# Machine Learning in Public Health

## Lecture 7: Regression and Classification Trees

Dr. Yang Feng

# What we have learned so far

- Regression
  - *Linear Regression*
  - *Linear Regression (Selection)*
  - *Linear Regression (Regularization)*
  - *KNN*
- Classification
  - *Logistic Regression*
  - *LDA*
  - *QDA*
  - *KNN*
- Today: a new nonparametric machine learning method.

# Decision Trees

- *Decision trees* are supervised learning methods.
- They can be used for both regression and classification .
- They involve partitioning the predictor space into a number of simple regions (boxes, in particular).
- To make a prediction for a given observation, we typically use the mean (regression) or the mode (classification) of the training observations in the region to which it belongs.

# Predicting a baseball player's Salary

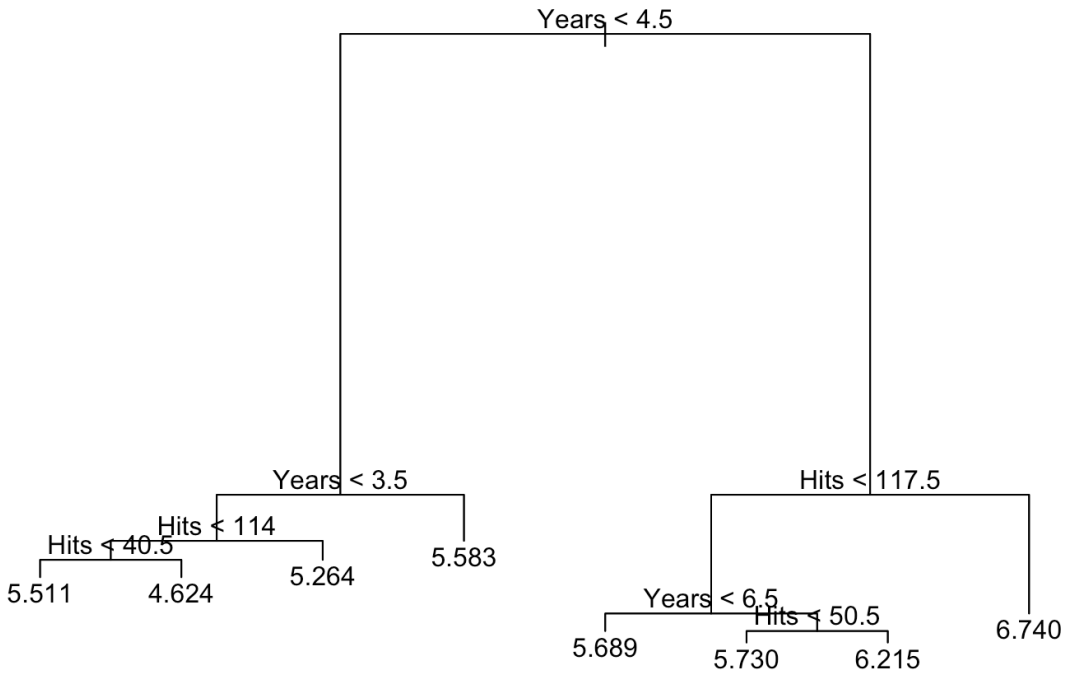
Using the Hitters data set. Based on Years (the number of years that he has played in the major leagues) and Hits (the number of hits that he made in the previous year).

