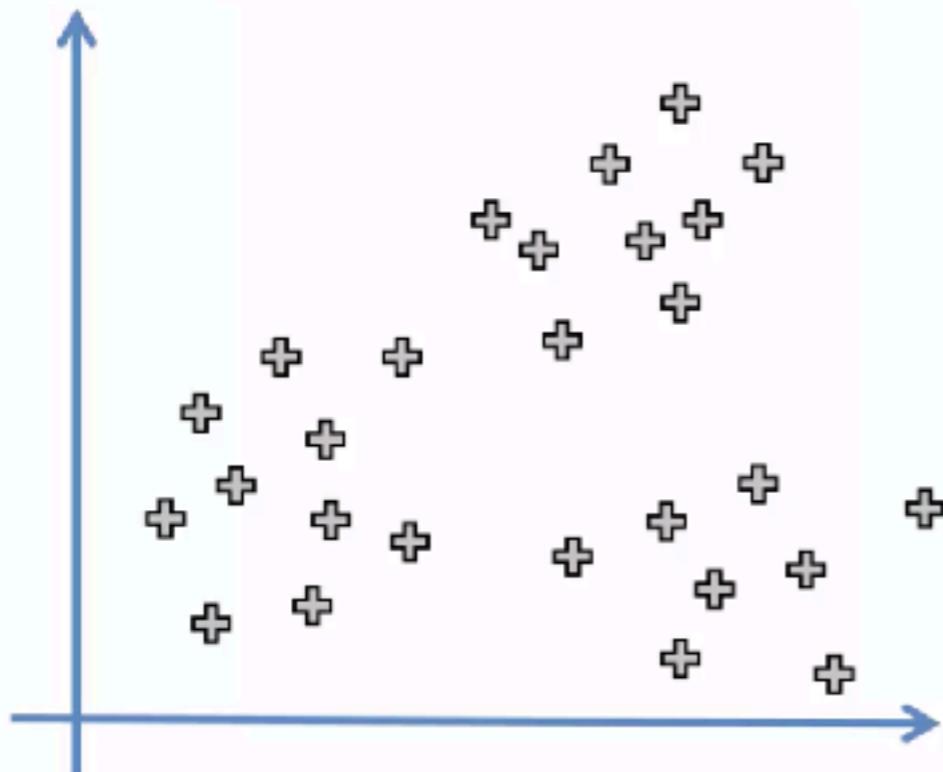


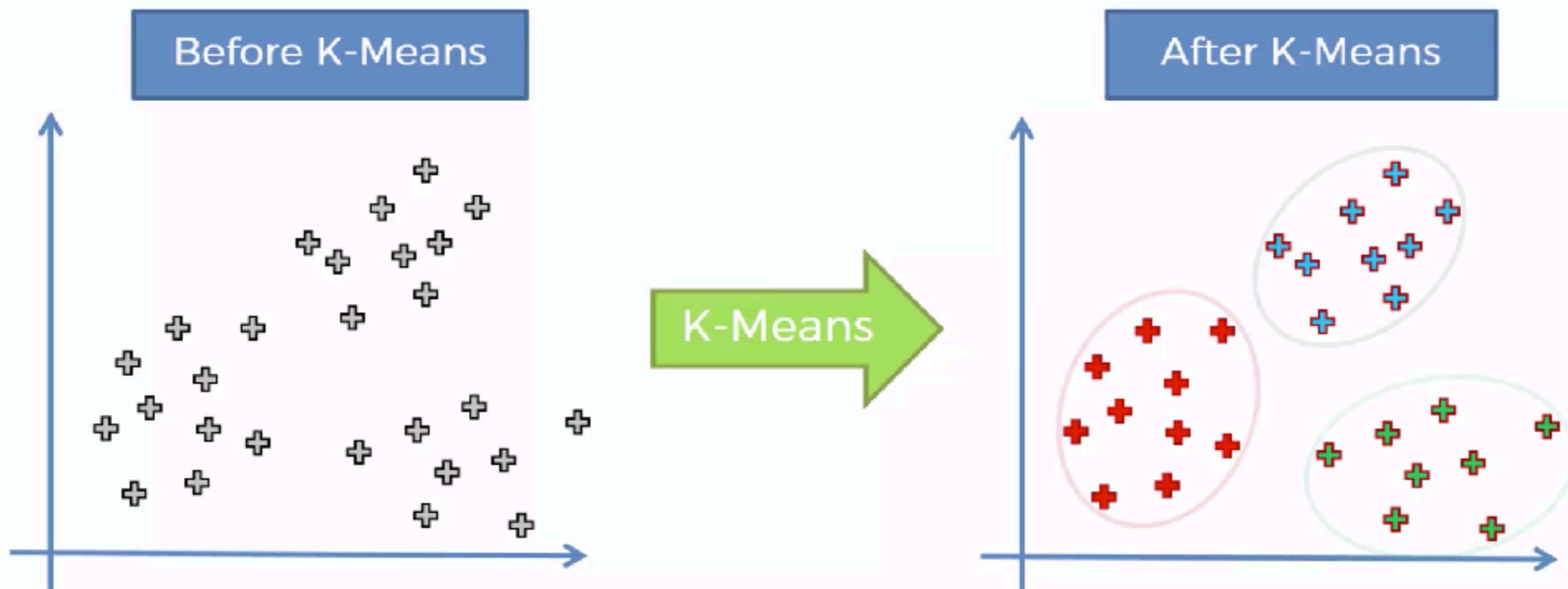
K-Means Intuition: Understanding K-Means

What K-Means does for you

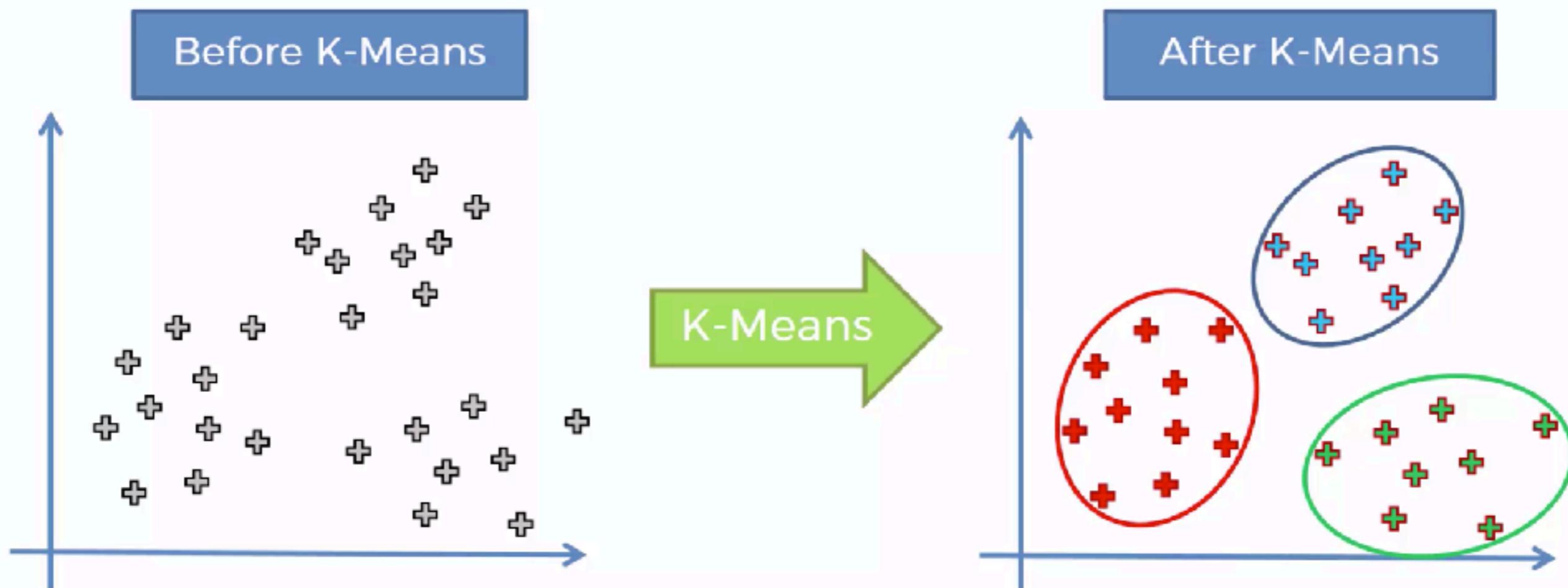
Before K-Means



What K-Means does for you



What K-Means does for you



How did it do that ?

STEP 1: Choose the number K of clusters



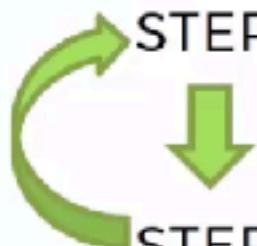
STEP 2: Select at random K points, the centroids (not necessarily from your dataset)



STEP 3: Assign each data point to the closest centroid ➔ That forms K clusters



STEP 4: Compute and place the new centroid of each cluster



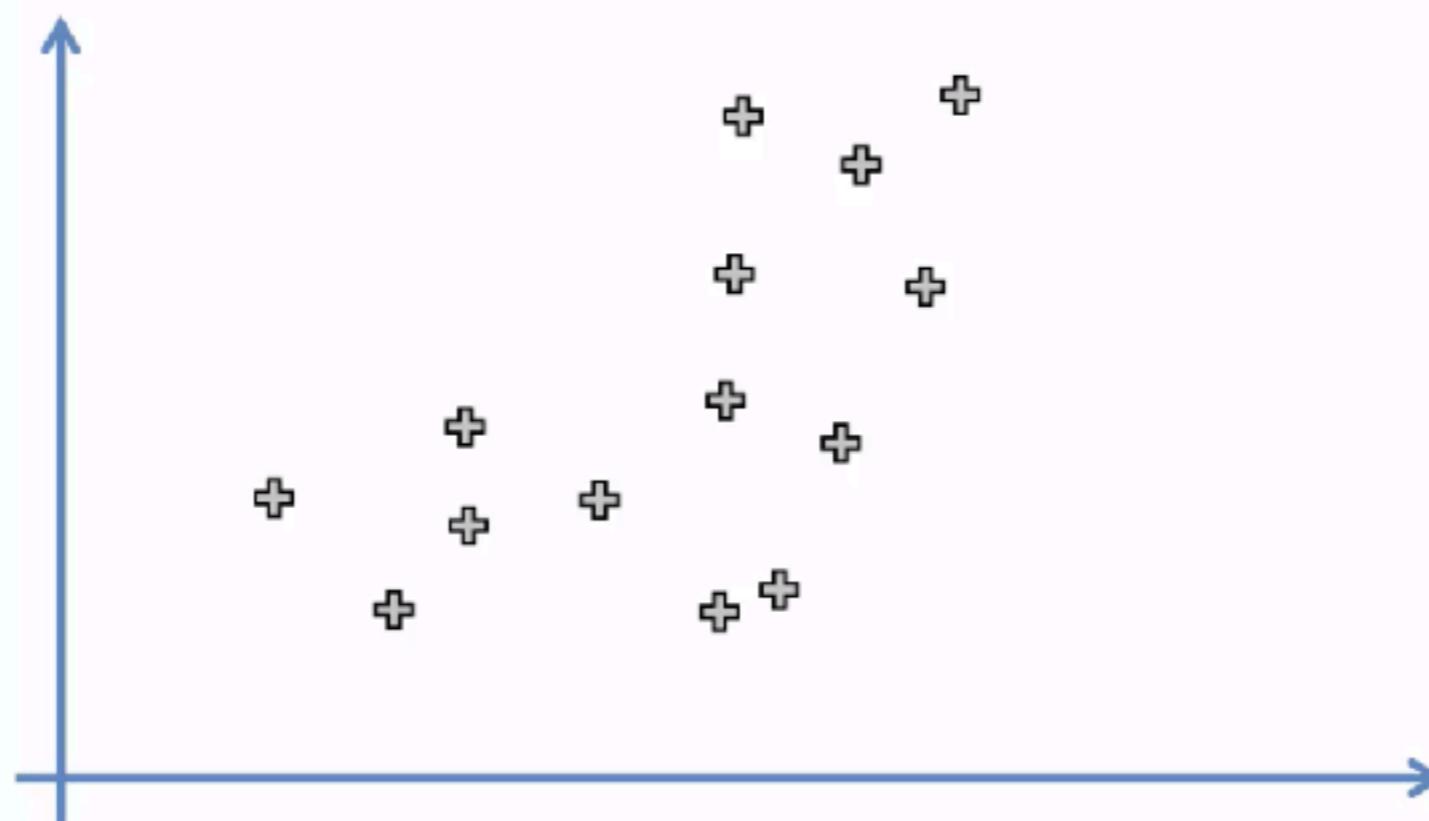
STEP 5: Reassign each data point to the new closest centroid.

If any reassignment took place, go to STEP 4, otherwise go to FIN.

Your Model is Ready

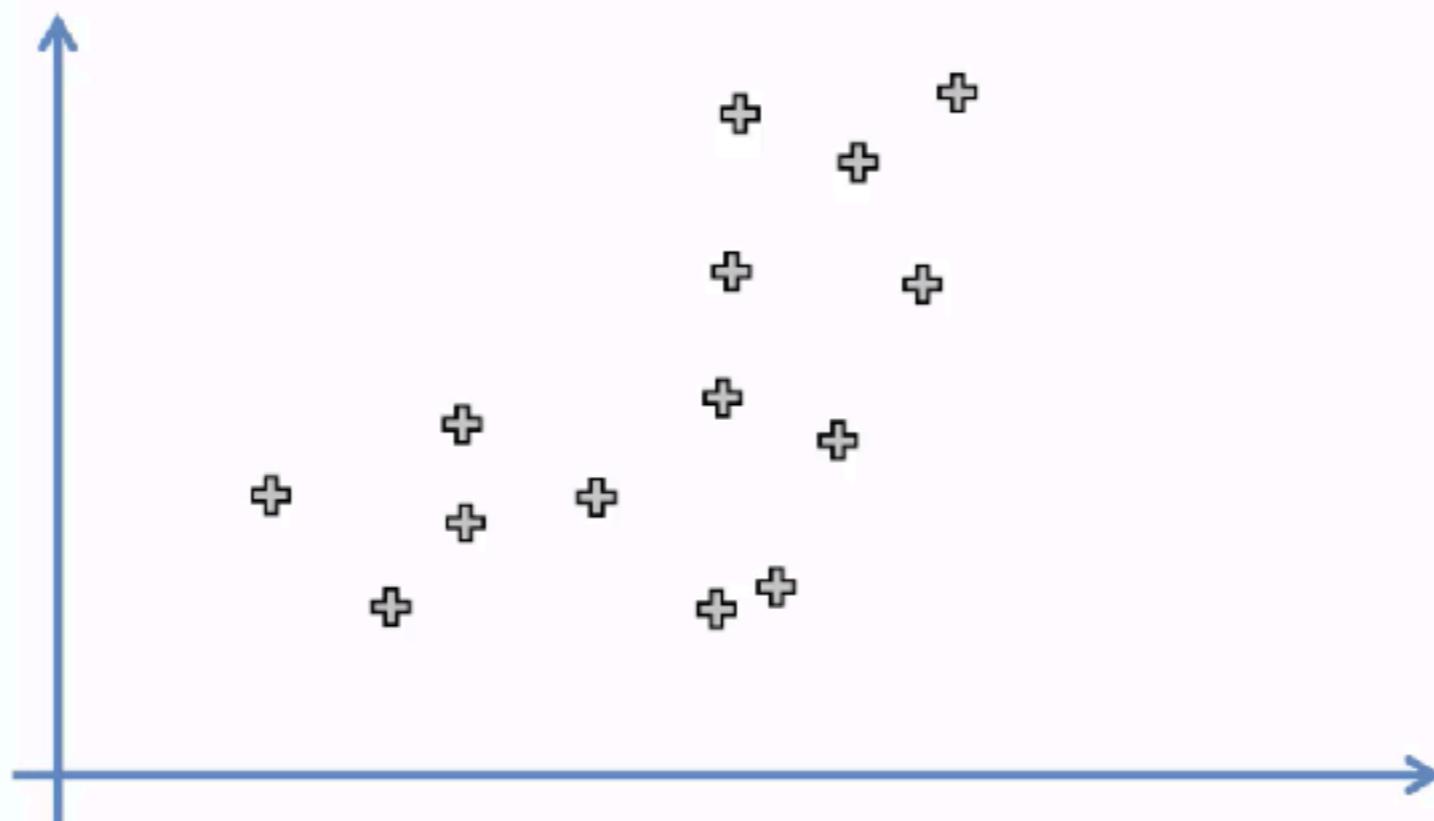
K-Means algorithm

STEP 1: Choose the number K of clusters: K = 2



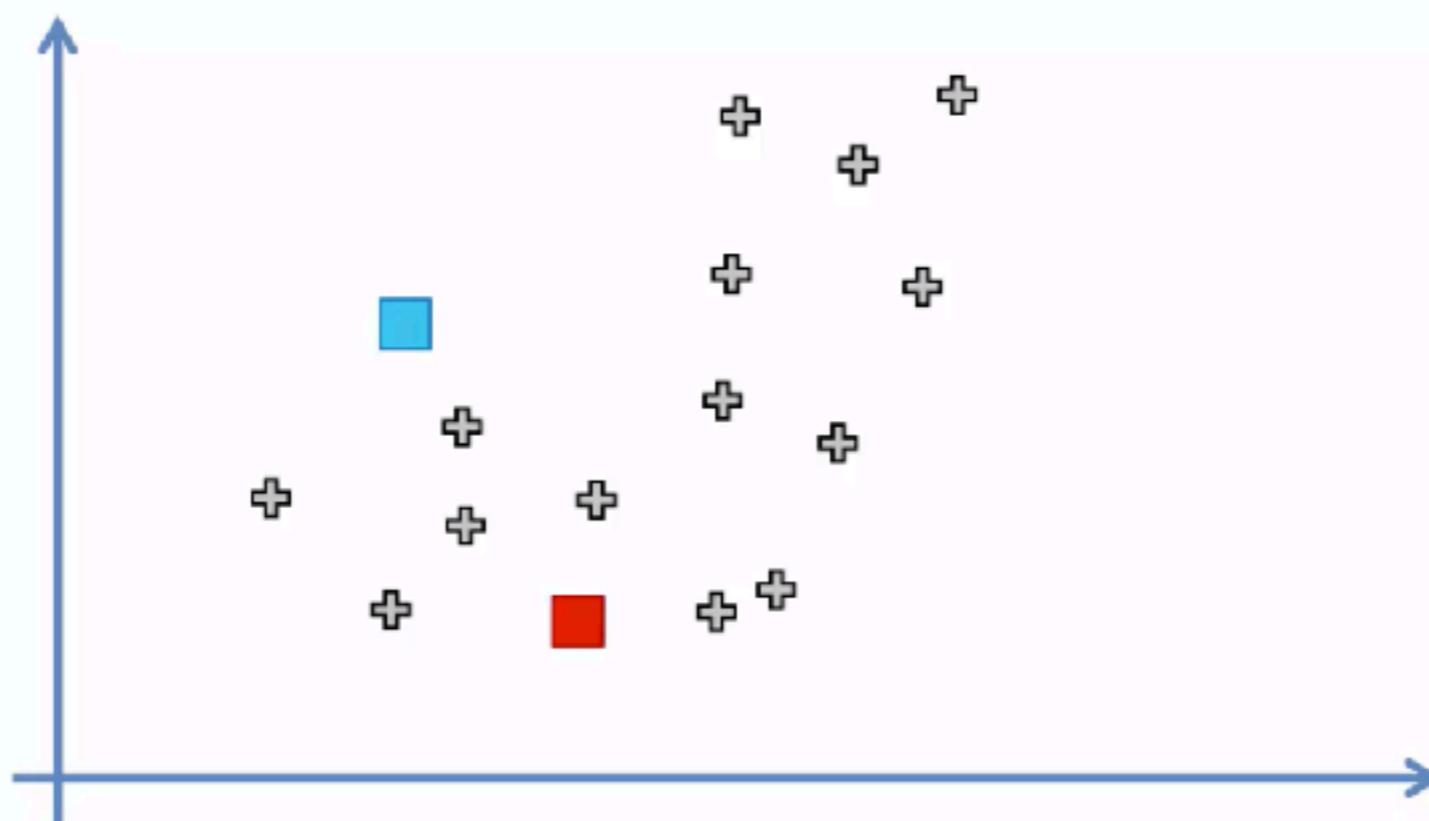
K-Means algorithm

STEP 2: Select at random K points, the centroids (not necessarily from your dataset)



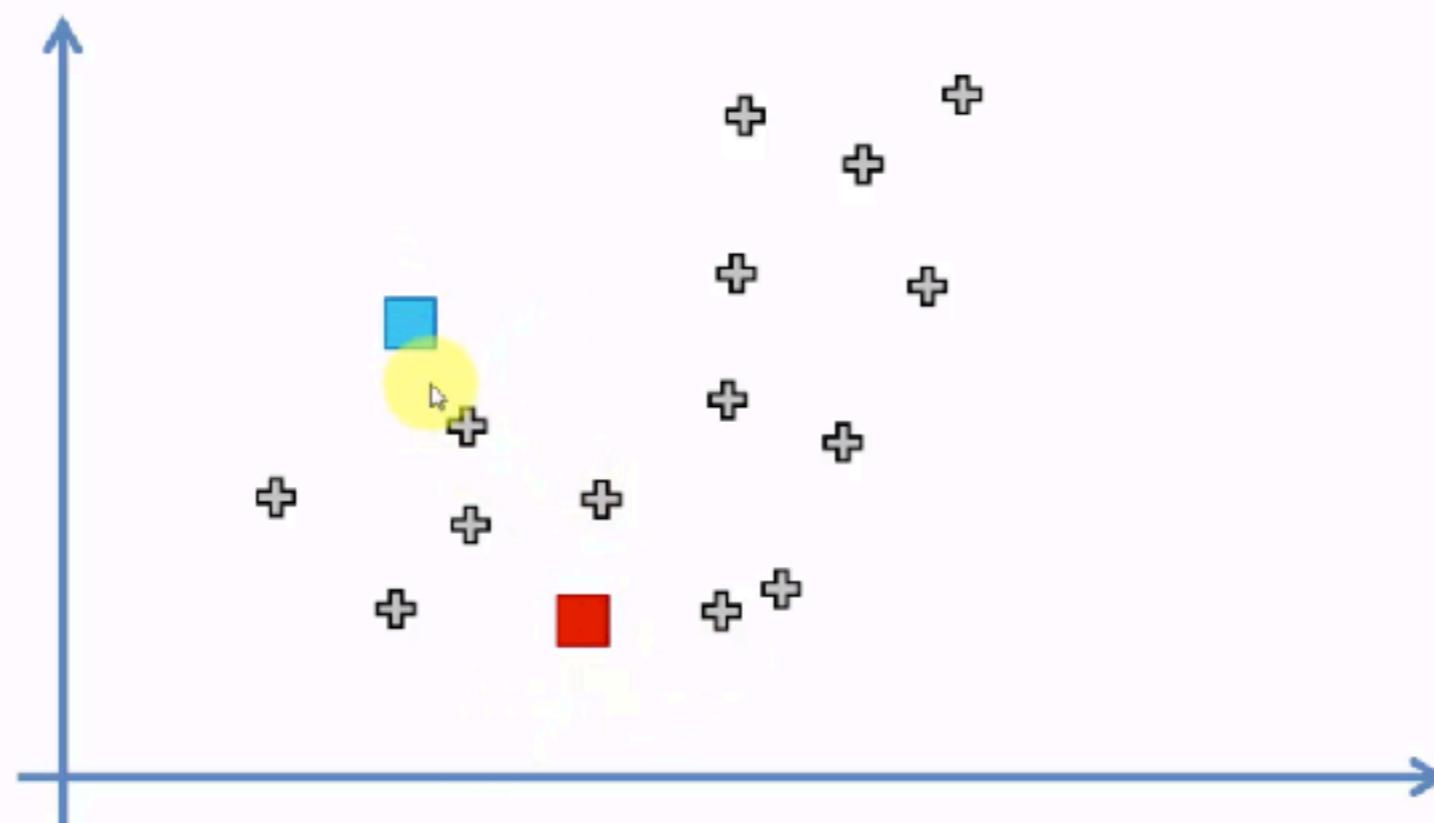
K-Means algorithm

STEP 2: Select at random K points, the centroids (not necessarily from your dataset)



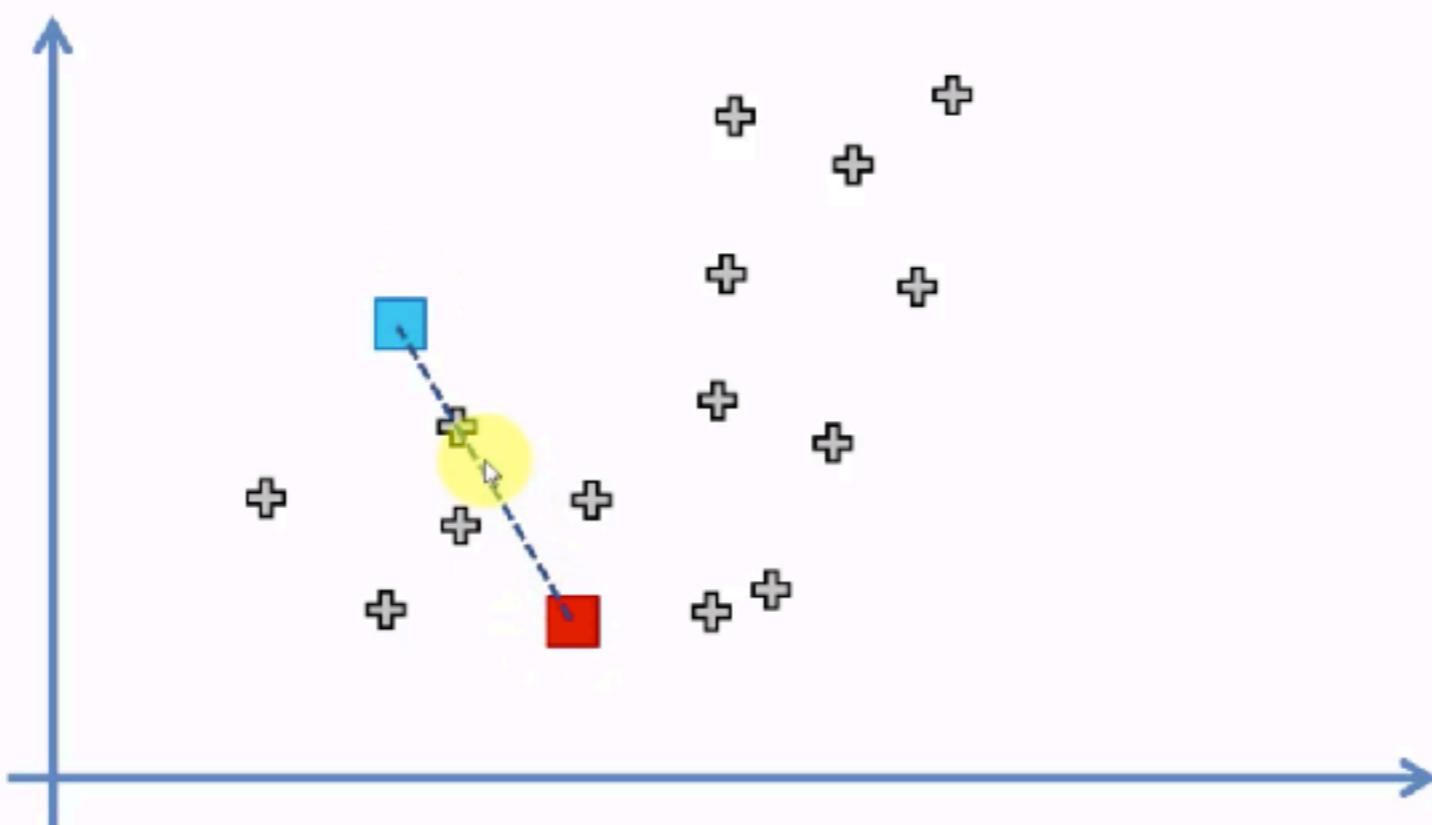
K-Means algorithm

STEP 3: Assign each data point to the closest centroid → That forms K clusters



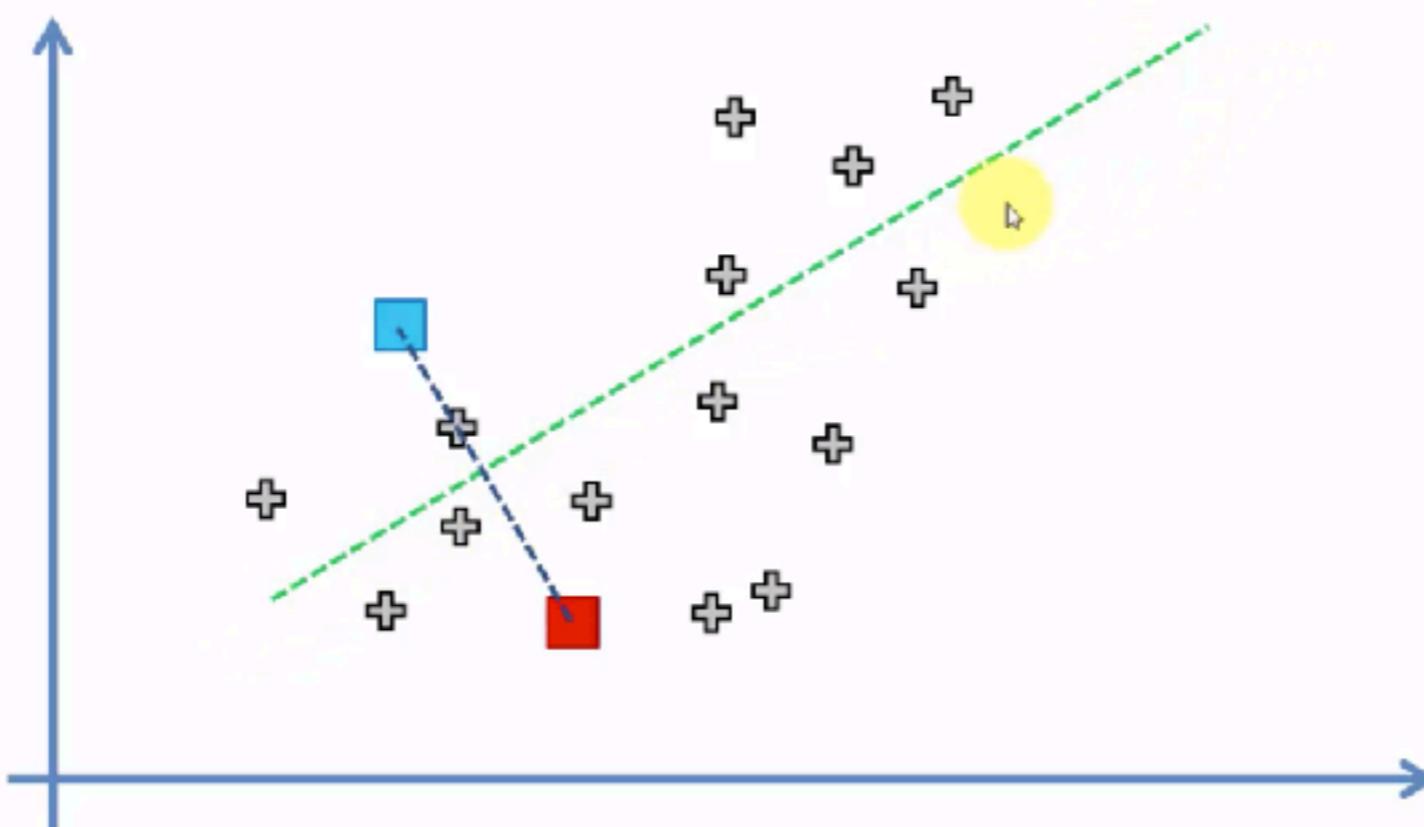
K-Means algorithm

STEP 3: Assign each data point to the closest centroid → That forms K clusters



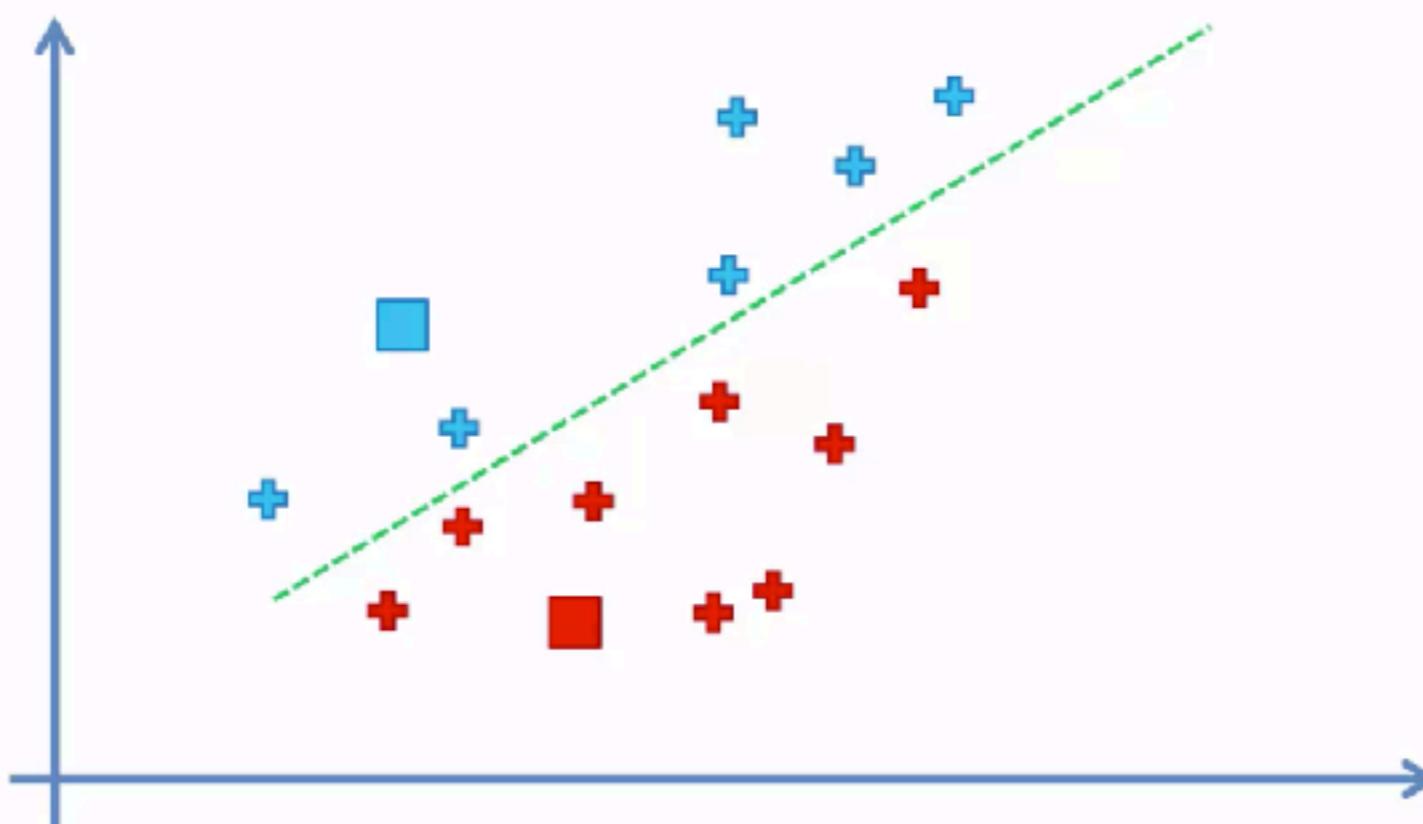
K-Means algorithm

STEP 3: Assign each data point to the closest centroid → That forms K clusters



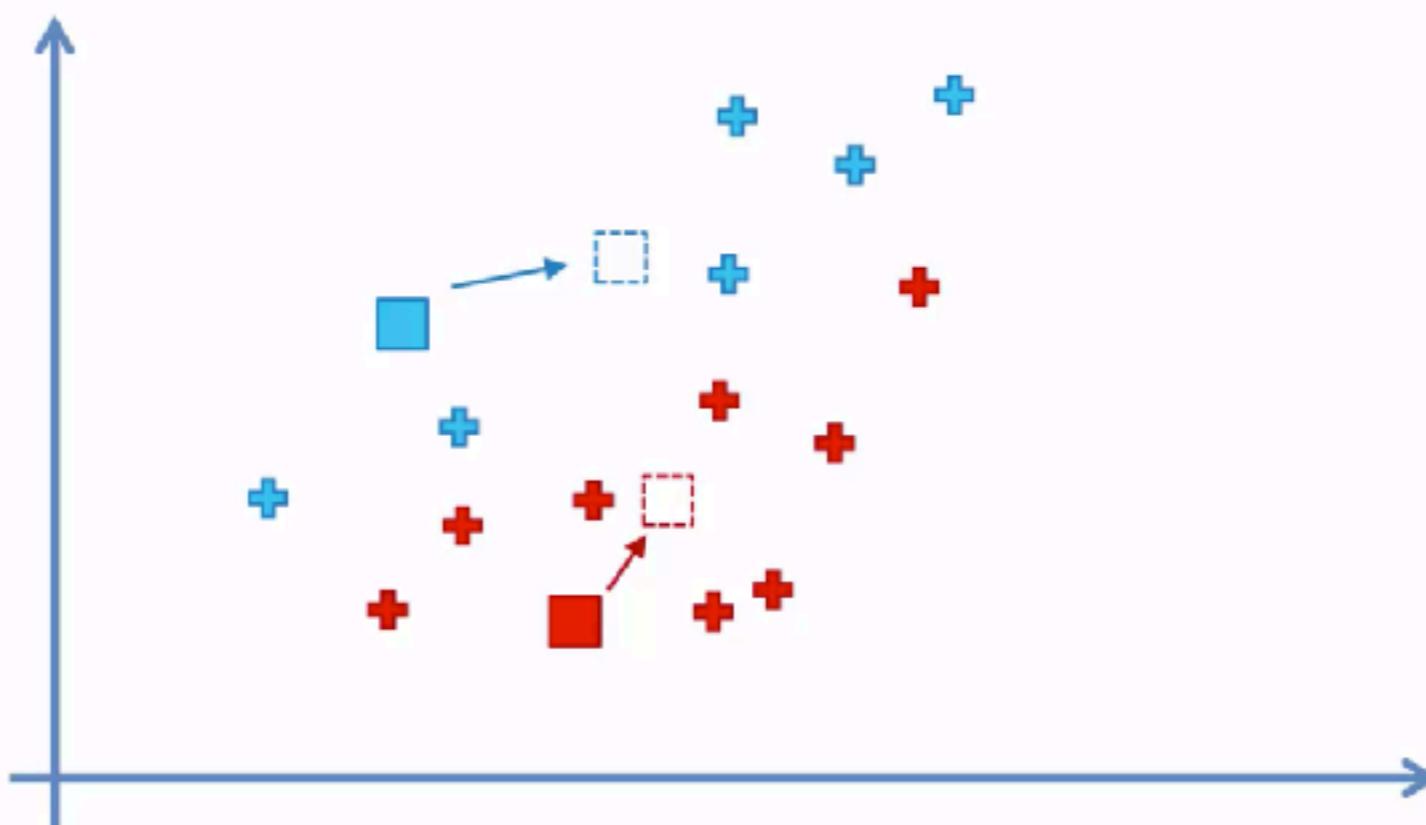
K-Means algorithm

STEP 3: Assign each data point to the closest centroid → That forms K clusters



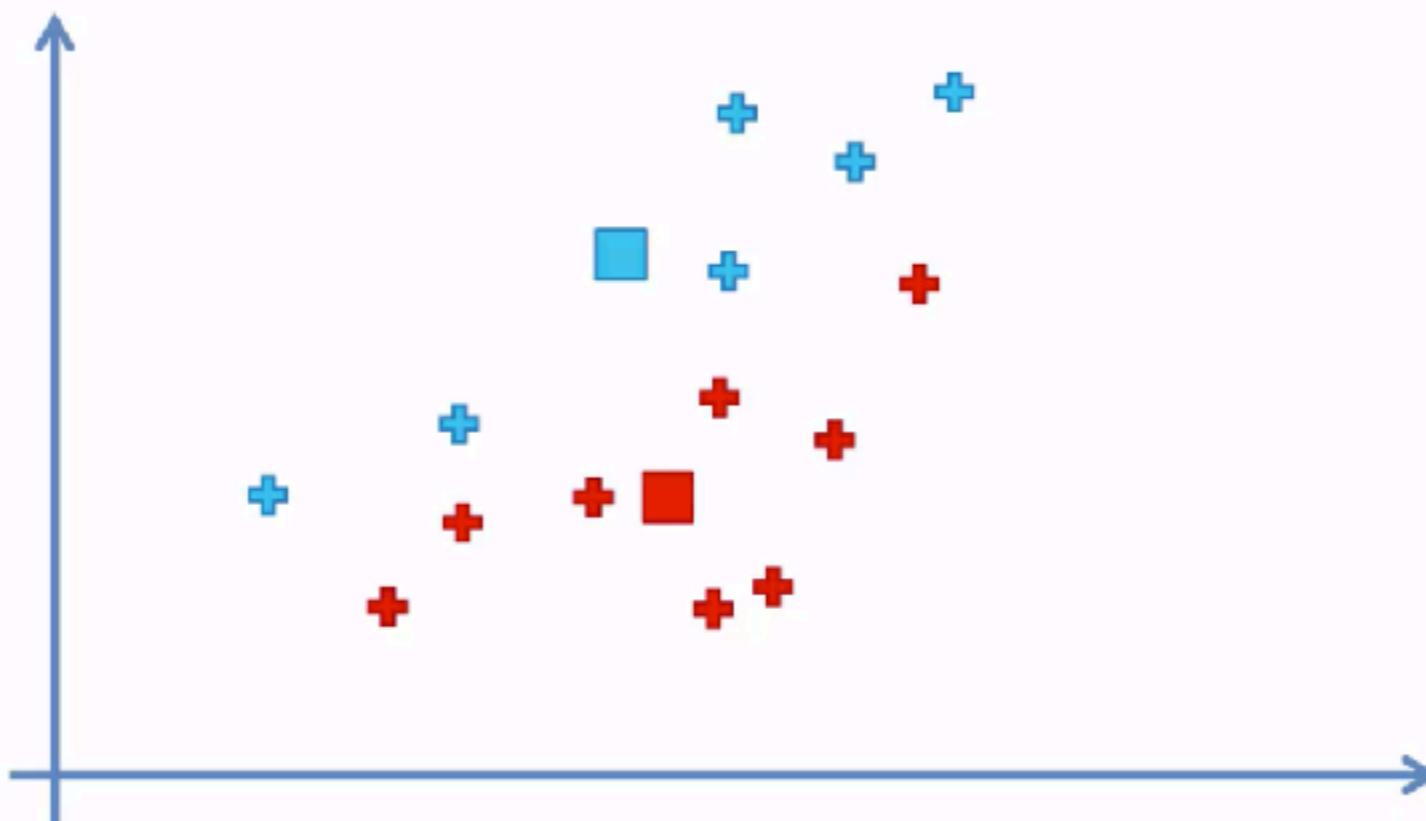
K-Means algorithm

STEP 4: Compute and place the new centroid of each cluster



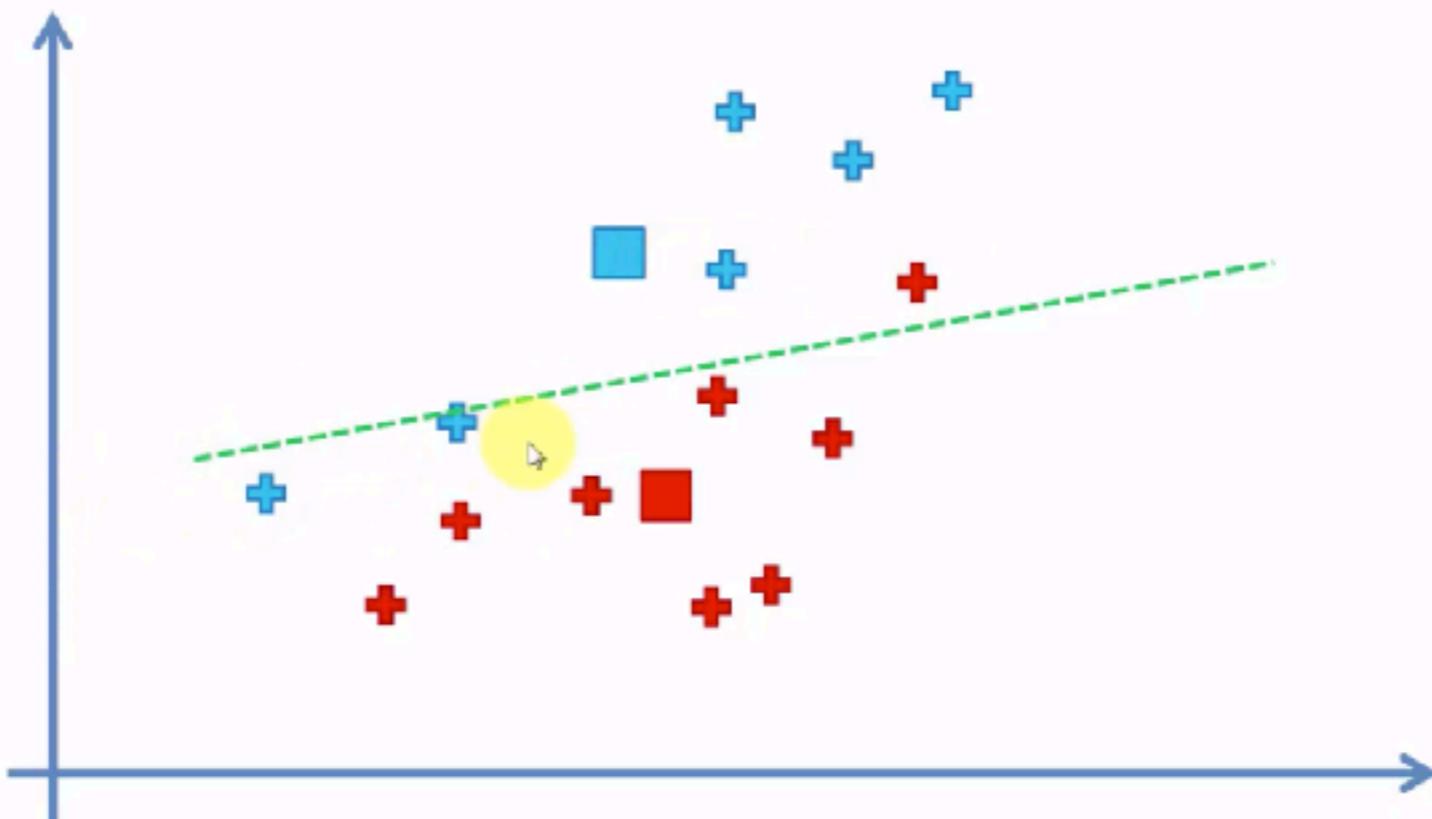
K-Means algorithm

STEP 5: Reassign each data point to the new closest centroid.
If any reassignment took place, go to STEP 4, otherwise go to FIN.



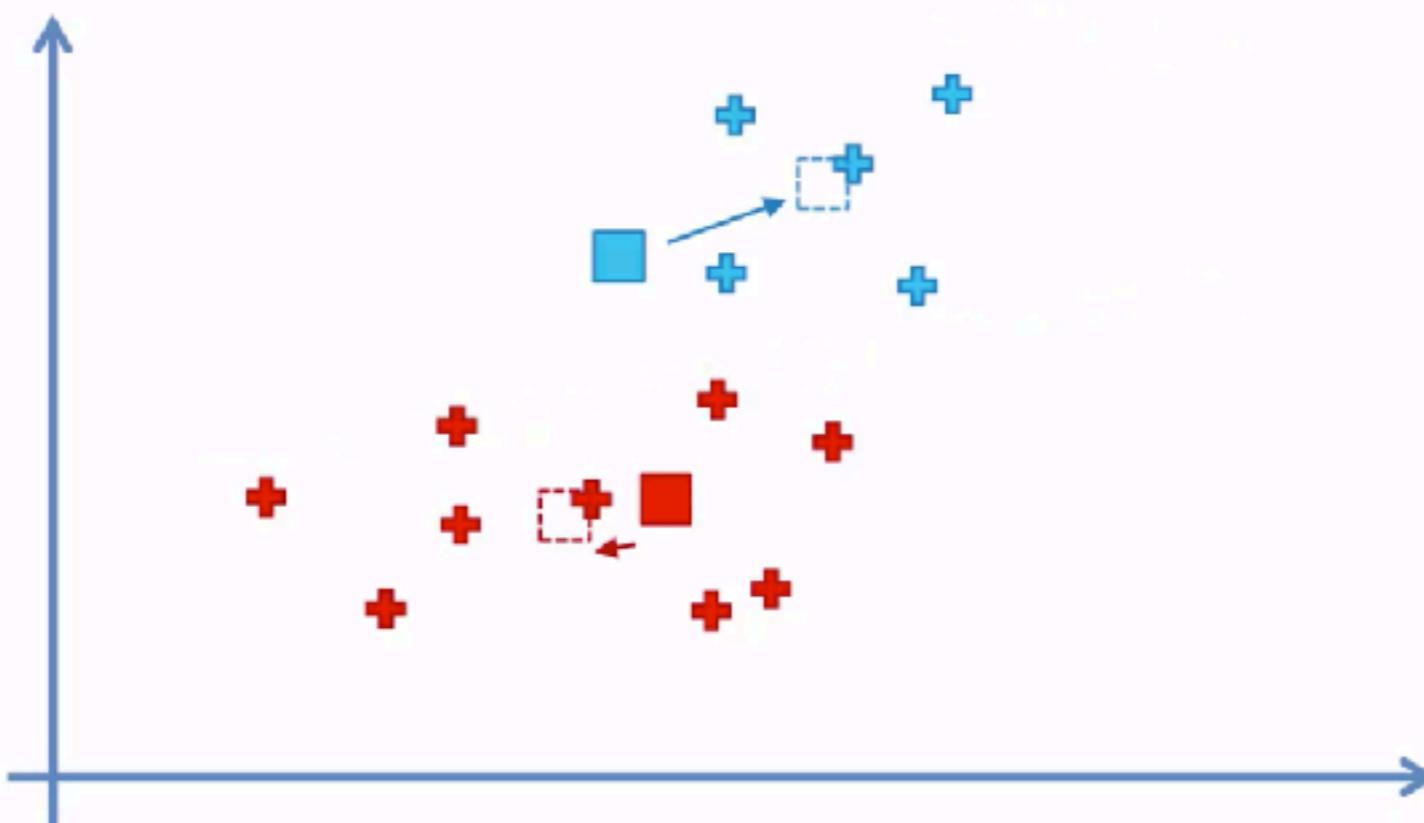
K-Means algorithm

STEP 5: Reassign each data point to the new closest centroid.
If any reassignment took place, go to STEP 4, otherwise go to FIN.



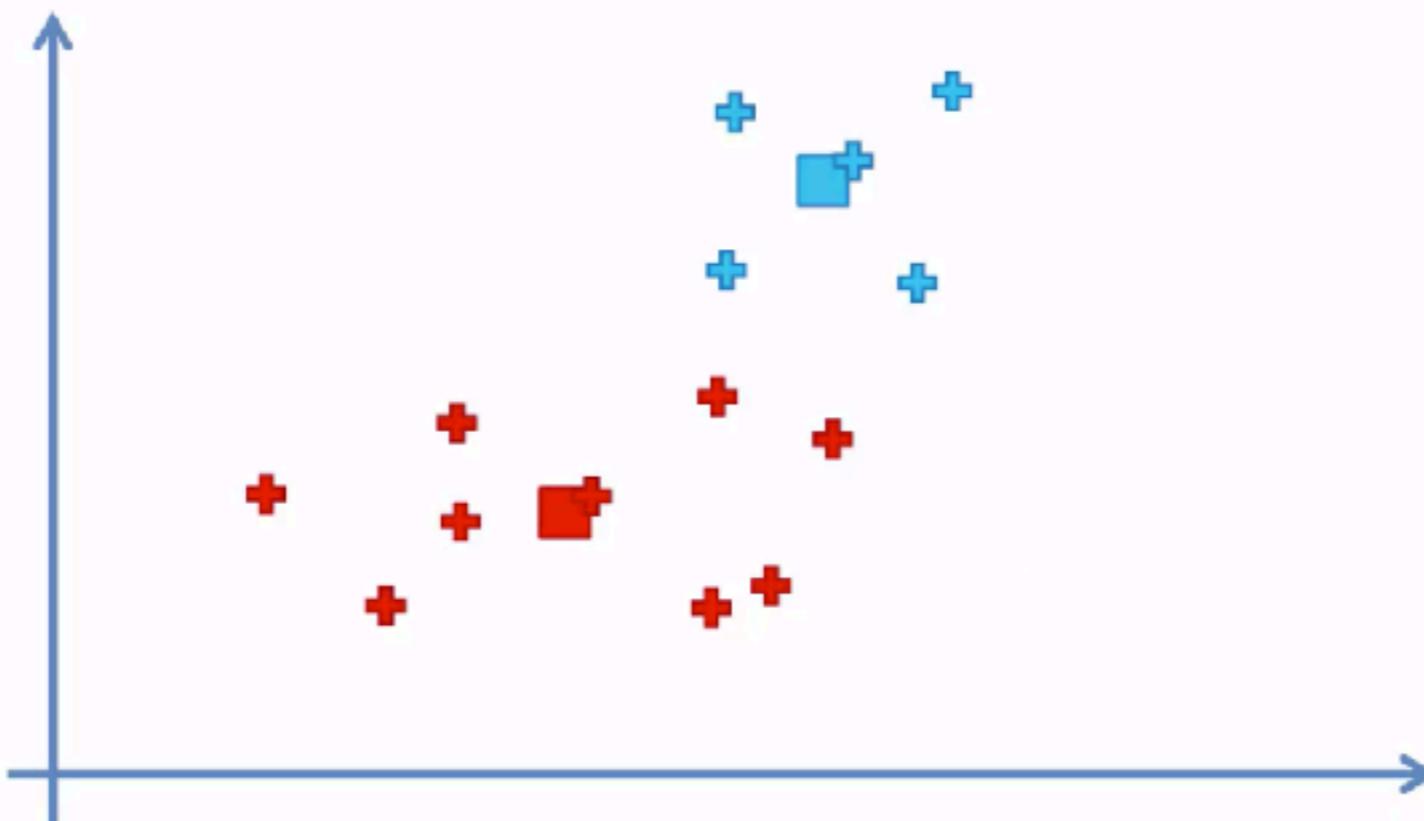
K-Means algorithm

STEP 4: Compute and place the new centroid of each cluster



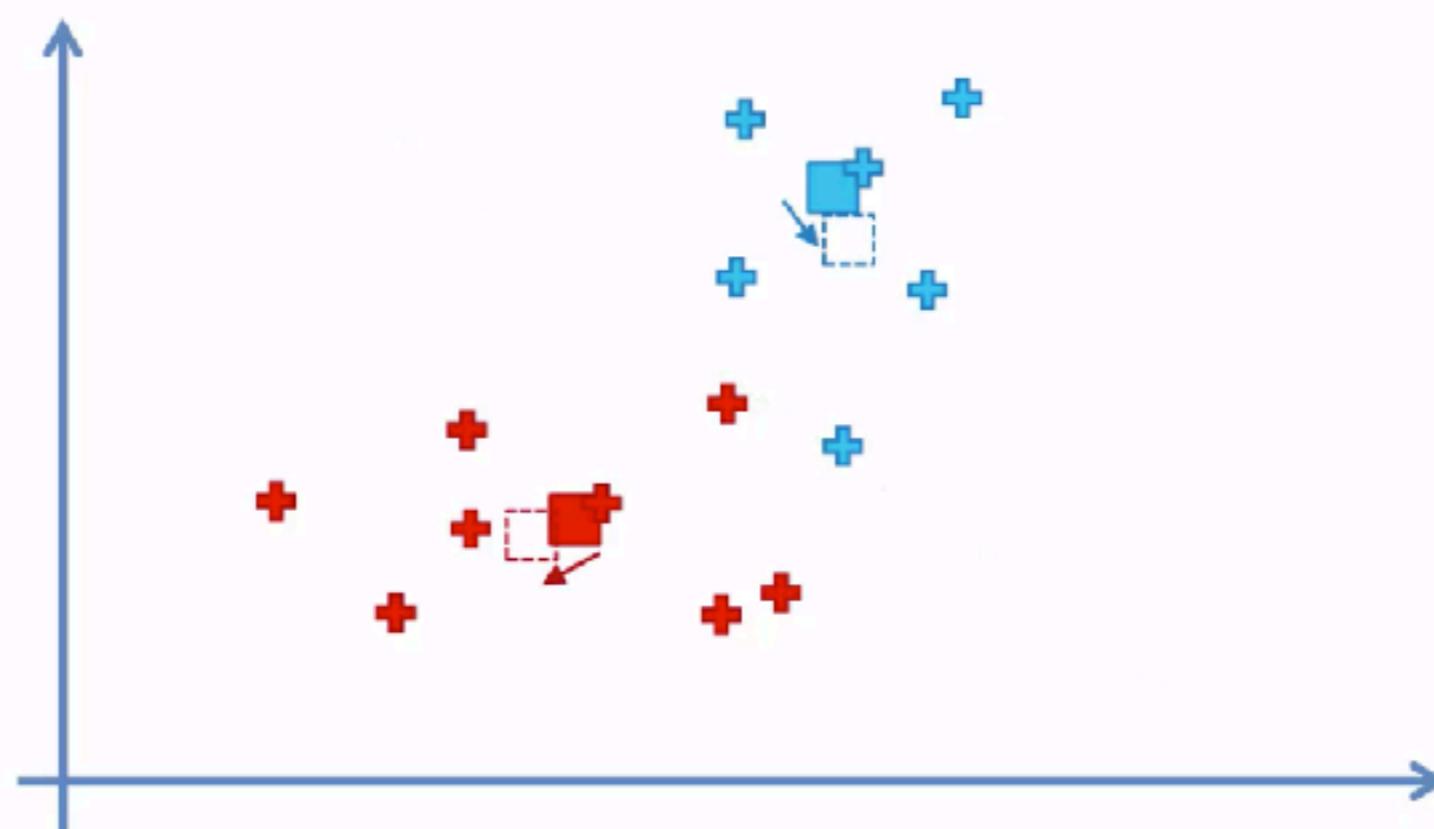
K-Means algorithm

STEP 5: Reassign each data point to the new closest centroid.
If any reassignment took place, go to STEP 4, otherwise go to FIN.



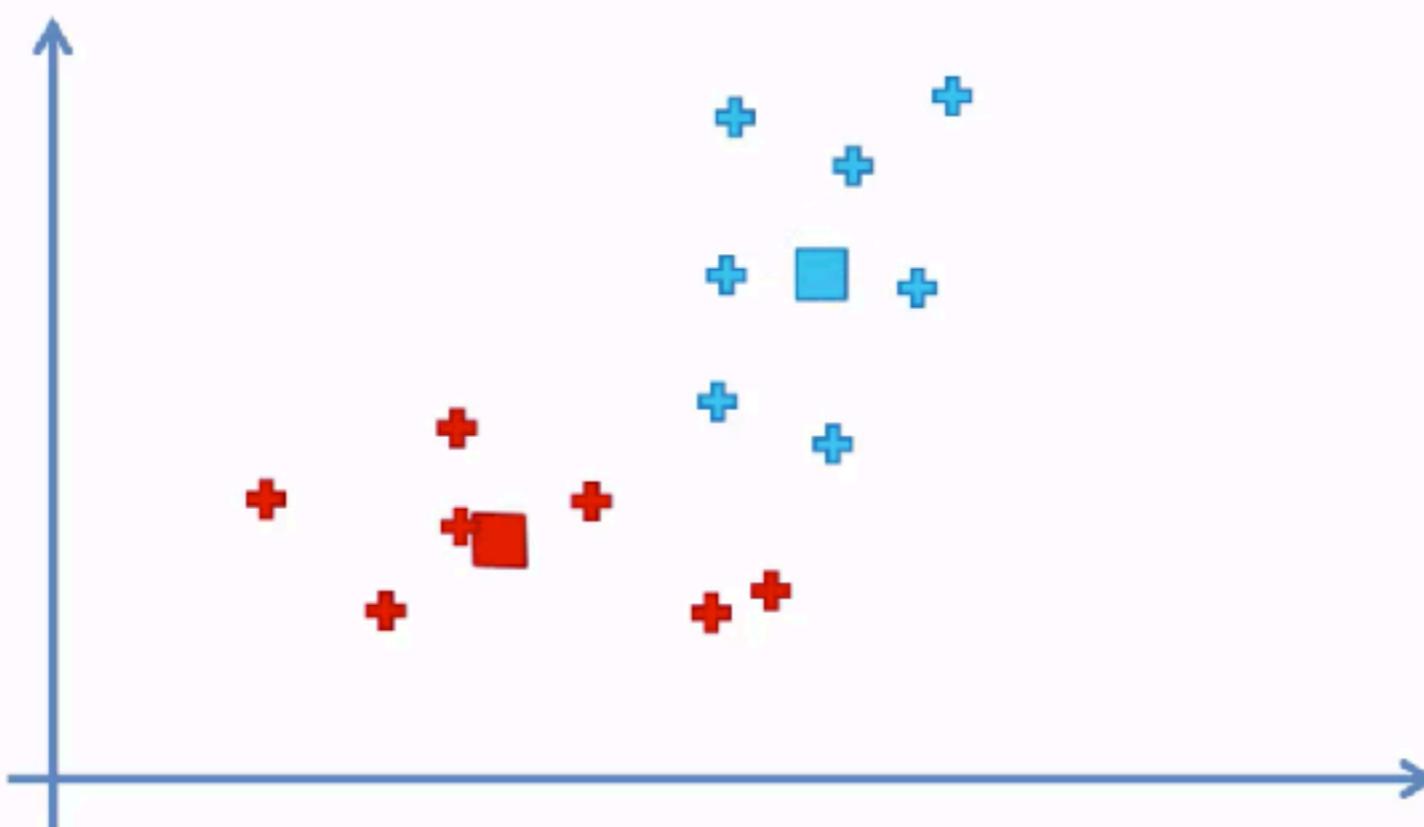
K-Means algorithm

STEP 4: Compute and place the new centroid of each cluster



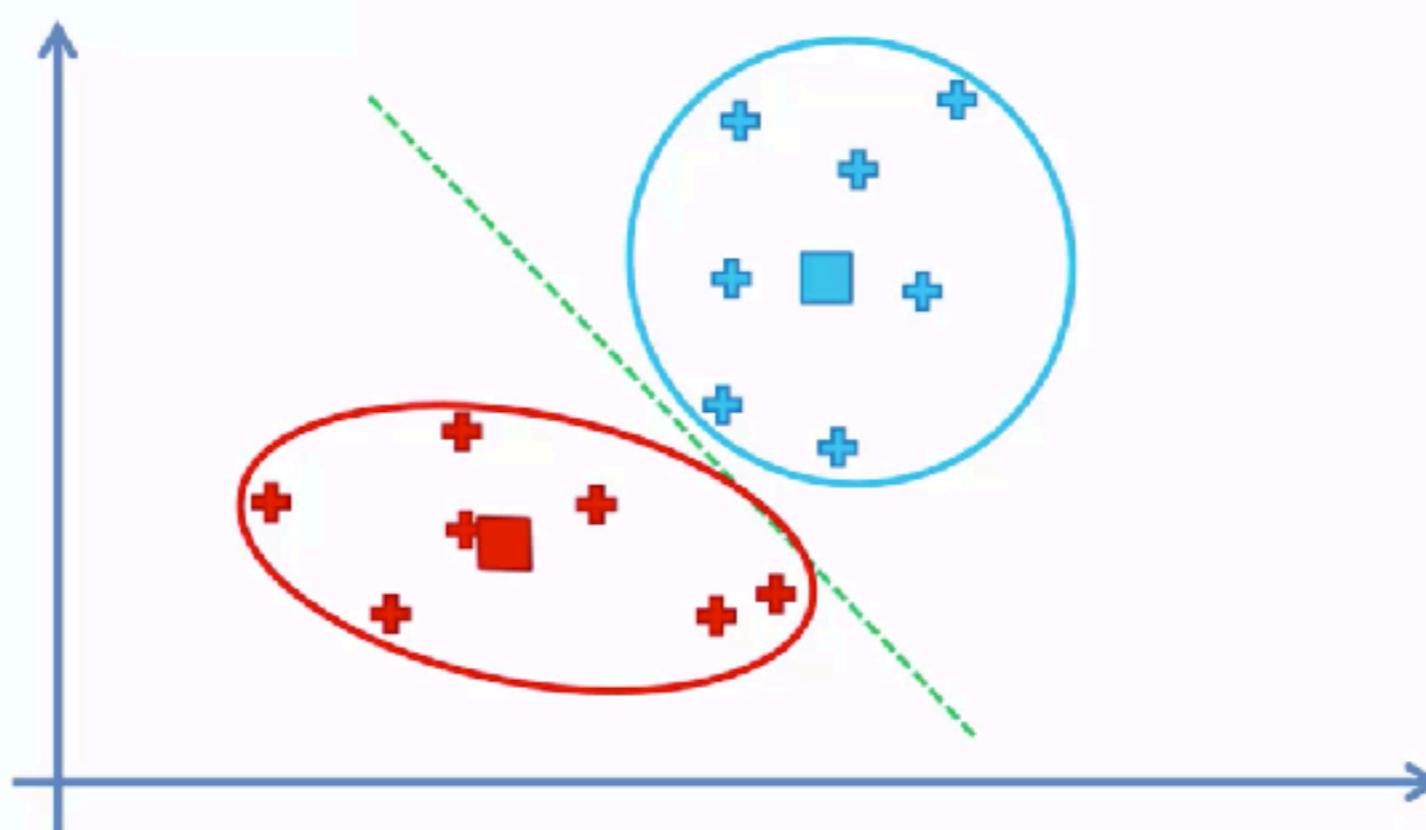
K-Means algorithm

STEP 4: Compute and place the new centroid of each cluster



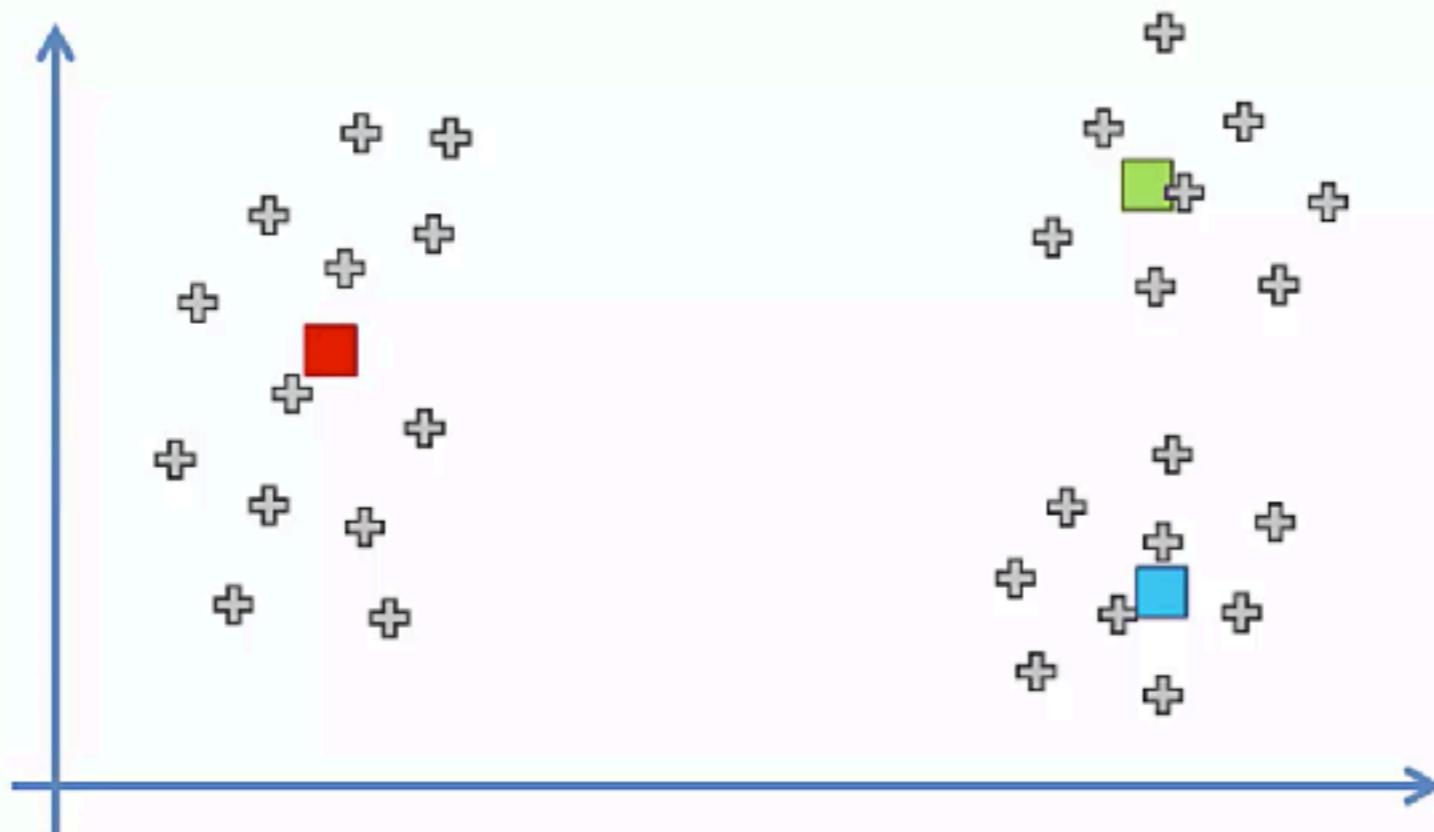
K-Means algorithm

FIN: Your Model Is Ready



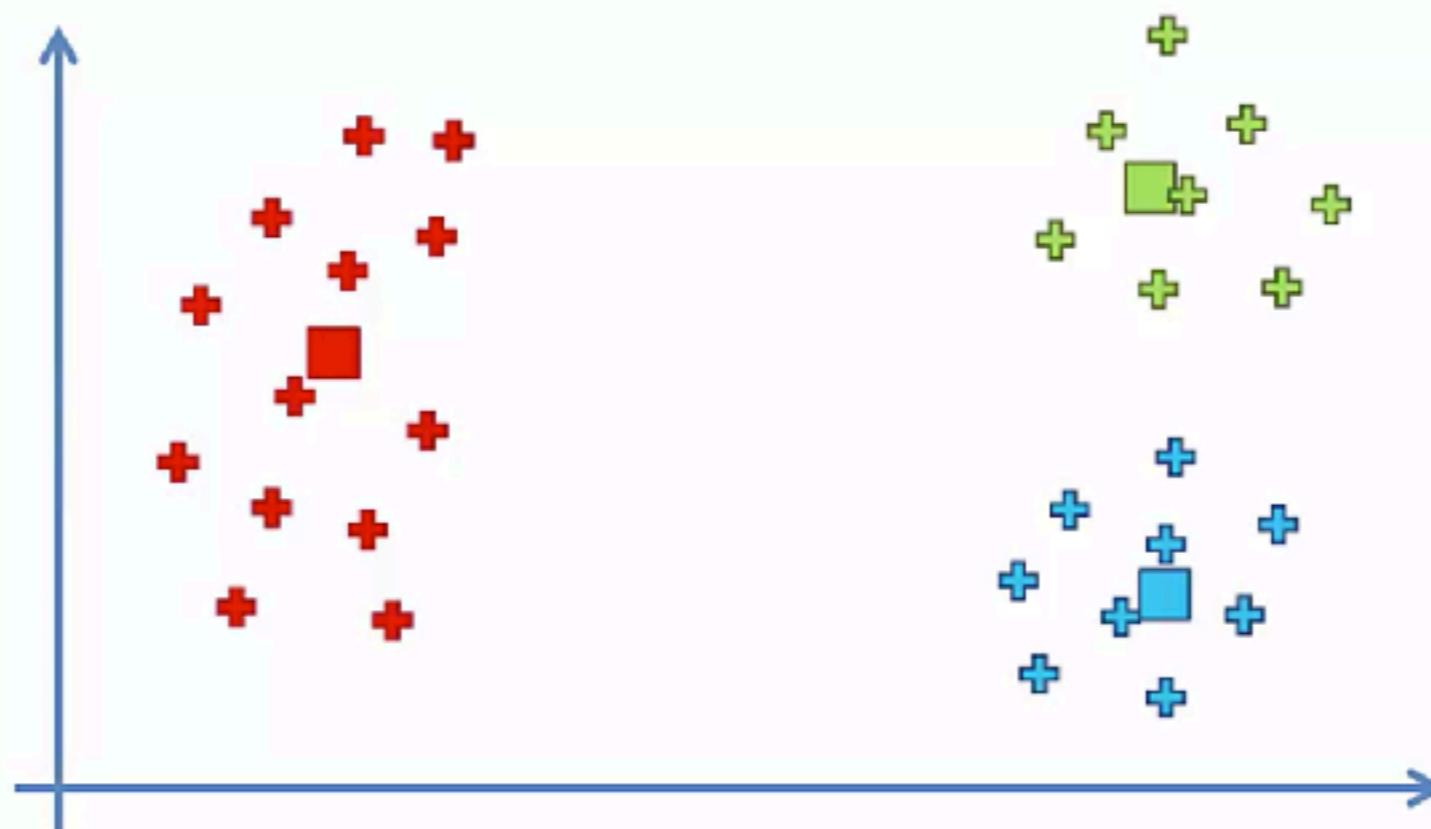
K-Means Intuition: Random Initialization Trap

Random Initialization Trap

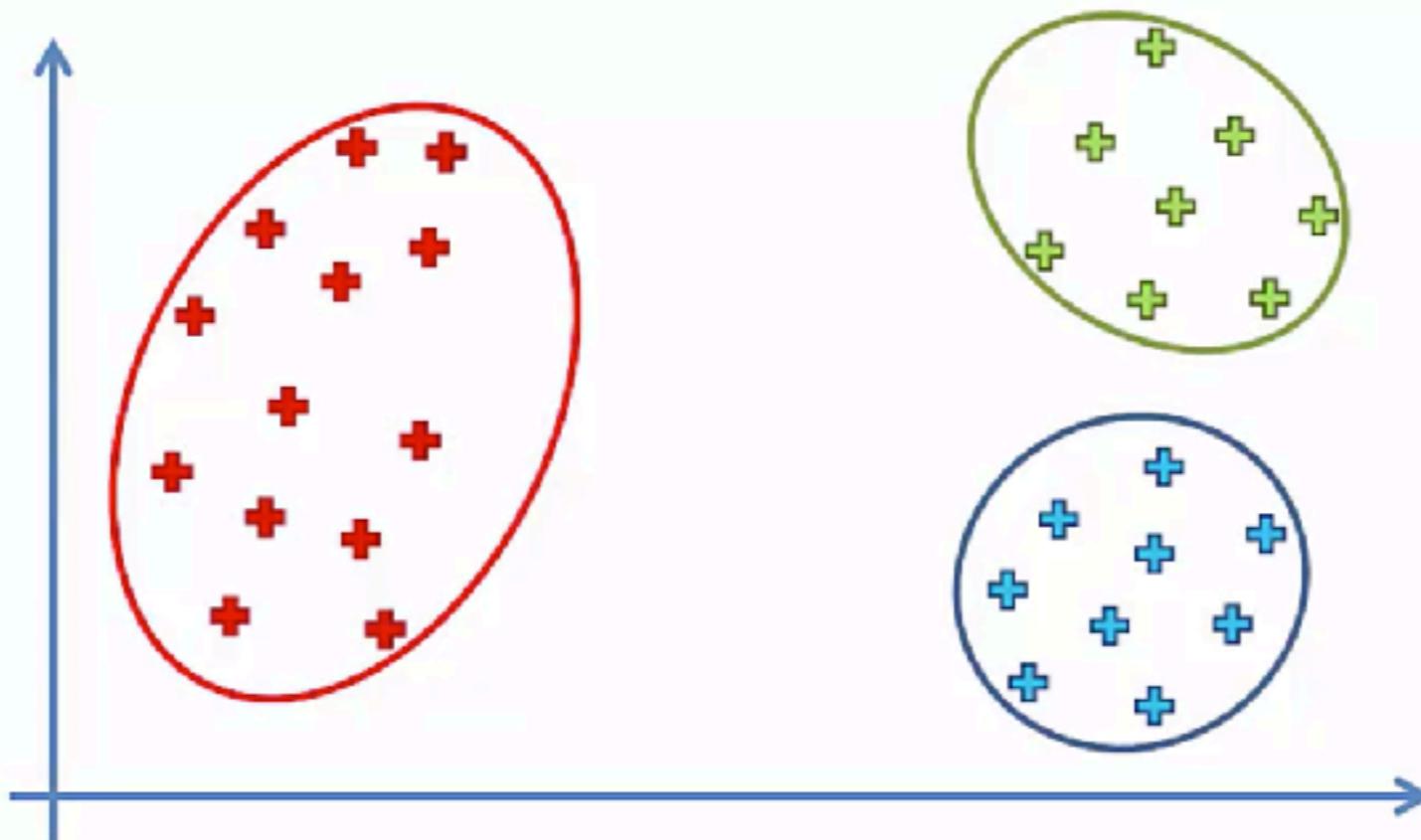


...this correct random initialisation would lead us
to...

Random Initialization Trap



Random Initialization Trap



...the following three clusters

Random Initialization Trap

But what would happen if we had a bad random initialisation ?

Random Initialization Trap

STEP 1: Choose the number K of clusters



STEP 2: Select at random K points, the centroids (not necessarily from your dataset)



STEP 3: Assign each data point to the closest centroid → That forms K clusters



STEP 4: Compute and place the new centroid of each cluster

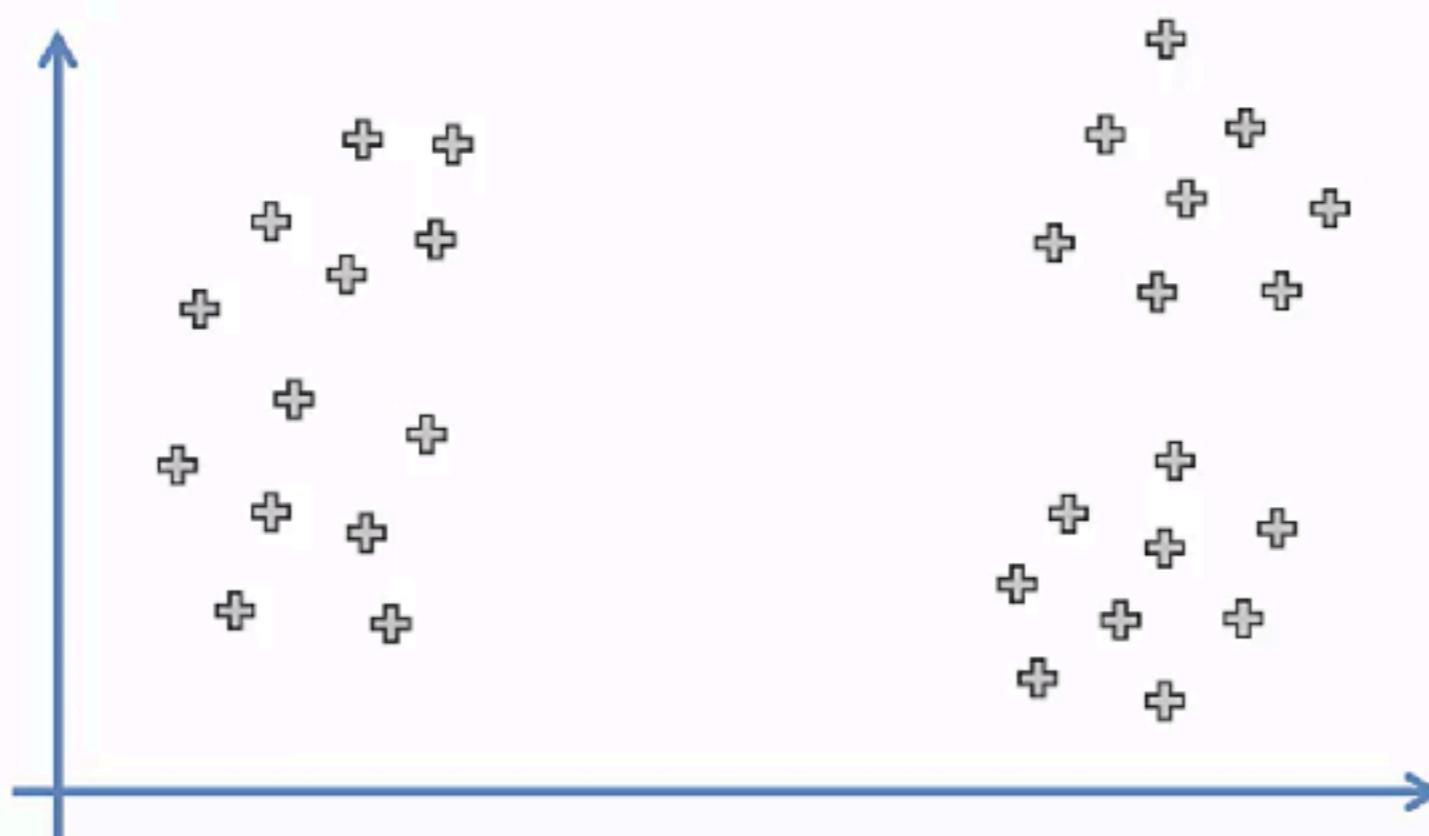


STEP 5: Reassign each data point to the new closest centroid.
If any reassignment took place, go to STEP 4, otherwise go to FIN.

Your Model is Ready

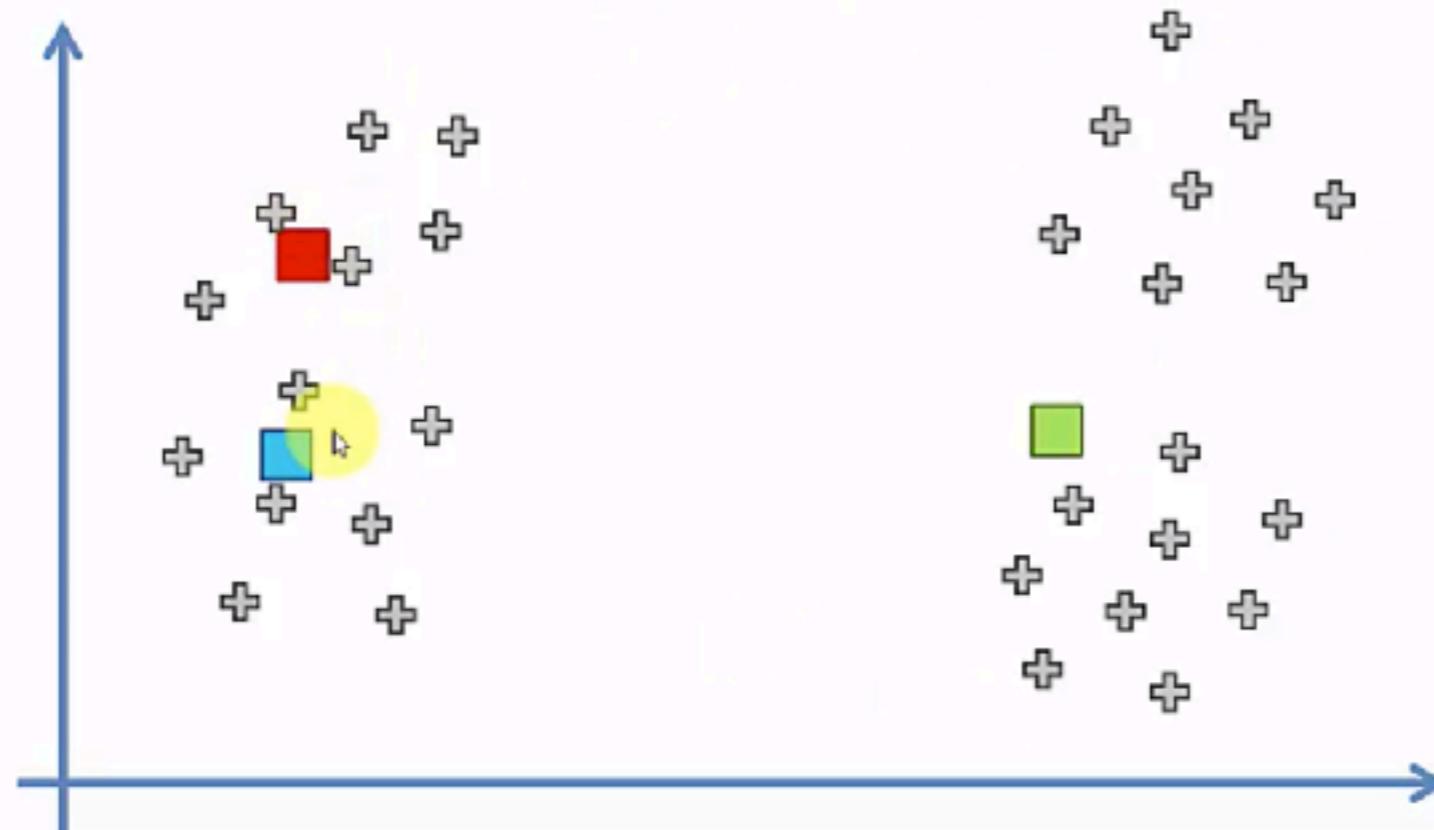
Random Initialization Trap

STEP 2: Select at random K points, the centroids (not necessarily from your dataset)



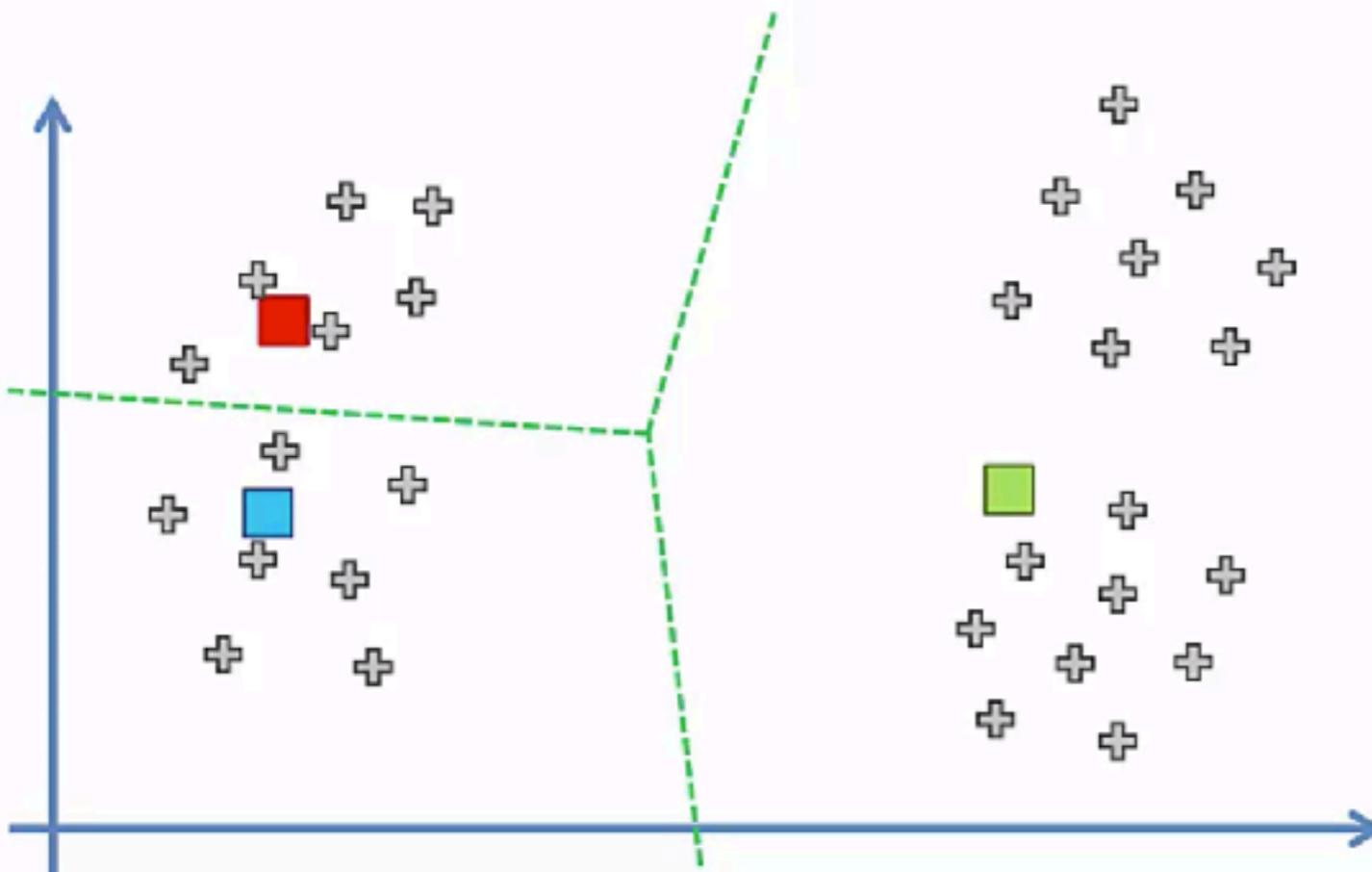
Random Initialization Trap

STEP 2: Select at random K points, the centroids (not necessarily from your dataset)



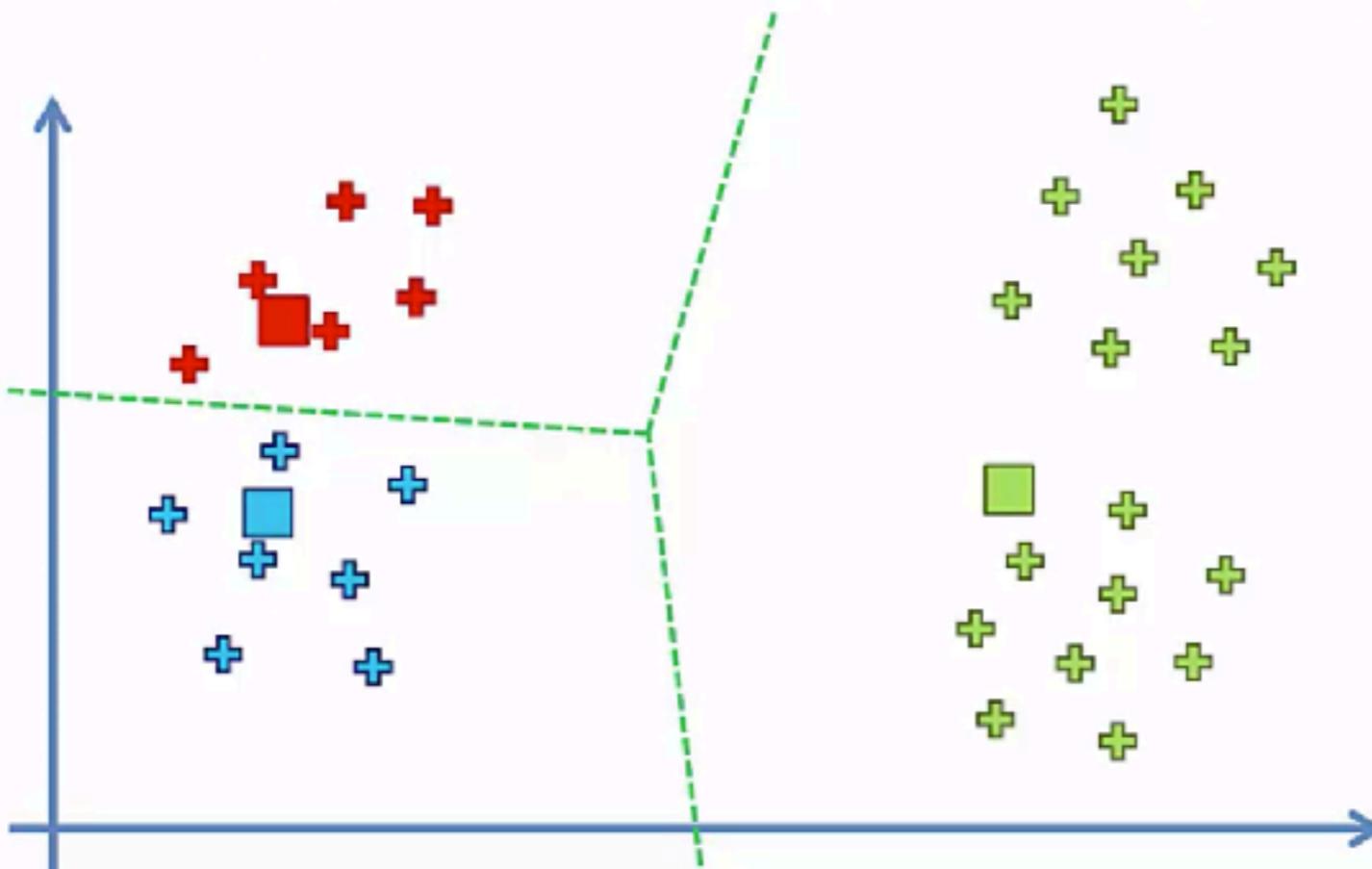
Random Initialization Trap

STEP 2: Select at random K points, the centroids (not necessarily from your dataset)



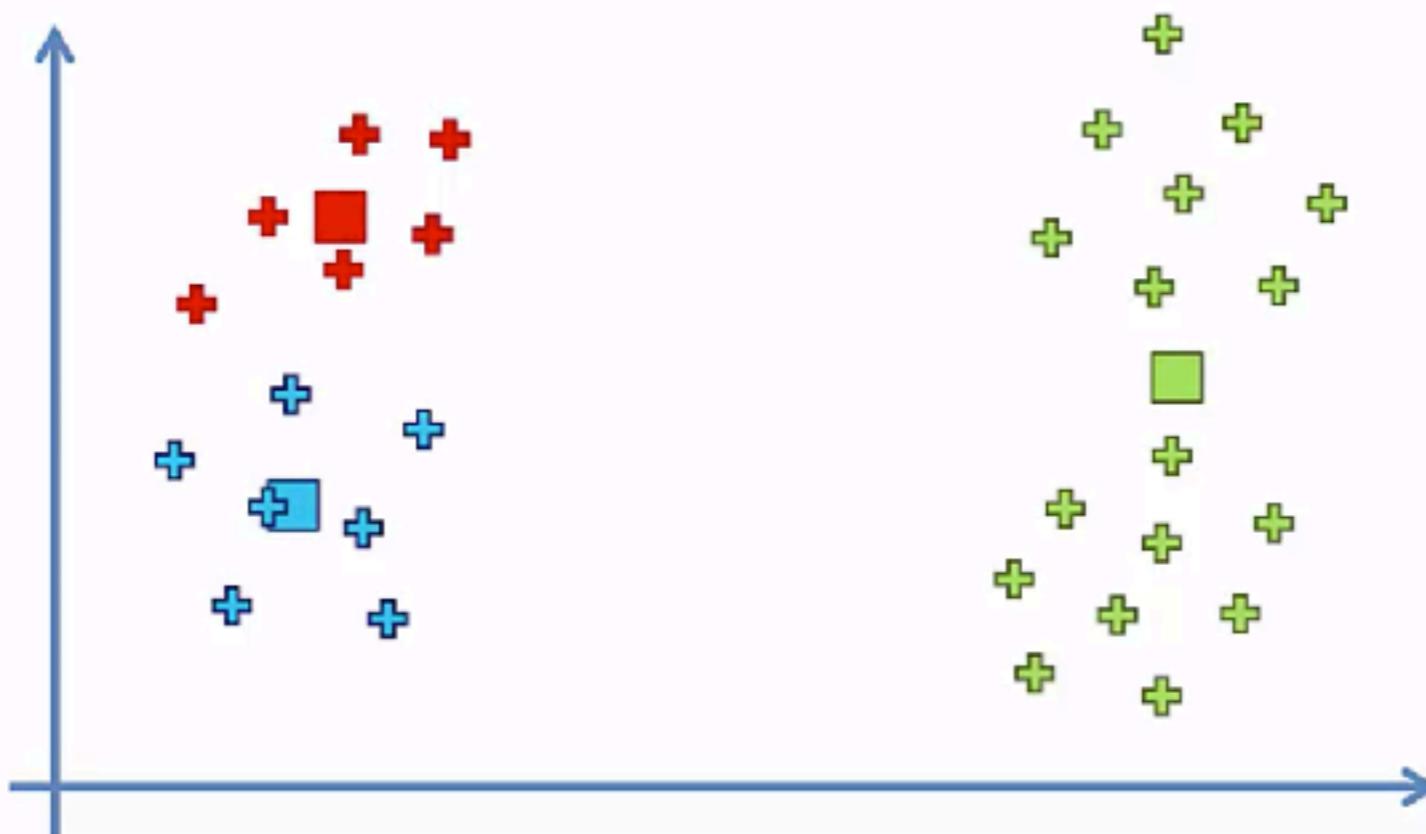
Random Initialization Trap

STEP 3: Assign each data point to the closest centroid \rightarrow That forms K clusters



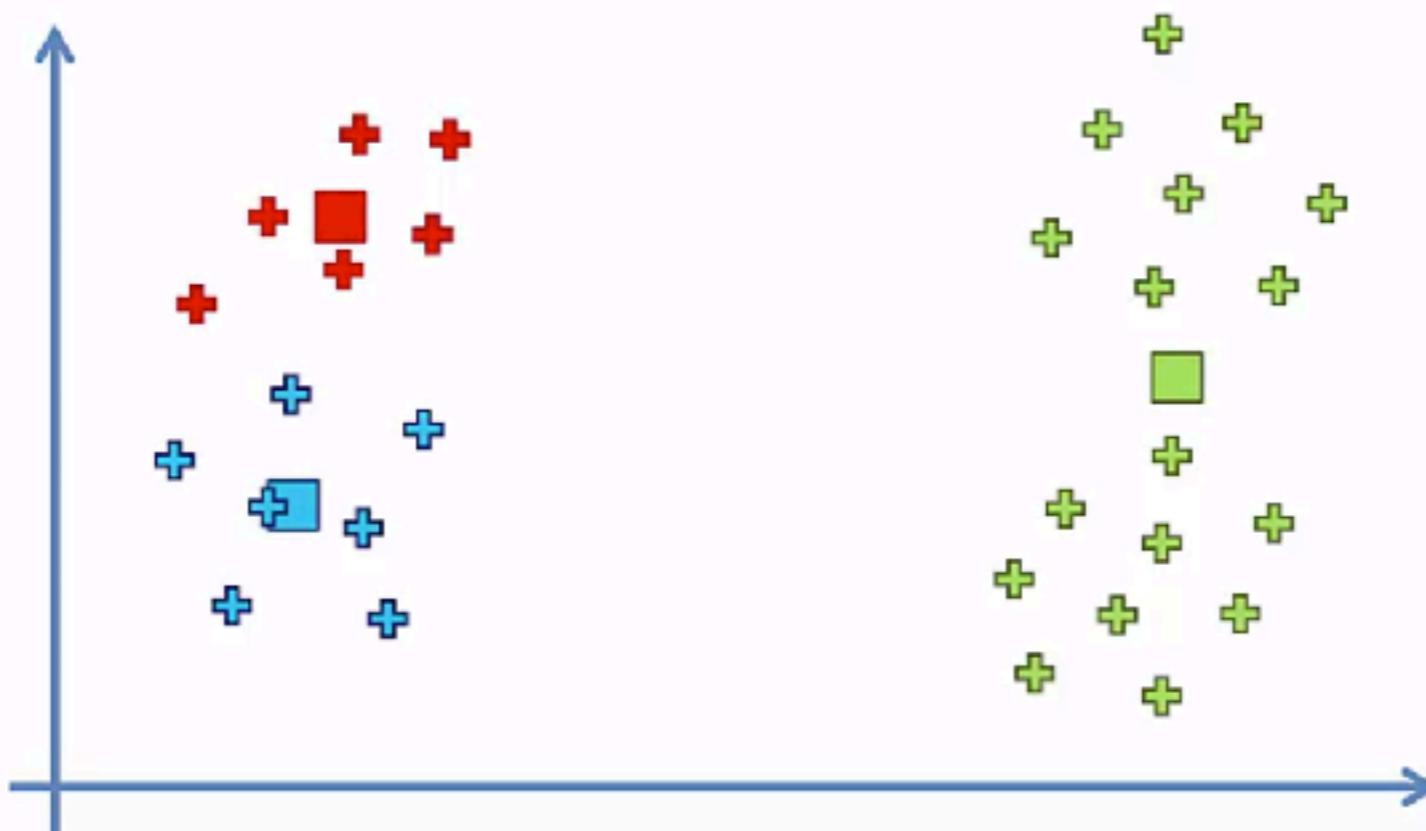
Random Initialization Trap

STEP 4: Compute and place the new centroid of each cluster



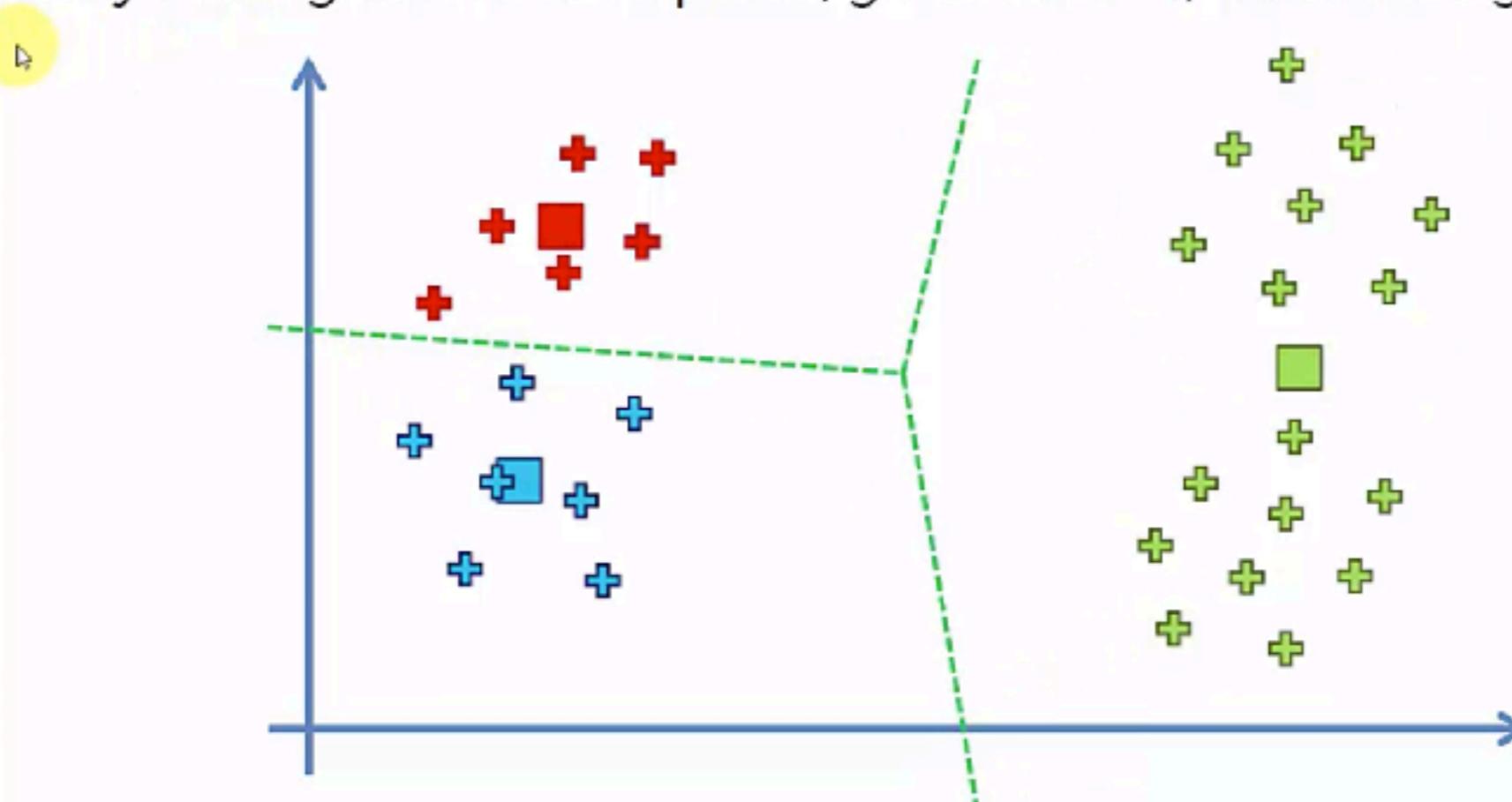
Random Initialization Trap

STEP 4: Compute and place the new centroid of each cluster



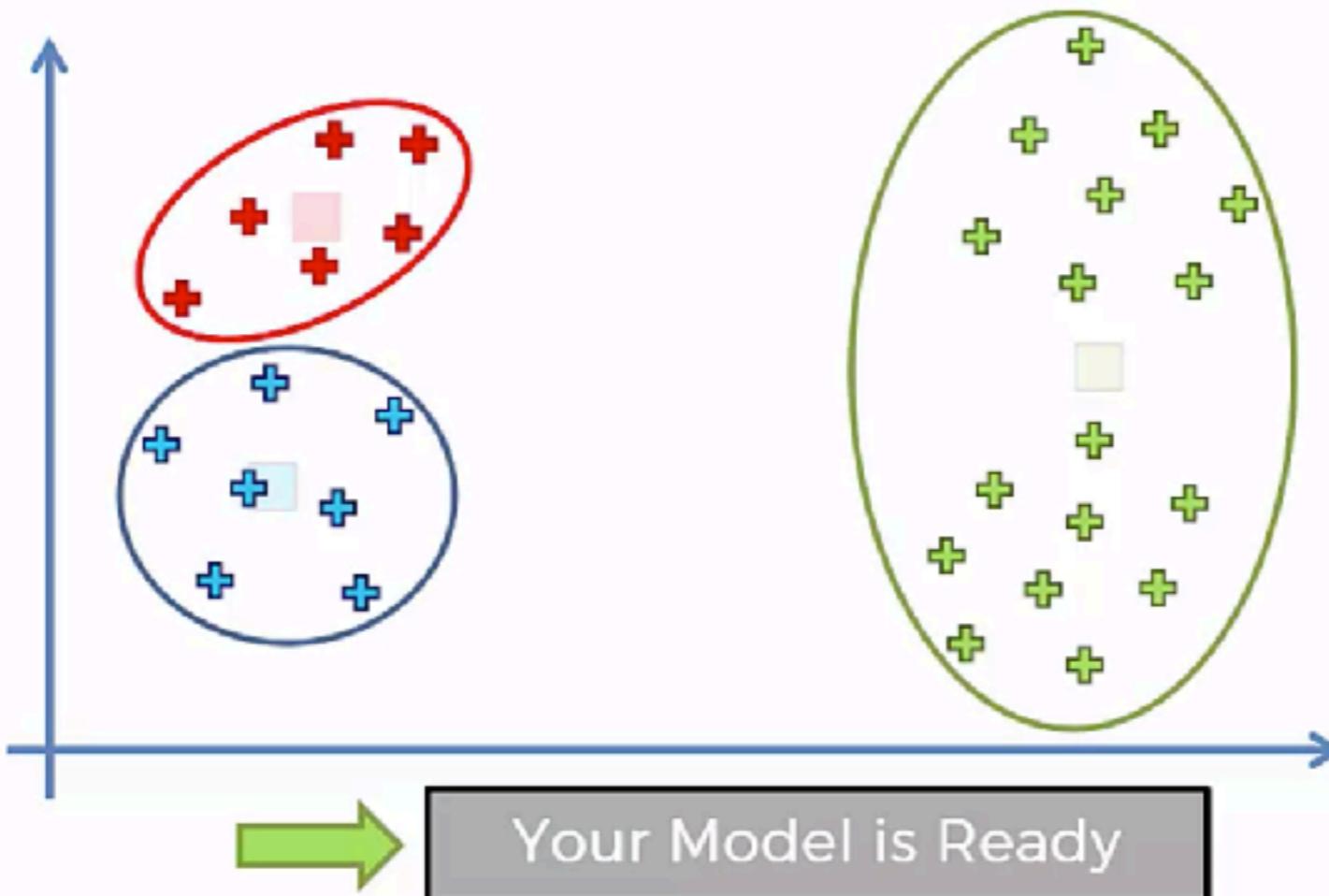
Random Initialization Trap

STEP 5: Reassign each data point to the new closest centroid.
If any reassignment took place, go to STEP 4, otherwise go to FIN.

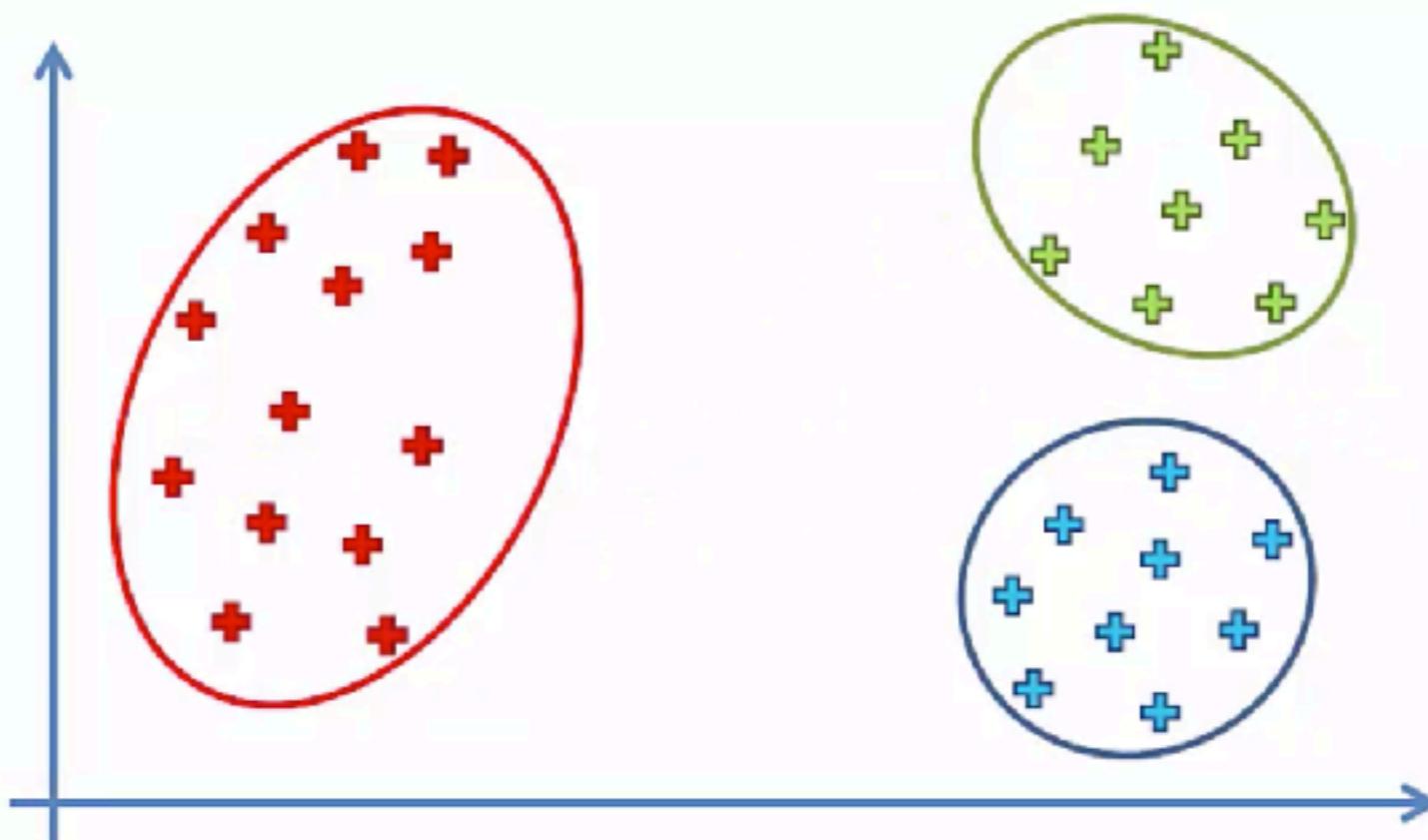


Random Initialization Trap

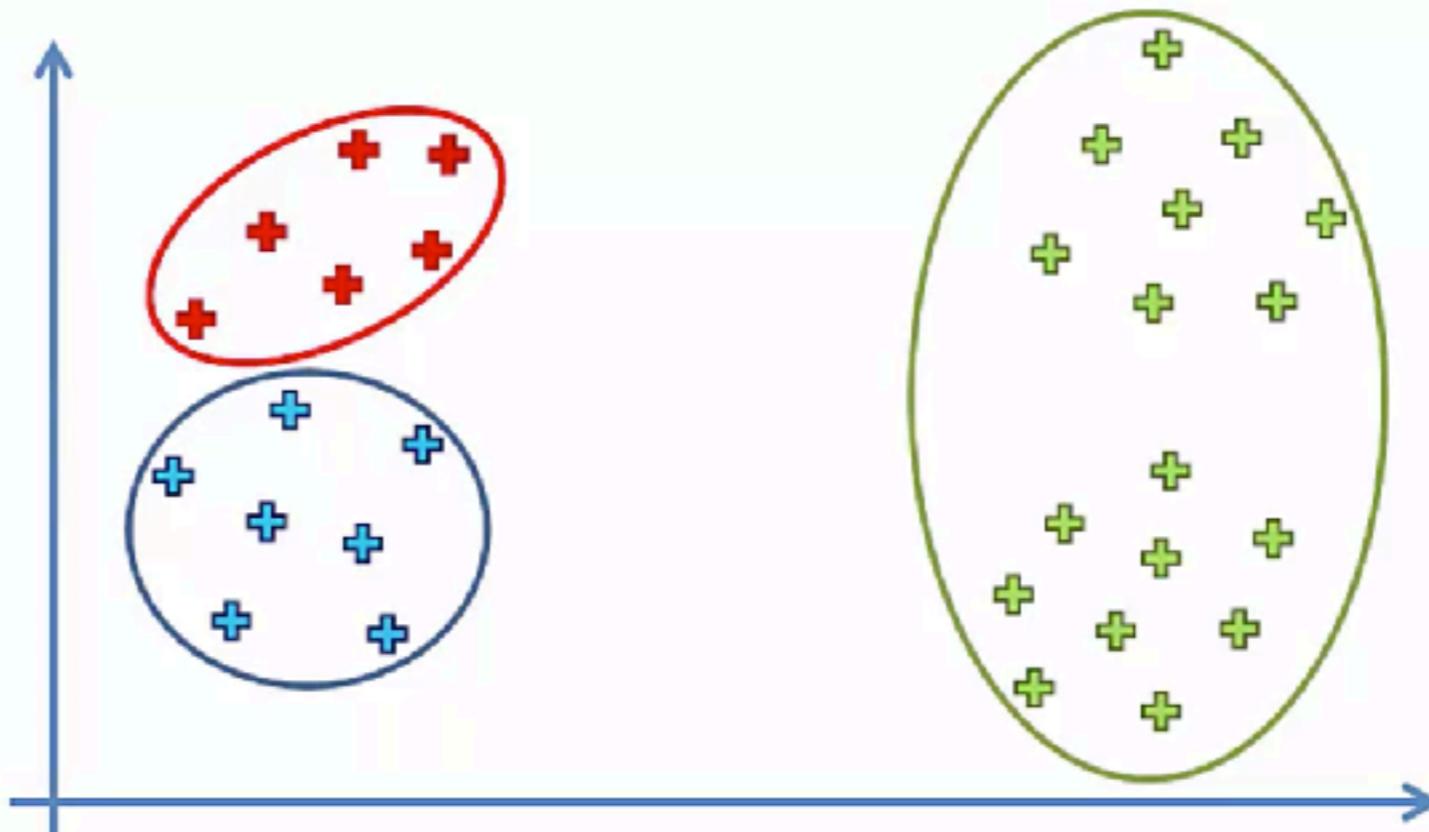
STEP 5: Reassign each data point to the new closest centroid.
If any reassignment took place, go to STEP 4, otherwise go to FIN.



