## Module: IV: Unsupervised Learning

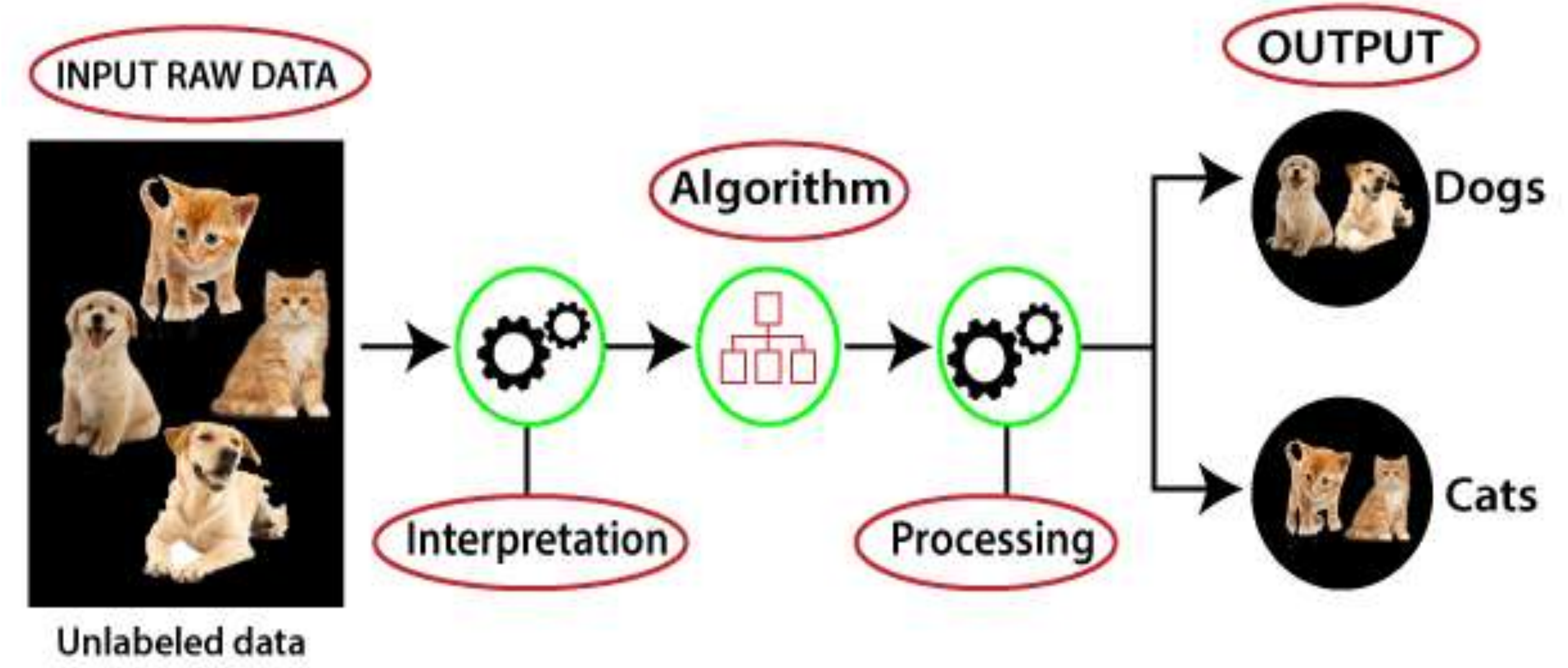
**Unsupervised Learning** – simple k Means clustering- simple and mini-batch; updating centroids incrementally; finding the optimal number of clusters using Elbow method; Silhoutte coefficient, drawbacks of K Means, k-Means++; Divisive hierarchical clustering – bisecting k-means, clustering using Minimum Spanning Tree (MST)

**Competitive Learning -** Clustering using Kohenen's Self Organising Maps (SOM), **Density Based Spatial Clustering – DBSCAN; clustering** using Gaussian Mixture Models (GMM) with EM algorithm; Outlier Detection methods – **Isolation Forest, Local Outlier Factor (LOF).**

# Introduction

- Unsupervised learning is a machine learning technique in which models are not supervised using training dataset. Instead, models itself find the hidden patterns and insights from the given data. It can be compared to learning which takes place in the human brain while learning new things.
- It can be defined as “*Unsupervised learning is a type of machine learning in which models are trained using unlabeled dataset and are allowed to act on that data without any supervision*”.

# Diagram for Unsupervised Learning



# K-Means Clustering Algorithm

- K-Means Clustering is an unsupervised learning algorithm that is used to solve the clustering problems in machine learning.
- 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.
- The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.

- The k-means clustering algorithm mainly performs two tasks:

 1) Determines the best value for K center points or centroids by an iterative process.

 2) Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.

