UNIT – III

Un-Supervised Learning

Un-supervised Learning

- Unlike supervised learning, in unsupervised learning, there is **no labelled training data** to **learn from** and **no prediction to be made.**
- The main aim of the **unsupervised learning** is **to group or categories the unsorted dataset** according to **the similarities, patterns, and differences.** Machines are **instructed** to find the **hidden patterns** from the **input dataset.**
- In unsupervised learning, the models are **trained with the data** that is **neither classified nor labelled,** and the model acts on **that data without any supervision.**

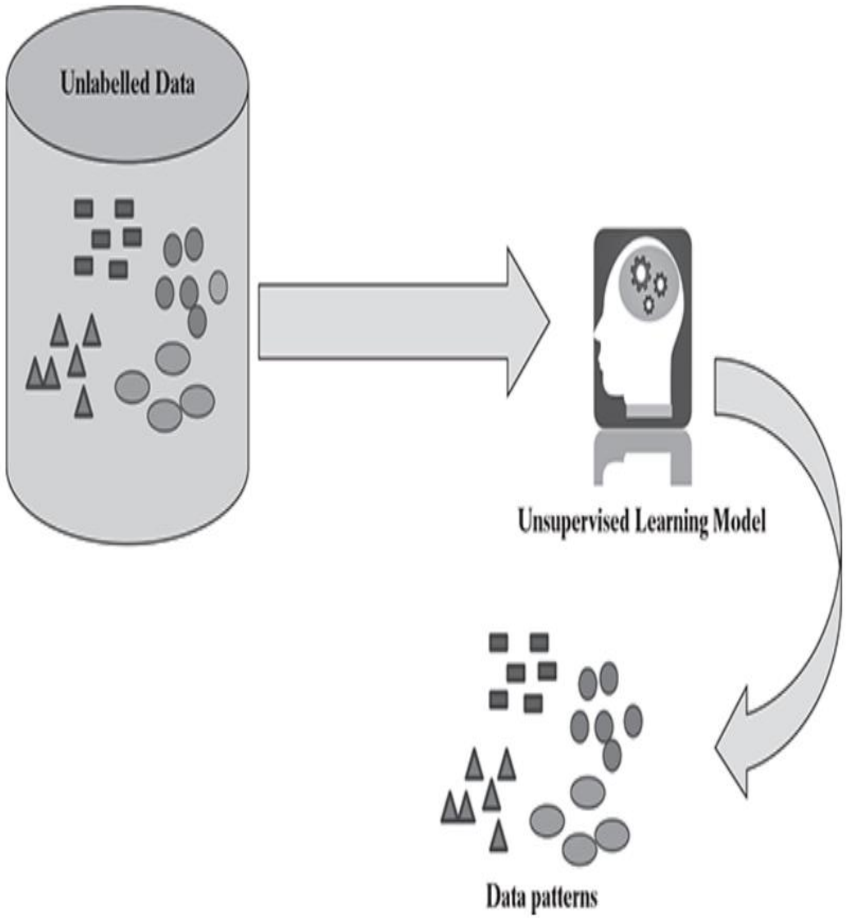


FIG. 1.8 Unsupervised learning

Unsupervised Learning

- Unsupervised learning is a **machine learning** concept where the **un-labelled** and **unclassified information** is **analyzed** to **discover hidden knowledge/patterns**
- The algorithms work on the **data without** any **prior training**, but they are constructed in such a way that they **can identify patterns, groupings, sorting order,** and **numerous other interesting knowledge** from **the set of data.**

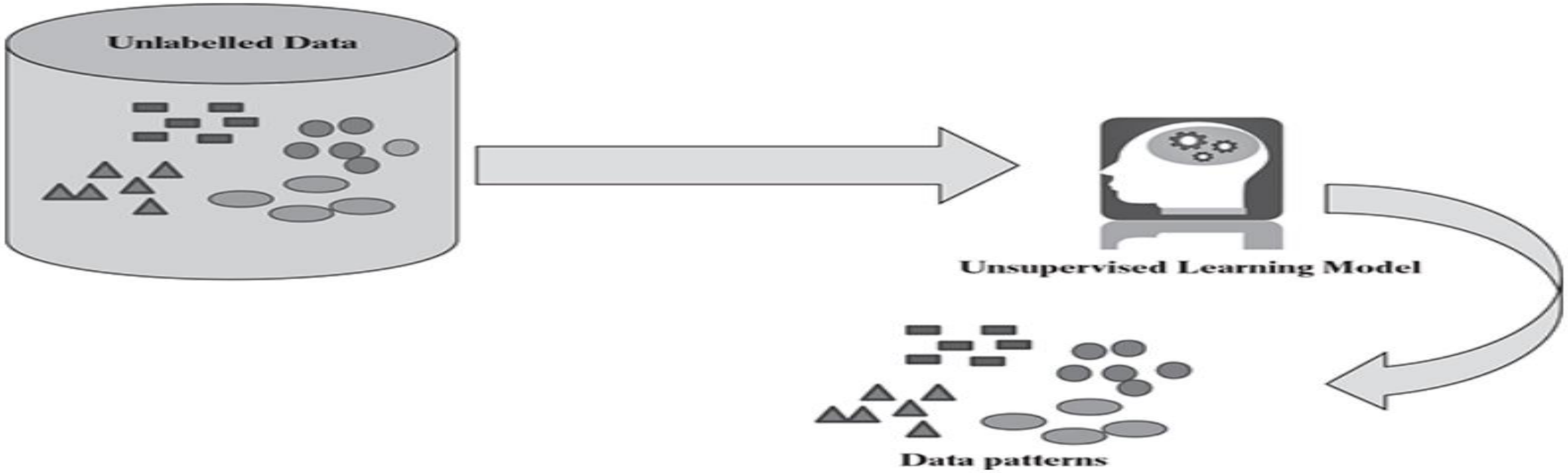


FIG. 1.8 Unsupervised learning

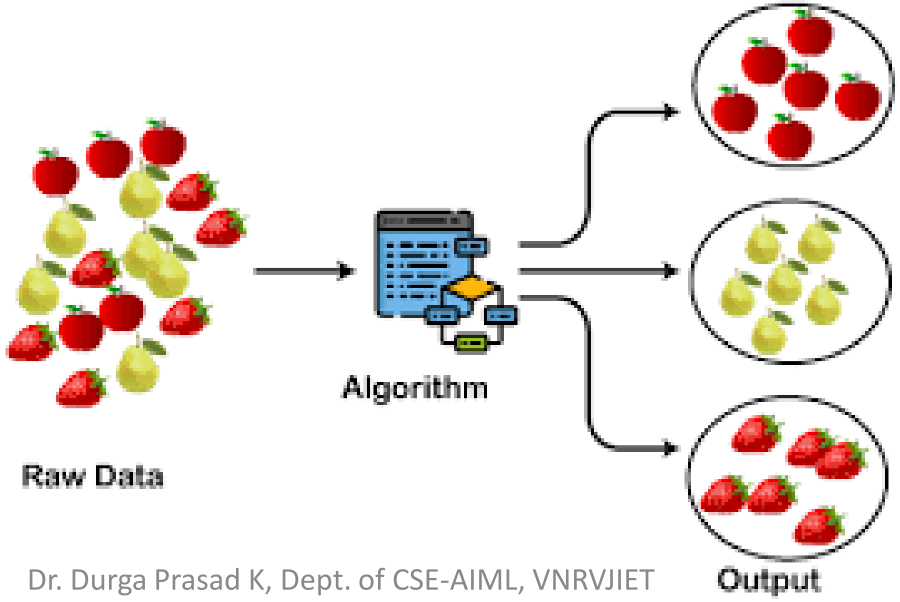
Unsupervised Learning Algorithms

1. **Clustering**
2. **Association Analysis**

Unsupervised Learning Algorithms

1. Clustering

- Clustering is the main type of **unsupervised learning**.
- It intends to group or organize **similar objects** or **data points** together.
- For that reason, objects belonging to the **same cluster** are **quite similar** to **each other** while objects belonging to **different clusters** are **quite dissimilar**



Unsupervised Learning Algorithms

2. Association Analysis

- An Association Analysis or Association rule is an unsupervised learning method used to find relationships between different items in a dataset.
- It determines the set of items that occurs together in the dataset.*
- Association rule makes **marketing strategy** more effective.
- Such as people who buy X item (suppose a bread) are also tend to purchase Y (Butter/Jam) item.
- A typical example of Association rule is Market Basket Analysis.

Unsupervised Vs Supervised Learning

SUPERVISED	UNSUPERVISED
This type of learning is used when you know how to classify a given data, or in other words classes or labels are available.	This type of learning is used when there is no idea about the class or label of a particular data. The model has to find pattern in the data.
Labelled training data is needed. Model is built based on training data.	Any unknown and unlabelled data set is given to the model as input and records are grouped.
The model performance can be evaluated based on how many misclassifications have been done based on a comparison between predicted and actual values.	Difficult to measure whether the model did something useful or interesting. Homogeneity of records grouped together is the only measure.
There are two types of supervised learning problems – classification and regression.	There are two types of unsupervised learning problems – clustering and association.
Simplest one to understand.	More difficult to understand and implement than supervised learning.

Unsupervised Vs Supervised Learning

Standard algorithms include

- Naïve Bayes
- *k*-nearest neighbour (kNN)
- Decision tree
- Linear regression
- Logistic regression
- Support Vector Machine SVM), etc.

Practical applications include

- Handwriting recognition
- Stock market prediction
- Disease prediction
- Fraud detection, etc.

Standard algorithms are

- *k*-means
- Principal Component Analysis (PCA)
- Self-organizing map (SOM)
- Apriori algorithm
- DBSCAN etc.

Practical applications include

- Market basket analysis
- Recommender systems
- Customer segmentation, etc.

Clustering as a Machine Learning task

- The primary driver of clustering knowledge is discovery rather than prediction, because we **may not even know** what we are looking for before starting the clustering analysis
- So, clustering is defined as an unsupervised machine learning task that automatically divides the data into clusters or groups of similar items
- The analysis achieves this without **prior knowledge** of the types of groups required and thus can provide an **insight into** the natural groupings within the data set
- The primary guideline of clustering task is that the data **inside a** cluster should be **very similar** to each other but very different from those outside the cluster
- whenever a **large set of diverse** and **varied data** is presented for analysis, clustering enables to represent the data in a **smaller number of groups**

Clustering as a Machine Learning task

- It helps to reduce the **complexity** and provides **insight** into **patterns** of relationships to generate meaningful and actionable structures within the data.
- The **effectiveness** of clustering is measured by the **homogeneity** within a group as well as the difference between distinct groups
- The main driver for our clustering was the **closeness of the points** to each other to **form a group.**
- The clustering algorithm uses a very similar approach to **measure how closely** the data points **are related** and decides whether they can be labelled as a homogeneous group

Clustering as a Machine Learning task

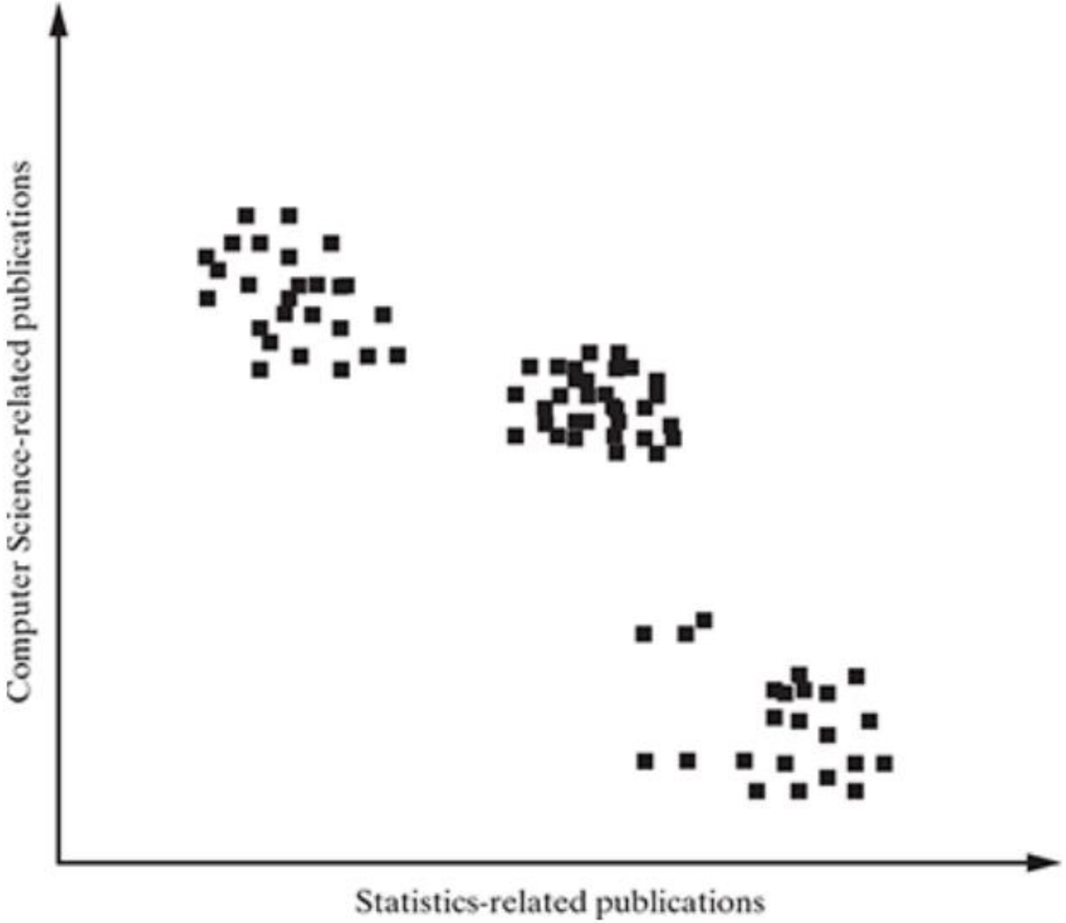


FIG. 9.2 Data set for the conference attendees

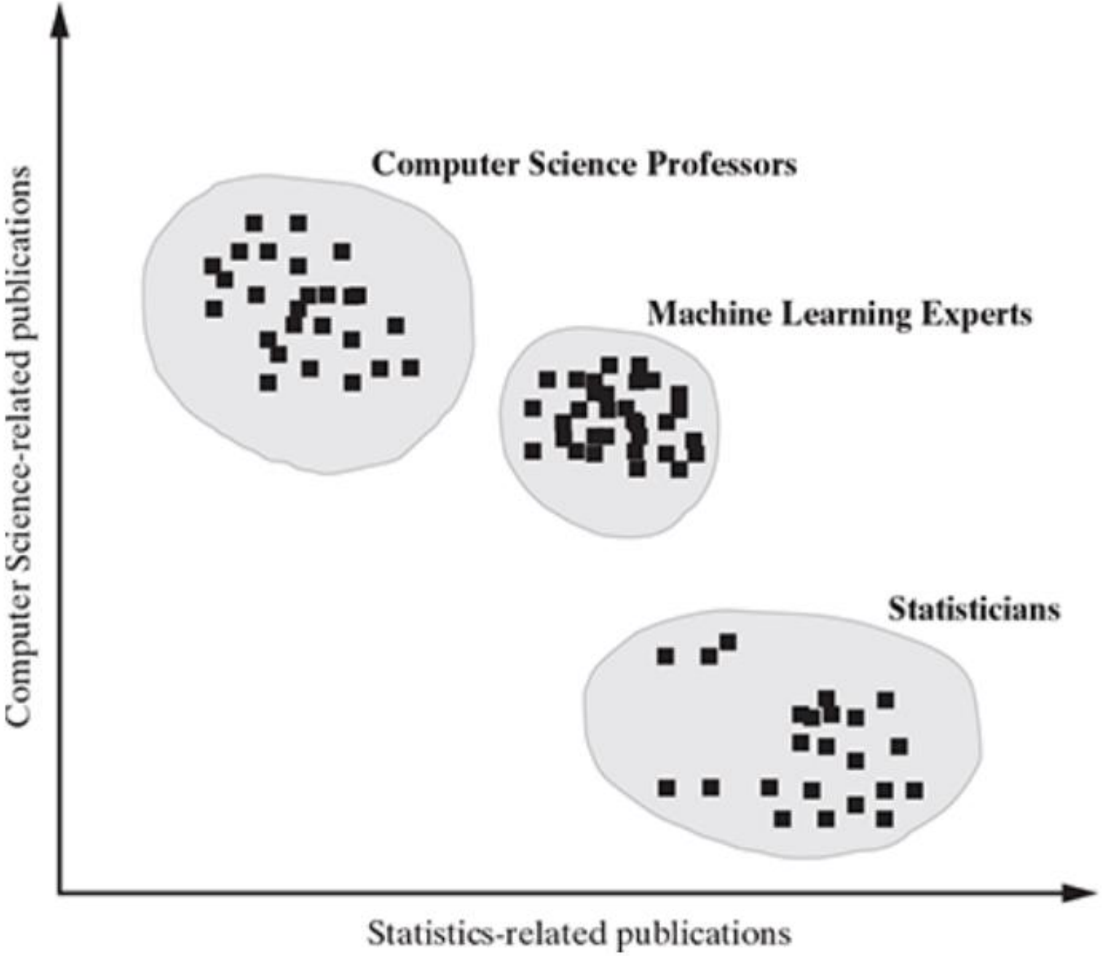


FIG. 9.3 Clusters for the conference attendees

Identifying a Professor who has good number of publications in statistics as well as Computer Science by doing Clustering analysis

Different types of Clustering Techniques

The major clustering techniques are based on :

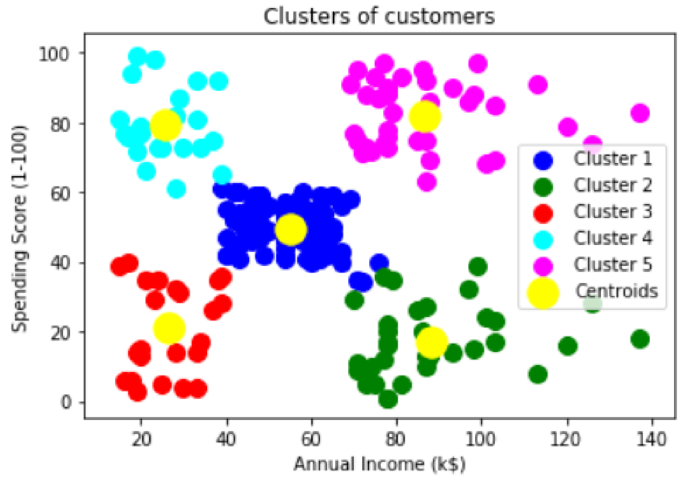
1. Partitioning methods
2. Hierarchical methods
3. Density-based methods

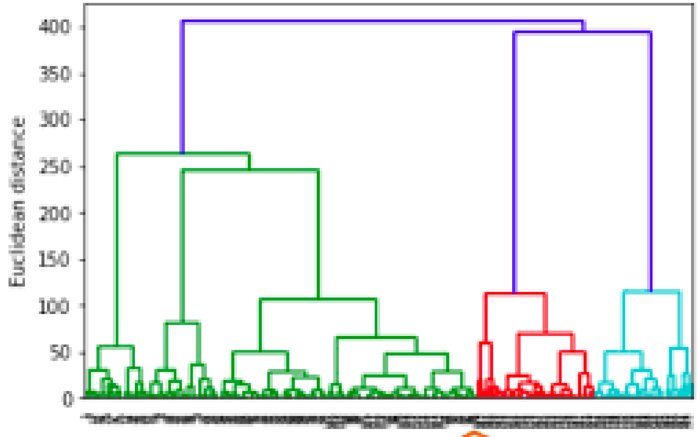
Their approach towards creating the clusters, way to measure the quality of the clusters, and applicability are different

