DATA SCIENCE INFINITY

K-MEANS CLUSTERING

A VISUAL GUIDE

K-Means is an unsupervised learning algorithm commonly used in Data Science.

The algorithm partitions data points into distinct groups, based on their similarity (or dissimilarity) with each other.

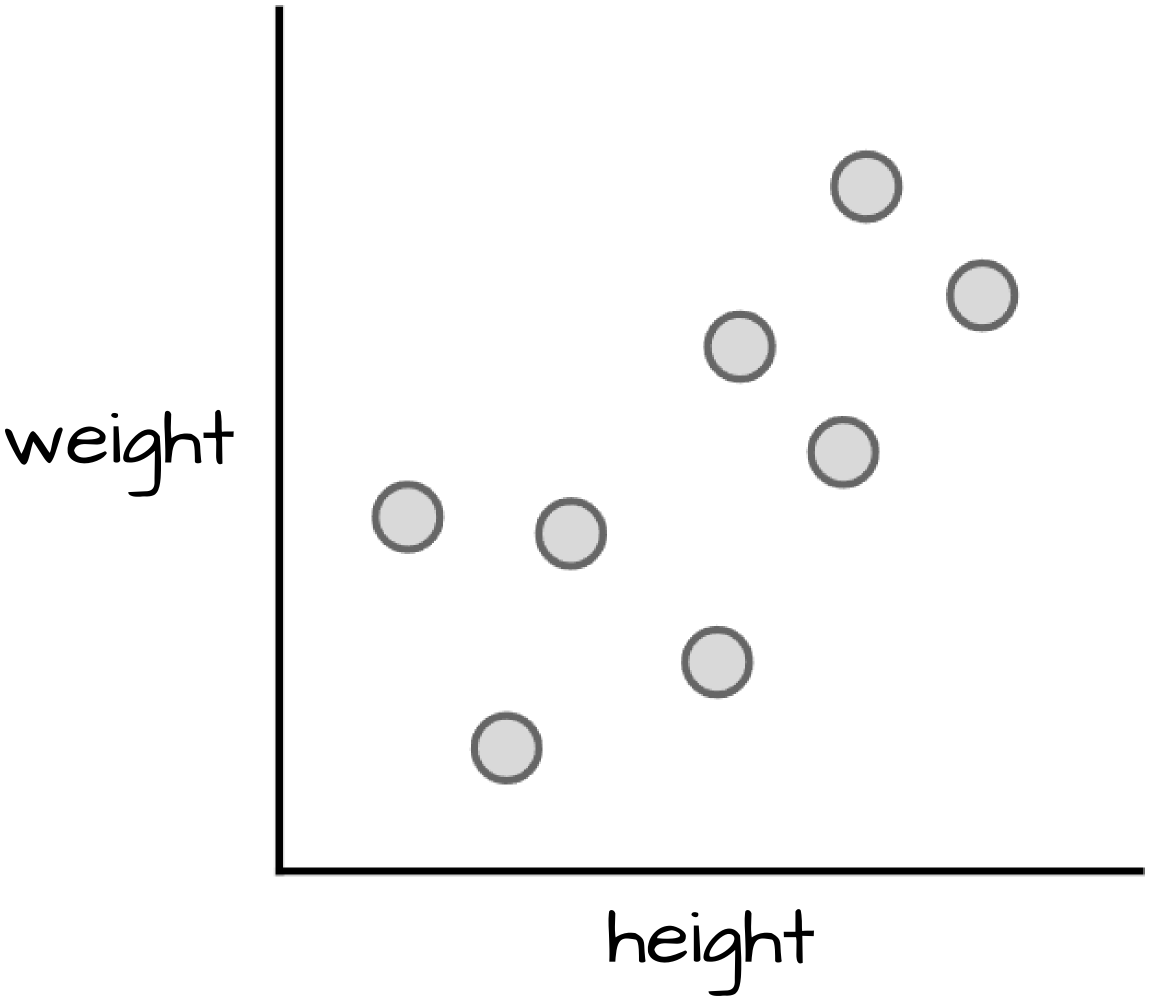
The number of distinct groups is determined by the value we set for k

So how does this algorithm work?

Let's take a look!

