Sigireddy bindhu sai bhargavi

**Regression and classification algorithms**

**How the Data Interpretation and Inference has done:-**

1. Collecting each topic analyzing and understanding
2. Data collection and writing a code and checking the accuracy
3. Most used libraries are Numpy, pandas,matplotlib, seaborn and Scikit-learn

Machine learning

Machine learning enables a machine to automatically learn from past data, improve performance from experiences, and predict things without being explicitly programmed.

Machine learning uses various algorithms for building mathematical models and making predictions using historical data or information. Currently, it is being used for various tasks such as image recognition, speech recognition, email filtering, Facebook auto-tagging, recommender system, and many more.

The term machine learning was first introduced by Arthur Samuel in 1959.

It learns from historical data, builds the prediction models, and whenever it receives new data, predicts the output for it. The accuracy of predicted output depends upon the amount of data, as the huge amount of data helps to build a better model which predicts the output more accurately.



## Classification of Machine Learning:

1. Supervised learning
2. Unsupervised learning
3. Reinforcement learning

### **Supervised Learning**

Supervised learning is a type of machine learning method in which we provide sample labelled data to the machine learning system in order to train it, and on that basis, it predicts the output. Example of supervised learning is spam filtering.

Supervised learning can be grouped further in two categories of algorithms:

* Classification
* Regression

**Classification:**

Classification algorithms are used when the output variable is categorical, which means there are two classes such as Yes-No, Male-Female, True-false, etc.

Spam Filtering,

* Random Forest
* Decision Trees
* Logistic Regression
* Support vector Machines

**Regression:**

Regression algorithms are used if there is a relationship between the input variable and the output variable. It is used for the prediction of continuous variables, such as Weather forecasting, Market Trends, etc. Below are some popular Regression algorithms which come under supervised learning:

* Linear Regression
* Regression Trees
* Non-Linear Regression
* Bayesian Linear Regression
* Polynomial Regression



**Unsupervised Learning**

Unsupervised learning is a learning method in which a machine learns without any supervision. We don’t have any labelled data and also we don’t have any predetermined results and The goal of unsupervised learning is to restructure the input data into new features or a group of objects with similar patterns. Example of unsupervised learning is market basket analysis,recommender system

It can be further classifieds into two categories of algorithms:

* **Clustering**
* **Association**
* Clustering: Clustering is a method of grouping the objects into clusters such that objects with most similarities remains into a group and has less or no similarities with the objects of another group. Cluster analysis finds the commonalities between the data objects and categorizes them as per the presence and absence of those commonalities.
* Association: An association rule is an unsupervised learning method which is used for finding the relationships between variables in the large database. It determines the set of items that occurs together in the dataset. Association rule makes marketing strategy more effective. Such as people who buy X item (suppose a bread) are also tend to purchase Y (Butter/Jam) item. A typical example of Association rule is Market Basket Analysis.

**Reinforcement Learning**

Reinforcement learning is a feedback-based learning method, in which a learning agent gets a reward for each right action and gets a penalty for each wrong action. The agent learns automatically with these feedbacks and improves its performance. Example of reinforcement leaning is automated robots and product advertising.

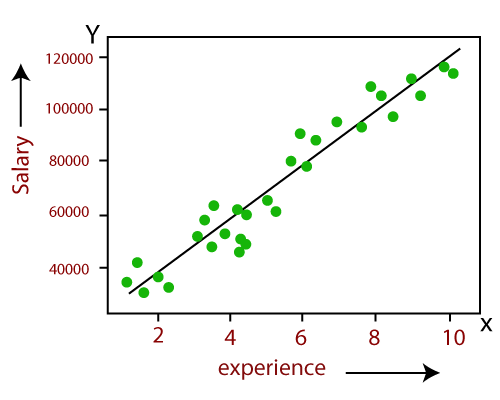
# Applications of Machine learning



### **Linear Regression:**

* Linear regression is a statistical regression method which is used for predictive analysis
* It is one of the very simple and easy algorithms which works on regression and shows the relationship between the continuous variables.
* Linear regression shows the linear relationship between the dependent(y) and one or more independent(x) variables

If there is only one input variable (x), then such linear regression is called simple linear regression. And if there is more than one input variable, then such linear regression is called multiple linear regression.



* Below is the mathematical equation for Linear regression Y= aX+b

**Y= dependent variables (target variables),  
X= Independent variables (predictor variables),  
a and b are the linear coefficients**

## Finding the best fit line:

When working with linear regression, our main goal is to find the best fit line that means the error between predicted values and actual values should be minimized. The best fit line will have the least error.

The different values for weights or the coefficient of lines (a0, a1) gives a different line of regression, so we need to calculate the best values for a0 and a1 to find the best fit line, so to calculate this we use cost function.

### **Cost function-**

* The different values for weights or coefficient of lines (a0, a1) gives the different line of regression, and the cost function is used to estimate the values of the coefficient for the best fit line.
* Cost function optimizes the regression coefficients or weights. It measures how a linear regression model is performing.
* We can use the cost function to find the accuracy of the mapping function, which maps the input variable to the output variable. This mapping function is also known as Hypothesis function.

For Linear Regression, we use the Mean Squared Error (MSE) cost function, which is the average of squared error occurred between the predicted values and actual values. It can be written as:

For the above linear equation, MSE can be calculated as:

Linear Regression in Machine Learning

**Where,**

N=Total number of observation  
Yi = Actual value  
(a1xi+a0)= Predicted value.

**Residuals:** The distance between the actual value and predicted values is called residual. If the observed points are far from the regression line, then the residual will be high, and so cost function will high. If the scatter points are close to the regression line, then the residual will be small and hence the cost function.

### **Gradient Descent:**

* Gradient descent is used to minimize the MSE by calculating the gradient of the cost function.
* A regression model uses gradient descent to update the coefficients of the line by reducing the cost function.
* It is done by a random selection of values of coefficient and then iteratively update the values to reach the minimum cost function.

**Model Performance:**

The Goodness of fit determines how the line of regression fits the set of observations. The process of finding the best model out of various models is called optimization. It can be achieved by below method:

1. R-squared method:

* R-squared is a statistical method that determines the goodness of fit.
* It measures the strength of the relationship between the dependent and independent variables on a scale of 0-100%.
* The high value of R-square determines the less difference between the predicted values and actual values and hence represents a good model.
* It is also called a coefficient of determination, or coefficient of multiple determination for multiple regression.
* It can be calculated from the below formula:

Linear Regression in Machine Learning

Some popular applications of linear regression are:

* Analyzing trends and sales estimates
* Salary forecasting
* Real estate prediction
* Arriving at ETAs in traffic.

**Non-Linear Regression:-**

Nonlinear regression is a form of regression analysis in which data is fit to a model and then expressed as a mathematical function. Simple linear [regression](https://www.investopedia.com/terms/r/regression.asp) relates two variables (X and Y) with a straight line (y = mx + b), while nonlinear regression relates the two variables in a nonlinear (curved) relationship.1

The goal of the model is to make the [sum of the squares](https://www.investopedia.com/terms/s/sum-of-squares.asp) as small as possible.  The sum of squares is a measure that tracks how far the Y observations vary from the nonlinear (curved) function that is used to predict Y.

It is computed by first finding the difference between the fitted nonlinear function and every Y point of data in the set. Then, each of those differences is squared. Lastly, all of the squared figures are added together. The smaller the sum of these squared figures, the better the function fits the data points in the set. Nonlinear regression uses logarithmic functions, trigonometric functions, exponential functions, power functions, Lorenz curves, Gaussian functions, and other fitting methods.

* Both linear and nonlinear regression predict Y responses from an X variable (or variables).
* Nonlinear regression is a curved function of an X variable (or variables) that is used to predict a Y variable
* Nonlinear regression can show a prediction of population growth over time.

# **Logistic Regression in Machine Learning:-**

* Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.
* Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.
* Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems.
* In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).
* The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.
* Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.
* Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification. The below image is showing the logistic function:



**Logistic Function (Sigmoid Function):-**

* The sigmoid function is a mathematical function used to map the predicted values to probabilities.
* It maps any real value into another value within a range of 0 and 1.
* The value of the logistic regression must be between 0 and 1, which cannot go beyond this limit, so it forms a curve like the "S" form. The S-form curve is called the Sigmoid function or the logistic function.
* In logistic regression, we use the concept of the threshold value, which defines the probability of either 0 or 1. Such as values above the threshold value tends to 1, and a value below the threshold values tends to 0.

## Assumptions for Logistic Regression:

* The dependent variable must be categorical in nature.
* The independent variable should not have multi-collinearity.

## Logistic Regression Equation:

The Logistic regression equation can be obtained from the Linear Regression equation. The mathematical steps to get Logistic Regression equations are given below:

* We know the equation of the straight line can be written as:

Logistic Regression in Machine Learning

* In Logistic Regression y can be between 0 and 1 only, so for this let's divide the above equation by (1-y):

Logistic Regression in Machine Learning

* But we need range between -[infinity] to +[infinity], then take logarithm of the equation it will become:

Logistic Regression in Machine Learning

The above equation is the final equation for Logistic Regression.

## Type of Logistic Regression:

On the basis of the categories, Logistic Regression can be classified into three types:

* Binomial: In binomial Logistic regression, there can be only two possible types of dependent variables, such as 0 or 1, Pass or Fail, etc.
* Multinomial: In multinomial Logistic regression, there can be 3 or more possible unordered types of the dependent variable, such as "cat", "dogs", or "sheep"
* Ordinal: In ordinal Logistic regression, there can be 3 or more possible ordered types of dependent variables, such as "low", "Medium", or "High".

