🧡

CH5 Kaggle

다중공선성이 있다고 판단했을 때 해결 방법을 찾아보았는데,

노래 인기도 예측에서 사용한 변수 선택과 추출 방법 외에도 정규화를 이용하는 방법도 있다고 해서 여러 정규화 방법에 대해 다루어 보았다.

⇒ TIP 정규화를 진행한 후, 변수 선택이나 추출법을 진행하면 훨씬 좋은 VIF 결과를 얻을 수 있음!

Normalization

- 특성들의 범위가 같은 크기를 가지도록 특성별 값을 비례적으로 조정하는 방법
- **Min-Max Scaler**
  특성들의 범위가 [0,1]이 되도록 맞춤

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

$$x' = a + \frac{(x - \min(x))(b - a)}{\max(x) - \min(x)}$$

```
import pandas as pd
from sklearn.preprocessing import MinMaxScaler

df = pd.DataFrame([
    [2, 1, 3],
    [3, 2, 5],
    [3, 4, 7],
    [5, 5, 10],
    [7, 5, 12],
    [2, 5, 7],
    [8, 9, 13],
    [9, 10, 13],
    [6, 12, 12],
    [9, 2, 13],
    [6, 10, 12],
    [2, 4, 6]
```

```
], columns=['hour', 'attendance', 'score'])
x_data = df.drop(['score'], axis=1)
y_data = df['score']

transformer = MinMaxScaler()
#transformer = MinMaxScaler(feature_range=(0, 1))
transformer.fit(x_data) #MinMaxScaler 모델에 x_train_df 데이터 적용 (최소값, 최대값 계산)
x_data = transformer.transform(x_data)
print(x_data)
```

- **Z-Score Scaler**
  원데이터가 정규분포를 따른다면, 평균이 0, 표준편차가 1인 표준 정규분포로 변경

```
import pandas as pd
from sklearn.preprocessing import StandardScaler

transformer = StandardScaler()
transformer.fit(x_data)
#StandardScaler 모델에 x_train_df 데이터 적용 (평균, 표준편차 계산)
x_data = transformer.transform(x_data)
print(x_data)
```

- **RobustScaler**

  평균과 분산 대신에 중간값과 사분위값 사용 → 아웃라이어의 영향을 최소화

```
from sklearn.preprocessing import RobustScaler

transformer = RobustScaler()
transformer.fit(x_data)
x_data = transformer.transform(x_data)
print(x_data)
```

- **Nomalizer**

  StandardScaler는 열에 적용되는 반면, Nomalizer는 행에 적용 → shape 주의해서 사용

```
from sklearn.preprocessing import Normalizer

transformer = Normalizer()
transformer.fit(x_data)
```

```
x_data = transformer.transform(x_data)
print(x_data)
```