**Problem Statement:**

**A cloth manufacturing company is interested to know about the segment or attributes causes high sale.**

**Approach - A decision tree can be built with target variable Sale (we will first convert it in categorical variable) & all other variable will be independent in the analysis.**

Ans🡪

install.packages("party")

install.packages("caret")

install.packages("party")

install.packages("C50")

install.packages("rpart")

install.packages("rattle")

install.packages("gmodels")

install.packages("randomForest")

install.packages("tree",repos = "http://cran.us.r-project.org")

# Load CART packages

install.packages("rmarkdown",repos = "http://cran.us.r-project.org")

library(rpart)

# install rpart package

install.packages("rpart.plot")

library(rpart.plot)

library(caret)

library(party)

library(C50)

library(rpart)

library(rattle)

library(gmodels)

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

> View(data)

> str(data)

'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" ...

> summary(data)

Sales CompPrice Income

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

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

Median : 7.490 Median :125 Median : 69.00

Mean : 7.496 Mean :125 Mean : 68.66

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

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

Advertising Population

Min. : 0.000 Min. : 10.0

1st Qu.: 0.000 1st Qu.:139.0

Median : 5.000 Median :272.0

Mean : 6.635 Mean :264.8

3rd Qu.:12.000 3rd Qu.:398.5

Max. :29.000 Max. :509.0

Price ShelveLoc

Min. : 24.0 Length:400

1st Qu.:100.0 Class :character

Median :117.0 Mode :character

Mean :115.8

3rd Qu.:131.0

Max. :191.0

Age Education

Min. :25.00 Min. :10.0

1st Qu.:39.75 1st Qu.:12.0

Median :54.50 Median :14.0

Mean :53.32 Mean :13.9

3rd Qu.:66.00 3rd Qu.:16.0

Max. :80.00 Max. :18.0

Urban US

Length:400 Length:400

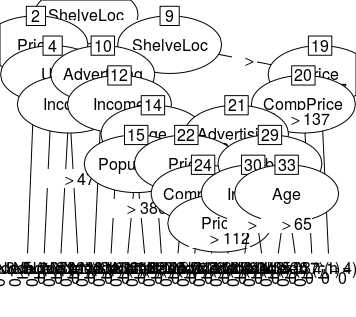
Class :character Class :character

Mode :character Mode :character

|  |
| --- |
| hist(data$Sales)    > sort(data$Sales)  [1] 0.00 0.16 0.37 0.53 0.91 1.42 1.82  [8] 2.05 2.07 2.23 2.34 2.52 2.66 2.67  [15] 2.86 2.93 2.99 3.02 3.07 3.13 3.15  [22] 3.24 3.42 3.45 3.47 3.47 3.58 3.62  [29] 3.63 3.67 3.72 3.89 3.90 3.90 3.91  [36] 3.98 4.10 4.10 4.10 4.11 4.12 4.15  [43] 4.16 4.17 4.19 4.20 4.21 4.34 4.36  [50] 4.38 4.42 4.42 4.43 4.47 4.53 4.53  [57] 4.55 4.56 4.62 4.67 4.68 4.69 4.69  [64] 4.74 4.78 4.81 4.81 4.83 4.88 4.90  [71] 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  [85] 5.16 5.17 5.17 5.21 5.25 5.27 5.28  [92] 5.30 5.30 5.31 5.32 5.32 5.33 5.35  [99] 5.36 5.36 5.40 5.42 5.47 5.52 5.53  [106] 5.55 5.56 5.57 5.58 5.58 5.61 5.64  [113] 5.68 5.68 5.71 5.73 5.74 5.81 5.83  [120] 5.86 5.87 5.87 5.87 5.90 5.93 5.94  [127] 5.94 5.97 5.98 5.99 6.01 6.03 6.03  [134] 6.10 6.11 6.14 6.15 6.18 6.20 6.20  [141] 6.20 6.23 6.37 6.38 6.39 6.41 6.41  [148] 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  [162] 6.63 6.64 6.67 6.67 6.67 6.68 6.71  [169] 6.71 6.80 6.81 6.85 6.87 6.88 6.88  [176] 6.88 6.89 6.90 6.92 6.93 6.95 6.97  [183] 6.97 6.98 7.02 