Clustering

Clustering

- **Clustering** is an unsupervised machine learning technique used to group similar data points together based on their characteristics. It is useful in exploratory data analysis to uncover patterns and relationships in data. The fundamental concepts of **similarity** and **dissimilarity** are crucial in clustering, as they define how data points are grouped.

- Clustering or cluster analysis is a machine learning technique, which groups the unlabelled dataset.

- It does it by finding some similar patterns in the unlabelled dataset such as shape, size, color, behavior, etc., and divides them as per the presence and absence of those similar patterns.

- It is an unsupervised learning method, hence no supervision is provided to the algorithm, and it deals with the unlabeled dataset.

- After applying this clustering technique, each cluster or group is provided with a cluster-ID. ML system can use this id to simplify the processing of large and complex datasets.

- The clustering technique is commonly used for **statistical data analysis.**

Example: Let's understand the clustering technique with the real-world example of Mall:

    - When we visit any shopping mall, we can observe that the things with similar usage are grouped together.

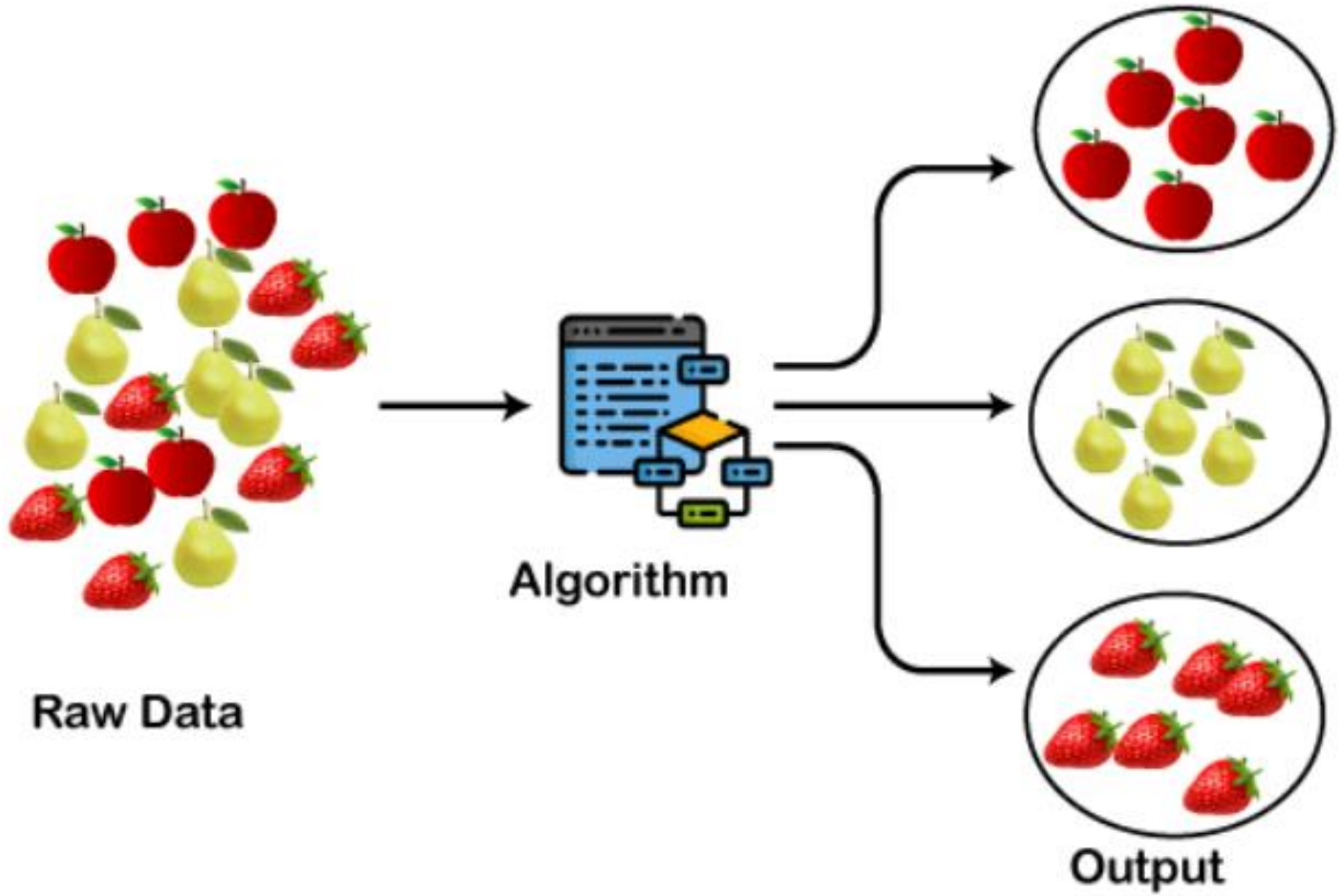
    - Such as the t-shirts are grouped in one section, and trousers are at other sections, similarly, at vegetable sections, apples, bananas, Mangoes, etc., are grouped in separate sections, so that we can easily find out the things.

    - The clustering technique also works in the same way. Other examples of clustering are grouping documents according to the topic.

- The clustering technique can be widely used in various tasks. Some most common uses of this technique are:
    - Market Segmentation
    - Statistical data analysis
    - Social network analysis
    - Image segmentation
    - Anomaly detection, etc.

- Apart from these general usages, it is used by the Amazon in its recommendation system to provide the recommendations as per the past search of products.

- Netflix also uses this technique to recommend the movies and web-series to its users as per the watch history.


Raw Data
Algorithm
Output

Similarity in Clustering:

- Similarity measures how close or alike two data points are. It is often quantified using various distance or similarity metrics.
- **Common Similarity Measures**:
  - **Euclidean Distance**: Measures the straight-line distance between two points in a multi-dimensional space.
  - **Cosine Similarity**: Measures the cosine of the angle between two vectors in a space, often used in text analysis.
  - **Jaccard Similarity**: Measures the similarity between two sets by dividing the size of the intersection by the size of the union.
  - **Pearson Correlation Coefficient**: Measures the linear correlation between two variables.

Dissimilarity in Clustering

- Dissimilarity measures how different two data points are from each other. It is often referred to as "distance" in a data space.
- **Common Dissimilarity Measures**:
  - **Manhattan Distance**: Measures the distance between two points by summing the absolute differences of their coordinates.
  - **Mahalanobis Distance**: Takes into account correlations between variables and scales the data by their variance before calculating the distance.
  - **Hamming Distance**: Used for categorical or binary data, measuring the number of positions at which two sequences differ.

Minimum within cluster distance criteria

The **minimum within-cluster distance criteria** aims to ensure that data points within a cluster are as similar as possible, which means minimizing the distance between all points in the same cluster. This leads to tighter, more cohesive clusters. In many clustering algorithms, this criterion plays a crucial role in optimizing the quality of the clusters.

Explanation of Minimum Within-Cluster Distance:

1. **Within-Cluster Distance**: It is a measure of how far apart the data points within a single cluster are from each other. The goal is to minimize this distance to ensure that all points in a cluster are closely related or similar.

2. **How It Works in Clustering Algorithms**:

- In algorithms like **K-means**, the cluster assignment is done based on minimizing the sum of squared distances between each data point and the cluster's centroid (center of mass). This is known as **within-cluster sum of squares (WCSS)**.

- In **Hierarchical clustering**, especially in the agglomerative approach, clusters are merged in such a way that the intra-cluster (within-cluster) distance is minimized after each step.

