New Car Case

Rajeev

10/30/2020

Cars Case Study

This project requires us to understand what mode of transport employees prefers to commute to their office. We need to predict whether or not an employee will use Car as a mode of transport.

1. Project Objective

- To predict whether or not an employee will use Car as a mode of transport, we need to investigate which variables are significant predictors behind the decision.
- Identify the challenging aspect to this problem & what methods will be used to deal with it.
- Prepare the data to create multiple models to explore which model performs the best (by using appropriate performance metrics).
- Summarize the findings.

2.Data Dictionary

Load Packages

```
library(caTools) # Split Data into Test and Train Set
library(caret) # for confusion matrix function
library(randomForest) # to build a random forest model
library(rpart) # to build a decision model
library(rpart.plot) # to plot decision tree model
library(rattle)
library(xgboost) # to build a XG Boost model
library(DMwR) # for SMOTE
library(naivebayes) # for implementation of the Naive Bayes
library(e1071) # to train SVM & obtain predictions from the model
library(mlr) # for a generic, object-oriented, and extensible framework
library(gbm) #For power-users with many variables
library(car) # use for multicollinearity test (i.e. Variance Inflation Factor(VIF))
library(MASS) # for step AIC
library(ggplot2) # use for visualization
library(grid) # for the primitive graphical functions
library(gridExtra) # To plot multiple ggplot graphs in a grid
library(corrplot) # for correlation plot
library(e1071) # to build a naive bayes model
library(ROCR) # To plot ROC-AUC curve
```

```r
library(InformationValue) # for Concordance-Discordance
library(class) # to build a KNN model
library(knitr) # Necessary to generate sourcecodes from a .Rmd File
```

3. Import Data

4. Exploratory Data Analysis

Check the dimension of the dataset

```r
dim(cars)
```

```
## [1] 418   9
```

Sanity Checks

```r
# Look at the first and last few rows to ensure that the data is read in properly
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
```

```r
tail(cars)
```

```
##     Age Gender Engineer MBA Work.Exp Salary Distance license        Transport
## 413  29 Female        1   0        6   14.9     17.0       0 Public Transport
## 414  29   Male        1   1        8   13.9     17.1       0 Public Transport
## 415  25   Male        1   0        3    9.9     17.2       0 Public Transport
## 416  27 Female        0   0        4   13.9     17.3       0 Public Transport
## 417  26   Male        1   1        2    9.9     17.7       0 Public Transport
## 418  23   Male        0   0        3    9.9     17.9       0 Public Transport
```

Check the structure of dataset

```r
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 ...
```

Observations: + Data set has 418 rows & 9 columns + Gender & Transport are 2 character variables. + Age, Work Experience, Salary, Distance are numerical variables + Engineer, MBA & License are categorical variables

Get Summary of the dataset

```
summary(cars)
```

```
##       Age           Gender       Engineer          MBA        
##  Min.   :18.00   Female:121   Min.   :0.0000   Min.   :0.0000  
##  1st Qu.:25.00   Male  :297   1st Qu.:0.2500   1st Qu.:0.0000  
##  Median :27.00                Median :1.0000   Median :0.0000  
##  Mean   :27.33                Mean   :0.7488   Mean   :0.2614  
##  3rd Qu.:29.00                3rd Qu.:1.0000   3rd Qu.:1.0000  
##  Max.   :43.00                Max.   :1.0000   Max.   :1.0000  
##                                                NA's   :1       
##     Work.Exp          Salary          Distance        license      
##  Min.   : 0.000   Min.   : 6.500   Min.   : 3.20   Min.   :0.0000  
##  1st Qu.: 3.000   1st Qu.: 9.625   1st Qu.: 8.60   1st Qu.:0.0000  
##  Median : 5.000   Median :13.000   Median :10.90   Median :0.0000  
##  Mean   : 5.873   Mean   :15.418   Mean   :11.29   Mean   :0.2033  
##  3rd Qu.: 8.000   3rd Qu.:14.900   3rd Qu.:13.57   3rd Qu.:0.0000  
##  Max.   :24.000   Max.   :57.000   Max.   :23.40   Max.   :1.0000  
##                                                                    
##             Transport  
##  2Wheeler        : 83  
##  Car             : 35  
##  Public Transport:300  
##                        
##                        
##                        
##                        
```

```
colnames(cars)
```

```
## [1] "Age"       "Gender"    "Engineer"  "MBA"       "Work.Exp"  "Salary"   
## [7] "Distance"  "license"   "Transport"
```

Observations: + AGE = Range from 18 to 43. There seems to be outliers here as the 3rd Quartile is at 29, while mean & median is at 27 + GENDER = of the 418 people in this data 71% are Male + ENGINEER = Almost 75% of people in data are Engineers + MBA = 26% are MBAs. There is an NA which we will deal with + WORK EXP = Ranges is from 0 to 24. There seems to be outliers as max experience is 24 while the 3rd Quartile shows 8. Mean is 5 while median is around 5.9. + SALARY = The range is from 6.5 to 57 with 3rd Quartile at around 15, which means we have outliers in salary. + DISTANCE = Distance traveled range from 3.2km to 23.40km. Mean 11.3km & Median 11km arent very far apart. There are outliers as 3rd Quartile shows 13.57km but max is 23.40 + Close to 80% of people in the data do not possess a license. + Majority of the people i.e. 71% use public transport. Around 20% use a 2Wheeler and around 8% travel using a car. + The column names seem good to go and don't need any treatment. + No typo found in the data.

Missing value treatment

```
colSums(is.na(cars))
```

```
##       Age    Gender  Engineer       MBA  Work.Exp    Salary  Distance   license
##         0         0         0         1         0         0         0         0
## Transport
##         0
```

```
cars$MBA[is.na(cars$MBA)] = mode(cars$MBA)
colSums(is.na(cars))
```

```
##       Age    Gender  Engineer       MBA  Work.Exp    Salary  Distance   license
##         0         0         0         0         0         0         0         0
## Transport
##         0
```

Observations: + The missing value in MBA is treated using the mode

Univariate analysis

```
#Distribution of the dependent variable
prop.table(table(cars$Transport))*100
```

```
##
##         2Wheeler              Car Public Transport
##        19.856459         8.373206        71.770335
```

Observations: +Majority of the people i.e. 71% use public transport. Around 20% use a 2Wheeler and around 8% travel using a car.

