Ensembles

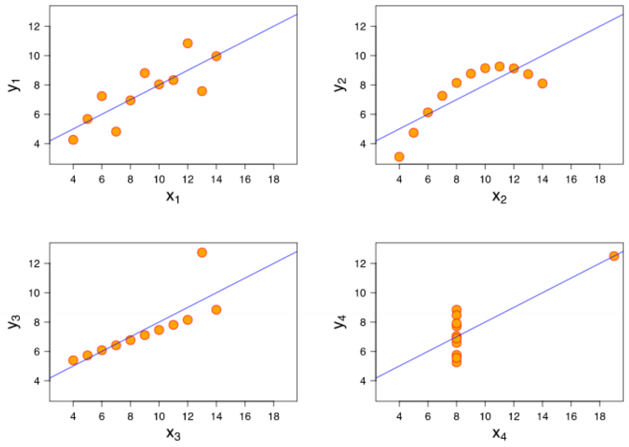
This whole lesson (on ensembles) is about how we can combine (or ensemble) the models you have already seen in a way that makes the combination of these models better at predicting than the individual models.

Commonly the "weak" learners you use are decision trees. In fact the default for most ensemble methods is a decision tree in sklearn. However, you can change this value to any of the models you have seen so far.

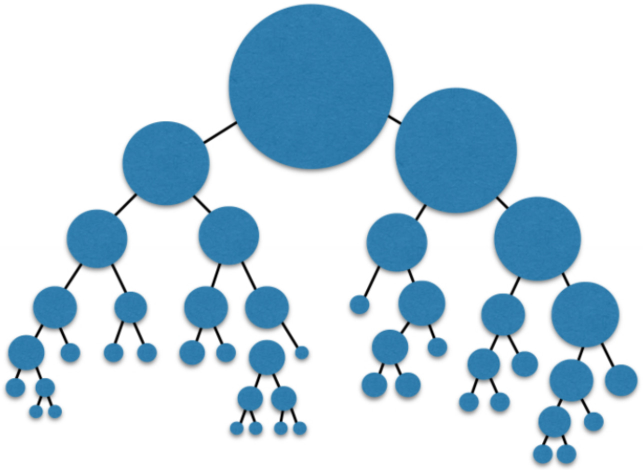
Why Would We Want to Ensemble Learners Together?

There are two competing variables in finding a well fitting machine learning model: **Bias** and **Variance**. It is common in interviews for you to be asked about this topic and how it pertains to different modeling techniques. As a first pass, the wikipedia is quite useful. However, I will give you my perspective and examples:

Bias: When a model has high bias, this means that means it doesn't do a good job of bending to the data. An example of an algorithm that usually has high bias is linear regression. Even with completely different datasets, we end up with the same line fit to the data. When models have high bias, this is bad.



Variance: When a model has high variance, this means that it changes drastically to meet the needs of every point in our dataset. Linear models like the one above has low variance, but high bias. An example of an algorithm that tends to have high variance and low bias is a decision tree (especially decision trees with no early stopping parameters). A decision tree, as a high variance algorithm, will attempt to split every point into its own branch if possible. This is a trait of high variance, low bias algorithms - they are extremely flexible to fit exactly whatever data they see.



By combining algorithms, we can often build models that perform better by meeting in the middle in terms of bias and variance. There are some other tactics that are used to combine algorithms in ways that help them perform better as well. These ideas are based on minimizing bias and variance based on mathematical theories, like the central limit theorem.

Introducing Randomness Into Ensembles

Another method that is used to improve ensemble methods is to introduce randomness into high variance algorithms before they are ensembled together. The introduction of randomness combats the tendency of these algorithms to overfit (or fit directly to the data available). There are two main ways tendency of these algorithms to overfit (or fit directly to the data available). There are two main ways that randomness is introduced:

1. **Bootstrap the data** - that is, sampling the data with replacement and fitting your algorithm to the sampled data.
2. **Subset the features** - in each split of a decision tree or with each algorithm used in an ensemble, only a subset of the total possible features are used.

In fact, these are the two random components used in the next algorithm you are going to see called **random forests**.

NEXT