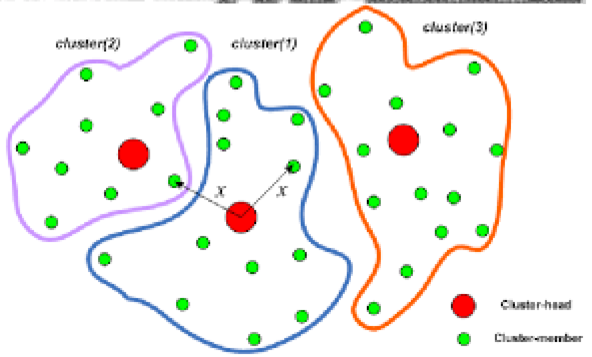
Different types of clustering Techniques

Table 9.1 *Different Clustering Methods*

Method	Characteristics
Partitioning methods	• Uses mean or medoid (etc.) to represent cluster centre • Adopts distance-based approach to refine clusters • Finds mutually exclusive clusters of spherical or nearly spherical shape • Effective for data sets of small to medium size
Hierarchical methods	• Creates hierarchical or tree-like structure through decomposition or merger • Uses distance between the nearest or furthest points in neighbouring clusters as a guideline for refinement • Erroneous merges or splits cannot be corrected at subsequent levels
Density-based methods	• Useful for identifying arbitrarily shaped clusters • Guiding principle of cluster creation is the identification of dense regions of objects in space which are separated by low-density regions • May filter out outliers





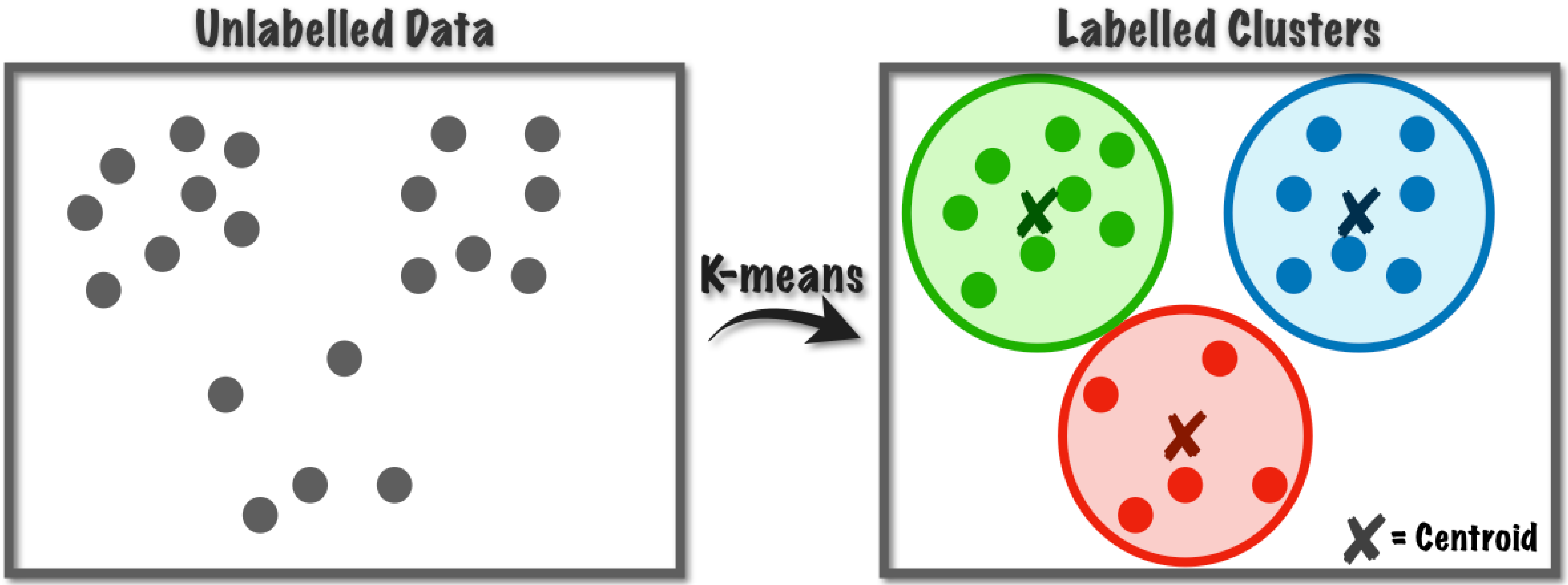


Different types of Clustering Techniques

1. Partitioning methods
    1. K-means
    2. K-mediod

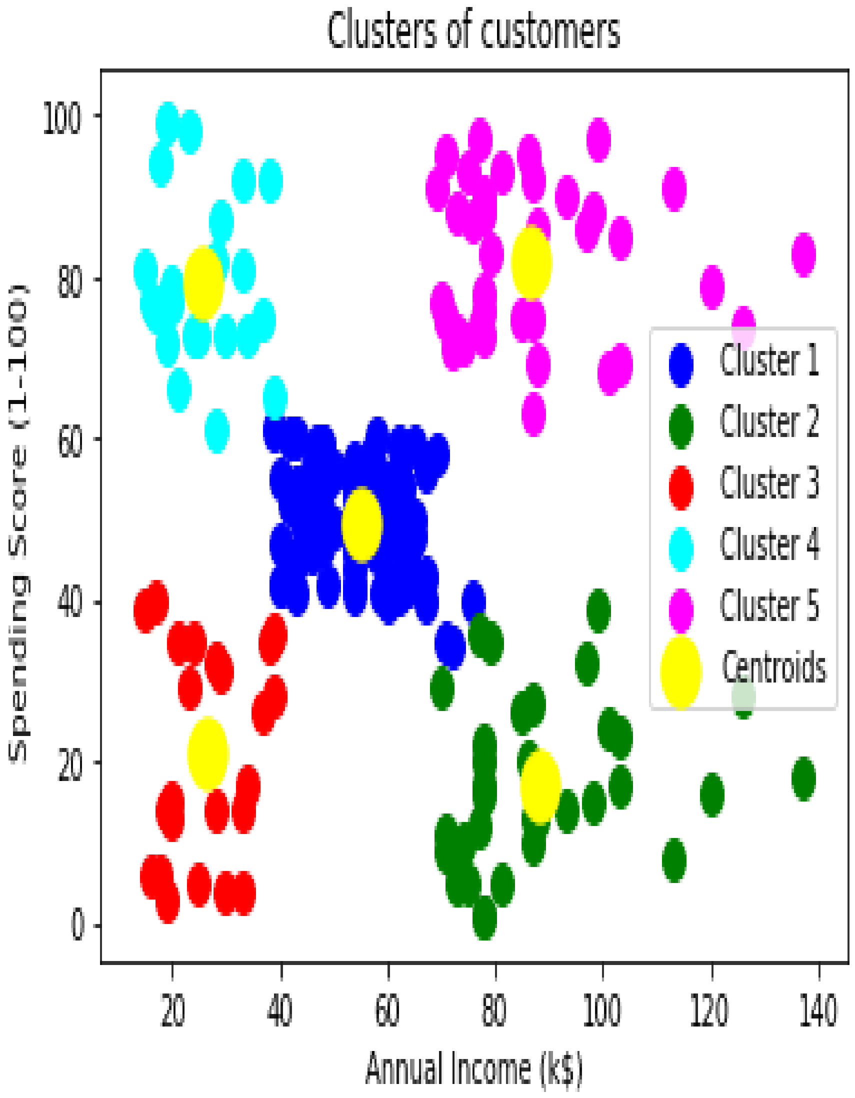
K-means clustering - A centroid-based technique

K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters based on the similarities.



What is K-means Clustering?

- It is an iterative algorithm that divides the **unlabeled dataset** into **'k'** different clusters in such a way that each dataset belongs **only one group** that has **similar properties**
- Here **K** defines the number of pre-defined clusters that need to be created in the process, as if **K=2,** there will be two clusters,
- and for **K=3,** there will be three clusters, and so on.



K- number of clusters

K-means clustering - A centroid-based technique

The algorithm works as follows:

1. Choose '**k**' number of random points (Data point from the data set or some other points). These points are also called "Centroids" or "Means".
2. **Assign all the data points** in the data set to the **closest centroid** by applying any **distance** formula like Euclidian distance, Manhattan distance, etc.
3. Now, choose **new centroids** by calculating the **mean of all the data points** in the clusters and go-to step 2
4. Continue step 3 until no data point changes classification between two iterations

K-means clustering - A centroid-based technique

Algorithm 9.1 shows the simple algorithm of *K*-means

Step 1: Select *K* points in the data space and mark them as initial centroids

loop

Step 2: Assign each point in the data space to the nearest centroid to form K clusters

Step 3: Measure the distance of each point in the cluster from the centroid

Step 4: Calculate the Sum of Squared Error (SSE) to measure the quality of the clusters

Step 5: Identify the new centroid of each cluster on the basis of distance between points

Step 6: Repeat Steps 2 to 5 to refine until centroids do not change

end loop

K-Means Clustering Example-1

- Suppose that the data mining task is to cluster points into three clusters,
- where the points are
- $A1(2, 10)$, $A2(2, 5)$, $A3(8, 4)$, $B1(5, 8)$, $B2(7, 5)$, $B3(6, 4)$, $C1(1, 2)$, $C2(4, 9)$.
- The distance function is Euclidean distance.
- Suppose initially we assign $A1$, $B1$, and $C1$ as the center of each cluster, respectively.

K-Means Clustering- Example-1

Initial Centroids:

A1: (2, 10)

B1: (5, 8)

C1: (1, 2)

Data Points	x_1	y_1	x_2 2	y_2 10	Distance to 5	8	1	2	Cluster	New Cluster
A1	2	10	0.00							
A2	2	5	5.00							
A3	8	4	8.49							
B1	5	8	3.61							
B2	7	5	7.07							
B3	6	4	7.21							
C1	1	2	8.06							
C2	4	9	2.24							

$$d(p_1, p_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

***Calculation of Euclidean distance of first centroid to all data points**

K-Means Clustering -Example-1

Initial Centroids:
A1: (2, 10)
B1: (5, 8)
C1: (1, 2)

Data Points			Distance to			Cluster	New Cluster
			2 10	5 8	1 2		
A1	2	10	0.00	3.61	8.06		
A2	2	5	5.00	4.24	3.16		
A3	8	4	8.49	5.00	7.28		
B1	5	8	3.61	0.00	7.21		
B2	7	5	7.07	3.61	6.71		
B3	6	4	7.21	4.12	5.39		
C1	1	2	8.06	7.21	0.00		
C2	4	9	2.24	1.41	7.62		

$$d(p_1, p_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

** **Calculation of Euclidean distance of second & third centroids to all data points**

K-Means Clustering - Example-1

Initial Centroids:

A1: (2, 10)

B1: (5, 8)

C1: (1, 2)

Data Points			Distance to						Cluster	New Cluster
			2	10	5	8	1	2		
A1	2	10	0.00		3.61		8.06		1	
A2	2	5	5.00		4.24		3.16		3	
A3	8	4	8.49		5.00		7.28		2	
B1	5	8	3.61		0.00		7.21		2	
B2	7	5	7.07		3.61		6.71		2	
B3	6	4	7.21		4.12		5.39		2	
C1	1	2	8.06		7.21		0.00		3	
C2	4	9	2.24		1.41		7.62		2	

$$d(p_1, p_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

***Assignment of data points to nearest clusters by considering least distance value**

K-Means Clustering - Example-1

*Assigning New centroids by calculating mean of clusters

Initial Centroids:
A1: (2, 10)
B1: (5, 8)
C1: (1, 2)

New Centroids:
A1: (2, 10)
B1: (6, 6)
C1: (1.5, 3.5)

Data Points			Distance to			Cluster	New Cluster
			2, 10	5, 8	1, 2		
A1	2	10	0.00	3.61	8.06	1	
A2	2	5	5.00	4.24	3.16	3	
A3	8	4	8.49	5.00	7.28	2	
B1	5	8	3.61	0.00	7.21	2	
B2	7	5	7.07	3.61	6.71	2	
B3	6	4	7.21	4.12	5.39	2	
C1	1	2	8.06	7.21	0.00	3	
C2	4	9	2.24	1.41	7.62	2	

$$d(p_1, p_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

New centroid - B1 (cluster-2)
(Calculation of mean of data points)

X= **A3**(8)+**B1**(5)+**B2**(7)+**B3**(6)+**C2**(4)/**5**= **6**
Y = **A3**(4)+**B1**(8)+**B2**(5)+**B3**(4)+**C2**(9)/**5** = **6**
(**X,Y**) = **(6,6)**

New centroid - C1 (cluster-3) = **A2**(2)+**C1**(1)/**2**= **1.5**
= **A2**(5)+**C1**(2)/**2** = **3.5**
(**X,Y**) = **(1.5,3.5)**

New centroid - A1 (cluster-1) = **A1**(2)/1= **2**
= **A1**(10)/1 = **10**
(**X,Y**) = **(2,10)**

K-Means Clustering - Example-1

Current Centroids:

A1: (2, 10)

B1: (6, 6)

C1: (1.5, 3.5)

Data Points			Distance to			Cluster	New Cluster
			2 10	6 6	1.5 1.5		
A1	2	10	0.00	5.66	6.52	1	
A2	2	5	5.00	4.12	1.58	3	
A3	8	4	8.49	2.83	6.52	2	
B1	5	8	3.61	2.24	5.70	2	
B2	7	5	7.07	1.41	5.70	2	
B3	6	4	7.21	2.00	4.53	2	
C1	1	2	8.06	6.40	1.58	3	
C2	4	9	2.24	3.61	6.04	2	

$$d(p_1, p_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

***Calculation of Euclidean distance of all centroids to all data points**

K-Means Clustering - Example-1

Current Centroids:
A1: (2, 10)
B1: (6, 6)
C1: (1.5, 3.5)

Data Points			Distance to						Cluster	New Cluster
			2	10	6	6	1.5	1.5		
A1	2	10	0.00		5.66		6.52		1	1
A2	2	5	5.00		4.12		1.58		3	3
A3	8	4	8.49		2.83		6.52		2	2
B1	5	8	3.61		2.24		5.70		2	2
B2	7	5	7.07		1.41		5.70		2	2
B3	6	4	7.21		2.00		4.53		2	2
C1	1	2	8.06		6.40		1.58		3	3
C2	4	9	2.24		3.61		6.04		2	1

$$d(p_1, p_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

***Assignment of data points to new clusters by considering least distance value**

K-Means Clustering - Example-1

Current Centroids:
A1: (2, 10)
B1: (6, 6)
C1: (1.5, 3.5)

Data Points			Distance to			Cluster	New Cluster
			2 10	6 6	1.5 1.5		
A1	2	10	0.00	5.66	6.52	1	1
A2	2	5	5.00	4.12	1.58	3	3
A3	8	4	8.49	2.83	6.52	2	2
B1	5	8	3.61	2.24	5.70	2	2
B2	7	5	7.07	1.41	5.70	2	2
B3	6	4	7.21	2.00	4.53	2	2
C1	1	2	8.06	6.40	1.58	3	3
C2	4	9	2.24	3.61	6.04	2	1

$$d(p_1, p_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

***Notice that, C2 data point moved from 2nd cluster to 1st cluster, <u>So that needs to do further clustering</u>**

K-Means Clustering - Example-1

Current Centroids:
A1: (2, 10)
B1: (6, 6)
C1: (1.5, 3.5)

New Centroids:
A1: (3, 9.5)
B1: (6.5, 5.25)
C1: (1.5, 3.5)

Data Points			Distance to						Cluster	New Cluster
			2	10	6	6	1.5	1.5		
A1	2	10	0.00		5.66		6.52		1	1
A2	2	5	5.00		4.12		1.58		3	3
A3	8	4	8.49		2.83		6.52		2	2
B1	5	8	3.61		2.24		5.70		2	2
B2	7	5	7.07		1.41		5.70		2	2
B3	6	4	7.21		2.00		4.53		2	2
C1	1	2	8.06		6.40		1.58		3	3
C2	4	9	2.24		3.61		6.04		2	1

$$d(p_1, p_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

***Assigning New centroids by calculating mean of data points of respective clusters**

K-Means Clustering - Example-1

Current Centroids:
A1: (3, 9.5)
B1: (6.5, 5.25)
C1: (1.5, 3.5)

Data Points			Distance to			Cluster	New Cluster
			3, 9.5	6.5, 5.25	1.5, 3.5		
A1	2	10	1.12	6.54	6.52	1	1
A2	2	5	4.61	4.51	1.58	3	3
A3	8	4	7.43	1.95	6.52	2	2
B1	5	8	2.50	3.13	5.70	2	1
B2	7	5	6.02	0.56	5.70	2	2
B3	6	4	6.26	1.35	4.53	2	2
C1	1	2	7.76	6.39	1.58	3	3
C2	4	9	1.12	4.51	6.04	1	1

$$d(p_1, p_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

***Assigning data points to new clusters**

***Notice that, B1 data point moved from 2nd cluster to 1st cluster, So that needs to do further clustering**

K-Means Clustering - Example-1

Current Centroids:
A1: (3, 9.5)
B1: (6.5, 5.25)
C1: (1.5, 3.5)

New Centroids:
A1: (3.67, 9)
B1: (7, 4.33)
C1: (1.5, 3.5)

Data Points			Distance to			Cluster	New Cluster
			3, 9.5	6.5, 5.25	1.5, 3.5		
A1	2	10	1.12	6.54	6.52	1	1
A2	2	5	4.61	4.51	1.58	3	3
A3	8	4	7.43	1.95	6.52	2	2
B1	5	8	2.50	3.13	5.70	2	1
B2	7	5	6.02	0.56	5.70	2	2
B3	6	4	6.26	1.35	4.53	2	2
C1	1	2	7.76	6.39	1.58	3	3
C2	4	9	1.12	4.51	6.04	1	1

$$d(p_1, p_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

***Assigning New centroids by calculating mean of data points of respective clusters**

K-Means Clustering - Example-1

Current Centroids:
A1: (3.67, 9)
B1: (7, 4.33)
C1: (1.5, 3.5)

