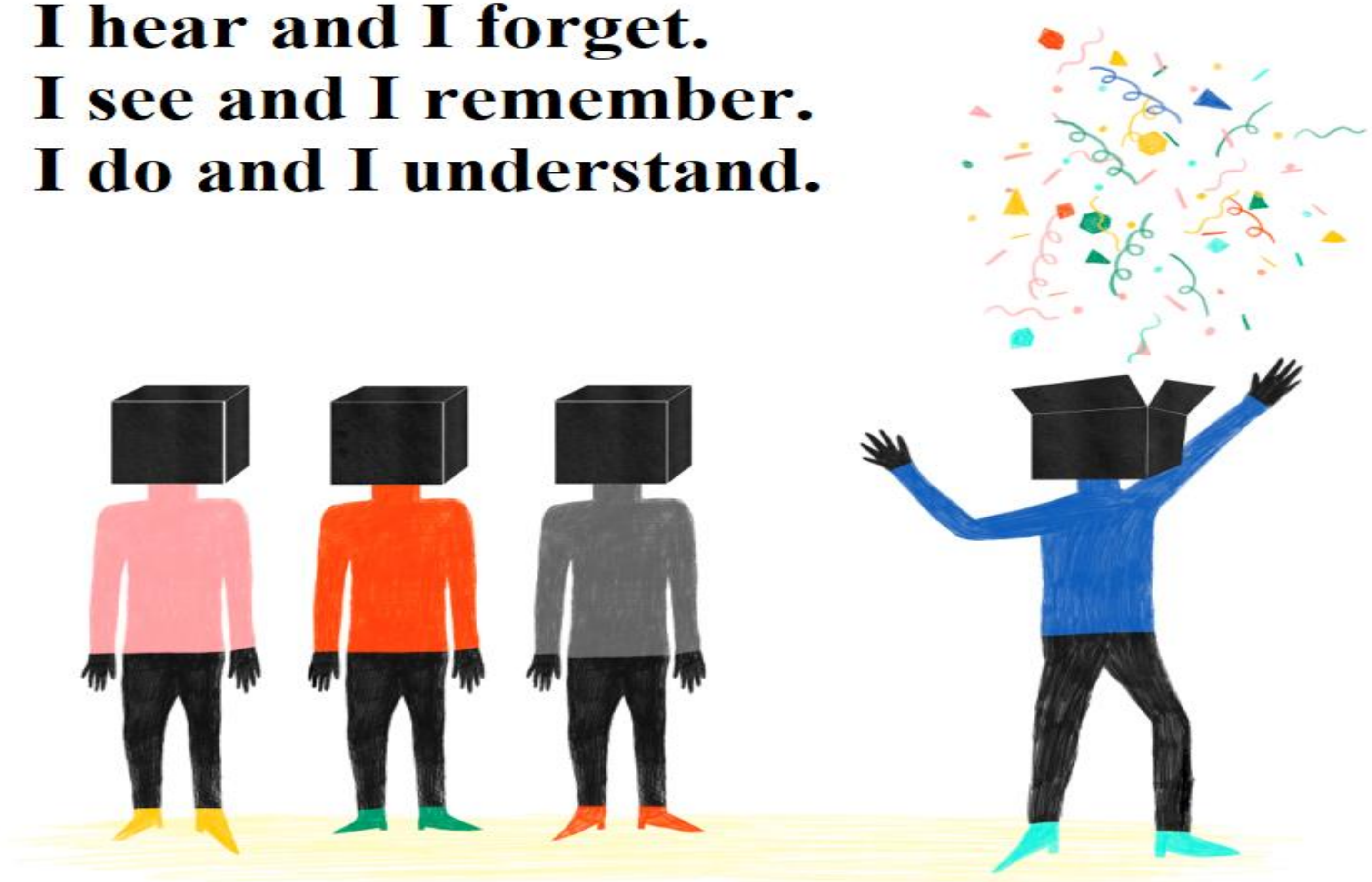
I hear and I forget.
I see and I remember.
I do and I understand.



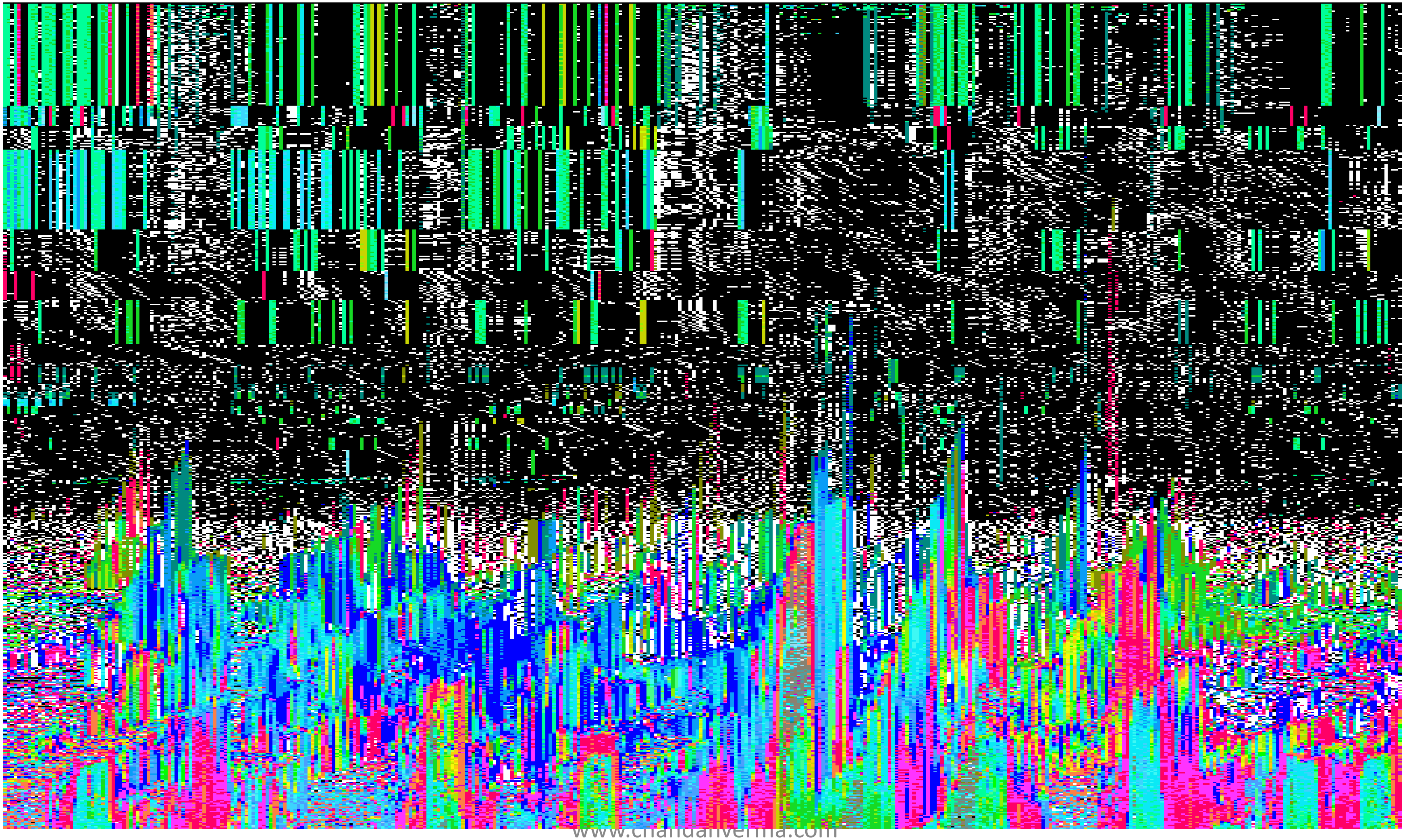
Chandan Verma

Corporate Trainer(Machine Learning,AI,Cloud Computing,IOT)

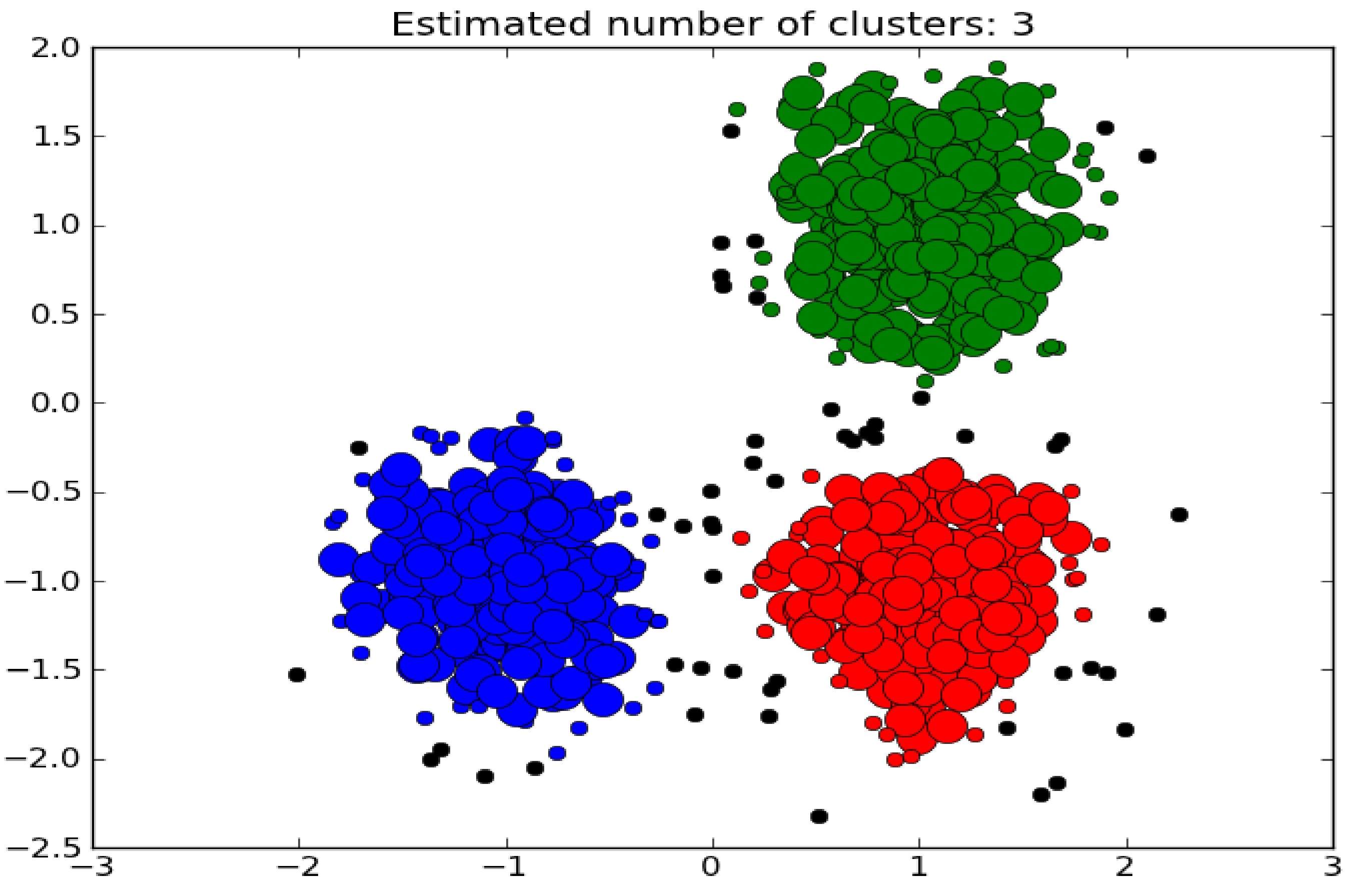
www.facebook.com/verma.chandan.070

Clustering

Clustering



Clustering


Estimated number of clusters: 3
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
−3
−2
−1
0
1
2
3

K Means Clustering

K means Clustering is one of the simplest and most commonly used unsupervised clustering algorithms around.

Clustering is dividing data points into homogeneous classes or clusters:

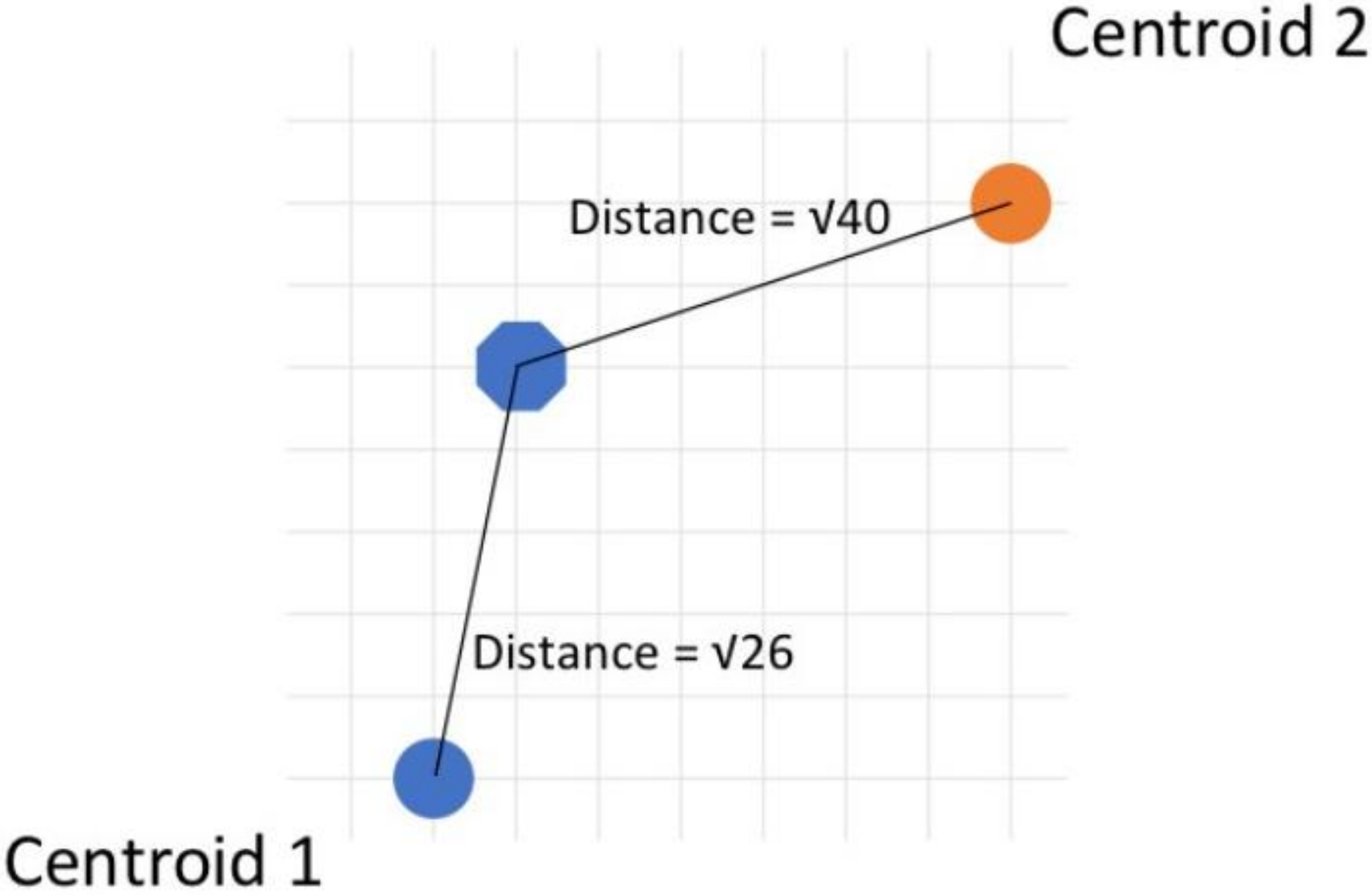
- Points in the same group are as similar as possible
- Points in different group are as dissimilar as possible

K-Means Clustering

K-Means clustering intends to partition n objects into k clusters in which each object belongs to the cluster with the nearest mean. This method produces exactly *k* different clusters of greatest possible distinction.

The objective of K-Means clustering is to minimize total intra-cluster variance, or, the squared error function:

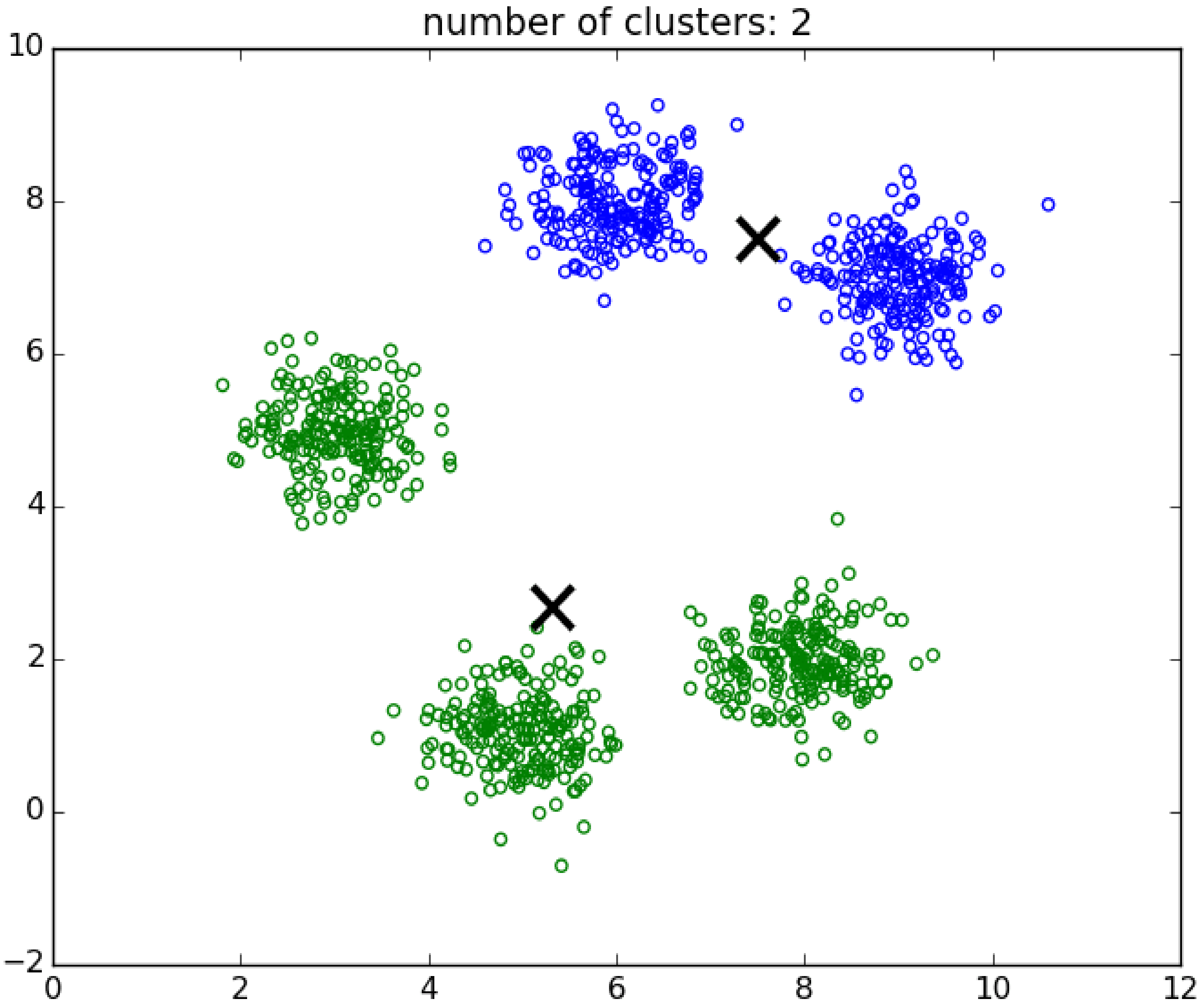
$$\text{objective function} \leftarrow J = \overset{\text{number of clusters}}{\sum_{j=1}^{k}} \overset{\text{number of cases}}{\sum_{i=1}^{n}} \underbrace{\left\| \overset{\text{case } i}{x_i^{(j)}} - \overset{\text{centroid for cluster } j}{c_j} \right\|^2}_{\text{Distance function}}$$


Centroid 2
Distance = √40
Distance = √26
Centroid 1

Clustering Algorithms

A Clustering Algorithm tries to analyze natural groups of data on the basis of some similarity. It locates the centroid of the group of data points. To carry out effective clustering, the algorithm evaluates the distance between each point from the centroid of the cluster.

The goal of clustering is to determine the intrinsic grouping in a set of unlabelled data


number of clusters: 2
10
8
6
4
2
0
−2
0
2
4
6
8
10
12

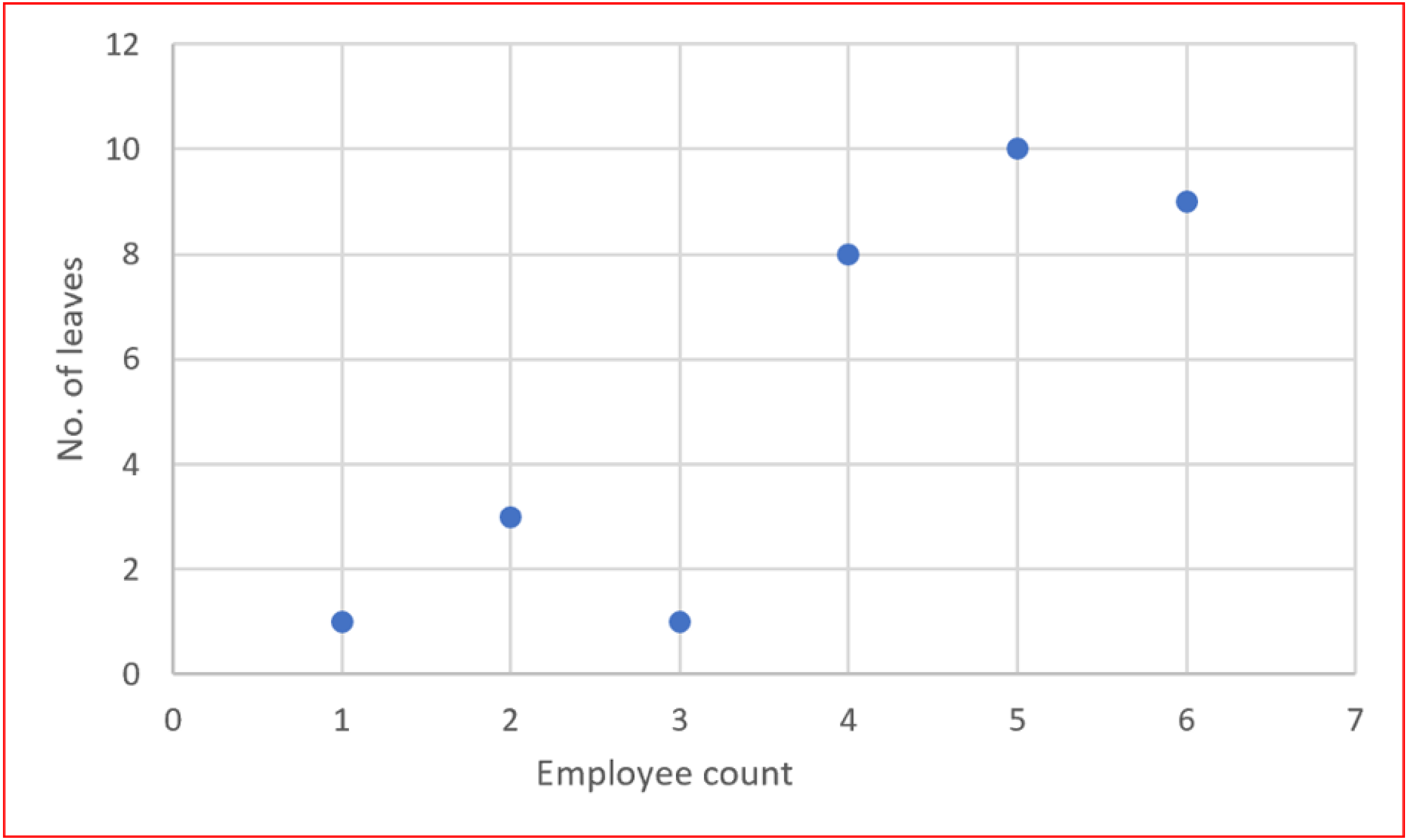
K Means Clustering

K-means clustering algorithm is an iterative algorithm where **K** specifies the number of clusters in which we need to group the data points.

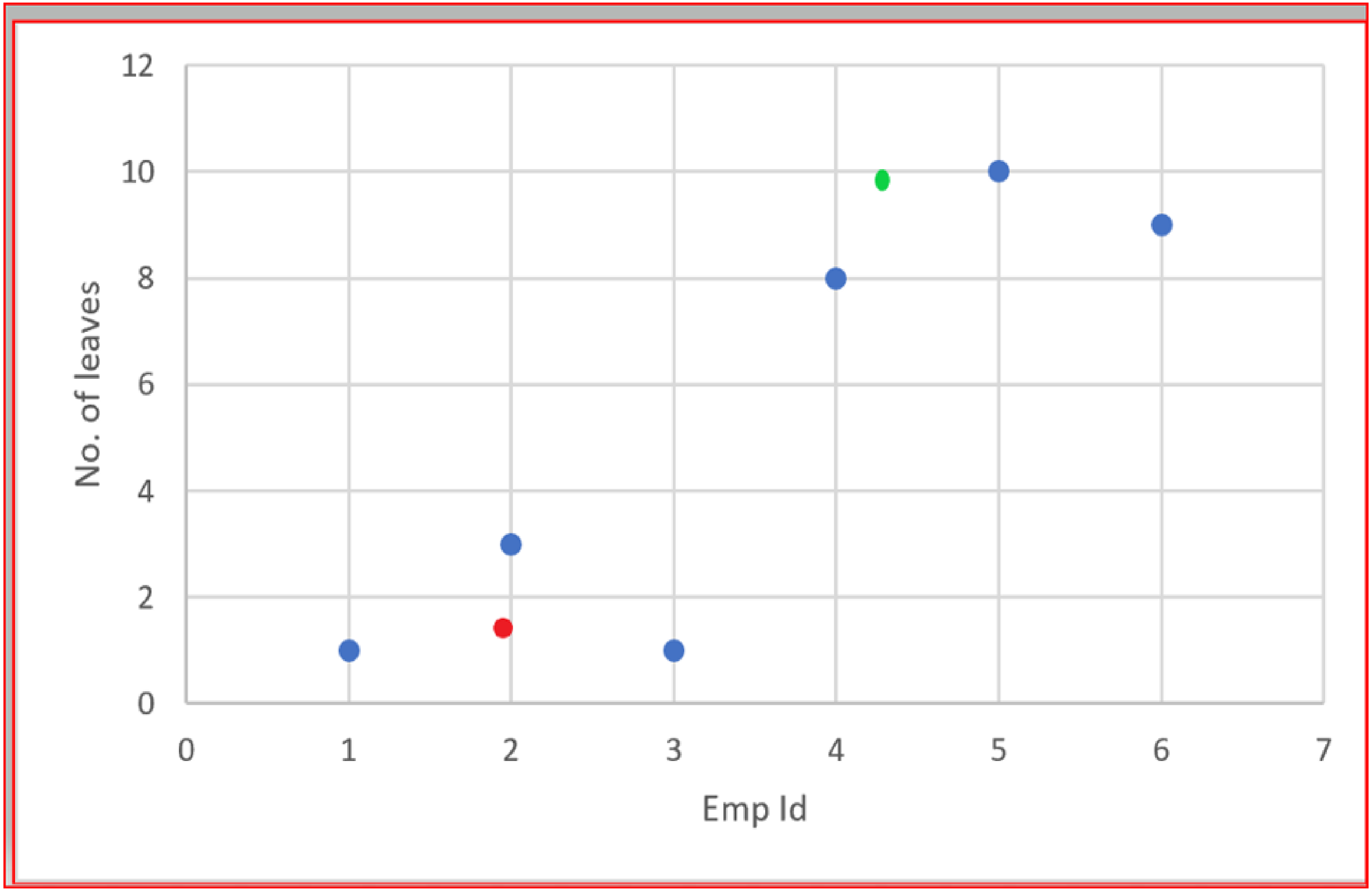
The goal of this algorithm is to find groups in the data, with the number of groups represented by the variable *K*. The algorithm works iteratively to assign each data point to one of *K* groups based on the features that are provided. Data points are clustered based on feature similarity. The results of the *K*-means clustering algorithm are:

K Means Clustering-Step

1. Choose k centroids randomly.
2. Calculate the distance from each point in the dataset to be classified to each centroid.
3. Assign each point to the nearest centroid.
4. Calculate the centroids of the resulting clusters.
5. Repeat until the centroids don't move too much.

