# LAB ASSIGNMENT – 10

## Agglomerative Hierarchical Clustering algorithm

**NAME : PRATHAPANI SATWIKA**
**REG.NO. : 20BCD7160**

**1. Develop an Agglomerative Hierarchical Clustering algorithm to apply clustering on the following data objects referred by (x, y) pair: A1(2, 10), A2(2, 5), A3(8, 4), A4(5, 8), A5(7, 5), A6(6, 4), A7(1, 2), A8(4,9) .Use Euclidian distance metric to calculate distance matrix. Methodology to use to form step wise hierarchy or to update the distance matrix are: Single Linkage or Nearest-Neighbour Clustering Complete Linkage or FarthestNeighbour Clustering Average Linkage_ Mean**

**Linkage Apply both methodologies and trace the process of Agglomerative clustering. Note:**

**Develop algorithm using core functionalities and do not use any predefined packages like mlxtend.**

```
[2]  import math
     def distance(p,q):
       return math.sqrt(sum([(pi-qi)**2 for pi,qi in zip(p,q)]))
     def single_link(ci,cj):
       return min([distance(vi,vj) for vi in ci for vj in cj])
     def complete_link(ci,cj):
       return max([distance(vi,vj) for vi in ci for vj in cj])
     def average_link(ci,cj):
       distances = [distance(vi,vj) for vi in ci for vj in cj]
       return sum(distances)/len(distances)
     def get_distance_measure(M):
         if M==0:
             return single_link
         elif M==1:
             return complete_link
         else:
             return average_link
```

```
[12] class AgglomerativeHierarchicalClustering:
         def __init__(self,data,K,M):
             self.data=data
             self.N=len(data)
             self.K=K
             self.measure=get_distance_measure(M)
             self.clusters=self.init_clusters()
         def init_clusters(self):
             return {data_id: [data_point] for data_id,data_point in enumerate(self.data)}
         def find_closest_clusters(self):
             min_dist=math.inf
             closest_clusters = None
             clusters_ids=list(self.clusters.keys())
             for i,cluster_i in enumerate(clusters_ids[:-1]):
                 for j,cluster_j in enumerate(clusters_ids[i+1:]):
                     dist = self.measure(self.clusters[cluster_i,self.clusters[cluster_j]])
                     if dist<min_dist:
                         min_dist,closest_clusters=dist,(cluster_i,cluster_j)
             return closest_clusters
```

```
[5]  def merge_and_form_new_clusters(self,ci_id,cj_id):
         new_clusters = {0:self.clusters[ci_id]+self.clusters[cj_id]}
         for clusters_id in self.clusters.keys():
             if(clusters_id==ci_id)|(clusters_id==cj_id):
                continue
             new_clusters[len(new_clusters.keys())]=self.clusters[clusters_id]
         return new_clusters
     def run_algorithm(self):
         while len(self.clusters.keys())>self.K:
             closest_clusters=self.find_closest_clusters()
             self.clusters=self.merge_and_form_new_clusters(*closest_clusters)
     def print(self):
         for id,points in self.clusters.items():
             print("Cluster: {}".format(id))
             for point in points:
                 print("      {}".format(point))
```

```python
def read_data(file_name,seperator=' '):
    data=[]
    with open(file_name) as input_file:
        for row in input_file.readlines():
            data.append([float(item) for item in row.split(seperator)])
    return data
dataset = read_data("lab10_input.txt")
N=len(dataset)
K=3
```

Lab > lab10_input.txt

```
2 10
2 5
8 4
5 8
7 5
6 4
1 2
4 9
```

```python
for i in range(0,3):
    M=i
    if i==0: print("Single Linkage Agglomerative Hierarchical Clustering: ")
    elif i==1: print("Complete Linkage Agglomerative Hierarchical Clustering: ")
    else: print("Average Linkage Agglomerative Hierarchical Clustering: ")
    agg_hierarchical_clustering = AgglomerativeHierarchicalClustering(dataset,K,M)
    agg_hierarchical_clustering.run_algorithm()
    agg_hierarchical_clustering.print()
```

```
Single Linkage Agglomerative Hierarchical Clustering:
Cluster: 0
    [2.0, 5.0]
    [1.0, 2.0]
Cluster: 1
    [5.0, 8.0]
    [4.0, 9.0]
    [2.0, 10.0]
Cluster: 2
    [8.0, 4.0]
    [7.0, 5.0]
    [6.0, 4.0]
```

•

```
    [6.0, 4.0]
Complete Linkage Agglomerative Hierarchical Clustering
Cluster: 0
    [5.0, 8.0]
    [4.0, 9.0]
    [2.0, 10.0]
Cluster: 1
    [2.0, 5.0]
    [1.0, 2.0]
Cluster: 2
    [8.0, 4.0]
    [7.0, 5.0]
    [6.0, 4.0]
Average Linkage Agglomerative Hierarchical Clustering
Cluster: 0
    [2.0, 5.0]
    [1.0, 2.0]
Cluster: 1
    [5.0, 8.0]
    [4.0, 9.0]
    [2.0, 10.0]
Cluster: 2
    [8.0, 4.0]
    [7.0, 5.0]
    [6.0, 4.0]
```