범주형 데이터 다루기 : Label Encoding & One-Hot Encoding

Label Encoding

Food Name	Categorical #	Calories
Apple	1	95
Chicken	2	231
Broccoli	3	50

→

One Hot Encoding

Apple	Chicken	Broccoli	Calories
1	0	0	95
0	1	0	231
0	0	1	50

인코딩을 사용하는 이유

- 분석을 위해 사용하는 모델들은 수치형 데이터만 다룰 수 있음
- 따라서, String형으로 된 범주형 변수는 모델이 인식할 수 있도록 일련의 변환 과정을 거쳐야 함 -> 이러한 과정을 인코딩(Encoding)이라고 함

범주형 데이터란?

- 수치형 변수 - 연령 변수와 같이 어떠한 범위 내 수치 값을 갖는 변수

- 범주형 변수 - 성별 변수와 같이 고정된 목록 중 하나의 값을 갖는 변수

> - 명목형 변수(Nominal) - 성별과 혈액형처럼 값이 달라짐에 따라 좋거나 나쁘다고 할 수 없는 경우
> - 순서형 변수(Oridinal) - 학점과 만족도 조사처럼 값이 커지거나 작아짐에 따라 좋거나 나쁘다고 할 수 있는 경우

```
import pandas as pd
import numpy as np

data = {
"Temperature": ["Hot", "Cold", "Very Hot", "Warm", "Hot", "Warm", "Warm", "Hot", "Hot", "Cold"],
"Color": ["Red", "Yellow", "Blue", "Blue", "Red", "Yellow", "Red", "Yellow", "Yellow", "Yellow"],
"Target": [1, 1, 1, 0, 1, 0, 1, 0, 1, 1]}

df = pd.DataFrame(data, columns = ["Temperature", "Color", "Target"])
df
```

	Temperature	Color	Target
0	Hot	Red	1
1	Cold	Yellow	1
2	Very Hot	Blue	1
3	Warm	Blue	0
4	Hot	Red	1
5	Warm	Yellow	0
6	Warm	Red	1
7	Hot	Yellow	0
8	Hot	Yellow	1
9	Cold	Yellow	1

```
# Label Encoding
from sklearn.preprocessing import LabelEncoder

print(df['Temperature'].unique())
print()

# LabelEncoder를 객체로 생성
encoder = LabelEncoder()

# fit, transform 메소드를 통한 레이블 인코딩
encoder.fit(df['Temperature'])

df["Temp_Label_Encoder"] = encoder.transform(df['Temperature'])
df
```

```
['Hot' 'Cold' 'Very Hot' 'Warm']
```

	Temperature	Color	Target	Temp_Label_Encoder
0	Hot	Red	1	1
1	Cold	Yellow	1	0
2	Very Hot	Blue	1	2
3	Warm	Blue	0	3
4	Hot	Red	1	1
5	Warm	Yellow	0	3
6	Warm	Red	1	3
7	Hot	Yellow	0	1
8	Hot	Yellow	1	1
9	Cold	Yellow	1	0

인코딩 기법 - One-Hot Encoding

- 각 카테고리를 0과 1로 구성된 벡터로 표현하는 기법

  - 이때, 카테고리의 수만큼 벡터가 생성되므로 각 카테고리가 새로운 변수가 되어 표현되며, 숫자의 크고 작은 특성(중요도)을 없앨 수 있음
  - 카테고리가 너무 많은 변수의 경우 데이터의 cardinality를 증가시켜 모델의 성능을 저하시킬 수 있다는 단점이 있음

```
from sklearn.preprocessing import OneHotEncoder

oh = OneHotEncoder()
encoder = oh.fit_transform(df['Temperature'].values.reshape(-1,1)).toarray() # 인코딩하기 전에 2차원 데이터로 변환

df_OneHot = pd.DataFrame(encoder, columns=["Temp_OneHot_Encoder_" + str(oh.categories_[0][i]) for i in range (len(oh.categories_[0]))])

df1 = pd.concat([df, df_OneHot], axis=1)
df1
```

	Temperature	Color	Target	Temp_Label_Encoder	Temp_OneHot_Encoder_Cold	Temp_OneHot_Encoder_Hot	Temp_OneHot_Encoder_V
0	Hot	Red	1	1	0.0	1.0	
1	Cold	Yellow	1	0	1.0	0.0	
2	Very Hot	Blue	1	2	0.0	0.0	
3	Warm	Blue	0	3	0.0	0.0	
4	Hot	Red	1	1	0.0	1.0	
5	Warm	Yellow	0	3	0.0	0.0	
6	Warm	Red	1	3	0.0	0.0	
7	Hot	Yellow	0	1	0.0	1.0	
8	Hot	Yellow	1	1	0.0	1.0	

<<<참조자료 사이트>>>

1.메타, 메타러닝이란 뭘까?
2."하나를 알려주면 열을 안다" [특별기획 AI 2030] ② 메타학습
3.EDA (Exploratory Data Analysis) 탐색적 데이터 분석
4.범주형 데이터 다루기 : Encoding 기법 (1)
5.데이터 전처리 : 레이블 인코딩과 원핫 인코딩
6.범주형 데이터 다루기 : Encoding 기법 (2)
7.Confusion Matrix(혼돈 행렬)과 분류 성능 평가 지표
8.파이썬 머신러닝 완벽가이드 - 평가, 정확도

9. 파이썬에서의 머신러닝 모델 평가 방법
10. 평가(정확도, 오차 행렬, 정밀도, 재현율)
11. 의미를 이해하는 통계학과 데이터 분석
12. ROC Curve
13. 파이썬 ROC 커브, AUC 면적 구하기 예제
