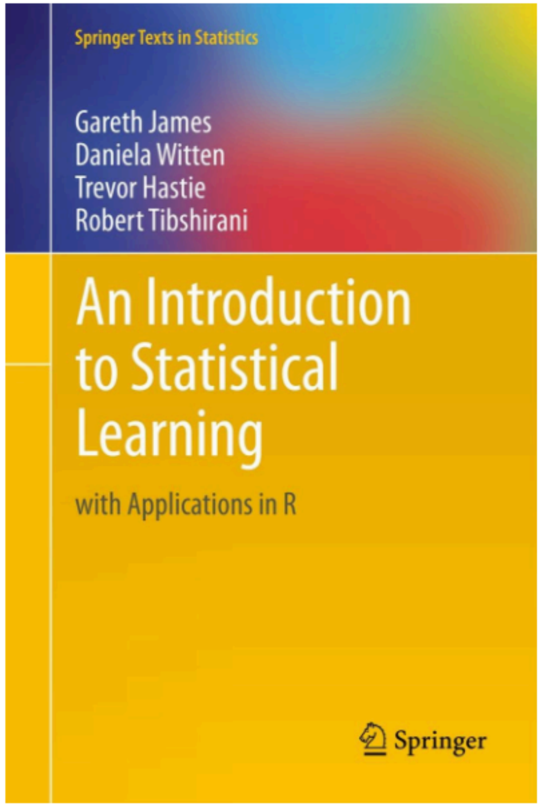
# 7. Decision Trees

## ESC Spring 2018 – Data Mining and Analysis

**SeoHyeong Jeong**




**Textbook:**

An Introduction to Statistical Learning

**Lecture Slides:**

Stanford Stats 202: Data Mining and Analysis

Spring 17' ESC Statistical Data Analysis

**Reading:**

An Introduction to Statistical Learning

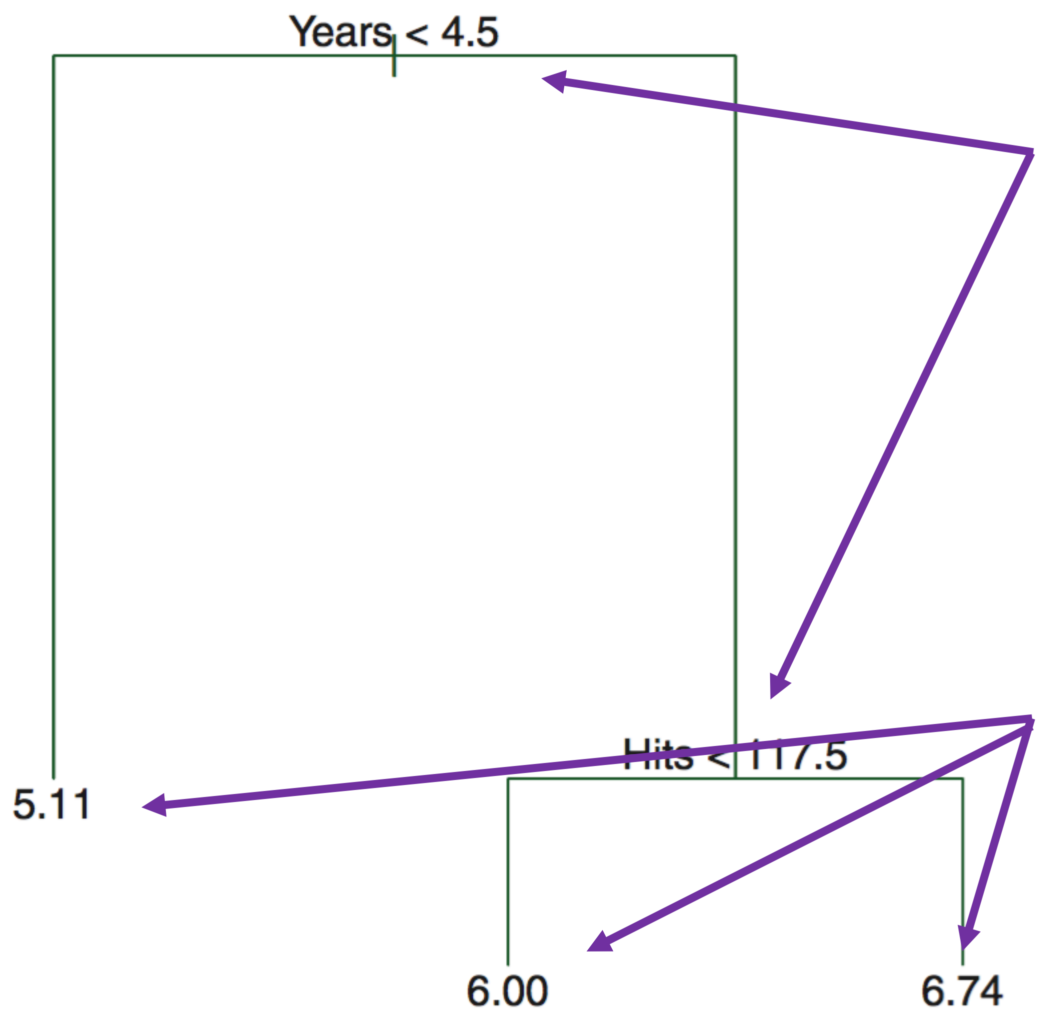
*chapter 8.1 The Basics of Decision Trees*

