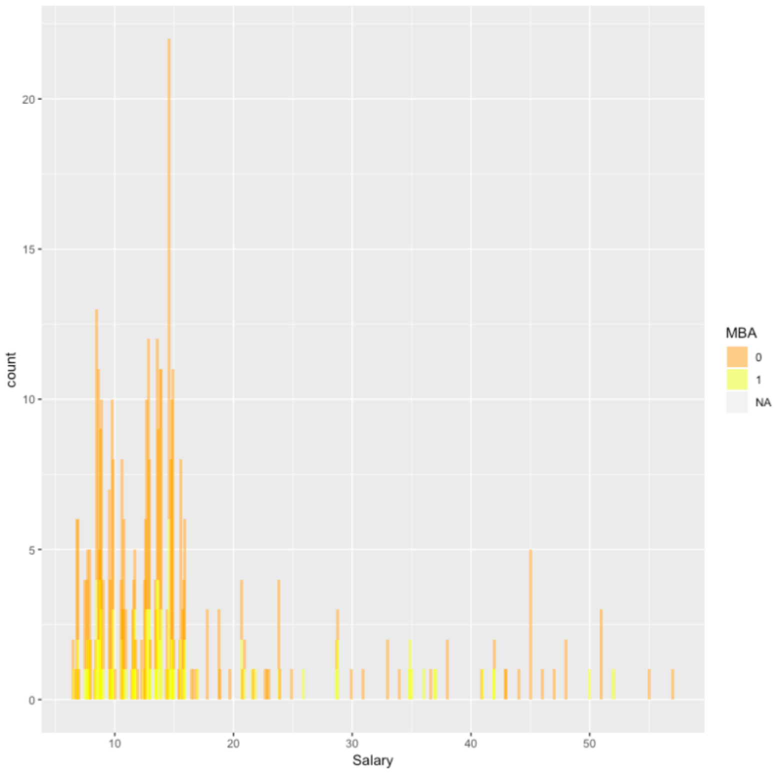
10.8	83.33333	16.66667
10.9	66.66667	33.33333
11.4	100.00000	0.00000
11.5	66.66667	33.33333
11.6	75.00000	25.00000
11.7	40.00000	60.00000
11.8	50.00000	50.00000
11.9	100.00000	0.00000
12.3	100.00000	0.00000
12.4	0.00000	100.00000
12.5	100.00000	0.00000
12.6	100.00000	0.00000
12.7	70.00000	30.00000
12.8	75.00000	25.00000
12.9	62.50000	37.50000
13	50.00000	50.00000
13.4	75.00000	25.00000
13.5	66.66667	33.33333
13.6	66.66667	33.33333
13.7	62.50000	37.50000
13.8	81.81818	18.18182

13.9	72.72727	27.27273
14.3	100.00000	0.00000
14.4	66.66667	33.33333
14.5	100.00000	0.00000
14.6	72.72727	27.27273
14.7	87.50000	12.50000
14.8	80.00000	20.00000
14.9	81.81818	18.18182
15	50.00000	50.00000
15.4	100.00000	0.00000
15.5	100.00000	0.00000
15.6	75.00000	25.00000
15.7	100.00000	0.00000
15.8	75.00000	25.00000
15.9	66.66667	33.33333
16.5	100.00000	0.00000
16.6	100.00000	0.00000
16.9	100.00000	0.00000
17	0.00000	100.00000
17.8	100.00000	0.00000
18.8	100.00000	0.00000

18.9	100.00000	0.00000
19.7	100.00000	0.00000
20.7	50.00000	50.00000
20.8	100.00000	0.00000
20.9	50.00000	50.00000
21.6	0.00000	100.00000
21.7	100.00000	0.00000
21.8	0.00000	100.00000
22.7	100.00000	0.00000
22.8	100.00000	0.00000
23	100.00000	0.00000
23.8	75.00000	25.00000
23.9	100.00000	0.00000
24.9	100.00000	0.00000
25.9	0.00000	100.00000
28.7	50.00000	50.00000
28.8	33.33333	66.66667
29.9	100.00000	0.00000
30.9	100.00000	0.00000
33	100.00000	0.00000
34	100.00000	0.00000

34.8	0.00000	100.00000
34.9	0.00000	100.00000
35	0.00000	100.00000
36	0.00000	100.00000
36.6	100.00000	0.00000
36.9	0.00000	100.00000
37	0.00000	100.00000
38	100.00000	0.00000
40.9	100.00000	0.00000
41	0.00000	100.00000
41.9	0.00000	100.00000
42	50.00000	50.00000
42.9	100.00000	0.00000
43	100.00000	0.00000
44	100.00000	0.00000
45	100.00000	0.00000
46	100.00000	0.00000
47	100.00000	0.00000
48	100.00000	0.00000
50	0.00000	100.00000
51	100.00000	0.00000

52	0.00000	100.00000
55	100.00000	0.00000
57	100.00000	0.00000



3) The most Challenging method in this was creating confusion Matrix. We can use table format to interpret data and can-do Exploratory Data Analysis.