7.22 7.22 7.23 7.30  [190] 7.32 7.36 7.37 7.38 7.40 7.41 7.41  [197] 7.43 7.44 7.45 7.49 7.49 7.50 7.52  [204] 7.52 7.53 7.54 7.56 7.56 7.57 7.58  [211] 7.60 7.62 7.63 7.64 7.67 7.68 7.70  [218] 7.71 7.71 7.72 7.74 7.77 7.78 7.78  [225] 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  [239] 8.03 8.07 8.09 8.14 8.19 8.19 8.21  [246] 8.22 8.23 8.25 8.31 8.32 8.33 8.39  [253] 8.41 8.43 8.44 8.47 8.47 8.54 8.55  [260] 8.55 8.57 8.61 8.64 8.65 8.67 8.67  [267] 8.68 8.68 8.69 8.70 8.71 8.73 8.74  [274] 8.75 8.77 8.77 8.77 8.78 8.79 8.80  [281] 8.85 8.86 8.87 8.89 8.97 8.98 9.00  [288] 9.01 9.01 9.03 9.08 9.09 9.10 9.14  [295] 9.16 9.16 9.24 9.31 9.32 9.32 9.32  [302] 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  [316] 9.58 9.62 9.64 9.70 9.71 9.95 10.00  [323] 10.01 10.04 10.06 10.07 10.08 10.10 10.14  [330] 10.21 10.26 10.26 10.27 10.31 10.36 10.43  [337] 10.44 10.48 10.49 10.50 10.51 10.59 10.61  [344] 10.62 10.64 10.66 10.71 10.77 10.81 10.96  [351] 10.98 11.07 11.17 11.18 11.19 11.19 11.22  [358] 11.27 11.27 11.28 11.48 11.48 11.54 11.62  [365] 11.67 11.70 11.70 11.82 11.85 11.91 11.93  [372] 11.96 11.99 12.01 12.04 12.11 12.13 12.29  [379] 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  [393] 13.39 13.44 13.55 13.91 14.37 14.90 15.63  [400] 16.27  > length(data$Sales)  [1] 400  > mean(data$Sales)  [1] 7.496325  > sort(data$Sales)[400/3\*2] #sales may be high,medium,low  [1] 8.67  > #converting sales to categorical type |
|  |
| |  | | --- | | > sales\_cat<- ifelse(data$Sales>7.49,"high","low")  > df <- data.frame(sales\_cat,data[,-1])  > View(df)  > #splitting data to train and test data  > set.seed(100)  > CD = data.frame(data,sales\_cat )  > # View(CD)  > CD\_train <- CD[1:200,]  > # View(CD\_train)  > CD\_test <- CD[201:400,]  >  > #building model  > #Using Party Function  >  > #op\_tree using party package  >  > op\_tree = ctree(as.factor(sales\_cat) ~ CompPrice + Income + Advertising + Population + Price  + + Age + Education, data = CD\_train)  > plot(op\_tree) | |
| > pred\_tree <- as.data.frame(predict(op\_tree,newdata=CD\_test))  > pred\_tree["final"] <- NULL  > pred1<- predict(op\_tree,newdata=CD\_test)  > mean(pred1==CD$sales\_cat) # Accuracy = 57.25 %  [1] 0.5725   |  | | --- | | > CrossTable(CD\_test$sales\_cat,pred1)    Cell Contents  |-------------------------|  | N |  | Chi-square contribution |  | N / Row Total |  | N / Col Total |  | N / Table Total |  |-------------------------|    Total Observations in Table: 200    | pred1  CD\_test$sales\_cat | high | low | Row Total |  ------------------|-----------|-----------|-----------|  high | 71 | 28 | 99 |  | 7.402 | 8.018 | |  | 0.717 | 0.283 | 0.495 |  | 0.683 | 0.292 | |  | 0.355 | 0.140 | |  ------------------|-----------|-----------|-----------|  low | 33 | 68 | 101 |  | 7.255 | 7.860 | |  | 0.327 | 0.673 | 0.505 |  | 0.317 | 0.708 | |  | 0.165 | 0.340 | |  ------------------|-----------|-----------|-----------|  Column Total | 104 | 96 | 200 |  | 0.520 | 0.480 | |  ------------------|-----------|-----------|-----------| | |  | | |  | | --- | | > | | |
|  |
| |  | | --- | | > | |