# Table of Contents


1. Introduction
2. Decision Trees
3. Tree Pruning
4. Advantages and Disadvantages

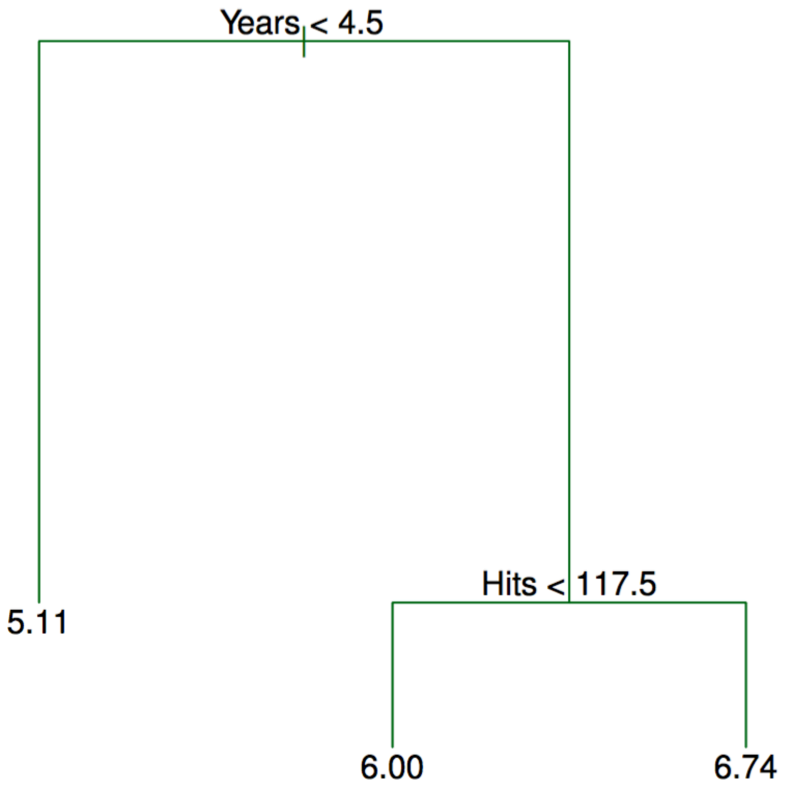

# Introduction

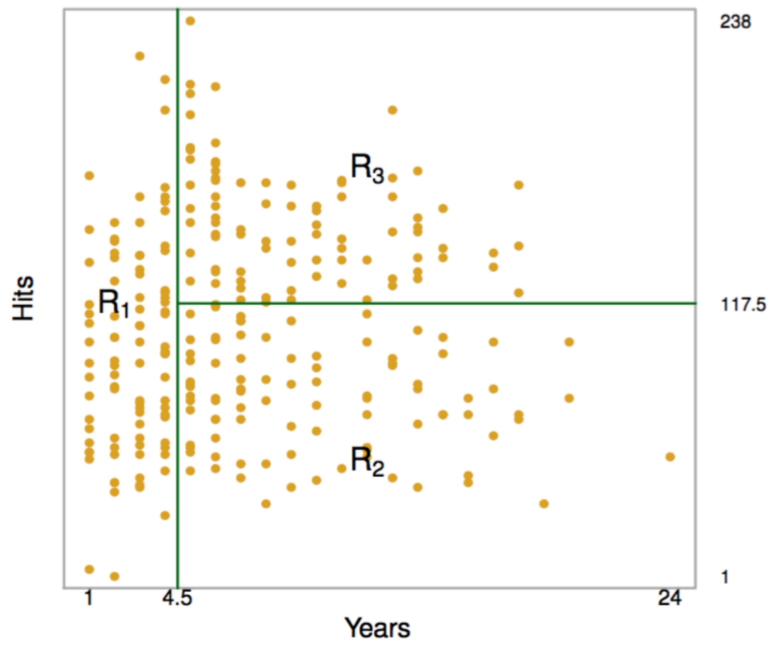
- Tree-based methods involve *stratifying* or *segmenting* the predictor space into a number of simple regions.
- Since the set of splitting rules can be summarized in a tree, these approaches are known as the *decision-tree* methods.
- Decision trees can be applied to both regression and classification problems.
- In the decision-tree methods:
  - We build a tree using *recursive binary splitting* (*recursive partitioning*)
  - Then *prune* the tree




- *Internal Nodes*: the points along the tree where the predictor space is split.
- *Terminal Nodes* (*leaves*): final nodes with the predicted value.

# Example of Predicting a Baseball Player's Salary




- The prediction for a point in region $R_i$ is the average of the training points in $R_i$.

# 1) Regression Tree

- The process is as follows:

1. We divide the predictor space—that is, the set of possible values for $X_1, X_2, \ldots, X_p$—into $J$ distinct and non-overlapping regions, $R_1, R_2, \ldots, R_J$.

2. For every observation that falls into the region $R_j$, we make the same prediction, which is simply the mean of the response values for the training observations in $R_j$.

- It is computationally infeasible to consider every possible partition of the feature space into $J$ boxes in step 1.

