Assignemnt-2

Avinash Ravipudi

2023-03-23

library(ISLR)  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(glmnet)

## Loading required package: Matrix

## Loaded glmnet 4.1-6

library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

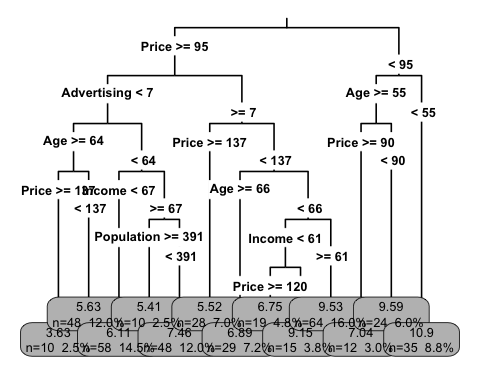
# filter the required attributes  
Carseats\_Filtered <- Carseats %>%   
 select("Sales", "Price", "Advertising", "Population", "Age", "Income", "Education")

QB1) Build a decision tree regression model to predict Sales based on all other attributes (“Price”, “Advertising”, “Population”, “Age”, “Income” and “Education”). Which attribute is used at the top of the tree (the root node) for splitting? Hint: you can either plot () and text() functions or use the summary() function to see the decision tree rules.

library(rpart)  
# build decision tree regression model  
Carseats\_Tree <- rpart(Sales ~ Price + Advertising + Population + Age + Income + Education, data = Carseats\_Filtered, method = "anova")  
  
