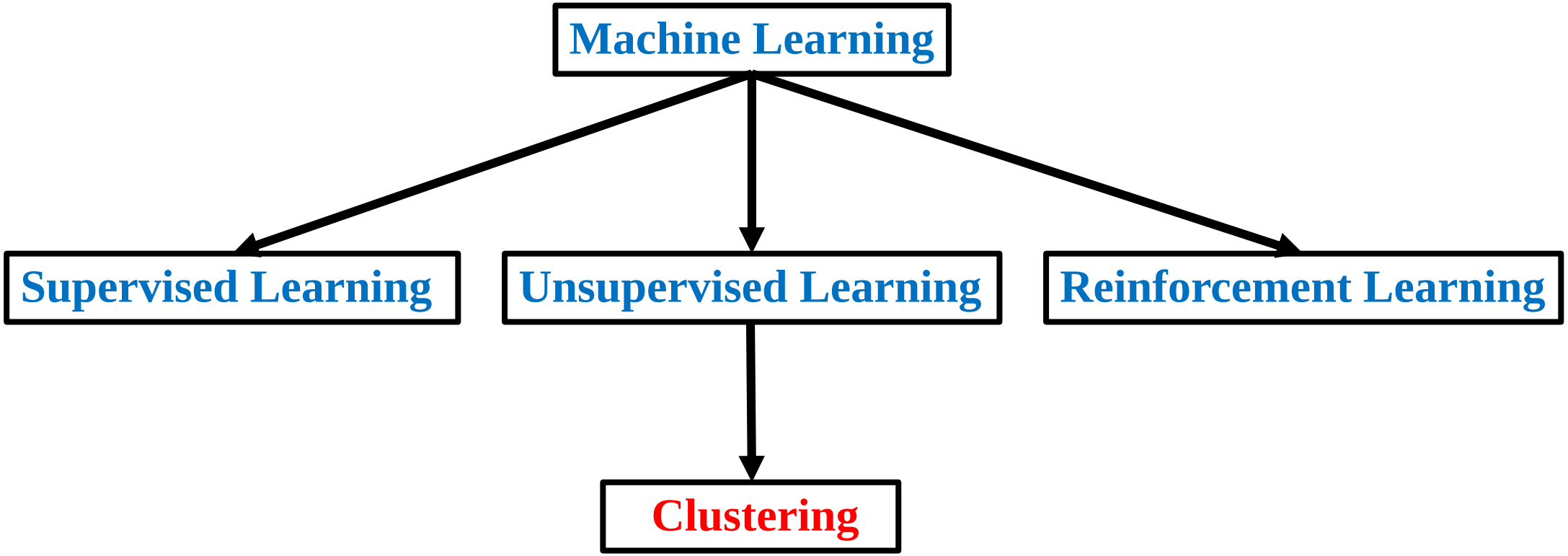
Unit V Syllabus

- **Clustering Algorithms:** K-Means, K-Medoids.
- **Hierarchical and Density – Based Clustering**.
- **Spectral Clustering.**
- **Outlier Analysis-** Introduction to isolation factor and local outlier factor.
- **Evaluation Metrics and Score:** Elbow method, extrinsic and intrinsic methods.

Types in Machine Learning



Types of Problems in Machine Learning

- **Regression**

  Output is a continuous quantity. To predict the weight of a person using height. e.g. Linear Regression.

- **Classification**

Output is a categorical value. To classify emails into two classes, spam and non-spam. e.g. classification algorithms such as Support Vector Machines, Naive Bayes, Logistic Regression, K Nearest Neighbor.

- **Clustering**

Problem involves assigning the input into two or more clusters based on feature similarity. Similar groups based on their interests, age, geography, etc can be done by using Unsupervised Learning algorithms like K-Means Clustering.

Clustering

- A way of grouping the data points into different clusters, consisting of similar data points.

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

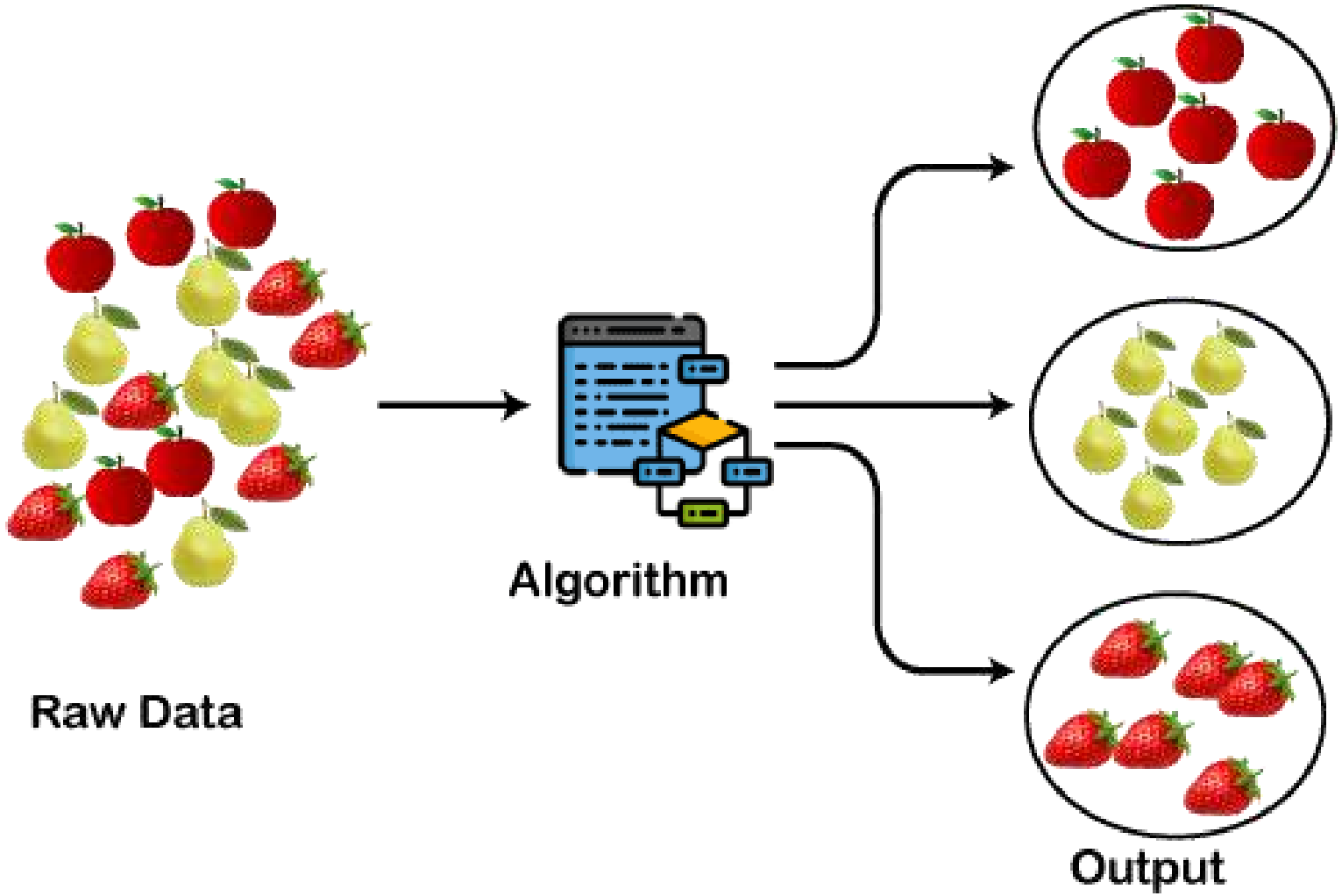
- It is an **unsupervised learning method**, hence no supervision is provided to the algorithm, and it deals with the **unlabeled dataset**.

- 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**.

Clustering

- The clustering technique can be widely used in various tasks. Some most common uses of this technique are:

1. Market Segmentation
2. Statistical data analysis
3. Social network analysis
4. Image segmentation
5. Anomaly detection, etc.



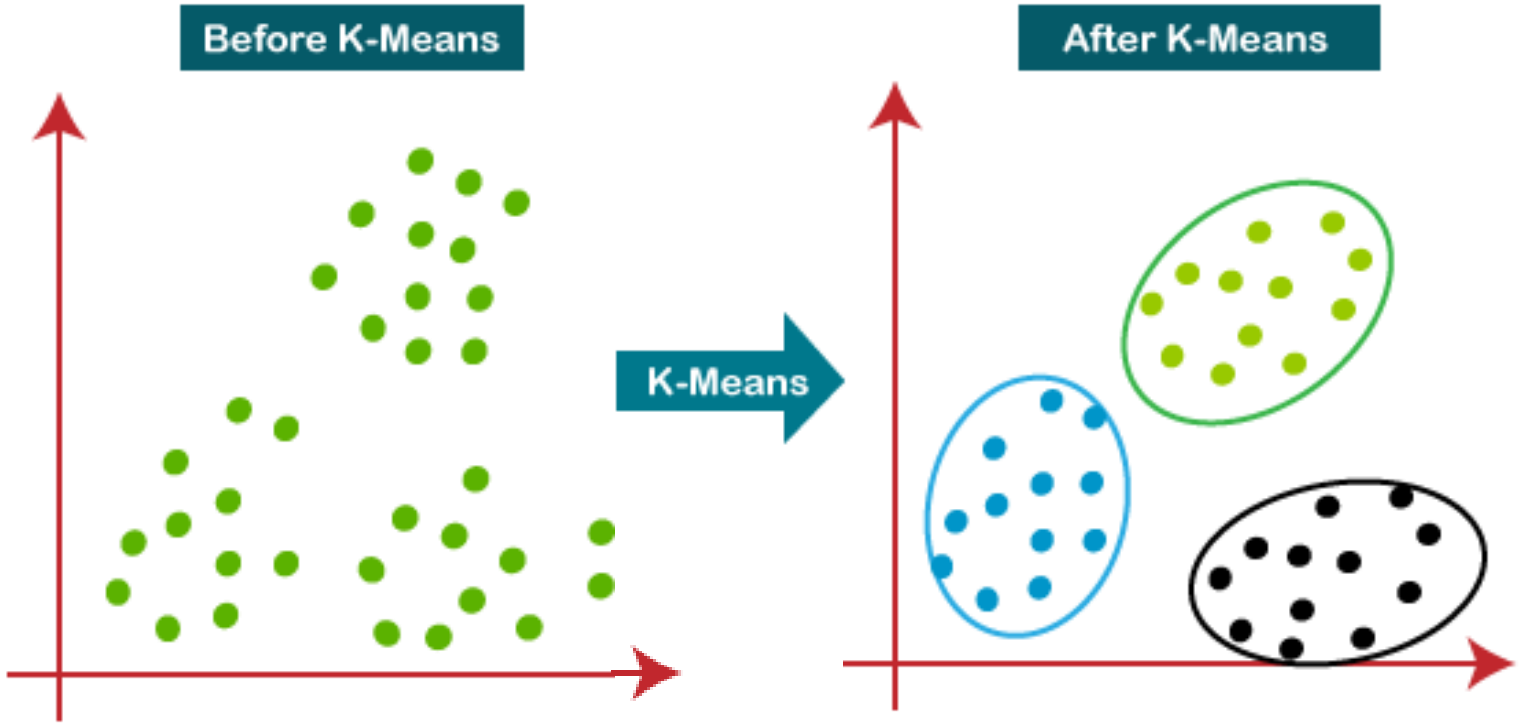
Types of Clustering

- The clustering technique can be divided into following types:

1. Partitioning Clustering (Centroid-based Clustering)
2. Density-based Clustering
3. Hierarchical Clustering
4. Distributed Model-based Clustering
5. Fuzzy Clustering

K-means Clustering

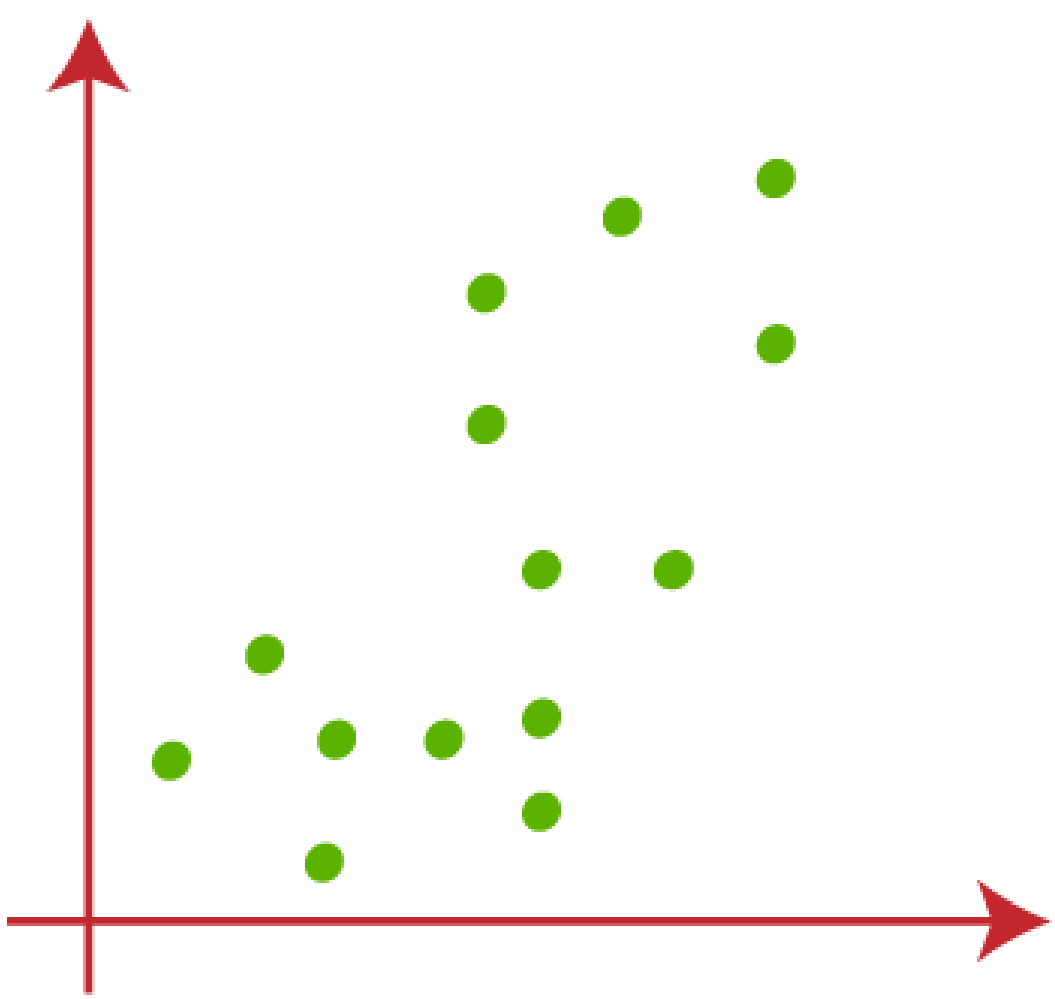
- **K-Means Clustering** is an **Unsupervised Learning Algorithm**, which groups the **unlabeled dataset** into different **clusters**.
- Here **K defines the number of pre-defined clusters** that need to be created in the process, as if **K=2**, there will be **two clusters**, and for **K=3**, there will be **three clusters**, and so on.
- It is a **centroid-based algorithm**, where each cluster is associated with a centroid. The main aim of this algorithm is to **minimize the sum of distances between the data point and their corresponding clusters.**

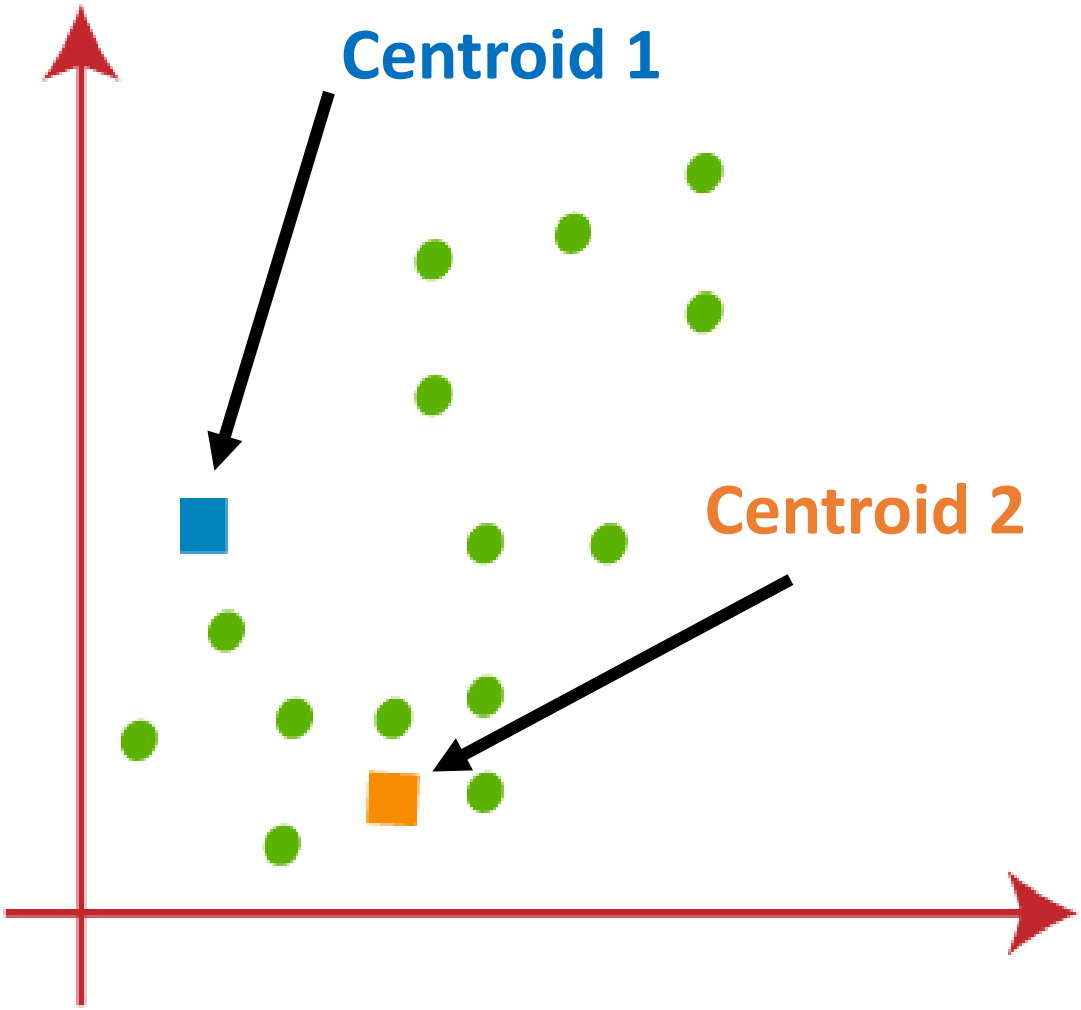


K-means Clustering

- **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 data point 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 Clustering

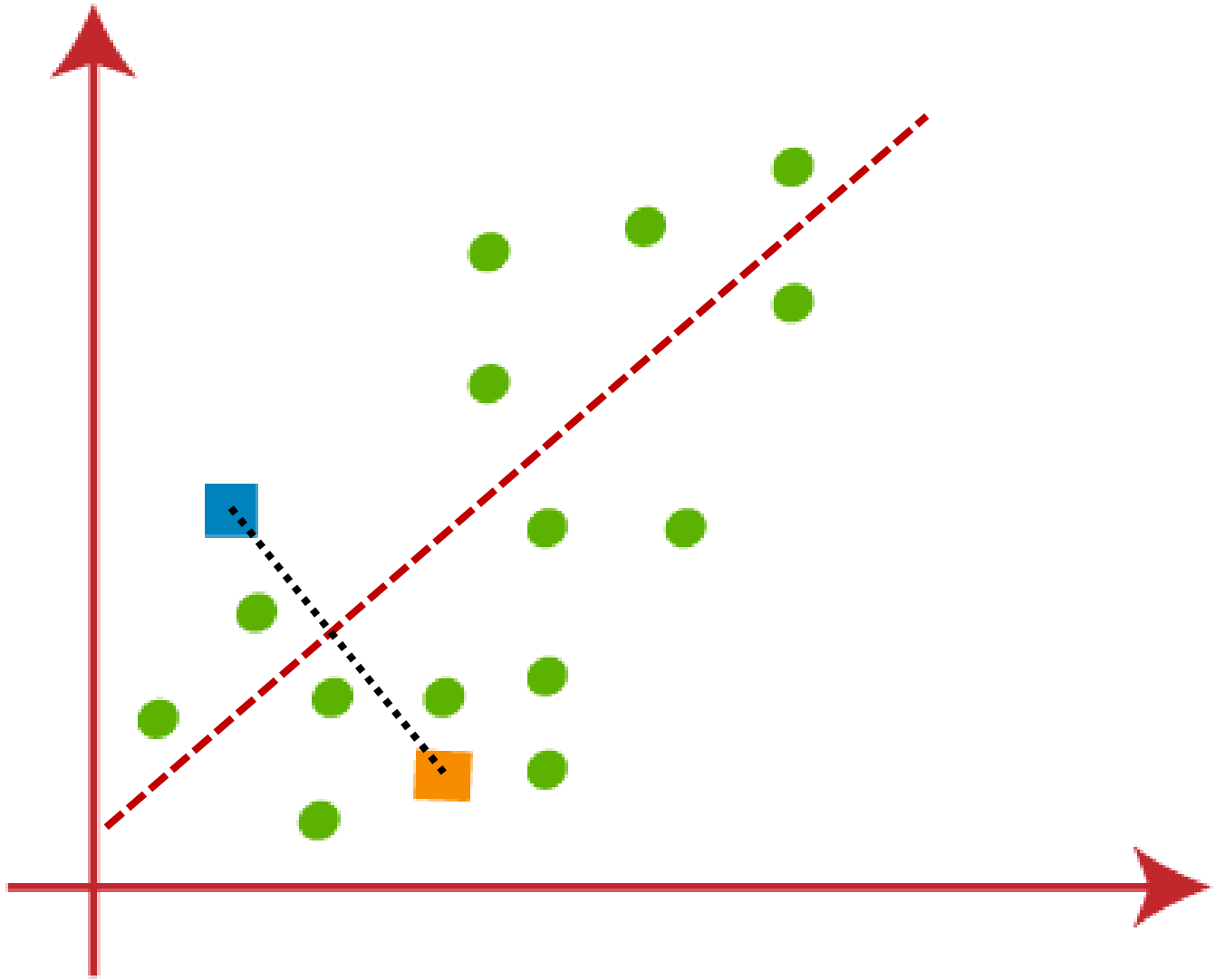




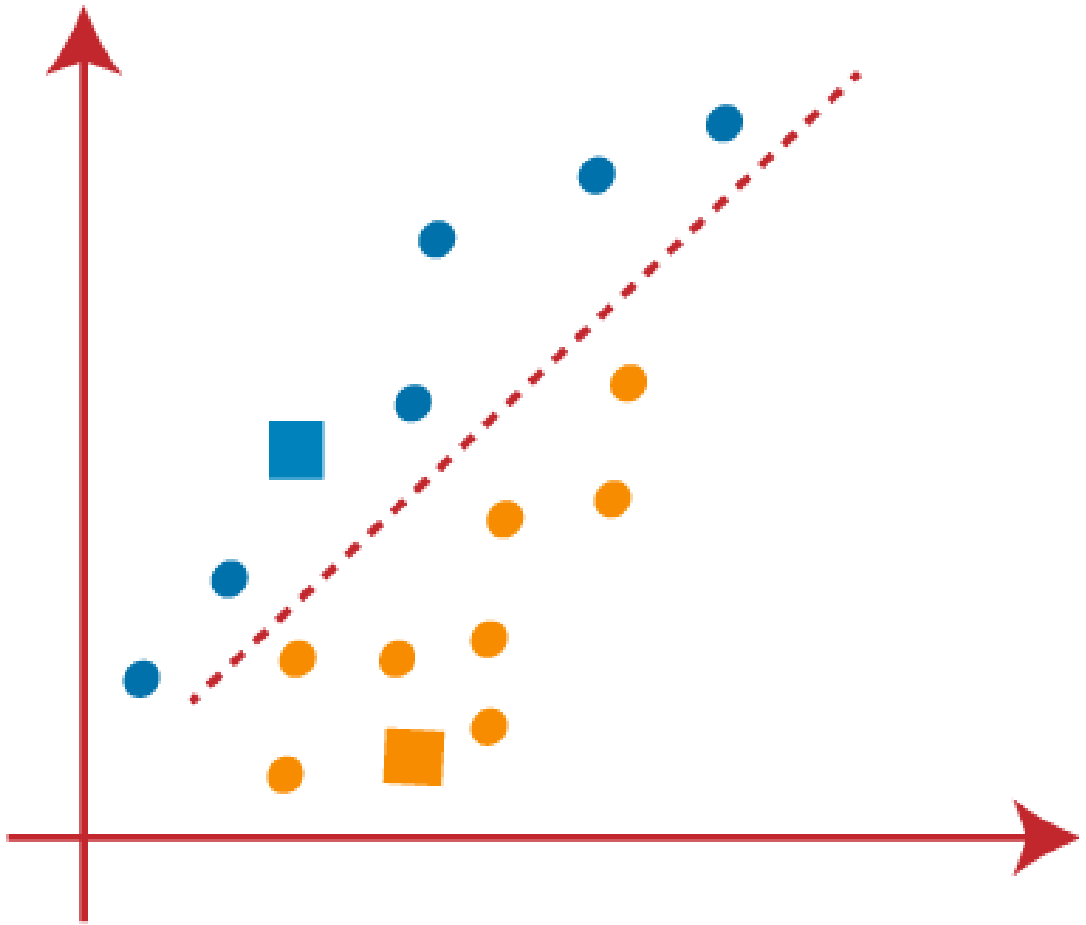
K-means Clustering

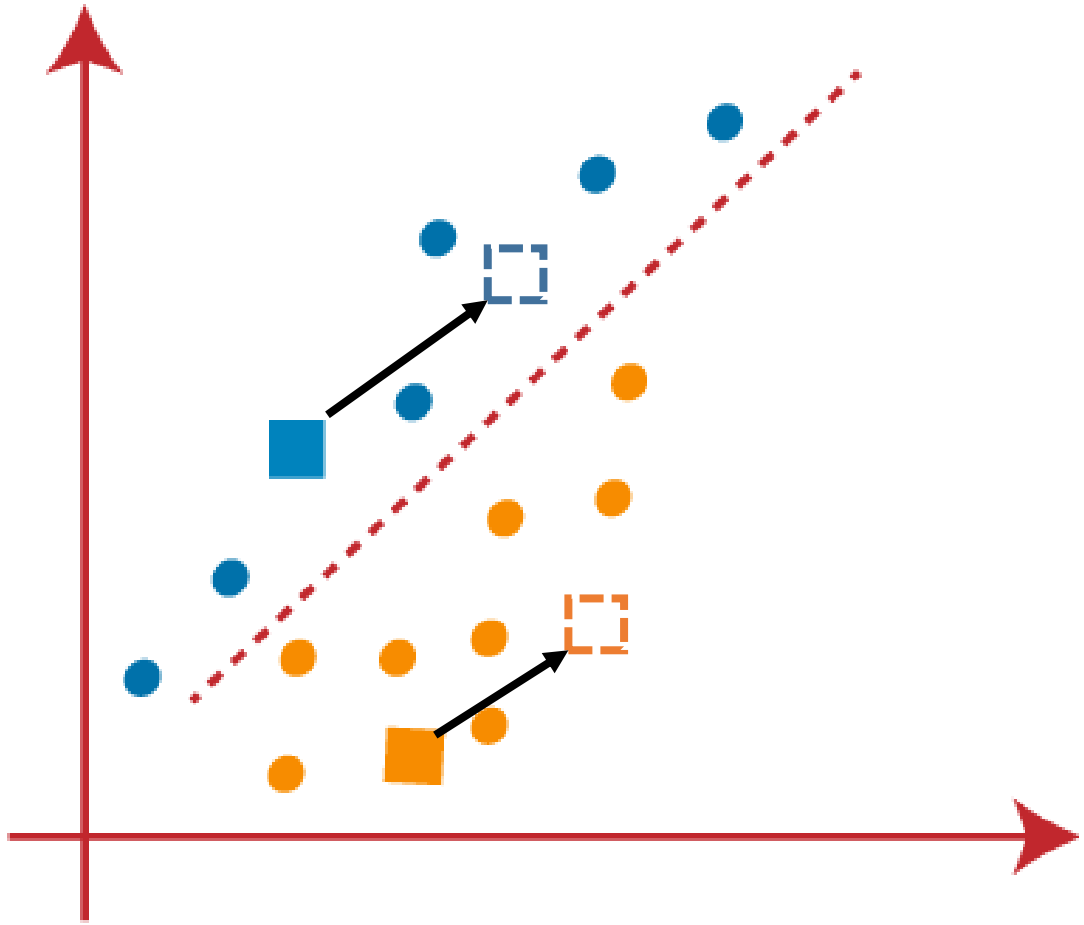
Calculate Distance using Euclidian Distance of Each Data Point from Centroid