Random Initialization Trap



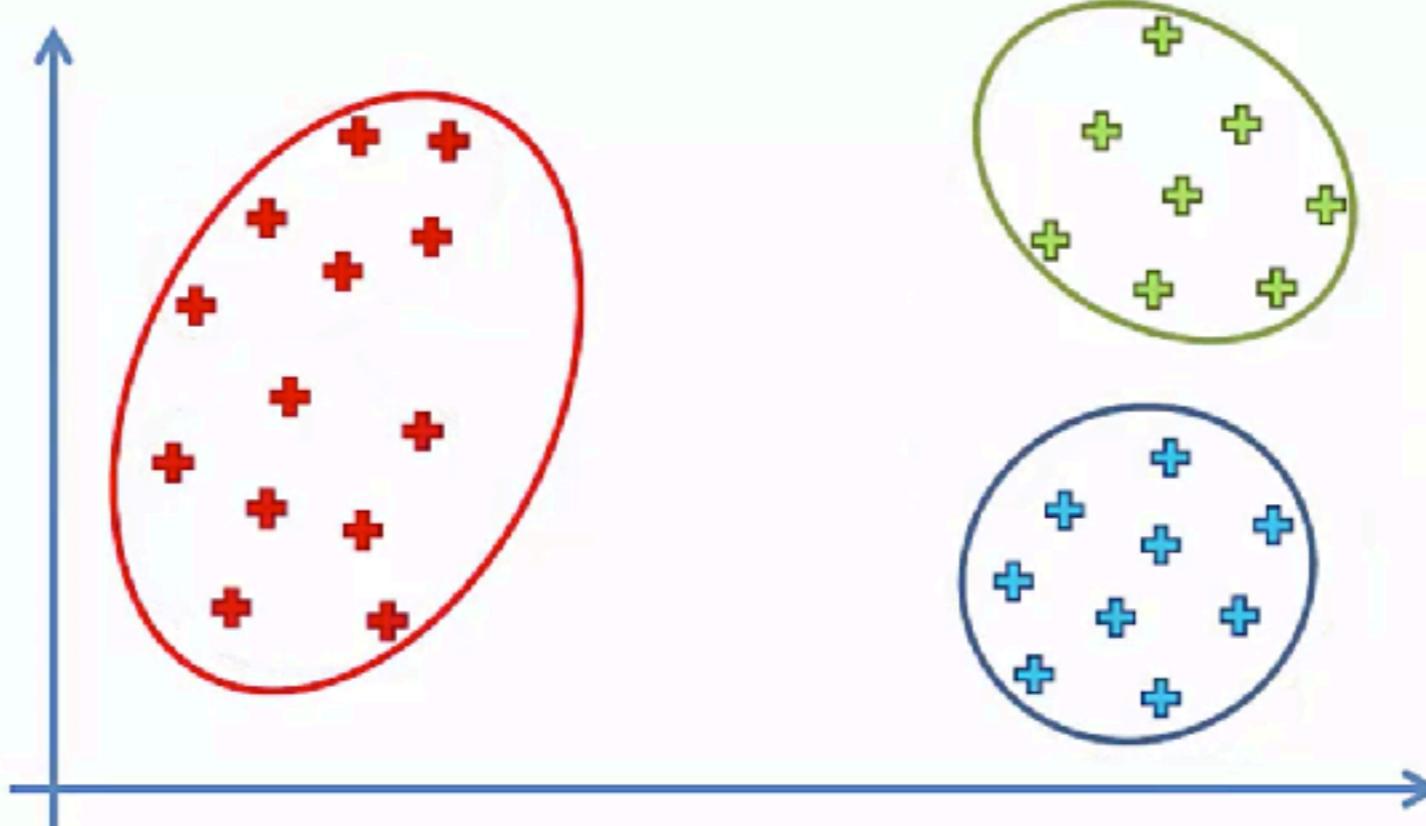
Random Initialization Trap



Random Initialization Trap

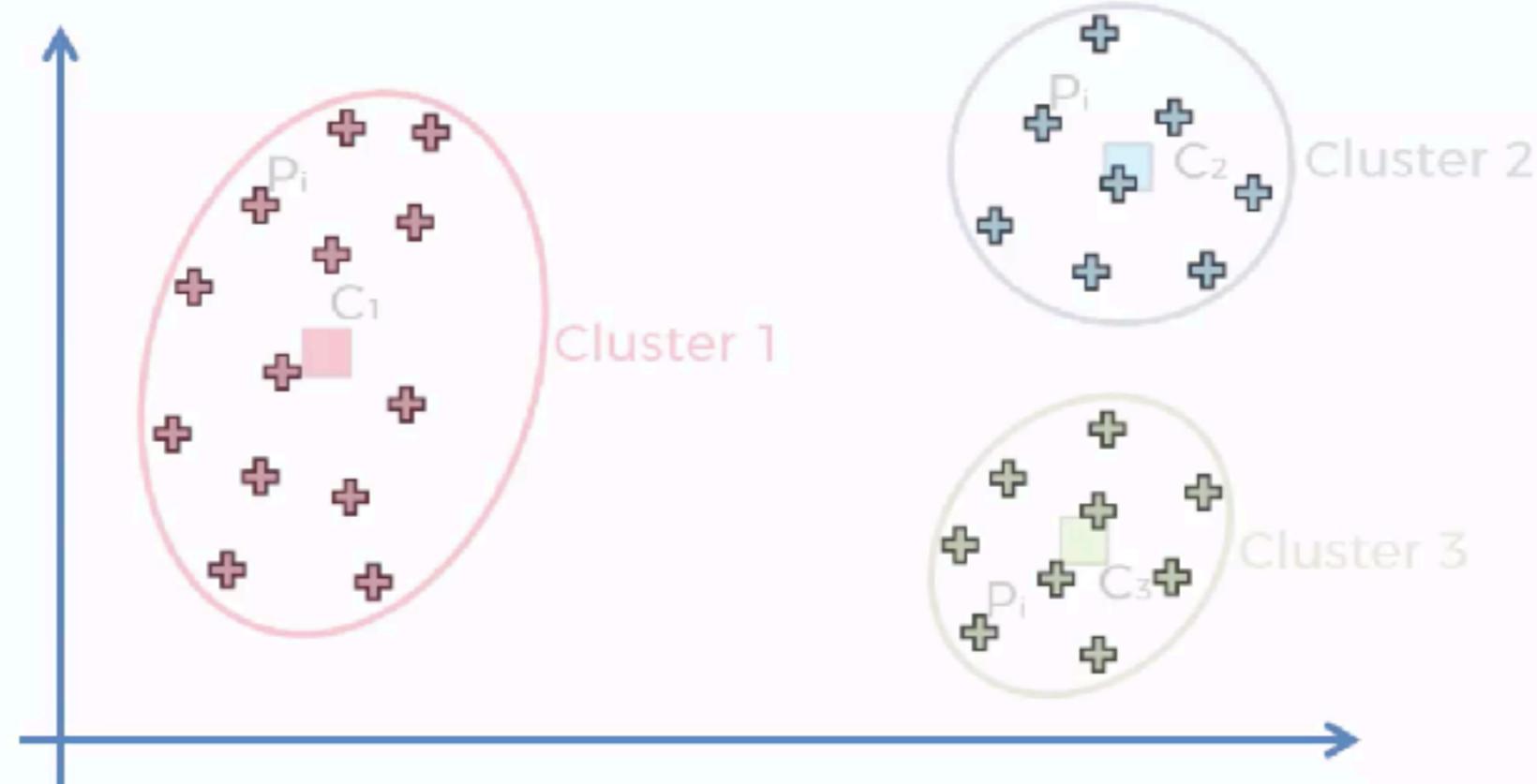


Random Initialization Trap

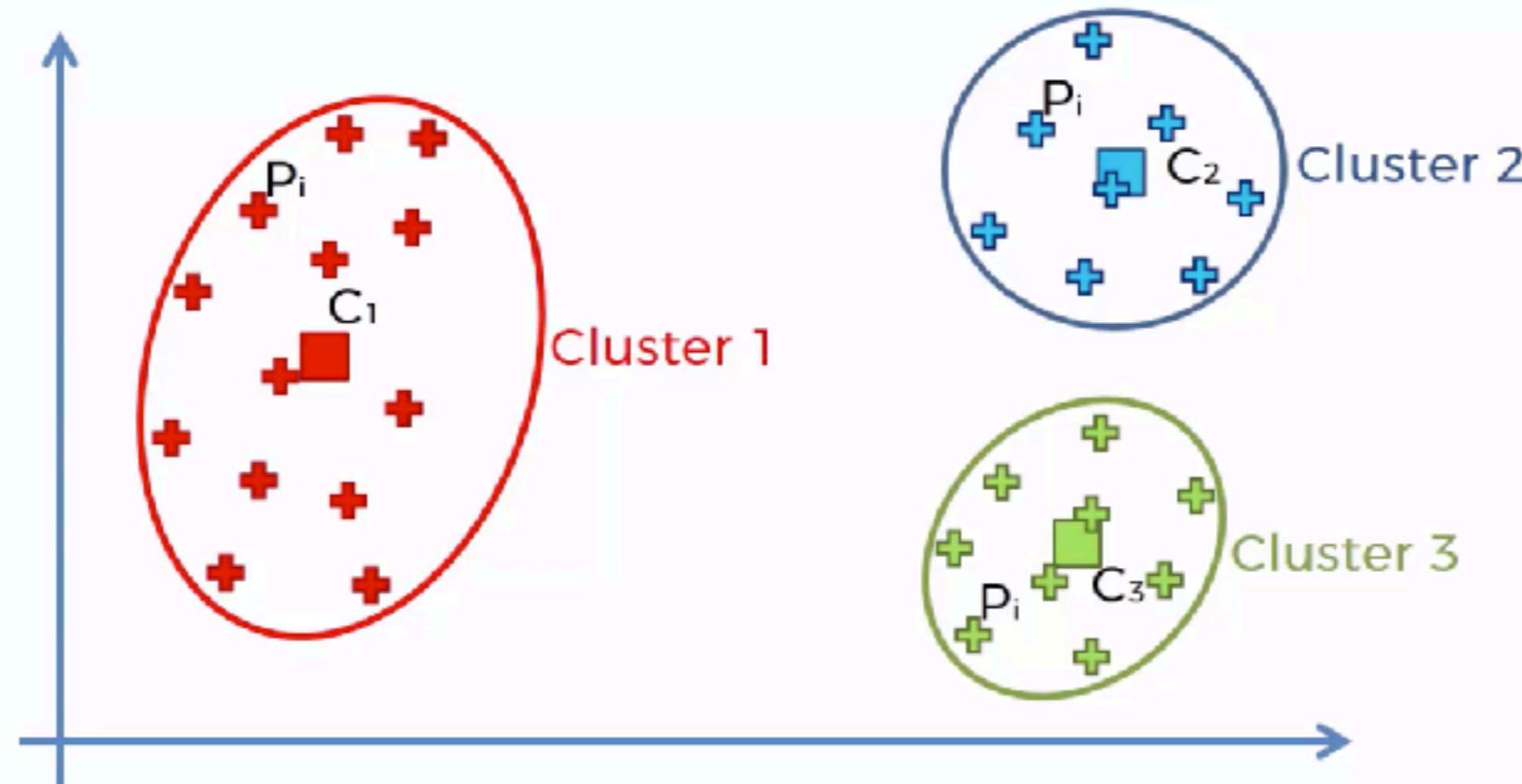


K-Means Intuition: Choosing the right number of clusters

Choosing the right number of clusters



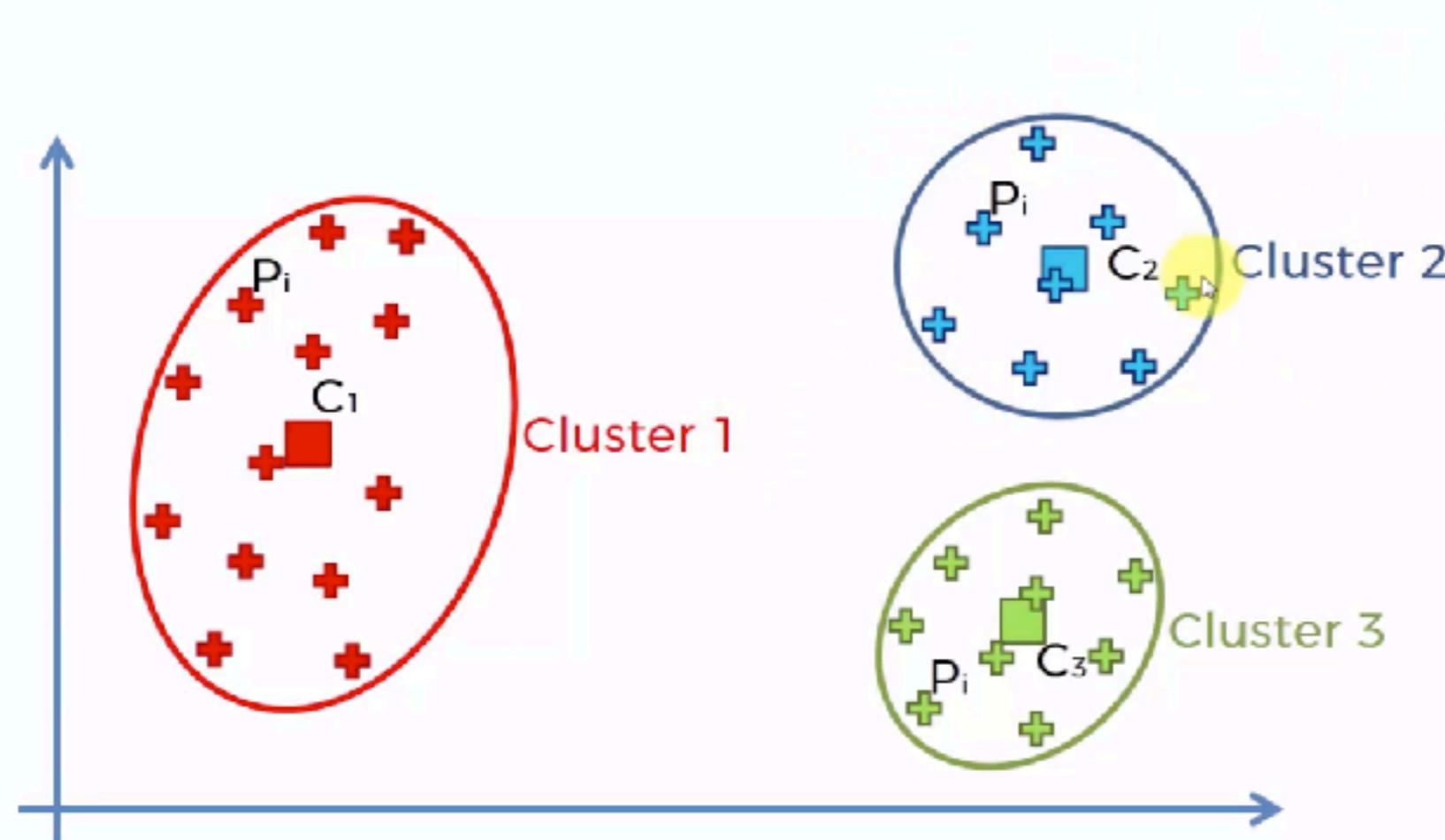
Choosing the right number of clusters



Choosing the right number of clusters

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

Choosing the right number of clusters

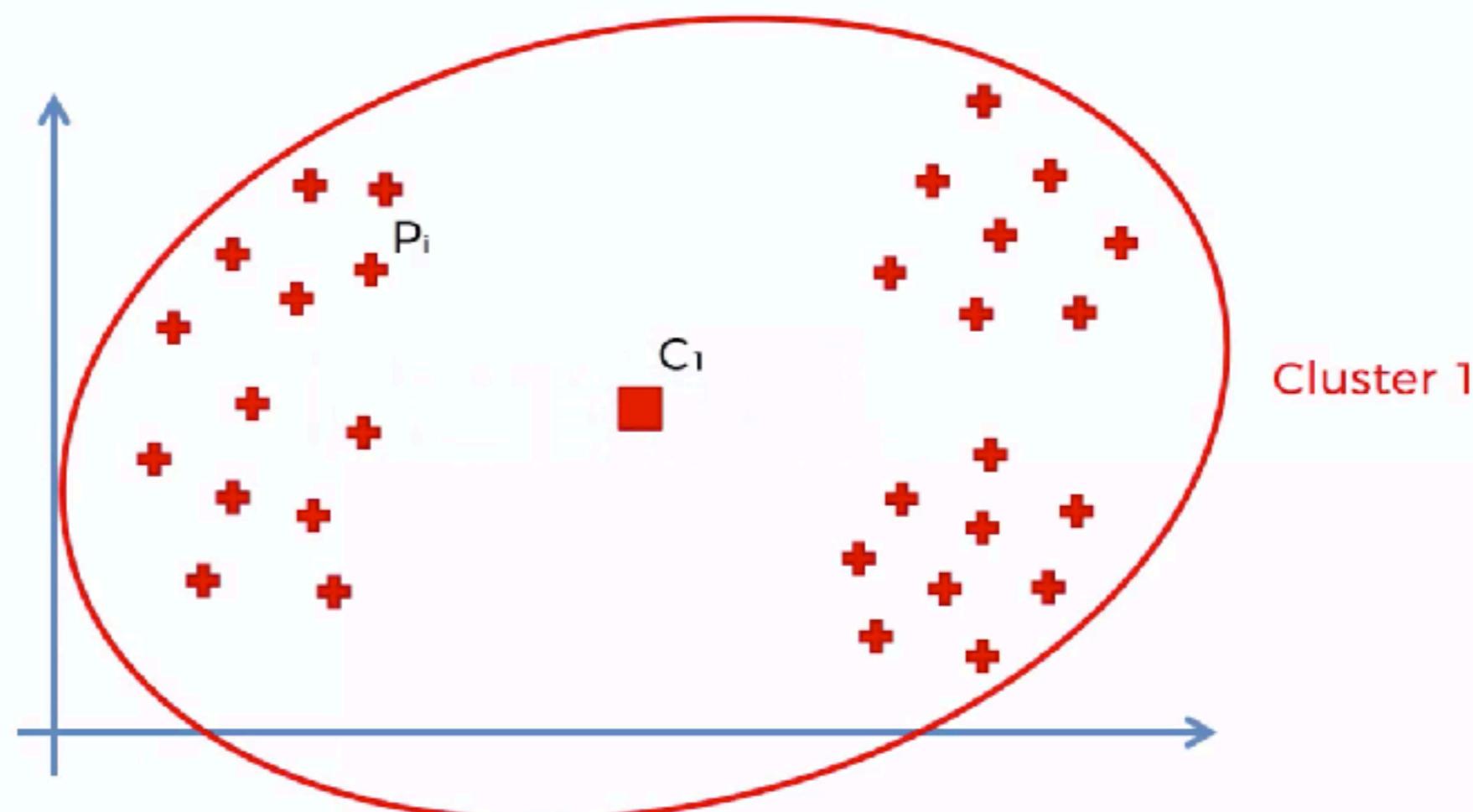


$$\text{WCSS} = \sum_{P_i \text{ in Cluster 1}} \text{distance}(P_i, C_1)^2 + \sum_{P_i \text{ in Cluster 2}} \text{distance}(P_i, C_2)^2 + \sum_{P_i \text{ in Cluster 3}} \text{distance}(P_i, C_3)^2$$

Choosing the right number of clusters

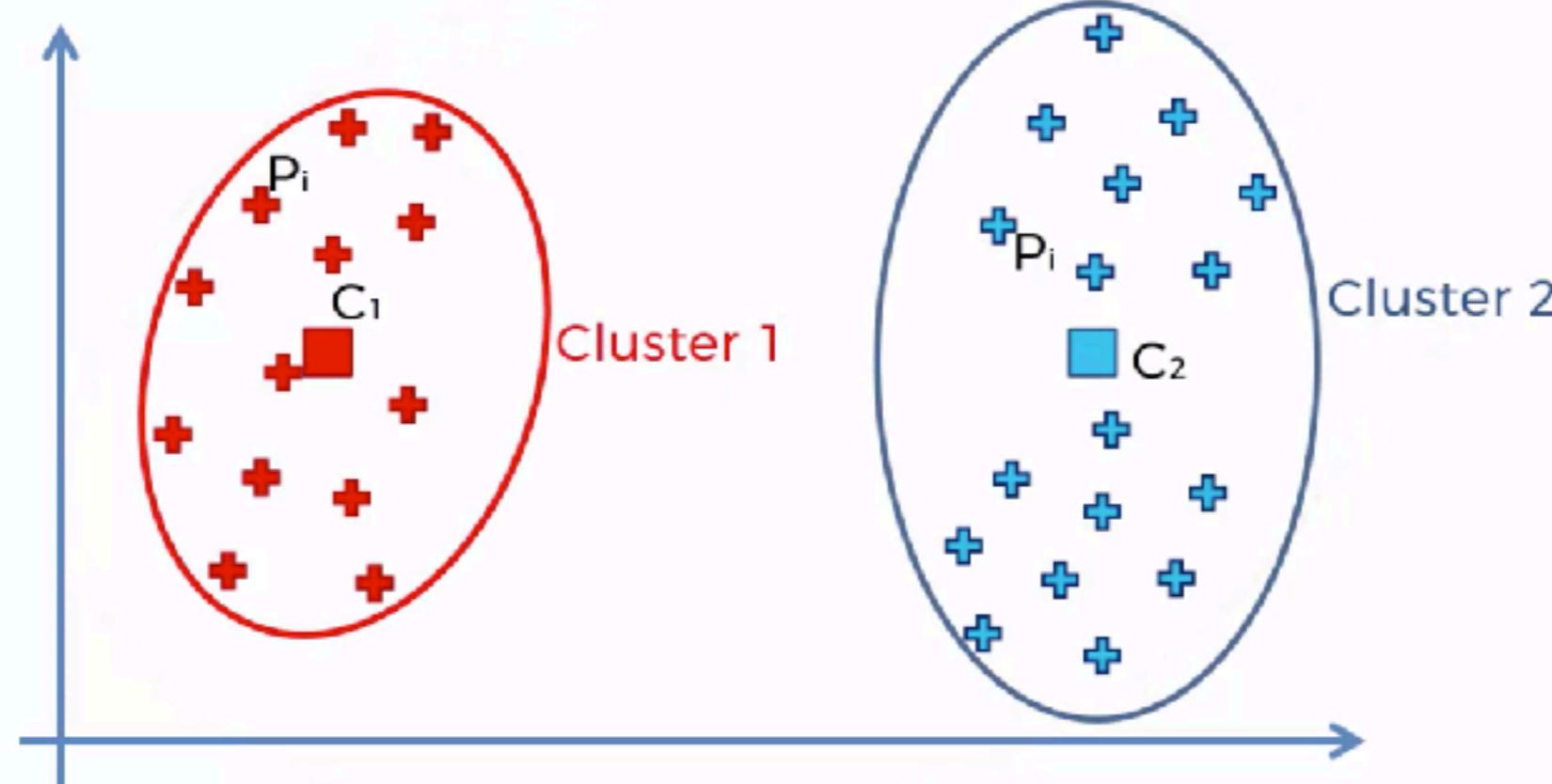
Rewind...

Choosing the right number of clusters



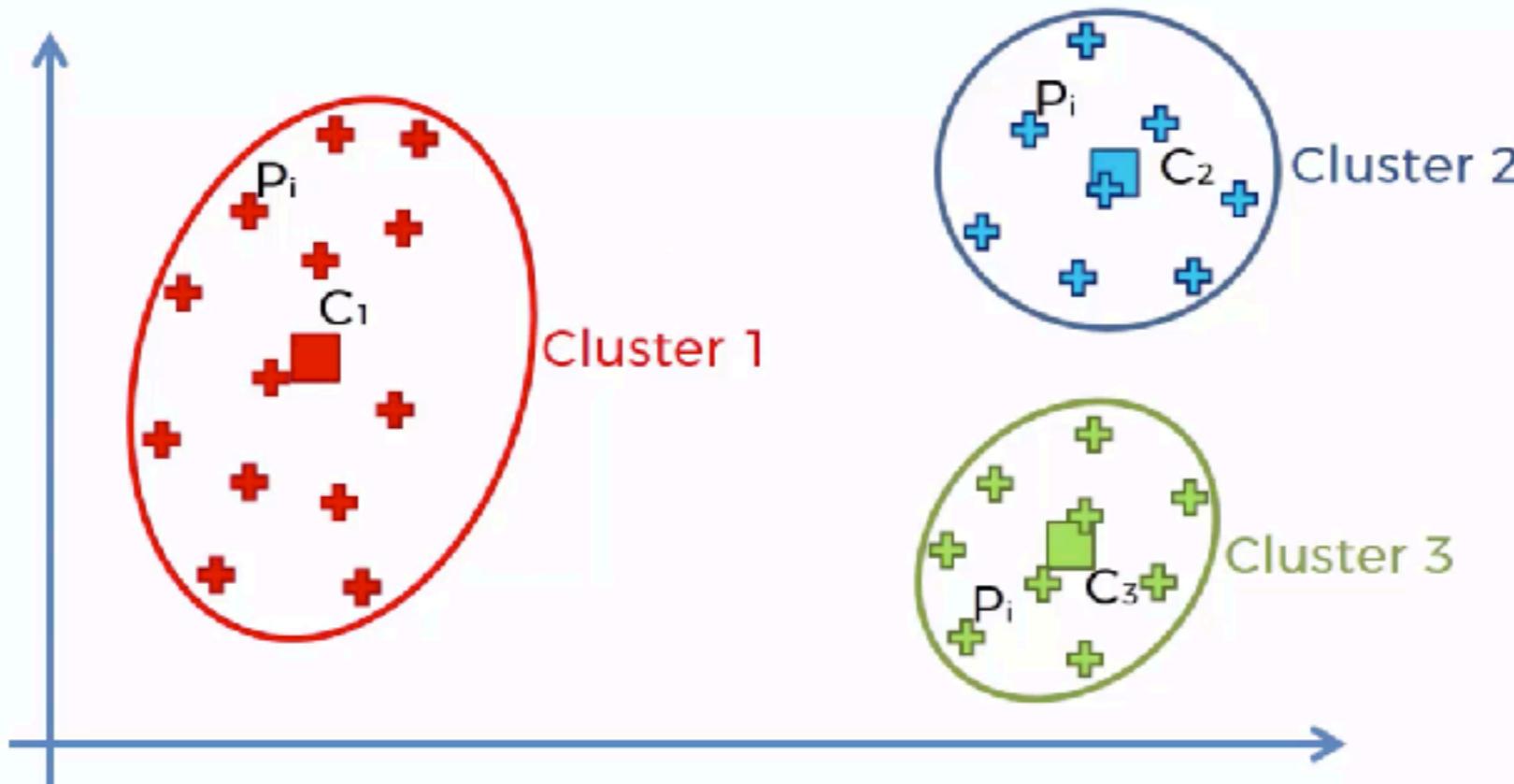
$$\text{WCSS} = \sum_{P_i \text{ in Cluster 1}} \text{distance}(P_i, C_1)^2$$

Choosing the right number of clusters



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

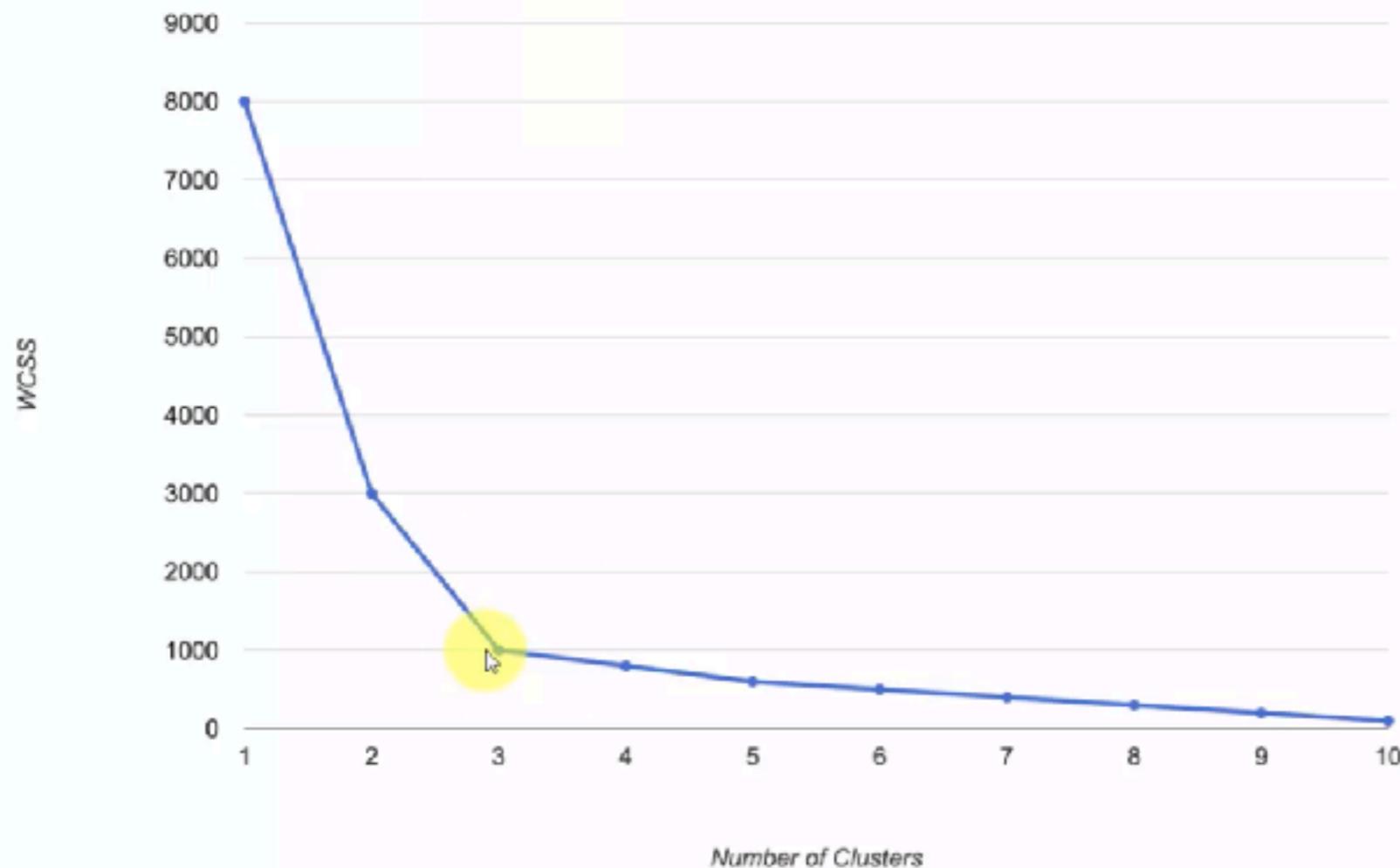
Choosing the right number of clusters



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

Choosing the right number of clusters

The Elbow Method



Choosing the right number of clusters

The Elbow Method

