CAR

Muthu Pandian G

December 25, 2019

## OBJECTIVE OF THE PROJECT:

# 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

# Illustrate the insights based on EDA

# Check for Multicollinearity - Plot the graph based on Multicollinearity & treat it

# Data Preparation

# Prepare the data for analysis (SMOTE)

# Modeling

# Create multiple models and explore how each model perform using appropriate model performance metrics

# 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.

# Actionable Insights & Recommendations

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

## Importing the Dataset

setwd("D:/Great Lakes/Projects/Machine Learning")  
getwd()

## [1] "D:/Great Lakes/Projects/Machine Learning"

cars <- read.csv("cars.csv",header = TRUE)

## Understanding the data

## Data Description

The dataset has details on 418 employees details with 9 Variables. ### Structure of Data

str(cars)

## 'data.frame': 418 obs. of 9 variables:  
## $ Age : int 28 24 27 25 25 21 23 23 24 28 ...  
## $ Gender : Factor w/ 2 levels "Female","Male": 2 2 1 2 1 2 2 2 2 2 ...  
## $ Engineer : int 1 1 1 0 0 0 1 0 1 1 ...  
## $ MBA : int 0 0 0 0 0 0 1 0 0 0 ...  
## $ Work.Exp : int 5 6 9 1 3 3 3 0 4 6 ...  
## $ Salary : num 14.4 10.6 15.5 7.6 9.6 9.5 11.7 6.5 8.5 13.7 ...  
## $ Distance : num 5.1 6.1 6.1 6.3 6.7 7.1 7.2 7.3 7.5 7.5 ...  
## $ license : int 0 0 0 0 0 0 0 0 0 1 ...  
## $ Transport: Factor w/ 3 levels "2Wheeler","Car",..: 1 1 1 1 1 1 1 1 1 1 ...

We see that License,Engineer,MBA Variables are taken as numerical variable. We need to convert it to categorical variable.

cars$license <- as.factor(cars$license)  
cars$Engineer <- as.factor(cars$Engineer)  
cars$MBA <- as.factor(cars$MBA)

Now lets look at the Structure of our Dataset

str(cars)

## 'data.frame': 418 obs. of 9 variables:  
## $ Age : int 28 24 27 25 25 21 23 23 24 28 ...  
## $ Gender : Factor w/ 2 levels "Female","Male": 2 2 1 2 1 2 2 2 2 2 ...  
## $ Engineer : Factor w/ 2 levels "0","1": 2 2 2 1 1 1 2 1 2 2 ...  
## $ MBA : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 2 1 1 1 ...  
## $ Work.Exp : int 5 6 9 1 3 3 3 0 4 6 ...  
## $ Salary : num 14.4 10.6 15.5 7.6 9.6 9.5 11.7 6.5 8.5 13.7 ...  
## $ Distance : num 5.1 6.1 6.1 6.3 6.7 7.1 7.2 7.3 7.5 7.5 ...  
## $ license : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 2 ...  
## $ Transport: Factor w/ 3 levels "2Wheeler","Car",..: 1 1 1 1 1 1 1 1 1 1 ...

### Summary

summary(cars)

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

We have 19 % of Employees who commute via thier own two wheelers and 8 % of employees via own car and 71 % of employees via Public Transport ## Checking NA Values/ Missing Values

colSums(is.na(cars))

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

We have only 1 Na value in our Entire dataset. So removing it won’t affect our dataset.

cars<- na.omit(cars)  
colSums(is.na(cars))

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

## Outlier Detection

Since, we need to predict whether or not an employee will use Car as a mode of transport We need to convert the employees who use Public Transport and 2 Wheeler into one Category and car users into one category.

cars$Transport <- ifelse(cars$Transport == "Car",1,0)  
cars$Transport <- as.factor(cars$Transport)

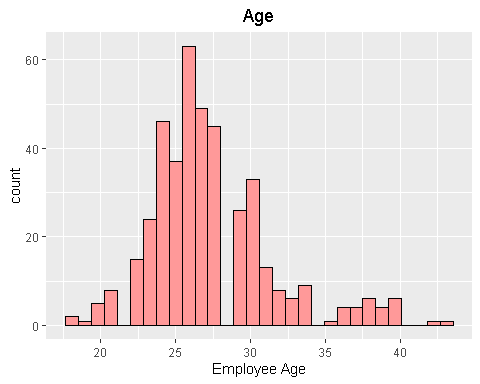
# Exploratory Data Analysis

## Univariate Analysis

## Frequency Distribution of each Independent numerical Variable

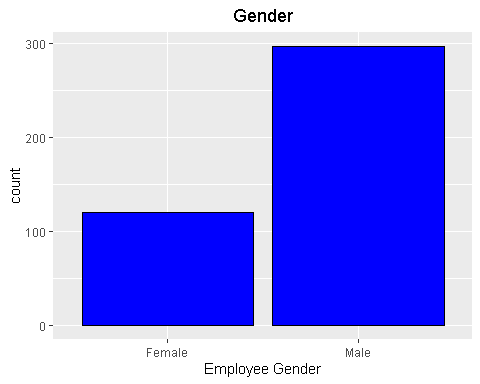
library(ggplot2)  
ggplot(cars,aes(x=Age))+geom\_histogram(fill = "#FF9999",colour = "Black")+ggtitle("Age")+theme(plot.title = element\_text(hjust = 0.5))+xlab("Employee Age")

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

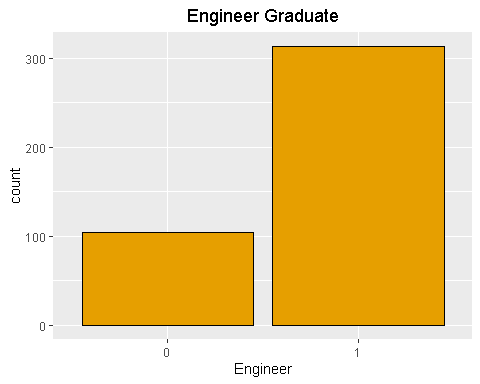
 Age - Most of our Employee are younger as thier average age is around 27.

ggplot(cars,aes(x=Gender))+geom\_bar(bins = 50,fill = "Blue",colour = "Black")+ggtitle("Gender")+theme(plot.title = element\_text(hjust = 0.5))+xlab("Employee Gender")

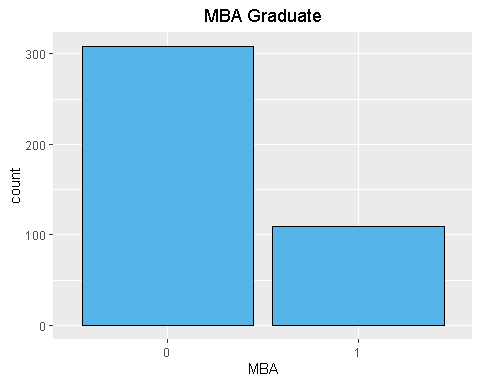
## Warning: Ignoring unknown parameters: bins

 Gender - Our Employee base has lot of Males(71%) than Females (29%)

