

# Ch 8.1: Decision Trees

## Lecture 25 - CMSE 381

Prof. Elizabeth Munch

Michigan State University

::

Dept of Computational Mathematics, Science & Engineering

Wed, Nov 8, 2023

## **Last time:**

- Cubic Splines

## **This lecture:**

- 8.1 Decision Trees

## **Announcements:**

- HW #6 Due tonight

# Section 1

## Decision Trees

## Big idea

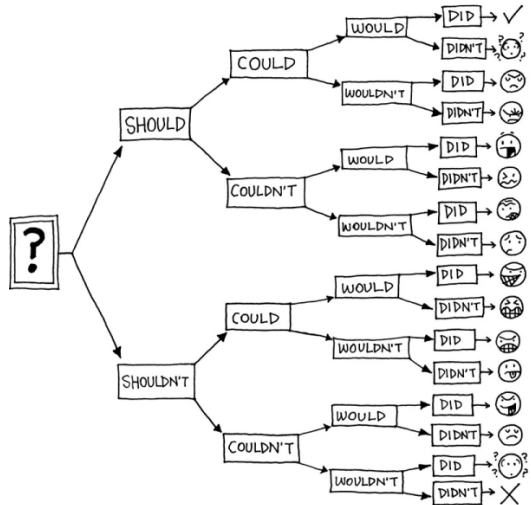


Image: <https://marekbennett.com/2014/02/14/decision-tree/>

# Subset of Hitters data

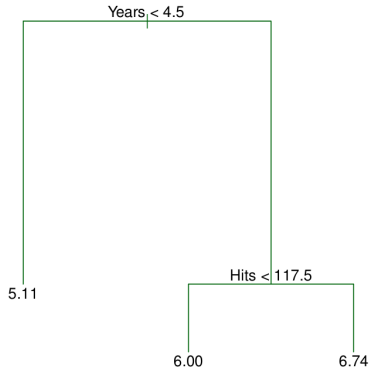
	Hits	Years	Salary	LogSalary
1	81	14	475.0	6.163315
2	130	3	480.0	6.173786
3	141	11	500.0	6.214608
4	87	2	91.5	4.516339
5	169	11	750.0	6.620073
...	...	...	...	...
317	127	5	700.0	6.551080
318	136	12	875.0	6.774224
319	126	6	385.0	5.953243
320	144	8	960.0	6.866933
321	170	11	1000.0	6.907755

# First decision tree example

	Hits	Years	LogSalary
1	81	14	6.163315
2	130	3	6.173786
3	141	11	6.214608
4	87	2	4.516339
5	169	11	6.620073
...	...	...	...
317	127	5	6.551080
318	136	12	6.774224
319	126	6	5.953243
320	144	8	6.866933
321	170	11	6.907755



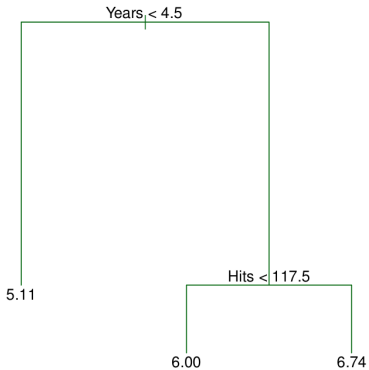
# Interpretation of example



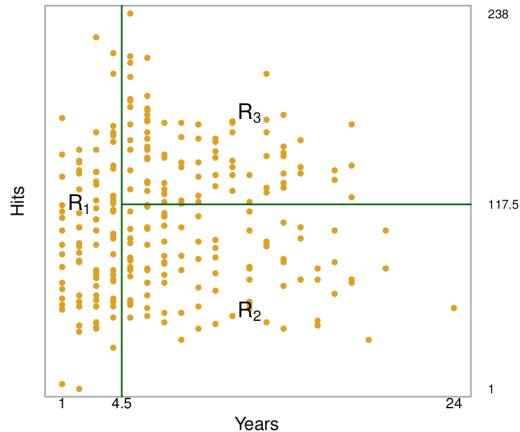
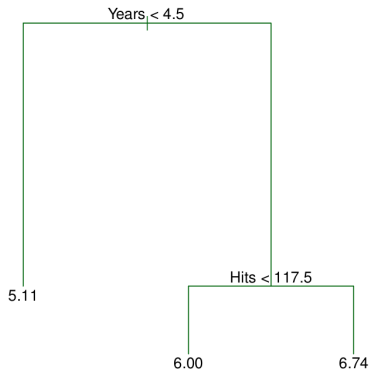
# Coding a regression decision tree



