



Bài giảng môn học:

Khoa Học Dữ Liệu (7080509)

CHƯƠNG 4: MỘT SỐ THƯ VIỆN PYTHON TRONG KHOA HỌC DỮ LIỆU (Phần 02)

Nội dung chương 4



4.1 Giới thiệu một số thư viện Python trong KHDL

4.2 Thư viện Numpy *

4.3 Thư viện Pandas *

4.4 Thư viện Matplotlib*

4.5 Thư viện Scikit-learn



Chương 4: Pandas package



Sử dụng thư viện Pandas trong phân tích và tiền xử lý dữ liệu

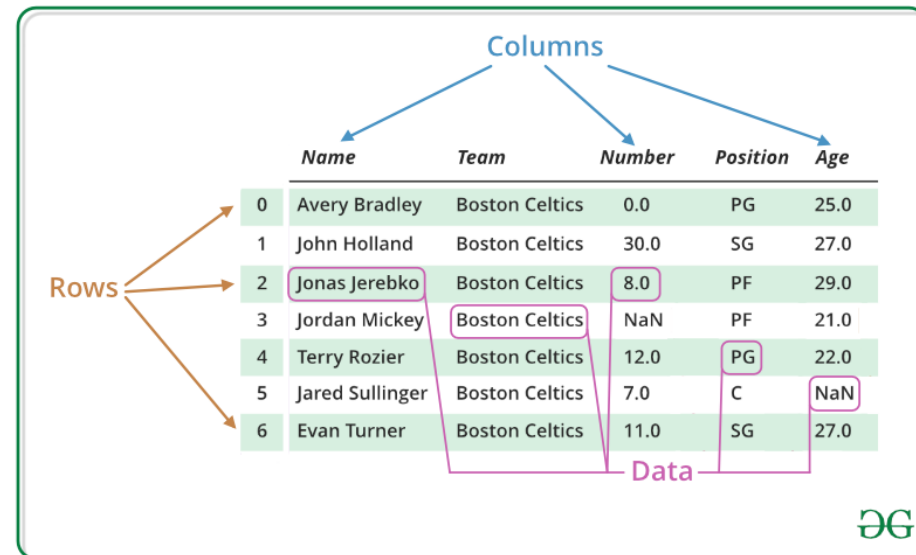


1. Giới thiệu Pandas

- **Pandas** là một thư viện mã nguồn mở với hiệu năng cao cho phép phân tích dữ liệu trong python được phát triển bởi Wes Mckinney năm 2008, một số tính năng nổi bật của pandas:
 - Có thể xử lý tập dữ liệu khác nhau về định dạng: chuỗi thời gian, bảng không đồng nhất, ma trận dữ liệu
 - Khả năng import dữ liệu từ nhiều nguồn khác nhau như CSV, DB/SQL
 - Có thể xử lý vô số phép toán cho tập dữ liệu: subsetting, slicing, filtering, merging, groupBy, re-ordering, and re-shaping,...
 - Xử lý dữ liệu mất mát theo ý người dùng mong muốn: bỏ qua hoặc chuyển sang 0
 - Xử lý, phân tích dữ liệu tốt như mô hình hoá và thống kê
 - Tích hợp tốt với các thư viện khác của python
 - Cung cấp hiệu suất tốt và có thể tăng tốc thậm chí hơn cả sử dụng Cython (extension C cho python)

2. Dataframe

- Là một trong 3 loại dữ liệu của Pandas (Series, Dataframe, Panel).
 - Dữ liệu 2 chiều, các cột có tên
 - Dữ liệu trên một cột là đồng nhất.
 - Các dòng có thể có thể có tên
 - Dữ liệu trong dataframe có thể bị thiếu



The diagram shows a table with 6 rows and 5 columns. The columns are labeled 'Name', 'Team', 'Number', 'Position', and 'Age'. The rows are indexed from 0 to 6. The data is as follows:

| | Name | Team | Number | Position | Age |
|---|-----------------|----------------|--------|----------|------|
| 0 | Avery Bradley | Boston Celtics | 0.0 | PG | 25.0 |
| 1 | John Holland | Boston Celtics | 30.0 | SG | 27.0 |
| 2 | Jonas Jerebko | Boston Celtics | 8.0 | PF | 29.0 |
| 3 | Jordan Mickey | Boston Celtics | NaN | PF | 21.0 |
| 4 | Terry Rozier | Boston Celtics | 12.0 | PG | 22.0 |
| 5 | Jared Sullinger | Boston Celtics | 7.0 | C | NaN |
| 6 | Evan Turner | Boston Celtics | 11.0 | SG | 27.0 |

Annotations: 'Columns' points to the header row. 'Rows' points to the row indices. 'Data' points to the data cells, highlighting missing values (NaN) in the 'Number' and 'Position' columns.

