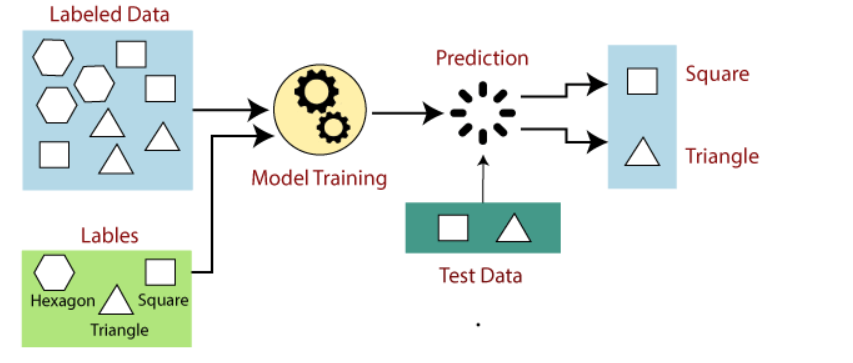
**Supervised Learning:**

* In Supervised Learning or Supervised Machine learning, the model is trained well using labeled data i.e. for some input data it is already matched with the correct output. And on the basis of trained data, the models/machines will predict the output for the unknown data.
* The aim of supervised learning is to find a mapping function to map the input variable(x) with the output variable(y).
* In the real-world, supervised learning can be used for Risk Assessment, Image classification, Fraud Detection, spam filtering, etc.

**How Supervised Learning works?**

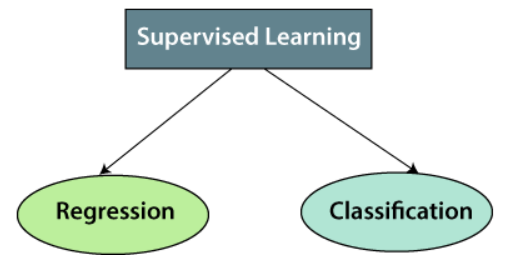
****

**Steps Involved in Supervised Learning:**

* First Determine the type of training dataset
* Collect/Gather the labeled training data.
* Split the training dataset into a training dataset, test dataset, and validation dataset.
* Determine the input features of the training dataset, which should have enough knowledge so that the model can accurately predict the output.
* Determine the suitable algorithm for the model, such as support vector machine, decision tree, etc.
* Execute the algorithm on the training dataset. Sometimes we need validation sets as the control parameters, which are the subset of training datasets.
* Evaluate the accuracy of the model by providing the test set. If the model predicts the correct output, which means our model is accurate

**Types of Supervised Learning:**

**2** types:



**Regression:**

* Regression is used when there is a relation between input variable and output variable. Regression is used for prediction of continuous variables such as Weather forecast, business trends(Stock market) etc.
* Some popular regression algorithms under supervised learning are:
  + Linear Regression.
  + Regression Trees.
  + Non-Linear Regression.
  + Bayesian Linear Regression.
  + Polynomial Regression.

**Classification:**

* Classification algorithm is a Supervised Learning technique that is used to identify the category of new observations on the basis of training data.
* In Classification, a program learns from the given dataset or observations and then classifies new observations into a number of classes or groups. Such as, Yes or No, 0 or 1, Spam or Not Spam, cat or dog, etc.
* Classes can be called as targets/labels or categories.
* In the classification algorithm, a discrete output function(y) is mapped to input variable(x).
  + y=f(x), where y = categorical output

**Types of Classification:**

**Two** types of classification.

* **Binary classification** : If the classification problem has two possible outcomes such as YES-NO, MALE-FEMALE, SPAM-NOTSPAM etc.
* **Multi-class Classification**: If the classification problem has more than two outcomes. Ex: Classifications of types of crops, Classification of types of music.

