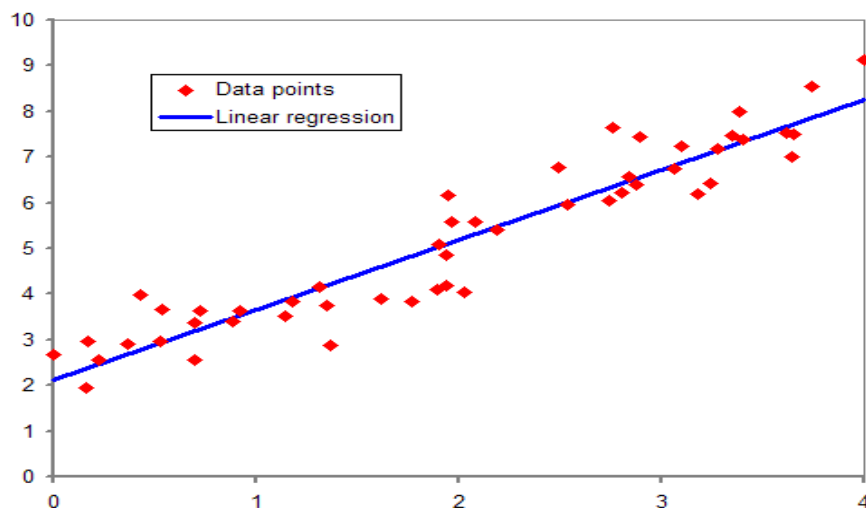


선형 회귀

회귀 (Regression)

- 정의 : 여러 개의 독립 변수와 한 개의 종속변수 간의 상관관계를 모델링 하는 기법
- 예를 들어, 아파트의 방 개수, 방 크기, 주변 학군 등 여러 개의 독립 변수에 따라 아파트 가격이라는 종속 변수가 어떤 관계를 나타내는지를 모델링하고 예측하는 것
$$Y = W_1 * X_1 + W_2 * X_2 + W_3 * X_3 + \dots + W_n * X_n$$
의 선형 회귀식 생성
 - Y : 종속 변수(아파트 가격)
 - X_1, X_2, \dots, X_n : 독립 변수(방 개수, 크기, 주변 학군 등)
 - W_1, W_2, \dots, W_n : 독립 변수의 값에 영향을 미치는 회귀 계수(=가중치)
- 머신 러닝 회귀 예측의 핵심은 주어진 피쳐와 결정 값 데이터 기반에서 학습을 통해 **최적의 회귀 계수**를 찾아 내는 것
- 독립 변수의 개수가 한 개인지 여러 개인지에 따라 단일 회귀, 다중 회귀로 나뉨



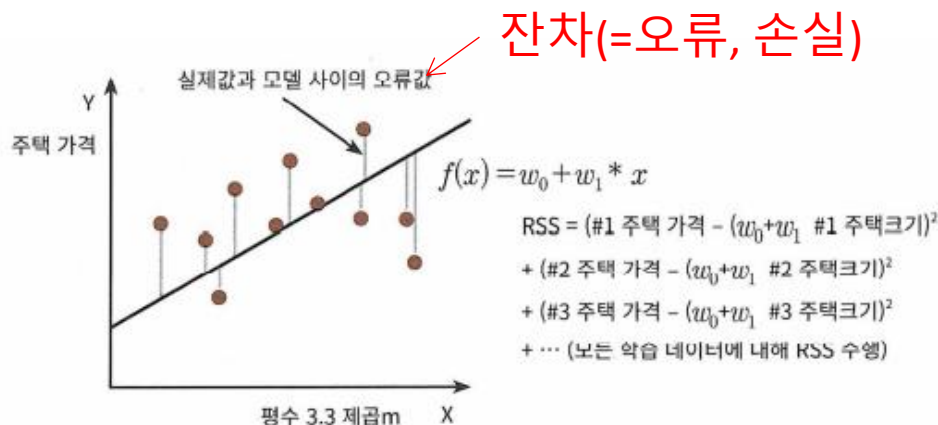
독립변수 1개와 종속 변수 1개를
가진 선형 회귀 예

출처 : 위키백과

선형 회귀 (Linear Regression)

