# Chapter 11 - exercise 2: Iris

## Cho dữ liệu iris.xls

1. Chuẩn hóa dữ liệu
2. Áp dụng Elbow tìm k
3. Áp dụng thuật toán K-Means để giải bài toán phân cụm theo K
4. So sánh giữa kết quả phân cụm với kết quả hiện có.
5. Cho X_test = np.array([[4.7, 3.2, 1.5, 0.4], [4.8, 3.5, 4.5,1.6], [6.1, 3.5, 5.7,2]]), cho biết những phần tử này thuộc cụm nào?
6. Vẽ hình, xem kết quả. Nhận xét kết quả.

Entrée [1]:
```
from sklearn.cluster import KMeans
from sklearn import metrics
from scipy.spatial.distance import cdist
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

Entrée [2]:
```
iris = pd.read_excel("Iris.xls")
iris.shape
```

Out[2]:
```
(150, 5)
```

Entrée [3]:
```
iris.head(3)
```

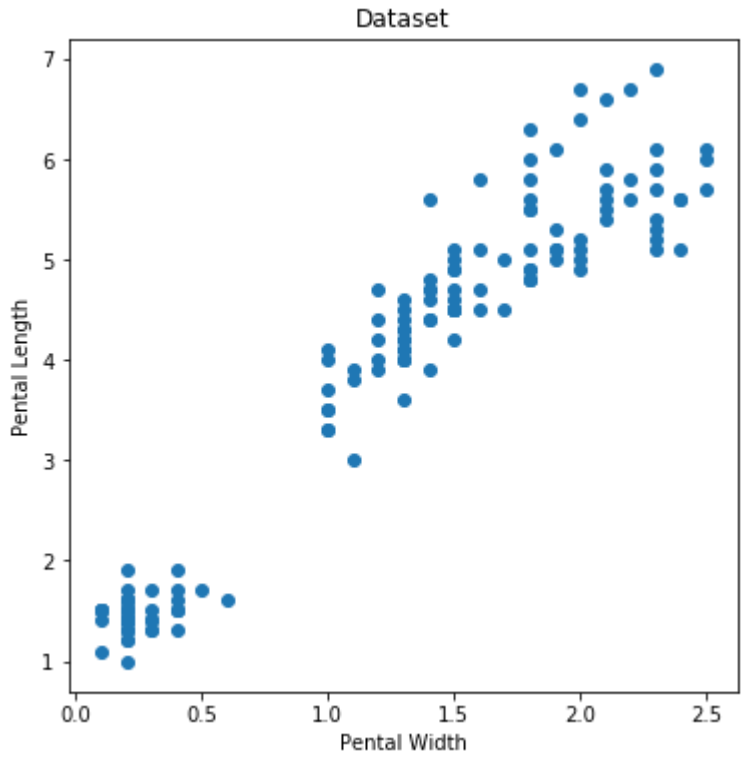
Out[3]:

| | sepallength | sepalwidth | petallength | petalwidth | iris |
|---|---|---|---|---|---|
| **0** | 5.1 | 3.5 | 1.4 | 0.2 | Iris-setosa |
| **1** | 4.9 | 3.0 | 1.4 | 0.2 | Iris-setosa |
| **2** | 4.7 | 3.2 | 1.3 | 0.2 | Iris-setosa |

Entrée [4]:
```
iris.groupby('iris').petallength.count()
```

Out[4]:
```
iris
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
Name: petallength, dtype: int64
```

Entrée [5]:
```
plt.figure(figsize=(6,6))
plt.scatter(iris.petalwidth, iris.petallength)
plt.title('Dataset')
plt.xlabel("Pental Width")
plt.ylabel("Pental Length")
plt.show()
```



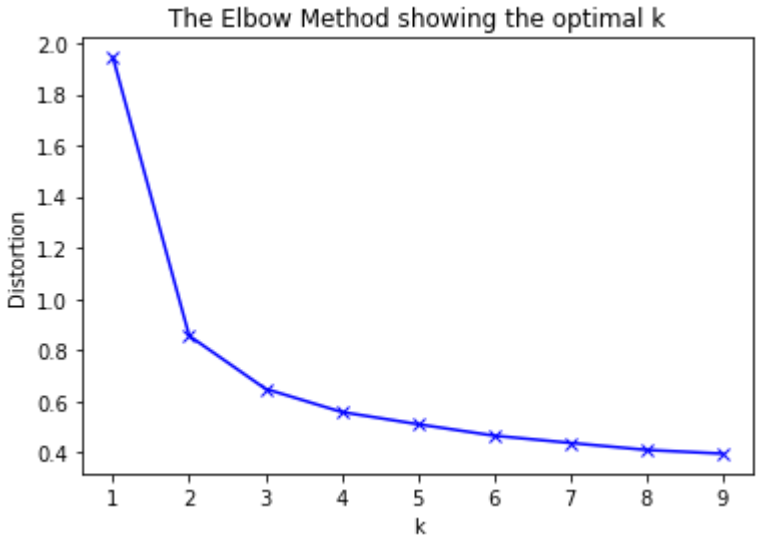

Entrée [6]:
```
X = iris.drop('iris', axis=1)
X.head(3)
```

Out[6]:

| | sepallength | sepalwidth | petallength | petalwidth |
|---|---|---|---|---|
| **0** | 5.1 | 3.5 | 1.4 | 0.2 |
| **1** | 4.9 | 3.0 | 1.4 | 0.2 |
| **2** | 4.7 | 3.2 | 1.3 | 0.2 |

Entrée [7]:
```
# k means determine k
distortions = []
K = range(1,10)
for k in K:
    kmeanModel = KMeans(n_clusters=k).fit(X)
    kmeanModel.fit(X)
    distortions.append(sum(np.min(cdist(X, kmeanModel.cluster_centers_,
                                        'euclidean'), axis=1)) / X.shape[0])

