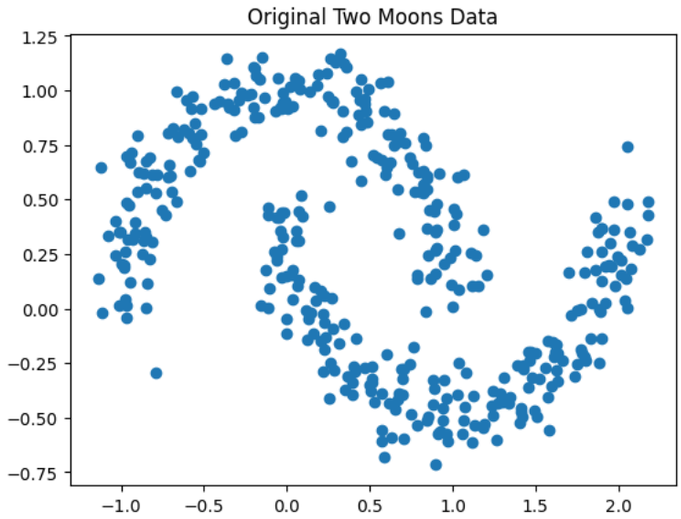
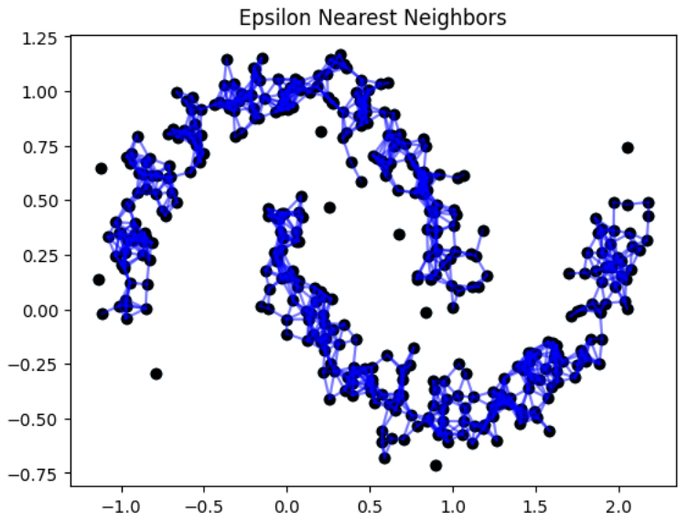
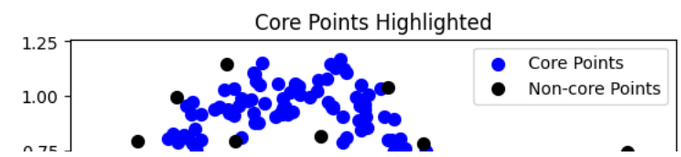
```
# Importing necessary libraries
import matplotlib.pyplot as plt
from sklearn.datasets import make_moons
import numpy as np
from sklearn.cluster import DBSCAN
%matplotlib inline

# Play around with these two parameters and see how they affect the final clustering
eps_val = 0.13
m_val = 5


plt.figure()
# Step 1: Create the dataset
X, y_true = make_moons(n_samples=400, noise=0.1)

# Plot the dataset
plt.scatter(X[:, 0], X[:, 1])
plt.title('Original Two Moons Data')
plt.show()

plt.figure()
# Plot the dataset
plt.scatter(X[:, 0], X[:, 1])

from sklearn.neighbors import KDTree

# Create the KDTree
kdt = KDTree(X, metric='euclidean')

ind = kdt.query_radius(X, r=eps_val, return_distance=False)

# Step 2: Build a nearest neighbor graph

# Get the indices of the points within eps distance, excluding the point itself
plt.scatter(X[:, 0], X[:, 1], color='black')
plt.title('Epsilon Nearest Neighbors')
# Connect each point to its neighbors
for i in range(X.shape[0]):
    for j in ind[i]:
        if i != j:
            plt.plot([X[i, 0], X[j, 0]], [X[i, 1], X[j, 1]], 'b-', alpha=0.3)
plt.show()

plt.show()

plt.figure()

dbscan = DBSCAN(eps=eps_val, min_samples=m_val)

dbscan.fit(X)

# Step 3: Highlight points with >m points
core_samples_mask = np.zeros_like(dbscan.labels_, dtype=bool)
core_samples_mask[dbscan.core_sample_indices_] = True

plt.scatter(X[core_samples_mask, 0], X[core_samples_mask, 1], s=50, color='blue', label='Core Points')
plt.scatter(X[~core_samples_mask, 0], X[~core_samples_mask, 1], s=50, color='black', label='Non-core Points')
plt.title('Core Points Highlighted')
plt.legend()
plt.show()
plt.figure()

# Step 4: Finish the DBSCAN algorithm and highlight the clusters and noise points
labels = dbscan.labels_

# Number of clusters in labels, ignoring noise if present
n_clusters_ = len(set(labels)) - (1 if -1 in labels else 0)
n_noise_ = list(labels).count(-1)

# Plot result
unique_labels = set(labels)
colors = [plt.cm.Spectral(each) for each in np.linspace(0, 1, len(unique_labels))]

for k, col in zip(unique_labels, colors):
    if k == -1:
        # Black used for noise
        col = [0, 0, 0, 1]

    class_member_mask = (labels == k)

    xy = X[class_member_mask & core_samples_mask]
    plt.plot(xy[:, 0], xy[:, 1], 'o', markerfacecolor=tuple(col), markeredgecolor='k', markersize=14)

    xy = X[class_member_mask & ~core_samples_mask]
    plt.plot(xy[:, 0], xy[:, 1], 'o', markerfacecolor=tuple(col), markeredgecolor='k', markersize=6)

plt.title('DBSCAN Algorithm Result. Estimated number of clusters: %d' % n_clusters_)
plt.show()
```


Original Two Moons Data
1.25
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0


Epsilon Nearest Neighbors
1.25
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0


Core Points Highlighted
1.25
1.00
Core Points
Non-core Points