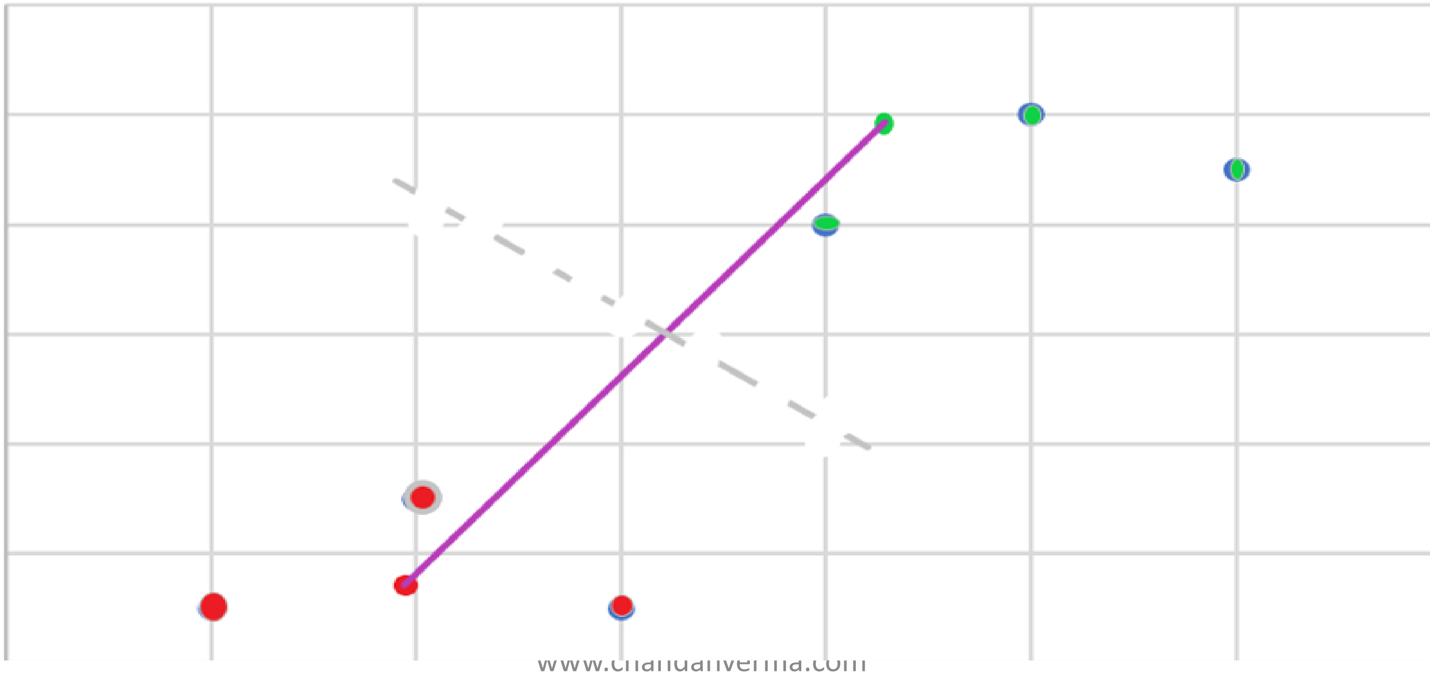

12
10
8
6
4
2
0
No. of leaves
0
1
2
3
4
5
6
7
Employee count

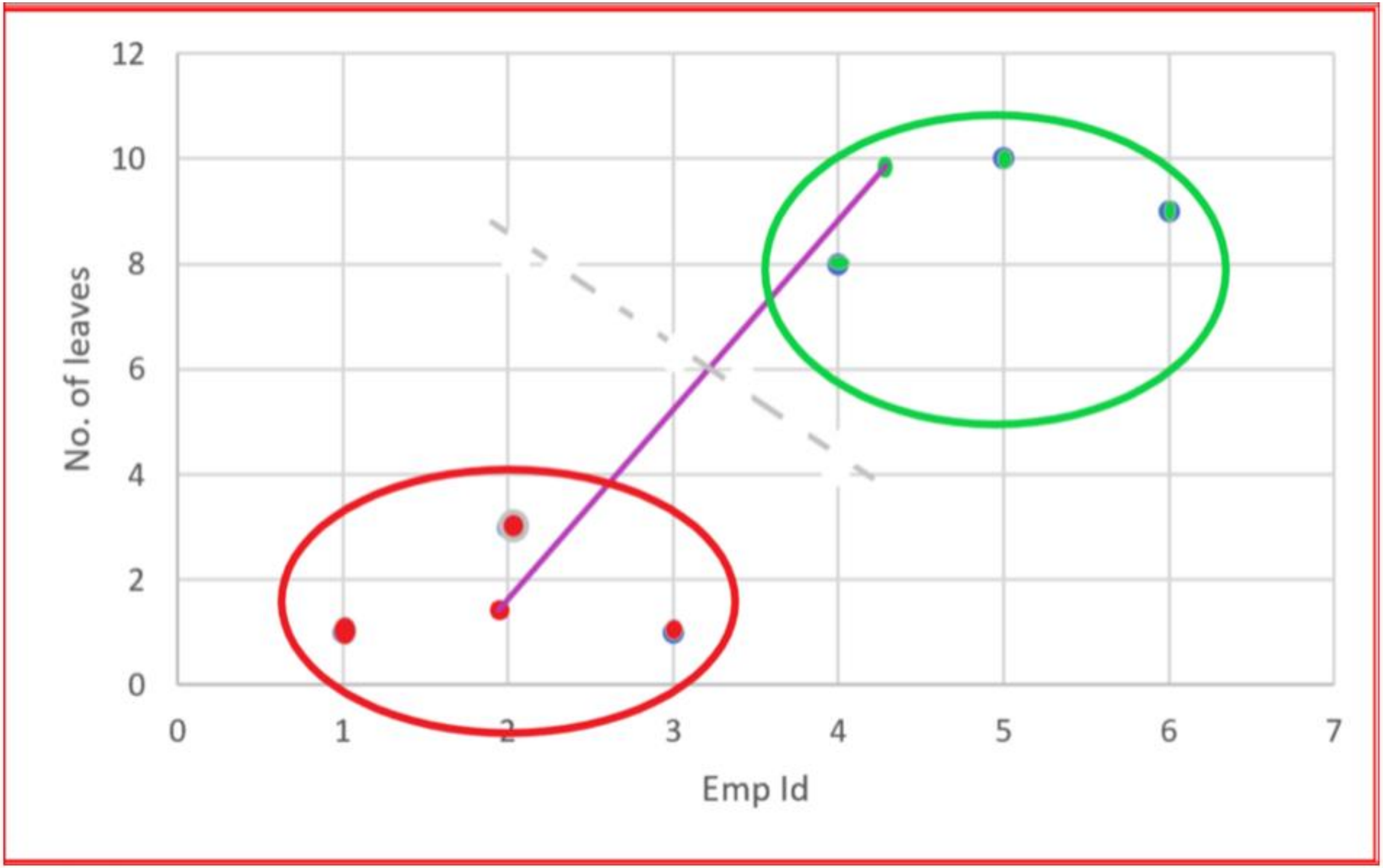
choose no. of clusters K =2



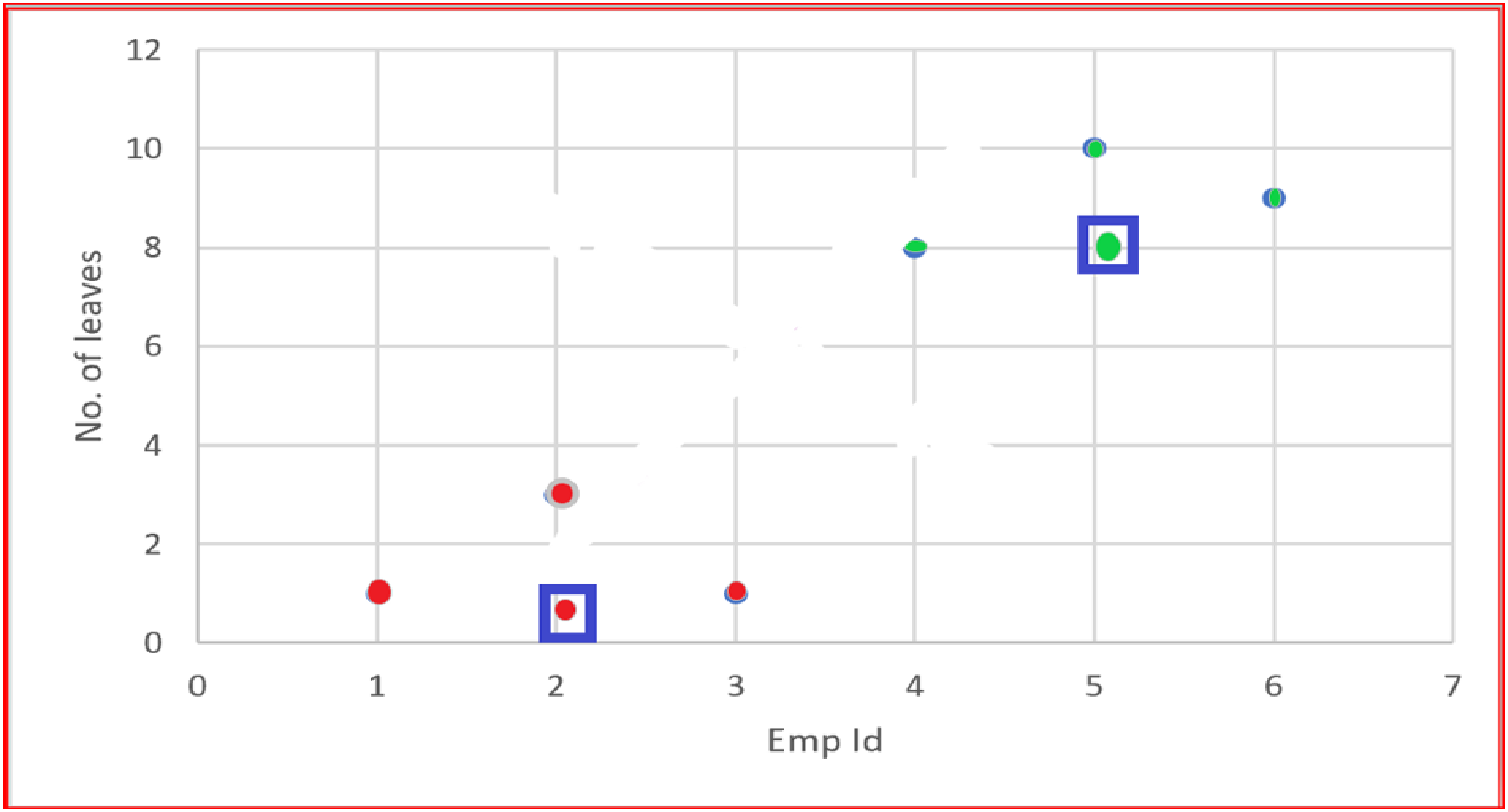
Assign each data point to the closest centroid. we use **euclidean distance,** which is the **ordinary straight line distance between two points**.

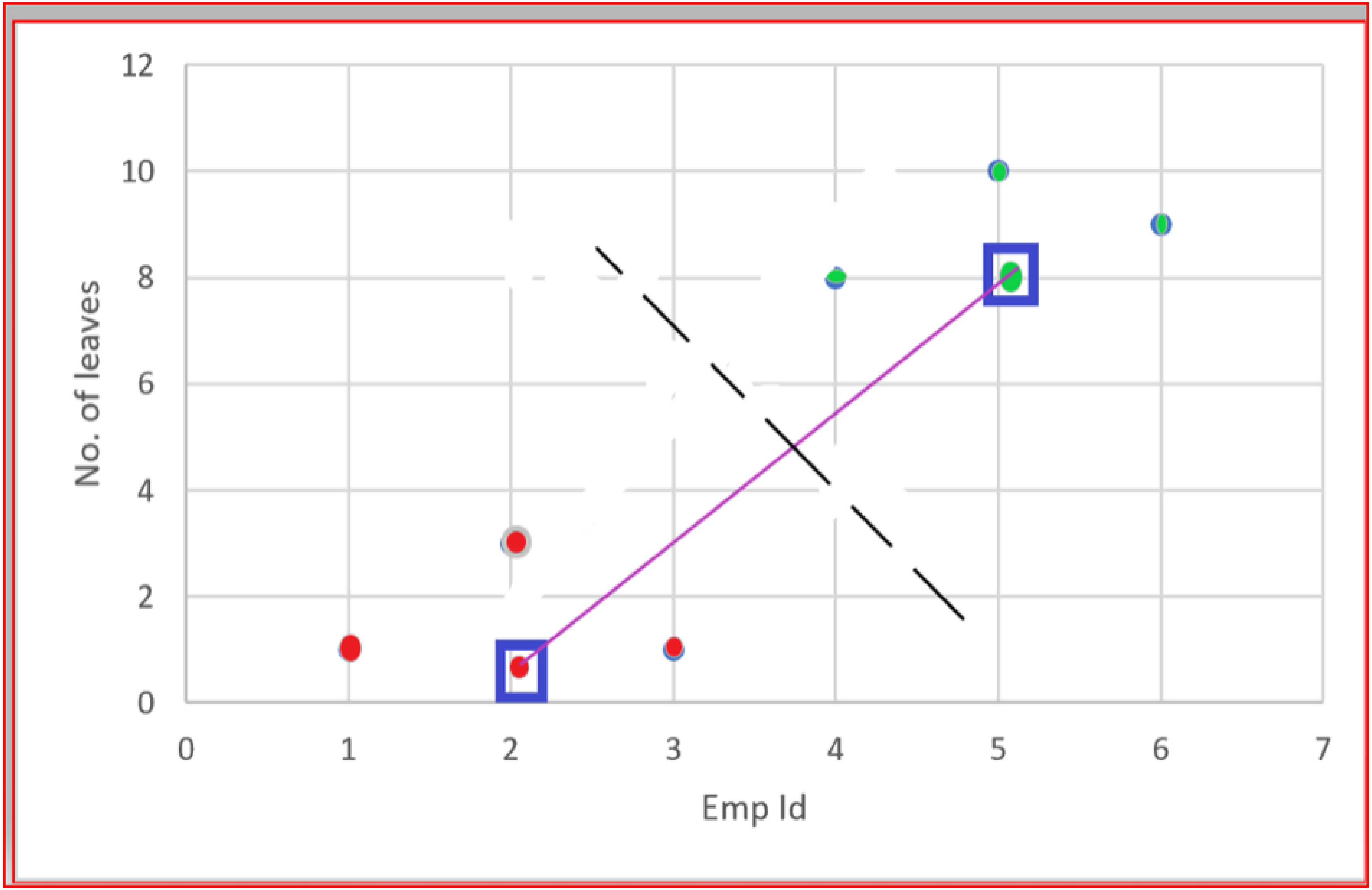
For identifying the closest centroid for each data point, we draw a straight line between the two centroids and draw a perpendicular line from the center of the the straight line joining the two centroids. Perpendicular line helps separate data points into different clusters.



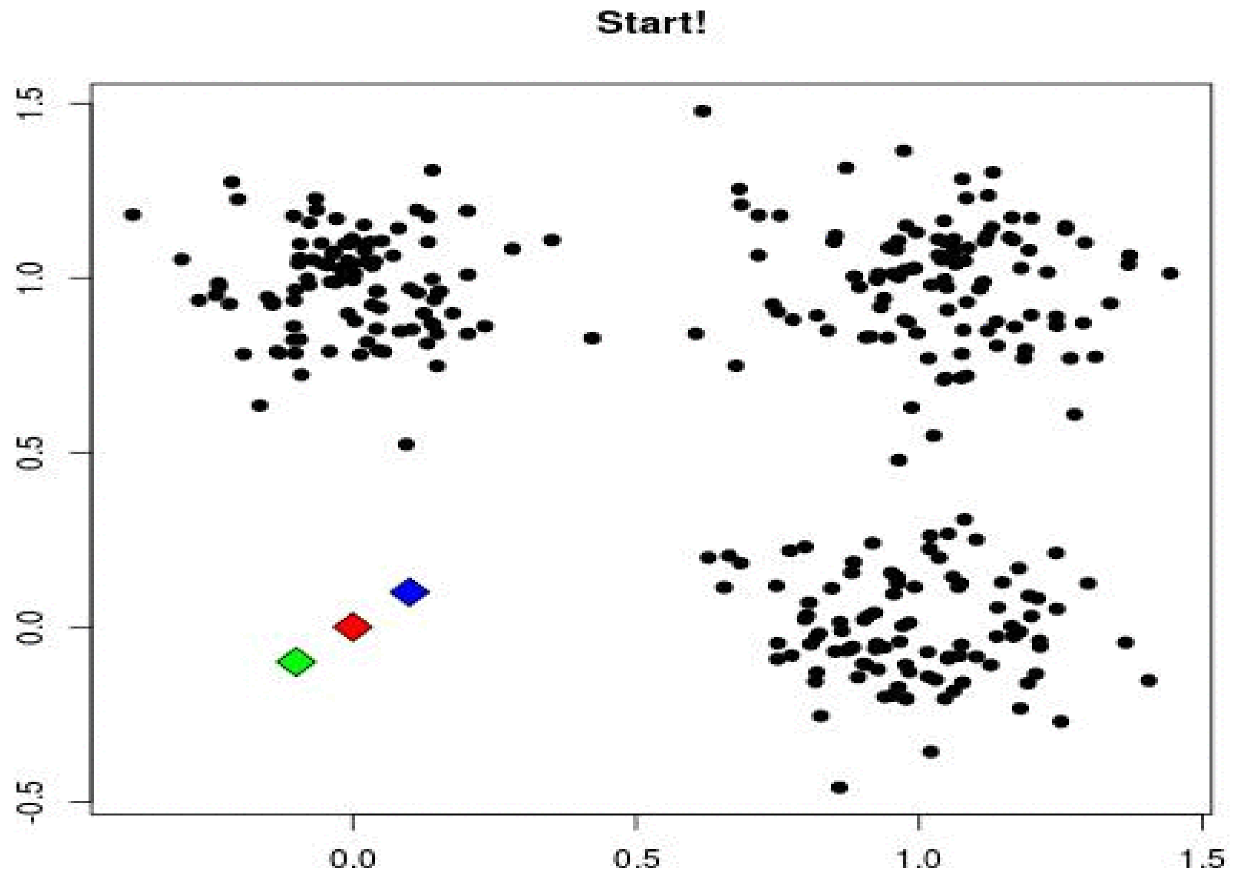

12
10
8
6
4
2
0
No. of leaves
0
1
2
3
4
5
6
7
Emp Id

Compute the new centroid and place it in each cluster. we calculate the average of the data points in the cluster and place the new centroid for each cluster




12
10
8
6
4
2
0
No. of leaves
0
1
2
3
4
5
6
7
Emp Id

Algorithm



Step

We randomly pick K cluster centers(centroids)

assign each input value to closest center. This is done by calculating Euclidean(L2) distance between the point and the each centroid.

$$\arg\min_{c_i \in C} dist(c_i, x)^2$$

Where dist(.) is the Euclidean distance

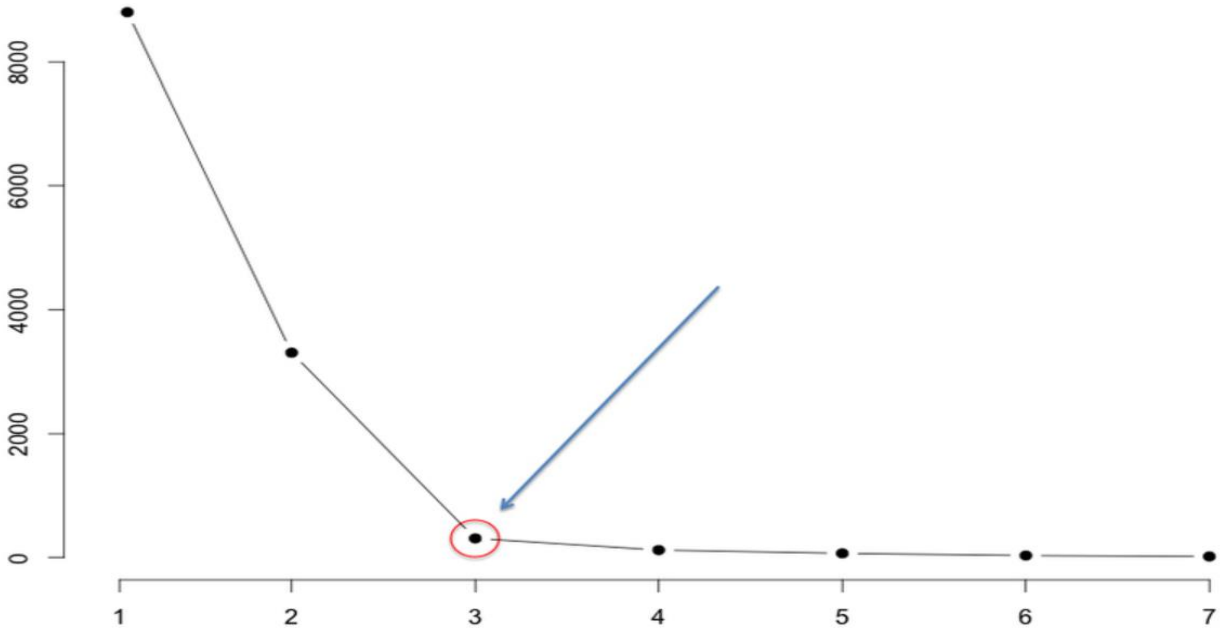
In this step, we find the new centroid by taking the average of all the points assigned to that cluster.

$$c_i = \frac{1}{|S_i|} \sum_{x_i \in S_i} x_i$$

Si is the set of all points assigned to the i*i*th cluster.

Choosing the Value of K

We run the algorithm for different values of K(say K = 10 to 1) and plot the K values against SSE(Sum of Squared Errors). And select the value of K for the elbow point as shown in the figure.



***n* = 19**

15,15,16,19,19,20,20,21,22,28,35,40,41,42,43,44,60,61,65

Initial clusters (random centroid or average):

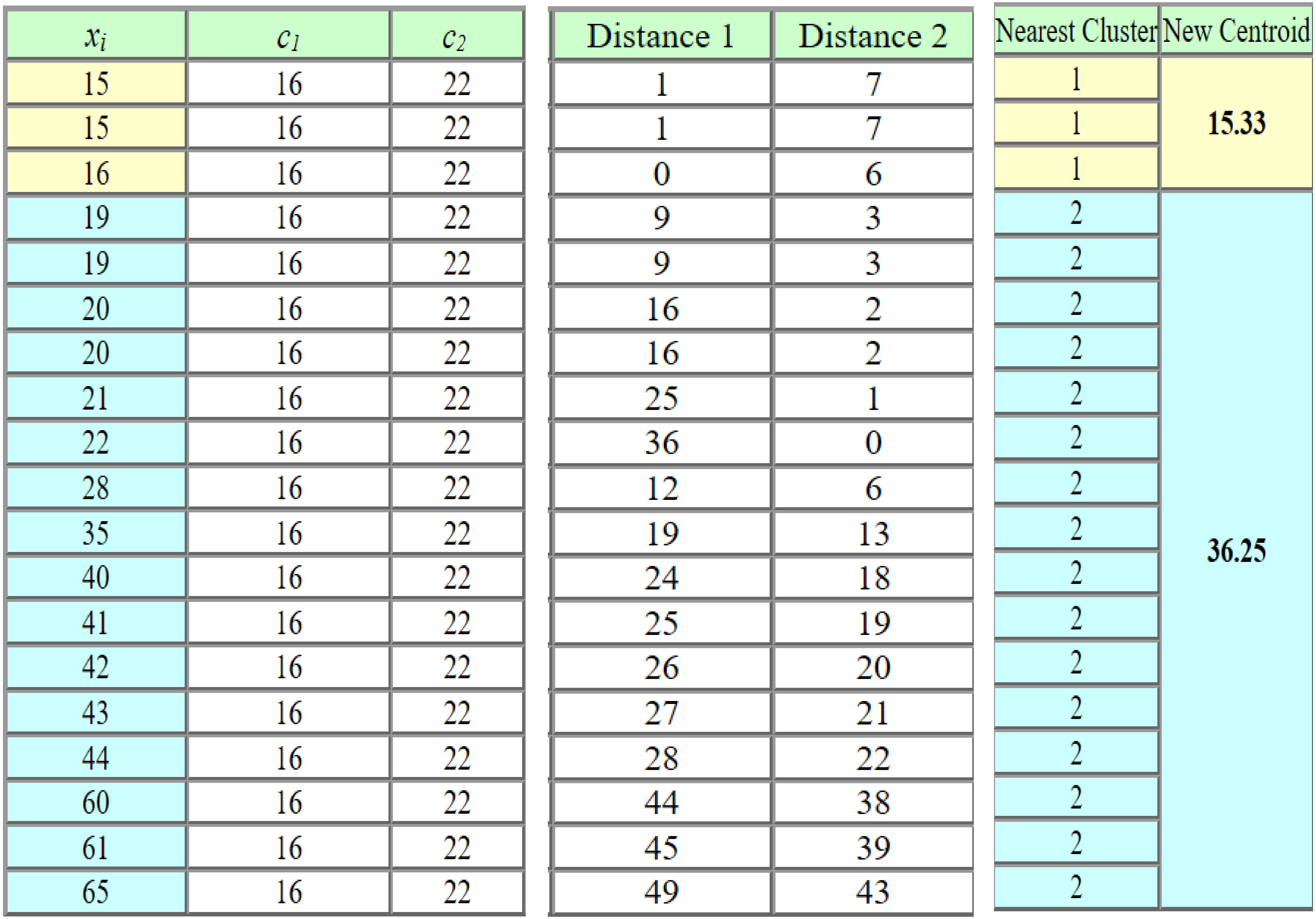
$\boldsymbol{k = 2}$

$c_1 = 16$

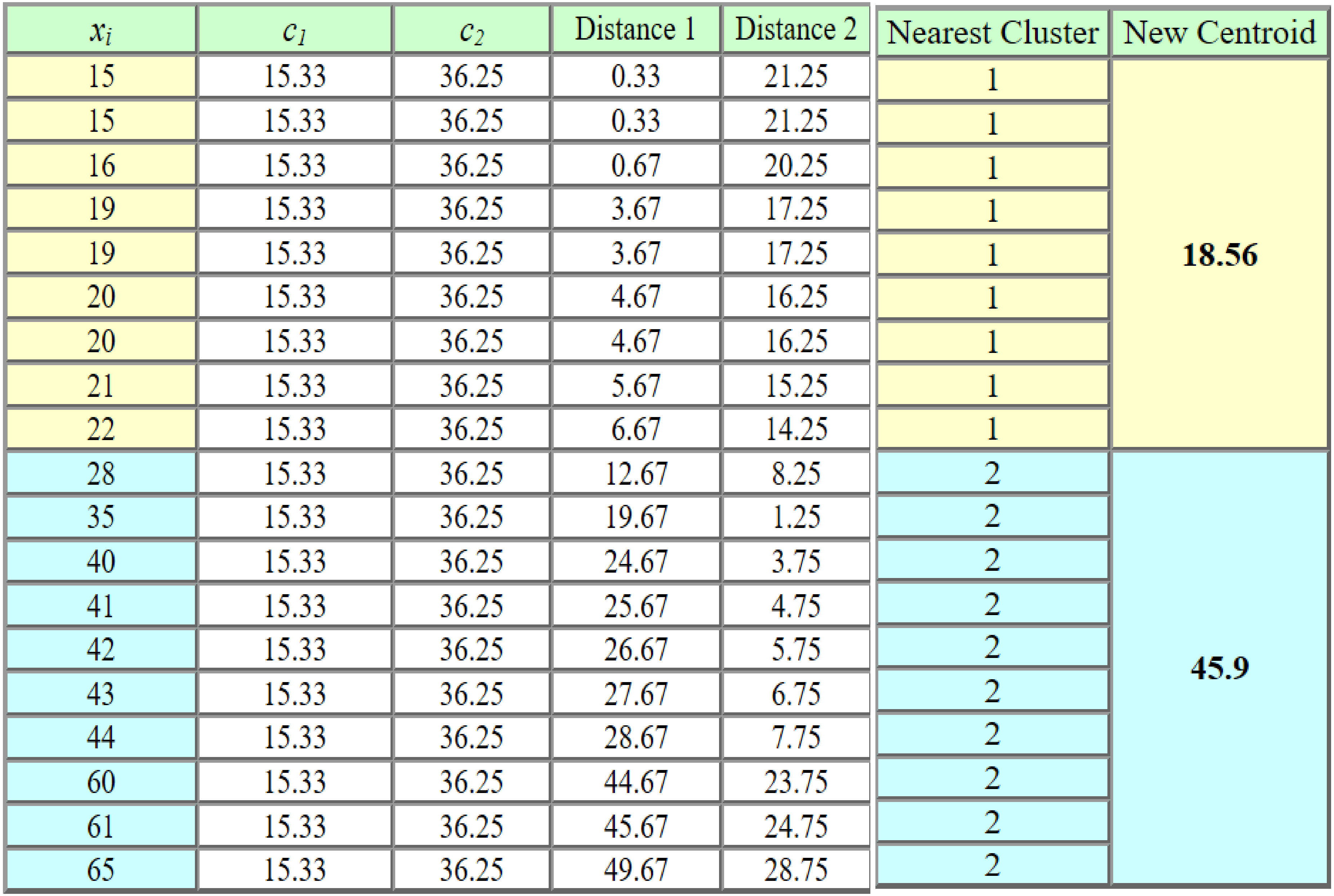
$c_2 = 22$

$$\text{Distance } 1 = |x_i - c_1|$$

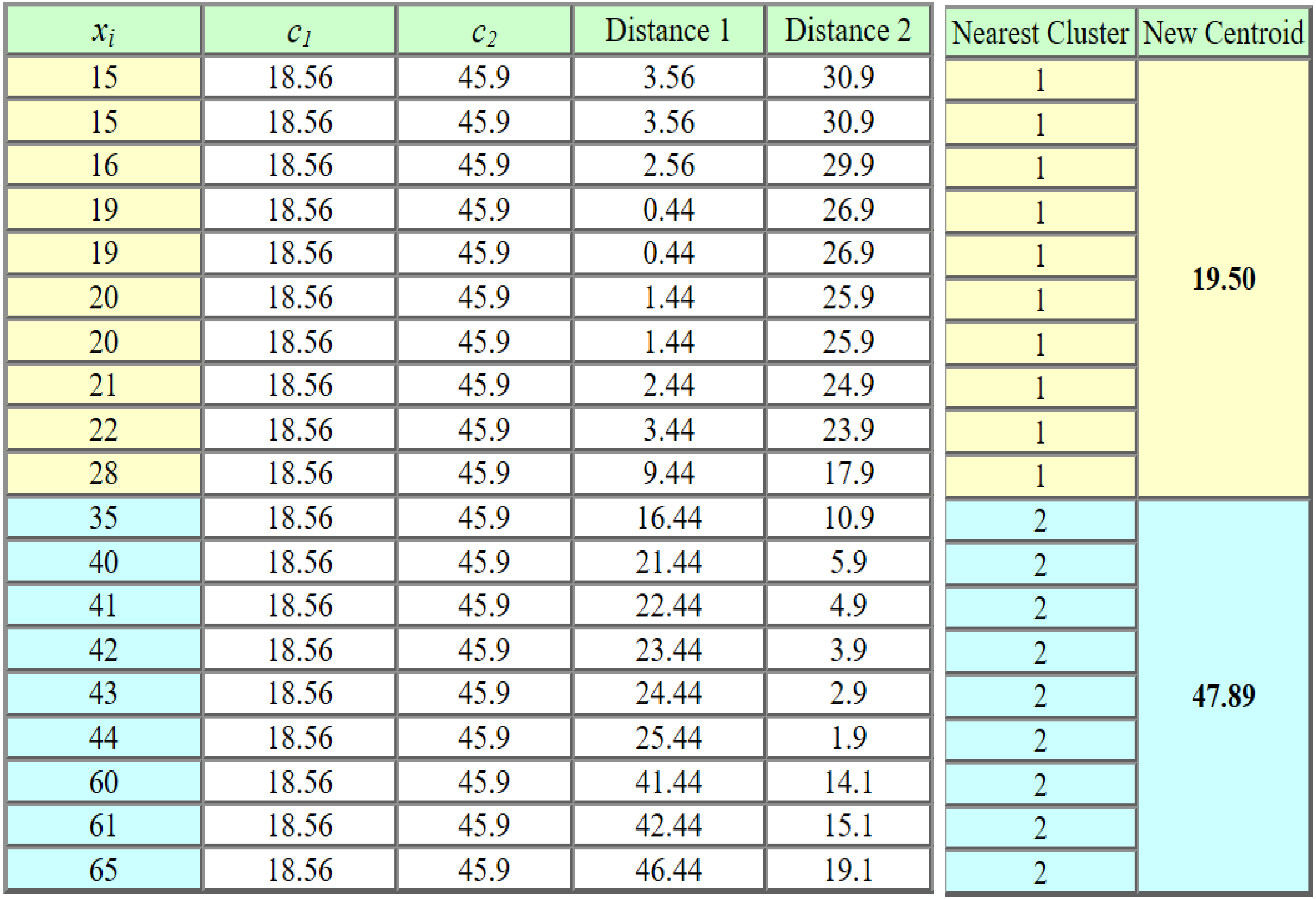
$$\text{Distance } 2 = |x_i - c_2|$$



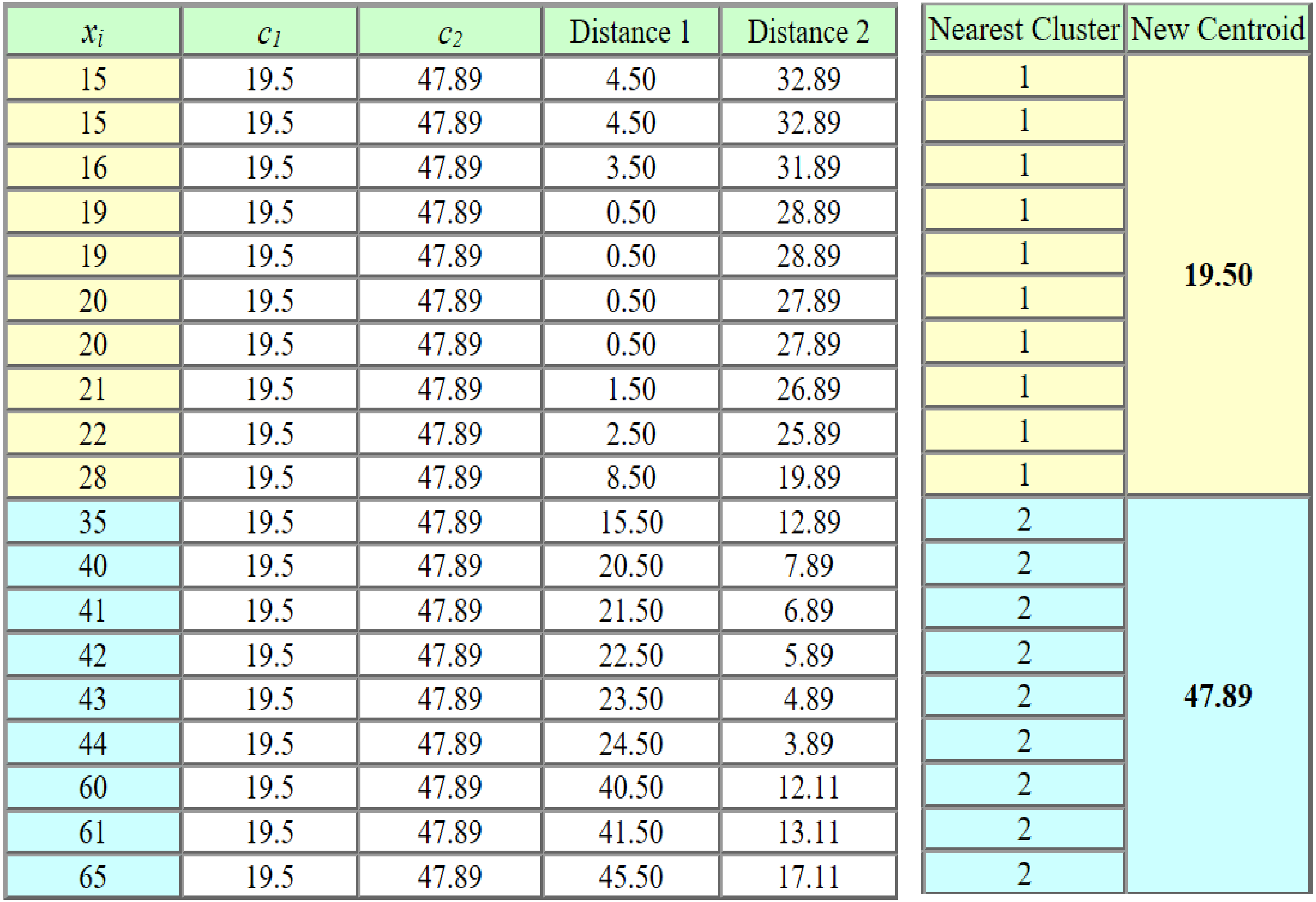
x_i	c_1	c_2	Distance 1	Distance 2	Nearest Cluster	New Centroid
15	16	22	1	7	1	**15.33**
15	16	22	1	7	1	
16	16	22	0	6	1	
19	16	22	9	3	2	**36.25**
19	16	22	9	3	2	
20	16	22	16	2	2	
20	16	22	16	2	2	
21	16	22	25	1	2	
22	16	22	36	0	2	
28	16	22	12	6	2	
35	16	22	19	13	2	
40	16	22	24	18	2	
41	16	22	25	19	2	
42	16	22	26	20	2	
43	16	22	27	21	2	
44	16	22	28	22	2	
60	16	22	44	38	2	
61	16	22	45	39	2	
65	16	22	49	43	2	



x_i	c_1	c_2	Distance 1	Distance 2	Nearest Cluster	New Centroid
15	15.33	36.25	0.33	21.25	1	**18.56**
15	15.33	36.25	0.33	21.25	1	
16	15.33	36.25	0.67	20.25	1	
19	15.33	36.25	3.67	17.25	1	
19	15.33	36.25	3.67	17.25	1	
20	15.33	36.25	4.67	16.25	1	
20	15.33	36.25	4.67	16.25	1	
21	15.33	36.25	5.67	15.25	1	
22	15.33	36.25	6.67	14.25	1	
28	15.33	36.25	12.67	8.25	2	**45.9**
35	15.33	36.25	19.67	1.25	2	
40	15.33	36.25	24.67	3.75	2	
41	15.33	36.25	25.67	4.75	2	
42	15.33	36.25	26.67	5.75	2	
43	15.33	36.25	27.67	6.75	2	
44	15.33	36.25	28.67	7.75	2	
60	15.33	36.25	44.67	23.75	2	
61	15.33	36.25	45.67	24.75	2	
65	15.33	36.25	49.67	28.75	2	

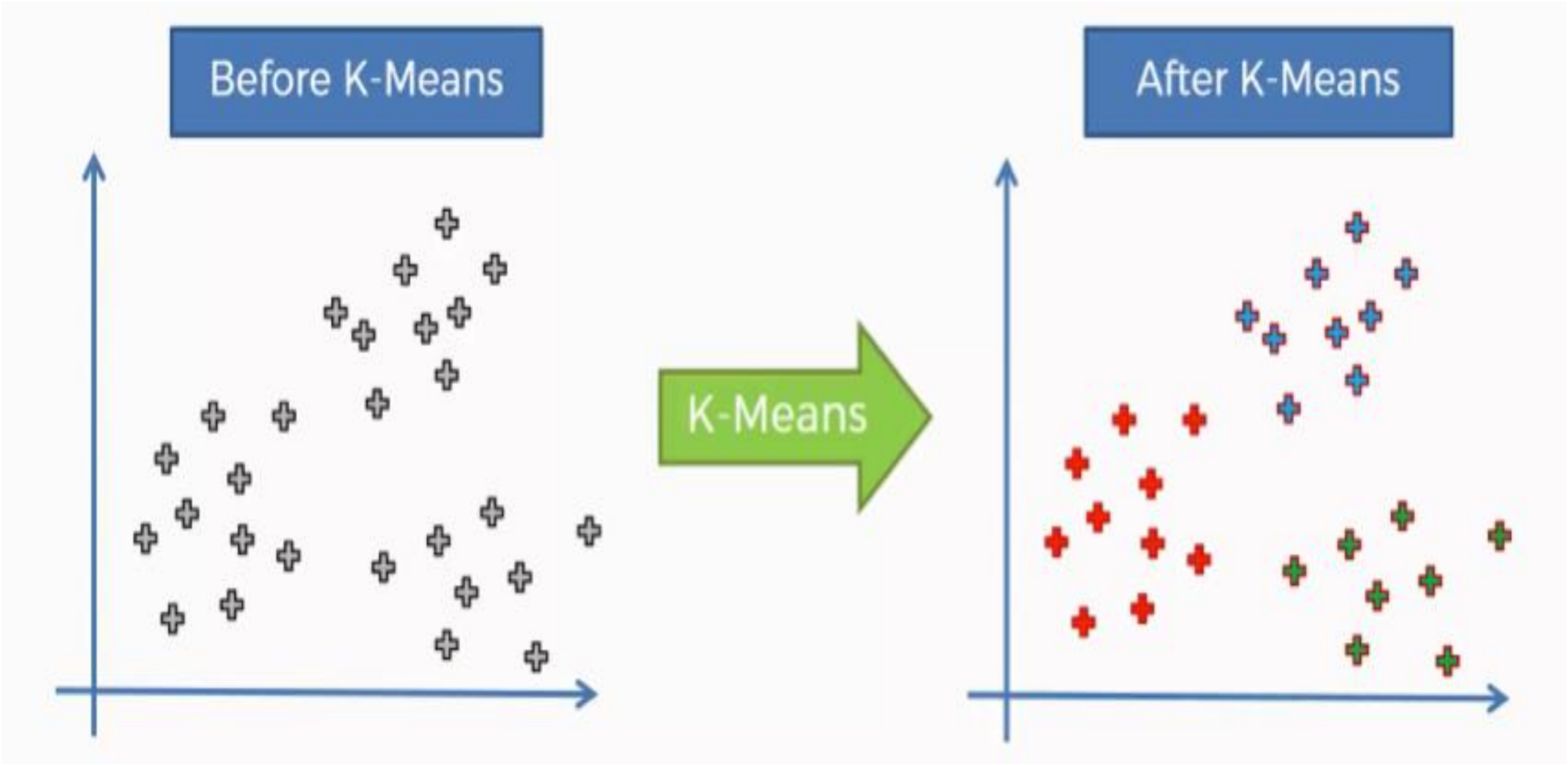


x_i	c_1	c_2	Distance 1	Distance 2	Nearest Cluster	New Centroid
15	18.56	45.9	3.56	30.9	1	**19.50**
15	18.56	45.9	3.56	30.9	1	
16	18.56	45.9	2.56	29.9	1	
19	18.56	45.9	0.44	26.9	1	
19	18.56	45.9	0.44	26.9	1	
20	18.56	45.9	1.44	25.9	1	
20	18.56	45.9	1.44	25.9	1	
21	18.56	45.9	2.44	24.9	1	
22	18.56	45.9	3.44	23.9	1	
28	18.56	45.9	9.44	17.9	1	
35	18.56	45.9	16.44	10.9	2	**47.89**
40	18.56	45.9	21.44	5.9	2	
41	18.56	45.9	22.44	4.9	2	
42	18.56	45.9	23.44	3.9	2	
43	18.56	45.9	24.44	2.9	2	
44	18.56	45.9	25.44	1.9	2	
60	18.56	45.9	41.44	14.1	2	
61	18.56	45.9	42.44	15.1	2	
65	18.56	45.9	46.44	19.1	2	



x_i	c_1	c_2	Distance 1	Distance 2	Nearest Cluster	New Centroid
15	19.5	47.89	4.50	32.89	1	**19.50**
15	19.5	47.89	4.50	32.89	1	
16	19.5	47.89	3.50	31.89	1	
19	19.5	47.89	0.50	28.89	1	
19	19.5	47.89	0.50	28.89	1	
20	19.5	47.89	0.50	27.89	1	
20	19.5	47.89	0.50	27.89	1	
21	19.5	47.89	1.50	26.89	1	
22	19.5	47.89	2.50	25.89	1	
28	19.5	47.89	8.50	19.89	1	
35	19.5	47.89	15.50	12.89	2	**47.89**
40	19.5	47.89	20.50	7.89	2	
41	19.5	47.89	21.50	6.89	2	
42	19.5	47.89	22.50	5.89	2	
43	19.5	47.89	23.50	4.89	2	
44	19.5	47.89	24.50	3.89	2	
60	19.5	47.89	40.50	12.11	2	
61	19.5	47.89	41.50	13.11	2	
65	19.5	47.89	45.50	17.11	2	

K-mean



sklearn.cluster.KMeans

n_clusters : int, optional, default: 8
The number of clusters to form as well as the number of centroids to generate.

init : {'k-means++', 'random' or an ndarray}
Method for initialization, defaults to 'k-means++':
'**k-means++' :** selects initial cluster centers for k-mean clustering in a smart way to speed up convergence.

n_init : int, default: 10
Number of time the k-means algorithm will be run with different centroid seeds. The final results will be the best output of n_init consecutive runs in terms of inertia.

max_iter : int, default: 300

Maximum number of iterations of the k-means algorithm for a single run.

tol : float, default: 1e-4

Relative tolerance with regards to inertia to declare convergence