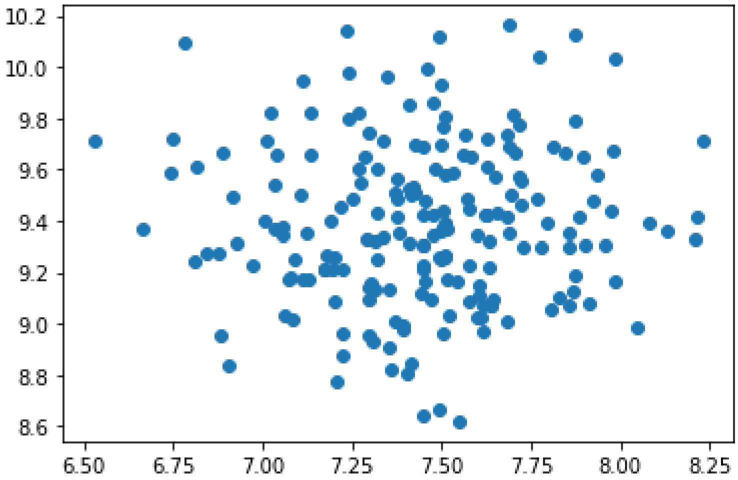
```
#Exp7: DBScan for outlier detection
#PART A: Applying on a toy dataset
```

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

```
random.seed(8)
x,y=make_blobs(n_samples=200,centers=1, cluster_std=.3)
```

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

```
<matplotlib.collections.PathCollection at 0x7fd47c7366d0>
```



```
dbscan=DBSCAN(eps=0.28,min_samples=20)
```

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

To undo cell deletion use ⌘/Ctrl+M Z or the 'Undo' option in the 'Edit' menu ✕

```
, 0 means it is a part
```

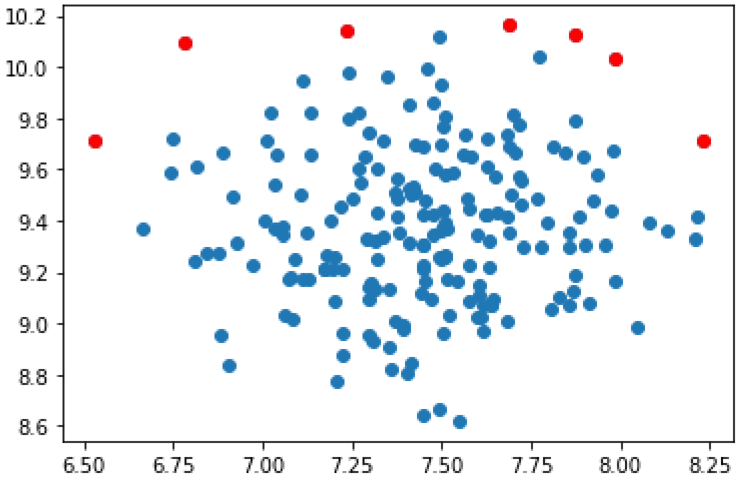
```
array([ 0,  0,  0,  0,  0,  0, -1,  0,  0,  0,  0, -1,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0, -1,  0,
        0,  0, -1,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0, 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

```
anomaly_index=where(pred==-1) #indexes of anomaly
```

```
values=x[anomaly_index]
```

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

```
    <matplotlib.collections.PathCollection at 0x7fd47c667290>
```

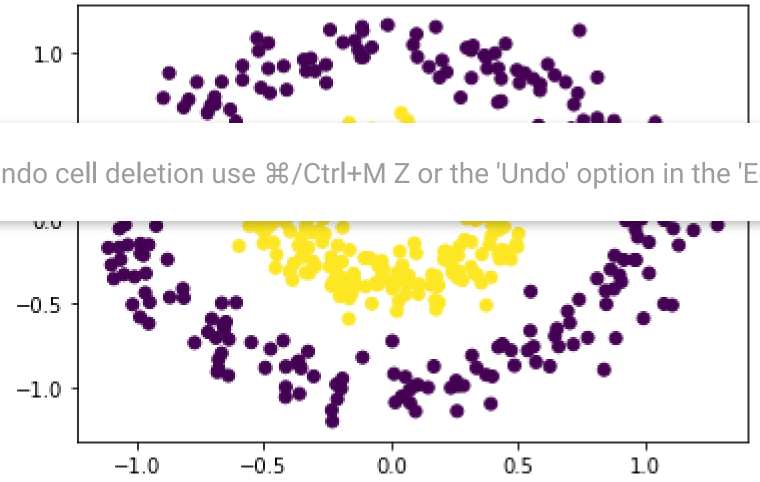


```
# PART B : dataset having a particular irregular shape
from sklearn.datasets import make_circles
from sklearn.preprocessing import StandardScaler
```

```
X,y=make_circles(n_samples=500,random_state=21, factor=0.4,noise=0.1)
```

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

```
    <matplotlib.collections.PathCollection at 0x7fd47ccd7b50>
```



To undo cell deletion use ⌘/Ctrl+M Z or the 'Undo' option in the 'Edit' menu ×

```
scaler=StandardScaler()
X=scaler.fit_transform(X)
```

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

```
y_pred #-1 are outliers again
```

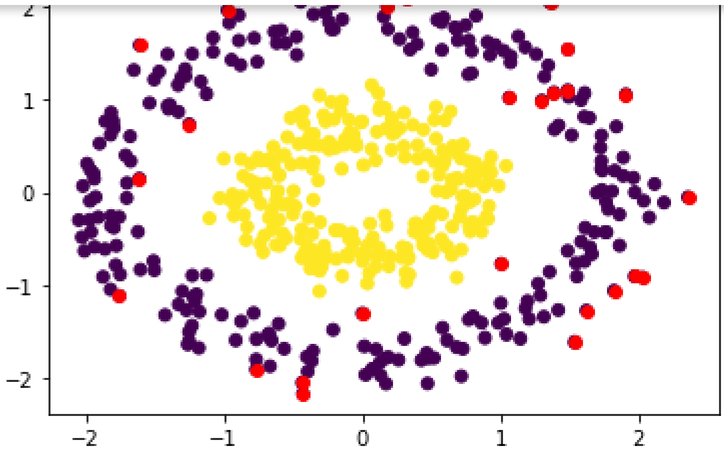
```
    array([ 0,  1,  1,  1,  1,  1,  7,  1,  2,  7,  1, 10,  1,  1,  9, -1,  3,
            2,  3,  6,  1,  1,  0,  4,  1,  1, -1,  5,  4,  5,  1,  6,  5, 10,
            1,  1,  1,  1,  1,  1,  5,  2,  4,  1,  3,  4,  5,  7,  3,  1,  5,
            3,  1,  1,  1,  3,  4,  2,  4,  1,  6,  4,  5,  3,  1,  6,  1,  1,
            2,  4,  4,  3,  5,  5,  1,  1,  2,  1,  1,  6,  1,  1,  9,  1,  9,
            1,  1,  1,  1,  0,  0,  1,  2,  1,  1,  1,  1,  1, 11,  3,  1,  1,
           -1,  1,  1,  1,  1,  3,  1,  3,  0,  2,  1,  6,  9,  1,  1, -1,  7,
            1,  1,  1,  4,  5,  5,  1,  4,  5,  1,  1,  7,  5, 11,  2,  8,  5,
            9, -1,  1,  1,  8,  1,  1, -1,  1,  2,  1,  1,  8,  1,  4, -1,  1,
            1,  3,  2,  5,  1,  1,  1,  9,  1,  8,  9,  9,  1,  4,  2,  7,  1,
            1, -1,  0,  1,  1,  8, -1,  0,  1, 10,  0, -1,  8,  1,  1,  2,  1,
            1,  1,  1,  9,  1,  6,  1,  1,  1,  1,  1,  1,  9,  3,  5,  1,  1,
            3,  5,  3,  9,  7,  1,  1,  9,  7,  3,  1,  1,  3,  1,  1,  5,  8,
            2,  8,  1,  0,  1,  1,  3,  1,  1,  2,  0,  1,  1,  8,  1,  4,  1,
            1,  1,  1,  1, -1,  1,  1, 10,  1,  2,  1,  1,  1,  1,  0,  1,  1,
            1,  1,  1, -1,  5,  9,  8,  3,  3,  3,  7, 10,  5,  6,  7, -1,  5,
            1,  1,  6, -1,  2, -1,  1,  2,  4, 10,  4, 11,  1, 11,  1,  7,  1,
            4, 11,  1,  5,  1,  1,  1,  5,  1,  1,  2,  0,  1,  1,  1, 10,  1,
            1,  1,  1,  0,  2,  1,  1, 10,  5,  1,  1,  9,  5, -1,  1,  1,  1,
            0,  1,  1,  1,  1,  1,  1,  1,  1,  1,  3,  1,  1,  1,  0,  1,  4,
           -1,  3,  1,  1,  7,  1,  4,  1,  1,  4,  5,  1,  1,  1,  1,  5,  1,
            3,  9,  4,  1,  1, 11,  2,  1, -1,  1,  1,  5,  1,  1,  1,  3,  1,
            1,  7,  1,  7,  1,  1,  5,  1,  1,  5,  1,  3,  1,  0, -1,  1,  1,
            4,  5,  2,  1,  1,  1,  1,  0,  2,  1,  1,  4,  1,  6,  8,  4,  1,
            5,  5,  0, -1, 10,  2,  1,  1,  9,  3,  4,  3,  7, -1,  1,  5,  1,
            1,  7, 10,  0,  1,  5,  1,  1,  1, -1,  1,  1,  1,  1,  2,  4,  1,
            1,  1,  1,  2,  3,  1,  1,  1,  1,  4,  1,  1,  4,  1,  1, 11,  1,
            1,  1,  4,  1,  3,  1,  1, -1,  2,  8,  5,  5,  1, -1,  1, 11,  1,
            4,  1,  2, -1,  1,  4,  1,  1,  1,  1, 10,  1,  5,  4,  9,  5,  1,
            2,  1,  3,  1,  1,  5,  1])
```

```
anomaly_index=where(y_pred==-1) #indexes of anomaly
```

```
values=X[anomaly_index]
```

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

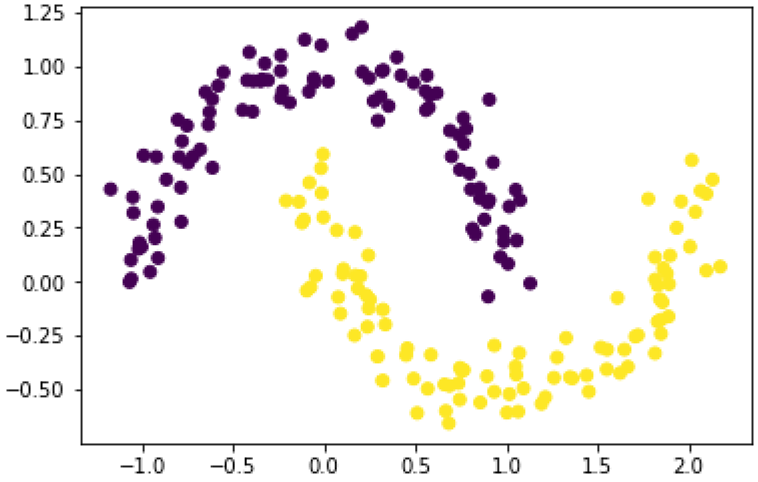
To undo cell deletion use ⌘/Ctrl+M Z or the 'Undo' option in the 'Edit' menu ×



```
from sklearn.datasets import make_moons
X,y=make_moons(n_samples=500,noise=0.1,random_state=21)
```

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

```
<matplotlib.collections.PathCollection at 0x7fd47c31ae50>
```



```
scaler=StandardScaler()
X=scaler.fit_transform(X)
dbscan=DBSCAN(eps=0.3, min_samples=10)
y_pred=dbscan.fit_predict(X)
```

