**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)

install.packages("e1071")

library(psych)

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

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 Length:400 Min. :25.00

1st Qu.:139.0 1st Qu.:100.0 Class :character 1st Qu.:39.75

Median :272.0 Median :117.0 Mode :character 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 Length:400 Length:400

1st Qu.:12.0 Class :character Class :character

Median :14.0 Mode :character Mode :character

Mean :13.9

3rd Qu.:16.0

Max. :18.0

> describe(company)

vars n mean sd median trimmed mad min max

Sales 1 400 7.50 2.82 7.49 7.43 2.87 0 16.27

CompPrice 2 400 124.97 15.33 125.00 125.04 14.83 77 175.00

Income 3 400 68.66 27.99 69.00 68.26 35.58 21 120.00

Advertising 4 400 6.64 6.65 5.00 5.89 7.41 0 29.00

Population 5 400 264.84 147.38 272.00 265.56 191.26 10 509.00

Price 6 400 115.80 23.68 117.00 115.92 22.24 24 191.00

ShelveLoc\* 7 400 2.31 0.83 3.00 2.38 0.00 1 3.00

Age 8 400 53.32 16.20 54.50 53.48 20.02 25 80.00

Education 9 400 13.90 2.62 14.00 13.88 2.97 10 18.00

Urban\* 10 400 1.71 0.46 2.00 1.76 0.00 1 2.00

US\* 11 400 1.65 0.48 2.00 1.68 0.00 1 2.00

range skew kurtosis se

Sales 16.27 0.18 -0.11 0.14

CompPrice 98.00 -0.04 0.01 0.77

Income 99.00 0.05 -1.10 1.40

Advertising 29.00 0.63 -0.57 0.33

Population 499.00 -0.05 -1.21 7.37

Price 167.00 -0.12 0.41 1.18

ShelveLoc\* 2.00 -0.62 -1.28 0.04

Age 55.00 -0.08 -1.14 0.81

Education 8.00 0.04 -1.31 0.13

Urban\* 1.00 -0.90 -1.20 0.02

US\* 1.00 -0.60 -1.64 0.02

> pairs(company)

Error in pairs.default(company) : non-numeric argument to 'pairs'

> 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 : chr "Bad" "Good" "Medium" "Medium" ...

$ 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 : chr "Yes" "Yes" "Yes" "Yes" ...

$ US : chr "Yes" "Yes" "Yes" "Yes" ...

> attach(company)

The following objects are masked from company (pos = 8):

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

Price, Sales, ShelveLoc, Urban, US

> ############### creating categorical data on sales variable ####################

> length(Sales)

[1] 400

> sort(Sales)

[1] 0.00 0.16 0.37 0.53 0.91 1.42 1.82 2.05 2.07 2.23 2.34

[12] 2.52 2.66 2.67 2.86 2.93 2.99 3.02 3.07 3.13 3.15 3.24

[23] 3.42 3.45 3.47 3.47 3.58 3.62 3.63 3.67 3.72 3.89 3.90

[34] 3.90 3.91 3.98 4.10 4.10 4.10 4.11 4.12 4.15 4.16 4.17

[45] 4.19 4.20 4.21 4.34 4.36 4.38 4.42 4.42 4.43 4.47 4.53

[56] 4.53 4.55 4.56 4.62 4.67 4.68 4.69 4.69 4.74 4.78 4.81

[67] 4.81 4.83 4.88 4.90 4.90 4.94 4.95 4.95 4.96 4.97 4.99

[78] 5.01 5.04 5.05 5.07 5.08 5.08 5.12 5.16 5.17 5.17 5.21

[89] 5.25 5.27 5.28 5.30 5.30 5.31 5.32 5.32 5.33 5.35 5.36

[100] 5.36 5.40 5.42 5.47 5.52 5.53 5.55 5.56 5.57 5.58 5.58

[111] 5.61 5.64 5.68 5.68 5.71 5.73 5.74 5.81 5.83 5.86 5.87

[122] 5.87 5.87 5.90 5.93 5.94 5.94 5.97 5.98 5.99 6.01 6.03

[133] 6.03 6.10 6.11 6.14 6.15 6.18 6.20 6.20 6.20 6.23 6.37

[144] 6.38 6.39 6.41 6.41 6.42 6.43 6.44 6.50 6.50 6.52 6.52

[155] 6.53 6.53 6.53 6.54 6.56 6.59 6.62 6.63 6.64 6.67 6.67

[166] 6.67 6.68 6.71 6.71 6.80 6.81 6.85 6.87 6.88 6.88 6.88

[177] 6.89 6.90 6.92 6.93 6.95 6.97 6.97 6.98 7.02 7.22 7.22

[188] 7.23 7.30 7.32 7.36 7.37 7.38 7.40 7.41 7.41 7.43 7.44

[199] 7.45 7.49 7.49 7.50 7.52 7.52 7.53 7.54 7.56 7.56 7.57

[210] 7.58 7.60 7.62 7.63 7.64 7.67 7.68 7.70 7.71 7.71 7.72

[221] 7.74 7.77 7.78 7.78 7.80 7.80 7.80 7.80 7.81 7.81 7.82

[232] 7.90 7.91 7.95 7.96 7.99 8.01 8.01 8.03 8.07 8.09 8.14

[243] 8.19 8.19 8.21 8.22 8.23 8.25 8.31 8.32 8.33 8.39 8.41

[254] 8.43 8.44 8.47 8.47 8.54 8.55 8.55 8.57 8.61 8.64 8.65

[265] 8.67 8.67 8.68 8.68 8.69 8.70 8.71 8.73 8.74 8.75 8.77

[276] 8.77 8.77 8.78 8.79 8.80 8.85 8.86 8.87 8.89 8.97 8.98

[287] 9.00 9.01 9.01 9.03 9.08 9.09 9.10 9.14 9.16 9.16 9.24

[298] 9.31 9.32 9.32 9.32 9.33 9.34 9.35 9.39 9.40 9.43 9.44

[309] 9.45 9.46 9.48 9.49 9.50 9.53 9.54 9.58 9.62 9.64 9.70

[320] 9.71 9.95 10.00 10.01 10.04 10.06 10.07 10.08 10.10 10.14 10.21

[331] 10.26 10.26 10.27 10.31 10.36 10.43 10.44 10.48 10.49 10.50 10.51

[342] 10.59 10.61 10.62 10.64 10.66 10.71 10.77 10.81 10.96 10.98 11.07

[353] 11.17 11.18 11.19 11.19 11.22 11.27 11.27 11.28 11.48 11.48 11.54

[364] 11.62 11.67 11.70 11.70 11.82 11.85 11.91 11.93 11.96 11.99 12.01

[375] 12.04 12.11 12.13 12.29 12.30 12.44 12.49 12.49 12.53 12.57 12.57

[386] 12.61 12.66 12.85 12.98 13.14 13.28 13.36 13.39 13.44 13.55 13.91

[397] 14.37 14.90 15.63 16.27

> mean(Sales)

[1] 7.496325

> sort(Sales)[400/3\*2]

[1] 8.67

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

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

> View(company)

> ############## splitting data to train and test ##########################

> set.seed(100)

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

> train\_comp <- company[cutt,]

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

> ############# model building ##############################

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

> companyforest

Call:

randomForest(formula = as.factor(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: 16.37%

Confusion matrix:

high low class.error

high 68 33 0.32673267

low 13 167 0.07222222

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

Confusion Matrix and Statistics

pred\_train 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] 0.7731092

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

Confusion Matrix and Statistics

pred\_test high low

high 24 9

low 18 68

Accuracy : 0.7731

95% CI : (0.6873, 0.8448)

No Information Rate : 0.6471

P-Value [Acc > NIR] : 0.002079

Kappa : 0.4778

Mcnemar's Test P-Value : 0.123658

Sensitivity : 0.5714

Specificity : 0.8831

Pos Pred Value : 0.7273

Neg Pred Value : 0.7907

Prevalence : 0.3529

Detection Rate : 0.2017

Detection Prevalence : 0.2773

Balanced Accuracy : 0.7273

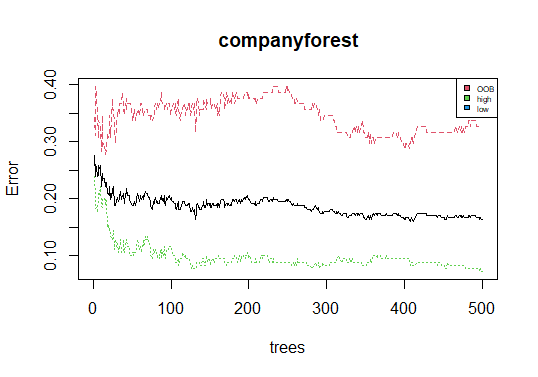
'Positive' Class : high

> # visualisation

> plot(companyforest)

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

>



# variable importance

> importance(companyforest)

MeanDecreaseGini

CompPrice 14.958170

Income 11.259949

Advertising 14.322134

Population 11.088417

Price 32.815283

ShelveLoc 18.462574

Age 15.512361

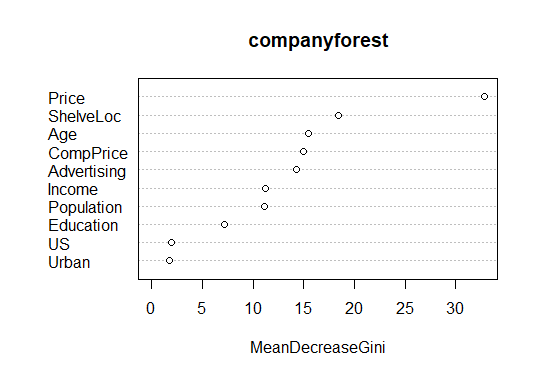
Education 7.157882

Urban 1.802789

US 2.006458

> varImpPlot(companyforest)

> # 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(as.factor(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)

+ }

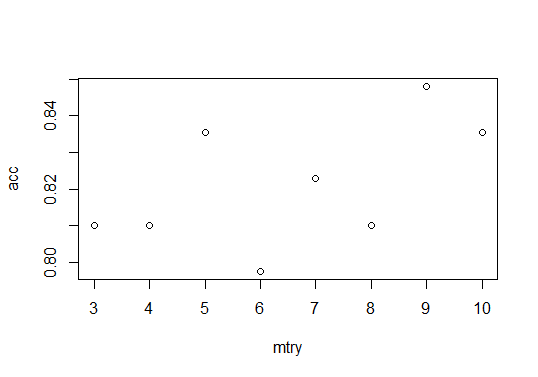
> a

[1] 0.8101266 0.8101266 0.8354430 0.7974684 0.8227848 0.8101266 0.8481013

[8] 0.8354430

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

> # we get highest accuracy for mtry = 3



**#We get highest accuracy for mtry = 3**

**# choosing mtry as 9 with less OOB error**

**> # choosing mtry as 9 with less OOB error**

**> finalmodel <- randomForest(as.factor(sales\_cat)~.,data = train\_comp,mtry=9,importance=TRUE)**

**> finalmodel**

**Call:**

**randomForest(formula = as.factor(sales\_cat) ~ ., data = train\_comp, mtry = 9, importance = TRUE)**

**Type of random forest: classification**

**Number of trees: 500**

**No. of variables tried at each split: 9**

**OOB estimate of error rate: 17.79%**

**Confusion matrix:**

**high low class.error**

**high 70 31 0.3069307**

**low 19 161 0.1055556**

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

**[1] 0.7731092**

**> ##### Company data using Smote \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**> data = read.csv(file.choose())**

**Data = data**

**>**

**> median(Data$Sales)**

**[1] 7.49**

**> Data$Sales <-factor(ifelse(Data$Sales<=7.49,'Low\_sales','High\_sales'))**

**> summary(Data)**

**Sales CompPrice Income Advertising**

**High\_sales:199 Min. : 77 Min. : 21.00 Min. : 0.000**

**Low\_sales :201 1st Qu.:115 1st Qu.: 42.75 1st Qu.: 0.000**

**Median :125 Median : 69.00 Median : 5.000**

**Mean :125 Mean : 68.66 Mean : 6.635**

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

**Max. :175 Max. :120.00 Max. :29.000**

**Population Price ShelveLoc Age**

**Min. : 10.0 Min. : 24.0 Length:400 Min. :25.00**

**1st Qu.:139.0 1st Qu.:100.0 Class :character 1st Qu.:39.75**

**Median :272.0 Median :117.0 Mode :character 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 Length:400 Length:400**

**1st Qu.:12.0 Class :character Class :character**

**Median :14.0 Mode :character Mode :character**

**Mean :13.9**

**3rd Qu.:16.0**

**Max. :18.0**

**> Data$ShelveLoc<- factor(Data$ShelveLoc)**

**> Data$Urban<-factor(Data$Urban)**

**> Data$US<-factor(Data$US)**

**> library(DMwR)**

**> ?SMOTE()**

**> x = data.frame(SMOTE(Sales~.,data=Data),perc.over = 600, perc.under = 80)**

**> View(x)**

**> table(x$Sales)**

**High\_sales Low\_sales**

**597 796**

**>**

**> rand.forest <- randomForest(Sales ~., data=x, mtry = 7,**

**+ importance = TRUE, proximity=TRUE, ntree = 1000**

**+ )**

**>**

**> rand.forest**

**Call:**

**randomForest(formula = Sales ~ ., data = x, mtry = 7, importance = TRUE, proximity = TRUE, ntree = 1000)**

**Type of random forest: classification**

**Number of trees: 1000**

**No. of variables tried at each split: 7**

**OOB estimate of error rate: 3.73%**

**Confusion matrix:**

**High\_sales Low\_sales class.error**

**High\_sales 551 46 0.077051926**

**Low\_sales 6 790 0.007537688**