Mathematically WCSS

- For a cluster Ci*Ci*, the **within-cluster distance** is often calculated as the sum of distances between each point in the cluster and the centroid μi*μi* of that cluster:
- WCSS=∑xj∈Ci∥xj−μi∥2*WCSS*=*xj*∈*Ci*∑∥*xj*−*μi*∥2
- Where:
- Xj represents the data points in the cluster.
- μi*μi* represents the centroid of cluster Ci.
- ∥xj−μi∥^2 is the squared distance between a data point and the centroid.
- The **goal** is to minimize this value across all clusters in a dataset.
- **Key Points:**
- Minimizing within-cluster distance improves the **homogeneity** of the cluster, ensuring that points in the cluster are more similar to each other.

Continued…

- Lower within-cluster distance leads to compact clusters, but there is often a trade-off between this and creating **well-separated clusters** (low between-cluster distance).
- **Elbow method** in K-means is commonly used to find the optimal number of clusters by looking at the point where the WCSS begins to diminish significantly (i.e., the "elbow" point).
- **Challenges:**
- Minimizing within-cluster distance may not always guarantee the best clustering solution if the data distribution is not well-suited for the algorithm. For example, in non-spherical clusters, minimizing WCSS might not result in meaningful clusters.

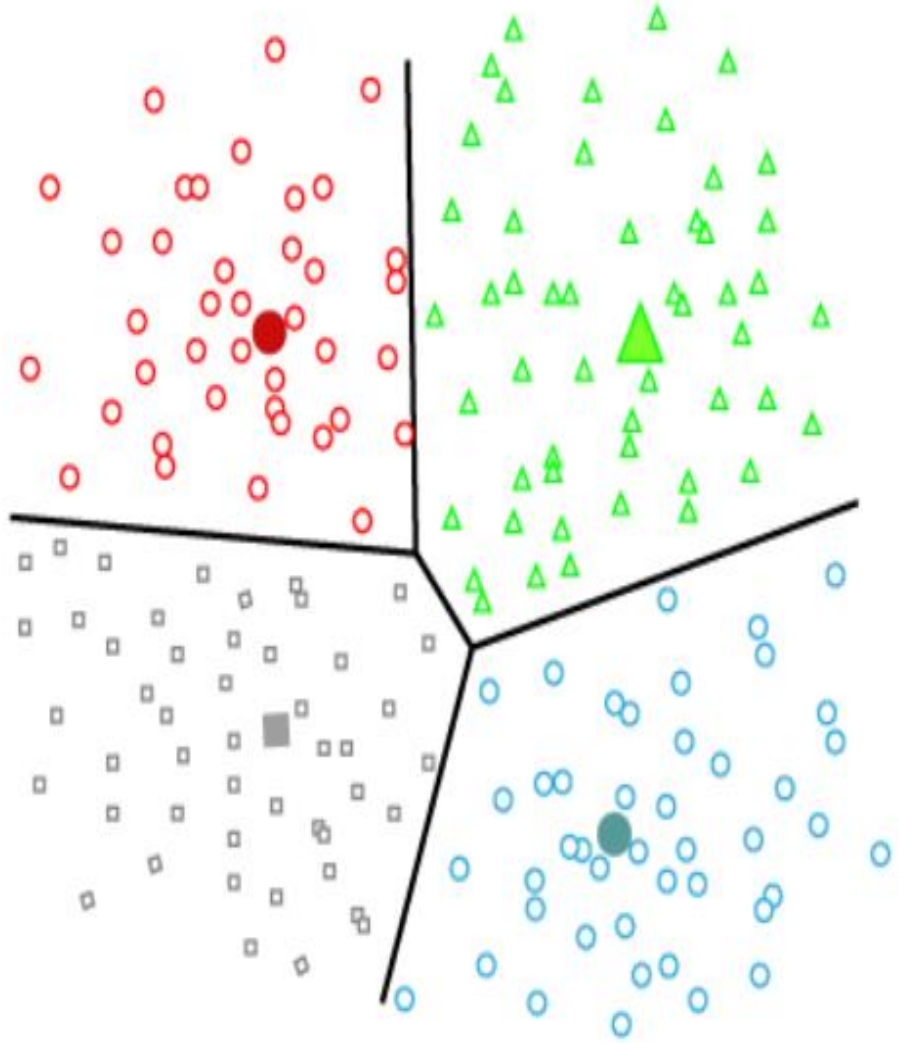
Types of Clustering Methods

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

- But there are also other various approaches of Clustering exist. Below are the main clustering methods used in Machine learning:
  1. Partitioning Clustering
  2. Density-Based Clustering
  3. Distribution Model-Based Clustering
  4. Hierarchical Clustering
  5. Fuzzy Clustering

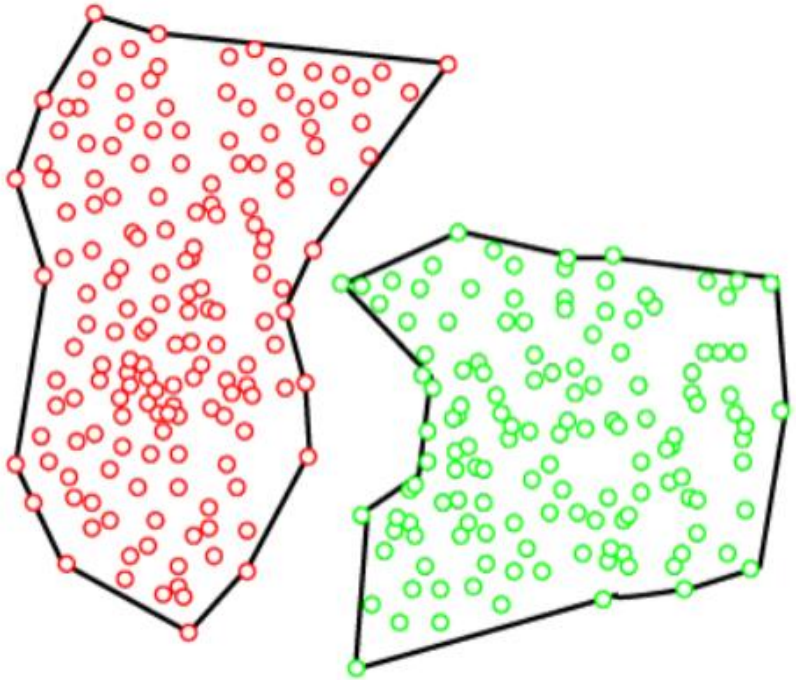
Partitioning Clustering

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



Density-Based Clustering

- The density-based clustering method connects the highly-dense areas into clusters, and the arbitrarily shaped distributions are formed as long as the dense region can be connected.



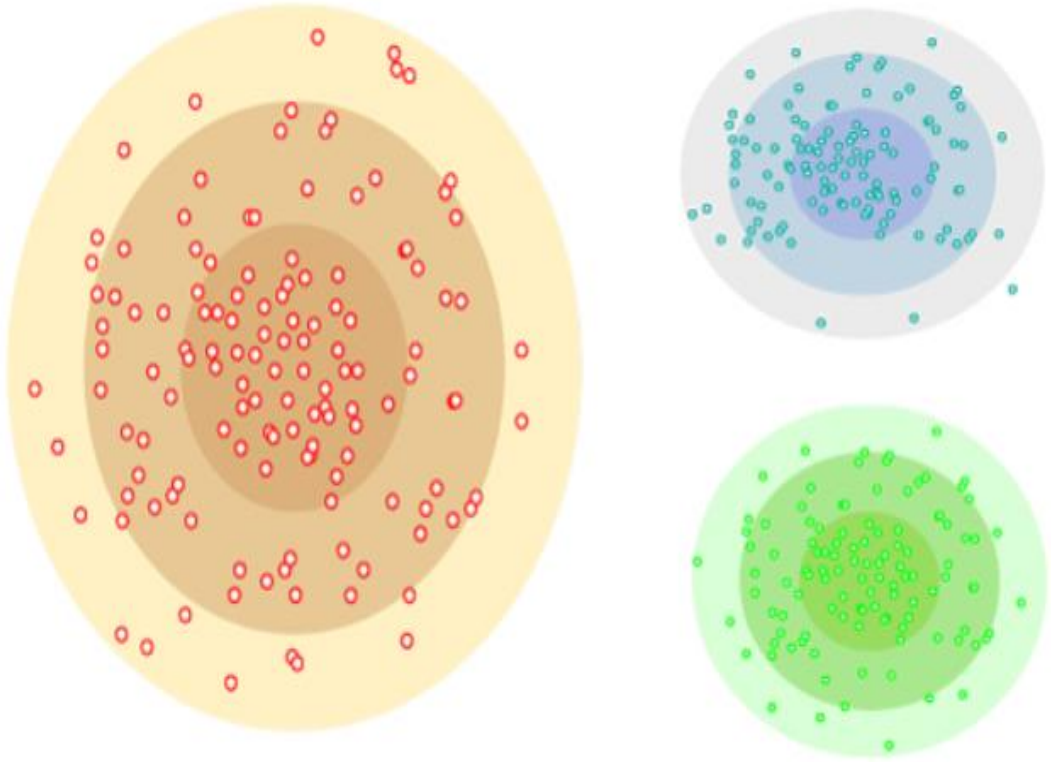
- This algorithm does it by identifying different clusters in the dataset and connects the areas of high densities into clusters.
- The dense areas in data space are divided from each other by sparser areas.
- These algorithms can face difficulty in clustering the data points if the dataset has varying densities and high dimensions.
- Example: DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

Model-Based Clustering

- This is a form of clustering which assumes that the data comes from a particular probability model.

- The model is based on 3 general assumptions:

1. We know the number of clusters before we start
2. Each observation in the data as a certain probability of belonging to each cluster.
3. The observations within each cluster follow a normal distribution (with the appropriate dimension)



- These assumptions leave us with two problems to solve when fitting the model:

    1. What are the means and covariances of each of the clusters?
    2. Which cluster does each observation belong to?

- The example of this type is the **Expectation-Maximization Clustering algorithm** that uses Gaussian Mixture Models (GMM).

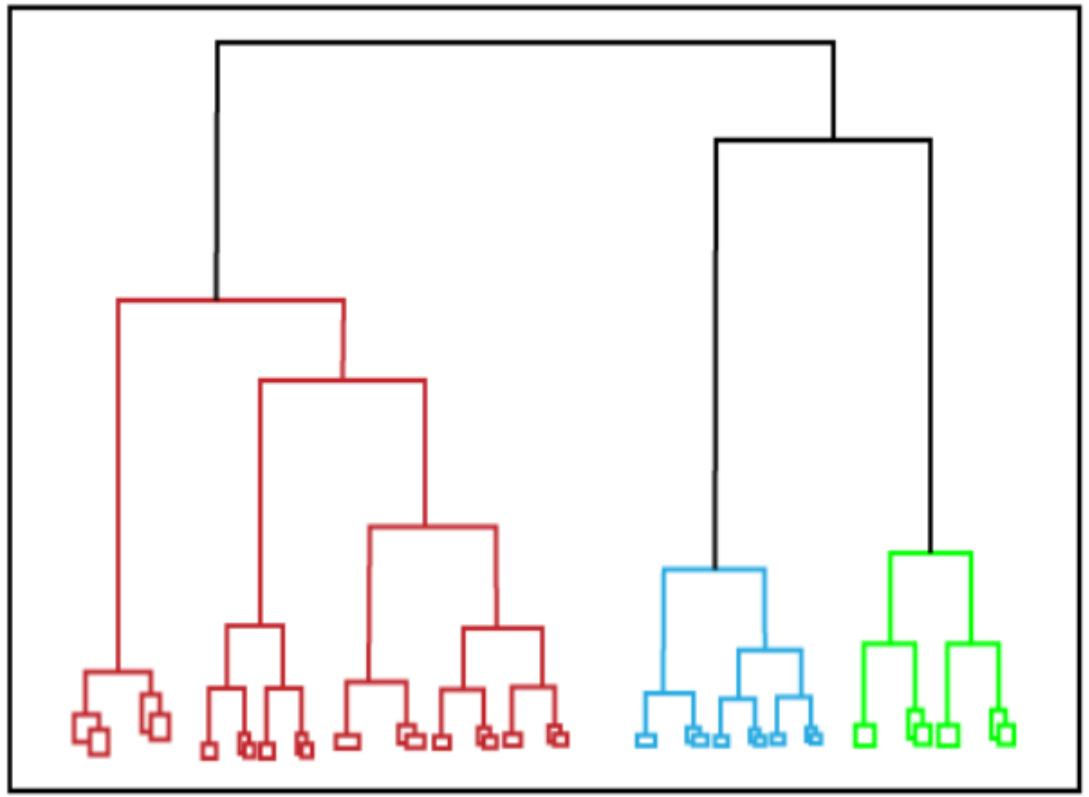
Hierarchical Clustering

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

- In this technique, the dataset is divided into clusters to create a tree-like structure, which is also called a **dendrogram**.

- The observations or any number of clusters can be selected by cutting the tree at the correct level.

- The most common example of this method is the **Agglomerative Hierarchical algorithm**.



Fuzzy Clustering

- Fuzzy clustering is a type of soft method in which a data object may belong to more than one group or cluster.

- Each dataset has a set of membership coefficients, which depend on the degree of membership to be in a cluster.

- **Fuzzy C-means algorithm** is the example of this type of clustering; it is sometimes also known as the Fuzzy k-means algorithm.

K Means Clustering

How does the K-Means Algorithm Work?

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

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

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

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

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

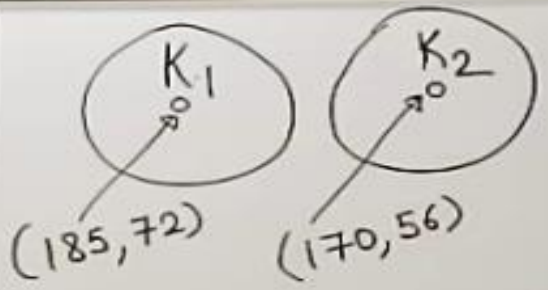
Step-5: Repeat the third steps, which means reassign each datapoint to the new closest centroid of each cluster.

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

Step-7: The model is ready.

K-means Algorithm

	Height	weight
①	185	72
②	170	56
③	168	60
④	179	68
⑤	182	72
⑥	188	77
⑦	180	71
⑧	180	70
⑨	183	84
⑩	180	88
⑪	180	67
⑫	177	76

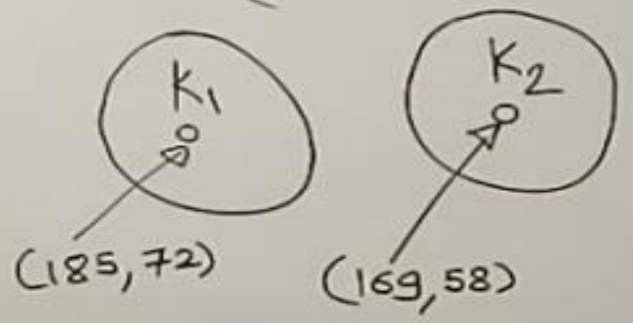


ED for ③ → K_1 → $\sqrt{(168-185)^2+(60-72)^2}$ = 20.80

→ K_2 → $\sqrt{(168-170)^2+(60-56)^2}$ = 4.48

New centeoid Calculation :-

for $K_2 = \left(\frac{170+168}{2}, \frac{60+56}{2}\right) = (169, 58)$



ED for ④ → $K_1 = \sqrt{(179-185)^2+(68-72)^2}$ = (6.32)

→ $K_2 = \sqrt{(179-169)^2+(68-58)^2}$ = 14.14

Euclidean Distance

$$\sqrt{(X_0-X_c)^2+(Y_0-Y_c)^2}$$

$K_1 \rightarrow \{1,4,5,6,7,8,9,10,11,12\}$

$K_2 \rightarrow \{2,3\}$

Hierarchal 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 tree-shaped structure is known as the **dendrogram**.

- Sometimes the results of K-means clustering and hierarchical clustering may look similar, but they both differ depending on how they work. As there is no requirement to predetermine the number of clusters as we did in the K-Means algorithm.

- The hierarchical clustering technique has two approaches:
  1. **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.
  2. **Divisive:** Divisive algorithm is the reverse of the agglomerative algorithm as it is a **top-down approach.**

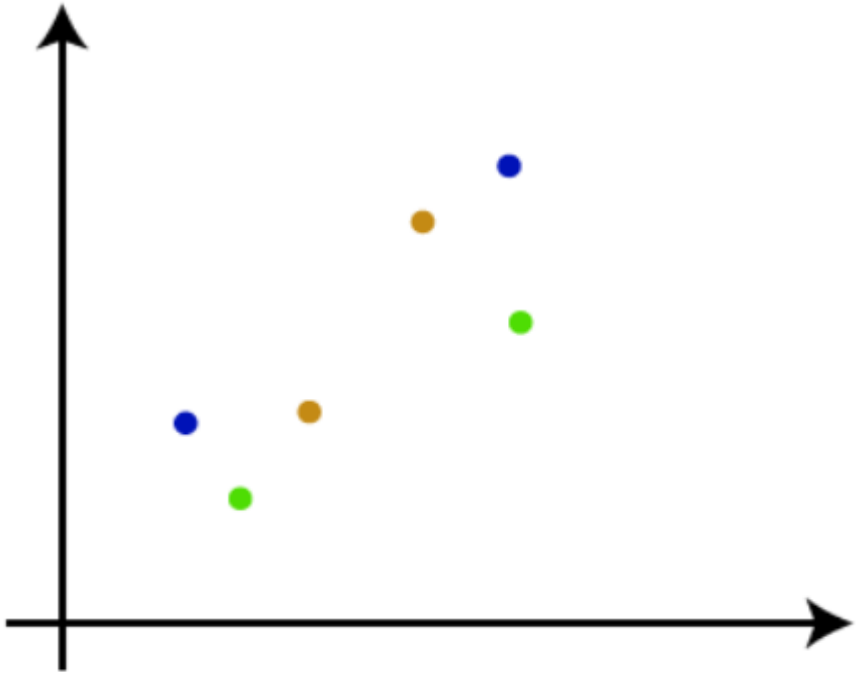
Agglomerative Hierarchical clustering

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

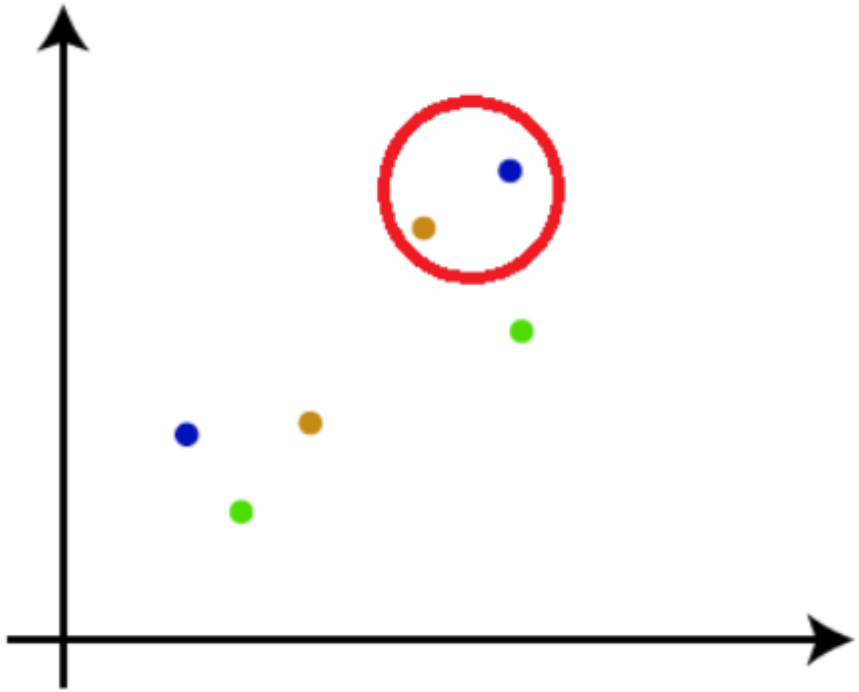
How the Agglomerative Hierarchical clustering Work?

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

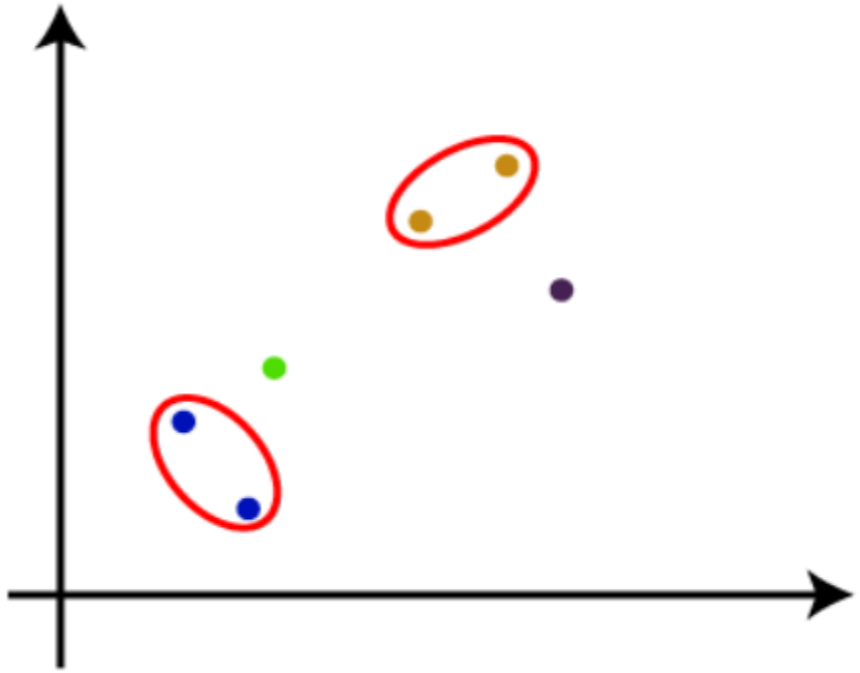
Step-1: Create each data point as a single cluster. Let's say there are N data points, so the number of clusters will also be N.