Data Preparation for analysis:

```
library(caTools)
set.seed(1234)
spl = sample.split(actualcar$Car_or_nocar, SplitRatio = 0.75)
Cars_train <- subset(actualcar,spl == TRUE)
Cars_test  <-  subset(actualcar,spl == FALSE)
dim(Cars_train)
dim(Cars_test)
table(Cars_train$Car_or_nocar)
table(Cars_test$Car_or_nocar))
```

Modeling:

KNN Modeling

```
##########################################KNN MODEL BUILDING########################
scale = preProcess(Cars_train, method = "range")
train.norm.data = predict(scale, Cars_train)
test.norm.data = predict(scale, Cars_test)
knn_fit = train(Transport~., data = train.norm.data, method = "knn",
          trControl = trainControl(method = "cv", number = 3),
```

```
tuneLength = 10)
knn_fit
knn_fit$bestTune$k
Accu_knn=knn_fit$results$Accuracy
plot((knn_fit$results$Accuracy)*100~knn_fit$results$k, type='b',xlab
="# Neighbors", ylab="Accuracy")
########Perform Metrics#############################
predict = predict(knn_fit, data = train.norm.data, type = "raw")
confusionMatrix(predict,train.norm.data$license, positive = "1")
```

This is the best model as we can see that Accuracy, Sensitivity and Specificity is best for this model.

Naïve Bayes:

```
library(e1071)
NB = naiveBayes(x=train.norm.data[-c(1,5,9)],
y=train.norm.data$Transport)
#####################Perform
Metrics############################################
pred = predict(NB, newdata = train.norm.data)
confusionMatrix(pred,train.norm.data$Transport,positive="1")
pred = predict(NB, newdata = test.norm.data)
```

```
confusionMatrix(pred,test.norm.data$Transport,positive="1")
```

Naïve Bayes model is not recommended as we are getting false values not related to problem which will not help to assist solve the problem. The only way we can build this model is by doing model performance measures.

Logistic Modelling:

```
logit_model1 = glm(Car_or_nocar ~ ., data = actualcar,
                family = binomial(link = "logit"))
summary(logit_model1)
library(car)
vif(logit_model1)
logit_model2 = glm(Car_or_nocar ~ . -MBA -license,
                data = actualcar,
                family = binomial(link = "logit"))
summary(logit_model2)
vif(logit_model2)
library(lmtest)
lrtest(logit_model2)
library(pscl)
pR2(logit_model2)
1-(-12.5981253/-120.2966558)
exp(coef(logit_model2))
exp(coef(logit_model2))/(1-exp(coef(logit_model2)))
```

```
nrow(actualcar[actualcar$Car_or_nocar == 0,])/nrow(actualcar)

pred = predict(logit_model2, data=actualcar, type="response")

y_pred_num = ifelse(pred>0.5)

y_pred = factor(y_pred_num, levels=c(0,1))

y_actual = actualcar$Car_or_nocar

View(pred)

Cars_test$log.pred<-predict(logit_model2,Cars_test[1:9],
type="response")

table(Cars_test$Transport,Cars_test$log.pred>1)
```

This model is also recommended as the factors are not affecting the main model.

Bagging a

```
install.packages('gbm')

library(gbm)

install.packages('xgboost')

library(xgboost)

library(caret)
```

```
library(ipred)

library(rpart)

Cars.bagging <- bagging(Car_or_nocar ~.,

data = actualcar,

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

actualcar$Car_or_nocar <- predict(Cars.bagging, actualcar)

#actualcar$Car_or_nocar <- ifelse(actualcar$car_or_nocar<0.5,0,1)

####confusionMatrix(data=factor(actualcar$car_or_nocar),

######          reference=factor(actualcar$car_or_nocar),

################             positive='1')

table(actualcar$Car_or_nocar,actualcar$Car_or_nocar)


#########################Boosting########################
######################

library(xgboost)

gbm.fit <- gbm(

  formula = Transport~ .,

  distribution = "bernoulli",

  data = Cars_train,

  n.trees = 10000,

  interaction.depth = 1,

  shrinkage = 0.001,
```

```
  cv.folds = 5,
  n.cores = NULL,
  verbose = FALSE
)
```

Prediction Summary			
Techniques	Accuracy	Sensitivity	Specificity
Logistic Regression	99.67	33.56	98.43
KNN	86.94	45.31	97.6
NAÏVE BAYES	99.36	100	100
BAGGING	99.43	48.33	99.96
xgBoosting	99.38	75	98.27

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

1. Naïve Bayes Model can be used only for Numerical Variables
2. KNN Model works the best in this scenario.

3. From the above sample I can conclude those with high work experience and MBA jobs use car as their mode of transport.
4. While Engineers and others do use cars and I also observed that lot of individuals use two wheelers instead of cars as their mode of transport.
5. Lastly I conclude by that personal transport and public transport both play a major role.