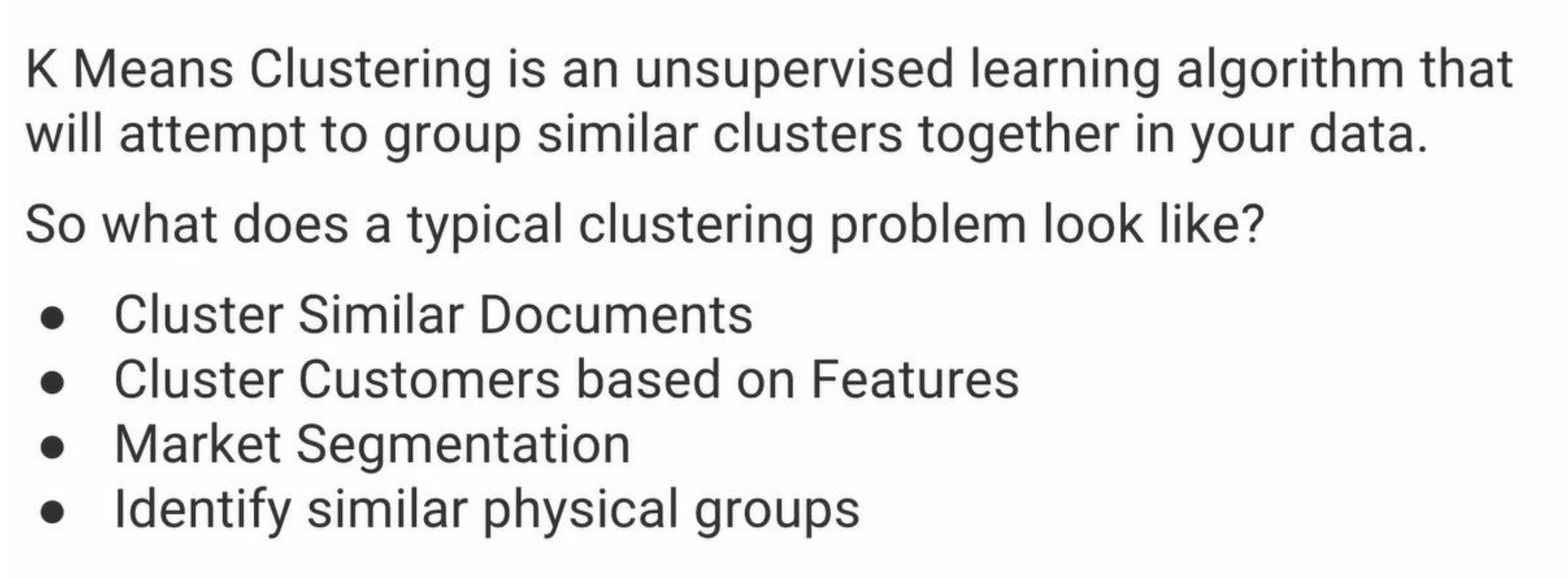
KMeans

KMeans

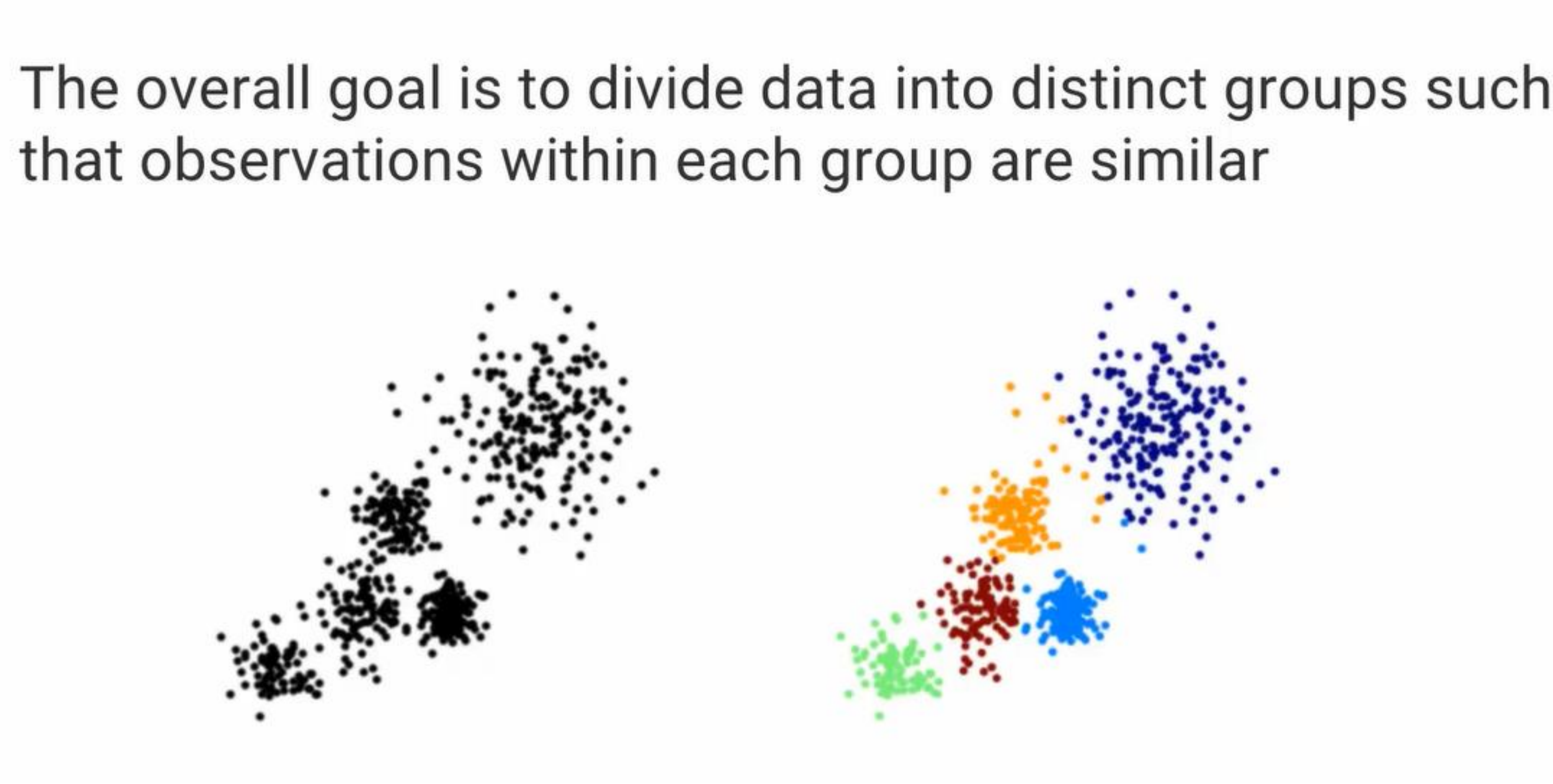


K Means Clustering is an unsupervised learning algorithm that will attempt to group similar clusters together in your data.

