**Random Forest**

Random Forest is another ensemble machine learning algorithm that follows the bagging technique. It is an extension of the bagging estimator algorithm. The base estimators in random forest are decision trees. Unlike bagging meta estimator, random forest randomly selects a set of features which are used to decide the best split at each node of the decision tree.

Looking at it step-by-step, this is what a random forest model does:

1. Random subsets are created from the original dataset (bootstrapping).
2. At each node in the decision tree, only a random set of features are considered to decide the best split.
3. A decision tree model is fitted on each of the subsets.
4. The final prediction is calculated by averaging the predictions from all decision trees.

**Note:**

The decision trees in random forest can be built on a subset of data and features. Particularly, the sklearn model of random forest uses all features for decision tree and a subset of features are randomly selected for splitting at each node.

To sum up, Random forest **randomly** selects data points and features, and builds **multiple trees (Forest) .**

**Parameters**

* **n\_estimators:**
  + It defines the number of decision trees to be created in a random forest.
  + Generally, a higher number makes the predictions stronger and more stable, but a very large number can result in higher training time.
* **criterion**:
  + It defines the function that is to be used for splitting.
  + The function measures the quality of a split for each feature and chooses the best split.
* **max\_features** :
  + It defines the maximum number of features allowed for the split in each decision tree.
  + Increasing max features usually improve performance but a very high number can decrease the diversity of each tree.
* **max\_depth**:
  + Random forest has multiple decision trees. This parameter defines the maximum depth of the trees.
* **min\_samples\_split:**
  + Used to define the minimum number of samples required in a leaf node before a split is attempted.
  + If the number of samples is less than the required number, the node is not split.
* **min\_samples\_leaf:**
  + This defines the minimum number of samples required to be at a leaf node.
  + Smaller leaf size makes the model more prone to capturing noise in train data.
* **max\_leaf\_nodes:**
  + This parameter specifies the maximum number of leaf nodes for each tree.
  + The tree stops splitting when the number of leaf nodes becomes equal to the max leaf node.
* **n\_jobs**:
  + This indicates the number of jobs to run in parallel.
  + Set value to -1 if you want it to run on all cores in the system.
* **random\_state**:
  + This parameter is used to define the random selection.
  + It is used for comparison between various models