# Machine Learning

## Chapter 4 지도 학습(Supervised Learning)

START

스마트인재개발원
Smart Human Resources Development

# 학습목표

- 데이터 스케일링의 필요성을 이해 할 수 있다.
- 다양한 스케일링 방법을 알 수 있다.
- 선형회귀 모델을 이해하고 사용 할 수 있다.
- 회귀 모델의 평가방법을 알 수 있다.

# 데이터 스케일링
# (Data scaling)

# 데이터 스케일링(Data scaling) 정의

## 데이터 스케일링 (Data scaling)

- 특성(Feature)들의 범위(range)를 정규화 해주는 작업
- 특성마다 다른 범위를  가지는 경우 머신러닝 모델들이 제대로 학습되지 않을 가능성이 있다.
  (KNN, SVM, Neural network 모델, Clustering 모델 등)

| 시력 | 키 |
|---|---|
| 0.2 | 178 |
| 1.0 | 156 |
| 0.5 | 168 |
| 0.3 | 188 |
| 0.6 | 149 |

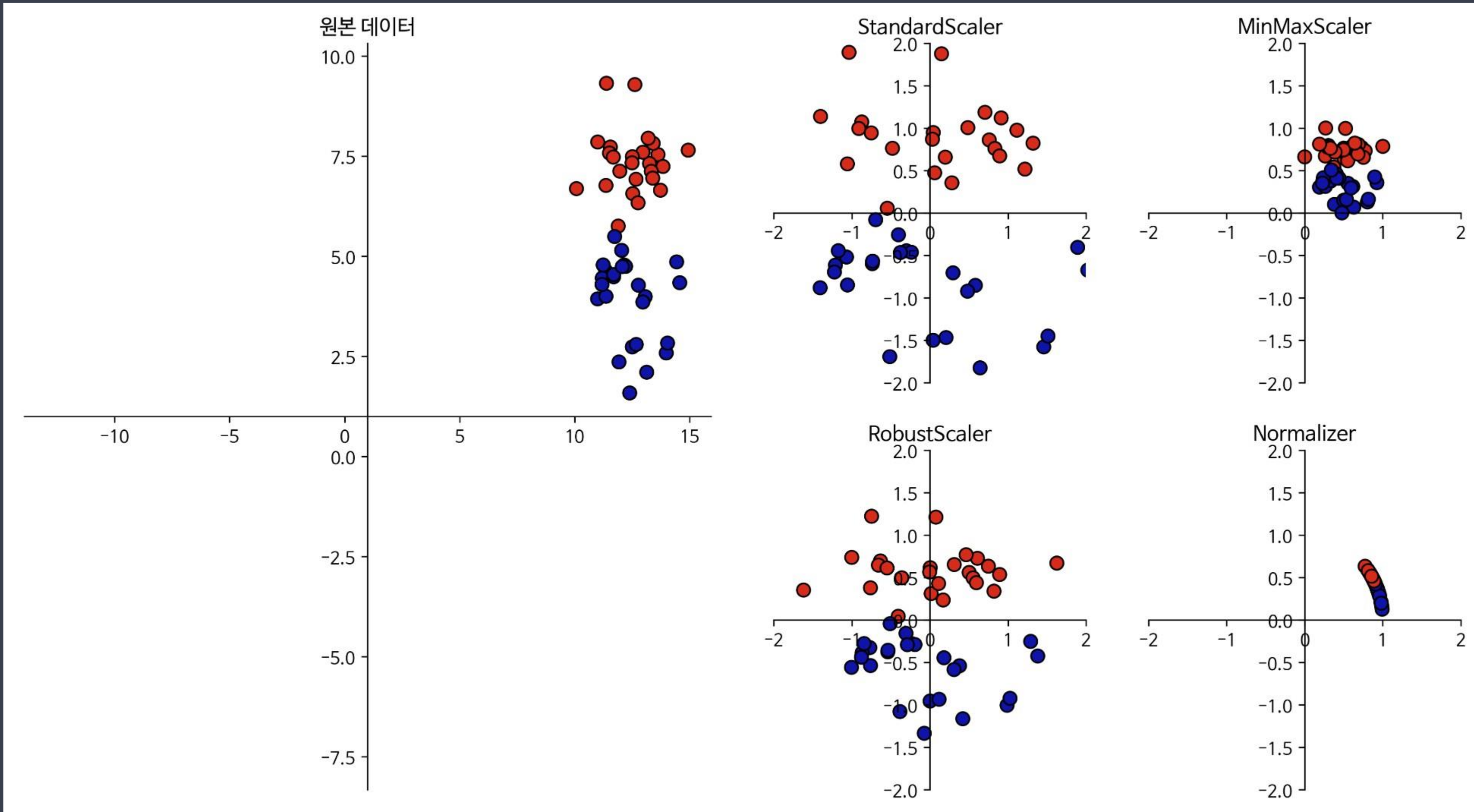
시력과 키를 함께 학습시킬 경우
키의 범위가 크기때문에 거리 값을
기반으로 학습 할 때 영향을 많이 준다.

# 데이터 스케일링(Data scaling) 장점

## 장점

- 특성들을 비교 분석하기 쉽게 만들어 준다.
- Linear Model, Neural network Model 등에서 학습의 안정성과 속도를 개선시킨다.
- 하지만 특성에 따라 원래 범위를 유지하는게 좋을 경우는 scaling을 하지 않아도 된다.

# 데이터 스케일링(Data scaling) 종류



# 데이터 스케일링(Data scaling) 종류

## StandardScaler

- 변수의 평균,표준편차를 이용해 정규분포 형태로 변환 (평균 0, 분산 1)
- 이상치(Outlier)에 민감하게 영향을 받는다.

## RobustScaler

- 변수의 사분위수를 이용해 변환
- 이상치(Outlier)가 있는 데이터 변환시 사용 할 수 있다.

## MinMaxScaler

- 변수의 Max 값, Min 값을 이용해 변환 (0 ~ 1 사이 값으로 변환)
- 이상치(Outlier)에 민감하게 영향을 받는다.