**Non-Linear Regression:-**

* Nonlinear regression is a form of regression analysis in which data is fit to a model and then expressed as a mathematical function.
* Simple linear [regression](https://www.investopedia.com/terms/r/regression.asp) relates two variables (X and Y) with a straight line (y = mx + b), while nonlinear regression relates the two variables in a nonlinear (curved) relationship.1

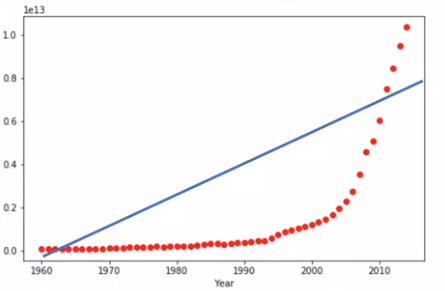
The goal of the model is to make the [sum of the squares](https://www.investopedia.com/terms/s/sum-of-squares.asp) as small as possible.  The sum of squares is a measure that tracks how far the Y observations vary from the nonlinear (curved) function that is used to predict Y.

It is computed by first finding the difference between the fitted nonlinear function and every Y point of data in the set. Then, each of those differences is squared. Lastly, all of the squared figures are added together. The smaller the sum of these squared figures, the better the function fits the data points in the set. Nonlinear regression uses logarithmic functions, trigonometric functions, exponential functions, power functions, Lorenz curves, Gaussian functions, and other fitting methods.

* Both linear and nonlinear regression predict Y responses from an X variable (or variables).
* Nonlinear regression is a curved function of an X variable (or variables) that is used to predict a Y variable
* Nonlinear regression can show a prediction of population growth over time.

### **KEY TAKEAWAYS**

* Both linear and nonlinear regression predict Y responses from an X variable (or variables).
* Nonlinear regression is a curved function of an X variable (or variables) that is used to predict a Y variable
* Nonlinear regression can show a prediction of population growth over time.



**K -Nearest Neighbour:**

* K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
* K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
* K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
* K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data
* **Example:** Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.
* Suppose there are two categories, i.e., Category A and Category B, and we have a new data point x1, so this data point will lie in which of these categories. To solve this type of problem, we need a K-NN algorithm. With the help of K-NN, we can easily identify the category or class of a particular dataset. Consider the below diagram:



How does K-NN work?

The K-NN working can be explained on the basis of the below algorithm:

* **Step-1:** Select the number K of the neighbors
* **Step-2:** Calculate the Euclidean distance of **K number of neighbors**
* **Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.
* **Step-4:** Among these k neighbors, count the number of the data points in each category.
* **Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum.
* **Step-6:** Our model is ready.

Suppose we have a new data point and we need to put it in the required category. Consider the below image:



* Firstly, we will choose the number of neighbors, so we will choose the k=5.
* Next, we will calculate the **Euclidean distance** between the data points. The Euclidean distance is the distance between two points, which we have already studied in geometry. It can be calculated as:
* 
* By calculating the Euclidean distance we got the nearest neighbors, as three nearest neighbors in category A and two nearest neighbors in category B. Consider the below image:



* As we can see the 3 nearest neighbors are from category A, hence this new data point must belong to category A.

# **Decision Tree Classification Algorithm:**

* Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is 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 a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.
* It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.
* It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.
* 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 based on the answer (Yes/No), it further split the tree into subtrees.
* Below diagram explains the general structure of a decision tree:

#### Note: A decision tree can contain categorical data (YES/NO) as well as numeric data.

* 

**Example:** Suppose there is a candidate who has a job offer and wants to decide whether he should accept the offer or Not. So, to solve this problem, the decision tree starts with the root node (Salary attribute by ASM). The root node splits further into the next decision node (distance from the office) and one leaf node based on the corresponding labels. The next decision node further gets split into one decision node (Cab facility) and one leaf node. Finally, the decision node splits into two leaf nodes (Accepted offers and Declined offer). Consider the below diagram

* 

## Attribute Selection Measures

While implementing a Decision tree, the main issue arises that how to select the best attribute for the root node and for sub-nodes. So, to solve such problems there is a technique which is called as **Attribute selection measure or ASM.**By this measurement, we can easily select the best attribute for the nodes of the tree. There are two popular techniques for ASM, which are:

* **Information Gain**
* **Gini Index**

### **1. Information Gain:**

* Information gain is the measurement of changes in entropy after the segmentation of a dataset based on an 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:

1. Information Gain= Entropy(S)- [(Weighted Avg) \*Entropy(each feature)

**Entropy:** Entropy is a metric to measure the impurity in a given attribute. It specifies randomness in data. Entropy can be calculated as:

Entropy(s)= -P(yes)log2 P(yes)- P(no) log2 P(no)

**Where,**

* **S= Total number of samples**
* **P(yes)= probability of yes**
* **P(no)= probability of no**

### **2. 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 can be calculated using the below formula:

Gini Index= 1- ∑jPj2

## Advantages of the Decision Tree

* It is simple to understand as it follows the same process which a human follow 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.
* There is less requirement of data cleaning compared to other algorithms.

## Disadvantages of the Decision Tree

* The decision tree contains lots of layers, which makes it complex.
* 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.

# **Random Forest Algorithm:**

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of **ensemble learning,** which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, ***"Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset."*** Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

**The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.**



.

## Why use Random Forest?

* It takes less training time as compared to other algorithms.
* It predicts output with high accuracy, even for the large dataset it runs efficiently.
* It can also maintain accuracy when a large proportion of data is missing.

## How does Random Forest algorithm work?

Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase.

The Working process can be explained in the below steps and diagram:

**Step-1:** Select random K data points from the training set.

**Step-2:** Build the decision trees associated with the selected data points (Subsets).

**Step-3:** Choose the number N for decision trees that you want to build.

**Step-4:** Repeat Step 1 & 2.

**Step-5:** For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

The working of the algorithm can be better understood by the below example:

**Example:** Suppose there is a dataset that contains multiple fruit images. So, this dataset is given to the Random forest classifier. The dataset is divided into subsets and given to each decision tree. During the training phase, each decision tree produces a prediction result, and when a new data point occurs, then based on the majority of results, the Random Forest classifier predicts the final decision. Consider the below image:



## Applications of Random Forest

There are mainly four sectors where Random forest mostly used:

1. **Banking:** Banking sector mostly uses this algorithm for the identification of loan risk.
2. **Medicine:** With the help of this algorithm, disease trends and risks of the disease can be identified.
3. **Land Use:** We can identify the areas of similar land use by this algorithm.
4. **Marketing:** Marketing trends can be identified using this algorithm.

## Advantages of Random Forest

* Random Forest is capable of performing both Classification and Regression tasks.
* It is capable of handling large datasets with high dimensionality.
* It enhances the accuracy of the model and prevents the overfitting issue.

## Disadvantages of Random Forest

* Although random forest can be used for both classification and regression tasks, it is not more suitable for Regression tasks.

# **Naïve Bayes Classifier Algorithm**

* Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems.
* It is mainly used in *text classification* that includes a high-dimensional training dataset.
* It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.
* Some popular examples of Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.

## Bayes' Theorem:

* Bayes' theorem is also known as Bayes' Rule or Bayes' law, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.
* The formula for Bayes' theorem is given as:
* Naïve Bayes Classifier Algorithm
* Where,
* P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B.
* P(B|A) is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.
* **P(A) is Prior Probability**: Probability of hypothesis before observing the evidence.
* **P(B) is Marginal Probability**: Probability of Evidence.

### **Advantages of Naïve Bayes Classifier:**

* Naïve Bayes is one of the fast and easy ML algorithms to predict a class of datasets.
* It can be used for Binary as well as Multi-class Classifications.
* It performs well in Multi-class predictions as compared to the other Algorithms.
* It is the most popular choice for **text classification problems**.

### **Disadvantages of Naïve Bayes Classifier:**

* Naive Bayes assumes that all features are independent or unrelated, so it cannot learn the relationship between features.

### **Applications of Naïve Bayes Classifier:**

* It is used for **Credit Scoring**.
* It is used in **medical data classification**.
* It can be used in **real-time predictions** because Naïve Bayes Classifier is an eager learner.
* It is used in Text classification such as **Spam filtering** and **Sentiment analysis**.

# **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:

**Example:** SVM can be understood with the example that we have used in the KNN classifier. Suppose we see a strange cat that also has some features of dogs, so if we want a model that can accurately identify whether it is a cat or dog, so such a model can be created by using the SVM algorithm. We will first train our model with lots of images of cats and dogs so that it can learn about different features of cats and dogs, and then we test it with this strange creature. So as support vector creates a decision boundary between these two data (cat and dog) and choose extreme cases (support vectors), it will see the extreme case of cat and dog. On the basis of the support vectors, it will classify it as a cat. Consider the below diagram:

**Example:** SVM can be understood with the example that we have used in the KNN classifier. Suppose we see a strange cat that also has some features of dogs, so if we want a model that can accurately identify whether it is a cat or dog, so such a model can be created by using the SVM algorithm. We will first train our model with lots of images of cats and dogs so that it can learn about different features of cats and dogs, and then we test it with this strange creature. So as support vector creates a decision boundary between these two data (cat and dog) and choose extreme cases (support vectors), it will see the extreme case of cat and dog. On the basis of the support vectors, it will classify it as a cat. Consider the below diagram:



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

## Types of SVM

**SVM can be of two types:**

* **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.

## How does SVM works?

**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**.



### **Kernel SVM:**

[Kernel support vector machines](https://www.geeksforgeeks.org/major-kernel-functions-in-support-vector-machine-svm/)(SVMs) are a variant of support vector machines (SVMs) that use kernel functions to find the maximum-margin hyperplane in non-linear classification or regression problems. In simple terms, a kernel function transforms the original data into a higher-dimensional space, where it becomes linearly separable. The maximum-margin hyperplane is then found in this higher-dimensional space using an SVM.

Kernel SVMs have several advantages over regular SVMs. They can handle non-linear classification or regression tasks without having to explicitly perform the data transformation, which can be computationally expensive. They also allow the use of different kernel functions, which can provide different types of non-linear transformations and can be chosen based on the specific characteristics of the data.

The most commonly used kernel functions in kernel SVMs are the linear, polynomial, and radial basis function (RBF) kernels. The linear kernel is used for linear classification or regression tasks, the polynomial kernel can handle non-linear problems, and the [RBF](https://www.geeksforgeeks.org/radial-basis-function-kernel-machine-learning/) kernel is often used in classification tasks with a large number of features.

Kernel SVMs have been widely used in many applications, such as image and text classification, protein classification, and regression problems. However, they can be computationally expensive and may not be suitable for very large datasets. Additionally, choosing the right kernel function and the corresponding hyperparameters can be challenging and requires some domain knowledge and experimentation.

## K-Means Algorithm

K-Means Clustering is an [Unsupervised Learning algorithm](https://www.javatpoint.com/unsupervised-machine-learning), which groups the unlabeled dataset into different clusters. Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on.

It is an iterative algorithm that divides the unlabeled dataset into k different clusters in such a way that each dataset belongs only one group that has similar properties

It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training.

It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters

The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.

The k-means [clustering](https://www.javatpoint.com/clustering-in-machine-learning) algorithm mainly performs two tasks:

* Determines the best value for K center points or centroids by an iterative process.
* Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.



## How does the K-Means Algorithm Work?

The working of the K-Means algorithm is explained in the below steps:

**Step-1:** Select the number K to decide the number of clusters.

**Step-2:** Select random K points or centroids. (It can be other from the input dataset).

**Step-3:** Assign each data point to their closest centroid, which will form the predefined K clusters.

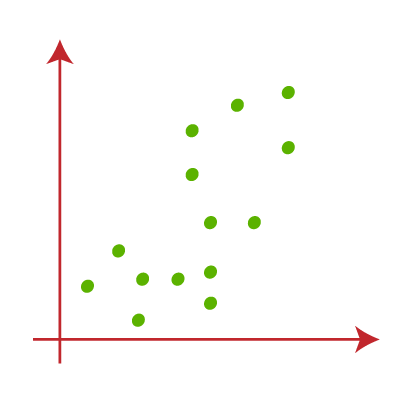
**Step-4:** Calculate the variance and place a new centroid of each cluster.

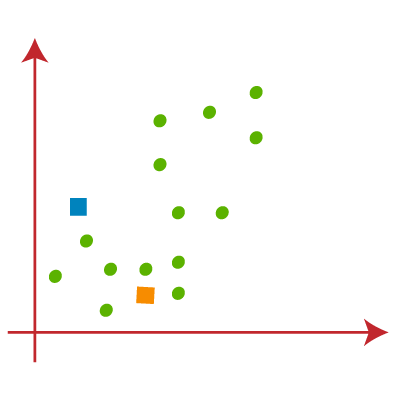
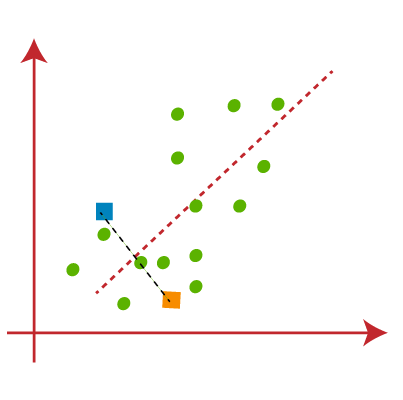
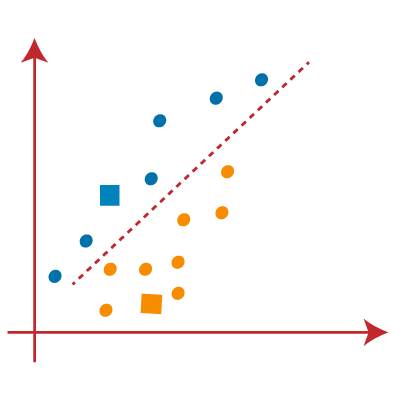
**Step-5:** Repeat the third steps, which means reassign each datapoint to the new closest centroid of each cluster.

