Course name: **Python programming and analytics by Rahul Sir**

Topic name: **random forest theory (Sumit Batch)**

Video name: **random forest theory Sumit batch**

Video length: **43 minutes 26 seconds**

Ensembled algorithms are those which combines more than one Algorithms of same or different kind for classifying objects.

A Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap Aggregation, commonly known as **bagging**. The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees.

Approach:

- Pick at random K data points from the training set.
- Build the decision tree associated with those K data points.
- Choose the number N tree of trees you want to build and repeat step 1 & 2.
- For a new data point, make each one of your N tree trees predict the value of Y for the data point, and assign the new data point the average across all of the predicted Y values.

Random forest is a supervised learning algorithm. It can be used both for classification and regression. It is also the most flexible and easy to use algorithm. A forest is comprised of trees. It is said that the more trees it has, the more robust a forest is. Random forest creates decision trees on randomly selected data samples, gets prediction from

each tree and selects the best solution by means of voting. It also provides a pretty good indicator of the feature importance.

Random forest has a variety of applications, such as recommendation engines, image classification and feature selection. It can be used to classify loyal loan applicants, identify fraudulent activity and predict diseases.

It technically is an ensemble method (based on the divide-and-conquer approach) of decision trees generated on a randomly split dataset. This collection of decision tree classifiers is also known as the forest. The individual decision trees are generated using an attribute selection indicator such as information gain, gain ratio, and Gini index for each attribute. Each tree depends on an independent random sample. In a classification problem, each tree votes and the most popular class is chosen as the final result. In the case of regression, the average of all the tree outputs is considered as the final result. It is simpler and more powerful compared to the other non-linear classification algorithms.

How does the algorithm work?

It works in four steps:

1. Select random samples from a given dataset.
2. Construct a decision tree for each sample and get a prediction result from each decision tree.
3. Perform a vote for each predicted result.
4. Select the prediction result with the most votes as the final prediction.

Random forest also offers a good feature selection indicator. Scikit-learn provides an extra variable with the model, which shows the relative importance or contribution of each feature in the prediction. It automatically computes the relevance score of each feature in the training phase. Then it scales the relevance down so that the sum of all scores is 1.

This score will help you choose the most important features and drop the least important ones for model building.

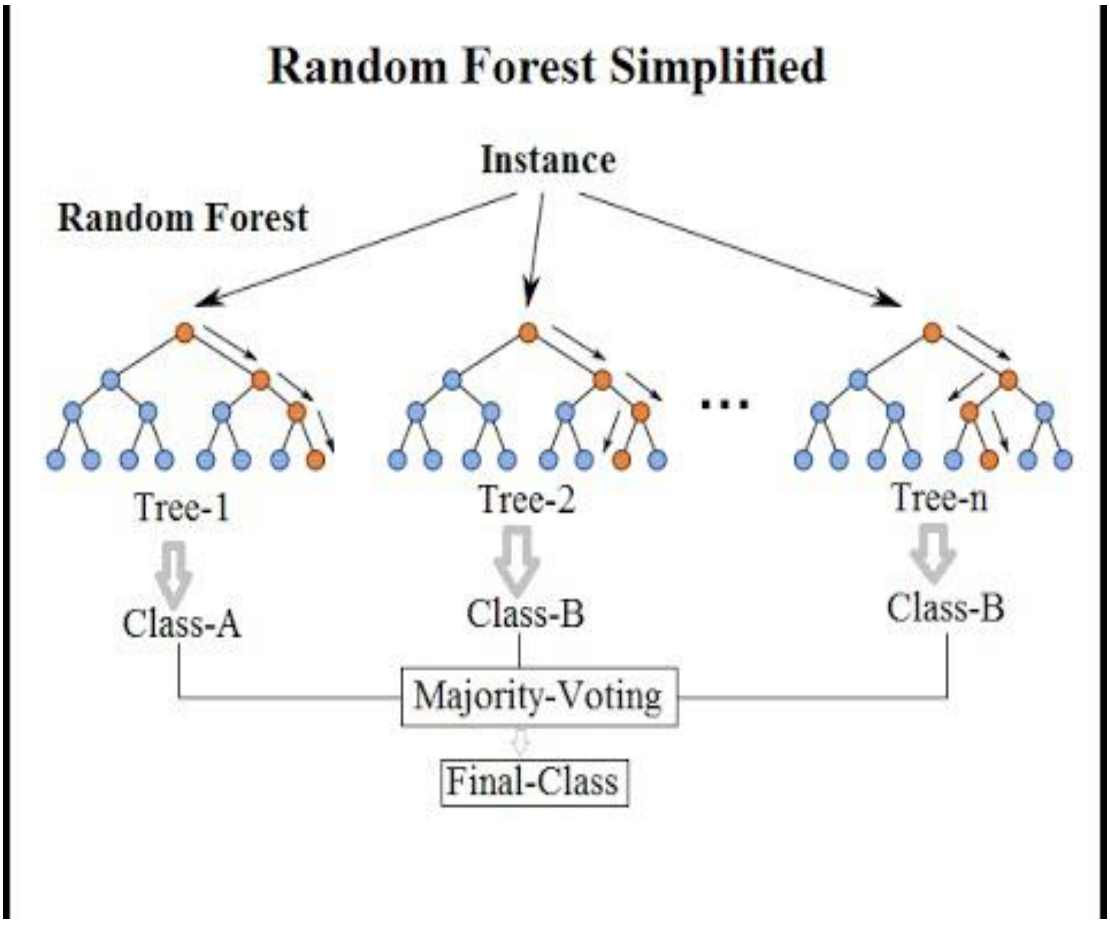
Random forest uses gini importance or mean decrease in impurity (MDI) to calculate the importance of each feature. Gini importance is also known as the total decrease in node impurity. This is how much the model fit or accuracy decreases when you drop a variable. The larger the decrease, the more significant the variable is. Here, the mean decrease is a significant parameter for variable selection. The Gini index can describe the overall explanatory power of the variables.