OUR EXAMPLE

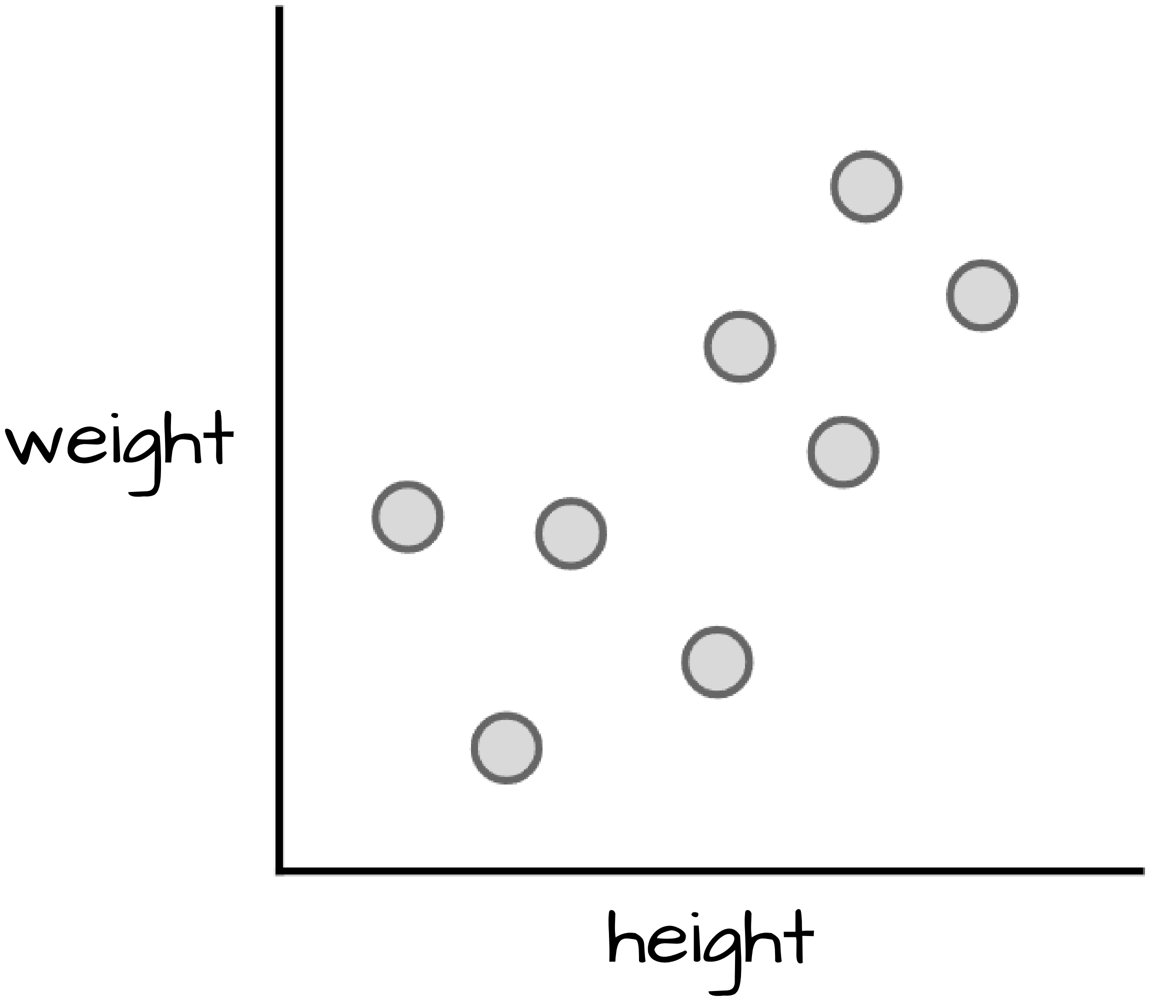
Let's say we have height and weight measurements for eight people



We want to know if there are any distinct groups that form within the data!