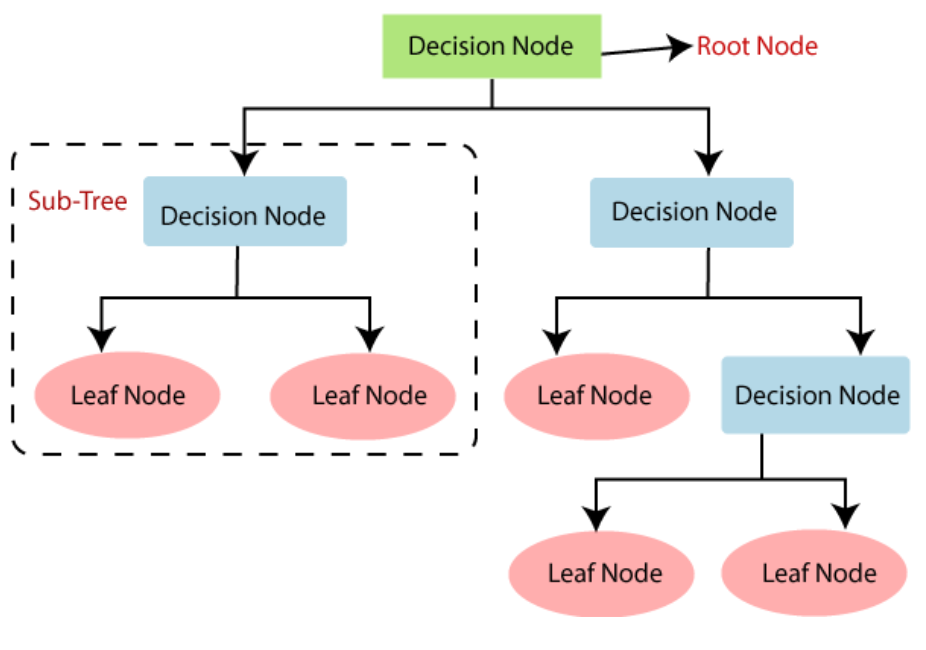
**Some popular classification algorithms:**

* **Linear Models**
  + Logistic Regression.
  + Support Vector Machine.
* **Non Linear Models**
  + K-Nearest Neighbors.
  + Kernel SVM.
  + Naïve Bayes.
  + Decision Tree Classification.
  + Random Forest Classification.

**Decision Tree:**

* Decision tree is a supervised learning technique that can be used for both classification and regression problems, but mostly preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.
* In the decision tree there are two nodes, which are Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches whereas leaf nodes are output of those decisions and do not contain any further progress.
* The decision will be taken on the basis of features given in the dataset.
* It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.
* In order to build a tree we use the **CART** algorithm which stands for Classification and Regression Tree Algorithm.
* A decision tree simply asks a question and the answer will be either **YES** or **NO** and based on the answer it splits into subtrees.

**What does the decision tree look like?**

****

**Why to use Decision Trees?**

* Decision trees usually mimic human thinking ability while making a decision, so it's easy to understand.
* The decision trees can be easily understood because they are tree-like structures.

**Decision tree terminologies:**

* **Root Node:** Root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneous sets.
* **Leaf Node:** Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.
* **Splitting:** Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.
* **Branch/Subtree:** A tree formed by splitting the tree.
* **Pruning:** Pruning is the process of removing the unwanted branches from the tree.
* **Parent/Child node:** The root node of the tree is called the parent node, and other nodes are called the child nodes.

**How does the decision tree algorithm work?**

* The decision trees for predicting the class of a given dataset, the algorithm starts from the root node of the tree.
* This algorithm compares the values of root attribute with the dataset attribute and based on the comparison, follows the branch and jumps onto the next node.
* For the next node, the algorithm again compares the attribute value with the other sub-nodes and moves further. It continues the process until it reaches the leaf node of the tree.

**The complete process:**

* **Step 1:** Begin the tree with the root node S i.e. best attribute in the dataset using Attribute Selection Measure.
* **Step 2 :** Divide the S into subsets that contain the possible values for best attributes.
* **Step 3 :** Generate the decision tree node, which contains the best attribute.
* **Step 4:** Recursively make new decision trees using the subsets of the dataset created in step -2 Continue this process until a stage is reached where you cannot further classify the nodes and call the final node as a leaf node.

**Attribute Selection Measures:**

While implementing a Decision tree, the main issue arises that how to select the best attribute for the root node and sub nodes. So to solve this we use Attribute Selection Measure or ASM.

There are **two** popular techniques for Attribute Selection Measure:

1. **Information Gain(ID3)**.
2. **Gini Index**.

**Information Gain(ID3):**

* Information Gain: Information Gain is the measurement of changes in entropy after the segmentation of a dataset based on attribute.
* It calculates how much information a feature provides us about a class.
* According to the value of information gain, we split the node and build the decision tree.

A decision tree algorithm always tries to maximize the value of information gain, and a node/attribute having the highest information gain is split first. It can be calculated using the below formula:

* + **Information Gain= Entropy(S)- [(Weighted Avg) \*Entropy(each feature) ]**

**Entropy:**

* Entropy is used to measure the impurity in a given attribute. It also specifies randomness in the data. Entropy is calculated as:
  + E(S) =
  + ‘p’, denotes the probability of E(S), which denotes the entropy. The feature or attribute with the highest ID3 gain is used as the root for the splitting.