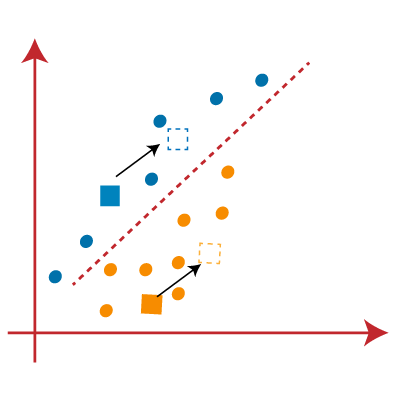
**Step-6:** If any reassignment occurs, then go to step-4 else go to FINISH.

**Step-7**: The model is ready.

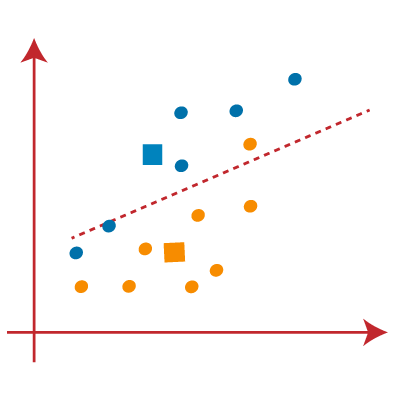
Suppose we have two variables M1 and M2. The x-y axis scatter plot of these two variables is given below:



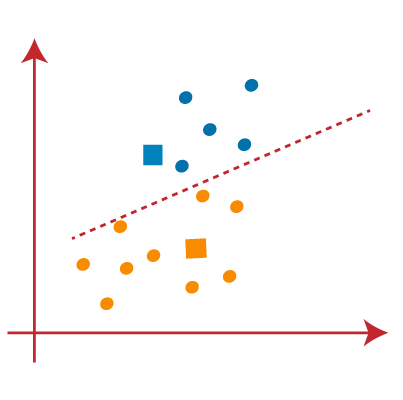
* Let's take number k of clusters, i.e., K=2, to identify the dataset and to put them into different clusters. It means here we will try to group these datasets into two different clusters.
* We need to choose some random k points or centroid to form the cluster. These points can be either the points from the dataset or any other point. So, here we are selecting the below two points as k points, which are not the part of our dataset. Consider the below image:  
  
* Now we will assign each data point of the scatter plot to its closest K-point or centroid. We will compute it by applying some mathematics that we have studied to calculate the distance between two points. So, we will draw a median between both the centroids. Consider the below image:  
  
* From the above image, it is clear that points left side of the line is near to the K1 or blue centroid, and points to the right of the line are close to the yellow centroid. Let's color them as blue and yellow for clear visualization.
* 

As we need to find the closest cluster, so we will repeat the process by choosing **a new centroid**. To choose the new centroids, we will compute the center of gravity of these centroids, and will find new centroids as below:  


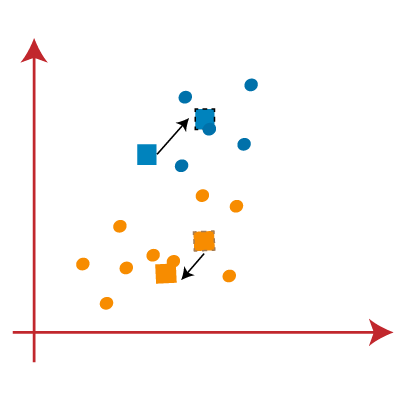
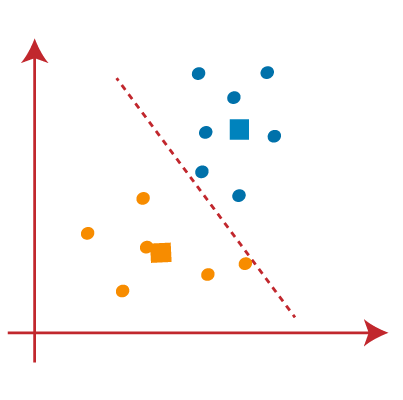
Next, we will reassign each datapoint to the new centroid. For this, we will repeat the same process of finding a median line. The median will be like below image:

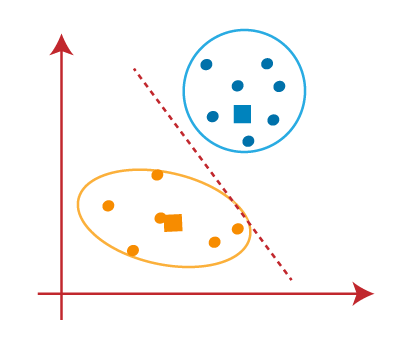


From the above image, we can see, one yellow point is on the left side of the line, and two blue points are right to the line. So, these three points will be assigned to new centroids.

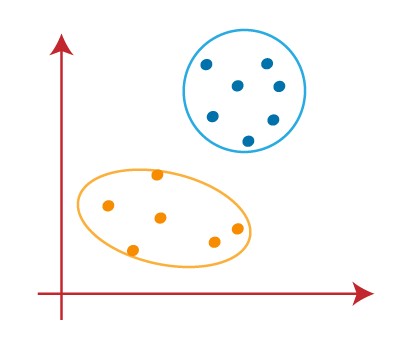


As reassignment has taken place, so we will again go to the step-4, which is finding new centroids or K-points.

