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.

> library(randomForest)

> library(caret)

> company <- read.csv(file.choose())

> View(company)

> summary(company)

Sales CompPrice Income Advertising

Min. : 0.000 Min. : 77 Min. : 21.00 Min. : 0.000

1st Qu.: 5.390 1st Qu.:115 1st Qu.: 42.75 1st Qu.: 0.000

Median : 7.490 Median :125 Median : 69.00 Median : 5.000

Mean : 7.496 Mean :125 Mean : 68.66 Mean : 6.635

3rd Qu.: 9.320 3rd Qu.:135 3rd Qu.: 91.00 3rd Qu.:12.000

Max. :16.270 Max. :175 Max. :120.00 Max. :29.000

Population Price ShelveLoc Age

Min. : 10.0 Min. : 24.0 Bad : 96 Min. :25.00

1st Qu.:139.0 1st Qu.:100.0 Good : 85 1st Qu.:39.75

Median :272.0 Median :117.0 Medium:219 Median :54.50

Mean :264.8 Mean :115.8 Mean :53.32

3rd Qu.:398.5 3rd Qu.:131.0 3rd Qu.:66.00

Max. :509.0 Max. :191.0 Max. :80.00

Education Urban US

Min. :10.0 No :118 No :142

1st Qu.:12.0 Yes:282 Yes:258

Median :14.0

Mean :13.9

3rd Qu.:16.0

Max. :18.0

> str(company)

'data.frame': 400 obs. of 11 variables:

$ Sales : num 9.5 11.22 10.06 7.4 4.15 ...

$ CompPrice : int 138 111 113 117 141 124 115 136 132 132 ...

$ Income : int 73 48 35 100 64 113 105 81 110 113 ...

$ Advertising: int 11 16 10 4 3 13 0 15 0 0 ...

$ Population : int 276 260 269 466 340 501 45 425 108 131 ...

$ Price : int 120 83 80 97 128 72 108 120 124 124 ...

$ ShelveLoc : Factor w/ 3 levels "Bad","Good","Medium": 1 2 3 3 1 1 3 2 3 3 ...

$ Age : int 42 65 59 55 38 78 71 67 76 76 ...

$ Education : int 17 10 12 14 13 16 15 10 10 17 ...

$ Urban : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 1 2 2 1 1 ...

$ US : Factor w/ 2 levels "No","Yes": 2 2 2 2 1 2 1 2 1 2 ...

> attach(company)

The following objects are masked from company (pos = 3):

Advertising, Age, CompPrice, Education, Income, Population,

Price, Sales, ShelveLoc, Urban, US

> sales\_cat <- ifelse(Sales>8.5,"high","low")

> company <- data.frame(sales\_cat,company[,-1])

> View(company)

#Splitting data in Train and test.

> cutt <- createDataPartition(sales\_cat,p=0.7,list=F)

> train\_comp <- company[cutt,]

> test\_comp <- company[-cutt,]

#Model Building

> companyforest <- randomForest(sales\_cat~.,ntree=500,mtry=3,data = train\_comp,importnce=T)

> companyforest

Call:

randomForest(formula = sales\_cat ~ ., data = train\_comp, ntree = 500, mtry = 3, importnce = T)

Type of random forest: classification

Number of trees: 500

No. of variables tried at each split: 3

OOB estimate of error rate: 16.73%

Confusion matrix:

high low class.error

high 70 31 0.30693069

low 16 164 0.08888889

#Prediction and accuracy based on train data

> pred\_train <- predict(companyforest,train\_comp)

> mean(pred\_train==train\_comp$sales\_cat) # acc = 100%

[1] 1

> confusionMatrix(pred\_train,train\_comp$sales\_cat)

Confusion Matrix and Statistics

Reference

Prediction high low

high 101 0

low 0 180

Accuracy : 1

95% CI : (0.987, 1)

No Information Rate : 0.6406

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 : 0.3594

Detection Rate : 0.3594

Detection Prevalence : 0.3594

Balanced Accuracy : 1.0000

'Positive' Class : high

#Prediction and accuracy based on test data

> pred\_test <- predict(companyforest,test\_comp)

> mean(pred\_test==test\_comp$sales\_cat) # acc = 79.83%

[1] 0.7983193

> confusionMatrix(pred\_test,test\_comp$sales\_cat)

Confusion Matrix and Statistics

Reference

Prediction high low

high 21 3

low 21 74

Accuracy : 0.7983

95% CI : (0.7149, 0.8663)

