주제

IRIS 데이터 클러스터링

최연석, 오서영

IRIS DATA

The Iris Dataset

Collected by Ronald Fisher in 1936

데이터 확인

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
import pandas as pd
from sklearn.cluster import AgglomerativeClustering
import mglearn
```

In [2]:
```
dataset = load_iris()
```

In [3]:
```
# 판다스로 데이터 확인하기
labels = pd.DataFrame(dataset.target)
labels.columns=['labels']
data = pd.DataFrame(dataset.data)
data.columns=dataset['feature_names']
data = pd.concat([data,labels],axis=1)

data
```

Out[3]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	labels
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
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	2
146	6.3	2.5	5.0	1.9	2
147	6.5	3.0	5.2	2.0	2
148	6.2	3.4	5.4	2.3	2
149	5.9	3.0	5.1	1.8	2

150 rows × 5 columns

In [6]:
```
x_train = X[:,0:2]
```

In [7]:
```
x_train.shape
```

Out[7]: (150, 2)

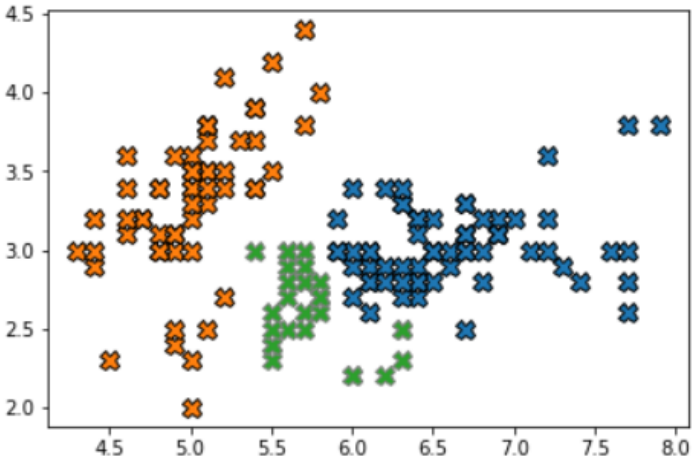
Agglomerative clustering

In [8]:
```
agg_2 = AgglomerativeClustering(n_clusters=2).fit(x_train)
print(agg_2.labels_)

agg_3 = AgglomerativeClustering(n_clusters=3).fit(x_train)
print(agg_3.labels_)

agg_4 = AgglomerativeClustering(n_clusters=4).fit(x_train)
print(agg_4.labels_)

agg_5 = AgglomerativeClustering(n_clusters=5).fit(x_train)
print(agg_5.labels_)

agg_6 = AgglomerativeClustering(n_clusters=6).fit(x_train)
print(agg_6.labels_)

agg_7 = AgglomerativeClustering(n_clusters=7).fit(x_train)
print(agg_7.labels_)

agg_8 = AgglomerativeClustering(n_clusters=8).fit(x_train)
print(agg_8.labels_)
```

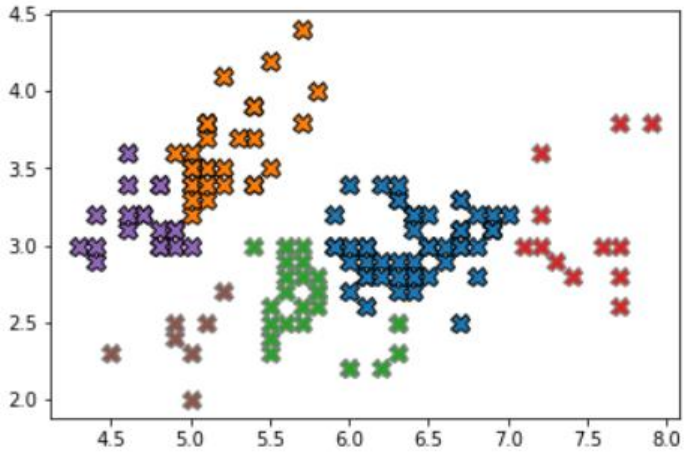
```
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0]
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 2 0 2 0 1 0 1 1 0 2 0 2 0 2 2 2 2 0 0 2 0
 0 0 0 0 0 2 2 2 2 0 2 0 0 2 2 2 2 0 2 1 2 2 2 0 1 2 0 2 0 0 0 0 1 0 0 0 0
 0 0 2 2 0 0 0 0 2 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 2 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 0 0 0 1 1 1 2 1 2 1 0 1 0 0 1 2 1 2 1 2 2 2 2 1 1 2 1
 1 1 1 1 1 2 2 2 2 1 2 1 1 2 2 2 2 1 2 0 2 2 2 1 0 2 1 2 3 1 1 3 0 3 1 3 1
 1 1 2 2 1 1 3 3 2 1 2 3 1 1 3 1 1 1 3 3 3 1 1 1 3 1 1 1 1 1 1 2 1 1 1 2 1
 1 1]
[4 0 0 0 4 4 0 4 0 0 4 0 0 0 4 4 4 4 4 4 4 4 0 4 0 0 4 4 4 0 0 4 4 4 0 4 4
 4 0 4 4 0 0 4 4 0 4 0 4 4 1 1 1 2 1 2 1 0 1 0 0 1 2 1 2 1 2 2 2 2 1 1 2 1
 1 1 1 1 1 2 2 2 2 1 2 1 1 2 2 2 2 1 2 0 2 2 2 1 0 2 1 2 3 1 1 3 0 3 1 3 1
 1 1 2 2 1 1 3 3 2 1 2 3 1 1 3 1 1 1 3 3 3 1 1 1 3 1 1 1 1 1 1 2 1 1 1 2 1
 1 1]
[1 4 4 4 1 1 4 1 4 4 1 4 4 4 1 1 1 1 1 1 1 1 4 1 4 4 1 1 1 4 4 1 1 1 4 1 1
 1 4 1 1 5 4 1 1 4 1 4 1 1 0 0 0 2 0 2 0 5 0 5 5 0 2 0 2 0 2 2 2 2 0 0 2 0
 0 0 0 0 0 2 2 2 2 0 2 0 0 2 2 2 2 0 2 5 2 2 2 0 5 2 0 2 3 0 0 3 5 3 0 3 0
 0 0 2 2 0 0 3 3 2 0 2 3 0 0 3 0 0 0 3 3 3 0 0 0 3 0 0 0 0 0 0 2 0 0 0 2 0
 0 0]
[0 4 4 4 0 0 4 0 4 4 0 4 4 4 0 0 0 0 0 0 0 0 4 0 4 4 0 0 0 4 4 0 0 0 4 0 0
 0 4 0 0 5 4 0 0 4 0 4 0 0 6 3 6 2 3 2 3 5 3 5 5 3 2 3 2 6 2 2 2 2 3 3 2 3
 3 3 3 6 3 2 2 2 2 3 2 3 6 2 2 2 2 3 2 5 2 2 2 3 5 2 3 2 1 3 3 1 5 1 3 1 3
 3 6 2 2 3 3 1 1 2 6 2 1 3 6 1 3 3 3 1 1 1 3 3 3 1 3 3 3 6 6 6 2 6 6 6 2 3
 3 3]
[7 4 4 4 7 2 4 7 4 4 2 4 4 4 2 2 2 7 2 2 7 2 4 7 4 4 7 7 7 4 4 7 2 2 4 7 7
 7 4 7 7 5 4 7 2 4 2 4 2 7 6 3 6 0 3 0 3 5 3 5 5 3 0 3 0 6 0 0 0 0 3 3 0 3
 3 3 3 6 3 0 0 0 0 3 0 3 6 0 0 0 0 3 0 5 0 0 0 3 5 0 3 0 1 3 3 1 5 1 3 1 3
 3 6 0 0 3 3 1 1 0 6 0 1 3 6 1 3 3 3 1 1 1 3 3 3 1 3 3 3 6 6 6 0 6 6 6 0 3
 3 3]
```

그래프 표현

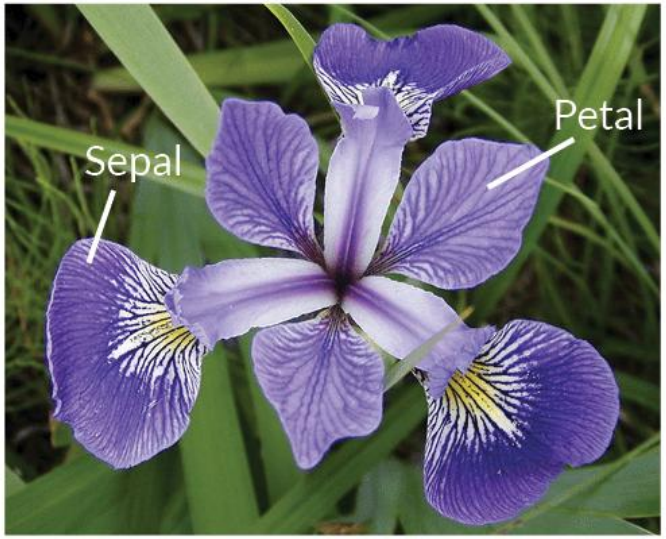
In [9]:
```
mglearn.discrete_scatter(x1 = x_train[:,0], x2 = x_train[:,1], y=agg_3.labels_, markers='X')
plt.show()
```



In [10]:
```
mglearn.discrete_scatter(x1 = x_train[:,0], x2 = x_train[:,1], y=agg_6.labels_, markers='X')
plt.show()
```



IRIS DATA



Iris Versicolor

Iris Setosa

Iris Virginica

실루엣 계수

1. 실루엣 계수는 한 클러스터 안에 데이터들이 다른 클러스터와 비교해서 얼마나 비슷한가를 나타낸다

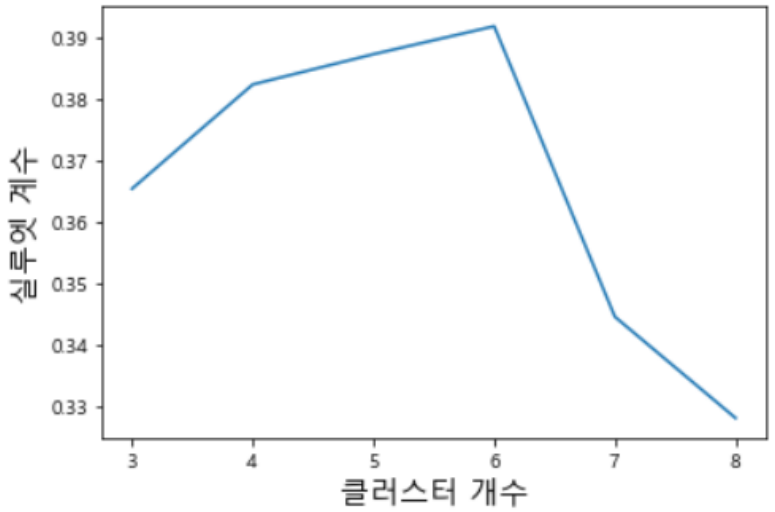
2. **1**에 가까울수록 잘 부합하는 데이터