- 정의 : 실제 값과 예측값의 차이(오류의 제곱 값)를 최소화하는 직선형 회귀선을 최적화하는 방식
- 대표적인 선형 회귀 모델
 1. 일반 선형 회귀 : 예측값과 실제 값의 RSS(차이의 제곱값)를 최소화할 수 있도록 회귀 계수를 최적화하며, 규제를 적용하지 않은 모델
 2. 릿지(Ridge) : 선형 회귀에 L2 규제를 추가한 회귀 모델. L2 규제는 상대적으로 큰 회귀 계수 값의 예측 영향도를 감소시키기 위해서 회귀 계수값을 더 작게 만드는 규제 모델
 3. 라쏘(Lasso) : 선형 회귀에 L1 규제를 추가한 회귀 모델. L1 규제는 예측 영향력이 작은 피처의 회귀 계수를 0으로 만들어 회귀 예측 시 피처가 선택되지 않게 하는 것
 4. 엘라스틱넷(ElasticNet) : L2, L1 규제를 함께 결합한 모델. 주로 피처가 많은 데이터 세트에서 적용
(L1 규제로 피처의 개수를 줄임과 동시에 L2 규제로 계수 값의 크기를 조정)
 5. 로지스틱 회귀(Logistic Regression) : 선형 회귀 방식을 분류에 적용한 알고리즘(분류에 사용)
(시그모이드 함수를 사용하여 데이터를 0 또는 1로 분류)

선형 회귀 (Linear Regression)



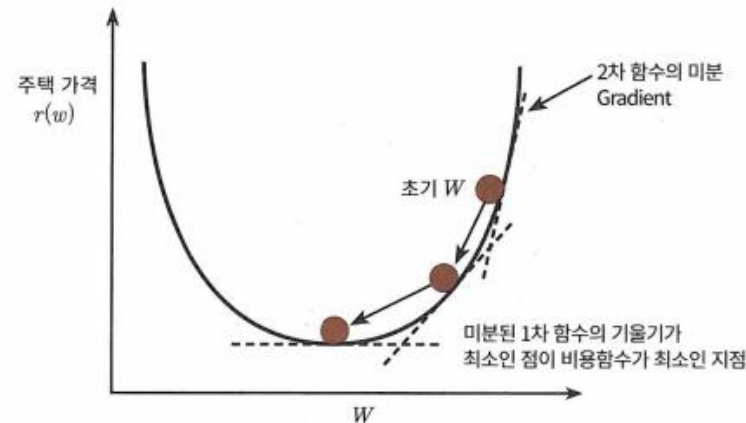
- 최적의 회귀 모델을 만든다는 것은 바로 전체 데이터의 잔차(오류 값) 합이 최소가 되는 모델을 만든다는 의미와 같음 (= 오류 값 합이 최소가 될 수 있는 최적의 회귀 계수를 찾는다)
- RSS : 잔차의 제곱을 구해서 더하는 방식, 오류 정도를 평가하는 지표
 - RSS 사용하는 이유 :
 1. 각 잔차에 대한 일관성 부여
 2. 미분을 용이하기 하기 위해서 (RSS는 2차 함수) (경사하강법을 적용하기 위해)
 - 회귀에서 이 RSS는 비용(Cost)이며 w변수(회귀 계수)로 구성되는 RSS를 비용함수(=손실함수)라고 함

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

[i는 1부터 학습 데이터의 총 건수 N까지]

- 머신 러닝 회귀 알고리즘은 데이터를 계속 학습하면서 이 비용함수가 반환하는 값(오류값)을 지속해서 감소시키고 최종적으로는 더 이상 감소하지 않는 최소의 오류 값을 구하는 것

경사 하강법(Gradient Descent) – 비용(오류) 최소화



- 정의 : ‘점진적으로’ 반복적인 계산을 통해 W 파라미터 값을 업데이트하면서 오류 값이 최소가 되는 W 파라미터를 구하는 방식

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

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

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

파이토치를 사용하여 선형 회귀 구현 :
(슬라이드쇼 누르고 도형 클릭)
(오류 나는 사진은 따로 첨부하였음)

파라미터의 업데이트(w_0, w_1)

$$\text{새로운 } w_1 = \text{이전 } w_1 + \eta \frac{2}{N} \sum_{i=1}^N x_i * (\text{실제값}_i - \text{예측값}_i)$$

$$\text{새로운 } w_0 = \text{이전 } w_0 + \eta \frac{2}{N} \sum_{i=1}^N (\text{실제값}_i - \text{예측값}_i)$$

$$W := W - \alpha \frac{\partial}{\partial W} \text{cost}(W) \quad \alpha = \text{learning rate}$$



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

이 밖에 MSE나 RMSE에 로그를 적용한 MSLE(Mean Squared Log Error)와 RMSLE(Root Mean Squared Log Error)도 사용합니다.

평가 방법	사이킷런 평가 지표 API	Scoring 함수 적용 값
MAE	<code>metrics.mean_absolute_error</code>	<code>'neg_mean_absolute_error'</code>
MSE	<code>metrics.mean_squared_error</code>	<code>'neg_mean_squared_error'</code>
R ²	<code>metrics.r2_score</code>	<code>'r2'</code>

Scoring 함수에 음수값을 반환하는 이유는 사이킷런의 Scoring 함수가 score 값이 클수록 좋은 평가 결과로 자동 평가하기 때문. 실제 값과 예측 값의 오류 차이를 기반으로 하는 회귀 평가 지표의 경우 값이 커지면 안 좋은 모델이므로 이를 보정하기 위해 -1을 곱함

다중 회귀 (Multiple Regression)

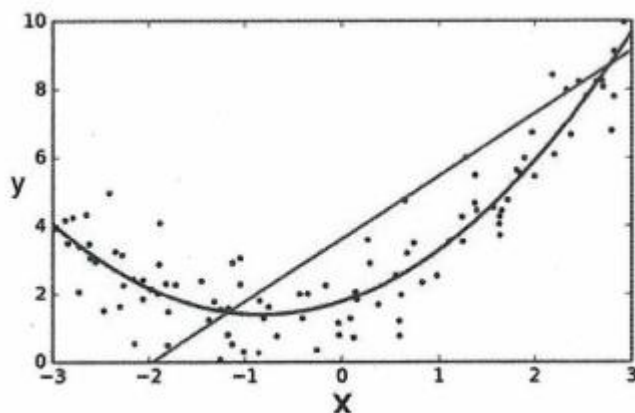
- 정의 : 독립 변수의 개수가 여러 개이고 하나인 종속변수의 관계를 나타내는 회귀 모델
 - 예를 들어, 아파트의 가격(종속변수)을 측정하는 것은 방 개수, 방 크기 등 다양한 독립변수에 의함.
- 다중공선성(Multicollinearity)
 - 정의 : 독립변수 간의 상관관계가 매우 높을 때, 하나의 독립 변수 변화가 다른 독립 변수에 영향을 미쳐 모델이 크게 흔들리는 것
(임의의 독립 변수 x 는 종속 변수 y 하고만 상관 관계가 있어야 하며, 독립 변수끼리 상관 관계가 있어서는 안됨)
 - 다중공선성 확인은 분산팽창지수(Variation Inflation Factor; VIF)로 확인 가능 $VIF_i = \frac{1}{1 - R_i^2}$
 - 일반적으로 VIF가 10이 넘으면 다중공선성 있다고 판단하며 5가 넘을 때는 주의할 필요가 있다고 봄
 - 관련 실습 코드 : 

다항 회귀(Polynomial Regression)

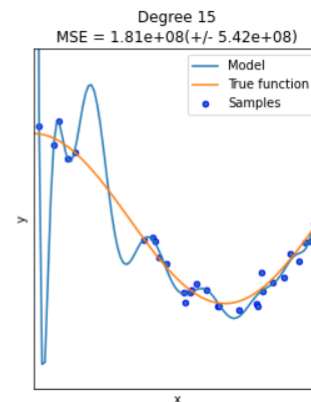
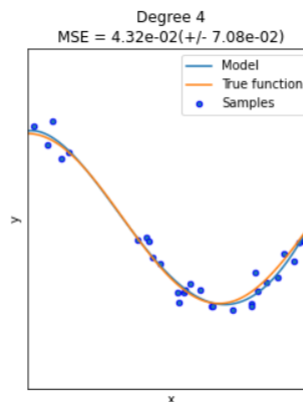
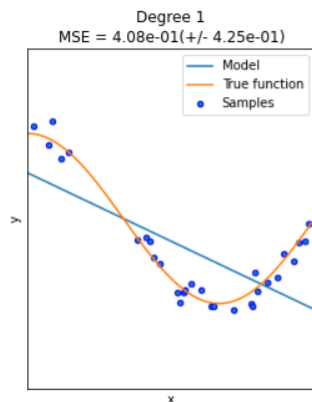
- 정의 : 회귀가 독립 변수의 단항식이 아닌 2차, 3차 방정식과 같은 다항식으로 표현되는 것

