

• 03/29 Simple Linear Regression 구현
(프로그래밍 실기)

① Training Data Set 준비 → Data pre-processing (데이터 전처리)
사용할 있는 형태로 준비

② Linear Regression Model을 정의 $y = wx + b$ → Model (예측 모델)
hypothesis (가설)

③ 최적의 w, b 를 구하려면 $\left[\begin{array}{l} \text{loss function (손실함수)} \\ \text{cost function (비용함수)} \end{array} \right] \rightarrow \underline{\underline{MSE}}$
weight bias

④ Gradient Descent Algorithm (경사하강법)

loss function을 편미분 $(w, b) \times \text{learning rate}$

$$w' = w - \alpha \cdot \frac{\partial E(w, b)}{\partial w}$$

우리가 설정!! 0.001
 $1e^{-3}$

⑤ 학습 진행 → "반복학습을 진행"

예측치

$\hat{y} = Wx + b$ (Hypothesis, model)

시간(x)	성적(y)
1	3
2	5
3	7
4	9
5	11

Training Data Set

예측치

$Wx_1 + b = y_1$
 $Wx_2 + b = y_2$
 $Wx_3 + b = y_3$
 $Wx_4 + b = y_4$
 $Wx_5 + b = y_5$

matrix

행렬곱연산으로
최적에게인

$x_1 W + b$
 $x_2 W + b$
 $x_3 W + b$
 $x_4 W + b$
 $x_5 W + b$

x_1
 x_2
 x_3
 x_4
 x_5

(5x1)

matrix
multiplication
(행렬곱)

W

$(W) + b$

(1x1)

y
 $x_1 W + b$
 $x_2 W + b$
 $x_3 W + b$
 $x_4 W + b$
 $x_5 W + b$

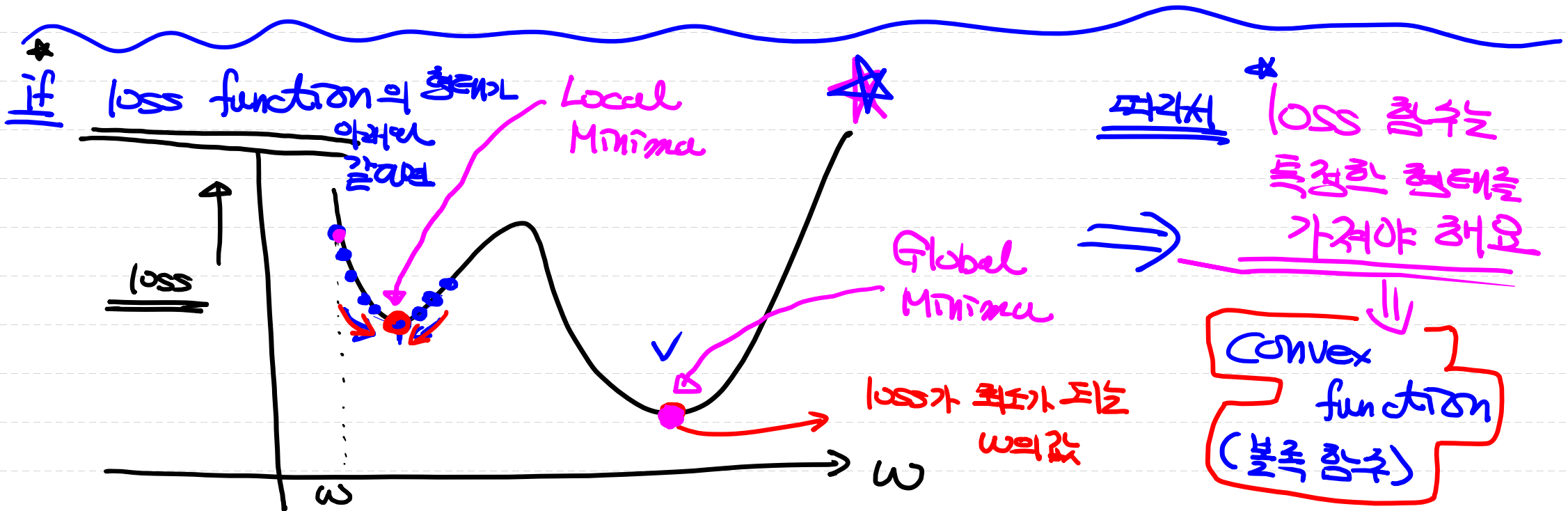
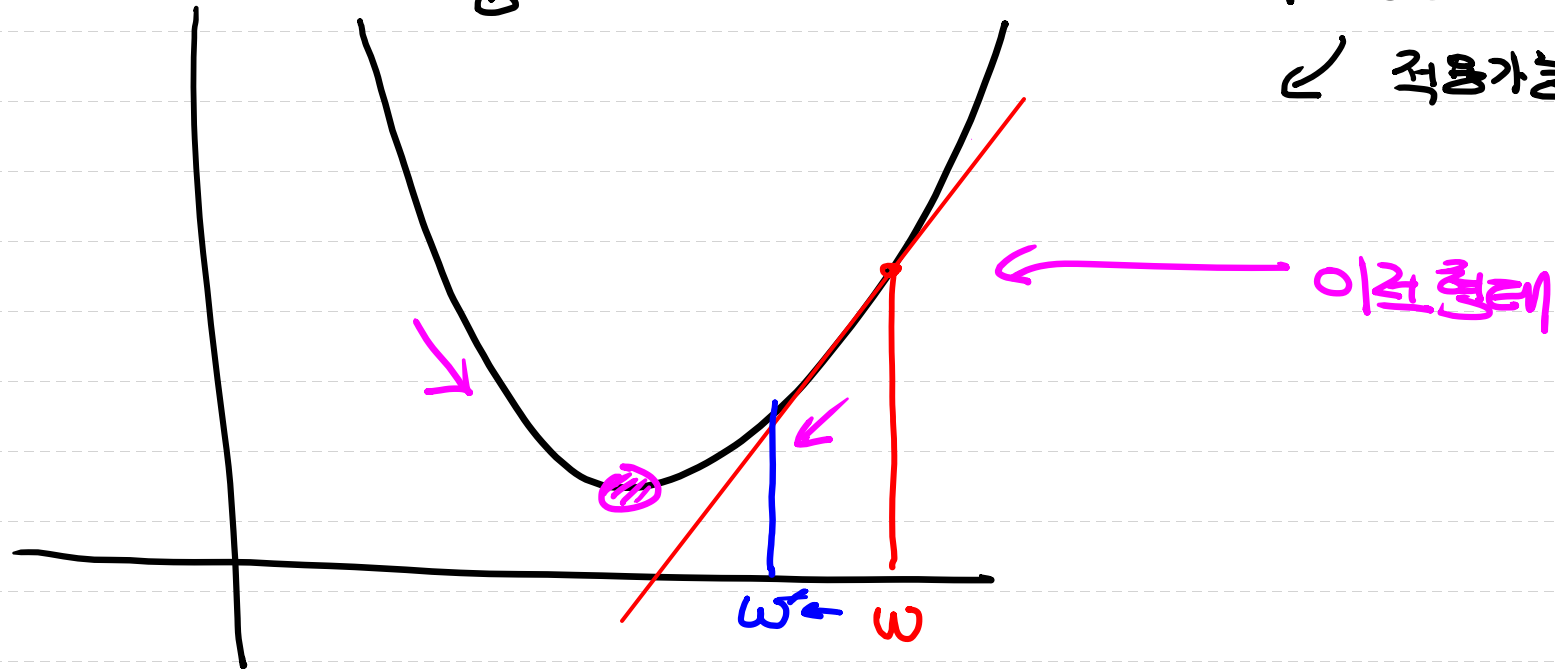
x_1
 x_2
 x_3
 x_4
 x_5

$y = X \cdot W + b$
 2차원 M 2차원 M

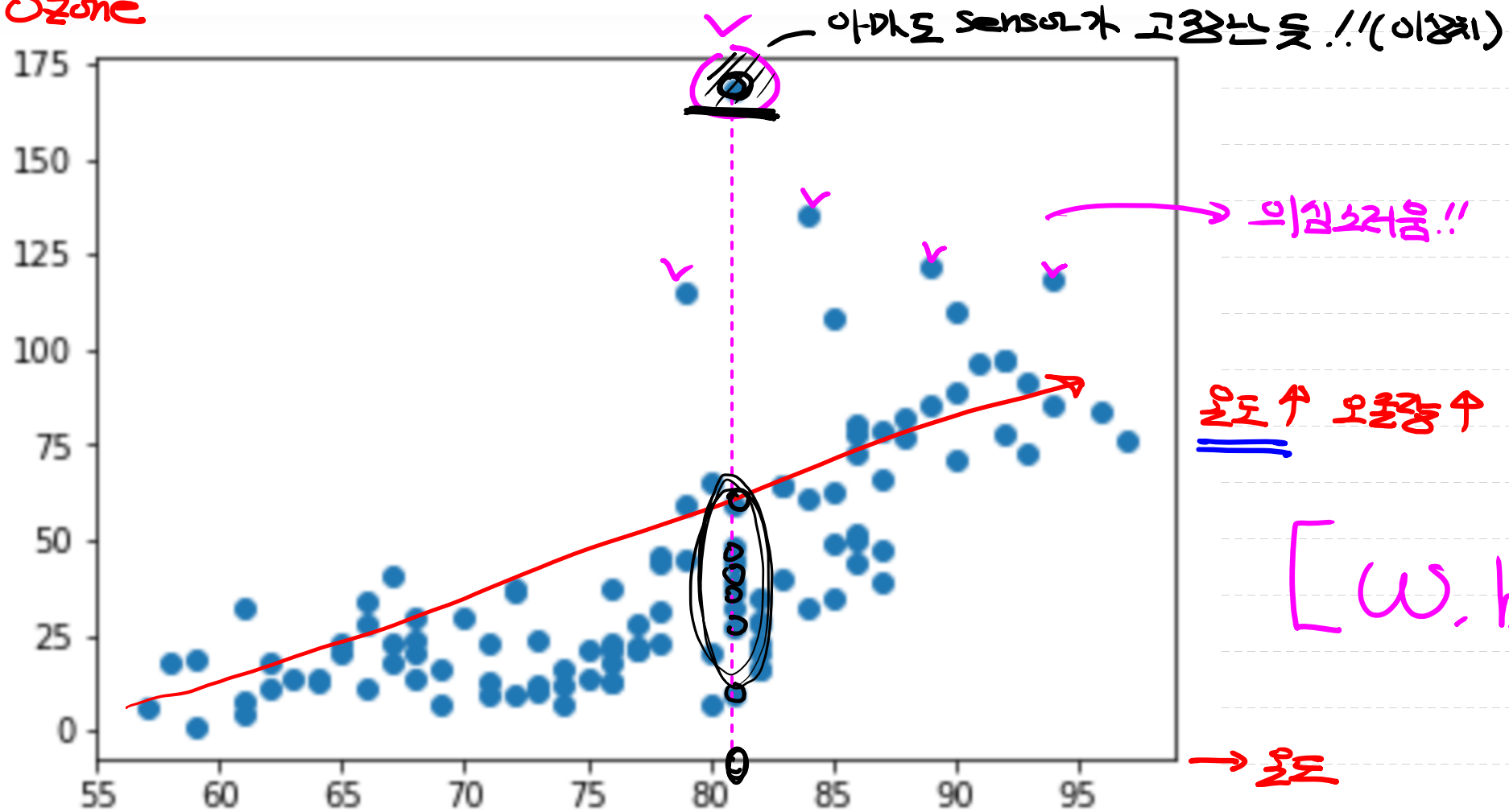
• loss function ↘

Gradient Descent

↙ 적용가능

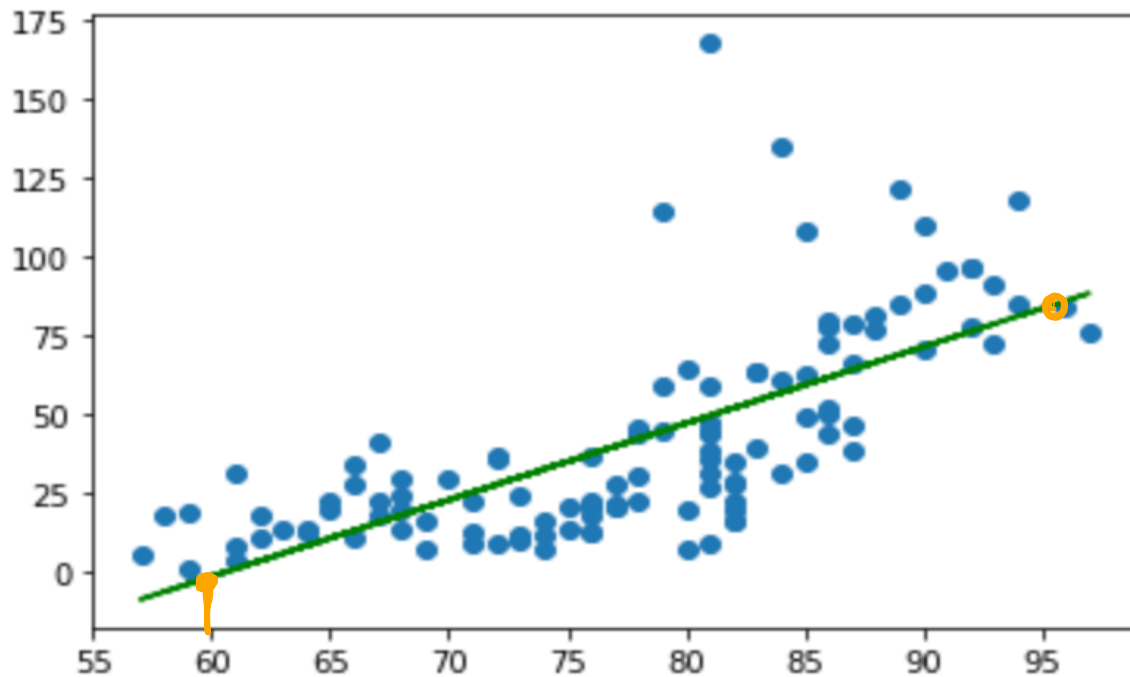
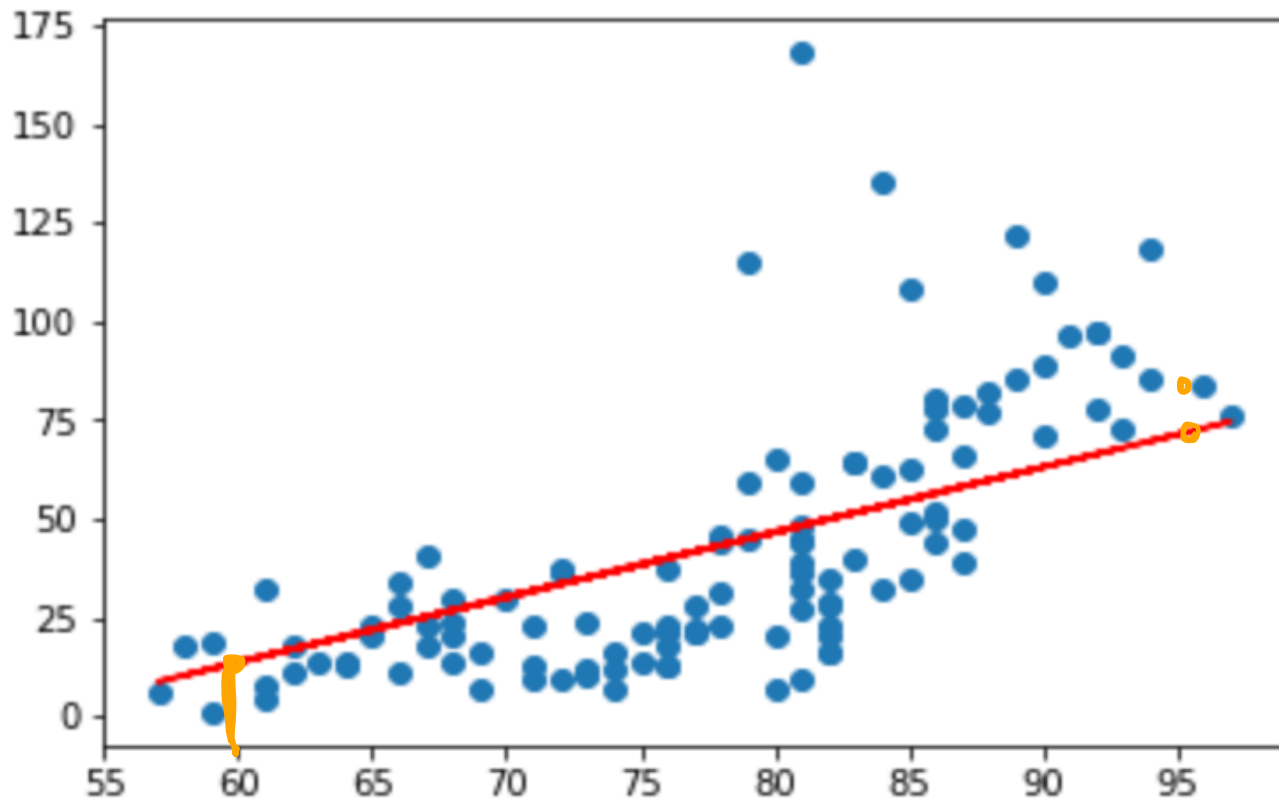


Ozone



→ 온도

$$\underline{y} = \underline{w}x + \underline{b} \text{ "직선"}$$



★★★★★
Data 처리가
잘 안되었어요!!

⇒ python 구현 ★

Why?

⇒ Sklearn 구현 ★

→ 가설기가 심해요

① Data Pre-processing (데이터 전처리)

① 결측값 (Missing Value) 처리

삭제
대체

↓
4월