Devin Hopkins

4190350

Assignment 2

|  |  |  |
| --- | --- | --- |
| **Step Number** | **Command** | **Output to Console (Other Output)** |
| 1 | library(readr) | Warning message:  package ‘readr’ was built under R version 3.6.3 |
|  | CarseatsSpam <- read\_csv("C:/Users/User/Desktop/School/Data\_Mining/Assignment2/CarseatsSpam.csv") | Parsed with column specification:  cols(  CompPrice = col\_double(),  Income = col\_double(),  Advertising = col\_double(),  Population = col\_double(),  Price = col\_double(),  ShelveLoc = col\_character(),  Age = col\_double(),  Education = col\_double(),  Urban = col\_character(),  US = col\_character(),  Spam = col\_character()  ) |
|  | View(CarseatsSpam) | None (displayed the data in R Studios) |
| 2 | dim(CarseatsSpam) | [1] 400 11 |
| 3 | sum(is.na(CarseatsSpam)) | [1] 0 |
| 4 | CarseatsSpam$ShelveLoc <- factor(CarseatsSpam$ShelveLoc) | None (changed the column ShelveLoc to numbers) |
|  | CarseatsSpam$Spam <- factor(CarseatsSpam$Spam) | None (changed the column Spam to numbers) |
|  | CarseatsSpam$US <- factor(CarseatsSpam$US) | None (changed the column US to numbers) |
|  | CarseatsSpam$Urban <- factor(CarseatsSpam$Urban) | None (changed the column Urban to numbers) |
| 5 | library(tree) | Warning message:  package ‘tree’ was built under R version 3.6.3 |
| 6 | tree.Spam <- tree(Spam ~ ., CarseatsSpam) | None (created the variable tree.Spam) |
| 7 | summary(tree.Spam) | Classification tree:  tree(formula = Spam ~ ., data = CarseatsSpam)  Variables actually used in tree construction:  [1] "ShelveLoc" "Price" "Income" "CompPrice" "Population" "Advertising"  [7] "Age" "US"  Number of terminal nodes: 27  Residual mean deviance: 0.4575 = 170.7 / 373  Misclassification error rate: 0.09 = 36 / 400 |
| 8 | plot(tree.Spam) | None (displayed the plot seen below) |
| 9 | text(tree.Spam, pretty = 0) | None (added text to the plot as seen below) |
| 10 | tree.Spam | node), split, n, deviance, yval, (yprob)  \* denotes terminal node  1) root 400 541.500 No ( 0.59000 0.41000 )  2) ShelveLoc: Bad,Medium 315 390.600 No ( 0.68889 0.31111 )  4) Price < 92.5 46 56.530 Yes ( 0.30435 0.69565 )  8) Income < 57 10 12.220 No ( 0.70000 0.30000 )  16) CompPrice < 110.5 5 0.000 No ( 1.00000 0.00000 ) \*  17) CompPrice > 110.5 5 6.730 Yes ( 0.40000 0.60000 ) \*  9) Income > 57 36 35.470 Yes ( 0.19444 0.80556 )  18) Population < 207.5 16 21.170 Yes ( 0.37500 0.62500 ) \*  19) Population > 207.5 20 7.941 Yes ( 0.05000 0.95000 ) \*  5) Price > 92.5 269 299.800 No ( 0.75465 0.24535 )  10) Advertising < 13.5 224 213.200 No ( 0.81696 0.18304 )  20) CompPrice < 124.5 96 44.890 No ( 0.93750 0.06250 )  40) Price < 106.5 38 33.150 No ( 0.84211 0.15789 )  80) Population < 177 12 16.300 No ( 0.58333 0.41667 )  160) Income < 60.5 6 0.000 No ( 1.00000 0.00000 ) \*  161) Income > 60.5 6 5.407 Yes ( 0.16667 0.83333 ) \*  81) Population > 177 26 8.477 No ( 0.96154 0.03846 ) \*  41) Price > 106.5 58 0.000 No ( 1.00000 0.00000 ) \*  21) CompPrice > 124.5 128 150.200 No ( 0.72656 0.27344 )  42) Price < 122.5 51 70.680 Yes ( 0.49020 0.50980 )  84) ShelveLoc: Bad 11 6.702 No ( 0.90909 0.09091 ) \*  85) ShelveLoc: Medium 40 52.930 Yes ( 0.37500 0.62500 )  170) Price < 109.5 16 7.481 Yes ( 0.06250 0.93750 ) \*  171) Price > 109.5 24 32.600 No ( 0.58333 0.41667 )  342) Age < 49.5 13 16.050 Yes ( 0.30769 0.69231 ) \*  343) Age > 49.5 11 6.702 No ( 0.90909 0.09091 ) \*  43) Price > 122.5 77 55.540 No ( 0.88312 0.11688 )  86) CompPrice < 147.5 58 17.400 No ( 0.96552 0.03448 ) \*  87) CompPrice > 147.5 19 25.010 No ( 0.63158 0.36842 )  174) Price < 147 12 16.300 Yes ( 0.41667 0.58333 )  348) CompPrice < 152.5 7 5.742 Yes ( 0.14286 0.85714 ) \*  349) CompPrice > 152.5 5 5.004 No ( 0.80000 0.20000 ) \*  175) Price > 147 7 0.000 No ( 1.00000 0.00000 ) \*  11) Advertising > 13.5 45 61.830 Yes ( 0.44444 0.55556 )  22) Age < 54.5 25 25.020 Yes ( 0.20000 0.80000 )  44) CompPrice < 130.5 14 18.250 Yes ( 0.35714 0.64286 )  88) Income < 100 9 12.370 No ( 0.55556 0.44444 ) \*  89) Income > 100 5 0.000 Yes ( 0.00000 1.00000 ) \*  45) CompPrice > 130.5 11 0.000 Yes ( 0.00000 1.00000 ) \*  23) Age > 54.5 20 22.490 No ( 0.75000 0.25000 )  46) CompPrice < 122.5 10 0.000 No ( 1.00000 0.00000 ) \*  47) CompPrice > 122.5 10 13.860 No ( 0.50000 0.50000 )  94) Price < 125 5 0.000 Yes ( 0.00000 1.00000 ) \*  95) Price > 125 5 0.000 No ( 1.00000 0.00000 ) \*  3) ShelveLoc: Good 85 90.330 Yes ( 0.22353 0.77647 )  6) Price < 135 68 49.260 Yes ( 0.11765 0.88235 )  12) US: No 17 22.070 Yes ( 0.35294 0.64706 )  24) Price < 109 8 0.000 Yes ( 0.00000 1.00000 ) \*  25) Price > 109 9 11.460 No ( 0.66667 0.33333 ) \*  13) US: Yes 51 16.880 Yes ( 0.03922 0.96078 ) \*  7) Price > 135 17 22.070 No ( 0.64706 0.35294 )  14) Income < 46 6 0.000 No ( 1.00000 0.00000 ) \*  15) Income > 46 11 15.160 Yes ( 0.45455 0.54545 ) \* |
| 11 | tree.Spam.predict <- predict(tree.Spam, CarseatsSpam, type = "class") | None (created the variable “tree.Spam.predict”) |
| 12 | None | A ton of downloaded and installed package messages |
| 13 | confusionMatrix(tree.Spam.predict, CarseatsSpam$Spam) | Confusion Matrix and Statistics  Reference  Prediction No Yes  No 213 13  Yes 23 151    Accuracy : 0.91  95% CI : (0.8776, 0.9362)  No Information Rate : 0.59  P-Value [Acc > NIR] : <2e-16    Kappa : 0.8157    Mcnemar's Test P-Value : 0.1336    Sensitivity : 0.9025  Specificity : 0.9207  Pos Pred Value : 0.9425  Neg Pred Value : 0.8678  Prevalence : 0.5900  Detection Rate : 0.5325  Detection Prevalence : 0.5650  Balanced Accuracy : 0.9116    'Positive' Class : No |