ch04 데이터 표현 특성공학

- One Hot Encoding 이해하기

학습 내용

- 01 기본 용어 이해
- 02 왜 사용하나?
- 03 레이블 인코딩, 원핫 인코딩 실습해 보기(1)
- 04 레이블 인코딩, 원핫 인코딩 실습해 보기(2)
- 05 군집알고리즘 정리해 보기

01 용어 이해해보기

1-1. 연속형, 범주형 feature

- 데이터가 실수형 - 연속형 feature
- 데이터가 정해진 값 - 범주형 feature, 이산형 feature

1-2. 특성 공학(feature engineering)

- 특정 애플리케이션의 가장 적합한 데이터 표현 찾기
- 올바른 데이터 표현은 지도학습 모델에서 적절한 매개변수를 선택하는 것보다 성능에 매우 중요.

1-3. Label Encoding을 알아보기

- A. 머신러닝 알고리즘은 범주형 데이터에서 직접적으로 작동하지 않는다.
- B. 범주형 데이터는 숫자로 변경되어야 함.
  - 범주형 문자를 숫자로 변경해 주는 것.
    - 국가명이 만약 US, KR, UK, JPN등이라면 이를 숫자로 0,1,2,3로 변경해 준다.
- 파이썬 라이브러리 sklearn에서 LabelEncoder의 함수를 사용

1-4. What is One Hot Encoding?(One Hot Encoding은 무엇인가?)

가. One Hot Encoding은 머신러닝 알고리즘에서 더 나은 예측을 위해 제공되는 하나의 과정입니다.
나. One-Hot Encoding은 범주형 변수를 바이너리벡터(0,1)로 표현한 것.
다. Label Encoding이 범주형 구분을 숫자로 변경하는 것이라면, OneHotEncoding은

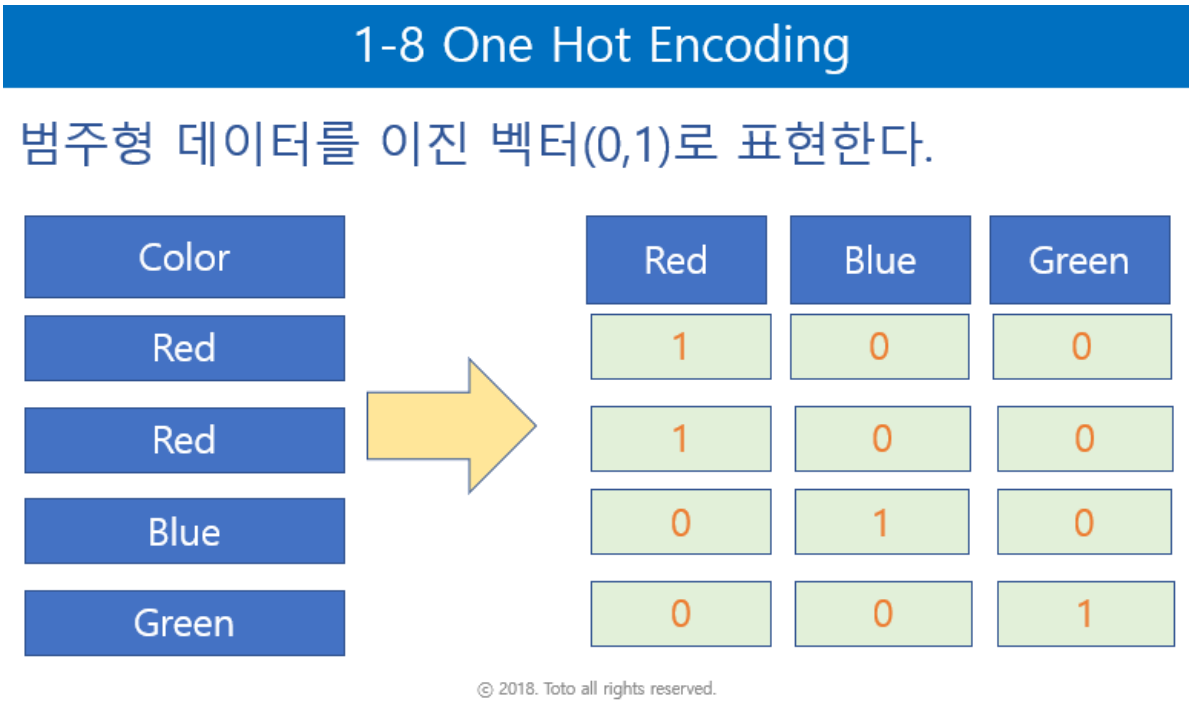
```
KR  => ( 1 , 0, 0, 0 )
US  => ( 0, 1, 0, 0 )
UK  => ( 0, 0, 1, 0 )
CN  => ( 0 , 0, 0, 1 )
```

로 벡터의 요소로 변경하는 것이다.

- 원핫인코딩을 다른말로 **가변수(dummy variable)**라고도 한다.
- 가변수는 범주형 변수를 0 또는 1값을 하나 이상의 새로운 특성으로 변경한 값이다.

In [2]:

```
from IPython.display import display, Image
display(Image(filename='img/onehotencoding.png'))
```



1-5. Why do you need one hot encoding?

(왜 필요할까?)

- Label 인코딩에 오류 부분(순서 개념이 있을 수 있음) 보완

Label 인코딩의 오류

Label 의 인코딩의 문제는 범주값이 높을수록 카테고리가 더 우수하다고 가정합니다.

(가) 범주형 값에 의해 가장 가치 있는 모델은 값이 높은 값이 가치있다고 생각합니다.
VW > Acura > Honda이다.
- 이 내용은 오류가 발생합니다. 이 값을 가지고 모델을 예측한다는 것은 많은 오류가 있다.
(나) 하지만 순서가 없을 경우, 문제가 될 수 있습니다.
- (dog, cat, bird..)
(다) 이 경우, 표현력이 있는 one-hot encoding를 이용하면 더 정밀한 예측이 가능해 질 수 있다.

02 왜 사용하나?

- 머신러닝이나 딥러닝 적용시에 문자를 이해가 어렵기에 해당 모델에 맞는 형태(숫자나 벡터로)로 만들어주어야 한다.

03 레이블 인코딩, 원핫 인코딩 실습해 보기(1)

In [3]:

```
### 01. 데이터 준비
import pandas as pd
data = { "eng": ["b", "c", "a", "d"] }
df = pd.DataFrame(data)
print(type(df))
df
```

```
<class 'pandas.core.frame.DataFrame'>
```

Out[3]:

	eng
0	b
1	c
2	a
3	d

In [4]:

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

LabelEncoder 사용하기

- LabelEncoder()
  - [].fit_transform([적용할열])

In [5]:

```
en_x = LabelEncoder()
df['라벨인코딩'] = en_x.fit_transform(df['eng'])
df
```

Out[5]:

	eng	라벨인코딩
0	b	1
1	c	2
2	a	0
3	d	3

데이터를 전처리

- OneHotEncoder() 적용을 위해 행렬로 변경

원핫 인코딩(OneHotEncoding) 실습

- OneHotEncoder()
  - [].fit_transform([적용할열])

In [6]:

```
df['라벨인코딩'].values
```

Out[6]:

```
array([1, 2, 0, 3])
```

In [7]:

```
onehot = OneHotEncoder()
val = df['라벨인코딩'].values.reshape(-1,1) # OneHotEncoder()를 사용을 위한 적합한 값으로 변경.
y = onehot.fit_transform( val ).toarray()    # 값을 변경후, 배열로 만들어준다.
y
```

Out[7]:

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

In [8]:

```
onehot_val = pd.DataFrame(y, dtype=int)
onehot_val
```

Out[8]:

	0	1	2	3
0	0	1	0	0
1	0	0	1	0
2	1	0	0	0
3	0	0	0	1

In [9]:

```
df_new = pd.concat([df, onehot_val], axis=1)
df_new
```

Out[9]:

	eng	라벨인코딩	0	1	2	3
0	b	1	0	1	0	0
1	c	2	0	0	1	0
2	a	0	1	0	0	0
3	d	3	0	0	0	1

04 레이블 인코딩, 원핫 인코딩 실습해 보기(2)

In [10]:

```
data = { "회사명": ["MS","Apple", "Google", "Google"]}
df1 = pd.DataFrame(data)
df2 = df1.copy()
df2
```

Out[10]:

	회사명
0	MS
1	Apple
2	Google
3	Google

In [11]:

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

In [12]:

```
df1['회사명']
```

Out[12]:

```
0       MS
1    Apple
2   Google
3   Google
Name: 회사명, dtype: object
```

In [13]:

```
### LabelEncoder
encoder_x = LabelEncoder()
df1['lbl_en'] = encoder_x.fit_transform(df1['회사명'])  #
df1
```

Out[13]:

	회사명	lbl_en
0	MS	2
1	Apple	0
2	Google	1
3	Google	1

In [14]:

```
df1['lbl_en'].values
```

Out[14]:

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

OneHotEncoding

In [15]:

```
onehot = OneHotEncoder()
y = onehot.fit_transform(df1['lbl_en'].values.reshape(-1,1) ).toarray()
print(y)
```

```
[[0. 0. 1.]
 [1. 0. 0.]
 [0. 1. 0.]
 [0. 1. 0.]]
```

In [16]:

```
# 변경된 값을 DataFrame형태로 변경
dx = pd.DataFrame(y, dtype=int)
dx
```

Out[16]:

	0	1	2
0	0	0	1
1	1	0	0
2	0	1	0
3	0	1	0

In [17]:

```
df1_new = pd.concat([df1, dx], axis=1)
df1_new
```

Out[17]:

	회사명	lbl_en	0	1	2
0	MS	2	0	0	1
1	Apple	0	1	0	0
2	Google	1	0	1	0
3	Google	1	0	1	0

05 Keras를 활용한 원핫 인코딩

- 케라스에서는 one hot encode를 위해 to_categorical() 함수를 제공한다.

In [19]:

```
from tensorflow.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 [21]:

```
# invert encoding
inverted = np.argmax(encoded[1])
print(inverted)
```

```
17
```

실습

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

06. Pandas를 활용한 원핫 인코딩

In [22]:

```
import pandas as pd
import os
```

In [23]:

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

	범주형_feature
0	양말
1	여우
2	양말
3	상자

In [24]:

```
onehot = pd.get_dummies(demo_df)
onehot
```

Out[24]:

	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 [25]:

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

Out[25]:

	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

과제

- 내가 좋아하는 과일을 딕셔너리 형태로 만들고, 이를 원핫 인코딩으로 만들어 보자.
  - scikit-learn의 클래스 활용하기
  - keras를 활용해 보기
  - pandas를 활용해 보기

History

- ver 1.01 2021-10 update