**Gini Index:**

* Gini index is a measure of impurity or purity used while creating a decision tree in the CART(Classification and Regression Tree) algorithm.
* An attribute with the low Gini index should be preferred as compared to the high Gini index.
* It only creates binary splits, and the CART algorithm uses the Gini index to create binary splits.
* Gini Index is calculated as:
  + Gini Index = 1 -

Pj = probability of an object being classified into a particular class.

**Advantages of Decision Tree:**

* It is simple to understand as it follows the same process which a human follows while making any decision in real-life.
* It can be very useful for solving decision-related problems.
* It helps to think about all the possible outcomes for a problem.

**Disadvantages of Decision Tree:**

* It may have an overfitting issue, which can be resolved using the **Random Forest algorithm.**
* For more class labels, the computational complexity of the decision tree may increase.

## **Introduction of Random Forest**

* Random forest is a *Supervised Machine Learning Algorithm* that is *used widely in Classification and Regression problems*. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.
* One of the most important features of the Random Forest Algorithm is that it can handle the data set containing *continuous variables* as in the case of regression and *categorical variables* as in the case of classification. It performs better results for classification problems.

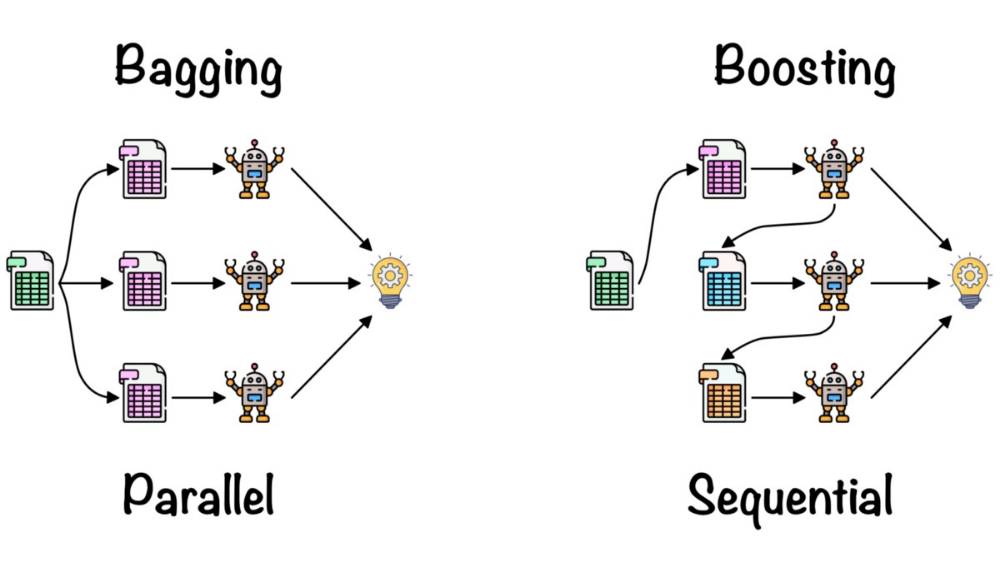
## **Working of Random Forest Algorithm**

* Before understanding the working of the random forest we must look into the ensemble technique.
* *Ensemble*simplymeans combining multiple models. Thus a collection of models is used to make predictions rather than an individual model.

***Ensemble uses two types of methods*:**

1. **Bagging**– It creates a different training subset from sample training data with replacement & the final output is based on majority voting. For example, Random Forest.

2. **Boosting**– It combines weak learners into strong learners by creating sequential models such that the final model has the highest accuracy. For example, ADA BOOST, XG BOOST.

