

# 머신러닝 - 회귀

2021



# 1. 개요

## ❖ 분류 vs 회귀

Classification



Category 값  
(이산값)

Regression



숫자값  
(연속값)

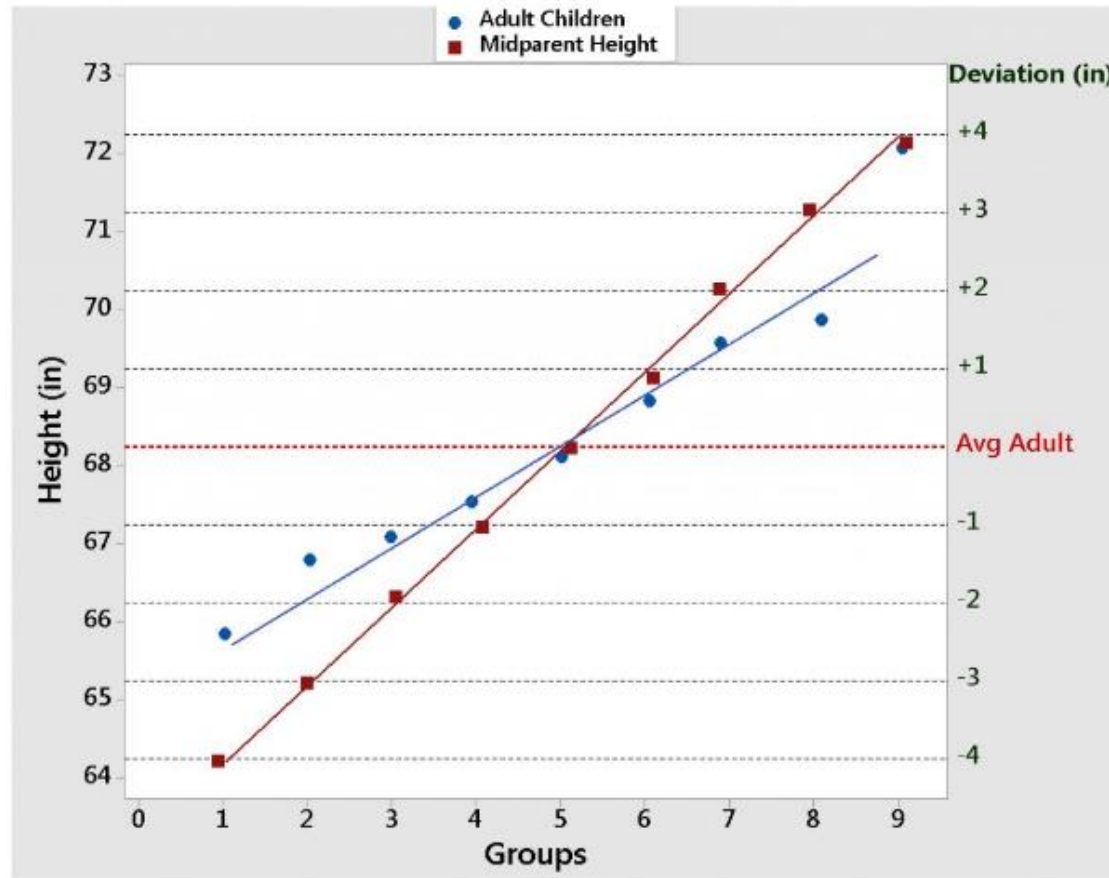
## ❖ 회귀 유형

독립변수 개수	회귀 계수의 결합
1개: 단일 회귀	선형: 선형 회귀
여러 개: 다중 회귀	비선형: 비선형 회귀

## 2. 단순 선형 회귀

### ❖ 아버지와 아들의 키

- Galton's Height Data(1885년) - 유전에 의하여 보통사람의 신장으로 회귀



※출처: 회귀분석의 유래 : 대체 왜 Regression(회귀)이라고 불릴까?

## 2. 단순 선형 회귀

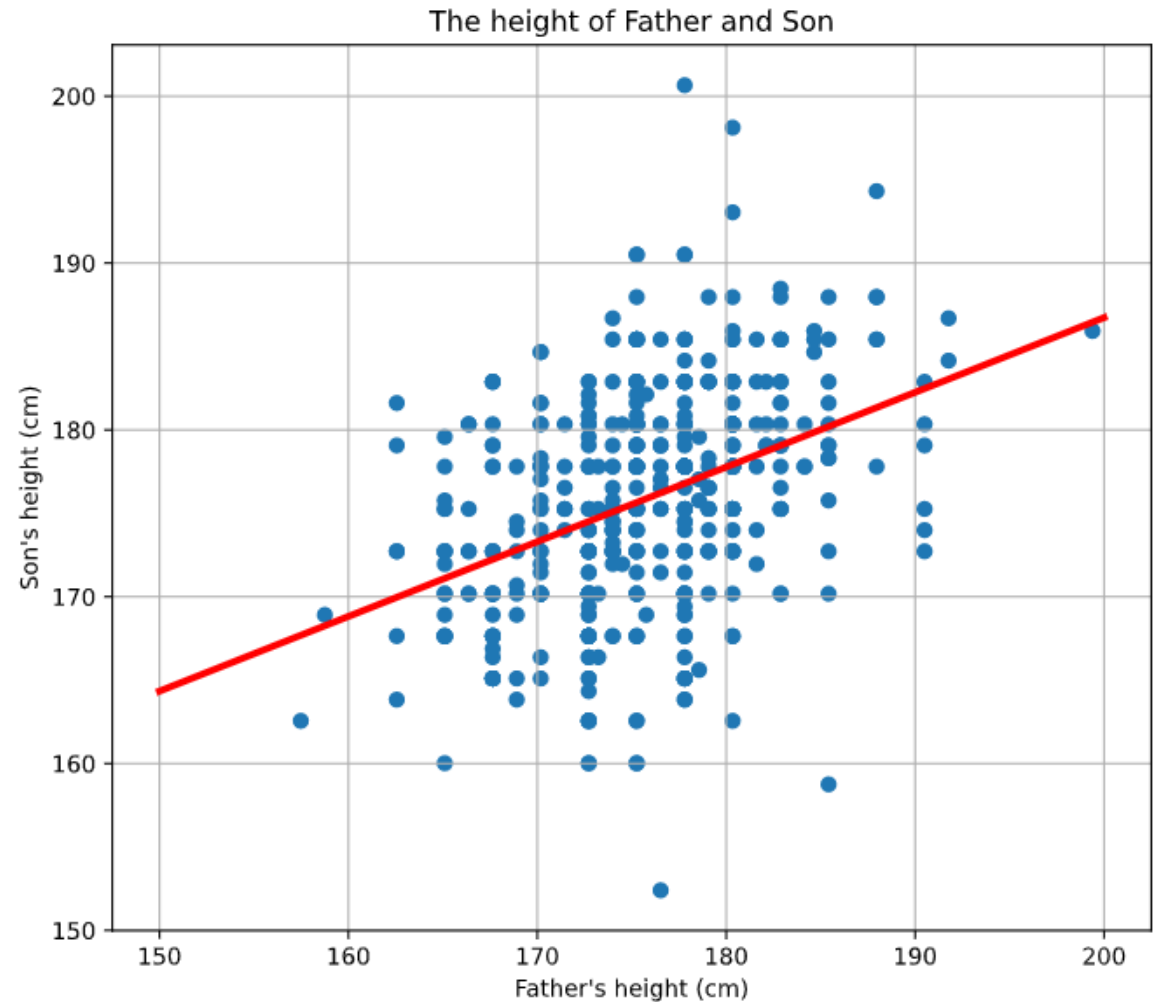
### ❖ 산점도와 선형 회귀선

- 데이터 소스: <http://www.randomservices.org/random/data/Galton.txt> (단위: 인치)

