

5장 Regression

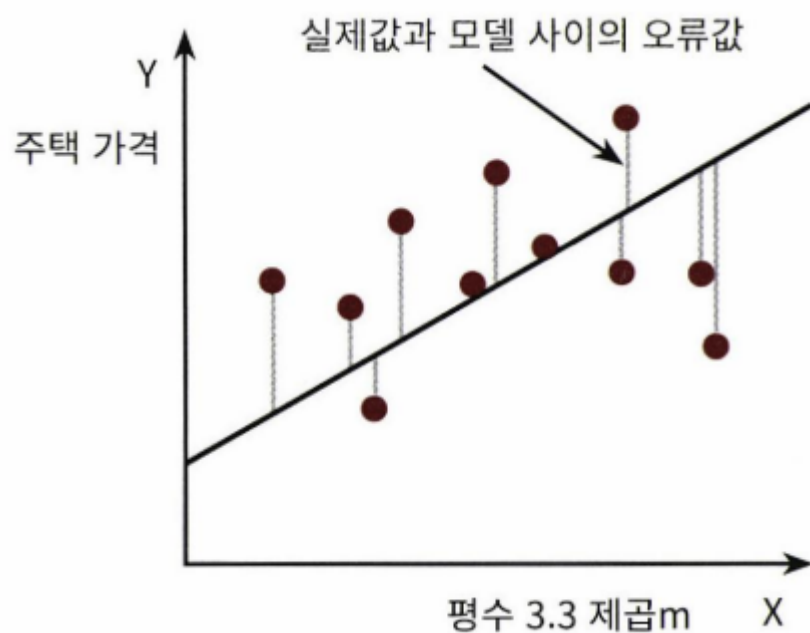
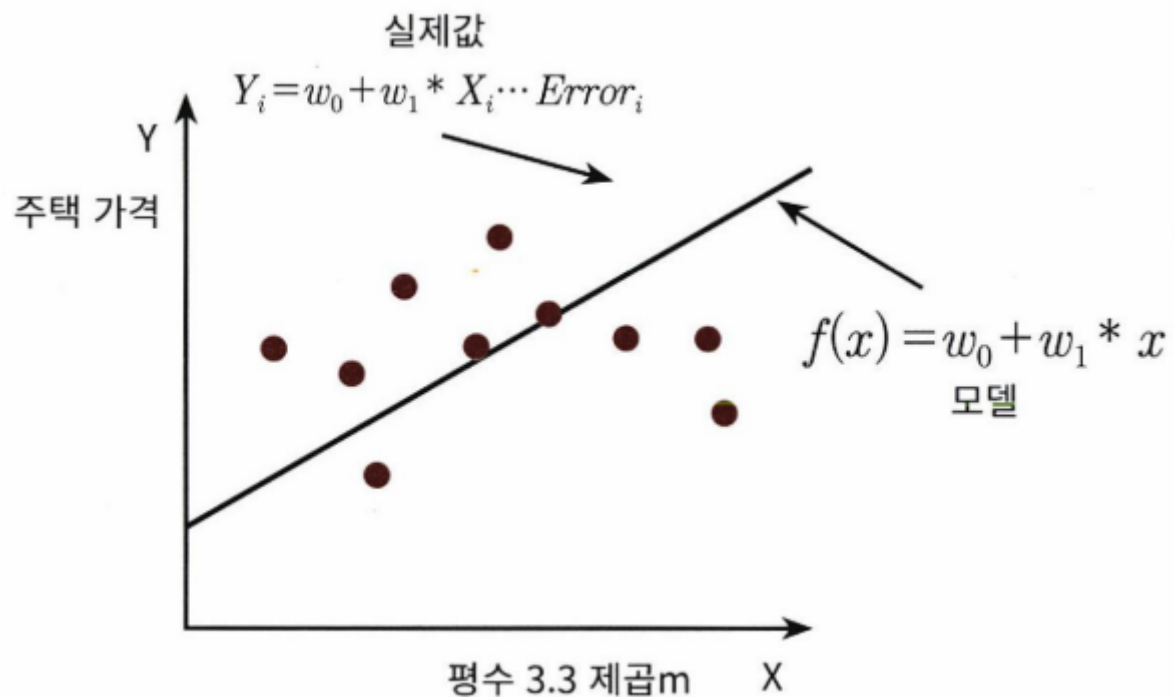
5.1 회귀 소개

