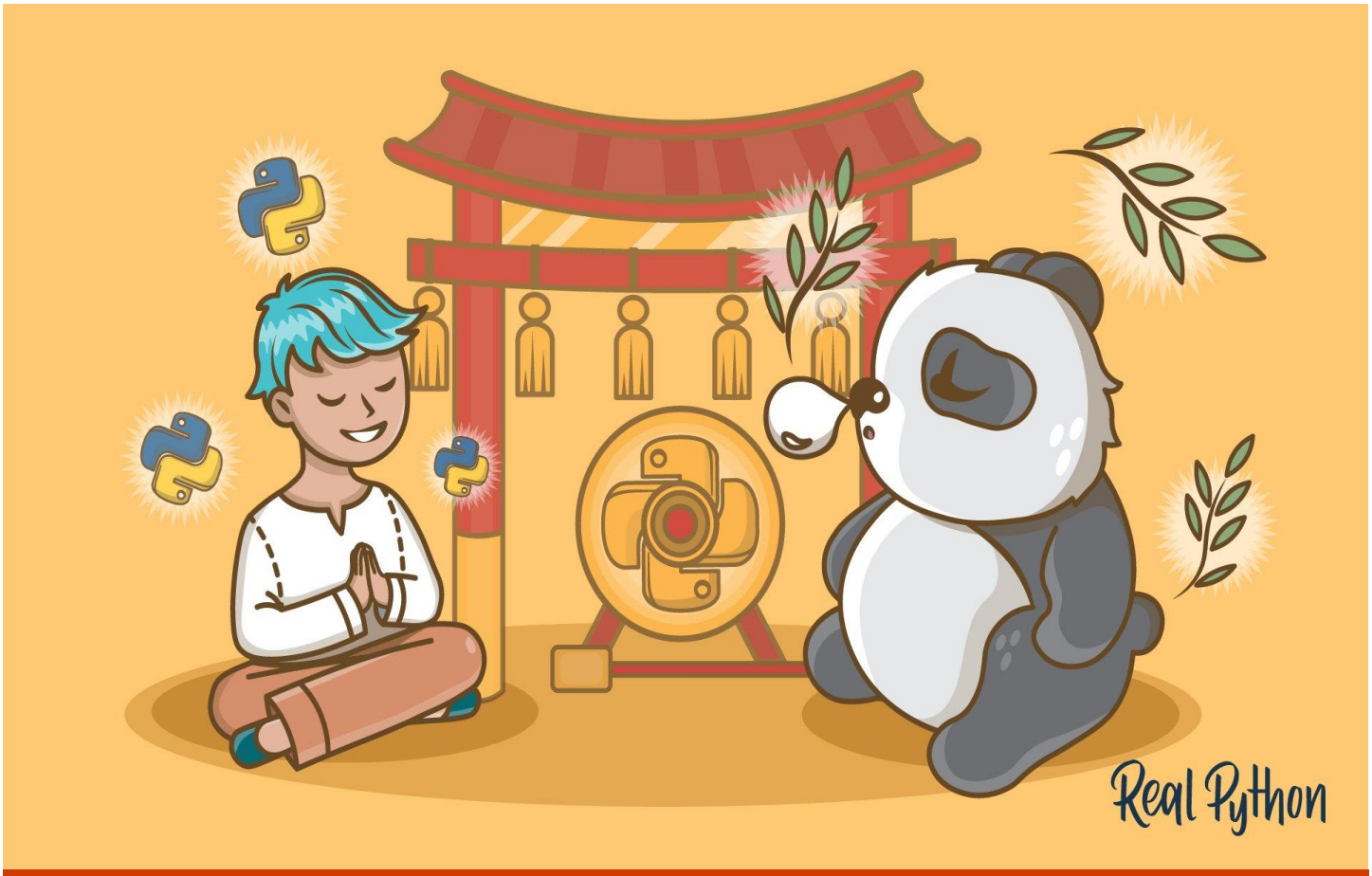


دوره جامع پایتون در یادگیری ماشین

قسمت چهارم، مقدمه و تعریف سری ها

Pandas[1]

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

و اما پانداس

یکی از بهترین و پرکاربرد ترین کتابخونه های پایتون برای اهالی علم داده **پانداس** محسوب میشه که کار اصلی اون هم **پاکسازی** و هم **پیش پردازش** داده هاست.یعنی زمانی که بخوایم روی کل داده هامون علمیات به خصوصی رو انجام بدیم.خب کتابخونه نامپای ام که یه همچین کارایی رو می کرد! پس فرقشون چیه؟! 🤷

کتابخونه نامپای خروجیش آرایه هست اما در پانداس با **سری ها** و **دیتا فریم ها** سر و کار داریم .نامپای فقط با اعداد سر و کار داره اما پانداس با متغیر هایی از جنس های مختلف. بزارین اینطوری بگیم که نامپای یه جورایی ابزار دست علم جبر خطی و فضا های ماتریسی و امثال اینا هست اما پانداس بیشتر مشابه ابزاری مثل اکسل عمل میکنه؛ این رو هم بدونین بعضی کتابخونه ها هستن مثل تنسورفلو که ورودیش حتما باید از جنس آرایه ها باشن .هم نامپای و هم پانداس از ابزار های پایه ایی علم داده ها محسوب میشن و نمیشه یکیشونو سر به نیست کرد. 😶

برای نصب این کتابخونه با توجه به محیطی که ازش استفاده می کنین یکی از دستورات زیر رو وارد کنین ، اگر نشد از گوگل کمک بگیرین دیگه! 😐

```
# pip install pandas
# conda install pandas
```