A VALUE FOR K

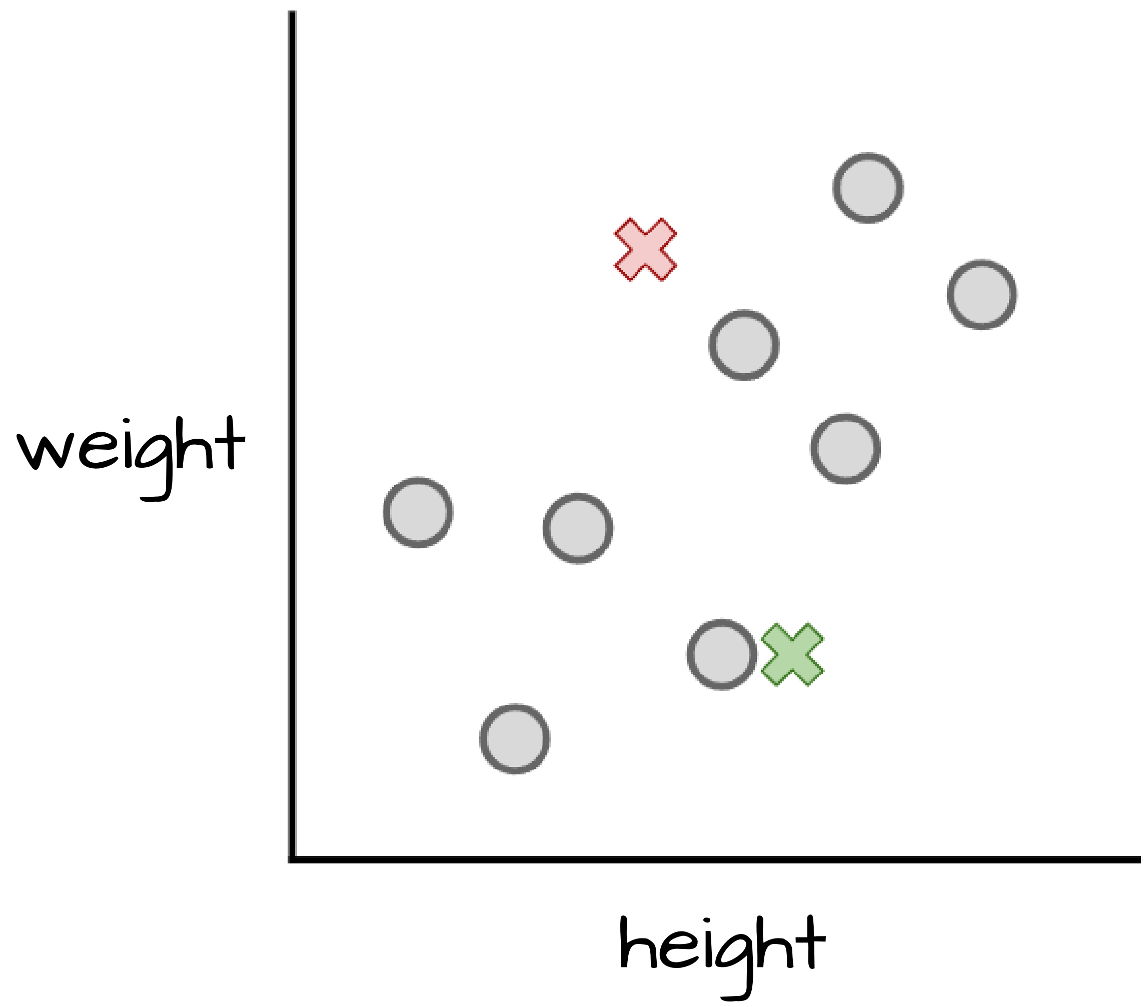
As the Data Scientist, we need to pre-specify a value for k (the number of clusters we want to end up with)



