کتابخانه Pandasدر پایتون

نوشته شده توسط معین حاج ملک

پانداس (Pandas) یک کتابخانه قدرتمند برای تحلیل پیش پردازش و بصری سازی داده ها است.

دانشمندان داده زمان زیادی را صرف پاک سازی و دیگر پردازش های داده ها می‌کنند تا داده‌ها را برای انجام تحلیل‌های گوناگون قابل استفاده کنند. یکی از کتابخانه‌های اصلی پایتون برای آماده‌سازی و پیش‌پردازش داده‌ها، پانداس (Pandas) است. بنابراین در این مطلب، به طور خلاصه چگونگی پیش‌پردازش داده‌ها با بهره‌گیری از پانداس نیز مورد بررسی قرار گرفته است. در نهایت، برخی از توابع و متدهای پایه‌ای برای کار روی دیتافریم‌ها نیز مورد بررسی قرار گرفته‌اند که از این جمله می‌توان به ()value_counts() ، ()groupby، ()merge، describe()، concatو ()count اشاره کرد. این متدها و توابع، به دسته‌بندی و اکتشاف دیتافریم‌ها در حین فرایند تحلیل کمک شایان توجهی می‌کنند.

ساختار داده ها در پانداس

پانداس (Pandas) دارای دو ساختار اصلی برای ذخیره‌سازی داده‌ها است که

عبارتند از:

- Series
- دیتافریم(DataFrame)

Series در پانداس

یک series مشابه با آرایه یک‌بُعدی است. seriesمی‌تواند داده‌ها از هر نوعی را

ذخیره کند. مقادیری که در series قرار می‌گیرند قابل تغییر هستند؛ اما

اندازه series پانداس، غیر قابل تغییر است. به اولین عنصر درseries ، اندیس ۰

تخصیص داده خواهد شد و اندیس آخرین عنصر در series برابر با N-1 است که در

آن، Nتعداد کل عنصرهای موجود در سری است. برای ساخت Series پانداس، ابتدا

باید بسته پانداس را با استفاده از دستور import پایتون، وارد (import) کرد.

```
import pandas as pd
```

برای ساختSeries ، متد pd.Series() فراخوانی می‌شود و یک آرایه، چنانکه در

زیر نمایش داده شده، پاس داده می‌شود.

```
series1 = pd.Series([1,2,3,4])
```

سپس، دستور print برای نمایش دادن محتوای Series مورد استفاده قرار می‌گیرد.

```
print(series1)
```

خروجی:

```
0 1
1 2
2 3
3 4
dtype: int64
```

می‌توان مشاهده کرد که دو ستون وجود دارد، ستون اول حاوی مقادیری است که از

اندیس صفر شروع می‌شوند و ستون دوم حاوی مقادیری است که به series اضافه

شده‌اند. ستون اول اندیس عناصر را نشان می‌دهد. کاربر ممکن است هنگام نمایش series با خطا مواجه شود. دلیل اصلی این خطا آن است که پانداس به دنبال اطلاعاتی می‌گردد که نمایش دهد. بنابراین، کاربر باید اطلاعات خروجی سیستم را فراهم کند. این خطا را می‌توان با اجرای کد به صورت زیر حل کرد.

```
import pandas as pd
import sys

sys.__stdout__ = sys.stdout

series1 = pd.Series([1,2,3,4])
print(series1)
```

یک Series ممکن است از آرایه numpy ساخته شود. در ادامه یک آرایه numpy ساخته می‌شود و سپس، این آرایه به Series پانداس تبدیل (Convert)می‌شود.

