Random Forest

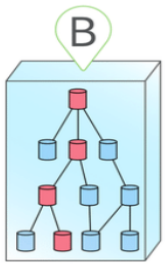
Random Forest

- Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML.
- 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.

Decision Tree vs Random Forest

A

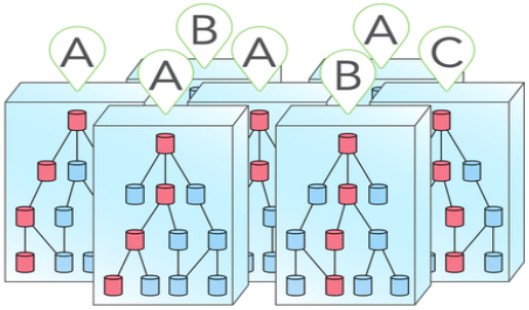
Decision tree



Single output

B

Random forest



A	B	C
5	2	1

Majority voting: A

DECISION TREE VERSUS RANDOM FOREST

DECISION TREE	RANDOM FOREST
A decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility	An ensemble learning method that operates by constructing a multitude of decision trees at training time and outputting the class depending on the individual trees
There is a possibility of overfitting	Reduced risk of overfitting
Gives less accurate results	Gives more accurate results
Simpler and easier to understand, interpret and visualize	Comparatively more complex

Difference in Decision Tree and Individual RF Tree

Are decision trees in Random Forest different from regular decision trees?

- It's easy to get confused by a single decision tree and a decision forest.
- Random Forest looks like a decision forest would be a bunch of single decision trees but It's a bunch of single decision trees but all of the **trees are mixed together randomly instead of separate trees growing individually.**
- When using a regular decision tree, you would input a training dataset with features and labels and it will formulate some set of rules which it will use to make predictions.
- If you entered that same information into a Random Forest algorithm, it will randomly select observations and features to build several decision trees and then average the results.

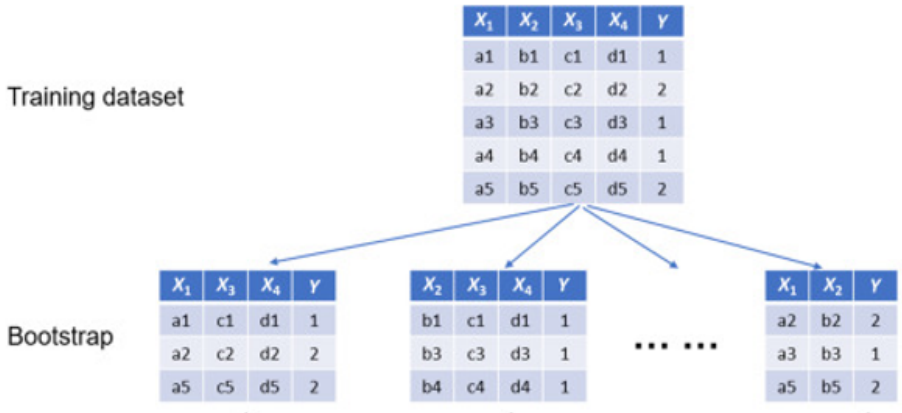
Working of Random Forest

Suppose we have a dataset which consist of **n observations** and **m features**.

Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase.

The Working process can be explained in the below steps

Step-1: Select any **k (k<=n) rows** (observations) and **p (p<=m) features** from the training set.



X_1	X_2	X_3	X_4	Y
a1	b1	c1	d1	1
a2	b2	c2	d2	2
a3	b3	c3	d3	1
a4	b4	c4	d4	1
a5	b5	c5	d5	2

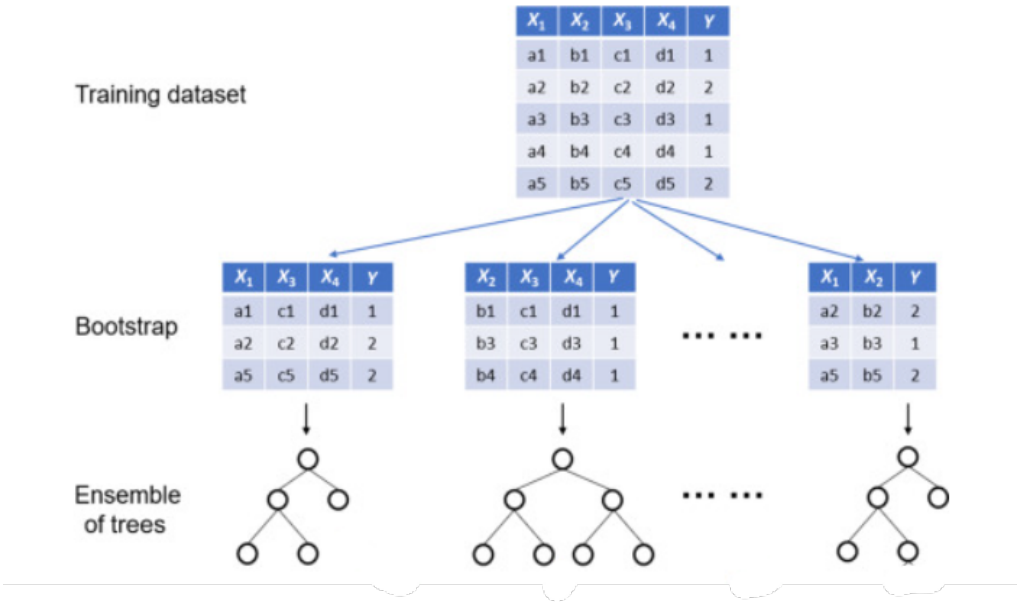
X_1	X_3	X_4	Y
a1	c1	d1	1
a2	c2	d2	2
a5	c5	d5	2

