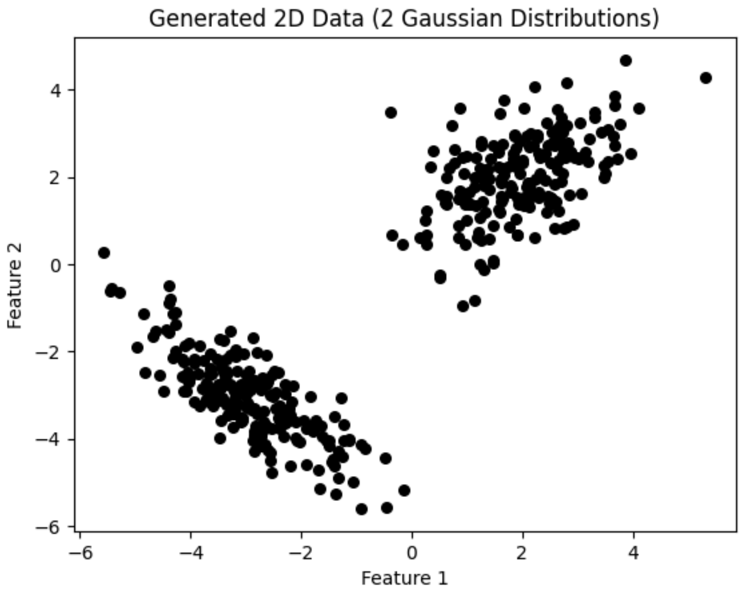
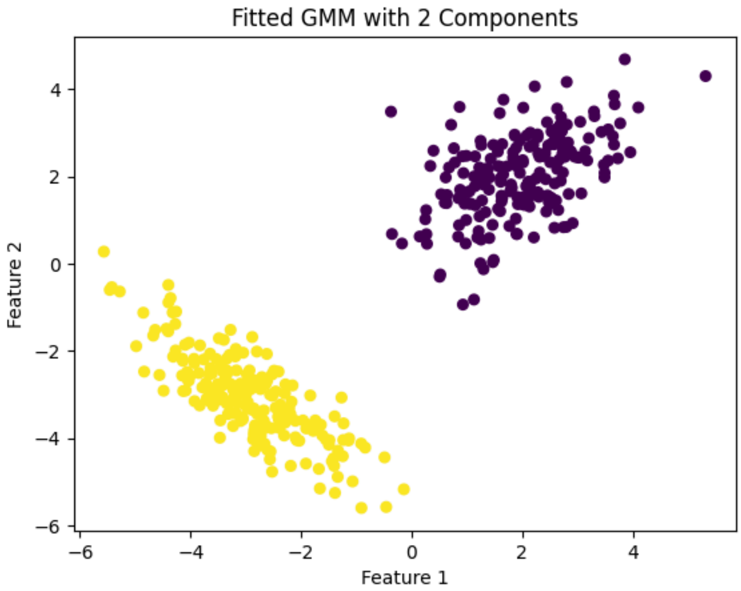
```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.mixture import GaussianMixture

np.random.seed(42)
# Gaussian 1 (centered at (2, 2))
gaussian1 = np.random.multivariate_normal([2, 2], [[1, 0.5], [0.5, 1]], 200)
# Gaussian 2 (centered at (-3, -3))
gaussian2 = np.random.multivariate_normal([-3, -3], [[1, -0.8], [-0.8, 1]], 200)
# Combine the two Gaussian datasets
X = np.vstack([gaussian1, gaussian2])

plt.scatter(X[:, 0], X[:, 1], c='black', s=30, marker='o')
plt.title("Generated 2D Data (2 Gaussian Distributions)")
plt.xlabel("Feature 1")
plt.ylabel("Feature 2")
plt.show()

gmm = GaussianMixture(n_components=2, covariance_type='full', random_state=42)
gmm.fit(X)

labels = gmm.predict(X)

plt.scatter(X[:, 0], X[:, 1], c=labels, cmap='viridis', s=30, marker='o')
plt.title("Fitted GMM with 2 Components")
plt.xlabel("Feature 1")
plt.ylabel("Feature 2")
plt.show()

# Step 6: Show the estimated parameters (means and covariances)
print("Means of the GMM components:\n", gmm.means_)
print("Covariances of the GMM components:\n", gmm.covariances_)
```

