**Decision Tree and Random Forest**

**Decision Tree**

Decision Tree Algorithm is a type of supervised machine learning model used for both Classification and Regression. It is simple nested if else condition. It uses tree representation to solve the problem.

Rootnode

Internal

node

Leaf node

**Root Node:**  Starting of the Tree.

**Internal Node:**  There are next after the root node is denoted by internal node.

**Leaf Node:**  This is last and there are nothing after this node. This represents the output. It represents a class and cannot split further.

This algorithm works by dividing the whole dataset into a tree-like structure based on some rules and conditions and then gives prediction based on those conditions. Let’s understand the approach to decision tree with a basic scenario. Suppose it’s Friday night and you are not able to decide if you should go out or stay at home. Let the decision tree decide it for you.

IS Time >10PM

NO

YES

Do you have enough money ?

Don’t Go

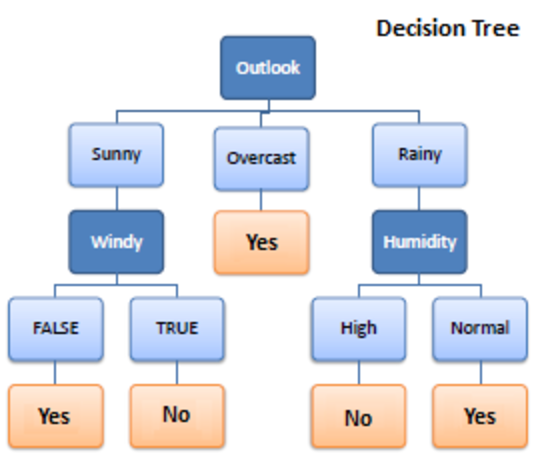
NO

Don’t go

YES

Go and Enjoy

Although we may or may not use the decision tree for such decisions, this was a basic example to help you understand how a decision tree makes a decision. So how did it work? \* It selects a root node based on a given condition, e.g. our root node was chosen as time >10 pm. \* Then, the root node was split into child notes based on the given condition. The right child node in the above figure fulfilled the condition, so no more questions were asked. \* The left child node didn’t fulfil the condition, so again it was split based on a new condition. \* This process continues till all the conditions are met or if you have predefined the depth of your tree, e.g. the depth of our tree is 3, and it reached there when all the conditions were exhausted.



The flow of this tree works downwards beginning at the top with the outlook. The outlook has three options like sunny, overcast, Rainy. If sunny, we travel down to next level. It will be windy? If it is True, we can choose not to go out to play or else if it is false then we can decide to go out to play.

If the outlook changes to overcast, we would end there and decide to play. If the outlook was rainy, we would then look at humidity. If the humidity was high, we would not play, if the humidity is normal, we would play.

Tree depth is an important concept. This represents how many questions are asked before we reach our predicted classification. We can see that the deepest the tree gets in the example above is two. The sunny and rainy routes both have a depth of two. The overcast route only has a depth of one, although the overall tree depth is denoted by its longest route. Thus, this tree has a depth of two.

***Pure Split:***

It has good conditions to know the decision, it has only single class label. Here we get leaf node.

**How to select the right feature?**

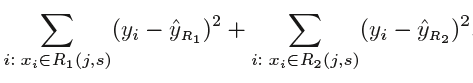
**Decision Tree for Regression**

When performing regression with a decision tree, we try to divide the given values of X into distinct and non-overlapping regions, e.g. for a set of possible values X1, X2,, Xp; we will try to divide them into J distinct and non-overlapping regions R1, R2, . . . , RJ. For a given observation falling into the region Rj , the prediction is equal to the mean of the response(y) values for each training observations(x) in the region Rj. The regions R1,R2, . . . , RJ are selected in a way to reduce the following sum of squares of residuals :

Where, yrj (second term) is the mean of all the response variables in the region ‘j’.

#### Recursive binary splitting (Greedy approach)

As mentioned above, we try to divide the X values into j regions, but it is very expensive in terms of computational time to try to fit every set of X values into j regions. Thus, decision tree opts for a top-down greedy approach in which nodes are divided into two regions based on the given condition, i.e. not every node will be split but the ones which satisfy the condition are split into two branches. It is called greedy because it does the best split at a given step at that point of time rather than looking for splitting a step for a better tree in upcoming steps. It decides a threshold value (say s) to divide the observations into different regions(j) such that the RSS for Xj>= s and Xj <s is minimum.



Here for the above equation, j and s are found such that this equation has the minimum value. The regions R1, R2 are selected based on that value of s and j such that the equation above has the minimum value. Similarly, more regions are split out of the regions created above based on some condition with the same logic. This continues until a stopping criterion (predefined) is achieved. Once all the regions are split, the prediction is made based on the mean of observations in that region.

