*Machine Learning*

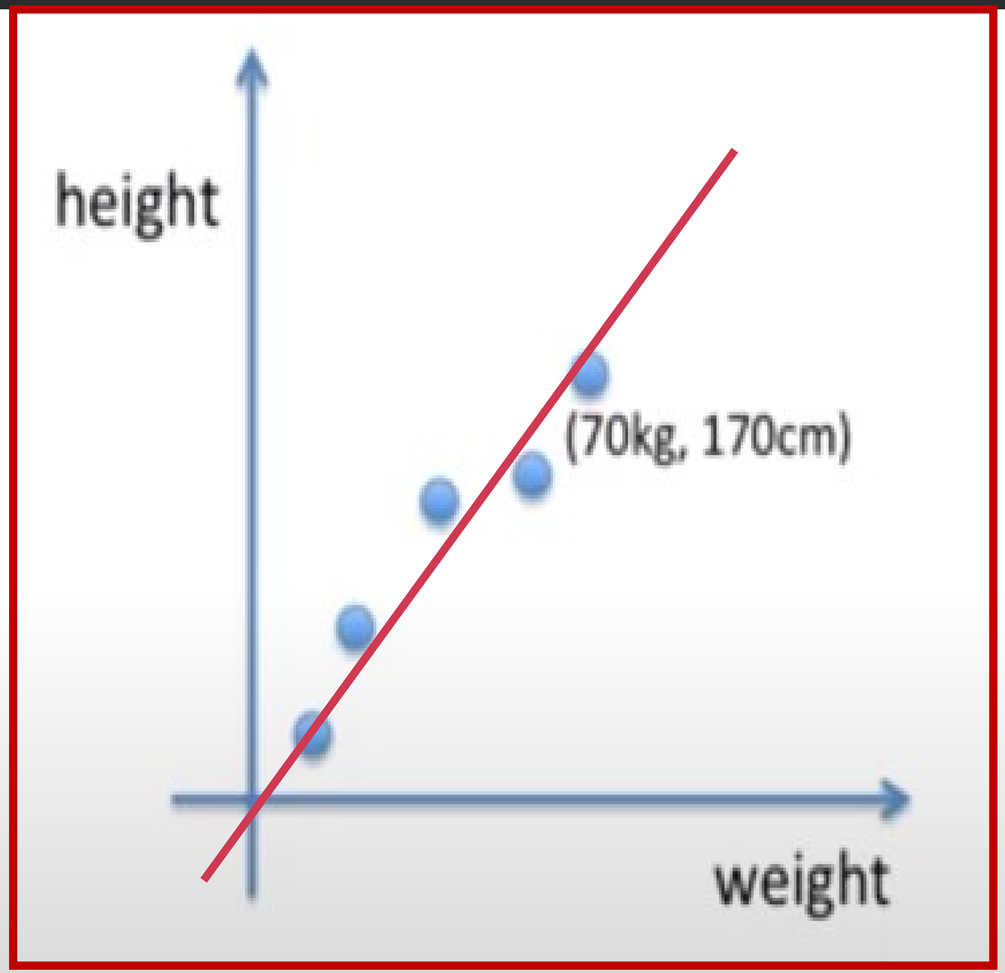
DATA KUBWA

# *SUPERVISED LEARNING*

# Regression

# Supervised Learning

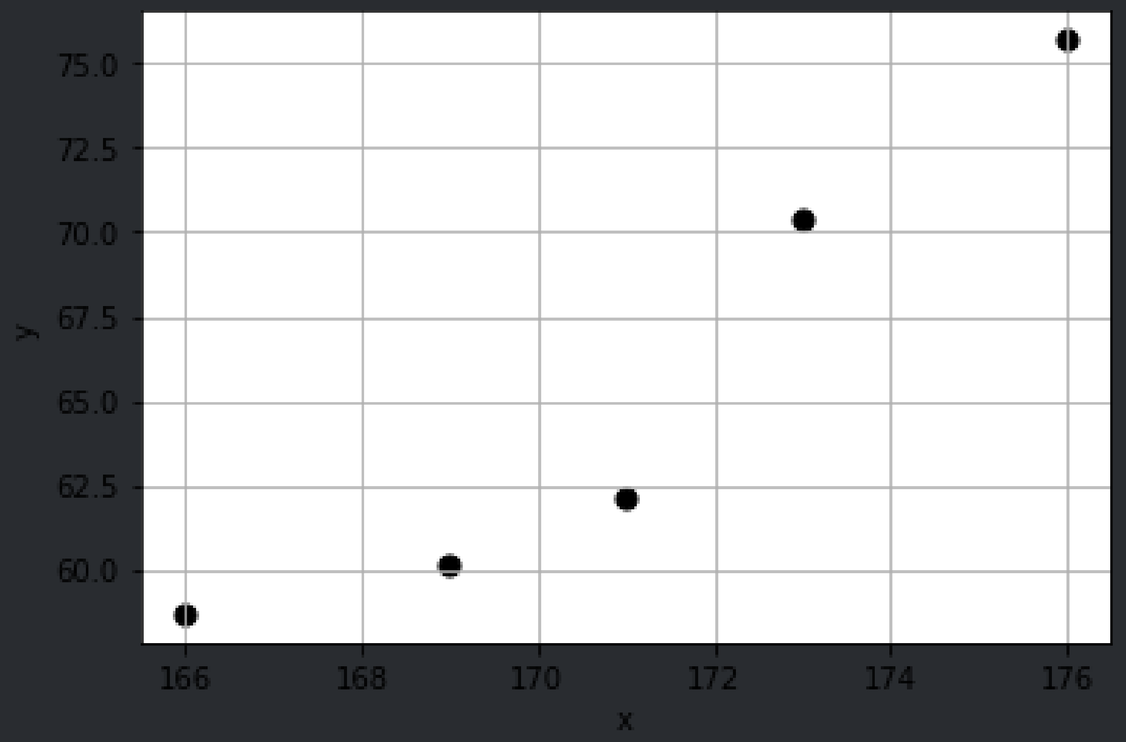

classify input into categorical output




how tall is he if his weight is 80kg?

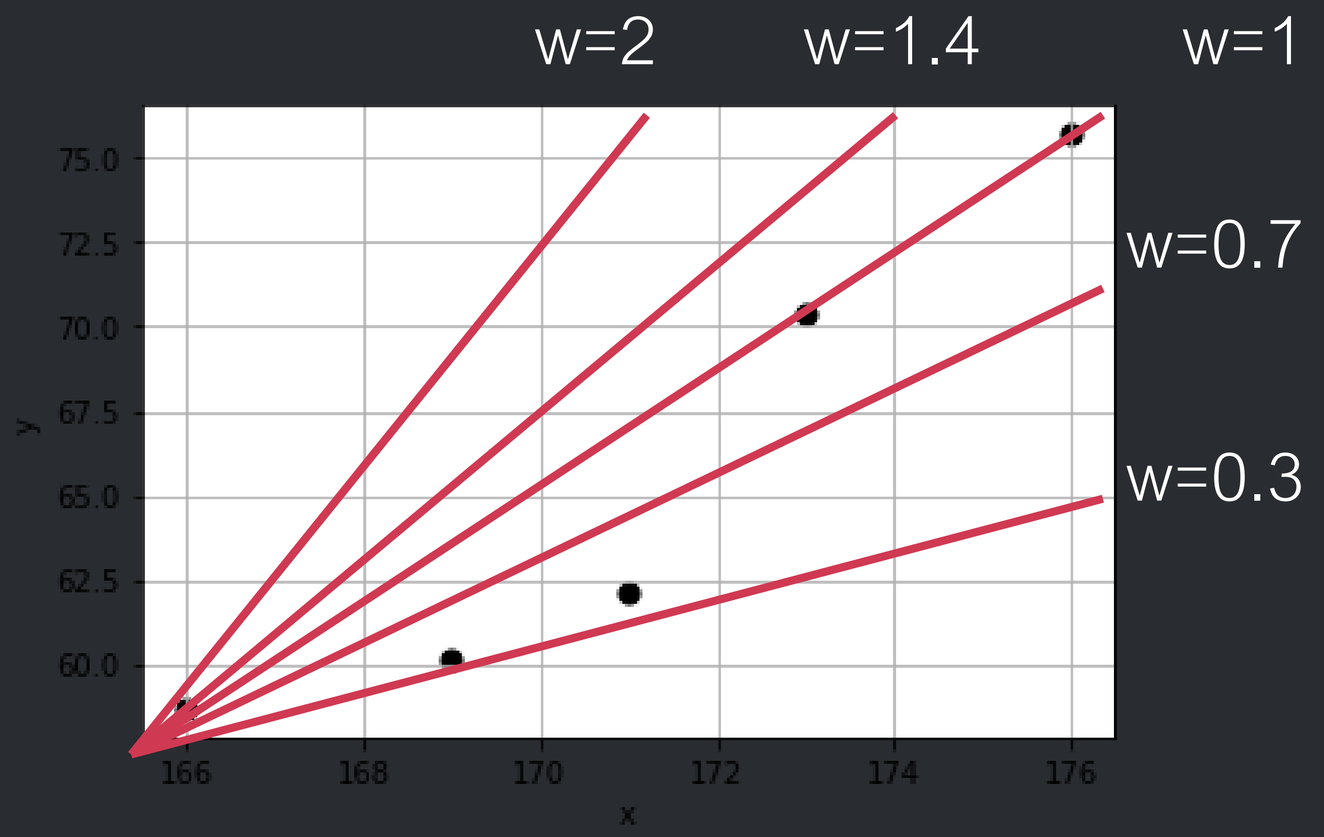
$$y = w_1 x_1$$

회귀 분석은 최적의 w를 찾는 것




$$y = wx$$

w는 무엇?

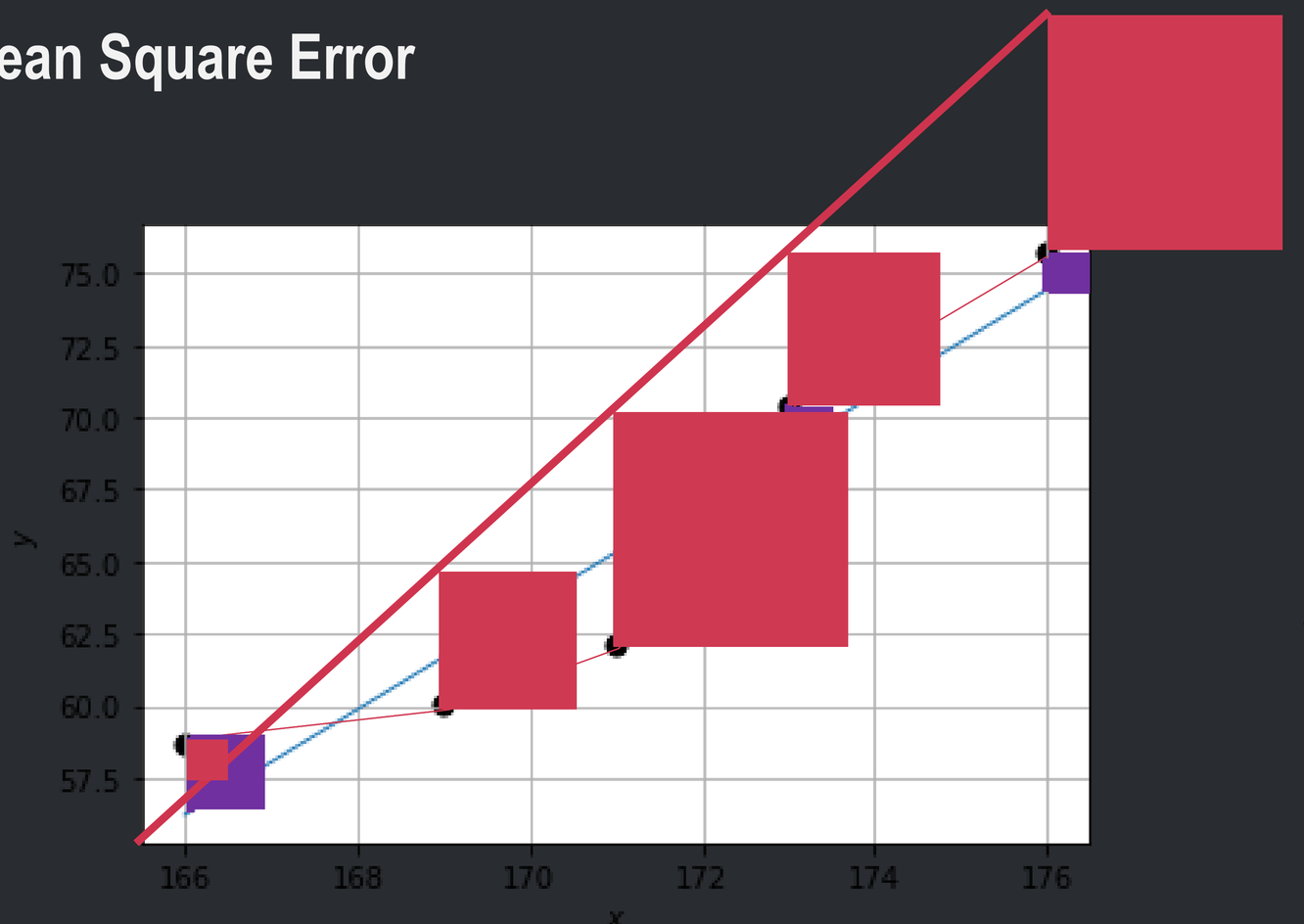

w=2
w=1.4
w=1
w=0.7
w=0.3
75.0
72.5
70.0
67.5
65.0
62.5
60.0
y
166
168
170
172
174
176
x

$$y = wx$$

## 에러제곱의 합: Square Error

## 에러제곱의 합의 평균: Mean Square Error

$y - \hat{y}$




1. 절대값
2. 제곱

y1 – ŷ1 + y/2 – ŷ2 + y3 – ŷ3 + y4 – ŷ4 + y5 – ŷ5

\+ - - + +

→ MIN, w

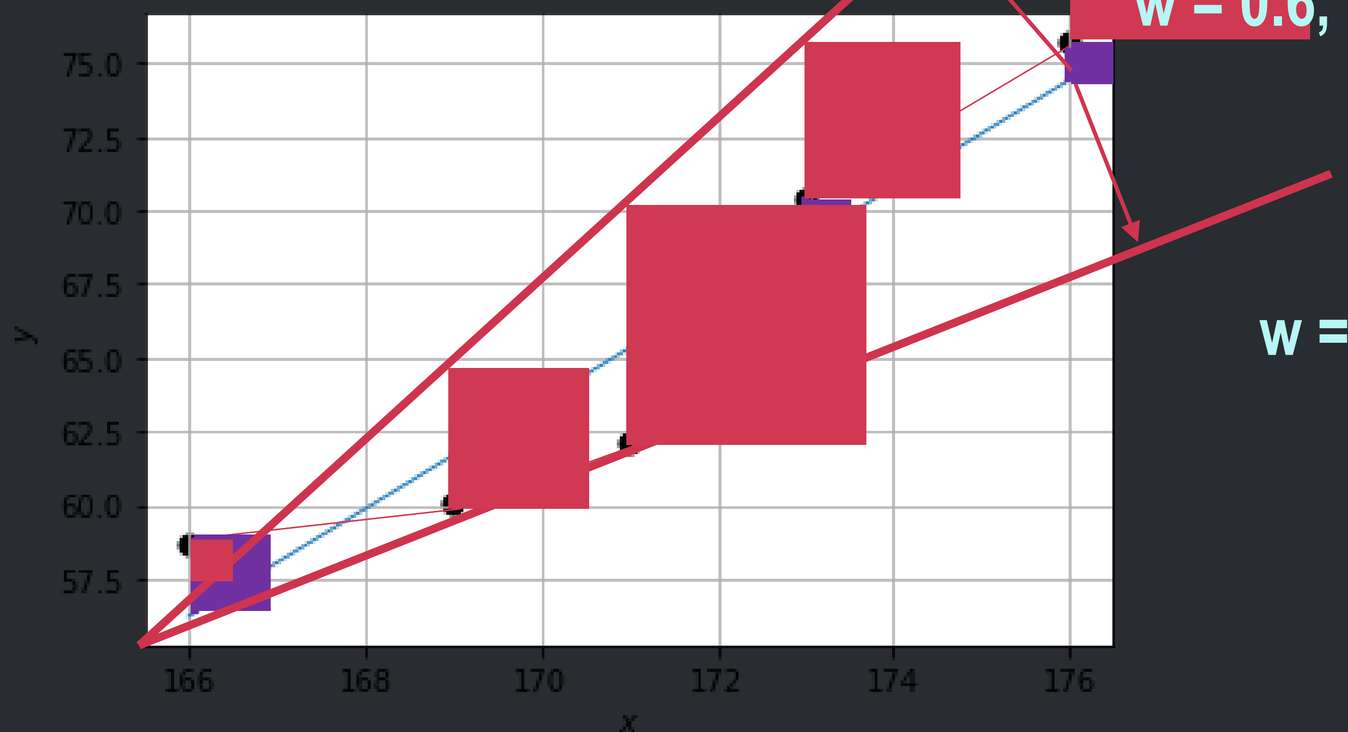
Mean Square Error = Cost function, RSS(w)

Cost function, RSS(w)의 최소값을 찾는 것

에러제곱의 합: Square Error

에러제곱의 합의 평균: Mean Square Error

$y - \hat{y}$




w = 0.9, RSS(0.9)

w = 0.6, RSS(w) → min

w = 0.3, RSS(0.3)

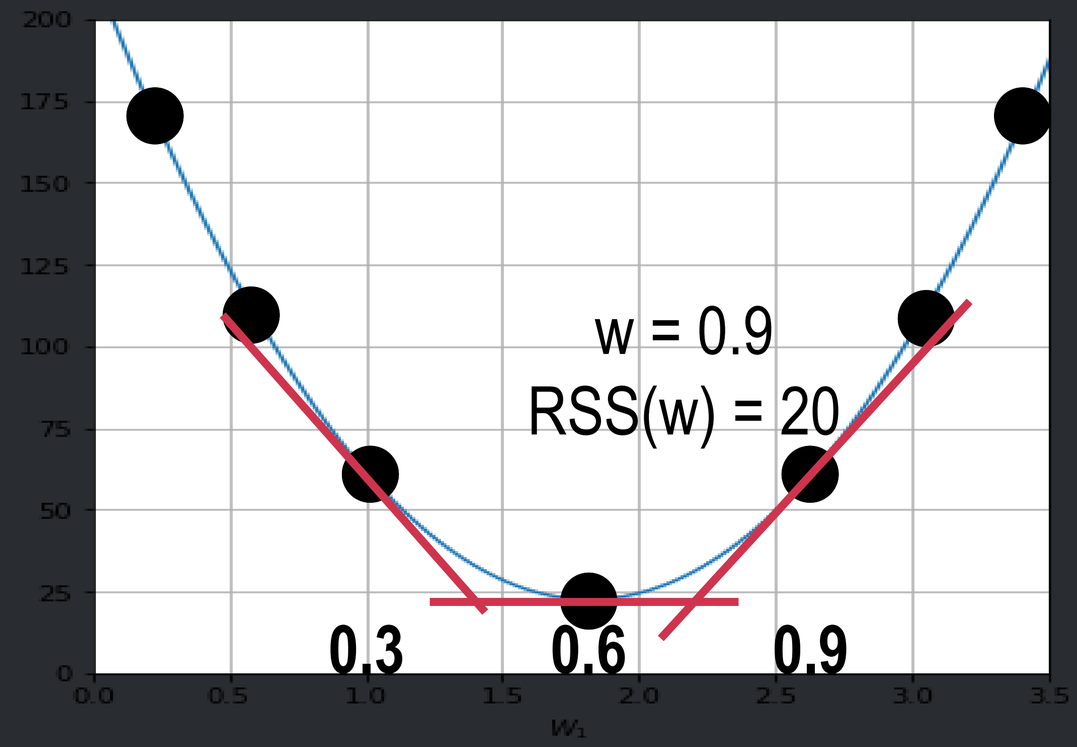
y1 – ŷ1 + y/2 – ŷ2 + y3 – ŷ3 + y4 – ŷ4 + y5 – ŷ5

\+ - - + +

→ MIN, w


delta = 3.1623
y
2
1
0
−1
−2
−3
−2
−1
0
1
2
3

# 경사 하강법




- 2

RSS(w)

+ 2

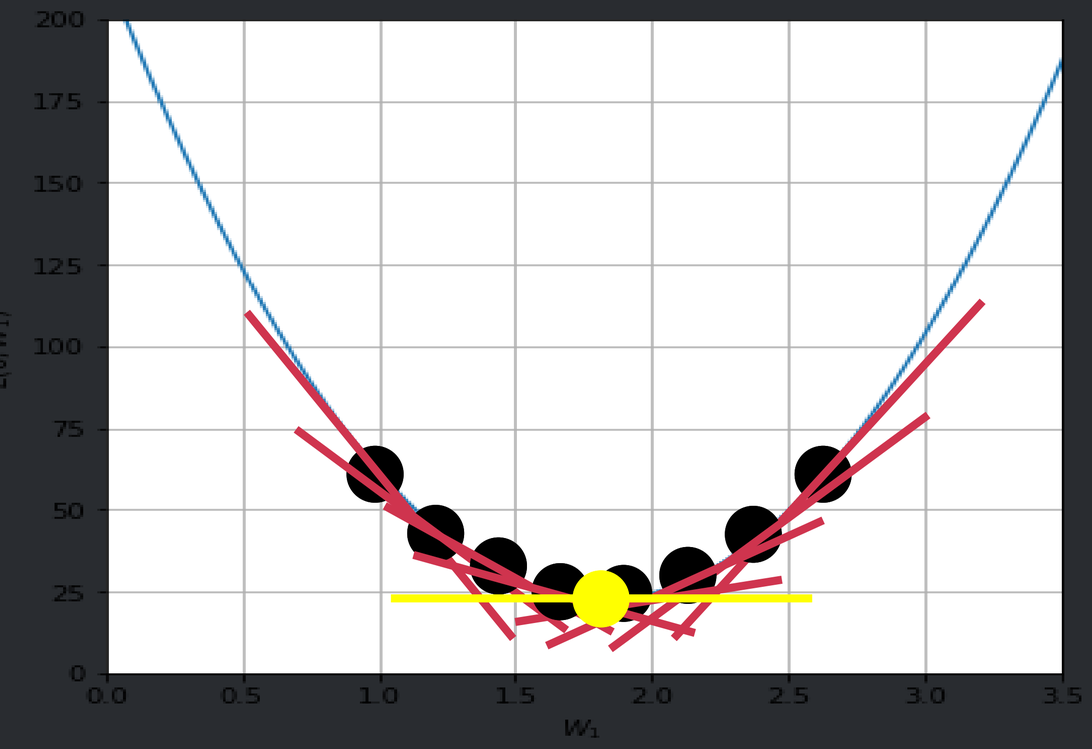
0

W

# 경사 하강법

RSS(w)

w = 0.3
RSS(w)
기울기 = -
w + ?




w

w = 0.9
RSS(w)
기울기 = +

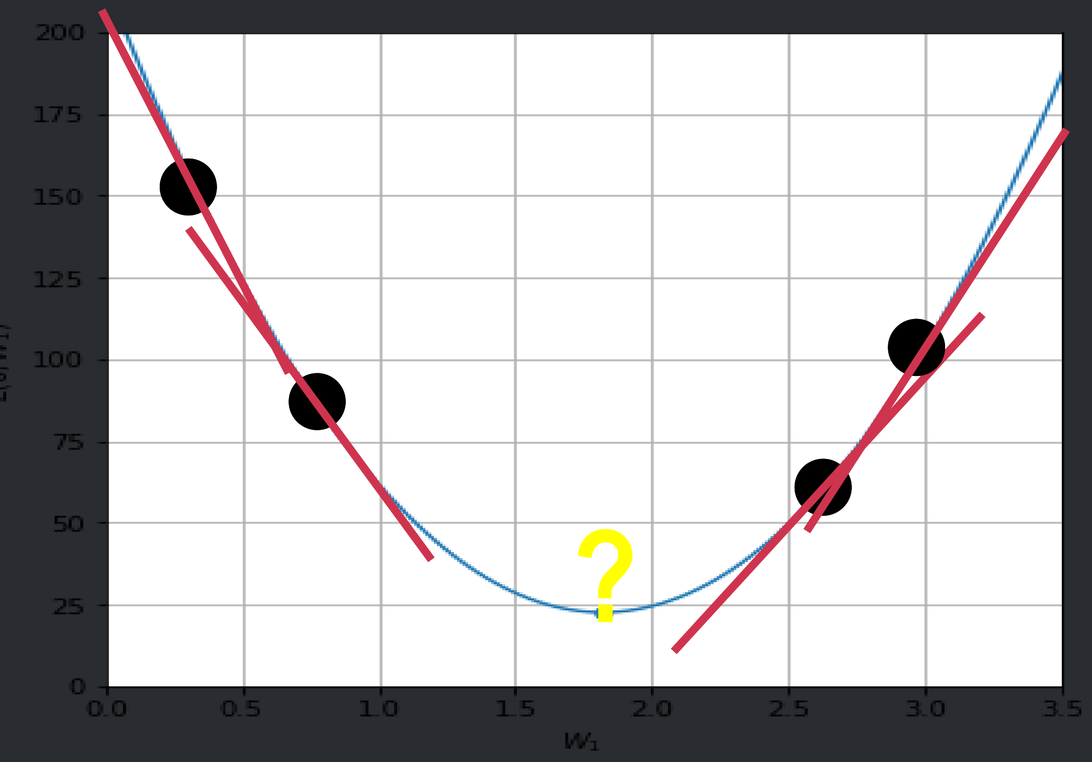
0.9 - ?

# 경사 하강법

RSS(w)

기울기 = -

0.2 + 0.9




w

w = 0.9

RSS(w)

기울기 = +

0.9 – 0.7

$$RSS(w) = \frac{1}{N}\sum_{i=1}^{n}(y_i - \grave{y}_i)^2$$

$$\grave{y}_i = w\ x_i$$

$$RSS(w) = \frac{1}{N}\sum_{i=1}^{n}(y_i - (w * x_i))^2$$

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

$(a-b)^2$

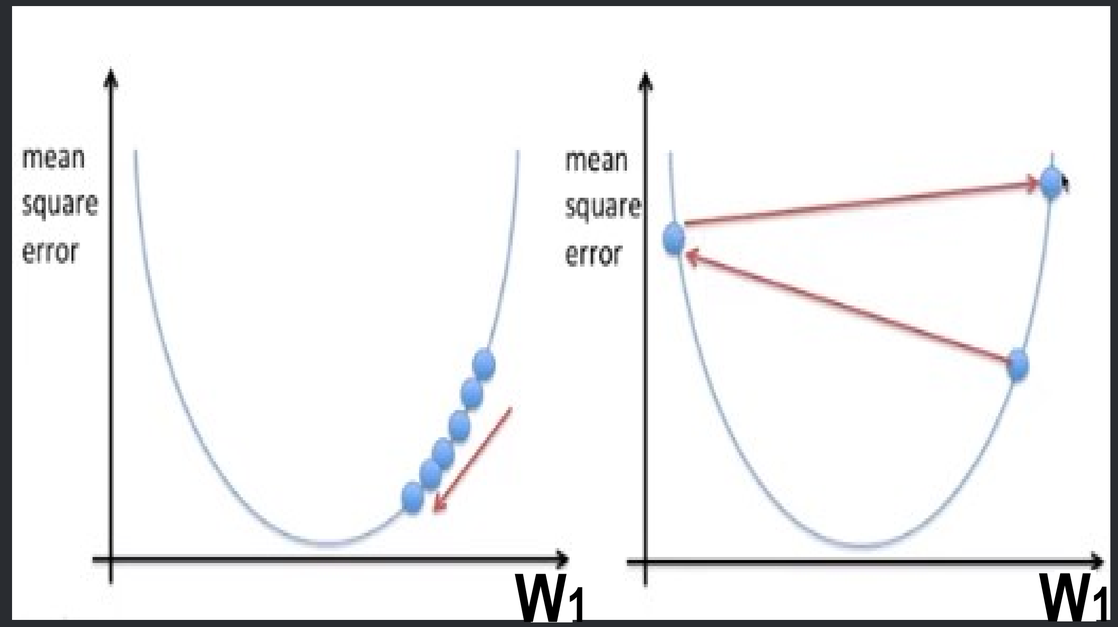
→ $a^2 - 2ab + b^2$

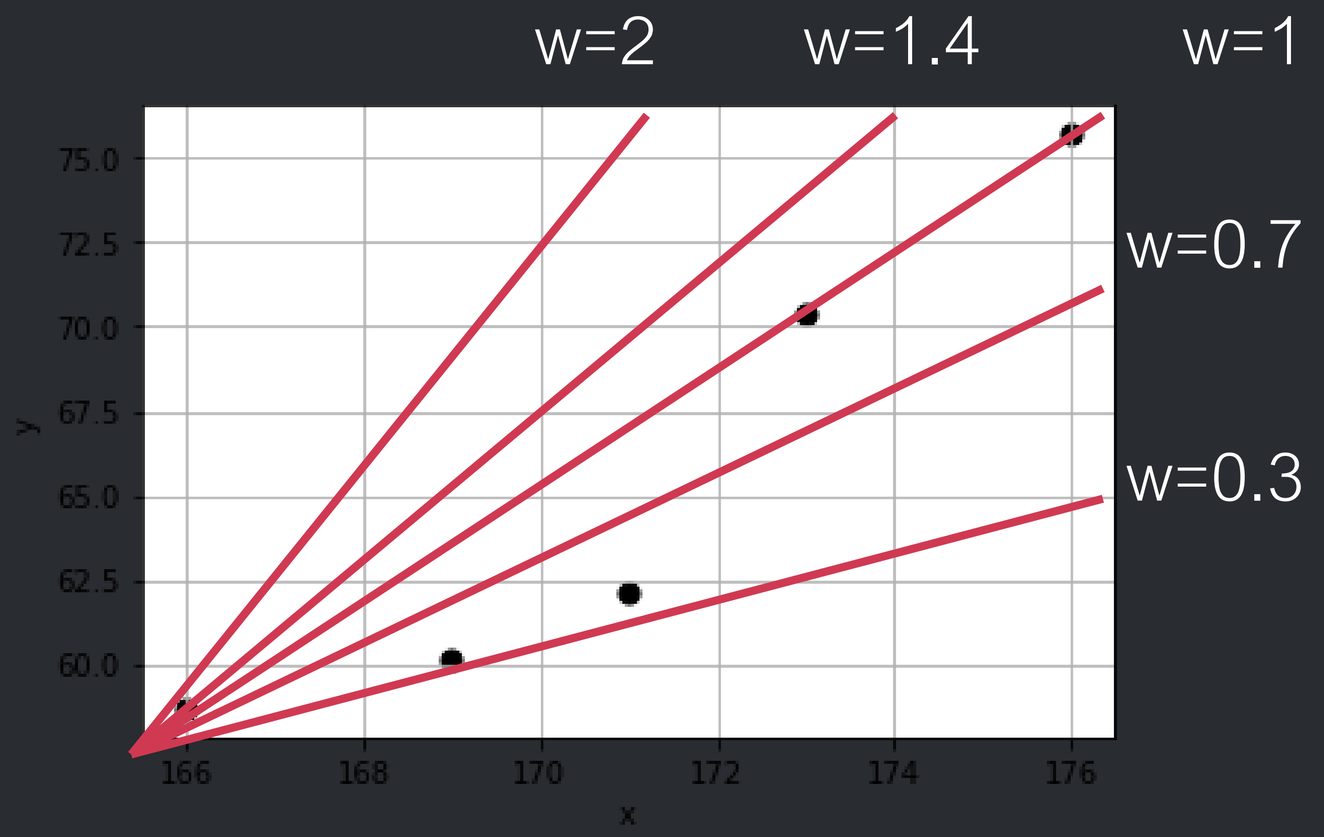
→ $y_i^2 - 2y_i(w * x_i) + (w * x_i)^2$

$$\frac{\partial R(w)}{\partial w} = \frac{2}{N}\sum_{i=1}^{n} -x_i * (y_i - (w * x_i) = -\frac{2}{N}\sum_{i=1}^{n} x_1 * (y_i - \hat{y}_i)$$

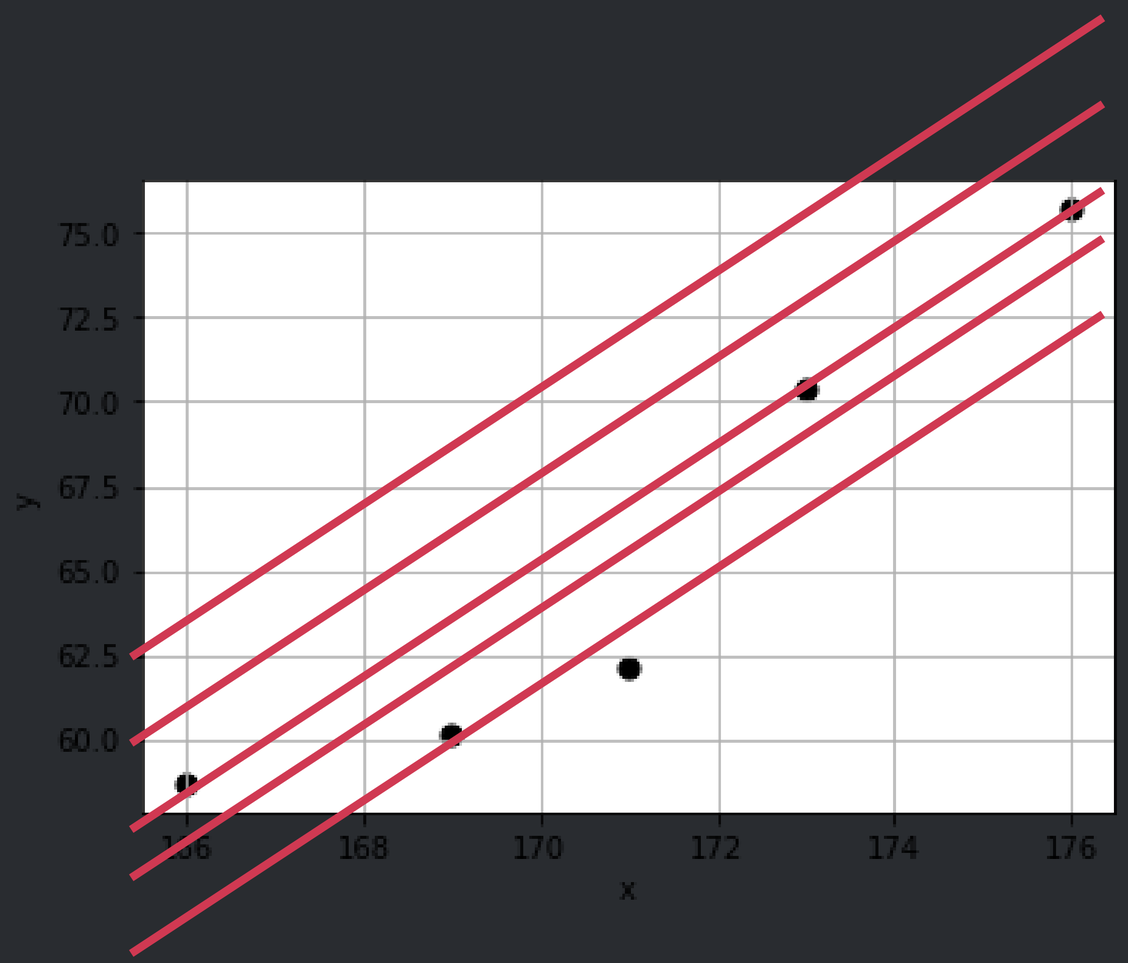
$$w = w - \alpha \frac{\partial R(w)}{\partial w}$$

α 는 learning rate

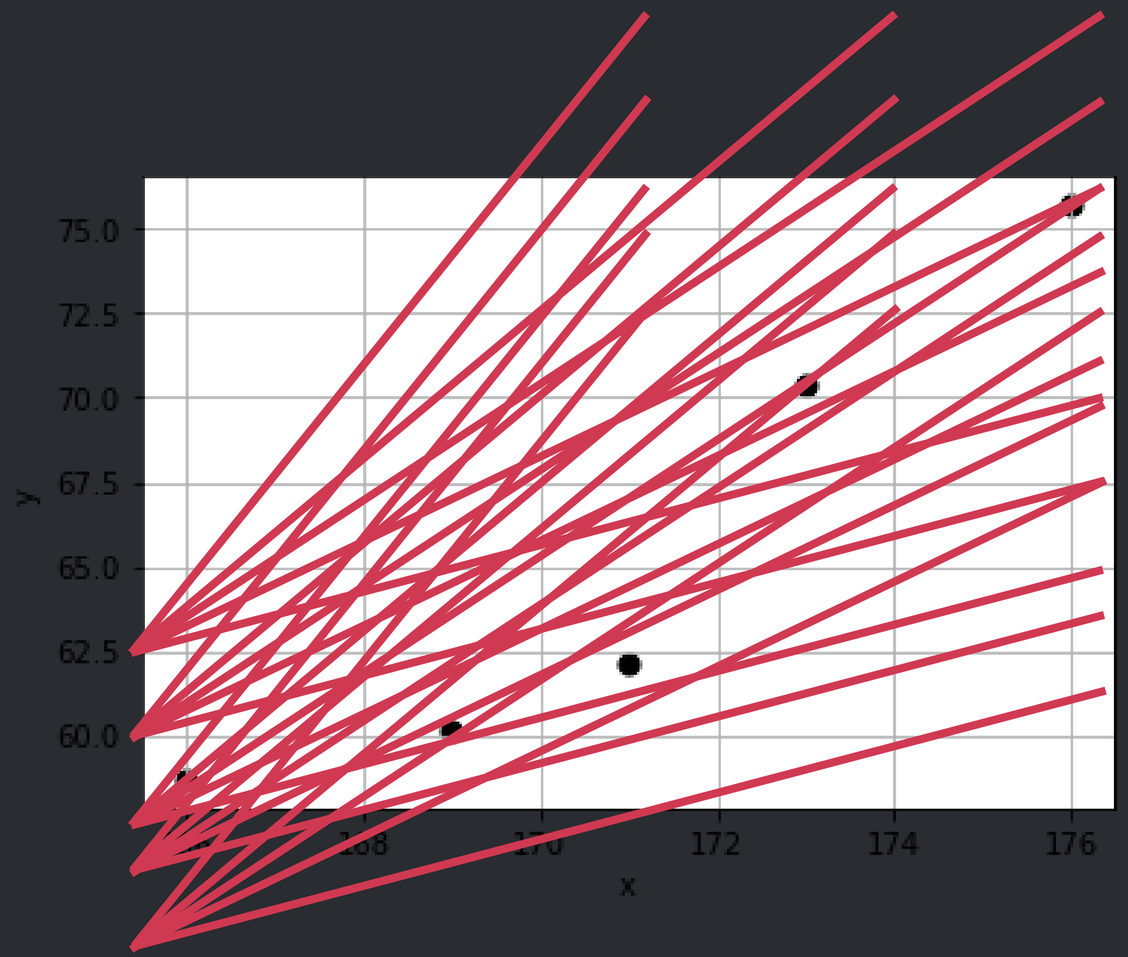



w=2
w=1.4
w=1
w=0.7
w=0.3
75.0
72.5
70.0
67.5
65.0
62.5
60.0
y
166
168
170
172
174
176
x