- 기울기: 0.4477

- 절편: 97.1776

- 잔차 제곱(RSS):  
17,556.60



## 2. 단순 선형 회귀

### ❖ 선형 회귀선의 기울기와 절편

- Numpy 최소자승법

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

df = pd.read_csv('http://www.randomservices.org/random/data/Galton.txt', sep='\t')
df = df[df.Gender == 'M']
height = df[['Father', 'Height']].rename(columns={'Height': 'Son'})
height.Father = height.Father * 2.54
height.Son = height.Son * 2.54

A = np.vstack([height.Father, np.ones(len(height.Father))]).T
reg = np.linalg.lstsq(A, height.Son, rcond=None)
m, c = reg[0]
rss = reg[1][0]
print(f'기울기: {m:.4f}, 절편: {c:.4f}, 잔차제곱: {rss:.2f}')

plt.figure(figsize=(8,7))
plt.scatter(height.Father, height.Son)
plt.plot([150, 200], [m*150+c, m*200+c], 'r', lw=3)
plt.title('The height of Father and Son')
plt.xlabel("Father's height (cm)"); plt.ylabel("Son's height (cm)")
plt.grid(); plt.show()
```

## 2. 단순 선형 회귀

### ❖ 선형 회귀선의 기울기와 절편

- Scikit-Learn

```
from sklearn.linear_model import LinearRegression
```

```
X = height.Father.values.reshape(-1,1)
```

```
y = height.Son.values
```

```
lr = LinearRegression()
```

```
lr.fit(X, y)
```

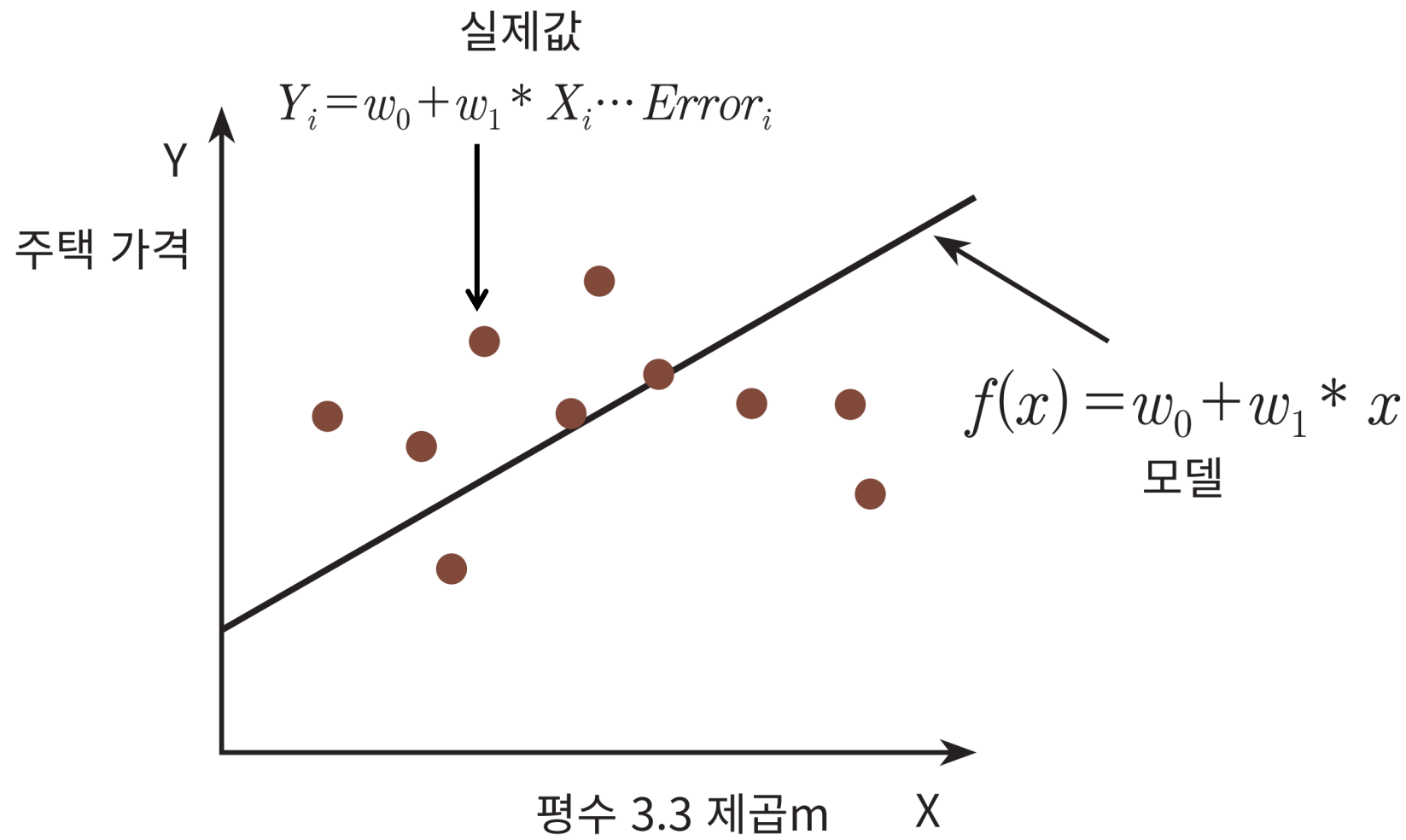
```
weight, bias, rss = lr.coef_, lr.intercept_, lr._residues
```

