What Is K-means Clustering?

by Elitsa Kaloyanova

24 Apr 2023 5 min read

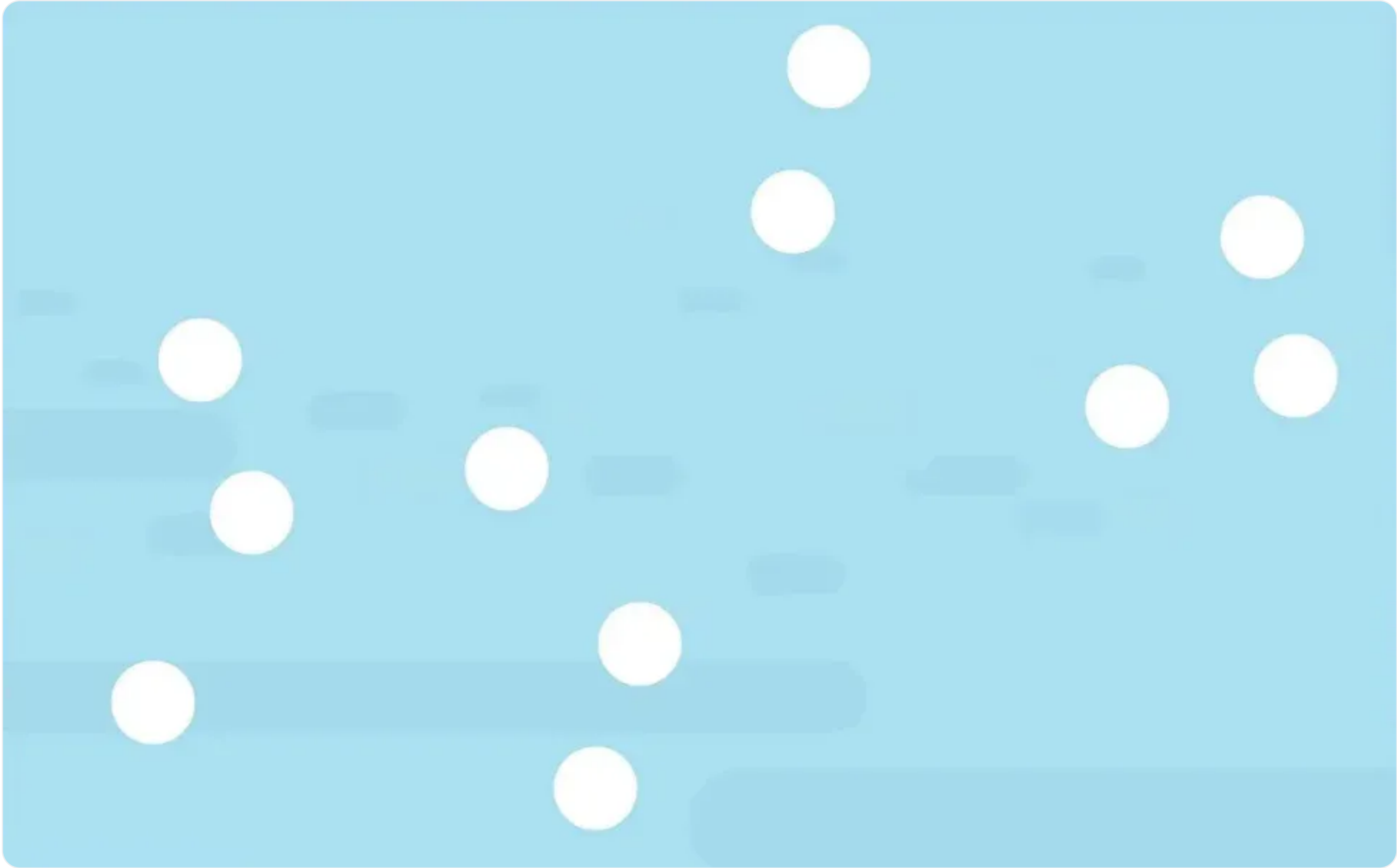
When trying to analyze data, one approach might be to look for meaningful groups or clusters. Clustering is dividing data into groups based on similarity. And K-means is one of the most commonly used methods in clustering. Why?

The main reason is its simplicity.

In this tutorial, we'll start with the theoretical foundations of the K-means algorithm, we'll discuss how it works and what pitfalls to avoid. Then, we'll see a practical application of

How Does K-means Clustering Work?

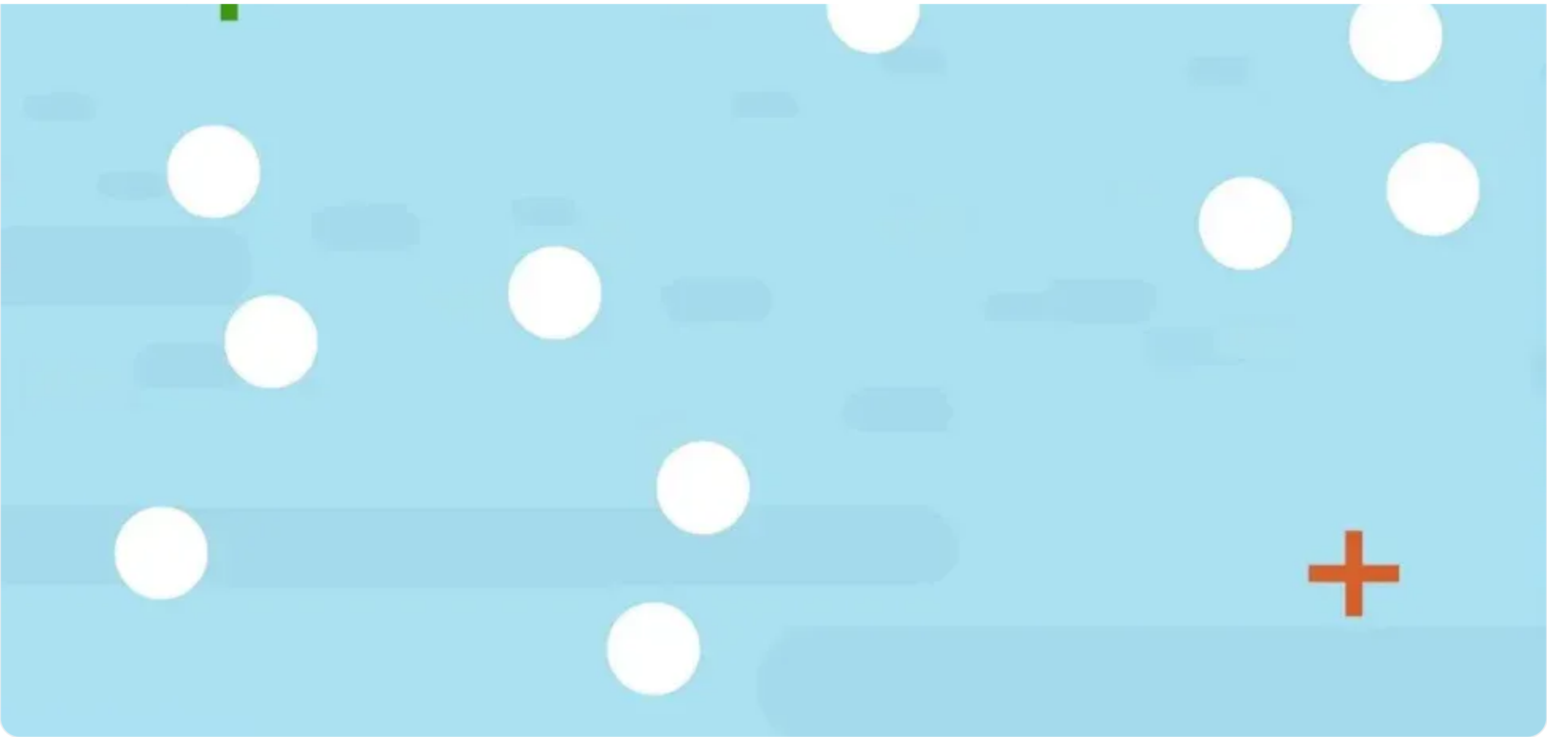
Let’s say we’d like to divide the following points into clusters.



First, we must choose how many clusters we’d like to have. The K in ‘K-means’ stands for the number of clusters we’re trying to identify. In fact, that’s where this method gets its name from. We can start by choosing two clusters.

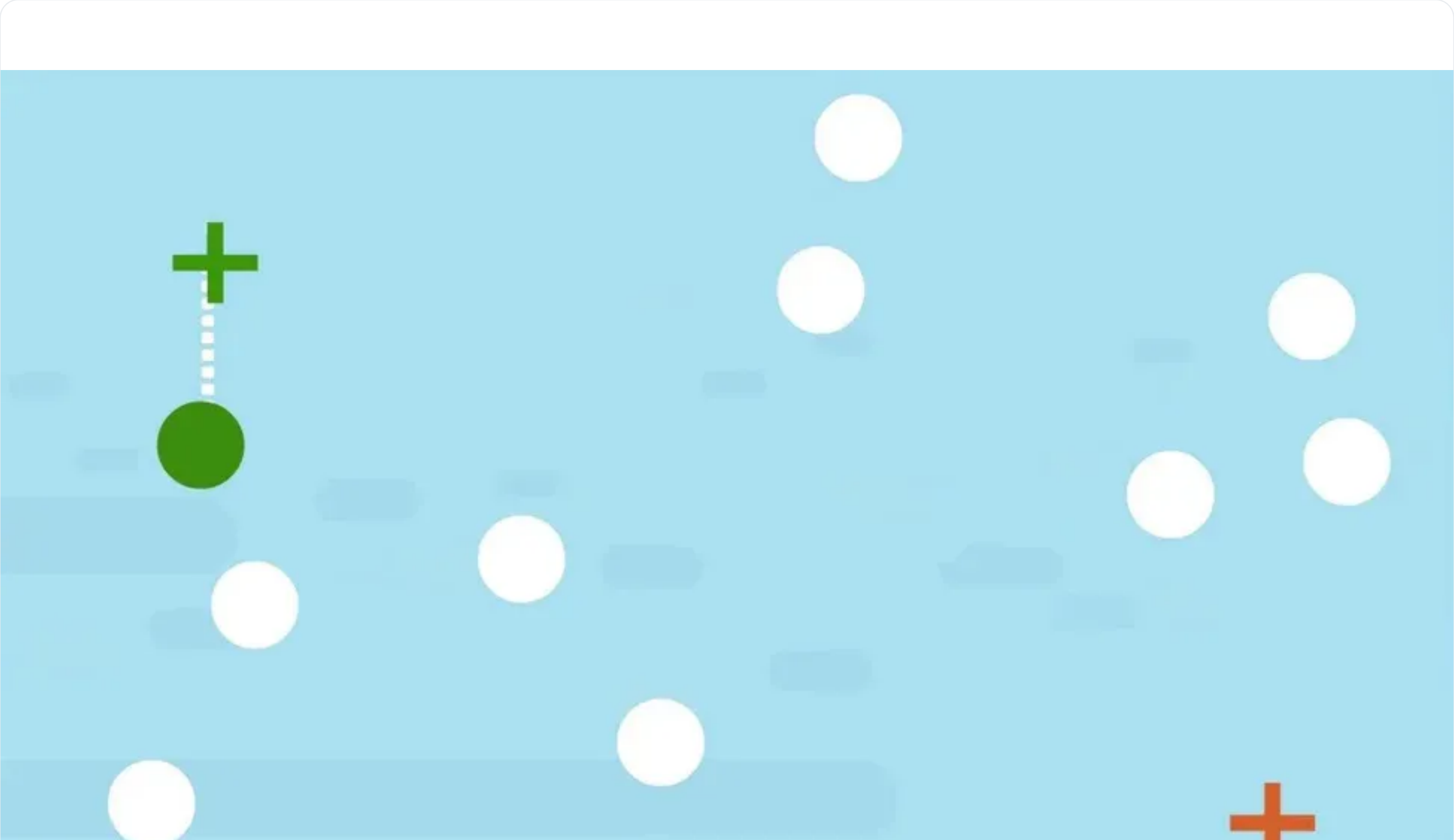
The second step is to specify the cluster seeds. A seed is basically a starting cluster centroid. It is chosen at random or is specified by the data scientist based on prior knowledge about the data.

One of the clusters will be the green cluster, and the other one - the orange cluster. And these are the seeds.

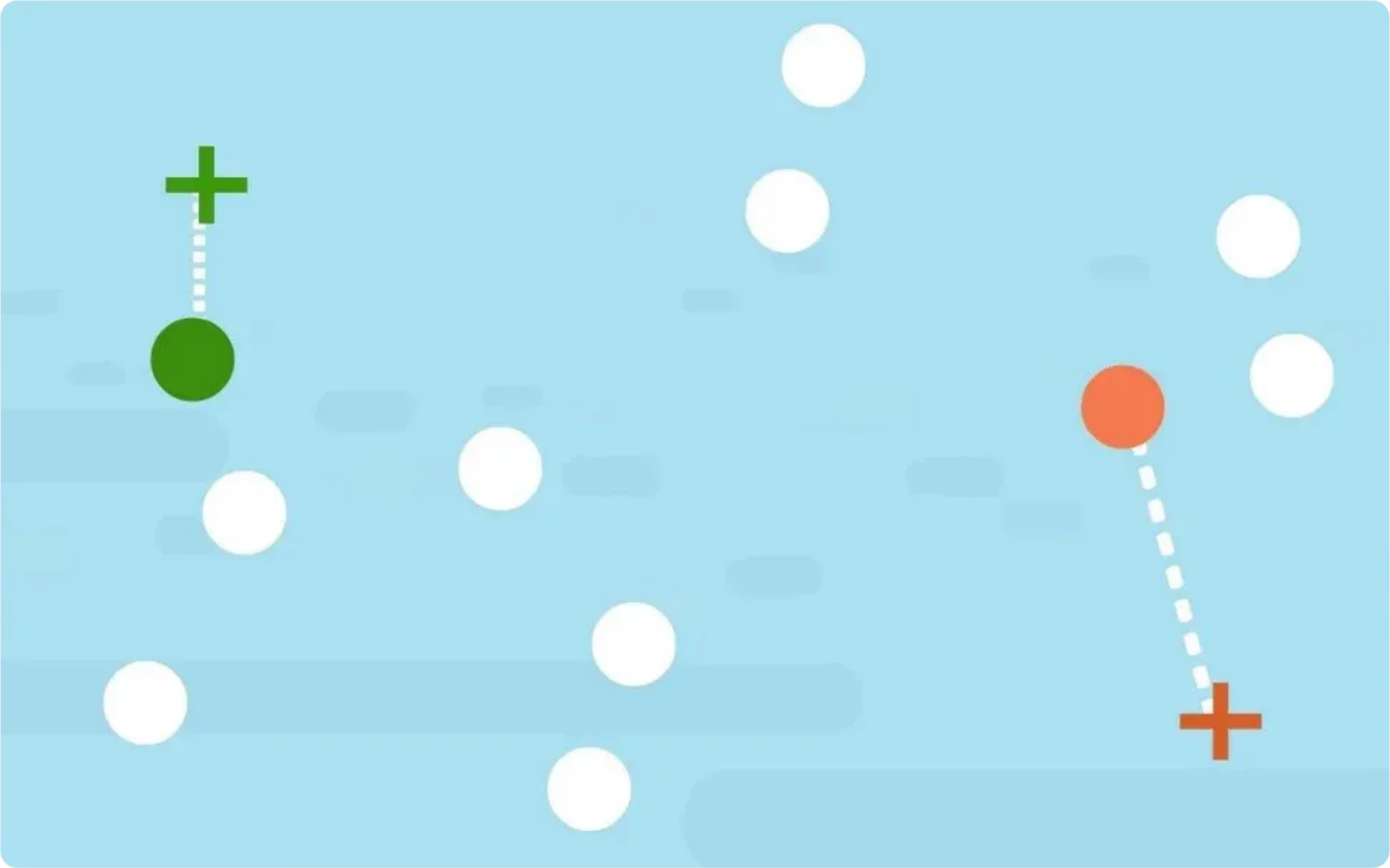


The next step is to assign each point on the graph to a seed. Which is done based on proximity.

For instance, this point is closer to the green seed than to the orange one. Therefore, it will belong to the green cluster.



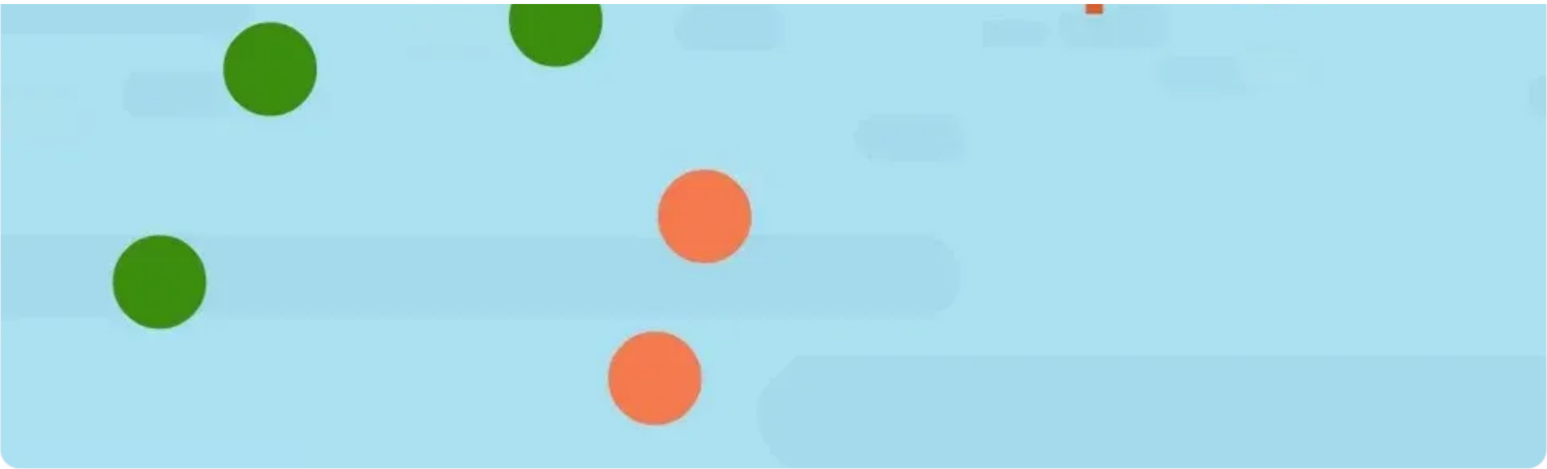
This point, on the other hand, is closer to the orange seed, therefore, it will be a part of the orange cluster.



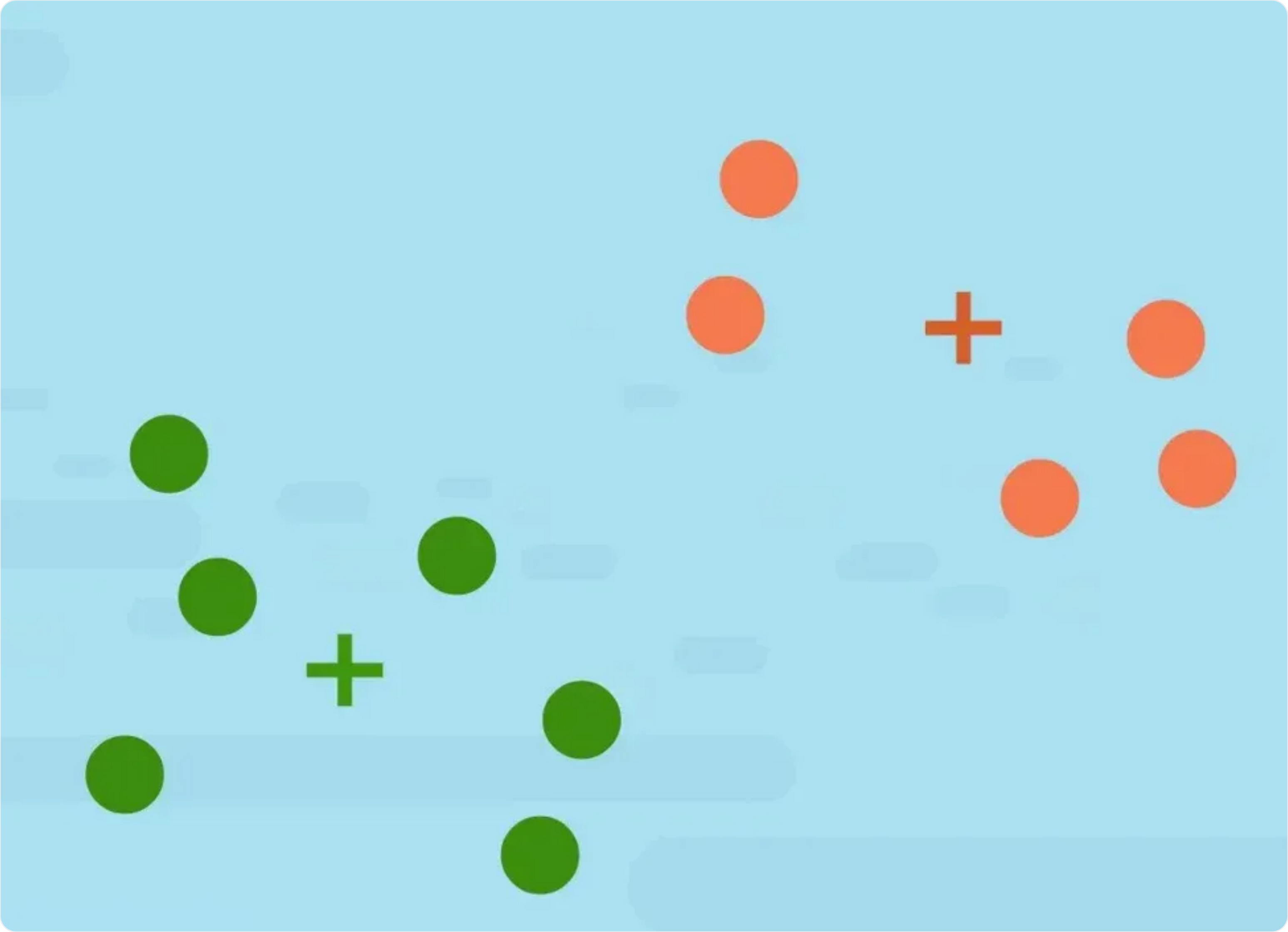
In this way, we can assign all points on the graph to a cluster, based on their Euclidean squared distance from the seeds.

The final step is to calculate the centroid or the geometrical center of the green points and the orange points.

The green seed will move closer to the green points to become their centroid and the orange will do the same for the orange points.