* We will repeat the process by finding the center of gravity of centroids, so the new centroids will be as shown in the below image:  
  
* As we got the new centroids so again will draw the median line and reassign the data points. So, the image will be:  
  

We can see in the above image; there are no dissimilar data points on either side of the line, which means our model is formed. Consider the below image:  


As our model is ready, so we can now remove the assumed centroids, and the two final clusters will be as shown in the below image:



## How to choose the value of "K number of clusters" in K-means Clustering?

The performance of the K-means clustering algorithm depends upon highly efficient clusters that it forms. But choosing the optimal number of clusters is a big task. There are some different ways to find the optimal number of clusters, but here we are discussing the most appropriate method to find the number of clusters or value of K. The method is given below:

### **Elbow Method**

The Elbow method is one of the most popular ways to find the optimal number of clusters. This method uses the concept of WCSS value. **WCSS** stands for **Within Cluster Sum of Squares**, which defines the total variations within a cluster. The formula to calculate the value of WCSS (for 3 clusters) is given below:

WCSS= ∑Pi in Cluster1 distance(Pi C1)2 +∑Pi in Cluster2distance(Pi C2)2+∑Pi in CLuster3 distance(Pi C3)2

In the above formula of WCSS,

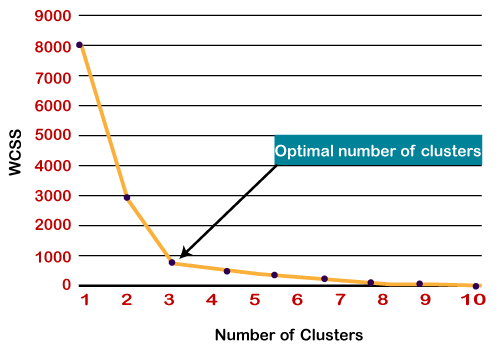
∑Pi in Cluster1 distance(Pi C1)2: It is the sum of the square of the distances between each data point and its centroid within a cluster1 and the same for the other two terms.

To measure the distance between data points and centroid, we can use any method such as Euclidean distance or Manhattan distance.

To find the optimal value of clusters, the elbow method follows the below steps:

* It executes the K-means clustering on a given dataset for different K values (ranges from 1-10).
* For each value of K, calculates the WCSS value.
* Plots a curve between calculated WCSS values and the number of clusters K.
* The sharp point of bend or a point of the plot looks like an arm, then that point is considered as the best value of K.

Since the graph shows the sharp bend, which looks like an elbow, hence it is known as the elbow method. The graph for the elbow method looks like the below image:



# **XGBoost**

Gradient boosted decision trees are implemented by the XGBoost library of Python, intended for speed and execution, which is the most important aspect of ML (machine learning).

**XgBoost**: XgBoost (Extreme Gradient Boosting) library of Python was introduced at the University of Washington by scholars. It is a module of Python written in C++, which helps ML model algorithms by the training for Gradient Boosting.

**Gradient boosting:** This is an AI method utilized in classification and regression assignments, among others. It gives an expectation model as a troupe of feeble forecast models, commonly called decision trees.

Gradient boosted decision trees are implemented by the XGBoost library of Python, intended for speed and execution, which is the most important aspect of ML (machine learning).

**XgBoost**: XgBoost (Extreme Gradient Boosting) library of Python was introduced at the University of Washington by scholars. It is a module of Python written in C++, which helps ML model algorithms by the training for Gradient Boosting.

**Gradient boosting:** This is an AI method utilized in classification and regression assignments, among others. It gives an expectation model as a troupe of feeble forecast models, commonly called decision trees.

How does Fundamental Gradient Boosting function?

* A loss function should be improved, which implies bringing down the loss function better than the result.
* To make expectations, weak learners are used in the model
* Decision trees are utilized in this, and they are utilized in a jealous way, which alludes to picking the best-divided focuses in light of Gini Impurity and so forth or to limit the loss function
* The additive model is utilized to gather every one of the frail models, limiting the loss function.
* Trees are added each, ensuring existing trees are not changed in the decision tree. Regularly angle plummet process is utilized to find the best hyper boundaries, post which loads are refreshed further.

XGBoost can give improved arrangements than other ML model algorithms. As a matter of fact, since its initiation, it has turned into the "best in class" ML model algorithm to manage organized information.

What Makes XGBoost So Famous?