The first few rows of the data

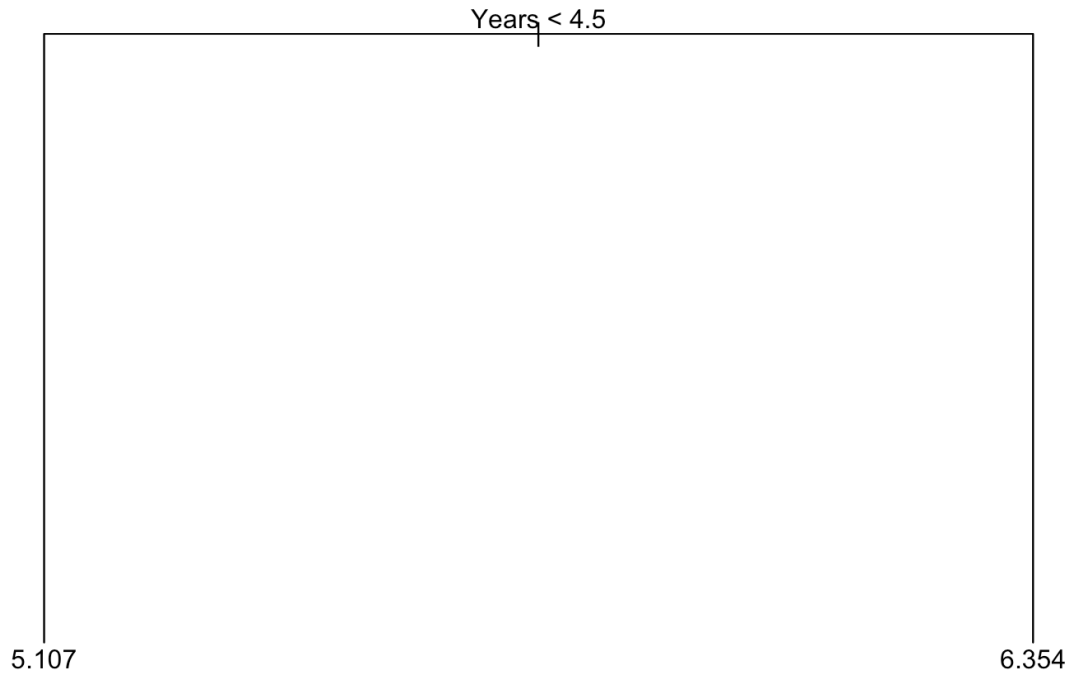
| | Salary <dbl> | Years <int> | Hits <int> |
|---|---|---|---|
| -Alan Ashby | 475.0 | 14 | 81 |
| -Alvin Davis | 480.0 | 3 | 130 |
| -Andre Dawson | 500.0 | 11 | 141 |
| -Andres Galarraga | 91.5 | 2 | 87 |
| -Alfredo Griffin | 750.0 | 11 | 169 |
| -Al Newman | 70.0 | 2 | 37 |