- 회귀 : 여러 개의 독립 변수와 한 개의 종속변수 간의 상관관계를 모델링하는 기법을 통칭한다.
- <회귀 유형 구분>

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

- **선형 회귀** : 실제 값과 예측값의 차이(오류의 제곱 값)를 최소화하는 직선형 회귀선을 최적화하는 방식
- **규제(Regularization)**에 따른 선형 회귀 방법 모델
 - 일반 선형 회귀 : 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 회귀 모델
 - 엘라스틱넷(ElasticNet) : L2, L1 규제를 결합한 모델
 - 로지스틱 회귀(Logistic Regression) : 사실상 분류에 사용되는 회귀 모델(매우 강력한 분류 알고리즘)

5.2 단순 선형 회귀를 통한 회귀 이해



- 목표(최적의 회귀 모델) : 전체 데이터의 잔차(오류 값) 합이 최소가 되는 모델을 만든다는 것!

그렇다면 전체 데이터의 잔차(오류 값) 합은 어떻게 계산될 수 있을까?

$$RSS(w_0, w_1) = 1/N * (\sum (y_i - (w_0 + w_1 * x_i))^2)$$

- RSS: 비용 함수(Cost Function)이라고 부르고, 이 비용 함수가 반환하는 값을 지속해서 감소시키고 최종적으로는 **더 이상 감소하지 않는 최소의 오류 값**을 구하는게 목적이다.

5.3 비용 최소화 하기 - 경사 하강법(Gradient Descent) 소개

지금까지 우리의 목적이 잔차의 최소화 즉 w 를 줄이는 것이라는 것을 알았다! 그렇다면 어떻게 줄일 수 있을까?

- 경사하강법 : “점진적으로” 반복적인 계산을 통해 w 파라미터 값을 업데이트하면서 오류 값이 최소가 되는 w 파라미터를 구하는 방식입니다.
- 핵심 아이디어 : “어떻게 하면 오류가 작아지는 방향으로 w 값을 보정할 수 있을까?”

$$R(w) = 1/N * (\sum (y_i - (w_o + w_i * x_i))^2)$$

윗 식을 w_1 에 관해 편미분하면 아래와 같다.

$$\frac{\sigma R(w)}{\sigma w_1} = 2/N * (\sum x_i * (y_i - (w_o + w_i * x_i)))$$

$$\frac{\sigma R(w)}{\sigma w_1} = -2/N * (\sum x_i * (\text{실제값}_i - \text{예측값}_i))$$

마찬가지로 w_0 에 관해 편미분하면 아래와 같다.

$$\frac{\sigma R(w)}{\sigma w_0} = 2/N * (\sum -(y_i - (w_o + w_i * x_i)))$$

$$\frac{\sigma R(w)}{\sigma w_0} = -2/N * (\sum (\text{실제값}_i - \text{예측값}_i))$$

이 후, 이렇게 편미분된 결과값을 마이너스하면서 적용한다.

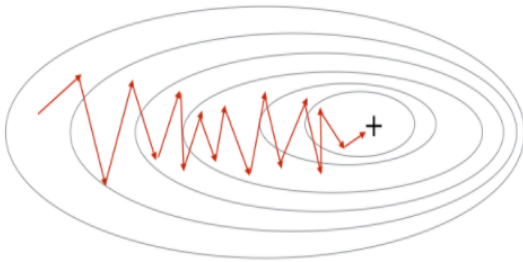
$$w_1(new) = w_1(old) + \eta \frac{2}{N} * (\sum x_i * (\text{실제값}_i - \text{예측값}_i))$$

$$w_0(new) = w_0(old) + \eta \frac{2}{N} * (\sum (\text{실제값}_i - \text{예측값}_i))$$

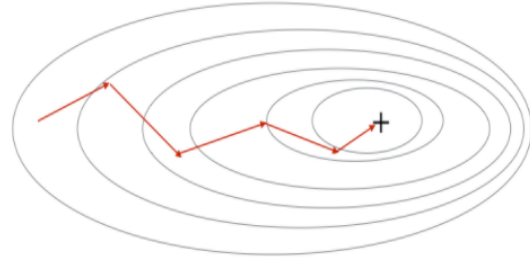
- 이를 업데이트 한 후에 다시 비용함수의 값을 계산한다. 이 를 반복적으로 수행하며 더 이상 비용 함수의 값이 감소하지 않으면 그때의 w_1 , w_0 을 구하고 반복을 중지한다.

