دوره جامع پایتون در یادگیری ماشین

قسمت هشتم، ادغام چند جدول با همدیگر

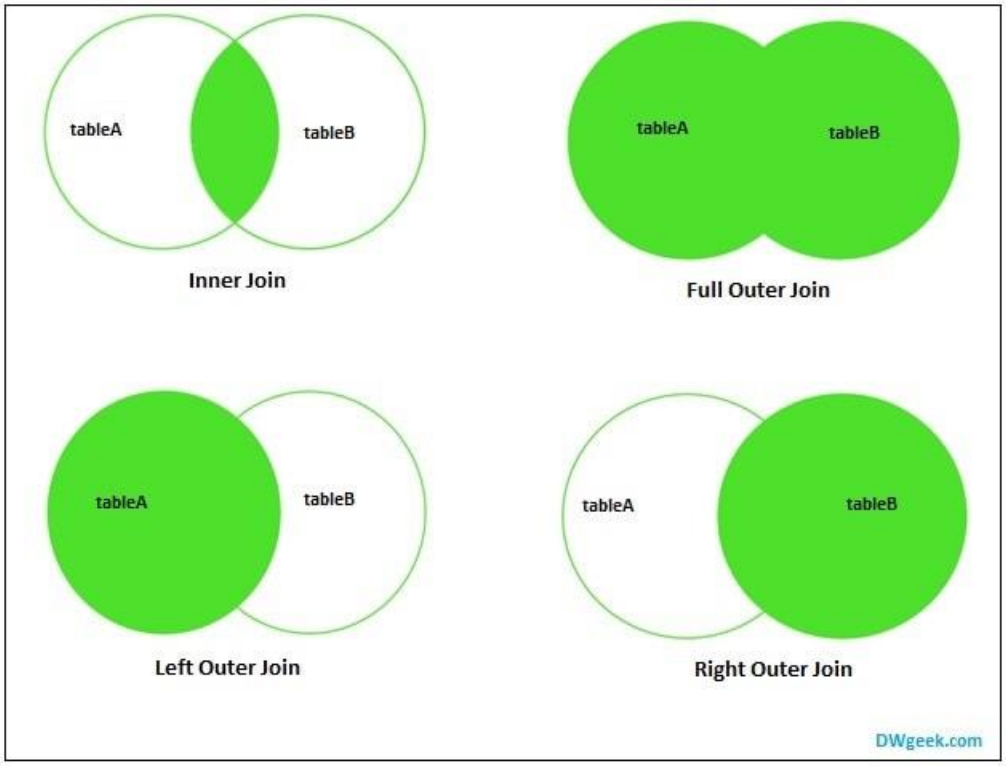
Pandas[5]

Datalk.ir
Created by : Ali Arabshahi
Contact us : Linkedin.com/in/mrAliArabshahi

ادغام چند جدول با همدیگر

Merging, Joining, and Concatenating

زمانی که چند تا **جدول مختلف** داریم و به هر دلیلی می خوایم اون ها رو با هم **ادغام** و در قالب **یک جدول** داشته باشیم، هر کدوم از این سه تا دستور یعنی **مِرج**، **جوین** و یا **کانکَتنِیت** می تونن به کارمون می یاد .فقط در نظر داشته باشین که درک مفهوم ادغام کردن یه مقداری زمان بر هست و معمولا دوستانی که تجربه بیشتری در کارکردن با دیتابیس های مختلف از جمله اس کیو ال دارن، خیلی راحت تر متوجه این موارد می شن .بهترین راهش دیدن مثال های زیاده که ما سعی کردیم در این جلسه این کار رو انجام بدیم اما بازم پیشنهاد می کنم که در آخر این جلسه برین و یک سرچی در یوتیوب راجع به این کلیدواژه ها انجام بدین تا مطلب بهتر براتون جا بیوفته .پیشاپیش از این که احیانا اگر یک کوچولو متوجه نشدین، عذر خواهی می کنم 😬 ولی بدونین واقعا اون قدر ها هم مطلب پیچیده ای نیست و فقط نیاز داره در عمل بیشتر با این چیزا سر و کار داشته باشین. بیاین شروع کنیم. 🤠



ایجاد دیتافریم

قبل از هر چیز سه تا دیتافرم ایجاد می کنیم:

```
import pandas as pd
```

```
df1 = pd.DataFrame({'A': ['A0', 'A1', 'A2', 'A3'],
                    'B': ['B0', 'B1', 'B2', 'B3'],
                    'C': ['C0', 'C1', 'C2', 'C3'],
                    'D': ['D0', 'D1', 'D2', 'D3']},
                   index=[0, 1, 2, 3])


df2 = pd.DataFrame({'A': ['A4', 'A5', 'A6', 'A7'],
                    'B': ['B4', 'B5', 'B6', 'B7'],
                    'C': ['C4', 'C5', 'C6', 'C7'],
                    'D': ['D4', 'D5', 'D6', 'D7']},
                     index=[4, 5, 6, 7])


df3 = pd.DataFrame({'A': ['A8', 'A9', 'A10', 'A11'],
                    'B': ['B8', 'B9', 'B10', 'B11'],
                    'C': ['C8', 'C9', 'C10', 'C11'],
                    'D': ['D8', 'D9', 'D10', 'D11']},
                   index=[8, 9, 10, 11])
```

```
df1
```

	A	B	C	D
0	A0	B0	C0	D0
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	A3	B3	C3	D3

df2

	A	B	C	D
4	A4	B4	C4	D4
5	A5	B5	C5	D5
6	A6	B6	C6	D6
7	A7	B7	C7	D7

df3

	A	B	C	D
8	A8	B8	C8	D8
9	A9	B9	C9	D9
10	A10	B10	C10	D10
11	A11	B11	C11	D11

کانکَتِنِیت (Concatenation)

کاری که این دستور انجام می ده اینه که می یاد و بر اساس سطر یا ستون، جداول رو با هم ادغام می کنه .این رو در نظر داشته باشین که **کانکَتِنِیت** معمولا زمانی کاربرد داره که جداول مون در امتداد و کامل کننده هم دیگه هستند سه تا دیتافریمی که در بالا ایجاد کردیم به این صورت با هم دیگه کانکَتِنِیت و یا به نوعی ادغام می شن🙂

```
pd.concat([df1,df2,df3])
```

	A	B	C	D
0	A0	B0	C0	D0
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	A3	B3	C3	D3
4	A4	B4	C4	D4
5	A5	B5	C5	D5
6	A6	B6	C6	D6
7	A7	B7	C7	D7
8	A8	B8	C8	D8
9	A9	B9	C9	D9
10	A10	B10	C10	D10
11	A11	B11	C11	D11

حواستون به این موضوع باشه که در حالت پیشفرض کانکتِنیت بر روی ردیف ها اعمال می شه یعنی اکسیس برابر صفر هست .می تونیم این ادغام رو توسط دستور زیر به صورت ستونی انجام بدیم .دلیل ایجاد داده های نال اینه که مثلا جدول شماره و سه که با هم ادغام شده اند، سطر های شماره صفر و یک و دو و سه و چهار که نداشتند !پس خروجی نال رو تحویل ما می ده. 🧐

```
pd.concat([df1,df2,df3],axis=1)
```

	A	B	C	D	A	B	C	D	A	B	C	D
0	A0	B0	C0	D0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	A1	B1	C1	D1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	A2	B2	C2	D2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	A3	B3	C3	D3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	NaN	NaN	NaN	NaN	A4	B4	C4	D4	NaN	NaN	NaN	NaN
5	NaN	NaN	NaN	NaN	A5	B5	C5	D5	NaN	NaN	NaN	NaN
6	NaN	NaN	NaN	NaN	A6	B6	C6	D6	NaN	NaN	NaN	NaN
7	NaN	NaN	NaN	NaN	A7	B7	C7	D7	NaN	NaN	NaN	NaN
8	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	A8	B8	C8	D8
9	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	A9	B9	C9	D9
10	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	A10	B10	C10	D10
11	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	A11	B11	C11	D11

ایجاد دیتا فریم

بیاین برای بررسی حالت های دیگه ادغام کردن، دو تا دیتا فریم زیر با نام های رایت و لفت رو تعریف کنیم.

```
left = pd.DataFrame({'key': ['K0', 'K1', 'K2', 'K3'],
                     'A': ['A0', 'A1', 'A2', 'A3'],
                     'B': ['B0', 'B1', 'B2', 'B3']})


right = pd.DataFrame({'key': ['K0', 'K1', 'K2', 'K3','K4'],
                      'C': ['C0', 'C1', 'C2', 'C3','C4'],
                      'D': ['D0', 'D1', 'D2', 'D3','C4']})
```

```
left
```

	key	A	B
0	K0	A0	B0
1	K1	A1	B1
2	K2	A2	B2
3	K3	A3	B3

right

	key	C	D
0	K0	C0	D0
1	K1	C1	D1
2	K2	C2	D2
3	K3	C3	D3
4	K4	C4	C4

مِرج کردن (Merging)

تابع **مِرج** به ما اجازه می ده که بر اساس یک ستون یا چند ستون مشترک بین جداولمون، این ادغام کردن رو انجام بدیم.

on مشخص کننده ستون هایی هست که می خوایم بر روی اون مِرج انجام بشه و how، بیانگر نوع مِرجی هست که قراره صورت بگیره و منظور از **'inner'**، این هست که بیا و مقادیر مشترک در ستون های تعریف شده رو بگیر و این ادغام سازی رو برای ما انجام بده .در این حالت داده ی نالی دریافت نمی کنیم چون بر اساس مقادیر مشترک عمل ادغام سازی انجام می شه.

```
pd.merge(left,right,how='inner',on='key')
```

	key	A	B	C	D
0	K0	A0	B0	C0	D0
1	K1	A1	B1	C1	D1
2	K2	A2	B2	C2	D2
3	K3	A3	B3	C3	D3

بیاین یه مسئله پیچیده تر رو بررسی کنیم 😎

```
left = pd.DataFrame({'key1': ['K0', 'K0', 'K1', 'K2'],
                     'key2': ['K0', 'K1', 'K0', 'K1'],
                        'A': ['A0', 'A1', 'A2', 'A3'],
                        'B': ['B0', 'B1', 'B2', 'B3']})
```

```
right = pd.DataFrame({'key1': ['K0', 'K1', 'K1', 'K2'],
                      'key2': ['K0', 'K0', 'K0', 'K0'],
                         'C': ['C0', 'C1', 'C2', 'C3'],
                         'D': ['D0', 'D1', 'D2', 'D3']})
```

در مثال زیر هم به جای یک ستون از دو ستون مشترک برای مرج کردن استفاده می کنیم .دقت کنین که در حالت پیشفرض گزینه how برابر با **'inner'** هست:

```
pd.merge(left, right, on=['key1', 'key2'])
```

	key1	key2	A	B	C	D
0	K0	K0	A0	B0	C0	D0
1	K1	K0	A2	B2	C1	D1
2	K1	K0	A2	B2	C2	D2

در این مثال هم نوع مرج رو به **'outer'** تغییر دادیم .در این حالت ادغام بر اساس همه ستون هایی که تعریف کردیم ،صورت می گیره و این یعنی ممکنه مقادیر نال هم در جاهایی که مقدار بین ستون ها در جداول مختلف یکسان نباشه مشاهده بشه:

```
pd.merge(left, right, how='outer', on=['key1', 'key2'])
```

	key1	key2	A	B	C	D
0	K0	K0	A0	B0	C0	D0
1	K0	K1	A1	B1	NaN	NaN
2	K1	K0	A2	B2	C1	D1
3	K1	K0	A2	B2	C2	D2
4	K2	K1	A3	B3	NaN	NaN
5	K2	K0	NaN	NaN	C3	D3

در این مثال هم ادغاممون از نوع **'right'** هست .تو این حالت می یاد و ستون های تعریف شده از جدول دوم رو مبنا قرار میده .یعنی اون ها رو فراخونی می کنه و بر اساسشون مقادیر رو در جاهایی که موجود هست فراخونی و در غیر

این صورت ، نال رو بر می گردونه .این رو بدونین که پانداس به طور پیشفرض، جدول اولی رو چپ و جدول دومی رو راست تعریف می کنه پس در اینجا منظور از راست جدول دومی هست .حالا هر اسمی که می خواد داشته باشه:

```
pd.merge(left, right, how='right', on=['key1', 'key2']).
```

	key1	key2	A	B	C	D
0	K0	K0	A0	B0	C0	D0
1	K1	K0	A2	B2	C1	D1
2	K1	K0	A2	B2	C2	D2
3	K2	K0	NaN	NaN	C3	D3

و در مثال آخرمون هم ادغام از نوع **'left'** هست که می یاد و ستون های تعریف شده از جدول اول یا جدول چپی رو بر اساس کلید های تعریف شده مبنا قرار میده

```
pd.merge(left, right, how='left', on=['key1', 'key2'])
```

	key1	key2	A	B	C	D
0	K0	K0	A0	B0	C0	D0
1	K0	K1	A1	B1	NaN	NaN
2	K1	K0	A2	B2	C1	D1
3	K1	K0	A2	B2	C2	D2
4	K2	K1	A3	B3	NaN	NaN

جوین کردن (Joining)

جوین کردن دقیقا مشابه مرج کردنه، با این تفاوت که در اینجا ادغام بر اساس ایندکس (نام ردیف) های مشابه صورت می گیره و نه ستون های تعریف شده.

برای این که دستمون گرم بشه، دو تا دیتافریم زیر رو تعریف می کنیم.🙃

```
left = pd.DataFrame({'A': ['A0', 'A1', 'A2'],
                     'B': ['B0', 'B1', 'B2']},
                      index=['K0', 'K1', 'K2'])
```

```
right = pd.DataFrame({'C': ['C0', 'C2', 'C3'],
                      'D': ['D0', 'D2', 'D3']},
                      index=['K0', 'K2', 'K3'])
```

و حالا می آییم و روی جدول با اسم چپ، جوین می زنیم جدول با اسم راست رو و این جوینمون از چه نوعی هست؟

از نوع 'left'.

یعنی جدول چپی یا اولی، ایندکس هاش حفظ می شه و بر اساس اون ایندکس ها داده های دیگه فراخونی می شوند و در جاهایی هم که داده ای با این ایندکس ها در جداول دیگه وجود نداشته باشند، مقدار نال برگردونده می شه و دقیقا مشابه مرج کردن، حالت های دیگر جوین رو در مثال های زیر مشاهده می کنیم:

```
left.join(right, how='left')
```

	A	B	C	D
K0	A0	B0	C0	D0
K1	A1	B1	NaN	NaN
K2	A2	B2	C2	D2

```
left.join(right, how='right')
```

	A	B	C	D
K0	A0	B0	C0	D0
K2	A2	B2	C2	D2
K3	NaN	NaN	C3	D3

```
left.join(right, how='outer')
```

	A	B	C	D
K0	A0	B0	C0	D0
K1	A1	B1	NaN	NaN
K2	A2	B2	C2	D2
K3	NaN	NaN	C3	D3


```
left.join(right, how='inner')
```

	A	B	C	D
K0	A0	B0	C0	D0
K2	A2	B2	C2	D2

خیلی خسته نباشین