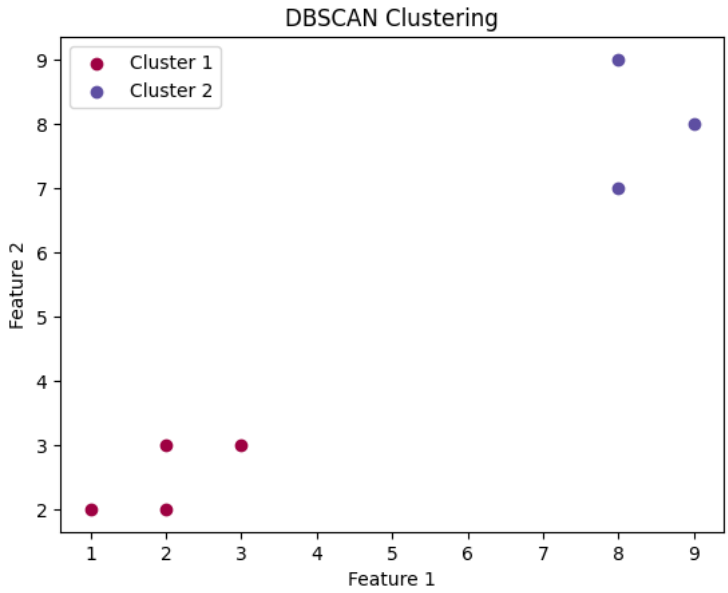
```
import numpy as np
import matplotlib.pyplot as plt

def dbscan(X, eps, min_samples):
    n_samples = X.shape[0]
    labels = np.zeros(n_samples, dtype=int)
    cluster_id = 0

    def region_query(p):
        return np.linalg.norm(X - X[p], axis=1) < eps

    def expand_cluster(p, neighbors):
        nonlocal cluster_id
        labels[p] = cluster_id

        while neighbors.size > 0:
            q_idx = neighbors[0]
            neighbors = neighbors[1:]

            if labels[q_idx] == 0:
                labels[q_idx] = cluster_id
                q_neighbors = region_query(q_idx)
                if np.sum(q_neighbors) >= min_samples:
                    neighbors = np.concatenate((neighbors, np.where(q_neighbors)[0]))

    for p in range(n_samples):
        if labels[p] != 0:
            continue

        neighbors = np.where(region_query(p))[0]
        if len(neighbors) < min_samples:
            labels[p] = -1  # Noise point
        else:
            cluster_id += 1
            expand_cluster(p, neighbors)
```

Automatic saving failed. This file was updated remotely or in another tab. Show diff

```
# Example usage:
X = np.array([[1, 2], [2, 3], [2, 2], [3, 3], [8, 7], [9, 8], [8, 9]])
eps = 1.5  # Maximum distance between points to be considered in the same neighborhood
min_samples = 2  # Minimum number of points in a neighborhood to form a cluster
labels = dbscan(X, eps, min_samples)

# Plot the clusters
unique_labels = np.unique(labels)
colors = plt.cm.Spectral(np.linspace(0, 1, len(unique_labels)))
for label, color in zip(unique_labels, colors):
    if label == -1:
        # Plot noise points in black
        plt.scatter(X[labels == label][:, 0], X[labels == label][:, 1], color='black', marker='x', label='Noise')
    else:
        # Plot clustered points with different colors
        plt.scatter(X[labels == label][:, 0], X[labels == label][:, 1], color=color, label=f'Cluster {label}')

plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.legend()
plt.title('DBSCAN Clustering')
plt.show()
```



✓ 0s completed at 6:43 PM