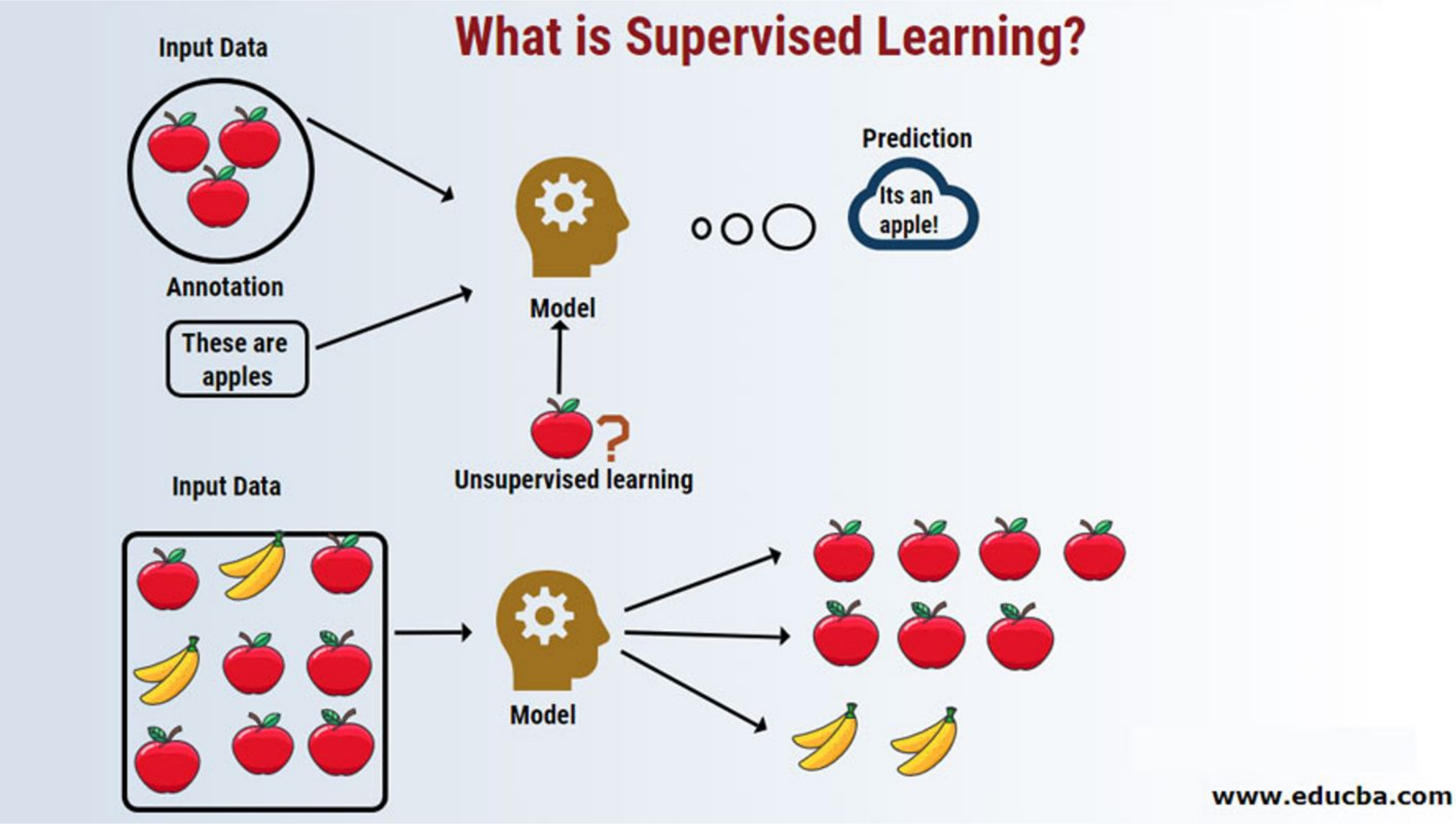
K-Means Clustering

Unsupervised Learning

Image recognition: supervised or unsupervised learning?

