세션 로딩 중


bada

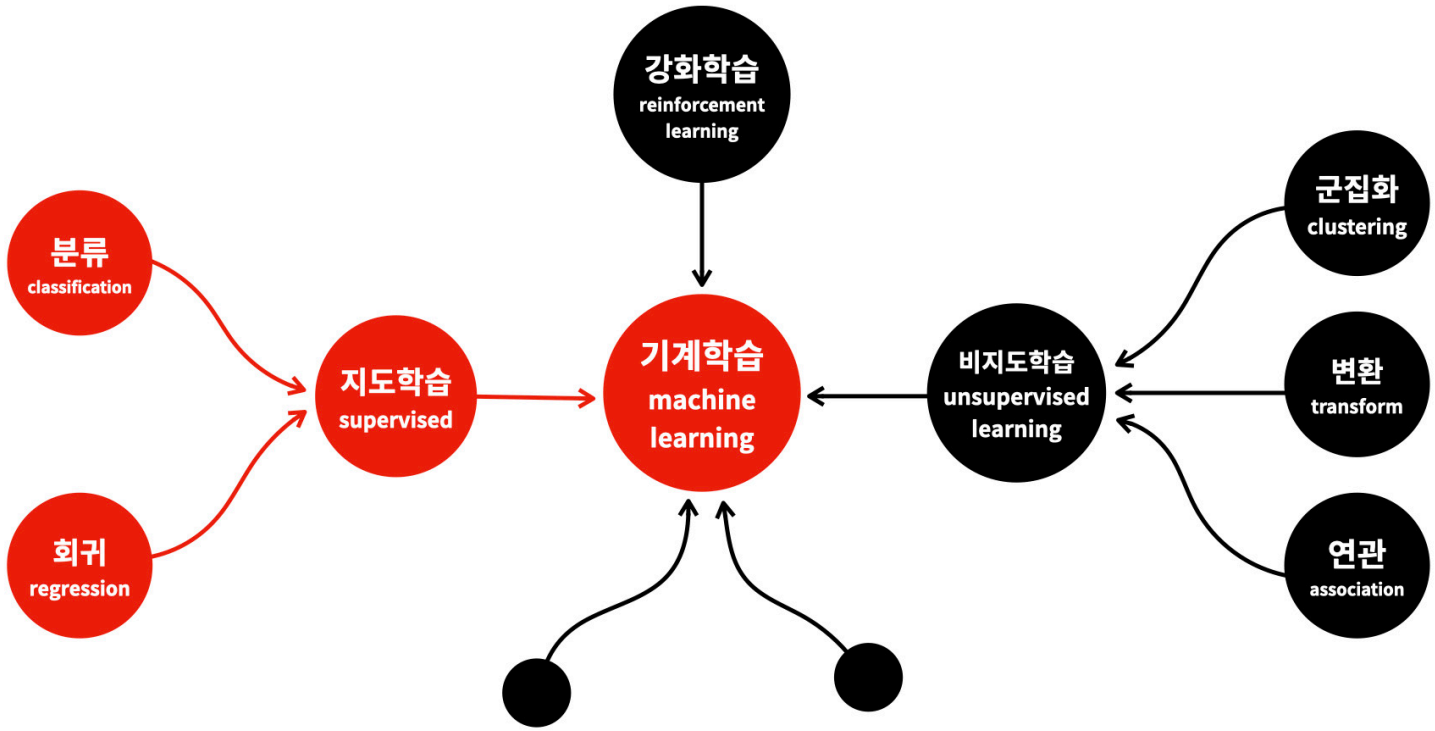
군집화 알고리즘

학습목표

- 분류 vs 회귀
- 군집화
- 연관규칙추론
- 아이디어톤

0. 지난시간 복습

지난시간 복습


강화학습
reinforcement learning
분류
classification
지도학습
supervised
기계학습
machine learning
비지도학습
unsupervised learning
군집화
clustering
변환
transform
회귀
regression
연관
association

