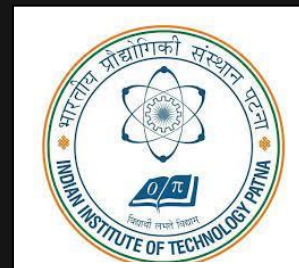**KUMAR SANATAN**

**ROLL NO – 2211AI24**

# CS 564
Foundations of Machine Learning

## ASSIGMENT 2:

INDIAN INSTITUTE OF TECHNOLOGY PATNA

**Date:** 1st Sept 2022      **Deadline:** 13th Sept 2022

# OBJECTIVE

The assignment targets to implement DBSCAN algorithms to cluster the 3 datasets with blob, moon and circle structures.

# CODE SUMMARY :

We have imported the required libraries and read the data for cluster blobs, cluster circles and cluster moons provided and further cleaning the data.

```python
import pandas as pd
from matplotlib import pyplot as plt
from sklearn.cluster import DBSCAN, KMeans
from sklearn import metrics

# Taking input
clusterBlobData=pd.read_csv("C:\\Users\\HP\\Desktop\\IIT_Patna_MTech_AI\\sem_1\\ML\\assignment2\\cluster_blobs.csv")
clusterCirclesData=pd.read_csv("C:\\Users\\HP\\Desktop\\IIT_Patna_MTech_AI\\sem_1\\ML\\assignment2\\cluster_circles.csv")
clusterMoonsData=pd.read_csv("C:\\Users\\HP\\Desktop\\IIT_Patna_MTech_AI\\sem_1\\ML\\assignment2\\cluster_moons.csv")

# Data cleaning
clusterBlobData.fillna(method ='ffill', inplace = True)
clusterCirclesData.fillna(method='ffill', inplace = True)
clusterMoonsData.fillna(method='ffill', inplace = True)
```
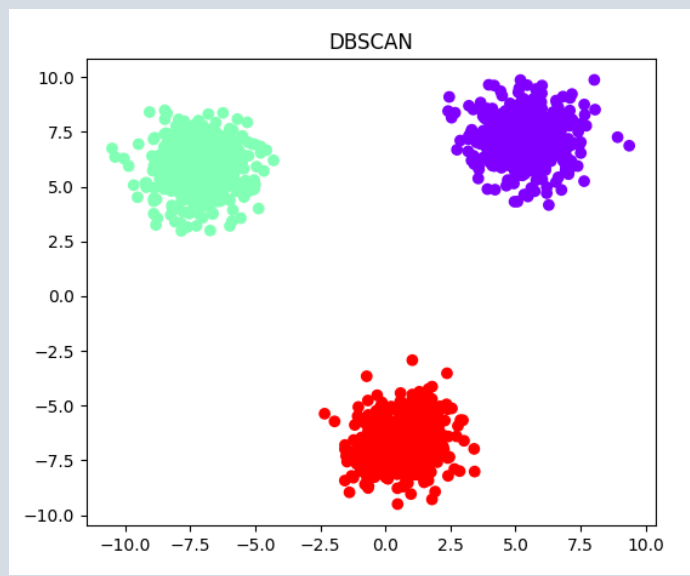
Implementation of DBSCAN algorithm for cluster blob data and calculating the silhoutte score for the same.

```python
# Cluster blobs using DBSCAN
mainClusterBlobDataDbscan = pd.DataFrame(clusterBlobData)
dbscan= DBSCAN(eps = 1.5, min_samples = 3).fit(mainClusterBlobDataDbscan).labels_
plt.figure(figsize =(10, 10))
mainClusterBlobDataDbscan.columns=['X1','X2']
plt.scatter(mainClusterBlobDataDbscan['X1'], mainClusterBlobDataDbscan['X2'],c=dbscan, cmap ='rainbow')
plt.title("DBSCAN")
plt.show()
print("The Silhoutte score of the dataset cluster_blobs using DBScan is => ",
metrics.silhouette_score(mainClusterBlobDataDbscan,dbscan))
```
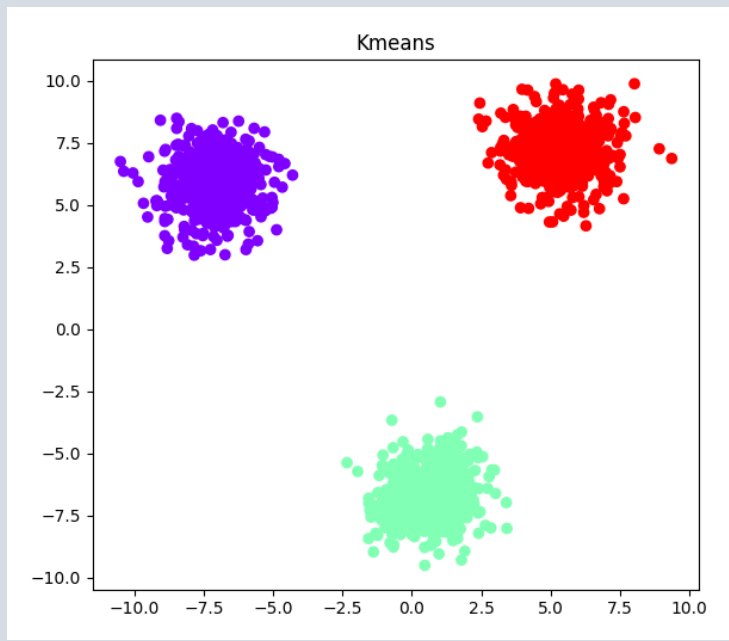
**RESULT:**



DBSCAN

The Silhoutte score of the dataset cluster_blobs using DBScan is =>  0.8625486425630479

Implementation of Kmeans algorithm for cluster blob data and calculating the silhoutte score for the same.

```python
# Cluster blobs using Kmeans
mainClusterBlobDataKmeans = pd.DataFrame(clusterBlobData)
kmeans= KMeans(3).fit(mainClusterBlobDataKmeans).labels_
plt.figure(figsize =(10, 10))
mainClusterBlobDataKmeans.columns=['X1','X2']
plt.scatter(mainClusterBlobDataKmeans['X1'], mainClusterBlobDataKmeans['X2'],c=kmeans, cmap ='rainbow')
plt.title("Kmeans")
plt.show()
print("The Silhoutte score of the dataset cluster_blobs using Kmeans is => ",
metrics.silhouette_score(mainClusterBlobDataKmeans,kmeans))
```
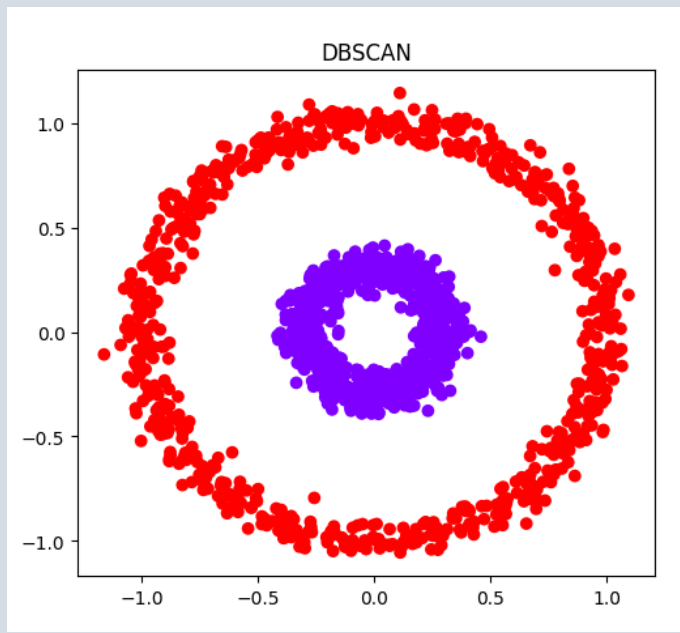
## RESULT:



Kmeans

```
The Silhoutte score of the dataset cluster_blobs using Kmeans is => 0.8625486425630479
```