X_2	X_3	X_4	Y
b1	c1	d1	1
b3	c3	d3	1
b4	c4	d4	1

X_1	X_2	Y
a2	b2	2
a3	b3	1
a5	b5	2

Working of Random Forest

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



X_1	X_2	X_3	X_4	Y
a1	b1	c1	d1	1
a2	b2	c2	d2	2
a3	b3	c3	d3	1
a4	b4	c4	d4	1
a5	b5	c5	d5	2

X_1	X_3	X_4	Y
a1	c1	d1	1
a2	c2	d2	2
a5	c5	d5	2

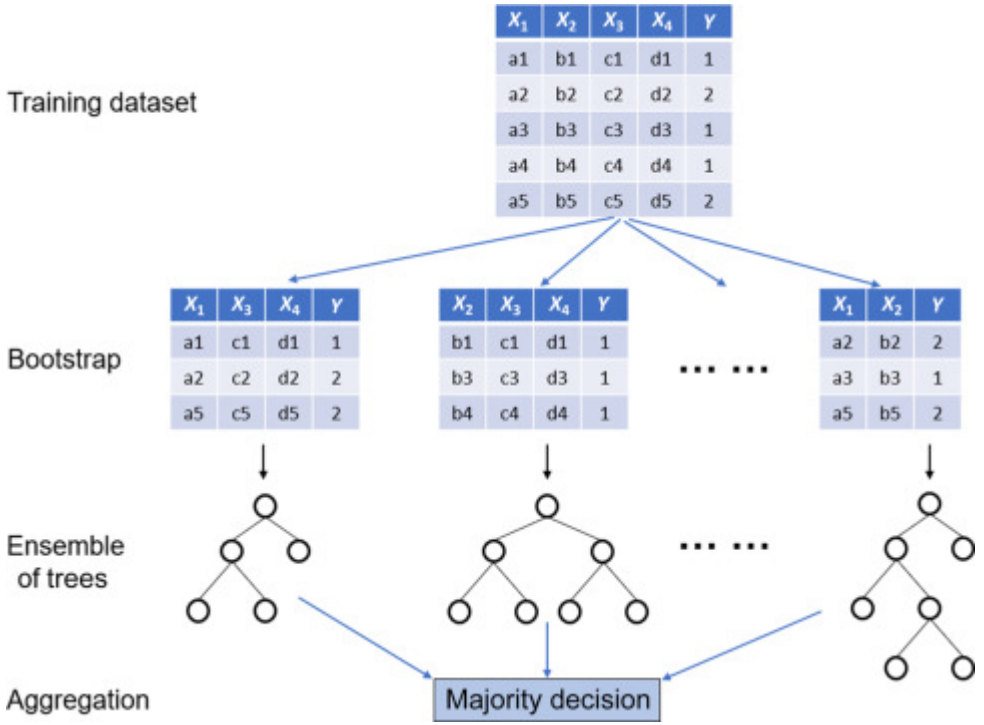
X_2	X_3	X_4	Y
b1	c1	d1	1
b3	c3	d3	1
b4	c4	d4	1

X_1	X_2	Y
a2	b2	2
a3	b3	1
a5	b5	2

Step 3: Each decision tree will generate an output.

