

```
In [1]: import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import DBSCAN
from scipy.spatial.distance import cdist
```

```
In [2]: X=np.array([[5,7],
                    [8,4],
                    [3,3],
                    [4,4],
                    [3,7],
                    [6,7],
                    [6,1],
                    [5,5]])
```

```
In [3]: eps=3.5
min_samples = 3
dbscan = DBSCAN(eps=eps,min_samples=min_samples,metric='euclidean')
labels=dbscan.fit_predict(X)
```

```
In [4]: core_samples_mask = np.zeros_like(labels, dtype=bool)
core_samples_mask[dbscan.core_sample_indices_] = True

core_points = X[core_samples_mask]
border_points = X[np.logical_and(~core_samples_mask, labels != -1)]
noise_points = X[labels == -1]
```

```
In [5]: print("Core Points:")
print("Data Point\tCore Points")
for i, x in enumerate(X):
    core_labels = [j for j, core in enumerate(core_points) if np.array_equal(x, core)]
    core_labels_str = ', '.join([f'S{j+1}' for j in core_labels])
    print(f'S{i+1}\t\t{core_labels_str}')
print()
```

Core Points:

Data Point	Core Points
S1	S1
S2	
S3	S2
S4	S3
S5	S4
S6	S5
S7	
S8	S6

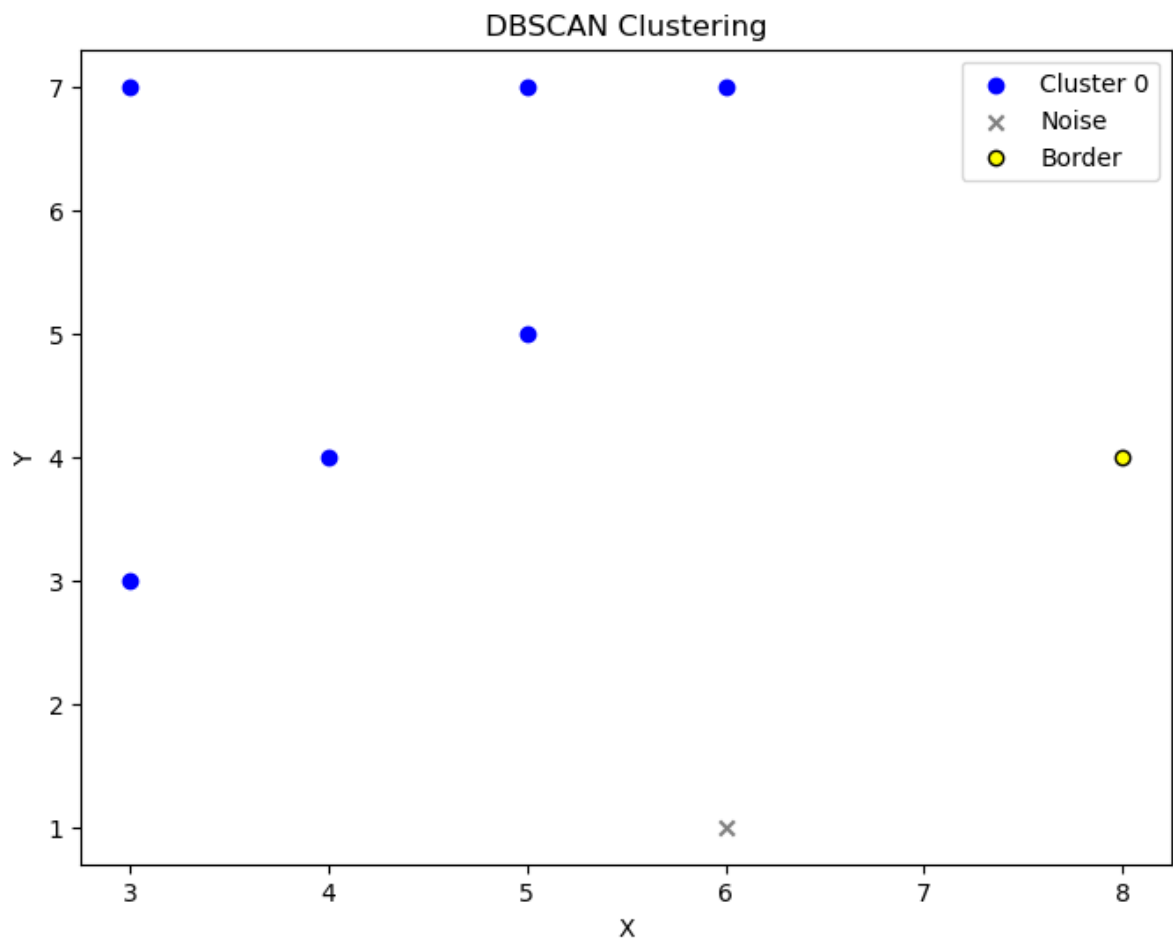
```

In [6]: plt.figure(figsize=(8, 6))
colors = ['blue', 'red', 'green', 'purple']
markers = ['o', 's', '^']
for label in set(labels):
    if label == -1:
        plt.scatter(noise_points[:, 0], noise_points[:, 1], color='gray', marker='x')
    elif label == 0:
        plt.scatter(core_points[:, 0], core_points[:, 1], color=colors[label],
                    label=f'Cluster {label}')
    else:
        cluster_points = X[labels == label]
        plt.scatter(cluster_points[:, 0], cluster_points[:, 1], color=colors[label],
                    marker=markers[label % len(markers)], label=f'Cluster {label}')

if len(border_points) > 0:
    plt.scatter(border_points[:, 0], border_points[:, 1], color='yellow', marker='o',
                linewidths=1, label='Border')

plt.xlabel('X')
plt.ylabel('Y')
plt.title('DBSCAN Clustering')
plt.legend()
plt.show()

```



In [ ]: