Tree_learning

Neha

09/02/2020

```
require(ISLR)
```

```
## Loading required package: ISLR
```

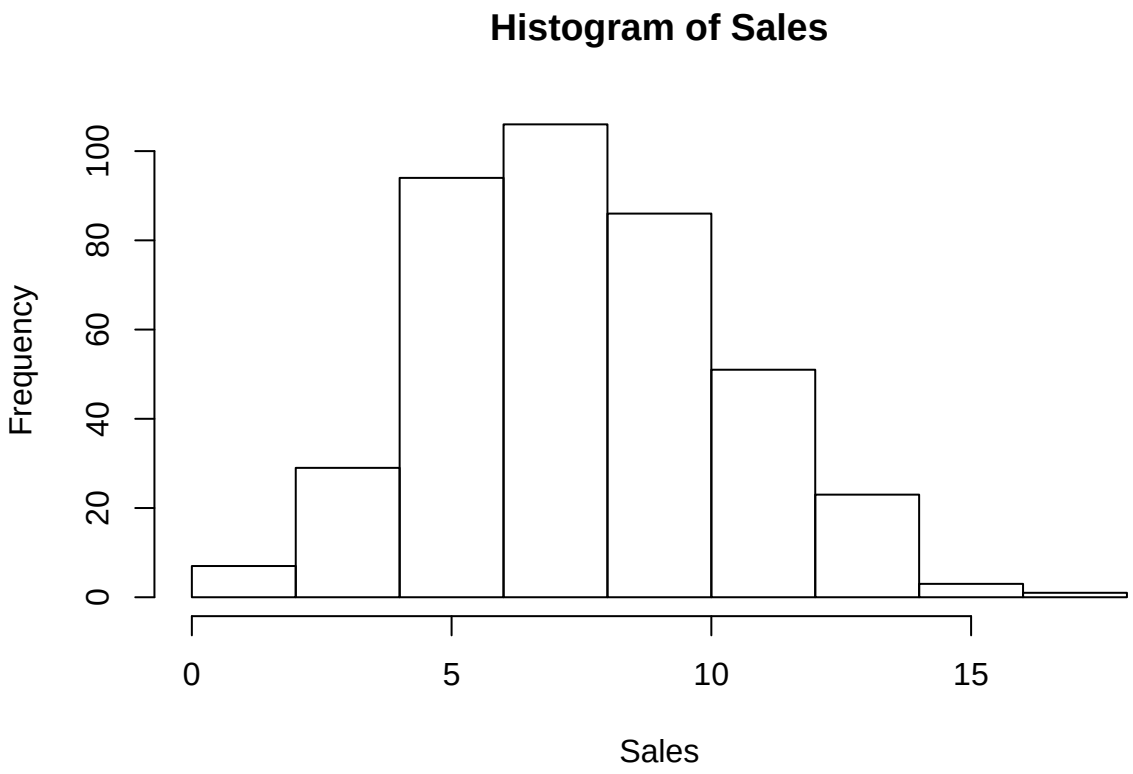
```
require(tree)
```

```
## Loading required package: tree
```

```
attach(Carseats)
summary(Carseats)
```

```
##      Sales          CompPrice       Income        Advertising
##  Min.   : 0.000   Min.   : 77   Min.   : 21.00   Min.   : 0.000
##  1st Qu.: 5.390   1st Qu.:115   1st Qu.: 42.75   1st Qu.: 0.000
##  Median : 7.490   Median :125   Median : 69.00   Median : 5.000
##  Mean   : 7.496   Mean   :125   Mean   : 68.66   Mean   : 6.635
##  3rd Qu.: 9.320   3rd Qu.:135   3rd Qu.: 91.00   3rd Qu.:12.000
##  Max.   :16.270   Max.   :175   Max.   :120.00   Max.   :29.000
##    Population        Price        ShelveLoc        Age
##  Min.   : 10.0   Min.   : 24.0   Bad   : 96   Min.   :25.00
##  1st Qu.:139.0   1st Qu.:100.0   Good  : 85   1st Qu.:39.75
##  Median :272.0   Median :117.0   Medium:219   Median :54.50
##  Mean   :264.8   Mean   :115.8                Mean   :53.32
##  3rd Qu.:398.5   3rd Qu.:131.0                3rd Qu.:66.00
##  Max.   :509.0   Max.   :191.0                Max.   :80.00
##    Education    Urban       US
##  Min.   :10.0   No :118   No :142
##  1st Qu.:12.0   Yes:282   Yes:258
##  Median :14.0
##  Mean   :13.9
##  3rd Qu.:16.0
##  Max.   :18.0
```

```
# A histogram of the sales data column
hist(Sales)
```