# print the summary of the decision tree  
summary(Carseats\_Tree)

## Call:  
## rpart(formula = Sales ~ Price + Advertising + Population + Age +   
## Income + Education, data = Carseats\_Filtered, method = "anova")  
## n= 400   
##   
## CP nsplit rel error xerror xstd  
## 1 0.14251535 0 1.0000000 1.0048554 0.06943207  
## 2 0.08034146 1 0.8574847 0.9253457 0.06549776  
## 3 0.06251702 2 0.7771432 0.8734717 0.06404059  
## 4 0.02925241 3 0.7146262 0.8445885 0.06013180  
## 5 0.02537341 4 0.6853738 0.8582760 0.06145080  
## 6 0.02127094 5 0.6600003 0.8642699 0.06044626  
## 7 0.02059174 6 0.6387294 0.8454185 0.05939355  
## 8 0.01632010 7 0.6181377 0.8336178 0.05949779  
## 9 0.01521801 8 0.6018176 0.8488291 0.06006585  
## 10 0.01042023 9 0.5865996 0.8595253 0.06015203  
## 11 0.01000559 10 0.5761793 0.8566919 0.06006183  
## 12 0.01000000 12 0.5561681 0.8528466 0.05992630  
##   
## Variable importance  
## Price Advertising Age Income Population Education   
## 49 18 16 8 6 3   
##   
## Node number 1: 400 observations, complexity param=0.1425153  
## mean=7.496325, MSE=7.955687   
## left son=2 (329 obs) right son=3 (71 obs)  
## Primary splits:  
## Price < 94.5 to the right, improve=0.14251530, (0 missing)  
## Advertising < 7.5 to the left, improve=0.07303226, (0 missing)  
## Age < 61.5 to the right, improve=0.07120203, (0 missing)  
## Income < 61.5 to the left, improve=0.02840494, (0 missing)  
## Population < 174.5 to the left, improve=0.01077467, (0 missing)  
##   
## Node number 2: 329 observations, complexity param=0.08034146  
## mean=7.001672, MSE=6.815199   
## left son=4 (174 obs) right son=5 (155 obs)  
## Primary splits:  
## Advertising < 6.5 to the left, improve=0.11402580, (0 missing)  
## Price < 136.5 to the right, improve=0.08411056, (0 missing)  
## Age < 63.5 to the right, improve=0.08091745, (0 missing)  
## Income < 60.5 to the left, improve=0.03394126, (0 missing)  
## Population < 23 to the left, improve=0.01831455, (0 missing)  
## Surrogate splits:  
## Population < 223 to the left, agree=0.599, adj=0.148, (0 split)  
## Education < 10.5 to the right, agree=0.565, adj=0.077, (0 split)  
## Age < 53.5 to the right, agree=0.547, adj=0.039, (0 split)  
## Income < 114.5 to the left, agree=0.547, adj=0.039, (0 split)  
## Price < 106.5 to the right, agree=0.544, adj=0.032, (0 split)  
##   
## Node number 3: 71 observations, complexity param=0.02537341  
## mean=9.788451, MSE=6.852836   
## left son=6 (36 obs) right son=7 (35 obs)  
## Primary splits:  
## Age < 54.5 to the right, improve=0.16595410, (0 missing)  
## Price < 75.5 to the right, improve=0.08365773, (0 missing)  
## Income < 30.5 to the left, improve=0.03322169, (0 missing)  
## Education < 10.5 to the right, improve=0.03019634, (0 missing)  
## Population < 268.5 to the left, improve=0.02383306, (0 missing)  
## Surrogate splits:  
## Advertising < 4.5 to the right, agree=0.606, adj=0.200, (0 split)  
## Price < 73 to the right, agree=0.592, adj=0.171, (0 split)  
## Population < 272.5 to the left, agree=0.592, adj=0.171, (0 split)  
## Income < 79.5 to the right, agree=0.592, adj=0.171, (0 split)  
## Education < 11.5 to the left, agree=0.577, adj=0.143, (0 split)  
##   
## Node number 4: 174 observations, complexity param=0.02127094  
## mean=6.169655, MSE=4.942347   
## left son=8 (58 obs) right son=9 (116 obs)  
## Primary splits:  
## Age < 63.5 to the right, improve=0.078712160, (0 missing)  
## Price < 130.5 to the right, improve=0.048919280, (0 missing)  
## Population < 26.5 to the left, improve=0.030421540, (0 missing)  
## Income < 67.5 to the left, improve=0.027749670, (0 missing)  
## Advertising < 0.5 to the left, improve=0.006795377, (0 missing)  
## Surrogate splits:  
## Income < 22.5 to the left, agree=0.678, adj=0.034, (0 split)  
## Price < 96.5 to the left, agree=0.672, adj=0.017, (0 split)  
## Population < 26.5 to the left, agree=0.672, adj=0.017, (0 split)  
##   
## Node number 5: 155 observations, complexity param=0.06251702  
## mean=7.935677, MSE=7.268151   
## left son=10 (28 obs) right son=11 (127 obs)  
## Primary splits:  
## Price < 136.5 to the right, improve=0.17659580, (0 missing)  
## Age < 73.5 to the right, improve=0.08000201, (0 missing)  
## Income < 60.5 to the left, improve=0.05360755, (0 missing)  
## Advertising < 13.5 to the left, improve=0.03920507, (0 missing)  
## Population < 399 to the left, improve=0.01037956, (0 missing)  
## Surrogate splits:  
## Advertising < 24.5 to the right, agree=0.826, adj=0.036, (0 split)  
##   
## Node number 6: 36 observations, complexity param=0.0163201  
## mean=8.736944, MSE=4.961043   
## left son=12 (12 obs) right son=13 (24 obs)  
## Primary splits:  
## Price < 89.5 to the right, improve=0.29079360, (0 missing)  
## Income < 39.5 to the left, improve=0.19043350, (0 missing)  
## Advertising < 11.5 to the left, improve=0.17891930, (0 missing)  
## Age < 75.5 to the right, improve=0.04316067, (0 missing)  
## Education < 14.5 to the left, improve=0.03411396, (0 