Clustering in Data Mining

Clustering is an unsupervised Machine Learning-based Algorithm that comprises a group of data points into clusters so that the objects belong to the same group.

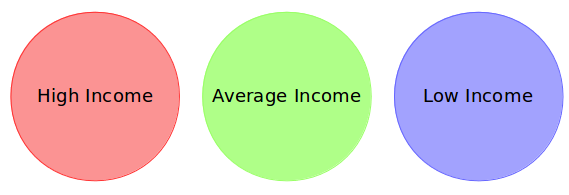
Clustering helps to splits data into several subsets. Each of these subsets contains data similar to each other, and these subsets are called clusters.

Example

A bank wants to give credit card offers to its customers. Currently, they look at the details of each customer and, based on this information, decide which offer should be given to which customer.

Now, the bank can potentially have millions of customers. Does it make sense to look at the details of each customer separately and then make a decision? Certainly not! It is a manual process and will take a huge amount of time.

So what can the bank do? One option is to segment its customers into different groups. For instance, the bank can group the customers based on their income:

[](https://cdn.analyticsvidhya.com/wp-content/uploads/2019/08/Screenshot-from-2019-08-07-15-19-27.png)

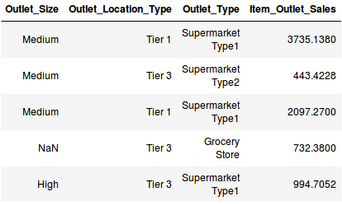
The bank can now make three different strategies or offers, one for each group. Here, instead of creating different strategies for individual customers, they only have to make 3 strategies. This will reduce the effort as well as the time.

**The groups are known as clusters, and the process of creating these groups is known as clustering.** Formally, we can say that:

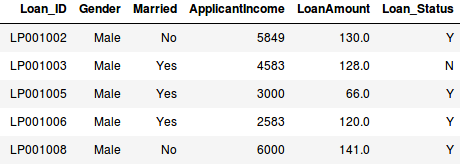
*Clustering is the process of dividing the entire data into groups (also known as clusters) based on the patterns in the data.*

How is Clustering an Unsupervised Learning Problem?

Let’s say you are working on a project where you need to predict the sales of a big mart:

[](https://cdn.analyticsvidhya.com/wp-content/uploads/2019/08/Screenshot-from-2019-08-07-16-25-31.png)

Or, a project where your task is to predict whether a loan will be approved or not:

[](https://cdn.analyticsvidhya.com/wp-content/uploads/2019/08/Screenshot-from-2019-08-07-16-29-45.png)

We have a fixed target to predict in both of these situations. In the sales prediction problem, we have to predict the *Item\_Outlet\_Sales* based on *outlet\_size, outlet\_location\_type*, etc., and in the loan approval problem, we have to predict the *Loan\_Status* depending on the *Gender, marital status, the income of the customers, etc*.

So, when we have a target variable to predict based on a given set of predictors or independent variables, such problems are called supervised learning problems.

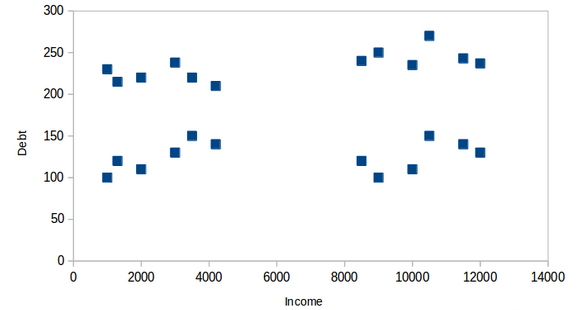
Now, there might be situations where we do *not* have any target variable to predict.

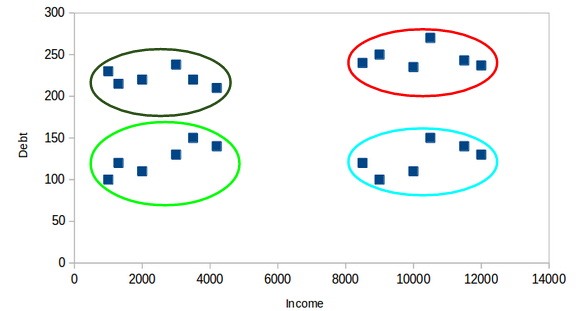
Such problems, without any fixed target variable, are known as unsupervised learning problems. In these problems, we only have the independent variables and no target/dependent variable.