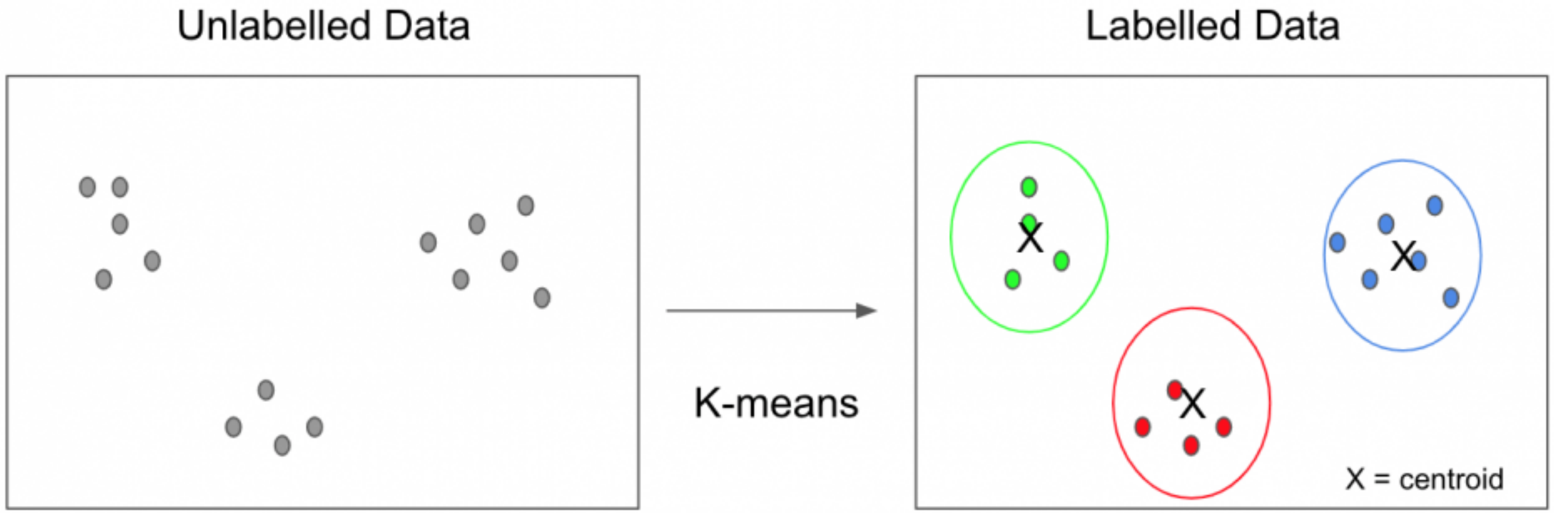
Supervised vs. Unsupervised Learning



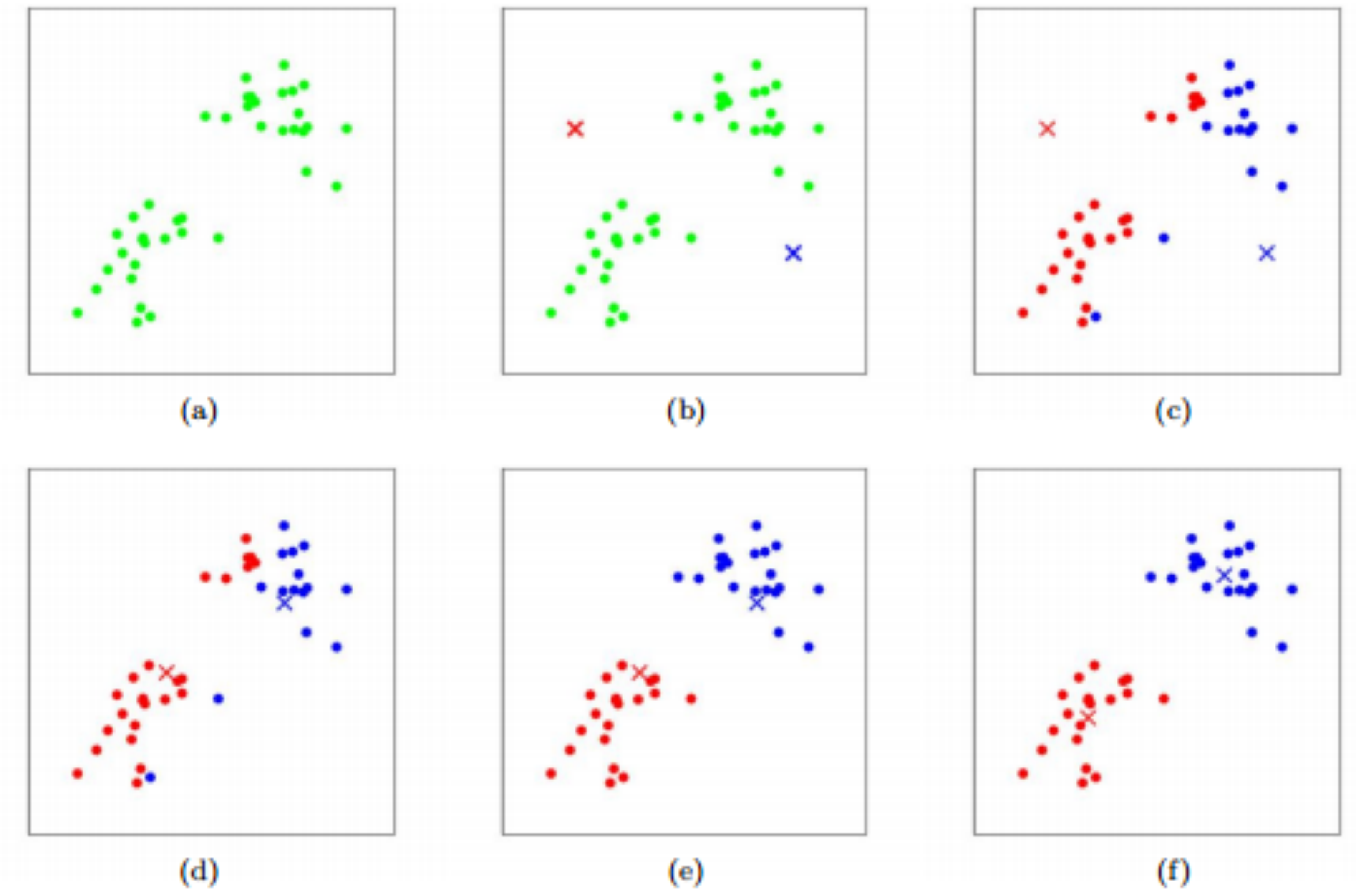
Clustering

- You can think of clustering as grouping unlabeled data.
- So far we’ve discussed supervised learning types algorithms, which include regression and classification. And to recap, the way supervised learning algorithms work is that you have some training data and answers in that training data that you supply to the model.
- On the other hand, unsupervised learning is different, in which we don’t have answers, and the model has to think for itself. In clustering, we don’t have any classes or categories in advance, we don’t have any training data, we just have data and we want to create the clusters (groups).

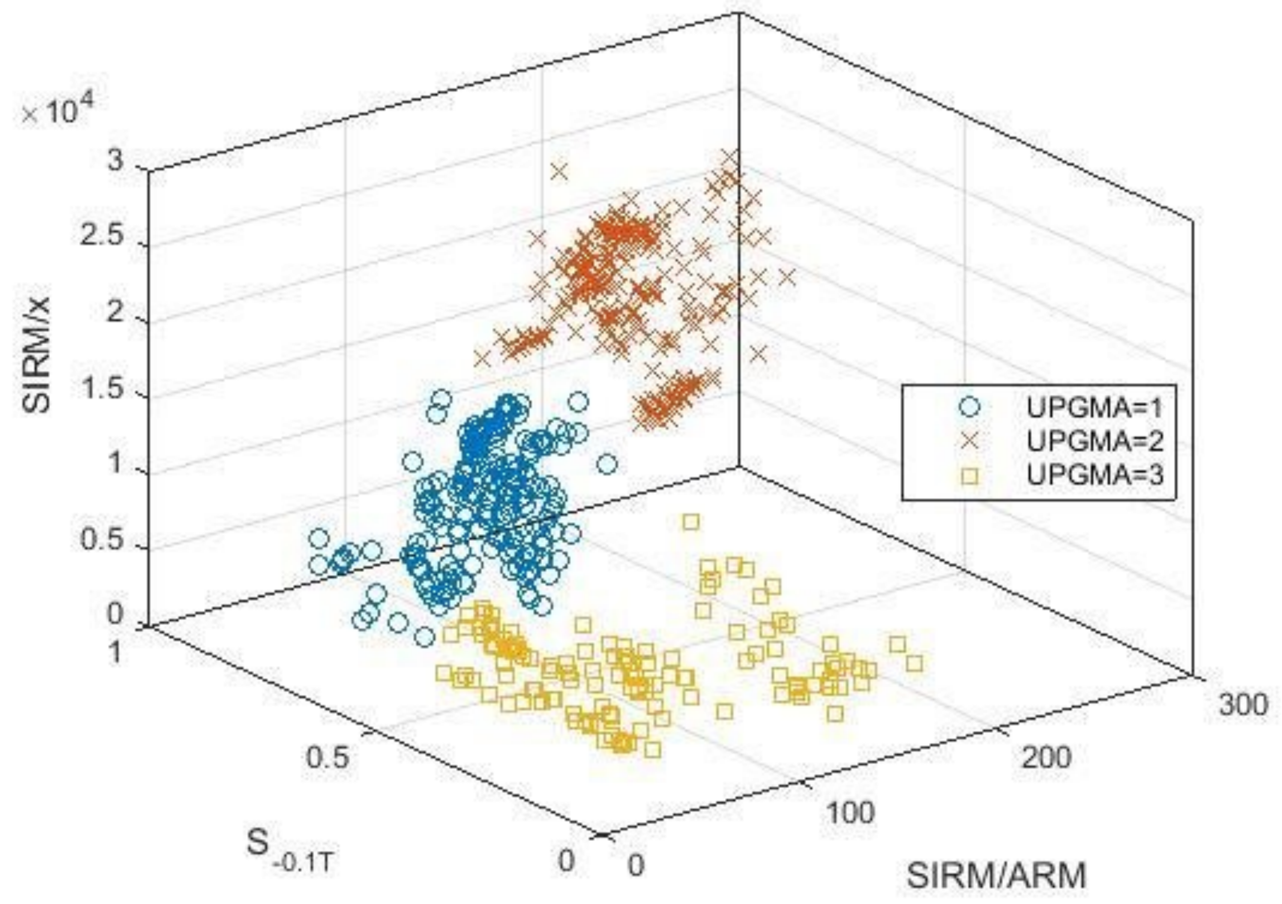
Clustering:



K-means clustering steps:



K-means clustering in higher dimensions

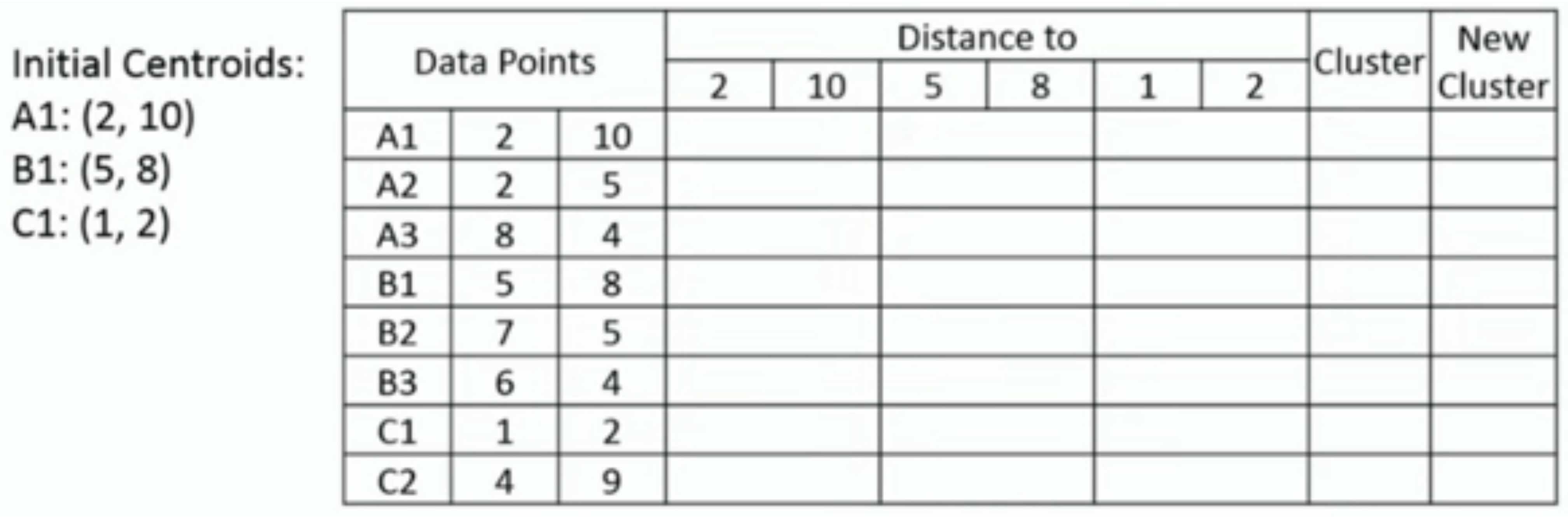


K-means clustering steps:

- Step1: Choose K number of random data points as initial centroids (cluster centers).
- Step2: repeat till cluster centers stabilize:

- Allocate each point to the nearest of centroid
- Compute centroid for the cluster using all points in the cluster

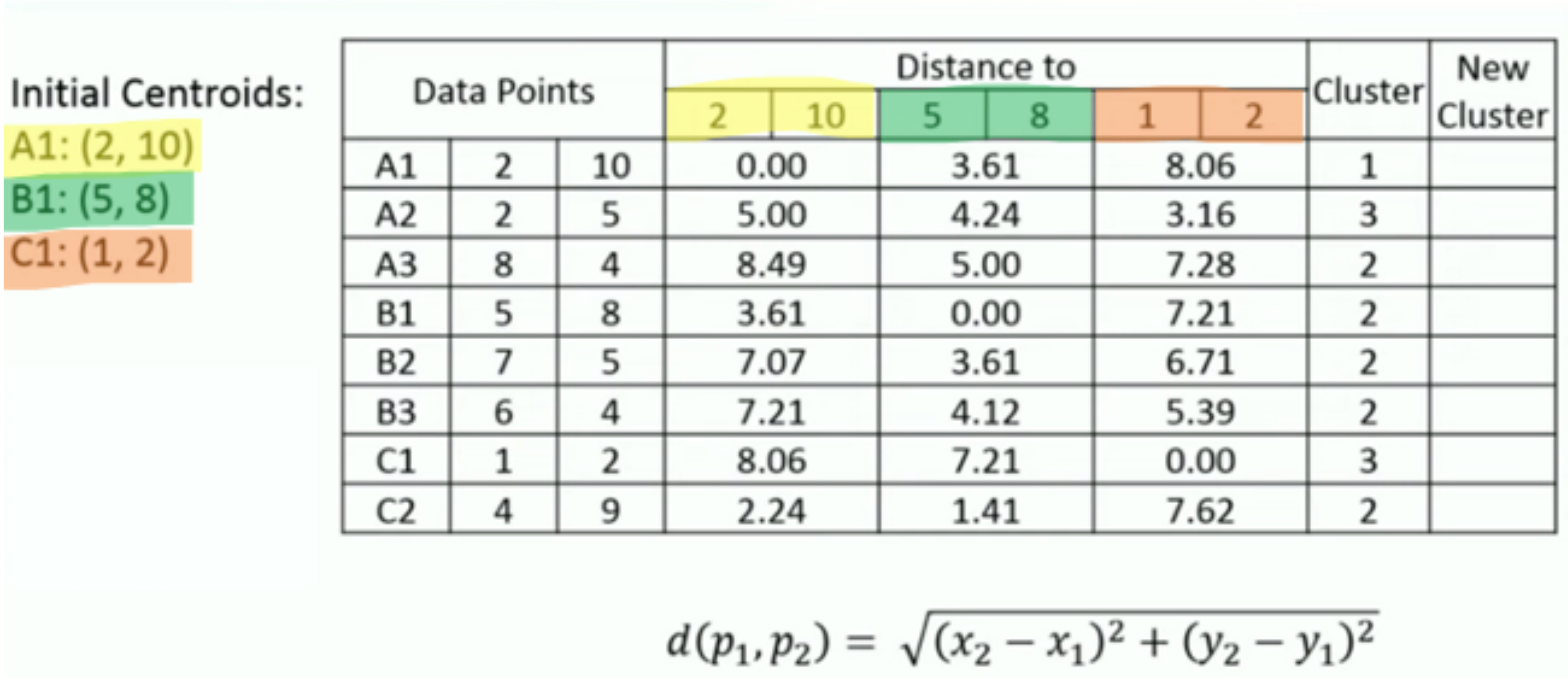
Example:



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)

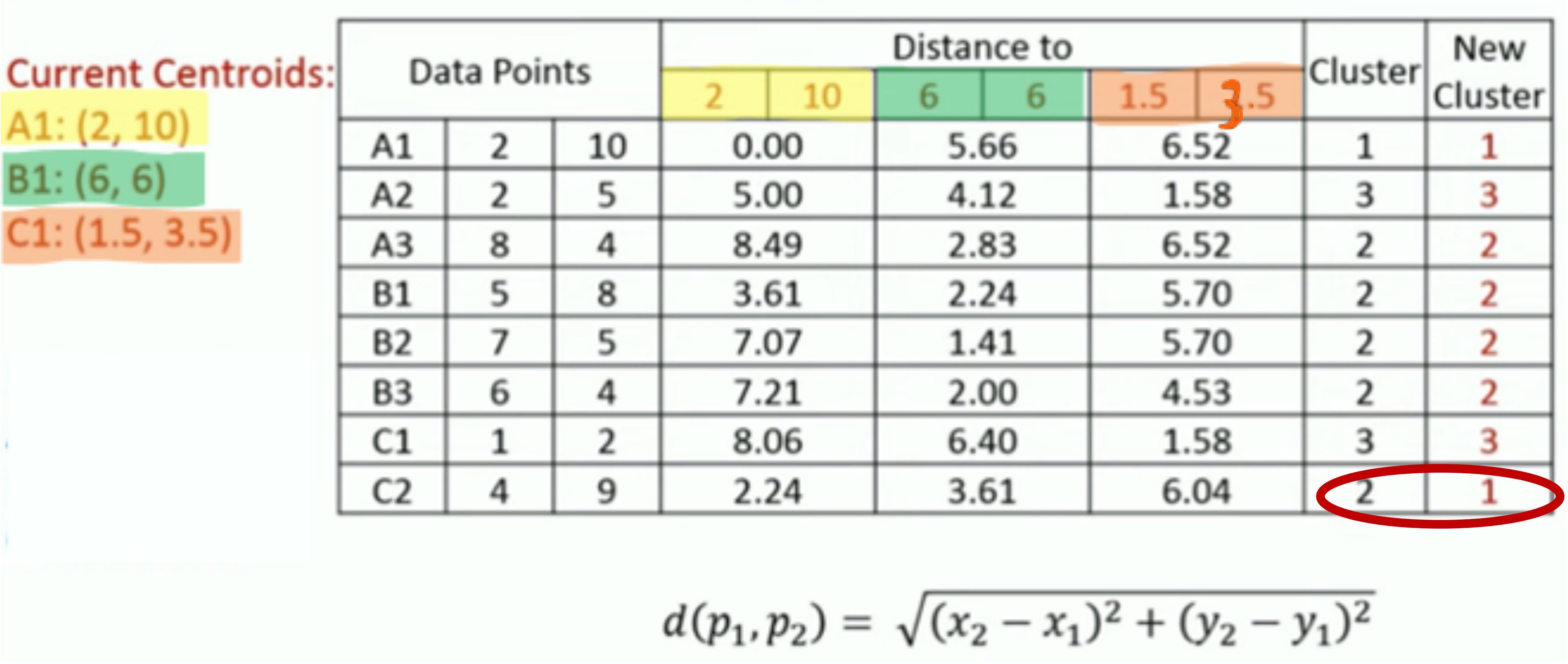
Data Points			Distance to			Cluster	New Cluster
			2 10	5 8	1 2		
A1	2	10	0.00	3.61	8.06	1	
A2	2	5	5.00	4.24	3.16	3	
A3	8	4	8.49	5.00	7.28	2	
B1	5	8	3.61	0.00	7.21	2	
B2	7	5	7.07	3.61	6.71	2	
B3	6	4	7.21	4.12	5.39	2	
C1	1	2	8.06	7.21	0.00	3	
C2	4	9	2.24	1.41	7.62	2	

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

Calculate new centroids(1st round):

- A1: (2,10)
- B1: ((8+5+7+6+4)/5 , (4+8+5+4+9)/5) = (6,6)
- C1: ((2+1)/2 , (5+2)/2) = (1.5,3.5)

New cluster based on the new centroids



Current Centroids:

A1: (2, 10)

B1: (6, 6)

C1: (1.5, 3.5)

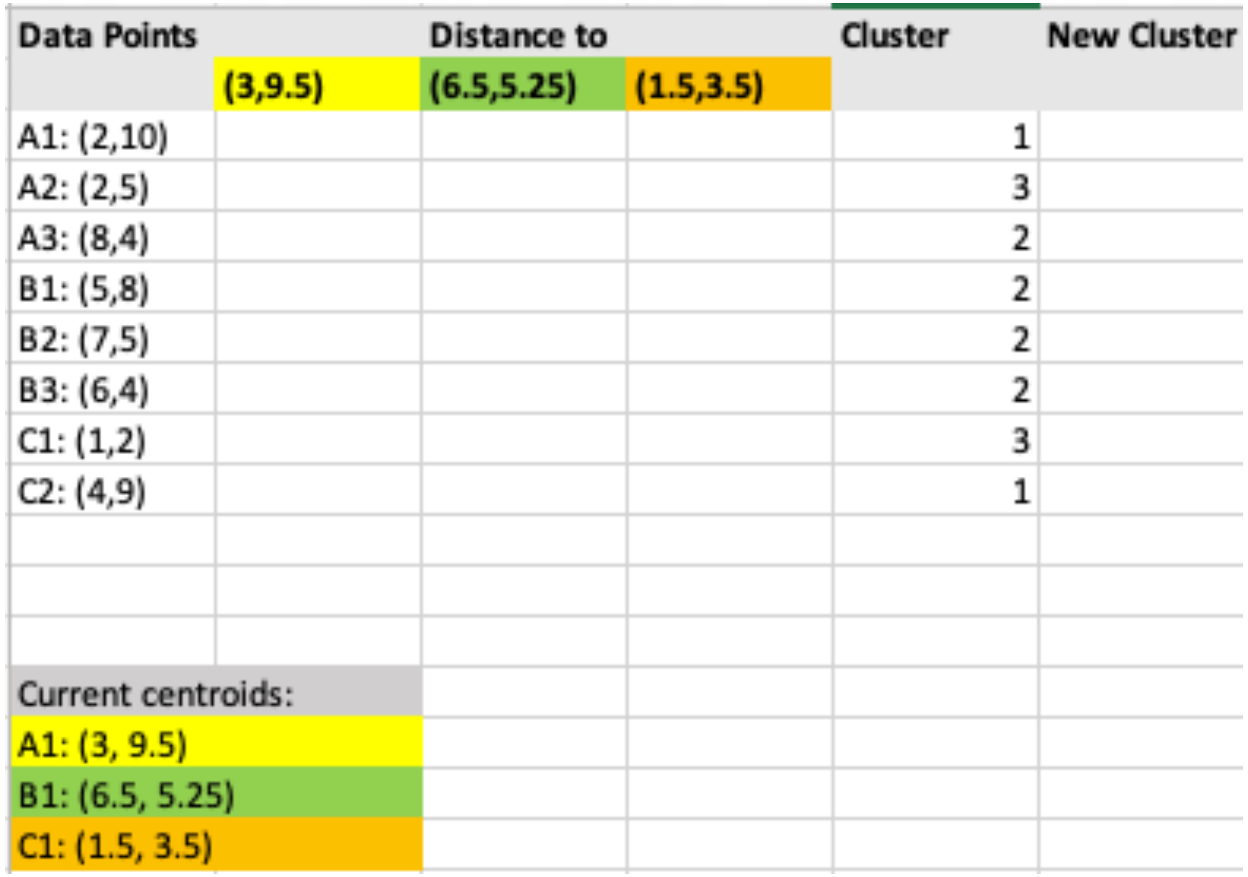
Data Points			Distance to			Cluster	New Cluster
			2 10	6 6	1.5 3.5		
A1	2	10	0.00	5.66	6.52	1	1
A2	2	5	5.00	4.12	1.58	3	3
A3	8	4	8.49	2.83	6.52	2	2
B1	5	8	3.61	2.24	5.70	2	2
B2	7	5	7.07	1.41	5.70	2	2
B3	6	4	7.21	2.00	4.53	2	2
C1	1	2	8.06	6.40	1.58	3	3
C2	4	9	2.24	3.61	6.04	2	1

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

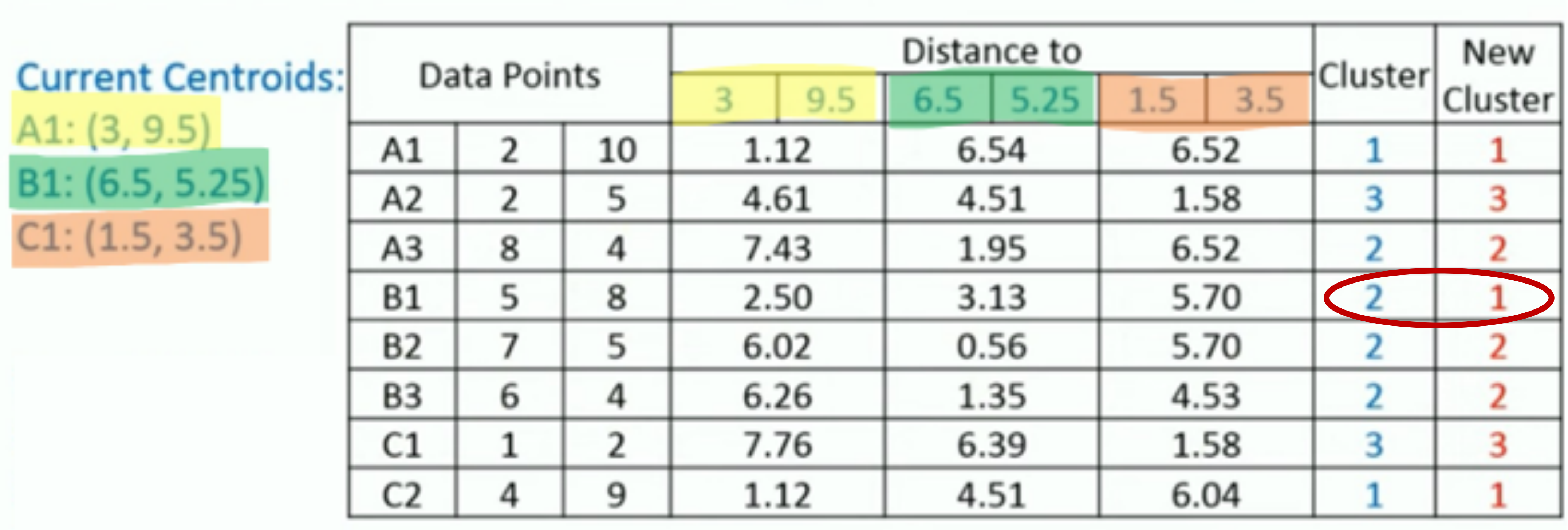
Calculate the new centroids (2nd round):