The process mentioned above has a high chance of overfitting the training data as it will be very complex.

### **Classification Trees**

Regression trees are used for quantitative data. In the case of qualitative data or categorical data, we use classification trees. In regression trees, we split the nodes based on RSS criteria, but in classification, it is done using classification error rate, Gini impurity and entropy. Let’s understand these terms in detail.

1. Entropy

Entropy measures the impurity of a collection of examples. It depends from the distribution of the random variable p.

* s is a collection of training examples
* the proportion of the positive examples in s
* the proportion of the negative examples in s

Entropy(s) = -

Entropy is to select right feature. Entropy ranges between 0 & 1. But we need only 0 values.so, we can calculate every node and calculate every node and know how pure the split is.

If we have 3Yes,2No both are same then when we calculate the entropy, we calculate the entropy, we get 1, This is worst split. Entropy for leaf node (means only one class). Only one class that is only Yes is there when, we have only one class, then we will have pure split. We get 0 if it has only one class, if we get entropy as 1 then it will have both Class Yes & No in same.

***Information Gain:***

Information gain is used to determine which feature/attribute gives us the maximum information about a class.

* Information Gain is based on the concept of entropy, which is degree of Uncertainty, impurity or disorder
* Information Gain aims to reduce the level of entropy starting from the root node to the leaf nodes.
* The Greater the reduction in the Uncertainty the more information is gained about y from x

Information gain is going to calculate the entropy of all the nodes for each & every node in our decision tree and subtract that from total entropy (The entropy of the internal nodes are separately calculated and those are subtract from the root node). If we have maximum information gain it means that we are having less entropy.

***NOTE:***

Entropy: We calculate entropy for single node

Information Gain: We calculate entropy for all nodes and subtract that from total entropy.

***2)Gini Index***

Gini index or Gini impurity measures the degree or probability of a particular variable.

Gini Index= 1-

=1-

For example:

If we have one node having 2 Yes, 2 No

4 Records----- 2Yes, 2 No

* Probability of positive class = 2/4
* Probability of negative class=2/4

If we apply these values in Gini index then,

Gini Index = 0.5

Gini Index ranges from 0 & 0.5

For pure split Gini index is 0

For impure split Gini index is 0.5

Entropy and Gini index are methods to find Purity of split.

**Hyperparameters of Decision Tree**

Hyperparameter tuning is searching the hyperparameter space for a set of values that will optimize your model architecture.

* Criterion: The function to measure the quality of a split. Supported criteria are “Gini” for the Gini impurity and “entropy” for the information gain.
* Splitter: This is how the decision tree searches the features for a split. The default values are set to “best”. That is, for each node, the algorithm considers all the features and chooses the best split. If you decide to set the splitter parameter to “random”, then a random subset of features will be considered.
* max\_depth: This determines the maximum depth of the tree. We use a depth of two to make our decision Tree. This will often result in over-fitted decision trees. The depth parameter is one of the ways in which we can regularize the tree, or limit the way it grows to prevent over-fitting. The tree perfectly fits the training data and fails to generalize on testing data.
* min\_samples\_split: Ideal range is 1 to 40 min\_samples\_split specifies the minimum number of samples required to split an internal node, while min\_samples\_leaf specifies the minimum number of samples required to be at a leaf node.
* min\_samples\_leaf: The minimum number of samples required to be at a leaf node. Similar to min sample split, this describes the minimum number of samples at the leaf, the base of tree. Ideal range is 1 to 20

##### ***Advantages of Decision Tree:***

* It can be used for both Regression and Classification problems.
* Decision Trees are very easy to grasp as the rules of splitting is clearly mentioned.
* Complex decision tree models are very simple when visualized. It can be understood just by visualising.
* Scaling and normalization are not needed.