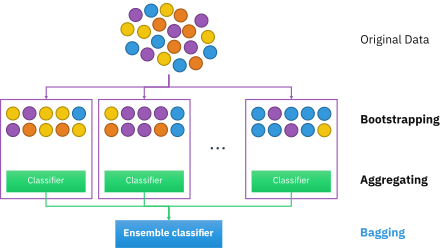


#### 

#### 

## Bagging

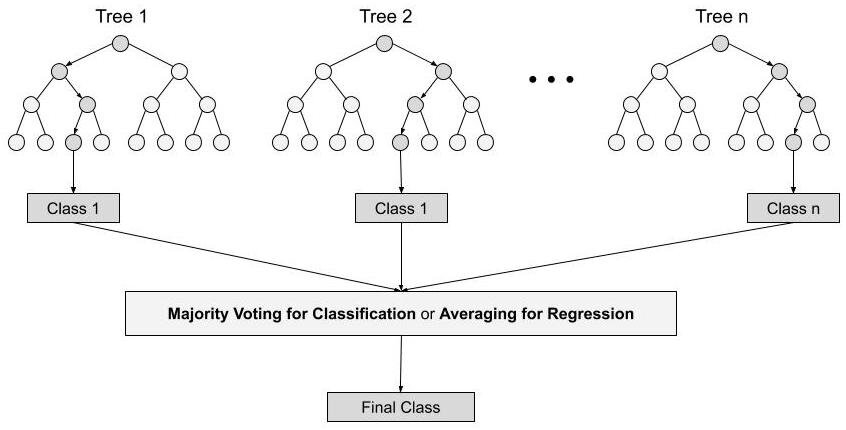
* Bagging, also known as *Bootstrap Aggregation* is the ensemble technique used by random forest.
* Bagging chooses a random sample from the data set. Hence each model is generated from the samples (Bootstrap Samples) provided by the Original Data with replacement known as *row sampling*.



* Now let’s look at an example by breaking it down with the help of the following figure. Here the bootstrap sample is taken from actual data (Bootstrap sample 01, Bootstrap sample 02, and Bootstrap sample 03) with a replacement which means there is a high possibility that each sample won’t contain unique data.
* Now the model (Model 01, Model 02, and Model 03) obtained from this bootstrap sample is trained independently. Each model generates results as shown.
* Now Happy emoji is having a majority when compared to sad emoji. Thus based on majority voting final output is obtained as Happy emoji.

**Steps involved in random forest algorithm:**

* Step 1: In Random forest n number of random records are taken from the data set having k number of records.
* Step 2: Individual decision trees are constructed for each sample.
* Step 3: Each decision tree will generate an output.
* Step 4: Final output is considered based on *Majority Voting or Averaging* for Classification and regression respectively.



## Important Features of Random Forest

* Diversity- Not all attributes/variables/features are considered while making an individual tree, each tree is different.
* Immune to the curse of dimensionality- Since each tree does not consider all the features, the feature space is reduced.
* Parallelization-Each tree is created independently out of different data and attributes. This means that we can make full use of the CPU to build random forests.

4. Train-Test split- In a random forest we don’t have to segregate the data for train and test as there will always be 30% of the data which is not seen by the decision tree.

5. Stability- Stability arises because the result is based on majority voting/ averaging.

## Difference Between Decision Tree & Random Forest

Random forest is a collection of decision trees; still, there are a lot of differences in their behavior.

| **Decision trees** | **Random Forest** |
| --- | --- |
| 1. Decision trees normally suffer from the problem of overfitting if it’s allowed to grow without any control. | 1. Random forests are created from subsets of data and the final output is based on average or majority ranking and hence the problem of overfitting is taken care of. |
| 2. A single decision tree is faster in computation. | 2. It is comparatively slower. |
| 3. When a data set with features is taken as input by a decision tree it will formulate some set of rules to do prediction. | 3. Random forest randomly selects observations, builds a decision tree and the average result is taken. It doesn’t use any set of formulas. |

* Thus random forests are much more successful than decision trees only if the trees are diverse and acceptable.

## Important Hyperparameters

Hyperparameters are used in random forests to either enhance the performance and predictive power of models or to make the model faster.

## Following hyperparameters increases the predictive power:

1. **n\_estimators**– number of trees the algorithm builds before averaging the predictions.

2. ***max\_features****–* maximum number of features random forest considers splitting a node.

*3.* ***mini\_sample\_leaf****–* determines the minimum number of leaves required to split an internal node.

*Following hyperparameters increases the speed:*

*1.* ***n\_jobs****–* it tells the engine how many processors it is allowed to use. If the value is 1, it can use only one processor but if the value is -1 there is no limit.

*2.* ***random\_state****–* controls randomness of the sample. The model will always produce the same results if it has a definite value of random state and if it has been given the same hyperparameters and the same training data.

*3.* ***oob\_score*** *– OOB* means out of the bag. It is a random forest cross-validation method. In this one-third of the sample is not used to train the data instead used to evaluate its performance. These samples are called out of bag samples.

## Advantages and Disadvantages of Random Forest Algorithm

## **Advantages**

* 1. It can be used in classification and regression problems.
* 2. It solves the problem of overfitting as output is based on majority voting or averaging.
* 3. It performs well even if the data contains null/missing values.
* 4. Each decision tree created is independent of the other thus it shows the property of parallelization.
* 5. It is highly stable as the average answers given by a large number of trees are taken.
* 6. It maintains diversity as all the attributes are not considered while making each decision tree though it is not true in all cases.
* 7. It is immune to the curse of dimensionality. Since each tree does not consider all the attributes, feature space is reduced.
* 8. 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

* 1. Random forest is highly complex when compared to decision trees where decisions can be made by following the path of the tree.
* 2. Training time is more compared to 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.

