Cars!!

Amit KULKARNI

CONTENT

EXPLORATORY DATA ANALYSIS:

To create various models and provide the ED Analysis below packages have been used

i. Ggplot2
ii. SDMTools
iii. pROC
iv. Hmisc
v. Tidyverse
vi. Class
vii. Dplyr
viii. Caret
ix. Plotrix
x. Rattle
xi. Data.table
xii. Scales
xiii. Ineq
xiv. MASS
xv. E1071
xvi. caTools
xvii. xgboost
xviii. ipred
xix. rpart

QUESTION I

Commuter's data pertaining to their

i. Age,
ii. If they are Engineer or MBA
iii. Gender,
iv. Salary,
v. Do they possess a License and
vi. Mode of transport taken

provided.

There are a total of 418 data points that are being observed.

Summary of the data provided below

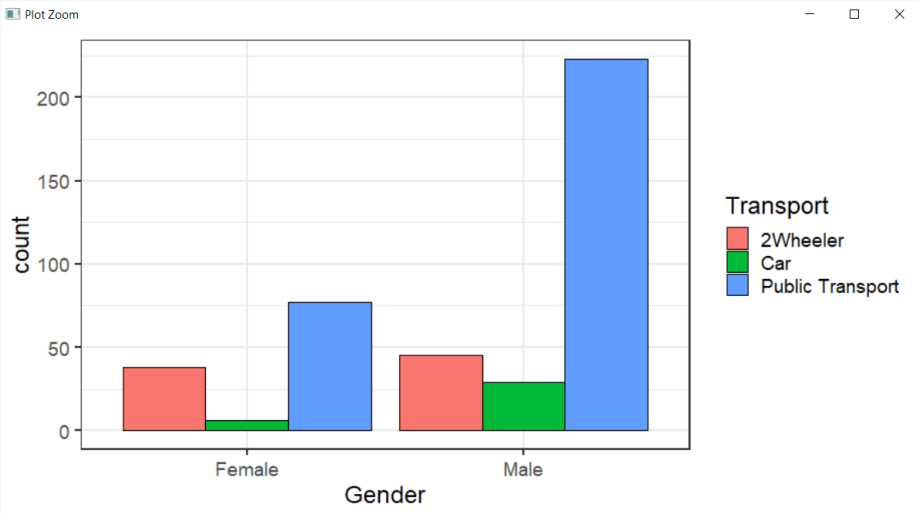
Age		Gender		Engineer		MBA		Work.Exp	
Min.	18.00	Female	121	Min.	0.00	Min.	0.00	Min.	0.00
1st Qu.	25.00	Male	297	1st Qu.	0.25	1st Qu.	0.00	1st Qu.	3.00
Median	27.00			Median	1.00	Median	0.00	Median	5.00
Mean	27.33			Mean	0.75	Mean	0.26	Mean	5.87
3rd Qu.	29.00			3rd Qu.	1.00	3rd Qu.	1.00	3rd Qu.	8.00
Max.	43.00			Max.	1.00	Max.	1.00	Max.	24.00

Salary		Distance		License		Transport	
Min.	6.50	Min.	3.20	Min.	0.00	2Wheeler	83
1st Qu.	9.63	1st Qu.	8.60	1st Qu.	0.00	Car	35
Median	13.00	Median	10.90	Median	0.00	Public Transport	300
Mean	15.42	Mean	11.29	Mean	0.20		
3rd Qu.	14.90	3rd Qu.	13.57	3rd Qu.	0.00		
Max.	57.00	Max.	23.40	Max.	1.00		

