**SUPERVISED LEARNING**

**Supervised Learning:** Supervised learning is a type of machine learning method in which we provide sample labeled data to the machine learning system in order to train it, and on that basis, it predicts the output.

The system creates a model using labeled data to understand the datasets and learn about each data, once the training and processing are done then we test the model by providing a sample data to check whether it is predicting the exact output or not.

The goal of supervised learning is to map input data with the output data. The supervised learning is based on supervision, and it is the same as when a student learns things in the supervision of the teacher. The example of supervised learning is spam filtering.Supervised learning can be grouped further in two categories of algorithms:

* 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
  + Linear Regression
  + Regression Trees
  + Non-Linear Regression
  + Bayesian Linear Regression
  + Polynomial 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.
  + Random Forest
  + Decision Trees
  + Logistic Regression
  + Support vector Machines

Advantages of Supervised Learning:

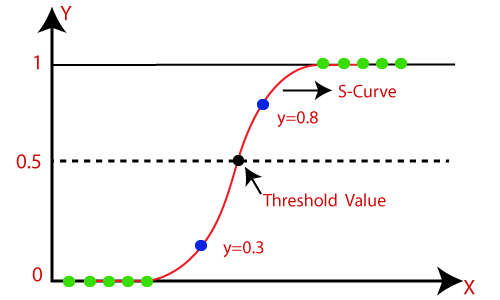
* With the help of supervised learning, the model can predict the output on the basis of prior experiences
* In supervised learning, we can have an exact idea about the classes of objects
* Supervised learning model helps us to solve various real-world problems such as fraud detection, spam filtering, etc

Disadvantages of Supervised Learning:

* Supervised learning models are not suitable for handling the complex tasks
* Supervised learning cannot predict the correct output if the test data is different from the training dataset
* Training required lots of computation times
* In supervised learning, we need enough knowledge about the classes of object

# LOGISTIC REGRESSION

* 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).



# 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.

**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:



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



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



# Type of Logistic Regression

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

# DECISION TREES

* 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.

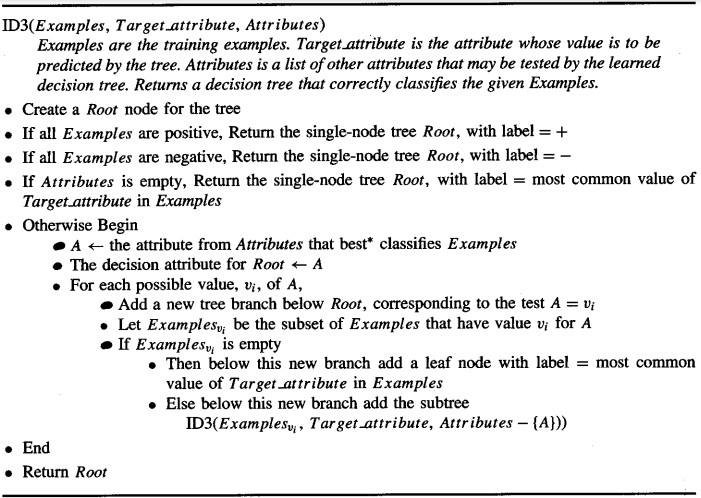
**Decision Tree Terminologies:**

* 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 subnodes according to the given conditions.
* Branch/Sub Tree: 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.

**Decision Tree Representation:**

* Decision trees classify instances by sorting them down the tree from the root to some leaf node, which provides the classification of the instance.
* Each node in the tree specifies a test of some attribute of the instance, and each branch descending from that node corresponds to one of the possible values for this attribute.
* An instance is classified by starting at the root node of the tree, testing the attribute specified by this node, then moving down the tree branch corresponding to the value of the attribute.
* This process is then repeated for the subtree rooted at the new node.
* In general, decision trees represent a disjunction of conjunctions of constraints on the attribute values of instances.
* Each path from the tree root to a leaf corresponds to a conjunction of attribute tests, and the tree itself to a disjunction of these conjunctions.

