

An Example of Recursive Partitioning and Pruning

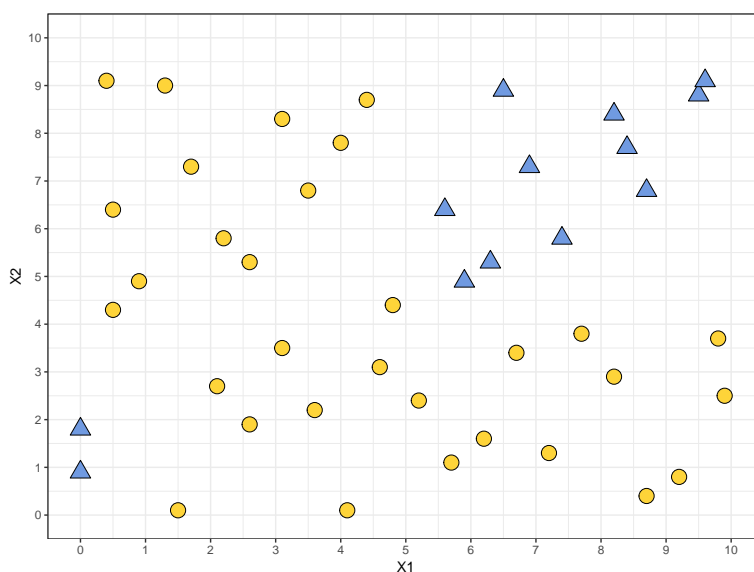
2019-04-06

In this activity you will learn more about the underlying algorithm for CART. To help in this endeavor, you will be creating a decision tree by mimicing the algorithm. Remember that an [algorithm](#) is “a process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer”. The algorithm that is used to create a decision tree is called *recursive partitioning*, and the rule that it uses is:

Partition (split) the data dichotomously (into two parts) to obtain the highest classification accuracy rate it can.

As you work through this activity to create the decision tree **you need to act like a computer**. In other words, you will not be trying to understand “why”, but rather only trying to find the data split that optimizes the classification accuracy.

Consider the following data that show an outcome (circles OR triangles) based on values for two predictors, X1 and X2.



X1	X2	Outcome	X1	X2	Outcome	X1	X2	Outcome	X1	X2	Outcome
0.0	0.9	Triangle	2.6	1.9	Circle	5.2	2.4	Circle	7.7	3.8	Circle
0.0	1.8	Triangle	2.6	5.3	Circle	5.6	6.4	Triangle	8.2	2.9	Circle
0.4	9.1	Circle	3.1	3.5	Circle	5.7	1.1	Circle	8.2	8.4	Triangle
0.5	4.3	Circle	3.1	8.3	Circle	5.9	4.9	Triangle	8.4	7.7	Triangle
0.5	6.4	Circle	3.5	6.8	Circle	6.2	1.6	Circle	8.7	0.4	Circle
0.9	4.9	Circle	3.6	2.2	Circle	6.3	5.3	Triangle	8.7	6.8	Triangle
1.3	9.0	Circle	4.0	7.8	Circle	6.5	8.9	Triangle	9.2	0.8	Circle
1.5	0.1	Circle	4.1	0.1	Circle	6.7	3.4	Circle	9.5	8.8	Triangle
1.7	7.3	Circle	4.4	8.7	Circle	6.9	7.3	Triangle	9.6	9.1	Triangle
2.1	2.7	Circle	4.6	3.1	Circle	7.2	1.3	Circle	9.8	3.7	Circle
2.2	5.8	Circle	4.8	4.4	Circle	7.4	5.8	Triangle	9.9	2.5	Circle

First Decision Rule

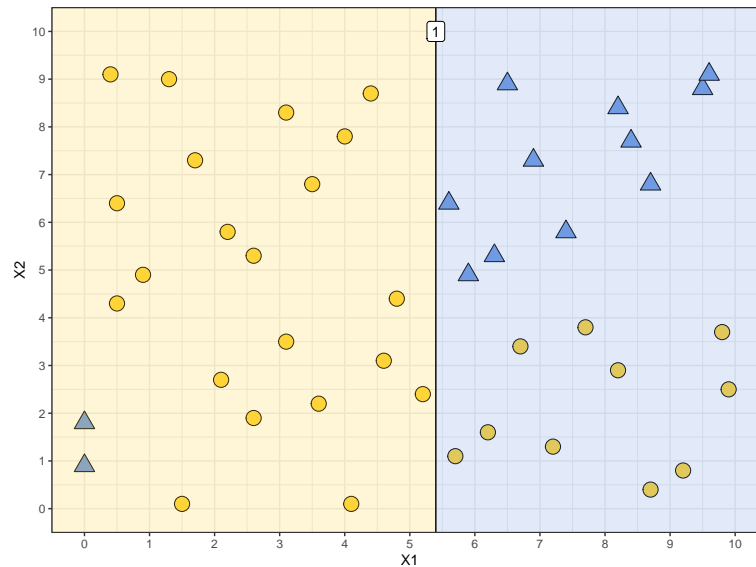
The first step in the recursive partitioning algorithm is to partition the data into two parts using *a single vertical or horizontal line*. This line represents a decision rule for classifying cases on one side of the line as “triangles” and cases on the other side of the line as “circles”. The algorithm will decide where to put this line by choosing the line that results in the highest classification accuracy possible for the data.

1. Draw *a single horizontal or vertical line* through the data that results in the highest possible classification accuracy for the data. Indicate the side of the line that is to be classified as “triangles” and that which will be classified as “circles”.

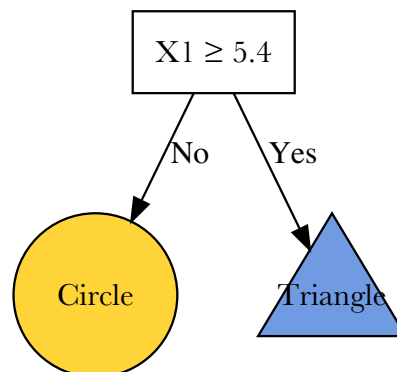
In our example, the first partition of the data is based on the rule:

Rule 1: IF ($X_1 \geq 5.4$) THEN (classify as “triangle”) ELSE (classify as “circle”).

Graphically this looks like this:



This rule results in an overall classification accuracy of $32/44 = 0.73$. This rule is subsequently the first (top) rule in your decision tree. For example,



Second Set of Decision Rules

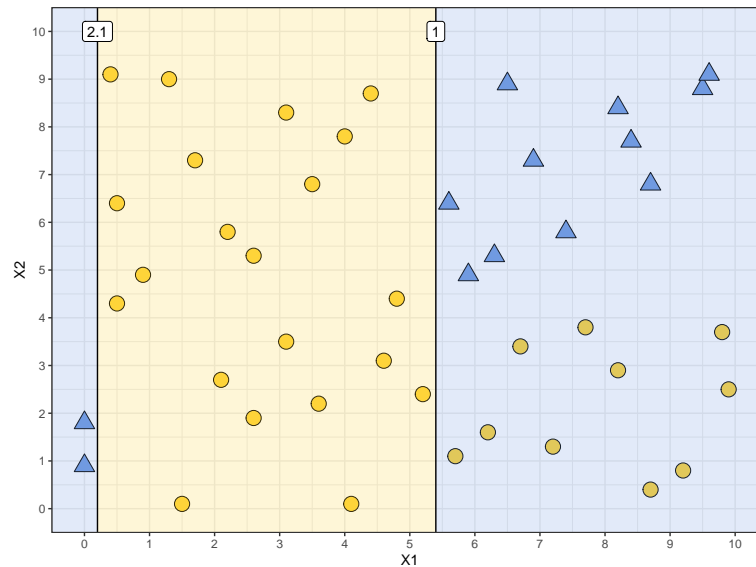
Based on the first decision rule the algorithm has partitioned the data into two parts. The algorithm now continues by treating each partition as a “new” plot. For each of these new plots, it will re-apply itself; again finding a single vertical or horizontal line to classify cases on one side of the line as “triangles” and cases on the other side of the line as “circles” and choosing that line such that it results in the highest classification accuracy possible for the data in that “new” plot.

3. Consider the cases that were classified as circles by your first decision rule (those with an X_1 value less than 5.4). Draw *a single horizontal or vertical line* through the data that results in the highest possible classification accuracy for those cases. Indicate the side of the line that is to be classified as “triangles” and that which will be classified as “circles”.

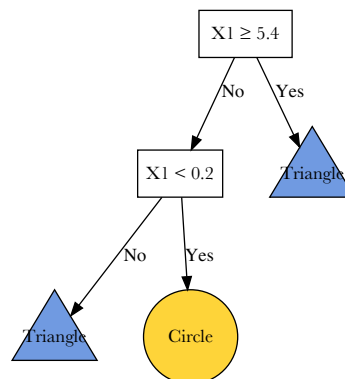
In our example, the second partition of the data for the cases previously classified as circles is based on the rule:

Rule 2.1: IF ($X1 < 0.2$) THEN (classify as “triangle”) ELSE (classify as “circle”).

Graphically this looks like this:



There are now two decision rules partitioning the data which, in our example, result in an overall classification accuracy of $34/44 = 0.77$. The tree diagram after including the second decision rule is as follows:

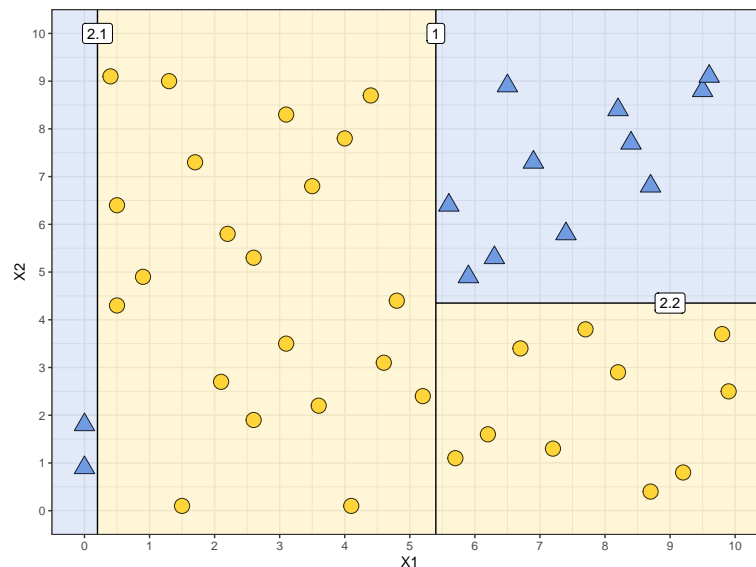


4. Consider the cases that were classified as triangles by your first decision rule (those with an $X1$ value greater than or equal to 5.4). Draw *a single horizontal or vertical line* through the data that results in the highest possible classification accuracy for those cases. Indicate the side of the line that is to be classified as “triangles” and that which will be classified as “circles”.

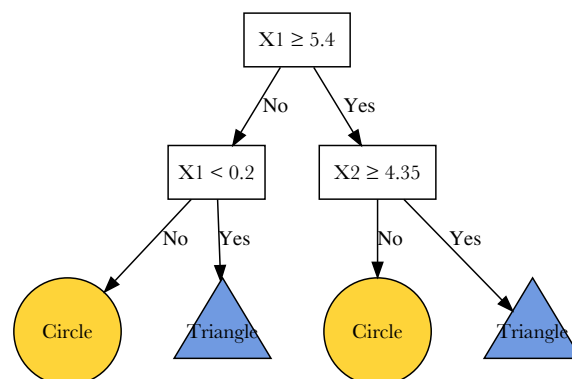
In our example, this partition of the data for the cases initially classified as triangles is based on the rule:

Rule 2.2: IF ($X_2 \geq 4.35$) THEN (classify as “triangle”) ELSE (classify as “circle”).

Graphically this looks like this:



There are now three decision rules partitioning the data which, in our example, result in an overall classification accuracy of $44/44 = 1.00$. The tree diagram after including the second decision rule is as follows:



More Decision Rules

We have perfectly classified 100% of the training data with our three decision rules. If we hadn't, the algorithm would have continued. For example, each of the decision rules we previously added resulted in “new” plots that could be further partitioned. The algorithm would look at any “new” plot that does not have 100% accuracy to see if it can be partitioned in a way that increases accuracy. Are you starting to understand why the algorithm is referred to as *recursive partitioning*?

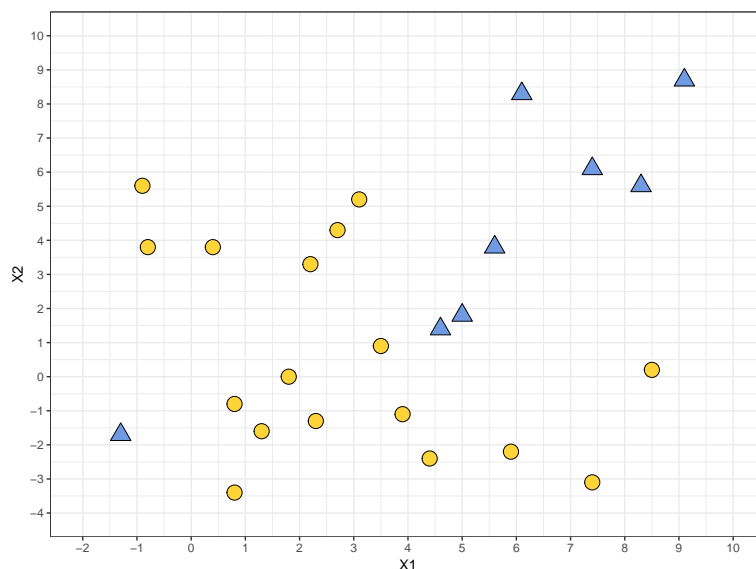
Tree Pruning

The decision tree resulting from the recursive partitioning carried out on the training data is overfitted to the training data. (After all, it ended up with 100% classification accuracy, and would likely not classify 100% of new cases perfectly.) To help reduce the problem of overfit, we will **prune** the decision tree.

The first decision rule in the tree was the most important in that it provided the highest initial classification accuracy. The second set of decisions was then next most important, and so on down to the last set of decision rules. The idea of pruning is that we prune, or trim, decision rules from the tree starting with the least important rules (the last rules added).

Recall that any evaluation of the decision tree needs to be carried out using a new set of data, the *validation data*. As pruning is a part of the evaluation process, it also needs to be carried out using validation data. Below we provide you with 25 new cases to use in this endeavor.

X1	X2	Outcome	X1	X2	Outcome
-1.3	-1.7	Triangle	3.9	-1.1	Circle
-0.9	5.6	Circle	9.1	8.7	Triangle
-0.8	3.8	Circle	4.4	-2.4	Circle
0.8	-3.4	Circle	4.6	1.4	Triangle
0.4	3.8	Circle	5	1.8	Triangle
0.8	-0.8	Circle	5.6	3.8	Triangle
1.3	-1.6	Circle	5.9	-2.2	Circle
1.8	0.0	Circle	6.1	8.3	Triangle
2.2	3.3	Circle	7.4	-3.1	Circle
2.3	-1.3	Circle	7.4	6.1	Triangle
2.7	4.3	Circle	8.3	5.6	Triangle
3.1	5.2	Circle	8.5	0.2	Circle
3.5	0.9	Circle			



5. Begin the process of evaluation by computing the classification accuracy for the model that classified 100% of the training cases correctly (includes Rule 1, 2.1, and 2.2). Note: *Do not update or change any of the decision rules. Just use them to compute the classification accuracy.*

Our next step in the pruning process is to remove a single decision rule from the bottom of the tree. In our tree there are two decision rules at the same stage (Stage 2) at the bottom of the tree. We only remove one of those rules to begin with. (Which of the two you begin with is up to you.) For consistency, we will remove the rule on the right-side of the tree, Rule 2.2.

6. Compute the classification accuracy for the model that includes all rules except Rule 2.2.

We can now compare the classification accuracies for the model that includes all the decision rules and the model that omits Rule 2.2. This comparison will allow us to evaluate whether or not including Rule 2.2 *improves* the classification accuracy. Our threshold for improvement will be: *to include the rule we need to improve the classification accuracy by at least 4 percentage points*. If the classification accuracy does not improve by at least 4 percentage points that is evidence for pruning that rule.

7. Compare the classification accuracies you computed in Questions 5 and 6. Should we prune Rule 2.2 or leave Rule 2.2 in the model? Explain.

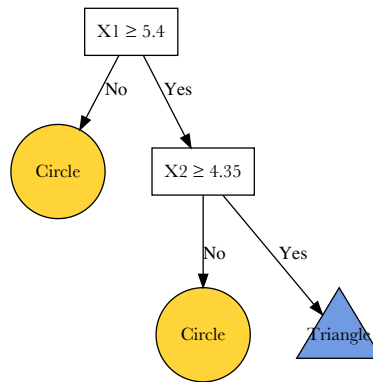
Including Rule 2.2 resulted in an improvement in classification accuracy of 8 percentage points. Thus we will keep Rule 2.2 in the model, as well as all the decision rules that precede Rule 2.2 in the tree. By keeping Rule 2.2, we also have to keep Rule 1 since it is further up the tree from Rule 2.2.

What about Rule 2.1? That rule, which was also added in Stage 2, does not precede Rule 2.2. Thus it is a candidate for pruning. Should Rule 2.1 be pruned or not? To determine whether or not to prune Rule 2.1, we need to compare the tree based on all the rules we are keeping (Rule 1 and Rule 2.2) to a tree that includes those rules and Rule 2.1 (e.g., a tree that includes Rules 1, 2.1, and 2.2).

8. Compute the classification accuracy for the two models that allow you to evaluate whether or not to prune Rule 2.1.

9. Use the classification accuracies you computed in Question 8 to evaluate whether or not to prune Rule 2.1. Explain.

Removing Rule 2.1 actually resulted in an improvement in classification accuracy of 4 percentage points. Therefore we will prune Rule 2.1. The tree for the resulting model is given below.



10. Based on the tree, and our evaluation so far, are there any more decision rules we can consider for pruning? Explain.