**In clustering, we do not have a target to predict. We look at the data, try to club similar observations, and form different groups. Hence it is an unsupervised learning problem.**

Properties of Clusters

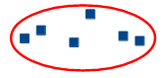
How about another example? We’ll take the same bank as before, which wants to segment its customers. For simplicity purposes, let’s say the bank only wants to use the income and debt to make the segmentation. They collected the customer data and used a scatter plot to visualize it:

[](https://cdn.analyticsvidhya.com/wp-content/uploads/2019/08/Screenshot-from-2019-08-08-14-46-17.png)  
On the X-axis, we have the income of the customer, and the y-axis represents the amount of debt. Here, we can clearly visualize that these customers can be segmented into 4 different clusters, as shown below:

[](https://cdn.analyticsvidhya.com/wp-content/uploads/2019/08/Screenshot-from-2019-08-08-14-47-20.png)  
This is how clustering helps to create segments (clusters) from the data. The bank can further use these clusters to make strategies and offer discounts to its customers. So let’s look at the properties of these clusters.

First Property of K-Means Clustering Algorithm

**All the data points in a cluster should be similar to each other.**

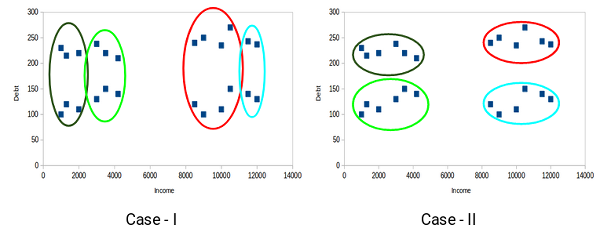
Example:[](https://cdn.analyticsvidhya.com/wp-content/uploads/2019/08/Screenshot-from-2019-08-08-14-49-11.png)

If the customers in a particular cluster are not similar to each other, then their requirements might vary, right? If the bank gives them the same offer, they might not like it, and their interest in the bank might reduce. Not ideal.

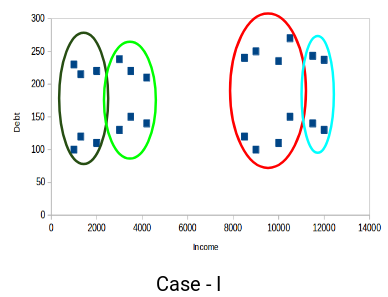
Having similar data points within the same cluster helps the bank to use targeted marketing. You can think of similar examples from your everyday life and consider how clustering will (or already does) impact the business strategy.

Second Property of K-Means Clustering Algorithm

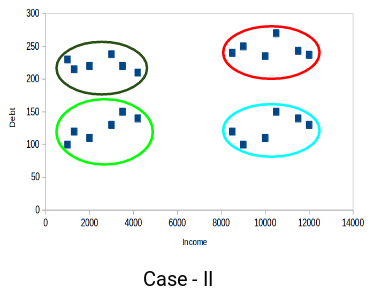
**The data points from different clusters should be as different as possible.** This will intuitively make sense if you’ve grasped the above property. Let’s again take the same example to understand this property:

[](https://cdn.analyticsvidhya.com/wp-content/uploads/2019/08/Screenshot-from-2019-08-08-14-51-31.png)

Which of these cases do you think will give us the better clusters? If you look at case I:

[](https://cdn.analyticsvidhya.com/wp-content/uploads/2019/08/Screenshot-from-2019-08-08-14-52-26.png)

Customers in the red and blue clusters are quite similar to each other. The top four points in the red cluster share similar properties to those of the blue cluster’s top two customers. They have high incomes and high debt values. Here, we have clustered them differently. Whereas, if you look at case II:

[](https://cdn.analyticsvidhya.com/wp-content/uploads/2019/08/Screenshot-from-2019-08-08-14-52-58.png)

Points in the red cluster completely differ from the customers in the blue cluster. All the customers in the red cluster have high income and high debt, while the customers in the blue cluster have high income and low debt value. Clearly, we have a better clustering of customers in this case.

Hence, data points from different clusters should be as different from each other as possible to have more meaningful clusters. The k-means algorithm uses an iterative approach to find the optimal cluster assignments by minimizing the sum of squared distances between data points and their assigned cluster centroid.

So far, we have understood what clustering is and the different properties of clusters. But why do we even need clustering? Let’s clear this doubt in the next section and look at some applications of clustering.

Applications of Clustering in Real-World Scenarios

Clustering is a widely used technique in the industry. It is being used in almost every domain, from banking and [recommendation engines](https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-recommendation-engine-python/?utm_source=blog&utm_medium=comprehensive-guide-k-means-clustering) to document clustering and image segmentation.

**Customer Segmentation**

One of the most common applications of clustering is customer segmentation. And it isn’t just limited to banking. This strategy is across functions, including telecom, e-commerce, sports, advertising, sales, etc.

**Document Clustering**

This is another common application of clustering. Let’s say you have multiple documents and you need to cluster similar documents together. Clustering helps us group these documents such that similar documents are in the same clusters.

[](https://cdn.analyticsvidhya.com/wp-content/uploads/2019/08/Screenshot-from-2019-08-08-15-02-44.png)

Image Segmentation

We can also use clustering to perform image segmentation. Here, we try to club similar pixels in the image together. We can apply clustering to create clusters having similar pixels in the same group.

[](https://cdn.analyticsvidhya.com/wp-content/uploads/2019/08/Screenshot-from-2019-08-08-15-03-22.png)

**Similarity and Distance measure in data mining**

The similarity measures in data mining, is a distance with dimensions describing object features. This means that in case the distance among two data points is small then there is a high degree of similarity among the objects and vice versa.

The similarity measure in data science is a way of measuring how data samples are related or close to each other. A dissimilarity measure is used to figure out how much the data objects are distinct.

These terms are generally used in clustering when similar data samples are grouped into one cluster. All other data samples are grouped into different ones.

S**imilarity measures formulas:**

**Euclidean Distance**

**Manhattan Distance (City Block Distance)**

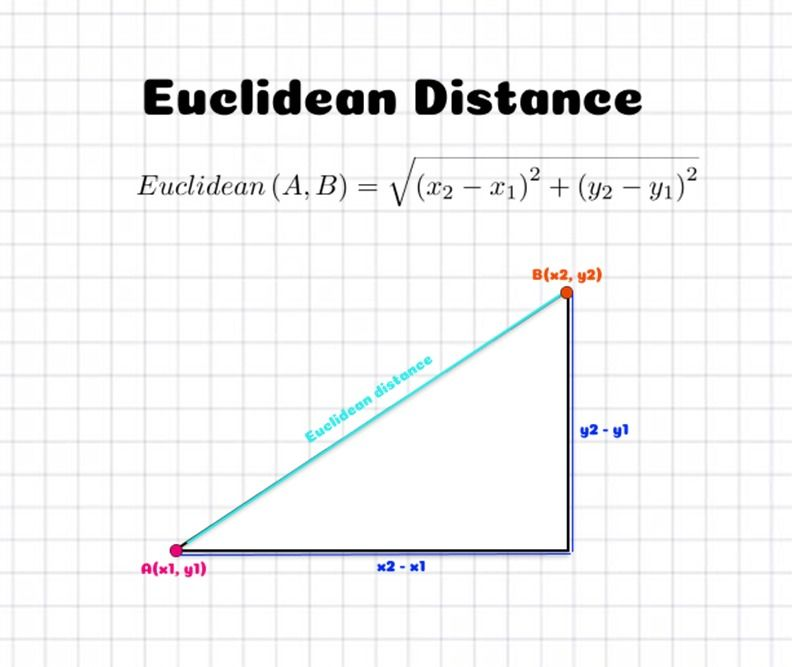
**Cosine similarity**

**Jaccard similarity**

**Minkowski distance**

**Euclidean Distance**

[Euclidean distance](https://www.engati.com/glossary/euclidean-distance) is widely used as the traditional metric to work on problems with geometry. You can explain it as the ordinary distance between two points. It happens to be one of the most widely used algorithms in cluster analysis. If you’d like an example of an algorithm that makes use of this formula, just look at the K-Means clustering algorithm. Mathematically it will compute the root of squared differences between the coordinates between two objects.



**Manhattan Distance (City Block Distance)**

The Manhattan Distance or City Block Distance determines the absolute difference among the pair of the coordinates. It might be surprising, but the simplest way of calculating the distance between two points is to go horizontally and then vertically until you get from one point to the other, instead of just going in a straight line. It is a simpler technique since it only needs you to subtract instead of performing more complicated calculations.

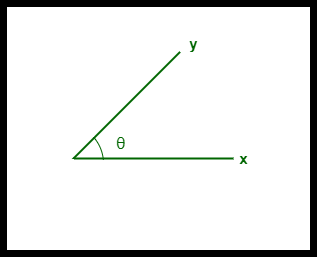
**Cosine similarity**

Cosine similarity is a widely used similarity measure in data mining and information retrieval. It measures the cosine of the angle between two non-zero vectors in a multi-dimensional space. In the context of data mining, these vectors represent the feature vectors of two data points. The cosine similarity score ranges from 0 to 1, with 0 indicating no similarity and 1 indicating perfect similarity.

The cosine similarity between two vectors is calculated as the dot product of the vectors divided by the product of their magnitudes. This calculation can be represented mathematically as follows –

cos(*θ*)=**A.B** / ∥**A** ∥∥**B**∥where **A** and **B** are the feature vectors of two data points, "." denotes the dot product, and

"||" denotes the magnitude of the vector

. 

**Jaccard Similarity**

The Jaccard similarity is another widely used similarity measure in data mining, particularly in text analysis and clustering. It measures the similarity between two sets of data by calculating the ratio of the intersection of the sets to their union. The Jaccard similarity score ranges from 0 to 1, with 0 indicating no similarity and 1 indicating perfect similarity.

The Jaccard similarity between two sets A and B is calculated as follows -

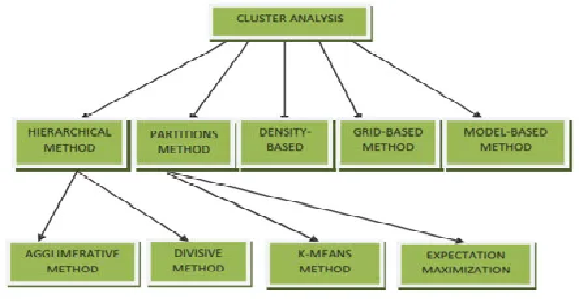
*J*(*A*,*B*)=∣**A**∩**B**∣ / ∣**A**∪**B**∣

where ∣**A**∩**B**∣ is the size of the intersection of sets **A** and **B**, and ∣**A**∪**B**∣ is the size of the union of sets **A** and **B**.

**Minkowski distance**

The Minkowski distance is the generalized form of the Euclidean and Manhattan Distance Measure.

**Data Mining Clustering Methods**

****

Let’s take a look at different types of clustering in data mining!

**1. Partitioning Clustering Method**

In this method, let us say that “m” partition is done on the “p” objects of the database. A cluster will be represented by each partition and m < p. K is the number of groups after the classification of objects. There are some requirements which need to be satisfied with this Partitioning Clustering Method .

they are:

1. One object should only belong to only one group.
2. There should be no group without even a single purpose.
3. There will be an initial partitioning if we already give no. of a partition (say m).

**2. Hierarchical Clustering Methods**

the given set of an object of data is created into a kind of hierarchical decomposition. The formation of hierarchical decomposition will decide the purposes of classification.

There are two types of approaches for the creation of hierarchical decomposition:

**1. Divisive Approach**

Another name for the Divisive approach is a top-down approach. At the beginning of this method, all the data objects are kept in the same cluster. Smaller clusters are created by splitting the group by using the continuous iteration. The constant iteration method will keep on going until the condition of termination is met. One cannot undo after the group is split or merged, and that is why this method is not so flexible.

**2. Agglomerative Approach**

Another name for this approach is the bottom-up approach. All the groups are separated in the beginning. Then it keeps on merging until all the groups are merged, or condition of termination is met.

**3. Density-Based Clustering Method**

In this method of clustering in Data Mining, density is the main focus. The notion of mass is used as the basis for this clustering method. In this clustering method, the cluster will keep on growing continuously. At least one number of points should be there in the radius of the group for each point of data.

**4. Grid-Based Clustering Method**

In this type of Grid-Based Clustering Method, a grid is formed using the object together. A Grid Structure is formed by quantifying the object space into a finite number of cells.