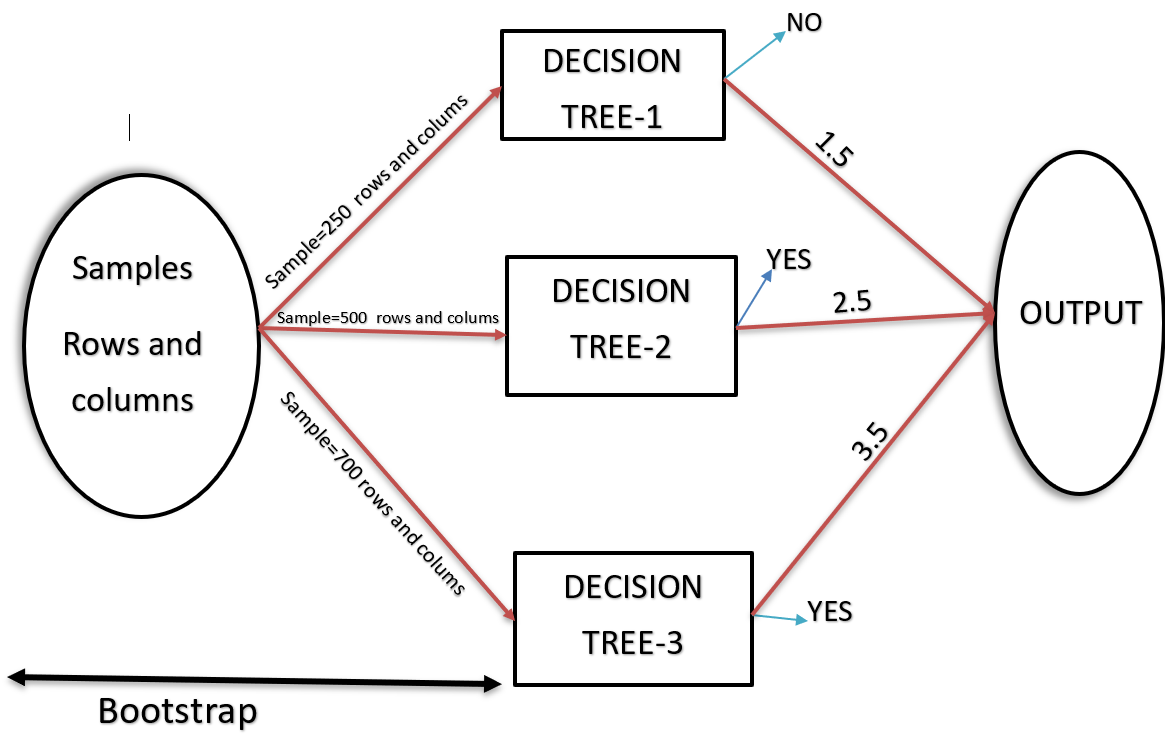
##### ***Disadvantages of Decision Tree:***

* A small change in data can cause instability in the model because of the greedy approach.
* Probability of overfitting is very high for Decision Trees.
* It takes more time to train a decision tree model than other classification algorithms.

# **what is random forest**

* A random forest is a supervised machine learning algorithm that is constructed from decision tree algorithms.
* A random forest is a machine learning technique that’s used to solve regression and classification problems. It utilizes ensemble learning, which is a technique that combines many classifiers to provide solutions to complex problems.
* The (random forest) algorithm establishes the outcome based on the predictions of the decision trees. It predicts by taking the average or mean of the output from various trees. Increasing the number of trees increases the precision of the outcome.

**working of Random forest**



**Output side called as Aggregation**

**What is bootstrap in random forest?**

* When training, each tree in a random forest learns from a random sample of the data points. The samples are drawn with replacement, known as bootstrapping, which means that some samples will be used multiple times in a single tree.

**For regression task it will take average**

**For classification it will count the output**

## **Hyperparameter Tuning**

* n\_estimators = number of trees in the foreset
* max\_features =These are the maximum number of features Random Forest is allowed to try in individual tree. There are multiple options available in Python to assign maximum features
* max\_depth =The depth of each tree in the forest. The deeper the tree, the more splits it has and it captures more information about the data
* min\_samples\_split =the minimum number of samples required to split an internal node. This can vary between considering at least one sample at each node to considering all of the samples at each node
* min\_samples\_leaf = minimum number of data points allowed in a leaf node

## **Advantages and Disadvantages of Random Forest Algorithm**

#### *Advantages*

* It can be used in classification and regression problems.
* It solves the problem of overfitting as output is based on majority voting or averaging.
* It performs well even if the data contains null/missing values.
* Each decision tree created is independent of the other; thus, it shows the property of parallelization.
* It is highly stable as the average answers given by a large number of trees are taken.
* It maintains diversity as all the attributes are not considered while making each decision tree though it is not true in all cases.
* It is immune to the curse of dimensionality. Since each tree does not consider all the attributes, feature space is reduced.
* We don’t have to segregate data into train and test as there will always be 30% of the data, which is not seen by the decision tree made out of bootstrap.

#### *Disadvantages*

* Random forest is highly complex compared to decision trees, where decisions can be made by following the path of the tree.
* Training time is more than other models due to its complexity. Whenever it has to make a prediction, each decision tree has to generate output for the given input data.

***Conclusion***

Random forest is a great choice if anyone wants to build the model fast and efficiently, as one of the best things about the random forest is it can handle missing values. It is one of the best techniques with high performance, widely used in various industries for its efficiency. It can handle binary, continuous, and categorical data. Overall, random forest is a fast, simple, flexible, and robust model with some limitations.