```
y_pred
```

```
array([ 0,  1,  1,  2,  2,  1,  3,  6,  4,  0,  1,  2, -1,  0,  6,  0,  4,
        6, -1,  1, -1,  3,  2, -1, -1,  0, -1,  4, -1,  0,  1,  0,  0,  2,
       -1,  3,  0,  0, -1,  3,  0,  4,  4,  1, -1,  5,  0, -1,  2,  0,  5,
        0,  3,  0,  2,  3,  0, -1,  1,  4,  5,  2,  4,  0,  0,  0,  1, -1,
        0,  0,  0,  0,  0, -1,  0, -1,  1,  0,  2,  4,  0,  3, -1,  1,  0,
        0, -1,  0,  0,  2,  0,  0, -1,  2,  6,  1,  2,  5,  0,  4,  2,  3,
        5, -1,  1,  0,  2,  0,  3,  3,  4,  3,  5,  6, -1,  5,  6,  0,  2,
        2,  0,  0,  2,  4,  3, -1,  6,  1,  4,  3,  0,  0,  0,  6,  2,  0,
        0,  3,  3, -1,  4,  1,  2, -1,  3,  1,  2,  3,  4,  3,  3,  0,  0,
        0,  0,  2,  5,  0,  4,  3,  0,  0, -1,  0,  2,  6,  4,  1,  5,  1,
        0, -1,  2,  5,  1,  3,  0,  3,  1,  0,  0,  4,  0,  3, -1, -1, -1,
                                                                   3])
```

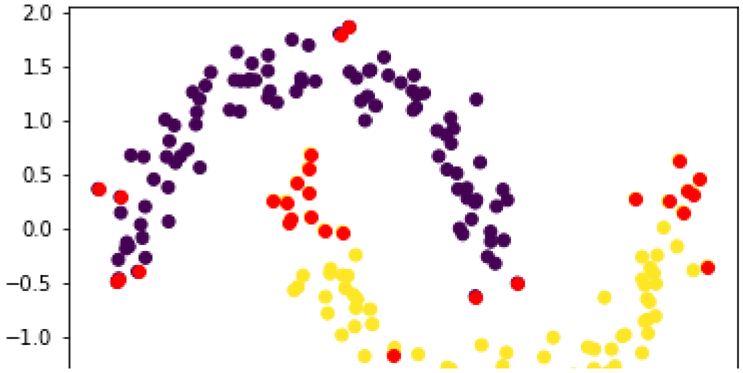
To undo cell deletion use ⌘/Ctrl+M Z or the 'Undo' option in the 'Edit' menu ×

```
anomaly_index=where(y_pred==-1) #indexes of anomaly
```

```
values=X[anomaly_index]
```

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

```
    <matplotlib.collections.PathCollection at 0x7fd47c2511d0>
```



```
#PART c : applying db scan on a dataset
from sklearn.datasets import load_iris
```

```
iris=load_iris()
```

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

```
dbscan.fit(iris.data)
```

```
    DBSCAN(algorithm='auto', eps=0.5, leaf_size=30, metric='euclidean',
           metric_params=None, min_samples=5, n_jobs=None, p=None)
```

```
dbscan.labels_#outliers have labels=-1
```

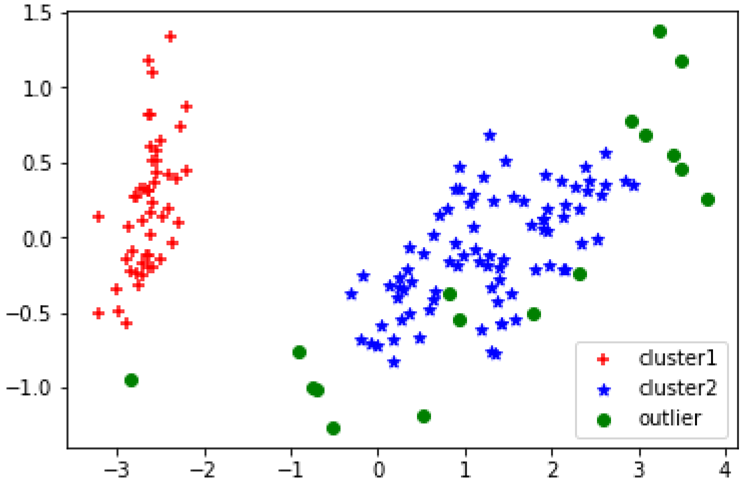
```
    array([ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
            0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
            0,  0,  0,  0,  0,  0,  0, -1,  0,  0,  0,  0,  0,  0,  0,  0,  1,
            1,  1,  1,  1,  1,  1, -1,  1,  1, -1,  1,  1, 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

To undo cell deletion use ⌘/Ctrl+M Z or the 'Undo' option in the 'Edit' menu ×

```
pca_2d=pca.transform(iris.data)
```

```
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='r',marker='+')
  elif dbscan.labels_[i]==1:
    c2=plt.scatter(pca_2d[i,0],pca_2d[i,1],c='b',marker='*')
  else:
    outlier=plt.scatter(pca_2d[i,0],pca_2d[i,1],c='g',marker='o')
plt.legend([c1,c2,outlier],['cluster1','cluster2','outlier'])
```

```
<matplotlib.legend.Legend at 0x7fd479e4ae50>
```



To undo cell deletion use ⌘/Ctrl+M Z or the 'Undo' option in the 'Edit' menu