DATA MINING

MODULE 4 & 5

Prediction

- Another process of data analysis is prediction. It is used to find a numerical output. Same as in classification, the training dataset contains the inputs and corresponding numerical output values. The algorithm derives the model or a predictor according to the training dataset. The model should find a numerical output when the new data is given. Unlike in classification, this method does not have a class label. The model predicts a continuous-valued function or ordered value.
- **Regression is generally used for prediction.** Predicting the value of a house depending on the facts such as the number of rooms, the total area, etc., is an example for prediction.
- Regression is a statistical method used in finance, investing, and other disciplines that attempts to determine the strength and character of the relationship between one dependent variable (usually denoted by Y) and a series of other variables (known as independent variables).

- Types of Regression : **Linear Regression**
  Non - Linear Regression

Linear Regression

Simple Linear Regression:

Models the relationship between one independent variable and the dependent variable with a linear equation.

Multiple Linear Regression:

Extends simple linear regression to model the relationship between multiple independent variables and the dependent variable.

Ridge Regression (L2 Regularization):

Adds a regularization term to the linear regression cost function to prevent overfitting, especially in the presence of multicollinearity.

Lasso Regression (L1 Regularization):

Similar to Ridge Regression but uses the absolute values of coefficients as the regularization term, aiding in feature selection.

Non-Linear Regression:

Polynomial Regression:

Introduces polynomial terms to the linear regression equation to model non-linear relationships.

Logistic Regression:

Despite its name, logistic regression is used for binary classification problems. It models the probability of an instance belonging to a particular class using the logistic function.

Support Vector Regression (SVR):

Utilizes support vector machines for regression tasks, allowing for non-linear relationships between variables.

Decision Tree Regression:

Employs decision trees to model complex relationships by recursively splitting the data based on features.

Random Forest Regression:

Ensemble technique that builds multiple decision trees and averages their predictions, providing better generalization.

Gradient Boosting Regression:

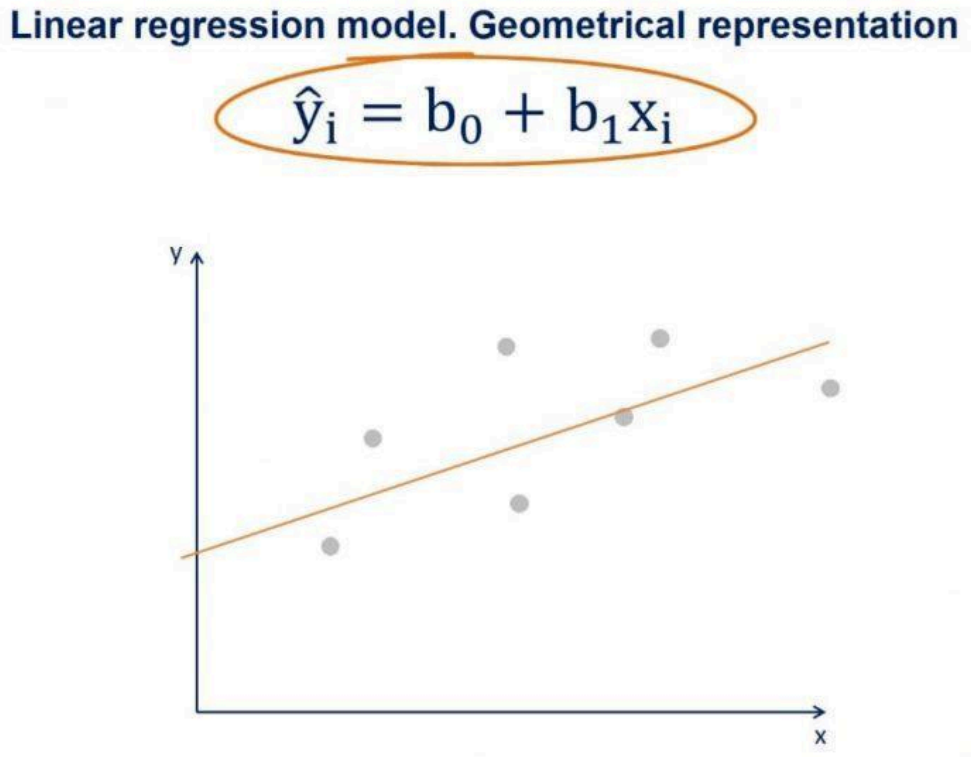
Builds an ensemble of weak learners (typically decision trees) sequentially, each correcting the errors of the previous one.

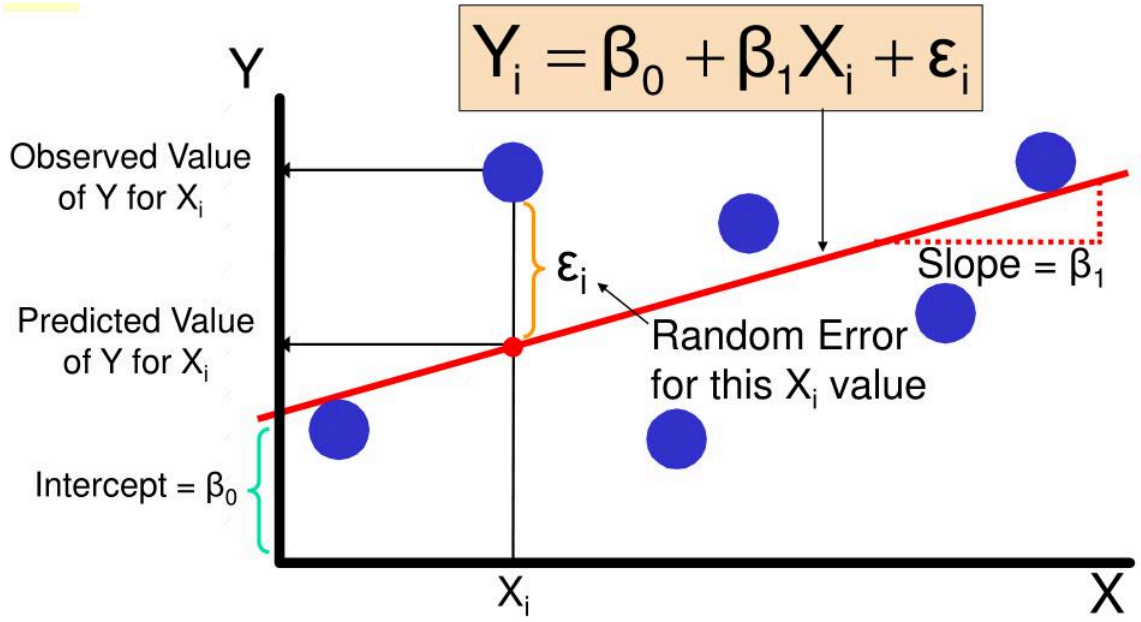
Linear Regression

- Linear regression is used to predict the relationship between two variables by applying a linear equation to observed data. There are two types of variable, one variable is called an independent variable, and the other is a dependent variable.
- The weight of the person is linearly related to their height. So, this shows a linear relationship between the height and weight of the person. According to this, as we increase the height, the weight of the person will also increase.
- It is not necessary that one variable is dependent on others, or one causes the other, but there is some critical relationship between the two variables. In such cases, we use a scatter plot to simplify the strength of the relationship between the variables.
- If there is no relation or linking between the variables then the scatter plot does not indicate any increasing or decreasing pattern. In such cases, the linear regression design is not beneficial to the given data.

The Regression Line

When we plot the data points on an x-y plane, the **regression line** is the best-fitting line through the data points.





- You can take a look at a plot with some data points in the picture above. We plot the line based on the **regression equation.**
- The gray points that are scattered are the observed values. B0, as we said earlier, is a constant and is the intercept of the **regression line** with the y-axis.

Simple Linear Regression Equation

- The measure of the relationship between two variables is shown by the correlation coefficient. The range of the coefficient lies between -1 to +1. This coefficient shows the strength of the association of the observed data between two variables.
- Linear Regression Equation is given below:

Y=a+bX

➔ *where **X** is the independent variable and it is plotted along the x-axis*
➔ ***Y** is the dependent variable and it is plotted along the y-axis*
➔ *Here, the slope of the line is **b(regression coefficient, calculated by Method of Least Squares)**, and **a** is the intercept (the value of y when*

x = 0).

$$a = \frac{[(\sum y)(\sum x^2) - (\sum x)(\sum xy)]}{[n(\sum x^2) - (\sum x)^2]}$$

$$b = \frac{n(\sum_{XY}) - (\sum_X)(\sum_Y)}{n(\sum_{x^2}) - (\sum_x)^2}$$

➔ ***n*** *represents the number of data points or observations in your dataset.*

Example:

Find a linear regression equation for the following two sets of data:

x	**2**	**4**	**6**	**8**
y	**3**	**7**	**5**	**10**

Solution:

Construct the table and find the value

x	**y**	**x^2**	**xy**
2	3	4	6
4	7	16	28
6	5	36	30
8	10	64	80

Σx = 20	Σy = 25	$\Sigma x^2 = 120$	Σxy = 144

The formula of the linear equation is y=a+bx.

Using the formula we will find the value of a and b.

$$a = \frac{[(\sum y)(\sum x^2) - (\sum x)(\sum xy)]}{[n(\sum x^2) - (\sum x)^2]}$$

$$\textbf{b=} \frac{n(\sum_{XY}) - (\sum_X)(\sum_Y)}{n(\sum_{x^2}) - (\sum_x)^2}$$

$$a = \frac{25 \times 120 - 20 \times 144}{4 \times 120 - 400}$$

$$a = \frac{120}{80}$$

a=1.5

$$\textbf{b=} \frac{n(\sum_{XY}) - (\sum_X)(\sum_Y)}{n(\sum_{x^2}) - (\sum_x)^2}$$

Put the values in the equation

$$b = \frac{4 \times 144 - 20 \times 25}{4 \times 120 - 400}$$

$$b = \frac{76}{80}$$

b=0.95

Hence we got the value of a = 1.5 and b = 0.95

- The linear equation is given by :Y = a + bx
- Now put the value of a and b in the equation

- **Hence equation of linear regression is y = 1.5 + 0.95x**

Multiple Linear Regression

- In a linear regression model we have one dependent and one independent variable. Multiple regression models involve multiple predictors or independent variables and one dependent variable.
- This is an extension of the linear regression problem.
- The multiple regression of two variables x1 and x2 is given as follows:

$$y = f(x_1, x_2)$$

$$y = a_0 + a_1x_1 + a_2x_2$$

- In general, this is given for 'n' independent variables as:

$$y = f(x_1, x_2, \ldots . x_n)$$

$$y = a_0 + a_1x_1 + a_2x_2 + \ldots \ldots + a_nx_n + \varepsilon$$

  - Here, x1, x2, ... xn are predictor variables, y is the dependent variable.
  - (a0, a1, a2, ... an) are the coefficients of the regression equation and ε is the error term.

Example:

x1 Product 1 Sales	x2 Product 2 Sales	Y Weekly Sales
1	4	1
2	5	6
3	8	8
4	2	12

- Here, the matrices for Y and X are given as follows:

$$X = \begin{pmatrix} 1 & 1 & 4 \\ 1 & 2 & 5 \\ 1 & 3 & 8 \\ 1 & 4 & 2 \end{pmatrix} \text{ and } Y = \begin{pmatrix} 1 \\ 6 \\ 8 \\ 12 \end{pmatrix}$$

The coefficients of multiple regression

$$a = \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix}.$$

- The regression coefficient for multiple regression is calculated the same way as linear regression:

$$\hat{a} = ((X^T X)^{-1} X^T) Y$$

$$X^T X = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 4 & 5 & 8 & 2 \end{pmatrix} \begin{pmatrix} 1 & 1 & 4 \\ 1 & 2 & 5 \\ 1 & 3 & 8 \\ 1 & 4 & 2 \end{pmatrix} = \begin{pmatrix} 4 & 10 & 19 \\ 10 & 30 & 46 \\ 19 & 46 & 109 \end{pmatrix}$$

$$(X^T X)^{-1} = \begin{pmatrix} 4 & 10 & 19 \\ 10 & 30 & 46 \\ 19 & 46 & 109 \end{pmatrix}^{-1} = \begin{pmatrix} 3.15 & -0.59 & -0.30 \\ -0.59 & 0.20 & 0.016 \\ -0.30 & 0.016 & 0.054 \end{pmatrix}$$

$$X^T X)^{-1} X^T = \begin{pmatrix} 3.15 & -0.59 & -0.30 \\ -0.59 & 0.20 & 0.016 \\ -0.30 & 0.016 & 0.054 \end{pmatrix} \times \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 4 & 5 & 8 & 2 \end{pmatrix} = \begin{pmatrix} 0.05 & 0.47 & -1.02 & 0.19 \\ -0.32 & -0.098 & 0.155 & 0.26 \\ -0.065 & 0.005 & 0.185 & -0.125 \end{pmatrix}$$

$$\hat{a} = ((X^T X)^{-1} X^T)Y = \begin{pmatrix} 0.05 & 0.47 & -1.02 & 0.19 \\ -0.32 & -0.098 & 0.155 & 0.26 \\ -0.065 & 0.005 & 0.185 & -0.125 \end{pmatrix} \times \begin{pmatrix} 1 \\ 6 \\ 8 \\ 12 \end{pmatrix} = \begin{pmatrix} -1.69 \\ 3.48 \\ -0.05 \end{pmatrix}$$

Then
a0= -1.69
a1= 3.48
a2= -0.05

y= a0 + a1x1 + a2x2

The the model is:

y= -1.69 + 3.48x1 - 0.05x2

Evaluating the performance of a linear regression model

- When evaluating the performance of a linear regression model, several metrics are commonly used to assess how well the model fits the data. Here are key evaluation criteria for linear regression:

Mean Squared Error (MSE):

- The MSE is the average of the squared differences between predicted and actual values. It measures the average squared deviation of predictions from the true values.

Formula:

MSE= $\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$

Lower MSE values indicate better model performance.

Root Mean Squared Error (RMSE):

- RMSE is the square root of the MSE, providing a measure of the average absolute deviation between predicted and actual values.

Formula:

RMSE = $\sqrt{MSE}$

Like MSE, lower RMSE values indicate better model performance.

Mean Absolute Error (MAE):

- MAE is the average of the absolute differences between predicted and actual values. It measures the average absolute deviation of predictions from the true values.

Formula:

MAE = $\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$

Lower MAE values indicate better model performance.

R-squared (R^2):

- R-squared represents the proportion of the variance in the dependent variable that is explained by the independent variables. It ranges from 0 to 1.
- Formula:

$$1 - \frac{\sum_{i=1}^{n}(yi - \widehat{yi})^2}{\sum_{i=1}^{n}(yi - \overline{yi})^2}$$

A higher R-squared value (closer to 1) indicates a better fit, while a value of 0 suggests that the model does not explain the variability in the data.

Adjusted R-squared:

- Adjusted R-squared adjusts the R-squared value based on the number

of predictors in the model. It penalizes the addition of irrelevant predictors.

A higher adjusted R-squared suggests a better trade-off between model complexity and explanatory power.

Residual Analysis:

- **Visual examination of residual plots** can help identify patterns or trends in the residuals. Residuals should be randomly distributed around zero, indicating that the model captures the underlying patterns in the data.
- When using these metrics, it's essential to consider the specific characteristics of the data and the goals of the modeling task. Different metrics may be more appropriate depending on the context and requirements of the regression analysis.

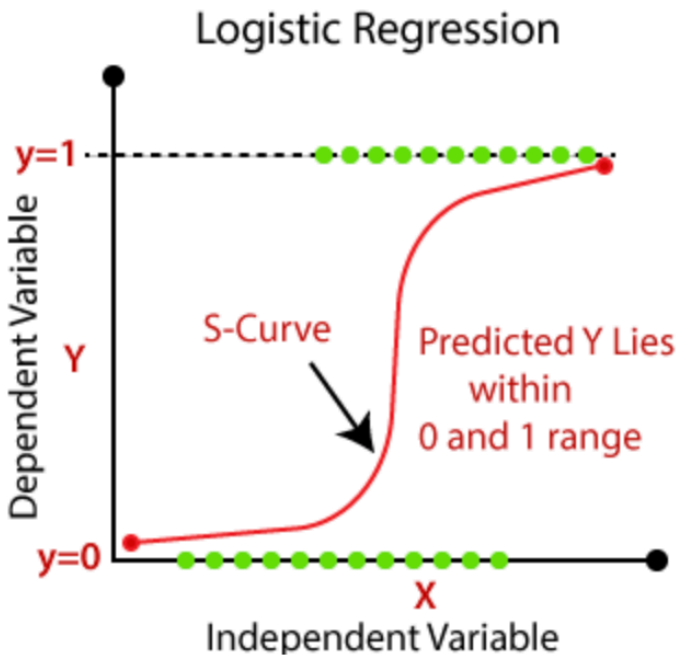
Logistic Regression

- Logistic regression is a statistical method used to model the relationship between a binary dependent variable and one or more independent variables.
- The goal of logistic regression is to predict the probability of an event occurring based on a set of predictor variables.
- In **logistic regression**, the dependent variable is binary, meaning **it can only take on two values**, typically labeled as 0 or 1. The independent variables can be either continuous or categorical.
- 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.
- For example whether a customer will buy a product or whether a patient has a certain disease or if the mail is spam or not etc.
- 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.



How does Logistic Regression work?

- The logistic regression model transforms the linear regression function continuous value output into categorical value output using a sigmoid function, which maps any real-valued set of independent variables input into a value between 0 and 1.
- **This function is known as the logistic function.**
- Let the independent input features be:

$$X = \begin{bmatrix} x_{11} & \cdots & x_{1m} \\ x_{21} & \cdots & x_{2m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nm} \end{bmatrix}$$

and the dependent variable is Y having only binary value i.e. 0 or 1.

$$Y = \begin{cases} 0 & \text{if } Class\ 1 \\ 1 & \text{if } Class\ 2 \end{cases}$$

then, apply the multi-linear function to the input variables X.

$$z = \left(\sum_{i=1}^{n} w_i x_i\right) + b$$

Here **xi** is the ith observation of X,

wi = [w1,w2,w3….,wm] is the **weights or Coefficient**,

b is the bias term also known as intercept. simply this can be represented as the dot product of weight and bias.

$$\mathbf{z = w.\ X + b}$$

- Now we use the sigmoid function where the **input will be z** and we find the probability between 0 and 1. i.e. predicted y.

$$\sigma(z) = \frac{1}{1+e^{-z}}$$

ie,

$$\sigma(z) = 1/(1+e^{-(w.X+b)})$$

- For a binary logistic regression model with multiple independent variables (x1,x2,...xn), the equation becomes :

$$\sigma(z) = 1/(1+e^{-(w1.X1+w2.X2+...+wn.Xn+b)})$$

- The probability of being a class can be measured as:

$$P(y=1) = \sigma(z)$$
$$P(y=0) = 1- \sigma(z)$$

Types of Logistic Regression

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

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”.

Polynomial Regression

- Polynomial Regression is a regression algorithm that models the relationship between a dependent(y) and independent variable(x) as nth degree polynomial.
- The Polynomial Regression equation is given below:

$$y= b_0+b_1x_1+ b_2x_1^2+ b_2x_1^3+..... b_nx_1^n$$

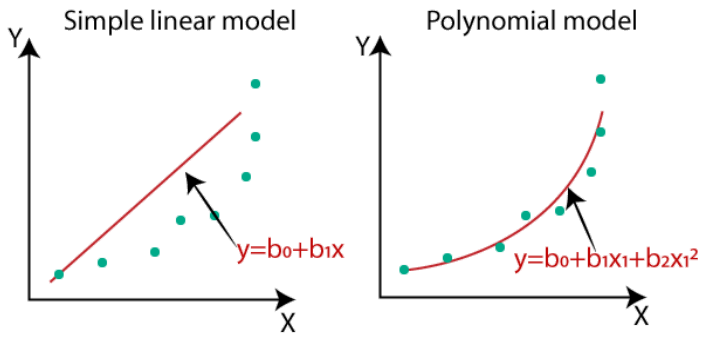
- It is also called the special case of Multiple Linear Regression in ML. Because we add some polynomial terms to the Multiple Linear regression equation to convert it into Polynomial Regression.
- It is a linear model with some modification in order to increase the accuracy.
- The dataset used in Polynomial regression for training is of non-linear nature.

- It makes use of a linear regression model to fit the complicated and non-linear functions and datasets.
- Hence, "In Polynomial regression, the original features are converted into Polynomial features of required degree (2,3,..,n) and then modeled using a linear model."

Need for Polynomial Regression:

The need of Polynomial Regression in ML can be understood in the below points:

- If we apply a linear model on a linear dataset, then it provides us a good result as we have seen in Simple Linear Regression, but if we apply the same model without any modification on a non-linear dataset, then it will produce a drastic output. Due to which loss function will increase, the error rate will be high, and accuracy will be decreased.
- So for such cases, where data points are arranged in a non-linear fashion, we need the Polynomial Regression model. We can understand it in a better way using the below comparison diagram of the linear dataset and non-linear dataset.



Clustering

Unsupervised Learning

- Unsupervised learning is the training of a machine using information that is neither classified nor labeled and allowing the algorithm to act on that information without guidance.
- The task of the machine is to group unsorted information according to similarities, patterns, and differences without any prior training of data.
- Unlike supervised learning,no training will be given to the machine. Therefore the machine is restricted to find the hidden structure in unlabeled data by itself.

What Is ClusterAnalysis?

The process of grouping a set of physical or abstract objects into classes of similar objects is called clustering. A cluster is a collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters. A cluster of data objects can be treated collectively as one group and so may be considered as a form of data compression.

Clustering in Machine Learning

- Clustering or cluster analysis is a machine learning technique, which groups the unlabeled dataset.
- "A way of grouping the data points into different clusters, 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 does it by finding some similar patterns in the unlabeled dataset such as shape, size,color,behavior,etc.,and divides them as per the presence and absence of those similar patterns.

Points to Remember

- A cluster of data objects can be treated as one group.
- While doing cluster analysis, we first partition the set of data into groups based on data similarity and then assign the labels to the groups.
- The main advantage of clustering over classification is that it is adaptable to changes and helps single out useful features that distinguish different groups.

Types of Clustering Methods

- The clustering methods are broadly divided into
  - Hard clustering(data point belongs to only one group)
  - Soft Clustering(data points can belong to another group also).

Below are the main clustering methods used in Machine learning:

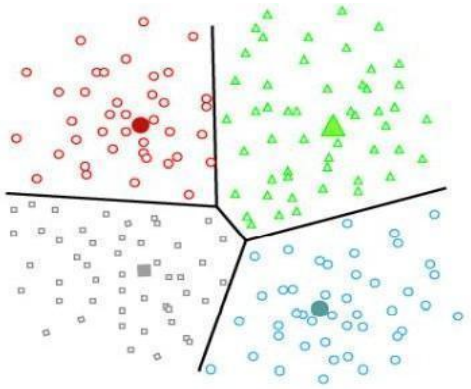
- Partitioning Clustering
- Density-Based Clustering
- Distribution Model-Based Clustering
- Hierarchical Clustering
- Fuzzy Clustering

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.
- Suppose we are given a database of 'n' objects and the partitioning method constructs 'k' partitions of data. Each partition will represent a cluster and $k \leq n$. It means that it will classify the data into k groups, which satisfy the following requirements −
  - Each group contains at least one object.
  - Each object must belong to exactly one group.

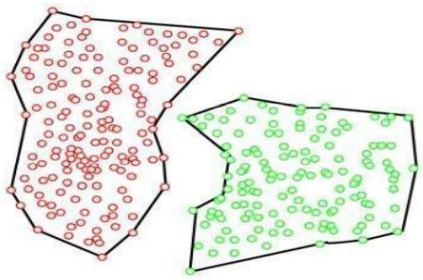
The clusters are formed to optimize an objective partitioning criterion, such as a dissimilarity function based on distance, so that the objects within a cluster are "similar," whereas the objects of different clusters are "dissimilar" in terms of the data set attributes.

- **The most common example of partitioning clustering is the K-Means Clustering** algorithm. In this type, the dataset is divided into a set of k groups, where K is used to define the number of pre-defined groups. The cluster center is created in such a way that the distance between the data points of one cluster is minimum as compared to another cluster centroid.



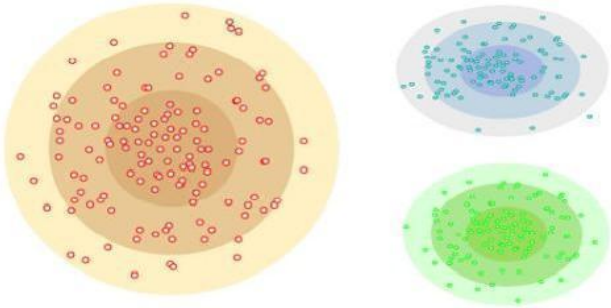
Density-BasedClustering

- The density-based clustering method connects the highly-dense areas into clusters, and the arbitrarily shaped distributions are formed as long as the dense region can be connected.
- This algorithm does it by identifying different clusters in the dataset and connects the areas of high densities into clusters.
- The dense areas in data space are divided from each other by sparse areas.
- To discover clusters with arbitrary shape, density-based clustering methods have been developed. These typically regard clusters as dense regions of objects in the data space that are separated by regions of low density (representing noise). The basic idea is to continue growing the given cluster as long as the density in the neighborhood exceeds some threshold,i.eFor each data point within a given cluster, the radius of a given cluster has to contain at least a minimum number of points.



Distribution Model-Based Clustering

- In the distribution model-based clustering method, 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).
- Model-based clustering methods attempt to optimize the fit between the given data and some mathematical model. Such methods are often based on the assumption that the data are generated by a mixture of underlying probability distributions. In this method, a model is hypothesized for each cluster to find the best fit of data for a given model. This method locates the clusters by clustering the density function. It reflects spatial distribution of the data points.
- This method also provides a way to automatically determine the number of clusters based on standard statistics,taking outlier or noise into account. It therefore yields robust clustering methods.



Hierarchical Clustering

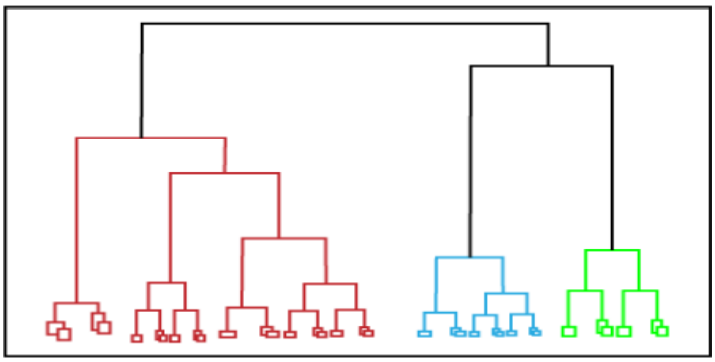
- Hierarchical clustering can be used as an alternative for the partitioned clustering as there is no requirement of pre-specifying the number of clusters to be created.

- In this technique, the dataset is divided into clusters to create a treelike structure,which is also called a dendrogram.
- The observations or any number of clusters can be selected by cutting the tree at the correct level.
    - Agglomerative(bottom up approach)
    - Divisive (top down approach)

Agglomerative hierarchical clustering: This bottom-up strategy starts by placing each object in its own cluster and then merges these atomic clusters into larger and larger clusters, until all of the objects are in a single cluster or until certain termination conditions are satisfied. Most hierarchical clustering methods belong to this category. They differ only in their definition of inter cluster similarity.

Divisive hierarchical clustering: This top-down strategy does the reverse of agglomerative hierarchical clustering by starting with all objects in one cluster. It subdivides the cluster into smaller and smaller pieces, until each object forms a cluster on its own or until it satisfies certain termination conditions, such as a desired number of clusters is obtained or the diameter of each cluster is within a certain threshold.

- A hierarchical clustering method works by grouping data objects into a tree of clusters. Hierarchical clustering methods can be further classified as either agglomerative or divisive, depending on whether the hierarchical decomposition is formed in a bottom-up (merging) or top-down (splitting) fashion.
- The quality of a pure hierarchical clustering method suffers from its inability to perform adjustment once a merge or split decision has been executed. That is, if a particular merge or split decision later turns out to have been a poor choice, the method cannot backtrack and correct it.



Grid-Based methods

Grid-based methods quantize the object space into a finite number of cells that form a grid structure. All of the clustering operations are performed on the grid structure (i.e., on the quantized space). The main advantage of this approach is its fast processing time, which is typically independent of the number of data objects and dependent only on the number of cells in each dimension in the quantized space.

K-Means Algorithm

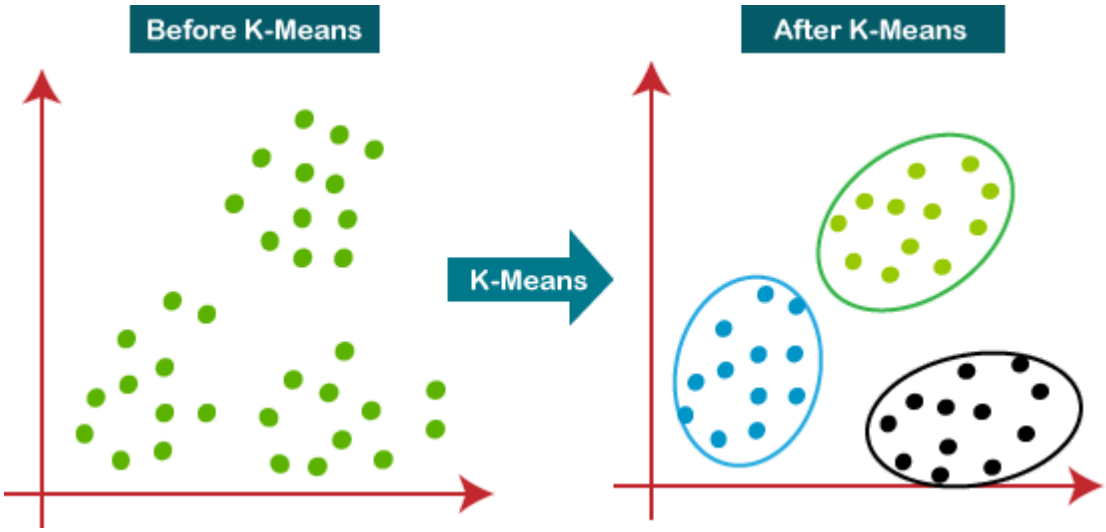
- K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters. Here K defines the number of predefined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on.
- It is an iterative algorithm that divides the unlabeled dataset into k different clusters in such a way that each dataset belongs to 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 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.

Hence each cluster has data points with some commonalities, and it is away from other clusters.

The below diagram explains the working of the K-means Clustering Algorithm:



An Example:

- A pizza chain wants to open its delivery centers across a city. What do you think would be the possible challenges?

- They need to analyze the areas from where the pizza is being ordered frequently. They need to understand how many pizza stores has to be opened to cover delivery in the area. They need to figure out the locations for the pizza stores within all these areas in order to keep the distance between the store and delivery points minimum.

How does the k-means algorithm work?

- The k-means algorithm defines the centroid of a cluster as the mean value of the points within the cluster. It proceeds as follows.
- First, it randomly selects k of the objects in D, each of which initially represents a cluster mean or center.
- Foreach of the remaining objects, an object is assigned to the cluster to which it is the most similar, based on the Euclidean distance between the object and the cluster mean.

- The k-means algorithm then iteratively improves the within-cluster variation.
- For each cluster, it computes the new mean using the objects assigned to the cluster in the previous iteration.
- All the objects are then reassigned using the updated means as the new cluster centers. The iterations continue until the assignment is stable, that is, the clusters formed in the current round are the same as those formed in the previous round.

K-Means Algorithm

• 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 different 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.

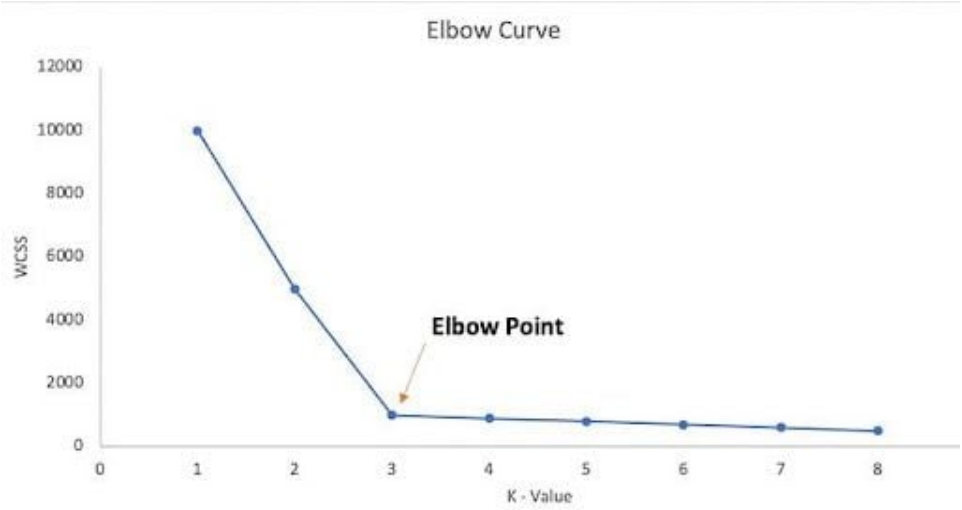
• **Step-6:** If any reassignment occurs, then go to step-4 else go to FINISH.

• **Step-7**: The model is ready.

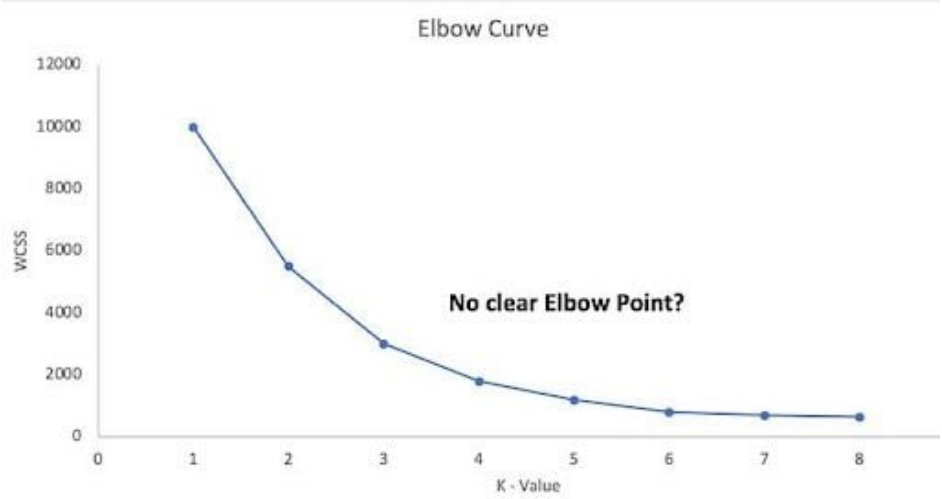
What Is the Elbow Method?