6 rows | 1-4 of 5 columns

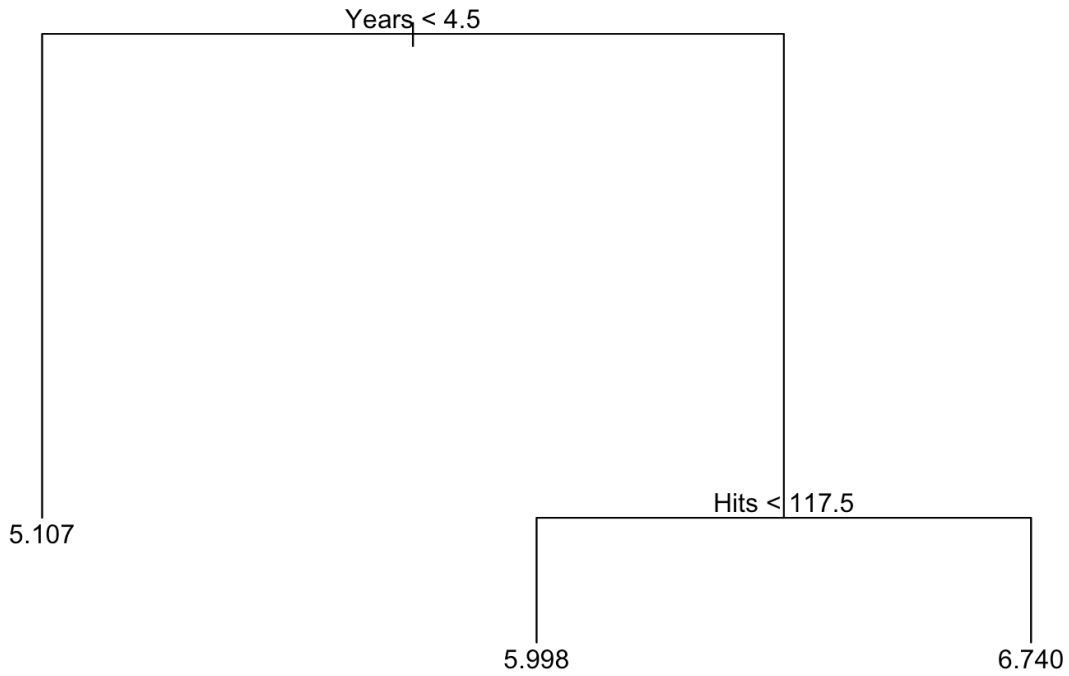
# Fit a Decision Tree



# Pruning the tree



# Pruning the tree



# The previous decision tree corresponds to a partition of the feature space

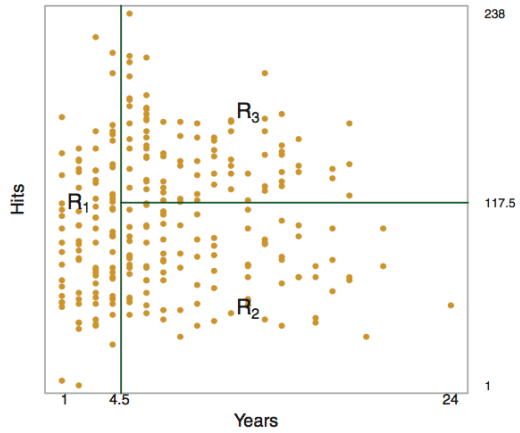



**FIGURE 8.2.** *The three-region partition for the* `Hitters` *data set from the regression tree illustrated in Figure 8.1.*

Leaves and prediction:

- $R_1 = \{X|Years < 4.5\}$->predicted log salary is 5.107
- $R_2 = \{X|Years >= 4.5, Hits < 117.5\}$->predicted log salary is

5.998

- $R_3 = \{X|Years >= 4.5, Hits >= 117.5\}$->predicted log salary is 6.740

- terminal nodes or leaves of the tree: $R_1, R_2, R_3$.

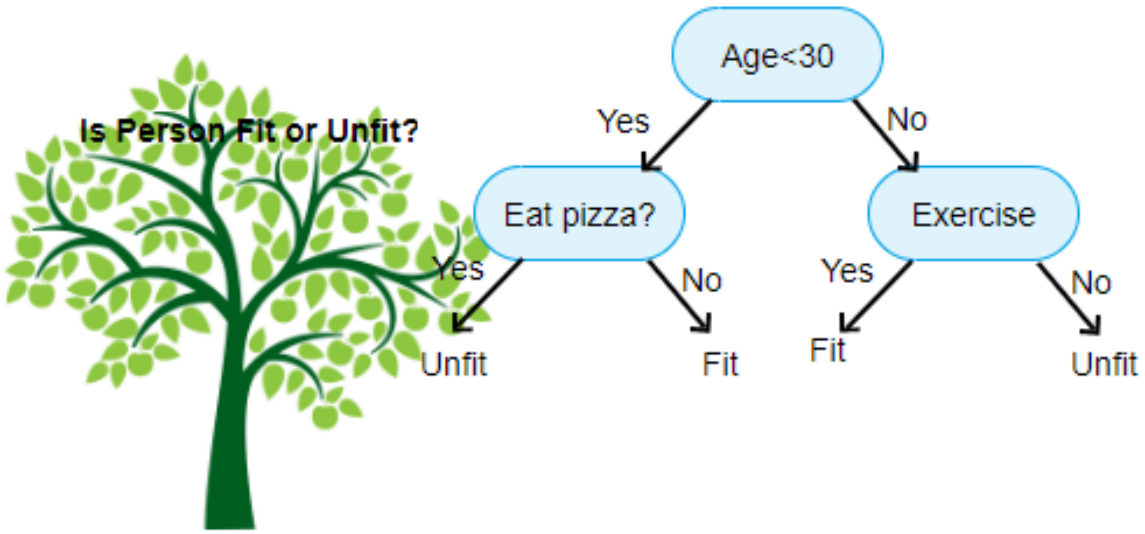
- internal nodes of the tree: $\{X|Years >= 4.5\}$.

# Some thoughts after seeing this first tree example

- In each split of the tree, we need to decide
  - *which variable to split?*
  - *where do we split this variable?*
- To answer the two above questions, we need a formal criterion
- In the least squares approach to linear regression, we used `RSS` as a criterion. Can we borrow it?
- Another question: we should think about a stopping rule. That is, when do we stop splitting?

# How to do the split?

- We take a top-down, greed approach that is known as **recursive binary splitting**.