2. Đọc/lưu dữ liệu vào dataframe

Read and Write to CSV

```
>>> pd.read_csv('file.csv', header=None, nrows=5)
>>> df.to_csv('myDataFrame.csv')
```

Read and Write to Excel

```
>>> pd.read_excel('file.xlsx')
>>> pd.to_excel('dir/myDataFrame.xlsx', sheet_name='Sheet1')
```

Read multiple sheets from the same file

```
>>> xlsx = pd.ExcelFile('file.xls')
>>> df = pd.read_excel(xlsx, 'Sheet1')
```

2. Đọc/lưu dữ liệu vào dataframe

| | A | B | C | D | E | F | G |
|----|--------------|--------|-------|---------|-----------|-------------|--------|
| 1 | time | Ha Noi | Vinh | Da Nang | Nha Trang | Ho Chi Minh | Ca Mau |
| 2 | 00 15-9-2019 | 25.65 | 24.79 | 24.01 | 25.06 | 25.48 | 24.97 |
| 3 | 01 15-9-2019 | 25.31 | 24.21 | 24.02 | 24.93 | 25.16 | 24.83 |
| 4 | 02 15-9-2019 | 25.05 | 23.73 | 23.89 | 24.79 | 24.8 | 24.55 |
| 5 | 03 15-9-2019 | 24.79 | 23.36 | 23.83 | 24.84 | 24.74 | 24.48 |
| 6 | 04 15-9-2019 | 24.59 | 23.05 | 23.69 | 24.82 | 24.8 | 24.38 |
| 7 | 05 15-9-2019 | 24.4 | 22.8 | 23.52 | 24.79 | 24.87 | 24.4 |
| 8 | 06 15-9-2019 | 24.38 | 22.79 | 23.68 | 25.1 | 24.71 | 24.41 |
| 9 | 07 15-9-2019 | 26.72 | 25.61 | 24.92 | 26.56 | 25.03 | 24.91 |
| 10 | 08 15-9-2019 | 28.84 | 26.93 | | | | |
| 11 | 09 15-9-2019 | 30.29 | 28.72 | | | | |
| 12 | 10 15-9-2019 | 31.35 | 29.97 | | | | |
| 13 | 11 15-9-2019 | 32.05 | 28.93 | | | | |
| 14 | 12 15-9-2019 | 31.31 | 28.94 | | | | |
| 15 | 13 15-9-2019 | 30.95 | 30.25 | | | | |
| 16 | 14 15-9-2019 | 30.56 | 30.62 | | | | |
| 17 | 15 15-9-2019 | 31.13 | 30.58 | | | | |
| 18 | 16 15-9-2019 | 30.8 | 30.2 | | | | |
| 19 | 17 15-9-2019 | 29.04 | 29.36 | | | | |

```

1 #Khai báo sử dụng thư viện Pandas
2 import pandas as pd
3 #Xác định đường dẫn tới file dữ liệu
4 path = 'Data_C4/Data_Temp.csv'
5 #Đọc file dữ liệu csv với pandas
6 data_df = pd.read_csv(path)
7
8 print('Kiểu dữ liệu:', type(data_df))

```

Kiểu dữ liệu: <class 'pandas.core.frame.DataFrame'>

3. Truy cập dữ liệu dataframe

- **df.head(num):** Truy cập num dòng dữ liệu đầu tiên của dataframe df (mặc định num = 5)
- **df.tail(num):** Truy cập num dòng dữ liệu cuối cùng của dataframe df (mặc định num = 5)

| 1 | #Hiển thị 5 dòng dữ liệu đầu tiên | | | | | |
|---|-----------------------------------|--------|-------|---------|-----|--|
| 2 | data_df.head() | | | | | |
| | time | Ha Noi | Vinh | Da Nang | Nha | |
| 0 | 00 15-9-2019 | 25.65 | 24.79 | 24.01 | | |
| 1 | 01 15-9-2019 | 25.31 | 24.21 | 24.02 | | |
| 2 | 02 15-9-2019 | 25.05 | 23.73 | 23.89 | | |
| 3 | 03 15-9-2019 | 24.79 | 23.36 | 23.83 | | |
| 4 | 04 15-9-2019 | 24.59 | 23.05 | 23.69 | | |

| 1 | #Hiển thị 10 dòng dữ liệu cuối cùng | | | | | | | |
|-----|-------------------------------------|--------|-------|---------|-----------|-------------|--------|--|
| 2 | data_df.tail(10) | | | | | | | |
| | time | Ha Noi | Vinh | Da Nang | Nha Trang | Ho Chi Minh | Ca Mau | |
| 182 | 14 22-9-2019 | 30.96 | 29.86 | 26.61 | 26.48 | 26.59 | 28.05 | |
| 183 | 15 22-9-2019 | 30.78 | 29.30 | 26.42 | 26.37 | 26.57 | 27.94 | |
| 184 | 16 22-9-2019 | 30.36 | 28.63 | 26.06 | 26.38 | 26.48 | 27.85 | |
| 185 | 17 22-9-2019 | 29.28 | 27.62 | 25.88 | 26.35 | 26.25 | 28.19 | |
| 186 | 18 22-9-2019 | 27.44 | 25.30 | 24.53 | 26.20 | 25.86 | 27.01 | |
| 187 | 19 22-9-2019 | 26.56 | 24.92 | 24.10 | 26.11 | 25.56 | 26.43 | |
| 188 | 20 22-9-2019 | 25.69 | 24.77 | 23.76 | 25.97 | 25.23 | 25.88 | |
| 189 | 21 22-9-2019 | 24.81 | 24.47 | 23.40 | 25.86 | 25.05 | 25.29 | |
| 190 | 22 22-9-2019 | 23.97 | 24.22 | 22.95 | 25.74 | 24.92 | 24.87 | |
| 191 | 23 22-9-2019 | 22.84 | 23.99 | 22.59 | 25.50 | 24.77 | 24.57 | |