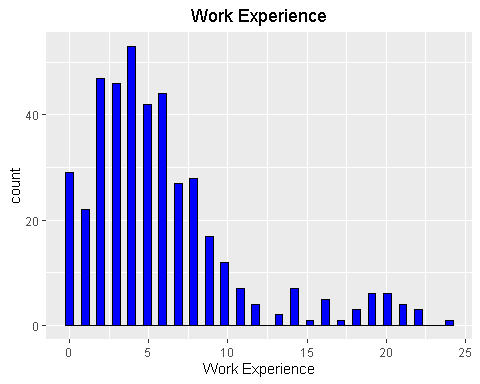
ggplot(cars,aes(x=Engineer))+geom\_bar(fill = "#E69F00",colour = "Black")+ggtitle("Engineer Graduate")+theme(plot.title = element\_text(hjust = 0.5))+xlab("Engineer")

 Engineer - Around 75% of our Employee are engineer graduates and only 25% Employee are non- engineers

ggplot(cars,aes(x=MBA))+geom\_bar(fill = "#56B4E9",colour = "Black")+ggtitle("MBA Graduate")+theme(plot.title = element\_text(hjust = 0.5))+xlab("MBA")

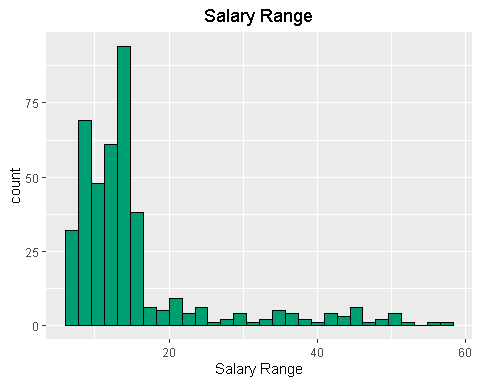
 MBA - Though,we have lot of Engineer graduates as our Employees but we have only 27% of MBA Graduates in our company.

ggplot(cars,aes(x=Work.Exp))+geom\_histogram(bins = 50,fill = "Blue",colour = "Black")+ggtitle("Work Experience")+theme(plot.title = element\_text(hjust = 0.5))+xlab("Work Experience")

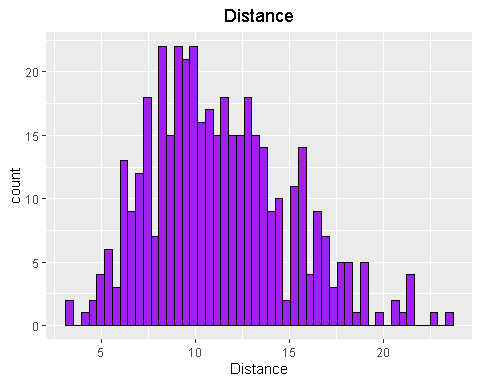
 Work Experience - Employee Work Experience varies from 0 - 25years and on an average our employees have 5 years of work experience

ggplot(cars,aes(x = Salary))+ geom\_histogram(fill = "#009E73",colour = "Black")+ ggtitle("Salary Range") + theme(plot.title = element\_text(hjust = 0.5))+xlab("Salary Range")

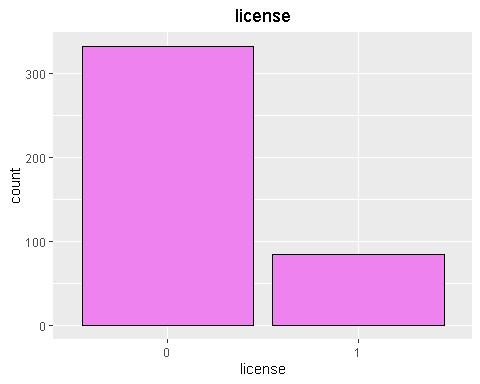
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

 Salary - Employee Salary ranges from 6.50 to 57 and on an Average our Employee get’s a salary of 15.42.

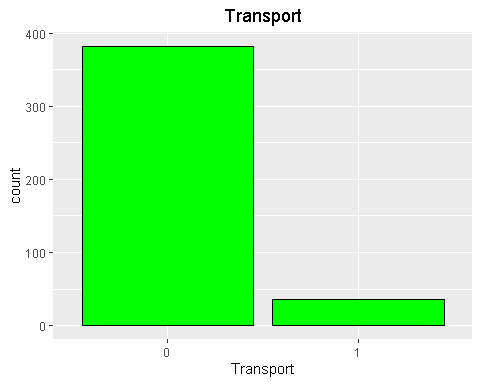
ggplot(cars,aes(x=Distance))+geom\_histogram(bins = 50,fill = "purple",colour = "Black")+ggtitle("Distance")+theme(plot.title = element\_text(hjust = 0.5))+xlab("Distance")

 Distance - Employee commute Distance ranges from 3.2 to 23.4 kilometers and on an average our Employee commutes a distance of 11.3 kilometers

ggplot(cars,aes(x=license))+geom\_bar(fill = "violet",colour = "Black")+ggtitle("license")+theme(plot.title = element\_text(hjust = 0.5))+xlab("license")

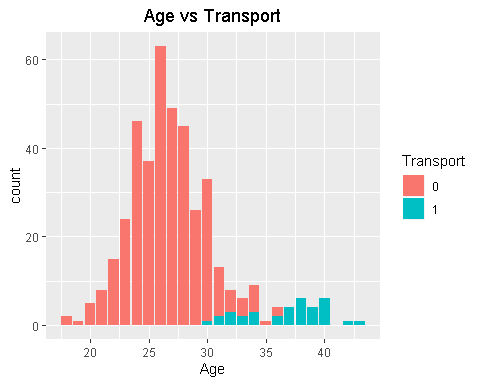
 license - Only 20% of our Employee has License, which is quite surprising.

ggplot(cars,aes(x=Transport))+geom\_bar(fill = "green",colour = "Black")+ggtitle("Transport")+theme(plot.title = element\_text(hjust = 0.5))+xlab("Transport")

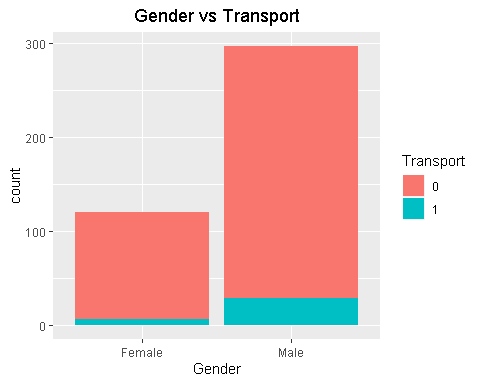
 Transport - We have 19 % of Employees who commute via thier own two wheelers and 8 % of employees via own car and 71 % of employees via Public Transport

## Bi Variable analysis

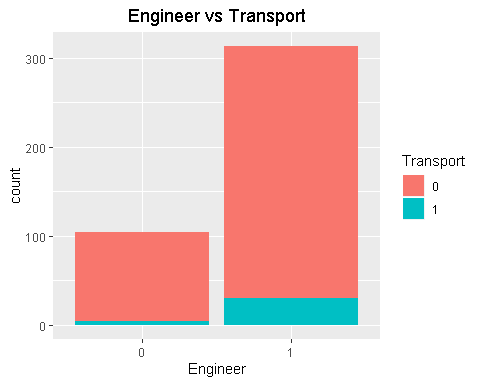
ggplot(cars,aes(Age,fill = Transport))+geom\_bar()+ggtitle("Age vs Transport")+ theme(plot.title = element\_text(hjust = 0.5))+ xlab("Age")

 From the plot, it’s evident that employee whose age above 30 are the ones who use car as a mode of transport and most employees are using public transport only.

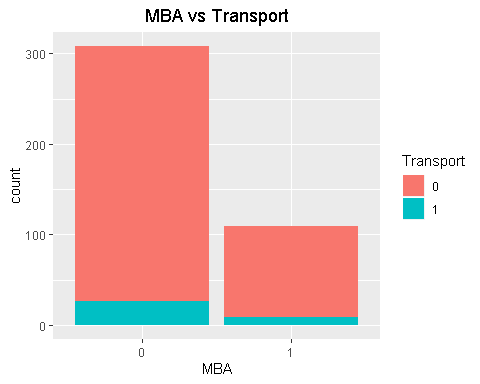
ggplot(cars,aes(Gender,fill = Transport))+geom\_bar()+ggtitle("Gender vs Transport")+ theme(plot.title = element\_text(hjust = 0.5))+ xlab("Gender")

 Out of 297 Male employees only 29 Male employees are driving Car and out of 120 Female employees only 6 Females are commuting via car to the office.

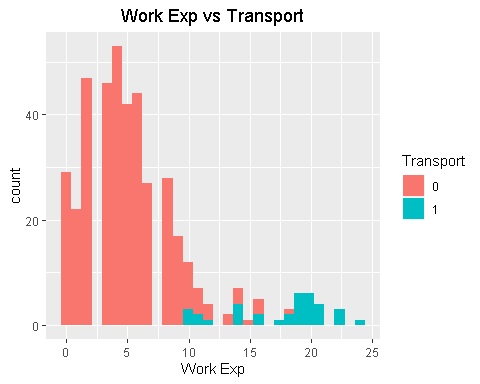
ggplot(cars,aes(Engineer,fill = Transport))+geom\_bar()+ggtitle("Engineer vs Transport")+ theme(plot.title = element\_text(hjust = 0.5))+ xlab("Engineer")

 Out of 313 Engineer graduates, only 30 graduates are commuting to the office via car and only 5 out of 104 non engineers own a car

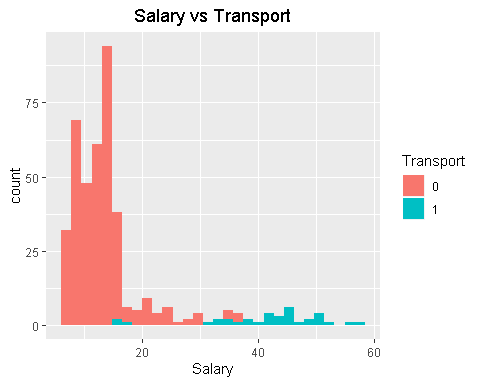
ggplot(cars,aes(MBA,fill = Transport))+geom\_bar()+ggtitle("MBA vs Transport")+ theme(plot.title = element\_text(hjust = 0.5))+ xlab("MBA")

 Out of 109 MBA graduates, only 9 graduates are commuting to the office via car and 26 out of 308 non MBA graduates own a car

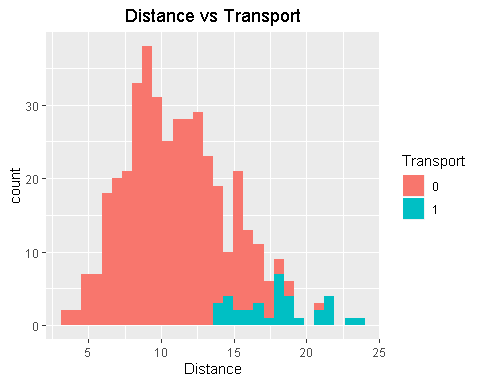
ggplot(cars,aes(Work.Exp,fill = Transport))+geom\_histogram(bins = 30)+ggtitle("Work Exp vs Transport")+ theme(plot.title = element\_text(hjust = 0.5))+ xlab("Work Exp")

 As expected, Higher the work experience higher the chance of commuting to the office via car.

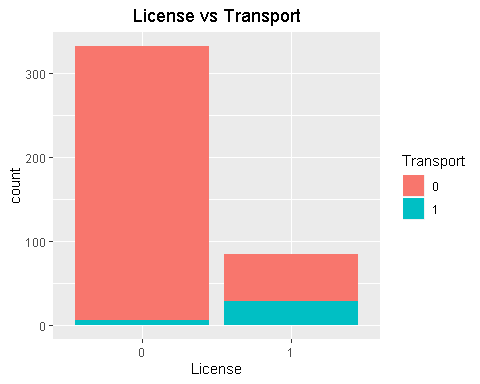
ggplot(cars,aes(Salary,fill = Transport))+geom\_histogram(bins = 30)+ggtitle("Salary vs Transport")+ theme(plot.title = element\_text(hjust = 0.5))+ xlab("Salary")

 As expected, Higher the Salary higher the chance of commuting to the office via car.

ggplot(cars,aes(Distance,fill = Transport))+geom\_histogram(bins = 30)+ggtitle("Distance vs Transport")+ theme(plot.title = element\_text(hjust = 0.5))+ xlab("Distance")

 Higher the distance, more the possibility of commuting to the office via car.

ggplot(cars,aes(license,fill = Transport))+geom\_bar()+ggtitle("License vs Transport")+ theme(plot.title = element\_text(hjust = 0.5))+ xlab("License")

 As expected, Most of the non licensed people uses Public Transport as the way of commute to thier offices and 29 out of 85 licensed employee uses car as a mode of tramsport.

## Multi-Collinearity

Let’s checkout the existence of Multi-collinearity between the Independent variables

library(corrgram)

## Registered S3 method overwritten by 'seriation':  
## method from   
## reorder.hclust gclus

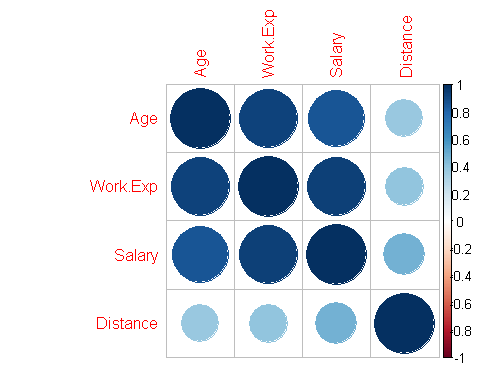
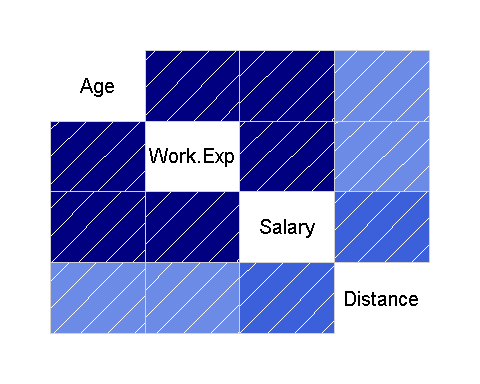
library(corrplot)

