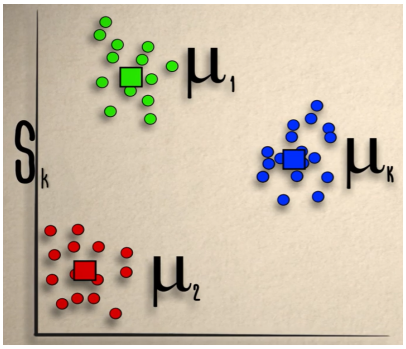
The K-Means Algorithm

We have seen the K-means problem as a particular subset of general clustering problems. We have assumed that each data point is a vector of continuous values and Square Euclidean Distance is used as the dissimilarity measure between two data points.

In the last example we have seen, the output of the K-means Clustering problem should be a set of cluster centers which are μ_1, μ_2, ... μ_k and a set of assignment of data points to clusters. For example, S_k is the set of data points assigned to the cluster μ_k.



The K-means clustering problem tells us exactly how we should choose the cluster centers and how to assign data points to clusters.

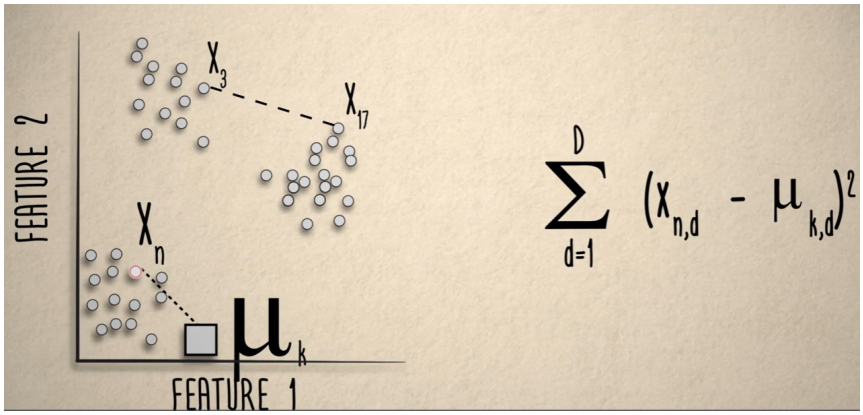
As we want to group the data according to similarity, the idea here is to **minimize dissimilarity** within each cluster.

We know how to calculate the dissimilarity between two data points.

Now we need to calculate the Global Dissimilarity.

Global Dissimilarity :

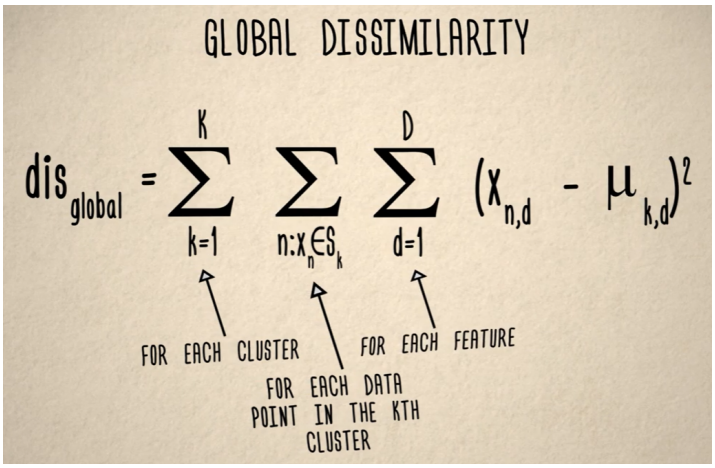
The dissimilarity between any data point and a cluster center is the distance from the



data point to the cluster center. We have used Squared Euclidean Distance here.

So, the dissimilarity within a cluster is the sum over the distances between data points in that cluster and the cluster center.

Finally the global dissimilarity is the sum over the cluster dissimilarity.
So, we calculate the Global Dissimilarity by iterating over each cluster, over each data point within the cluster and over each feature of the data point.

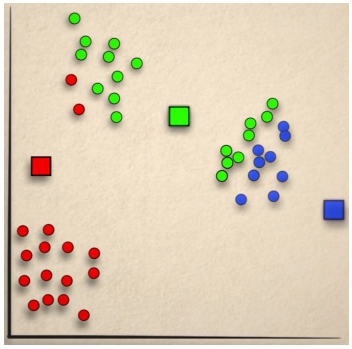


So, the main target of the K-Means Clustering Algorithm is to minimize the global dissimilarity which is also called the **K-Means Objective Function.**
We minimize the objective function by choosing a set of K-clusters and by properly assigning data points to clusters.

We are still assuming that we know the value of K in advance but we generally don't.
But the K-Means clustering algorithm or LLOYD's algorithm seems to perform very well in such a situation, thus solving this problem in a way.

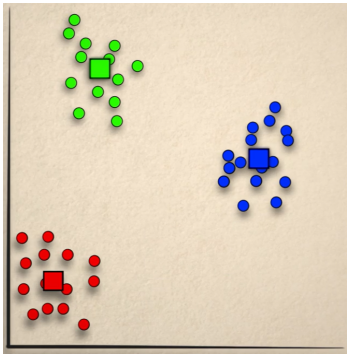
The K-Means Clustering Algorithm:

We don't know the cluster centers in advance. So, we start by initializing the cluster centers to some values. Then we alternate between the below 2 steps:



1. We assign each data point on the cluster with the closest cluster center.
2. We update the cluster centers to be the mean of all the data points in the cluster.

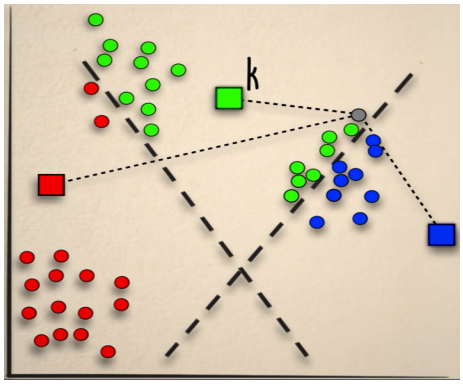
We iterate through these two steps until we can't make any more changes.
And finally, the cluster center's position looks something like this.



One important point here is:
How do we initialize the cluster centers at first?

- One option here is to draw the cluster centers uniformly at random from the existing data points.
- Some more sophisticated initialization methods are available, which is followed in K-means ++.

But, no matter how we choose to initialize the cluster centers, once we have some values for the centers, we can start iterating the main steps of the algorithm.



So, first we assign each data point to the cluster with the closest center. For doing this for each data point, at first we compute the distance to all K cluster centers. The cluster center for which the distance between the data point and cluster center is minimum is the cluster center to which that data point gets assigned.

The distance calculator can be done completely separate for every data point. So, it's extremely easy to divide up the data points across cores or processors and perform these calculations separately to speed up the running time. This is called **Embarrassingly Parallel Computation.**
So, after the assignment of data points to clusters, we recalculate the cluster centers for every cluster by taking the mean of all the data points in that particular cluster.
That is, for each feature, we sum up the values of the data points in the feature and divide it by the total number of data points in the cluster.
One significant thing here is that, luckily the K-means algorithm always stops in finite time.
In summary in this article, we get familiar with what the K-Means clustering problem is and how to develop the algorithm to solve the problem.