Working of Random Forest

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



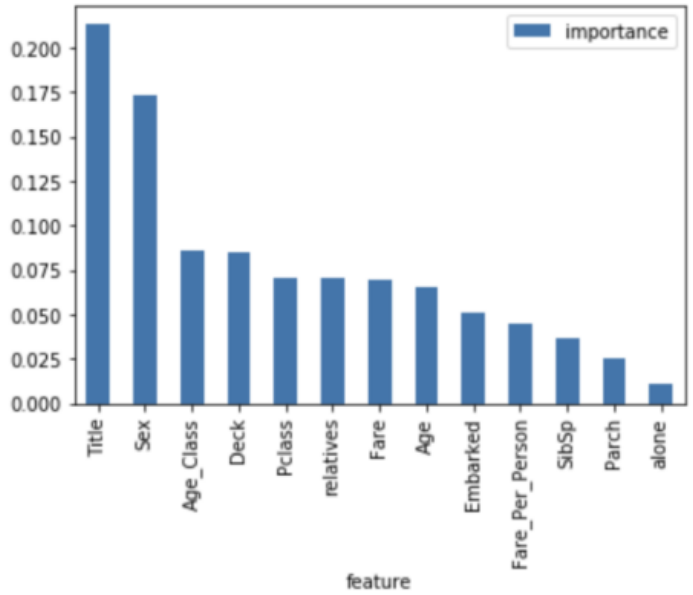
Note : Random Forest Regressor works similar to Random Forest Classifier except for the fact that in case of regression the predictions generated from each tree are averaged rather than max voting.

Random Forest Feature Importance

- Another great quality of the random forest algorithm is that it is very easy to measure the relative importance of each feature on the prediction.
- Sklearn provides a great tool for this that measures a feature's importance by looking at how much the tree nodes that use that feature reduce impurity across all trees in the forest.
- It computes this score automatically for each feature after training and scales the results so the sum of all importance is equal to one.
- By looking at the feature importance you can decide which features to possibly drop because they don't contribute enough (or sometimes nothing at all) to the prediction process.
- This is important because a general rule in machine learning is that the more features you have the more likely your model will suffer from overfitting and vice versa.

Random Forest Feature Importance

feature	importance
Title	0.213
Sex	0.173
Age_Class	0.086
Deck	0.085
Pclass	0.071
relatives	0.070
Fare	0.069
Age	0.065
Embarked	0.051
Fare_Per_Person	0.045
SibSp	0.037
Parch	0.025
alone	0.011



Random Forest Important Parameters

Hyperparameters are used in random forests to either enhance the performance and predictive power of models or to make the model faster.

Hyperparameters that increases predictive power:

1. n_estimators– number of trees the algorithm builds before averaging the predictions.

2. max_features– maximum number of features random forest considers splitting a node.

3. min_sample_leaf– determines the minimum number of leaves required to split an internal node.

Random Forest Important Parameters

Hyperparameters that increases speed of modelling:

1. ***n_jobs***– it tells the engine how many processors it is allowed to use. If the value is 1, it can use only one processor but if the value is -1 there is no limit.

2. ***random_state***– controls randomness of the sample. The model will always produce the same results if it has a definite value of random state and if it has been given the same hyperparameters and the same training data.

3. ***oob_score*** – OOB means out of the bag. It is a random forest cross-validation method. In this one-third of the sample is not used to train the data instead used to evaluate its performance. These samples are called out of bag samples.

Sklearn Parameters and default values

n_estimators: default = 100

The number of trees in the forest.

criterion : {“gini”, “entropy”, “log_loss”}, default=”gini”

The function to measure the quality of a split.

max_depth : default = None

The maximum depth of the tree. If None, then nodes are expanded until all leaves are pure or until all leaves contain less than **min_samples_split** samples.

min_samples_split : default = 2

The minimum number of samples required to split an internal node:

min_samples_leaf : default = 1

The minimum number of samples required to be at a leaf node.

Sklearn Parameters and default values

max_features: {“sqrt”, “log2”, None} default=”sqrt”

The number of features to consider when looking for the best split

oob_score: default=False

Whether to use out-of-bag samples to estimate the generalization score.

n_jobs: default=None

The number of jobs to run in parallel. None means 1. -1 means using all processors.

random_state: default=None

Controls both the randomness of the bootstrapping of the samples used when building trees (if bootstrap=True) and the sampling of the features to consider when looking for the best split at each node (if max_features < n_features).

Sklearn Parameters and default values

ccp_alpha: default=0.0

Complexity parameter used for Minimal Cost-Complexity Pruning.

max_samples: default=None

If bootstrap is True, the number of samples to draw from X to train each base estimator.

Important Characteristics 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 that we can make full use of the CPU to build random forests.
- **Stability :** Stability arises because the result is based on majority voting/ averaging.

Disadvantages of Random Forest

High computation power

- Because random forest uses many decision trees, it can require a lot of memory on larger projects. This can make it slower than some other, more efficient, algorithms.

Prone to overfit

- Sometimes, because this is a decision tree-based method and decision trees often suffer from overfitting, this problem can affect the overall forest.
- This problem is usually prevented by Random Forest by default because it uses random subsets of the features and builds smaller trees with those subsets. This can slow down processing speed but increase accuracy.

Ensemble Methods : Boosting