Random Forest

What is Random Forest?

Random Forest is a versatile ensemble machine learning method capable of performing both regression and classification tasks. It also undertakes dimensional reduction methods, treats missing values, outlier values and other essential steps of data exploration, and does a fairly good job. It is a type of ensemble learning method, where a group of weak models combine to form a powerful model.

How it works:

In Random Forest, we grow multiple trees as opposed to a single tree in DT model to classify/predict a new object based on attributes, each tree gives a classification and we say the tree "votes" for that class. The forest chooses the classification having the most votes (over all the trees in the forest) and in case of regression, it takes the average of outputs by different trees.

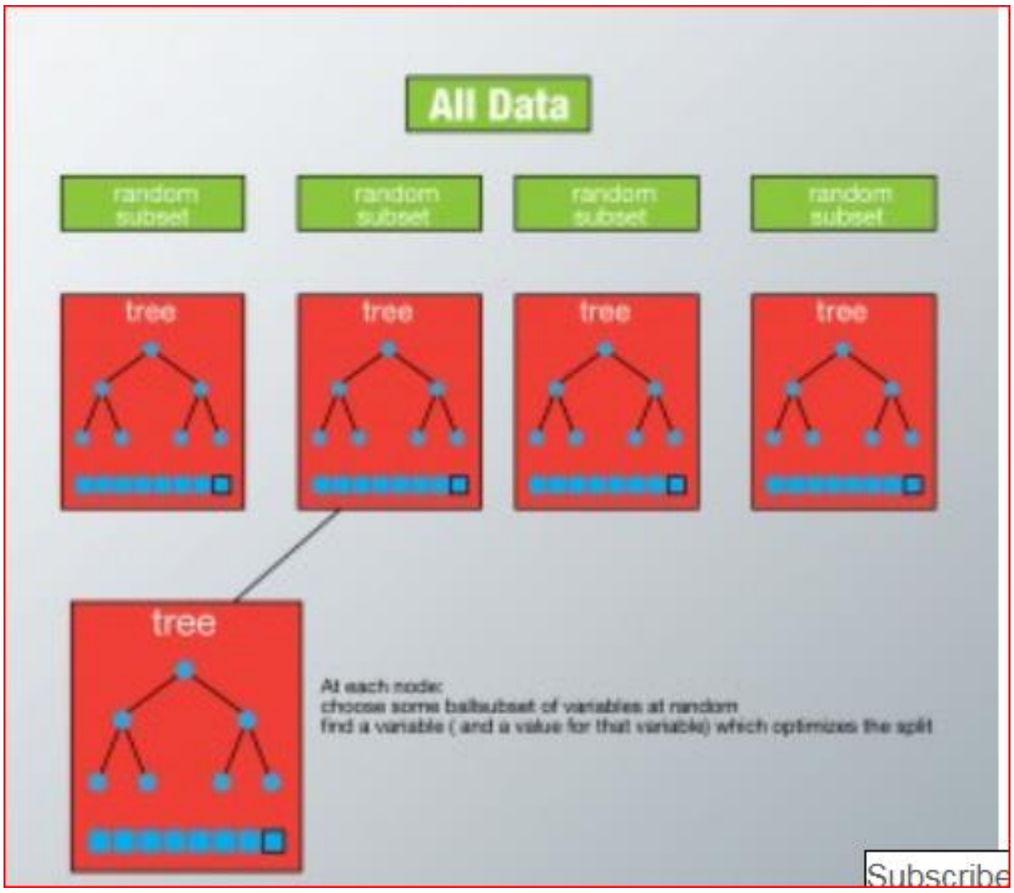
It works in the following manner. Each tree is planted & grown as follows:

- Assume number of cases in the training set is N. Then, sample of these N cases is taken at random but with replacement. This sample will be the training set for growing the tree.
- If there are M input variables, a number m<M is specified such that at each node, m variables are selected at random out of the M. The best split on these m is used to split the node. The value of m is held constant while we grow the forest.

- Each tree is grown to the largest extent possible and there is no pruning.

- Predict new data by aggregating the predictions of the tree trees (i.e., majority votes for classification, average for regression).

The same above steps illustrated below in a picture


All Data
random subset
random subset
random subset
random subset
tree
tree
tree
tree
tree
At each node:
choose some balsubset of variables at random
find a variable (and a value for that variable) which optimizes the split
Subscribe

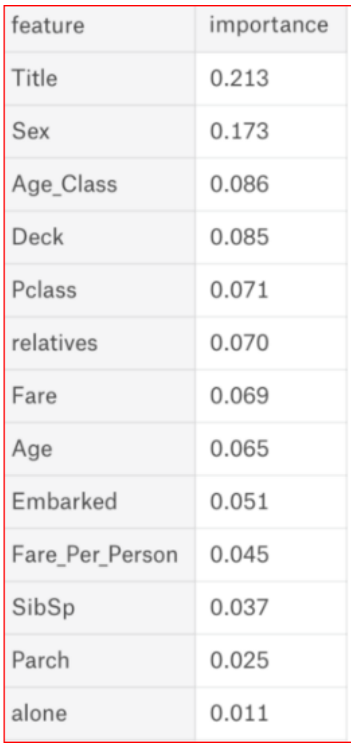
Random Forest adds additional randomness to the model, while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model.

Feature Importance

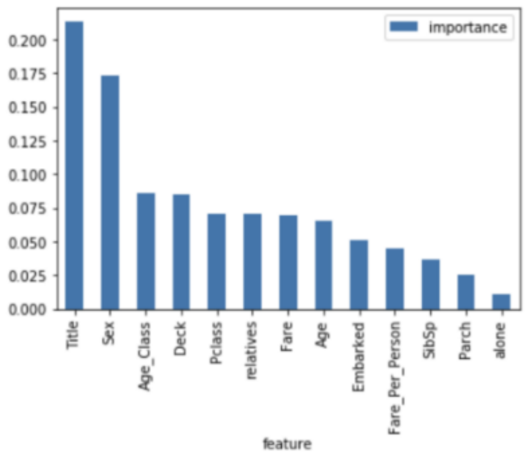
Another great quality of the random forest algorithm is that it is very easy to measure the relative importance of each feature on the prediction. It measures a features importance by looking at how much the tree nodes, which use that feature, reduce impurity across all trees in the forest. It computes this score automatically for each feature after training and scales the results, so that the sum of all importance is equal to 1.

Through looking at the feature importance, you can decide which features you may want to drop, because they don’t contribute enough or nothing 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 over fitting and vice versa.

Below is the one of the example of how you will get variable importance in random forest



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



Important Hyper parameters:

Parameter Tuning: Mainly, there are three parameters in the random forest algorithm which you should look at (for tuning):

- Ntree - As the name suggests, the number of trees to grow. Larger the tree, it will be more computationally expensive to build models.
- mtry - It refers to how many variables we should select at a node split. Also as mentioned above, the default value is p/3 for regression and sqrt(p) for classification. We should always try to avoid using smaller values of mtry to avoid overfitting.
- nodesize - It refers to how many observations we want in the terminal nodes. This parameter is directly related to tree depth. Higher the number, lower the tree depth. With lower tree depth, the tree might even fail to recognize useful signals from the data.

Advantages and Disadvantages of Random Forest

Advantages

It is robust to correlated predictors.

1. It is used to solve both regression and classification problems.
2. It can handle thousands of input variables without variable selection.
3. It can be used as a feature selection tool using its variable importance plot.
4. It takes care of missing data internally in an effective manner.

Disadvantages

The Random Forest model is difficult to interpret.

1. It tends to return erratic predictions for observations out of range of training data. For example, the training data contains two variable x and y. The range of x variable is 30 to 70. If the test data has x = 200, random forest would give an unreliable prediction.

2. It can take longer than expected time to computer a large number of trees.