Understanding Random Forest

TO BETTER YOUR MODELS

Decision Trees

In layman terms Random forest is just a bunch of decision trees (which is a simple algorithm) developed to predict something using the given data. Random forest(RF) averages all the predictions across those decision trees to give out the final prediction.

Let us see how a decision tree works

A decision tree splits the data into groups based on some condition. If we have a tabled data, each column could be the condition to split our data. For example, a class of 70 students can be split into groups based on grades, height, sex etc.

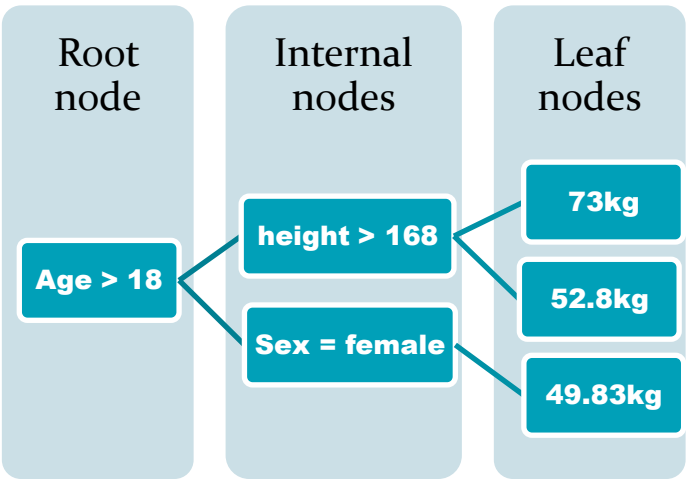
NOTE : For simplicity always assume that upper box is the True part of the condition the data was split on and lower box is the False part

Now let us look at some data to predict a dependent variable (weight in our case) with some independent variables. We would need some data to build our model to make predictions so let's use the following table.

S. no	Age	Sex	Height	Weight
1	21	Male	173	68
2	19	Male	185	78
3	22	Female	152	49.5
4	17	Male	168	61.7
5	20	Female	158	54
6	20	Female	149	46

So, to predict someone's weight based on these independent variables (Age, Sex, Height), we need to keep splitting the groups with certain conditions to arrive at a group with highly similar people in it such that it cannot be split further. Then we average their weights to represent that group.

Here we see that the data was first split based on whether they are older than **18** or not. Further each subgroup was further split on their height and sex subsequently. Therefore, when we get a new data to predict the weight, we run this decision tree and return the average answer that we got from this training data. For example, if our test data is a female and older than **18** our model would say she's **49.83kg**.



Now the important questions,

- Which variable should we first choose to split the data or in what order?
- If it's a continuous variable, on which value should we make the split if that variable is chosen for the split

METRIC: The metric used for this purpose is "*Gini impurity*" (for classification i.e. predicting YES or NO) and "*Mean squared error*" (MSE, for predicting continuous values i.e. regression). These concepts are not beyond our scope of simple understanding, you can see more about them here and here.

FOREST OF TREES

Now we have the decision tree but the only problem is, it generally overfits to the given data. This is where the concept of forest of trees comes in. Creating a forest of non correlated trees and averaging the results over all the trees would eliminate the errors from each tree. We will see how RF model creates this non correlated (thus called random) trees using some parameters and thus how we can tweak the hyper-parameters (just 'parameters' if you like) in the RF model to suit our data or the results we want to best fit the data.

Hyper-parameters in RF

- *n_estimators* : Number of trees to be created in the forest. A sufficient number that would be able to reduce the errors in each tree sufficiently is preferred. It is taken as **100** by default.
- *max_features* : This specifies the number of independent variables to be taken for consideration in each split. That is if we give the value **0.5**, every tree will have **50**% chance of having a different column that the data was split on thus being more random.
- *criterion* : Gives the function to measure the quality of each split. "gini" for classification and "mse" for regression problems.
- *max_depth* : Maximum number of splits in a tree (depth of the tree). Giving a value would restrict the tree from splitting the data till the last leaf containing just a single value. This reduces over fitting.
- *min_samples_split* : Minimum number of samples to be present in an internal node for it be split. Default value is **2**. This too might reduce over fitting.
- *min_samples_leaf* : Minimum number of samples required to be present in the leaf nodes. Default value is **1**.
- *bootstrap* : This is an important feature. It 'bootstraps' the whole dataset for every tree, meaning each tree will get a subset of the dataset which is of the same length as the original dataset but with data samples picked at random with replacement. Example, if abcd represents the dataset it could be bootstrapped to abaa, abbc, abdd, bbbd etc to different trees. Thus, creating a forest of non-correlated trees trained on different subset of data.
- *oob_score* : As some datapoints are left out while bootstrapping/bagging, those points are used to predict the values and check the score for the specific tree. Thus, we get a sense of how good our model is with just the training data. By default, it is 'False' unless specified otherwise.

These are just some of the hyper-parameters that we can tweak to make our forest as non-correlated as possible to predict better results.

Conclusion

Overall, we understand that Random Forest is just a simple concept made of simpler concepts like decision trees. Intuitive understanding of the theory behind this along with some statistical knowledge would enable us to make better prediction models.

Some useful links for reference;

- Gini Impurity & Mean Squared Error
- Intuitive understanding of Random forest