회귀(Regression)

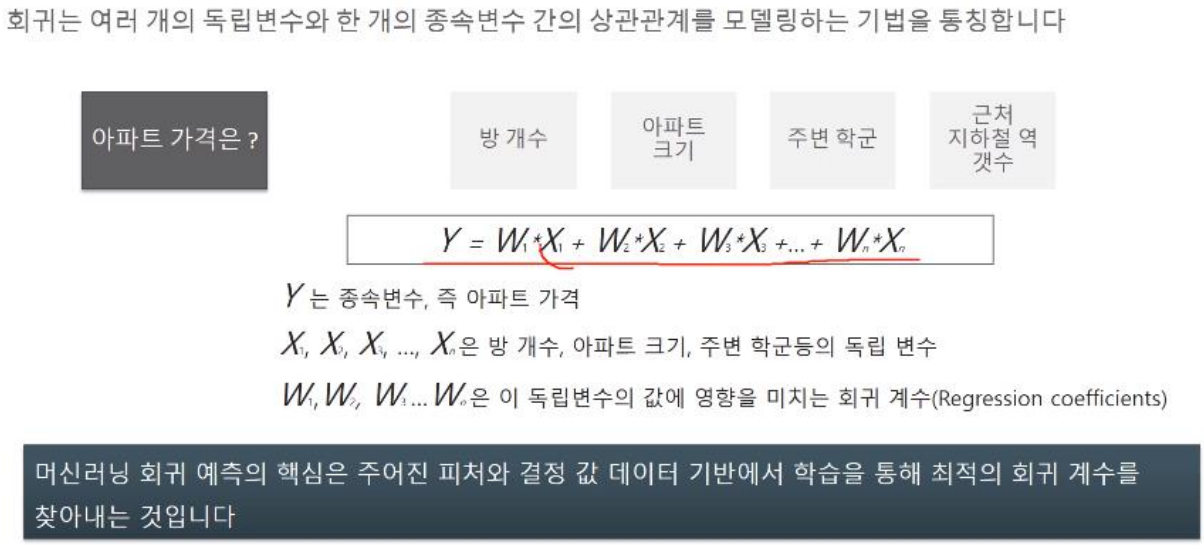
회귀는 현대 통계학을 이루는 큰 축

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"

⇨ 이처럼 데이터 값이 **평균과 같은 일정한 값으로 돌아가려는 경향**을 이용한 통계학 기법

여러 개의 독립변수와 한 개의 종속변수 간의 상관관계를 모델링하는 기법을 통칭한다.



y는 결정 값, W는 회귀 계수, X는 독립 변수(피처)

결정 값이 주어져 있으므로 지도 학습

정형 데이터일 경우에 선형 회귀가 비선형 회귀보다 예측 성능이 좋다.

회귀 계수에 따라 선형 회귀, 비선형 회귀로 나뉜다.

- **회귀 계수가 선형 -> 선형 회귀**
- **회귀 계수가 비선형 -> 비선형 회귀**

분류(Classification)와 회귀(Regression)

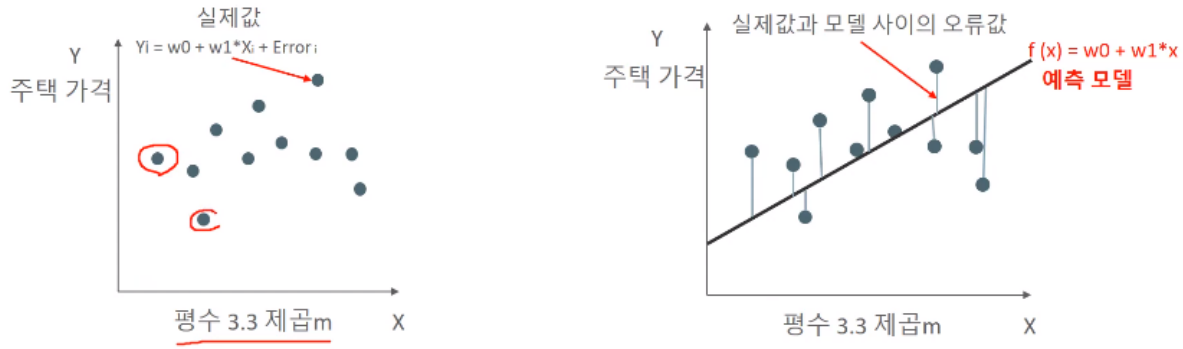
- 분류는 결과값이 **Category 값(이산 값)** ex) 0 or 1 or …
- 회귀는 결과값이 **숫자 값(연속 값)** ex) 1.231, 14.415. …

선형 회귀

선형 회귀의 종류

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

주택 가격이 주택 의 크기로만 결정 되는 단순 선형 회귀로 가정하면 다음과 같이 주택 가격은 주택 크기에 대해 선형(직선 형태)의 관계로 표현할 수 있습니다.