## corrplot 0.84 loaded

library(car)

## Loading required package: carData

corrplot::corrplot(corrgram(cars[,-c(2,3,4,8,9)]))



cor(cars[,-c(2,3,4,8,9)])

## Age Work.Exp Salary Distance  
## Age 1.0000000 0.9244489 0.8579114 0.3754669  
## Work.Exp 0.9244489 1.0000000 0.9318574 0.3945957  
## Salary 0.8579114 0.9318574 1.0000000 0.4783049  
## Distance 0.3754669 0.3945957 0.4783049 1.0000000

It’s Evident that Multicollinearity is exist in the dataset.Now, let’s calculate the VIF value and decide how to treat Multi-Collinearity

model <- glm(Transport~.,cars,family = "binomial")  
summary(model)

##   
## Call:  
## glm(formula = Transport ~ ., family = "binomial", data = cars)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.84326 -0.00930 -0.00202 -0.00020 2.21440   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -61.7598 34.3933 -1.796 0.07254 .   
## Age 1.5130 1.1274 1.342 0.17958   
## GenderMale -2.2754 1.7055 -1.334 0.18215   
## Engineer1 0.4954 1.8071 0.274 0.78398   
## MBA1 -1.9522 1.7152 -1.138 0.25505   
## Work.Exp -0.6739 0.8970 -0.751 0.45247   
## Salary 0.2441 0.1827 1.336 0.18163   
## Distance 0.9479 0.3487 2.718 0.00656 \*\*  
## license1 2.7683 2.0922 1.323 0.18577   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 240.42 on 416 degrees of freedom  
## Residual deviance: 22.29 on 408 degrees of freedom  
## AIC: 40.29  
##   
## Number of Fisher Scoring iterations: 11

vif(model)

## Age Gender Engineer MBA Work.Exp Salary Distance   
## 21.262525 2.345914 1.144534 2.458458 28.209208 9.719722 3.091251   
## license   
## 3.671921

As a rule of thumb, a VIF value that exceeds 5 or 10 indicates a problematic amount of collinearity" As hinted in the Correlation Matrix plot,we can clearly see that Work Experience and salary has high Vif Let’s remove the Age,Work Experience and check the VIF for other predictors

cars1 <- cars[,-c(1,5)]  
model1 <- glm(Transport~.,cars1,family = "binomial")  
vif(model1)

## Gender Engineer MBA Salary Distance license   
## 1.352185 1.057026 1.140748 1.556626 1.474680 1.455595

The Vif of all the other variables are around 1,i.e. they are less correlated with each other.

### Key-Insights From EDA

## Uni-Variate Analysis:

Age - Most of our Employee are younger as thier average age is around 27. Gender - Our Employee base has lot of Males(71%) than Females (29%) Engineer - Around 75% of our Employee are engineer graduates and only 25% Employee are non- engineers MBA - Though,we have lot of Engineer graduates as our Employees but we have only 27% of MBA Graduates in our company. Distance - Employee commute Distance ranges from 3.2 to 23.4 kilometers and on an average our Employee commutes a distance of 11.3 kilometers license - Only 20% of our Employee has License, which is quite surprising. Transport - We have 19 % of Employees who commute via thier own two wheelers and 8 % of employees via own car and 71 % of employees via Public Transport

## Bi-Variate Analysis:

Employee whose age above 30 are the ones who use car as a mode of transport and most employees are using public transport only. Out of 297 Male employees only 29 Male employees are driving Car and out of 120 Female employees only 6 Females are commuting via car to the office. Out of 313 Engineer graduates, only 30 graduates are commuting to the office via car and only 5 out of 104 non engineers own a car Out of 109 MBA graduates, only 9 graduates are commuting to the office via car and 26 out of 308 non MBA graduates own a car As expected, Higher the work experience higher the chance of commuting to the office via car. As expected, Higher the Salary higher the chance of commuting to the office via car. Higher the distance, more the possibility of commuting to the office via car. As expected, Most of the non licensed people uses Public Transport as the way of commute to thier offices and 29 out of 85 licensed employee uses car as a mode of tramsport.

## Multi-collinearity:

Then, we figured out that the Work Experience,Salary are highly correlated with the other variables and causing Misintepretation.So we removed them from our Data.

## Data Preparation

Before Building a model, let’s check the imbalance of our Dataset.

prop.table(table(cars1$Transport))

##   
## 0 1   
## 0.91606715 0.08393285

From the above output, it’s pretty evident that we have high unbalanced classifiers and we need to treat the imbalance by SMOTE Method.

## SMOTE (Synthetic Minority Oversampling TEchnique)

library("DMwR")

## Warning: package 'DMwR' was built under R version 3.6.2

## Loading required package: lattice

##   
## Attaching package: 'lattice'

## The following object is masked from 'package:corrgram':  
##   
## panel.fill

## Loading required package: grid

## Registered S3 method overwritten by 'xts':  
## method from  
## as.zoo.xts zoo

## Registered S3 method overwritten by 'quantmod':  
## method from  
## as.zoo.data.frame zoo

smoted\_data <- SMOTE(Transport~.,cars1, perc.over=100, perc.under=600, k=5)  
prop.table(table(smoted\_data$Transport))

##   
## 0 1   
## 0.75 0.25

After we did SMOTE we have increased our minority class level from 8% to 25%. By doing so we have made the data with more balanced classifiers.

## Model Building

Now,Let’s Build the Model with the Smoted data and check how it performs on the training and testing dataset.

library(caTools)# Used for Spliting the Data   
set.seed(1234)  
split <- sample.split(smoted\_data$Transport, SplitRatio = 0.7)  
train <- subset(smoted\_data,split== TRUE)  
test <- subset(smoted\_data,split == FALSE)  
LogTrainModel <- glm(Transport~Gender+Engineer+MBA+Distance+license,train,family = "binomial")  
summary(LogTrainModel)

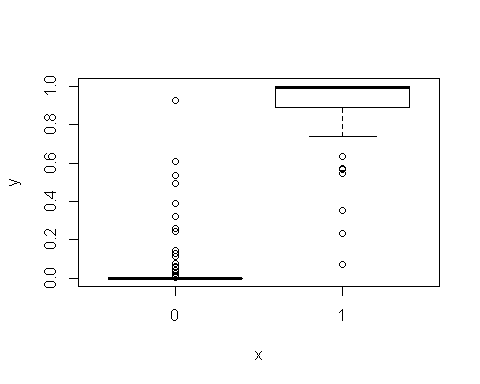
##   
## Call:  
## glm(formula = Transport ~ Gender + Engineer + MBA + Distance +   
## license, family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.28911 -0.02598 -0.00209 0.00014 2.28347   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -31.0795 7.9823 -3.894 9.88e-05 \*\*\*  
## GenderMale -1.1724 0.9554 -1.227 0.219762   
## Engineer1 6.9460 2.3072 3.011 0.002608 \*\*   
## MBA1 -1.9202 1.1312 -1.697 0.089613 .   
## Distance 1.4797 0.3888 3.806 0.000141 \*\*\*  
## license1 6.5546 1.7793 3.684 0.000230 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 220.435 on 195 degrees of freedom  
## Residual deviance: 41.818 on 190 degrees of freedom  
## AIC: 53.818  
##   
## Number of Fisher Scoring iterations: 9

vif(LogTrainModel)

## Gender Engineer MBA Distance license   
## 1.366866 2.571031 1.496295 3.073153 4.183224

Now Let’s see how our model performs on both Training and Test Dataset Logistic regression does not return directly the class of observations. It allows us to estimate the probability (p) of class membership.The probability will range between 0 and 1. We need to decide the threshold probability at which the category flips from one to the other.

