회귀

작성일시	@2024년 11월 9일 오후 1:36
강의 번호	Euron
유형	스터디 그룹
복습	☐

1. 소개

- 회귀: 여러 개의 독립변수와 한 개의 종속변수 간의 상관관계 모델링 기법

$$Y = W_1 * X_1 + W_2 * X_2 + ... + W_n * X_n$$
$$X_1, X_2 \ ... \ X_n : \text{독립변수}$$
$$W_1, W_2 \ ... \ W_n : \text{회귀계수}$$

독립변수	피처
종속변수	결정 값 (레이블)

- [머신러닝] 학습 → 최적의 회귀계수 찾기
- 회귀 분류

독립변수 개수	회귀계수 결합
단일 회귀 (1개)	선형 회귀 (직선형)
다중 회귀 (2개 이상)	비선형 회귀

- 지도학습

	분류	회귀
예측값	이산형 클래스	연속형 숫자

선형회귀

- 실제 값과 예측값 차이 (오류의 제곱) 최소화하는 회귀계수 찾기
- 규제: 회귀 계수에 패널티 값 적용 ⇒ 과적합 문제 해결

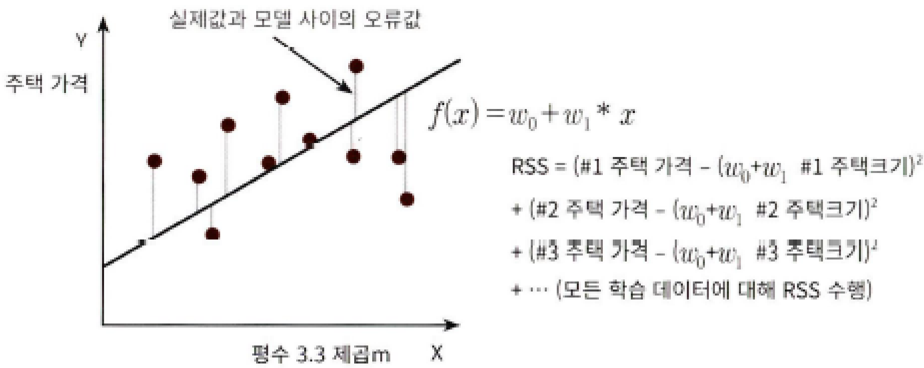
일반 선형 회귀	RSS 최소화	규제 X

릿지	RSS 최소화	L2 규제
라쏘	RSS 최소화	L1 규제
엘라스틱넷	피처가 많은 데이터셋에 적용됨	L1, L2 규제 결합
로지스틱 회귀	분류에 사용되는 선형 모델	-

- L2 규제: 큰 회귀계수 값의 예측 영향도 감소하기 위해 회귀 계수값을 작게 만드는 모델
- L1 규제: 예측 영향력이 작은 피처의 회귀 계수를 0으로 ⇒ 예측 시 해당 피처가 선택되지 않게 함 (피처 선택 기능)

2. 단순 선형 회귀

- 단순 선형 회귀: 독립변수 1개, 종속변수 1개
- 잔차: 실제값과 회귀 모델의 차이로 인한 오류 값
  - 최적의 회귀 모델 = 전체 데이터의 **잔차 합 최소**인 모델
  - (+/-) 오류 합 → 오류 합이 줄어들 수 있음
    ⇒ Mean Absolute Error (절댓값의 합), RSS (오류 값 제곱의 합)



- 비용 함수

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

[i는 1부터 학습 데이터의 총 건수 N까지]

