SciKit-Learn and Decision Trees

MSXPPDSX2016-V005100

Dive Deeper

0:00 / 7:16 1.0x

Decision Trees are a very easy to use and abuse classifier, so trend cautiously. Setting them up in SciKit-Learn should be very familiar by now:

```
# You know the drill...
>>> from sklearn import tree
>>> model = tree.DecisionTreeClassifier(max_depth=9, criterion="entropy")
>>> model.fit(X,y)
DecisionTreeClassifier(class_weight=None, criterion='entropy', max_depth=9,
            max_features=None, max_leaf_nodes=None, min_samples_leaf=1,
            min_samples_split=2, min_weight_fraction_leaf=0.0,
            presort=False, random_state=None, splitter='best')

# .DOT files can be rendered to .PNGs, if you've already `brew install graphviz`.
>>> tree.export_graphviz(model.tree_, out_file='tree.dot', feature_names=X.columns)

>>> from subprocess import call
>>> call(['dot', '-T', 'png', 'tree.dot', '-o', 'tree.png'])
```

SciKit-Learn's trees are quite configurable:

- **criterion** By default, SciKit-Learn uses Gini, which is an impurity rating. Alternatively, you could also make use of information gain, or entropy instead.
- **splitter** Lets you control of the algorithm chooses the best split or not. We'll discuss why that's importance once you move to random forest classifier.
- **max_features** One of the possible splitter options for splitter above is called 'best'. SciKit-Learn runs a bunch of tests on your features to figure out which mechanism should be used when searching for the best split. This parameter limits the number of features to consider while doing this.

After you've gone ahead and trained your tree, you can of course get back all the end-node, leaf classifications that the tree has reached, as well as the entire tree object if you like. For those leaf nodes that aren't 100% pure due to having samples belonging to multiple classes within them, then the end-result class the leaf takes is a weighted mode vote, based on the number of each class label inside of it.

You can also get back a **feature_importances** vector that stores, in order of importance, the features that used to make the labeling decisions of your tree.