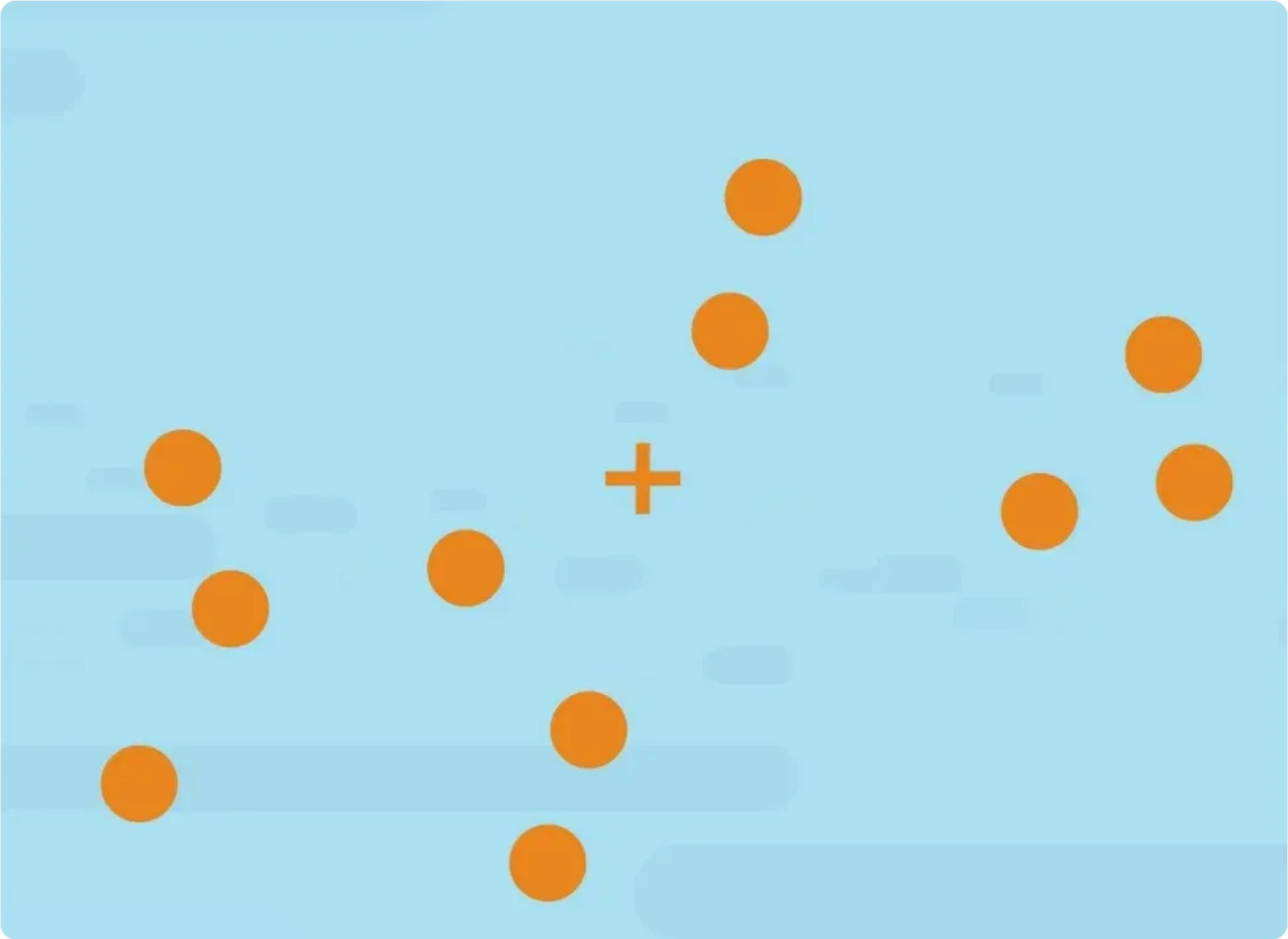
From here, we can repeat the last two steps. We can do that 10, 15 or 1000 times until we've reached a clustering solution where we can no longer adjust any of the clusters.



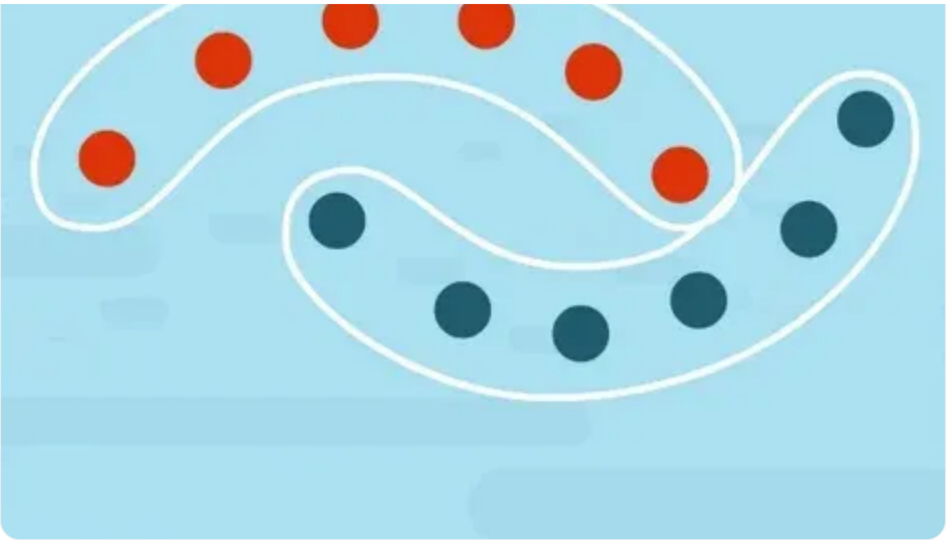
Sounds simple, right?

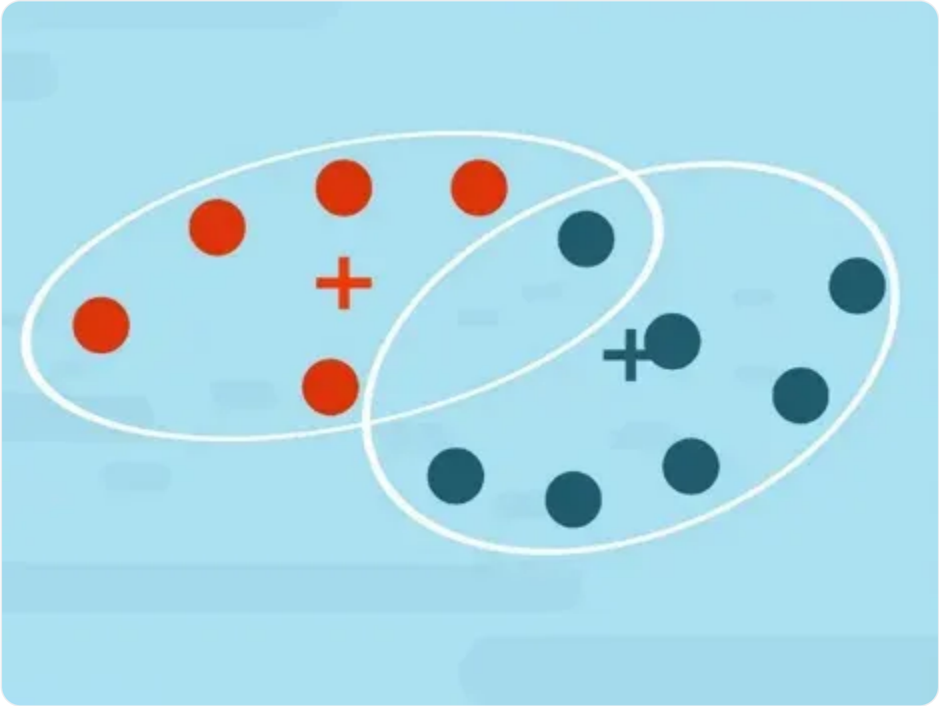
What Are the Disadvantages of K-means?

One disadvantage arises from the fact that in K-means we have to specify the number of clusters before starting. In fact, this is an issue that a lot of the clustering algorithms share. In the case of K-means if we choose K too small, the cluster centroid will not lie inside the clusters.



As we can see, in this example, this is not representative of the data. In cases where K is too large, some of the clusters may be split into two. Reasonable enough, right? Another important issue is that K-means enforces clusters with a spherical shape or blobs. The reason is that we are trying to minimize the distance from the centroids in a straight line. So, if we have clusters, which are more elongated, K-means will have difficulty separating them.





Now that you're familiar with what K-means is and how it works, let's bring theory into practice with an example.

What's K-Means Clustering's Application?

One of K-means' most important applications is dividing a data set into clusters. So, as an example, we'll see how we can implement K-means in Python. To do that, we'll use the sklearn library, which contains a number of clustering modules, including one for K-means. Let's say we have our segmentation data in a csv file. After we've read the file (in our case using the pandas method) we can proceed with implementing K-means.