Implementation of DBSCAN algorithm for cluster circle data and calculating the silhoutte score for the same.

```python
# Cluster circles using DBSCAN
mainClusterCirclesDataDbscan = pd.DataFrame(clusterCirclesData)
dbscan= DBSCAN(eps = 0.15, min_samples = 3).fit(mainClusterCirclesDataDbscan).labels_
plt.figure(figsize =(10, 10))
mainClusterCirclesDataDbscan.columns=['X1','X2']
plt.scatter(mainClusterCirclesDataDbscan['X1'], mainClusterCirclesDataDbscan['X2'],c=dbscan, cmap ='rainbow')
plt.title("DBSCAN")
plt.show()
print("The Silhoutte score of the dataset cluster_circles using DBScan => ",
metrics.silhouette_score(mainClusterCirclesDataDbscan,dbscan))
```
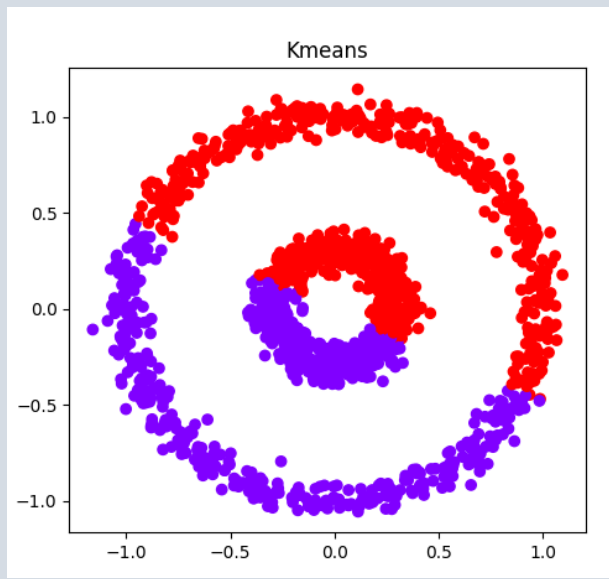
**RESULT:**



DBSCAN

The Silhoutte score of the dataset cluster_circles using DBScan => 0.20829174427109967

Implementation of Kmeans algorithm for cluster circle data and calculating the silhoutte score for the same.

```
# Cluster circles using Kmeans
mainClusterCirclesDataKmeans = pd.DataFrame(clusterCirclesData)
kmeans_1= KMeans(2).fit(mainClusterCirclesDataKmeans).labels_
plt.figure(figsize =(10, 10))
mainClusterCirclesDataKmeans.columns=['X1','X2']
plt.scatter(mainClusterCirclesDataKmeans['X1'], mainClusterCirclesDataKmeans['X2'],c=kmeans_1, cmap ='rainbow')
plt.title("Kmeans")
plt.show()
print("The Silhoutte score of the dataset cluster_circles using Kmeans is => ",
metrics.silhouette_score(mainClusterCirclesDataKmeans,kmeans_1))
```
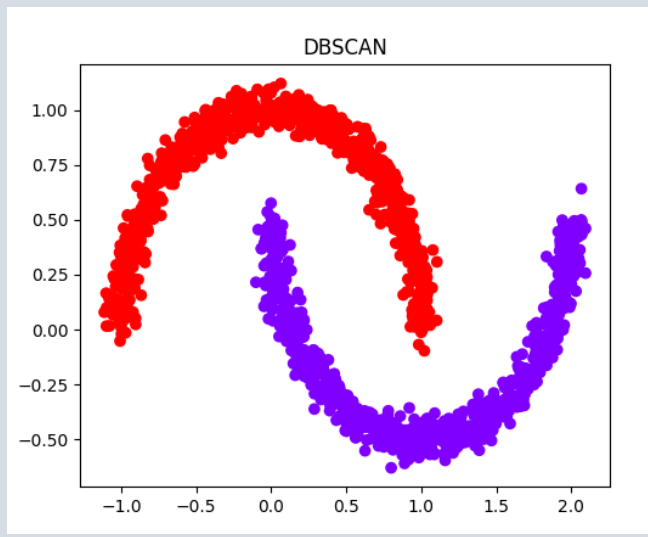
**RESULT:**



