K Mean Algorithm For Clustering

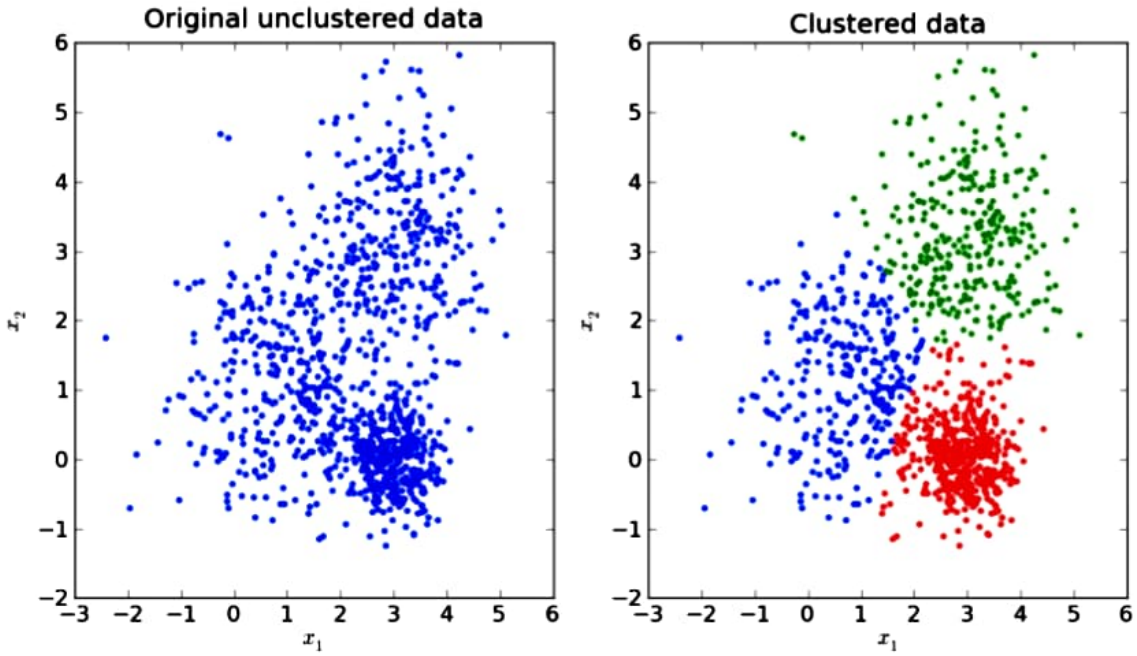
Clustering is a type of Unsupervised learning.
This is very often used when you don't have labeled data.
K-Means Clustering is one of the popular clustering algorithm.
The goal of this algorithm is to find groups(clusters) in the given data.

K Means Clustering is one of the most popular Machine Learning algorithms for cluster analysis in data mining. K-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster.

K Means algorithm is an unsupervised learning algorithm, ie. it needs no training data, it performs the computation on the actual dataset. This should be apparent from the fact that in K Means, we are just trying to group similar data points into clusters, there is no prediction involved.

The K Means algorithm is easy to understand and to implement. It works well in a large number of cases and is a powerful tool to have in the closet.
The algorithm has a loose relationship to the k-nearest neighbor classifier, a popular machine learning technique for classification.

K-Means is a very simple algorithm which clusters the data into K number of clusters. The following image from PyPR is an example of K-Means Clustering.



The K Means algorithm is iterative based, it repeatedly calculates the cluster centroids, refining the values until they do not change much.

The k-means algorithm takes a dataset of 'n' points as input, together with an integer parameter 'k' specifying how many clusters to create(supplied by the programmer). The output is a set of 'k' cluster centroids and a labeling of the dataset that maps each of the data points to a unique cluster.

In the beginning, the algorithm chooses k centroids in the dataset. Then it calculates the distance of each point to each centroid. Each centroid represents a cluster and the points closest to the centroid are assigned to the cluster. At the end of the first iteration, the centroid values are recalculated, usually taking the arithmetic mean of all points in the cluster.

After the new values of centroid are found, the algorithm performs the same set of steps over and over again until the differences between old centroids and the new centroids are negligible.

K Mean Algorithm

Our algorithm works as follows, assuming we have inputs x1,x2,x3,...,xn and value of K

Step 1 -
Pick K random points as cluster centers called centroids.

Step 2 -
Assign each x_i to nearest cluster by calculating its distance to each centroid.

Step 3 -
Find new cluster center by taking the average of the assigned points.

Step 4 -
Repeat Step 2 and 3 until none of the cluster assignments change.

Detailed Explanation :

Step 1
We randomly pick K cluster centers(centroids). Let's assume these are
c_1, c_2, ..., c_k, and we can say that;
C = c_1, c_2,..., c_k
C is the set of all centroids.

Step 2
In this step we assign each input value to closest center. This is done by calculating Euclidean(L2) distance between the point and the each centroid.

$$\arg\min_{c_i \in C} dist(c_i, x)^2$$

Where dist(.) is the Euclidean distance.

Step 3
In this step, we find the new centroid by taking the average of all the points assigned to that cluster.

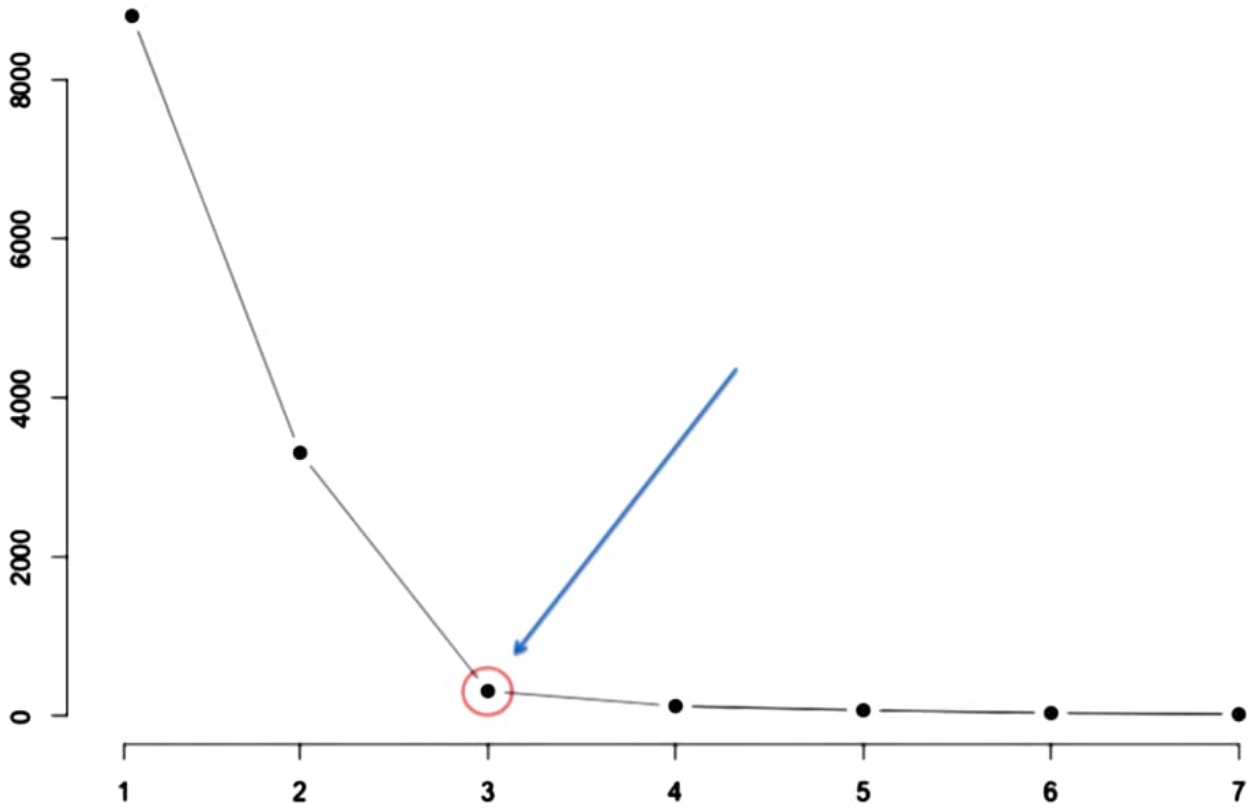
$$c_i = \frac{1}{|S_i|} \sum_{x_i \in S_i} x_i$$

Si is the set of all points assigned to the ith cluster.

Step 4
In this step, we repeat step 2 and 3 until none of the cluster assignments change. That means until our clusters remain stable, we repeat the algorithm.

Choosing the Value of K

We often know the value of K. In that case we use the value of K. Else we use the Elbow Method.



We run the algorithm for different values of K(say K = 10 to 1) and plot the K values against SSE(Sum of Squared Errors). And select the value of K for the elbow point as shown in the figure.