* **Execution and Speed:** Originally built on C++, it is similarly fast to other gathering classifiers.
* **Center calculation is parallelizable**: it can outfit the force of multi-center PCs because the center XGBoost calculation is parallelizable. Moreover, it is parallelizable onto GPUs and across organizations of PCs, making it attainable to prepare on a huge dataset.
* **Reliably outflanks other technique calculations:** It has shown better output on many AI benchmark datasets.
* **Wide assortment of tuning boundaries**: XGBoost inside has boundaries for scikit-learn viable API, missing qualities, regularization, cross-approval, client characterized objective capacities, tree boundaries, etc.

XGBoost (Extreme Gradient Boosting) has a place with a group of helping calculations and utilizations of the slope supporting (GBM) structure at its center.

# **Principal Component Analysis**

Principal Component Analysis is an unsupervised learning algorithm that is used for the dimensionality reduction in [machine learning](https://www.javatpoint.com/machine-learning). It is a statistical process that converts the observations of correlated features into a set of linearly uncorrelated features with the help of orthogonal transformation. These new transformed features are called the **Principal Components**. It is one of the popular tools that is used for exploratory data analysis and predictive modeling. It is a technique to draw strong patterns from the given dataset by reducing the variances.

PCA generally tries to find the lower-dimensional surface to project the high-dimensional data.

PCA works by considering the variance of each attribute because the high attribute shows the good split between the classes, and hence it reduces the dimensionality. Some real-world applications of PCA are **image processing, movie recommendation system, optimizing the power allocation in various communication channels.** It is a feature extraction technique, so it contains the important variables and drops the least important variable.

The PCA algorithm is based on some mathematical concepts such as

* Variance and Covariance
* Eigenvalues and Eigen factors

Some common terms used in PCA algorithm:

* **Dimensionality:** It is the number of features or variables present in the given dataset. More easily, it is the number of columns present in the dataset.
* **Correlation:** It signifies that how strongly two variables are related to each other. Such as if one changes, the other variable also gets changed. The correlation value ranges from -1 to +1. Here, -1 occurs if variables are inversely proportional to each other, and +1 indicates that variables are directly proportional to each other.
* **Orthogonal:** It defines that variables are not correlated to each other, and hence the correlation between the pair of variables is zero.
* **Eigenvectors:** If there is a square matrix M, and a non-zero vector v is given. Then v will be eigenvector if Av is the scalar multiple of v.
* **Covariance Matrix:** A matrix containing the covariance between the pair of variables is called the Covariance Matrix.

### **Principal Components in PCA**

As described above, the transformed new features or the output of PCA are the Principal Components. The number of these PCs are either equal to or less than the original features present in the dataset. Some properties of these principal components are given below:

* The principal component must be the linear combination of the original features.
* These components are orthogonal, i.e., the correlation between a pair of variables is zero.
* The importance of each component decreases when going to 1 to n, it means the 1 PC has the most importance, and n PC will have the least importance

### **Steps for PCA algorithm**

1. **Getting the dataset**  
   Firstly, we need to take the input dataset and divide it into two subparts X and Y, where X is the training set, and Y is the validation set.
2. **Representing data into a structure**  
   Now we will represent our dataset into a structure. Such as we will represent the two-dimensional matrix of independent variable X. Here each row corresponds to the data items, and the column corresponds to the Features. The number of columns is the dimensions of the dataset.
3. **Standardizing the data**  
   In this step, we will standardize our dataset. Such as in a particular column, the features with high variance are more important compared to the features with lower variance.  
   If the importance of features is independent of the variance of the feature, then we will divide each data item in a column with the standard deviation of the column. Here we will name the matrix as Z.
4. **Calculating the Covariance of Z**  
   To calculate the covariance of Z, we will take the matrix Z, and will transpose it. After transpose, we will multiply it by Z. The output matrix will be the Covariance matrix of Z.
5. **Calculating the Eigen Values and Eigen Vectors**  
   Now we need to calculate the eigenvalues and eigenvectors for the resultant covariance matrix Z. Eigenvectors or the covariance matrix are the directions of the axes with high information. And the coefficients of these eigenvectors are defined as the eigenvalues.
6. **Sorting the Eigen Vectors**  
   In this step, we will take all the eigenvalues and will sort them in decreasing order, which means from largest to smallest. And simultaneously sort the eigenvectors accordingly in matrix P of eigenvalues. The resultant matrix will be named as P\*.
7. **Calculating the new features Or Principal Components**  
   Here we will calculate the new features. To do this, we will multiply the P\* matrix to the Z. In the resultant matrix Z\*, each observation is the linear combination of original features. Each column of the Z\* matrix is independent of each other.
8. **Remove less or unimportant features from the new dataset.**  
   The new feature set has occurred, so we will decide here what to keep and what to remove. It means, we will only keep the relevant or important features in the new dataset, and unimportant features will be removed out.

## Applications of Principal Component Analysis

* PCA is mainly used as the dimensionality reduction technique in various AI applications such **as computer vision, image compression, etc.**
* It can also be used for finding hidden patterns if data has high dimensions. Some fields where PCA is used are Finance, data mining, Psychology, etc.