> #model2 using c50 package

> model2 <- C5.0(train[,-1],as.factor(train$sales\_cat),trails=100)



> pred2 <- predict.C5.0(model2,test)

> table(pred2,test$sales\_cat)

pred2 0 1

0 60 9

1 17 14

> mean(pred2==test$sales\_cat) #74.0% acc

[1] 0.74

> plot(model2)

> C5imp(model2)

Overall

ShelveLoc 100.00

Price 76.00

Advertising 57.67

CompPrice 37.33

Income 25.67

Urban 15.33

Age 11.33

US 9.00

Population 3.67

Education 0.00

> CrossTable(test$sales\_cat,pred2)

Cell Contents

|-------------------------|

| N |

| Chi-square contribution |

| N / Row Total |

| N / Col Total |

| N / Table Total |

|-------------------------|

Total Observations in Table: 100

| pred2

test$sales\_cat | 0 | 1 | Row Total |

---------------|-----------|-----------|-----------|

0 | 60 | 17 | 77 |

| 0.888 | 1.977 | |

| 0.779 | 0.221 | 0.770 |

| 0.870 | 0.548 | |

| 0.600 | 0.170 | |

---------------|-----------|-----------|-----------|

1 | 9 | 14 | 23 |

| 2.974 | 6.619 | |

| 0.391 | 0.609 | 0.230 |

| 0.130 | 0.452 | |

| 0.090 | 0.140 | |

---------------|-----------|-----------|-----------|

Column Total | 69 | 31 | 100 |

| 0.690 | 0.310 | |

---------------|-----------|-----------|-----------|

|  |
| --- |
| #model3 bagging method  >  > acc <- c()  > for(i in 1:100){  + print(i)  + splitting <- createDataPartition(sales\_cat,p=0.85,list = F)  + training <- df[splitting,]  + testing <- df[-splitting,]  +  + modelfit <- C5.0(training[,-1],as.factor(training$sales\_cat))  + predictfit <- predict(modelfit,testing)  + a <- table(predictfit,testing$sales\_cat)  + acc <- c(acc,sum(diag(a))/sum(a))  + }  [1] 1  [1] 2  [1] 3  [1] 4  [1] 5  [1] 6  [1] 7  [1] 8  [1] 9  [1] 10  [1] 11  [1] 12  [1] 13  [1] 14  [1] 15  [1] 16  [1] 17  [1] 18  [1] 19  [1] 20  [1] 21  [1] 22  [1] 23  [1] 24  [1] 25  [1] 26  [1] 27  [1] 28  [1] 29  [1] 30  [1] 31  [1] 32  [1] 33  [1] 34  [1] 35  [1] 36  [1] 37  [1] 38  [1] 39  [1] 40  [1] 41  [1] 42  [1] 43  [1] 44  [1] 45  [1] 46  [1] 47  [1] 48  [1] 49  [1] 50  [1] 51  [1] 52  [1] 53  [1] 54  [1] 55  [1] 56  [1] 57  [1] 58  [1] 59  [1] 60  [1] 61  [1] 62  [1] 63  [1] 64  [1] 65  [1] 66  [1] 67  [1] 68  [1] 69  [1] 70  [1] 71  [1] 72  [1] 73  [1] 74  [1] 75  [1] 76  [1] 77  [1] 78  [1] 79  [1] 80  [1] 81  [1] 82  [1] 83  [1] 84  [1] 85  [1] 86  [1] 87  [1] 88  [1] 89  [1] 90  [1] 91  [1] 92  [1] 93  [1] 94  [1] 95  [1] 96  [1] 97  [1] 98  [1] 99  [1] 100  > acc  [1] 0.7796610 0.7966102 0.7457627 0.6779661  [5] 0.7457627 0.6779661 0.6949153 0.8305085  [9] 0.7627119 0.6949153 0.7457627 0.7457627  [13] 0.8135593 0.8644068 0.6949153 0.8135593  [17] 0.7288136 0.6610169 0.6949153 0.7796610  [21] 0.7288136 0.7457627 0.7457627 0.8644068  [25] 0.7627119 0.7457627 0.8135593 0.7118644  [29] 0.7118644 0.6949153 0.6949153 0.7457627  [33] 0.7457627 0.7966102 0.7457627 0.8813559  [37] 0.7796610 0.7627119 0.8135593 0.8305085  [41] 0.7457627 0.7796610 0.7457627 0.8474576  [45] 0.7627119 0.7796610 0.7457627 0.7627119  [49] 0.8474576 0.7627119 0.7966102 0.7796610  [53] 0.8474576 0.7796610 0.6949153 0.7288136  [57] 0.7796610 0.8135593 0.7288136 0.7966102  [61] 0.7966102 0.6610169 0.8474576 0.7796610  [65] 0.8305085 0.7457627 0.8474576 0.7796610  [69] 0.7457627 0.6779661 0.6610169 0.6610169  [73] 0.7118644 0.7457627 0.7796610 0.7796610  [77] 0.6949153 0.8305085 0.8305085 0.7288136  [81] 0.6610169 0.7796610 0.7627119 0.8305085  [85] 0.7627119 0.7966102 0.6101695 0.7457627  [89] 0.7796610 0.8305085 0.8305085 0.6610169  [93] 0.8305085 0.6779661 0.6271186 0.8813559  [97] 0.7796610 0.8813559 0.7796610 0.8305085  > summary(acc)  Min. 1st Qu. Median Mean 3rd Qu. Max.  0.6102 0.7288 0.7627 0.7627 0.8008 0.8814 |
|  |
| |  | | --- | | > | |
| pred4 <- predict(model4,test)  > table(pred4,test$sales\_cat)    pred4 0 1  0.0408163265306122 22 1  0.0447761194029851 17 0  0.0769230769230769 4 1  0.111111111111111 7 1  0.142857142857143 4 1  0.333333333333333 3 0  0.5 4 0  0.538461538461538 0 2  0.615384615384615 2 2  0.642857142857143 4 4  0.825 7 4  0.975609756097561 3 7  > CrossTable(test$sales\_cat,pred4)    Cell Contents  |-------------------------|  | N |  | Chi-square contribution |  | N / Row Total |  | N / Col Total |  | N / Table Total |  |-------------------------|    Total Observations in Table: 100    | pred4  test$sales\_cat | 0.0408163265306122 | 0.0447761194029851 | 0.0769230769230769 | 0.111111111111111 | 0.142857142857143 | 0.333333333333333 | 0.5 | 0.538461538461538 | 0.615384615384615 | 0.642857142857143 | 0.825 | 0.975609756097561 | Row Total |  ---------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|  0 | 22 | 17 | 4 | 7 | 4 | 3 | 4 | 0 | 2 | 4 | 7 | 3 | 77 |  | 1.039 | 1.168 | 0.006 | 0.115 | 0.006 | 0.206 | 0.275 | 1.540 | 0.379 | 0.757 | 0.255 | 2.869 | |  | 0.286 | 0.221 | 0.052 | 0.091 | 0.052 | 0.039 | 0.052 | 0.000 | 0.026 | 0.052 | 0.091 | 0.039 | 0.770 |  | 0.957 | 1.000 | 0.800 | 0.875 | 0.800 | 1.000 | 1.000 | 0.000 | 0.500 | 0.500 | 0.636 | 0.300 | |  | 0.220 | 0.170 | 0.040 | 0.070 | 0.040 | 0.030 | 0.040 | 0.000 | 0.020 | 0.040 | 