다항 회귀는 $y = w_0 + w_1 * x_1 + w_2 * x_2 + w_3 * x_1 * x_2 + w_4 * x_1^2 + w_5 * x_2^2$ 과 같이 표현 $\leq x_1, x_2$ 을 2차로

- 피쳐들의 상호작용을 보여줄 수 있고 모델의 차수가 높아져 선형이 아니라 곡선형의 모델로 데이터 설명 가능



〈 주어진 데이터 세트에서 다항 회귀가 더 효과적임 〉



차수에 따른 예측 성능 비교

변수 선택법 (Variable Selection)

- 다중공선성 해결 방안
- 전진 선택법(Forward Selection)
 - 독립 변수를 아무것도 넣지 않은 상태에서 기존 모형에서 가장 설명력이 좋은 독립 변수를 하나씩 추가
 - 전진 선택법에서는 변수를 추가할지 말지 결정하는 유의 수준을 설정
 - 장점 : 구현 과정이 간단하고 독립 변수가 많은 상황에서도 사용 가능
 - 단점 : 한 번 선택된 독립 변수는 계속 모형에 존재하며, 일치성(샘플 수가 많아질수록 실제 모형에 수렴하는 성질)이 만족되지 않음
- 후진선택법(Backward Selection)
 - 모든 독립 변수가 포함된 모형에서 설명력이 가장 적은 변수를 제거해나가는 방법
 - 장점 : 구현 과정이 간단하고 독립 변수가 많은 상황에서도 사용 가능
 - 단점 : 한 번 선택된 독립 변수는 계속 모형에 존재하며, 일치성이 만족되지 않음
- 단계별 선택법(Forward Stepwise Selection)
 - 전진 선택법에서 후진 소거법을 추가한 방법(독립 변수가 0개인 상태에서 추가하거나 뺌)
 - 장점 : 구현 과정이 간단하고 한 번 들어간 독립 변수는 계속 포함된다는 전진 선택법의 단점을 일부 보완
 - 단점 : 독립 변수가 많아지면 계산량이 늘어나고 일치성을 만족하지 않음

과적합 (Overfitting)

- 정의 : 학습을 할수록 정확도(Accuracy)가 올라가는 것이 정상이지만, 학습 과정에서 학습 모델이 주어진 데이터에 너무 과하게 맞춰져서(Overfit) 조금이라도 다른 데이터(테스트 데이터)만 들어와도 다른 결과로 예측하여 결과적으로 정확도가 낮아지는 현상
- 이유
 - 학습 데이터 부족
 - 데이터 대비 높은 모델 복잡도
- 해결 방법
 - Feature 수 줄이기(차수가 높아져서 과적합 발생)
 - 규제(Regularization)
 - Dropout(딥러닝, 뉴런을 무작위로 제외)
 - Early stopping(딥러닝, 과적합이 일어나기 전에 학습을 종료)
 - Cross Validation(교차 검증) : K-Fold 교차 검증(대표적), 훈련데이터로 학습/검증데이터로 성능을 높임/테스트 데이터로 최종 성능을 파악

