Name: Jay Goyal

Btech Extc C017

DBSCAN for outlier detection

```
#Part A: Outlier Detection using a randomly generated dataset


from sklearn.cluster import DBSCAN
from sklearn.datasets import make_blobs
from numpy import random,where
import matplotlib.pyplot as plt



random.seed(121)


#generate the dataset
X,y = make_blobs(n_samples=200, centers=1, cluster_std=0.3)


plt.scatter(X[:,0],X[:,1])
```

```
<matplotlib.collections.PathCollection at 0x7f72a24bcb50>
```

```
dbscan = DBSCAN(eps=0.3, min_samples=20)
```


```
pred = dbscan.fit_predict(X)
```

```
y
```

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0])
```

```
pred
```

```
array([ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0, -1,  0,  0,  0,  0,  0, -1,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0, -1,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0, -1,  0,  0,  0,  0, -1,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0])
```

```
#sample number of outlier
anom_index = where(pred==-1)
values=X[anom_index]
```

```
anom_index
```

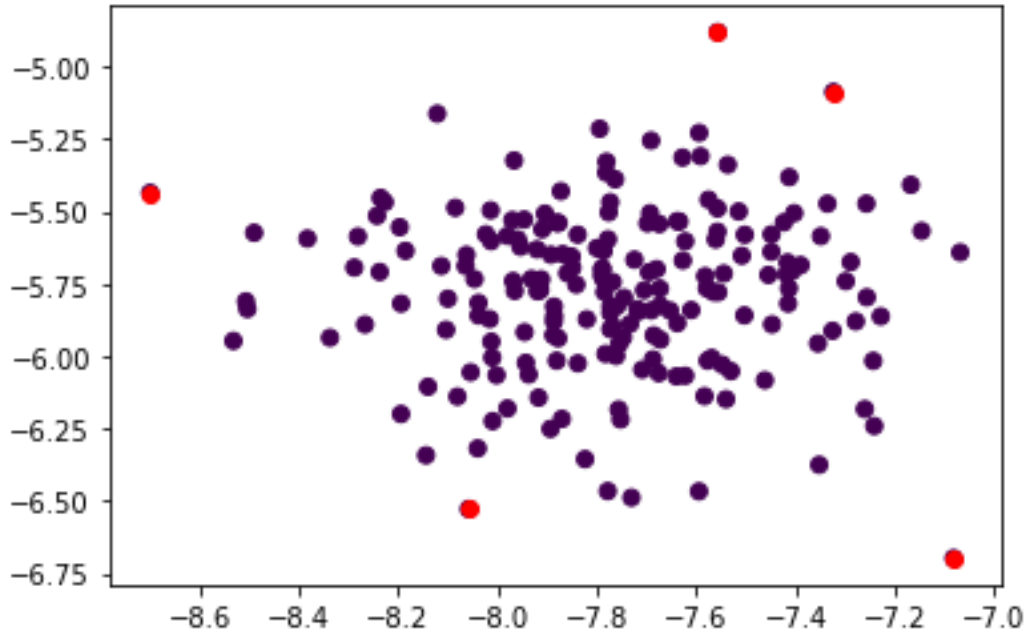
```
(array([ 72,  78,  93, 142, 147]),)
```

```
values
```

```
array([[-7.55672854, -4.87998128],
       [-8.05785552, -6.52778885],
       [-7.32379337, -5.08660199],
       [-7.08177953, -6.69838552],
       [-8.69940082, -5.43534583]])
```

```
plt.scatter(X[:,0],X[:,1],c=y)
plt.scatter(values[:,0],values[:,1],color='r')
```