Draw median between two centroids

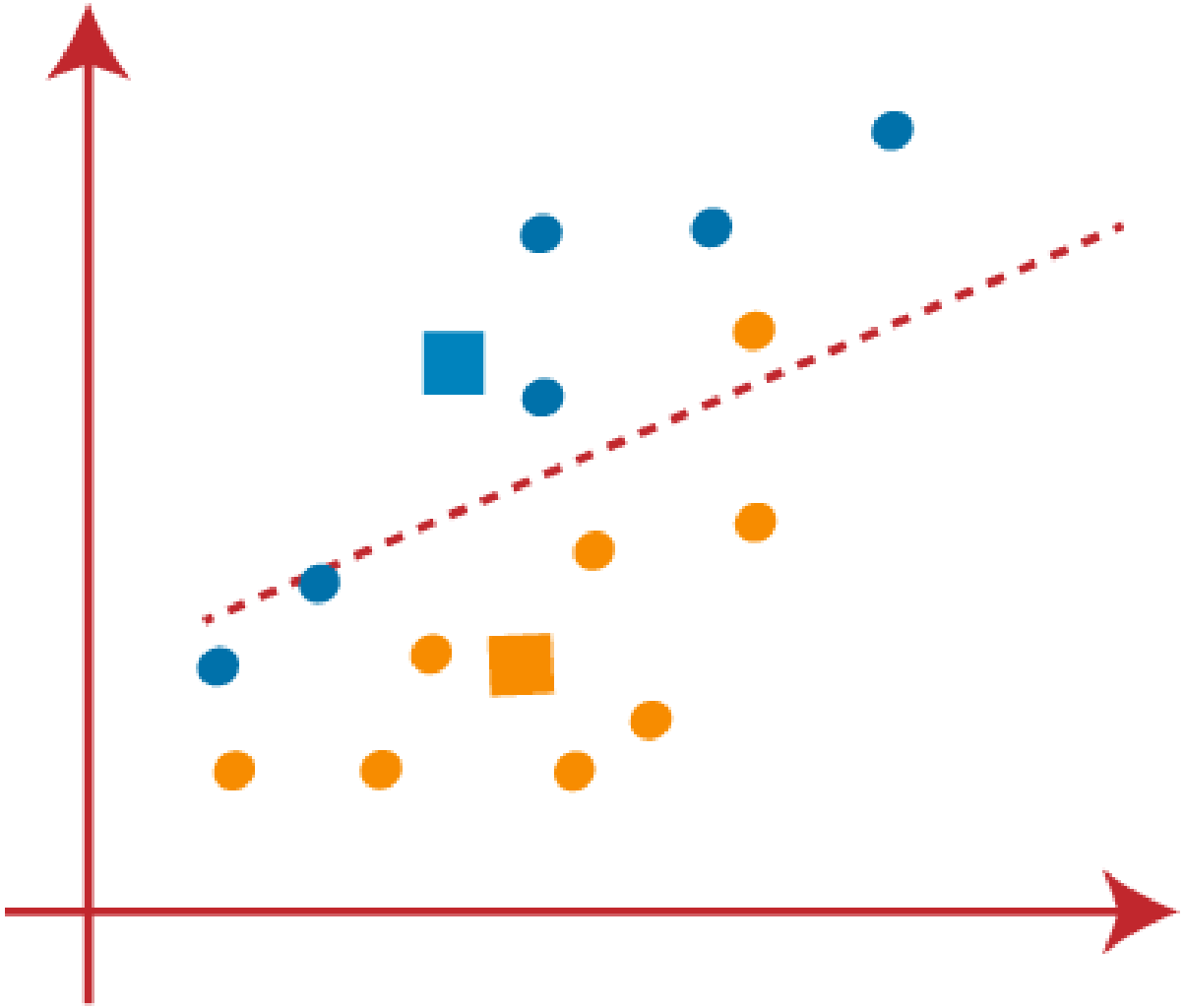


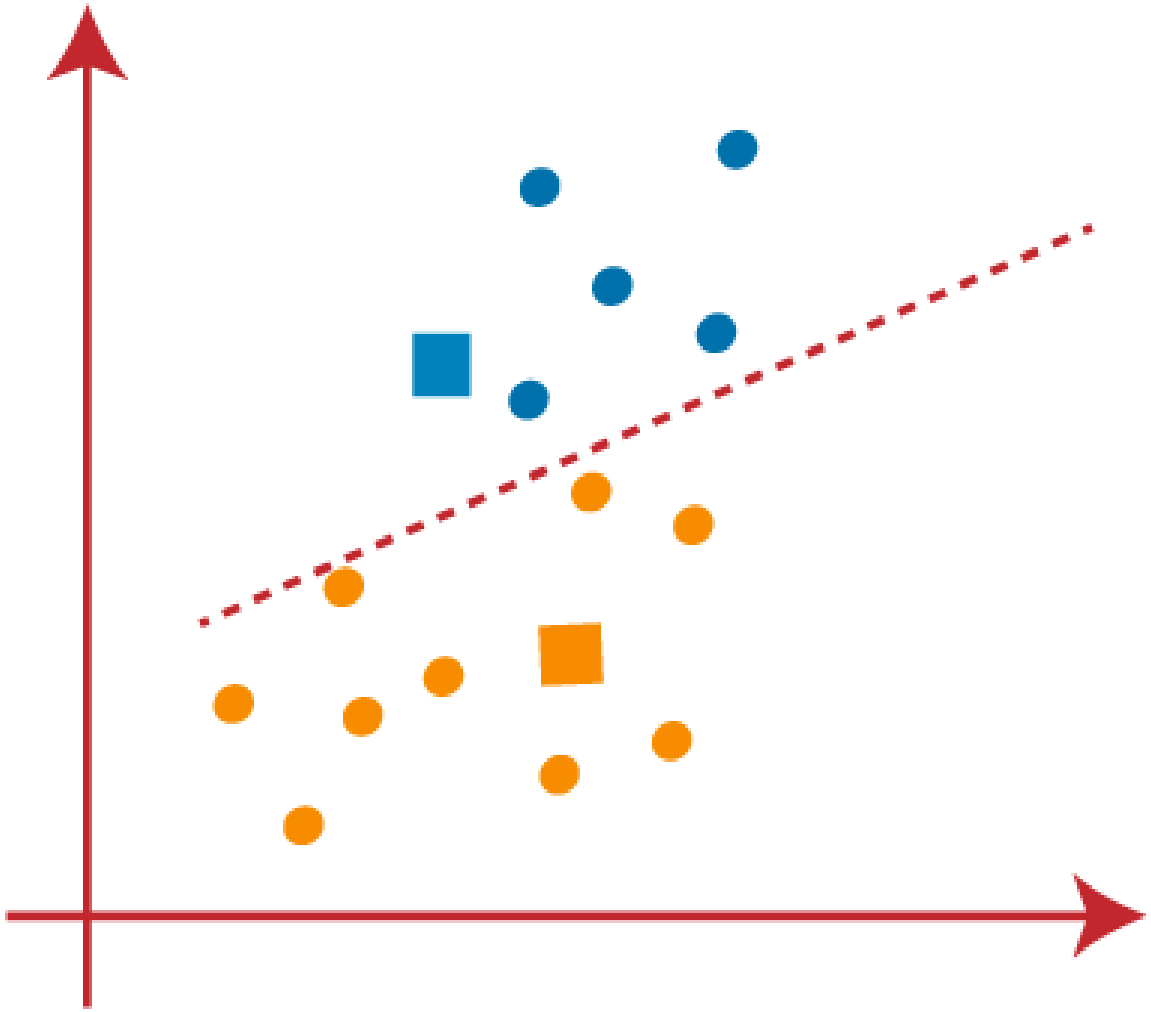
K-means Clustering



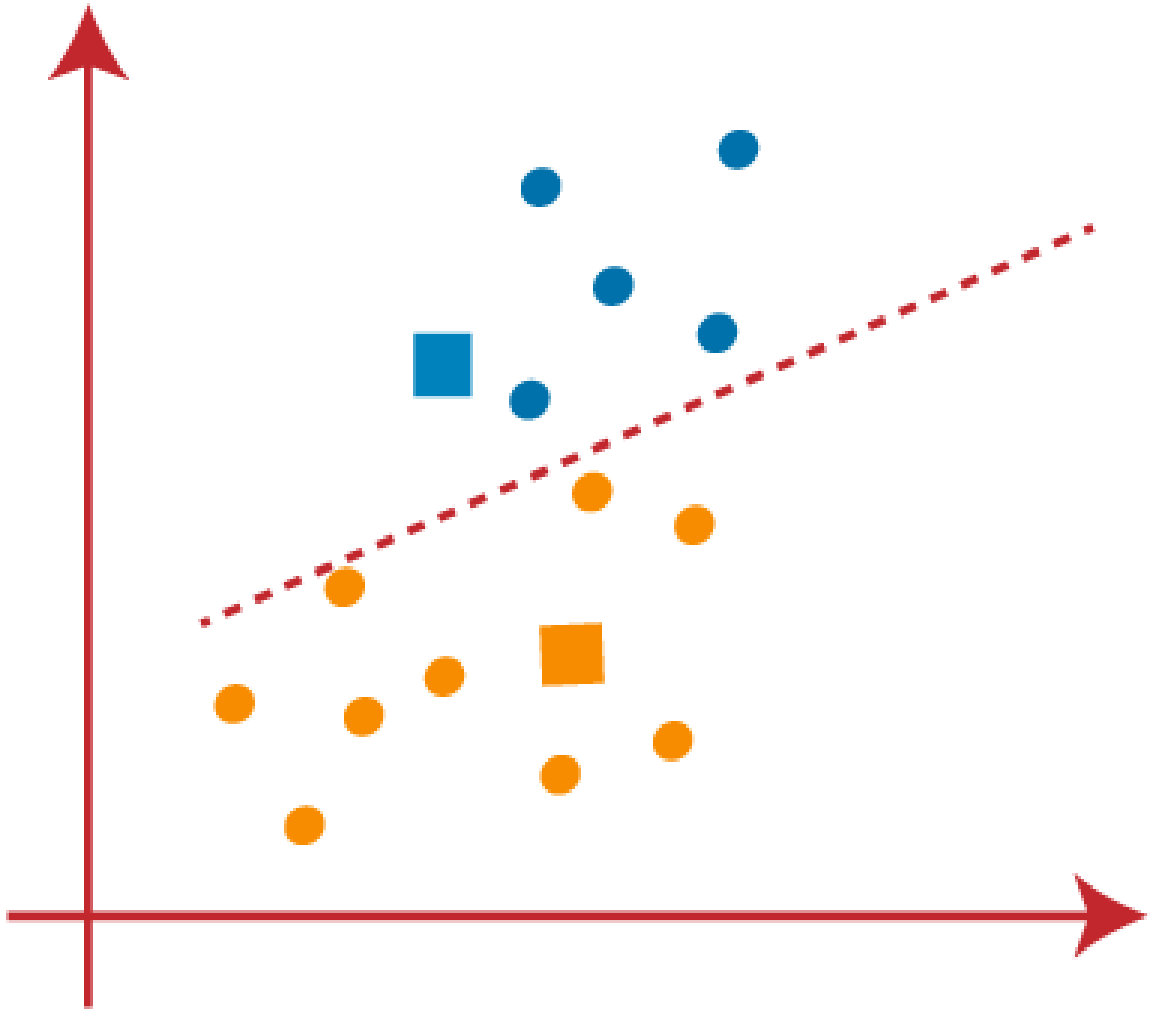


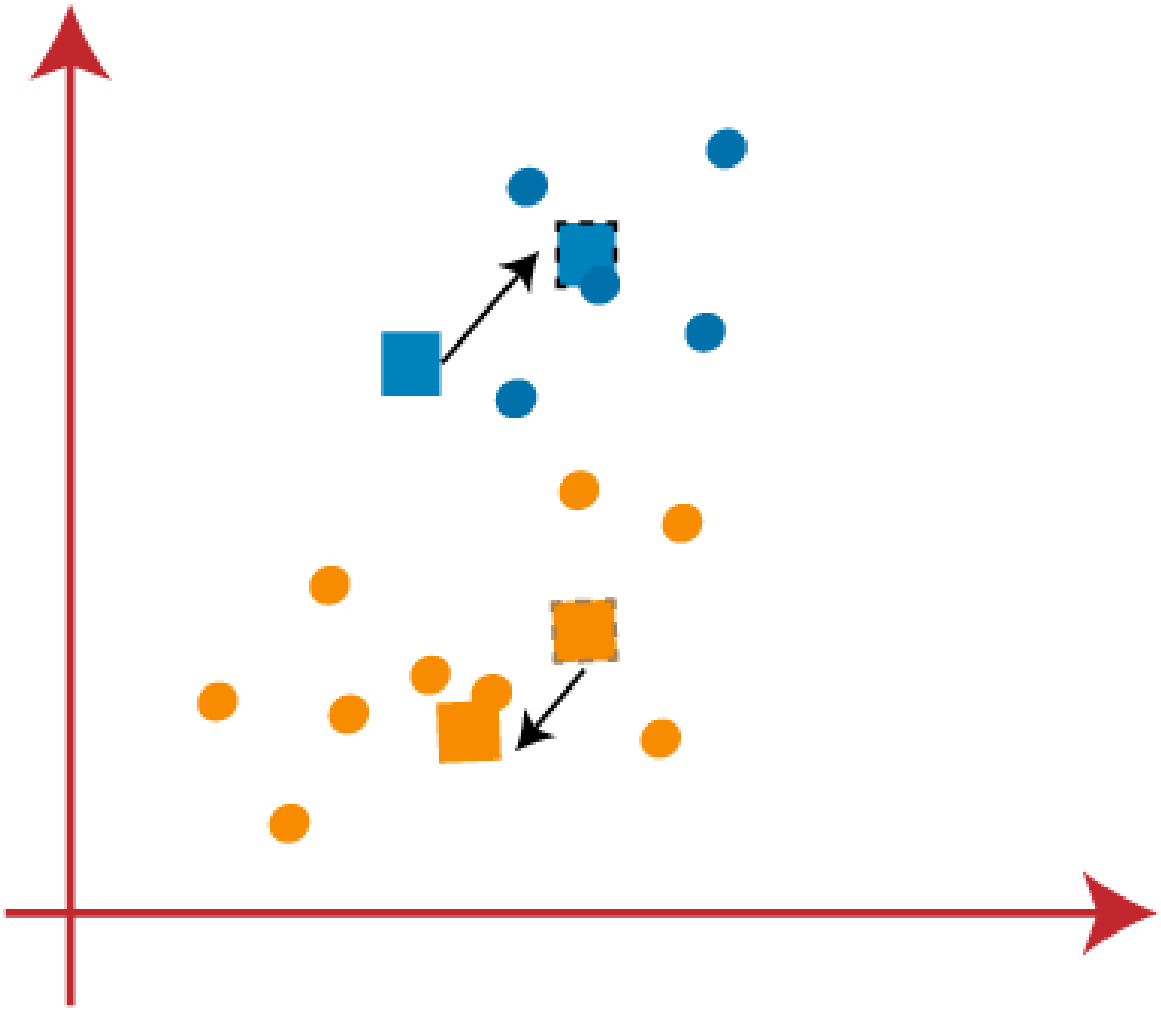
K-means Clustering



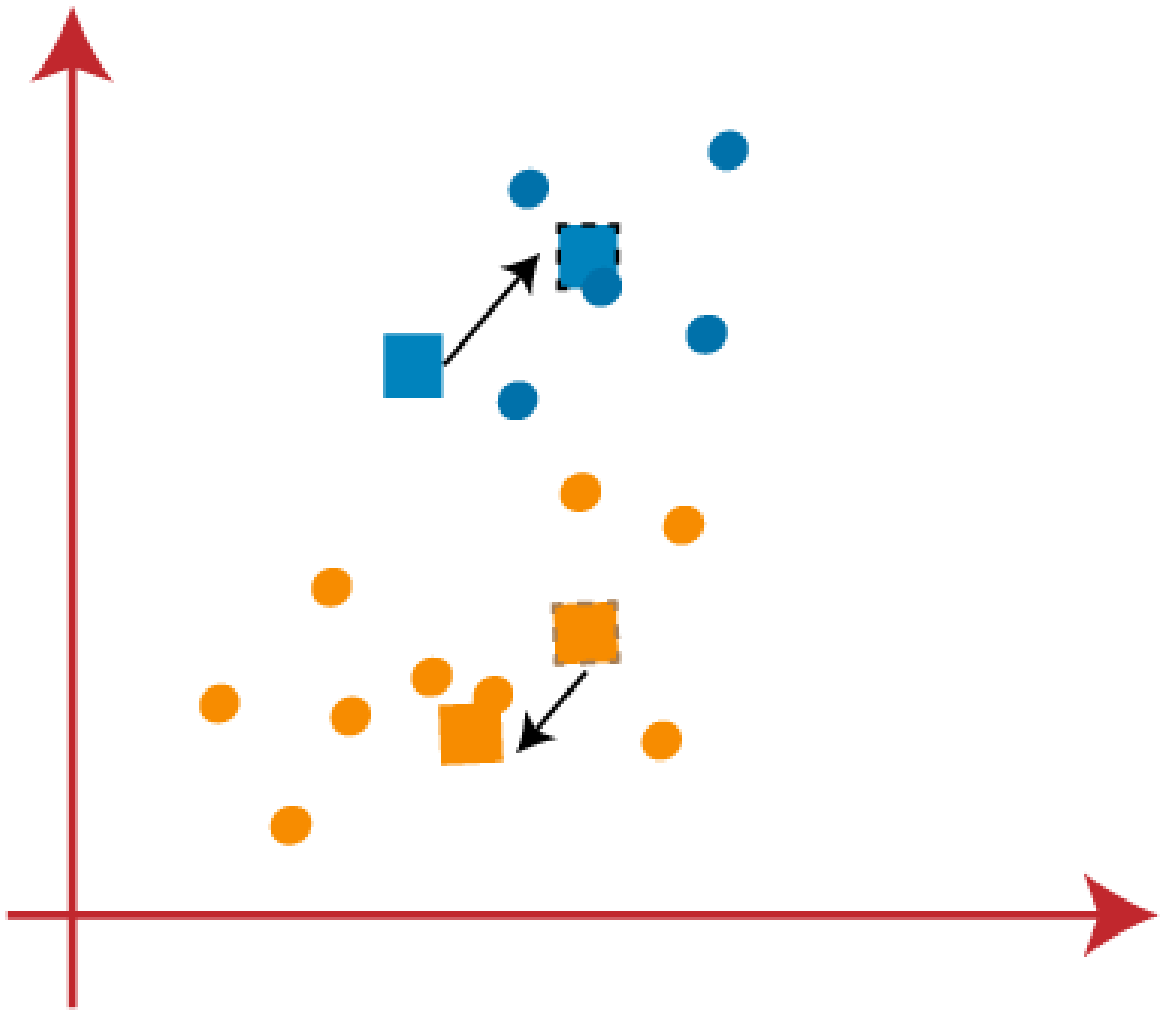


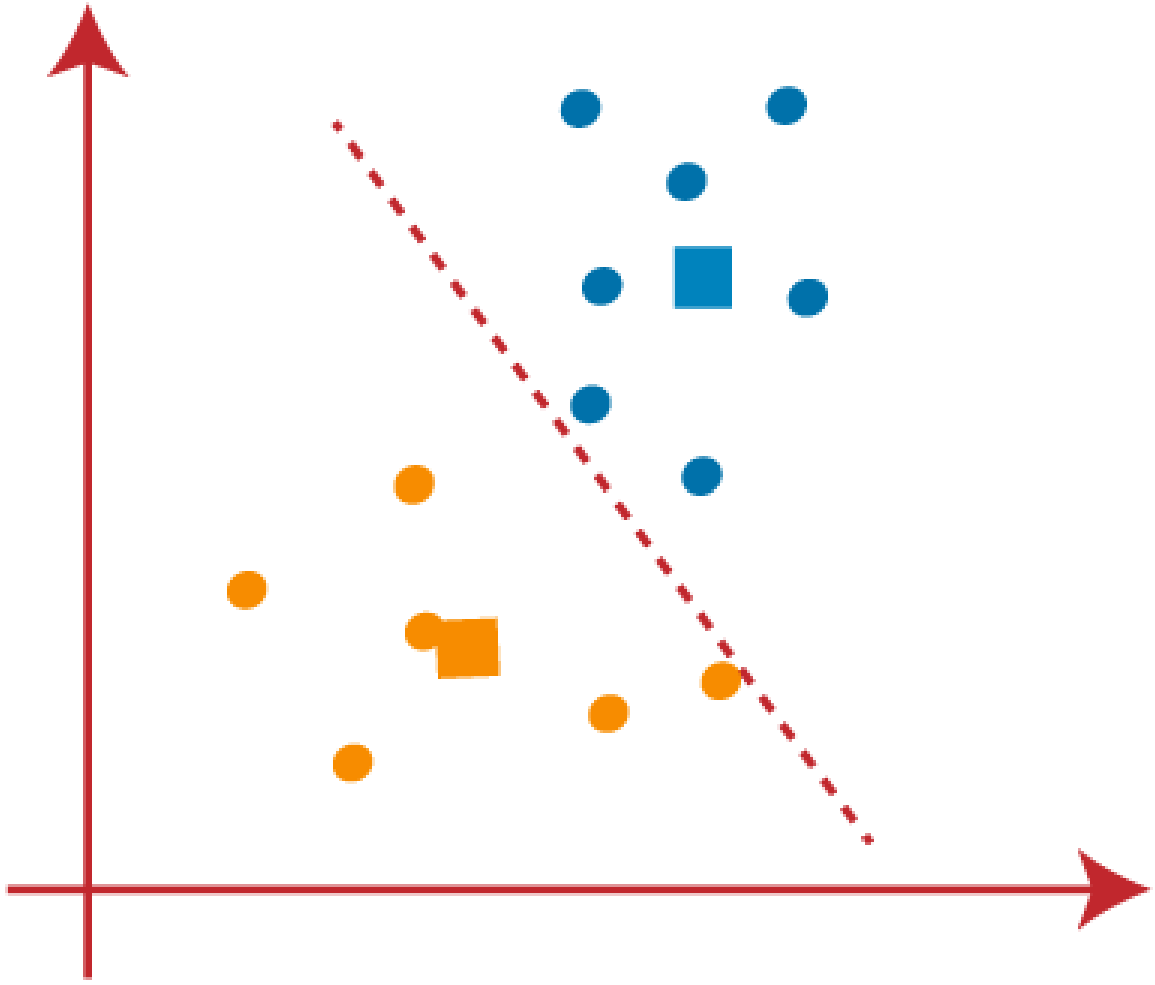
K-means Clustering



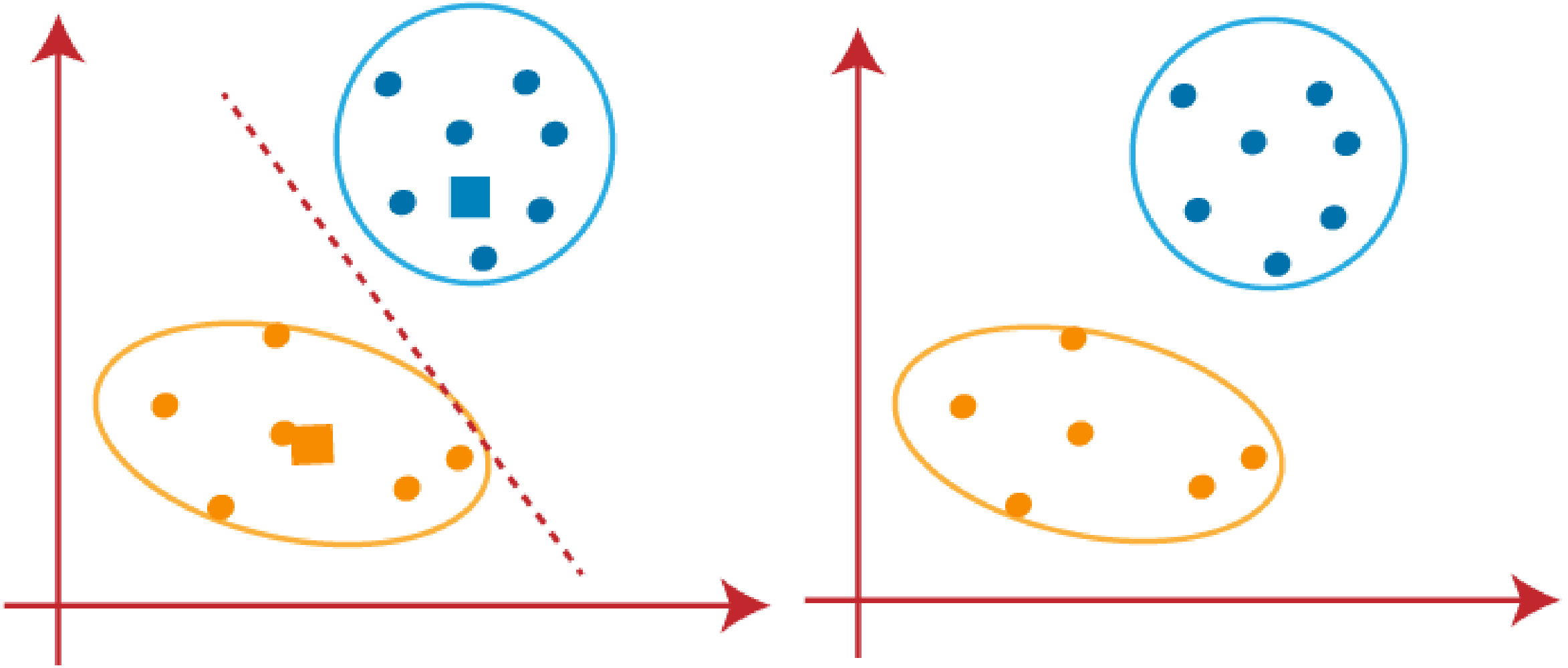


K-means Clustering





K-means Clustering



K-medoids Clustering

1. **K-Medoids** (also called **Partitioning Around Medoid**) algorithm was proposed in 1987 by Kaufman and Rousseeuw.
2. A medoid can be defined as the point in the cluster, whose dissimilarities with all the other points in the cluster are minimum.
3. The dissimilarity of the medoid (Ci) and object (Pi) is calculated by using E = |Pi – Ci|

$$\text{Cost of K} - \text{medoids is given by}$$

$$c = \sum\sum |P_i - C_i|$$

K-medoids Clustering

Build phase:

1. Select k objects to become the medoids, or in case these objects were provided use them as the medoids;
2. Calculate the dissimilarity matrix if it was not provided;
3. Assign every object to its closest medoid;

Swap phase:

4. For each cluster search if any of the object of the cluster decreases the average dissimilarity coefficient; if it does, select the entity that decreases this coefficient the most as the medoid for this cluster;
5. If at least one medoid has changed go to (3), else end the algorithm.

K-Medoids Clustering

Samples	X	Y
1	2	6
2	3	4
3	3	8
4	4	7
5	6	2
6	6	4
7	7	3
8	7	4
9	8	5
10	7	6

Apply K-Medoid Algorithm to form two Clusters

Use Manhattan Distance to find distance between data point and medoid.

K-Medoids Clustering

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

Select Two Medoids

C1= (3,4) and C2= (7,4)

Calculate C1 and C2
Use Manhattan Distance
|x1-x2| + |y1-y2|

Mdist [(2,6) , (3,4)] =|2 - 3| + |6-4|
Mdist [(2,6) , (3,4)] = 3

Clusters are
C1 = {(2,6), (3,4), (3,8), (4,7)}

C2 = {(6,2), (6,4), (7,3), (7,4), (8,5), (7,6)}

K-Medoids Clustering

Clusters are

C1 = {(2,6), (3,4), (3,8), (4,7)}

C2 = {(6,2), (6,4), (7,3), (7,4), (8,5), (7,6)}

$$Cost(C, X) = \sum |C_i - X_i|$$

Total Cost = { Cost((3,4),(2,6)) + Cost((3,4),(3,8)) + Cost((3,4),(4,7)) +

Cost((7,4),(6,2)) + Cost((7,4),(6,4)) + Cost((7,4),(8,5)) + Cost((7,4),(7,6))}

Total Cost = 3 + 4 + 4 + 2 + 3 + 1 + 1 + 2 = 20

K-Medoids Clustering

Samples	X	Y	C1	C3	Cluster
1	2	6	3	8	C1
2	3	4	0	5	C1
3	3	8	4	9	C1
4	4	7	4	7	C1
5	6	2	5	2	C3
6	6	4	3	2	C3
7	7	3	5	0	C3
8	7	4	4	1	C3
9	8	5	6	3	C3
10	7	6	6	3	C3

Randomly Select One non-medoid And recalculate the cost

C1= (3,4) and C3= (7,3)

Calculate C1 and C3
Use Manhattan Distance
|x1-x2| + |y1-y2|

Mdist [(2,6) , (7,3)] =|2 - 7| + |6-3|
Mdist [(2,6) , (7,3)] = 8

Clusters are
C1 = {(2,6), (3,4), (3,8), (4,7)}

C2 = {(6,2), (6,4), (7,3), (7,4), (8,5), (7,6)}

K-Medoids Clustering

Clusters are

C1 = {(2,6), (3,4), (3,8), (4,7)}

C2 = {(6,2), (6,4), (7,3), (7,4), (8,5), (7,6)}

$$Cost(C, X) = \sum |C_i - X_i|$$

Total Cost = { Cost((3,4),(2,6)) + Cost((3,4),(3,8)) + Cost((3,4),(4,7)) +

Cost((7,3),(6,2)) + Cost((7,3),(6,4)) + Cost((7,3),(8,5)) + Cost((7,3),(7,6))}

Total Cost = 3 + 4 + 4 + 2 + 2 + 1 + 3 + 3 = 22

K-Medoids Clustering

Cost of Swapping of Medoid from C2 with C3

S = Current Total Cost – Previous Total Cost

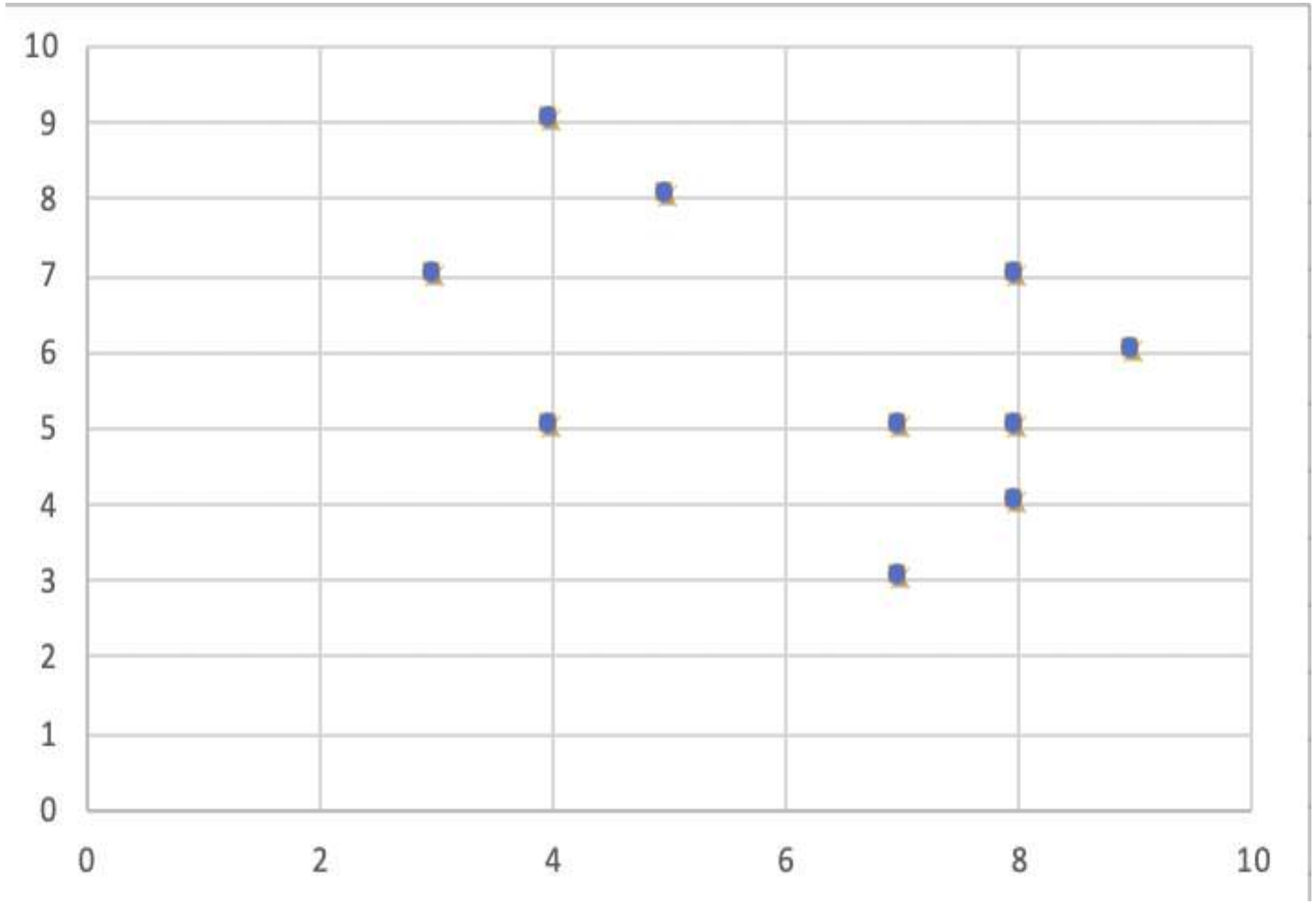
S = 22 – 20 = 2 > 0

Hence Swapping C2 with C3 is not a good idea.

So, Medoids are C1 = (3, 4) and C2 = (7, 4)

K-medoids Clustering

	X	Y
0	8	7
1	3	7
2	4	9
3	9	6
4	8	5
5	5	8
6	7	3
7	8	4
8	7	5
9	4	5



K-medoids Clustering

	X	Y	Dissimilarity from C1	Dissimilarity from C2
0	8	7	6	2
1	3	7	3	7
2	4	9	4	8
3	9	6	6	2
4	**8**	**5**	-	-
5	5	8	4	6
6	7	3	5	3
7	8	4	5	1
8	7	5	3	1
9	**4**	**5**	-	-

The points 1, 2, 5 go to cluster C1 and 0, 3, 6, 7, 8 go to cluster C2.

The Cost = (3 + 4 + 4) + (3 + 1 + 1 + 2 + 2) = 20

K-medoids Clustering

	X	Y	Dissimilarity from C1	Dissimilarity from C2
0	8	7	6	3
1	3	7	3	8
2	4	9	4	9
3	9	6	6	3
4	8	5	4	1
5	5	8	4	7
6	7	3	5	2
7	**8**	**4**	-	-
8	7	5	3	2
9	**4**	**5**	-	-

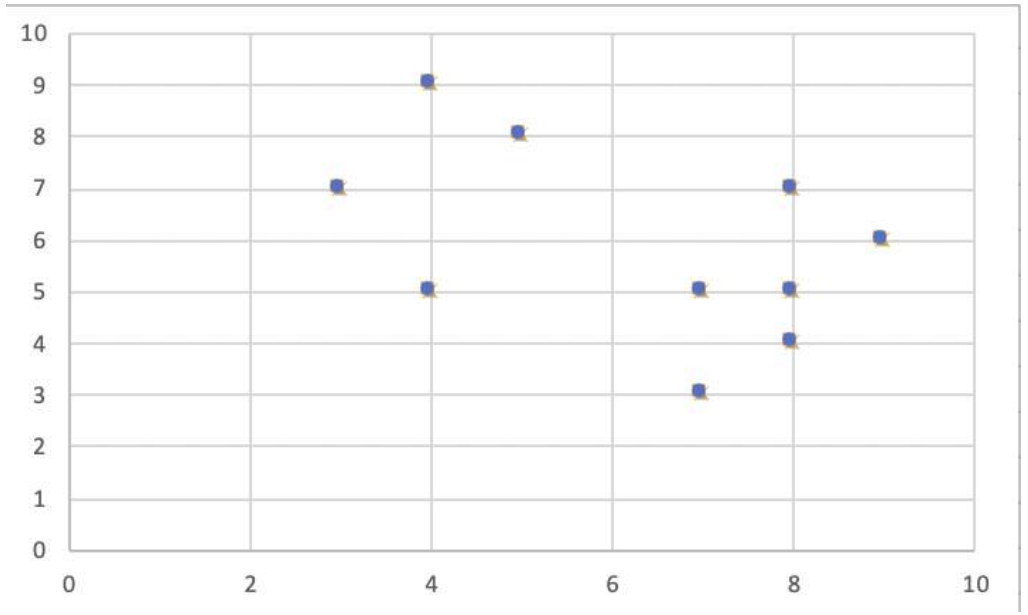
The points 1, 2, 5 go to cluster C1 and 0, 3, 6, 7, 8 go to cluster C2.

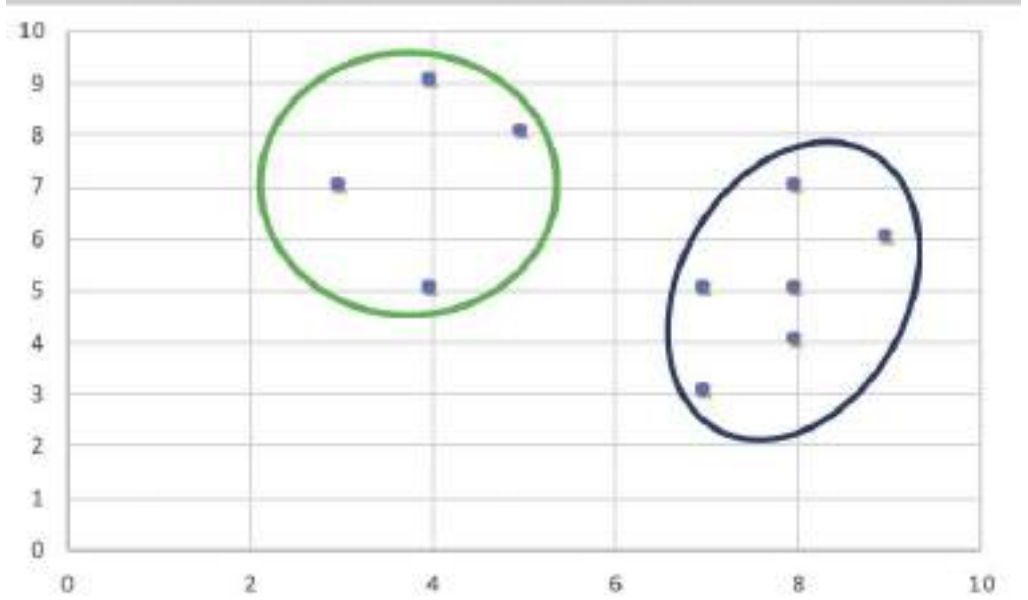
The New cost = (3 + 4 + 4) + (2 + 2 + 1 + 3 + 3) = 22

Swap Cost = New Cost – Previous Cost = 22 – 20 and **2 >0**

K-medoids Clustering

	X	Y
0	8	7
1	3	7
2	4	9
3	9	6
4	8	5
5	5	8
6	7	3
7	8	4
8	7	5
9	4	5





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 tree-shaped structure is known as the **Dendrogram**.

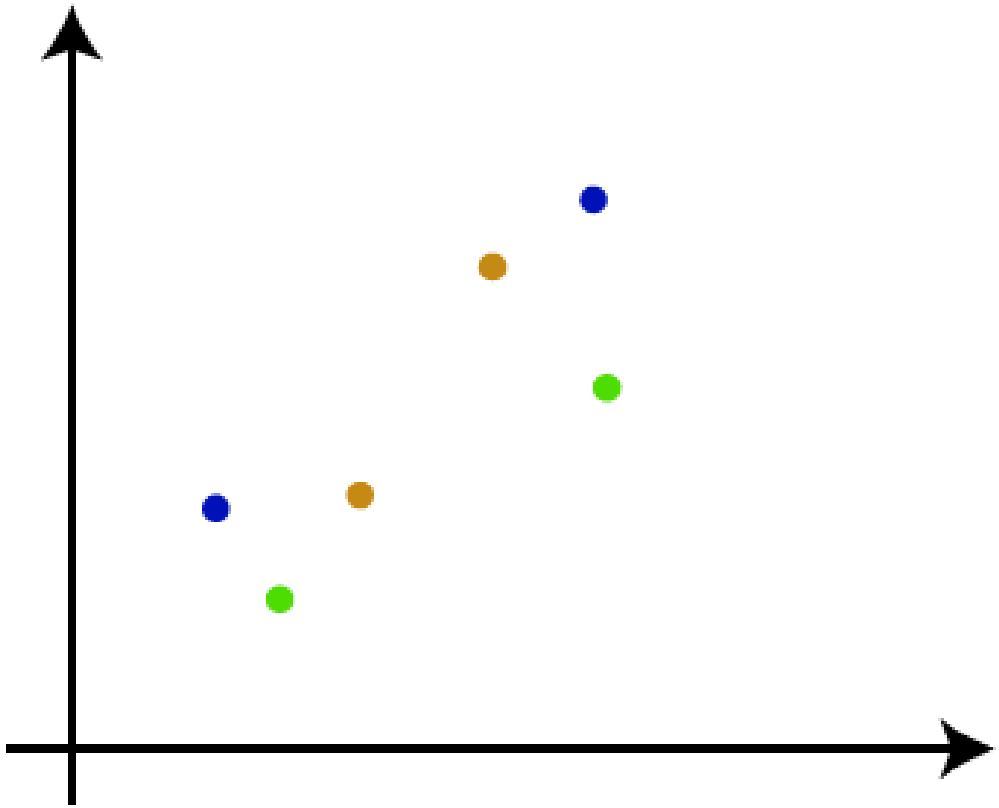
- **Agglomerative:** Agglomerative is a **bottom-up** approach, in which the algorithm starts with taking all data points as single clusters and merging them until one cluster is left.
- **Divisive:** Divisive algorithm is the reverse of the agglomerative algorithm as it is a **top-down approach.**

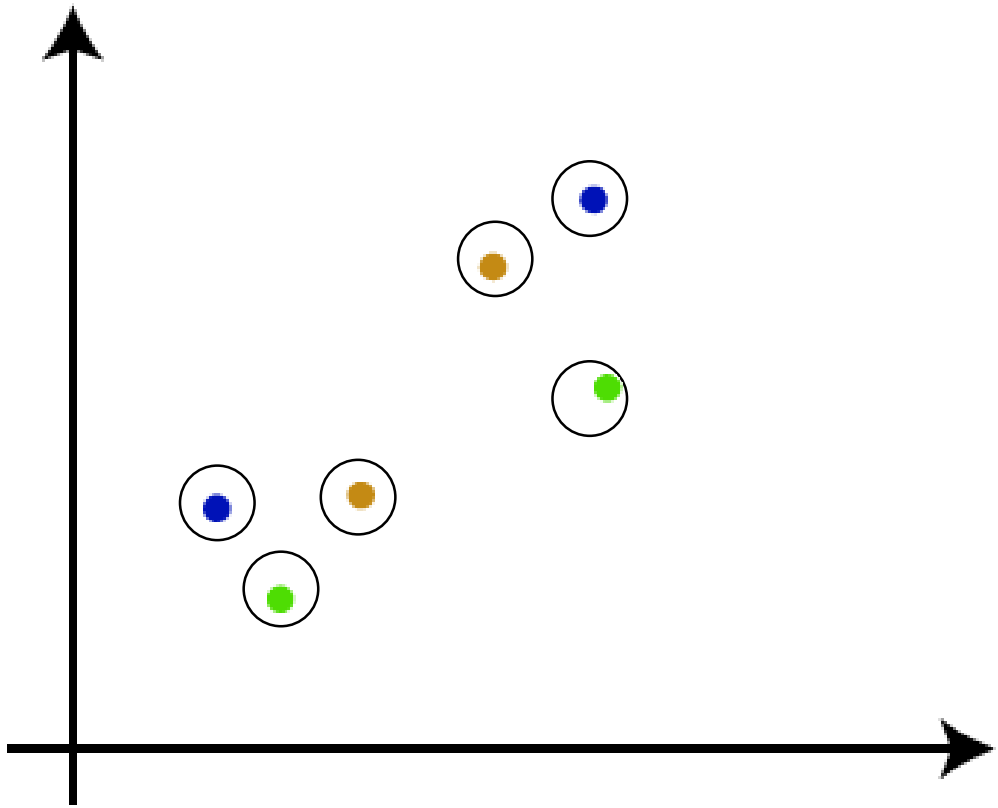
Agglomerative Hierarchical Clustering (AHC)

- 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 data points 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.

