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

When we build a decision tree, it considers all columns in dataset and chooses the best column branch by branch to make the split. So the column that made split at root node is considered more important than columns at the other branches and hence all predictions will have huge impact of column at the top and when in real time scenario if there comes an observation for which the first column is not the most important feature to make prediction, the model will fail to predict well.

This problem leads to high variance i.e. accuracy decreases when we change the dataset to predict. The solution is to use random forest model which reduces variance by training models using a random subset of features on a random subset of samples. So the output is not from one decision tree where columns with their fixed importance levels are used for all samples' predictions but rather from the output of many decision trees where different features were considered different importances.

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, the 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 models 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. This brings row randomness

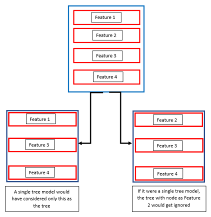
- Second, for each tree all columns are considered but while model building when each time a split in a tree is considered, a random sample of ‘k’ predictors is chosen as a split candidate from the full set of ‘p’ predictors. A split can use only one of the ‘k’ predictors where

k=p/3 in case of regression

k = sqrt(p) in case of classification

Although all columns are considered for all decision tree models, the columns 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 is also very significant, unable to contribute to the model. In a Random Forest, where there are several trees, one of the trees would consider 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 the same dataset and they come up with different predictions.


Feature 1
Feature 2
Feature 3
Feature 4
Feature 1
Feature 3
Feature 4
Feature 2
Feature 3
Feature 4
A single tree model would have considered only this as the tree
If it were a single tree model, the tree with node as Feature 2 would get ignored

- 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.