```
In [11]: from sklearn.metrics.cluster import silhouette_score
         print(silhouette_score(x_train, agg_3.labels_))
         print(silhouette_score(x_train, agg_4.labels_))
         print(silhouette_score(x_train, agg_5.labels_))
         print(silhouette_score(x_train, agg_6.labels_))
         print(silhouette_score(x_train, agg_7.labels_))
         print(silhouette_score(x_train, agg_8.labels_))

         0.3653346819163389
         0.38231594211850395
         0.38724618388871157
         0.3918000357829499
         0.34449589365226896
         0.32799364657656743
```

실루엣 계수

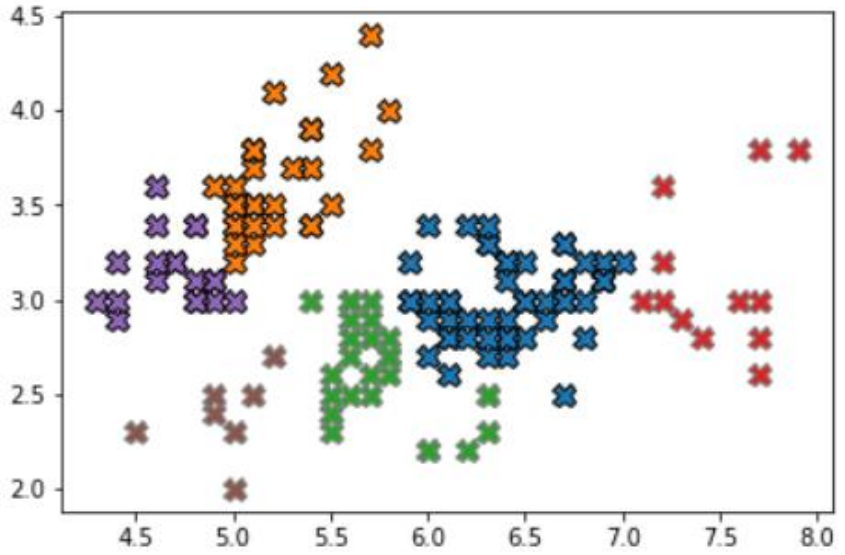
In [12]:
```
from matplotlib import font_manager, rc
font_name = font_manager.FontProperties(fname="c:/Windows/Fonts/malgun.ttf").get_name()
rc('font', family=font_name)
```

In [13]:
```
sil = []
num = np.arange(3,9)
for i in range(3,9):
    agg = AgglomerativeClustering(n_clusters=i).fit(x_train)
    sil.append(silhouette_score(x_train, agg.labels_))

