ams580\_proj1\_cart

EMVP

2023-03-13

#1.For the entire dataset, please perform the data cleaning as instructed before; namely, exclude the variables Name, Ticket, and Cabin and delete missing values in the variable Age. Please report how many passengers are left after this step. Then please use the random seed 123 to divide the cleaned data into 80% training and 20% testing.  
  
#~Install Packages  
if (!requireNamespace("caTools")) install.packages('caTools')

## Loading required namespace: caTools

if (!requireNamespace("tidyverse")) install.packages('tidyverse')

## Loading required namespace: tidyverse

if (!requireNamespace("caret")) install.packages('caret')

## Loading required namespace: caret

if (!requireNamespace("rpart")) install.packages('rpart')  
if (!requireNamespace("rattle")) install.packages('rattle')

## Loading required namespace: rattle

library(caTools)  
library(tidyverse)

## ── Attaching packages  
## ───────────────────────────────────────  
## tidyverse 1.3.2 ──

## ✔ ggplot2 3.4.0 ✔ purrr 1.0.1   
## ✔ tibble 3.1.8 ✔ dplyr 1.0.10  
## ✔ tidyr 1.3.0 ✔ stringr 1.5.0   
## ✔ readr 2.1.4 ✔ forcats 1.0.0   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library(caret)

## Loading required package: lattice  
##   
## Attaching package: 'caret'  
##   
## The following object is masked from 'package:purrr':  
##   
## lift

library(rpart)  
library(rattle)

## Loading required package: bitops  
## Rattle: A free graphical interface for data science with R.  
## Version 5.5.1 Copyright (c) 2006-2021 Togaware Pty Ltd.  
## Type 'rattle()' to shake, rattle, and roll your data.

#~Read File  
data <- read.csv("~/Desktop/stuff/semesters/spring 2023/ams 580/examples/Titanic.csv")  
  
#~Remove rows with empty cells for age   
data <- subset(data, is.na(Age) == FALSE)  
  
#~Remove `Name`, `Ticket`, and `Cabin` columns  
data <- subset(data, select = -c(Name,Ticket,Cabin))  
  
#~Make the Sex column binary  
data$Sex <- ifelse(data$Sex=="male",1,0)  
  
#~Present cleaned data  
cat('There are', nrow(data), 'rows left.')

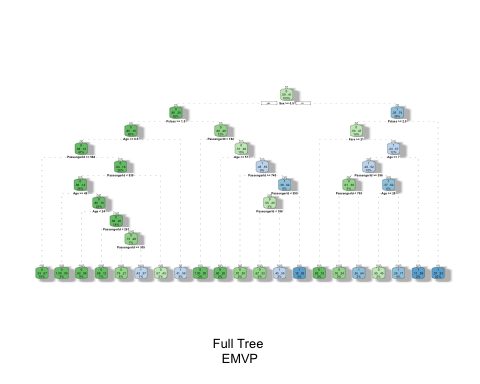
## There are 714 rows left.

data$Survived <- as.factor(data$Survived)  
str(data)

## 'data.frame': 714 obs. of 9 variables:  
## $ PassengerId: int 1 2 3 4 5 7 8 9 10 11 ...  
## $ Survived : Factor w/ 2 levels "0","1": 1 2 2 2 1 1 1 2 2 2 ...  
## $ Pclass : int 3 1 3 1 3 1 3 3 2 3 ...  
## $ Sex : num 1 0 0 0 1 1 1 0 0 0 ...  
## $ Age : num 22 38 26 35 35 54 2 27 14 4 ...  
## $ SibSp : int 1 1 0 1 0 0 3 0 1 1 ...  
## $ Parch : int 0 0 0 0 0 0 1 2 0 1 ...  
## $ Fare : num 7.25 71.28 7.92 53.1 8.05 ...  
## $ Embarked : chr "S" "C" "S" "S" ...

