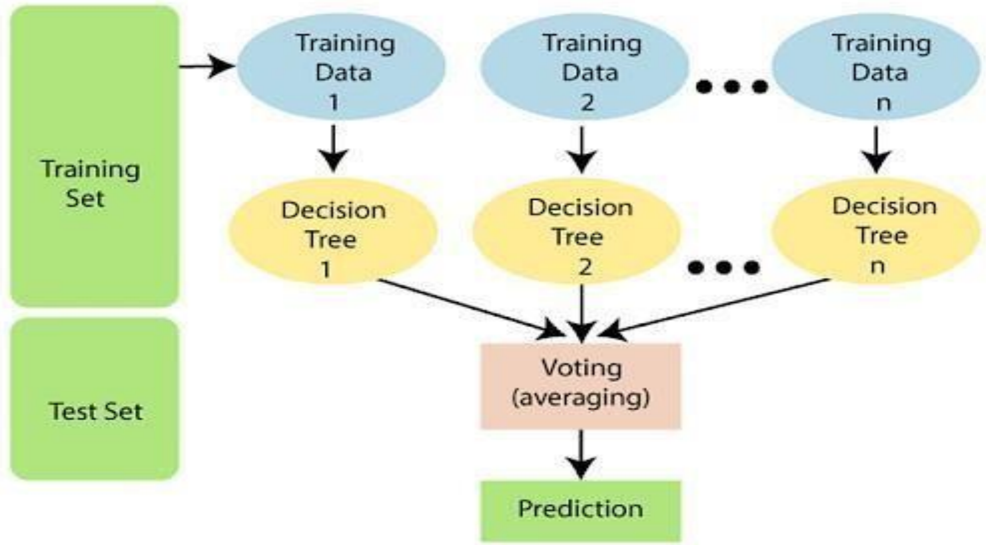
Random Forests

A Random Forest Algorithm is a supervised machine learning algorithm which is extremely popular and is used for Classification and Regression problems in Machine Learning. We know that a forest comprises numerous trees, and the more trees more it will be robust. Similarly, the greater the number of trees in a Random Forest Algorithm, the higher its accuracy and problem-solving ability. Random Forest is a classifier that contains several decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. It is based on the concept of ensemble learning which is a process of combining multiple classifiers to solve a complex problem and improve the performance of the model.



The following steps explain the working Random Forest Algorithm:

Step 1: Select random samples from a given data or training set.

Step 2: This algorithm will construct a decision tree for every training data.

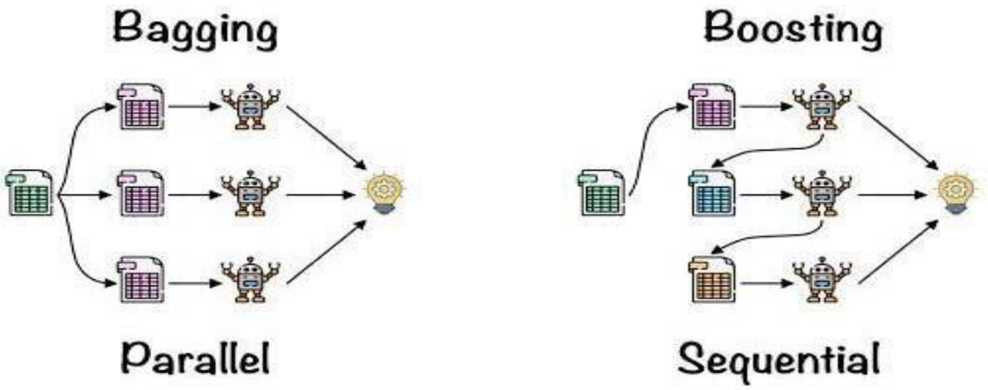
Step 3: Voting will take place by averaging the decision tree.

Step 4: Finally, select the most voted prediction result as the final prediction result.