# Recursive binary splitting

- It is a greedy approach because at each step of the tree-building process, the best split is made at that particular step, rather than looking ahead and picking a split that will lead to a better tree in some future step.
- For feature $j$ and split-point $s$, we define the pair of half-planes by

$$R_1(j, s) = \{X | X_j < s\}, \text{ and } R_2(j, s) = \{X | X_j \geq s\},$$

and set the values of $j$ and $s$ that minimize the equation

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

where $\hat{y}_{R_1}$ is the mean response for the training observations in $R_1(j, s)$, and $\hat{y}_{R_2}$ is the mean response for the training observations in

$R_2(j, s)$.

# When do we stop?

When we grow a very deep and bushy tree $T_0$, we have overfitted the training data (this tree has high variance or high bias?). To solve the problem:

- Stop when the region contains less than certain number of observations (e.g., *five*).
- Cost complexity pruning (weakest link pruning): consider a sequence of trees indexed by a non-negative tuning parameter $\alpha$. For each value of $\alpha$, there corresponds a *substree* $T \subset T_0$ such that

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

is the smallest.

- As $\alpha$ increases, we get a sequence of nested trees
- The next algorithm summarizes the entire tree building process

# Building a tree with CV and Pruning

**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 $k = 1, \ldots, K$:

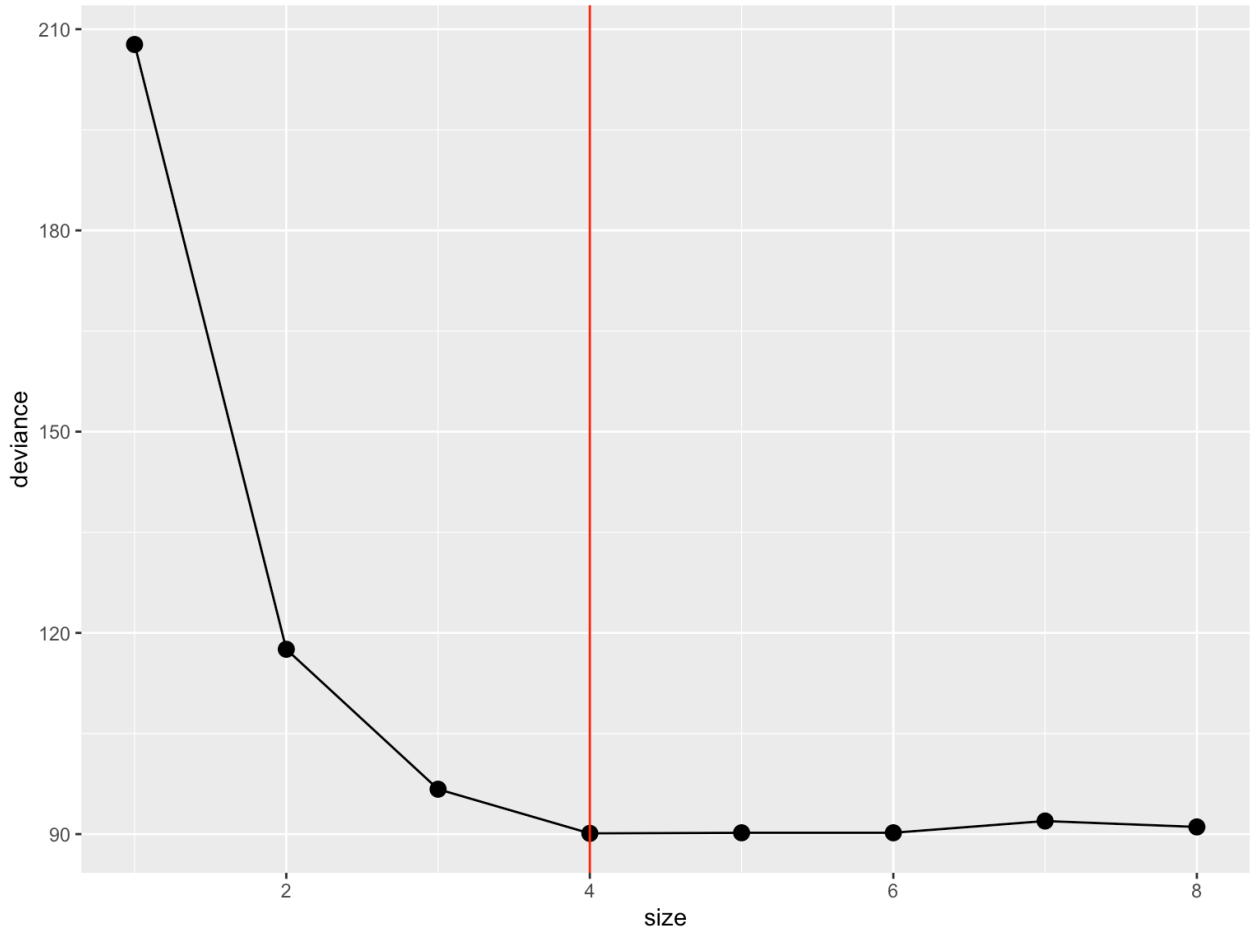
   (a) Repeat Steps 1 and 2 on all but the $k$th fold of the training data.