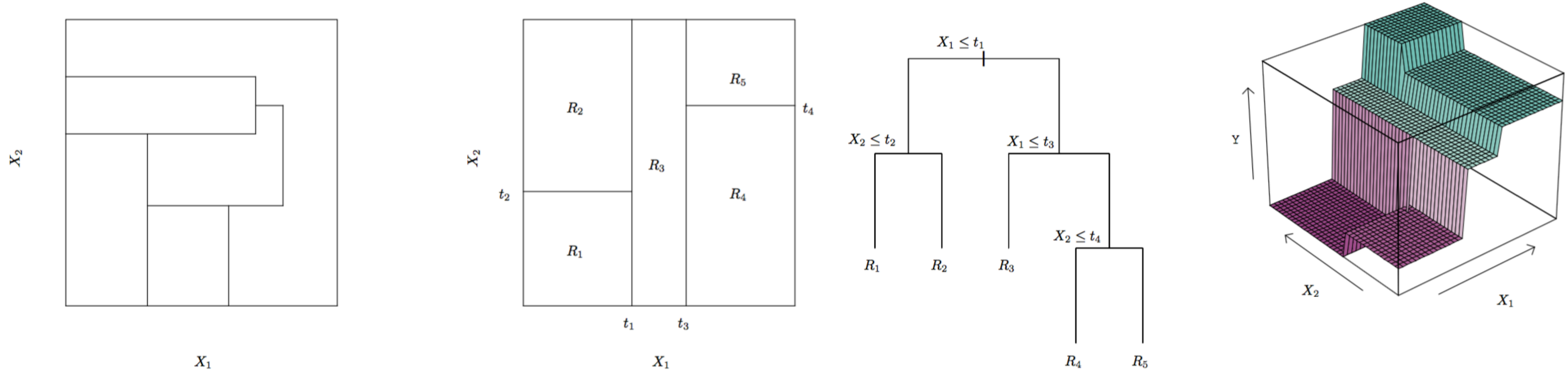
# Regression Tree

- **Solution**: *top-down greedy approach* a.k.a. *recursive binary splitting*
- Start with a single region $R_1$ (entire input space), and iterate:
  1. 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:

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

  2. Redefine the regions with this additional split.
- Terminate when a stopping criterion is met. E.g. when there are 5 observations or fewer in each region.
- This grows the tree from the root towards the leaves (top-down).

# Five Regions Example of Regression Tree




From left to right; Top Left, Top Right, Bottom Left, Bottom Right

**FIGURE 8.3.** Top Left: *A partition of two-dimensional feature space that could not result from recursive binary splitting.* Top Right: *The output of recursive binary splitting on a two-dimensional example.* Bottom Left: *A tree corresponding to the partition in the top right panel.* Bottom Right: *A perspective plot of the prediction surface corresponding to that tree.*

# 2) Classification Tree

- They work much like regression trees.
- We predict the response by **majority vote,** i.e. pick the most common class in every region (mode).
- Instead of trying to minimize the RSS:

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

we minimize a *classification loss* function.

# Classification Losses

- **The 0-1 loss or misclassification rate**:

$$\sum_{m=1}^{|T|} \sum_{x_i \in R_m} \mathbf{1}(y_i \neq \hat{y}_{R_m})$$

- **The Gini index**:

$$\sum_{m=1}^{|T|} q_m \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}),$$

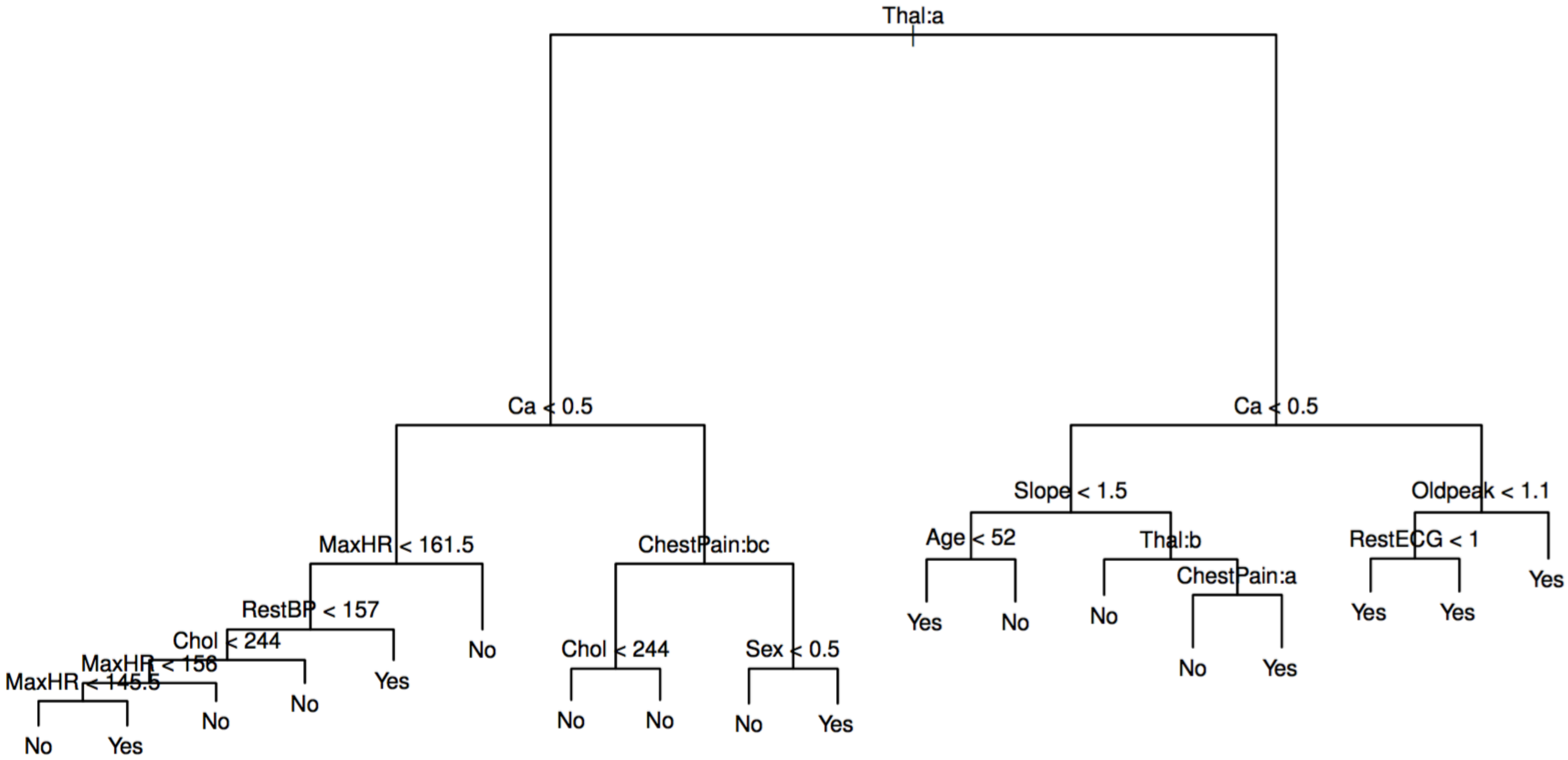
where $\hat{p}_{m,k}$ is the proportion of class $k$ within $R_m$, and $q_m$ is the proportion of samples in $R_m$