QII. VISUALIZATION AND INFERENCES

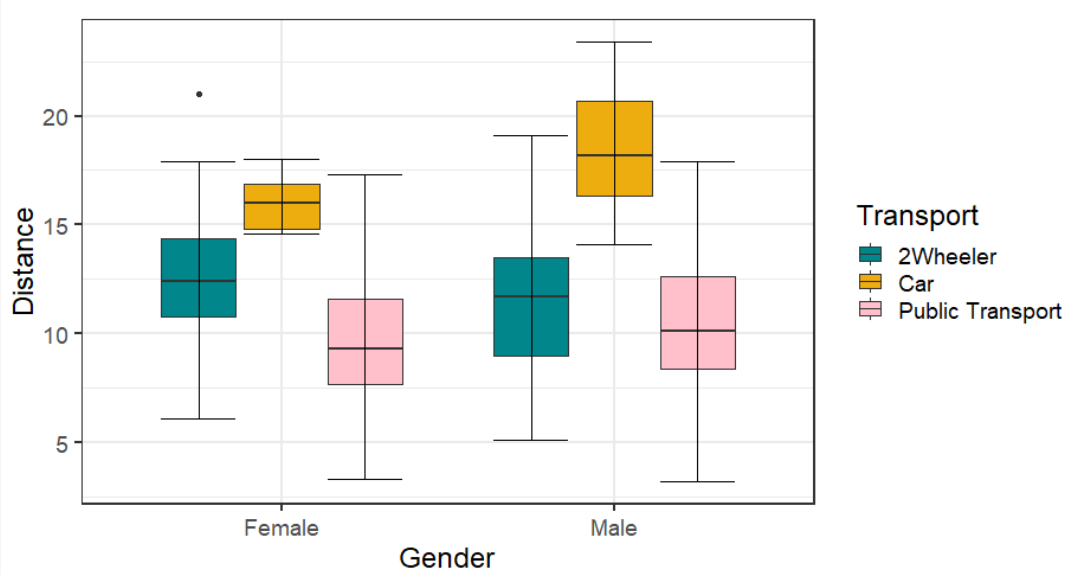
Below Visualizations have been created and observations done basis the same

I. Split of the various transport modes used at Gender Level



The summary also demonstrated that for every 3 males, we had 1 female record with us. Thus the number of users for male was apparently high. However it can be observed that the usage of all the modes of transport share similar proportion. Public Transport being used the most, followed by Two Wheelers and then by Car

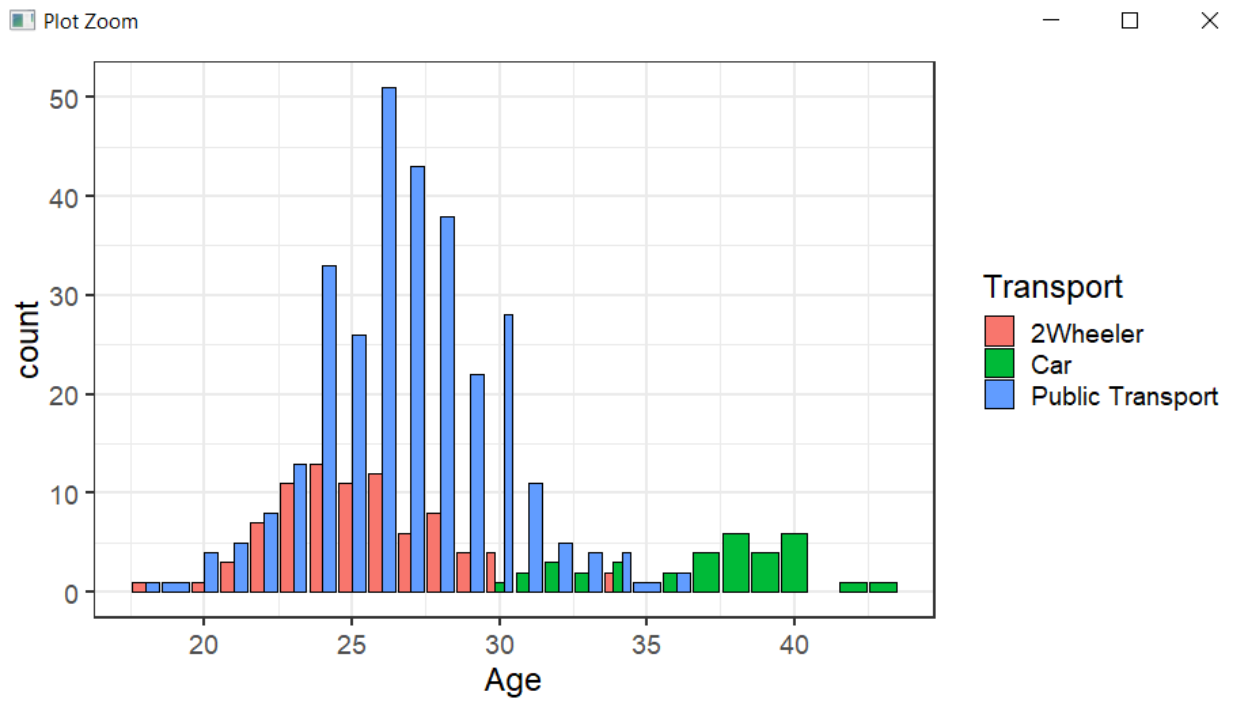
II. Box Plot analysis for distance and Gender shown below