- 일반적인 경사 하강법은 모든 데이터에 대해서 반복적으로 비용함수 최소화를 업데이트 하기 때문에 수행시간이 오래걸린다.
- 개선한 방법 : (미니 배치) 확률적 경사 하강법

Stochastic Gradient Descent



Mini-Batch Gradient Descent



- **(미니 배치) 확률적 경사 하강법** : 일부 데이터만 이용해 w가 업데이트되는 값을 계산하므로 경사 하강법에 비해서 빠른 속도를 보장합니다.

지금까지는 피처가 1개, 독립변수가 1개인 단순 선형 회귀에만 경사 하강법을 적용하였다. 그렇다면 **다중 선형 회귀**에서는 어떤식으로 경사 하강법을 적용할 수 있을까?

- 만약 피처가 M개(X_1, X_2, \dots, X_{100})있다면 그에 따른 회귀 계수는 $M+1$ 개로 도출된다.

$$\hat{Y} = w_0 + w_1 * X_1 + w_2 * X_2 + \dots + w_{100} * X_{100}$$

이를 아래와 같은 그림으로 도식화 시킬수 있다.

$$\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \dots \\ \hat{y}_n \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1m} \\ x_{21} & x_{22} & \dots & x_{2m} \\ \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & \dots & x_{nm} \end{bmatrix} \underset{\text{내적}}{\star} \begin{bmatrix} w_1 & w_2 & \dots & w_m \end{bmatrix}^T + w_0$$

하지만, 위의 그림은 w_0 를 포함하지 못하므로 이를 포함시키는 새로운 X_{mat} 을 만들어 준다.

\hat{Y} 1값을 가진 피쳐 추가 X_{mat}

$$\begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{bmatrix} = \begin{bmatrix} \text{Feat 0} & \text{Feat 1} & \text{Feat 2} & \dots & \text{Feat M} \\ 1 & x_{11} & x_{12} & \dots & x_{1m} \\ 1 & x_{21} & x_{22} & \dots & x_{2m} \\ \dots & \dots & \dots & \dots & \dots \\ 1 & x_{n1} & x_{n2} & \dots & x_{nm} \end{bmatrix} \begin{bmatrix} w_0 & w_1 & w_2 & \dots & w_m \end{bmatrix}^T$$

w_0 을 W 배열 내에 포함

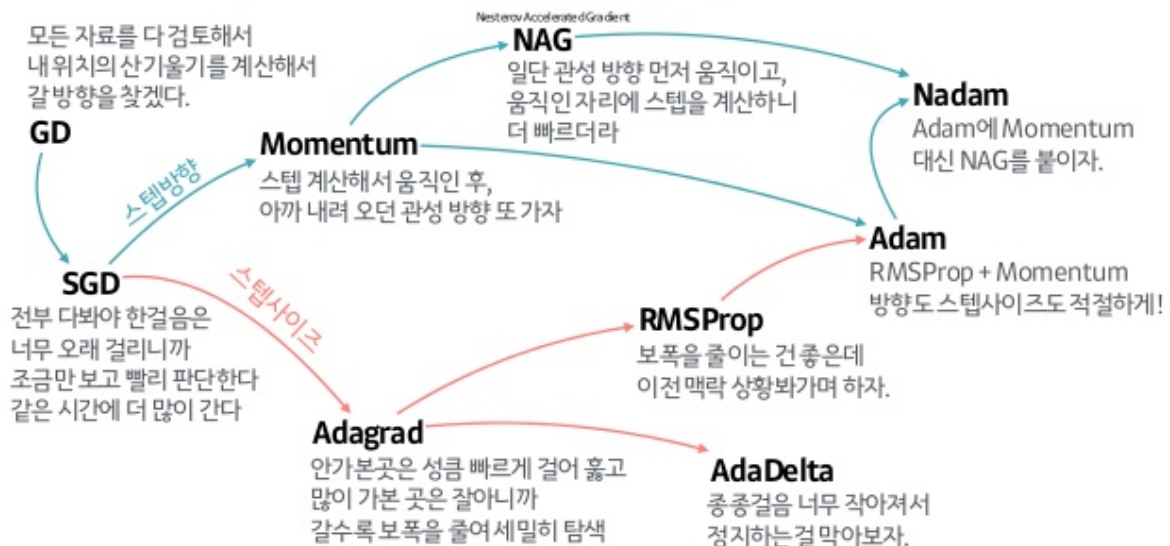
\star 내적

$$\hat{Y} = X_{mat} * W^T$$

코드 구현 부분 자세히 살펴보기

< 대표적인 딥러닝 optimizer 예시)