   (b) Evaluate the mean squared prediction error on the data in the left-out $k$th fold, as a function of $\alpha$.

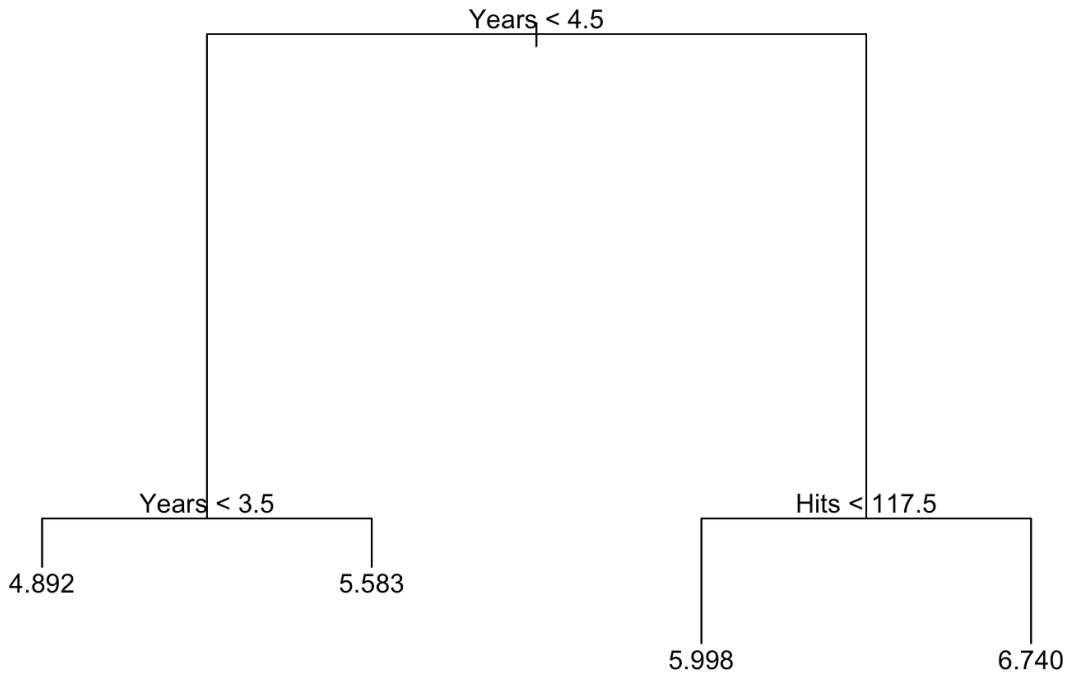
   Average the results for each value of $\alpha$, and pick $\alpha$ to minimize the average error.

4. Return the subtree from Step 2 that corresponds to the chosen value of $\alpha$.

# Cross-validation to predict the log Salary



# Visualizing the pruned regression tree



# Trees for classification

- `RSS` is not a proper criterion for classification problems
- The most natural and intuitive substitute is the classification error
- But it turns out that classification error is not sufficiently sensitive for tree-growing
- Suppose $\hat{p}_{mk}$ is the proportion of training observations in the $m$-th region that are from the $k$th class, and $K$ is the total number of classes
- In practice we use
  - *entropy* $D = -\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}$ *(Default option)*