# Regions defined by the tree

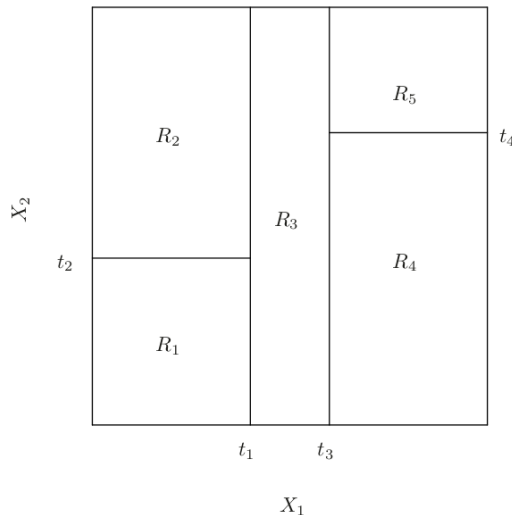


# Viewing Regions Defined by Tree



# How do we actually get the tree? Two steps

- 1 We divide the predictor space — that is, the set of possible values for  $X_1, X_2, \dots, X_p$  — into  $J$  distinct and non-overlapping regions,  $R_1, R_2, \dots, R_J$ .
- 2 For every observation that falls into the region  $R_j$ , we make the same prediction = the mean of the response values for the training observations in  $R_j$ .



## Step 1: How do we decide on $R_j$ s?

### Goal:

Find boxes  $R_1, \dots, R_J$  that minimize

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

$\hat{y}_{R_j}$  = mean response for training  
observations in  $j$ th box

# Recursive Binary Splitting

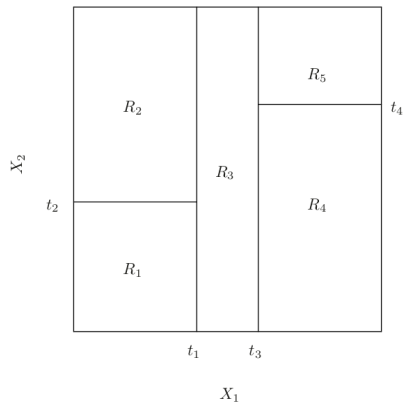
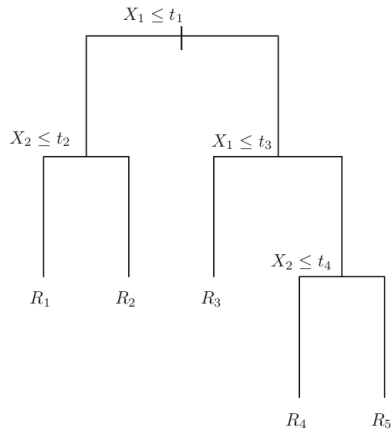
- Pick  $X_j$
- Pick  $s$  so that splitting into  $\{X \mid X_j < s\}$  and  $\{X \mid X_j \geq s\}$  results in largest possible reduction in RSS

$$R_1(j, s) = \{X \mid X_j < s\}$$

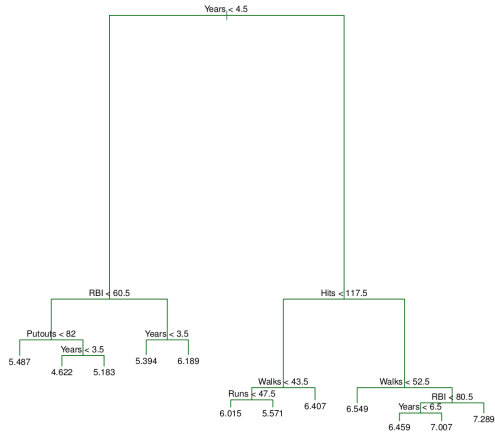
$$R_2(j, s) = \{X \mid X_j \geq s\}$$

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

# Rinse and repeat



# Pruning



# Weakest Link Pruning

Also called Cost complexity pruning

For every  $\alpha$ , there is a subtree  $T$  that minimizes:

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

- $|T|$  = number of terminal nodes of  $T$
- $R_m$  is rectangle for  $m$ th terminal node
- $\hat{y}_{R_m}$  is mean of training observations in  $R_m$



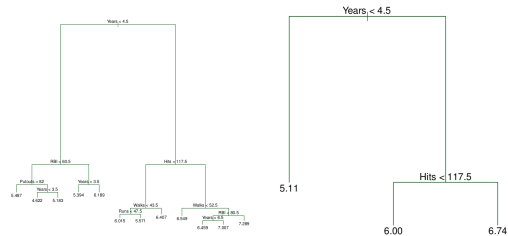
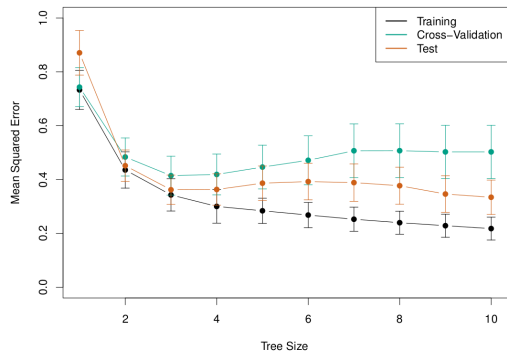
---

**Algorithm 8.1** *Building a Regression Tree*

---

1. Use recursive binary splitting to grow a large tree on the training data, stopping only when each terminal node has fewer than some minimum number of observations.
  2. Apply cost complexity pruning to the large tree in order to obtain a sequence of best subtrees, as a function of  $\alpha$ .
  3. Use K-fold cross-validation to choose  $\alpha$ . That is, divide the training observations into  $K$  folds. For each  $k = 1, \dots, K$ :
    - (a) Repeat Steps 1 and 2 on all but the  $k$ th fold of the training data.
    - (b) Evaluate the mean squared prediction error on the data in the left-out  $k$ th fold, as a function of  $\alpha$ .Average the results for each value of  $\alpha$ , and pick  $\alpha$  to minimize the average error.
  4. Return the subtree from Step 2 that corresponds to the chosen value of  $\alpha$ .
-

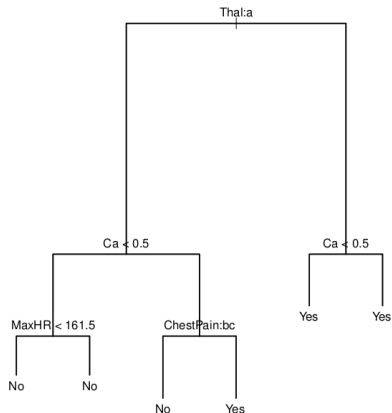
# Messing with $\alpha$



## Section 2

### Classification Decision Tree

# Basic idea



- $\hat{p}_{mk}$  = proportion of training observations in  $R_m$  from the  $k$ th class
- $E = 1 - \max_k(\hat{p}_{mk})$

