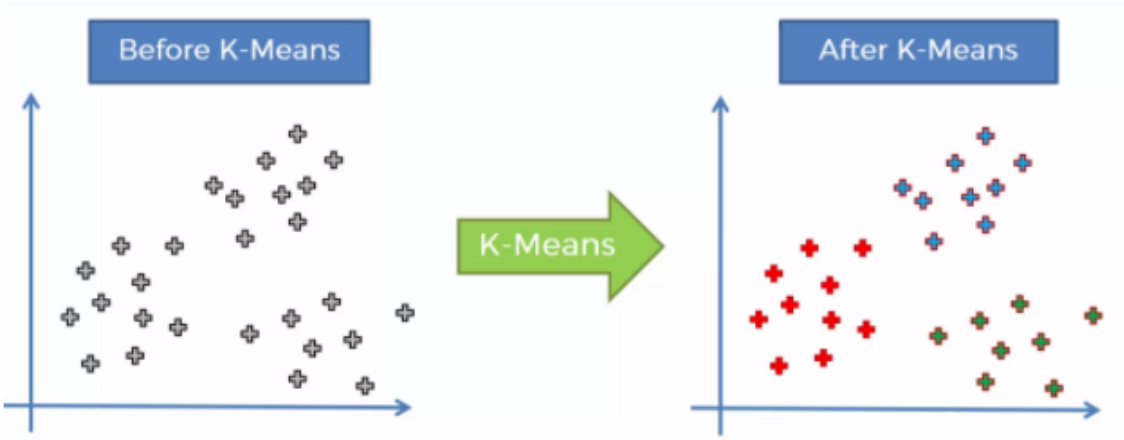
← Go Back to Making Sense of Unstructured Data

Course Content

K-means Clustering

K-means Clustering is an unsupervised learning algorithm. Like other clustering algorithms, it tries to aggregate similar objects into groups called clusters. In K-means Clustering, **K** refers to the number of clusters required. The concept of a centroid, which is the geometric center of a cluster, is used to determine the clusters that K-means finds in the dataset.

Let's understand this using an example. Suppose you go to a vegetable shop to buy some vegetables. There, you'll see different kinds of vegetables. One thing you may notice is that the vegetables will be arranged in a group of their type. The carrots and radishes will probably be kept together in one place, onion and garlic will probably be arranged in another place, potatoes will be kept together, and so on. This arrangement resembles a group or a cluster, where each vegetable is kept within its kind of group, forming the clusters.



The image on the left is *before clustering*, where all the categories or groups appear to be mixed up (same color), while the image on the right is *after clustering* where groups of similar data points seem to be clustered together and depicted with different colors.

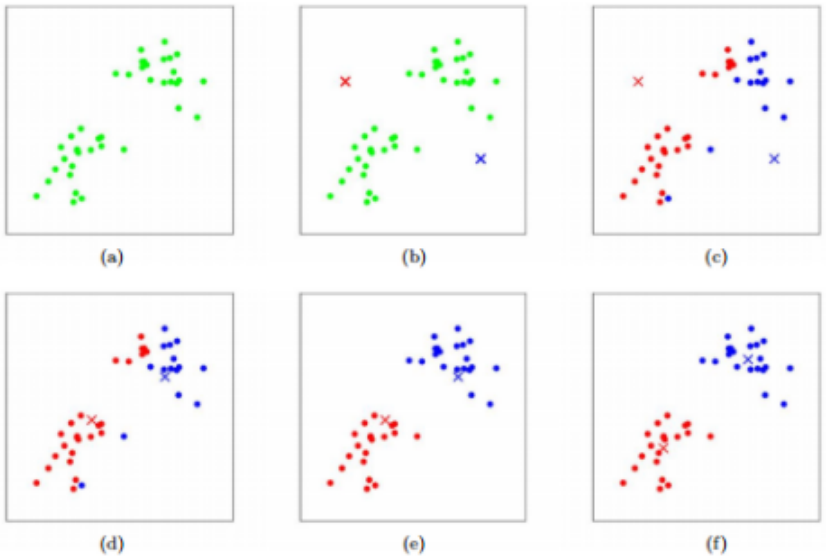
To human eyes, it can be difficult to analyze or understand a mixed-up group like the one represented by the image on the left. So we apply clustering techniques like K-means Clustering to convert this into a more visually distinct dataset with separate groups or categories of data.