3. Truy cập dữ liệu dataframe

- **df[start:finish]:** Truy cập từ dòng start → finish của dataframe df

```
1 #Hiển thị dữ liệu từ dòng 144 tới 149
2 data_df[144:150]
```

| | time | Ha Noi | Vinh | Da Nang | Nha Trang | Ho Chi Minh | Ca Mau |
|-----|--------------|--------|-------|---------|-----------|-------------|--------|
| 144 | 00 21-9-2019 | 25.41 | 24.30 | 23.92 | 25.25 | 25.06 | 24.87 |
| 145 | 01 21-9-2019 | 24.71 | 23.92 | 23.41 | 25.03 | 24.79 | 24.83 |
| 146 | 02 21-9-2019 | 23.83 | 23.58 | 23.02 | 24.95 | 24.54 | 24.77 |
| 147 | 03 21-9-2019 | 23.25 | 24.10 | 22.65 | 24.86 | 24.33 | 24.64 |
| 148 | 04 21-9-2019 | 22.92 | 24.15 | 22.40 | 24.71 | 24.08 | 24.28 |
| 149 | 05 21-9-2019 | 22.58 | 24.09 | 22.24 | 24.54 | 23.92 | 24.12 |

3. Truy cập dữ liệu dataframe

- **df.columns** : Liệt kê tên các cột trong dataframe df
- **df[['Col1', 'Col2', 'Col3']]**: Chỉ truy cập dữ liệu của các cột có tên Col1, Col2, Col3 trong dataframe df

```
1 #Lấy danh sách tên các cột trong data frame
2 data_df.columns
```

```
Index(['time', 'Ha Noi', 'Vinh', 'Da Nang', 'Nha Trang', 'Ho Chi Minh',  
      'Ca Mau'],  
      dtype='object')
```

```
1 #Hiển thị dữ liệu thời gian và nhiệt độ của Đà Nẵng
2 data_df[['time', 'Da Nang']]
```

| | time | Da Nang |
|---|--------------|---------|
| 0 | 00 15-9-2019 | 24.01 |
| 1 | 01 15-9-2019 | 24.02 |
| 2 | 02 15-9-2019 | 23.89 |
| 3 | 03 15-9-2019 | 23.83 |

3. Truy cập dữ liệu dataframe

- `df.iloc[num_row, num_col]`: Truy cập tới dữ liệu của hàng và cột qua chỉ số `num_row`, `num_col`

```
1 # sử dụng .iloc để truy vấn dữ liệu
2 #Lấy dữ liệu tại dòng thứ 10 trong data frame
3 data_df.iloc[10]
```

1

Out[30]:

```
time          10 15-9-2019
Ha Noi        31.35
Vinh          29.97
Da Nang       26.96
Nha Trang     27.23
Ho Chi Minh   27.68
Ca Mau        27.53
Name: 10, dtype: object
```

```
1 #Lựa chọn dữ liệu từ hàng 24 đến 29
2 #Cột dữ liệu 0 và 3
3 data_df.iloc[24:30,[0,3]]
```

2

Out[33]:

| | time | Da Nang |
|----|--------------|---------|
| 24 | 00 16-9-2019 | 23.49 |
| 25 | 01 16-9-2019 | 23.49 |
| 26 | 02 16-9-2019 | 23.71 |
| 27 | 03 16-9-2019 | 23.76 |
| 28 | 04 16-9-2019 | 23.52 |
| 29 | 05 16-9-2019 | 23.38 |

3. Truy cập dữ liệu dataframe

- `df.iloc[[num_row],[num_col]]`: Truy cập tới dữ liệu của hàng và cột qua **chỉ số num_row, num_col**

```
1 # sử dụng .iloc để truy vấn dữ liệu
2 #Lấy dữ liệu tại dòng thứ 10 trong data frame
3 data_df.iloc[10]
```

Out[30]:

```
time          10 15-9-2019
Ha Noi        31.35
Vinh          29.97
Da Nang       26.96
Nha Trang    27.23
Ho Chi Minh   27.68
Ca Mau       27.53
Name: 10, dtype: object
```

```
1 #Lựa chọn dữ liệu từ hàng 24 đến 29
2 #Cột dữ liệu 0 và 3
3 data_df.iloc[24:30,[0,3]]
```