```
print(f'기울기: {weight[0]:.4f}, 절편: {bias:.4f}, 잔차제곱: {rss:.2f}')
```

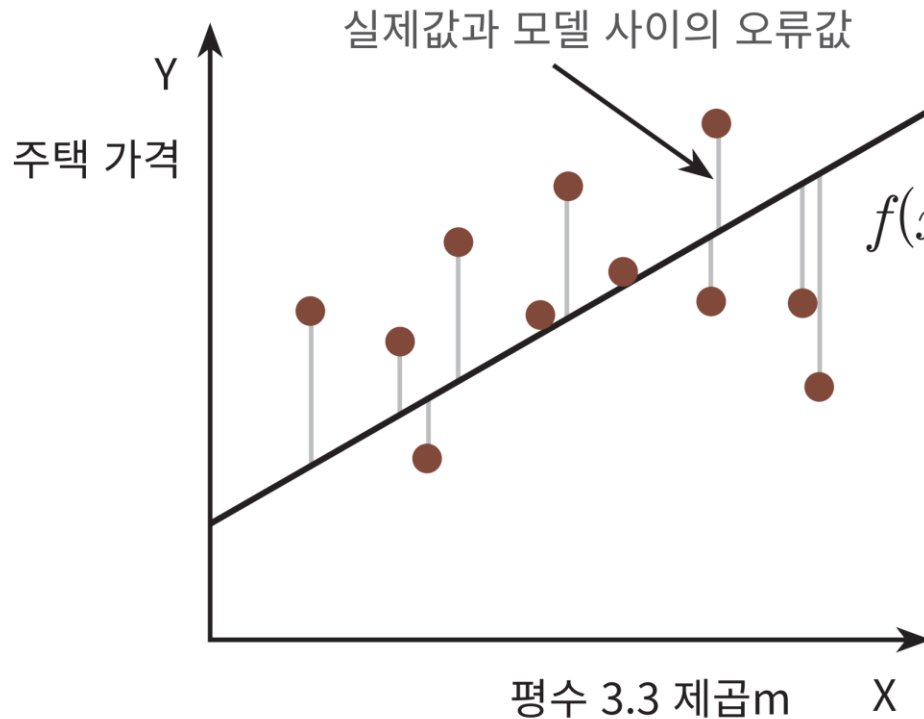
```
score = lr.score(X, y)
```

```
print(f'R_squared score: {score:.4f}')
```

## 2. 단순 선형 회귀



## 2. 단순 선형 회귀



$$\begin{aligned} \text{RSS} = & (\#1 \text{ 주택 가격} - (w_0 + w_1 \text{ \#1 주택크기}))^2 \\ & + (\#2 \text{ 주택 가격} - (w_0 + w_1 \text{ \#2 주택크기}))^2 \\ & + (\#3 \text{ 주택 가격} - (w_0 + w_1 \text{ \#3 주택크기}))^2 \\ & + \dots (\text{모든 학습 데이터에 대해 RSS 수행}) \end{aligned}$$

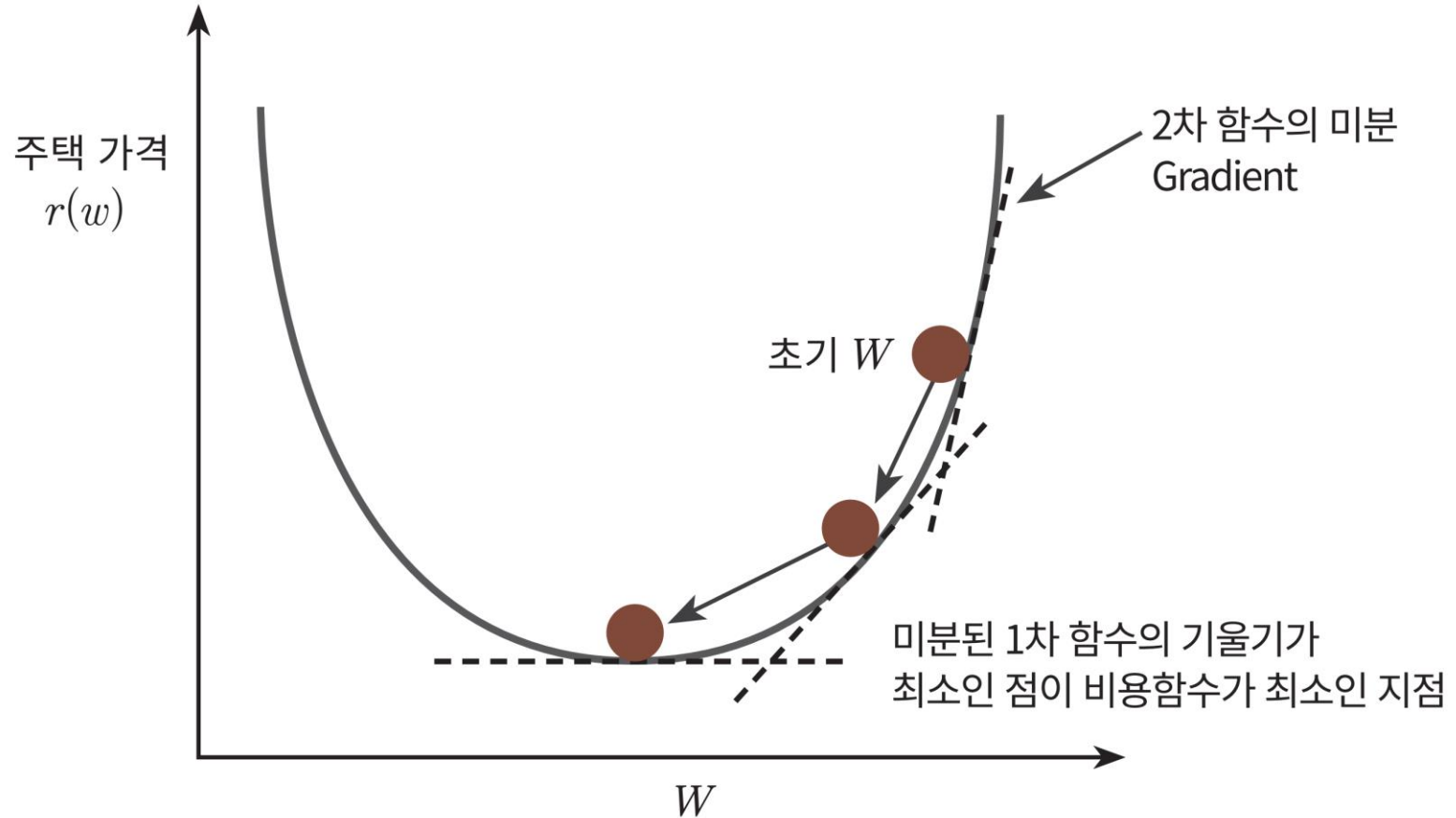
$$\text{RSS}(w_0, w_1) = \frac{1}{N} \sum_{i=1}^N (y_i - (w_0 + w_1 * x_i))^2$$

Residual Sum of Square  
잔차(표본집단의 오차) 제곱 합



### 3. 비용 최소화하기 – 경사 하강법(Gradient Descent)

#### ❖ 경사 하강법(Gradient Descent)

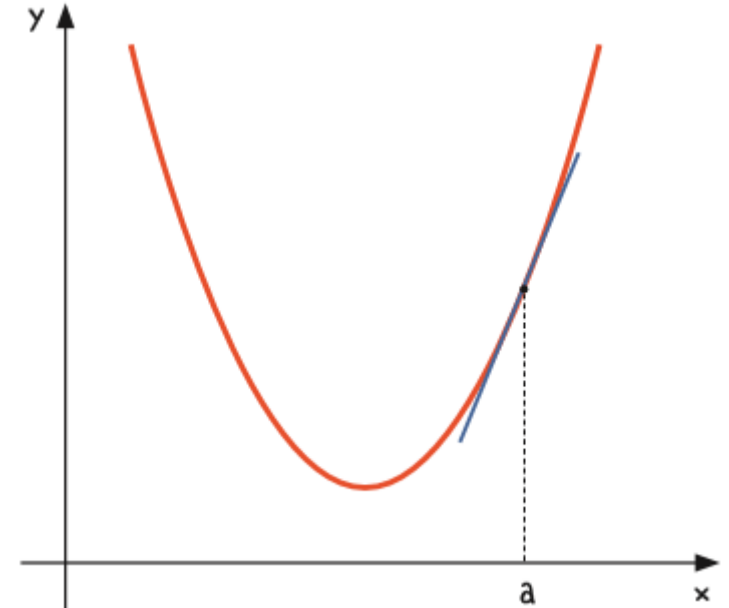


### 3. 비용 최소화하기 - 경사 하강법(Gradient Descent)

#### ❖ 미분의 개념

##### ▪ 순간 변화율의 의미

- $x$  값의 변화량이 0에 가까울 만큼 아주 미세하게 변화했다면,
- $y$  값의 변화 역시 아주 미세하게 변화했을 것
- 순간 변화율은 '어느 쪽'이라는 방향성을 지니고 있으므로 이 방향에 맞추어 직선을 그릴 수가 있음
- 이 선이 바로 이 점에서의 '기울기'라고 불리는 접선



a에서의 순간 변화율은 곧 기울기

### 3. 비용 최소화하기 – 경사 하강법(Gradient Descent)

#### ❖ 미분의 개념

##### ■ 미분이란?

- x 값이 아주 미세하게 움직일 때의 y 변화량을 구한 뒤,
- 이를 x의 변화량으로 나누는 과정
- 한 점에서의 순간 기울기

- “함수  $f(x)$ 를 미분하라”는  $\frac{d}{dx}f(x)$  라고 표기함.

