**Steps for K-Means clustering algorithm:**

1. Choose the number k for k clusters
2. Select at random the k points, the centroids (not necessarily from the dataset)
3. Assign each data point to the closest centroid, that will form k clusters.
4. Computer and place the new centroid to each cluster.
5. Reassign each data point to new closest centroid, if any reassignment took place than go to the step 4 again otherwise if no new data point is assigned to any of the cluster, then finish the algorithm.

**Random initialization trap:**

Consider the centroids are selected randomly in the scatter plot in such a way that can change each time with data points.

With that it is also possible that we select the k centroids in such a way that neither change the points with it nor give the optimal categories.

This means that the selection of the random initialization of the clusters at the start can effect the overall result of the algorithm, if we choose the bad random initialization of the centroid (it will give us wrong clusters in the data points), so how would we battle that? For that we have random initialization trap, but this answer is not straight forward because there is modification to the k-means which allows to correctly select and the solution here is k-means plus plus….

**Choosing the right number of clusters:**

This topic is related to the algorithm to find the optimal number of clusters.

In order to understand how many clusters are better for the given problem, we need certain matrix and certain way for choosing that better number of clusters.

This metrics is called “With in cluster sum of square or WCSS”.

The wcss is the metrics that measures the sum of square of distances between each data point and centroid with in each cluster. If this metrics compute the minimum result, the k-means will be fit better.

Now consider for choosing the best number of clusters we start from the number of clusters = 1, and computer the wcss for each of the data points.

Then we choose the number of clusters = 2 and compute the wcss and we will be able to see the difference between the wcss values, it will reduce by increasing the number of clusters and we also know that the model having less value for wcss will best fil on the data set.

But problem here is that we can increase the number of clusters upto the number of data points we have, we can create each data point a single cluster in that case every cluster’s wcss would be 0 and there summation would also be zero, in that case is this metrics will best fit our model ? the answer is no.

What we can do is to plot a graph between all the possible number of clusters we could have and the wcss and then we have to choose manually for the optimal number of clusters we could have.

This method is called elbow method.

In this method we look for the elbow where the large amount of jump stops.