#~Split data into test/train  
set.seed(123)  
training.samples <- data$Survived %>%  
 createDataPartition(p = 0.8, list = FALSE)  
training <- data[training.samples, ]  
testing <- data[-training.samples, ]

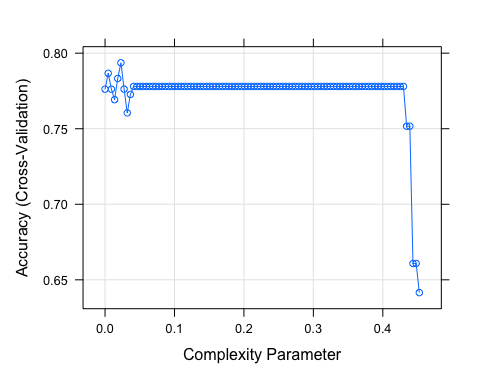
#2.Please first build a fully grown tree using the training data, and draw the tree plot using rattle. Next please use this tree to predict the survival of passengers in the testing data. Please compute the Confusion matrix and report the sensitivity, specificity and the overall accuracy for the testing data.  
  
#~Build/draw full tree  
model <- rpart(Survived ~., data = training, control = rpart.control(cp=0))  
par(xpd = NA)  
fancyRpartPlot(model, sub = "Full Tree \n EMVP")



#~Predicting in testing, conf matrix  
pred1 <- predict(model,newdata = testing, type ='class')  
pred1 <- ifelse(pred1 == 1, 1, 0)  
confusionMatrix(factor(pred1), factor(testing$Survived), positive = '1')

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 71 11  
## 1 13 47  
##   
## Accuracy : 0.831   
## 95% CI : (0.759, 0.8886)  
## No Information Rate : 0.5915   
## P-Value [Acc > NIR] : 7.518e-10   
##   
## Kappa : 0.6521   
##   
## Mcnemar's Test P-Value : 0.8383   
##   
## Sensitivity : 0.8103   
## Specificity : 0.8452   
## Pos Pred Value : 0.7833   
## Neg Pred Value : 0.8659   
## Prevalence : 0.4085   
## Detection Rate : 0.3310   
## Detection Prevalence : 0.4225   
## Balanced Accuracy : 0.8278   
##   
## 'Positive' Class : 1   
##

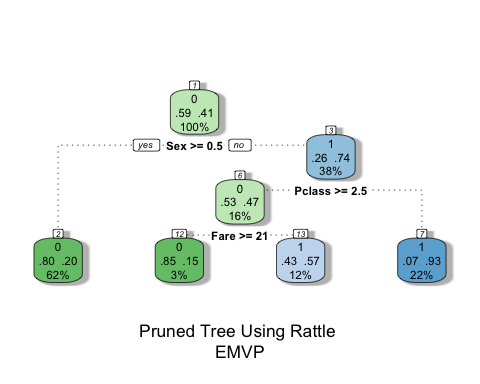
#3.To make the tree more robust, we will prune the fully grown tree using the training data with 10-fold cross-validation. Please (1) show the complexity plot, (2) report the best CP value, and (3) draw the pruned tree using rattle.   
  
#(1) show the complexity plot  
set.seed(123)  
model2 <- train(  
 Survived ~., data = training, method = "rpart",  
 trControl = trainControl("cv", number = 10),  
 tuneLength = 100)  
plot(model2)



#(2) report the best CP value  
model2$bestTune

## cp  
## 6 0.02285789

#(3) draw the pruned tree using rattle  
fancyRpartPlot(model2$finalModel, sub = "Pruned Tree Using Rattle \n EMVP")



#4.Please use this optimal pruned tree to predict the survival of passengers in the testing data. Please compute the Confusion matrix and report the sensitivity, specificity and the overall accuracy for the testing data.  
  
pred2 <- predict(model2, newdata = testing)  
pred2 <- ifelse(pred2 == 1, 1, 0)  
confusionMatrix(factor(pred2), factor(testing$Survived), positive = '1')