Now that we have understood the basic rationale behind clustering, let's look into the working of K-Means Clustering specifically.

Working of K-Means Clustering

The steps involved in K-means Clustering are:

1. Choose the number of clusters K
2. Initialize the centroids
3. Assign each data point to the closest centroid.
4. Update the centroid by taking the mean of the cluster.
    - Repeat steps 3 and 4 until convergence i.e No discernible change in centroids is observed.



For step 3, to assign each data point to the nearest centroid, we use the distance between the centroid and the data point. This distance can be found using Euclidean distance.

The Euclidean distance d between two points (x1,y1) and (x2,y2), is defined as,

$$d = \sqrt{|x2 - x1|^2 + |y2 - y1|^2}$$

Therefore, K-means clustering uses the Euclidean distance to allocate each of the data points to its nearest cluster, so that the sum of squares within each cluster is minimum.

For step 4 we update the centroid using the mean value of all the points in the cluster, hence the name *K-means clustering*.

Example:

Let's understand this with an example:

N={2,3,4,5,10,11,13,15} - Performing K-means on these points, all points on the x-axis .

Step 1: Choose K, K=2

Step 2: Initialize the centroids randomly

- c1=3

- c2=12

Note : As y=0 on the x-axis , d=|x2-x1|

Step 3.1: For each point calculate distance and assign it to the closest centroid

Data Point	Distance from C1=3	Distance from C2=12	Cluster Assigned
2	1	10	c1
3	0	9	c1
4	1	8	c1
5	2	7	c1
10	7	2	c2
11	8	1	c2
13	10	1	c2
15	12	3	c2

Cluster c1 contains {2,3,4,5}

Cluster c2 contains {10,11,13,15}

Step 3.2: Update the centroids

- c1=(2+3+4+5) / 4 = 3.5
- c2=(10+11+13+15) / 4=12.25

Step 3 should repeat until there is no change in the centroids.

Things to consider :

- It is always better to standardize/normalize the data points with any distance-based algorithm like K-means clustering, because different variables may have different scales, and that may affect the sum of squared error. Hence, it is a good practice to bring all the data points under one scale.
- We initialize centroids randomly, so different initializations may lead to different clusters. There is a

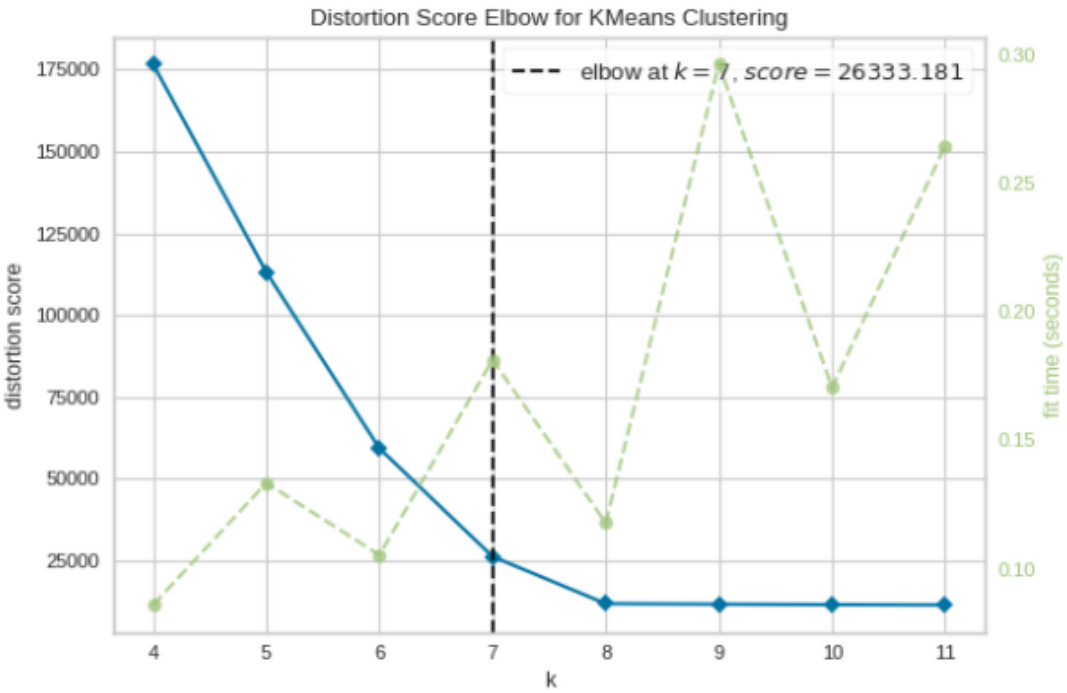
possibility that the algorithm falls into a local optimum rather than the global optimum. So it's better to iterate with different initializations and select the one with the least sum of squared distances within clusters.

