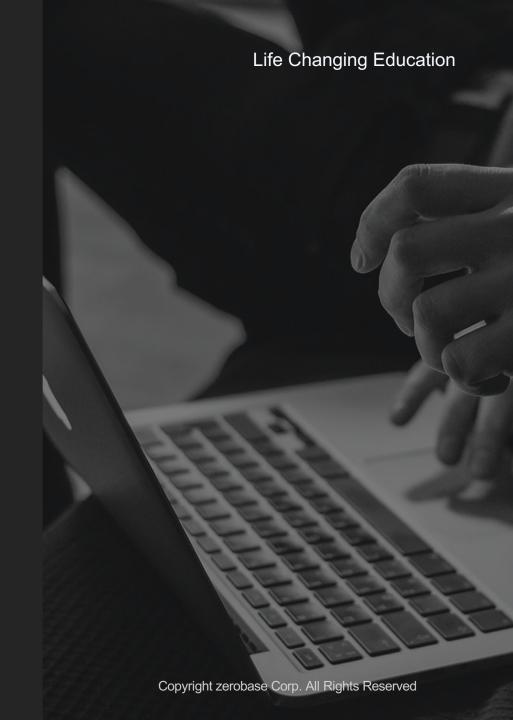
zero-base/

Chapter 05. label_encoder, min_max, standard, robust scaler



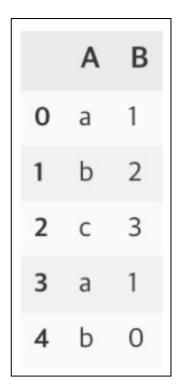
간단한 데이터 하나~

```
import pandas as pd

df = pd.DataFrame( {
        'A' : ['a', 'b', 'c', 'a', 'b'],
        'B' : [1, 2, 3, 1, 0]
})

df
```

이렇게 생김



label encoder란?

```
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()
le.fit(df['A'])

LabelEncoder()
```

대상이 되는 문자로 된 데이터를

```
le.classes_
```

```
array(['a', 'b', 'c'], dtype=object)
```

숫자 - 카테고리컬한 데이터로~

```
df['le_A'] = le.transform(df['A'])
df

A B le_A
0 a 1 0
1 b 2 1
2 c 3 2
3 a 1 0
4 b 0 1
```

fit 한 후에 transform을 하면 변환됨

```
le.transform(['a', 'b'])

array([0, 1])
```

한 번에 줄이는 것은 fit_transform

le.fit_transform(df['A'])

array([0, 1, 2, 0, 1])

혹시 역변환하고 싶다면

```
le.inverse_transform([1,2,2,2])
```

```
array(['b', 'c', 'c', 'c'], dtype=object)
```

min-max scaling이란?

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

역시 데이터 준비하고

```
df = pd.DataFrame( {
    'A' : [10, 20, -10, 0, 25],
    'B' : [1, 2, 3, 1, 0]
})
df
      А В
   o 10 1
  1 20 2
   2 -10 3
  3 0 1
   4 25 0
```

fit~

```
from sklearn.preprocessing import MinMaxScaler

mms = MinMaxScaler()

mms.fit(df)

MinMaxScaler()
```

그러면 이런 요소를 찾게 된다

```
mms.data_max_, mms.data_min_
(array([25., 3.]), array([-10., 0.]))
```

transform 시키면 이렇게

역변환 시키면 이렇게

한번에는 이렇게~^^

표준정규분포

$$z=\frac{X-\mu}{\sigma}$$

아까부터 사용하던 원 데이터



이번에도 fit~

```
from sklearn.preprocessing import StandardScaler

ss = StandardScaler()
ss.fit(df)

StandardScaler()
```

평균과 표준편차

```
ss.mean_, ss.scale_
```

(array([9. , 1.4]), array([12.80624847, 1.0198039]))

transform

fit_transform

inverse_transform

Robust Scaler

$$\frac{x_i - Q_2}{Q_3 - Q_1}$$

데이터를 새롭게...

```
df = pd.DataFrame({
    'A':[-0.1, 0., 0.1, 0.2, 0.3, 0.4, 1.0, 1.1, 5]
})
```

방금 배운 것 포함 Robust 스케일러까지

```
from sklearn.preprocessing import MinMaxScaler, StandardScaler, RobustScaler

mm = MinMaxScaler()
ss = StandardScaler()
rs = RobustScaler()
```

이번에는 모두 불러와서

```
df_scaler = df.copy()

df_scaler['MinMax'] = mm.fit_transform(df)

df_scaler['Standard'] = ss.fit_transform(df)

df_scaler['Robust'] = rs.fit_transform(df)
```

스케일링 된 결과

df_scaler

| | Α | MinMax | Standard | Robust |
|---|------|----------|-----------|-----------|
| 0 | -0.1 | 0.000000 | -0.656688 | -0.44444 |
| 1 | 0.0 | 0.019608 | -0.590281 | -0.333333 |
| 2 | 0.1 | 0.039216 | -0.523875 | -0.222222 |
| 3 | 0.2 | 0.058824 | -0.457468 | -0.111111 |
| 4 | 0.3 | 0.078431 | -0.391061 | 0.000000 |
| 5 | 0.4 | 0.098039 | -0.324655 | 0.111111 |
| 6 | 1.0 | 0.215686 | 0.073785 | 0.777778 |
| 7 | 1,1 | 0.235294 | 0.140192 | 0.888889 |
| 8 | 5.0 | 1.000000 | 2.730051 | 5.22222 |

box_plot으로~

```
import seaborn as sns
import matplotlib.pyplot as plt
sns.set_theme(style="whitegrid")
plt.figure(figsize=(16, 6))
sns.boxplot(data=df_scaler, orient="h");
```

각 스케일러의 차이와 그 중 Robust 스케일러의 특징을 보자

