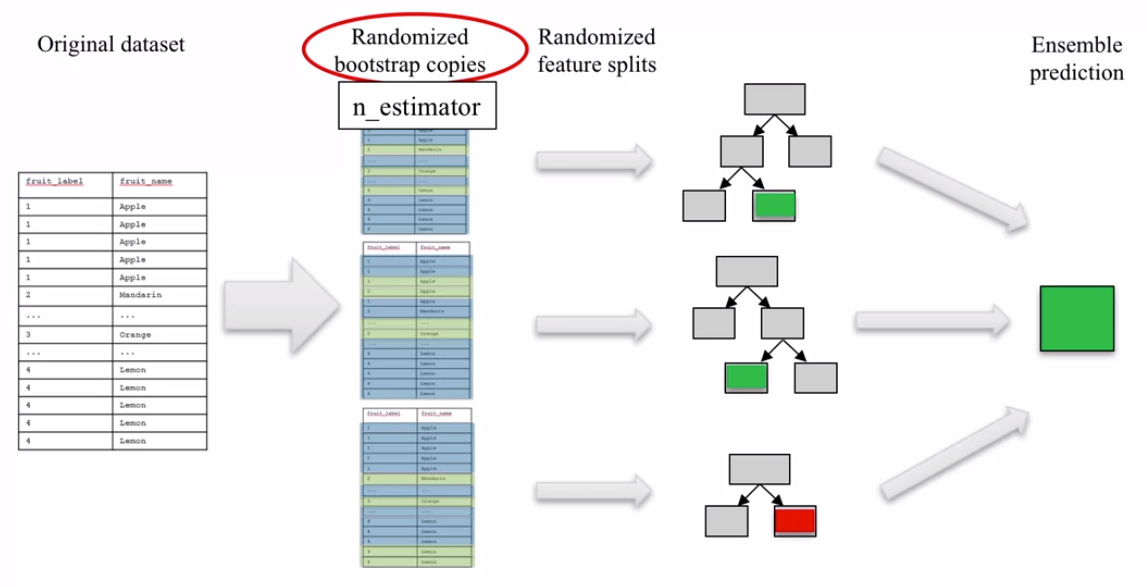
**Random Forests:**

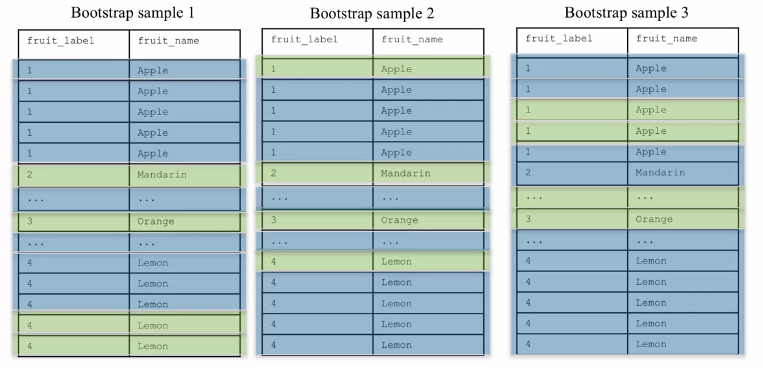
An **ensemble** machine learning algorithm takes individual algorithms and then combines them together to produce an aggregate model that is more powerful. The reason this works is because each model will make unique mistakes by either overfitting to different parts of the data. By combining these models, we can **average out** their **individual mistakes** to reduce the risk of overfitting while **maintaining strong prediction** performance.

Random forests use this ensemble method with decision trees, this method is very effective and achieves a good result on a wide variety of problems. The decision trees are built using random variations.

**How is a random forest built?**

First the data to build each decision tree is selected randomly, secondly the features chosen in each split are randomly selected.





