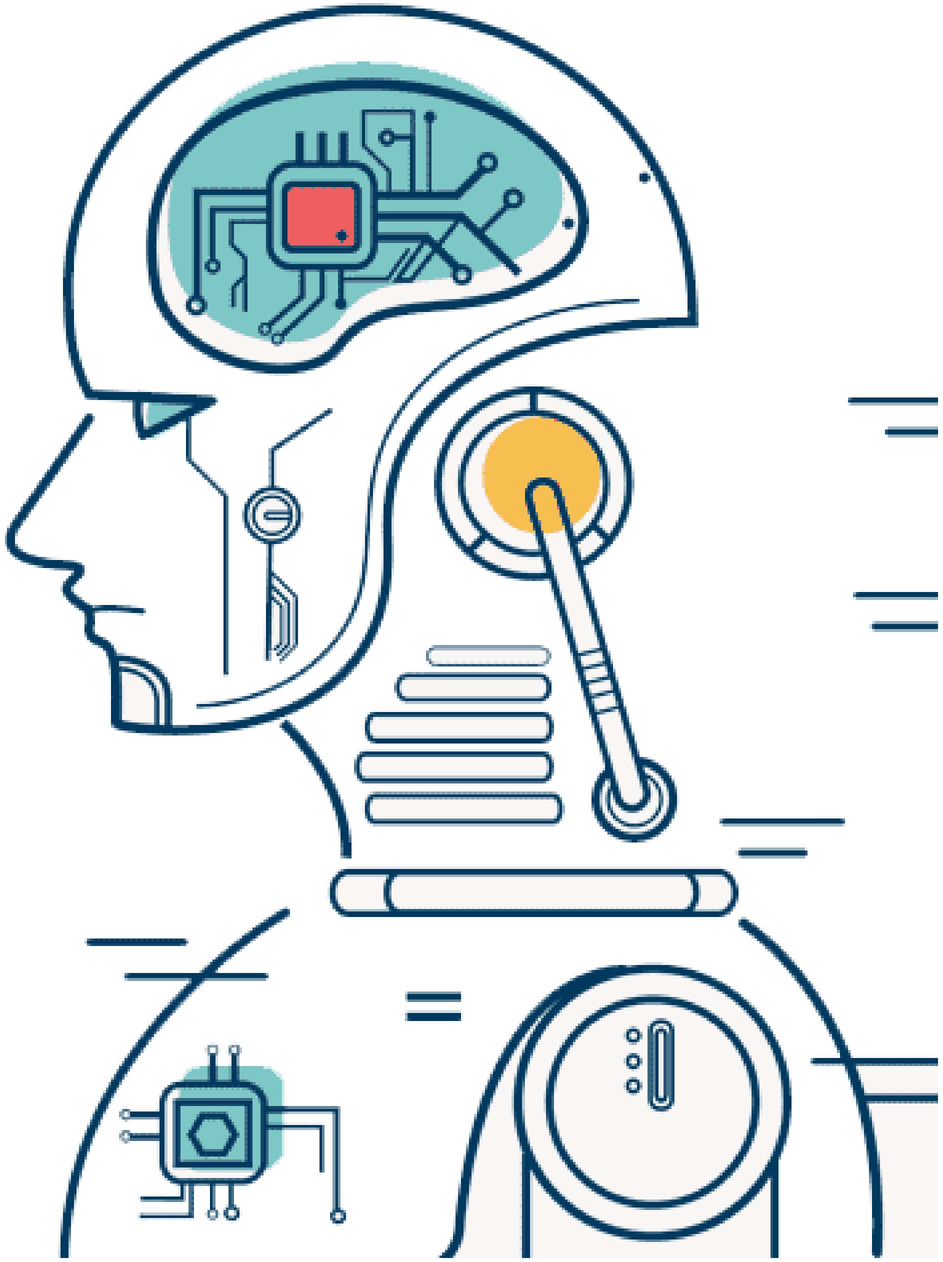
스마트인재개발원
Smart Human Resources Development

Machine Learning

Chapter_5 지도학습
(Linear Regression, Ridge(L1), Lasso(L2), 회귀평가지표)



김은영

학습목표

- 회귀 및 선형회귀의 개념과 필요성을 이해할 수 있다.
- 선형회귀 모델을 사용할 수 있다.
- 회귀 모델의 평가방법을 알 수 있다.
- 데이터 스케일링의 필요성을 이해하고 다양한 스케일링 방법을 알 수 있다.

Regression?

- 갈톤(Galton) : 부모와 자식 간의 키의 상관관계 분석 연구
- 사람의 키는 평균 키로 회귀하려는 경향을 가진다 ➔ 자연의 법칙 존재
- 데이터 값이 평균과 같은 일정한 값으로 돌아가려는 경향을 이용한 통계학 기법
- 여러 개의 독립변수와 한 개의 종속변수 간의 상관관계를 모델링 하는 기법
- 회귀 계수(Regression coefficients) : 독립변수의 값에 영향
- 회귀 유형 구분

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

머신러닝에서의 회귀(Regression) 분석?

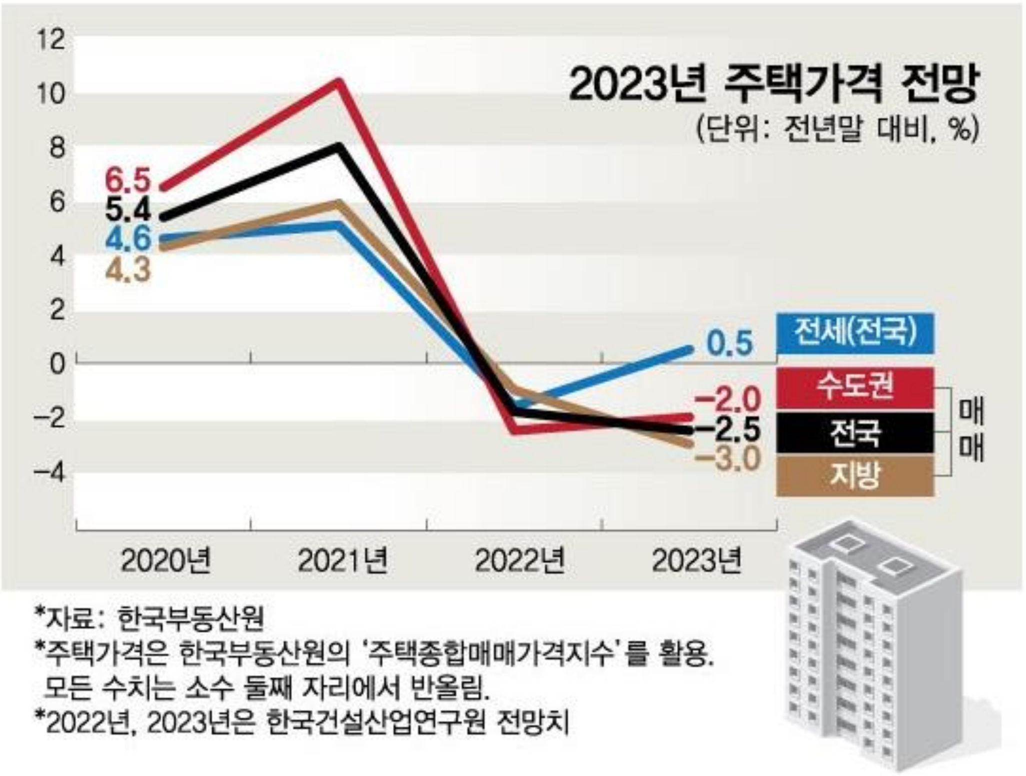
- 시간에 따라 변화하는 데이터나 영향, 가설적 실험, 인과 관계 모델링 등에서 많이 사용
- 종속변수(목표)와 하나 이상의 독립변수(예측변수) 간 미래 사건을 예측하는 방법

  ex) 1. 난폭운전과 운전자에 의한 교통사고 총 건수 사이의 상관관계 예측

  2. 비즈니스 상황에서 특정 금액을 광고에 사용했을 때 그것이 판매에 미치는 영향 사이의 관계 예측
- 수치적 가치를 추정

회귀(Regression)분석이 중요한 이유?

- 연속 숫자를 포함하는 머신러닝 문제 해결에 필수적, 딥 러닝 이론의 기초

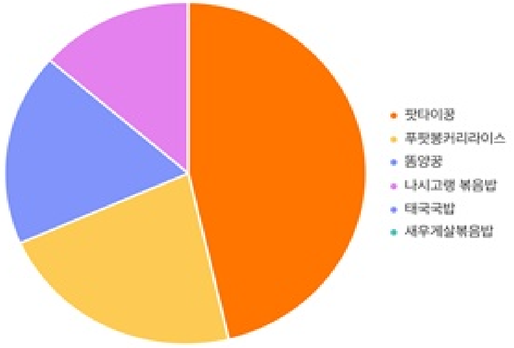


[머니투데이 22.11.02]

"내일은 얼마나 팔릴까?" 외식 수요예측 고민 AI로 해결한다

일별 대표메뉴 수요예측

날짜	구분	합계	똠양꿍	팟타이꿍	태국국밥	나시고랭 볶음밥	새우게살볶음밥	뿌팟봉커리라이스
어제	실제매출	78	9	32	0	12	2	23
오늘	예측매출	67	12	31	0	9	0	15
내일	예측매출	77	12	35	0	10	0	20



순위	메뉴명	어제 실제매출	오늘 예측매출	지난주 같은 요일 대비
	판매수량 합계	78	67	▼ -32%
1	팟타이꿍	32	31	▼ -12%
2	뿌팟봉커리라이스	23	15	▼ -12%
3	똠양꿍	9	12	▼ -12%
4	나시고랭 볶음밥	12	9	▼ -12%
5	태국국밥	0	0	▼ -12%
6	새우게살볶음밥	2	0	▼ -12%

[푸드경제신문 22.02.24]

Linear Model?

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

Linear Model

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

시험성적 데이터

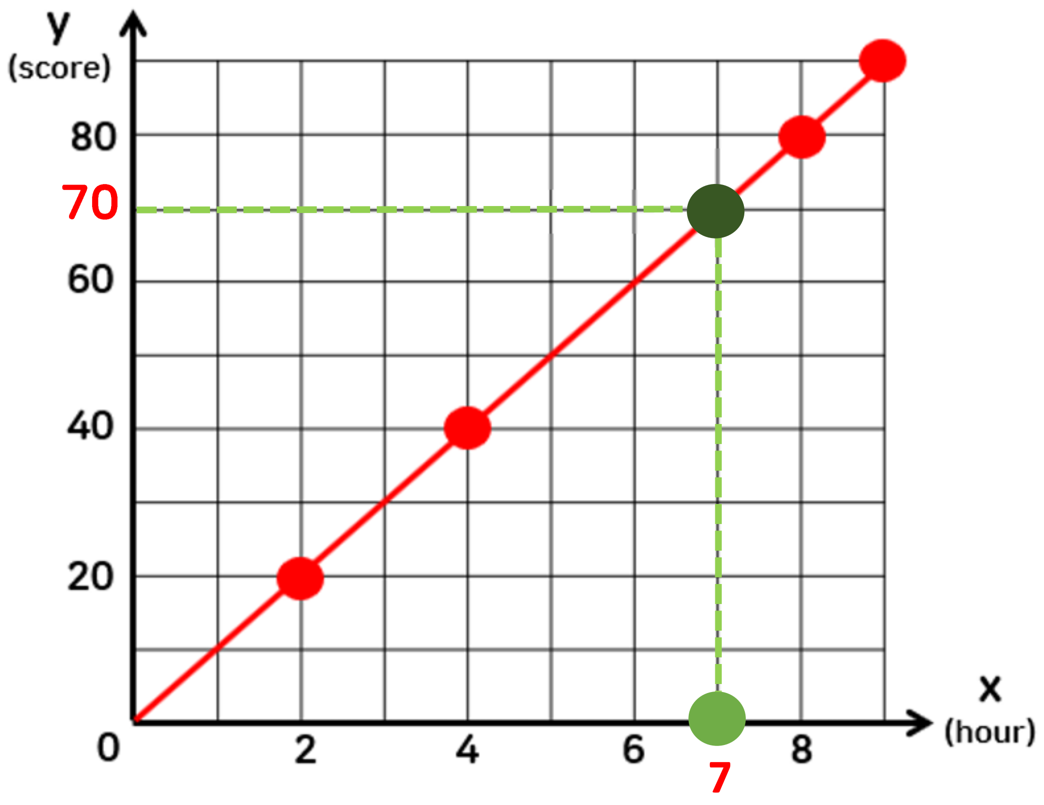
7시간 공부 할 경우 성적은
몇 점 일까?

Linear Model

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

$y = ax + b$

$y = 10x + 0$



Linear Model

종속(응답) 변수 기울기(가중치) → w

$$y = ax + b$$

독립(입력) 변수 절편(편향)

Linear Model - Regression

스마트인재개발원 Smart Human Resources Development

선형 회귀 함수

$$\hat{y} = w_1x_1 + w_2x_2 + w_3x_3 + \cdots + w_px_p + b$$

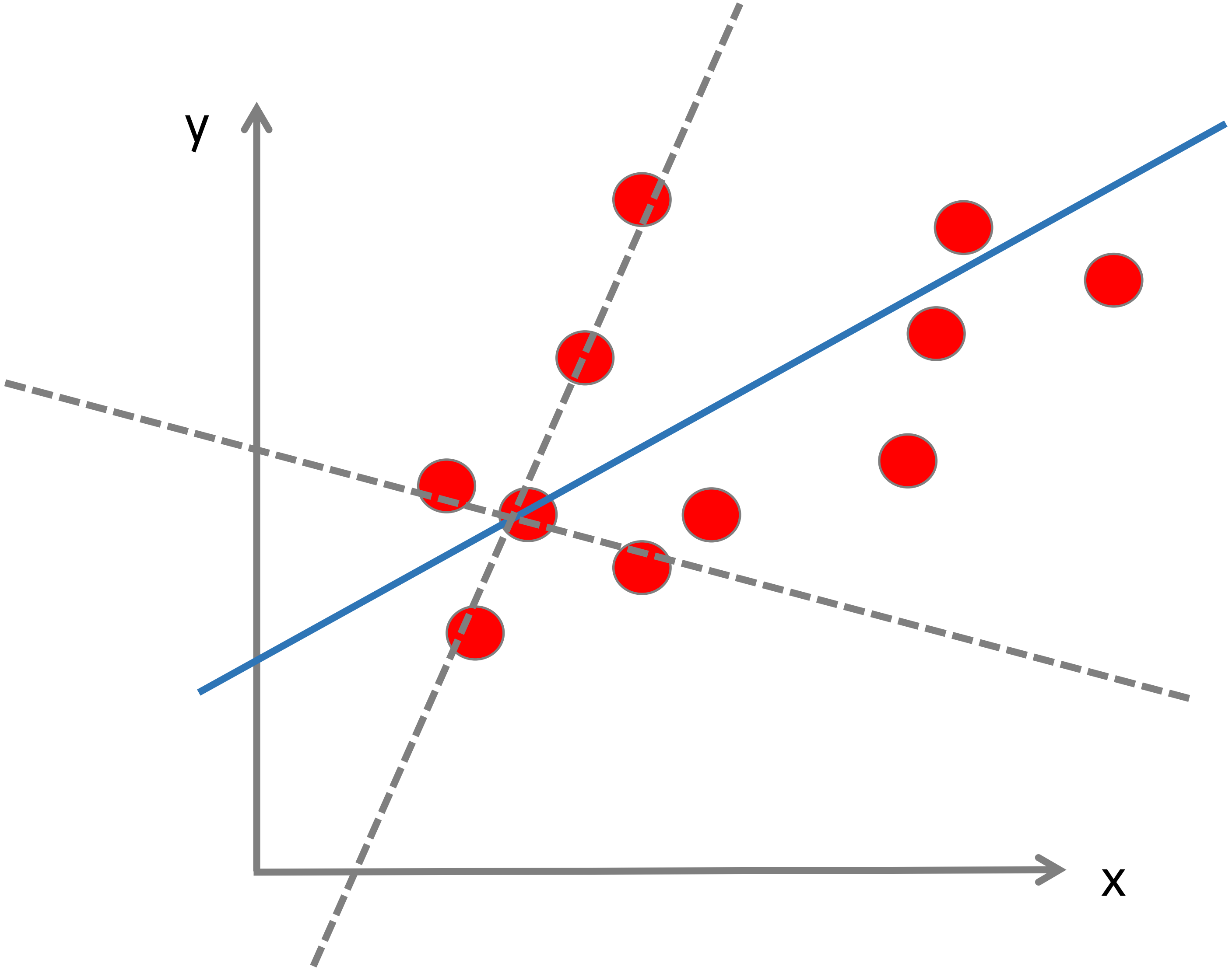
- w : 가중치(weight), 계수(coefficient)
- b : 절편(intercept), 편향(bias)

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

선형회귀 모델의 w, b 값 구하기 실습

회귀 모델의 성능은 어떻게 평가해야 할까?

회귀모델 평가지표 – MSE(Mean Squared Error)



평균제곱오차가 최소인
w와 b를 찾는다

회귀모델 평가지표 – MSE(Mean Squared Error)

Cost function

비용함수 : 수식을 검증

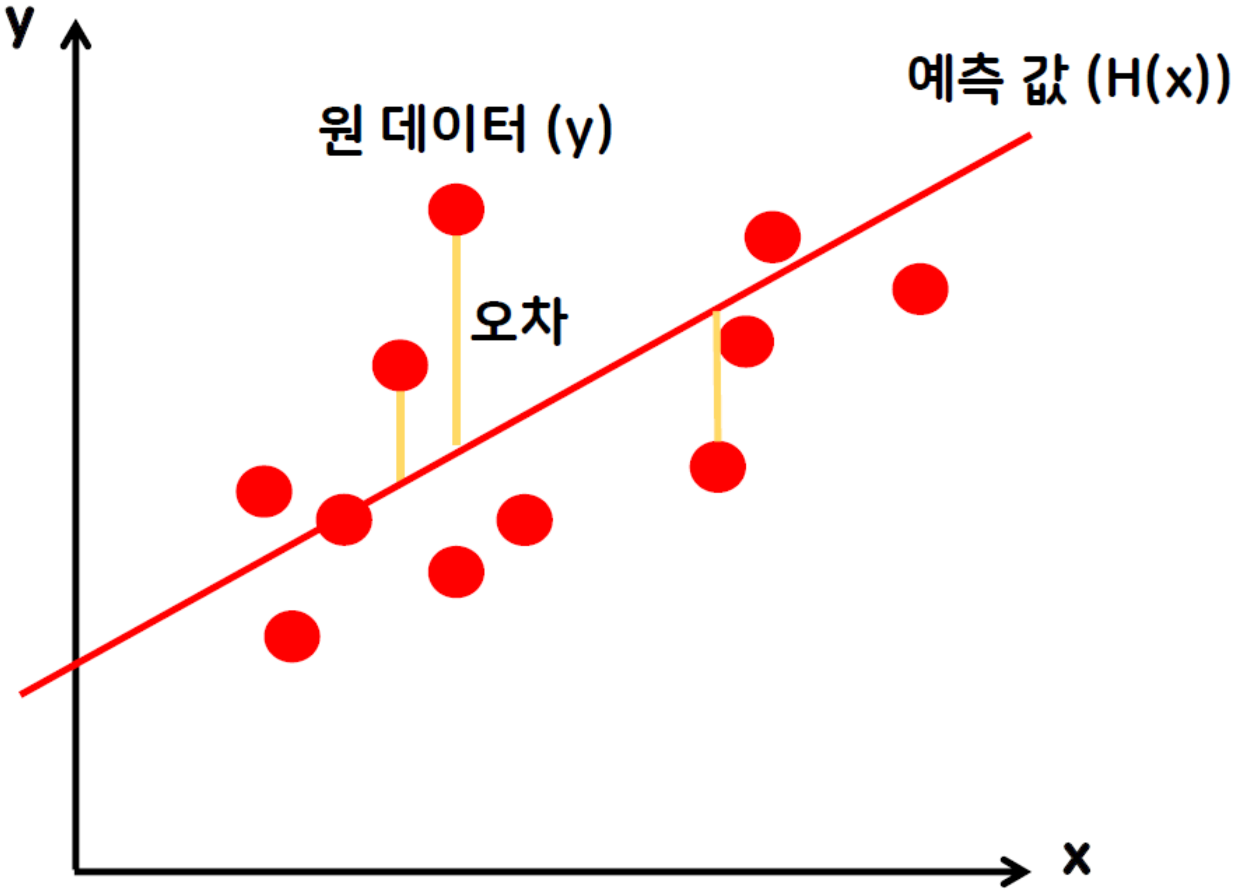
예측값

H(x) = w * x + b

예측값 정답값

오차 = H(x) - y

제곱 값 사용



회귀모델 평가지표 – MSE(Mean Squared Error)

평균제곱오차
(MSE)

$$MSE = \frac{1}{m}\sum_{i=1}^{m}\underbrace{\boxed{(\overset{\text{실제값}}{y_i} - \overset{\text{예측값}}{\hat{y}})}}_{\text{오차}}{}^{2}$$

$$H(x) = Wx + b \Rightarrow \hat{y}$$

평균제곱근오차
(Root MSE)

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

회귀모델 평가지표 – MSE(Mean Squared Error)

- 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 – 수학 공식을 이용한 해석적 방법(OLS)

스마트인재개발원
Smart Human Resources Development

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

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

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

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

OLS ➔ LinearRegression 클래스로 구현

Linear Model – 경사하강법(Gradient Descent Algorithm)

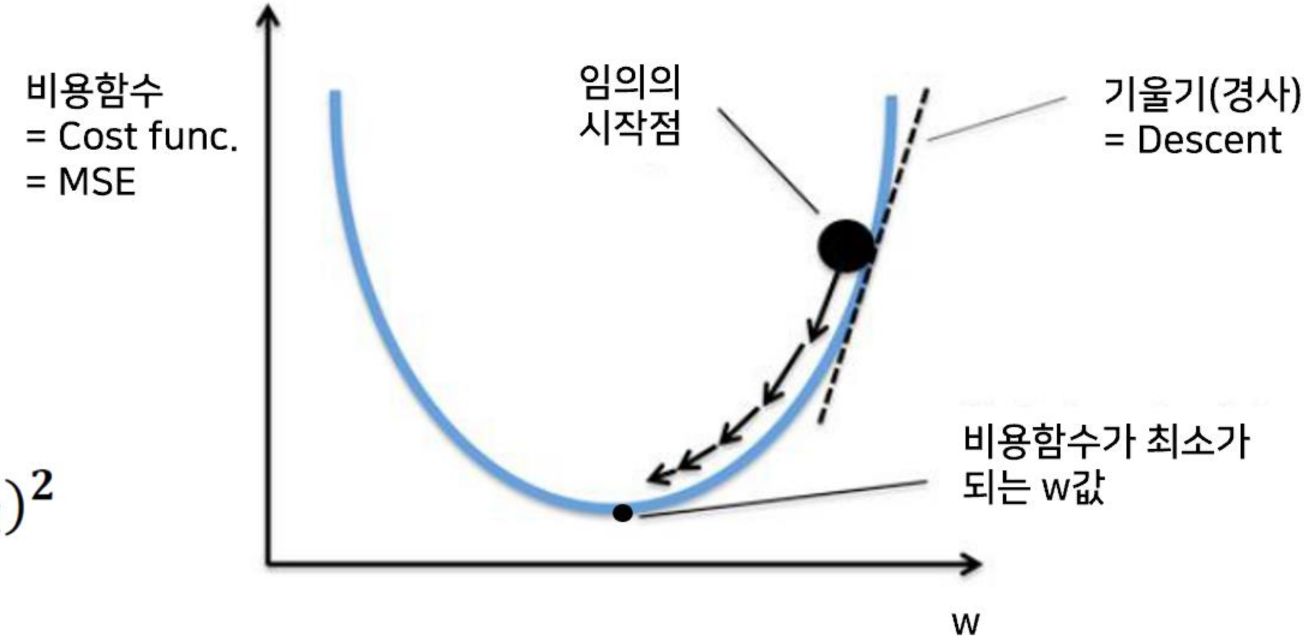
스마트인재개발원
Smart Human Resources Development

- 평균 제곱 오차(MSE)가 최소가 되게 하는 최적의 w, b값을 찾는 방법론
- 기계가 스스로 학습한다는 머신, 딥러닝의 개념이 있게 한 핵심 알고리즘

Linear Model – 경사하강법(Gradient Descent Algorithm)

선형 회귀의 비용함수
= 평균제곱오차(MSE)

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



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

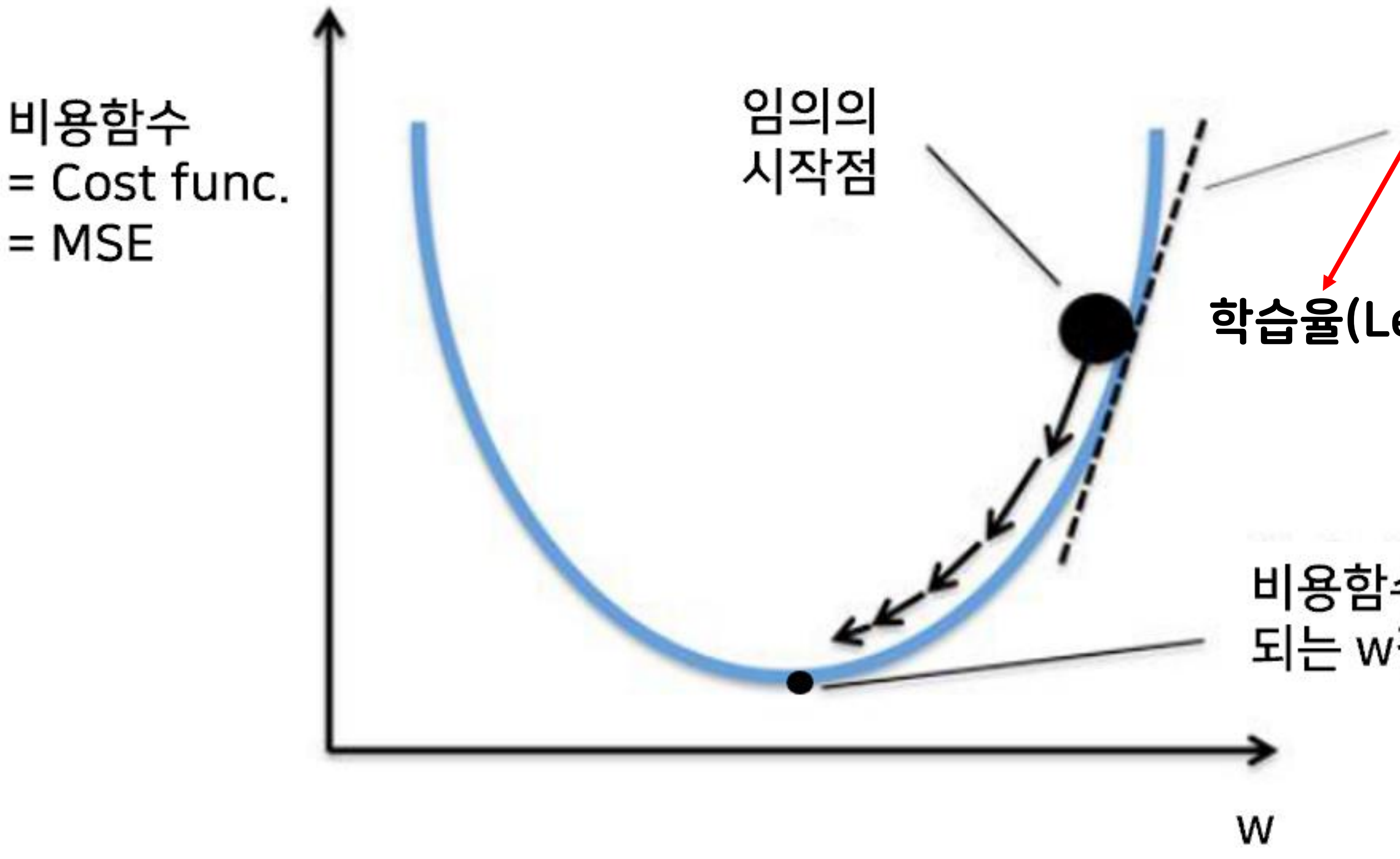
Linear Model – 경사하강법(Gradient Descent Algorithm)

스마트인재개발원
Smart Human Resources Development

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

학습율(Learning rate)

기울기(경사)
= Descent



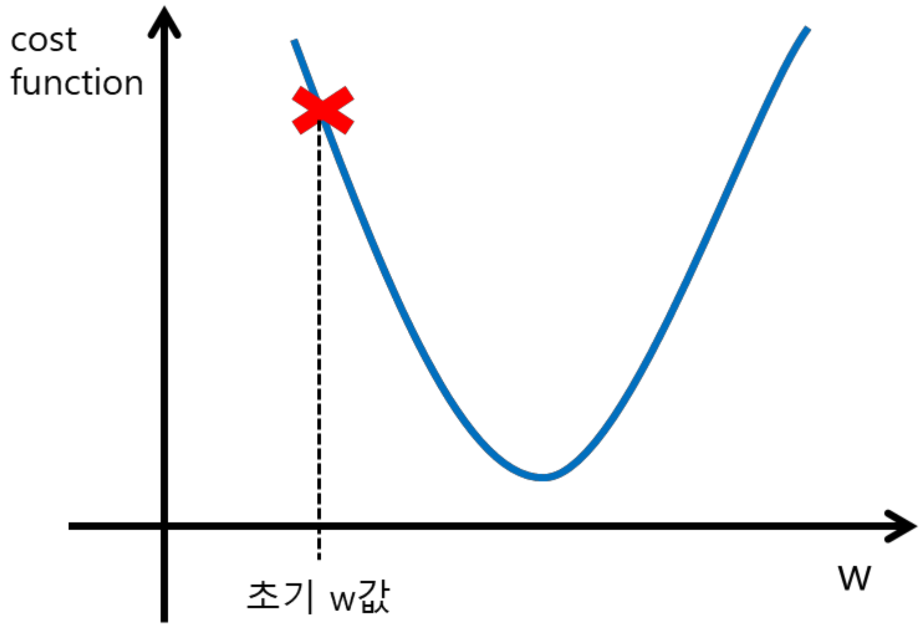
Linear Model – 경사하강법(Gradient Descent Algorithm)

