Ch8 - Decision Tree

Contents

8A Subsection

A.1 Basics
A.2 Decision Tree
A.3 Tree Growing
A.4 Tree Pruning
A.5 Pruning
A.6 Pruning
A.7 Classification Trees
A.8 Tree vs Linear Models
A.9 Bagging
A.10 Large Variance
A.11 Random Forests
A.12 Boosting
A.13 Boston Data
A.14 party Package

Textbook: James et al. ISLR 2ed.

8A Subsection

[ToC]

A.1 Basics

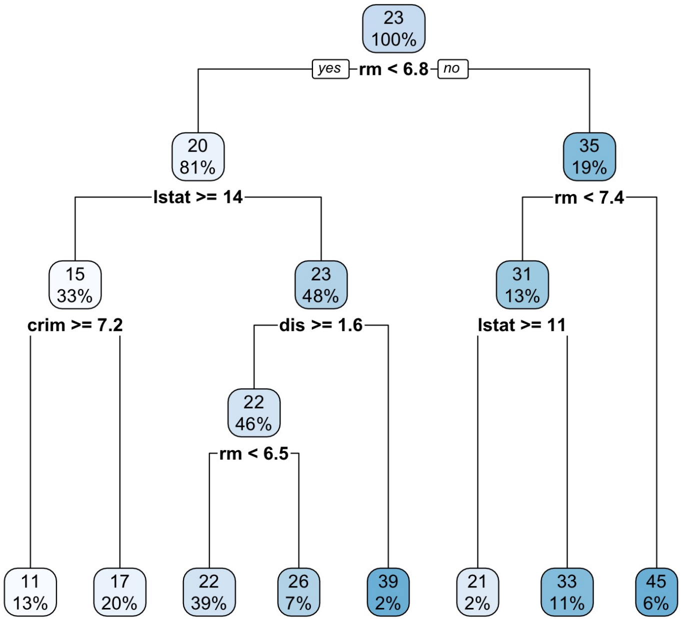
Boston Data (400 training, 96 test)

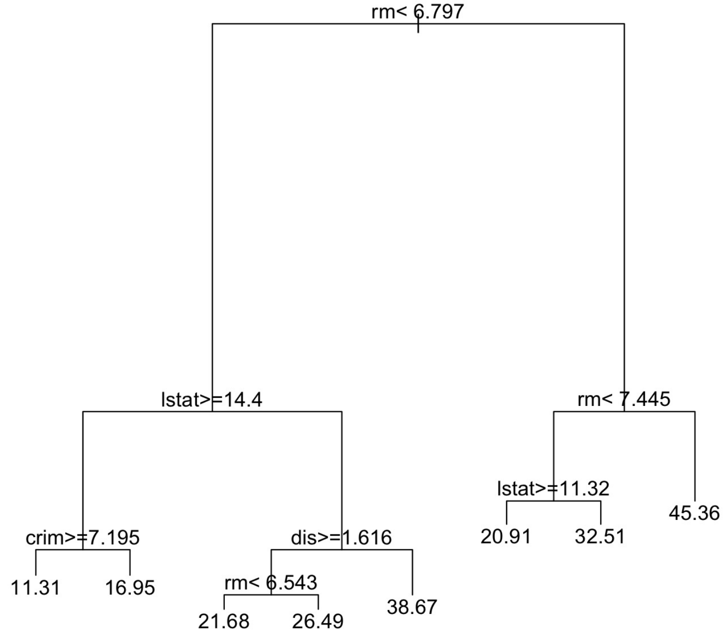
```
Reg01 = lm(medv~., data=Train.set)
> summary(Reg01)

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  35.023724   6.198554   5.650 3.11e-08 ***
crim         -0.115721   0.037334  -3.100 0.002080 **
zn            0.041696   0.016893   2.468 0.014013 *
indus        -0.013844   0.072338  -0.191 0.848329
chas          2.666581   1.005233   2.653 0.008315 **
nox         -16.880242   4.557814  -3.704 0.000244 ***
rm            3.928494   0.498349   7.883 3.30e-14 ***
age          -0.002590   0.015995  -0.162 0.871431
```

```
dis          -1.514405   0.237707  -6.371 5.35e-10 ***
rad           0.296458   0.075489   3.927 0.000102 ***
tax          -0.010113   0.004365  -2.317 0.021054 *
ptratio      -0.925880   0.150607  -6.148 1.96e-09 ***
black         0.008767   0.003380   2.594 0.009849 **
lstat        -0.520376   0.060880  -8.548 2.97e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.958 on 386 degrees of freedom
Multiple R-squared:  0.7303,Adjusted R-squared:  0.7212
F-statistic: 80.41 on 13 and 386 DF,  p-value: < 2.2e-16
```


23
100%
yes
rm < 6.8
no
20
81%
lstat >= 14
35
19%
rm < 7.4
15
33%
crim >= 7.2
23
48%
dis >= 1.6
31
13%
lstat >= 11
22
46%
rm < 6.5
11
13%
17
20%
22
39%
26
7%
39
2%
21
2%
33
11%
45
6%



rm< 6.797
lstat>=14.4
rm< 7.445
lstat>=11.32
45.36
crim>=7.195
dis>=1.616
20.91
32.51
11.31
16.95
rm< 6.543
38.67
21.68
26.49

A.2 Decision Tree

- Can be used in Classification / Regression
- Root Node / Parent Node / Child Node
- Terminal Nodes / Leaves
- Every observation in the terminal node gets same prediction (mean of region)
- Regression Tree: minimize RSS

$$\sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2$$

- Classification Tree: 3 choices for the measure.

- Classification Error Rate
- Gini Index
- Entropy (Cross-Entropy) Gini and Entropy are numerically similar.

- Recursive Binary Splitting: Can't go through all the possibilities; Use Top-Down approach.

A.3 Tree Growing

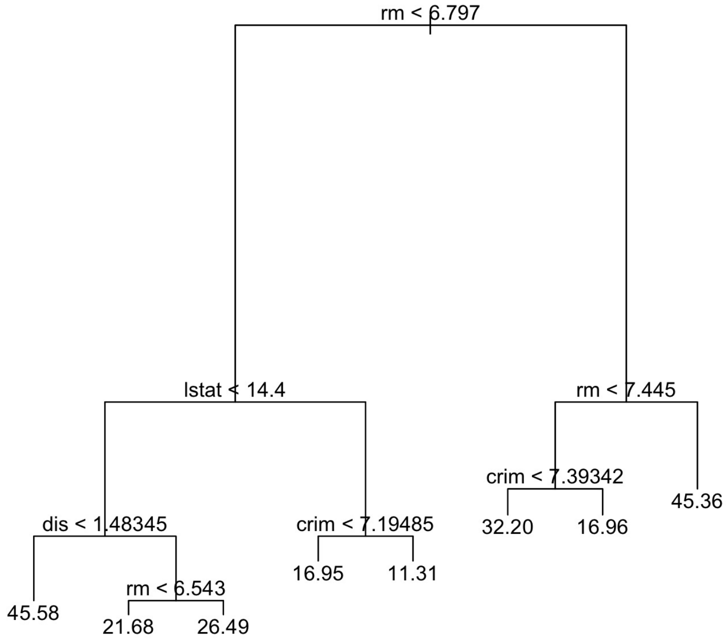
Recursive Binary Splitting

1. Pick the 1st feature, and go through all possible splitting points s, calculating (RSS/Entropy) at each point.

2. Go throgh all featurs, pick the feature that gives min (RSS/Entropy). That's the best feature to be split at.

3. Repeat. (feature that was used for previous split is still in the pool)

```
#1  crim     per capita crime rate by town.
#2  zn       proportion of residential land zoned for lots over 25,000 sq.ft.
#3  indus    proportion of non-retail business acres per town.
#4  chas     Charles River dummy variable (= 1 if tract bounds river; 0 otherwise).
#5  nox      nitrogen oxides concentration (parts per 10 million).
#6  rm       average number of rooms per dwelling.
#7  age      proportion of owner-occupied units built prior to 1940.
#8  dis      weighted mean of distances to five Boston employment centres.
#9  rad      index of accessibility to radial highways.
#10 tax      full-value property-tax rate per $10,000.
#11 ptratio  pupil-teacher ratio by town.
#12 black    1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town.
#13 lstat    lower status of the population (percent).
#14 medv     median value of owner-occupied homes in \$1000s.
```


rm < 6.797
lstat < 14.4
rm < 7.445
crim < 7.39342
45.36
dis < 1.48345
crim < 7.19485
32.20
16.96
16.95
11.31
rm < 6.543
45.58
21.68
26.49

A.4 Tree Pruning

- Seemingly worthless split may lead to large reduction in RSS later on. (Grow more)
- Growing is likely to overfit, because the resulting tree might be too complex. (Grow less)

- A smaller tree with fewer splits might lead to lower variance and better interpretation at the cost of a little bias.
- Grow a large tree (stop only when each terminal node has fewer than some min num of obs. Then prune later.

A.5 Pruning

- Grow a large tree, then try to select a subtree that leads to the lowest TEST error rate.
- Given a subtree, we can estimate its test error using CV.
- Since there can be too many subtrees, cost complexity pruning selects a small set of subtrees for consideration. (AKA weakest link pruning).
- Rather than considering every possible subtree, we consider a sequence of trees indexed by a nonnegative tuning parameter α.

A.6 Pruning

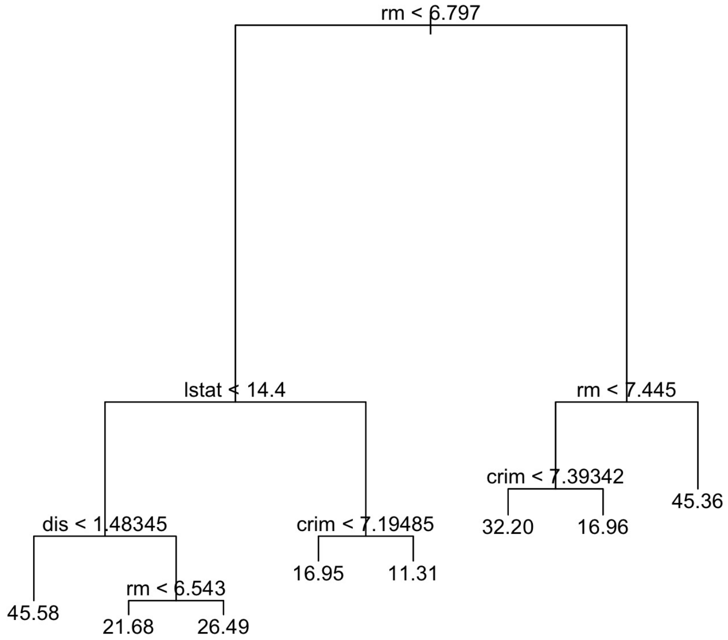
- For each value of α there corresponds a subtree $T \subset T0$ such that

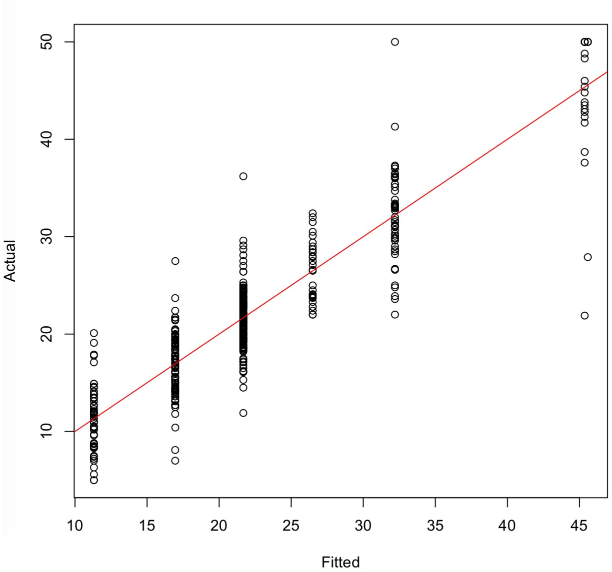
$$\sum_{m=1}^{|T|} \sum_{x_i \in R_m} (y_i - \hat{y}_{Rm})^2 + \alpha|T|$$

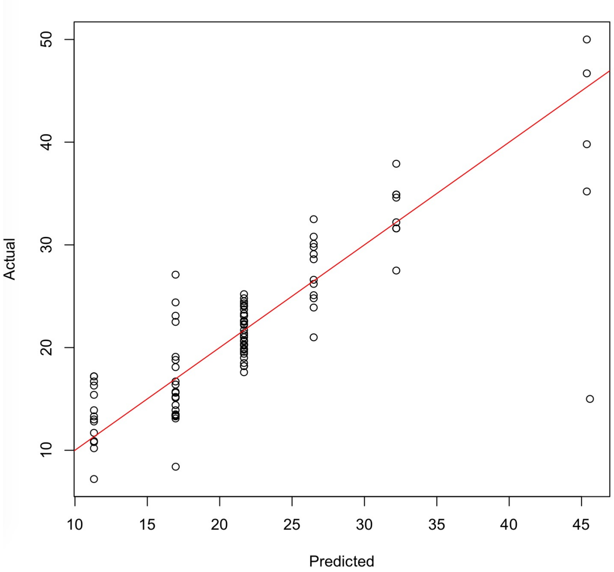
  is as small as possible.

- $|T|$ is the number of terminal nodes, Rm is the rectangle (i.e. the subset of predictor space) corresponding to the mth terminal node.

- Use CV to choose best α. (Algorithm 8.1 on p309)


rm < 6.797
lstat < 14.4
rm < 7.445
crim < 7.39342
45.36
dis < 1.48345
crim < 7.19485
32.20
16.96
16.95
11.31
rm < 6.543
45.58
21.68
26.49


Actual
Fitted
10
15
20
25
30
35
40
45
50

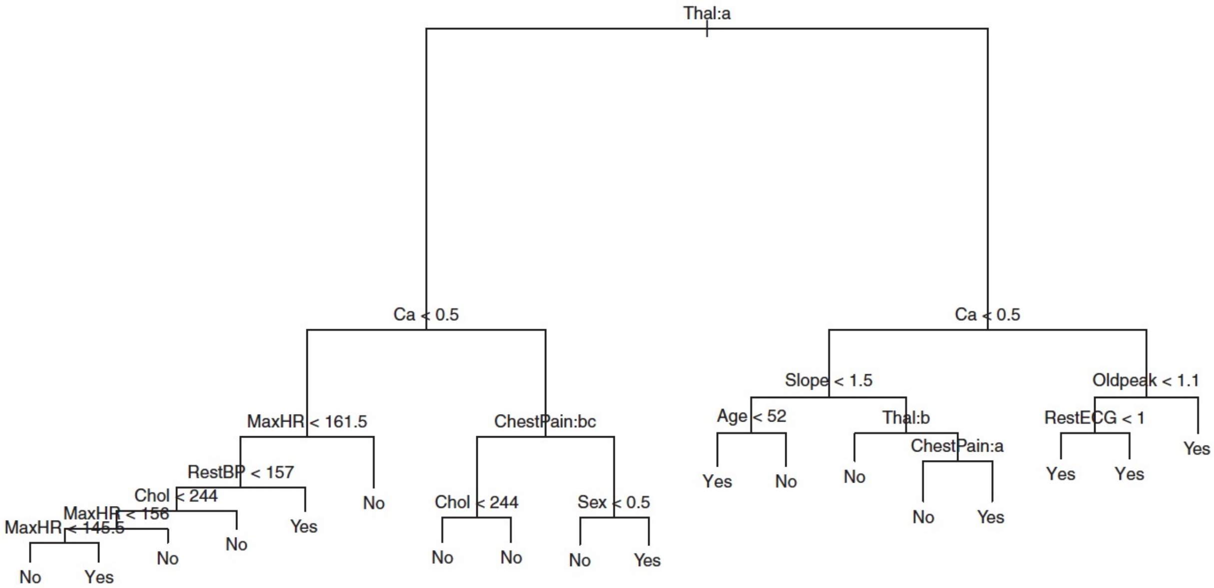

Actual
Predicted
10
15
20
25
30
35
40
45
50

Test.RSS

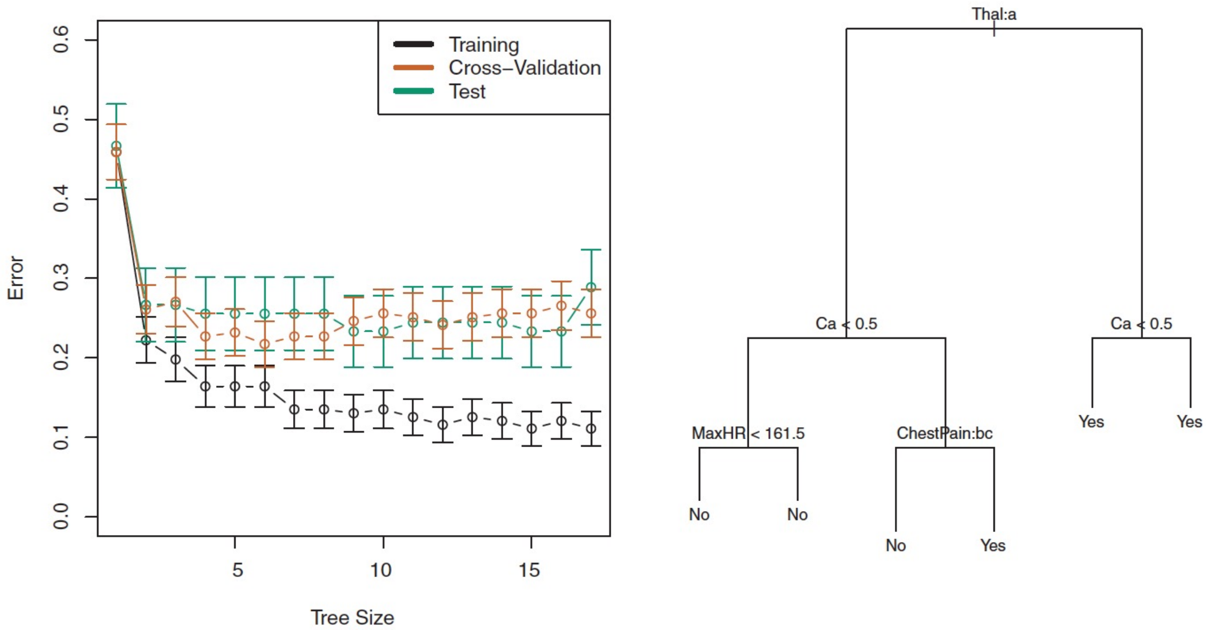
```
      RMSE   Rsquare
1 4.946511 0.6353647
```

A.7 Classification Trees

Heart data



Thal:a
Ca < 0.5
Ca < 0.5
MaxHR < 161.5
ChestPain:bc
Slope < 1.5
Oldpeak < 1.1
Age < 52
Thal:b
ChestPain:a
RestECG < 1
RestBP < 157
Chol < 244
MaxHR < 156
MaxHR < 145.5
Chol < 244
Sex < 0.5
No
Yes
No
No
Yes
No
No
No
No
Yes
Yes
No
No
No
Yes
Yes
Yes
Yes



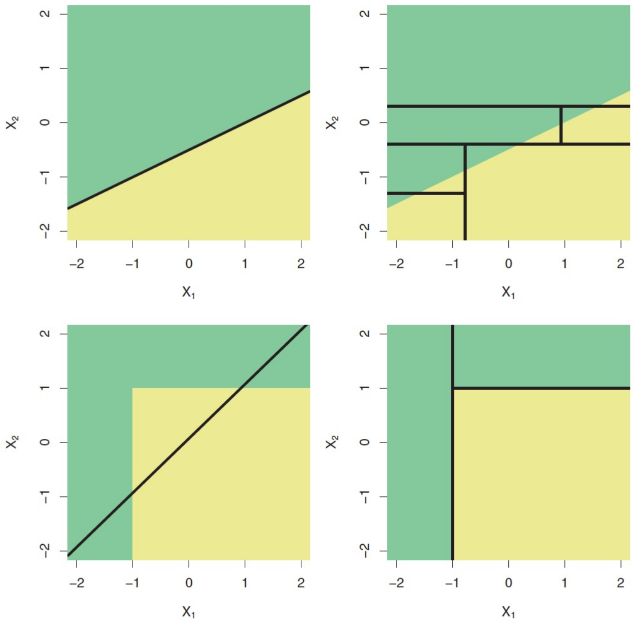
Training
Cross−Validation
Test
Error
Tree Size
0.0
0.1
0.2
0.3
0.4
0.5
0.6
5
10
15
Thal:a
Ca < 0.5
Ca < 0.5
MaxHR < 161.5
ChestPain:bc
Yes
Yes
No
No
No
Yes

A.8 Tree vs Linear Models

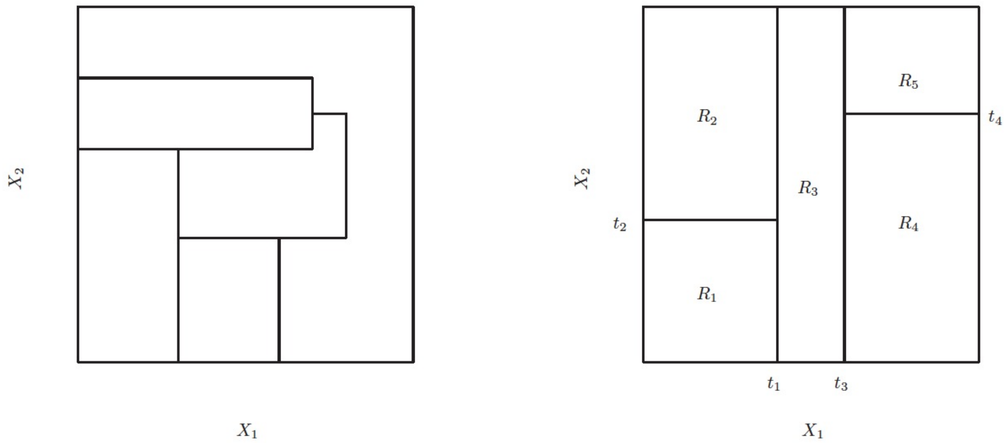
- $$f(X) = \beta_0 + \sum_{j=1}^{p} X_j \beta_j$$

  $$f(X) = \sum_{m=1}^{M} c_m \cdot I(x \in R_m)$$

- Trees are easy to interpret
- Trees may resemble how humans make decisions
- Trees can handle qualitative predictors easier than linear models
- However, trees in general does not have comparable predictive power as other models.



X_1
X_2
−2
−1
0
1
2

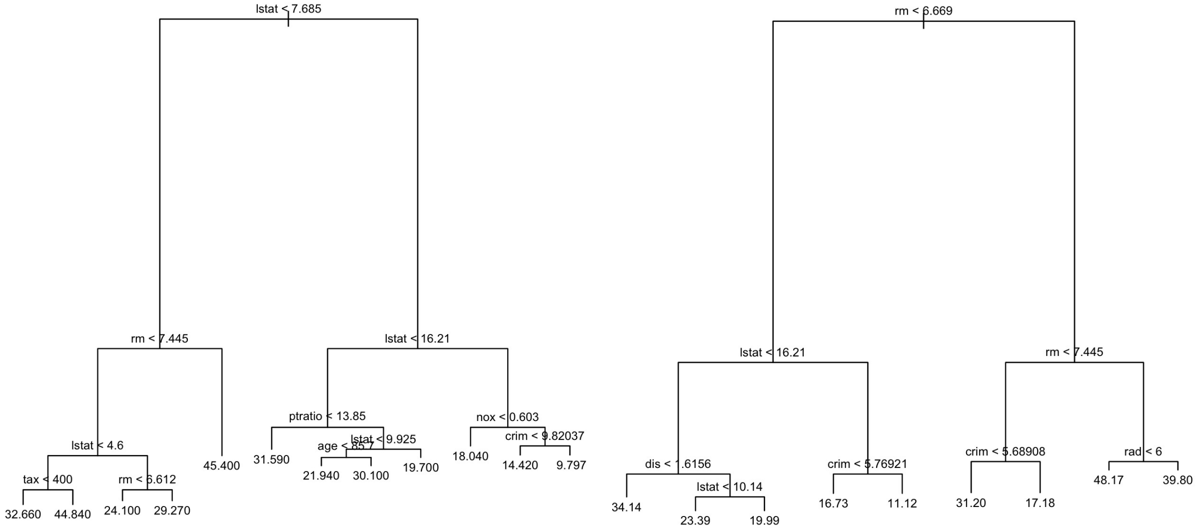

X_2
X_1
R_1
R_2
R_3
R_4
R_5
t_1
t_2
t_3
t_4
X_2
X_1

A.9 Bagging

- Trees have too large of variance
- Bagging = Bootstrap aggregation
- can be used outside of DT, but drastically improves DT.
- Use majority class rule for classification prediction
- may cause DT to lose the interpretability (no more Tree diagram)
- Record RSS decrease due to a single split, average over all bootstrapped samples.
- Out-of-Bag Error Estimation (Estimate Test error of Bagged model)
- When the training set for the current tree is drawn by sampling with replacement, about one-third of the cases are left out of the sample. $(e^{-1} = \lim(1 - 1/n)^n)$

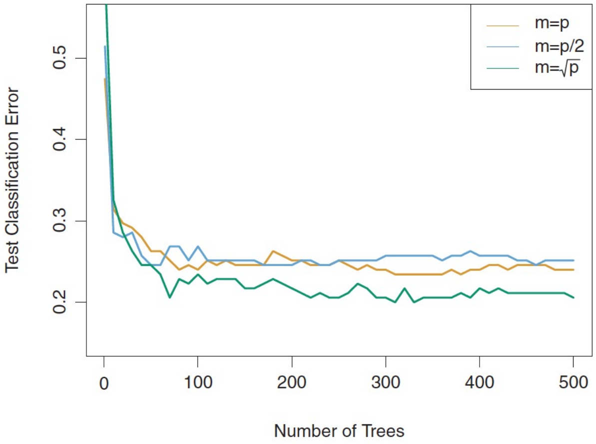
A.10 Large Variance

From Boston data CV1 and CV2


lstat < 7.685
rm < 7.445
lstat < 16.21
lstat < 4.6
tax < 400
rm < 6.612
32.660
44.840
24.100
29.270
45.400
ptratio < 13.85
31.590
lstat < 9.925
age < 85.7
21.940
30.100
19.700
nox < 0.603
18.040
crim < 9.82037
14.420
9.797
rm < 6.669
lstat < 16.21
dis < 1.6156
34.14
lstat < 10.14
23.39
19.99
crim < 5.76921
16.73
11.12
rm < 7.445
crim < 5.68908
31.20
17.18
rad < 6
48.17
39.80

A.11 Random Forests

- Like Bagging, but de-correlates
- Each time split occurs, only random sample of m predictor can be considered. Choose $m = \sqrt{p}$.



m=p
m=p/2
m=√p
Test Classification Error
0.2
0.3
0.4
0.5
0
100
200
300
400
500
Number of Trees

A.12 Boosting

- Set $\hat{f}(x) = 0$ and $r_i = y_i$ for all i in the training set.
- for $b = 1, 2, \ldots, B$ repeat:
  1. Fit a tree $\hat{f}^b$ with d splits (d=depth) ($d+1$ terminal nodes) to the training data (X, r).
  2. Update $\hat{f}$ by adding in a shrunken version of the new tree:

$$\hat{(}x) \leftarrow \hat{f}(x) + \lambda \hat{f}^b(x)$$

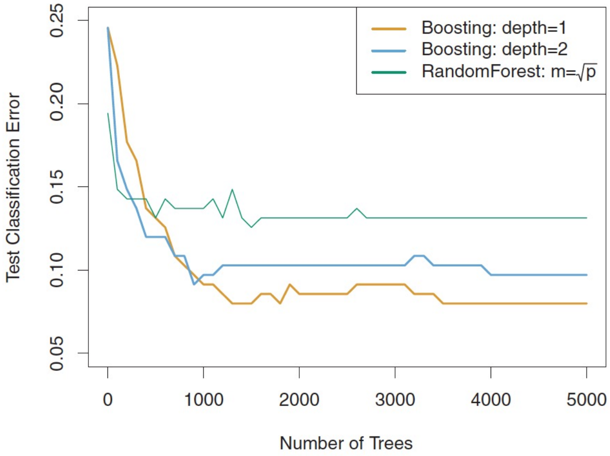
  3. Updata the residuals,

$$r_i \leftarrow r_i - \lambda \hat{f}^b(x_i)$$

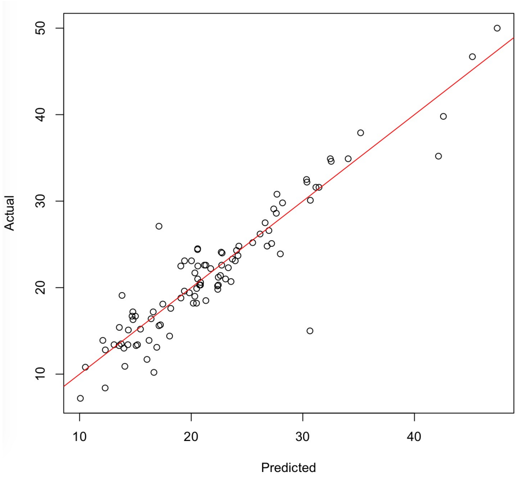
- Output the boosted model

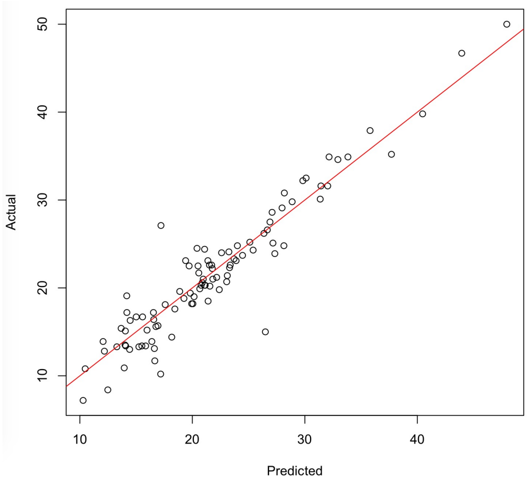
$$\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x)$$

- Boosting Learns SLOWLY.



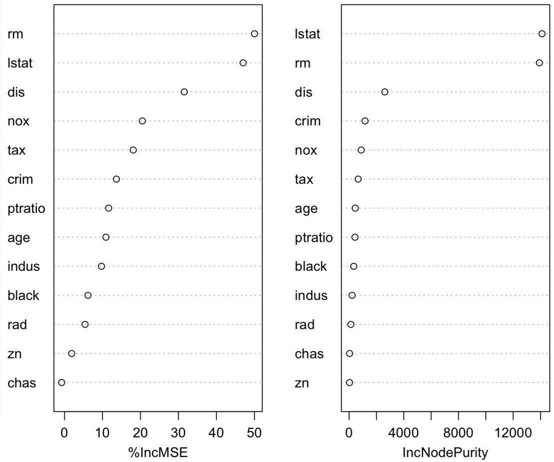
A.13 Boston Data





```
Bagging (m=13)
Test.RSS
         RMSE   Rsquare
medv 2.891555 0.8635642
```

```
Random Forest (m=6)
Test.RSS
         RMSE   Rsquare
medv 2.569209 0.8931857
```


rm
lstat
dis
nox
tax
crim
ptratio
age
indus
black
rad
zn
chas
0
10
20
30
40
50
%IncMSE
lstat
rm
dis
crim
nox
tax
age
ptratio
black
indus
rad
chas
zn
0
4000
8000
12000
IncNodePurity

A.14 party Package

```
---------------
https://ademos.people.uic.edu/Chapter24.html

R package 'party::ctree()'

Uses ID3 Algorithm (Iterative Dichotomiser 3) created by JR Quinlan

  wiki: https://en.wikipedia.org/wiki/ID3_algorithm
    (note Examples -> Observations)

  Uses criteria of min entropy to grow trees. Does not prune.

  Some source say it looks at Information Gain,
  but max(IG) iff min(entropy), some other say.
```

IG formula is on Wiki.

Stopping criteria is one of following 3:

- every element in the subset belongs to the same class;
- there are no more attributes to be selected
- there are no examples in the subset,
  which happens when no example in the parent set was found to match a specific value of the selected attribute.
  An example could be the absence of a person among the population with age over 100 years.