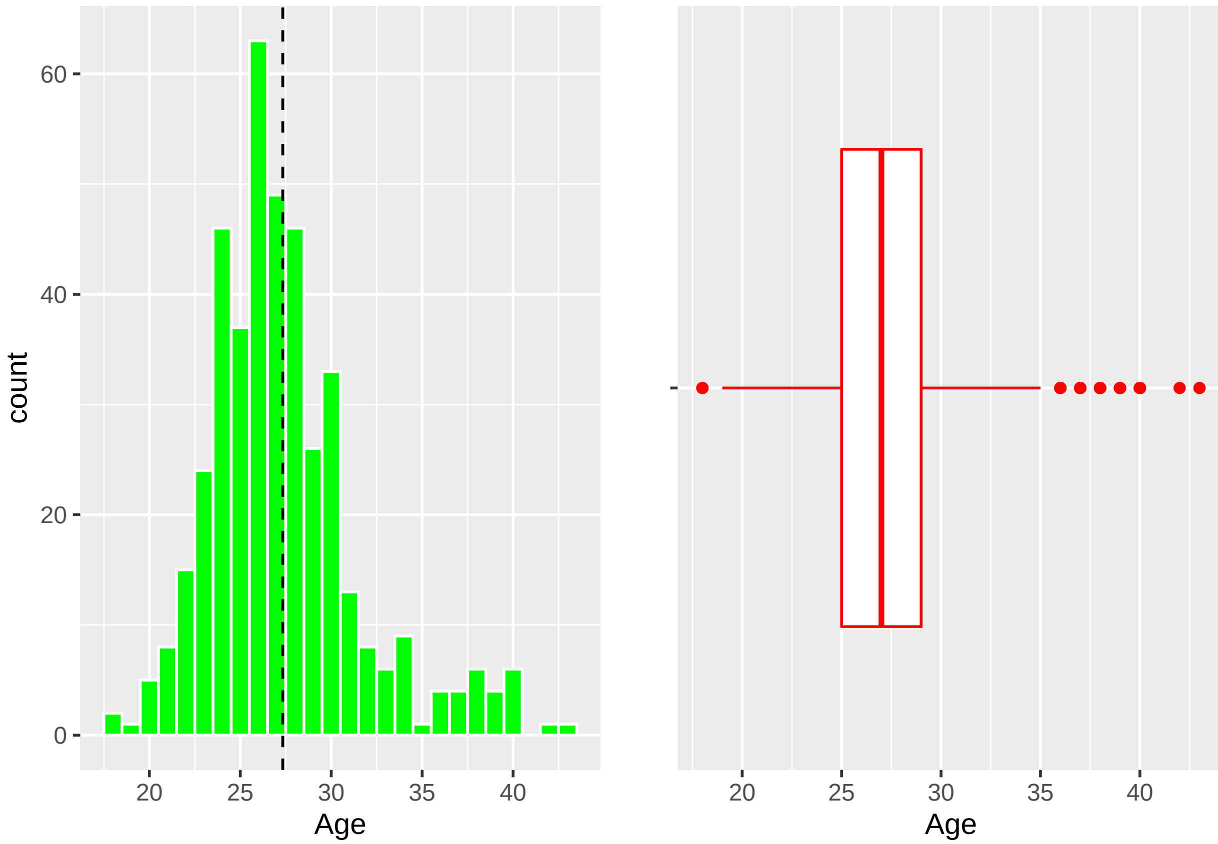
```
plot_histogram_n_boxplot = function(variable, variableNameString, binw){
  h = ggplot(data = cars, aes(x= variable))+
    labs(x = variableNameString,y ='count')+
    geom_histogram(fill = 'green',col = 'white',binwidth = binw)+
    geom_vline(aes(xintercept=mean(variable)),
              color="black", linetype="dashed", size=0.5)
  b = ggplot(data = cars, aes('',variable))+
    geom_boxplot(outlier.colour = 'red',col = 'red',outlier.shape = 19)+
    labs(x = '',y = variableNameString)+ coord_flip()
  grid.arrange(h,b,ncol = 2)
}
```

Function to draw histogram and boxplot of numerical variables using ggplot

Visualize properties of all categorical variables

a. Observations on Age

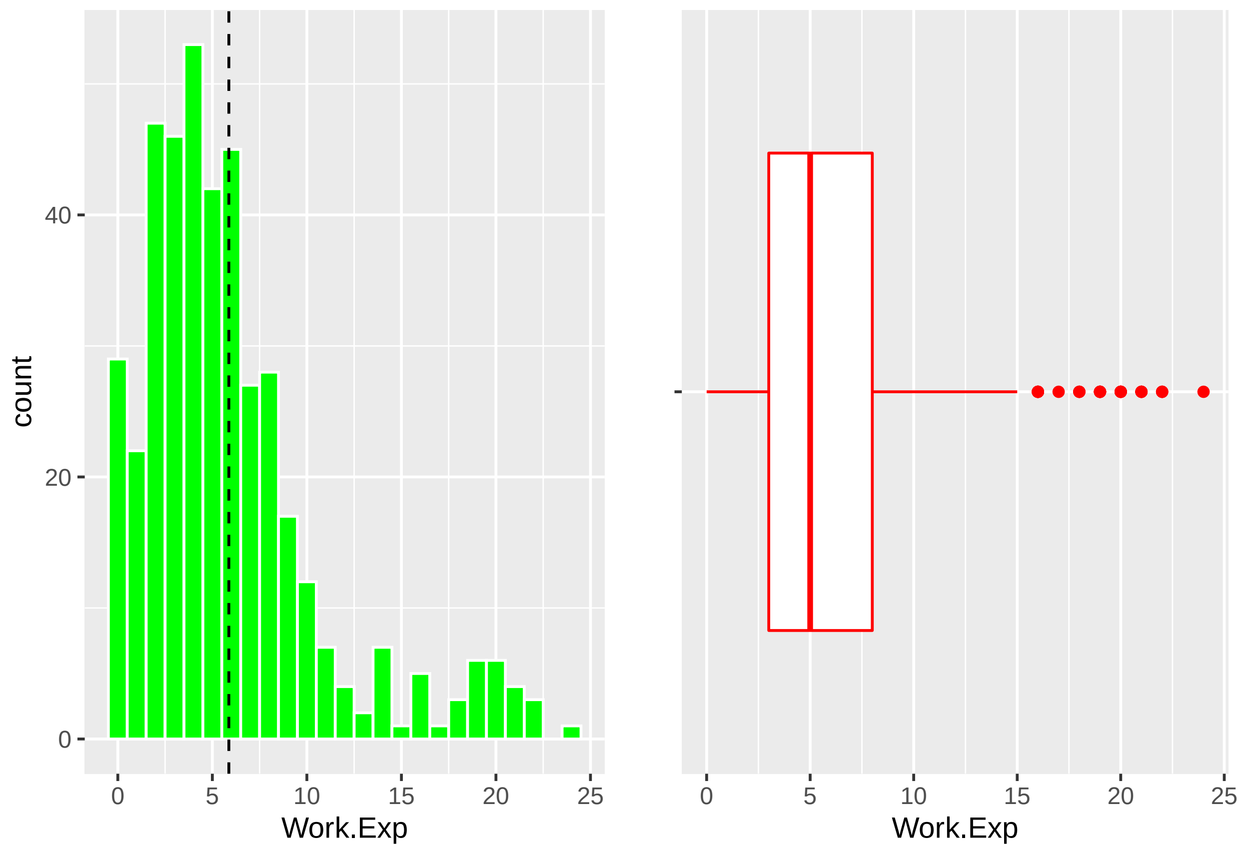
```
plot_histogram_n_boxplot(cars$Age,"Age",1)
```



Observations: + The Age has a normal curve with a spread out range. Also, it has many outliers beyond 35. + Outliers are predominantly in the range between 35 & 43. There is also an outlier at 18.

b. Observations on Work Experience

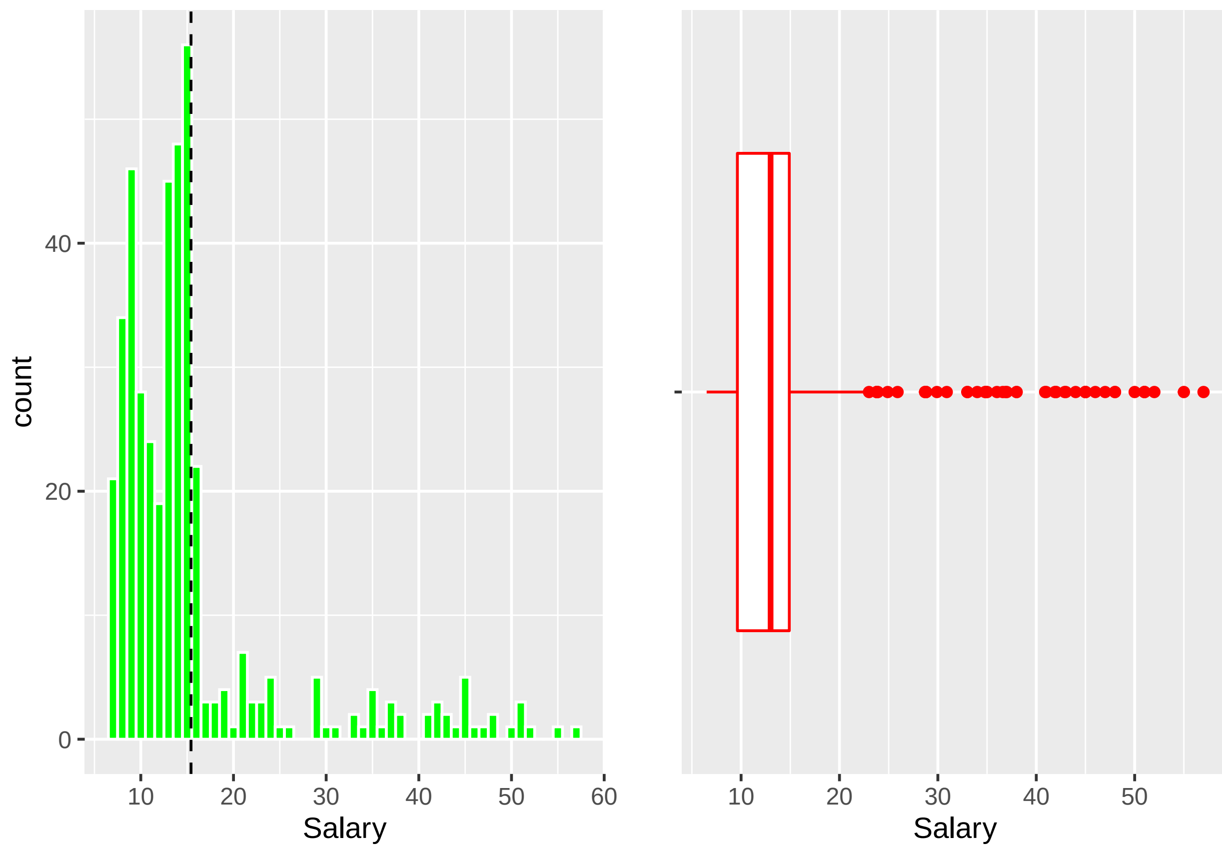
```r
plot_histogram_n_boxplot(cars$Work.Exp,"Work.Exp",1)
```



Observations: + The curve is right skewed with range between 3 & 8. + Quite a few outliers beyond beyond 15 upto 24.

c. Observations on Salary

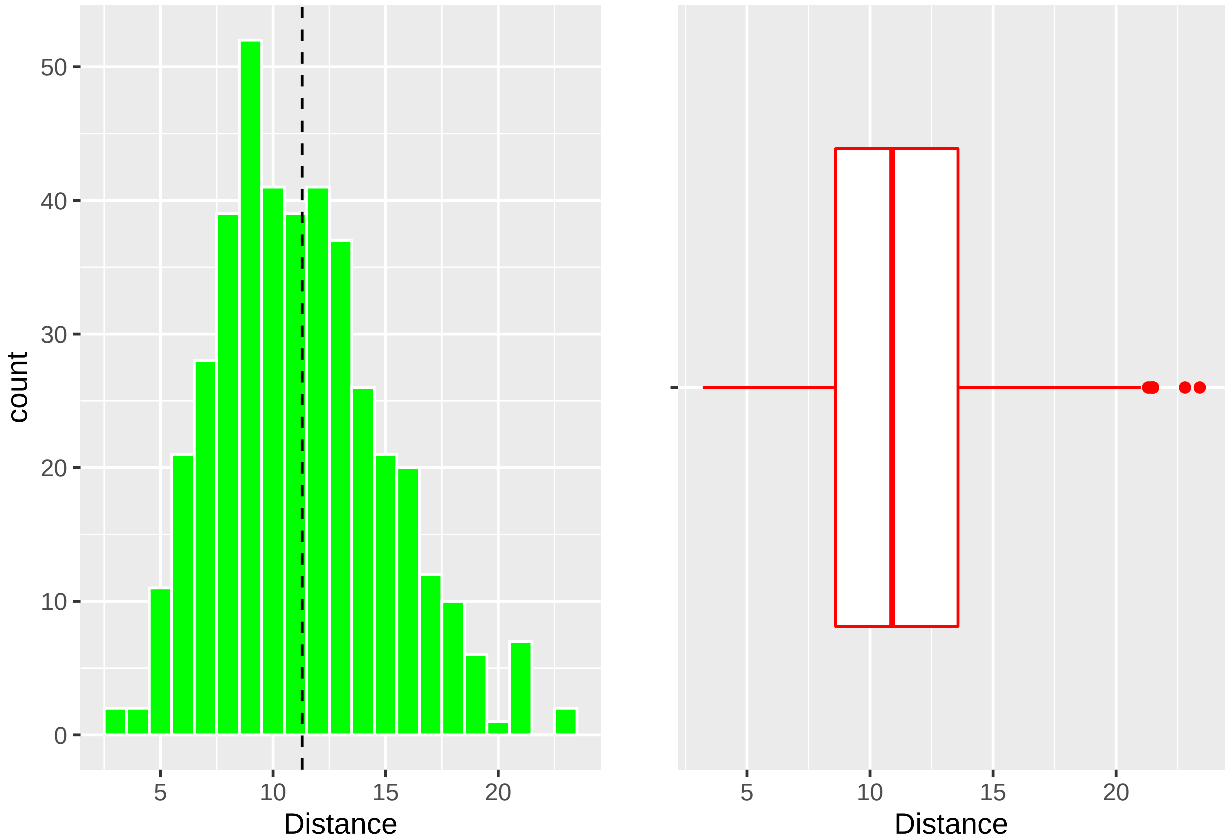
```r
plot_histogram_n_boxplot(cars$Salary,"Salary",1)
```



Observations: + The curve is right skewed with concentrayion between 10 & 15. + the range is spread out with huge amount of outliers beyond 20 right upto 57.

d. Observations on Distance

```
plot_histogram_n_boxplot(cars$Distance,"Distance",1)
```



Observations: + Distance has a normal curve with rance between 8 & 14. + Some outliers beyond 20.

```
unipar = theme(legend.position = "none") +
  theme(axis.text = element_text(size = 10),
        axis.title = element_text(size = 11),
        title = element_text(size = 13, face = "bold"))

# Define color brewer
col1 = "Set2"
```

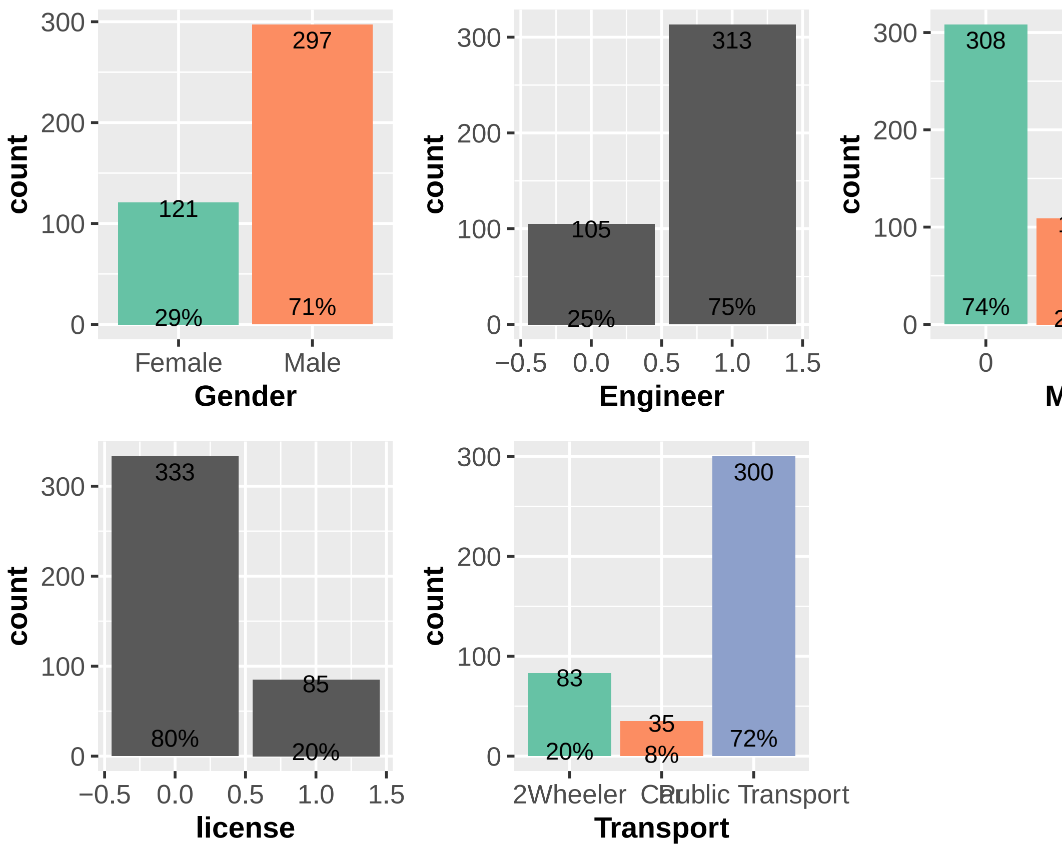
Setting up the aesthetics

```
g1=ggplot(cars, aes(x=Gender, fill=Gender)) + geom_bar()+ unipar + scale_fill_brewer(palette=col1) +
  geom_text(aes(label = scales::percent(..prop..), group = 1), stat= "count", size = 3.3, position = pos
  geom_text(aes(label = ..count.., group = 1), stat= "count", size = 3.3, position = position_stack(0.95

g2=ggplot(cars, aes(x=Engineer, fill=Engineer)) + geom_bar()+ unipar + scale_fill_brewer(palette=col1) +
  geom_text(aes(label = scales::percent(..prop..), group = 1), stat= "count", size = 3.3, position = pos
  geom_text(aes(label = ..count.., group = 1), stat= "count", size = 3.3, position = position_stack(0.95
```

```
g3=ggplot(cars, aes(x=MBA, fill=MBA)) + geom_bar()+ unipar + scale_fill_brewer(palette=col1) +
  geom_text(aes(label = scales::percent(..prop..), group = 1), stat= "count", size = 3.3, position = pos
  geom_text(aes(label = ..count.., group = 1), stat= "count", size = 3.3, position = position_stack(0.95

g4=ggplot(cars, aes(x=license, fill=license)) + geom_bar()+ unipar + scale_fill_brewer(palette=col1) +
  geom_text(aes(label = scales::percent(..prop..), group = 1), stat= "count", size = 3.3, position = pos
  geom_text(aes(label = ..count.., group = 1), stat= "count", size = 3.3, position = position_stack(0.95

g5=ggplot(cars, aes(x=Transport, fill=Transport)) + geom_bar()+ unipar + scale_fill_brewer(palette=col1
  geom_text(aes(label = scales::percent(..prop..), group = 1), stat= "count", size = 3.3, position = pos
  geom_text(aes(label = ..count.., group = 1), stat= "count", size = 3.3, position = position_stack(0.95
```