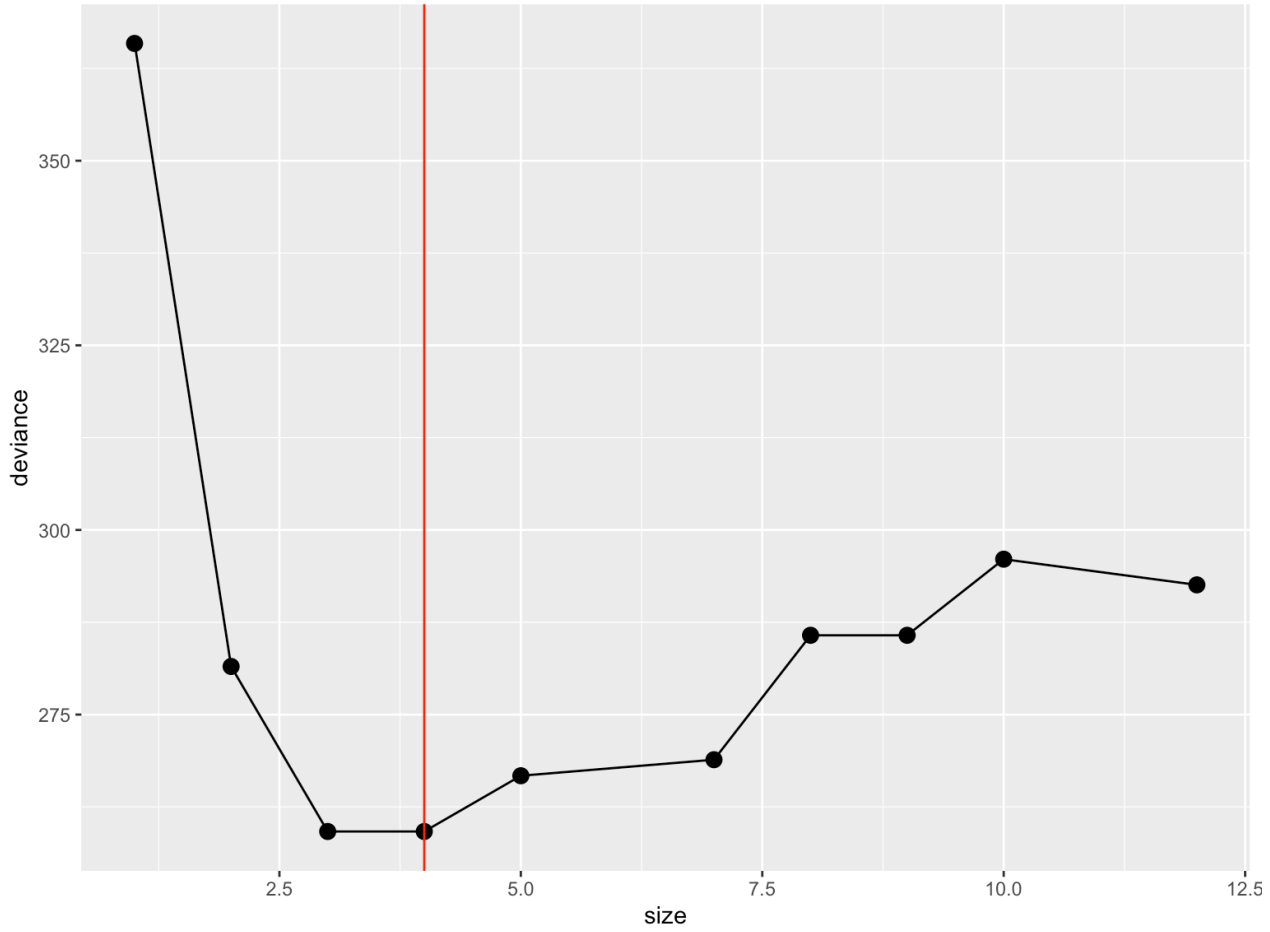
- *Gini index* $G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$ *(Set* `split = "gini"`*)*