**Decision tree based ID3 Algorithm:**



**Entropy:** Entropy measures the impurity of a collection of examples.



Where, p+ is the proportion of positive examples in S p– is the proportion of negative examples in S

**Information Gain:** Information Gain is the expected reduction in entropy caused by partitioning the examples according to this attribute. The information gain, Gain(S, A) of an attribute A, relative to a collection of examples S, is defined as



Where, Values (A) is the set of all possible values for attribute A

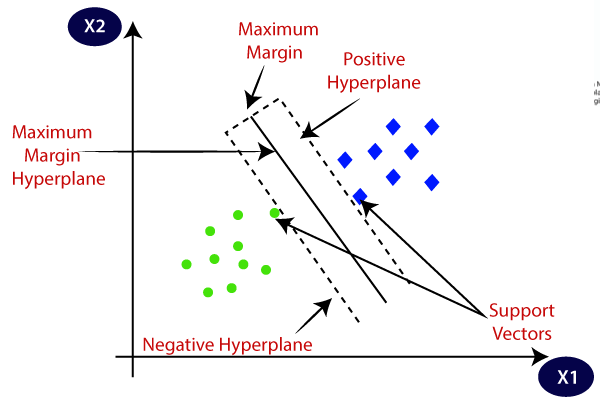
SV is the subset of S for which attribute A has value v

# SUPPORT VECTOR MACHINES

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. SVM algorithm can be used for Face detection, image classification, text categorization, etc. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:



**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.

**Hyperplaneand Support Vectors in the SVM Algorithm:**

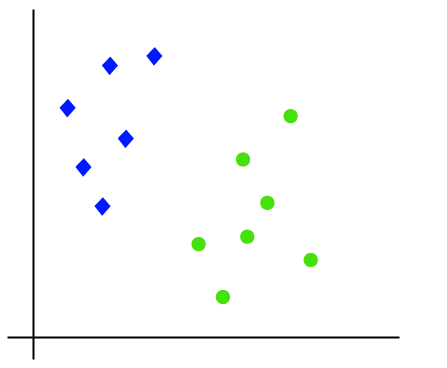
* Hyperplane: There can be multiple lines/decision boundaries to segregate the classes in n-dimensional space, but we need to find out the best decision boundary that helps to classify the data points. This best boundary is known as the hyperplane of SVM.

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.

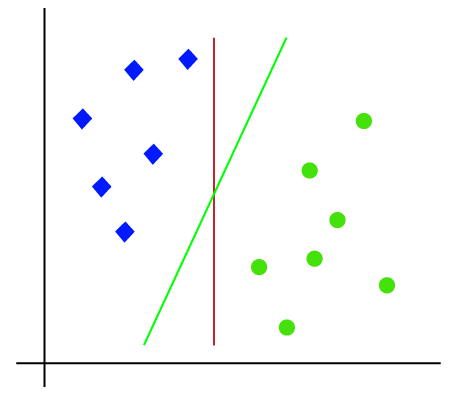
* Support Vectors: The data points or vectors that are the closest to the hyperplane and which affect the position of the hyperplane are termed as Support Vector. Since these vectors support the hyperplane, hence called a Support vector.

**Working of SVM:**

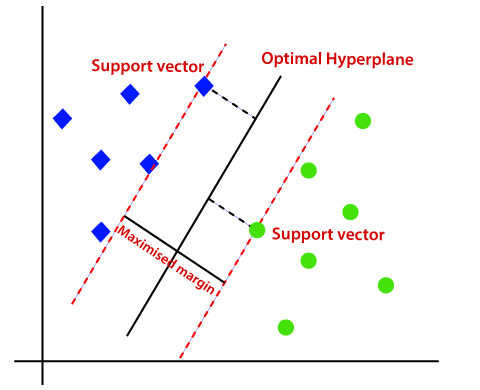
Linear SVM: The working of the SVM algorithm can be understood by using an example. Suppose we have a dataset that has two tags (green and blue), and the dataset has two features x1 and x2. We want a classifier that can classify the pair(x1, x2) of coordinates in either green or blue. Consider the below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



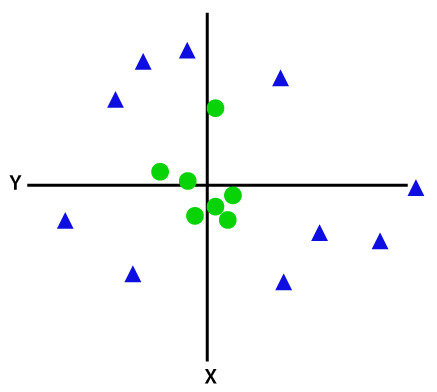
So as it is 2-d space so by just using a straight line, we can easily separate these two classes. But there can be multiple lines that can separate these classes. Consider the below image:



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:

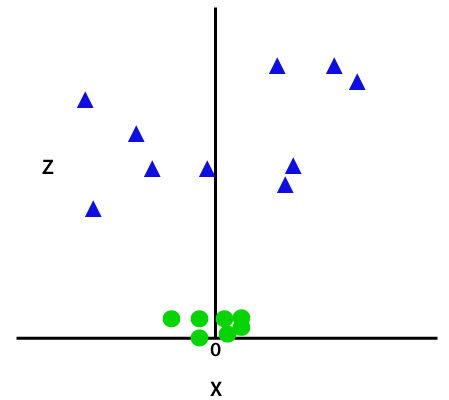


So to separate these data points, we need to add one more dimension. For linear data, we have used two dimensions x and y, so for non-linear data, we will add a third dimension z.

