Regression trees and Classification trees

Riet van Bork

Tree-based methods

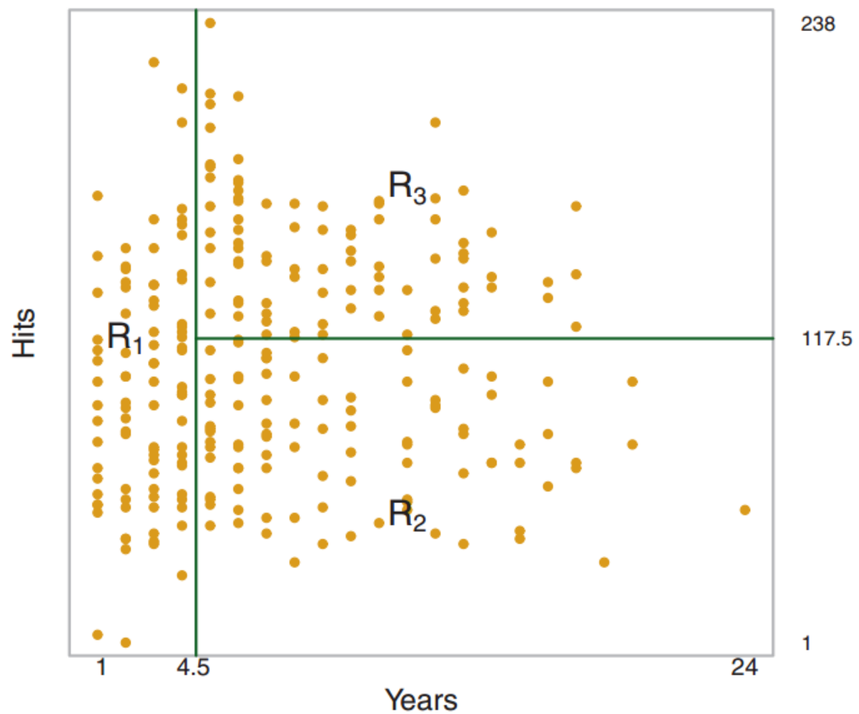
Segmenting the predictor space

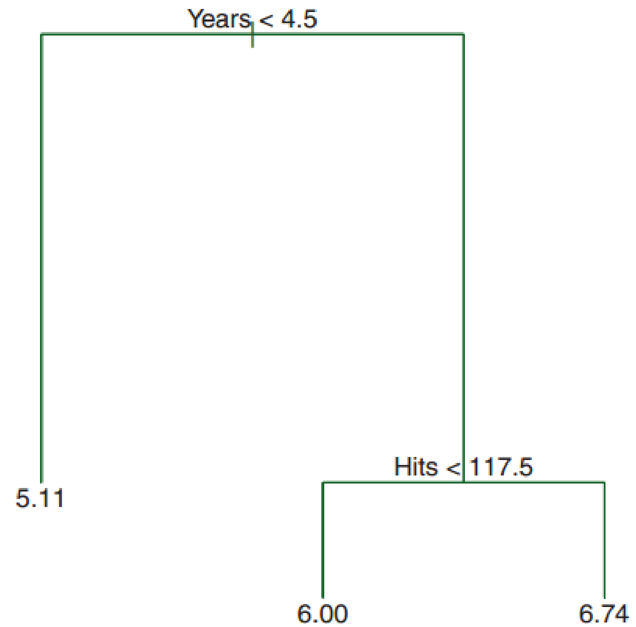
+ Simple and useful for interpretation

- not competitive with best supervised learning approaches

Can be applied to both regression and classification problems.

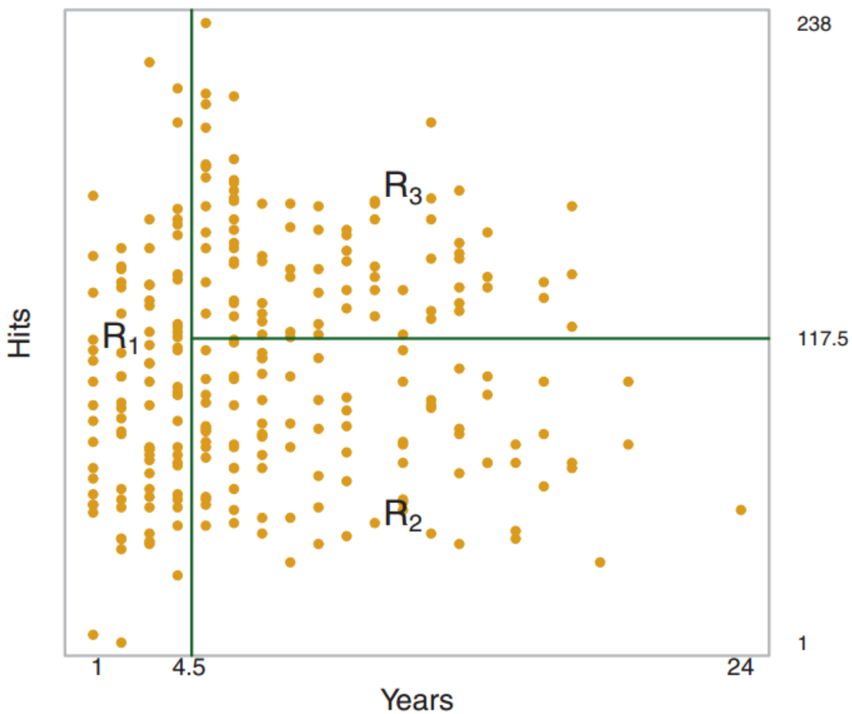
Example

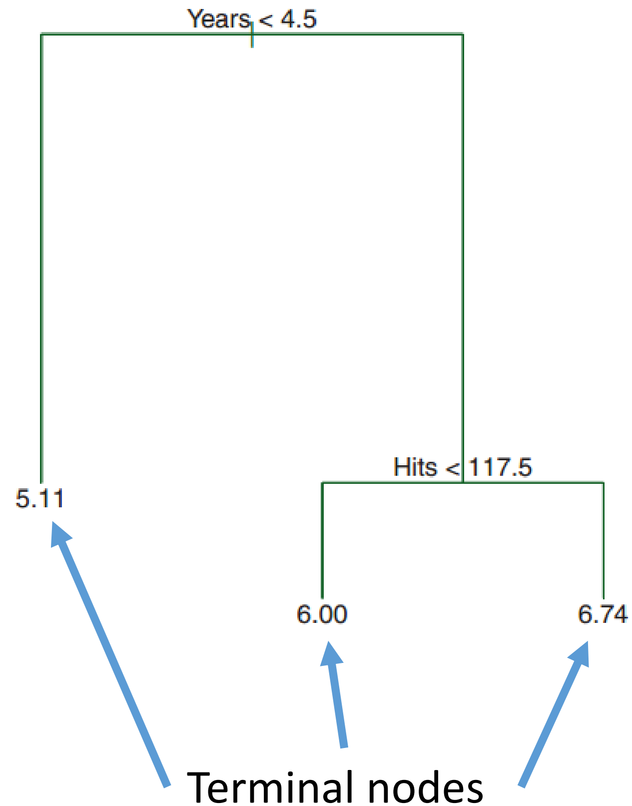




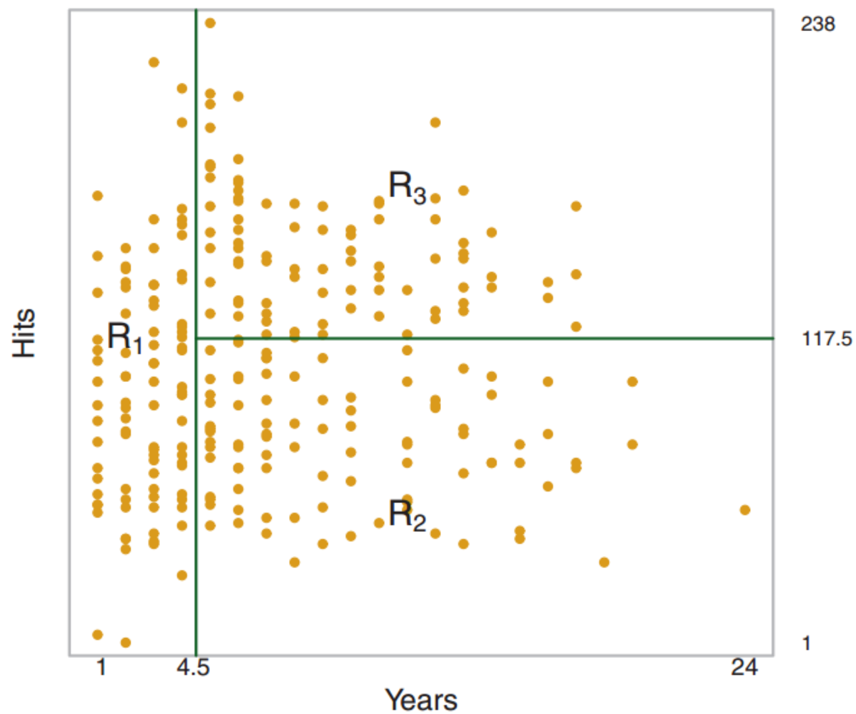
The `Hitters` data: predicting log salary based on number of years that player has played in major league and the number of hits this person made in the previous year.

Example

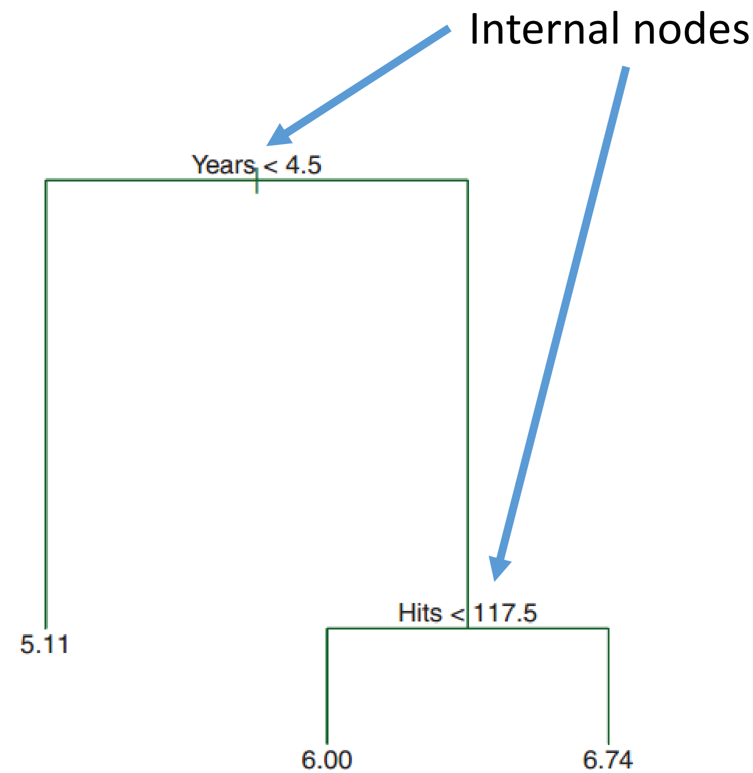

238
117.5
1
R1
R3
R2
Hits
1
4.5
24
Years


Years < 4.5
Hits < 117.5
5.11
6.00
6.74
Terminal nodes

Example

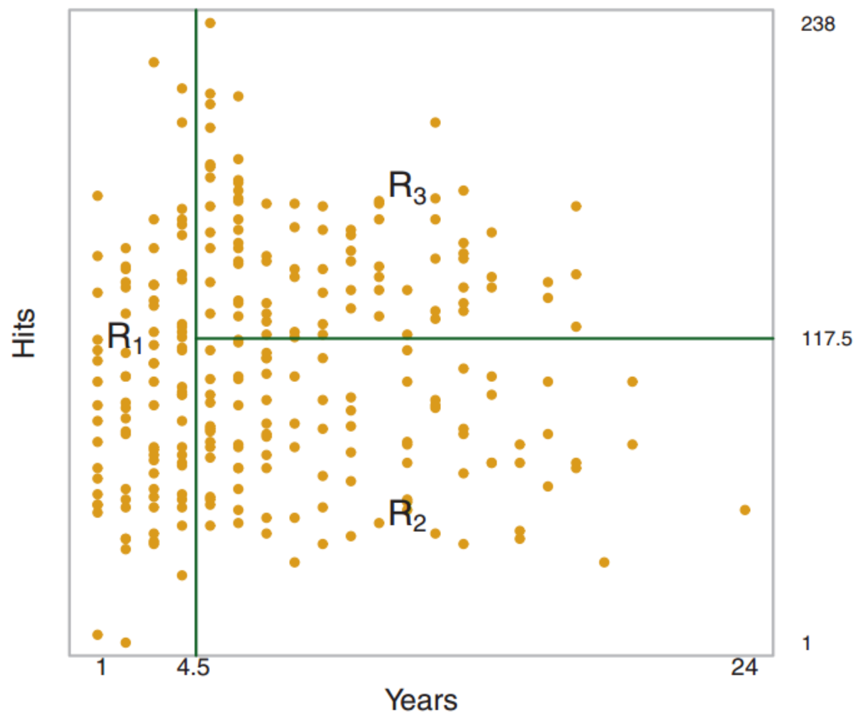


Hits
Years
1
4.5
24
238
117.5
1
R_1
R_2
R_3

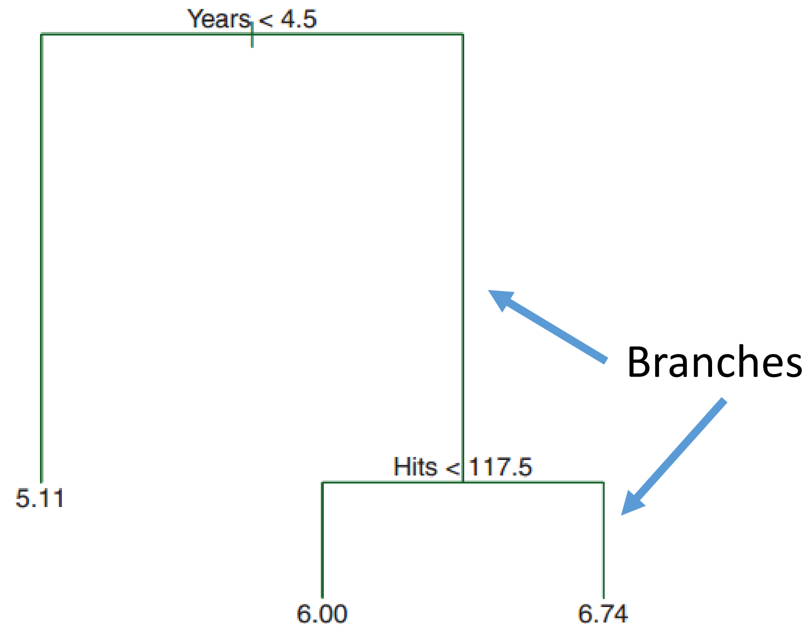


Internal nodes
Years < 4.5
Hits < 117.5
5.11
6.00
6.74

Example



Hits
Years
R1
R2
R3
238
117.5
1
1
4.5
24



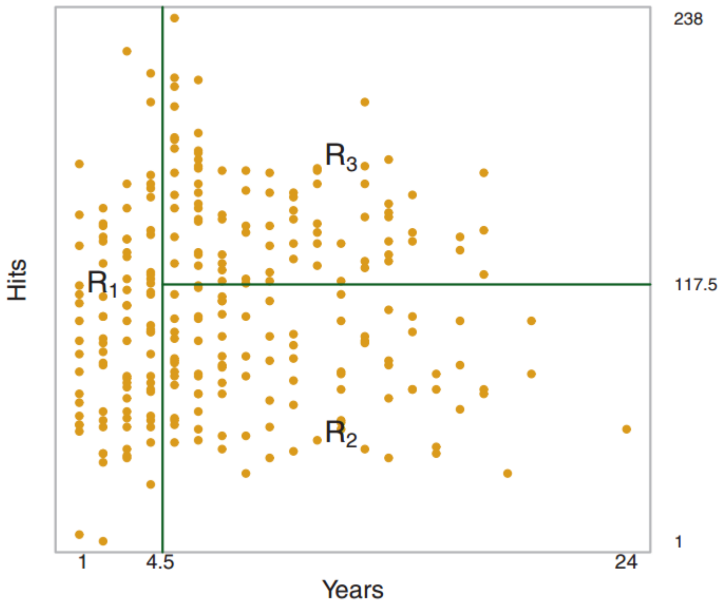
Years < 4.5
Hits < 117.5
5.11
6.00
6.74
Branches

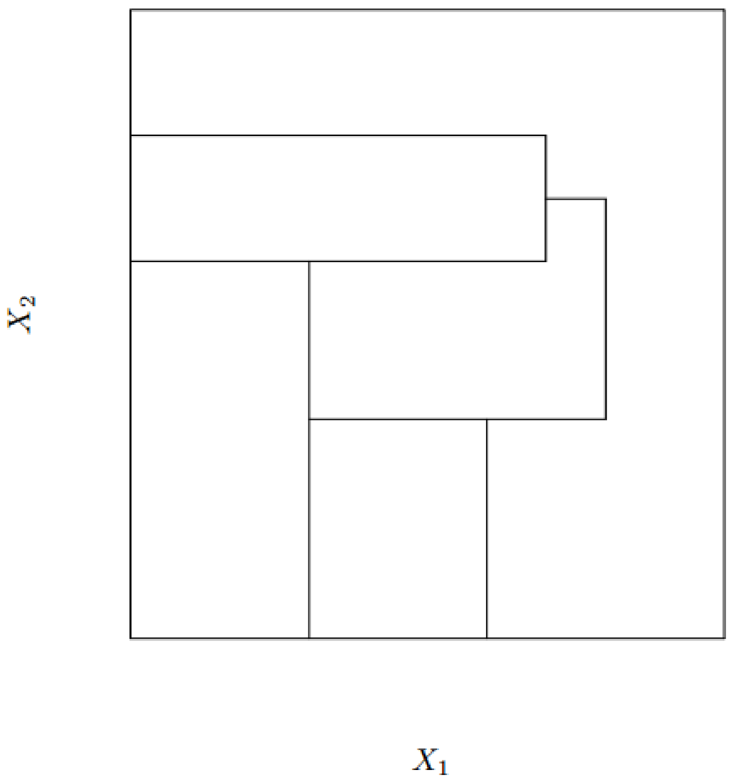
How to build a tree?

1) Divide the predictor space (i.e., the set of possible values for X_1, X_2, .., X_p) into *J* distinct non-overlapping regions, R_1, R_2, .., R_p.