$$\frac{d}{dx}f(x) = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

① 함수  $f(x)$ 를  $x$ 로 미분하라는 것은

②  $x$ 의 변화량이 0에 가까울 만큼 작을 때

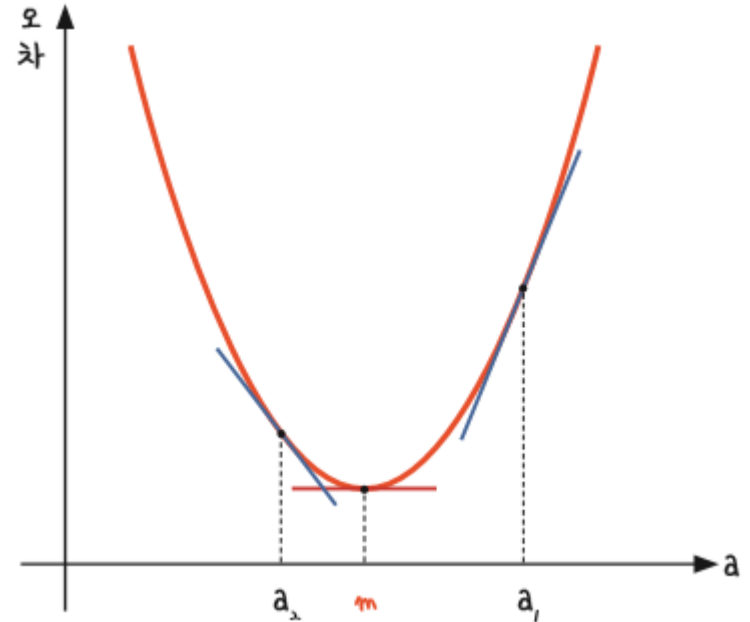
③  $y$  변화량의 차이를

④  $x$  변화량으로 나눈 값(= 순간 변화율)을 구하라는 뜻

### 3. 비용 최소화하기 - 경사 하강법(Gradient Descent)

#### ❖ 경사 하강법 개요

- $y = x^2$  그래프에서  $x$ 에  $a_1, a_2$   
그리고  $m$ 을 대입하여  
그 자리에서 미분하면  
그림처럼 각 점에서의 기울기가 그려짐
- 기울기가 0인 점이 최소값
- 따라서 우리가 할 일은  
'미분 값이 0인 지점'을 찾는 것!



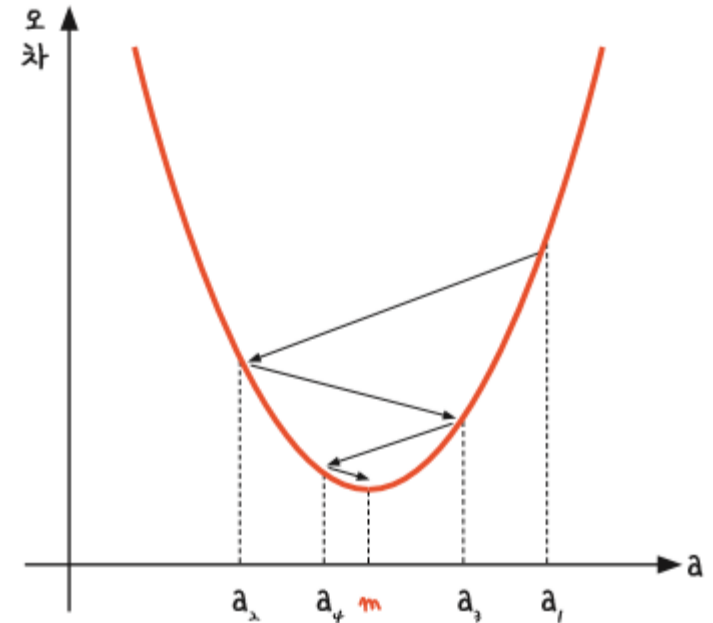
기울기가 0인 점이 최소값

### 3. 비용 최소화하기 - 경사 하강법(Gradient Descent)

#### ❖ 기울기가 0인 점을 찾는 방법

- 1)  $a_1$ 에서 미분값을 구한다.
- 2) 구해진 기울기의 반대 방향으로 얼마간 이동시킨  $a_2$ 에서 미분값을 구한다.
- 3) 위에서 구한 값이 0이 아니면  $a_2$ 에서 2)번 과정을 반복한다.
- 4) 그러면 그림처럼 이동 결과 한 점으로 수렴함

#### ❖ 경사 하강법은 이렇게 반복적으로 기울기 $a$ 를 변화시켜서 $m$ 의 값을 찾아내는 방법

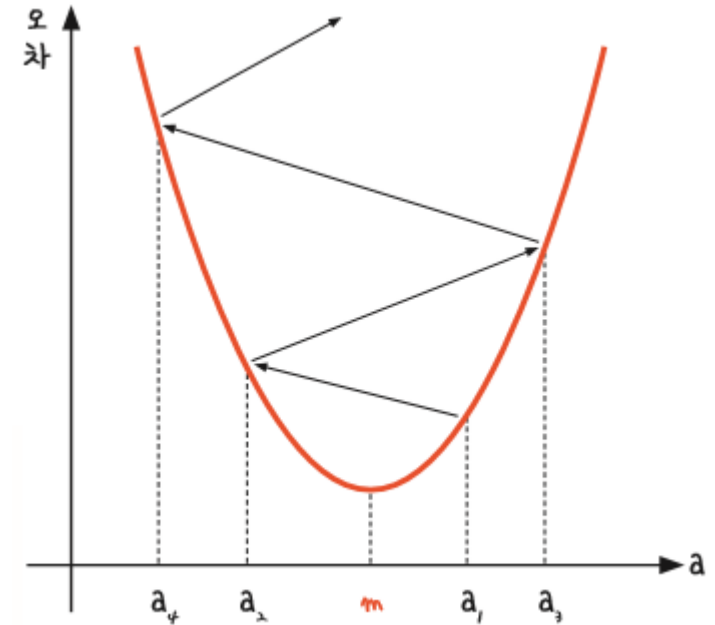


기울기가 0인 점  $m$ 을 찾는 방법

### 3. 비용 최소화하기 - 경사 하강법(Gradient Descent)

#### ❖ 학습률

- 기울기의 부호를 바꿔 이동시킬 때  
적절한 거리를 찾지 못해  
너무 멀리 이동시키면  
 $a$  값이 한 점으로 모이지 않고  
위로 치솟아 버림
- 어느 만큼 이동시킬지를 정해주는 것  
→ 학습률(Learning Rate)



학습률을 너무 크게 잡으면  
한 점으로 수렴하지 않고 발산함

## 4. Scikit-Learn 단순 선형회귀

### ❖ 당뇨병 데이터셋

```
diabetes = sklearn.datasets.load_diabetes()
```

### ❖ make\_regression()

#### ▪ 입력 파라미터

- n\_samples: 표본의 갯수(디폴트 100)
- n\_features: 독립변수(feature)의 갯수
- n\_targets: 종속변수(target)의 개수
- bias: 절편

#### ▪ 리턴 값

- X: [n\_samples, n\_features] 형상의 2차원 배열
- y: [n\_samples] 형상의 1차원 배열 또는  
[n\_samples, n\_targets] 형상의 2차원 배열

## 4. Scikit-Learn 단순 선형회귀

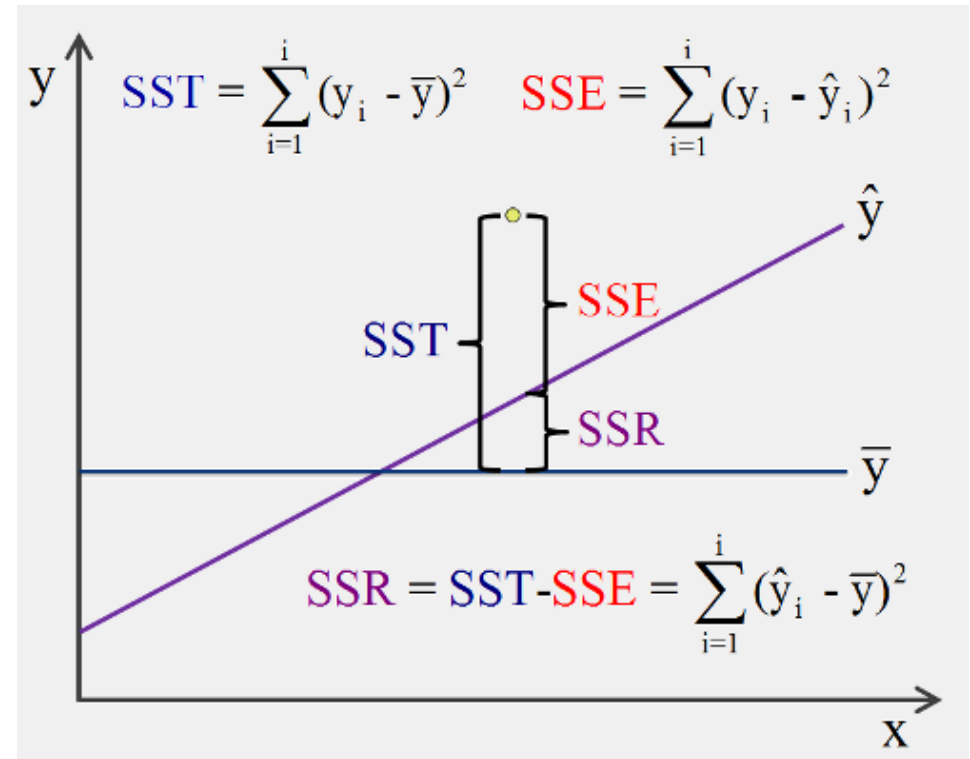
### ❖ 모델의 성능

- R-Squared(결정 계수)
  - 0 에서 1 사이의 값을 가짐
  - 1에 가까울수록 설명력이 높다.  
(모델이 데이터를 잘 설명해줌)

$$SST = SSR + SSE$$

$$R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST}$$

- SSE가 잔차제곱합(RSS)
- `r2_score( )`로 구함

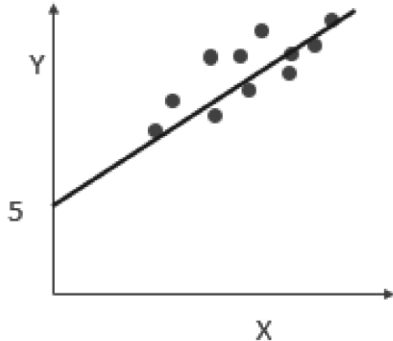
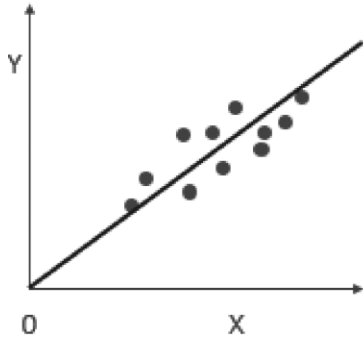


SST: Sum of Square Total  
SSR: Sum of Square Regression  
SSE: Sum of Square Error



## 5. 보스턴 주택 가격 예측

### ❖ LinearRegression 클래스

입력 파라미터	<p><b>fit_intercept</b>: 불린 값으로, 디폴트는 True입니다. Intercept(절편) 값을 계산할 것인지 말지를 지정합니다. 만일 False로 지정하면 intercept가 사용되지 않고 0으로 지정됩니다.</p> <div data-bbox="772 432 1655 853"><div><p>fit_intercept=True</p></div><div><p>fit_intercept=False</p></div></div> <p><b>normalize</b>: 불린 값으로 디폴트는 False입니다. fit_intercept가 False인 경우에는 이 파라미터가 무시됩니다. 만일 True이면 회귀를 수행하기 전에 입력 데이터 세트를 정규화합니다.</p>
속성	<p><b>coef_</b>: fit() 메서드를 수행했을 때 회귀 계수가 배열 형태로 저장하는 속성. Shape는 (Target 값 개수, 피쳐 개수).</p> <p><b>intercept_</b>: intercept 값</p>