Log\_Prediction\_Train <- predict(LogTrainModel,data = "train",type = "response")  
  
plot(train$Transport,Log\_Prediction\_Train)

 From the above Plot,we can clearly see that most employers who use Public Transport and 2-wheeler as a mode of transport lies within 0-0.4.

So, let’s take the threshold of 0.4. The Probabilty predicted by our Model above 0.4 will be taken as 1 (Employees who use car as a mode of transport)

## Model Perfomance on Training Data

Log\_model.predicted <- ifelse(Log\_Prediction\_Train<0.4,0,1)  
Logmodel <- table(train$Transport,Log\_model.predicted)  
print(Logmodel)

## Log\_model.predicted  
## 0 1  
## 0 141 6  
## 1 4 45

## Confusion Matrix

#Accuracy   
accuracy <- round(sum(diag(Logmodel))/sum(Logmodel),2)  
print(accuracy)

## [1] 0.95

# Sensitivity   
sensitivity <- round(44/(44+9),2)  
print(sensitivity)

## [1] 0.83

# 0.83  
# Specificity  
specificity <-round(138/(138 + 5),2)  
print(specificity)

## [1] 0.97

#0.97

Based On Confusion Matrix,With 93% accuracy on the Training Dataset,Our model has done well in predicting both the 0 (0.97) (Employees who use Public Transport and 2-wheeler) and 1 (83%) (Employees who use car as a mode of transport). Now, Let’s Check Our Model with other Model Perfomance measures like AUC, Gini,KS

## AUC & Gini

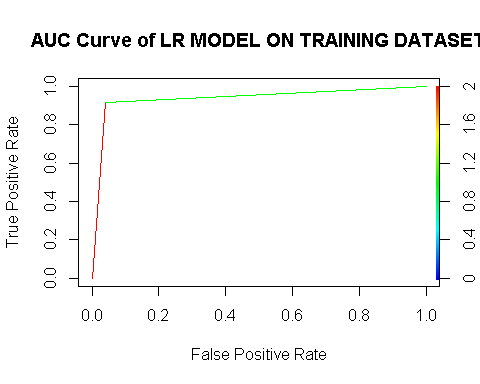
library(ROCR)

## Loading required package: gplots

##   
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':  
##   
## lowess

ROCRpred <- prediction(Log\_model.predicted, train$Transport)  
ROCRperf <- performance(ROCRpred, 'tpr','fpr')  
plot(ROCRperf,colorize = TRUE, text.adj = c(-0.2,1.7),main="AUC Curve of LR MODEL ON TRAINING DATASET",xlab="False Positive Rate",ylab="True Positive Rate")



auc = performance(ROCRpred,"auc");  
auc = as.numeric(auc@y.values)  
print(auc)

## [1] 0.9387755

library(ineq)  
gini = ineq(Log\_model.predicted, type="Gini")  
print(gini)

## [1] 0.7397959

### Thumb Rule - Larger the auc and gini coefficient better the model is.

We have a auc of 92% and gini coefficient of 72% which conveys the message that our model has done a Ok Job in training datset.

## kS

KS Statistic or Kolmogorov-Smirnov statistic is the maximum difference between the cumulative true positive and cumulative false positive rate.It is often used as the deciding metric to judge the efficacy of models in credit scoring. The higher the ks\_stat, the more efficient is the model at capturing the Ones. This should not be confused with the ks.test function.

KS = max(ROCRperf@y.values[[1]]-ROCRperf@x.values[[1]]) # The Maximum the Better   
print(KS)

## [1] 0.877551

Here,In Training Dataset our Logistic Model done a good job (0.83) in Predicting the Employees who use car as a mode of transport.

## Model Performance on Test Data

# Confusion Matrix

Log\_Prediction\_Test <- predict(LogTrainModel,test,type = "response")  
Log\_model.predicted1 <- ifelse(Log\_Prediction\_Test <0.4,0,1)  
Logmodel1 <- table(test$Transport,Log\_model.predicted1)  
print(Logmodel1)

## Log\_model.predicted1  
## 0 1  
## 0 61 2  
## 1 1 20

# Accuracy   
Test\_accuracy <- round(sum(diag(Logmodel1))/sum(Logmodel1),2)  
print(Test\_accuracy)

## [1] 0.96

# Sensitivity   
sensitivity <- round(20/(20+4),2)  
print(sensitivity)

## [1] 0.83

# Specificity  
specificity <-round(59/(59 + 1),2)  
print(specificity)

## [1] 0.98

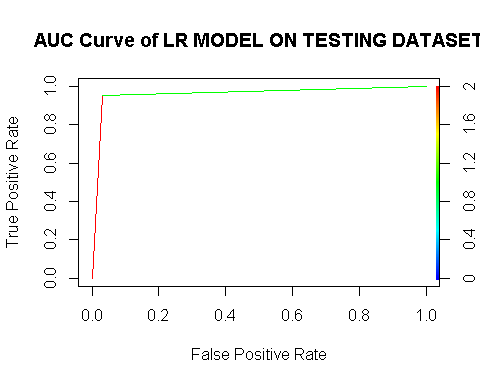
## Confusion Matrix Inference on Test Dataset:

Based On Confusion Matrix,With 94% accuracy on the Training Dataset,Our model has done well in predicting both the 0 (98%) (Employees who use Public Transport and 2-wheeler) and 1 (83%) (Employees who use car as a mode of transport).

Now Let’s Check Our Logistic Model with other Model Perfomance measures like AUC, Gini,KS

# AUC & GINI

TestROCRpred <- prediction(Log\_model.predicted1, test$Transport)  
TestROCRperf <- performance(TestROCRpred, 'tpr','fpr')  
plot(TestROCRperf,colorize = TRUE, text.adj = c(-0.2,1.7),main="AUC Curve of LR MODEL ON TESTING DATASET",xlab="False Positive Rate",ylab="True Positive Rate")



Testauc = performance(TestROCRpred,"auc");  
Testauc = as.numeric(Testauc@y.values)  
print(Testauc)

## [1] 0.9603175

# Gini on Test dataset  
Testgini = ineq(Log\_model.predicted1, type="Gini")  
print(Testgini)

## [1] 0.7380952

## Thumb Rule - Larger the auc and gini coefficient better the model is.

We have a auc of 94% and gini coefficient of 71% which conveys the message that our model has done a good Job in the test datset.

# kS

The higher the ks\_stat, the more efficient is the model at capturing the Ones.

TestKS = max(TestROCRperf@y.values[[1]]-TestROCRperf@x.values[[1]]) # The Maximum the Better   
print(TestKS)

## [1] 0.9206349

Here,In Test Dataset our Logistic Model done Poorly(0.88) in Predicting the employees who will use car as a mode of transport.

