2주차 : 회귀(단순,다중선형회귀,규제선형모델)

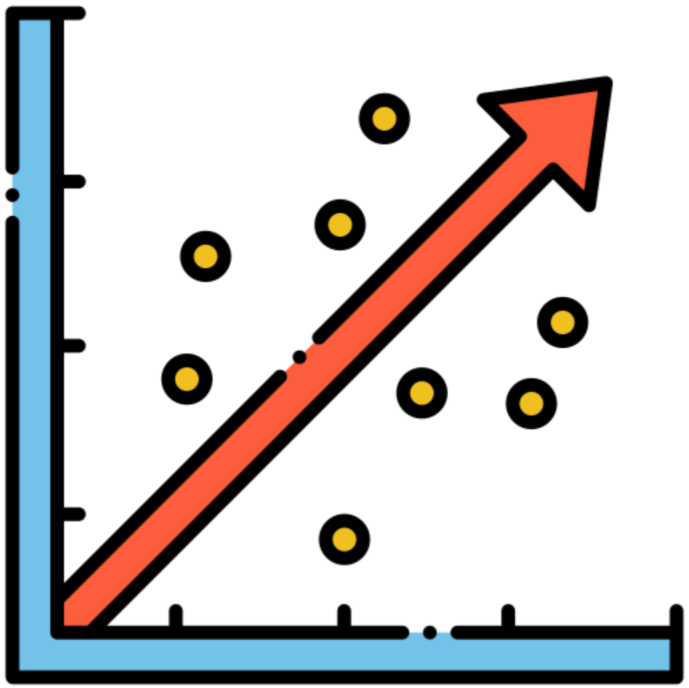
쿠글 8기 노동환, 이승민

Kuggle

Contents

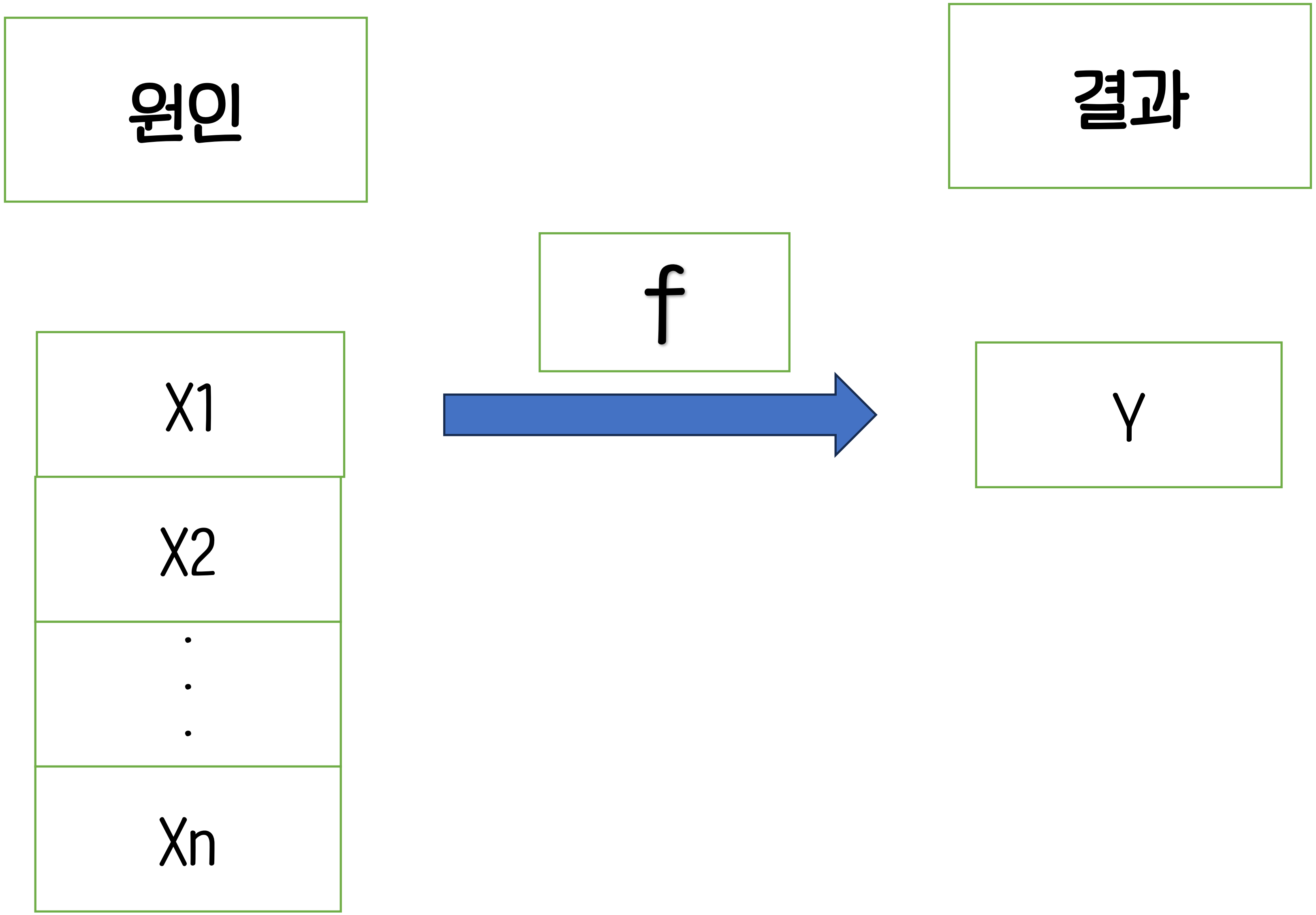
Kuggle

1. 회귀분석



1. 회귀분석

회귀분석 소개



1. 회귀분석

회귀분석 소개

모형화(modeling)

Y=f(x1, x2, x3, x4,······ xn)

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \varepsilon$$

1. 회귀분석

회귀식 소개

Regression Model

- Simple (단순)
  - Linear(선형)
    $y=bx+b0+\varepsilon$
  - Nonlinear (비선형)
    $y=a\cdot e^{bx}+\varepsilon$
- Multiple (다중)
  - Linear(선형)
    $y=\beta_0+\beta_1X_1+\beta_2X_2+\cdots+\beta_pX_p+\varepsilon$
  - Nonlinear (비선형)
    $y=\beta_0+\beta_1\cdot X_1+\beta_2\cdot X^2+\beta_3\cdot \sin(X_3)+\varepsilon$

1. 회귀분석

단순선형회귀

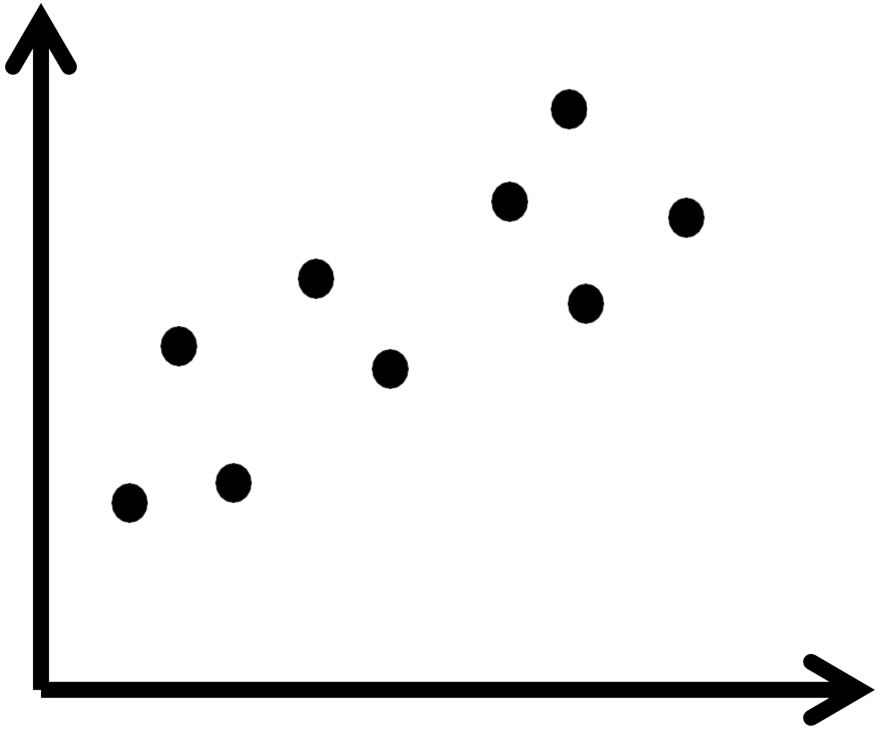
Simple linear regression

단순선형회귀 모델 $Y = \beta_0 + \beta_1 X + \varepsilon$

- β_0: constant, intercept
- β_1: slope, coefficient
- ε: error, 오차, x로 설명되지 않는 어떤 것

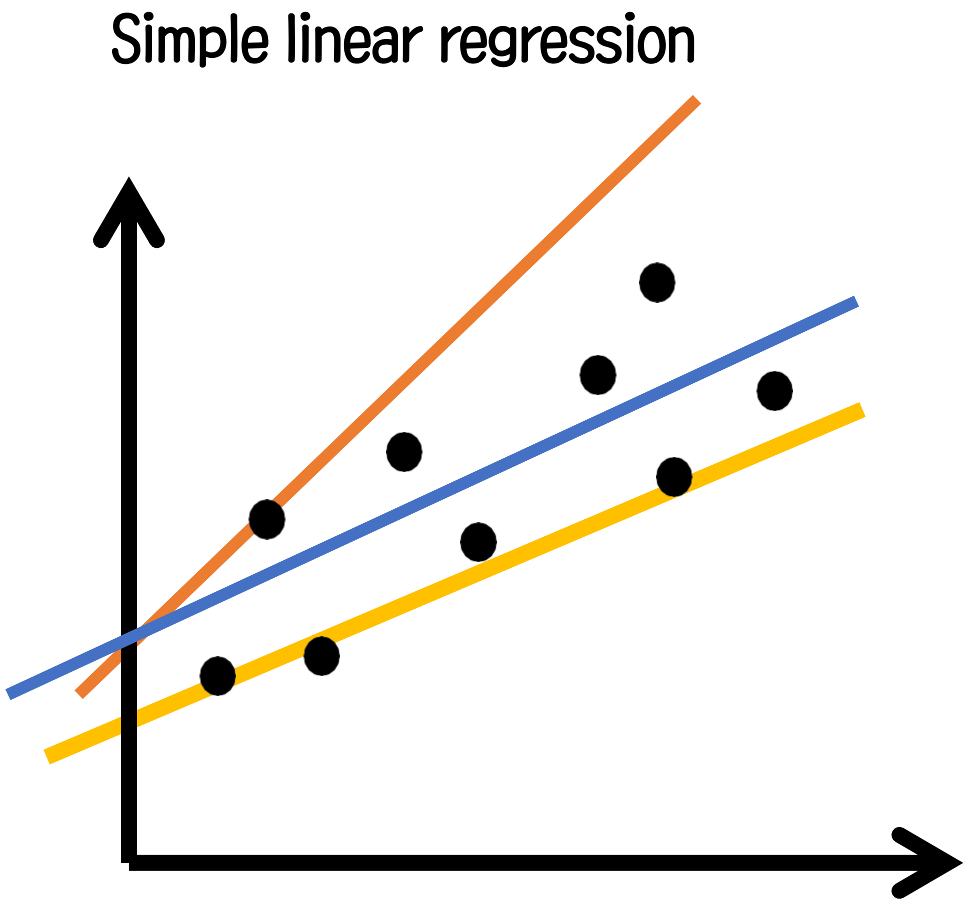
1. 회귀분석

단순선형회귀



1. 회귀분석

단순선형회귀

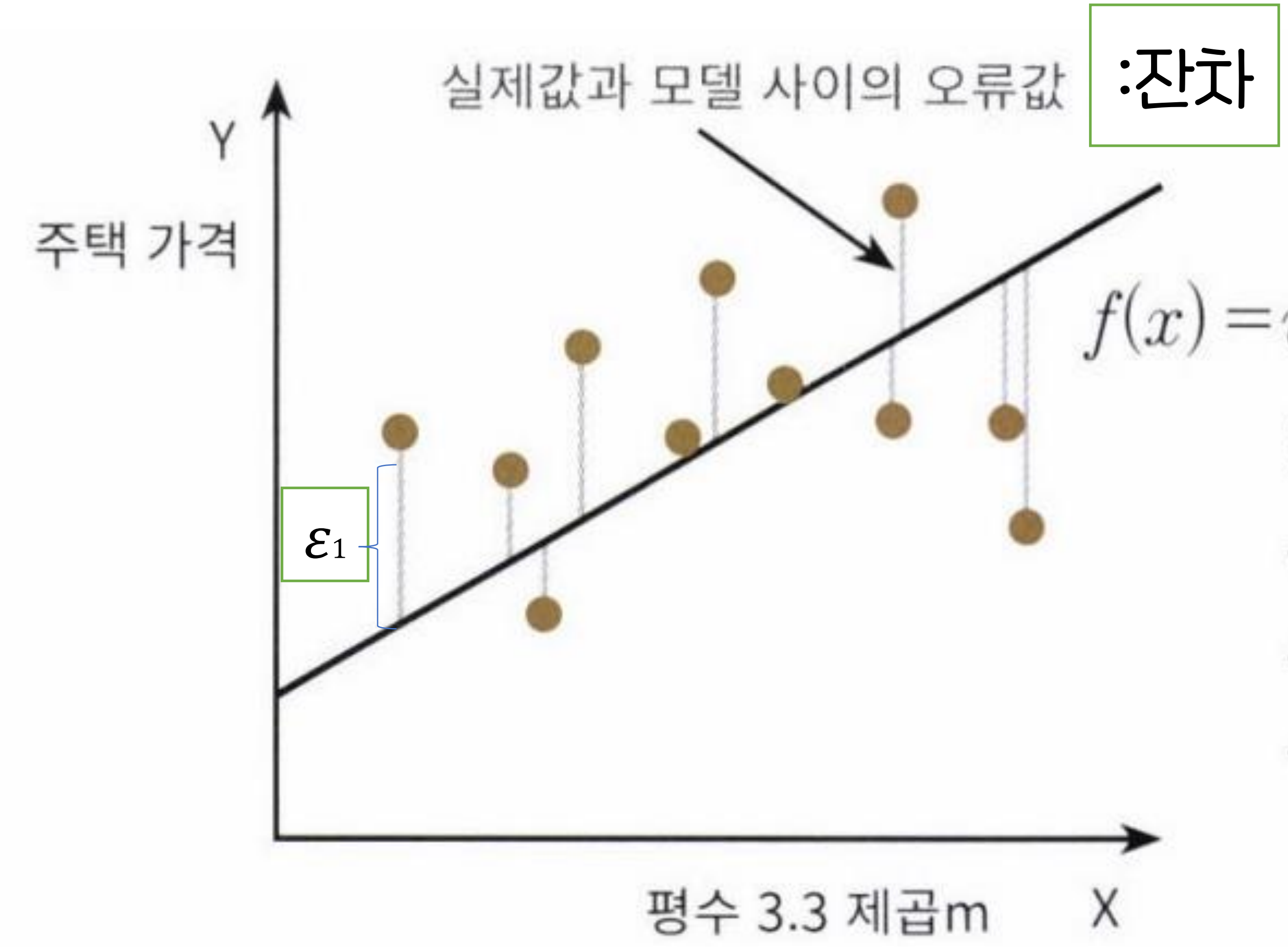


파란 선: 최적의 회귀직선

실제 관측된 값과 우리가 고려하고 있는
여러가지 직선들 사이의 거리를 측정하여
그 거리를 가장 작게 해주는 선을 찾으면
이 선이 데이터를 가장 잘 설명해주는 선이다.
-> 최소제곱의 아이디어

1. 회귀분석

RSS



$$f(x) = w_0 + w_1 * x$$

RSS = (#1 주택 가격 – (w_0+w_1 #1 주택크기)^2

+ (#2 주택 가격 – (w_0+w_1 #2 주택크기)^2

+ (#3 주택 가격 – (w_0+w_1 #3 주택크기)^2

+ ⋯ (모든 학습 데이터에 대해 RSS 수행)

1. 회귀분석

RSS

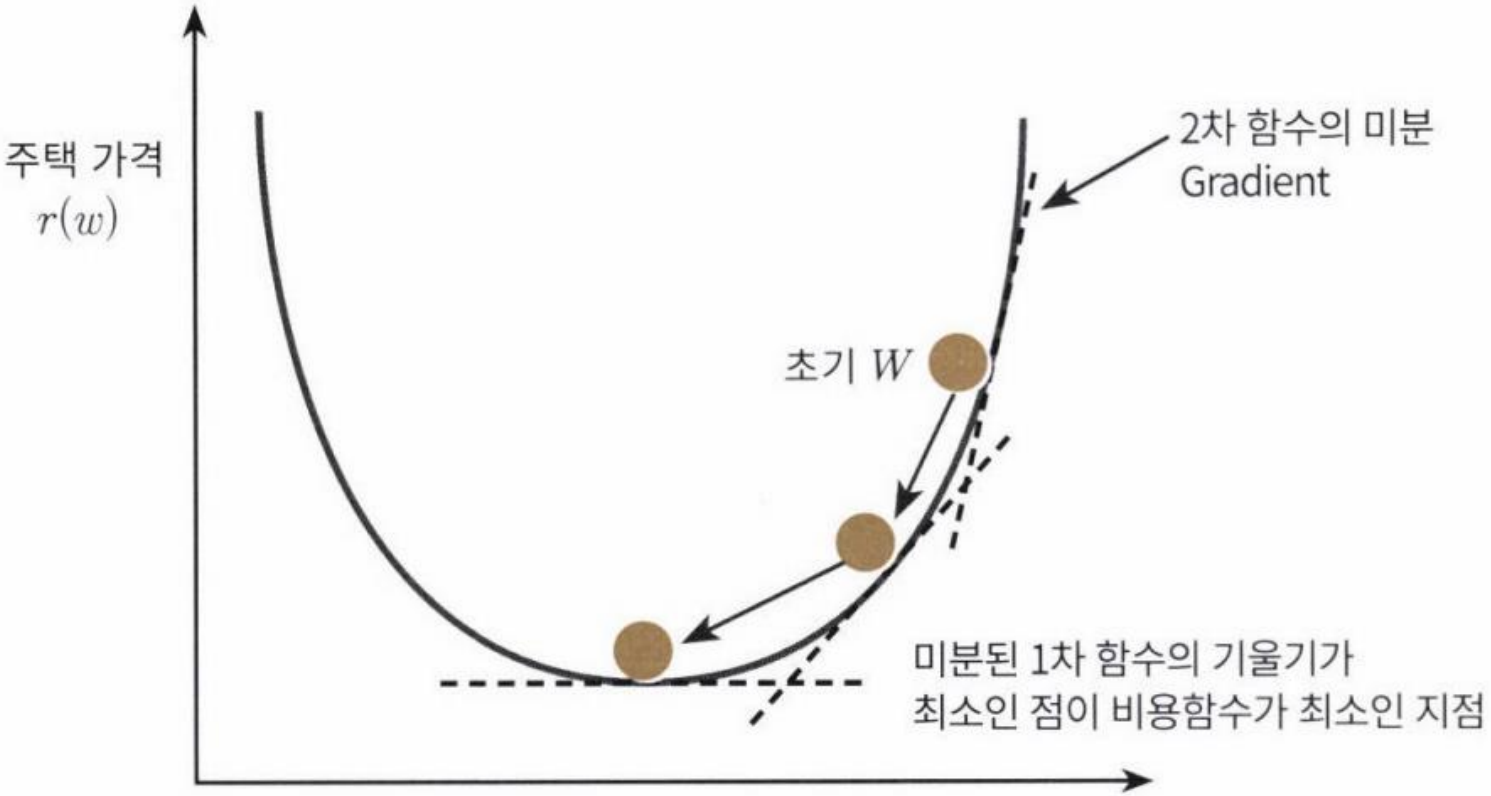
$$RSS(w_0, w_1) = \frac{1}{N}\sum_{i=1}^{N}(y_i - (w_0 + w_1 * x_i))^2$$

(i는 1부터 학습 데이터의 총 건수 N까지)

회귀에서 이 RSS는 비용이라 하며
이 비용을 최소로 하게하는 W_0 ,W_1을 학습을 통해 찾는 것이
머신러닝 기반 회귀의 핵심 사항이다.

1. 회귀분석

경사 하강법



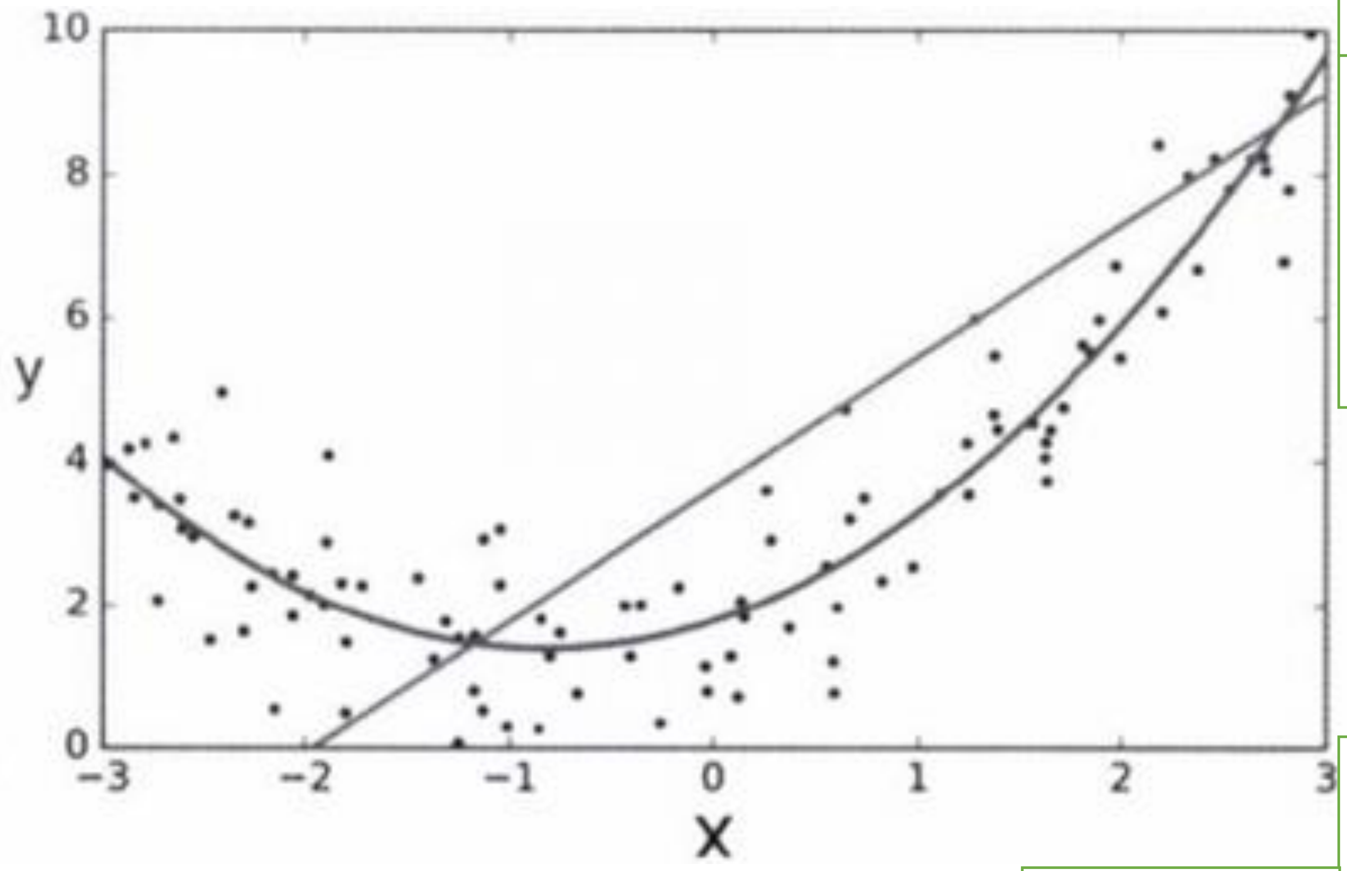
r(w)를 각각 w_0, w_1으로 순차적으로 편미분 하여 R(w)를 최소화 하는 w_0, w_1 구하기

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

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

1. 회귀분석

다항회귀



다항회귀

y=$w0 + w1 * x1 + w2 * x2 + w3 * x1 * x2 + w4 * x1^2$+$w5 * x2^2$

Z=[x1, x2, x1*x2, $x1^2$, $x2^2$]

선형회귀

y=w0+w1*z1+w2*z2+w3*z3+w4*z4+w5*z5

2. 평가지표 및 실습

2. 평가지표 및 실습

평가지표

평가 지표	설명	수식
MAE	Mean Absolute Error(MAE)이며 실제 값과 예측값의 차이를 절댓값으로 변환해 평균한 것입니다.	$MAE = \frac{1}{n}\sum_{i=1}^{n}\lvert Yi - \hat{Y}i \rvert$
MSE	Mean Squared Error(MSE)이며 실제 값과 예측값의 차이를 제곱해 평균한 것입니다.	$MSE = \frac{1}{n}\sum_{i=1}^{n}(Yi - \hat{Y}i)^2$
RMSE	MSE 값은 오류의 제곱을 구하므로 실제 오류 평균보다 더 커지는 특성이 있으므로 MSE에 루트를 씌운 것이 RMSE(Root Mean Squared Error)입니다.	$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Yi - \hat{Y}i)^2}$
R^2	분산 기반으로 예측 성능을 평가합니다. 실제 값의 분산 대비 예측값의 분산 비율을 지표로 하며, 1에 가까울수록 예측 정확도가 높습니다.	$R^2 = \frac{\text{예측값 } Variance}{\text{실제값 } Variance}$

RMSE를 구하는 이유?