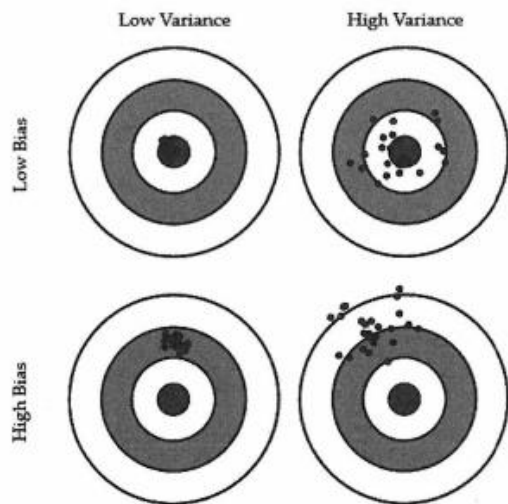


K-fold 교차검증

모든 데이터를 학습과 평가에 활용할 수 있기 때문에 테스트 데이터에 과적합 발생 방지

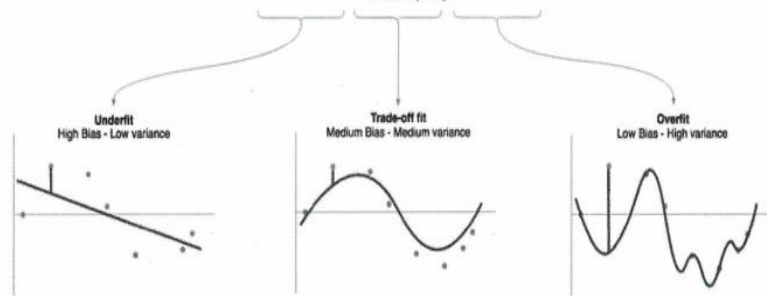
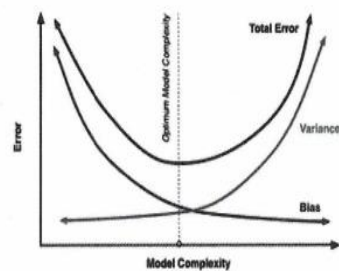
편향-분산 트레이드오프 (Bias-Variance Trade off)

- 편향-분산 트레이드 오프는 머신 러닝이 극복해야 할 가장 중요한 이슈 중 하나
- 다항 회귀의 Degree 1과 같은 모델은 매우 단순화된 모델로서 지나치게 한 방향으로 치우친 경향
 - 이와 같은 모델을 고편향(High Bias)성을 가졌다고 표현
- 다항 회귀의 Degree 15와 같은 모델은 학습 데이터 하나하나의 특성을 반영하면서 매우 복잡한 모델이 되고 높은 변동성을 가짐
 - 이와 같은 모델을 고분산(High Variance)성을 가졌다고 표현



〈 편향과 분산의 고/저에 따른 표현 〉

<http://scott.fortmann-roe.com/docs/BiasVariance.html>에서 발췌

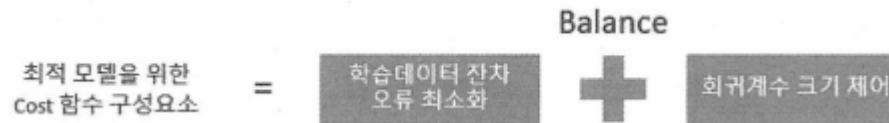


〈 편향과 분산에 따른 전체 오류 값(Total Error) 곡선. <http://scott.fortmann-roe.com/docs/BiasVariance.html>에서 발췌. 〉

- 편향과 분산이 서로 트레이드오프를 이루면서 비용 값이 최대한로 낮아지는 모델을 구축하는 것이 목표!

규제 (Regulation) 선형 모델

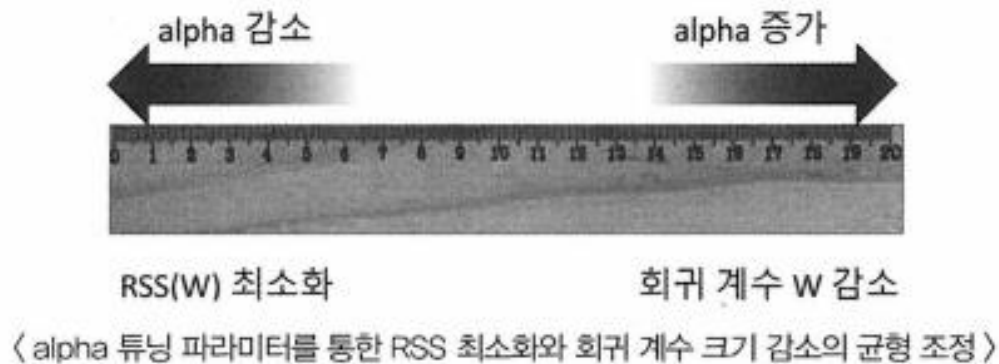
- Overfitting(과적합)을 해결하는 방법 중 하나(다른 방법은 피처 개수 줄이기-변수선택법)
- 비용 함수는 학습 데이터의 잔차 오류 값을 최소로 하는 RSS 최소화 방법과 과적합을 방지하기 위해 회귀 계수 값이 커지지 않도록 하는 방법이 서로 균형을 이뤄야 함



- 규제의 정의 : 비용함수에 alpha 값으로 패널티를 부여해 회귀 계수 값의 크기를 감소시켜
과적합을 개선하는 방식
alpha : 학습 데이터 적합 정도와 회귀 계수 값의 크기 제어를 수행하는 튜닝 파라미터
- 릿지 회귀 (Ridge Regression)
- 라쏘 회귀 (Lasso Regression)
- 엘라스틱넷 회귀 (ElasticNet Regression)

규제 선형 모델 – 릿지 회귀 (Ridge Regression)

- 정의 : L2 규제를 적용한 회귀
- L2 규제 : W 의 제곱에 대해 패널티를 부여하는 방식 \Rightarrow 비용 함수 목표 = $\text{Min}(\text{RSS}(W) + \alpha * \|W\|_2^2)$
- α 값을 크게 하면 비용 함수는 회귀 계수 W 의 값을 작게 해 과적합을 개선할 수 있으며, α 값을 작게 하면 회귀 계수 W 의 값이 커져도 어느 정도 상쇄가 가능하므로 학습 데이터 적합 개선



- α 값을 0에서부터 지속적으로 값을 증가시키면 회귀 계수 값의 크기를 감소할 수 있음

규제 선형 모델 – 라쏘 회귀 (Lasso Regression)

- 정의 : L1 규제를 적용한 회귀
- L1 규제 : W의 절대값에 대해 패널티를 부여하는 방식 $\Rightarrow \text{RSS}(W) + \alpha * \|W\|_1$
- L2 규제가 회귀 계수의 크기를 감소시키는 데 반해, L1 규제는 불필요한 회귀 계수를 급격하게 감소시켜 0으로 만들고 제거(alpha 값이 커지면 회귀 계수가 0이 되는 특징)
- L1 규제는 적절한 피처만 회귀에 포함시키는 피처 선택의 특성을 가짐
- 회귀 계수를 0으로 만들어 해당 변수를 모델에서 삭제하고 모델을 더 단순하게 만들어 해석에 용이

Ridge	Lasso
L_2 -norm regularization	L_1 -norm regularization
변수 선택 불가능	변수 선택 가능
Closed form solution 존재 (미분으로 구함)	Closed form solution이 존재하지 않음 (numerical optimization 이용)
변수 간 상관관계가 높은 상황 (collinearity) 에서 좋은 예측 성능	변수 간 상관관계가 높은 상황에서 ridge에 비해 상대적으로 예측 성능이 떨어짐
크기가 큰 변수를 우선적으로 줄이는 경향 이 있음	

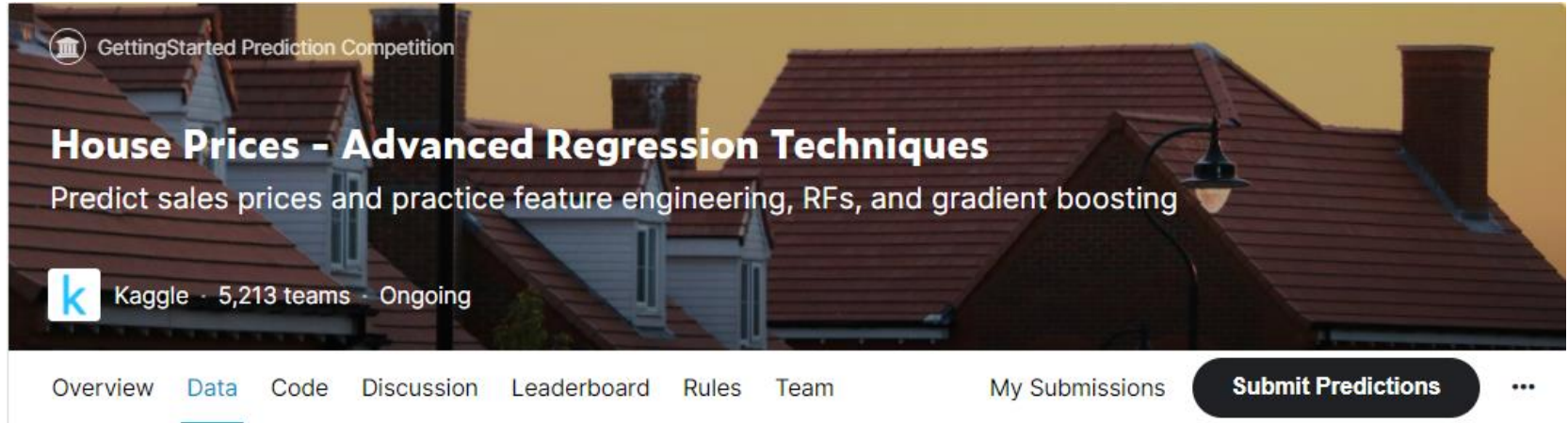
규제 선형 모델 – 엘라스틱넷 회귀 (ElasticNet Regression)

- 정의 : L2 규제와 L1 규제를 결합한 회귀
 - 비용 함수 목표 : $RSS(W) + \alpha_2 * ||W||_2^2 + \alpha_1 * ||W||_1$ (이 식을 최소화하는 w 찾기)
 - 라쏘 회귀(L1)에서 중요 피쳐들만 남기고, 나머지를 0으로 만드는 문제를 개선하기 위해 L2 규제를 추가한 것(이러한 규제 때문에 라쏘 회귀보다 사라진 회귀 계수가 적음)
 - 단점 : L1과 L2 규제가 결합된 규제로 인해 수행시간이 상대적으로 오래 걸림
 - l1_ratio 파라미터를 가지는데 이는 $a/(a+b)$ 이다. l1_ratio가 0이면 a가 0이므로 L2 규제와 같고 l1_ratio가 1이면 b가 0이므로 L1 규제와 같다.
-
- 릿지 & 라쏘 & 엘라스틱 회귀를 최종 다중 회귀 코드에서 사용 후 비교하였음

선형 회귀 모델을 위한 데이터 전처리

- 선형 회귀 모델은 일반적으로 피처와 타겟값 간에 선형의 관계가 있다고 가정하고 최적의 선형 함수를 찾아내 결과값을 예측
- 선형 회귀 모델은 피처값과 타겟값의 분포가 정규 분포 형태를 매우 선호→스케일링/정규화 작업
(왜곡된 형태는 예측 성능에 부정적인 영향을 미칠 가능성이 높음)
 - 피처 데이터 세트에 적용하는 변환 작업
 - StandardScaler 클래스를 이용해 평균이 1, 분산이 1인 표준 정규 분포를 가진 데이터 세트로 변환하거나 MinMaxScaler 클래스를 이용해 최솟값이 0이고 최대값이 1인 값으로 정규화를 수행
 - 스케일링/정규화를 수행한 데이터 세트에 다시 다항 특성을 적용하여 변환
 - 로그 변환을 이용하여 정규 분포에 가까운 형태로 만듦
(가장 많이 사용하는 방법, 1번 방법은 성능 향상을 기대하기 어렵고 2번은 과적합 문제)
 - 타겟 데이터 세트에 적용하는 변환 작업
 - 일반적으로 로그 변환 적용(`np.log1p()`)
- 결측치 제거
- 이상치 제거
- 범주형(카테고리) 데이터 처리 – 원핫 인코딩

다중 회귀 모델 코드 구현



https://github.com/LeeYunseol/Lab_study/blob/main/%EC%84%A0%ED%98%95%ED%9A%8C%EA%B7%80/%EB%8B%A4%EC%A4%91%20%ED%9A%8C%EA%B7%80%20%EB%AA%A8%EB%8D%B8%20%EC%BD%94%EB%93%9C%20%EA%B5%AC%ED%98%84.ipynb

참고 자료

- 파이썬 머신러닝 완벽 가이드
- Pytorch로 시작하는 딥러닝 입문
- 귀퉁이 서재 블로그(<https://bkshin.tistory.com/>)