Our Model Has almost performed the Sameway in both the train and Test dataset.

Now, Let’s Build a KNN Model and Measure it’s Performance

# KNN

library(class)  
knntrain <- train  
knntest <- test  
knntrain$Gender <- as.numeric(knntrain$Gender)   
knntrain$Engineer <- as.numeric(knntrain$Engineer)  
knntrain$MBA <- as.numeric(knntrain$MBA)  
knntrain$license <- as.numeric(knntrain$license)  
knntrain$Transport <- as.numeric(knntrain$Transport)  
str(knntrain)

## 'data.frame': 196 obs. of 7 variables:  
## $ Gender : num 1 1 2 2 1 2 1 2 2 2 ...  
## $ Engineer : num 2 2 2 2 1 2 2 2 2 2 ...  
## $ MBA : num 1 1 1 1 1 1 1 1 1 1 ...  
## $ Salary : num 11.5 21.7 14.8 12.7 6.8 22.7 9.6 9.9 13.9 12.9 ...  
## $ Distance : num 5.2 7.3 14.3 8.7 12.2 11.3 8.1 17.2 9.5 13.3 ...  
## $ license : num 1 1 1 1 1 2 1 1 1 1 ...  
## $ Transport: num 1 1 1 1 1 1 1 1 1 1 ...

knntest$Gender <- as.numeric(knntest$Gender)   
knntest$Engineer <- as.numeric(knntest$Engineer)  
knntest$MBA <- as.numeric(knntest$MBA)  
knntest$license <- as.numeric(knntest$license)  
knntest$Transport <- as.numeric(knntest$Transport)  
str(knntest)

## 'data.frame': 84 obs. of 7 variables:  
## $ Gender : num 1 2 2 2 1 2 1 2 1 1 ...  
## $ Engineer : num 2 2 1 1 2 2 1 2 2 2 ...  
## $ MBA : num 1 1 1 1 1 2 2 2 1 1 ...  
## $ Salary : num 14.6 8.6 6.9 8.7 12.8 11.7 8.5 13.5 14.7 12.8 ...  
## $ Distance : num 8.1 9.4 13.7 8.4 13.6 11.7 7.9 8.8 8.5 11.8 ...  
## $ license : num 1 1 1 1 2 1 1 1 1 1 ...  
## $ Transport: num 1 1 1 1 1 1 1 1 1 1 ...

knntrain$Transport[knntrain$Transport == 1] <- 0  
knntrain$Transport[knntrain$Transport == 2] <- 1  
knntrain$Gender[knntrain$Gender == 1] <- 0  
knntrain$Gender[knntrain$Gender == 2] <- 1  
knntrain$Engineer[knntrain$Engineer == 1] <- 0  
knntrain$Engineer[knntrain$Engineer == 2] <- 1  
knntrain$MBA[knntrain$MBA == 1] <- 0  
knntrain$MBA[knntrain$MBA == 2] <- 1  
knntrain$Salary[knntrain$Salary == 1] <- 0  
knntrain$Salary[knntrain$Salary == 2] <- 1  
knntrain$Distance[knntrain$Distance == 1] <- 0  
knntrain$Distance[knntrain$Distance == 2] <- 1  
knntrain$license[knntrain$license == 1] <- 0  
knntrain$license[knntrain$license == 2] <- 1  
  
knntest$Transport[knntest$Transport == 1] <- 0  
knntest$Transport[knntest$Transport == 2] <- 1  
knntest$Gender[knntest$Gender == 1] <- 0  
knntest$Gender[knntest$Gender == 2] <- 1  
knntest$Engineer[knntest$Engineer == 1] <- 0  
knntest$Engineer[knntest$Engineer == 2] <- 1  
knntest$MBA[knntest$MBA == 1] <- 0  
knntest$MBA[knntest$MBA == 2] <- 1  
knntest$Salary[knntest$Salary == 1] <- 0  
knntest$Salary[knntest$Salary == 2] <- 1  
knntest$Distance[knntest$Distance == 1] <- 0  
knntest$Distance[knntest$Distance == 2] <- 1  
knntest$license[knntest$license == 1] <- 0  
knntest$license[knntest$license == 2] <- 1  
  
knnmodel <- knn(scale(knntrain),scale(knntest),knntrain$Transport,k=17)  
summary(knnmodel)

## 0 1   
## 63 21

### Interpretation:

After Trail and Error Method @ k = 17 the Model performs well in predicting Both 0 (Customer who wil-l not cancel) and 1 (Customer who will cancel) when compared to Logistic Regression model. Our Model Predicted 64 ‘0’ and 20 ‘1’.Now, Let’s Check how well it have performed by using Confusion Matrix

## Confusion Matrix

knntable <- table(test$Transport,knnmodel)  
print(knntable)

## knnmodel  
## 0 1  
## 0 62 1  
## 1 1 20

# Accuracy   
knnaccuracy <- round(sum(diag(knntable))/sum(knntable),2)  
print(knnaccuracy)

## [1] 0.98

# Sensitivity   
sensitivity <- round(20/(20+0),2)  
print(sensitivity)

## [1] 1

# Specificity  
specificity <-round(63/(63 + 1),2)  
print(specificity)

## [1] 0.98

With 99% accuracy our KNN-Model has Done well in predicting both the 0 (98%) (employees who use 2wheeler and car as a mode of transport to the office) and 1 (99.9%) (Employees who are using car as a mode of transport).

Let’s Look, How Naive Bayes Model works on this Dataset.

## Naive Bayes

The Naive Bayes is a classification algorithm that is suitable for binary and multiclass classification. Generally,Naïve Bayes performs well in cases of categorical input variables compared to numerical variables. so therefore we can use Naive Bayes Algorithm for this use case.Let’s Build the model and see its performance on the Train and Test data.

library(e1071)  
NBModel <- naiveBayes(Transport~Gender+Engineer+MBA+Distance+license,data = train)  
print(NBModel)

##   
## Naive Bayes Classifier for Discrete Predictors  
##   
## Call:  
## naiveBayes.default(x = X, y = Y, laplace = laplace)  
##   
## A-priori probabilities:  
## Y  
## 0 1   
## 0.75 0.25   
##   
## Conditional probabilities:  
## Gender  
## Y Female Male  
## 0 0.3401361 0.6598639  
## 1 0.2448980 0.7551020  
##   
## Engineer  
## Y 0 1  
## 0 0.2653061 0.7346939  
## 1 0.1020408 0.8979592  
##   
## MBA  
## Y 0 1  
## 0 0.8231293 0.1768707  
## 1 0.7346939 0.2653061  
##   
## Distance  
## Y [,1] [,2]  
## 0 10.61088 3.550982  
## 1 17.63776 2.266136  
##   
## license  
## Y 0 1  
## 0 0.8775510 0.1224490  
## 1 0.1836735 0.8163265

NBPredictTrain <- predict(NBModel,newdata = train)

The model creates the conditional probability for each feature separately. We also have the a-priori probabilities which indicates the distribution of our data. Let’s see how the model performs on the Training data.