So what does a typical clustering problem look like?

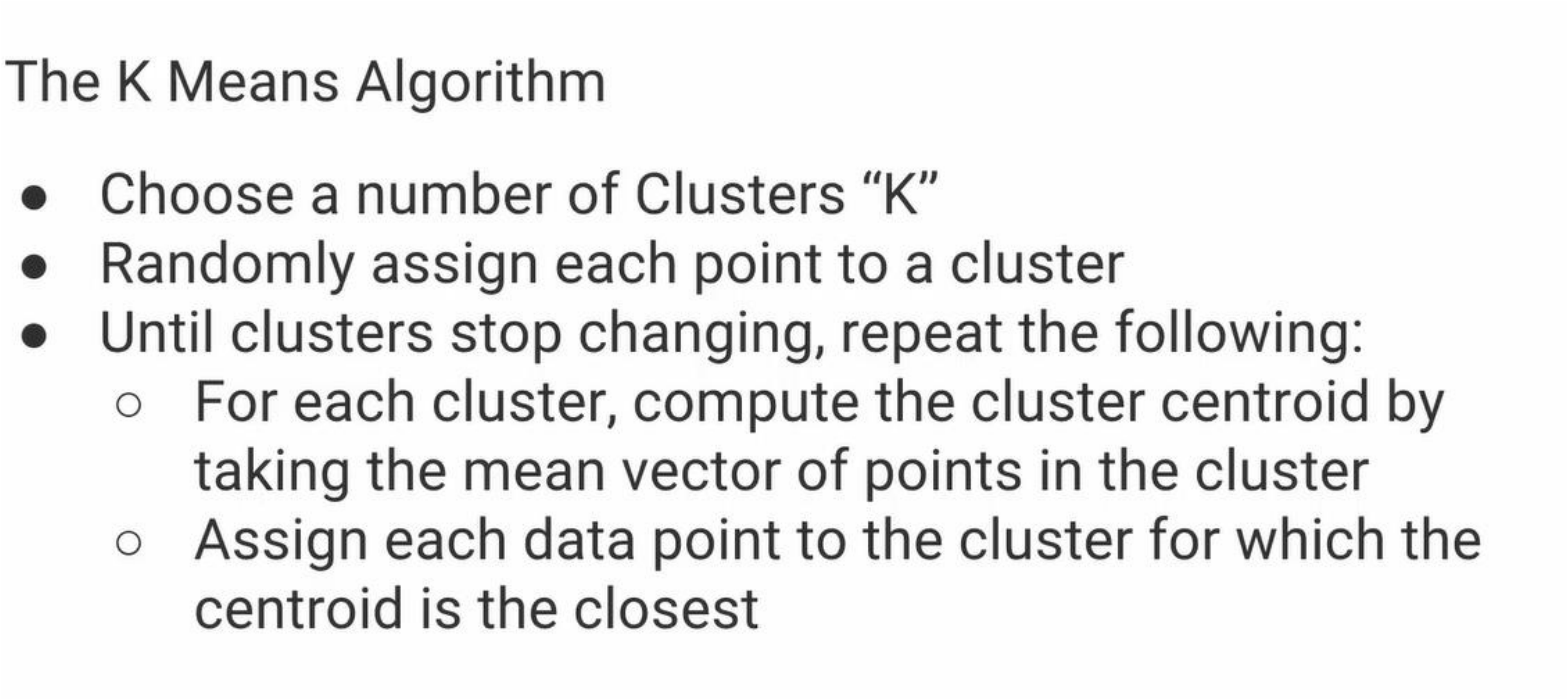
- Cluster Similar Documents
- Cluster Customers based on Features
- Market Segmentation
- Identify similar physical groups

KMeans



The overall goal is to divide data into distinct groups such that observations within each group are similar

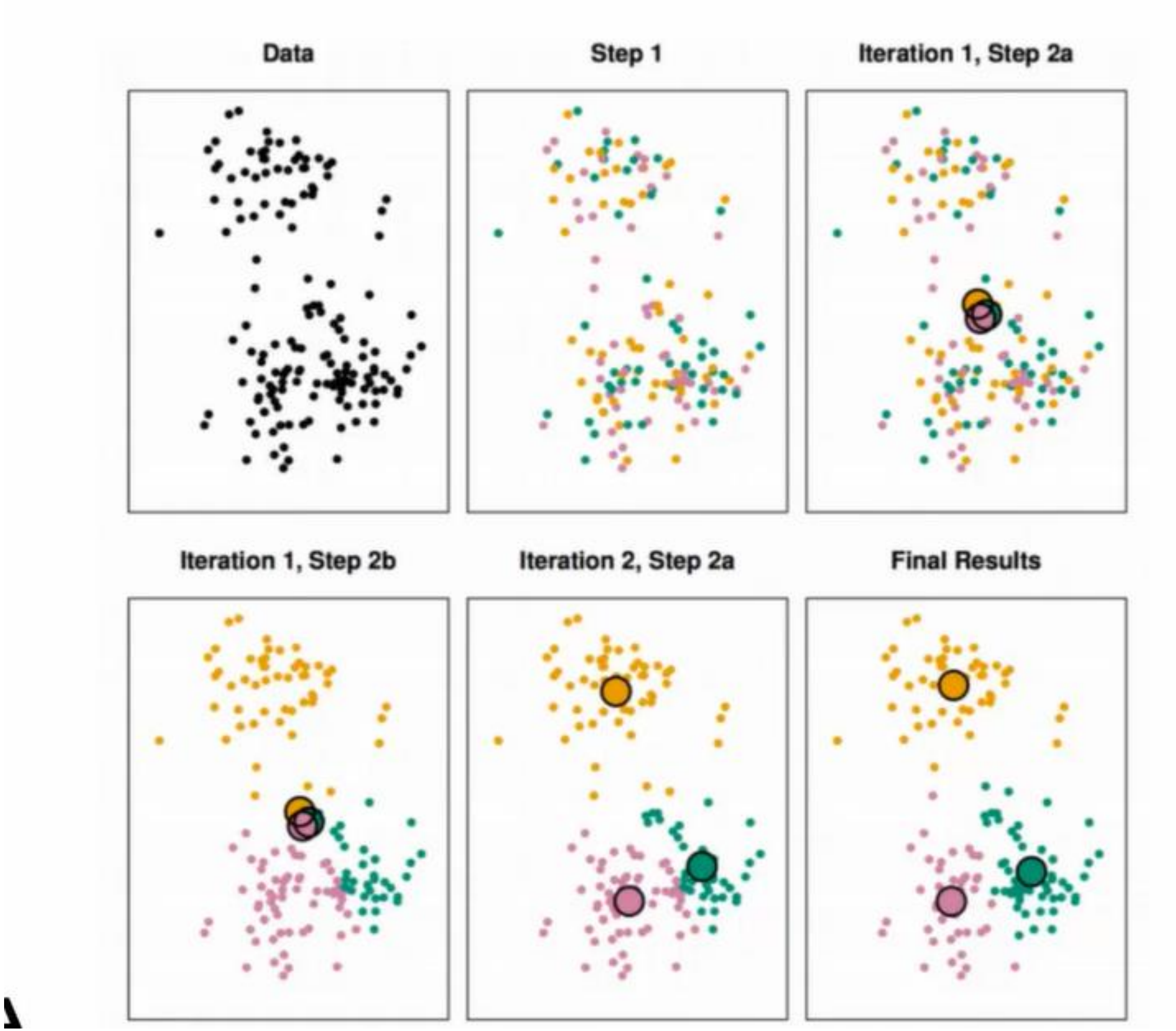
KMeans



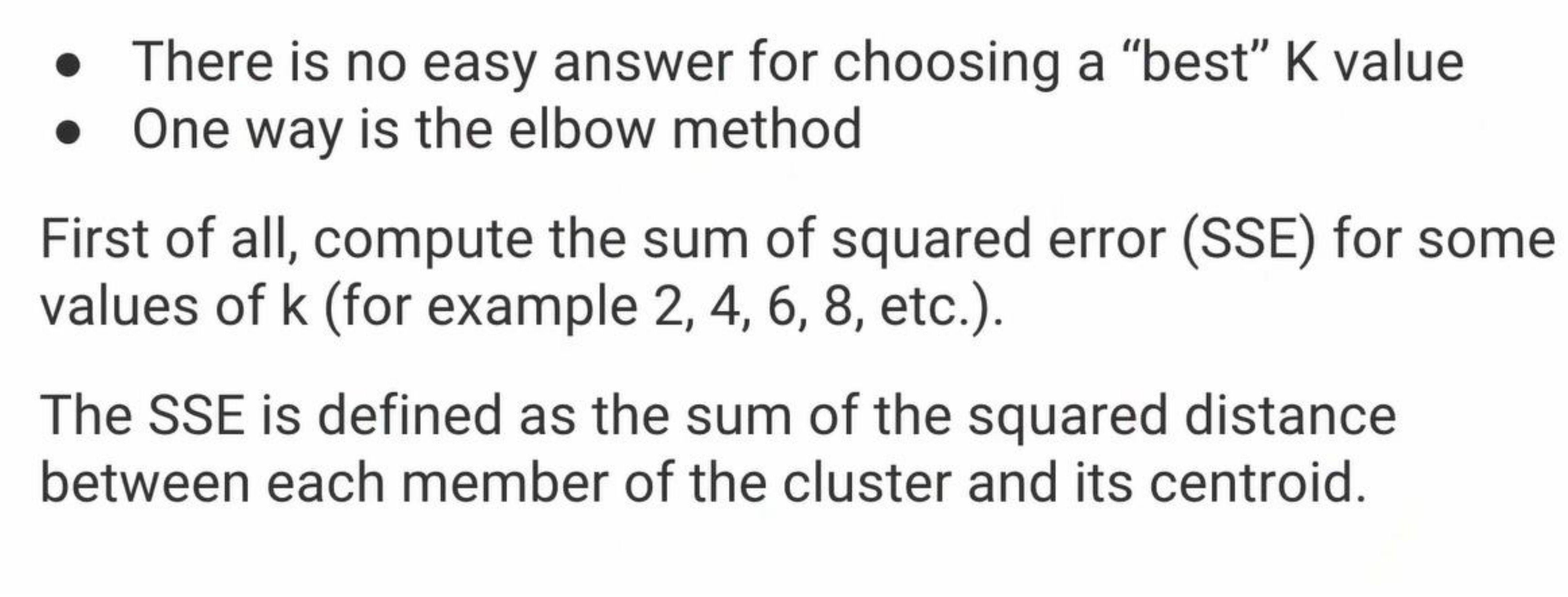
The K Means Algorithm

- Choose a number of Clusters “K”
- Randomly assign each point to a cluster
- Until clusters stop changing, repeat the following:
  - For each cluster, compute the cluster centroid by taking the mean vector of points in the cluster
  - Assign each data point to the cluster for which the centroid is the closest

KMeans


Data
Step 1
Iteration 1, Step 2a
Iteration 1, Step 2b
Iteration 2, Step 2a
Final Results

KMeans

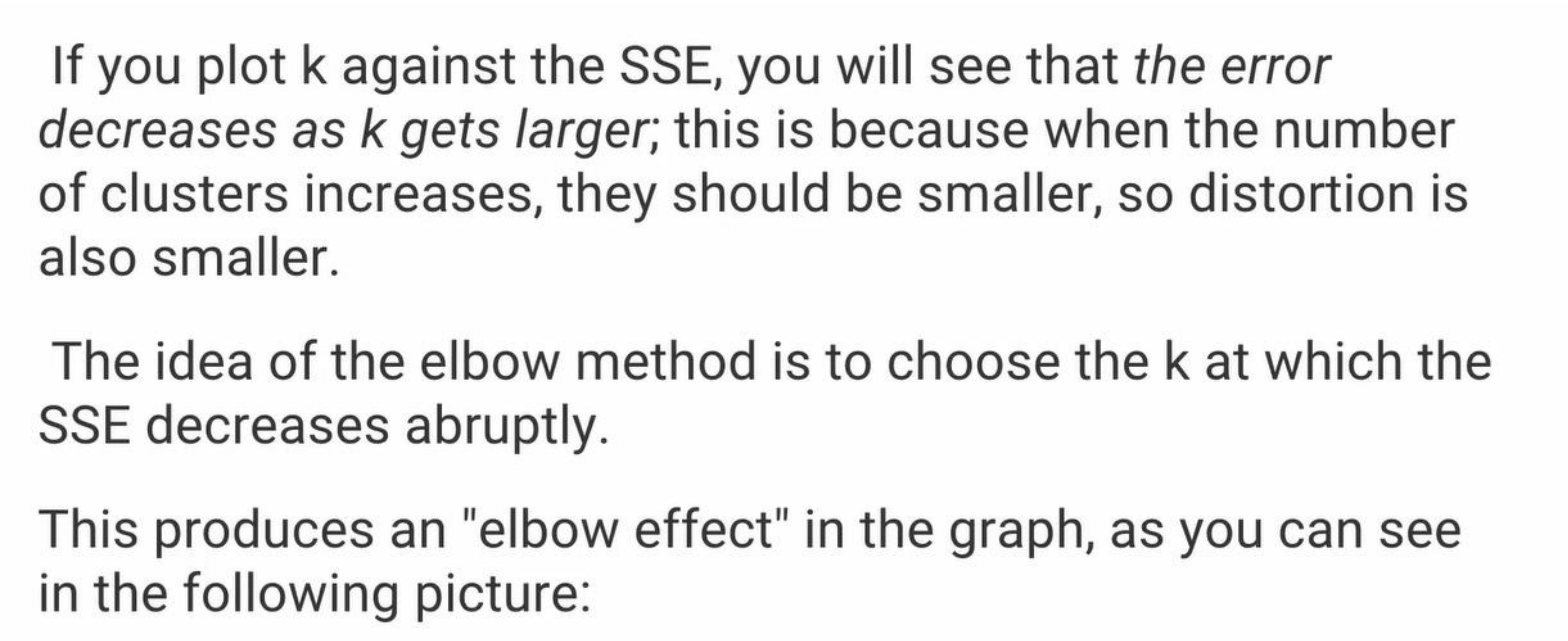


- There is no easy answer for choosing a “best” K value
- One way is the elbow method

First of all, compute the sum of squared error (SSE) for some values of k (for example 2, 4, 6, 8, etc.).

The SSE is defined as the sum of the squared distance between each member of the cluster and its centroid.

KMeans

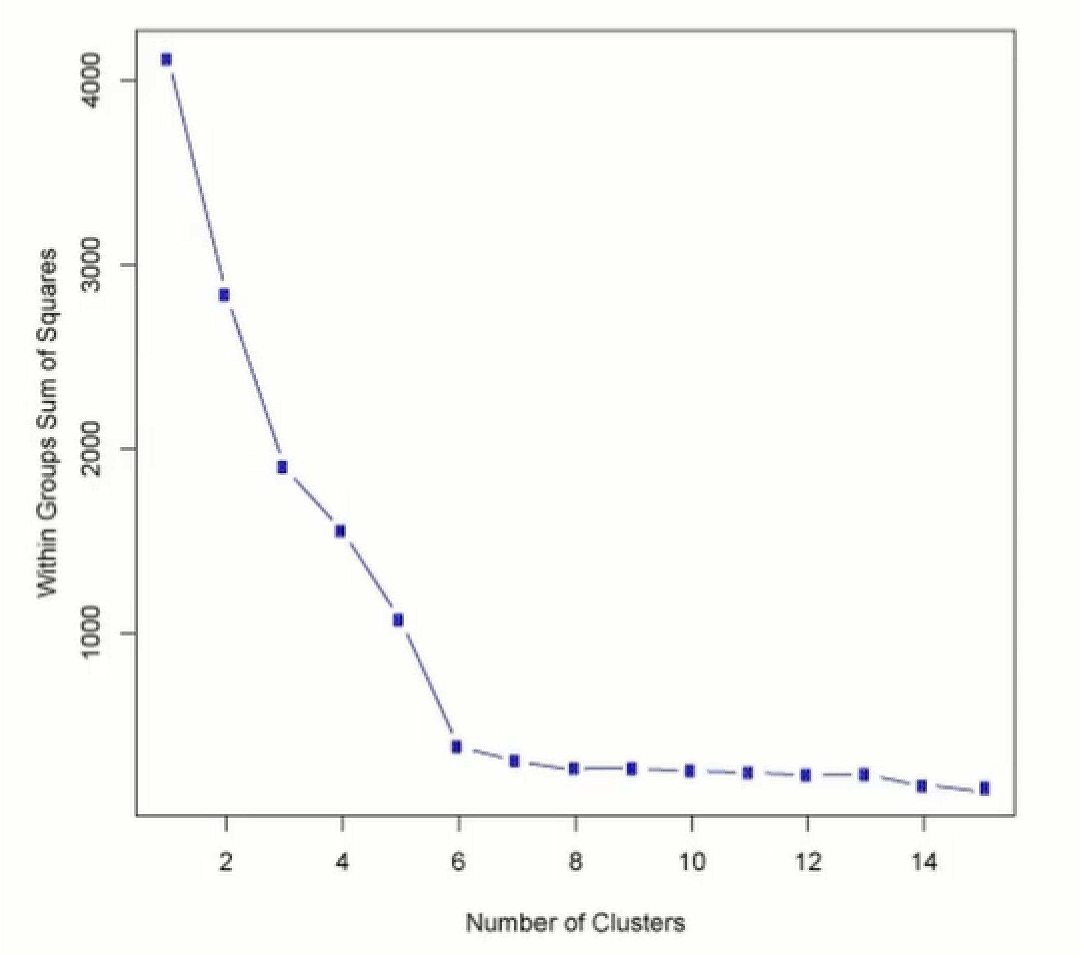


If you plot k against the SSE, you will see that *the error decreases as k gets larger*; this is because when the number of clusters increases, they should be smaller, so distortion is also smaller.

The idea of the elbow method is to choose the k at which the SSE decreases abruptly.

This produces an "elbow effect" in the graph, as you can see in the following picture:

KMeans


Within Groups Sum of Squares
4000
3000
2000
1000
2
4
6
8
10
12
14
Number of Clusters

What is Clustering Algorithm??

- K means clustering algorithm is a very common unsupervised learning algorithm. This algorithm clusters n objects into k clusters, where each object belongs to a cluster with the nearest mean.

- Clustering is nothing but dividing a set of data into groups of similar points or features, where data points in the same group are as similar as possible and data points in different groups are as dissimilar as possible. We use clustering in our day-to-day lives; for instance, in a supermarket, all vegetables are grouped as one group and all fruits are grouped as another group. This clustering helps customers fasten their shopping process.

What Is K means Clustering Algorithm?

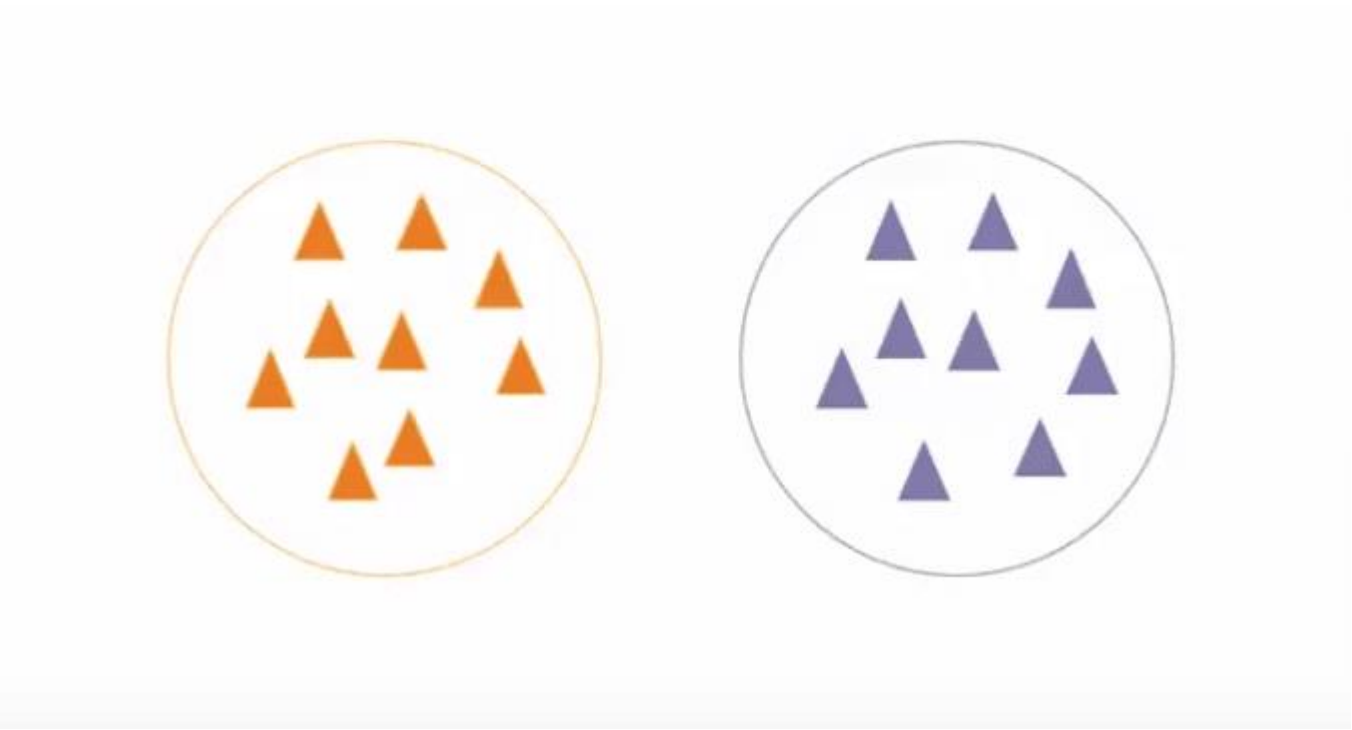
- K means clustering is an algorithm, where the main goal is to group similar data points into a cluster. In K means clustering, *k* represents the total number of groups or clusters. K means clustering runs on Euclidean distance calculation. Now, let us understand K means clustering with the help of an example.
- Say, we have a dataset consisting of height and weight information of 10 players. We need to group them into two clusters based on their height and weight.

Example

- Another clustering example we might have come across is the Amazon or Flipkart product recommendation. Amazon or Flipkart recommend us products based on our previous search. How do they do it? Well, the concept behind this is clustering.
- Now that we know what clustering is, let us discuss the categories that we have in clustering.

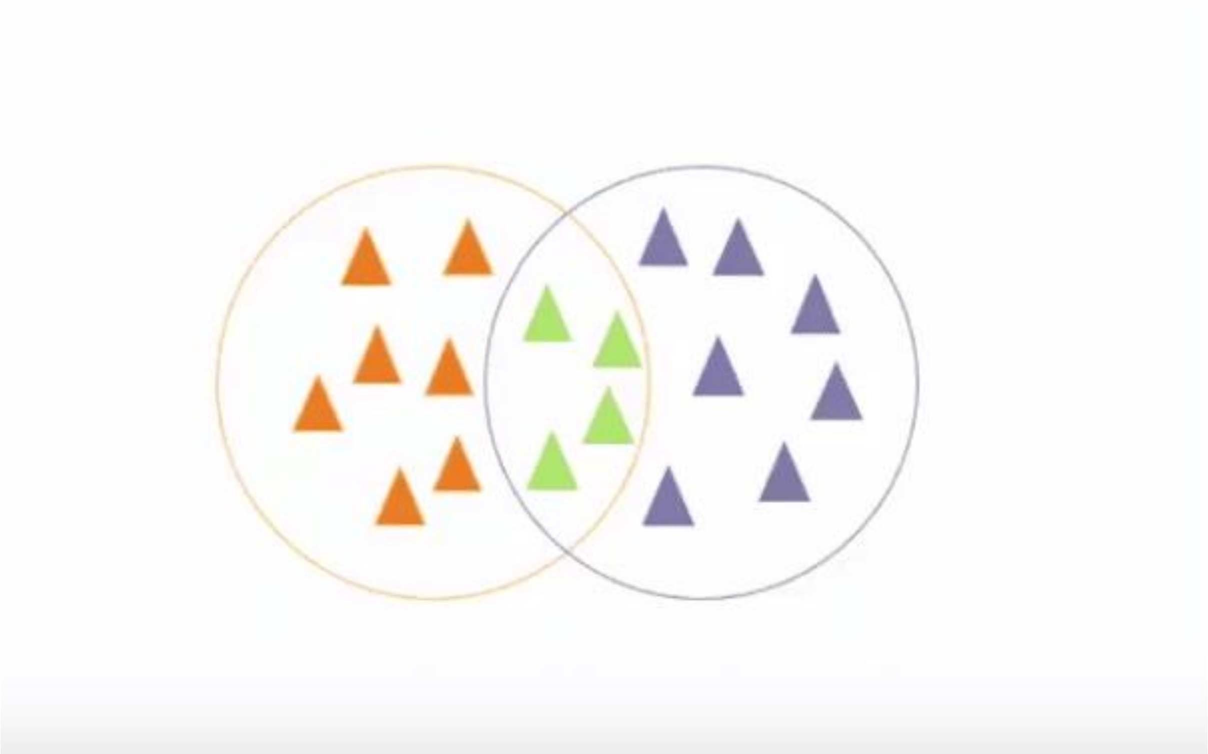
Types of Clustering

- Exclusive clustering: In exclusive clustering, data are grouped in an exclusive way so that a certain datum belongs to only one definite cluster.



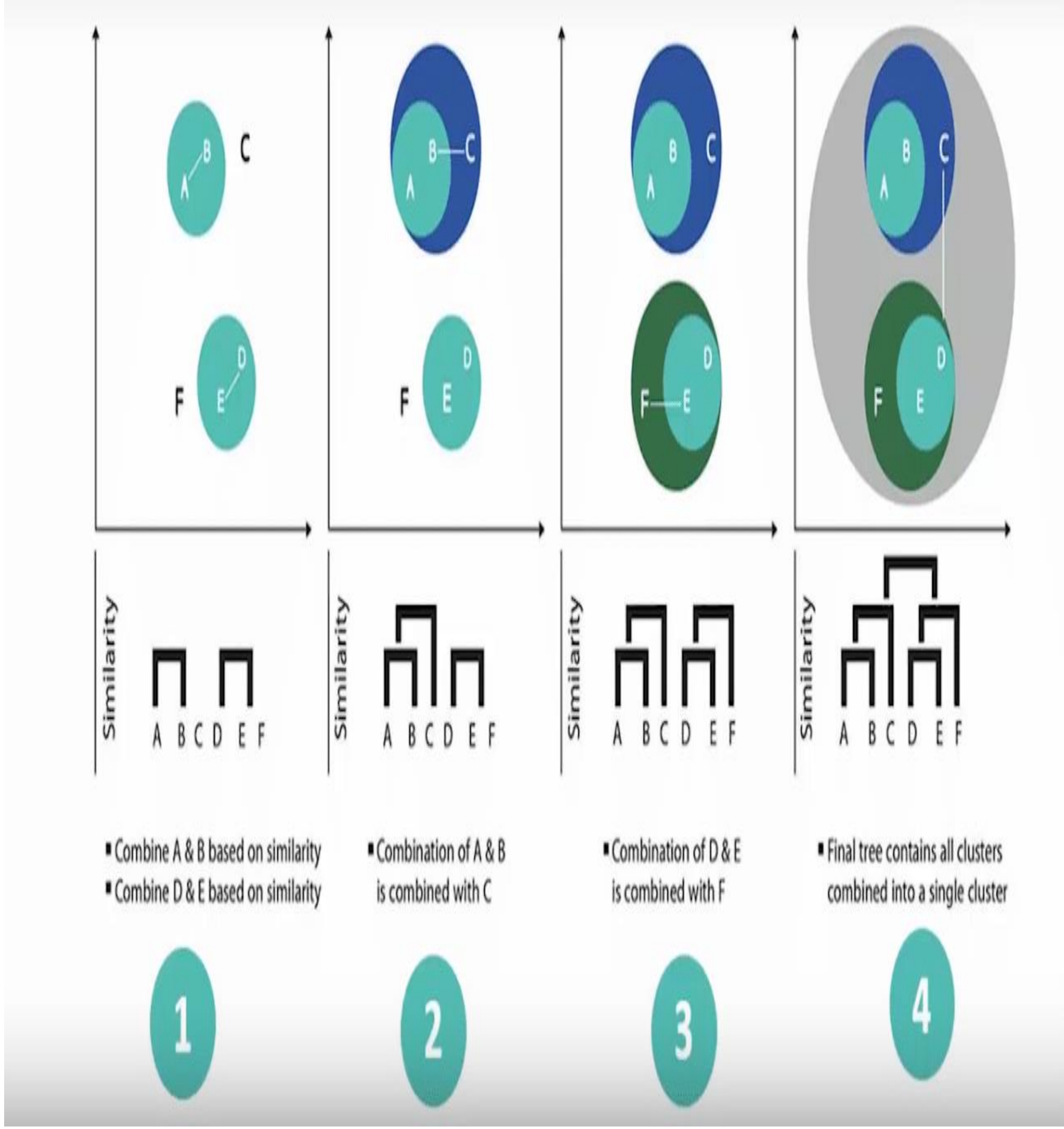
Types of Clustering:

- Overlapping clustering: In overlapping clustering, each point may belong to two or more clusters.

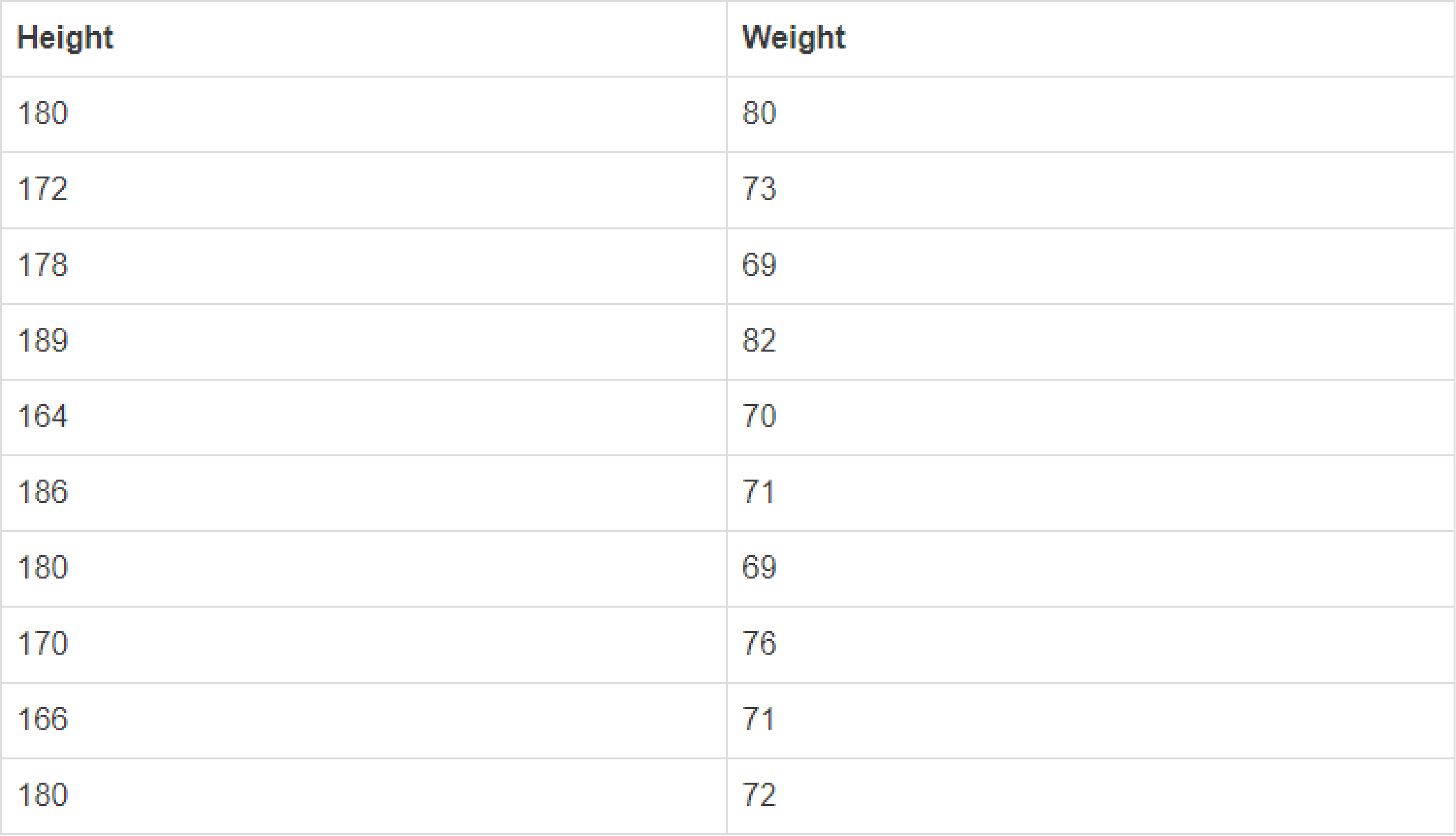


Types of Clustering

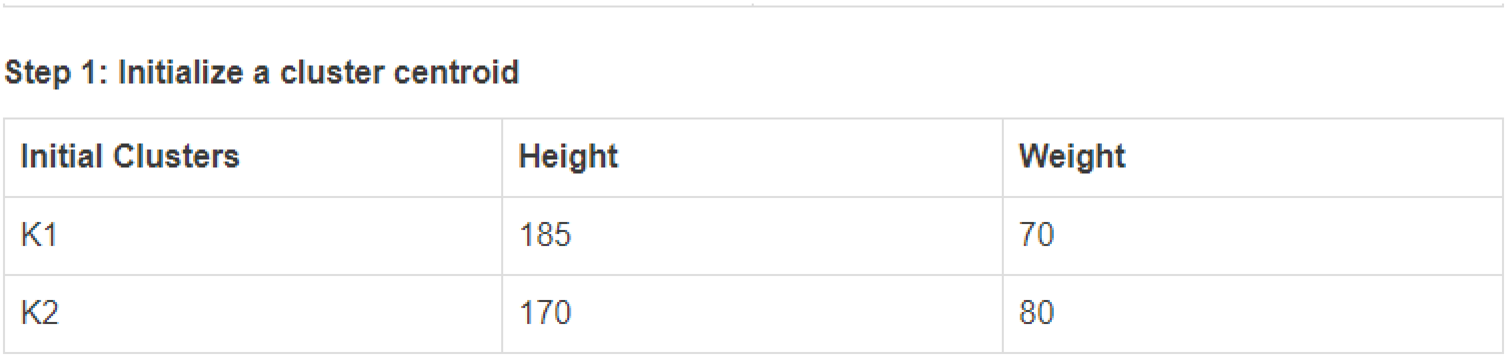
- Hierarchical clustering: In this technique, the first step is to assign all data points clusters of their own. The second step is to merge two nearer clusters into one cluster. The third step is to compute distances between the new cluster and each of the old clusters. Again, repeat the second and third steps until only one cluster is left.


A
B
C
D
E
F
Similarity
A B C D E F
Combine A & B based on similarity
Combine D & E based on similarity
Combination of A & B is combined with C
Combination of D & E is combined with F
Final tree contains all clusters combined into a single cluster
1
2
3
4

Table

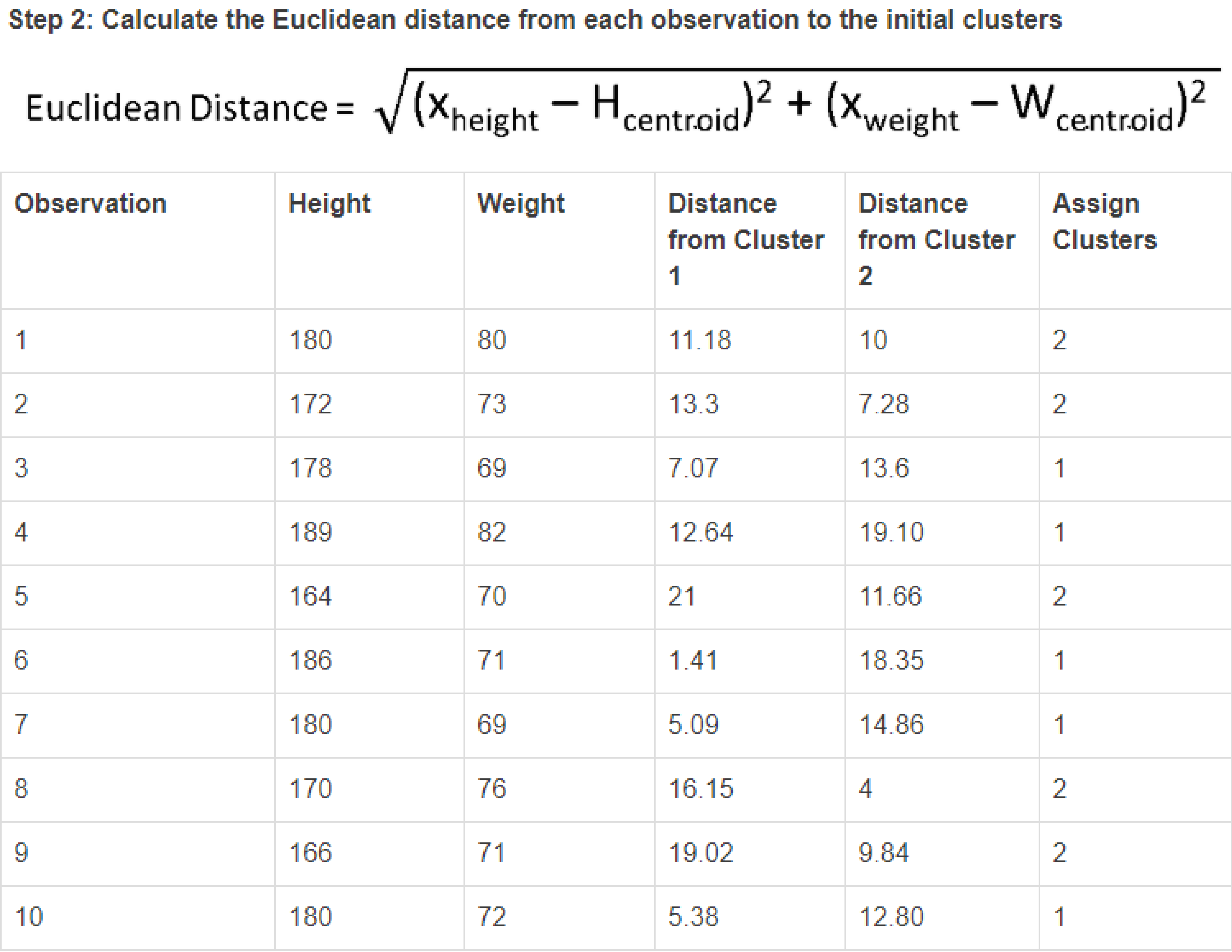


Height	Weight
180	80
172	73
178	69
189	82
164	70
186	71
180	69
170	76
166	71
180	72



Step 1: Initialize a cluster centroid

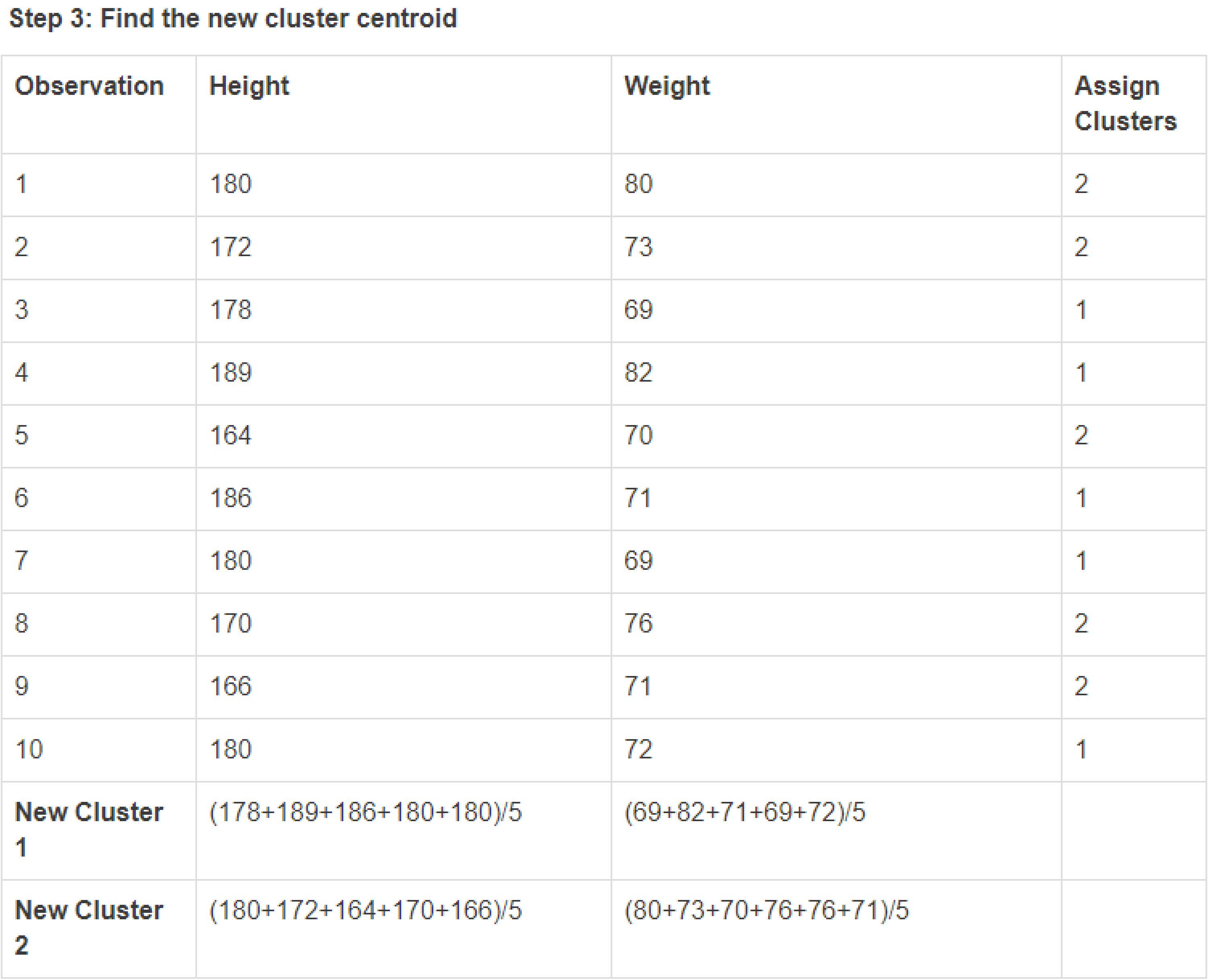
Initial Clusters	Height	Weight
K1	185	70
K2	170	80



Step 2: Calculate the Euclidean distance from each observation to the initial clusters

$$\text{Euclidean Distance} = \sqrt{(x_{height} - H_{centroid})^2 + (x_{weight} - W_{centroid})^2}$$

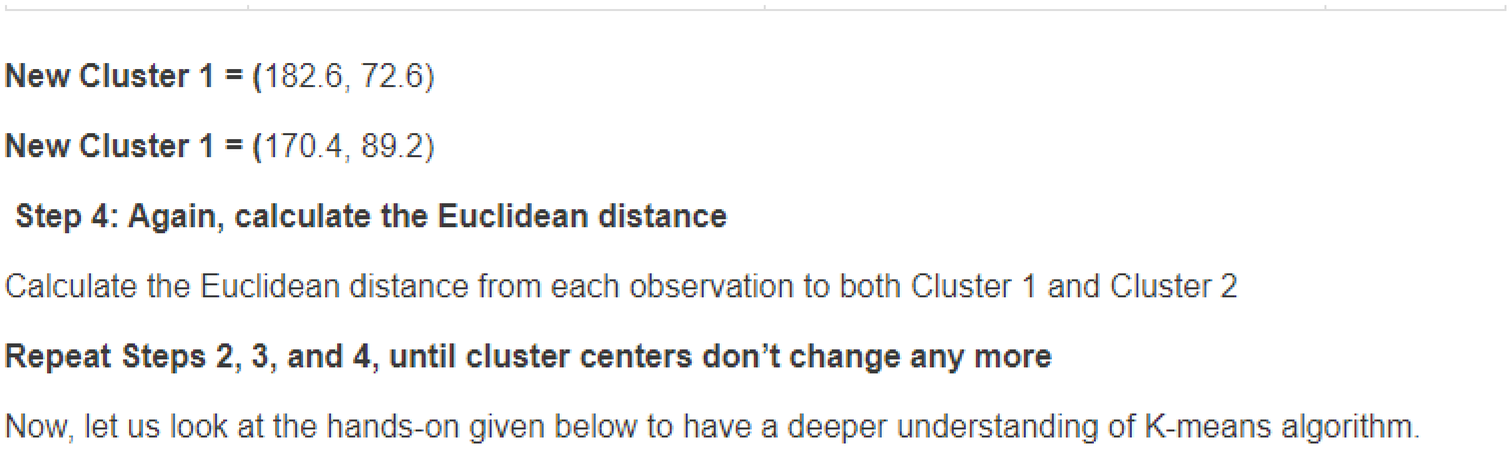
Observation	Height	Weight	Distance from Cluster 1	Distance from Cluster 2	Assign Clusters
1	180	80	11.18	10	2
2	172	73	13.3	7.28	2
3	178	69	7.07	13.6	1
4	189	82	12.64	19.10	1
5	164	70	21	11.66	2
6	186	71	1.41	18.35	1
7	180	69	5.09	14.86	1
8	170	76	16.15	4	2
9	166	71	19.02	9.84	2
10	180	72	5.38	12.80	1



Step 3: Find the new cluster centroid

Observation	Height	Weight	Assign Clusters
1	180	80	2
2	172	73	2
3	178	69	1
4	189	82	1
5	164	70	2
6	186	71	1
7	180	69	1
8	170	76	2
9	166	71	2
10	180	72	1
New Cluster 1	(178+189+186+180+180)/5	(69+82+71+69+72)/5	
New Cluster 2	(180+172+164+170+166)/5	(80+73+70+76+76+71)/5	

Repeat the Same Steps



New Cluster 1 = (182.6, 72.6)

New Cluster 1 = (170.4, 89.2)

Step 4: Again, calculate the Euclidean distance

Calculate the Euclidean distance from each observation to both Cluster 1 and Cluster 2

Repeat Steps 2, 3, and 4, until cluster centers don’t change any more

Now, let us look at the hands-on given below to have a deeper understanding of K-means algorithm.