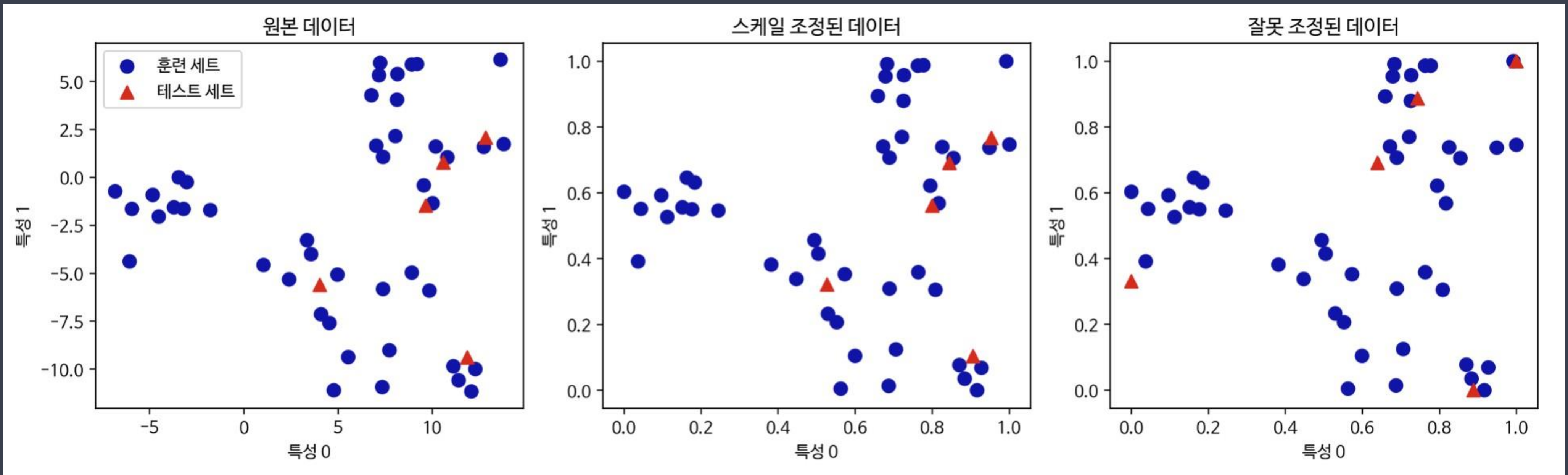
## Normalizer

- 특성 벡터의 길이가 1이 되도록 조정 (행마다 정규화 진행)
- 특성 벡터의 길이는 상관 없고 데이터의 방향(각도)만 중요할 때 사용.

# 데이터 스케일링(Data scaling) 주의점

## 주의점

- 훈련세트와 테스트세트에 같은 변환을 적용해야 한다.
- 예를 들어 StandardScaler의 경우 **훈련세트**의 **평균과 표준편차**를 이용해 **훈련세트를 변환**하고, **테스트세트**의 **평균과 표준편차**를 이용해 **테스트세트를 변환**하면 잘못된 결과가 나온다.



# Decision Tree 활용 Titanic 실습 - ex05 추가

Titanic 데이터를 학습한 KNN 모델에
scaler를 적용하여 결과 확인

# Linear Model
## (Regression)

# Linear Model

## Linear Model (선형 모델)

- 입력 특성에 대한 선형 함수를 만들어 예측을 수행
- 다양한 선형 모델이 존재한다
- 분류와 회귀에 모두 사용 가능

# Linear Model - Regression

## 회귀의 선형모델

| x(hour) | y(score) |
|---|---|
| 9 | 90 |
| 8 | 80 |
| 4 | 40 |
| 2 | 20 |

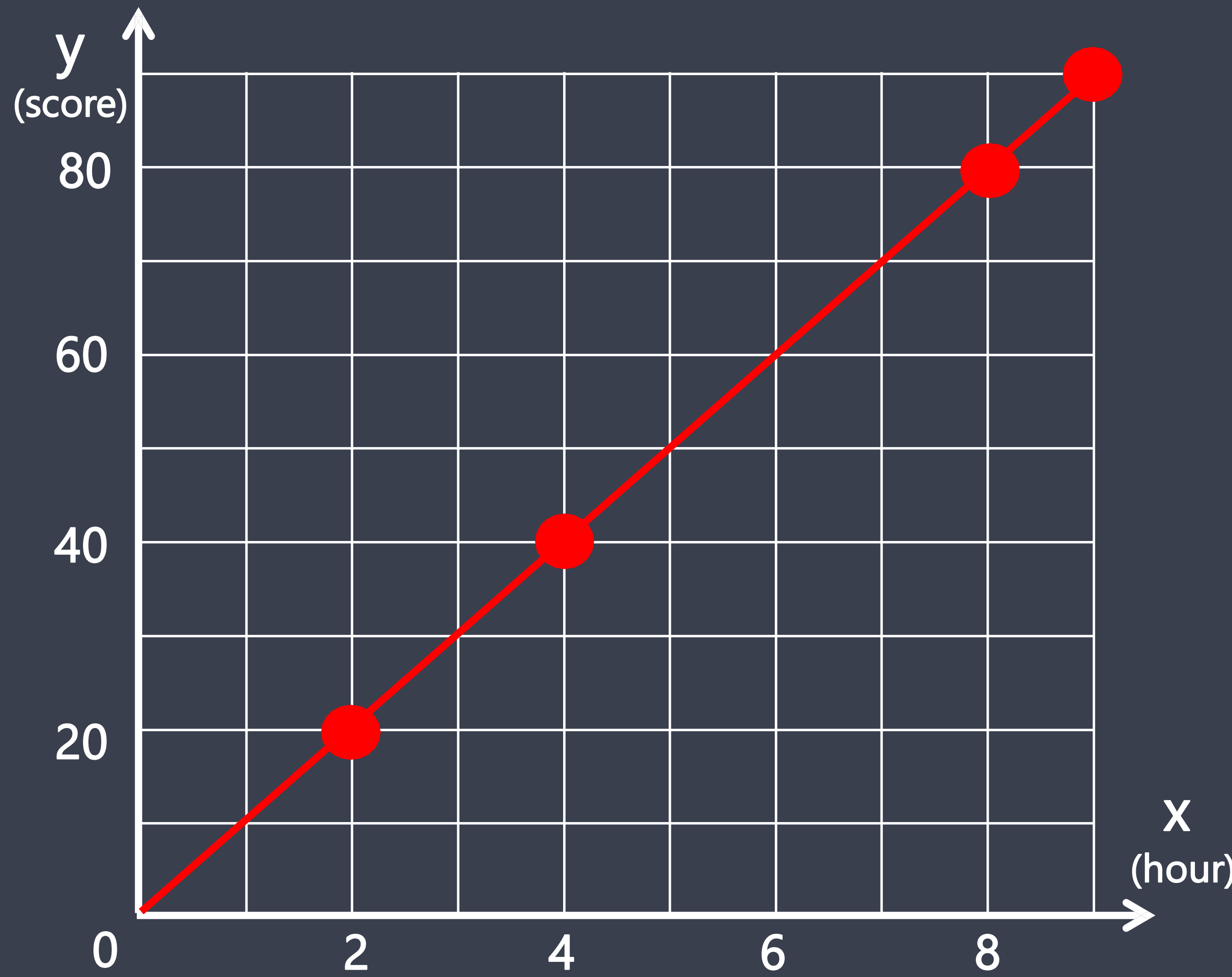
시험성적 데이터

7시간 공부 할 경우
성적은 몇 점 일까?

# Linear Model - Regression

## 시험 성적 데이터

| x(hour) | y(score) |
|---|---|
| 9 | 90 |
| 8 | 80 |
| 4 | 40 |
| 2 | 20 |

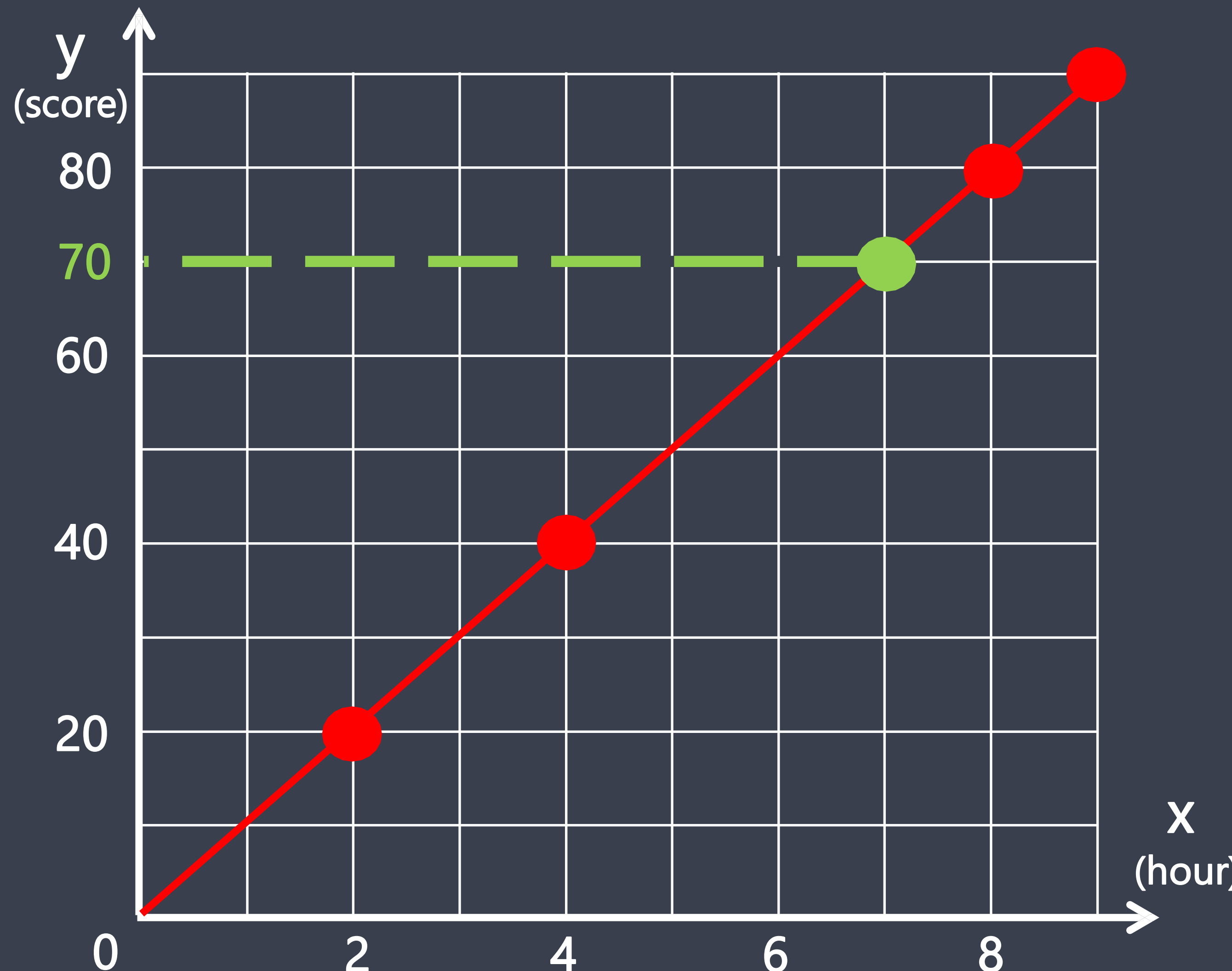


# Linear Model - Regression

시험 성적 데이터

$$y = ax + b$$

$$y = 10x + 0$$



# Linear Model - Regression

기울기 절편

$$y = ax + b$$

# Linear Model - Regression (MSE)

## 선형 회귀 함수

$$y = \boldsymbol{w_1 x_1} + \boldsymbol{w_2 x_2} + \boldsymbol{w_3 x_3} + \cdots + \boldsymbol{w_p x_p} + \boldsymbol{b}$$