지난시간 복습

물고기의 종류를 예측하는 모델

Vs

물고기의 무게를 예측하는 모델

Vs

물고기의 확률을 예측하는 모델

지난시간 복습

물고기의 종류 = 범주형 데이터 like 객관식

Vs

물고기의 무게 = 수치형 데이터 like 주관식

Vs

물고기의 무게 = 확률형 데이터 like 0 ~ 1

지난시간 복습

가지고 있는 데이터에 **독립변수**와 **종속변수**가 있고,

종속변수가 **이름**일때 **분류**를 이용하면 됩니다.

가지고 있는 데이터에 **독립변수**와 **종속변수**가 있고,

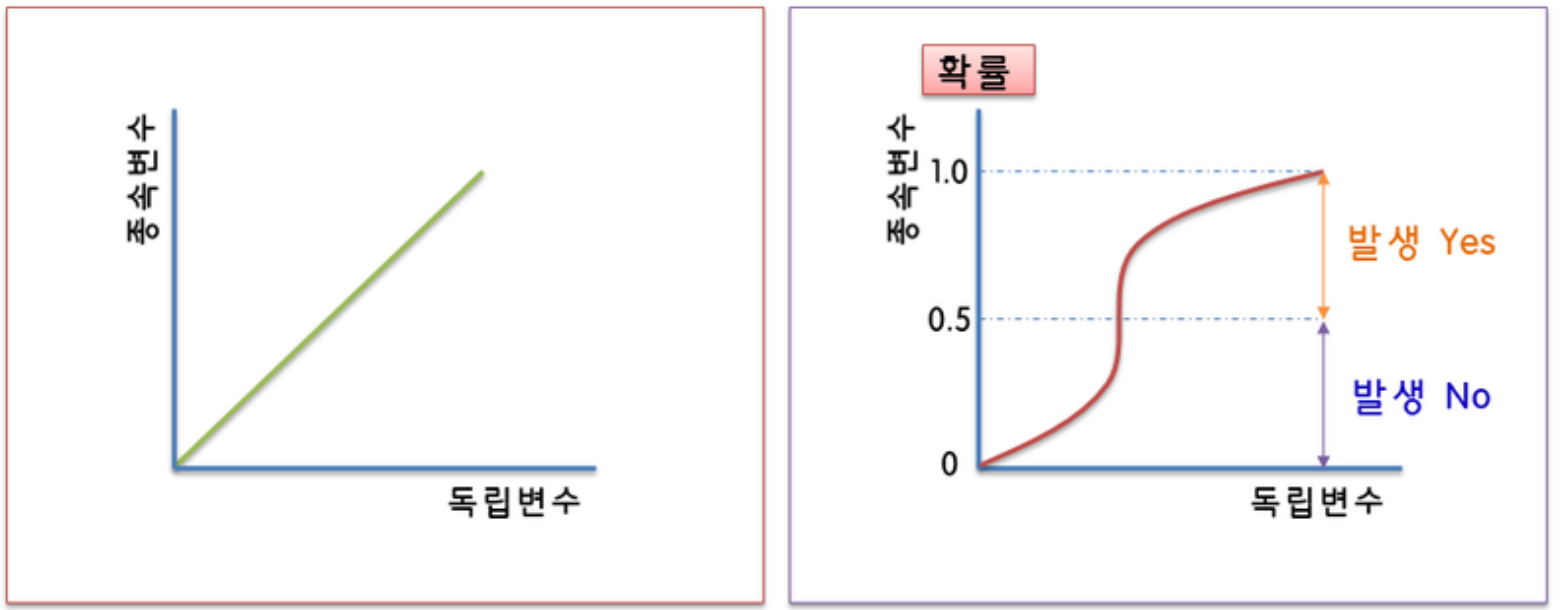
종속변수가 **숫자**일때 **회귀**를 이용하면 됩니다.

가지고 있는 데이터에 **독립변수**와 **종속변수**가 있고,

종속변수가 **확률**일때 **로지스틱 회귀**를 이용하면 됩니다.

지난시간 복습

독립변수와 종속변수의 관계



지난시간 복습

수치형 데이터

Or

범주형 데이터

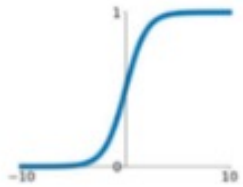
확률형 데이터

로지스틱 회귀

Activation Functions

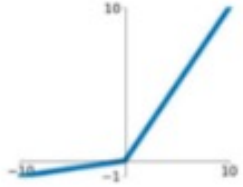
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



Leaky ReLU

$$\max(0.1x, x)$$



tanh

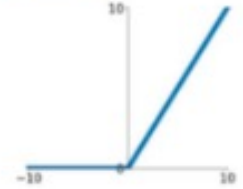
$$\tanh(x)$$

Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

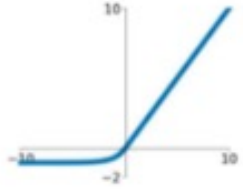
ReLU

$$\max(0, x)$$



ELU

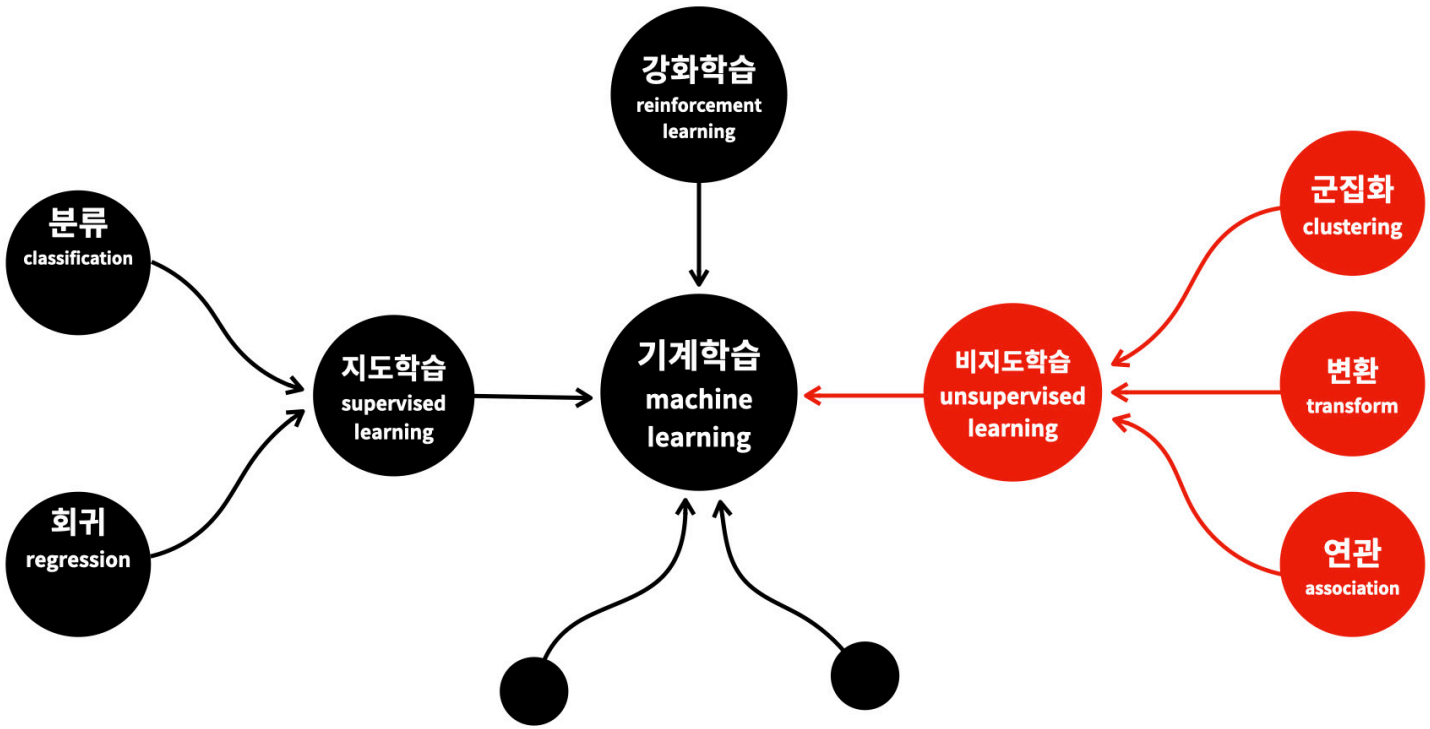
$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



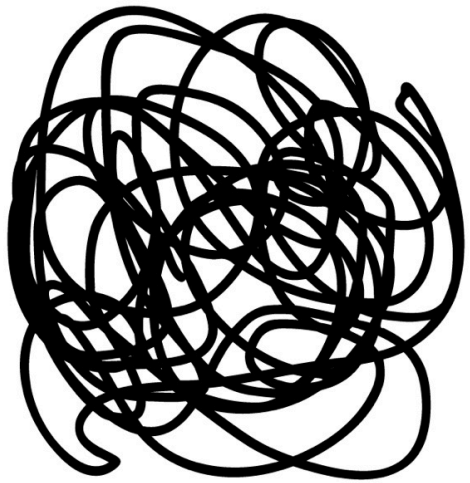
Different Activation Functions and their Graphs

1. 군집화 알고리즘

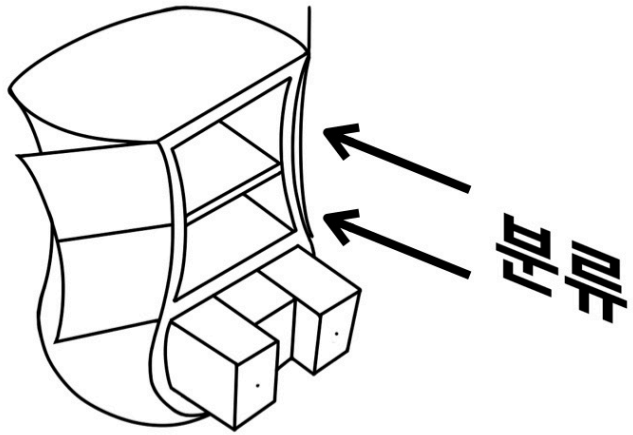
비지도 학습


강화학습
reinforcement learning
분류
classification
지도학습
supervised learning
기계학습
machine learning
비지도학습
unsupervised learning
군집화
clustering
변환
transform
회귀
regression
연관
association

군집화 알고리즘



군집화 알고리즘



군집화

?

2. 연관규칙추론

연관규칙추론

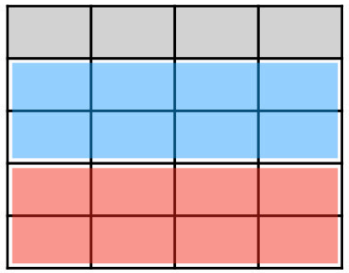
주문번호	라면	계란	식빵	우유	햄
1	O	O	X	X	O
2	O	O	X	X	X
3	O	O	X	X	X
4	X	X	O	O	X

연관규칙추론

주문번호	라면	계란	식빵	우유	햄	…
1	O	O	X	X	O	…
2	O	O	X	X	X	…
3	O	O	X	X	X	…
4	X	X	O	O	X	…
…	…	…	…	…	…	…

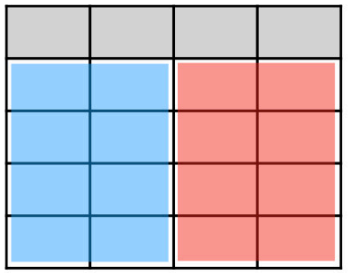
3. 군집화 vs 연관규칙추론

군집화 vs 연관규칙추론



군집화

clustering



연관규칙

association rule

<코드 미리보기>

코드 미리보기

```
from sklearn import datasets

iris = datasets.load_iris()
```

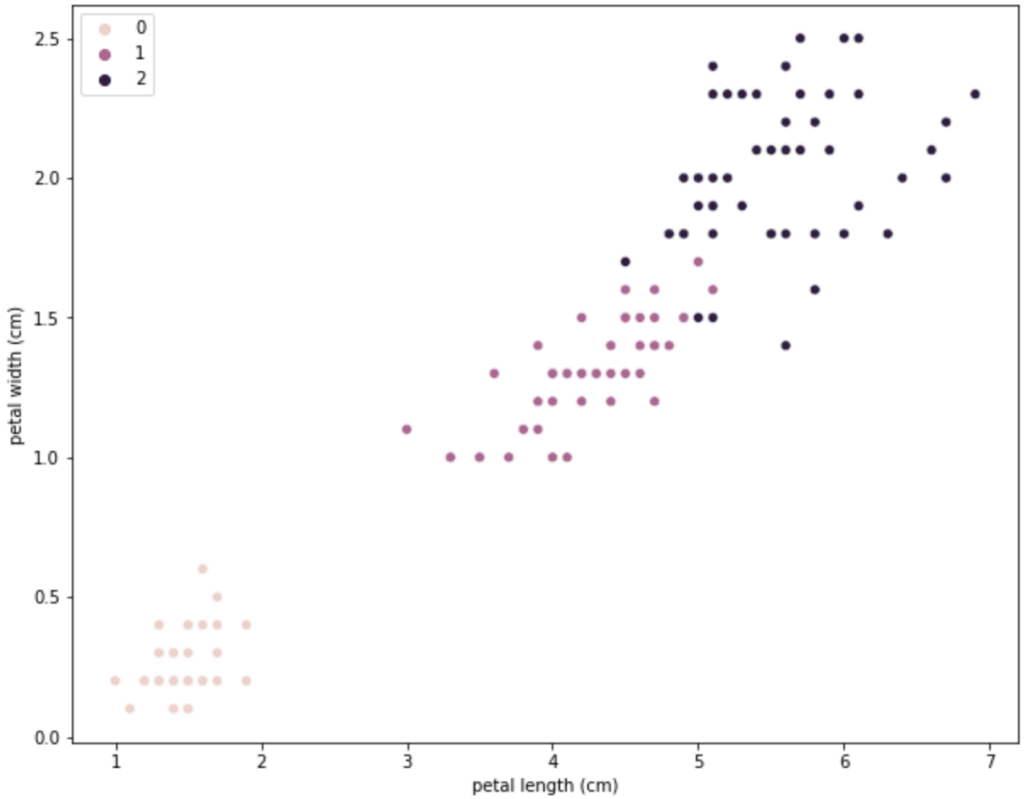
```
type(iris)
```

```
sklearn.utils.Bunch
```

코드 미리보기

```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(10,8))
sns.scatterplot(data=df_correlated, x='petal length (cm)', y='petal width (cm)', hue=data_label.reshape(-1))
```

```
<AxesSubplot:xlabel='petal length (cm)', ylabel='petal width (cm)'>
```