- **The cross-entropy**:

$$-\sum_{m=1}^{|T|} q_m \sum_{k=1}^{K} \hat{p}_{mk} \log(\hat{p}_{mk}).$$

- The Gini index and cross-entropy are better measures of the purity of a region, i.e. they are low when the region is mostly one category.
- **Motivation for the Gini index:**
  If instead of predicting the most likely class, we predict a random sample from the distribution $(\hat{p}_{m,1}, \ldots, \hat{p}_{m,K})$, the Gini index is the expected misclassification rate.
- It is typical to use the Gini index or cross-entropy for growing the tree, while using the misclassification rate when pruning the tree.

# Example of a Classification Tree



# How Do We Control Overfitting?

- Building a decision tree may produce good predictions on the training set, but it is likely to *overfit* the data, leading to poor test set performance.

  It may split the predictor space into $n$ regions, which contains each of the response observations $y_1, y_2, \ldots, y_n$.

- A smaller tree with fewer splits (that is, fewer regions $R_1, R_2, \ldots, R_J$) might lead to lower variance and better interpretation at the cost of a little bias.

- **Idea 1**: Find the optimal subtree by cross validation

  -> There are too many possibilities, so we would still overfit.

- **Idea2**: Stop growing the tree when the RSS doesn't drop by more than a threshold with any new cut.

  -> In our greedy algorithm, it is possible to find good cuts after bad ones.

# Tree Pruning

- **Solution**: Prune a large tree from the leaves to the root.

- **Cost Complexity Pruning**:
  - Minimize the following objective over all prunings $T$ of $T_0$:

$$\text{minimize} \sum_{R_m \in T} \sum_{x_i \in R_m} (y_i - \bar{y}_{R_m})^2 + \alpha |T|.$$

  where $|T|$ indicates the number of terminal nodes of the tree $T$.
  - When $\alpha = \infty$, we select the null tree (= tree with one leaf node.)
  - When $\alpha = 0$, we select the full tree.
  - Choose the optimal $\alpha$ (the optimal $T_i$) by cross validation.

# Cross Validation (the wrong way)

1. Construct a sequence of trees $T_0, T_1, \dots, T_m$ for a range of values of $\alpha$.
2. Split the training points into 10 folds.
3. For $k = 1, \dots, 10$,
   - For each tree $T_i$, use every fold except the $k$th to estimate the averages in each region.
   - For each tree $T_i$, calculate the RSS in the test fold.
4. For each tree $T_i$, average the 10 test errors, and select the value of $\alpha$ that minimizes the error.