Plotting the bar charts

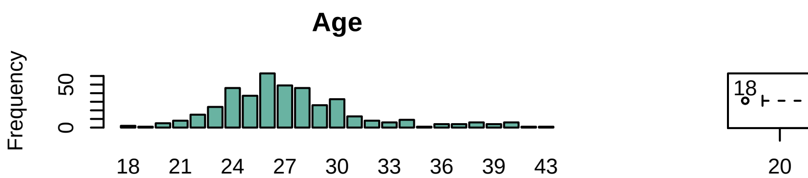
```
grid.arrange(g1,g2,g3,g4,g5,ncol=3)
```

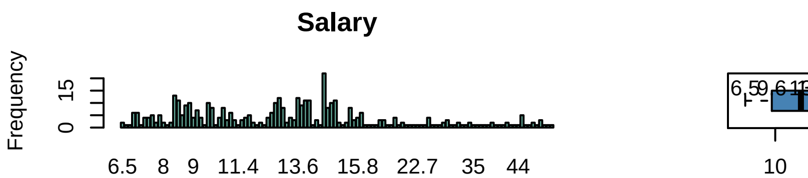


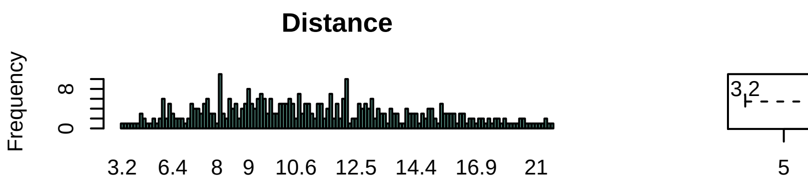
Partitioning the barcharts

```
par(mfrow = c(3,2));

text(x= barplot(table(cars$Age),col='#69b3a2', main = "Age",ylab = "Frequency"),
```

```r
     y = 0, table(cars$Age), cex=1,pos=1);
boxplot(cars$Age, col = "steelblue", horizontal = TRUE, main = "Age");
text(x = fivenum(cars$Age), labels = fivenum(cars$Age), y = 1.25)

text(x= barplot(table(cars$Salary),col='#69b3a2', main = "Salary",ylab = "Frequency"),
     y = 0, table(cars$Salary), cex=1,pos=1);
boxplot(cars$Salary, col = "steelblue", horizontal = TRUE, main = "Salary");
text(x = fivenum(cars$Salary), labels = fivenum(cars$Salary), y = 1.25)

text(x= barplot(table(cars$Distance),col='#69b3a2', main = "Distance",ylab = "Frequency"),
     y = 0, table(cars$Distance), cex=1,pos=1);
boxplot(cars$Distance, col = "steelblue", horizontal = TRUE, main = "Distance");
text(x = fivenum(cars$Distance), labels = fivenum(cars$Distance), y = 1.25)
```







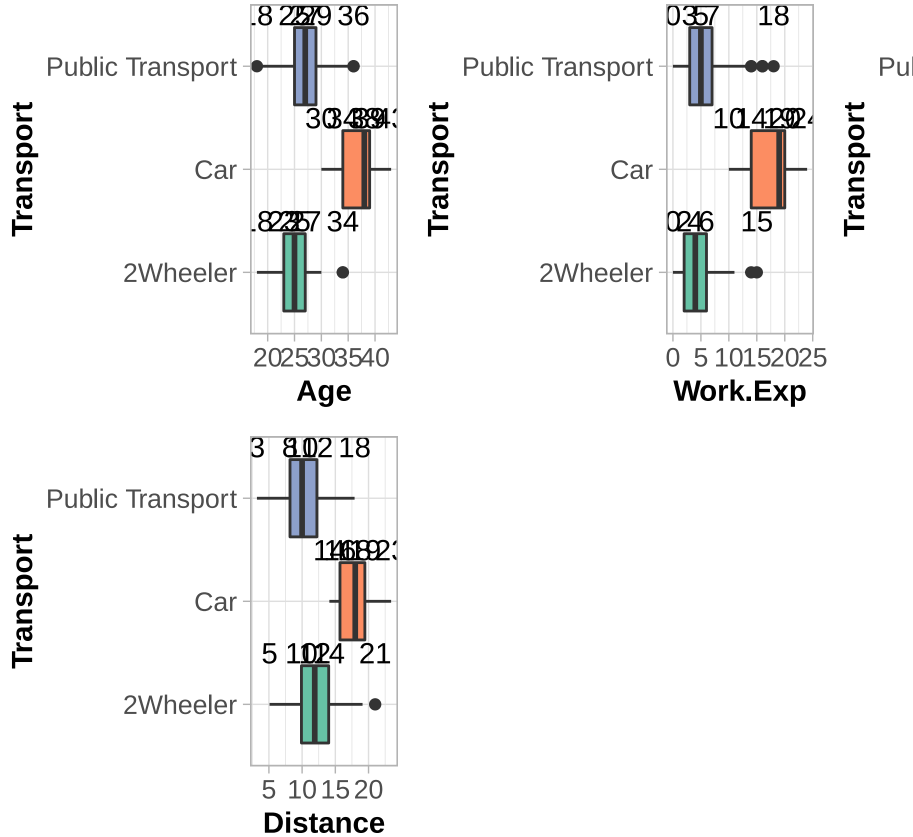
Visualize properties of all continuous variables

BIVARIATE ANALYSIS

```r
bipar1 = theme(legend.position = "none") + theme_light() +
  theme(axis.text = element_text(size = 10),
        axis.title = element_text(size = 11),
        title = element_text(size = 13, face = "bold"))

# Define color brewer
col2 = "Set2"
```

Setting up the aesthetics

```
p1=ggplot(cars, aes(x = Transport, y = Age, fill = Transport)) + geom_boxplot(show.legend = FALSE)+ bipa
  stat_summary(fun = quantile, geom = "text", aes(label=sprintf("%1.0f", ..y..)),position=position_nudge


p2=ggplot(cars, aes(x = Transport, y = Work.Exp, fill = Transport)) + geom_boxplot(show.legend = FALSE)+
  stat_summary(fun = quantile, geom = "text", aes(label=sprintf("%1.0f", ..y..)),position=position_nudge


p3=ggplot(cars, aes(x = Transport, y = Salary, fill = Transport)) + geom_boxplot(show.legend = FALSE)+ b
  stat_summary(fun = quantile, geom = "text", aes(label=sprintf("%1.0f", ..y..)),position=position_nudge


p4=ggplot(cars, aes(x = Transport, y = Distance, fill = Transport)) + geom_boxplot(show.legend = FALSE)+
  stat_summary(fun = quantile, geom = "text", aes(label=sprintf("%1.0f", ..y..)),position=position_nudge

# Partitioning the boxplots
grid.arrange(p1,p2,p3,p4,ncol=3)
```



TransportType vs numerical variables

Observations: * Public Transport
+ Age : Most commuters are in the range of 19 & 35 with maximum in between 25 & 29. There are outliers at both ends at 18 & 36. + Work Exp : The range is predominantly between0 & 13 with concentration

around 3 & 7 years. Though there are outliers between 14 & 18 years. + Salary : Most are concentrated between 6K to 22K with most making around 10K & 15K. There are quite a few outliers at a higher range between 25K to 37K. + Distance : Most commuters are in the range of 3kms to 18kms from office, majority of them staying between 23kms and 27kms from the office

- 2 Wheeler
  - Age : Most commuters are in the range of 18 & 30 with maximum in between 23 & 27, with an outlier at 34.
  - Work Exp : The range is predominantly between 0 & 12 with concentration around 2 & 6 years. The it a outliers around 14 & 15.
  - Salary : Most are concentrated between 6K & 24K with maximum in between 9K & 15K. A few outliers between 24K & 37K.
  - Distance : Most commuters are in the range of 5kms to 19kms from office, majority of them staying between 10kms and 14kms from the office, with an outlier at 21kms.
- Car :
  - Age : Most commuters are in the range of 30 & 43 years with maximum in between 34 & 39
  - Work Exp : The range is predominantly between 10 & 24 with concentration around 14 & 20 years.
  - Salary : The salaries are at a higher range between 31K to 57K while most are concentrated between 37K & 48K. A few outliers at a lower end around 15K & 16K.
  - Distance : Most commuters are in the range of 14kms to 23kms from office, majority of them staying between 16kms and 18kms from the office
- It can be concluded that:
- Age = People traveling by Car are older than the ones commuting by 2 Wheeler & Public Transport.The range of commuters traveling by Public Transport is widest.
- Work Experience = Like Age the people traveling by Car are much more experienced than the others. Their experience coincides with their Age.
- Salary = Similar story with Salary. Coinciding with their Age & Experience, the commuters traveling in Car make more than double the salaried made by commuters traveling by 2 Wheelers and Public Transport. An important observation though is that some commuters using Public Transport make higher salaries in the range of 25K to 37K
- Distance = Commuters traveling in Car stay further away from the office compared to others.

```
bipar2 = theme(legend.position = "top",
               legend.direction = "horizontal",
               legend.title = element_text(size = 10),
               legend.text = element_text(size = 8)) +
  theme(axis.text = element_text(size = 10),
        axis.title = element_text(size = 11),
        title = element_text(size = 13, face = "bold"))
```

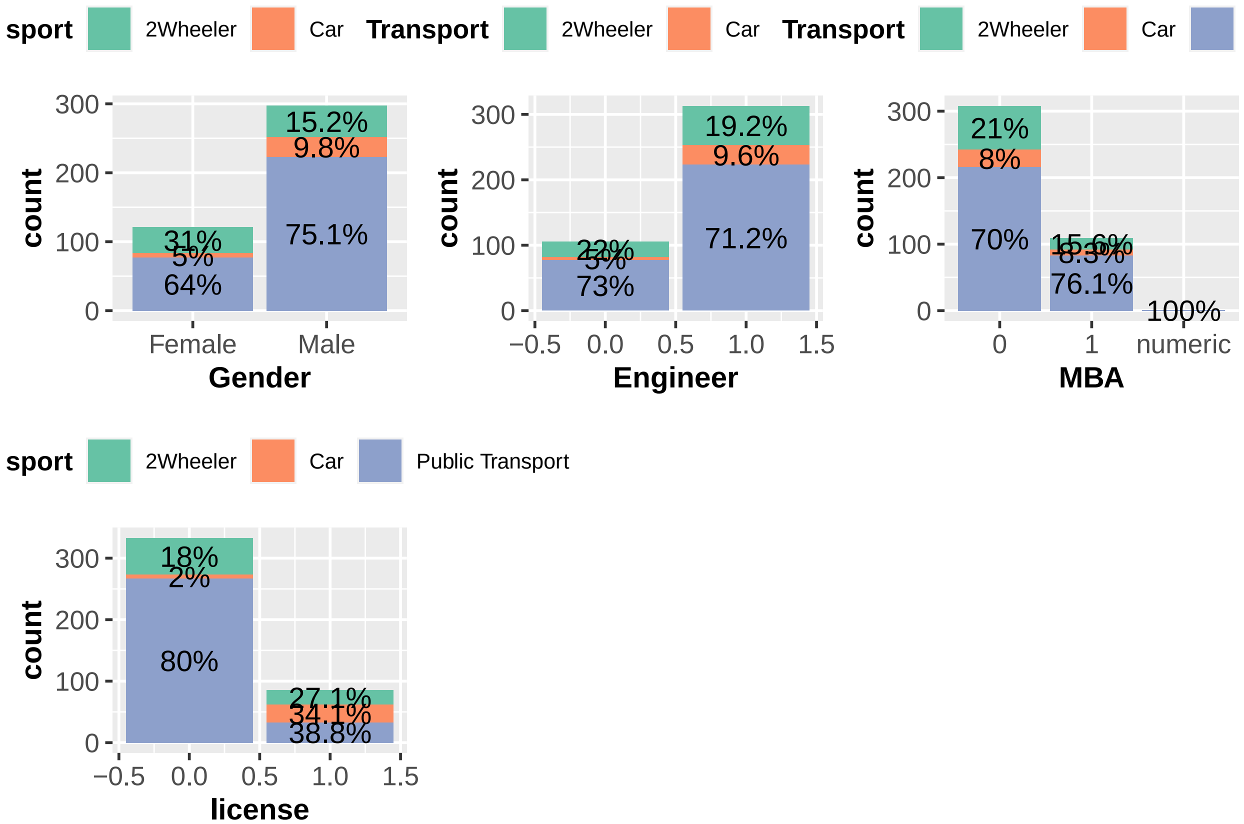
Setting up the aesthetics

```
library(dplyr)
```

Transport Type vs categorical variables

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:gridExtra':
##
##     combine

## The following object is masked from 'package:MASS':
##
##     select

## The following object is masked from 'package:car':
##
##     recode

## The following object is masked from 'package:xgboost':
##
##     slice

## The following object is masked from 'package:randomForest':
##
##     combine

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
d8 <- cars %>% group_by(Gender) %>% count(Transport) %>% mutate(ratio=scales::percent(n/sum(n)))
p8=ggplot(cars, aes(x=Gender, fill=Transport)) + geom_bar()+ bipar2 + scale_fill_brewer(palette=col2) +
  geom_text(data=d8, aes(y=n,label=ratio),position=position_stack(vjust=0.5))

d9 <- cars %>% group_by(Engineer) %>% count(Transport) %>% mutate(ratio=scales::percent(n/sum(n)))
p9=ggplot(cars, aes(x=Engineer, fill=Transport)) + geom_bar()+ bipar2 + scale_fill_brewer(palette=col2)
  geom_text(data=d9, aes(y=n,label=ratio),position=position_stack(vjust=0.5))

d10 <- cars %>% group_by(MBA) %>% count(Transport) %>% mutate(ratio=scales::percent(n/sum(n)))
p10=ggplot(cars, aes(x=MBA, fill=Transport)) + geom_bar()+ bipar2 + scale_fill_brewer(palette=col2) +
  geom_text(data=d10, aes(y=n,label=ratio),position=position_stack(vjust=0.5))

d11 <- cars %>% group_by(license) %>% count(Transport) %>% mutate(ratio=scales::percent(n/sum(n)))
p11=ggplot(cars, aes(x=license, fill=Transport)) + geom_bar()+ bipar2 + scale_fill_brewer(palette=col2)
  geom_text(data=d11, aes(y=n,label=ratio),position=position_stack(vjust=0.5))

# Partitioning the boxplots
grid.arrange(p8,p9,p10,p11,ncol=3)
```



Observations:

- Gender
  - Among the 121 females, 64% take the Public Transport, while 31% use a 2Wheeler
  - Among the 297 males, majority of 75% use the Public Transport while 15.2% take 2 Wheeler & almost 10% have a Car.
- Engineer
  - Of the 313 Engineers 71% Engineers & of the 105 non-Engineers, 73% take Public Transport
  - Almost 10% of Engineers drive a Car to work.
- MBA
  - Of the 109 MBAs 76% & of the 309 non MBAs - 70% commute using Public Transport.
  - Almost 8% of both cohort drive Car to office.
- License
  - Of 333 not owning license, 80% use Public Transport while 18% & 2% use 2 Wheeler & Car respectively.
  - The 85 who possess a license have 34% driving Car, 27% riding a 2 Wheeler

Create new factor variable using "Transport" variable

```
cars$TransportType=cars$Transport
cars$TransportType=as.character(cars$TransportType)
cars$TransportType[cars$TransportType=="2Wheeler"|
                        cars$TransportType=="Public Transport"] <- "Other.Transport"
cars$TransportType=as.factor(cars$TransportType)
cars<- cars[-9]
```

Observations: + For the benefit of our analysis, we need to group the transport variable into people using "Car" & 'Other Transport" i.e. not using the car to commute to office. + We convert the "Transport" into "Transport Type" and group "2 Wheeler" & "Public Transport" in one title, namely "Other Transport"

Outlier Treatment

```
outlier_treatment_fun = function(data,var_name){
  capping = as.vector(quantile(data[,var_name],0.99))
  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)
}

new_vars = c('Age', 'Work.Exp', 'Salary', 'Distance')
```

- The outliers observed in Age, Work Experience, Salary & Distance are treated with Outlier Treatment to make sure the outliers do not wrongly impact the models that will be build.

Create a subset of data with only the numeric variables

```
subset_cars = cars[, c("Age","Work.Exp","Salary","Distance")]
```

Creating a filtered data frame

```
highCorr <- findCorrelation(cor(subset_cars[,-4]), cutoff = 0.8)
```

Storing the result of findCorrelation function in a variable

```
filter_cor_data <- subset_cars[, -highCorr]
filter_cor_data$TransportType<-cars$Transport
```

filtering the data i.e. removing the highly correlated columns

New Data without the highly correlated columns

```
  cars1=cars[,-c(1,4,5)]
```

5. Modelling: Create Multiple Models

Split the Data into Train & Test (80-20 split)

```
set.seed(123)

  trainIndex <- createDataPartition(cars1$Transport, p = .80, list = FALSE)

  cars_Train <- cars1[ trainIndex,]
  cars_Test  <- cars1[-trainIndex,]


  prop.table(table(cars1$Transport))*100
```

```
##
##             Car Other.Transport
##        8.373206       91.626794
```

```
  prop.table(table(cars_Train$Transport))*100
```

```
##
##             Car Other.Transport
##        8.358209       91.641791
```

```
  prop.table(table(cars_Test$Transport))*100
```

```
##
##             Car Other.Transport
##        8.433735       91.566265
```

Observation: The Train & Test Split Data is almost same to the refrerred data. The split of "Car" & "Other Transport" is almost the same.

Setting up the general parameters for training multiple models

```


fitControl <- trainControl(
  method = 'repeatedcv',            # k-fold cross validation
  number = 5,                      # number of folds or k
  repeats = 1,                      # repeated k-fold cross-validation
  allowParallel = TRUE,
  classProbs = TRUE,
  summaryFunction=twoClassSummary# should class probabilities be returned
)
```

Define the training control Note: We set up a training control parameter for the various models that we will be creating and exploring.

Model 1 : Logistic Regression Model

```
lrmod <- caret::train(TransportType ~ .,
                       method     = "glm",
                       metric     = "Sensitivity",
                       data       = cars_Train)
```

```
lrpred<-predict(lrmod,newdata=cars_Test)
```

Predicting on Test data

```
caret::confusionMatrix(cars_Test$TransportType,lrpred,positive="Other.Transport")
```

Checking the confusion matrix

```
## Confusion Matrix and Statistics
##
##                  Reference
## Prediction        Car Other.Transport
##   Car               5               2
##   Other.Transport   0              76
##
##                Accuracy : 0.9759
##                  95% CI : (0.9157, 0.9971)
##     No Information Rate : 0.9398
##     P-Value [Acc > NIR] : 0.1169
##
##                   Kappa : 0.8207
##
##  Mcnemar's Test P-Value : 0.4795
##
##             Sensitivity : 0.9744
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.7143
##              Prevalence : 0.9398
##          Detection Rate : 0.9157
##    Detection Prevalence : 0.9157
##       Balanced Accuracy : 0.9872
##
##        'Positive' Class : Other.Transport
##
```

Observation: + Logistic Regression model shows Accuracy of 98.80%, Sensitivity of 98.70%% & Specificity of 100%% + The True positive rate is good with only 1 False positive prediction which is better than the KNN output. The True Negative is 100% with no false negative denoting its a good model.

```r
caret::varImp(lrmod)
```

Checking the Variable importance

```
## glm variable importance
##
##            Overall
## Salary      100.00
## Distance     88.32
## Engineer     21.86
## license      12.37
## GenderMale    0.00
```

- Observation:
  - "Salary" comes out as the clear most important variable in determining the choice of commute for the office going staff.
  - The "Distance" also determines the choice of commute and we had observed during our analysis that people staying further away from office has more Cars comparatively.

Model 2 : Naive Bayes

```r
cars_Train$TransportType<- as.factor(cars_Train$TransportType)
  cars_Test$TransportType<- as.factor(cars_Test$TransportType)
  model_nb <- caret::train(TransportType ~ ., data = cars_Train,
                           method = "naive_bayes")
```

```r
summary(model_nb)
```

Checking the confusion matrix

```
##
## ================================== Naive Bayes ==================================
##
## - Call: naive_bayes.default(x = x, y = y, laplace = param$laplace, usekernel = TRUE,     adjust = pa
## - Laplace: 0
## - Classes: 2
## - Samples: 335
## - Features: 5
## - Conditional distributions:
##     - KDE: 5
## - Prior probabilities:
```

```
##     - Car: 0.0836
##     - Other.Transport: 0.9164
##
## ---------------------------------------------------------------------------------
```

```
  nb_predictions_test <- predict(model_nb, newdata = cars_Test, type = "raw")
  nb_predictions_test=as.numeric(nb_predictions_test)
  cars_Test$TransportType=as.numeric(cars_Test$TransportType)
  confusionMatrix(nb_predictions_test, cars_Test$TransportType)
```

```
##   1  2
## 1 5 78
```

Model 3 : KNN

```
set.seed(123)

  cars_Train$TransportType <- as.factor(cars_Train$TransportType)
  cars_Test$TransportType <- as.factor(cars_Test$TransportType)

  set.seed(123)
  knn_model <- caret::train(TransportType ~ ., data = cars_Train,
                          preProcess = c("center" ),
                          method = "knn",
                          tuneLength = 3,
                          trControl = fitControl,
                          metric     = "Accuracy")


knn_model
```

```
## k-Nearest Neighbors
##
## 335 samples
##   5 predictor
##   2 classes: 'Car', 'Other.Transport'
##
## Pre-processing: centered (5)
## Resampling: Cross-Validated (5 fold, repeated 1 times)
## Summary of sample sizes: 267, 269, 268, 269, 267
## Resampling results across tuning parameters:
##
##   k  ROC        Sens       Spec
##   5  0.9620219  0.8866667  1.0000000
##   7  0.9607650  0.8866667  0.9967742
##   9  0.9607650  0.8533333  0.9934955
##
## ROC was used to select the optimal model using the largest value.
## The final value used for the model was k = 5.
```

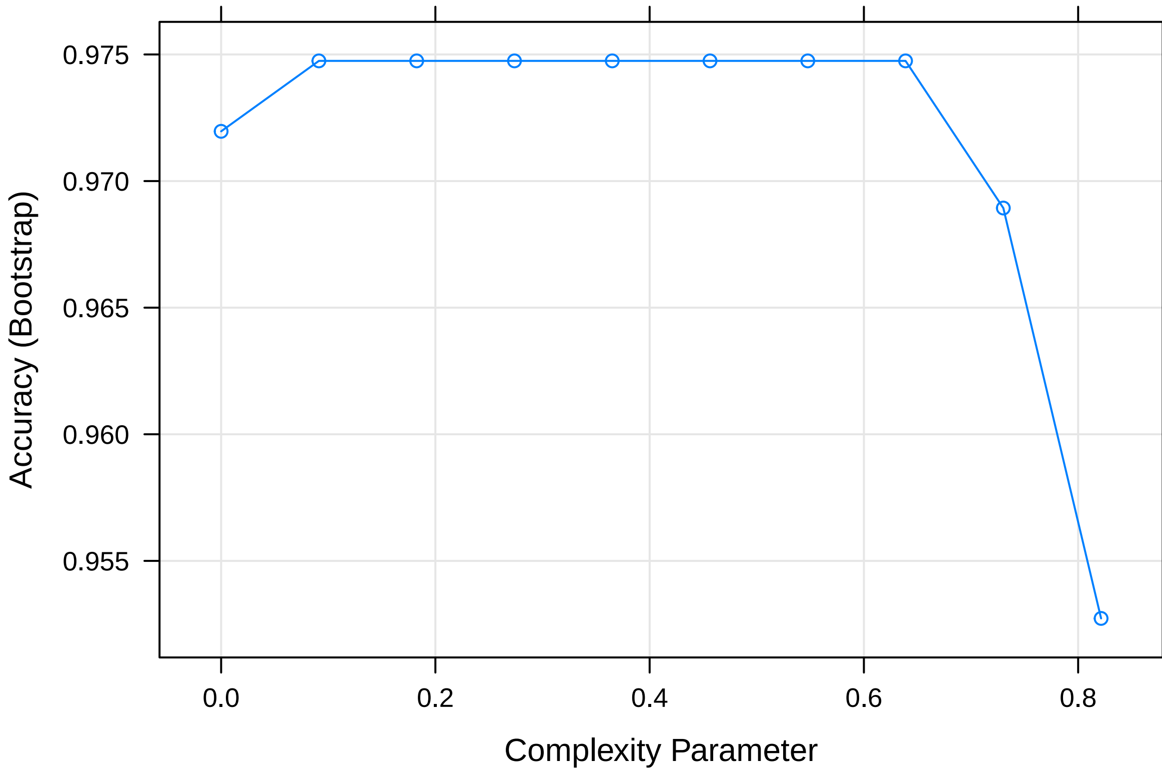
Model 4 : Rpart : Single CART decision tree

```r
cars1$TransportType <- as.factor(cars1$TransportType)
cars_Train$TransportType <- as.factor(cars_Train$TransportType)
cars_Test$TransportType <- as.factor(cars_Test$TransportType)

model_dtree <- caret::train(TransportType ~ ., data = cars_Train[,-1],
                              method = "rpart",
                              minbucket = 100,
                              cp = 0,
                              tuneLength = 10,
                              na.action=na.roughfix)

  model_dtree
```

```
## CART
##
## 335 samples
##   4 predictor
##   2 classes: 'Car', 'Other.Transport'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 335, 335, 335, 335, 335, 335, ...
## Resampling results across tuning parameters:
##
##   cp          Accuracy   Kappa
##   0.00000000  0.9719641  0.8022071
##   0.09126984  0.9747459  0.8290909
##   0.18253968  0.9747459  0.8290909
##   0.27380952  0.9747459  0.8290909
##   0.36507937  0.9747459  0.8290909
##   0.45634921  0.9747459  0.8290909
##   0.54761905  0.9747459  0.8290909
##   0.63888889  0.9747459  0.8290909
##   0.73015873  0.9689340  0.7536992
##   0.82142857  0.9527275  0.5406617
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was cp = 0.6388889.
```

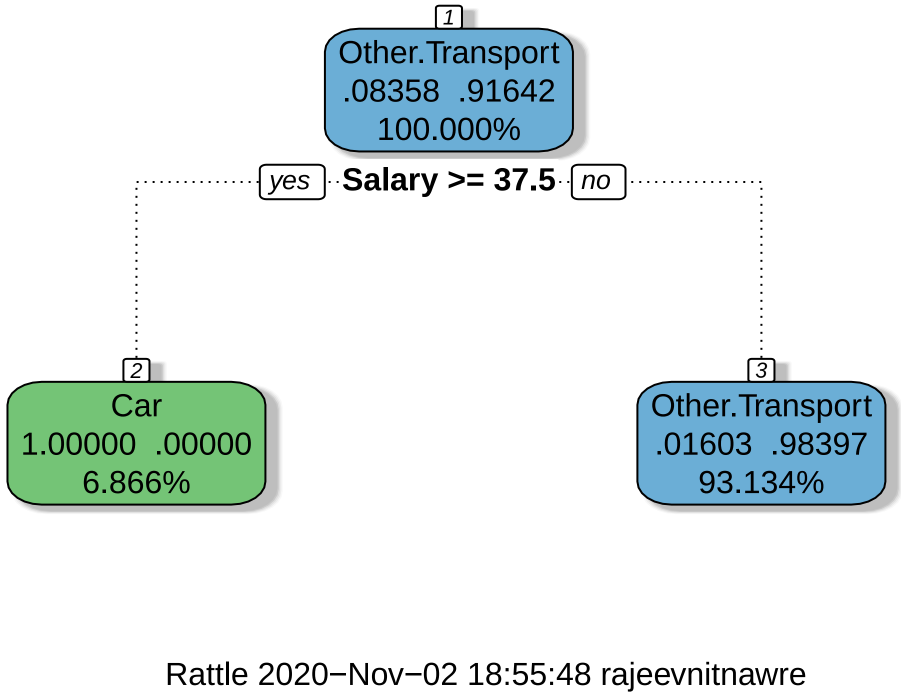
Plot the cp vs ROC values to see the effect of cp on ROC

```r
plot(model_dtree)
```



Plot the CP values

```
fancyRpartPlot(model_dtree$finalModel,digits = 5 )
```



Plot the tree

```
 dtree_predictions_test = predict(model_dtree$finalModel, newdata = cars_Test[,-1], type = "vector")
  cars_Test$TransportType=as.numeric(cars_Test$TransportType)
  dtree_predictions_test=as.numeric(dtree_predictions_test)
  confusionMatrix(dtree_predictions_test, cars_Test$TransportType)
```

Predict using the trained model & check performance on test set

```
##    1  2
## 1  3 80
```

Model_5 : Random Forest

```
cars1$Transport <- as.factor(cars1$TransportType)
  cars_Train$TransportType <- as.factor(cars_Train$TransportType)
  cars_Test$TransportType <- as.factor(cars_Test$TransportType)

  model_rf <- caret::train(TransportType ~ ., data = cars_Train,
                           method = "rf",
                           ntree = 30,
                           maxdepth = 5,
                           tuneLength = 10)
```

```
## note: only 4 unique complexity parameters in default grid. Truncating the grid to 4 .
```

```r
rf_predictions_test <- predict(model_rf, newdata = cars_Test, type = "raw")

  cars_Test$TransportType=as.numeric(cars_Test$TransportType)
  length(cars_Test$TransportType)
```

Predict using the trained model & check performance on test set

```
## [1] 83
```

```r
  length(rf_predictions_test)
```

```
## [1] 83
```

```r
confusionMatrix(rf_predictions_test, cars_Test$TransportType)
```

```
##   Car Other.Transport
## 1   5              78
```

Model_6 : Gradient Boosting Machines

```r
 cars1$TransportType <- as.factor(cars1$TransportType)
  cars_Train$TransportType <- as.factor(cars_Train$TransportType)
  cars_Test$TransportType <- as.factor(cars_Test$TransportType)
  gbm_model <- caret::train(TransportType ~ ., data = cars_Train,
                          method = "gbm",
                          na.action=na.roughfix,
                          verbose = FALSE)
```

```r
gbm_predictions_test <- predict(gbm_model, newdata = cars_Test, type = "raw")
  cars_Test$TransportType=as.numeric(cars_Test$TransportType)
  gbm_predictions_test=as.numeric(gbm_predictions_test)
  confusionMatrix(gbm_predictions_test, cars_Test$TransportType)
```

Predict using the trained model & check performance on test set

```
##   1  2
## 1 5 78
```

Model_7 : Xtreme Gradient boosting Machines [without smote or with highly unbalanced data]

```r
cv.ctrl <- trainControl(method = "repeatedcv", repeats = 1,number = 3,
                          summaryFunction = twoClassSummary,
                          classProbs = TRUE,
                          allowParallel=T)

  xgb.grid <- expand.grid(nrounds = 100,
                          eta = c(0.01),
                          max_depth = c(2,4),
                          gamma = 0,               #default=0
                          colsample_bytree = 1,    #default=1
                          min_child_weight = 1,    #default=1
                          subsample = 1            #default=1
  )

  xgb_model <- caret::train(TransportType~.,
                            data=cars_Train,
                            method="xgbTree",
                            trControl=cv.ctrl,
                            tuneGrid=xgb.grid,

                            verbose=T,

  )
```

```
## Warning in train.default(x, y, weights = w, ...): The metric "Accuracy" was not
## in the result set. ROC will be used instead.
```

Predict using the trained model & check performance on test set

```r
 xgb_predictions_test <- predict(xgb_model, newdata = cars_Test, type = "raw")

  cars_Test$TransportType=as.numeric(cars_Test$TransportType)
  xgb_predictions_test=as.numeric(xgb_predictions_test)
  confusionMatrix(xgb_predictions_test, cars_Test$TransportType)
```

```
##    1  2
## 1 5 78
```

SMOTE

```r
cars_Train <- cars1[ trainIndex,]
  cars_Test  <- cars1[-trainIndex,]
  cars1$Transport <- as.factor(cars1$Transport )
  cars_Train$TransportType <- as.factor(cars_Train$TransportType)
  cars_Test$TransportType <- as.factor(cars_Test$TransportType)

  table(cars_Train$TransportType)
```

```
##
```

```
##             Car Other.Transport
##              28             307
```

```r
prop.table(table(cars_Train$TransportType))
```

```
##
##             Car Other.Transport
##      0.08358209      0.91641791
```

```r
smote_train <- SMOTE(TransportType ~ ., data  = cars_Train,
                     perc.over = 3700,
                     perc.under = 300,
                     k = 5)


prop.table(table(smote_train$TransportType))*100
```

```
##
##             Car Other.Transport
##        25.50336        74.49664
```

```r
table(smote_train$TransportType)
```

```
##
##             Car Other.Transport
##            1064            3108
```

#Model_8 : Xtreme Gradient boosting Machines [with smote or with less unbalanced data]

```r
cv.ctrl <- trainControl(method = "repeatedcv", repeats = 1,number = 3,
                        summaryFunction = twoClassSummary,
                        classProbs = TRUE,
                        allowParallel=T)

xgb.grid <- expand.grid(nrounds = 500,
                        eta = c(0.01),
                        max_depth = c(2,4),
                        gamma = 0,               #default=0
                        colsample_bytree = 1,    #default=1
                        min_child_weight = 1,
                        subsample = 1          #default=1
)

smote_xgb_model <- caret::train(TransportType~.,
                                data=smote_train,
                                method="xgbTree",
                                trControl=cv.ctrl,
                                tuneGrid=xgb.grid,
                                verbose=T,
                                nthread = 2,na.action=na.roughfix
)
```

```
## Warning in train.default(x, y, weights = w, ...): The metric "Accuracy" was not
## in the result set. ROC will be used instead.
```

Predict using the trained model & check performance on test set

```
xgb_predictions_test <- predict(smote_xgb_model, newdata = cars_Test, type = "raw")
cars_Test$TransportType=as.numeric(cars_Test$TransportType)
xgb_predictions_test=as.numeric(xgb_predictions_test)
confusionMatrix(xgb_predictions_test, cars_Test$TransportType)
```

```
##   1  2
## 1 7 76
```

Bagging:

```
library(ipred)
library(rpart)
cars_Train=cars_Train[,-7]
cars_Test=cars_Test[,-7]
mod.bagging= bagging(TransportType~.,data = cars_Train, control= rpart.control(maxdepth = 5, minsplit =
```

Predict using the trained model & check performance on test set

```
bag.predict= predict(mod.bagging,cars_Test)
cars_Test$TransportType=as.numeric(cars_Test$TransportType)
bag.predict=as.numeric(bag.predict)
confusionMatrix(bag.predict,cars_Test$TransportType)
```

```
##   1  2
## 1 5 78
```

COMPARING MODELS

```
Name = c("KNN", "Logistic_Regression","CART")
Accuracy = c(0.00,97.59, 97.56)
Sensitivity=c(88.66,97.44,0.00)
Specificity=c(100.00,100.00,0.00)
ROC=c(96.20,0.00,0.00)
models_to_compare = data.frame(Name,Accuracy,Sensitivity,Specificity,ROC)
models_to_compare
```

```
##                   Name Accuracy Sensitivity Specificity  ROC
## 1                 KNN     0.00       88.66         100 96.2
## 2 Logistic_Regression    97.59       97.44         100  0.0
## 3                CART    97.56        0.00           0  0.0
```

- Observation:
  - Looking at the Accuracy / ROC the Logistic Regression has an edge over the KNN & Decison Tree (CART)
  - Sensitivity for Logistic Regression is much batter than the KNN model
  - Specificity for both are at the maximum.

```r
Name = c("CART_Decision_tree","Random_Forest","Gradient_boosting","Xtreme_Gradient","Smote_Xtreme_Gradi
Car = c(3,5,5,5,7,5,5,5)
Other.Transport = c(80,78,78,78,76,78,78,78)
models_to_compare1 = data.frame(Name,Car,Other.Transport)
models_to_compare1
```

```
##                   Name Car Other.Transport
## 1    CART_Decision_tree   3              80
## 2         Random_Forest   5              78
## 3     Gradient_boosting   5              78
## 4       Xtreme_Gradient   5              78
## 5 Smote_Xtreme_Gradient   7              76
## 6           Naive.Bayes   5              78
## 7   Logistic.Regression   5              78
## 8               Bagging   5              78
```

- Observation:
  - Most of the models are coming out with similar outcomes for predicting Car & Other Transport.
  - There isn't much that these models are generation different from each other
  - Using the SMOTE, the extreme Gradient shows a better output compared to the others.

Actionable Insights & Recommendations:

Conclusion:

- The data was explored and worked upon. The processed and clean data was later checked for missing values, outliers and multi-colinearity.
- We created the following models:
  - Linear Regression

  - KNN
  - Naive Bayes
  - CART_Decision_tree
  - Random_Forest
  - Gradient_boosting
  - Xtreme_Gradient
  - Smote_Xtreme_Gradient
  - Bagging
- The model which will give a good insight to the data to predict if the employee will use a Car as a mode of transport should be the Logistic Regression Model. Using the SMOTE the Extreme Gradient also shows an improvement.
- Salary comes across as the most influential variable in deciding if Car would be the preferred mode of commute to office.
- The Distance for traveling to office also determines what mode of transport the employees choose.

Recommendations:

- Further Performance can be checked using multivariate analysis i.e plots among the independent variables to generate more insights.
- Use variable transformation like taking ratios of independent variables and check if the model performance improves.
- Overfitting & Underfitting techniques can be tried for imbalanced data.
- Further deeper investigation on the various mode of transport in proportion to variables like distance & Salary could give more insight.