One-hot encoding 정리해보기

학습 목표

가. 정수 인코딩과 원핫 인코딩은 무엇인가?
나. scikit-learn, Pandas, Keras 라이브러리를 사용하여 파이썬에서 데이터를 레이블 인코딩에 대해 알아본다.
다. scikit-learn 및 Keras 라이브러리를 사용하여 OneHotEncoding 하는 방법 알아보기

학습내용

- 1. 간단 One-hot encoding 해보기
- 1. 개요
- 1. One-hot encoding 이란?
- 1. 왜 One-hot encoding를 사용하는가?
- 1. scikit-learn를 이용한 One-hot encoding 해보기
- 1. One-hot encoding with Keras (케라스 이용)
- 1. One-hot encoding with pandas

In [1]:
```
from IPython.display import display, Image
```

01. 기본 실습 - One-hot encoding

간단한 데이터를 준비하여, 목표 feature인 'target'를 labelencode 후, 이후, 결과값을 이용하여 one-hot-encoding를 수행한다.

가. 데이터 준비

In [2]:
```
### 01. 데이터 준비
import pandas as pd
data = { "feature1":[2,3,8,4],
         "feature2":[22,32,82,42],
         "target": ["b","c", "a", "d"]
       }
df = pd.DataFrame(data)
df
```

Out[2]:

	feature1	feature2	target
0	2	22	b
1	3	32	c
2	8	82	a
3	4	42	d

In [3]:
```
from sklearn import preprocessing
```

나. LabelEncoder하기

a,b,c,d가 숫자 0,1,2,3로 변경

In [4]:
```
label_encoder = preprocessing.LabelEncoder()
df['lbl_en'] = label_encoder.fit_transform(df['target'])
df
```

Out[4]:

	feature1	feature2	target	lbl_en
0	2	22	b	1
1	3	32	c	2
2	8	82	a	0
3	4	42	d	3

다. 행렬변경(4X1)

In [5]:
```
train_y = df['lbl_en'].values.reshape(len(df), 1)
train_y
```

Out[5]:
```
array([[1],
       [2],
       [0],
       [3]])
```

라. One-hot encoding 하기

- 범주형 변수를 one-hot 수치형 배열로 변환

In [6]:
```
onehot_encoder = preprocessing.OneHotEncoder(sparse=False)
train_y_onehot = onehot_encoder.fit_transform(train_y)
print(train_y_onehot)
print(train_y_onehot.shape)
```

```
[[0. 1. 0. 0.]
 [0. 0. 1. 0.]
 [1. 0. 0. 0.]
 [0. 0. 0. 1.]]
(4, 4)
```

실습

- A. OneHotEncoder의 sparse를 False로 하면 어떤 결과가 나오는가?
- B. 희소 행렬은 무엇을 의미하는가?
- C. 변경한 내용을 기존의 데이터에 붙여보기

02. 개요

- **A. 머신러닝 알고리즘은 범주형 데이터에서 직접적으로 작동하지 않는다.**
- **B. 범주형 데이터는 숫자로 변경되어야 함.**

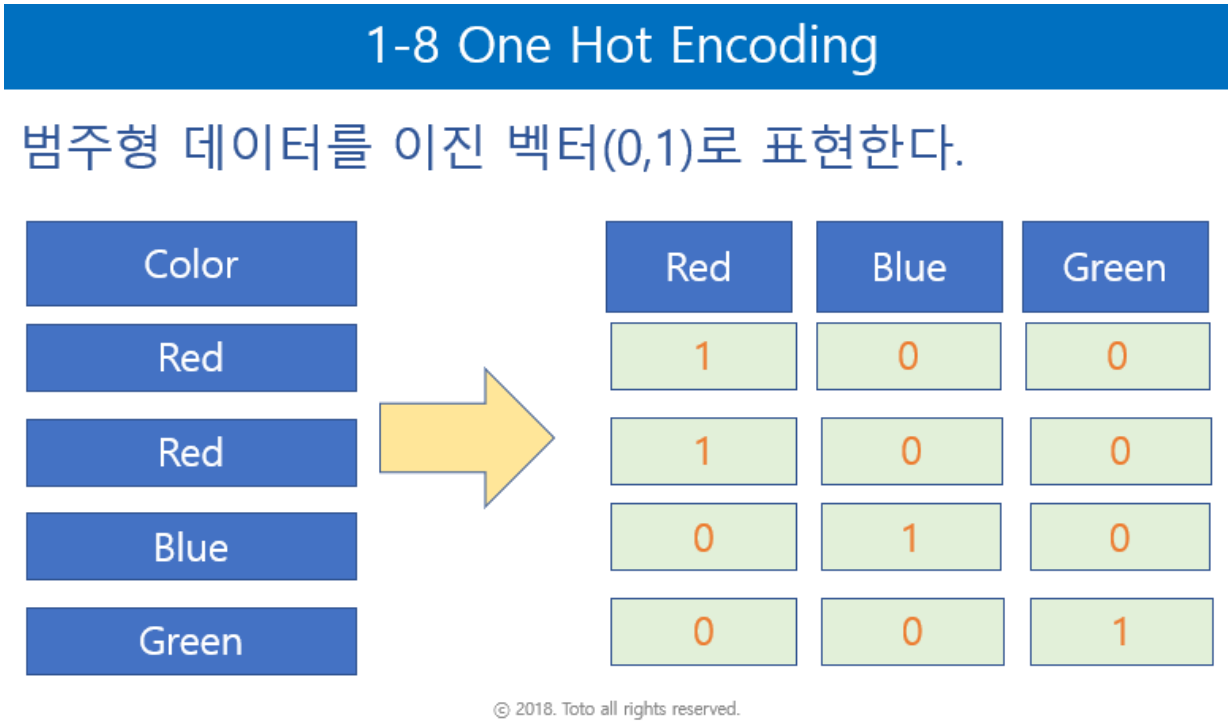
  `Categorical data must be converted to numbers.`
- **C. 신경망과 같은 심층적인 학습 방법을 사용할 때 적용.**

  `This applies when you are working with a sequence classification type problem and plan on using deep learning methods such as Long Short-Term Memory recurrent neural networks.`

03. One-hot encoding이란?

* 가. One-Hot Encoding은 범주형 변수를 바이너리벡터(0,1)로 표현한 것.
나. 작업 절차는
    A. 범주형 변수는 정수값으로 변경되어야 하고,
    B. 각각의 정수값은 해당되는 위치에 1로 표시되고 나머지는 0으로 표시.