실제 값과 모델의 예측 값의 차이가 적을수록 모델이 잘 만들어졌다고 할 수 있다.

최적의 회귀 모델을 만든다는 것 :

- **전체 데이터의 잔차(오류 값) 합이 최소가 되는 모델을 만든다는 의미.**
- **동시에 오류 값 합이 최소가 될 수 있는 최적의 회귀 계수를 찾는다는 의미도 된다.**

제곱 Or 절댓값

RSS 기반의 회귀 오류 측정

오류 값은 (+)일 수도, (-)일 수도 있기 때문에 제곱을 구해서 합하는 방식으로 오류 값을 계산한다.

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

(i는 1부터 학습 데이터의 총 건수 N까지)

RSS는 회귀식의 독립변수 X, 종속변수 Y가 핵심이 아니라 W 변수(회귀 계수)가 핵심임을 인지해야 한다.

RSS는 비용 함수(Cost Function, Loss Function, Objective Function라고 함.)

이것이 최소가 되었을 때, 모델이 잘 만들어지는 것.

Gradient Descent

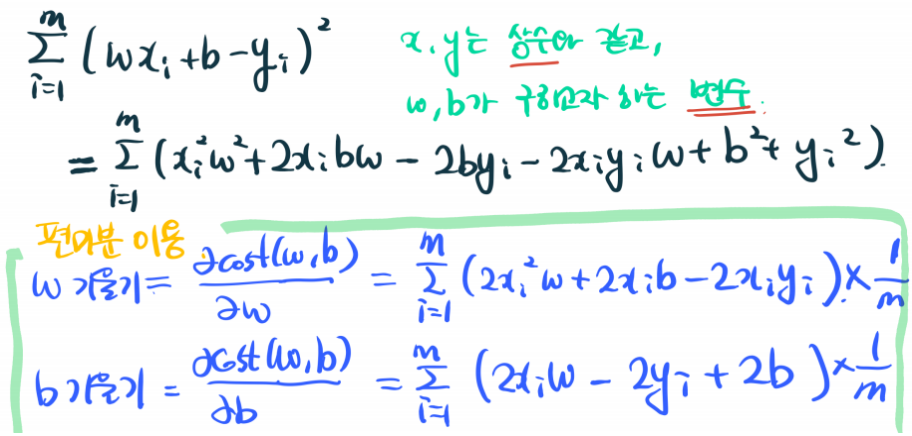
W를 계속 Update하며, Cost가 최소가 되는 곳을 찾는 것.

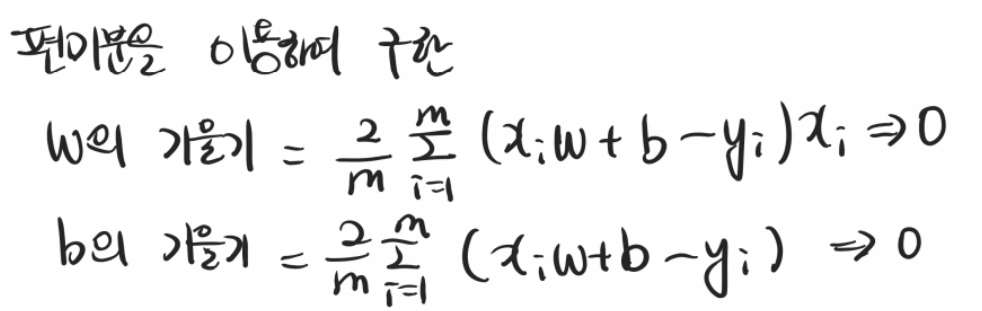
즉, 예측 값과 실제 값의 차이가 작아지는 방향성을 가지고 W parameter를 계속해서 보정해 나간다.

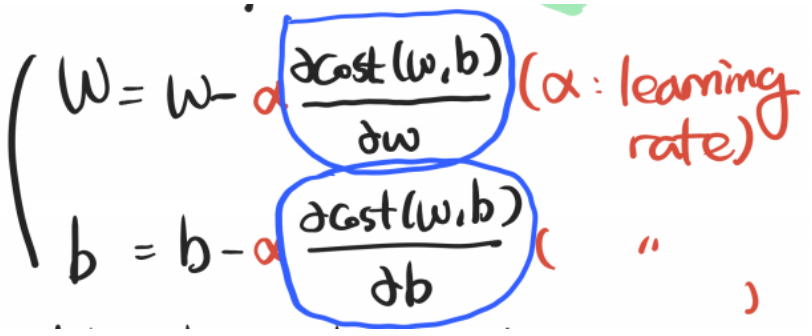
지속적으로 오류 값이 작아지는 방향으로 W를 계속 업데이트해 나가면서, 오류 값이 더 이상 작아지지 않으면 그 오류 값을 최소 비용으로 반환하고 그때의 W 값을 최적 parameter로 반환한다.

핵심 : 어떻게 오류가 작아지는 방향으로 W값을 보정할 수 있을까?

⇨ 미분을 적용하여 순차적으로 W를 업데이트 한다.







경사 하강법(Gradient Descent) 수행 프로세스

- Step 1: w_1, w_0를 임의의 값으로 설정하고 첫 비용 함수의 값을 계산합니다.
- Step 2: w_1을 $w_1 - \eta \frac{2}{N} \sum_{i=1}^{N} x_i * (\text{실제값}_i - \text{예측값}_i)$, w_0을 $w_0 - \eta \frac{2}{N} \sum_{i=1}^{N} (\text{실제값}_i - \text{예측값}_i)$으로 업데이트한 후 다시 비용 함수의 값을 계산합니다.
- Step 3: 비용 함수의 값이 감소했으면 다시 Step 2를 반복합니다. 더 이상 비용 함수의 값이 감소하지 않으면 그때의 w_1, w_0를 구하고 반복을 중지합니다.

요약 : 초기값 => learning rate와 편미분(gradient descent) 적용하여 비용함수 계산(더 이상 cost funtion이 감소하지 않을 때까지 반복)

- gradient descent
- stochastic gradient descent(mini-batch로 나눠서 random sampling으로 가져온 뒤에 gradient descent 적용)

Linear Regression

: 예측 값과 실제 값의 RSS(Residual Sum of Squares)를 최소화해 OLS(Ordinary Least Squares) 추정 방식으로 구현한 클래스

```
class sklearn.linear_model.LinearRegression(fit_intercept=True, normalize=False, copy_X=True, n_jobs=1)
```

fit() 메소드로 X, y 배열을 입력받으면 회귀 계수(Coefficients)인 W를 coef_ 속성에 저장한다.

fit_intercept가 **True**이면 절편(intercept)를 사용하는 것, **False**이면 intercept가 사용되지 않고 **0**으로 지정된다.

선형 회귀의 다중 공선성 문제

- 일반적으로 선형 회귀는 입력 피처의 독립성에 많은 영향을 받는다. 피처 간의 상관관계가 매우 높은 경우 분산이 매우 커져서 오류에 매우 민감해진다. -> **다중 공선성.**
- 일반적으로 상관관계가 높은 feature가 많은 경우 **독립적인 중요한 feature만 남기고 제거하거나 규제를 적용한다.**

회귀 평가 지표

평가 지표	설명	수식
MAE	Mean Absolute Error(MAE)이며 실제 값과 예측값의 차이를 절댓값으로 변환해 평균한 것입니다	$MAE = \frac{1}{n}\sum_{i=1}^{n}\lvert Yi - \hat{Y}i\rvert$
MSE	Mean Squared Error(MSE)이며 실제 값과 예측값의 차이를 제곱해 평균한 것입니다.	$MSE = \frac{1}{n}\sum_{i=1}^{n}(Yi - \hat{Y}i)^2$
MSLE	MSE에 로그를 적용한 것입니다. 결정값이 클 수록 오류값도 커지기 때문에 일부 큰 오류값들로 인해 전체 오류값이 커지는 것을 막아줍니다.	Log(MSE)
RMSE	MSE 값은 오류의 제곱을 구하므로 실제 오류 평균보다 더 커지는 특성이 있으므로 MSE에 루트를 씌운 것이 RMSE(Root Mean Squared Error)입니다	$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Yi - \hat{Y}i)^2}$
RMSLE	RMSE에 로그를 적용한 것입니다. 결정값이 클 수록 오류값도 커지기 때문에 일부 큰 오류값들로 인해 전체 오류값이 커지는 것을 막아줍니다.	Log(RMSE)
R^2	분산 기반으로 예측 성능을 평가합니다. 실제 값의 분산 대비 예측값의 분산 비율을 지표로 하며, 1에 가까울수록 예측 정확도가 높습니다.	$R^2 - \frac{예측값\ Variance}{실제\ 값\ Variance}$

사이킷런에는 RMSLE가 제공되지 않아서 Log(RMSE)로 직접 구현할 필요가 있다.

