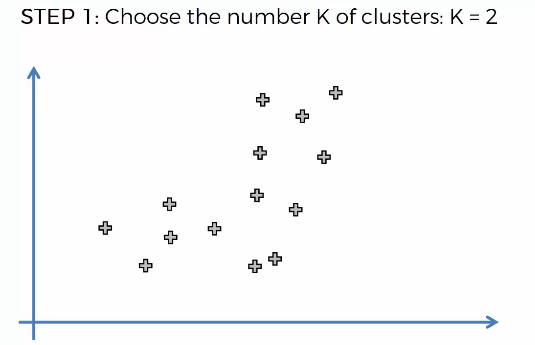
Clustering is similar to classification, but the basis is different. In Clustering, you don’t know what you are looking for, and you are trying to identify some segments or clusters in your data. When you use clustering algorithms on your dataset, unexpected things can suddenly pop up like structures, clusters and groupings you would have never thought of otherwise.

We will look at the following Machine Learning Clustering Models:

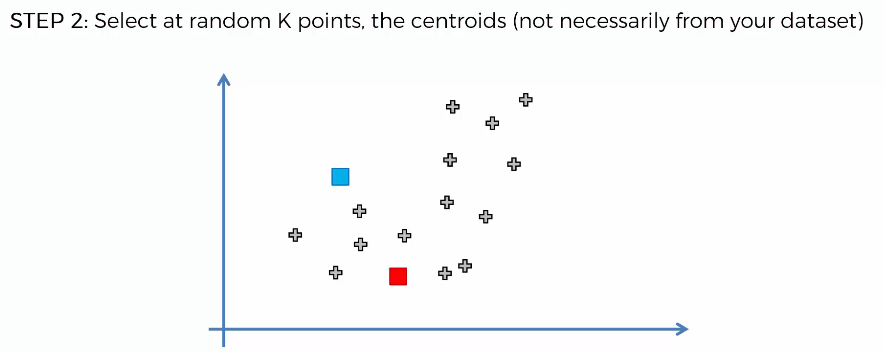
* K-Means Clustering
* Hierarchical Clustering

K-Means Clustering Intuition

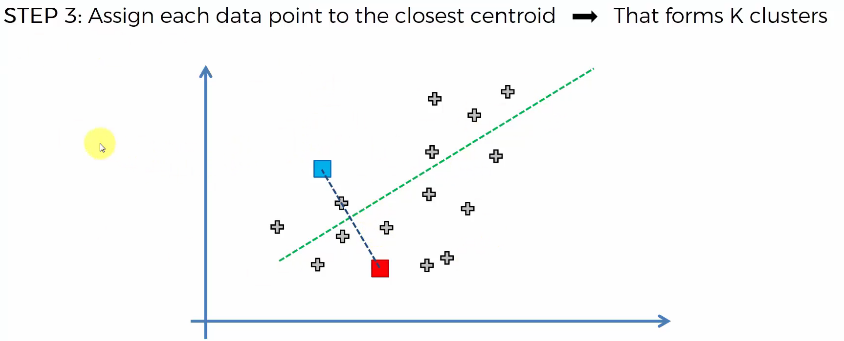
I. Choose the number K of clusters



II. Select at random K points, the centroids (not necessarily from your dataset)

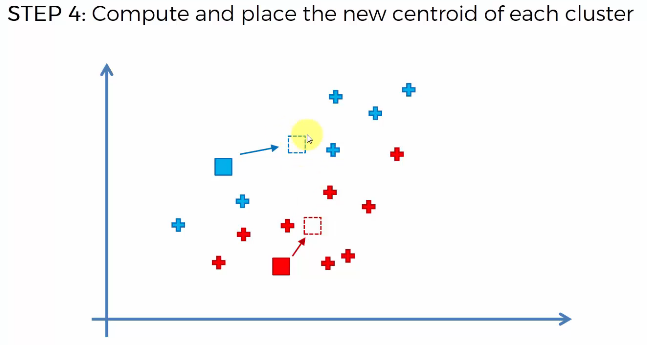


III. Assign each data point to the closest centroid. That forms K clusters



The green line above is drawn such that it passes through the center of the line connecting the two centroids. Data points falling below the green line will belong to the red cluster while the ones above the green line will belong to the blue cluster

IV. Compute and place the new centroid of each cluster

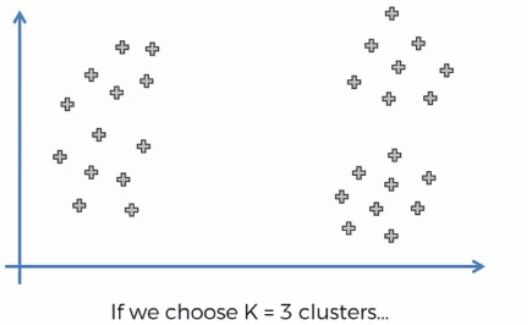


The new centroid is located at the center of gravity of all the points belonging to its cluster

