What is Clustering?

Cluster analysis is a technique used in data mining and machine learning to group similar objects into clusters. K-means clustering is a widely used method for cluster analysis where the aim is to partition a set of objects into K clusters in such a way that the sum of the squared distances between the objects and their assigned cluster mean is minimized.

## What Is K-Means Clustering?

K-means clustering is a method for grouping n observations into K clusters. It uses vector quantization and aims to assign each observation to the cluster with the nearest mean or centroid, which serves as a prototype for the cluster. Originally developed for signal processing, [K-means clustering](https://www.analyticsvidhya.com/blog/2021/11/understanding-k-means-clustering-in-machine-learningwith-examples/) is now widely used in machine learning to partition data points into K clusters based on their similarity. The goal is to minimize the sum of squared distances between the data points and their corresponding cluster centroids, resulting in clusters that are internally homogeneous and distinct from each other.

Recall the first property of clusters – it states that the points within a cluster should be similar to each other. So,**our aim here is to minimize the distance between the points within a cluster.**

There is an algorithm that tries to minimize the distance of the points in a cluster with their centroid – the k-means clustering technique.

K-means is a centroid-based algorithm or a distance-based algorithm, where we calculate the distances to assign a point to a cluster. In K-Means, each cluster is associated with a centroid.

**The main objective of the K-Means algorithm is to minimize the sum of distances between the points and their respective cluster centroid.**

Optimization plays a crucial role in the [k-means clustering algorithm](https://www.analyticsvidhya.com/blog/2021/11/understanding-k-means-clustering-in-machine-learningwith-examples/). The goal of the optimization process is to find the best set of centroids that minimizes the sum of squared distances between each data point and its closest centroid. This process is repeated multiple times until convergence, resulting in the optimal clustering solution.

## How to Apply K-Means Clustering Algorithm?

Let’s now take an example to understand how K-Means actually [works](https://www.analyticsvidhya.com/blog/2021/11/understanding-k-means-clustering-in-machine-learningwith-examples/):

[](https://cdn.analyticsvidhya.com/wp-content/uploads/2019/08/Screenshot-from-2019-08-09-12-21-43.png)

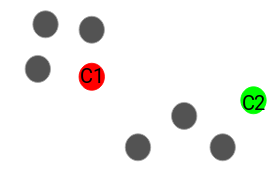
Time needed: 10 minutes

We have these 8 points, and we want to apply k-means to create clusters for these points. Here’s how we can do it.

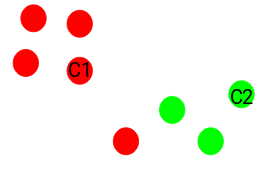
1. **Choose the number of clusters k**

The first step in k-means is to pick the number of clusters, k.

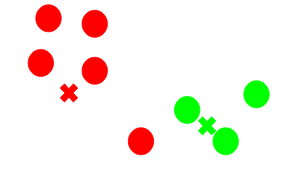
1. **Select k random points from the data as centroids**

Next, we randomly select the centroid for each cluster. Let’s say we want to have 2 clusters, so k is equal to 2 here. We then randomly select the centroid:  
  
Here, the red and green circles represent the centroid for these clusters.

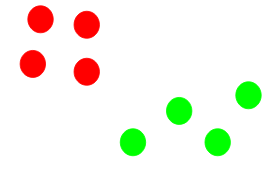
1. **Assign all the points to the closest cluster centroid**

Once we have initialized the centroids, we assign each point to the closest cluster centroid:  
Here you can see that the points closer to the red point are assigned to the red cluster, whereas the points closer to the green point are assigned to the green cluster.

1. **Recompute the centroids of newly formed clusters**

Now, once we have assigned all of the points to either cluster, the next step is to compute the centroids of newly formed clusters:  
  
Here, the red and green crosses are the new centroids.

1. **Repeat steps 3 and 4**

We then repeat steps 3 and 4:  
  
The step of computing the centroid and assigning all the points to the cluster based on their distance from the centroid is a single iteration. But wait – when should we stop this process? It can’t run till eternity, right?

#### **Stopping Criteria for K-Means Clustering**

There are essentially three stopping criteria that can be adopted to stop the K-means algorithm:

1. Centroids of newly formed clusters do not change
2. Points remain in the same cluster
3. Maximum number of iterations is reached

We can stop the algorithm if the centroids of newly formed clusters are not changing. Even after multiple iterations, if we are getting the same centroids for all the clusters, we can say that the algorithm is not learning any new pattern, and it is a sign to stop the training.

Another clear sign that we should stop the training process is if the points remain in the same cluster even after training the algorithm for multiple iterations.

Finally, we can stop the training if the maximum number of iterations is reached. Suppose we have set the number of iterations as 100. The process will repeat for 100 iterations before stopping.

Practical 8 **Perform the data clustering using clustering algorithm.**

k-means clustering using R

#apply K means to iris and store result

**newiris <- iris**

**newiris$Species <- NULL**

**(kc <- kmeans(newiris,3))**

#Compare the Species label with the clustering result

table(iris$Species,kc$cluster)

#Plot the clusters and their centers

plot(newiris[c("Sepal.Length","Sepal.Width")],col=kc$cluster)

**points(kc$centers[,c("Sepal.Length","Sepal.Width")],col=1:3,pch=8,cex=2)**

dev.off()

**#Plot the clusters and their centre**

plot(newiris[c("Sepal.Length","Sepal.Width")],col=kc$cluster)