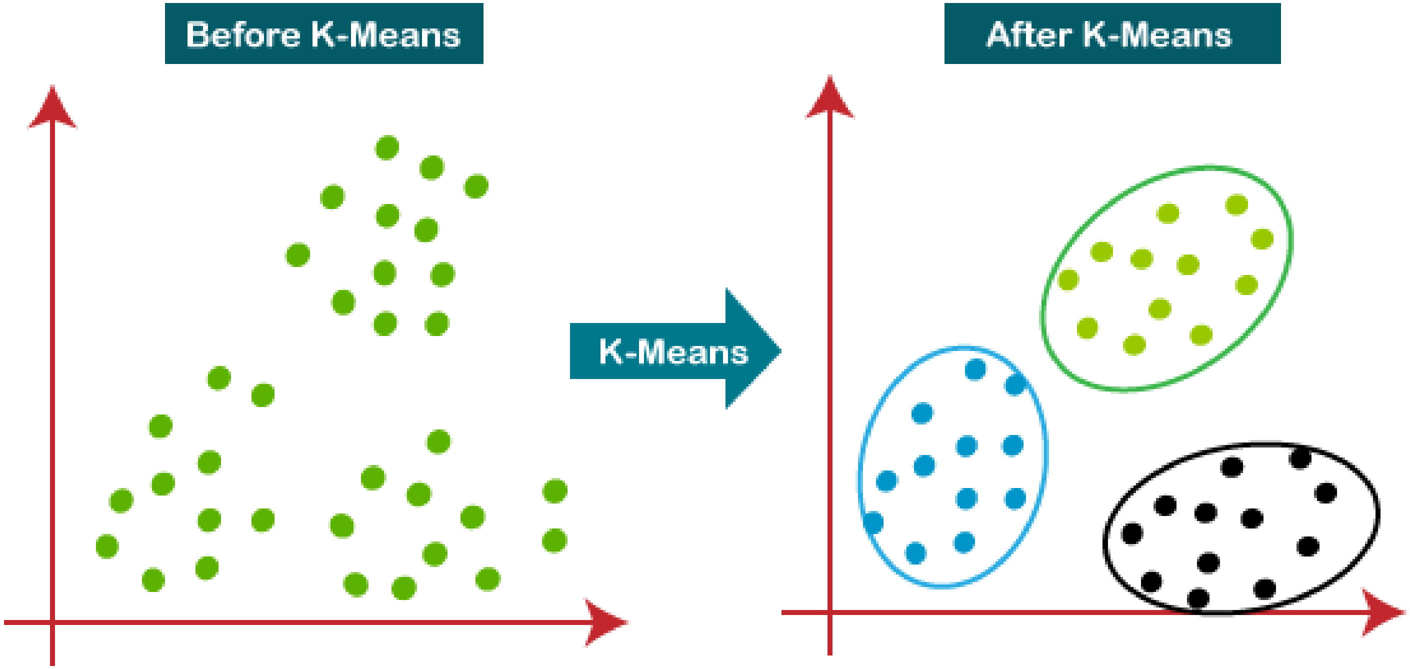


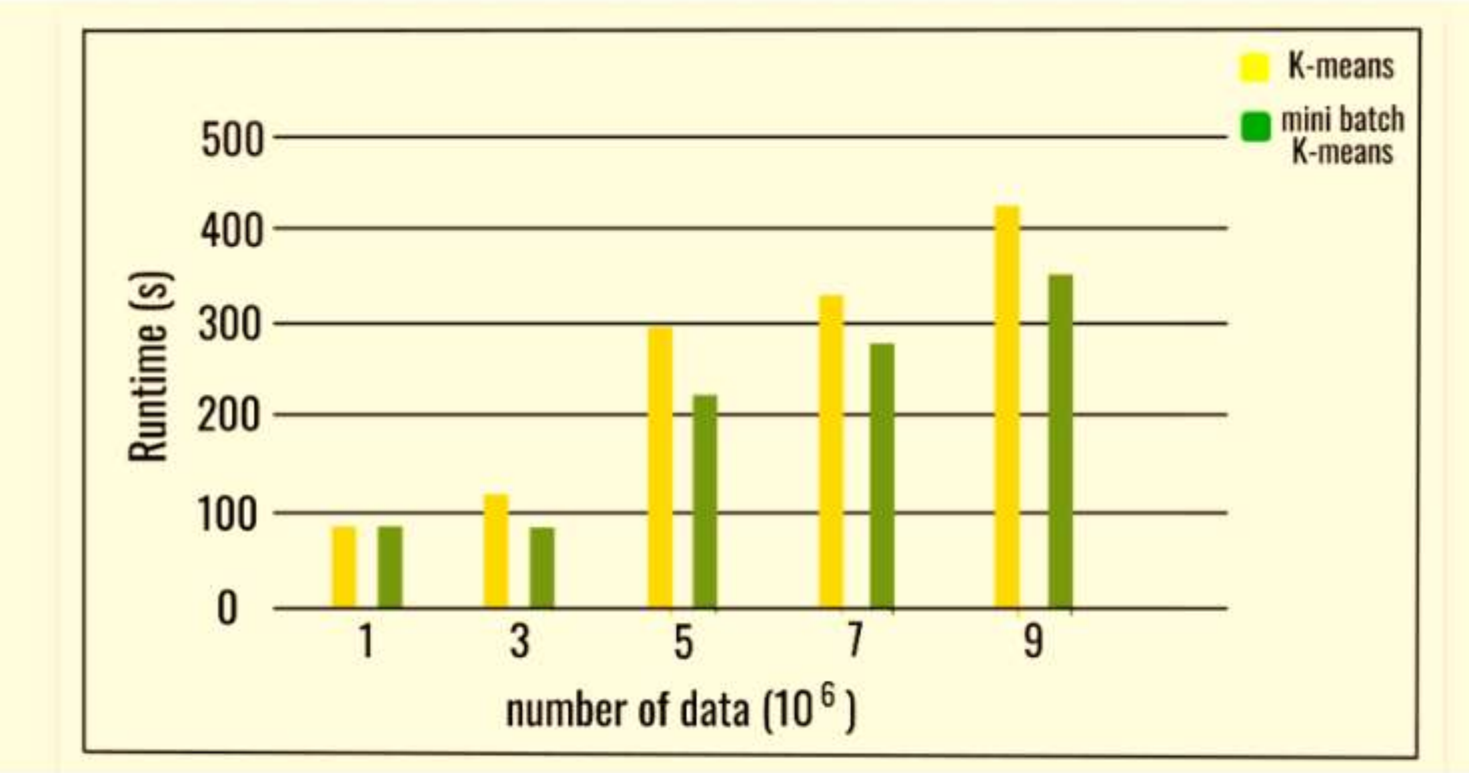
Fig:- The above diagram shows the working of the K-means Clustering Algorithm

# Mini Batch K-means clustering algorithm

- The main idea of Mini Batch K-means algorithm is to utilize small random samples of fixed in size data, which allows them to be saved in memory.
- Every time a new random sample of the dataset is taken and used to update clusters; the process is repeated until convergence.
- Each mini-batch updates the clusters with an approximate combination of the prototypes and the data results, using the learning rate, which reduces with the number of iterations.

# Conti..

- This rate of learning is the reverse of the number of data assigned to the cluster as it goes through the process.
- As the number of iterations increases, the effect of new data is reduced, so convergence can be detected when no changes in the clusters occur in several consecutive iterations.
- Each batch of data is assigned to clusters based on the prior locations of the cluster's centroids as shown in the diagram (Slide no-09)


K-means
mini batch K-means
500
400
300
200
100
0
Runtime (s)
1
3
5
7
9
number of data (10 6 )

# Working of K-Means Algorithm

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

**Step-2:** Select random K points or centroids.

**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.

- Suppose that the data mining task is to cluster points into three clusters,
- where the points are
- *A*1(2, 10), *A*2(2, 5), *A*3(8, 4), *B*1(5, 8), *B*2(7, 5), *B*3(6, 4), *C*1(1, 2), *C*2(4, 9).
- The distance function is Euclidean distance.
- Suppose initially we assign *A*1, *B*1, and *C*1 as the center of each cluster, respectively.

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 | | | | | | | | |
| A2 | 2 | 5 | | | | | | | | |
| A3 | 8 | 4 | | | | | | | | |
| B1 | 5 | 8 | | | | | | | | |
| B2 | 7 | 5 | | | | | | | | |
| B3 | 6 | 4 | | | | | | | | |
| C1 | 1 | 2 | | | | | | | | |
| C2 | 4 | 9 | | | | | | | | |

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

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 | |

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

Current Centroids:

A1: (2, 10)

B1: (6, 6)

C1: (1.5, 3.5)

<table>
<tr><th colspan="3" rowspan="2">Data Points</th><th colspan="6">Distance to</th><th rowspan="2">Cluster</th><th rowspan="2">New Cluster</th></tr>
<tr><th>2</th><th>10</th><th>6</th><th>6</th><th>1.5</th><th>1.5</th></tr>
<tr><td>A1</td><td>2</td><td>10</td><td colspan="2"></td><td colspan="2"></td><td colspan="2"></td><td>1</td><td></td></tr>
<tr><td>A2</td><td>2</td><td>5</td><td colspan="2"></td><td colspan="2"></td><td colspan="2"></td><td>3</td><td></td></tr>
<tr><td>A3</td><td>8</td><td>4</td><td colspan="2"></td><td colspan="2"></td><td colspan="2"></td><td>2</td><td></td></tr>
<tr><td>B1</td><td>5</td><td>8</td><td colspan="2"></td><td colspan="2"></td><td colspan="2"></td><td>2</td><td></td></tr>
<tr><td>B2</td><td>7</td><td>5</td><td colspan="2"></td><td colspan="2"></td><td colspan="2"></td><td>2</td><td></td></tr>
<tr><td>B3</td><td>6</td><td>4</td><td colspan="2"></td><td colspan="2"></td><td colspan="2"></td><td>2</td><td></td></tr>
<tr><td>C1</td><td>1</td><td>2</td><td colspan="2"></td><td colspan="2"></td><td colspan="2"></td><td>3</td><td></td></tr>
<tr><td>C2</td><td>4</td><td>9</td><td colspan="2"></td><td colspan="2"></td><td colspan="2"></td><td>2</td><td></td></tr>
</table>

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 | |

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

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 |

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

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 |

New Centroids:

A1: (3, 9.5) ✓

B1: (6.5, 5.25) ✓

C1: (1.5, 3.5) ✓

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

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

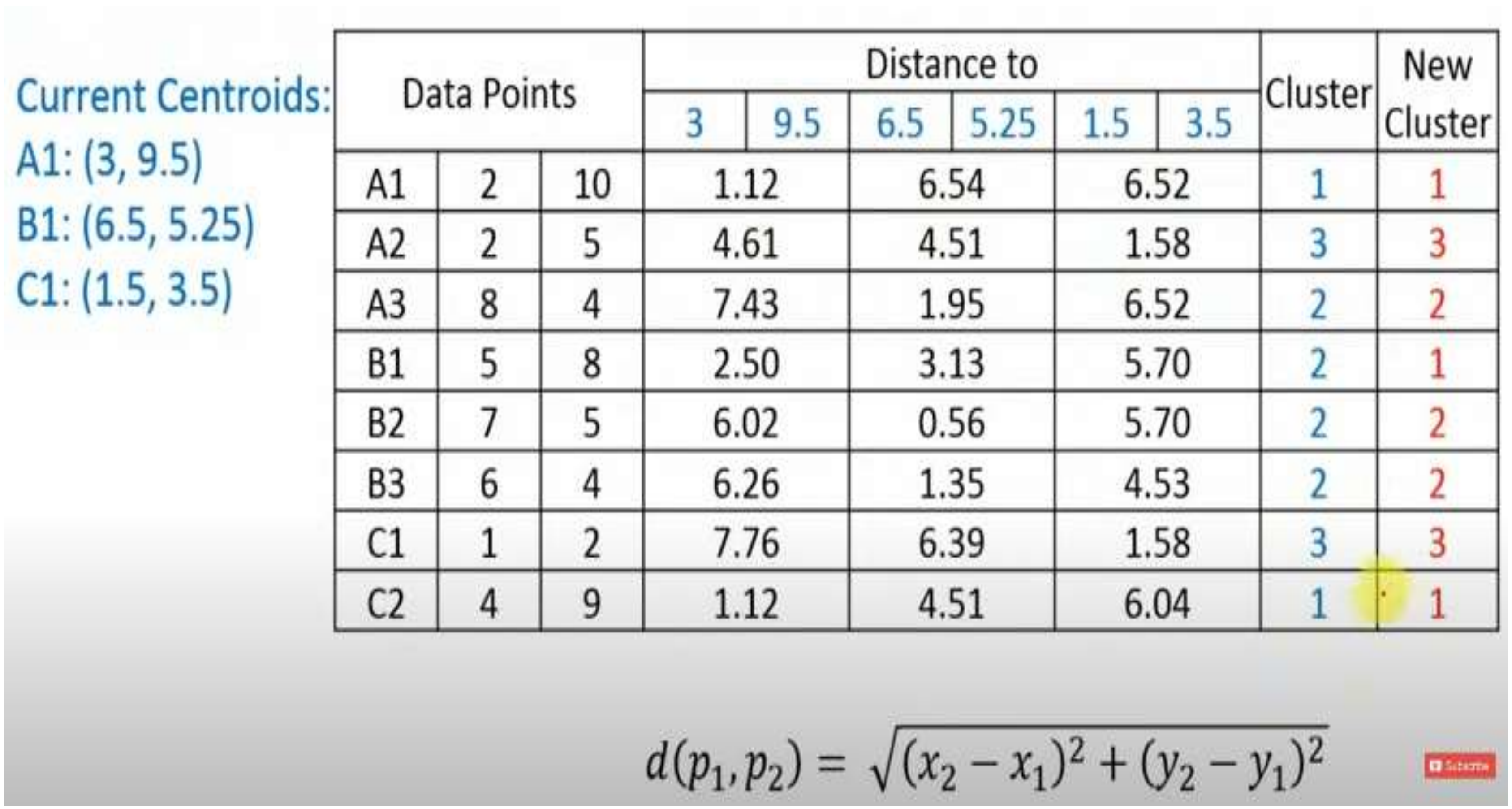
| Data Points | | | Distance to | | | | | | Cluster | New Cluster |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | 3 | 9.5 | 6.5 | 5.25 | 1.5 | 3.5 | | |
| A1 | 2 | 10 | | | | | | | 1 | |
| A2 | 2 | 5 | | | | | | | 3 | |
| A3 | 8 | 4 | | | | | | | 2 | |
| B1 | 5 | 8 | | | | | | | 2 | |
| B2 | 7 | 5 | | | | | | | 2 | |
| B3 | 6 | 4 | | | | | | | 2 | |
| C1 | 1 | 2 | | | | | | | 3 | |
| C2 | 4 | 9 | | | | | | | 1 | |

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

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 | |
| A2 | 2 | 5 | 4.61 | | 4.51 | | 1.58 | | 3 | |
| A3 | 8 | 4 | 7.43 | | 1.95 | | 6.52 | | 2 | |
| B1 | 5 | 8 | 2.50 | | 3.13 | | 5.70 | | 2 | |
| B2 | 7 | 5 | 6.02 | | 0.56 | | 5.70 | | 2 | |
| B3 | 6 | 4 | 6.26 | | 1.35 | | 4.53 | | 2 | |
| C1 | 1 | 2 | 7.76 | | 6.39 | | 1.58 | | 3 | |
| C2 | 4 | 9 | 1.12 | | 4.51 | | 6.04 | | 1 | |

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



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 |

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

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 |

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

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 |

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

# To update the centroids incrementally, we can use the following steps:

**Step1:-** Initialize the centroids: Randomly select K data points as the initial centroids.

**Step2:-** Assign data points to centroids: For each data point, compute its distance to each centroid and assign it to the nearest centroid.

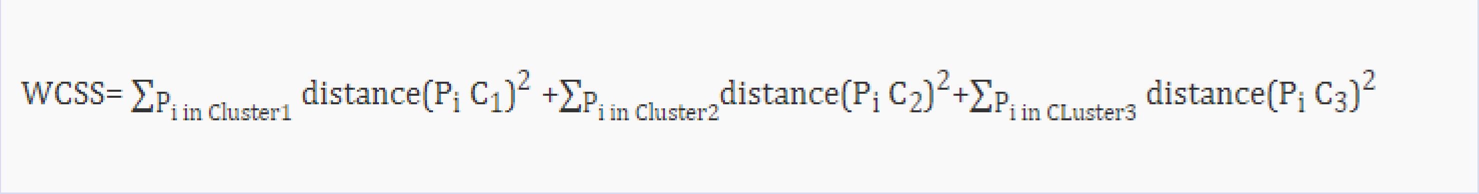
**Step3:-** Update the centroids: For each centroid, compute the mean of the assigned data points and use it as the new centroid.

**Step4:-** Incremental update: When a new data point is added to the dataset, we can update the centroids incrementally by only computing the mean of the new data point and the current centroid. This avoids the need to recompute the mean of all the data points assigned to the centroid.

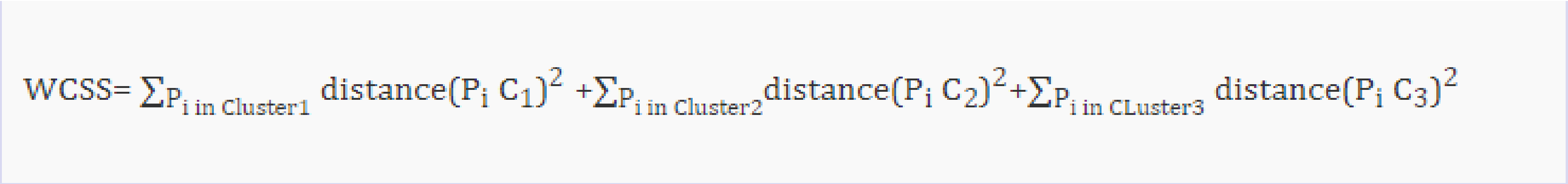
**Step5:-** Repeat steps 2-4 until the centroids no longer move significantly or a maximum number of iterations is reached.

# Elbow Method in K-Means Clustering

- The Elbow method is one of the most popular ways to find the optimal number of clusters.
- This method uses the concept of WCSS value. **WCSS** stands for **Within Cluster Sum of Squares**, which defines the total variations within a cluster.
- The formula to calculate the value of WCSS (for 3 clusters) is given below:



$$\text{WCSS}= \sum P_{\text{i in Cluster1}} \text{ distance}(P_i\ C_1)^2 + \sum P_{\text{i in Cluster2}} \text{distance}(P_i\ C_2)^2 + \sum P_{\text{i in CLuster3}} \text{ distance}(P_i\ C_3)^2$$



$$WCSS= \sum_{P_{i \text{ in Cluster1}}} \text{distance}(P_i\ C_1)^2 + \sum_{P_{i \text{ in Cluster2}}} \text{distance}(P_i\ C_2)^2 + \sum_{P_{i \text{ in CLuster3}}} \text{distance}(P_i\ C_3)^2$$

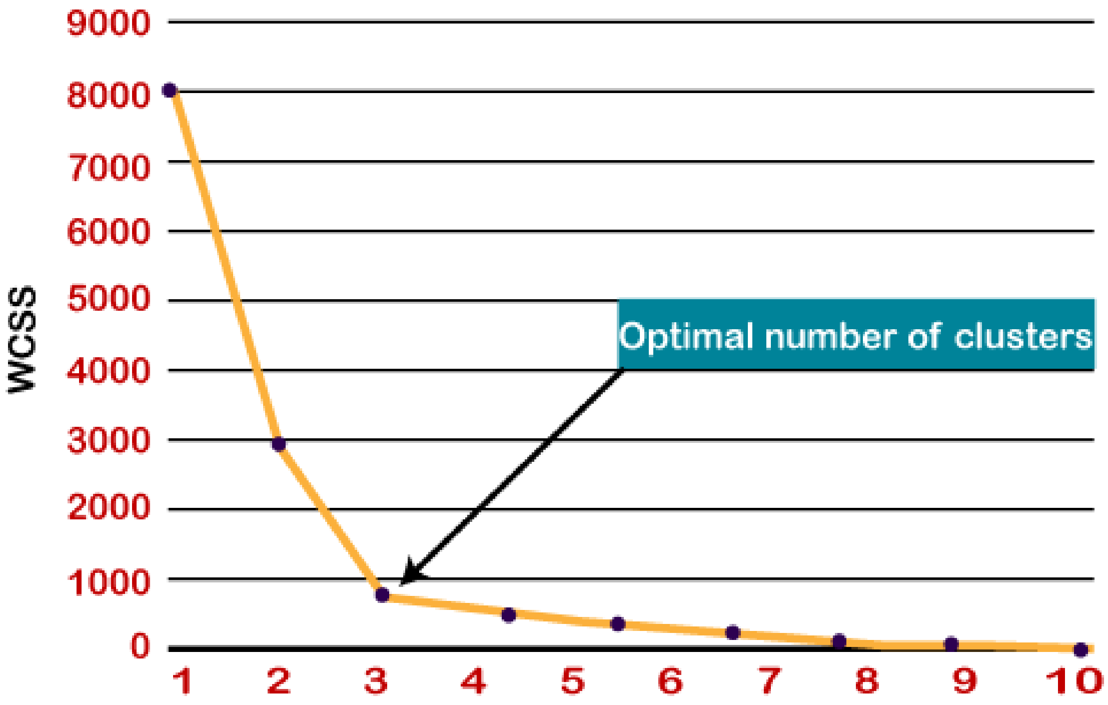
In the above formula of WCSS,

- $\sum_{\text{Pi in Cluster1}}$ distance$(P_i\ C_1)^2$: It is the sum of the square of the distances between each data point and its centroid within a cluster1 and the same for the other two terms.
- To measure the distance between data points and centroid, we can use any method such as Euclidean distance or Manhattan distance.