**5. Model-Based Clustering Methods**

In this type of clustering method, every cluster is hypothesized so that it can find the data which is best suited for the model. The density function is clustered to locate the group in this method.

**6. Constraint-Based Clustering Method**

Application or user-oriented constraints are incorporated to perform the clustering. The expectation of the user is referred to as the constraint. In this process of grouping, communication is very interactive, which is provided by the restrictions.

**OUTLIERS**

We all have heard of the idiom ‘odd one out ‘ which means something unusual in comparison to the others in a group. Similarly, an Outlier is an observation in a given dataset that lies far from the rest of the observations. That means an outlier is vastly larger or smaller than the remaining values in the set.

An outlier may occur due to the variability in the data, or due to experimental error/human error.

In statistics, we have three measures of central tendency namely Mean, Median, and Mode. They help us describe the data.

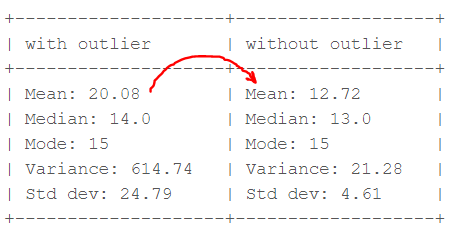
* Mean is the accurate measure to describe the data when we do not have any outliers present.
* Median is used if there is an outlier in the dataset.
* Mode is used if there is an outlier AND about ½ or more of the data is the same.

Mean’ is the only measure of central tendency that is affected by the outliers which in turn impacts Standard deviation.

Example

Consider a small dataset, sample=[15, 101, 18, 7, 13, 16, 11, 21, 5, 15, 10, 9]. By looking at it, one can quickly say ‘101’ is an outlier that is much larger than the other values.

**Computation with and without outlier**



From the above calculations, we can clearly say the Mean is more affected than the Median.

Detecting Outliers

If our dataset is small, we can detect the outlier by just looking at the dataset. But what if we have a huge dataset, how do we identify the outliers then? We need to use visualization and mathematical techniques.

Below are some of the techniques of detecting outliers

* Boxplots
* Z-score
* Inter Quantile Range(IQR)

Hierarchical clustering in data mining

Hierarchical clustering refers to an unsupervised learning procedure that determines successive clusters based on previously defined clusters. It works via grouping data into a tree of clusters. Hierarchical clustering treats each data point as an individual cluster. The endpoint refers to a different set of clusters, where each cluster is different from the other cluster, and the objects within each cluster are the same as one another.

There are two types of hierarchical clustering

* Agglomerative Hierarchical Clustering
* Divisive Clustering

Agglomerative hierarchical clustering

Agglomerative clustering is one of the most common types of hierarchical clustering used to group similar objects in clusters. Agglomerative clustering is also known as AGNES (Agglomerative Nesting). In agglomerative clustering, each data point act as an individual cluster and at each step, data objects are grouped in a bottom-up method. Initially, each data object is in its cluster. At each iteration, the clusters are combined with different clusters until one cluster is formed.

