Supervised Machine Learning and Learning Theory

Lecture 8: Decision trees and bagging

October 1, 2024

Warm-up questions

- Could you write down the cross-entropy loss?
- What is the pros and cons of forward stepwise selection vs. best subset selection?
- Could you write down the objective of ridge regression in dimension p (i.e., assume the input features are of dimension p)?
- Following up the above question, can you write down the objective of LASSO?

Case study

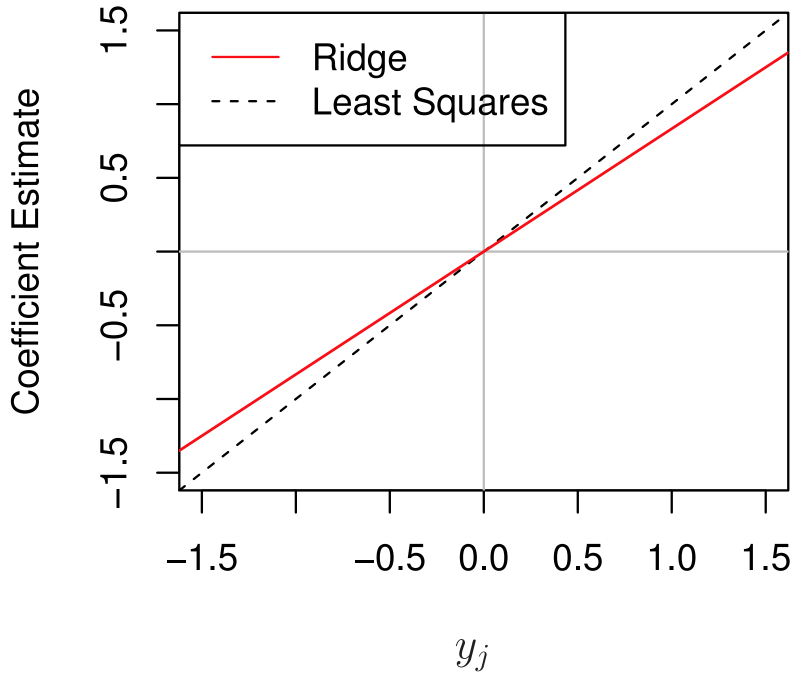
- Suppose $n = p$ and the predictors are $X = \mathrm{Id}_{p \times p}$

- **Linear regression:** minimizes $\sum_{j=1}^{p} (y_j - \beta_j)^2$
  - Solution: $\hat{\beta}_j = y_j$

- **Ridge regression:** minimizes $\sum_{j=1}^{p} (y_j - \beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$
  - Solve β_j by minimizing $(y_j - \beta_j)^2 + \lambda \beta_j^2$
  - Solution: $\hat{\beta}_{j,\lambda}^{R} = \frac{y_j}{1+\lambda}$

Shrinkage via Ridge

- **Linear regression coefficients**: $\hat{\beta}_j = y_j$
- **Ridge regression coefficients**: $\hat{\beta}^R_{j,\lambda} = \frac{y_j}{1+\lambda}$
- **Interpretation**: Shrinks $\hat{\beta}_j$ by $\frac{1}{1+\lambda}$



Why LASSO shrinks model coefficients to zero

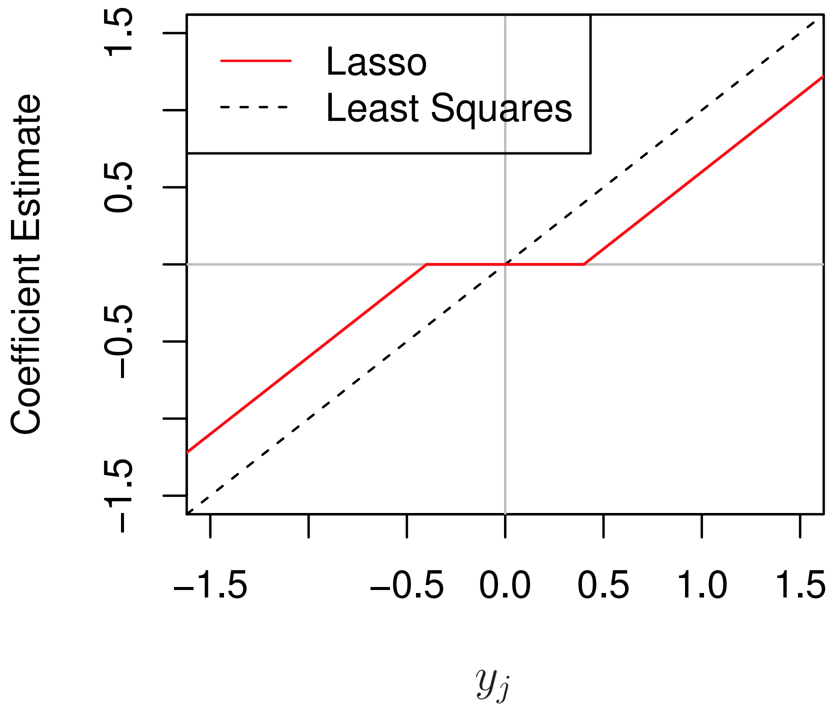
- **Case study**: Suppose $n = p$ and matrix of predictors is $X = Identity$
- **Linear regression**: minimizes $\sum_{j=1}^{p}(y_j - \beta_j)^2$
  - Solution: $\hat{\beta}_j = y_j$
- **LASSO**: minimizes $\sum_{j=1}^{p}(y_j - \beta_j)^2 + \lambda \sum_{j=1}^{p}|\beta_j|$
  - Solve β_j by minimizing $(y_j - \beta_j)^2 + \lambda|\beta_j|$

$$\hat{\beta}_{j,\lambda}^{L} = \begin{cases} y_j - \lambda/2 & \text{if } y_j > \lambda/2 \\ y_j + \lambda/2 & \text{if } y_j < -\lambda/2 \\ 0 & \text{if } |y_j| < \lambda/2 \end{cases}$$

Why LASSO shrinks model coefficients to zero

- **Linear regression solution:** $\hat{\beta}_j = y_j$
- **LASSO coefficients**

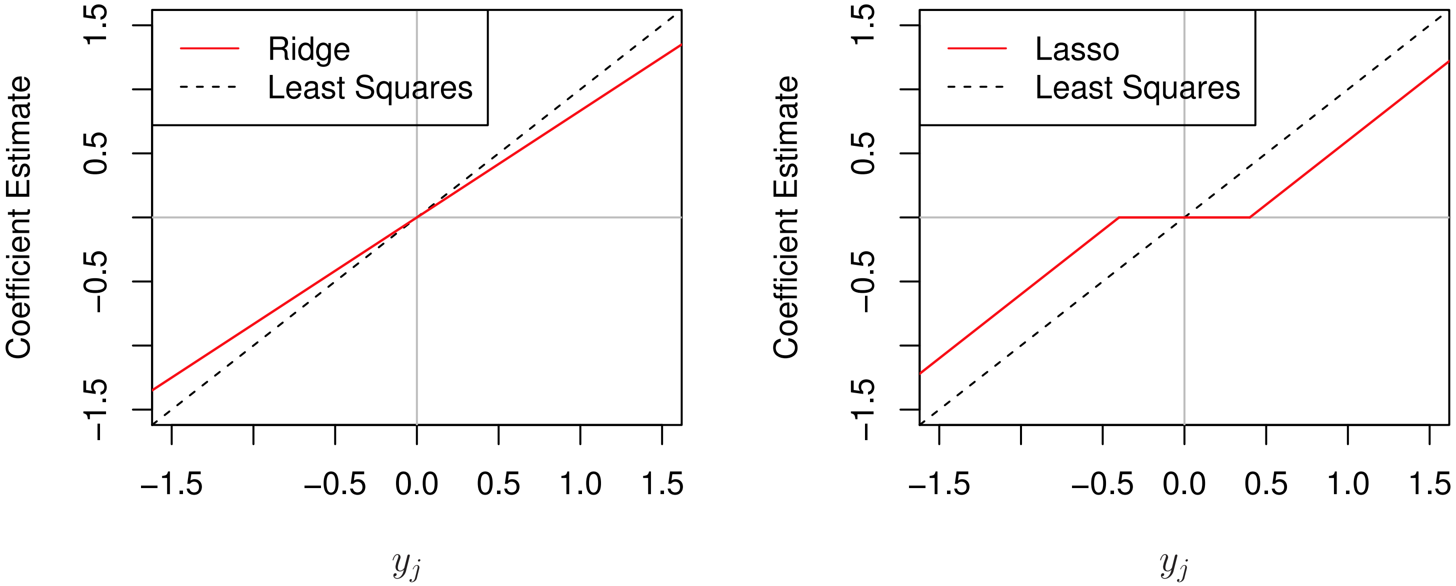
$$\hat{\beta}_\lambda^L = \begin{cases} y_j - \lambda/2 & \text{if } y_j > \lambda/2 \\ y_j + \lambda/2 & \text{if } y_j < -\lambda/2 \\ 0 & \text{if } |y_j| < \lambda/2 \end{cases}$$