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


sys.__stdout__ = sys.stdout


fruits = np.array(['apple','orange','mango','pear'])
series2 = pd.Series(fruits)
print(series2)
```

خروجی:

```
0 apple
1 orange
2 mango
3 pear
dtype: object
```

کار با وارد کردن کتابخانه‌های لازم، از جمله numpy آغاز شده است. سپس،

تابع array() فراخوانی شده تا یک آرایه از میوه‌ها ساخته شود. پس از آن، از

تابع Series() پانداس استفاده شده و آرایه‌ای که کاربر تمایل دارد آن را به

یک series تبدیل کند به آن پاس داده می‌شود. در نهایت، تابع print() برای نمایش Series مورد استفاده قرار می‌گیرد.

دیتافریم

ساختار داده دیتافریم (DataFrame) در پانداس را می‌توان به عنوان یک جدول در نظر گرفت. دیتافریم، داده‌ها را در سطرها و ستون‌ها سازمان‌دهی می‌کند و از آن‌ها یک ساختار داده دوبُعدی می‌سازد. ستون‌ها می‌توانند حاوی مقادیری از انواع گوناگون باشند و در عین حال، اندازه دیتافریم قابل تغییر است؛ بنابراین می‌توان آن را ویرایش کرد.

برای ساخت دیتافریم، می‌توان کار را از پایه شروع کرد و یا ساختار داده‌هایی مانند آرایه‌های نام‌پای (Numpy) را به یک دیتافریم مبدل ساخت. در ادامه، کد مربوط به چگونگی ساخت یک DataFrame از پایه، آورده شده است.

```
import pandas as pd
```

```python
df = pd.DataFrame({
    "Column1": [1, 4, 8, 7, 9],
    "Column2": ['a', 'column', 'with', 'a', 'string'],
    "Column3": [1.23, 23.5, 45.6, 32.1234, 89.453],
    "Column4": [True, False, True, False, True]
})
print(df)
```

خروجی:

```
 Column1 Column2 Column3 Column4
0 1 a 1.2300 True
1 4 column 23.5000 False
2 8 with 45.6000 True
3 7 a 32.1234 False
4 9 string 89.4530 True
```

در مثال بالا، یک دیتافریم با نام df ساخته شده است. ستون اول دیتافریم (DataFrame)حاوی مقادیر صحیح، دومین ستون حاوی یک رشته، ستون سوم حاوی مقادیر «ممیز شناور (Floating Point) «و ستون چهارم حاوی مقادیر «بولی (Boolean) «است. دستور print(df) محتوای دیتافریم را با استفاده از کنسول به کاربر نمایش می‌دهد و این امکان را فراهم می‌کند تا کاربر این محتوا را

بررسی و تایید کند. اگرچه، هنگام نمایش دیتافریم، ممکن است کاربر متوجه شود که یک ستون اضافی در آغاز جدول وجود دارد که عناصر آن از ۰ شروع می‌شوند (اندیس‌ها). برای ساخت دیتافریم، باید متد pd.DataFrame() به صورتی که در مثال بالا نمایش داده شده فراخوانی شود. ساخت یک DataFrame از لیست یا یک مجموعه از لیست‌ها، کاری دشوار است. اکنون فقط باید متد pd.DataFrame() فراخوانی شود و سپس، متغیر لیست به عنوان تنها آرگومان به آن پاس داده می‌شود.

مثال زیر در همین راستا قابل توجه است.

```
import pandas as pd
mylist = [4, 8, 12, 16, 20]
df = pd.DataFrame(mylist)
print(df)
```

خروجی:

```
 0
0 4
```

```
1 8
2 12
3 16
4 20
```

در این مثال، لیستی با عنوان mylist با توالی از پنج عدد صحیح ساخته شده است.

سپس، متد DataFrame() فراخوانی شده و نام لیست به عنوان آرگومان به آن

پاس داده شده است. این همان جایی است که تبدیل لیست به دیتافریم اتفاق می‌افتد.

سپس، محتوای دیتافریم پرینت می‌شود. دیتافریم دارای یک ستون پیش‌فرض است که

اندیس‌ها را نمایش می‌دهد و در آن، اندیس اول از صفر شروع می‌شود و اندیس آخر

N-1است و در آن، Nتعداد کل عناصر موجود در دیتافریم به حساب می‌آید. مثال

دیگری از این مورد، در ادامه آورده شده است.

```
import pandas as pd

items = [['Phone', 2000], ['TV', 1500], ['Radio', 800]]

df = pd.DataFrame(items, columns=['Item', 'Price'],
dtype=float)

print(df)
```

خروجی:

```
 Item Price
0 Phone 2000.0
1 TV 1500.0
2 Radio 800.0
```

در اینجا، لیستی از عناصر با مجموعه‌ای از ۳ عنصر ساخته شده است. برای هر عنصر، یک نام و قیمت وجود دارد. سپس، لیست به متد DataFrame() پاس داده می‌شود تا آن را به یک شی DataFrame مبدل سازد. در این مثال، نام ستون‌ها برای دیتافریم نیز تعیین شده است. مقادیر عددی نیز به مقادیر ممیز شناور تبدیل می‌شوند زیرا آرگومان dtype از نوع ممیز شناور «float» تعریف شده است. برای دریافت خلاصه داده‌های آیتم‌ها، می‌توان تابع describe() را روی متغیر دیتافریم که df است فراخوانی کرد.

```
df.describe()
```

خروجی:

```
 Price
count 3.000000
mean 1433.333333
std 602.771377
min 800.000000
25% 1150.000000
50% 1500.000000
75% 1750.000000
max 2000.000000
```

تابع describe() جزئیات آماری متداولی مانند میانگین،معیار، عنصر حداقل، عنصر حداکثر و دیگر جزئیات را ارائه می‌کند. این تابع راهکار خوبی برای کسب اطلاعات سریع و کلی پیرامون داده‌هایی محسوب می‌شود که کاربر در حال کار با آن‌ها است؛ به ویژه اگر پیرامون این داده‌ها اطلاعات زیادی نداشته باشد. همچنین، راهکار خوبی برای مقایسه سریع دو مجموعه داده مجزا که حاوی داده‌های مشابهی هستند، محسوب می‌شود.

وارد کردن داده ها

معمولا، نیاز به استفاده از کتابخانه پانداس برای کار با داده‌هایی است که در یک فایل اکسل (Excel) یا CSV ذخیره شده‌اند. این کار نیاز به آن دارد که فایل این داده‌ها باز و داده‌های آن در پانداس وارد (Import) شوند. خوشبختانه، پانداس متدهای زیادی را فراهم می‌کند که می‌توان از آن‌ها برای بارگذاری داده‌ها از چنین منابعی در دیتافریم پانداس استفاده کرد.

وارد کردن داده های CSV

یک فایلCSV (این عبارت، سرنامی برای Comma Separated Value است) یک فایل متنی با مقادیری است که به وسیله کاما (,) از یکدیگر جدا شده‌اند. این نوع فایل بسیار شناخته شده و استانداردی است که اغلب مورد استفاده قرار می‌گیرد. از کتابخانه پانداس می‌توان برای خواندن فایل CSV به صورت کامل یا بخش‌هایی از

آن، استفاده کرد. برای مثال، یک فایل CSV با نام cars.csv ساخته می‌شود. این فایل باید حاوی داده‌های زیر باشد.

```
Number,Type,Capacity
SSD,Premio,1800
KCN,Fielder,1500
USG,Benz,2200
TCH,BMW,2000
KBQ,Range,3500
TBD,Premio,1800
KCP,Benz,2200
USD,Fielder,1500
UGB,BMW,2000
TBG,Range,3200
```

می‌توان داده‌ها را کپی کرد و در ویرایشگر متنی مانند نُت‌پد (Notepad) چسباند؛ سپس، این فایل را با نام cars.csv در همان پوشه‌ای که اسکریپت‌های پایتون ذخیره می‌شوند، ذخیره کرد. پانداس حاوی متدی با عنوان read_csv است که برای خواندن مقادیر CSV در یک دیتافریم پانداس مورد استفاده قرار خواهند گرفت. این متد، مسیر (PATH) به فایل CSV را به صورت آرگومان دریافت می‌کند. کد زیر برای خواند فایل cars.csv مورد استفاده قرار می‌گیرد.