**Agglomerative Clustering Algorithm**

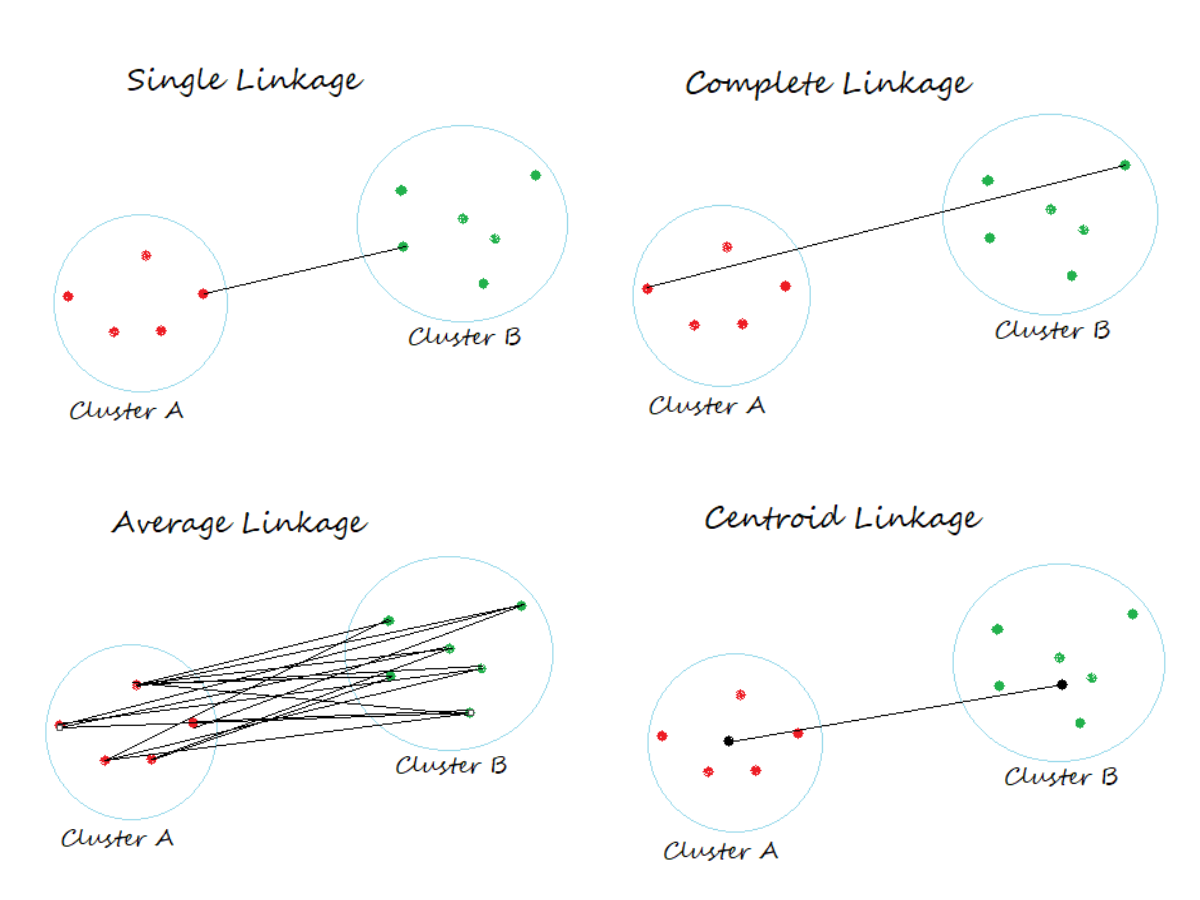
Following are the steps in agglomerative clustering.

1. We start by assigning each data point to its own cluster.
2. Next, we compute the distance between each pair of clusters and select the pair of clusters with the smallest distance.
3. Then, we merge the pair of clusters with the smallest distance into a single cluster and update the distance between the newly formed cluster and every other cluster.
4. We repeat steps 2 and 3 until all data points are in one cluster.

**Calculation of Distance Between Two Clusters**

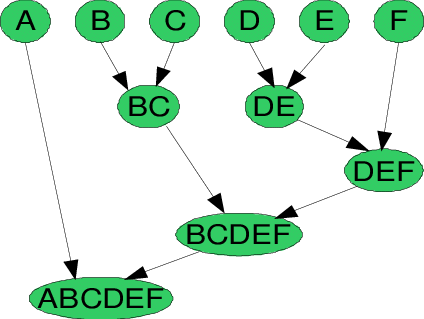
The distance between clusters in agglomerative clustering can be calculated using three approaches namely single linkage, complete linkage, and average linkage.

* In the **single linkage approach**, we take the distance between the nearest points in two clusters as the distance between the clusters.
* In the **complete linkage approach**, we take the distance between the farthest points in two clusters as the distance between the clusters.
* In the **average linkage approach**, we take the average distance between each pair of points in two given clusters as the distance between the clusters. You can also take the distance between the centroids of the clusters as their distance from each other.



To understand better let’s see a pictorial representation of the Agglomerative Hierarchical clustering Technique.Let us say we have six data points {A,B,C,D,E,F}.

* Step- 1: In the initial step, we calculate the proximity of individual points and consider all the six data points as individual clusters as shown in the image below.