Agglomerative Hierarchical Clustering (AHC)

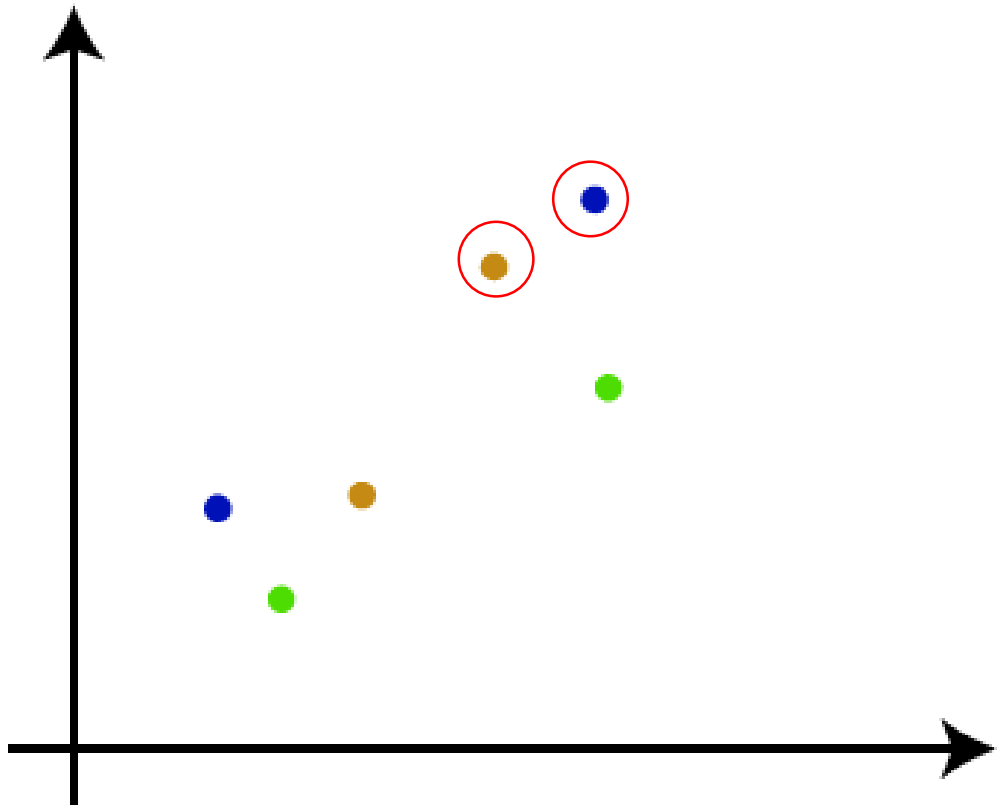
Step-1: Create each data point as a single cluster. Let's say there are N data points, so the number of clusters will also be N.

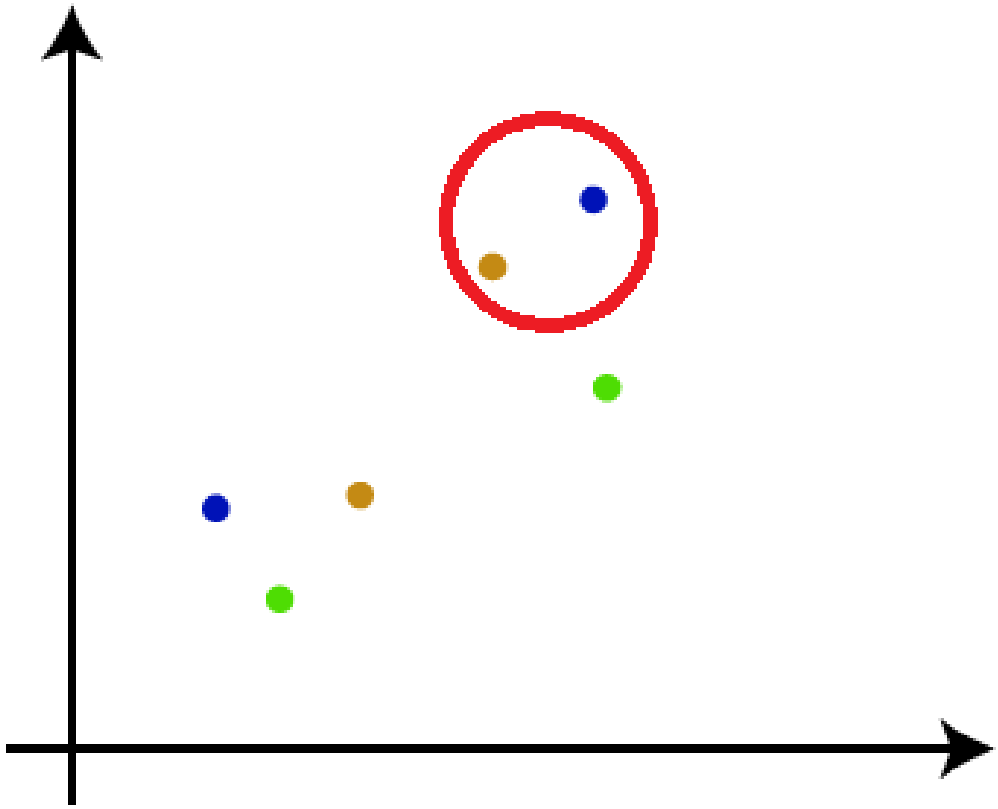




Agglomerative Hierarchical Clustering (AHC)

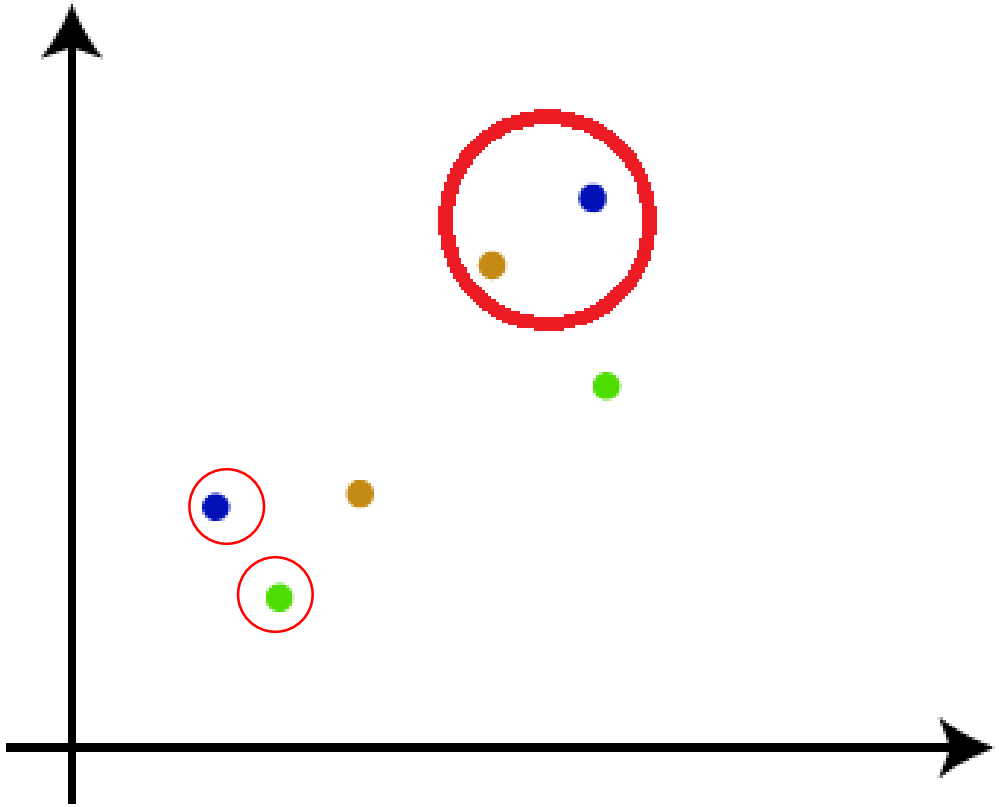
Step-2: Take two closest data points or clusters and merge them to form one cluster. So, there will now be N-1 clusters.

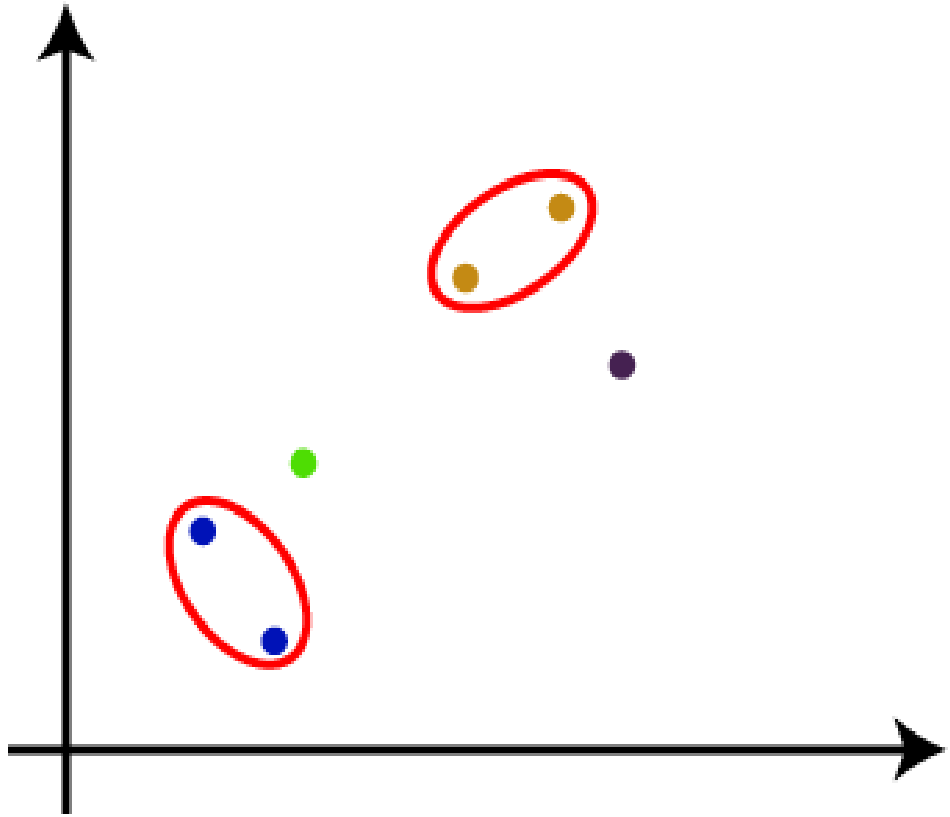




Agglomerative Hierarchical Clustering (AHC)

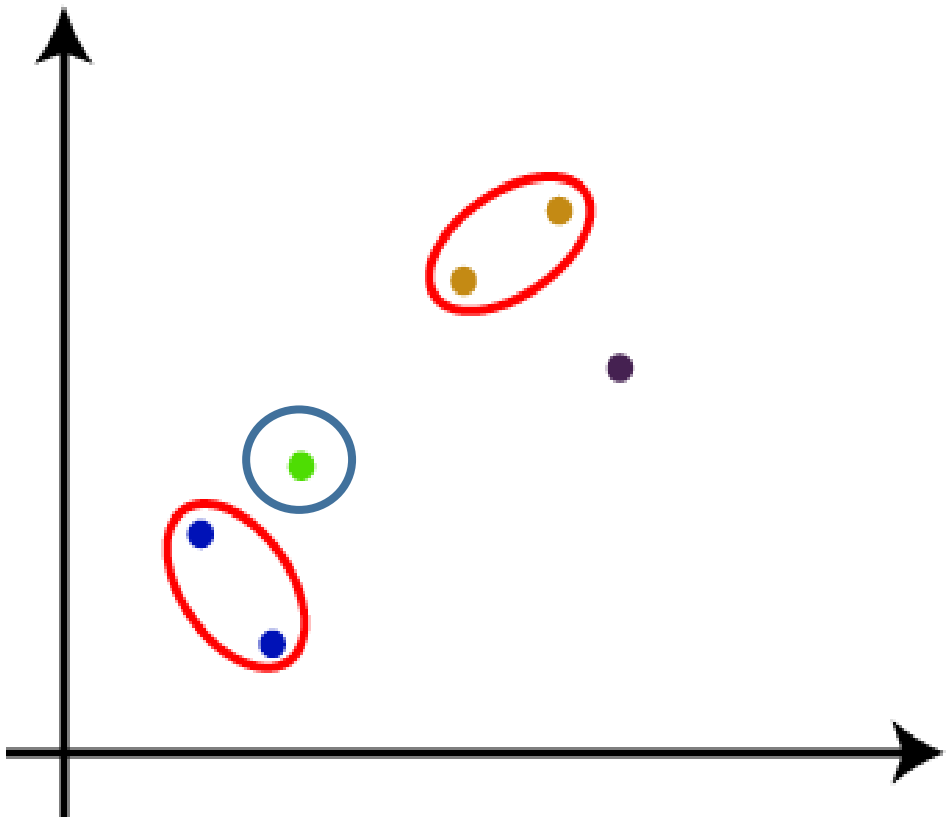
Step-3: Again, take the two closest clusters and merge them together to form one cluster. There will be N-2 clusters.

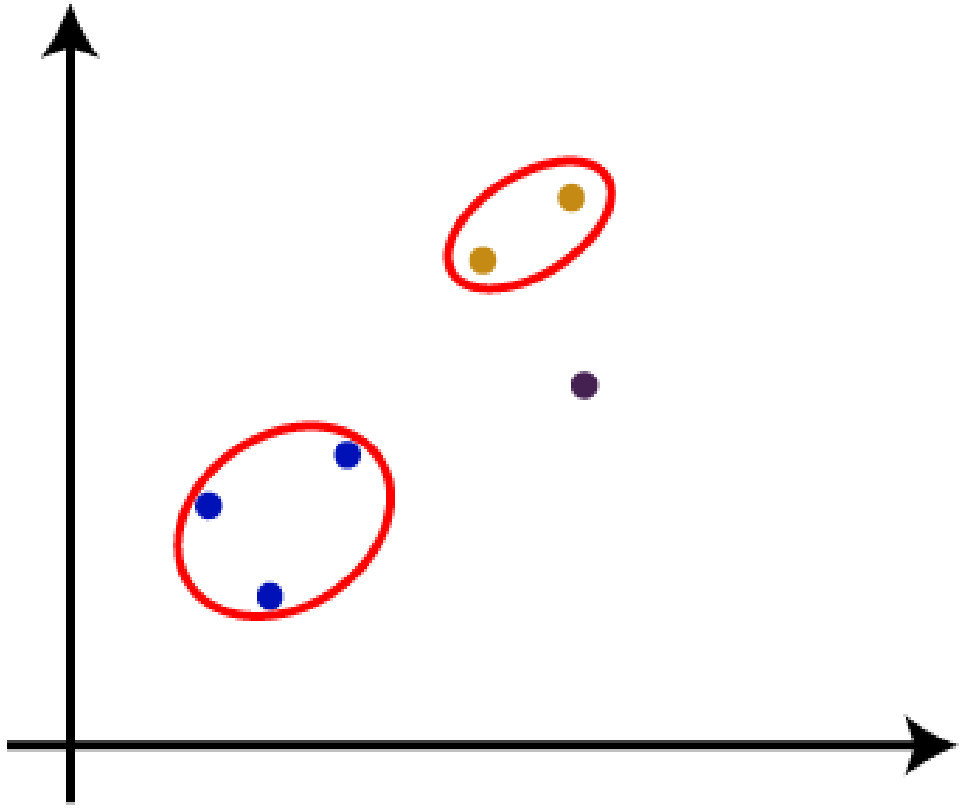




Agglomerative Hierarchical Clustering (AHC)

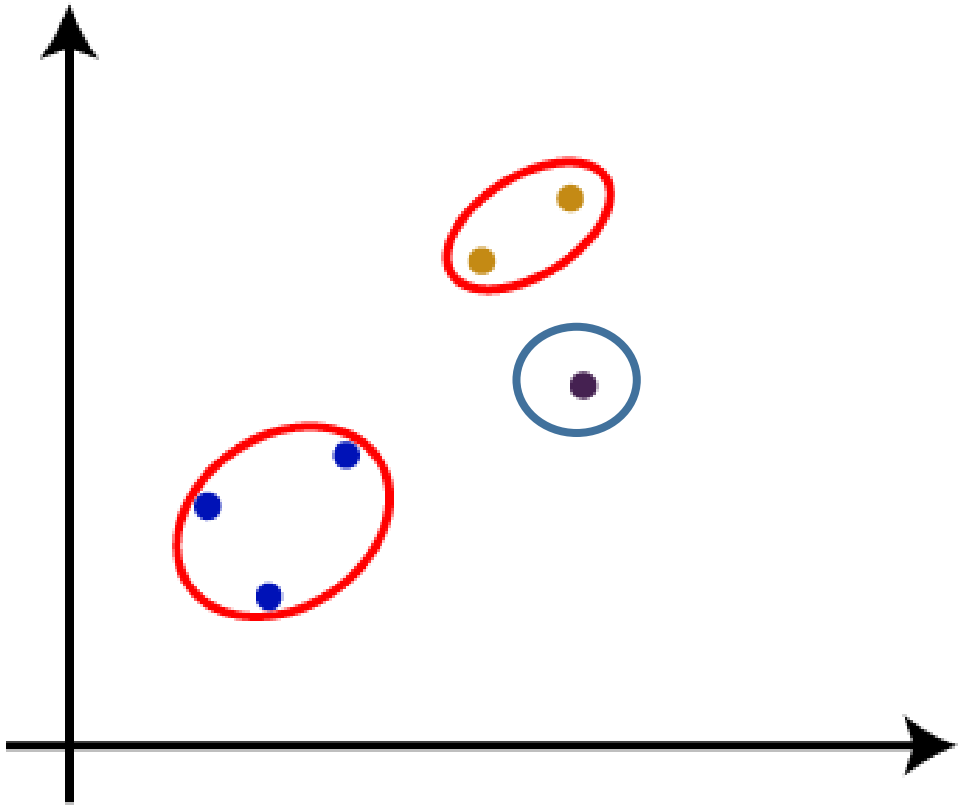
Step-4: Repeat Step 3 until only one cluster left. So, we will get the following clusters. Consider the below images:

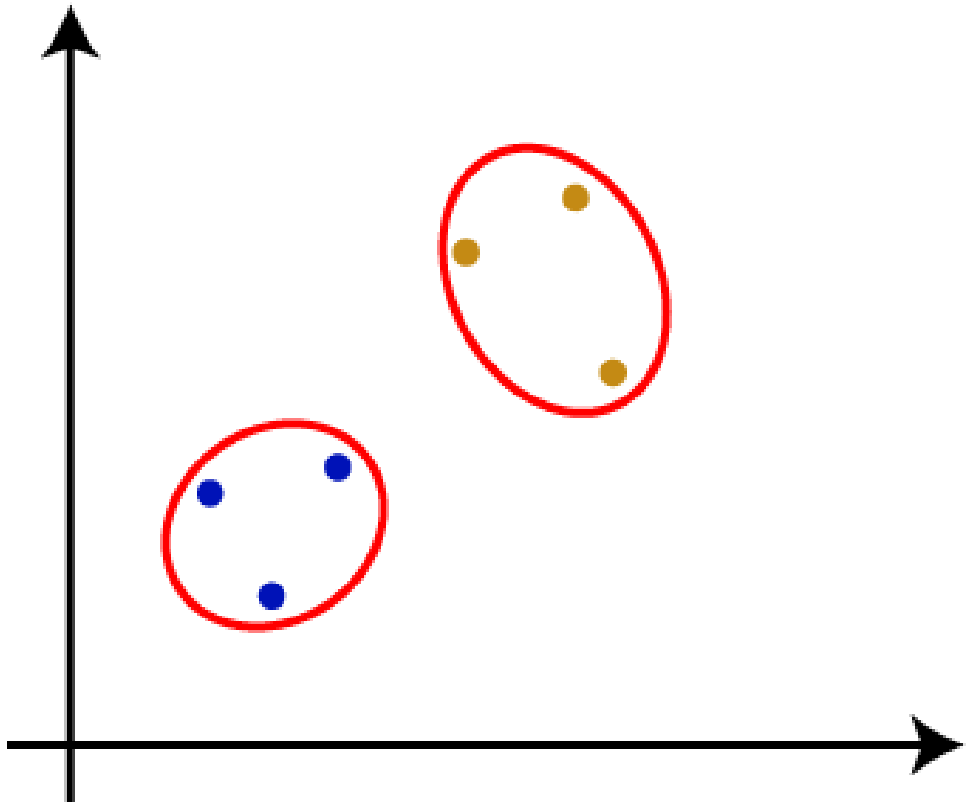




Agglomerative Hierarchical Clustering (AHC)

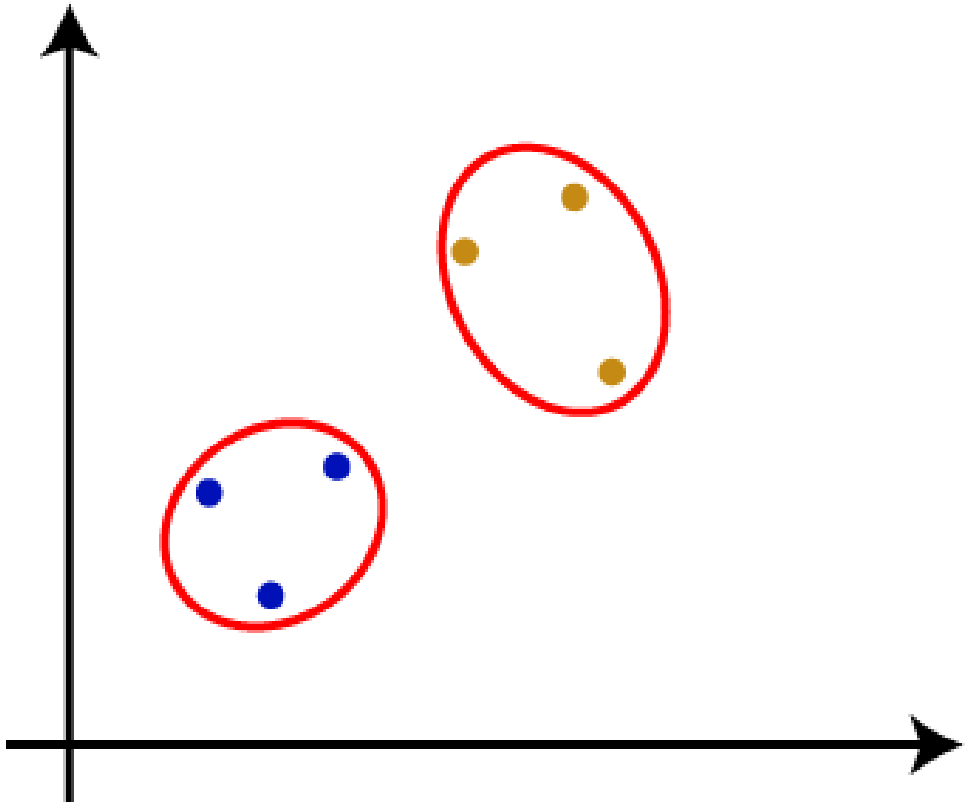
Step-4: Repeat Step 3 until only one cluster left. So, we will get the following clusters. Consider the below images:

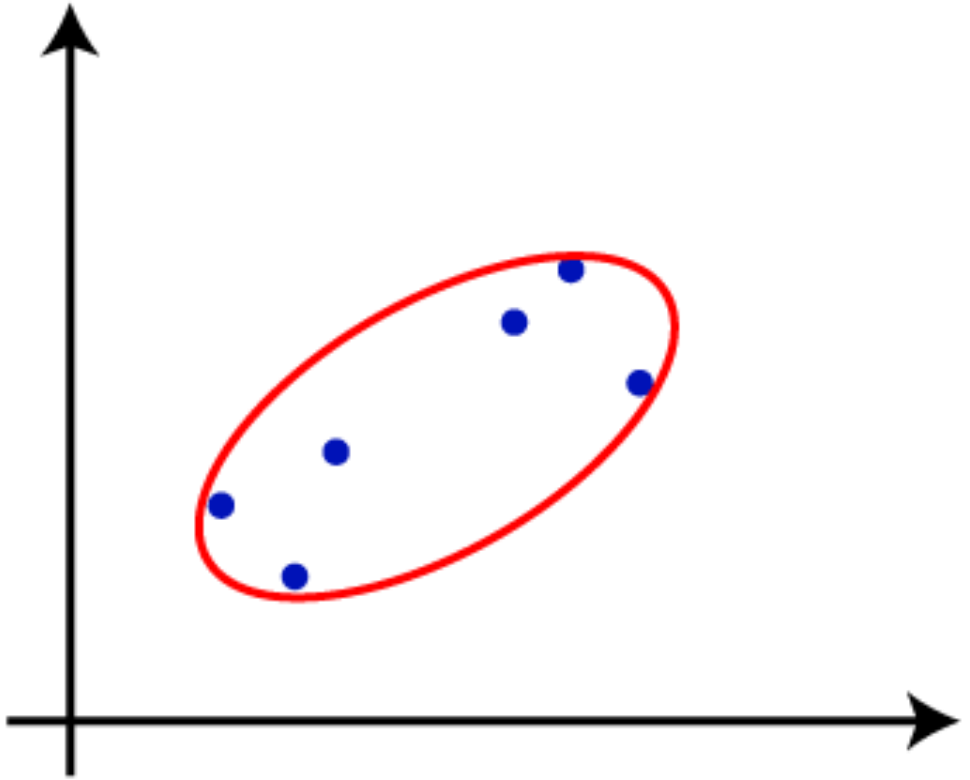




Agglomerative Hierarchical Clustering (AHC)

