How Kmeans Clustering works ?

How it works ?

The working mechanism of K-means can be described as follows:

- **Initialization**: The algorithm begins by randomly selecting K initial centroids from the dataset or by using more sophisticated initialization methods like K-means++, which strategically places initial centroids far apart from each other.
- **Assignment Step**: Each data point is assigned to the nearest centroid based on a distance metric (typically Euclidean distance). This creates K initial clusters, where each cluster contains all the points closest to its centroid.
- **Update Step**: After all points have been assigned, the algorithm recalculates the position of each centroid by computing the mean of all data points assigned to that cluster. This is where the "means" in K-means comes from.
- **Iteration**: The assignment and update steps are repeated iteratively. In each iteration, data points may switch clusters as centroids move, and new centroids are computed based on the updated cluster memberships.
- **Convergence**: The algorithm continues until convergence is achieved – either when centroids no longer move significantly between iterations, when cluster assignments remain stable, or when a maximum number of iterations is reached.

Step 1 - Initialization

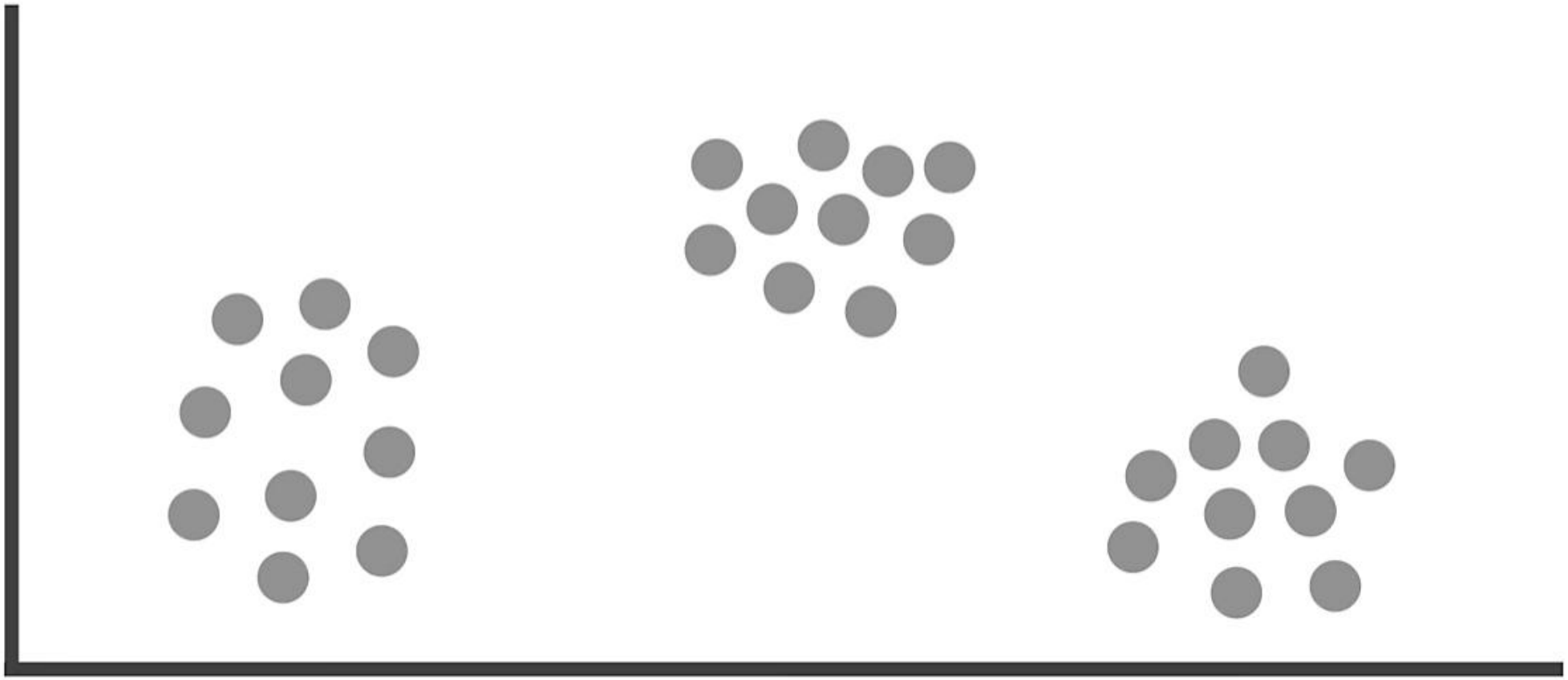
Step 1 - Initialization

- **Specify the number of clusters (K):** You must first define the number of clusters you want the algorithm to find in the data. This is typically determined by the user or an external method.

- **Randomly choose initial cluster centers:** The algorithm then selects K data points from the dataset to serve as the initial, random centroids. All subsequent steps will work to refine the positions of these centers to find the optimal clusters.

K-Means

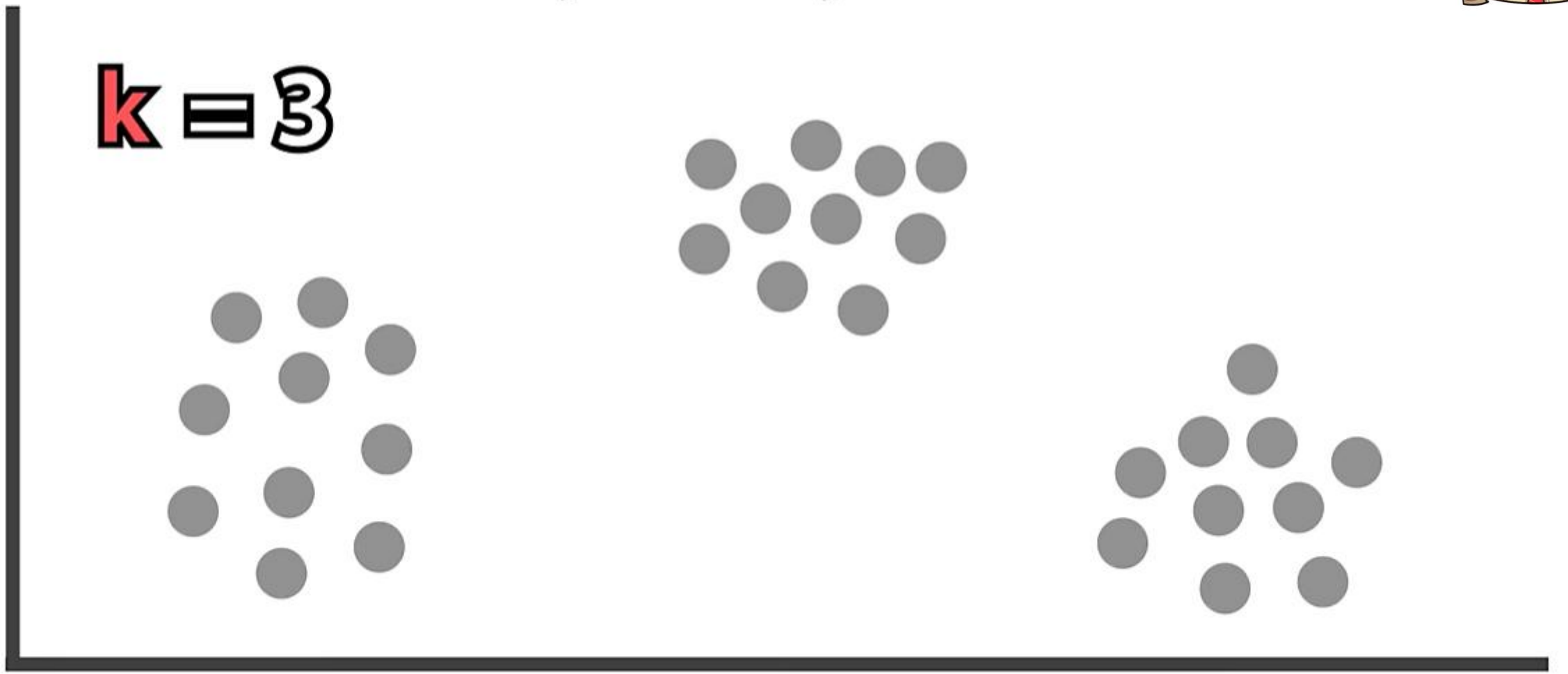
How many clusters you want to find?



K-Means

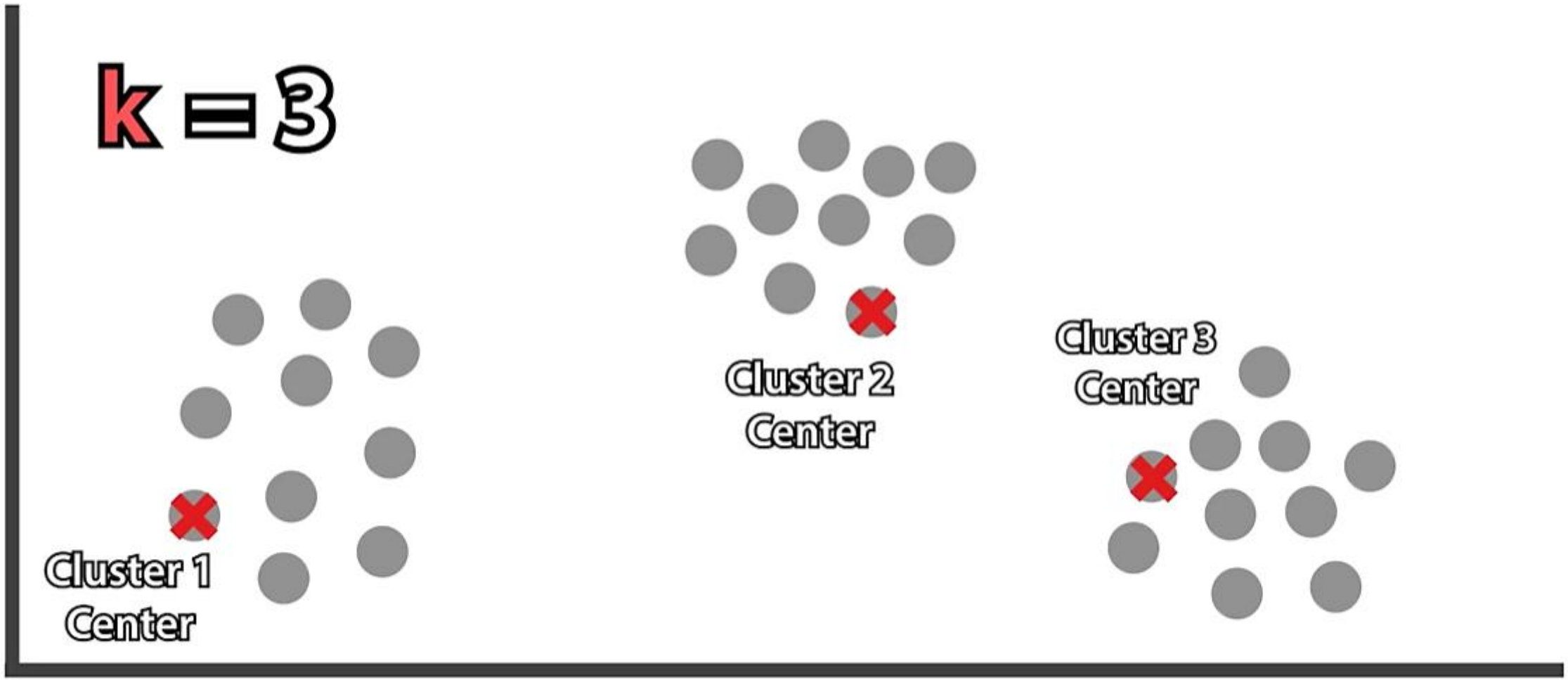
How many clusters you want to find?