# Finding the optimal number of clusters using Elbow method

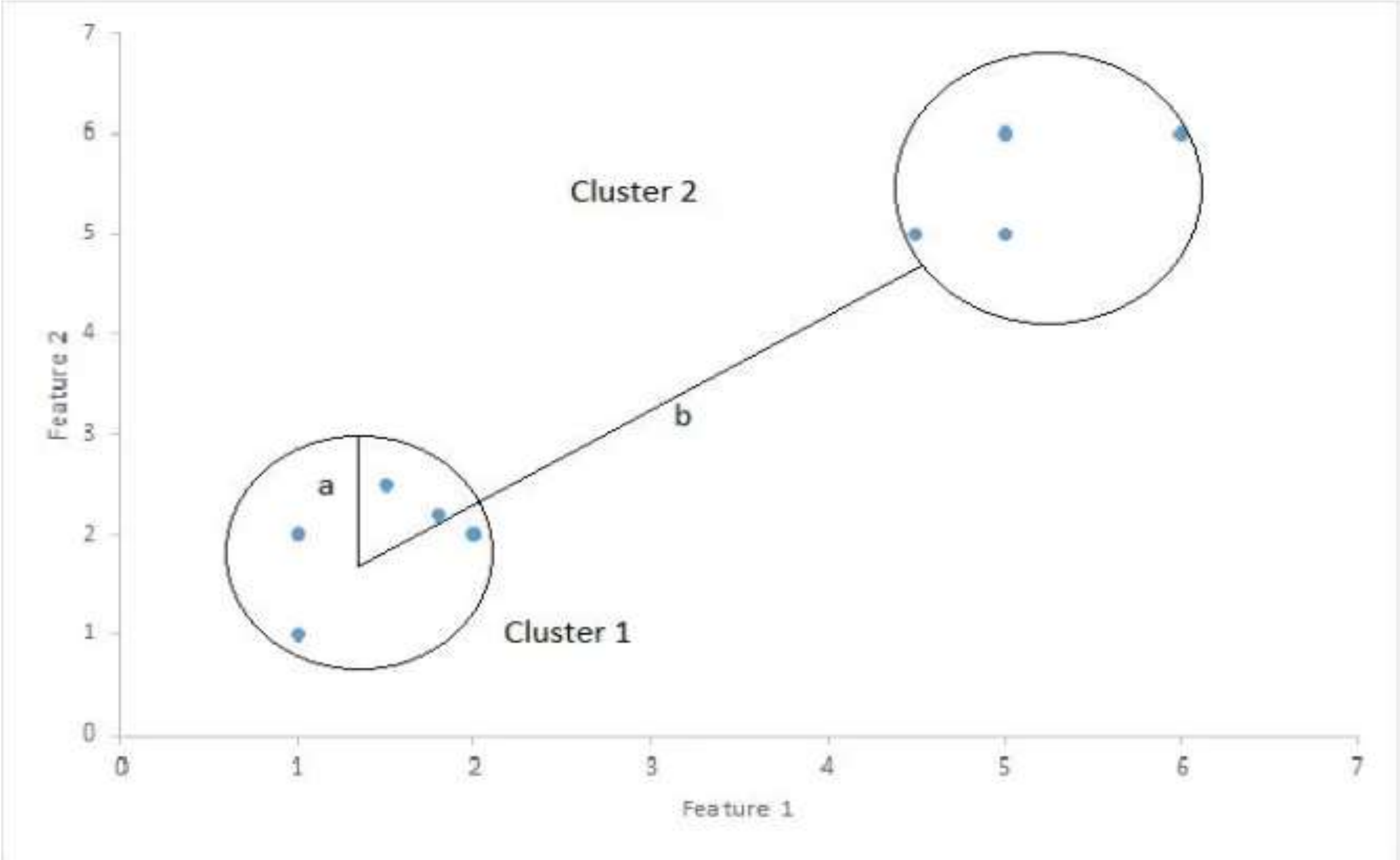
- It executes the K-means clustering on a given dataset for different K values (ranges from 1-10).
- For each value of K, calculates the WCSS value.
- Plots a curve between calculated WCSS values and the number of clusters K.
- The sharp point of bend or a point of the plot looks like an arm, then that point is considered as the best value of K.

Since the graph shows the sharp bend, which looks like an elbow, hence it is known as the elbow method. The graph for the elbow method looks like the below image:



# Silhoutte coefficient

- ✓Silhouette Coefficient or silhouette score is a metric used to calculate the goodness of a clustering technique. Its value ranges from -1 to 1.
- ✓1: Means clusters are well apart from each other and clearly distinguished.
- ✓0: Means clusters are indifferent, or we can say that the distance between clusters is not significant.
- ✓-1: Means clusters are assigned in the wrong way.


Cluster 2
b
a
Cluster 1
Feature 2
Feature 1
0
1
2
3
4
5
6
7
0
1
2
3
4
5
6
7

# Conti..

- Silhouette Score = (b-a)/max(a,b)
- where
- a= average intra-cluster distance i.e the average distance between each point within a cluster.
- b= average inter-cluster distance i.e the average distance between all clusters.

# Drawbacks of K Means

K-Means Clustering Algorithm has the following disadvantages-

- It requires to specify the number of clusters (k) in advance.
- It can not handle noisy data and outliers.
- It is not suitable to identify clusters with non-convex shapes.

# K-Means++

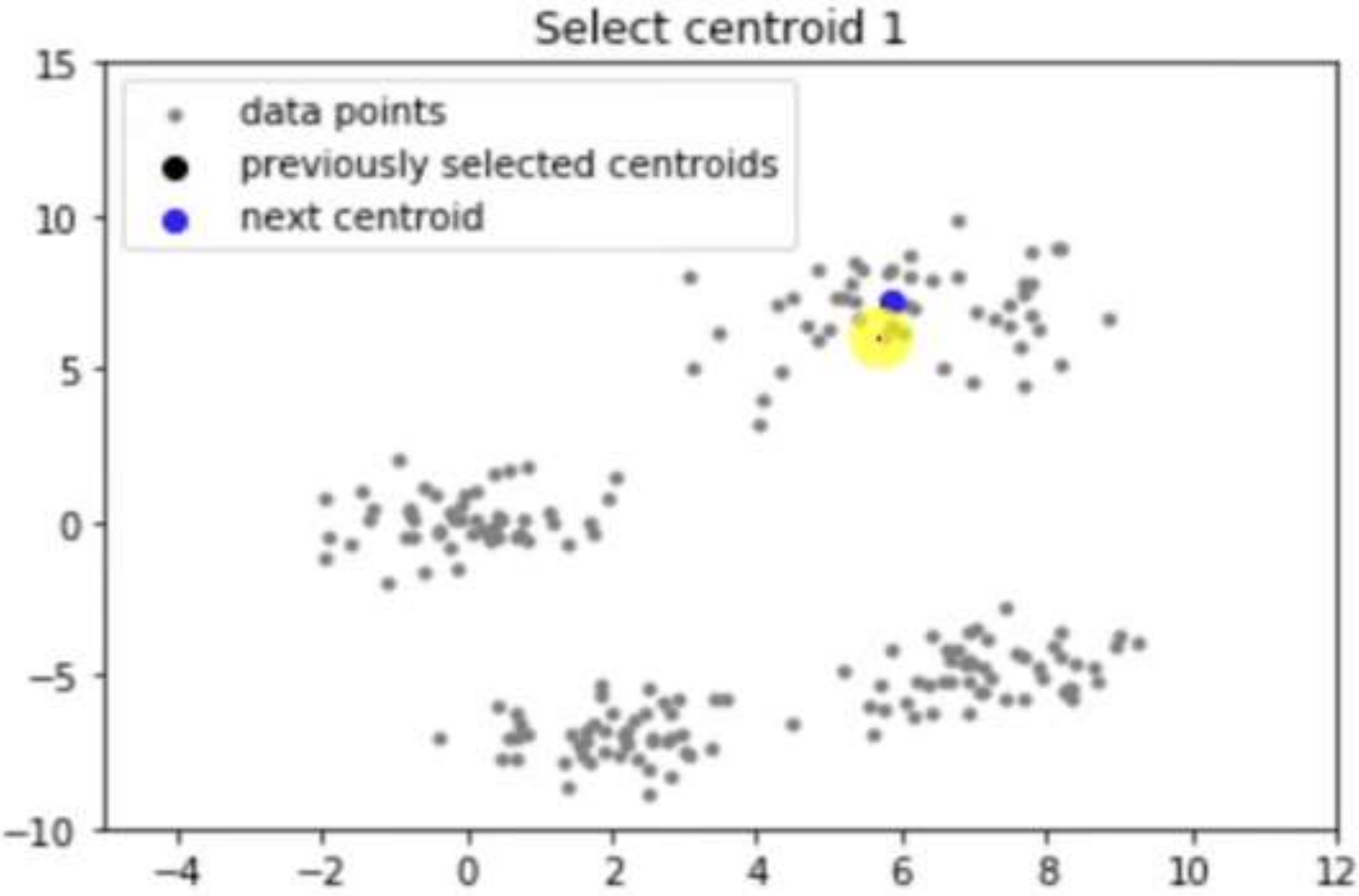
- K-Means++ is an improvement over the original K-Means algorithm that addresses the sensitivity to the initial centroids problem.
- Instead of randomly selecting the initial centroids, K-Means++ uses a smarter initialization strategy that selects the initial centroids with a higher probability of being far from each other.
- This helps to improve the quality of the clustering solution and reduce the number of iterations needed to converge.