- The elbow method involves finding the optimal k via a graphical representation. It works by finding the within-cluster sum of square (WCSS), i.e. the sum of the square distance between points in a cluster and the cluster centroid.
- The elbow graph shows WCSS values on the y-axis corresponding to the different values of K on the x-axis.

- When we see an elbow shape in the graph, we pick the K-value where the elbow gets created. We can call this the elbow point.
- Beyond the elbow point, increasing the value of ‘K’ does not lead to a significant reduction in WCSS.



What is Silhouette Method?



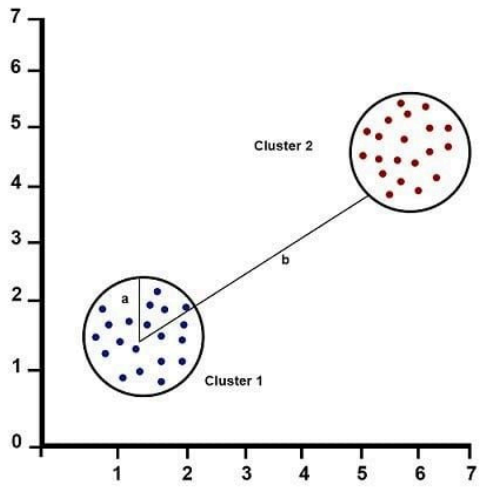
➔ The Silhouette score is a very useful method to find the number of K when the elbow method doesn’t show the elbow point.

➔ The value of the Silhouette score ranges from -1 to 1. Following is the interpretation of the Silhouette score.

➔ **1:** Points are perfectly assigned in a cluster and clusters are easily distinguishable.

➔ **0:** Clusters are overlapping.

➔ **-1:** Points are wrongly assigned in a cluster.




```
Silhouette Score = (b-a)/max(a,b)
```

Where:

- `a` **= average intra-cluster distance, i.e the average distance between each point within a cluster.**
- `b` **= average inter-cluster distance i.e the average distance between all clusters.**

K-Medoids Clustering Algorithm (Partitioning Around Medoid - PAM)

Having an overview of K-Medoids clustering, let us discuss the algorithm for the same.

1. First, we select K random data points from the dataset and use them as medoids.
2. Now, we will calculate the distance of each data point from the medoids. You can use any of the Euclidean, Manhattan distance, or squared Euclidean distance as the distance measure.

3. Once we find the distance of each data point from the medoids, we will assign the data points to the clusters associated with each medoid. The data points are assigned to the medoids at the closest distance.
4. After determining the clusters, we will calculate the sum of the distance of all the non-medoid data points to the medoid of each cluster. Let the cost be Ci.
5. Now, we will select a random data point Dj from the dataset and swap it with a medoid Mi. Here, Dj becomes a temporary medoid. After swapping, we will calculate the distance of all the non-medoid data points to the current medoid of each cluster. Let this cost be Cj.
6. If Ci>Cj, the current medoids with Dj as one of the medoids are made permanent medoids. Otherwise, we undo the swap, and Mi is reinstated as the medoid.
7. Repeat 4 to 6 until no change occurs in the clusters.

K-Medoids Clustering Numerical Example With Solution

The dataset for clustering is as follows.

Point	Coordinates
A1	(2, 6)
A2	(3, 8)
A3	(4, 7)
A4	(6, 2)
A5	(6, 4)
A6	(7, 3)
A7	(7,4)
A8	(8, 5)
A9	(7, 6)
A10	(3, 4)

Iteration 1

➔ Suppose that we want to group the above dataset into two clusters. So, we will randomly choose two medoids.

➔ Here, the choice of medoids is important for efficient execution. Hence, we have selected two points from the dataset that can be potential medoid for the final clusters. Following are two points from the dataset that we have selected as medoids.

- **M1 = (3, 4)**
- **M2 = (7, 3)**

➔ Now, we will calculate the distance between each data point and the medoids using the Manhattan distance measure.

$$d = |X1 - X2| + |Y1 - Y2|$$

The results have been tabulated as follows.

Point	Coordinates	Distance From M1 (3,4)	Distance from M2 (7,3)	Assigned Cluster
A1	(2, 6)	3	8	Cluster 1
A2	(3, 8)	4	9	Cluster 1
A3	(4, 7)	4	7	Cluster 1
A4	(6, 2)	5	2	Cluster 2
A5	(6, 4)	3	2	Cluster 2
A6	(7, 3)	5	0	Cluster 2
A7	(7,4)	4	1	Cluster 2
A8	(8, 5)	6	3	Cluster 2
A9	(7, 6)	6	3	Cluster 2
A10	(3, 4)	0	5	Cluster 1

Iteration 1

The clusters made with medoids (3, 4) and (7, 3) are as follows.

- Points in cluster1= {(2, 6), (3, 8), (4, 7), (3, 4)}
- Points in cluster 2= {(7,4), (6,2), (6, 4), (7,3), (8,5), (7,6)}

After assigning clusters, we will calculate the cost for each cluster and find their sum. The cost is nothing but the sum of distances of all the data points from the medoid of the cluster they belong to.

Hence, the cost for the current cluster will be 3+4+4+2+2+0+1+3+3+0=22.

Iteration 2

Now, we will select another non-medoid point (7, 4) and make it a temporary medoid for the second cluster. Hence,

- **M1 = (3, 4)**
- **M2 = (7, 4)**

Now, let us calculate the distance between all the data points and the current medoids.

Point	Coordinates	Distance From M1 (3,4)	Distance from M2 (7,4)	Assigned Cluster
A1	(2, 6)	3	7	Cluster 1
A2	(3, 8)	4	8	Cluster 1
A3	(4, 7)	4	6	Cluster 1
A4	(6, 2)	5	3	Cluster 2
A5	(6, 4)	3	1	Cluster 2
A6	(7, 3)	5	1	Cluster 2
A7	(7,4)	4	0	Cluster 2
A8	(8, 5)	6	2	Cluster 2
A9	(7, 6)	6	2	Cluster 2
A10	(3, 4)	0	4	Cluster 1

Iteration 2

The data points haven't changed in the clusters after changing the medoids. Hence, clusters are:

- Points in cluster1: {(2, 6), (3, 8), (4, 7), (3, 4)}
- Points in cluster 2: {(7,4), (6,2), (6, 4), (7,3), (8,5), (7,6)}

Now, let us again calculate the cost for each cluster and find their sum. The total cost this time will be 3+4+4+3+1+1+0+2+2+0=20.

Here, the **current cost is less than the cost calculated in the previous iteration**. Hence, we will make the swap permanent and **make (7,4) the medoid for cluster 2.** If the cost this time was greater than the previous cost i.e. 22, we would have to revert the change.

New medoids after this iteration are (3, 4) and (7, 4) with no change in the clusters.

Iteration 3

Now, **let us again change the medoid of cluster 2 to (6, 4).** Hence, the new medoids for

the clusters are **M1=(3, 4) and M2= (6, 4).**
Let us calculate the distance between the data points and the above medoids to find the new cluster. The results have been tabulated as follows.

Point	Coordinates	Distance From M1 (3,4)	Distance from M2 (6,4)	Assigned Cluster
A1	(2, 6)	3	6	Cluster 1
A2	(3, 8)	4	7	Cluster 1
A3	(4, 7)	4	5	Cluster 1
A4	(6, 2)	5	2	Cluster 2
A5	(6, 4)	3	0	Cluster 2
A6	(7, 3)	5	2	Cluster 2
A7	(7,4)	4	1	Cluster 2
A8	(8, 5)	6	3	Cluster 2
A9	(7, 6)	6	3	Cluster 2
A10	(3, 4)	0	3	Cluster 1

Iteration 3
Again, the clusters haven't changed. Hence, clusters are:

- Points in cluster1:{(2, 6), (3, 8), (4, 7), (3, 4)}
- Points in cluster 2:{(7,4), (6,2), (6, 4), (7,3), (8,5), (7,6)}

Now, let us again calculate the cost for each cluster and find their sum. The total cost this time will be 3+4+4+2+0+2+1+3+3+0=22.

- The current cost is 22 which is greater than the cost in the previous iteration i.e. 20. Hence, we will revert the change and the point **(7, 4) will again be made the medoid for cluster 2.**
- So, the clusters after this iteration will be cluster1 = {(2, 6), (3, 8), (4, 7), (3, 4)} and cluster 2= {(7,4), (6,2), (6, 4), (7,3), (8,5), (7,6)}. The medoids are (3,4) and (7,4).
- We keep replacing the medoids with a non-medoid data point. The set of medoids for which the cost is the least, the medoids, and the associated clusters are made permanent. So, after all the iterations, you will get the final clusters and their medoids.
- **The K-Medoids clustering algorithm is a computation-intensive algorithm that requires many iterations. In each iteration, we need to calculate the distance between the medoids and the data points, assign clusters, and compute the**

cost. Hence, K-Medoids clustering is not well suited for large data sets.

Advantages:

- It is simple to understand and easy to implement.
- K-Medoid Algorithm is fast and converges in a fixed number of steps.
- PAM is less sensitive to outliers than other partitioning algorithms.

Disadvantages:

- The main disadvantage of K-Medoid algorithms is that it is not suitable for clustering non-spherical (arbitrary shaped) groups of objects.
- This is because it relies on minimizing the distances between the non-medoid objects and the medoid (the cluster centre) – briefly, it uses compactness as clustering criteria instead of connectivity.
- It may obtain different results for different runs on the same dataset because the first k medoids are chosen randomly.

"Which method is more robust—k-means or k-medoids?"

- The k-medoids method is more robust than k-means in the presence of noise and outliers because a medoid is less influenced by outliers or other extreme values than a mean.
- However, the complexity of each iteration in the k-medoids algorithm is O(k(n−k)^2).
- For large values of n and k,such computation becomes very costly, and much more costly than the k-means method.
- Both methods require the user to specify k, the number of clusters.

How can we scale up the k-medoids method?

- A typical k-medoids partitioning algorithm like PAM works effectively for small data sets, but does not scale well for large data sets.
- To deal with larger data sets, a sampling-based method called CLARA (Clustering Large Applications) can be used.
- Instead of taking the whole data set into consideration, CLARA uses a random sample of the data set.
- The PAM algorithm is then applied to compute the best medoids from the sample. Ideally, the sample should closely represent the original data set.
- In many cases, a large sample works well if it is created so that each object

has equal probability of being selected into the sample.

- The representative objects (medoids) chosen will likely be similar to those that would have been chosen from the whole data set.
- CLARA builds clustering from multiple random samples and returns the best clustering as the output. The complexity of computing the medoids on a random sample is O(ks2 + k(n − k)), where s is the size of the sample, k is the number of clusters, and n is the total number of objects.
- CLARA can deal with larger data sets than PAM. The effectiveness of CLARA depends on the sample size.
- Notice that PAM searches for the best k- medoids among a given data set, whereas CLARA searches for the best k-medoids among the selected sample of the data set.
- CLARA cannot find a good clustering if any of the best sampled medoids is far from the best k-medoids. **If an object is one of the best k-medoids but is not selected during sampling, CLARA will never find the best clustering.**

Clustering Large Applications (CLARA)

- CLARA is an extension to k-medoids (PAM) methods to deal with data containing a large number of objects (more than several thousand observations) in order to reduce computing time and RAM storage problems.
- This is achieved using the sampling approach.
- A random sample should closely represent the original data.
- The chosen medoids will likely be similar to what would have been chosen from the whole data set.

Algorithm

- Create randomly, from the original dataset, multiple subsets with fixed size(sample size)
- Compute PAM algorithm on each subset and choose the corresponding k representative objects (medoids). Assign each observation of the entire data set to the closest medoid.
- Calculate the mean (or the sum) of the dissimilarities of the observations to their closest medoid. This is used as a measure of the goodness of the clustering.

- Retain the sub-dataset for which the mean(or sum) is minimal. A further analysis is carried out on the final partition.

CLARA Properties

• Complexity of each Iteration is:

• O(ks2+k(n-k))

  • s: the size of the sample

  • k: number of clusters

  • n: number of objects

• PAM finds the best k medoids among a given data, and CLARA finds the best k medoids among the selected samples

• **Problems**

  • The best k medoids may not be selected during the sampling process, in this case, CLARA will never find the best clustering

  • If the sampling is biased we cannot have a good clustering

  • Trade Off-of efficiency.

"How might we improve the quality and scalability of CLARA?"

- Recall that when searching for better medoids, PAM examines every object in the data set against every current medoid, whereas CLARA confines the candidate medoids to only a random sample of the data set.
- A randomized algorithm called CLARANS (Clustering Large Applications based upon Randomized Search) presents a trade-off between the cost and the effectiveness of using samples to obtain clustering.
- First, it randomly selects k objects in the data set as the current medoids. It then randomly selects a current medoid x and an object y that is not one of the current medoids.
- Can replacing x by y improve the absolute-error criterion? If yes, the replacement is made. CLARANS conducts such a randomized search l times.
- The set of the current medoids after the l steps is considered a local optimum. CLARANS repeats this randomized process m times and returns the best local optimal as the final result.

CLARANS(Clustering Large Applications based upon Randomized Search)

- CLARANS (Clustering Large Applications based upon Randomized Search) was proposed to improve the quality and the scalability of CLARA.
- It combines sampling techniques with PAM.
- It does not confine itself to any sample at a given time.
- It draws a sample with some randomness in each step of the search(draws sample of neighbors dynamically)
- The clustering process can be presented as searching a graph where every node is a potential solution, that is, a set of k medoids.
- If the local optimum is found, CLARANS starts with a new randomly selected node in search for a new local optimum.
- It is more efficient and scalable than both PAM and CLARA.
- Focusing techniques and spatial access structures may further improve its performance.

*CLARA works by taking **random samples** and applying k-medoid, while CLARANS explores the dataset through a **randomized search to find representative medoids**. Both algorithms aim to cluster large datasets efficiently but differ in their approaches to achieve this goal.*

Advantages:

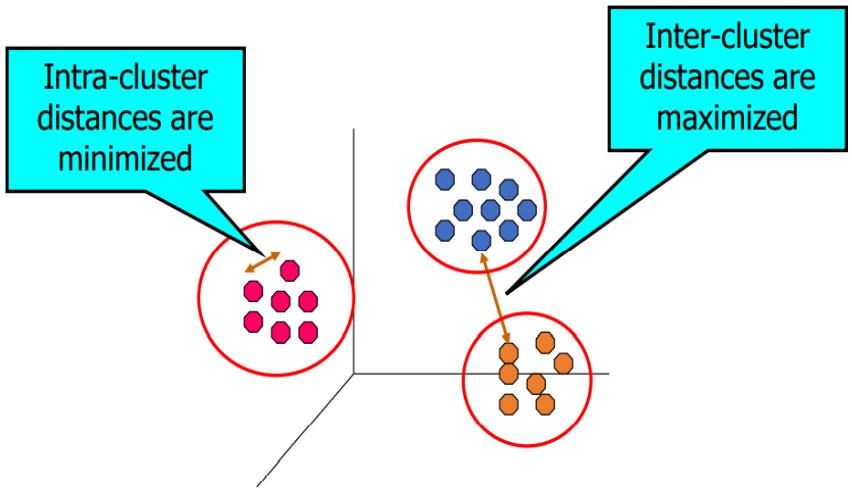
- Experiments show that CLARANS is more effective than both PAM and CLARA.
- Handles outliers.

Disadvantages:

- The computational complexity of CLARANS is $O(n^2)$, where n is the number of objects.
- The clustering quality depends on the sampling method.

Cluster Analysis

Finding groups of objects such that the objects in a group will be similar(correlated) to one another and different from (or unrelated to) the objects in other groups



Quality: What Is Good Clustering?

• A good clustering method will produce high quality clusters with

  • **High intra-class similarity**

  • **Low inter-class similarity**

• The quality of a clustering result depends on both the similarity measure used by the method and its implementation.

• The quality of a clustering method is also measured by its ability to discover some or all of the hidden patterns.

Measure the Quality of Clustering

- Dissimilarity/Similarity metric: Similarity is expressed in terms of a distance function, typically metric: d(i, j)
- There is a separate "quality" function that measures the "goodness" of a cluster.
- The definitions of distance functions are usually very different for interval-scaled, boolean, categorical, ordinal ratio, and vector variables.
- Weights should be associated with different variables based on applications and data semantics.
- It is hard to define "similar enough" or "good enough" ; the answer is typically highly subjective.

Data Structures

- Main memory-based clustering algorithms typically operate neither of the following two data structures.

Types of data structures in cluster analysis are:

- **Data Matrix**(or object by variable structure)
- **Dissimilarity Matrix**(or object by object structure)

Data Matrix

- This represents n objects, such as persons, with p variables (also called measurements or attributes), such as age, height, weight, gender, race and so on. The structure is in the form of a relational table, or n-by-p matrix (n objects x p variables).
- The Data Matrix is often called a two-mode matrix since the rows and columns of this represent the different entities.

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix}$$

Dissimilarity Matrix

- This stores a collection of proximities that are available for all pairs of n objects.
- It is often represented by a n–by–n table, where d(i,j) is the measured difference or dissimilarity between objects i and j.
- In general, d(i,j) is a non-negative number that is close to 0 when objects i and j are higher similar or “near” each other and become larger the more they differ.
- Since d(i,j) = d(j,i) and d(i,i) =0, we have the matrix in figure.
- This is also called a one mode matrix since the rows and columns of this represent the same entity.

Types Of Data In Cluster Analysis Are:

- Interval-scaled variables
- Binary variables

- Nominal, ordinal, and ratio variables
- Variables of mixed types

• Dissimilarity Between Objects

• Distances are normally used to measure the similarity or dissimilarity between two data objects described by interval-scaled variables

- Euclidean distance: the most popular distance measure

$$d(i,j)=\sqrt{(|x_{i_1}-x_{j_1}|^2+|x_{i_2}-x_{j_2}|^2+...+|x_{i_p}-x_{j_p}|^2)}$$

  – where $i=(x_{i1}, x_{i2}, \ldots, x_{ip})$ and $j=(x_{j1}, x_{j2}, \ldots, x_{jp})$ are two p-dimensional data objects

- Manhattan (city block) distance: another well-known metric

$$d(i,j)=|x_{i_1}-x_{j_1}|+|x_{i_2}-x_{j_2}|+...+|x_{i_p}-x_{j_p}|$$

•
•

- Minkowski distance: a generalization of both Euclidean distance and Manhattan distance

$$d(i,j)=\sqrt[q]{(|x_{i_1}-x_{j_1}|^q+|x_{i_2}-x_{j_2}|^q+...+|x_{i_p}-x_{j_p}|^q)}$$

  – Where q is a positive integer

  – It represents the Manhattan distance when $q = 1$ *and* *Euclidean distance* when $q = 2$

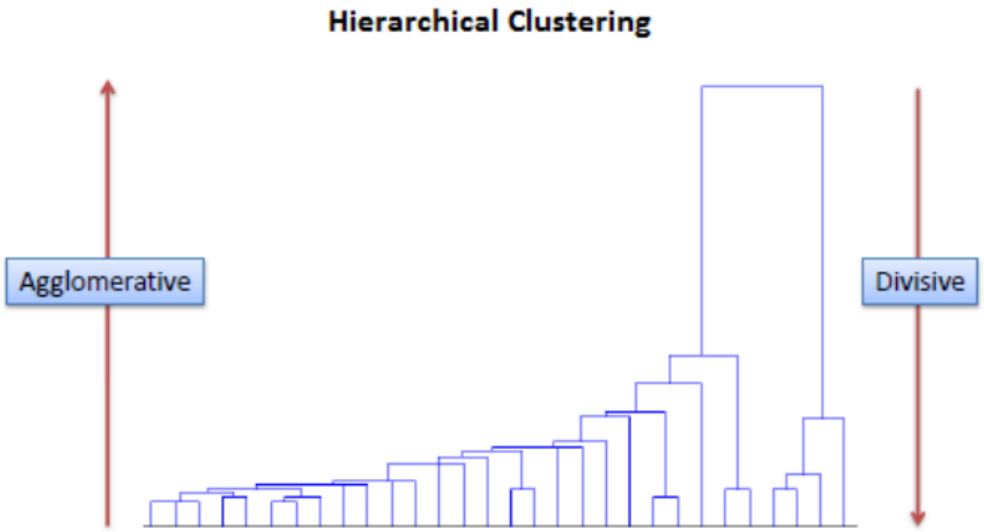
Hierarchical Clustering

- A hierarchical clustering method works by grouping data objects into a hierarchy or "tree" of clusters. Representing data objects in the form of a hierarchy is useful for data summarization and visualization.
- In this algorithm, we develop the hierarchy of clusters in the form of a tree, and this tree-shaped structure is known as the **dendrogram.**
- Set of methods that recursively cluster two items at a time.

The hierarchical clustering technique has two approaches:

- ***Agglomerative***: Agglomerative is a bottom-up approach, in which the algorithm starts with taking all data points as single clusters and merging them until one cluster is left.

- ***Divisive***: Divisive algorithm is the reverse of the agglomerative algorithms as It is a top-down approach.



Why hierarchical clustering?

- As we already have other clustering algorithms, then why do we need hierarchical clustering?
- So, as we have seen in the K-means clustering, there are some challenges with this algorithm, which are a predetermined number of clusters, and it always tries to create the clusters of the same size.
- ***To solve these two challenges, we can opt for the hierarchical clustering algorithm because, in this algorithm, we don't need to have***

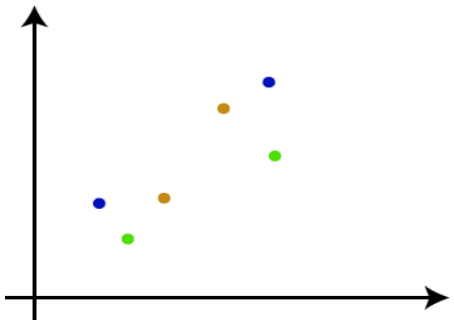
knowledge about the predefined number of clusters.

Agglomerative Hierarchical Clustering

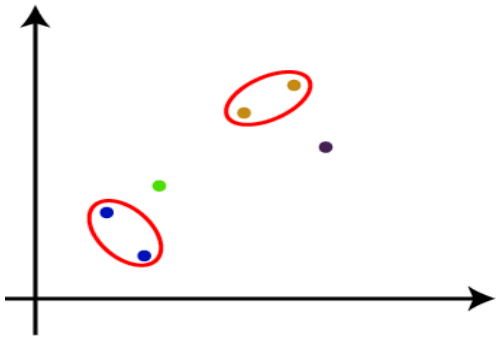
- The agglomerative hierarchical clustering algorithm is a popular example of HCA.
- **An agglomerative hierarchical clustering method is called AGNES (AGglomerative NESting).**
- To group the datasets into clusters, it follows the **bottom-up approach**.
- It means, this algorithm considers each dataset as a single cluster at the beginning, and then starts combining the closest pair of clusters together.
- It does this until all the clusters are merged into a single cluster that contains all the datasets.
- This hierarchy of clusters is represented in the form of the dendrogram.

How does Agglomerative Hierarchical clusteringWork?

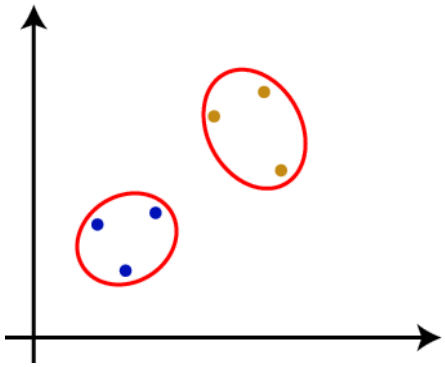
The working of the AHC algorithm can be explained using below steps:

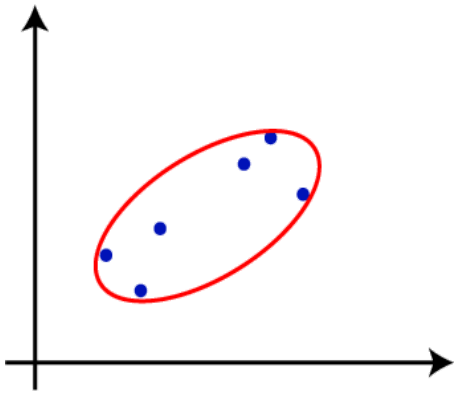
- **Step-1:**Create each data point as a single cluster. Let's say there are N data points, so the number of clusters will also be N.
- 

- **Step-2:**Take two closest data points or clusters and merge them to form one cluster. So, there will now be N-1 clusters.
- **Step-3**: Again, take the two closest clusters and merge them together to form one cluster. There will be N-2 clusters.



- **Step-4:** Repeat Step3 until only one clusterleft. So, we will get the following clusters.





- **Step-5:** Once all the clusters are combined into one big cluster, develop the dendrogram to divide the clusters as per the problem.

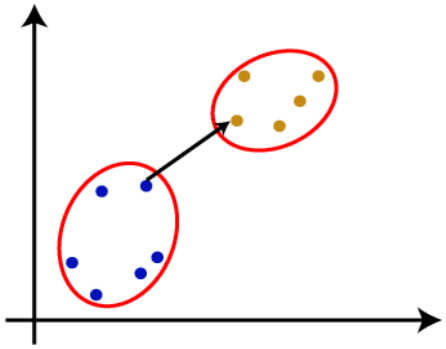
Measure for the distance between two clusters

- As we have seen, the **closest distance** between the two clusters is crucial for hierarchical clustering.
- There are various ways to calculate the distance between two clusters,

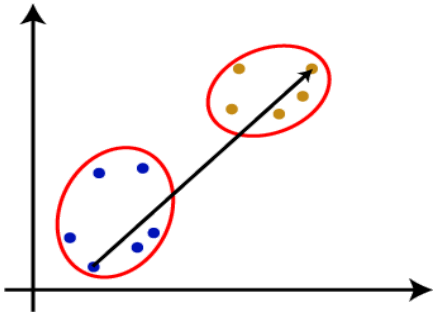
and these ways decide the rule for clustering.

- These measures are called **Linkage methods**. Some of the popular linkage methods are given below:

1. **Single Linkage:** It is the Shortest Distance between the closest points of the clusters. Consider the below image:



2. **Complete Linkage:** It is the farthest distance between the two points of two different clusters. It is one of the popular linkage methods as it forms tighter clusters than single-linkage.

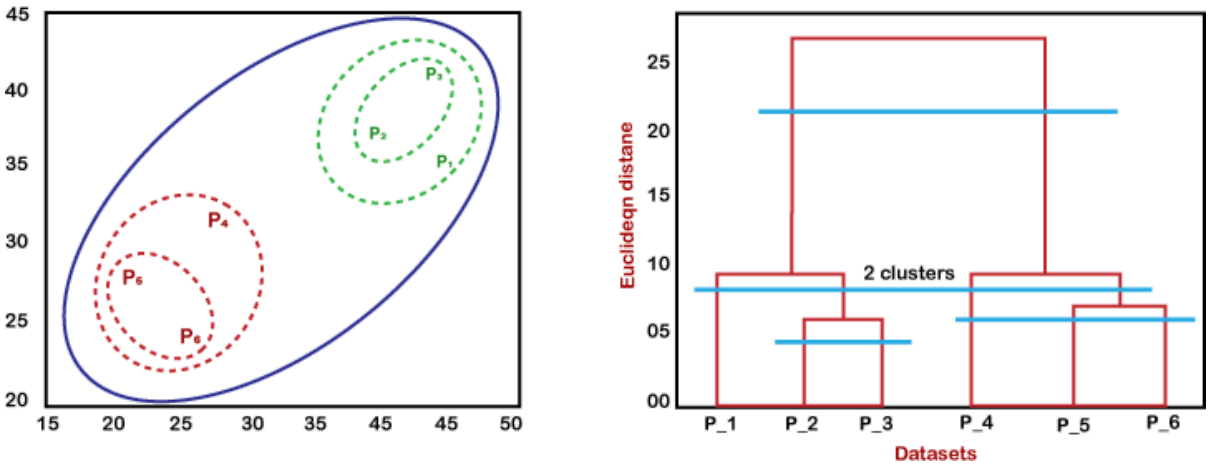


3. **Average Linkage:** It is the linkage method in which the distance between each pair of datasets is added up and then divided by the total number of datasets to calculate the average distance between two clusters. It is also one of the most popular linkage methods.
4. **Centroid Linkage:** It is the linkage method in which the distance between the centroid of the clusters is calculated.

From the above-given approaches, we can apply any of them according to the type of problem or business requirement.

Working of Dendrogram in Hierarchical Clustering

- The dendrogram is a tree-like structure that is mainly used to store each step as a memory that the HC algorithm performs.
- In the dendrogram plot, the Y-axis shows the Euclidean distances between the data points, and the x-axis shows all the data points of the given dataset.
- The working of the dendrogram can be explained using the below diagram:



In the above diagram, the left part is showing how clusters are created in agglomerative clustering, and the right part is showing the corresponding dendrogram.

- As we have discussed above, firstly, the data points P2 and P3 combine together and form a cluster, correspondingly a dendrogram is created, which connects P2 and P3 with a rectangular shape. The height is decided according to the Euclidean distance between the data points.
- In the next step, P5 and P6 form a cluster, and the corresponding dendrogram is created. It is higher than the previous, as the Euclidean distance between P5 and P6 is a little bit greater than the P2 and P3.
- Again, two new dendrograms are created that combineP1,P2,and P3 in one dendrogram, and P4, P5, and P6, in another dendrogram.
- At last, the final dendrogram is created that combines all the data points together.
- We can cut the dendrogram tree structure at any level as per our requirement.

Summary of Linkage functions

Name of linkage function	Form of linkage function
Single	f=min(d(x,y))
Complete	f=max(d(x,y))
Average	f=average(d(x,y))
Centroid	d(ave(X),ave(Y)) where we take the average over all items in each cluster

Example Single and Complete Linkage:

For the given data set, create the distance matrix and perform single link and complete link hierarchical clustering. Show your results by drawing a dendrogram. The dendrogram should clearly show the order in which the points are merged.

Sample No.	X	Y
P1	0.40	0.53
P2	0.22	0.38
P3	0.35	0.32
P4	0.26	0.19
P5	0.08	0.41
P6	0.45	0.30

Solution:

Create the distance matrix using Euclidean distance:

	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.23	0				
P3	0.22	0.14	0			
P4	0.37	0.19	0.13	0		
P5	0.34	0.14	0.28	0.23	0	
P6	0.24	0.24	0.10	0.22	0.39	0

- From this, find out the minimum value. Here it is 0.10, the distance between P6 and P3.
- So make them as a single cluster (P3,P6) .
- Now eliminate the row and column of P6.

	P1	P2	P3, P6	P4	P5
P1	0				
P2	0.23	0			
P3, P6	0.22	0.14	0		
P4	0.37	0.19	0.13	0	
P5	0.34	0.14	0.28	0.23	0

- Now find the next minimum value. Here it is 0.13, that is between (P3,P6) and P4.
- So make them into a cluster.

	P1	P2	P3, P6, P4	P5
P1	0			
P2	0.23	0		
P3, P6, P4	0.22	0.14	0	
P5	0.34	0.14	0.28	0

-
- Now consider the minimum value, say P5. (Check the correct value

after two decimal places) or choose an arbitrary value.

$$\begin{pmatrix} & P1 & P2, P5 & P3, P6, P4 \\ P1 & 0 & & \\ P2, P5 & 0.23 & 0 & \\ P3, P6, P4 & 0.22 & 0.14 & 0 \end{pmatrix}$$

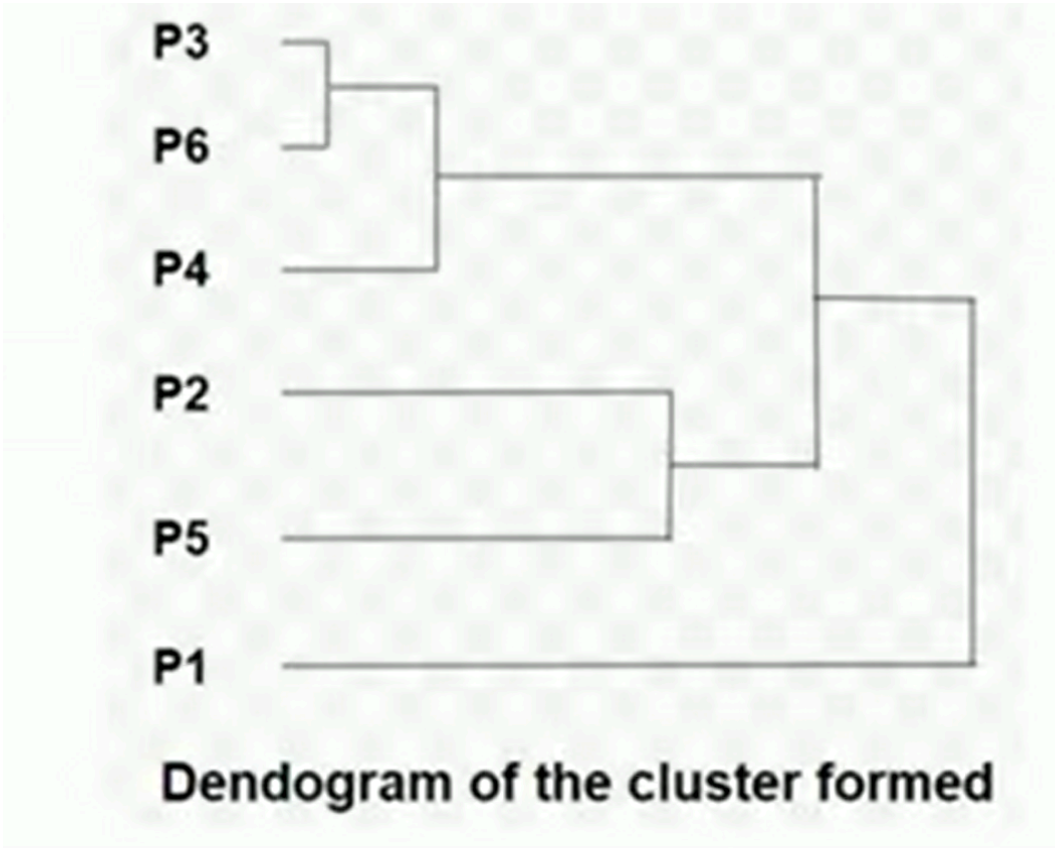
P2 and P5 made a cluster.

$$\begin{pmatrix} & P1 & P2, P5, P3, P6, P4 \\ P1 & 0 & \\ P2, P5, P3, P6, P4 & 0.22 & 0 \end{pmatrix}$$

[{(P3, P6), P4}, (P2, P5)]

