Decision Trees

2024-04-23

Decision Trees

Decision trees can be used for regression type problems or classification type problems. In the former case, they are called Regression Trees, and in the latter, Classification Trees.

The Basics

- We use the 'rpart' library from R to implement Decisions Trees (both for classification and regression)
- The function rpart() has a parameter called method. If the method is set to 'anova' the model will do regression. If the method is set to 'class' the model will be a classifier.
- There is also an optional control parameter, minsplit with default value of 30, which says how many observation we should have at least at each node before attempting to split it further.

```
library(rpart) #We will use this library for decision trees
library(rpart.plot)
```

```
## Warning: package 'rpart.plot' was built under R version 4.3.3
```

```
library(rattle) # We will use this library to print out a "fancy" tree
```

```
## Loading required package: tibble
```

```
## Loading required package: bitops
```

```
## Warning: package 'bitops' was built under R version 4.3.3
```

```
## 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.
```

Example - Regression Tree

```
library(ISLR)
MyData <- Carseats[,1:8]
set.seed(2342)
Model1 = rpart(Sales~.,data=MyData,method="anova") # Use regression tree
summary(Model1)
```

```
## Call:
## rpart(formula = Sales ~ ., data = MyData, method = "anova")
##   n= 400
##
##            CP nsplit rel error    xerror       xstd
## 1  0.25051039      0 1.0000000 1.0085959 0.06965686
## 2  0.10507256      1 0.7494896 0.7597915 0.05157154
## 3  0.05112059      2 0.6444171 0.6731735 0.04558220
## 4  0.04567126      3 0.5932965 0.6375077 0.04162517
## 5  0.03359237      4 0.5476252 0.6051916 0.04158015
## 6  0.02406279      5 0.5140328 0.6209222 0.04222797
## 7  0.02394780      6 0.4899700 0.6362229 0.04162421
## 8  0.02216327      7 0.4660222 0.6287085 0.04171513
## 9  0.01604252      8 0.4438590 0.5965567 0.04024666
## 10 0.01402704      9 0.4278165 0.5905546 0.03972022
## 11 0.01314537     11 0.3997624 0.5889412 0.04016603
## 12 0.01271091     12 0.3866170 0.5873226 0.04032371
## 13 0.01214708     13 0.3739061 0.5805813 0.03988911
## 14 0.01188778     14 0.3617590 0.5748148 0.03931995
## 15 0.01077845     15 0.3498712 0.5744957 0.03873513
## 16 0.01050614     16 0.3390928 0.5778249 0.03820342
## 17 0.01000000     17 0.3285866 0.5799048 0.03752935
##
## Variable importance
##   ShelveLoc       Price   CompPrice Advertising      Income         Age
##          40          26           9           8           7           6
##  Population
##           4
##
## Node number 1: 400 observations,    complexity param=0.2505104
##   mean=7.496325, MSE=7.955687
##   left son=2 (315 obs) right son=3 (85 obs)
##   Primary splits:
##       ShelveLoc   splits as  LRL,       improve=0.25051040, (0 missing)
##       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)
##
## Node number 2: 315 observations,    complexity param=0.1050726
##   mean=6.762984, MSE=5.903364
##   left son=4 (207 obs) right son=5 (108 obs)
##   Primary splits:
##       Price       < 105.5 to the right, improve=0.17981130, (0 missing)
##       ShelveLoc   splits as  L-R,       improve=0.11418740, (0 missing)
##       Advertising < 7.5   to the left,  improve=0.09324535, (0 missing)
##       Age         < 68.5  to the right, improve=0.06549277, (0 missing)
##       Income      < 60.5  to the left,  improve=0.04926766, (0 missing)
##   Surrogate splits:
##       CompPrice  < 113.5 to the right, agree=0.749, adj=0.269, (0 split)
##       Population < 507.5 to the left,  agree=0.667, adj=0.028, (0 split)
##       Income     < 22.5  to the right, agree=0.660, adj=0.009, (0 split)
##
## Node number 3: 85 observations,    complexity param=0.05112059
```

```
##   mean=10.214, MSE=6.182615
##   left son=6 (57 obs) right son=7 (28 obs)
##   Primary splits:
##       Price       < 109.5 to the right, improve=0.30955830, (0 missing)
##       Age         < 61.5  to the right, improve=0.15092700, (0 missing)
##       Advertising < 13.5  to the left,  improve=0.11044180, (0 missing)
##       Population  < 345.5 to the left,  improve=0.06445623, (0 missing)
##       Income      < 35    to the left,  improve=0.03453837, (0 missing)
##   Surrogate splits:
##       CompPrice  < 113.5 to the right, agree=0.729, adj=0.179, (0 split)
##       Population < 77    to the right, agree=0.706, adj=0.107, (0 split)
##       Age        < 26.5  to the right, agree=0.694, adj=0.071, (0 split)
##
## Node number 4: 207 observations,    complexity param=0.04567126
##   mean=6.018792, MSE=4.621123
##   left son=8 (61 obs) right son=9 (146 obs)
##   Primary splits:
##       ShelveLoc   splits as  L-R,       improve=0.15193670, (0 missing)
##       CompPrice   < 124.5 to the left,  improve=0.10356310, (0 missing)
##       Advertising < 10.5  to the left,  improve=0.09626173, (0 missing)
##       Price       < 135.5 to the right, improve=0.09312515, (0 missing)
##       Age         < 50.5  to the right, improve=0.06643781, (0 missing)
##   Surrogate splits:
##       Population < 14.5  to the left,  agree=0.715, adj=0.033, (0 split)
##
## Node number 5: 108 observations,    complexity param=0.03359237
##   mean=8.189352, MSE=5.264976
##   left son=10 (65 obs) right son=11 (43 obs)
##   Primary splits:
##       Age       < 54.5  to the right, improve=0.1880001, (0 missing)
##       CompPrice < 123.5 to the left,  improve=0.1855506, (0 missing)
##       ShelveLoc splits as  L-R,       improve=0.1471907, (0 missing)
##       Price     < 88    to the right, improve=0.1193571, (0 missing)
##       Income    < 57.5  to the left,  improve=0.1097152, (0 missing)
##   Surrogate splits:
##       CompPrice   < 132.5 to the left,  agree=0.685, adj=0.209, (0 split)
##       Advertising < 17.5  to the left,  agree=0.630, adj=0.070, (0 split)
##       Population  < 494   to the left,  agree=0.630, adj=0.070, (0 split)
##       Price       < 77.5  to the right, agree=0.630, adj=0.070, (0 split)
##
## Node number 6: 57 observations,    complexity param=0.02406279
##   mean=9.244386, MSE=4.864302
##   left son=12 (48 obs) right son=13 (9 obs)
##   Primary splits:
##       Advertising < 13.5  to the left,  improve=0.2761775, (0 missing)
##       Price       < 135   to the right, improve=0.2437479, (0 missing)
##       Age         < 61.5  to the right, improve=0.1851980, (0 missing)
##       Income      < 35.5  to the left,  improve=0.1424100, (0 missing)
##       Population  < 345.5 to the left,  improve=0.1076962, (0 missing)
##
## Node number 7: 28 observations
##   mean=12.18786, MSE=3.056331
##
## Node number 8: 61 observations,    complexity param=0.01214708
```