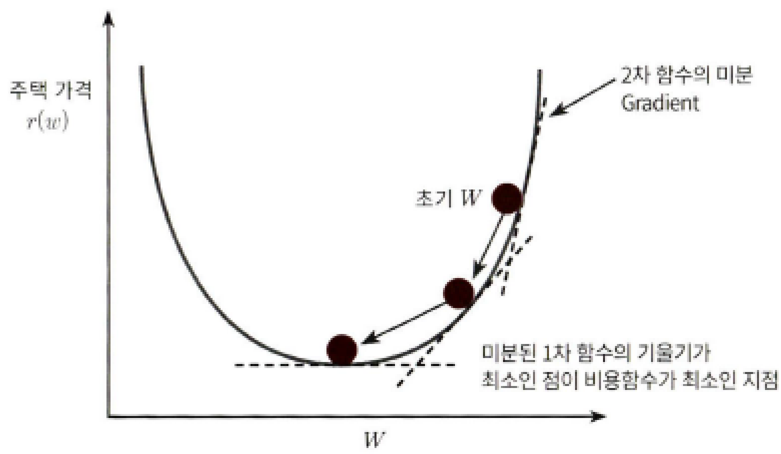
  - 학습을 통해 RSS (= 비용, 오류 값) 감소 → 최소 오류 값

- 손실 함수 (loss function)

3. 비용 최소화하기 - 경사하강법

경사 하강법

- 고차원 방정식에서 비용 최소화하는 회귀 계수(W parameter) 구하는 방법
- 오류를 줄이는 방향으로 W parameter 업데이트 → 최소 비용 (오류값), 최적 파라미터 (W 값) 반환



  - ex) 비용 함수가 2차 함수일 때
    - 최초 W에서부터 미분 값(직선 기울기) 감소하는 방향으로 W를 반복적 업데이트
- 경사 하강법 과정

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

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

1. w0, w1 임의로 설정한다

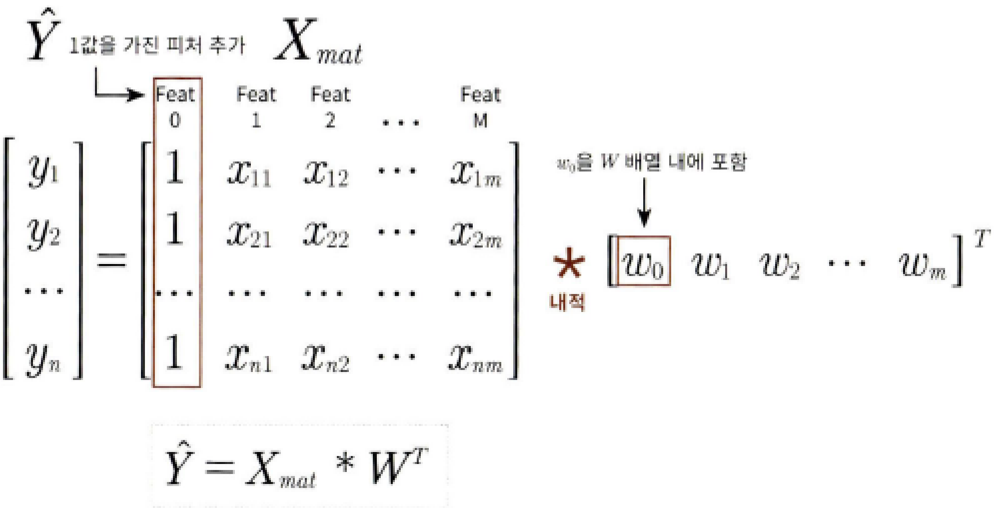
2. w0, w1에 대해 편미분한 값에 학습률을 곱하여 w0, w1에 더한다. (파라미터 업데이트)
   비용 함수 R(w) 값을 계산한다
   - 학습률: 편미분 값이 너무 클 수 있기 때문에 곱하는 보정계수
3. 비용 함수 값 감소 → 2번 반복한다

   비용 함수 값 감소하지 않음 → w0, w1 구하고 반복 중지한다

- 모든 학습 데이터에 대해 반복적으로 학습, 업데이트 → 시간이 오래 걸림

  ⇒ **확률적 경사 하강법 (SGD)** 이용

- 다중 선형 회귀



확률적 경사 하강법 (SGD)

- 일부 데이터로 w 업데이트 → 빠른 수행 속도
- 대용량의 데이터일 때:

  확률적 경사 하강법, 미니 배치 확률적 경사 하강법 이용
- SGD, 경사 하강법 두 가지 방법으로 구했을 때 예측 성능에 큰 차이가 없음

  ⇒ 대용량 데이터일 때 SGD 사용!