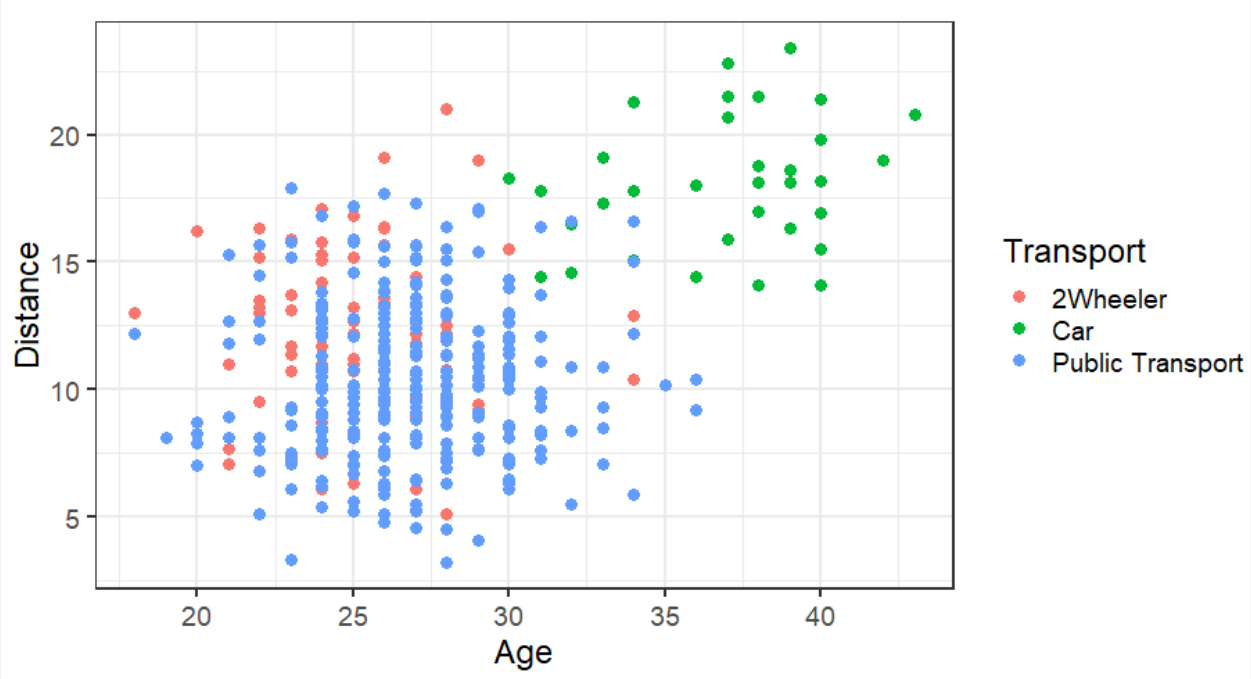
The boxplot clearly demonstrates both Females and Males

i. have preferred public transport for distances up to 10 kms,
ii. Two Wheelers up to 15 kms and
iii. Anything greater than 15 kms, they have preferred cars

III. Below bar plot demonstrates that Age is one of the driving factors for usage of car as commuting since it can be observed that the data points for car usages are heavily concentrated towards 35 years and above



IV. Adding an additional parameter to the Age and mode of transport i.e. Distance, we observe elder commuters travelling longer distances prefer cars. It can be observed that there is a heavy concentration of data points for usage of cars for commuting with commuters who are elderly and travel longer distance as depicted in the scatter plot



LINEAR REGRESSION TO IDENTIFY SIGNIFICANT VARIABLE

Below code has been executed with all the variables except Engineer and MBA

```
linreg<- lm(cars$TransportCar~cars$Salary+cars$Age+cars$Distance+cars$Engineer+cars$MBA+cars$license, data=cars)
summary(linreg)
```