This combination of multiple models is called Ensemble. Ensemble uses two methods:

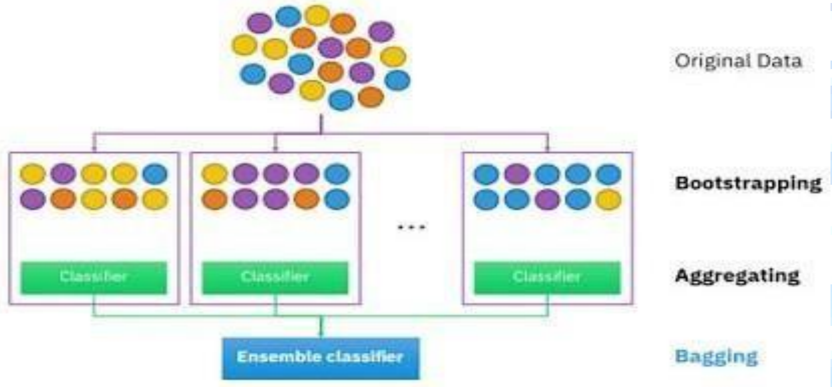
1. Bagging: Creating a different training subset from sample training data with replacement is called Bagging. The final output is based on majority voting.

2. Boosting: Combing weak learners into strong learners by creating sequential models



such that the final model has the highest accuracy is called Boosting. Example: ADA BOOST, XG BOOST.

Bagging: From the principle mentioned above, we can understand Random forest uses the Bagging code. Now, let us understand this concept in detail. Bagging is also known as Bootstrap Aggregation used by random forest. The process begins with any original random data. After arranging, it is organised into samples known as Bootstrap Sample. This process is known as Bootstrapping.Further, the models are trained individually, yielding different results known as Aggregation. In the last step, all the results are combined, and the generated output is based on majority voting. This step is known as Bagging and is done using an Ensemble Classifier.



Essential Features of Random Forest

- Miscellany: Each tree has a unique attribute, variety and features concerning other trees. Not all trees are the same.
- Immune to the curse of dimensionality: Since a tree is a conceptual idea, it requires no features to be considered. Hence, the feature space is reduced.
- Parallelization: We can fully use the CPU to build random forests since each

tree is created autonomously from different data and features.

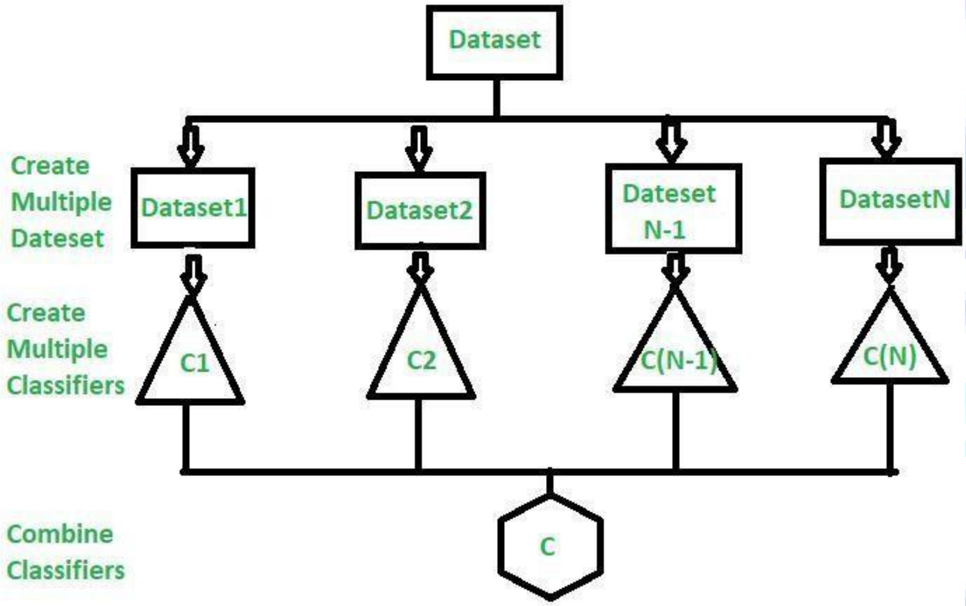
- Train-Test split: In a Random Forest, we don’t have to differentiate the data for train and test because the decision tree never sees 30% of the data.

- Stability: The final result is based on Bagging, meaning the result is based on majority voting or average.

Ensemble learning

Ensemble learning helps improve machine learning results by combining several models. This approach allows the production of better predictive performance compared to a single model. Basic idea is to learn a set of classifiers (experts) and to allow them to vote.

