MONASH University

ETC3250 Business Analytics: Advanced Classification - Trees & Forests

Souhaib Ben Taieb, Di Cook, Rob Hyndman

October 5, 2015

Decision trees

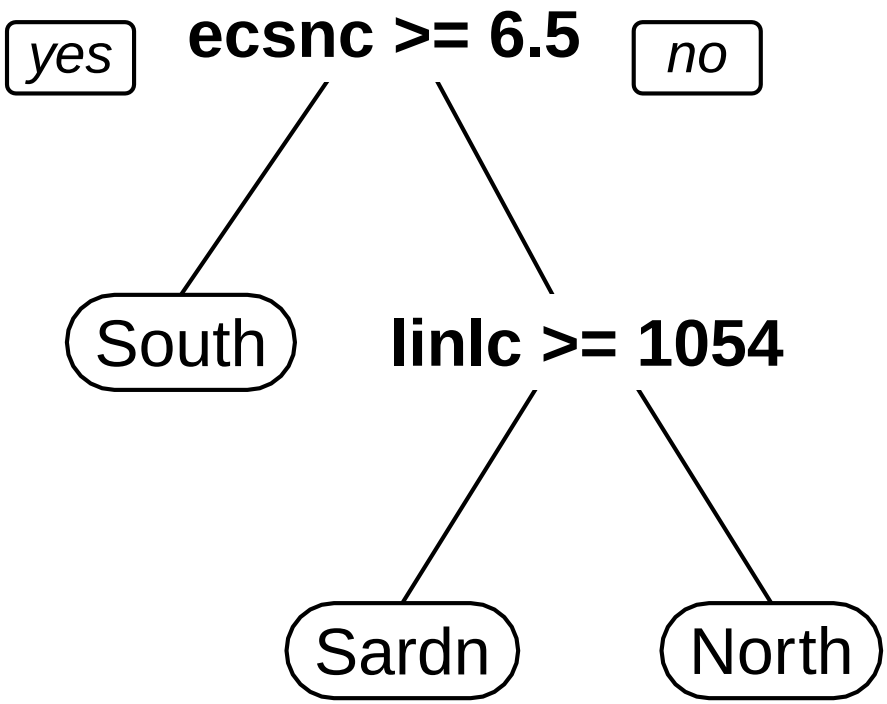
- Recursive binary splitting
- Compute all possible splits $(n-1)$ on every variable (p)
- Choose the best split of the data, the one that separates it into two groups which are the most "pure"
- Continue to operate on each of the subsets until a stopping criteria is satisfied (e.g. all cases in the subset are of one class, there are less than m cases in a subset, ...)

Example: olive oils

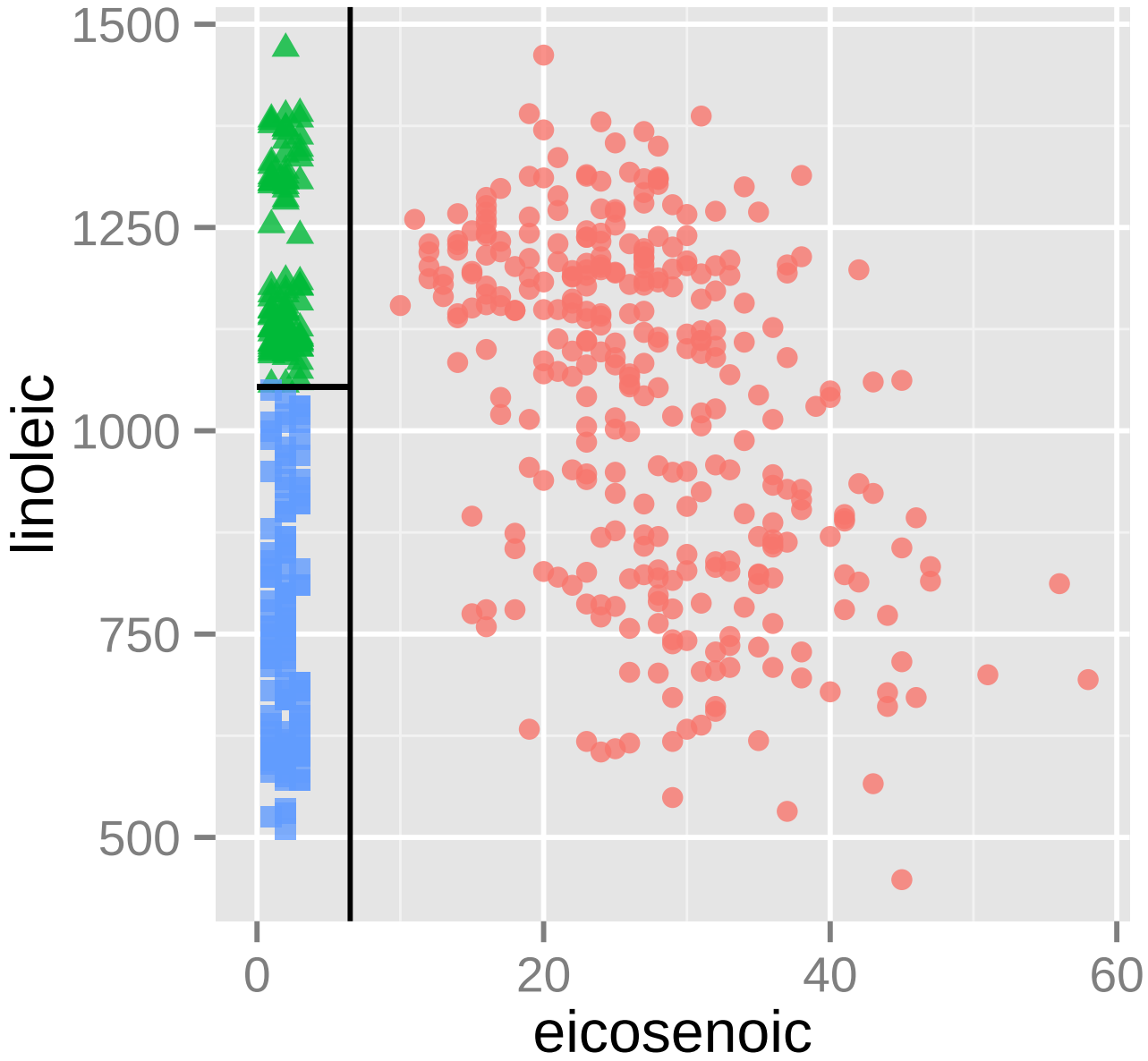
```
## n= 572
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
## 1) root 572 250 South (0.56 0.17 0.26)
##   2) eicosenoic>=6.5 323   0 South (1.00 0.00 0.00) *
##   3) eicosenoic< 6.5 249  98 North (0.00 0.39 0.61)
##     6) linoleic>=1.1e+03 98   0 Sardinia (0.00 1.00 0.00) *
##     7) linoleic< 1.1e+03 151   0 North (0.00 0.00 1.00) *

##
##              South Sardinia North
##   South        323        0     0
##   Sardinia       0       98     0
##   North          0        0   151
```

