**AdaBoost**

Adaptive boosting or AdaBoost is one of the simplest boosting algorithms. Usually, decision trees are used for modelling. Multiple sequential models are created, each correcting the errors from the last model. AdaBoost assigns weights to the observations which are incorrectly predicted and the subsequent model works to predict these values correctly.

Below are the steps for performing the AdaBoost algorithm:

1. Initially, all observations in the dataset are given equal weights.
2. A model is built on a subset of data.
3. Using this model, predictions are made on the whole dataset.
4. Errors are calculated by comparing the predictions and actual values.
5. While creating the next model, higher weights are given to the data points which were predicted incorrectly.
6. Weights can be determined using the error value. For instance, higher the error more is the weight assigned to the observation.
7. This process is repeated until the error function does not change, or the maximum limit of the number of estimators is reached.

**Parameters**

* **base\_estimators**:
  + It helps to specify the type of base estimator, that is, the machine learning algorithm to be used as base learner.
* **n\_estimators:** 
  + It defines the number of base estimators.
  + The default value is 10, but you should keep a higher value to get better performance.
* **learning\_rate:** 
  + This parameter controls the contribution of the estimators in the final combination.
  + There is a trade-off between learning\_rate and n\_estimators.
* **max\_depth**:
  + Defines the maximum depth of the individual estimator.
  + Tune this parameter for best performance.
* **n\_jobs**
  + Specifies the number of processors it is allowed to use.
  + Set value to -1 for maximum processors allowed.
* **random\_state** :
  + An integer value to specify the random data split.
  + A definite value of random\_state will always produce same results if given with same parameters and training data.