산 내려오는 작은 오솔길 찾기(Optimizer)의 발달 계보



5.4 사이킷런 LinearRegression을 이용한 보스턴 주택 가격 예측

- LinearRegression - RSS를 최소화해 OLS(Ordinary Least Squares - 최소 제곱법 (Least Square Method) 추정 방식으로 구현

- 회귀 평가 지표

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 Y_i - \hat{Y}_i $
MSE	Mean Squared Error(MSE)이며 실제 값과 예측값의 차이를 제곱해 평균한 것입니다.	$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$
RMSE	MSE 값은 오류의 제곱을 구하므로 실제 오류 평균보다 더 커지는 특성이 있으므로 MSE에 루트를 씌운 것이 RMSE(Root Mean Squared Error)입니다.	$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2}$
R^2 결정계수	분산 기반으로 예측 성능을 평가합니다. 실제 값의 분산 대비 예측값의 분산 비율을 지표로 하며, 1에 가까울수록 예측 정확도가 높습니다. 0이 가까울수록 설명력이 낮고, 1이 가까울수록 설명력이 높다고 해석할 수 있음	$R^2 = \frac{\text{예측값 Variance}}{\text{실제값 Variance}}$

5.5 다항 회귀와 과(대)적합/과소적합 이해

- 다항 회귀 이해!

지금까지 설명한 회귀는 독립변수와 종속변수의 관계가 일차 방정식으로 표현되는 회귀이다.

하지만 이 외에도 독립변수의 단항식이 아닌 2차, 3차 방정식과 같은 다항식으로 표현되는 것을 **다항(Polynomial) 회귀**라고 한다.

- 주의 : 다항 회귀를 비선형 회귀로 혼동하기 쉽지만, 다항 회귀는 선형 회귀다.

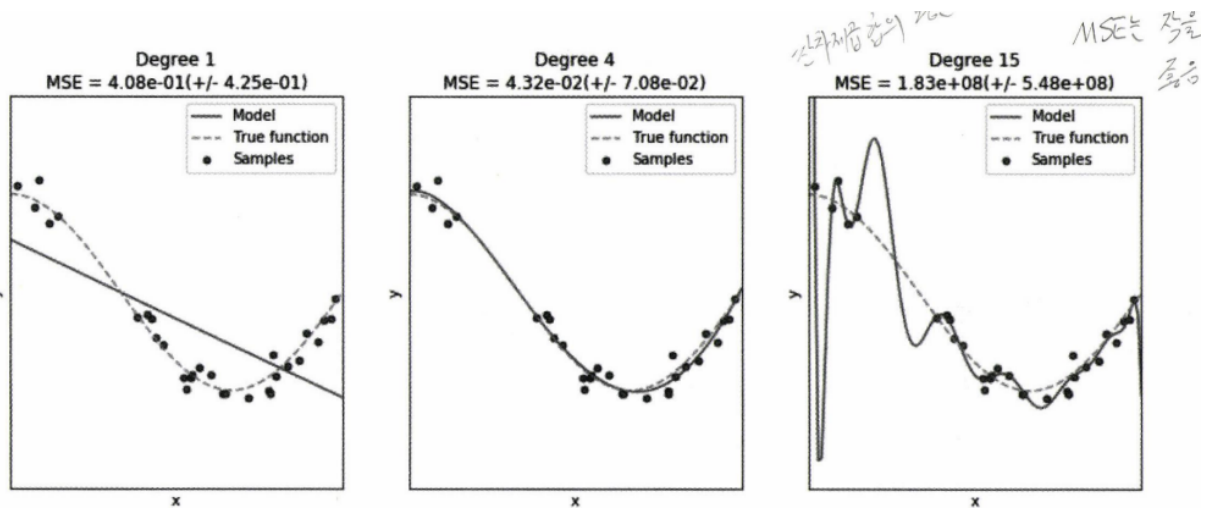
→ 회귀에서 선형 회귀/비선형 회귀를 나누는 기준은 회귀 계수의 선형/비선형성이다!

$$y = w_0 + w_1 \times x_1 + w_2 \times x_2 + w_3 \times x_1 \times x_2 + w_4 \times x_1^2 + w_5 \times x_2^2$$

아쉽지만 사이킷런은 다항 회귀를 위한 클래스를 명시적으로 제공하지 않는다! (직접 구현해보자)

- 코드를 통해 살펴본다.