MSE의 단점이 뚜렷하기 때문

1. 오차의 합을 제곱한 것이기 때문에 에러의 차원과 MSE의 차원이 서로 다름

2. 제곱값이기 때문에 값이 매우 커질 수 있음

-> 루트만 씌웠을 뿐인데 단점을 해결할 수 있음

결정계수 R^2

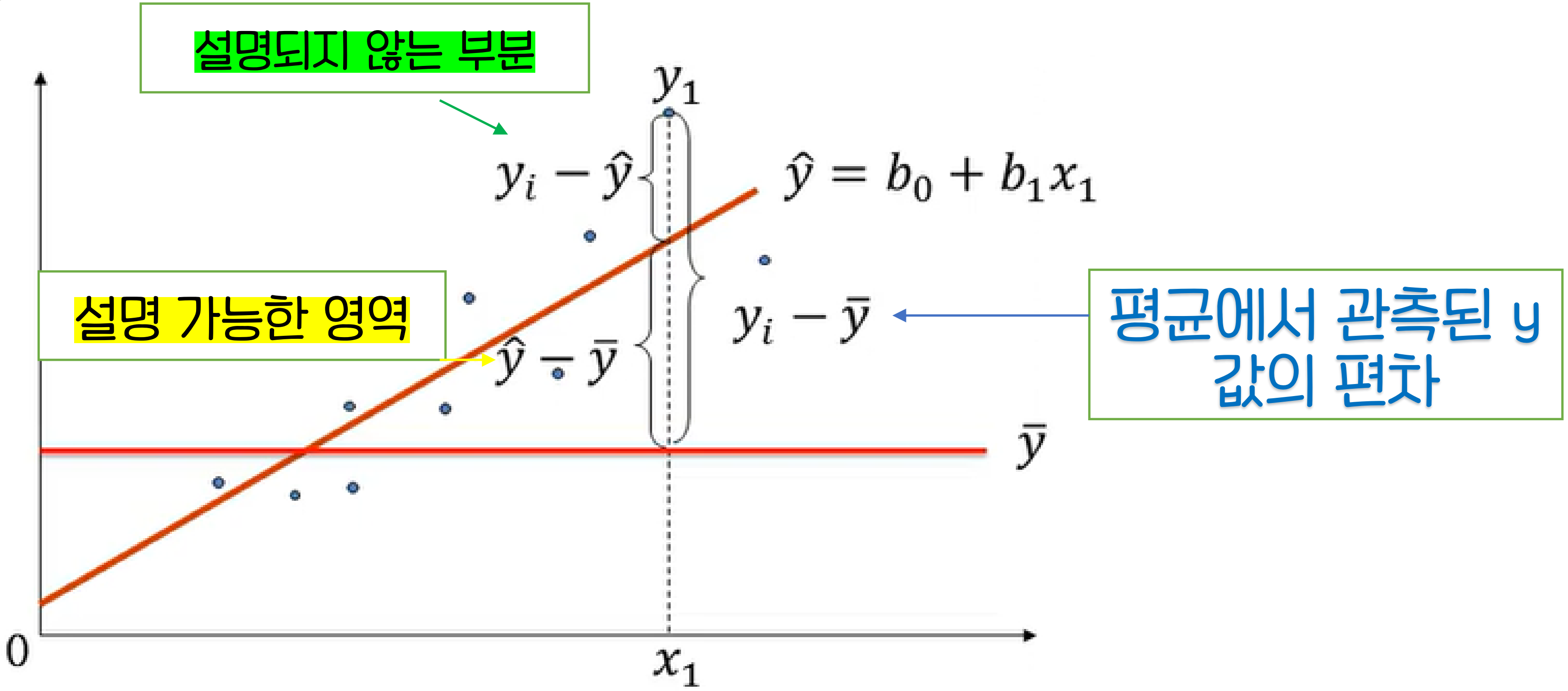
- 회귀선에 의해 종속변수가 설명되어지는 정도를 나타낸 것
- $R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST}$
- $0 \leq R^2 \leq 1$

(0에 가까우면 데이터를 잘 설명하지 못하는 회귀직선,

1에 가까우면 데이터를 잘 설명하는 회귀직선)

2. 평가지표 및 실습

합의 제곱 분해



$$\sum (y_i - \bar{y})^2 = \sum (\hat{y} - \bar{y})^2 + \sum (y_i - \hat{y})^2$$

SST = SSR + SSE

2. 평가지표 및 실습

코드

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
from scipy import stats
from sklearn.datasets import fetch_california_housing
from sklearn.datasets import fetch_openml
%matplotlib inline

# california 데이타셋 로드
housing = fetch_openml(name="house_prices", as_frame=True)

# california 데이타셋 DataFrame 변환
california = fetch_california_housing(as_frame=True)
californiaDF = california['frame']
californiaDF.head()
```

데이터 시각화

데이터 시각화

통계 관련 함수

California Housing 데이터셋

2. 평가지표 및 실습

데이터 설명

변수	설명
MedInc	중간 소득
House Age	중간 주택 연도
Ave Rooms	평균 방의 수
Ave Bedrms	평균 침실의 수
Population	인구
Ave Occup	평균 주택점유율
Latitude	위도
Longitude	경도

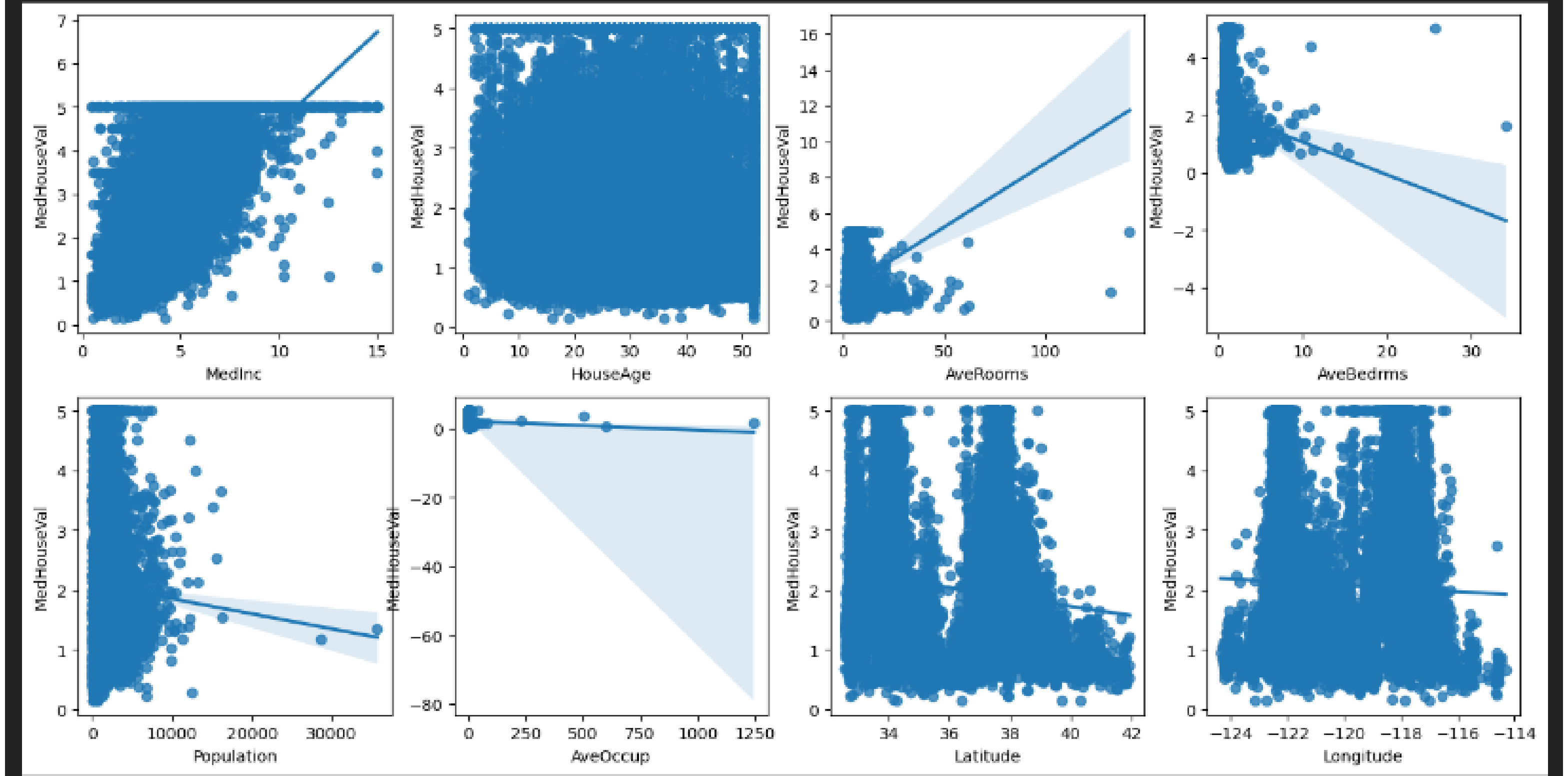
	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude	MedHouseVal
0	8.3252	41.0	6.984127	1.023810	322.0	2.555556	37.88	-122.23	4.526
1	8.3014	21.0	6.238137	0.971880	2401.0	2.109842	37.86	-122.22	3.585
2	7.2574	52.0	8.288136	1.073446	496.0	2.802260	37.85	-122.24	3.521
3	5.6431	52.0	5.817352	1.073059	558.0	2.547945	37.85	-122.25	3.413
4	3.8462	52.0	6.281853	1.081081	565.0	2.181467	37.85	-122.25	3.422

2.실습

실습