```
<matplotlib.collections.PathCollection at 0x7f72a1fd7510>
```

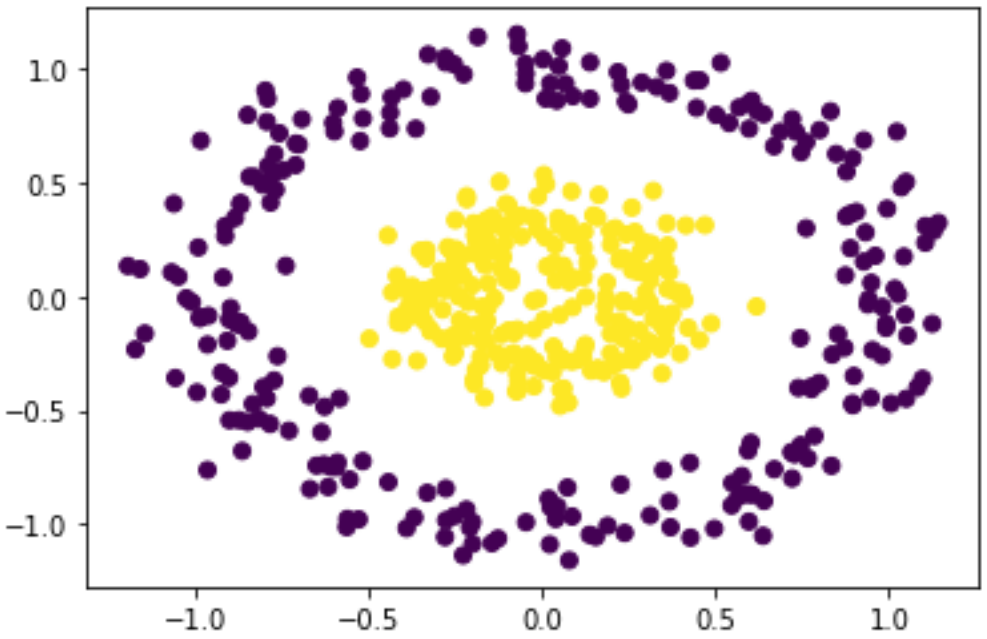



```
#Part B: Using different shapes for the dataset
from sklearn.datasets import make_circles
from sklearn.preprocessing import StandardScaler
import numpy as np
```

```
X,y =make_circles(n_samples=500, shuffle=True, noise=0.1, random_st
```

```
plt.scatter(X[:,0],X[:,1] ,c=y)
```

```
<matplotlib.collections.PathCollection at 0x7f72a1ee91d0>
```




```
#preprocess the data
scaler = StandardScaler()
X=scaler.fit_transform(X)


dbscan=DBSCAN(eps=0.3, min_samples=10)


pred=dbscan.fit_predict(X)


#sample number of outlier
anom_index = where(pred==-1)
values=X[anom_index]


anom_index
```

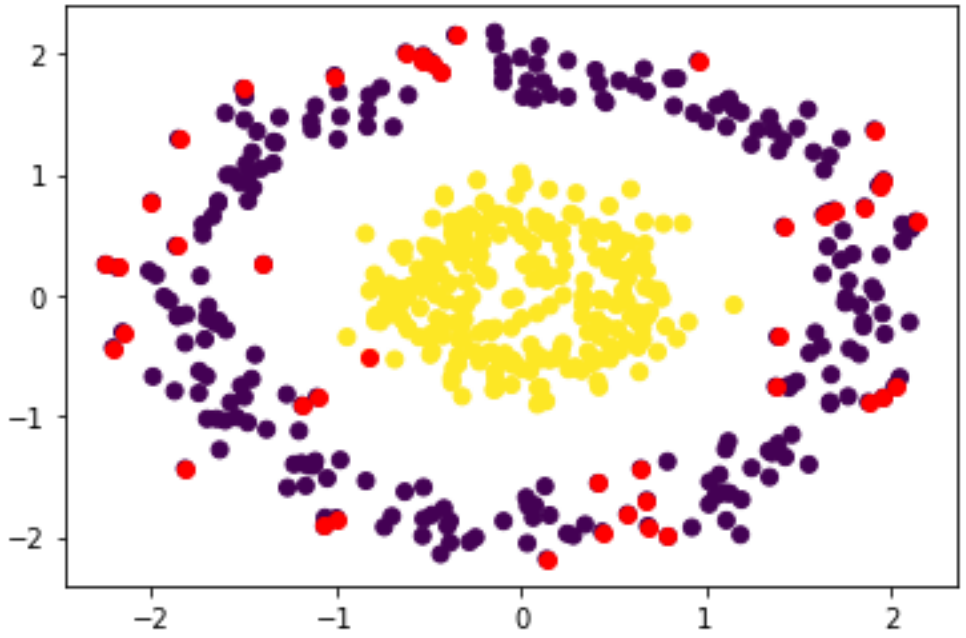
```
(array([  0,   6,   9,  33,  45,  50,  56,  59,  61,  93, 101,
        115, 133, 158, 184, 198, 227, 238, 241, 245, 247, 249,
        320, 322, 338, 357, 360, 367, 373, 376, 378, 381, 395,
        407, 412, 422, 434, 459, 474]),)
```