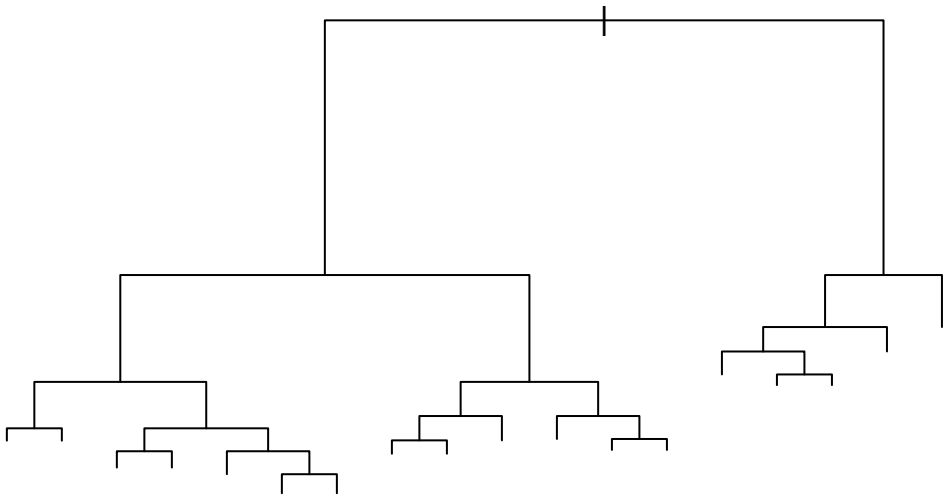
```
##   mean=4.722459, MSE=3.947864
##   left son=16 (25 obs) right son=17 (36 obs)
##   Primary splits:
##       Population  < 196.5 to the left,  improve=0.16051570, (0 missing)
##       Price       < 143.5 to the right, improve=0.13922810, (0 missing)
##       Age         < 61.5  to the right, improve=0.13117950, (0 missing)
##       CompPrice   < 124.5 to the left,  improve=0.11991600, (0 missing)
##       Advertising < 10.5  to the left,  improve=0.06447994, (0 missing)
##   Surrogate splits:
##       Advertising < 1.5   to the left,  agree=0.787, adj=0.48, (0 split)
##       Age         < 67.5  to the right, agree=0.639, adj=0.12, (0 split)
##       Price       < 118.5 to the left,  agree=0.623, adj=0.08, (0 split)
##       CompPrice   < 116.5 to the left,  agree=0.607, adj=0.04, (0 split)
##
## Node number 9: 146 observations,    complexity param=0.02216327
##   mean=6.560411, MSE=3.906946
##   left son=18 (77 obs) right son=19 (69 obs)
##   Primary splits:
##       Advertising < 5.5   to the left,  improve=0.1236463, (0 missing)
##       Age         < 47.5  to the right, improve=0.1217696, (0 missing)
##       CompPrice   < 124.5 to the left,  improve=0.1212571, (0 missing)
##       Price       < 127   to the right, improve=0.1151467, (0 missing)
##       Income      < 57.5  to the left,  improve=0.1037510, (0 missing)
##   Surrogate splits:
##       Population < 208.5 to the left,  agree=0.603, adj=0.159, (0 split)
##       CompPrice  < 131.5 to the right, agree=0.568, adj=0.087, (0 split)
##       Income     < 78.5  to the right, agree=0.562, adj=0.072, (0 split)
##       Age        < 50.5  to the right, agree=0.562, adj=0.072, (0 split)
##       Price      < 115.5 to the right, agree=0.548, adj=0.043, (0 split)
##
## Node number 10: 65 observations,    complexity param=0.0239478
##   mean=7.380154, MSE=4.662414
##   left son=20 (56 obs) right son=21 (9 obs)
##   Primary splits:
##       Income    < 105.5 to the left,  improve=0.25146590, (0 missing)
##       Price     < 89.5  to the right, improve=0.23921200, (0 missing)
##       ShelveLoc splits as  L-R,       improve=0.13768880, (0 missing)
##       CompPrice < 123.5 to the left,  improve=0.13500380, (0 missing)
##       Age       < 68.5  to the right, improve=0.08350763, (0 missing)
##   Surrogate splits:
##       Price < 68.5  to the right, agree=0.877, adj=0.111, (0 split)
##
## Node number 11: 43 observations,    complexity param=0.01188778
##   mean=9.412558, MSE=3.689777
##   left son=22 (13 obs) right son=23 (30 obs)
##   Primary splits:
##       Income      < 57.5  to the left,  improve=0.2384349, (0 missing)
##       ShelveLoc   splits as  L-R,       improve=0.2239864, (0 missing)
##       CompPrice   < 124   to the left,  improve=0.1612473, (0 missing)
##       Advertising < 9.5   to the left,  improve=0.1569666, (0 missing)
##       Age         < 31    to the right, improve=0.1034396, (0 missing)
##
## Node number 12: 48 observations,    complexity param=0.01271091
##   mean=8.7425, MSE=3.862923
```

```
##   left son=24 (12 obs) right son=25 (36 obs)
##   Primary splits:
##       Price       < 142.5 to the right, improve=0.21815090, (0 missing)
##       Age         < 60.5  to the right, improve=0.18938760, (0 missing)
##       Income      < 35.5  to the left,  improve=0.16116570, (0 missing)
##       Advertising < 0.5   to the left,  improve=0.15901940, (0 missing)
##       CompPrice   < 146.5 to the left,  improve=0.09296587, (0 missing)
##   Surrogate splits:
##       CompPrice  < 154.5 to the right, agree=0.792, adj=0.167, (0 split)
##       Income     < 29.5  to the left,  agree=0.771, adj=0.083, (0 split)
##       Population < 138.5 to the left,  agree=0.771, adj=0.083, (0 split)
##       Age        < 33    to the left,  agree=0.771, adj=0.083, (0 split)
##
## Node number 13: 9 observations
##   mean=11.92111, MSE=1.696721
##
## Node number 16: 25 observations
##   mean=3.7672, MSE=3.529172
##
## Node number 17: 36 observations
##   mean=5.385833, MSE=3.164863
##
## Node number 18: 77 observations,    complexity param=0.01604252
##   mean=5.902468, MSE=3.637837
##   left son=36 (34 obs) right son=37 (43 obs)
##   Primary splits:
##       Price      < 127   to the right, improve=0.18225370, (0 missing)
##       Age        < 45.5  to the right, improve=0.15208150, (0 missing)
##       CompPrice  < 124.5 to the left,  improve=0.14069050, (0 missing)
##       Income     < 62.5  to the left,  improve=0.12376450, (0 missing)
##       Population < 310.5 to the right, improve=0.04555018, (0 missing)
##   Surrogate splits:
##       CompPrice   < 133.5 to the right, agree=0.688, adj=0.294, (0 split)
##       Income      < 40.5  to the left,  agree=0.636, adj=0.176, (0 split)
##       Advertising < 4.5   to the right, agree=0.597, adj=0.088, (0 split)
##       Population  < 194.5 to the right, agree=0.597, adj=0.088, (0 split)
##
## Node number 19: 69 observations,    complexity param=0.01402704
##   mean=7.294638, MSE=3.185089
##   left son=38 (19 obs) right son=39 (50 obs)
##   Primary splits:
##       CompPrice   < 121.5 to the left,  improve=0.13522770, (0 missing)
##       Price       < 124.5 to the right, improve=0.12488490, (0 missing)
##       Income      < 60.5  to the left,  improve=0.12481490, (0 missing)
##       Age         < 54    to the right, improve=0.10232200, (0 missing)
##       Advertising < 13.5  to the left,  improve=0.08556251, (0 missing)
##   Surrogate splits:
##       Age < 74.5  to the right, agree=0.739, adj=0.053, (0 split)
##
## Node number 20: 56 observations,    complexity param=0.01314537
##   mean=6.946071, MSE=3.625588
##   left son=40 (20 obs) right son=41 (36 obs)
##   Primary splits:
##       ShelveLoc  splits as  L-R,       improve=0.2060364, (0 missing)
```

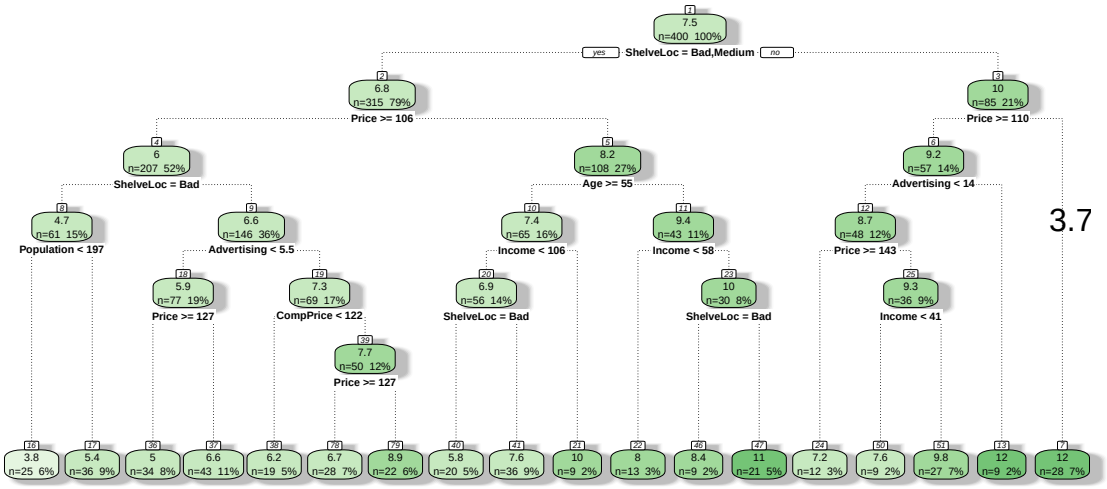
```
##       Age        < 68.5  to the right, improve=0.1548314, (0 missing)
##       Price      < 92    to the right, improve=0.1288858, (0 missing)
##       CompPrice  < 125.5 to the left,  improve=0.1238552, (0 missing)
##       Population < 272.5 to the right, improve=0.1094719, (0 missing)
##   Surrogate splits:
##       Advertising < 10.5  to the right, agree=0.714, adj=0.20, (0 split)
##       Income      < 88.5  to the right, agree=0.696, adj=0.15, (0 split)
##       Age         < 68.5  to the right, agree=0.696, adj=0.15, (0 split)
##       Price       < 85.5  to the left,  agree=0.661, adj=0.05, (0 split)
##
## Node number 21: 9 observations
##   mean=10.08111, MSE=2.646165
##
## Node number 22: 13 observations
##   mean=7.987692, MSE=1.480218
##
## Node number 23: 30 observations,    complexity param=0.01077845
##   mean=10.03, MSE=3.386247
##   left son=46 (9 obs) right son=47 (21 obs)
##   Primary splits:
##       ShelveLoc   splits as  L-R,       improve=0.33764030, (0 missing)
##       CompPrice   < 123   to the left,  improve=0.20687400, (0 missing)
##       Age         < 34.5  to the right, improve=0.08885492, (0 missing)
##       Advertising < 9.5   to the left,  improve=0.07675072, (0 missing)
##       Price       < 88    to the right, improve=0.05137712, (0 missing)
##   Surrogate splits:
##       Income < 108   to the right, agree=0.733, adj=0.111, (0 split)
##
## Node number 24: 12 observations
##   mean=7.1525, MSE=3.053935
##
## Node number 25: 36 observations,    complexity param=0.01050614
##   mean=9.2725, MSE=3.008985
##   left son=50 (9 obs) right son=51 (27 obs)
##   Primary splits:
##       Income      < 40.5  to the left,  improve=0.30864420, (0 missing)
##       Age         < 61    to the right, improve=0.26749430, (0 missing)
##       CompPrice   < 121.5 to the left,  improve=0.21834020, (0 missing)
##       Advertising < 0.5   to the left,  improve=0.19501770, (0 missing)
##       Population  < 234   to the right, improve=0.02238297, (0 missing)
##
## Node number 36: 34 observations
##   mean=4.986765, MSE=3.92764
##
## Node number 37: 43 observations
##   mean=6.626512, MSE=2.221441
##
## Node number 38: 19 observations
##   mean=6.23, MSE=2.122821
##
## Node number 39: 50 observations,    complexity param=0.01402704
##   mean=7.6992, MSE=2.994367
##   left son=78 (28 obs) right son=79 (22 obs)
##   Primary splits:
```

```
##       Price       < 127   to the right, improve=0.39779130, (0 missing)
##       Advertising < 13.5  to the left,  improve=0.17705760, (0 missing)
##       Income      < 60.5  to the left,  improve=0.15779500, (0 missing)
##       Age         < 37.5  to the right, improve=0.08458044, (0 missing)
##       CompPrice   < 141.5 to the left,  improve=0.01681669, (0 missing)
##   Surrogate splits:
##       CompPrice   < 131.5 to the right, agree=0.70, adj=0.318, (0 split)
##       Age         < 64.5  to the left,  agree=0.66, adj=0.227, (0 split)
##       Income      < 60.5  to the left,  agree=0.62, adj=0.136, (0 split)
##       Population  < 92.5  to the right, agree=0.62, adj=0.136, (0 split)
##       Advertising < 11.5  to the left,  agree=0.60, adj=0.091, (0 split)
##
## Node number 40: 20 observations
##   mean=5.7865, MSE=3.848003
##
## Node number 41: 36 observations
##   mean=7.590278, MSE=2.340019
##
## Node number 46: 9 observations
##   mean=8.396667, MSE=2.528489
##
## Node number 47: 21 observations
##   mean=10.73, MSE=2.120524
##
## Node number 50: 9 observations
##   mean=7.603333, MSE=1.091978
##
## Node number 51: 27 observations
##   mean=9.828889, MSE=2.409714
##
## Node number 78: 28 observations
##   mean=6.731786, MSE=2.571229
##
## Node number 79: 22 observations
##   mean=8.930455, MSE=0.8257862
```

```
plot(Model1)
```



```
fancyRpartPlot(Model1)
text(Model1)
```



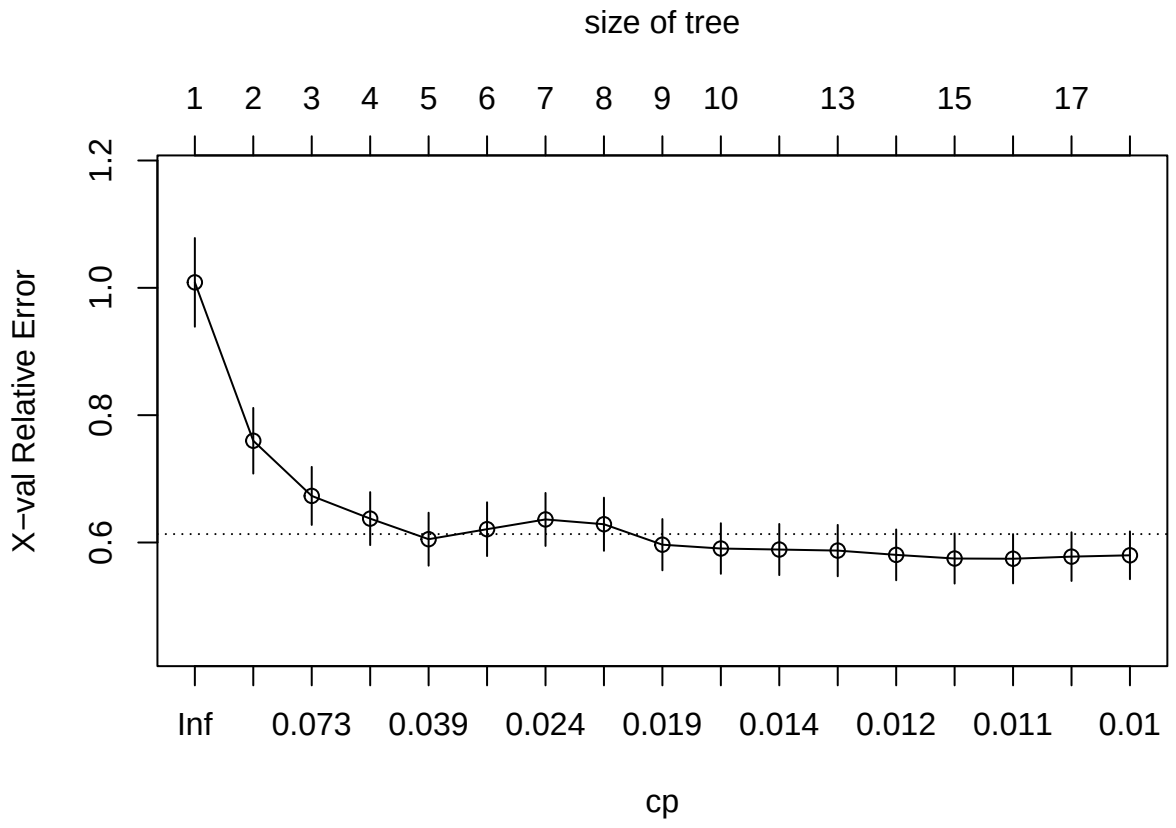
Rattle 2025−Mar−31 00:14:25 muralishanker