Out[33]:

| | time | Da Nang |
|----|--------------|---------|
| 24 | 00 16-9-2019 | 23.49 |
| 25 | 01 16-9-2019 | 23.49 |
| 26 | 02 16-9-2019 | 23.71 |
| 27 | 03 16-9-2019 | 23.76 |
| 28 | 04 16-9-2019 | 23.52 |
| 29 | 05 16-9-2019 | 23.38 |

3. Truy cập dữ liệu dataframe

- `df.loc[[num_row],[name_col]]`: Truy cập tới dữ liệu của hàng và cột qua **chỉ số num_row, tên cột name_col**

```
1 #Sử dụng .loc để truy vấn dữ liệu trong data frame
2 data_df.loc[24:30,['time','Da Nang']]
```

| | time | Da Nang |
|----|--------------|---------|
| 24 | 00 16-9-2019 | 23.49 |
| 25 | 01 16-9-2019 | 23.49 |
| 26 | 02 16-9-2019 | 23.71 |
| 27 | 03 16-9-2019 | 23.76 |
| 28 | 04 16-9-2019 | 23.52 |
| 29 | 05 16-9-2019 | 23.38 |
| 30 | 06 16-9-2019 | 23.32 |

Khác nhau giữa .iloc và .loc ở tham số thứ 2 (select column):

.iloc truyền vào là chỉ số cột (int)

.loc truyền vào tên cột (label)

4. Quan sát dữ liệu với Pandas



4. Quan sát dữ liệu dataframe

- **df.info():** Hiển thị thông tin tổng quan của dataframe df bao gồm: Số hàng, số cột, số lượng dữ liệu không null, kiểu dữ liệu của từng thuộc tính.

```
1 # sử dụng .info để quan sát dữ liệu Data frame
2 data_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 192 entries, 0 to 191
Data columns (total 7 columns):
time                192 non-null object
Ha Noi              192 non-null float64
Vinh                192 non-null float64
Da Nang             192 non-null float64
Nha Trang           192 non-null float64
Ho Chi Minh         192 non-null float64
Ca Mau              192 non-null float64
dtypes: float64(6), object(1)
memory usage: 10.6+ KB
```


4. Quan sát dữ liệu dataframe (2)

- **df.shape:** Kích thước của dataframe
- **df.count():** Đếm số dòng dữ liệu không null trong dataframe

```
1 #Xác định kích cỡ của Data Frame
2 print('Kích thước của Data:',data_df.shape)
3 #Đếm số Lượng hàng dữ liệu không null theo từng cột
4 print('Số liệu của từng cột:')
5 print(data_df.count())
```

Kích thước của Data: (192, 7)

Số liệu của từng cột:

| | |
|-------------|-------|
| time | 192 |
| Ha Noi | 192 |
| Vinh | 192 |
| Da Nang | 192 |
| Nha Trang | 192 |
| Ho Chi Minh | 192 |
| Ca Mau | 192 |
| dtype: | int64 |

4. Quan sát dữ liệu dataframe (3)

- **df.describe ()**: Thực hiện tính toán các đặc trưng thống kê của dataframe (các thuộc tính số) bao gồm: Tổng số giá trị, giá trị trung bình, độ lệch chuẩn, giá trị max, min...

```
1 #Thực hiện thống kê dữ liệu
2 data_df.describe()
```

| | Ha Noi | Vinh | Da Nang | Nha Trang | Ho Chi Minh | Ca Mau |
|-------|------------|------------|------------|------------|-------------|------------|
| count | 192.000000 | 192.000000 | 192.000000 | 192.000000 | 192.000000 | 192.000000 |
| mean | 27.712292 | 26.719896 | 25.522500 | 26.166875 | 26.159219 | 26.732552 |
| std | 2.749369 | 2.314602 | 1.932761 | 0.923510 | 1.719259 | 1.821799 |
| min | 21.680000 | 22.600000 | 20.930000 | 24.500000 | 23.220000 | 23.990000 |
| 25% | 25.645000 | 24.875000 | 24.010000 | 25.485000 | 24.797500 | 25.315000 |
| 50% | 27.685000 | 26.360000 | 25.310000 | 26.085000 | 25.930000 | 26.265000 |
| 75% | 29.947500 | 28.022500 | 26.932500 | 26.795000 | 27.485000 | 28.057500 |
| max | 33.450000 | 32.570000 | 29.880000 | 28.680000 | 31.060000 | 31.370000 |

Những thông tin trên giúp cho chúng ta có cảm nhận tổng quan và sự phân tán về dữ liệu, từ đó ta tìm kiếm phương pháp phù hợp để xử lý tiếp theo.

4. Quan sát dữ liệu dataframe (4)

- **df.describe (include=['O']):** Thực hiện tính toán các đặc trưng thống kê của dataframe (các thuộc tính có kiểu Object) bao gồm: Tổng số giá trị (count), số giá trị khác nhau xuất hiện trong thuộc tính (unique), Tên giá trị xuất hiện nhiều nhất (top), Số lần xuất hiện của thuộc tính đó (freq).

```
In [24]: 1 #Thông kê tập dữ liệu Train các thuộc tính có dtype: Object  
2 train_df.describe(include=['O'])
```

Out[24]:

| | Name | Sex | Ticket | Cabin | Embarked |
|--------|-----------------------------------|------|--------|-------|----------|
| count | 891 | 891 | 891 | 204 | 889 |
| unique | 891 | 2 | 681 | 147 | 3 |
| top | Mellinger, Miss. Madeleine Violet | male | 347082 | G6 | S |
| freq | 1 | 577 | 7 | 4 | 644 |

5. Phát hiện và **xử lý** dữ liệu mất mát (missing data) với pandas



Dữ liệu mất mát/thiếu (missing data)

Các nguyên nhân dẫn đến missing data:

- Khuyết ngẫu nhiên (Missing at Random – MAR):
- Khuyết hoàn toàn ngẫu nhiên (Missing Completely at Random – MCAR):
- Khuyết không ngẫu nhiên (Missing not at Random – MNAR):

| | A | B | C | D | E | F | G |
|----|--------------|--------|-------|---------|-----------|-------------|--------|
| 1 | time | Ha Noi | Vinh | Da Nang | Nha Trang | Ho Chi Minh | Ca Mau |
| 2 | 00 15-9-2019 | 25.65 | 24.79 | 24.01 | 25.06 | 25.48 | 24.97 |
| 3 | 01 15-9-2019 | | 24.21 | 24.02 | 24.93 | 25.16 | 24.83 |
| 4 | 02 15-9-2019 | 25.05 | 23.73 | 23.89 | 24.79 | 24.8 | 24.55 |
| 5 | 03 15-9-2019 | 24.79 | 23.36 | 23.83 | | 24.74 | 24.48 |
| 6 | 04 15-9-2019 | 24.59 | 23.05 | 23.69 | 24.82 | 24.8 | 24.38 |
| 7 | 05 15-9-2019 | 24.4 | | 23.52 | 24.79 | 24.87 | 24.4 |
| 8 | 06 15-9-2019 | 24.38 | 22.79 | 23.68 | 25.1 | 24.71 | 24.41 |
| 9 | 07 15-9-2019 | 26.72 | 25.61 | 24.92 | 26.56 | 25.03 | 24.91 |
| 10 | 08 15-9-2019 | 28.84 | 26.93 | 26.51 | 26.53 | 25.75 | 25.85 |
| 11 | 09 15-9-2019 | 30.29 | 28.72 | 27.48 | 26.95 | 26.64 | 26.79 |
| 12 | 10 15-9-2019 | | 29.97 | | | 27.68 | 27.53 |
| 13 | 11 15-9-2019 | 32.05 | 28.93 | 26.86 | 27.38 | 28.43 | 28.98 |
| 14 | 12 15-9-2019 | 31.31 | 28.94 | 26.65 | 27.47 | 28.29 | 29.24 |
| 15 | 13 15-9-2019 | 30.95 | | 27.83 | 27.44 | 28 | 30.66 |
| 16 | 14 15-9-2019 | 30.56 | 30.62 | 26.49 | 27.16 | 27.67 | 30.97 |
| 17 | 15 15-9-2019 | 31.13 | 30.58 | 26.29 | 26.68 | 27.29 | 30.59 |
| 18 | 16 15-9-2019 | 30.8 | 30.2 | | 26.45 | 27.29 | 29.13 |
| 19 | 17 15-9-2019 | 29.94 | 29.36 | 25.8 | 26.67 | 26.69 | 28.72 |
| 20 | 18 15-9-2019 | 28.53 | 27.48 | 24.82 | 25.92 | 25.81 | 27.46 |
| 21 | 19 15-9-2019 | 28.89 | 27.03 | 24.93 | 25.88 | 25.93 | 27.07 |
| 22 | 20 15-9-2019 | 28.06 | 26.41 | 24.7 | | 25.97 | 26.75 |
| 23 | 21 15-9-2019 | 27.43 | 26.2 | 24.41 | 25.62 | 25.94 | 26.32 |
| 24 | 22 15-9-2019 | 26.98 | 25.79 | 24.17 | 25.6 | 25.9 | 26.29 |
| 25 | 23 15-9-2019 | 26.68 | 25.31 | 23.81 | 25.53 | 25.8 | 26.36 |
| 26 | | | | | | | |
| 27 | | | | | | | |

```

1 #Đọc file dữ liệu chứa missing
2 #Khái báo sử dụng thư viện Pandas
3 import pandas as pd
4 #Xác định đường dẫn tới file dữ liệu missing
5 path = 'Data_C4/Data_Temp_missing.csv'
6 #Đọc file dữ liệu csv với pandas
7 data_temp = pd.read_csv(path)
8 data_temp

```

| | time | Ha Noi | Vinh | Da Nang | Nha Trang | Ho Chi Minh | Ca Mau |
|---|--------------|--------|-------|---------|-----------|-------------|--------|
| 0 | 00 15-9-2019 | 25.65 | 24.79 | 24.01 | 25.06 | 25.48 | 24.97 |
| 1 | 01 15-9-2019 | NaN | 24.21 | 24.02 | 24.93 | 25.16 | 24.83 |
| 2 | 02 15-9-2019 | 25.05 | 23.73 | 23.89 | 24.79 | 24.80 | 24.55 |
| 3 | 03 15-9-2019 | 24.79 | 23.36 | 23.83 | NaN | 24.74 | 24.48 |