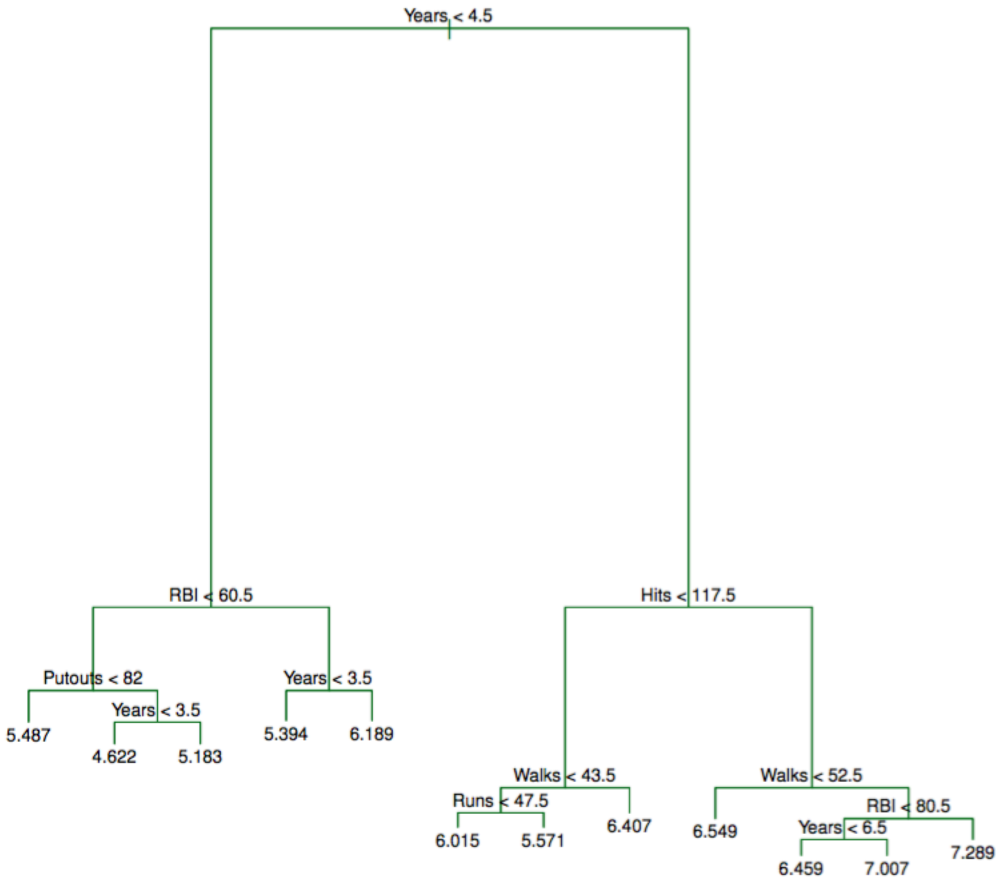
WRONG WAY TO DO CROSS VALIDATION

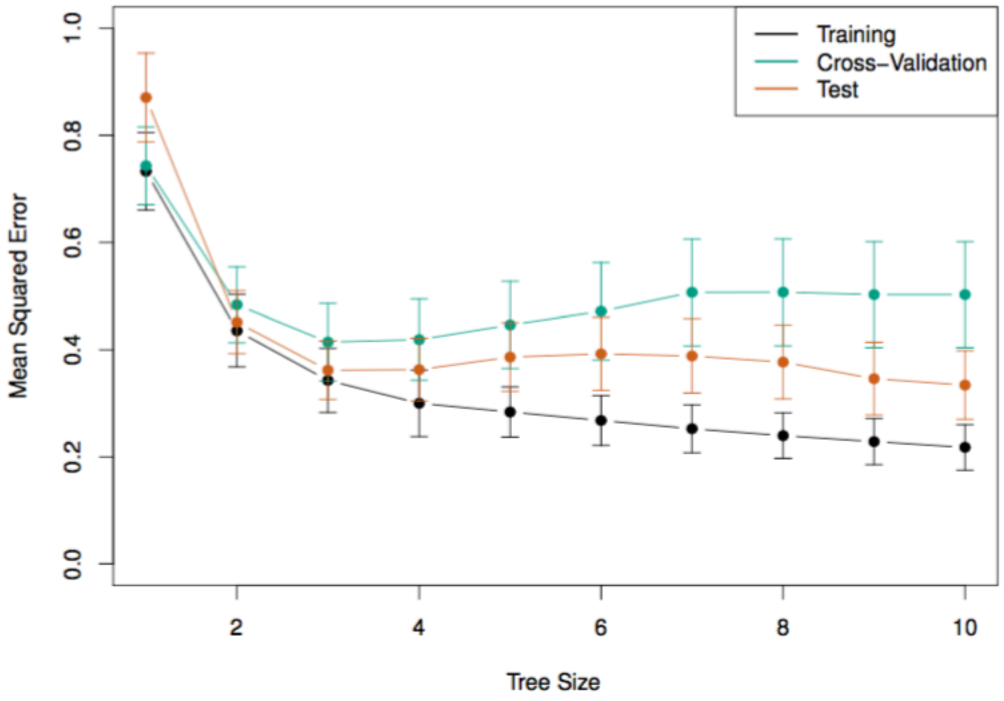
# Cross Validation (the right way)

1. Split the training points into 10 folds.
2. For $k = 1, \ldots, 10$, using every fold except the $k$th:
   - Construct a sequence of trees $T_1, \ldots, T_m$ for a range of values of $\alpha$, and find the prediction for each region in each one.
   - For each tree $T_i$, calculate the RSS in the test fold.
3. Select the parameter $\alpha$ that minimizes the average test error.

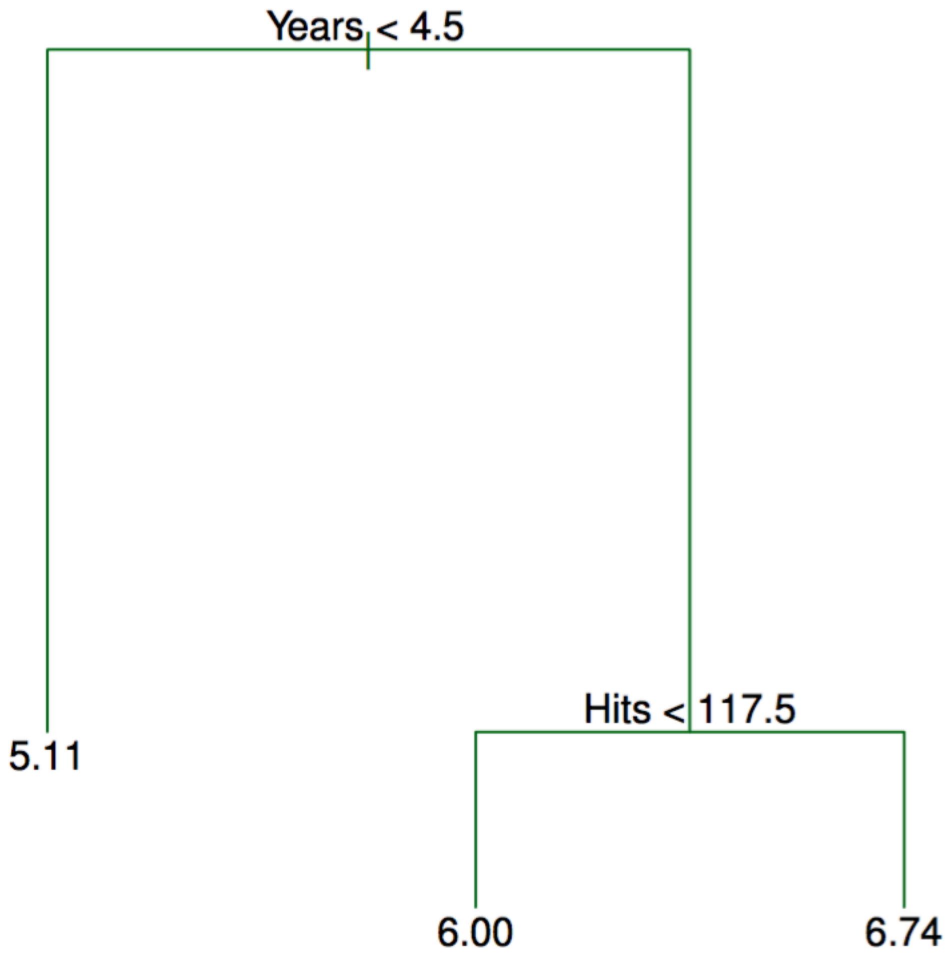
*NOTE*: We are doing all fitting, **including the construction of the trees**, using only the training data.

