2.2 범주형 데이터 처리

범주형 데이터의 정의

범주형 데이터는 'A', 'B', 'C'와 같이 종류를 표시하는 데이터를 말한다. 카테고리(category(데이터라고도 부른다. 다음과 같은 데이터는 모두 범주형 데이터의 예다.

- 성별: 남자, 여자
- 혈액형: A, B, O, AB
- 이름: 홍길동, 성춘향, ...
- 주소: 서울, 부산, 대전, ...

반드시 문자만 범주형 데이터인 것은 아다. 예를 들어 소속을 나타내는 '1반', '2반', '3반'과 같은 데이터는 숫자로 표현된 값이지만 '1'이라는 글자를 이용한 것 뿐이지 숫자로서의 의미는 없다. 즉, '2'라는 값이 '1'이라는 값보다 2배 더 크다는 뜻이 아니므로 이 경우는 범주형 값으로 보아야 한다.

범주형 데이터의 변형

대부분의 데이터 분석 모형은 숫자만 입력으로 받을 수 있기 때문에 범주형 데이터는 숫자로 변환해야 한다. 범주형 데이터를 숫자로 변환하는 방법은 두가지다.

- 더미변수화
- 카테고리 임베딩

더미변수화

더미변수(dummy variable)는 0 또는 1만 가지는 값으로 어떤 특징이 존재하는가 존재하지 않는가를 표시한다. 다음과 같은 명칭으로도 불린다.

- 이진지시자(Boolean indicator)
- 이진변수(binary variable)
- 지시변수(indicator variable)
- 설계변수(design variable)
- 처치(treatment)

카테고리값을 더미변수화하면 복수의 더미변수 벡터로 표시한다. 예를 들어 성별 x는 2개의 더미변수 (d_1, d_2)로 표현할 수 있다. 더미변수 d_1는 남자면 1, 여자면 0이 되는 값이다. 더미변수 d_2는 남자면 0, 여자면 1이 되는 값이다.

$$
\begin{aligned}
x = \text{남자} &\rightarrow d_1 = 1, d_2 = 0 \\
x = \text{여자} &\rightarrow d_1 = 0, d_2 = 1
\end{aligned}
$$

혈액형은 4개의 더미변수 (d_1, d_2, d_3, d_4)로 표현할 수 있다. d_1은 혈액형이 A형인지 아닌지를 나타내는 더미변수이고 d_2은 혈액형이 B형인지 아닌지를 나타내는 더미변수, d_3, d_4는 각각 혈액형이 O형인지 아닌지를 나타내는 더미변수, AB형인지 아닌지를 나타내는 더미변수다.

$$
\begin{aligned}
x = \text{A형} &\rightarrow d_1 = 1, d_2 = 0, d_3 = 0, d_4 = 0 \\
x = \text{B형} &\rightarrow d_1 = 0, d_2 = 1, d_3 = 0, d_4 = 0 \\
x = \text{O형} &\rightarrow d_1 = 0, d_2 = 0, d_3 = 1, d_4 = 0 \\
x = \text{AB형} &\rightarrow d_1 = 0, d_2 = 0, d_3 = 0, d_4 = 1
\end{aligned}
$$

위 예제들에서 볼 수 있듯이 1부터 K까지의 값을 가질 수 있는 범주형값은 K개의 더미변수 벡터로 표시할 수 있다. 각 더미변수는 특정한 하나의 카테고리값인가 아닌가를 나타내는 지시자(indicator)가 된다.

patsy 패키지를 사용한 더미변수화

patsy 패키지의 `dmatrix()` 함수는 데이터프레임의 문자열 범주값을 더미변수로 바꿔준다. 예를 들어 다음과 같이 성별을 나타내는 "Male", "Female"값 데이터가 있는 경우,

In [1]:

```
df1 = pd.DataFrame(["Male", "Female"], columns=["x"])
df1
```

Out[1]:

	x
0	Male
1	Female

`dmatrix()` 함수에 넣으면 `x[Female]`, `x[Male]` 이라는 두 개의 더미변수를 만들어준다. `x[Female]` 는 값이 여자인지 아닌지를 나타내는 더미변수고 `x[Male]` 는 값이 남자인지 아닌지를 나타내는 더미변수다. 주의할 점은 **formula 문자열에 항상 `+ 0` 을 추가하여 상수항이 생기지 않도록 해야 한다.** 만약 이렇게 하지 않으면 뒤에서 설명할 축소랭크(reduce-rank)방식이라는 다른 방식으로 더미변수를 만들게 된다.

In [2]:

```
from patsy import dmatrix

dmatrix("x + 0", df1)
```

Out[2]:

```
DesignMatrix with shape (2, 2)
  x[Female]  x[Male]
          0        1
          1        0
  Terms:
    'x' (columns 0:2)
```

다음은 혈액형 데이터를 더미변수로 바꾸는 예제 코드다.

In [3]:

```
df2 = pd.DataFrame(["A", "B", "AB", "O"], columns=["x"])
df2
```

Out[3]:

	x
0	A
1	B
2	AB
3	O

각각의 범주값에 대응하는 더미변수는 알파벳 순서로 정해진다. 혈액형의 경우에는 d_1이 A형, d_2이 AB형, d_3이 B형, d_4이 O형이다.

In [4]:

```
dmatrix("x + 0", df2)
```

Out[4]:

```
DesignMatrix with shape (4, 4)
  x[A]  x[AB]  x[B]  x[O]
     1      0     0     0
     0      0     1     0
     0      1     0     0
     0      0     0     1
  Terms:
    'x' (columns 0:4)
```

데이터가 범주형 값이지만 정수로 표시된 경우에는 `C()` 연산자를 이용하여 범주형 값임을 명시적으로 지정할 수 있다.

In [5]:

```
df3 = pd.DataFrame([1, 2, 3, 4], columns=["x"])
df3
```

Out[5]:

	x
0	1
1	2
2	3
3	4

In [6]:

```
dmatrix("C(x) + 0", df3)
```

Out[6]:

```
DesignMatrix with shape (4, 4)
  C(x)[1]  C(x)[2]  C(x)[3]  C(x)[4]
        1        0        0        0
        0        1        0        0
        0        0        1        0
        0        0        0        1
  Terms:
    'C(x)' (columns 0:4)
```

`C()` 연산자를 사용하면 각 범주값이 대응하는 더미변수의 순서도 바꿀 수 있다. 예를 들어 혈액형의 경우, d_1이 A형, d_2이 B형, d_3이 AB형, d_4이 O형으로 만들고 싶으면 다음과 같이 `level` 인수를 사용한다.

In [7]:

```
dm = dmatrix("C(x, levels=['A', 'B', 'AB', 'O']) + 0", df2)
dm
```

Out[7]:

```
DesignMatrix with shape (4, 4)
  Columns:
    ["C(x, levels=['A', 'B', 'AB', 'O'])[A]",
     "C(x, levels=['A', 'B', 'AB', 'O'])[B]",
     "C(x, levels=['A', 'B', 'AB', 'O'])[AB]",
     "C(x, levels=['A', 'B', 'AB', 'O'])[O]"]
  Terms:
    "C(x, levels=['A', 'B', 'AB', 'O'])" (columns 0:4)
  (to view full data, use np.asarray(this_obj))
```

In [8]:

```
np.asarray(dm)
```

Out[8]:

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

축소랭크 방식

지금까지 설명한 더미변수 방식을 **풀랭크(full-rank)** 방식이라고 한다. 이와 달리 **축소랭크(reducec-rank) 방식에서는 특정한 하나의 범주값을 기준값(reference, baseline)으로 하고 기준값에 대응하는 더미변수의 가중치는 항상 1으로 놓는다.** 다른 범주형 값을 가지는 경우는 기준값 더미변수도 1이고 추가적인 특성을 나타내는 더미변수도 1인 것으로 간주한다. 예를 들어 다음 축소랭크 방식은 $x = A$를 기준값으로 하는 경우이다.

$$
\begin{aligned}
x = A &\rightarrow d_1 = 1, d_2 = 0, d_3 = 0, d_4 = 0 \\
x = B &\rightarrow d_1 = 1, d_2 = 1, d_3 = 0, d_4 = 0 \\
x = AB &\rightarrow d_1 = 1, d_2 = 0, d_3 = 1, d_4 = 0 \\
x = O &\rightarrow d_1 = 1, d_2 = 0, d_3 = 0, d_4 = 1
\end{aligned}
$$

반대로 $x = B$를 기준값으로 하면 다음과 같아진다.

$$
\begin{aligned}
x = A &\rightarrow d_1 = 1, d_2 = 1, d_3 = 0, d_4 = 0 \\
x = B &\rightarrow d_1 = 0, d_2 = 1, d_3 = 0, d_4 = 0 \\
x = AB &\rightarrow d_1 = 0, d_2 = 1, d_3 = 1, d_4 = 0 \\
x = O &\rightarrow d_1 = 0, d_2 = 1, d_3 = 0, d_4 = 1
\end{aligned}
$$

`dmatrix()` 함수를 사용할 때 formula 문자열에 `+ 0` 을 생략하면 축소랭크 방식으로 더미변수를 만든다. 기준이 되는 더미변수의 이름이 `Intercept` 가 된다. 기준이 되는 더미변수는 알파벳 순서로 가장 앞의 값이 된다. 다음 예제코드에서는 `Female` 이 기준 범주값이 된다.

