← Go Back to Practical Data Science

Course Content

Tree Pruning

In the previous week, we learned about the bias-variance tradeoff and the problem of overfitting.

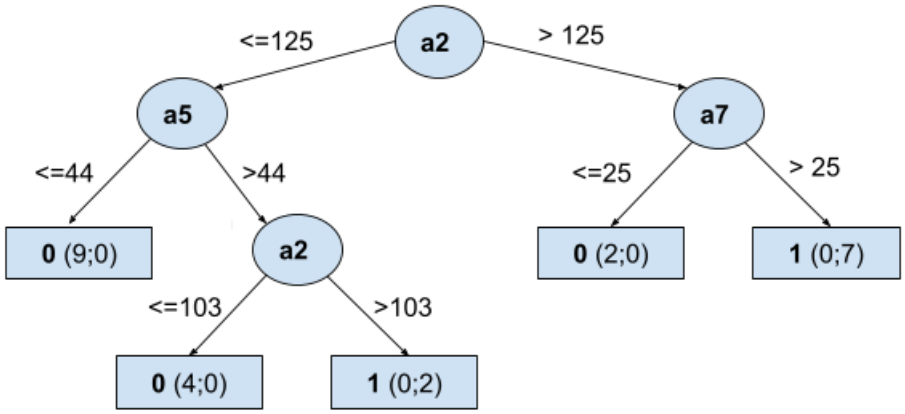
We know that Decision Trees make splits to get maximum gain. As the tree grows large, the tendency to overfit on the train data increases because, in a large tree, splits are made even to get small gains.

One of the techniques used to handle the overfitting problem of Decision Trees is **Tree Pruning**.

Tree Pruning

Pruning **selectively removes** certain parts of a tree to improve the tree's structure and reduce overfitting. It reduces the size of a Decision Tree, a step that may slightly increase your training error but may also drastically decrease your testing error, making it more adaptable to new, unseen data.

Consider the decision tree below, which initially has a depth equal to 3. In the below figure, the decision nodes are represented as circles and the leaf nodes are represented as rectangles.

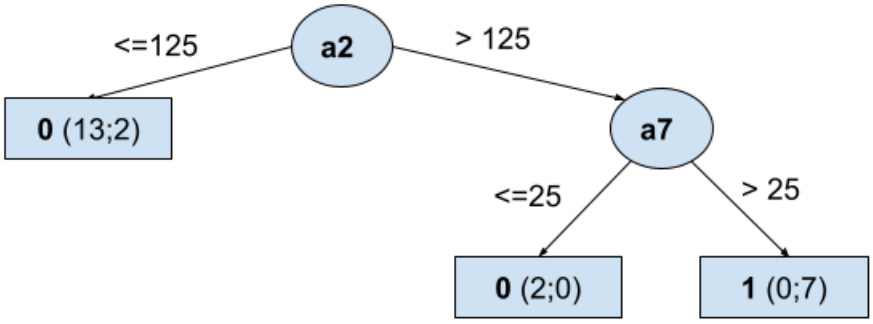


a) The initial tree

The pruning algorithm starts at the leaf node in a decision tree. Following recursively upwards, it determines the significance of each subtree. In the above tree, assume that the subtree at the decision node **a5** is insignificant, then this subtree can be replaced by a leaf node.

In this subtree, we have 9 samples in the left leaf node of **a5** and 4 samples in the left leaf node of **a2** that belong to class 0. So, in total there are 13 samples which belong to class 0 in this subtree. Similarly, there are 2 samples that belong to class 1 in this subtree.

After pruning, all the samples (15=13+2) are assigned to a single leaf node where the most common class will be the prediction of that leaf node, i.e., class 0 in this case.



b) Tree after pruning

Note that the pruned tree has a depth equal to 2. Thus, pruning reduces the size of a decision tree.

Previous

Next