Pandas – Các Kiến Thức Cốt Lõi Cho Data Science

Mai soạn cho Hùng

2025

Mục lục

1 Tạo Series và DataFrame

Series

Series là một mảng một chiều có nhãn, giống như một cột trong bảng tính. Tham số `index` gán nhãn cho từng phần tử.

```
import pandas as pd
s = pd.Series([1, 2, 3], index=['a', 'b', 'c'])
```

Kết quả: Series với các giá trị 1, 2, 3 tương ứng với nhãn 'a', 'b', 'c'.

DataFrame

DataFrame là bảng 2 chiều với các cột và hàng có nhãn. Dữ liệu thường được tạo từ từ điển.

```
df = pd.DataFrame({
    'Name': ['Alice', 'Bob', 'Charlie'],
    'Age': [25, 30, 35],
    'Department': ['HR', 'IT', 'Finance'],
    'Salary': [50000, 60000, 55000],
    'Date': ['2023-01-01', '2023-02-01', '2023-03-01']
})
```

Kết quả: DataFrame với 3 hàng, 5 cột, mỗi cột có kiểu dữ liệu riêng.

2 Truy cập dữ liệu

Truy cập cột

Truy cập cột bằng tên, trả về một **Series**.

```
df['Age']
```

Kết quả: Series chứa các giá trị [25, 30, 35].

Truy cập hàng

Sử dụng `loc[]` (theo nhãn) hoặc `iloc[]` (theo chỉ số).

```
df.loc[0, 'Name']  # Alice
df.iloc[1]['Age']  # 30
```

Tham số: `loc[`$row_label, column_label$`]`, `iloc[`$row_index, column_name/index$`]`.

3 Khám phá

- `df.head(n=5)`: Xem n dòng đầu (mặc định 5).
- `df.tail(n=5)`: Xem n dòng cuối (mặc định 5).
- `df.info()`: Hiển thị kiểu dữ liệu và thông tin cột.

- df.describe(): Thống kê mô tả (min, max, mean, v.v.) cho cột số.
- df.shape: Kích thước (số dòng, số cột).
- df.columns: Danh sách tên cột.
- df.dtypes: Kiểu dữ liệu của từng cột.

```
df.head()
df.info()
df.describe()
```

4 Xử lý dữ liệu thiếu

Kiểm tra và thay thế

- isnull(): Kiểm tra giá trị NaN, trả về True/False.
- fillna(value): Thay NaN bằng value (ví dụ: 0, mean).
- dropna(axis=0): Xóa hàng (axis=0) hoặc cột (axis=1) chứa NaN.

```
df.isnull()
df.fillna(0)
df.dropna()
```

Tham số: axis=0 (hàng), axis=1 (cột); how='any' (xóa nếu có NaN), how='all' (xóa nếu tất cả NaN).

5 Thống kê nhanh

Tính toán trên cột số liệu

- mean(): Trung bình.
- std(): Độ lệch chuẩn.

- value$_c ounts(normalize = False)$: Œmtnsut, normalize=True$trvt œl$.

```
df['Age'].mean()
df['Age'].std()
df['Department'].value_counts()
```

Ma trận tương quan

Tính tương quan Pearson giữa các cột số.

```
df[['Age', 'Salary']].corr()
```

Tham số: method='pearson' (mặc định), 'spearman', hoặc 'kendall'.

6 Lọc dữ liệu theo điều kiện

Lọc hàng dựa trên điều kiện logic, dùng & (AND), | (OR).

```
df[df['Age'] > 30]
df[(df['Age'] > 25) & (df['Department'] == 'IT')]
```

Lưu ý: Đặt mỗi điều kiện trong dấu ngoặc để tránh lỗi ưu tiên toán tử.

7 Sắp xếp dữ liệu

Sắp xếp DataFrame theo cột.

- sort$_values(by, ascending = True) : Spxptheoctby.$

```
df.sort_values('Age')
df.sort_values('Age', ascending=False)
```

Tham số: by: tên cột hoặc danh sách cột; ascending: True (tăng), False (giảm).

8 Nhóm và tổng hợp

GroupBy cơ bản

Nhóm dữ liệu theo cột và tính toán.

```
df.groupby('Department')['Age'].mean()
```

Tham số: by: cột để nhóm; áp dụng hàm như mean(), sum().

Tổng hợp nhiều cột

Tính toán khác nhau cho từng cột.

```
df.groupby('Department').agg({'Age': 'mean', 'Salary': 'sum'})
```

Tham số: agg: từ điển ánh xạ cột tới hàm (ví dụ: 'mean', 'sum', 'count').

9 Ghép bảng (Join/Merge)

Gộp hai DataFrame dựa trên cột chung.

- merge(df1, df2, on, how): Gộp theo cột on.
- how: 'inner' (giao), 'left' (trái), 'right' (phải), 'outer' (hợp).

```
df1 = pd.DataFrame({'ID': [1, 2], 'Name': ['Alice', 'Bob']})
df2 = pd.DataFrame({'ID': [1, 3], 'Salary': [5000, 6000]})
pd.merge(df1, df2, on='ID', how='inner')
pd.merge(df1, df2, on='ID', how='left')
```

Kết quả: inner chỉ giữ ID chung, left giữ tất cả hàng của df1.

10 Pivot table và Crosstab

Pivot Table

Tổng hợp dữ liệu đa chiều.

```
df.pivot_table(values='Salary', index='Department', columns='Date', aggfunc='sum')
```

Tham số: values: cột để tổng hợp; index: cột làm hàng; columns: cột làm cột; aggfunc: hàm tổng hợp ('sum', 'mean', v.v.).

Crosstab

Tạo bảng chéo tần suất.

```
pd.crosstab(df['Department'], df['Age'])
```

Tham số: normalize: False (đếm), 'index', 'columns', hoặc True (tỷ lệ).

11 Xử lý thời gian

Xử lý chuỗi thời gian với datetime.

```
df['Date'] = pd.to_datetime(df['Date'])
df = df.set_index('Date')
df.resample('M').mean()
```

Tham số resample: 'M' (tháng), 'D' (ngày), 'Y' (năm); áp dụng mean(), sum(), v.v.

12 Đọc/Ghi file

Đọc/ghi dữ liệu từ nhiều định dạng.

```
# CSV
df = pd.read_csv('data.csv')
df.to_csv('output.csv', index=False)

# Excel
df = pd.read_excel('data.xlsx')
df.to_excel('output.xlsx', index=False)

# JSON
df = pd.read_json('data.json')
df.to_json('output.json')
```

Tham số: index=False ngăn ghi chỉ số vào file.

13 Áp dụng hàm

Áp dụng hàm tùy chỉnh lên cột hoặc hàng.

- apply(func): Áp dụng hàm func lên cột hoặc hàng.

- replace(to$_{r}eplace, value)$: $Thaythgitr$.

```
df['Age'].apply(lambda x: x + 1)
df['Department'].replace({'HR': 'Human Resources'})
```

Tham số: axis=0 (cột), axis=1 (hàng) cho apply.

Tài liệu tham khảo

- Pandas chính thức: https://pandas.pydata.org/docs/
- Học Pandas trên Kaggle: https://www.kaggle.com/learn/pandas