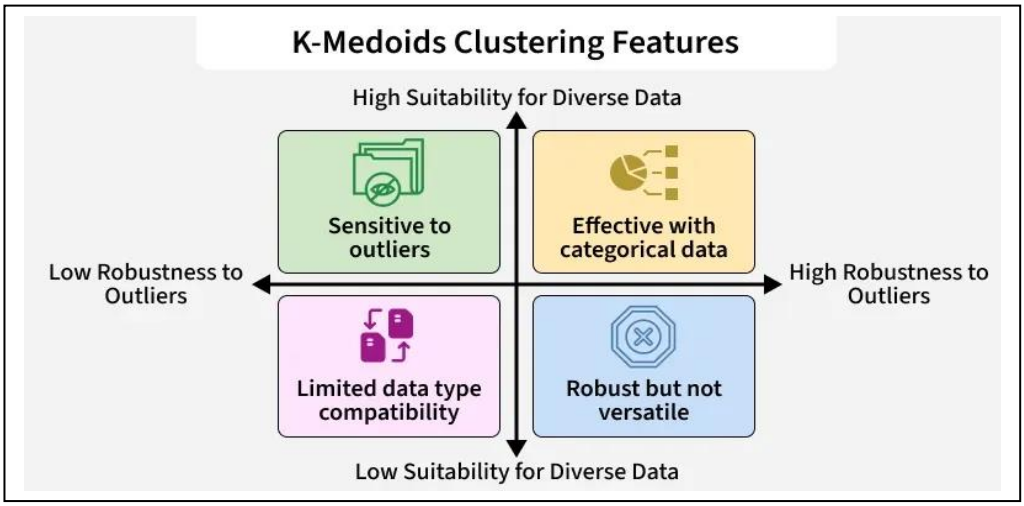
K-MEDOIDS CLUSTERING

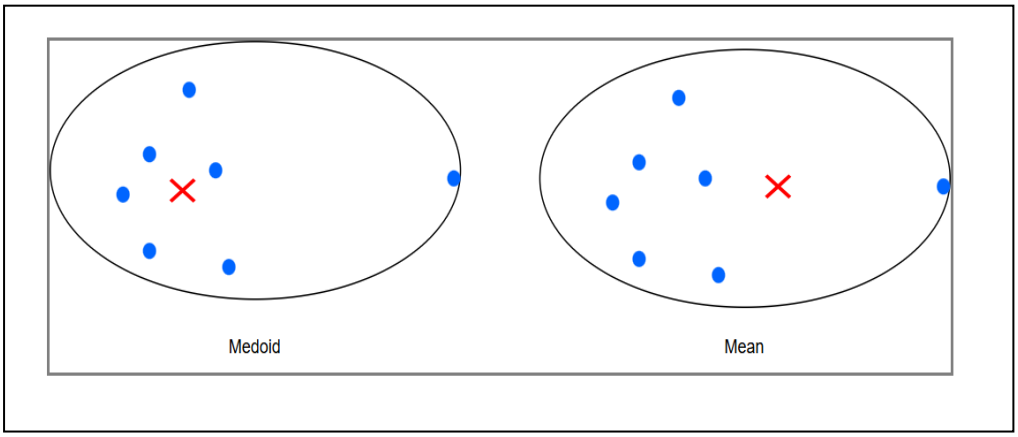
- K-Medoids, also known as Partitioning Around Medoids (PAM), is a clustering algorithm introduced by Kaufman and Rousseeuw.
- It is similar to K-Means, but instead of using the mean of points as a cluster center, it uses an actual data point called a medoid.



- K-medoids, also known as partitioning around medoids (PAM), is a popular clustering algorithm that groups k data points into clusters by selecting k representative objects within a dataset. Clustering is a robust unsupervised machine-learning algorithm that establishes patterns by identifying clusters or groups of data points with similar characteristics within a specific dataset.
- The k-means clustering algorithm uses centroids.
- K-medoids is an alternative clustering algorithm that uses medoids instead.

Medoids

- A medoids can be defined as a point in the cluster within a dataset from which the sum of distances to other points is minimal. It is the data point in a cluster characterized by the lowest dissimilarity with other data points.
- The k-means algorithm is sensitive to outliers. The k-medoids algorithm, on the other hand, mitigates that sensitivity by eliminating reliance on centroids.
- The k-medoids algorithm aims to group data points into k clusters, where each data point is assigned to a medoid, and the sum of distances between data points and their assigned medoid is minimized.
- The algorithm iteratively assigns each data point to the closest medoid and swaps the medoid of each cluster until convergence.



Manhattan distance

- The distance between each data point from both medoids is calculated using the Manhattan distance formula. It is also known as the cost.

$$Distance = |x2 - x1| + |y2 - y1|$$

How the k-medoids algorithm works

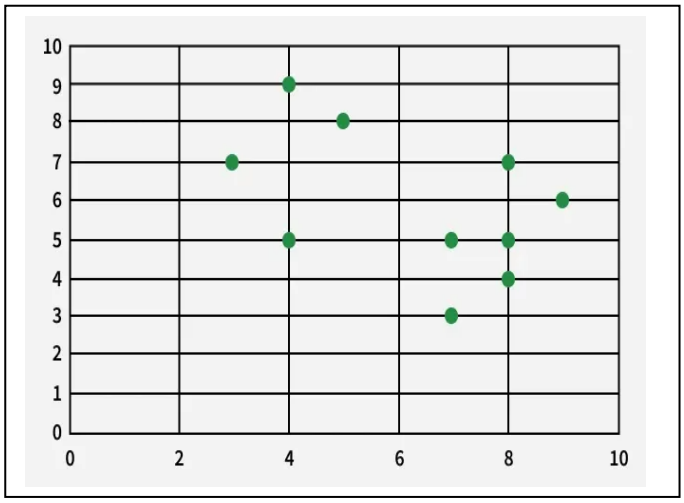
- 1. Initialize: Randomly select k data points from the dataset as initial medoids.
- 2. Assign Points: Assign each data point to the nearest medoid using a distance metric (e.g., Manhattan or Euclidean).
- 3. Update Step (Swap): For each medoid m, try swapping it with a non-medoid point ooo.
- Recalculate the cost for this new configuration.
- If the total cost decreases, accept the swap; otherwise, revert.
- 4. Repeat: Continue until no further cost reduction is possible.

EXAMPLE DEMONSTRATING THE K-MEDOID ALGORITHM.

- Let's consider the following example.

	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

If a graph is drawn using the above data points, we obtain the following:



Step 1: Initialization

- Let the randomly selected 2 medoids be: k=2 and
- let
- C1 = (4, 5) and
- C2 = (8, 5).

Step 2: Calculating cost

- The dissimilarity of each non-medoid point with the medoids is calculated and tabulated:

	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	-	-

Point	**Dist to (4,5)**	**Dist to (8,5)**	**Cluster**
0 (8,7)	$\sqrt{((8-4)^2+(7-5)^2)}=\sqrt{20}=4.47$	$\sqrt{((8-8)^2+(7-5)^2)}=2.00$	Cluster 2
1 (3,7)	$\sqrt{((3-4)^2+(7-5)^2)}=\sqrt{5}=2.24$	$\sqrt{((3-8)^2+(7-5)^2)}=\sqrt{29}=5.39$	Cluster 1
2 (4,9)	$\sqrt{((4-4)^2+(9-5)^2)}=4.00$	$\sqrt{((4-8)^2+(9-5)^2)}=\sqrt{32}=5.66$	Cluster 1
3 (9,6)	$\sqrt{((9-4)^2+(6-5)^2)}=\sqrt{26}=5.10$	$\sqrt{((9-8)^2+(6-5)^2)}=\sqrt{2}=1.41$	Cluster 2
4 (8,5)	$\sqrt{((8-4)^2+(5-5)^2)}=4.00$	0.00	Cluster 2
5 (5,8)	$\sqrt{((5-4)^2+(8-5)^2)}=\sqrt{10}=3.16$	$\sqrt{((5-8)^2+(8-5)^2)}=\sqrt{18}=4.24$	Cluster 1

6 (7,3)	√((7-4)²+(3-5)²)=√13=3.61	√((7-8)²+(3-5)²)=√5=2.24	Cluster 2
7 (8,4)	√((8-4)²+(4-5)²)=√17=4.12	√((8-8)²+(4-5)²)=1.00	Cluster 2
8 (7,5)	√((7-4)²+(5-5)²)=3.00	√((7-8)²+(5-5)²)=1.00	Cluster 2
9 (4,5)	0.00	4.00	Cluster 1

→ Clusters formed:

- Cluster 1 (medoid 4,5): {1,2,5,9}
- Cluster 2 (medoid 8,5): {0,3,4,6,7,8}

Distances to Medoids

We calculate Euclidean distances:

Cluster C1 (4,5):

Point 1 (3,7): $\sqrt{((3-4)^2+(7-5)^2)} = \sqrt{5} \approx 2.24 \rightarrow$ round ≈ 3
Point 2 (4,9): $\sqrt{((4-4)^2+(9-5)^2)} = \sqrt{16} = 4$
Point 5 (5,8): $\sqrt{((5-4)^2+(8-5)^2)} = \sqrt{10} \approx 3.16 \rightarrow$ round ≈ 4

Cluster C2 (8,5):

Point 0 (8,7): $\sqrt{((8-8)^2+(7-5)^2)} = 2$
Point 3 (9,6): $\sqrt{((9-8)^2+(6-5)^2)} = \sqrt{2} \approx 1.41 \rightarrow$ round ≈ 1
Point 6 (7,3): $\sqrt{((7-8)^2+(3-5)^2)} = \sqrt{5} \approx 2.24 \rightarrow$ round ≈ 2
Point 7 (8,4): $\sqrt{((8-8)^2+(4-5)^2)} = 1$
Point 8 (7,5): $\sqrt{((7-8)^2+(5-5)^2)} = 1$

Step 3: Cost Calculation

- Total cost = sum of distances of all points to their medoid.
- Cluster C1 cost = (3 + 4 + 4) = 11
- Cluster C2 cost = (2 + 1 + 2 + 1 + 1) = 7

Total cost = 11 + 7 = 18

Step 4: Swap and Recalculate

Randomly select Point (8,4) as a new medoid.

- Now medoids are:

- C1 = (4,5)

- C2 = (8,4)

Step 5: Re-Assignment

Point	Dist to (4,5)	Dist to (8,4)	Cluster
0 (8,7)	$\sqrt{((8-4)^2+(7-5)^2)}=\sqrt{20}=4.47$	$\sqrt{((8-8)^2+(7-4)^2)}=3.00$	C2
1 (3,7)	$\sqrt{((3-4)^2+(7-5)^2)}=\sqrt{5}=2.24$	$\sqrt{((3-8)^2+(7-4)^2)}=\sqrt{34}=5.83$	C1
2 (4,9)	$\sqrt{((4-4)^2+(9-5)^2)}=4.00$	$\sqrt{((4-8)^2+(9-4)^2)}=\sqrt{41}=6.40$	C1
3 (9,6)	$\sqrt{((9-4)^2+(6-5)^2)}=\sqrt{26}=5.10$	$\sqrt{((9-8)^2+(6-4)^2)}=\sqrt{5}=2.24$	C2
4 (8,5)	$\sqrt{((8-4)^2+(5-5)^2)}=4.00$	$\sqrt{((8-8)^2+(5-4)^2)}=1.00$	C2
5 (5,8)	$\sqrt{((5-4)^2+(8-5)^2)}=\sqrt{10}=3.16$	$\sqrt{((5-8)^2+(8-4)^2)}=\sqrt{25}=5.00$	C1
6 (7,3)	$\sqrt{((7-4)^2+(3-5)^2)}=\sqrt{13}=3.61$	$\sqrt{((7-8)^2+(3-4)^2)}=\sqrt{2}=1.41$	C2
7 (8,4)	$\sqrt{((8-4)^2+(4-5)^2)}=\sqrt{17}=4.12$	0.00	C2

8 (7,5)	$\sqrt{((7-4)^2+(5-5)^2)}=3.00$	$\sqrt{((7-8)^2+(5-4)^2)}=\sqrt{2}=1.41$	C2
9 (4,5)	0.00	$\sqrt{((4-8)^2+(5-4)^2)}=\sqrt{17}=4.12$	C1

Step 3: New Clusters

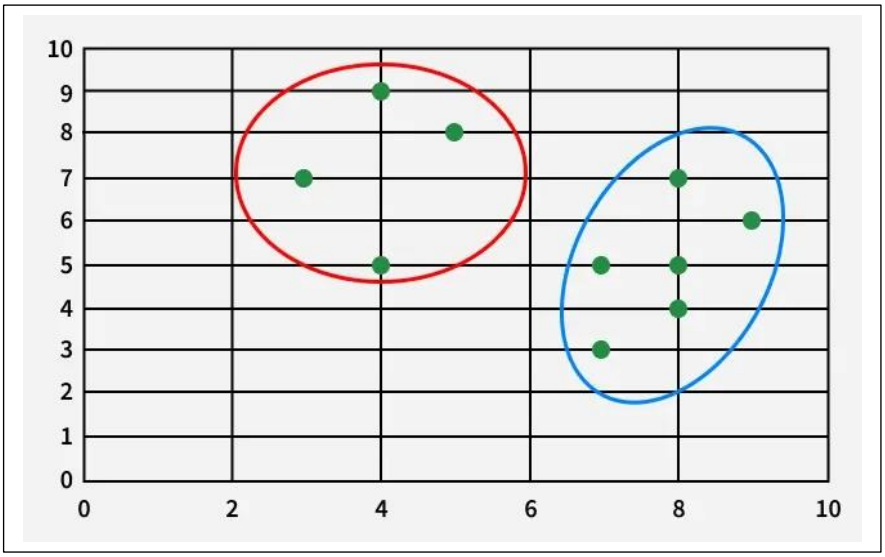
- Cluster C1 (4,5): {1,2,5,9}
- Cluster C2 (8,4): {0,3,4,6,7,8}

Step 4: Cost Calculation

- Cluster C1 cost: 2.24 + 4.00 + 3.16 + 0 = 9.40
- Cluster C2 cost: 3.00 + 2.24 + 1.00 + 1.41 + 0 + 1.41 = 9.06
- Total cost = 9.40 + 9.06 = 18.46

Result After Swap

- New medoids: (4,5) and (8,4)
- Clusters unchanged in membership compared to before (just medoid shifted).
- Cost = 18.46, which is slightly higher than the previous cost (18).
- 👉 Therefore, the swap does not improve the clustering. The algorithm would reject this swap and keep the previous medoids (4,5) and (8,5) as optimal.



Advantages

- It is simple to understand and easy to implement.
- K-Medoids converges in a fixed number of steps.
- It is less sensitive to outliers compared to other partitioning algorithms.

Disadvantages

- Not suitable for non-spherical or arbitrarily shaped clusters.
- Results may differ across runs due to random initialization of medoids.