Introduction to classification and regression trees, random forests and model-based recursive partitioning in R

Day 1: Single trees

Exercise 1: Impurity measures

We have a root node, in which a proportion of .7999 of the observations belong to class 1. One of the splitting candidates is a categorical variable with two levels. If this variable would be used to split the observations, a proportion of .565 would go left, and a proportion of .435 would go right. The proportion of class 1 observations in the left node would be 1.0. The proportion of class 1 observations in the right node would be .54.

Would this split improve purity according to the Gini index? According to the Shannon entropy? According to the classification error?

```
# According to the classification error:
##
## rootnode:
1 - .799
```

```
## [1] 0.201
```

```
##
## left daughter node:
## 0
##
## right daughter node:
1 - .54
```

```
## [1] 0.46
```

```
##
## weighted classification error in both nodes:
(.565*0) + (.435*.46)
```

```
## [1] 0.2001
```

```
##
# The impuritu increases.
##
##
##
# According to the Gini index:
##
## rootnode:
.7999 * .2001 + .2001 * .7999
```

```
## [1] 0.32012
```

```
##
## left daughter node:
## 0
##
## right daughter node:
.54 * .46 + .46 * .54
```

```
## [1] 0.4968
```

```
##
## weigthed gini index in both nodes:
.565 * 0 + .435 * .4968
```

```
## [1] 0.216108
```

```
##
# The impurity decreases.
##
##
##
# According to the Shannon entropy:
##
## rootnode:
- (.7999 * log(.7999) + .2001 * log(.20011))
```

```
## [1] 0.500531
```

```
##
## left daughternode:
##
## 0 # note that if we were to perform the actual calculations, log(0) is not
## defined, so should be replaced by e.g., log(1e-99)
##
## right daughternode:
- (.54 * log(.54) + .46 * log(.46))
```

```
## [1] 0.6899438
```

```
##
## weighted shannon entropy in both nodes:
## .565*0 + .435*.6899
##
# The impurity decreases.
##
##
##
# According to misclassification error, the impurity would not be reduced and
# the split would not be made.
#
# According to the Gini index and Shannon entropy, the impurite would be
# reduced and the split would be made.
```

Exercise 2: Predicting carseat sales

The Carseats dataset from the package ISLR contains data of child car seat sales in 400 different stores:

```
data("Carseats", package = "ISLR")
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
```

```
?ISLR::Carseats
```

```
## starting httpd help server ...
```

```
##  done
```

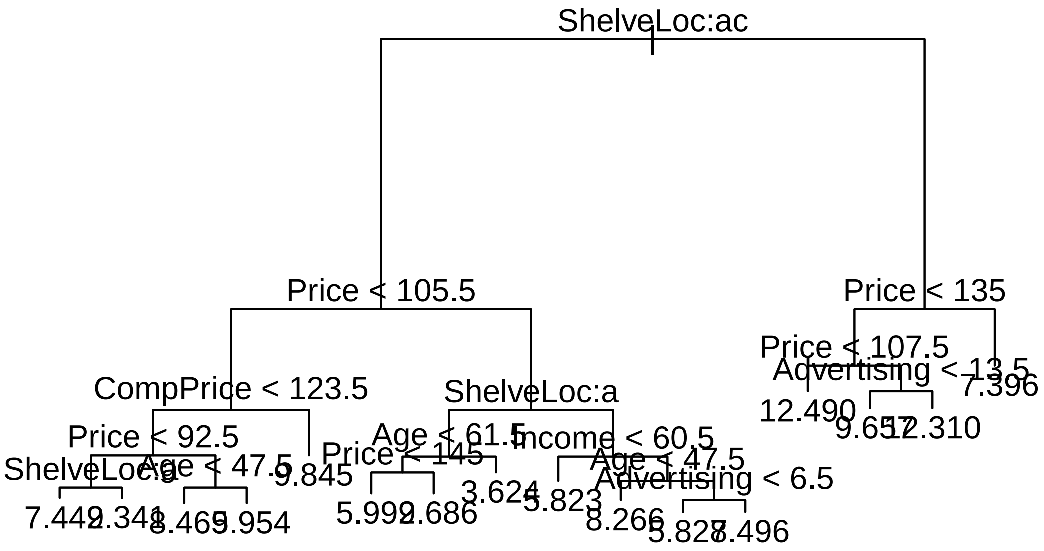
Our aim is to predict car seat sales using the other ten predictor variables in the dataset. Note that we do not need to specify all ten predictor variables, `Sales ~ .` would instruct R to regress Sales on all the remaining variables in the dataset.

a) Randomly separate the dataset in 300 training and 100 test observations (don't forget to set the random seed!)

```
set.seed(42)
train <- sample(1:400, 300)
```

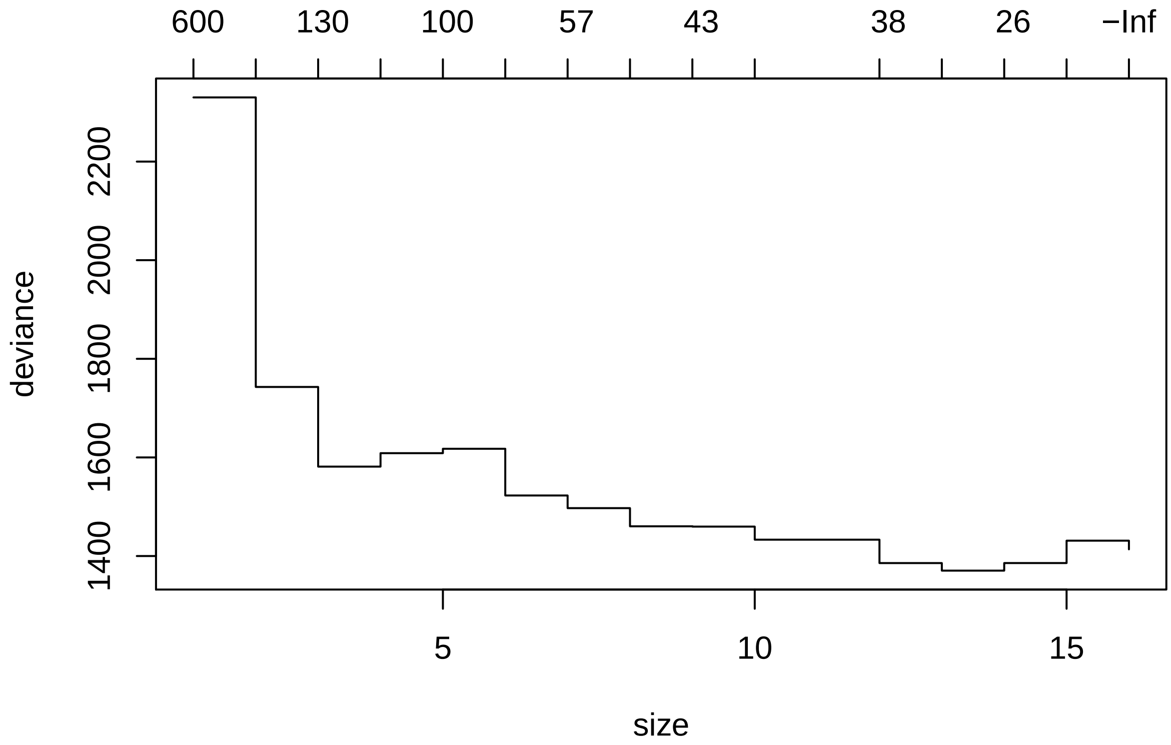
b) Fit and plot a CART tree using the `tree()` function.

```
library(tree)
CART.cs <- tree(Sales ~ ., data = Carseats[train,])
plot(CART.cs)
text(CART.cs)
```

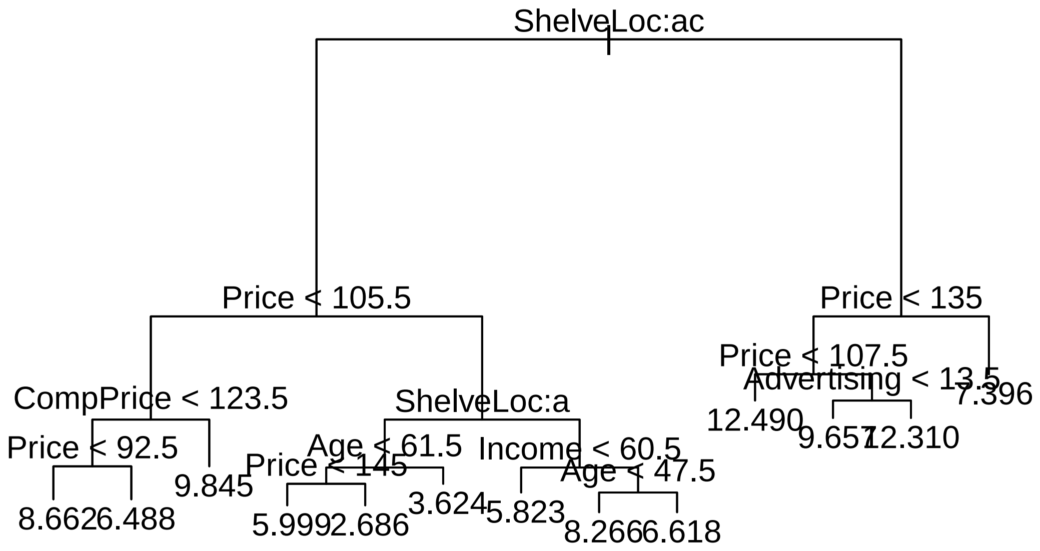


c) Obtain the optimal complexity parameter value using the `cv.tree()` function and prune the tree using the `prune.tree()` function.

```
CART.cs.cv <- cv.tree(CART.cs)
plot(CART.cs.cv)
```



```
CART.cs.pruned <- prune.tree(CART.cs, best = 13)
plot(CART.cs.pruned)
text(CART.cs.pruned)
```



d) What seems to be the most important predictor of carseat sales?

```
# Shelve location is the most important predictor (followed by Price)
```

e) Using the `predict()` function, generate predictions for the test observations. For both the pruned and the unpruned CART tree, calculate the correlation with the true car seat sales in the test data. Did pruning the CART tree indeed improve predictive accuracy?

```
CART_preds_pruned <- predict(CART.cs.pruned, newdata =  Carseats[-train,])
CART_preds <- predict(CART.cs, newdata =  Carseats[-train,])
cor(cbind(Carseats[-train, "Sales"], CART_preds, CART_preds_pruned))
```

```
##                             CART_preds CART_preds_pruned
##                   1.0000000  0.7169186         0.6830204
## CART_preds        0.7169186  1.0000000         0.9762475
## CART_preds_pruned 0.6830204  0.9762475         1.0000000
```

```
# Pruning reduced the accuracy on test data (in this case, should not always be the case).
```