OneHotEncoding

학습 목표

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

학습내용

01. 간단 OneHotEncoding 해보기

02. 개요

03. One Hot Encoding이란?

04. 왜 One Hot Encoding를 사용하는가?

Why Use a One Hot Encoding?

05. 'hello world'를 onehotencoding하기

06. scikit-learn를 이용한 One Hot Encode 해보기

One Hot Encode with scikit-learn

07. One Hot Encode with Keras (케라스 이용)

01. 간단 OneHotEncoding 해보기

간단한 데이터를 준비하여, 목표 feature인 'target'를 labelencode 후, 이 후, 결과값을 이용하여 onehotencode를 수행한다.

가. 데이터 준비

In [23]:
```
### 01. 데이터 준비
import pandas as pd
data = { "target": ["b","c", "a", "d"],
         "Value1":[2,3,8,4],
         "Value2":[22,32,82,42]
       }
df = pd.DataFrame(data)
df
```

Out[23]:

	Value1	Value2	target
0	2	22	b
1	3	32	c
2	8	82	a
3	4	42	d

In [24]:
```
from sklearn import preprocessing
label_encoder = preprocessing.LabelEncoder()
onehot_encoder = preprocessing.OneHotEncoder()
```

나. LabelEncoder하기

a,b,c,d가 숫자 0,1,2,3로 변경

In [25]:
```
train_y = label_encoder.fit_transform(df['target'])
print(train_y)
print(train_y.shape)
```

```
[1 2 0 3]
(4,)
```

다. 행렬변경(4X1)

In [26]:
```
train_y = train_y.reshape(len(train_y), 1)
print(train_y.shape)
```

```
(4, 1)
```

라. onehotencoding 하기

In [27]:
```
train_y = onehot_encoder.fit_transform(train_y)
print(train_y)
print(train_y.shape)
```

```
  (0, 1)        1.0
  (1, 2)        1.0
  (2, 0)        1.0
  (3, 3)        1.0
(4, 4)
```

02. 개요

A. 머신러닝 알고리즘은 범주형 데이터에서 직접적으로 작동하지 않는다.

B. 범주형 데이터는 숫자로 변경되어야 함.

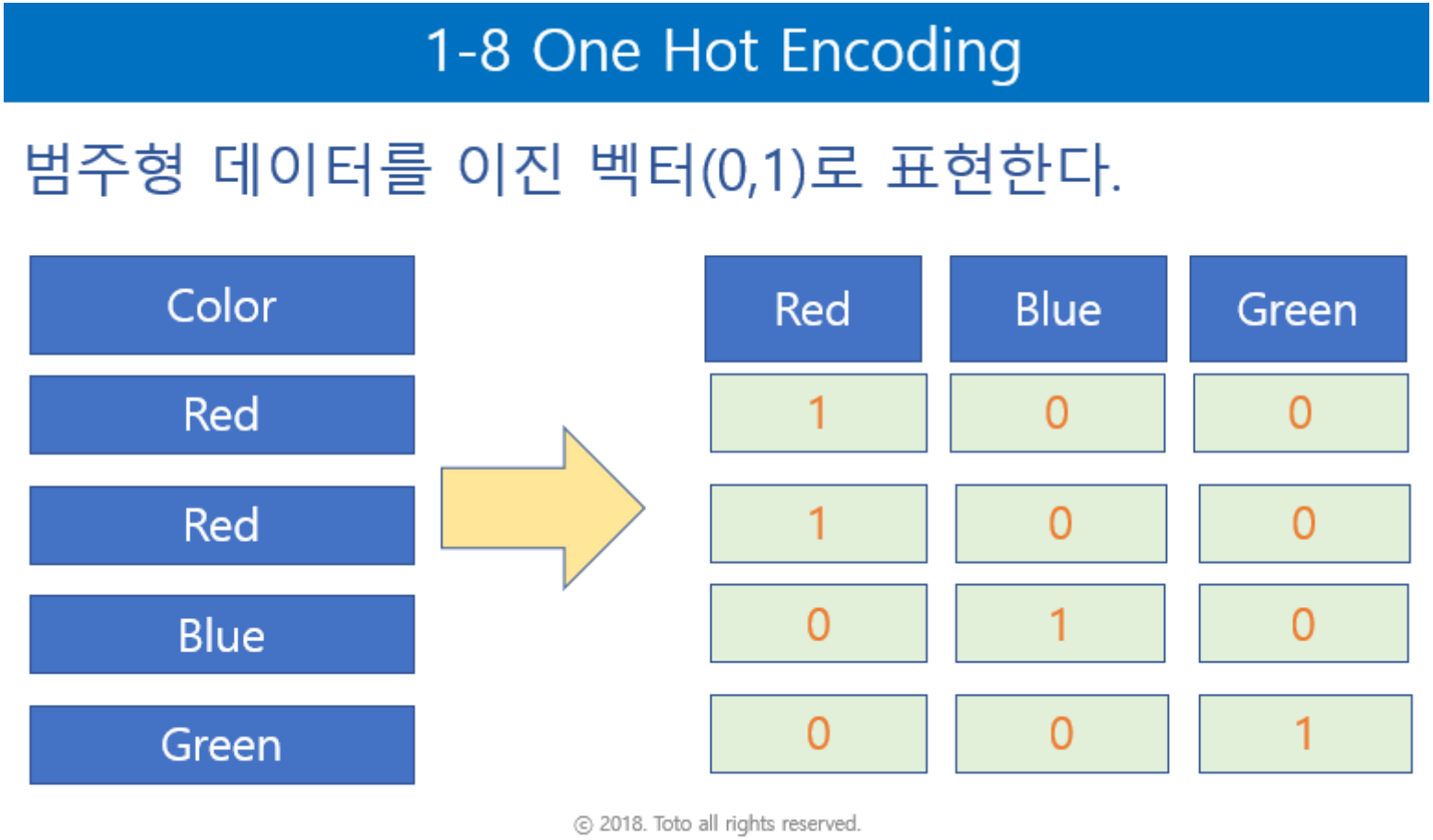
Categorical data must be converted to numbers.

