

클러스터링1

```
import matplotlib.pyplot as plt
from sklearn import cluster
from sklearn import datasets

# iris 데이터를 로드
iris = datasets.load_iris()
data = iris["data"]

# 학습 → 클러스터 생성
model = cluster.KMeans(n_clusters=3)
#n_clusters =3 : 클러스터 개수 지정
model.fit(data)

# 학습 결과의 라벨 취득
labels = model.labels_

### 그래프 그리기
x_index = 2
y_index = 3

data_x=data[:,x_index]
data_y=data[:,y_index]

x_max = 7.5
x_min = 0
y_max = 3
y_min = 0
x_label = iris["feature_names"][x_index]
y_label = iris["feature_names"][y_index]

plt.scatter(data_x[labels==0], data_y[labels==0],c='black' ,alpha=0.3,s=100,
marker="o",label="cluster 0")
plt.scatter(data_x[labels==1], data_y[labels==1],c='black' ,alpha=0.3,s=100,
marker="o",label="cluster 1")
plt.scatter(data_x[labels==2], data_y[labels==2],c='black' ,alpha=0.3,s=100,
marker="o",label="cluster 2")
plt.xlim(x_min, x_max)
```

```
plt.ylim(y_min, y_max)
plt.xlabel(x_label,fontsize='large')
plt.ylabel(y_label,fontsize='large')
plt.show()
```

클러스터링 2

```
import matplotlib.pyplot as plt
from sklearn import cluster
from sklearn import datasets

# iris 데이터를 로드
iris = datasets.load_iris()
data = iris['data']

# 학습 → 클러스터 생성
model = cluster.KMeans(n_clusters=3)
model.fit(data)

# 학습 결과의 라벨 취득
labels = model.labels_

# 그래프 그리기
ldata = data[labels == 0]
plt.scatter(ldata[:, 2], ldata[:, 3],
            c='black' ,alpha=0.3,s=100 ,marker="o")

ldata = data[labels == 1]
plt.scatter(ldata[:, 2], ldata[:, 3],
            c='black' ,alpha=0.3,s=100 ,marker="^")

ldata = data[labels == 2]
plt.scatter(ldata[:, 2], ldata[:, 3],
            c='black' ,alpha=0.3,s=100 ,marker="*")

# 축 라벨의 설정
plt.xlabel(iris["feature_names"][2],fontsize='large')
plt.ylabel(iris["feature_names"][3],fontsize='large')

plt.show()
```

클러스터링 3

```
import matplotlib.pyplot as plt
from sklearn import cluster
from sklearn import datasets

# iris 데이터를 로드
iris = datasets.load_iris()
data = iris['data']

# 학습 → 클러스터 생성
model = cluster.KMeans(n_clusters=3)
model.fit(data)

# 学習結果のラベル取得
labels = model.labels_

### グラフの描画
MARKERS = ["o", "^", "*", "v", "+", "x", "d", "p", "s", "1", "2"]

# 指定されたインデックスの feature 値で散布図を作成する関数
def scatter_by_features(feaut_idx1, feaut_idx2):
    for lbl in range(labels.max() + 1):
        clustered = data[labels == lbl]
        plt.scatter(clustered[:, feaut_idx1], clustered[:, feaut_idx2],
                    c='black', alpha=0.3, s=100,
                    marker=MARKERS[lbl], label='label {}'.format(lbl))

    plt.xlabel(iris["feature_names"][feaut_idx1], fontsize='xx-large')
    plt.ylabel(iris["feature_names"][feaut_idx2], fontsize='xx-large')

plt.figure(figsize=(10, 10))

# feature "sepal length" と "sepal width"
plt.subplot(3, 2, 1)
scatter_by_features(0, 1)

# feature "sepal length" と "petal length"
```

```
plt.subplot(3, 2, 2)
scatter_by_features(0, 2)

# feature "sepal length" と "petal width"
plt.subplot(3, 2, 3)
scatter_by_features(0, 3)

# feature "sepal width" と "petal length"
plt.subplot(3, 2, 4)
scatter_by_features(1, 2)

# feature "sepal width" と "petal width"
plt.subplot(3, 2, 5)
scatter_by_features(1, 3)

# feature "petal length" と "petal width"
plt.subplot(3, 2, 6)
scatter_by_features(2, 3)

plt.tight_layout()
plt.show()
```