회귀 알고리즘

회귀 소개

- 회귀는 현대 통계학을 이루는 큰 축
- 회귀 분석은 유전적 특성을 연구하던 영국의 통계학자 갈톤(Galton)이 수행한 연구에서 유래했다는 것이 일반론

"부모의 키가 크더라도 자식의 키가 대를 이어 무한정 커지지 않으며,
부모의 키가 작더라도 대를 이어 자식의 키가 무한정 작아지지 않는다."

- 회귀 분석은 이처럼 데이터 값이 평균과 같은 일정한 값으로 돌아가려는 경향을 이용한 통계학 기법

선형 회귀의 종류

- 일반 선형 회귀(LinearRegression)
  - 예측값과 실제 값의 RSS(Residual Sum of Squares)를 최소화할 수 있도록 회귀 계수를 최적화하며, 규제(Regularization)를 적용하지 않은 모델
- 릿지(Ridge)
  - 릿지 회귀는 선형 회귀에 L2 규제를 추가한 회귀 모델
- 라쏘(Lasso)
  - 라쏘 회귀는 선형 회귀에 L1 규제를 적용한 방식
- 엘라스틱넷(ElasticNet)
  - L2, L1 규제를 함께 결합한 모델
- 로지스틱 회귀(Logistic Regression)
  - 로지스틱 회귀는 회귀라는 이름이 붙어있지만, 사실은 분류에 사용되는 선형 모델

회귀(Regression) 개요

- 회귀는 여러 개의 독립변수(X)와 한 개의 종속변수(y) 간의 상관관계를 모델링하는 기법을 통칭한다.

아파트 가격

방 개수

아파트 크기

주변 학군

역과의 거리

회귀(Regression) 개요

- 회귀는 여러 개의 독립변수(X)와 한 개의 종속변수(y) 간의 상관관계를 모델링하는 기법을 통칭한다.
  - $y = f(X)$

아파트 가격 | 방 개수 | 아파트 크기 | 주변 학군 | 역과의 거리

$$y = w_1x_1 + w_2x_2 + w_3x_3 + \cdots + w_nx_n$$

- y 는 종속변수, 즉 아파트 가격
- $x_1, x_2, x_3, \cdots, x_n$은 방 개수, 아파트 크기, 주변 학군, 역과의 거리 등 독립 변수
- $w_1, w_2, w_3, \cdots, w_n$은 이 독립 변수의 값에 영향을 미치는 회귀 계수(Regression Coefficients)

회귀(Regression) 개요

- 회귀는 여러 개의 독립변수(X)와 한 개의 종속변수(y) 간의 상관관계를 모델링하는 기법을 통칭한다.
  - $y = f(X)$

아파트 가격 방 개수 아파트 크기 주변 학군 역과의 거리

$$y = w_1x_1 + w_2x_2 + w_3x_3 + \cdots + w_nx_n$$

- y 는 종속변수, 즉 아파트 가격
- $x_1, x_2, x_3, \cdots, x_n$은 방 개수, 아파트 크기, 주변 학군, 역과의 거리 등 독립 변수
- $w_1, w_2, w_3, \cdots, w_n$은 이 독립 변수의 값에 영향을 미치는 회귀 계수(Regression Coefficients)

머신러닝 회귀 예측의 핵심은 주어진 Feature와 Target 데이터 기반에서 학습을 통해 최적의 회귀 계수를 찾아내는 것!

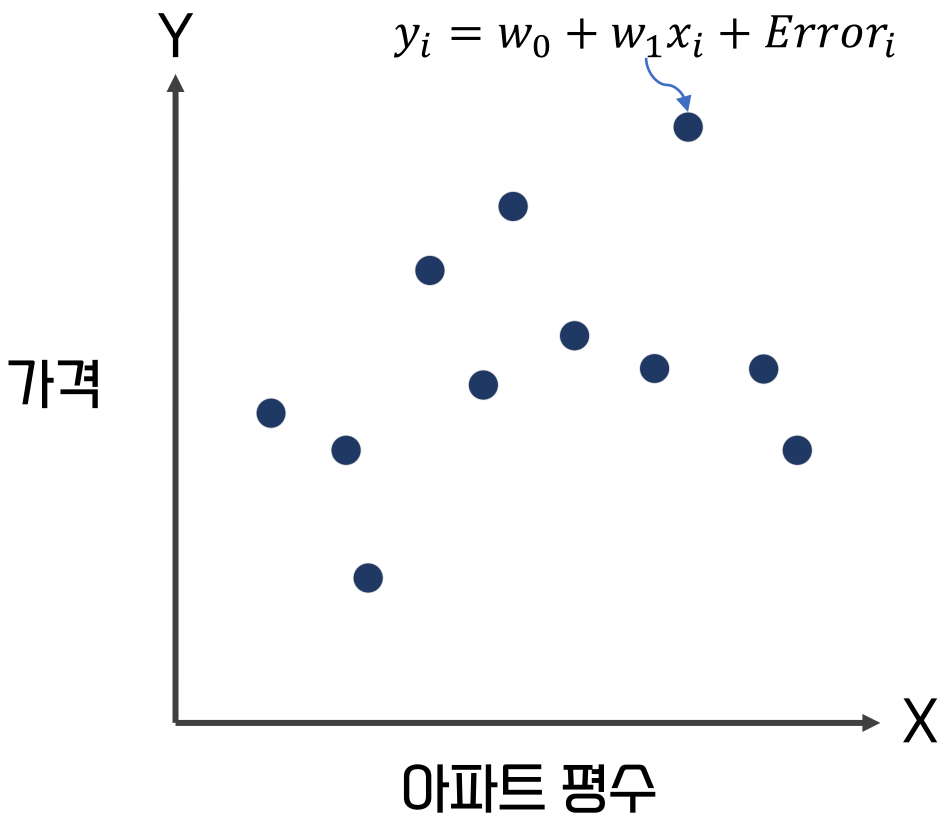
회귀의 유형

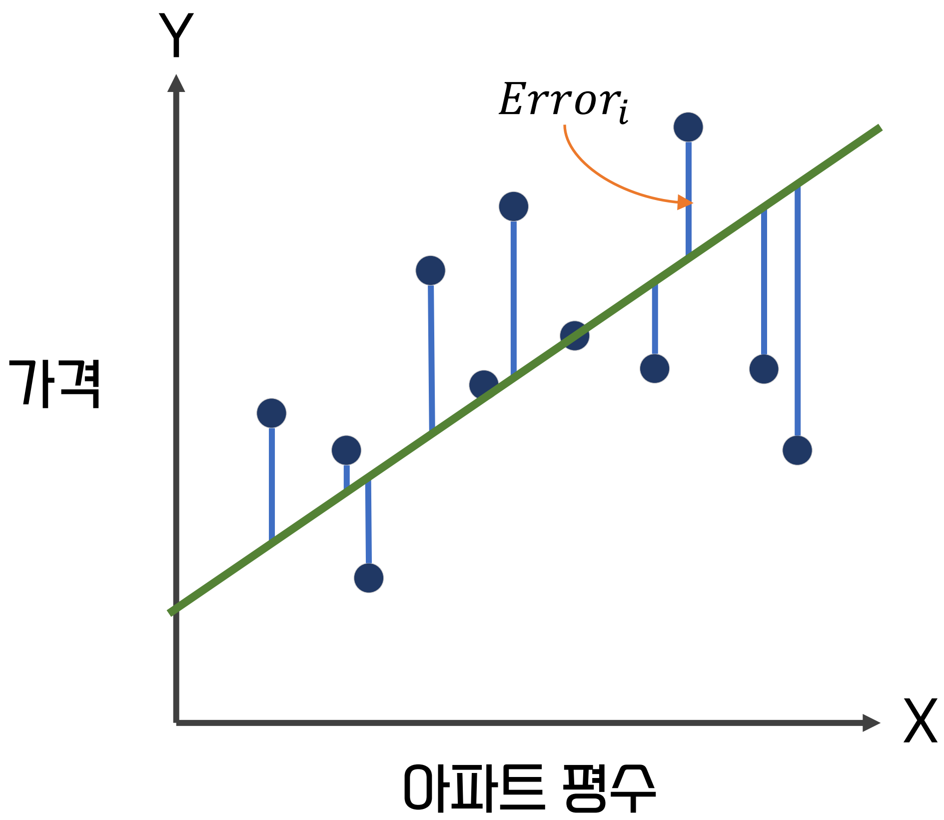
- 회귀는 회귀 계수의 선형/비선형 여부, 독립변수의 개수, 종속변수의 개수에 따라 여러 가지 유형으로 나뉜다.
- 회귀에서 가장 중요한 것은 회귀 계수로서, 이 회귀 계수가 선형인지 아닌지에 따라 선형 회귀와 비선형 회귀로 나눌 수 있다.
- 독립변수의 개수가 한 개인지 여러 개인지에 따라 단일 회귀, 다중 회귀로 나뉘게 된다.
- 정형 데이터의 경우 선형 회귀가 비선형 회귀보다 대부분의 경우 성능이 월등히 좋다.

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

단순 선형 회귀(Simple Regression)를 통한 회귀의 이해

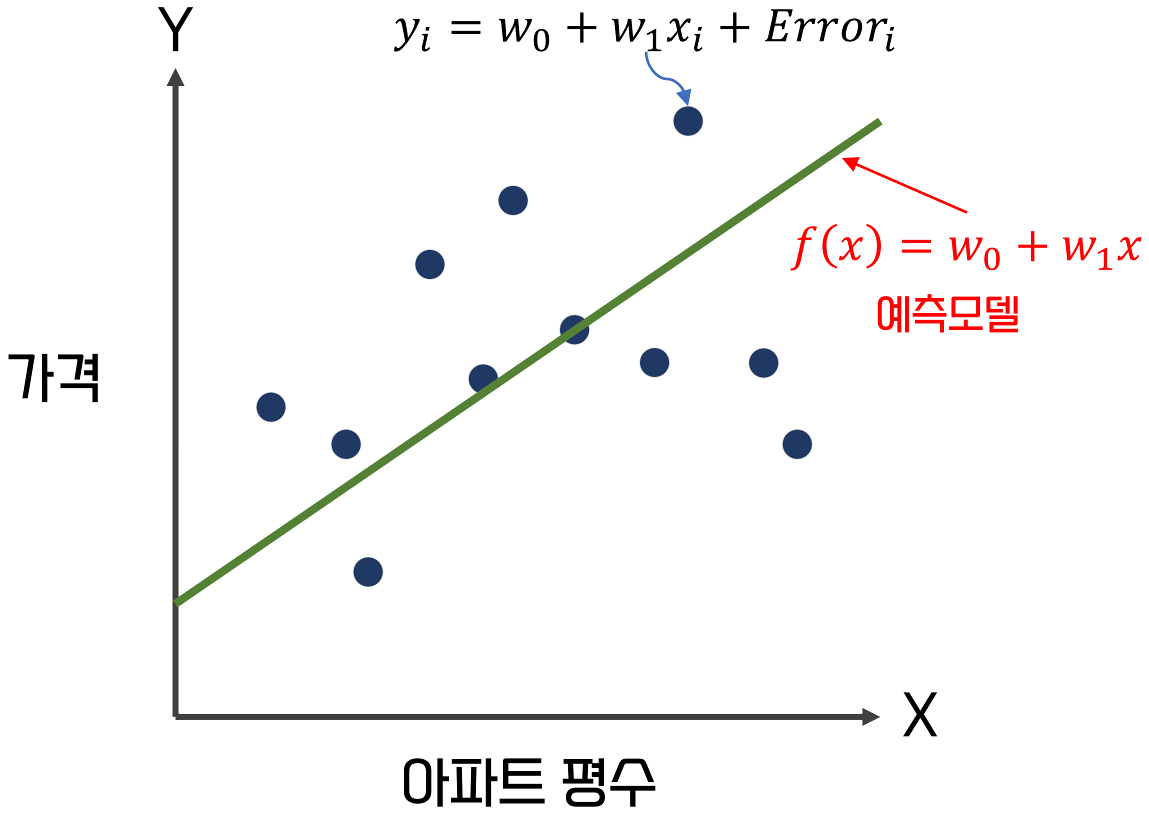
- 주택 가격이 주택의 크기로만 결정 되는 단순 선형 회귀로 가정하면 다음과 같이 주택 가격은 주택 크기에 대해 선형(직선 형태)의 관계로 표현이 가능하다.

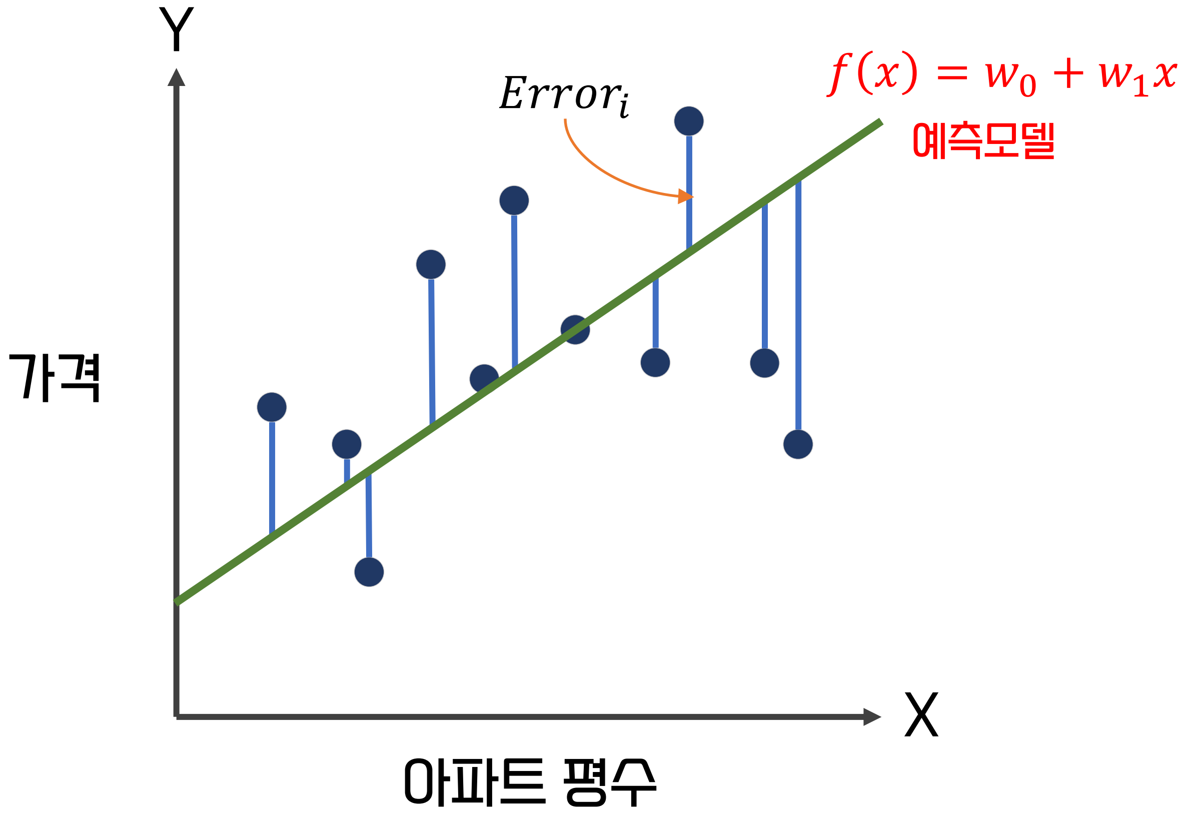




단순 선형 회귀(Simple Regression)를 통한 회귀의 이해

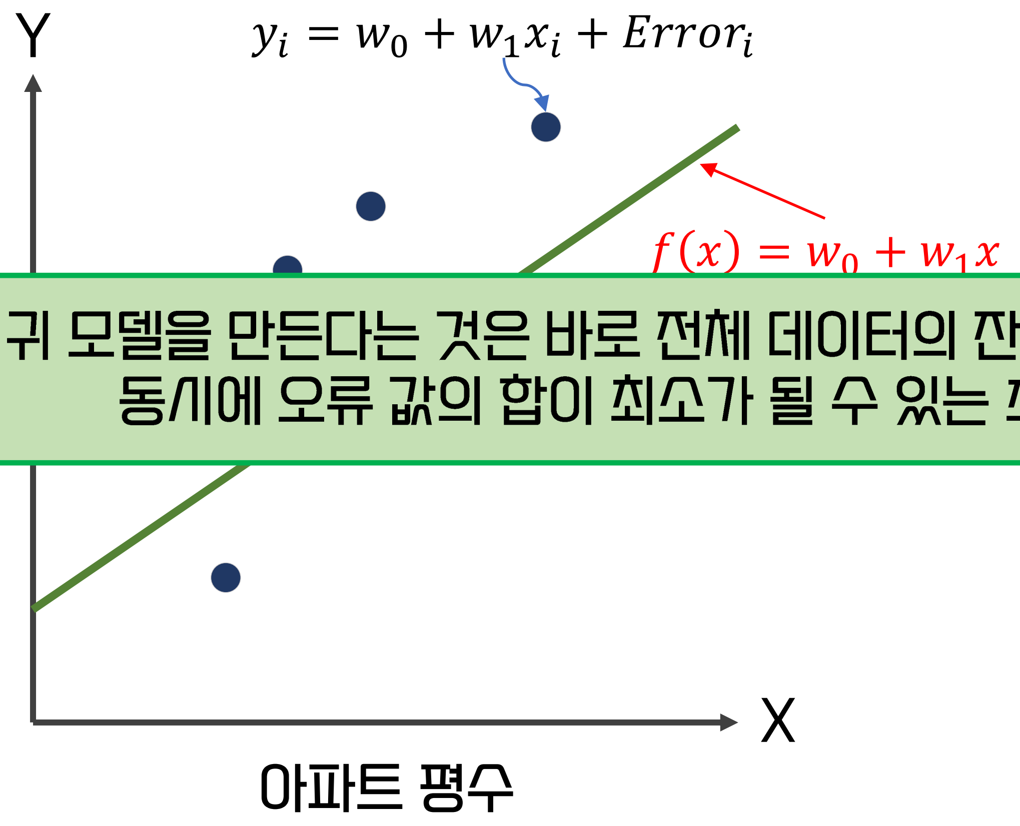
- 주택 가격이 주택의 크기로만 결정 되는 단순 선형 회귀로 가정하면 다음과 같이 주택 가격은 주택 크기에 대해 선형(직선 형태)의 관계로 표현이 가능하다.

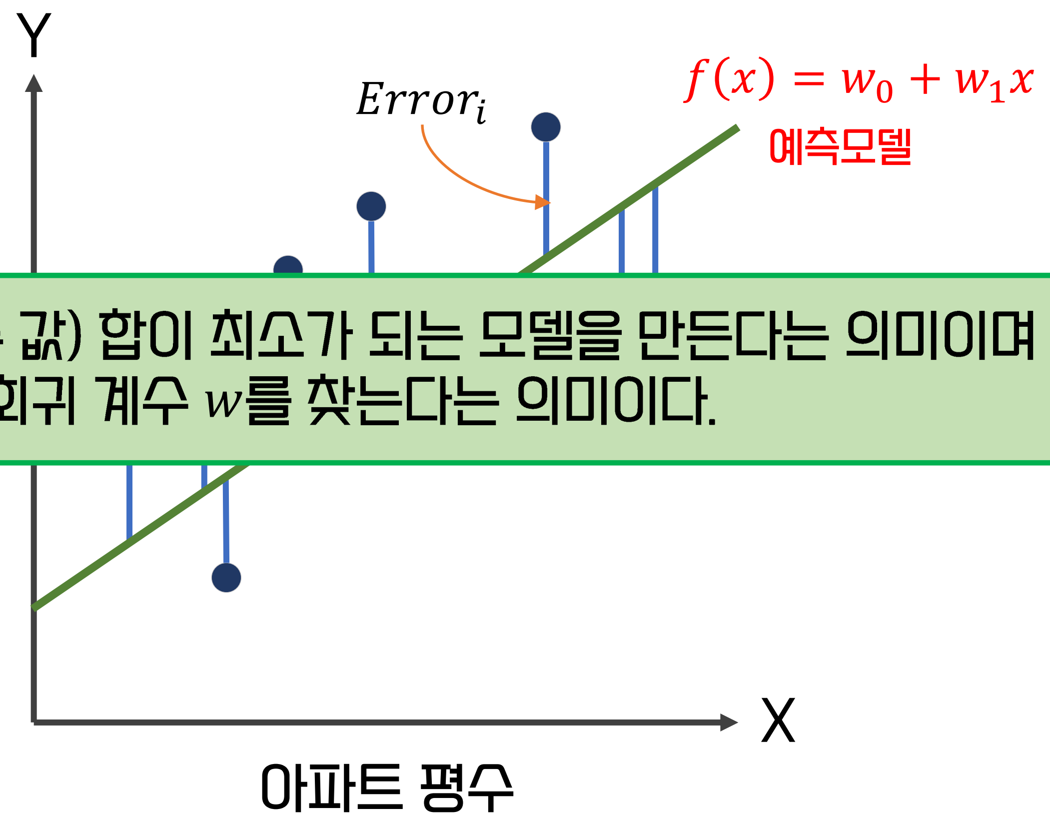




단순 선형 회귀(Simple Regression)를 통한 회귀의 이해

- 주택 가격이 주택의 크기로만 결정 되는 단순 선형 회귀로 가정하면 다음과 같이 주택 가격은 주택 크기에 대해 선형(직선 형태)의 관계로 표현이 가능하다.



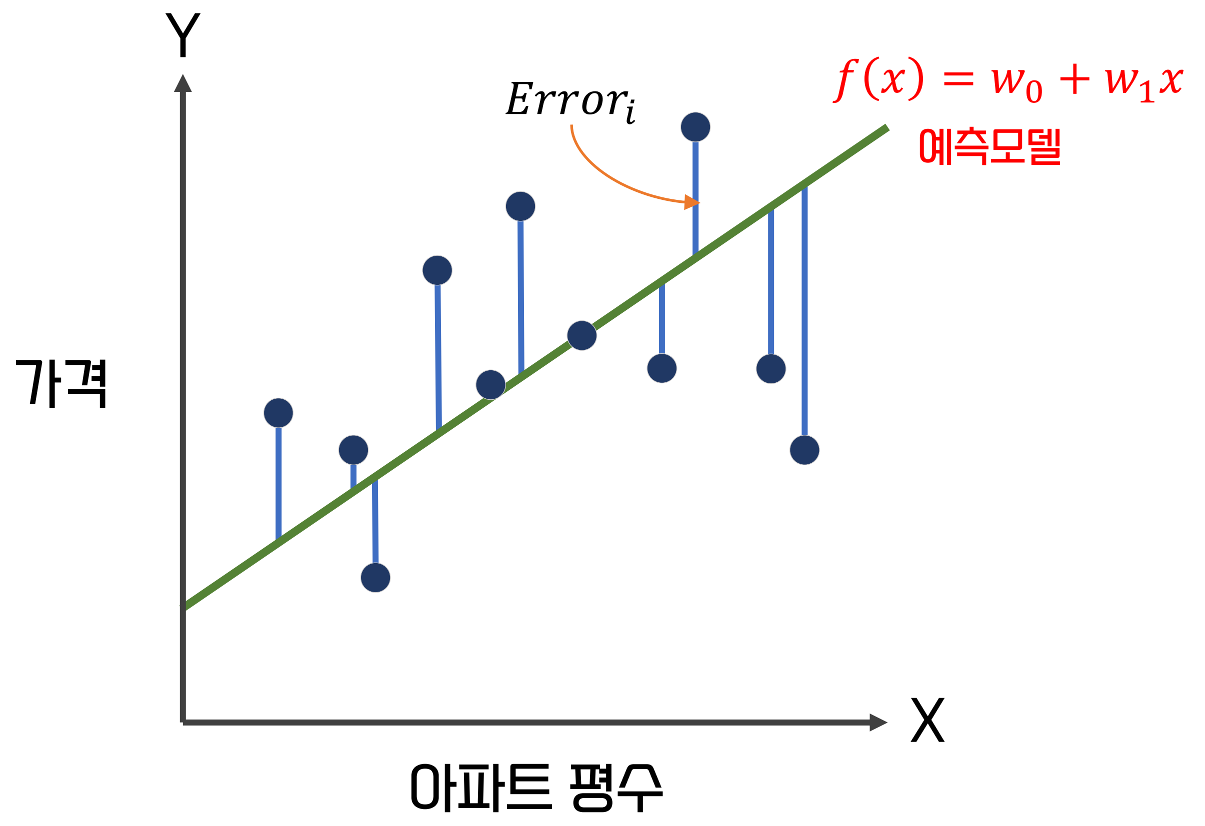


최적의 회귀 모델을 만든다는 것은 바로 전체 데이터의 잔차(오류 값) 합이 최소가 되는 모델을 만든다는 의미이며 동시에 오류 값의 합이 최소가 될 수 있는 최적의 회귀 계수 w를 찾는다는 의미이다.

RSS와 경사하강법

RSS 기반의 회귀 오류 측정

- RSS : 각 데이터 포인트의 오류 값($Error_i$)의 제곱을 구해서 더하는 방식. 일반적으로 미분 등의 계산을 편리하게 하기 위해서 RSS 방식으로 오류 합을 구한다. 즉, $\sum Error^2 = RSS$ 이다.



RSS의 이해

- RSS는 이제 변수가 w_0, w_1인 식으로 표현할 수 있으며, 이 RSS를 최소로 하는 회귀 계수(w_0, w_1)를 학습을 통해서 찾는 것이 머신러닝 기반 회귀의 핵심 사항
- RSS는 회귀식의 독립변수 X, 종속변수 Y가 중심 변수가 아니라 회귀 계수 w가 중심 변수임을 인지하는 것이 매우 중요!
  - 학습 데이터로 입력되는 독립변수와 종속변수는 RSS에서 모두 상수로 간주
- 일반적으로 RSS는 학습 데이터의 건수로 나누어서 다음과 같이 정규화된 식으로 표현

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

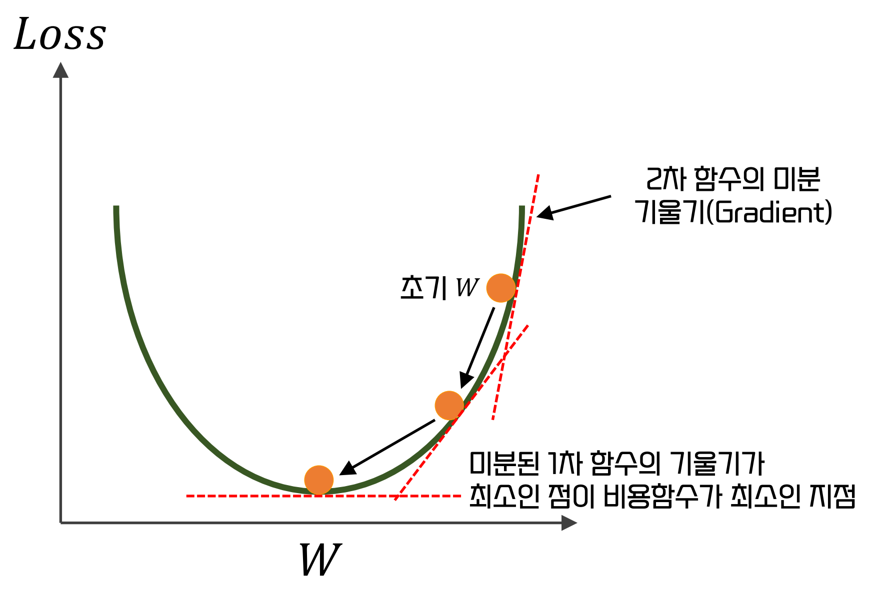
RSS – 회귀의 비용 함수

- 회귀에서 RSS는 비용(Cost)이며 w 변수(회귀 계수)로 구성되는 RSS를 비용 함수(Cost Function)라 한다.
- 머신러닝 회귀 알고리즘은 데이터를 계속 학습하면서 이 비용 함수가 반환하는 값(오류 값)을 지속해서 감소
- 최종적으로는 더 이상 감소하지 않는 최소의 오류 값을 구하는 것이 목적.
- 비용 함수를 ★손실함수(Loss Function)★라고도 함

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

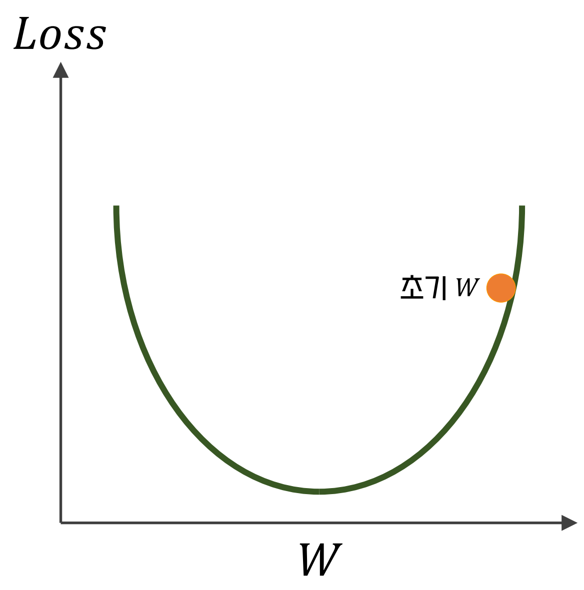
비용 최소화 하기. 경사 하강법(Gradient Descent)

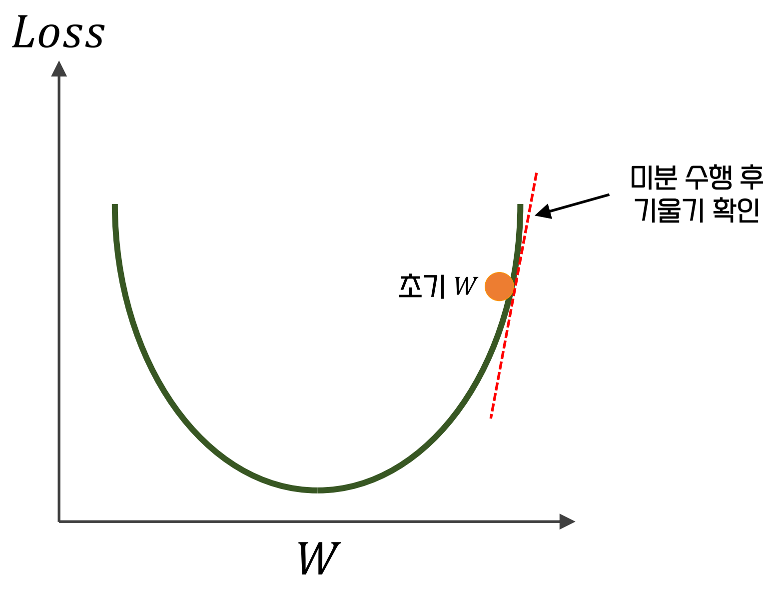
- W 파라미터의 개수가 적다면 고차원 방정식으로 비용 함수가 최소가 되는 w 변숫값을 도출할 수 있다.
- 하지만 w 파라미터가 많다면 고차원 방정식을 동원하더라도 해결하기가 힘들다.
    - w가 2개라면 이차방정식, 3개라면 삼차방정식… 100개면? 백차방정식이 된다.
- 경사 하강법은 이러한 고차원 방정식에 대한 문제를 해결해주면서 비용 함수 RSS를 최소화하는 방법을 직관적으로 제공하는 뛰어난 방식이다.

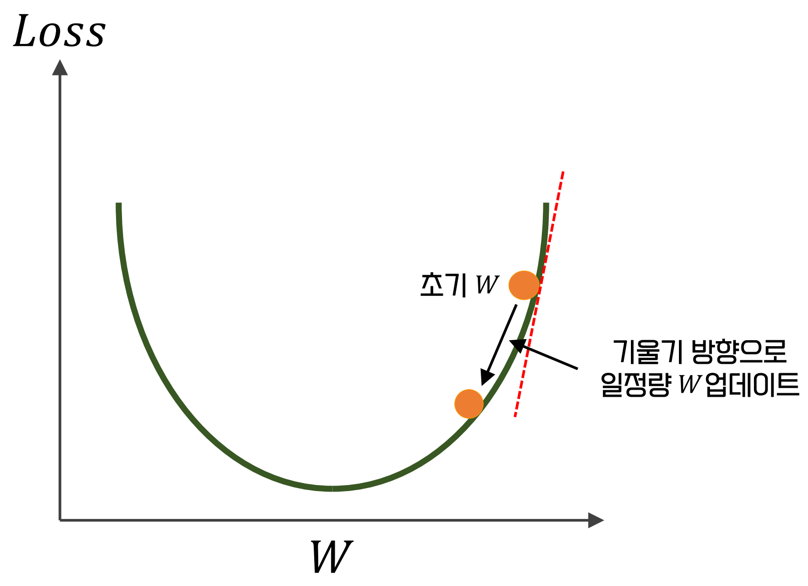


비용 최소화 하기. 경사 하강법(Gradient Descent)

- 경사 하강법의 사전적 의미는 '점진적 하강' 으로 '점진적으로' 반복적인 계산을 통해 W 파라미터 값을 업데이터 하면서 오류 값이 최소가 되는 W 파라미터를 구하는 방식이다.







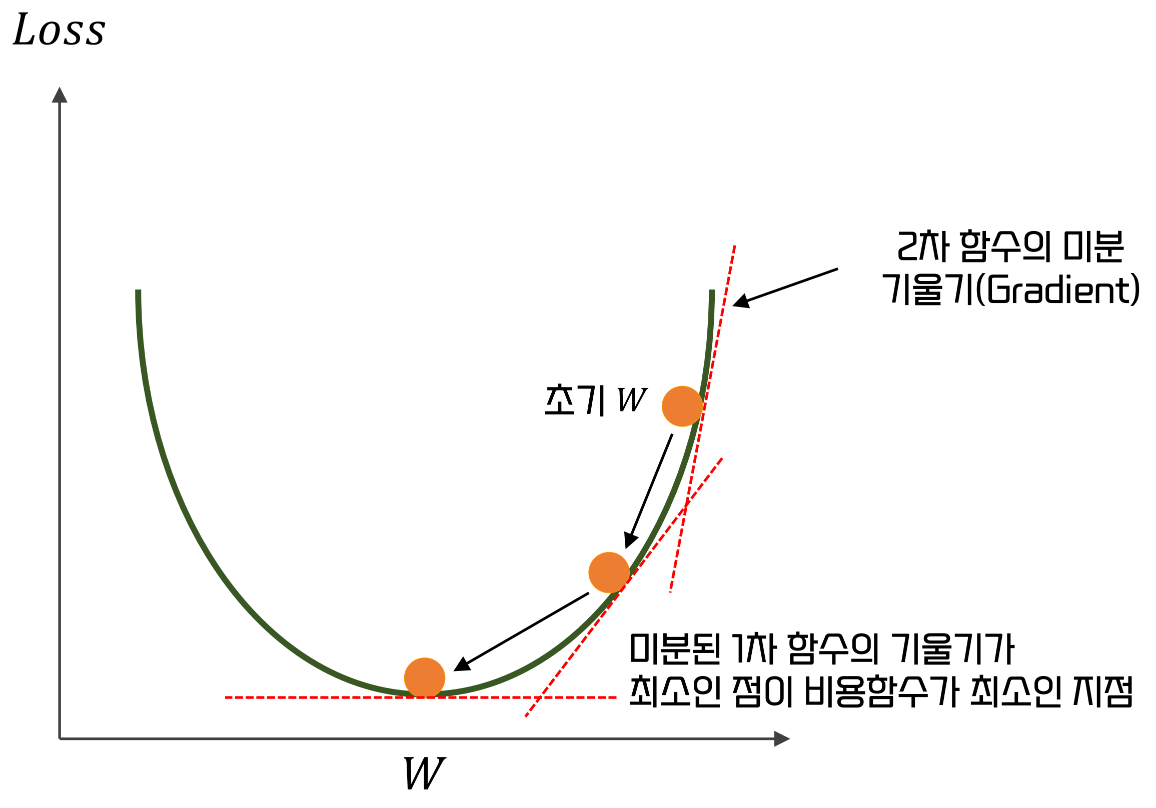
비용 최소화 하기. 경사 하강법(Gradient Descent)

- 경사 하강법은 반복적으로 비용 함수의 반환 값, 즉 예측값과 실제값의 차이가 작아지는 방향성을 가지고 W 파라미터를 지속해서 보정해 나간다.
- 최소 오류 값이 100이었다면 두 번째 오류 값은 100보다 작은 90, 세 번째는 80과 같은 방식으로 지속해서 오류를 감소시키는 방향으로 W 파라미터 값을 계속 업데이트 해 나간다.
- 오류 값이 더 이상 작아지지 않으면 그 오류 값을 최소 비용으로 판단하고 그 때의 W 파라미터를 최적 파라미터로 반환

미분을 통한 비용 함수의 최소값 찾기

- 어떻게 하면 오류가 작아지는 방향으로 W 값을 보정할 수 있을까?

비용 함수가 다음 그림과 같은 포물선 형태의 2차함수라면 경사 하강법은 최초 W 에서부터 미분을 적용한 뒤 이 미분 값이 계속 감소하는 방향으로 순차적으로 W 를 업데이트 한다.
마침내 더 이상 미분 된 1차 함수의 기울기가 감소하지 않는 지점을 비용 함수가 최소인 지점으로 간주하고 그 때의 W 를 반환한다.



LinearRegression

LinearRegression 클래스

```
sklearn.linear_model.LinearRegression(fit_intercept=True, normalize=False)
```

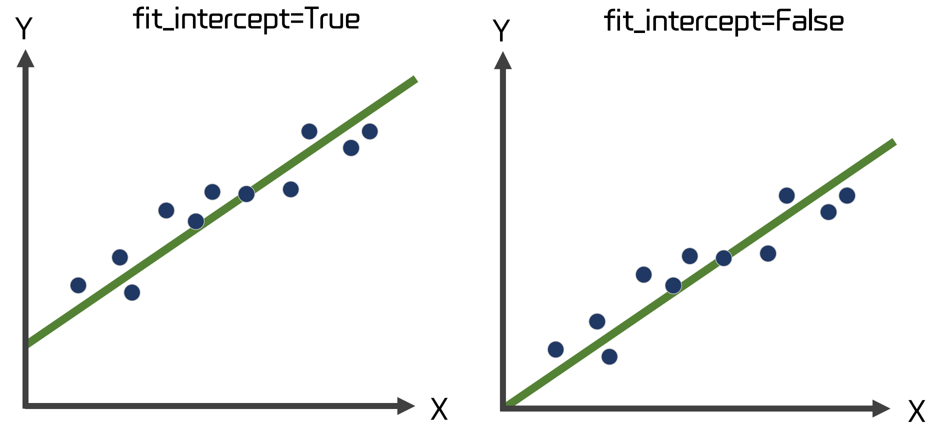
- LinearRegression 클래스는 예측값과 실제값의 RSS를 최소화하는 OLS(Ordinary Least Squares) 추정 방식으로 구현한 클래스임

- LinearRegression 클래스는 훈련(fit) 시에 회귀 계수(Coefficients)인 W를 coef_ 속성에 저장함.
  - 단, 절편(bias, intercept)는 intercept_ 속성에 저장됨

LinearRegression 클래스

sklearn.linear_model.LinearRegression(fit_intercept=True, normalize=False)

- fit_intercept
  - 절편 값을 계산할 것인지 여부 결정. False로 설정 시 intercept가 0으로 지정됨



- normalize
  - fit_intercept가 False인 경우엔 무시됨.
  - True로 설정하면 회귀를 수행 하기 전 입력 데이터 세트를 정규화(Standard Scaling)함.
  - 대체로 사용하지 않는 것을 추천

선형 회귀의 다중 공선성(multi-collinearity) 문제

- 일반적으로 선형 회귀는 입력 Feature의 독립성에 많은 영향을 받음.
- Feature간의 상관관계가 매우 높은 경우 분산이 매우 커져서 오류에 매우 민감해지는 현상을 다중공선성
- 일반적으로 상관관계가 높은 Feature가 많은 경우 독립적인 중요한 Feature만 남기고 제거하거나 규제를 적용한다.

- 예시
  - 평수(25평, 34평), 제곱 미터(59m^2, 84m^2)
  - 자동차 무게와 연비

사이킷런 LinearRegression 클래스

파이썬 Wrapper	사이킷런 Wrapper	하이퍼 파라미터 설명
lambda	reg_lambda	L2 규제(regularization) 적용 값. 기본값은 1로서 값이 클 수록 규제 값이 커진다. 과적합 제어
alpha	reg_alpha	L1 규제(regularization) 적용 값. 기본값은 0으로서 값이 클 수록 규제 값이 커진다. 과적합 제어
early_stopping_round	early_stopping_rounds	학습 조기 종료를 위한 early stopping interval 값
num_leaves	num_leaves	최대 리프 노드 개수
min_sum_hessian_in_leaf	min_child_weight	결정트리의 min_child_leaf와 유사. 과적합 조절용

num_leaves의 개수를 중심으로 min_child_samples(min_data_in_leaf), max_depth를 함께 조정하면서 모델의 복잡도를 줄이는 것이 기본 튜닝 방안

회귀 평가 지표

회귀 평가 지표

평가 지표	설명	수식
MAE	Mean Absolute Error(MAE)이며 실제 값과 예측값의 차이를 절댓값으로 변환해 평균	$MAE = \frac{1}{N}\sum_{i=1}^{N}\lvert y_i - \hat{y}_i\rvert$
MSE ★	Mean Squared Error(MSE)이며 실제 값과 예측값의 차이를 제곱해 평균	$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$
MSLE	MSE에 Log를 적용한 값으로 결정값이 클 수록 오류값도 커지기 때문에 일부 큰 오류값들로 인해 전체 오류값이 커지는 것을 막아준다.	$MSLE = \frac{1}{N}\sum_{i=1}^{N}(\log(y_i + 1) - \log(\hat{y}_i + 1))^2$
RMSE ★	MSE 값은 오류의 제곱을 구하므로 실제 오류 평균보다 더 커지는 특성이 있으므로 MSE에 루트를 씌운 것이 RMSE(Root Mean Squared Error)이다.	$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$
RMSLE	RMSE에 로그를 적용한 것으로서 결정값이 클 수록 오류값도 커지기 때문에 일부 큰 오류값들로 인해 전체 오류값이 커지는 것을 막아준다.	$RMSLE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\log(y_i + 1) - \log(\hat{y}_i + 1))^2}$
R^2 ★	분산 기반으로 예측 성능을 평가한다. 실제 값의 분산 대비 예측값의 분산 비율을 지표로 하며, 1에 가까울수록 예측 정확도가 높아지게 된다	$R^2 = \frac{\text{예측값 } variance}{\text{실제 값 } variance}$

MAE vs RMSE

- MAE에 비해 RMSE는 큰 오류에 상대적인 패널티를 더 부여함

- 예를 들어 다섯 개의 오류값(실제값과 예측값의 차)이 10, 20, 10, 10, 100과 같이 다른 값에 비해 큰 오류값이 존재하는 경우 RMSE는 전반적으로 MAE보다 높다.

$$MAE=(10+20+10+10+100)/5 = 30$$

$$RMSE = \sqrt{\frac{100+400+100+100+10000}{5}} = \sqrt{2140} = 46.26$$

회귀 평가 API

- 평가 방법에 대한 사이킷런의 API 및 cross_val_score나 GridSearchCV에서 평가 시 사용되는 scoring 파라미터의 적용 값

평가 지표	사이킷 런 평가지표 API	Scoring 함수 적용 값
MAE	metrics.mean_absolute_error	'neg_mean_absoulte_error'
MSE ★	metrics.mean_squared_error	'neg_mean_squared_error'
MSLE	metrics.mean_squared_error를 사용하되 sqaured 파라미터를 False로 설정	'neg_root_mean_squared_error'
RMSE ★	metrics.mean_squared_log_error	'neg_mean_squared_log_error'
R^2 ★	metrics.r2_score	'r2'

Scoring 함수에 회귀 평가 적용 시 유의 사항

cross_val_score, GridSearchCV와 같은 Scoring 함수에 회귀 평가 지표 적용 시 유의 사항

- MAE의 사이킷런 scoring 파라미터 값은 'neg_mean_absoulte_error'이다. 이는 Negative(음수) 값을 가진다는 의미인데, MAE는 절댓값의 합이기 때문에 음수가 될 수 없다.

- Scoring 함수에 'neg_mean_absolute_error'를 적용해 음수값을 반환하는 이유는 사이킷런의 Scoring 함수가 Score값이 클수록 좋은 평가 결과로 자동 평가 하기 때문이다. 따라서 원래의 평가 지표 값에 -1을 곱해 음수(Negative)를 만들어 작은 오류 값이 더 큰 숫자로 인식되게 하는 것이다.

- metrics.mean_absolute_error()와 같은 사이킷런 평가 지표 API는 정상적으로 양수의 값을 반환한다. 하지만 Scoring 함수의 scoring 파라미터 값 'neg_mean_absolute_error'가 의미 하는 것은 -1 * metrics.mean_absolute_error()이다.

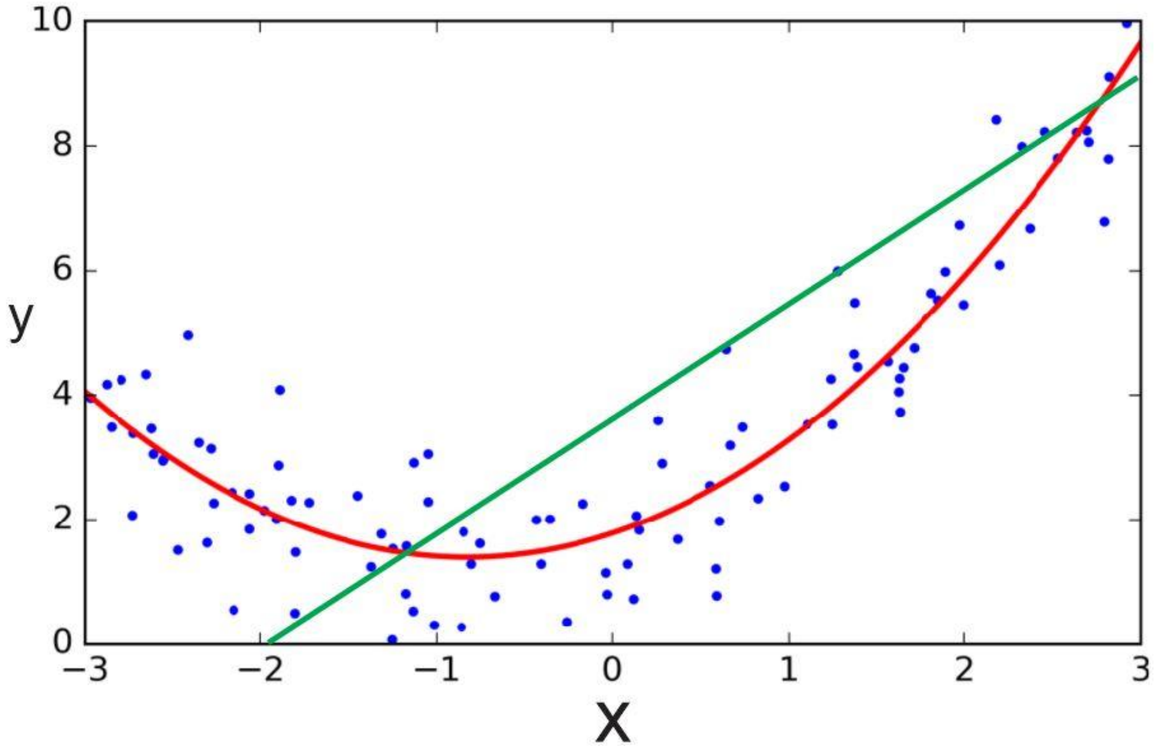
실습

LinearRegression을 이용한 보스턴 주택가격 예측

다항회귀

다항 회귀(Polynormial Regression) 개요

다항 회귀는 $y = w_0 + w_1x_1 + w_2x_2 + w_3x_1x_2 + w_4x_1^2 + w_5x_2^2$과 같이 회귀식이 독립변수의 단항식이 아닌 2차, 3차 방정식과 같은 형태로 표현되는 것을 지칭한다.



데이터 세트에 대해서 Feature에 대해 Target 값의 관계를 단순 선형 회귀 직선으로 표현한 것 보다 다항 회귀 곡선으로 표현한 것이 더 예측 성능이 높다.

선형 회귀와 비선형 회귀의 구분

- 선형 회귀
  - $y = w_0 + w_1x_1 + w_2x_2 + w_3x_1x_2 + w_4x_1^2 + w_5x_2^2$ 에서 만약 새로운 변수인 Z를 $Z = [x_1,\ x_2,\ x_1x_2,\ x_1^2,\ x_2^2]$로 정의한다면 $y = w_0 + w_1z_1 + w_2z_2 + w_3z_3 + w_4z_4 + w_5z_5$ 로 표현되기 때문에 다항 회귀가 곡선으로 표현된다고 해서 비선형 회귀인 것은 아니다.
  - 즉 다항 회귀는 선형 회귀 이며 회귀에서 선형 회귀 / 비선형 회귀를 나누는 기준은 회귀 계수가 선형 / 비선형인지에 따른 것이지 독립 변수의 선형 / 비선형 여부와는 무관하다.

- 비선형 회귀
  - $y = w_1 \cos(x + w_4) + w_2 \cos(2x + w_4) + w_3$ cos 함수를 이용해 각각의 계수 w가 묶여 있기 때문에 비선형 회귀이다.

사이킷런에서의 다항회귀

- 사이킷런에는 다항회귀 API가 존재하지는 않기 때문에 PolynomialFeatures 클래스를 이용해 원본 단항 피처를 단항 피처로 변환해 LinearRegression을 적용시켜야 한다.

- 단항 피처 $[x_1, x_2]$의 차수(degree)를 2차 다항 피처로 변환시키면 $(x_1 + x_2)^2$의 식 전개에 대응되는 $[1, x_1, x_2, x_1x_2, x_1^2, x_2^2]$의 다항 피처로 변환된다.
    - 1차 단항 피처가 $[x_1, x_2] = [0, 1]$이라면 2차 다항 피처는 $[1, x_1 = 0, x_2 = 1, x_1x_2 = 0, x_1^2 = 0, x_2^2 = 1]$ 형태인 $[1, 0, 1, 0, 0, 1]$이 된다.

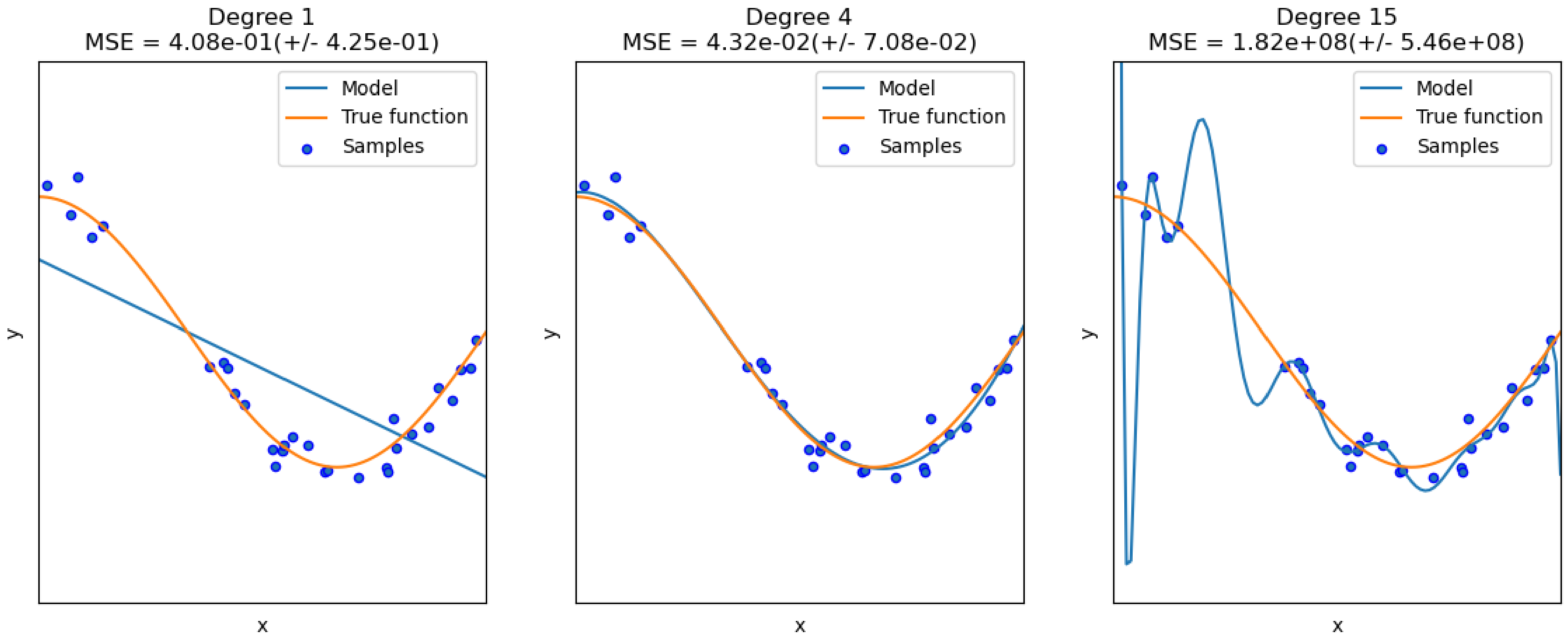
실습

다항회귀를 이용한 보스턴 주택가격 예측

다항회귀를 이용한 과소적합, 과대적합 이해하기

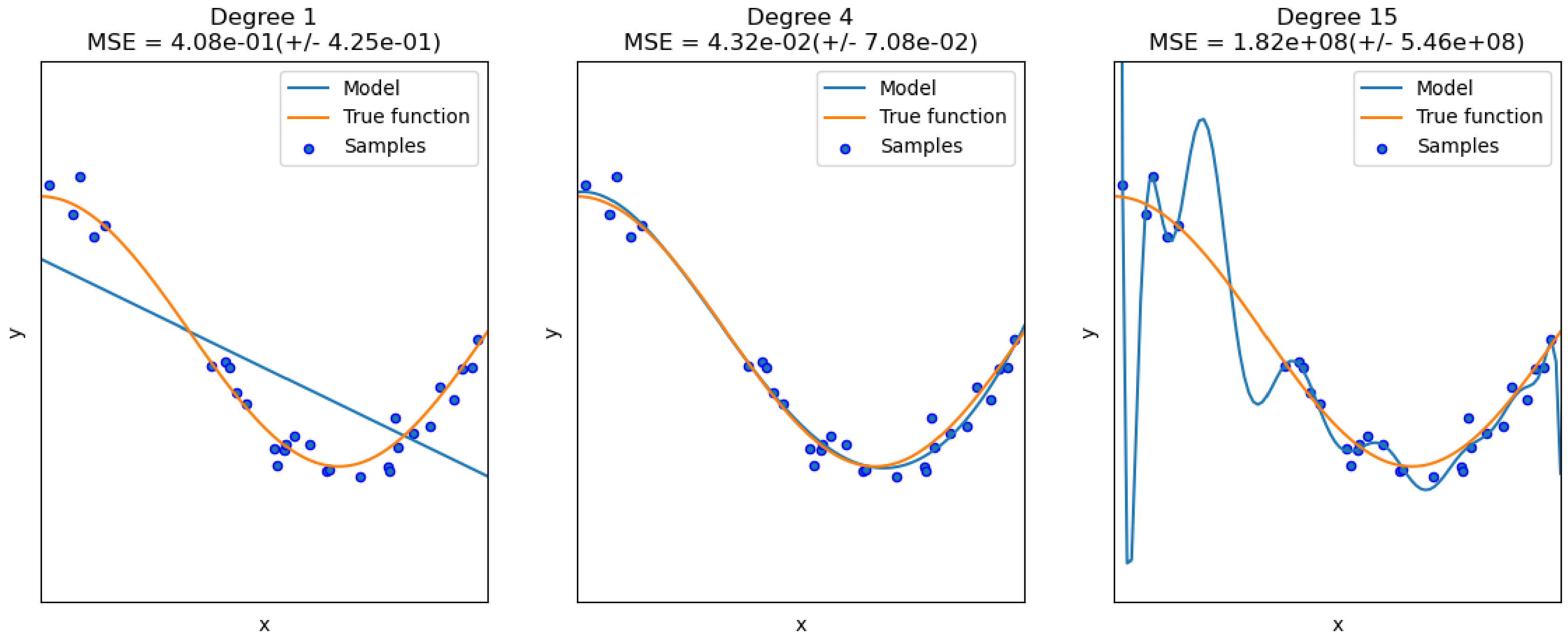
다항회귀를 이용한 과대적합, 과소적합 확인

- Degree 1 그림은 방향성만 고려하고, 실제 데이터를 고려하지는 못함.
- Degree 4 그림은 데이터의 방향성과 실제 데이터를 적절히 고려함
- Degree 15 그림은 모델의 변동성이 매우 심하다. 즉 데이터를 너무 과하게 해석함



