Random Forest

Random Forest is a supervised learning algorithm. As the name suggests, it creates a forest and makes it somehow random. The “forest” it builds, is an ensemble of Decision Trees, most of the time trained with the “bagging” method. The general idea of the bagging method is that a combination of learning models increases the overall result. In Layman terms, Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction.

One big advantage of random forest is, that it can be used for both classification and regression problems.

Random Forest has nearly the same hyperparameters as a decision tree or a bagging classifier. Fortunately, you don’t have to combine a decision tree with a bagging classifier and can just easily use the classifier-class of Random Forest. Like I already said, with Random Forest, you can also deal with Regression tasks by using the Random Forest regressor.

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.

Uses:

The random forest algorithm is used in a lot of different fields, like Banking, Stock Market, Medicine and E-Commerce. In Banking it is used for example to detect customers who will use the bank’s services more frequently than others and repay their debt in time. In this domain it is also used to detect fraud customers who want to scam the bank. In finance, it is used to determine a stock’s behaviour in the future. In the healthcare domain it is used to identify the correct combination of components in medicine and to analyze a patient’s medical history to identify diseases. And lastly, in E-commerce random forest is used to determine whether a customer will actually like the product or not.

Difference between Decision Trees and Random Forests:

Like I already mentioned, Random Forest is a collection of Decision Trees, but there are some differences.If you input a training dataset with features and labels into a decision tree, it will formulate some set of rules, which will be used to make the predictions.

For example, if you want to predict whether a person will click on an online advertisement, you could collect the ad's the person clicked in the past and some features that describe his decision. If you put the features and labels into a decision tree, it will generate some rules. Then you can predict whether the advertisement will be clicked or not. In comparison, the Random Forest algorithm randomly selects observations and features to build several decision trees and then averages the results.

Another difference is that "deep" decision trees might suffer from overfitting. Random Forest prevents overfitting most of the time, by creating random subsets of the features and building smaller trees using these subsets. Afterwards, it combines the subtrees. Note that this doesn't work every time and that it also makes the computation slower, depending on how many trees your random forest builds.

How Random Forests work:

Random Forest pseudocode:

1. Randomly select "k" features from total "m" features, where k << m.
2. Among the "k" features, calculate the node "d" using the best split point.
3. Split the node into daughter nodes using the best split.
4. Repeat 1 to 3 steps until "l" number of nodes has been reached.
5. Build forest by repeating steps 1 to 4 for "n" number times to create "n" number of trees.

The beginning of random forest algorithm starts with randomly selecting "k" features out of total "m" features. In the image, you can observe that we are randomly taking features and observations.

In the next stage, we are using the randomly selected "k" features to find the root node by using the best split approach.

The next stage, We will be calculating the daughter nodes using the same best split approach. Will the first 3 stages until we form the tree with a root node and having the target as the leaf node.

Finally, we repeat 1 to 4 stages to create “n” randomly created trees. This randomly created trees forms the random forest.

Random forest prediction pseudocode:

To perform prediction using the trained random forest algorithm uses the below pseudocode.

1. Takes the test features and use the rules of each randomly created decision tree to predict the oucome and stores the predicted outcome (target)
2. Calculate the votes for each predicted target.
3. Consider the high voted predicted target as the final prediction from the random forest algorithm.

Advantages of random forest algorithm:

- The overfitting problem will never come when we use the random forest algorithm in any classification problem.
- The same random forest algorithm can be used for both classification and regression task.
- The random forest algorithm can be used for feature engineering. Which means identifying the most important features out of the available features from the training dataset.