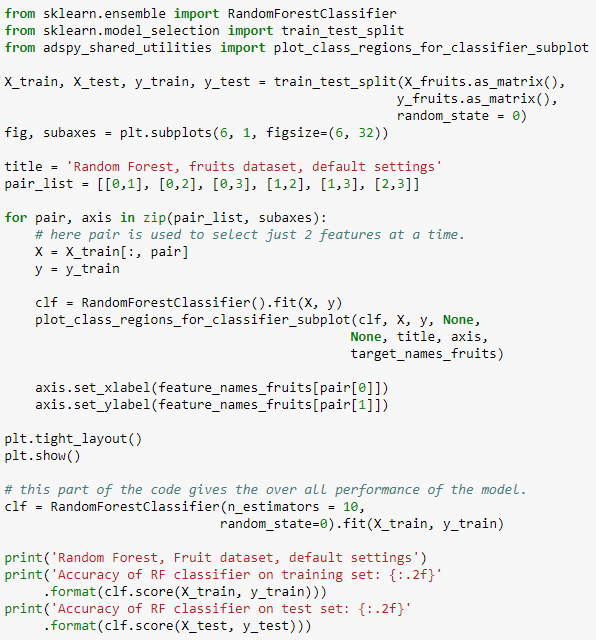
To build a random forest you first decide how many decision trees you’re going to build, this is set by the “n\_estimator” parameter. Each of these trees is built from a different random sample of the data called the **bootstrap** **sample**. What is bootstrapping? This is a new data set created from the original set, the method randomly picks rows of data from the original, while allowing for duplication, until the new data set is the same size as the original one. You then repeat this process to create N different bootstrapped data sets, each new data set will be used to create one decision tree. However, there is one important but significant difference when creating the decision tree. When picking the best split for a node, instead of finding the best split across all possible features, a random subset of features is chosen, and the best split is found within that smaller subset of features. The number of features that a decision tree can chose from is controlled by the “max\_features” parameter, this in turn determines how many “layers” the individual decision trees have. If the route node selection is suboptimal the decision tree will require many levels to achieve a prediction.

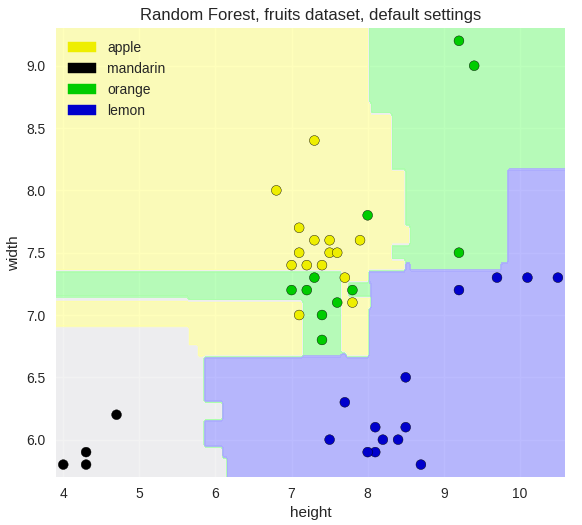
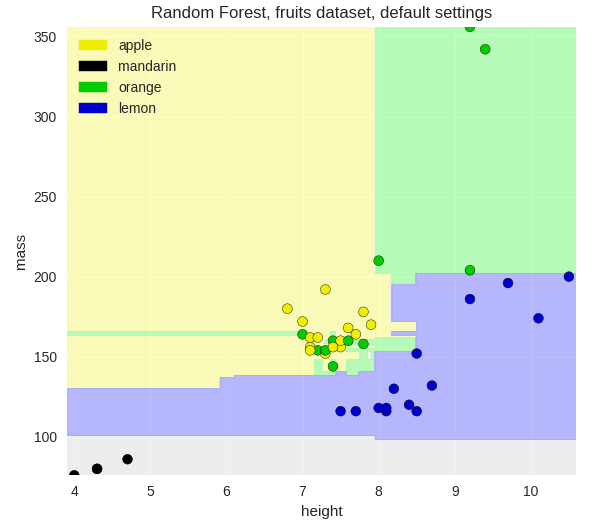
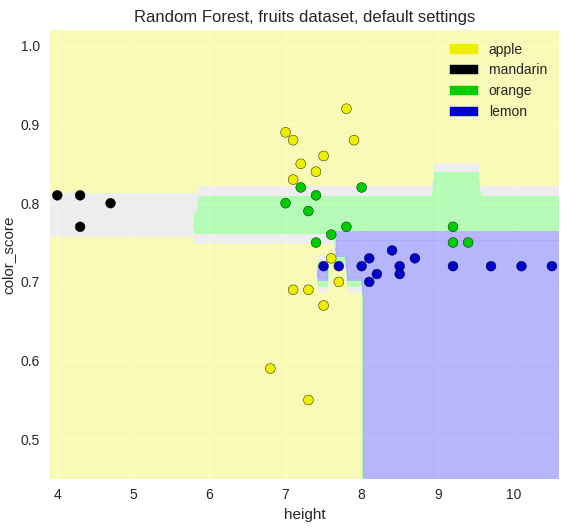
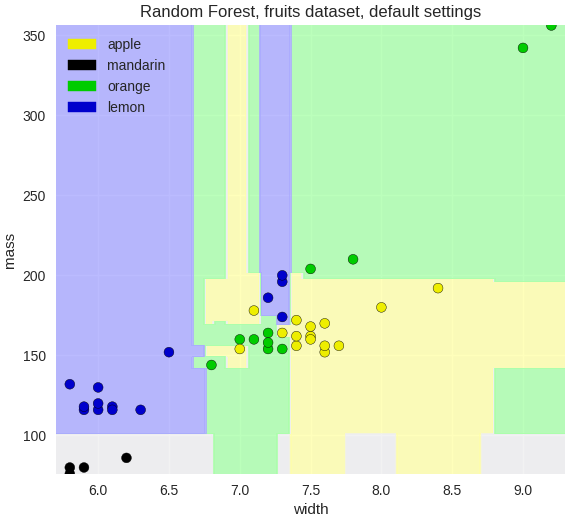
* Small Max features = forests with diverse, more complex trees.
* Large max features = similar forests with simpler trees.

