

## Food for Thought

What is the difference between a decision tree algorithm and a random forest algorithm?

What sorts of hyperparameters are available for random forest algorithms?

Is bagging or pasting generally a more efficient method for training random forests?

## What Is a Random Forest?

A random forest is an easy to use machine learning algorithm that does a very good job of randomly producing information. Its popularity can be traced back to its simplicity and diverse results. Another huge advantage of the Random Forest function is that it can be used for both regression and classification problems.

To execute this algorithm, a variety of empty decision trees are built. It fills in the trees based on random subsets of the data, and links them together.

Random Forests are very similar to, yet not equal to, decision trees. The simplest difference is that of 'tree' vs 'forest' where a tree has one root and many stems. A random forest is a collection of trees. Another key difference is how they make decisions. A tree analyzes one piece of data, a forest will have many trees analyze data and put in a 'vote', and based upon that vote, the 'winning' piece of data is selected.

## Random Forest Classifiers

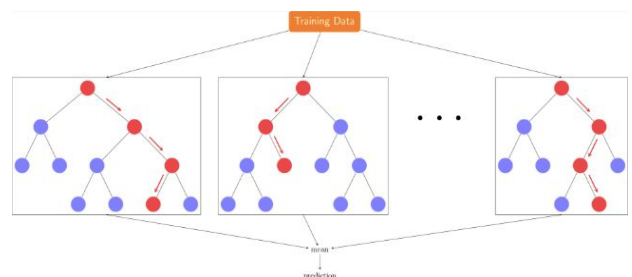
- It is an *ensemble algorithm* which means it combines more than one algorithms of same/different kind.
- Creates a set of decision trees from randomly selected subset of a training set then it aggregates the votes from different decision trees to decide the final class of the test object.

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
iris = load_iris()
X_train, X_test = train_test_split(iris["data"], test_size=0.2, random_state=42)
y_train, y_test = train_test_split(iris["target"], test_size=0.2, random_state=42)
rnd_clf = RandomForestClassifier(n_estimators=500, max_leaf_nodes=16, n_jobs=-1)
rnd_clf.fit(iris["data"], iris["target"])
y_pred_clf = rnd_clf.predict(X_test)
```

## Random Forest Regressors

- The algorithm fits a number of classifying decision trees on various sub-samples of the dataset and uses the average to improve prediction

```
# Don't use the iris dataset for this algorithm
from sklearn.ensemble import RandomForestRegressor
forest_reg = RandomForestRegressor()
forest_reg.fit(X_train, y_train)
y_pred_reg = forest_reg.predict(X_test)
```



## Resources

<https://builtin.com/data-science/random-forest-algorithm>

<https://www.datacamp.com/community/tutorials/random-forests-classifier-python>

<https://www.geeksforgeeks.org/random-forest-regression-in-python/>

<https://medium.com/machine-learning-101/chapter-5-random-forest-classifier-56dc7425c3e1>