- w : 가중치(weight), 계수(coefficient)
- b : 절편(intercept), 편향(bias)

- 모델 w 파라미터 : model.coef_
- 모델 b 파라미터 : model.intercept_

# Linear Model - Regression (MSE)

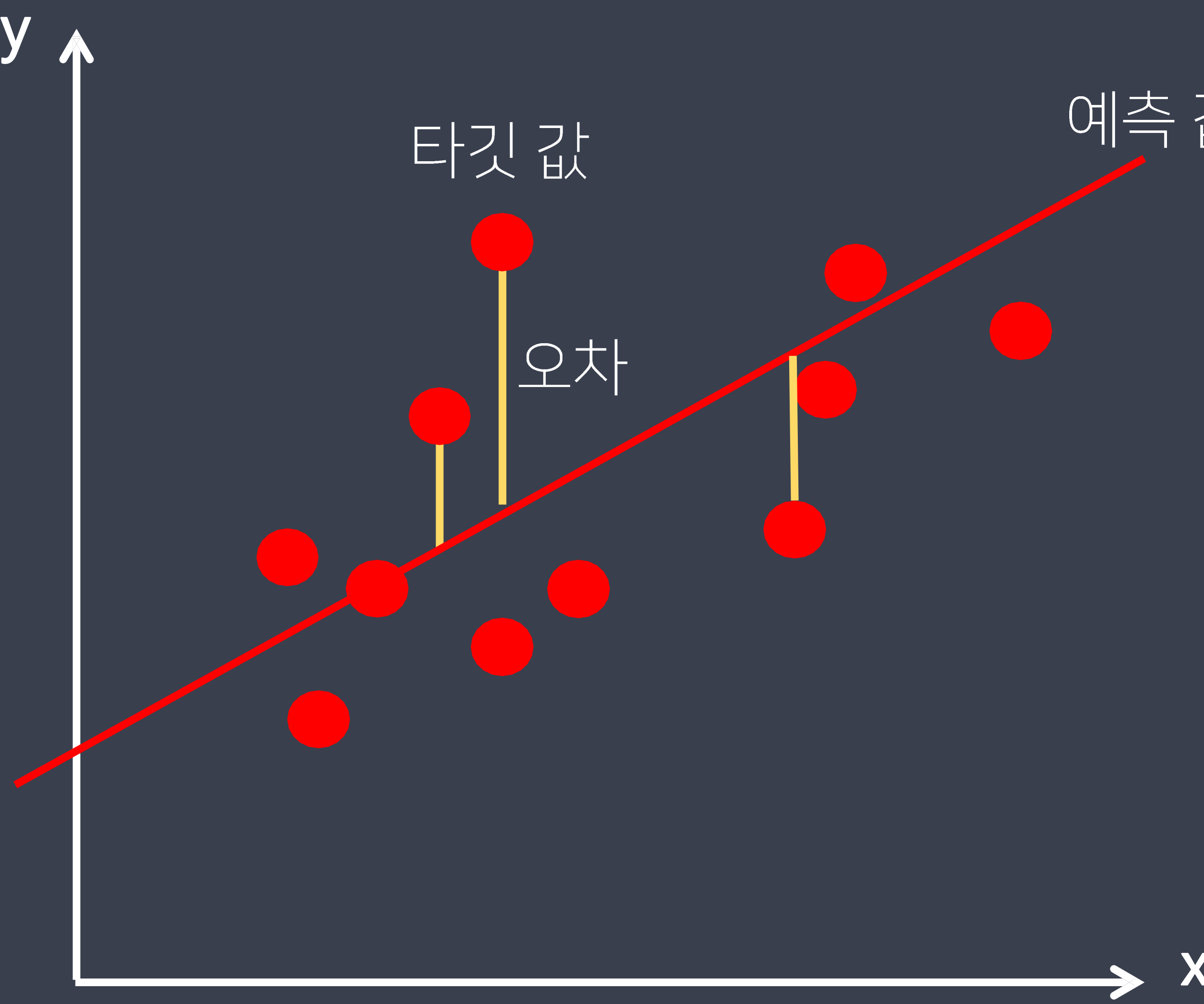
Cost function

비용함수 : 수식을 검증

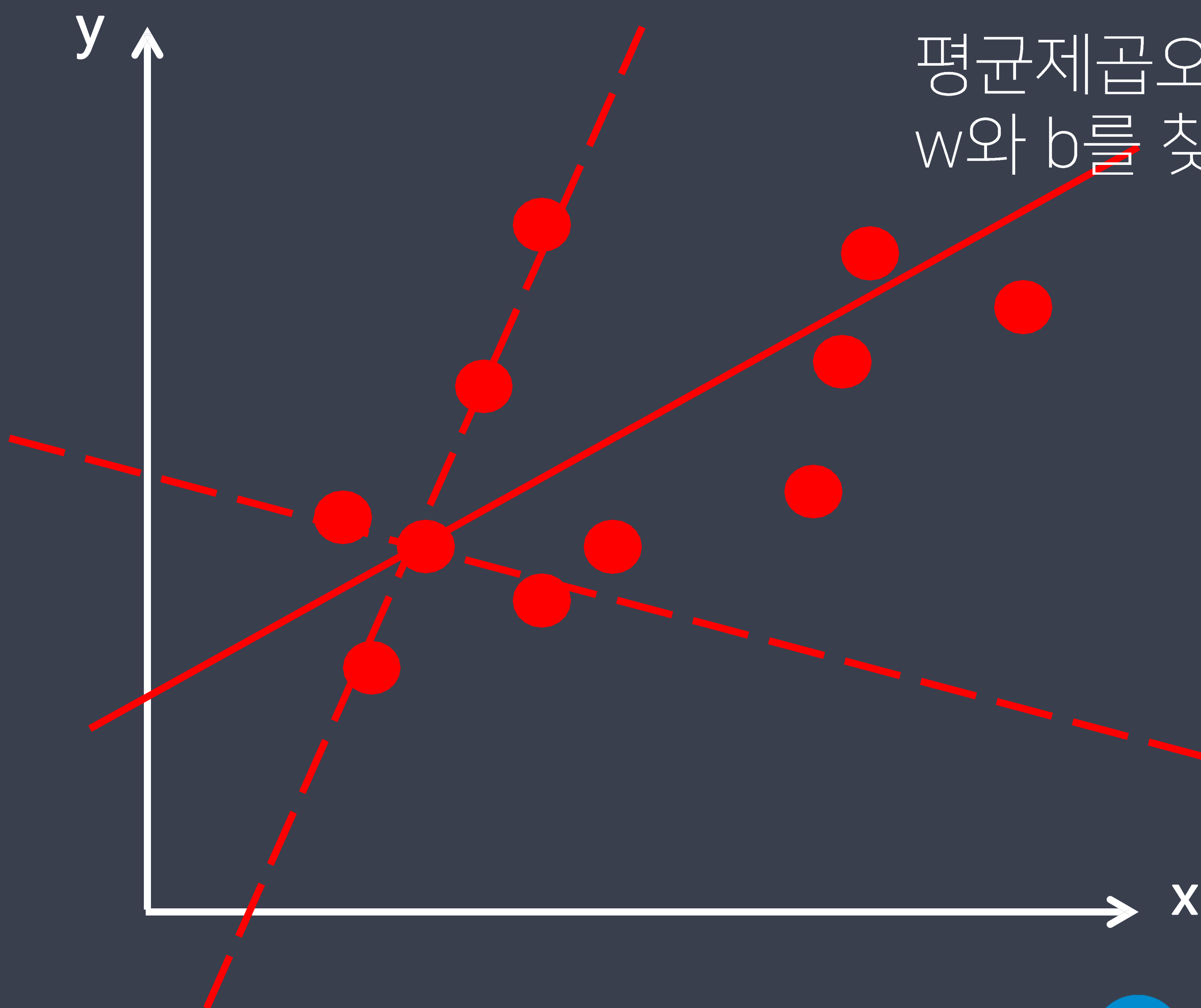
$H(x) = w * x + b$

$H(x) - y$

제곱 값 사용