스마트인재개발원
Smart Human Resources Development

1. 임의의 w값을 하나 선정

   - 운이 아주 좋다면 최적의 값이겠지만 그렇지 않을 확률이 훨씬 더 큼

   - 대부분 최적의 w값과는 거리가 먼 값으로 설정



Linear Model – 경사하강법(Gradient Descent Algorithm)

2. 최적의 w값을 찾아가기 위해 시작점에서 손실 곡선의 기울기 계산

   → 비용함수를 w에 대해서 편미분

3. 파라미터를 곱한 것을 초기 설정된 w값에서 빼 줌

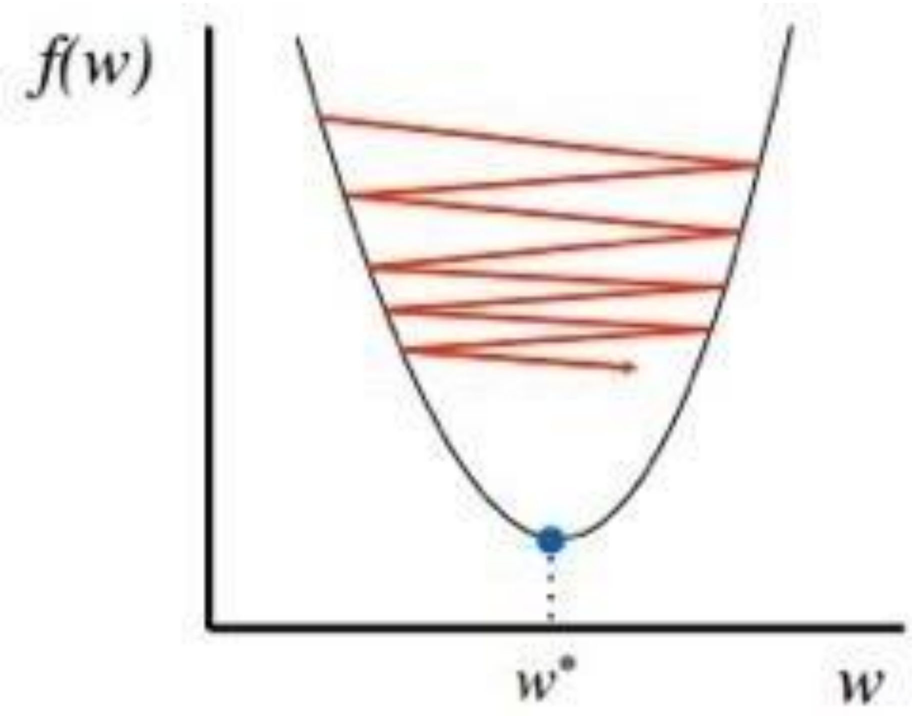
- 학습률(learning rate) : 기울기의 보폭
- 학습률이 너무 작으면 : 최적의 w를 찾는데 오래 걸림
- 학습률이 너무 크면 : 값이 건너뛰어 버리고 발산

$$w' = w - \alpha \frac{\partial e}{\partial w} \qquad b' = b - \alpha \frac{\partial e}{\partial b}$$

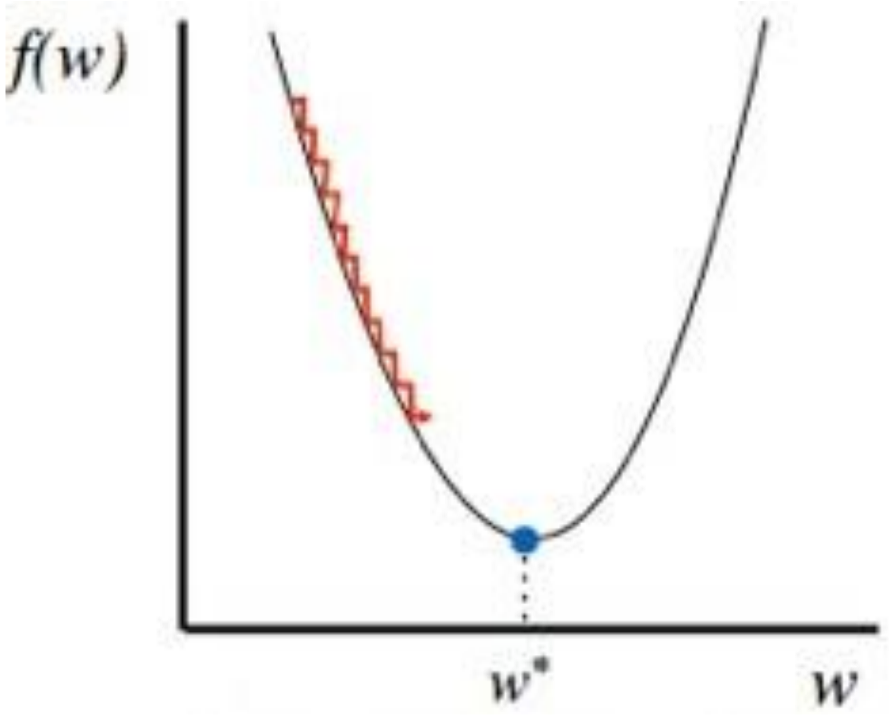
Linear Model – 경사하강법(Gradient Descent Algorithm)

스마트인재개발원
Smart Human Resources Development

Learning rate가 큰 경우

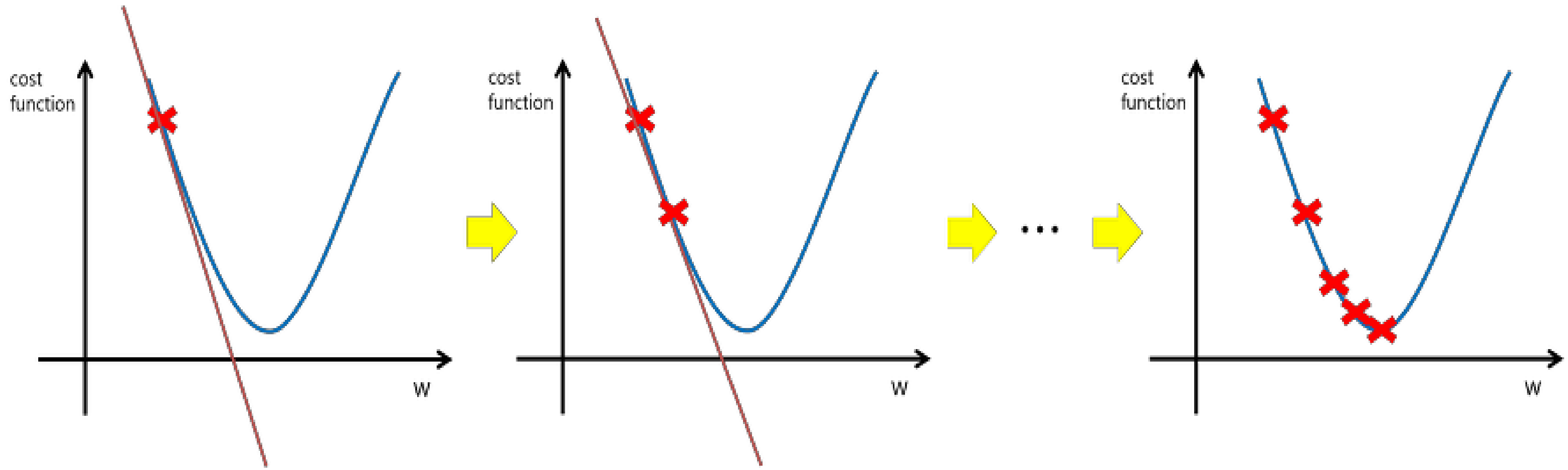


Learning rate가 작은 경우

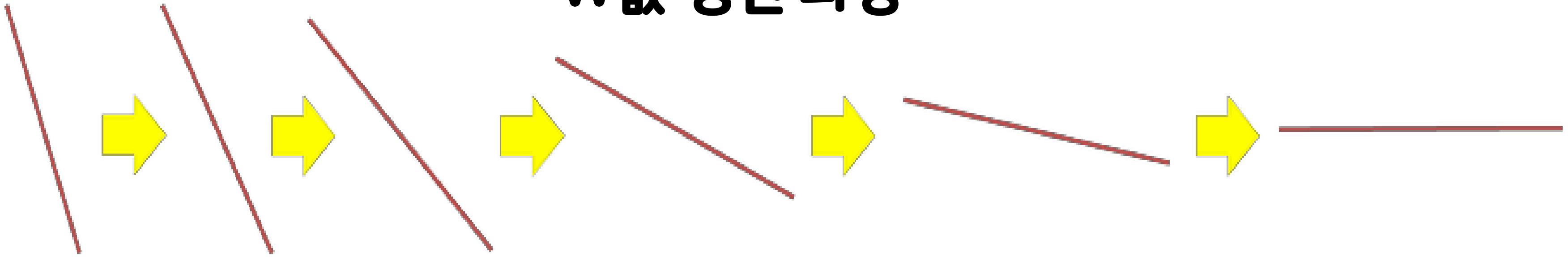


Linear Model – 경사하강법(Gradient Descent Algorithm)

스마트인재개발원
Smart Human Resources Development

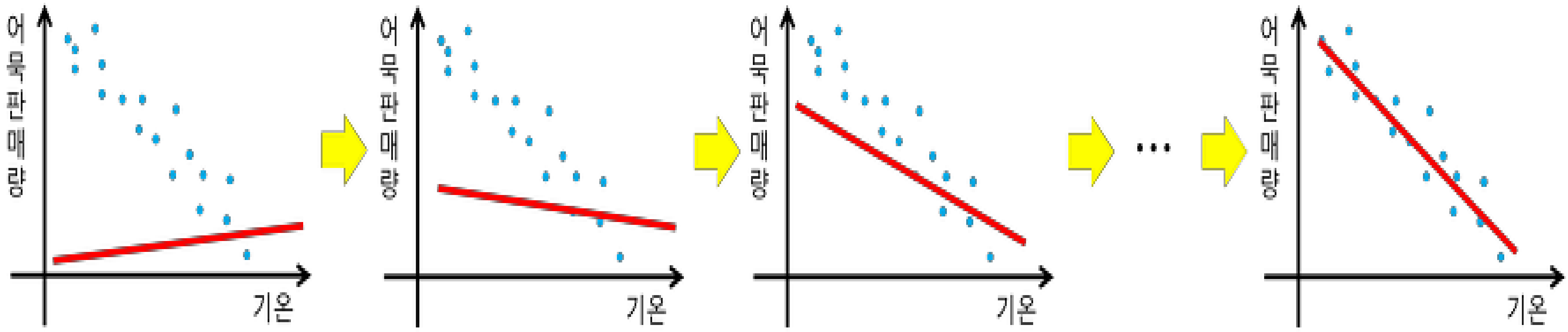


W값 갱신 과정



경사가 점차 감소되는 현상을 이용하므로 경사감소법!

Linear Model – 경사하강법(Gradient Descent Algorithm)



초기 w, b값은 데이터를 잘 반영하는 일차함수식을 만들지 못했지만,

점차적으로 데이터를 잘 반영해내는 값들로 갱신

LinearRegression 사용하기

경사하강법으로 학습하는

SGDRegressor 사용하기

주요 매개변수(Hyperparameter)

scikit-learn의 경우

```
SGDRegressor(max_iter, eta0)
```

- 가중치 업데이트 횟수 : max_iter
- 학습률 : eta0

Linear Model 장점

- 결과예측(추론) 속도가 빠름
- 대용량 데이터에도 충분히 활용 가능
- 특성이 많은 데이터 세트라면 훌륭한 성능을 낼 수 있음

Linear Model - 단점

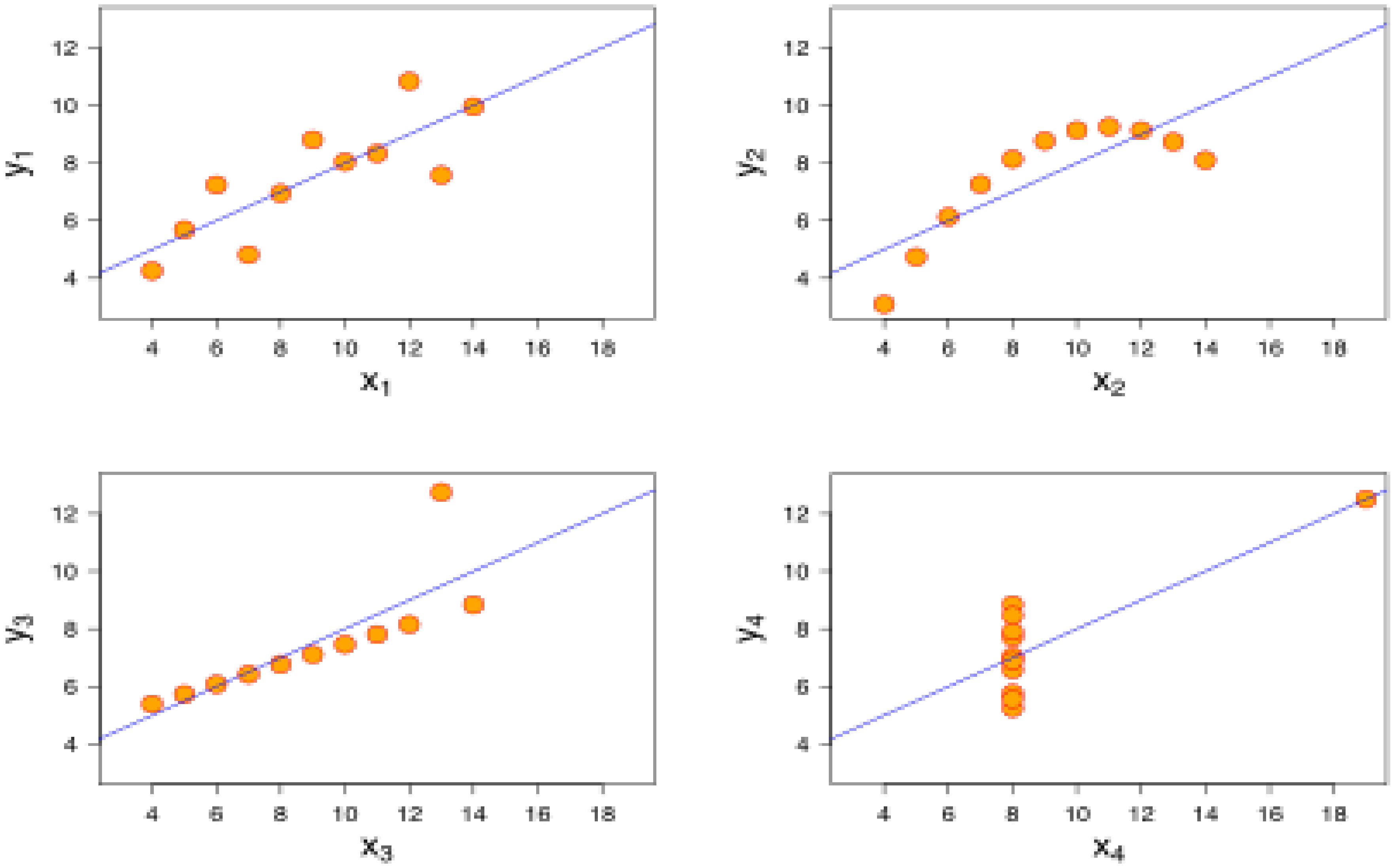
스마트인재개발원
Smart Human Resources Development

- 특성이 적은 저차원 데이터에서는 다른 모델의 일반화 성능이 더 좋을 수 있음

  ➔ 특성 확장 필요

- LinearRegression 모델은 복잡도를 제어할 방법이 없어 과대적합 되기 쉬움 ➔ 모델 정규화(Regularization)-규제를 통해 과대적합 제어

Linear Model - 단점



Linear Model – 보스턴 주택 가격 예측 실습

Linear 모델에 보스턴 주택 가격 데이터를

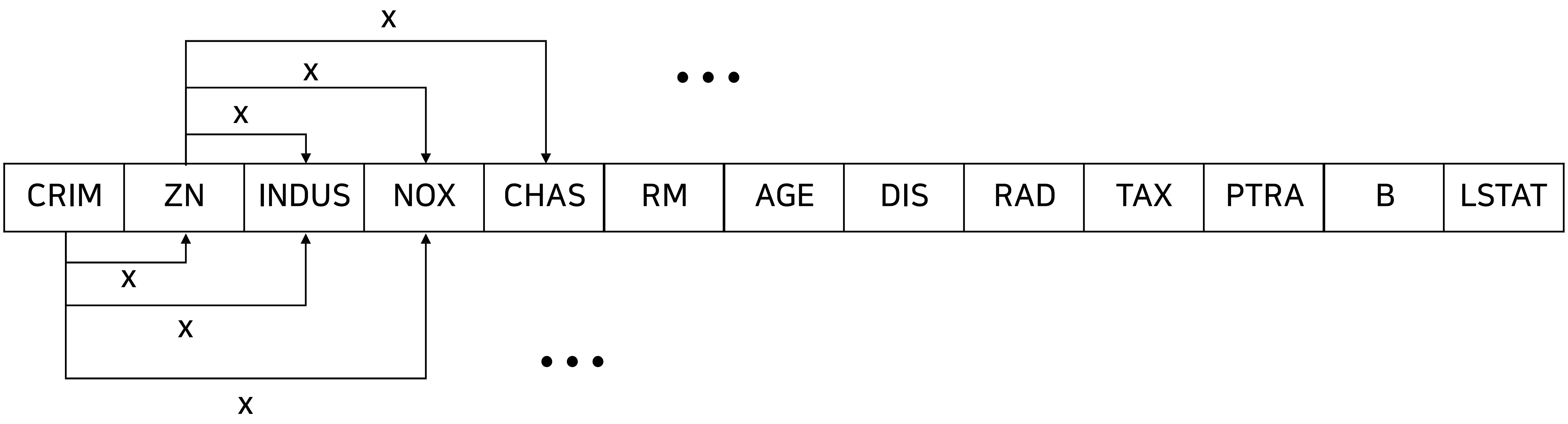
이용하여 주택 가격을 예측해 보자

회귀모델 평가지표 – R2 Score

$$R^2 = 1 - \frac{\text{오차의 제곱의 합}}{\text{편차의 제곱의 합}}$$

- 회귀 함수(직선)가 평균에 비해 얼마나 그 데이터를 잘 설명할 수 있는가에 대한 점수
- 편차 = 예측값과 평균과의 거리
- 오차 = 예측값과 회귀 직선과의 거리
- 0 ~ 1사이의 값 → 예측이 심하게 어긋날 경우 '-'값이 나올 수 있음
- '-'값은 회귀 직선이 평균보다 더 데이터를 잘 설명하지 못한다는 의미

보스턴 주택 가격 데이터 셋



Linear Model – 보스턴 주택 가격 예측 실습

스마트인재개발원
Smart Human Resources Development

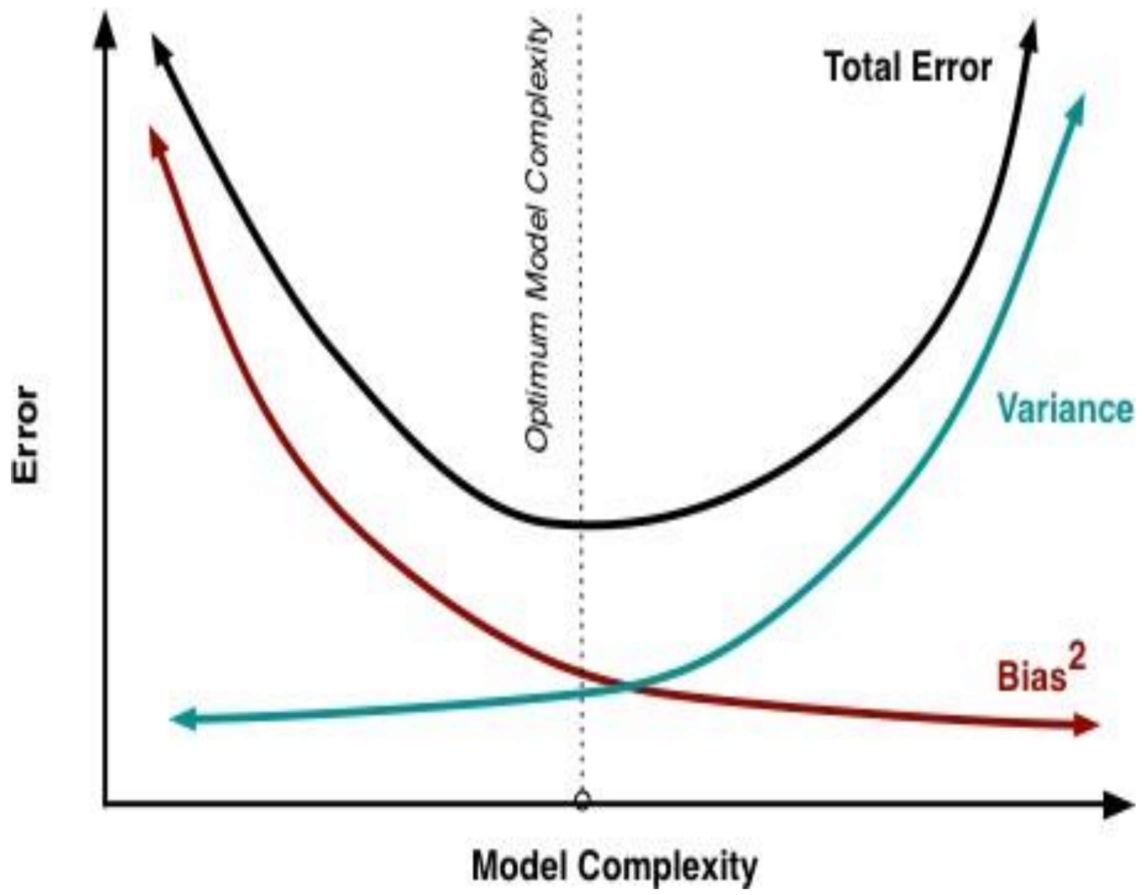
특성을 확장한 보스턴 주택 가격을 적용하여
학습해보자

Linear Model – 모델정규화(Regularization)

스마트인재개발원
Smart Human Resources Development

과대적합(overfitting) 문제 해결

- 데이터의 복잡도 줄이기
- 정규화를 통한 분산 감소



Linear Model – 모델정규화(Regularization) LinearRegression

- w가 크다면 입력 x가 조금만 달라져도 y가 크게 변함
  ➔ w에 규제를 주어 영향을 줄이도록 하는 것

- L1 규제(Lasso)
  - w의 모든 원소에 똑같은 힘으로 규제를 적용하는 방법
  - 특정 계수들은 0이 됨
  - 특성선택(Feature Selection)이 자동으로 이루어짐

- L2 규제(Ridge)
  - W의 모든 원소에 골고루 규제를 적용하여 0에 가깝게 만듦

Linear Model - 모델정규화(Regularization)

정규화 : cost 함수

규제의 강도

L1 규제(Lasso)

$$J(w)_{LASSO} = \sum_{i=1}^{n} (y^{(i)} - y^{\wedge(i)})^2 + \lambda \underbrace{\sum_{j=1}^{m} |w_j|}_{\text{L1 규제}}$$

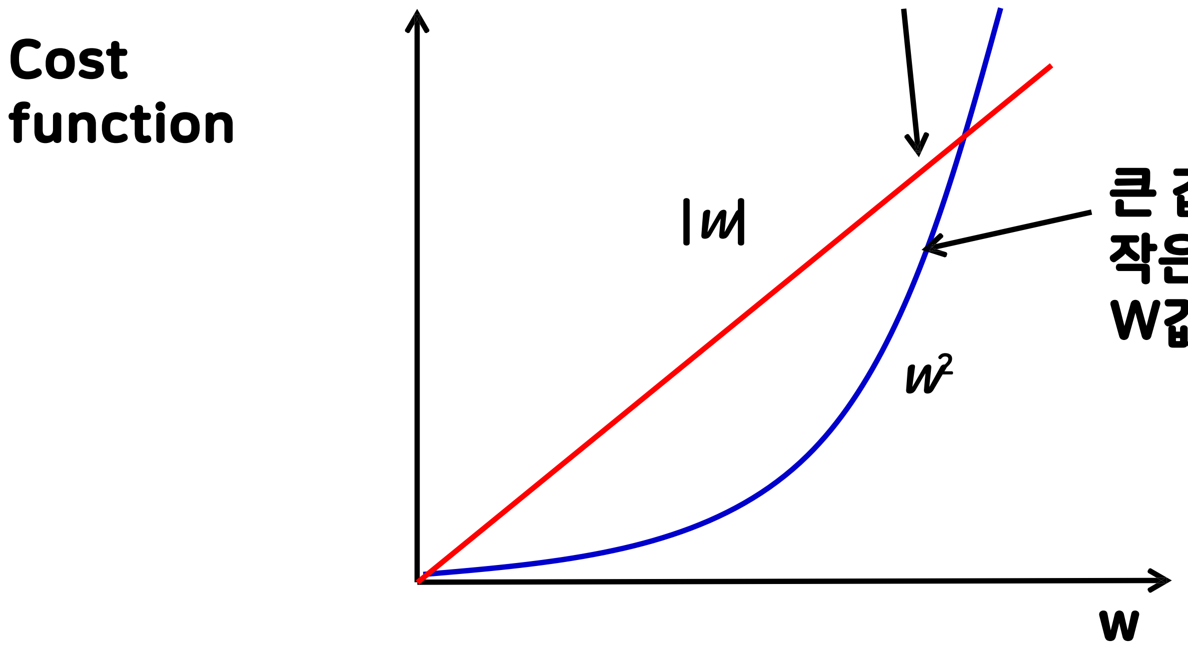
L2 규제(Ridge)

$$J(w)_{Ridge} = \sum_{i=1}^{n} (y^{(i)} - y^{\wedge(i)})^2 + \lambda \underbrace{\sum_{j=1}^{m} {w_j}^2}_{\text{L2 규제}}$$

Linear Model – 모델정규화(Regularization)

정규화 : cost 함수

일정 크기로 W값을 조정

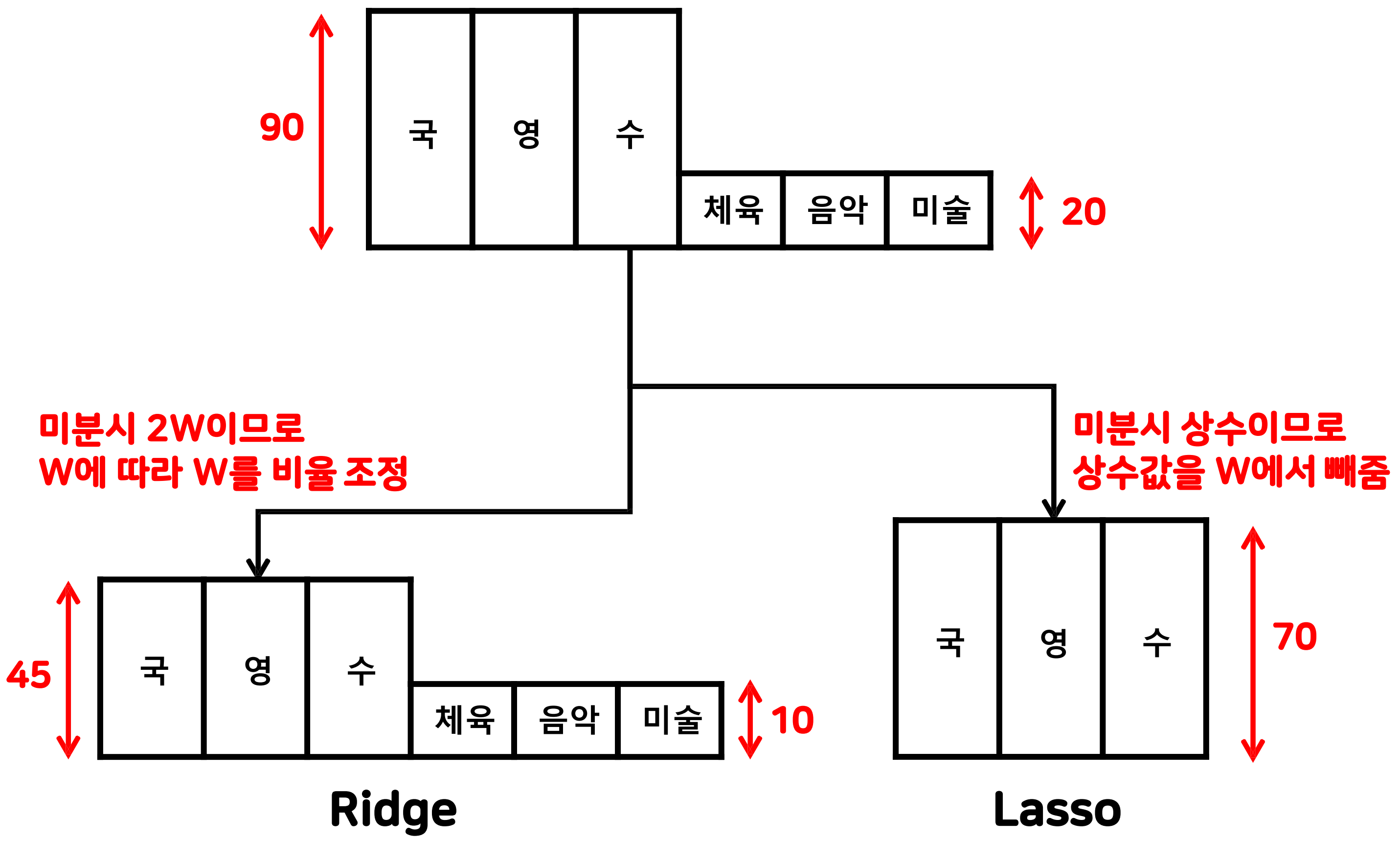


큰 값은 큰 비율로
작은 값은 작은 비율로
W값을 조정

Linear Model – 모델정규화(Regularization)