Data Points			Distance to			Cluster	New Cluster
			3.67 9	7 4.33	1.5 3.5		
A1	2	10	1.94	7.56	6.52	1	1
A2	2	5	4.33	5.04	1.58	3	3
A3	8	4	6.62	1.05	6.52	2	2
B1	5	8	1.67	4.18	5.70	1	1
B2	7	5	5.21	0.67	5.70	2	2
B3	6	4	5.52	1.05	4.53	2	2
C1	1	2	7.49	6.44	1.58	3	3
C2	4	9	0.33	5.55	6.04	1	1

$$d(p_1, p_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

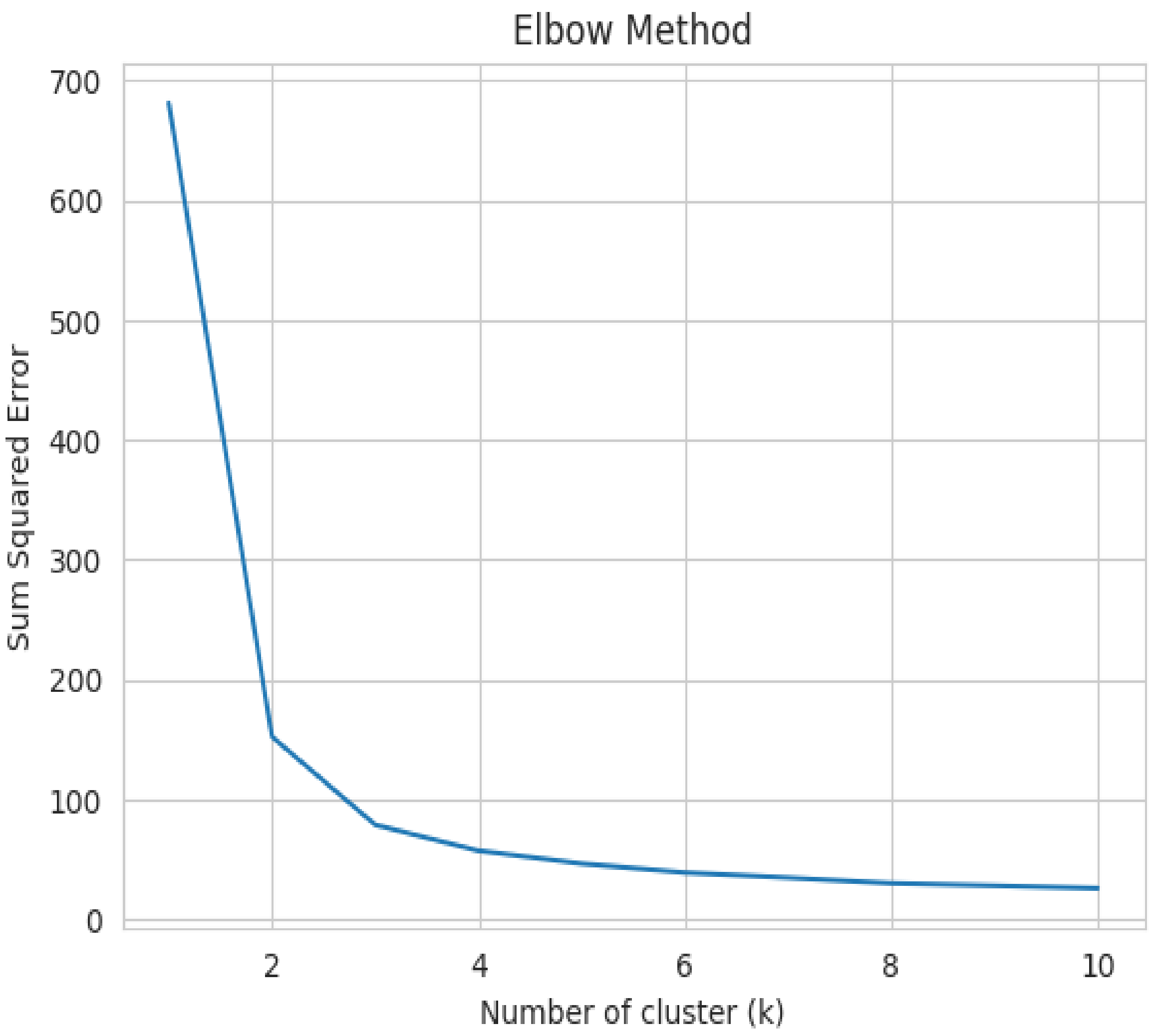
***Assigning data points to new clusters**

***Both Old and New clusters are same, So that we can stop further clustering And consider these are final clusters**

How to identify the optimal 'K' value for K-Means

1. Elbow Method

- Finding the ideal number of groups to divide the data into is a basic stage in any unsupervised algorithm.
- One of the most common techniques for figuring out this ideal value of k is the elbow approach.
- From the beside graph, we can observe that at k=2 and k=3 elbow-like situation. So, we are considering K=3



How to identify the optimal ‘K’ value for K-Means

2. Another traditional approach

$$K = \sqrt{\frac{n}{2}}$$

- Where n is number of data samples

Concerns with K-Means algorithm

- The problem with the K-Means algorithm is that the algorithm needs to handle outlier data
- In general an outlier is **a point different** from the rest of the points.
- All the outlier data points show up in a different cluster and will attract other clusters to merge with it.
- Outlier data increases **the mean** of **a cluster** by **up to 10 units.**
- Hence, **K-Means clustering is** highly affected **by outlier data.**

K-Mediod Clustering

K-Mediod Clustering

- K-medoids is an **unsupervised method** with **un-labelled data** to be clustered.
- It is an improvised version of the K-Means algorithm mainly designed to deal with outlier data sensitivity. Compared to other partitioning algorithms, the algorithm is **simple, fast, and easy to implement.**
- **K-Medoids** (also called ***Partitioning Around Medoid*-PAM**) algorithm was proposed by Kaufman and Rousseeuw.
- A medoid can be defined as a point in the cluster, whose dissimilarities with all the other points in the cluster are **minimum.**
- The dissimilarity of the **medoid(Ci)** and **object(Pi)** is calculated by using

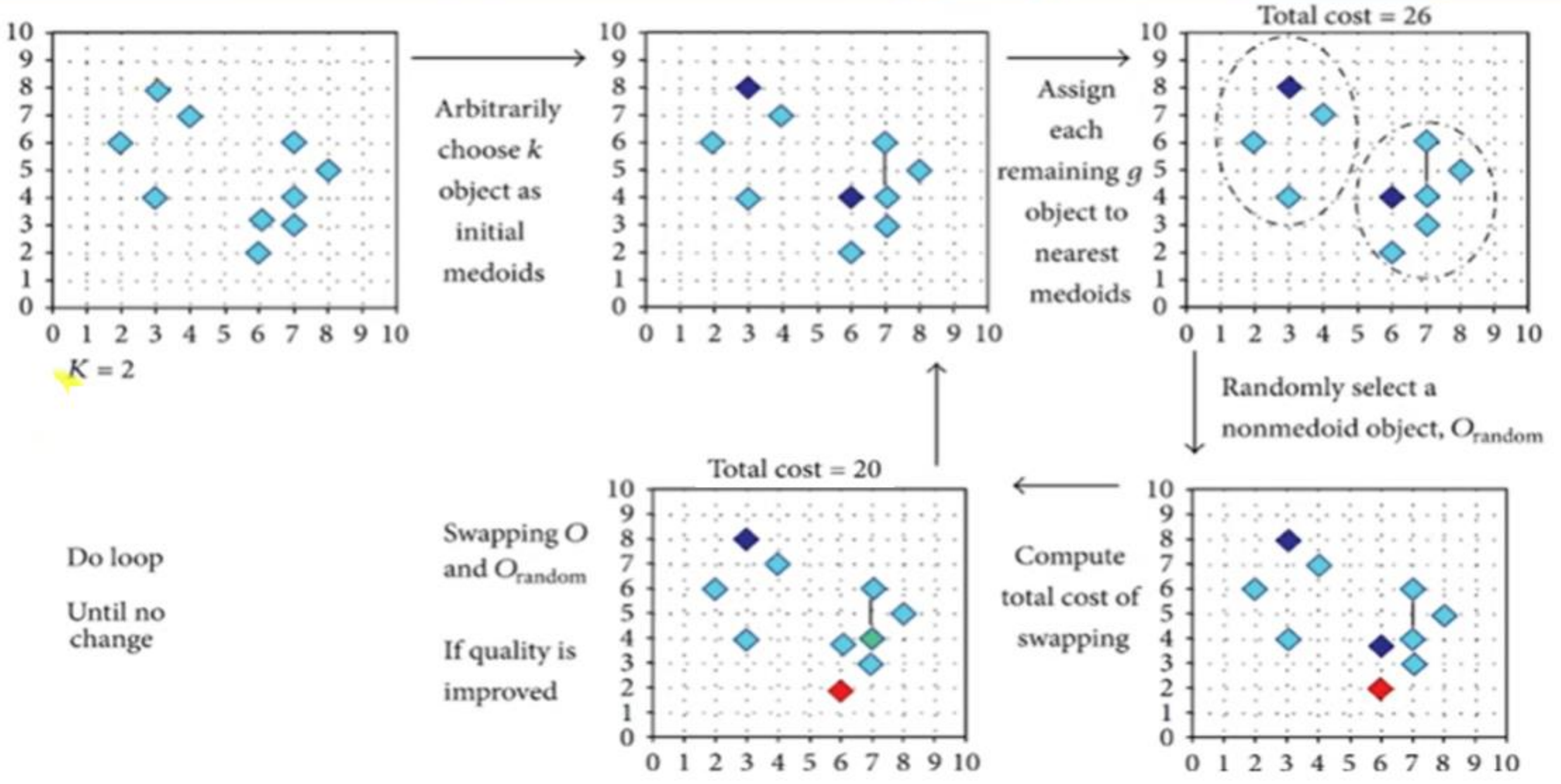
  E = |Pi – Ci|

$$c = \sum_{Ci} \sum_{Pi \in Ci} |Pi - Ci|$$

K-Medoid vs K-Means Clustering

- Partitioning Around Medoids or the K-medoids algorithm is a partitional clustering algorithm which is slightly modified from the K-means algorithm.
- In K-means algorithm, they choose means as the centroids but in the K-medoids, data points are chosen to be the medoids.

K-Medoid Clustering Algorithm



***Note: Total costs 20 & 26 taken as example….)**

K-Medoid Clustering Algorithm

1. Initially select **k random** points as the medoids from the given **n data points** of the data set.
2. **Associate each data point** to **the closest medoid** by using any of the most common **distance metrics.**
3. **Calculate the cost** as the total sum of the distances (also called dissimilarities) of the data points from the assigned medoid. $c = \sum_{Ci} \sum_{Pi \in Ci} |Pi - Ci|$
4. **Swap one medoid point** with a **non-medoid point** and recalculate the cost.
5. If the **calculated cost** with the **new medoid point** is **more** than the previous cost, we **undo the swap**, and the algorithm converges else; **we repeat step 4.**

K-Mediod Clustering - Example-1

Point	X	Y
1	1	4
2	5	1
3	5	2
4	5	4
5	10	4
6	25	4
7	25	6
8	25	7
9	25	8
10	29	7

- Apply K-Medoid clustering algorithm to form two clusters.
- Use Euclidean distance to find the distance between data point and medoid.

K-Mediod Clustering - Example-1

(x_1, y_1)

Step 1

(x_2, y_2) (x_2, y_2)

- Let us choose that (1, 4) and (10, 4) are the medoids
- C1=(1, 4) and C2=(10, 4)
- (x_1, y_1) *and* (x_2, y_2) *are data points*
- $\boldsymbol{Eucladian\ Dist} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$
- $\boldsymbol{EDist}[(1,4),(1,4)] = \sqrt{(1-1)^2 + (4-4)^2} = 0$
- $\boldsymbol{EDist}[(5,1),(1,4)] = \sqrt{(1-5)^2 + (4-1)^2} = 5$

$(x_1, y_1)(x_2, y_2)$

Point	X	Y	C1	C2	Cluster
1	1	4	0		
2	5	1	5		
3	5	2			
4	5	4			
5	10	4			
6	25	4			
7	25	6			
8	25	7			
9	25	8			
10	29	7			

***Choose Medoids randomly & Calculation of Euclidean distance from First Medoid to all data points**

K-Mediod Clustering - Example-1

(x_1, y_1)

Step 1

(x_2, y_2) (x_2, y_2)

- Let us choose that (1, 4) and (10, 4) are the medoids
- C1=(1, 4) and C2=(10, 4)
- (x_1, y_1) *and* (x_2, y_2) *are data points*
- $\boldsymbol{Eucladian\ Dist} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$
- $\boldsymbol{EDist}[(1,4),(1,4)] = \sqrt{(1-1)^2 + (4-4)^2} = 0$
- $\boldsymbol{EDist}[(5,1),(1,4)] = \sqrt{(1-5)^2 + (4-1)^2} = 5$

Point	X	Y	C1	C2	Cluster
1	1	4	0		
2	5	1	5		
3	5	2	4.47		
4	5	4	4		
5	10	4	9		
6	25	4	24		
7	25	6	24.08		
8	25	7	24.19		
9	25	8	24.33		
10	29	7	28.16		

*Calculation of Euclidean distance from First Medoid to all data points (non-Medoid)

K-Mediod Clustering - Example-1

(x_1, y_1)

Step 1

(x_2, y_2) (x_2, y_2)

- Let us choose that (1, 4) and (10, 4) are the medoids
- C1=(1, 4) and C2=(10, 4)
- (x_1, y_1) *and* (x_2, y_2) *are data points*
- $\boldsymbol{Eucladian\ Dist} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

Point	X	Y	C1	C2	Cluster
1	1	4	0	9	
2	5	1	5	5.83	
3	5	2	4.47	5.39	
4	5	4	4	5	
5	10	4	9	0	
6	25	4	24	15	
7	25	6	24.08	15.13	
8	25	7	24.19	15.30	
9	25	8	24.33	15.52	
10	29	7	28.16	19.24	

*Calculation of Euclidean distance from Second Medoid to all data points (non-Medoid)

K-Mediod Clustering - Example-1

Step 1

- Cluster are
- C1: {(1,4), (5,1), (5,2), (5,4)} and
- C2: {(10,4), (25,4), (25,6), (25,7), (25,8), (29,7)}.
- Now calculating the cost which is nothing but the sum of distance of each non-medoid point from the medoid of the cluster it belongs to.

Point	X	Y	C1	C2	Cluster
1	1	4	0	9	C1
2	5	1	5	5.83	C1
3	5	2	4.47	5.39	C1
4	5	4	4	5	C1
5	10	4	9	0	C2
6	25	4	24	15	C2
7	25	6	24.08	15.13	C2
8	25	7	24.19	15.30	C2
9	25	8	24.33	15.52	C2
10	29	7	28.16	19.24	C2

***Assignment of data points to nearest clusters by considering least distance value**

K-Medoid Clustering - Example-1

*Calculation of Total cost from Mediod data points to Non-Medoid data points

- C1: {(1,4), (5,1), (5,2), (5,4)} and C2: {(10,4), (25,4), (25,6), (25,7), (25,8), (29,7)}.
- Calculate the Total Cost
- $Cost(c,x) = \sum_i |c_i - x_i|$
- $Total\ Cost = cost((1,4),(5,1)) + cost((1,4),(5,2)) + cost((1,4),(5,4)) + cost((10,4),(25,4)) + cost((10,4),(25,6)) + cost((10,4),(25,7)) + cost((10,4),(25,8)) + cost((10,4),(29,7))$
- $Total\ Cost = \{|1-5| + |4-1|\} + \{|1-5| + |4-2|\} + \{|1-5| + |4-4|\} + \{|10-25| + |4-4|\} + \{|10-25| + |4-6|\} + \{|10-25| + |4-7|\} + \{|10-25| + |4-8|\} + \{|10-29| + |4-7|\}$
- $Total\ Cost = 4+3+4+2+4+0+15+0+15+2+15+3+15+4+19+3 = 108$

K-Mediod Clustering - Example-1

Step 2

- **Randomly select two non-medoid point and recalculate the cost.**
- C3=(5, 4) and C4=(25, 7)
- **Swap C1 with C3 and C2 with C4**
- **New Medoids**
- **C1=(5, 4) and C2=(25, 7)**

Point	X	Y	C1	C2	Cluster
1	1	4			
2	5	1			
3	5	2			
4	5	4			
5	10	4			
6	25	4			
7	25	6			
8	25	7			
9	25	8			
10	29	7			

***Selection of two non-Medoid points and swaping with existing Mediod points**

K-Mediod Clustering - Example-1

Step 2

- C1=(5, 4) and C2=(25, 7)
- $Eucladian\ Dist = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$
- $EDist[(1,4),(5,4)] = \sqrt{(5-1)^2 + (4-4)^2} = 4$
- $EDist[(1,4),(25,7)] =$

  $\sqrt{(25-1)^2 + (7-4)^2} = 24.19$

Point	X	Y	C1	C2	Cluster
1	1	4	4	24.19	
2	5	1	3	20.8	
3	5	2	2	20.62	
4	5	4	0	20.22	
5	10	4	5	15.30	
6	25	4	20	3	
7	25	6	20.10	1	
8	25	7	20.22	0	
9	25	8	20.40	1	
10	29	7	24.19	4	

***Calculation of Euclidean distance from new Medoids to all data points**

K-Mediod Clustering - Example-1

Step 2

- C1=(5, 4) and C2=(25, 7)
- $Eucladian\ Dist = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$
- $EDist[(1,4),(5,4)] = \sqrt{(5-1)^2 + (4-4)^2} = 4$
- $EDist[(1,4),(25,7)] =$

  $\sqrt{(25-1)^2 + (7-4)^2} = 24.19$

Point	X	Y	C1	C2	Cluster
1	1	4	4	24.19	C1
2	5	1	3	20.8	C1
3	5	2	2	20.62	C1
4	5	4	0	20.22	C1
5	10	4	5	15.30	C1
6	25	4	20	3	C2
7	25	6	20.10	1	C2
8	25	7	20.22	0	C2
9	25	8	20.40	1	C2
10	29	7	24.19	4	C2

***Assignment of data points to nearest clusters by considering least distance value**

K-Mediod Clustering - Example-1

Step 2

- **Cluster are**
- **C1: {(1,4), (5,1), (5,2), (5,4), (10, 4)} and**
- **C2: {(25,4), (25,6), (25,7), (25,8), (29,7)}.**

Point	X	Y	C1	C2	Cluster
1	1	4	4	24.19	C1
2	5	1	3	20.8	C1
3	5	2	2	20.62	C1
4	5	4	0	20.22	C1
5	10	4	5	15.30	C1
6	25	4	20	3	C2
7	25	6	20.10	1	C2
8	25	7	20.22	0	C2
9	25	8	20.40	1	C2
10	29	7	24.19	4	C2

***Formed New clusters by considering least distance value**

K-Mediod Clustering - Example-1

***Calculation of Total cost from New Mediod data points to Non-Medoid data points**

- **C1: {(1,4), (5,1), (5,2), (5,4), (10, 4)} and**
- **C2: {(25,4), (25,6), (25,7), (25,8), (29,7)}.**
- **Calculate the Total Cost**
- $Cost(c,x) = \sum_i |c_i - x_i|$
- $Total\ Cost = cost((5,4),(1,4)) + cost((5,4),(5,1)) + cost((5,4),(5,2)) + cost((5,4),(10,4)) + cost((25,7),(25,4)) + cost((25,7),(25,6)) + cost((25,7),(25,8)) + cost((25,7),(29,7))$
- $Total\ Cost = 23$

K-Mediod Clustering - Example-1

*Calculation of Swapping cost (Comparing Previous Total Cost with Current Total Cost)

Step 2

- $Swapping\ cost = \ Current\ Total\ Cost - \ Previous\ Total\ Cost$
- $Swapping\ cost = \ 23 - 108 = -85 < 0$
- If swapping cost > 0, we stop the algorithm here, and clusters formed in Step 1 is the final answer

*Less Total Cost is considerable

* if swapping cost > 0 , Can stop the further clustering procedure

K-Mediod Clustering - Example-1

Step 3

- **Randomly select two non-medoid point and recalculate the cost.**
- C5=(10, 4) and C6=(29, 7)
- **Swap C1 with C5 and C2 with C6**
- **New Medoids**
- **C1=(10, 4) and C2=(29, 7)**

Point	X	Y	C1	C2	Cluster
1	1	4			
2	5	1			
3	5	2			
4	5	4			
5	10	4			
6	25	4			
7	25	6			
8	25	7			
9	25	8			
10	29	7			

***Selection of two non-Medoid points and swaping with existing Mediod points**

K-Mediod Clustering - Example-1

- New Medoids
- C1=(10, 4) and C2=(29, 7)

Step 3

- Cluster are
- C1: {(1,4), (5,1), (5,2), (5,4), (10, 4)} and
- C2: {(25,4), (25,6), (25,7), (25,8), (29,7)}.

- Total cost $= 9+5+3+5+2+5+0+4+3+4+1+4+0+4+1 = 46$

Point	X	Y	C1	C2	Cluster
1	1	4	9	28.16	C1
2	5	1	5.8	24.73	C1
3	5	2	5.3	24.51	C1
4	5	4	5	24.18	C1
5	10	4	0	19.23	C1
6	25	4	15	5	C2
7	25	6	15.13	4.12	C2
8	25	7	15.29	4	C2
9	25	8	15.52	4.12	C2
10	29	7	19.23	0	C2

*Calculation of Euclidean distance from new Medoids to all data points (Non-Medoid points)

*Calculation of Total cost from New Mediod data points to Non-Medoid data points

K-Mediod Clustering - Example-1

***Calculation of Swapping cost (Comparing Previous Total Cost with Current Total Cost)**

Step 3

- $Swapping\ cost = Current\ Total\ Cost - Previous\ Total\ Cost$
- $Swapping\ cost = 46 - 23 = 23 > 0$
- Swapping cost > 0, we stop the algorithm here.

- Our final clusters are
- C1: {(1,4), (5,1), (5,2), (5,4), (10,4)} and
- C2: {(25,4), (25,6), (25,7), (25,8), (29,7)}

***Current Total Cost higher than previous so that previous Total cost is considerable**
***Swapping cost > 0 , Can stop the further clustering procedure**
***Hence Previous clustering is final (Ignore current clusters because of Higher Total cost value)**

K-Mediod Clustering

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 (arbitrarily shaped)** groups of objects.
- This is because it relies on minimizing the distances between the non-medoid objects and the medoid (the cluster center) – 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.

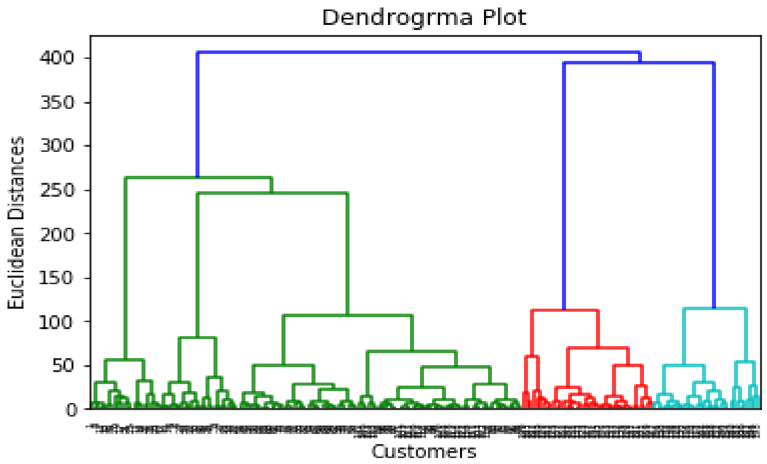
Hierarchical clustering

Hierarchical clustering

- Till now, we have discussed the various methods for partitioning the data into different clusters.
- But there are situations when the data needs to be partitioned into groups at different levels such as in a hierarchy.
- The hierarchical clustering methods are used to group the data into hierarchy or tree-like structure.
- For example, in a machine learning problem of organizing employees of a university in different departments…
- first the employees are grouped under the **different departments** in the university, and then within each department, the employees can be grouped according to **their roles** such as professors, assistant professors, supervisors, lab assistants, etc.
- This creates a hierarchical structure of the employee data and eases visualization and analysis.

Hierarchical clustering

- Hierarchical clustering is another unsupervised machine learning algorithm, which is used to group the unlabeled datasets into a cluster and also known as hierarchical cluster analysis or HCA
- In this algorithm, we develop the **hierarchy of clusters** in the **form of a tree**, and this **Hierarchical clustering**is known as the **dendrogram**
- A **dendrogram** is a commonly used tree structure representation of step-by-step creation of **hierarchical clustering**



dendrogram

Hierarchical clustering

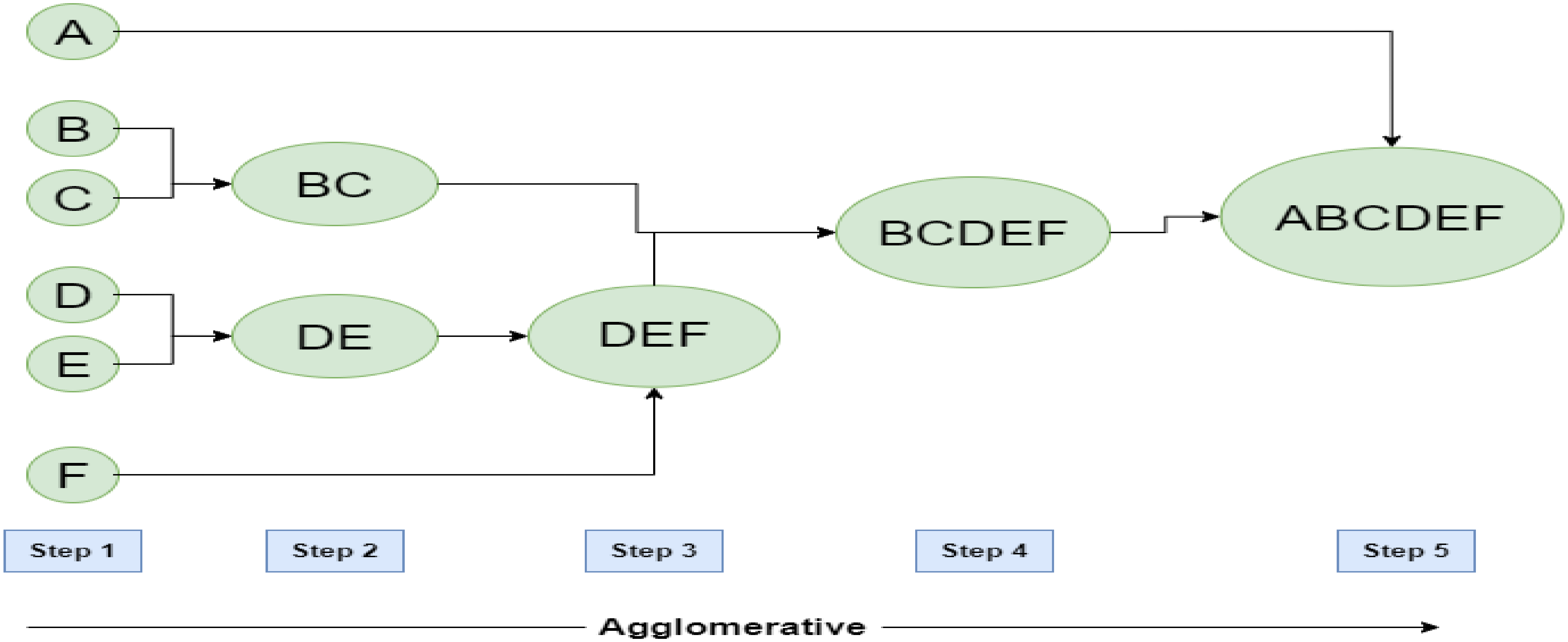
The hierarchical clustering technique has **two approaches**:

- **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 algorithm as it is a **top-down approach**.

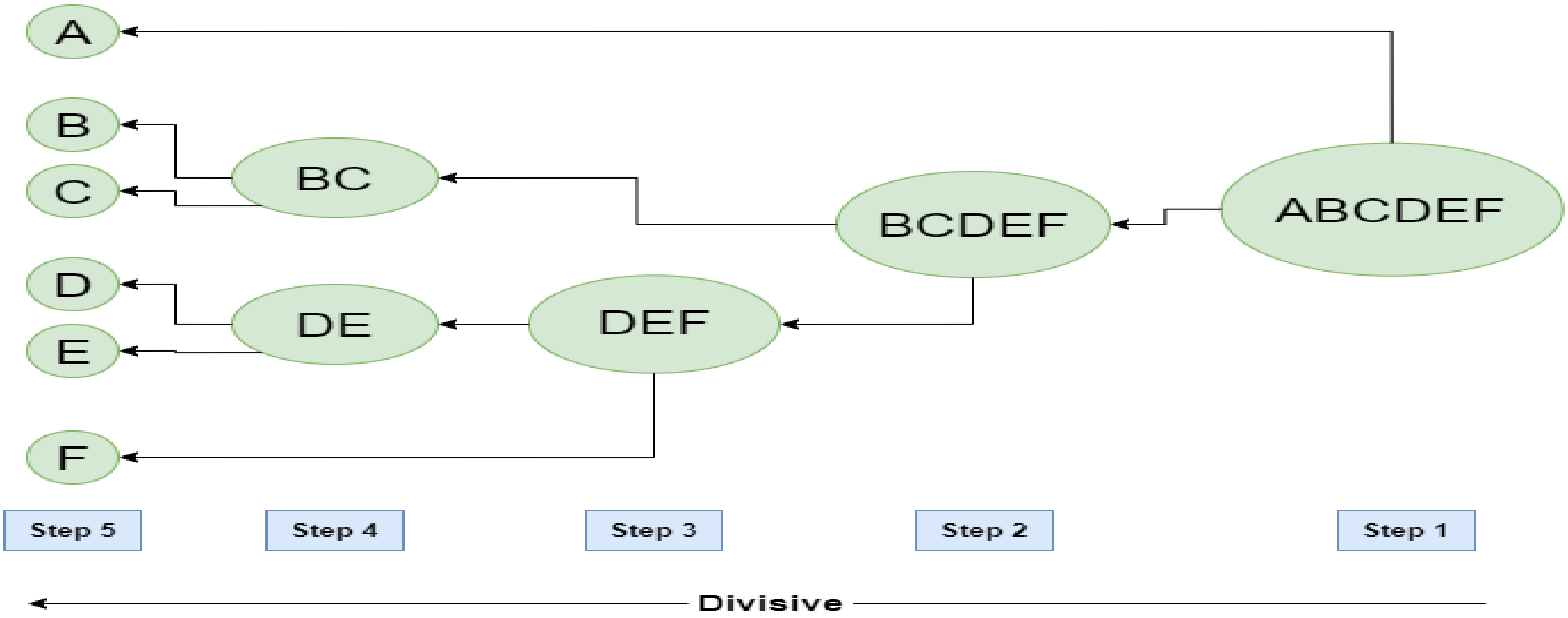
Hierarchical clustering- Agglomerative

- Agglomerative clustering is a bottom-up technique which starts with individual objects as clusters and then iteratively merges them to form larger clusters.



Hierarchical clustering- divisive clustering

- The divisive method starts with one cluster with all given objects and then splits it iteratively to form smaller clusters



Hierarchical clustering- Agglomerative

Calculation of distances :

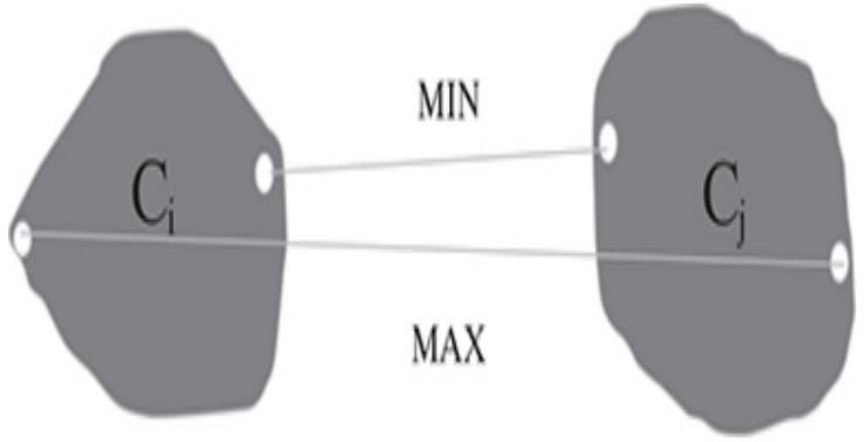
Let C_i and C_j be the two clusters with n_i and n_j respectively. p_i and p_j represents the points in clusters C_i and C_j respectively. We will denote the mean of cluster C_i as m_i.

Minimum distance $D_{\min}(C_i, C_j) = \min_{p_i \in C_i,\ p_j \in C_j} \{|p_i - p_j|\}$

Maximum distance $D_{\max}(C_i, C_j) = \max_{p_i \in C_i,\ p_j \in C_j} \{|p_i - p_j|\}$

Mean distance $D_{\text{mean}}(C_i, C_j) = \{|m_i - m_j|\}$

Average distance $D_{\text{avg}}(C_i, C_j) = \frac{1}{n_i n_j} \sum_{p_i \in C_i,\ p_j \in C_j} |pi - pj|$

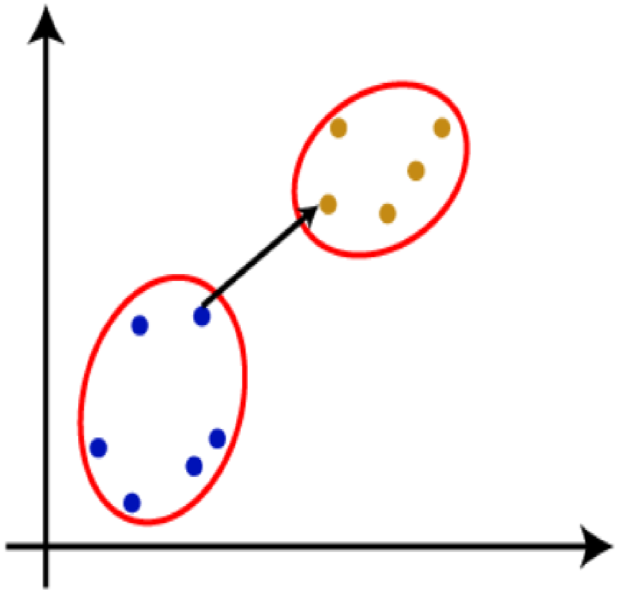


Hierarchical clustering- Agglomerative

Ways to Measure for the distance between two clusters:

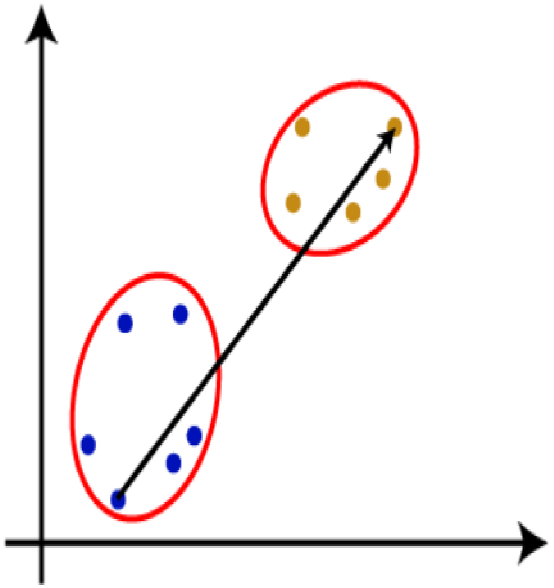
The **closest distance** between the **two clusters** is **crucial** for the **hierarchical clustering**. There are various ways to calculate the distance between two clusters, and these ways decide the **rule for clustering.** These measures are called **Linkage methods.**

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



Minimum distance $D_{min}(C_i, C_j) = \min_{p_i \in C_i,\, p_j \in C_j} \{|p_i - p_j|\}$

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.

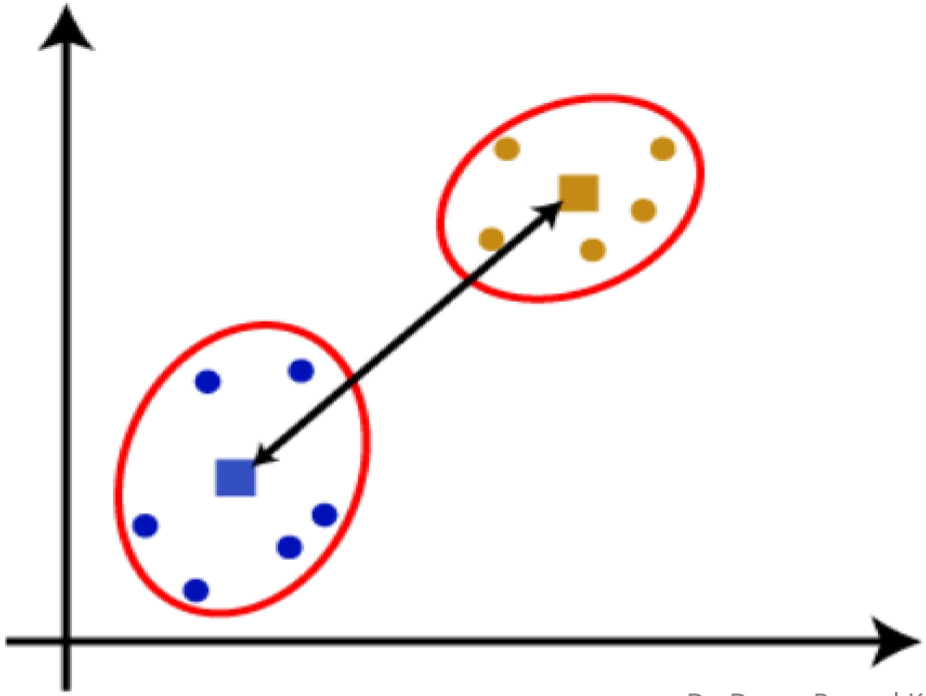


Maximum distance $D_{max}(C_i, C_j) = \max_{p_i \in C_i,\, p_j \in C_j} \{|p_i - p_j|\}$

Hierarchical clustering- Agglomerative

Measure for the distance between two clusters:

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. Consider the below image:



Hierarchical clustering

Algorithm for Agglomerative Hierarchical Clustering :

1. Calculate the similarity of one cluster with all the other clusters **(calculate proximity matrix or distance matrix or symmetric matrix)**
2. Consider every data point as an individual cluster
3. **Merge** the clusters which are **highly similar** or **close to each other**
4. Recalculate the proximity matrix for each cluster
5. Repeat Steps 3 and 4 until only a single cluster remains.

Hierarchical clustering- Agglomerative (Single Linkage)

Example- 1

Hierarchical clustering- Agglomerative (Single Linkage)
Example- 1

- Consider the following set of 6 one dimensional data points:
- 18, 22, 25, 42, 27, 43
- Apply the **agglomerative hierarchical clustering** algorithm to build the hierarchical clustering **dendogram**.
- Merge the clusters using **Min distance** and update the proximity matrix accordingly.
- Clearly show the **proximity matrix** corresponding to each iteration of the algorithm.