The Silhoutte score of the dataset cluster_circles using Kmeans is => 0.29549709057676377

Implementation of DBSCAN algorithm for cluster moons data and calculating the silhoutte score for the same.

```python
# Cluster moons using DBSCAN
mainClusterMoonsDataDbscan = pd.DataFrame(clusterMoonsData)
dbscan= DBSCAN(eps = 0.15, min_samples = 3).fit(mainClusterMoonsDataDbscan).labels_
plt.figure(figsize =(10, 10))
mainClusterMoonsDataDbscan.columns=['X1','X2']
plt.scatter(mainClusterMoonsDataDbscan['X1'], mainClusterMoonsDataDbscan['X2'],c=dbscan, cmap ='rainbow')
plt.title("DBSCAN")
plt.show()
print("The Silhoutte score of the dataset cluster_moons using DBScan is => ",
metrics.silhouette_score(mainClusterMoonsDataDbscan,dbscan))
```

**RESULT:**

Implementation of DBSCAN algorithm for cluster moons data and calculating the silhoutte score for the same.

```
# Cluster moons using Kmeans
mainClusterMoonsDataKmeans = pd.DataFrame(clusterMoonsData)
kmeans_2= KMeans(2).fit(mainClusterMoonsDataKmeans).labels_
plt.figure(figsize =(10, 10))
mainClusterMoonsDataKmeans.columns=['X1','X2']
plt.scatter(mainClusterMoonsDataKmeans['X1'], mainClusterMoonsDataKmeans['X2'],c=kmeans_2, cmap ='rainbow')
plt.title("Kmeans")
plt.show()
print("The Silhoutte score of the dataset cluster_moons using Kmeans is => ",
metrics.silhouette_score(mainClusterMoonsDataKmeans,kmeans_2))
```

**RESULT:**
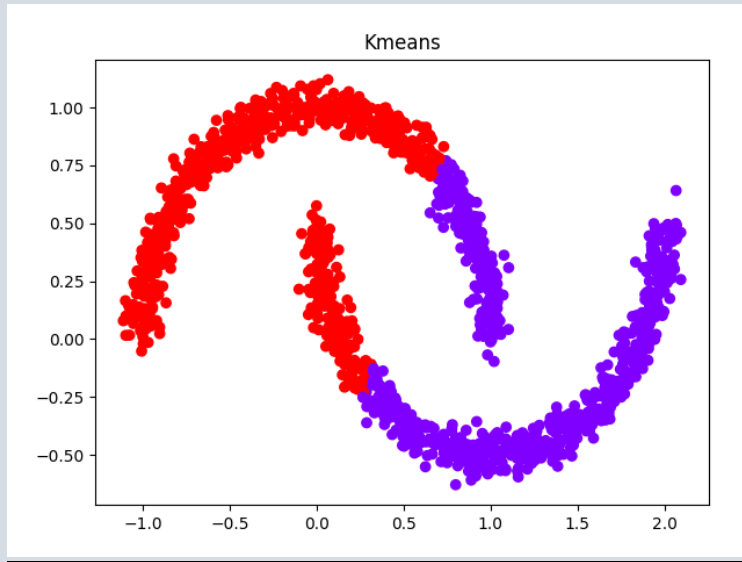


Kmeans

The Silhoutte score of the dataset cluster_moons using Kmeans is => 0.49041236609392724

**OBSERVATIONS SUMMARY:**

We have assumed all the hyperparameters via referring to the scale of the outputs projected and the efficiency in the measuring index with respect to the assumed hyperparameter value.

| SILHOUTTE SCORE | DBSCAN | KMEANS |
|---|---|---|
| Cluster blobs | 0.8625486425630479 | 0.8625486425630479 |
| Cluster circles | 0.20829174427109967 | 0.29549709057676377 |
| Cluster moons | 0.3334160815186463 | 0.49041236609392724 |