개요

- 보통의 경우 학습에 사용되는 데이터들은 각 컬럼이 가지는 값의 범위가 다양하게 나타남
- 데이터셋에서 어떤 컬럼은 범위가 0~1이고 어떤 컬럼은 -1000~1000 일수도 있음
- 대부분의 분석 알고리즘은 컬럼 간 데이터의 범위가 크게 차이 날 경우 성능이 좋지 않음
- 값의 범위가 작은 컬럼에 비해 값의 범위가 큰 컬럼이 타깃 변수를 예측하는데 큰 영향을 준다고 판단하게 되는 것이다.
- 따라서 각 컬럼을 변수로 하여 연산을 하는 분석모델을 사용하는 경우 스케일링(scaling)을 통해 모든 컬럼의 값의 범위를 같게 만들어 주어야 함

처리 순서

1. Scaler 선택
2. 객체 생성
3. fit()
4. train 데이터를 transform() 사용하여 변환
5. test 데이터도 transform() 사용하여 변환
6. inverse_transform()으로 원 데이터로 변환

Standard Scaler

$$x_{scaled} = \frac{x - \bar{x}}{\sigma}$$

- 표준화 방식으로 기본 스케일링 방식으로 컬럼들을 평균이 0, 분산이 1인 정규분포로 스케일링
- 최솟값과 최댓값의 크기를 제한하지 않아 이상치에 매우 민감
- 그렇기 때문에 이상치를 미리 확인 및 정제한 후 사용하는 것이 좋다
- 회귀보다는 분류분석에서 유용함 src_데이터 스케일링 > ^71f1ba

Min-max Scaler

$$x_{scaled} = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

- 정규화 방식
- 컬럼들을 0과 1 사이의 값으로 스케일링하는 방식

- 최솟값이 0, 최댓값이 1
- 이상치에 매우 민감하므로 이상치를 미리 정제한 후 수행하는 것이 좋음
- 분류보다는 회귀에 유용한 방식
- **회귀 vs. 분류:**
  - Min-Max 정규화는 주로 회귀 문제에 유용
  - 회귀 모델은 연속적인 값을 예측하므로 특성 값의 범위를 제한하는 것이 예측 성능에 도움이 될 수 있음
  - 분류 문제에서는 특성 값의 범위를 제한하는 것이 항상 유리하지 않음
  - 특히, 특성 값이 특정 클래스를 구분하는 데 중요한 정보를 담고 있을 경우 Min-Max 정규화는 이러한 정보를 잃게 만들 수 있음

src_데이터 스케일링 > ^ba08ae

Max Abs Scaler

$$x_{scaled} = \frac{x}{max(|x|)}$$

- 최대절댓값과 0이 각각 1, 0이 되도록 스케일링하는 정규화 방식
- 모든 값이 -1과 1사이에 표현되며, 데이터가 양수인 경우 Min-Max Scaler와 동일
- 이상치에 매우 민감하며, 분류보다는 회귀분석에서 유용

src_데이터 스케일링 > ^bd2d77

Robust Scaler

- 평균과 분산 대시 중앙값(Median)과 사분위 값을 활용하는 방식
- 중앙값을 0으로 설정하고 IQR을 사용하여 이상치의 영향을 최소화함
- quantile_range 파라마티(default[0.25, 0.75])를 조정하여 더 넓거나 좁은 범위의 값을 이상치로 설정하여 정제할 수 있음

src_데이터 스케일링 > ^76f1e9

log 변환

원래 값에 log 함수를 적용하면 보다 정규 분포에 가까운 형태로 값이 분포됩니다.

- 로그 변환은 매우 유용한 변환이며, 실제로 선형 회귀에서는 로그 변환이 훨씬 많이 사용되는 변환 방법입니다.
- np.log()가 아니라 np.log1p()를 이용하는 이유는 log() 함수를 적용하면 언더 플로우가 발생하기 쉬워서 1 + log() 함수를 적용합니다.

- 이를 구현한 함수가 np.log1p()입니다.