This code is to assign very large dataset(two dimensional points) to clustering based on the training data(initial ten points). k-means clustering is commonly used in Vector quantization, Cluster analysis and Feature learning. It is popular for cluster analysis in data mining.

k-means clustering is a method of vector quantization, aiming 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. This results in a partitioning of the data space into Voronoi cells.

The following are the steps of k-means algorithm

* **Step 1:** Select K points at random (Centers)
* **Step 2**: For each data point, assign it to the closest center
  + Now we formed K clusters
* **Step 3:** For each cluster, re-compute the centers
  + E.g., in the case of 2D points 🡺
    - X: average over all x-axis points in the cluster
    - Y: average over all y-axis points in the cluster
* **Step 4:** If the new centers are different from the old centers (previous iteration) 🡺 Go to Step 2

After several iterations, the large dataset would be assigned to right clustering.

The K\_means.java is under Hadoop Map-Reduce framework to deal with big input data.

On the map side:

* + Each map reads the K-centroids + one block from dataset
  + Assign each point to the closest centroid
  + Output <centroid, point>

On the reduce side:

* + Gets all points for a given centroid
  + Re-compute a new centroid for this cluster
  + Output: <new centroid>

And there is iteration control in main, that if the center change slightly or the number of iterations exceed the threshold, the loop would stop.

The Utils.java contains several methods used in K\_means.java.

Method “getCenters” gather the input data and put it into 2-D ArrayList, method “isFinished” determines if the centers are converge.