ch04 데이터 표현 특성공학

- One Hot Encoding 이해하기

학습 내용

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

01 용어 이해해보기

A. 연속형, 범주형 feature

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

B. 특성 공학(feature engineering)

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

C. What is One Hot Encoding?(One Hot Encoding은 무엇인가?)

```
가. One Hot Encoding은 머신러닝 알고리즘에서 더 나은 예측을 위해 제공되는 하나의 과정입니
다.
나. 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.

D. Label Encoding을 알아보기

- 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

E. 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.

범주형 값에 의해 가장 가치 있는 모델은
VW > Acura > Honda이다.
평균을 계산해서 확인하면 평균은 2이고, 이것이 의미하는 바는 VM와 Honda의 평균은 Acura이다.
이 내용은 오류가 발생합니다. 이 값을 가지고 모델을 예측한다는 것은 많은 오류가 있다.

02 왜 사용하나?

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

03 라벨 인코딩, 원핫 인코딩 실습해 보기(1)

In [32]:

```
### 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[32]:

	eng
0	b
1	c
2	a
3	d

In [33]:

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

LabelEncoder 사용하기

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

In [34]:

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

Out[34]:

	eng	라벨인코딩
0	b	1
1	c	2
2	a	0
3	d	3

데이터를 전처리

- OneHotEncoder() 적용을 위해 행렬로 변경

In [35]:

```
val = df['라벨인코딩']
new_val = val.values
print(new_val.shape)
n = new_val.reshape(-1,1)
n.shape
```

```
(4,)
```

Out[35]:

```
(4, 1)
```

원핫 인코딩(OneHotEncoding) 실습

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

In [36]:

```
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[36]:

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

In [37]:

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

Out[37]:

	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 [38]:

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

Out[38]:

	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 라벨 인코딩, 원핫 인코딩 실습해 보기(2)

In [39]:

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

Out[39]:

	companyName
0	MS
1	Apple
2	Google
3	Google

In [40]:

```
df1.values
```

Out[40]:

```
array([['MS'],
       ['Apple'],
       ['Google'],
       ['Google']], dtype=object)
```

In [41]:

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

In [42]:

```
### LabelEncoder
df1_val = df1.values
print("데이터 값 (x) :")
print(df1_val)
encoder_x = LabelEncoder()
df1['encoding'] = encoder_x.fit_transform(df1['companyName'])  #
df1
```

```
데이터 값 (x) :
[['MS']
 ['Apple']
 ['Google']
 ['Google']]
```

Out[42]:

	companyName	encoding
0	MS	2
1	Apple	0
2	Google	1
3	Google	1

In [43]:

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

```
[2 0 1 1]
[[0. 0. 1.]
 [1. 0. 0.]
 [0. 1. 0.]
 [0. 1. 0.]]
```

In [44]:

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

Out[44]:

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

In [45]:

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

Out[45]:

	companyName	encoding	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

과제

- 내가 좋아하는 과일을 딕셔너리 형태로 만들고, 이를 원핫 인코딩으로 만들어 보자.

History

- ver 1.00 2020-05 update