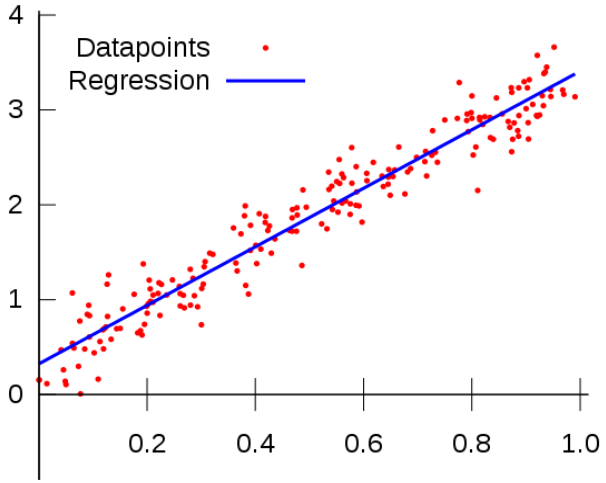
Algorithm Study

1. Linear Regression (선형회귀 분석)

- 선형 모델은 입력 특성에 대한 선형 함수를 만들어 예측을 수행한다.

$$\hat{y} = w[i] * x[i] + b$$

- 일반화된 함수는 위의 수식과 같이 표현된다.
- i는 하나의 데이터 포인트에 대한 특성을 나타내며 i의 개수가 증가하면 $w[0], w[1]$과 같이 수식이 길어진다.
- W와 b는 모델이 학습할 파라미터이며 직선의 방정식 상에서 기울기와 절편을 나타낸다.
- 이를 다르게 생각하면 예측값 Y는 입력 특성(x)의 각 가중치(w)를 곱해서 더한 가중치 합과 편향(b)의 합이다.



- 회귀 모델은 특성이 하나인 경우 직선이고, 그 수가 늘어날 수록 초평면이 되는 특징을 가지고 있다.
- 특성이 많은 데이터 셋의 선형 모델은 훌륭한 성능 기대 가능
- 회귀를 위한 선형 모델은 다양하며 모델 간의 차이는 훈련 데이터에서 파라미터 값 W와 b를 학습하는 방법과 모델의 모델의 복잡도를 제어하는 방법에서 차이가 난다.
- R2값이 훈련 세트와 테스트 셋의 차이가 많이 나면 과대적합 차이는 적지만 R2값이 낮은 경우는 과소적합
- 과소적합시에는 추가적인 피쳐 고려해야 하고, 과대적합인 경우 모델의 복잡도 개선이 필요

Algorithm Study

1. Linear Regression (선형회귀 분석)

1) 최소제곱법(Ordinary least Squares)

- 가장 간단하고 오래된 알고리즘
- 예측과 훈련 세트에 있는 타겟(y) 사이의 평균 제곱오차(Mean Squared Error)를 최소화하는 파라미터 W와 b를 찾는다.
- MSE는 예측값과 타겟값의 차이를 제곱하여 더한 후에 샘플 개수로 나눈 것이다.
  (RMSE는 MSE값에 제곱근을 취함)
- 매개변수가 없는 것이 장점이지만, 반대로 모델의 복잡도를 제어할 방법도 없다.
- W 기울기 파라미터는 가중치(weight) or 계수(coefficient)로 표현
  (입력값에 하나씩 대응되는 값 보유)
- B 파라미터는 편향(bias) or 절편(intercept)로 표현 (입력 특성이 하나)
- 대표적으로 이용 가능한 파이썬의 라이브러리는 statsmodel의 OLS와
  sklearn의 LinearRegression이 있다.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2$$

- MSE의 산출 식이며 n은 데이터 피쳐 개수 이다.

Algorithm Study

1. Linear Regression (선형회귀 분석)
2) 릿지 회귀(Ridge)

- 회귀를 위한 선형 모델이므로 최소적합법에서 사용한 것과 같은 예측 함수를 이용
- L2규제를 적용하여 w의 모든 원소가 0에 가깝게 되길 원하여 모든 특성이 출력에 주는 영향을 최소한으로 만든다 – 가중치의 절댓값을 작게 만드는 것 (모델이 과대적합이 되지 않도록 모델을 강제로 제한한다는 의미= 규제) *부록참고
- Alpha 매개변수를 1.0보다 높은 값을 높이면 w를 0더 가깝게 만든다.
- 릿지는 과소적합법 보다는 과대적합이 적어진다. 모델의 복잡도가 낮아지면 훈련 세트에서 성능은 나빠지지만 자유로운 모델이기 때문에 더 일반화된 모델이 된다.
- 릿지는 모델을 단순하게 해주고 훈련 세트와 테스트 세트 사이 절충 할 수 있는 방안
- 릿지 모델의 MSE 수식은 아래와 같다.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2 + a\sum_{j=1}^{m} w_j^2$$

a = 알파값 , 크게하면 w 감소, 작게하면 w 증가

Algorithm Study

1. Linear Regression (선형회귀 분석)

3) 라소 회귀(Lasso)

- 릿지와 마찬가지로 w를 0에 가깝게 만들려고 함
  (다만 L1규제를 이용하여 방식이 달라 어떤 계수는 0이 되는 경우도 있다)
  -> 즉 모델에서 완전히 제외되는 피쳐 발생가능

- 일부 계수를 0으로 만들면 모델을 이해하기 쉽고 중요한 특성이 드러나
  피쳐 셀렉션이 자동으로 이루어진다고 볼 수 있음

- 릿지와 마찬가지로 w의 값을 얼마나 강하게 0으로 보낼지를 조절하는 alpha 매개변수 지원

- L1,L2규제를 함께 쓰는 Elastic-Net 방식에서 L2규제를 제외한 것
  (scikit-learn에서는 릿지와 라소의 패널티를 결합한 ElasticNet도 제공)

- Alpha 값을 줄이게 되면 가장 낮은 오차를 찾아가는 반복 횟수가 늘어남
  (반복횟수는 lasso.n_iter로 값 확인 가능)
  -> 피쳐가 매우 많고 그 중 일부만 중요하고 분석하기 쉬운 모델을 원하면 라소를 이용

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2 + a\sum_{j=1}^{m}|w_j|$$

Algorithm Study

1. Linear Regression (선형회귀 분석)

*부록

(1) 학습곡선

- 데이터 세트의 크기에 따른 모델의 성능 변화를 나타낸 그래프
- 규제가 적용된 릿지는 과소적합법 보다 성능이 낮음
- 하지만 데이터의 양이 많아질 수록 과대적합으로 인하여 과소적합법의 성능이 떨어짐
- 테스트 데이터의 경우 릿지 회귀의 점수가 더 높으며 데이터 세트의 크기가 작을수록 릿지가 유리
(데이터의 양이 충분하면 규제의 중요성 감소)
- 그러나 데이터가 충분하다면, 릿지 회귀와 선형 회귀의 성능은 같아짐
(데이터의 양이 충분하다면 과대적합 가능성 감소)