다항회귀를 이용한 과대적합, 과소적합 확인

- Degree 1 그림은 방향성만 고려하고, 실제 데이터를 고려하지는 못함. (과소적합)
- Degree 4 그림은 데이터의 방향성과 실제 데이터를 적절히 고려함. (일반화)
- Degree 15 그림은 모델의 변동성이 매우 심하다. 즉 데이터를 너무 과하게 해석함. (과대적합)

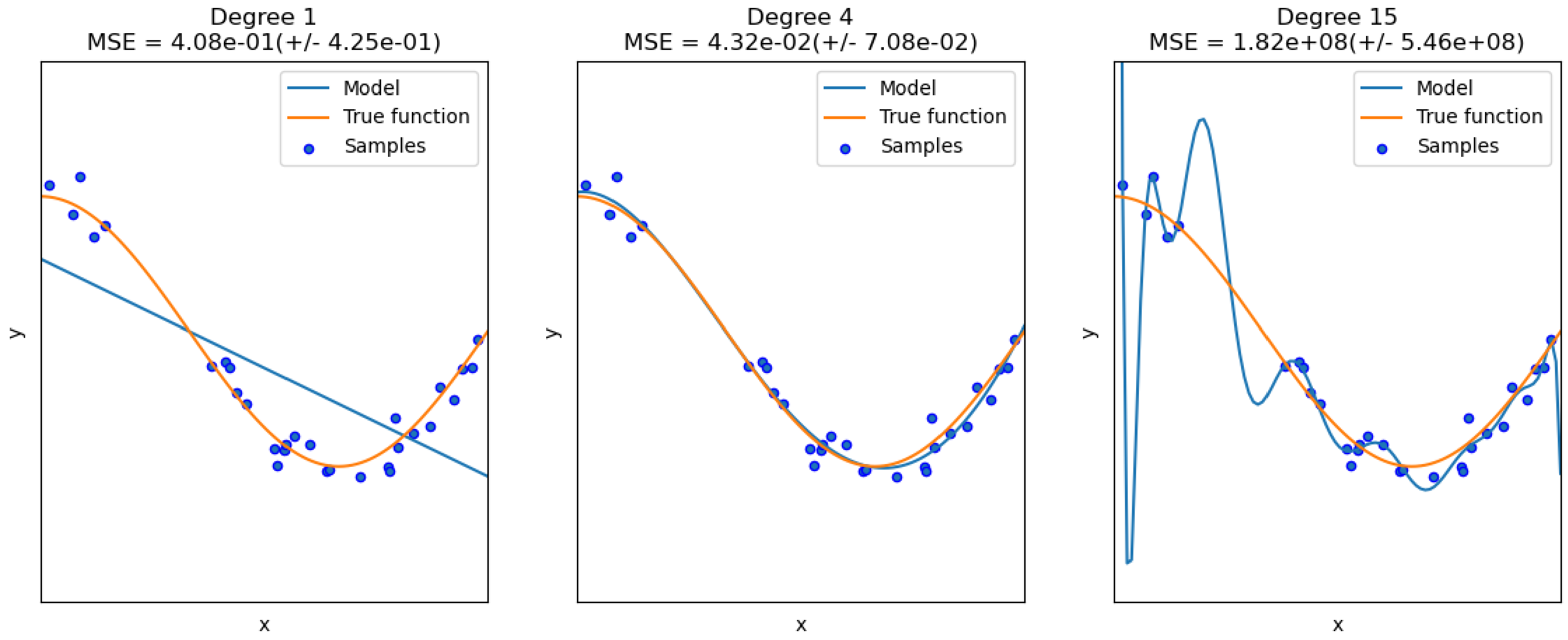


실습

차수에 따른 과소, 과대적합 확인해 보기

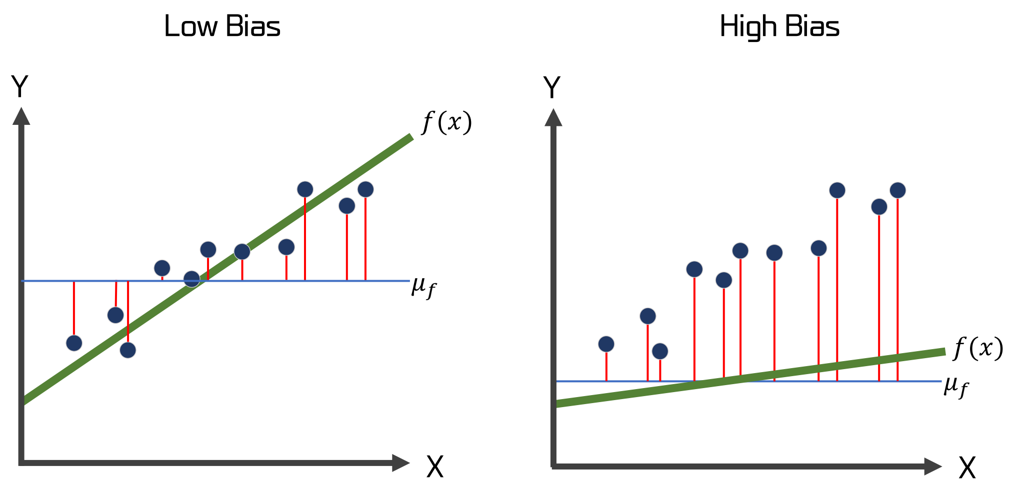
편향-분산 트레이드 오프(Bias-Variance Trade off)

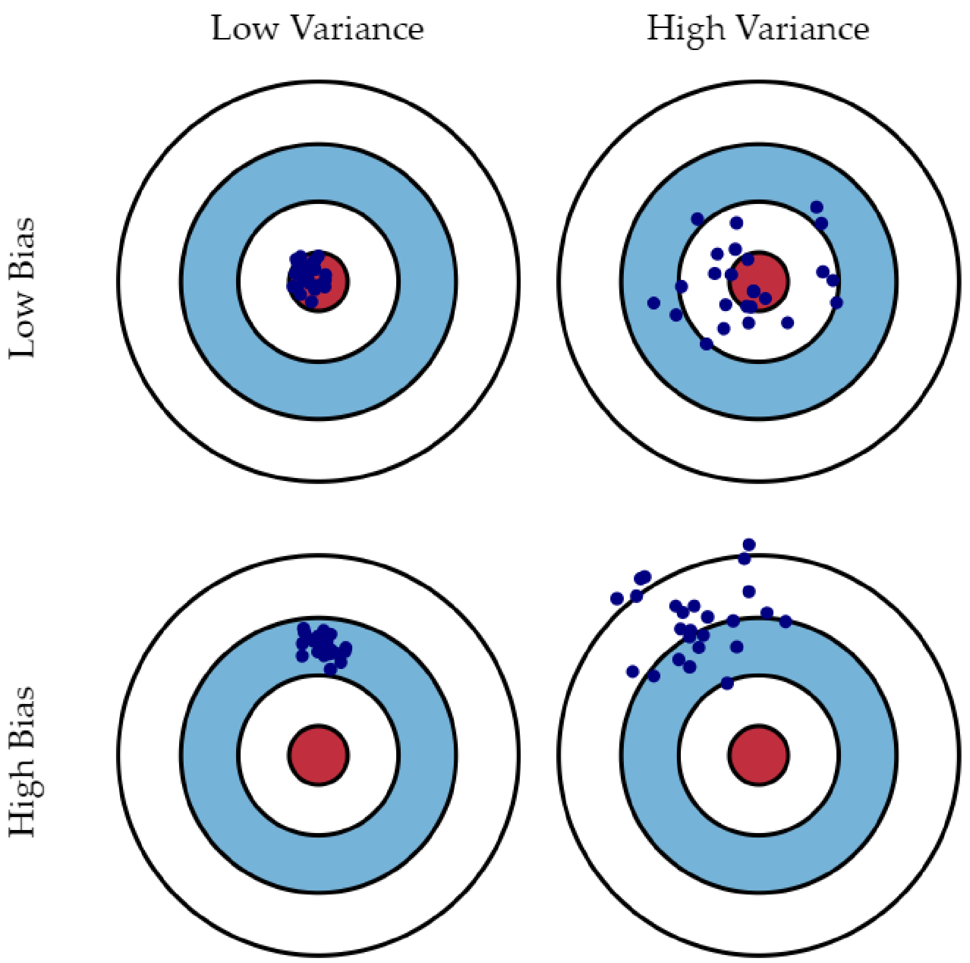
- 과소적합은 문제를 굉장히 단순하게 보기 때문에 약간만 문제가 복잡해 져도 해결하지 못한다.
- 과대적합은 단순한 문제를 고차원으로 풀기 때문에 훈련 데이터에 대해서는 잘 맞을지 모르나, 테스트 데이터에 대한 성능은 매우 떨어질 수 있다.



편향-분산 트레이드 오프(Bias-Variance Trade off) ★

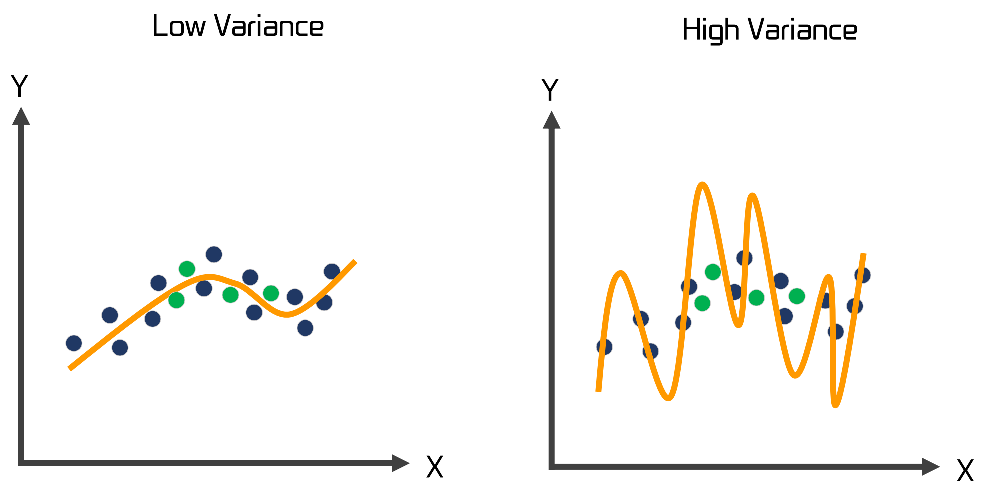
- Bias(편향)는 훈련 데이터에 대한 방향성, 예측이 정확하게 방향성을 잘 잡고 가고 있는가를 의미한다.
- 훈련 데이터와 모델 예측의 차이. 즉 에러를 의미한다.

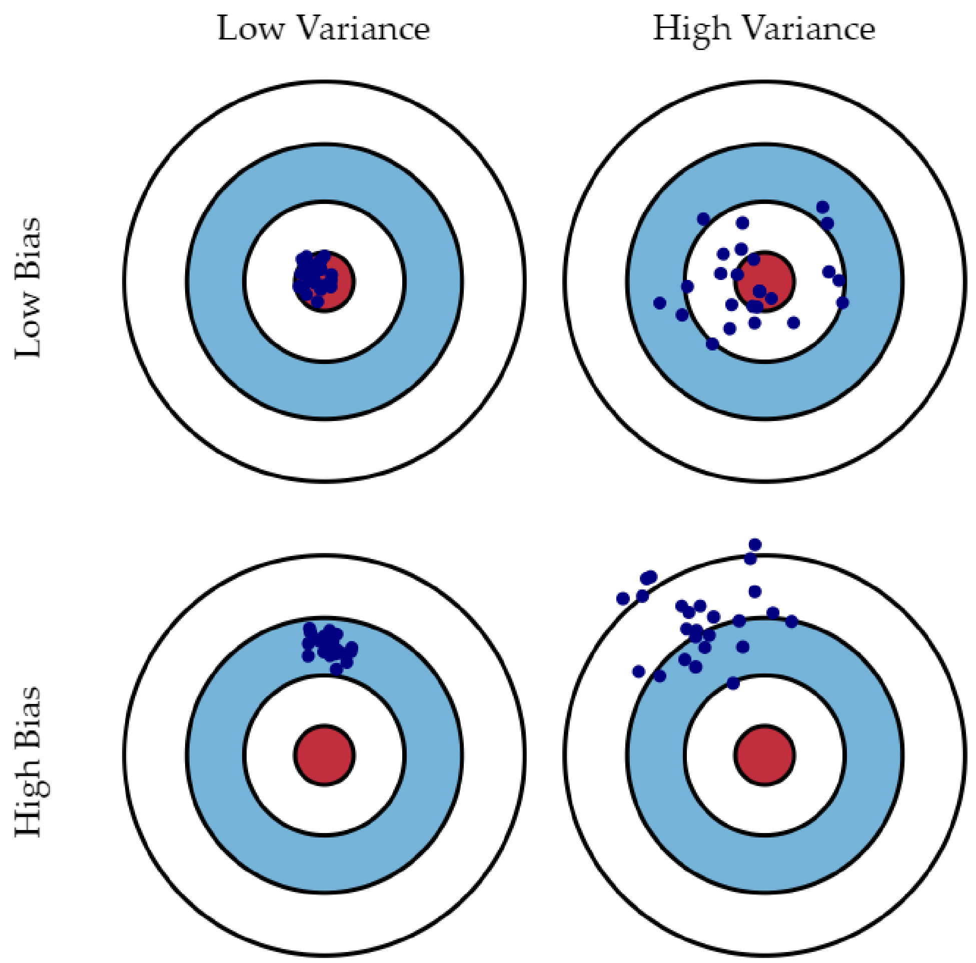




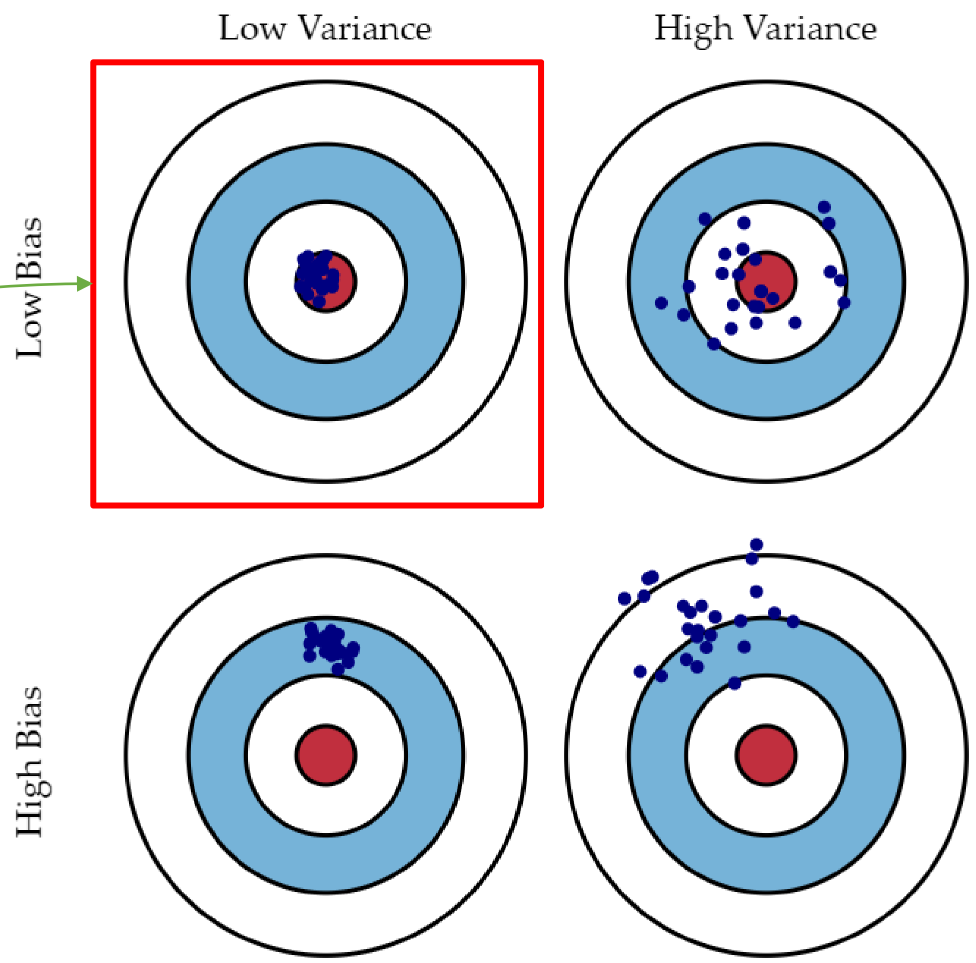
편향-분산 트레이드 오프(Bias-Variance Trade off) ★

- Variance(분산)은 테스트 데이터를 예측 할 때 마다 초점에서 얼마만큼 벗어나는가를 의미한다.
- 즉 훈련 데이터 예측에 대한 분산이 커지면, 테스트 세트의 예측이 잘 안된다.



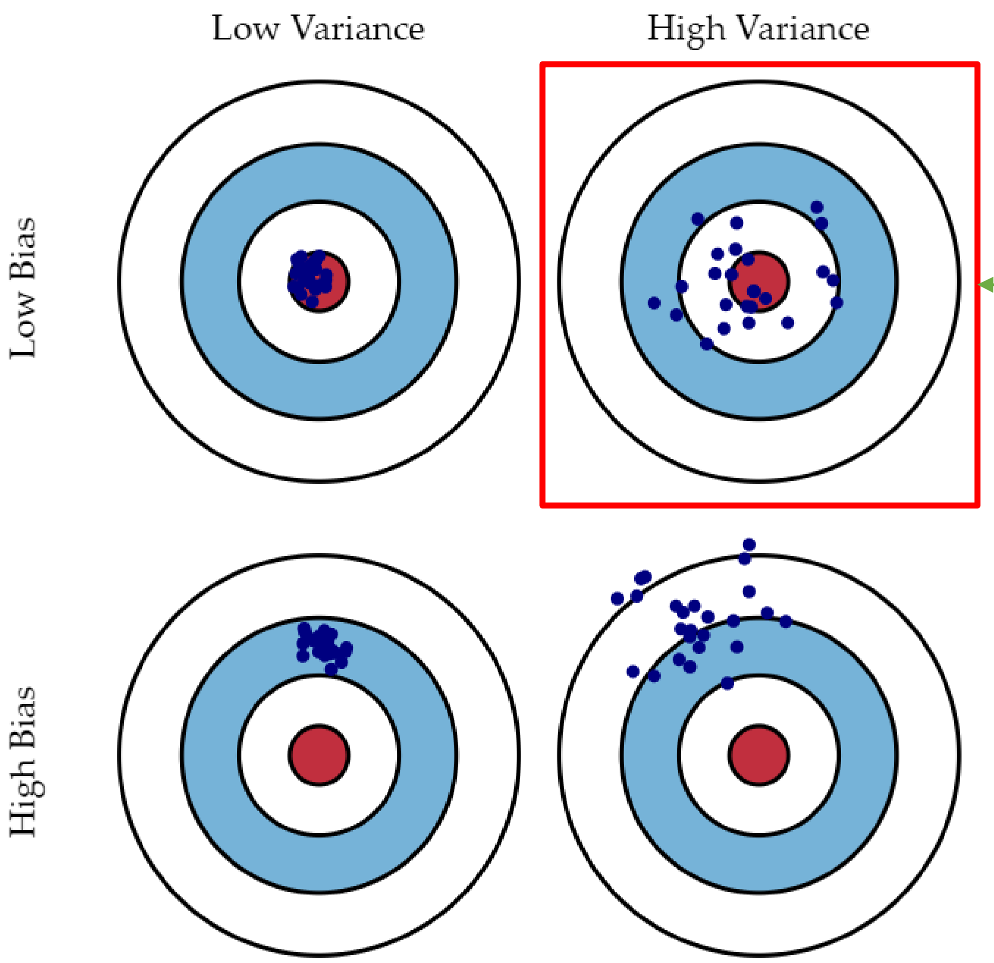


편향-분산 트레이드 오프(Bias-Variance Trade off) ★



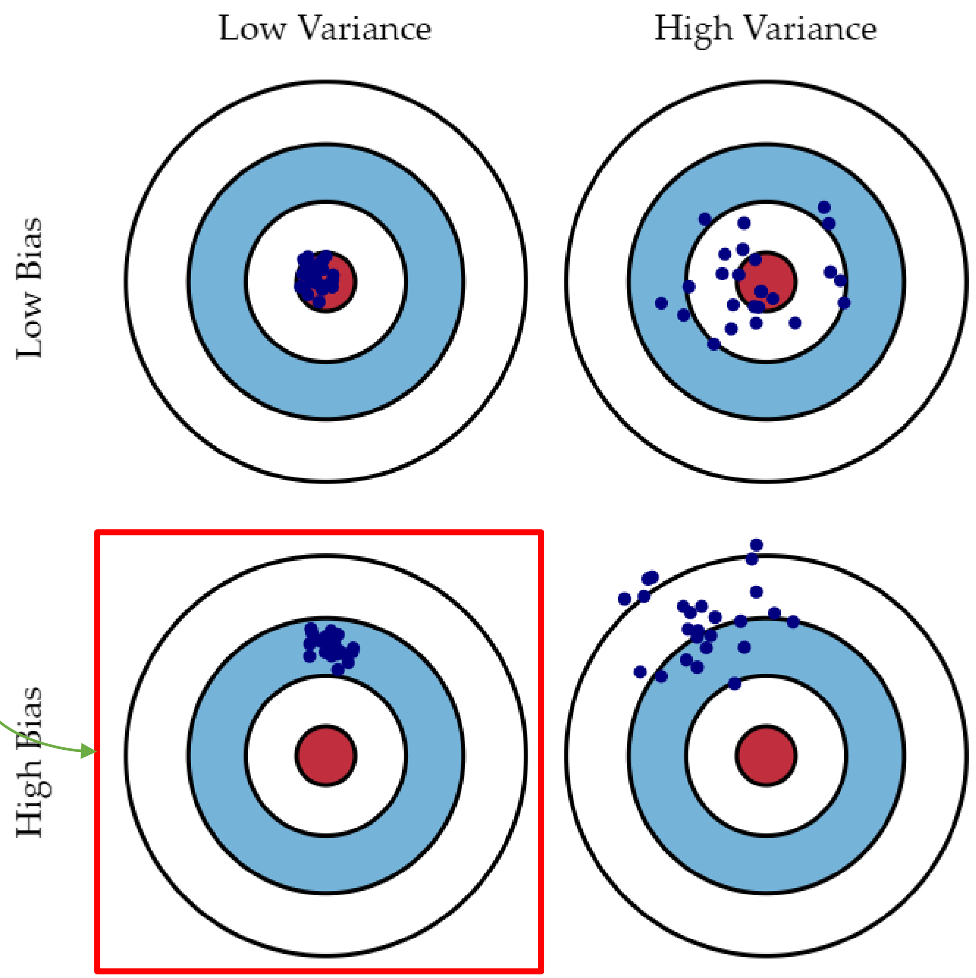
Best인 상황. 방향성과 예측하고자 하는 초점이 정확하게 맞는다.

편향-분산 트레이드 오프(Bias-Variance Trade off) ★



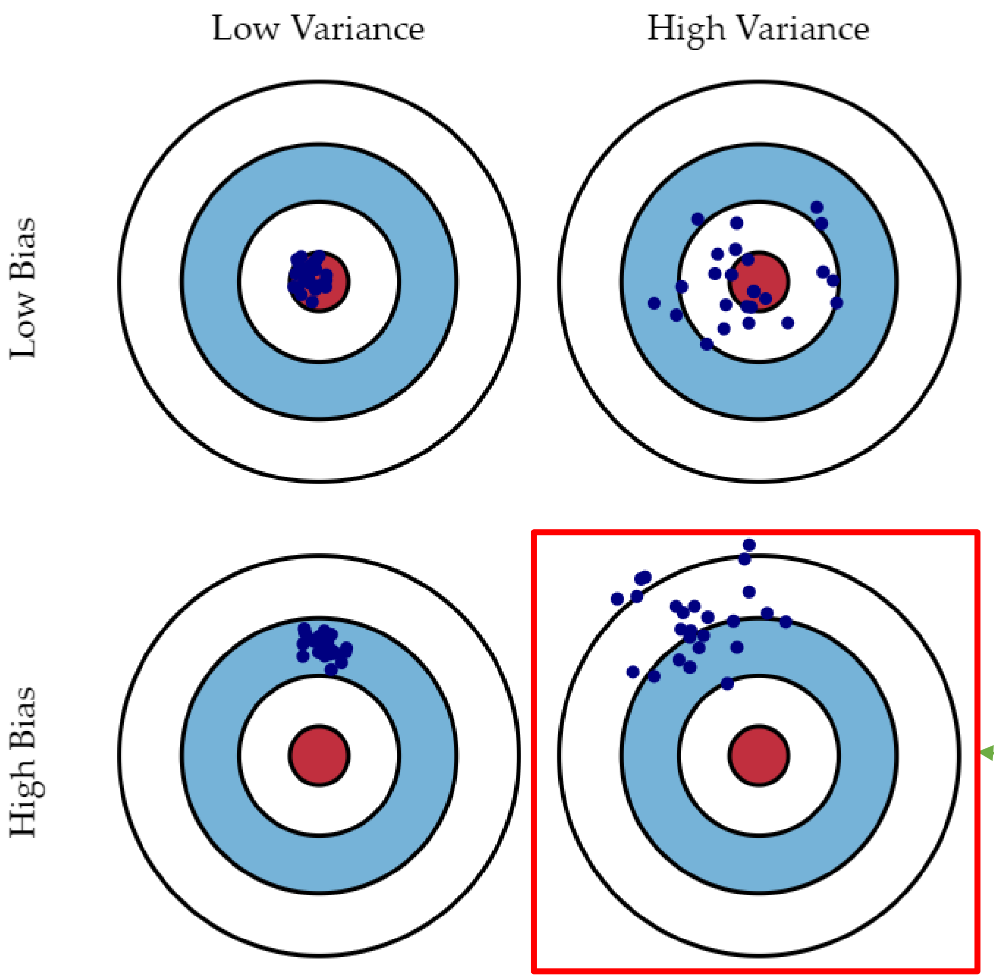
머신러닝 모델을 만들었을 때 많이 만나볼 수 있는 상황이다. 방향성은 괜찮지만 예측 시 이리저리 산만하게 분산되는 것을 알 수 있다. 이 상태가 바로 과대적합 상태이다..

편향-분산 트레이드 오프(Bias-Variance Trade off) ★



예측 방향이 잘못 됐으나, 테스트 할 때마다 비슷하게 값이 모여 있는 경우이다. 예측 전 세팅이 잘못 되어 있는 경우이다

편향-분산 트레이드 오프(Bias-Variance Trade off)

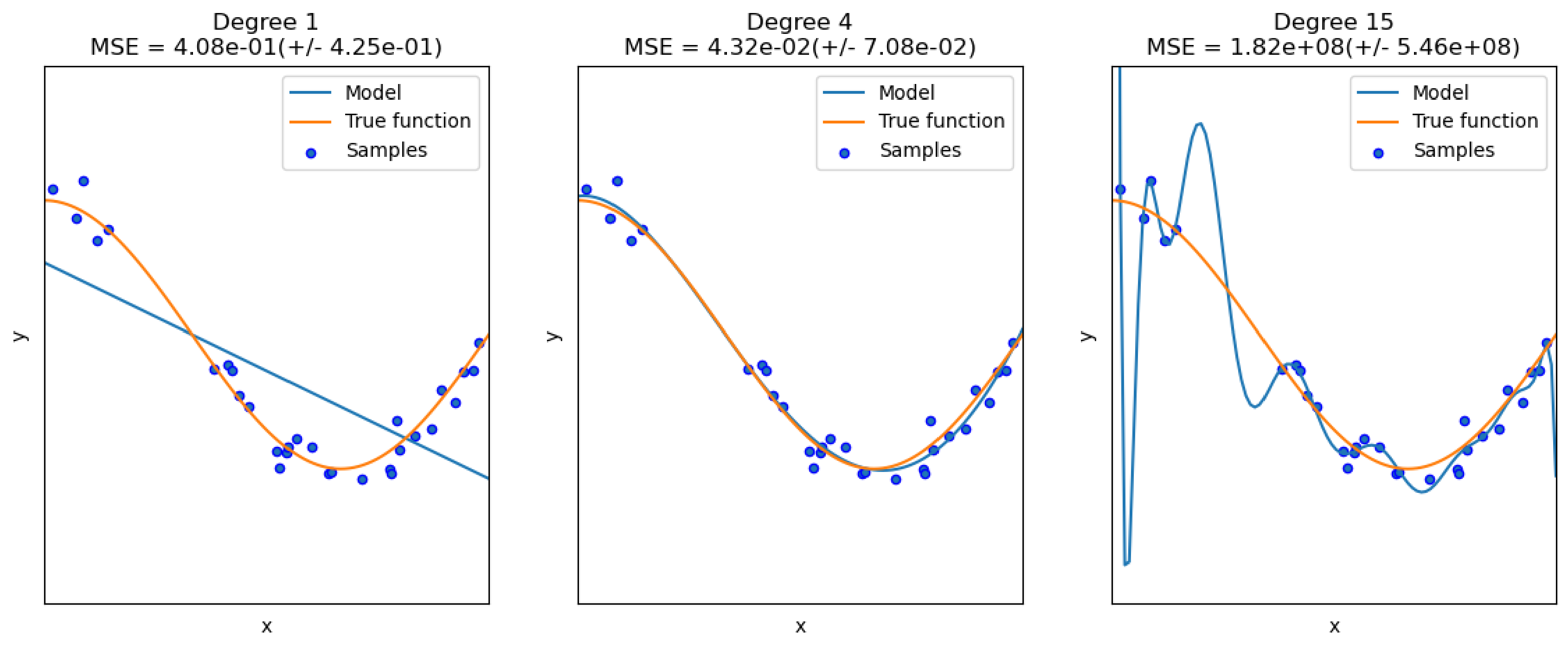


방향성, 예측의 초점이 모두 엉망이다.

즉 데이터를 하나도 반영하지 못하는 과소적합 상태라고 볼 수 있다.

편향-분산 트레이드 오프(Bias-Variance Trade off)

- 일반적으로 편향이 높으면 분산은 낮아지고
- 편향이 낮아지면 분산이 높은 경향이 많다.

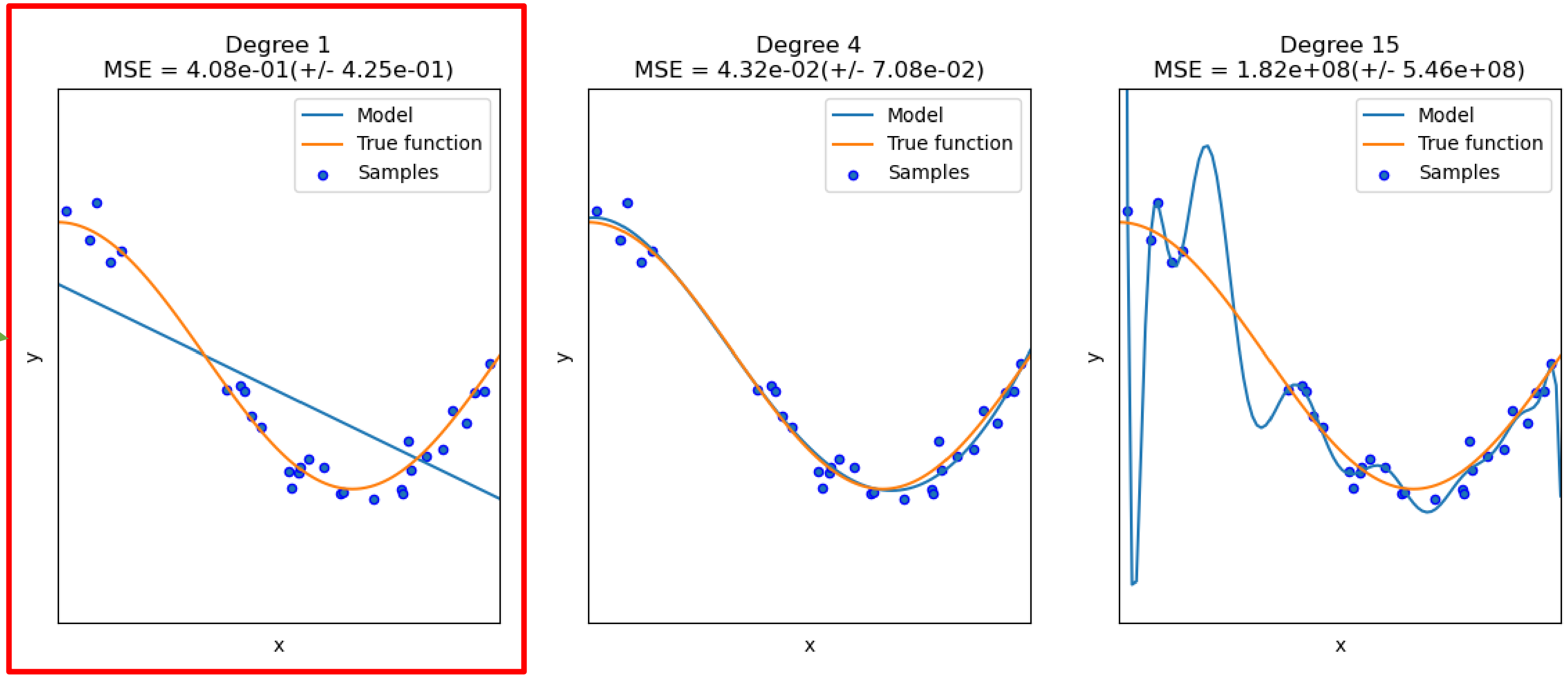


편향-분산 트레이드 오프(Bias-Variance Trade off) ★

- 일반적으로 편향이 높으면 분산은 낮아지고
- 편향이 낮아지면 분산이 높은 경향이 많다.