평가 방법	사이킷런 평가 지표 API	Scoring 함수 적용 값
MAE	metrics.mean_absolute_error	'neg_mean_absolute_error' ←
MSE	metrics.mean_squared_error	'neg_mean_squared_error'
R^2	metrics.r2_score	'r2'

neg_를 적용하는 이유는, 일반적으로 score값이 클수록 좋은 평가결과로 자동 평가하는데 회귀에서는 오류를 나타내므로 작을수록 좋기 때문에 이를 보정하기 위함.

metrics.mean_absolute_error()와 같은 사이킷런 평가 지표 API는 정상적으로 양수의 값을 반환하는데, Scoring 함수의 scoring parameter 파라미터 **neg_mean_absolute_error**가 의미하는 것은

-1*metrics_mean_absolute_error()임을 유의

다항 회귀

회귀에서 선형/비선형 회귀를 나누는 기준은 **회귀 계수가 선형/비선형인지에 따른 것**이지 독립 변수의 선형/비선형 여부와는 무관하다.

사이킷런은 다항회귀를 바로 API로 제공하지 않는다.

대신, PolynomialFeatures 클래스로 원본 단항 feature들을 다항 feature들로 변환한 데이터 세트에 Linear Regression 객체를 적용하여 다항 회귀 기능을 제공한다.

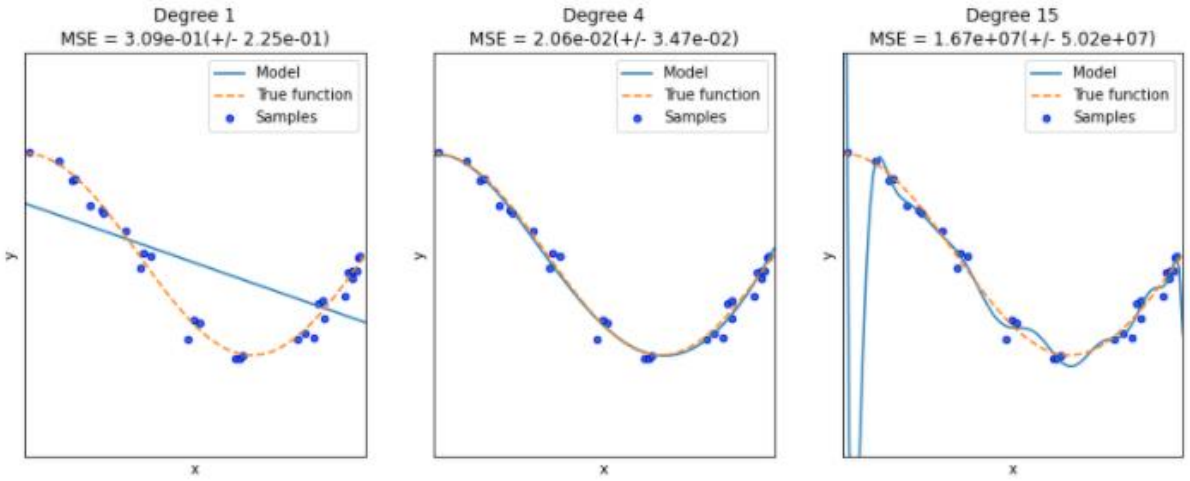
단항 피처 $[x_1, x_2]$ 를 Degree = 2, 즉 2차 다항 피처로 변환한다면?
$(x_1 + x_2)^2$ 의 식 전개에 대응되는 $[1, x_1, x_2, x_1x_2, x_1^2, x_2^2]$의
다항 피처들로 변환

1차 단항 피처들의 값이 $[x_1, x_2]$ = [0 1] 일 경우
2차 다항 피처들의 값은 $[1, x_1 = 0, x_2 = 1, x_1x_2 = 0, x_1^2 = 0, x_2^2 = 1]$
형태인 [1, 0, 1, 0, 0, 1] 로 변환

사이킷런에서는 일반적으로 **Pipeline** 클래스를 이용하여 PolynomialFeatures 변환과 LinearRegression 학습/예측을 결합하여 다항 회귀를 구현한다.

다항회귀의 단점

- Overfitting 이 발생하기 쉽다. 다항식 값(ㅁ차)가 높아질수록 overfitting 이 발생하기 쉽다.

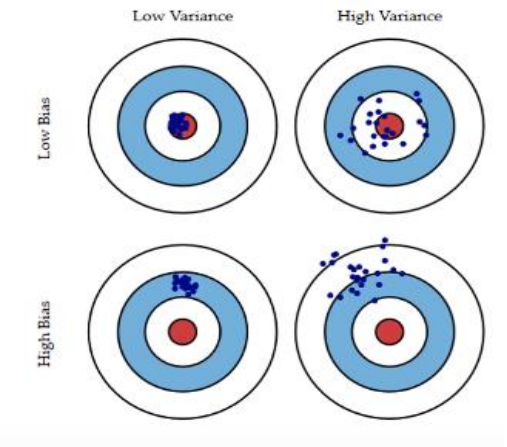


각각 Degree 1, 4, 15

Degree 1 -> 과소 적합(Underfitting) : High bias

Degree 4 -> Well fit

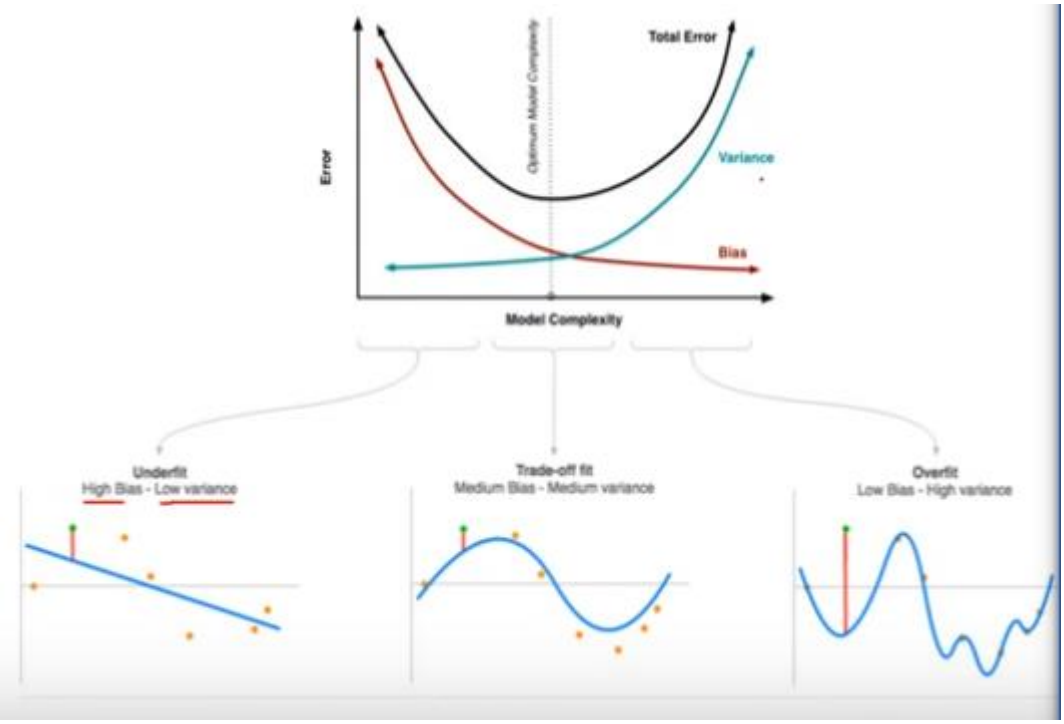
Degree 15 -> 과대 적합(Overfitting) : High variance



Well fit 되기 쉽지 않다.

일반적으로 Bias-Variance는 Trade off 관계를 가진다.

딥러닝은 이를 극복하여 어느정도의 성능을 보장하기 때문에 각광받는 것.



규제 선형 회귀

: 데이터에 적합하면서도 회귀 계수가 기하급수적으로 커지는 것을 제어할 수 있어야 한다.

학습데이터 잔차 오류 최소화 + 회귀계수 크기 제어

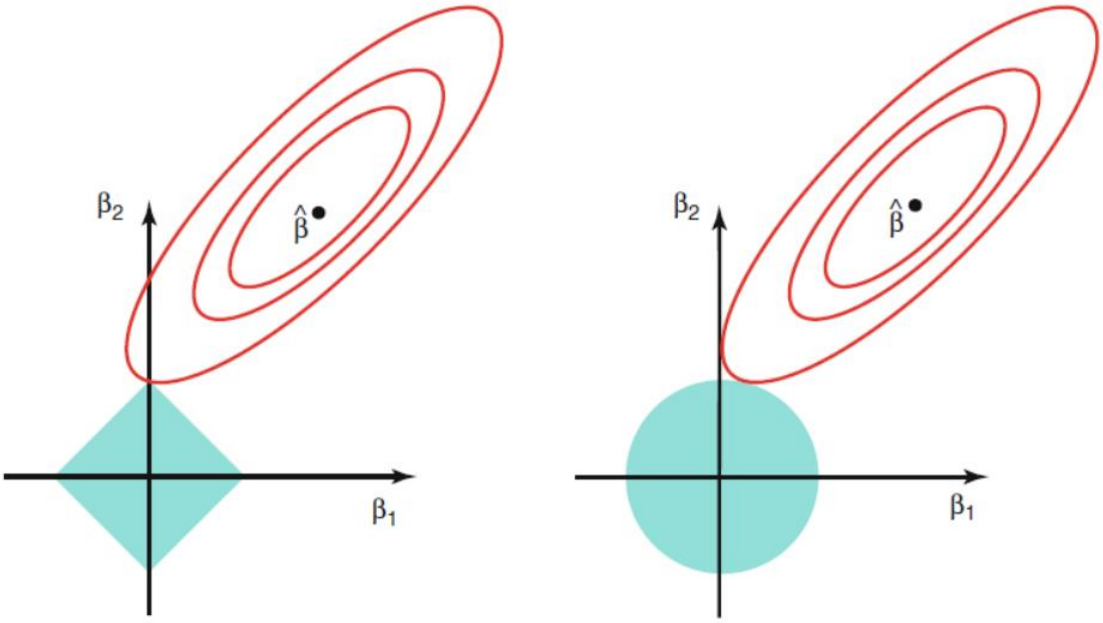
$$\text{비용 함수 목표} = \text{Min}\ (\ \text{RSS(W)} + \text{alpha} * ||W||_2^2\ )$$

- alpha가 0(또는 매우 작은 값)이면 비용 함수 식은 기존과 동일한 Min(RSS(W) + 0)이 된다.
- alpha가 무한대(또는 매우 큰 값)이면 W값을 0(또는 매우 작은 값)으로 만들어야 Cost가 최소화 되는 비용 함수 목표를 달성할 수 있다.
- alpha 값을 크게 하면 비용 함수는 회귀 계수 W의 값을 작게 해 과적합을 개선할 수 있으며, alpha 값을 작게 하면 회귀 계수 W의 값이 커져도 어느 정도 상쇄가 가능하므로 학습 데이터 적합을 더 개선할 수 있다.

비용 함수에 **alpha 값으로 penalty**를 부여해 회귀 계수 값의 크기를 감소시켜 과적합을 개선하는 방식을 규제(Regularization)이라고 함.

L1 규제 적용(Lasso) : W의 **절댓값**에 대해 패널티 부여

L2 규제 적용(Ridge) : W의 **제곱**에 대해 패널티 부여



왼쪽은 Lasso(절댓값에 따른 그래프), 오른쪽은 Ridge(제곱에 따른 그래프)

Lasso가 불필요한 회귀 계수를 0으로, Ridge가 0이 안되고 계수 값의 크기가 작아지는 것을 시각적으로 확인할 수 있다.

L1, L2 규제를 결합(ElasticNet) : L1, L2 규제를 함께 결합한 모델. 피처가 많은 데이터 세트에서 적용되며, L1 규제로 피처의 개수를 줄임과 동시에 L2 규제로 계수 값의 크기를 조정한다.

L2 규제는 주기적으로 회귀 계수를 감소시키는 반면, L1 규제는 불필요한 회귀 계수를 0으로 만듦

이런 특성상 Feature Selection으로도 L1 규제를 볼 수 있다.

ElasticNet의 alpha = L1 규제의 alpha(Lasso) + L2 규제의 alpha(Ridge)

선형 회귀 모델을 위한 데이터 변환

: 회귀 모델과 같은 선형 모델은 일반적으로 feature와 타겟 값 간에 선형의 관계가 있다고 가정하고, 이러한 최적의 선형 함수를 찾아내 결과를 예측. 피처값과 타겟값의 분포가 정규분포일 때 성능이 높다.-> **정규분포인지 알 수 있는 방법 -> 그림 이후 표준편차, 정규분포인지 검사하는 방법이 있기는 함.**

타겟값 변환 : 타겟값은 **반드시 정규분포를 가져야 함. 이를 위해 주로 로그 변환을 적용**

피처값 변환 :

- StandardScaler 클래스 적용(평균 0, 분산 1)하거나, MinMaxScaler 클래스(최솟값 0, 최댓값 1)로 정규화 -> 예측 성능의 향상을 기대하기 어려운 경우가 많다.
- 스케일링/정규화를 수행한 데이터 세트에 **다시 다항 특성을 적용하여 변환** 보통 1번 방법을 통해 예측 성능에 향상이 없을 경우 이와 같은 방법을 이용 -> Feature의 개수가 매우 많을 경우에는 과적합이 발생할 수 있다.
- **로그 변환(가장 자주 쓰이는 방법)**

선형 회귀의 데이터 인코딩은 일반적으로 Label Encoding(Category : 0, 1, 2, 3 숫자가 아니라 전자레인지, 냉장고, …)이 아니라 **one-hot Encoding**(값의 상관도 없애기 위해서 사용)을 적용한다.

로지스틱 회귀 예측

시그모이드 함수 $y = \frac{1}{1+e^{-x}}$

: 0과 1의 이진 분류에서 사용된다.

0.5를 대칭점으로해서 0.5보다 크면 1로 간주하고, 0.5보다 작으면 0으로 간주한다.

이 0.5라는 값을 조절할 수도 있음(hyperparameter의 일종)

로지스틱 회귀는 0과 1을 예측하기에 단순 회귀식은 의미가 없습니다.
하지만 Odds(성공확률/실패확률)을 통해 선형 회귀식에 확률을 적용할 수 있습니다.

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

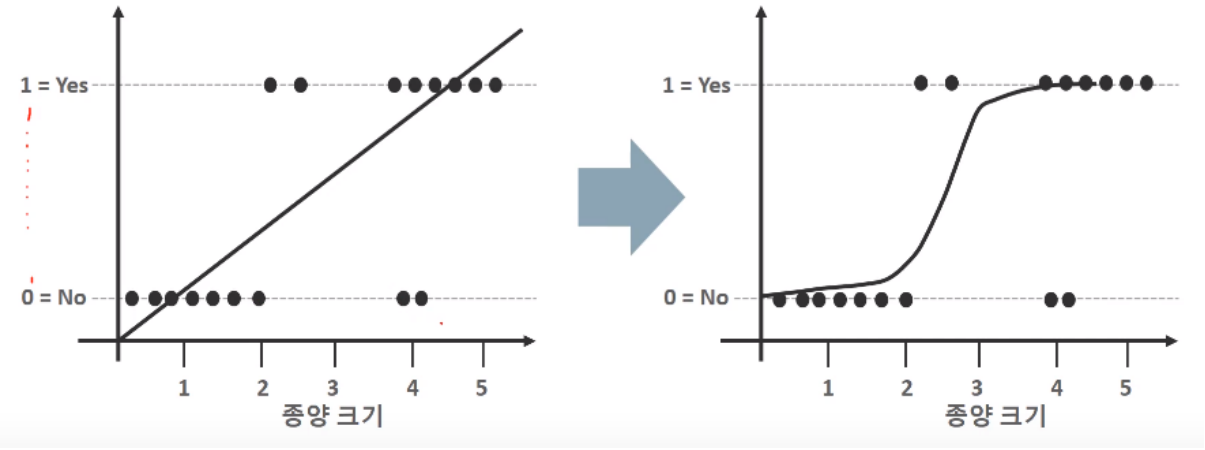
하지만 확률p의 범위가 (0, 1)이므로 선형 회귀의 반환값인 (-무한대, + 무한대)에
대응하기 위하여 로그 변환을 수행하고 이 값에 대한 선형 회귀를 적용합니다.

$$Log(Odds(p)) = w1x + w0$$

해당 식을 데이터 값 x의 확률 p로 정리하면 아래와 같습니다.

$$p(x) = \frac{1}{1+e^{-(w1x+w0)}}$$

학습을 통해 sigmoid의 w를 최적화하는 값을 찾는다. -> 이를 기반으로 예측



왼쪽처럼 선형 회귀로도 유추 가능하지만 억지스럽다.(무리가 있다.)

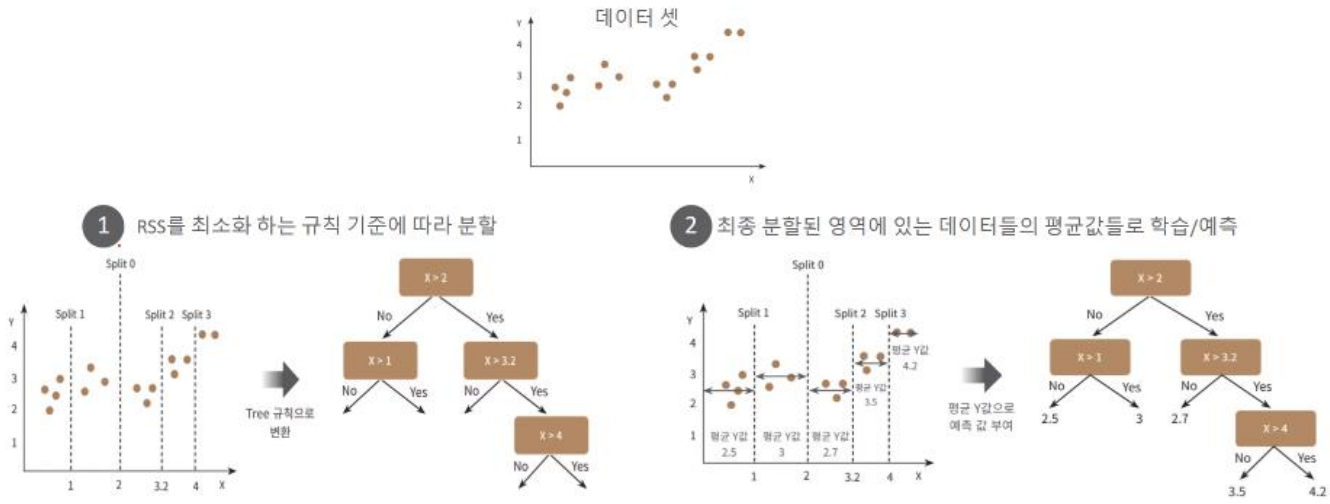
오른쪽처럼 선형 회귀에 sigmoid를 이용하여 Logistic Regression로 약 2.7보다 작으면 종양이 아니고, 2.7보다 크면 종양이라고 **분류**에서도 잘 동작시킬 수 있다

- 로지스틱 회귀는 가볍고 빠르지만, 이진 분류 예측 성능 또한 뛰어나다. 이 때문에 로지스틱 회귀를 이진 분류의 기본 모델로 사용하는 경우가 많다. 또한, 로지스틱 회귀는 희소한 데이터 세트 분류에도 뛰어난 성능을 보여 텍스트 분류에도 자주 사용된다.
- 사이킷런은 LogisticRegression 클래스로 로지스틱 회귀를 구현한다. 주요 hyperparameter로 penalty와 C가 있는데,
  - penalty는 규제의 유형을 설정하며 l1 -> L1규제, l2 -> L2 규제이다. default는 L2 규제이다.
  - C는 규제 강도를 조절하는 alpha의 역수이다. C 값이 작을수록 규제 강도가 크다.

회귀 트리

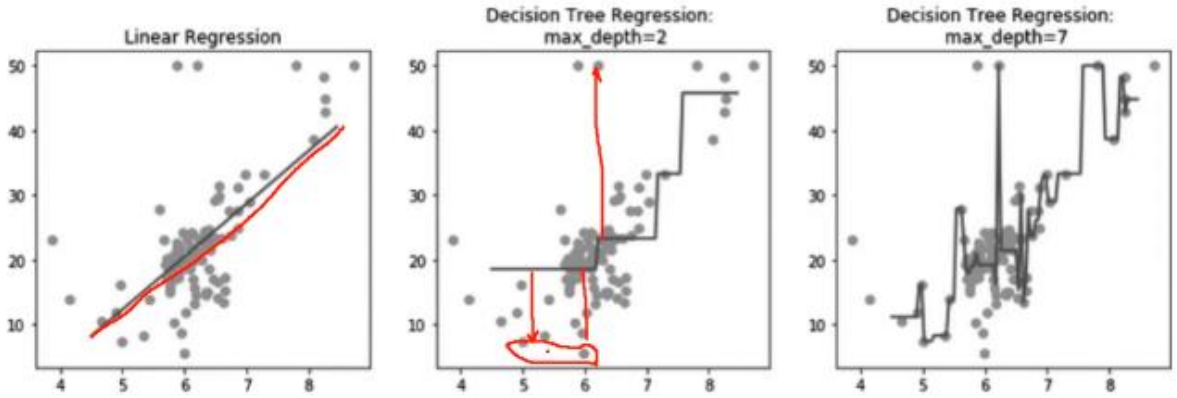
- 사이킷런의 결정 트리 및 결정 트리 기반 앙상블 알고리즘은 분류뿐만 아니라 회귀도 가능하다.
- 이는 트리가 CART(Classification And Regression Tree이기 때문이다.
- CART 회귀 트리는 분류와 유사하게 분할하며, 분할 기준은 **RSS(SSE)가 최소가 될 수 있는 기준을 찾아서 분할된다.**
- 최종 분할이 완료된 후에 각 분할 영역에 있는 데이터 결정 값들의 **평균 값**으로 학습/예측한다.

CART : Classification And Regression Trees



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

회귀 트리 역시 복잡한 트리 구조를 가질 경우 오버피팅하기 쉬우므로 **트리의 크기**와 **노드 개수의 제한** 등의 방법을 통해 **오버피팅을 개선**할 수 있다



max_depth = 7인 경우에 이상치에 맞춰진 overfitting 이 발생한 것을 확인할 수 있다.

출처 :

인프런 머신러닝 완벽가이드

[1] Low, High Variance, Bias (http://scott.fortmann-roe.com/docs/BiasVariance.html)

[2] Bias Variance Trade off (http://scott.fortmann-roe.com/docs/BiasVariance.html)