Advantage : Improvement in predictive accuracy.
Disadvantage : It is difficult to understand an ensemble of classifiers



Types of Ensemble Classifier –

Bagging:

Bagging (Bootstrap Aggregation) is used to reduce the variance of a decision tree. Suppose a set D of d tuples, at each iteration *i*, a training set D_i of d tuples is sampled with replacement from D (i.e., bootstrap). Then a classifier model M_i is learned for each training set D < i. Each classifier M_i returns its class prediction. The bagged classifier M* counts the votes and assigns the class with the most votes to X (unknown sample).

Boosting

Boosting is an ensemble learning method that combines a set of weak learners into a strong learner to minimize training errors. In boosting, a random sample of data is selected, fitted with a model and then trained sequentially—that is, each model tries to compensate for the weaknesses of its predecessor. With each iteration, the weak rules from each individual classifier are combined to form one, strong prediction rule

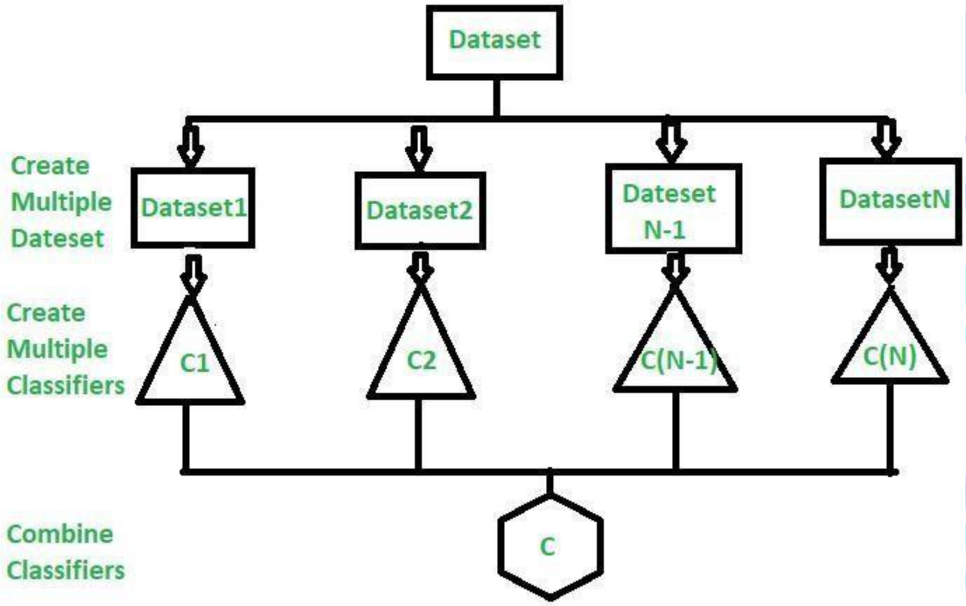
Types of boosting

Boosting methods are focused on iteratively combining weak learners to build a strong learner that can predict more accurate outcomes. As a reminder, a weak learner classifies data slightly better than random guessing. This approach can provide robust results for prediction problems, and can even outperform neural networks and support vector machines for tasks like image retrieval.

Boosting algorithms can differ in how they create and aggregate weak learners during the sequential process. Three popular types of boosting methods include:

- **Adaptive boosting or AdaBoost:** Yoav Freund and Robert Schapire are credited with the creation of the AdaBoost algorithm. This method operates iteratively, identifying misclassified data points and adjusting their weights to minimize the training error. The model continues optimize in a sequential fashion until it yields the strongest predictor.
- **Gradient boosting:** Building on the work of Leo Breiman, Jerome H. Friedman developed gradient boosting, which works by sequentially adding predictors to an ensemble with each one correcting for the errors of its predecessor. However, instead of changing weights of data points like AdaBoost, the gradient boosting trains on the residual errors of the previous predictor. The name, gradient boosting, is used since it combines the gradient descent algorithm and boosting method.
- **Extreme gradient boosting or XGBoost:** XGBoost is an implementation of gradient boosting that’s designed for computational speed and scale. XGBoost leverages multiple cores on the CPU, allowing for learning to occur in parallel during training.

.


Dataset
Create
Multiple
Dateset
Dataset1
Dataset2
Dateset
N-1
DatasetN
Create
Multiple
Classifiers
C1
C2
C(N-1)
C(N)
Combine
Classifiers
C