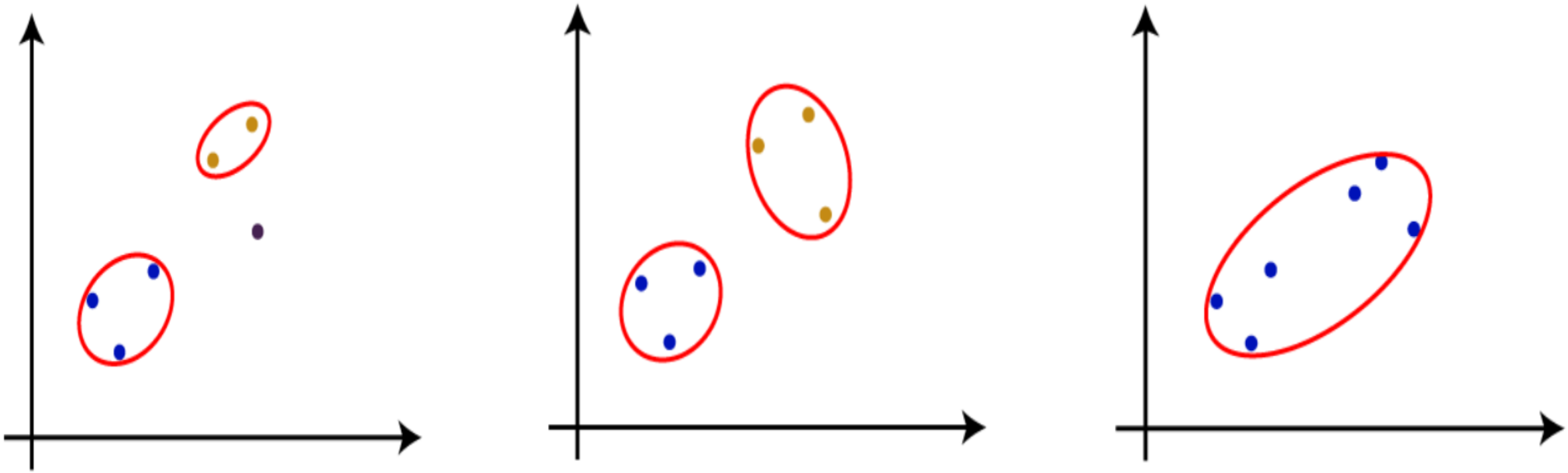
Step-2: Take two closest data points or clusters and merge them to form one cluster. So, there will now be N-1 clusters.



Step-3: Again, take the two closest clusters and merge them together to form one cluster. There will be N-2 clusters.



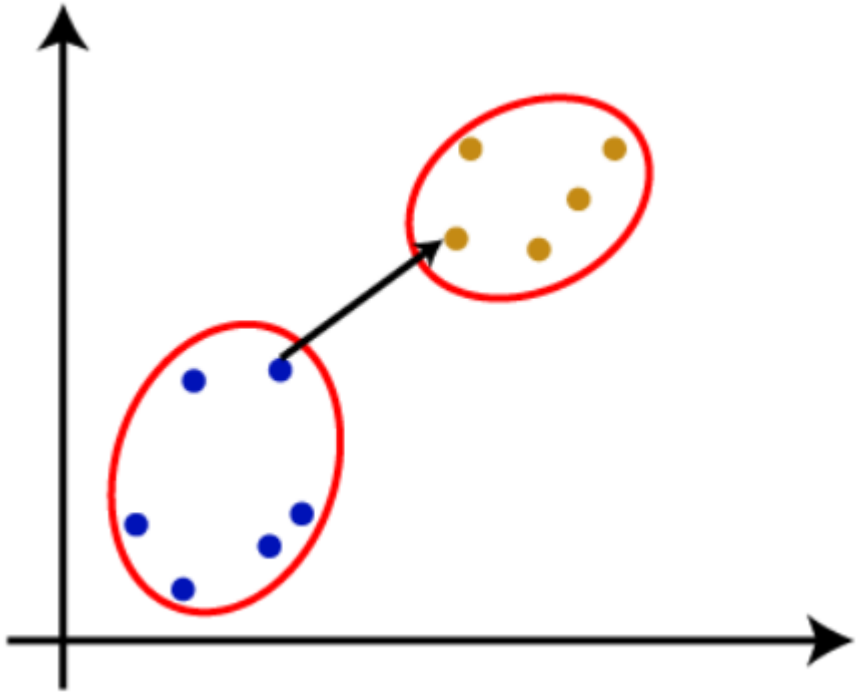
Step-4: Repeat Step 3 until only one cluster left. So, we will get the following clusters. Consider the below images:



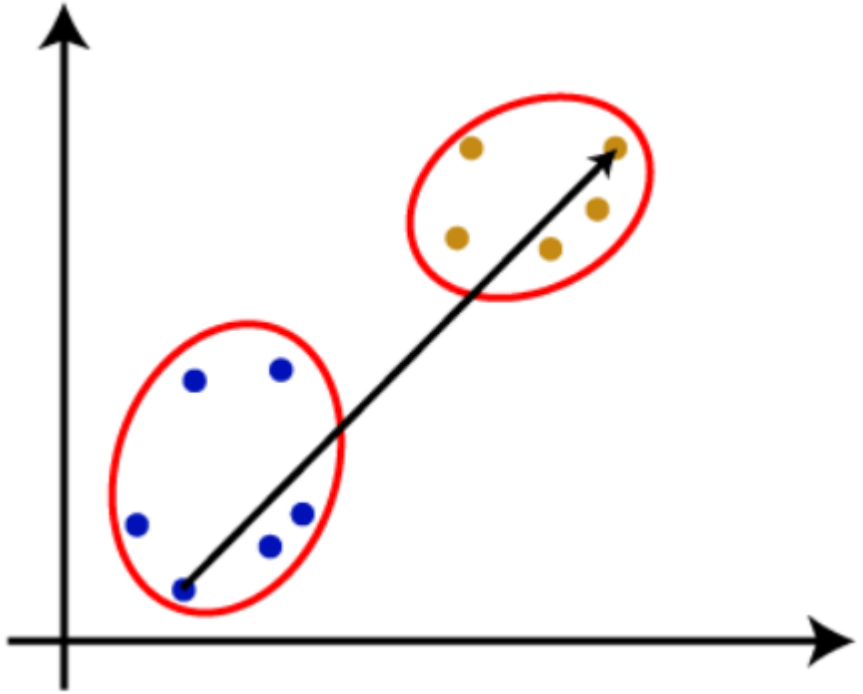
Step-5: Once all the clusters are combined into one big cluster, develop the dendrogram to divide the clusters as per the problem.

Measure for the distance between two clusters

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

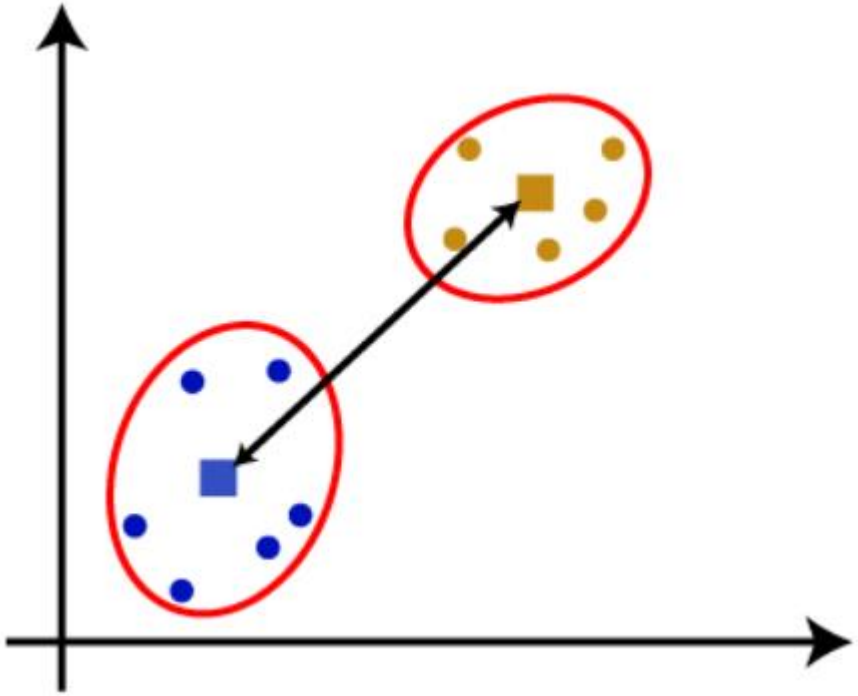


Complete Linkage: It is the farthest distance between the two points of two different clusters. It is one of the popular linkage methods as it forms tighter clusters than single-linkage.



