**Random Forest**

1. Random Forest is an ensemble machine learning technique capable of performing both **regression and classification** tasks using multiple decision trees and a statistical technique called **Bagging (Bootstrap Aggregation).**
2. A **RF** besides just averaging the prediction of trees it uses [two key concepts](https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf) that give it the name random:
3. Random sampling of training observations (rows) when building trees.
4. Random subsets of features (columns) for splitting nodes.
5. Random forest builds multiple decision trees and merge their predictions together to get a more **accurate** and **stable** prediction rather than relying on individual decision trees.
6. It combines the simplicity of the Decision Tree with flexibility resulting in a vast increase in accuracy.
7. It combines a number of **weak estimators (individual decision trees)** and merges them into a strong estimator.
8. Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model’s prediction.
9. A large number of relatively **uncorrelated models (trees)** operating as a committee will outperform any of the individual constituent models.
10. **It uses Bootstrapping (selection with replacement) to select random subsets of records and/or columns for creating individual decision trees. It means** that some samples will be used multiple times in a single tree.

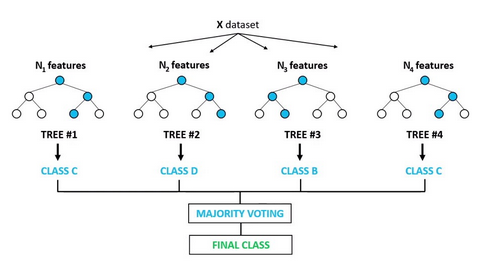
**Bootstrapping**

Bootstrapping is randomly selecting samples from training data with replacement. The samples so generated are called as bootstrap samples.

****

**Working of Random Forest**

1. Select random samples from a given dataset.
2. Construct a decision tree for each sample and get a prediction result from each decision tree.
3. Perform a vote for each predicted result.
4. Select the prediction result with the most votes as the final prediction.
5. The random forest combines hundreds or thousands of decision trees, trains each one on a slightly different set of the observations, splitting nodes in each tree considering a limited number of the features.
6. The final predictions of the random forest are made by averaging the predictions of each individual tree.
7. The random forest is a classification algorithm consisting of many decisions trees. It uses bagging and feature randomness when building each individual tree to try to create an uncorrelated forest of trees whose prediction by committee is more accurate than that of any individual tree.



**Parameters of Random Forest**

1. **n\_estimators :** This is the number of trees you want to build before taking the maximum voting or averages of predictions. Higher number of trees give you better performance but makes your code slower.
2. **min\_samples\_split :** The minimum number of samples required to split an internal node. The algorithm keeps splitting the nodes as long as a node has more samples (data points) than the number specified with min\_samples\_split parameter.
3. **criterion: gini or entropy** for classification, **mse or mae** for regression
4. **max\_depth:** determines max\_depth of a decision tree.
5. **max\_features:** These are the maximum number of features Random Forest is allowed to try in individual tree. The values it can have are:-
   * 1. ***Auto/None* :** This will simply take all the features which make sense in every tree. Here we simply do not put any restrictions on the individual tree.
     2. ***sqrt* :** This option will take square root of the total number of features in individual run. For instance, if the total number of features are 100, we can only take 10 of them in individual tree.
     3. ***0.x* :** This option allows the random forest to take 70% of variables in individual run. We can assign and value in a format “0.x” where we want x% of features to be considered.
6. **bootstrap – true/false.** If it is set to true samples are drawn with replacement and samples are drawn without replacement if bootstrap is set to false

**Pros**

1. Random Forest is considered as a highly accurate and robust method because of the number of decision trees participating in the process.
2. It **overcomes the problem of overfitting** faced by decision tree. The main reason is that it takes the average of all the predictions.

**Cons**

1. Random forest is slow in generating predictions because it has multiple decision trees. Whenever it makes a prediction, all the trees in the forest have to make a prediction for the same given input and then perform voting on it. This whole process is time-consuming.