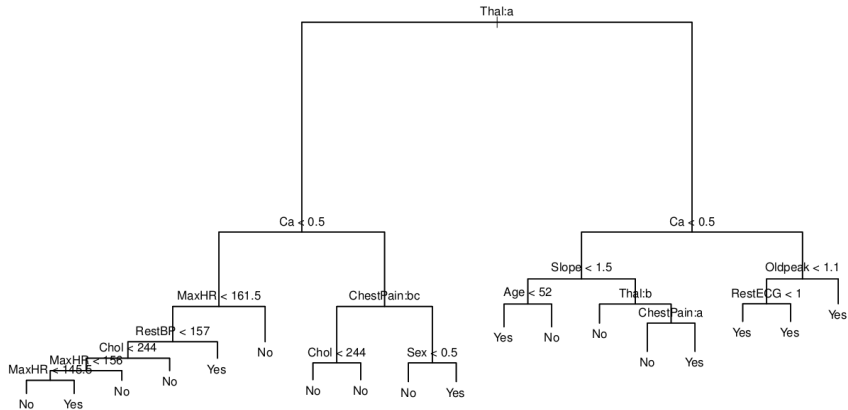
# Gini index

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

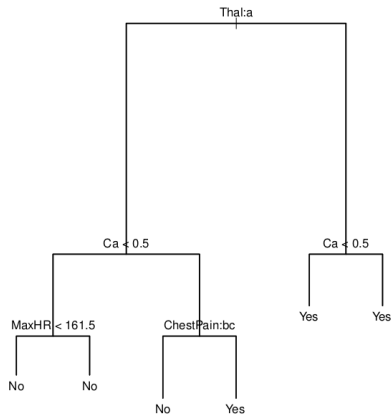
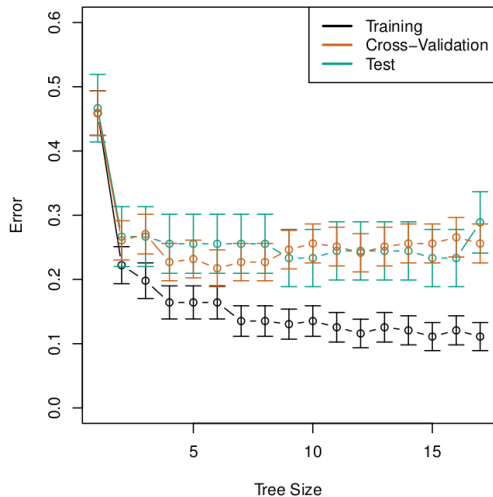
# Entropy

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

# Example



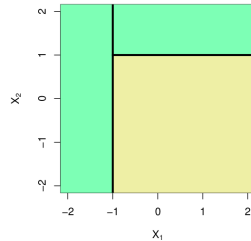
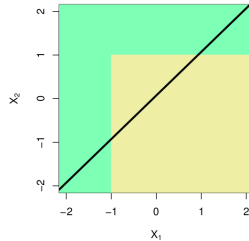
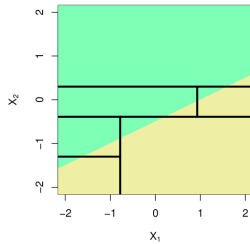
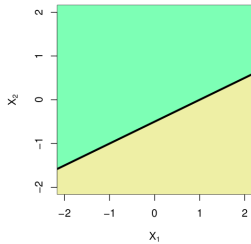
# Pruning the example





# More coding!

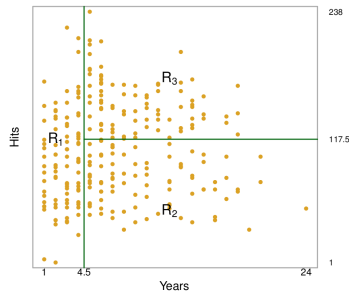
# Linear models vs trees



**Pros:**

**Cons:**

- Split into regions by greedily decreasing RSS
- Prune tree by using cost complexity
- Not robust - Next time, figure out how to aggregate trees



# Next time

Status	Lec #	Date		Reading	Homeworks
		Mon	Oct 23	No class - Fall break	
		Wed	Oct 25	<b>Midterm #2</b>	
Done	20	Fri	Oct 27	Dimension Reduction	6.3
Done	21	Mon	Oct 30	More dimension reduction; High dimensions	6.4
Done	22	Wed	Nov 1	Polynomial & Step Functions	7.1, 7.2
Pushed	23	Fri	Nov 3	Step Functions; Basis functions; Start Splines	7.2 - 7.4
	24	Mon	Nov 6	Regression Splines	7.4
	25	Wed	Nov 8	Decision Trees	8.1
	26	Fri	Nov 10	Random Forests	8.2.1, 8.2.2
	27	Mon	Nov 13	Maximal Margin Classifier	9.1
	28	Wed	Nov 15	SVC	9.2
	29	Fri	Nov 17	SVM	9.3, 9.4
	30	Mon	Nov 20	Single layer NN	10.1
	31	Wed	Nov 22	Virtual: Project office hours	
		Fri	Nov 24	No class - Thanksgiving	
		Mon	Nov 27	<b>Review</b>	
		Wed	Nov 29	<b>Midterm #3</b>	
	32	Fri	Dec 1	Multi Layer NN	10.2
	33	Mon	Dec 4	CNN	10.3
	34	Wed	Dec 6	Unsupervised Learning & Clustering	12.1, 12.4
	35	Fri	Dec 8	Virtual: Project office hours	Project due