One Hot Encoding

#01. 범주형 속성(피쳐)의 처리 방법

레이블 인코딩(Label encoding)

문자열(범주형) 값을 0 부터 1씩 증가하는 값으로 변환(숫자형 카테고리)

숫자의 차이가 모델에 영향을 주지 않는 트리 계열 모델(의사결정나무, 랜덤포레스트)에 적용 가능

숫자의 차이가 모델에 영향을 미치는 선형 계열 모델(로지스틱회귀, SVM, 신경망)에는 사용하지 않음

더미화 - 원핫 인코딩(One-Hot encoding)

N개의 값을 갖는 피처를 N차원의 One-Hot 벡터로 표현되도록 변환

고유값들을 피처로 만들고 정답에 해당하는 열은 1로 나머진 0으로 표시

숫자의 차이가 모델에 영향을 미치는 선형 계열 모델(로지스틱회귀, SVM, 신경망)에서 범주형 데이터 변환시 라벨 인코딩 보다 원핫 인코딩을 사용

#02. 작업 준비

패키지 가져오기

```
import numpy as np
from pandas import DataFrame
```

```
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
```

#03. Label Encoding

원 데이터가 DataFrame 형식인 경우

```
df = DataFrame({'item' : ['TV', '냉장고', '전자레인지', '컴퓨터', 'TV', '선풍
df
```

	item
0	TV
1	냉장고
2	전자레인지
3	컴퓨터
4	TV
5	선풍기
6	선풍기
7	믹서
8	믹서

라벨링 수행

리턴되는 객체는 numpy 배열 타입이다.

```
le = LabelEncoder()
fit = le.fit_transform(df['item'])

print(type(fit))
print(fit)
```

```
<class 'numpy.ndarray'>
[0 1 4 5 0 3 3 2 2]
```

라벨링에 사용된 카테고리 값의 종류 확인

```
le.classes_
```

```
array(['TV', '냉장고', '믹서', '선풍기', '전자레인지', '컴퓨터'], dtype=objec
```

라벨링 결과에 따른 실제 값 확인

```
le.inverse_transform(fit)
```

```
array(['TV', '냉장고', '전자레인지', '컴퓨터', 'TV', '선풍기', '선풍기', '믹서
```

```
      dtype=object)
```

원 데이터가 데이터프레임 이외의 형식인 경우

```
origin = ['TV', '냉장고', '전자레인지', '컴퓨터', 'TV', '선풍기', '선풍기', ''

le = LabelEncoder()
fit = le.fit_transform(origin)

print(type(fit))
print(fit)
print(le.classes_)
print(le.inverse_transform(fit))
```

```
<class 'numpy.ndarray'>
[0 1 4 5 0 3 3 2 2]
['TV' '냉장고' '믹서' '선풍기' '전자레인지' '컴퓨터']
['TV' '냉장고' '전자레인지' '컴퓨터' 'TV' '선풍기' '선풍기' '믹서' '믹서']
```

#04. One Hot Encoding

원 데이터가 DataFrame인 경우

```
df = DataFrame({'item' : ['TV', '냉장고', '전자레인지', '컴퓨터', 'TV', '선풍
df
```

	item
0	TV
1	냉장고
2	전자레인지
3	컴퓨터
4	TV
5	선풍기
6	선풍기
7	믹서
8	믹서

```
encoder = OneHotEncoder(dtype='int64')
fit = encoder.fit_transform(df)
fit
```

```
<9×6 sparse matrix of type '<class 'numpy.int64'>'
        with 9 stored elements in Compressed Sparse Row format>
```

값의 종류 확인

```
encoder.categories_
```

```
[array(['TV', '냉장고', '믹서', '선풍기', '전자레인지', '컴퓨터'], dtype=obje
```

생성된 피쳐의 이름 확인

```
encoder.get_feature_names_out()
```

```
array(['item_TV', 'item_냉장고', 'item_믹서', 'item_선풍기', 'item_전자레인지
       'item_컴퓨터'], dtype=object)
```

변환 결과를 배열로 추출

```
fit.toarray()
```

```
array([[1, 0, 0, 0, 0, 0],
       [0, 1, 0, 0, 0, 0],
       [0, 0, 0, 0, 1, 0],
       [0, 0, 0, 0, 0, 1],
```

```
       [1, 0, 0, 0, 0, 0],
       [0, 0, 0, 1, 0, 0],
       [0, 0, 0, 1, 0, 0],
       [0, 0, 1, 0, 0, 0],
       [0, 0, 1, 0, 0, 0]], dtype=int64)
```

```
one_hot_df = DataFrame(fit.toarray(), columns=encoder.get_feature_names_
one_hot_df
```

	item_TV	item_냉장고	item_믹서	item_선풍기	item_전자레인지	item_컴퓨터
0	1	0	0	0	0	0
1	0	1	0	0	0	0
2	0	0	0	0	1	0
3	0	0	0	0	0	1
4	1	0	0	0	0	0
5	0	0	0	1	0	0
6	0	0	0	1	0	0
7	0	0	1	0	0	0
8	0	0	1	0	0	0