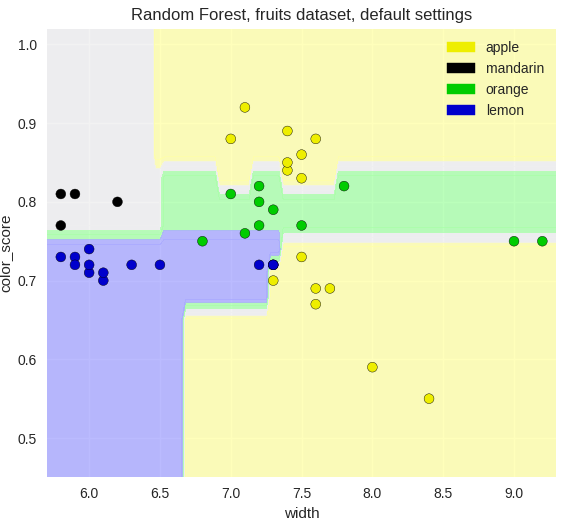
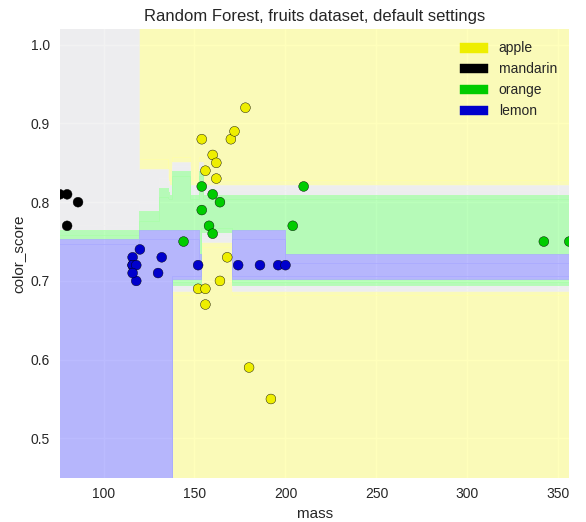
The predicted outcome from the decision trees is then found from some form of combination of all the decision trees, for regression problems this is often the mean value, for classification the prediction is determined by a weighted vote.

**Example:**

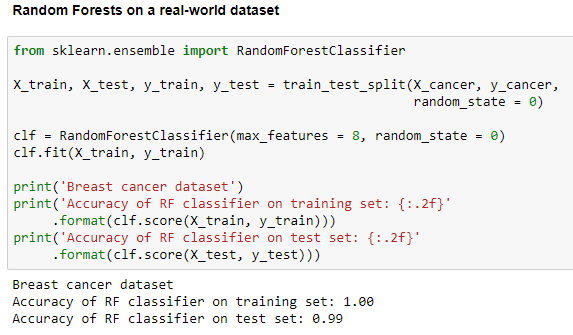
The following example uses the fruits data set:







**Note that the results with no feature scaling or parameter tuning produces very good test set performance!**

* You don’t have to scale the data to be used in a random forest!

