NAME : NAMAN THAKER

20BCE529

PRACTICAL 8 : DBSCAN

```
import pandas as pd
from sklearn.preprocessing import StandardScaler
import numpy as np

cluster_point = [[0,0],[1,0],[1,1],[9,3],[3,1],[3,0],[0,1],[3,2],[6,3],[8,3],[7,4]]
cluster_point
```

```
     [[0, 0],
      [1, 0],
      [1, 1],
      [9, 3],
      [3, 1],
      [3, 0],
      [0, 1],
      [3, 2],
      [6, 3],
      [8, 3],
      [7, 4]]
```

```
X = StandardScaler().fit_transform(cluster_point)
eps=1
MinPts=2
```

```
def dbscan(D, eps, MinPts):
    labels = [0]*len(D)
    C = 0

    for P in range(0, len(D)):
        if not (labels[P] == 0):
            continue

        NeighborPts = region_query(D, P, eps) # find neighboring point of P

        # If the number is below MinPts, this point is noise.
        if len(NeighborPts) < MinPts:
            labels[P] = -1
        else:
            # Otherwise, if there are at least MinPts nearby, use this point as the seed f
            C += 1
            grow_cluster(D, labels, P, NeighborPts, C, eps, MinPts)

    return labels


def grow_cluster(D, labels, P, NeighborPts, C, eps, MinPts):
    labels[P] = C
    i = 0
    while i < len(NeighborPts):
        # Get the next point from the queue.
        Pn = NeighborPts[i]

        if labels[Pn] == -1:
            labels[Pn] = C

        elif labels[Pn] == 0:
            # Add Pn to cluster C (Assign cluster label C).
            labels[Pn] = C

            PnNeighborPts = region_query(D, Pn, eps)

            # If Pn has at least MinPts neighbors, it's a branch point! Add all of its nei
            if len(PnNeighborPts) >= MinPts:
                NeighborPts = NeighborPts + PnNeighborPts

        i += 1


def region_query(D, P, eps):

    neighbors = []

    # For each point in the dataset...
    for Pn in range(0, len(D)):

        # If the distance is below the threshold, add it to the neighbors list.
        if np.linalg.norm(D[P] - D[Pn]) < eps:
            neighbors.append(Pn)

    return neighbors
```

```
my_labels = dbscan(X, eps, MinPts)
```

```
my_labels
```

```
[1, 1, 1, 2, 1, 1, 1, 1, 2, 2, 2]
```