- Both Gini index and entropy measure node purity

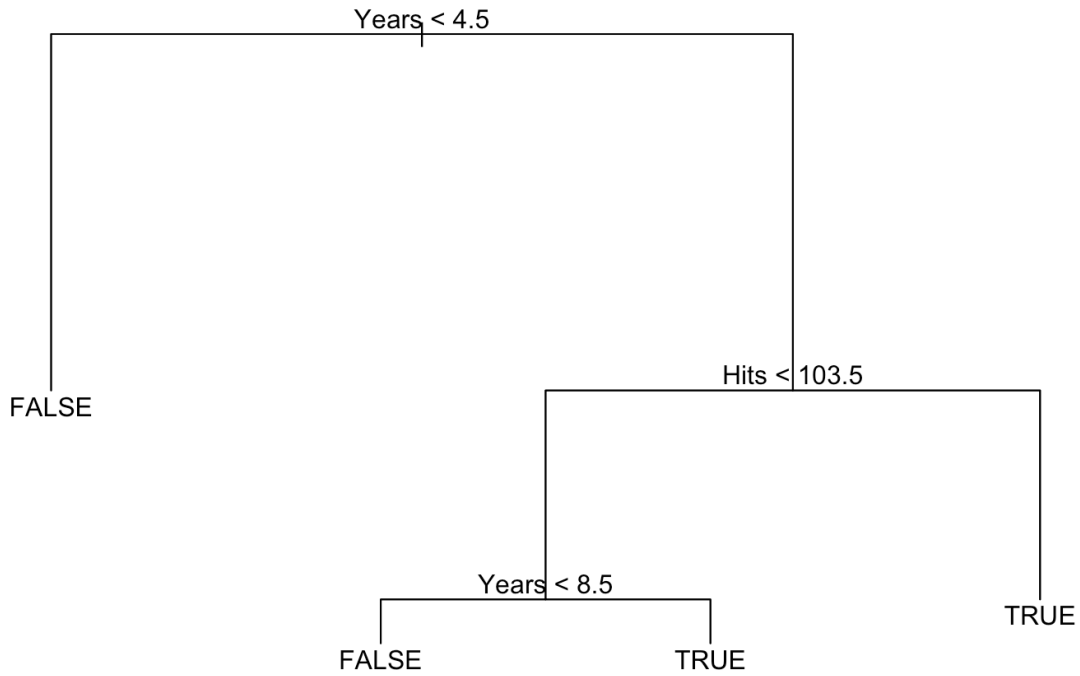
- When $K = 2$, what is the maximum and minimum values for Gini index?

# Example for predicting high salary

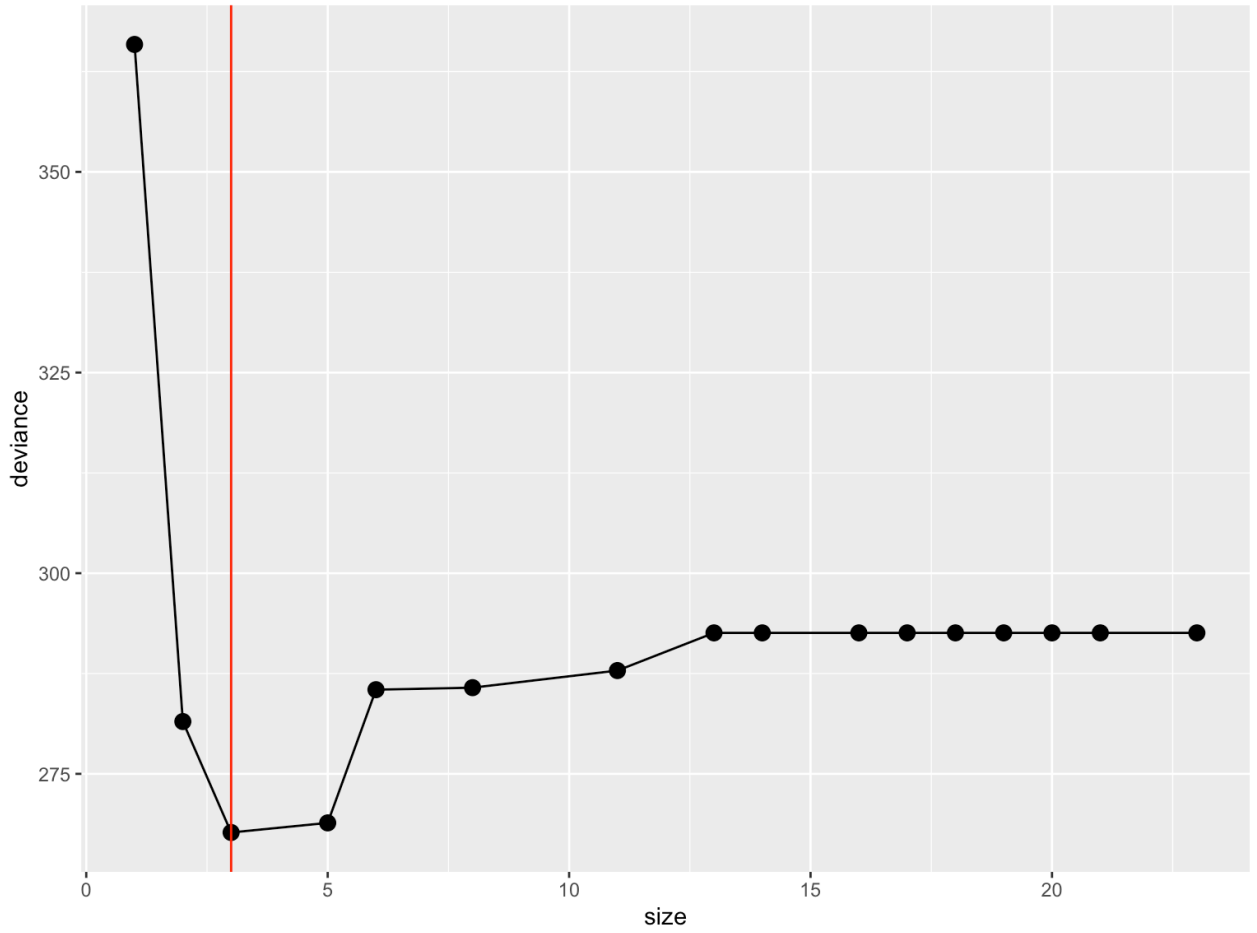
Create `salary <- salary %>% mutate(high_salary = logSalary > median(logSalary))`.