Hierarchical clustering- Agglomerative-(Single Linkage)- Example- 1

Example : 18, 22, 25, 42, 27, 43

- Step – 1

	18	22	25	27	42	43
18	0	4	7	9	24	25
22	4	0	3	5	20	21
25	7	3	0	2	17	18
27	9	5	2	0	15	16
42	24	20	17	15	0	1
43	25	21	18	16	1	0

Proximity matrix for taken one dimensional data points

***Calculation of distances of all** data points **and designing of distance matrix or Proximity matrix**

Hierarchical clustering- Agglomerative-(Single Linkage)-
Example- 1

- Step – 1

	18	22	25	27	42	43
18	0	4	7	9	24	25
22	4	0	3	5	20	21
25	7	3	0	2	17	18
27	9	5	2	0	15	16
42	24	20	17	15	0	1
43	25	21	18	16	1	0

	18	22	25	27	42	43
18	0	4	7	9	24	25
22	4	0	3	5	20	21
25	7	3	0	2	17	18
27	9	5	2	0	15	16
42	24	20	17	15	0	1
43	25	21	18	16	1	0

(42, 43)

***Identifying minimum distance on proximity matrix**

Hierarchical clustering- Agglomerative-(Single Linkage)- Example- 1

	18	22	25	27	42	43
18	0	4	7	9	24	25
22	4	0	3	5	20	21
25	7	3	0	2	17	18
27	9	5	2	0	15	16
42	24	20	17	15	0	1
43	25	21	18	16	1	0

(42, 43)

	18	22	25	27	42, 43
18	0	4	7	9	24
22	4	0	3	5	20
25	7	3	0	2	17
27	9	5	2	0	15
42, 43	24	20	17	15	0

***Merging clusters based on minimum distance**

***Removing of 43 data point row & Column and**

***Merging 43 data point with 42 data point** in to **one cluster because of less distance or very similarity on entire matrix**

Hierarchical clustering- Agglomerative-(Single Linkage)- Example- 1

	18	22	25	27	42, 43
18	0	4	7	9	24
22	4	0	3	5	20
25	7	3	0	2	17
27	9	5	2	0	15
42, 43	24	20	17	15	0

	18	22	25	27	42, 43
18	0	4	7	9	24
22	4	0	3	5	20
25	7	3	0	2	17
27	9	5	2	0	15
42, 43	24	20	17	15	0

(42, 43), (25, 27)

***Identifying minimum distance on proximity matrix**

Hierarchical clustering- Agglomerative-(Single Linkage)- Example- 1

	18	22	25	27	42, 43
18	0	4	7	9	24
22	4	0	3	5	20
25	7	3	0	2	17
27	9	5	2	0	15
42, 43	24	20	17	15	0

(42, 43), (25, 27)

	18	22	25, 27	42, 43
18	0	4	7	24
22	4	0	3	20
25, 27	7	3	0	15
42, 43	24	20	15	0

***Merging clusters based on minimum distance**

***Removing of ’ 27 ’ data point row & Column and**

***Merging 27 data point with 25 data point** in to **one cluster because of less distance or very similarity on entire matrix**

Hierarchical clustering- Agglomerative-(Single Linkage)- Example- 1

	18	22	25, 27	42, 43
18	0	4	7	24
22	4	0	3	20
25, 27	7	3	0	15
42, 43	24	20	15	0

	18	22	25, 27	42, 43
18	0	4	7	24
22	4	0	3	20
25, 27	7	3	0	15
42, 43	24	20	15	0

(42, 43), ((25, 27), 22)

***Identifying minimum distance on proximity matrix**

Hierarchical clustering- Agglomerative-(Single Linkage)- Example- 1

	18	22	25, 27	42, 43
18	0	4	7	24
22	4	0	3	20
25, 27	7	3	0	15
42, 43	24	20	15	0

	18	22, 25, 27	42, 43
18	0	4	24
22, 25, 27	4	0	15
42, 43	24	15	0

(42, 43), ((25, 27), 22)

***Merging clusters based on minimum distance**

***Removing of ' 25,27' data point row & Column and**

*__Merging 25,27 data points with 22 data point__ in to **one cluster because of less distance or very similarity on entire matrix**

Hierarchical clustering- Agglomerative-(Single Linkage)- Example- 1

	18	22, 25, 27	42, 43
18	0	4	24
22, 25, 27	4	0	15
42, 43	24	15	0

	18	22, 25, 27	42, 43
18	0	4	24
22, 25, 27	4	0	15
42, 43	24	15	0

(42, 43), (((25, 27), 22), 18)

***Identifying minimum distance on proximity matrix**

Hierarchical clustering- Agglomerative-(Single Linkage)- Example- 1

	18	22, 25, 27	42, 43
18	0	4	24
22, 25, 27	4	0	15
42, 43	24	15	0

(42, 43), (((25, 27), 22), 18)

	18, 22, 25, 27	42, 43
18, 22, 25, 27	0	15
42, 43	15	0

***Merging clusters based on minimum distance**

***Removing of '18' data point row & Column and**

***Merging 18 data point with 22,25,27 data points** in to **one cluster because of less distance or very similarity on entire matrix**

Hierarchical clustering- Agglomerative-(Single Linkage)- Example- 1

	18, 22, 25, 27	42, 43
18, 22, 25, 27	0	15
42, 43	15	0

	18, 22, 25, 27	42, 43
18, 22, 25, 27	0	15
42, 43	15	0

((42, 43), (((25, 27), 22), 18))

***Identifying minimum distance on proximity matrix**

Hierarchical clustering- Agglomerative-(Single Linkage)- Example- 1

	18, 22, 25, 27	42, 43
18, 22, 25, 27	0	15
42, 43	15	0

((42, 43), (((25, 27), 22), 18))

→

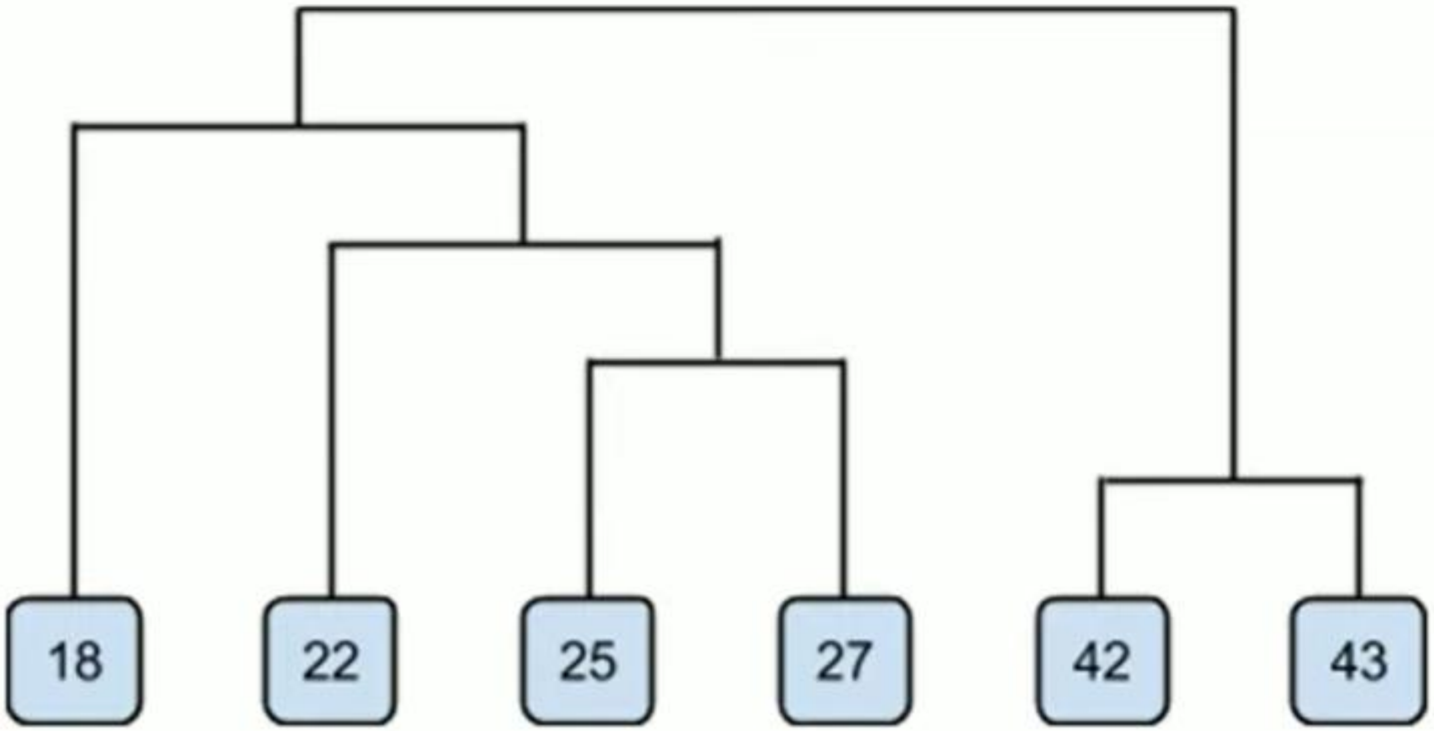
	18, 22, 25, 27, 42, 43
18, 22, 25, 27, 42, 43	0

*Merging clusters based on minimum distance
*Removing of '42,43' data point row & Column and
*Merging 42,43 data points with 18,22,25,27 data points in to one cluster because of less distance or very similarity on entire matrix

Hierarchical clustering- Agglomerative-(Single Linkage)- Example- 1

- **Dendrogram**

((42, 43), (((25, 27), 22), 18))



***Representing Final cluster with Dendrogram**

Hierarchical clustering- Agglomerative (Single Linkage)

Example- 2

Hierarchical clustering- Agglomerative (Single Linkage)
Example- 2

Problem Definition:

For the given dataset find the clusters using a single link technique. Use Euclidean distance and draw the Dendrogram.

Hierarchical clustering- Agglomerative (Single Linkage)
Example- 2

Problem Definition:

For the given dataset find the clusters using a single link technique. Use Euclidean distance and draw the Dendrogram.

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

Hierarchical clustering- Agglomerative (Single Linkage)
Example- 2

Step 1: Compute the distance matrix

- So we have to find the Euclidean distance between each and every points.
- Let $A(x_1, y_1)$ and $B(x_2, y_2)$ are two points.
- Then Euclidean distance between

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

Hierarchical clustering- Agglomerative (Single Linkage)

Example- 2

	Sample No.	X	Y
(x_1, y_1)	P1	0.40	0.53
(x_2, y_2)	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

$$d(p_1, p_2) = \sqrt{(0.22 - 0.40)^2 + (0.38 - 0.53)^2}$$
$$= 0.23$$

$$d(p_1, p_3) = \sqrt{(0.35 - 0.40)^2 + (0.32 - 0.53)^2}$$
$$= 0.22$$

$$d(p_2, p_3) = \sqrt{(0.35 - 0.22)^2 + (0.32 - 0.38)^2}$$
$$= 0.14$$

Hierarchical clustering- Agglomerative (Single Linkage)
Example- 2

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

$$\begin{pmatrix} & P1 & P2 & P3 & P4 & P5 & P6 \\ P1 & 0 & & & & & \\ P2 & 0.23 & 0 & & & & \\ P3 & 0.22 & 0.14 & 0 & & & \\ P4 & 0.37 & 0.19 & 0.13 & 0 & & \\ P5 & 0.34 & 0.14 & 0.28 & 0.23 & 0 & \\ P6 & 0.24 & 0.24 & 0.10 & 0.22 & 0.39 & 0 \end{pmatrix}$$

Hierarchical clustering- Agglomerative (Single Linkage)
Example- 2

Step 2: Merging the two closest members.

- Here the **minimum value is 0.10** and hence we combine P3 and P6 (as 0.10 came in the P6 row and P3 column).
- Now, form clusters of elements corresponding to the minimum value and update the distance matrix.

Hierarchical clustering- Agglomerative (Single Linkage)

Example- 2

Now we will update the Distance Matrix:

$$
\begin{pmatrix}
 & P1 & P2 & P3 & P4 & P5 & P6 \\
P1 & 0 & & & & & \\
P2 & 0.23 & 0 & & & & \\
P3 & 0.22 & 0.14 & 0 & & & \\
P4 & 0.37 & 0.19 & 0.13 & 0 & & \\
P5 & 0.34 & 0.14 & 0.28 & 0.23 & 0 & \\
P6 & 0.24 & 0.24 & 0.10 & 0.22 & 0.39 & 0
\end{pmatrix}
\begin{pmatrix}
 & P1 & P2 & P3, P6 & P4 & P5 \\
P1 & 0 & & & & \\
P2 & 0.23 & 0 & & & \\
P3, P6 & 0.22 & 0.14 & 0 & & \\
P4 & 0.37 & 0.19 & 0.13 & 0 & \\
P5 & 0.34 & 0.14 & 0.28 & 0.23 & 0
\end{pmatrix}
$$

(P3, P6)

Hierarchical clustering- Agglomerative (Single Linkage)
Example- 2

Now we will repeat the same process.

Merge two closest members of the two clusters. The **minimum value is 0.13** and hence we combine P3, P6 and P4.

Now we will update the Distance Matrix:

$$
\begin{array}{c|ccccc}
 & P1 & P2 & P3,P6 & P4 & P5 \\
\hline
P1 & 0 & & & & \\
P2 & 0.23 & 0 & & & \\
P3,P6 & 0.22 & 0.14 & 0 & & \\
P4 & 0.37 & 0.19 & 0.13 & 0 & \\
P5 & 0.34 & 0.14 & 0.28 & 0.23 & 0
\end{array}
\qquad
\begin{array}{c|cccc}
 & P1 & P2 & P3,P6,P4 & P5 \\
\hline
P1 & 0 & & & \\
P2 & 0.23 & 0 & & \\
P3,P6,P4 & 0.22 & 0.14 & 0 & \\
P5 & 0.34 & 0.14 & 0.28 & 0
\end{array}
$$

{(P3, P6), P4}

Hierarchical clustering- Agglomerative (Single Linkage)
Example- 2

Now we will update the Distance Matrix:

$$\begin{array}{c} \\ P1 \\ P2 \\ P3,P6,P4 \\ P5 \end{array}\begin{array}{ccccc} P1 & P2 & P3,P6,P4 & P5 \\ \end{array}$$

	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

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

{(P3, P6), P4} and (P2, P5)

Hierarchical clustering- Agglomerative (Single Linkage)

Example- 2

Now we will update the Distance Matrix:

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

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

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

*Merging **P2, P5** with **P3,P6,P4** and Removing **P2, P5**

Hierarchical clustering- Agglomerative (Single Linkage)

Example- 2

Now we will update the Distance Matrix:

$$\begin{array}{c|cc} & P1 & P3, P6, P4 \\ \hline P1 & 0 & \\ P2, P5, P3, P6, P4 & 0.22 & 0 \end{array}$$

[{(P3, P6), P4}, (P2, P5)] , P1 ⟸ **Final cluster**

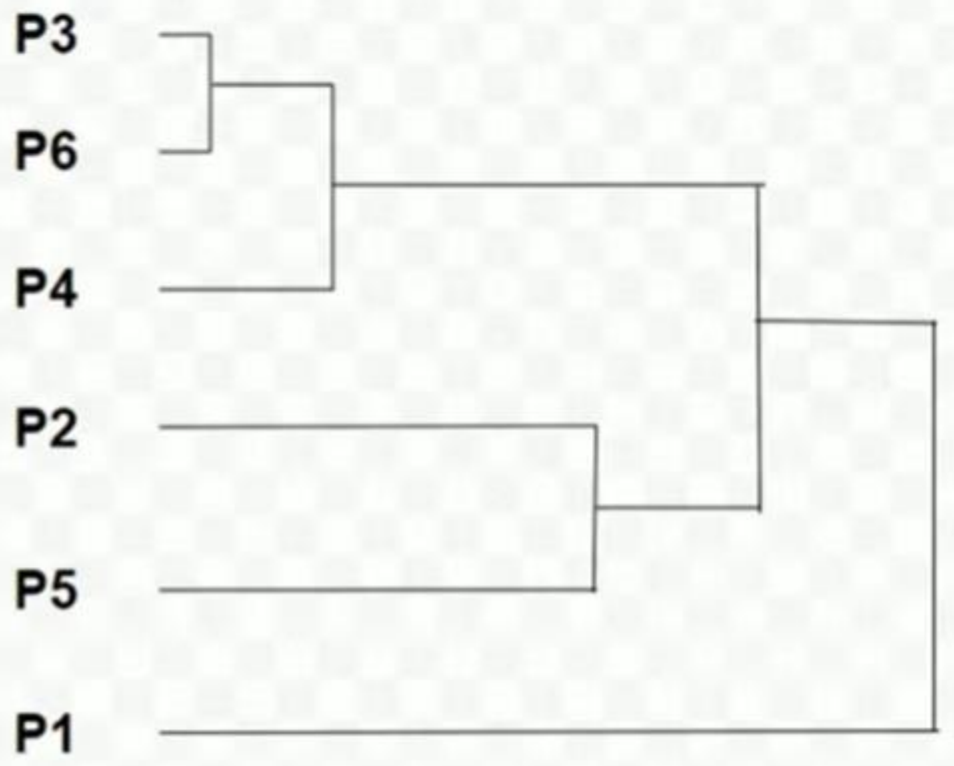
Hierarchical clustering- Agglomerative (Single Linkage)

Example- 2

Dendogram of Final clustering

So now we have reached to the solution, the dendrogram for those question will be as follows:

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



Dendogram of the cluster formed

Hierarchical clustering- Agglomerative (Complete Linkage)

Example- 1

Hierarchical clustering- Agglomerative (Complete Linkage)

Example- 1

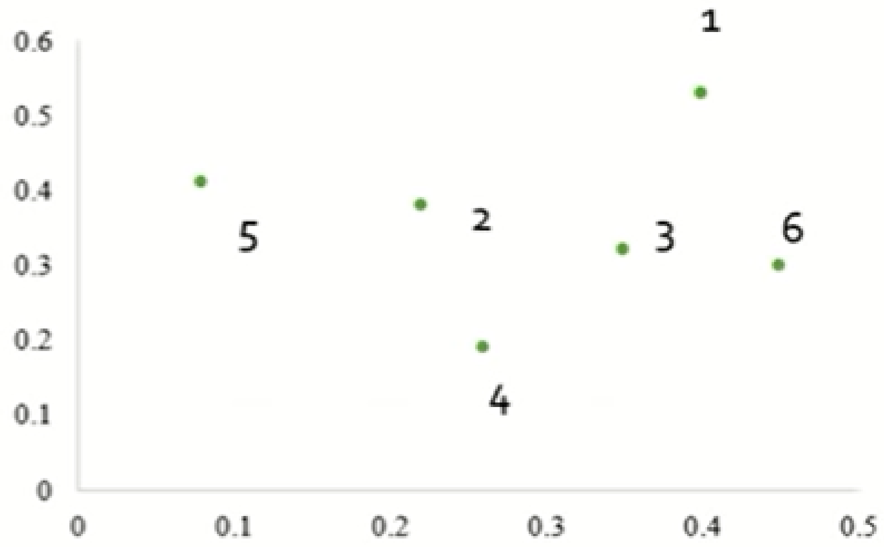
- Find the clusters using complete link technique. Use Euclidean distance, and draw the dendrogram.