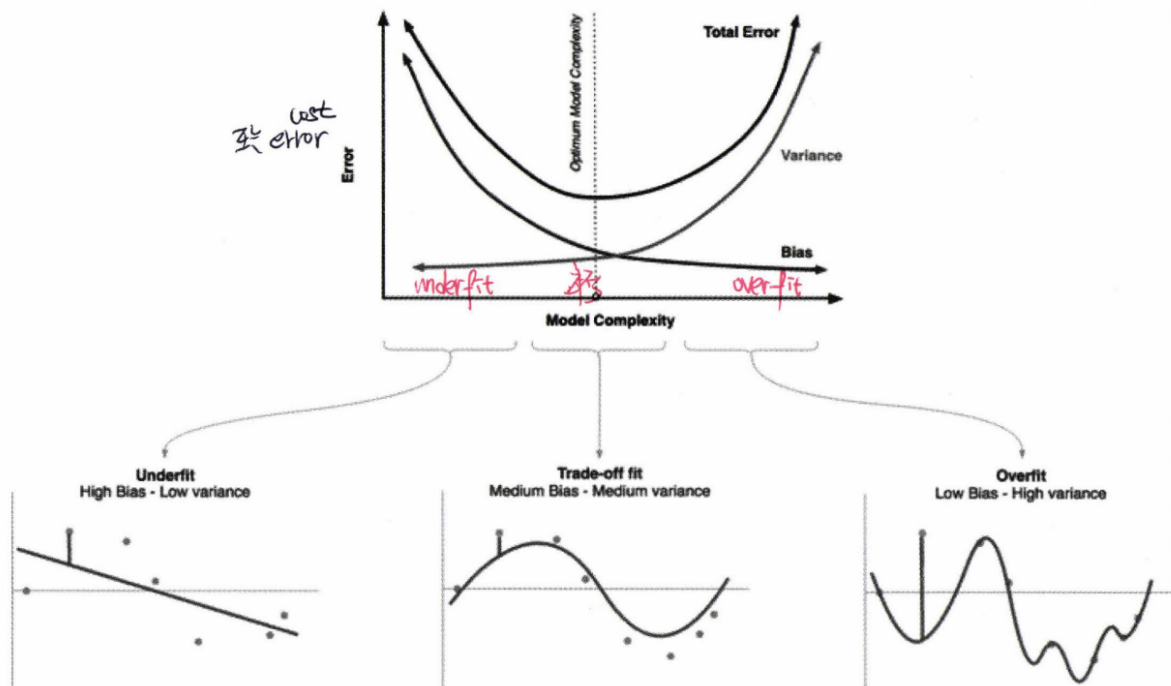
다항회귀를 이용한 과소적합 및 과적합 이해



1. 맨 왼쪽의 경우 과소적합
2. 중간 그림의 경우 잡음까지는 예측하지 못했지만 비교적 잘 예측함
3. 맨 오른쪽 그림은 학습데이터만 정확히 예측하고 테스트 데이터서는 완전히 다른 과대적합이다.

편향-분산 트레이드오프(Bias-Variance Trade off)

- 일반적으로 편향과 분산은 한 쪽이 높으면 한 쪽이 낮아지는 경향이 있다. 즉, 편향이 높으면 분산은 낮아지고(과소적합) 반대로 분산이 높으면 편향이 낮아진다(과적합)



〈 편향과 분산에 따른 전체 오류 값(Total Error) 곡선. <http://scott.fortmann-roe.com/docs/BiasVariance.html>에서 발췌. 〉

즉 **편향과 분산이 트레이드오프**를 이루면서 **오류 Cost 값이 최대한 낮아지는 모델**을 구축하는 것이 가장 효율적인 머신러닝 예측 모델을 만드는 방법이다.

5.6 규제 선형 모델 - 릿지(Ridge), 라쏘(Rasso), 엘라스틱넷(ElasticNet)

회귀 모델은 적절히 데이터에 적합하면서도 회귀 계수가 기하급수적으로 커지는 것을 제어할 수 있어야한다. → **규제 모델**의 필요성 대두!

아이디어 : 학습데이터 잔차 오류 최소화 + 회귀계수 크기 제어(무조건 크면 과적합)

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

- **alpha = 0**인 경우는 W가 커도 alpha가 곱해진 후항이 0이 되어 비용함수가 Min(RSS(w))가 된다.(기존 그대로)
- 반면 **alpha 값을 키워가며 회귀 계수 값의 크기를 감소시켜 과적합을 개선** 할 수 있다.
- **alpha = 무한대**인 경우 후항이 무한대가 되므로 W를 0에 가깝게 최소화 되는 것도 유의하자.
- L2 규제(릿지) : $\alpha * W^2$ 에서 W의 제곱에 패널티를 부여하는 방식
- L1 규제(라쏘) : $\alpha * |W|$ 와 같이 W의 절대값에 대해 패널티를 부여합니다.

릿지(Ridge) 회귀

- alpha L2 규제를 사용

라쏘(Lasso) 회귀

- W의 절대값에 패널티를 부여하는 L1 규제를 적용
- 불필요한 회귀 계수를 급격하게 감소시켜, 0으로 만든다. → 피쳐 선택의 특성을 보여줌

엘라스틱넷 회귀

- L2 규제와 L1 규제를 결합한 회귀입니다.
- 서로 상관관계가 높은 피쳐들 중에서 중요 피쳐만을 선택하고 다른 피쳐들은 모두 회귀 계수를 0으로 만드는 성향이 강하다.

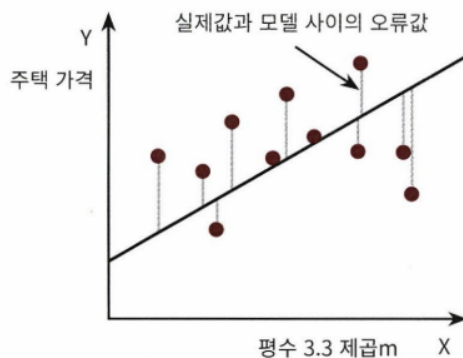
선형 회귀 모델을 위한 데이터 변환

1. StandardScaler를 이용한 정규 분포 변환

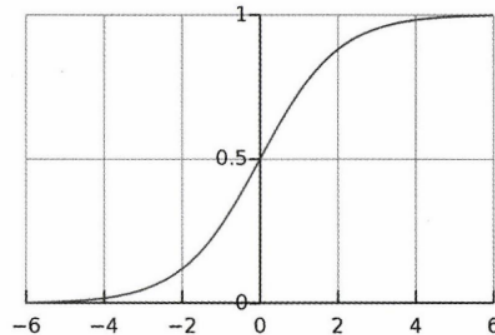
2. 1번의 방식 후 성능 향상이 없을시 다항 특성을 적용하여 변환하는 방법이 있다. → 과도한 피쳐 개수로 과적합 주의
3. 로그 변환을 통해 정규 분포에 가까운 형태로 값을 변경시킨다. ($\log_{10} p$) → 성능 향상이 좋다고 함.

5.7 로지스틱 회귀

- 로지스틱 회귀 : 선형 회귀 방식을 분류에 적용한 알고리즘이다. 선형 회귀를 기반으로 하되 시그모이드 함수를 이용해 분류를 수행하는 회귀이다.



〈 선형 회귀의 선형 함수 〉

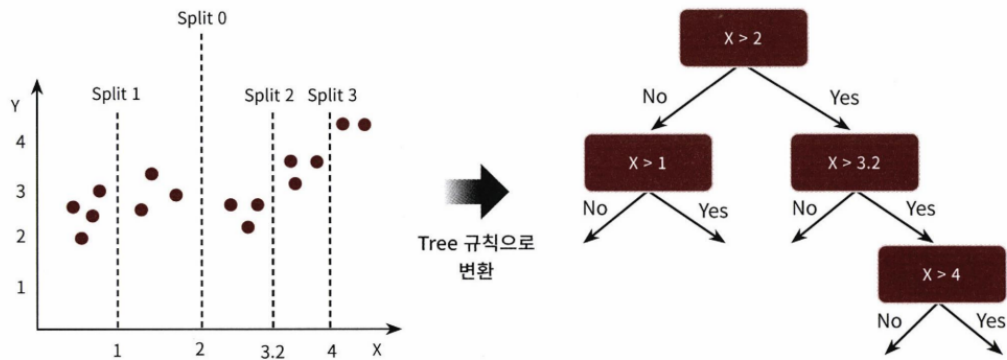


〈 로지스틱 회귀의 시그모이드 함수 〉

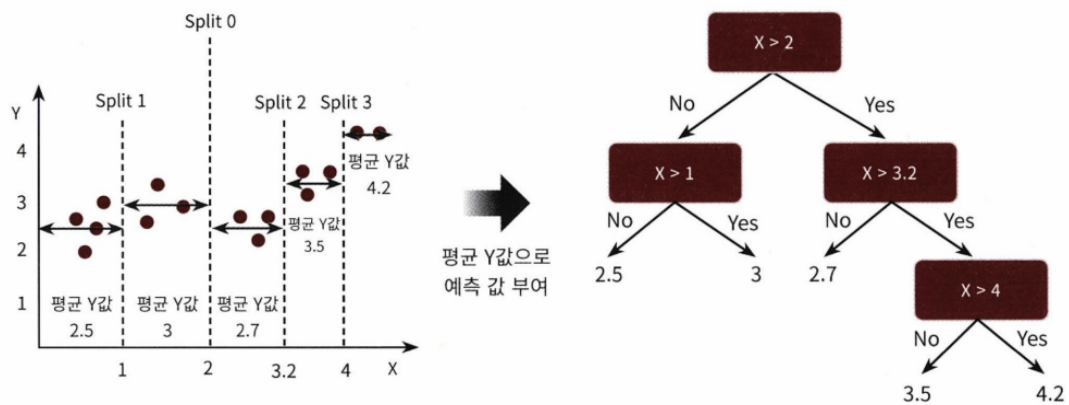
- 최대 우도법, 경사하강법, 뉴턴법 등의 최적화 알고리즘을 통해 최적해를 예측 가능하다.
- 세부 수학적 방법은 아래 블로그를 참고해주세요
- <https://datascienceschool.net/03 machine learning/10.01 로지스틱 회귀분석.html>

5.8 회귀 트리

- 지금까지는 선형 회귀에 대해 알아보았습니다. 즉 회귀 계수의 관계가 모두 선형으로 가정하는 방식이었습니다.
- 비선형 회귀 역시 비선형 회귀 계수의 관계를 통해 결과값을 예측합니다. 다만 비선형 회귀는 회귀 계수의 결합이 비선형이다.
- 회귀 트리는 분류 트리와 크게 다르지 않는데, 리프 노드에서 예측 결정 값을 만드는 과정에서 차이가 있다.
- 분류 트리는 특정 클래스 레이블을 결정하는 것과는 다르게 회귀 트리는 리프 노드에 속한 **데이터 값의 평균값**을 구해 회귀 예측값을 계산한다.



- 처음 위와 같은 스플릿은 X값의 균일도를 반영한 **지니계수에 따라 분할**이 가능하다.



- 이 후에, 각 리프 노드에 소속된 데이터 값의 평균값을 구해서 최종적으로 리프 노드에 결정 값으로 할당한다.
- 사이킷런의 트리 기반 회귀와 분류의 Estimator 클래스

알고리즘	회귀 Estimator 클래스	분류 Estimator 클래스
Decision Tree	DecisionTreeRegressor	DecisionTreeClassifier
Gradient Boosting	GradientBoostingRegressor	GradientBoostingClassifier
XGBoost	XGBRegressor	XGBClassifier
LighGBM	LGBMRegressor	LGBMClassifier

5.9 회귀 실습 - 자전거 대여 수요 예측

- 성능이 좋지 못한 경우 - target의 분포를 살펴볼것(정규 분포를 띄는것이 예측에 유리하다.)

- 연도같은 숫자를 변수로 사용할 경우 그 자체의 수치자체가 크기 때문에 예측 성능을 저하하기도 한다. → 원-핫 인코딩 처리가 필요하다.(숫자형-카테고리형 변수들)

5.10 회귀 실습 - 캐글 주택 가격 : 고급 회귀 기법