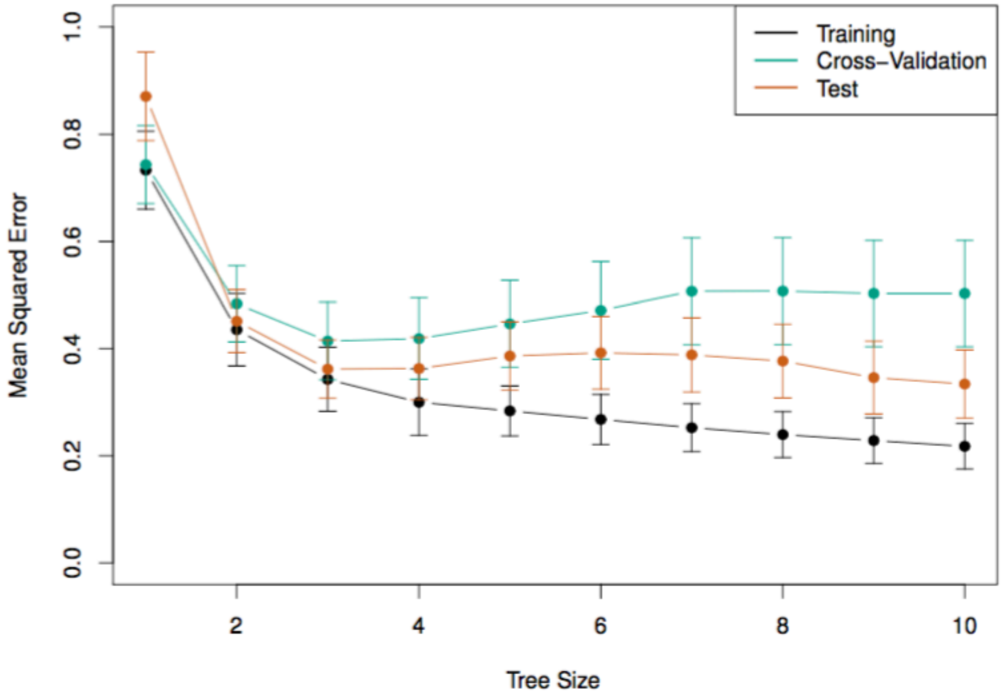
# Example. Predicting Baseball Salaries



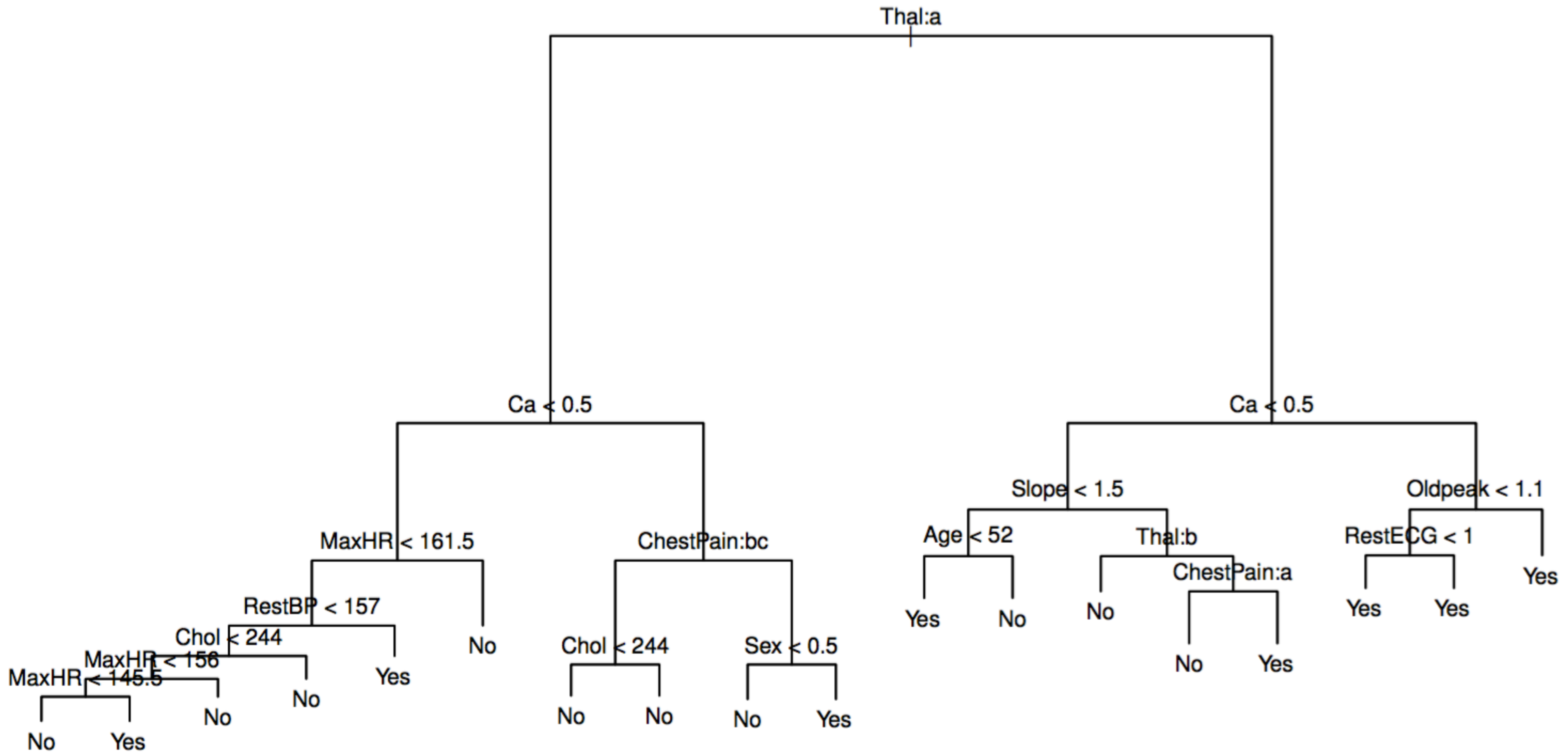



# Example. Predicting Baseball Salaries

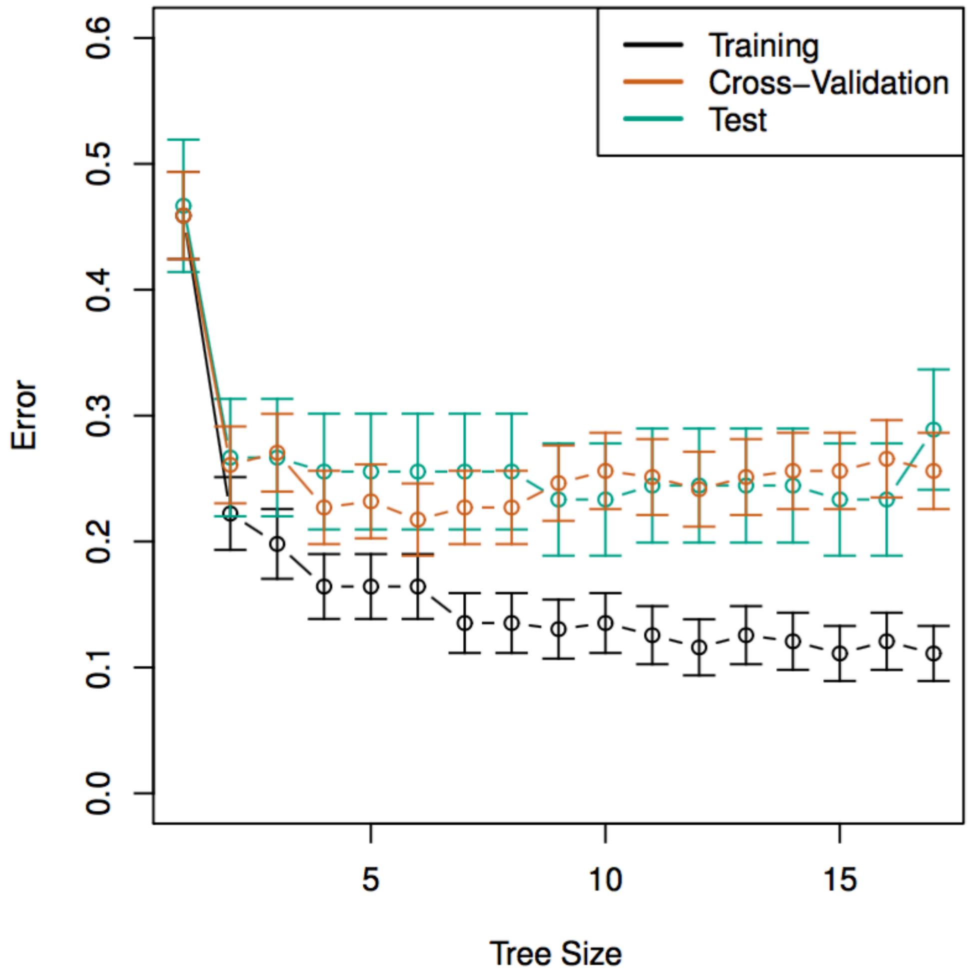


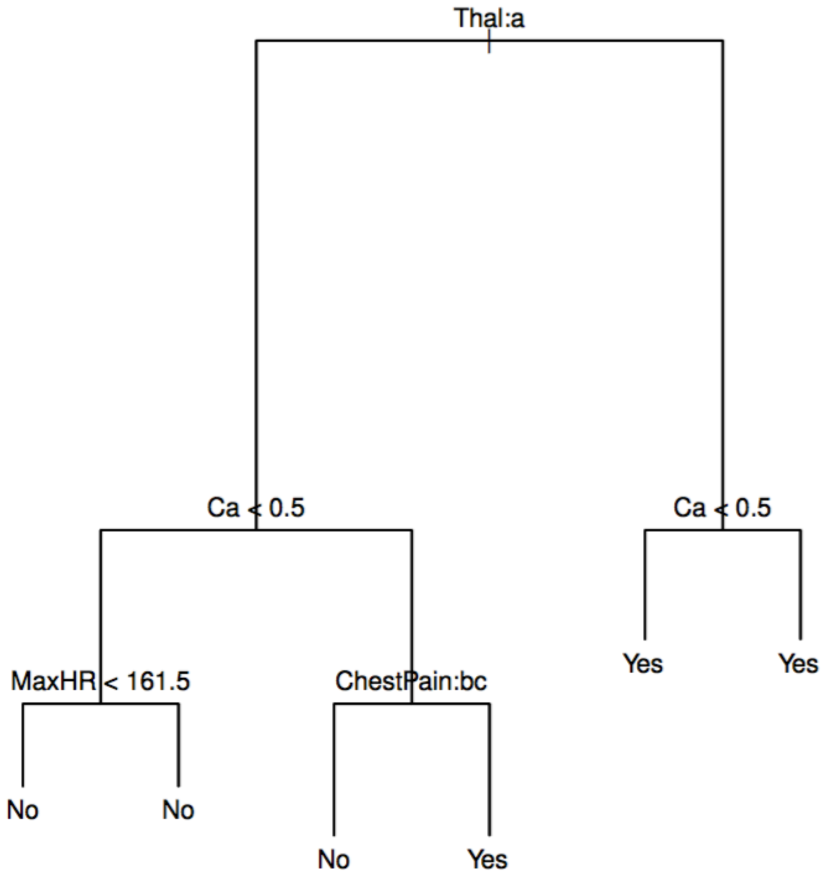



# Example. Heart Dataset



# Example. Heart Dataset



# Trees vs. Linear Model

- Linear regression assumes a model of the form

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

whereas regression trees assume a model of the form

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

- Linear relationship between the features and the response: linear regression will outperform regression tree