	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

Hierarchical clustering- Agglomerative (Complete Linkage)

Example- 1

	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



***Plotting with given data**

Hierarchical clustering- Agglomerative (Complete Linkage)

Example- 1

- Calculate Euclidean distance, create the distance matrix.

$$\textbf{Eucladian Dist} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Distance (P1,P2) $= \sqrt{(0.40 - 0.22)^2 + (0.53 - 0.38)^2}$

(0.40,0.53), (0.22,0.38) $= \sqrt{(0.18)^2 + (0.15)^2}$

$= \sqrt{0.0324 + 0.0225}$

$= \sqrt{0.0549}$

$= 0.23$

	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

***Calculating distance of data points to each other data points**

Hierarchical clustering- Agglomerative (Complete Linkage)

Example- 1

- The distance matrix is

	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

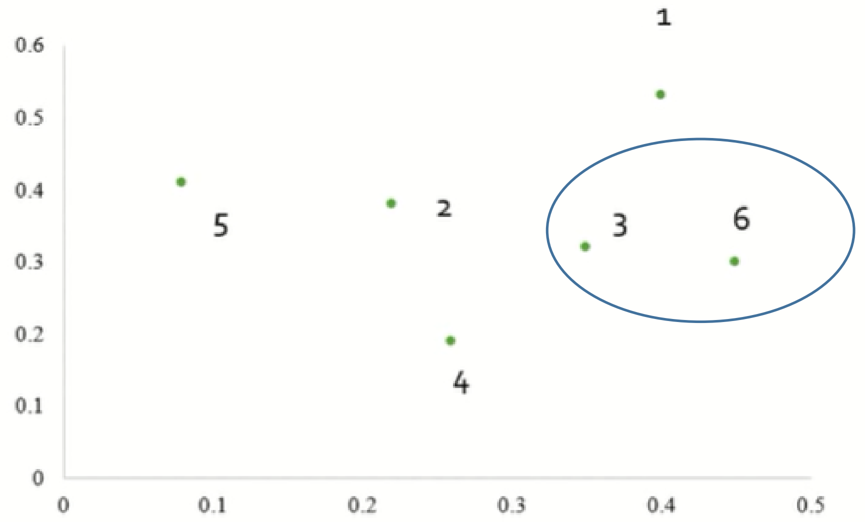
	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.23	0				
P3	0.22	0.15	0			
P4	0.37	0.20	0.15	0		
P5	0.34	0.14	0.28	0.29	0	
P6	0.23	0.25	0.11	0.22	0.39	0

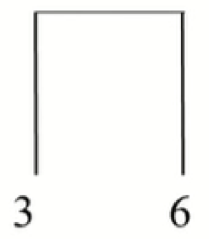
***Distance matrix**

Hierarchical clustering- Agglomerative (Complete Linkage)
Example- 1

- The distance matrix is

	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.23	0				
P3	0.22	0.15	0			
P4	0.37	0.20	0.15	0		
P5	0.34	0.14	0.28	0.29	0	
P6	0.23	0.25	0.11	0.22	0.39	0





*Identifying **least distance** from distance matrix **to form cluster**

* **Here [P3, P6] form cluster**

Hierarchical clustering- Agglomerative (Complete Linkage)
Example- 1

- The distance matrix for cluster P3,P6

	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.23	0				
P3	0.22	0.15	0			
P4	0.37	0.20	0.15	0		
P5	0.34	0.14	0.28	0.29	0	
P6	0.23	0.25	0.11	0.22	0.39	0

- To update the distance matrix MAX[dist(P3,P6),P1)]
- MAX(dist(P3,P1), (P6,P1))

  = MAX[(0.22,0.23)]

  = 0.23
- To update the distance matrix MAX[dist(P3,P6),P2)]
- MAX(dist(P3,P2), (P6,P2))

  = MAX[(0.15,0.25)]

  = 0.25

***Calculation of MAX DISTANCE from New [P3,P6] cluster to all data points**

Hierarchical clustering- Agglomerative (Complete Linkage)

Example- 1

- The distance matrix for cluster P3,P6

	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.23	0				
P3	0.22	0.15	0			
P4	0.37	0.20	0.15	0		
P5	0.34	0.14	0.28	0.29	0	
P6	0.23	0.25	0.11	0.22	0.39	0

- To update the distance matrix MAX[dist(P3,P6),P4)]
- MAX(dist(P3,P4), (P6,P4))

  = MAX[(0.15,0.22)]

  = 0.22
- To update the distance matrix MAX[dist(P3,P6),P5)]
- MAX(dist(P3,P5), (P6,P5))

  = MAX[(0.28,0.39)]

  = 0.39

***Calculation of MAX distance from New [P3,P6] cluster to all data points**

Hierarchical clustering- Agglomerative (Complete Linkage)
Example- 1

- The updated distance matrix for cluster P3,P6

	P1	P2	P3,P6	P4	P5
P1	0				
P2	0.23	0			
P3,P6	0.23	0.25	0		
P4	0.37	0.20	0.22	0	
P5	0.34	0.14	0.39	0.29	0

*** Merging [P3,P6] clusters & Updating [P3,P6] cluster values with Newly calculated values in distance matrix**

Hierarchical clustering- Agglomerative (Complete Linkage)

Example- 1

- The distance matrix

	P1	P2	P3,P6	P4	P5
P1	0				
P2	0.23	0			
P3,P6	0.23	0.25	0		
P4	0.37	0.20	0.22	0	
P5	0.34	0.14	0.39	0.29	0

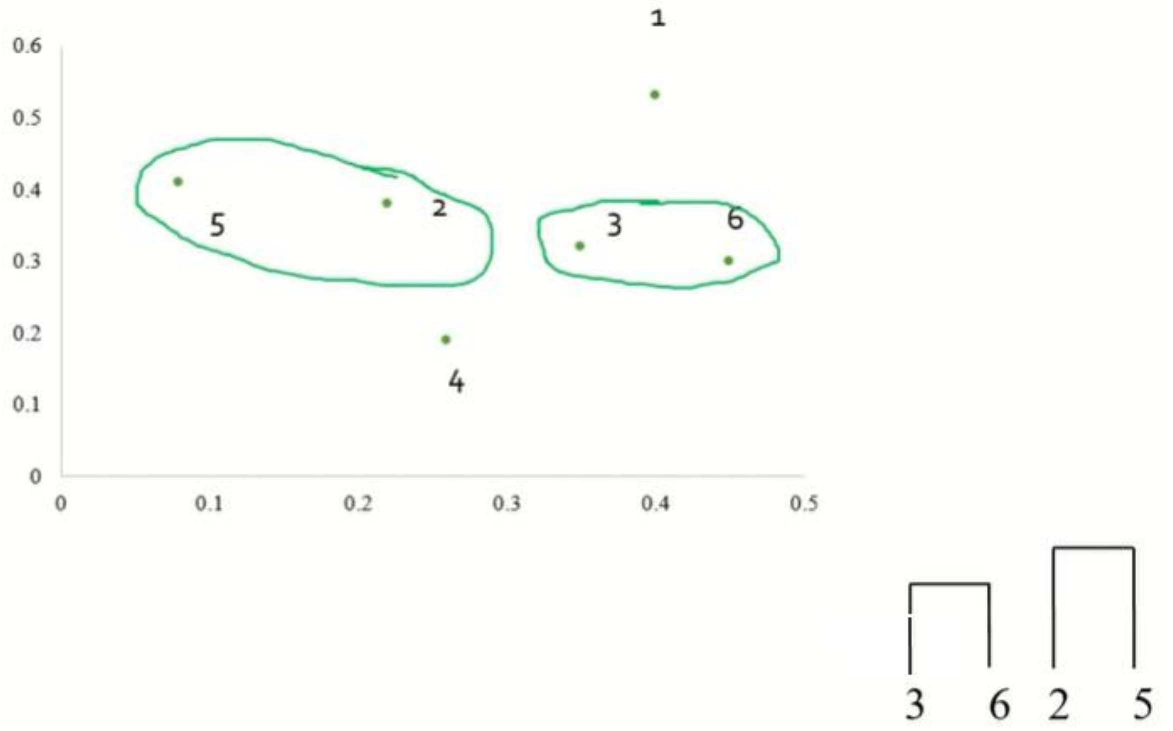
***Identifying least distance** from distance matrix **to form cluster**

*** Here [P2, P5] form cluster**

Hierarchical clustering- Agglomerative (Complete Linkage)
Example- 1

- The distance matrix

	P1	P2	P3,P6	P4	P5
P1	0				
P2	0.23	0			
P3,P6	0.23	0.25	0		
P4	0.37	0.20	0.22	0	
P5	0.34	0.14	0.39	0.29	0



Hierarchical clustering- Agglomerative (Complete Linkage)
Example- 1

- The distance matrix , cluster $\boxed{(P2,P5)}$

	P1	P2	P3,P6	P4	P5
P1	0				
P2	0.23	0			
P3,P6	0.23	0.25	0		
P4	0.37	0.20	0.22	0	
P5	0.34	0.14	0.39	0.29	0

- To update the distance matrix MAX[dist(P2,P5),P1)]
- MAX[dist(P2,P1), (P5,P1)]

  = MAX[(0.23,0.34)]

  = 0.34
- To update the distance matrix MAX[dist(P2,P5),(P3,P6)]
- MAX[dist(P2,(P3,P6)), (P5,(P3,P6))]

  = MAX[(0.25,0.39)]

  = 0.39

***Calculation of MAX distance from New [P2,P5] cluster to all data points**

Hierarchical clustering- Agglomerative (Complete Linkage)
Example- 1

- The distance matrix , cluster (P2,P5)

	P1	P2	P3,P6	P4	P5
P1	0				
P2	0.23	0			
P3,P6	0.23	0.25	0		
P4	0.37	0.20	0.22	0	
P5	0.34	0.14	0.39	0.29	0

- To update the distance matrix MAX[dist(P2,P5),P4)]
- MAX[dist(P2,P4), (P5,P4)]

  = MAX[(0.20,0.29)]

  = 0.29

***Calculation of MAX distance from New [P2,P5] cluster to all data points**

Hierarchical clustering- Agglomerative (Complete Linkage)
Example- 1

- The updated distance matrix for cluster (P2,P5)

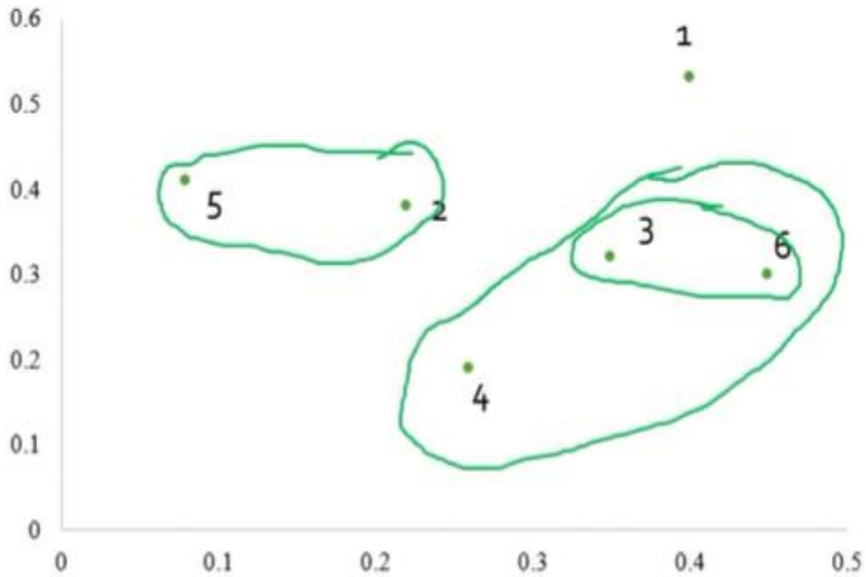
	P1	P2,P5	P3,P6	P4
P1	0			
P2,P5	0.34	0		
P3,P6	0.23	0.39	0	
P4	0.37	0.29	0.22	0

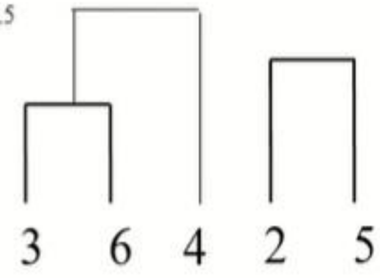
*** Merging clusters [P2,P5] & Updating [P2,P5] cluster values in distance matrix with Newly calculated values**

Hierarchical clustering- Agglomerative (Complete Linkage)
Example- 1

- The distance matrix is

	P1	P2,P5	P3,P6	P4
P1	0			
P2,P5	0.34	0		
P3,P6	0.23	0.39	0	
P4	0.37	0.29	0.22	0





Identifying least distance** from distance matrix **to form cluster

***** Here [[P3,P6],P4]] form cluster**

Hierarchical clustering- Agglomerative (Complete Linkage)

Example- 1

[[P3,P6],P4]]

- The distance matrix is

	P1	P2,P5	P3,P6	P4
P1	0			
P2,P5	0.34	0		
P3,P6	0.23	0.39	0	
P4	0.37	0.29	0.22	0

- To update the distance matrix[((P3,P6),P4) , P1]
- MAX[dist(P3,P6),P1), (P4,P1)]

  = MAX[(0.23,0.37)]

  = 0.37

- To update the distance matrix MAX[dist((P3,P6),P4),(P2,P5)]
- MAX[dist((P3,P6),(P2,P5)), (P4,(P2,P5))]

  = MAX[(0.39,0.29)]

  = 0.39

***Calculation of MAX distance from New [[P3,P6],P4]] cluster to all data points**

Hierarchical clustering- Agglomerative (Complete Linkage)
Example- 1

- The updated distance matrix for cluster P3,P6,P4

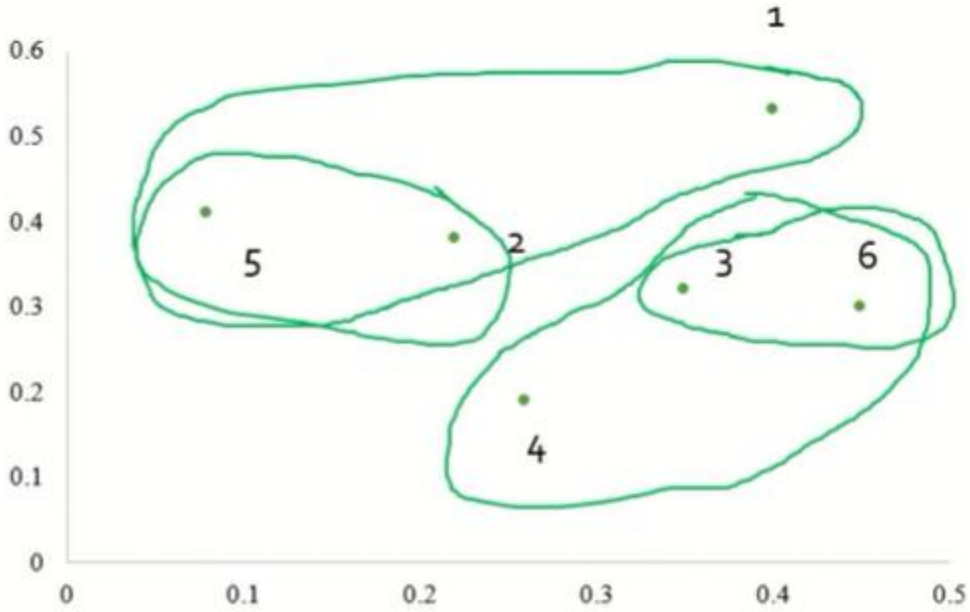
	P1	P2,P5	P3,P6,P4
P1	0		
P2,P5	0.34	0	
P3,P6,P4	0.37	0.39	0

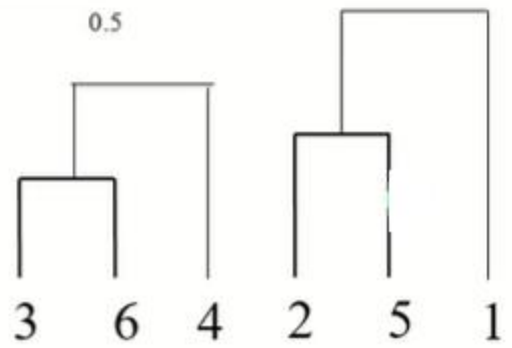
***Merging [[P3,P6],P4] clusters & Updating [[P3,P6],P4] cluster values in distance matrix with Newly calculated values**

Hierarchical clustering- Agglomerative (Complete Linkage)
Example- 1

- The distance matrix is

	P1	P2,P5	P3,P6,P4
P1	0		
P2,P5	0.34	0	
P3,P6,P4	0.37	0.39	0





***Identifying least distance** from distance matrix **to form cluster**

*** Here [[P2,P5],P1] form cluster**

Hierarchical clustering- Agglomerative (Complete Linkage)
Example- 1

- The distance matrix is

[[P2,P5],P1]]

	P1	P2,P5	P3,P6,P4
P1	0		
P2,P5	0.34	0	
P3,P6,P4	0.37	0.39	0

- **Cluster** [(P2,P5),P1]
- MAX[dist(P2,P5),(P3,P6,P4), (P1,(P3,P6,P4))]

  = MAX[(0.39,0.37)]

  = 0.39

***Calculation of MAX distance from New [[P2,P5],P1]] cluster to all data points**

Hierarchical clustering- Agglomerative (Complete Linkage)
Example- 1

- The updated distance matrix for cluster P2,P5,P1

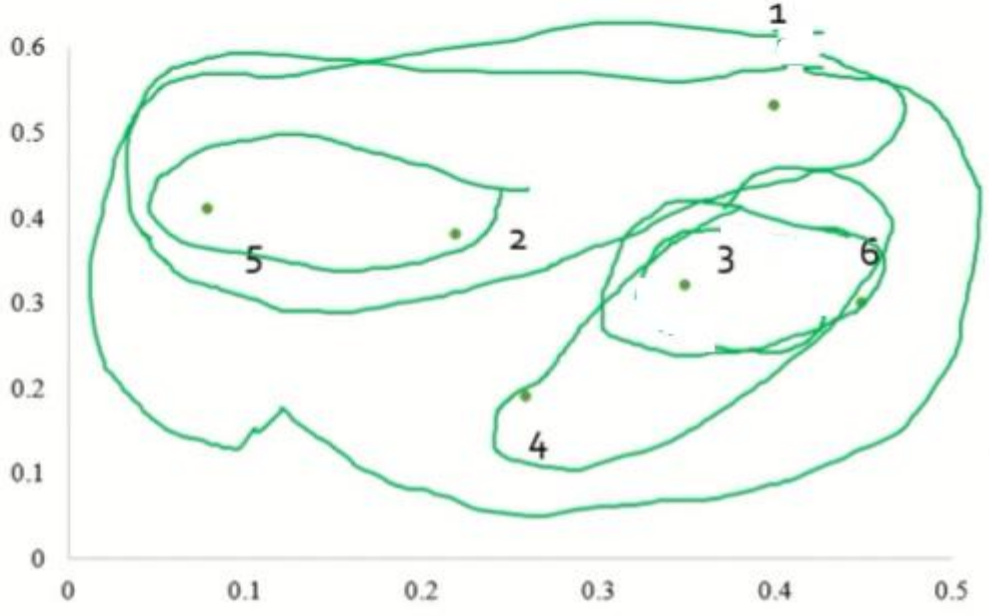
	P2,P5,P1	P3,P6,P4
P2,P5,P1	0	
P3,P6,P4	0.39	0

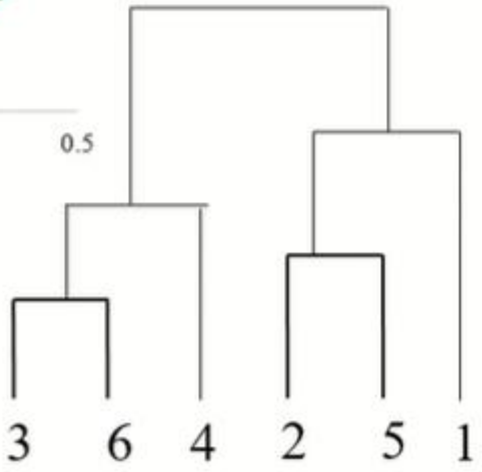
***Merging [[P2,P5],P1]] clusters & Updating [[P2,P5],P1]] cluster values in distance matrix with Newly calculated values**

Hierarchical clustering- Agglomerative (Complete Linkage)
Example- 1

- The distance matrix

	P2,P5,P1	P3,P6,P4
P2,P5,P1	0	
P3,P6,P4	0.39	0





Identifying least distance** from distance matrix **to form cluster

*** Here [P3,P6,P4,P2,P5,P1] form cluster**

Hierarchical clustering- Agglomerative (Complete Linkage)

Example- 1

- The distance matrix

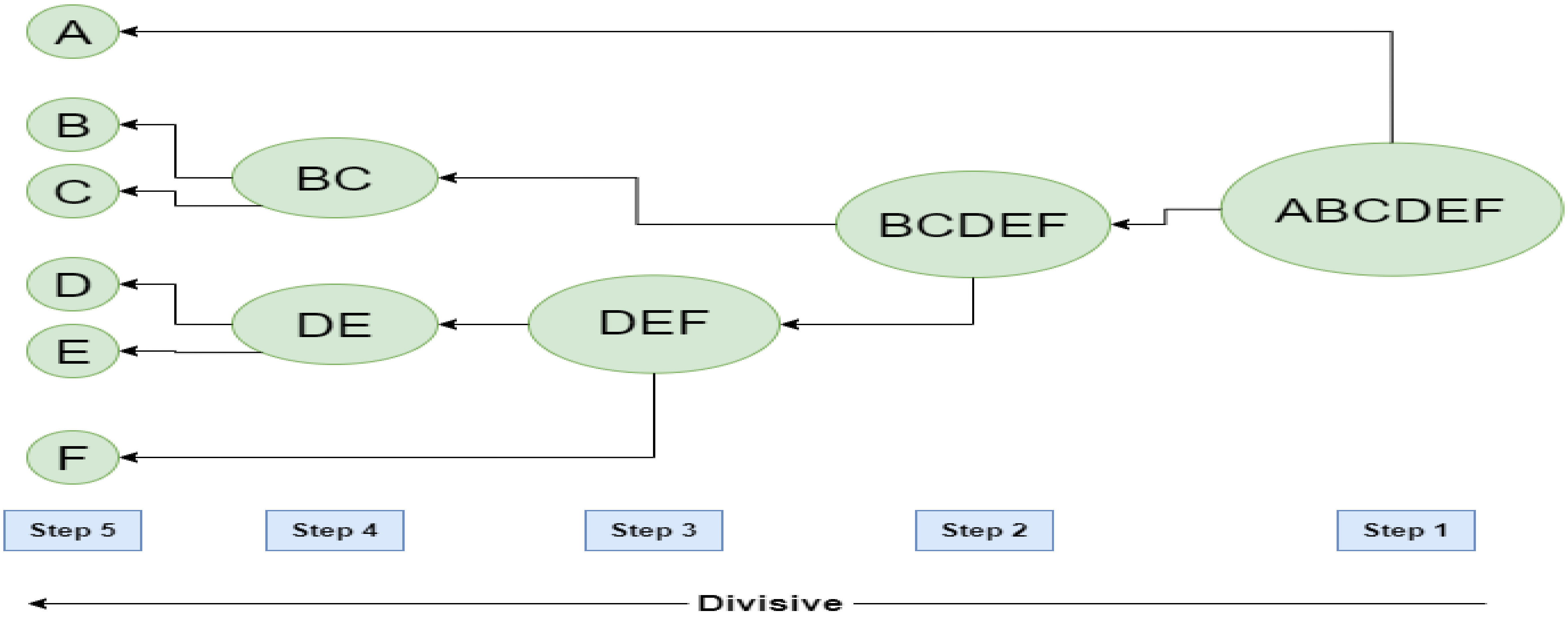
	P2,P5,P1	P3,P6,P4
P2,P5,P1	0	
P3,P6,P4	0.39	0

