Name - Prathamesh Rajbhoj

Batch - A2

Roll No - 53

Practical 09 - K MEANS ALGORITHM

```
import numpy as np
import math
```

```
x = [1.0, 1.5, 3.0, 5.0, 3.5, 4.5, 3.5]
y = [1.0, 2.0, 4.0, 7.0, 5.0, 5.0, 4.5]
```

```
c1x = x[0]
c1y = y[0]
c2x = x[3]
c2y = y[3]
```

```
prevClusters = []
currClusters = []
```

```
def calCentroid():
  global c1x, c1y, c2x, c2y
  xsum1, ysum1, cnt1, xsum2, ysum2, cnt2, xsum3, ysum3, cnt3 = 0,0,0,0,0,0,0,0,0
  for i in range(0, len(x)):
    if(currClusters[i] == 1):
      xsum1 += x[i]
      ysum1 += y[i]
      cnt1 +=1

    elif(currClusters[i]==2):
      xsum2 += x[i]
      ysum2 += y[i]
      cnt2+=1

  c1x = xsum1/cnt1
  c1y = ysum1/cnt1
  c2x = xsum2/cnt2
  c2y = ysum2/cnt2

  print("Centroid1 = ", c1x,c1y)
  print("Centroid2 = ", c2x,c2y)
```

```
def findClusters(a,b):
  global c1x, c1y, c2x, c2y, c3x, c3y
  dist1 = math.sqrt((a-c1x)*(a-c1x) + (b-c1y)*(b-c1y))
  dist2 = math.sqrt((a-c2x)*(a-c2x) + (b-c2y)*(b-c2y))

  mini = min(dist1, dist2)
  if mini == dist1:
    return 1
  elif mini == dist2:
    return 2
```

```
while(1):
    if len(prevClusters)!=0 and prevClusters == currClusters:
      break

    prevClusters[:] = currClusters

    currClusters.clear()
    for i in range(0, len(x)):
      currClusters.append(findClusters(x[i], y[i]))

    print("Clusters array - ", currClusters)
    calCentroid()
```

```
 Clusters array -  [1, 1, 1, 2, 2, 2, 2]
 Centroid1 =  1.8333333333333333 2.3333333333333335
 Centroid2 =  4.125 5.375
 Clusters array -  [1, 1, 2, 2, 2, 2, 2]
 Centroid1 =  1.25 1.5
 Centroid2 =  3.9 5.1
```

```
Clusters array -  [1, 1, 2, 2, 2, 2, 2]
Centroid1 =  1.25 1.5
Centroid2 =  3.9 5.1
```