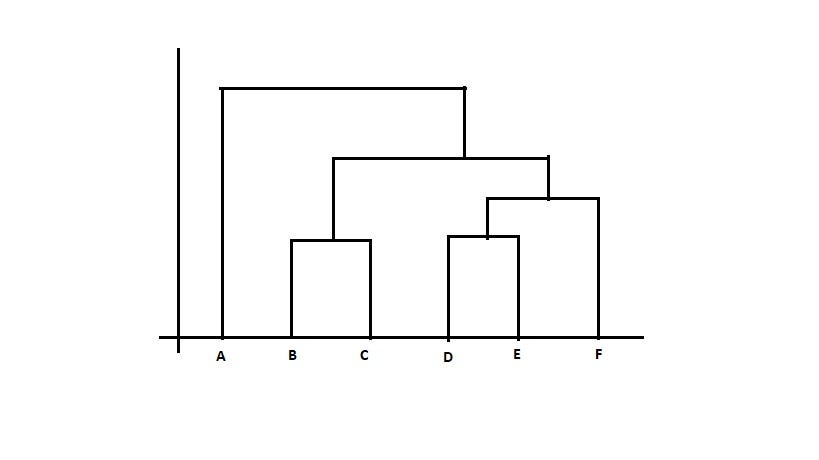


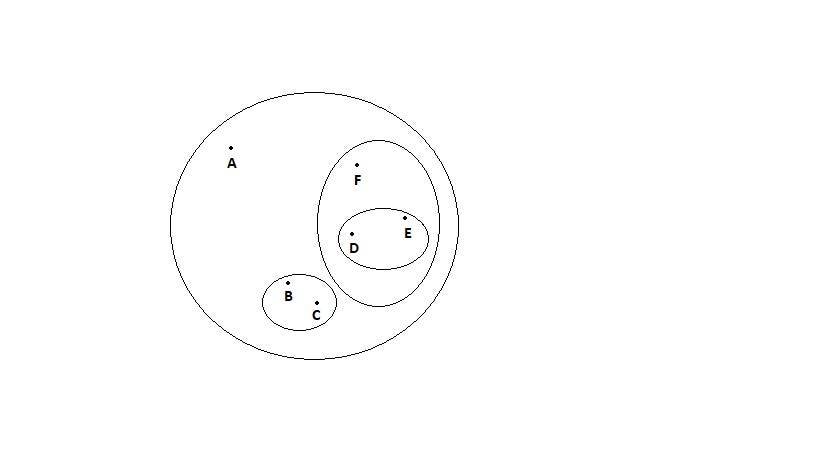
Agglomerative Hierarchical Clustering Technique

* Step- 2: In step two, similar clusters are merged together and formed as a single cluster. Let’s consider B,C, and D,E are similar clusters that are merged in step two. Now, we’re left with four clusters which are A, BC, DE, F.
* Step- 3: We again calculate the proximity (near to something)of new clusters and merge the similar clusters to form new clusters A, BC, DEF.
* Step- 4: Calculate the proximity of the new clusters. The clusters DEF and BC are similar and merged together to form a new cluster. We’re now left with two clusters A, BCDEF.
* Step- 5: Finally, all the clusters are merged together and form a single cluster.

**Dendrograms** are used to represent hierarchical clustering results.

A**Dendrogram**is atree-like diagram that records the sequences of merges or splits.





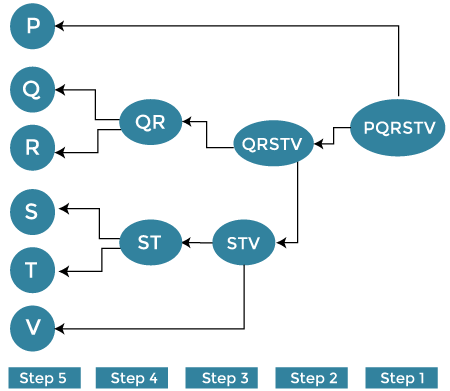
Dia: Dendrogram representation

Divisive Hierarchical Clustering

Divisive hierarchical clustering is exactly the opposite of Agglomerative Hierarchical clustering. In Divisive Hierarchical clustering, all the data points are considered an individual cluster, and in every iteration, the data points that are not similar are separated from the cluster. The separated data points are treated as an individual cluster. Finally, we are left with N clusters.

Algorithm:

1. Compute a Minimum Spanning Tree(MST) for the given adjacency matrix
2. Repeat
3. Create a new cluster by breaking the link corresponding to the largest distance.
4. Until only single cluster remains.



Advantages of Hierarchical clustering

* It is simple to implement and gives the best output in some cases.
* It is easy and results in a hierarchy, a structure that contains more information.
* It does not need us to pre-specify the number of clusters.