Average Linkage: It is the linkage method in which the distance between each pair of points of two different clusters is added up and then divided by the total number of pairs to calculate the average distance between two clusters. It is also one of the most popular linkage methods.

Centroid Linkage: It is the linkage method in which the distance between the centroid of the clusters is calculated. Consider the below image:

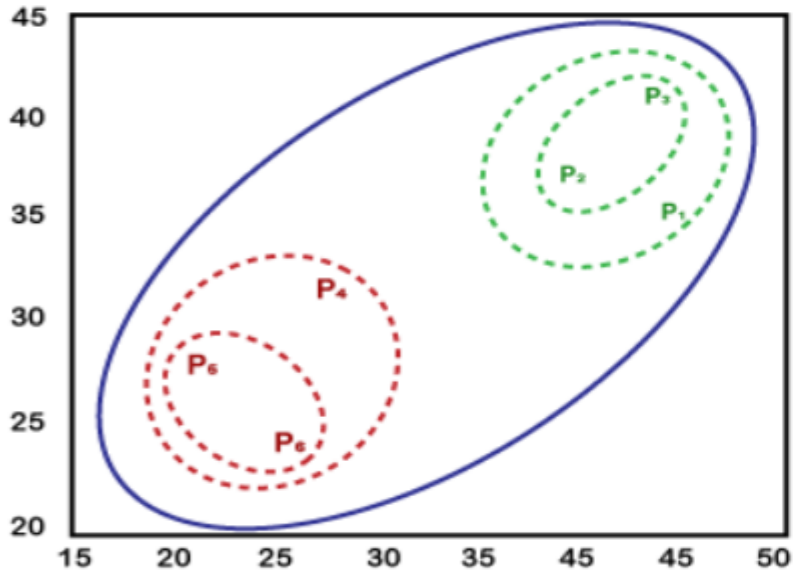


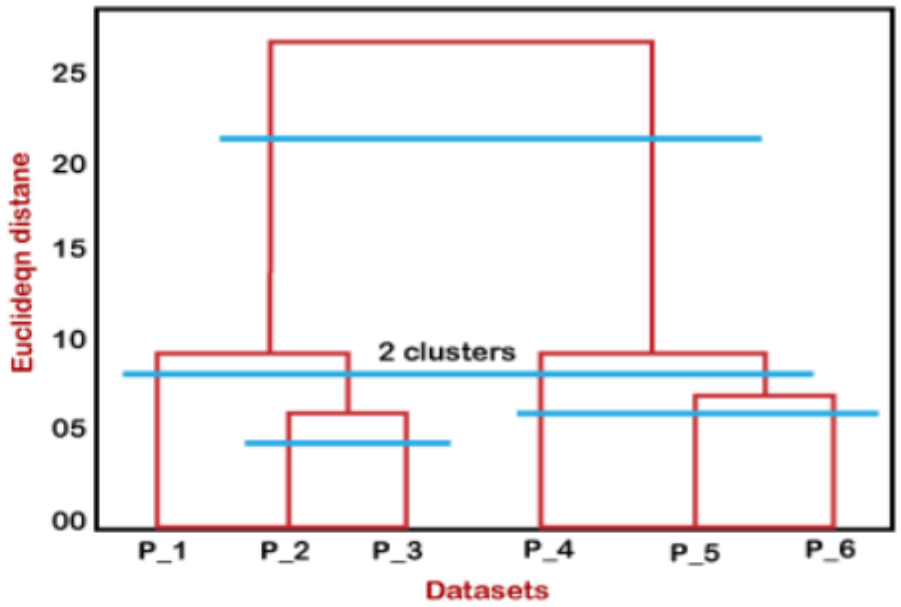
Working of Dendrogram in Hierarchical clustering

- The dendrogram is a tree-like structure that is mainly used to store each step as a memory that the HC algorithm performs.

- In the dendrogram plot, the Y-axis shows the Euclidean distances between the data points, and the x-axis shows all the data points of the given dataset.

- The working of the dendrogram can be explained using the below diagram:





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

•As we have discussed above, firstly, the datapoints P2 and P3 combine together and form a cluster, correspondingly a dendrogram is created, which connects P2 and P3 with a rectangular shape. The height is decided according to the Euclidean distance between the data points.

•In the next step, P5 and P6 form a cluster, and the corresponding dendrogram is created. It is higher than of previous, as the Euclidean distance between P5 and P6 is a little bit greater than the P2 and P3.

•Again, two new dendrograms are created that combine P1, P2, and P3 in one dendrogram, and P4, P5, and P6, in another dendrogram.

•At last, the final dendrogram is created that combines all the data points together.
We can cut the dendrogram tree structure at any level as per our requirement.

Agglomerative Clustering

	P_1	P_2	P_3	P_4	P_5
P_1	0				
P_2	9	0			
P_3	3	7	0		
P_4	6	5	9	0	
P_5	11	10	(2)	8	0

$\Rightarrow$

	P_1	P_2	$[P_3, P_5]$	P_4
P_1	0			
P_2	9	0		
$[P_3, P_5]$	3	7	0	
P_4	6	5	8	0

$\Rightarrow d(P_1, [P_3, P_5])$

$\Rightarrow \min(d(P_1, P_3), d(P_1, P_5))$

$\Rightarrow \min(3, 11) \Rightarrow 3$

$\Rightarrow d(P_2, [P_3, P_5])$

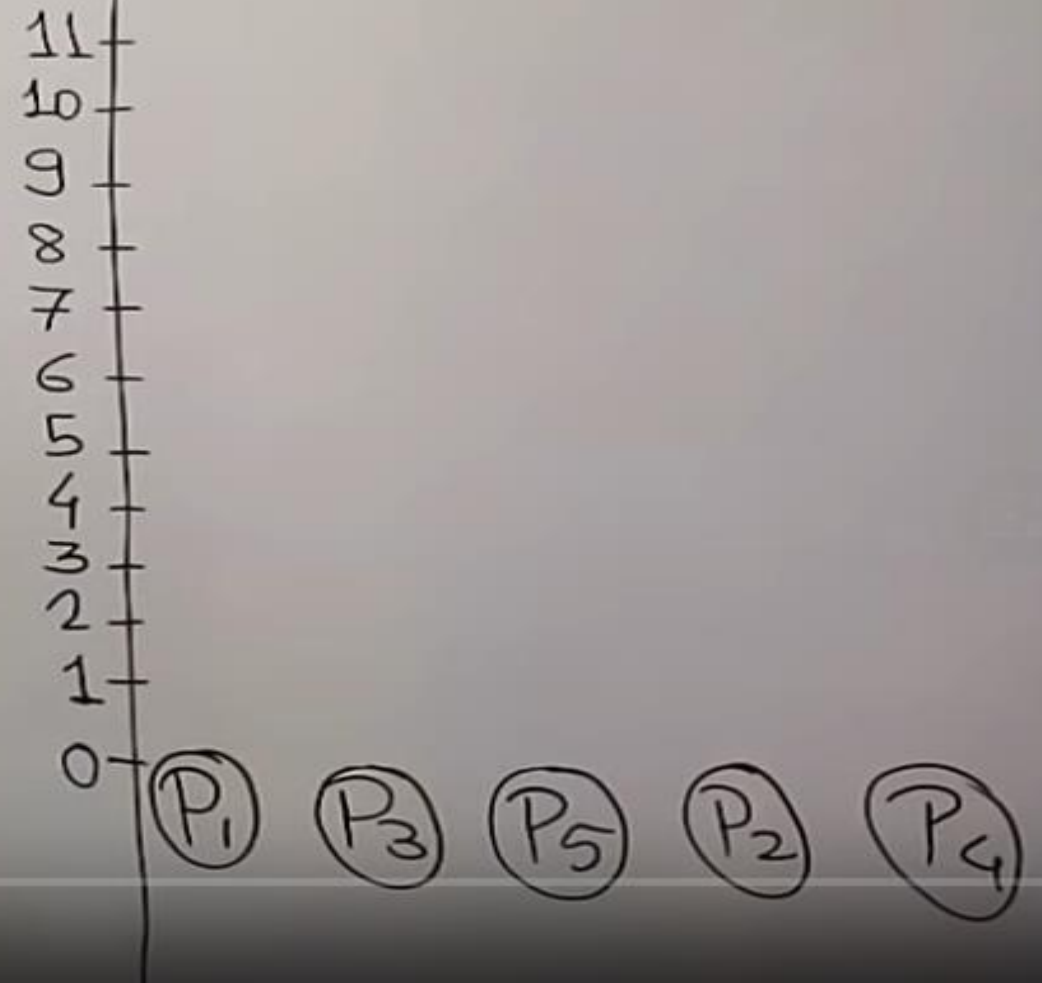
$\Rightarrow \min(d(P_2, P_3), d(P_2, P_5))$

$\Rightarrow \min(7, 10) \Rightarrow 7$

$\Rightarrow d(P_4, [P_3, P_5])$

$\Rightarrow \min(d(P_4, P_3), d(P_4, P_5))$

$\Rightarrow \min(9, 8) \Rightarrow 8$



Agglomerative Clustering

	P_1	P_2	P_3	P_4	P_5
P_1	0				
P_2	9	0			
P_3	3	7	0		
P_4	6	5	9	0	
P_5	11	10	2	8	0

	$[P_1 P_3 P_5]$	P_2	P_4
$[P_1 P_3 P_5]$	0		
P_2	7	0	
P_4	6	5	0

$d(P_2, [P_1 P_3 P_5])$

$\Rightarrow \min(d(P_2,P_1), d(P_2,P_3), d(P_2,P_5))$

$\Rightarrow \min(9, 7, 10) \Rightarrow 7$

	$[P_1 P_3 P_5]$	$[P_2 P_4]$
$[P_1 P_3 P_5]$	0	
$[P_2 P_4]$	6	0

$d([P_1 P_3 P_5], [P_2 P_4])$

$\Rightarrow \min(d(P_2,P_1), d(P_2,P_3), d(P_2,P_5), d(P_4,P_1), d(P_4,P_3), d(P_4,P_5))$