0.070 | 0.030 | |  ---------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|  1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 2 | 2 | 4 | 4 | 7 | 23 |  | 3.479 | 3.910 | 0.020 | 0.383 | 0.020 | 0.690 | 0.920 | 5.156 | 1.268 | 2.536 | 0.854 | 9.604 | |  | 0.043 | 0.000 | 0.043 | 0.043 | 0.043 | 0.000 | 0.000 | 0.087 | 0.087 | 0.174 | 0.174 | 0.304 | 0.230 |  | 0.043 | 0.000 | 0.200 | 0.125 | 0.200 | 0.000 | 0.000 | 1.000 | 0.500 | 0.500 | 0.364 | 0.700 | |  | 0.010 | 0.000 | 0.010 | 0.010 | 0.010 | 0.000 | 0.000 | 0.020 | 0.020 | 0.040 | 0.040 | 0.070 | |  ---------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|  Column Total | 23 | 17 | 5 | 8 | 5 | 3 | 4 | 2 | 4 | 8 | 11 | 10 | 100 |  | 0.230 | 0.170 | 0.050 | 0.080 | 0.050 | 0.030 | 0.040 | 0.020 | 0.040 | 0.080 | 0.110 | 0.100 | |  ---------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------| |
|  |
| |  | | --- | | > control\_stacking <- trainControl(method="repeatedcv", number=2, repeats=2, savePredictions=TRUE, classProbs=TRUE)  >  > algorithms\_to\_use <- c('rpart', 'knn')  > stacked\_models <- caretList(sales\_cat~.,data = train, trControl=control\_stacking, methodList=algorithms\_to\_use)  >  > stacking\_results <- resamples(stacked\_models)  >  > summary(stacking\_results)  Call:  summary.resamples(object = stacking\_results)  Models: rpart, knn  Number of resamples: 4  MAE  Min. 1st Qu. Median Mean 3rd Qu. Max. NA's  rpart 0.3163902 0.3257621 0.3496619 0.3480210 0.3719208 0.3763698 0  knn 0.4377778 0.4500000 0.4562963 0.4568519 0.4631481 0.4770370 0  RMSE  Min. 1st Qu. Median Mean 3rd Qu. Max. NA's  rpart 0.4034438 0.4568302 0.4766072 0.4614365 0.4812135 0.4890878 0  knn 0.4735819 0.4786295 0.4906288 0.4962694 0.5082687 0.5302379 0  Rsquared  Min. 1st Qu. Median Mean 3rd Qu.  rpart 0.0978288836 0.127323315 0.1539015 0.18368749 0.21026566  knn 0.0007791167 0.008118074 0.0276507 0.03302385 0.05255648  Max. NA's  rpart 0.32911812 0  knn 0.07601488 0  > xyplot(resamples(stacked\_models)) | |

# Pruning

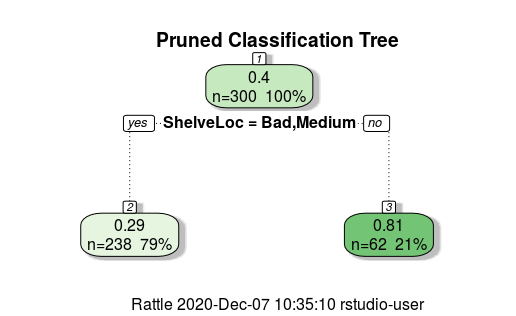
library(DMwR)

install.packages("fancyrpartplot")

install.packages("tree")

zp<-prune(model4,cp=0.1)

fancyRpartPlot(zp, uniform=TRUE, main="Pruned Classification Tree")



printcp(zp)

Regression tree:

rpart(formula = sales\_cat ~ ., data = train)

Variables actually used in tree construction:

[1] ShelveLoc

Root node error: 72/300 = 0.24

n= 300

CP nsplit rel error xerror xstd

1 0.17932 0 1.00000 1.00261 0.023667

2 0.10000 1 0.82068 0.82689 0.048876