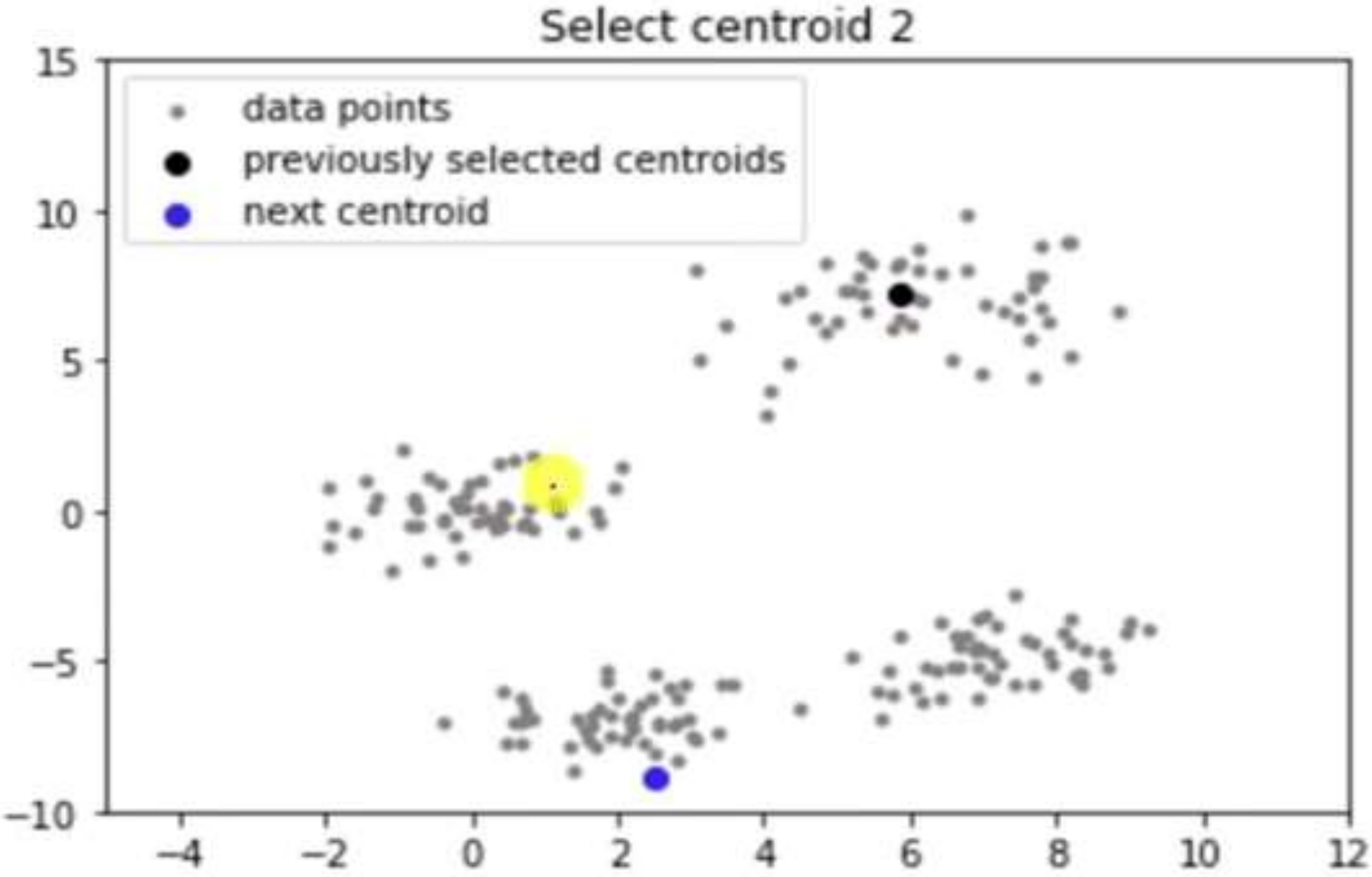
# K-Means++ Clustering – Steps

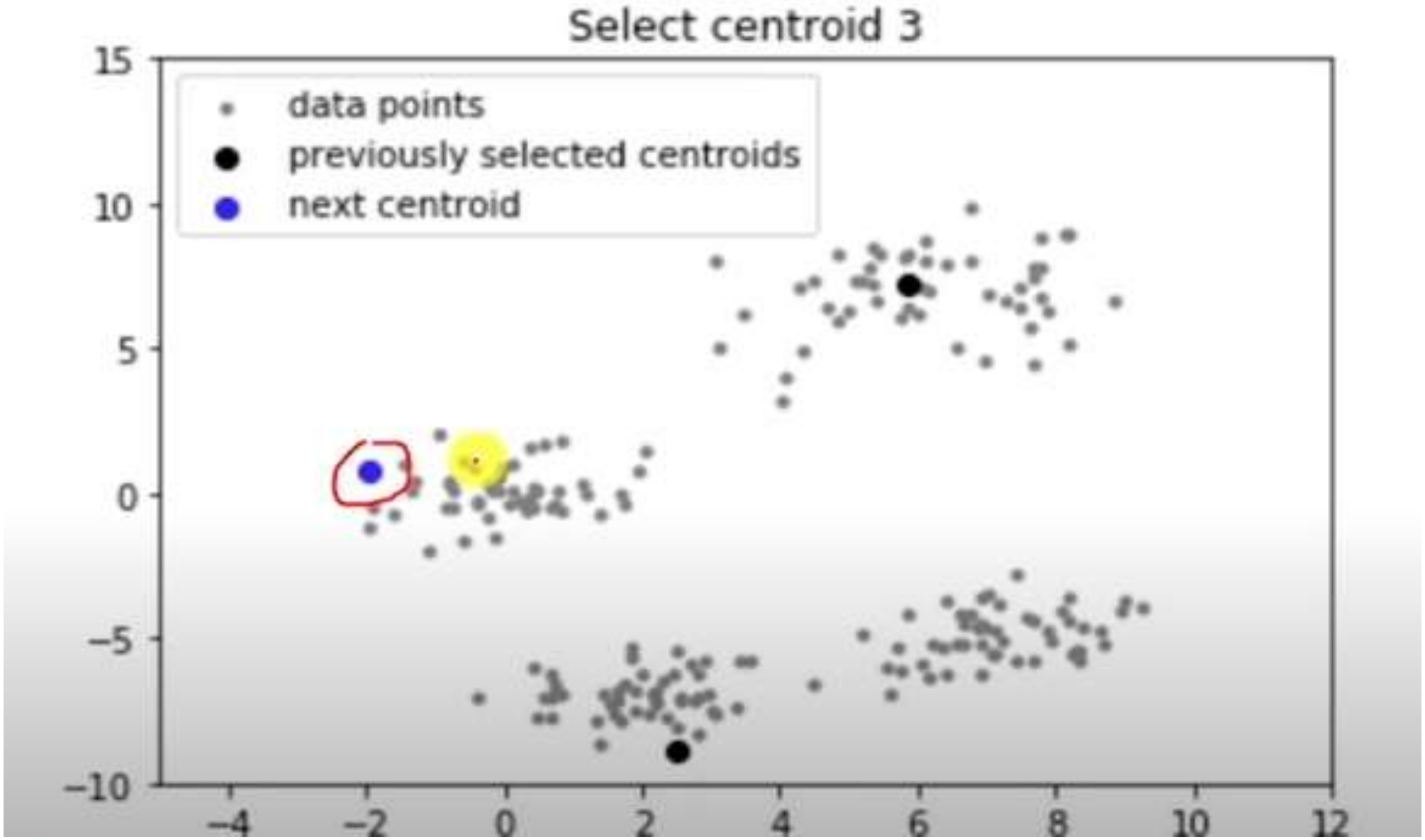
**Step 1:** Randomly select the first centroid from the data points.

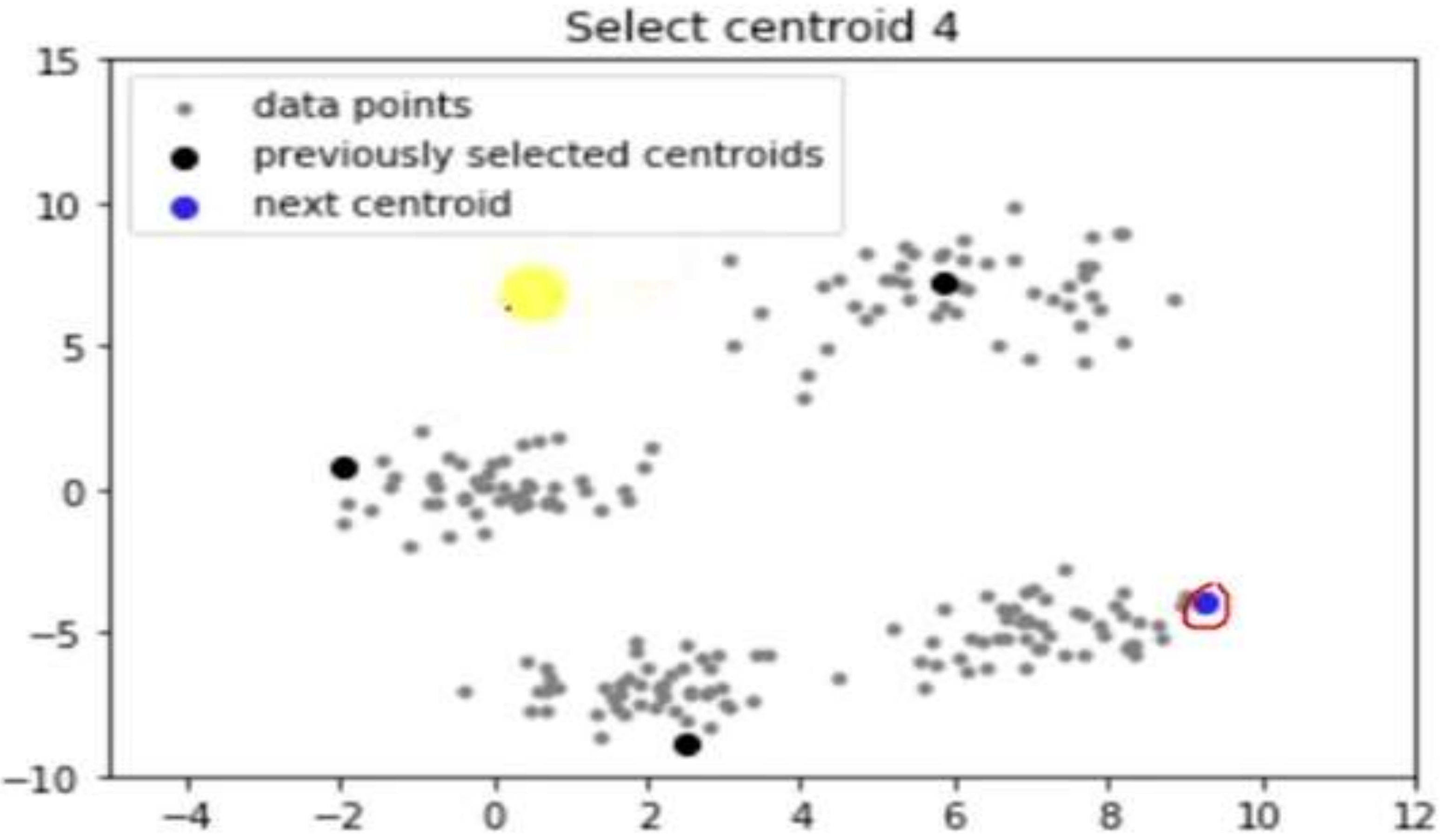
**Step 2:** For each data point compute its distance from the nearest, previously chosen centroid.