What is cp, or cost complexity pruning?

For each value of α there corresponds a subtree $T \subset T_0 \in \sum_{m=1}^{|T|} \sum_{x_i \in R_m} (y_i - \hat{Y}_{R_m})^2 + \alpha|T|$

Let us now use this information to prune the model

```
plotcp(Model1)
```




```
printcp(Model1)
```

```
##
## Regression tree:
## rpart(formula = Sales ~ ., data = MyData, method = "anova")
##
## Variables actually used in tree construction:
## [1] Advertising Age         CompPrice   Income      Population  Price
## [7] ShelveLoc
##
## Root node error: 3182.3/400 = 7.9557
##
## n= 400
##
##          CP nsplit rel error  xerror     xstd
## 1  0.250510      0   1.00000 1.00860 0.069657
## 2  0.105073      1   0.74949 0.75979 0.051572
## 3  0.051121      2   0.64442 0.67317 0.045582
## 4  0.045671      3   0.59330 0.63751 0.041625
## 5  0.033592      4   0.54763 0.60519 0.041580
## 6  0.024063      5   0.51403 0.62092 0.042228
## 7  0.023948      6   0.48997 0.63622 0.041624
## 8  0.022163      7   0.46602 0.62871 0.041715
## 9  0.016043      8   0.44386 0.59656 0.040247
## 10 0.014027      9   0.42782 0.59055 0.039720
```

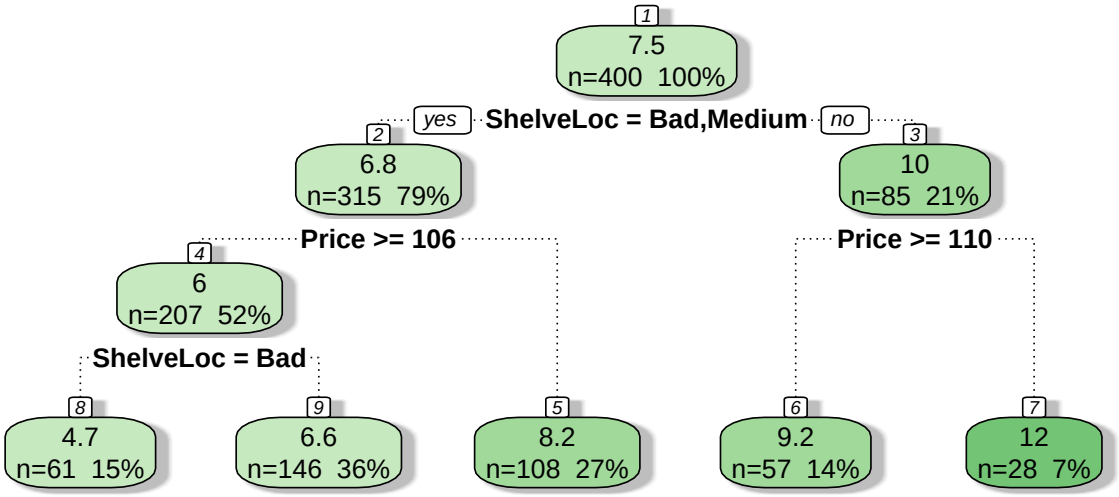
```
## 11 0.013145     11   0.39976 0.58894 0.040166
## 12 0.012711     12   0.38662 0.58732 0.040324
## 13 0.012147     13   0.37391 0.58058 0.039889
## 14 0.011888     14   0.36176 0.57481 0.039320
## 15 0.010778     15   0.34987 0.57450 0.038735
## 16 0.010506     16   0.33909 0.57782 0.038203
## 17 0.010000     17   0.32859 0.57990 0.037529
```

```
P_Model1 = prune.rpart(Model1,cp=0.03359237)
summary(P_Model1)
```

```
## Call:
## rpart(formula = Sales ~ ., data = MyData, method = "anova")
##   n= 400
##
##           CP nsplit rel error    xerror       xstd
## 1 0.25051039      0 1.0000000 1.0085959 0.06965686
## 2 0.10507256      1 0.7494896 0.7597915 0.05157154
## 3 0.05112059      2 0.6444171 0.6731735 0.04558220
## 4 0.04567126      3 0.5932965 0.6375077 0.04162517
## 5 0.03359237      4 0.5476252 0.6051916 0.04158015
##
## Variable importance
##  ShelveLoc      Price  CompPrice Population        Age
##         59         31          7          2          1
##
## Node number 1: 400 observations,    complexity param=0.2505104
##   mean=7.496325, MSE=7.955687
##   left son=2 (315 obs) right son=3 (85 obs)
##   Primary splits:
##       ShelveLoc   splits as  LRL,       improve=0.25051040, (0 missing)
##       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)
##
## Node number 2: 315 observations,    complexity param=0.1050726
##   mean=6.762984, MSE=5.903364
##   left son=4 (207 obs) right son=5 (108 obs)
##   Primary splits:
##       Price       < 105.5 to the right, improve=0.17981130, (0 missing)
##       ShelveLoc   splits as  L-R,       improve=0.11418740, (0 missing)
##       Advertising < 7.5   to the left,  improve=0.09324535, (0 missing)
##       Age         < 68.5  to the right, improve=0.06549277, (0 missing)
##       Income      < 60.5  to the left,  improve=0.04926766, (0 missing)
##   Surrogate splits:
##       CompPrice  < 113.5 to the right, agree=0.749, adj=0.269, (0 split)
##       Population < 507.5 to the left,  agree=0.667, adj=0.028, (0 split)
##       Income     < 22.5  to the right, agree=0.660, adj=0.009, (0 split)
##
## Node number 3: 85 observations,    complexity param=0.05112059
##   mean=10.214, MSE=6.182615
##   left son=6 (57 obs) right son=7 (28 obs)
##   Primary splits:
```

```
##       Price       < 109.5 to the right, improve=0.30955830, (0 missing)
##       Age         < 61.5  to the right, improve=0.15092700, (0 missing)
##       Advertising < 13.5  to the left,  improve=0.11044180, (0 missing)
##       Population  < 345.5 to the left,  improve=0.06445623, (0 missing)
##       Income      < 35    to the left,  improve=0.03453837, (0 missing)
##   Surrogate splits:
##       CompPrice  < 113.5 to the right, agree=0.729, adj=0.179, (0 split)
##       Population < 77    to the right, agree=0.706, adj=0.107, (0 split)
##       Age        < 26.5  to the right, agree=0.694, adj=0.071, (0 split)
##
## Node number 4: 207 observations,    complexity param=0.04567126
##   mean=6.018792, MSE=4.621123
##   left son=8 (61 obs) right son=9 (146 obs)
##   Primary splits:
##       ShelveLoc   splits as  L-R,       improve=0.15193670, (0 missing)
##       CompPrice   < 124.5 to the left,  improve=0.10356310, (0 missing)
##       Advertising < 10.5  to the left,  improve=0.09626173, (0 missing)
##       Price       < 135.5 to the right, improve=0.09312515, (0 missing)
##       Age         < 50.5  to the right, improve=0.06643781, (0 missing)
##   Surrogate splits:
##       Population < 14.5  to the left,  agree=0.715, adj=0.033, (0 split)
##
## Node number 5: 108 observations
##   mean=8.189352, MSE=5.264976
##
## Node number 6: 57 observations
##   mean=9.244386, MSE=4.864302
##
## Node number 7: 28 observations
##   mean=12.18786, MSE=3.056331
##
## Node number 8: 61 observations
##   mean=4.722459, MSE=3.947864
##
## Node number 9: 146 observations
##   mean=6.560411, MSE=3.906946
```

```
fancyRpartPlot(P_Model1)
```