```
values
```

```
array([[-0.35341453,  2.15379532],
       [ 0.44101127, -1.95870939],
       [ 1.85931305,  0.73019719],
       [ 0.57835623, -1.81302439],
       [ 0.68191211, -1.69838726],
       [-2.24451714,  0.25523622],
       [ 1.93683609,  0.90729873],
       [-0.81559669, -0.51747552],
       [-0.99694867, -1.84575949],
       [-2.15281985, -0.3050887 ],
       [ 0.64997619, -1.43722915],
       [-1.39089533,  0.25652777],
       [-2.20414164, -0.43622594],
       [ 1.96290139, -0.84559291],
       [-1.10117324, -0.8440594 ],
       [-1.06184702, -1.90978427],
       [ 1.6383841 ,  0.6662127 ],
       [ 2.13722856,  0.6133235 ],
       [-0.48174288,  1.92776633],
       [ 1.66047802,  0.68638365],
       [ 1.42195854,  0.56838816],
       [ 0.14132392, -2.18521801],
       [-1.18174617, -0.91185992],
       [-1.85042094,  1.29436377],
       [-0.62134058,  2.00821214],
       [-0.52830199,  1.94267296],
       [-1.81423586, -1.43652453],
       [-1.50133281,  1.70989646],
       [ 1.69323596,  0.70446813],
       [ 0.68884671, -1.91125038],
       [-1.00562483,  1.81713578],
       [-1.99600895,  0.77185964],
       [ 1.39026253, -0.34279892],
       [ 1.88080901, -0.88467617],
       [-2.17854115,  0.2295521 ],
       [ 0.41952732, -1.55553666],
       [ 0.79660118, -1.99734076],
       [ 0.96048521,  1.94027705],
       [ 1.37996781, -0.75594469],
       [ 2.02384007, -0.75293127],
       [-0.42920487,  1.84410939],
       [-0.52741806,  1.98577761],
       [-1.86420837,  0.40873987],
       [ 1.96195129,  0.95224451],
       [ 1.91234314,  1.3686071 ]])
```

```
plt.scatter(X[:,0],X[:,1], c=y)
plt.scatter(values[:,0],values[:,1],color='r')
```

```
<matplotlib.collections.PathCollection at 0x7f72a2446d10>
```



Using dataset make_moons

```
from sklearn.datasets import make_moons

#generate the dataset
X,y = make_moons(n_samples=500, noise=0.15, random_state=1)

plt.scatter(X[:,0],X[:,1],c=y)
```

```
<matplotlib.collections.PathCollection at 0x7f72a1e2d110>
```


```
dbscan = DBSCAN(eps=0.2, min_samples=5)
```

```
pred=dbscan.fit_predict(X)
```

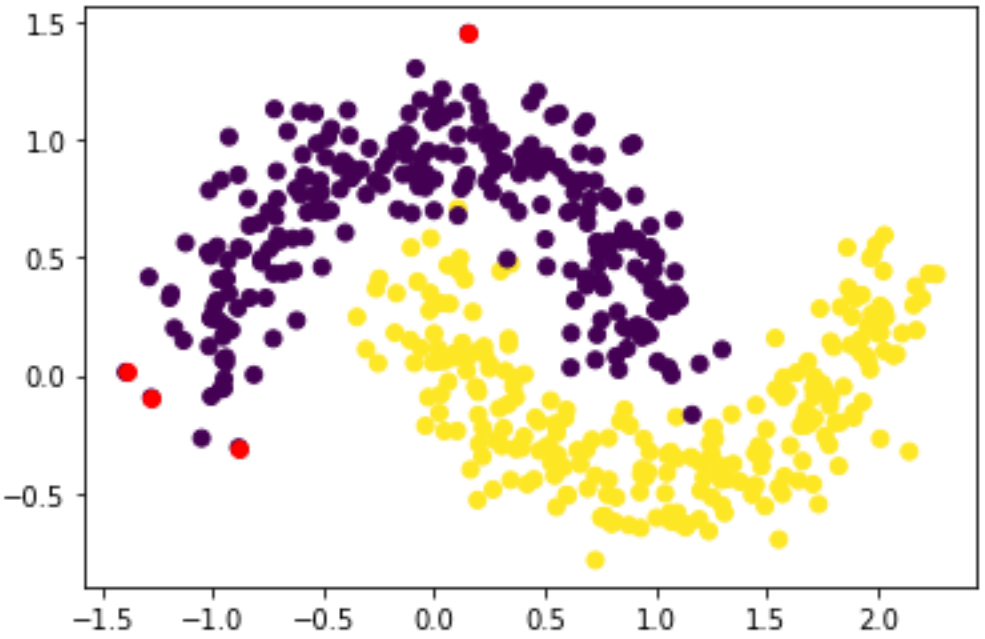

```
pred
```

```
array([ 0,  0,  0,  0,  0,  0,  0,  0, -1,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0, -1,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0, -1,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0])
```

```
#sample number of outlier
anom_index = where(pred==-1)
values=X[anom_index]


plt.scatter(X[:,0],X[:,1], c=y)
plt.scatter(values[:,0],values[:,1],color='r')
```

```
<matplotlib.collections.PathCollection at 0x7f72a1e15990>
```




```
#Part C: Apply DBscan on iris dataset


from sklearn.datasets import load_iris


iris = load_iris()


dbscan = DBSCAN(eps=0.5, min_samples=5)


dbscan.fit_predict(iris.data)
```

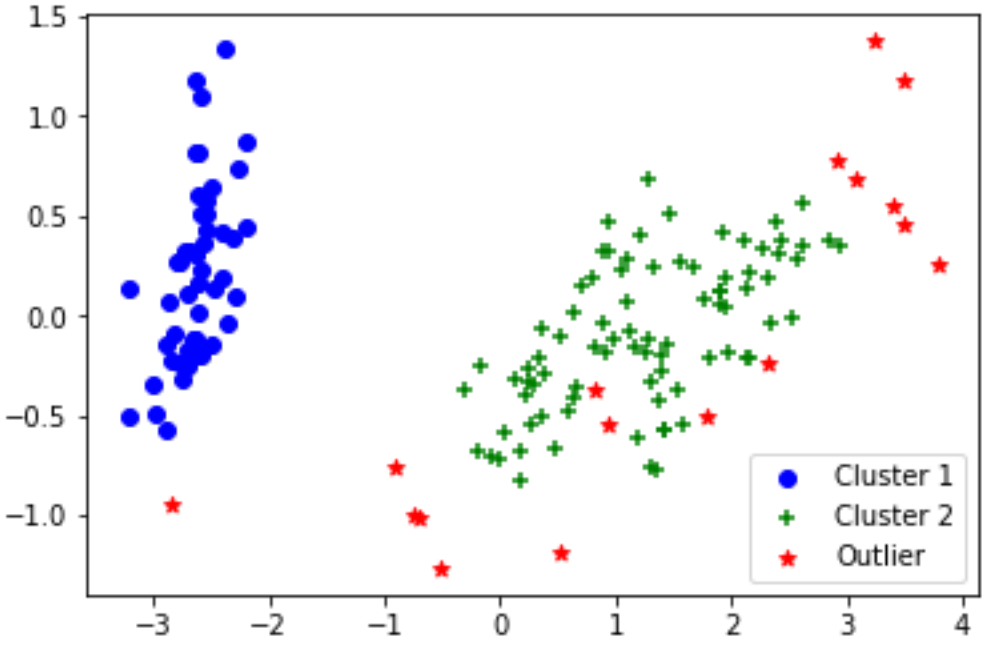
```
array([ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0, -1,  0,  0,  0,  0,  0,  0,
        1,  1,  1,  1,  1,  1, -1,  1,  1, -1,  1,  1,  1,  1,
       -1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,
        1,  1, -1,  1,  1,  1,  1,  1, -1,  1,  1,  1,  1, -1,
```

```
1,  1,  1, -1, -1,  1, -1, -1,  1,  1,  1,  1,  1,  1,
1,  1,  1, -1,  1,  1,  1,  1,  1,  1,  1, -1,  1,
1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1]
```

```
#reduce the feature space of iris data(originally four features: pe
from sklearn.decomposition import PCA
pca = PCA(n_components=2).fit(iris.data)
pca_2d = pca.transform(iris.data)
```

```
#Visualization the clusters formed by DBSCAN and identifying anomal
for i in range(0,pca_2d.shape[0]):
  if dbscan.labels_[i]==0:
    c1 = plt.scatter(pca_2d[i,0], pca_2d[i,1], c='b', marker ='o')
  elif dbscan.labels_[i]==1:
    c2 = plt.scatter(pca_2d[i,0], pca_2d[i,1], c='g', marker ='+')
  elif dbscan.labels_[i]==-1:
    c3 = plt.scatter(pca_2d[i,0], pca_2d[i,1], c='r', marker ='*')
plt.legend([c1,c2,c3],['Cluster 1','Cluster 2','Outlier'])
```

```
<matplotlib.legend.Legend at 0x7f729fb87dd0>
```



✓ 0s completed at 5:25 PM