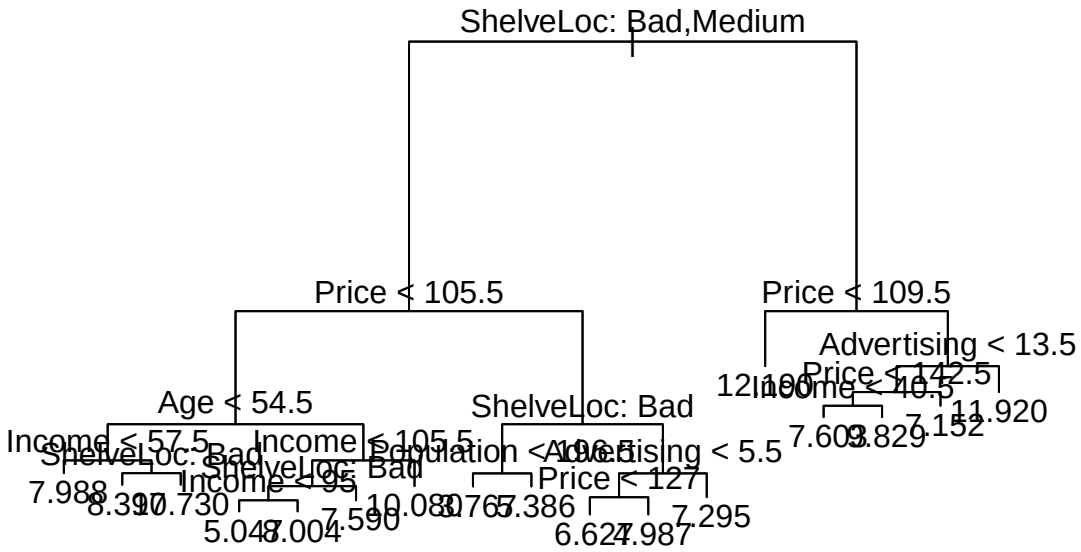



```
# Make this binary based on distribution
High=ifelse(Sales<=8,"No","Yes")
#Rewrite into dataframe
Carseats=data.frame(Carseats, High)
```

```
#fit a tree to the data
#Exclude Sales --- from the right-hand side of the formula, because the response is derived from it.
tree.carseats=tree(Sales~.-High,data=Carseats)
#summarize the tree data
summary(tree.carseats)
```

```
##
## Regression tree:
## tree(formula = Sales ~ . - High, data = Carseats)
## Variables actually used in tree construction:
## [1] "ShelveLoc"   "Price"       "Age"         "Income"      "Population"
## [6] "Advertising"
## Number of terminal nodes:  17
## Residual mean deviance:  2.878 = 1102 / 383
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -4.98700 -1.23000 -0.06125  0.00000  1.22500  4.75400
```

```
plot(tree.carseats)
#add text to tree splits
text(tree.carseats,pretty=0)
```



For a detailed summary of the tree, print it:

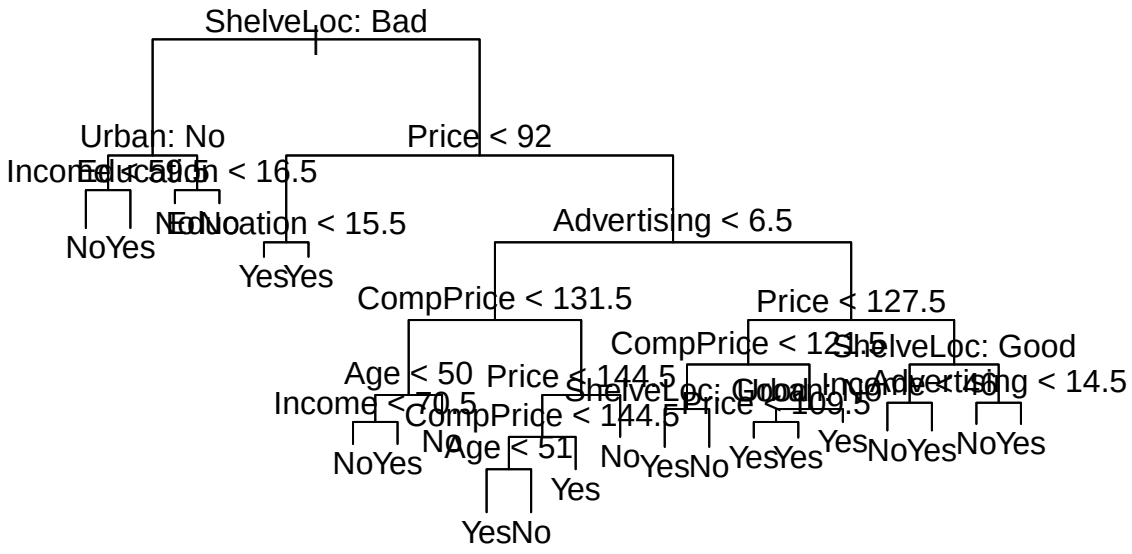
```
tree.carseats
```

```
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 400 3182.0000  7.496
##    2) ShelveLoc: Bad,Medium 315 1860.0000  6.763
##      4) Price < 105.5 108  568.6000  8.189
##        8) Age < 54.5 43  158.7000  9.413
##         16) Income < 57.5 13   19.2400  7.988 *
##         17) Income > 57.5 30  101.6000 10.030
##           34) ShelveLoc: Bad 9   22.7600  8.397 *
##           35) ShelveLoc: Medium 21   44.5300 10.730 *
##        9) Age > 54.5 65  303.1000  7.380
##         18) Income < 105.5 56  203.0000  6.946
##           36) ShelveLoc: Bad 20   76.9600  5.786
##             72) Income < 95 15   43.8100  5.047 *
##             73) Income > 95 5    0.3687  8.004 *
##           37) ShelveLoc: Medium 36   84.2400  7.590 *
##         19) Income > 105.5 9   23.8200 10.080 *
```

```
##      5) Price > 105.5 207  956.6000  6.019
##       10) ShelveLoc: Bad 61  240.8000  4.722
##         20) Population < 196.5 25   88.2300  3.767 *
##         21) Population > 196.5 36  113.9000  5.386 *
##       11) ShelveLoc: Medium 146  570.4000  6.560
##         22) Advertising < 5.5 77  280.1000  5.902
##           44) Price < 127 43   95.5200  6.627 *
##           45) Price > 127 34  133.5000  4.987 *
##         23) Advertising > 5.5 69  219.8000  7.295 *
##    3) ShelveLoc: Good 85  525.5000 10.210
##      6) Price < 109.5 28   85.5800 12.190 *
##      7) Price > 109.5 57  277.3000  9.244
##       14) Advertising < 13.5 48  185.4000  8.743
##         28) Price < 142.5 36  108.3000  9.272
##           56) Income < 40.5 9    9.8280  7.603 *
##           57) Income > 40.5 27   65.0600  9.829 *
##         29) Price > 142.5 12   36.6500  7.152 *
##       15) Advertising > 13.5 9   15.2700 11.920 *
```

create a training and test set (250,150) split of the 400 observations, grow the tree on the training set, and evaluate its performance on the test set.

```
set.seed(1001)
#Take 250 training samples
train=sample(1:nrow(Carseats),250)
#train a tree model
tree.carseats=tree(High~.-Sales,Carseats,subset=train)
#Plot the tree
plot(tree.carseats);text(tree.carseats,pretty=0)
```




```
#Predict on remaining data
tree.pred=predict(tree.carseats,Carseats[-train,],type="class")
with(Carseats[-train,],table(tree.pred,High))
```

```
##          High
## tree.pred No Yes
##       No  71  22
##       Yes 18  39
```

```
#final accuracy
(72+33)/150
```

```
## [1] 0.7
```

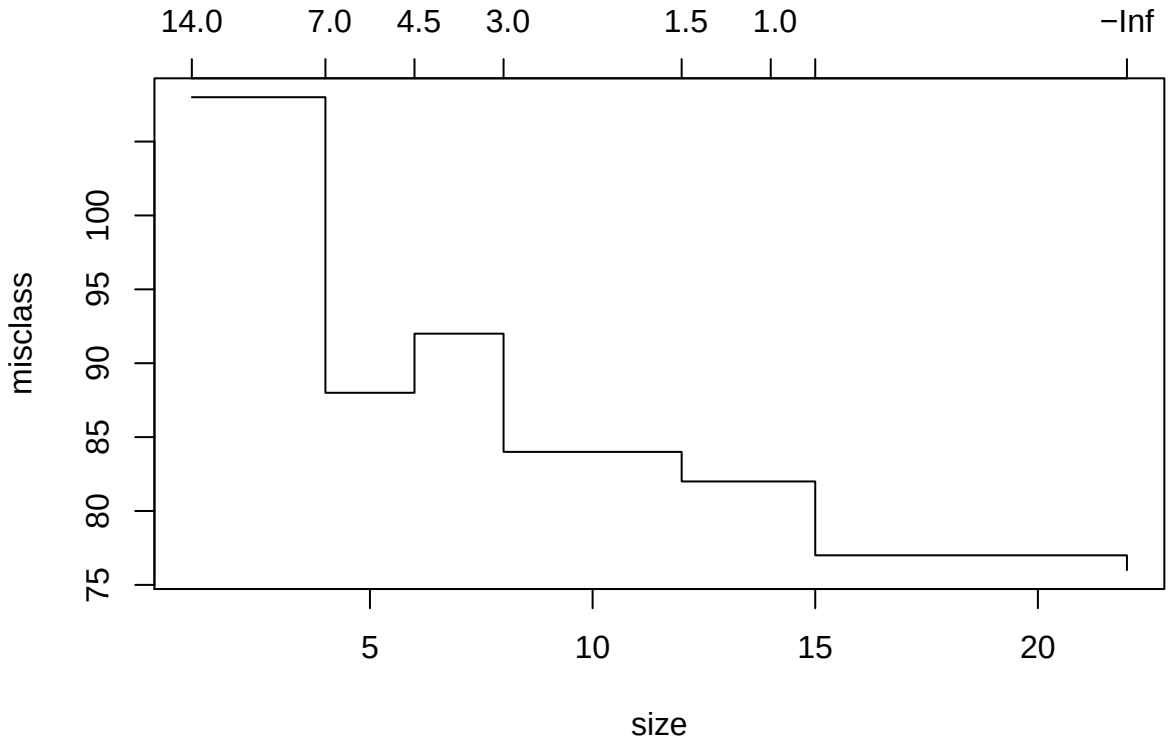
PRUNING TREE This tree was grown to full depth, and might be too variable. We now use CV to prune it.

```
#using the cross validation method
cv.carseats=cv.tree(tree.carseats,FUN=prune.misclass)
cv.carseats
```