Disadvantages of hierarchical clustering

* It breaks the large clusters.
* It is Difficult to handle different sized clusters and convex shapes.
* It is sensitive to noise and outliers.
* The algorithm can never be changed or deleted once it was done previously.

**PARTITION METHOD:**

**K-Means Clustering**

K-means clustering is a method for grouping n observations into K clusters. It uses vector quantization and aims to assign each observation to the cluster with the nearest mean or centroid, which serves as a prototype for the cluster. Originally developed for signal processing, K-means clustering is now widely used in machine learning to partition data points into K clusters based on their similarity. The goal is to minimize the sum of squared distances between the data points and their corresponding cluster centroids, resulting in clusters that are internally homogeneous and distinct from each other.

Recall the first property of clusters – it states that the points within a cluster should be similar to each other. So,**our aim here is to minimize the distance between the points within a cluster.**

K-means is a centroid-based algorithm or a distance-based algorithm, where we calculate the distances to assign a point to a cluster. In K-Means, each cluster is associated with a centroid.

***The main objective of the K-Means algorithm is to minimize the sum of distances between the points and their respective cluster centroid.***

Optimization plays a crucial role in the k-means clustering algorithm. The goal of the optimization process is to find the best set of centroids that minimizes the sum of squared distances between each data point and its closest centroid. This process is repeated multiple times until convergence, resulting in the optimal clustering solution.

How to Apply K-Means Clustering Algorithm?

Let’s now take an example to understand how K-Means actually works:

[](https://cdn.analyticsvidhya.com/wp-content/uploads/2019/08/Screenshot-from-2019-08-09-12-21-43.png)

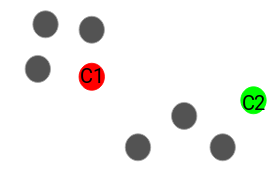
Time needed: 10 minutes

We have these 8 points, and we want to apply k-means to create clusters for these points.

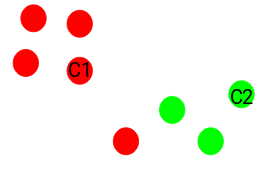
**Choose the number of clusters *k***

The first step in k-means is to pick the number of clusters, k.

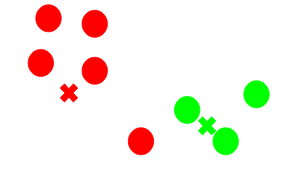
1. **Select k random points from the data as centroids**

Next, we randomly select the centroid for each cluster. Let’s say we want to have 2 clusters, so k is equal to 2 here. We then randomly select the centroid:  
  
Here, the red and green circles represent the centroid for these clusters.

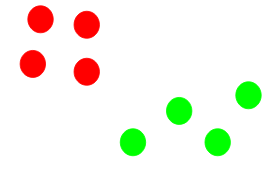
1. **Assign all the points to the closest cluster centroid**

Once we have initialized the centroids, we assign each point to the closest cluster centroid:  
Here you can see that the points closer to the red point are assigned to the red cluster, whereas the points closer to the green point are assigned to the green cluster.

1. **Recompute the centroids of newly formed clusters**

Now, once we have assigned all of the points to either cluster, the next step is to compute the centroids of newly formed clusters:  
  
Here, the red and green crosses are the new centroids.

1. **Repeat steps 3 and 4**

We then repeat steps 3 and 4:  
  
*The step of computing the centroid and assigning all the points to the cluster based on their distance from the centroid is a single iteration*. But wait – when should we stop this process? It can’t run till eternity, right?

Stopping Criteria for K-Means Clustering

There are essentially three stopping criteria that can be adopted to stop the K-means algorithm:

1. Centroids of newly formed clusters do not change
2. Points remain in the same cluster
3. Maximum number of iterations is reached

We can stop the algorithm if the centroids of newly formed clusters are not changing. Even after multiple iterations, if we are getting the same centroids for all the clusters, we can say that the algorithm is not learning any new pattern, and it is a sign to stop the training.

Another clear sign that we should stop the training process is if the points remain in the same cluster even after training the algorithm for multiple iterations.

Finally, we can stop the training if the maximum number of iterations is reached. Suppose we have set the number of iterations as 100. The process will repeat for 100 iterations before stopping.