plt.plot(num, sil)
plt.xlabel("클러스터 개수", fontsize = 15)
plt.ylabel("실루엣 계수", fontsize = 15)
plt.show()
```



결과 시각화

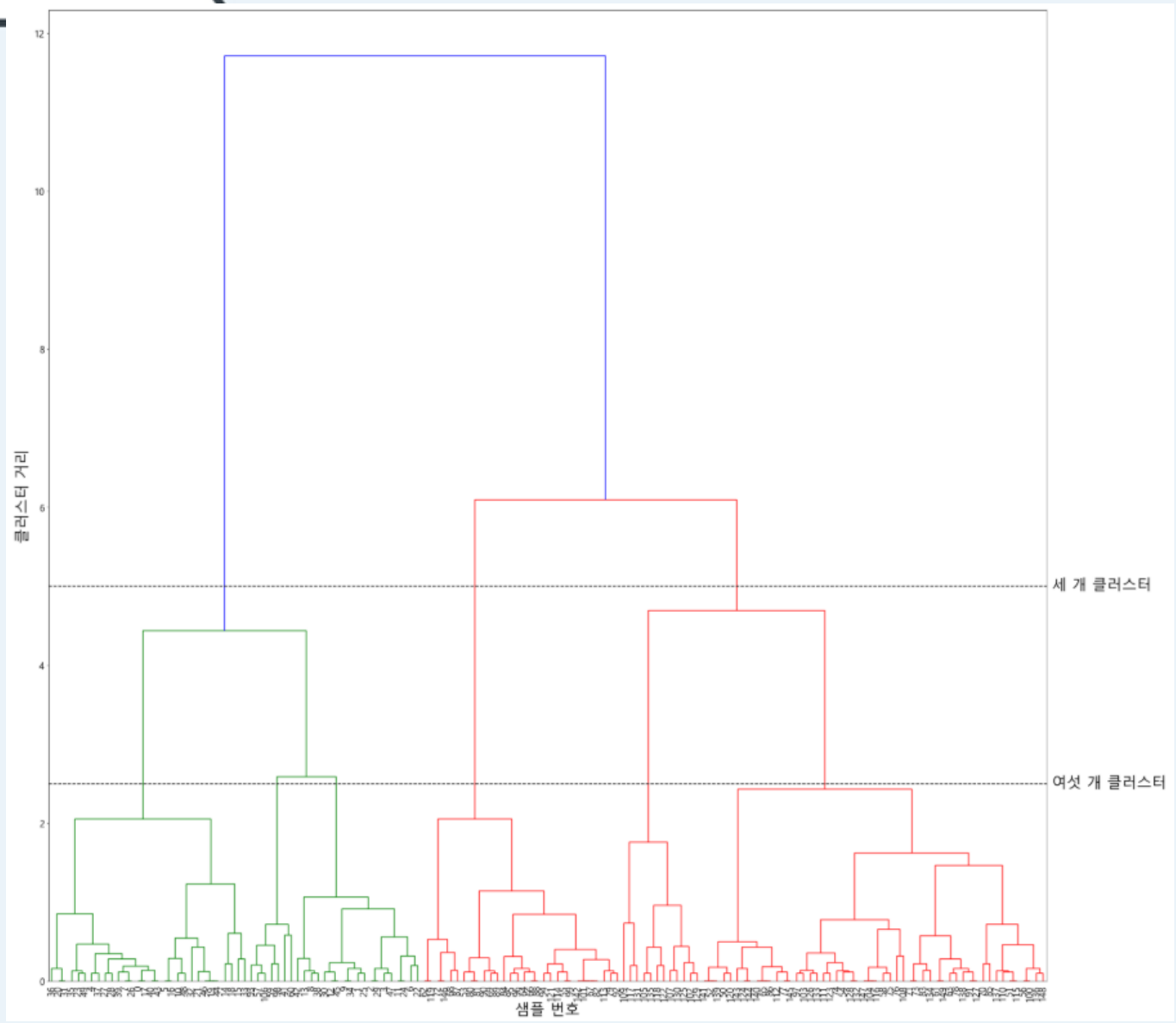
In [10]:
```
mglearn.discrete_scatter(x1 = x_train[:,0], x2 = x_train[:,1], y=agg_6.labels_, markers='X')
plt.show()
```



Cluster Dendrogram

In [15]:
```
from scipy.cluster.hierarchy import dendrogram, ward
plt.rcParams["figure.figsize"] = (30,30)
linkage_array = ward(x_train)
dendrogram(linkage_array)

ax = plt.gca()
bounds = ax.get_xbound()
ax.plot(bounds, [2.5, 2.5], '--', c='k')
ax.plot(bounds, [5, 5], '--', c='k')

ax.text(bounds[1], 2.5, ' 여섯 개 클러스터', va='center', fontdict={'size': 25})
ax.text(bounds[1], 5, ' 세 개 클러스터', va='center', fontdict={'size': 25})
plt.xlabel("샘플 번호", fontsize = 25)
plt.ylabel("클러스터 거리", fontsize = 25)
plt.xticks(fontsize=16)
plt.yticks(fontsize=16)
plt.show()
```

Cluster Dendrogram



감사합니다!