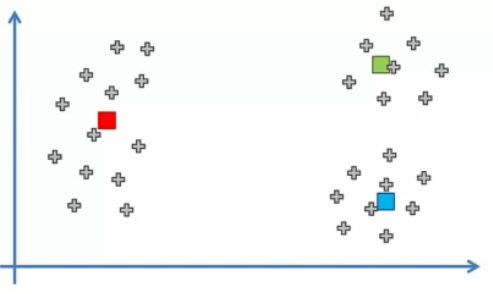
V. Reassign each data point to the new closest centroid. If any reassignment took place, go to Step 4, otherwise finish

K-Means Random Initialization Trap

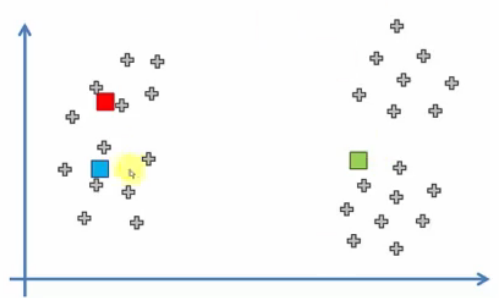
Consider the following data distribution.



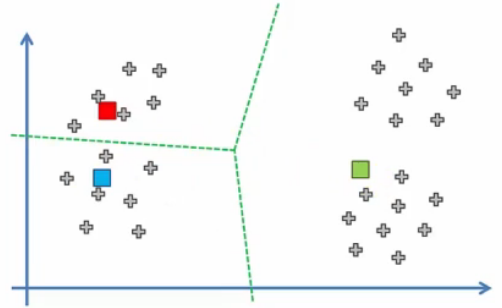
Correct random initialization would lead use to the three clusters



What would happen if we had a bad random initialization? Lets say we choose the centroids in the following manner



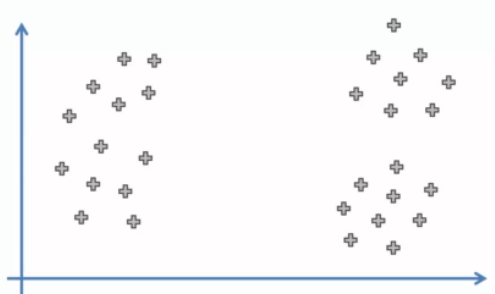
If we draw a line to divide the clusters, we get the following result



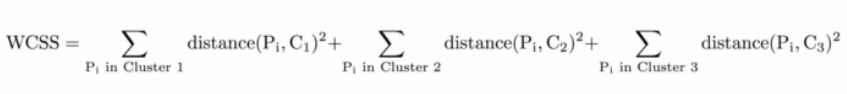
According to step 4, the centroids will move to a new location and our model is ready. But these clusters are different from what we got earlier. So, now we are trapped. To solve this, there is an additional step in the k-means clustering algorithm that allows you to select the correct centroid and the solution is the k-means++ algorithm.

K-means Selecting the Number of Clusters

Consider the following data distribution



To choose the number of clusters K, we use the metric WCSS. Its formula is given as follows:

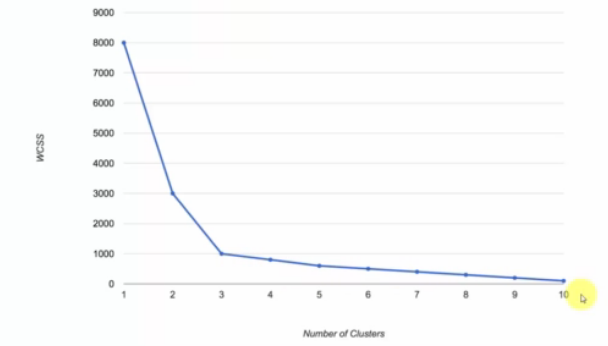


For K = 1, WCSS comes out to be very large

For K = 2, WCSS decreases to some extent

For K = 3, WCSS decreases further

WCSS decreases with increase in clusters. But how do we choose K based on WCSS? While calculating WCSS for different K, the value of WCSS for a certain K will drop heavily .And after the heavy drop, WCSS will decrease slowly. That is the hint for us to choose the K value. This is also called as the elbow method. In the diagram below, an elbow is formed at K = 3. Hence, we choose K as 3 for the given data distribution.



NOTE: When K becomes equal to the number of datapoints, the WCSS becomes 0.