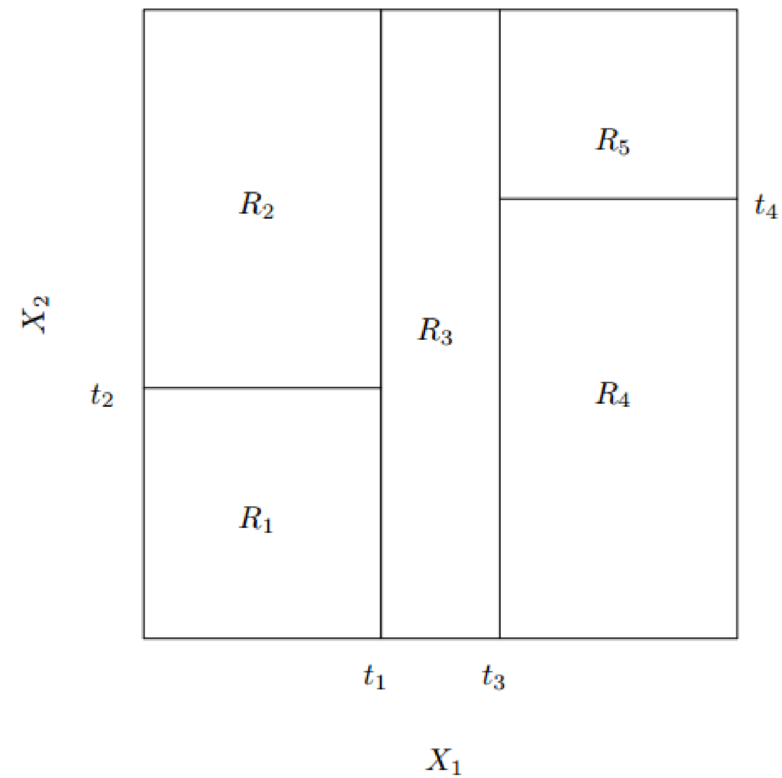
2) Every observation that falls into region R_j gets the same prediction: the mean of the response values for the training observations in R_j.

3) The regions are chosen such that it minimizes: $\sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2$





X_2
X_1



R_5
R_2
t_4
X_2
R_3
t_2
R_4
R_1
t_1
t_3
X_1

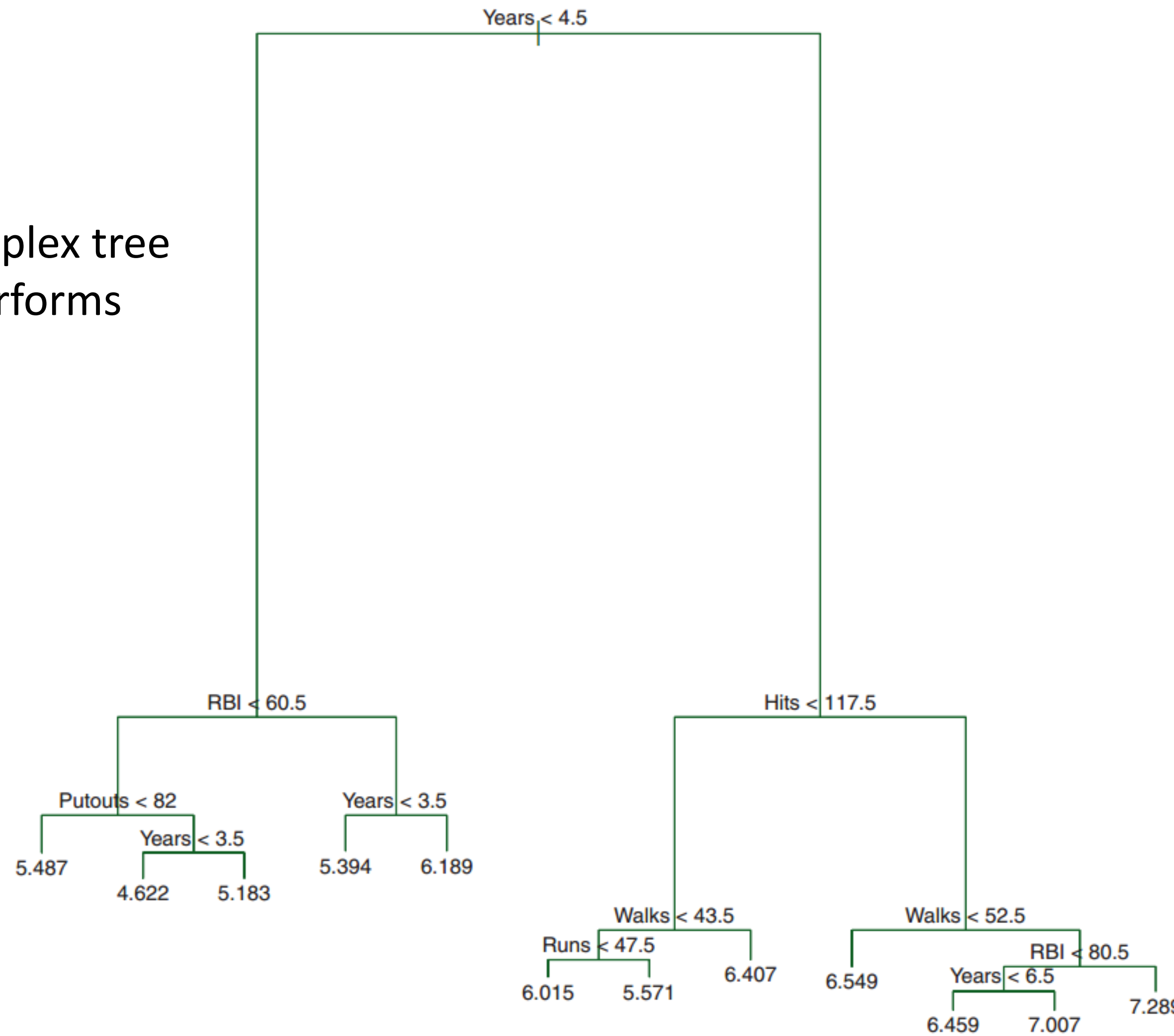
Building the tree

For any predictor j and every cut-point s we consider:
$R_1(j, s) = \{X|X_j < s\}$ and $R_2(j, s) = \{X|X_j \geq s\}$
and choose j and s that result in the smallest RSS.

Next, this process is repeated, looking for the best predictor and cut-point. But this time only the predictor space within a region is split.

Process continues until some stopping rule is reached (e.g., go on until no region has more than five observations)

This will likely result in a complex tree that overfits the data and performs poorly on a test set.



Tree pruning

Select a *subtree* of the very large tree, T_0, we ended up with on the previous slide.

Cost complexity pruning/ weakest link pruning:
Instead of considering every possible subtree, we consider a sequence of trees indexed by a nonnegative tuning parameter α.

$\sum_{m=1}^{|T|} \sum_{i:\, x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha |T|$

m = terminal node
|*T*|= total number of terminal nodes

α of 0 results in T_0

Branches get pruned in a nested fashion.
α can be selected with cross validation or a validation test set.

Cross validation to select α

Divide training data in K folds. For each k = 1, … , K:

A)

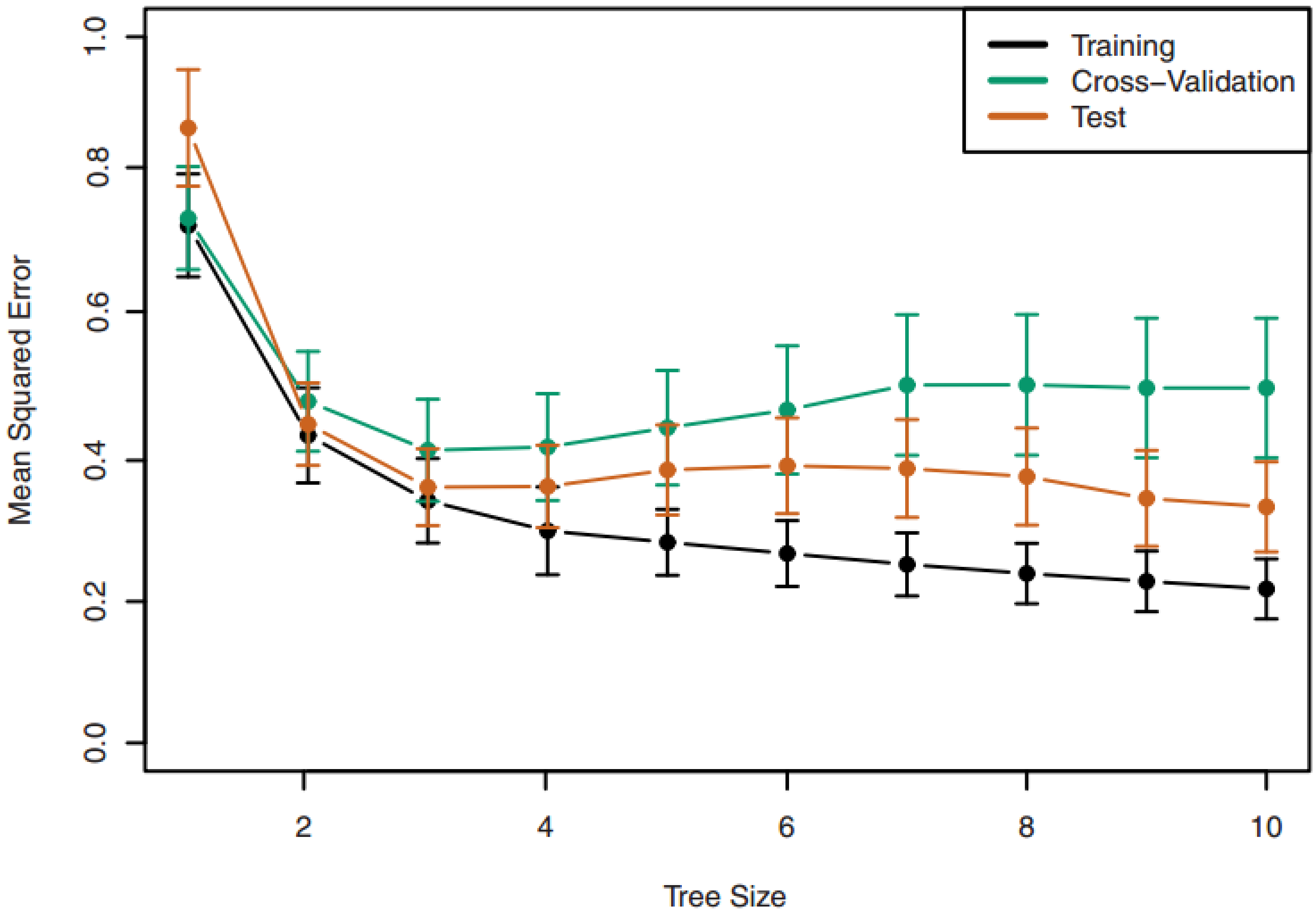
1. Use recursive binary splitting to grow a large tree on all but the kth fold
2. Apply cost complexity pruning to the large tree to obtain a sequence of best subtrees, as a function of α.

B) Evaluate the mean squared prediction error on the data of the left out kth fold, as a function of α.

C) Average the results for each value of α over all folds, and pick α to minimize the average error.

Now that α is picked, we choose the subtree of the full training data that corresponds to this α. (performing (1) and (2) on the full training set)

Cross validation to select α



Classification trees

Assign observation in a given region to the most commonly occurring class of training observations in that region.

$$E = 1 - \max_k(\hat{p}_{mk})$$

Classification error is not sensitive enough to grow trees.

Gini index:

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

Pure terminal nodes

Cross-entropy:

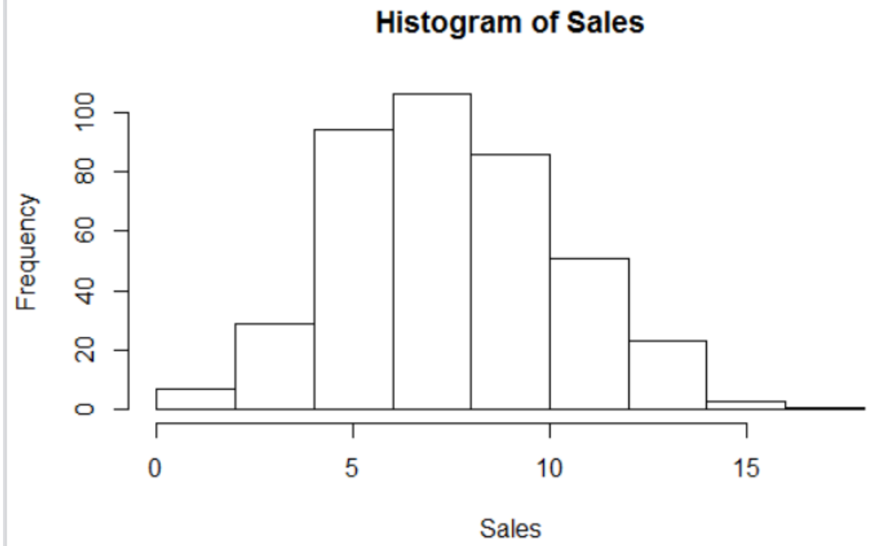
$$D = -\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}$$