Below is the result of the code execution

```
Coefficients:
              Estimate Std. Error t value
(Intercept)  -0.355613   0.082674  -4.301
cars$Salary   0.019875   0.001650  12.042
cars$Age     -0.001426   0.003546  -0.402
cars$Distance 0.013910   0.002341   5.941
cars$Engineer 0.007451   0.017458   0.427
cars$MBA     -0.020662   0.017213  -1.200
cars$license  0.071973   0.021159   3.402
             Pr(>|t|)
(Intercept)  2.12e-05 ***
cars$Salary   < 2e-16 ***
cars$Age     0.687676
cars$Distance 6.03e-09 ***
cars$Engineer 0.669742
cars$MBA     0.230696
cars$license  0.000736 ***
---
Signif. codes:
  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
  ' ' 1

Residual standard error: 0.1538 on 411 degrees of freedom
Multiple R-squared:  0.6968,    Adjusted R-squared:  0.6923
F-statistic: 157.4 on 6 and 411 DF,  p-value: < 2.2e-16
```

All the variables put together in the regression analysis shows that ~70% of Y i.e. commuting with Car

At individual level it can be seen that Salary is significant variable driving the commuters to use car followed by distance and license

```
cor(cars$TransportCar, cars$Distance )
cor(cars$TransportCar, cars$Salary )
cor(cars$TransportCar, cars$license )
```

```
> cor(cars$TransportCar, cars$Distance )
[1] 0.5392434
> cor(cars$TransportCar, cars$Salary )
[1] 0.8107034
> cor(cars$TransportCar, cars$license )
[1] 0.4695844
```

CHALLENGES WITH THE DATASET

1. There was only 1 data point missing which needed imputation. The record could have been deleted or a value to be imputed. It could have been chosen to be ignored since the variable in itself was not significant
2. With 418 observations, the data provided is too little. More observations would have helped create efficient models
3. Variables like city, working shift, if transport provided by the org etc. could have been added if possible

The Models can be still developed since significant variables are available to build the models

But one major problem with the dataset is that the predictor variable is categorical and it can't be predicted since it is part of one entire variable.

This challenge will have to be dealt with by converting the Transport variable to dummy variables so that the Two Wheelers, Cars and Public Transport can be separated and Cars can be used as the predicted class and build our models around it.

DATA PREPARATION:

Various models need the data in various forms and shapes.

1. Typically nearly all the models need only numeric values to build their models.

In this case Categorical data has to be converted to dummy variables or given numerical value of 0,1,2 etc.

Below codes can be executed to create the dummy variables

```
Transport.d = model.matrix(~Transport -1, data=bagdata)
Gender.d = model.matrix(~Gender -1, data=bagdata)
bagdata = data.frame(bagdata, Gender.d, Transport.d)
```

These codes convert dummy variables with separate columns like shown below

Transport	Transport2Wheeler	TransportCar	TransportPublic.Transport
2Wheeler	1	0	0

These variables are then used to prepare the models.

2. Models do not work when there are NA values so it is good to check if there are any blank or NA values.

Once found you can either impute, delete or ignore the missing data.

```
sum(is.na(NB.Train))
```

```
> sum(is.na(NB.Train))
[1] 0
```

3. Various variables are measured in different units and may not be comparable outright. It is a good idea to have the data scaled for model building. This is typically required in kNN model building. It can be done as below

```
normdata=scale(knnsclddata)
View(knnsclddata)
usuable.data=cbind(data1[,13], normdata)
```

MODELLING

KNN

Steps pertaining to data preparations mentioned above. The model building is done as below.