# Linear Model - Regression (MSE)



# Linear Model - Regression (MSE)

평균제곱오차 (Mean Squared Error) ⬅ RMSE를 사용하기도 한다

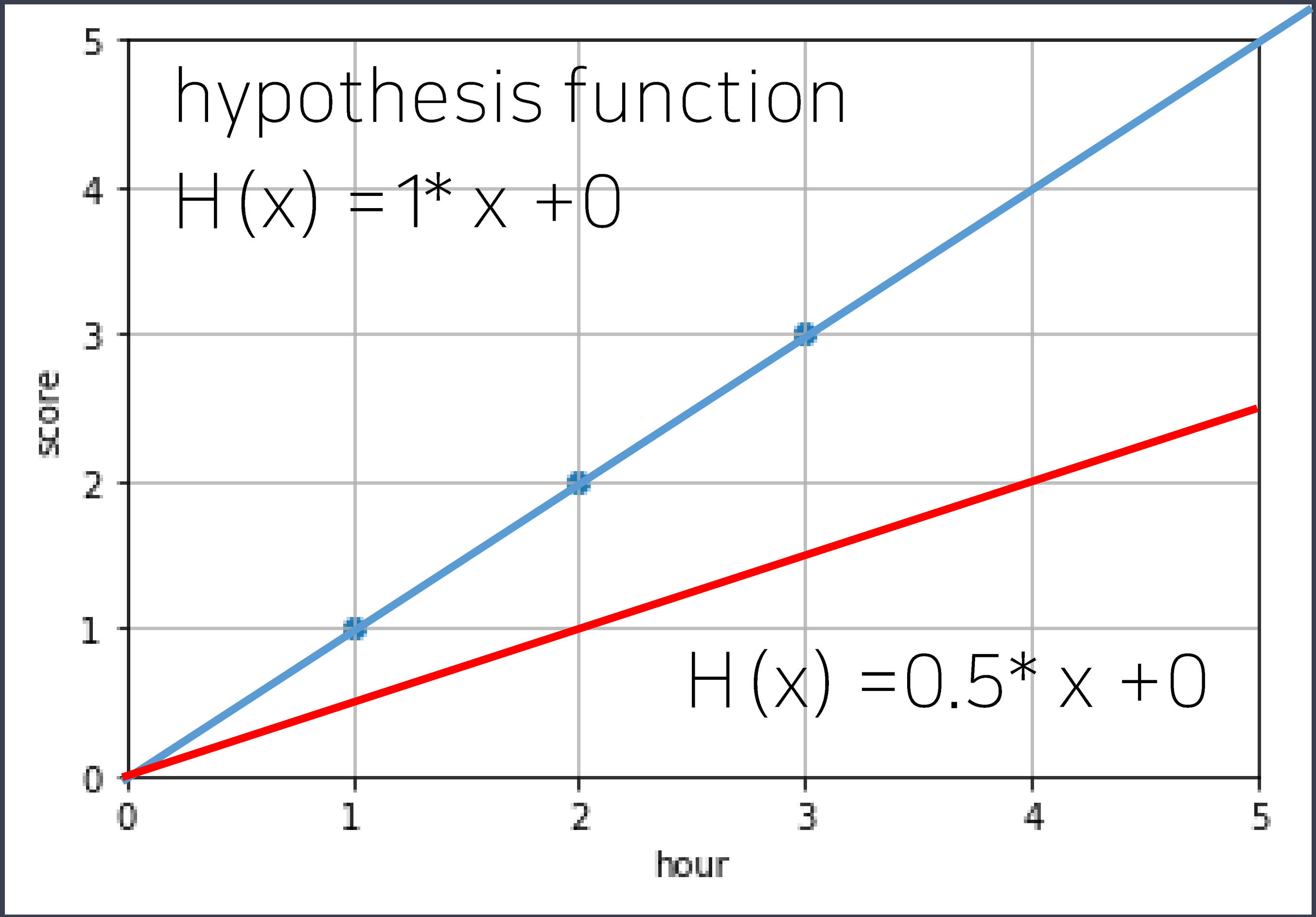
$$cost = \frac{1}{m}\sum_{i=1}^{m}\left(H(x_i) - y_i\right)^2$$

$$H(x) = Wx + b$$

# Linear Model - Regression (MSE 값 계산)

두 가설의 MSE 값을 계산해보자.

