회귀 소개

- 현대 통계학을 떠받치고 있는 주요 기둥 중 하나
- **데이터 값이 평균과 같은 일정한 값으로 돌아가려는 경향**을 이용한 통계학 기법
- 여러 개의 독립변수와 한 개의 종속변수 간의 상관관계를 모델링하는 기법을 통칭
- ex) 다음과 같은 선형 회귀식이 있을 때 :

$$Y = W_1 * X_1 + W_2 * X_2 + W_3 * X_3 + \ldots + W_n * X_n$$

- Y : **종속변수**, X : **독립변수**, W : **회귀 계수** - 독립변수의 값에 영향을 미치는 값
- **최적의 회귀 계수를 찾아내는 것**이 회귀 예측의 핵심

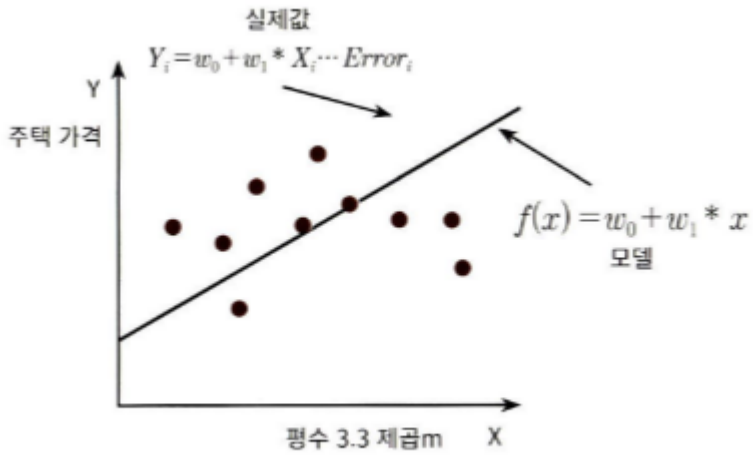
회귀의 유형

- 회귀 계수의 선형/비선형 여부, 독립변수의 개수, 종속변수의 개수에 따라 여러 유형으로 나눌 수 있음.
- 회귀 계수가 선형이냐 아니냐
  - 선형 : **선형 회귀**
  - 비선형 : **비선형 회귀**
- 독립변수의 개수
  - 한 개 : **단일 회귀**
  - 여러 개 : **다중 회귀**
- **선형 회귀**가 가장 많이 사용됨
  - 실제 값과 예측값의 차이를 최소화하는 직선형 회귀선을 최적화하는 방식
- **규제** 방법에 따라 별도의 유형으로 나뉠 수 있음
  - 선형 회귀의 과적합 문제를 해결하기 위해 회귀 계수에 페널티 값을 적용하는 것
- 대표적인 선형 회귀 모델
  - 일반 선형 회귀

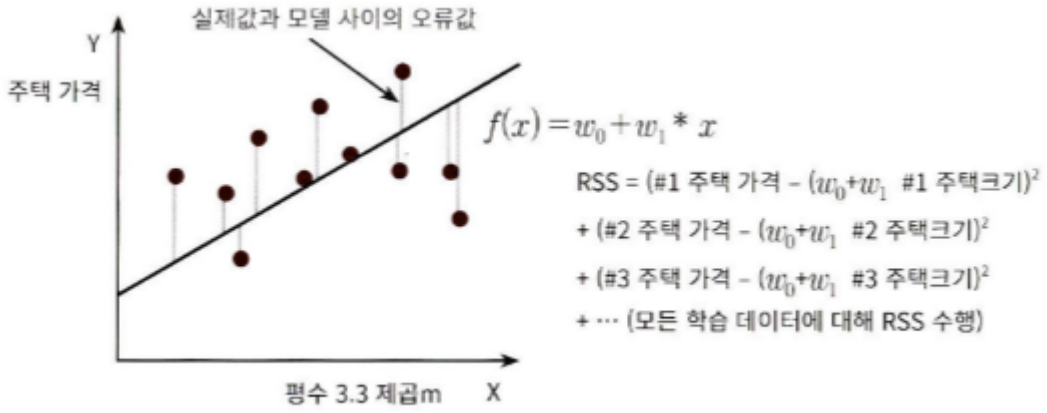
- 릿지
- 라쏘
- 엘라스텍넷
- 로지스틱 회귀

단순 선형 회귀를 통한 회귀 이해

- **단순 선형 회귀** : 독립변수도 하나, 종속변수도 하나인 선형 회귀
- ex) 주택 가격이 주택의 크기로만 결정된다고 할 때 :
  - 주택의 크기 증가 -> 가격 증가
  - 다음과 같은 선형의 관계로 표현할 수 있음.



- 회귀 계수 : 기울기 w_1, 절편 w_0
- 실제 주택 가격 : 1차 함수 값에서 실제 값만큼의 오류 값을 뺀/더한 값 ($w_0 + w_1 * X + 오류값$)
- **잔차** : 실제 값과 회귀 모델의 차이에 따른 오류 값
- **최적의 회귀 모델을 만든다는 것** : 잔차의 합이 최소가 되는 모델을 만든다는 의미
- 오류 합을 더하는 방법
  - 절댓값을 취해서 더하기
  - *오류 값의 제곱을 구해서 더하기(RSS)*



- RSS : 회귀 계수인 w가 중심 변수임을 인지하는 것이 중요
- 다음과 같은 정규화된 식으로 표현함

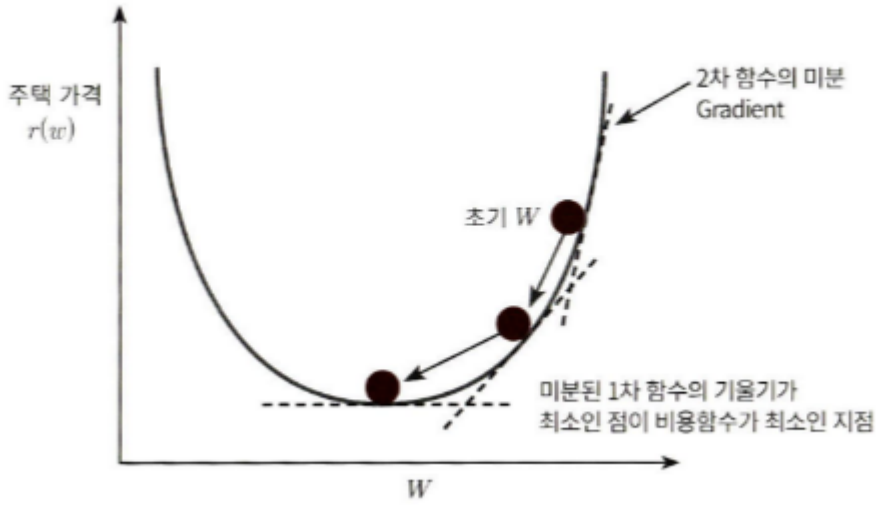
$$RSS(w_0, w_1) = \frac{1}{N}\sum_{i=1}^{N}(y_i - (w_0 + w_1 * x_i))^2$$

(*i*는 1부터 학습 데이터의 총 건수 *N*까지)

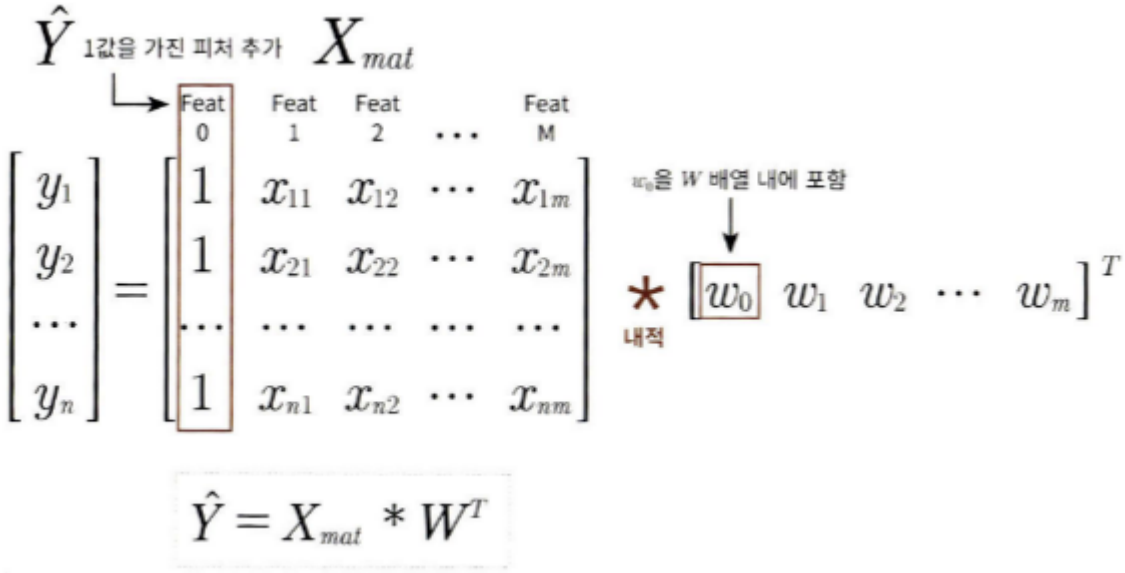
- w로 구성되는 RSS : **비용 함수, 손실 함수**
- 비용함수가 반환하는 값을 감소시키고 최종적으로 감소하지 않는 최소의 오류값을 구하는 것을 목적으로 함.

비용 최소화하기 - 경사 하강법 소개

- **경사 하강법**
  - w 파라미터가 많은 경우 비용 함수 RSS를 최소화하는 방법을 직관적으로 제공하는 방식
  - *점진적으로* 반복적인 계산을 통해 w 파라미터 값을 업데이트하면서 오류 값이 최소가 되는 w 파라미터 값을 구하는 방식
  - 비용 함수의 반환값이 작아지는 방향성을 가지고 w 파라미터를 지속적으로 보정
  - 핵심 : **'어떻게 하면 오류가 작아지는 방향으로 w값을 보정할 수 있을까?'**
  - 비용 함수가 2차 함수일 때 경사 하강법을 그림으로 나타내면 다음과 같음 :



- 경사 하강법의 일반적인 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.
- 모든 학습 데이터에 대해 값을 업데이트 하기 때문에 수행 시간이 매우 오래 걸린다는 단점 존재
  - -> **확률적 경사 하강법** 이용
    - 일부 데이터만 이용해 w가 업데이트되는 값을 계산하는 방법
- 피처가 여러 개인 경우 : 1개인 경우를 확장해서 유사하게 도출 가능
- 다음과 같은 수식으로 표현 가능 :



사이킷런 LinearRegression

- 예측값과 실제 값의 RSS(Residual Sum of Squares)를 최소화해 OLS(Ordinary Least Squares) 추정 방식으로 구현한 클래스
- *fit()* 메소드로 X, y 배열 입력받은 뒤, *coef_* 속성에 회귀 계수 W 저장
- **다중 공선성 문제** : OLS 기반의 회귀 계수 계산에서 입력 피처 간의 상관관계가 매우 높은 경우 분산이 매우 커져서 오류에 민감해지는 문제
- 해결 방법
  - 독립적인 중요한 피처만 남기고 제거
  - 규제를 적용
  - PCA를 통해 차원 축소 수행

회귀 평가 지표

평가 지표	설명	수식
MAE	Mean Absolute Error 실제 값과 예측값의 차이를 절댓값으로 변환해 평균한 것	$MAE = \frac{1}{n}\sum_{i=1}^{n}\lvert Yi - \hat{Y}i\rvert$
MSE	Mean Squared Error 실제 값과 예측값의 차이를 제곱해 평균한 것	$MSE = \frac{1}{n}\sum_{i=1}^{n}(Yi - \hat{Y}i)^2$

평가 지표	설명	수식
RMSE	Root Mean Squared Error MSE에 루트를 씌운 것	$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Yi - \hat{Y}i)^2}$
R^2	분산 기반으로 예측 성능을 평가함 1에 가까울 수록 예측 정확도가 높음	$R^2 = \frac{\text{예측값 } Variance}{\text{실제값 } Variance}$

- 사이킷런 : MAE, MSE, R^2 제공

다항 회귀와 과(대)적합/과소적합 이해

다항 회귀 이해

- **다항 회귀** : 회귀가 2차, 3차 방정식과 같은 다항식으로 표현되는 것
- 다음과 같이 표현 가능 :

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

- 다항 회귀 : **선형 회귀**
- 사이킷런 : 다항 회귀를 위한 클래스 명시적 제공 x
  - -> 비선형 함수를 선형 모델에 적용시키는 방법을 사용해 구현
  - PolynomialFeatures 클래스를 통해 피처를 다항식 피처로 변환

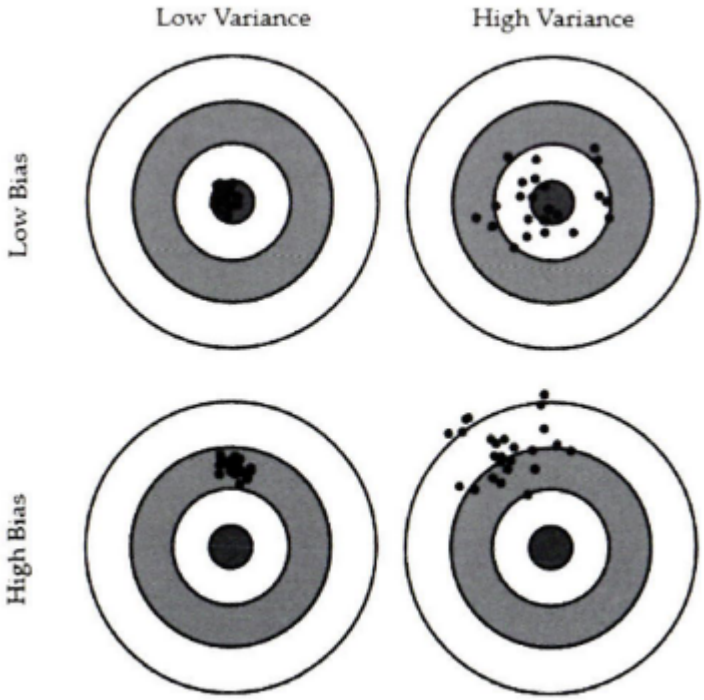
다항 회귀를 이용한 과소적합 및 과적합 이해

- 차수를 높일수록 데이터에 너무 맞춘 학습이 이루어져 **과적합의 문제가 크게 발생**
- 좋은 예측 모델 : 학습 데이터 패턴을 잘 반영하면서도 복잡하지 않은 균형 잡힌 모델

편향-분산 트레이드오프

- degree가 낮은 모델 : 지나치게 한 방향성으로 치우쳐짐 -> **고편향성**을 가짐
- degree가 높은 모델 : 지나치게 높은 변동성을 가짐 -> **고분산성**을 가짐

- 다음과 같은 그림으로 표현 가능 :



- 편향과 분산 : 한 쪽이 높으면 한 쪽이 낮아지는 경향이 있음
- 높은 편향/낮은 분산 : **과소적합**되기 쉬움
- 낮은 편향/높은 분산 : **과적합**되기 쉬움
- 편향과 분산이 트레이드오프를 이루면서 오류 cost 값이 최대로 낮아지는 모델을 구축하는 것이 중요

규제 선형 모델 - 릿지, 라쏘, 엘라스틱넷

규제 선형 모델의 개요

- 비용 함수 : 학습 데이터의 잔차 오류 값을 최소로 하는 **RSS 최소화 방법**과 과적합을 방지하기 위해 **회귀 계수 값이 커지지 않도록 하는 방법**이 서로 균형을 이루어야 함
- 다음과 같은 수식을 목표로 변경될 수 있음 :

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

- alpha : 학습 데이터 적합 정도와 회귀 계수 값의 크기 제어를 수행하는 튜닝 파라미터
  - 작을 때 : 기존과 동일한 식이 됨 -> w의 값을 상쇄함으로써 학습 데이터 적합 개선

    - 클 때 : w를 작게 만들어야 cost가 최소화되는 비용 함수 목표를 달성할 수 있음 -> 과적합 개선
- **규제** : 비용 함수에 alpha 값으로 페널티를 부여해 회귀 계수 값의 크기를 감소시켜 과적합을 개선하는 방식

릿지 회귀

- 사이킷런 : Ridge 클래스 사용
- LinearRegression보다 더 뛰어난 예측 성능을 보임

라쏘 회귀

- W의 절댓값에 페널티를 부여하는 **L1 규제**를 선형 회귀에 적용한 것
- 불필요한 회귀 계수를 급격하게 감소시켜 0으로 만드록 제거
- 적절한 피처만 회귀에 포함시키는 *피처 선택*의 특성을 가지고 있음
- 사이킷런 : Lasso 클래스 사용

엘라스틱넷 회귀

- L2 규제와 L1 규제를 결합한 회귀
- 라쏘 회귀 : 상관 관계가 높은 피처들 중에 중요 피처만을 셀렉션하고 다른 피처들을 모두 회귀 계수를 0으로 만드는 성향이 강함
    - -> 이를 완화하기 위해 L2 규제를 결합한 것 : **엘라스틱넷**
- 두 규제를 결합함으로 인해 수행시간이 상대적으로 오래 걸린다는 단점 존재
- 사이킷런 : ElasticNet 클래스 사용

선형 회귀 모델을 위한 데이터 변환

- 선형 모델 : 피처와 타깃값 간에 선형에 관계가 있다고 가정하고 결과값 예측
- 선형 회귀 모델 : 정규 분포 형태를 선호 -> 왜곡된 형태의 분포도일 경우 예측 성능에 부정적인 영향 미칠 수 있음.
    - -> 모델 적용 전 데이터에 대한 스케일링/정규화 작업을 수행하는 것이 일반적

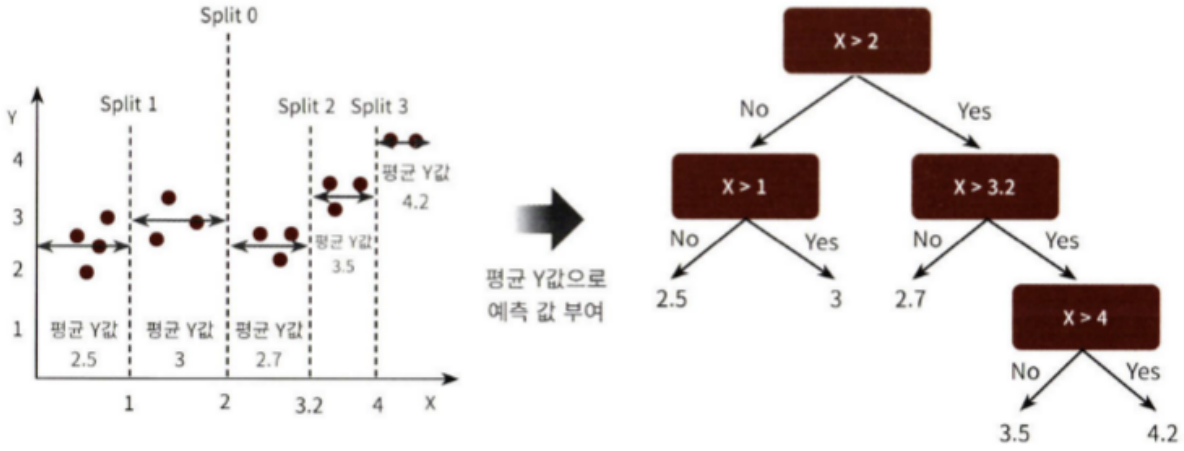
- 다음과 같은 방법이 존재
  1. StandardScaler/MinMaxScaler 사용
  2. 스케일링/정규화를 수행한 데이터 세트에 다시 다항 특성을 적용하여 변환
  3. *로그 변환*(가장 많이 사용됨) : 원래 값에 log 함수를 적용

로지스틱 회귀

- 선형 회귀 방식을 분류에 적용한 알고리즘
- 시그모이드 함수 최적선을 찾고 이 시그모이드의 반환값을 확률로 간주해 분류를 결정
- 가볍고 빠름, 이진 분류 예측 성능이 뛰어남

회귀 트리

- 회귀 함수를 기반으로 하지 않고 회귀를 위한 트리를 생성
- 회귀 트리 : 리프 노드에 속한 데이터의 평균값을 구해 회귀 예측값을 계산
- 다음과 같은 방식으로 트리를 분할 :



- 앞서 배운 모든 트리 기반 알고리즘 : 회귀도 가능!