- Highly non-linear relationship between the features and the response: regression tree may outperform classical approaches

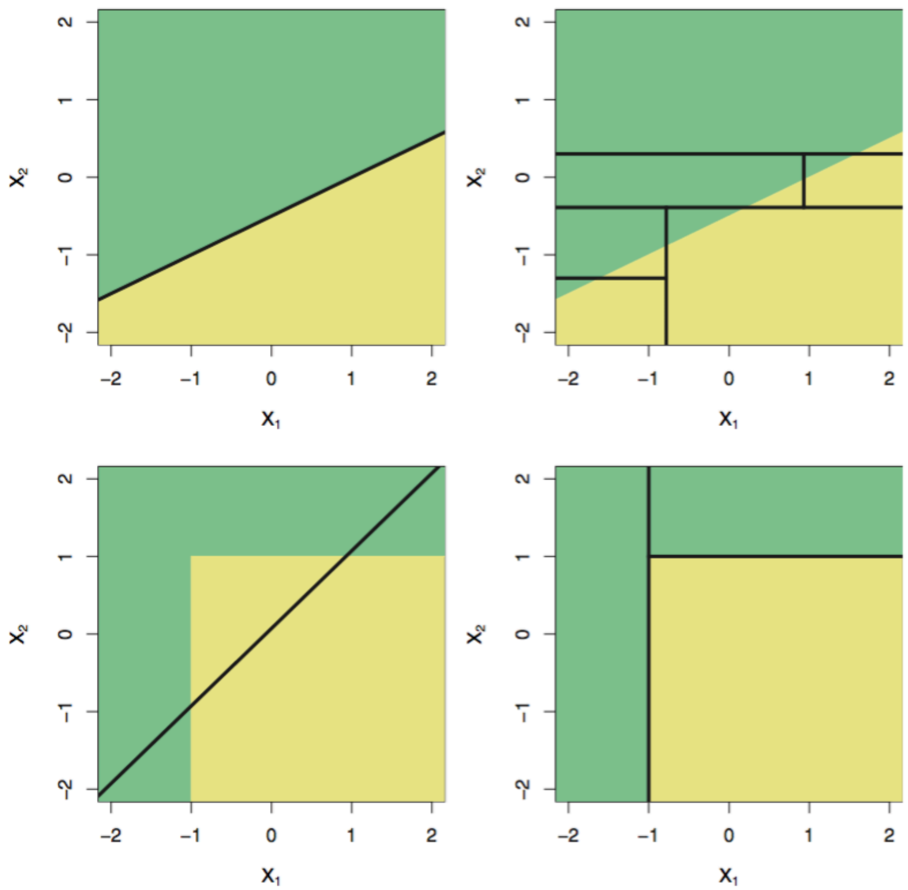



**FIGURE 8.7.** Top Row: *A two-dimensional classification example in which the true decision boundary is linear, and is indicated by the shaded regions. A classical approach that assumes a linear boundary (left) will outperform a decision tree that performs splits parallel to the axes (right).* Bottom Row: *Here the true decision boundary is non-linear. Here a linear model is unable to capture the true decision boundary (left), whereas a decision tree is successful (right).*

# Advantages of Decision Trees

- Trees are very easy to explain to people.
- Some people believe that decision trees more closely mirror human decision-making than do the regression and classification approaches.
- Trees can be displayed graphically, and are easily interpreted even by a non-expert (especially if they are small).
- Trees can easily handle qualitative predictors without the need to create dummy variables.

# Disadvantages of Decision Trees

- Trees generally do not have the same level of predictive accuracy as some of the other regression and classification approaches.
- Trees can be very non-robust. In other words, a small change in the data can cause a large change in the final estimated tree.

- To improve the predictive performance of trees, we learn:

  *bagging, random forest* and *boosting*