missing)  
## Surrogate splits:  
## Advertising < 16.5 to the right, agree=0.722, adj=0.167, (0 split)  
## Income < 37.5 to the left, agree=0.722, adj=0.167, (0 split)  
## Age < 56.5 to the left, agree=0.694, adj=0.083, (0 split)  
##   
## Node number 7: 35 observations  
## mean=10.87, MSE=6.491674   
##   
## Node number 8: 58 observations, complexity param=0.01042023  
## mean=5.287586, MSE=3.93708   
## left son=16 (10 obs) right son=17 (48 obs)  
## Primary splits:  
## Price < 137 to the right, improve=0.14521540, (0 missing)  
## Education < 15.5 to the right, improve=0.07995394, (0 missing)  
## Income < 35.5 to the left, improve=0.04206708, (0 missing)  
## Age < 79.5 to the left, improve=0.02799057, (0 missing)  
## Population < 52.5 to the left, improve=0.01914342, (0 missing)  
##   
## Node number 9: 116 observations, complexity param=0.01000559  
## mean=6.61069, MSE=4.861446   
## left son=18 (58 obs) right son=19 (58 obs)  
## Primary splits:  
## Income < 67 to the left, improve=0.05085914, (0 missing)  
## Population < 392 to the right, improve=0.04476721, (0 missing)  
## Price < 127 to the right, improve=0.04210762, (0 missing)  
## Age < 37.5 to the right, improve=0.02858424, (0 missing)  
## Education < 14.5 to the left, improve=0.01187387, (0 missing)  
## Surrogate splits:  
## Education < 12.5 to the right, agree=0.586, adj=0.172, (0 split)  
## Age < 58.5 to the left, agree=0.578, adj=0.155, (0 split)  
## Price < 144.5 to the left, agree=0.569, adj=0.138, (0 split)  
## Population < 479 to the right, agree=0.560, adj=0.121, (0 split)  
## Advertising < 2.5 to the right, agree=0.543, adj=0.086, (0 split)  
##   
## Node number 10: 28 observations  
## mean=5.522857, MSE=5.084213   
##   
## Node number 11: 127 observations, complexity param=0.02925241  
## mean=8.467638, MSE=6.183142   
## left son=22 (29 obs) right son=23 (98 obs)  
## Primary splits:  
## Age < 65.5 to the right, improve=0.11854590, (0 missing)  
## Income < 51.5 to the left, improve=0.08076060, (0 missing)  
## Advertising < 13.5 to the left, improve=0.04801701, (0 missing)  
## Education < 11.5 to the right, improve=0.02471512, (0 missing)  
## Population < 479 to the left, improve=0.01908657, (0 missing)  
##   
## Node number 12: 12 observations  
## mean=7.038333, MSE=2.886964   
##   
## Node number 13: 24 observations  
## mean=9.58625, MSE=3.834123   
##   
## Node number 16: 10 observations  
## mean=3.631, MSE=5.690169   
##   
## Node number 17: 48 observations  
## mean=5.632708, MSE=2.88102   
##   
## Node number 18: 58 observations  
## mean=6.113448, MSE=3.739109   
##   
## Node number 19: 58 observations, complexity param=0.01000559  
## mean=7.107931, MSE=5.489285   
## left son=38 (10 obs) right son=39 (48 obs)  
## Primary splits:  
## Population < 390.5 to the right, improve=0.10993270, (0 missing)  
## Price < 124.5 to the right, improve=0.07534567, (0 missing)  
## Advertising < 0.5 to the left, improve=0.07060488, (0 missing)  
## Age < 45.5 to the right, improve=0.04611510, (0 missing)  
## Education < 11.5 to the right, improve=0.03722944, (0 missing)  
##   
## Node number 22: 29 observations  
## mean=6.893793, MSE=6.08343   
##   
## Node number 23: 98 observations, complexity param=0.02059174  
## mean=8.933367, MSE=5.262759   
## left son=46 (34 obs) right son=47 (64 obs)  
## Primary splits:  
## Income < 60.5 to the left, improve=0.12705480, (0 missing)  
## Advertising < 13.5 to the left, improve=0.07114001, (0 missing)  
## Price < 118.5 to the right, improve=0.06932216, (0 missing)  
## Education < 11.5 to the right, improve=0.03377416, (0 missing)  
## Age < 49.5 to the right, improve=0.02289004, (0 missing)  
## Surrogate splits:  
## Education < 17.5 to the right, agree=0.663, adj=0.029, (0 split)  
##   
## Node number 38: 10 observations  
## mean=5.406, MSE=2.508524   
##   
## Node number 39: 48 observations  
## mean=7.4625, MSE=5.381106   
##   
## Node number 46: 34 observations, complexity param=0.01521801  
## mean=7.811471, MSE=4.756548   
## left son=92 (19 obs) right son=93 (15 obs)  
## Primary splits:  
## Price < 119.5 to the right, improve=0.29945020, (0 missing)  
## Advertising < 11.5 to the left, improve=0.14268440, (0 missing)  
## Income < 40.5 to the right, improve=0.12781140, (0 missing)  
## Population < 152 to the left, improve=0.03601768, (0 missing)  
## Age < 49.5 to the right, improve=0.02748814, (0 missing)  
## Surrogate splits:  
## Education < 12.5 to the right, agree=0.676, adj=0.267, (0 split)  
## Advertising < 7.5 to the right, agree=0.647, adj=0.200, (0 split)  
## Age < 53.5 to the left, agree=0.647, adj=0.200, (0 split)  
## Population < 240 to the right, agree=0.618, adj=0.133, (0 split)  
## Income < 41.5 to the right, agree=0.618, adj=0.133, (0 split)  
##   
## Node number 47: 64 observations  
## mean=9.529375, MSE=4.5078   
##   
## Node number 92: 19 observations  
## mean=6.751053, MSE=3.378915   
##   
## Node number 93: 15 observations  
## mean=9.154667, MSE=3.273025

library(rpart.plot)  
# plot decision tree  
rpart.plot(Carseats\_Tree, type = 3, extra = 101, cex = 0.8, box.col = "grey", branch.lwd = 2, digits = 3)

 QB2) Consider the following input: • Sales=9 • Price=6.54 • Population=124 • Advertising=0 • Age=76 • Income= 110 • Education=10 What will be the estimated Sales for this record using the decision tree model?

# create a data frame with the input record  
input <- data.frame(Price = 6.54, Advertising = 0, Population = 124, Age = 76, Income = 110, Education = 10)  
  
# predict the Sales using the decision tree model  
predicted\_sales <- predict(Carseats\_Tree, newdata = input)  
  
# print the predicted Sales  
predicted\_sales

## 1   
## 9.58625

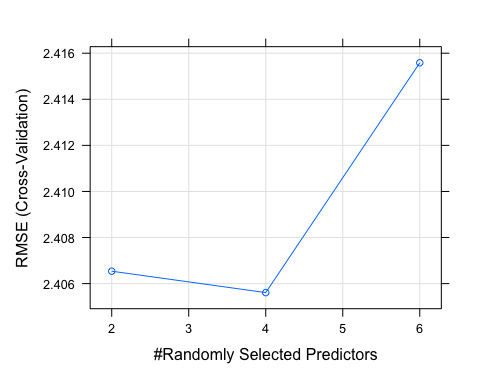
Estimated sales for this record is 9.59

QB3) Use the caret function to train a random forest (method=’rf’) for the same dataset. Use the caret default settings. By default, caret will examine the “mtry” values of 2,4, and 6. Recall that mtry is the number of attributes available for splitting at each splitting node. Which mtry value gives the best performance? (Make sure to set the random number generator seed to 123)

# set the seed for reproducibility  
set.seed(123)  
  
# define the train control  
train\_control <- trainControl(method = "cv", number = 5)  
  
# train the random forest model  
Carseats\_RandomForest <- train(Sales ~ Price + Advertising + Population + Age + Income + Education, data = Carseats\_Filtered, method = "rf", trControl = train\_control)  
  
# print the results  
Carseats\_RandomForest

## Random Forest   
##   
## 400 samples  
## 6 predictor  
##   
## No pre-processing  
## Resampling: Cross-Validated (5 fold)   
## Summary of sample sizes: 320, 321, 319, 320, 320   
## Resampling results across tuning parameters:  
##   
## mtry RMSE Rsquared MAE   
## 2 2.406539 0.2838955 1.926998  
## 4 2.405609 0.2874877 1.916925  
## 6 2.415585 0.2834264 1.924429  
##   
## RMSE was used to select the optimal model using the smallest value.  
## The final value used for the model was mtry = 4.

# plot the results  
plot(Carseats\_RandomForest)

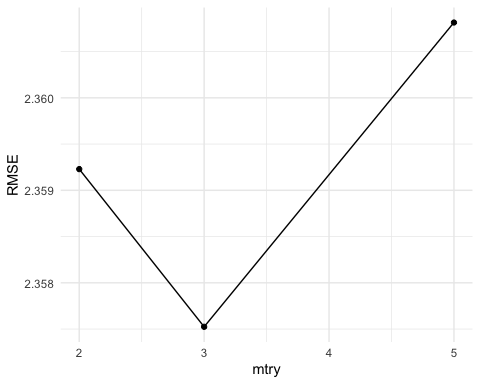
 The Mtr value at 4 gives the best performanace

QB4) Customize the search grid by checking the model’s performance for mtry values of 2, 3 and 5 using 3 repeats of 5-fold cross validation.

# define the search grid  
mtry\_grid <- expand.grid(mtry = c(2, 3, 5))  
  
# define the train control with 3 repeats of 5-fold cross validation  
train\_control <- trainControl(method = "repeatedcv", repeats = 3, number = 5)  
  
# train the random forest model with the customized search grid  
Carseats\_RandomForest <- train(Sales ~ Price + Advertising + Population + Age + Income + Education,   
 data = Carseats\_Filtered,   
 method = "rf",   
 trControl = train\_control,   
 tuneLength = 3,   
 tuneGrid = mtry\_grid)  
  
# print the results  
Carseats\_RandomForest

## Random Forest   
##   
## 400 samples  
## 6 predictor  
##   
## No pre-processing  
## Resampling: Cross-Validated (5 fold, repeated 3 times)   
## Summary of sample sizes: 319, 320, 320, 321, 320, 320, ...   
## Resampling results across tuning parameters:  
##   
## mtry RMSE Rsquared MAE   
## 2 2.359229 0.3091352 1.887352  
## 3 2.357524 0.3091737 1.884695  
## 5 2.360814 0.3095562 1.886026  
##   
## RMSE was used to select the optimal model using the smallest value.  
## The final value used for the model was mtry = 3.

library(ggplot2)  
  
# extract the results from the train object  
results <- as.data.frame(Carseats\_RandomForest$results)  
  
# create a line plot of RMSE as a function of mtry  
ggplot(results, aes(x = mtry, y = RMSE)) +  
 geom\_line() +  
 geom\_point() +  
 labs(x = "mtry", y = "RMSE") +  
 theme\_minimal()

 mtry values of 2, 3 and 5 using 3 repeats of 5-fold cross validation is at 3