Example: olive oils



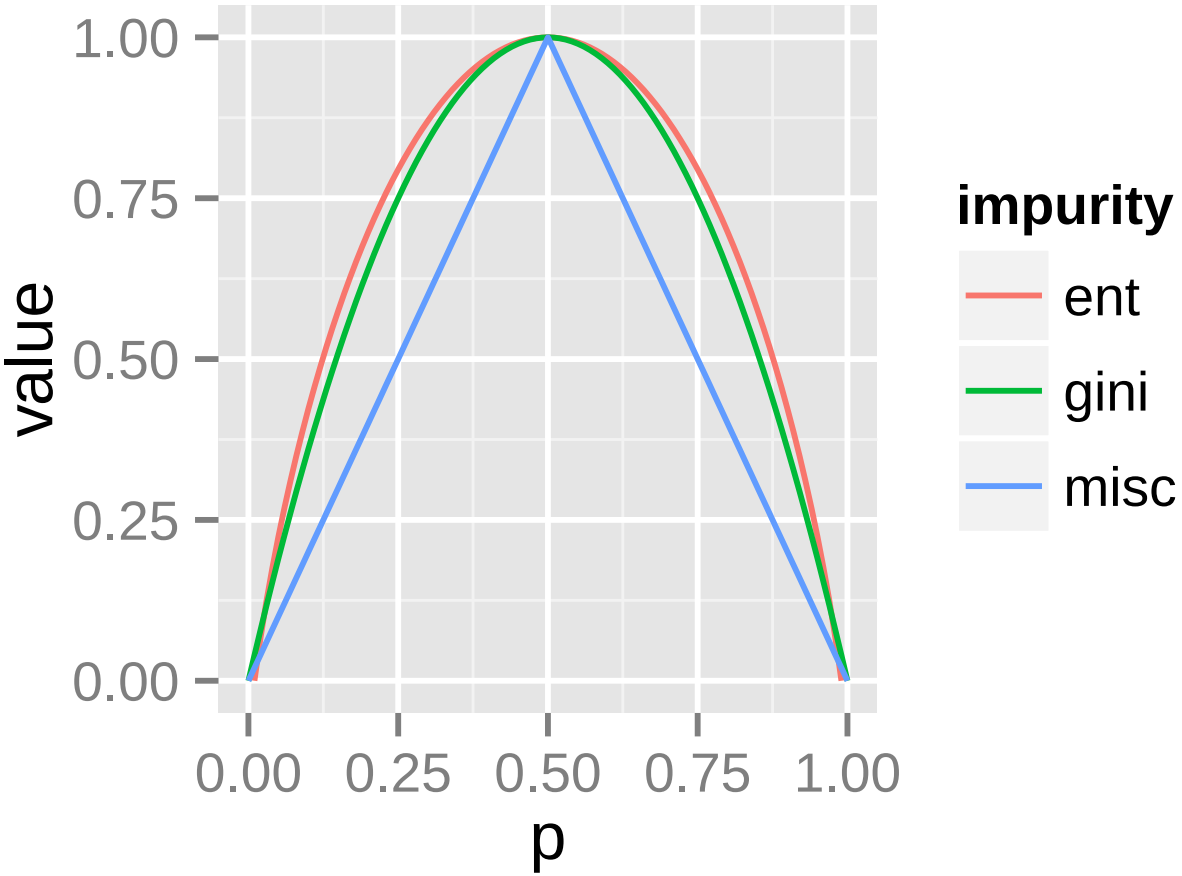
Example: olive oils



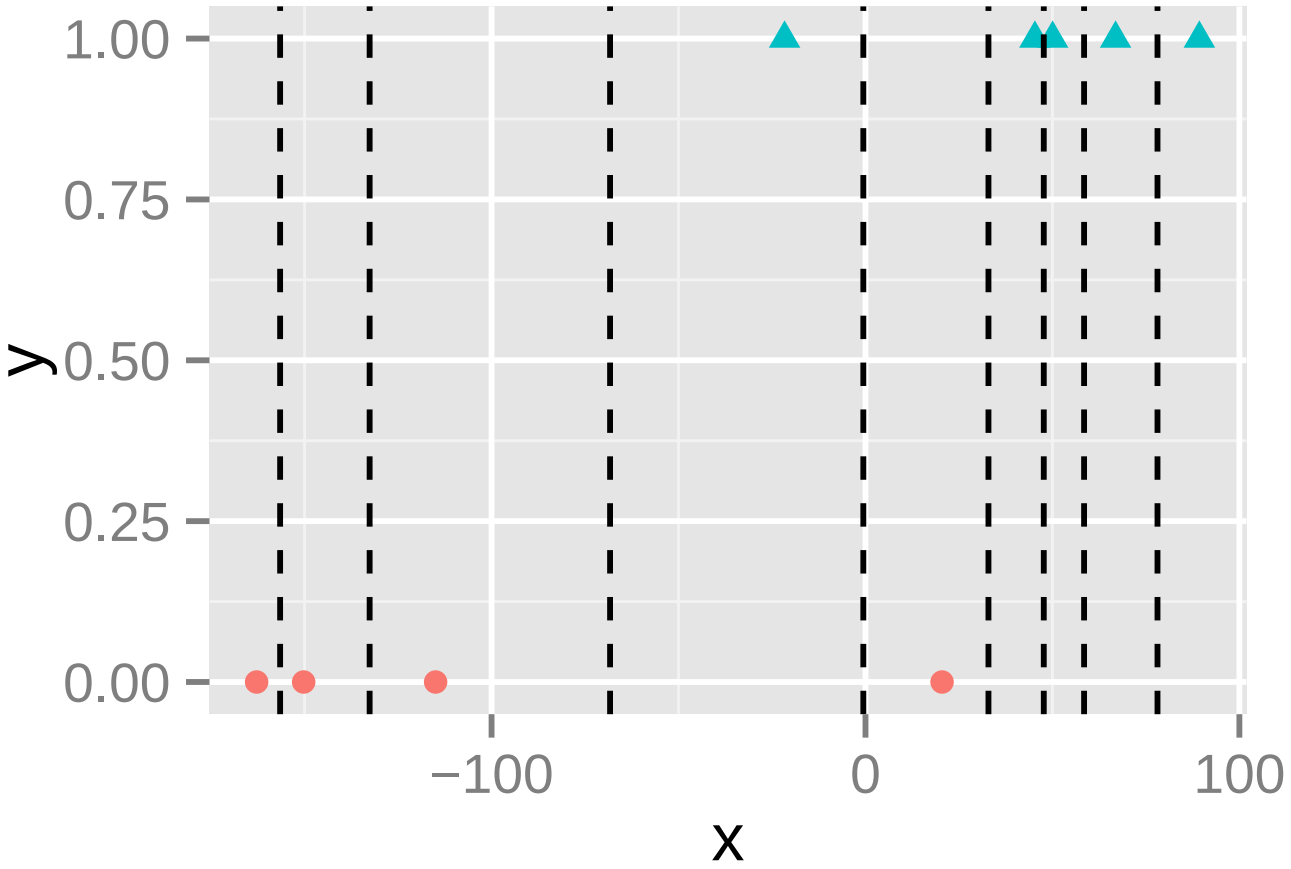
Measuring the quality of splits

- Explanation for two classes (0,1), and p=proportion in class 0
- Entropy: $-p(log_e p) - (1-p)log_e(1-p)$
- Gini: $2p(1-p)$
- Misclassification: $1 - 2|p - 0.5|$

What these look like



Choosing the best split

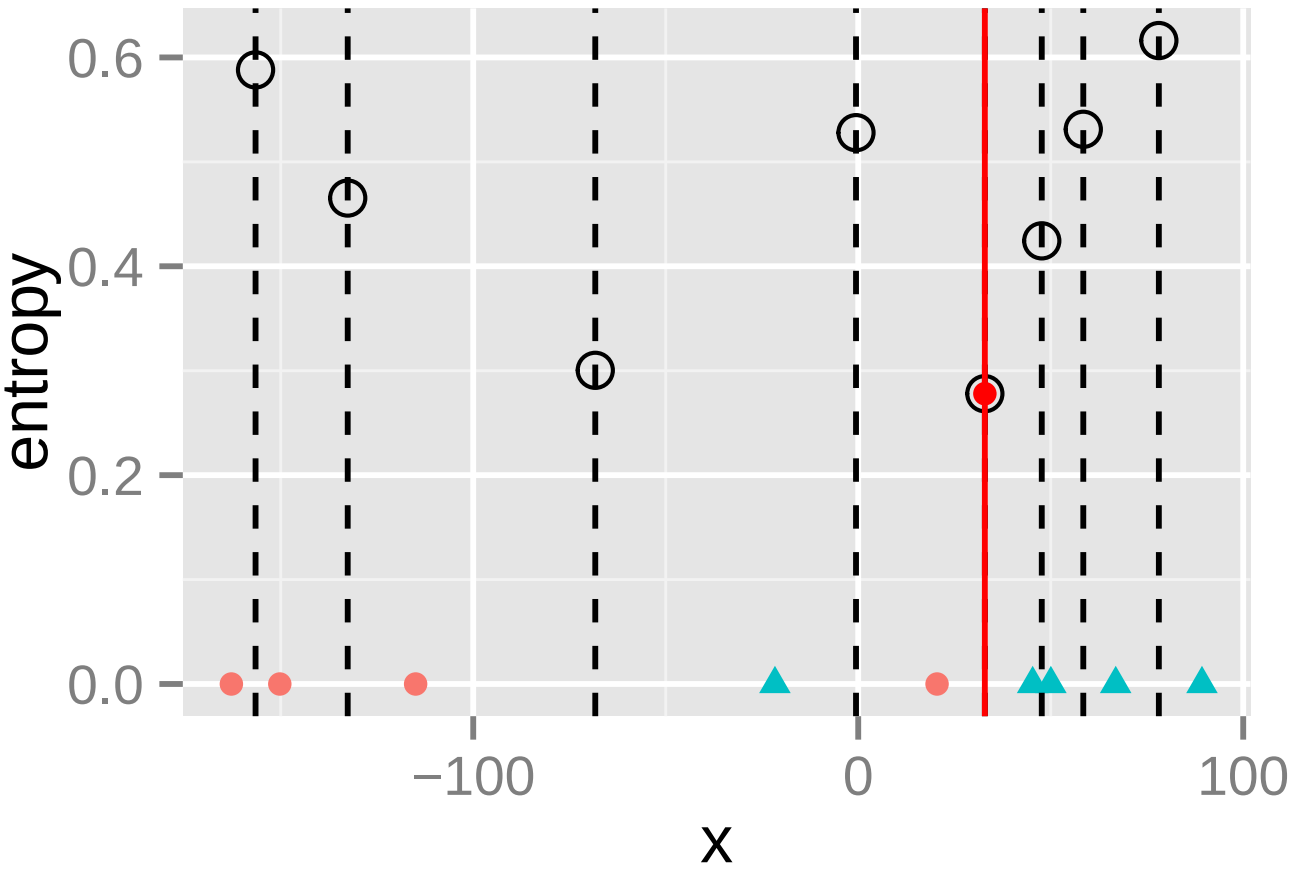


Choosing the best split

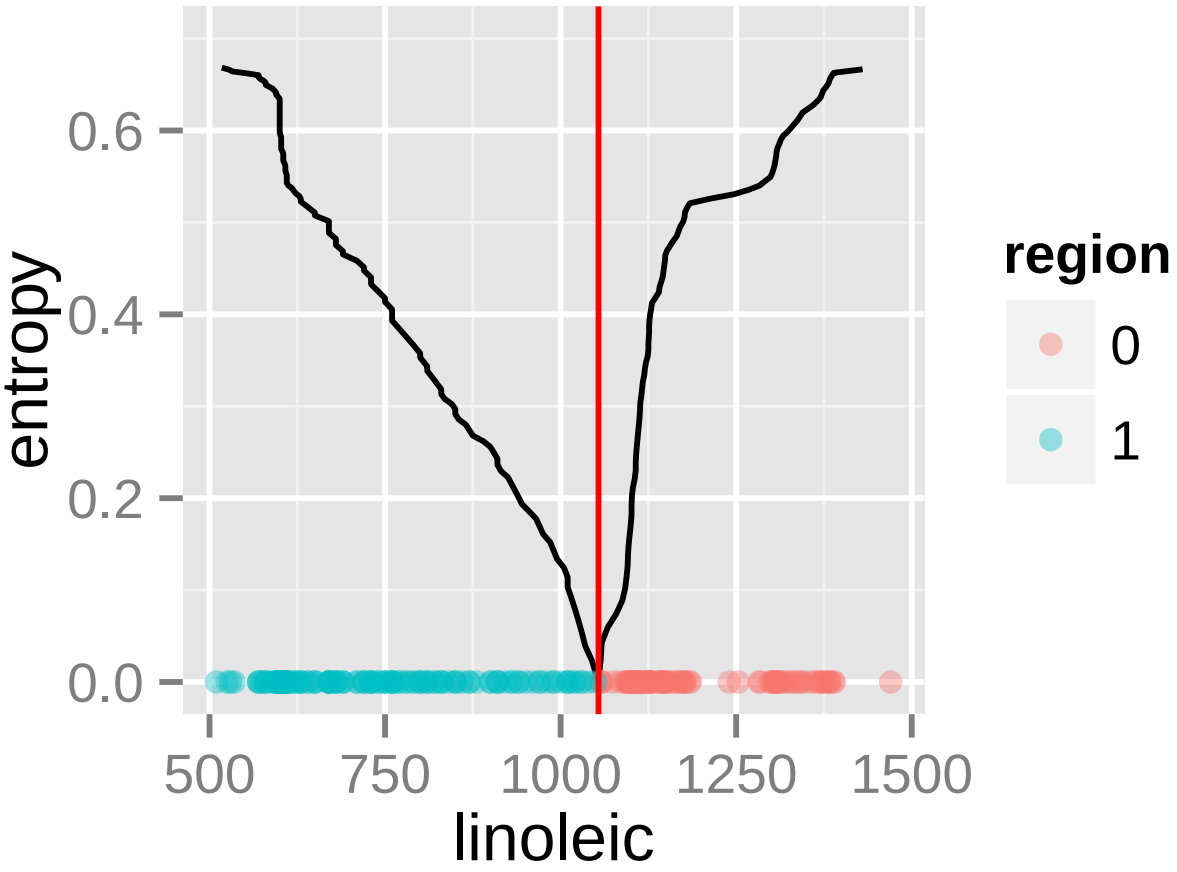
- Calculate the impurity for each subset, and combine
- p^L impurity $_L + p^R$ impurity$_R$

```
##          x  ent
## 1 -156.53 0.59
## 2 -132.60 0.47
## 3  -68.29 0.30
## 4   -0.57 0.53
## 5   32.90 0.28
## 6   47.68 0.42
## 7   58.46 0.53
## 8   78.09 0.62
```