سری ها

اولین نوع از داده هایی که باهاشون آشنا می شیم **سری ها** هستن اما قبل از هر چیز بیاین اون رو به همراه کتابخونه نامپای فراخونی کنیم.

```
import numpy as np
import pandas as pd
```

سری ها خیلی مشابه آرایه های یک بعدی نامپای هستن اما یکی از فرق های خیلی مهم بین اونا اینه که سری ها قابلیت گرفتن لیبل رو به خودشون دارن .در آرایه ها این لیبل به طور پیشفرض وجود داشت اما از صفر شروع می شد. همچنین همون طور که اشاره کردیم مقادیر در پانداس می تونن هر چیزی باشن اما در نامپای فقط با عدد سر و کار داریم .بیاین این مسئله رو با مثال بیشتر باز کنیم.

ساختن یک سری

.برای این کار میتونیم یک لیست و یا یک آرایه نامپای و حتی یک دیکشنری رو به سری تبدیل کنیم پس بیاین قبل از هر چیز هر سه نوع این انواع رو، یعنی یک لیست، دیکشنری و آرایه، تعریف کنیم. 😎

```
labels = ['a','b','c']
my_list = [10,20,30]
arr = np.array([10,20,30])
d = {'a':10,'b':20,'c':30}
```


تبدیل لیست به سری

```
pd.Series(data=my_list)
```

```
0    10
1    20
2    30
dtype: int64
```

در سری ها به طور پیشفرض ایندکس ها یا لیبل ها یا نام ردیف ها از صفر شروع میشن(مشابه آرایه ها در نامپای) اما می تونیم این لیبل ها رو مانند پایین در مقادیر دلخواهی تعریف کنیم :

```
pd.Series(data=my_list,index=labels)
```

```
a    10
b    20
c    30
dtype: int64
```

کاری که در بالا انجام دادیم رو میتونیم به شرح زیر هم انجام بدیم، اولین مقدار همیشه معنی دیتای ما و دومین مقدار معنی لیبل ها رو میده و پانداس این رو می دونه.

```
pd.Series(my_list,labels)
```

```
a    10
b    20
c    30
dtype: int64
```

تبدیل آرایه ها به سری

```
pd.Series(arr)
```

```
0    10
1    20
2    30
dtype: int32
```

```
pd.Series(arr,labels)
```

```
a    10
b    20
c    30
dtype: int32
```


بنابرین پانداس آرایه رو می فهمه! و این یعنی کتابخونه های نامپای و پانداس رابطه خیلی خوبی با هم دارند و جدا ی از هم دیگه نیستن.😍

تبدیل دیکشنری ها به سری

اگه یادتون باشه دیکشنری ها شامل دو جز بودن، ولیو ها (values) و کیی ها (keys) یا مقادیر و کلید ها !پانداس همیشه کلید ها رو به عنوان لیبل و مقادیر رو به عنوان دیتای ما در نظر می گیره. **پس هر دیکشنری قابلیت تبدیل شدن به یک دیتافریم رو داره**😎

```
pd.Series(d)
```

```
a    10
b    20
c    30
dtype: int64
```

حتما یادتون هست که گفتیم یکی از ویژگی های باحال نامپای اینه که میتونه مقادیر غیر عددی رو هم در خودش جای بده .همون طور که مشاهده می کنین، این بار لیبل هامون رو به عنوان دیتاهامون تعریف می کنیم.

```
pd.Series(data=labels)
```

```
0    a
1    b
2    c
dtype: object
```

استفاده کردن از ایندکس ها

برای این که سری ها رو بهتر درک کنیم، خیلی خوبه که متوجه مفهوم و کاربرد ایندکس در اون ها بشیم .دو تا سری زیر رو تعریف می کنیم و بعد هر دوشون رو با هم جمع می کنیم .پانداس بر اساس ایندکس های مشابه ای که دارن این جمع رو انجام می ده و در مواقعی که ایندکس ها متفاوت هستن نال یا بی معنی رو بهمون تحویل میده:

```
ser1 = pd.Series([1,2,3,4],index = ['USA', 'Germany','USSR', 'Japan'])
ser1
```

```
USA        1
Germany    2
USSR       3
Japan      4
dtype: int64
```

```
ser2 = pd.Series([1,2,5,4],index = ['USA', 'Germany','Italy', 'Japan'])
ser2
```

```
USA        1
Germany    2
```

```
Italy      5
Japan      4
dtype: int64
```

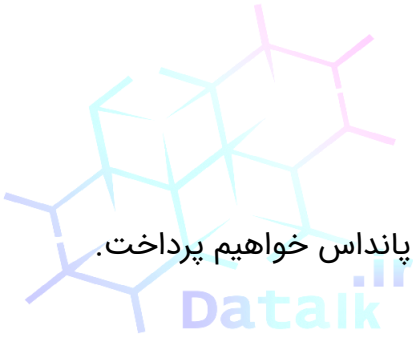
```
ser1['USA']
```

```
1
```

همان طور که مشاهده می کنیم ، عملیات ریاضی بر اساس ایندکس ها صورت می گیره:

```
ser1 + ser2
```

```
Germany    4.0
Italy      NaN
Japan      8.0
USA        2.0
USSR       NaN
dtype: float64
```



در جلسه بعدی به مفهوم دیتافریم ها در پانداس خواهیم پرداخت.

تا اون موقع، خدانگهدار