[{(P3, P6), P4}, (P2, P5)] , P1

- Now draw the dendrogram



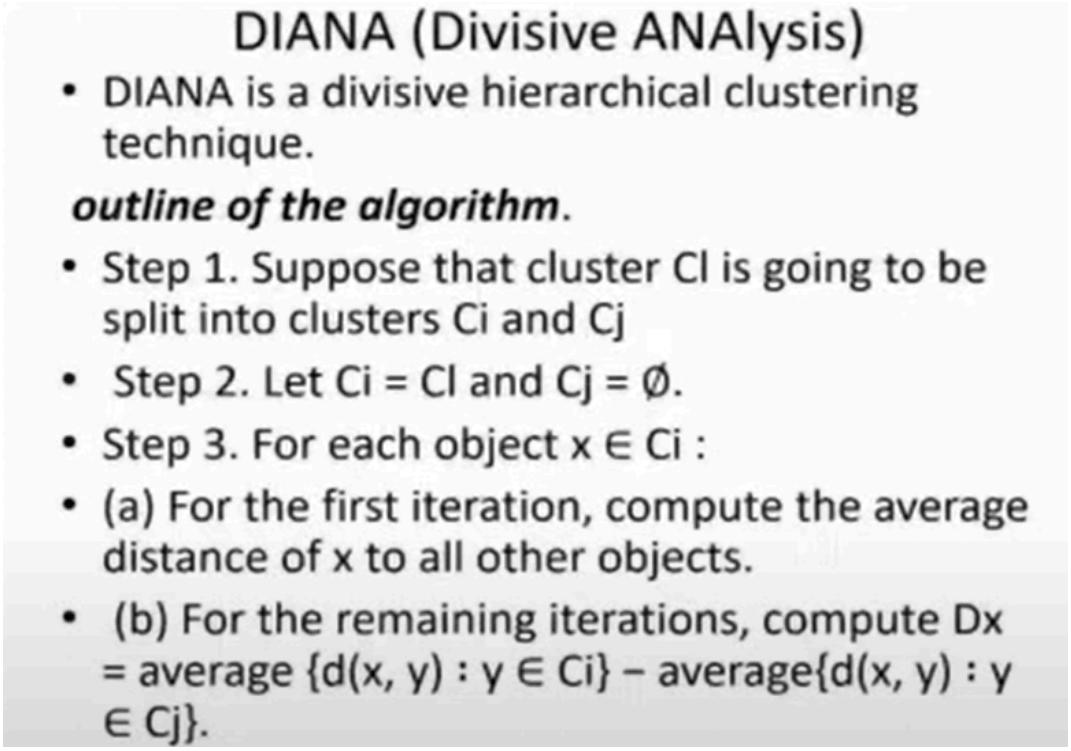
Dendogram of the cluster formed

- ***Do the same method for Complete linkage, but select maximum distances instead of minimum.***

Note :Nearest Neighbor Algorithm focuses on merging the closest data points iteratively, while Single Linkage Algorithm merges clusters based on the minimum distance between any two points in the clusters.

Divisive Hierarchical Clustering

- In the divisive or top-down clustering method we assign all of the observations to a single cluster and then partition the cluster to two least similar clusters.
- **DIANA (DIvisive ANAlysis), a divisive hierarchical clustering method.**
- Finally, we proceed recursively on each cluster until there is one cluster for each observation.
- There is evidence that divisive algorithms produce more accurate hierarchies than agglomerative algorithms in some circumstances but is conceptually more complex.



Step 4. (a) For the first iteration, move the object with the maximum average distance to C_j.

(b) For the remaining iterations, find an object x in C_i for which D_x is the largest. If $D_x > 0$ then move x to C_j.

Step 5. Repeat Steps 3(b) and 4(b) until all differences D_x are negative. Then C_l is split into C_i and C_j.

Step 6. Select the smaller cluster with the largest diameter. (The diameter of a cluster is the largest dissimilarity between any two of its objects.) Then divide this cluster, following Steps 1-5.

Step 7. Repeat Step 6 until all clusters contain only a single object.

Example: Consider the below distance matrix

	a	b	c	d	e
a	0	9	3	6	11
b	9	0	7	5	10
c	3	7	0	9	2
d	6	5	9	0	8
e	11	10	2	8	0

- **Step 1:** Initially $C_l = \{a;\ b;\ c;\ d;\ e\}$
- **Step 2:** $C_i = C_l \ \ and\ C_j = \varphi$
- **Step 3: Initial iteration**
- Let us calculate the average dissimilarities of the objects in C_i with the other objects in C_i.
- **Average dissimilarity of a**
- $a = \frac{1}{4} * \big(d(a,b) + d(a,c) + d(a,d) + d(a,e)\big)$
- $a = \frac{1}{4}(9 + 3 + 6 + 11) = \mathbf{7.25}$

	a	b	c	d	e
a	0	9	3	6	11
b	9	0	7	5	10
c	3	7	0	9	2
d	6	5	9	0	8
e	11	10	2	8	0

- Similarly, we have :
- Average dissimilarity of $b = 7.75$
- Average dissimilarity of $c = 5.25$
- Average dissimilarity of $d = 7.00$
- Average dissimilarity of $e = 7.75$
- The highest average distance is $\mathbf{7.75}$ and there are two corresponding objects.
- We choose one of them, $\boldsymbol{b}$, arbitrarily.

	a	b	c	d	e
a	0	9	3	6	11
b	9	0	7	5	10
c	3	7	0	9	2
d	6	5	9	0	8
e	11	10	2	8	0

- We move $\boldsymbol{b}$ to $\boldsymbol{C_j}$.
- We now have $C_i = \{a, c, d, e\}$ and $C_j = \emptyset \cup \{b\} = \{b\}$

- **Step 3: Remaining iterations** $C_i = \{a, c, d, e\}$ and $C_j = \{b\}$
- **(i) 2nd iteration.**
- $D_a = \frac{1}{3}\big(d(a,c) + d(a,d) + d(a,e)\big) - \frac{1}{1}(d(a,b)) = \frac{20}{3} - 9 = \mathbf{-2.33}$
- $D_c = \frac{1}{3}\big(d(c,a) + d(c,d) + d(c,e)\big) - \frac{1}{1}\big(d(c,b)\big) = \frac{14}{3} - 7 = \mathbf{-2.33}$
- $D_d = \frac{1}{3}(d(d,a) + d(d,c) + d(d,e)) - \frac{1}{1}(d(c,b)) = \frac{23}{3} - 7 = \mathbf{0.67}$
- $D_e = \frac{1}{3}\ (d(e,a) + d(e,c) + d(e,d)) - \frac{1}{1}(d(e,b)) = \frac{21}{3} - 7 = \mathbf{0}$
- D_d *is the largest and* $D_d > 0$.
- *So we move,* $\boldsymbol{d}$ ***to*** $\boldsymbol{C_j}$
- *We now have*
- $C_i = \{a, c, e\}$ and $C_j = \{b\} \cup \{d\} = \{b, d\}$

	a	b	c	d	e
a	0	9	3	6	11
b	9	0	7	5	10
c	3	7	0	9	2
d	6	5	9	0	8
e	11	10	2	8	0

- **(ii) 3rd iteration** $C_i = \{a, c, e\}$ and $C_j = \{b, d\}$
- $D_a = \frac{1}{2}\big(d(a,c) + d(a,e)\big) - \frac{1}{2}\big(d(a,b) + d(a,d)\big)$

- $D_a = \frac{14}{2} - \frac{15}{2} = \mathbf{-0.5}$
- $D_c = \frac{1}{2}\big(d(c,a) + d(c,e)\big) - \frac{1}{2}\big(d(c,b), +d(c,d)\big)$
- $D_c = \frac{5}{2} - \frac{16}{2} = \mathbf{-13.5}$
- $D_e = \frac{1}{2}\big(d(e,a) + d(e,c)\big) - \frac{1}{2}\big(d(e,b) + d(e,d)\big)$
- $D_e = \frac{13}{2} - \frac{18}{2} = \mathbf{-2.5}$
- **All are negative.** So we stop and form the clusters C_i and C_j .

- **Step 4:** $C_i = \{a, c, e\}$ *and* $C_j = \{b, d\}$
- To divide, C_i and C_j, we compute their diameters.
- $diameter(C_i) = \max\{d(a,c), d(a,e), d(c,e)\}$
- $diameter(C_i) = \max\{3, 11, 2\} =$ **11**
- $diameter(C_j) = \max\{d(b,d)\} =$ **5**
- The cluster with the largest diameter is $\boldsymbol{C_i}$.
- So we now split $\boldsymbol{C_i}$.
- **We repeat the process by taking C_l = {a, c, e}.**

	a	b	c	d	e
a	0	9	3	6	11
b	9	0	7	5	10
c	3	7	0	9	2
d	6	5	9	0	8
e	11	10	2	8	0

Decision Measures

'euclidean':	Usual square distance between the two vectors (2 norm).
'maximum':	Maximum distance between two components of x and y (supremum norm)
'manhattan':	Absolute distance between the two vectors (1 norm).
'canberra':	∑(|xi−yi|/|xi+yi|). Terms with zero numerator and denominator are omitted from the sum and treated as if the values were missing.
'minkowski':	The p norm, the pth root of the sum of the pth powers of the differences of the components.
'correlation':	1 - r where r is the Pearson or Spearman correlation
'absolute correlation'	1 - | r |

Density-based Clustering

Density-based clustering is an unsupervised learning method used to identify distinctive groups or clusters in data. It operates based on the concept that a cluster in a data space is a contiguous region of high point density, separated from other clusters by regions of low point density. Here are the key points about density-based clustering:

1. **Core Objects and ε-Neighborhood:**
    - In density-based clustering, we define two parameters:
        - Eps (ε): The maximum radius of the neighborhood.
        - MinPts: The minimum number of points in an Eps neighborhood of a point.
    - An object is considered a core object if its ε-neighborhood contains at least MinPts points.
    - The ε-neighborhood of an object consists of all points within a radius ε from that object.
2. Directly Density Reachable:
    - Point i is directly density reachable from point k with respect to ε and MinPts if i belongs to the ε-neighborhood of k.
    - A core object is directly density reachable from another core

object.

3. Density Reachable:
    - Point i is density reachable from point j if there is a sequence chain of points i1, i2, ..., in such that:
        - i1 = j
        - i1 is directly density reachable from i2
        - i2 is directly density reachable from i3
        - ...
        - in-1 is directly density reachable from in
        - in = i
4. Density Connected:
    - Point i is density connected to point j if there exists an object o such that both i and j are density reachable from o.

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

- DBSCAN is a popular density-based clustering algorithm.
- It identifies clusters of arbitrary size in a spatial database, including outliers.
- DBSCAN relies on the concept of density reachability and density connectivity.
- It is particularly useful for managing noise in data clusters.

- **DBSCAN defines two parameters: epsilon (ε),** which specifies the radius of the neighborhood around a point, and **MinPts,** which is the minimum number of points required to form a dense region. A core point is a point that has at least MinPts points (including itself) within its ε-neighborhood.
- **Border Points:** A border point is a point that is within the ε-neighborhood of a core point but does not have enough points in its own neighborhood to be considered a core point.
- **Noise Points:** Points that are neither core points nor border points are considered noise points.
- **Cluster Formation:** DBSCAN starts with an arbitrary point and retrieves all points in its ε-neighborhood. If the number of points is greater than or equal to MinPts, a cluster is formed. If not, the point is labeled as

noise. For each core point, all connected core points (directly or indirectly) form a single cluster. Border points may be part of a cluster but are not used to expand the cluster.

- **Result:** The algorithm continues this process until all points are assigned to a cluster or labeled as noise. The final result is a set of clusters and noise points.

Advantages

The advantages of density-based clustering, particularly DBSCAN, include its ability to handle clusters of arbitrary shapes, robustness to noise, and the ability to automatically determine the number of clusters. However, it requires careful parameter tuning for ε and MinPts, and it may struggle with clusters of varying densities.

Overall, density-based clustering methods are powerful techniques for discovering clusters in datasets with complex structures and noise, making them widely used in various domains such as spatial data analysis, anomaly detection, and pattern recognition.

Example:

Data Points:	
P1: (3, 7)	P2: (4, 6)
P3: (5, 5)	P4: (6, 4)
P5: (7, 3)	P6: (6, 2)
P7: (7, 2)	P8: (8, 4)
P9: (3, 3)	P10: (2, 6)
P11: (3, 5)	P12: (2, 4)

- Use Eucladian distance and calculate the distance between each points.

$$Distance(A(x_1, y_1), B(x_2, y_2)) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

minPts = 4 and epsilon (ε) = 1.9

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
P1	0											
P2	1.41	0										
P3	2.83	1.41	0									
P4	4.24	2.83	1.41	0								
P5	5.66	4.24	2.83	1.41	0							
P6	5.83	4.47	3.16	2.00	1.41	0						
P7	6.40	5.00	3.61	2.24	1.00	1.00	0					
P8	5.83	4.47	3.16	2.00	1.41	2.83	2.24	0				
P9	4.00	3.16	2.83	3.16	4.00	3.16	4.12	5.10	0			
P10	1.41	2.00	3.16	4.47	5.83	5.66	6.40	6.32	3.16	0		
P11	2.00	1.41	2.00	3.16	4.47	4.24	5.00	5.10	2.00	1.41	0	
P12	3.16	2.83	3.16	4.00	5.10	4.47	5.39	6.00	1.41	2.00	1.41	0

P1: P2, P10
P2: P1, P3, P11
P3: P2, P4
P4: P3, P5
P5: P4, P6, P7, P8
P6: P5, P7
P7: P5, P6
P8: P5
P9: P12
P10: P1, P11
P11: P2, P10, P12
P12: P9, P11

- Check and mark the clusters as CORE if it includes the minimum of 4 points(as per the minPts) and otherwise mark as NOISE. And if any of the noise points is a border point of any of the clusters, mark it as border.

P1: P2, P10
P2: P1, P3, P11
P3: P2, P4
P4: P3, P5
P5: P4, P6, P7, P8
P6: P5, P7
P7: P5, P6
P8: P5
P9: P12
P10: P1, P11
P11: P2, P10, P12
P12: P9, P11

minPts = 4 and epsilon (ε) = 1.9

Point	Status	
P1	Noise	Border
P2	Core	
P3	Noise	Border
P4	Noise	Border
P5	Core	
P6	Noise	Border
P7	Noise	Border
P8	Noise	Border
P9	Noise	
P10	Noise	Border
P11	Core	
P12	Noise	Border

P1: P2, P10

P2: P1, P3, P11

P3: P2, P4

P4: P3, P5

P5: P4, P6, P7, P8

P6: P5, P7

P7: P5, P6

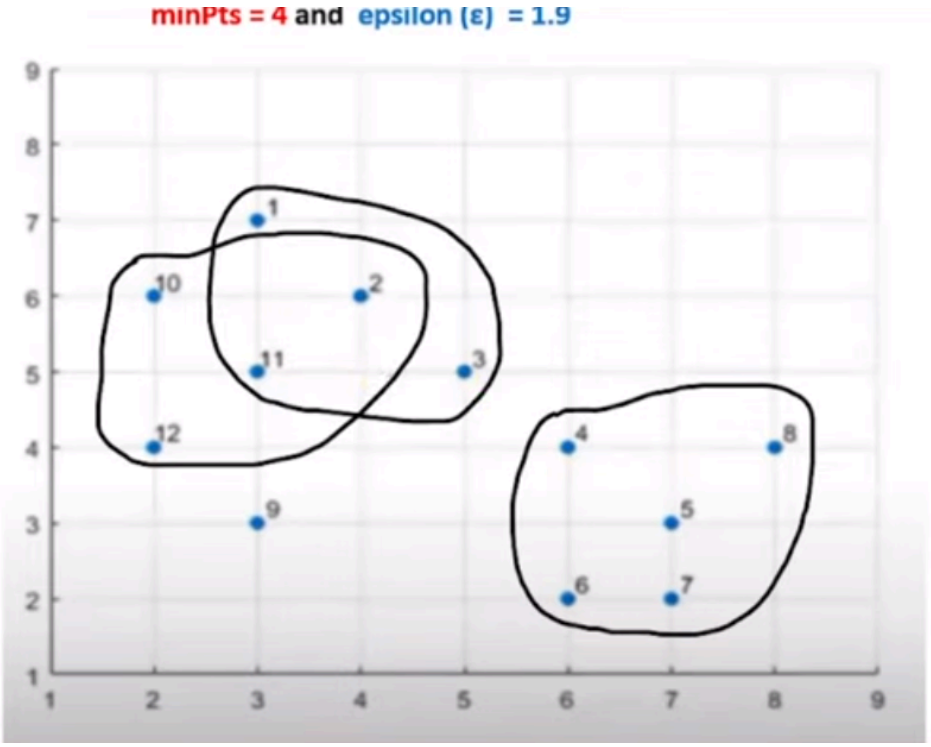
P8: P5

P9: P12

P10: P1, P11

P11: P2, P10, P12

P12: P9, P11



- Here the point 9, can be taken as outlier

Grid-Based Clustering

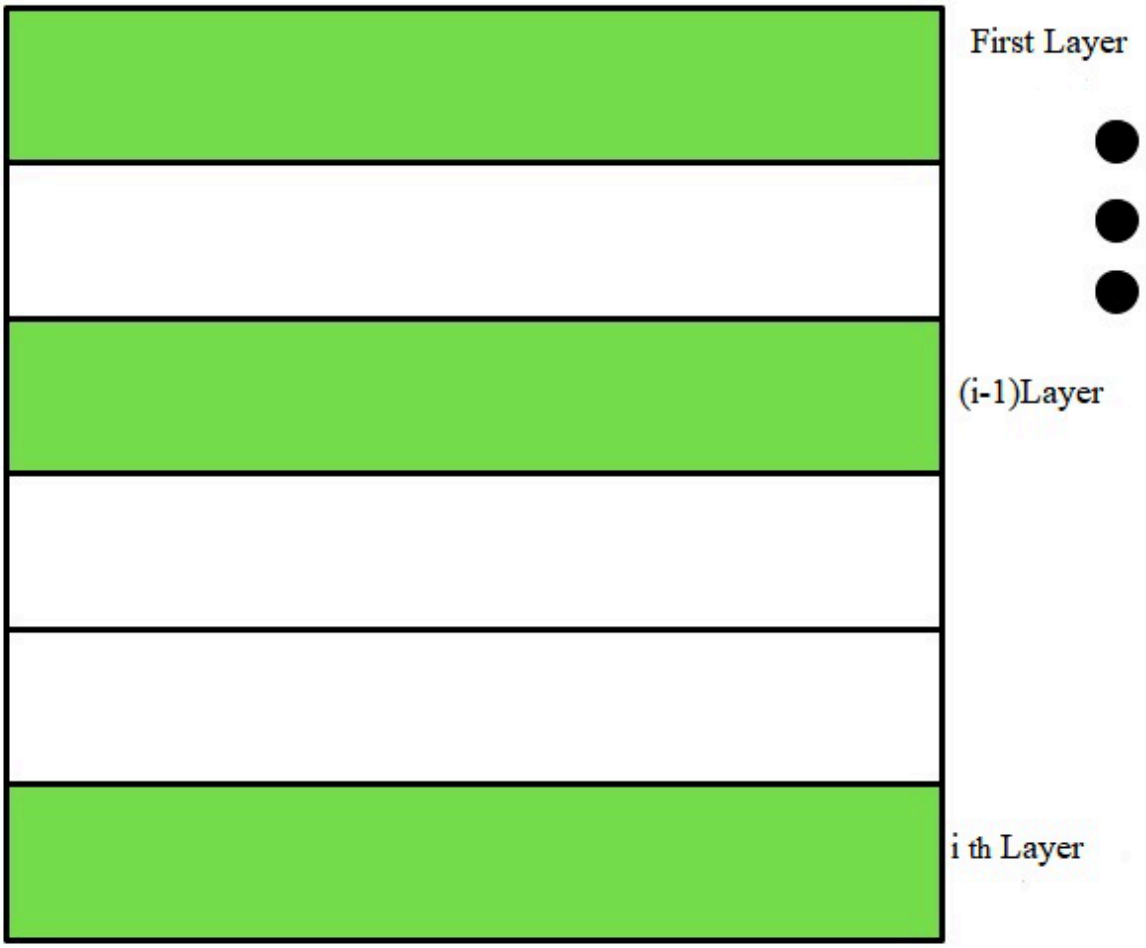
- In Grid-Based Methods, the space of instance is divided into a grid structure.
- Clustering techniques are then applied using the Cells of the grid, instead of individual data points, as the base units.
- The biggest advantage of this method is to improve the processing time.

Statistical Information Grid(STING):

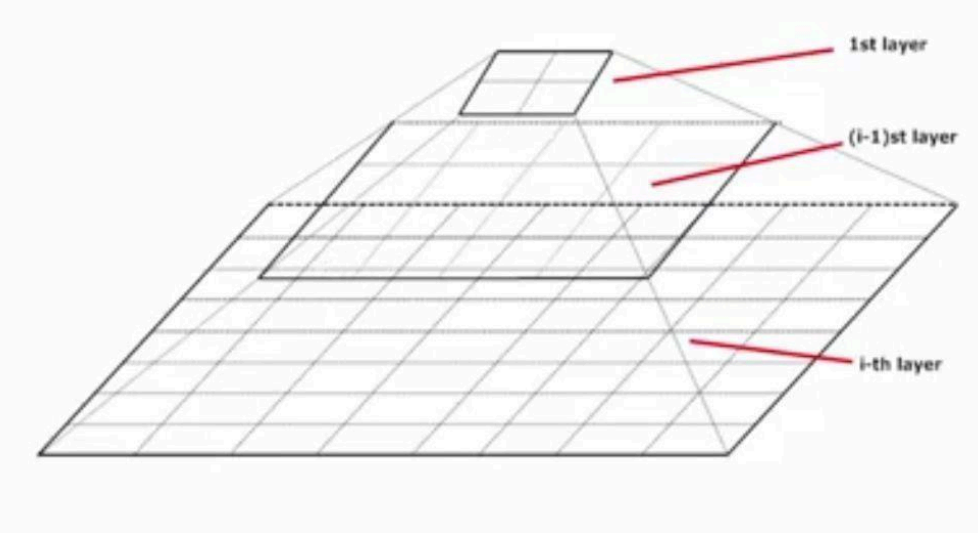
- A STING is a grid-based clustering technique. It uses a multidimensional grid data structure that quantifies space into a finite number of cells. Instead of focusing on data points, it focuses on the value space

surrounding the data points.

- In STING, the spatial area is divided into rectangular cells and several levels of cells at different resolution levels. High-level cells are divided into several low-level cells.
- In STING Statistical Information about attributes in each cell, such as mean, maximum, and minimum values, are precomputed and stored as statistical parameters. These statistical parameters are useful for query processing and other data analysis tasks.



The statistical parameter of higher-level cells can easily be computed from the parameters of the lower-level cells.



How STING Work:

Step 1: Determine a layer, to begin with.

Step 2: For each cell of this layer, it calculates the confidence interval or estimated range of probability that this cell is relevant to the query.

Step 3: From the interval calculated above, it labels the cell as relevant or not relevant.

Step 4: If this layer is the bottom layer, go to point 6, otherwise, go to point 5.

Step 5: It goes down the hierarchy structure by one level. Go to point 2 for those cells that form the relevant cell of the high-level layer.

Step 6: If the specification of the query is met, go to point 8, otherwise go to point
Step 7: Retrieve those data that fall into the relevant cells and do further processing. Return the result that meets the requirement of the query. Go to point 9.

Step 8: Find the regions of relevant cells. Return those regions that meet the requirement of the query. Go to point 9.

Step 9: Stop or terminate.

Advantages:

- Grid-based computing is query-independent because the statistics stored in each cell represent a summary of the data in the grid cells and are query-independent.
- The grid structure facilitates parallel processing and incremental updates.