Carseats-DecisionTree-Example.R

duter

2021-03-06

## TASK 1 ##  
  
# load library ISLR and view the data frame Carseats  
library(ISLR)  
View(Carseats)  
  
# use the help function to learn about Carseats  
help(Carseats)

## starting httpd help server ... done

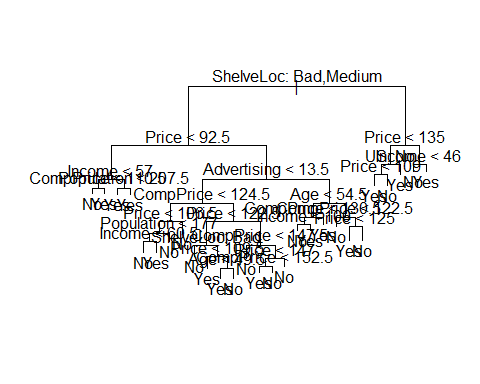
# load library tree, you may have to install the package tree first  
library(tree)

## Warning: package 'tree' was built under R version 4.0.4

## TASK 2 ##  
  
# add new column High to Carseats, which will indicate yes if Sales > 8 and no otherwise  
High = ifelse(Carseats$Sales > 8, "Yes", "No")  
Carseats1 = data.frame(Carseats, High, stringsAsFactors = TRUE)  
  
## TASK 3 ##  
  
# create a decision tree to predict high or low sales. The response is High, the predictors are everything except Sales  
tree.carseats = tree(High ~ . - Sales, Carseats1)  
summary(tree.carseats)

##   
## Classification tree:  
## tree(formula = High ~ . - Sales, data = Carseats1)  
## Variables actually used in tree construction:  
## [1] "ShelveLoc" "Price" "Income" "CompPrice" "Population"   
## [6] "Advertising" "Age" "US"   
## Number of terminal nodes: 27   
## Residual mean deviance: 0.4575 = 170.7 / 373   
## Misclassification error rate: 0.09 = 36 / 400

# you can also plot the tree  
plot(tree.carseats)  
  
# to see what the branches are, you can do the following  
text(tree.carseats, pretty = 0)



## TASK 4 ##  
  
# set random seed  
set.seed(2)  
  
# nrow(Carseats1) will give us the number of rows in Carseats1. we are randomly picking  
# half of those rows to be our training data. We are storing the row numbers in our training data  
# in a variable called train.  
train = sample(1:nrow(Carseats1), nrow(Carseats1) / 2)  
  
# the remaining rows are set aside as test data  
Carseats.test = Carseats1[-train, ]  
  
# we store the actual High vs. Low observations for test data in a vector called High.test  
High.test = High[-train]  
  
  
##TASK 5##  
  
# create a decision tree to predict high or low sales, using only training data.  
# The response is High, the predictors are everything except Sales  
tree.carseats = tree(High ~ . - Sales, Carseats1, subset = train)  
  
## TASK 6 ##  
  
# using the tree created in TASK 6, we make High vs. Low predictions for the test data  
# the predictions are stored in a vector called tree.pred  
tree.pred = predict(tree.carseats, Carseats.test, type = "class")  
  
# display a table that shows predictions for test data versus actuals for test data  
table(tree.pred, High.test)

## High.test  
## tree.pred No Yes  
## No 104 33  
## Yes 13 50

# calculate prediction accuracy  
mean(tree.pred == High.test)

## [1] 0.77

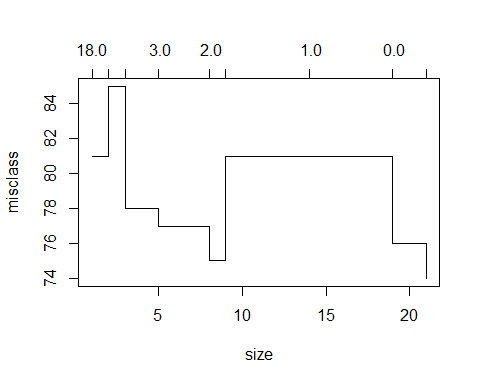
## TASK 7 ##  
  
# set random seed  
set.seed(3)  
  
# use cross-validation and pruning to obtain smaller trees and their prediction accuracy  
# store the results in a variable called cv.carseats  
cv.carseats = cv.tree(tree.carseats, FUN = prune.misclass)  
  
# check the names of columns in cv.carseats -- two of them are the important ones for us:  
# size is the size of a tree, specifically, the number of nodes in the tree  
# dev, in this case, is the fraction of seats that were misclassified (high-sales seats classified as low or vice versa)  
names(cv.carseats)

## [1] "size" "dev" "k" "method"

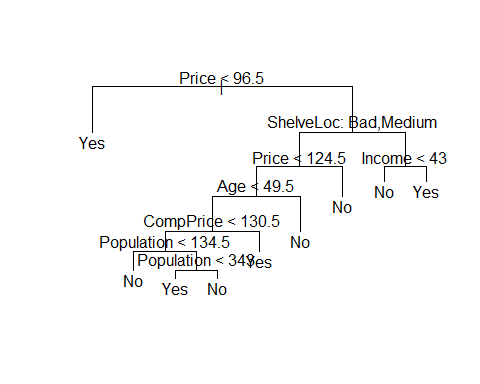
# display the information in cv.carseats  
# osberve that dev is minimum when size is 9, so the best tree has a size of 9  
cv.carseats

## $size  
## [1] 21 19 14 9 8 5 3 2 1  
##   
## $dev  
## [1] 74 76 81 81 75 77 78 85 81  
##   
## $k  
## [1] -Inf 0.0 1.0 1.4 2.0 3.0 4.0 9.0 18.0  
##   
## $method  
## [1] "misclass"  
##   
## attr(,"class")  
## [1] "prune" "tree.sequence"

plot(cv.carseats)



## TASK 8 ##  
  
# use a function called prune.misclass() to obtain the best tree, which we know has a size of 9  
# store the resulting tree in a variable called prune.carseats  
prune.carseats = prune.misclass(tree.carseats, best = 9)  
  
# plot the best tree  
plot(prune.carseats)  
  
# display branch names on the tree  
text(prune.carseats, pretty = 0)



## TASk 9 ##  
  
# using the best tree obtained in TASK 8, we make High vs. Low predictions for the test data  
# the predictions are stored in a vector called tree.pred  
tree.pred = predict(prune.carseats, Carseats.test, type = "class")  
  
# display a table that shows predictions for test data versus actuals for test data  
table(tree.pred, High.test)

## High.test  
## tree.pred No Yes  
## No 97 25  
## Yes 20 58

# calculate prediction accuracy  
mean(tree.pred == High.test)

## [1] 0.775