Name: Mareena Fernandes

TE IT	Roll number : 8669
Expt number : 5	Date of implementation: 10/05/2021
Aim : To implement clustering algorithm	
Programming language used : Python	
Related Course outcome : CO2 Upon completion of this course students will be able to Implement the appropriate data mining methods like classification, clustering or association mining on large data sets	

Theory : The process of grouping a set of physical or abstract objects into classes of similar objects is called clustering. A cluster is a collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters. A cluster of data objects can be treated collectively as one group and so may be considered as a form of data compression. Cluster analysis is an important human activity. Cluster analysis has been widely used in numerous applications., including market research, pattern recognition, data analysis and image processing.

Clustering is also called data segmentation in some applications because clustering partitions large data sets into groups according to their similarity. Clustering can also be used in outlier detection where outliers may be more interesting than common case. In general, the major clustering methods can be classified into the following categories.

(1) **Partitioning Methods** : Given a database of n objects or data tuples, a partitioning method constructs k partitions of the data, where each partition represents a cluster and k<=n. That is it classifies the data into k groups, in such a way that each group contains atleast one object and each belongs to exactly one group. Given k, the number of partitions to construct, a partitioning method creates an initial partitioning. It then uses an iterative relocation technique that attempts to improve the partitioning by moving objects from one group to another.

(2) **Hierarchical methods** : This method creates a hierarchical decomposition of the given set of data objects. A hierarchical method can be classified as being either agglomerative or divisive, based on how the hierarchical decomposition is formed. Agglomerative approach also called the bottom-up approach, successively merges the objects that are close together, until all of the groups are merged into one. The divisive approach , also called top down approach, starts with all of the objects in same cluster. In each successive iteration, a cluster is split up into smaller clusters, until each object is in one cluster.

(3) **Density-based methods** : The general idea of these methods is to continue growing the given cluster as long as the density (number of objects or data points) in the "neighborhood" exceeds some threshold; that is for each data point within a given cluster, the neighborhood of a given radius has to contain at least a

minimum number of points. Such a method can be used to filter out noise (outliers) and discover clusters of arbitrary shape

Algorithm implemented is **K-Means Clustering Algorithm**

Code:

```
def dist(l1,l2):
   return(((l1[0]-l2[0])**2+(l1[1]-l2[1])**2)**0.5)

def get_no(new_data,no):
   global centroid
   l=list()
   for i in range(0,no):
      l.append(dist(new_data,centroid[i]))
   return(l.index(min(l)))

file = open("k-mean.csv", "r")
data=list()
for i in file:
   l=i.split(",")
   first=int(l[0].lstrip("\n"))
   second=int(l[1].rstrip('\n'))
   m=list()
   m.append(first)
   m.append(second)
   data.append(m)

centroid=dict()
no=int(input("Enter the number of clusters"))
for i in range(0,no):
   centroid[i]=data[i]

clusters=dict()
for i in range(0,no):
   clusters[i]=i

for i in range(no,len(data)):
   clust_no=get_no(data[i],no)
   clusters[i]=clust_no
   centroid1=centroid.get(clust_no)
   centroid2=data[i]
   new_centroid=list()
   new_centroid.append((centroid1[0]+centroid2[0])/2)
   new_centroid.append((centroid1[1]+centroid2[1])/2)
   centroid[clust_no]=new_centroid
print(clusters)

for i in range(0,len(data)):
   print("Tuple ",i," is in cluster ",clusters[i])
```

Output:

```
Enter the number of clusters: 2
{0: 0, 1: 1, 2: 1, 3: 0, 4: 0, 5: 0, 6: 0, 7: 0, 8: 0, 9: 0, 10: 1, 11: 0}
Tuple 0 is in cluster 0
Tuple 1 is in cluster 1
Tuple 2 is in cluster 1
Tuple 3 is in cluster 0
Tuple 4 is in cluster 0
Tuple 5 is in cluster 0
Tuple 6 is in cluster 0
Tuple 7 is in cluster 0
Tuple 8 is in cluster 0
Tuple 9 is in cluster 0
Tuple 10 is in cluster 1
Tuple 11 is in cluster 0
Users\Leandra$\Desktop\dmbi:python dmbi.py
Enter the number of clusters: 3
{0: 0, 1: 1, 2: 2, 3: 0, 4: 0, 5: 0, 6: 0, 7: 0, 8: 0, 9: 0, 10: 2, 11: 0}
Tuple 0 is in cluster 0
Tuple 1 is in cluster 1
Tuple 2 is in cluster 2
Tuple 3 is in cluster 0
Tuple 4 is in cluster 0
Tuple 5 is in cluster 0
Tuple 6 is in cluster 0
Tuple 7 is in cluster 0
Tuple 8 is in cluster 0
Tuple 9 is in cluster 0
Tuple 10 is in cluster 2
Tuple 11 is in cluster 0
```

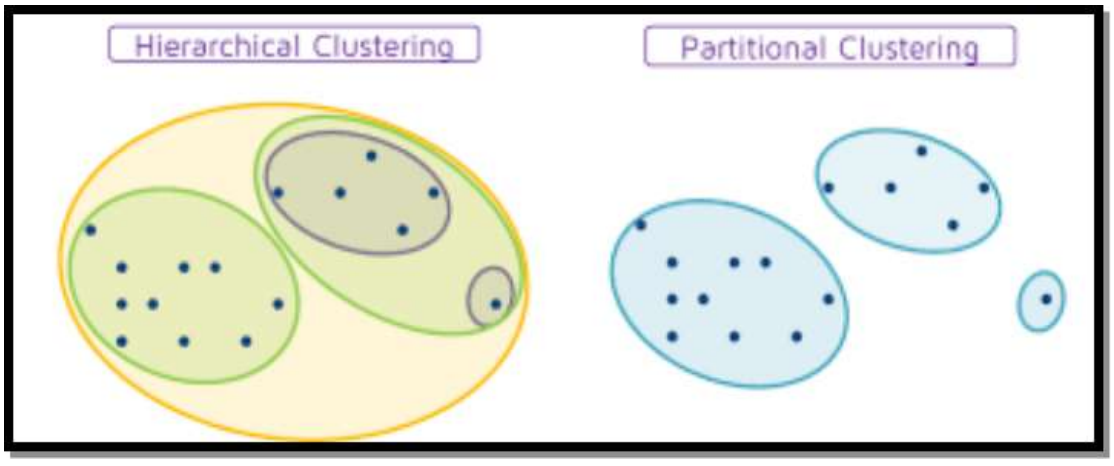
Post-lab Questions:

1. Suppose that the data mining task is to cluster the following eight points into three clusters: $A_1(2,10)$, $A_2(2,5)$,$A_3(8,4)$,$B_1(5,8)$,$B_2(7,5)$,$B_3(6,4)$,$C_1(1,2)$,$C_2(4,9)$
   The distance function is Euclidean distance. Suppose initially we assign A_1,B_1,C_1 as the centre of each cluster, respectively. Use the K-means algorithm to show only
   a) The three cluster centers after the first round execution.
   b) The final three clusters.

Ans:

a) The three cluster centers after the first round of execution. The three clusters are A1, B1, and C1, so calculating the Euclidean distance between each point from all the three clusters.
   First Iteration:
   A1 A2 A3 B1 B2 B3 C1 C2
   Centroid:1(A1)
   0 5 8.48 3.6 7.07 7.21 8.06 2.23
   Centroid:2(B1) 3.6 4.24 5 0 3.6 4.12 7.21 1.41
   Centroid:3(C1)
   8.06 3.16 7.28 7.21 6.7 5.38 0 7.61
   The three clusters with cluster points are:
   Cluster 1 = {A1(2,10)}
   Cluster 2 = {A3(8,4), B1(5,8), B2(7,5), B3(6,4), C2(4,9)}
   Cluster 3= {A2(2,5), C1(4,9)}
   Calculating the center(centroid) after the first round:
   Center1 = (2,10)
   Center2 = {(5+8+7+6+4)/5, (8+4+5+4+9)/5} = (6,6)
   Center3 = (1.5, 3.5)

   b) The final three clusters
   .Second iteration:A1 A2 A3 B1 B2 B3 C1 C2
   Centroid:1(A1)0 5 8.48 3.6 7.07 7.21 8.06 2.23
   Centroid:2(B1) 4.12 4.12 2.82 2.23 1.41 2 6.4 3.6
   Centroid:3(C1)6.51 1.58 6.51 5.7 5.7 4.52 1.58 6.04
   After the third iteration the final clusters are:
   Cluster 1: {(A1, C2, B1}
   Cluster 2: {(A3, B2, B3)}
   Cluster 3: {(A2, C1)}



2. What is dendrogram ? Give examples.
Ans:

A dendrogram is a visual representation. Specifically, it is a tree or branch diagram where there are many elements at one end, and few, or one, at the other. The branches represent categories or classes and the diagram implies an order or relationship between these categories or classes. The members of these categories are similar in some fashion or have a number of characteristics in common. Another name for these categories or classes is cluster. And the process of placing the items into a specific cluster is known as clustering. Dendrograms are often used with clusters.