Evaluation Methods:

For K-means clustering we need a fixed value of K, and that should be known before performing the method. It is not a learned parameter, and we will have to figure it out. To find out the optimal K number, one method we can use is called the ELBOW method.

ELBOW Method:

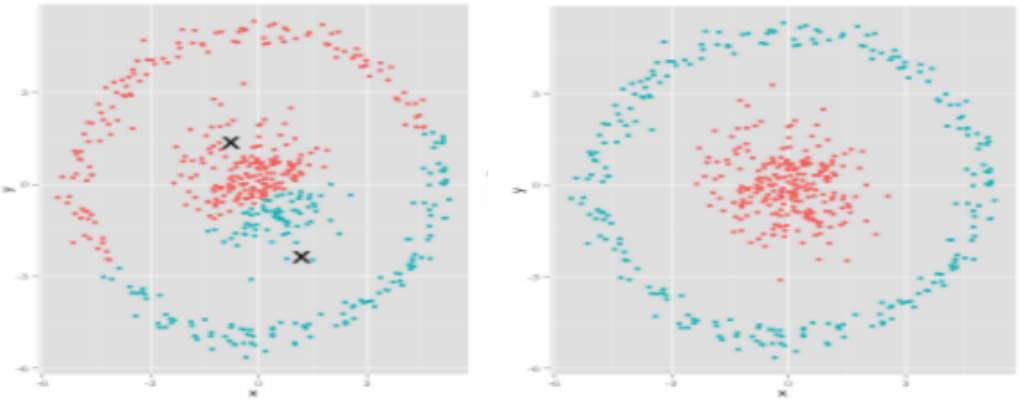
In this method, we take different values of K i.e from 1 to the range required. For each value of K, we calculate the sum of squares within clusters, usually called WCSS (Within Cluster Sum of Squares). So we calculate the sum of the squared distance between the centroids and the data points for each cluster and do a summation of all the clusters. Then, we plot WCSS vs K - the plot shape looks similar to that of an elbow. As we see in the graph, as K increases, the WCSS will decrease. The optimal point is at the elbow - after that as K increases, the fall in WCSS is not that significant. In the given example, K=7 appears to be the optimal value of K.



Source

Assumptions:

1. K-means Clustering is limited to linear cluster boundaries: The fundamental model assumptions of K-means Clustering (points will be closer to their cluster center than to others), means that the algorithm will often be ineffective if the clusters have complicated geometries



2. All clusters are of the same size.
3. Clusters have the same extent in every direction. This assumption would not be true in a dataset where different measurements are in different units.
4. Clusters have similar numbers of points assigned to them.

Image Source

Applications:

1. Document Clustering - Group similar documents together
2. Customer Segmentation - Divide customers into similar groups
3. Image Segmentation - Grouping similar pixels together

Previous

Next