| x(hour) | y(score) |
|---|---|
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |



# Linear Model - Regression

평균제곱오차(MSE)가 최소가 되는 w와 b를 찾는 방법

1. 수학 공식을 이용한 해석적 방법 (**Ordinary Least Squares**)
2. 경사하강법 (Gradient Descent Algorithm)

# Linear Model - Regression

## 수학 공식을 이용한 해석적 방법 (Ordinary Least Squares)

$$a\sum x^2 + b\sum x = \sum xy$$

$$a\sum x + bn = \sum y$$

$$a = \frac{n\Sigma XY - \Sigma X \Sigma Y}{n\Sigma X^2 - \Sigma X \Sigma X}$$

$$b = \frac{\Sigma X^2 \Sigma Y - \Sigma X \Sigma XY}{n\Sigma X^2 - \Sigma X \Sigma X}$$

LinearRegression 클래스로 구현되어 있다.

# Linear Model - Regression

## 수학 공식을 이용한 해석적 방법 (Ordinary Least Squares)

| x(hour) | y(score) |
|---|---|
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |

$$a = \frac{n\Sigma XY - \Sigma X \Sigma Y}{n\Sigma X^2 - \Sigma X \Sigma X}$$

$$b = \frac{\Sigma X^2 \Sigma Y - \Sigma X \Sigma XY}{n\Sigma X^2 - \Sigma X \Sigma X}$$

LinearRegression 클래스로 구현되어 있다.

# Linear Model - Regression 실습 : ex06

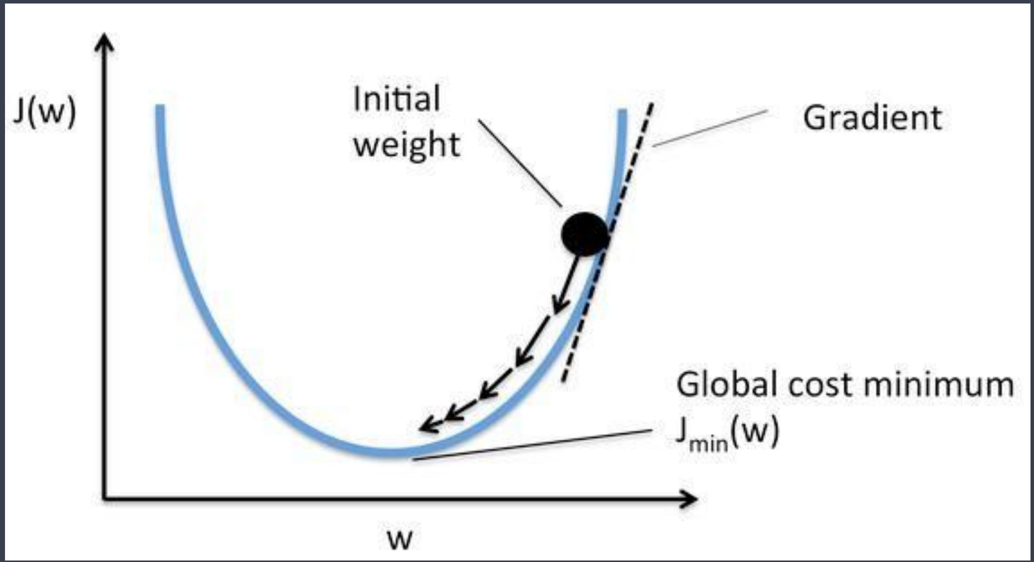
## LinearRegression 사용하기

# Linear Model - Regression(Gradient descent algorithm)

## 경사하강법 (Gradient Descent Algorithm)

# Linear Model - Regression(Gradient descent algorithm)

## 경사하강법 (Gradient Descent Algorithm)



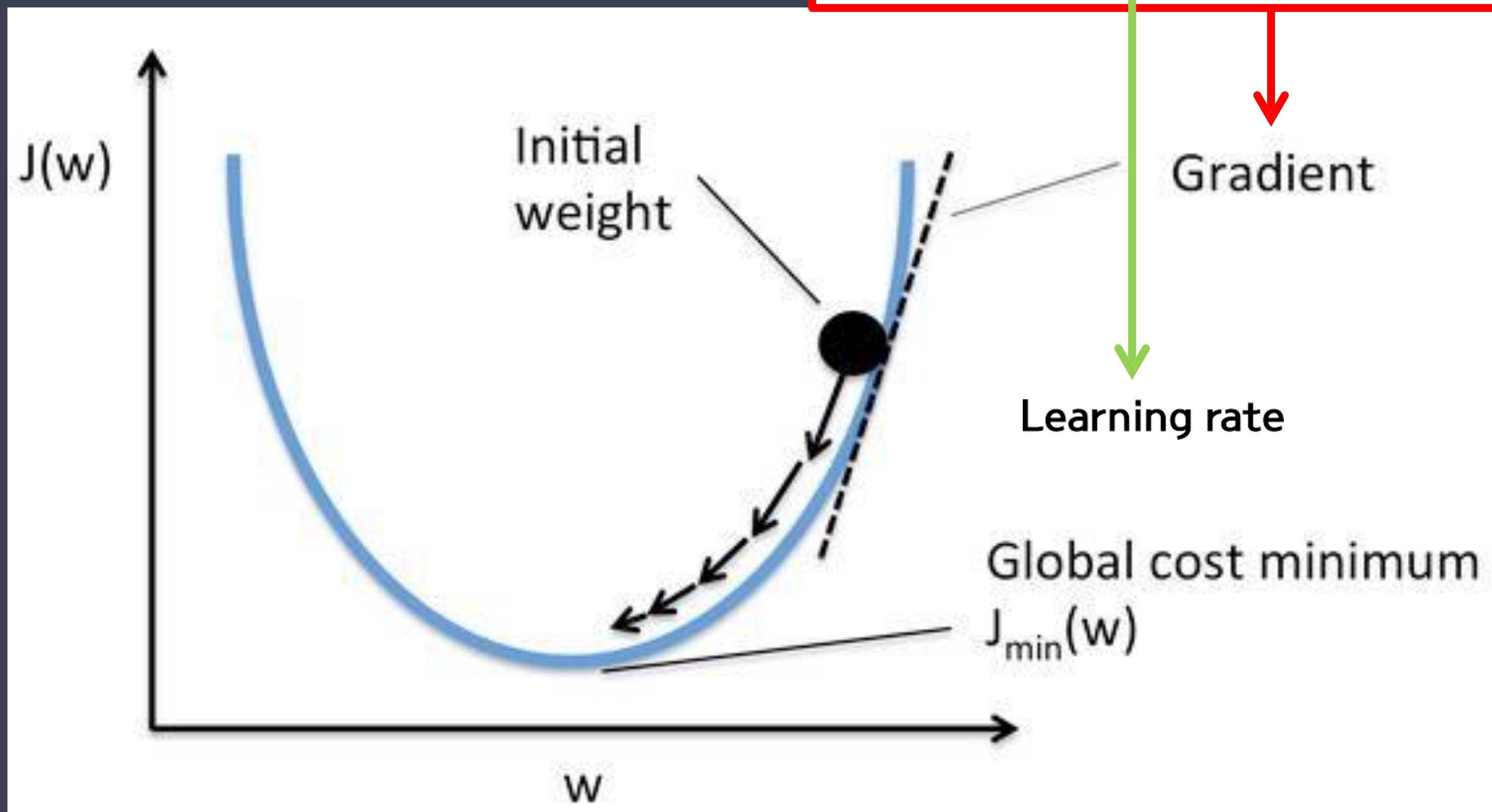

비용함수의 기울기(경사)를 구하여 기울기가 낮은 쪽으로
계속 이동하여 값을 최적화 시키는 방법

# Linear Model - Regression 실습 : ex06 추가

가중치 변화에 따른 비용함수 값의 변화를
그래프로 그려보자.

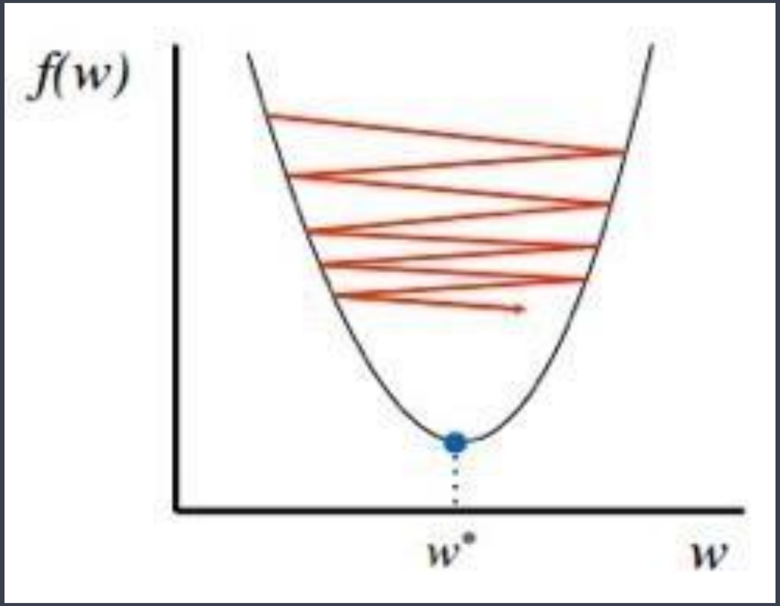
# Linear Model - Regression(Gradient descent algorithm)

$$W := W - \boxed{\alpha} \frac{\partial}{\partial W} cost(W)$$

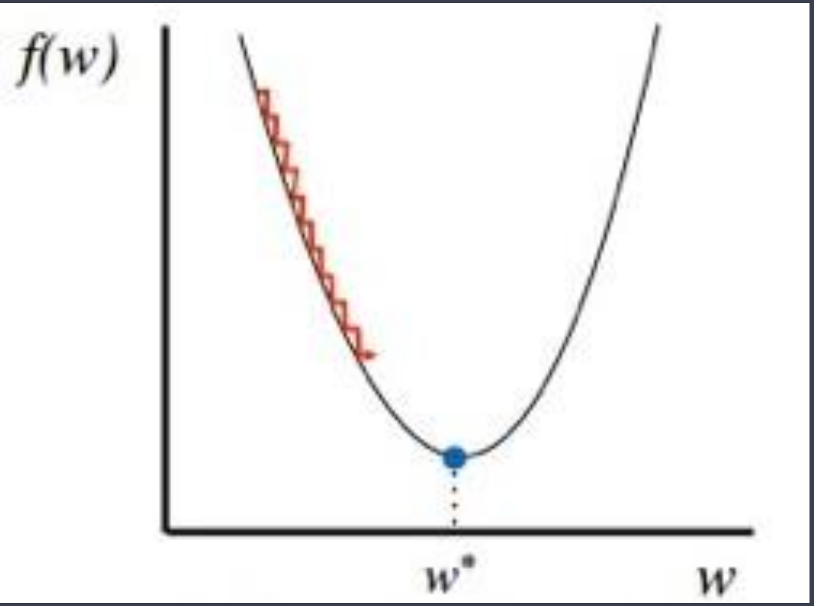


# Linear Model - Regression

## Learning rate가 큰 경우




## Learning rate가 작은 경우



# Linear Model - Regression 실습 : ex06 추가

경사하강법으로 학습하는
SGDRegressor 사용하기

# Linear Model - Regression

## Linear Model 장점

- **결과예측(추론) 속도가 빠르다.**
- **대용량 데이터에도 충분히 활용 가능하다.**
- **특성이 많은 데이터 세트라면 훌륭한 성능을 낼 수 있다.**

# Linear Model - Regression

## Linear Model 단점

- 특성이 적은 저차원 데이터에서는 다른 모델의 일반화 성능이 더 좋을 수 있다. ➡ 특성확장을 하기도 한다.
- LinearRegression Model은 복잡도를 제어할 방법이 없어 과대적합 되기 쉽다.

⬇

모델 정규화(Regularization)을 통해 과대적합을 제어한다.

# Linear Model - Regularization

## 모델 정규화

- 가중치(w)의 값을 조정하여 제약을 주는 것.

- L1 규제 : Lasso

  w의 모든 원소에 똑같은 힘으로 규제를 적용하는 방법. 특정 계수들은 0이 됨.
  특성선택(Feature Selection)이 자동으로 이루어진다.

- L2 규제 : Ridge

  w의 모든 원소에 골고루 규제를 적용하여 0에 가깝게 만든다.

# Linear Model - Regularization

정규화 : cost 함수

alpha hpyerparameter로 조정

L1 규제 : Lasso

$$J(w)_{LASSO} = \sum_{i=1}^{n}(y^{(i)} - y^{\wedge(i)})^2 + \lambda \sum_{j=1}^{m}|w_j|$$

L2 규제 : Ridge

$$J(w)_{Ridge} = \sum_{i=1}^{n}(y^{(i)} - y^{\wedge(i)})^2 + \lambda \sum_{j=1}^{m} w_j^{\,2}$$

# Linear Model - Regression 평가지표

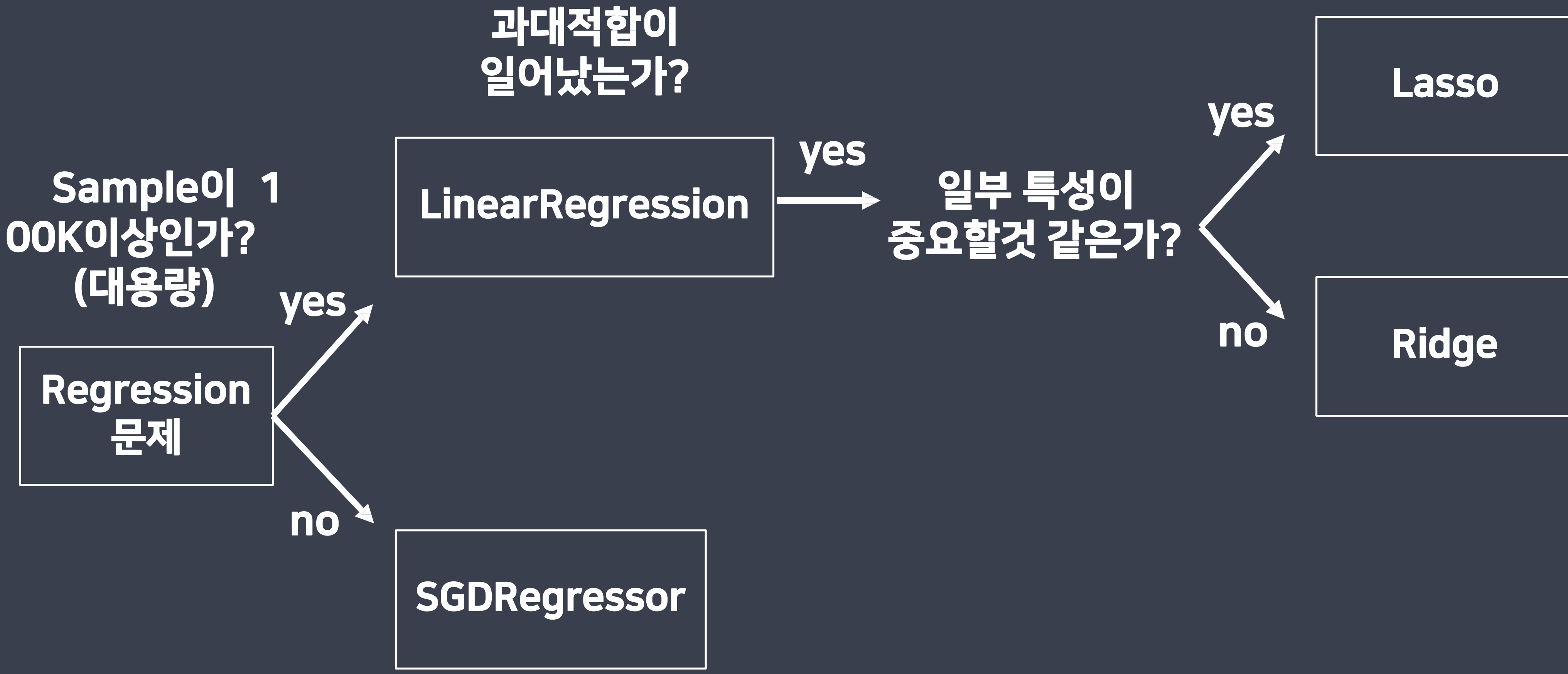
- 회귀에서는 평가지표로 MSE, RMSE 등을 사용할 수 있다.
- 그런데 MSE,RMSE 값은 성능 지표로 사용하기에는 아쉬운 점이 있다.
  - 예를 들어 키를 예측하였는데 MSE 값이 5.7cm이 나왔다고 하면 이것의 성능이 얼마나 우수한지 다른 경우와 비교하기 어렵다.
  - 몸무게를 예측하였는데 MSE값이 3.8kg이라면 얼마나 우수한 것인가?
- sklearn에서는 **Coefficient of Determination (R-squared)를 기본으로 사용한다**

$$R^2 \equiv 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$$

$$SS_{\text{res}} = \sum_i (y_i - f_i)^2 = \sum_i e_i^2$$

$$SS_{\text{tot}} = \sum_i (y_i - \bar{y})^2$$

# Linear Model - Regression cheat-sheet



# 보스턴 주택 값 예측 실습

Linear 모델을 이용해 보스턴 지역 주택 가격을 예측 해보자.