Classification-Tree-With-State.R

obite

2021-06-29

library(rpart)  
library(rpart.plot)

## Warning: package 'rpart.plot' was built under R version 4.0.5

library(caret)

## Warning: package 'caret' was built under R version 4.0.5

## Loading required package: lattice

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.0.5

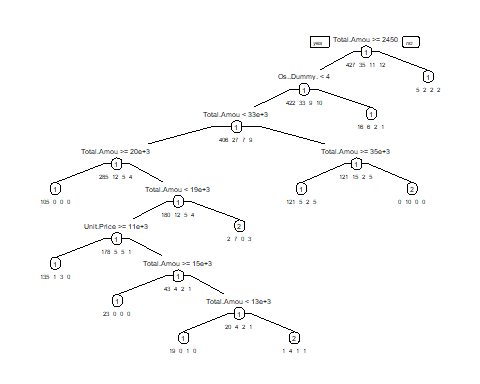
df <- read.csv("Dummy Data With State.csv")  
selected = c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21)  
  
# Splitting the dataset into the Training set and Test set  
ind = sample(2, nrow(df), replace=TRUE, prob=c(0.6,0.4))  
training = df[ind==1,selected]  
testing = df[ind==2,selected]  
  
dim(training)

## [1] 485 21

dim(testing)

## [1] 316 21

# classification tree  
default.ct <- rpart(Order.Status..Dummy. ~ ., data = training, method = "class")  
  
# plot tree  
prp(default.ct, type = 1, extra = 1, under = TRUE, split.font = 1, varlen = -10)



#CODE FOR TESTING ACCURACY   
dpred.train <- predict(default.ct,training,type = "class")  
dpred.test <- predict(default.ct,testing,type = "class")  
  
str(testing$Order.Status..Dummy.)

## int [1:316] 1 1 1 1 1 1 1 1 1 1 ...

str(dpred.test)

## Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1 1 1 1 1 ...  
## - attr(\*, "names")= chr [1:316] "1" "7" "8" "10" ...

#GENERATE CONFUSION MATRIX  
  
#default tree: training  
confusionMatrix(dpred.train, as.factor(training$Order.Status..Dummy.))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 1 2 3 4  
## 1 424 14 10 8  
## 2 3 21 1 4  
## 3 0 0 0 0  
## 4 0 0 0 0  
##   
## Overall Statistics  
##   
## Accuracy : 0.9175   
## 95% CI : (0.8894, 0.9404)  
## No Information Rate : 0.8804   
## P-Value [Acc > NIR] : 0.005337   
##   
## Kappa : 0.5088   
##   
## Mcnemar's Test P-Value : NA   
##   
## Statistics by Class:  
##   
## Class: 1 Class: 2 Class: 3 Class: 4  
## Sensitivity 0.9930 0.60000 0.00000 0.00000  
## Specificity 0.4483 0.98222 1.00000 1.00000  
## Pos Pred Value 0.9298 0.72414 NaN NaN  
## Neg Pred Value 0.8966 0.96930 0.97732 0.97526  
## Prevalence 0.8804 0.07216 0.02268 0.02474  
## Detection Rate 0.8742 0.04330 0.00000 0.00000  
## Detection Prevalence 0.9402 0.05979 0.00000 0.00000  
## Balanced Accuracy 0.7206 0.79111 0.50000 0.50000

#default tree: testing  
confusionMatrix(factor(dpred.test, levels = 1:4), factor(testing$Order.Status..Dummy., levels = 1:4))

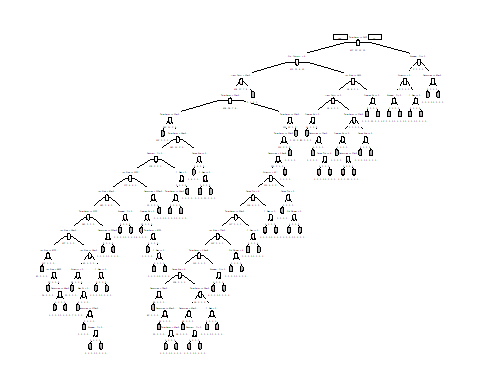
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 1 2 3 4  
## 1 276 10 8 3  
## 2 4 12 1 2  
## 3 0 0 0 0  
## 4 0 0 0 0  
##   
## Overall Statistics  
##   
## Accuracy : 0.9114   
## 95% CI : (0.8745, 0.9403)  
## No Information Rate : 0.8861   
## P-Value [Acc > NIR] : 0.08879   
##   
## Kappa : 0.4564   
##   
## Mcnemar's Test P-Value : NA   
##   
## Statistics by Class:  
##   
## Class: 1 Class: 2 Class: 3 Class: 4  
## Sensitivity 0.9857 0.54545 0.00000 0.00000  
## Specificity 0.4167 0.97619 1.00000 1.00000  
## Pos Pred Value 0.9293 0.63158 NaN NaN  
## Neg Pred Value 0.7895 0.96633 0.97152 0.98418  
## Prevalence 0.8861 0.06962 0.02848 0.01582  
## Detection Rate 0.8734 0.03797 0.00000 0.00000  
## Detection Prevalence 0.9399 0.06013 0.00000 0.00000  
## Balanced Accuracy 0.7012 0.76082 0.50000 0.50000

