DATAWAREHOUSING AND DATA MINING MODEL LABORATORY EXAMINATION

NAME: KARTHIKAPRIYAA V.S

CLASS: CSE –B

ROLL NO: 18BCS047

SUBJECT CODE :U18CSI6203L

EXPERIMENT

1. Download a suitable dataset for clustering from any Repository. List the attributes and its type in a word Doc.

2. Apply DBSCAN algorithm to make clusters to identify dense region using the dataset.

3. Upload in your github account. Provide the link for access.

DATA SET DESCRIPTION

DATASET:

Make_blobs from sklearn.datasets -(importing dataset)

ATTRIBUTE DESCRIPTION:

Xndarray of shape (n_samples, n_features)

The generated samples.

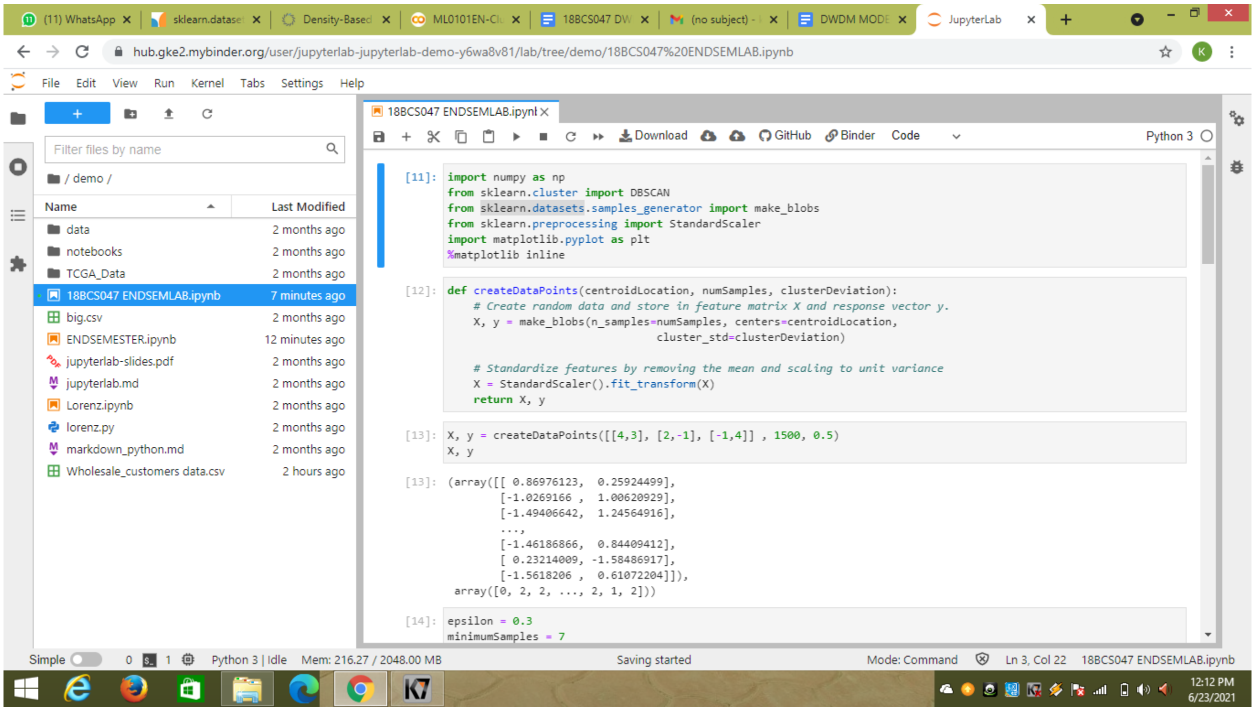
yndarray of shape (n_samples,)

The integer labels for cluster membership of each sample.

***centers**ndarray of shape (n_centers, n_features)*

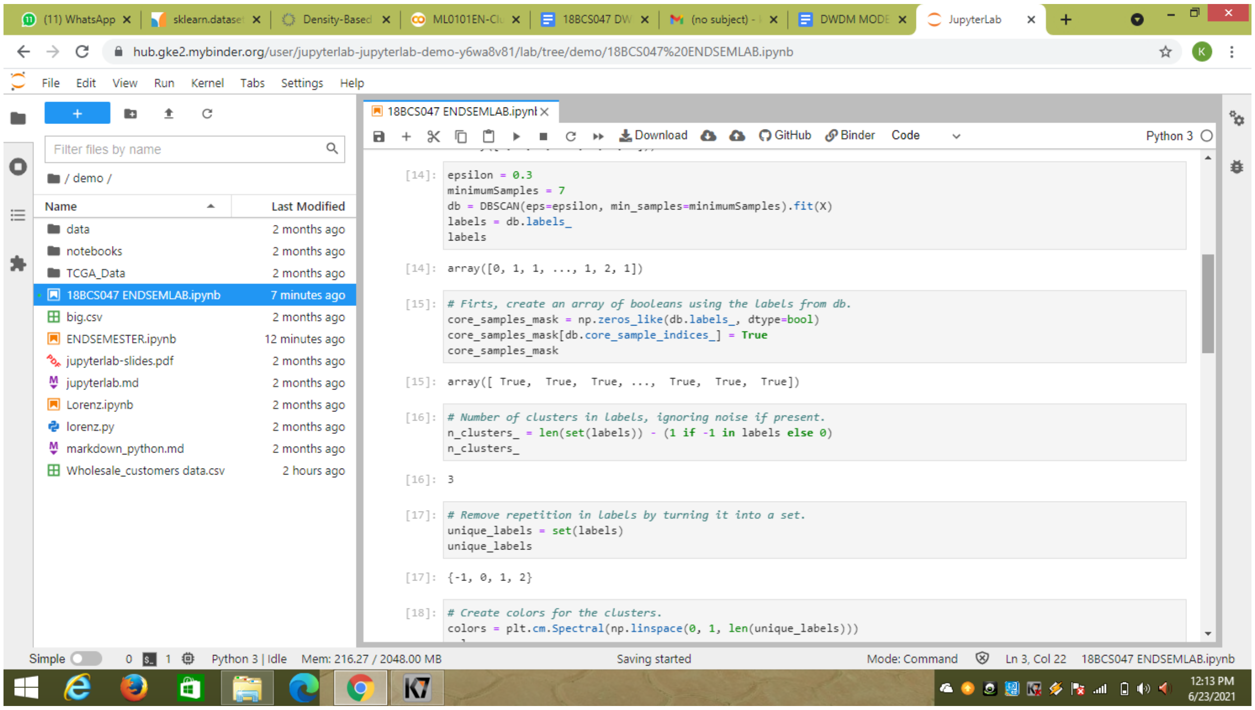
The centers of each cluster. Only returned if `return_centers=True`.

