세션 로딩 중


bada

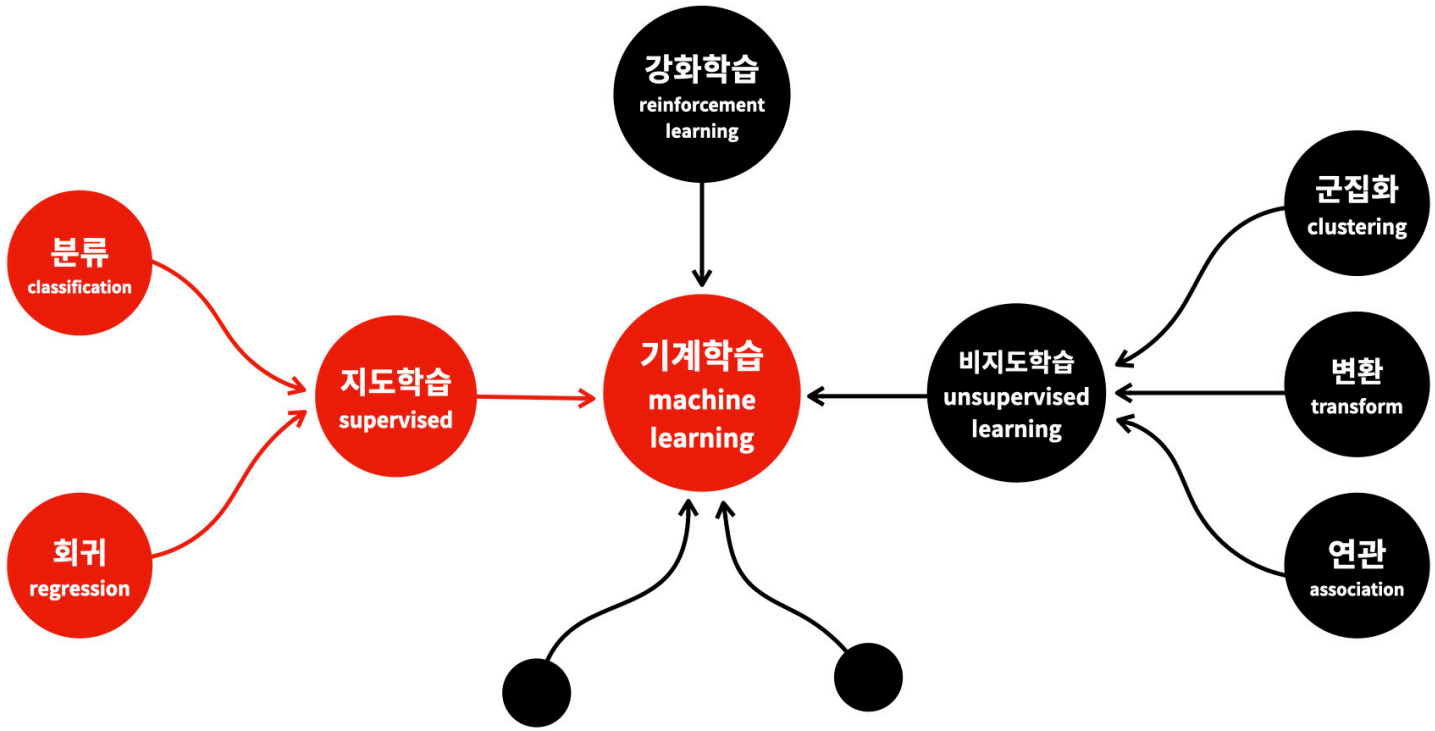
로지스틱 회귀

학습목표

- 지도학습 이해 : 분류 vs 회귀
- 사례 기반의 회귀 vs 모델 기반의 회귀
- 최근접이웃 회귀 vs 선형 회귀
- 특성 공학과 규제

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 종류 = 범주형 데이터 like 객관식

Vs

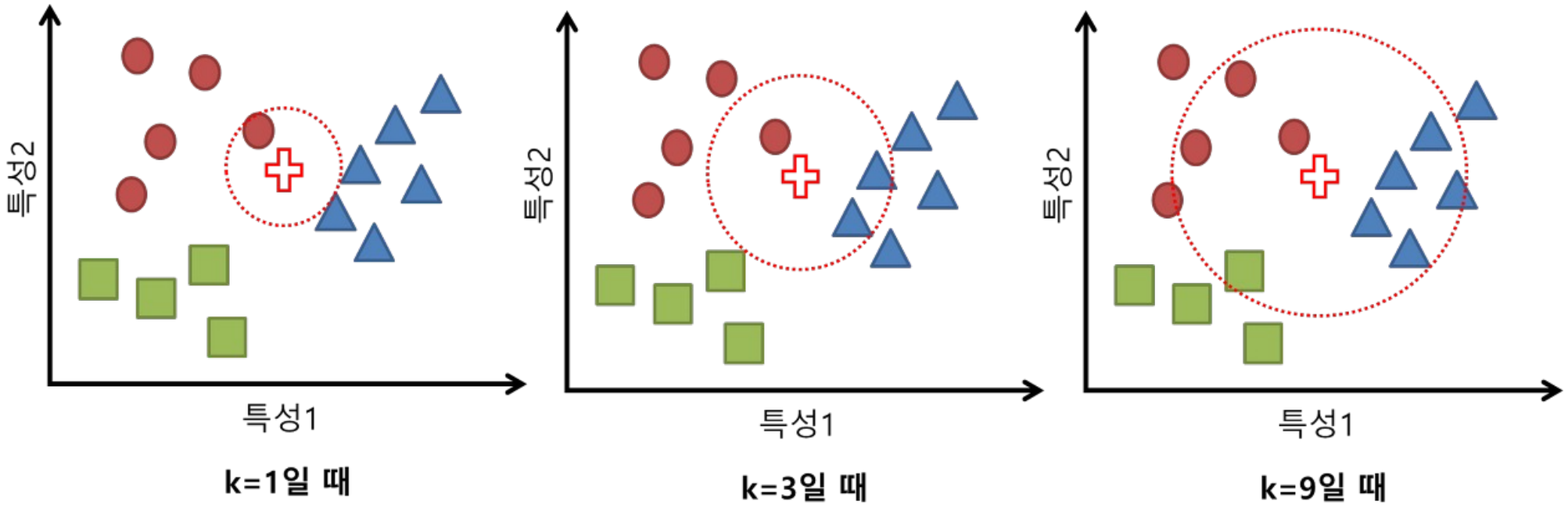
물고기의 무게 = 수치형 데이터 like 주관식

지난시간 복습

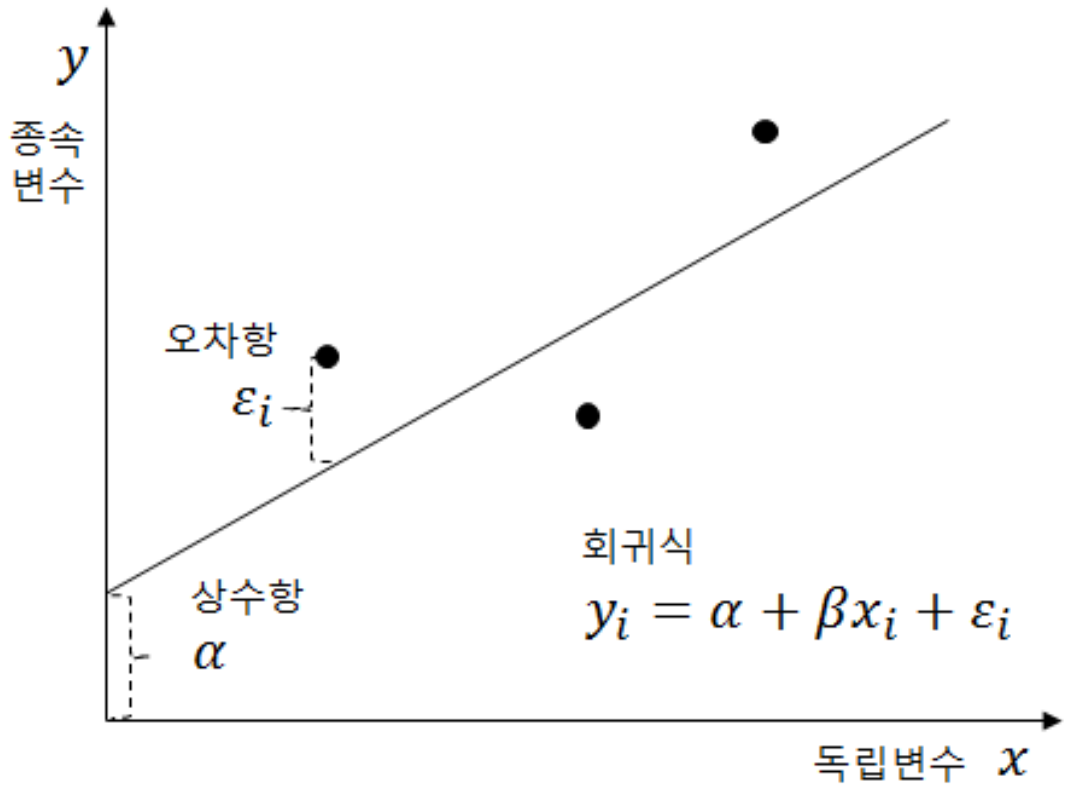
“가지고 있는 데이터에 **독립변수**와 **종속변수**가 있고,
종속변수가 **숫자**일 때 **회귀**를 이용하면 됩니다.”

“가지고 있는 데이터에 **독립변수**와 **종속변수**가 있고,
종속변수가 **이름**일 때 **분류**를 이용하면 됩니다.”

지난시간 복습



지난시간 복습


y
종속
변수
오차항
ε_i
상수항
α
회귀식
$y_i = \alpha + \beta x_i + \varepsilon_i$
독립변수 x

상황 살펴보기

티라노 닭강정은

인기 있는 신생 프렌차이즈 업체이다.

그러던 어느날,

티라노 사장님은

심각한 고민에 빠졌다.

그것은 바로

점포의 갯수는 점점 늘어나는데,

그에 따른 수요예측이 힘들었던 것이다.

그렇게 티라노 사장은

고려대학교 경영대학 경영데이터분석학회 bada를 찾게 되는데...

상황 살펴보기

독립변수		종속변수
점포의 갯수	→	수요량

지난시간 복습

```
df_total = pd.concat([df1,df2], axis=0, ignore_index=True)
df_total
```

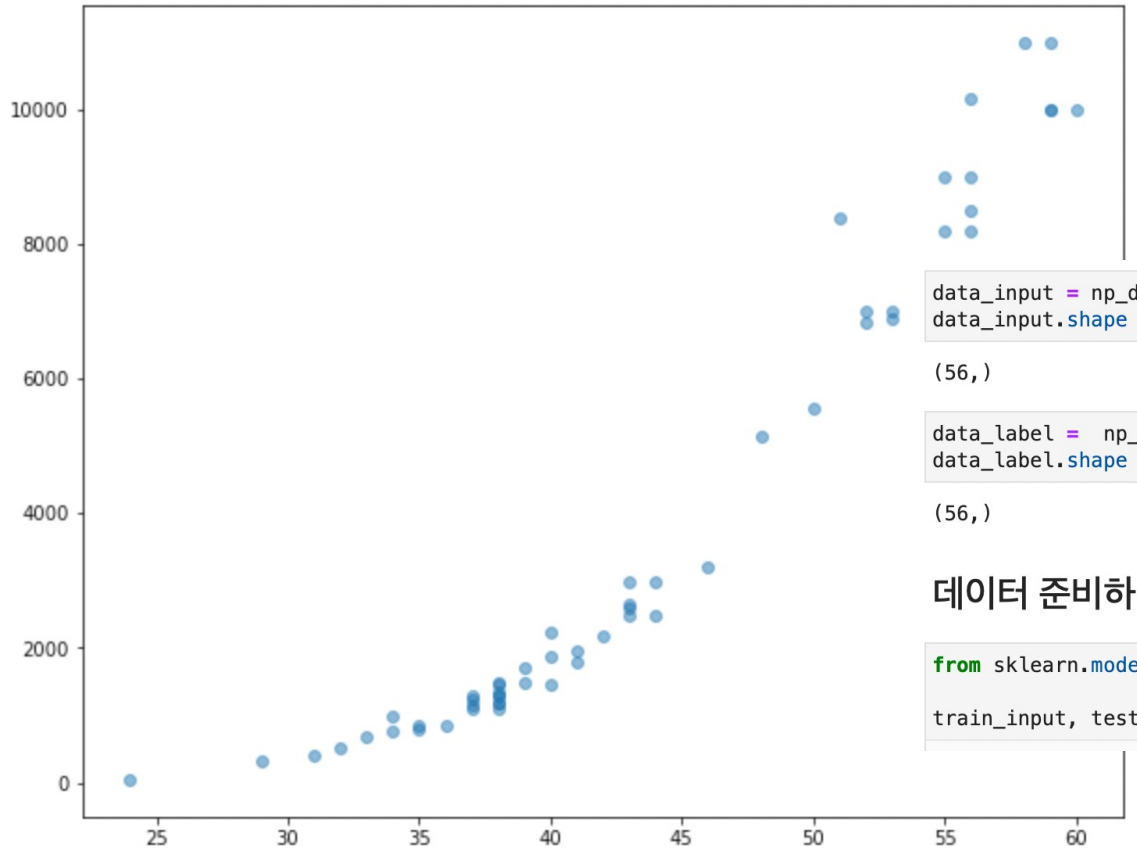
```
df_processed = df_total.drop('Unnamed: 0', axis=1)
df_processed
```

```
df_processed.dropna().tail()
```

	branch_num	sales_total
51	58.0	10993.0
52	59.0	9993.0
53	59.0	10993.0
54	59.0	9993.0
55	60.0	9993.0

지난시간 복습

```
import matplotlib.pyplot as plt

plt.figure(figsize=(10,8))
plt.scatter(df_processed['branch_num'],df_processed['sales_total'], alpha=0.5)
plt.show()
```




```
data_input = np_data[:,0]
data_input.shape
```

```
(56,)
```

```
data_label =  np_data[:,1]
data_label.shape
```

```
(56,)
```

데이터 준비하기

```
from sklearn.model_selection import train_test_split

train_input, test_input, train_label, test_label = train_test_split(data_input, data_label, shuffle=True)
```

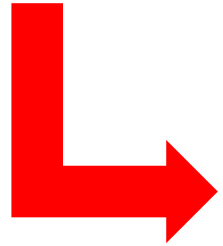
지난시간 복습

모델링

```
from sklearn.linear_model import LinearRegression

lr = LinearRegression()
lr.fit(train_input.reshape(-1,1), train_label)
```

```
LinearRegression()
```

	A	B
1	32, 34, 123, 45, 564	
2		
3		
4		
5		
6		



	A	B
1	32	
2	34	
3	123	
4	45	
5	564	
6		

지난시간 복습

```
lr.score(train_input.reshape(-1,1), train_label)
```

```
0.9467333340263063
```

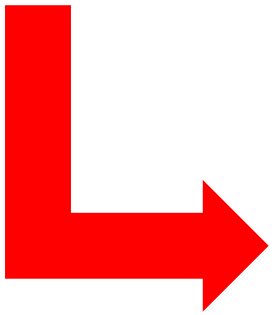
```
lr.score(test_input.reshape(-1,1), test_label)
```

```
0.7909571483180031
```

지난시간 복습

```
data_input
```

```
array([24., 29., 31., 32., 33., 34., 34., 35., 35., 36., 37., 37., 37.,
       37., 38., 38., 38., 38., 38., 38., 38., 38., 39., 39., 40., 40.,
       40., 41., 41., 42., 43., 43., 43., 43., 44., 44., 46., 48., 50.,
       51., 52., 52., 53., 53., 55., 55., 55., 56., 56., 56., 56., 58.,
       59., 59., 59., 60.])
```



```
data_input_squared = data_input * data_input
```

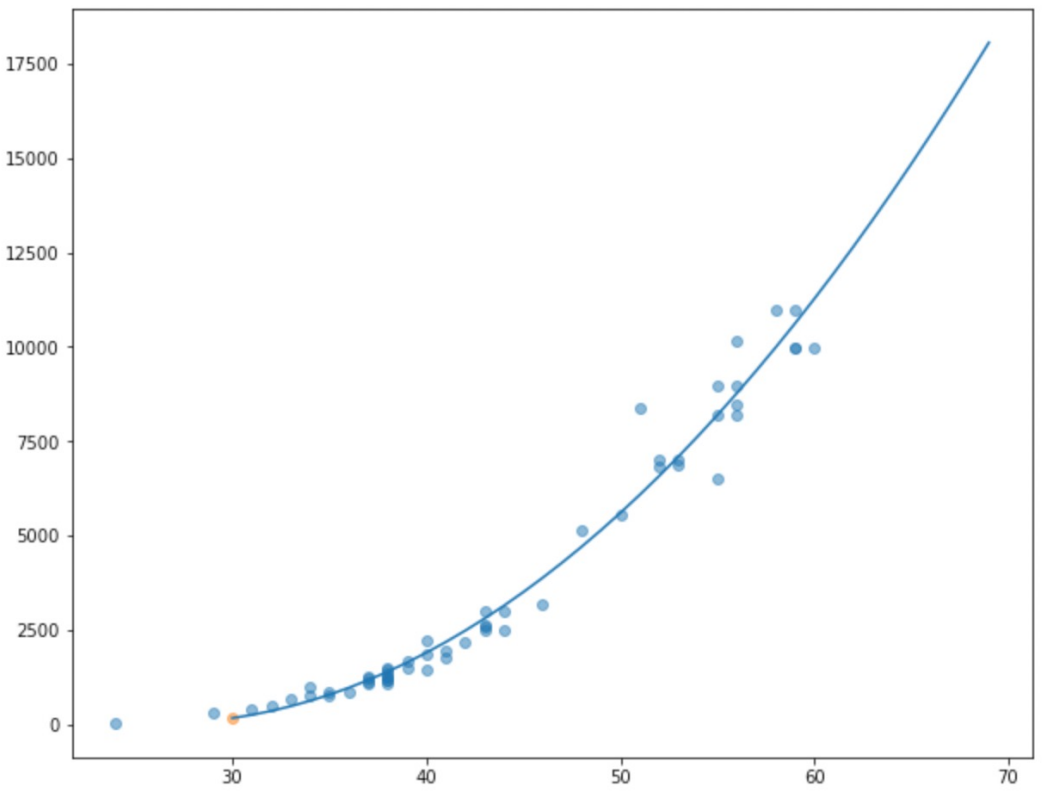
```
data_input_new = list(zip(data_input_squared, data_input))
data_input_new
```

```
[(576.0, 24.0),
 (841.0, 29.0),
 (961.0, 31.0),
 (1024.0, 32.0),
 (1089.0, 33.0),
 (1156.0, 34.0),
```

지난시간 복습

```
import matplotlib.pyplot as plt
import numpy as np

inp = int(input())
prediction = lr_new.predict([[inp**2, inp]])

plt.figure(figsize=(10,8))
plt.scatter(df_processed['branch_num'],df_processed['sales_total'], alpha=0.5)

point = np.arange(30, 70)
plt.plot(
    point,
    (lr_new.coef_[0]*(point ** 2)) + (lr_new.coef_[1]*point) + (lr_new.intercept_)
)

plt.scatter(inp, prediction, alpha=0.5)
plt.show()
```

30



1. 회귀 알고리즘 정리

회귀 알고리즘 정리

물고기의 종류를 예측하는 모델

Vs

물고기의 무게를 예측하는 모델

Vs

물고기의 확률을 예측하는 모델

회귀 알고리즘 정리

물고기의 종류 = 범주형 데이터 like 객관식

Vs

물고기의 무게 = 수치형 데이터 like 주관식

Vs

물고기의 무게 = 확률형 데이터 like 0 ~ 1

회귀 알고리즘 정리

가지고 있는 데이터에 **독립변수**와 **종속변수**가 있고,

종속변수가 **이름**일때 **분류**를 이용하면 됩니다.

가지고 있는 데이터에 **독립변수**와 **종속변수**가 있고,

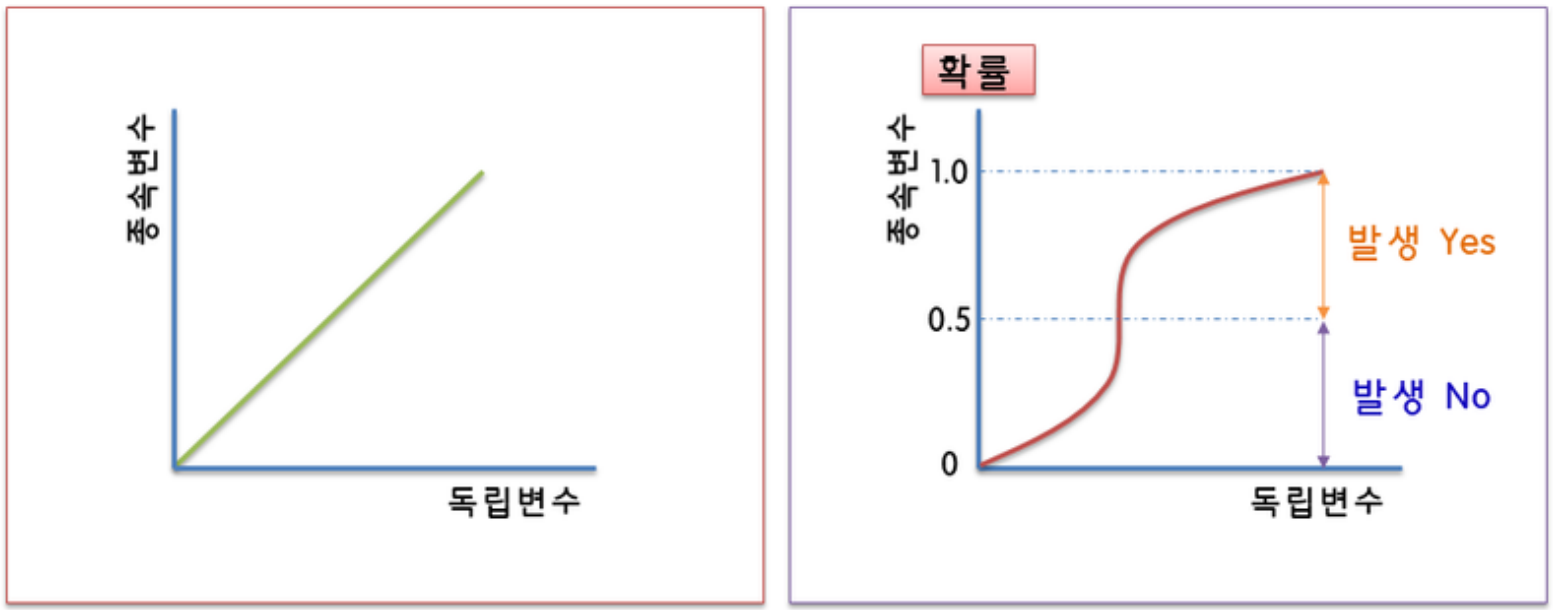
종속변수가 **숫자**일때 **회귀**를 이용하면 됩니다.

가지고 있는 데이터에 **독립변수**와 **종속변수**가 있고,

종속변수가 **확률**일때 **로지스틱 회귀**를 이용하면 됩니다.

회귀 알고리즘 정리

독립변수와 종속변수의 관계



?

2. 로지스틱 회귀

로지스틱 회귀

수치형 데이터

Or

범주형 데이터

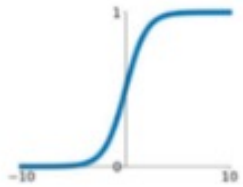
확률형 데이터

로지스틱 회귀

Activation Functions

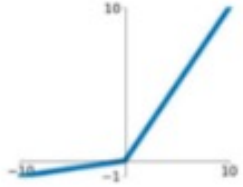
Sigmoid

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



Leaky ReLU

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



tanh

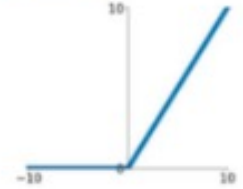
$$\tanh(x)$$

Maxout

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

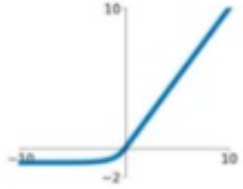
ReLU

$$\max(0, x)$$



ELU

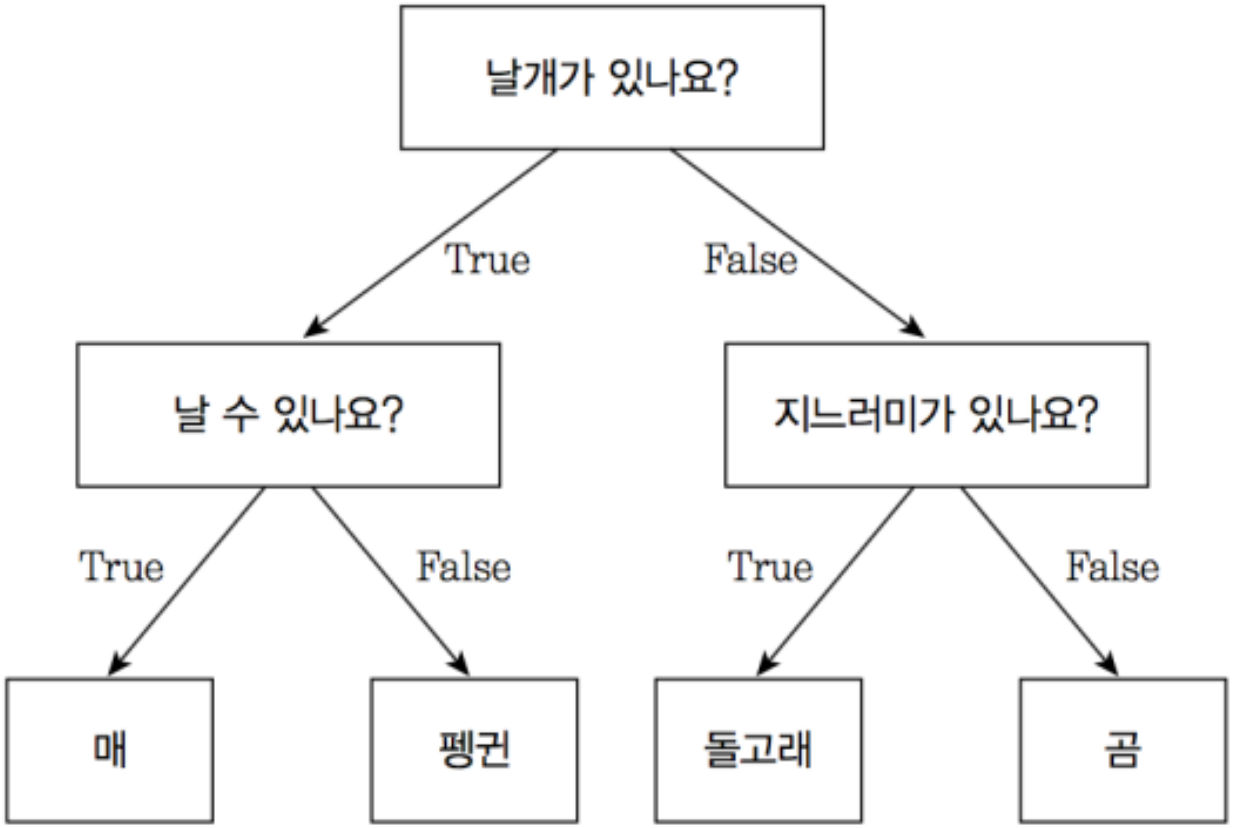
$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



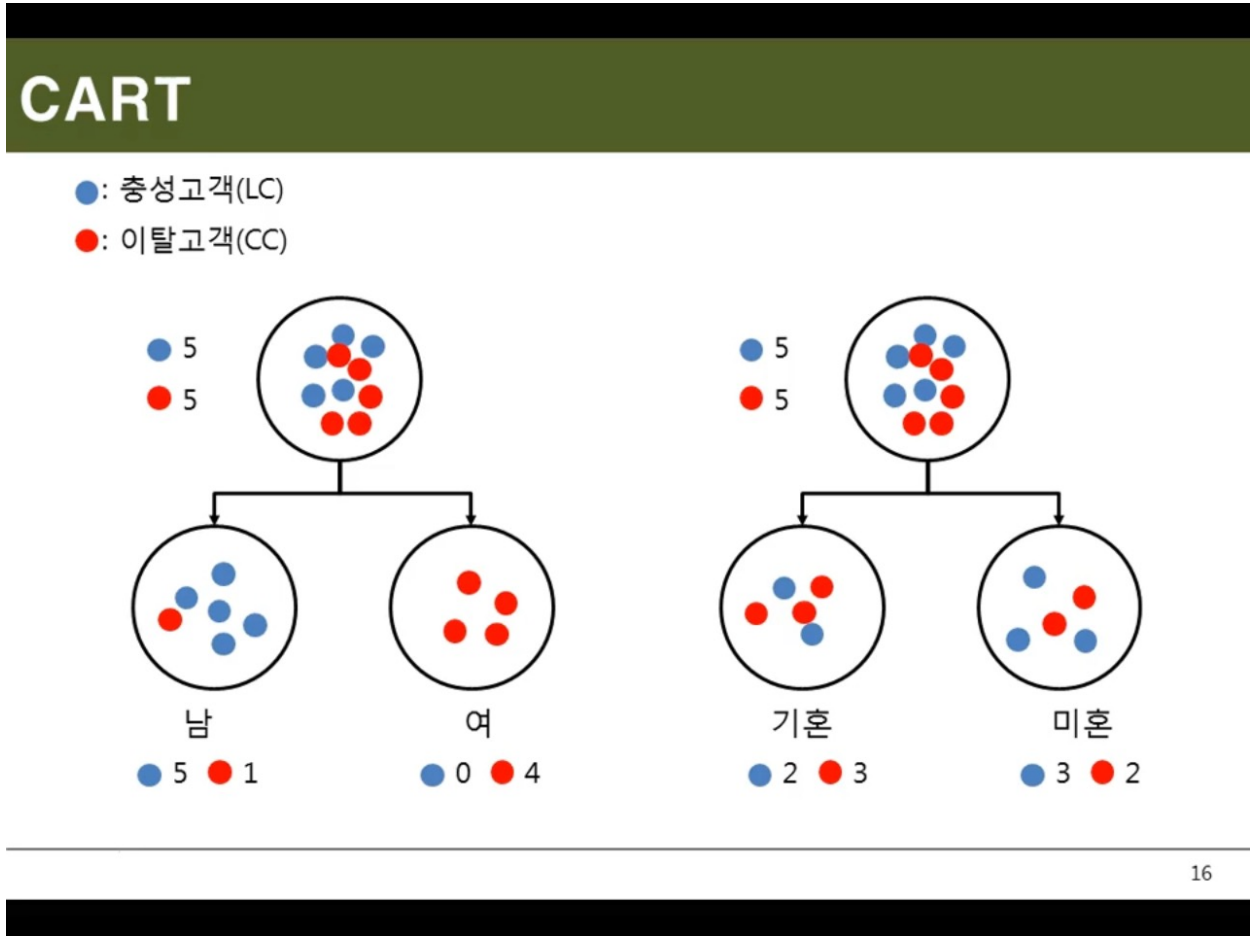
Different Activation Functions and their Graphs

3. 의사결정나무 알고리즘

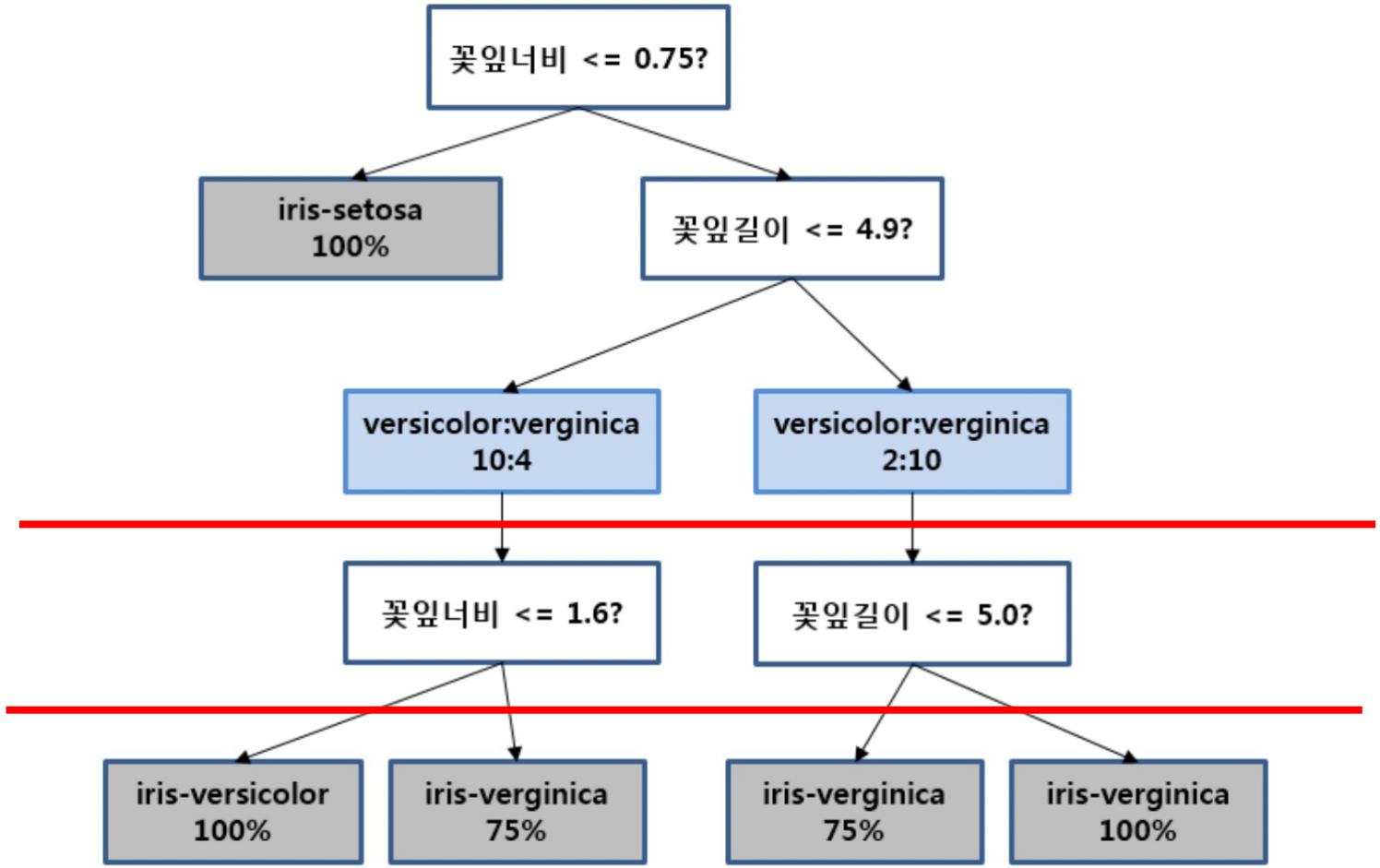
의사결정나무 알고리즘


날개가 있나요?
True
False
날 수 있나요?
지느러미가 있나요?
True
False
True
False
매
펭귄
돌고래
곰

의사결정나무 알고리즘



의사결정나무 알고리즘


꽃잎너비 <= 0.75?
iris-setosa
100%
꽃잎길이 <= 4.9?
versicolor:verginica
10:4
versicolor:verginica
2:10
꽃잎너비 <= 1.6?
꽃잎길이 <= 5.0?
iris-versicolor
100%
iris-verginica
75%
iris-verginica
75%
iris-verginica
100%

<코드 미리보기>

코드 미리보기

```
from sklearn.preprocessing import StandardScaler

ss = StandardScaler()
ss.fit(train_input)
train_scaled = ss.transform(train_input)
test_scaled = ss.transform(test_input)
```

코드 미리보기

```
from sklearn.linear_model import LogisticRegression

lr = LogisticRegression()
lr.fit(train_scaled, train_target)

print(lr.score(train_
print(lr.score(test_s
```

```
0.7808350971714451
0.7776923076923077
```

```
df = pd.DataFrame(lr.predict_proba(train_scaled), columns=lr.classes_)
df.head()
```

	0.0	1.0
0	0.061893	0.938107
1	0.217426	0.782574
2	0.407036	0.592964
3	0.452267	0.547733
4	0.005308	0.994692

코드 미리보기

```
from sklearn.tree import DecisionTreeClassifier

dt = DecisionTreeClassifier(random_state=42)
dt.fit(train_scaled, train_target)

print(dt.score(train_scaled, tra
print(dt.score(test_scaled, test
```

```
import matplotlib.pyplot as plt
from sklearn.tree import plot_tree

plt.figure(figsize=(10,7))
plot_tree(dt)
plt.show()
```

코드 미리보기

```
from sklearn.tree import DecisionTreeClassifier

dt = DecisionTreeClassifier(random_state=42)
dt.fit(train_scaled, train_target)

print(dt.score(train_scaled, tra
print(dt.score(test_scaled, test
```

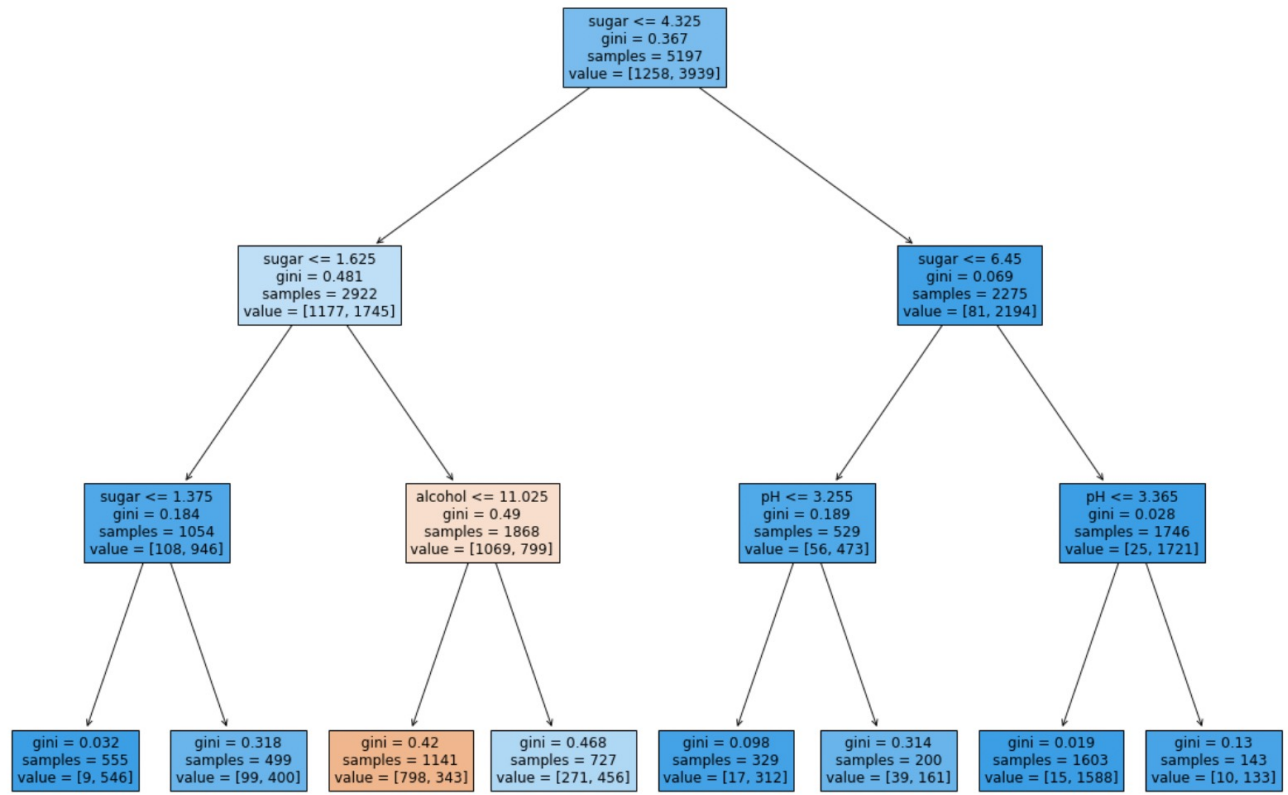
```
import matplotlib.pyplot as plt
from sklearn.tree import plot_tree

plt.figure(figsize=(10,7))
plot_tree(dt)
plt.show()
```

<실습>

	Species	Weight	Length	Diagonal	Height	Width
0	Bream	242.0	25.4	30.0	11.5200	4.0200
1	Bream	290.0	26.3	31.2	12.4800	4.3056
2	Bream	340.0	26.5	31.1	12.3778	4.6961
3	Bream					
4	Bream					

	Bream	Parkki	Perch	Pike	Roach	Smelt	Whitefish
0	7.249191e-06	0.013512	0.841261	0.000315	0.135681	0.006671	0.002552
1	7.149746e-09	0.002556	0.043900	0.000034	0.007312	0.946192	0.000005
2	1.863769e-05	0.000003	0.034024	0.934879	0.015023	0.016022	0.000030
3	1.093327e-02	0.034052	0.305554	0.006618	0.566544	0.000069	0.076231
4	4.490259e-06	0.000367	0.904002	0.002416	0.089471	0.002410	0.001329

	0.0	1.0
0	0.061893	0.938107
1	0.217426	0.782574
2	0.407036	0.592964
3	0.452267	0.547733
4	0.005308	0.994692



?

세션 끝!