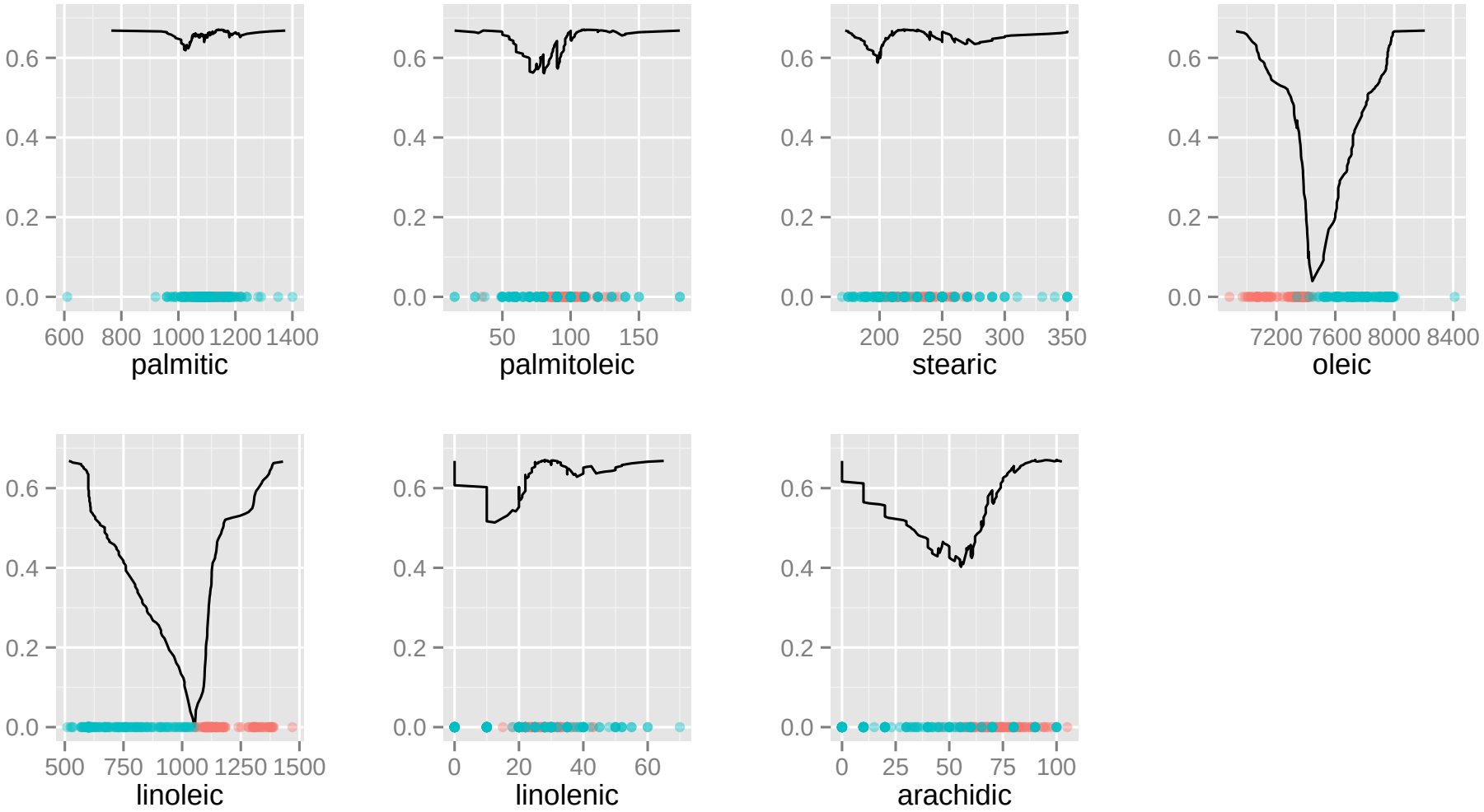
Choosing the best split



Example: olive oils



Example: olive oils



Stopping rules

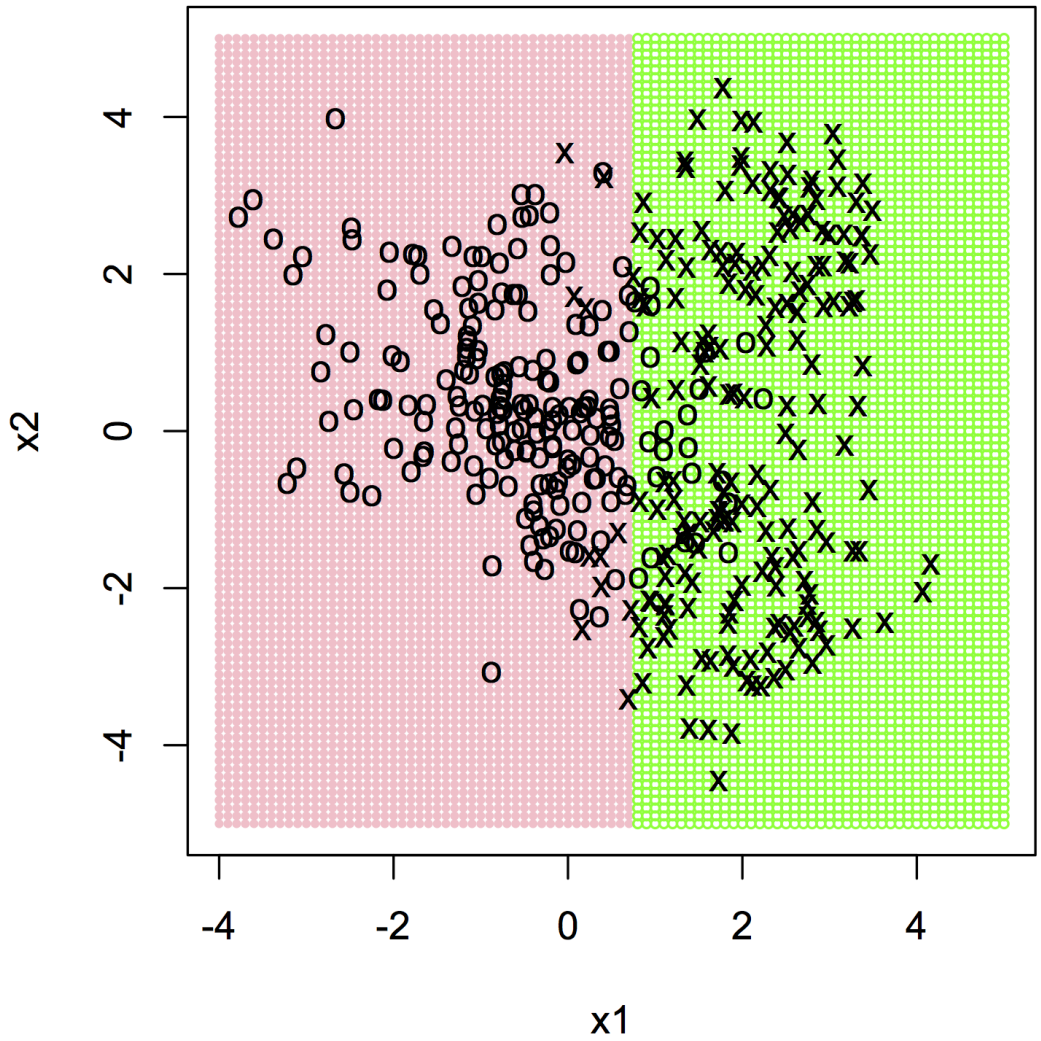
- *minsplit:* minimum number of observations allowed in order to consider splitting
- *minbucket:* minimum number of observations in a terminal node
- *cp (0.01):* complexity parameter. The decrease in impurity cannot be less than this.

Overfitting

- It is possible to force a tree to fit the training sample very closely, by tweaking these stopping rules.
- This could lead to overfitting, really small error with the training data and much higher error with validation data, and hence test data.
- Tuning the algorithm control parameters with the validation set is really important.

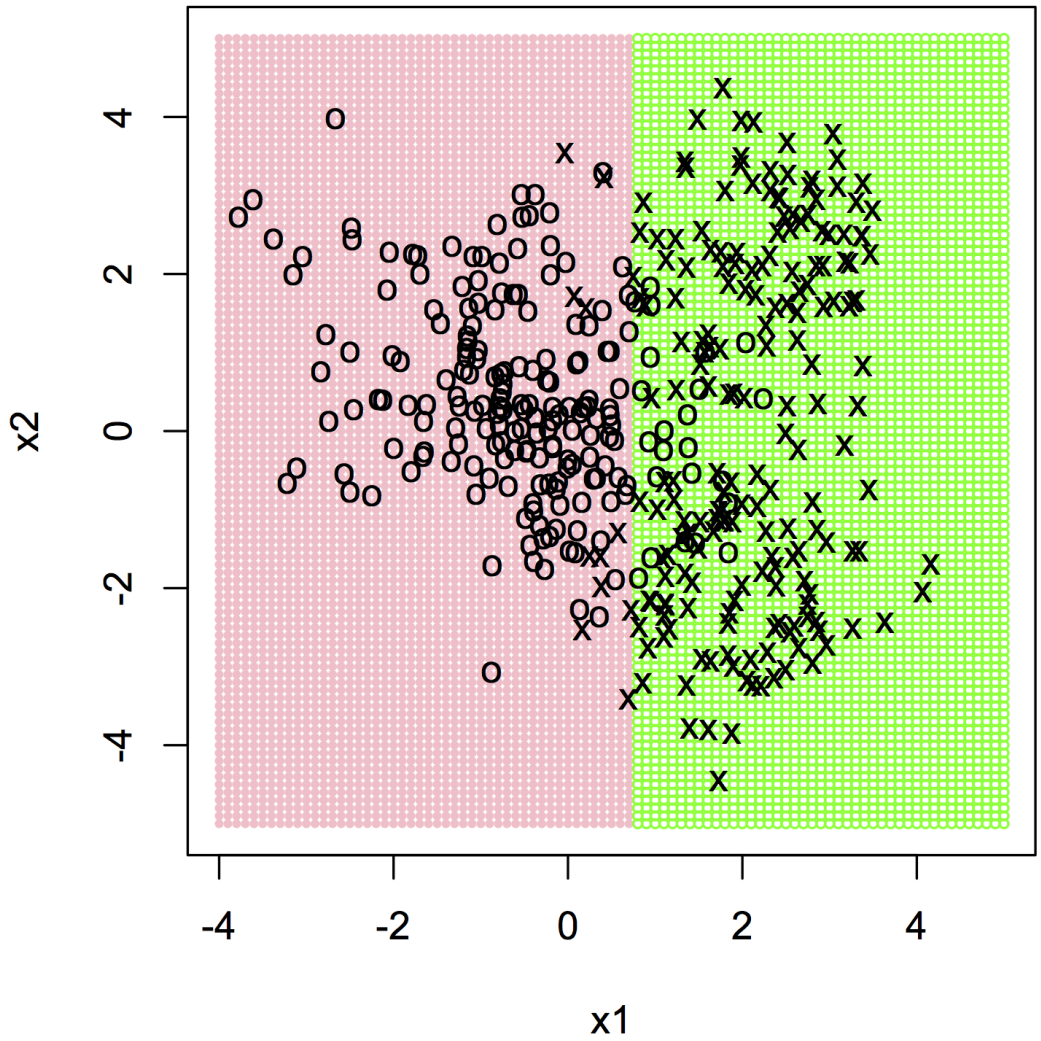
Overfitting example

```
> df.rp <- rpart(y~., data=df,
+                control=rpart.control(minsplit=30, cp=0.01))
```



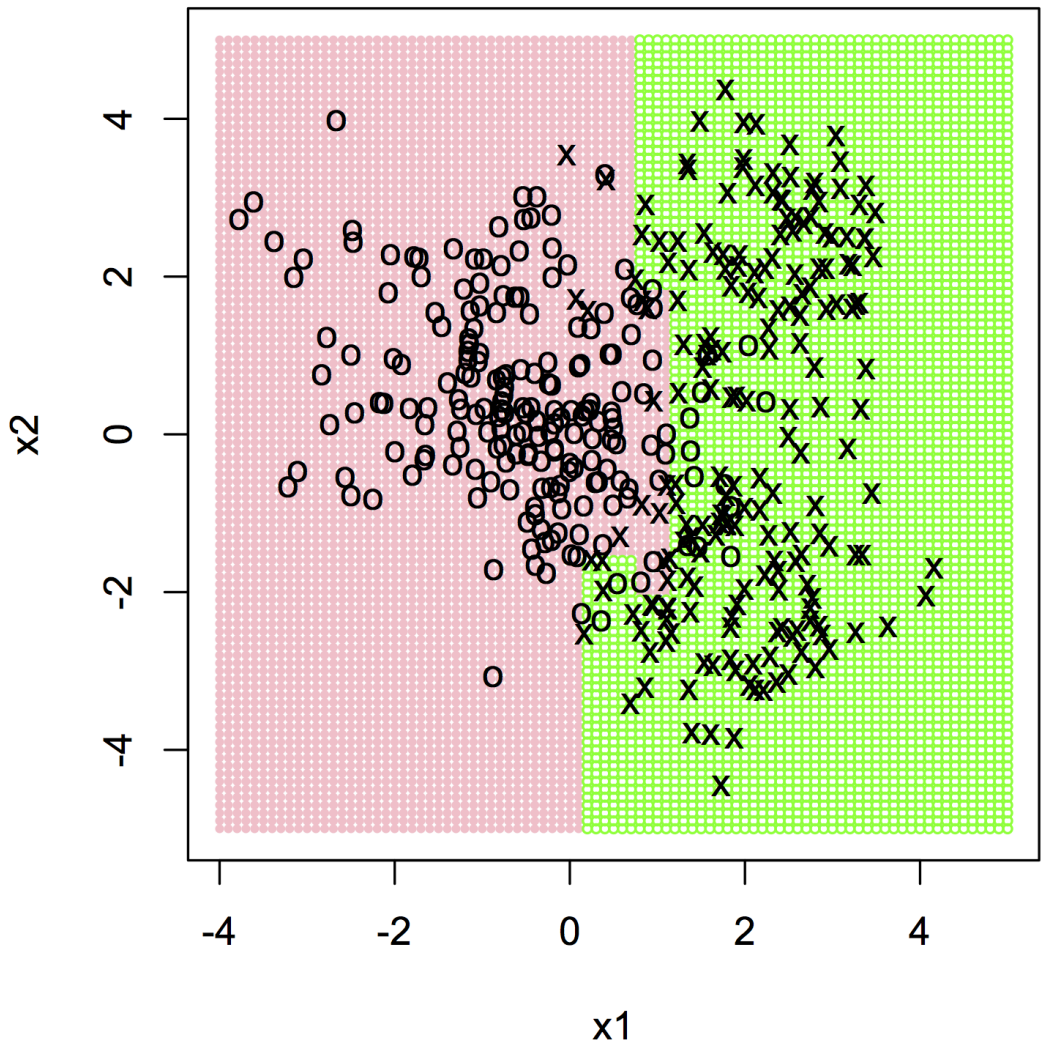
Overfitting example

```
> df.rp <- rpart(y~., data=df,
+                control=rpart.control(minsplit=30, cp=0.01))
```



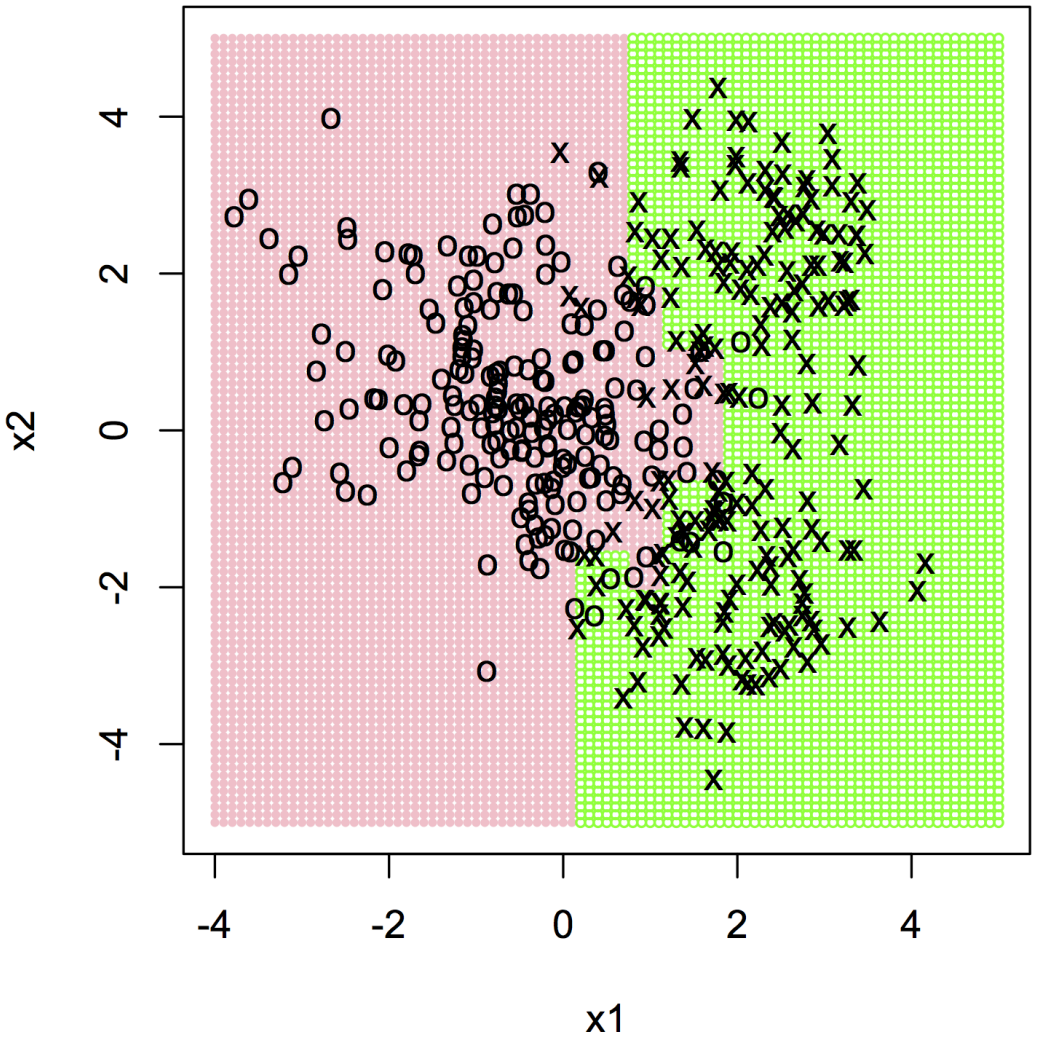
Overfitting example

```
> df.rp <- rpart(y~., data=df,
+                control=rpart.control(minsplit=10, cp=0.01))
```



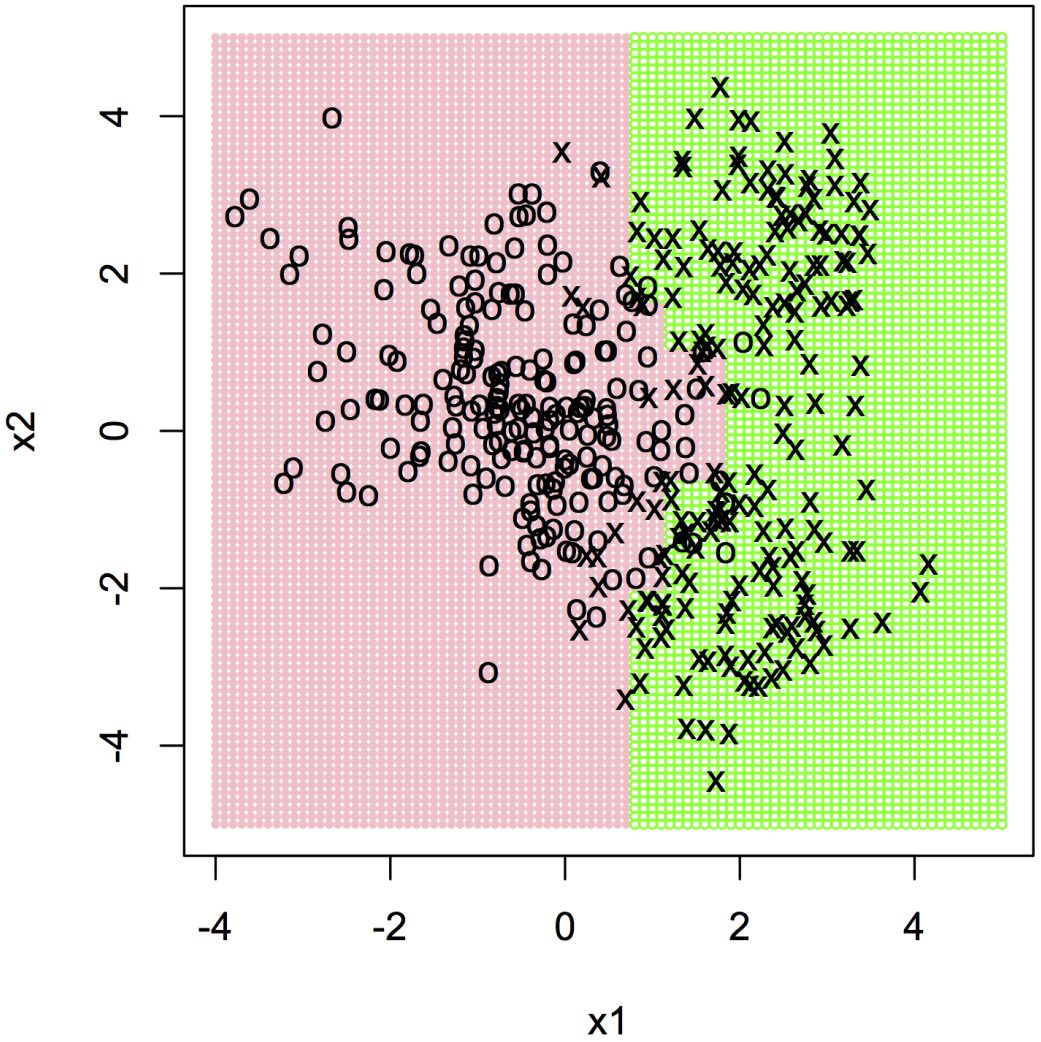
Overfitting example

```
> df.rp <- rpart(y~., data=df,
+                control=rpart.control(minsplit=2, cp=0.01))
```



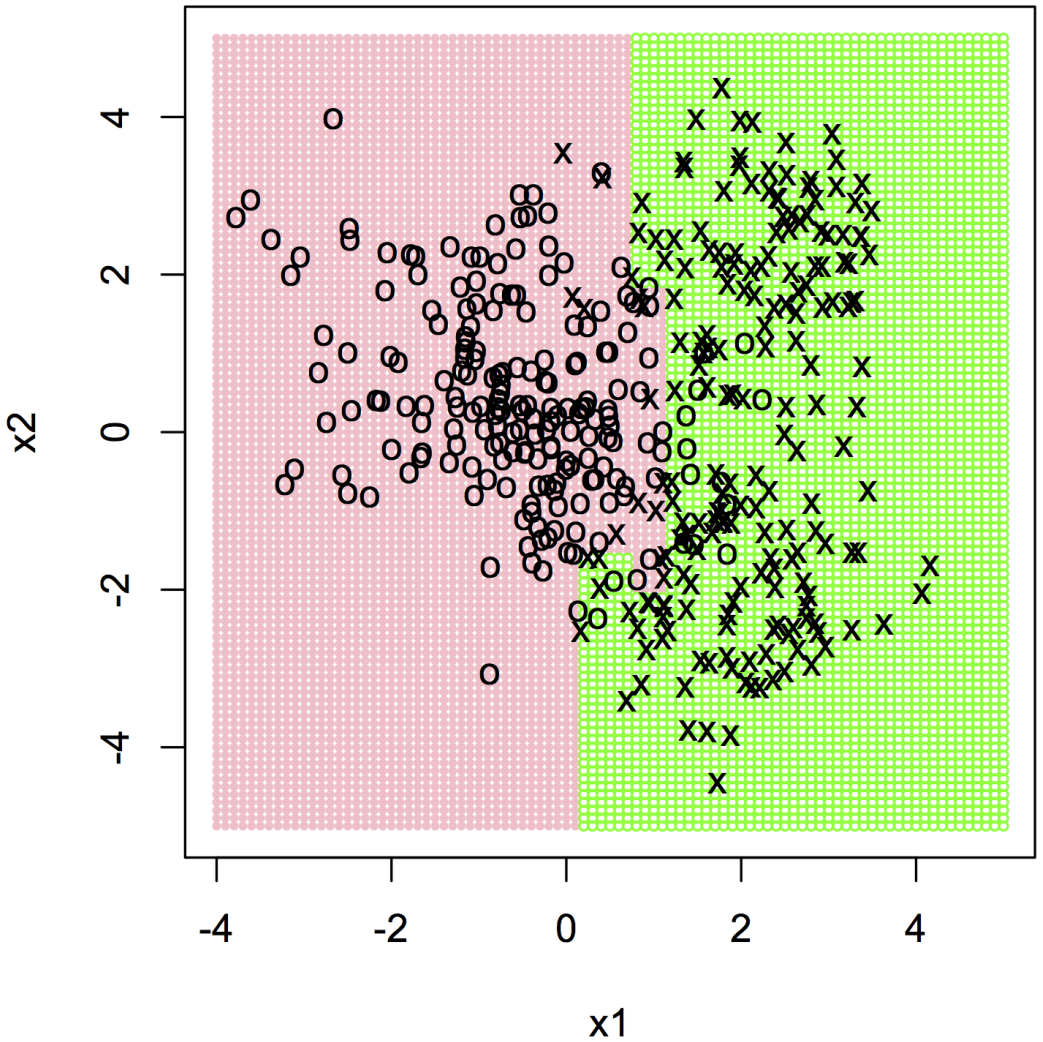
Overfitting example

```
> df.rp <- rpart(y~., data=df,
+                control=rpart.control(minsplit=20, cp=0.000001))
```



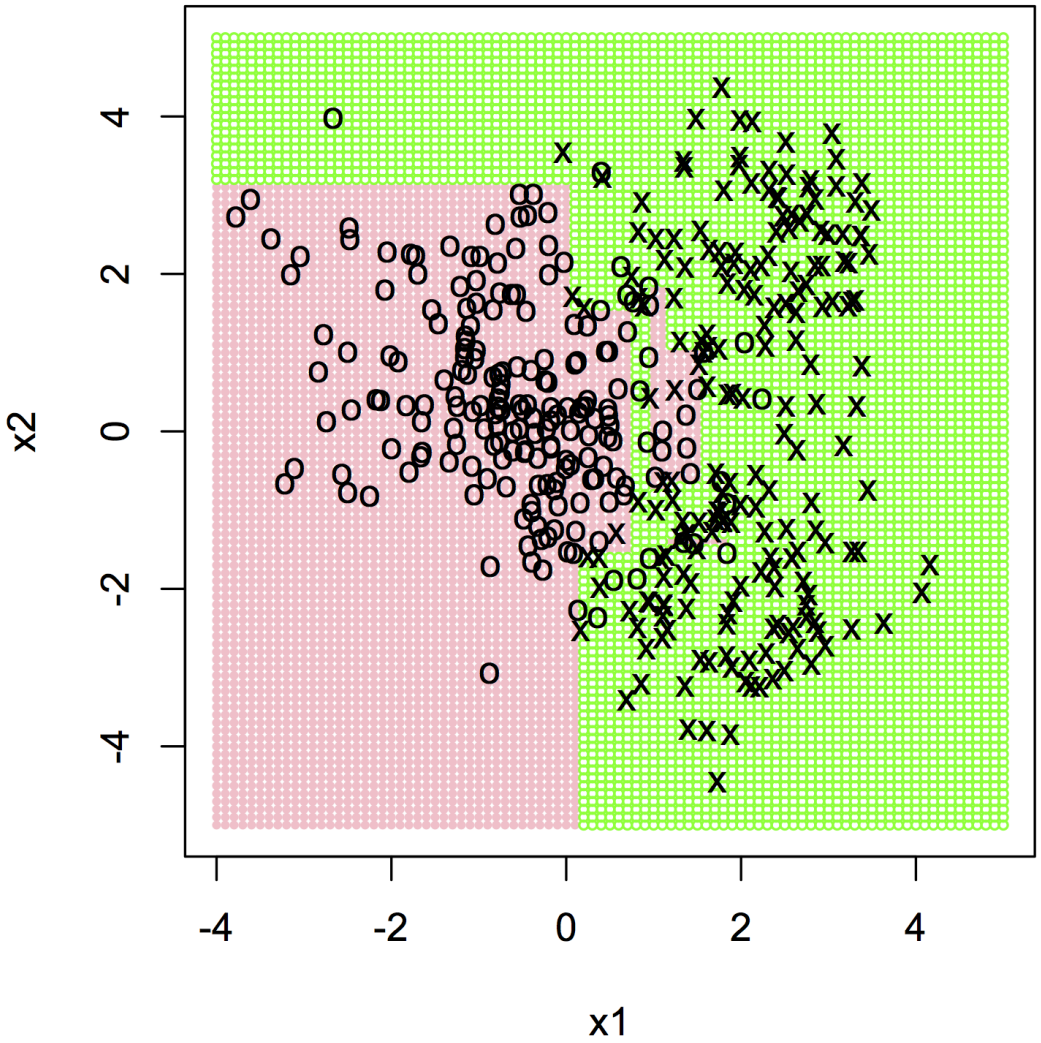
Overfitting example

```
> df.rp <- rpart(y~., data=df,
+               control=rpart.control(minsplit=10, cp=0.01))
```



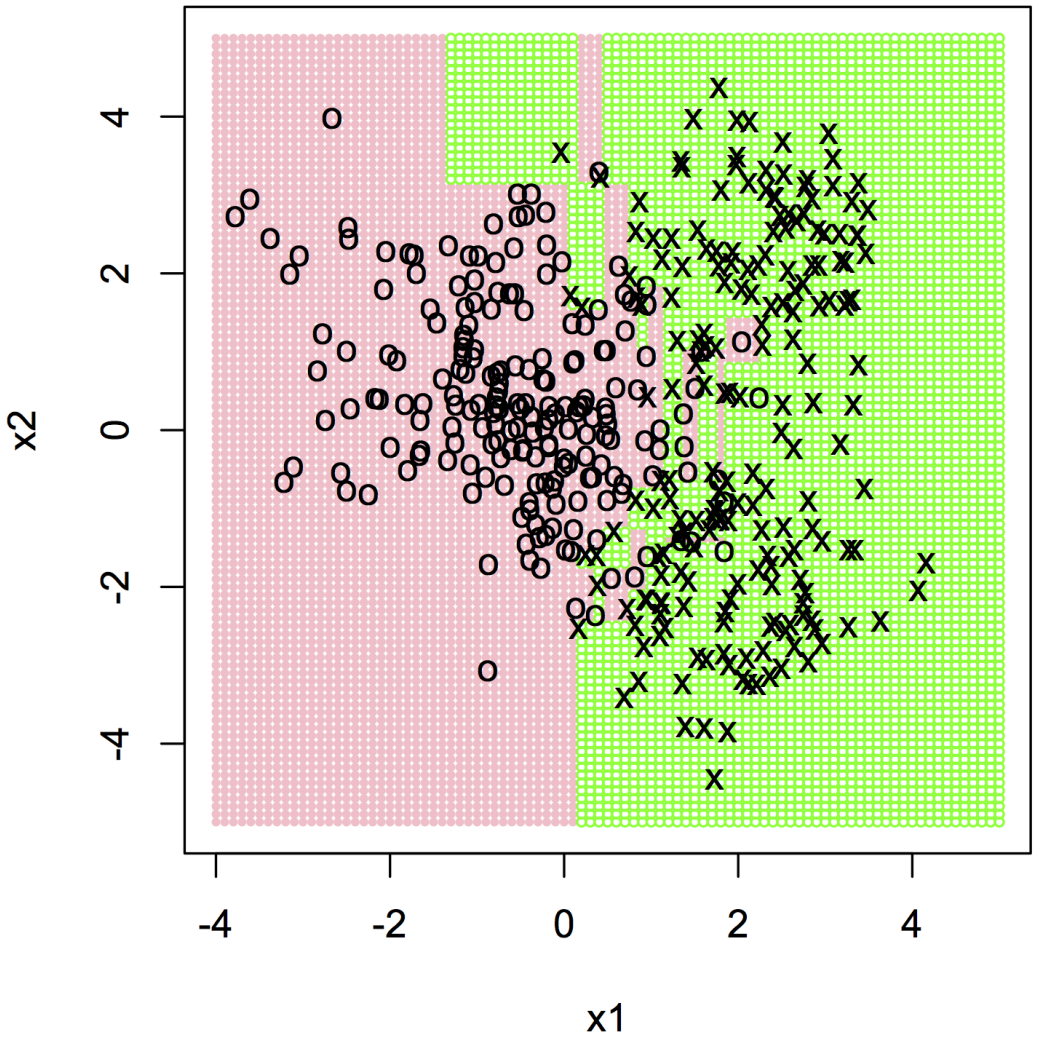
Overfitting example

```
> df.rp <- rpart(y~., data=df,
+                control=rpart.control(minsplit=10, cp=0.001))
```

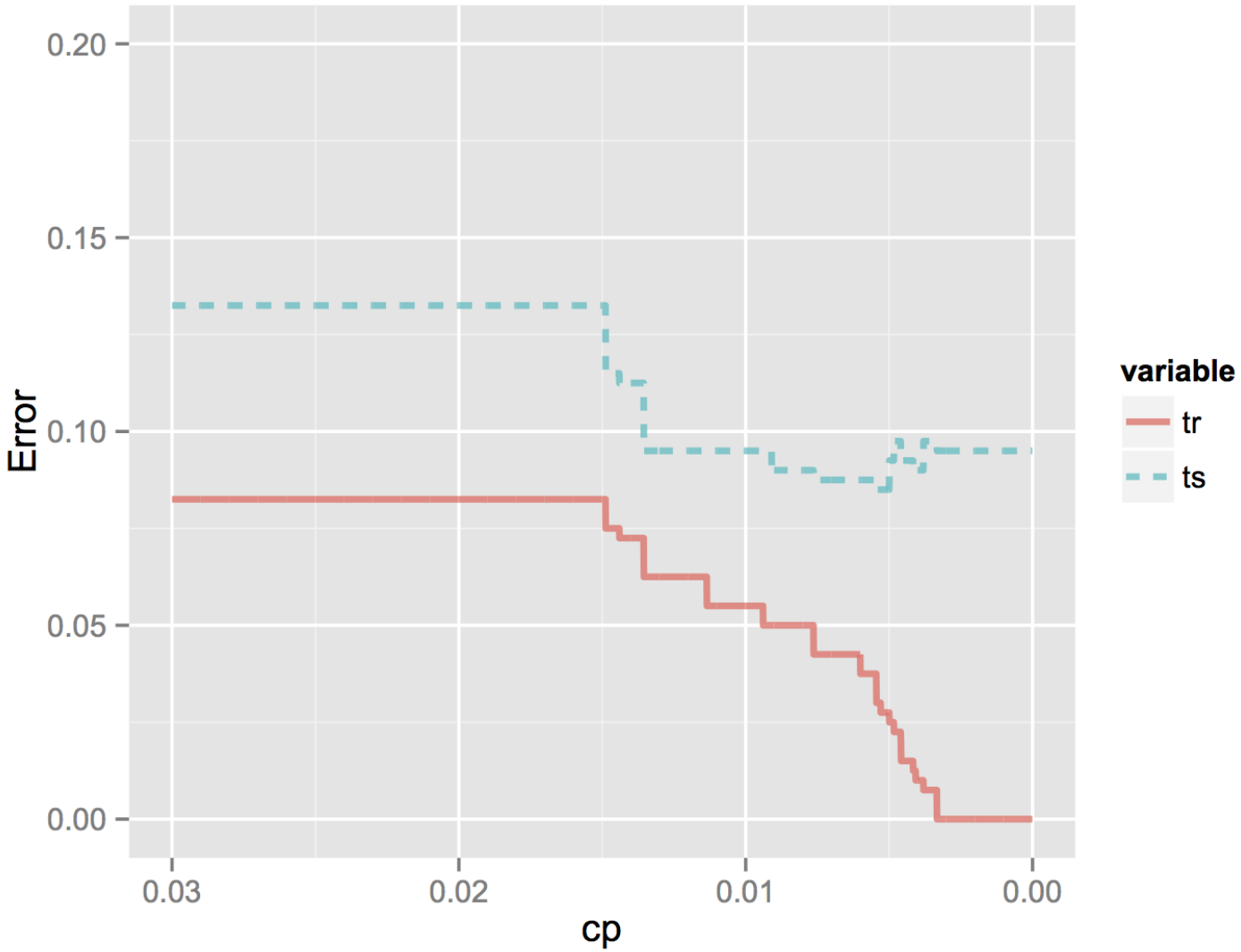


Overfitting example

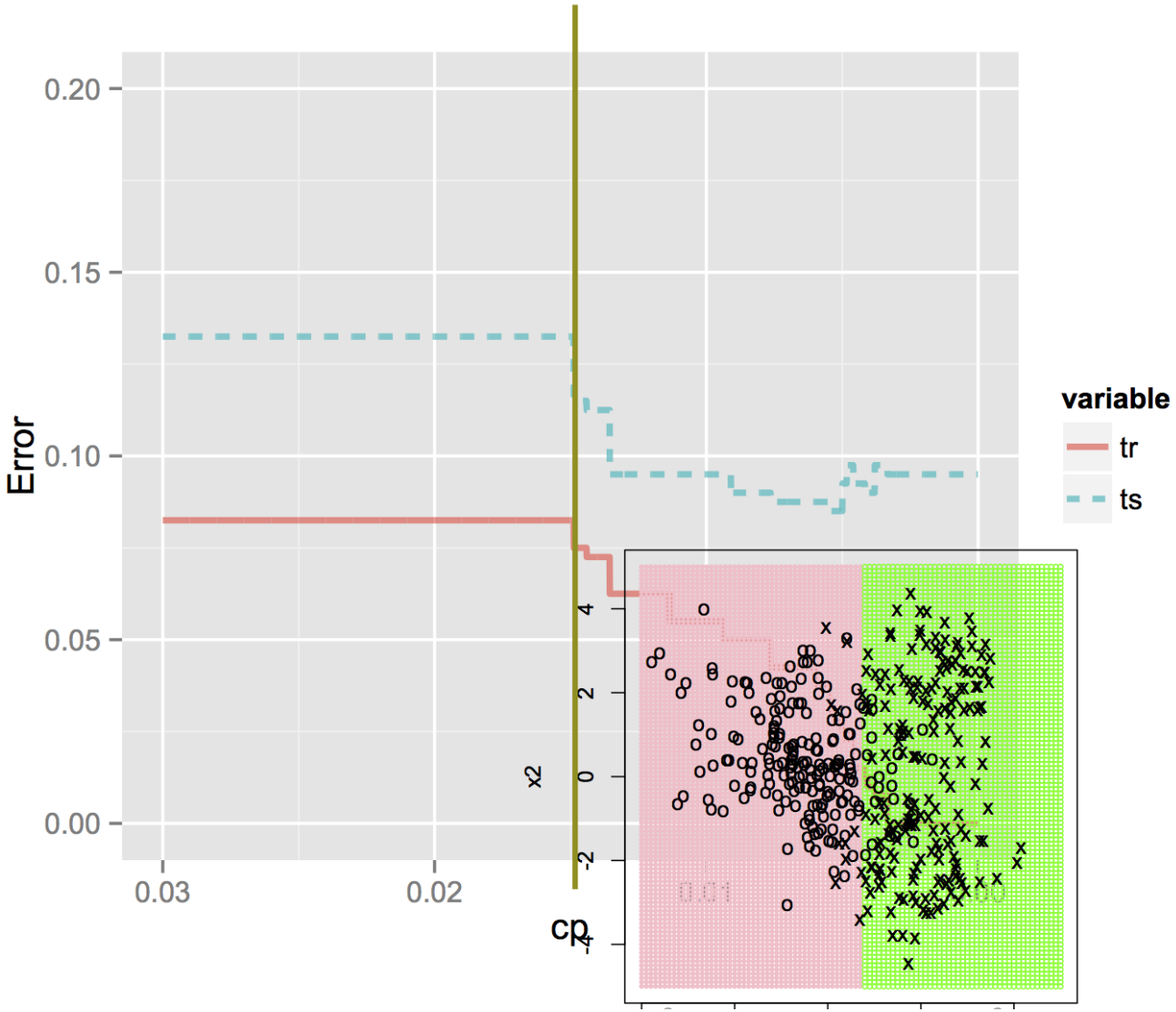
```
> df.rp <- rpart(y~., data=df,
+                control=rpart.control(minsplit=2, cp=0.001))
```



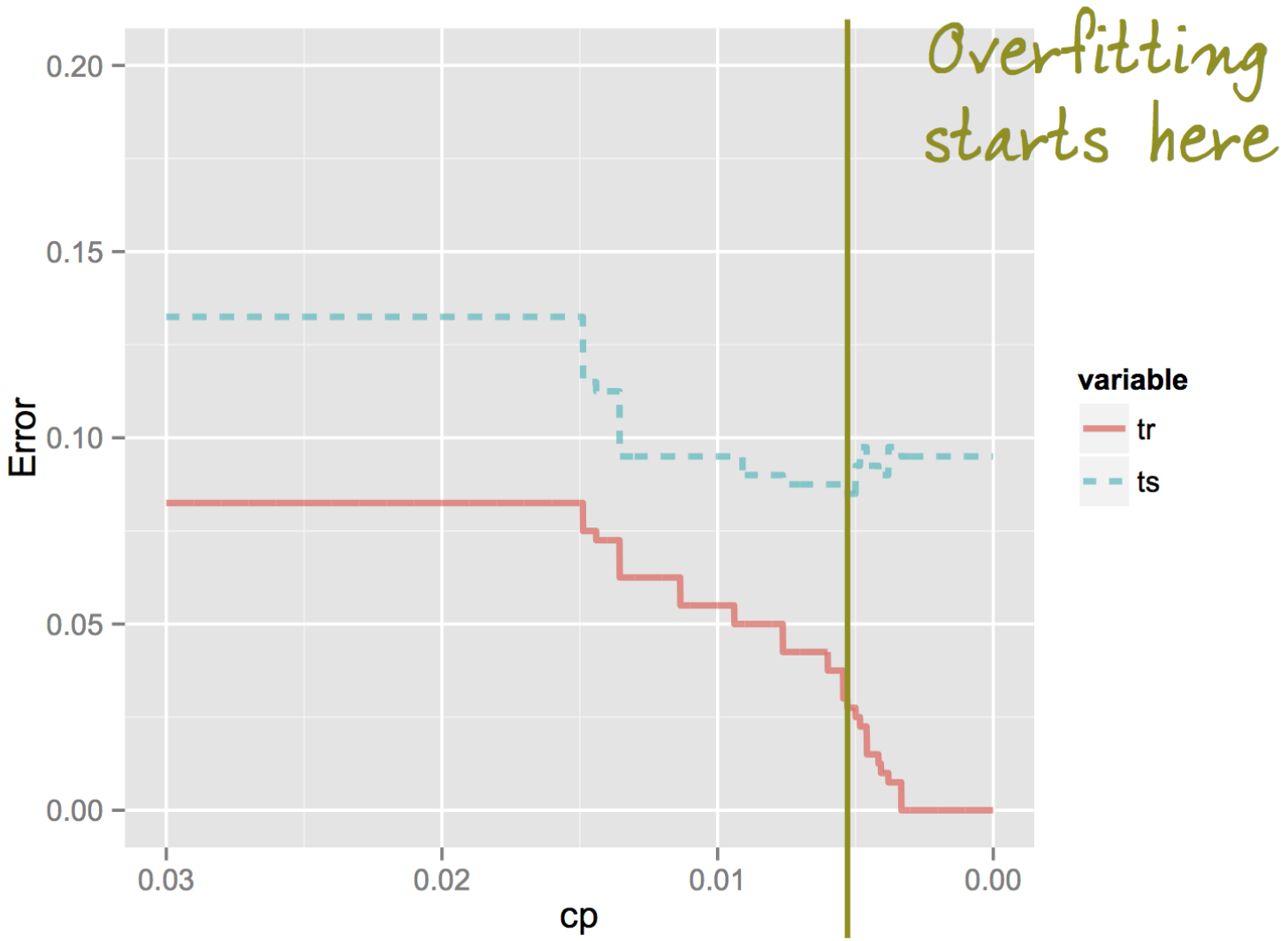
Overfitting example



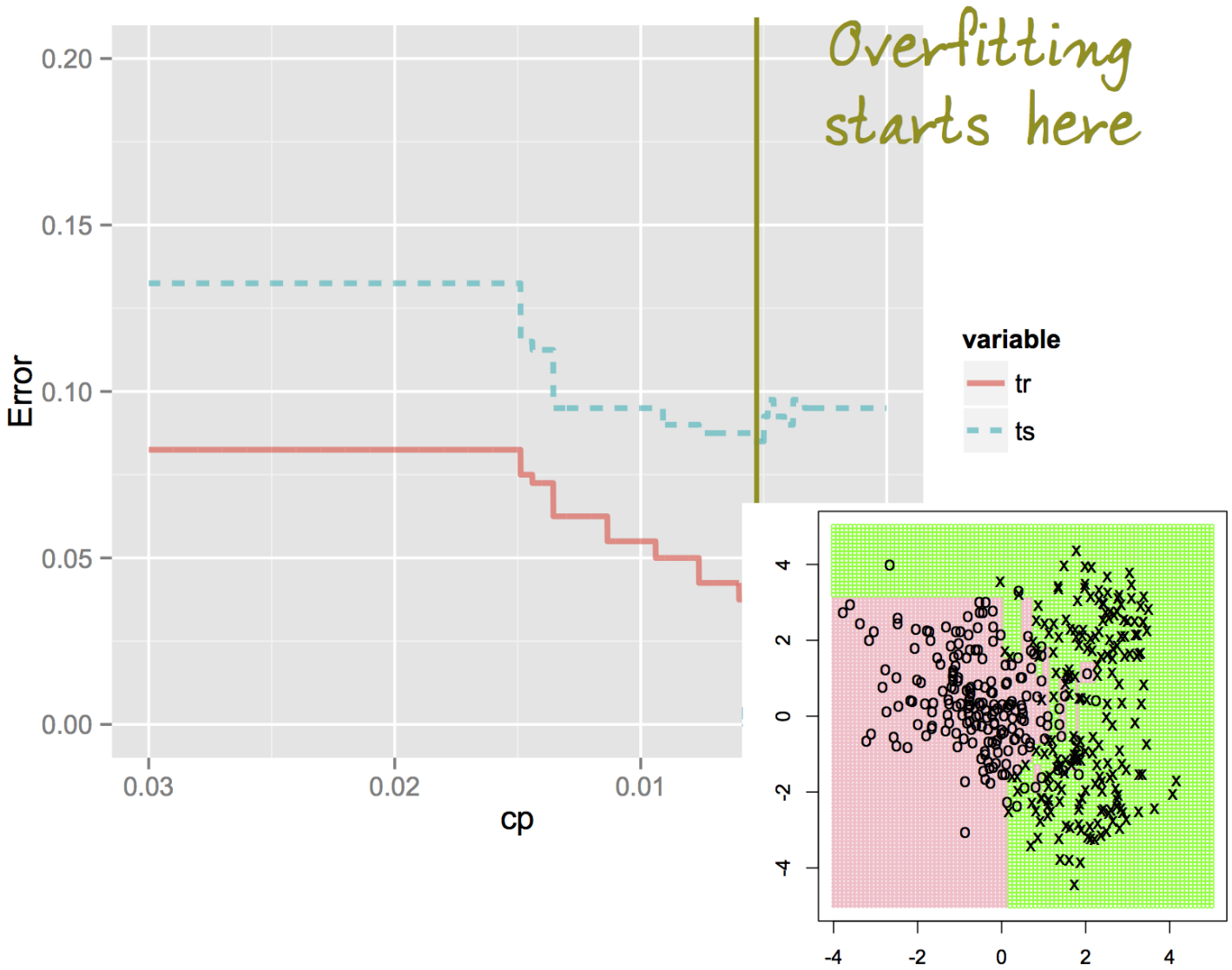
Overfitting example



Overfitting example



Overfitting example



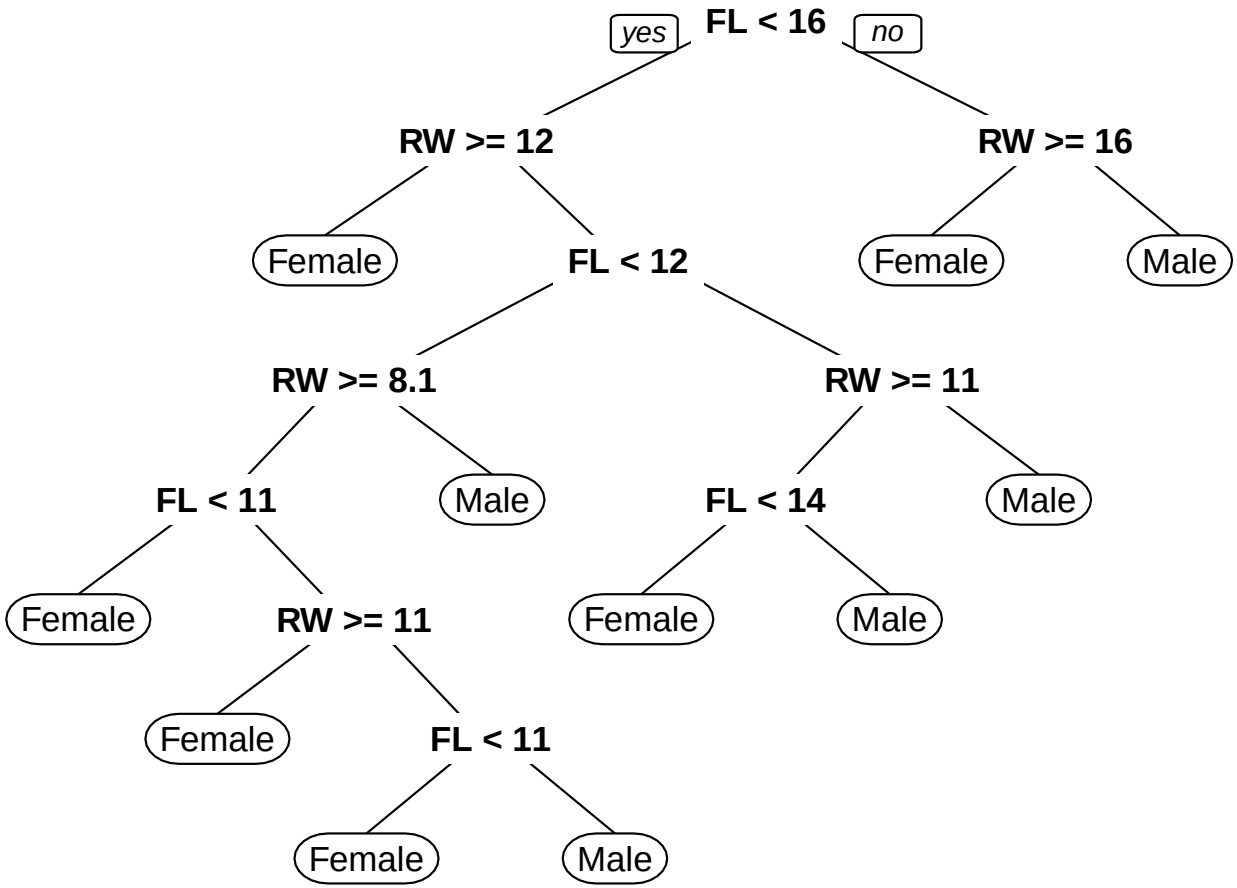
Pruning

- Grow an overly complex tree
- Prune back the weakest branches, using the cost complexity, or cross-validation to get the lowest validation error

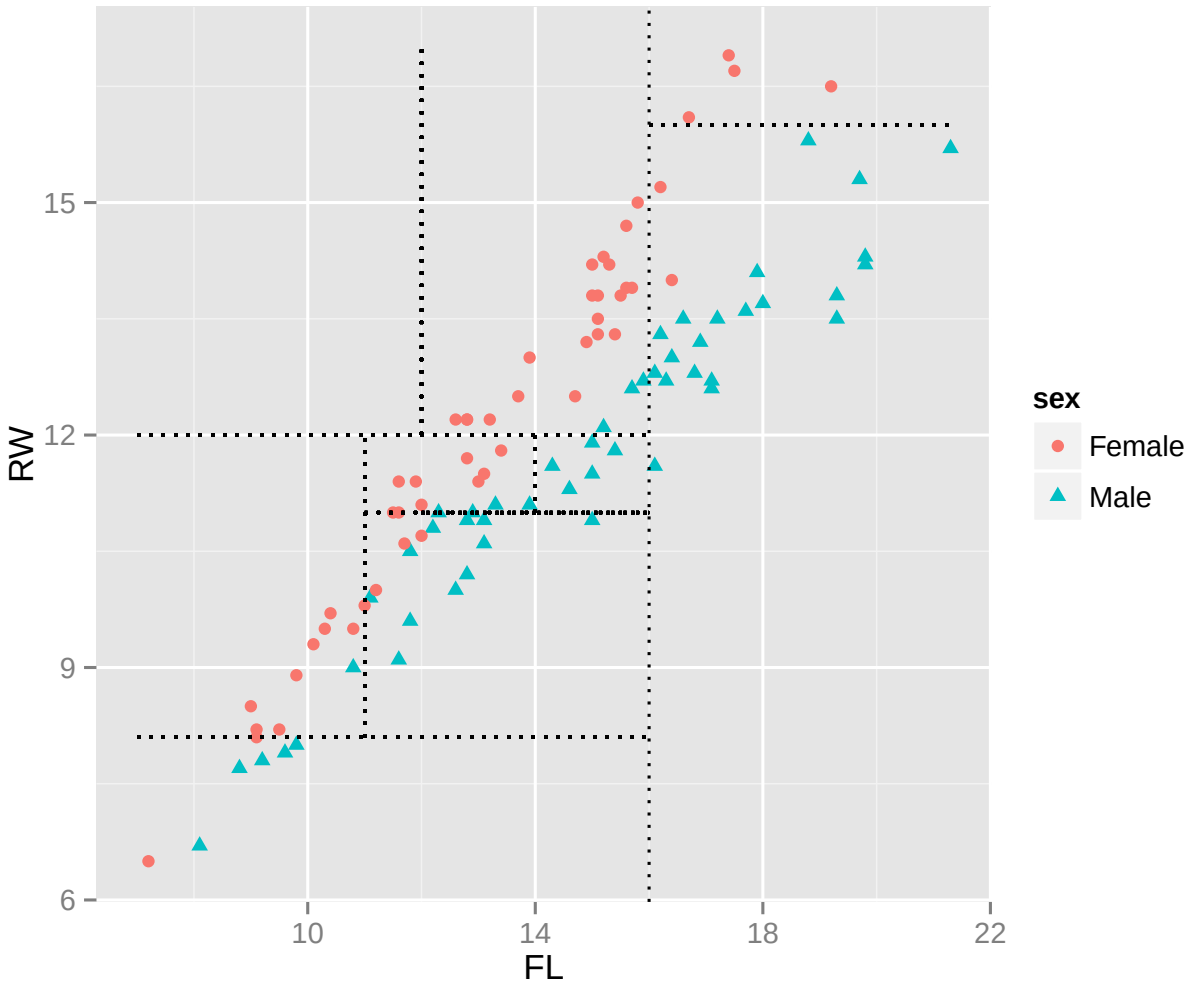
Example: crabs

```
## n= 100
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##   1) root 100 50 Female (0.500 0.500)
##     2) FL< 16 72 28 Female (0.611 0.389)
##       4) RW>=12 22  1 Female (0.955 0.045) *
##       5) RW< 12 50 23 Male (0.460 0.540)
##        10) FL< 12 29 10 Female (0.655 0.345)
##          20) RW>=8.1 23  5 Female (0.783 0.217)
##            40) FL< 11 8  0 Female (1.000 0.000) *
##            41) FL>=11 15  5 Female (0.667 0.333)
##              82) RW>=11 7  0 Female (1.000 0.000) *
##              83) RW< 11 8  3 Male (0.375 0.625)
##               166) FL< 11 5  2 Female (0.600 0.400) *
##               167) FL>=11 3  0 Male (0.000 1.000) *
```

Example: crabs

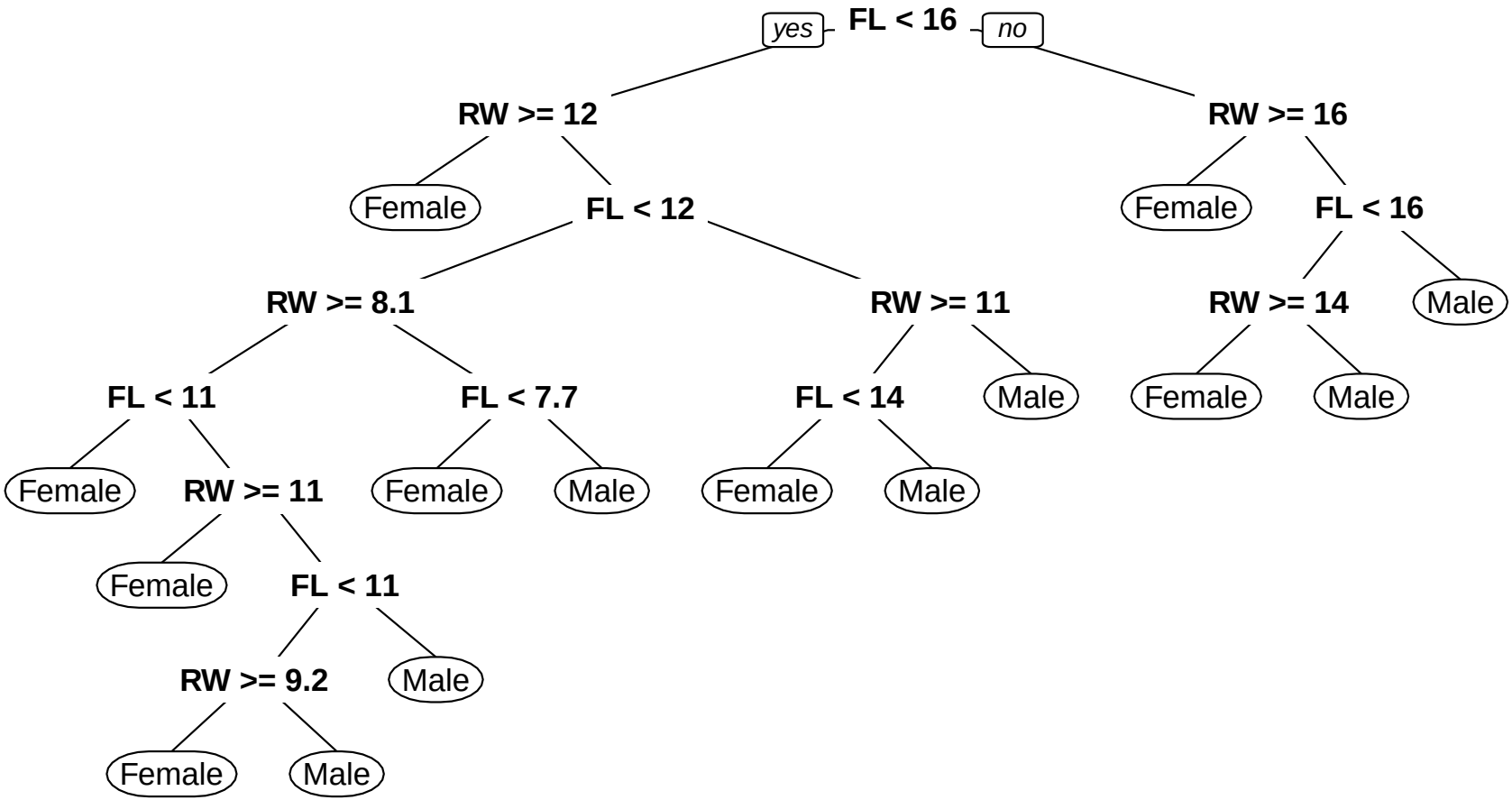


Example: crabs



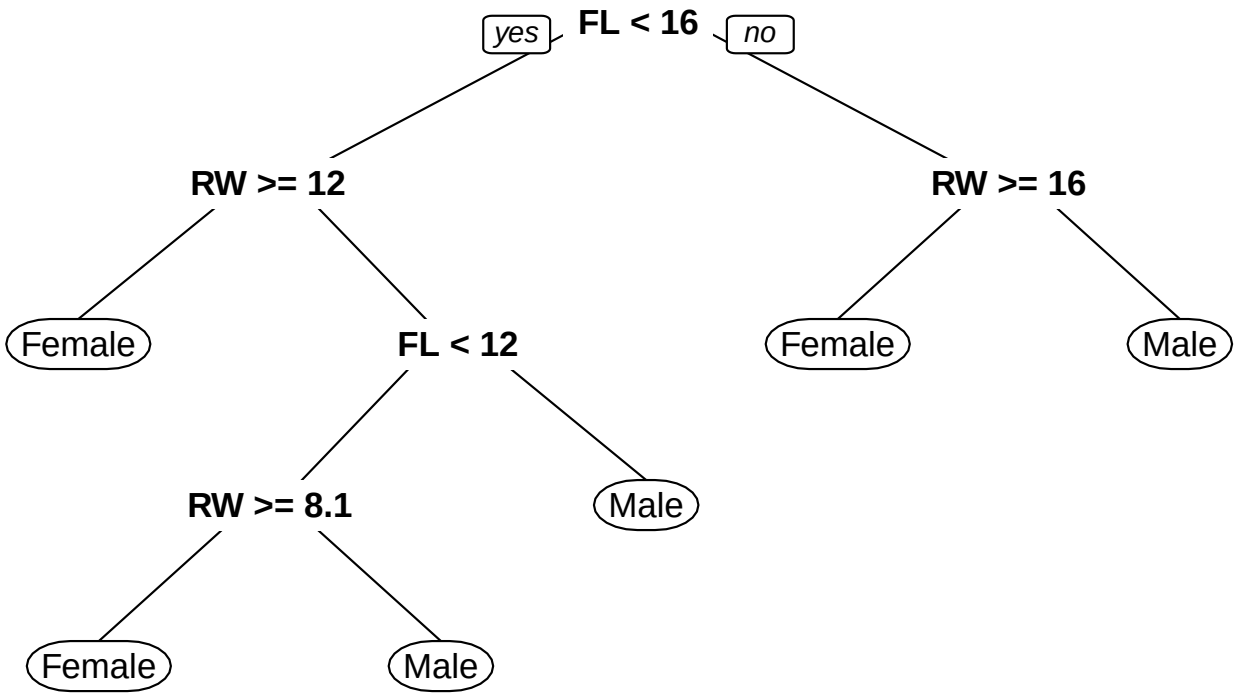
Example: crabs

Complex tree



Example: crabs

Pruned tree



Advantages and disadvantages

- The decision rules provided by trees are very easy to explain, and follow. A simple classification model.
- Trees can handle a mix of predictor types, categorical, quantitative, ...
- Trees efficiently operate when there are missing values in the predictors.
-
- Algorithm is greedy, a better final solution might be obtained by taking a second best split earlier
- When separation is in linear combinations of variables trees struggle to provide a good classification

Random forests