```
# 2개의 행과 4개의 열을 가진 subplots를 이용. axs는 4x2개의 ax를 가짐.
fig, axs = plt.subplots(figsize=(16, 8), ncols=4, nrows=2)
lm_features = ['MedInc','HouseAge','AveRooms', 'AveBedrms','Population', 'AveOccup',
               'Latitude', 'Longitude']

for i, feature in enumerate(lm_features):
    row = int(i / 4)
    col = i % 4
    # 시본의 regplot을 이용해 산점도와 선형 회귀 직선을 함께 표현
    sns.regplot(x=feature, y='MedHouseVal', data=californiaDF, ax=axs[row][col])
```



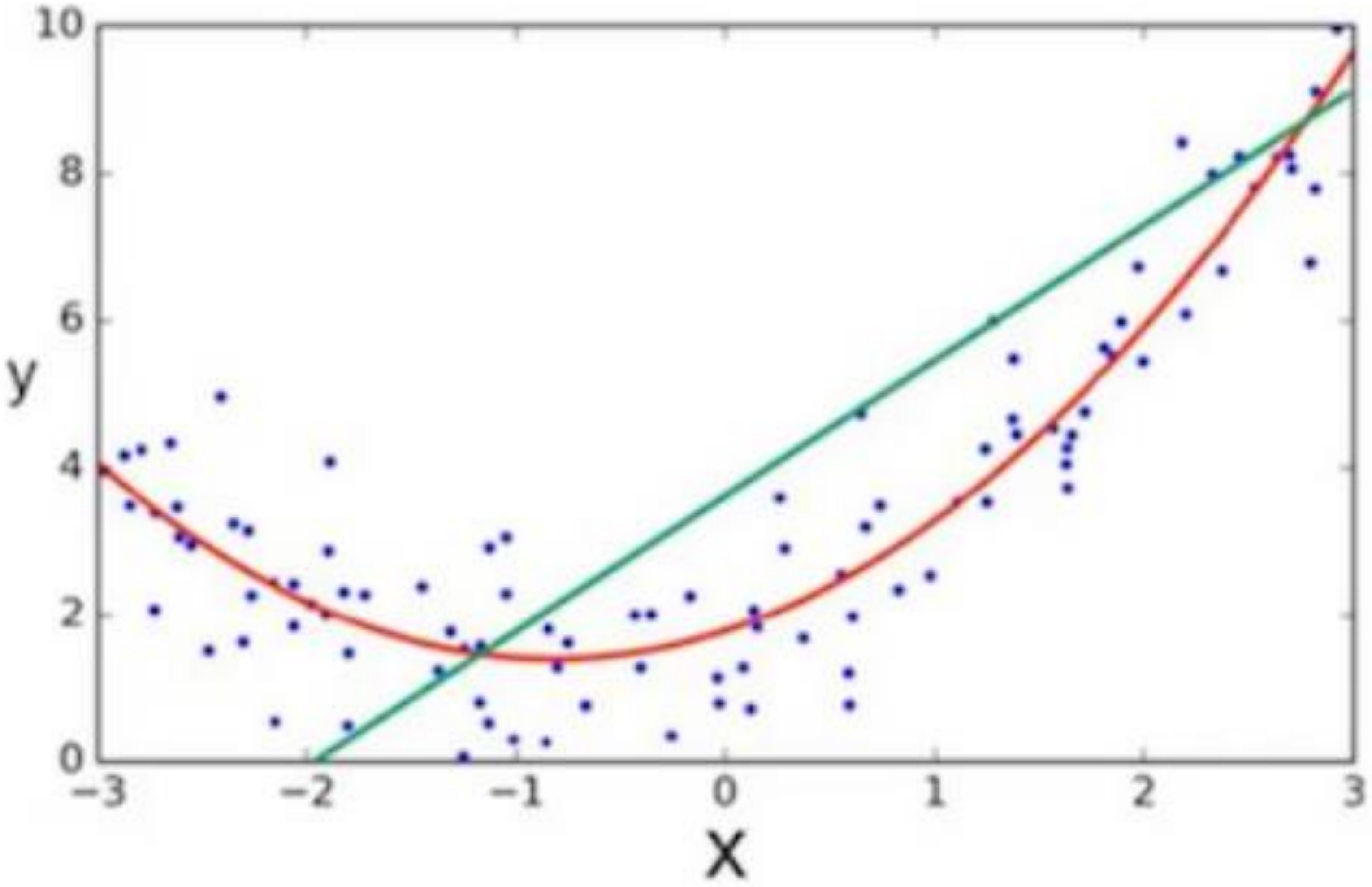
Kuggle

3. 과(대)/과소 적합

3. 과(대)/과소 적합

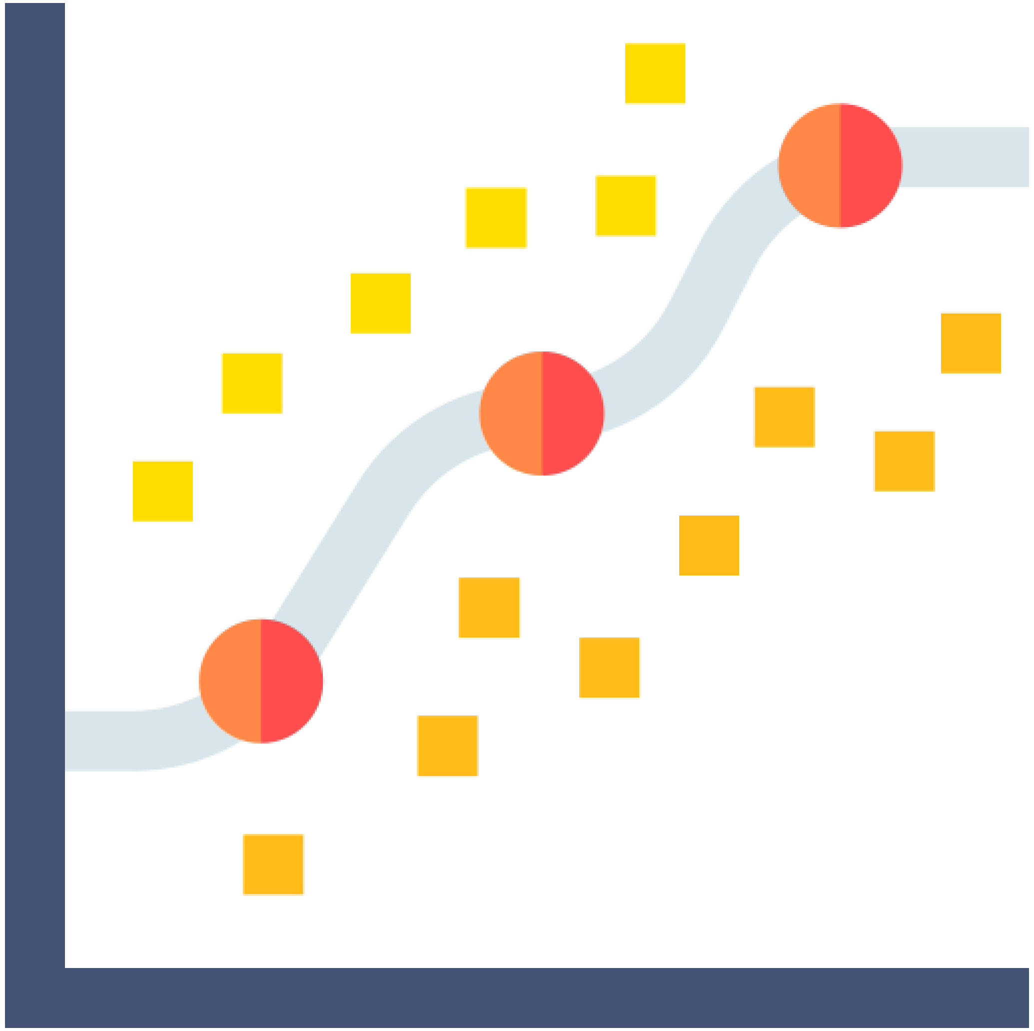
다항회귀

적절한 모델은 직선인가 곡선인가?



3. 과(대)/과소 적합

다항회귀



다항회귀

- 독립변수가 다항식으로 표현
- 복잡한 비선형 관계 모델링
- 적절한 **차수** 선택이 중요

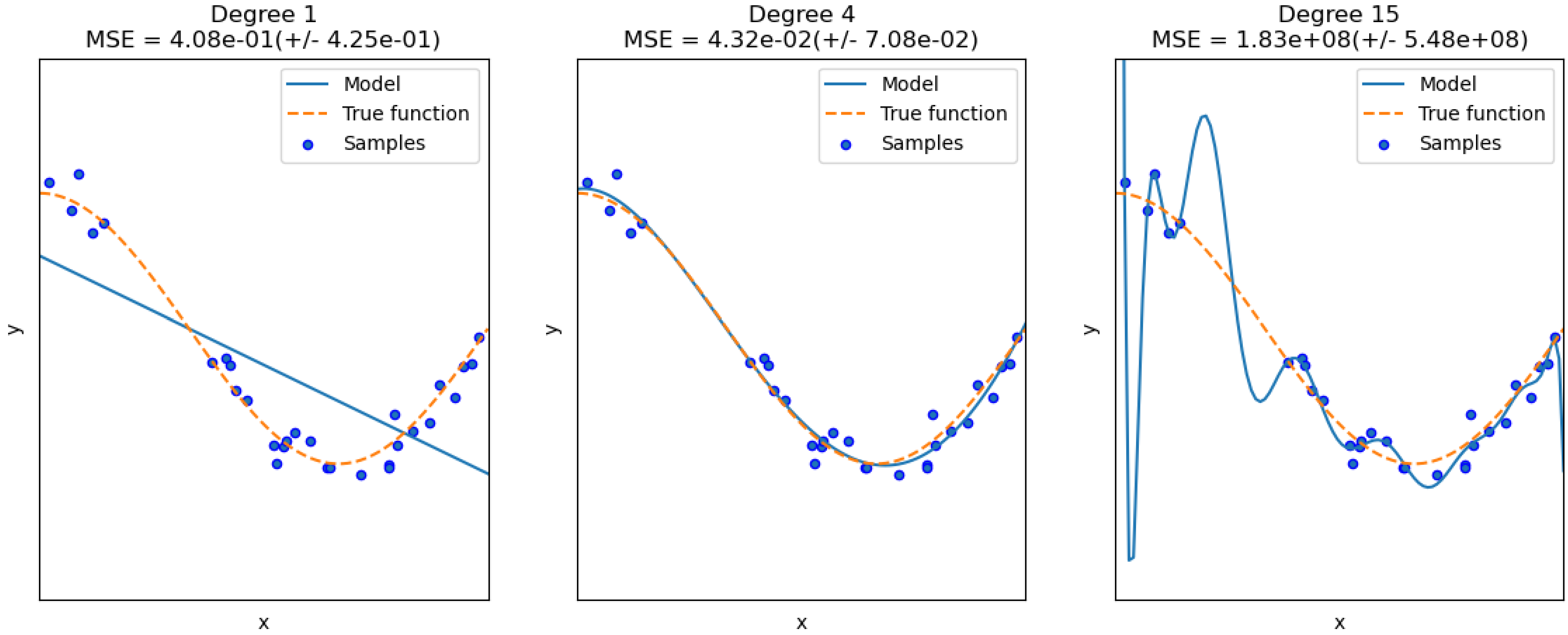
차수(degree) ↑

장점 : 복잡한 변수(피처) 모델링 가능

단점 : 예측 정확도가 떨어질수 있음

3. 과(대)/과소 적합

차수 적용



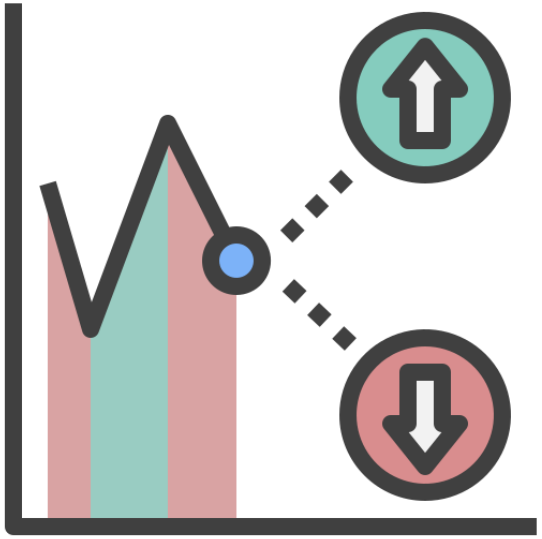
High Bias

High Variance

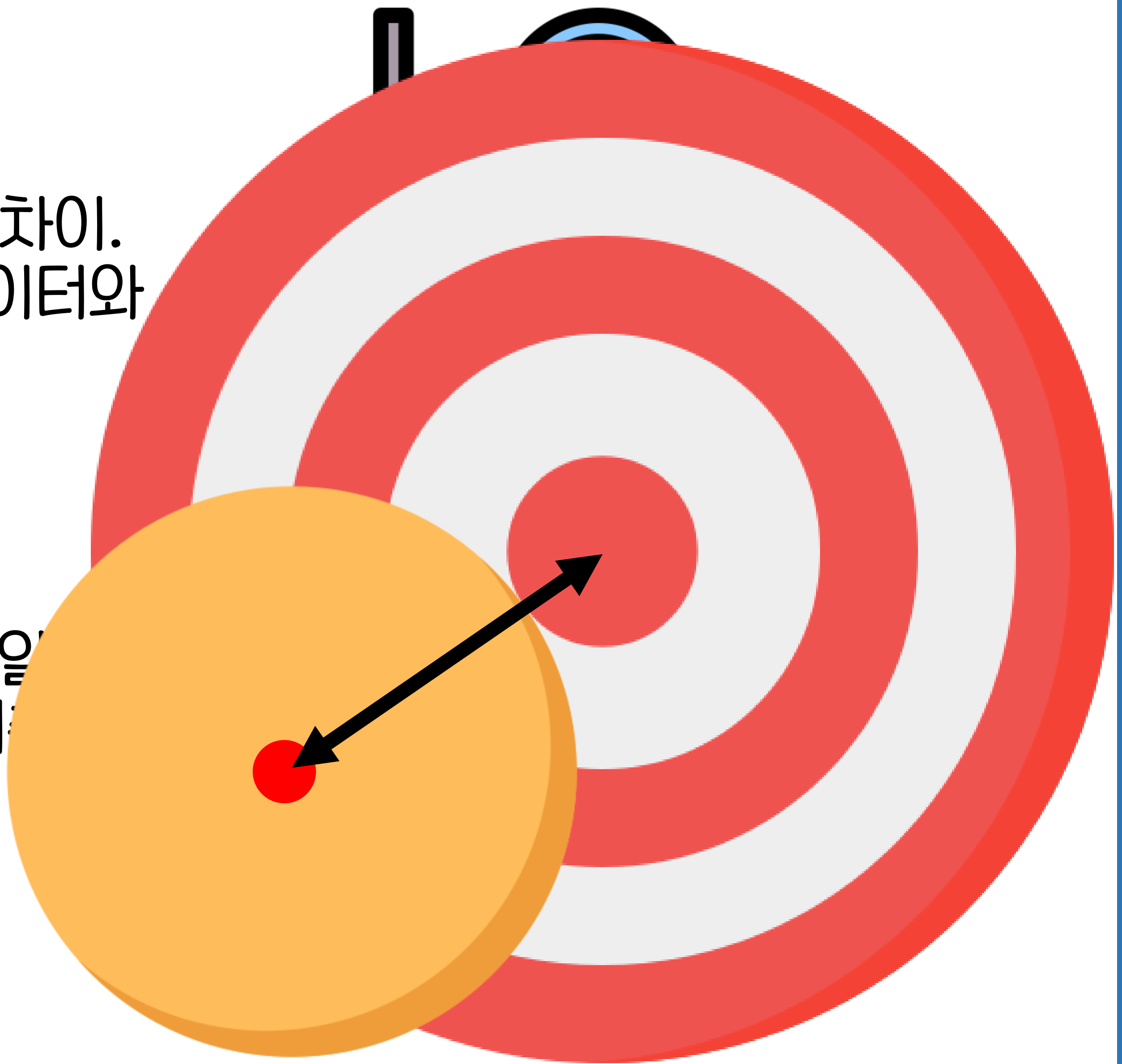
3. 과(대)/과소 적합

편향, 분산

편향(Bias) : 모델의 예측값과 실제 값의 차이.
추정값들의 중심이 실제 데이터와
얼마나 떨어져 있는가?

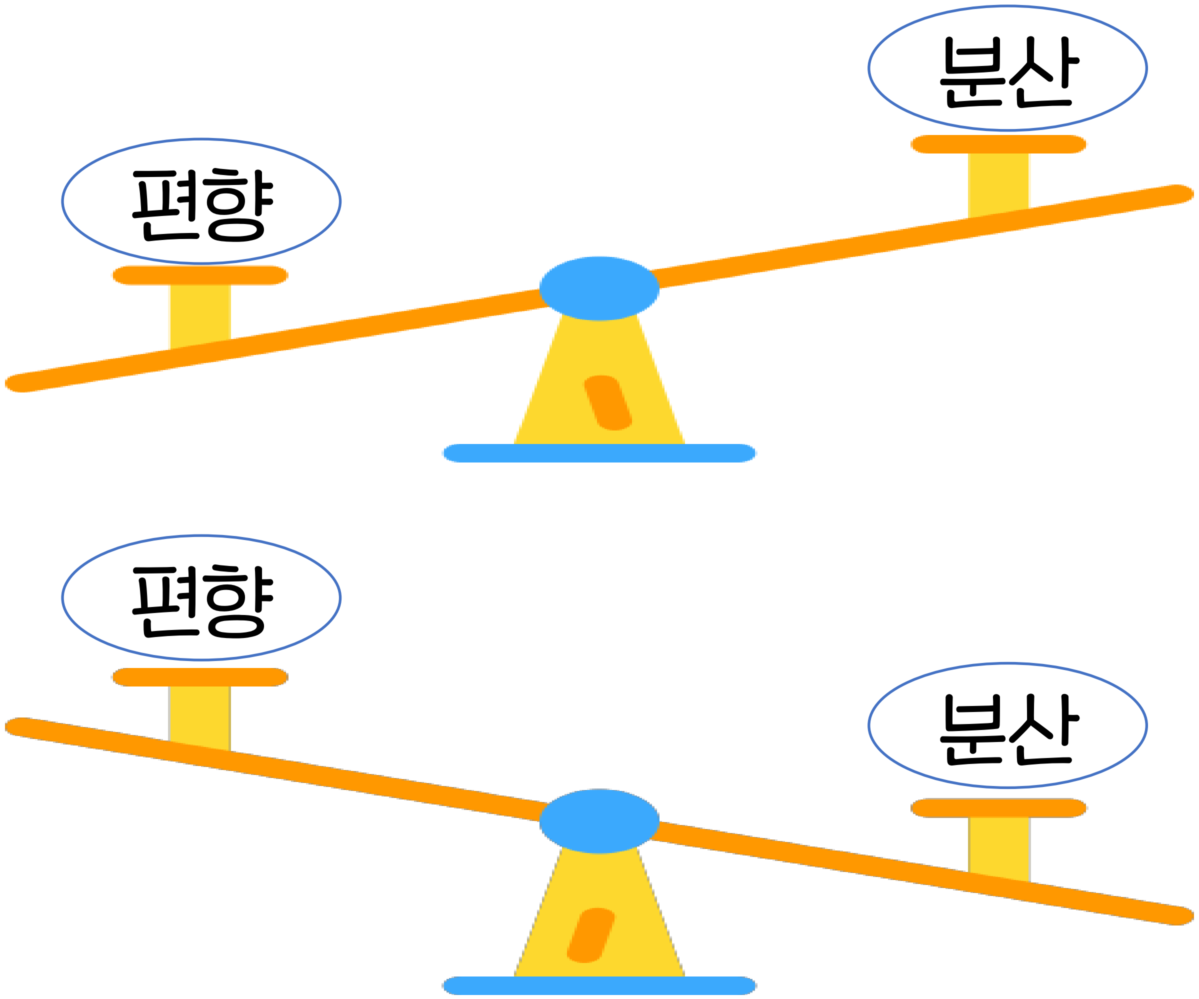


분산(Variance) : 동일
예



3. 과(대)/과소 적합

과적합/과소적합



과(대)적합

과소적합

3. 과(대)/과소 적합

dilemma

편향-분산 트레이드오프

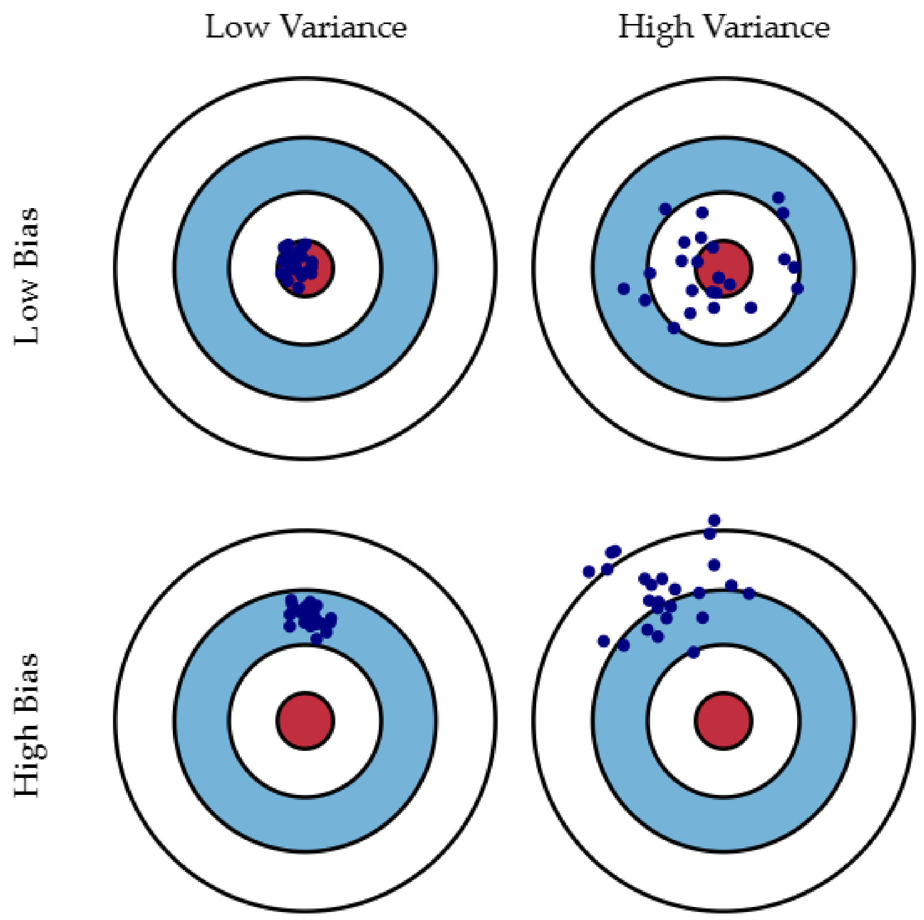


Fig. 1 Graphical illustration of bias and variance.

https://scott.fortmann-roe.com/docs/BiasVariance.html

Degree 1 Model

- 매우 단순화된 모델
- 지나치게 한 방향성으로 치우침
- **고편향(High Bias)**

Degree 15 Model

- 매우 복잡한 모델
- 지나치게 높은 변동성