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 79 22  
## 1 5 36  
##   
## Accuracy : 0.8099   
## 95% CI : (0.7355, 0.8708)  
## No Information Rate : 0.5915   
## P-Value [Acc > NIR] : 2.405e-08   
##   
## Kappa : 0.5878   
##   
## Mcnemar's Test P-Value : 0.002076   
##   
## Sensitivity : 0.6207   
## Specificity : 0.9405   
## Pos Pred Value : 0.8780   
## Neg Pred Value : 0.7822   
## Prevalence : 0.4085   
## Detection Rate : 0.2535   
## Detection Prevalence : 0.2887   
## Balanced Accuracy : 0.7806   
##   
## 'Positive' Class : 1   
##

#5.Please output the test data along with the prediction results from both the Full Tree, as well as the Optimal Pruned Tree. Please generate a 2x2 table to show the agreement between these two prediction methods.   
  
#~Test Data and Pred Results  
str(testing)

## 'data.frame': 142 obs. of 9 variables:  
## $ PassengerId: int 1 4 7 14 19 24 35 38 39 53 ...  
## $ Survived : Factor w/ 2 levels "0","1": 1 2 1 1 1 2 1 1 1 2 ...  
## $ Pclass : int 3 1 1 3 3 1 1 3 3 1 ...  
## $ Sex : num 1 0 1 1 0 1 1 1 0 0 ...  
## $ Age : num 22 35 54 39 31 28 28 21 18 49 ...  
## $ SibSp : int 1 1 0 1 1 0 1 0 2 1 ...  
## $ Parch : int 0 0 0 5 0 0 0 0 0 0 ...  
## $ Fare : num 7.25 53.1 51.86 31.27 18 ...  
## $ Embarked : chr "S" "S" "S" "S" ...

confusionMatrix(factor(pred1), factor(testing$Survived), positive = '1')

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 71 11  
## 1 13 47  
##   
## Accuracy : 0.831   
## 95% CI : (0.759, 0.8886)  
## No Information Rate : 0.5915   
## P-Value [Acc > NIR] : 7.518e-10   
##   
## Kappa : 0.6521   
##   
## Mcnemar's Test P-Value : 0.8383   
##   
## Sensitivity : 0.8103   
## Specificity : 0.8452   
## Pos Pred Value : 0.7833   
## Neg Pred Value : 0.8659   
## Prevalence : 0.4085   
## Detection Rate : 0.3310   
## Detection Prevalence : 0.4225   
## Balanced Accuracy : 0.8278   
##   
## 'Positive' Class : 1   
##

confusionMatrix(factor(pred2), factor(testing$Survived), positive = '1')

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 79 22  
## 1 5 36  
##   
## Accuracy : 0.8099   
## 95% CI : (0.7355, 0.8708)  
## No Information Rate : 0.5915   
## P-Value [Acc > NIR] : 2.405e-08   
##   
## Kappa : 0.5878   
##   
## Mcnemar's Test P-Value : 0.002076   
##   
## Sensitivity : 0.6207   
## Specificity : 0.9405   
## Pos Pred Value : 0.8780   
## Neg Pred Value : 0.7822   
## Prevalence : 0.4085   
## Detection Rate : 0.2535   
## Detection Prevalence : 0.2887   
## Balanced Accuracy : 0.7806   
##   
## 'Positive' Class : 1   
##

#~2x2 table  
m1 <- c(71, 11, 13, 47)  
m2 <- c(79, 22, 5, 36)  
pred1Matrix <- matrix(m1, nrow=2)  
pred2Matrix <- matrix(m2, nrow=2)  
pred1Matrix

## [,1] [,2]  
## [1,] 71 13  
## [2,] 11 47

pred2Matrix

## [,1] [,2]  
## [1,] 79 5  
## [2,] 22 36