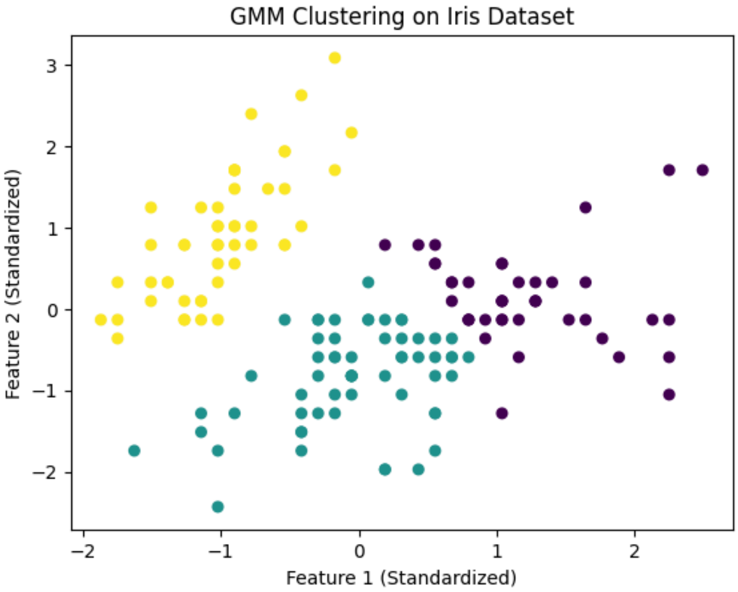




```
Means of the GMM components:
 [[ 1.97512889  2.01388373]
 [-2.94739559 -3.06267896]]
Covariances of the GMM components:
 [[[ 0.89373728  0.45299025]
  [ 0.45299025  0.93868511]]

 [[ 1.02989977 -0.81518413]
  [-0.81518413  1.00195099]]]
```

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.mixture import GaussianMixture
from sklearn import datasets
from sklearn.preprocessing import StandardScaler

# Step 1: Load the Iris dataset
iris = datasets.load_iris()
X = iris.data[:, :2]  # Using only the first two features for visualization

# Step 2: Standardize the dataset
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Step 3: Fit a Gaussian Mixture Model with 3 components (since Iris has 3 classes)
gmm = GaussianMixture(n_components=3, covariance_type='full', random_state=42)
gmm.fit(X_scaled)

# Step 4: Predict the cluster membership (labels)
labels = gmm.predict(X_scaled)

# Step 5: Visualize the GMM result
plt.scatter(X_scaled[:, 0], X_scaled[:, 1], c=labels, cmap='viridis', s=30, marker='o')
plt.title("GMM Clustering on Iris Dataset")
plt.xlabel("Feature 1 (Standardized)")
plt.ylabel("Feature 2 (Standardized)")
plt.show()
```

```
# Step 6: Show the estimated parameters (means and covariances)
print("Means of the GMM components:\n", gmm.means_)
print("Covariances of the GMM components:\n", gmm.covariances_)
```



```
Means of the GMM components:
 [[ 1.01009763 -0.03099225]
 [ 0.05876606 -0.75803909]
 [-1.0038089   0.90771236]]
Covariances of the GMM components:
 [[[0.52745679 0.12205945]
  [0.12205945 0.43834273]]

 [[0.40115983 0.23224156]
  [0.23224156 0.48141623]]

 [[0.17537594 0.24664977]
  [0.24664977 0.63045924]]]
```

```
#end of code
```