- **고분산 (High Variance)**

3. 과(대)/과소 적합

골디락스 지점

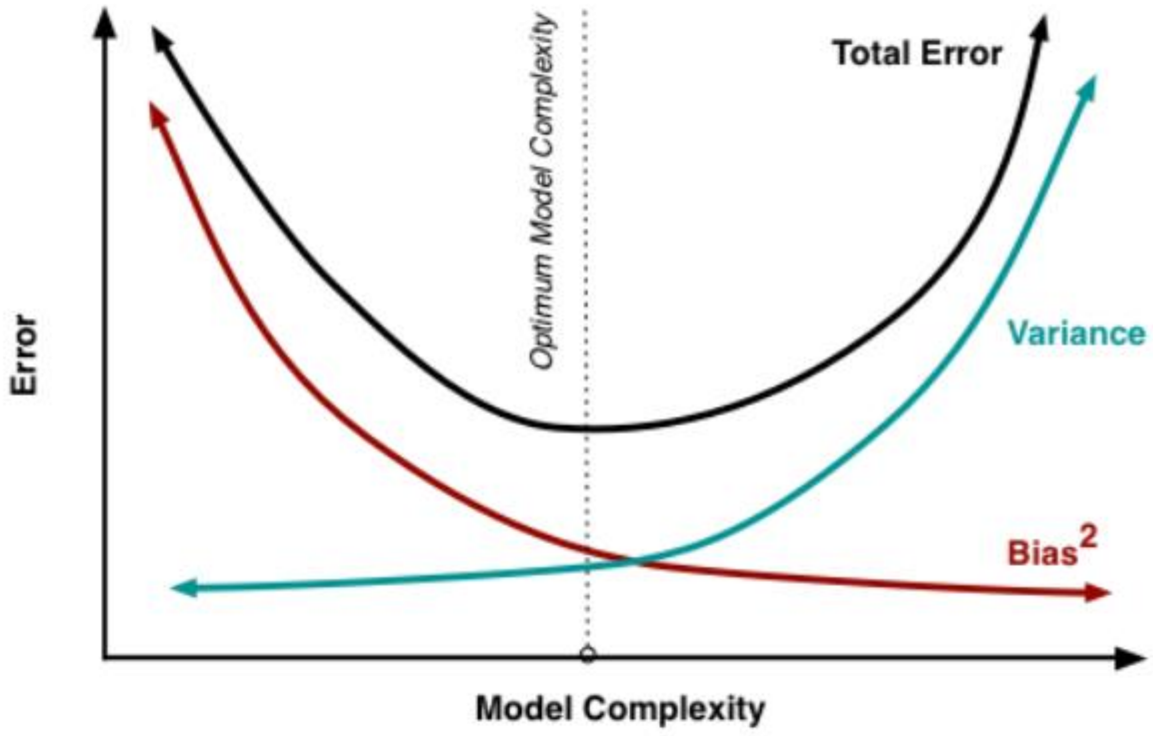


Fig. 6 Bias and variance contributing to total error.

https://scott.fortmann-roe.com/docs/BiasVariance.html

'골디락스' 지점

- 최적화 지점
- 편향은 낮추고 분산은 높여 전체 오류가 가장 낮아지는 점

Kuggle

4. 규제 선형 모델

4. 규제 선형 모델

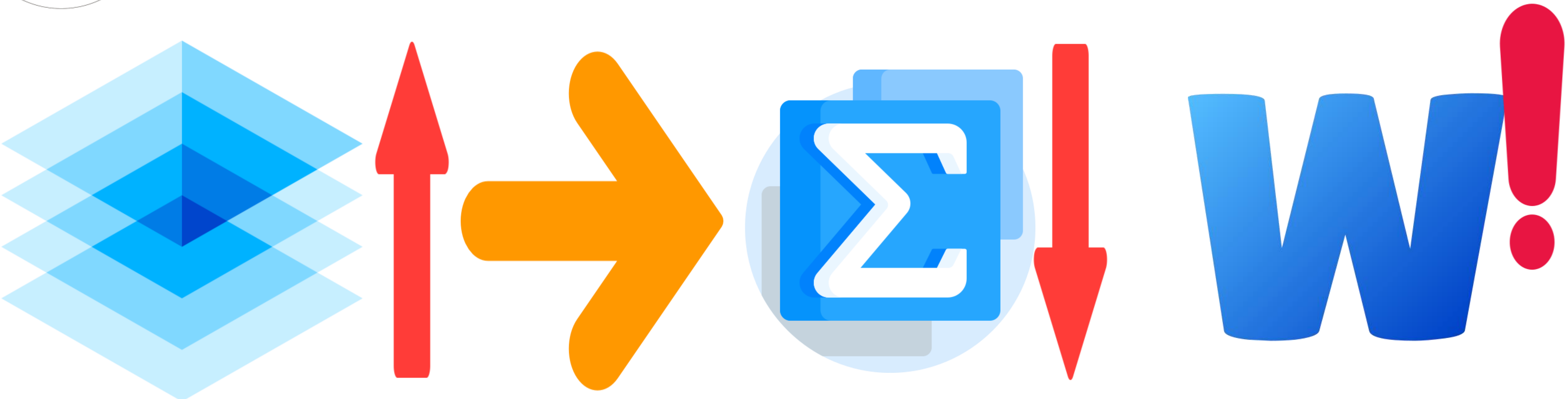
좋은 머신러닝 회귀 모델

비용함수가
최적화된 모델

Balance!

4. 규제 선형 모델

차원의 저주(Curse of Dimensionality)



Dim이 커지면
(Feature가 많아지면)

RSS는 작아지고

회귀계수에 영향

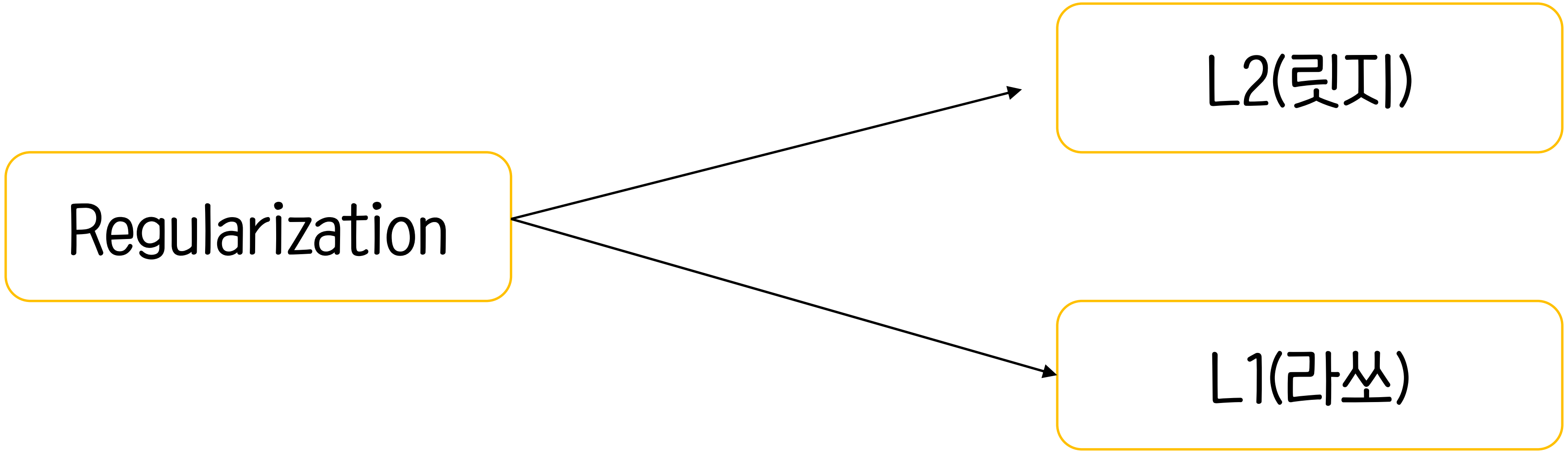
과적합 문제, 테스트데이터에서 예측 성능 저하

4. 규제 선형 모델

규제 필요

$$비용\ 함수\ 목표 = \mathrm{Min}(\mathrm{RSS}(W) + \mathrm{alpha} * \| W \|_2^2)$$

규제(Regularization) : alpha로 패널티 부여해 과적합을 개선



4. 규제 선형 모델

Ridge regression

```
# 필요한 라이브러리 임포트
from sklearn.model_selection import train_test_split
from sklearn.linear_model import Ridge
from sklearn.metrics import mean_squared_error

# 데이터 로드
df = load_df()
X, y = df.data, df.target

# 데이터 분할 (훈련 세트와 테스트 세트로)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# 릿지 회귀 모델 생성 및 학습
ridge = Ridge(alpha=1.0) # alpha는 규제 강도를 조절하는 파라미터
ridge.fit(X_train,y_train)

# 예측 수행
y_pred = ridge.predict(X_test)

# 성능 평가 (RMSE 계산)
mse = mean_squared_error(y_test,y_pred)
rmse = mse ** 0.5 # or np.sqrt(mse)
print('RMSE: ', rmse)
```

4. 규제 선형 모델

Lasso regression

```
# 필요한 라이브러리 임포트
from sklearn.model_selection import train_test_split
from sklearn.linear_model import Lasso
from sklearn.metrics import mean_squared_error

# 데이터 로드
df = load_df()
X, y = df.data, df.target

# 데이터 분할 (훈련 세트와 테스트 세트로)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# 라쏘 회귀 모델 생성 및 학습
lasso = Lasso(alpha=1.0) # alpha는 규제 강도를 조절하는 파라미터
lasso.fit(X_train,y_train)

# 예측 수행
y_pred = lasso.predict(X_test)

# 성능 평가 (RMSE 계산)
mse = mean_squared_error(y_test,y_pred)
rmse = mse ** 0.5 # or np.sqrt(mse)
print('RMSE: ', rmse)
```

4. 규제 선형 모델

Elastic net regression

라쏘회귀

- 상관계수 높으면 중요 피처만 선택
- 다른 피처 회귀계수 0으로 하는 성향

Alpha에 따라 회귀 계수가 급변

$$RSS(W) + alpha2 * \|W\|_2^2 + alpha1 * \|W\|_1$$

* 엘라스틱 넷에서 규제

$a \cdot L1 + b \cdot L2$

* 엘라스틱 넷에서 α 파라미터

$a+b$

* 엘라스틱 넷에서 l1_ratio 파라미터

$\frac{a}{a+b}$

if) i) l1_ratio = 0 → L2 규제

ii) l1_ratio = 1 → L1 규제

4. 규제 선형 모델

Elastic net regression

```
# 필요한 라이브러리 임포트
from sklearn.model_selection import train_test_split
from sklearn.linear_model import ElasticNet
from sklearn.metrics import mean_squared_error

# 데이터 로드
df = load_df()
X, y = df.data, df.target

# 데이터 분할 (훈련 세트와 테스트 세트로)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# 엘라스틱넷 회귀 모델 생성 및 학습
# alpha는 규제 강도를 조절하는 파라미터이고 l1_ratio는 L1 규제의 비율을 의미
elastic_net = ElasticNet(alpha=1.0 , l1_ratio=0.5)
elastic_net.fit(X_train,y_train)

# 예측 수행
y_pred = elastic_net.predict(X_test)

# 성능 평가 (RMSE 계산)
mse = mean_squared_error(y_test,y_pred)
rmse = mse ** 0.5 # or np.sqrt(mse)
print('RMSE: ', rmse)
```

2주차 과제

과제

1. y=10+9X+e에 해당하는 자료(산점도) 구성하기.

2. y에는 'MedHouseVal', X에는 'MedHouseVal'를 제외한 나머지 피처를 이용하여 다중회귀 모델 만들기

3. 릿지회귀

```
15]: # 앞의 Lin
from sklea
from sklea

#릿지회귀
ridge =
#교차검증
neg_mse_sc
```

4. MAE가 아닌 RMSE를 사용하는 이유는 무엇인가요?

MAE는 실제 값과 예측값의 차이를 절댓값으로 변환해 평균낸 것입니다. MAE는 다른 지표들에 비해 직관적이라는 특징을 가지고 있는데, 이러한 장점을 가진 MAE가 아닌 RMSE를 사용하는 이유는 무엇인지 설명해주세요.

- 참고자료: https://data101.oopy.io/mae-vs-rmse

5.편향-분산 트레이드 오프에 대해서 자세히 설명후 해결방안을 찾아주세요