#CODE FOR CREATING A DEEPER CLASSIFICATION TREE  
  
deeper.ct <- rpart(Order.Status..Dummy. ~ ., data = training, method = "class", cp = 0, minsplit = 1)  
  
# count number of leaves  
length(deeper.ct$frame$var[deeper.ct$frame$var == "<leaf>"])

## [1] 68

# plot tree  
prp(deeper.ct, type = 1, extra = 1, under = TRUE, split.font = 1, varlen = -10,  
 box.col=ifelse(deeper.ct$frame$var == "<leaf>", 'gray', 'white'))

## Warning: labs do not fit even at cex 0.15, there may be some overplotting



#CODE FOR TESTING ACCURACY   
# classify records in the testing data.  
#set argument type = "class" in predict() to generate predicted class membership.  
deeperpred.train <- predict(deeper.ct,training,type = "class")  
deeperpred.test <- predict(deeper.ct,testing,type = "class")  
  
#GENERATE CONFUSION MATRIX  
  
#deeper tree: training  
confusionMatrix(deeperpred.train, as.factor(training$Order.Status..Dummy.))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 1 2 3 4  
## 1 427 0 0 0  
## 2 0 35 0 0  
## 3 0 0 11 0  
## 4 0 0 0 12  
##   
## Overall Statistics  
##   
## Accuracy : 1   
## 95% CI : (0.9924, 1)  
## No Information Rate : 0.8804   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 1   
##   
## Mcnemar's Test P-Value : NA   
##   
## Statistics by Class:  
##   
## Class: 1 Class: 2 Class: 3 Class: 4  
## Sensitivity 1.0000 1.00000 1.00000 1.00000  
## Specificity 1.0000 1.00000 1.00000 1.00000  
## Pos Pred Value 1.0000 1.00000 1.00000 1.00000  
## Neg Pred Value 1.0000 1.00000 1.00000 1.00000  
## Prevalence 0.8804 0.07216 0.02268 0.02474  
## Detection Rate 0.8804 0.07216 0.02268 0.02474  
## Detection Prevalence 0.8804 0.07216 0.02268 0.02474  
## Balanced Accuracy 1.0000 1.00000 1.00000 1.00000

#deeper tree: testing  
confusionMatrix(factor(deeperpred.test,levels = 1:4), factor(testing$Order.Status..Dummy., levels = 1:4))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 1 2 3 4  
## 1 255 9 4 2  
## 2 14 13 1 0  
## 3 6 0 0 1  
## 4 5 0 4 2  
##   
## Overall Statistics  
##   
## Accuracy : 0.8544   
## 95% CI : (0.8106, 0.8914)  
## No Information Rate : 0.8861   
## P-Value [Acc > NIR] : 0.9649   
##   
## Kappa : 0.382   
##   
## Mcnemar's Test P-Value : NA   
##   
## Statistics by Class:  
##   
## Class: 1 Class: 2 Class: 3 Class: 4  
## Sensitivity 0.9107 0.59091 0.00000 0.400000  
## Specificity 0.5833 0.94898 0.97720 0.971061  
## Pos Pred Value 0.9444 0.46429 0.00000 0.181818  
## Neg Pred Value 0.4565 0.96875 0.97087 0.990164  
## Prevalence 0.8861 0.06962 0.02848 0.015823  
## Detection Rate 0.8070 0.04114 0.00000 0.006329  
## Detection Prevalence 0.8544 0.08861 0.02215 0.034810  
## Balanced Accuracy 0.7470 0.76994 0.48860 0.685531