스마트인재개발원
Smart Human Resources Development

정규화 : cost 함수



Linear Model – 모델정규화(Regularization)

스마트인재개발원
Smart Human Resources Development

구분	릿지회귀	라쏘회귀
제약식	L_2 norm	L_1 norm
변수선택	불가능	가능
solution	closed form	명시해 없음
장점	변수간 상관관계가 높으면 좋은 성능	변수간 상관관계가 높으면 성능↓
특징	크기가 큰 변수를 우선적으로 줄임	비중요 변수를 우선적으로 줄임

주요 매개변수(Hyperparameter)

scikit-learn의 경우

Ridge(alpha)

Lasso(alpha)

- 규제의 강도 : alpha

Linear Model – 모델정규화(Regularization)

스마트인재개발원
Smart Human Resources Development

총 104개의 특성을 라쏘 회귀 모델을 만들기 위해 사용

↓

α=1로 설정했더니 104개의 가중치 중에서 50개가 0이 되면서 특성은 단 54개만 사용

↓

훈련셋에서의 점수와 테스트셋에서의 점수를 보니 과소적합

↓

복잡도를 높이기 위해서 α=0.0001로 설정했더니 가중치 중에서 0개가 0이 되면서 104개의 특성이 사용

↓

훈련셋과 테스트셋에서의 점수를 보니 훈련셋과 테스트셋이 많이 좋아짐

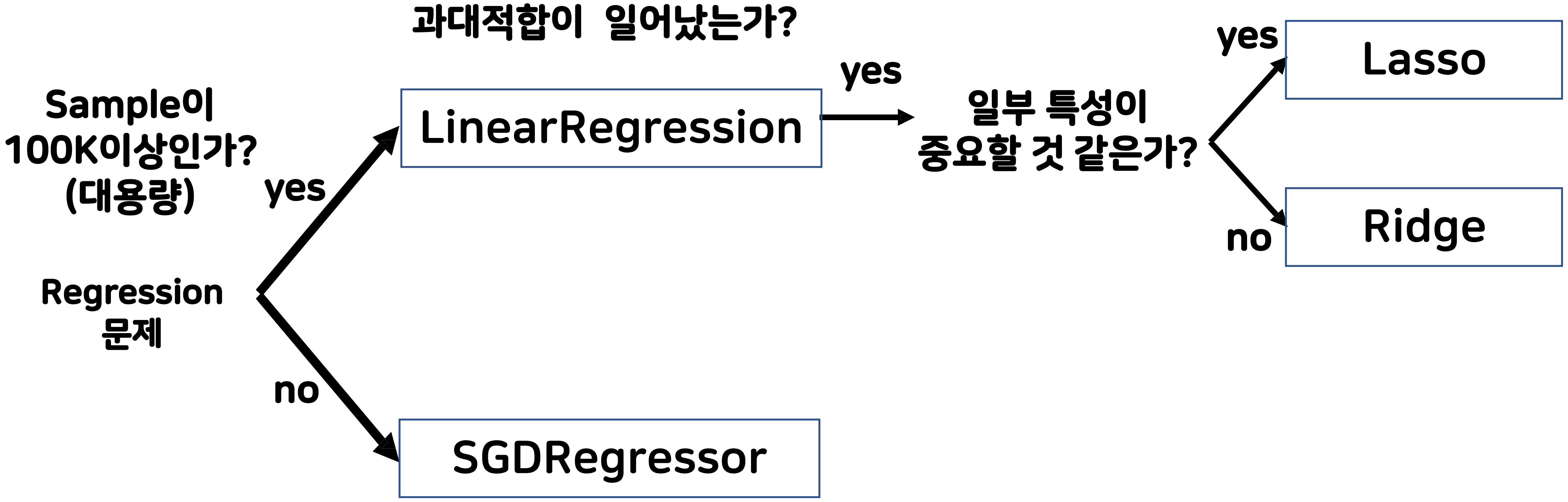
↓

가장 좋은 모델인지 확인하기 위해 다시 복잡도를 α=0.1로 설정했더니 105개의 가중치 중에서 25개가 0이 되면서 79개의 특성이 사용 → 훈련점수와 테스트점수가 조금씩 떨어짐

Ridge와 Lasso 모델을 이용하여

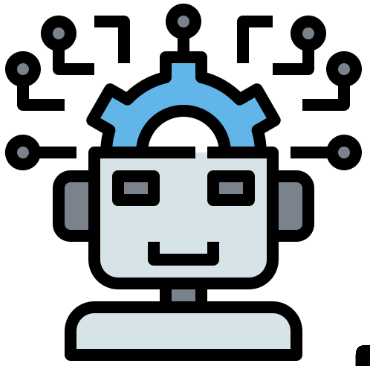
확장 보스턴 주택 가격의 과대적합 문제를 해결해보자

Linear Model - Regression



이전 실습의

- Lasso 모델의 최적의 alpha 값을 구해보자.
- 파라미터 튜닝 과정을 그래프로 그려보자



데이터 스케일링
(Data scaling)

데이터 스케일링(Data scaling) 정의

- 특성(Feature)들의 범위(range)를 정규화 해주는 작업
- 특성마다 다른 범위를 가지는 경우 머신러닝 모델들이 제대로 학습되지 않을 가능성이 있음.
  (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 거리값을 기반으로 학습할 때 영향을 많이 줌

데이터 스케일링(Data scaling) 장단점

장점

- 특성들을 비교 분석하기 쉽게 만들어 줌
- Linear Model, Neural network Model 등에서 학습의 안정성과 속도를 개선

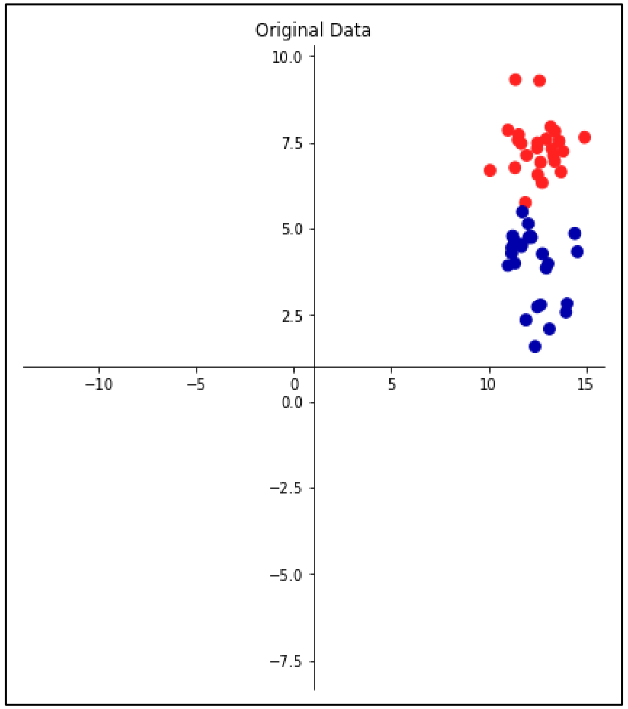
단점

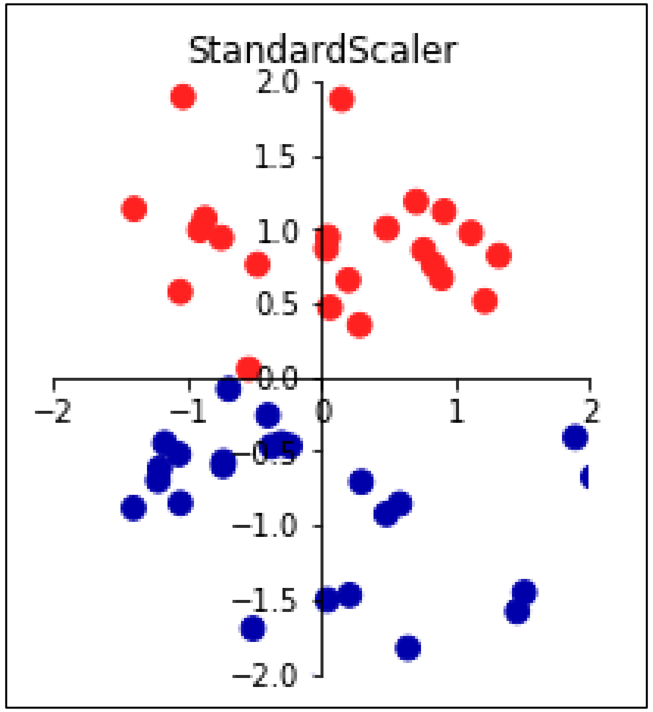
- 특성에 따라 원래 범위를 유지하는 게 좋을 경우 → scaling을 하지 않음

데이터 스케일링(Data scaling) 종류

StandardScaler

- 변수의 평균, 분산을 이용해 정규분포 형태로 변환 (평균 0, 분산 1)
- 데이터가 정규분포인 경우에 사용

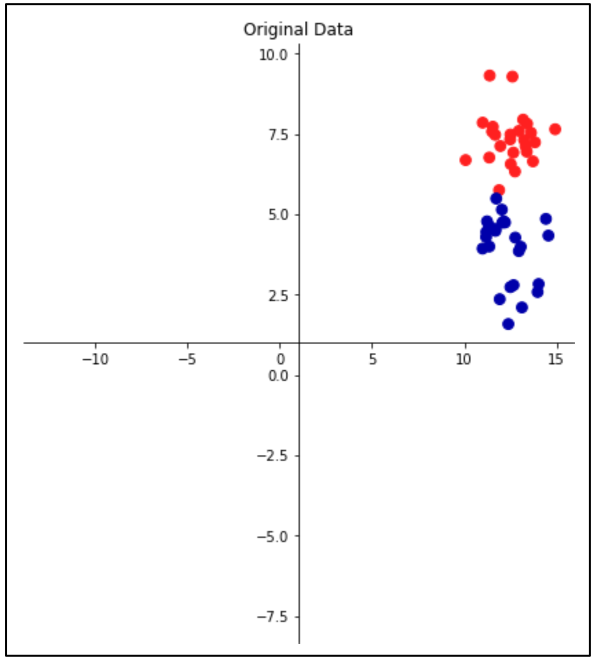


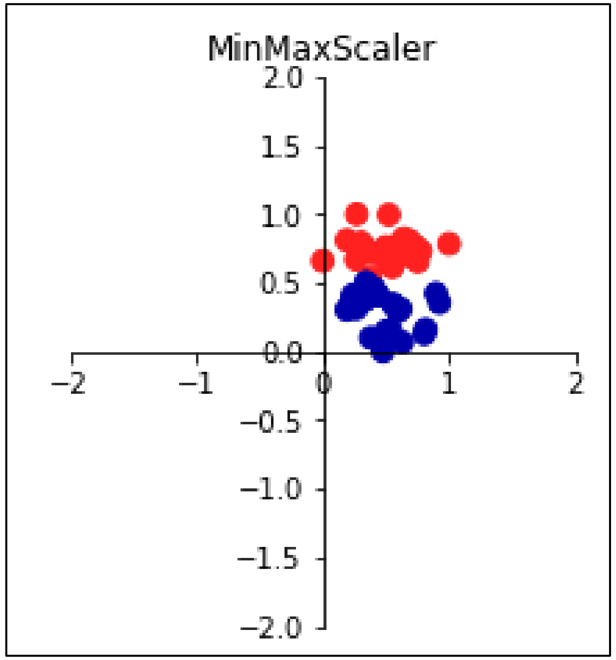


데이터 스케일링(Data scaling) 종류

MinMaxScaler

- 변수의 최대값 1로, 최소값을 0으로 하여 변환 (0 ~ 1 사이 값으로 변환)
- 데이터가 비정규분포인 경우에 사용
- 이상치(Outlier)에 크게 영향 → 이상치가 있는 경우 사용 못함