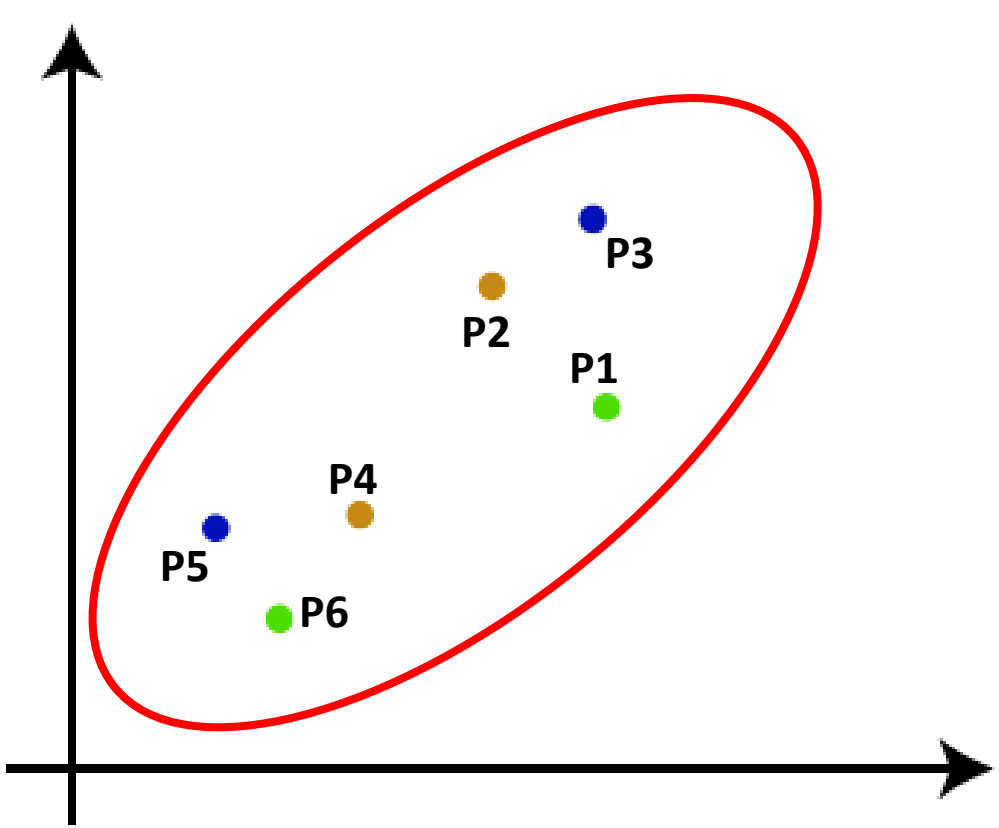
Step-4: Repeat Step 3 until only one cluster left. So, we will get the following clusters. Consider the below images:

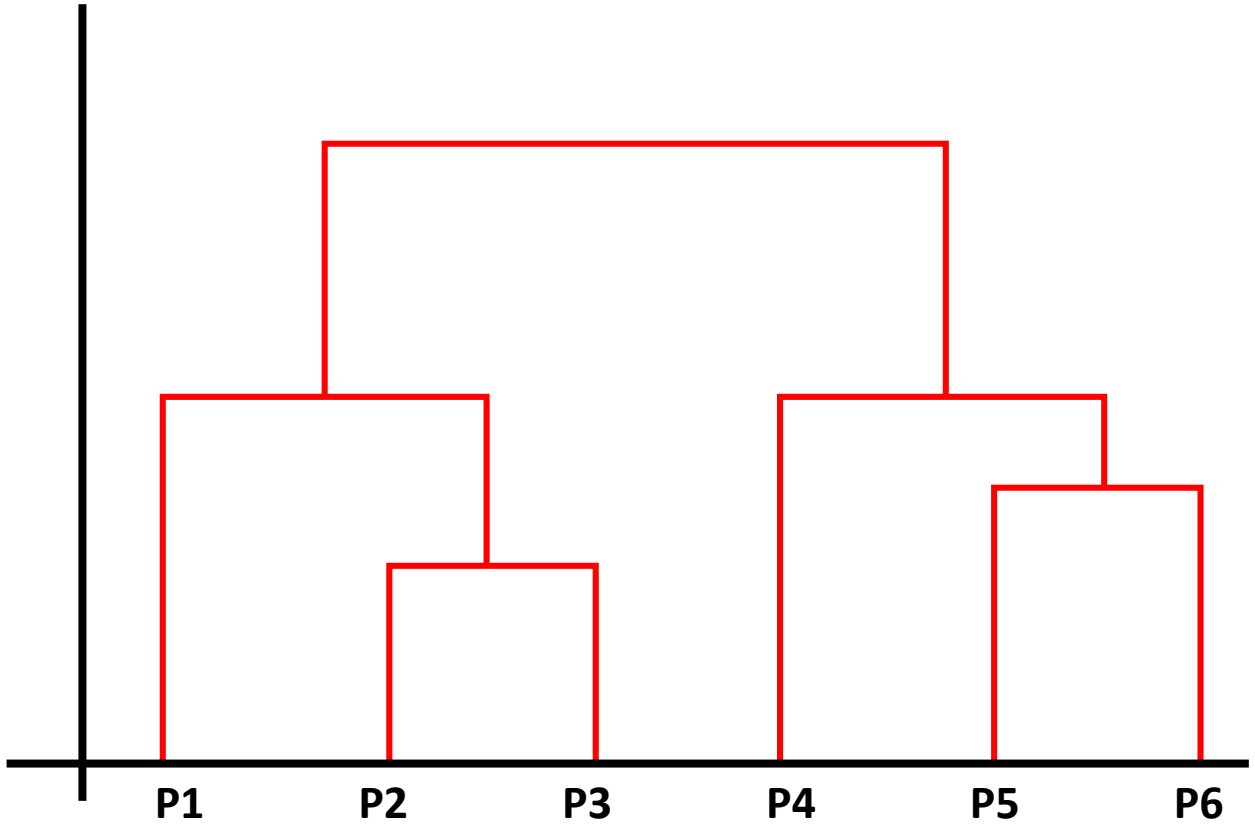




Agglomerative Hierarchical Clustering (AHC)

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

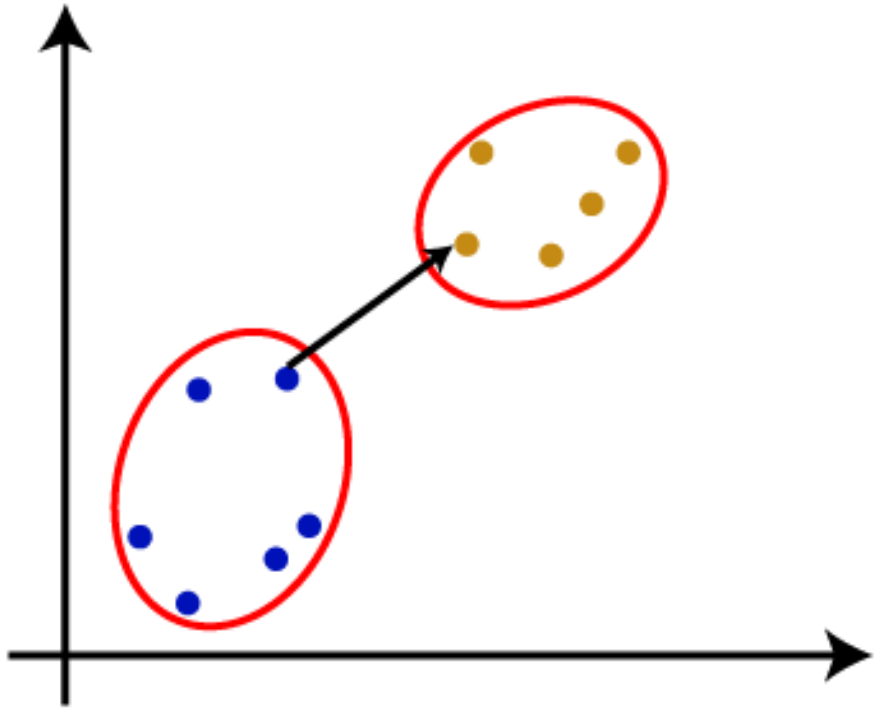




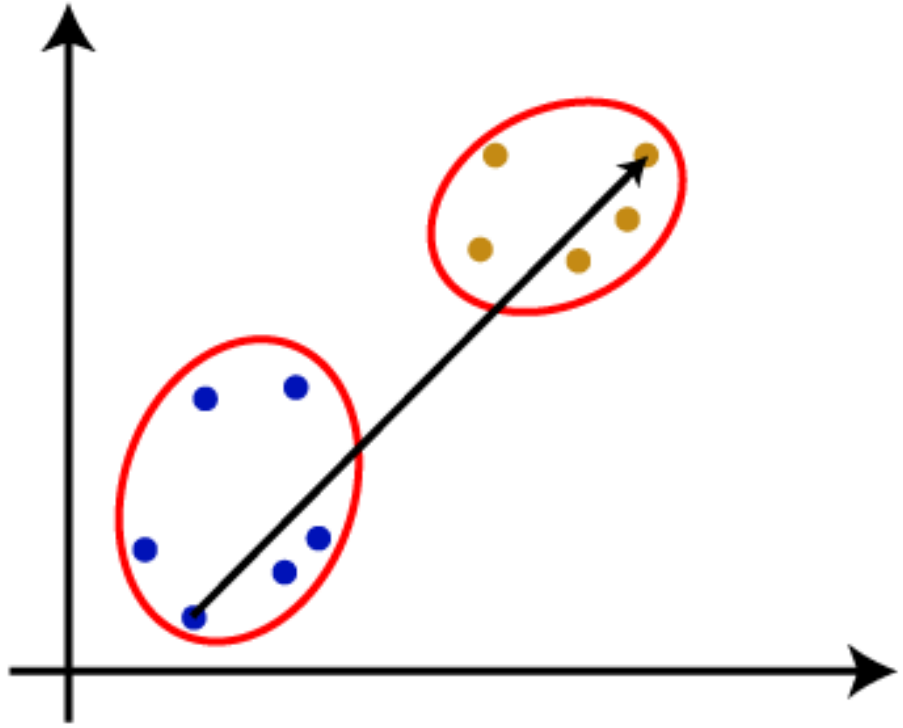
Distance Metrics used in AHC

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**.

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

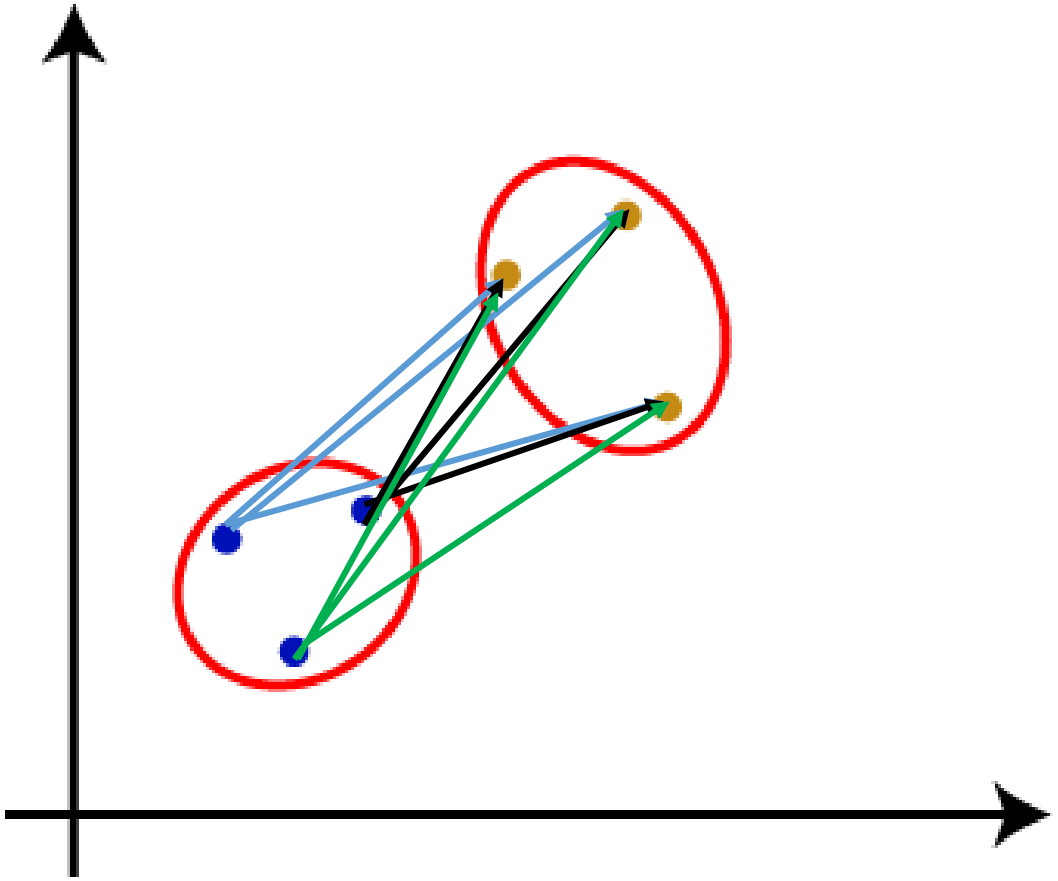


Complete Linkage: It is the farthest distance between the two points of two different clusters.

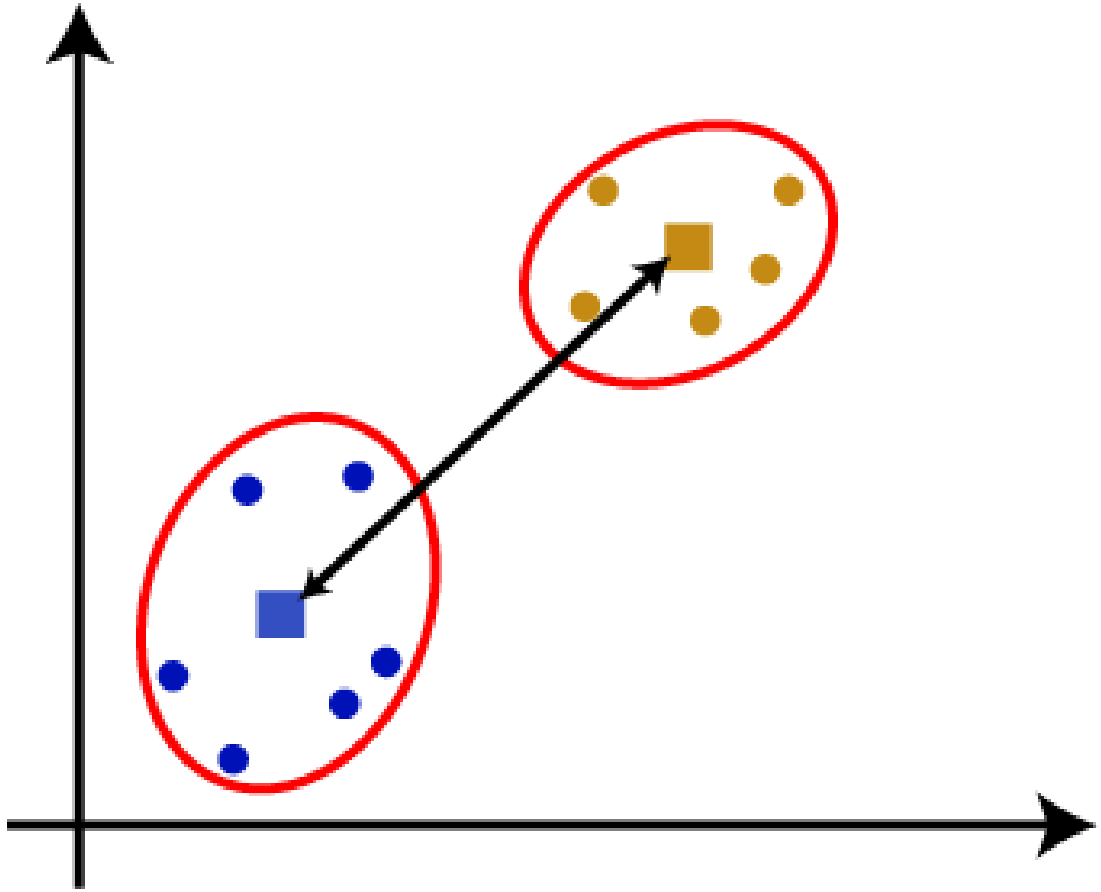


Distance Metrics used in AHC

Average Linkage: Distance between each pair of datasets is added up and then divided by the total number of data points to calculate the average distance between two clusters.



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



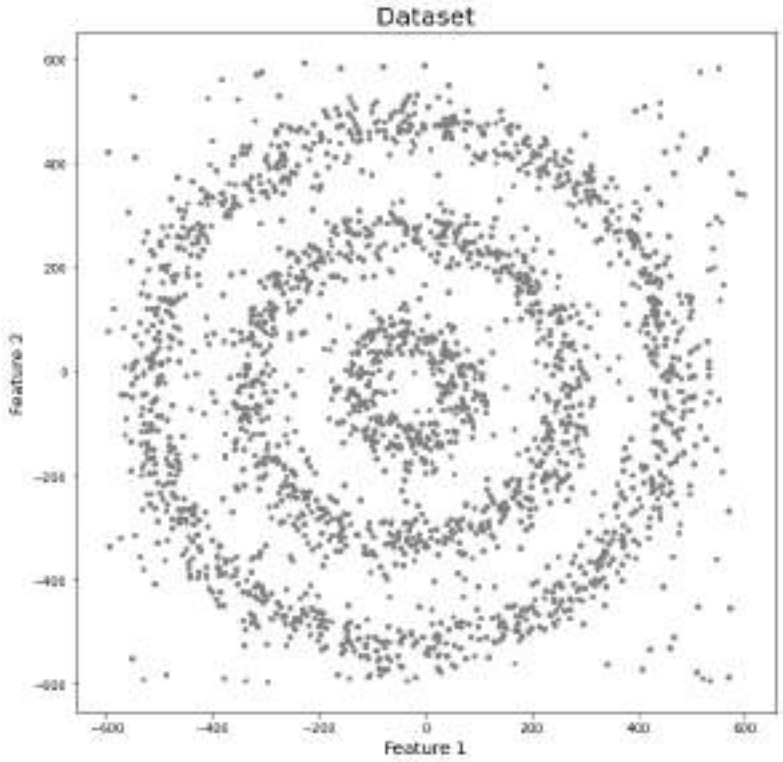
Density-based Clustering

K-mean, K-medoids and Hierarchical Clustering:

- Prone to outliers or noise.
- Suitable for compact data points and well separated data points.

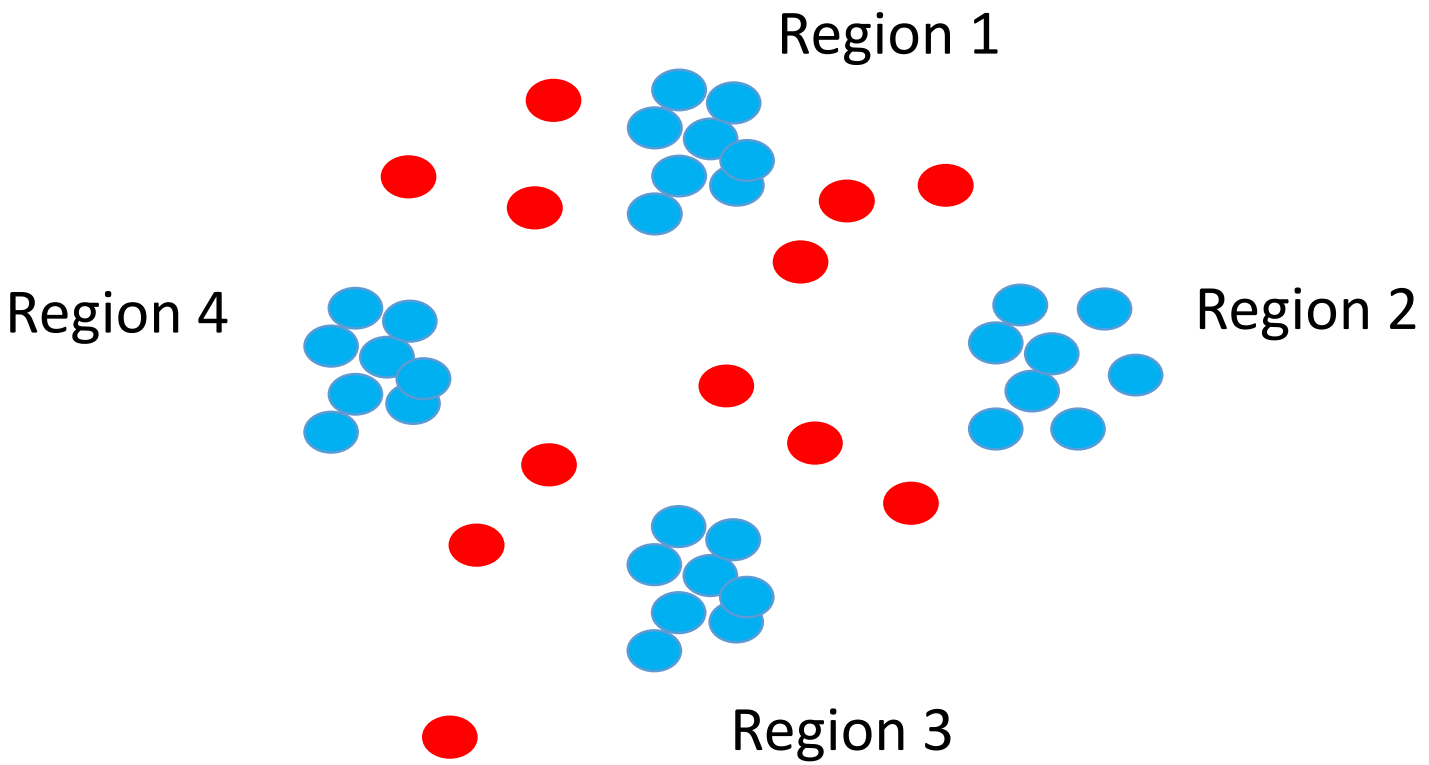
Real life data may contain irregularities, like:

- Clusters can be of arbitrary shape such as those shown in the figure below.
- Data may contain noise.



Density-based Clustering

Density-Based Clustering refers to **unsupervised learning methods** that identify distinctive groups/clusters in the data, based on the idea that a cluster in data space is a **contiguous region of high point density**, separated from other such clusters by **contiguous regions of low point density**.



DBSCAN

- **Density-Based Spatial Clustering of Applications with Noise** (DBSCAN) is a base algorithm for density-based clustering.

- It can discover clusters of different shapes and sizes from a large amount of data, which is containing noise and outliers.

The DBSCAN algorithm uses two parameters:

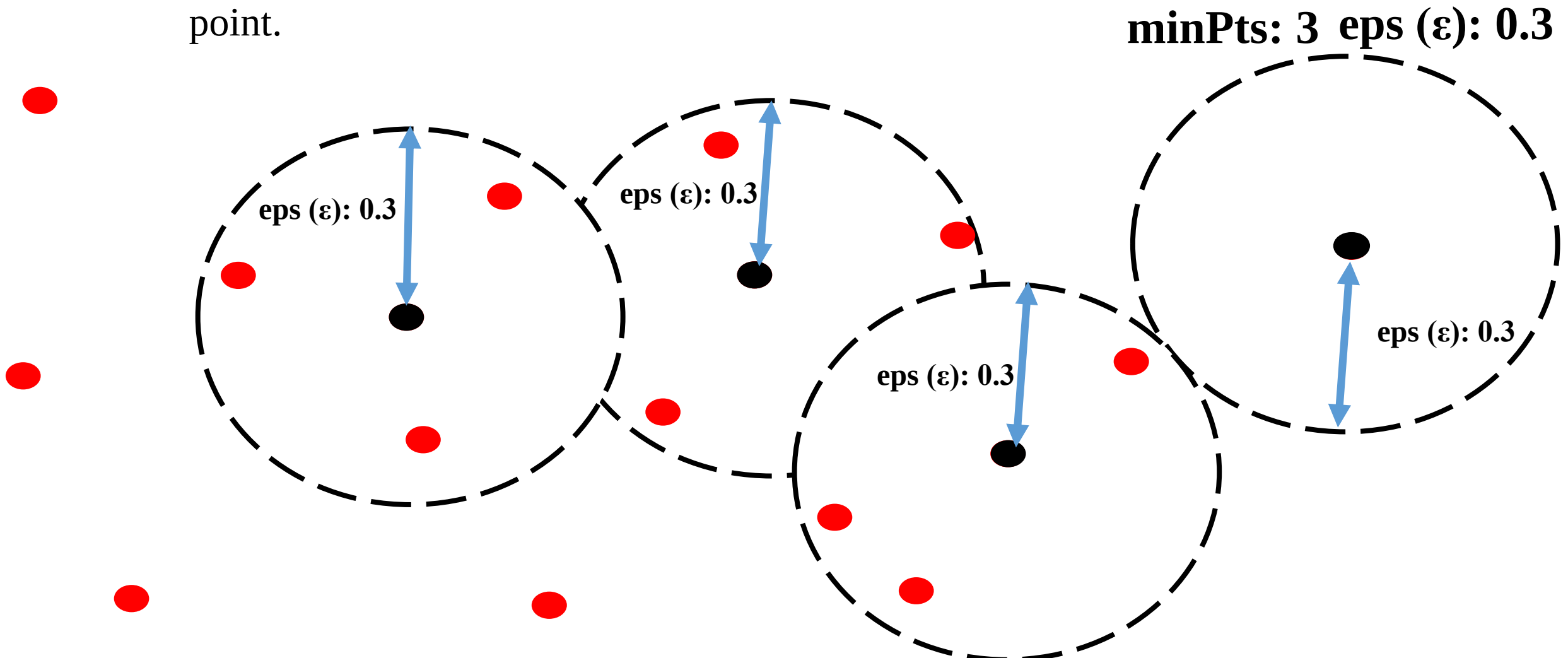
- **minPts:** The minimum number of points (a threshold) clustered together for a region to be considered dense.