No Information Rate : 0.6471

P-Value [Acc > NIR] : 0.0002380

Kappa : 0.5108

Mcnemar's Test P-Value : 0.0005202

Sensitivity : 0.5000

Specificity : 0.9610

Pos Pred Value : 0.8750

Neg Pred Value : 0.7789

Prevalence : 0.3529

Detection Rate : 0.1765

Detection Prevalence : 0.2017

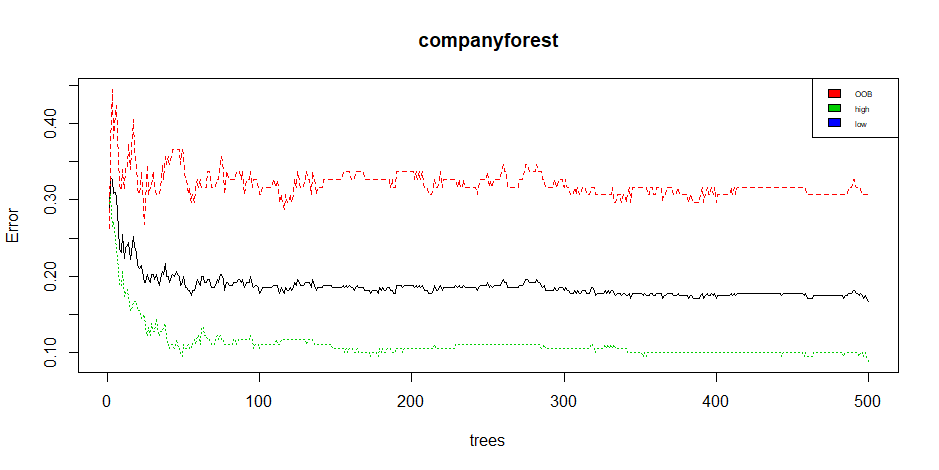
Balanced Accuracy : 0.7305

'Positive' Class : high

#Data Visualisation

> plot(companyforest)

> legend("topright",col = 2:11,colnames(companyforest$err.rate),fill = 2:11,cex = 0.5)



#Variable importance

> importance(companyforest)

MeanDecreaseGini

CompPrice 14.295512

Income 11.265232

Advertising 12.812723

Population 11.496819

Price 32.075575

ShelveLoc 22.143516

Age 14.401068

Education 6.811421

Urban 1.641995

US 1.807325

#From above observation, Price is the most significant variable

#Bagging

>a <- c()

>for(i in 3:10){

set.seed(100)

bag <- createDataPartition(sales\_cat,p=0.8,list = F)

train\_bag <- company[bag,]

test\_bag <- company[-bag,]

bag\_model <- randomForest(sales\_cat~.,data = train\_bag,mtry=i,importance=TRUE)

pred\_bag <- predict(bag\_model,test\_bag,type='class')

a[i-2] <- mean(pred\_bag==test\_bag$sales\_cat)

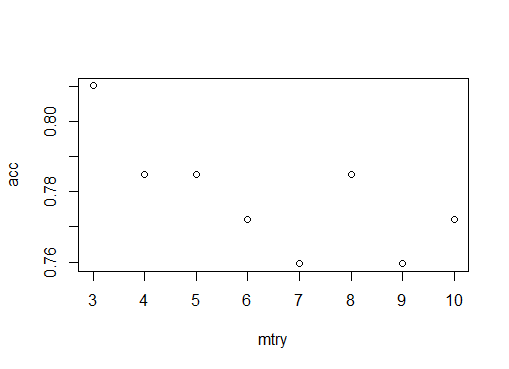
}

> a

[1] 0.8101266 0.7848101 0.7848101 0.7721519 0.7594937 0.7848101 0.7594937

[8] 0.7721519

> plot(3:10,a,xlab = "mtry",ylab = "acc")



#We get highest accuracy for mtry = 3

#So Building our final model based on highest accuracy for mtry = 3

> finalmodel <- randomForest(sales\_cat~.,data = train\_comp,mtry=3,importance=TRUE)

> finalmodel

Call:

randomForest(formula = sales\_cat ~ ., data = train\_comp, mtry = 3, importance = TRUE)

Type of random forest: classification

Number of trees: 500

No. of variables tried at each split: 3

OOB estimate of error rate: 18.51%

Confusion matrix:

high low class.error

high 65 36 0.35643564

low 16 164 0.08888889

#Accuracy of model will be

> mean(predict(finalmodel,test\_comp)==test\_comp$sales\_cat)

[1] 0.8151261