

kmeans.java

```
1  import java.text.DecimalFormat;
2  import java.util.Arrays;
3  import java.util.Random;
4
5  public class kmeans {
6      int x[], y[];           // data points
7      int num;                // number of data points (supplied by the user)
8      int k;                  // number of clusters (supplied by the user)
9      double meanX[],meanY[]; // cluster centres
10     double oldX[],oldY[];   // backup old cluster centres
11     int cAssign[];          // cluster assignment
12
13     private kmeans(int num,int clusters){
14         this.num = num;
15         x= new int[num];
16         y= new int[num];
17         k=clusters;
18         meanX = new double[k];
19         meanY = new double[k];
20         oldX = new double[k];
21         oldY = new double[k];
22         cAssign = new int[num];
23
24     }
25
26     void randomMean() {
27         //Initialize meanX and meanY with random values between 0 and 500 for all k centres
28         //using the nextInt() method in the java.util.Random class
29         for (int i = 0; i < k; i++) {
30             Random rand = new Random();
31             int first = rand.nextInt(499)+1;
32             int second = rand.nextInt(499)+1;
33
34             this.meanX[i]=first*1.0;
35             this.meanY[i]=second*1.0;
36
37         }
38     }
39
40 }
41
42
43 void randomData(){
44     // for random x and y data
45     for (int i = 0; i < num; i++) {
46         Random rand = new Random();
47         int first = rand.nextInt(499)+1;
48         int second = rand.nextInt(499)+1;
49         x[i]=first;
50         y[i]=second;
51
52     }
53
54 }
```

```

55 void assignCluster() {
56     // Calculate the distance between the point and the cluster centre
57     // The Euclidean distance between the jth data point and ith cluster centre is:
58     double distance[]=new double[k]; // stores the k distances
59     for (int i = 0; i < num; i++) {
60         for (int j = 0; j < k; j++) {
61             distance[j] = Math.sqrt(Math.pow(x[i]-meanX[j],2) + Math.pow(y[i]-meanY[j],2));
62         }
63
64         double minValue = distance[0];
65         int minIndex =0;
66         for (int j = 0; j < k; j++) {
67             if (distance[j] < minValue) {
68                 minValue = distance[j];
69                 minIndex = j;
70             }
71         }
72         cAssign[i]=minIndex;
73     }
74
75     // Calculate distance for all k clusters and assign j to whichever i has the smallest
76     distance
77     // Assign this i to cAssign
78 }
79
80 void updateMeans() {
81     // Before updating the centres, backup meanX and meanY (copy to oldX and oldY)
82
83     for (int j = 0; j < k; j++) {
84         oldX[j]=meanX[j];
85         oldY[j]=meanY[j];
86     }
87
88     // Calculate meanX and meanY for each cluster
89
90     //this stores total sum in meanX and meanY
91     for (int j = 0; j < k; j++) {
92         for (int i = 0; i < num; i++) {
93             if(cAssign[i]==j){
94                 meanX[j]+=x[i];
95                 meanY[j]+=y[i];
96             }
97         }
98     }
99
100     //updatindg meanX and meanY by dividing it with num ie 100 [which eventually the mean]
101     for (int j = 0; j < k; j++) {
102         meanX[j]=meanX[j]/num;
103         meanY[j]=meanY[j]/num;
104     }
105
106 }
107
108 boolean isDifferent() {
109     //return true if meanX!=oldX or meanY!=oldY for one or more clusters
110     //Otherwise return false
111

```

```

112
113
114     if(meanX[0]!=oldX[0]){
115         return true;    //returns true
116     }else{
117         return false;
118     }
119 }
120
121
122
123 void doClustering() {
124     //This is where you implement the clustering algorithm. Simple isn't it.
125     randomMean();
126     randomData();
127     do {
128         assignCluster();
129         updateMeans();
130     } while(isDifferent());
131 }
132
133 public static void main(String[] args){
134
135     kmeans km = new kmeans(100,2);
136     km.doClustering();
137     System.out.println("The X and Y values are");
138     System.out.println(Arrays.toString(km.x));
139     System.out.println(Arrays.toString(km.y));
140     System.out.println("The value after k clustering");
141     System.out.println(Arrays.toString(km.cAssign));
142     //     System.out.println(Arrays.toString(km.meanX));
143     //     System.out.println(Arrays.toString(km.meanY));
144     //     System.out.println(Arrays.toString(km.oldX));
145     //     System.out.println(Arrays.toString(km.oldY));
146
147
148
149 }
150
151 }
152

```