$$y = wx$$



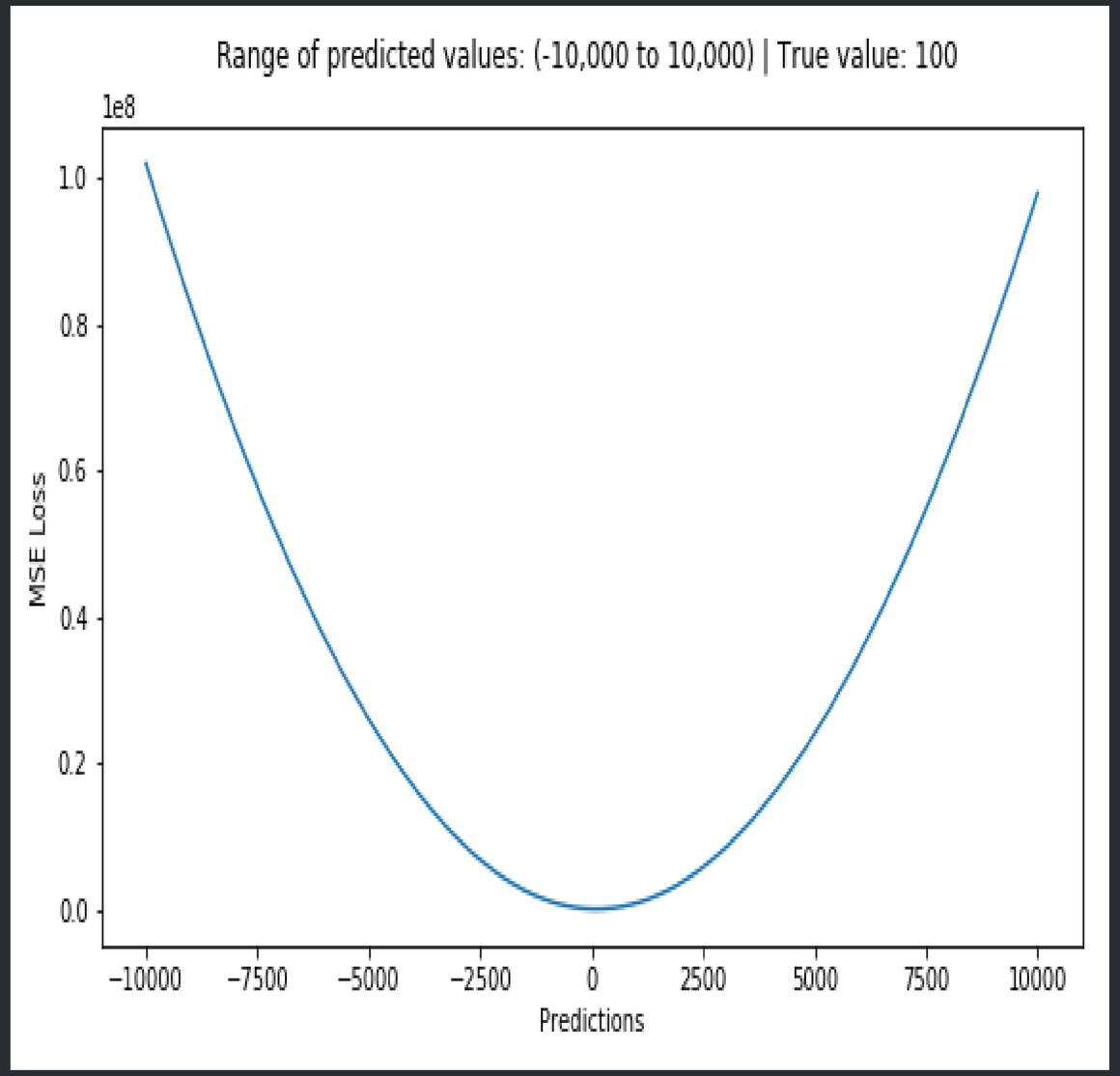

$$y = wx + w_1$$

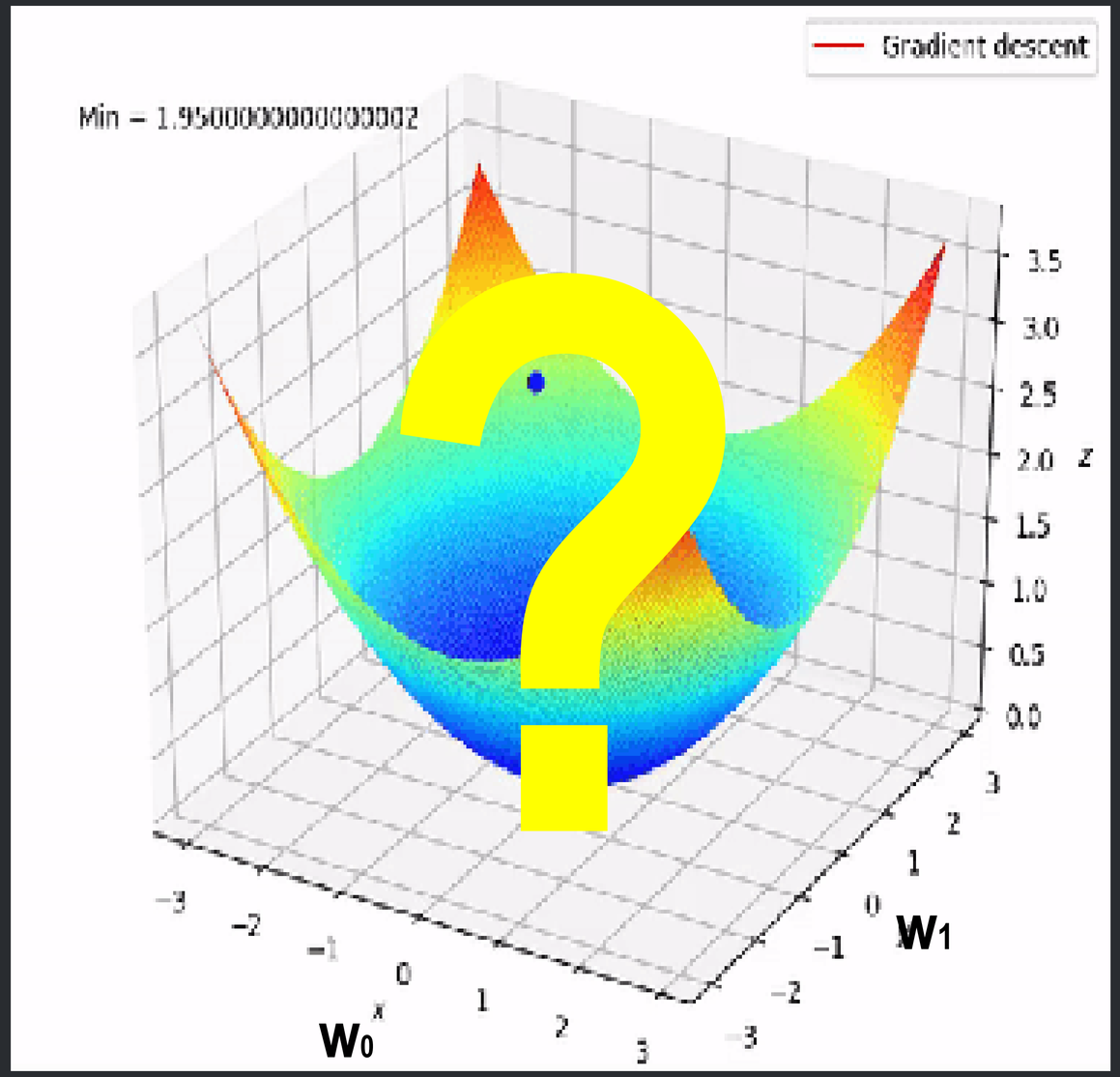



$$y = wx + w_1$$

# $Y = WX$

# $Y = W_0X + W_1$


Range of predicted values: (-10,000 to 10,000) | True value: 100
1e8
1.0
0.8
0.6
0.4
0.2
0.0
MSE Loss
−10000
−7500
−5000
−2500
0
2500
5000
7500
10000
Predictions


Gradient descent
Min = 1.9500000000000002
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
z
−3
−2
−1
0
1
2
3
x
W0
W1

RSS(W) → RSS(W0, W1)

$$RSS(w_0 + w_1) = \frac{1}{N}\sum_{i=1}^{n}(y_i - \grave{y}_i)^2$$

$$\grave{y}_i = w_0 + w_1 x_i$$

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

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

$$(a - b)^2$$

$$a^2 - 2ab + b^2$$

$$y_i^2 - 2y_i(w_0 + w_1 * x_i) + (w_0 + w_1 * x_i)^2$$

변수가 두개 이상일 때의 미분?

# 편미분

$$\frac{\partial R(w)}{\partial w_1} = \frac{2}{N}\sum_{i=1}^{n} -x_1 * (y_i - (w_0 + w_1 * x_i) = -\frac{2}{N}\sum_{i=1}^{n} -x_1 * (\text{실제값}_i - \text{예측값}_i)$$

$$w_1 = w_1 - \alpha * \frac{\partial R(w)}{\partial w_1}$$

$$\frac{\partial R(w)}{\partial w_0} = \frac{2}{N}\sum_{i=1}^{n} -(y_i - (w_0 + w_1 * x_i) = -\frac{2}{N}\sum_{i=1}^{n} (\text{실제값}_i - \text{예측값}_i)$$

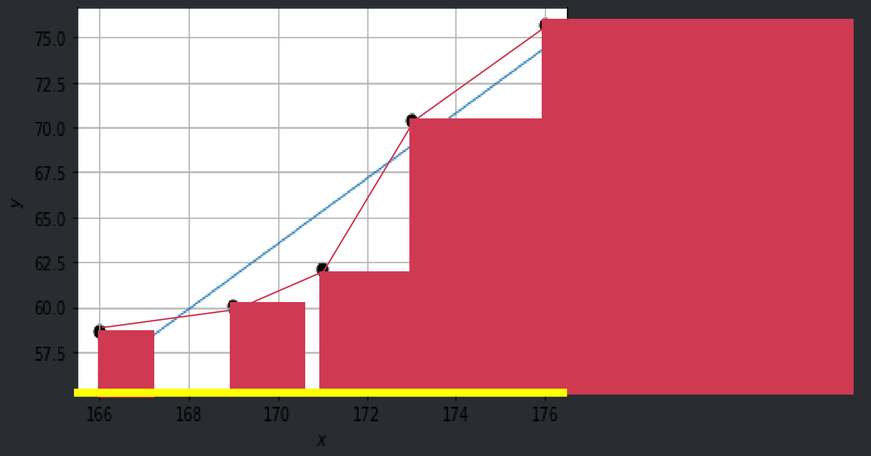
$$w_0 = w_0 - \alpha * \frac{\partial R(w)}{\partial w_0}$$

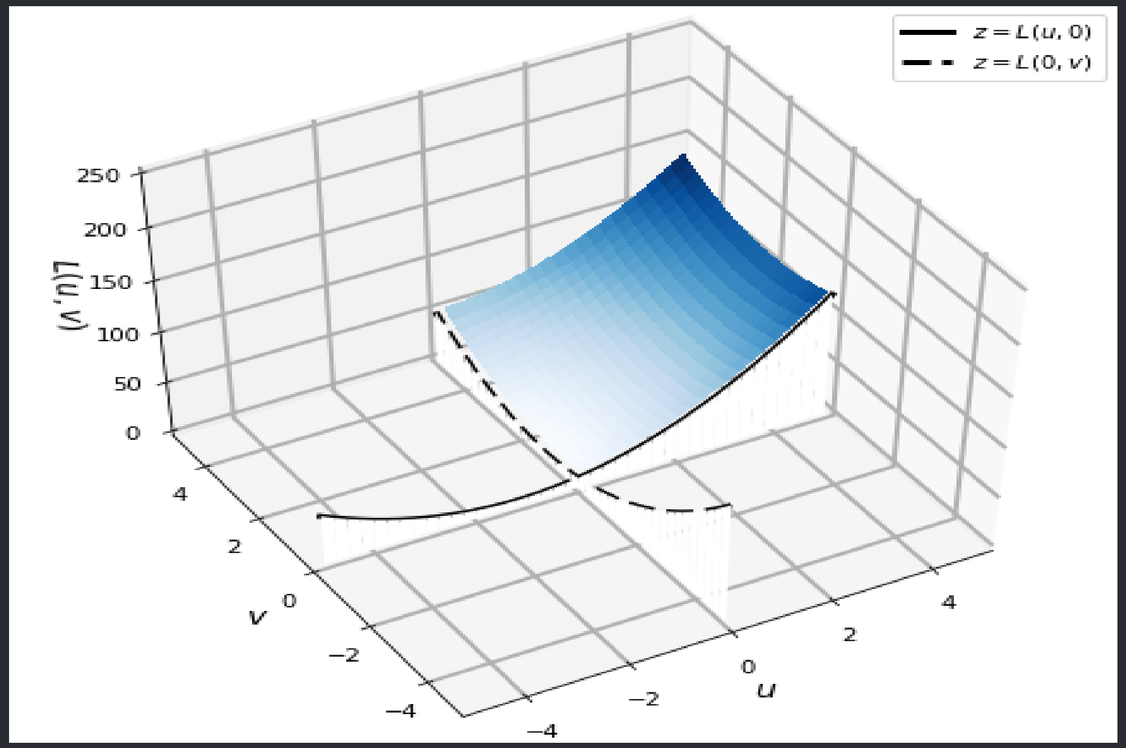
$$y = w_0 + w_1 x$$

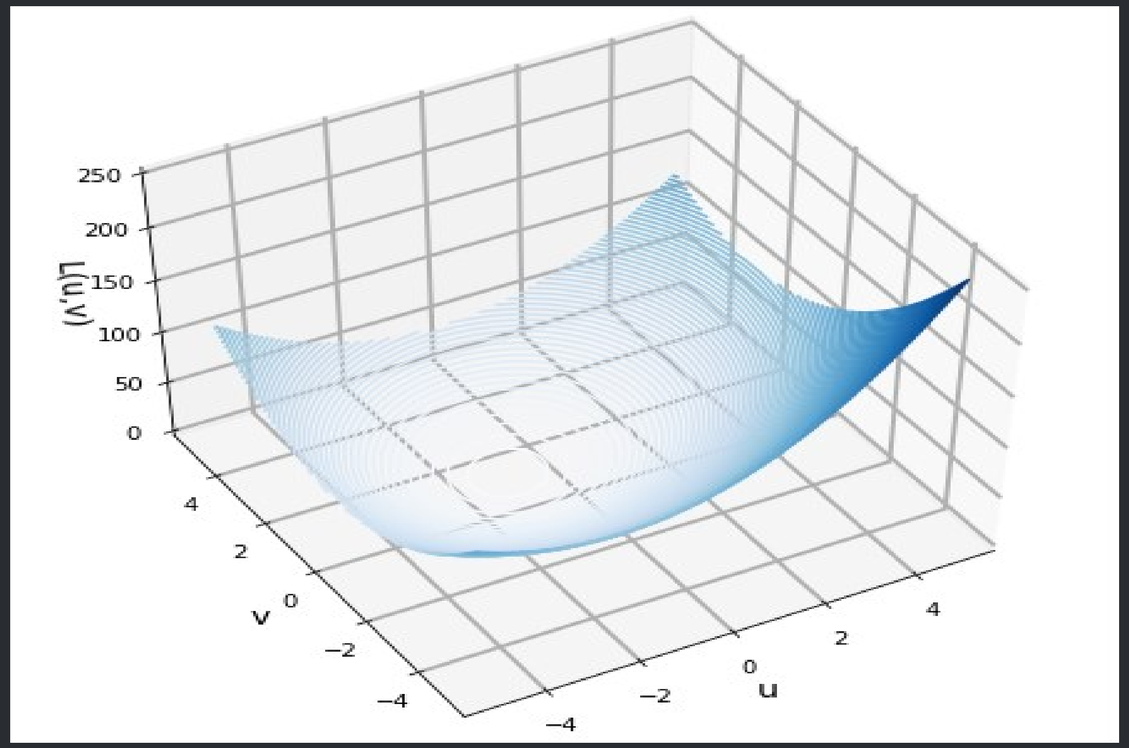
| x | y | ŷ | |
|---|---|---|---|
| 1 | 4 | 0 | w0 = 0, w1 = 0 |
| 2 | 8 | 0 | |
| 3 | 7 | 0 | |
| 4 | 10 | 0 | |
| 5 | 11 | 0 | |

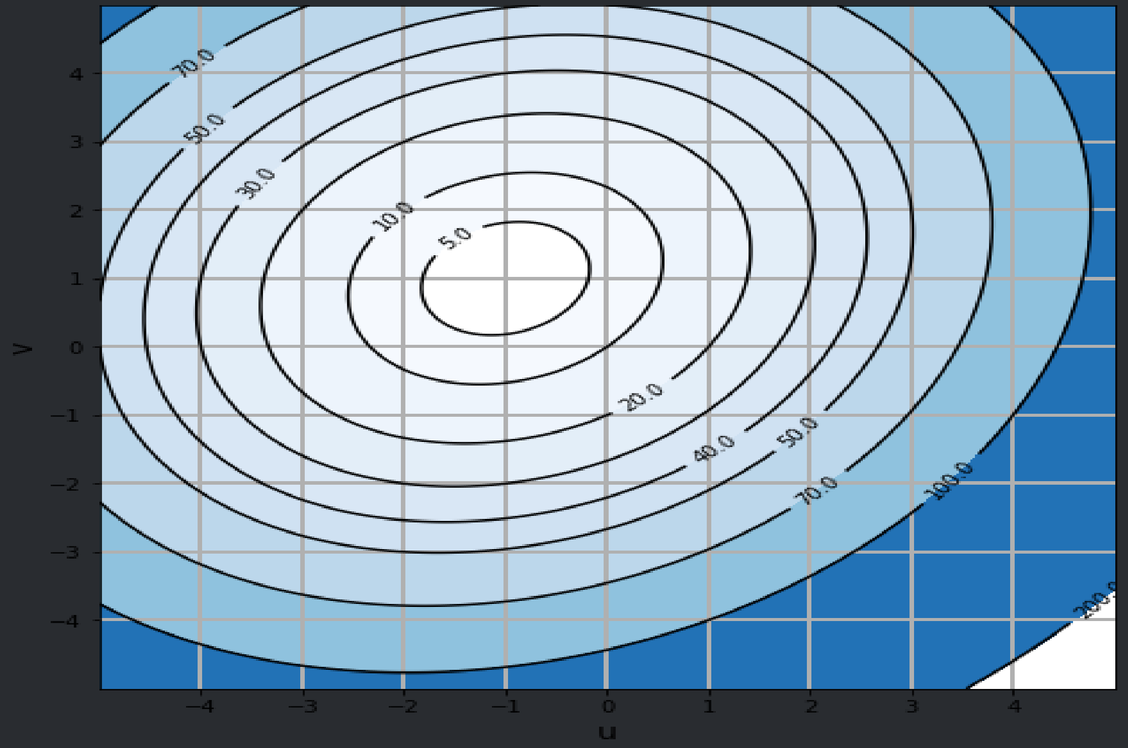
$$w_1 = w_1 - \alpha * \frac{\partial R(w)}{\partial w_1}$$

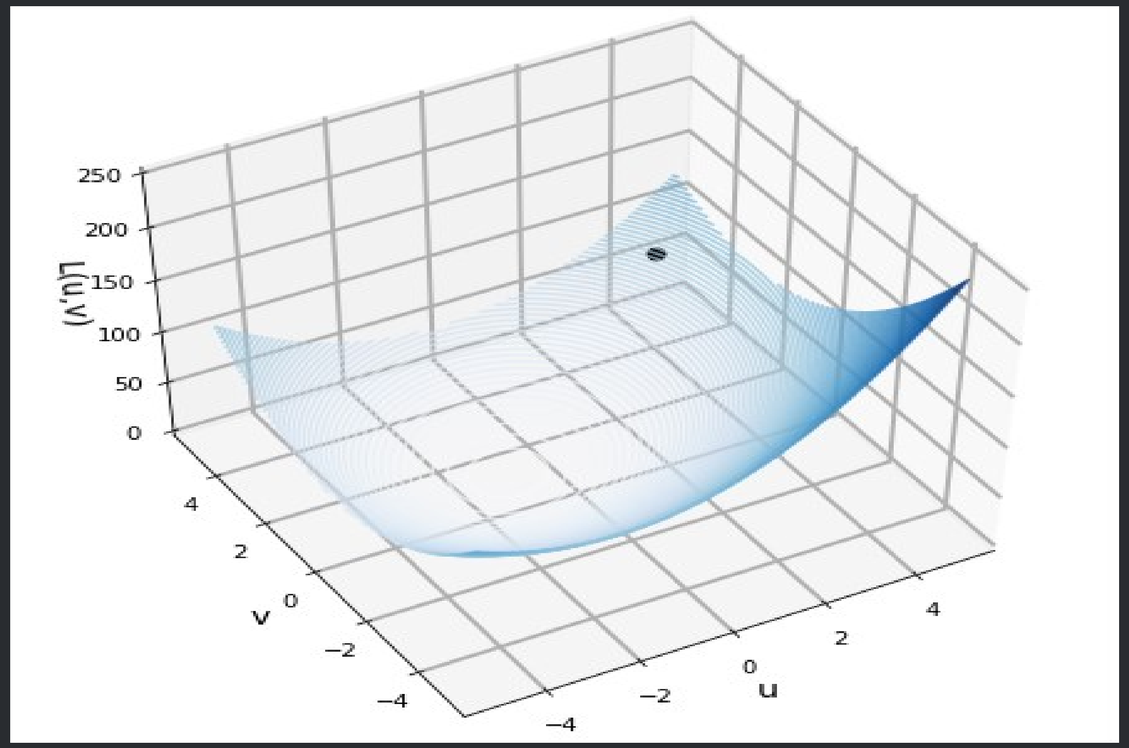
$$w_0 = w_0 - \alpha * \frac{\partial R(w)}{\partial w_0}$$

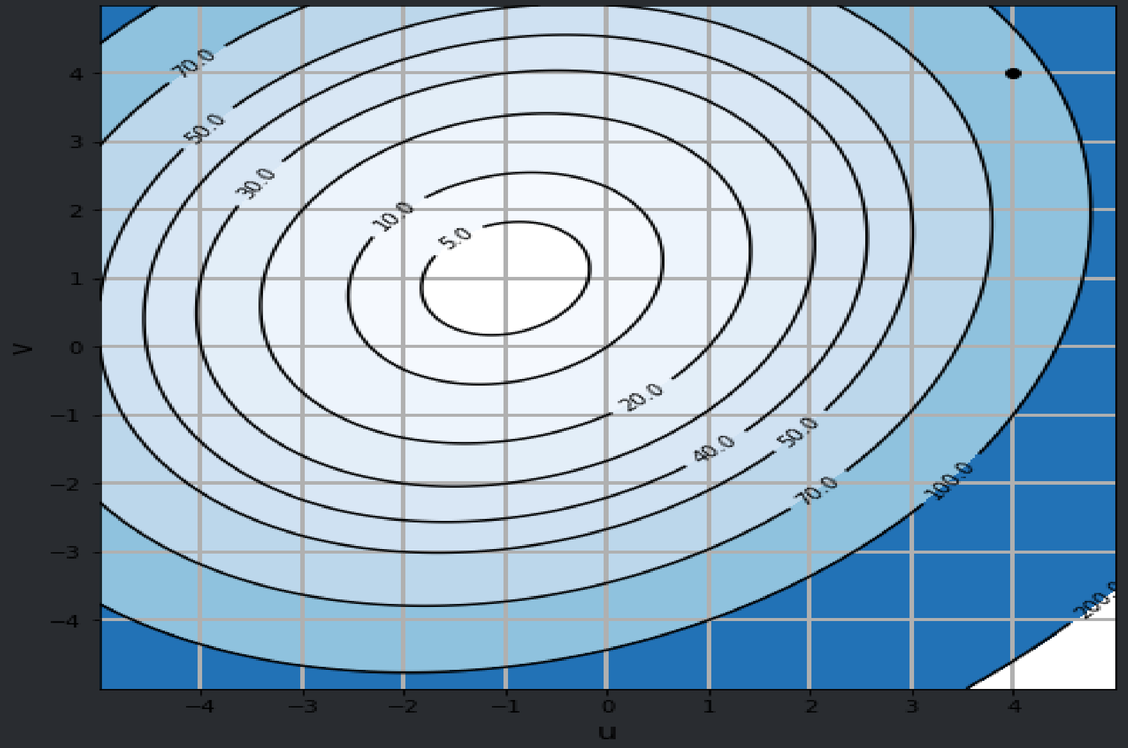


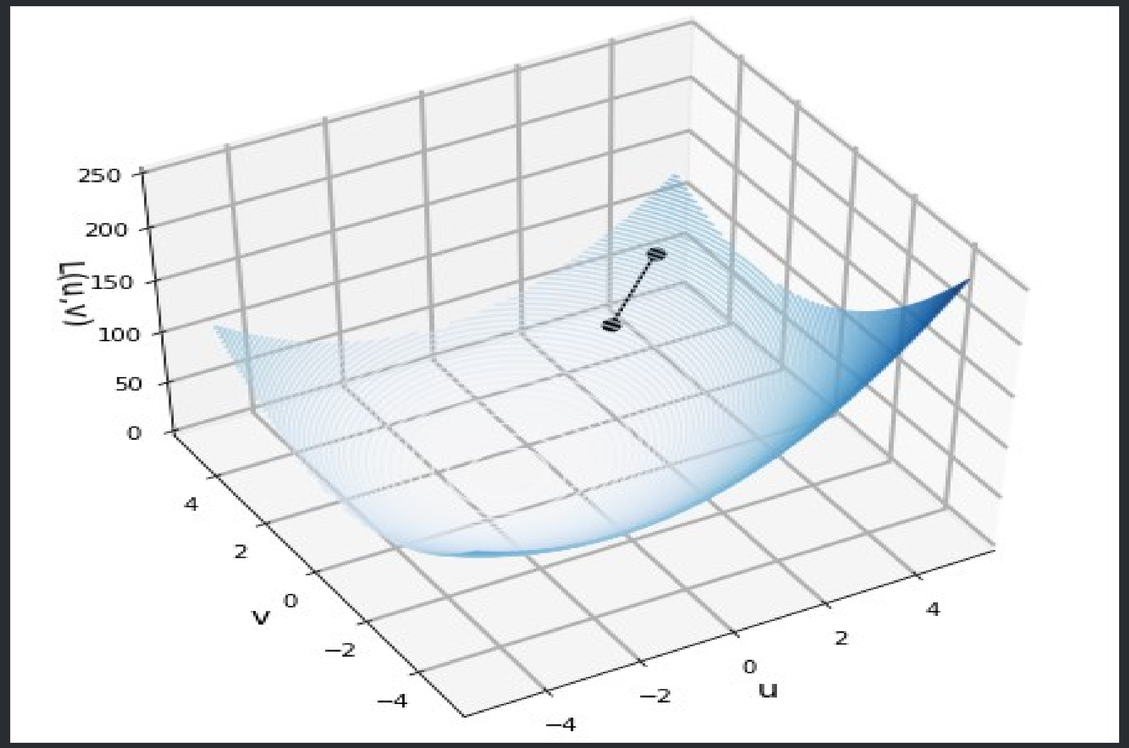

$z = L(u, 0)$
$z = L(0, v)$
250
200
150
100
50
0
$L(u, v)$
4
2
0
−2
−4
$v$
−4
−2
0
2
4
$u$

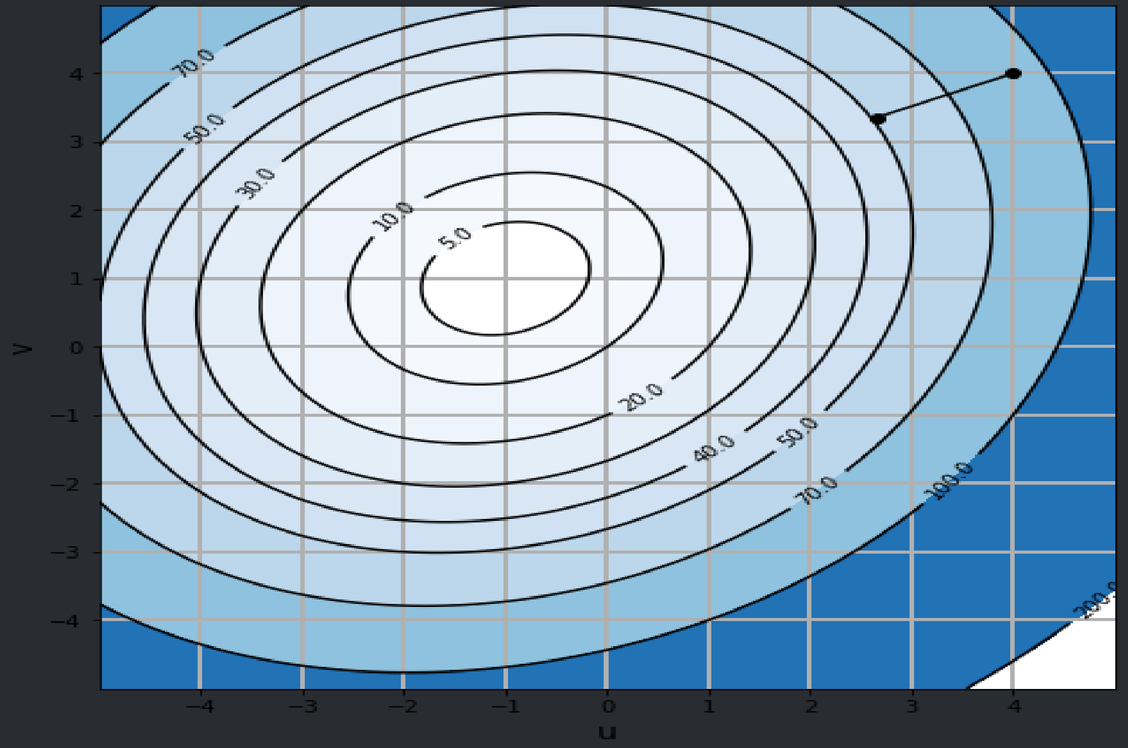

250
200
150
100
50
0
L(u,v)
4
2
0
−2
−4
v
−4
−2
0
2
4
u

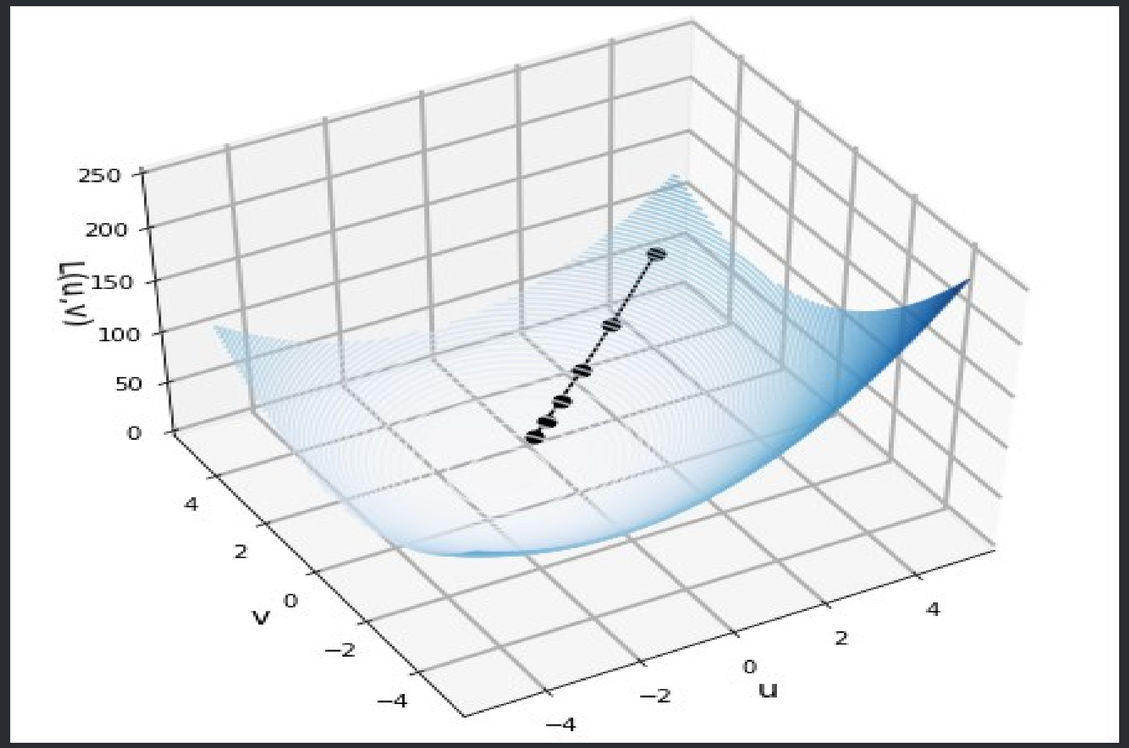

70.0
50.0
30.0
10.0
5.0
20.0
40.0
50.0
70.0
100.0
4
3
2
1
0
−1
−2
−3
−4
v
−4
−3
−2
−1
0
1
2
3
4
u

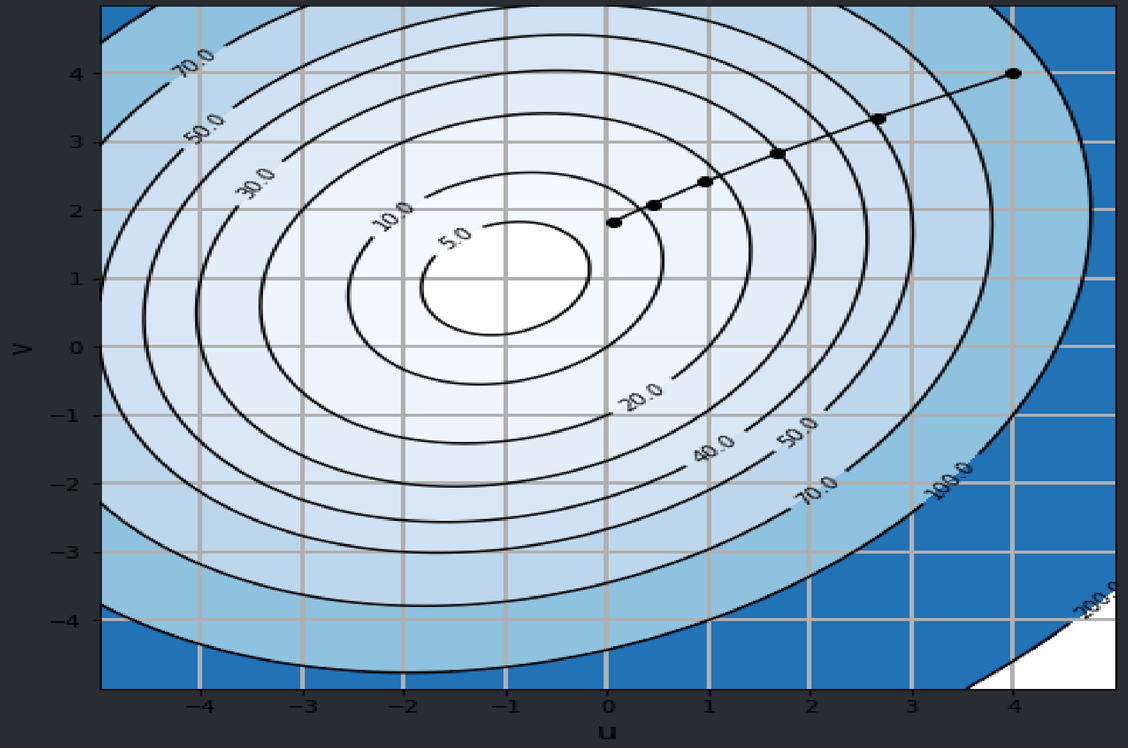

250
200
150
100
50
0
L(u,v)
4
2
0
−2
−4
v
−4
−2
0
2
4
u

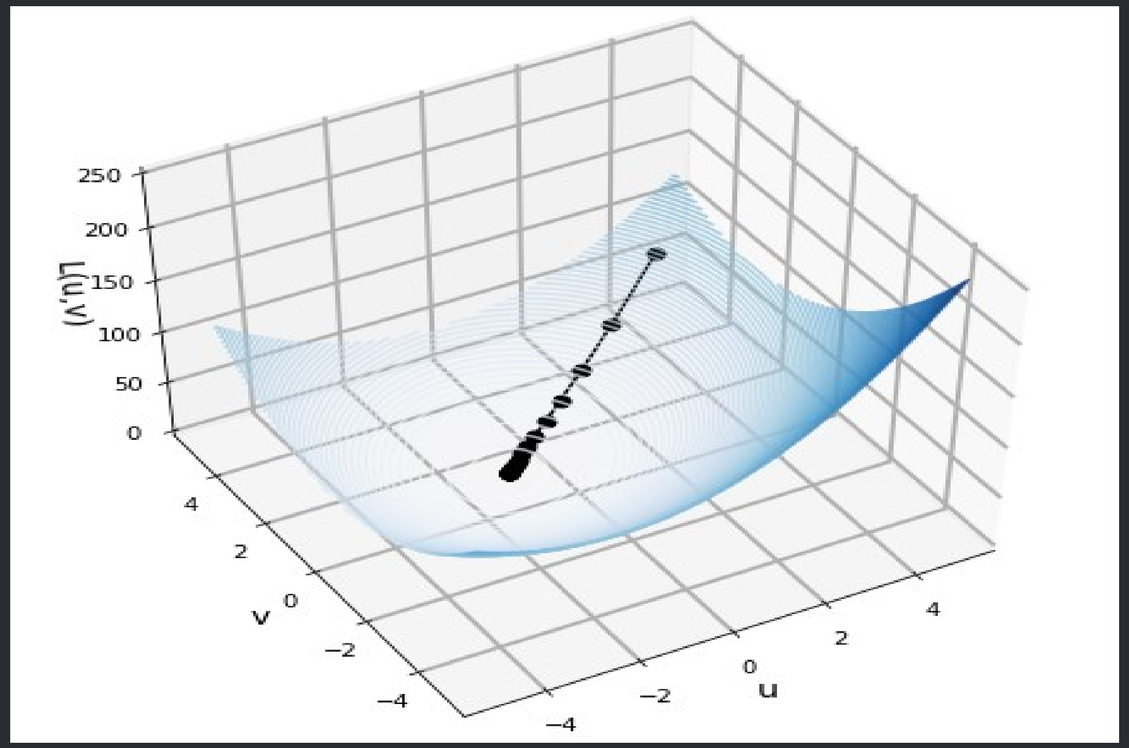

70.0
50.0
30.0
10.0
5.0
20.0
40.0
50.0
70.0
100.0
4
3
2
1
0
−1
−2
−3
−4
v
−4
−3
−2
−1
0
1
2
3
4
u

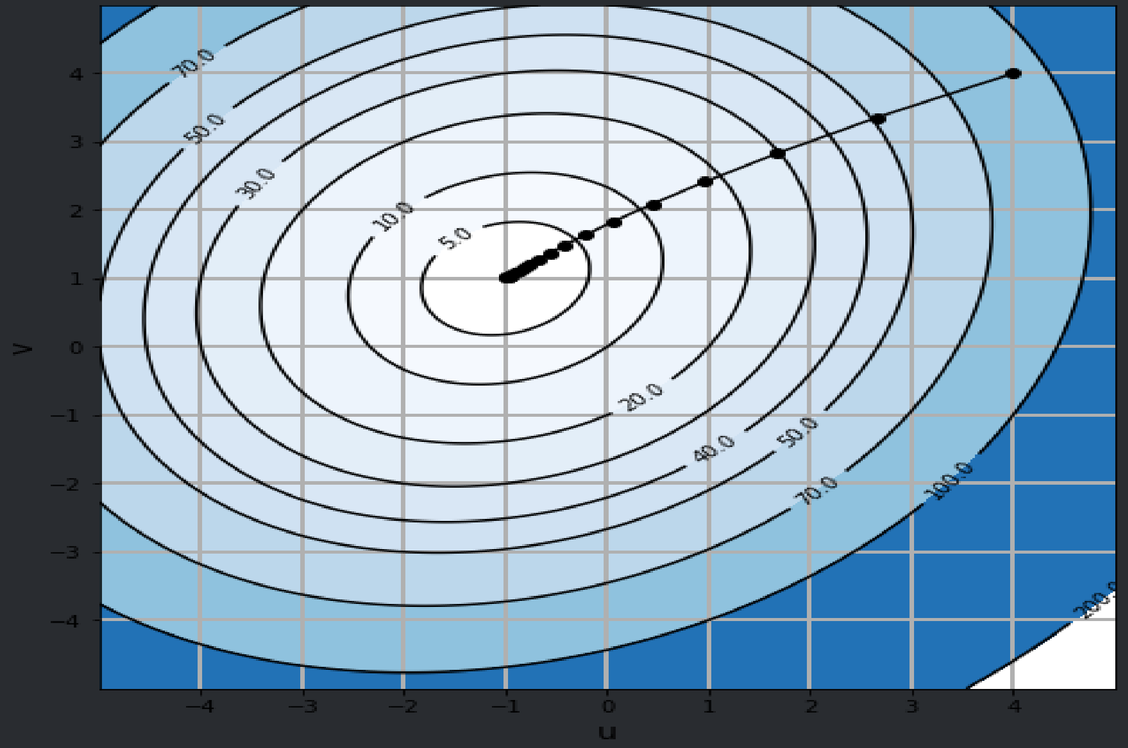

250
200
150
100
50
0
L(u,v)
4
2
0
−2
−4
v
−4
−2
0
2
4
u

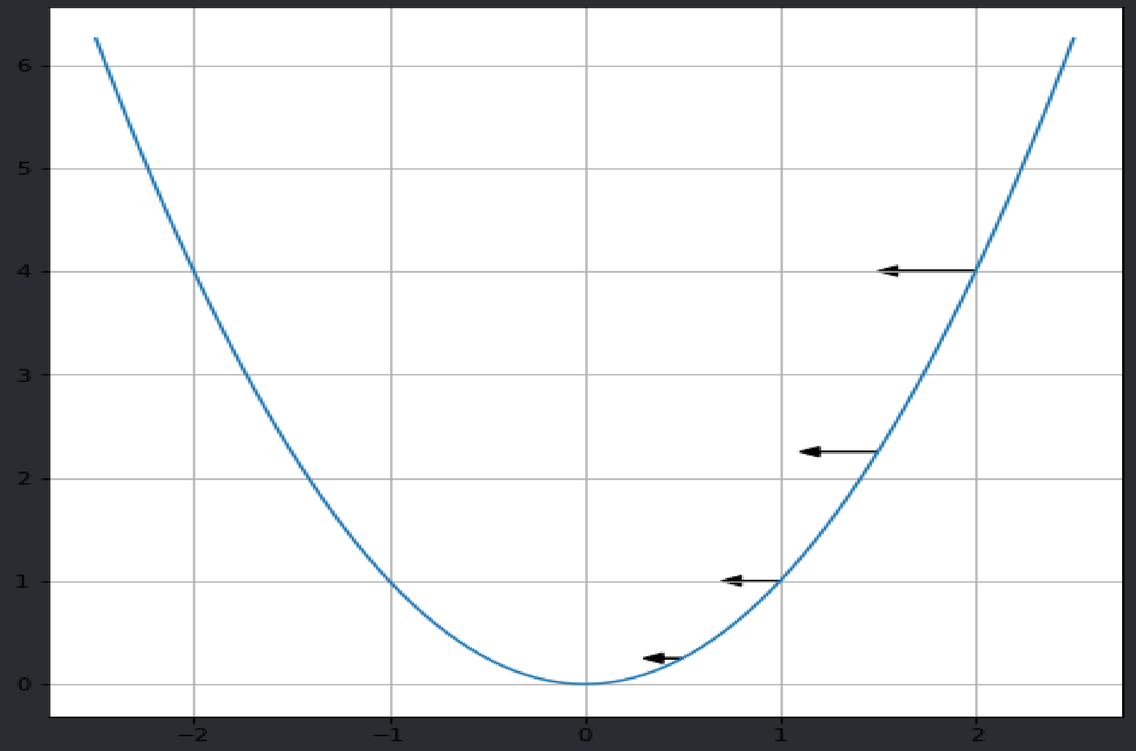

70.0
50.0
30.0
10.0
5.0
20.0
40.0
50.0
70.0
100.0
4
3
2
1
0
−1
−2
−3
−4
v
−4
−3
−2
−1
0
1
2
3
4
u

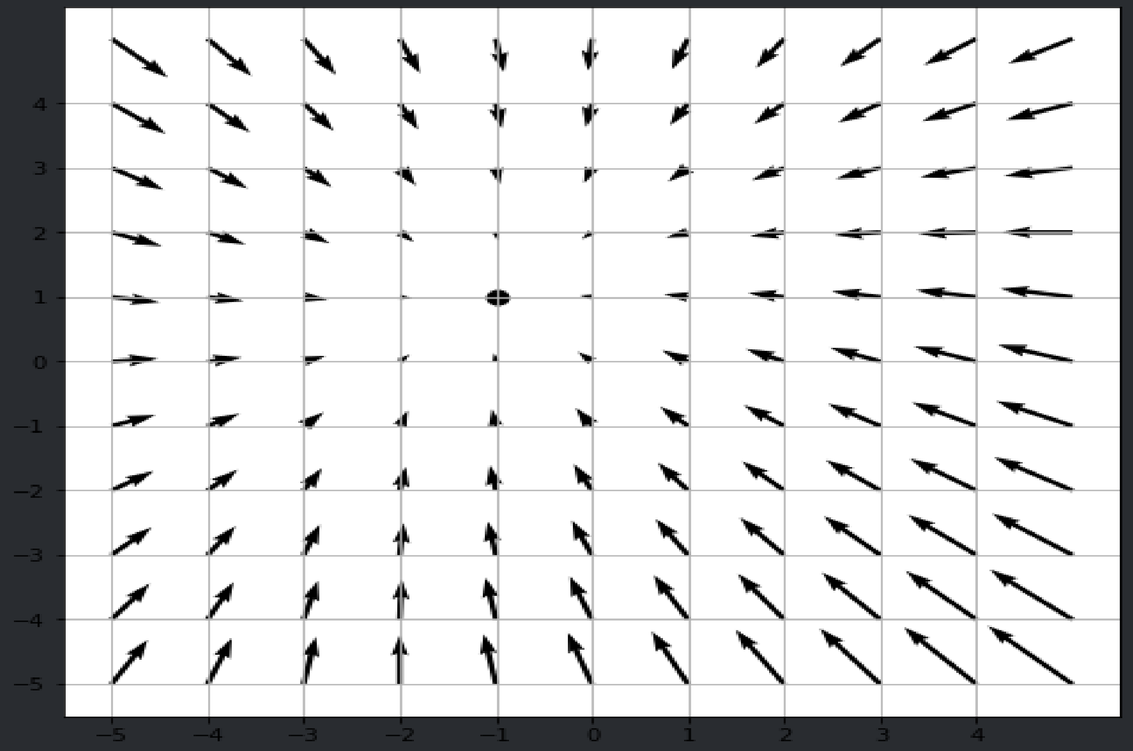

250
200
150
100
50
0
L(u,v)
4
2
0
−2
−4
v
−4
−2
0
2
4
u


70.0
50.0
30.0
10.0
5.0
20.0
40.0
50.0
70.0
100.0
4
3
2
1
0
−1
−2
−3
−4
v
−4
−3
−2
−1
0
1
2
3
4
u


250
200
150
100
50
0
L(u,v)
4
2
0
−2
−4
v
−4
−2
0
2
4
u


70.0
50.0
30.0
10.0
5.0
20.0
40.0
50.0
70.0
100.0
4
3
2
1
0
−1
−2
−3
−4
v
−4
−3
−2
−1
0
1
2
3
4
u


6
5
4
3
2
1
0
−2
−1
0
1
2


4
3
2
1
0
−1
−2
−3
−4
−5
−5
−4
−3
−2
−1
0
1
2
3
4

RSS(w0, w1) → MIN

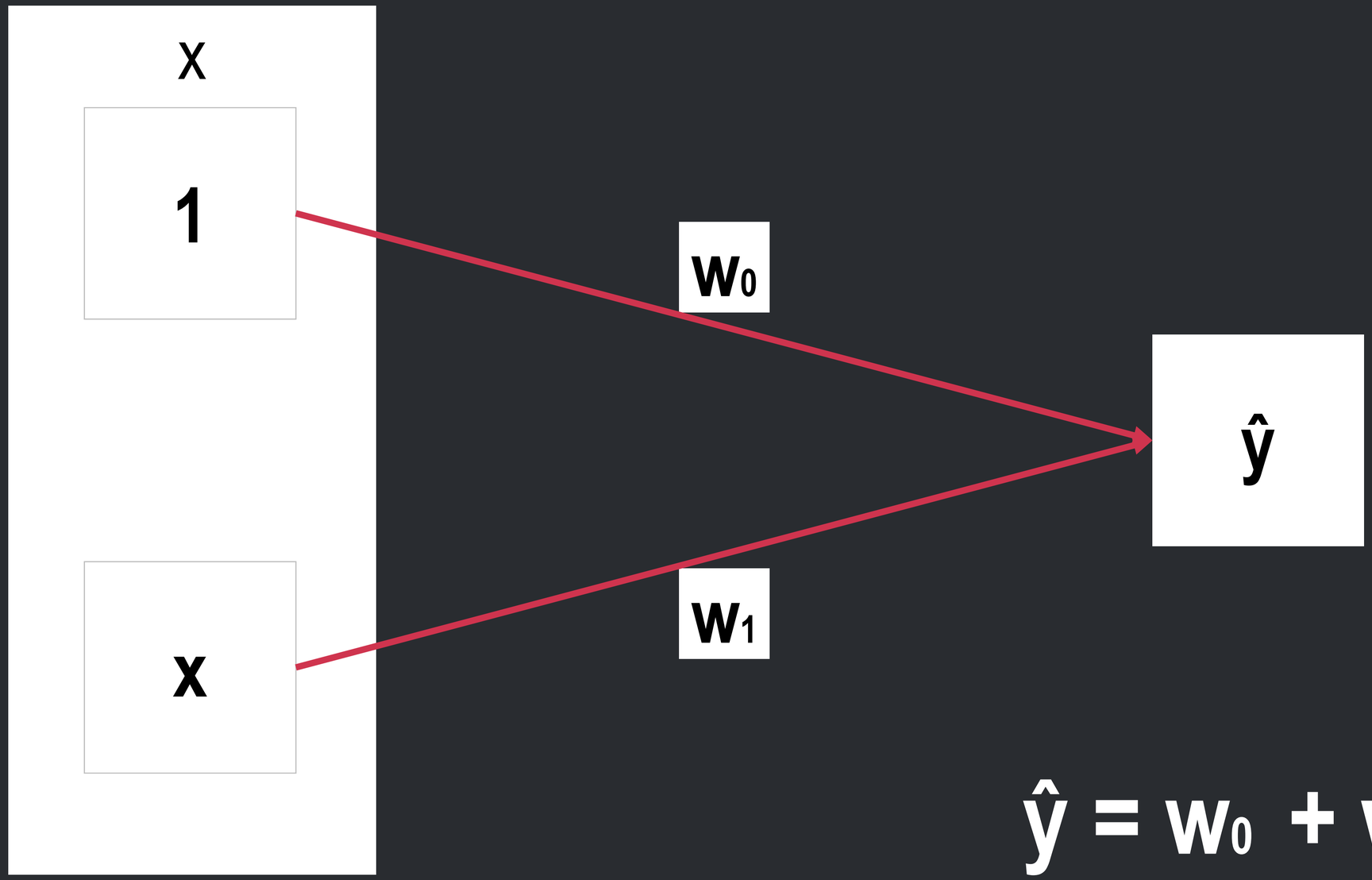
w1 = 3.4

w0 = 7.8

$\hat{y} = 3.4x + 7.8$

x = 5

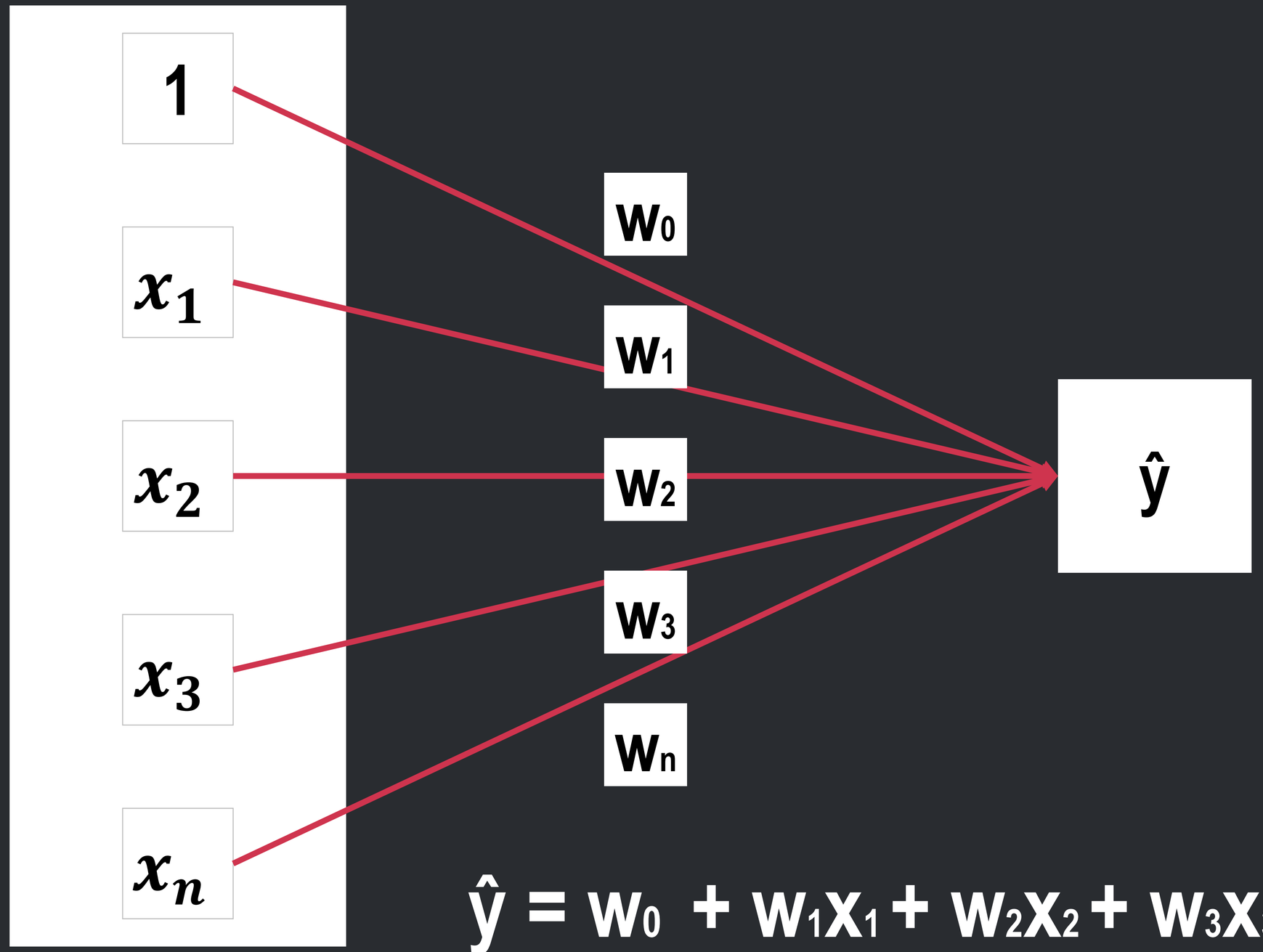
$\hat{y} = 24.8$




$$\hat{y} = w_0 + w_1 x$$

$$Y = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 + \ldots + w_n x_n$$

$$\frac{\partial R(w)}{\partial w_0}, \frac{\partial R(w)}{\partial w_1}, \frac{\partial R(w)}{\partial w_2}, \frac{\partial R(w)}{\partial w_3}, \cdots, \frac{\partial R(w)}{\partial w_n}$$




$$\hat{y} = w_0 + w_1x_1 + w_2x_2 + w_3x_3 + \ldots + w_nx_n$$

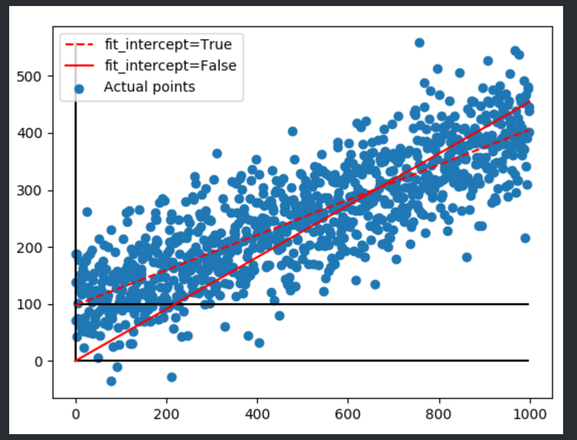
# 사이킷런

sklearn.linear_model.LinearRegression(Parameter)

# PARAMETER

- fit_intercept: True/False
  - 디폴트는 **True**
  - 절편 값을 계산할 것인지 여부
  - **False**는 절편이 **0**으로 지정

- normalize: True/False
  - 디폴트는 False
  - fit_intercept가 False인 경우 이 파라미터 무시
  - True이면 회귀를 수행하기 전 입력 데이터 세트를 정규화



# 사이킷런 LINEAR REGRESSION을 이용한 보스턴 주택 가격 예측

DATA KUBWA

- 사이킷런, linear_models 모듈
- http://scikit-learn.org/stable/modules/classes.html#module-sklearn.linear_model

- CRIM: 지역별 범죄발생율
- ZN: 25,000 평방 피트를 초과하는 거주지역 비율
- INDUS: 비 상업 지역 넓이 비율. 에이커(4050 평방미터) 기준
- CHAS: 찰스 강에 대한 더미 변수 (강의 경계에 위치한 경우 1, 그렇지 않으면 0).
- nox: 질소 산화물 농도.
- rm: 주택 당 평균 방 개수
- Age: 1940 년 이전에 지어진 집주인 주거 주택의 비율
- dis: 5 개의 보스턴 고용 센터까지의 가중 거리.
- rad: 고속도로 접근성 지수.
- tax: $ 10,000 당 제산세율.
- ptratio: 지역별 학생-교사 비율.
- black: 1000 (Bk-0.63) ^ 2. 지역 별 흑인 비율 (인종차별?)
- lstat: 하위 계층 비율 (기준 ?)
- medv: 집주인 주거 주택 가격 (중앙값). $ 1000 단위

$$Y = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 + \ldots + w_{13} x_{13}$$

$$\frac{\partial R(w)}{\partial w_0}, \frac{\partial R(w)}{\partial w_1}, \frac{\partial R(w)}{\partial w_2}, \frac{\partial R(w)}{\partial w_3}, \cdots, \frac{\partial R(w)}{\partial w_{13}}$$

# 회귀평가지표

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}|$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2$$

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2}$$

$$R^2 = 1 - \frac{\sum(y_i - \hat{y})^2}{\sum(y_i - \bar{y})^2}$$

Where,

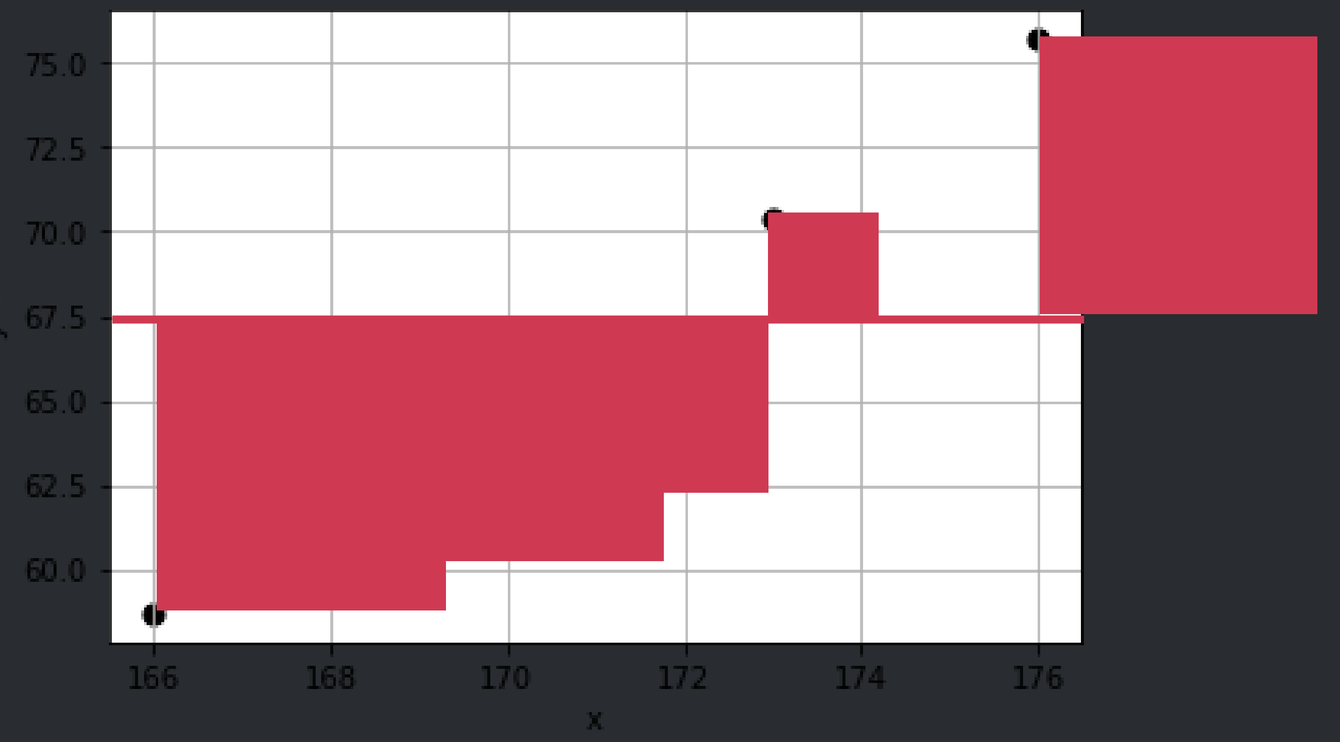
$\hat{y}$ – *predicted value of* $y$

$\bar{y}$ – *mean value of* $y$

- MAE (Mean Absolute Error)
- MSE (Mean Squared Error)
- RMSE (Root Mean Squared Error)
- $R^2$
- $R^2$이 0이 되는 경우?
- 음수가 나오는 경우?

# $R^2$이 0이 되는 경우

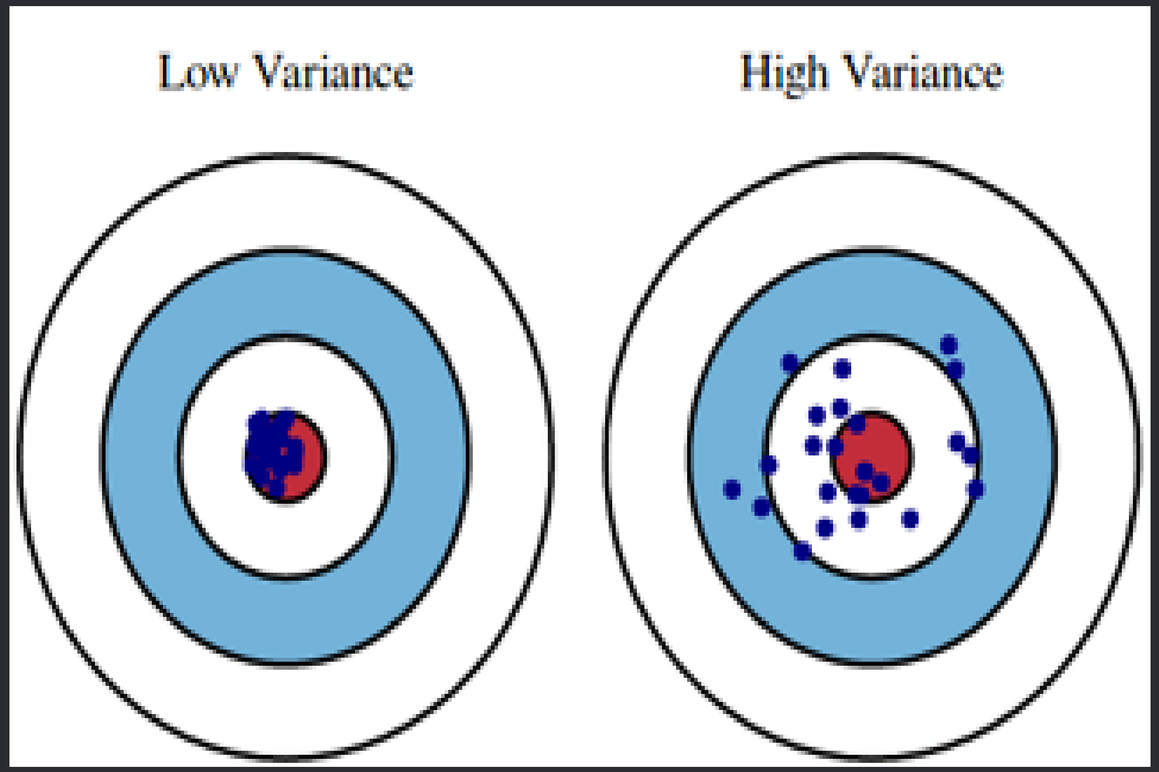
μ = 67




y = 67

$R^2$이 음수가 되는 경우?

$$S^2 = \frac{1}{n-1} \sum_{i=1}^{N} (x_i - \bar{x})^2$$




$$S^2 = \frac{1}{N} \sum_{i=1}^{N} (x_i - \bar{x})^2$$

| | 수현 | 수지 | 효주 | 세영 | 진구 | 호준 | 서진 | 지우 | 재석 | 준하 |
|---|---|---|---|---|---|---|---|---|---|---|
| 키(cm) | 177 | 167 | 160 | 162 | 174 | 180 | 176 | 158 | 172 | 184 |

$$\mu = \frac{1}{10}(177 + 167 + 160 + 162 + 174 + 180 + 176 + 158 + 172 + 184) = 171$$

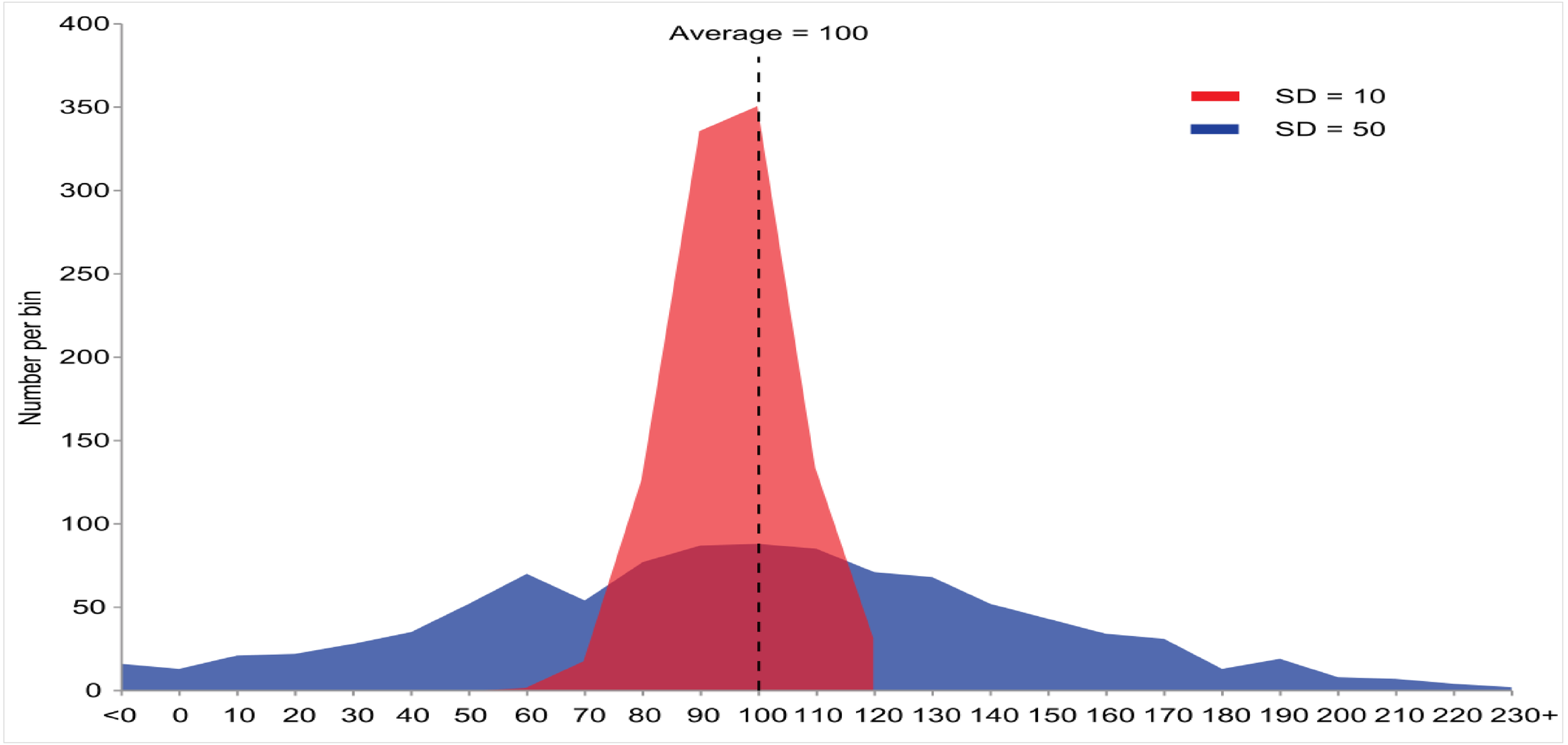
| | 수현 | 수지 | 효주 | 세영 | 진구 | 호준 | 서진 | 지우 | 재석 | 준하 | 합계 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $\lvert X - \mu \rvert$ | 6 | 4 | 11 | 9 | 3 | 9 | 5 | 13 | 1 | 13 | 74 |
| $(X - \mu)^2$ | 36 | 16 | 121 | 81 | 9 | 81 | 25 | 169 | 1 | 169 | 708 |

$$\mathrm{V}(X) = \mathrm{E}((X - \mu)^2) = \frac{708}{10} = 70.8$$

$$\sigma_X = \sqrt{\mathrm{V}(X)} = \sqrt{70.8} \approx 8.4$$

# 분산과 표준편차

DATA KUBWA



# 다중공선성 (MULTI-COLLINEARITY)

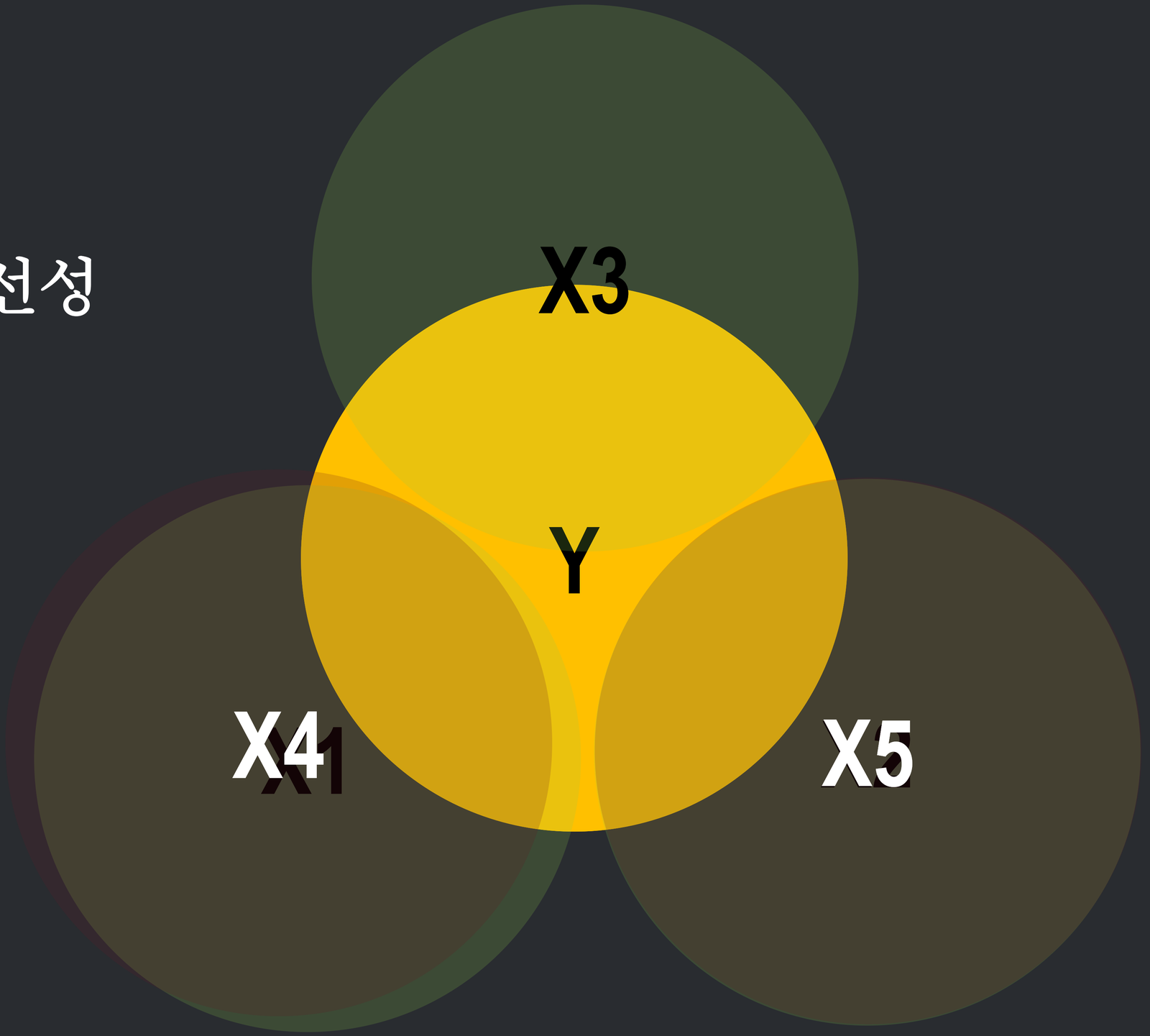
DATA KUBWA

- 다중공선성(Multi-collinearity) 문제: 다수의 독립변수가 서로 지나치게 높은 상관관계를 가지면서 회귀계수 추정의 오류가 발생하는 문제

- 분산팽창계수(VIF, Variance Inflation Factor)를 구하여 판단

- 엄밀한 기준은 없으나 보통 10보다 크면 다중공선성이 있다고 판단(5를 기준으로 하기도 함)

# 다중공선성



1. 산포도 & 상관계수 확인

   너무 높은 상관계수 (약 0.9이상)은 다중공선성 의심

2. Tolerance를 확인

   한 개의 독립변수를 종속변수로 나머지 독립변수를 독립변수로 하는 회귀분석 실시

   R^2 가 1이라면 독립변수 간에 심각한 상관관계가 있음을 의미

   Tolerance = 1 – R^2 → Tolerance가 0에 가까워질수록 상관성이 매우 높다

3. 분산팽창지수 (VIF: Variance Inflation Factor)

   VIF = 1 / tolerance = 1 / (1-R^2)

   VIF가 크다는 것은 다중공선성이 크다

   일반적으로 10보다 크면 문제가 있다고 판단 (연속형 변수)

   더미변수가 3보다 크면 문제 (범주형 변수)

해결책

1. 애초에 일어나지 않게 독립 변수를 잘 선택 (무조건 넣지 않는다)

2. 비슷한 피처가 연구목적 피처가 아닌 경우 피처 삭제

3. 주성분 분석으로 변수를 재조합 (경우에 따라 이상한 결과)

4. 다중공선성 발생 독립변수 합체 (평균 기준. 그러나 유의하게 나왔을 때 해석은 어떻게?)

5. 릿지 리그레션

6. Mean centring

# 실습

Wine quality 데이터로 linear regression 분석

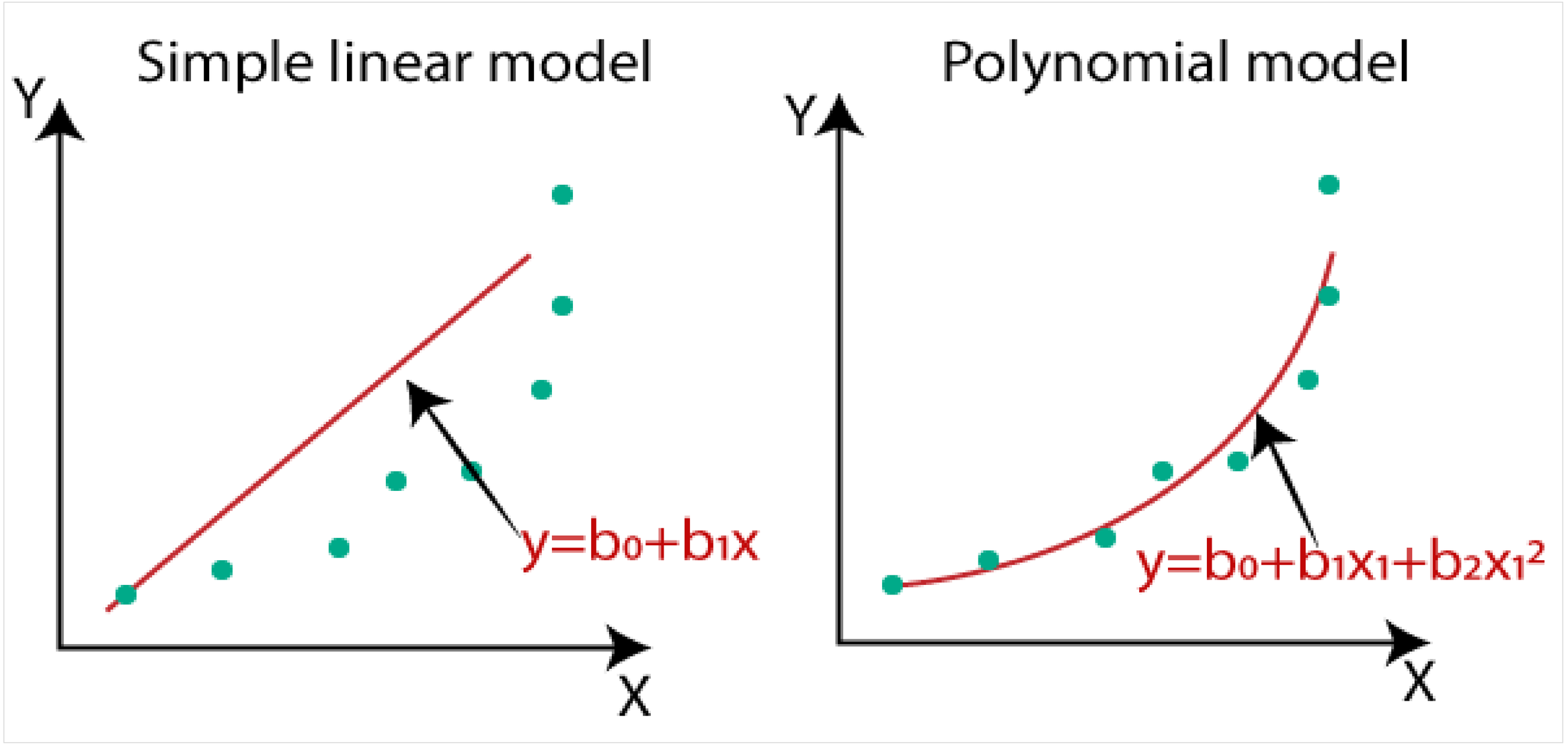
# 다항회귀

지금까지는?

$$y = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 + \ldots + w_{13} x_{13}$$

# 다항회귀

DATA KUBWA



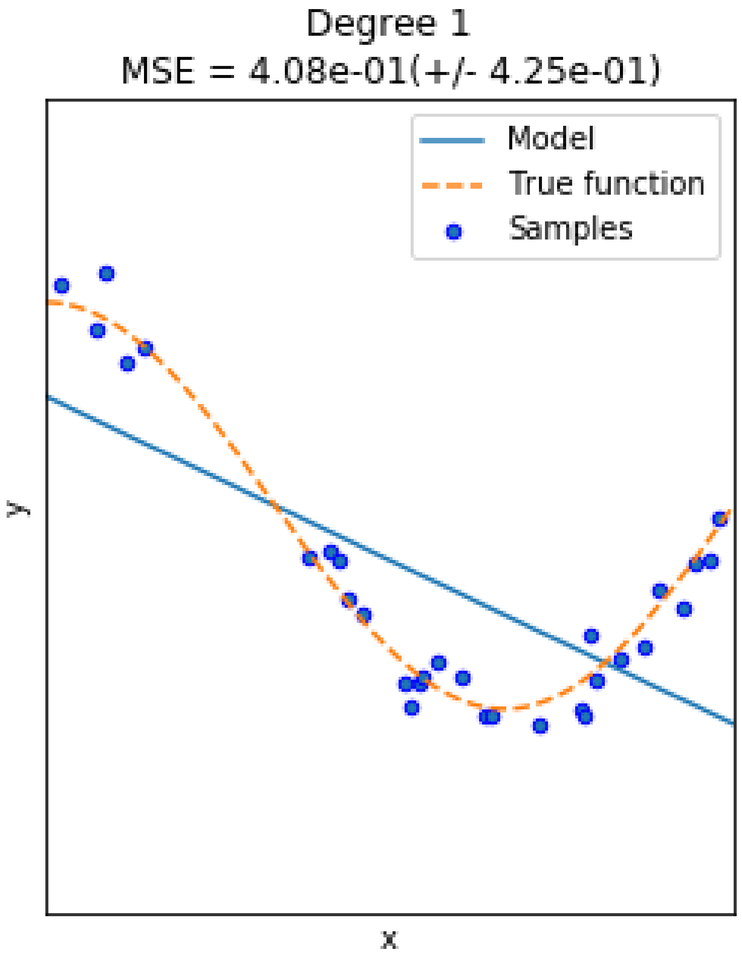
# 다항회귀

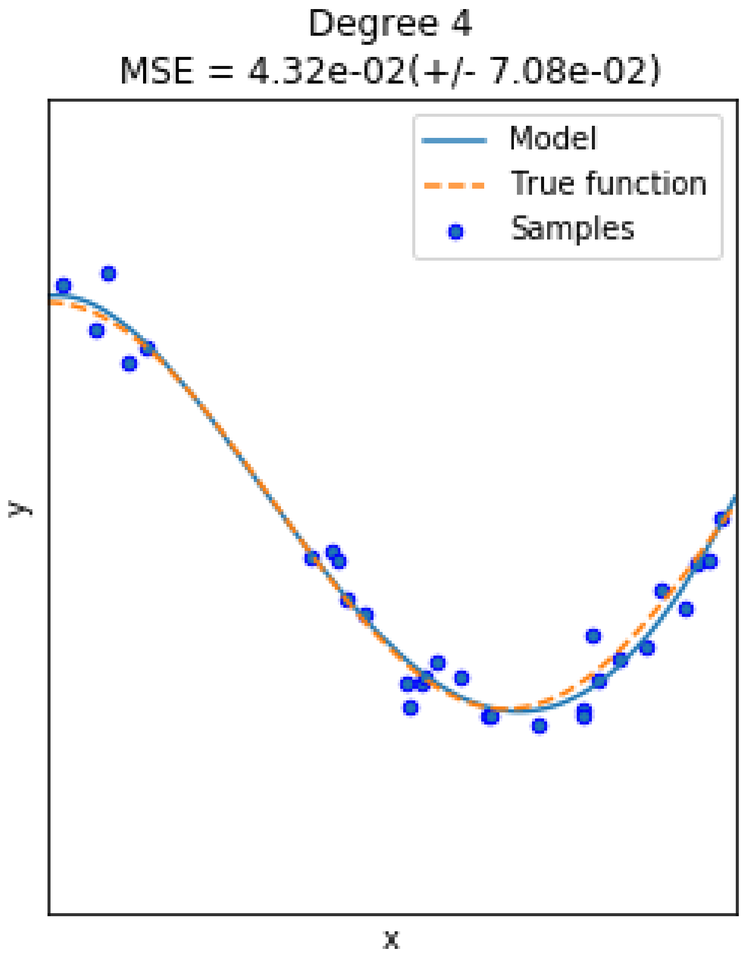
$$y = w_0 + w_1 x_1 + w_2 x_2 + w_3 \; x_1 {}^{*} w_3 \; x_2 \; + w_4 \; x_1{}^{2} + w_5 \; x_2{}^{2}$$

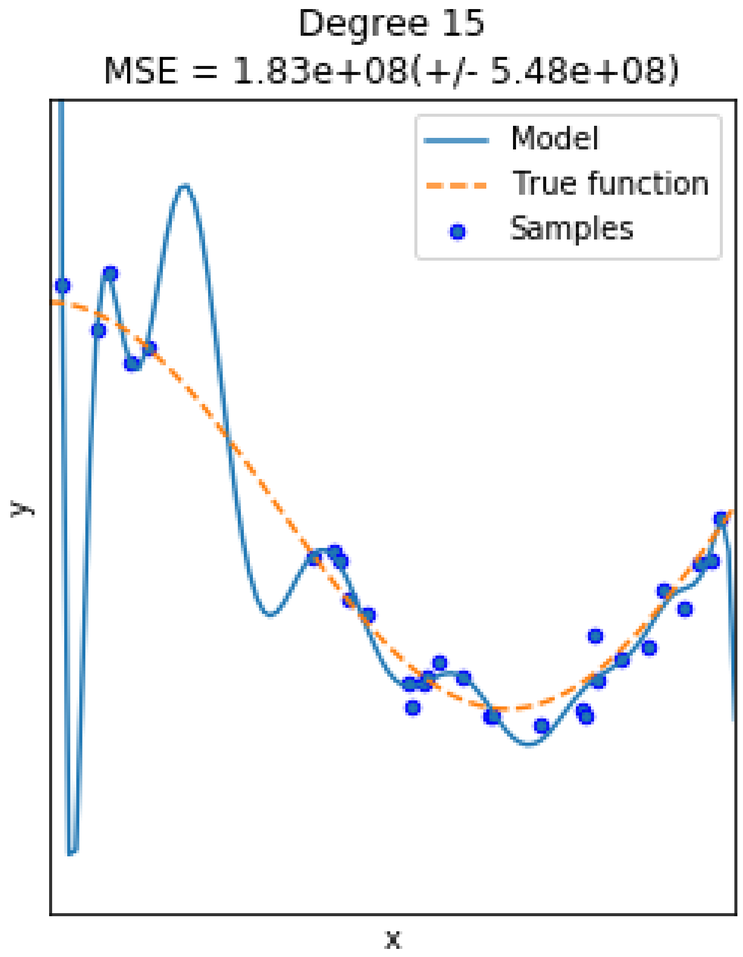
```
from sklearn.preprocessing import PolynomialFeatures
```

## High Bias

## High Variance




0.407

0.043

182,815,432

# 편향-분산 트레이드오프 (BIAS-VARIANCE TRADE OFF)

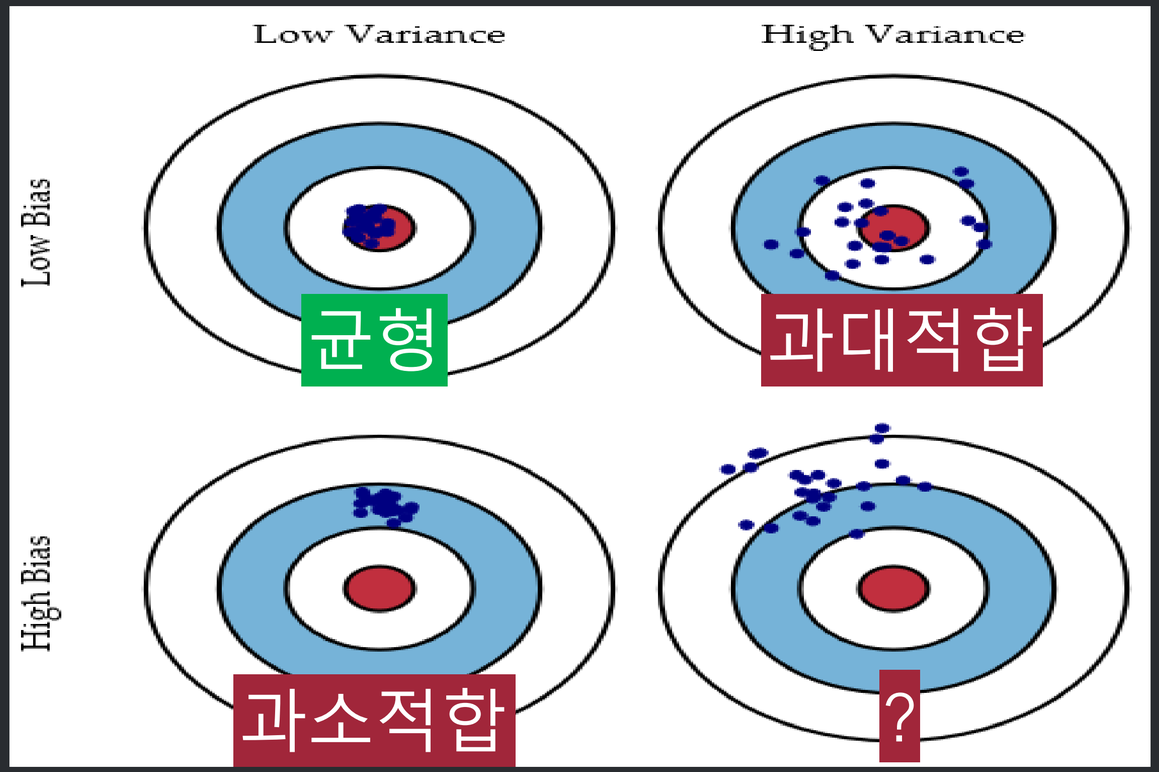
DATA KUBWA

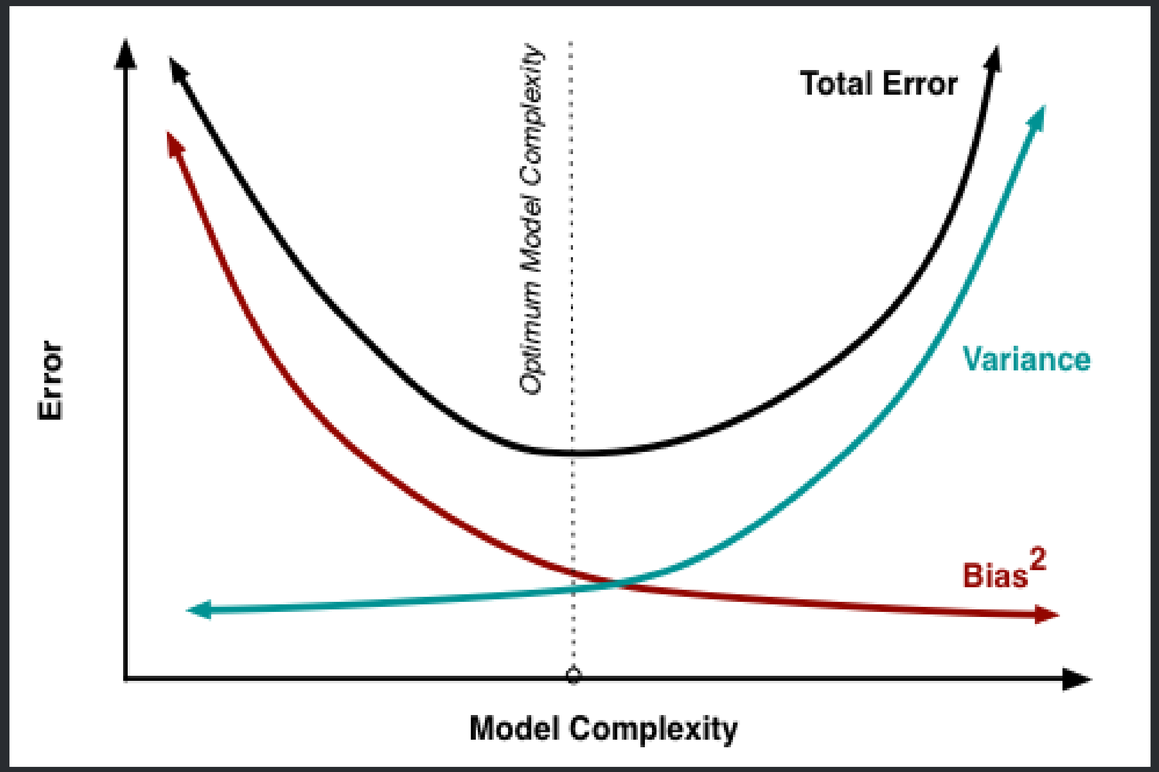
- **Bias (편향성)**

- 지도학습 알고리즘이 학습데이터 내 입력변수들과 출력변수의 관계를 잘 **fitting**하지 못해 발생하는 오차

- **Variance (변동성)**

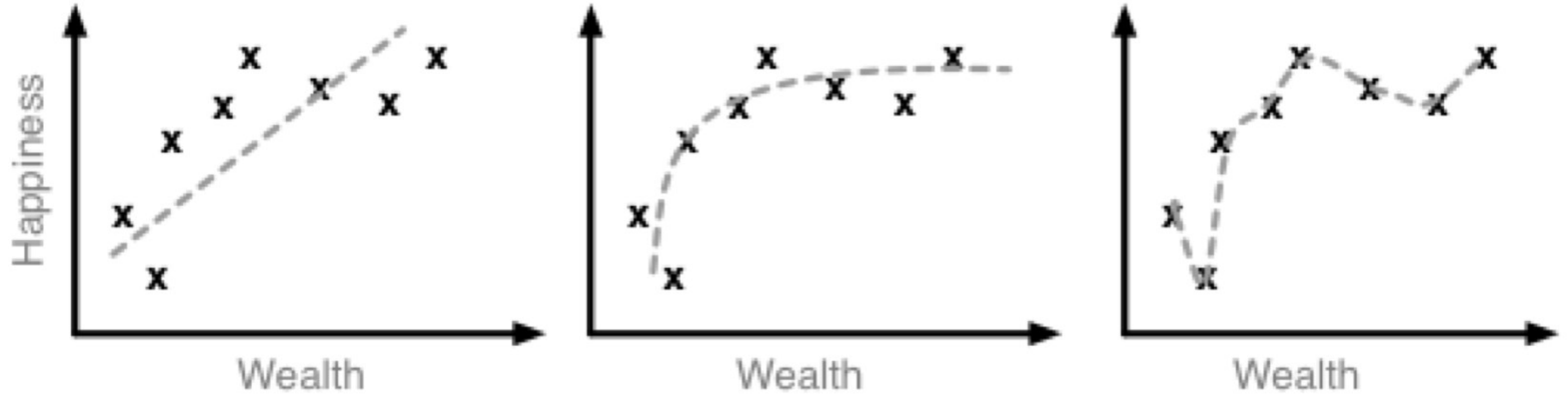
- 학습데이터에 내재되어 있는 변동**(fluctuation)**에 의해 발생하는 오차
- 학습데이터가 모집단을 완벽하게 대표할 수 없기 때문에 발생

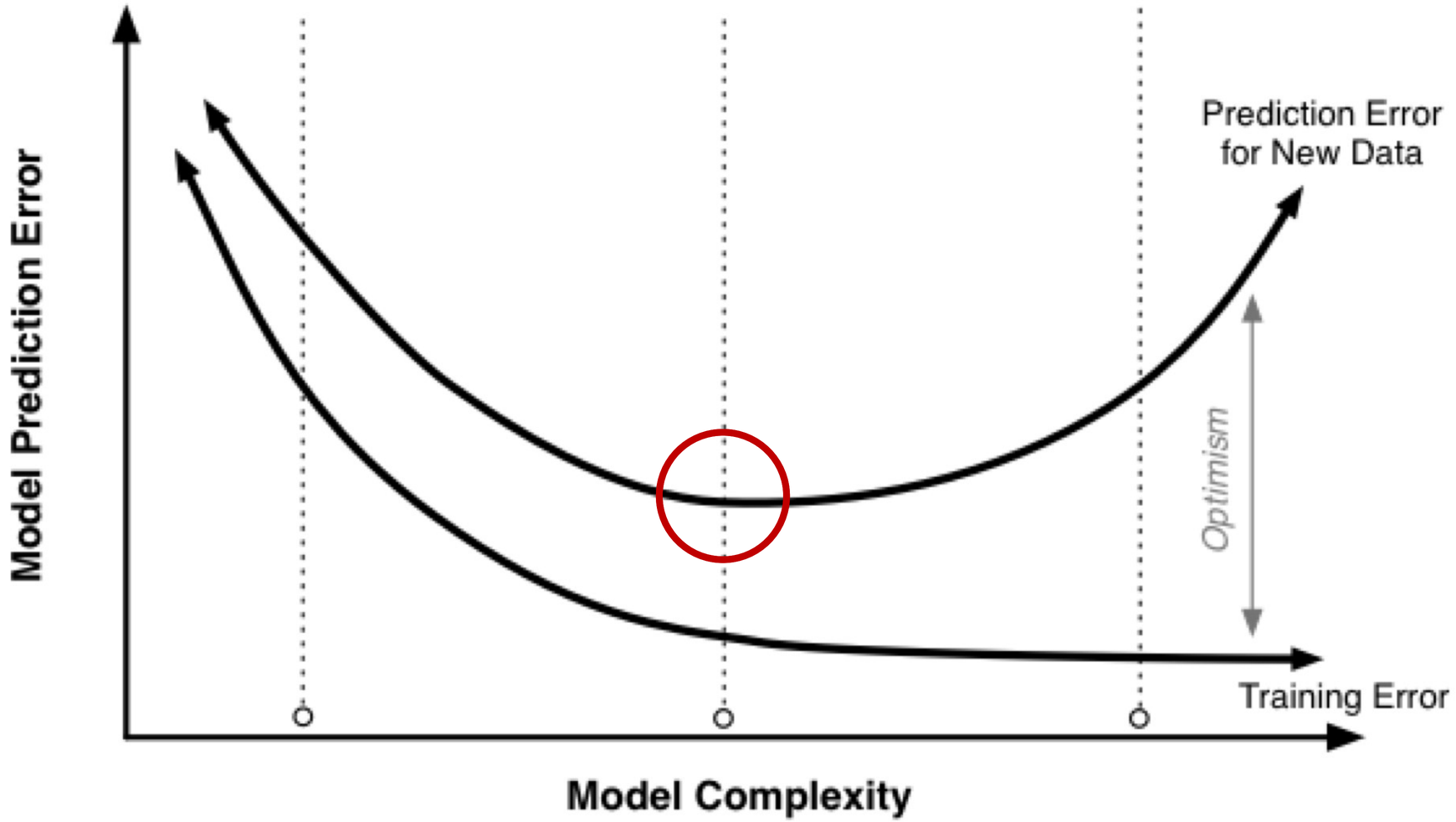

Low Variance
High Variance
Low Bias
High Bias
균형
과대적합
과소적합
?


Total Error
Optimum Model Complexity
Variance
Bias$^2$
Error
Model Complexity

과소적합 균형 과대적합

# 예 제


Happiness
Wealth
Wealth
Wealth

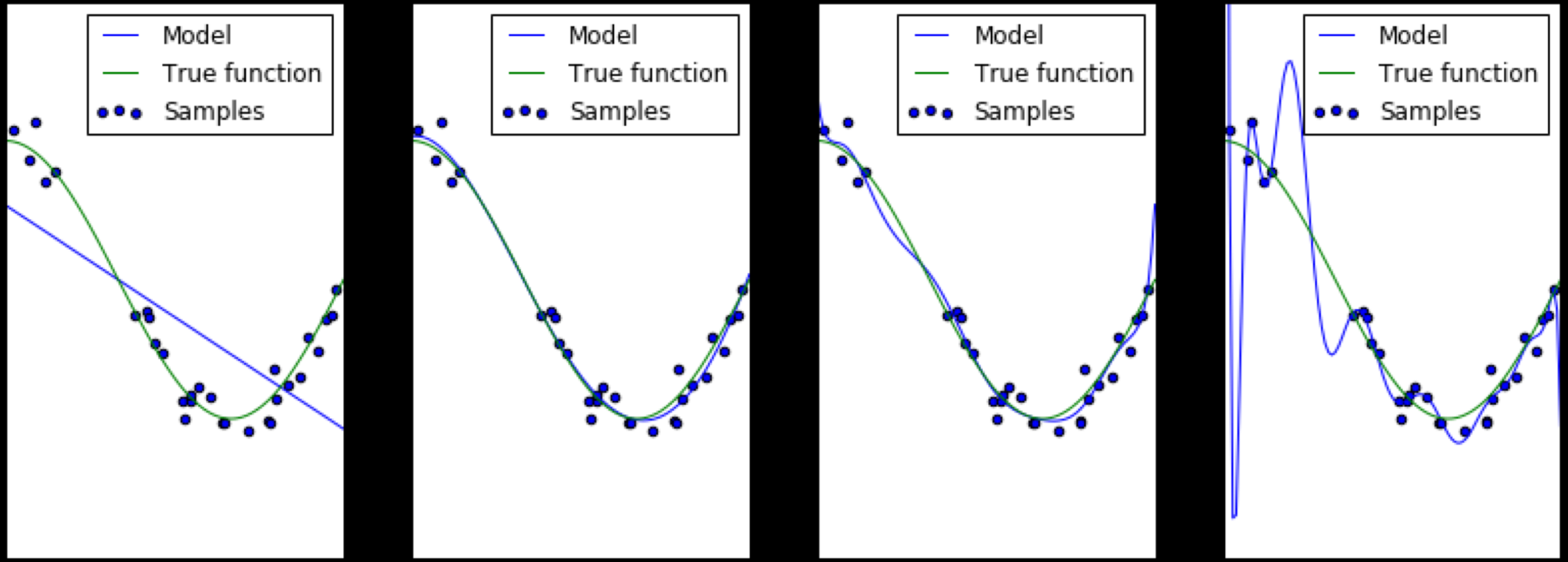

Model Prediction Error
Prediction Error
for New Data
Optimism
Training Error
Model Complexity

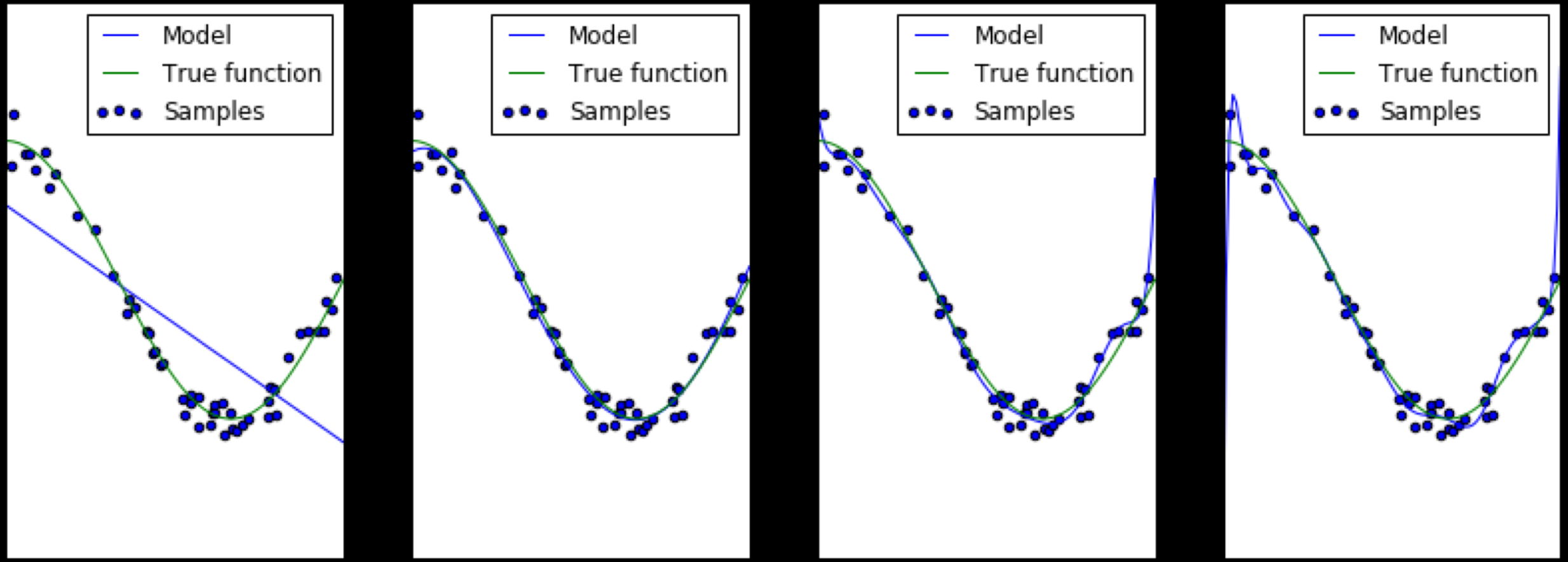
1. 편향 감소 (Underfitting 모델인 경우)

- 더 많은 피처 추가 (파생변수 생성, 데이터 통합 등)
- 더욱 세련된 알고리즘 사용 (모델에 복잡성을 더한다)

2. 분산 감소 (Overfitting 모델인 경우)

- 더 적은 피처 사용
- 더 많은 데이터 포인트 사용
- 제약 (regularization) 사용


Model
True function
Samples
Model
True function
Samples
Model
True function
Samples
Model
True function
Samples


Model
True function
Samples
Model
True function
Samples
Model
True function
Samples
Model
True function
Samples

1. 편향 감소 (Underfitting 모델인 경우)

- 더 많은 피처 추가 (파생변수 생성, 데이터 통합 등)
- 더욱 세련된 알고리즘 사용 (모델에 복잡성을 더한다)

2. 분산 감소 (Overfitting 모델인 경우)

- 더 적은 피처 사용
- 더 많은 데이터 포인트 사용
- 제약 (regularization) 사용

Question: 다음 set 중 제약(Regularization)에 사용해야 할 데이터 셋은?

Test set

Validation set

Training set

- 모델의 overfitting 방지
- 모델의 복잡도 축소
- 모델의 파라미터 탐색

지금까지 회귀모델의 목적은?

Cost function을 최소화하는 것!

지금부터는?

# 규제 (Regularization)


$$\text{Min} ( \text{RSS}(w) + \text{alpha} * || w ||_2^2 )$$

비용함수 최소화 + 회귀계수크기제어

Linear Regression

+

규제 (Regularization)

L1 규제 Rasso Regression

L2 규제 Ridge Regression

# Ridge Regression

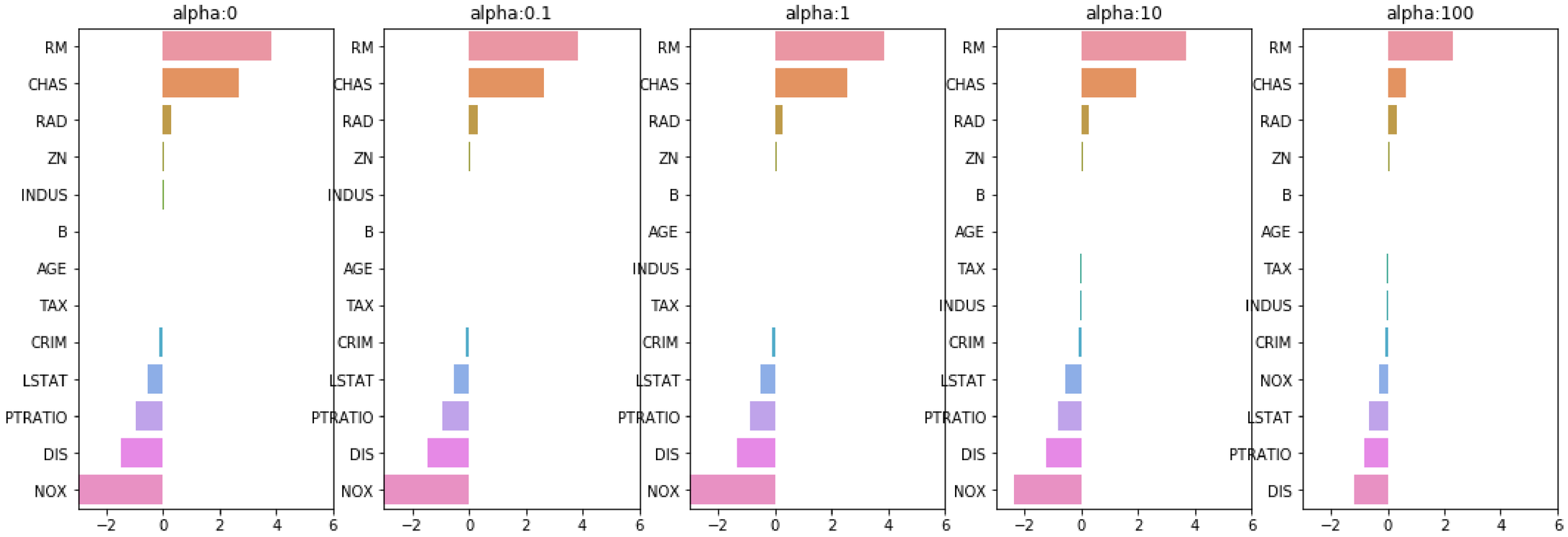
```
from sklearn.linear_model import Ridge

ridge = Ridge(alpha = 10)
```

5 folds 의 개별 Negative MSE scores:  [-12.46 -26.05 -33.07 -80.76 -33.31]
5 folds 의 개별 RMSE scores :  [3.53 5.1  5.75 8.99 5.77]
5 folds 의 평균 RMSE : 5.829

5 folds 의 개별 Negative MSE scores:  [-11.422 -24.294 -28.144 -74.599 -28.517]
5 folds 의 개별 RMSE scores :  [3.38  4.929 5.305 8.637 5.34 ]
5 folds 의 평균 RMSE : 5.518

alpha 0 일 때 5 folds 의 평균 RMSE : 5.829
alpha 0.1 일 때 5 folds 의 평균 RMSE : 5.788
alpha 1 일 때 5 folds 의 평균 RMSE : 5.653
alpha 10 일 때 5 folds 의 평균 RMSE : 5.518
alpha 100 일 때 5 folds 의 평균 RMSE : 5.330


alpha:0
RM
CHAS
RAD
ZN
INDUS
B
AGE
TAX
CRIM
LSTAT
PTRATIO
DIS
NOX
−2
0
2
4
6
alpha:0.1
RM
CHAS
RAD
ZN
INDUS
B
AGE
TAX
CRIM
LSTAT
PTRATIO
DIS
NOX
−2
0
2
4
6
alpha:1
RM
CHAS
RAD
ZN
B
AGE
INDUS
TAX
CRIM
LSTAT
PTRATIO
DIS
NOX
−2
0
2
4
6
alpha:10
RM
CHAS
RAD
ZN
B
AGE
TAX
INDUS
CRIM
LSTAT
PTRATIO
DIS
NOX
−2
0
2
4
6
alpha:100
RM
CHAS
RAD
ZN
B
AGE
TAX
INDUS
CRIM
NOX
LSTAT
PTRATIO
DIS
−2
0
2
4
6

| | alpha:0 | alpha:0.1 | alpha:1 | alpha:10 | alpha:100 |
|---|---|---|---|---|---|
| RM | 3.809865 | 3.818233 | 3.854000 | 3.702272 | 2.334536 |
| CHAS | 2.686734 | 2.670019 | 2.552393 | 1.952021 | 0.638335 |
| RAD | 0.306049 | 0.303515 | 0.290142 | 0.279596 | 0.315358 |
| ZN | 0.046420 | 0.046572 | 0.047443 | 0.049579 | 0.054496 |
| INDUS | 0.020559 | 0.015999 | -0.008805 | -0.042962 | -0.052826 |
| B | 0.009312 | 0.009368 | 0.009673 | 0.010037 | 0.009393 |
| AGE | 0.000692 | -0.000269 | -0.005415 | -0.010707 | 0.001212 |
| TAX | -0.012335 | -0.012421 | -0.012912 | -0.013993 | -0.015856 |
| CRIM | -0.108011 | -0.107474 | -0.104595 | -0.101435 | -0.102202 |
| LSTAT | -0.524758 | -0.525966 | -0.533343 | -0.559366 | -0.660764 |
| PTRATIO | -0.952747 | -0.940759 | -0.876074 | -0.797945 | -0.829218 |
| DIS | -1.475567 | -1.459626 | -1.372654 | -1.248808 | -1.153390 |
| NOX | -17.766611 | -16.684645 | -10.777015 | -2.371619 | -0.262847 |

# Rasso Regression

```
from sklearn.linear_model import Lasso
```

alpha 0 일 때 5 folds 의 평균 RMSE : 5.829

alpha 0.1 일 때 5 folds 의 평균 RMSE : 5.788

alpha 1 일 때 5 folds 의 평균 RMSE : 5.653

alpha 10 일 때 5 folds 의 평균 RMSE : 5.518

alpha 100 일 때 5 folds 의 평균 RMSE : 5.330


L2: Ridge 규제

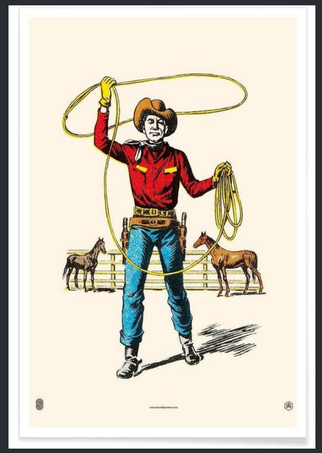
alpha 0.07일 때 5 폴드 세트의 평균 RMSE: 5.612

alpha 0.1일 때 5 폴드 세트의 평균 RMSE: 5.615

alpha 0.5일 때 5 폴드 세트의 평균 RMSE: 5.669

alpha 1일 때 5 폴드 세트의 평균 RMSE: 5.776

alpha 3일 때 5 폴드 세트의 평균 RMSE: 6.189



L1: Lasso 규제


| | alpha:0 | alpha:0.1 | alpha:1 | alpha:10 | alpha:100 |
|---|---|---|---|---|---|
| RM | 3.809865 | 3.818233 | 3.854000 | 3.702272 | 2.334536 |
| CHAS | 2.686734 | 2.670019 | 2.552393 | 1.952021 | 0.638335 |
| RAD | 0.306049 | 0.303515 | 0.290142 | 0.279596 | 0.315358 |
| ZN | 0.046420 | 0.046572 | 0.047443 | 0.049579 | 0.054496 |
| INDUS | 0.020559 | 0.015999 | -0.008805 | -0.042962 | -0.052826 |
| B | 0.009312 | 0.009368 | 0.009673 | 0.010037 | 0.009393 |
| AGE | 0.000692 | -0.000269 | -0.005415 | -0.010707 | 0.001212 |
| TAX | -0.012335 | -0.012421 | -0.012912 | -0.013993 | -0.015856 |
| CRIM | -0.108011 | -0.107474 | -0.104595 | -0.101435 | -0.102202 |
| LSTAT | -0.524758 | -0.525966 | -0.533343 | -0.559366 | -0.660764 |
| PTRATIO | -0.952747 | -0.940759 | -0.876074 | -0.797945 | -0.829218 |
| DIS | -1.475567 | -1.459626 | -1.372654 | -1.248808 | -1.153390 |
| NOX | -17.766611 | -16.684645 | -10.777015 | -2.371619 | -0.262847 |

| | alpha:0.07 | alpha:0.1 | alpha:0.5 | alpha:1 | alpha:3 |
|---|---|---|---|---|---|
| RM | 3.789725 | 3.703202 | 2.498212 | 0.949811 | 0.000000 |
| CHAS | 1.434343 | 0.955190 | 0.000000 | 0.000000 | 0.000000 |
| RAD | 0.270936 | 0.274707 | 0.277451 | 0.264206 | 0.061864 |
| ZN | 0.049059 | 0.049211 | 0.049544 | 0.049165 | 0.037231 |
| B | 0.010248 | 0.010249 | 0.009469 | 0.008247 | 0.006510 |
| NOX | -0.000000 | -0.000000 | -0.000000 | -0.000000 | 0.000000 |
| AGE | -0.011706 | -0.010037 | 0.003604 | 0.020910 | 0.042495 |
| TAX | -0.014290 | -0.014570 | -0.015442 | -0.015212 | -0.008602 |
| INDUS | -0.042120 | -0.036619 | -0.005253 | -0.000000 | -0.000000 |
| CRIM | -0.098193 | -0.097894 | -0.083289 | -0.063437 | -0.000000 |
| LSTAT | -0.560431 | -0.568769 | -0.656290 | -0.761115 | -0.807679 |
| PTRATIO | -0.765107 | -0.770654 | -0.758752 | -0.722966 | -0.265072 |
| DIS | -1.176583 | -1.160538 | -0.936605 | -0.668790 | -0.000000 |

# 선형 회귀 모델

DATA KUBWA

- 일반선형회귀
- 릿지 (Ridge) = 일반선형회귀 + L2 규제
- 라쏘 (Lasso) = 일반선형회귀 + L1 규제
- 엘라스틱넷 (ElasticNet) = 일반선형회귀 + L1 규제 + L2 규제
- 로지스틱 회귀 (Logistic Regression)

# ElasticNet Regression

L1 + L2


parameter: alpha, l1_ratio

$$a * L1 + b * L2$$

$$alpha = a + b$$

$$l1\_ratio = a/(a+b)$$

# 선형 회귀 모델

DATA KUBWA

- 엘라스틱넷 (ElasticNet) = 일반선형회귀 + L1 규제 + L2 규제
- 로지스틱 회귀 (Logistic Regression)