- **eps (ε):** A distance measure that will be used to locate the points in the neighborhood of any point.

DBSCAN

The DBSCAN algorithm uses two parameters:

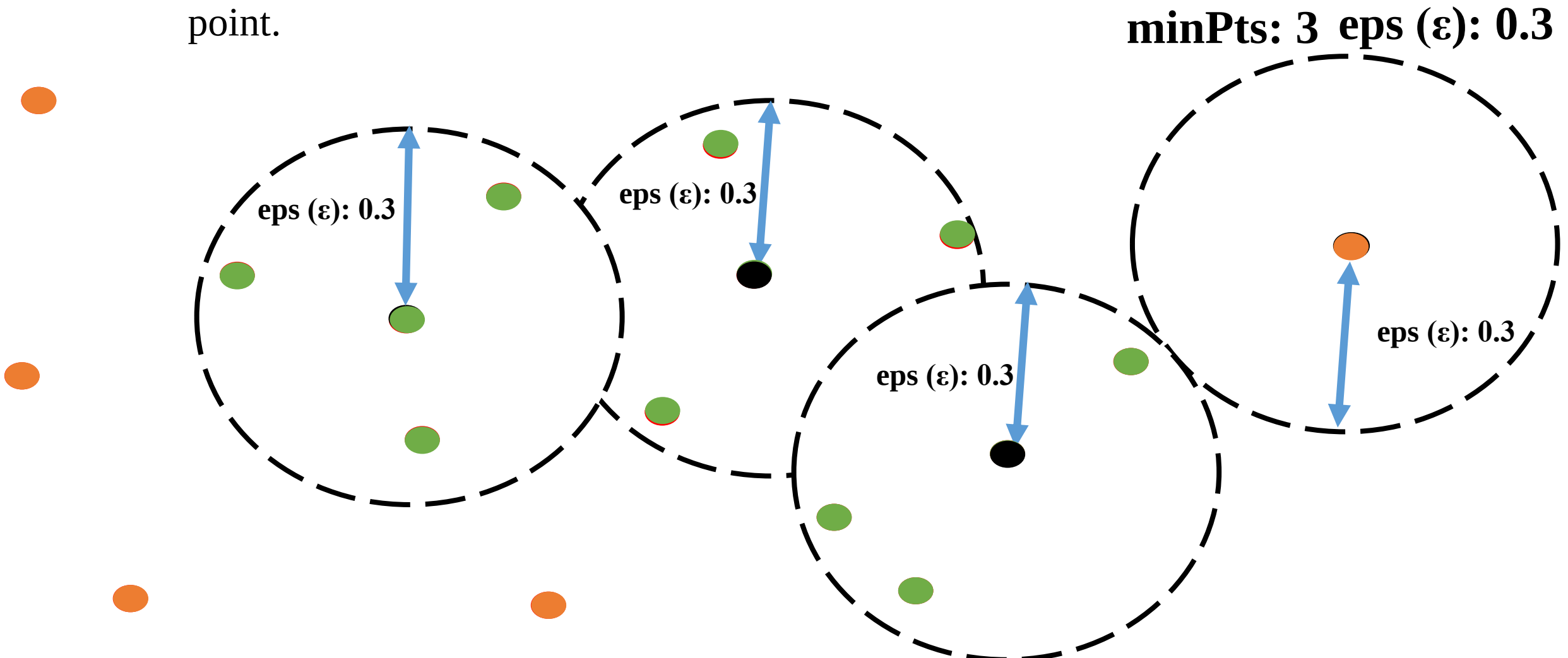
- **minPts:** The minimum number of points (a threshold) clustered together for a region to be considered dense.
- **eps (ε):** A distance measure that will be used to locate the points in the neighborhood of any point.



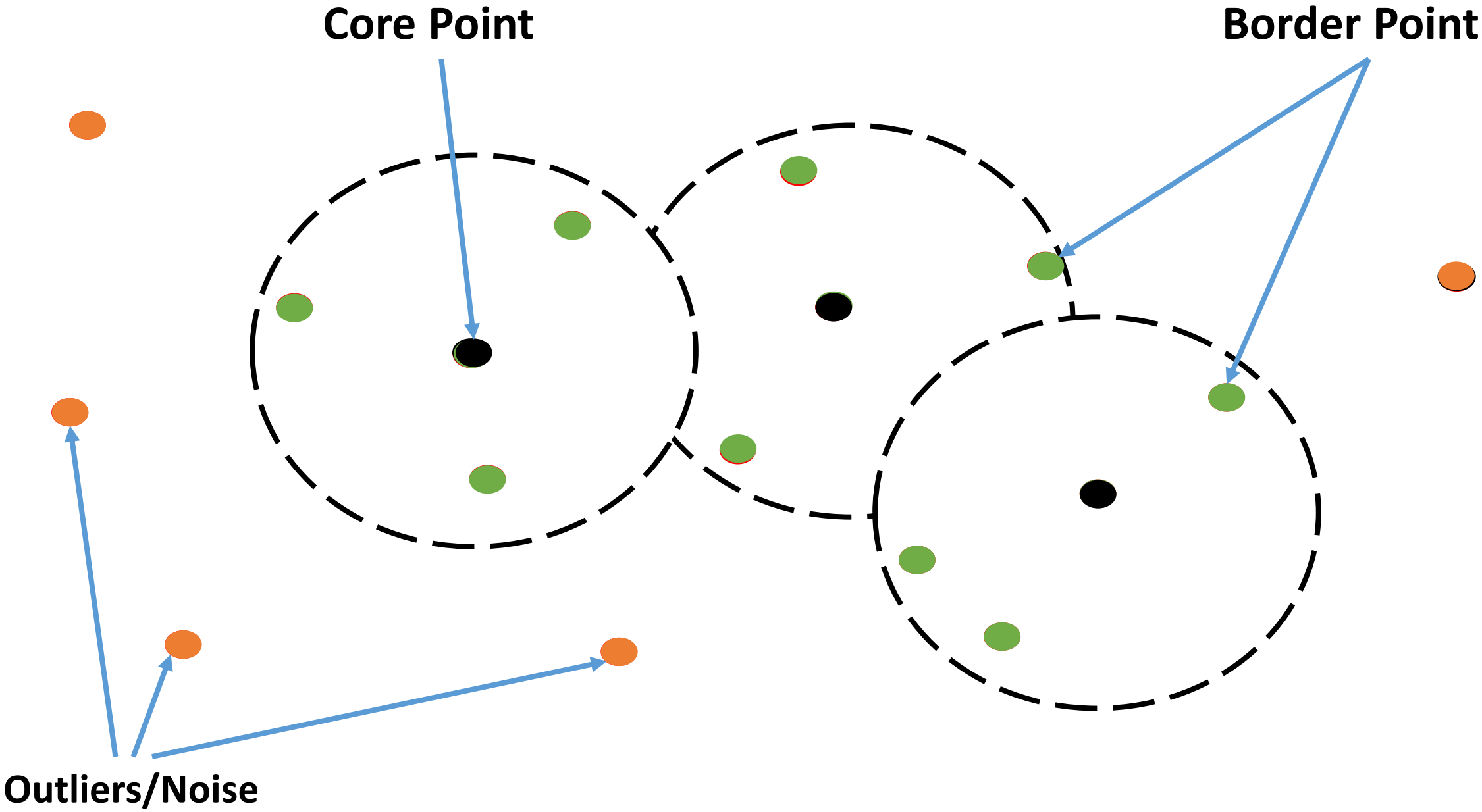
DBSCAN

The DBSCAN algorithm uses two parameters:

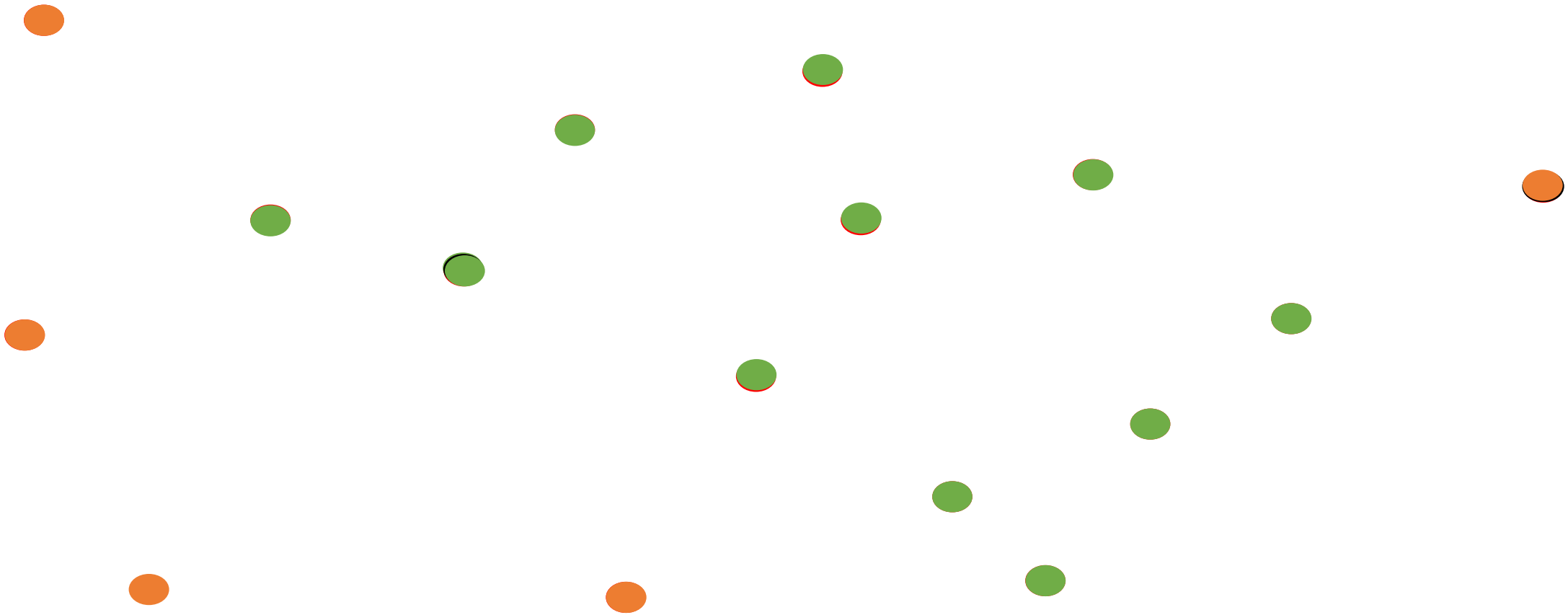
- **minPts:** The minimum number of points (a threshold) clustered together for a region to be considered dense.
- **eps (ε):** A distance measure that will be used to locate the points in the neighborhood of any point.

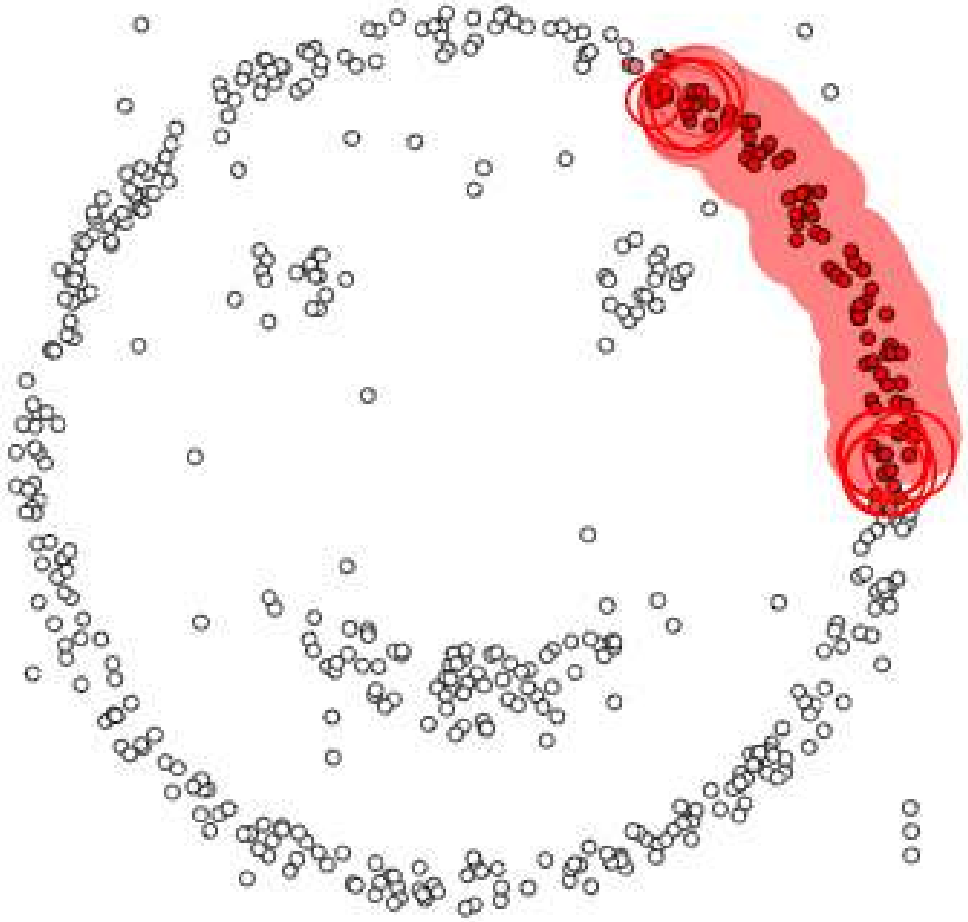


DBSCAN



DBSCAN

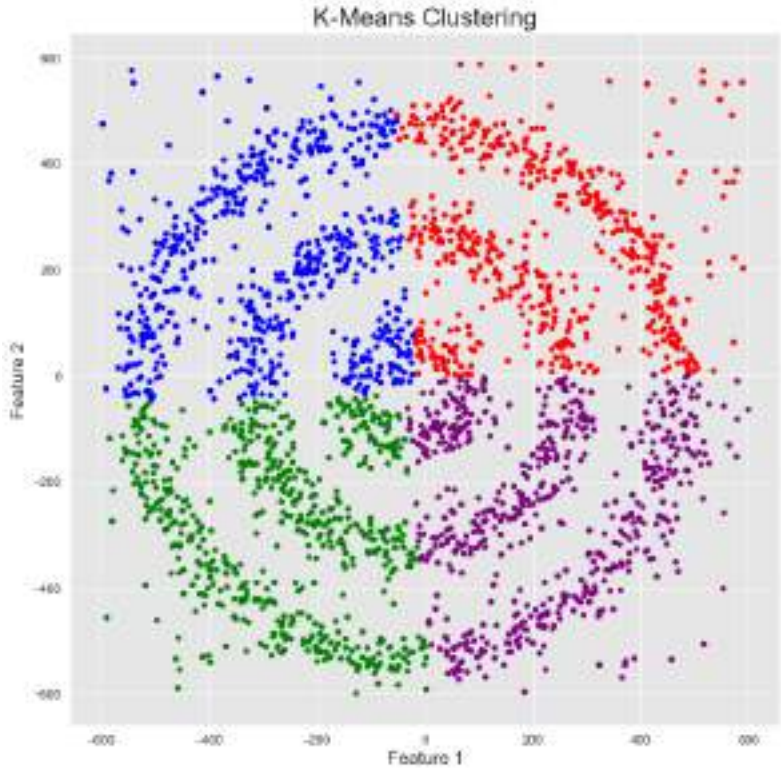


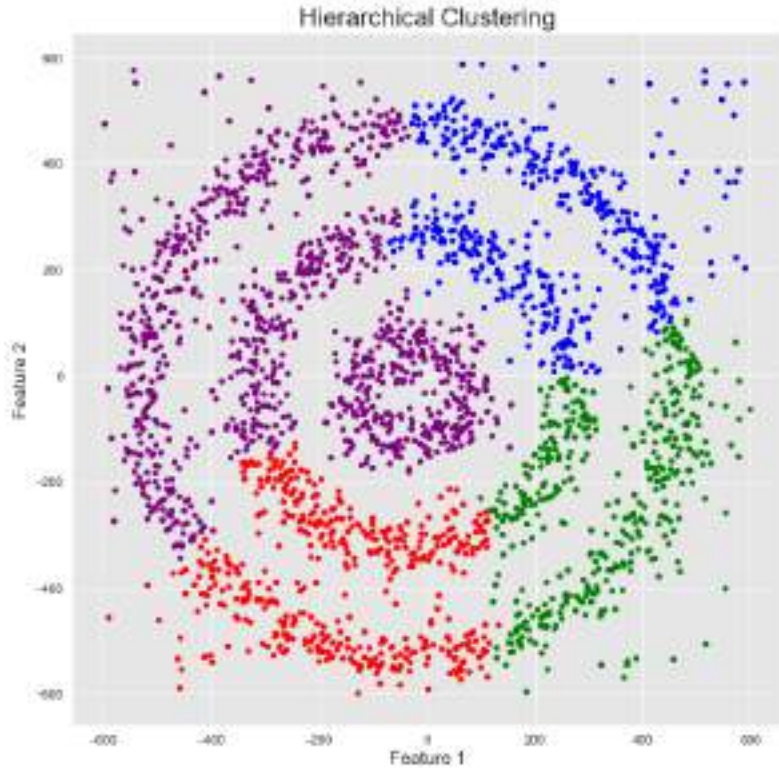


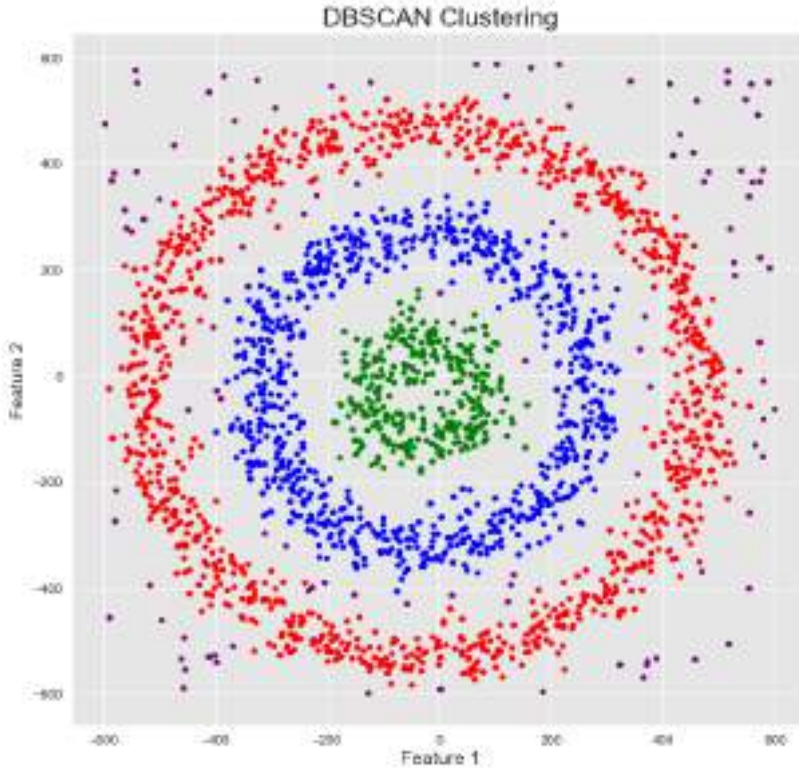
epsilon = 1.00
minPoints = 4

Restart

Pause


K-Means Clustering
Feature 2
Feature 1


Hierarchical Clustering
Feature 2
Feature 1


DBSCAN Clustering
Feature 2
Feature 1

Spectral Clustering

- Spectral clustering is a technique with roots in graph theory, where the approach is used to identify communities of nodes in a graph based on the edges connecting them.

Three Basic Steps in Spectral Clustering

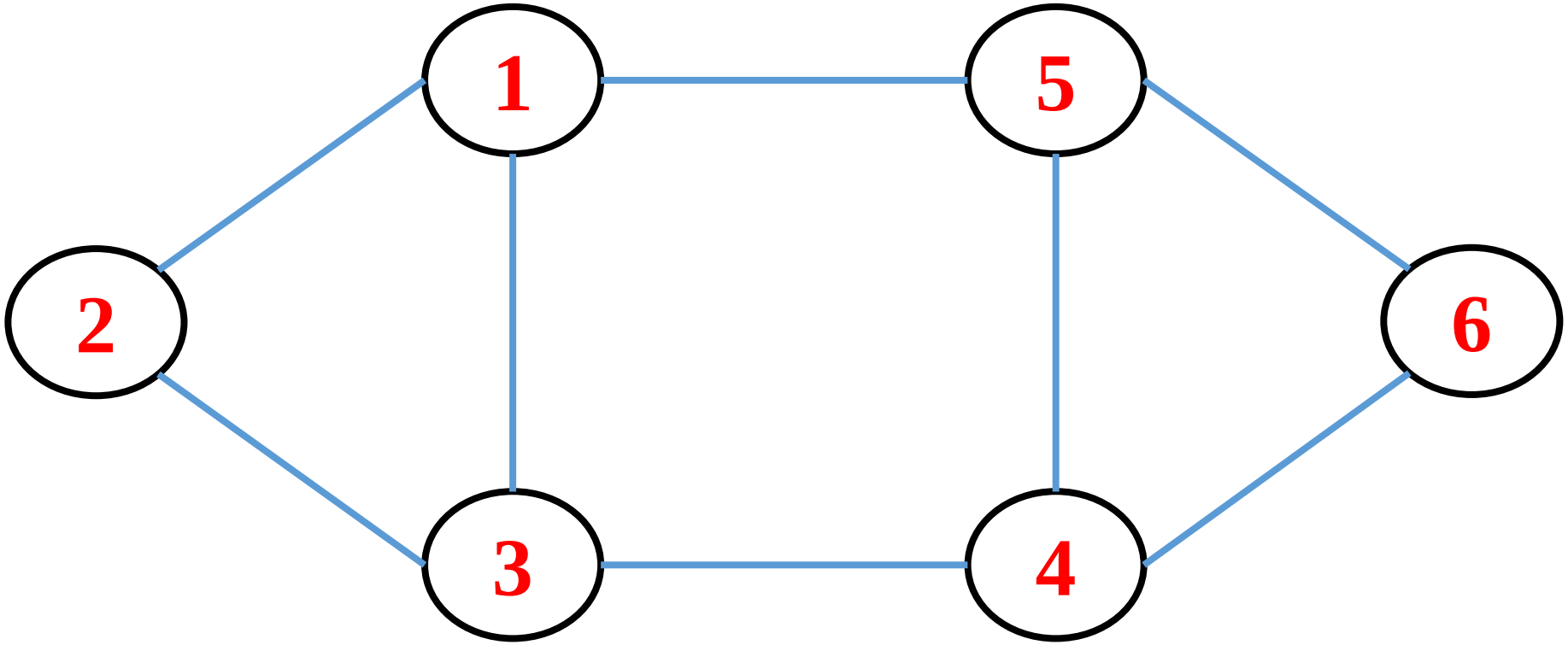
1. **Preprocessing:** Construct a Matrix Representation of Graph.

2. **Decomposition:** Compute Eigenvalues and Eigenvectors of the matrix.

3. **Grouping/Clustering:** Assign the data points to the cluster.

Spectral Clustering

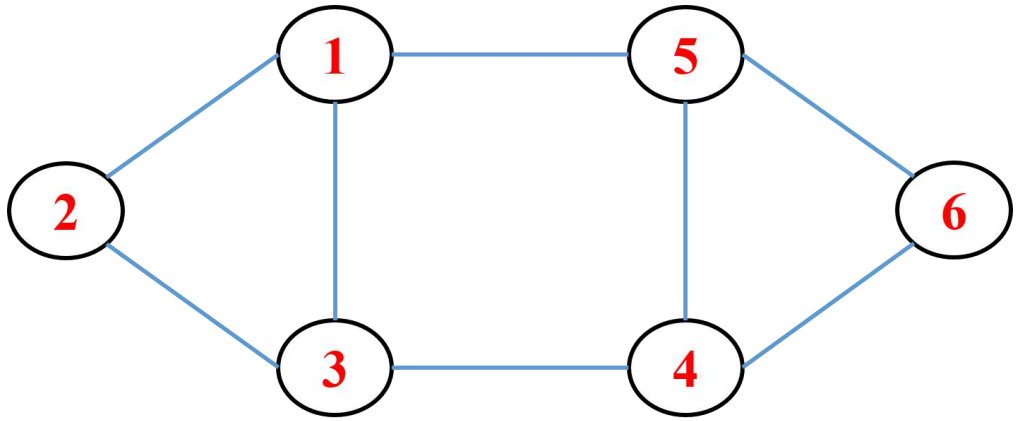
Preprocessing: Construct a Matrix Representation of Graph.



Spectral Clustering

Preprocessing: Construct a Laplacian Matrix Representation of Graph.

To Construct a Laplacian Matrix (L): First Find **Adjacency Matrix (A)**.

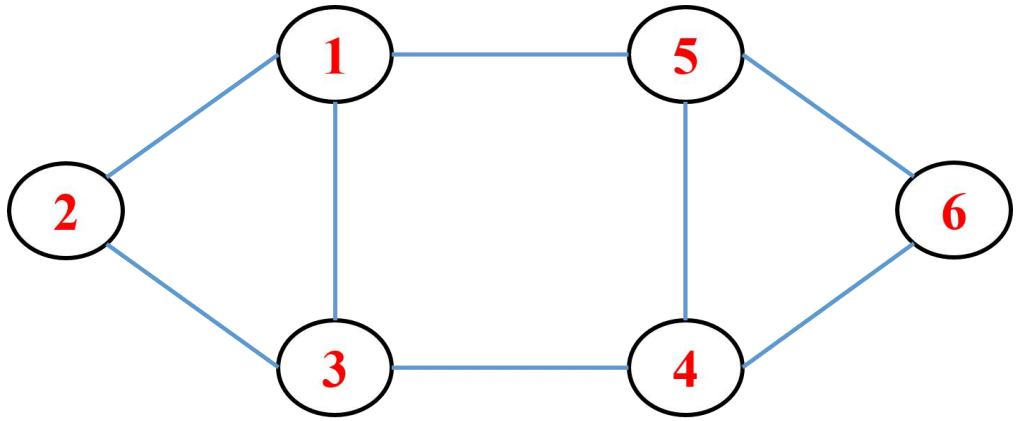


A	1	2	3	4	5	6
1	0	1	1	0	1	0
2	1	0	1	0	0	0
3	1	1	0	1	0	0
4	0	0	1	0	1	1
5	1	0	0	1	0	1
6	0	0	0	1	1	0

Spectral Clustering

Preprocessing: Construct a Laplacian Matrix Representation of Graph.

To Construct a Laplacian Matrix (L): Second Find **Degree Matrix (D)**.



D	1	2	3	4	5	6
1	3	0	0	0	0	0
2	0	2	0	0	0	0
3	0	0	3	0	0	0
4	0	0	0	3	0	0
5	0	0	0	0	3	0
6	0	0	0	0	0	2

Spectral Clustering

Preprocessing: Construct a Laplacian Matrix Representation of Graph.

To Construct a Laplacian Matrix (L): **L = D - A**

L =

D	1	2	3	4	5	6
1	3	0	0	0	0	0
2	0	2	0	0	0	0
3	0	0	3	0	0	0
4	0	0	0	3	0	0
5	0	0	0	0	3	0
6	0	0	0	0	0	2

D

-

A	1	2	3	4	5	6
1	0	1	1	0	1	0
2	1	0	1	0	0	0
3	1	1	0	1	0	0
4	0	0	1	0	1	1
5	1	0	0	1	0	1
6	0	0	0	1	1	0

A

Spectral Clustering

Preprocessing: Construct a Laplacian Matrix Representation of Graph.

To Construct a Laplacian Matrix (L): **L = D - A**

L =

L	1	2	3	4	5	6
1	3	-1	-1	0	-1	0
2	-1	2	-1	0	0	0
3	-1	-1	3	-1	0	0
4	0	0	-1	3	-1	-1
5	-1	0	0	-1	3	-1
6	0	0	0	-1	-1	2

Spectral Clustering

Decomposition: Compute Eigenvalues and Eigenvector for Laplacian Matrix (L).

	1	2	3	4	5	6
1	0.4	0.3	-1	2.1	0.1	0.2
2	0.2	0.6	0.4	2.6	0.3	0.3
3	0.1	0.3	0.5	0.1	0.2	0.5
4	0.6	-0.3	0.5	0.2	0.4	0.4
5	0.0	-0.3	0.2	0.4	0.5	0.5
6	0.0	-0.5	0.1	0.2	0.1	0.6

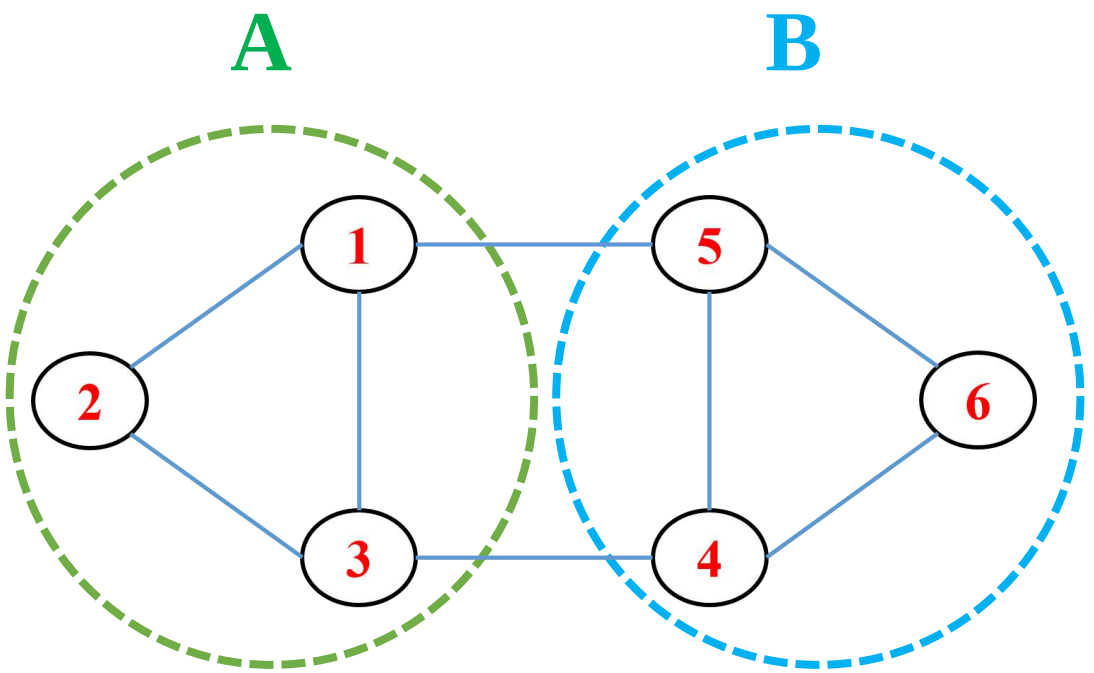
Spectral Clustering

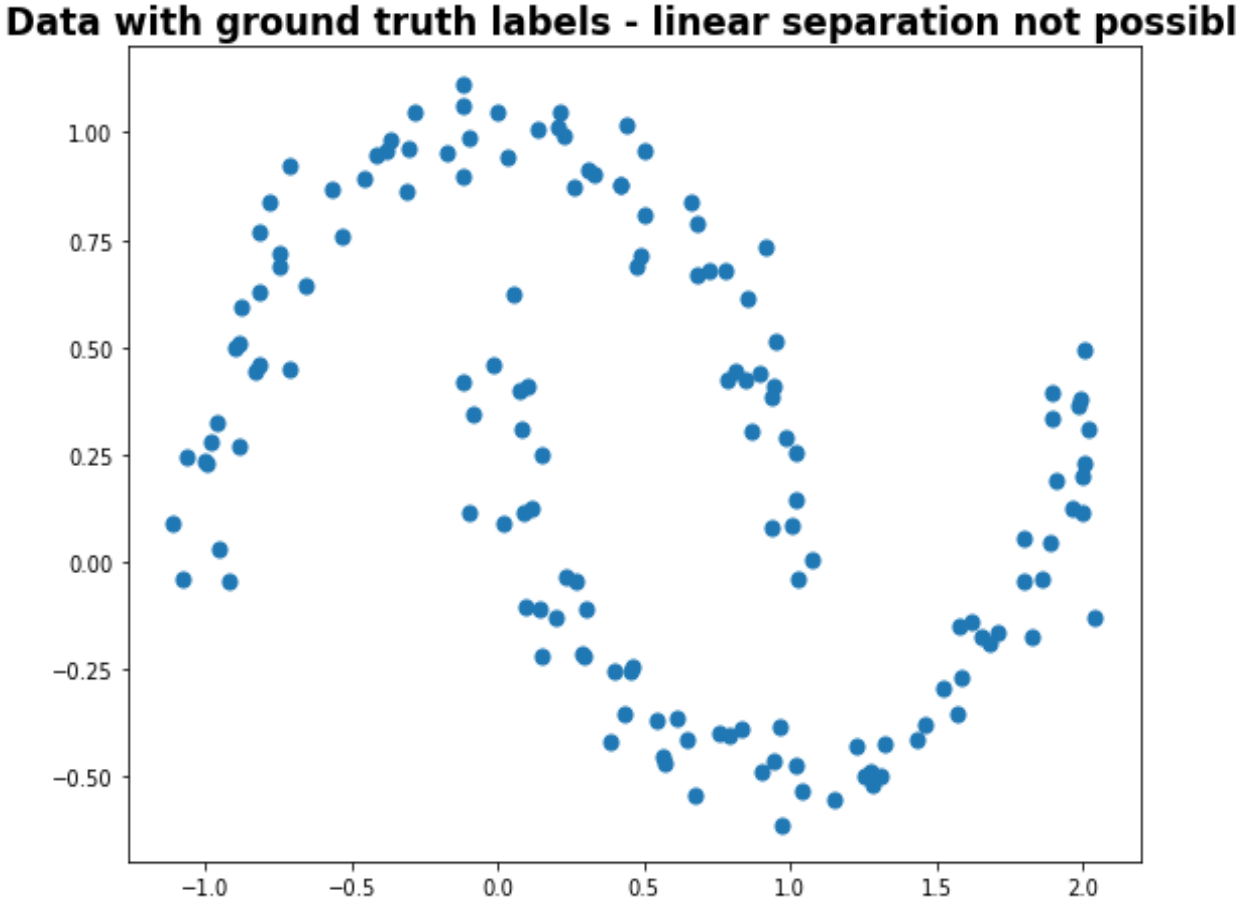
Decomposition: Compute Eigenvalues and Eigenvector for Laplacian Matrix (L).