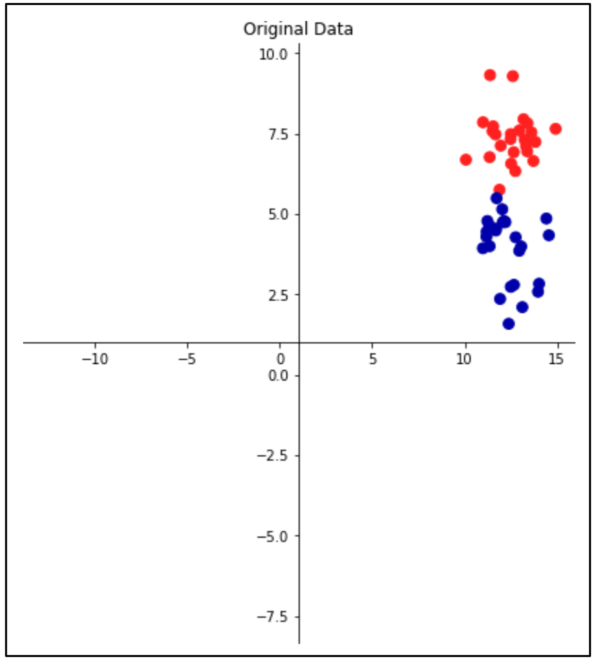


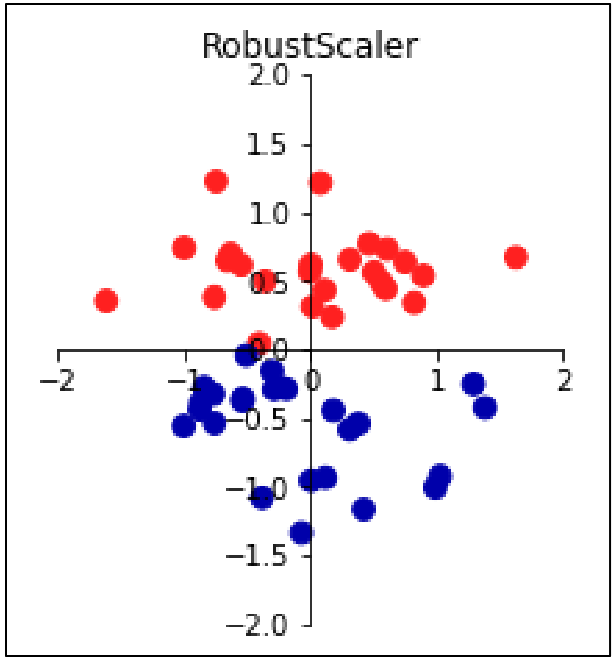


데이터 스케일링(Data scaling) 종류

RobustScaler

- 사분위수를 활용 – 변수의 25% 지점을 0으로 75%지점을 1로 하여 변환
- 이상치(Outlier)가 있는 경우 사용

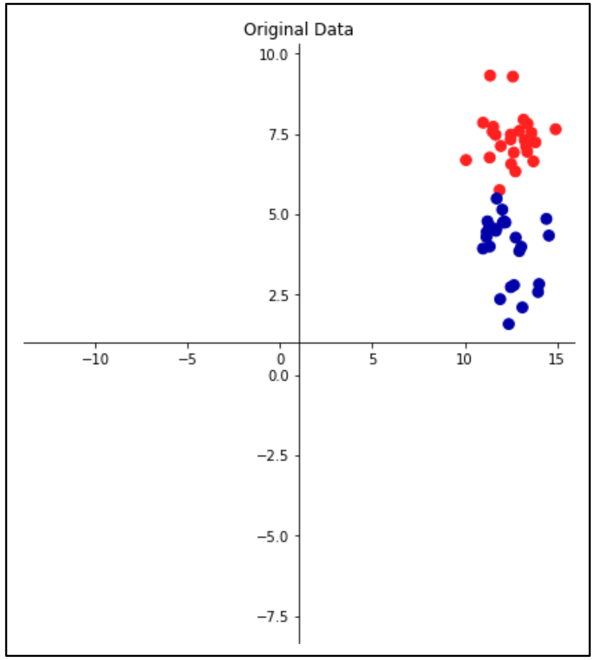


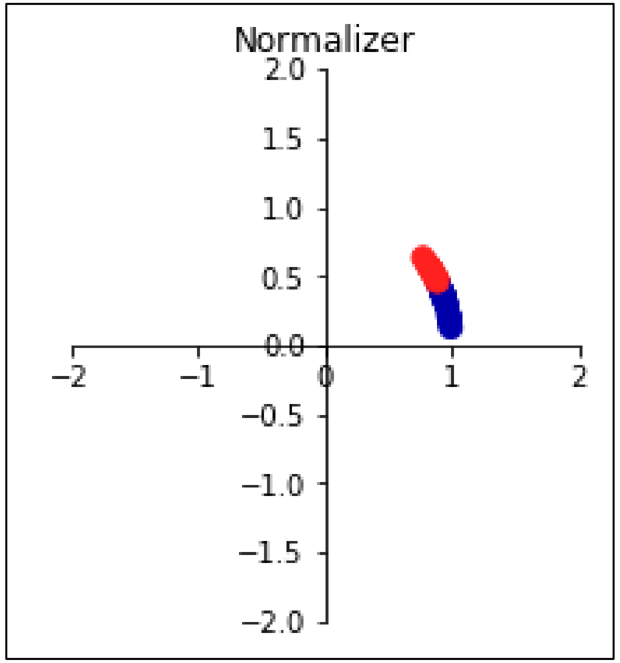


데이터 스케일링(Data scaling) 종류

Normalizer

- 특성 벡터의 유클리디안 길이가 1이 되도록 조정(지름이 1인 원에 투영)
- 특성 벡터의 길이는 상관 없고 데이터의 방향(각도)만 중요할 때 사용.



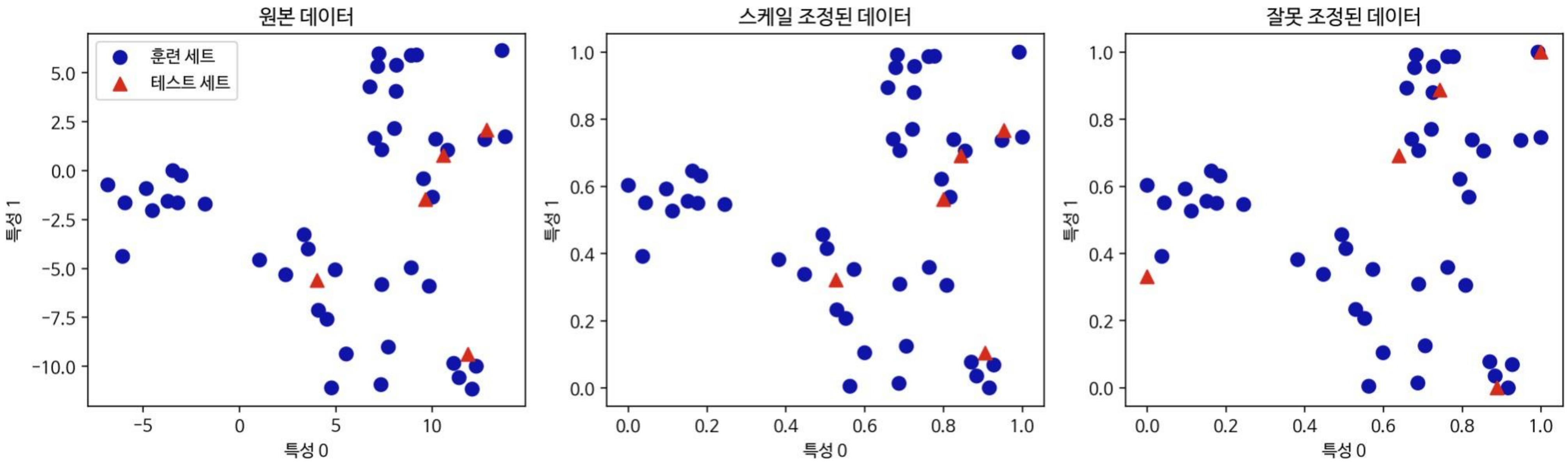


데이터 스케일링(Data scaling) 사용 시 주의점

- 훈련세트와 테스트세트에 같은 변환을 적용
- ex) 훈련세트의 평균과 분산을 이용해 훈련세트를 변환

  테스트세트의 평균과 분산을 이용해 테스트세트를 각각 변환하면?

  ➔ 잘못된 결과가 나올 수 있음 (왜? 적용된 값의 범위가 다를 수 있기때문에)



데이터 스케일링(Data scaling) 실습

유방암 데이터를
- KNN 모델로 학습하고
- scaler를 적용하여 결과를 확인해 보자