

Kungfu Pandas

Lê Huỳnh Đức

2021-05-22

Contents

1	Cấu trúc và kiểu dữ liệu	5
1.1	Series	5
1.2	DataFrame	7
1.3	Data type trong pandas	7
2	Nhập xuất trong pandas	9
2.1	Đọc và lưu file	9
2.2	Cấu hình pandas	9
3	Một số hàm cơ bản	11
4	Lặp trong Pandas	13
4.1	Sử dụng vectorizer	13
4.2	Sử dụng apply	13
4.3	Sử dụng iterator	13
4.4	Xử lý song song trong pandas	13
5	Select và Filter	15
5.1	Index	15
5.2	loc và iloc	15
5.3	Lọc theo điều kiện	15
6	Các cách kết hợp nhiều bảng với nhau	17
6.1	Join	17
6.2	Merge	17
6.3	Concat	17
7	Groupby và Aggregate	19
8	Làm việc với 1 số kiểu dữ liệu	21
8.1	Xử lý dữ liệu dạng text	21
8.2	Xử lý dữ liệu dạng timestamp	21
8.3	Category trong pandas	21
8.4	Xử lý Missing data	21
9	Một số kiến thức nâng cao	23
9.1	MultiIndex	23
9.2	Pivot và Merge	23
9.3	Resample	23
9.4	Window	23
10	Anomaly Detection Project	25

11 Visualize với Matplotlib 27**Lời nói đầu 29**

Giới thiệu cuốn sách 29

Cài đặt Jupyter Lab 29

Cài đặt Pandas 29

Chapter 1

Cấu trúc và kiểu dữ liệu

1.1 Series

Trong Pandas, **Series** là mảng 1 chiều bao gồm một danh sách giá trị, và một mảng chứa index của các giá trị. Trong dữ liệu dạng bảng, mỗi Series được xem như là một cột của bảng đó. Cách đơn giản để tạo 1 series như sau

```
s = pd.Series(data, index=None, name=None)
```

Trong đó **data** có thể có dạng:

- numpy.ndarray, List
- Python dict
- Scalar

index có thể truyền hoặc không, tùy vào dạng của **data** mà **index** sẽ được định nghĩa khác nhau. **name** là tên của **Series**, giá trị này cũng không nhất thiết phải truyền vào.

Các cách khởi tạo

array

```
pd.Series(data=[0, 1, 2], index=["a", "b", "c"], name="meow")
```

Python dict

```
In [1]: pd.Series({"a":0, "b":1, "c": 2})
Out[1]:
a      0
b      1
c      2
dtype: int64
```

Lưu ý: Trong trường hợp tạo Series với python dict, Series chỉ chứa các giá trị của dict có key nằm trong *index*, với các index không có trong keys của dict, Series sẽ tạo ra các giá trị bị thiếu NaN

Scalar

```
In [1]: pd.Series(data=1, index=["a", "b", "c"])
Out[1]:
a    1
b    1
c    1
dtype: int64
```

Một số thao tác cơ bản

Thao tác trên `Series` cũng giống với thao tác trên `numpy.array`. Ngoài ra chúng ta còn có thể thao tác với `Series` dựa vào index

Ví dụ:

```
In [1]: s = pd.Series(data=[0, 1, 2, 3, 4, 5], index=["a", "b", "c", "d", "e", "f"])
```

Lấy theo indice

```
In [2]: s[2]
Out[2]: 2
```

Lấy theo index

```
In [3]: s["c"]
Out[3]: 2
```

Slice indice

```
In [4]: s[1:3]
Out[4]:
b    1
d    2
dtype: int64
```

Slice index

```
In [5]: s["b":"c"]
Out[5]:
b    1
c    2
dtype: int64
```

List indice

```
In [6]: s[[1,2,4]]
Out[6]:
b    1
c    2
e    4
dtype: int64
```

List index

```
In [7]: s[["b", "c", "e"]]
Out[7]:
b    1
c    2
```

```
e    4
dtype: int64
```

Điều kiện

```
In [5]: s[s > s.mean()]
Out[5]:
d    3
e    4
f    5
dtype: int64
```

1.2 DataFrame

1.3 Data type trong pandas

Các kiểu dữ liệu phổ biến	Numpy/Pandas object	Hiển thị
Boolean	np.bool	<i>bool</i>
Integer	np.int	<i>int</i>
Float	np.float	<i>float</i>
Object	np.object	<i>O, object</i>
Datetime	np.datetime64, pd.Timestamp	<i>datetime64</i>
Timedelta	np.timedelta64, pd.Timedelta	<i>timedelta64</i>
Category	pd.categorical	<i>category</i>

Chapter 2

Nhập xuất trong pandas

2.1 Đọc và lưu file

2.2 Cấu hình pandas

Chapter 3

Một số hàm cơ bản

Chapter 4

Lặp trong Pandas

4.1 Sử dụng vectorizer

4.2 Sử dụng apply

4.3 Sử dụng iterator

4.4 Xử lý song song trong pandas

Chapter 5

Select và Filter

5.1 Index

5.2 loc và iloc

5.3 Lọc theo điều kiện

Chapter 6

Các cách kết hợp nhiều bảng với nhau

6.1 Join

6.2 Merge

6.3 Concat

Chapter 7

Groupby và Aggregate

Chapter 8

Làm việc với 1 số kiểu dữ liệu

8.1 Xử lý dữ liệu dạng text

8.2 Xử lý dữ liệu dạng timestamp

8.3 Category trong pandas

8.4 Xử lý Missing data

Chapter 9

Một số kiến thức nâng cao

9.1 MultiIndex

9.2 Pivot và Merge

9.3 Resample

9.4 Window

Chapter 10

Anomaly Detection Project

Chapter 11

Visualize với Matplotlib

Lời nói đầu

Giới thiệu cuốn sách

Cài đặt Jupyter Lab

Cài đặt Pandas