**RANDOM FOREST**

Random forest is a tree-based algorithm which involves building several trees (decision trees), then combining their output to improve generalization ability of the model. It creates decision trees on randomly selected data samples, gets prediction from each tree and selects the best solution by means of voting.

Random Forest is an extension over bagging. Random forests use another trick to make the multiple fitted trees a bit less correlated with each other when growing each tree and to bring additional randomness to the model while growing the trees. Instead of only sampling over the observations in the dataset to generate a bootstrap sample, we also sample over features and keep only a random subset of them to build the tree. Sampling over features has indeed the effect that all trees do not look at the exact same information to make their decisions and so it reduces the correlation between the different returned outputs.

The more the number of trees the better the model. Prediction error will not increase with higher number of trees but it just won't decrease any further at some point.

Random Forest is also a pretty good indicator of the feature importance.

Unlike a tree no pruning takes place in random forest i.e. each tree is grown fully. Pruning is a method to avoid overfitting in decision tree. Random Forest overcomes the problem of overfitting by random feature selection.

**Problems with Decision Trees**

- Overfitting is a practical problem while building a decision tree model. It generally happens when it builds many branches in the model most likely to catch patterns that are only applicable to our dataset and not real word. Since the algorithm continues splitting on attributes until either it classifies all the data points or there are no more attributes to splits on. As a result, it is prone to creating decision trees that over fit by performing really well on the training data at the expense of accuracy with respect to the entire distribution of data. In other words, tree suffers from high variance. Also Averaging several highly correlated trees doesn't lead to a large reduction in variance. But how do correlated trees emerge? Let's say a data set has a very strong predictor, along with other moderately strong predictors. In bagging, a tree grown every time would consider the very strong predictor at its root node, thereby resulting in trees similar to each other.

One way Random Forests reduce variance is by training models on a random subset of features. Thus in each tree is getting built on random features. This inclusion of many features will help limit our error due to variance and bias as well. It will also make the multiple fitted trees a bit less correlated with each other when growing each tree and to bring additional randomness to the model while growing the trees. Considering only a subset of predictors at a split result in trees with different predictors at top split results in decorrelated trees and more reliable average output.

**How does Random Forest work?**

- First it uses the Bagging (Bootstrap Aggregating) algorithm to create random samples for each tree. Given a data set D1 (m rows), it creates a new dataset D2 (n rows) by row sampling at random with replacement from the original data where n is a small percent of m (ex - 2%) for each decision tree model. If oob score is true, number of samples for each decision tree is increased by ⅓ of samples for that tree. So n will be increased by ⅓ size and trained model will be tested on these ⅓ samples. Ex - m = 1000, n = 20(2% of m), oob score samples = 10 (⅓ of n), final number of samples of n = 30.

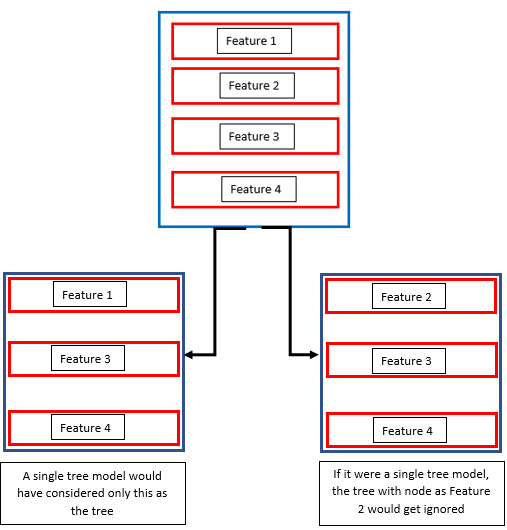
- Second, for each tree all columns are considered but while model building when each time a split in a tree is considered, a random sample of ‘k’ predictors is chosen as split candidate from the full set of ‘p’ predictors. A split can use only one of the ‘k’ predictors where

k=p/3 in case of regression

k = sqrt(p) in case of classification

Although all columns are considered for all decision tree model yet the samples for each decision tree is different and hence the column finalized to split the tree at different levels will be different in different decision tree models. This brings in the column randomness.

A normal decision tree model would only consider feature 1 and make a node and go on to build a tree. This would make feature 2, which also very significant unable to contribute to the model. In a Random Forest, where there are several trees, a tree would also be made considering feature 2 as the node. This way, the model picks up nodes in a random manner and makes a forest. These trees are then trained differently on same dataset and they come up with different predictions.



- Several trees (each tree is grown on a different sample of original data) are grown, tested on oob samples and the final prediction is obtained by voting (in case of classification) or averaging (in case of regression). Final oob score is the average of oob scores of all models.