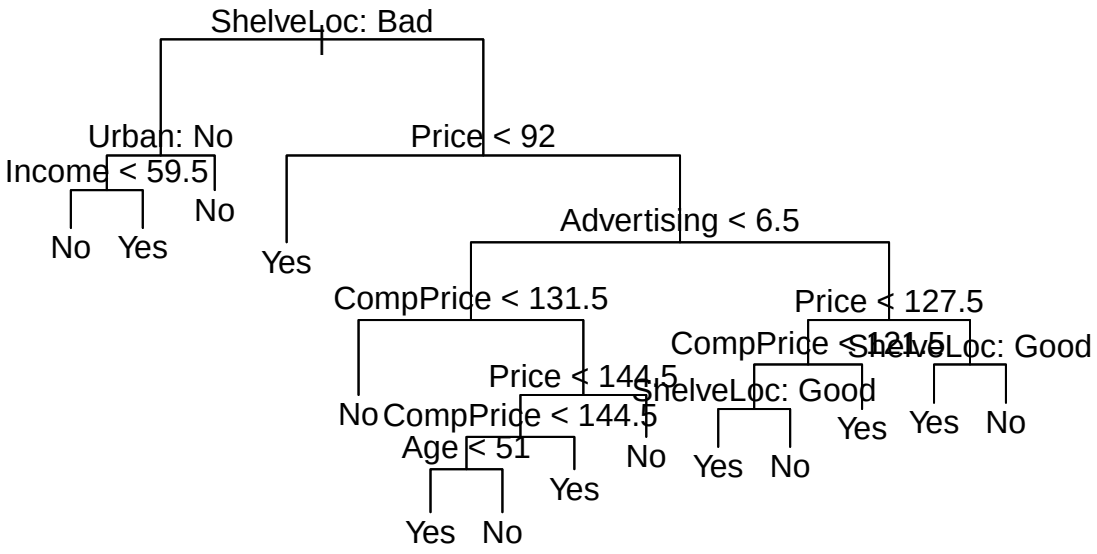
```
## $size
## [1] 22 15 14 12  8  6  4  1
##
```

```
## $dev
## [1]  76  77  82  82  84  92  88 108
##
## $k
## [1]     -Inf  0.00000  1.00000  1.50000  3.00000  4.50000  7.00000 13.66667
##
## $method
## [1] "misclass"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```

```
plot(cv.carseats)
```




```
#Missclassification errors
prune.carseats=prune.misclass(tree.carseats,best=13)
plot(prune.carseats)
text(prune.carseats,pretty=0)
```



Missclassification is jumpy because there are many cross validation points. To prune optimally, pick a middle value from graph, low error. (Like 13) So the printed tree now has size 13 and is much easier to read.

We can now test how the pruned tree does on the test data.

```
tree.pred=predict(prune.carseats,Carseats[-train,],type="class")
with(Carseats[-train,],table(tree.pred,High))
```

```
##          High
## tree.pred No Yes
##       No  74  23
##       Yes 15  38
```

```
(72+32)/150
```

```
## [1] 0.6933333
```

It has done about the same as our original tree. So pruning did not hurt us wrt misclassification errors, and gave us a simpler tree.