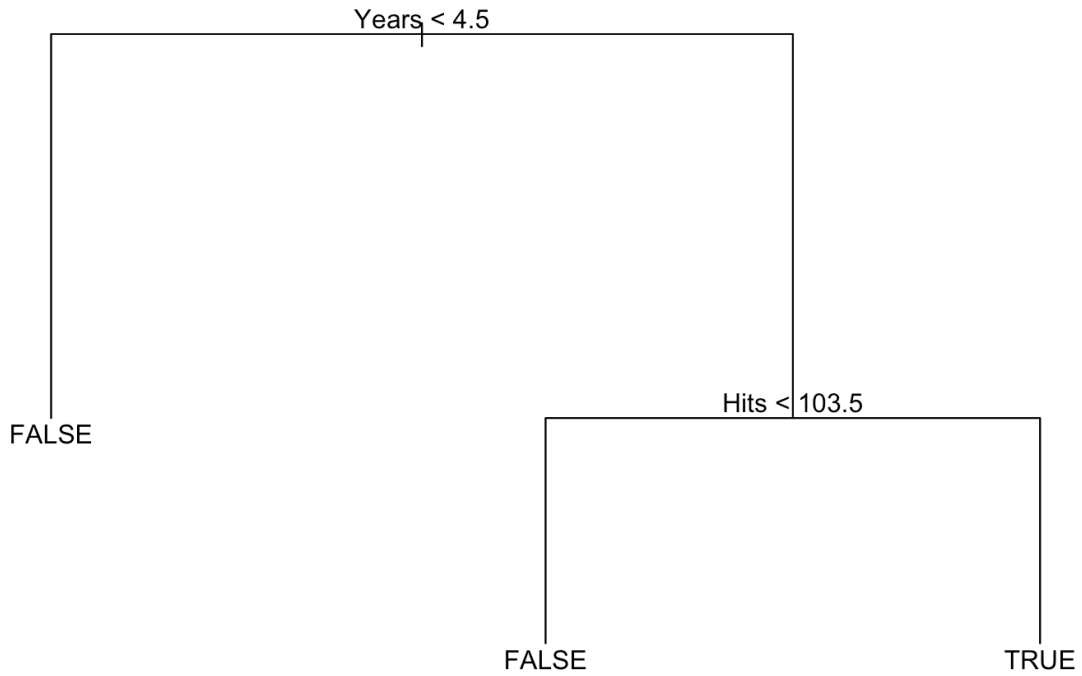
# Visualizing the pruned classification tree



# Using the Gini criterion to split



# Visualizing the pruned classification tree



# Advantages and disadvantages of trees

- Trees are very easy to explain to people. In fact, they are even easier to explain than linear regression!
- Why? No mathematical formula needed in the communication
- Some people believe that decision trees more closely mirror human decision-making than do the regression and classification approaches seen in previous chapters.
- Trees can be displayed graphically, and are easily interpreted even by a non-expert (especially if they are small).
- Unfortunately, tress generally do not have the great predictive accuracy.
- This disadvantage motivates the ensemble methods such as bagging,

random forests and boosting.

# Next class

- Bagging
- Random Forest
- Boosting