코드 미리보기

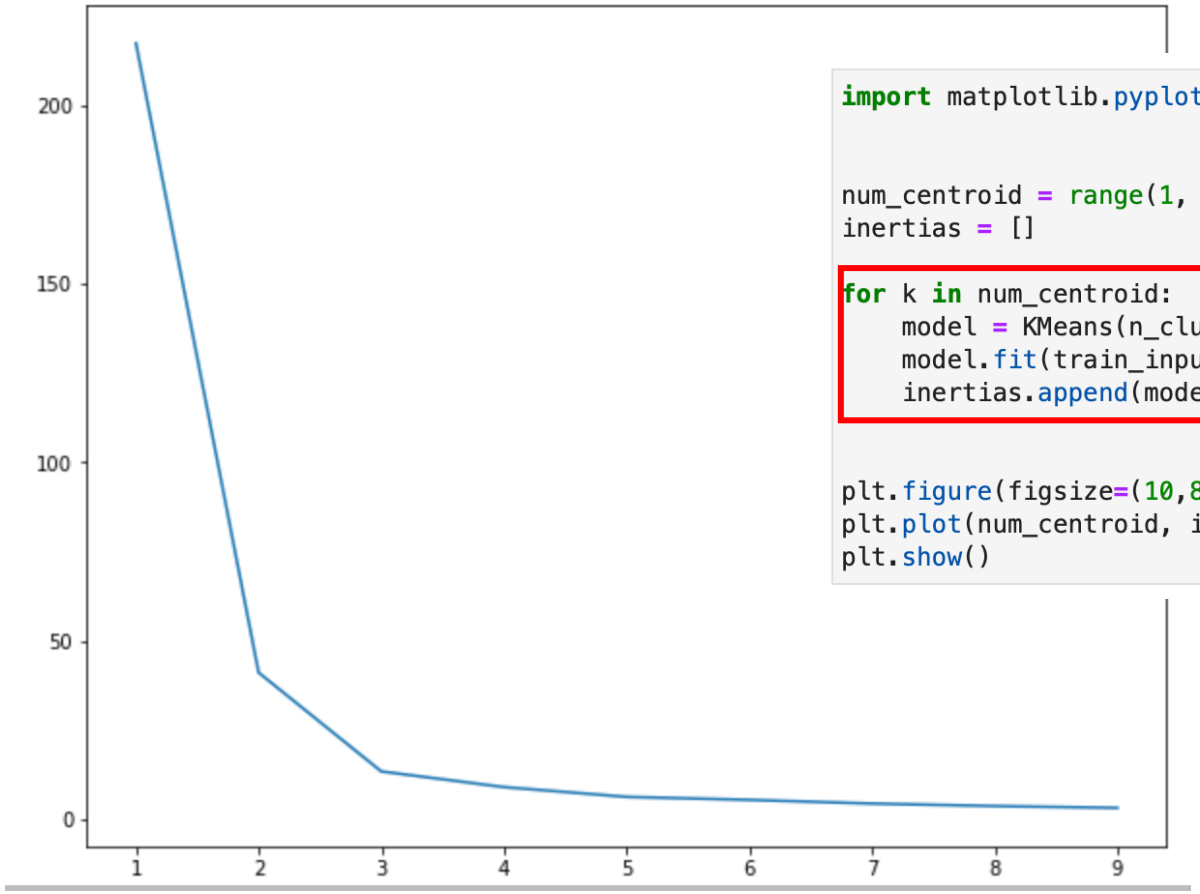
```
from sklearn.cluster import KMeans

km = KMeans(n_clusters=2)
km.fit(train_input)
```

```
KMeans(n_clusters=2)
```

```
km.cluster_centers_
```

```
array([[ 0.61194963,  0.6063856 ],
       [-1.3064313 , -1.24784939]])
```

