Chapter 15 - Exercise 1: Iris (2 features)

Cho dữ liệu iris.xls

1. Chuẩn hóa dữ liệu X chứa petalwidth, petallength
2. Tìm số cụm phù hợp k?
3. Áp dụng thuật toán GMM để giải bài toán phân cụm với số cụm đã tìm được ở câu 2.
4. Cho X_now = np.array([[0.4, 1.5], [1.6, 4.5], [2, 5.7]]), cho biết phần tử này thuộc cụm nào?
5. Vẽ hình, xem kết quả

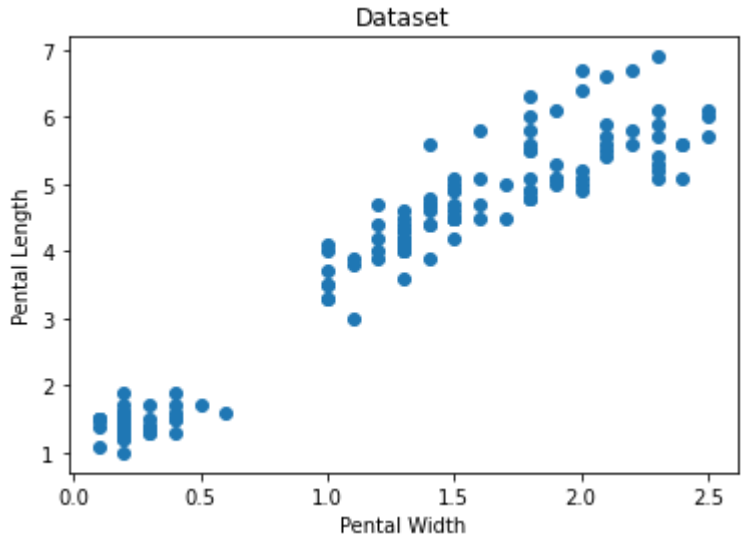
In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.mixture import GaussianMixture
```

In [2]:
```
data = pd.read_excel('iris.xls')
```

In [3]:
```
# data.info()
```

In [4]:
```
# data.head()
```

In [5]:
```
plt.scatter(data.petalwidth, data.petallength)
plt.title('Dataset')
plt.ylabel("Pental Length")
plt.xlabel("Pental Width")
plt.show()
```



In [6]:
```
X_train = data[['petalwidth', 'petallength']]
```

In [7]:
```
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 [8]:
```
# Plot
plt.plot(K, list_sil, 'bx-')
plt.xlabel('k')
plt.ylabel('sil_score')
plt.title('The silhouette_score & k')
plt.show()
```

In [9]:
```
# Select k = 2
```

In [10]:
```
gmm = GaussianMixture(n_components=2)
gmm.fit(X_train)
```

Out[10]:
```
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 [11]:
```
print(gmm.weights_)
```

```
[0.66684538 0.33315462]
```

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

```
[[1.67565066 4.9051612 ]
 [0.24393108 1.46383257]]
```

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

```
[[[0.17903929 0.28685725]
  [0.28685725 0.67721583]]

 [[0.01124836 0.00557197]
  [0.00557197 0.02945446]]]
```

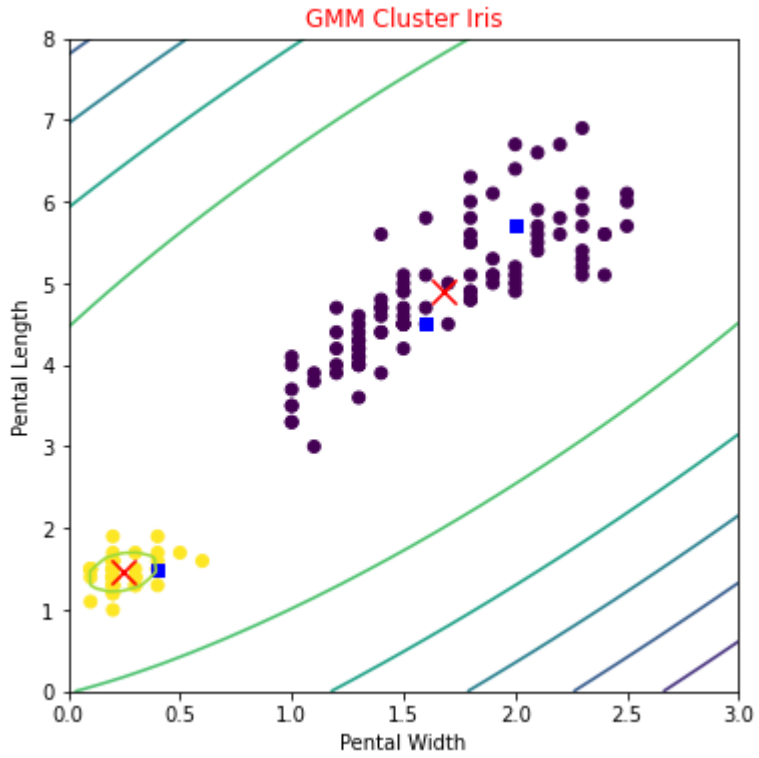
In [14]:
```
types = gmm.predict(X_train) # 0, 1, 2
```

In [15]:
```
X_now = np.array([[0.4, 1.5], [1.6, 4.5], [2, 5.7]])
y_now = gmm.predict(X_now)
y_now
```

Out[15]:
```
array([1, 0, 0], dtype=int64)
```

In [16]:
```
# plot mixture of Gaussians
plt.figure(figsize=(6,6))
X, Y = np.meshgrid(np.linspace(0, 3), np.linspace(0,8))
XX = np.array([X.ravel(), Y.ravel()]).T
Z = gmm.score_samples(XX)
Z = Z.reshape((50,50))

plt.contour(X, Y, Z)
plt.scatter(X_train['petalwidth'], X_train['petallength'], c=types)
plt.scatter(X_now[:,0], X_now[:,1], marker="s", c='b')
plt.scatter(gmm.means_[:,0], gmm.means_[:,1], color="red", marker='x', s=150)
plt.xlabel("Pental Width")
plt.ylabel("Pental Length")
plt.title("GMM Cluster Iris", color="red")
plt.show()
```



In [18]:
```
# Sau khi thực hiện với 2 thuộc tính
# => Thử thực hiện bài toán này với cả 4 thuộc tính của IRIS
```