Classification Trees

A classification tree is similar to a regression tree, except that it is used to predict a qualitative response rather than a quantitative one. In classification trees, we predict that each observation belongs to the *most commonly occuring class* of training observations in the region to which it belongs.

Criteria

- Classification error rate

$E = 1 - \max_k(\hat{p}_{mk})$ where $\hat{p}_{mk}$represents the proportion of training observations in the mth region that are from the kth class

* Gini Index

$$G = 1 - \sum_{k=1}^{K} \hat{p}_{mk}^2$$

- Cross-entropy

$$D = -\sum_{k=1}^{K} \hat{p}_{mk} log(\hat{p}_{mk})$$

Example

```r
attach(MyData)
High = ifelse(Sales<=8, "No","Yes") # Create a qualitative response
MyData = data.frame(MyData,High)
ModelC = rpart(High~.-Sales,data=MyData,method="class")
ModelC
```

```
## n= 400
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##   1) root 400 164 No (0.59000000 0.41000000)
##     2) ShelveLoc=Bad,Medium 315  98 No (0.68888889 0.31111111)
##       4) Price>=92.5 269  66 No (0.75464684 0.24535316)
##         8) Advertising< 13.5 224  41 No (0.81696429 0.18303571)
##          16) CompPrice< 124.5 96   6 No (0.93750000 0.06250000) *
##          17) CompPrice>=124.5 128  35 No (0.72656250 0.27343750)
##            34) Price>=109.5 107  20 No (0.81308411 0.18691589)
##              68) Price>=126.5 65   6 No (0.90769231 0.09230769) *
##              69) Price< 126.5 42  14 No (0.66666667 0.33333333)
##               138) Age>=49.5 22   2 No (0.90909091 0.09090909) *
##               139) Age< 49.5 20   8 Yes (0.40000000 0.60000000) *
##            35) Price< 109.5 21   6 Yes (0.28571429 0.71428571) *
##         9) Advertising>=13.5 45  20 Yes (0.44444444 0.55555556)
##          18) Age>=54.5 20   5 No (0.75000000 0.25000000) *
##          19) Age< 54.5 25   5 Yes (0.20000000 0.80000000) *
##       5) Price< 92.5 46  14 Yes (0.30434783 0.69565217)
##        10) Income< 57 10   3 No (0.70000000 0.30000000) *
##        11) Income>=57 36   7 Yes (0.19444444 0.80555556) *
##     3) ShelveLoc=Good 85  19 Yes (0.22352941 0.77647059)
##       6) Price>=142.5 12   3 No (0.75000000 0.25000000) *
##       7) Price< 142.5 73  10 Yes (0.13698630 0.86301370) *
```

```r
summary(ModelC)
```

```
## Call:
## rpart(formula = High ~ . - Sales, data = MyData, method = "class")
##   n= 400
##
##           CP nsplit rel error    xerror       xstd
## 1 0.28658537      0 1.0000000 1.0000000 0.05997967
## 2 0.10975610      1 0.7134146 0.7134146 0.05547692
## 3 0.04573171      2 0.6036585 0.6402439 0.05365767
## 4 0.03658537      4 0.5121951 0.6341463 0.05349198
## 5 0.02743902      5 0.4756098 0.6036585 0.05262923
## 6 0.02439024      7 0.4207317 0.5853659 0.05208331
## 7 0.01219512      8 0.3963415 0.5914634 0.05226769
## 8 0.01000000     10 0.3719512 0.6219512 0.05315381
##
## Variable importance
##       Price   ShelveLoc         Age Advertising   CompPrice      Income
##          35          25          11          11           9           5
##  Population
```

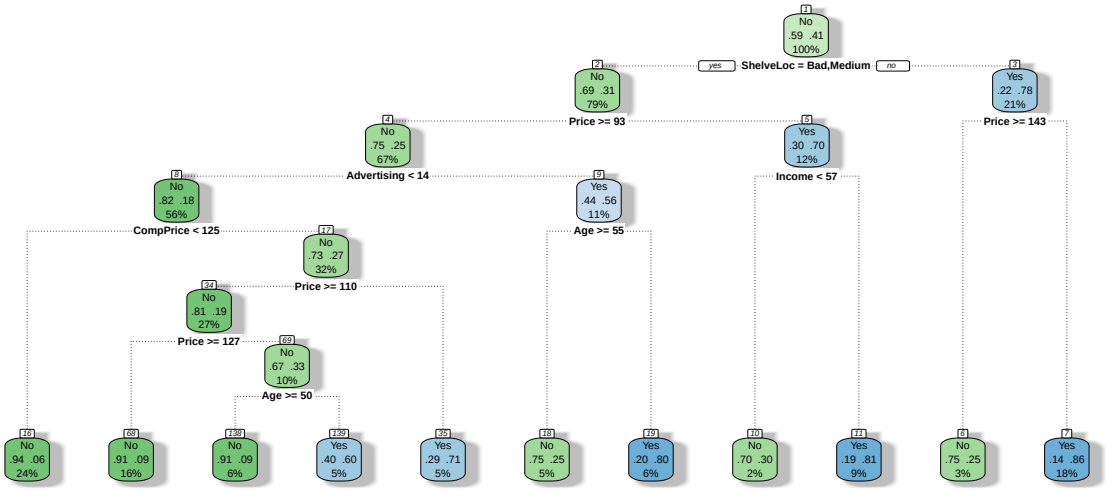
```
##           3
##
## Node number 1: 400 observations,    complexity param=0.2865854
##   predicted class=No   expected loss=0.41  P(node) =1
##     class counts:   236   164
##    probabilities: 0.590 0.410
##   left son=2 (315 obs) right son=3 (85 obs)
##   Primary splits:
##       ShelveLoc   splits as  LRL,       improve=28.991900, (0 missing)
##       Price       < 92.5  to the right, improve=19.463880, (0 missing)
##       Advertising < 6.5   to the left,  improve=17.277980, (0 missing)
##       Age         < 61.5  to the right, improve= 9.264442, (0 missing)
##       Income      < 60.5  to the left,  improve= 7.249032, (0 missing)
##
## Node number 2: 315 observations,    complexity param=0.1097561
##   predicted class=No   expected loss=0.3111111  P(node) =0.7875
##     class counts:   217    98
##    probabilities: 0.689 0.311
##   left son=4 (269 obs) right son=5 (46 obs)
##   Primary splits:
##       Price       < 92.5  to the right, improve=15.930580, (0 missing)
##       Advertising < 7.5   to the left,  improve=11.432570, (0 missing)
##       ShelveLoc   splits as  L-R,       improve= 7.543912, (0 missing)
##       Age         < 50.5  to the right, improve= 6.369905, (0 missing)
##       Income      < 60.5  to the left,  improve= 5.984509, (0 missing)
##   Surrogate splits:
##       CompPrice < 95.5  to the right, agree=0.873, adj=0.13, (0 split)
##
## Node number 3: 85 observations,    complexity param=0.03658537
##   predicted class=Yes  expected loss=0.2235294  P(node) =0.2125
##     class counts:    19    66
##    probabilities: 0.224 0.776
##   left son=6 (12 obs) right son=7 (73 obs)
##   Primary splits:
##       Price       < 142.5 to the right, improve=7.745608, (0 missing)
##       Income      < 35    to the left,  improve=4.529433, (0 missing)
##       Advertising < 6     to the left,  improve=3.739996, (0 missing)
##       Population  < 342   to the left,  improve=2.385882, (0 missing)
##       Age         < 61.5  to the right, improve=1.943953, (0 missing)
##   Surrogate splits:
##       CompPrice < 154.5 to the right, agree=0.882, adj=0.167, (0 split)
##
## Node number 4: 269 observations,    complexity param=0.04573171
##   predicted class=No   expected loss=0.2453532  P(node) =0.6725
##     class counts:   203    66
##    probabilities: 0.755 0.245
##   left son=8 (224 obs) right son=9 (45 obs)
##   Primary splits:
##       Advertising < 13.5  to the left,  improve=10.400090, (0 missing)
##       Age         < 49.5  to the right, improve= 8.083998, (0 missing)
##       ShelveLoc   splits as  L-R,       improve= 7.023150, (0 missing)
##       CompPrice   < 124.5 to the left,  improve= 6.749986, (0 missing)
##       Price       < 126.5 to the right, improve= 5.646063, (0 missing)
##
```

```
## Node number 5: 46 observations,    complexity param=0.02439024
##   predicted class=Yes  expected loss=0.3043478  P(node) =0.115
##     class counts:    14    32
##    probabilities: 0.304 0.696
##   left son=10 (10 obs) right son=11 (36 obs)
##   Primary splits:
##       Income      < 57    to the left,  improve=4.000483, (0 missing)
##       ShelveLoc   splits as  L-R,       improve=3.189762, (0 missing)
##       Advertising < 9.5   to the left,  improve=1.388592, (0 missing)
##       Price       < 80.5  to the right, improve=1.388592, (0 missing)
##       Age         < 64.5  to the right, improve=1.172885, (0 missing)
##
## Node number 6: 12 observations
##   predicted class=No   expected loss=0.25  P(node) =0.03
##     class counts:     9     3
##    probabilities: 0.750 0.250
##
## Node number 7: 73 observations
##   predicted class=Yes  expected loss=0.1369863  P(node) =0.1825
##     class counts:    10    63
##    probabilities: 0.137 0.863
##
## Node number 8: 224 observations,    complexity param=0.02743902
##   predicted class=No   expected loss=0.1830357  P(node) =0.56
##     class counts:   183    41
##    probabilities: 0.817 0.183
##   left son=16 (96 obs) right son=17 (128 obs)
##   Primary splits:
##       CompPrice   < 124.5 to the left,  improve=4.881696, (0 missing)
##       Age         < 49.5  to the right, improve=3.960418, (0 missing)
##       ShelveLoc   splits as  L-R,       improve=3.654633, (0 missing)
##       Price       < 126.5 to the right, improve=3.234428, (0 missing)
##       Advertising < 6.5   to the left,  improve=2.371276, (0 missing)
##   Surrogate splits:
##       Price      < 115.5 to the left,  agree=0.741, adj=0.396, (0 split)
##       Age        < 50.5  to the right, agree=0.634, adj=0.146, (0 split)
##       Population < 405   to the right, agree=0.629, adj=0.135, (0 split)
##       Income     < 22.5  to the left,  agree=0.580, adj=0.021, (0 split)
##
## Node number 9: 45 observations,    complexity param=0.04573171
##   predicted class=Yes  expected loss=0.4444444  P(node) =0.1125
##     class counts:    20    25
##    probabilities: 0.444 0.556
##   left son=18 (20 obs) right son=19 (25 obs)
##   Primary splits:
##       Age       < 54.5  to the right, improve=6.722222, (0 missing)
##       CompPrice < 121.5 to the left,  improve=4.629630, (0 missing)
##       ShelveLoc splits as  L-R,       improve=3.250794, (0 missing)
##       Income    < 99.5  to the left,  improve=3.050794, (0 missing)
##       Price     < 127   to the right, improve=2.933429, (0 missing)
##   Surrogate splits:
##       Population  < 363.5 to the left,  agree=0.667, adj=0.25, (0 split)
##       Income      < 39    to the left,  agree=0.644, adj=0.20, (0 split)
##       Advertising < 17.5  to the left,  agree=0.644, adj=0.20, (0 split)
```

```
##       CompPrice   < 106.5 to the left,  agree=0.622, adj=0.15, (0 split)
##       Price       < 135.5 to the right, agree=0.622, adj=0.15, (0 split)
##
## Node number 10: 10 observations
##   predicted class=No   expected loss=0.3  P(node) =0.025
##     class counts:     7     3
##    probabilities: 0.700 0.300
##
## Node number 11: 36 observations
##   predicted class=Yes  expected loss=0.1944444  P(node) =0.09
##     class counts:     7    29
##    probabilities: 0.194 0.806
##
## Node number 16: 96 observations
##   predicted class=No   expected loss=0.0625  P(node) =0.24
##     class counts:    90     6
##    probabilities: 0.938 0.062
##
## Node number 17: 128 observations,    complexity param=0.02743902
##   predicted class=No   expected loss=0.2734375  P(node) =0.32
##     class counts:    93    35
##    probabilities: 0.727 0.273
##   left son=34 (107 obs) right son=35 (21 obs)
##   Primary splits:
##       Price     < 109.5 to the right, improve=9.764582, (0 missing)
##       ShelveLoc splits as  L-R,       improve=6.320022, (0 missing)
##       Age       < 49.5  to the right, improve=2.575061, (0 missing)
##       Income    < 108.5 to the right, improve=1.799546, (0 missing)
##       CompPrice < 143.5 to the left,  improve=1.741982, (0 missing)
##
## Node number 18: 20 observations
##   predicted class=No   expected loss=0.25  P(node) =0.05
##     class counts:    15     5
##    probabilities: 0.750 0.250
##
## Node number 19: 25 observations
##   predicted class=Yes  expected loss=0.2  P(node) =0.0625
##     class counts:     5    20
##    probabilities: 0.200 0.800
##
## Node number 34: 107 observations,    complexity param=0.01219512
##   predicted class=No   expected loss=0.1869159  P(node) =0.2675
##     class counts:    87    20
##    probabilities: 0.813 0.187
##   left son=68 (65 obs) right son=69 (42 obs)
##   Primary splits:
##       Price     < 126.5 to the right, improve=2.9643900, (0 missing)
##       CompPrice < 147.5 to the left,  improve=2.2337090, (0 missing)
##       ShelveLoc splits as  L-R,       improve=2.2125310, (0 missing)
##       Age       < 49.5  to the right, improve=2.1458210, (0 missing)
##       Income    < 60.5  to the left,  improve=0.8025853, (0 missing)
##   Surrogate splits:
##       CompPrice   < 129.5 to the right, agree=0.664, adj=0.143, (0 split)
##       Advertising < 3.5   to the right, agree=0.664, adj=0.143, (0 split)
```