- **Interpretation:** Hard thresholding

Comparing Ridge and LASSO

- **Ridge regression**: Shrinks by the same proportion $\hat{\beta}_\lambda^R = \frac{y_j}{1+\lambda}$
- **LASSO**: Hard thresholding at $\frac{\lambda}{2}$, otherwise reduce by $\frac{\lambda}{2}$



Lecture plan

- **Elastic net**

Elastic net

- Elastic net combines LASSO with ridge, and minimizes

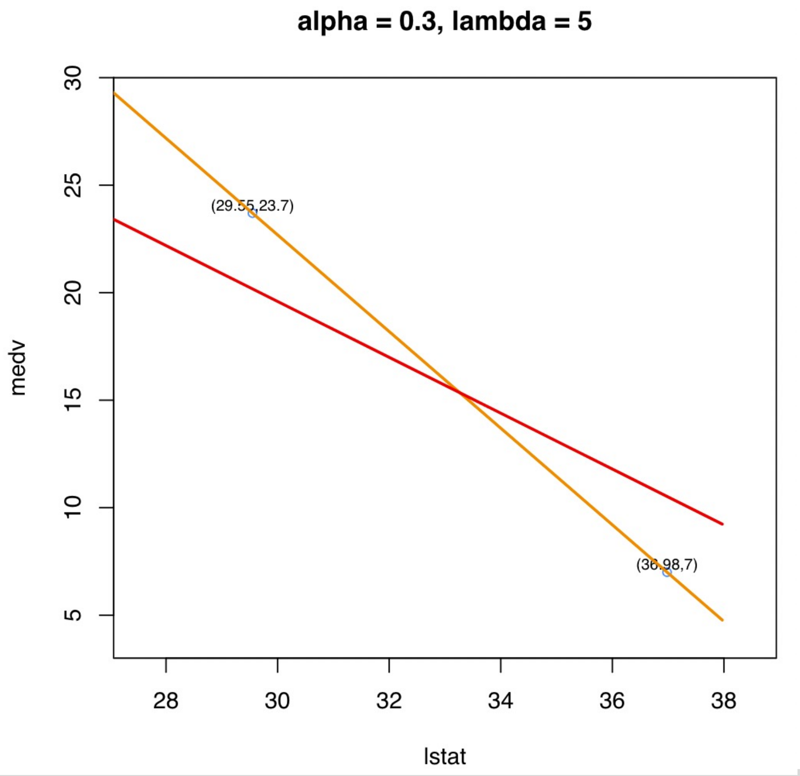
$$\sum_{i=1}^{n}(medv_i - \beta_0 - lstat_i \cdot \beta_1)^2 + \lambda \cdot (1-\alpha) \cdot \frac{\beta_1^2}{2} + \lambda \cdot \alpha \cdot |\beta_1|$$

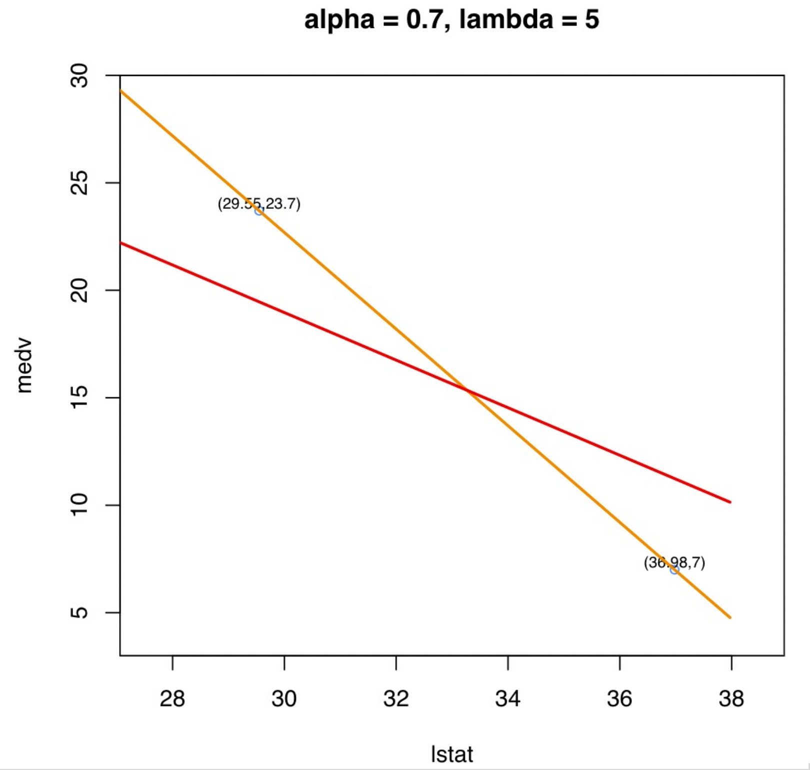
  - $\lambda \geq 0$: tuning hyper-parameter
  - $\alpha \in [0,1]$: tuning hyper-parameter
    - $\alpha = 0$: ridge
    - $\alpha = 1$: LASSO

Role of α and λ in elastic net

$$\sum_{i=1}^{n}(medv_i - \beta_0 - lstat_i \cdot \beta_1)^2 + \lambda \cdot (1-\alpha) \cdot \frac{\beta_1^2}{2} + \lambda \cdot \alpha \cdot |\beta_1|$$

- $\alpha = 0.3, \lambda = 5$: $\hat{\beta}_1^E = -1.299$; $\alpha = 0.7, \lambda = 5$: $\hat{\beta}_1^E = -1.107$

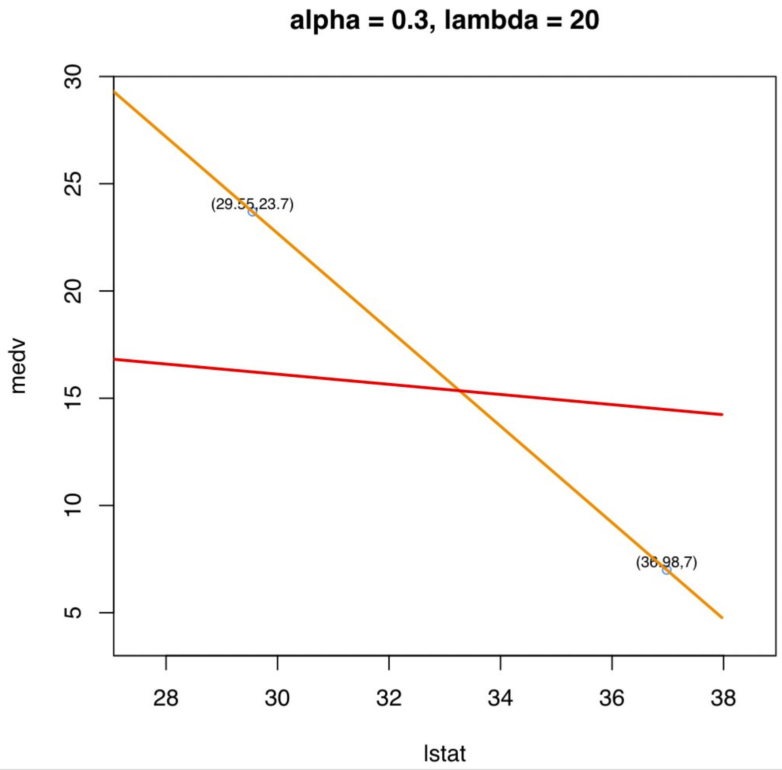


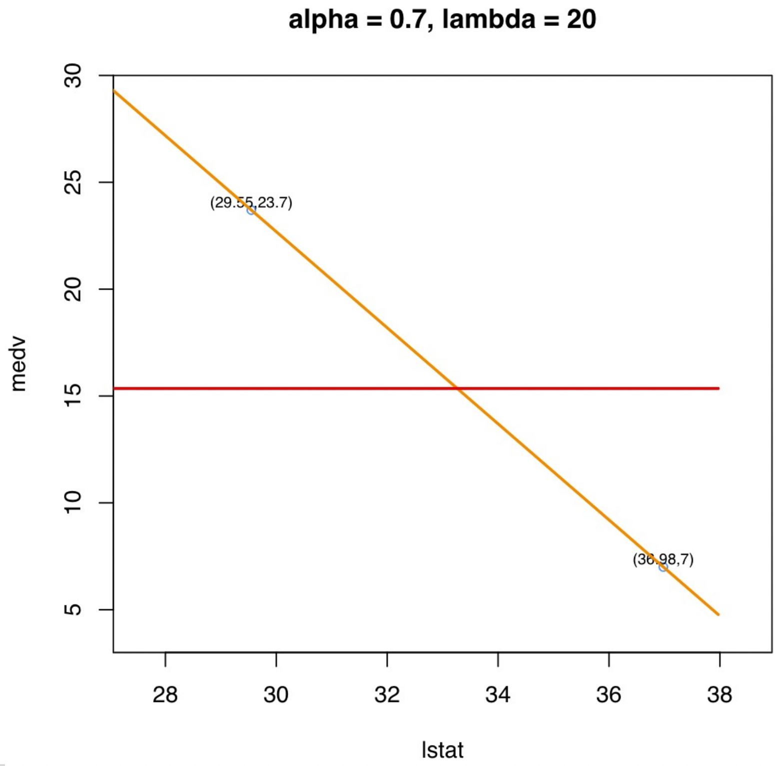


Role of α and λ in elastic net

$$\sum_{i=1}^{n}(medv_i - \beta_0 - lstat_i \cdot \beta_1)^2 + \lambda \cdot (1-\alpha) \cdot \frac{\beta_1^2}{2} + \lambda \cdot \alpha \cdot |\beta_1|$$

- $\alpha = 0.3, \lambda = 20$: $\hat{\beta}_1^E = -0.236$; $\alpha = 0.7, \lambda = 20$: $\hat{\beta}_1^E = 0$





Choose α and λ by cross-validation

- The procedure is the same for ridge and LASSO

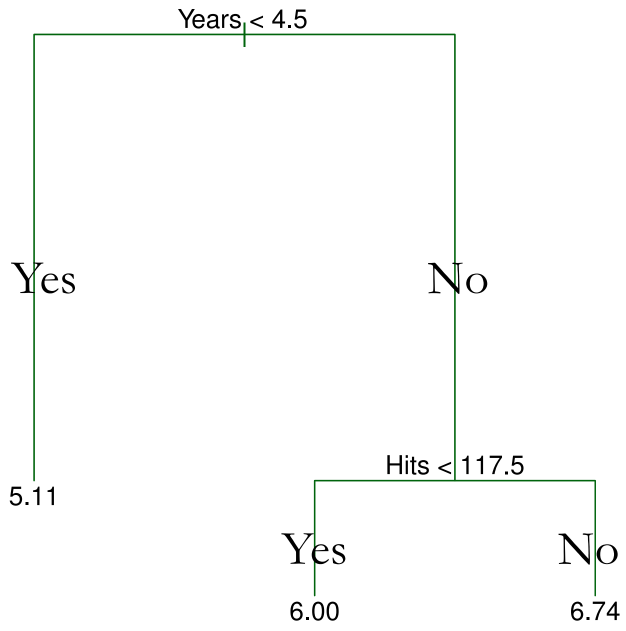
1. Choose a grid of α values and a grid of λ values
2. Compute the cross-validation error for each (α, λ) value
3. Select the (α, λ) with the smallest cross-validation error
4. Refit the model using all observations and selected (α, λ)

Lecture plan

- **Regression tree**

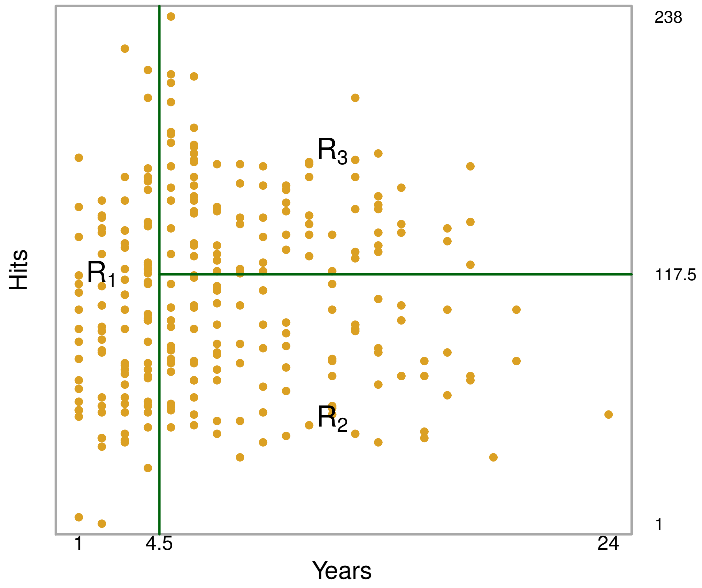
Example

- **Example**: predict a baseball player's salary Y_i
  - Years: The number of years played in the league
  - Hit: The number of hits made in the previous year
- **Regression tree** consists of a series of splitting rules
  - $\text{Years}_i < 4.5$: predicted salary $\hat{Y}_i = 5.11$
  - $\text{Years}_i \geq 4.5$ & $\text{Hits}_i < 117.5$: predicted salary $\hat{Y}_i = 6.00$
  - $\text{Years}_i \geq 4.5$ & $\text{Hits}_i \geq 117.5$: predicted salary $\hat{Y}_i = 6.74$



Example

- **Regression tree** segments the feature space to disjoint regions



$$R_1 = \{X | \text{Years}_i < 4.5\}$$

$$R_2 = \{X | \text{Years}_i \geq 4.5, \text{Hits}_i < 117.5\}$$

$$R_3 = \{X | \text{Years}_i \geq 4.5, \text{Hits}_i \geq 117.5\}$$

How to build a decision tree?

Two main steps

1. Partition the feature space into J **distinct and non-overlapping** regions, $R_1, R_2, \cdots, R_J$
2. Make the **same** prediction for every observation in region R_j: Mean of the training observations in R_j

Example: $(\text{Years}_i, \text{Hits}_i, Y_i)$

- Alan: (14, 81, 6.16)
- Al: (2, 37, 4.25)
- Andres: (2, 81, 4.32)
- Bill: (18, 168, 6.66)
- Brian: (14, 137, 6.80)
- Bob: (7, 49, 5.70)

How to build a decision tree?

Example

- Alan: (14, 81, 6.16)
- Al: (2, 37, 4.25)
- Andres: (2, 81, 4.32)
- Bill: (18, 168, 6.66)
- Brian: (14, 137, 6.80)
- Bob: (7, 49, 5.70)

$R_1 = \{X \mid \text{Years}_i < 4.5\}$ $\hat{Y}_{R_1} = \frac{4.25 + 4.32}{2}$

$R_2 = \{X \mid \text{Years}_i \geq 4.5, \text{Hits}_i < 117.5\}$ $\hat{Y}_{R_2} = \frac{6.16 + 5.70}{2}$

$R_3 = \{X \mid \text{Years}_i \geq 4.5, \text{Hits}_i \geq 117.5\}$ $\hat{Y}_{R_3} = \frac{6.66 + 6.80}{2}$

How to build a decision tree?

Two main steps

Partition the feature space into J **distinct and non-overlapping** regions, $R_1, R_2, \cdots, R_J$

- Find boxes that minimize the RSS $\sum_{j=1}^{J} \sum_{i \in R_j} \left(y_i - \hat{y}_{R_j}\right)^2$
- $\hat{y}_{R_j}$ is the mean label value for the training observations in R_j

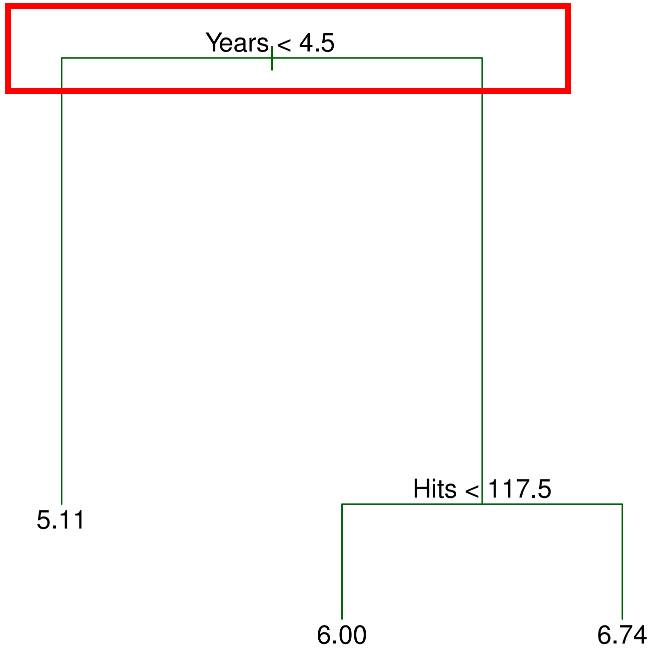
1st cut point

- **Select 1st cut point:** Select a predictor X_j and a cut point s
  - Define the pair of half-planes $R_1(j,s) = \{X|X_j < s\}$ and $R_2(j,s) = \{X|X_j \geq s\}$ that minimize

$$\sum_{i:x_i \in R_1(j,s)} (y_i - \hat{y}_{R_1})^2 + \sum_{i:x_i \in R_2(j,s)} (y_i - \hat{y}_{R_2})^2$$

$$\sum_i (y_i - \bar{y})^2 = 207.15$$

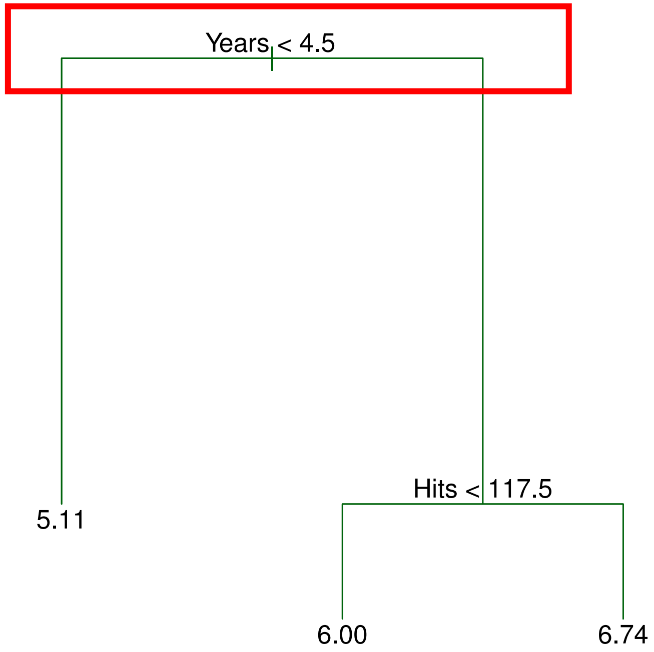
X_j	s	RSS
Year	4	120.18
Year	4.5	115.06
Year	6	133.30
Hits	110	163.75
Hits	120	164.53



Cut rule

Example: This cut point defines two regions

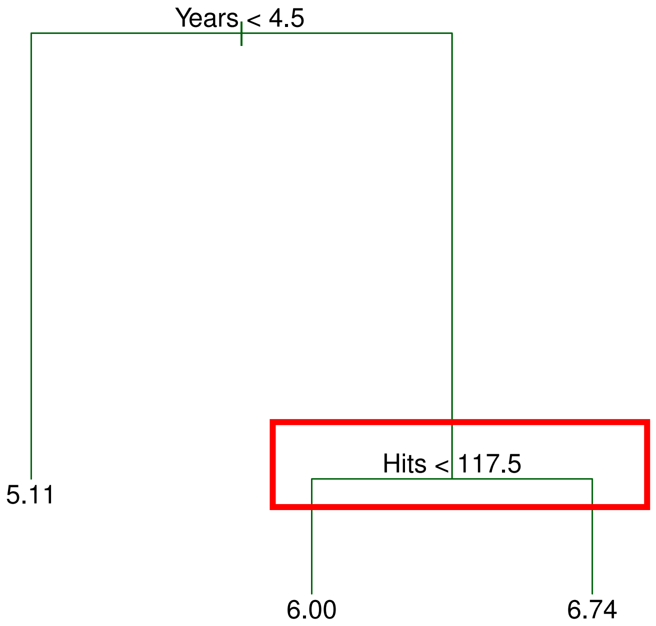
$$R_1 = \{X | \text{Years}_i < 4.5\}$$
$$R_2 = \{X | \text{Years}_i \geq 4.5\}$$



2^{nd} cut point

- **Select 2^{nd} cut point:** Select a region R_k, a predictor X_j and a splitting point s with the criterion $X_j < s$ produces the largest decrease in RSS
- $R_1 = \{X|\text{Years} < 4.5\}$ and $R_2 = \{X|\text{Years} \geq 4.5\}$

R_k	X_j	s	RSS
R_1	Year	3.5	105.85
R_1	Hits	110	107.66
R_1	Hits	120	108.88
R_2	Year	5.5	107.65
R_2	Hits	110	95.91
R_2	Hits	117.5	95.18
R_2	Hits	120	96.23



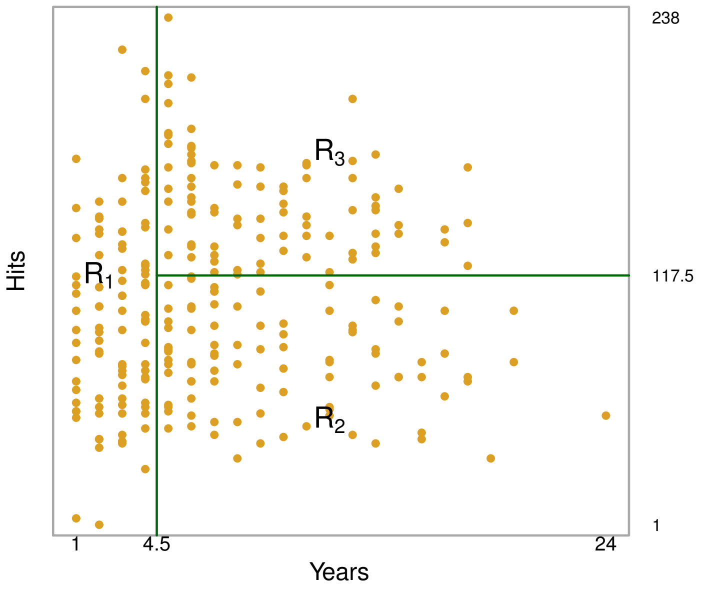
2nd cut rule

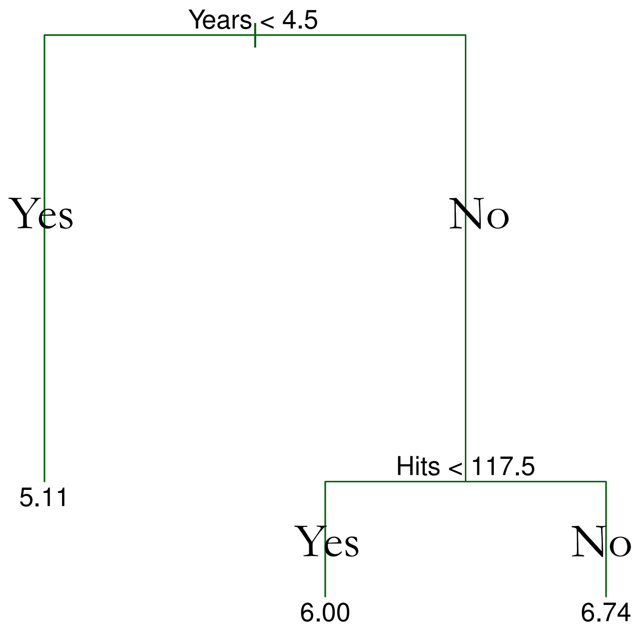
- **Illustration:** Combining both cut points

$$R_1 = \{X | \text{Years}_i < 4.5\}$$

$$R_2 = \{X | \text{Years}_i \geq 4.5, \text{Hits}_i < 117.5\}$$

$$R_3 = \{X | \text{Years}_i \geq 4.5, \text{Hits}_i \geq 117.5\}$$





Binary recursive search

- **Select 3rd cut point**: Repeat the same

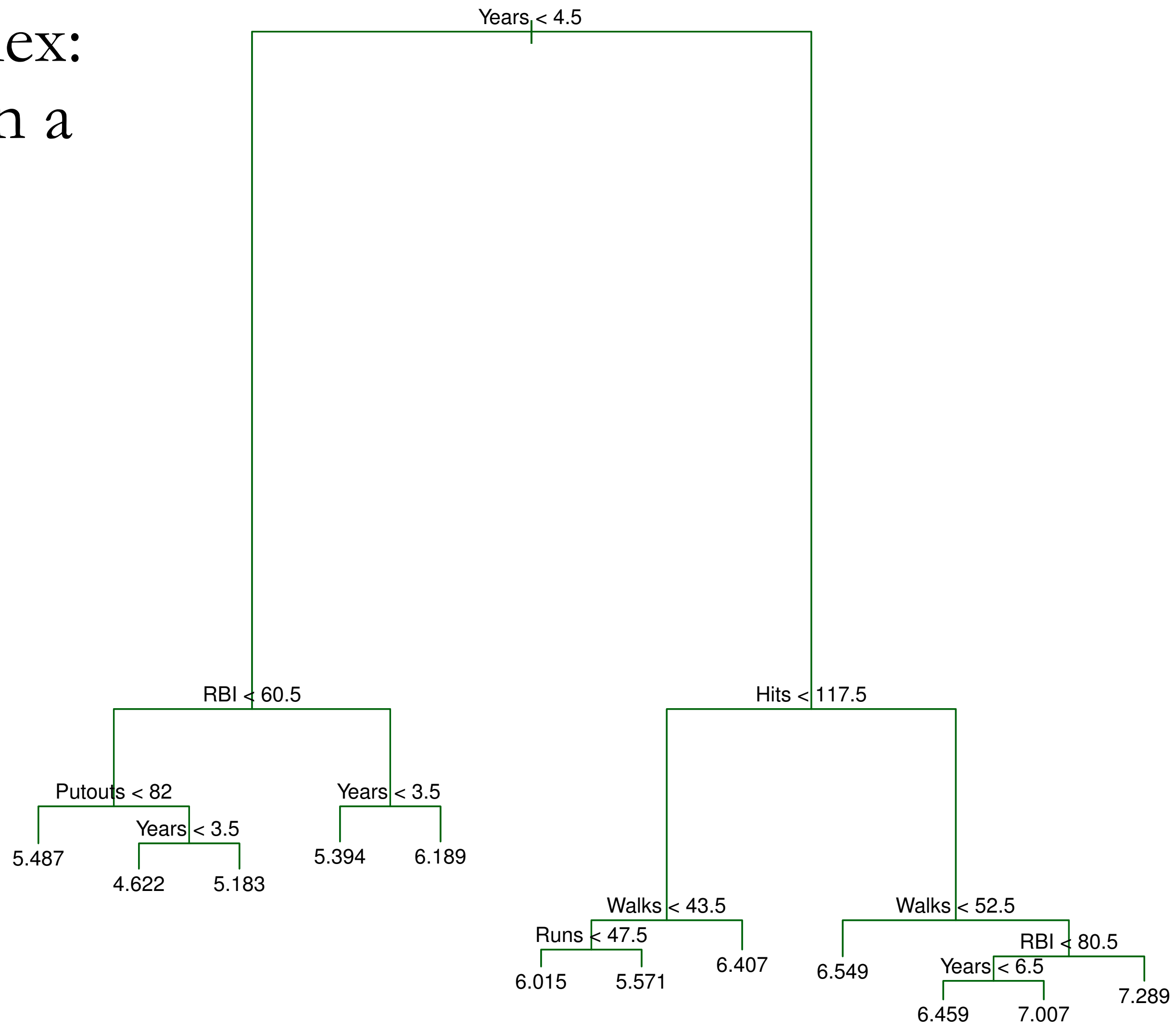
- Select a region R_k, a predictor X_j and a splitting point s, such that splitting R_k with the criterion $X_j < s$ produces the largest decrease in RSS

- …

- **Stopping rule**: Terminate when there are few observations in each region

Overfitting

- The tree might be too complex: A leaf node may only contain a handful of data points



Reduce overfitting

- **Proposed solution**: Add a penalty term to quantify the decision tree's complexity
- **Cost complexity pruning**

$$\min \sum_{j=1}^{|T|} \sum_{i \in R_j} \left(y_i - \hat{y}_{R_j}\right)^2 + \alpha |T|$$

- $|T|$: number of terminal nodes of the tree T

- If α is larger, then $|T|$ tends to be ______
  A. larger
  B. smaller

How do we reduce overfitting?

- Cost complexity pruning

$$\min \sum_{j=1}^{|T|} \sum_{i \in R_j} \left(y_i - \hat{y}_{R_j}\right)^2 + \alpha |T|$$

  - $|T|$: number of terminal nodes of the tree T
  - If α is larger, then $|T|$ tends to be smaller

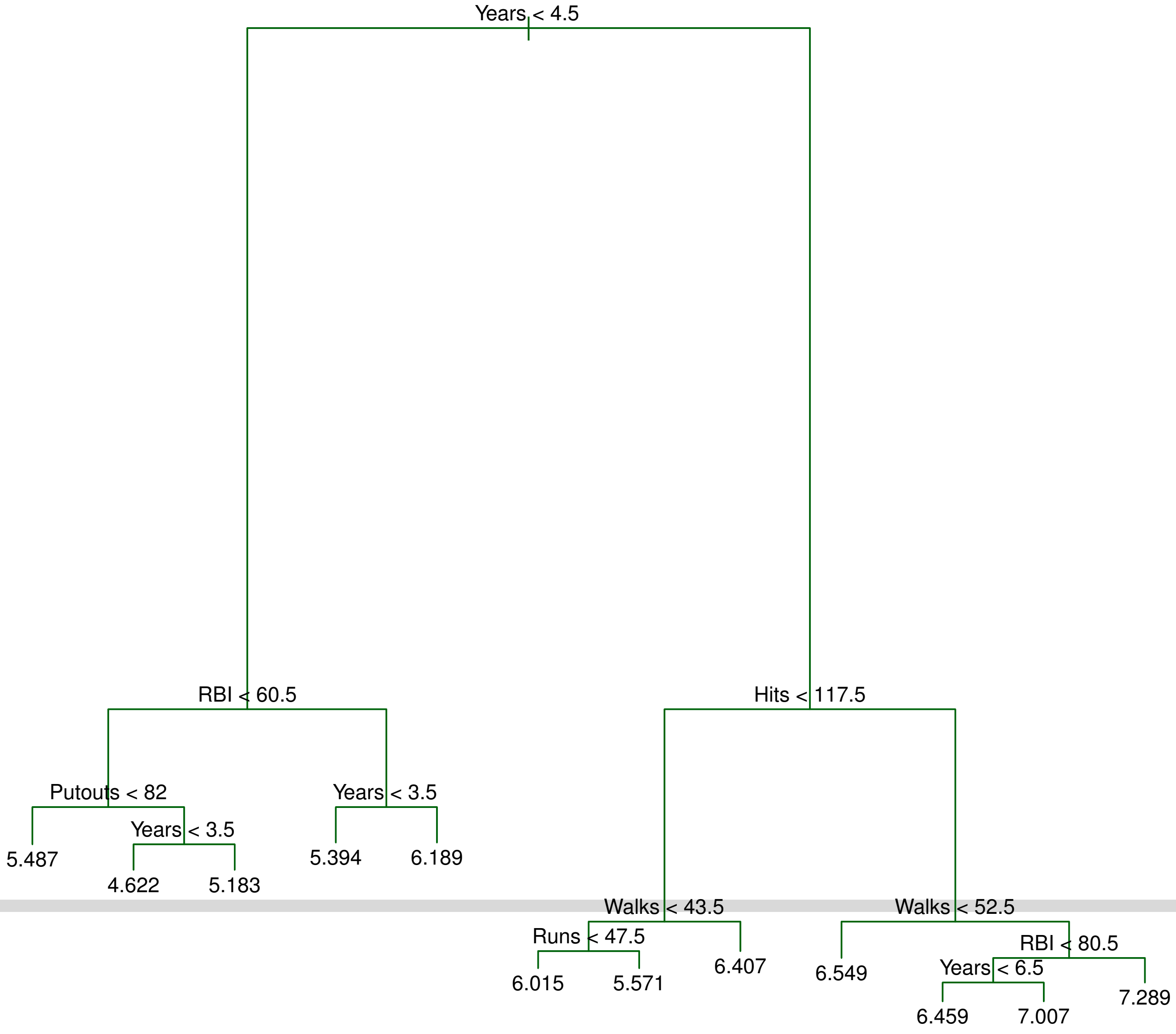
  - When $\alpha = 0$, we select the full tree ($T = T_0$)
  - When $\alpha = \infty$, we select the null tree ($|T| = 0$)
  - For $0 < \alpha_1 < \alpha_2 < \cdots < \alpha_m$ (the corresponding trees are $T_0, T_1, T_2, \cdots, T_m$), choose the optimal α (the optimal T_i) by cross validation

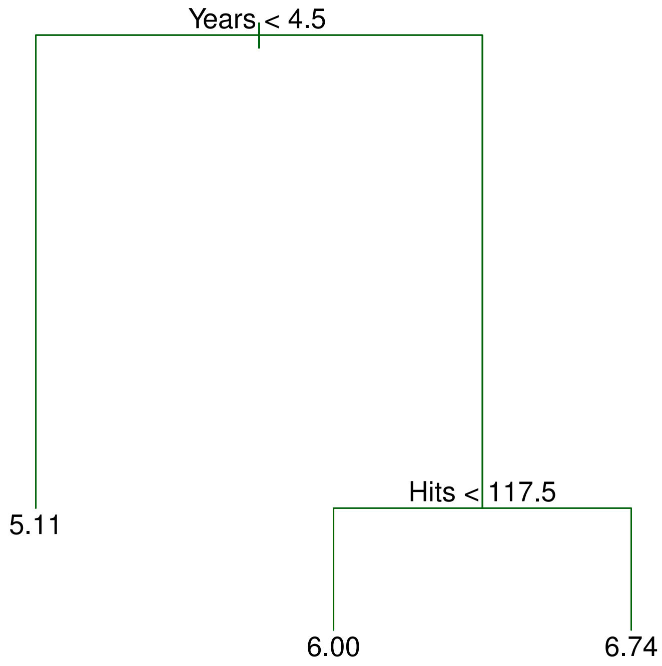
Select α

- Ten-fold cross validation
  - For a range of values $\alpha_1, \alpha_2, \cdots, \alpha_m$, construct the corresponding sequence of trees $T_1^{(k)}, T_2^{(k)}, \cdots, T_m^{(k)}$
  - The sequence of trees vary with the hold-out fold; Make prediction for each region in each tree $T_i^{(k)}$
  - For each tree $T_i^{(k)}$, calculate the RSS on the hold-out fold k
  - Select the parameter α that minimizes the average error across ten folds

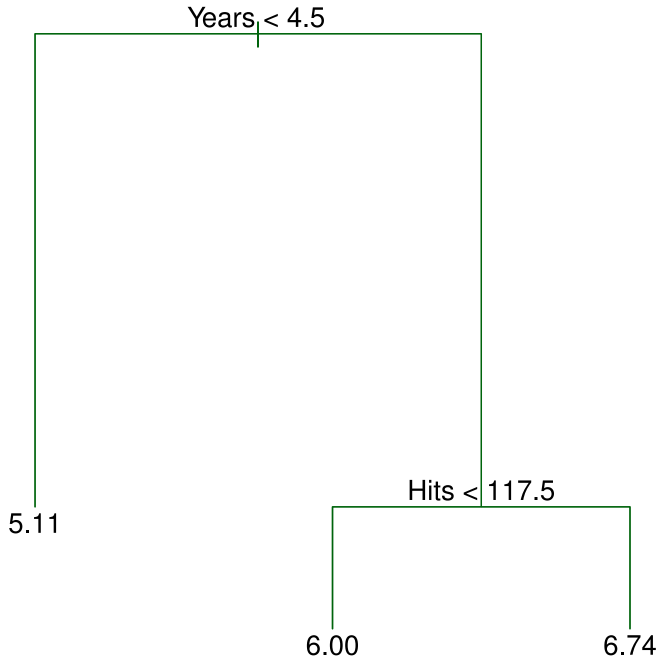
Example

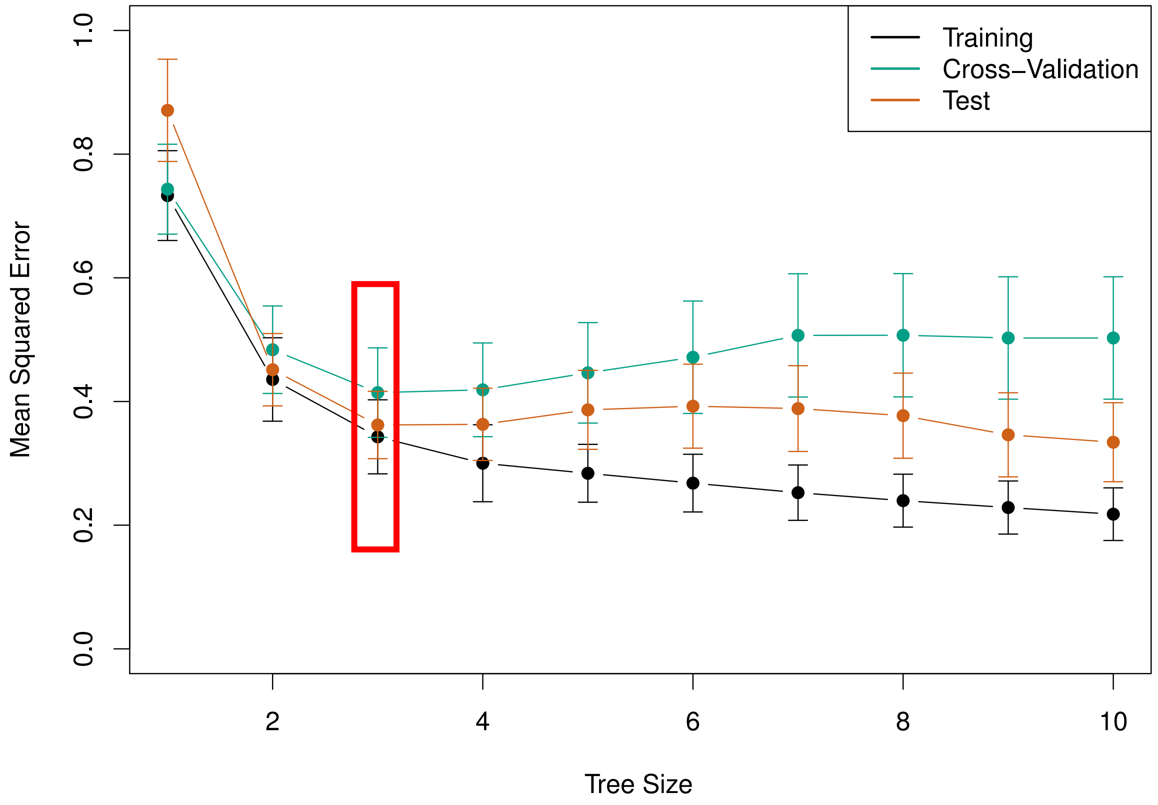
- Unpruned tree vs. pruned tree with cost complexity tuning





Cross-validation results





Lecture plan

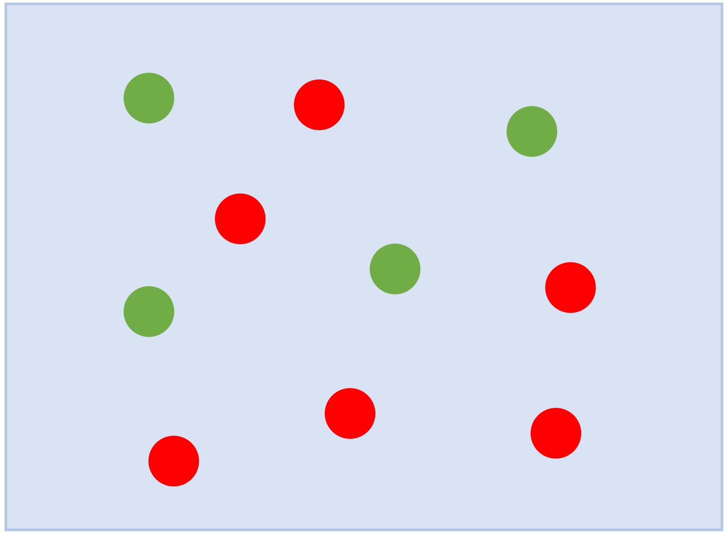
- **Classification tree**

Classification tree

- Classification trees work similar to regression trees
  1. Partition the feature space into J **distinct and non-overlapping** regions, $R_1, R_2, \cdots, R_J$
  2. Make the **same** prediction for every observation in region R_j: Mean of the training observations in R_j
- Step 1: Minimize classification error rate
- Step 2: Predict response by **majority vote**, pick the most common class in a region

Metrics

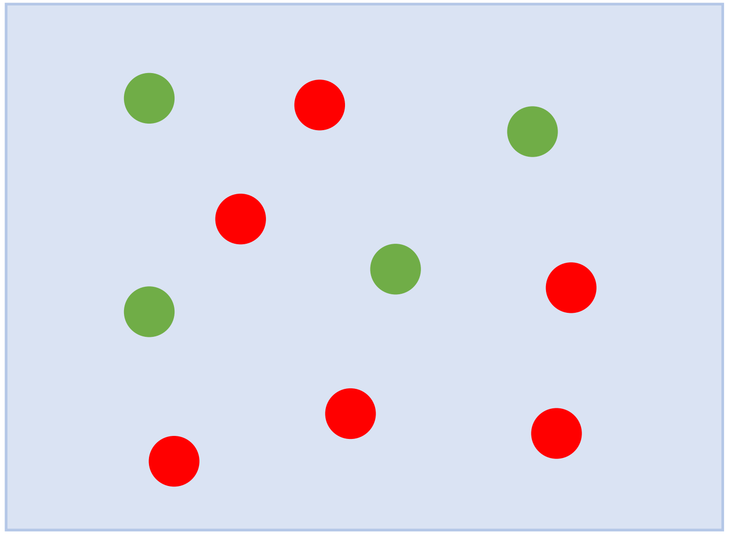
- The 0–1 loss or misclassification rate in region m: $\sum_{i \in R_m} 1(y_i \neq \hat{y}_{R_m})$
  - Example: $\hat{y}_{R_m} = \text{red}$, $\sum_{i \in R_m} 1(y_i \neq \hat{y}_{R_m}) = 4$
- The Gini index in region m: $G_m = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$
  - $\hat{p}_{m,\text{red}} = \frac{6}{10} = 0.6$
  - $\hat{p}_{m,\text{green}} = \frac{4}{10} = 0.4$
  - $G_m = 0.6(1 - 0.6) + 0.4(1 - 0.4) = 0.48$



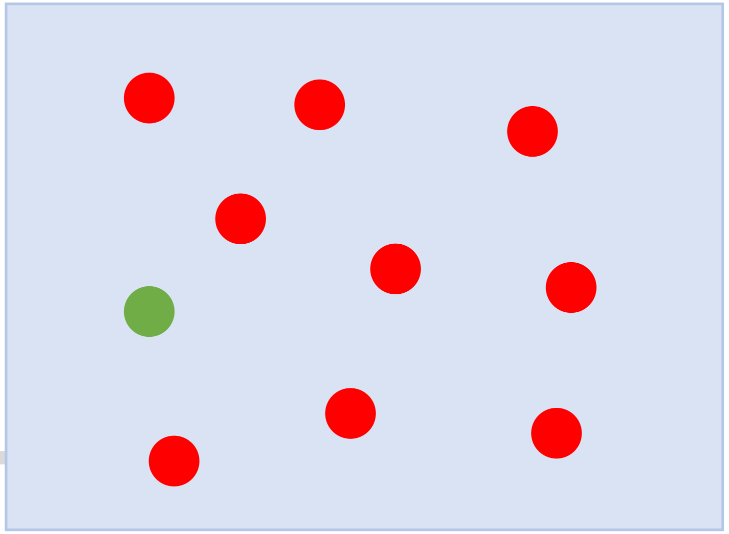
Region m

Metrics

- The entropy in region m: $D_m = -\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}$
- Example:
  - $\hat{p}_{m,\text{red}} = \frac{6}{10} = 0.6$
  - $\hat{p}_{m,\text{green}} = \frac{4}{10} = 0.4$
  - $D_m = -0.6 \log 0.6 - 0.4 \log 0.4 = 0.673$
- Example:
  - $\hat{p}_{m,\text{red}} = \frac{9}{10} = 0.9$
  - $\hat{p}_{m,\text{green}} = \frac{1}{10} = 0.1$
  - $D_m = -0.9 \log 0.9 - 0.1 \log 0.1 = 0.461$
  - D_m is also a measure of purity: D_m is small if all $\hat{p}_{mk}$'s are close to zero or one

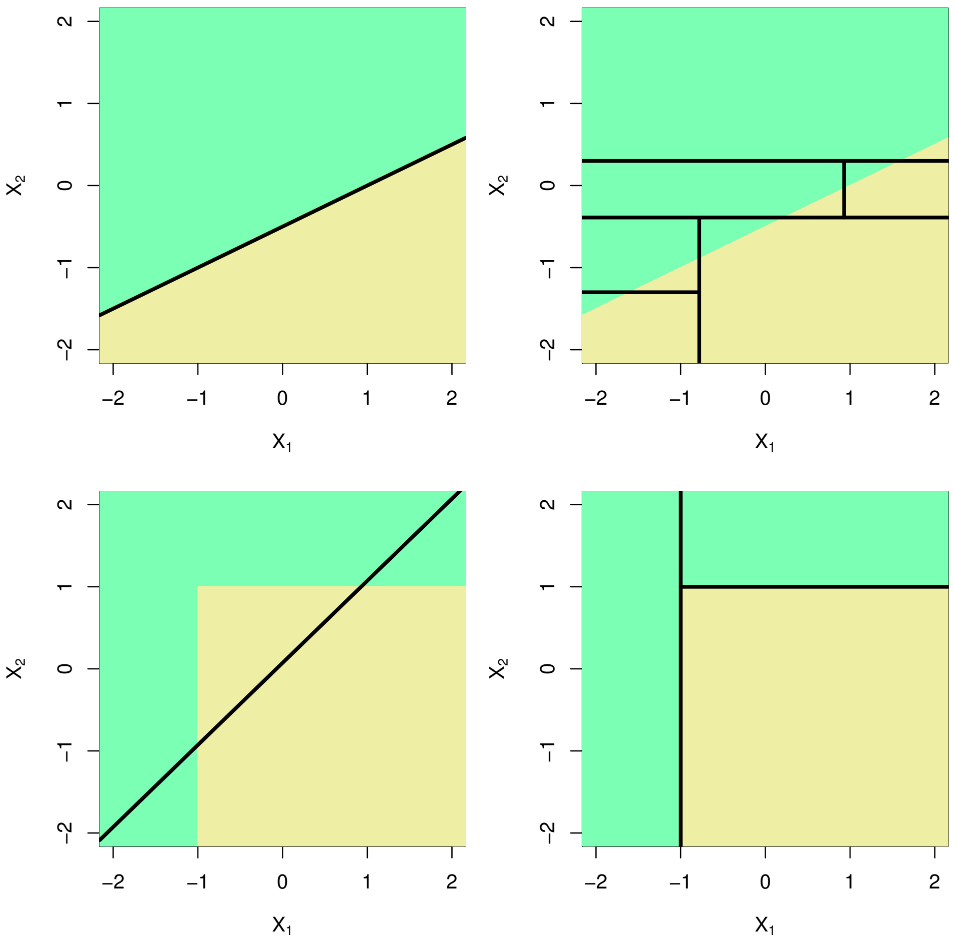


Region m



Region m

Decision boundaries: linear model vs. decision tree

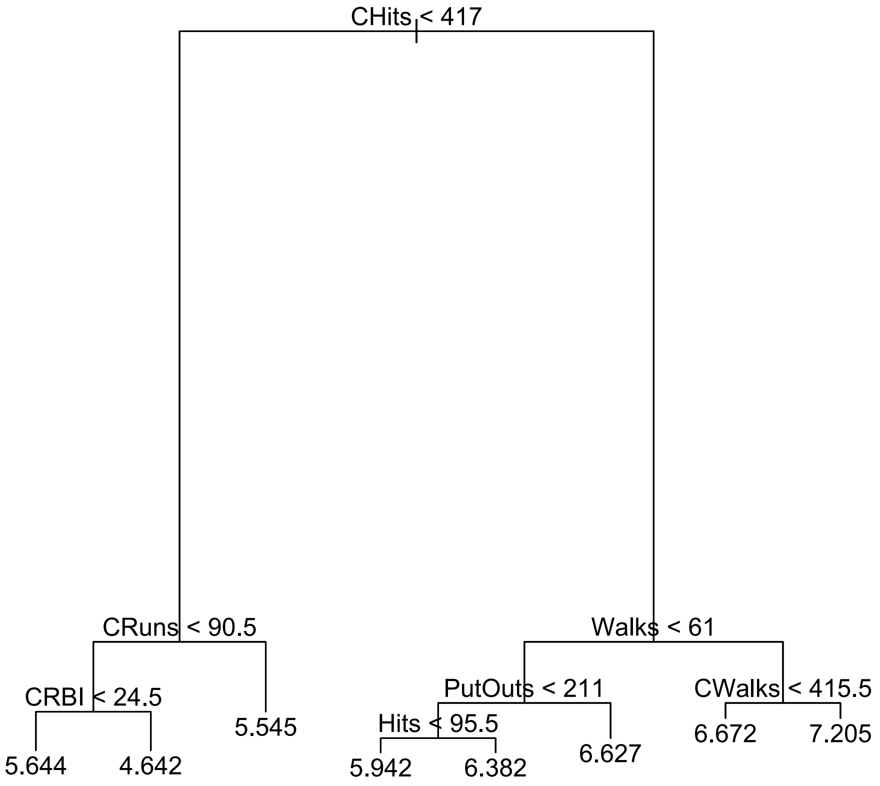


Lecture plan

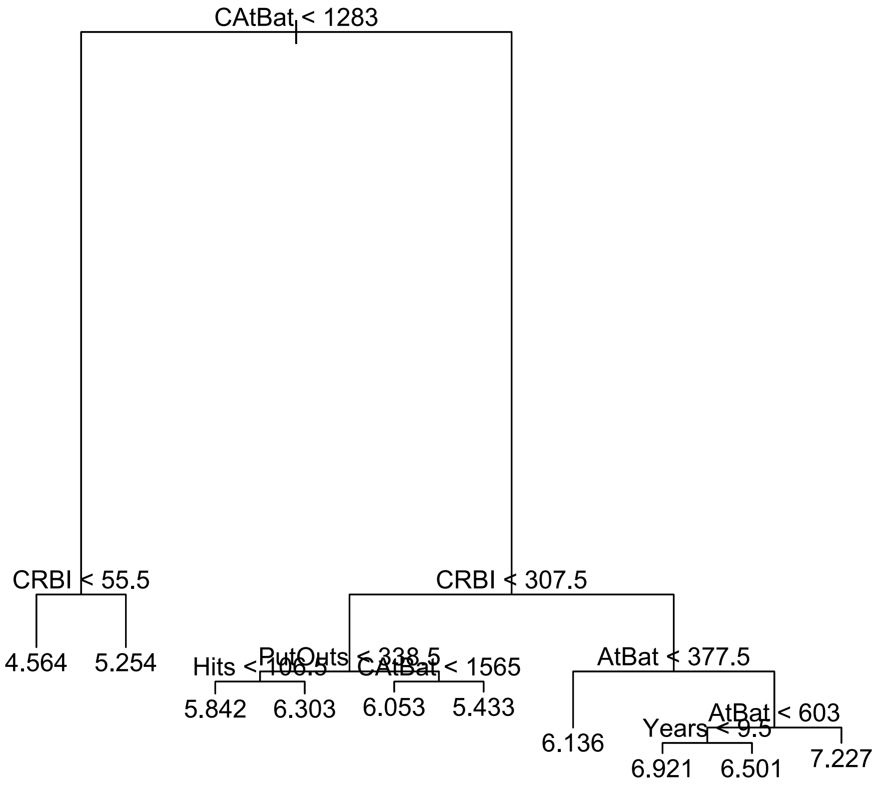
- **Bagging**

Decision tree has a high variance

- **Example:** Predicting a baseball player's salary
  - Split the training data into two equal-sized parts at random creates disparity



Subsample 1



Subsample 2

Bagging

- Bagging is a way to reduce such variance
- **Idea: Bootstrap aggregation**
- **Example**: Estimate the mean of Z

Z_1	1.03
Z_2	1.56
Z_3	2.37
Z_4	2.13
Z_5	2.47

$$\bar{Z} = 1.91$$

$$\mathrm{Var}(\bar{Z}) = \frac{\sigma^2}{n} = \frac{1}{5} = 0.2$$

Data generating process: $Z \sim N(2,1)$

Toy example

- Suppose we have many independent sampling of datasets

Dataset 1

$Z_1^{(1)}$	1.03
$Z_2^{(1)}$	1.56
$Z_3^{(1)}$	2.37
$Z_4^{(1)}$	2.13
$Z_5^{(1)}$	2.47

$\bar{Z}^{(1)} = 1.91$
$\text{Var}(\bar{Z}^{(1)}) = 0.2$

Dataset 2

$Z_1^{(2)}$	3.44
$Z_2^{(2)}$	3.06
$Z_3^{(2)}$	2.42
$Z_4^{(2)}$	2.40
$Z_5^{(2)}$	-0.78

$\bar{Z}^{(2)} = 2.11$
$\text{Var}(\bar{Z}^{(2)}) = 0.2$

Dataset 3

$Z_1^{(3)}$	-0.13
$Z_2^{(3)}$	2.28
$Z_3^{(3)}$	2.09
$Z_4^{(3)}$	2.72
$Z_5^{(3)}$	1.40

$\bar{Z}^{(3)} = 1.67$
$\text{Var}(\bar{Z}^{(3)}) = 0.2$

Dataset 4

$Z_1^{(4)}$	0.94
$Z_2^{(4)}$	1.84
$Z_3^{(4)}$	1.92
$Z_4^{(4)}$	2.49
$Z_5^{(4)}$	2.37

$\bar{Z}^{(4)} = 1.91$
$\text{Var}(\bar{Z}^{(4)}) = 0.2$

$$\bar{Z}_{agg} = (\bar{Z}^{(1)} + \bar{Z}^{(2)} + \bar{Z}^{(3)} + \bar{Z}^{(4)})/4 = 1.90$$

$$\text{Var}\left(\bar{Z}_{agg}\right) = \frac{0.2}{4} = 0.05$$

Toy example

- In practice, we only have one training dataset
- How can we create many datasets? **Idea: Bootstrap**

Z_1	1.03
Z_2	1.56
Z_3	2.37
Z_4	2.13
Z_5	2.47

Sampling with replacement →

Sample #1

Z_1	1.03
Z_2	1.56
Z_1	1.03
Z_5	2.47
Z_4	2.13

Sample #2

Z_4	2.13
Z_1	1.03
Z_3	2.37
Z_2	1.56
Z_3	2.37

Sample #3

Z_5	2.47
Z_2	1.56
Z_3	2.37
Z_2	1.56
Z_1	1.03

Sample #4

Z_5	2.47
Z_3	2.37
Z_3	2.37
Z_1	1.03
Z_2	1.56

Bagging to reduce variance

- Estimate the mean on each bootstrap sampling set

Sample #1

Z_1	1.03
Z_2	1.56
Z_5	2.47
Z_5	2.47
Z_4	2.13

$\bar{Z}^{(1)} = 1.93$

Sample #3

Z_5	2.47
Z_2	1.56
Z_3	2.37
Z_2	1.56
Z_1	1.03

$\bar{Z}^{(3)} = 1.80$

Sample #2

Z_4	2.13
Z_1	1.03
Z_3	2.37
Z_2	1.56
Z_3	2.37

$\bar{Z}^{(2)} = 1.89$

Sample #4

Z_5	2.47
Z_3	2.37
Z_3	2.37
Z_1	1.03
Z_2	1.56

$\bar{Z}^{(4)} = 1.96$

Toy example

- Average all estimates

$\bar{Z}^{(1)} = 1.93$ $\bar{Z}^{(2)} = 1.89$ $\bar{Z}^{(3)} = 1.80$ $\bar{Z}^{(4)} = 1.96$

$$\frac{\bar{Z}^{(1)} + \bar{Z}^{(2)} + \bar{Z}^{(3)} + \bar{Z}^{(4)}}{4} = 1.90$$

- This is called **bagging** (**B**ootstrap **agg**regat**ing**)
  - Bagging amounts to averaging the fits from B independent data sets, which would reduce the variance by a factor $\frac{1}{B}$

Bagging for decision trees

- Estimate a decision tree model $f(x)$ using bootstrap

X_1	Y_1
X_2	Y_2
X_3	Y_3
X_4	Y_4
X_5	Y_5

Sampling with replacement →

Sample #1

X_1	Y_1
X_2	Y_2
X_1	Y_1
X_5	Y_5
X_4	Y_4

Sample #2

X_4	Y_4
X_1	Y_1
X_3	Y_3
X_2	Y_2
X_3	Y_3

Sample #3

X_5	Y_5
X_2	Y_2
X_3	Y_3
X_2	Y_2
X_1	Y_1

Sample #4

X_5	Y_5
X_3	Y_3
X_3	Y_3
X_1	Y_1
X_2	Y_2

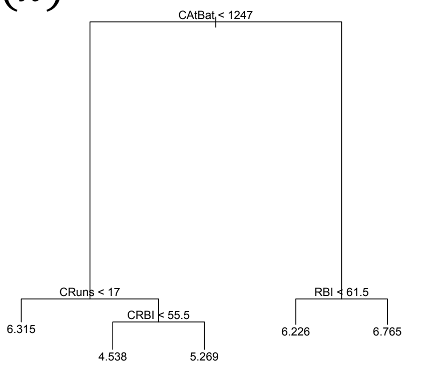
Bagging for decision trees

- Estimate a decision tree model $f(x)$ using bootstrap

Sample #1

X_1	Y_1
X_2	Y_2
X_1	Y_1
X_5	Y_5
X_4	Y_4

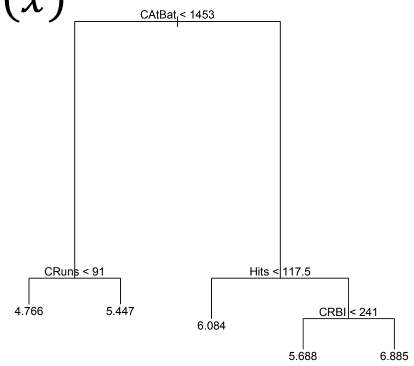
$\hat{f}^1(x)$



Sample #3

X_5	Y_5
X_2	Y_2
X_3	Y_3
X_2	Y_2
X_1	Y_1

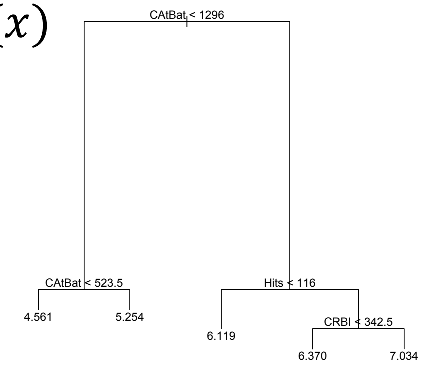
$\hat{f}^3(x)$



Sample #2

X_4	Y_4
X_1	Y_1
X_3	Y_3
X_2	Y_2
X_3	Y_3

$\hat{f}^2(x)$



Sample #4

X_5	Y_5
X_3	Y_3
X_3	Y_3
X_1	Y_1
X_2	Y_2

$\hat{f}^4(x)$