# Confusion Matrix on Train Dataset

NBTrainTable <- table(train$Transport,NBPredictTrain)  
print(NBTrainTable)

## NBPredictTrain  
## 0 1  
## 0 138 9  
## 1 8 41

# Accuracy   
NBTrainaccuracy <- round(sum(diag(NBTrainTable))/sum(NBTrainTable),2)  
print(NBTrainaccuracy)

## [1] 0.91

# Sensitivity   
NBTrainsensitivity <- round(42/(42+4),2)  
print(NBTrainsensitivity)

## [1] 0.91

# Specificity  
NBTrainspecificity <-round(143/(143 + 7),2)  
print(NBTrainspecificity)

## [1] 0.95

Based On Confusion Matrix,With 94% accuracy on the Training Dataset Our Naive Bayes Model Done well in predicting both the 0 (95%) and 1 (91%)

Let’s Look, how the Naive Bayes Model performs on the Test Data set

NBTestPredict <- predict(NBModel,newdata = test)

## Confusion Matrix on Test Dataset

NBTestTable <- table(test$Transport,NBTestPredict)  
print(NBTestTable)

## NBTestPredict  
## 0 1  
## 0 61 2  
## 1 1 20

# Accuracy   
NBTestaccuracy <- round(sum(diag(NBTestTable))/sum(NBTestTable),2)  
print(NBTestaccuracy)

## [1] 0.96

# Sensitivity   
NBTestsensitivity <- round(20/(20+1),2)  
print(NBTestsensitivity)

## [1] 0.95

# Specificity  
NBTestspecificity <-round(62/(62 + 1),2)  
print(NBTestspecificity)

## [1] 0.98

Based On Confusion Matrix,With 98% accuracy on the Test Dataset Our Naive Bayes Model Done well in predicting both the 0 (98%) and 1 (95%)

Let’s check out how bagging and boosting models perform in this dataset # Bagging

Bagging is a way to decrease the variance in the prediction by generating additional data for training from dataset using combinations with repetitions to produce multi-sets of the original data

library(gbm) # basic implementation using AdaBoost

## Warning: package 'gbm' was built under R version 3.6.2

## Loaded gbm 2.1.5

library(xgboost) # a faster implementation of a gbm

## Warning: package 'xgboost' was built under R version 3.6.2

library(caret) # an aggregator package for performing many machine learning models  
library(ipred)  
library(rpart)  
  
bagging = bagging(Transport~ Gender+Engineer+MBA+Distance+license,data=train,control=rpart.control(maxdepth = 5,minsplit =15 ))  
  
Bagging\_Prediction = predict(bagging,test)  
  
Bagging\_CM=table(test$Transport,Bagging\_Prediction)  
Bagging\_CM

## Bagging\_Prediction  
## 0 1  
## 0 62 1  
## 1 1 20

# Model performance for bagging

#specificity  
bag\_Specificity = round(61/(61 + 2),2)  
bag\_Specificity

## [1] 0.97

#sensitivity  
  
bag\_Sensitiviity=round(19/(19 + 2),2)  
bag\_Sensitiviity

## [1] 0.9

#accuracy  
bag\_Accuracy=round(sum(diag(Bagging\_CM))/sum(Bagging\_CM),2)  
bag\_Accuracy

## [1] 0.98

Based On Confusion Matrix,With 95% accuracy on the Test Dataset Our Bagging Model has done well in predicting both the 0 (97%) and 1 (90%)

#ROC Curve  
library(pROC)

## Type 'citation("pROC")' for a citation.

##   
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':  
##   
## cov, smooth, var

test$Transport =as.numeric(test$Transport)  
Bagging\_Prediction=as.numeric(Bagging\_Prediction)  
roc(test$Transport,Bagging\_Prediction)

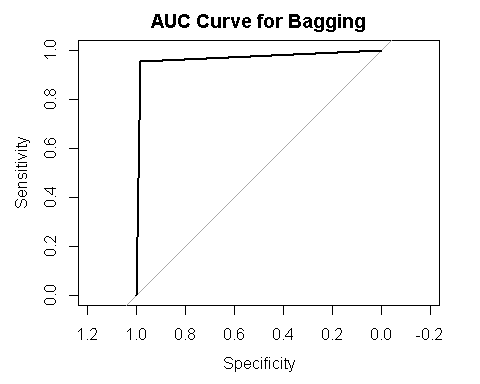
## Setting levels: control = 1, case = 2

## Setting direction: controls < cases

##   
## Call:  
## roc.default(response = test$Transport, predictor = Bagging\_Prediction)  
##   
## Data: Bagging\_Prediction in 63 controls (test$Transport 1) < 21 cases (test$Transport 2).  
## Area under the curve: 0.9683

plot.roc(test$Transport,Bagging\_Prediction, main="AUC Curve for Bagging")

## Setting levels: control = 1, case = 2  
## Setting direction: controls < cases

 # Boosting

Now let’s try some general boosting techniques.

str(train)

## 'data.frame': 196 obs. of 7 variables:  
## $ Gender : Factor w/ 2 levels "Female","Male": 1 1 2 2 1 2 1 2 2 2 ...  
## $ Engineer : Factor w/ 2 levels "0","1": 2 2 2 2 1 2 2 2 2 2 ...  
## $ MBA : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...  
## $ Salary : num 11.5 21.7 14.8 12.7 6.8 22.7 9.6 9.9 13.9 12.9 ...  
## $ Distance : num 5.2 7.3 14.3 8.7 12.2 11.3 8.1 17.2 9.5 13.3 ...  
## $ license : Factor w/ 2 levels "0","1": 1 1 1 1 1 2 1 1 1 1 ...  
## $ Transport: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...

boosttrain <- train  
boosttest <- test  
boosttrain$Gender <- as.numeric(boosttrain$Gender)   
boosttrain$Engineer <- as.numeric(boosttrain$Engineer)  
boosttrain$MBA <- as.numeric(boosttrain$MBA)  
boosttrain$license <- as.numeric(boosttrain$license)  
boosttrain$Transport <- as.numeric(boosttrain$Transport)  
str(boosttrain)

## 'data.frame': 196 obs. of 7 variables:  
## $ Gender : num 1 1 2 2 1 2 1 2 2 2 ...  
## $ Engineer : num 2 2 2 2 1 2 2 2 2 2 ...  
## $ MBA : num 1 1 1 1 1 1 1 1 1 1 ...  
## $ Salary : num 11.5 21.7 14.8 12.7 6.8 22.7 9.6 9.9 13.9 12.9 ...  
## $ Distance : num 5.2 7.3 14.3 8.7 12.2 11.3 8.1 17.2 9.5 13.3 ...  
## $ license : num 1 1 1 1 1 2 1 1 1 1 ...  
## $ Transport: num 1 1 1 1 1 1 1 1 1 1 ...

boosttest$Gender <- as.numeric(boosttest$Gender)   
boosttest$Engineer <- as.numeric(boosttest$Engineer)  
boosttest$MBA <- as.numeric(boosttest$MBA)  
boosttest$license <- as.numeric(boosttest$license)  
boosttest$Transport <- as.numeric(boosttest$Transport)  
str(boosttest)