- **Ensemble**: use of ***multiple learning algorithms*** to obtain better ***predictive performance*** than could be obtained from any of the constituent learning algorithms
- **Bootstrap aggregating**, also called **bagging:** Given a standard training set *D* of size *n*, bagging generates *m* new training sets Di, each of size *n'*, by sampling from D uniformly with replacement. By sampling with replacement, some observations may be repeated in each Di. The kind of sample is called Bootstrap. The *m* models are fitted using the above *m* bootstrap samples and combined (aggregated) by averaging the output (for regression) or voting (for classification).

Ensemble Technique

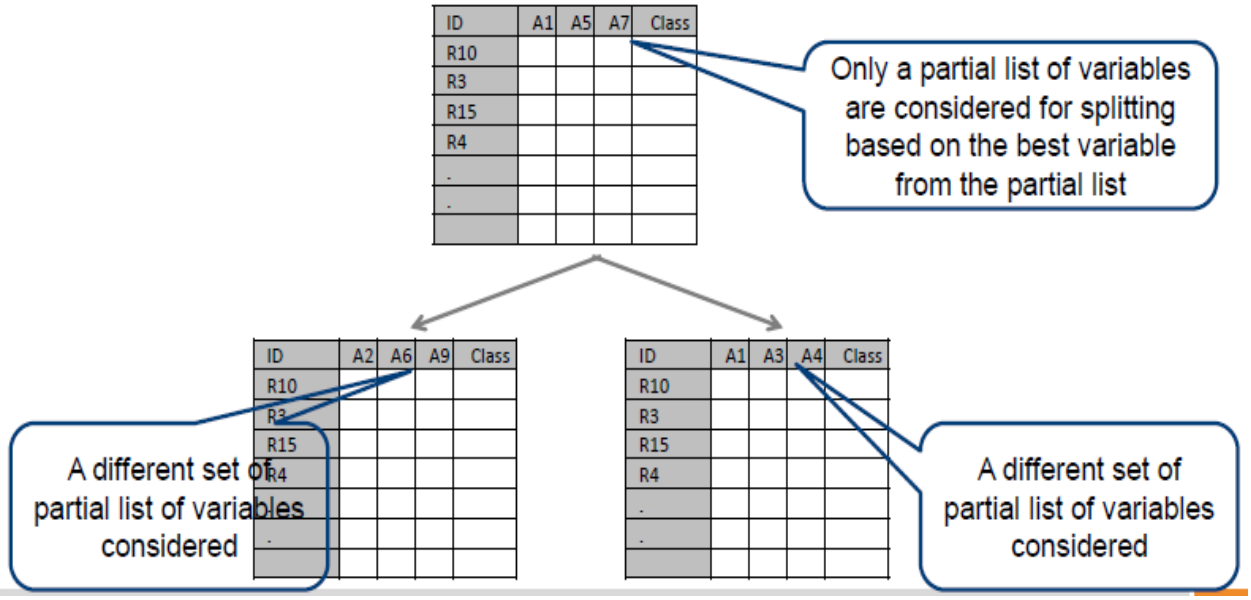
- Involves constructing multitude of decision trees at training time
- Prediction is based on mode for classification tree and mean for regression tree
- Help reduce over-fitting

Note: there is possibility of high over-fitting at individual tree level but averaging removes the bias



Step1: random sampling

- Step 2: Building the tree for each sample with only partial set of 'm' variable being considered at each node
- m << M where M is total number of predictor variables



- **Step 3**: Classifying
  - Based on 'n' samples… 'n' tree are built
  - Each record is classified based on the n tree
  - Final class for each record is decided based on voting
- **Some original papers on RF proved that the RF error rate depends on two factors**
  - **1.** The *correlation* between any two trees in the forest. Increasing the correlation increases the forest error rate.
  - **2.** The *strength* of each individual tree in the forest. A tree with a low error rate is a strong classifier. Increasing the strength of the individual trees decreases the forest error rate.

- **3.**Reducing m reduces both the correlation and the strength. Increasing it increases both. Somewhere in between is an "optimal" range of m -usually quite wide

Advantages:

- Random forests is considered as a highly accurate and robust method because of the number of decision trees participating in the process.
- It does not suffer from the overfitting problem. The main reason is that it takes the average of all the predictions, which cancels out the biases.
- The algorithm can be used in both classification and regression problems.
- Random forests can also handle missing values. There are two ways to handle these: using median values to replace continuous variables, and computing the proximity-weighted average of missing values.
- You can get the relative feature importance, which helps in selecting the most contributing features for the classifier.

Disadvantages:

- Random forests is slow in generating predictions because it has multiple decision trees. Whenever it makes a prediction, all the trees in the forest have to make a prediction for the same given input and then perform voting on it. This whole process is time-consuming.

- The model is difficult to interpret compared to a decision tree, where you can easily make a decision by following the path in the tree.

Random Forests vs Decision Trees

- Random forest is a set of multiple decision trees.
- Deep decision trees may suffer from overfitting, but random forest prevents overfitting by creating trees on random subsets.
- Decision trees are computationally faster.
- Random forest is difficult to interpret, while a decision tree is easily interpretable and can be converted to rules.