

# Decision Tree Exercises

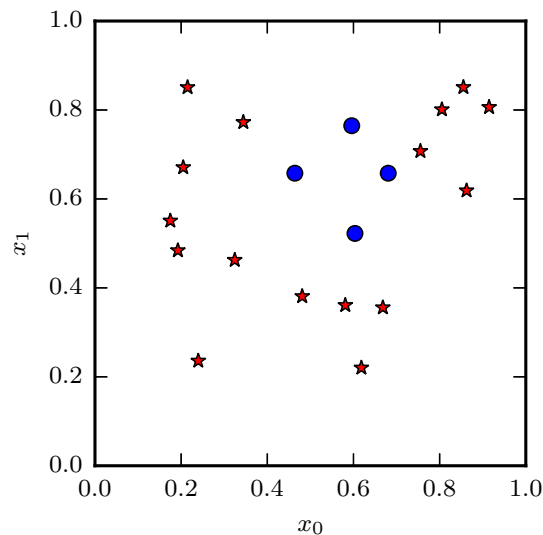
## 1. Gini Impurity

The goal in building a decision tree is to create the smallest possible tree in which each leaf node contains training data from only one class. In evaluating possible splits, it is useful to have a way of measuring the *purity* of a node. The purity describes how close the node is to containing data from only one class. Gini purity is defined as follows<sup>1</sup>:

$$\phi(\mathbf{p}) = \sum_i p_i(1 - p_i)$$

Where  $\mathbf{p} = (p_1, \dots, p_n)$  and each  $p_i$  is the fraction of elements from class  $i$ . This expresses the fractions of incorrect predictions in the node if the class of each element was predicted by randomly selecting a label according to the distribution of classes in the node. This value will be 0 if all elements are from the same class, and it increases as the mix becomes more uniform.

Calculate the Gini impurity of the following data set:



**Solution:** Assuming that the squares are class 1 and the stars are class 2, we have:

$$p_1 = \frac{4}{20} = .2$$

$$p_2 = \frac{16}{20} = .8$$

Therefore,

$$\phi(\mathbf{p}) = .2 \times .8 + .8 \times .2 = .32$$

<sup>1</sup>Notation follows: Breiman, Leo. "Technical note: Some properties of splitting criteria." Machine Learning 24.1 (1996): 41-47.

## 2. Tree Construction

The decision tree construction algorithm proceeds by recursively splitting the training data into increasingly smaller subsets. When splitting a node in the tree we search across all dimensions and all split points to select the split that results in the greatest decrease in impurity. This goodness-of-split value can be expressed as:

$$\Theta(s, t) = \phi(\mathbf{p}) - P_L \phi(\mathbf{p}_L) - P_R \phi(\mathbf{p}_R)$$

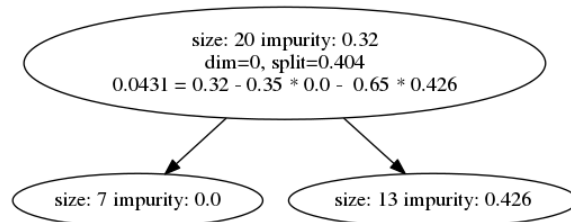
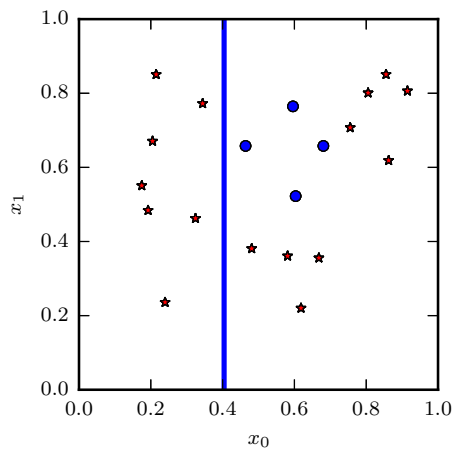
Where  $s$  is a possible split,  $t$  is the node and  $P_L$  and  $P_R$  represent the fraction of elements that ended up in the left child and right child respectively. Higher values represent better splits.

Execute the recursive tree-construction algorithm on the data above and draw the resulting tree. Calculate the impurity of each node and the goodness-of-split for each split.

### Solution:

The full tree construction algorithm would need to evaluate every possible split to choose the one with the largest goodness-of-split. For this exercise we can eyeball it to see that the best split will be the one that gets as many like elements as possible on the same side of the split:

#### SPLIT 1



The impurity on the left is:

$$\phi(\mathbf{p}_L) = 0 \times 1 + 1 \times 0 = 0$$

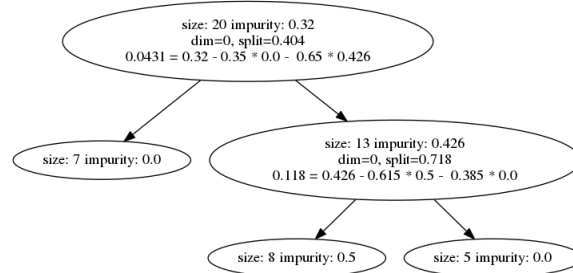
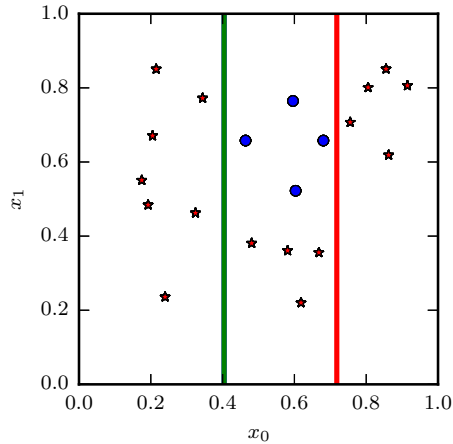
The impurity on the right is:

$$\phi(\mathbf{p}_R) = 4/13 \times 9/13 + 9/13 \times 4/13 = .426$$

This makes the goodness-of-split:

$$\begin{aligned} \Theta(s, t) &= .32 - P_L \times 0 + P_R \times .426 \\ &= .32 - .35 \times 0 + .65 \times .426 = .0431 \end{aligned}$$

#### SPLIT 2



The impurity on the left is:

$$\phi(\mathbf{p}_L) = 4/8 \times 4/8 + 4/8 \times 4/8 = .5$$

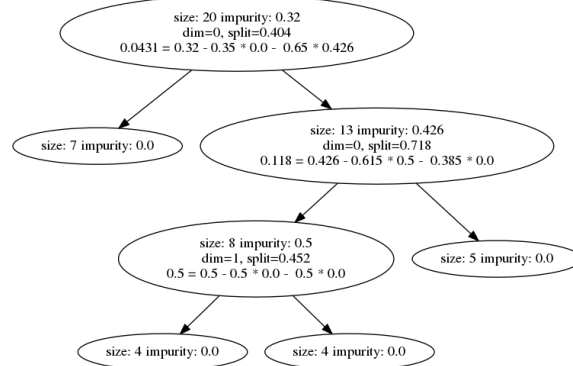
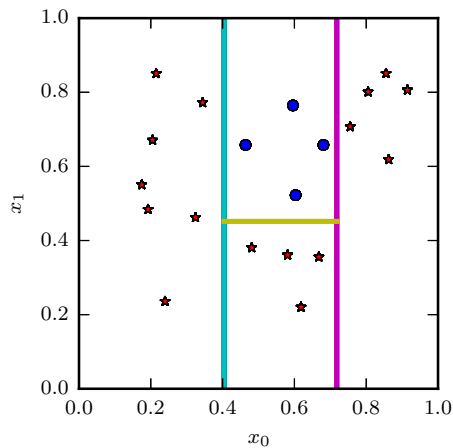
The impurity on the right is:

$$\phi(\mathbf{p}_R) = 0 \times 1 + 1 \times 0 = 0$$

This makes the goodness-of-split:

$$\Theta(s, t) = .426 - 8/13 \times .5 + 5/13 \times 0 = .118$$

### SPLIT 3



The impurity on the left is:

$$\phi(\mathbf{p}_L) = 1 \times 0 + 0 \times 1 = 0$$

The impurity on the right is:

$$\phi(\mathbf{p}_R) = 0 \times 1 + 1 \times 0 = 0$$

This makes the goodness-of-split:

$$\Theta(s, t) = .5 - 4/8 \times 0 + 4/8 \times 0 = .5$$

Since all nodes now have 0 impurity, no more splits are necessary.

### 3. Classification

Classify the following three points using your decision tree.

$(.4, 1.0)$

$(.6, 1.0)$

$(.6, 0)$

**Solution:**

$(.4, 1.0) \rightarrow 2(\textit{star})$

$(.6, 1.0) \rightarrow 1(\textit{circle})$

$(.6, 0) \rightarrow 2$