코드 미리보기

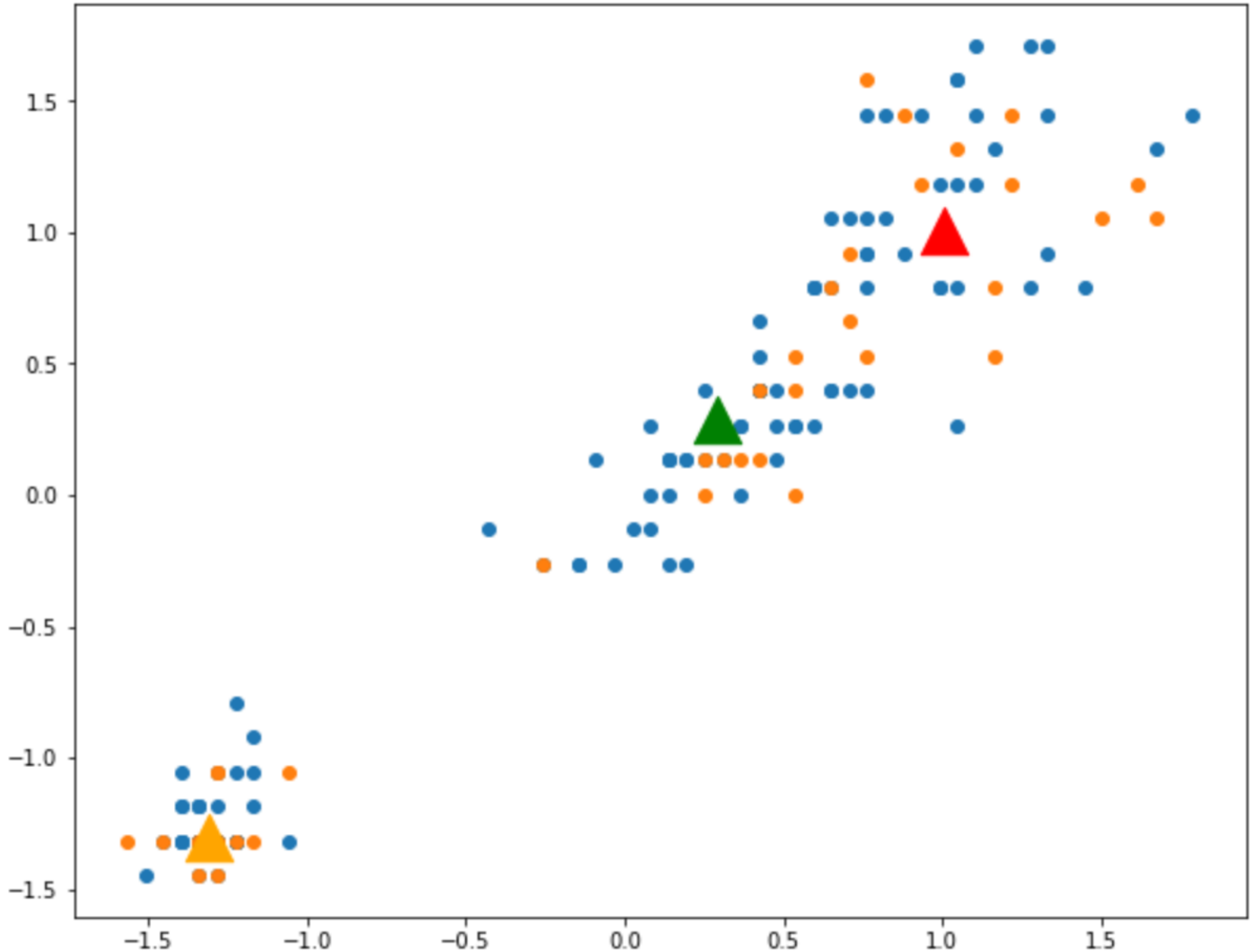



```python
import matplotlib.pyplot as plt


num_centroid = range(1, 10)
inertias = []

for k in num_centroid:
    model = KMeans(n_clusters=k)
    model.fit(train_input)
    inertias.append(model.inertia_)


plt.figure(figsize=(10,8))
plt.plot(num_centroid, inertias)
plt.show()
```

<실습>

```
import pandas as pd

context = iris['data']
df_iris = pd.DataFrame(context, columns=iris['feature_names'])
df_iris.head()
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
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
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2


1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5

?

세션 끝!