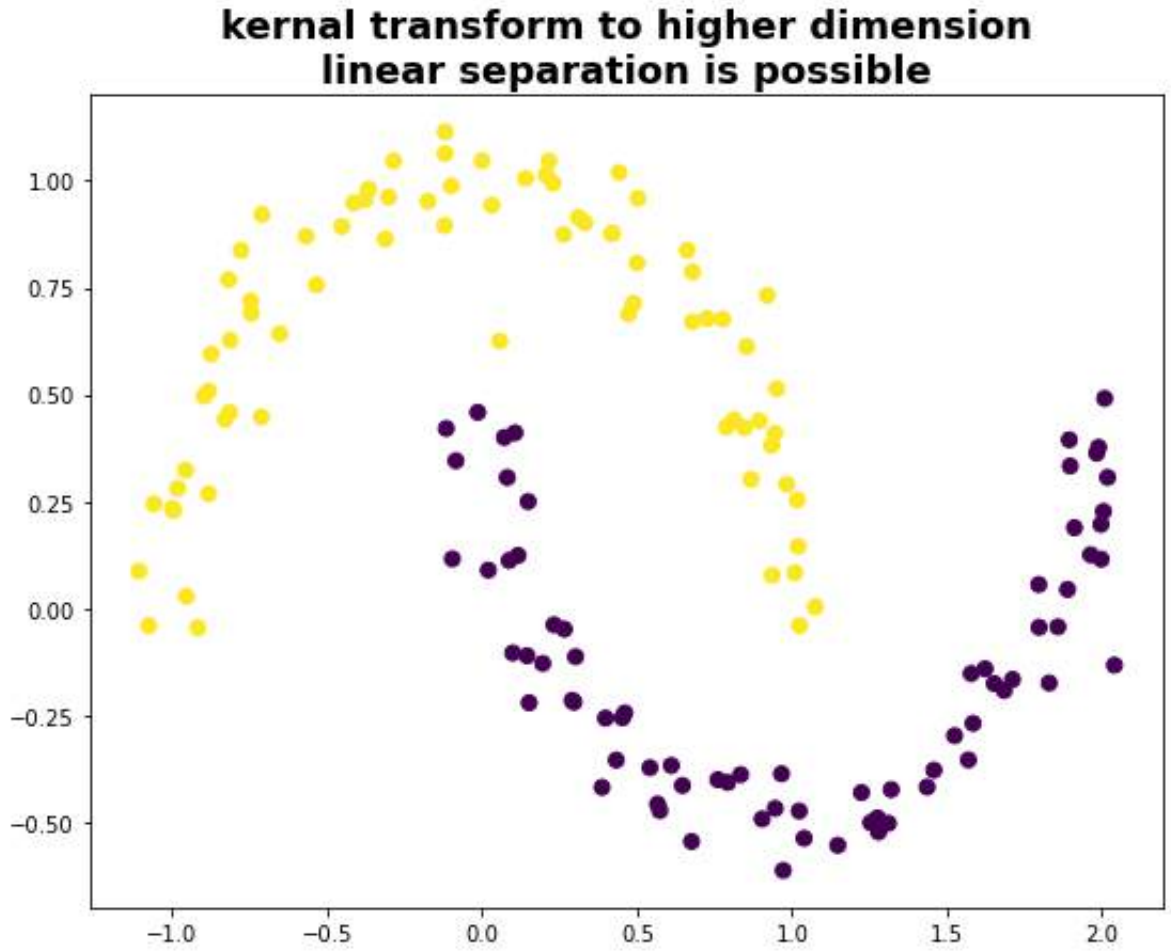
1	0.3
2	0.6
3	0.3
4	-0.3
5	-0.3
6	-0.5

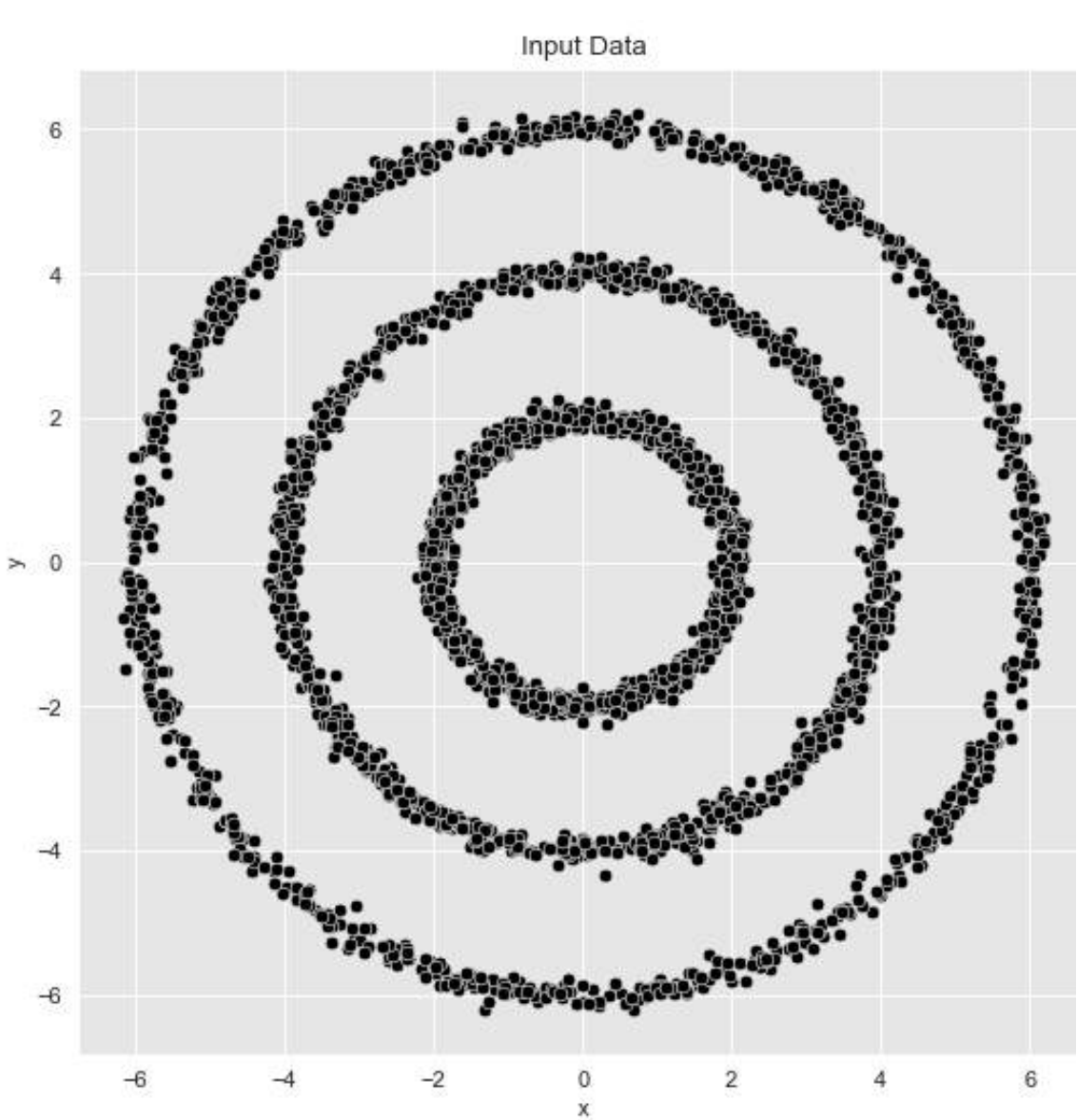
1	0.3
2	0.6
3	0.3

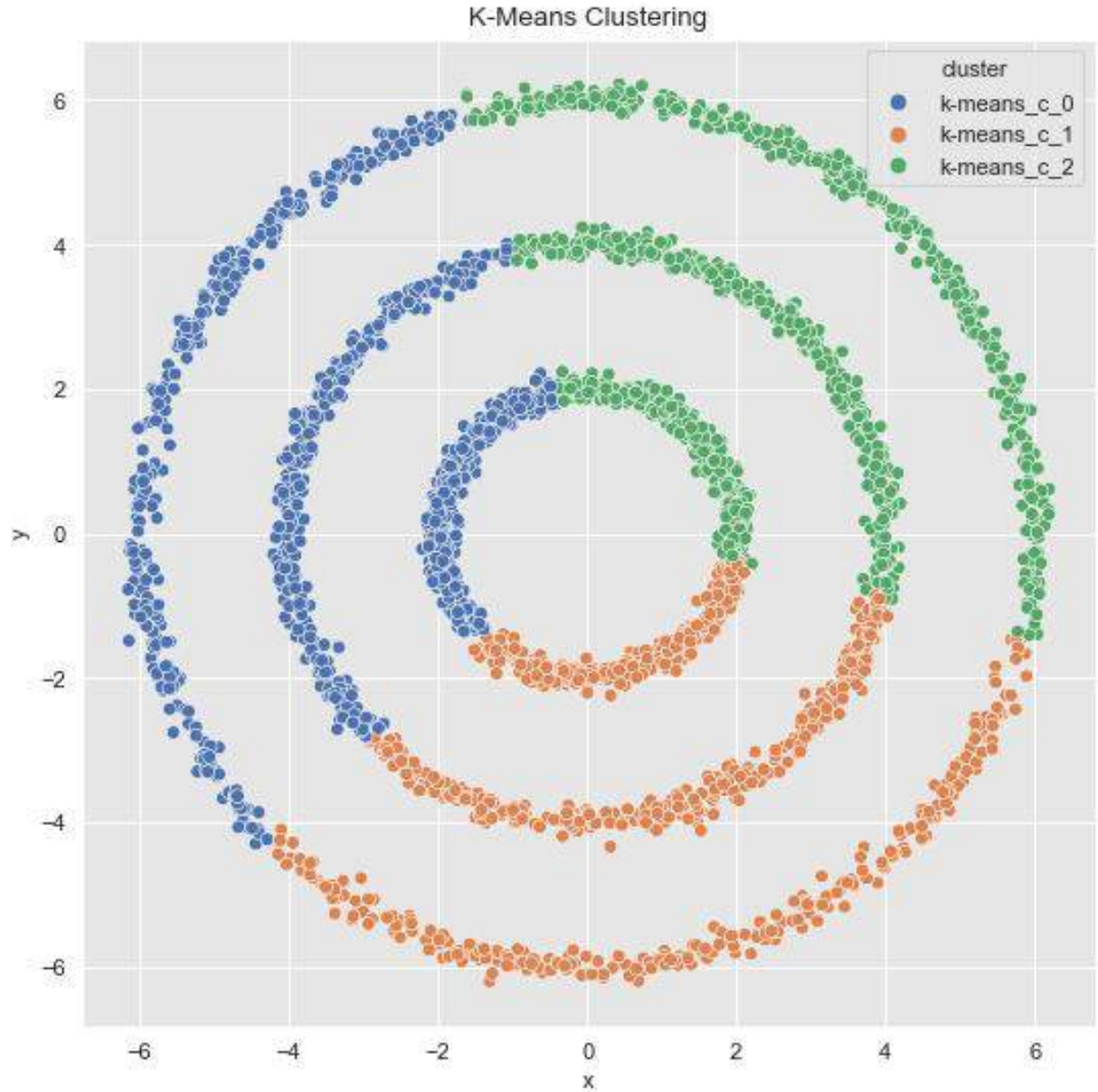
4	-0.3
5	-0.3
6	-0.5

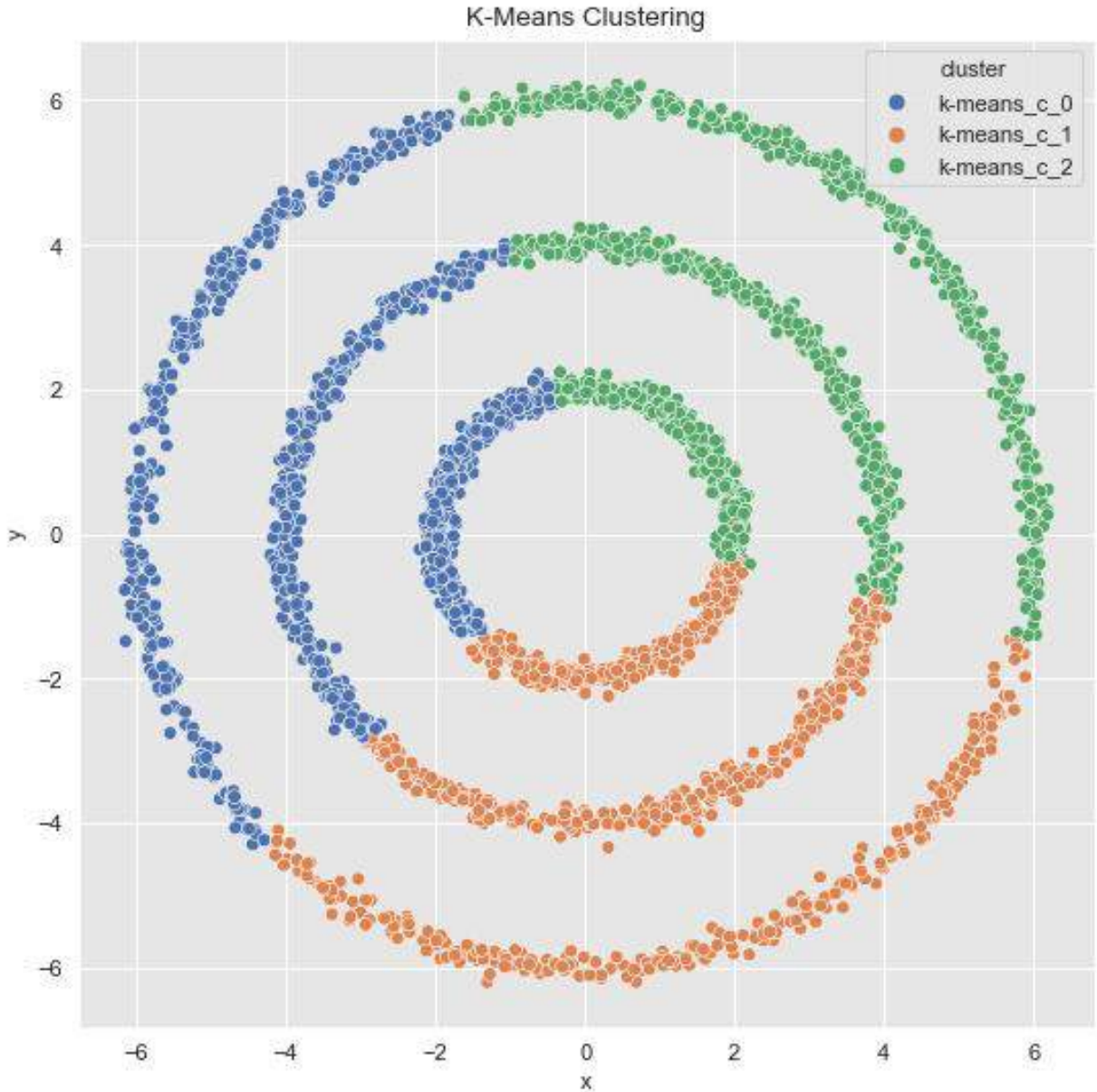


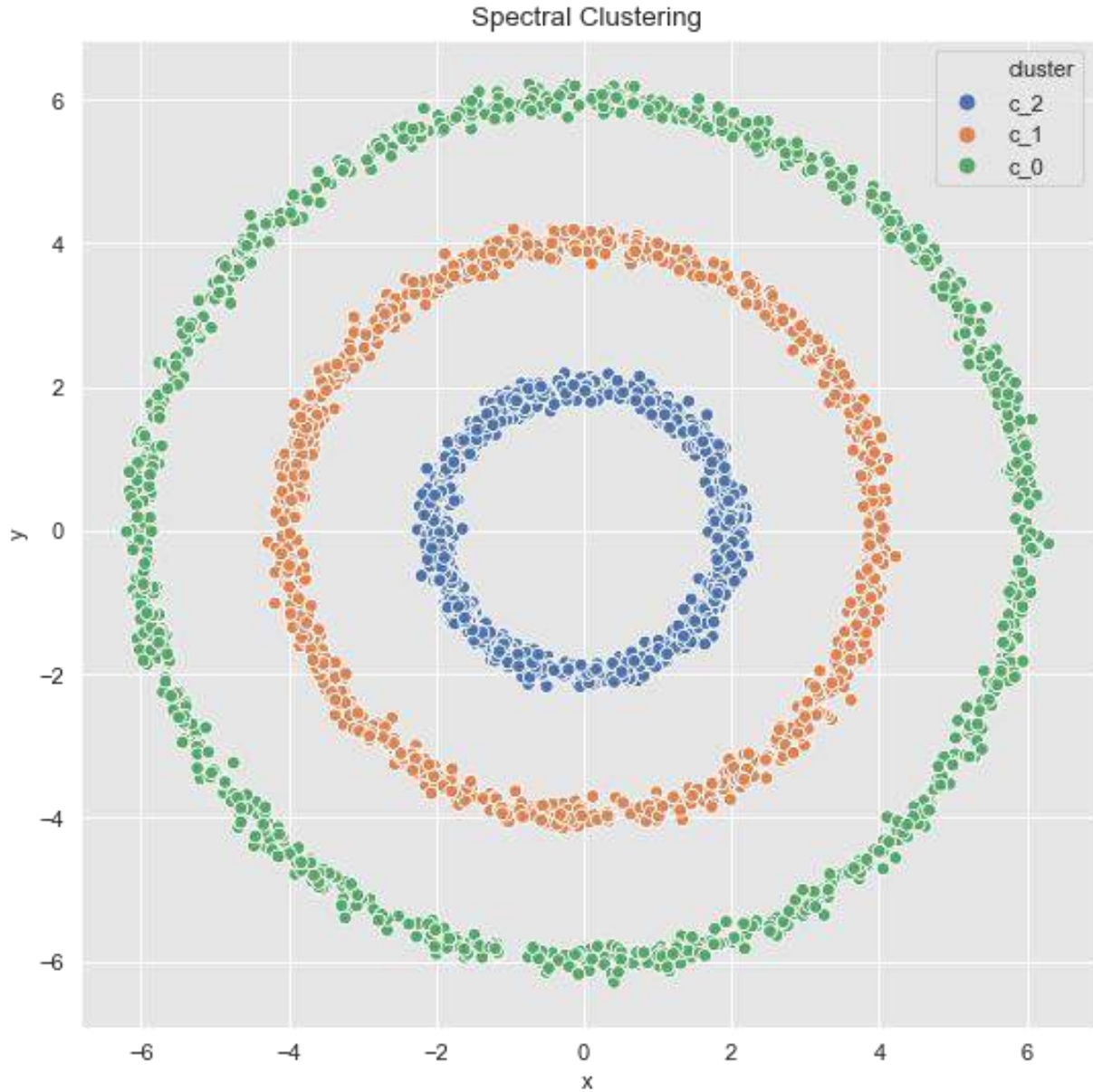

Data with ground truth labels - linear separation not possibl
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0


kernal transform to higher dimension
linear separation is possible
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0


Input Data
6
4
2
0
−2
−4
−6
y
−6
−4
−2
0
2
4
6
x


K-Means Clustering
duster
k-means_c_0
k-means_c_1
k-means_c_2
6
4
2
0
−2
−4
−6
y
−6
−4
−2
0
2
4
6
x


K-Means Clustering
cluster
k-means_c_0
k-means_c_1
k-means_c_2
x
y


Spectral Clustering
cluster
c_2
c_1
c_0
x
y

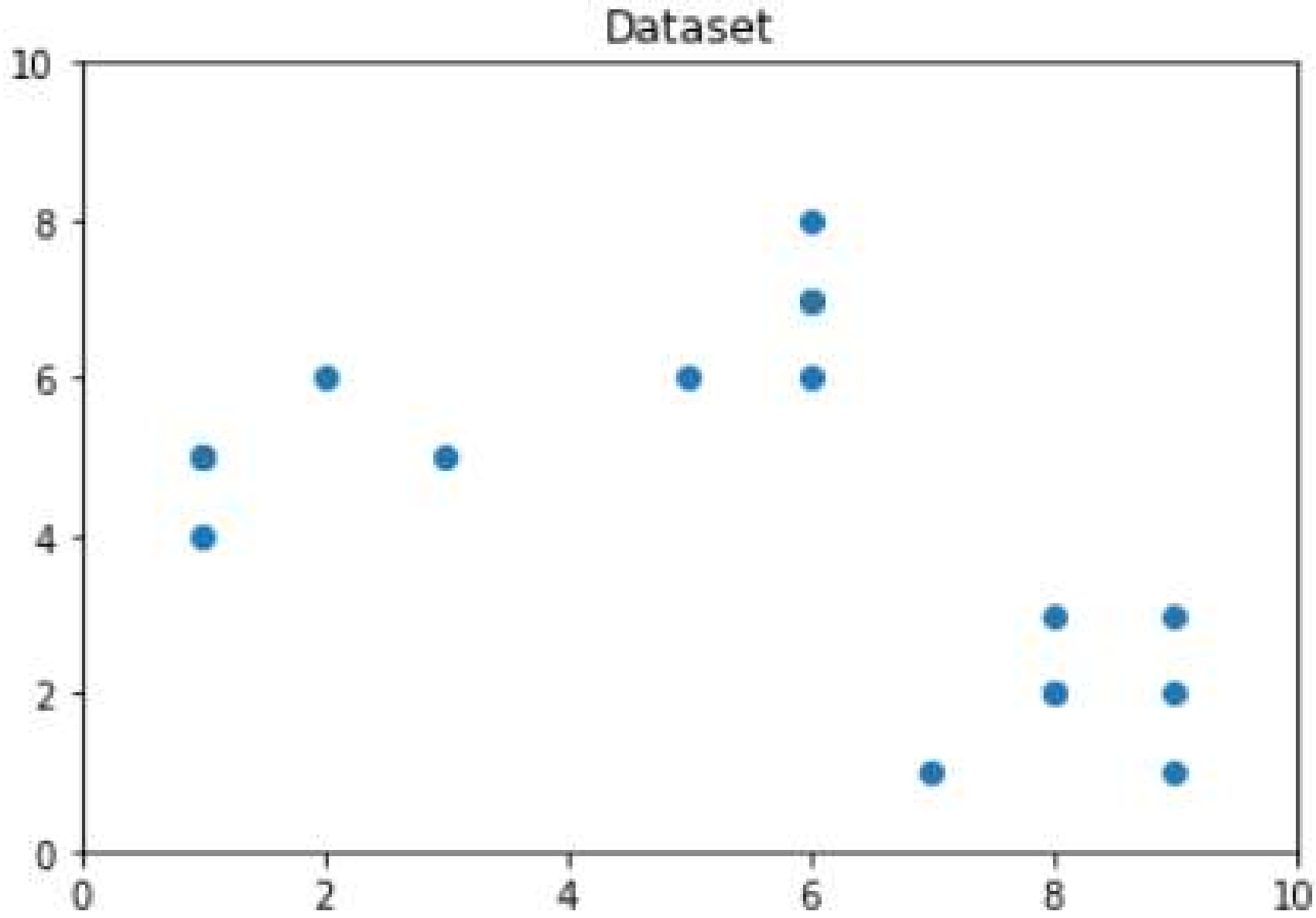
Elbow Method

- A fundamental step for any unsupervised algorithm is to determine the **optimal number** of **clusters** into which the data may be clustered.
- The **Elbow Method** is one of the most popular methods to determine this optimal value of **k**.

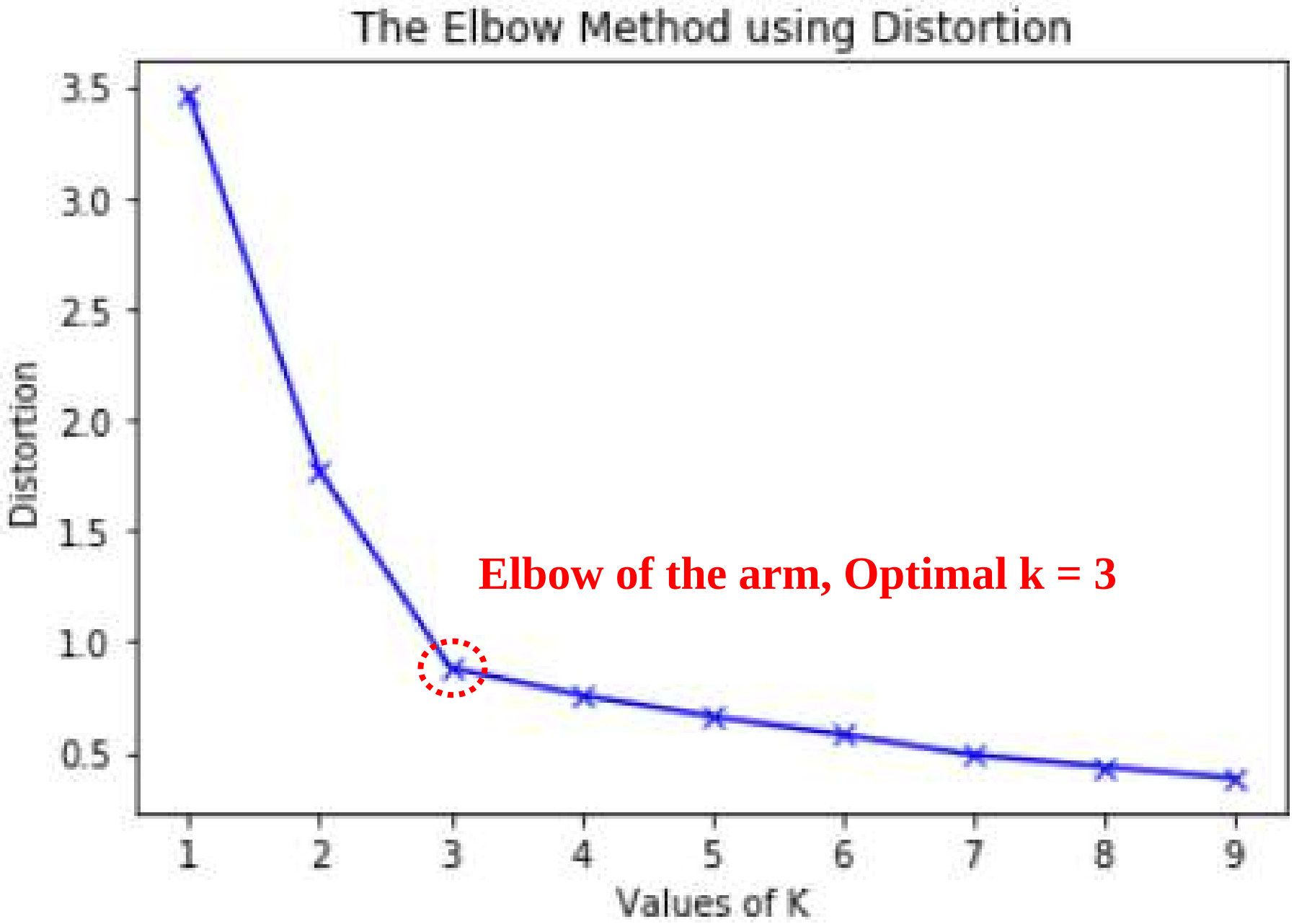
Two Concepts in Elbow Method:

- **Distortion:** It is calculated as the average of the squared distances from the cluster centers of the respective clusters (Euclidean distance metric is used).
- **Inertia:** It is the sum of squared distances of samples to their closest cluster center.

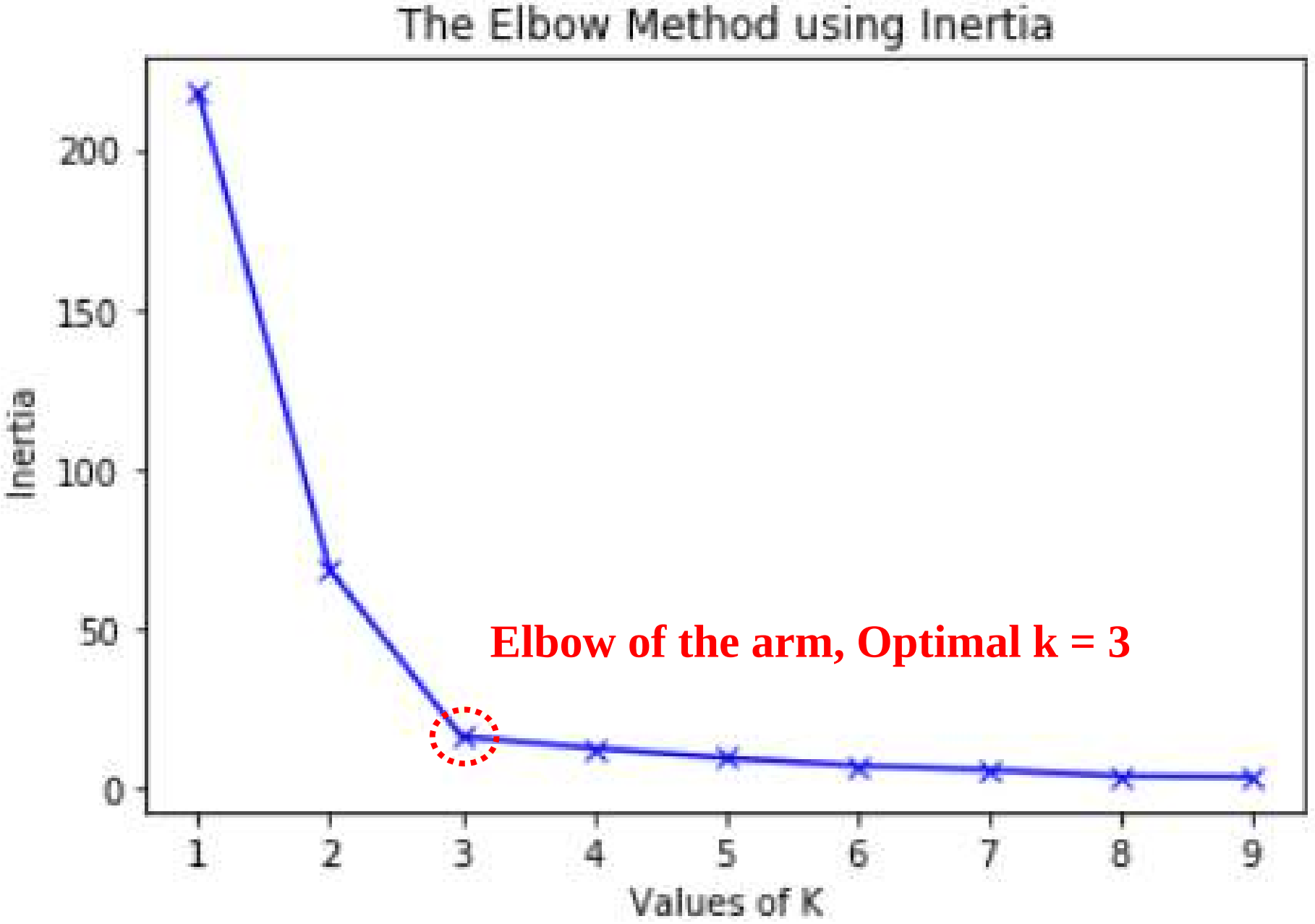
Elbow Method



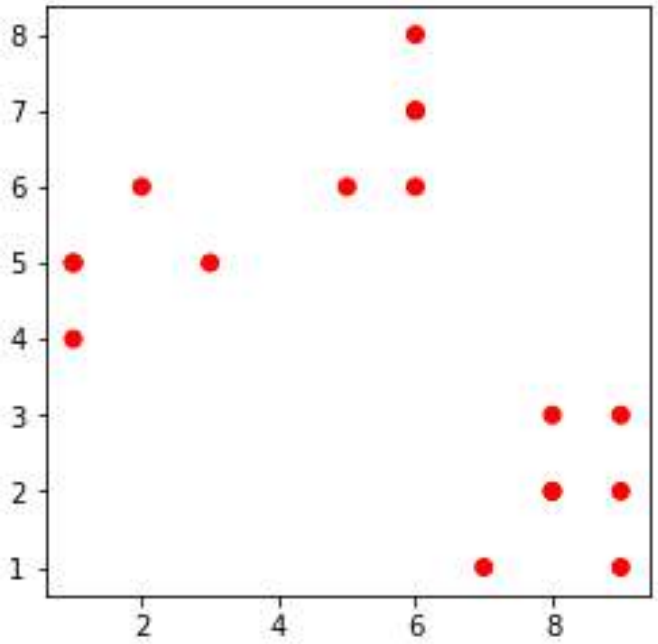
Elbow Method



Elbow Method

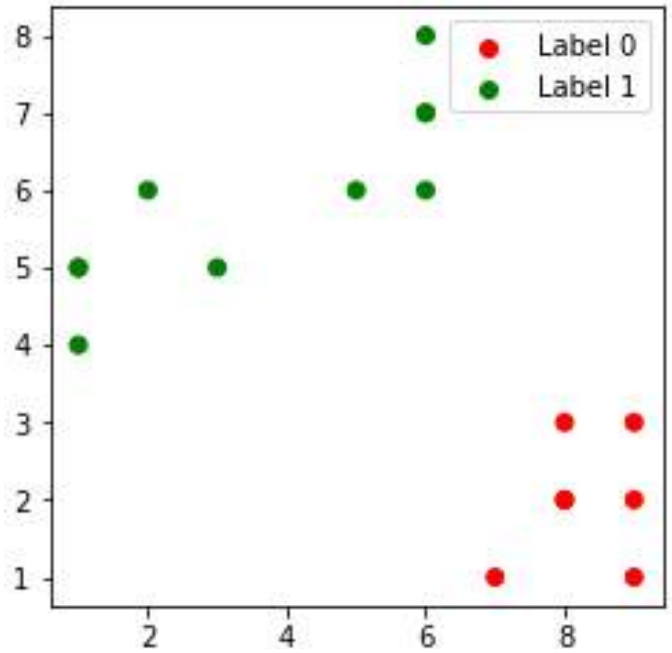


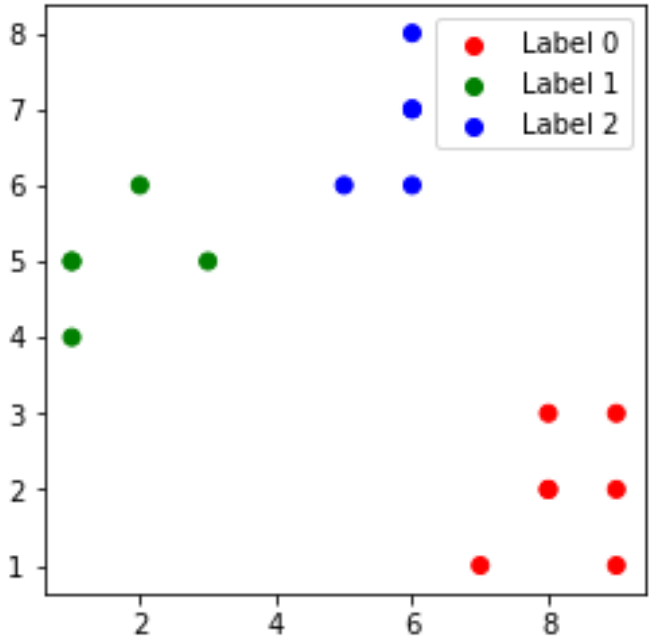
Elbow Method



K = 1

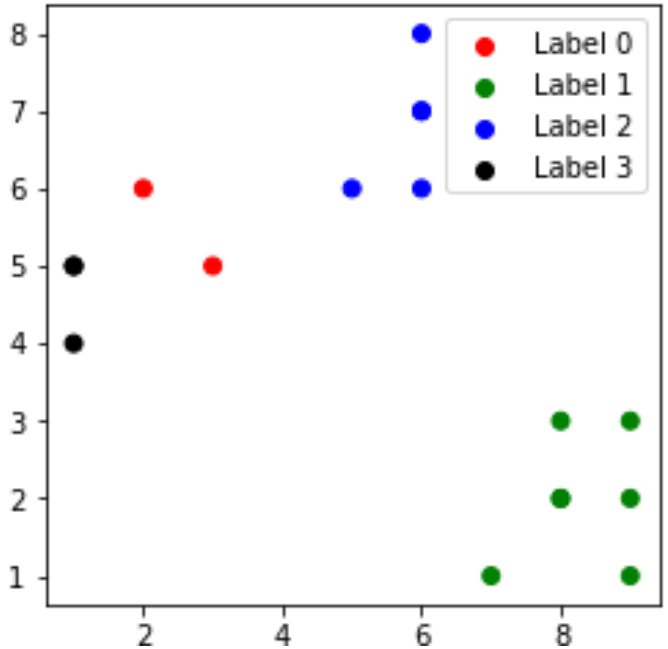
K = 2





K = 3

K = 4



Clustering Quality

- Ideal clustering is characterised by minimal intra cluster distance and maximal inter cluster distance.

There are majorly two types of measures to assess the clustering performance.

- ***Extrinsic Measures*** which require ground truth labels. Examples are Adjusted Rand index, Fowlkes-Mallows scores, Mutual information based scores, Homogeneity, Completeness and V-measure.

- ***Intrinsic Measures*** that does not require ground truth labels. Some of the clustering performance measures are Silhouette Coefficient, Calinski-Harabasz Index, Davies-Bouldin Index etc.

Assignment V

Q1. What Clustering? Give example of using clustering to solve real-life problems.

Q2. Explain working of k-means clustering algorithm.

Q3. Write short note on Hierarchical Clustering.

Q4. Explain working of Elbow method.

Check your assignment before 30/10/2023

References

Test Books

1. Bishop, Christopher M., and Nasser M. Nasrabadi, “Pattern recognition and machine learning”,Vol. 4. No. 4. New York: springer, 2006.
2. Ethem Alpaydin, “ Introduction to Machine Learning”, PHI 2nd Edition-2013

Reference Books

1. Tom Mitchell, “Machine learning”, McGraw-Hill series in Computer Science.
2. Shalev-Shwartz, Shai, and Shai Ben-David, “Understanding machine learning: From theory to algorithms”, Cambridge university press, 2014.
3. Jiawei Han, Micheline Kamber, and Jian Pie, “Data Mining: Concepts and Techniques”, Elsevier Publishers 3 Edition.
4. Hastie, Trevor, et al., “The elements of statistical learning: data mining, inference, and prediction”, Vol. 2. New York: springer, 2009.
5. McKinney, “Python for Data Analysis “,O' Reilly media.
6. Trent hauk, “Scikit-learn”, Cookbook , Packt Publishing, ISBN: 9781787286382
7. Goodfellow I.,Bengio Y. and Courville, “ A Deep Learning”, MIT Press, 2016