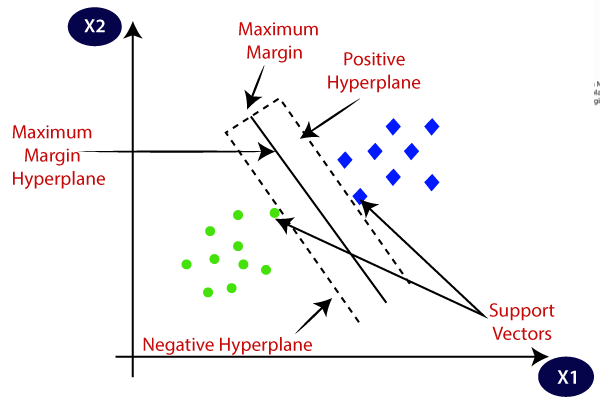
## 

Support Vector Machine Algorithm

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:



SVM algorithm can be used for Face detection, image classification, text categorization, etc.

Types of SVM

* **Linear SVM**: Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.
* **Non-linear SVM**: Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

Hyperplane and Support Vectors in the SVM algorithm:

* **Hyperplane**: There can be multiple lines/decision boundaries to segregate the classes in n-dimensional space, but we need to find out the best decision boundary that helps to classify the data points. This best boundary is known as the hyperplane of SVM.

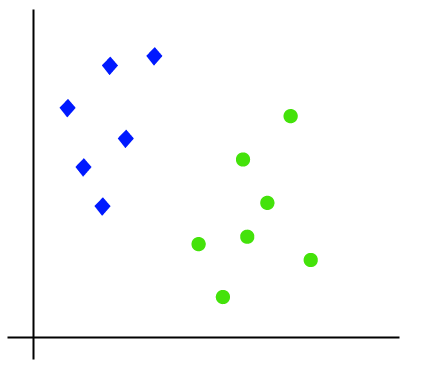
The dimensions of the hyperplane depend on the features present in the dataset, which means if there are 2 features (as shown in image), then hyperplane will be a straight line. And if there are 3 features, then hyperplane will be a 2-dimension plane.

We always create a hyperplane that has a maximum margin, which means the maximum distance between the data points.

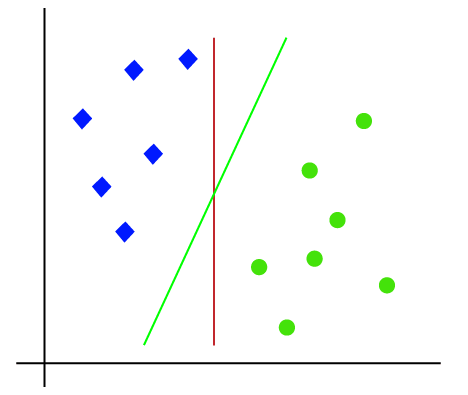
* **Support Vectors:** The data points or vectors that are the closest to the hyperplane and which affect the position of the hyperplane are termed as Support Vector. Since these vectors support the hyperplane, hence called a Support vector.

**Linear SVM**:

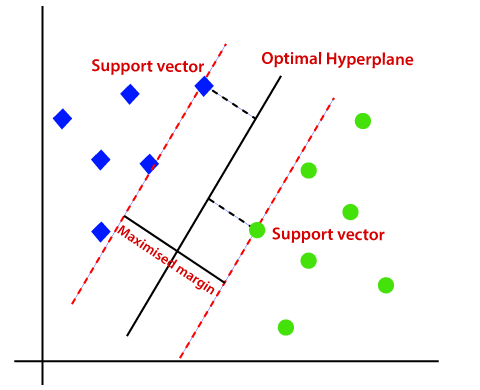
The working of the SVM algorithm can be understood by using an example. Suppose we have a dataset that has two tags (green and blue), and the dataset has two features x1 and x2. We want a classifier that can classify the pair(x1, x2) of coordinates in either green or blue. Consider the below image:



So as it is 2-d space so by just using a straight line, we can easily separate these two classes. But there can be multiple lines that can separate these classes. Consider the below image:



Hence, the SVM algorithm helps to find the best line or decision boundary; this best boundary or region is called as a hyperplane. SVM algorithm finds the closest point of the lines from both the classes. These points are called support vectors. The distance between the vectors and the hyperplane is called as margin. And the goal of SVM is to maximize this margin. The hyperplane with maximum margin is called the optimal hyperplane.



**Non-Linear SVM:**

If data is linearly arranged, then we can separate it by using a straight line, but for non-linear data, we cannot draw a single straight line. Consider the below image:

