

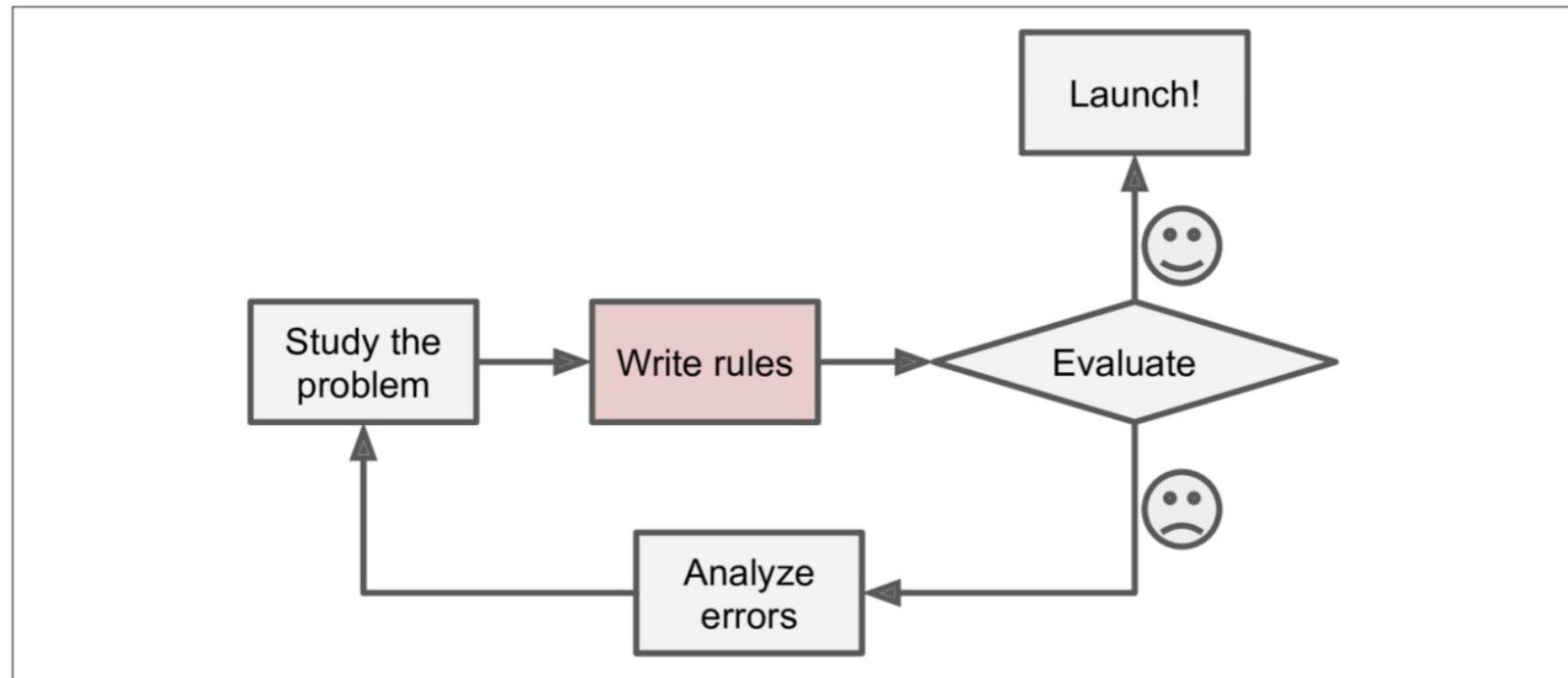
Chapter 12. Basic of Regression



Previous Story

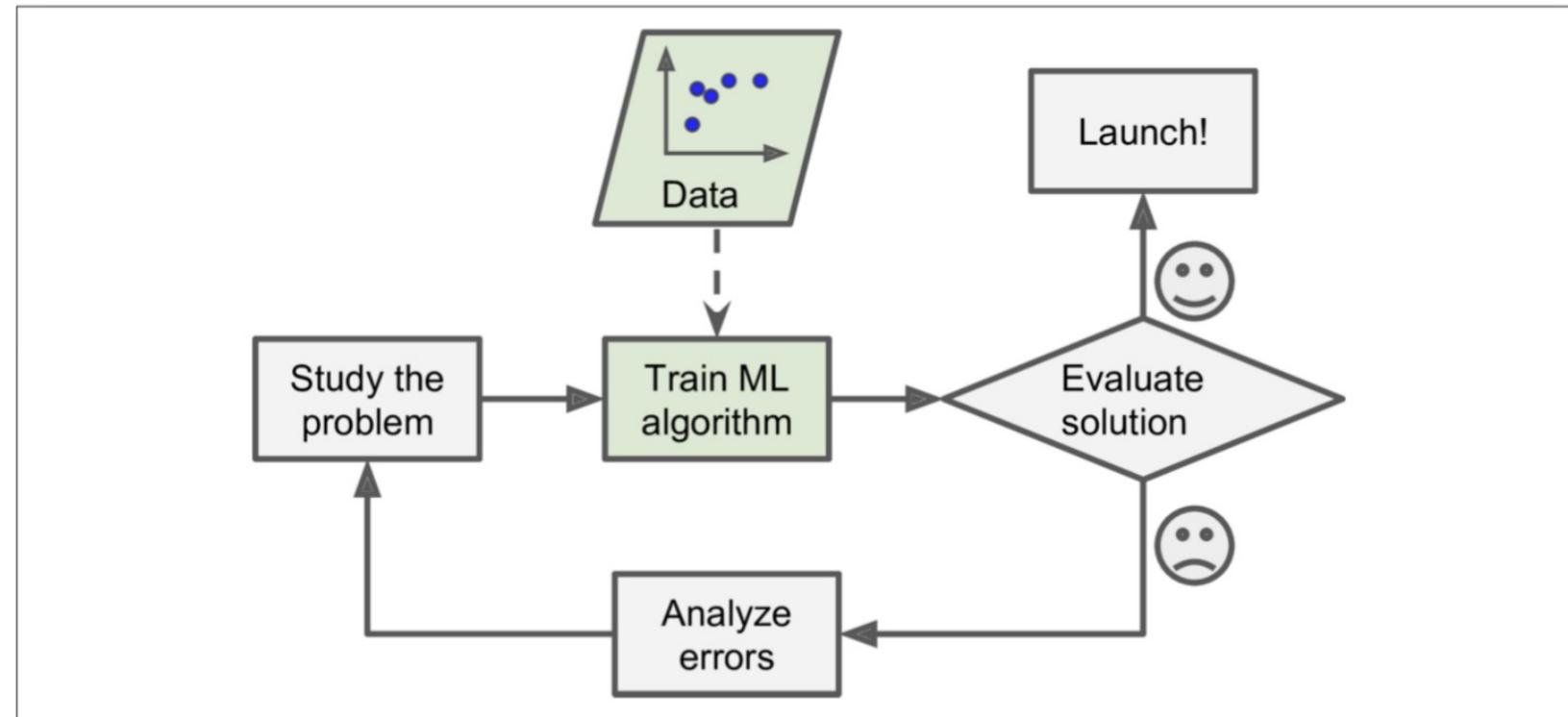
[Previous Story](#)

일반적인 문제해결 절차



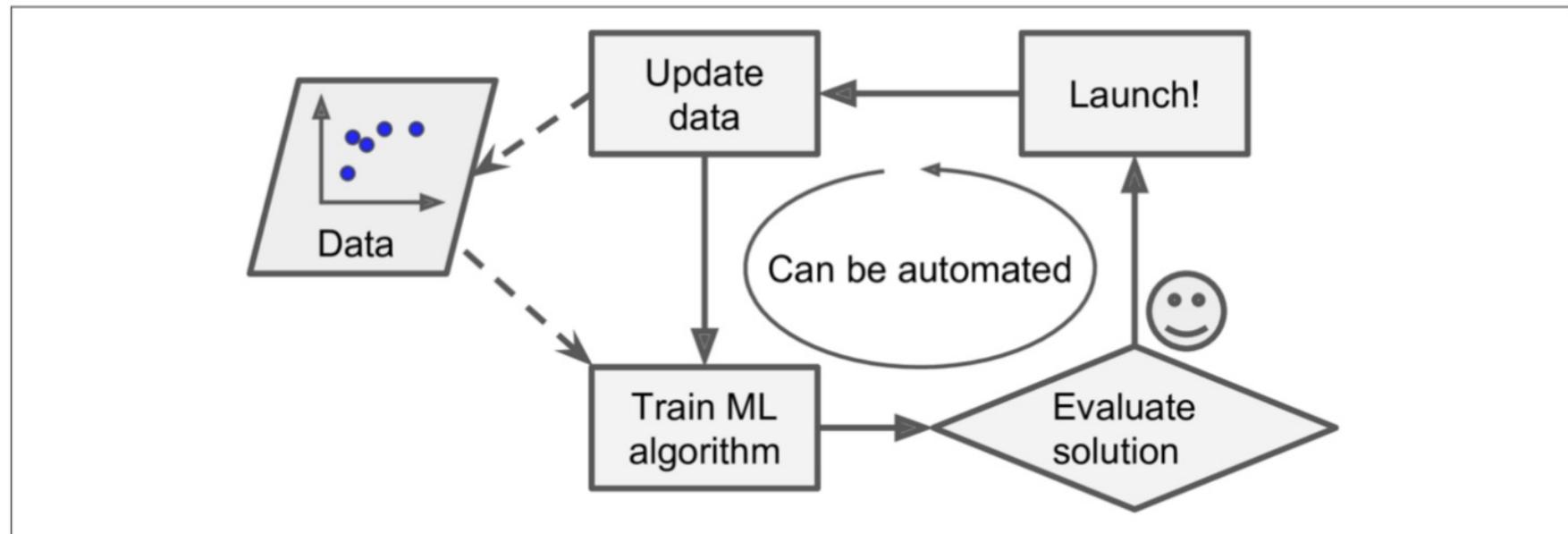
[Previous Story](#)

만약 데이터를 기반으로 한다면



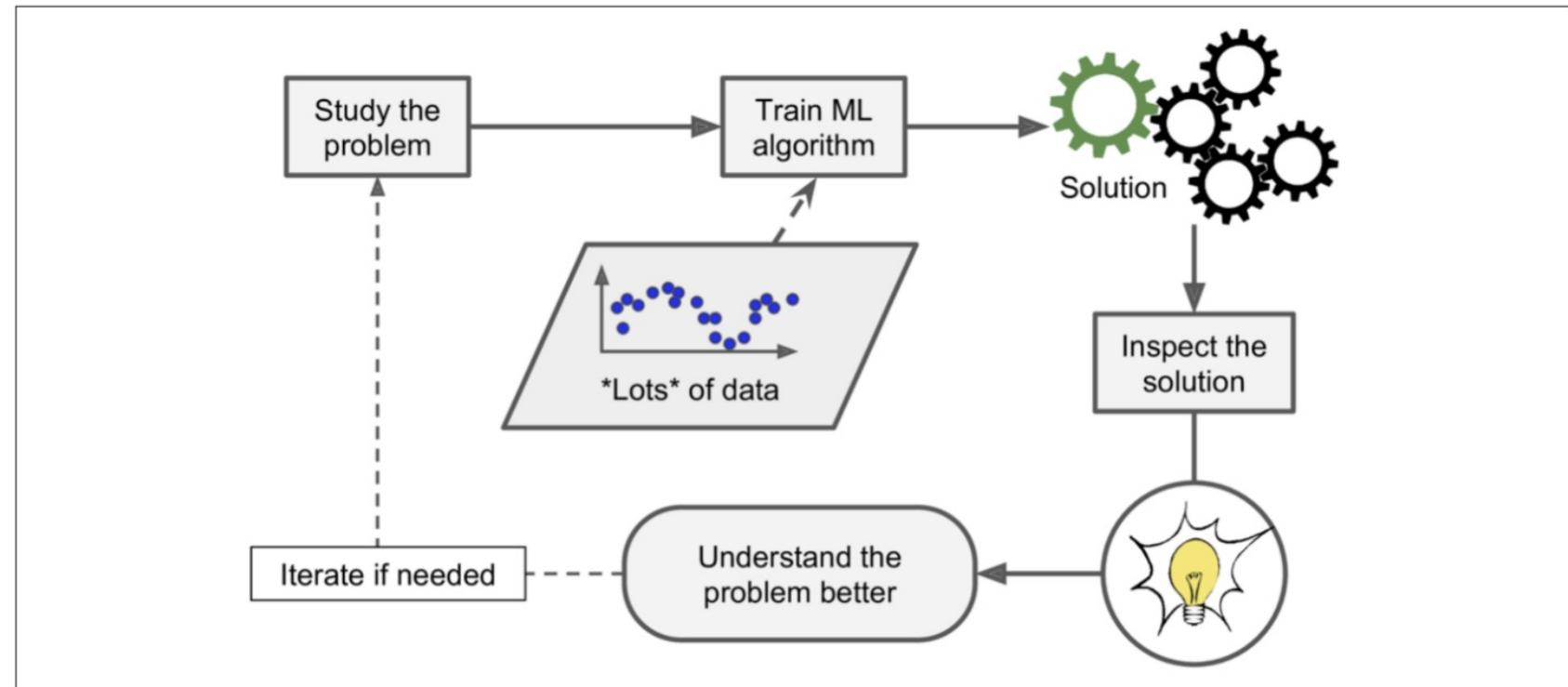
[Previous Story](#)

모델 스스로 데이터를 기반으로 변화에 대응할 수 있음



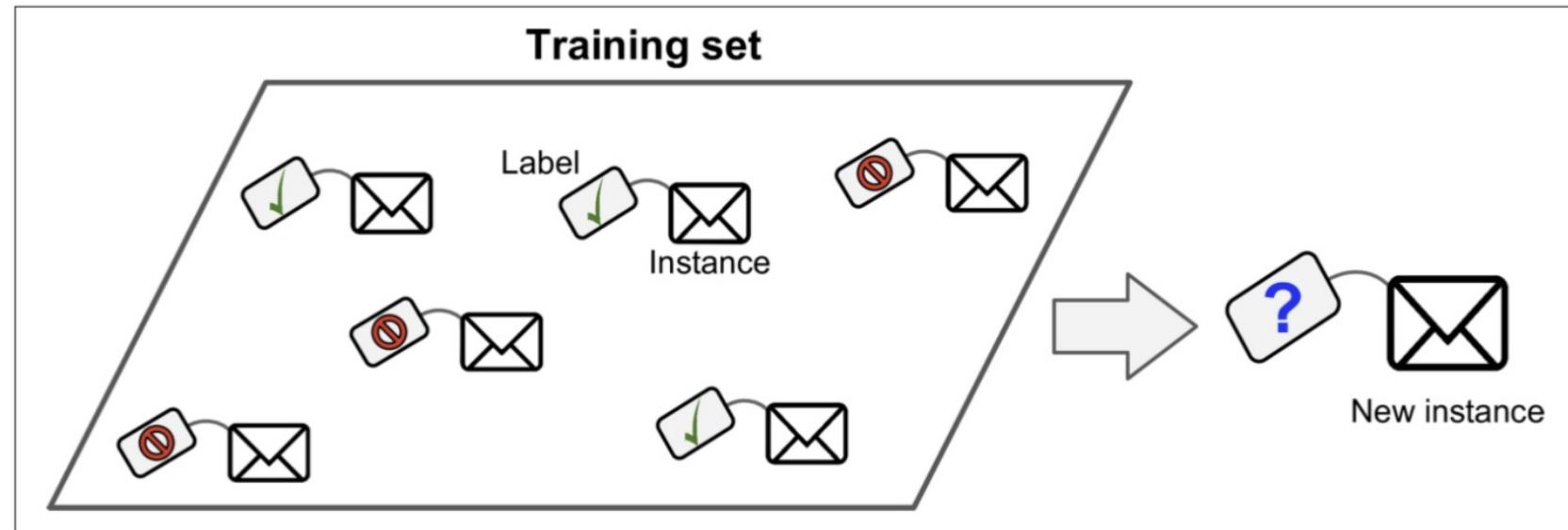
[Previous Story](#)