$$0 \leq -\,\hat{p}_{mk} \log \hat{p}_{mk}$$

Trees in R

```
require(ISLR)
require(tree)
attach(Carseats)
```

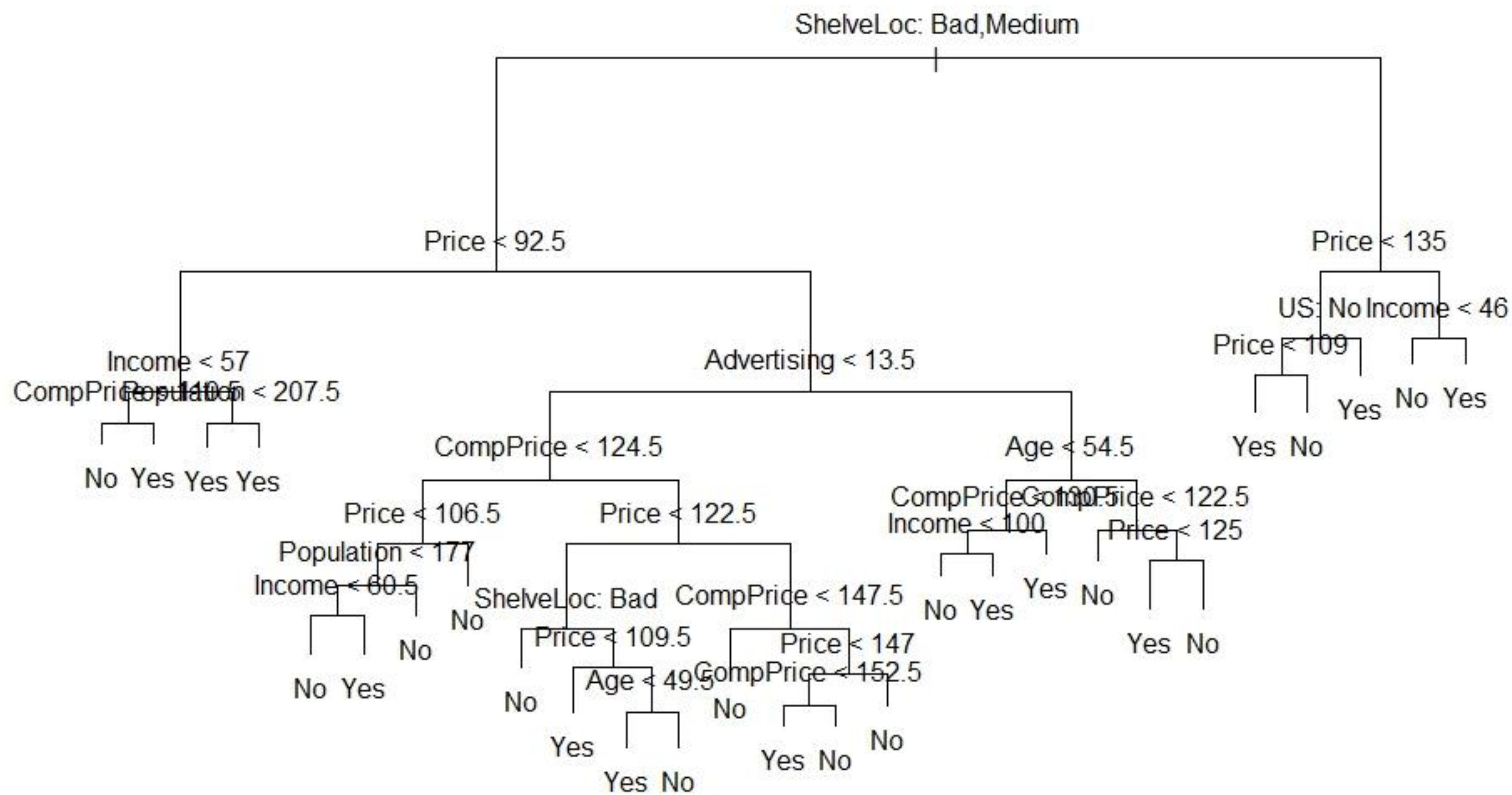
400 obs: stores




```
> head(Carseats)
  Sales CompPrice Income Advertising Population Price ShelveLoc Age Education
1  9.50       138     73          11        276   120       Bad  42        17
2 11.22       111     48          16        260    83      Good  65        10
3 10.06       113     35          10        269    80    Medium  59        12
4  7.40       117    100           4        466    97    Medium  55        14
5  4.15       141     64           3        340   128       Bad  38        13
6 10.81       124    113          13        501    72       Bad  78        16
  Urban  US High
1   Yes Yes  Yes
2   Yes Yes  Yes
3   Yes Yes  Yes
4   Yes Yes   No
5   Yes  No   No
6    No Yes  Yes
>
```

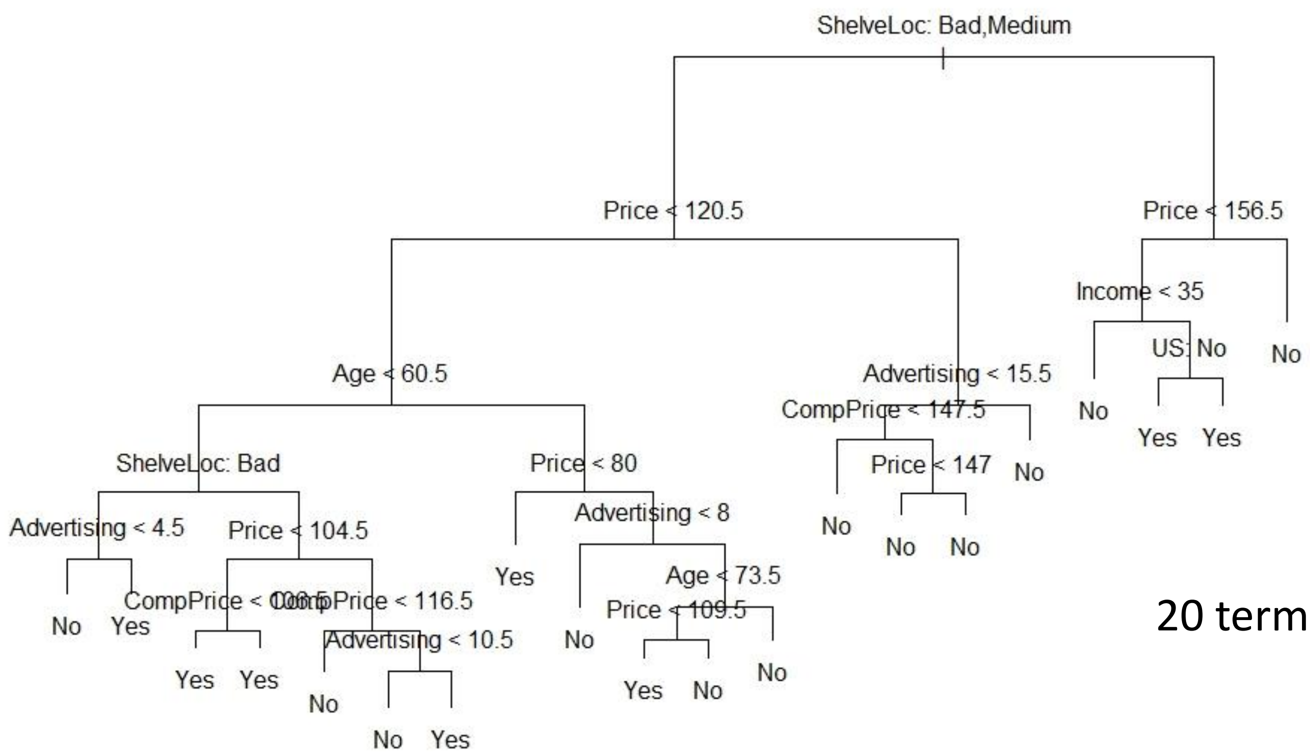
Trees in R

```
tree.carseats<-tree(High~.-Sales,data=Carseats)
summary(tree.carseats)
plot(tree.carseats)
text(tree.carseats,pretty=0)
```