## 'data.frame': 84 obs. of 7 variables:  
## $ Gender : num 1 2 2 2 1 2 1 2 1 1 ...  
## $ Engineer : num 2 2 1 1 2 2 1 2 2 2 ...  
## $ MBA : num 1 1 1 1 1 2 2 2 1 1 ...  
## $ Salary : num 14.6 8.6 6.9 8.7 12.8 11.7 8.5 13.5 14.7 12.8 ...  
## $ Distance : num 8.1 9.4 13.7 8.4 13.6 11.7 7.9 8.8 8.5 11.8 ...  
## $ license : num 1 1 1 1 2 1 1 1 1 1 ...  
## $ Transport: num 1 1 1 1 1 1 1 1 1 1 ...

boosttrain$Transport[boosttrain$Transport == 1] <- 0  
boosttrain$Transport[boosttrain$Transport == 2] <- 1  
boosttrain$Gender[boosttrain$Gender == 1] <- 0  
boosttrain$Gender[boosttrain$Gender == 2] <- 1  
boosttrain$Engineer[boosttrain$Engineer == 1] <- 0  
boosttrain$Engineer[boosttrain$Engineer == 2] <- 1  
boosttrain$MBA[boosttrain$MBA == 1] <- 0  
boosttrain$MBA[boosttrain$MBA == 2] <- 1  
boosttrain$Salary[boosttrain$Salary == 1] <- 0  
boosttrain$Salary[boosttrain$Salary == 2] <- 1  
boosttrain$Distance[boosttrain$Distance == 1] <- 0  
boosttrain$Distance[boosttrain$Distance == 2] <- 1  
boosttrain$license[boosttrain$license == 1] <- 0  
boosttrain$license[boosttrain$license == 2] <- 1  
  
boosttest$Transport[boosttest$Transport == 1] <- 0  
boosttest$Transport[boosttest$Transport == 2] <- 1  
boosttest$Gender[boosttest$Gender == 1] <- 0  
boosttest$Gender[boosttest$Gender == 2] <- 1  
boosttest$Engineer[boosttest$Engineer == 1] <- 0  
boosttest$Engineer[boosttest$Engineer == 2] <- 1  
boosttest$MBA[boosttest$MBA == 1] <- 0  
boosttest$MBA[boosttest$MBA == 2] <- 1  
boosttest$Salary[boosttest$Salary == 1] <- 0  
boosttest$Salary[boosttest$Salary == 2] <- 1  
boosttest$Distance[boosttest$Distance == 1] <- 0  
boosttest$Distance[boosttest$Distance == 2] <- 1  
boosttest$license[boosttest$license == 1] <- 0  
boosttest$license[boosttest$license == 2] <- 1  
  
boost\_model=gbm(Transport ~ Gender+Engineer+MBA+Distance+license,distribution = "bernoulli",data=boosttrain,n.trees = 100,interaction.depth =1,shrinkage = 0.001,cv.folds = 5,n.cores=NULL,verbose=FALSE)  
  
boost\_prediction <- predict(boost\_model,boosttest,type="response")

## Using 100 trees...

boost\_prediction <-ifelse(boost\_prediction>0.27,"1","0")  
print(boost\_prediction)

## [1] "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0"  
## [18] "0" "0" "0" "0" "0" "0" "1" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0"  
## [35] "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0"  
## [52] "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "1"  
## [69] "1" "1" "1" "1" "1" "1" "1" "0" "0" "0" "1" "1" "1" "0" "1" "1"

boost\_CM= table(boosttest$Transport,boost\_prediction)  
print(boost\_CM)

## boost\_prediction  
## 0 1  
## 0 62 1  
## 1 8 13

# Confusion Matrix

#specificity  
boost\_Specificity = round(55/(55 + 3),2)  
print(boost\_Specificity)

## [1] 0.95

#sensitivity  
  
bag\_Sensitiviity=round(18/(18 + 8),2)  
print(bag\_Sensitiviity)

## [1] 0.69

#accuracy  
boost\_Accuracy=round(sum(diag(boost\_CM))/sum(boost\_CM),2)  
print(boost\_Accuracy)

## [1] 0.89

Based On Confusion Matrix,With 87% accuracy on the Test Dataset Our Boosting Model has done well in predicting the 0 (95%) than in predicting 1 (69%)

# ROC Curve

boosttest$Transport <- as.numeric(boosttest$Transport)  
boost\_prediction <- as.numeric(boost\_prediction)  
roc(test$Transport,boost\_prediction)

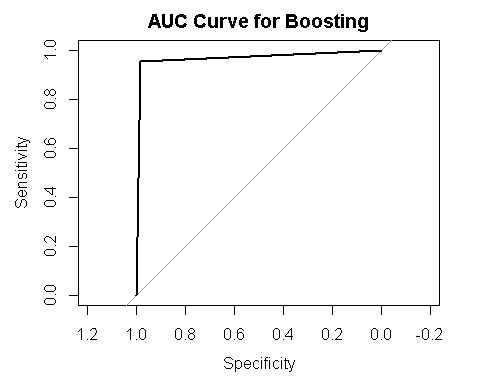
## Setting levels: control = 1, case = 2

## Setting direction: controls < cases

##   
## Call:  
## roc.default(response = test$Transport, predictor = boost\_prediction)  
##   
## Data: boost\_prediction in 63 controls (test$Transport 1) < 21 cases (test$Transport 2).  
## Area under the curve: 0.8016

plot.roc(test$Transport,Bagging\_Prediction, main="AUC Curve for Boosting")

## Setting levels: control = 1, case = 2  
## Setting direction: controls < cases



library(fastAdaboost)

## Warning: package 'fastAdaboost' was built under R version 3.6.2

library(xgboost)  
  
features\_train = as.matrix(boosttrain[,-7])  
label\_train = as.matrix(boosttrain[,-c(1:6)])  
features\_test = as.matrix(boosttest[,-7])  
tp\_xbg=vector()  
lr=c(0.001,0.01,0.1,0.3,0.5,0.7,1)  
md=c(1,3,5,7,9,15)  
nr=c(2,50,100,1000,1000)  
for (i in lr) {  
 xgb.fit=xgboost(  
 data = features\_train,  
 label= label\_train,  
 eta = 0.001,  
 max\_depth = 5,  
 nrounds = 10,  
 nfold=5,  
 objective = "binary:logistic",   
 verbose = 0,  
 early\_stopping\_rounds = 10  
 )   
 XGBpredTest=boosttest$xgb.pred = predict(xgb.fit, features\_test)  
 sum(boosttest$Transport==1&boosttest$xgb.pred>=0.5)  
 tabXGB=table(boosttest$Transport, XGBpredTest>0.5)  
}  
xgboost\_CM <- table(boosttest$Transport,boosttest$xgb.pred>=0.5)

# XG Boost Confusion Matrix

#specificity  
xgboost\_Specificity = round(63/(63 + 3),2)  
print(xgboost\_Specificity)

## [1] 0.95

#sensitivity  
  
xgboost\_Sensitiviity=round(18/(18 + 0),2)  
print(xgboost\_Sensitiviity)

## [1] 1

#accuracy  
xgboost\_Accuracy=round(sum(diag(xgboost\_CM))/sum(xgboost\_CM),2)  
print(boost\_Accuracy)

## [1] 0.89

Based On Confusion Matrix,with 87% accuracy on the Test Dataset Our Boosting Model has done well in predicting the 0 (95%) than in predicting 1 (100%)