**What is Random Forest Algorithm?**

Random Forest is one of the most popular and commonly used algorithms by Data Scientists. Random forest is a ***Supervised***[***Machine Learning Algorithm***](https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/) that is ***used widely in Classification and Regression problems***. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

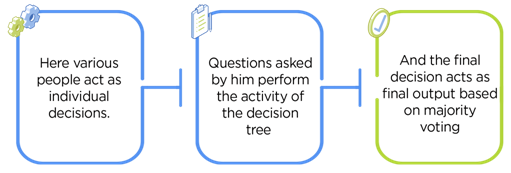
Random forest is a versatile machine learning algorithm developed by Leo Breiman and Adele Cutler. It leverages an ensemble of multiple decision trees to generate predictions or classifications. By combining the outputs of these trees, the random forest algorithm delivers a consolidated and more accurate result.

Its widespread popularity stems from its user-friendly nature and adaptability, enabling it to tackle both classification and regression problems effectively. The algorithm’s strength lies in its ability to handle complex datasets and mitigate overfitting, making it a valuable tool for various predictive tasks in machine learning.

One of the most important features of the Random Forest Algorithm is that it can handle the data set containing ***continuous variables,*** as in the case of regression, and ***categorical variables,*** as in the case of classification. It performs better for classification and regression tasks. In this tutorial, we will understand the working of random forest and implement random forest on a classification task.

**Real-Life Analogy of Random Forest**

Let’s dive into a real-life analogy to understand this concept further. A student named X wants to choose a course after his 10+2, and he is confused about the choice of course based on his skill set. So he decides to consult various people like his cousins, teachers, parents, degree students, and working people. He asks them varied questions like why he should choose, job opportunities with that course, course fee, etc. Finally, after consulting various people about the course he decides to take the course suggested by most people.



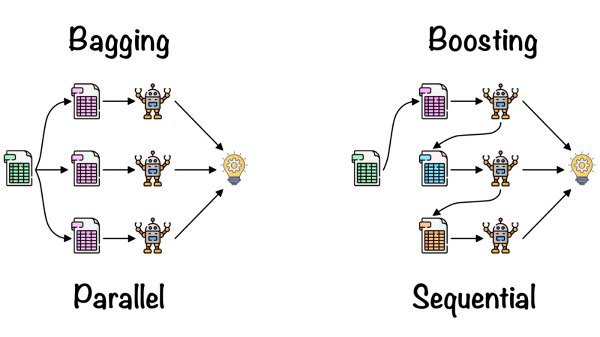
**Working of Random Forest Algorithm**

Before understanding the working of the random forest algorithm in machine learning, we must look into the ensemble learning technique. ***Ensemble*** simplymeans combining multiple models. Thus a collection of models is used to make predictions rather than an individual model.

Ensemble uses two types of methods:

1. **Bagging**– It creates a different training subset from sample training data with replacement & the final output is based on majority voting. For example,  Random Forest.

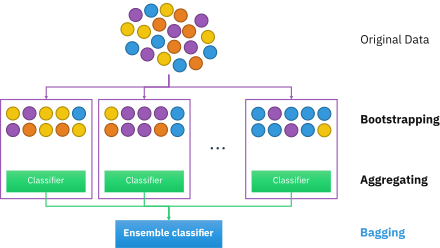
2. **Boosting**– It combines weak learners into strong learners by creating sequential models such that the final model has the highest accuracy. For example,  ADA BOOST, XG BOOST.



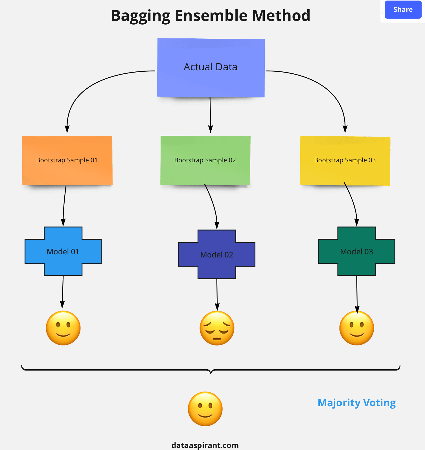
As mentioned earlier, Random forest works on the Bagging principle. Now let’s dive in and understand bagging in detail.

**Bagging**

Bagging, also known as ***Bootstrap Aggregation,*** is the ensemble technique used by random forest.Bagging chooses a random sample/random subset from the entire data set. Hence each model is generated from the samples (Bootstrap Samples) provided by the Original Data with replacement known as ***row sampling***. This step of row sampling with replacement is called ***bootstrap***. Now each model is trained independently, which generates results. The final output is based on majority voting after combining the results of all models. This step which involves combining all the results and generating output based on majority voting, is known as ***aggregation***.



Now let’s look at an example by breaking it down with the help of the following figure. Here the bootstrap sample is taken from actual data (Bootstrap sample 01, Bootstrap sample 02, and Bootstrap sample 03) with a replacement which means there is a high possibility that each sample won’t contain unique data. The model (Model 01, Model 02, and Model 03) obtained from this bootstrap sample is trained independently. Each model generates results as shown. Now the Happy emoji has a majority when compared to the Sad emoji. Thus based on majority voting final output is obtained as Happy emoji.



**Boosting**

Boosting is one of the techniques that use the concept of ensemble learning. A boosting algorithm combines multiple simple models (also known as weak learners or base estimators) to generate the final output. It is done by building a model by using weak models in series.

There are several boosting algorithms; AdaBoost was the first really successful boosting algorithm that was developed for the purpose of binary classification. AdaBoost is an abbreviation for Adaptive Boosting and is a prevalent boosting technique that combines multiple “weak classifiers” into a single “strong classifier.” There are Other Boosting techniques. For more, you can visit

**Steps Involved in Random Forest Algorithm**

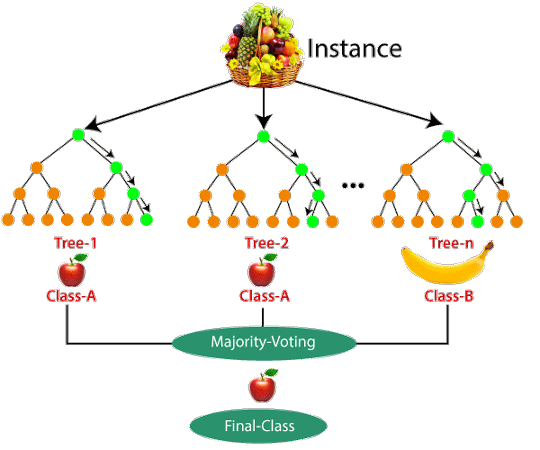
Step 1: In the Random forest model, a subset of data points and a subset of features is selected for constructing each decision tree. Simply put, n random records and m features are taken from the data set having k number of records.

Step 2: Individual decision trees are constructed for each sample.

Step 3: Each decision tree will generate an output.

Step 4: Final output is considered based on ***Majority Voting or Averaging*** for Classification and regression, respectively.

For example:  consider the fruit basket as the data as shown in the figure below. Now n number of samples are taken from the fruit basket, and an individual decision tree is constructed for each sample. Each decision tree will generate an output, as shown in the figure. The final output is considered based on majority voting. In the below figure, you can see that the majority decision tree gives output as an apple when compared to a banana, so the final output is taken as an apple.



**Important Features of Random Forest**

* **Diversity:** Not all attributes/variables/features are considered while making an individual tree; each tree is different.
* **Immune to the curse of dimensionality:** Since each tree does not consider all the features, the feature space is reduced.
* **Parallelization:** Each tree is created independently out of different data and attributes. This means we can fully use the CPU to build random forests.
* **Train-Test split:** In a random forest, we don’t have to segregate the data for train and test as there will always be 30% of the data which is not seen by the decision tree.
* **Stability:** Stability arises because the result is based on majority voting/ averaging.

**Difference Between Decision Tree and Random Forest**

Random forest is a collection of decision trees; still, there are a lot of differences in their behavior.

|  |  |
| --- | --- |
| **Decision trees** | **Random Forest** |
| 1. Decision trees normally suffer from the problem of overfitting if it’s allowed to grow without any control. | 1. Random forests are created from subsets of data, and the final output is based on average or majority ranking; hence the problem of overfitting is taken care of. |
| 2. A single decision tree is faster in computation. | 2. It is comparatively slower. |
| 3. When a data set with features is taken as input by a decision tree, it will formulate some rules to make predictions. | 3. Random forest randomly selects observations, builds a decision tree, and takes the average result. It doesn’t use any set of formulas. |

Thus random forests are much more successful than decision trees only if the trees are diverse and acceptable.