Random Forest

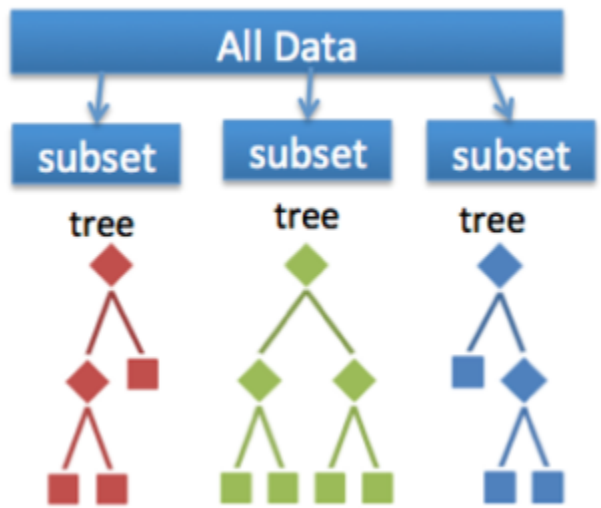
With the demand for more complex computations, we cannot rely on simplistic algorithms.

We must utilize algorithms with higher computational capabilities and one such algorithm is the Random Forest.

Random forest algorithm is a supervised classification and regression algorithm.

This algorithm randomly creates a forest with several trees.

The more trees in the forest the more robust the forest looks like. Similarly, in the random forest classifier, the higher the number of trees in the forest, greater is the accuracy of the results.


All Data
subset
subset
subset
tree
tree
tree

- Random forest builds multiple decision trees (called the forest) and glues them together to get a more accurate and stable prediction.
- The forest it builds is a collection of Decision Trees, trained with the bagging method.

Why Use Random Forest?

- Even though Decision trees are convenient and easily implemented, they lack accuracy.
- Decision trees work very effectively with the training data that was used to build them, but they’re not flexible when it comes to classifying the new sample. Which means that the accuracy during testing phase is very low.
- This happens due to a process called Over-fitting.
- Over-fitting occurs when a model studies the training data to such an extent that it negatively influences the performance of the model on new data.
- This means that the disturbance in the training data is recorded and learned as concepts by the model.
- But the problem here is that these concepts do not apply to the testing data and negatively impact the model’s ability to classify the new data, hence reducing the accuracy on the testing data.
- This is where Random Forest comes in.
- It is based on the idea of bagging, which is used to reduce the variation in the predictions by combining the result of multiple Decision trees on different samples of the data set

Creating a Random Forest

- ***Step 1: Create a Bootstrapped Data Set***
- Bootstrapping is an estimation method used to make predictions on a data set by re-sampling it.
- To create a bootstrapped data set, we must randomly select samples from the original data set.
- A point to note here is that we can select the same sample more than once.

Step 2

Step 2:Creating Decision Trees

- Our next task is to build a Decision Tree by using the bootstrapped data set created in the previous step.
- In Random Forest the entire data set that we created is not considered, instead use a random subset of variables at each step.
- Begin at the root node, here we randomly select the variables as candidates for the root node.
- Out of these variables, we must now select the variable that best separates the samples.
- The more significant predictor assigned as the root node.
- Our next step is to repeat the same process for each of the upcoming branch nodes.
- Again select two variables at random as candidates for the branch node and then choose a variable that best separates the samples.

- Step 3:Go to step1 and repeat
- Random Forest is a collection of Decision Trees.
- Each Decision Tree predicts the output class based on the respective predictor variables used in that tree.
- Finally, the outcome of all the Decision Trees in a Random Forest is recorded and the class with the majority votes is computed as the output class.
- Now create more decision trees by considering a subset of random predictor variables at each step.
- To do this, go back to step 1, create a new bootstrapped data set and then build a Decision Tree by considering only a subset of variables at each step.
- This iteration is performed 100’s of times, therefore creating multiple decision trees with each tree computing the output, by using a subset of randomly selected variables at each step.

- ***Step 4: Predicting the outcome of a new data point***
- ***A*** random forest is created and we run this data down the other decision trees and keep a track of the class predicted by each tree.
- After running the data down all the trees in the Random Forest, we check which class got the majority votes.
- Bootstrap the data and use the aggregate from all the trees to make a decision, this process is known as Bagging.

- ***Step 5: Evaluate the Model***
- Our final step is to evaluate the Random Forest model.
- In a real-world problem, about 1/3rd of the original data set is not included in the bootstrapped data set.
- The Out-Of-Bag data set is used to check the accuracy of the model, since the model wasn't created using this OOB data it will give us a good understanding of whether the model is effective or not.