**Step 3:** Select the next centroid from the data points such that the probability of choosing a point as centroid is directly proportional to its distance from the nearest, previously chosen centroid. (i.e. the point having maximum distance from the nearest centroid is most likely to be selected next as a centroid)


Select centroid 1
data points
previously selected centroids
next centroid
15
10
5
0
−5
−10
−4
−2
0
2
4
6
8
10
12

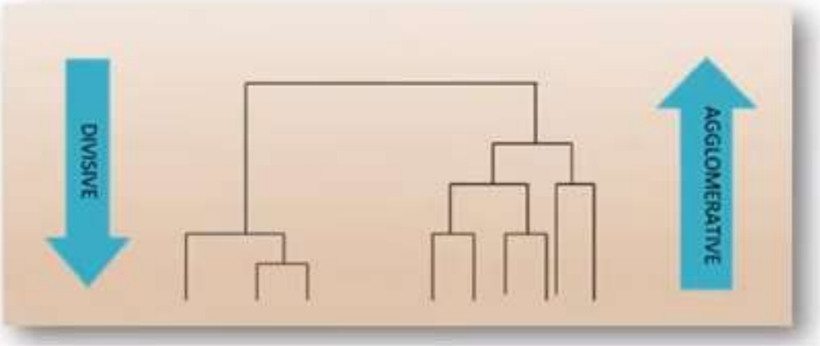

Select centroid 2
data points
previously selected centroids
next centroid
15
10
5
0
−5
−10
−4
−2
0
2
4
6
8
10
12

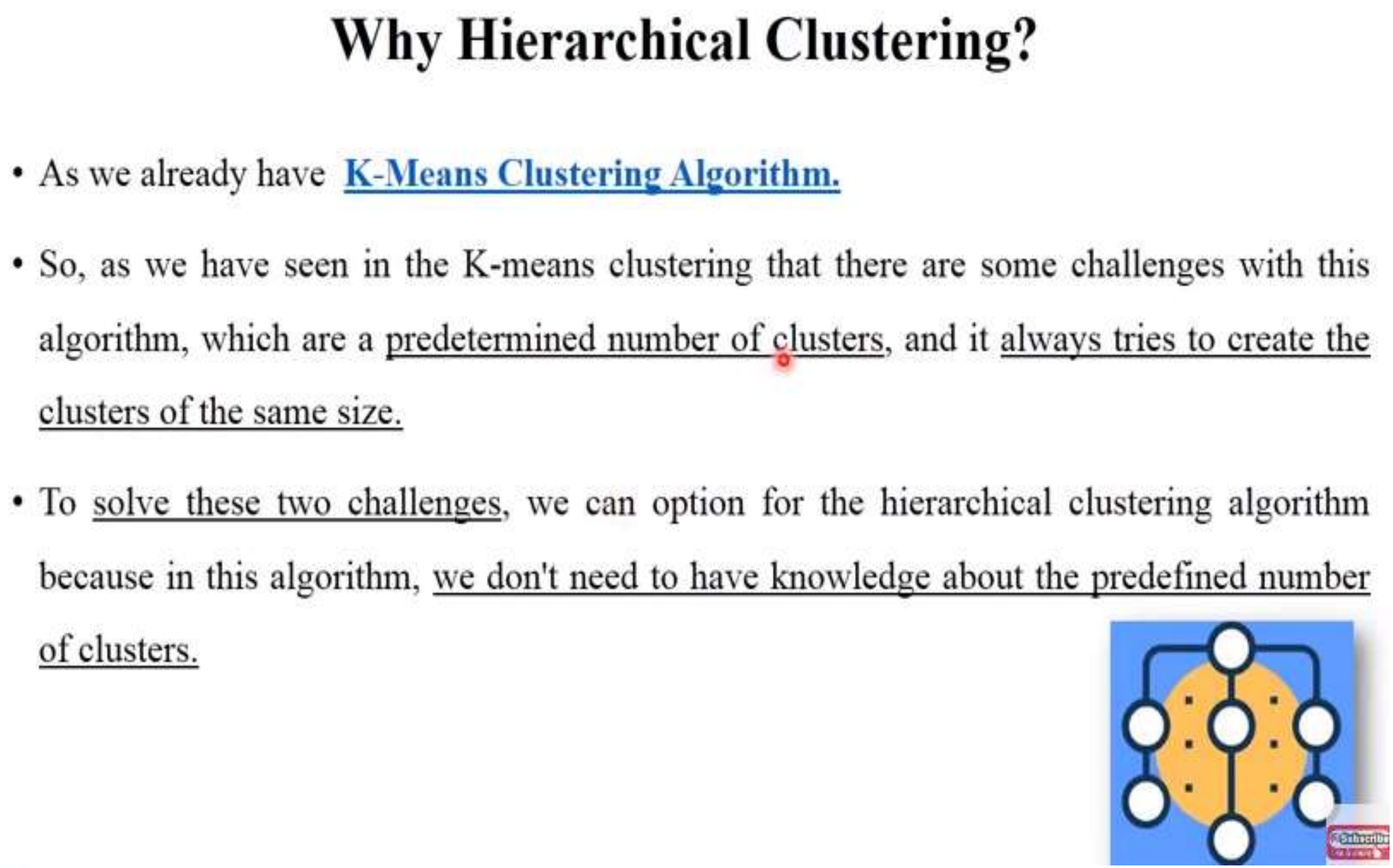

Select centroid 3
data points
previously selected centroids
next centroid
15
10
5
0
−5
−10
−4
−2
0
2
4
6
8
10
12


Select centroid 4
data points
previously selected centroids
next centroid
15
10
5
0
−5
−10
−4
−2
0
2
4
6
8
10
12

# 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





# Why Hierarchical Clustering?

- As we already have **K-Means Clustering Algorithm.**
- So, as we have seen in the K-means clustering that there are some challenges with this algorithm, which are a predetermined number of clusters, and it always tries to create the clusters of the same size.
- To solve these two challenges, we can option for the hierarchical clustering algorithm because in this algorithm, we don't need to have knowledge about the predefined number of clusters.

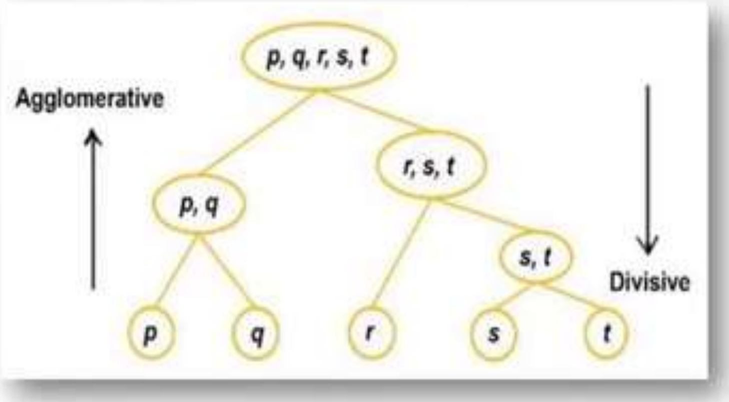
# Types of Hierarchical Clustering

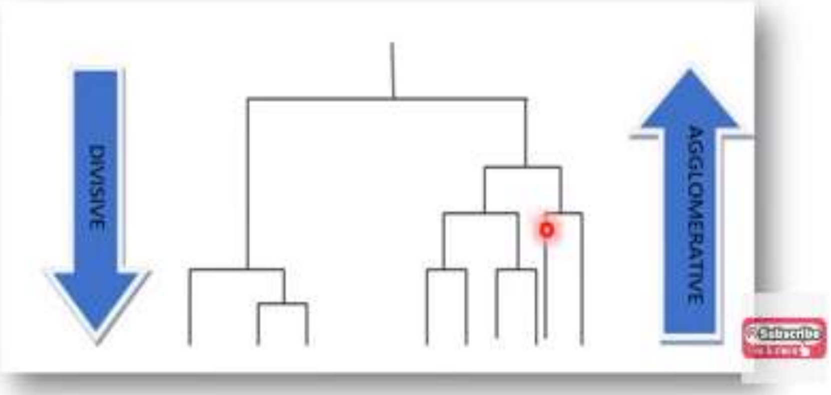
**Type 1: Agglomerative Clustering:**

- 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.

**Type 2: Divisive Clustering:**

- Divisive algorithm is the reverse of the agglomerative algorithm as it is a **top-down approach.**





# Divisive Hierarchical Clustering

- Divisive hierarchical clustering is a type of hierarchical clustering algorithm that works by recursively dividing a dataset into smaller and smaller subsets until each subset contains only one data point.
- Two commonly used methods for performing divisive hierarchical clustering are bisecting K-Means and clustering using Minimum Spanning Tree (MST).

# Bisecting K-Means Algorithm

Bisecting K-Means Algorithm is a modification of the K-Means algorithm. It is a hybrid approach between partitional and hierarchical clustering. It can recognize clusters of any shape and size. This algorithm is convenient because:

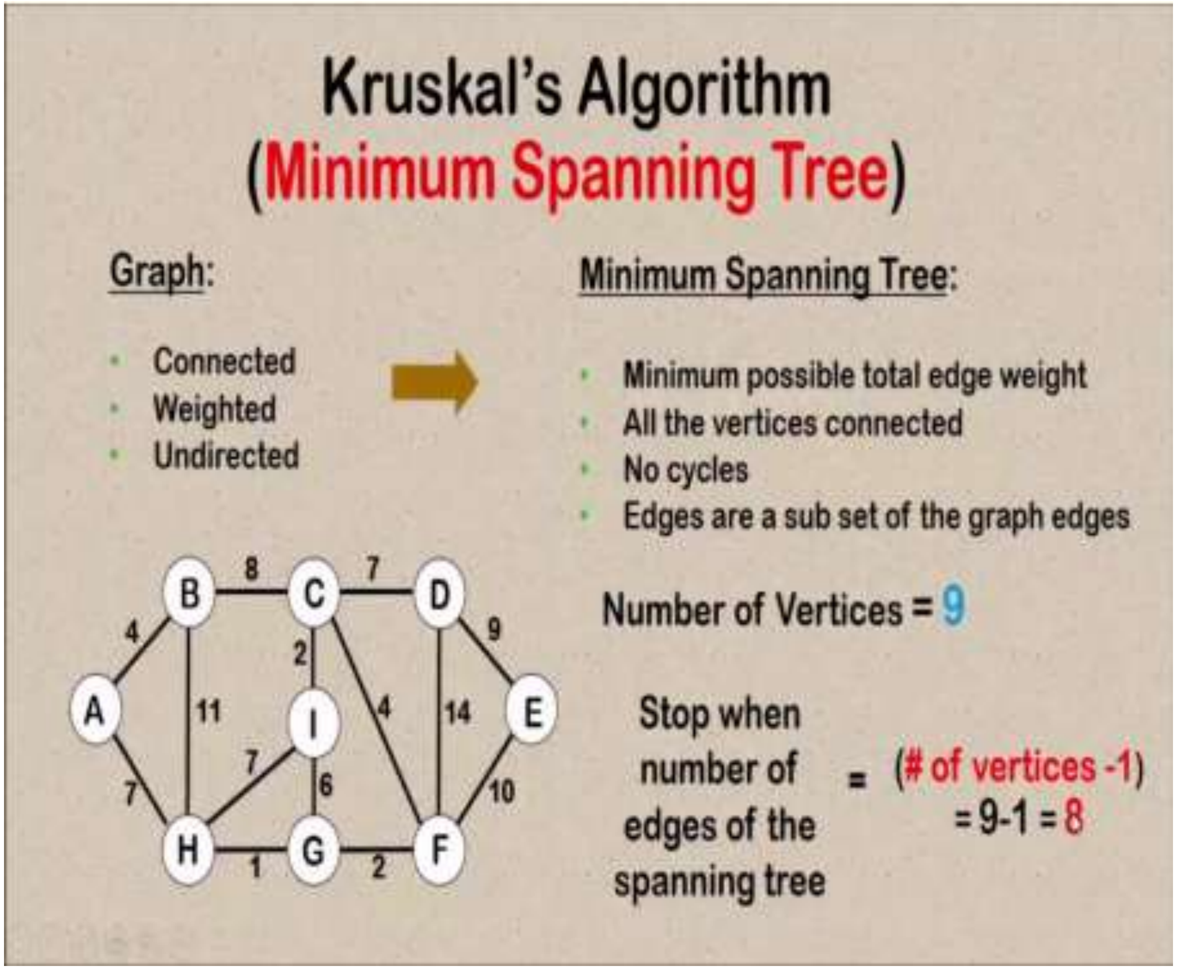
- It beats K-Means in entropy measurement.
- When K is big, bisecting k-means is more effective. Every data point in the data collection and k centroids are used in the K-means method for computation.

# Conti..

- On the other hand, only the data points from one cluster and two centroids are used in each Bisecting stage of Bisecting k-means.
- While k-means is known to yield clusters of varied sizes, bisecting k-means results in clusters of comparable sizes.

# Clustering using Minimum Spanning Tree (MST):

- Clustering using Minimum Spanning Tree (MST) is a type of divisive hierarchical clustering that works by constructing a Minimum Spanning Tree of the dataset and recursively dividing it into smaller subtrees.
- The algorithm starts by constructing a Minimum Spanning Tree of the dataset, which is a tree that connects all the data points with the minimum total edge weight.
- The tree is then recursively bisected into two subtrees using a clustering criterion such as K-Means. The process continues until the desired number of clusters is reached.



# Kruskal's Algorithm (Minimum Spanning Tree)

**Graph:**

- Connected
- Weighted
- Undirected

**Minimum Spanning Tree:**

- Minimum possible total edge weight
- All the vertices connected
- No cycles
- Edges are a sub set of the graph edges

Number of Vertices = 9

Stop when number of edges of the spanning tree = (# of vertices -1) = 9-1 = 8

# Competitive Learning:

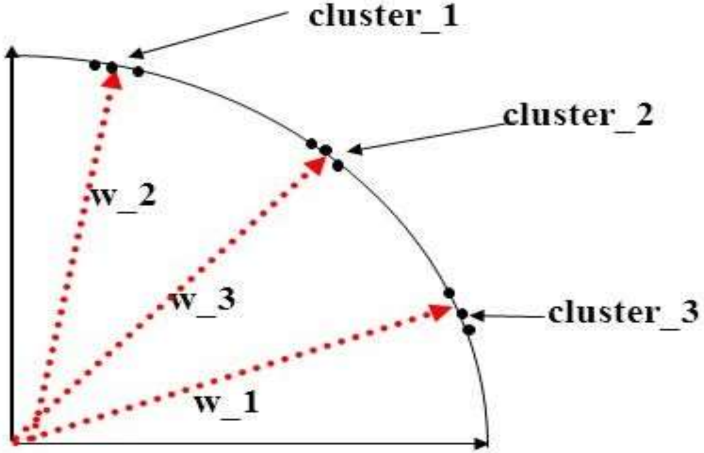
- Competitive learning is a type of unsupervised learning that involves training a neural network to learn a set of features or patterns from the input data.
- One popular competitive learning algorithm is Kohonen’s Self-Organizing Maps (SOM), which is a type of neural network that can be used for clustering.

# Clustering using Kohonen’s Self-Organizing Maps:

- Kohonen’s Self-Organizing Maps (SOM) is a type of neural network that can be used for clustering.
- The algorithm works by mapping the input data onto a low-dimensional grid of neurons.
- Each neuron is connected to the input data through a set of weights, which are adjusted during the training process to learn the patterns in the data.
- The neurons are organized in such a way that neurons that are close to each other on the grid respond to similar input patterns.
- The resulting clusters are represented by groups of neurons that are close to each other on the grid

# Self-Organizing Maps (SOM) (§ 5.5)

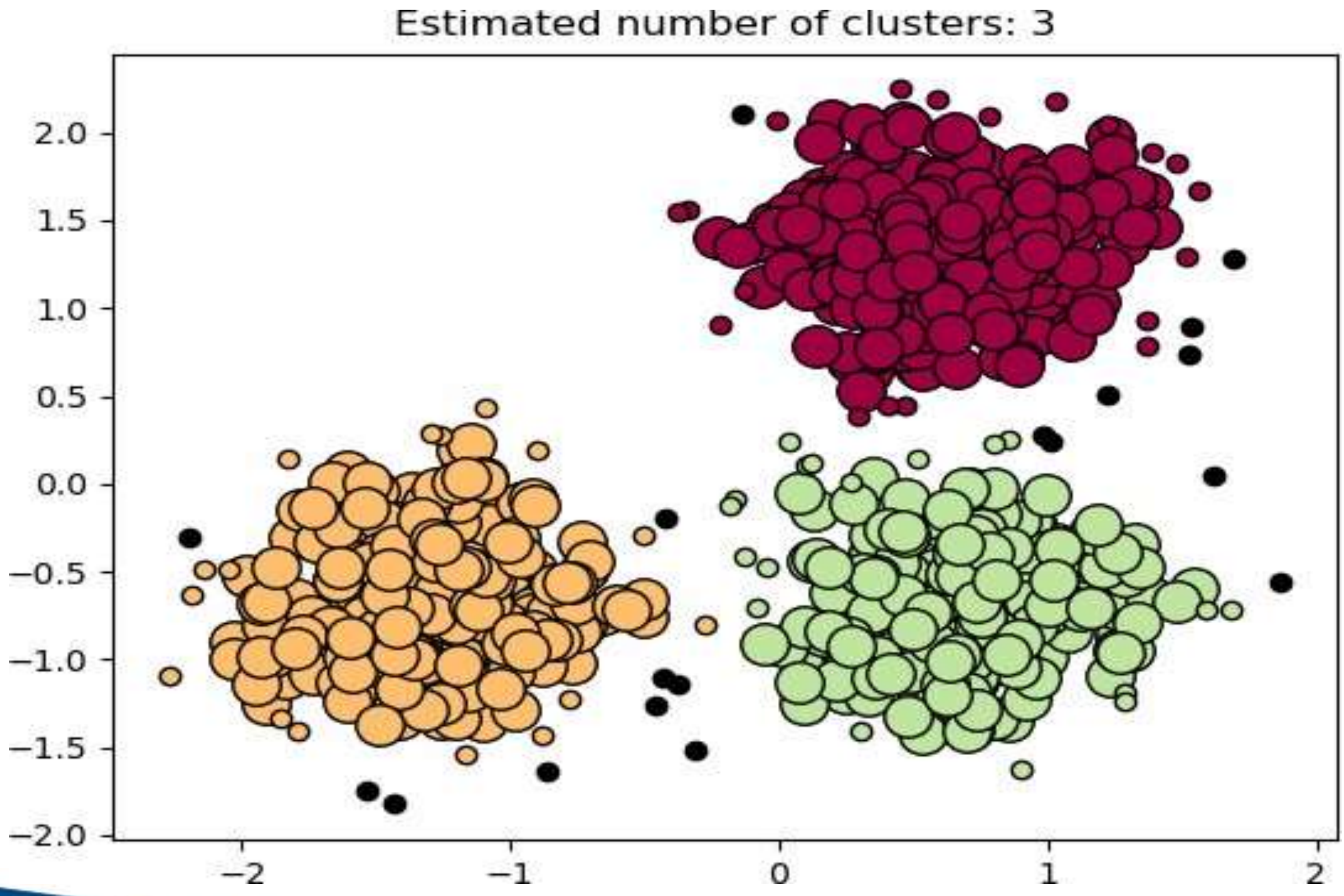
- Competitive learning (Kohonen 1982) is a special case of SOM (Kohonen 1989)
- In competitive learning,
  - the network is trained to organize input vector space into subspaces/classes/clusters
  - each output node corresponds to one class
  - the output nodes are not ordered: ***random map***



- The topological order of the three clusters is 1, 2, 3
- The order of their maps at output nodes are 2, 3, 1
- The map does not preserve the topological order of the training vectors

# Density-Based Spatial Clustering - DBSCAN:

- Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a clustering algorithm that can be used to identify clusters of arbitrary shape.
- The algorithm works by identifying regions of high density in the data and grouping the data points that belong to these regions into clusters.
- DBSCAN has two parameters: epsilon $\epsilon$, which determines the radius of the neighborhood around each data point, and minPts, which determines the minimum number of data points required to form a cluster.


Estimated number of clusters: 3
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2
−1
0
1
2

# Gaussian Mixture Models (GMM):

- Gaussian Mixture Models (GMM) is a clustering algorithm that models the data as a mixture of Gaussian distributions.
- The algorithm works by estimating the parameters of the Gaussian distributions using the Expectation-Maximization (EM) algorithm.
- The resulting clusters are represented by the Gaussian distributions that best fit the data.
- GMM is a flexible algorithm that can model clusters of different shapes and sizes.

# Outlier Detection methods

- Outlier detection is a type of unsupervised learning that involves identifying data points that are significantly different from the majority of the data.
- Two popular outlier detection methods are :-

  1) Isolation Forest

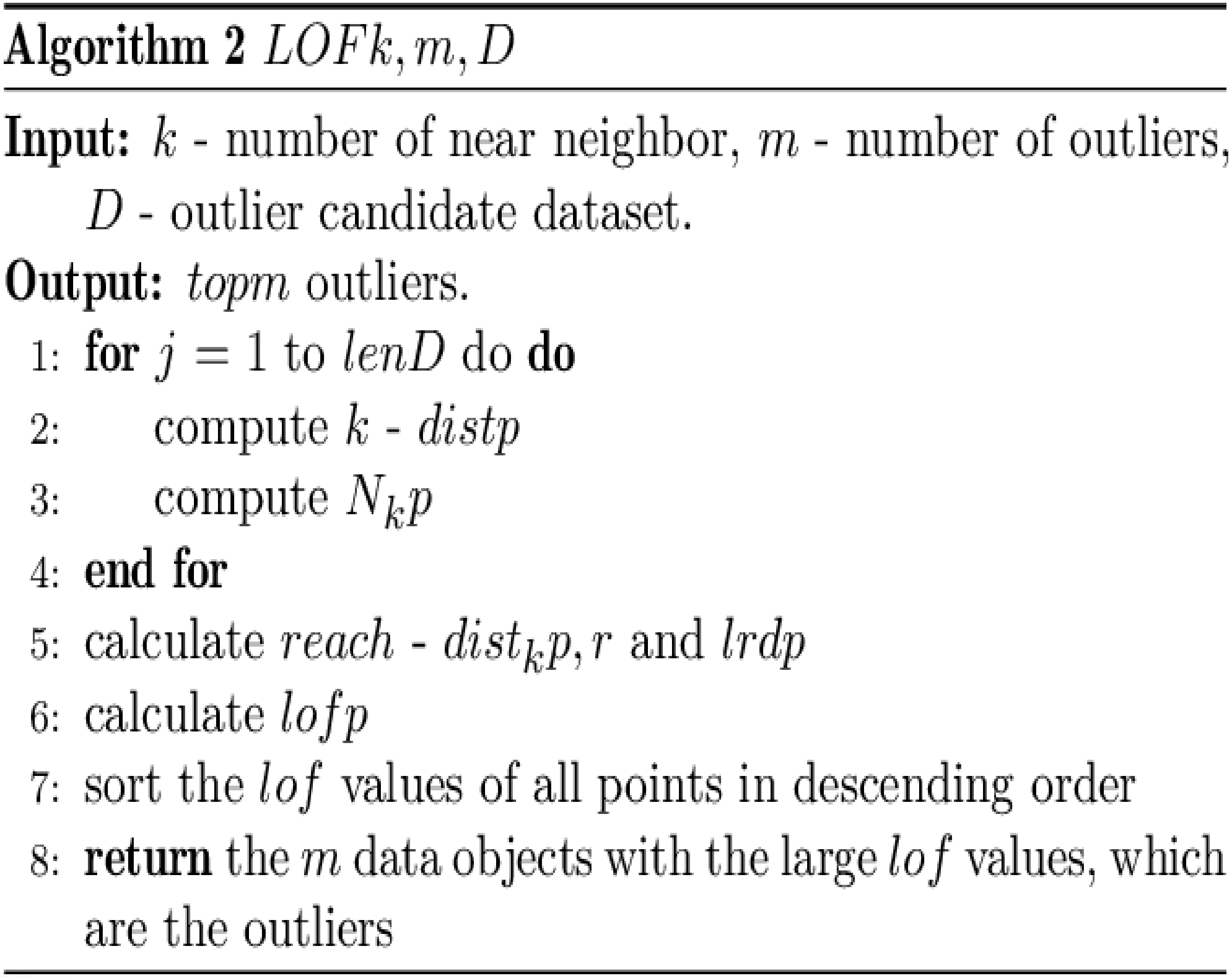
  2) Local Outlier Factor (LOF)

# Isolation Forest:

- Isolation Forest is an outlier detection algorithm that works by randomly partitioning the data points into subsets until each subset contains only one data point.
- The algorithm then computes the path length for each data point in the tree and uses it to determine the anomaly score.

# Local Outlier Factor (LOF)

- Local Outlier Factor (LOF) is an outlier detection algorithm that works by measuring the local density of a data point relative to its neighbors.
- The algorithm computes a score for each data point based on its distance to its k-nearest neighbors and their average distance to each other. Points with a high LOF score are considered outliers.



**Algorithm 2** $LOFk, m, D$

**Input:** $k$ - number of near neighbor, $m$ - number of outliers,
$D$ - outlier candidate dataset.

**Output:** $topm$ outliers.

```
for $j = 1$ to $lenD$ do do
    compute $k$ - $distp$
    compute $N_k p$
end for
calculate $reach$ - $dist_k p, r$ and $lrdp$
calculate $lofp$
sort the $lof$ values of all points in descending order
return the $m$ data objects with the large $lof$ values, which
    are the outliers
```

thank
you