## 5. 보스턴 주택 가격 예측

### ❖ 회귀 평가 지표

평가 지표	설명	수식
MAE	Mean Absolute Error(MAE)이며 실제 값과 예측값의 차이를 절댓값으로 변환해 평균한 것입니다.	$MAE = \frac{1}{n} \sum_{i=1}^n  Y_i - \hat{Y}_i $
MSE	Mean Squared Error(MSE)이며 실제 값과 예측값의 차이를 제곱해 평균한 것입니다.	$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$
RMSE	MSE 값은 오류의 제곱을 구하므로 실제 오류 평균보다 더 커지는 특성이 있으므로 MSE에 루트를 씌운 것이 RMSE(Root Mean Squared Error)입니다.	$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2}$
R <sup>2</sup>	분산 기반으로 예측 성능을 평가합니다. 실제 값의 분산 대비 예측값의 분산 비율을 지표로 하며, 1에 가까울수록 예측 정확도가 높습니다.	$R^2 = \frac{\text{예측값 Variance}}{\text{실제값 Variance}}$
평가 방법	사이킷런 평가 지표 API	Scoring 함수 적용 값
MAE	metrics.mean_absolute_error	'neg_mean_absolute_error'
MSE	metrics.mean_squared_error	'neg_mean_squared_error'
R <sup>2</sup>	metrics.r2_score	'r2'

## 5. 보스턴 주택 가격 예측

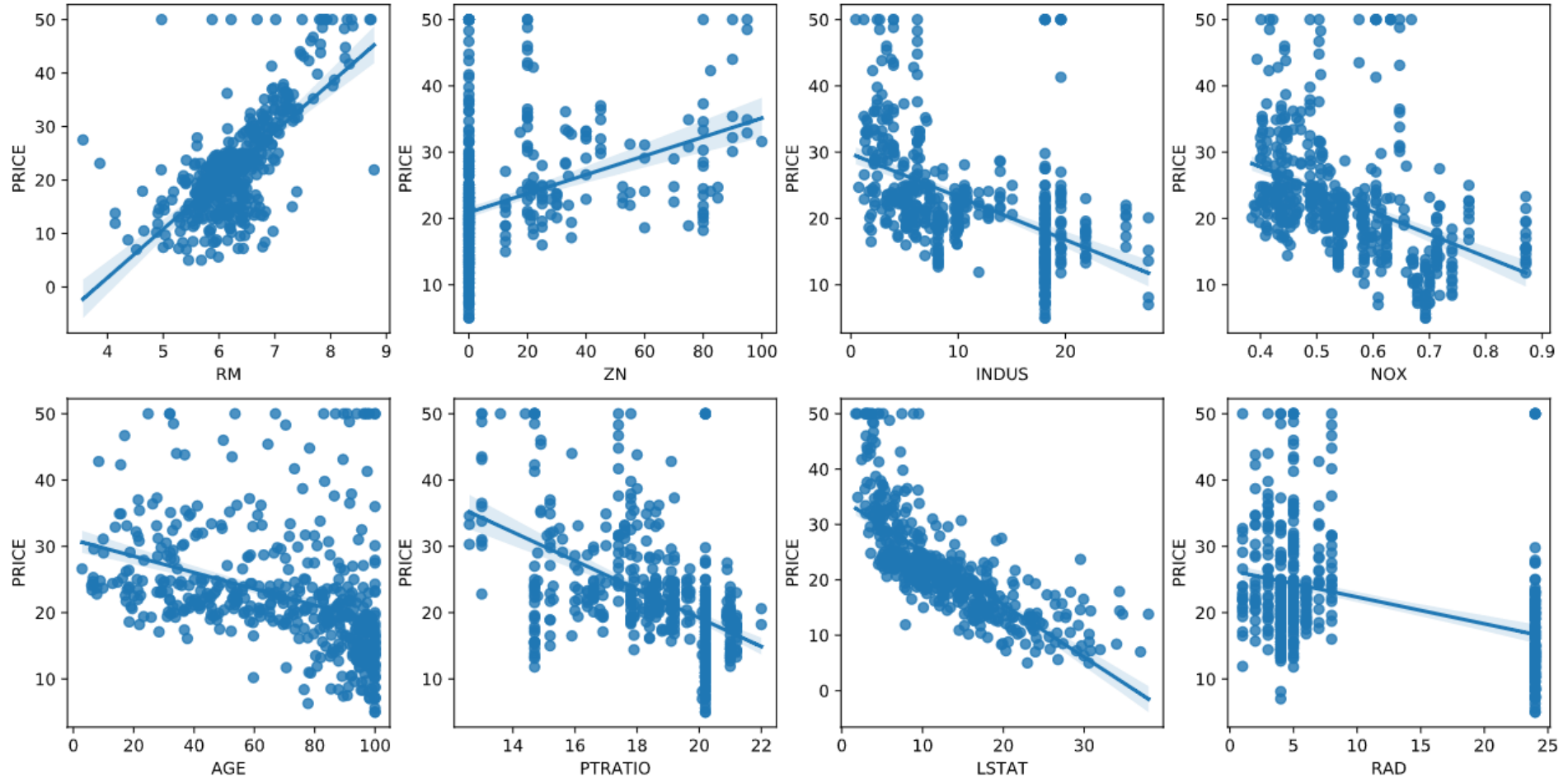
### ❖ 보스턴 주택 가격



- CRIM: 지역별 범죄 발생률
- ZN: 25,000평방피트를 초과하는 거주 지역의 비율
- INDUS: 비상업 지역 넓이 비율
- CHAS: 찰스강에 대한 더미 변수(강의 경계에 위치한 경우는 1, 아니면 0)
- NOX: 일산화질소 농도
- RM: 거주할 수 있는 방 개수
- AGE: 1940년 이전에 건축된 소유 주택의 비율
- DIS: 5개 주요 고용센터까지의 가중 거리
- RAD: 고속도로 접근 용이도
- TAX: 10,000달러당 재산세율
- PTRATIO: 지역의 교사와 학생 수 비율
- B: 지역의 흑인 거주 비율
- LSTAT: 하위 계층의 비율
- MEDV: 본인 소유의 주택 가격(중앙값)

## 5. 보스턴 주택 가격 예측

### ❖ 산점도와 선형 회귀 직선



## 5. 보스턴 주택 가격 예측

### ❖ 선형 회귀 결과

- 절편: intercept\_ 속성

40.99559517216412

- 회귀 계수: coef\_ 속성

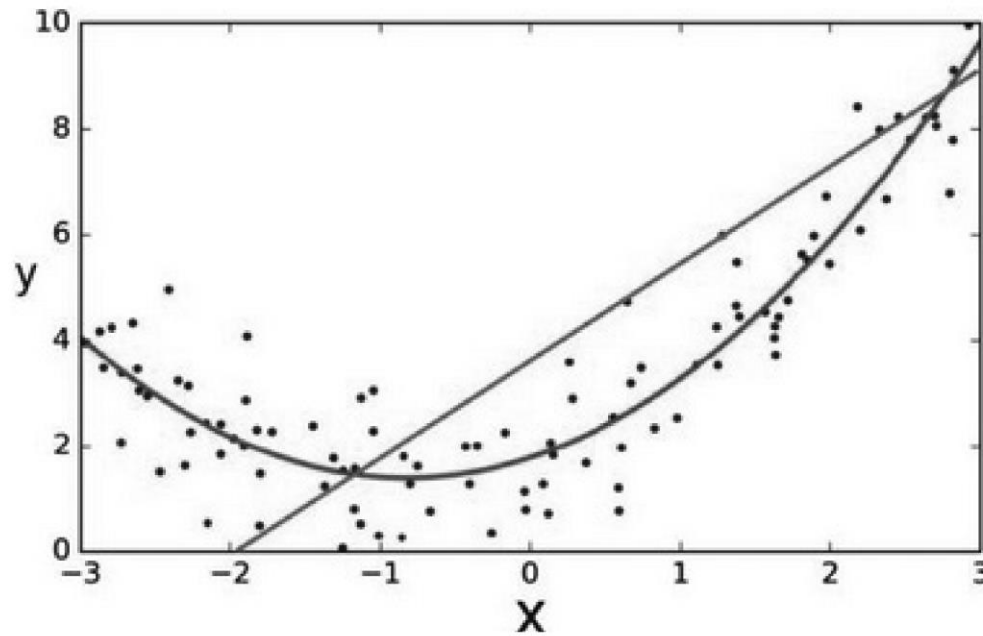
[ -0.1 0.1 0. 3. -19.8 3.4 0. -1.7 0.4 -0. -0.9 0. -0.6]

- 회귀식

$$y = -0.1*CRIM + 0.1*ZN + 0*INDUS + 3*CHAS - 19.8*NOX + 3.4*RM + 0*AGE + \\ -1.7*DIS + 0.4*RAD - 0*TAX - 0.9*PTRATIO + 0*B - 0.6*LSTAT + 41$$

## 6. 다항 회귀와 과(대)적합/과소적합

### ❖ 다항 회귀

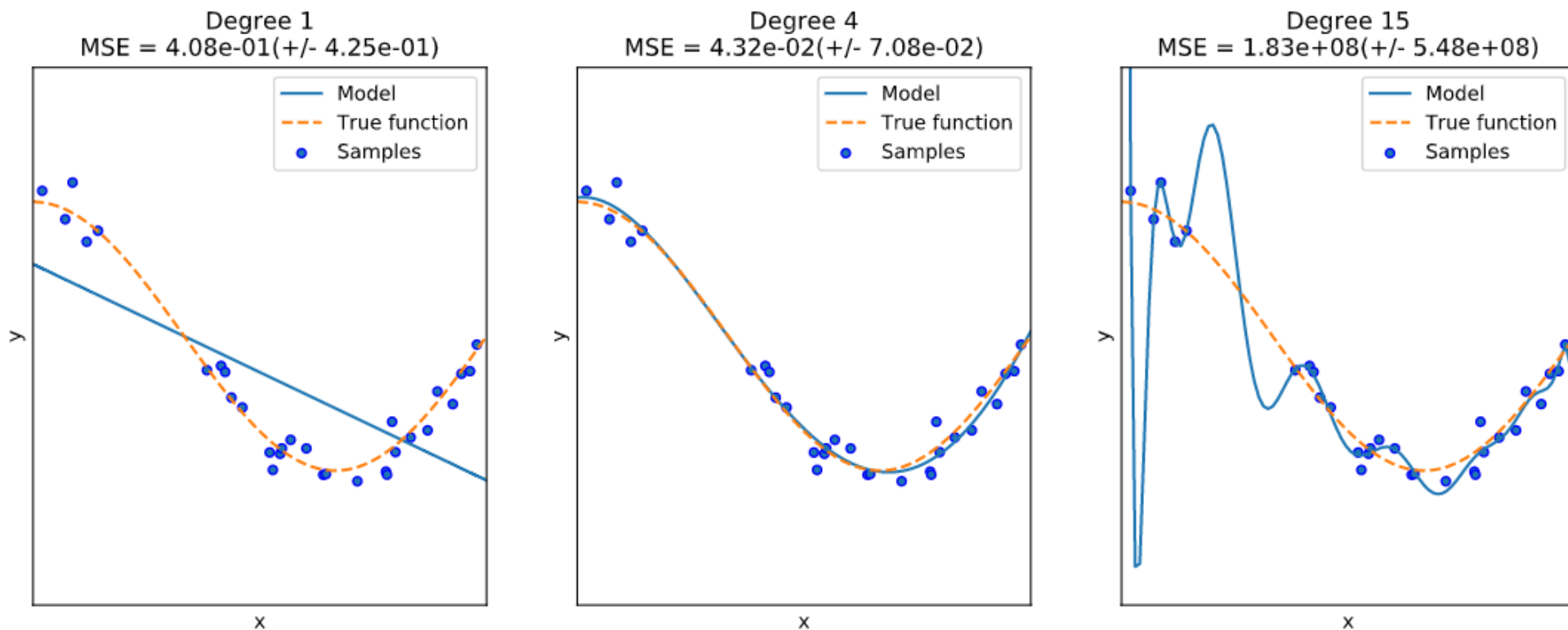


〈 주어진 데이터 세트에서 다항 회귀가 더 효과적임 〉

- 2차식:  $[x_1, x_2] \rightarrow [1, x_1, x_2, x_1^2, x_1x_2, x_2^2]$

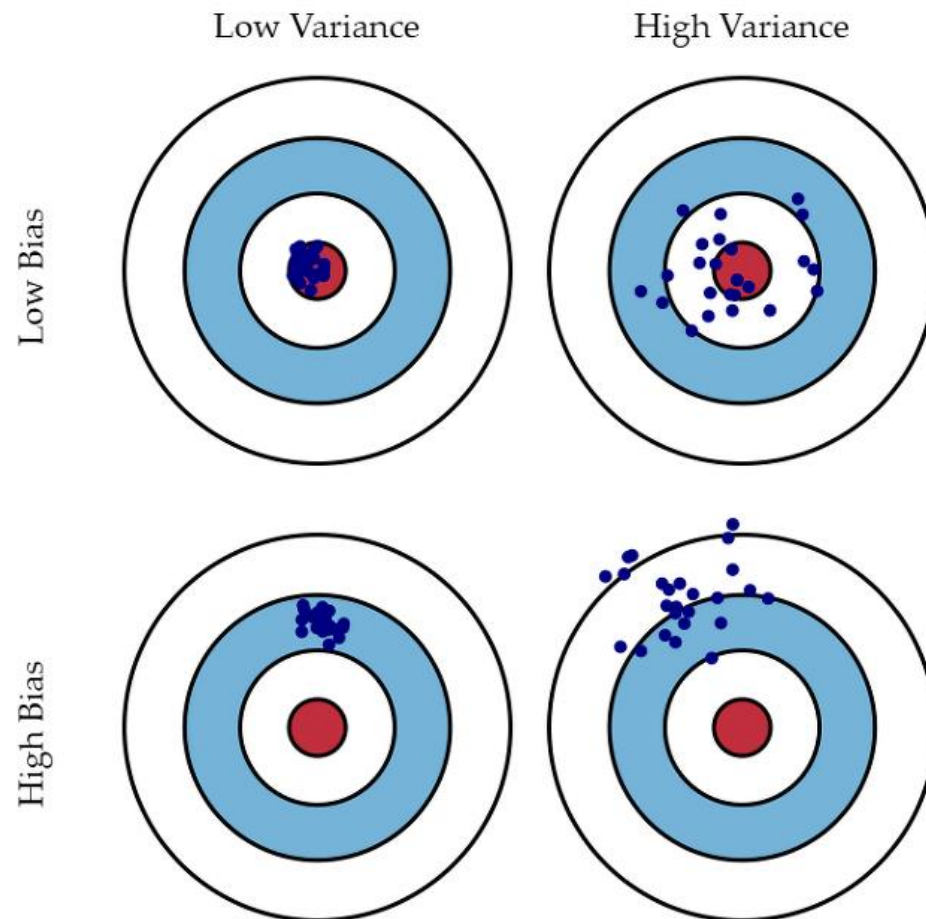
## 6. 다항 회귀와 과(대)적합/과소적합

### ❖ 과(대)적합/과소적합



## 6. 다항 회귀와 과(대)적합/과소적합

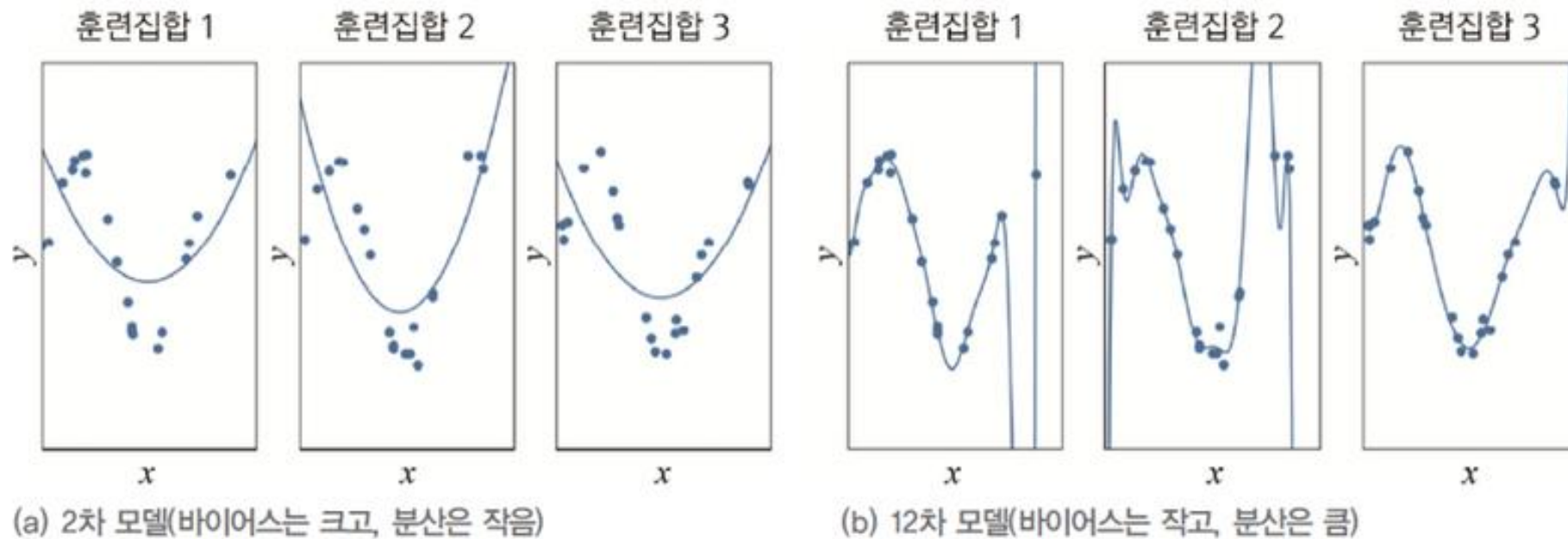
### ❖ 편향-분산 트레이드오프





## 6. 다항 회귀와 과(대)적합/과소적합

### ❖ 편향-분산 트레이드오프



## 7. 다양한 회귀

### ❖ 대표적인 선형회귀 모델

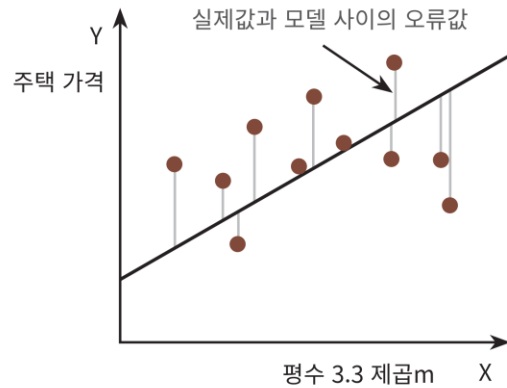
- 일반 선형 회귀
- Ridge 회귀 : 선형 회귀에 L2 규제 적용
- Lasso 회귀 : 선형 회귀에 L1 규제 적용
- ElasticNet 회귀 : L2, L1 규제를 결합한 모델
- Logistic 회귀 : 분류에 사용되는 선형 모델

### ❖ 규제의 종류

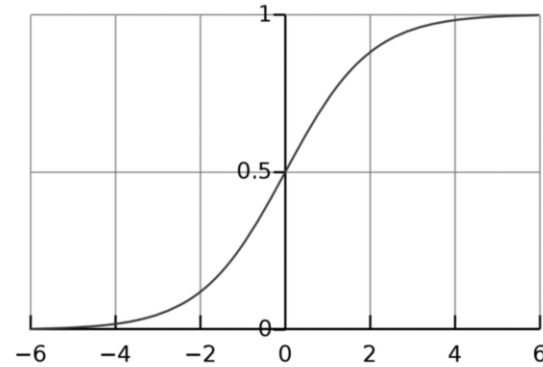
- L2 규제 : 상대적으로 큰 회귀 계수 값의 예측 영향도를 감소시키기 위해  
회귀 계수값을 더 작게 만드는 규제
- L1 규제 : 예측 영향력이 적은 feature의 회귀 계수를 0으로 만들어  
회귀 예측시 피처가 선택되지 않도록 하는 것  
Feature 선택 기능

## 8. 로지스틱 회귀

### ■ 선형 회귀 방식을 분류에 적용한 알고리즘



〈 선형 회귀의 선형 함수 〉



〈 로지스틱 회귀의 시그모이드 함수 〉