C. 신경망과 같은 심층적인 학습 방법을 사용할 때 적용함.

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. OneHotEncoding은 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.



'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 OneHotEncoding 방법을 이용하면 더 정밀한 예측을 가능하게 된다.

05. 'hello world'를 원핫인코딩하기

In [29]:
```
from numpy import argmax
# define input string
data = 'hello world'
print(data)
# define universe of possible input values
alphabet = 'abcdefghijklmnopqrstuvwxyz '
# define a mapping of chars to integers
char_to_int = dict((c, i) for i, c in enumerate(alphabet))
int_to_char = dict((i, c) for i, c in enumerate(alphabet))
# integer encode input data
integer_encoded = [char_to_int[char] for char in data]
print(integer_encoded)
# one hot encode
onehot_encoded = list()
for value in integer_encoded:
    letter = [0 for _ in range(len(alphabet))]
    letter[value] = 1
    onehot_encoded.append(letter)
print(onehot_encoded)
# invert encoding
inverted = int_to_char[argmax(onehot_encoded[0])]
print(inverted)
```

```
hello world
[7, 4, 11, 11, 14, 26, 22, 14, 17, 11, 3]
[[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, 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, 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,
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, 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, 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, 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, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
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, 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, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0]]
h
```

06. scikit-learn를 이용한 One Hot Encode

가. 우리는 3개의 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 [39]:
```
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']

C:₩Users₩WITHJS₩Anaconda3₩lib₩site-packages₩sklearn₩preprocessing₩label.py:151: DeprecationWarning: The truth value of an empty ar
```

```
ray is ambiguous. Returning False, but in future this will result in an error. Use `array.size > 0` to check that an array is not
empty.
  if diff:
```

07. One Hot Encode with Keras

케라스에서는 one hot encode를 위해 to_categorical() 함수를 제공한다.

In [40]:
```
from numpy import array
from numpy import argmax
from keras.utils import to_categorical
# define example
data = [2, 3, 2, 0, 3, 2, 0, 1, 0, 1]
data = array(data)
print(data)

# one hot encode
encoded = to_categorical(data)
print(encoded)

# invert encoding
inverted = argmax(encoded[0])
print(inverted)
```

```
C:₩Users₩WITHJS₩Anaconda3₩lib₩site-packages₩h5py₩__init__.py:36: FutureWarning: Conversion of the second argument of issubdtype fr
om `float` to `np.floating` is deprecated. In future, it will be treated as `np.float64 == np.dtype(float).type`.
  from ._conv import register_converters as _register_converters
Using TensorFlow backend.

[1 3 2 0 3 2 2 1 0 1]
[[0. 1. 0. 0.]
 [0. 0. 0. 1.]
 [0. 0. 1. 0.]
 [1. 0. 0. 0.]
 [0. 0. 0. 1.]
 [0. 0. 1. 0.]
 [0. 0. 1. 0.]
 [0. 1. 0. 0.]
 [1. 0. 0. 0.]
 [0. 1. 0. 0.]]
1
```

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.

In [ ]: