Experiment No 8

Title: To implement DB Scan

Lab Objective: PO2: To acquire in depth understanding of various supervised and unsupervised Algorithm.

Theory:

DBSCAN Clustering (where DBSCAN is short for Density-Based Spatial Clustering of Applications with Noise) involves finding high-density areas in the domain and expanding those areas of the feature space around them as clusters.

… we present the new clustering algorithm DBSCAN relying on a density-based notion of clusters which is designed to discover clusters of arbitrary shape. DBSCAN requires only one input parameter and supports the user in determining an appropriate value for it

— A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, 1996.

The technique is described in the paper:

A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, 1996.

It is implemented via the DBSCAN class and the main configuration to tune is the “eps” and “min_samples” hyperparameters.

The complete example is listed below.

Program Code:

```
# dbscan clustering
from numpy import unique
from numpy import where
from sklearn.datasets import make_classification
from sklearn.cluster import DBSCAN
from matplotlib import pyplot
# define dataset
X, _ = make_classification(n_samples=1000, n_features=2, n_informative=2, n_redundant=0,
n_clusters_per_class=1, random_state=4)
# define the model
model = DBSCAN(eps=0.30, min_samples=9)
# fit model and predict clusters
yhat = model.fit_predict(X)
# retrieve unique clusters
clusters = unique(yhat)
# create scatter plot for samples from each cluster
for cluster in clusters:
        # get row indexes for samples with this cluster
        row_ix = where(yhat == cluster)
        # create scatter of these samples
        pyplot.scatter(X[row_ix, 0], X[row_ix, 1])
# show the plot
pyplot.show()
```

Implementation:

Exp8

File Edit View Insert Runtime Tools Help All changes saved

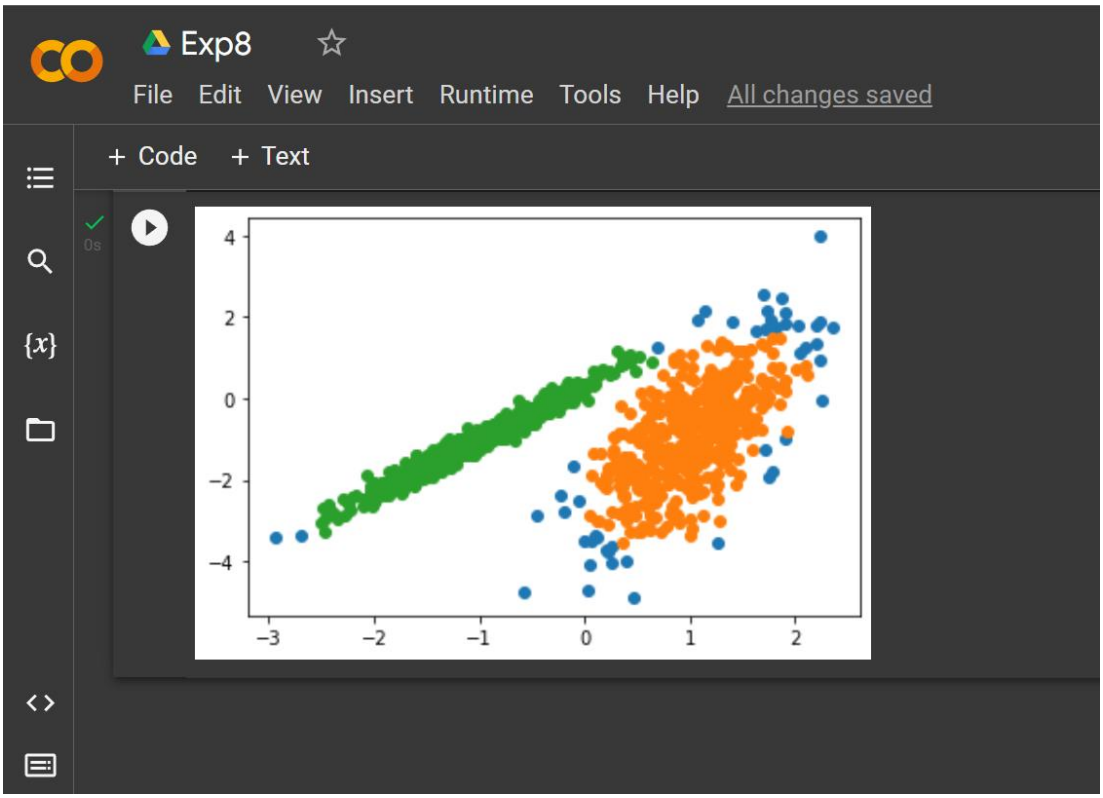
+ Code + Text

```
[1] # dbscan clustering
    from numpy import unique
    from numpy import where
    from sklearn.datasets import make_classification
    from sklearn.cluster import DBSCAN
    from matplotlib import pyplot
```

```
[2] # define dataset
    X, _ = make_classification(n_samples=1000, n_features=2, n_informative=2, n_redundant=0, n_clusters_per_class=1, random_state=4)
    # define the model
    model = DBSCAN(eps=0.30, min_samples=9)
    # fit model and predict clusters
    yhat = model.fit_predict(X)
```

```
# retrieve unique clusters
clusters = unique(yhat)
# create scatter plot for samples from each cluster
for cluster in clusters:
  # get row indexes for samples with this cluster
  row_ix = where(yhat == cluster)
  # create scatter of these samples
  pyplot.scatter(X[row_ix, 0], X[row_ix, 1])
# show the plot
pyplot.show()
```

Output:



Conclusion: Thus, in this practical we studied and performed DB Scan using python programming language.