In [7]:
```
display(Image(filename='img/onehotencoding.png'))
```



'red', 'red', 'blue', 'green'

정수로 encoding하기 (정보의 형태나 형식을 변환하는 처리방식)

0,0,1,2

one hot encoding하기

```
[1,0,0]
[1,0,0]
[0,1,0]
[0,0,1]
```

실습 1

spring, summer, autumn, winter을 레이블 인코딩, OneHotEncoding를 해보자.

04. 왜 One-hot encoding를 사용하는가?

가. 범주형 데이터를 숫자로 변경합니다. 단 이 데이터는 자연스러운 순서가 있다.
    하지만 순서가 없을 경우, 문제가 될 수 있습니다.
    (dog, cat, bird..)
나. 이 경우, 좀 더 표현력이 있는 One-hot encoding 방법을 이용하면 더 정밀한 예측을 가능하게 된다.

05. scikit-learn를 이용한 One-Hot Encoding

가. 우리는 4개의 레이블을 가지고 있다.

'spring', 'summer', 'autumn', 'winter'
We will assume the case where you have an output sequence of the labels

나. 10개의 데이터를 가지고 있다.

spring, spring, summer, spring, autumn, autumn, winter, spring, summer, autumn

다. scikit-learn 라이브러리(library)를 이용

LabelEncoder : label를 정수값으로 변경
OneHotEncoder : 정수로 인코딩된 값을 One Hot Encode로 만든다.

In [8]:
```
from numpy import array
from numpy import argmax
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import OneHotEncoder

data = ['spring', 'spring', 'summer', 'spring', 'autumn',
        'autumn', 'winter', 'spring', 'summer', 'autumn']
values = array(data)
print(values)

# integer encode
label_encoder = LabelEncoder()
integer_encoded = label_encoder.fit_transform(values)
print(integer_encoded)

# binary encode
onehot_encoder = OneHotEncoder(sparse=False)
integer_encoded = integer_encoded.reshape(len(integer_encoded), 1)
onehot_encoded = onehot_encoder.fit_transform(integer_encoded)
print(onehot_encoded)

# LabelEncoder에 입력하여 역변환 4번째 행의 값을 되돌리기
inverted = label_encoder.inverse_transform([argmax(onehot_encoded[4, :])])
print(inverted)
```

```
['spring' 'spring' 'summer' 'spring' 'autumn' 'autumn' 'winter' 'spring'
 'summer' 'autumn']
[1 1 2 1 0 0 3 1 2 0]
[[0. 1. 0. 0.]
 [0. 1. 0. 0.]
 [0. 0. 1. 0.]
 [0. 1. 0. 0.]
 [1. 0. 0. 0.]
 [1. 0. 0. 0.]
 [0. 0. 0. 1.]
 [0. 1. 0. 0.]
 [0. 0. 1. 0.]
 [1. 0. 0. 0.]]
['autumn']
```

06. One Hot Encode with Keras

케라스에서는 one hot encode를 위해 to_categorical() 함수를 제공한다.

In [9]:
```
from keras.utils import to_categorical
```

```
import numpy as np

# define example
data = [15,17,5,10,0]
dat = np.array(data)
print(dat)

# one hot encode
encoded = to_categorical(dat)
print(encoded)
```

```
[15 17  5 10  0]
[[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1.]
 [0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0.]
 [1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]]
```

In [10]:
```
encoded[0]
```

Out[10]:
```
array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0.,
       0.], dtype=float32)
```

In [11]:
```
# invert encoding
inverted = argmax(encoded[1])
print(inverted)
```

```
17
```

실습 2(scikit)

집을 선택할 때, 다음과 같은 유형의 조건이 있다.
Inside, Corner, FR2, CulDSac 이에 대한 정보를 레이블 인코딩, OneHotEncoding를 해보자.

실습 3 (keras)

Inside, Corner, FR2, CulDSac 이에 대한 정보를 레이블 인코딩, OneHotEncoding를 해보자.

07. One Hot Encode with Pandas

판다스에서는 one hot encode를 위해 get_dummies() 함수를 제공한다.

In [12]:
```
import pandas as pd
import os
```

In [13]:
```
demo_df = pd.DataFrame({"범주형_feature":['양말', '여우', '양말', '상자']})
display(demo_df)
```

	범주형_feature
0	양말
1	여우
2	양말
3	상자

In [14]:
```
onehot = pd.get_dummies(demo_df)
onehot
```

Out[14]:

	범주형_feature_상자	범주형_feature_양말	범주형_feature_여우
0	0	1	0
1	0	0	1
2	0	1	0
3	1	0	0

In [15]:
```
df = pd.concat([demo_df, onehot], axis=1)
df
```

Out[15]:

	범주형_feature	범주형_feature_상자	범주형_feature_양말	범주형_feature_여우
0	양말	0	1	0
1	여우	0	0	1
2	양말	0	1	0
3	상자	1	0	0