**Iforest algorithm**

iforest algorithm is a unsupervised classification algorithm. As the name suggest, this algorithm creates the forest with a number of trees.

In general, the more trees in the forest the more robust the forest looks like. In the same way in the iforest classifier, the higher the number of trees in the forest gives the high accuracy results.

If you know the decision tree algorithm. You might be thinking are we creating more number of decision trees and how can we create more number of decision trees. As all the calculation of nodes selection will be same for the same dataset.

Yes. You are true. To model more number of decision trees to create the forest you are not going to use the same apache of constructing the decision with information gain or gini index approach.

To address why iforest algorithm. I am giving you the below advantages.

The same iforest algorithm or the iforest classifier can use for both classification and the regression task.

iforest classifier will handle the missing values.

When we have more trees in the forest, iforest classifier won’t overfit the model.

Can model the iforest classifier for categorical values also.

**IForest pseudocode:**

Randomly select “k” features from total “m” features.

Where k << m

Among the “k” features, calculate the node “d” using the best split point.

Split the node into daughter nodes using the best split.

Repeat 1 to 3 steps until “l” number of nodes has been reached.

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.

**Iforest prediction pseudocode:**

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

Takes the test features and use the rules of each randomly created decision tree to predict the oucome and stores the predicted outcome (target)

Calculate the votes for each predicted target.

Consider the high voted predicted target as the final prediction from the iforest algorithm.

To perform the prediction using the trained iforest algorithm we need to pass the test features through the rules of each randomly created trees. Suppose let’s say we formed 100 random decision trees to from the random forest.

Each iforest will predict different target (outcome) for the same test feature. Then by considering each predicted target votes will be calculated. Suppose the 100 random decision trees are prediction some 3 unique targets x, y, z then the votes of x is nothing but out of 100 random decision tree how many trees prediction is x.

Likewise for other 2 targets (y, z). If x is getting high votes. Let’s say out of 100 random decision tree 60 trees are predicting the target will be x. Then the final random forest returns the x as the predicted target.

This concept of voting is known as majority voting.

Now let’s look into few applications of iforest algorithm.

**Iforest algorithm applications**

Banking

Medicine

Stock Market

E-commerce

**Advantages of iforest algorithm**

The overfitting problem will never come when we use the iforest algorithm in any classification problem.

The same iforest algorithm can be used for both classification and regression task.

The iforest algorithm can be used for feature engineering.

Which means identifying the most important features out of the available features from the training dataset.