Trees in R

```
train=sample(1:nrow(Carseats),250)
tree.carseats=tree(High~.-Sales,Carseats,subset=train)
plot(tree.carseats);text(tree.carseats,pretty=0)
```

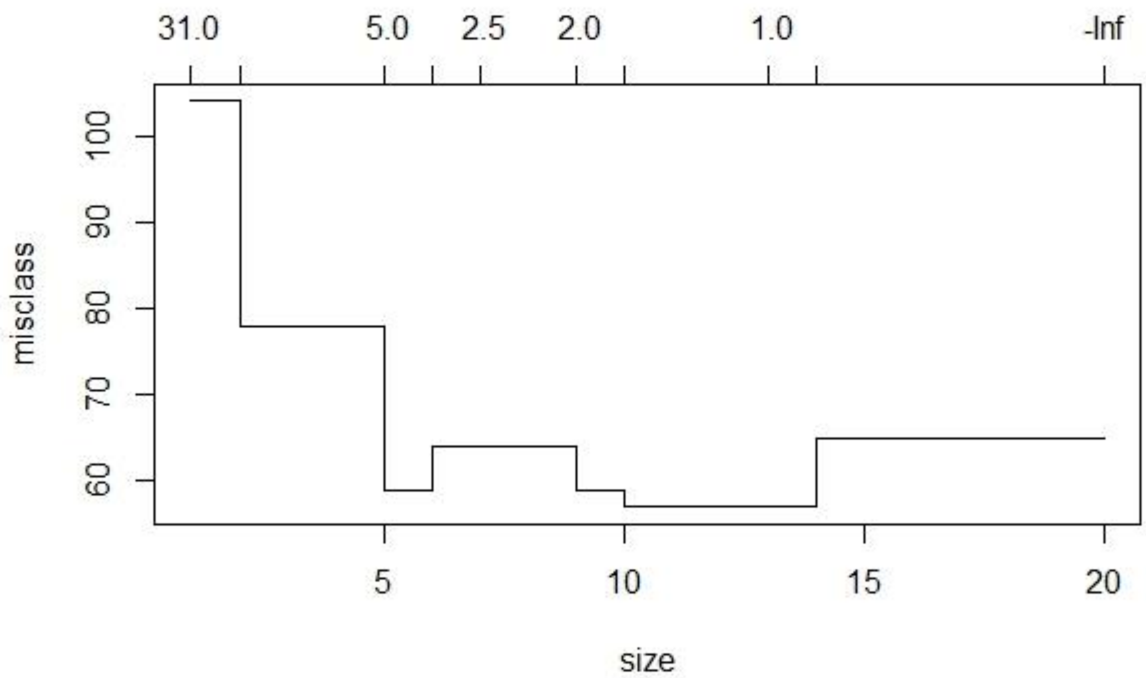


20 terminal nodes

Trees in R

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

```
> with(Carseats[-train,],table(tree.pred,High))
         High
tree.pred No Yes
      No  72  27
      Yes 18  33
> (72+33)/150
[1] 0.7
```

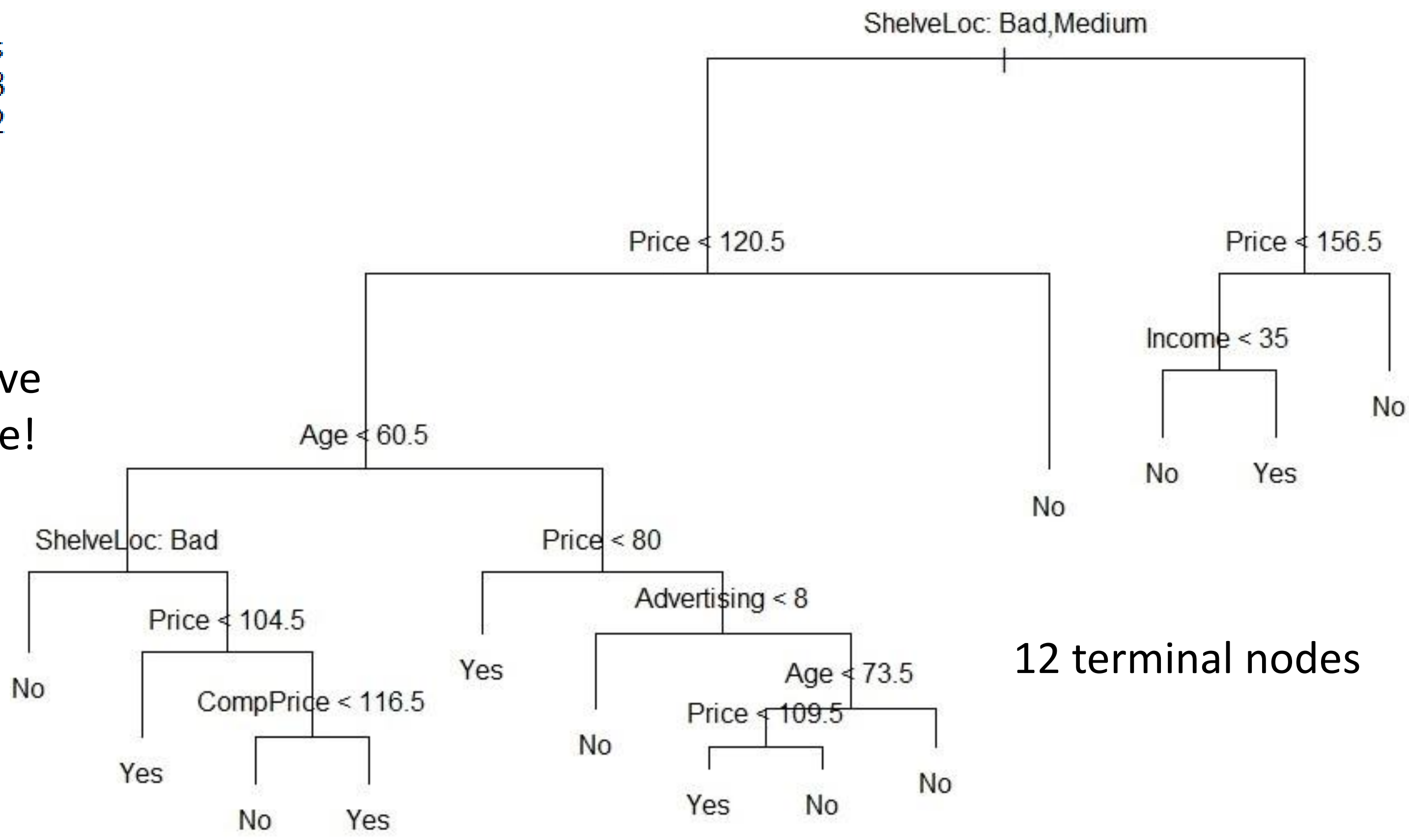


Trees in R

```
prune.carseats<-prune.misclass(tree.carseats,best=12)
plot(prune.carseats);text(prune.carseats,pretty=0)
```

```
> tree.pred=predict(prune.carseats,Carseats[-train,],type="class")
> with(Carseats[-train,],table(tree.pred,High))
         High
tree.pred No Yes
      No  72  28
      Yes 18  32
> (72+32)/150
[1] 0.6933333
```

Test error the same but CV gave us a simpler tree!

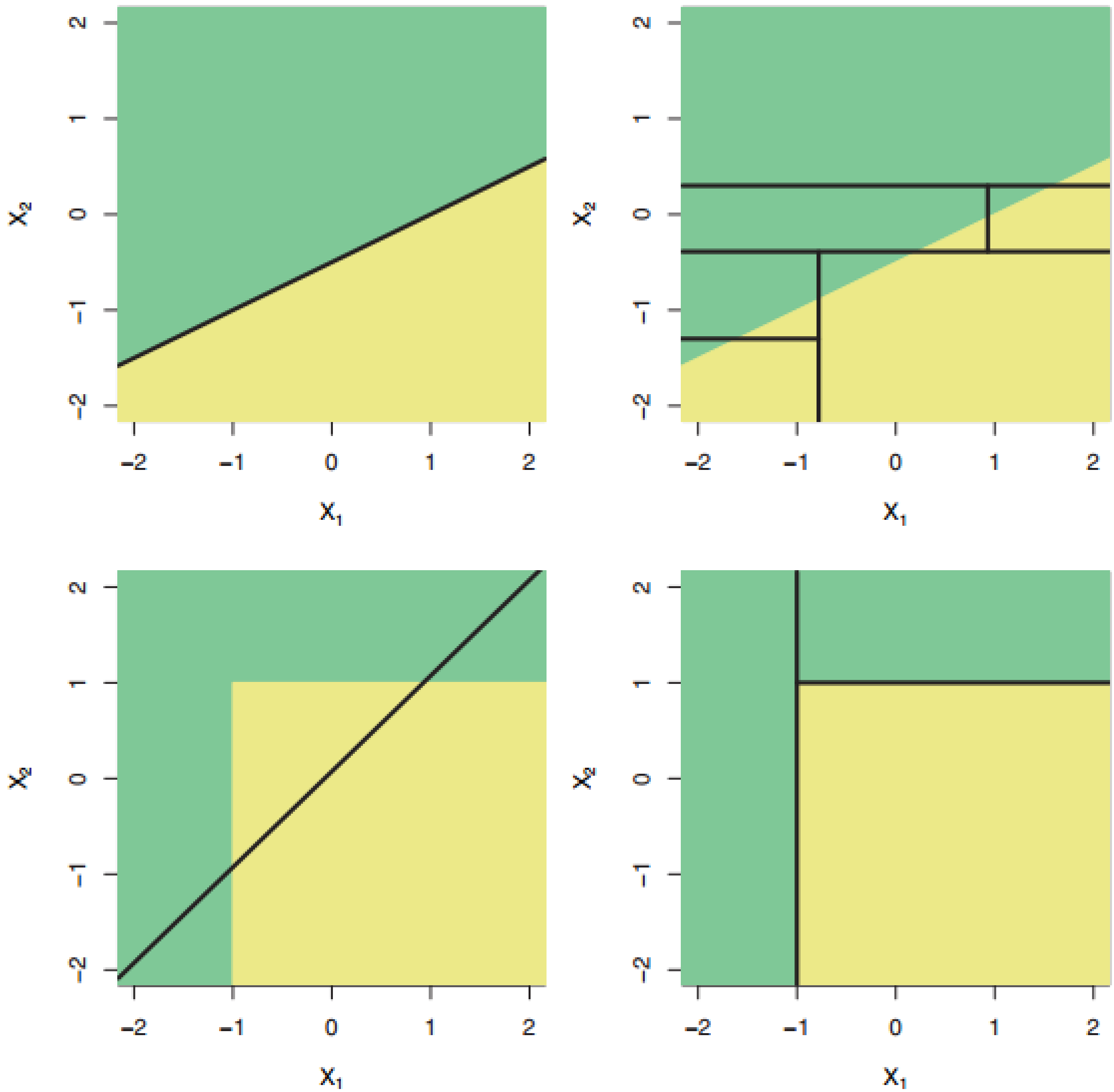


12 terminal nodes

Trees or Linear Models?

With a linear relation, a linear model will outperform trees, while if there is a highly non-linear and complex relationship between the features and the response, a tree is more suited.

Test the performance with cross-validation or a validation set.



Advantages and Disadvantages:

+ Trees are easy to explain to people, even easier than linear regression.

+ Some people believe that decision trees more closely mirror human decision-making than do other regression and classification approaches.

+ Trees can be displayed graphically and are easily interpreted.

+ Trees can easily handle qualitatie predictors without the need to create dummy variables.

- Trees generally do not have the same level of predictive accuracy as some of the orher regression and classification approaches.

- Trees can be very non-robust.

But, using random forests, boosting and bagging will improve the predictive performance!

The End