→ [P3,P6,P4,P2,P5,P1]

***Merging [P3,P6,P4] & [P2,P5,P1] clusters to form one cluster [P3,P6,P4,P2,P5,P1]**

Hierarchical clustering- divisive clustering

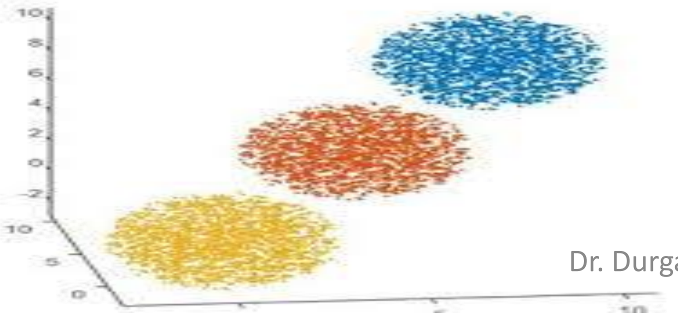
- The divisive method starts with one cluster with all given objects and then splits it iteratively to form smaller clusters

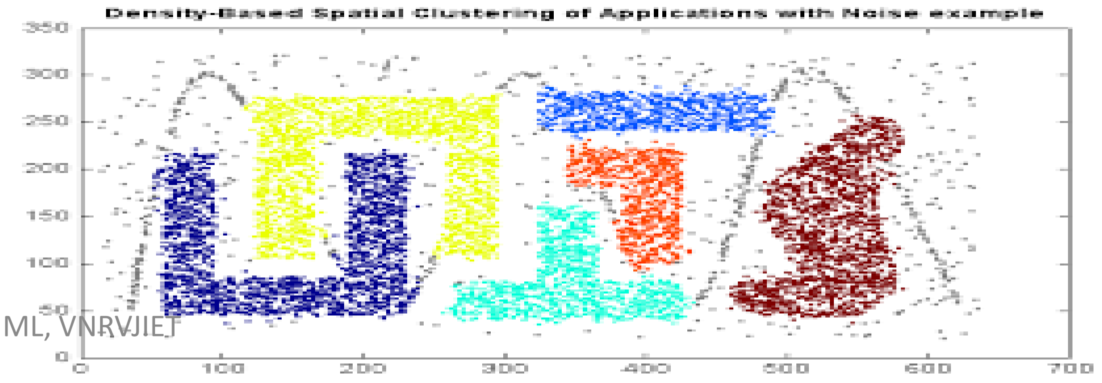


Density-Based Spatial Clustering of Applications with Noise clustering (DBSCAN)

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

- When we used the partitioning and hierarchical clustering methods, the resulting clusters are spherical or nearly spherical in nature.
- In the case of the other shaped clusters such as **S-shaped** or **uneven shaped clusters**, the above two types of method do not provide accurate results.
- The **density-based clustering approach** provides a solution to identify **clusters of arbitrary shapes.**
- The principle is based on identifying the dense area and sparse area within the data set and then run the clustering algorithm.
- DBSCAN is one of the popular density-based algorithm which creates clusters by using connected regions with high density.





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

- DBSCAN is a different type of clustering algorithm with some unique advantages.
- As the name indicates, this method focuses more on the proximity and density of observations to form clusters.
- This is very different from KMeans, where an observation becomes a part of cluster represented by nearest centroid.
- DBSCAN clustering can identify outliers, observations which won't belong to any cluster.
- Since DBSCAN clustering identifies the number of clusters as well, it is very useful with unsupervised learning of the data when we don't know how many clusters could be there in the data.

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

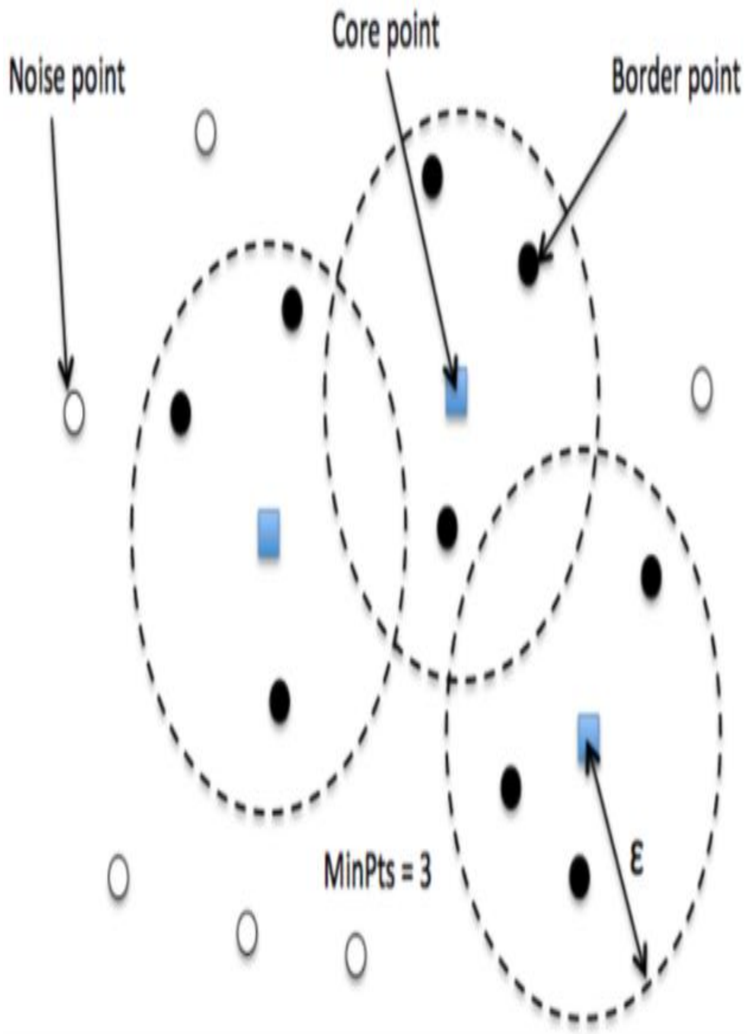
- Density-Based Clustering refers to one of the most popular **unsupervised learning** methodologies used **in model building** and machine learning algorithms.
- Density-Based Clustering refers to unsupervised machine learning methods that **identify distinctive clusters in the data**, based on the idea that a cluster/group in a data space is a contiguous region of **high point density,** separated from other clusters by **sparse regions.**
- The data points in the sparse regions are typically considered **noise/outliers.**
- The ***DBSCAN algorithm*** is based on this intuitive notion of "**clusters**" and "**noise**".
- It can discover clusters of **different shapes** and **sizes** from a large amount of data, which is containing **noise** and **outliers.**

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

In DBSCAN, clustering happens based on **two important parameters** viz.,

Epsilon - Ɛ (abbreviated as ***eps***):

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



MinPts: Minimum number of points

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

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

In this algorithm, we have 3 types of data points

<u>Core Point:</u>

*A point is a core point if it has more than or equal (at least one) to **MinPts points within eps** from itself*

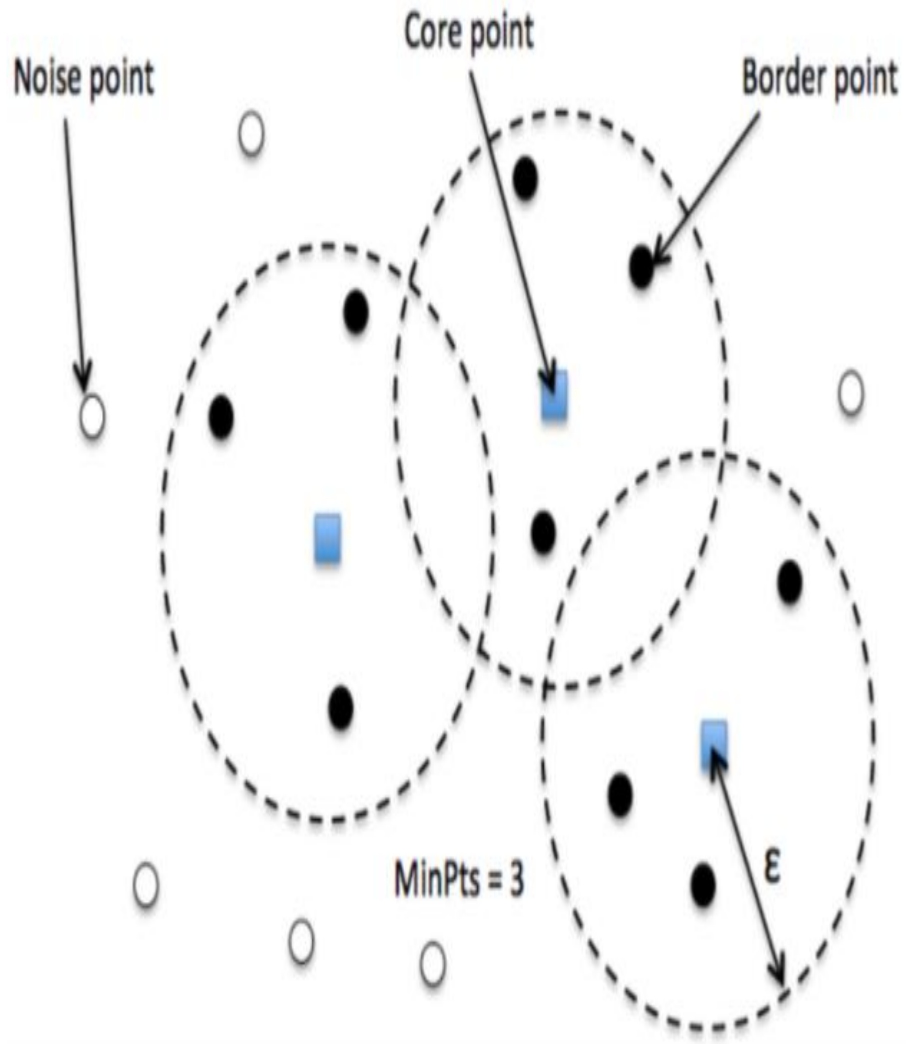
<u>Border Point:</u>

*A point which has fewer (at least one) than **MinPts** within eps but it is in the neighborhood of **a core point.** **Or***

This is a point that has **at least one** **Core point** *within eps*

<u>Noise</u> *or* ***<u>outlier:</u>***

A point which is not a core point or border point and it has **less than** ***MinPts*** *within eps* .



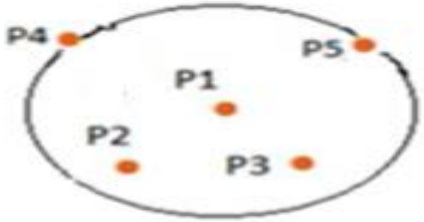
Density-Based Spatial Clustering of Applications with Noise clustering (DBSCAN)

DBSCAN divide data points into three types MinPts=4 Eps=2 Unit

1) Core Point 2) Noise Point 3) Border Point

1) Core Point

p is core point if **{q |dist(p, q)<=Eps}>=MinPts**



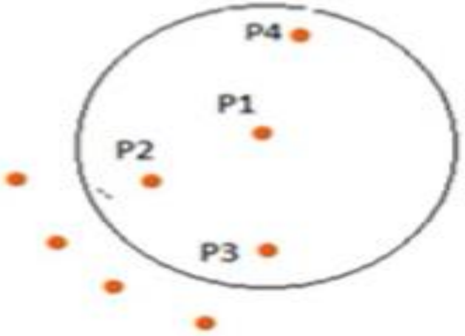
2) Noise Point

p is noise point **if {q |dist(p, q)<=Eps}<MinPts**

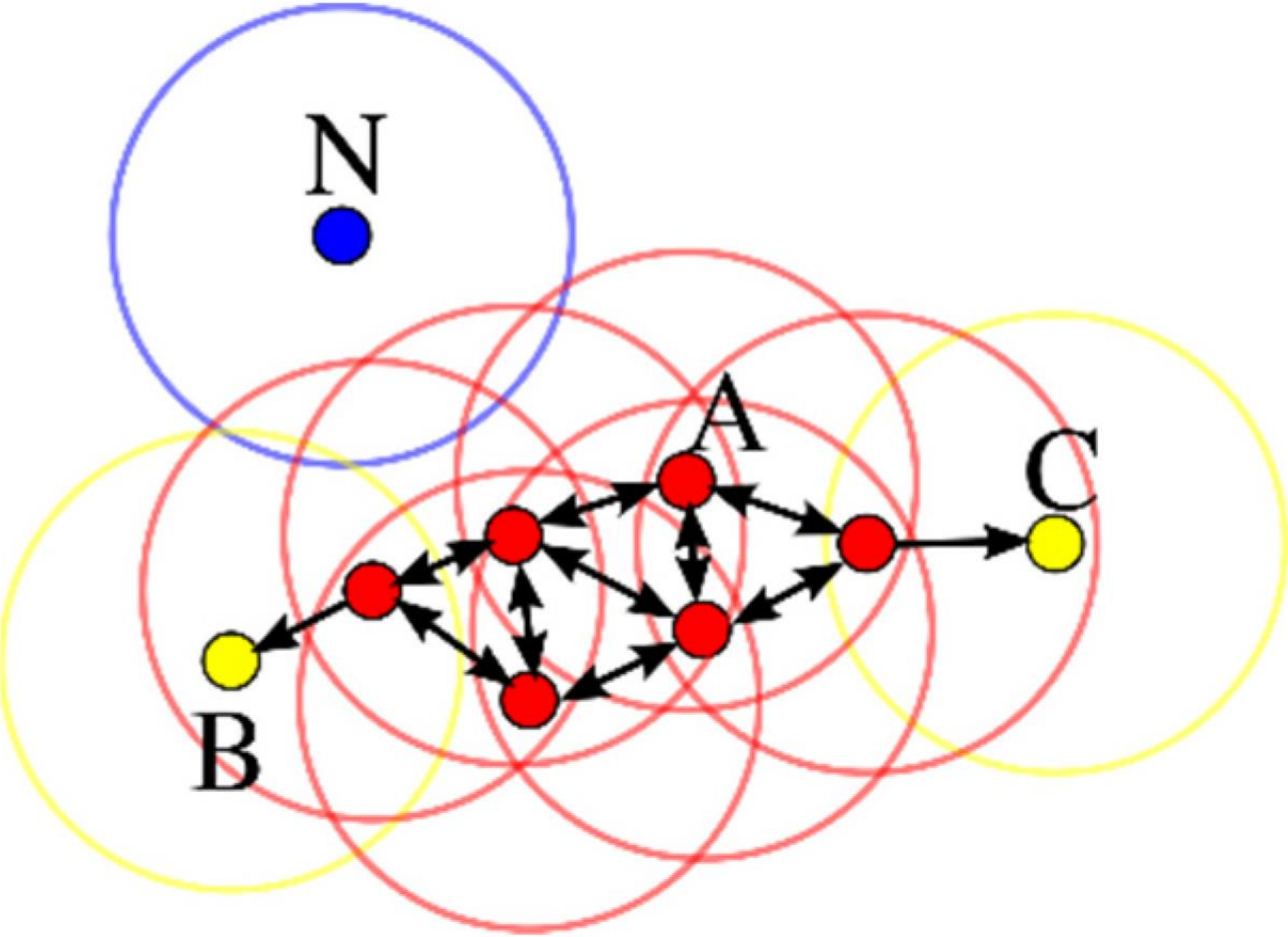
3) Border Point:

Point q is border point if **dist(p, q)<=Eps**

And p is core point while q is noise point



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



Overview of Density based clusters

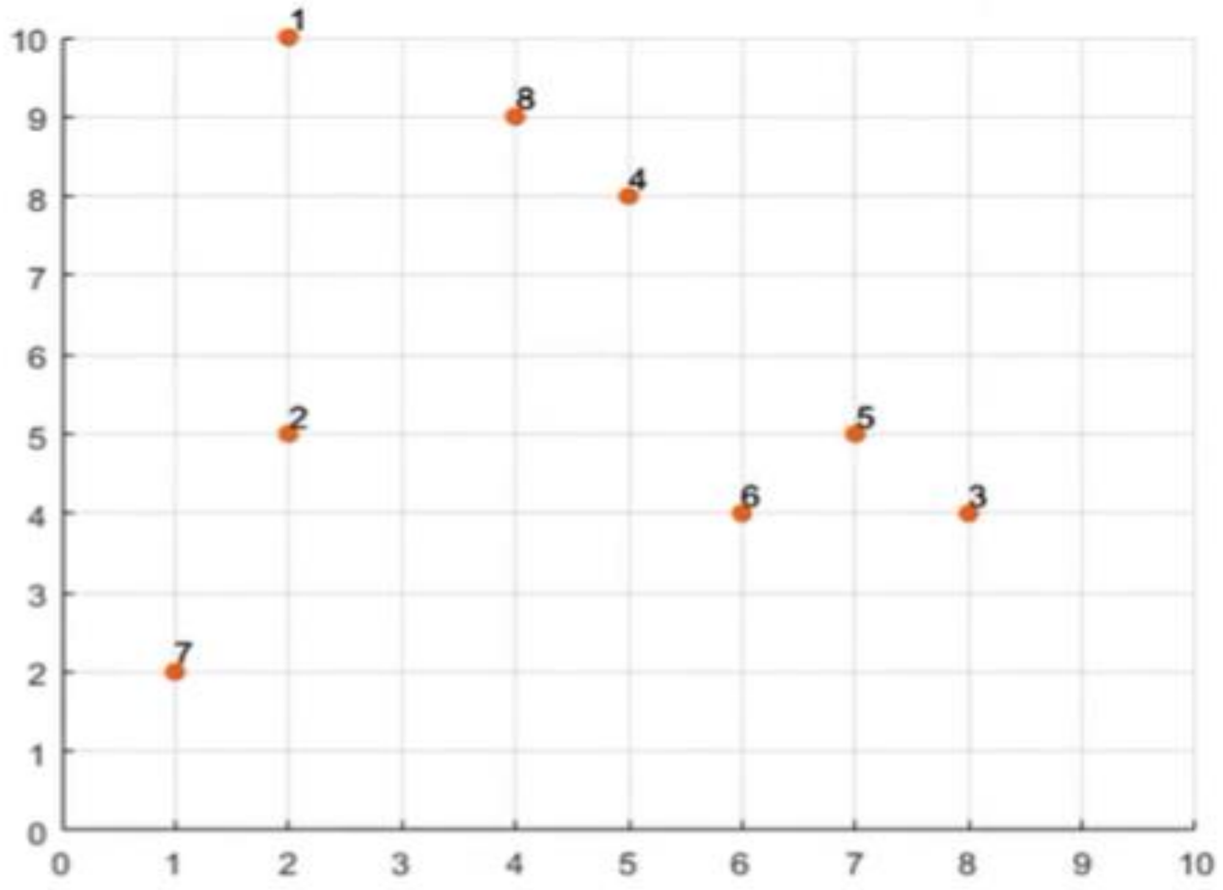
Density-Based Spatial Clustering of Applications with Noise clustering (DBSCAN)

DBSCAN Algorithm

- The algorithm proceeds by arbitrarily picking up a point in the dataset (until all points have been visited).
- If there are at least ‘minPoint’ points within a radius of ‘ε’ to the point then we consider all these points to be part of the same cluster.
- The clusters are then expanded by recursively repeating the neighborhood calculation for each neighboring point

Density-Based Spatial Clustering of Applications with Noise clustering (DBSCAN) – **Example-1**

Point	X	Y
P1	2	10
P2	2	5
P3	8	4
P4	5	8
P5	7	5
P6	6	4
P7	1	2
P8	4	9



Eps=2 MinPts=3

Density-Based Spatial Clustering of Applications with Noise clustering (DBSCAN) – **Example-1**

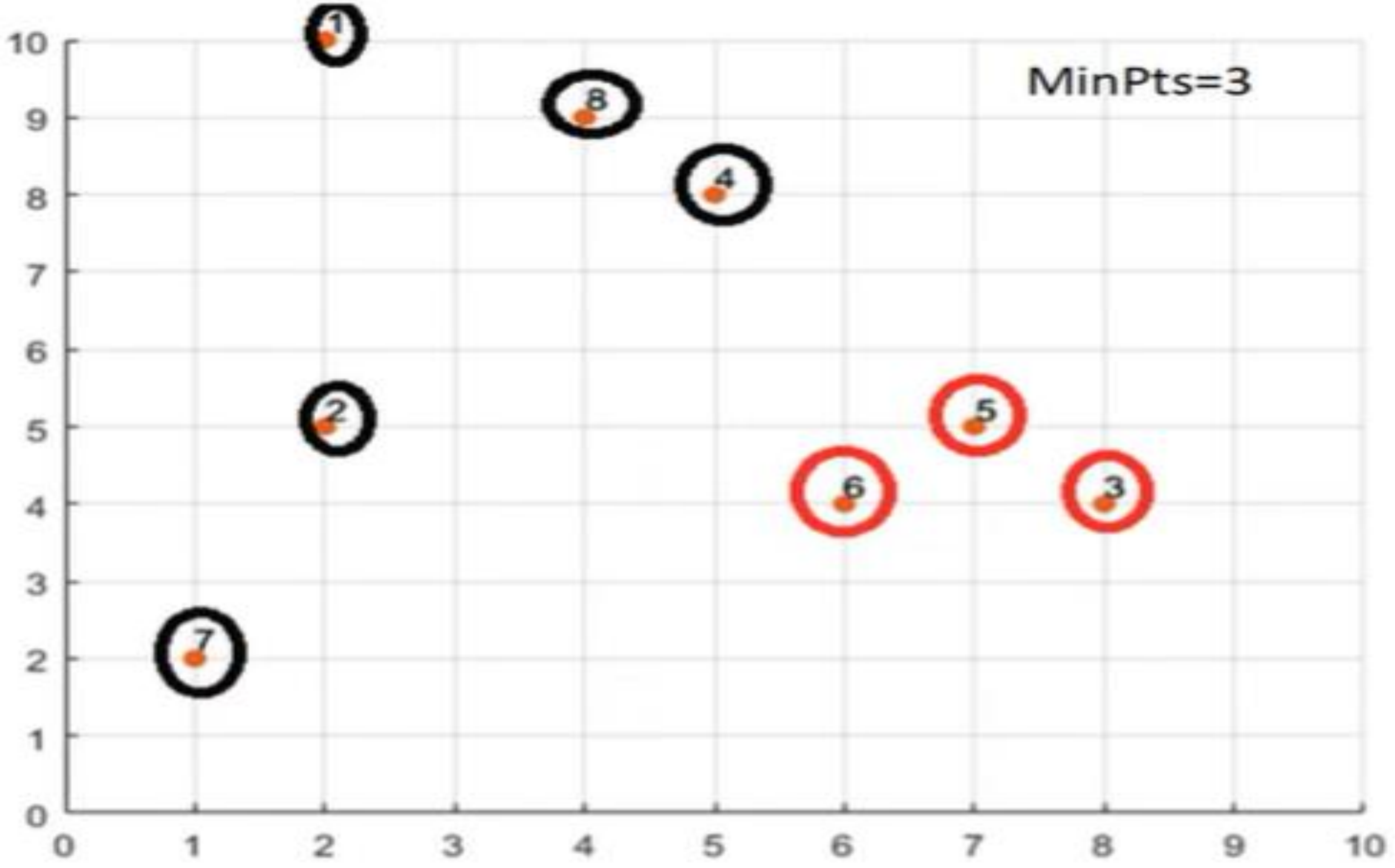
P1	0.00							
P2	5.00	0.00						
P3	8.49	6.08	0.00					
P4	3.61	4.24	5.00	0.00				
P5	7.07	5.00	1.41	3.61	0.00			
P6	7.21	4.12	2.00	4.12	1.41	0.00		
P7	8.06	3.16	7.28	7.21	6.71	5.39	0.00	
P8	2.24	4.47	6.40	1.41	5.00	5.39	7.62	0.00
	P1	**P2**	**P3**	**P4**	**P5**	**P6**	**P7**	**P8**

Eps=2

MinPts=3

P1:	P2:	P3:P5,P6	P4: P8
P5:P3,P6	P6:P3, P5	P7:	P8:P4

Density-Based Spatial Clustering of Applications with Noise clustering (DBSCAN) – **Example-1**

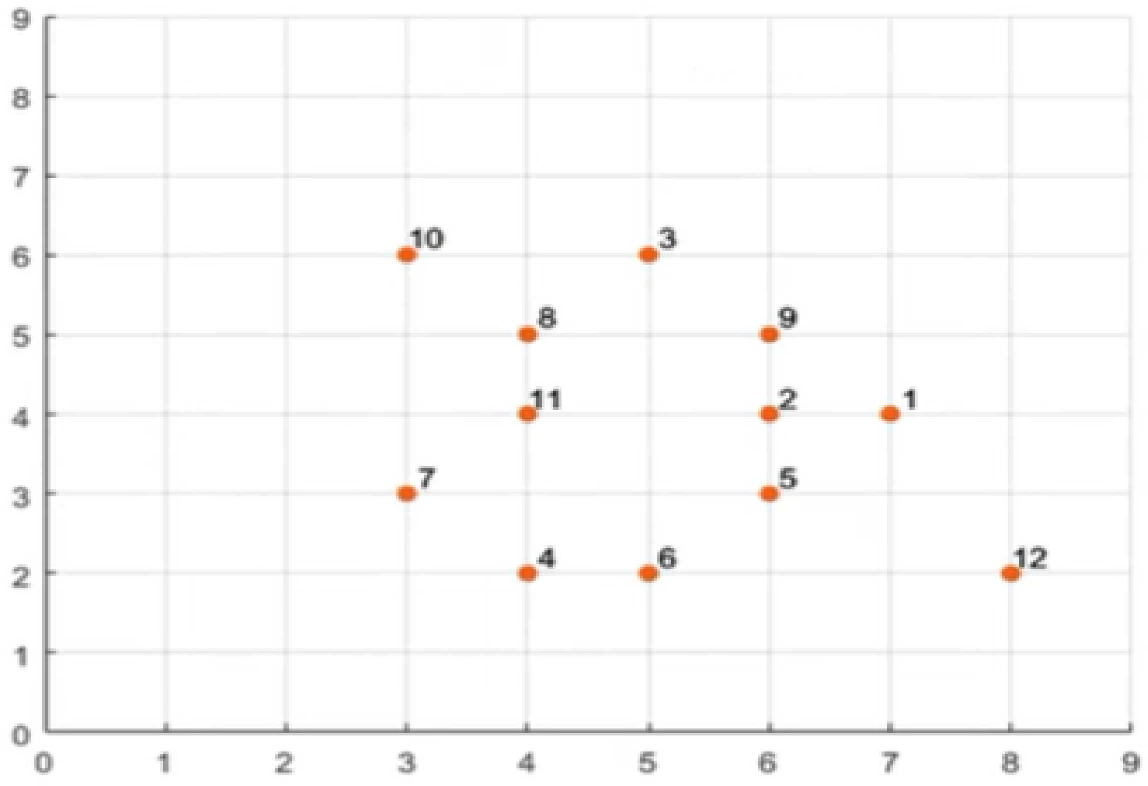


Point	Status	
P1	Noise	
P2	Noise	
P3	Core	
P4	Noise	
P5	Core	
P6	Core	
P7	Noise	
P8	Noise	

P1:	P2:	**P3:P5,P6**	P4:P8
P5:P3,P6	**P6:P3, P5**	P7:	P8:P4

Density-Based Spatial Clustering of Applications with Noise clustering (DBSCAN) – **Example-2**

Point	X	Y
P1	7	4
P2	6	4
P3	5	6
P4	4	2
P5	6	3
P6	5	2
P7	3	3
P8	4	5
P9	6	5
P10	3	6
P11	4	4
P12	8	2



Eps=1.9 MinPts=4

Density-Based Spatial Clustering of Applications with Noise clustering (DBSCAN) – **Example-2**

P1	0.00											
P2	1.00	0.00										
P3	2.83	2.24	0.00									
P4	3.61	2.83	4.12	0.00								
P5	1.41	1.00	3.16	2.24	0.00							
P6	2.83	2.24	4.00	1.00	1.41	0.00						
P7	4.12	3.16	3.61	1.41	3.00	2.24	0.00					
P8	3.16	2.24	1.41	3.00	2.83	3.16	2.24	0.00				
P9	1.41	1.00	1.41	3.61	2.00	3.16	3.61	2.00	0.00			
P10	4.47	3.61	2.00	4.12	4.24	4.47	3.00	1.41	3.16	0.00		
P11	3.00	2.00	2.24	2.00	2.24	2.24	1.41	1.00	2.24	2.24	0.00	
P12	2.24	2.83	5.00	4.00	2.24	3.00	5.10	5.00	3.61	6.40	4.47	0.00
	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12

Eps=1.9

MinPts=4

Density-Based Spatial Clustering of Applications with Noise clustering (DBSCAN) – **Example-2**

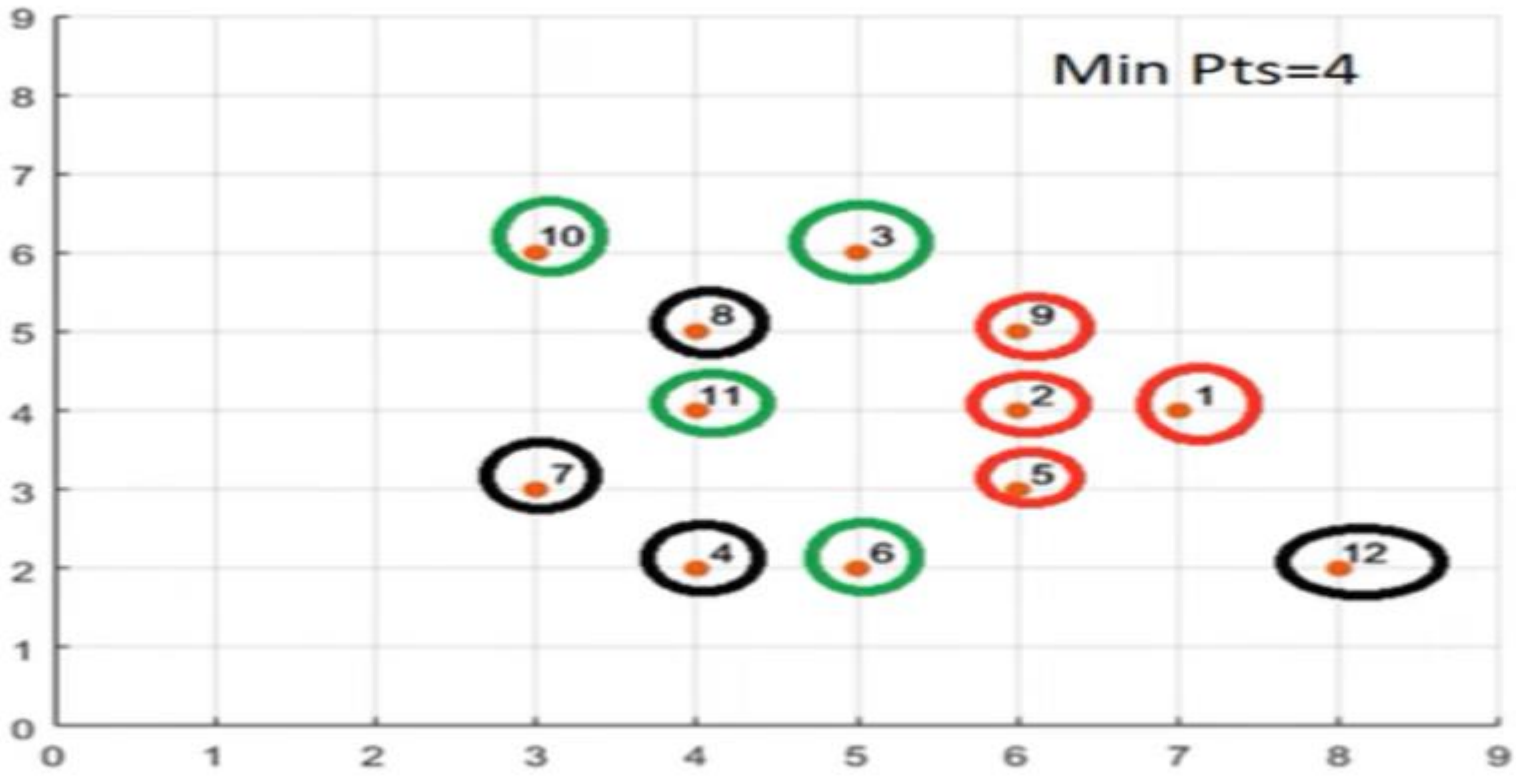
P1	0.00											
P2	1.00	0.00										
P3	2.83	2.24	0.00									
P4	3.61	2.83	4.12	0.00								
P5	1.41	1.00	3.16	2.24	0.00							
P6	2.83	2.24	4.00	1.00	1.41	0.00						
P7	4.12	3.16	3.61	1.41	3.00	2.24	0.00					
P8	3.16	2.24	1.41	3.00	2.83	3.16	2.24	0.00				
P9	1.41	1.00	1.41	3.61	2.00	3.16	3.61	2.00	0.00			
P10	4.47	3.61	2.00	4.12	4.24	4.47	3.00	1.41	3.16	0.00		
P11	3.00	2.00	2.24	2.00	2.24	2.24	1.41	1.00	2.24	2.24	0.00	
P12	2.24	2.83	5.00	4.00	2.24	3.00	5.10	5.00	3.61	6.40	4.47	0.00
	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12

Eps=1.9

MinPts=4

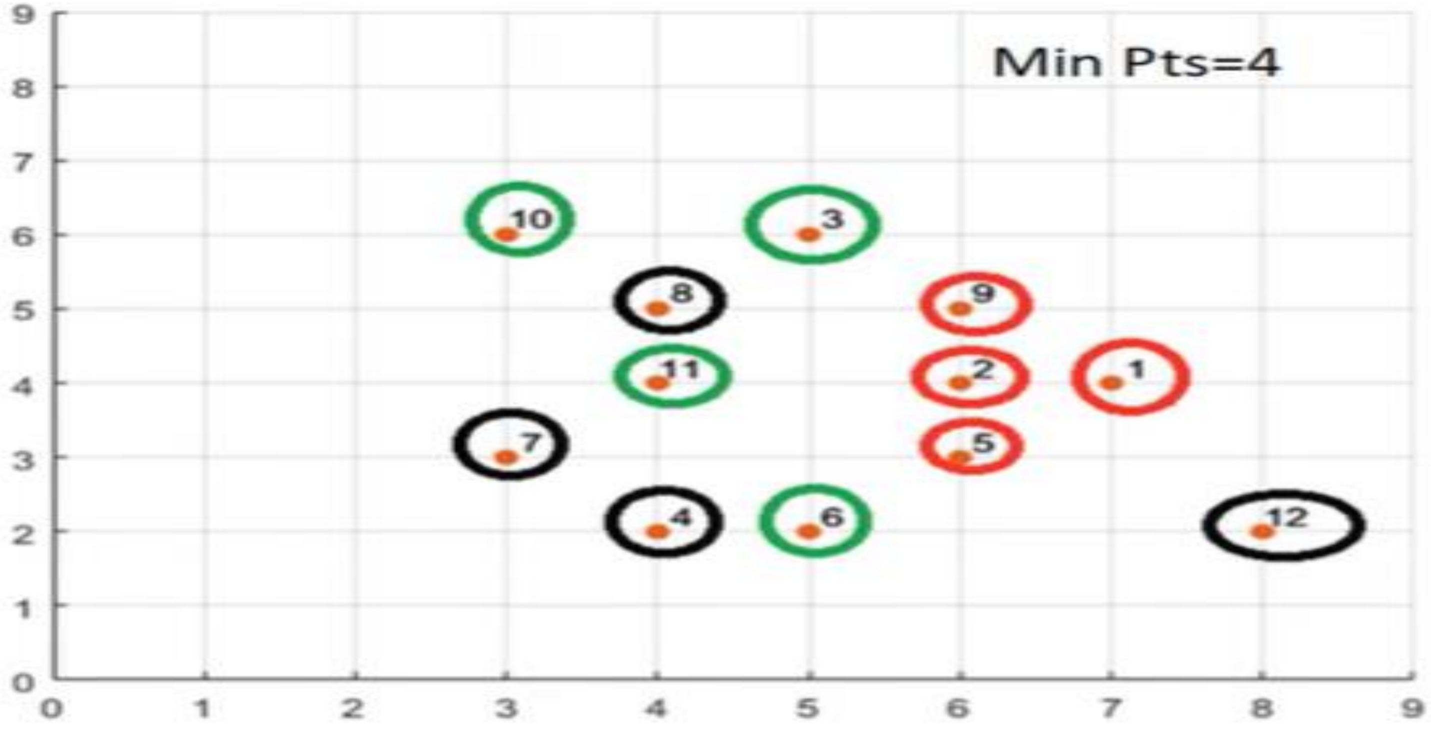
P1: P2,P5, P9	**P2:P1, P5, P9**	P3:P8, P9	P4: P6, P7
P5:P1,P2,P6	P6:P4, P5	P7: P4, P11	**P8:P3, P10, P11**
P9:P1, P2, P3	P10:P8	P11: P7, P8	P12:

Density-Based Spatial Clustering of Applications with Noise clustering (DBSCAN) – **Example-2**



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

Density-Based Spatial Clustering of Applications with Noise clustering (DBSCAN) – **Example-2**



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

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

Applications Of Unsupervised Learning :

- **Segmentation** of target consumer populations by an advertisement consulting agency on the basis of few dimensions such as **demography, financial data, purchasing habits**, etc. so that the advertisers can reach their **target consumers** efficiently
- Anomaly or fraud detection in the banking sector by identifying the pattern of loan defaulters
- Image processing and image segmentation such as **face recognition, expression identification,** etc.
- Grouping of important characteristics in **genes** to identify important influencers in new areas of **genetics**
- Utilization by data scientists to reduce the dimensionalities in sample data to simplify modelling
- Document clustering and identifying potential labelling options