Experiment No.8

Title:Implementation of Anamoly Detection model to identify unusual pattern

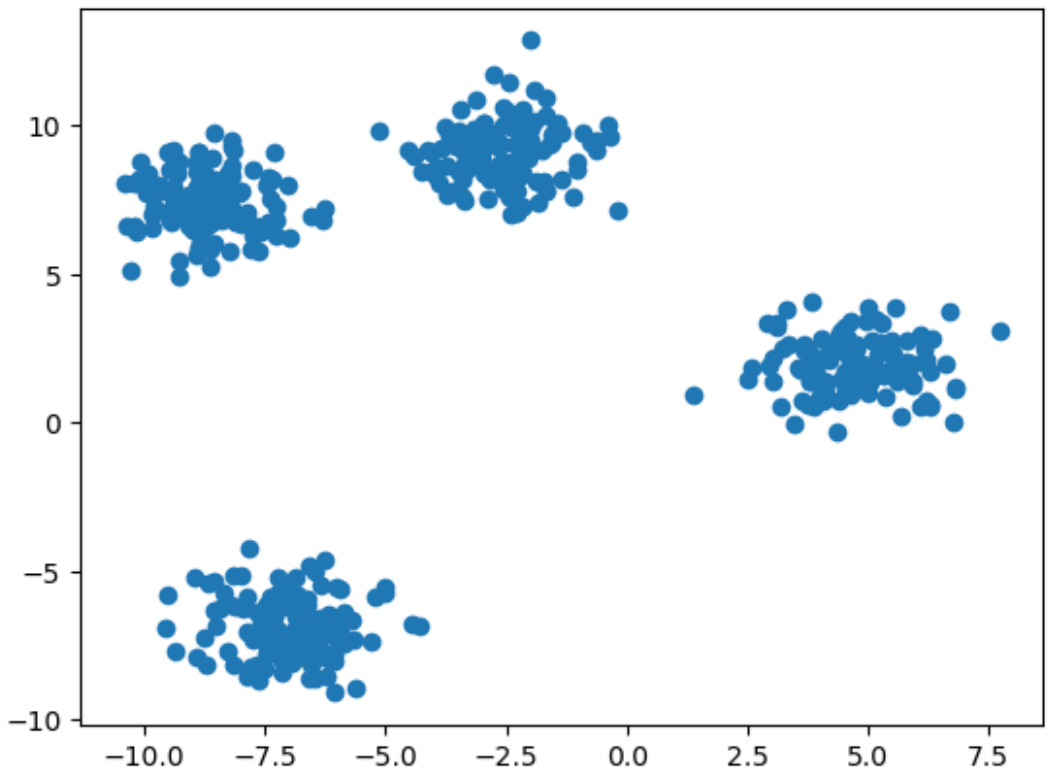
Import Libraries

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import DBSCAN
from sklearn.datasets import make_blobs
```

Make a Dataset for Clustering Experimentation

```
X,y=make_blobs(n_samples=500,centers=4,random_state=42,cluster_std=1)
plt.scatter(X[:,0],X[:,1])
```

```
<matplotlib.collections.PathCollection at 0x7ab975ad77d0>
```

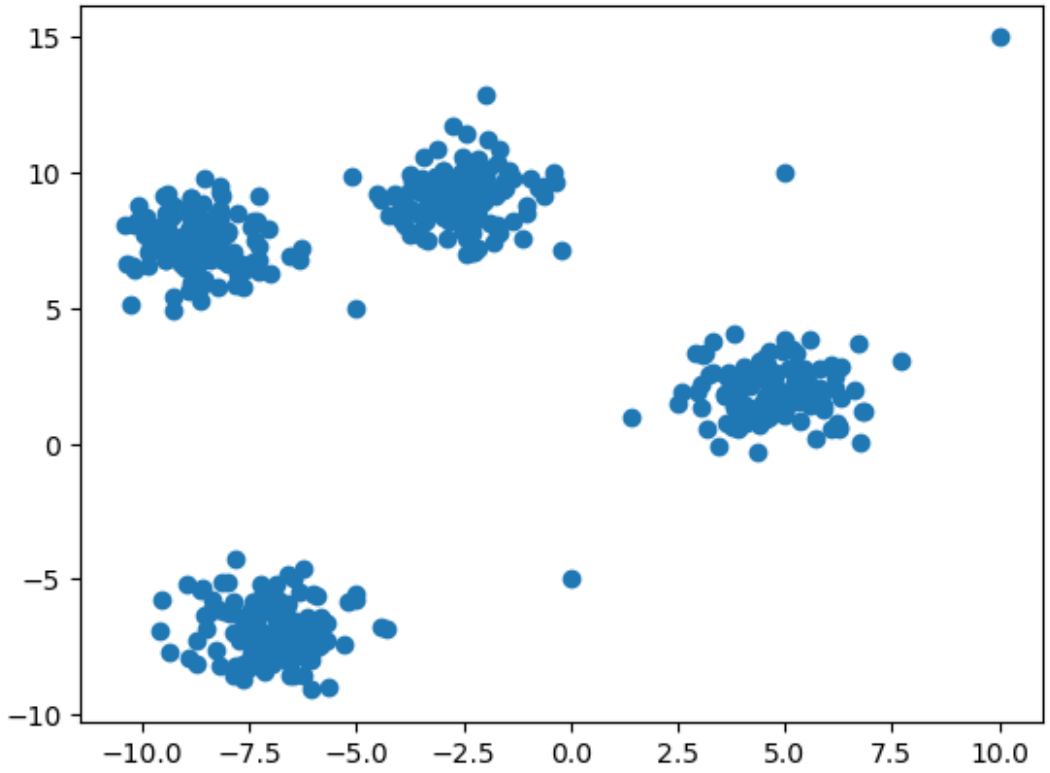


Add Some Anomalies/Outliers in the Dataset

```
import matplotlib.pyplot as plt
import numpy as np
```

```
anomalies = np.array([[10, 15], [-5, 5], [0, -5], [5, 10]])
X = np.concatenate((X, anomalies))
y = np.concatenate((y, np.array([4, 4, 4, 4])))  # Assign a new
cluster label for outliers

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



Make a DBSCAN Clustering model and find the Clusters

```
dbsacn=DBSCAN(eps=1.5,min_samples=10)
labels=dbsacn.fit_predict(X)

labels
```

```
array([ 0,  1,  2,  3,  1,  1,  0,  1,  2,  1,  2,  3,  2,  3,  1,  2,
3,
        0,  0,  3,  2,  3,  2,  0,  0,  1,  1,  0,  0,  3,  1,  3,  3,
3,
        1,  1,  2,  2,  0,  0,  1,  2,  3,  3,  3,  2,  2,  2,  1,  0,
1,
        3,  0,  1,  2,  3,  3,  0,  1,  0,  0,  3,  1,  0,  2,  1,  1,
0,
        2,  1,  2,  1,  1,  0,  3,  0,  3,  1,  2,  3,  1,  2,  1,  3,
0,
        0,  0,  0,  2,  3,  0,  1,  2,  1,  2,  0,  3,  2,  3,  0,  2,
```

```
2,
       2,  0,  0,  3,  3,  0,  3,  0,  1,  0,  0,  0,  0,  1,  2,  0,
1,
       1,  3,  2,  1,  2,  0,  2,  2,  1,  1,  0,  0,  2,  2,  1,  2,
0,
       0,  0,  2,  2,  1,  0,  2,  2,  1,  0,  0,  3,  3,  3,  1,  1,
2,
       2,  3,  0,  3,  0,  1,  1,  0,  0,  2,  2,  1,  3,  2,  1,  0,
0,
       1,  3,  3,  0,  0,  3,  3,  1,  1,  1,  3,  0,  3,  3,  0,  0,
3,
       2,  3,  1,  1,  0,  0,  1,  3,  1,  3,  3,  0,  1,  3,  3,  1,
0,
       2,  0,  1,  2,  2,  0,  0,  1,  0,  3,  3,  1,  3,  0,  3,  2,
2,
       3,  0,  3,  2,  3,  3,  1,  1,  2,  1,  2,  3,  1,  0,  1,  3,
1,
       2,  2,  2,  1,  3,  2,  0,  0,  2,  3,  3,  1,  3,  3,  3,  3,
2,
       2,  0,  3,  1,  2,  3,  3,  0,  3,  1,  1,  2,  3,  2,  1,  3,
3,
       2,  0,  3,  3,  3,  2,  0,  1,  3,  2,  2,  3,  1,  3,  2,  0,
2,
       1,  1,  3,  3,  2,  0,  2,  0,  2,  1,  2,  1,  0,  2,  2,  3,
2,
       1,  1,  2,  2,  1,  0,  2,  2,  2,  1,  0,  2,  2,  2,  0,  2,
2,
       0,  0,  0,  1,  0,  0,  0,  0,  0,  3,  0,  2,  0,  3,  3,  0,
2,
       2,  2,  3,  3,  1,  3,  1,  3,  2,  1,  1,  2,  1,  0,  0,  1,
0,
       3,  1,  3,  2,  3,  1,  1,  0,  2,  1,  2,  2,  0,  3,  1,  2,
1,
       0,  2,  0,  2,  0,  0,  1,  0,  3,  1,  0,  1,  3,  1,  0,  1,
0,
       2,  3,  3,  1,  1,  2,  3,  3,  3,  2,  0,  1,  3,  2,  2,  1,
0,
       1,  1,  1,  3,  2,  1,  3,  2,  3,  3,  2,  3,  2,  3,  0,  1,
2,
       3,  1,  0,  3,  2,  2,  1,  1,  1,  0,  1,  2,  0,  3,  3,  0,
2,
       0,  2,  3,  1,  1,  1,  3,  2,  2,  0,  3,  1,  1,  1,  2,  1,
3,
       2,  0,  2,  1,  2,  3,  2,  3,  2,  1,  3,  0,  0,  3,  3,  0,
0,
       3,  1,  2,  1,  2,  3,  0,  0,  3,  1,  0,  0,  2,  0,  0,  3,
0,
       1,  3,  0,  3,  1,  1,  2, -1, -1, -1, -1])
```

Find the Anomalies

Note: The Anomalies are with lablel as -1

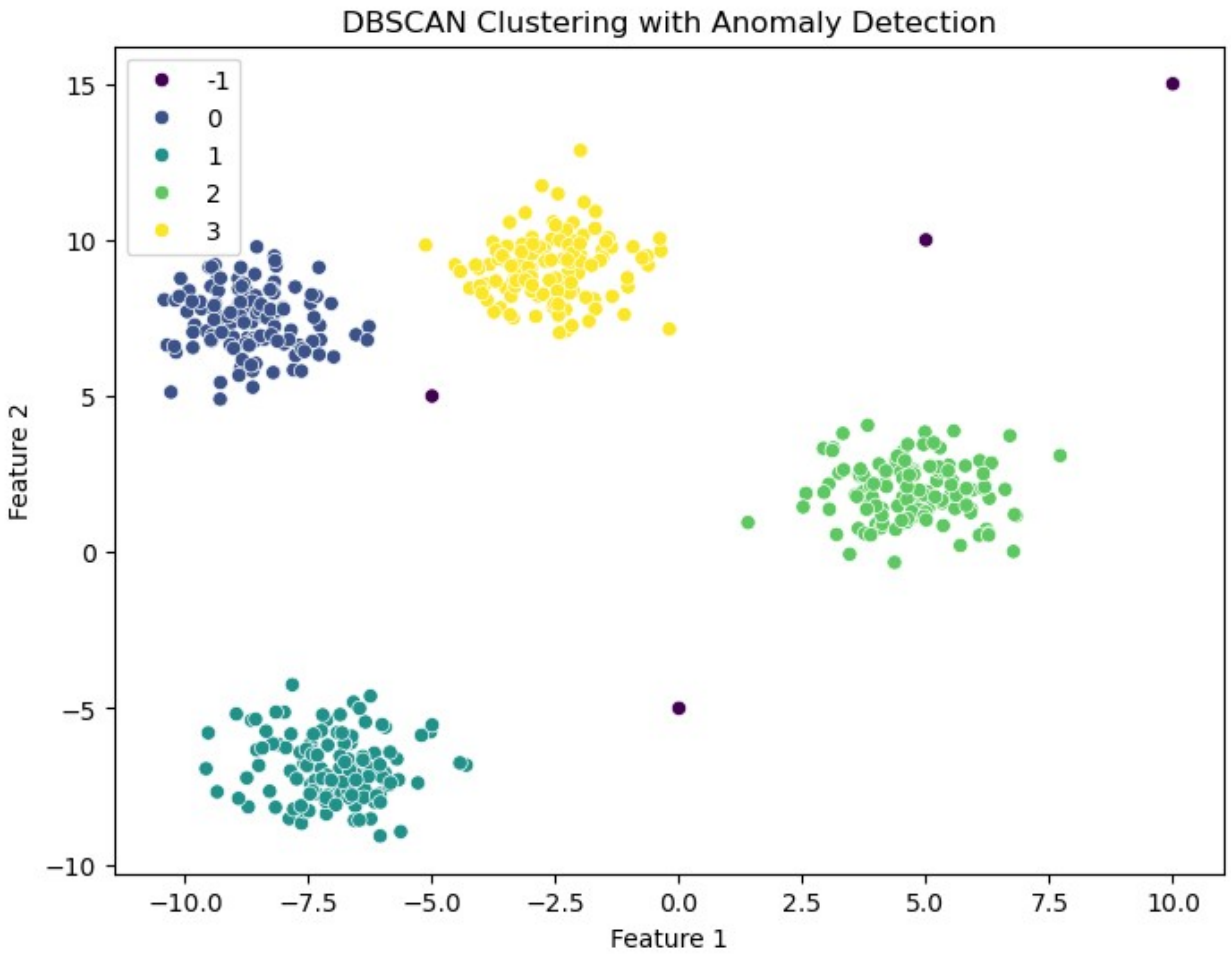
```
pred_anomalies=X[labels==-1]
pred_anomalies
```

```
array([[10., 15.],
       [-5.,  5.],
       [ 0., -5.],
       [ 5., 10.]])
```

visualization of anomalies found by DBSCAN model

```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(8, 6))
sns.scatterplot(x=X[:, 0], y=X[:, 1], hue=labels, palette="viridis",
legend="full")
plt.title("DBSCAN Clustering with Anomaly Detection")
plt.xlabel("Feature 1")
plt.ylabel("Feature 2")
plt.show()
```



Merits of Anomaly Detection Using DBSCAN

1. **Density-Based Approach**: DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is highly effective for identifying anomalies as outliers in regions with low data density, making it well-suited for detecting unusual patterns in datasets with irregular distributions.

2. **No Need for Predefined Number of Clusters**: Unlike methods like K-means, DBSCAN does not require the number of clusters to be specified beforehand, allowing for flexible anomaly detection without prior knowledge of the dataset.

3. **Detection of Arbitrarily Shaped Clusters**: DBSCAN can detect clusters of arbitrary shape, which is an advantage in datasets where anomalies occur in irregular regions that are not easily captured by traditional clustering algorithms.

4. **Robust to Noise**: The algorithm naturally classifies points in sparse regions as noise, effectively identifying outliers (anomalies) without requiring separate training or labeling of outliers.

Demerits of Anomaly Detection Using DBSCAN

1. **Sensitive to Hyperparameters**: DBSCAN relies on two important hyperparameters: `eps` (the maximum distance between two samples for them to be considered part of the same cluster) and `min_samples` (the minimum number of points required to form a dense region). Choosing these values incorrectly can lead to poor detection of anomalies or false positives.

2. **Scalability Issues with High Dimensional Data**: DBSCAN struggles with high-dimensional data as the density concept breaks down in high dimensions (the curse of dimensionality). This limits its performance on complex, large-scale datasets.

3. **Difficulty Handling Varying Densities**: DBSCAN performs less effectively when the dataset contains clusters of varying densities. It may fail to separate dense clusters from noise or anomalies in such cases.

4. **Non-Deterministic in Borderline Cases**: In cases where points lie on the border between dense and sparse regions, DBSCAN's classification may be non-deterministic, leading to instability in detecting certain anomalies.

Conclusion

The implementation of an anomaly detection model using DBSCAN proved effective for identifying unusual patterns in low-dimensional datasets where density-based separation of normal points and outliers is feasible. DBSCAN's ability to detect arbitrary-shaped clusters without specifying the number of clusters provides flexibility and robustness in certain anomaly detection tasks. However, the sensitivity of DBSCAN to the choice of hyperparameters and its challenges in handling high-dimensional data or clusters of varying densities limit its effectiveness in more complex scenarios.

In conclusion, DBSCAN is a powerful tool for anomaly detection in specific contexts where data is well-structured with distinct densities, but its limitations must be carefully considered in large-scale or high-dimensional applications.