$k = 3$



K-Means

Randomly choose the cluster centers



Step 2 – Assignment Step

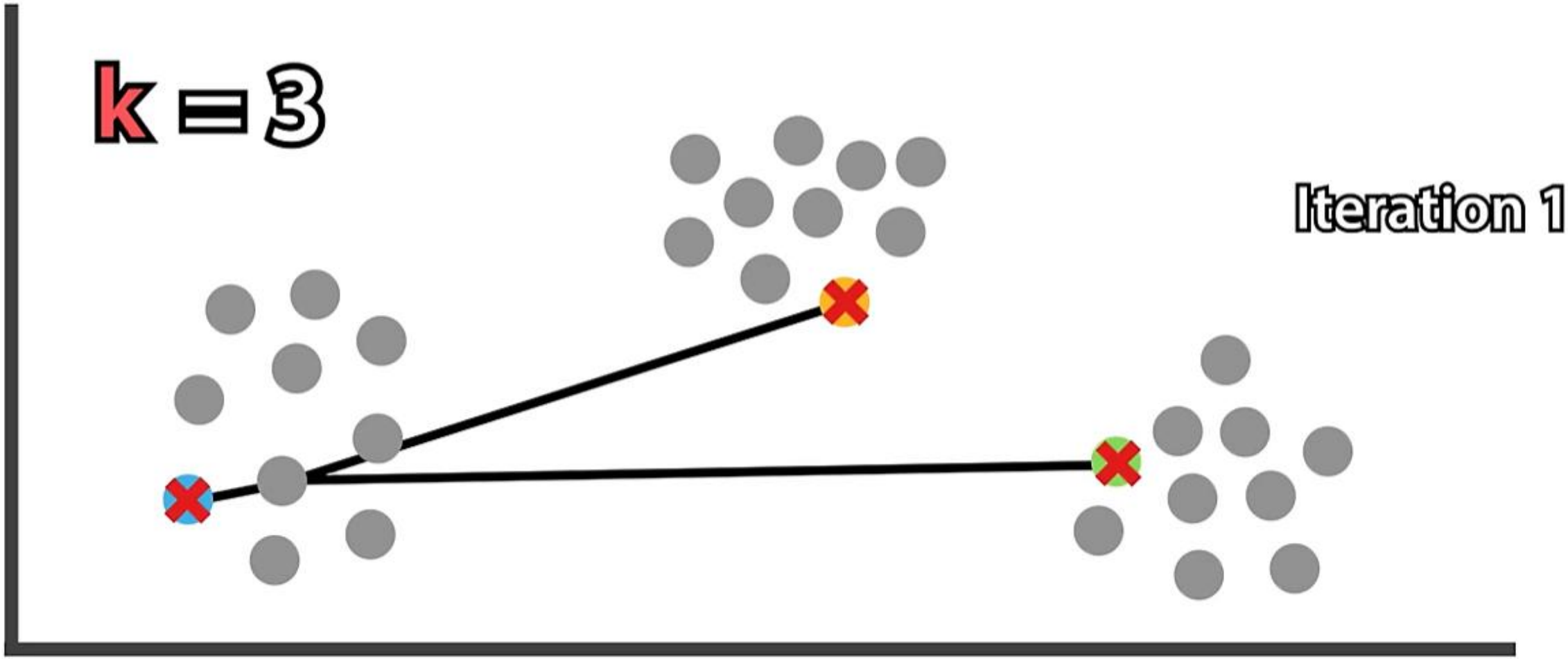
Step 2 – Assignment Step

- **Calculate Distance**: For every single data point in the dataset, the algorithm calculates its distance to each of the K centroids. The most common distance metric used is Euclidean distance .
- **Assign to Closest Centroid**: After calculating the distances, the algorithm assigns the data point to the cluster whose centroid has the minimum distance. This is the core logic that forms the clusters.

This step is repeated for all data points, effectively partitioning the dataset into K initial clusters based on their proximity to the randomly placed centroids.

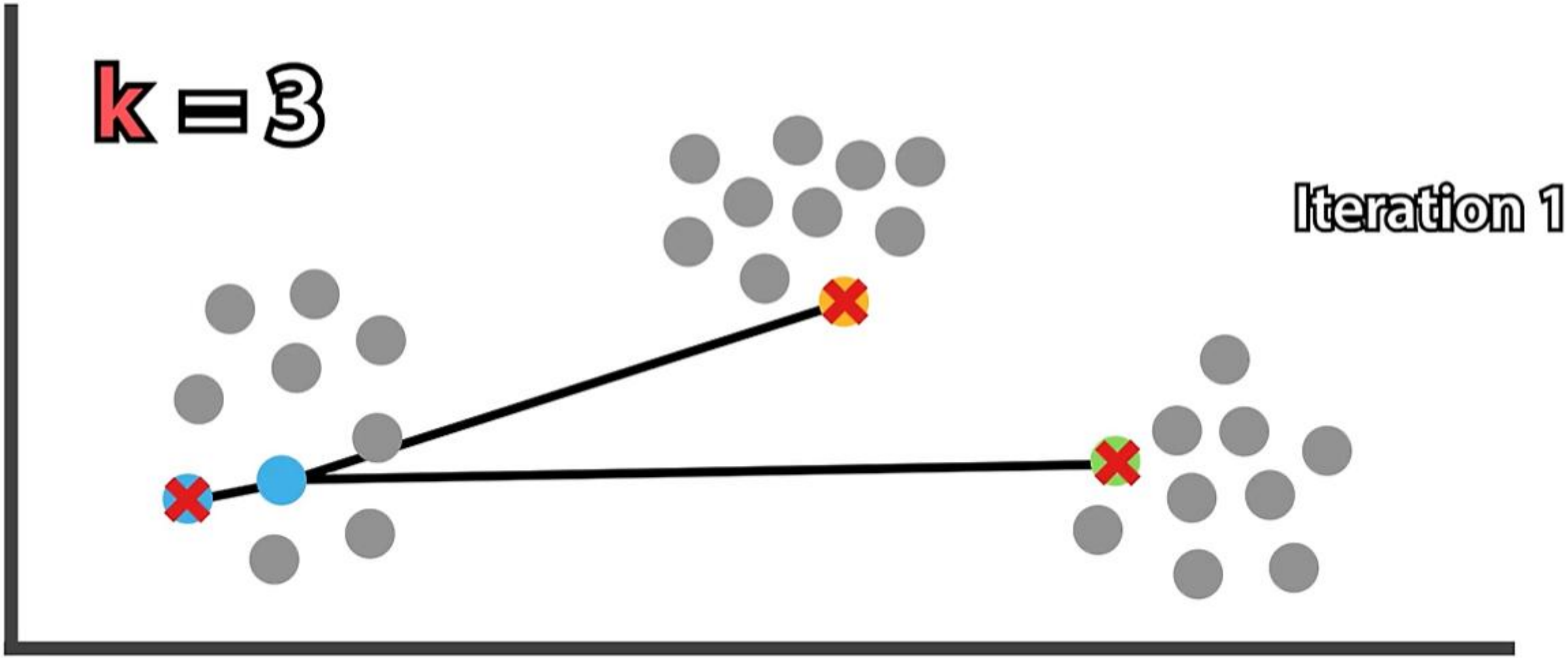
K-Means

Calculate the distance of each point from the clusters

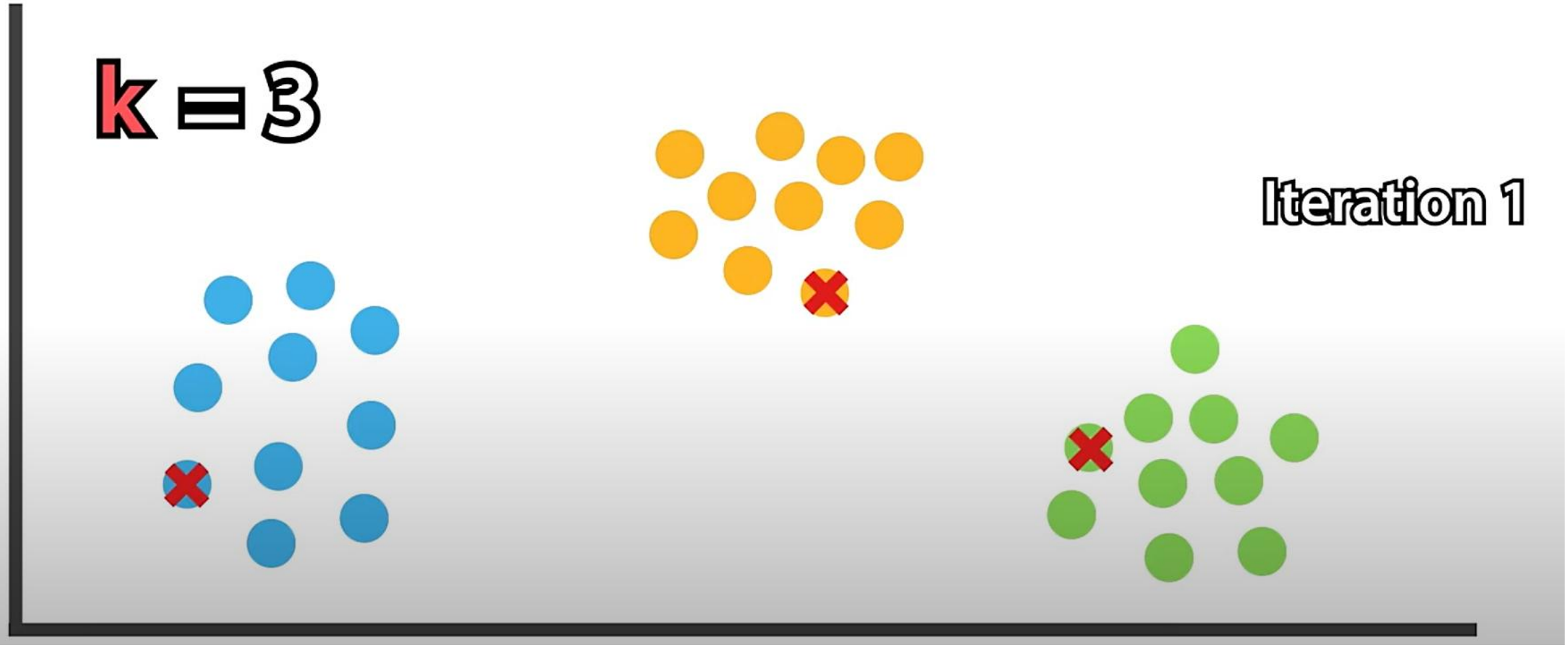


K-Means

Assign the point to the cluster with the closest centroid



K-Means



Step 3 – Update Step

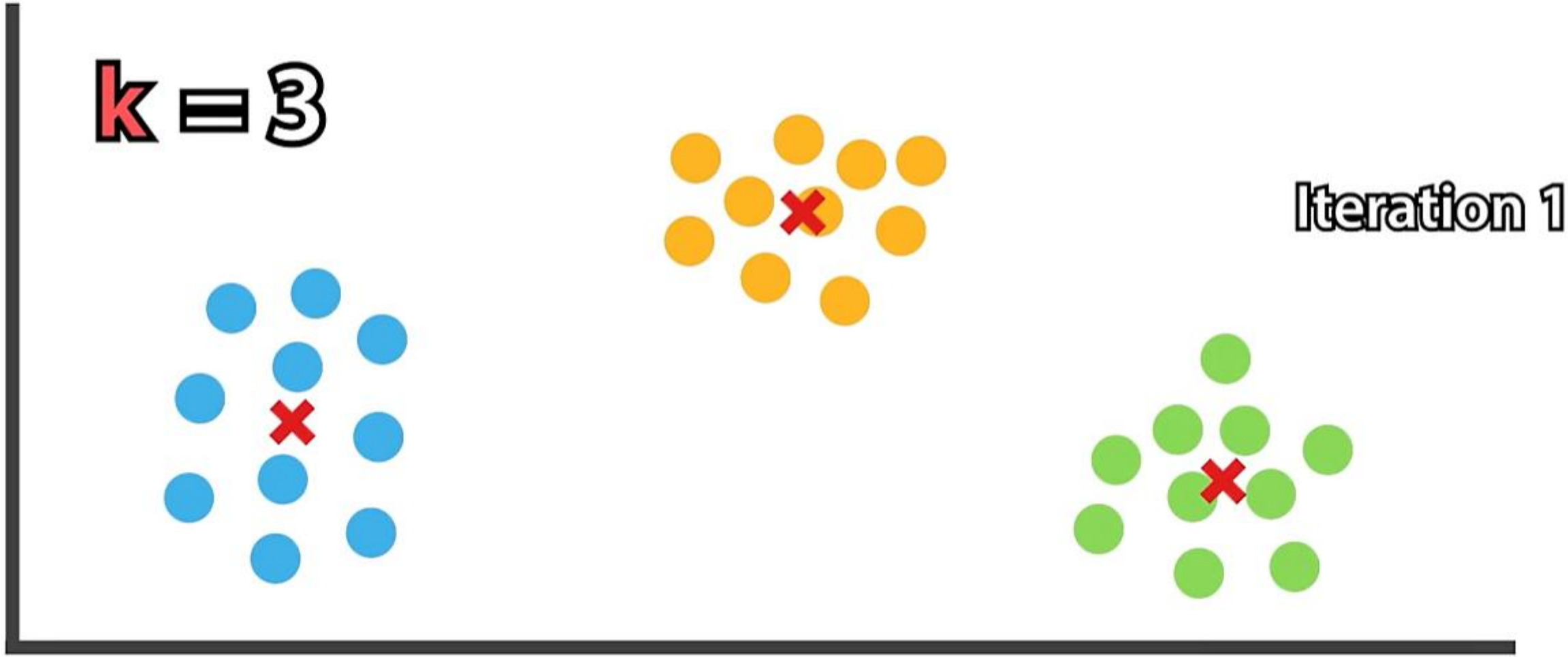
Step 3 – Update Step

- **Find the Mean:** The algorithm takes all the data points that were assigned to a particular cluster in the previous step.
- **Reposition the Centroid:** It calculates the **mean** (average) of the coordinates of all these data points. This mean becomes the new, updated position of the cluster's centroid.

This process ensures that the centroid is always at the geometric center of the points currently assigned to its cluster, which is the key to minimizing the overall within-cluster sum of squares. These two steps (assignment and update) are repeated iteratively until the centroids no longer move significantly, indicating that the algorithm has converged.

K-Means

Recalculate the centroid of each cluster



Step 4 - Iteration

Step 4 – Iteration

The fundamental steps in the K-Means algorithm's iteration phase are to **repeat the assignment and update steps** until the cluster centroids stabilize. This is the main loop that allows the algorithm to converge on a solution.

- **Assignment Step:** Each data point is assigned to its closest centroid.
- **Update Step:** The centroid of each cluster is recalculated as the mean of all the data points assigned to that cluster.

The algorithm continues to repeat these two steps. With each iteration, the centroids move to a better position, and the clusters become more refined. The process stops when one of two conditions is met:

- The centroids no longer move significantly between iterations.
- A pre-defined maximum number of iterations is reached.

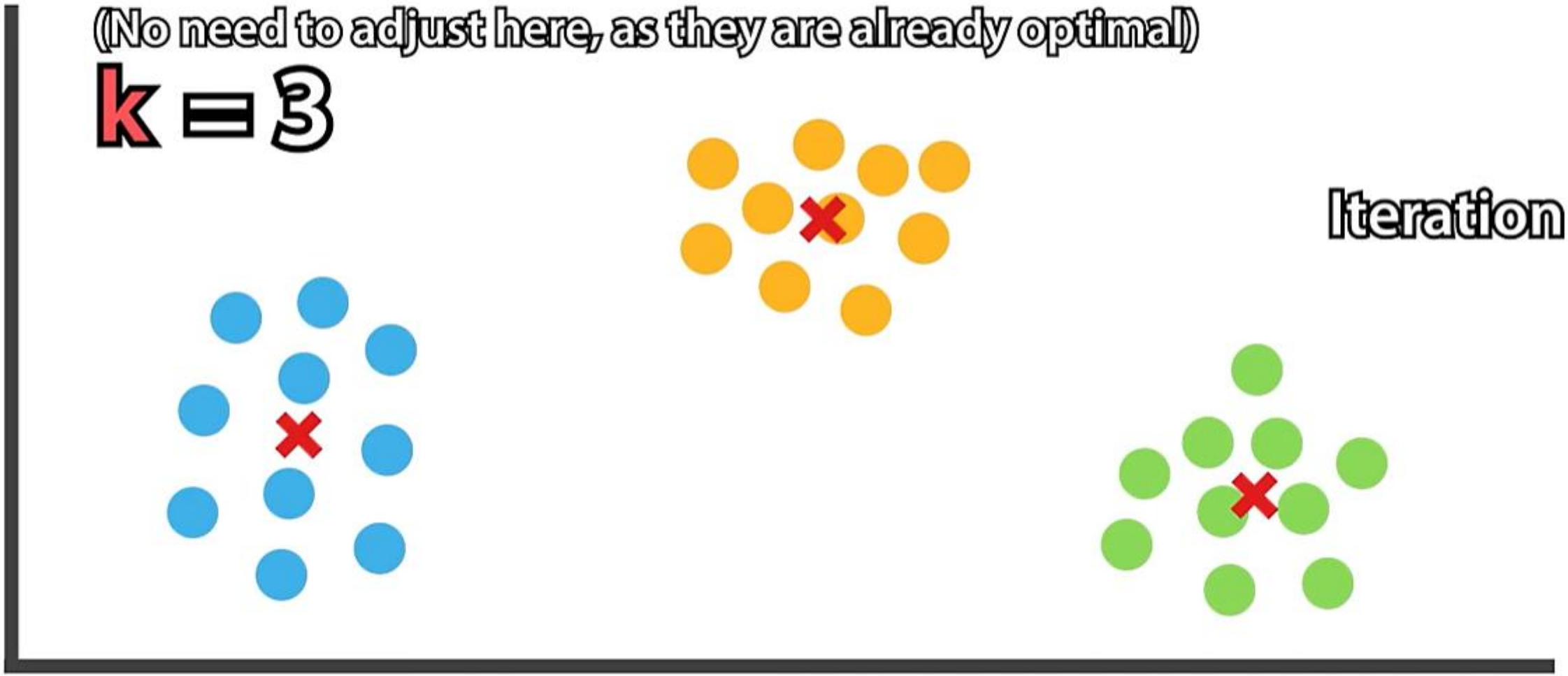
This iterative refinement process is what distinguishes K-Means from a simple, one-time data partitioning.

K-Means

Reassign points based on the new centroid position

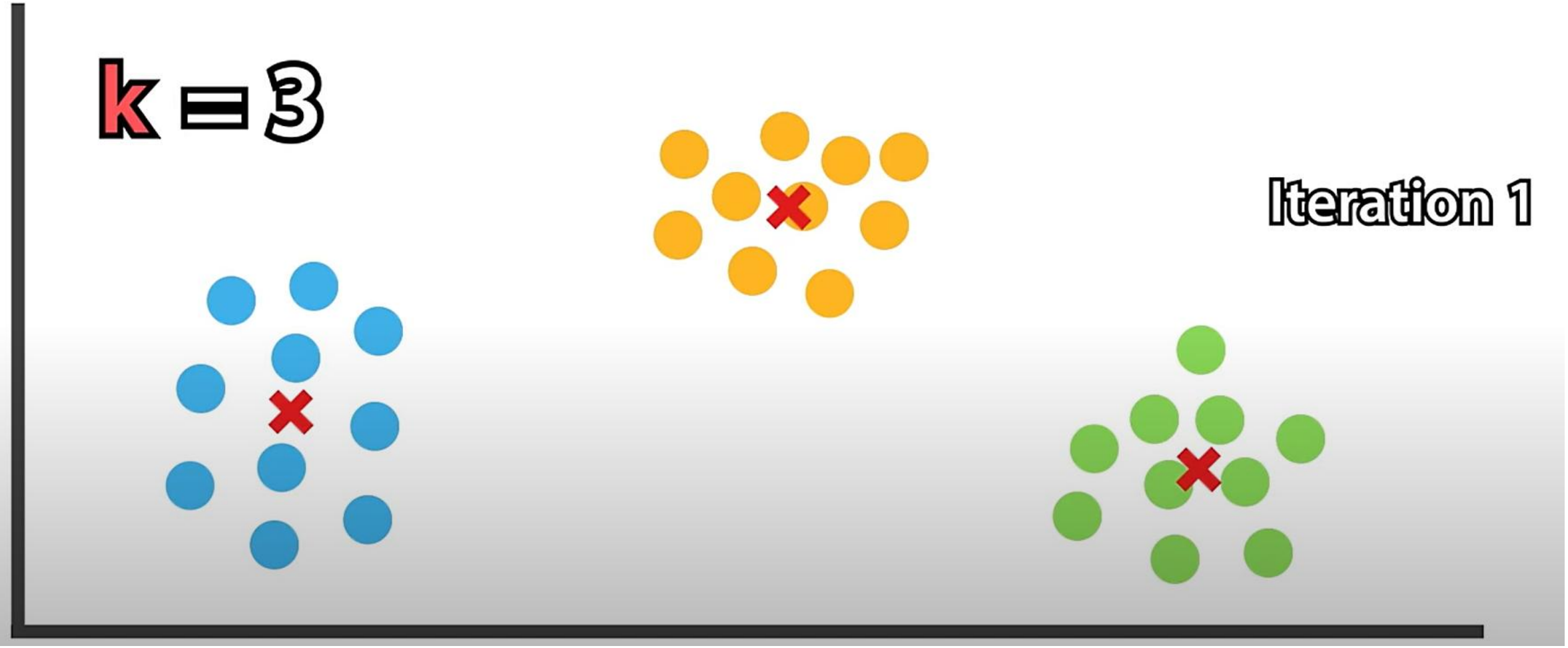
(No need to adjust here, as they are already optimal)

$k = 3$



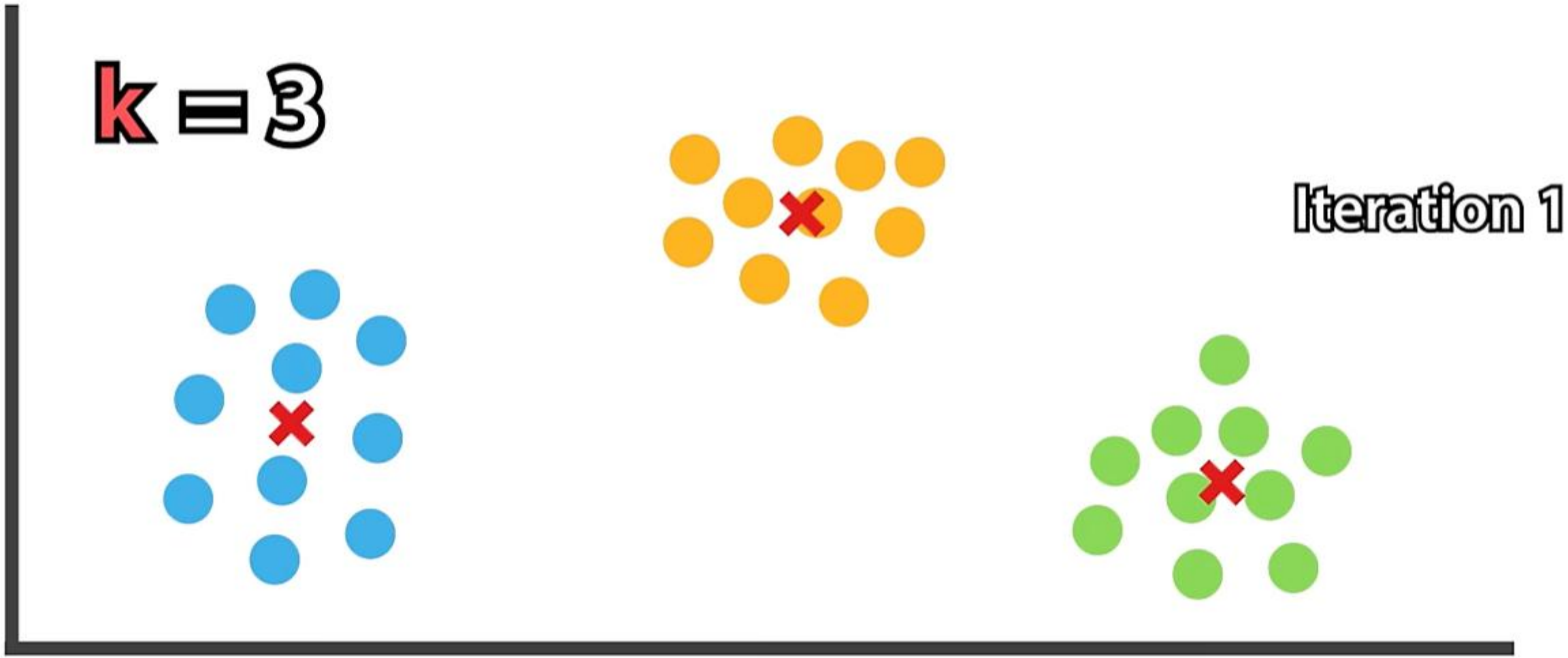
K-Means

After reassigning, calculate centroids again



K-Means

Repeat the procedure until all data points are assigned



Step 5 - Convergence

Step 5 – Convergence

The K-Means algorithm doesn't have a separate "convergence step" in the same way it has an assignment and update step. Convergence is the **stopping condition** that determines when the algorithm's iterative loop should end.

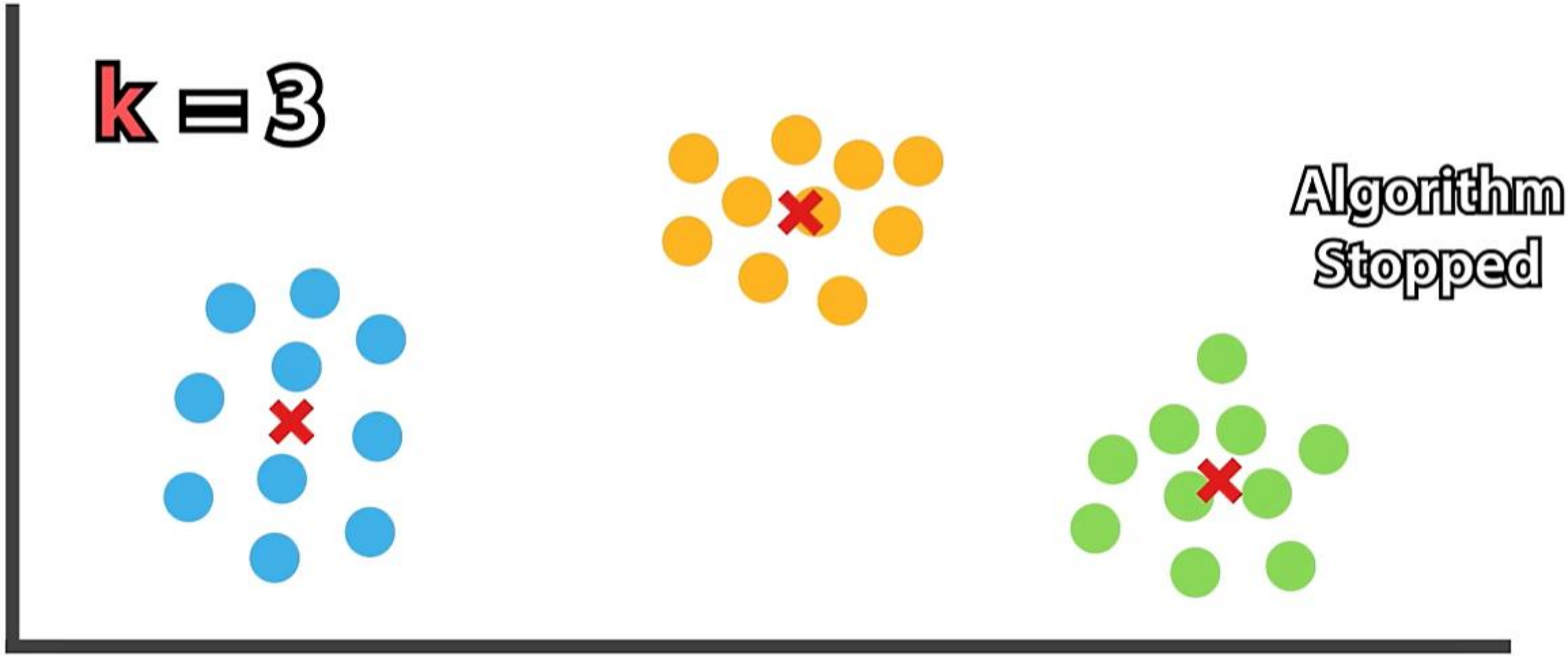
The fundamental steps in the convergence process are:

- **Monitor Centroid Movement:** After each update step, the algorithm checks how much the cluster centroids have moved from their previous positions.
- **Evaluate Stability:** The algorithm has converged when the positions of the centroids either no longer change or change by a very small, insignificant amount. This indicates that the data points have been stably assigned to their optimal clusters.

The algorithm can also be stopped if it reaches a pre-defined maximum number of iterations, even if the centroids are still moving slightly.

K-Means

Algorithm stops when centroids no longer move significantly



LET’S GO