# Plot the elbow
plt.plot(K, distortions, 'bx-')
plt.xlabel('k')
plt.ylabel('Distortion')
plt.title('The Elbow Method showing the optimal k')
plt.show()
```




Entrée [8]:
```
# Chọn k = 3
kmeans = KMeans(n_clusters=3)
kmeans.fit(X)

centroids = kmeans.cluster_centers_
labels = kmeans.labels_

print(centroids)
print(labels)
```

```
[[5.006      3.418      1.464      0.244     ]
 [5.9016129  2.7483871  4.39354839 1.43387097]
 [6.85       3.07368421 5.74210526 2.07105263]]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 2 2 2 1 2 2 2 2
 2 2 1 1 2 2 2 2 1 2 1 2 1 2 2 1 1 2 2 2 2 1 2 2 2 1 2 2 2 1 2 2 2 1 2
 2 1]
```

Entrée [9]:
```
X['iris'] = pd.Series(labels)
```

Entrée [10]:
```
X.head(3)
```

Out[10]:

| | sepallength | sepalwidth | petallength | petalwidth | iris |
|---|---|---|---|---|---|
| **0** | 5.1 | 3.5 | 1.4 | 0.2 | 0 |
| **1** | 4.9 | 3.0 | 1.4 | 0.2 | 0 |
| **2** | 4.7 | 3.2 | 1.3 | 0.2 | 0 |

Entrée [11]:
```
X_test = np.array([[4.7, 3.2, 1.5, 0.4], [4.8, 3.5, 4.5,1.6], [6.1, 3.5, 5.7,2]]
pred = kmeans.predict(X_test)
pred
```

Out[11]:
```
array([0, 1, 2])
```

Entrée [12]:
```
plt.figure(figsize=(8,8))
plt.scatter(centroids[:, 3],centroids[:, 2], marker = "x", s=150, color='r')
plt.scatter(X.petalwidth, X.petallength, c=X.iris)
plt.scatter(X_test[:,3], X_test[:,2], marker="s", c='b')
plt.xlabel("Pental Width")
plt.ylabel("Pental Length")
plt.title("K-Means Cluster Iris", color="red")
plt.show()
```