4. 사이킷런 LinearRegression 이용한 보스턴 주택 가격 예측

LinearRegression 클래스 - OLS (Ordinary Least Squares)

- RSS 최소화해서 OLS 추정 방식으로 구현한 클래스
    - OLS: 오차 제곱 최소화하는 파라미터 추정

```
class sklearn.linear_model.LinearRegression(fit_intercept=Tru
```

- 입력 파라미터

fit_intercept	- boolean (디폴트 True) - fit_intercept=False: intercept 사용되지 않음
normalize	- boolean (디폴트 False) - fit_intercept =False → 파라미터 무시 - normalize =True → 데이터셋 정규화

- 속성

coef_	- fit() 메서드 수행 시, 회귀 계수가 배열 형태로 저장 - Shape: (Target 수, 피처 수)
intercept_	절편 값

회귀 평가 지표

평가 지표	설명
MAE	|실제 값-예측값| 평균값
MSE	(실제 값-예측값)^2 평균값
RMSE	MSE는 실제 오류 평균보다 커짐 → 루트값 구함
R^2	- 분산 기반 예측 성능 평가 - 1에 가까울수록 예측 정확도 높음

수식
$MAE = \frac{1}{n}\sum_{i=1}^{n}\|Yi - \hat{Y}i\|$
$MSE = \frac{1}{n}\sum_{i=1}^{n}(Yi - \hat{Y}i)^2$
$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Yi - \hat{Y}i)^2}$
$R^2 = \frac{\text{예측값 } Variance}{\text{실제값 } Variance}$

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'

- 사이킷런에서 RMSE 제공하지 않음 (→ MSE에 제곱근 씌워서 계산)
- ** 유의하기 **
    - Scoring 함수는 score 값이 클수록 좋은 평가 결과로 인식함
    - 회귀 평가 지표는 오류값을 기반으로 함

      ⇒ (-1) * (평가 지표 값) 보정이 필요!
    - ex) neg_mean_absolute_error : (-1)*metrics.mean_absolute_error

사이킷런 LinearRegression 이용한 보스턴 주택 가격 회귀 구현

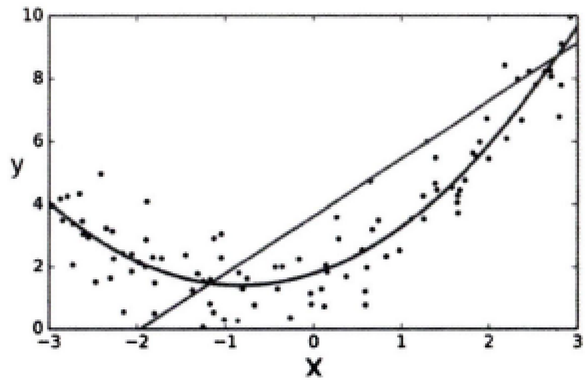
- CRIM: 지역별 범죄 발생률
- ZN: 25,000평방피트를 초과하는 거주 지역의 비율
- INDUS: 비상업 지역 넓이 비율
- CHAS: 찰스강에 대한 더미 변수(강의 경계에 위치한 경우는 1, 아니면 0)
- NOX: 일산화질소 농도
- RM: 거주할 수 있는 방 개수
- AGE: 1940년 이전에 건축된 소유 주택의 비율
- DIS: 5개 주요 고용센터까지의 가중 거리
- RAD: 고속도로 접근 용이도
- TAX: 10,000달러당 재산세율
- PTRATIO: 지역의 교사와 학생 수 비율
- B: 지역의 흑인 거주 비율
- LSTAT: 하위 계층의 비율
- MEDV: 본인 소유의 주택 가격(중앙값)

5. 다항 회귀와 과(대)적합/과소적합

다항 회귀 이해

- 다항 회귀: 단항식(일차 방정식 형태)이 아닌 2차, 3차 방정식으로 표현되는 것

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



〈 주어진 데이터 세트에서 다항 회귀가 더 효과적임 〉

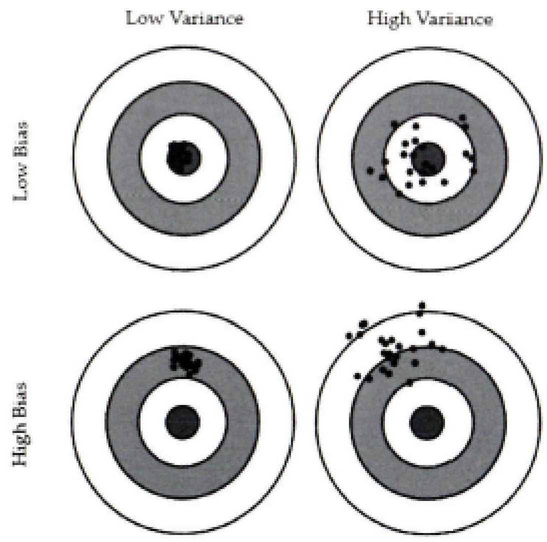
- ** 유의 **
  - 다항 회귀는 **선형 회귀**
  - 선형/ 비선형 회귀 구분 기준: **회귀 계수가 선형/ 비선형인지**
  - 독립변수의 선형/ 비선형 여부와 무관!

다항 회귀 이용한 과소적합, 과적합 이해

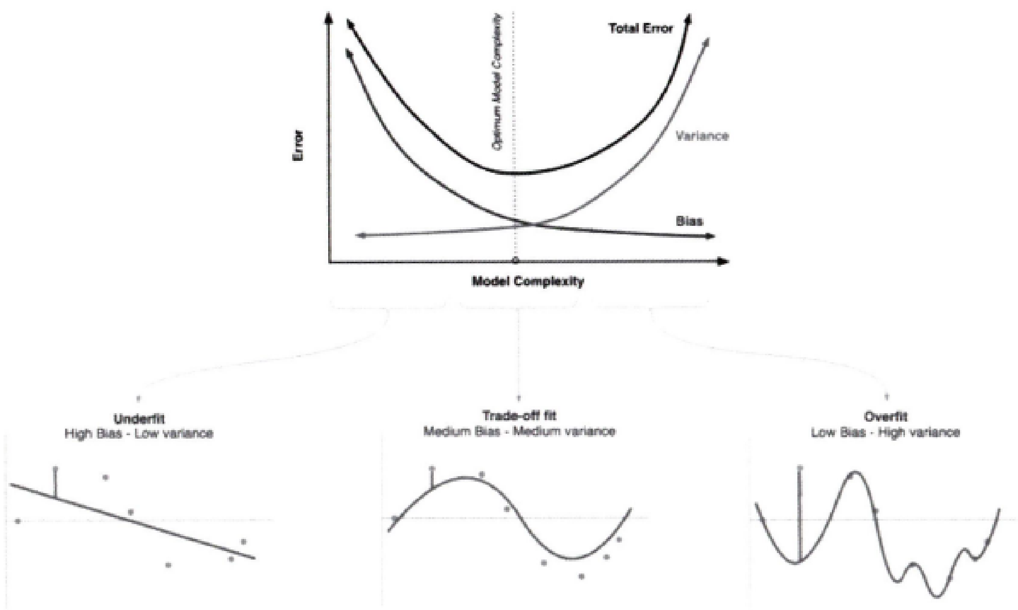
- 다항식 차수(degree) 높일 수록
  - 복잡한 관계 모델링 가능
  - 학습 데이터에만 적합한 학습 이루어짐 (**과적합 문제**)
- 좋은 예측 모델 : 학습 데이터의 패턴 반영 & 복잡하지 않은 균형 잡힌 모델

편향-분산 트레이드 오프

- 고편향성 (High Bias): 지나치게 한 방향으로 치우침
- 고분산성 (High Variance): 개별 데이터의 특성 반영 → 매우 복잡해짐 (지나치게 높은 변동성)
- 저편향 / 저분산이 가장 좋음



〈 편향과 분산의 고/저에 따른 표현.
http://scott.fortmann-roe.com/docs/BiasVariance.html에서 발췌 〉



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

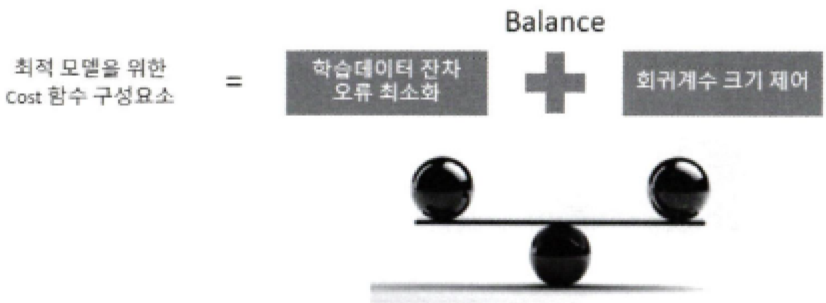
- 편향 & 분산: 하나가 높으면 다른 하나가 낮아지는 경향
    - 고편향 / 저분산 (과소적합)
    - 저편향 / 고분산 (과대적합)
- 골디락스 지점: 편향 낮추고 분산 높이면서 전체 오류가 가장 낮아지는 지점
- [머신러닝] **오류 cost 값이 가장 낮아지는** 모델 구축

6. 규제 선형 모델 - 릿지, 라쏘, 엘라스틱넷

규제 선형 모델 개요

- RSS 최소화 → 학습 데이터에 지나치게 적합, *회귀 계수 커짐* (변동성 증가) → 예측 성능 저하

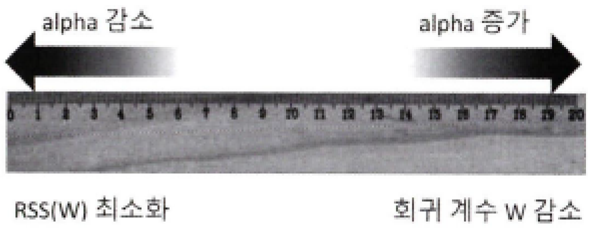
  ⇒ 회귀 계수 커지는걸 막자



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

- 규제: 비용 함수에 페널티(alpha) 부여 → 회귀 계수 값 크기 감소 ⇒ 과적합 개선
  - L2 규제
    - W 제곱에 페널티(alpha) 부여하는 방식
  - L1 규제
    - $|W|$ 에 페널티(alpha) 부여하는 방식
    - 영향력이 크지 않은 회귀 계수 값 =0 으로 변환

- alpha = 0인 경우는 W가 커도 $\text{alpha} * \|W\|_2^2$가 0이 되어 비용 함수는 Min(RSS(W))
- alpha = 무한대인 경우 $\text{alpha} * \|W\|_2^2$도 무한대가 되므로 비용 함수는 W를 0에 가깝게 최소화 해야 함.



〈 alpha 튜닝 파라미터를 통한 RSS 최소화와 회귀 계수 크기 감소의 균형 조정 〉

릿지 회귀

- 주요 생성 파라미터: alpha (L2 규제 계수)

라쏘 회귀

- L1 규제 적용함
- 불필요한 회귀 계수 급격하게 감소 → **피처 선택**(제거)하는 특성
- 상관관계 높은 피처일 때 → 중요 피처만 선택하고 나머지 피처들은 제거(회귀 계수=0)

  ⇒ *alpha 값에 따라 회귀 계수 값 급변*

엘라스틱넷 회귀

- L1, L2 규제 결합하여 적용함

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

- 특징
  1. 라쏘 회귀에서 alpha 값에 따라 회귀 계수 값 급변 → L2 규제 적용!
  2. 규제 결합 → 수행시간 오래 걸림
- ElasticNet 클래스를 통한 엘라스틱넷 회귀 구현
  - 파라미터: alpha, l1_ratio
  - 규제: a
  - 파라미터 값 = a+b

주요 생성 파라미터	alpha, l1_ratio
규제	a*L1 + b*L2 - a: L1 alpha 값 - b: L2 alpha 값
alpha 파라미터 값	a+b
l1_ratio 파라미터 값	a/(a+b) - l1_ratio=0: a=0(L2 규제) - l1_ratio=1: b=0(L1 규제)

선형 회귀 모델을 위한 데이터 변환

- 선형 회귀 모델의 피처값, 타깃값 분포가 정규분포 형태 선호

- 피처값, 타깃값 분포가 왜곡된 형태일 때 → 예측 성능에 부정적인 영향 미칠 수 있다

  ⇒ 스케일링 / 정규화 작업

1. 피처 데이터 세트
   a. StandardScaler 클래스: 표준 정규분포 가진 데이터 세트로 변환

      MinMaxScaler 클래스: 최솟값 0, 최댓값 1인 값으로 정규화

   b. (주로 a번 방법 실행해도 예측 성능이 향상되지 않을 때)

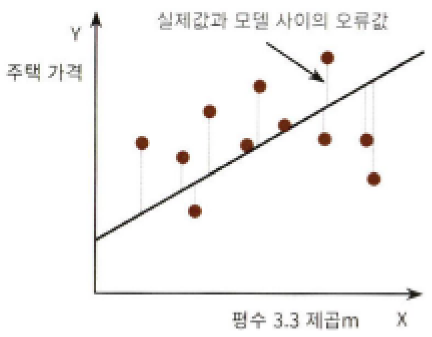
      스케일링/ 정규화된 데이터 세트에 다시 다항 특성 적용하여 변환 [피처 변환]

   c. 로그 변환: 원래 값에 log 함수 적용 (** 가장 많이 사용됨 **)
      - a번에서 예측 성능 향상을 크게 기대하기 어려움
      - b번에서 피처 개수 많을 때 다항 변환 시, 피처 개수가 급격하게 늘어남 → 과적합
2. 타킷 데이터 세트
   a. 로그 변환 적용

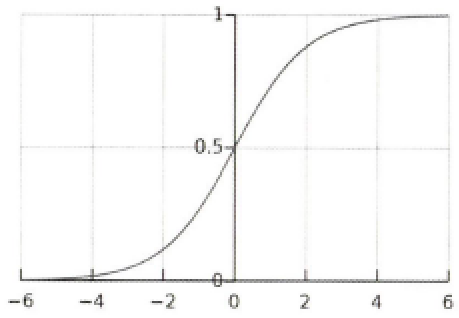
7. 로지스틱 회귀

- 분류에 사용되는 선형 회귀 계열 알고리즘
- 알고리즘
  1. 시그모이드 함수 최적선 찾는다
  2. 함수 반환 값을 확률로 간주한다
  3. 분류를 결정한다

  cf. 선형 회귀: 선형 함수의 회귀 최적선 찾는다



〈 선형 회귀의 선형 함수 〉

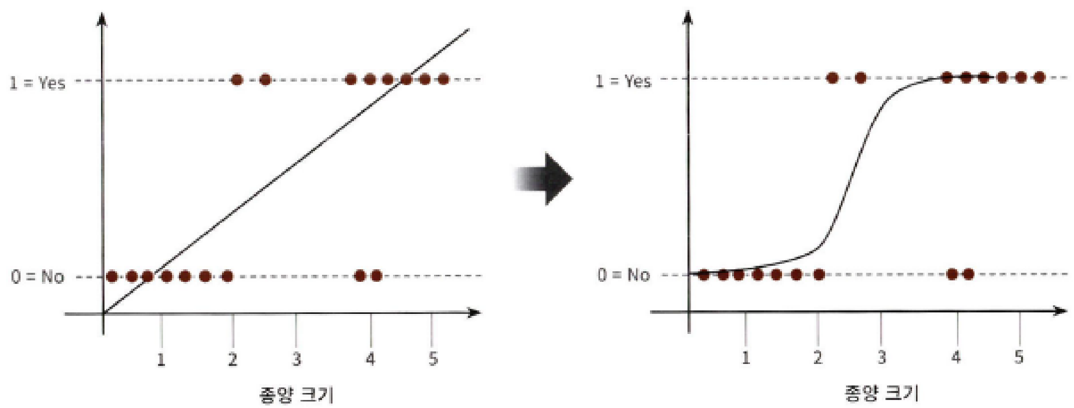


〈 로지스틱 회귀의 시그모이드 함수 〉

- 시그모이드 함수

$$y = \frac{1}{1 + e^{-x}}$$

  - x 값에 상관 없이 y 값은 항상 0~1



  - 시그모이드 함수를 이용하여 정확하게 0,1로 분류할 수 있음
    - 선형 회귀선: 0,1 제데로 분류하지 못함 (정확도 낮음)

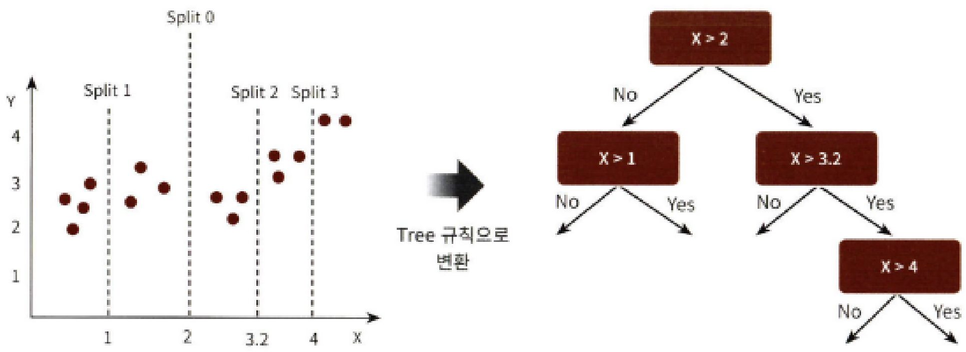
- 하이퍼 파라미터

파라미터명	설명
penalty	규제 유형 - l2: L2 규제 (디폴트) - l1: L1 규제
C	규제 강도 조절 (1/alpha) - 작을 수록 규제 강도 커짐

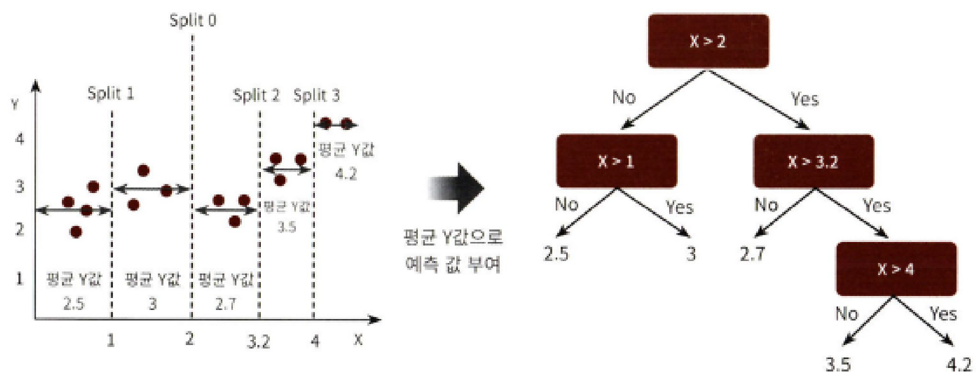
- 특징
  - 가볍고 빠름
  - 이진 분류 예측 성능 뛰어남
  - 텍스트 분류에서 자주 사용됨

8. 회귀 트리

- 회귀를 위한 트리 생성 → 회귀 예측
  - 회귀 트리: 리프 노드에 속한 데이터 평균값 → 회귀 예측값 계산
  - 회귀 트리 알고리즘
    1. X 피처 분할



    2. 리프 노드에 속한 데이터 값의 평균값을 리프 노드의 결정값으로 할당



- 트리 기반 회귀, 분류 Estimator

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