Chapter 15: Demo GMM

In [1]:
```
# from google.colab import drive
# drive.mount("/content/gdrive", force_remount=True)
```

In [2]:
```
# %cd '/content/gdrive/My Drive/LDS6_MachineLearning/practice/Chapter15_GMM/'
```

In [3]:
```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.mixture import GaussianMixture
```

In [4]:
```
X_train = np.load('data.npy')
```

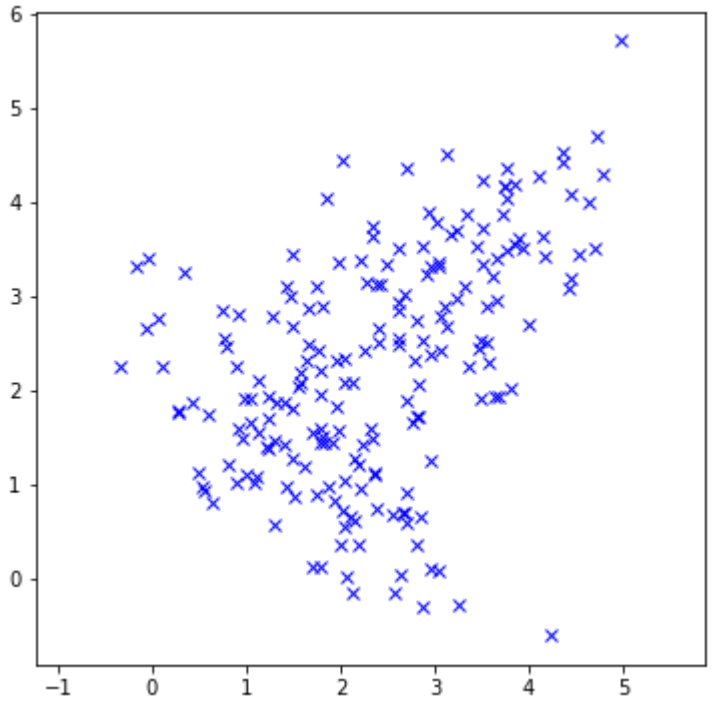
In [5]:
```
X_train[0]
```

Out[5]:
```
array([2.61755903, 2.49389168])
```

In [6]:
```
X_train.size
```

Out[6]:
```
400
```

In [7]:
```
# vẽ hình
plt.figure(figsize=(6,6))
plt.plot(X_train[:,0], X_train[:,1], 'bx')
plt.axis('equal')
plt.show()
```



In [8]:
```
# Quan sát ta thấy dữ liệu không phân cụm rõ ràng
# => Dùng GMM để thực hiện việc phân cụm
```

In [9]:
```
# Select K components
# https://scikit-learn.org/stable/modules/clustering.html#silhouette-coefficient
from sklearn import metrics
list_sil = [] # chua danh sach cac gia tri sil
K = range(2,8) # chua danh sach cac k
for k in K:
    gmm = GaussianMixture(n_components=k) # 2, 3, 4...
    gmm.fit(X_train)
    labels = gmm.predict(X_train)
    # k = 2 => 0,1
    # k = 3 => 0, 1, 2
    sil = metrics.silhouette_score(X_train, labels, metric='euclidean')
    list_sil.append(sil)
```

In [10]:
```
# Plot
plt.plot(K, list_sil, 'bx-')
plt.xlabel('k')
plt.ylabel('sil_score')
plt.title('The silhouette_score & k')
plt.show()
```

In [11]:
```
# select components = 2
gmm = GaussianMixture(n_components=2)
gmm.fit(X_train)
```

Out[11]:
```
GaussianMixture(covariance_type='full', init_params='kmeans', max_iter=100,
                means_init=None, n_components=2, n_init=1, precisions_init=Non
e,
                random_state=None, reg_covar=1e-06, tol=0.001, verbose=0,
                verbose_interval=10, warm_start=False, weights_init=None)
```

Sau khi model đã hội tụ, weights, means, và covariances cần phải được giải quyết. In các thông số này:

In [12]:
```
print(gmm.weights_)
```

```
[0.52279335 0.47720665]
```

In [13]:
```
print(gmm.means_)
```

```
[[3.0363831  3.09828041]
 [1.60629419 1.3470999 ]]
```

In [14]:
```
print(gmm.covariances_)
```

```
[[[ 0.8465178   0.38644336]
  [ 0.38644336  0.73395863]]

 [[ 0.75275611 -0.5054196 ]
  [-0.5054196   0.74286061]]]
```

In [15]:
```
types = gmm.predict(X_train)
# Mẫu thuộc nhóm nào? 0, 1
# X[x,y], loai =>
```

In [16]:
```
# plot mixture of Gaussians
plt.figure(figsize=(6,6))
X, Y = np.meshgrid(np.linspace(-1, 6), np.linspace(-1,6))
XX = np.array([X.ravel(), Y.ravel()]).T # ma tran [[x,y]] ~ X_train
Z = gmm.score_samples(XX)
Z = Z.reshape((50,50))

plt.contour(X, Y, Z)
plt.scatter(X_train[:, 0], X_train[:, 1], c=types) # 0, 1
plt.scatter(gmm.means_[:,0], gmm.means_[:,1], color="red")
plt.show()
```