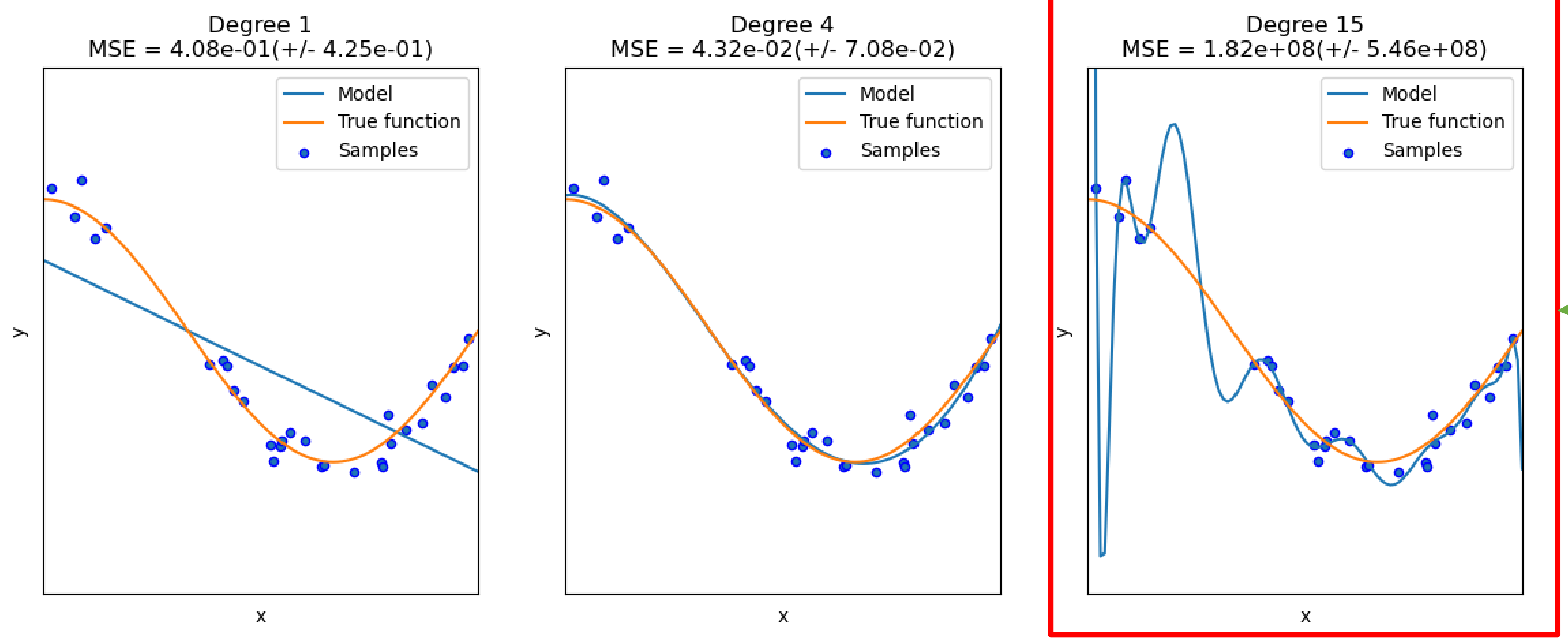
1차원 직선으로
예측했기 때문에
예측의 분산이 낮다.

하지만 편향이 높다.



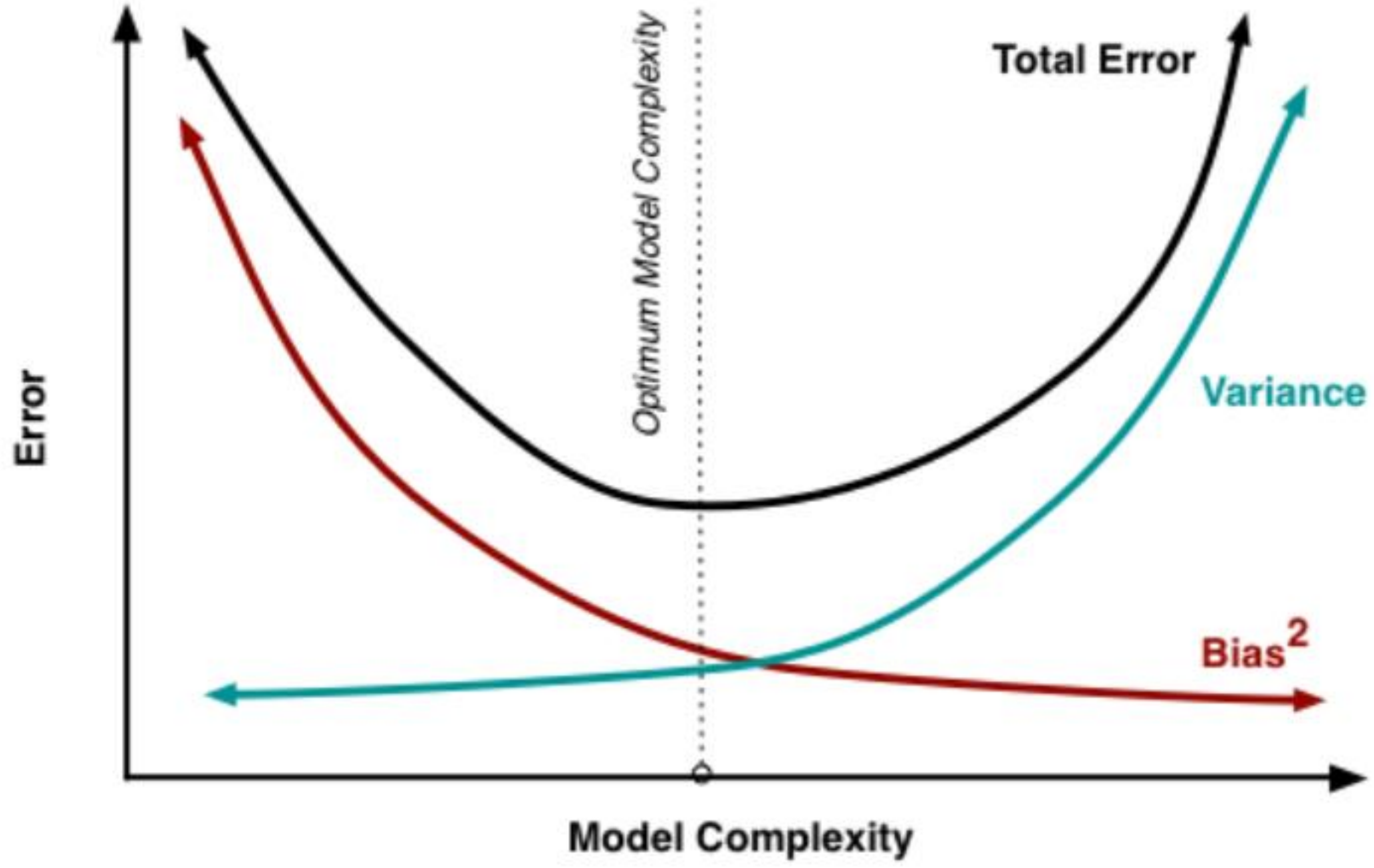
편향-분산 트레이드 오프(Bias-Variance Trade off) ★

- 일반적으로 편향이 높으면 분산은 낮아지고
- 편향이 낮아지면 분산이 높은 경향이 많다.



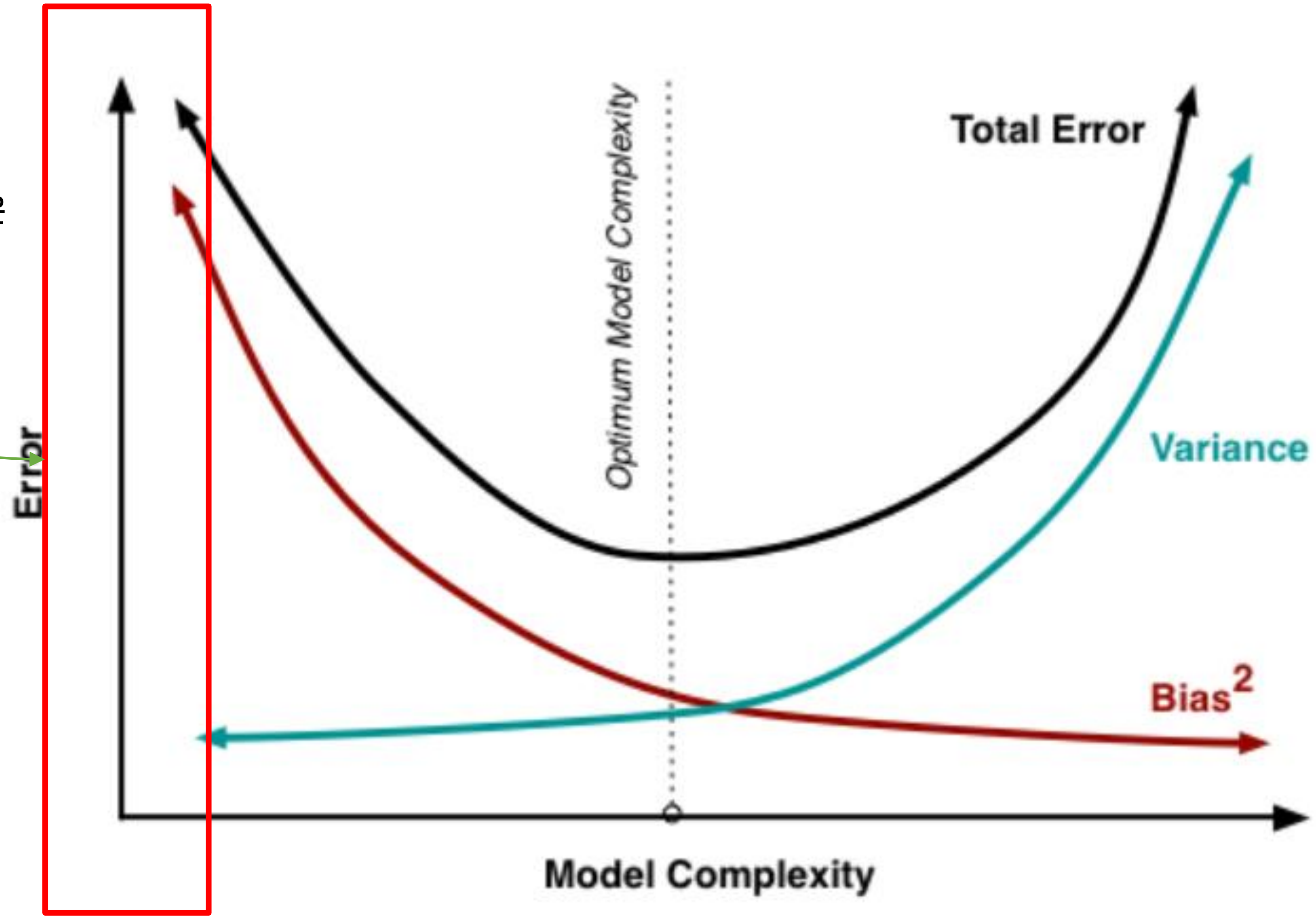
모든 데이터 포인트를
예측을 하였기 때문에
편향은 낮으나 모델
예측의 분산은 높다

모델 복잡도에 따른 편향-분산 트레이드 오프



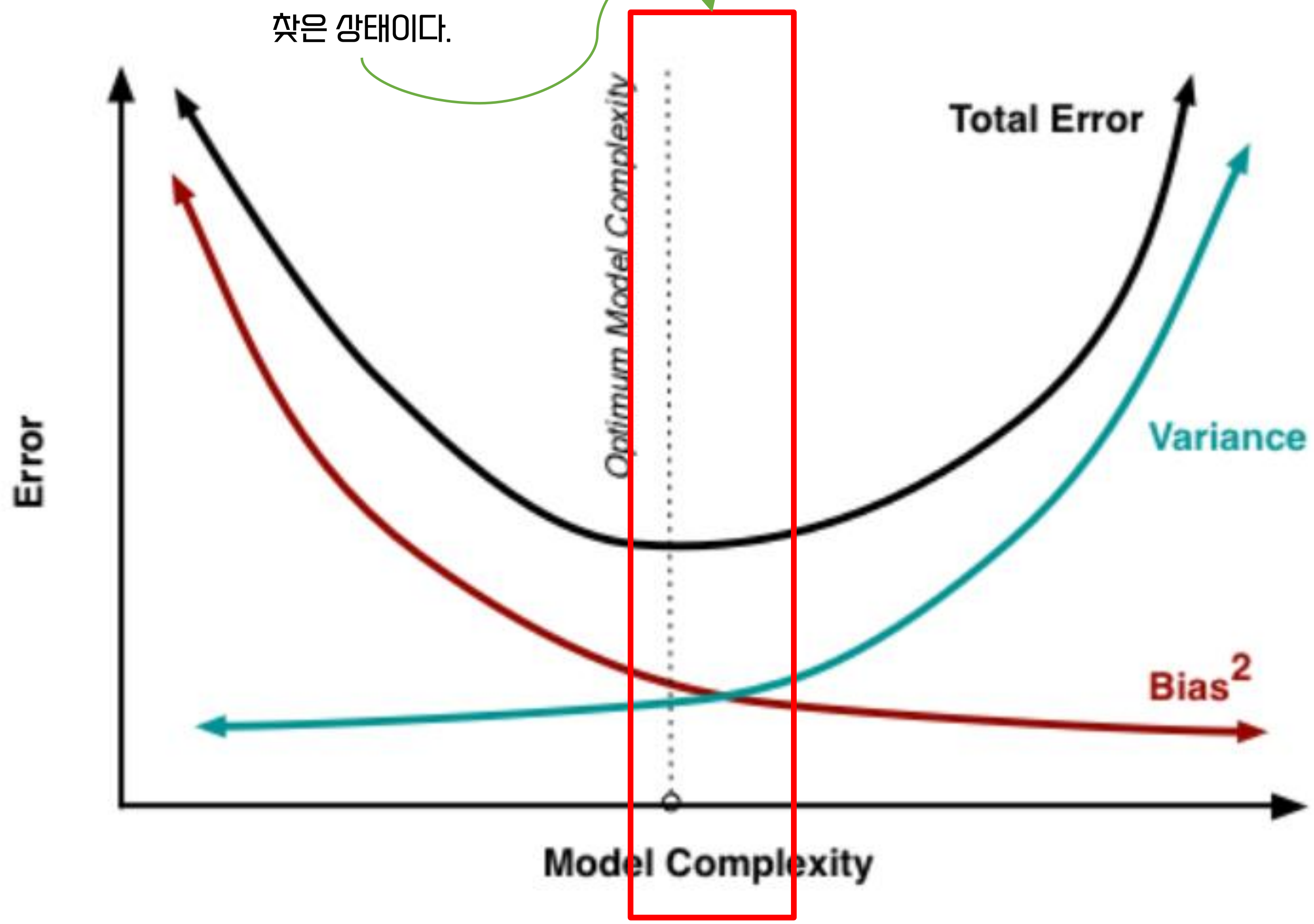
모델 복잡도에 따른 편향-분산 트레이드 오프

모델의 복잡도가 낮은 상태.
모델이 훈련 데이터를 잘 파악하지 못하고 있는 상태이다.
편향이 크고(방향성이 맞지 않고) 분산은 낮은(모델은 단순한) 과소적합 상태를 뜻한다.

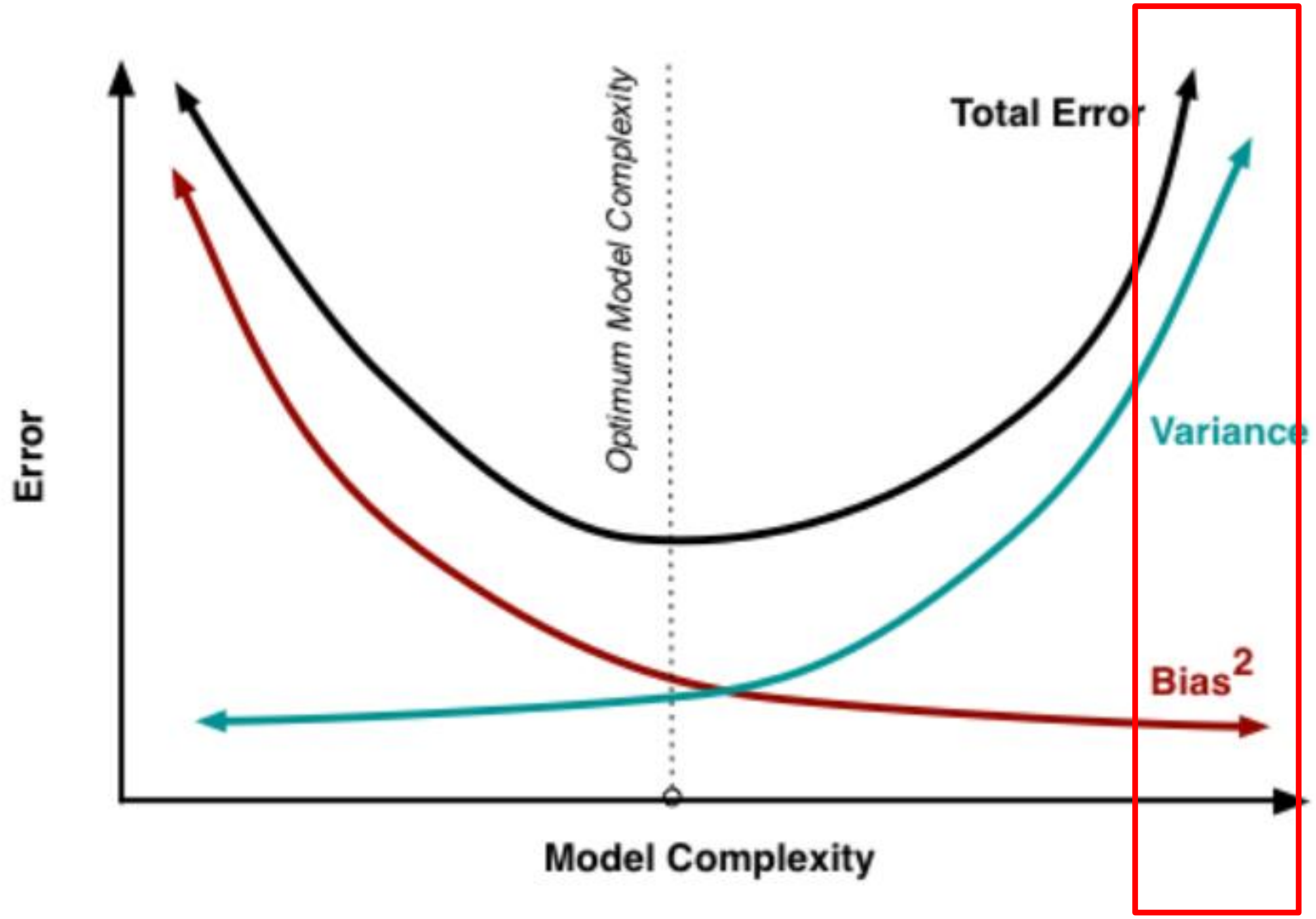


모델 복잡도에 따른 편향-분산 트레이드 오프

모델이 최적의 편향과 분산을
찾은 상태이다.

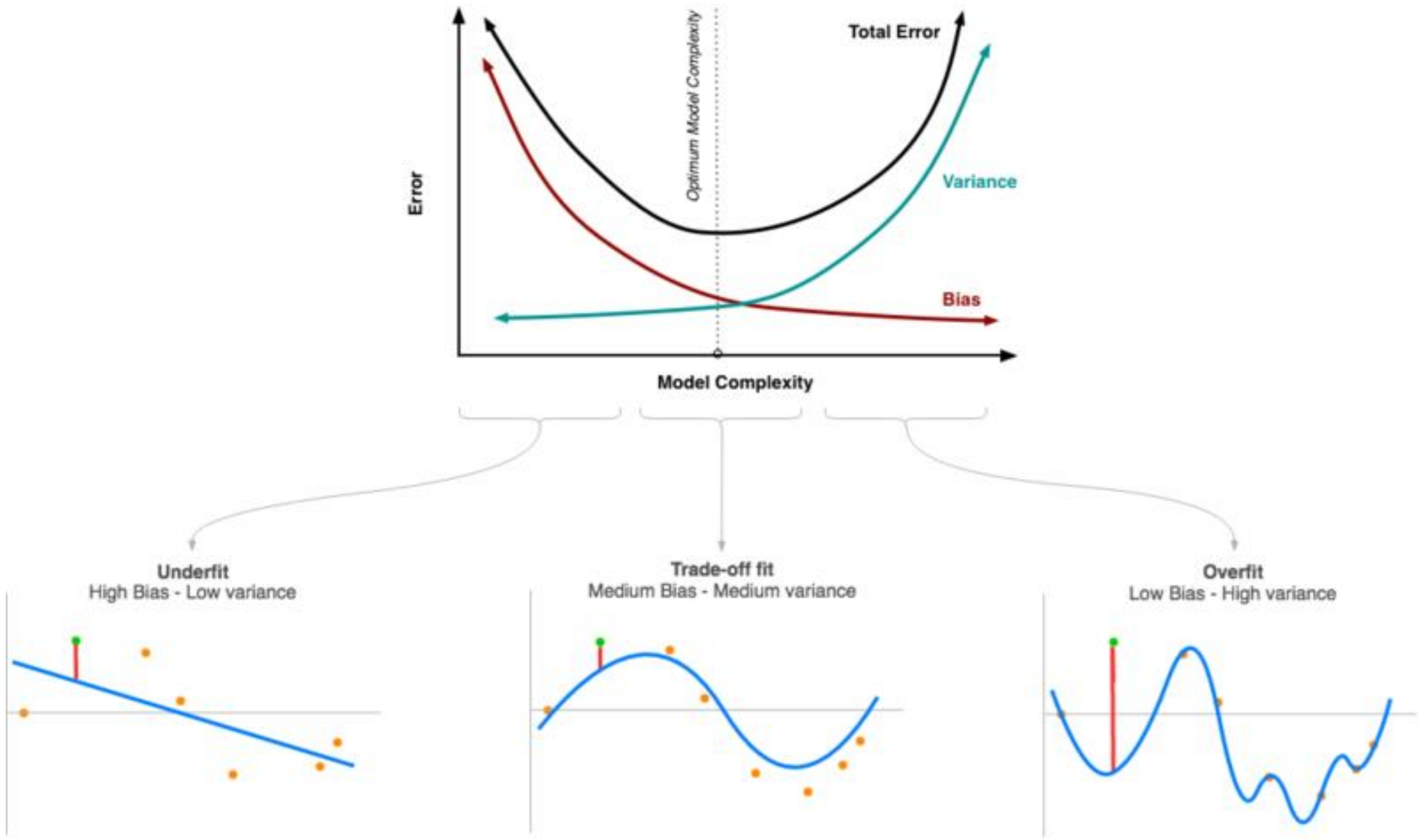


모델 복잡도에 따른 편향-분산 트레이드 오프



모델이 훈련 데이터를 과하게 해석한 상태이다.
편향은 낮으나(방향성은 제대로)
분산이 매우 높아(모델이 매우 복잡함)
과대적합 상태를 뜻한다.

모델 복잡도에 따른 편향-분산 트레이드 오프



규제 선형 회귀

규제 선형 회귀 개요(Regularized Linear Regression)

- 앞서 본 Degree=15의 다항회귀 처럼 지나치게 모든 데이터에 적합한 회귀식을 만들기 위해서 다항식이 복잡해지고 회귀 계수가 매우 크게 설정이 되면서 과대적합이 되었고, 테스트 데이터 세트에 대해 좋지 않는 성능을 보였다.
- 따라서 회귀 모델은 적절히 데이터에 적합하면서도 회귀 계수가 기하급수적으로 커지는 것을 제어할 수 있어야 한다.
- 즉 최적의 모델이 되기 위한 손실 함수의 목표는 학습 데이터에 대한 잔차 오류도 최소화 되어야 하지만, 회귀 계수의 크기 제어도 목적이 되어야 한다.

규제 선형 회귀 개요(Regularized Linear Regression)

- 앞서 본 Degree=15의 다항회귀 처럼 지나치게 모든 데이터에 적합한 회귀식을 만들기 위해서 다항식이 복잡해지고 회귀 계수가 매우 크게 설정이 되면서 과대적합이 되었고, 테스트 데이터 세트에 대해 좋지 않는 성능을 보였다.

- 따라서 회귀 모델은 적절히 데이터에 적합하면서도 회귀 계수가 기하급수적으로 커지는 것을 제어할 수 있어야 한다.

- 즉 최적의 모델이 되기 위한 손실 함수의 목표는 학습 데이터에 대한 잔차 오류도 최소화 되어야 하지만, 회귀 계수의 크기 제어도 목적이 되어야 한다.

$$Loss(x,y) = \arg\min_{w} \sum (y-\hat{y})^2 + \alpha \sum w^2 = \arg\min_{w} \sum RSS(W) + \alpha \sum w^2$$

$$Loss(x,y) = \arg\min_{w} \sum (y-\hat{y})^2 + \alpha \sum |w| = \arg\min_{w} \sum RSS(W) + \alpha \sum |w|$$

규제 선형 모델의 alpha의 역할

- α가 0 또는 매우 작은 값이라면 손실 함수의 식은 기존과 동일한 $Loss(x, y) = \arg\min_{w} RSS(W) + 0$이 될 것이다.
- 반면에 α가 무한대 또는 매우 큰 값이라면 손실 함수 식은 $RSS(W)$에 비해 $\alpha \sum w^2$ 또는 $\alpha \sum |w|$의 값이 너무 커지게 되므로 W를 작게 만들어야 손실이 최소화되는 비용 함수 목표를 달성할 수 있게 된다.
- 즉 α값을 크게 하면 비용 함수는 회귀 계수 W의 값을 작게 해 과적합을 개선할 수 있으며, α값을 작게 하면 회귀 계수 W의 값이 커져도 어느 정도 상쇄가 가능하므로 학습 데이터 적합을 더 개선할 수 있게 된다.

- $\alpha = 0$인 경우에는 W가 커도 $\alpha \sum w^2$ 또는 $\alpha \sum |w|$이 0이 되어 비용 함수는 $\arg\min_{w} RSS(W)$
- $\alpha = \infty$인 경우에는 $\alpha \sum w^2$ 또는 $\alpha \sum |w|$ 역시 무한대가 되므로 비용 함수는 W를 0에 가깝게 최소화 해야 함

규제 선형 회귀의 유형

- 이처럼 손실 함수에 α값으로 패널티를 부여해 회귀 계수 값의 크기를 감소시켜 과적합을 개선하는 방식을 규제(Regularization)라고 한다.
- 규제는 크게 L2 방식과 L1 방식으로 구분된다.
  - L2 방식 $Loss(x, y) = \arg\min_{w} \sum RSS(W) + \alpha \sum w^2$
  - L1 방식 $Loss(x, y) = \arg\min_{w} \sum RSS(W) + \alpha \sum |w|$
- 릿지(Ridge) 회귀는 L2 방식을 적용한 회귀이며, L2 방식을 적용하면 회귀 계수 값을 무한히 0에 가깝게 만들지만 0이 되진 않는다.
- 라쏘(Lasso) 회귀는 L1 방식을 적용한 회귀이며, L1 방식을 적용하면 영향력이 크지 않은 회귀 계수 값을 0으로 반환한다.
- 엘라스틱 넷(ElasticNet)은 L2, L1 방식을 결합한 모델로서, 주로 Feature가 많은 데이터 세트에 적용된다. L1 규제로 Feature의 개수를 줄임과 동시에 L2 규제로 계수 값의 크기를 조정한다.

릿지(Ridge) 회귀

- 릿지 회귀는 α값을 이용하여 회귀 계수의 크기를 조절한다.
  - α값이 크면 회귀 계수의 값이 작아진다.
  - α값이 작아지면 회귀 계수의 값이 커진다.

실습

Ridge 회귀를 이용한 보스턴 주택가격 회귀 예측

라쏘(Lasso) 회귀

- L2 규제가 회귀 계수의 크기를 감소만 시키는 데 반해, L1 규제는 불필요한 회귀 계수를 급격하게 감소시켜 0으로 만들어 버리고 제거하는 역할이 있다.

- 즉 L1 규제는 적절한 Feature만 회귀에 포함시키는 특성 선택(Feature Selection)의 특징도 가지고 있다.

실습

Lasso 회귀를 이용한 보스턴 주택가격 회귀 예측

엘라스틱 넷(Elastic Net) 회귀

- 엘라스틱 넷(Elastic Net) 회귀는 L2 규제와 L1 규제를 결합한 회귀 모델이다. 따라서 엘라스틱 넷 회귀 손실 함수의 목표 $Loss(x, y) = \arg\min_{w} \sum RSS(W) + \alpha_2 \sum w^2 + \alpha_1 \sum |w|$ 식을 최소화하는 W를 찾는 것이다.
- 엘라스틱 넷은 라쏘 회귀가 서로 상관관계가 높은 피처들의 경우에 이들 중에서 중요 피처만을 선택하고 다른 피처들은 모두 회귀 계수를 0으로 만드는 성향이 강하여 α값에 의해 회귀 계수의 값이 급격히 변동하는 것을 완화해 주기 위해 L2 규제를 라쏘 회귀에 추가해 준 것이다.

엘라스틱 넷(Elastic Net) 회귀

- ElasticNet 클래스의 주요 하이퍼 파라미터는 alpha와 l1_ratio이다.

- alpha
  - alpha 파라미터는 L1, L2 규제에 사용될 alpha의 합이다.
  - $\sum RSS(W) + \alpha_2 \sum w^2 + \alpha_1 \sum |w|$ 에서 $\alpha = \alpha_2 + \alpha_1$이 된다.
- l1_ratio
  - l1_ratio는 L1 규제에 사용할 alpha의 비율로서 $\alpha_1/(\alpha_1 + \alpha_2)$이다.
  - l1_ratio가 0이면 $\alpha_1 = 0$이 되면서 L2 규제와 동일해 진다.
  - l1_ratio가 1이면 $\alpha_2 = 0$이 되면서 L1 규제와 동일해 진다.
  - 0 < l1_ratio < 1 이면 L1과 L2 규제를 적절히 잘 적용한다.
- alpha=10, l1_ratio=0.7 이라면 $\alpha_1 = 10 \times 0.7 = 7$이 되고, $\alpha_2 = 10 \times (1 - 0.7) = 10 \times 0.3 = 3$이 된다.

실습

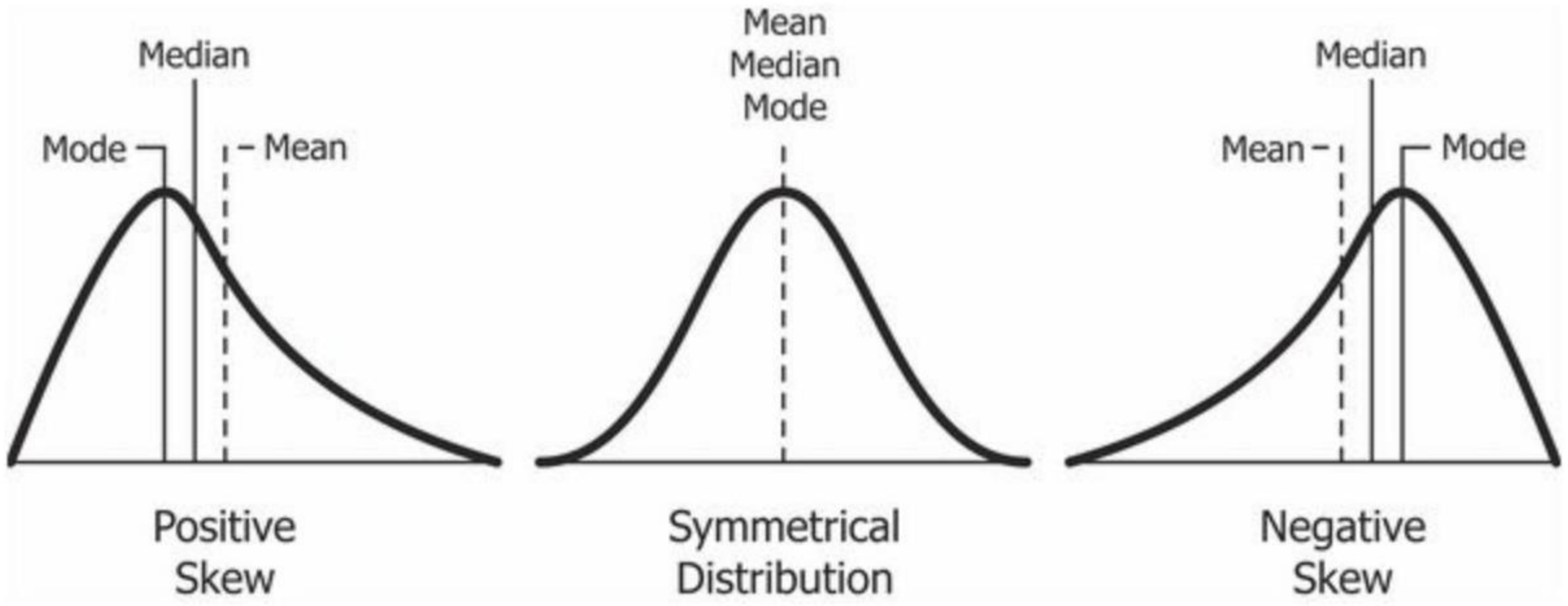
ElasticNet을 이용한 보스턴 주택가격 회귀 예측

선형 회귀 모델을 위한 데이터 변환

선형 회귀를 위한 데이터 변환

- 선형 회귀 모델은 일반적으로 피처와 타겟값 간에 선형의 관계가 있다고 가정하고 이러한 최적의 선형 함수를 찾아내 결과 값을 예측한다.

- 선형 회귀 모델은 피처값과 타겟값의 분포가 정규 분포인 형태를 선호한다.



선형 회귀를 위한 데이터 변환 방법

변환 대상	설명
Target 변환	타겟값은 정규 분포를 선호한다. Skew 되어 있을 경우 주로 로그 변환을 적용한다.
Feature 변환 – Scaling	Feature들에 대한 균일한 스케일링 / 정규화를 적용한다. StandardScaler를 이용하여 표준 정규 분포 형태 또는 MinMaxScaler를 이용하여 최솟값 0, 최댓값 1로 변환한다.
Feature 변환 – 다항 특성 변환	스케일링 / 정규화를 수행한 데이터 세트에 다시 다항 특성(Polynomial Feature)을 적용하여 변환한다.
Feature 변환 – 로그 변환	왜도(Skewness)가 심한 중요 Feature들에 대해서 로그 변환을 적용한다. 일반적으로 많이 사용된다.

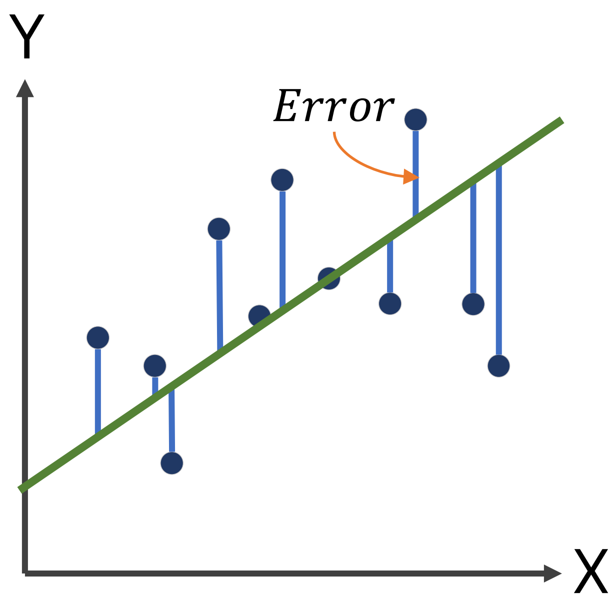
실습

데이터 변환에 따른 선형 회귀 모델 성능 변화 확인하기

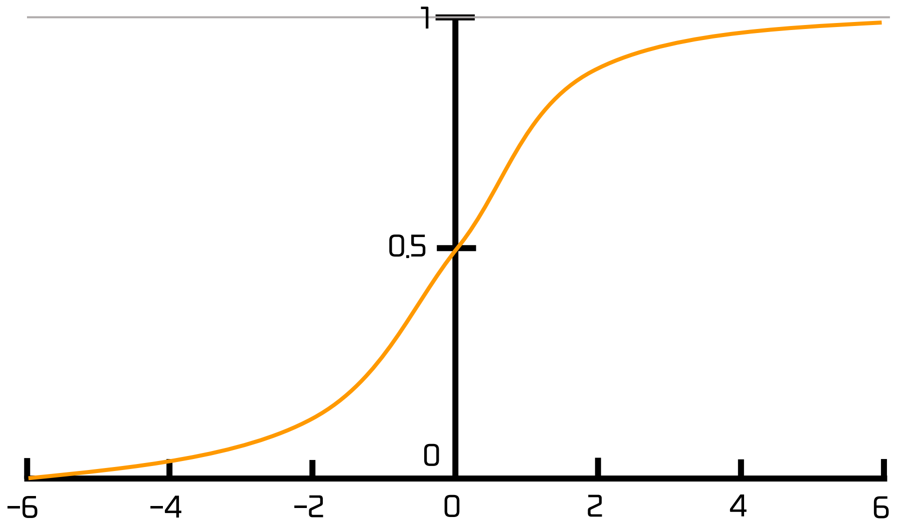
로지스틱 회귀(LogisticRegression)

로지스틱 회귀(Logistic Regression) 개요

- 로지스틱 회귀는 선형 회귀 방식을 분류에 적용한 알고리즘이다. 즉, 로지스틱 회귀는 분류에 사용된다.
- 로지스틱 회귀가 선형 회귀와 다른 점은 선형 함수의 회귀 최적선을 찾는 것이 아니라 시그모이드(σ) 함수의 최적선을 찾고 이 시그모이드 함수의 반환 값을 확률로 간주해 확률에 따라 분류를 결정한다는 것이다.



선형 회귀의 선형 함수

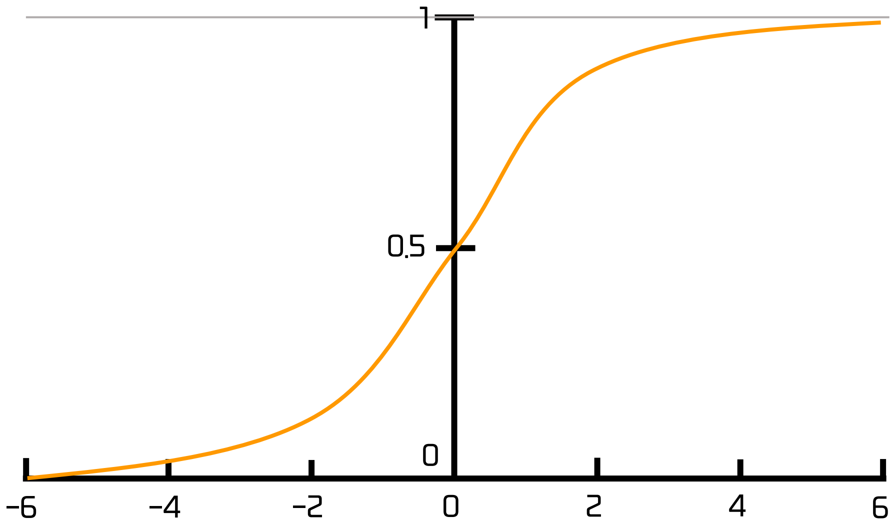


로지스틱 회귀의 시그모이드 함수

로지스틱 회귀 예측

- 로지스틱 회귀는 주로 이진 분류(0과 1)에 사용된다. 다중 분류에도 사용이 될 수 있다. 로지스틱 회귀에서 예측 값은 예측 확률을 의미하며 예측 값(예측 확률)이 0.5 이상이면 1로, 0.5 이하이면 0으로 예측한다. 로지스틱 회귀의 예측 확률은 시그모이드 함수의 출력값으로 계산된다.

시그모이드 함수 $\sigma(x) = \frac{1}{1+e^{-x}}$



- 단순 선형 회귀 $y = w_1x + w_0$가 있다고 할 때

로지스틱 회귀는 0과 1을 예측하기에 단순 회귀식에 적용할 수는 없다. 하지만 Odds(성공확률 p)을 통해 선형 회귀식에 확률을 적용한다. 성공확률이 p이면 실패 확률은 1-p이다.

$$Odds(p) = p/(1-p)$$

하지만 확률 p의 범위가 0 ~ 1 사이이고, 선형 회귀의 반환값인 $-\infty \sim +\infty$값에 대응하기 위해서 로그 변환을 수행하고 아래와 같이 선형 회귀를 적용한다. 이를 로짓 변환(Logit)이라고 한다.

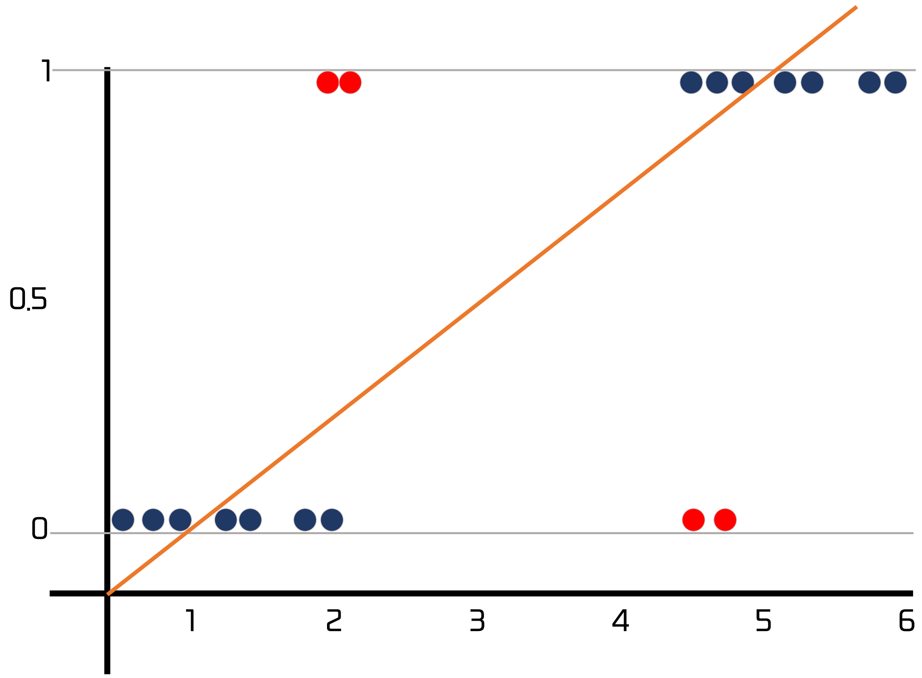
$$\log(Odds(p)) = w_1x + w_0$$

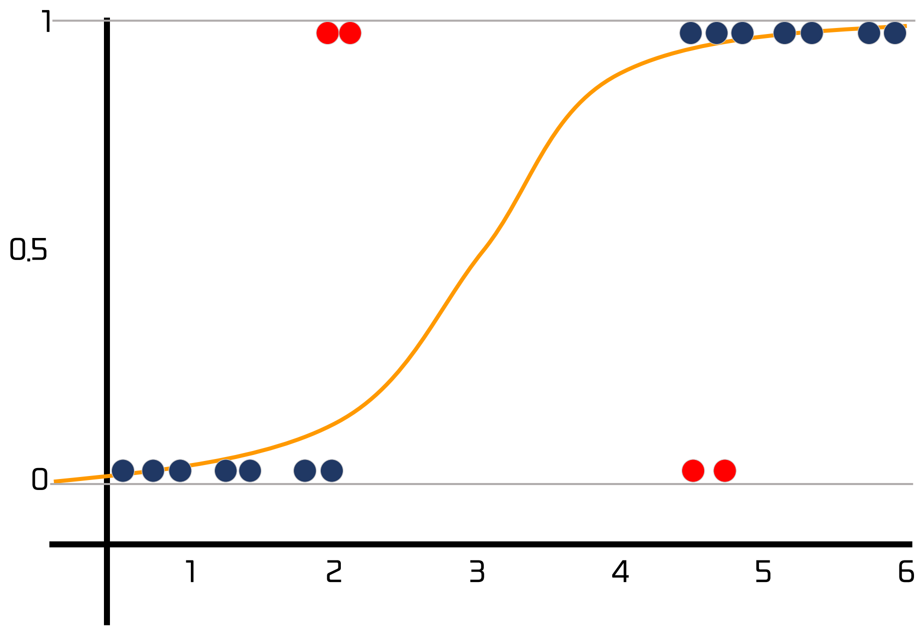
해당 식을 데이터 값 x의 확률 p로 정리하면 다음과 같다.

$$p(x) = \frac{1}{1+e^{-(w_1x+w_0)}}$$

로지스틱 회귀는 학습을 통해서 시그모이드 함수의 w를 최적화하여 예측하는 것이다.

시그모이드를 이용한 로지스틱 회귀 예측





사이킷런 로지스틱 회귀

- 사이킷런은 로지스틱 회귀를 LogisticRegression 클래스로 구현
- LogisticRegression의 주요 하이퍼 파라미터로 penalty, C, solver가 있다.
  - penalty : 규제 유형 설정. 'l2', 'l1' 설정 가능
  - C : 규제 강도를 조절하는 α의 역수. 즉 C=1/alpha. C가 작을 수록 규제 강도가 커짐
  - solver : 회귀 계수 최적화를 위한 다양한 최적화(Optimization) 방식
    - lbfgs : 사이킷런 버전 0.22 부터 solver의 기본 설정값. 메모리 공간을 절약할 수 있고 CPU 코어 수가 많다면 최적화를 병렬로 수행 가능
    - liblinear : 사이킷런 버전 0.21 까지 solver의 기본 설정값. 다차원이고 작은 데이터 세트에서 효과적으로 동작하지만 국소 최적화(Local Minimum)에 이슈가 있고, 병렬 최적화가 불가능
    - newtown-cg : 좀 더 정교한 최적화를 가능하게 하지만, 대용량의 데이터에서 속도가 많이 느려짐
    - sag : Stochastic Average Gradient로서 경사 하강법 기반의 최적화를 사용. 대용량의 데이터에서 빠르게 최적화 가능
    - saga : sag와 유사한 최적화 방식이며 L1 정규화를 가능하게 해준다.

실습

LogisticRegression을 이용한 위스콘신 암 예측 모델

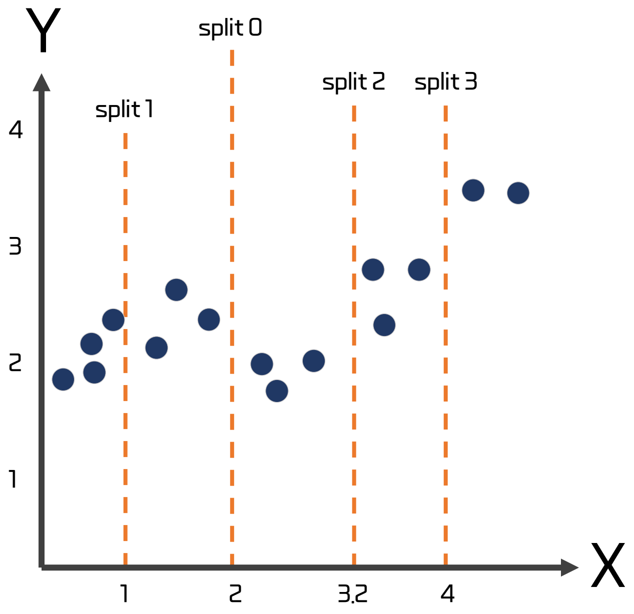
Tree 기반 회귀모델 이해하기

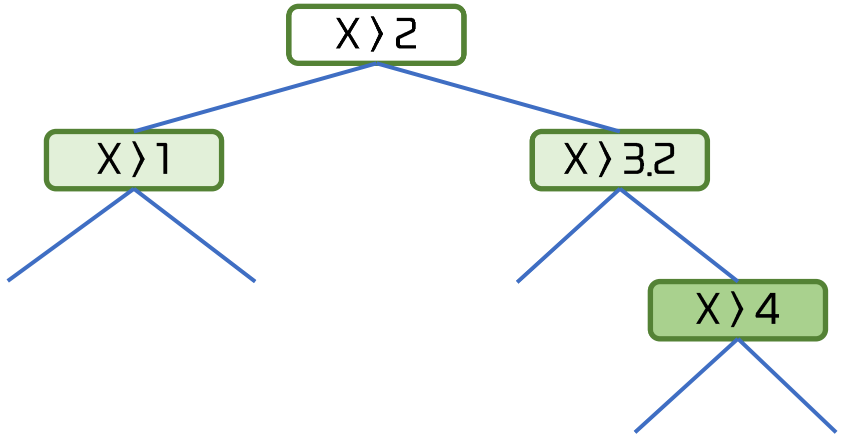
회귀 트리 개요

- 사이킷런의 결정 트리 및 결정 트리 기반의 앙상블 알고리즘은 분류 뿐만 아니라 회귀도 가능
- 이는 사이킷런의 트리가 CART(Classification And Regression Tree)를 기반으로 만들어졌기 때문이다. CART는 분류 뿐만 아니라 회귀도 가능한 트리 분할 알고리즘
- CART 회귀 트리는 분류와 유사하게 분할을 하며, 최종 분할이 완료된 후에 각 분할 영역에 있는 데이터 결정값들의 평균값으로 학습 / 예측을 수행

회귀 트리 프로세스

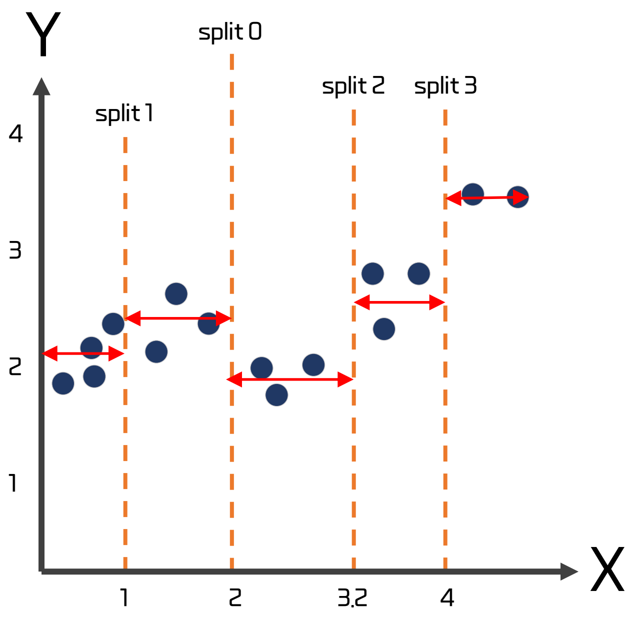
1. 기준에 따라 트리 분할

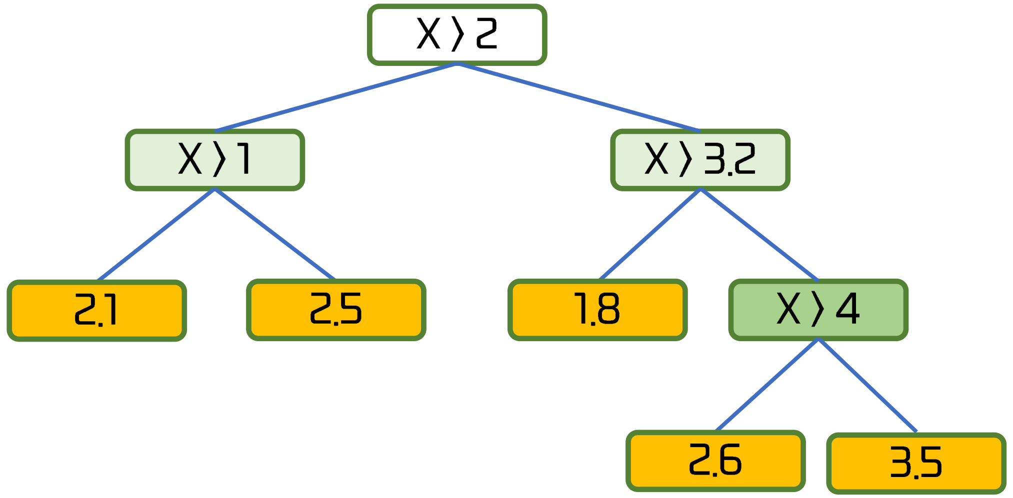




회귀 트리 프로세스

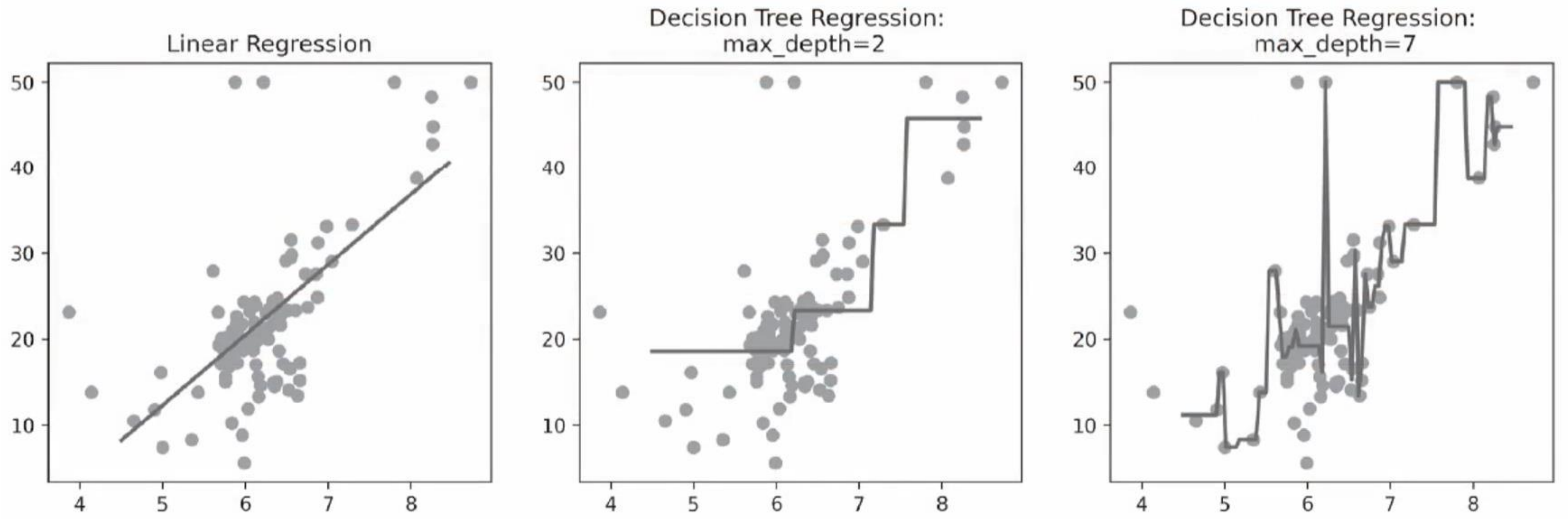
2. 최종 분할된 영역에 있는 데이터들의 평균값들로 학습 / 예측





회귀 트리 오버 피팅

- 회귀 트리 역시 복잡한 트리구조를 가질 경우 과적합 되기 쉽다. 따라서 트리의 크기와 노드 개수의 제한 등의 방법을 통해 과적합을 개선할 수 있다.



실습

자전거 대여(공유) 수요 예측 모델 만들기

실습

캐글 주택가격 예측 모델 만들기