- A1: ((2+4)/2 , (10+9)/2) = (3,9.5)
- B1: ((8+5+7+6)/4 , (4+8+5+4)/4) = (6.5,5.25)
- C1: ((2+1)/2 , (5+2)/2) = (1.5,3.5)

Fill out the following chart:



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



Current Centroids:

A1: (3, 9.5)

B1: (6.5, 5.25)

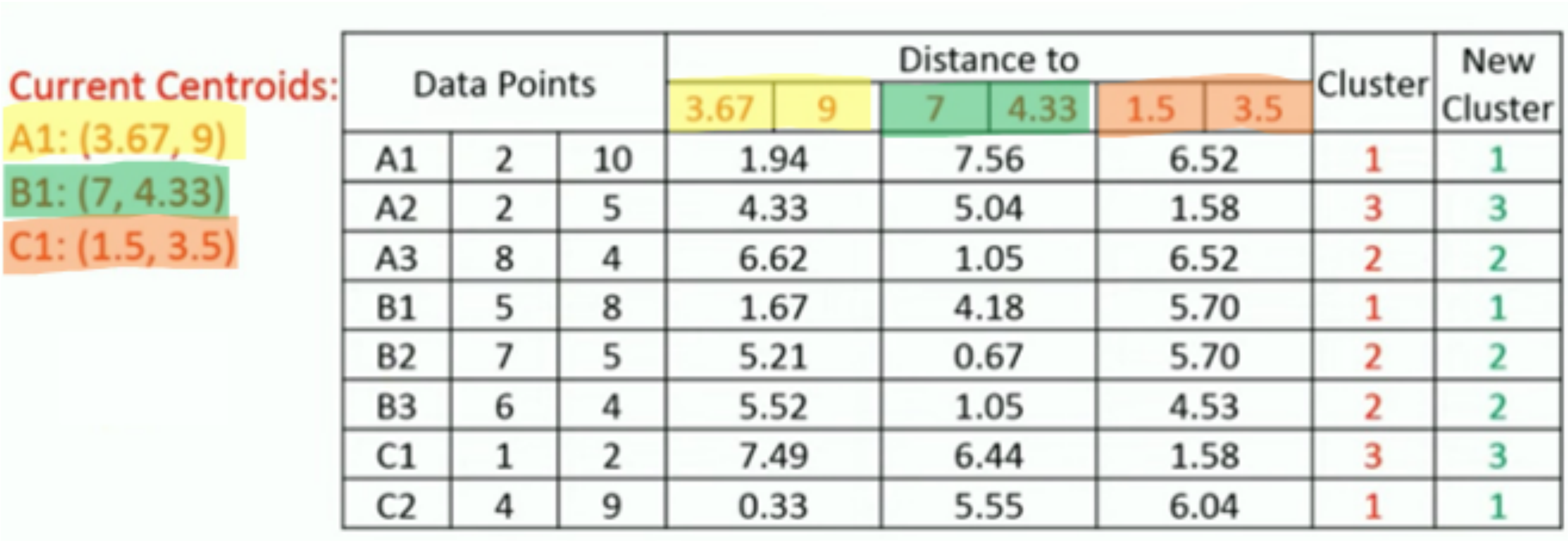
C1: (1.5, 3.5)

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

Calculate the new centroids (3rd round):

- A1: (3.67,9)
- B1: (7,4.33)
- C1: (1.5,3.5)

Last step:



Current Centroids:

A1: (3.67, 9)

B1: (7, 4.33)

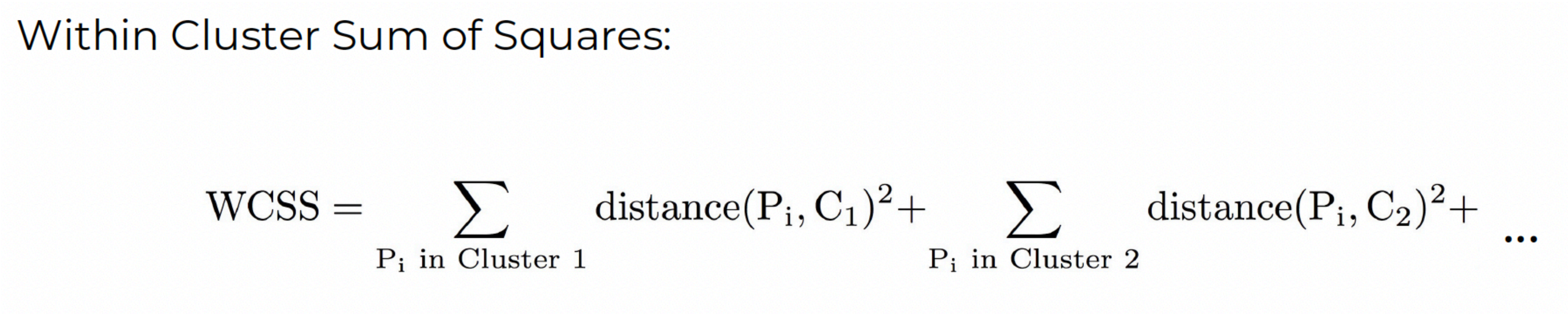
C1: (1.5, 3.5)

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

how to choose the optimal number of clusters?

- We use the elbow method for that.

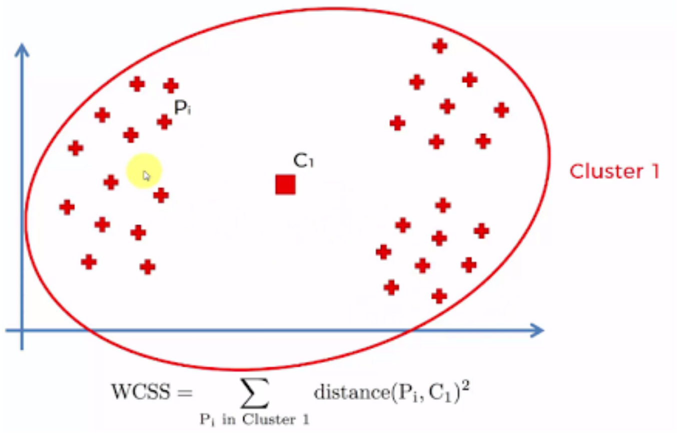
  The formula:



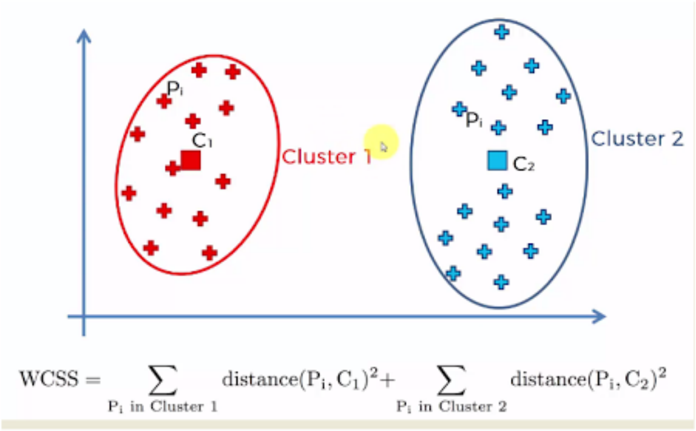
Within Cluster Sum of Squares:

$$\mathrm{WCSS} = \sum_{\mathrm{P_i\ in\ Cluster\ 1}} \mathrm{distance}(\mathrm{P_i}, \mathrm{C_1})^2 + \sum_{\mathrm{P_i\ in\ Cluster\ 2}} \mathrm{distance}(\mathrm{P_i}, \mathrm{C_2})^2 + \ \dots$$

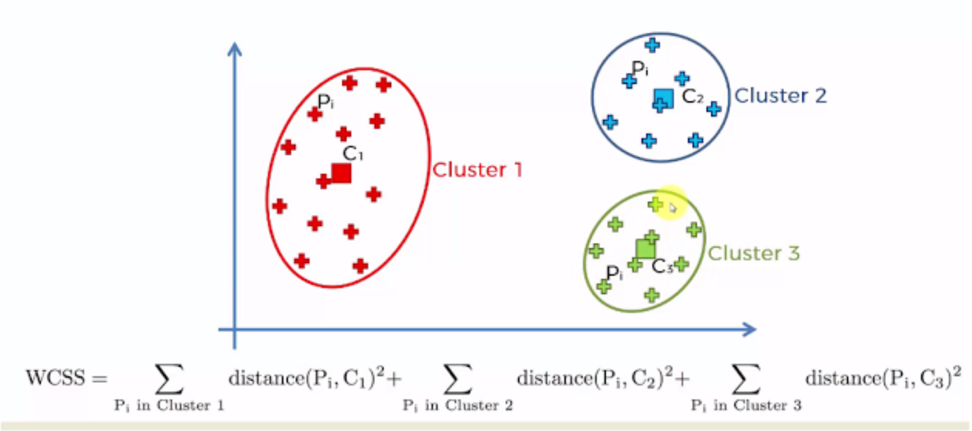
The elbow method:



$$\mathrm{WCSS} = \sum_{P_i \text{ in Cluster 1}} \mathrm{distance}(P_i, C_1)^2$$

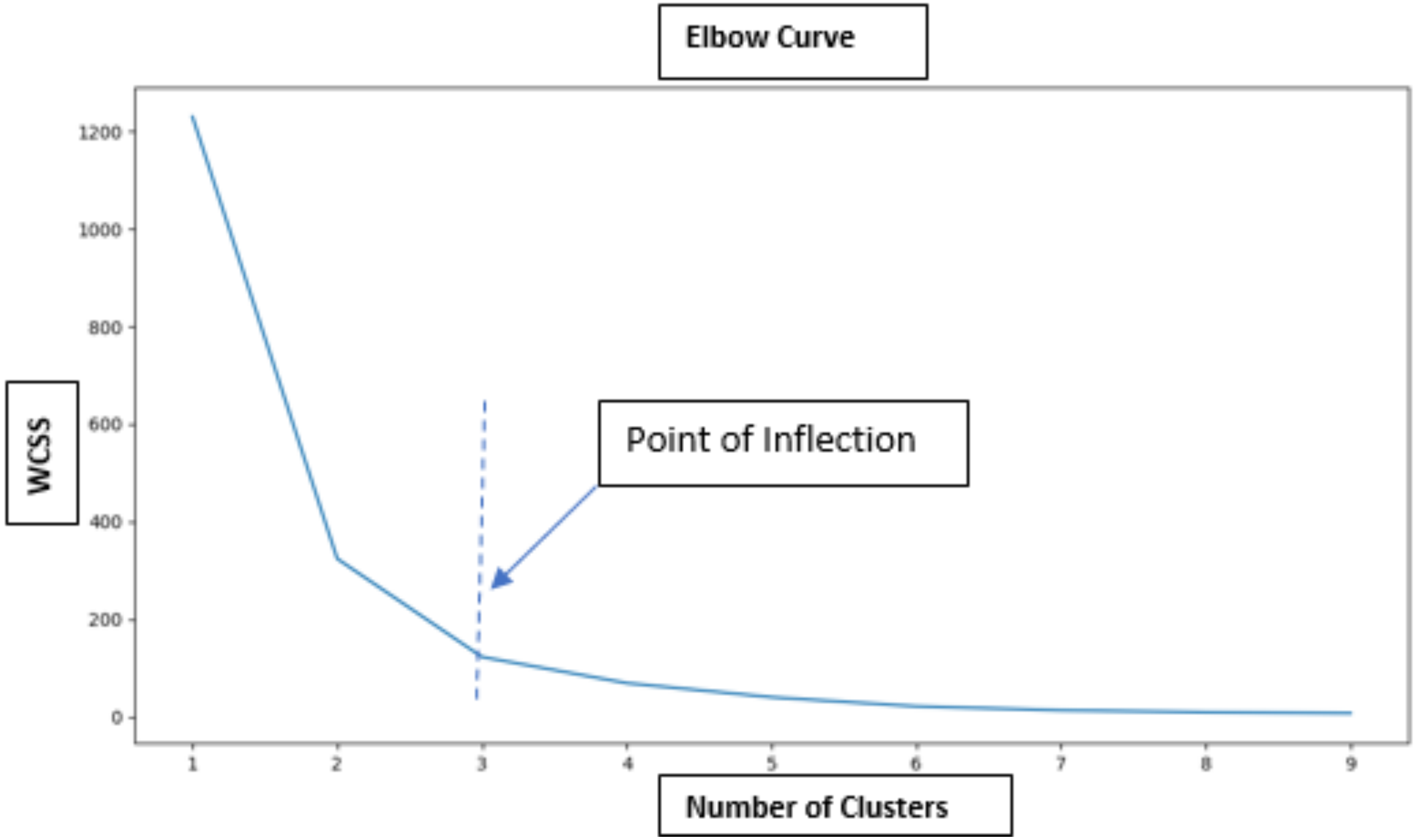


$$\mathrm{WCSS} = \sum_{P_i \text{ in Cluster 1}} \mathrm{distance}(P_i, C_1)^2 + \sum_{P_i \text{ in Cluster 2}} \mathrm{distance}(P_i, C_2)^2$$

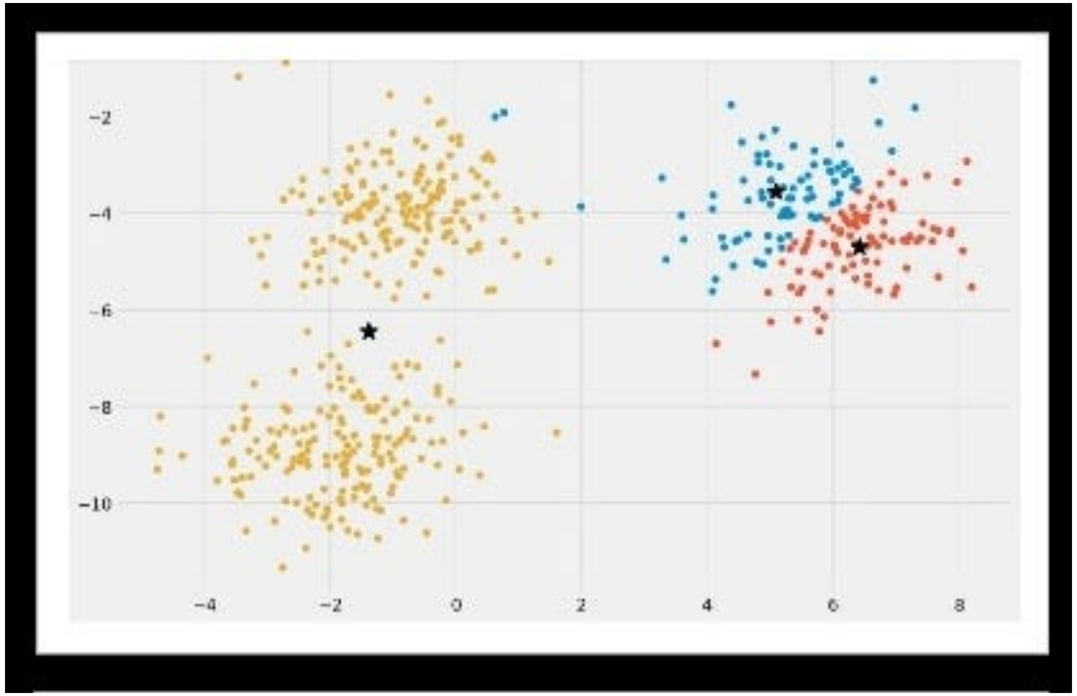


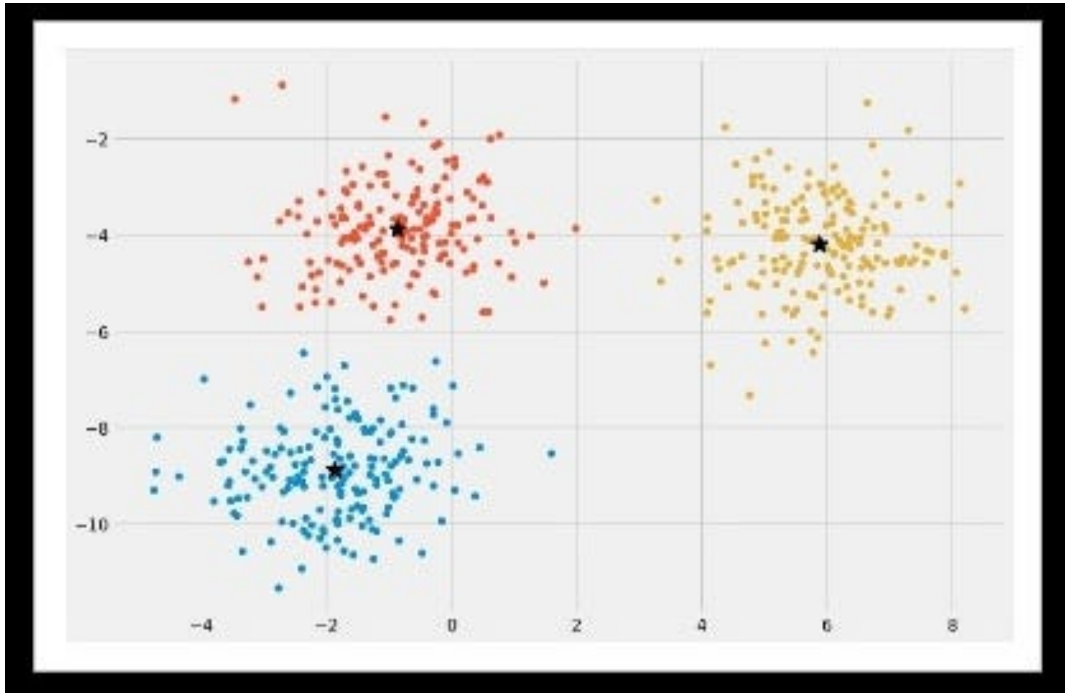
$$\mathrm{WCSS} = \sum_{P_i \text{ in Cluster 1}} \mathrm{distance}(P_i, C_1)^2 + \sum_{P_i \text{ in Cluster 2}} \mathrm{distance}(P_i, C_2)^2 + \sum_{P_i \text{ in Cluster 3}} \mathrm{distance}(P_i, C_3)^2$$

The elbow method:

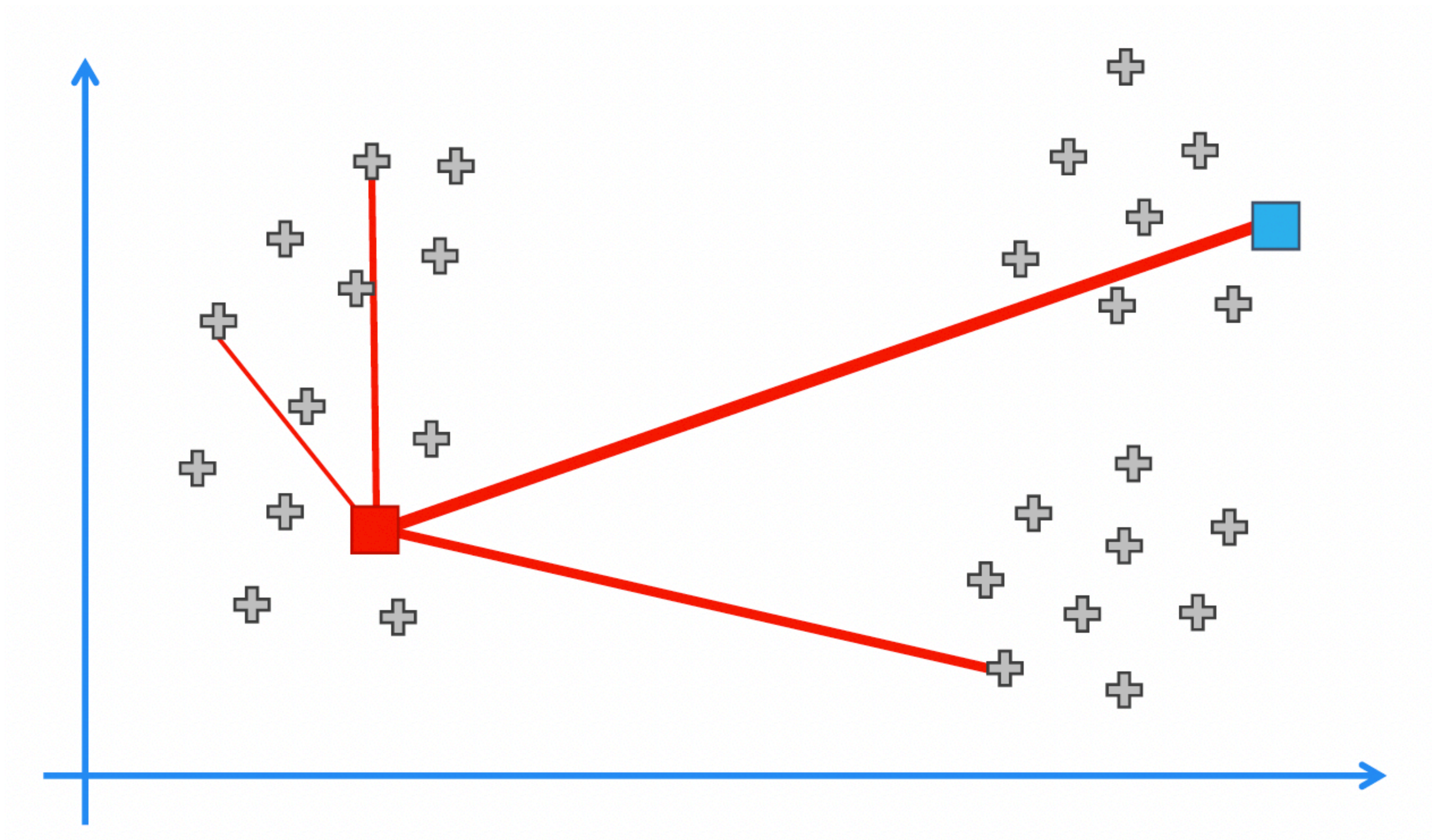


How to choose the centroids: K-Means++

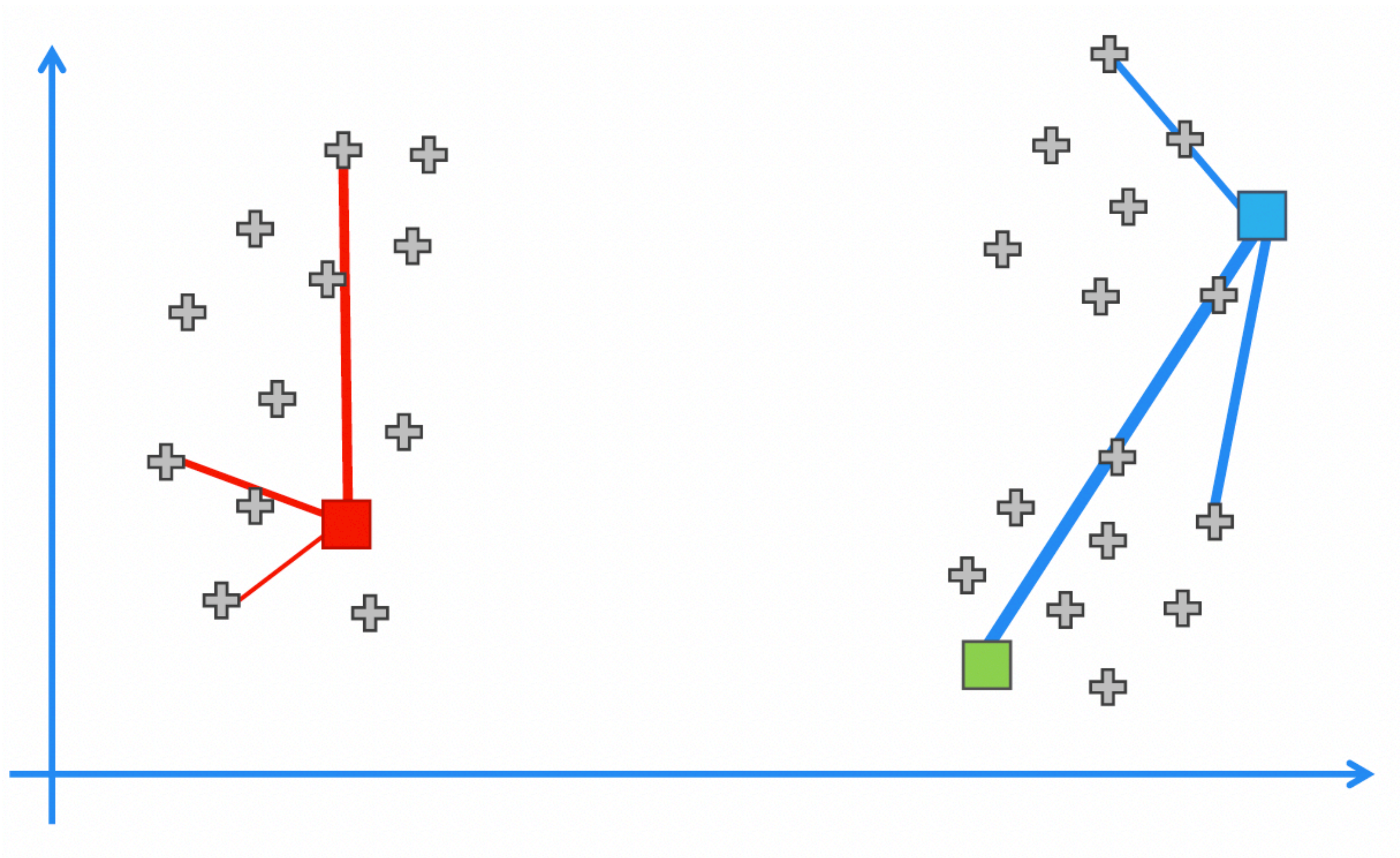




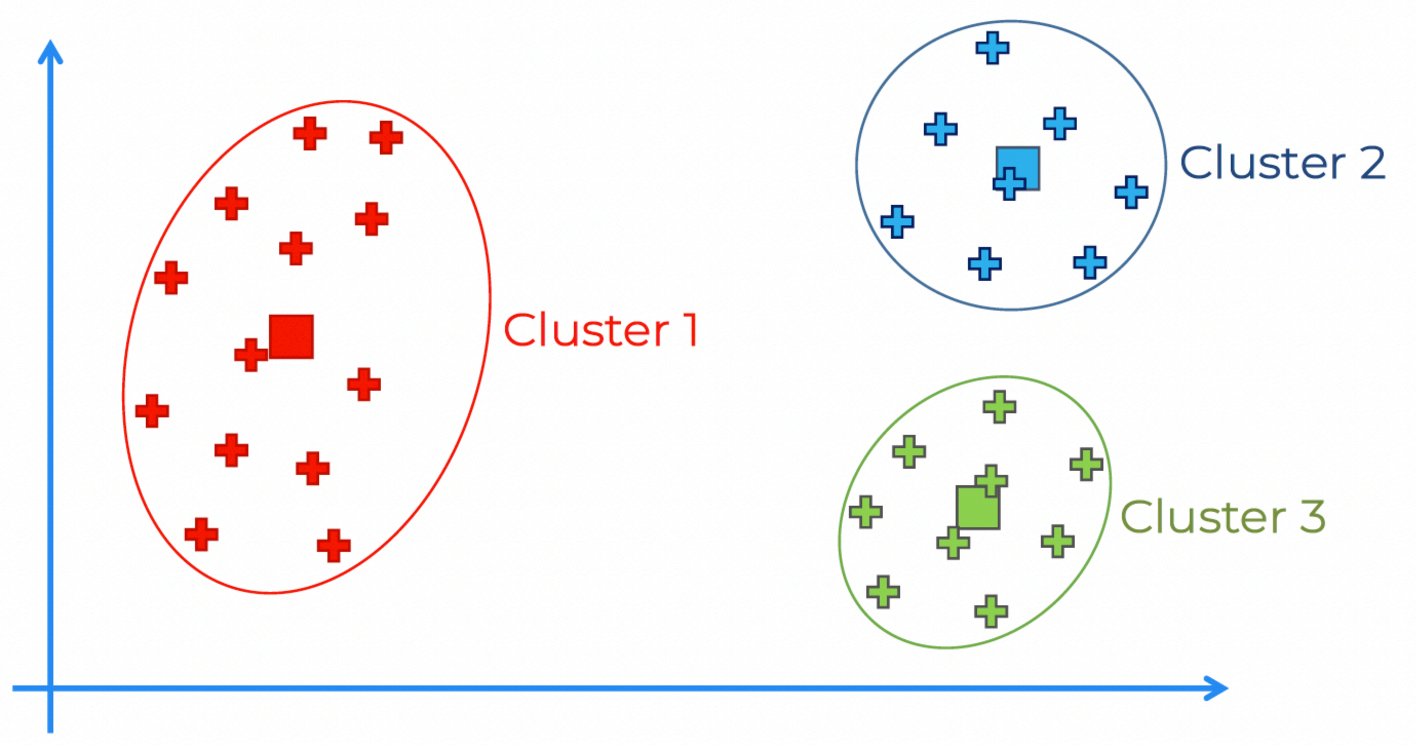
K-Means++



K-Means++



K-Means++



K-Means++ Initialization Algorithm:

- Step1: choose first centroid at random among data points
- Step2: For each of the remaining data points compute the distance to the nearest out of already selected centroids
- Step3: choose next centroid among remaining data points using weighted random selected – weighted by D^2
- Step4: repeat steps 2 and 3 until all k centroids have been selected
- Step5: proceed with standard k-means clustering

K-Means++

- It doesn’t guarantee that there will be no issue in terms of initialization; because it does starts at random.
- But because it’s done in a weight random fashion, the chances of that happening is much lower; and this does help with the problems we saw with the random initialization trap.