For simplicity - let's say we start with k = 2

STEP 1

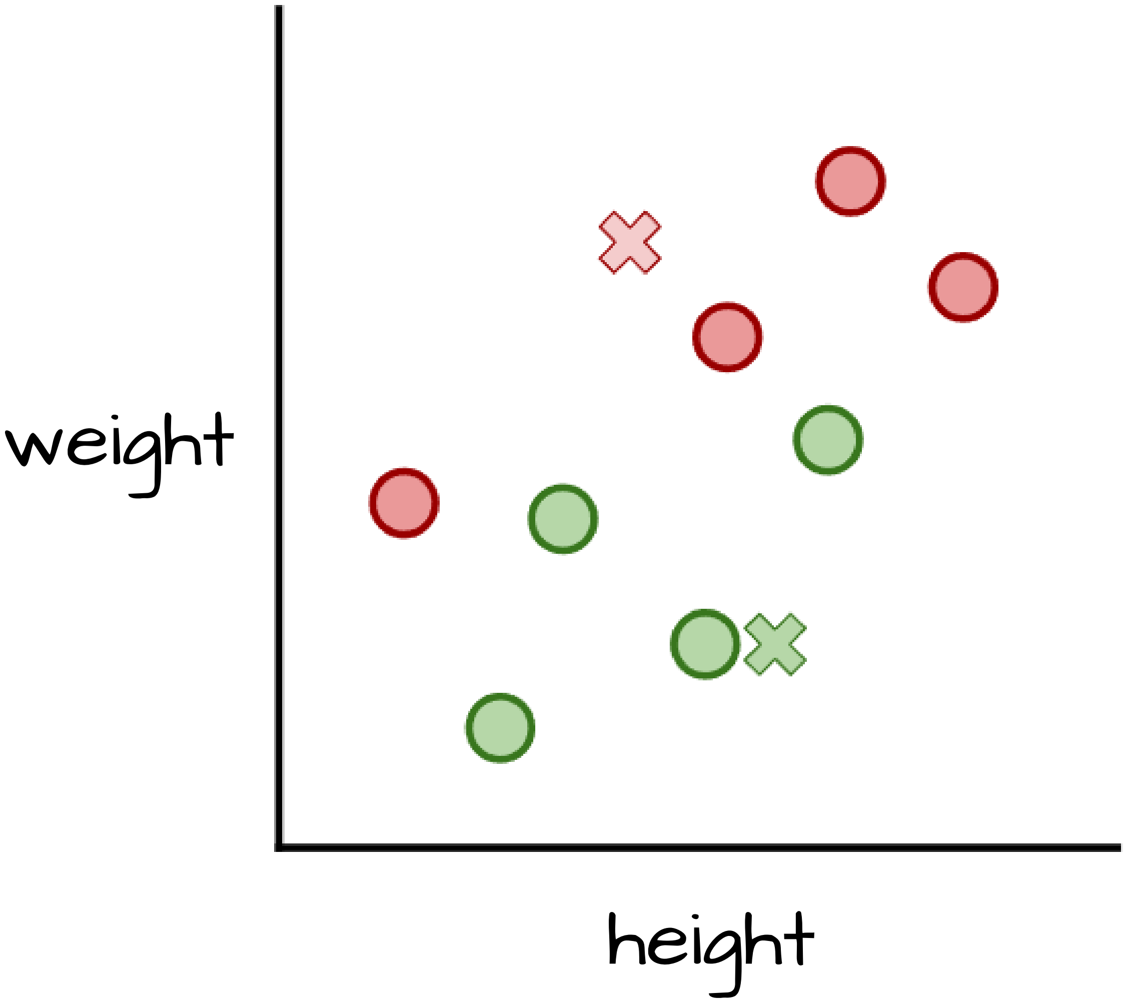
The algorithm selects k (here k = 2) random points in space. These are called centroids



In our image above we have centroid 1 shown in red, and centroid 2 shown in green

STEP 2

With the centroids in place, the algorithm assigns each data point to the nearest centroid forming our clusters

