Concat 과 Merge

Contents

Concat 과 Merge 는 두개 이상의 DataFrame/Series 을 키 값(매칭을 위한 값)으로 합칠 때 쓰는 메소드입니다.

먼저 concat 를 해 보겠습니다. concat 는 axis 라는 인수를 사용해서 위-아래로 합할 것인지, 좌-우로 합할 것인지 알려줍니다. 좌-우로 합치는 경우는 index(행) 를 기준으로 하고, 위-아래로 합치는 경우는 column(열) 을 기준으로 합니다. 다른 인수로는 join 이 있습니다. 주로 axis=1 로 병합(좌-우)하는 경우가 많은데요. 양쪽 데이터셋에 동시에 존재하는 index 만으로 합칠 때는 join='inner' 를 넣어주고, 모든 index 를 남기고 싶을 때는 join='outer' 를 넣어줍니다. 아래 예제에서 Series s1 과 Series s2 의 index 가 동일하므로, 'inner' 나 'outer' 로 합쳐도 동일한 결과가 나옵니다. 두개의 데이터셋이 같은 지 체크하는 메소드는 equal 입니다. 체크한 결과 True 를 얻었습니다. 참고로 함수의 ()안에 커서를 놓고, 'Shift+Tab' 를 하면 활용 가능한 모든 인수와 설명이 나옵니다.

```
import pandas as pd

s1 = pd.Series([1,2,3,4,5], name='s1') # 두 Series를 합친 후, 어느 Series 에서 알기위해 이름 지정
s2 = pd.Series(['a','b','c','d','e'], name='s2')

horizontal = pd.concat([s1, s2], axis=1) # axis=1 이면 index (행) 기준으로 합함. 즉. 좌-우로 합함
print(horizontal)

print('\n')
vertical = pd.concat([s1, s2], axis=0) # axis=0 이면 column(열) 기준으로 합함. 즉 위-아래로 합함
print(vertical)

print('\n')
vertical_1 = pd.concat([s1, s2], axis=1, join='outer') # axis=1 인덱스 기준으로 합함. 양쪽 Series 에 존재하는 모든 index 는 남김
vertical_2 = pd.concat([s1, s2], axis=1, join='inner') # axis=1 인덱스 기준으로 합함. 양쪽 Series 에 동시에 존재하는 index 만 남김
print(vertical_1.equals(vertical_2)) # 두개의 DataFrame 이 서로 동일한지 체크. 인덱스가 동일하므로 동일 결과가 됨.
```

```
   s1 s2
0   1  a
1   2  b
2   3  c
3   4  d
4   5  e


0    1
1    2
2    3
3    4
4    5
0    a
1    b
2    c
3    d
4    e
dtype: object


True
```

concat 에서 두 Series 의 index 가 다르경우, 원하는 결과가 안 나온다는 것의 유의합니다. 아래 예제에서 index 가 서로 다른 Series 를 합쳐보겠습니다. join='inner' 조건에서는 동일한 index 가 없으므로 concat 후 결과가 없습니다. 단지 좌-우로 합치는 것이 목적이라면 기존의 index 를 제거하고 default index 인 숫자를 넣어주고 concat 하면 됩니다. 기존의 index 를 제거할 때는 reset_index(drop=True) 를 합니다.

```
s3 = pd.Series([1,2,3,4,5], index = ['a','b','c','d','e'] , name='s3')
s4 = pd.Series([11,12,13,14,15], index = ['f','g','h','i','j'], name='s4')

print(pd.concat([s3, s4], axis=1, join='inner')) # axis=1 이면 인덱스 기준으로 합함.
즉. 좌-우로 합함

print('\n')
print(pd.concat([s3.reset_index(drop=True), s4.reset_index(drop=True)], axis=1,
join='inner')) # axis=1 이면 인덱스 기준으로 합함. 즉. 좌-우로 합함
```

```
Empty DataFrame
Columns: [s3, s4]
Index: []


   s3  s4
0   1  11
1   2  12
2   3  13
3   4  14
4   5  15
```

Index 가 동일하고, 단순한 병합일 때는 concat 를 쓰지만, 서로 다른 컬럼으로 병합을 할 때는 Merge 를 씁니다. 만약 두 데이터셋이 있고, 고객번호로 서로 Merge 하려고 한다고 합시다. 그런데 한 데이터셋에는 고객번호가 cust_id 로 되어 있고, 다른 데이터셋에는 Cust_Number 로 되어 있으면 concat 를 활용하기 어렵습니다. 이 경우는 merge 를 쓰는 것이 편리합니다. merge 는 넣어야하는 인수가 concat 보다많아, 단순한 병합은 concat 으로 합니다. 먼저 예제 DataFrame 을 생성합니다.

```
cust_list = [10, 11, 12, 13, 14, 15]
product_list = ['a','b','c','d','e', 'f']
df1 = pd.DataFrame({'cust_id': cust_list, 'product': product_list})

cust_list = [12, 13, 14, 15, 16, 17]
grade_list = ['p1','p2','p3','p4','p5','p6']
df2 = pd.DataFrame({'cust_number': cust_list, 'grade': grade_list})

print(df1)
print('\n')
print(df2)
```

```
   cust_id product
0       10       a
1       11       b
2       12       c
3       13       d
4       14       e
5       15       f


   cust_number grade
0           12    p1
1           13    p2
2           14    p3
3           15    p4
4           16    p5
5           17    p6
```

Merge 로 데이터셋을 병합하는 방법에는 여러가지가 있습니다. 예제에서는 index 를 기준으로 합치는 방법을 해 보겠습니다. 일단, df1 과 df2 에서 키가 되는 고객번호가 존재합니다. 이 고객번호를 index 로 만드는 법은 아래와 같습니다.

```
print(df1.set_index('cust_id'))
print(df2.set_index('cust_number'))
```

```
         product
cust_id
10             a
11             b
12             c
13             d
14             e
15             f
             grade
cust_number
12              p1
13              p2
14              p3
15              p4
16              p5
17              p6
```

다음은 만들어진 index 를 이용하여 두 데이터셋을 병합(merge) 합니다. left_index=True, right_index=True 를 인수로 넣어, index 키로 병합한다는 것을 알려줍니다. 병합하는 방법은 how 인수로 알려줍니다. how='inner' 면 df1, df2 동시에 존재하는 index 만을 남기겠다는 인수입니다. 아래 예제에서 두 번째 방식으로도 가능하나, 제 생각에는 첫 번째가 직관적입니다.

```
df1.set_index('cust_id').merge(df2.set_index('cust_number'), left_index=True,
right_index=True, how='inner')
```

	product	grade
12	c	p1
13	d	p2
14	e	p3
15	f	p4

```
pd.merge(left=df1.set_index('cust_id'), right=df2.set_index('cust_number'),
left_index=True, right_index=True, how='inner')
```

	product	grade
12	c	p1
13	d	p2
14	e	p3
15	f	p4

Index 가 된 고객번호를 다시 DataFrame 으로 가져오고 싶으면, reset_index() 로 index 를 없앤 후, rename 메소드에서 원하는 이름으로 변경해주면 됩니다. 아래 예제와 같이 파이썬에서는 여러가지 데이터처리를 '.' (dot notation) 을 이용하여 한 줄에 처리할 수 있습니다.

```
df1.set_index('cust_id').merge(df2.set_index('cust_number'), left_index=True,
right_index=True, how='inner').reset_index().rename(columns={'index':'cust_id'})
```

	cust_id	product	grade
0	12	c	p1
1	13	d	p2
2	14	e	p3
3	15	f	p4