SCREENSHOT



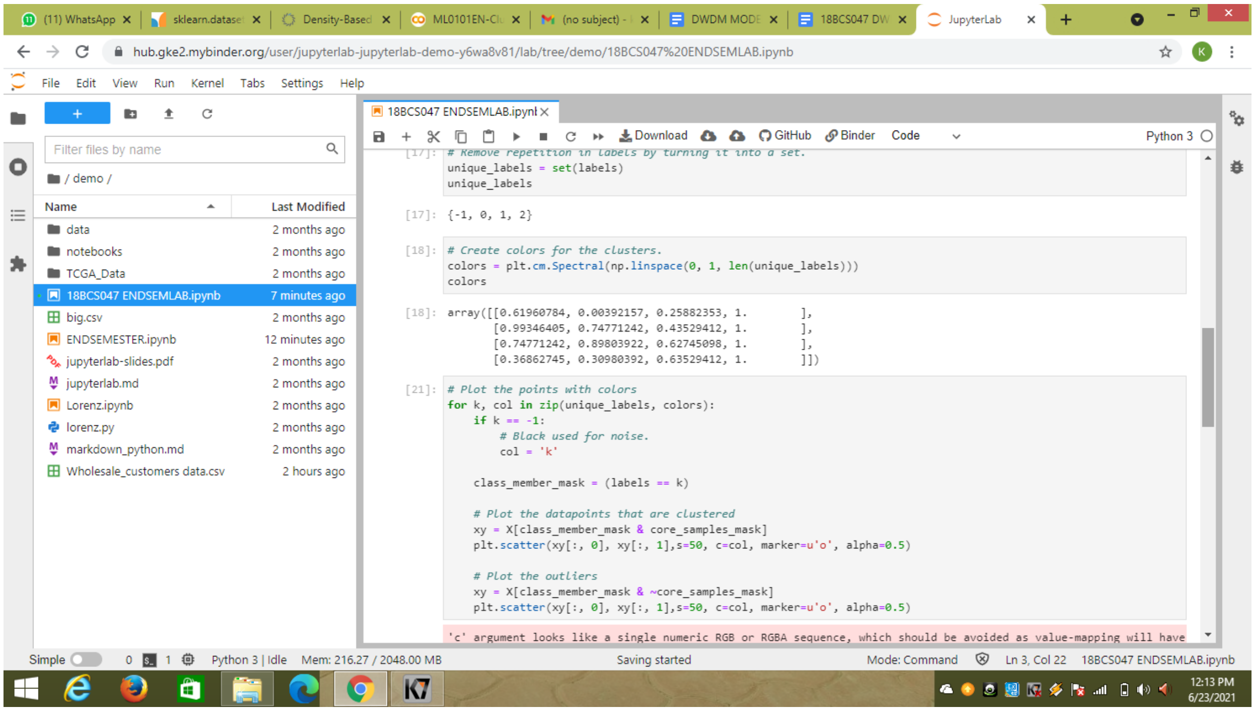
Name	Last Modified
data	2 months ago
notebooks	2 months ago
TCGA_Data	2 months ago
18BCS047 ENDSEMLAB.ipynb	7 minutes ago
big.csv	2 months ago
ENDSEMESTER.ipynb	12 minutes ago
jupyterlab-slides.pdf	2 months ago
jupyterlab.md	2 months ago
Lorenz.ipynb	2 months ago
lorenz.py	2 months ago
markdown_python.md	2 months ago
Wholesale_customers data.csv	2 hours ago

```
[11]: import numpy as np
      from sklearn.cluster import DBSCAN
      from sklearn.datasets.samples_generator import make_blobs
      from sklearn.preprocessing import StandardScaler
      import matplotlib.pyplot as plt
      %matplotlib inline

[12]: def createDataPoints(centroidLocation, numSamples, clusterDeviation):
          # Create random data and store in feature matrix X and response vector y.
          X, y = make_blobs(n_samples=numSamples, centers=centroidLocation,
                                      cluster_std=clusterDeviation)

          # Standardize features by removing the mean and scaling to unit variance
          X = StandardScaler().fit_transform(X)
          return X, y

[13]: X, y = createDataPoints([[4,3], [2,-1], [-1,4]] , 1500, 0.5)
      X, y

[13]: (array([[ 0.86976123,  0.25924499],
              [-1.0269166 ,  1.00620929],
              [-1.49406642,  1.24564916],
              ...,
              [-1.46186866,  0.84409412],
              [ 0.23214009, -1.58486917],
              [-1.5618206 ,  0.61072204]]),
       array([0, 2, 2, ..., 2, 1, 2]))

[14]: epsilon = 0.3
      minimumSamples = 7
```


(11) WhatsApp
sklearn.datase
Density-Based
ML0101EN-Cl
18BCS047 DW
(no subject) -
DWDM MODE
JupyterLab
hub.gke2.mybinder.org/user/jupyterlab-jupyterlab-demo-y6wa8v81/lab/tree/demo/18BCS047%20ENDSEMLAB.ipynb
File Edit View Run Kernel Tabs Settings Help
Filter files by name
/ demo /
18BCS047 ENDSEMLAB.ipynb
Download GitHub Binder Code
Python 3
Simple 0 1 Python 3 | Idle Mem: 216.27 / 2048.00 MB Saving started Mode: Command Ln 3, Col 22 18BCS047 ENDSEMLAB.ipynb
12:13 PM
6/23/2021

Name	Last Modified
data	2 months ago
notebooks	2 months ago
TCGA_Data	2 months ago
18BCS047 ENDSEMLAB.ipynb	7 minutes ago
big.csv	2 months ago
ENDSEMESTER.ipynb	12 minutes ago
jupyterlab-slides.pdf	2 months ago
jupyterlab.md	2 months ago
Lorenz.ipynb	2 months ago
lorenz.py	2 months ago
markdown_python.md	2 months ago
Wholesale_customers data.csv	2 hours ago

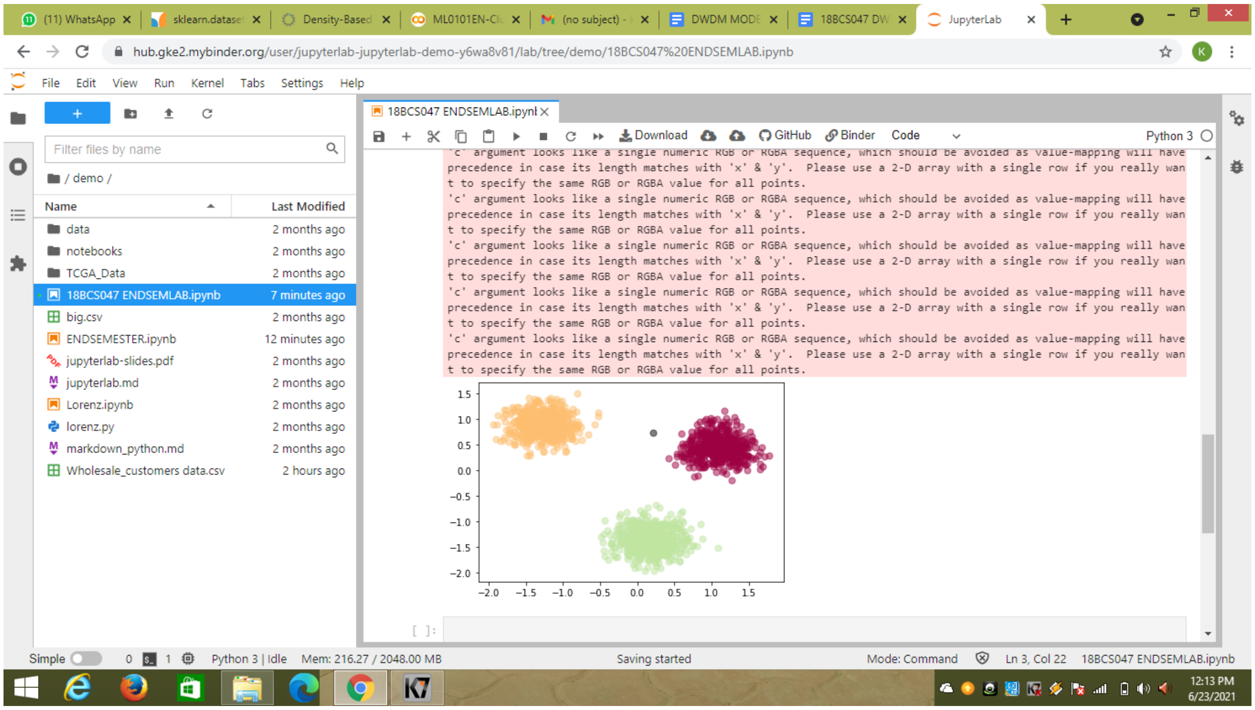
```
[14]: epsilon = 0.3
      minimumSamples = 7
      db = DBSCAN(eps=epsilon, min_samples=minimumSamples).fit(X)
      labels = db.labels_
      labels

[14]: array([0, 1, 1, ..., 1, 2, 1])

[15]: # Firts, create an array of booleans using the labels from db.
      core_samples_mask = np.zeros_like(db.labels_, dtype=bool)
      core_samples_mask[db.core_sample_indices_] = True
      core_samples_mask

[15]: array([ True,  True,  True, ...,  True,  True,  True])

[16]: # Number of clusters in labels, ignoring noise if present.
      n_clusters_ = len(set(labels)) - (1 if -1 in labels else 0)
      n_clusters_

[16]: 3

[17]: # Remove repetition in labels by turning it into a set.
      unique_labels = set(labels)
      unique_labels

[17]: {-1, 0, 1, 2}

[18]: # Create colors for the clusters.
      colors = plt.cm.Spectral(np.linspace(0, 1, len(unique_labels)))
```


(11) WhatsApp
sklearn.datase
Density-Based
ML0101EN-Cl
(no subject) -
DWDM MODE
18BCS047 DW
JupyterLab
hub.gke2.mybinder.org/user/jupyterlab-jupyterlab-demo-y6wa8v81/lab/tree/demo/18BCS047%20ENDSEMLAB.ipynb
File Edit View Run Kernel Tabs Settings Help
Filter files by name
/ demo /
18BCS047 ENDSEMLAB.ipynb
Download GitHub Binder Code
Python 3
Simple 0 1 Python 3 | Idle Mem: 216.27 / 2048.00 MB Saving started Mode: Command Ln 3, Col 22 18BCS047 ENDSEMLAB.ipynb
12:13 PM
6/23/2021

Name	Last Modified
data	2 months ago
notebooks	2 months ago
TCGA_Data	2 months ago
18BCS047 ENDSEMLAB.ipynb	7 minutes ago
big.csv	2 months ago
ENDSEMESTER.ipynb	12 minutes ago
jupyterlab-slides.pdf	2 months ago
jupyterlab.md	2 months ago
Lorenz.ipynb	2 months ago
lorenz.py	2 months ago
markdown_python.md	2 months ago
Wholesale_customers data.csv	2 hours ago

```
[17]: # Remove repetition in labels by turning it into a set.
      unique_labels = set(labels)
      unique_labels

[17]: {-1, 0, 1, 2}

[18]: # Create colors for the clusters.
      colors = plt.cm.Spectral(np.linspace(0, 1, len(unique_labels)))
      colors

[18]: array([[0.61960784, 0.00392157, 0.25882353, 1.        ],
             [0.99346405, 0.74771242, 0.43529412, 1.        ],
             [0.74771242, 0.89803922, 0.62745098, 1.        ],
             [0.36862745, 0.30980392, 0.63529412, 1.        ]])

[21]: # Plot the points with colors
      for k, col in zip(unique_labels, colors):
          if k == -1:
              # Black used for noise.
              col = 'k'

          class_member_mask = (labels == k)

          # Plot the datapoints that are clustered
          xy = X[class_member_mask & core_samples_mask]
          plt.scatter(xy[:, 0], xy[:, 1],s=50, c=col, marker=u'o', alpha=0.5)

          # Plot the outliers
          xy = X[class_member_mask & ~core_samples_mask]
          plt.scatter(xy[:, 0], xy[:, 1],s=50, c=col, marker=u'o', alpha=0.5)

      'c' argument looks like a single numeric RGB or RGBA sequence, which should be avoided as value-mapping will have
```

OUTPUT



Name	Last Modified
data	2 months ago
notebooks	2 months ago
TCGA_Data	2 months ago
18BCS047 ENDSEMLAB.ipynb	7 minutes ago
big.csv	2 months ago
ENDSEMESTER.ipynb	12 minutes ago
jupyterlab-slides.pdf	2 months ago
jupyterlab.md	2 months ago
Lorenz.ipynb	2 months ago
lorenz.py	2 months ago
markdown_python.md	2 months ago
Wholesale_customers data.csv	2 hours ago

```
'c' argument looks like a single numeric RGB or RGBA sequence, which should be avoided as value-mapping will have
precedence in case its length matches with 'x' & 'y'.  Please use a 2-D array with a single row if you really wan
t to specify the same RGB or RGBA value for all points.
'c' argument looks like a single numeric RGB or RGBA sequence, which should be avoided as value-mapping will have
precedence in case its length matches with 'x' & 'y'.  Please use a 2-D array with a single row if you really wan
t to specify the same RGB or RGBA value for all points.
'c' argument looks like a single numeric RGB or RGBA sequence, which should be avoided as value-mapping will have
precedence in case its length matches with 'x' & 'y'.  Please use a 2-D array with a single row if you really wan
t to specify the same RGB or RGBA value for all points.
'c' argument looks like a single numeric RGB or RGBA sequence, which should be avoided as value-mapping will have
precedence in case its length matches with 'x' & 'y'.  Please use a 2-D array with a single row if you really wan
t to specify the same RGB or RGBA value for all points.
'c' argument looks like a single numeric RGB or RGBA sequence, which should be avoided as value-mapping will have
precedence in case its length matches with 'x' & 'y'.  Please use a 2-D array with a single row if you really wan
t to specify the same RGB or RGBA value for all points.
```

IMPLEMENTATION:

GITHUB LINK:

https://github.com/Karthika1712/END-SEMESTER-DWDM-LAB/upload