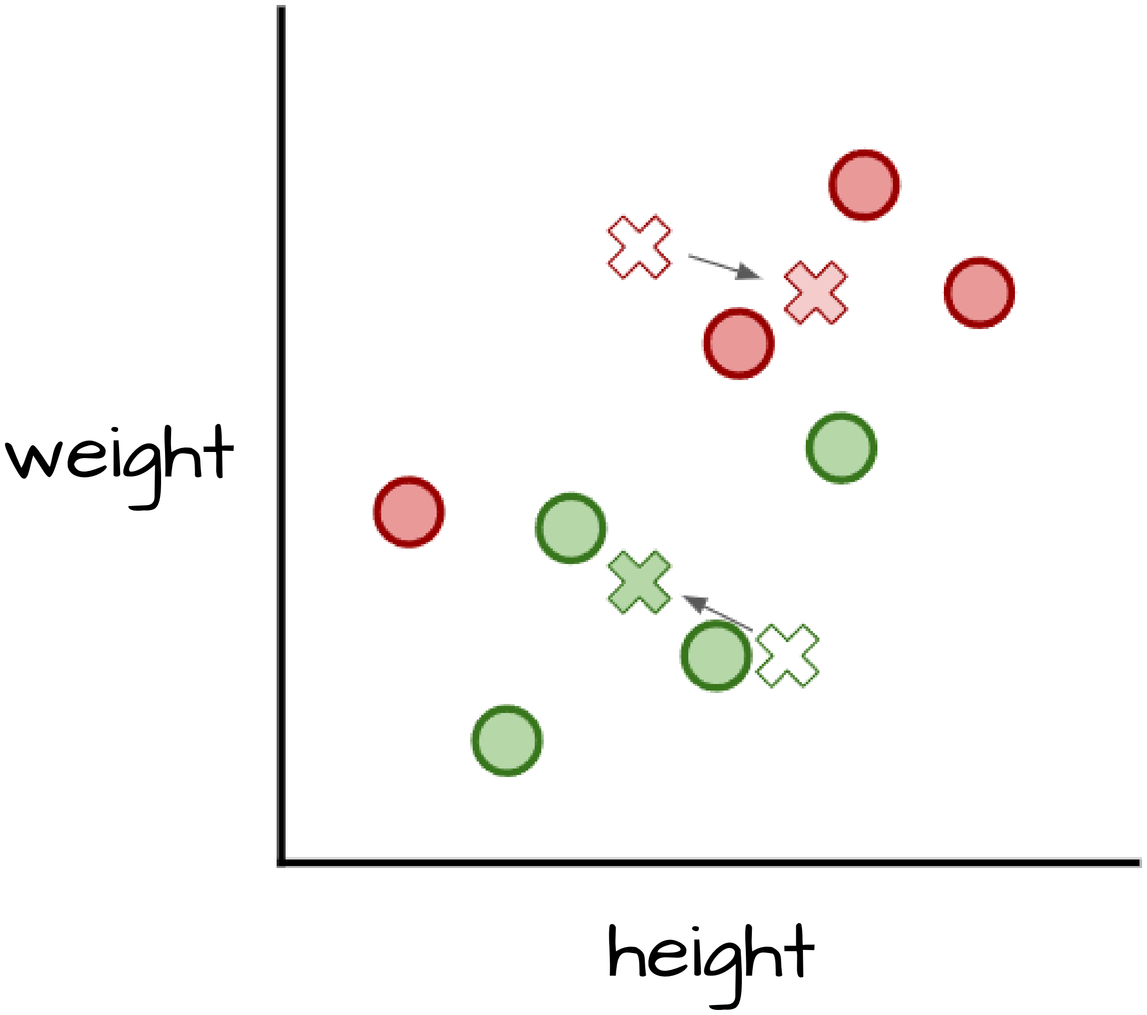


In k-means, euclidean (straight line) distance is most commonly used to assign data points to centroids

However, as our centroids have been randomly assigned at this stage, our clusters are not yet perfect!