To build a KNN model below packages are necessary

```
library(tidyverse)
library(class)
library(dplyr)
library(caret)
library(plotrix)
library(rattle)
library(data.table)
library(scales)
library(ineq)
library(MASS)
```

Below codes executed for

1. Calling dataset and defining objects
2. Converting dummy variables for categorical data points
3. Subsetting data by excluding unwanted variables
4. Scaling the variables
5. Setting the seed and splitting the data in 70:30 ratio for Train and test data respectively
6. Creating the model
7. Predicting the values on the test data
8. Calculating the accuracy

```
setwd("C:/Users/Amit Kulkarni/Documents/R Programming/Machine Learning/Project 7")
knndata= read.csv("cars.csv")
str(knndata)
Transport.d = model.matrix(~Transport -1, data=knndata)
Gender.d = model.matrix(~Gender -1, data=knndata)
data1 = data.frame(knndata, Gender.d, Transport.d)
knnscaledate=data1[,-13]
knnscaledate=data1[,-2:-4,-9]
knnsclddata= knnscaledate[, -6]
normdata=scale(knnsclddata)
usuable.data=cbind(data1[,13], normdata)
usuable.data = as.data.frame(usuable.data)
library(caTools)
spl=sample.split(usuable.data$TransportCar, SplitRatio = 0.7)
knntrain=subset(usuable.data, spl==T)
knntest=subset(usuable.data, spl==F)
library(class)
pred3=knn(knntrain[,-10], knntest[,-10], knntrain[,10], k=19)
table.knn=table(knntest[,10], pred3)
sum(diag(table.knn))/sum(table.knn)
summary(table.knn)
table.knn
```

Key outputs of the code executions

Accuracy of the model is

```
> sum(diag(table.knn))/sum(table.knn)
[1] 0.984127
```

NAÏVE BAYES

Steps pertaining to data preparations mentioned above. The model building is done as below.

To build a NAÏVE BAYES model below packages are necessary

```
library(tidyverse)
library(class)
library(dplyr)
library(caret)
library(plotrix)
library(rattle)
library(data.table)
library(scales)
library(ineq)
library(MASS)
```

Below codes executed for

1. Calling dataset and defining objects
2. Converting dummy variables for categorical data points
3. Subsetting data by excluding unwanted variables
4. Scaling the variables
5. Setting the seed and splitting the data in 70:30 ratio for Train and test data respectively
6. Creating the model
7. Predicting the values on the test data
8. Calculating the accuracy

```
setwd("C:/Users/Amit Kulkarni/Documents/R Programming/Machine Learning/Proje
NBdata= read.csv("cars.csv")
str(NBdata)
View(NBdata)
Transport.d = model.matrix(~Transport -1, data=NBdata)
Gender.d = model.matrix(~Gender -1, data=NBdata)
data1 = data.frame(NBdata, Gender.d, Transport.d)
names(data1)
dim(data1)
#x = data1 %>% dplyr::select(-Gender, -Transport)
#data1.scaled = scale(x)
#data1.scaled = cbind(data1[2], data1.scaled)
set.seed(111)
trainIndex <- sample(c(1:nrow(data1)), round(nrow(data1) * 0.7,0), replace =
NB.Train = data1[trainIndex,]
NB.Test = data1[-trainIndex,]
View(NB.Train)
dim(NB.Test)
NB.Train$TransportCar = as.factor(NB.Train$TransportCar)
NB.Test$TransportCar = as.factor(NB.Test$TransportCar)
#install.packages("e1071")
library(e1071)
NB = naiveBayes(x = NB.Train, y = NB.Train$TransportCar)
ypred.NB = predict(NB, newdata = NB.Test)
ypred.NB
tab.NB = table(NB.Test[,12], ypred.NB)
tab.NB
accuracy.NB = sum(diag(tab.NB))/sum(tab.NB)
accuracy.NB
```

Key Outputs of the code execution

Confusion matrix

```
> tab.NB = table(NB.Test[,12], ypred.NB)
> tab.NB
   ypred.NB
     0  1
  0 84 13
  1 28  0
```

Considering the above matrix: The model has been 67.2% times accurate to predict who will not use car for commuting but if we have to consider the True Positive, the model has not been able to predict even 1 case correctly and has predicted 28 cases correctly. Though the overall accuracy is 67.20% of the model but it is not able to predict the true positive values

```
> 84/(84+13+28)
[1] 0.672
```

LOGISTIC REGRESSION

Steps pertaining to data preparations mentioned above. The model building is done as below.

To build a Logistic Regression model below packages are necessary

```
#install.packages(c("SDMTools", "pROC", "Hmisc"))
library(SDMTools)
library(pROC)
library(Hmisc)
```

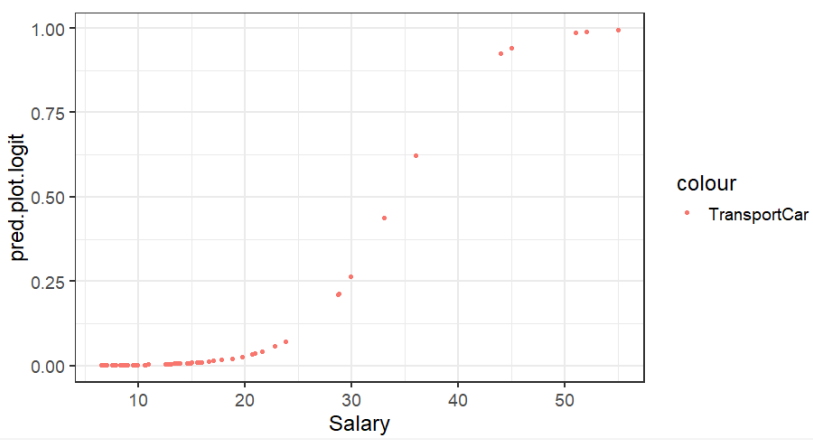
Below codes executed for

1. Calling dataset and defining objects
2. Converting dummy variables for categorical data points
3. Subsetting data by excluding unwanted variables
4. Scaling the variables
5. Setting the seed and splitting the data in 70:30 ratio for Train and test data respectively
6. Creating the model
7. Predicting the values on the test data
8. Calculating the accuracy

```
setwd("C:/Users/Amit Kulkarni/Documents/R Programming/Machine Learning/Project 7")
data = read.csv("cars.csv")
Transport.d = model.matrix(~Transport -1, data=data)
data1 = data.frame(data, Transport.d)
Gender.d = model.matrix(~Gender -1, data=data)
data1 = data.frame(data, Gender.d, Transport.d)
#data.frame
set.seed(111)
train.logit <- sample(c(1:nrow(data1)), round(nrow(data1) * 0.7,0), replace = FALSE)
LR.Train<-data1[train.logit,]
LR.Test<-data1[-train.logit,]
#Fit the Sigmoid function
logit.1<-TransportCar~Salary
logit.plot<- glm(logit.1, data=LR.Train, family=binomial())
summary(logit.plot)
pred.plot.logit <- predict.glm(logit.plot, newdata=LR.Test, type="response")
pred.plot.logit
LR.Test$pred<-pred.plot.logit
predprob<-exp(-8.51757+0.25059*80)/(1+exp(-8.51757+0.25059*80))
predprob
qplot(Salary, pred.plot.logit, data=LR.Test,color="TransportCar")
#All Variables
LRModel = glm(TransportCar ~ Age+Salary+Distance+Work.Exp, data=LR.Train, family= binomial)
summary(LRModel)
predTest<-predict(LRModel, newdata=LR.Test, type="response")
table(LR.Test$TransportCar, predTest>0.5)
(111+12)/nrow(na.omit(LR.Test))
library(ROCR)
ROCRpred=prediction(predTest, LR.Test$TransportCar)
as.numeric(performance(ROCRpred,"auc")@y.values)
perf= performance(ROCRpred, "tpr","fpr")
plot(perf)
```

Key outputs of the code execution

Fitting a Sigmoid Function for 1 variable



Confusion Matrix, Overall accuracy scores and model evaluation using the auc method

```
> table(LR.Test$TransportCar, predTest>0.5)

    FALSE TRUE
  0   111    2
  1     0   12
> (111+12)/nrow(na.omit(LR.Test))
[1] 0.984
> library(ROCR)
> ROCRpred=prediction(predTest, LR.Test$TransportCar)
> as.numeric(performance(ROCRpred,"auc")@y.values)
[1] 0.9992625
```

The model is overall accurate at98.40% however to predict the True Positives it is scoring 100% scores.

The validation of the model also yields 99.93%

XGBOOSTING

Steps pertaining to data preparations mentioned above. The model building is done as below.