$\Rightarrow \min(9, 7, 10, 6, 9, 8)$

$\Rightarrow 6$

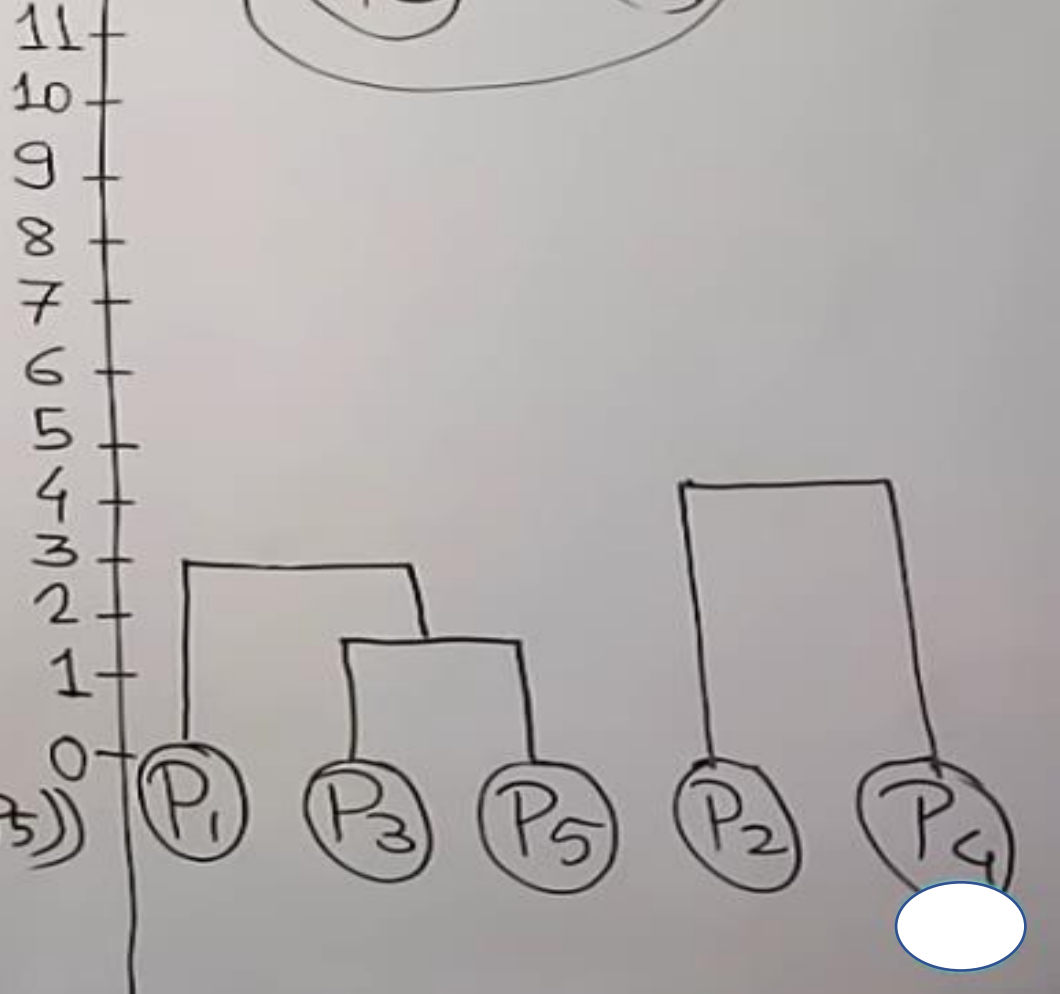
$d(P_4, [P_1 P_3 P_5])$

$\Rightarrow \min(d(P_4,P_1), d(P_4,P_3), d(P_4,P_5))$

$\Rightarrow \min(6, 9, 8)$

$\Rightarrow 6$

P_2 P_4



Agglomerative Clustering

	P_1	P_2	P_3	P_4	P_5
P_1	0				
P_2	9	0			
P_3	3	7	0		
P_4	6	5	9	0	
P_5	11	10	2	8	0

$\Rightarrow$

	P_1	P_2	$[P_3\ P_5]$	P_4
P_1	0			
P_2	9	0		
$[P_3\ P_5]$	11	10	0	
P_4	6	5	9	0

$d(P_2, [P_3\ P_5])$

$\Rightarrow \max(d(P_2, P_3), d(P_2, P_5)) \Rightarrow \max(7, 10) = 10$

$d(P_1, [P_3\ P_5])$

$\Rightarrow \max(d(P_1, P_3), d(P_1, P_5)) \Rightarrow \max(3, 11) = 11$

$d(P_4, [P_3\ P_5])$

$\Rightarrow \max(d(P_4, P_3), d(P_4, P_5)) \Rightarrow \max(9, 8) = 9$

	$[P_1\ P_2\ P_4]$	$[P_3\ P_5]$
$[P_1\ P_2\ P_4]$	0	
$[P_3\ P_5]$	11	0

	P_1	$[P_2\ P_4]$	$[P_3\ P_5]$
P_1			
$[P_2\ P_4]$	9	0	
$[P_3\ P_5]$	11	10	0

P_3 P_5

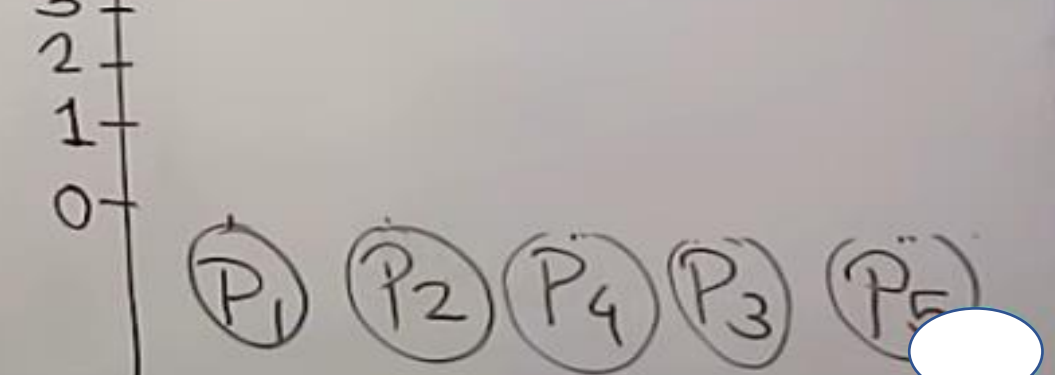
P_1.

P_3

P_2.

P_5

P_4.



DBSCAN Clustering

DBSCAN algorithm

- DBSCAN stands for **d**ensity-**b**ased **s**patial **c**lustering of **a**pplications with **n**oise. It is able to find arbitrary shaped clusters and clusters with noise (i.e. outliers).

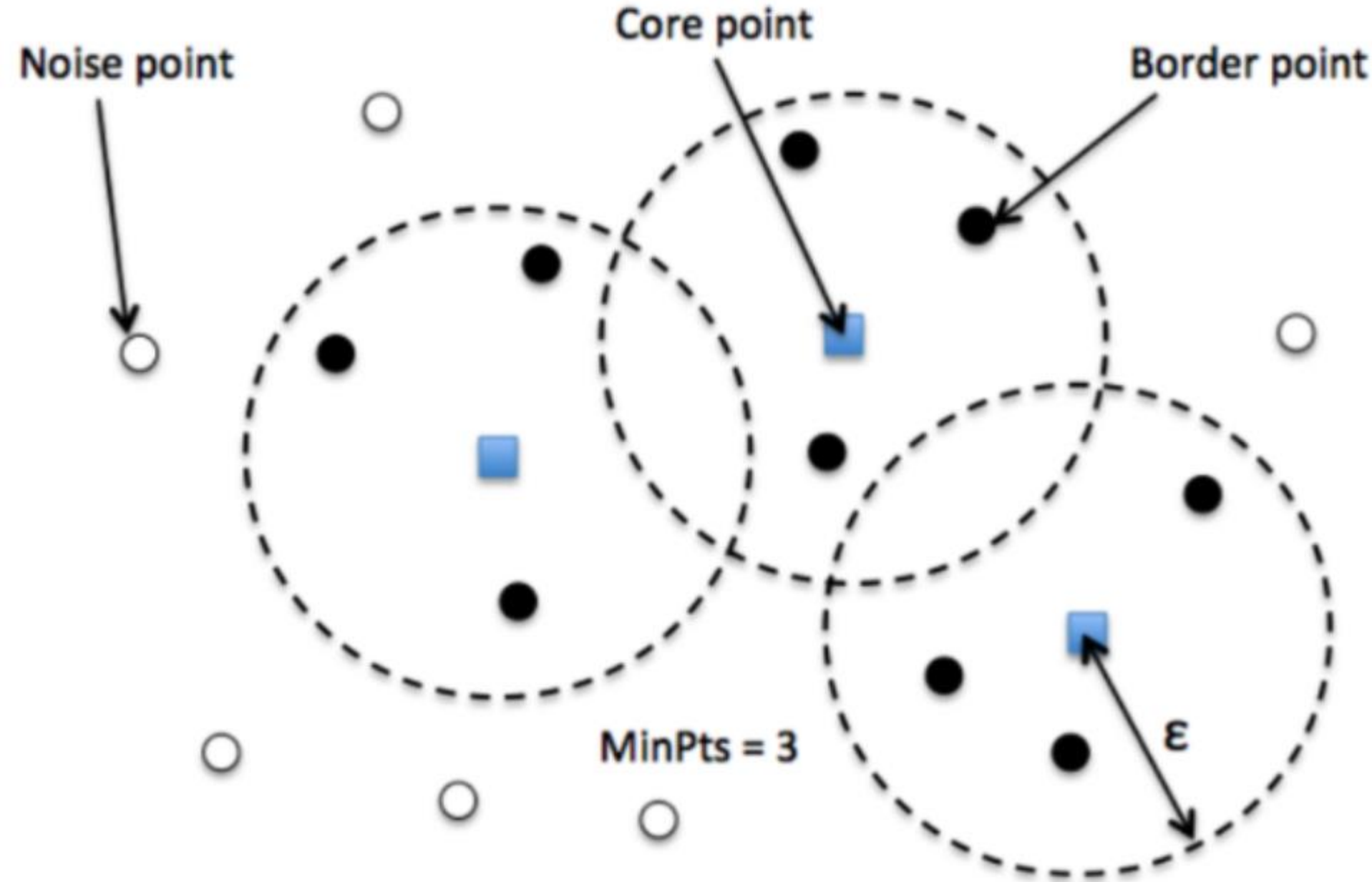
- The main idea behind DBSCAN is that a point belongs to a cluster if it is close to many points from that cluster.

- There are two key parameters of DBSCAN:
  - **eps**: The distance that specifies the neighborhoods. Two points are considered to be neighbors if the distance between them are less than or equal to **eps**.

    - If the eps value is chosen too small then large part of the data will be considered as outliers. If it is chosen very large then the clusters will merge and majority of the data points will be in the same clusters. Therefore, suitable value of **eps** needs to find out.

  - **minPts:** Minimum number of data points to define a cluster.

- Based on these two parameters, points are classified as core point, border point, or outlier:
  - **Core point:** A point is a core point if there are at least minPts number of points (including the point itself) in its surrounding area with radius eps.
  - **Border point:** A point is a border point if it is reachable from a core point and there are less than minPts number of points within its surrounding area.
  - **Outlier:** A point is an outlier if it is not a core point and not reachable from any core points.


Noise point
Core point
Border point
MinPts = 3
ε

Algorithmic steps for DBSCAN clustering

1. Find all the neighbor points within eps and identify the core points or visited with more than MinPts neighbors.

2. For each core point if it is not already assigned to a cluster, create a new cluster.

3. Find recursively all its density connected points and assign them to the same cluster as the core point.

   A point *a* and *b* are said to be density connected if there exist a point *c* which has a sufficient number of points in its neighbors and both the points *a* and *b* are within the *eps distance*. This is a chaining process. So, if *b* is neighbor of *c*, *c* is neighbor of *d*, *d* is neighbor of *e*, which in turn is neighbor of *a* implies that *b* is neighbor of *a*.

4. Iterate through the remaining unvisited points in the dataset. Those points that do not belong to any cluster are noise.