

# 5장. 회귀

## 01 회귀 소개

- 정의: 여러 개의 독립변수와 한 개의 종속변수 간의 상관 관계를 모델링하는 기법
  - 선형 회귀식은 종속변수, 독립변수, 독립변수의 값에 영향을 미치는 회귀 계수로 구성
  - 머신 러닝 회귀 예측의 핵심은 주어진 피처와 결정 값 데이터 기반 학습을 통해 최적의 회귀 계수를 찾아내는 것

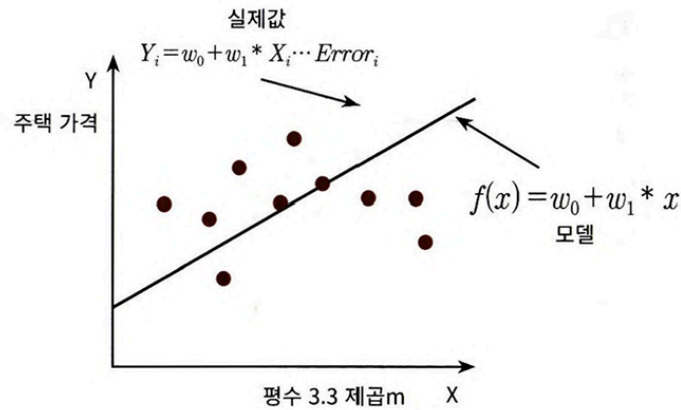
- 유형

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

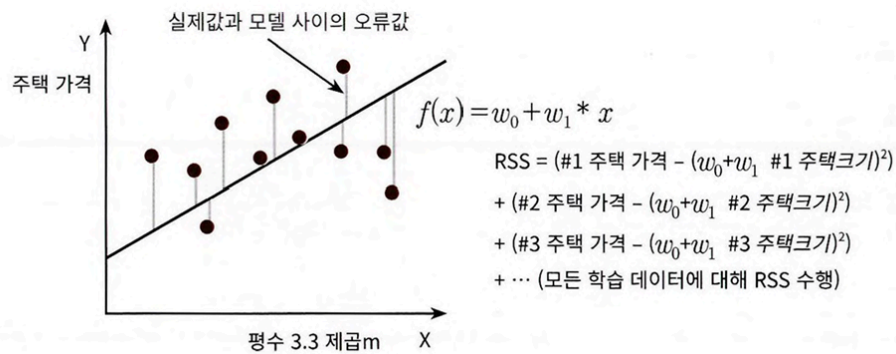
- 실제 값과 예측값의 차이를 최소화하는 직선형 회귀선을 최적화하는 선형 회귀가 가장 많이 사용됨
- 선형 회귀는 규제에 방법에 따라 다시 별도의 유형으로 나뉨

## 02 단순 선형 회귀를 통한 회귀 이해

- 정의: 독립변수도 하나, 종속변수도 하나인 선형 회귀



- 잔차: 실제 값과 회귀 모델의 차이에 따른 오류값
- 최적의 회귀 모델 == 잔차합이 최소가 되는 모델



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

( $i$ 는 1부터 학습 데이터의 총 건수  $N$ 까지)

- RSS: 오류 값의 제곱을 구하여 더해 오류 합을 구하는 방식
- 회귀에서 RSS은 비용(Cost)
- $w$ 변수(회귀 계수)로 구성된 RSS를 비용 함수라고 함
- 회귀 알고리즘은 비용 함수가 반환하는 값을 감소시키고, 최종적으로 더 이상 감소하지 않는 최소의 오류 값 구하는 것!

## 03 비용 최소화하기 - 경사 하강법(Gradient Descent)

- 정의: 점진적으로 반복적인 계산을 통해 W 파라미터 값을 업데이트하며 오류값을 최소화하는 W 파라미터 구하는 방식
- 핵심: "어떻게 하면 오류가 작아지는 방향으로 W 값을 보정할 수 있을까?"
- 방법:

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

일 때,  $w_0$ 와  $w_1$ 값을 순차적으로 편미분 한 식인

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

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

를 업데이트 하며 최소가 되는 값을 찾음

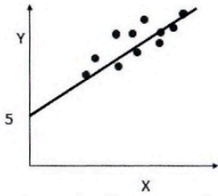
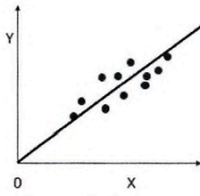
정리하자면 다음과 같다.

- Step 1:  $w_1, w_0$ 를 임의의 값으로 설정하고 첫 비용 함수의 값을 계산합니다.
- Step 2:  $w_1$ 을  $w_1 + \eta \frac{2}{N} \sum_{i=1}^N x_i * (\text{실제값}_i - \text{예측값}_i)$ ,  $w_0$ 을  $w_0 + \eta \frac{2}{N} \sum_{i=1}^N (\text{실제값}_i - \text{예측값}_i)$ 으로 업데이트한 후 다시 비용 함수의 값을 계산합니다.
- Step 3: 비용 함수가 감소하는 방향으로 주어진 횟수만큼 Step 2를 반복하면서  $w_1$ 과  $w_0$ 를 계속 업데이트합니다.

## 04 사이킷런 LinearRegression을 이용한 보스턴 주택 가격 예측

LinearRegression 클래스 - Ordinary Least Squares

- 예측값과 실제 값의 RSS를 최소화해 OLS(Ordinary Least Squares) 추정 방식으로 구현한 클래스
- fit() 메서드로 X, y 배열을 입력받아 회귀 계수인 W를 coef\_속성에 저장

입력 파라미터	<p><b>fit_intercept</b>: 불린 값으로, 디폴트는 True입니다. Intercept(절편) 값을 계산할 것인지 말지를 지정합니다. 만일 False로 지정하면 intercept가 사용되지 않고 0으로 지정됩니다.</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>fit_intercept=True</p>  </div> <div style="text-align: center;"> <p>fit_intercept=False</p>  </div> </div>
	<p><b>normalize</b>: 불린 값으로 디폴트는 False입니다. fit_intercept가 False인 경우에는 이 파라미터가 무시됩니다. 만일 True이면 회귀를 수행하기 전에 입력 데이터 세트를 정규화합니다.</p>
속성	<p><b>coef_</b>: fit() 메서드를 수행했을 때 회귀 계수가 배열 형태로 저장하는 속성. Shape는 (Target 값 개수, 피쳐 개수).</p> <p><b>intercept_</b>: intercept 값</p>

- 다중 공선성: 입력 피쳐의 독립성에 많은 영향을 받아 피쳐 간 상관관계가 매우 높은 경우 분산이 매우 커져 오류에 매우 민감해짐

## 회귀 평가 지표

- 실제값과 회귀 예측값의 차이 값을 기반으로 함
- 오류의 절댓값 평균이나 제곱, 또는 제곱한 뒤 루트를 씌운 평균값을 구함

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 <sup>2</sup>	분산 기반으로 예측 성능을 평가합니다. 실제 값의 분산 대비 예측값의 분산 비율을 지표로 하며, 1에 가까울수록 예측 정확도가 높습니다.	$R^2 = \frac{\text{예측값 Variance}}{\text{실제값 Variance}}$

- 사이킷런은 RMSE를 제공하지 않아 MSE에 제곱근을 씌워 계산하는 함수를 직접 만들어야 함
- 사이킷런 Scoring함수는 값이 클 수록 좋은 평가 결과로 평가하기 때문에, 'neg\_mean\_absolute\_error'를 적용해 음수값을 반환함
- 즉, neg\_mean\_abosolute\_error는  $-1 * \text{metrics.mean\_absolute\_error}()$ 를 의미

## LinearRegression을 이용해 보스턴 주택 가격 회귀 구현

# 05 다항 회귀와 과(대)적합/과소적합 이해

## 다항 회귀 이해

- 정의: 독립변수의 단항식이 아닌 2차 이상의 다항식으로 표현되는 회귀를 다항 회귀라고 함
- 다항 회귀는 선형 회귀(선형/비선형 회귀를 나누는 기준은 회귀 계수의 선형/비선형 여부에 따름)
- 사이킷런은 다항 회귀를 위한 클래스를 명시적으로 제공하지 않아 비선형 함수를 선형 모델에 적용시키는 방법을 사용해 구현
- PolynomialFeatures 클래스를 통해 피처를 Polynomial 피처로 변환한 후, degree 파라미터를 통해 입력 받은 단항식 피처를 degree에 해당하는 다항식 피처로 변환함

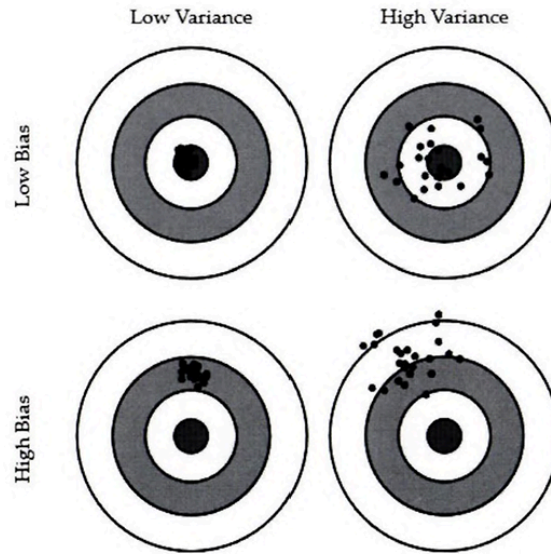
## 다항 회귀를 이용한 과소적합 및 과적합 이해

- 다항식의 차수가 높아질 수록 복잡한 피처 간 관계까지 모델링이 가능하지만, 학습 데이터에만 맞춘 학습이 이루어져 과적합의 문제가 발생함
- 학습 데이터의 패턴을 잘 반영하면서도 복잡하지 않은 균형 잡힌 모델이 좋은 예측 모델이다.

## 편향-분산 트레이드오프(Bias-Variance Trade off)

- 고편향성(High Bias): 지나치게 단순화되어 지나치게 한 방향으로 치우친 모델

- 고분산성(High Variance): 매우 복잡해 지나치게 높은 변동성을 가지게 된 모델



- 일반적으로 편향과 분산은 한쪽이 높으면 한쪽이 낮아지는 경향이 있음
- 높은 편향/낮은 분산에서는 과소적합되기, 낮은 편향/높은 분산에서는 과적합되기 쉬움

## 06 규제 선형 모델 - 릿지, 라쏘, 엘라스틱넷

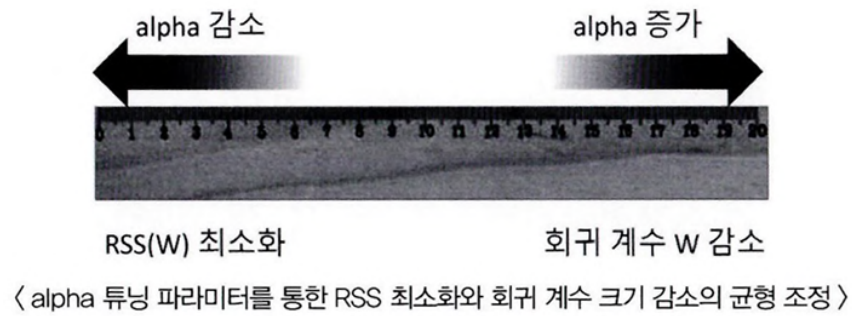
### 규제 선형 모델의 개요

- 적절히 데이터에 적합하면서도 회귀 계수가 기하급수적으로 커지는 것을 제어할 수 있어야 함
- 학습 데이터의 잔차 오류 값을 최소로 하는 RSS 최소화 방법과 과적합 방지를 위한 회귀 계수 값이 커지지 않도록 균형을 이뤄야 함
- 회귀 계수 값의 크기를 제어하는 식

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

alpha 값을 크게 하면 비용 함수는 회귀 계수 W값을 작게 해 비용 함수 목표 달성

alpha 값을 작게 하면 회귀 계수 W의 값이 커져도 어느 정도 상쇄가 가능해 학습 데이터 적합을 개선



- 이처럼 alpha 값으로 페널티를 부여해 회귀 계수 값의 크기를 감소시켜 과적합을 개선하는 방식을 규제라 함

L1 방식	W의 절댓값에 대해 페널티 부여 ⇒ 라쏘 회귀
L2 방식	W의 제곱에 대해 페널티 부여 ⇒ 릿지 회귀

## 릿지 회귀

- 사이킷런 Ridge 클래스를 통해 구현
- alpha L2 규제 계수에 해당하는 alpha를 주요 생성 파라미터로 가짐

## 라쏘 회귀

- W의 절댓값에 페널티를 부여하는 L1 규제를 선형 회귀에 적용
- 불필요한 회귀 계수를 급격하게 감소시켜 0으로 만들고 제거함
- 적절한 피처만 회귀에 포함시키는 피처 선택의 특성을 가짐
- 사이킷런의 Lasso 클래스를 통해 라쏘 회귀 구현
- alpha L1 규제 계수에 해당하는 alpha를 주요 생성 파라미터로 가짐

## 엘라스틱넷 회귀

- L2 규제와 L1 규제를 결합한 회귀
- alpha 값에 따라 회귀 계수의 값이 급격히 변동하는 라쏘 회귀의 특성을 완화하기 위해 L2 규제를 추가
- L2 규제와 L1 규제의 결합으로 수행 시간이 상대적으로 오래 걸린다는 단점
- 사이킷런의 ElasticNet 클래스를 통해 구현

- 주요 생성 파라미터로 alpha와 l1\_ratio를 가짐

## 선형 회귀 모델을 위한 데이터 변환

- 피처와 타깃값 간 선형의 관계가 있다고 가정하고 최적의 선형함수를 찾아내 결괏값 예측
- 선형 회귀 모델은 피처값과 타깃값의 분포가 정규 분포 형태인 것을 선호함
- 그렇지 않을 때에는 왜곡으로 인해 예측 성능에 부정적인 영향을 미치기 때문에 선형 회귀 모델을 적용하기 전 먼저 데이터에 대한 스케일링/정규화 작업을 수행함
- 다음은 사이킷런을 이용해 피처 데이터 세트에 적용하는 변환 작업

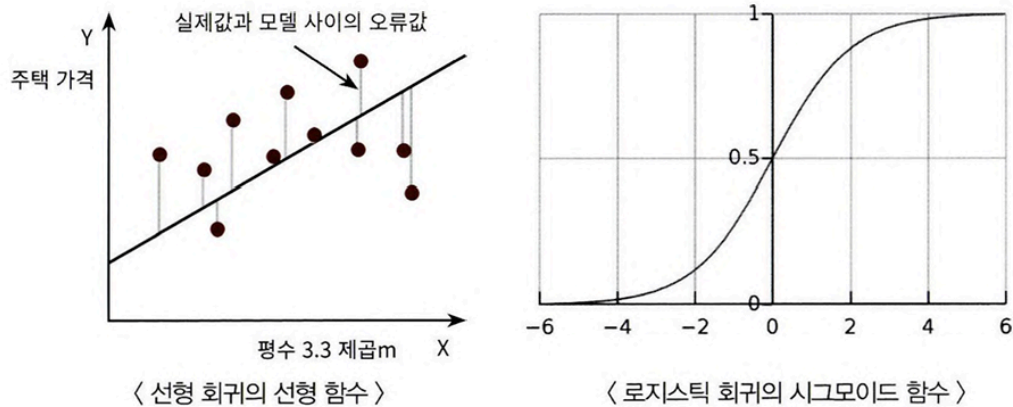
1. StandardScaler 클래스를 이용해 평균이 0, 분산이 1인 표준 정규 분포를 가진 데이터 세트로 변환하거나 MinMaxScaler 클래스를 이용해 최솟값이 0이고 최댓값이 1인 값으로 정규화를 수행합니다.
2. 스케일링/정규화를 수행한 데이터 세트에 다시 다항 특성을 적용하여 변환하는 방법입니다. 보통 1번 방법을 통해 예측 성능에 향상이 없을 경우 이와 같은 방법을 적용합니다.
3. 원래 값에 log 함수를 적용하면 보다 정규 분포에 가까운 형태로 값이 분포됩니다. 이러한 변환을 로그 변환(Log Transformation)이라고 부릅니다. 로그 변환은 매우 유용한 변환이며, 실제로 선형 회귀에서는 앞에서 소개한 1, 2번 방법보다 로그 변환이 훨씬 많이 사용되는 변환 방법입니다. 왜냐하면 1번 방법의 경우 예측 성능 향상을 크게 기대하기 어려운 경우가 많으며 2번 방법의 경우 피처의 개수가 매우 많을 경우에는 다항 변환으로 생성되는 피처의 개수가 기하급수로 늘어나서 과적합의 이슈가 발생할 수 있기 때문입니다.

타깃값의 경우 일반적으로 로그 변환을 적용함

## 07 로지스틱 회귀

- 선형 회귀 방식을 분류에 적용한 알고리즘
- 학습을 통해 시그모이드 함수 최적선을 찾고, 시그모이드 함수의 반환 값을 확률로 간주해 확률에 따라 분류 결정





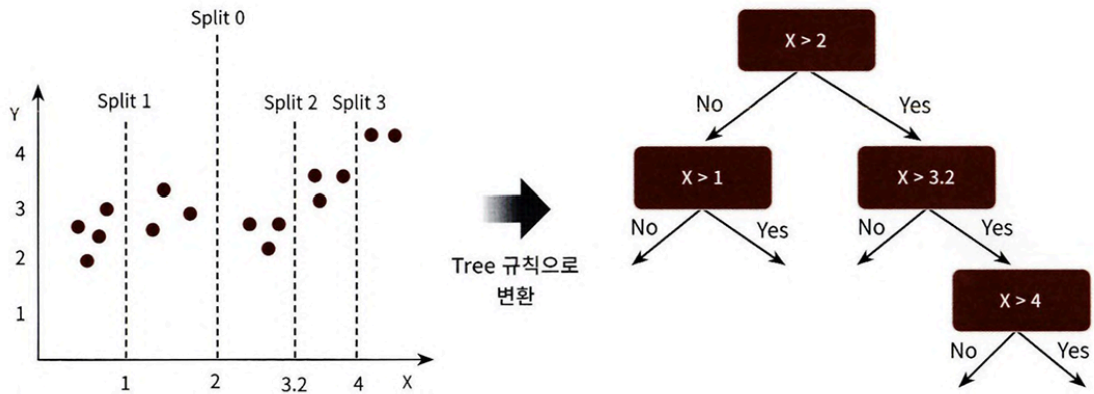
- 사이킷런의 LogisticRegression 클래스를 사용
- 회귀 계수 최적화를 위해 경사 하강법 외 다양한 방안 선택 가능
  - lbfgs: 사이킷런 버전 0.22부터 solver의 기본 설정값입니다. 메모리 공간을 절약할 수 있고, CPU 코어 수가 많다면 최적화를 병렬로 수행할 수 있습니다.
  - liblinear: 사이킷런 버전 0.21까지에서 solver의 기본 설정값입니다. 다차원이고 작은 데이터 세트에서 효과적으로 동작하지만 국소 최적화(Local Minimum)에 이슈가 있고, 병렬로 최적화할 수 없습니다.
  - newton-cg: 좀 더 정교한 최적화를 가능하게 하지만, 대용량의 데이터에서 속도가 많이 느려집니다.
  - sag: Stochastic Average Gradient로서 경사 하강법 기반의 최적화를 적용합니다. 대용량의 데이터에서 빠르게 최적화합니다.
  - saga: sag과 유사한 최적화 방식이며 L1 정규화를 가능하게 해줍니다.

일반적으로 lbfgs 또는 liblinear를 선택

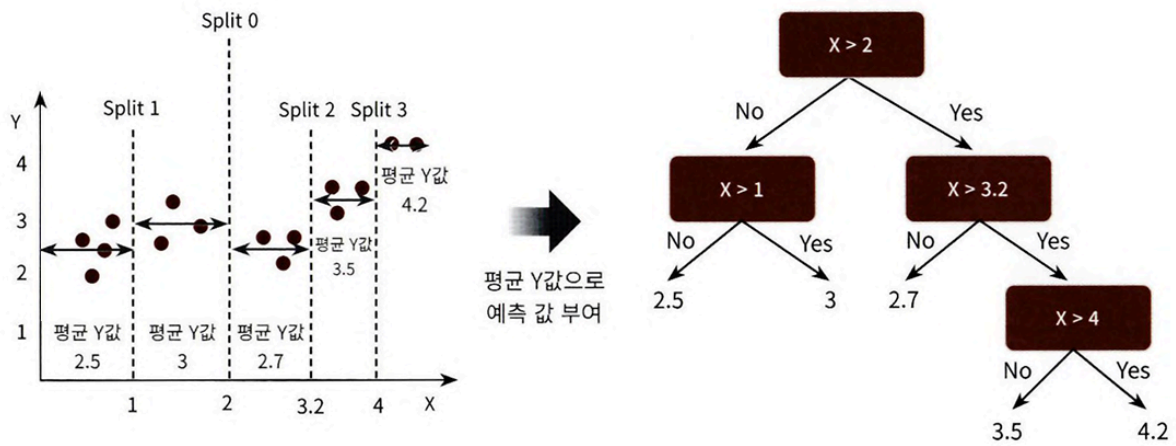
- 로지스틱 회귀는 가볍고 빠르면서 이진 분류 예측 성능 및 희소한 데이터 세트 분류에 뛰어남
- 이진 분류의 기본 모델, 텍스트 분류에서 자주 사용됨

## 08 회귀 트리

- 회귀 함수를 기반으로 하지 않고, 회귀를 위한 트리를 생성해 회귀 예측을 함
- 리프 노트에 속한 데이터 값의 평균값을 구해 회귀 예측값을 계산



X 피처를 결정 트리 기반으로 분할해 X값의 균일도를 반영한 지니 계수에 따라 왼쪽 그림과 같이 분할



리프 노드에 소속된 데이터 값의 평균값을 구해 최종적으로 리프 노드에 결정 값으로 할당