In [9]:

```
dmatrix("x", df1)
```

Out[9]:

```
DesignMatrix with shape (2, 2)
  Intercept  x[T.Male]
          1          1
          1          0
  Terms:
    'Intercept' (column 0)
    'x' (column 1)
```

만약 기준 범주값을 다른 값으로 바꾸려면 `Treatment()` 함수를 formula 문자열에서 사용한다.

In [10]:

```
dmatrix("C(x, Treatment('Male'))", df1)
```

Out[10]:

```
DesignMatrix with shape (2, 2)
  Intercept  C(x, Treatment('Male'))[T.Female]
          1                                  0
          1                                  1
  Terms:
    'Intercept' (column 0)
    "C(x, Treatment('Male'))" (column 1)
```

혈액형 데이터를 축소랭크방식으로 더미변수화하면 다음과 같다. 기준 범주값은 `A` 다.

In [11]:

```
dmatrix("x", df2)
```

Out[11]:

```
DesignMatrix with shape (4, 4)
  Intercept  x[T.AB]  x[T.B]  x[T.O]
          1        0       0       0
          1        0       1       0
          1        1       0       0
          1        0       0       1
  Terms:
    'Intercept' (column 0)
    'x' (columns 1:4)
```

두 개의 범주형 변수가 있는 경우

두 개의 범주형 변수가 있는 경우에는 다음과 같은 두 가지 방법을 사용할 수 있다.

- 통합 축소형 방식
- 상호작용 방식

통합 축소형 방식은 각각의 변수를 축소형으로 기준값을 더미변수화한다. 다만 기준값을 나타내는 더미변수는 변수의 갯수와 상관없이 하나로 통합한다.

예를 들어 A, B 값을 가지는 범주형 변수 x1과 X, Y 값을 가지는 범주형 변수 x2가 있는 경우를 예로 들어보자.

In [12]:

```
df4 = pd.DataFrame([["A", "X"], ["B", "X"], ["A", "Y"], ["B", "Y"]], columns=["x1", "x2"])
df4
```

Out[12]:

	x1	x2
0	A	X
1	B	X
2	A	Y
3	B	Y

통합 축소형 방식에서는 다음과 같이 3개의 더미변수 (d_1, d_2, d_3)를 만든다. d_1은 x1이 A, x_2가 X라는 기준값을 나타낸다. d_2은 x1이 A가 아닌 B라는 것을 표시한다. d_3은 x2이 X가 아닌 Y라는 것을 표시한다.

$$
\begin{aligned}
x_1 = A, x_2 = X \quad &\rightarrow \quad d_1 = 1, d_2 = 0, d_3 = 0 \\
x_1 = B, x_2 = X \quad &\rightarrow \quad d_1 = 1, d_2 = 1, d_3 = 0 \\
x_1 = A, x_2 = Y \quad &\rightarrow \quad d_1 = 1, d_2 = 0, d_3 = 1 \\
x_1 = B, x_2 = Y \quad &\rightarrow \quad d_1 = 1, d_2 = 1, d_3 = 1
\end{aligned}
$$

In [13]:

```
dmatrix("x1 + x2", df4)
```

Out[13]:

```
DesignMatrix with shape (4, 3)
  Intercept  x1[T.B]  x2[T.Y]
          1        0        0
          1        1        0
          1        0        1
          1        1        1
  Terms:
    'Intercept' (column 0)
    'x1' (column 1)
    'x2' (column 2)
```

상호작용 방식은 두 범주형 변수를 곱해서 각각의 변수의 조합을 나타내는 새로운 범주형 변수를 만드는 방식이다. 즉 앞의 예제에서는 범주형 독립변수가 하나가 되고 대신 범주형 값이 두 독립변수의 범주형 값들의 조합인 AX, BX, AY, BY의 네가지가 된다.

In [14]:

```
dmatrix("x1:x2 + 0", df4)
```

Out[14]:

```
DesignMatrix with shape (4, 4)
  x1[A]:x2[X]  x1[B]:x2[X]  x1[A]:x2[Y]  x1[B]:x2[Y]
            1            0            0            0
            0            1            0            0
            0            0            1            0
            0            0            0            1
  Terms:
    'x1:x2' (columns 0:4)
```

카테고리 임베딩

카테고리 임베딩(embedding)은 범주값 대신 범주값의 특성을 나타내는 연속값 혹은 연속값 벡터를 사용하는 방법이다.

예를 들면 운동선수의 이름을 나타내는 범주값의 경우 해당 운동선수의 나이, 연봉, 신체능력치 등을 대신 사용한다. 또 다른 예로 지역명을 나타내는 범주값의 경우에는 해당 지역의 면적, 인구수 등을 사용할 수 있다.

하지만 임베딩을 사용하는 경우에는 데이터 분석 목적에 맞게 특징을 선택해야 하고 현재 가지고 있는 데이터가 아닌 외부의 추가적인 데이터를 조사해야 한다는 부담이 있다.