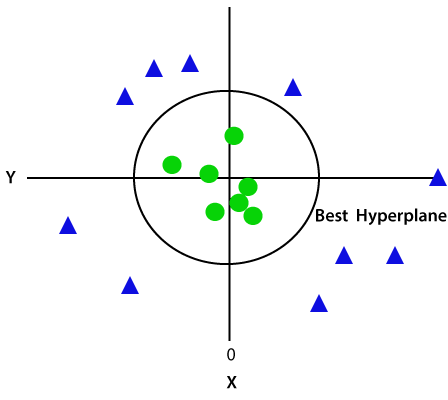
It can be calculated as:

# Z =x2 +y2

By adding the third dimension, the sample space will become as below image:



Since we are in 3-d Space, hence it is looking like a plane parallel to the x-axis. If we convert it in 2d space with z=1, then it will become as:



Hence we get a circumference of radius 1 in case of non-linear data.

# NAIVE BAYES

* 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.
* Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which help in building the fast machine learning models that can make quick predictions.
* 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.

The Naïve Bayes algorithm is comprised of two words Naïve and Bayes:

* Naïve: It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the bases of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each feature individually contributes to identify that it is an apple without depending on each other.
* Bayes: It is called Bayes because it depends on the principle of Bayes' Theorem.

**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:



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

**Working of Naïve Bayes' Classifier:** Consider a dataset of weather conditions and corresponding target variable "Play". So using this dataset we need to decide that whether we should play or not on a particular day according to the weather conditions. So, to solve this, we need to follow the below steps:

* Convert the given dataset into frequency tables
* Generate Likelihood table by finding the probabilities of given features • Now, use Bayes theorem to calculate the posterior probability

**UNSUPERVISED LEARNING**

**Unsupervised Learning:** Unsupervised learning is a learning method in which a machine learns without any supervision.

The training is provided to the machine with the set of data that has not been labeled, classified, or categorized, and the algorithm needs to act on that data without any supervision. The goal of unsupervised learning is to restructure the input data into new features or a group of objects with similar patterns.

In unsupervised learning, we don't have a predetermined result. The machine tries to find useful insights from the huge amount of data. It can be further classifieds into two categories of algorithms:

* 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.
  + K-Means Clustering algorithm
  + Mean-shift algorithm
  + DBSCAN Algorithm
  + Principal Component Analysis
  + Independent Component Analysis

* 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.
  + Apriori algorithm
  + FP-growth algorithm

Advantages of Unsupervised Learning:

* Unsupervised learning is used for more complex tasks as compared to supervised learning because, in unsupervised learning, we don't have labeled input data.
* Unsupervised learning is preferable as it is easy to get unlabeled data in comparison to labeled data.

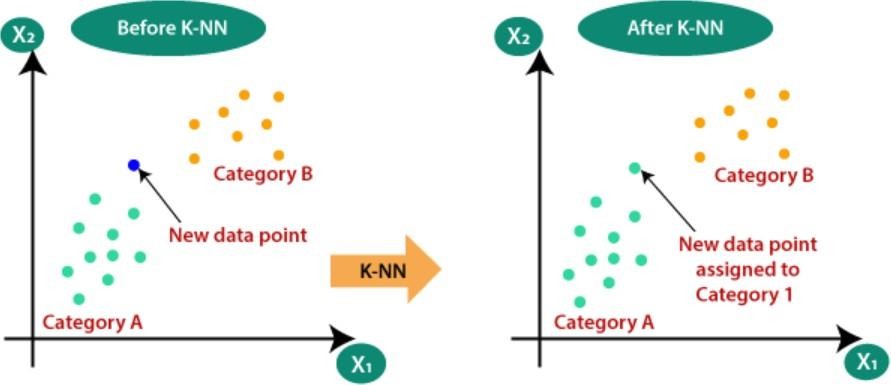
Disadvantages of Unsupervised Learning:

* Unsupervised learning is intrinsically more difficult than supervised learning as it does not have corresponding output.
* The result of the unsupervised learning algorithm might be less accurate as input data is not labeled, and algorithms do not know the exact output in advance.

# K-NEAREST-NEIGHBOR ALGORITHM

* 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 stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
* 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.
* It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
* KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

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:



**K-NN Algorithm:**

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

**How to select the value of K in the K-NN Algorithm?**

* There is no particular way to determine the best value for "K", so we need to try some values to find the best out of them. The most preferred value for K is 5.
* A very low value for K such as K=1 or K=2, can be noisy and lead to the effects of outliers in the model.
* Large values for K are good, but it may find some difficulties.

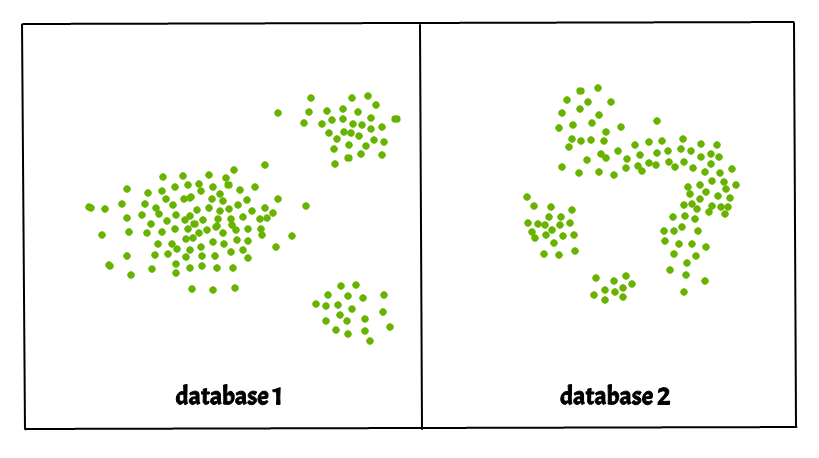
Advantages:

* It is simple to implement.
* It is robust to the noisy training data
* It can be more effective if the training data is large.

Disadvantages:

* Always needs to determine the value of K which may be complex some time.
* The computation cost is high because of calculating the distance between the data points for all the training samples.

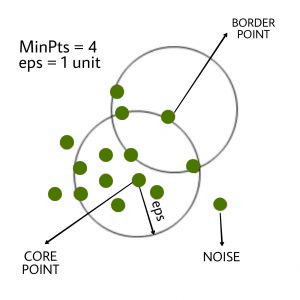
**Density-Based Spatial Clustering Of Applications With Noise (DBSCAN)**

* Clusters are dense regions in the data space, separated by regions of the lower density of points. The DBSCAN algorithm is based on this intuitive notion of “clusters” and “noise”. The key idea is that for each point of a cluster, the neighbourhood of a given radius has to contain at least a minimum number of points.

**Parameters Required For DBSCAN Algorithm**

**1.**eps: It defines the neighborhood around a data point i.e. if the distance between two points is lower or equal to ‘eps’ then they are considered neighbors. If the eps value is chosen too small then a large part of the data will be considered as an outlier. If it is chosen very large then the clusters will merge and the majority of the data points will be in the same clusters. One way to find the eps value is based on the k-distance graph.

1. MinPts: Minimum number of neighbors (data points) within eps radius. The larger the dataset, the larger value of MinPts must be chosen. As a general rule, the minimum MinPts can be derived from the number of dimensions D in the dataset as, MinPts >= D+1. The minimum value of MinPts must be chosen at least 3.

* In this algorithm, we have 3 types of data points.  
  Core Point: A point is a core point if it has more than MinPts points within eps.   
  Border Point: A point which has fewer than MinPts within eps but it is in the neighborhood of a core point.   
  Noise or outlier: A point which is not a core point or border point.
* 
* Steps Used In DBSCAN Algorithm
* Find all the neighbor points within eps and identify the core points or visited with more than MinPts neighbors.
* For each core point if it is not already assigned to a cluster, create a new cluster.
* Find recursively all its density-connected points and assign them to the same cluster as the core point.   
  A point a and b are said to be density connected if there exists a point c which has a sufficient number of points in its neighbors and both points a and b are within the eps distance. This is a chaining process. So, if b is a neighbor of c, c is a neighbor of d, and d is a neighbor of e, which in turn is  neighbor of a implying that b is a neighbor of a.
* Iterate through the remaining unvisited points in the dataset. Those points that do not belong to any cluster are noise.

**CLUSTERING**

Clustering (or) cluster analysis is a machine learning technique, which groups the unlabelled dataset. It can be defined as "A way of grouping the data points into differentclusters, consisting of similar data points. The objects with the possible similarities remain in a group that has less or no similarities with another group."

It is an unsupervised learning method, hence no supervision is provided to the algorithm, and it deals with the unlabeled dataset. The clustering technique can be widely used in various tasks. Some most common uses of this technique are:

 Market Segmentation

 Statistical data analysis

 Social network analysis

 Image segmentation

 Anomaly detection, etc.

On the left, a graph of symptom severity vs. symptom count
   displaying datapoints that suggest three clusters.
   On the right, the same graph but with each of the three clusters colored.

**Types of Clustering Methods:**

**1. Partitioning Clustering:** It is a type of clustering that divides the data into non-

hierarchical groups. It is also known as the centroid-based method. The most common

examples of partitioning clustering are the K-Means Clustering algorithm, CLARANS

(Clustering Large Applications based upon Randomized Search), etc.

**2. Density-Based Clustering**: These methods consider the clusters as the dense region

having some similarities and differences from the lower dense region of the space.

These methods have good accuracy and the ability to merge two clusters. Examples

are DBSCAN (Density-Based Spatial Clustering of Applications with Noise),

OPTICS (Ordering Points to Identify Clustering Structure), etc.

**3. Distribution Model-Based Clustering:** In this, the data is divided based on the

probability of how a dataset belongs to a particular distribution. The grouping is done

by assuming some distributions commonly Gaussian Distribution. The example of

this type is the Expectation-Maximization Clustering algorithm that uses Gaussian

Mixture Models (GMM).

**4. Hierarchical Clustering:** Hierarchical clustering can be used as an alternative for the

partitioned clustering as there is no requirement of pre-specifying the number of

clusters to be created. In this technique, the dataset is divided into clusters to create a

tree-like structure, which is also called a dendrogram. Examples are Agglomerative

(bottom-up approach) & Divisive (top-down approach).

5. Fuzzy Clustering: Fuzzy clustering is a type of soft method in which a data object

may belong to more than one group or cluster. Each dataset has a set of membership

coefficients, which depend on the degree of membership to be in a cluster. Fuzzy C-

means algorithm is the example of this type of clustering; it is sometimes also known

as the Fuzzy k-means algorithm.

**K-MEANS CLUSTERING ALGORITHM**

K-Means Clustering is an Unsupervised Learning algorithm, which groups the

unlabeled dataset into different clusters “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 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

algorithm mainly performs two tasks:

1. Determines the best value for K center points or centroids by an iterative process.

2. Assigns each data point to its closest k-center. Those data points which are near to the

particular k-center, create a cluster.

**Working of K-Means Algorithm:**

1. Select the number K to decide the number of clusters.

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

3. Assign each data point to their closest centroid, which will form the predefined K

clusters.

4. Calculate the variance and place a new centroid of each cluster.

5. Repeat the third step, which means reassign each datapoint to the new closest centroid

of each cluster.

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

7. The model is ready.

**DBSCAN (Density-based spatial clustering of applications with noise)**

Partitioning and hierarchical methods are designed to find spherical-shaped clusters.

They have difficulty finding clusters of arbitrary shape such as “S” shape and oval clusters.

Density based clustering methods can be used to find clusters of arbitrary shape (or) non

spherical shape.

DBSCAN is one of the most popular unsupervised learning algorithms. The

DBSCAN algorithm is based on the intuitive notion of “clusters” and “noise”. The key idea is

that for each point of a cluster, the neighbourhood of a given radius has to contain at least a

minimum number of points.

**DBSCAN Algorithm requires two parameters:**

**1.**eps : It defines the neighbourhood around a data point i.e. if the distance between two

points is lower or equal to ‘eps’ then they are considered neighbors. If the eps value is

chosen too small then large part of the data will be considered as outliers.

**2.**MinPts: Minimum number of neighbors (data points) within eps radius. Larger the

dataset, the larger value of MinPts must be chosen.

**DBSCAN Algorithm has 3 types of data points:**

**1.** Core Point: A point is a core point if it has more than MinPts points within eps

**2.**Border Point: A point which has fewer than MinPts within eps but it is in the

neighbourhood of a core point

**3.**Noise or outlier: A point which is not a core point or border point