################################Random Forest#####################################

#Problem Statement:

# A cloth manufacturing company is interested to know about the segment or attributes causes high sale.Approach - A Random Forest can be built with target variable Sales (we will first convert it in categorical variable) & all other variable will be independent in the analysis.

##################################################################################

install.packages("caret",dependencies = TRUE)

install.packages("randomForest")

library(randomForest)

library(caret)

#Company\_Data.csv

companyData<-read.csv(file.choose())

head(companyData)

#Sales CompPrice Income Advertising Population Price ShelveLoc Age Education Urban US

#1 9.50 138 73 11 276 120 Bad 42 17 Yes Yes

#2 11.22 111 48 16 260 83 Good 65 10 Yes Yes

#3 10.06 113 35 10 269 80 Medium 59 12 Yes Yes

#4 7.40 117 100 4 466 97 Medium 55 14 Yes Yes

#5 4.15 141 64 3 340 128 Bad 38 13 Yes No

#6 10.81 124 113 13 501 72 Bad 78 16 No Yes

set.seed(234)

highSales <- ifelse(companyData$Sales>10,"Yes","No")

companyDetails <- data.frame(companyData[2:11],highSales)

View(companyDetails)

table(companyDetails$highSales)

#No Yes

#322 78

#Training and test data

index\_row <- sample(2,nrow(companyDetails),replace = T,prob=c(0.7,0.3))

train\_data <- companyDetails[index\_row == 1,]

test\_data <- companyDetails[index\_row == 2,]

#Building the model with training data

rfmodel <- randomForest(highSales ~.,data=train\_data)

#randomForest(formula = highSales ~ ., data = train\_data)

#Type of random forest: classification

#Number of trees: 500

#No. of variables tried at each split: 3

#OOB estimate of error rate: 15.49%

#Confusion matrix:

# No Yes class.error

#No 218 9 0.03964758

#Yes 35 22 0.61403509

#prediction on training data

pred1 <- predict(rfmodel,train\_data)

head(pred1)

#4 5 6 7 8 10

#No No Yes No Yes No

#Levels: No Yes

head(train\_data$highSales)

#[1] No No Yes No Yes No

#Levels: No Yes

#First six original and predicted values matches

confusionMatrix(pred1,train\_data$highSales)

#Confusion Matrix and Statistics

#Reference

#Prediction No Yes

#No 227 0

#Yes 0 57

#Accuracy : 1

#95% CI : (0.9871, 1)

#No Information Rate : 0.7993

#P-Value [Acc > NIR] : < 2.2e-16

#Kappa : 1

#Mcnemar's Test P-Value : NA

# Sensitivity : 1.0000

# Specificity : 1.0000

# Pos Pred Value : 1.0000

# Neg Pred Value : 1.0000

# Prevalence : 0.7993

# Detection Rate : 0.7993

# Detection Prevalence : 0.7993

# Balanced Accuracy : 1.0000

# 'Positive' Class : No

#100 % accuracy on training data. 95 % confidence interval, sensitivity for Yes and No

# No is 100%

#prediction on test data

pred2 <- predict(rfmodel,test\_data)

confusionMatrix(pred2,test\_data$highSales)

#Confusion Matrix and Statistics

#Reference

#Prediction No Yes

#No 91 11

#Yes 4 10

#Accuracy : 0.8707

#95% CI : (0.7957, 0.9258)

#No Information Rate : 0.819

#P-Value [Acc > NIR] : 0.08853

#Kappa : 0.4988

#Mcnemar's Test P-Value : 0.12134

# Sensitivity : 0.9579

# Specificity : 0.4762

# Pos Pred Value : 0.8922

# Neg Pred Value : 0.7143

# Prevalence : 0.8190

# Detection Rate : 0.7845

# Detection Prevalence : 0.8793

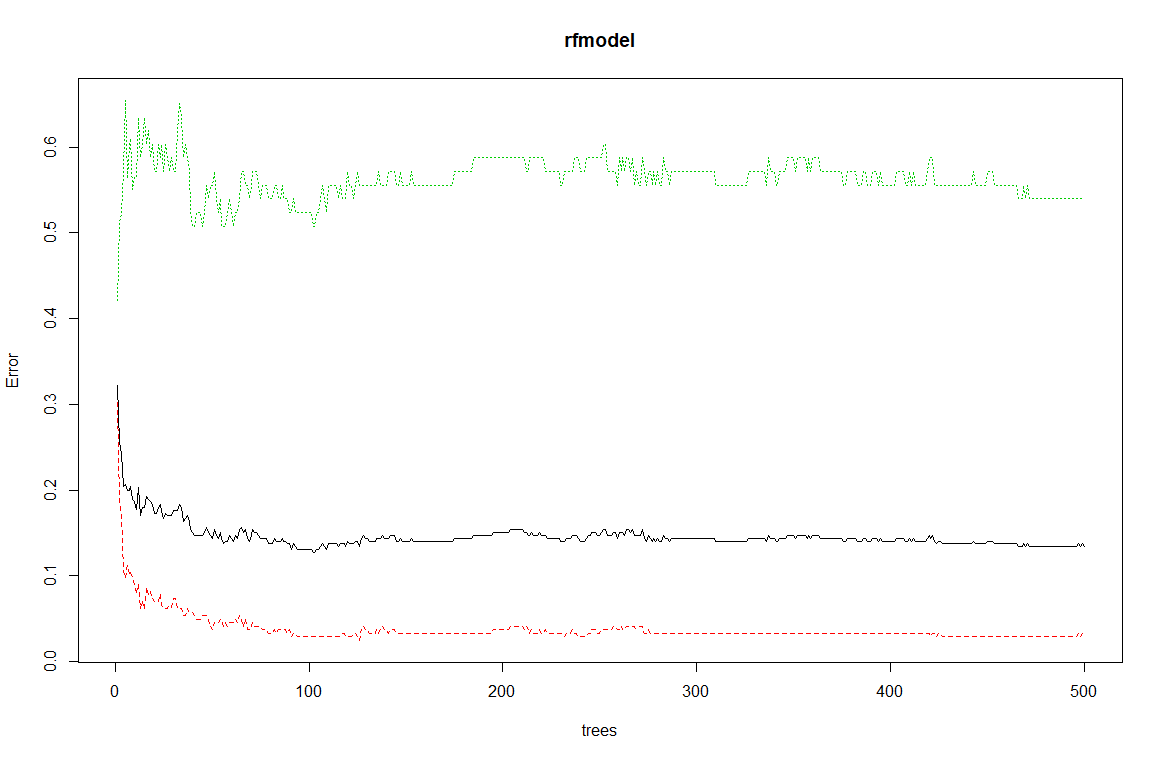
# Balanced Accuracy : 0.7170

#'Positive' Class : No

#87% accuracy on test data with 95% coinfidence interval.

#error rate in random forest

plot(rfmodel)



#tune random forest

tunerf <-tuneRF(train\_data[,-11],train\_data[,11],stepFactor = 0.5,plot=TRUE,ntreeTry = 300,

trace=TRUE,improve=0.05)

#mtry = 3 OOB error = 15.14%

#Searching left ...

