**Random Forest:**

<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>

Breiman, the creator of random forests and bagging, states that the vital element for gaining accuracy thanks to bagging is the **instability** of the prediction method.

Definition of Stability: How much an algorithm will change based on a small change in input: **Unstable Learners**: Decision Trees, Neural Networks **Stable Learners**: K-Nearest Neighbors, Support Vector Machines, Regularized Least Squares Regression

Most explanations online for why RF’s work is that: A) Random feature splits allow to create many uncorrelated trees which focus on different parts of the problem. B) Bagging of uncorrelated trees “average” out and reduces the variance error term.

Roughly speaking, some of the potential over-fitting that might happen in a single tree (which is a reason you do pruning generally) is mitigated by two things in a Random Forest:

1. The fact that the samples used to train the individual trees are "bootstrapped".
2. The fact that you have a multitude of random trees using random features and thus the individual trees are strong but not so correlated with each other.

Bagging is not very friendly to mathematical analysis. The fact is, random forests rose because of empirical successes rather than sound mathematical theory (Citation). There are many theories: [#1](https://homes.cs.washington.edu/~pedrod/papers/kdd97.pdf), [#2](http://www.math.univ-toulouse.fr/~agarivie/Telecom/apprentissage/articles/BaggingML.pdf), #3. I’ll explain two theories: 1) The most popular “variance reduction” one 2) A secondary theory that states “variance reduction” is a side effect of equalizing **leverage points**(more on this later)

**Theory #1**: Decision Trees suffer from high variance. Bagging improves variance by averaging from multiple different trees on variants of the training set, which helps the model see different parts of the problem. This variance-reduction argument was introduced by this [paper](http://robotics.stanford.edu/~ronnyk/vote.pdf).

Parameters:

1. **n\_estimators :**
2. **criterion:**
3. **max\_depth:**
4. **min\_samples\_split :**
5. **min\_samples\_leaf :**
6. **max\_features :**

Attributes:

1. **estimators\_ :**
2. **feature\_importances\_ :**

Methods:

1. **fit** (self, X, y[, sample\_weight]) **:** Build a forest of trees from the training set (X, y).
2. **predict** (self, X) **:** Predict class for X.
3. **predict\_proba** (self, X)**:** Predict class probabilities for X
4. **score** (self, X, y[, sample\_weight]) **:** Returns the mean accuracy on the given test data and labels