심지어 머신러닝을 통해 우리가 배울 수도 있다



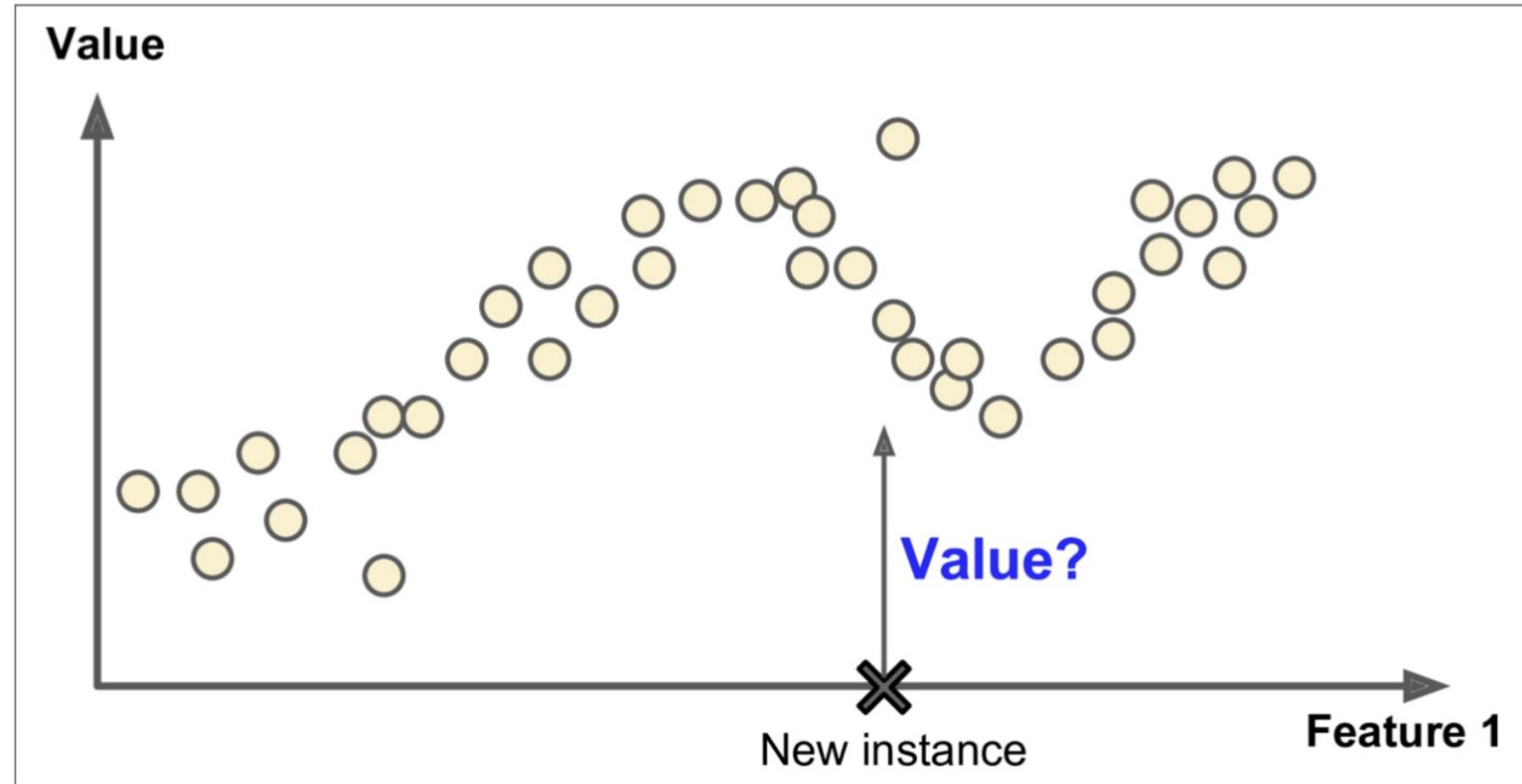
[Previous Story](#)

지도학습 - 분류 Classification



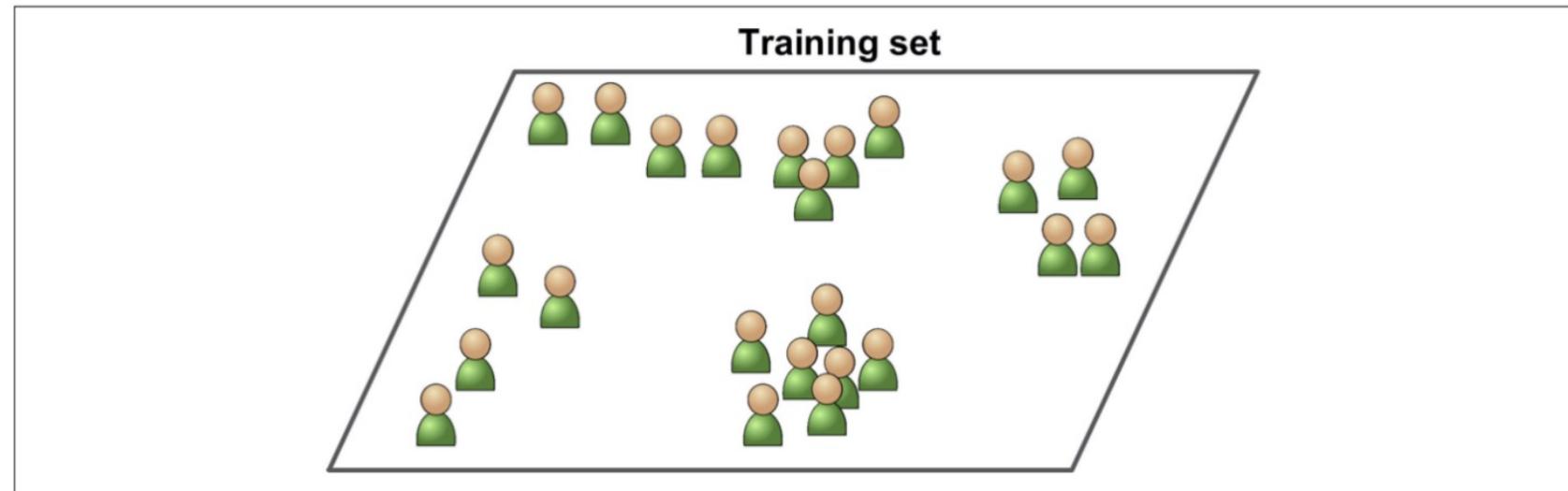
[Previous Story](#)

지도학습 - 회귀 Regression



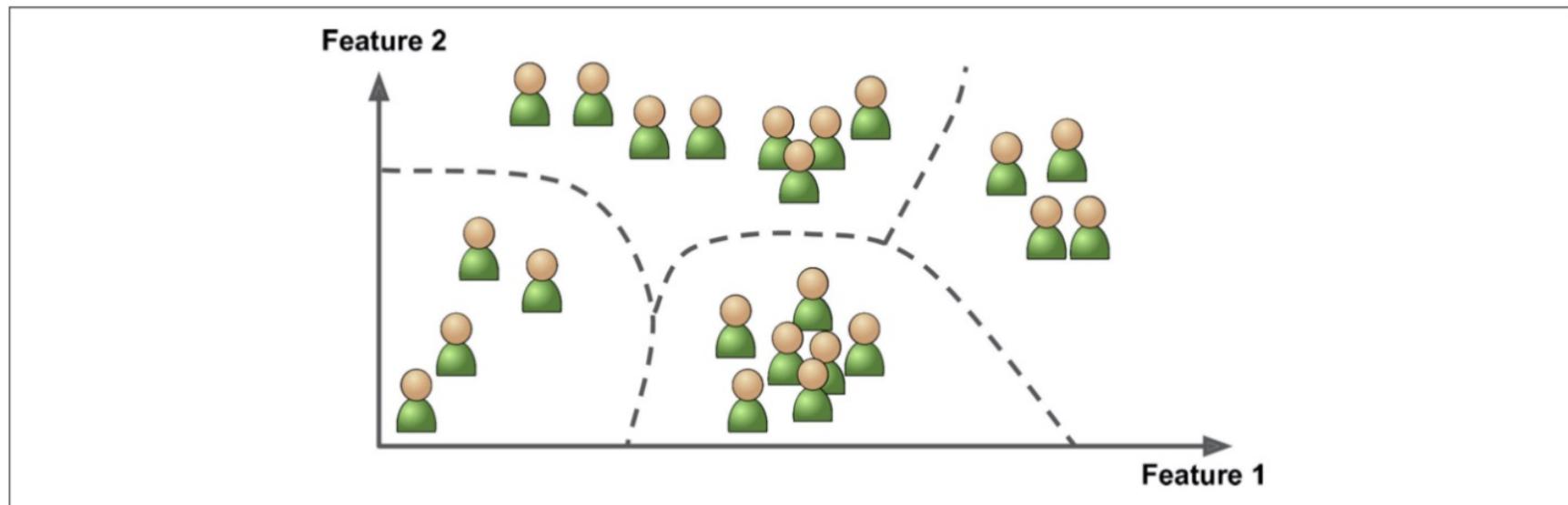
Previous Story

비지도학습은 레이블이 없다



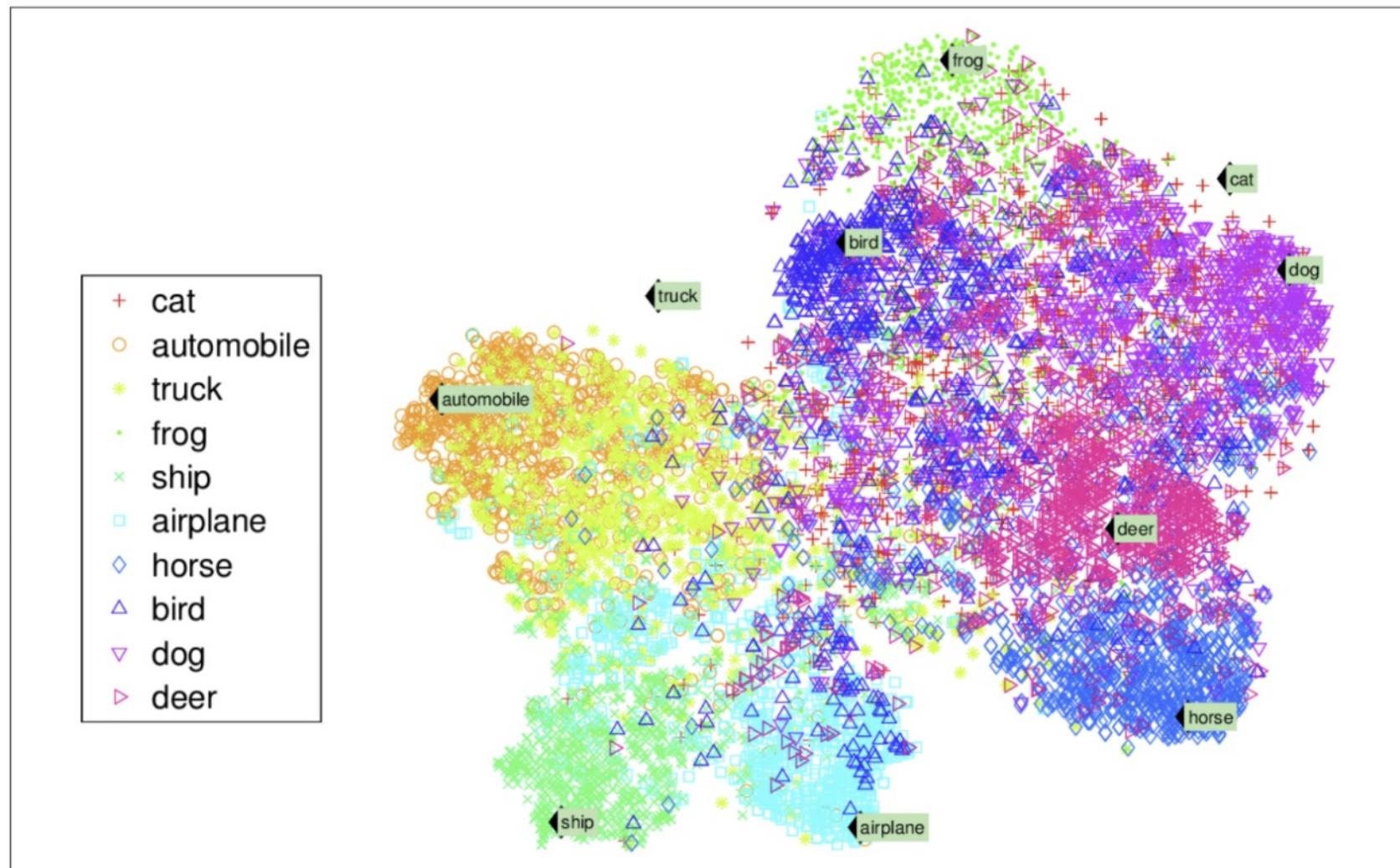
Previous Story

비지도 학습 - 군집



Previous Story

비지도 학습 - 차원 축소



Regression? 회귀?

Regression? 회귀?

만약 주택의 넓이과 가격이라는 데이터가 있고 주택가격을 예측한다면

● 머신러닝 모델을 어떻게 만들까요?

주택 가격 예측

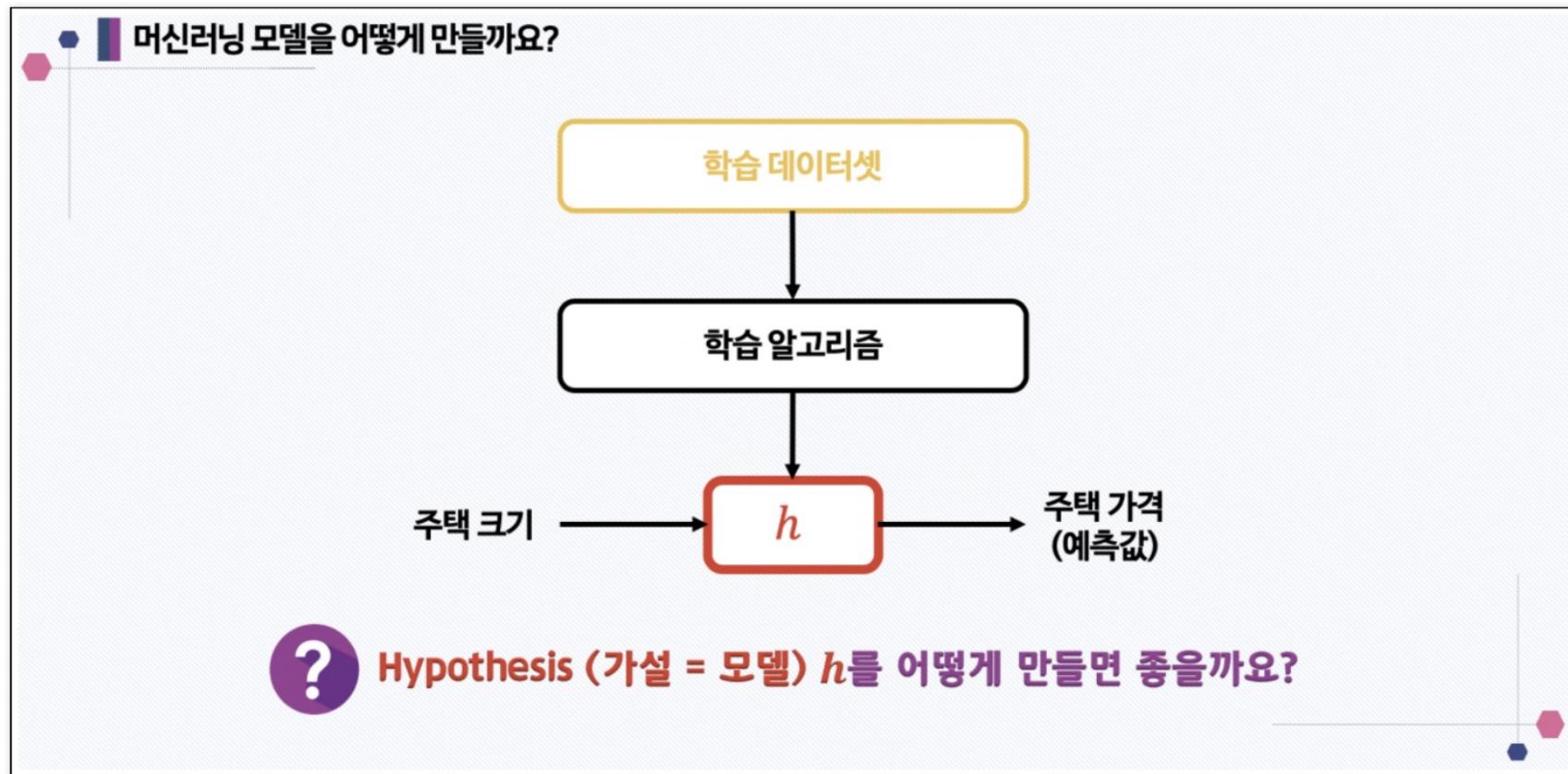
학습 데이터셋 (Training Dataset)

주택 규모 (m^2)	주택 가격 (백만원)
220	325
135	295
85	250
55	176
...	...

“ 학습 데이터 각각에 정답(주택 가격)이 주어져 있으므로 **지도학습**(Supervised Learning)이며, 주택 가격을 연속된 값으로 예측하는 것이므로 **회귀**(Regression) 문제임 ”

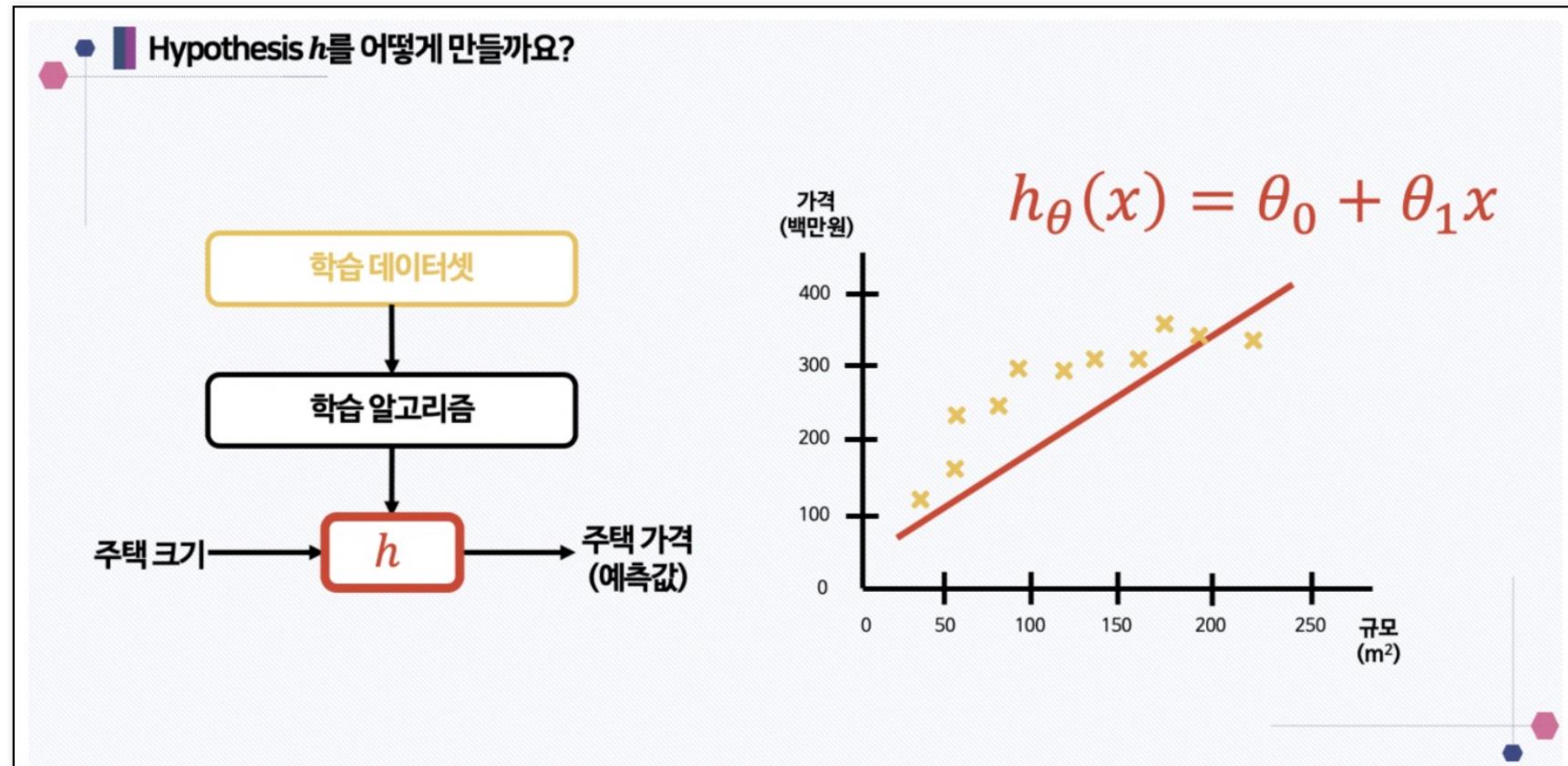
Regression? 회귀?

머신러닝 모델 만들기



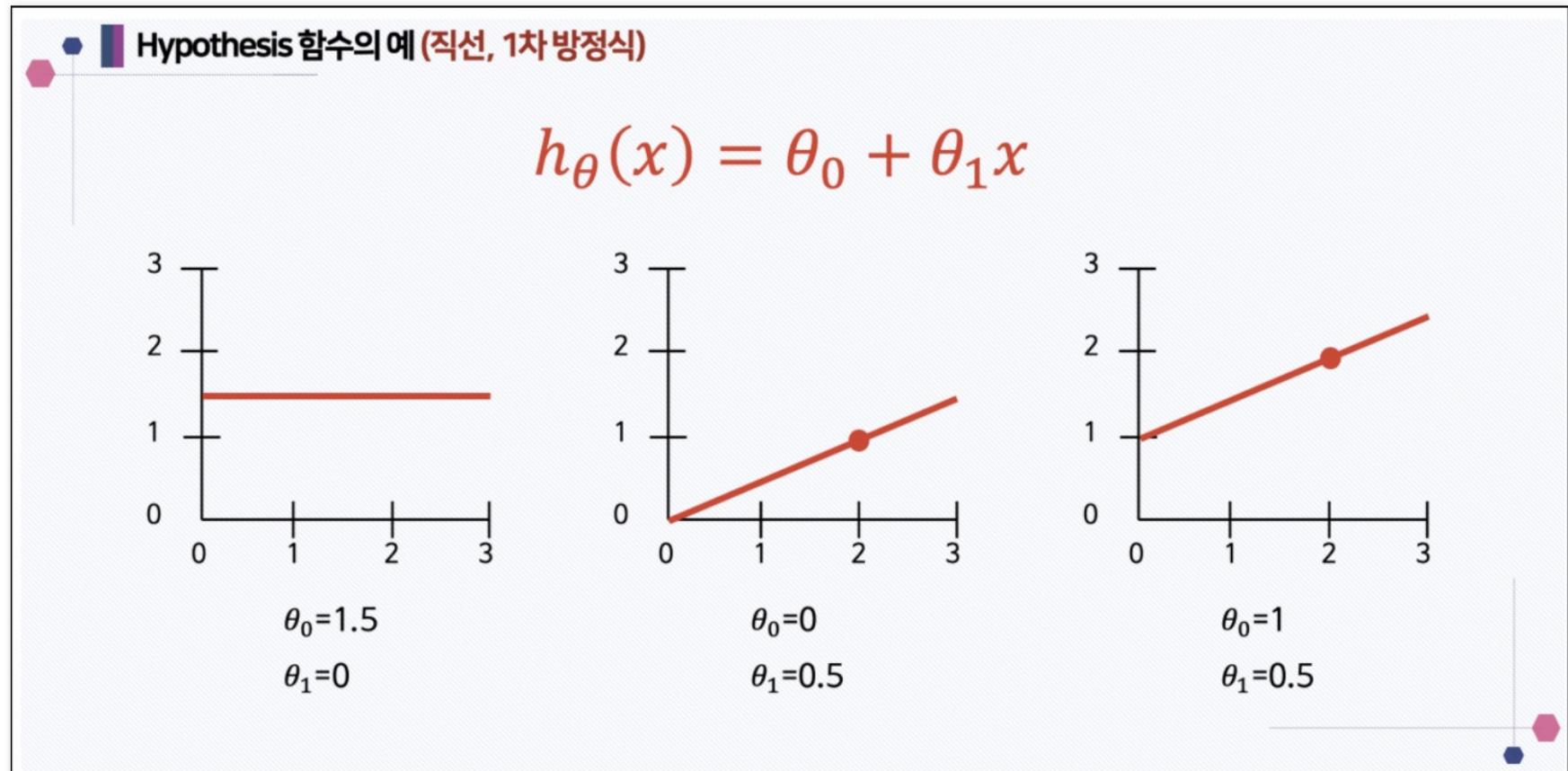
Regression? 회귀?

어떻게 만들까? 그~ 모델



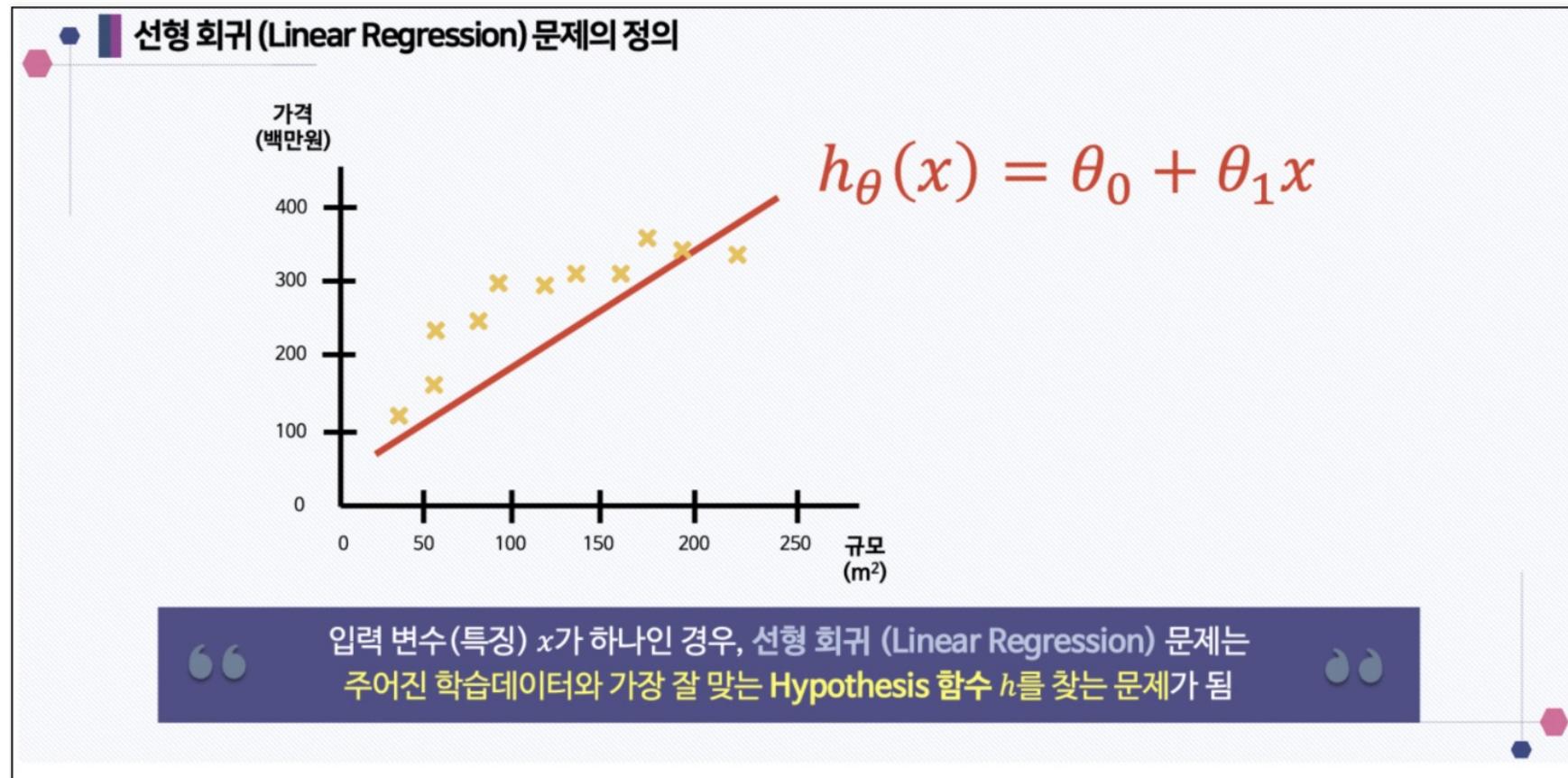
Regression? 회귀?

만약 1차 함수라면~



Regression? 회귀?

선형 회귀

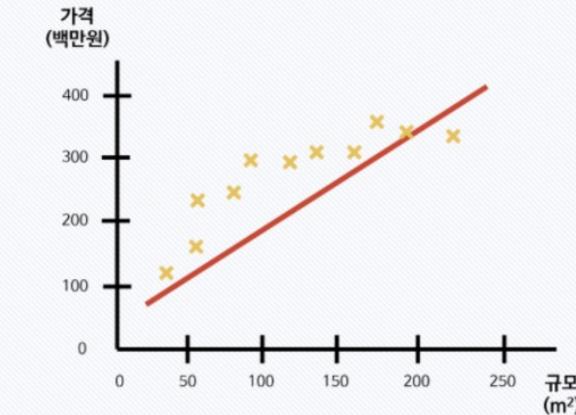


Regression? 회귀?

모델을 구성하는 파라미터를 어떻게 찾을까?

Hypothesis h 의 파라미터 θ_0 와 θ_1 을 어떻게 찾을까요?

주택 규모(m^2)	주택 가격(백만원)
$x^{(1)} = 220$	$y^{(1)} = 325$
$x^{(2)} = 135$	$y^{(2)} = 295$
$x^{(3)} = 85$	$y^{(3)} = 250$
$x^{(4)} = 55$	$y^{(4)} = 176$
...	...

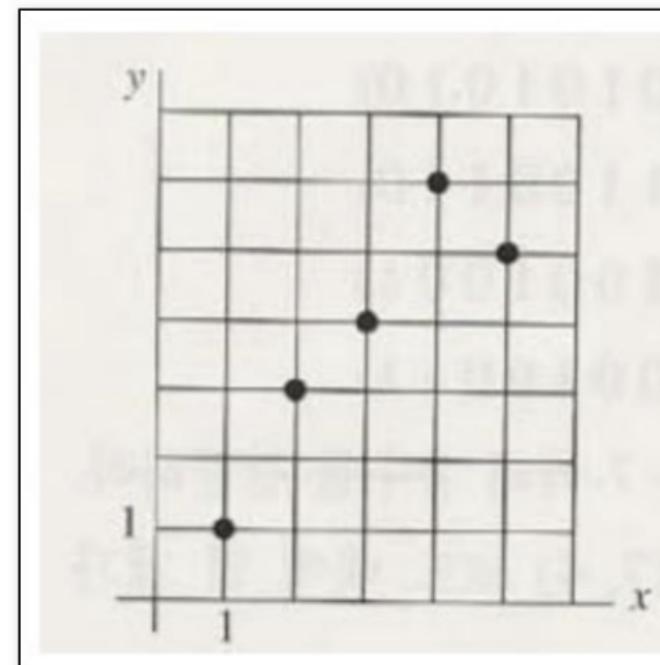


주어진 학습데이터 $x^{(i)}$ 에 대해 정답 $y^{(i)}$ 와 예측값 $h(x^{(i)})$ 의 차이가
최소가 되게 파라미터 θ_0 와 θ_1 의 값을 결정하면 어떨까요?

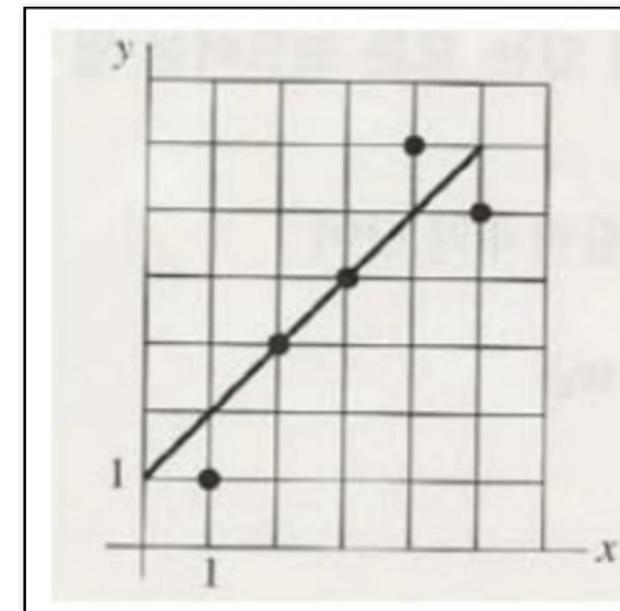
OLS : Ordinary Linear Least Square

OLS : Ordinary Linear
Least Square

이렇게 생긴 데이터를 하나의 직선으로 만든다면



이런 직선



OLS : Ordinary Linear
Least Square

데이터를 모두 직선에 대입

$$\begin{aligned}y_1 &= ax_1 + b \\y_2 &= ax_2 + b \\\vdots \\y_n &= ax_n + b\end{aligned}$$

- 우리가 찾고 싶은 건 a와 b

OLS : Ordinary Linear
Least Square

문제를 벡터와 행렬로 표현하면

$$\mathbf{Y} = \mathbf{AX}, \quad \text{여기서} \quad \mathbf{Y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \quad \mathbf{A} = \begin{pmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} a \\ b \end{pmatrix}$$

OLS : Ordinary Linear
Least Square

우리가 찾고 싶은 모델은

$$f(x) = ax + b$$

OLS : Ordinary Linear
Least Square

행렬로 정리하면

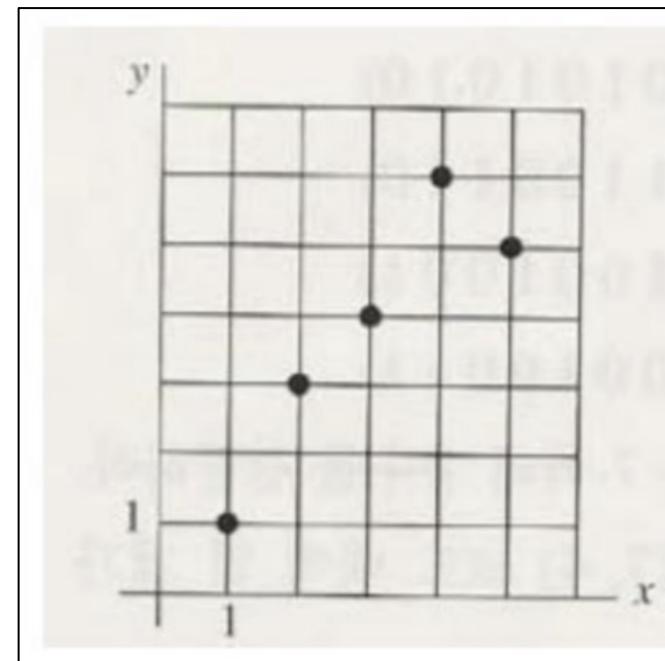
$$\mathbf{A}^T \mathbf{A} \mathbf{X} = \mathbf{A}^T \mathbf{Y}$$

OLS : Ordinary Linear
Least Square

드디어 X 를 찾을 수 있다

$$\mathbf{X} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{Y}$$

다시 본래의 문제



OLS : Ordinary Linear
Least Square

데이터는 이렇게

(1, 1), (2, 3), (3, 4), (4, 6), (5, 5)

OLS : Ordinary Linear
Least Square

원식은 이렇게

$$a + b = 1$$

$$2a + b = 3$$

$$3a + b = 4$$

$$4a + b = 6$$

$$5a + b = 5$$

OLS : Ordinary Linear
Least Square

정리하면

$$\mathbf{Y} = \begin{pmatrix} 1 \\ 3 \\ 4 \\ 6 \\ 5 \end{pmatrix} \quad \text{and} \quad \mathbf{A} = \begin{pmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 1 \\ 4 & 1 \\ 5 & 1 \end{pmatrix} \quad \text{we have} \quad \mathbf{A}^T \mathbf{A} = \begin{pmatrix} 55 & 15 \\ 15 & 5 \end{pmatrix}$$

OLS : Ordinary Linear
Least Square

적용하면 a와 b를 구할 수 있고

$$\begin{aligned}\mathbf{X} &= \begin{pmatrix} 55 & 15 \\ 15 & 5 \end{pmatrix}^{-1} \begin{pmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 1 \\ 4 & 1 \\ 5 & 1 \end{pmatrix}^T \begin{pmatrix} 1 \\ 3 \\ 4 \\ 6 \\ 5 \end{pmatrix} = \frac{1}{50} \begin{pmatrix} 5 & -15 \\ -15 & 55 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 3 \\ 4 \\ 6 \\ 5 \end{pmatrix} \\ &= \frac{1}{50} \begin{pmatrix} 5 & -15 \\ -15 & 55 \end{pmatrix} \begin{pmatrix} 68 \\ 19 \end{pmatrix} = \begin{pmatrix} 1.1 \\ 0.5 \end{pmatrix}\end{aligned}$$

OLS : Ordinary Linear
Least Square

최종 모델은 이렇게

$$y=1.1x+0.5$$

OLS : Ordinary Linear
Least Square

모델의 성능을 표현하자면

$$\begin{aligned} E &= [1 - f(1)]^2 + [3 - f(2)]^2 + [4 - f(3)]^2 + [6 - f(4)]^2 + [5 - f(5)]^2 \\ &= [1 - 1.6]^2 + [3 - 2.7]^2 + [4 - 3.8]^2 + [6 - 4.9]^2 + [5 - 6]^2 = 2.7. \end{aligned}$$

방금 내용 실습하기

전에 설치하기

- pip install statsmodels

OLS : Ordinary Linear
Least Square

데이터로 만들고

```
| import pandas as pd\n\ndata = {'x':[1., 2., 3., 4., 5.], 'y':[1., 3., 4., 6., 5.]}\n\ndf = pd.DataFrame(data)\n\ndf
```

	x	y
0	1.0	1.0
1	2.0	3.0
2	3.0	4.0
3	4.0	6.0
4	5.0	5.0

OLS : Ordinary Linear
Least Square

가설을 세워주고

```
import statsmodels.formula.api as smf

lm_model = smf.ols(formula="y ~ x", data=df).fit()
```

OLS : Ordinary Linear
Least Square

결과

```
| lm_model.params  
  
Intercept      0.5  
x              1.1  
dtype: float64
```

OLS : Ordinary Linear
Least Square

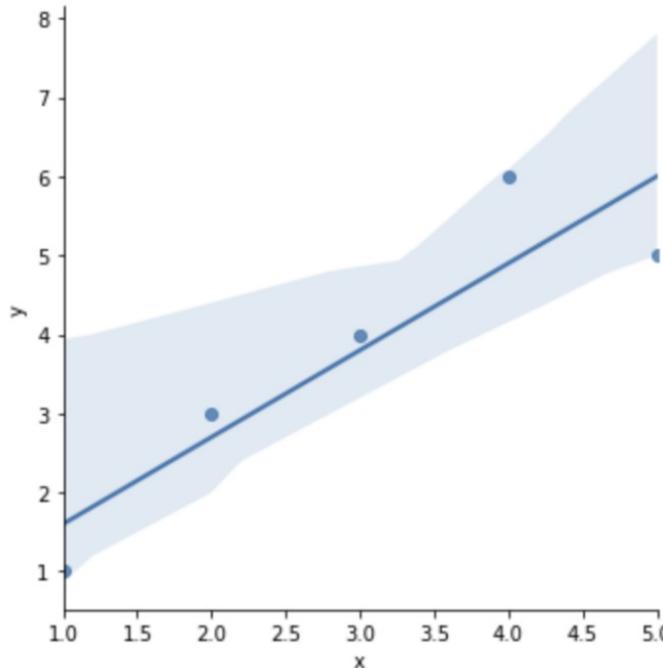
seaborn을 import하고

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

OLS : Ordinary Linear
Least Square

seaborn을 이용해서 plot

```
| sns.lmplot(x='x', y='y', data=df);
```



OLS : Ordinary Linear
Least Square

잔차 평가 residue

- 잔차는 평균이 0인 정규분포를 따르는 것 이어야 함
- 잔차 평가는 잔차의 평균이 0이고 정규분포를 따르는지 확인

OLS : Ordinary Linear
Least Square

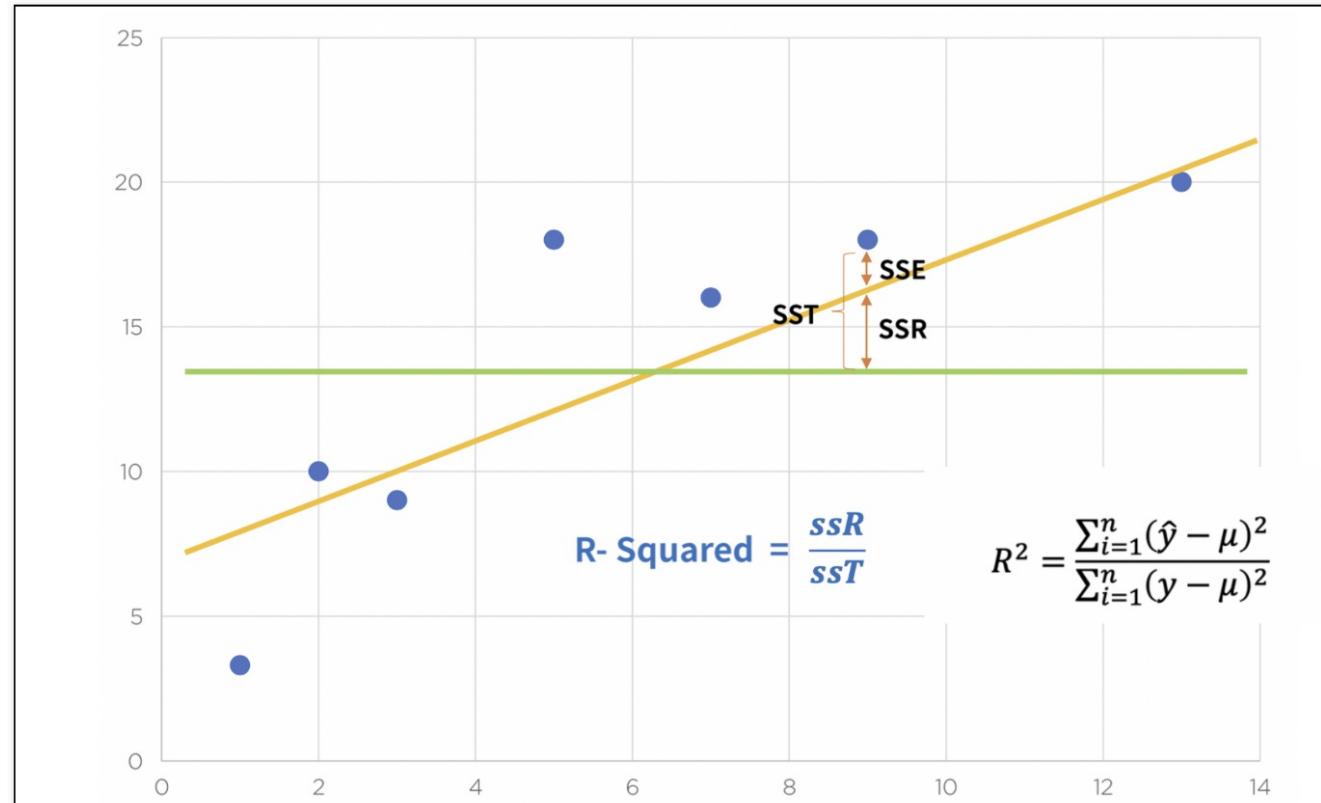
잔차 확인

```
| resid = lm_model.resid  
resid
```

```
0    -0.6  
1     0.3  
2     0.2  
3     1.1  
4    -1.0  
dtype: float64
```

OLS : Ordinary Linear
Least Square

결정계수 R-Squared



- $y_{\hat{}}$ 은 예측된 값
- 예측 값과 실제 값(y)이 일치하면 결정계수는 1이 됨 (즉 결정계수가 높을 수록 좋은 모델)

OLS : Ordinary Linear
Least Square

numpy로 직접 결정계수 계산

```
| import numpy as np  
  
mu = np.mean(df.y)  
y = df.y  
yhat = lm_model.predict()  
np.sum((yhat - mu)**2 / np.sum((y - mu)**2))
```

0.8175675675675682

OLS : Ordinary Linear
Least Square

뭐.. 간단하게 구할 수도

```
| lm_model.rsquared
```

```
0.8175675675675674
```

OLS : Ordinary Linear
Least Square

잔차의 분포도 확인

```
| sns.distplot(resid, color='black');
```