```
import pandas as pd

data = pd.read_csv('cars.csv')

print(data)
```

خروجی:

```
 Number Type Capacity
0 SSD Premio 1800
1 KCN Fielder 1500
2 USG Benz 2200
3 TCH BMW 2000
4 KBQ Range 3500
5 TBD Premio 1800
6 KCP Benz 2200
7 USD Fielder 1500
8 UGB BMW 2000
9 TBG Range 3200
```

در اینجا، فایل CSV در دایرکتوری اسکریپت پایتون ذخیره شده است، بنابراین نام فایل به متد read_csv پاس داده می‌شود و این متد می‌داند که باید پوشه کاری جاری را بررسی کند. اگر فایل در مسیر متفاوتی ذخیره شده است، باید اطمینان حاصل کند که مسیر درستی را به عنوان آرگومان متد پاس داده است. این مسیر می‌تواند مانند cars.csv/.. نسبی و یا مانند Users/nicholas/data/cars.csv/

مطلق باشد. در برخی موارد، ممکن است هزاران سطر در مجموعه داده وجود داشته باشد. در چنین مواردی، بهتر است به جای کل مجموعه داده، تنها چند خط اول در کنسول چاپ شود. این کار می‌تواند با فراخوانی متد head() روی دیتافریم به صورتی که در زیر نشان داده شده، انجام شود.

```
data.head()
```

برای داده‌های بالا، دستور تنها پنج سطر اول مجموعه داده را باز می‌گرداند و این امکان را فراهم می‌کند که کاربر، بخش کوچکی از داده‌ها را مورد بررسی قرار دهد. خروجی کد بالا در زیر نشان داده شده است.

خروجی:

```
 Number Type Capacity
0 SSD Premio 1800
1 KCN Fielder 1500
2 USG Benz 2200
3 TCH BMW 2000
4 KBQ Range 3500
```

متد loc() ابزار مناسبی است که به کاربر کمک می‌کند تا تنها سطرهای معینی را در مجموعه داده بخواند .این مورد، در مثال زیر نمایش داده شده است.

```
import pandas as pd
data = pd.read_csv('cars.csv')


print (data.loc[[0, 4, 7], ['Type']])
```

خروجی:

```
 Type
0 Premio
4 Range
7 Fielder
```

در اینجا، از متد loc() برای خواندن عناصر در اندیس ۰، ۴ و ۷ از ستونType ، استفاده شده است. گاهی ممکن است تنها نیاز به خواندن ستون خاصی باشد و دیگر ستون‌ها خوانده نشوند. این کار با استفاده از متد loc() انجام شده که در مثال زیر نشان داده شده است.

```
import pandas as pd

data = pd.read_csv('cars.csv')


print (data.loc[:, ['Type', 'Capacity']])
```

خروجی:

```
Type Capacity
0 Premio 1800
1 Fielder 1500
2 Benz 2200
3 BMW 2000
4 Range 3500
5 Premio 1800
6 Benz 2200
7 Fielder 1500
8 BMW 2000
9 Range 3200
```

در اینجا از متد loc() برای خواندن همه سطرهای (بخش :) متعلق به تنها دو ستون از مجموعه داده، یعنی ستون‌های Type و Capacity که در آرگومان تعیین شده‌اند، استفاده شده است.

وارد کردن داده های اکسل

علاوه بر متدread_csv ، پانداس از تابع read_excel نیز استفاده می‌کند که می‌تواند برای خواندن داده‌های Excel در یک دیتافریم پانداس استفاده شود. در این مثال، از فایل اکسل با نام workers.xlsx همراه با جزئیات کارگران شرکت استفاده شده است. کد زیر را می‌توان برای بارگذاری محتوای فایل اکسل در یک دیتافریم پانداس استفاده کرد.

```
import pandas as pd

data = pd.read_excel('workers.xlsx')

print (data)
```

خروجی:

```
 ID Name Dept Salary
0 1 John ICT 3000
1 2 Kate Finance 2500
2 3 Joseph HR 3500
3 4 George ICT 2500
4 5 Lucy Legal 3200
5 6 David Library 2000
6 7 James HR 2000
7 8 Alice Security 1500
8 9 Bosco Kitchen 1000
9 10 Mike ICT 3300
```

پس از فراخوانی تابعread_excel ، نام فایل به عنوان آرگومان به آن پاس داده

می‌شود read_excel .برای باز کردن/بارگذاری فایل و سپس، تجزیه داده‌ها مورد

استفاده قرار می‌گیرد. همانطور که از مثال پیشین مشهود است، تابع print() به کاربر کمک می‌کند تا محتوای دیتافریم را نمایش دهد. همچنین، همانطور که در مثال مروبط به کار با فایل CSV بیان شد، این تابع را می‌توان با متد loc() ترکیب کرد تا به خواندن سطرها و ستون‌های خاصی از فایل اکسل کمک کند. برای مثال:

```
import pandas as pd
data = pd.read_excel('workers.xlsx')

print (data.loc[[1,4,7],['Name','Salary']])
```

خروجی:

```
Name Salary
1 Kate 2500
4 Lucy 3200
7 Alice 1500
```

از متد loc() برای بازیابی مقادیر Name و Salary از عناصر در اندیس‌های ۱، ۴ و ۷ استفاده شده است. همچنین، Pandas این امکان را برای کاربر فراهم می‌کند تا

از دو فایل اکسل به طور هم‌زمان بخواند. فرض می‌شود که داده‌های قبلی در Sheet1 هستند و داده‌های دیگری در Sheet2 از همان فایل اکسل قرار دارند. کدی که در ادامه آمده، نشان می‌دهد که چگونه می‌توان از دو شیت به طور هم‌زمان خواند.

```
import pandas as pd
with pd.ExcelFile('workers.xlsx') as x:
    s1 = pd.read_excel(x, 'Sheet1')
    s2 = pd.read_excel(x, 'Sheet2')

print("Sheet 1:")
print (s1)
print("")
print("Sheet 2:")
print (s2)
```

خروجی:

```
Sheet 1:
```

```
   ID    Name      Dept  Salary
0   1    John       ICT    3000
1   2    Kate   Finance    2500
2   3  Joseph        HR    3500
3   4  George       ICT    2500
4   5    Lucy     Legal    3200
5   6   David   Library    2000
6   7   James        HR    2000
7   8   Alice  Security    1500
8   9   Bosco   Kitchen    1000
9  10    Mike       ICT    3300

Sheet 2:
   ID    Name  Age  Retire
0   1    John   55    2023
1   2    Kate   45    2033
2   3  Joseph   55    2023
3   4  George   35    2043
4   5    Lucy   42    2036
5   6   David   50    2028
6   7   James   30    2048
7   8   Alice   24    2054
8   9   Bosco   33    2045
9  10    Mike   35    2043
```

اتفاقی که در کد بالا می‌افتد آن است که تابع read_excel() با کلاس

پوشش‌دهنده ExcelFile ترکیب شده است. متغیر x هنگامی ساخته شده است که

کلاس wrapper با کلیدواژه پایتون with فراخوانی شده است؛ این کلیدواژه برای

باز کردن موقتی فایل مورد استفاده قرار می‌گیرد. از ExcelFile متغیرx ، دو متغیر

دیگر s1 و s2 ساخته شده است تا محتوایی که ازSheet های مختلف خوانده

می‌شوند، با بهره‌گیری از آن‌ها نمایش داده شوند. سپس، از دستور print برای نمایش محتوای دو «کاربرگ (sheet)» در کنسول استفاده شده است. دستور print خالی، یعنیprint("")[1] ، برای چاپ کردن یک خط خالی بین کاربرگ‌ها مورد استفاده قرار می‌گیرد.

درک شکل داده ها با Count و value_counts در پانداس

در صورتی که کاربر با یک دیتافریم بزرگ کار می‌کند، نیاز به استفاده از اکتشافات گوناگون برای درک شکل داده‌ها دارد. در این بخش، دو متد count و value_counts پانداس برای ارزیابی دیتافریم مورد استفاده قرار خواهند گرفت. متد count تعداد مقادیر در هر ستون از DataFrame را نشان می‌دهد. با استفاده از دیتافریم بالا، خروجی زیر حاصل می‌شود.

```
>>> df.count()
date 15
symbol 15
```

```
open 15
high 15
low 15
close 15
volume 15
dtype: int64
```

خروجی برای کاربر مفید نیست، زیرا هر یک از ۱۵ سطر دارای یک مقدار برای هر ستون هستند. اگرچه، این کار در صورتی می‌تواند مفید باشد که مجموعه داده تعداد بیشتری از مقادیر را از دست بدهد. با استفاده از متد count می‌توان به شناسایی ستون‌هایی که غیر کامل هستند کمک کرد. از آنجا، می‌توان تصمیم گرفت که یک ستون دارای مقادیر ناموجود را از پردازش‌ها حذف کرد یا مقادیری برای مقادیر ناموجود یافت و جایگزین کرد.

متد value_counts در پایتون

در این مثال، value_counts مفیدتر است. این متد تعداد مقادیر یکتا را برای یک ستون خاص باز می‌گرداند. اگر کاربر، مقادیر پیوسته دارد (مانند ستون‌های دیتافریم

بالا)، می‌تواند از آرگومان اختیاری bins برای جداسازی مقادیر در bins نیمه باز استفاده کند. اکنون، می‌توان از متد value_counts پانداس برای نمایش شکل ستون volume استفاده کرد.

```
>>> df['volume'].value_counts(bins=4)
(1072952.085, 7683517.5]    10
(20851974.5, 27436203.0]     3
(14267746.0, 20851974.5]     2
(7683517.5, 14267746.0]      0
Name: volume, dtype: int64
```

در خروجی بالا، پانداس (Pandas) چهار دسته جدا برای ستون حجم (volume)ساخته است و تعداد سطرهایی که در هر bin قرار دارند را نشان می‌دهد counts() .و value_counts() ابزارهای مناسبی برای درک سریع شکل داده‌ها هستند.

الحاق

الحاق داده ها که در واقع به معنای افزودن یک مجموعه از داده‌ها به دیگری است، به وسیله فراخوانی تابع concat() قابل انجام است. در ادامه، چگونگی الحاق مجموعه داده‌ها با استفاده از دو دیتافریم پیشین که در بالا معرفی شدند یعنی df1 وdf2 ، هر یک با دو ستون subject_id وstudent_name ، بیان شده است.

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

خروجی:

```
   subject_id student_name
0           1         John
1           2        Emily
2           3         Kate
3           4       Joseph
4           5       Dennis
0           4        Brian
1           5      William
2           6       Lilian
```

```
3           7         Grace
4           8         Caleb
```

آمار توصیفی

چنانکه پیش‌تر نشان داده شد، با استفاده از تابع describe() ، آمار توصیفی برای ستون‌های عددی ارائه می‌شود، اما ستون‌های حاوی کاراکتر توسط این تابع در نظر گرفته نمی‌شوند. در ادامه، ابتدا یک دیتافریم ساخته می‌شود که در آن، اسامی دانش‌آموزان و رتبه آن‌ها در ریاضیات (Math) و انگلیسی (English) نمایش داده شده است.

```
import pandas as pd

data = {
    'Name': ['John', 'Alice', 'Joseph', 'Alex'],
    'English': [64, 78, 68, 58],
    'Maths': [76, 54, 72, 64]
```

```
}


df = pd.DataFrame(data)

print(df)
```

خروجی:

```
 English Maths Name
0 64 76 John
1 78 54 Alice
2 68 72 Joseph
3 58 64 Alex
```

روی دیتافریم و دریافت سنجه‌های describe() اکنون، فقط نیاز به فراخوانی تابع

گوناگون مانند میانگین، انحراف معیار، میانه، عنصر بیشینه، عنصر کمینه و دیگر موارد

.است. کد زیر در این راستا قابل توجه است

```
df.describe()
```

خروجی:

```
 English Maths
```

```
count 4.000000 4.000000
mean 67.000000 66.500000
std 8.406347 9.712535
min 58.000000 54.000000
25% 62.500000 61.500000
50% 66.000000 68.000000
75% 70.500000 73.000000
max 78.000000 76.000000
```

همانطور که مشهود است، متد describe() به طور کامل ستون Name را نادیده گرفت است، زیرا مقادیر آن عددی نیستند. این کار به کاربر کمک می‌کند تا بدون داشتن دغدغه حذف ستون‌های حاوی مقادیر غیر عددی به منظور دریافت آمارهای مربوط به مقادیر عددی، بتواند با داده‌ها کار کند.

نتیجه گیری

پانداس یک کتابخانه بسیار مفید به ویژه برای علم داده است. توابع گوناگون پانداس به ساده‌سازی فرایند پیش‌پردازش داده‌ها کمک قابل توجهی می‌کنند.

منابع

blog.faradars.org