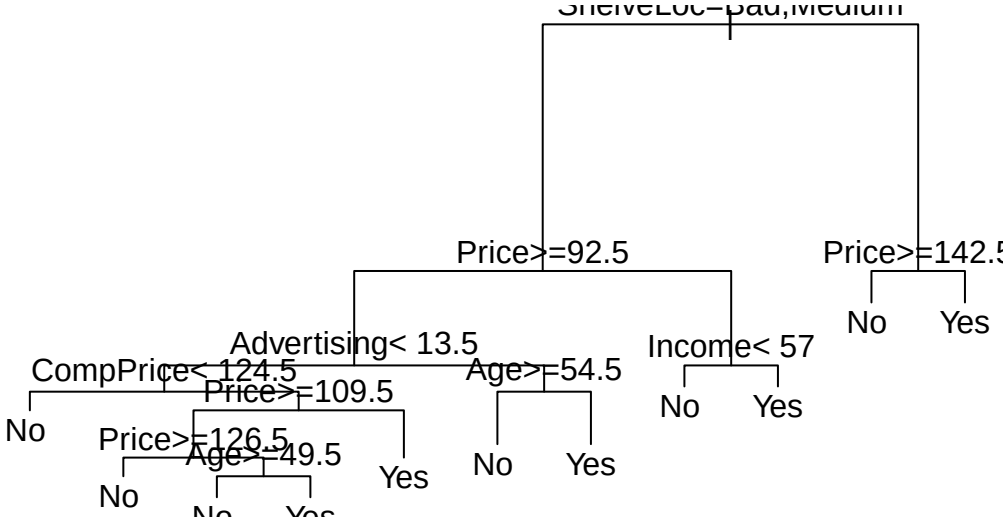
```
##       Population  < 53.5  to the right, agree=0.645, adj=0.095, (0 split)
##       Age         < 77.5  to the left,  agree=0.636, adj=0.071, (0 split)
##
## Node number 35: 21 observations
##   predicted class=Yes  expected loss=0.2857143  P(node) =0.0525
##     class counts:     6    15
##    probabilities: 0.286 0.714
##
## Node number 68: 65 observations
##   predicted class=No   expected loss=0.09230769  P(node) =0.1625
##     class counts:    59     6
##    probabilities: 0.908 0.092
##
## Node number 69: 42 observations,    complexity param=0.01219512
##   predicted class=No   expected loss=0.3333333  P(node) =0.105
##     class counts:    28    14
##    probabilities: 0.667 0.333
##   left son=138 (22 obs) right son=139 (20 obs)
##   Primary splits:
##       Age         < 49.5  to the right, improve=5.4303030, (0 missing)
##       CompPrice   < 137.5 to the left,  improve=2.1000000, (0 missing)
##       Advertising < 5.5   to the left,  improve=1.8666670, (0 missing)
##       ShelveLoc   splits as  L-R,       improve=1.4291670, (0 missing)
##       Population  < 382   to the right, improve=0.8578431, (0 missing)
##   Surrogate splits:
##       Income      < 46.5  to the left,  agree=0.595, adj=0.15, (0 split)
##       CompPrice   < 131.5 to the right, agree=0.571, adj=0.10, (0 split)
##       Advertising < 5.5   to the left,  agree=0.571, adj=0.10, (0 split)
##       Population  < 221.5 to the left,  agree=0.571, adj=0.10, (0 split)
##       Price       < 116.5 to the left,  agree=0.571, adj=0.10, (0 split)
##
## Node number 138: 22 observations
##   predicted class=No   expected loss=0.09090909  P(node) =0.055
##     class counts:    20     2
##    probabilities: 0.909 0.091
##
## Node number 139: 20 observations
##   predicted class=Yes  expected loss=0.4  P(node) =0.05
##     class counts:     8    12
##    probabilities: 0.400 0.600
```

```
fancyRpartPlot(ModelC)
```



Rattle 2025−Mar−31 00:14:25 muralishanker

```
plot(ModelC)
text(ModelC, pretty = 0)
```



Let's apply this approach to a hold-out set

```
set.seed(12)
training = sample(1:nrow(MyData),200)
MyData.train = MyData[training,]
MyData.test <- MyData[-training,]
ModelCa = rpart(High~.-Sales,data=MyData,subset=training,method="class")
summary(ModelCa)
```

```
## Call:
## rpart(formula = High ~ . - Sales, data = MyData, subset = training,
##     method = "class")
##   n= 200
##
##           CP nsplit rel error    xerror       xstd
## 1 0.28571429      0 1.0000000 1.0000000 0.08309490
## 2 0.17857143      1 0.7142857 0.8095238 0.07975302
## 3 0.07142857      2 0.5357143 0.5714286 0.07190319
## 4 0.02380952      4 0.3928571 0.4642857 0.06670386
## 5 0.01000000      6 0.3452381 0.4761905 0.06734350
##
## Variable importance
##       Price Advertising   ShelveLoc   CompPrice         Age      Income
##          38          27          21          10           2           1
##  Population
##           1
```

```
##
## Node number 1: 200 observations,    complexity param=0.2857143
##   predicted class=No   expected loss=0.42  P(node) =1
##     class counts:   116    84
##    probabilities: 0.580 0.420
##   left son=2 (168 obs) right son=3 (32 obs)
##   Primary splits:
##       Price       < 92.5  to the right, improve=15.773330, (0 missing)
##       ShelveLoc   splits as  LRL,       improve=15.054580, (0 missing)
##       Advertising < 8.5   to the left,  improve= 9.881667, (0 missing)
##       Age         < 61.5  to the right, improve= 5.230355, (0 missing)
##       Income      < 25.5  to the left,  improve= 1.798348, (0 missing)
##   Surrogate splits:
##       CompPrice < 103.5 to the right, agree=0.85, adj=0.063, (0 split)
##
## Node number 2: 168 observations,    complexity param=0.1785714
##   predicted class=No   expected loss=0.3333333  P(node) =0.84
##     class counts:   112    56
##    probabilities: 0.667 0.333
##   left son=4 (135 obs) right son=5 (33 obs)
##   Primary splits:
##       ShelveLoc   splits as  LRL,       improve=12.746130, (0 missing)
##       Advertising < 8.5   to the left,  improve= 9.782531, (0 missing)
##       Age         < 61.5  to the right, improve= 5.710132, (0 missing)
##       CompPrice   < 131.5 to the left,  improve= 3.984190, (0 missing)
##       Price       < 109.5 to the right, improve= 1.619887, (0 missing)
##
## Node number 3: 32 observations
##   predicted class=Yes  expected loss=0.125  P(node) =0.16
##     class counts:     4    28
##    probabilities: 0.125 0.875
##
## Node number 4: 135 observations,    complexity param=0.07142857
##   predicted class=No   expected loss=0.237037  P(node) =0.675
##     class counts:   103    32
##    probabilities: 0.763 0.237
##   left son=8 (117 obs) right son=9 (18 obs)
##   Primary splits:
##       Advertising < 15.5  to the left,  improve=7.667236, (0 missing)
##       CompPrice   < 128.5 to the left,  improve=4.632447, (0 missing)
##       Age         < 49.5  to the right, improve=3.890993, (0 missing)
##       ShelveLoc   splits as  L-R,       improve=2.984893, (0 missing)
##       Population  < 57.5  to the left,  improve=1.345759, (0 missing)
##
## Node number 5: 33 observations,    complexity param=0.07142857
##   predicted class=Yes  expected loss=0.2727273  P(node) =0.165
##     class counts:     9    24
##    probabilities: 0.273 0.727
##   left son=10 (8 obs) right son=11 (25 obs)
##   Primary splits:
##       Advertising < 1     to the left,  improve=7.6609090, (0 missing)
##       Price       < 127   to the right, improve=2.7453210, (0 missing)
##       Income      < 43    to the left,  improve=2.6209090, (0 missing)
##       Age         < 68.5  to the right, improve=2.6209090, (0 missing)
```

```
##       Population  < 165   to the right, improve=0.6464646, (0 missing)
##   Surrogate splits:
##       Price < 142.5 to the right, agree=0.879, adj=0.5, (0 split)
##
## Node number 8: 117 observations,    complexity param=0.02380952
##   predicted class=No   expected loss=0.1709402  P(node) =0.585
##     class counts:    97    20
##    probabilities: 0.829 0.171
##   left son=16 (94 obs) right son=17 (23 obs)
##   Primary splits:
##       CompPrice   < 137.5 to the left,  improve=3.9857050, (0 missing)
##       Age         < 33.5  to the right, improve=2.2576310, (0 missing)
##       ShelveLoc   splits as  L-R,       improve=1.4786090, (0 missing)
##       Price       < 109.5 to the right, improve=0.7421841, (0 missing)
##       Advertising < 8.5   to the left,  improve=0.7394047, (0 missing)
##   Surrogate splits:
##       Price < 158   to the left,  agree=0.829, adj=0.13, (0 split)
##
## Node number 9: 18 observations
##   predicted class=Yes  expected loss=0.3333333  P(node) =0.09
##     class counts:     6    12
##    probabilities: 0.333 0.667
##
## Node number 10: 8 observations
##   predicted class=No   expected loss=0.125  P(node) =0.04
##     class counts:     7     1
##    probabilities: 0.875 0.125
##
## Node number 11: 25 observations
##   predicted class=Yes  expected loss=0.08  P(node) =0.125
##     class counts:     2    23
##    probabilities: 0.080 0.920
##
## Node number 16: 94 observations
##   predicted class=No   expected loss=0.106383  P(node) =0.47
##     class counts:    84    10
##    probabilities: 0.894 0.106
##
## Node number 17: 23 observations,    complexity param=0.02380952
##   predicted class=No   expected loss=0.4347826  P(node) =0.115
##     class counts:    13    10
##    probabilities: 0.565 0.435
##   left son=34 (13 obs) right son=35 (10 obs)
##   Primary splits:
##       Price       < 130.5 to the right, improve=2.4889630, (0 missing)
##       Age         < 50.5  to the right, improve=2.4889630, (0 missing)
##       Income      < 59.5  to the left,  improve=1.1073780, (0 missing)
##       Advertising < 6     to the left,  improve=0.8876812, (0 missing)
##       CompPrice   < 146.5 to the right, improve=0.6428094, (0 missing)
##   Surrogate splits:
##       CompPrice   < 146.5 to the right, agree=0.783, adj=0.5, (0 split)
##       Age         < 60    to the right, agree=0.739, adj=0.4, (0 split)
##       Income      < 35.5  to the right, agree=0.696, adj=0.3, (0 split)
##       Advertising < 9.5   to the left,  agree=0.696, adj=0.3, (0 split)
```

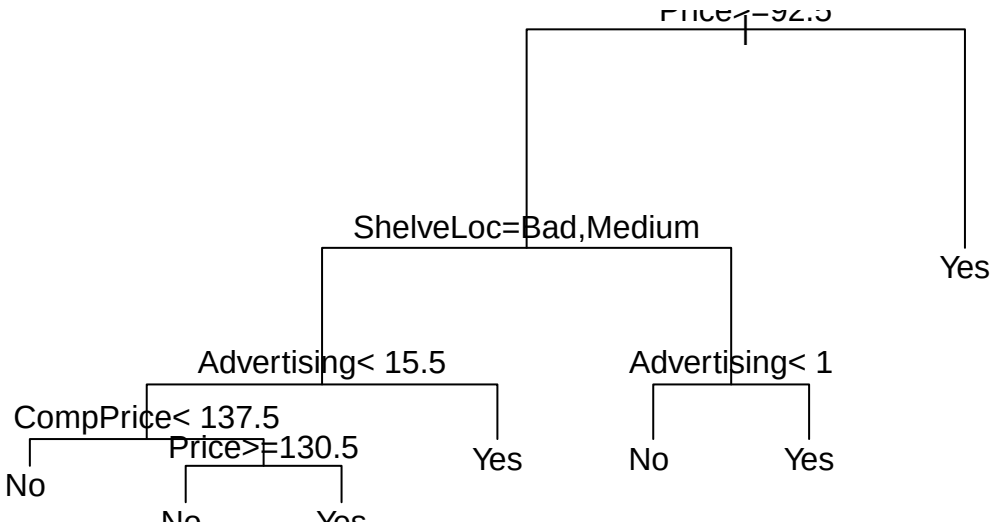
```
##       Population  < 97.5  to the right, agree=0.652, adj=0.2, (0 split)
##
## Node number 34: 13 observations
##   predicted class=No   expected loss=0.2307692  P(node) =0.065
##     class counts:    10     3
##    probabilities: 0.769 0.231
##
## Node number 35: 10 observations
##   predicted class=Yes  expected loss=0.3  P(node) =0.05
##     class counts:     3     7
##    probabilities: 0.300 0.700
```

```
ModelCa.predict = predict(ModelCa, MyData.test,type="class")
table(ModelCa.predict,MyData.test$High)
```

```
##
## ModelCa.predict No Yes
##             No  95  29
##             Yes 25  51
```

Let's prune the tree based on cp value

```
ModelCa.prune <- prune.rpart(ModelCa,cp=0.02380952)
plot(ModelCa.prune)
text(ModelCa.prune,pretty = 0)
```



```
ModelCa.prune.predict <- predict(ModelCa.prune,MyData.test,type="class")
table(ModelCa.prune.predict,MyData.test$High)
```

```
##
## ModelCa.prune.predict No Yes
##                    No  95  29
##                    Yes 25  51
```

Using the Gini Index

```
gini  <- function(tree){
  # calculate gini index for `rpart` tree
  ylevels <- attributes(tree)[["ylevels"]]
  nclass <- length(ylevels)
  yval2 <- tree[["frame"]][["yval2"]]
  vars <- tree[["frame"]][["var"]]
  labls = labels(tree)
  df = data.frame(matrix(nrow=length(labls), ncol=5))
  colnames(df) <- c("Name", "GiniIndex", "Class", "Items", "ItemProbs")

  for(i in 1:length(vars)){
    row <- yval2[i , ]
    node.class <- row[1]
    j <- 2
    node.class_counts = row[j:(j+nclass-1)]
    j <- j+nclass
    node.class_probs = row[j:(j+nclass-1)]

    gini = 1-sum(node.class_probs^2)
    gini = round(gini,5)
    name = paste(vars[i], " (", labls[i], ")")
    df[i,] = c(name, gini, node.class, toString(round(node.class_counts,5)), toString(round(node.class_
  }
  return(df)
}
gini(ModelCa)
```

```
##                              Name GiniIndex Class   Items        ItemProbs
## 1                   Price  ( root )    0.4872     1 116, 84       0.58, 0.42
## 2        ShelveLoc  ( Price>=92.5 )   0.44444     1 112, 56 0.66667, 0.33333
## 3     Advertising  ( ShelveLoc=ac )    0.3617     1 103, 32 0.76296, 0.23704
## 4  CompPrice  ( Advertising< 15.5 )   0.28344     1  97, 20 0.82906, 0.17094
## 5      <leaf>  ( CompPrice< 137.5 )   0.19013     1  84, 10 0.89362, 0.10638
## 6       Price  ( CompPrice>=137.5 )   0.49149     1  13, 10 0.56522, 0.43478
## 7          <leaf>  ( Price>=130.5 )   0.35503     1   10, 3 0.76923, 0.23077
## 8          <leaf>  ( Price< 130.5 )      0.42     2    3, 7         0.3, 0.7
## 9     <leaf>  ( Advertising>=15.5 )   0.44444     2   6, 12 0.33333, 0.66667
## 10     Advertising  ( ShelveLoc=b )   0.39669     2   9, 24 0.27273, 0.72727
## 11       <leaf>  ( Advertising< 1 )   0.21875     1    7, 1     0.875, 0.125
## 12       <leaf>  ( Advertising>=1 )    0.1472     2   2, 23       0.08, 0.92
## 13         <leaf>  ( Price< 92.5 )   0.21875     2   4, 28     0.125, 0.875
```