To build a XGBOOST model we need xgboost package

Below codes executed for

1. Calling dataset and defining objects
2. Converting dummy variables for categorical data points
3. Subsetting data by excluding unwanted variables
4. Scaling the variables
5. Setting the seed and splitting the data in 70:30 ratio for Train and test data respectively
6. Creating the model
7. Predicting the values on the test data
8. Calculating the accuracy

Code for Execution

```
#Boositng
setwd("C:/Users/Amit Kulkarni/Documents/R Programming/Machine Learning/Project 7")
xgbdata= read.csv("cars.csv")
Transport.d = model.matrix(~Transport -1, data=xgbdata)
Gender.d = model.matrix(~Gender -1, data=xgbdata)
xgbdata = data.frame(xgbdata, Gender.d, Transport.d)
View(xgbdata)
xgbdata=xgbdata[,-2:-4,-9]
xgbdata=xgbdata[,-6]
#library(caTools)
set.seed(123)
split=sample.split(xgbdata$TransportCar, SplitRatio = 0.80)
xgbtrain=subset(xgbdata, split==T)
xgbtest=subset(xgbdata, split==F)
View(xgbtrain)
dim(xgbtest)
sum(xgbtrain$TransportCar)
sum(xgbtest$TransportCar)
#install.packages("xgboost")
#library(xgboost)
classifier = xgboost(data = as.matrix(xgbtrain[,-9]), label=xgbtrain$TransportCar, nrounds=1
y_pred = predict(classifier, newdata = as.matrix(xgbtest[,-9]))
y_pred=(y_pred>=0.5)
cm=table(xgbtest[,9], y_pred)
cm
```

Key outputs of the model
Confusion matrix

```
> cm
   y_pred
     FALSE TRUE
  0     77    0
  1      0    7
```

Here the overall and the True Positive accuracy rate is 100%

BAGGING

Steps pertaining to data preparations mentioned above. The model building is done as below.

To build a BAGGING model we need ipred and caret package

Below codes executed for

1. Calling dataset and defining objects
2. Converting dummy variables for categorical data points
3. Subsetting data by excluding unwanted variables
4. Scaling the variables
5. Setting the seed and splitting the data in 70:30 ratio for Train and test data respectively
6. Creating the model
7. Predicting the values on the test data
8. Calculating the accuracy

Code execution

```
setwd("C:/Users/Amit Kulkarni/Documents/R Programming/packages")
#install.packages("ipred")
#library(ipred)
#library(rpart)
setwd("C:/Users/Amit Kulkarni/Documents/R Programming/Machine Learning/Project 7")
bagdata=read.csv("cars.csv")
Transport.d = model.matrix(~Transport -1, data=bagdata)
Gender.d = model.matrix(~Gender -1, data=bagdata)
bagdata = data.frame(bagdata, Gender.d, Transport.d)
bagdata= bagdata[,-2:-4,-9]
bagdata = bagdata[,-6]
View(bagdata)
library(caTools)
set.seed(123)
split=sample.split(bagdata$TransportCar, SplitRatio = 0.80)
bagtrain=subset(bagdata, split==T)
bagtest=subset(bagdata, split==F)
?bagging
Carbagging<- bagging(bagtrain$TransportCar~ Age+Salary+Work.Exp+license, data=bagtrain,
                     control=rpart.control(maxdepth = 5, minsplit = 15))
bagtest$pred.class<-predict(Carbagging, bagtest)
table(bagtest$TransportCar, bagtest$pred.class>0.75)
6/7
```

Key Outputs of the model
Confusion Matrix and Accuracy

```
   FALSE TRUE
0     77    0
1      1    6
> 6/7
[1] 0.8571429
```

The True Positive rate in this case is 85.71% and the overall accuracy is 91.67%

Summary accuracy of all models and their comparison to bagging and boosting

Model	Accuracy	True Positive
KNN	**98.41%**	-
Naïve Bayes	**67.20%**	**0.00%**
Logistic Regression	**98.40%**	**100.00%**
Boosting	**100.00%**	**100.00%**
Bagging	**91.67%**	**85.71%**

Boosting has been able to achieve 100% on the overall accuracy and the True Positive accuracy as well when compared to other models whereas even Logistic Regression also has fared well in terms of predicting the test dataset

Summary: Various predicting models have been used to predict about the commuters choice of using cars to commute Logistic model and xgboost have been fairly good in predicting the outcome close to True.
The study can be done extensive by adding more variables if required and can be studied by an org in case if they wish to extend any commuting facility etc.