STEP 3

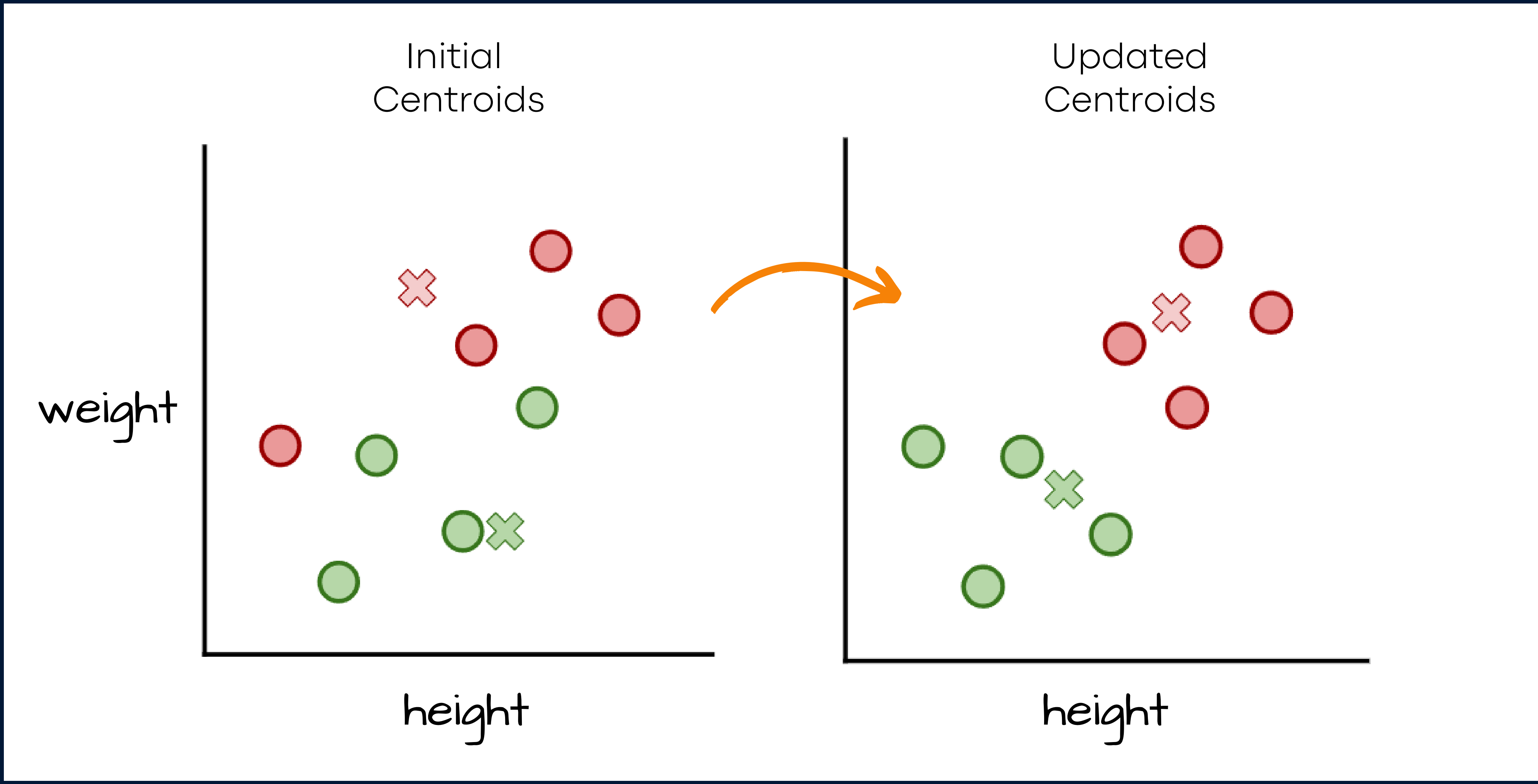
The algorithm now looks to re-position the centroids to new locations that better represent the clusters



It does this by shifting each centroid to the mean height & weight value for the data points in each cluster

STEP 4

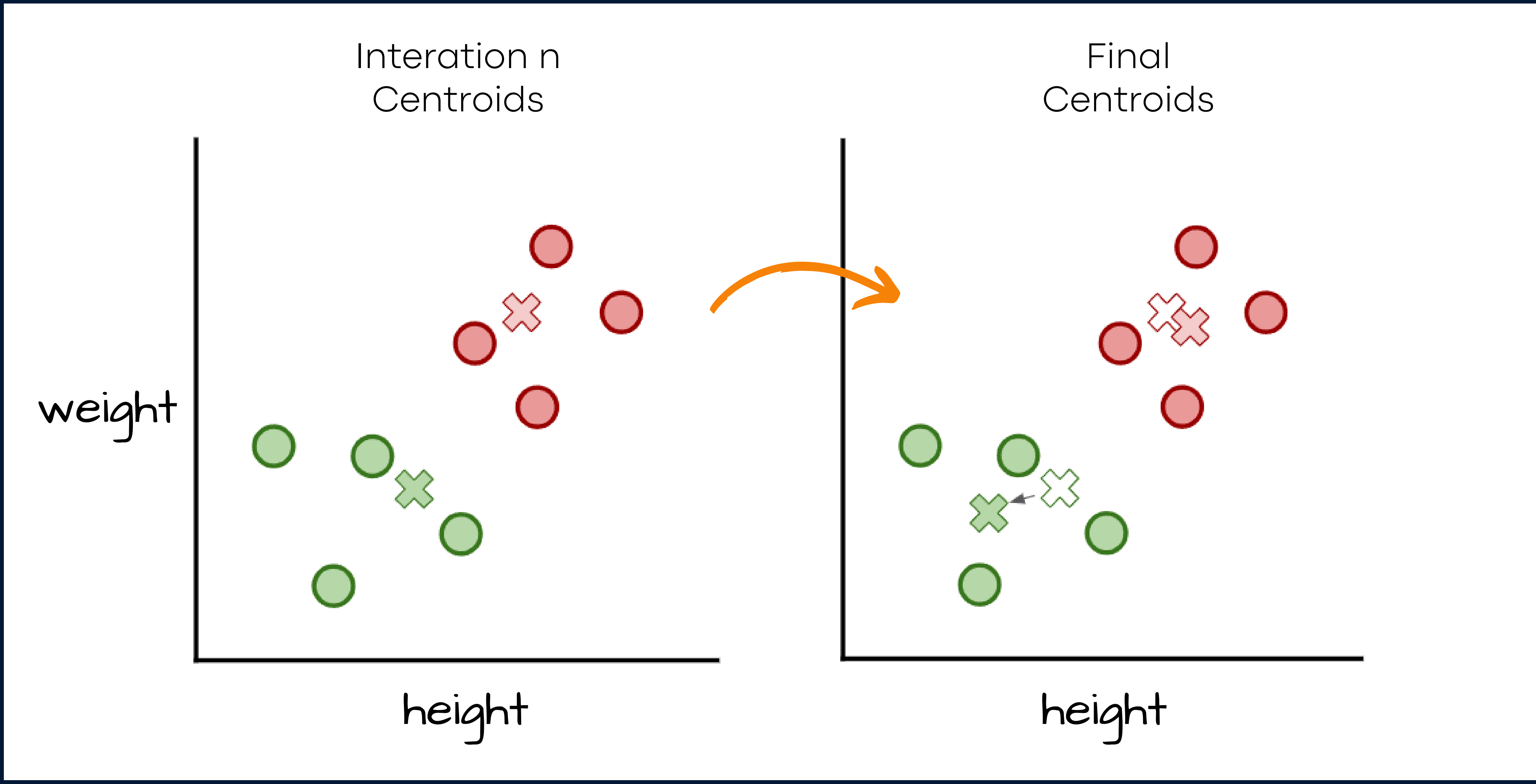
With the new centroids in place, the algorithm then re-assigns each data point to the nearest centroid



With the new centroids, some of our data points have switched allegiances to the other cluster

REPEAT...

The algorithm then repeats Step 3 (repositioning centroids and Step 4 (re-assigning data-points) until...



...no data points switch allengience, essentially meaning the clusters are stable

TO NOTE

Our example applied k-means across only 2 dimensions (height & weight) but the algorithm can be applied across any number of dimensions

It is extremely important when applying k-means (or any distance based algorithm) that your input data is scaled, and thus is comparable across each dimension

In our example we used k=2 but we could have used k=3 or any other value up to the number of data points in our set. Common ways to understand a "good" value for k are using Within Cluster Sum of Squares (WCSS) or the Silhouette Value

Want to land an incredible role in the exciting, future-proof, and lucrative field of **Data Science?**

DATA SCIENCE INFINITY

"I landed my new role at Amazon totally thanks to DSI. It's been the best academic choice of my life - giving better results than two degrees"

- Andrea

"DSI is incredible - everything is taught in such a clear and simple way, even the more complex concepts!"

- Arianna

"The best program I've been a part of, hands down"

- Christian

"I had over 40 interviews without an offer. After DSI I quickly got 7 offers including one at KPMG and my amazing new role at Deloitte!"

- Ritesh

DATA SCIENCE INFINITY

>> https://data-science-infinity.teachable.com