#mtry = 6 OOB error = 13.73%

#0.09302326 0.05

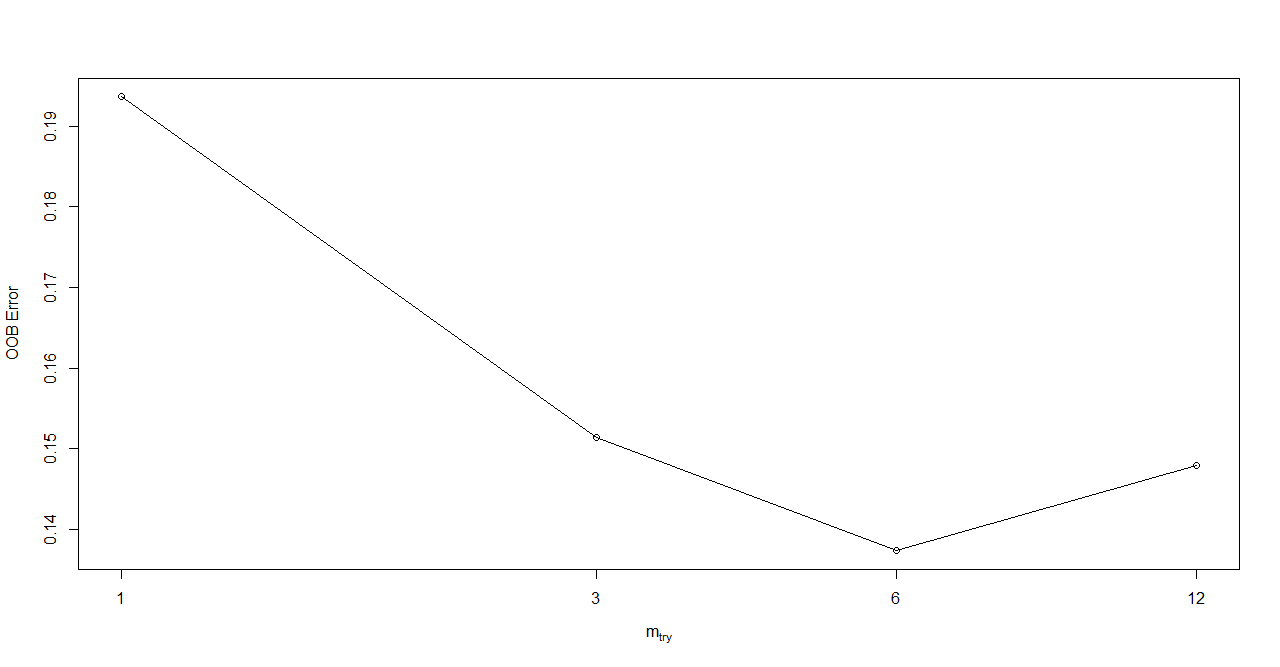
#mtry = 12 OOB error = 14.79%

#-0.07692308 0.05

#Searching right ...

#mtry = 1 OOB error = 19.37%

#-0.4102564 0.05



#building model with mtry=6 and ntree = 300

rfmodel1 <- randomForest(highSales~.,data=train\_data,ntree=300,mtry=6,importance=T)

#Type of random forest: classification

#Number of trees: 300

#No. of variables tried at each split: 6

#OOB estimate of error rate: 13.73%

#Confusion matrix:

# No Yes class.error

#No 217 10 0.04405286

#Yes 29 28 0.50877193

pred1 <- predict(rfmodel1,train\_data)

confusionMatrix(pred1,train\_data$highSales)

# 100 accuracy on training data with 95% confidence interval.

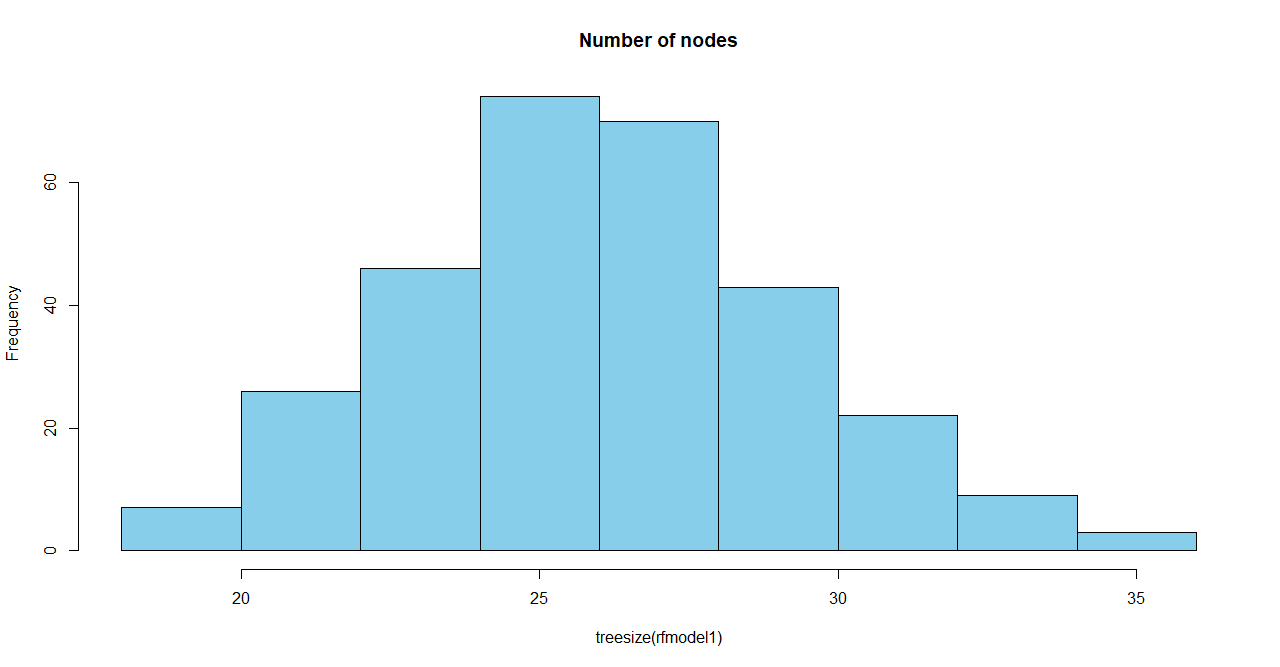
# test data prediction using tuned random forest model

pred2 <- predict(rfmodel1,test\_data)

confusionMatrix(pred2,test\_data$highSales)

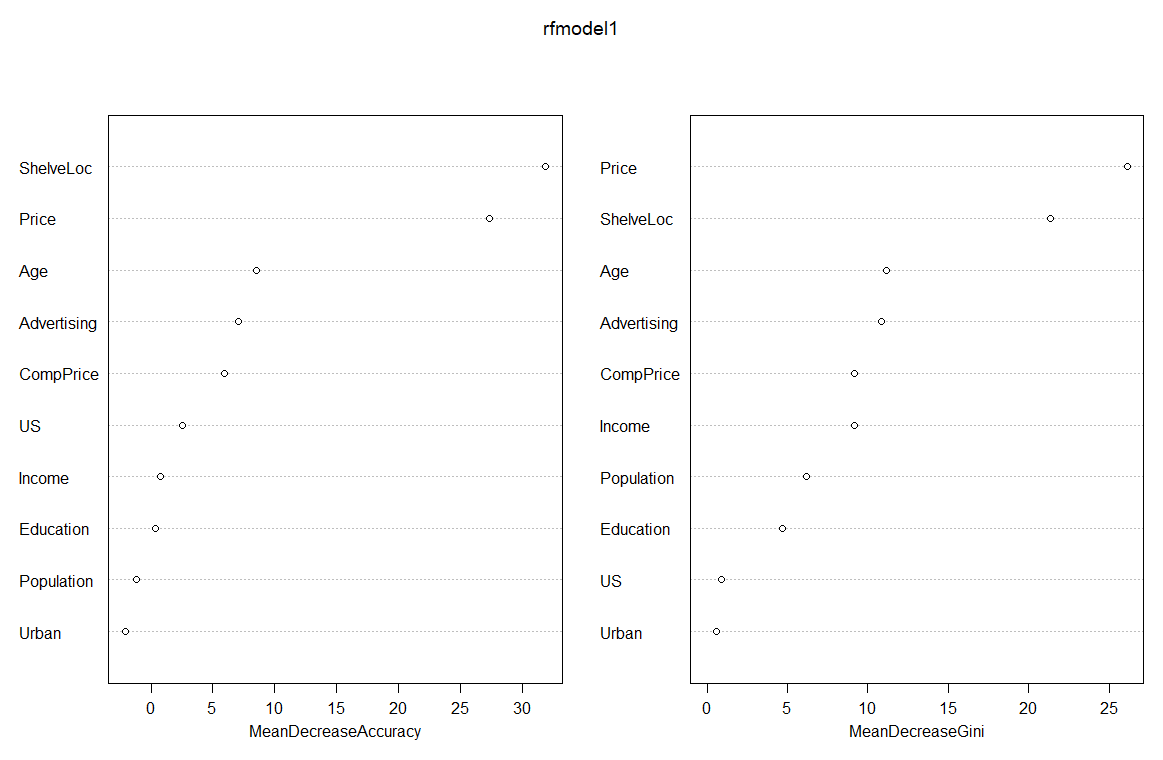
#87% accuracy with 95% confidence interval.

hist(treesize(rfmodel1),main="Number of nodes",col="skyblue")



#MOst of the tree has an average of 30 to 35 nodes.

varImpPlot(rfmodel1)



#MeanDecreaseAccuracy shows Shelveloc is most important variable for prediction,

#population & education have least value.

#MeanDecreaseGini.Price is important feature for prediction where as Urban dont effect.

varImpPlot(rfmodel1,Sort=T,n.var=5,main="Top 5 important variable")