5.1 Phát hiện dữ liệu missing

- Thống kê dữ liệu missing trong dataframe:
 - `df.isnull().sum()`

```
1 #Thống kê số liệu missing trong Data frame
2 #Theo từng cột
3 print('Số lượng missing data trong file dữ liệu:')
4 print(data_temp.isnull().sum())
```

Số lượng missing data trong file dữ liệu:

| | |
|-------------|-------|
| time | 0 |
| Ha Noi | 2 |
| Vinh | 2 |
| Da Nang | 2 |
| Nha Trang | 3 |
| Ho Chi Minh | 0 |
| Ca Mau | 0 |
| dtype: | int64 |

5.1 Phát hiện dữ liệu missing (2)

- **Thống kê dữ liệu missing trong dataframe:**
 - Xây dựng hàm thống kê `missing_values()`

```
1  #Xây dựng hàm thống kê dữ liệu missing trong dataframe:
2  #-----
3  #Đầu vào của hàm là 1 biến Dataframe
4  #Đầu ra bao gồm các thông số:
5  #Tổng số cột của file dữ liệu
6  #Tổng số cột có chứa dữ liệu missing
7  #Danh sách các cột chứa dữ liệu missing với 2 thống số:
8  #Tổng số giá trị missing tương ứng với cột đó
9  #Tỷ lệ % dữ liệu missing trên tổng số dữ liệu của cột
10 def missing_values(df):
11     mis_val = df.isnull().sum()
12     mis_val_percent = 100 * df.isnull().sum() / len(df)
13     mis_val_table = pd.concat([mis_val, mis_val_percent], axis=1)
14     mis_val_table_ren_columns = mis_val_table.rename(
15         columns = {0 : 'Số giá trị Missing', 1 : 'Tỷ lệ % missing'})
16     mis_val_table_ren_columns = mis_val_table_ren_columns[
17         mis_val_table_ren_columns.iloc[:,1] != 0].sort_values(
18         'Tỷ lệ % missing', ascending=False).round(1)
19     print ("File dữ liệu bao gồm có: " + str(df.shape[1]) + " cột.\n"
20           "Có " + str(mis_val_table_ren_columns.shape[0]) +
21           " cột chứa missing values.")
22     return mis_val_table_ren_columns
```


5.1 Phát hiện dữ liệu missing (2)

- **Thống kê dữ liệu missing trong dataframe:**
 - Xây dựng hàm thống kê `missing_values()`

```
1 missing_values(data_temp)
```

File dữ liệu bao gồm có: 7 cột.
Có 4 cột chứa missing values.

| | Số giá trị Missing | Tỷ lệ % missing |
|-----------|--------------------|-----------------|
| Nha Trang | 3 | 12.5 |
| Ha Noi | 2 | 8.3 |
| Vinh | 2 | 8.3 |
| Da Nang | 2 | 8.3 |

5.1 Phát hiện dữ liệu missing (3)

- **df.isnull().any(axis=1):** Kiểm tra trong từng hàng có thuộc tính nào chứa giá trị missing hay không? Nếu trong hàng chỉ cần có 1 thuộc tính missing – True

```
1  #Liệt kê danh sách các row bị missing data
2  #(Row có chứa thuộc tính bất kỳ bị missing - True )
3  #axis=1: Liệt kê các hàng | axis=0: Liệt kê các cột
4  data_temp.isnull().any(axis=1)
```

```
0    False
1     True
2    False
3     True
4    False
5     True
6    False
7    False
8    False
9    False
10    True
11   False
```

5.1 Phát hiện dữ liệu missing (4)

- `df[df.isnull().any(axis=1)]`: Liệt kê chi tiết các hàng có chứa giá trị null trong một thuộc tính bất kỳ

```
1 #Liệt kê chi tiết các hàng có chứa giá trị null trong một thuộc tính bất kỳ
2 data_temp[data_temp.isnull().any(axis=1)]
```

| | time | Ha Noi | Vinh | Da Nang | Nha Trang | Ho Chi Minh | Ca Mau |
|----|--------------|--------|-------|---------|-----------|-------------|--------|
| 1 | 01 15-9-2019 | NaN | 24.21 | 24.02 | 24.93 | 25.16 | 24.83 |
| 3 | 03 15-9-2019 | 24.79 | 23.36 | 23.83 | NaN | 24.74 | 24.48 |
| 5 | 05 15-9-2019 | 24.40 | NaN | 23.52 | 24.79 | 24.87 | 24.40 |
| 10 | 10 15-9-2019 | NaN | 29.97 | NaN | NaN | 27.68 | 27.53 |
| 13 | 13 15-9-2019 | 30.95 | NaN | 27.83 | 27.44 | 28.00 | 30.66 |
| 16 | 16 15-9-2019 | 30.80 | 30.20 | NaN | 26.45 | 27.29 | 29.13 |
| 20 | 20 15-9-2019 | 28.06 | 26.41 | 24.70 | NaN | 25.97 | 26.75 |

5.1 Phát hiện dữ liệu missing (5)

- `pd.isnull(df["F1"])`: Liệt kê các hàng có chứa giá trị null trong một thuộc tính được chỉ định.

```
1 #Liệt kê các hàng có chứa giá trị null trong một cột được chỉ định.
2 x = pd.isnull(data_temp['Ha Noi'])
3 data_temp[x]
```

| | time | Ha Noi | Vinh | Da Nang | Nha Trang | Ho Chi Minh | Ca Mau |
|----|--------------|--------|-------|---------|-----------|-------------|--------|
| 1 | 01 15-9-2019 | NaN | 24.21 | 24.02 | 24.93 | 25.16 | 24.83 |
| 10 | 10 15-9-2019 | NaN | 29.97 | NaN | NaN | 27.68 | 27.53 |

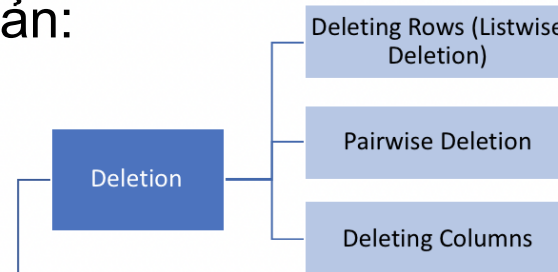
```
1 #Liệt kê các hàng có chứa giá trị null trong một cột được chỉ định.
2 x = pd.isnull(data_temp['Nha Trang'])
3 data_temp[x]
```

| | time | Ha Noi | Vinh | Da Nang | Nha Trang | Ho Chi Minh | Ca Mau |
|----|--------------|--------|-------|---------|-----------|-------------|--------|
| 3 | 03 15-9-2019 | 24.79 | 23.36 | 23.83 | NaN | 24.74 | 24.48 |
| 10 | 10 15-9-2019 | NaN | 29.97 | NaN | NaN | 27.68 | 27.53 |
| 20 | 20 15-9-2019 | 28.06 | 26.41 | 24.70 | NaN | 25.97 | 26.75 |

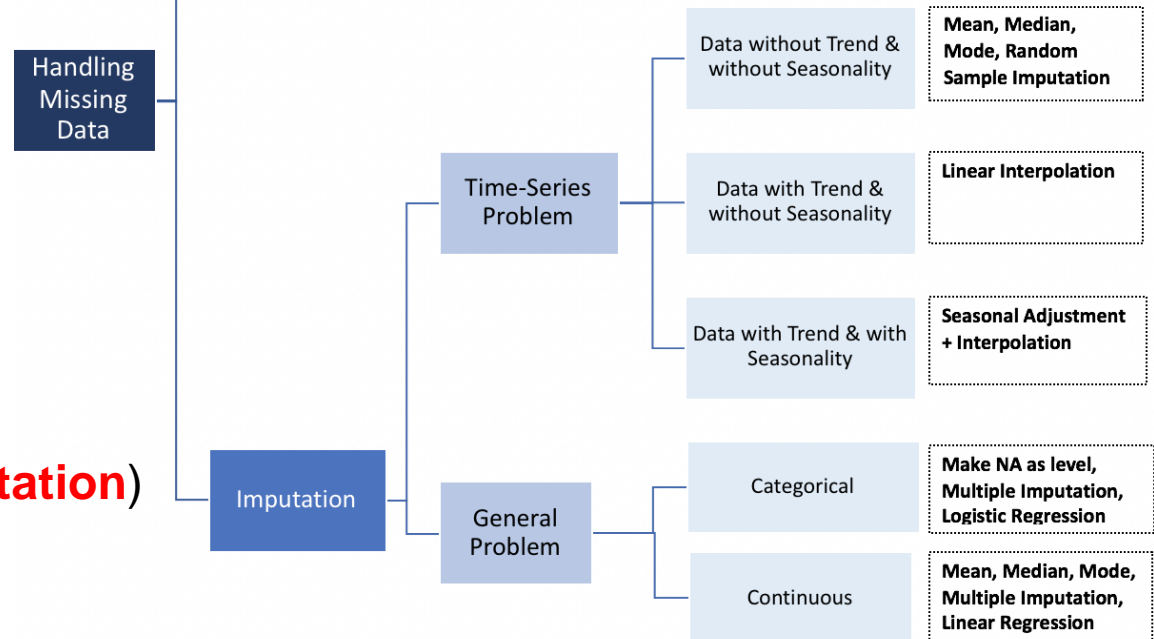
5.2 Xử lý dữ liệu missing

- Để xử lý dữ liệu missing cần phải hiểu sâu sắc tập dữ liệu, việc lựa chọn phương pháp nào phụ thuộc vào từng bài toán cụ thể, một số phương pháp xử lý dữ liệu missing cơ bản:

1) Loại bỏ các missing (**Deletion**)



2) Thay thế các missing (**Imputation**)



1) Loại bỏ các missing (Deletion)

df.dropna(axis=0) → loại bỏ hàng

```

1 #1) Phương pháp 1: Loại bỏ các dữ liệu missing (Deletion)
2
3 #Xóa toàn bộ các hàng chứa missing data: axis=0 -> xóa hàng
4 data_new = data_temp.dropna(axis=0, how='any')
5 #Kết quả sau khi loại bỏ các row chứa missing
6 print(data_new)

```

| | | time | Ha Noi | Vinh | Da Nang | Nha Trang | Ho Chi Minh | Ca Mau |
|----|----|-----------|--------|-------|---------|-----------|-------------|--------|
| 0 | 00 | 15-9-2019 | 25.65 | 24.79 | 24.01 | 25.06 | 25.48 | 24.97 |
| 2 | 02 | 15-9-2019 | 25.05 | 23.73 | 23.89 | 24.79 | 24.80 | 24.55 |
| 4 | 04 | 15-9-2019 | 24.59 | 23.05 | 23.69 | 24.82 | 24.80 | 24.38 |
| 6 | 06 | 15-9-2019 | 24.38 | 22.79 | 23.68 | 25.10 | 24.71 | 24.41 |
| 7 | 07 | 15-9-2019 | 26.72 | 25.61 | 24.92 | 26.56 | 25.03 | 24.91 |
| 8 | 08 | 15-9-2019 | 28.84 | 26.93 | 26.51 | 26.53 | 25.75 | 25.85 |
| 9 | 09 | 15-9-2019 | 30.29 | 28.72 | 27.48 | 26.95 | 26.64 | 26.79 |
| 11 | 11 | 15-9-2019 | 32.05 | 28.93 | 26.86 | 27.38 | 28.43 | 28.98 |
| 12 | 12 | 15-9-2019 | 31.31 | 28.94 | 26.65 | 27.47 | 28.29 | 29.24 |
| 14 | 14 | 15-9-2019 | 30.56 | 30.62 | 26.49 | 27.16 | 27.67 | 30.97 |
| 15 | 15 | 15-9-2019 | 31.13 | 30.58 | 26.29 | 26.68 | 27.29 | 30.59 |
| 17 | 17 | 15-9-2019 | 29.94 | 29.36 | 25.80 | 26.67 | 26.69 | 28.72 |
| 18 | 18 | 15-9-2019 | 28.53 | 27.48 | 24.82 | 25.92 | 25.81 | 27.46 |
| 19 | 19 | 15-9-2019 | 28.89 | 27.03 | 24.93 | 25.88 | 25.93 | 27.07 |
| 21 | 21 | 15-9-2019 | 27.43 | 26.20 | 24.41 | 25.62 | 25.94 | 26.32 |
| 22 | 22 | 15-9-2019 | 26.98 | 25.79 | 24.17 | 25.60 | 25.90 | 26.29 |
| 23 | 23 | 15-9-2019 | 26.68 | 25.31 | 23.81 | 25.53 | 25.80 | 26.36 |

1) Loại bỏ các missing (Deletion)

df.dropna(axis=1) → loại bỏ cột

```
1 #1) Phương pháp 1: Loại bỏ các dữ liệu missing (Deletion)
2
3 #Xóa toàn bộ các cột chứa missing data: axis=1 -> xóa cột
4 data_new = data_temp.dropna(axis=1, how='any')
5 #Kết quả sau khi loại bỏ các cột chứa missing
6 print(data_new)
```

| | time | Ho Chi Minh | Ca Mau |
|----|--------------|-------------|--------|
| 0 | 00 15-9-2019 | 25.48 | 24.97 |
| 1 | 01 15-9-2019 | 25.16 | 24.83 |
| 2 | 02 15-9-2019 | 24.80 | 24.55 |
| 3 | 03 15-9-2019 | 24.74 | 24.48 |
| 4 | 04 15-9-2019 | 24.80 | 24.38 |
| 5 | 05 15-9-2019 | 24.87 | 24.40 |
| 6 | 06 15-9-2019 | 24.71 | 24.41 |
| 7 | 07 15-9-2019 | 25.03 | 24.91 |
| 8 | 08 15-9-2019 | 25.75 | 25.85 |
| 9 | 09 15-9-2019 | 26.64 | 26.79 |
| 10 | 10 15-9-2019 | 27.68 | 27.53 |
| 11 | 11 15-9-2019 | 28.43 | 28.98 |
| 12 | 12 15-9-2019 | 28.29 | 29.24 |
| 13 | 13 15-9-2019 | 28.00 | 30.66 |
| 14 | 14 15-9-2019 | 27.67 | 30.97 |
| 15 | 15 15-9-2019 | 27.29 | 30.59 |
| 16 | 16 15-9-2019 | 27.29 | 29.13 |
| 17 | 17 15-9-2019 | 26.69 | 28.72 |

Các cột **Hà Nội, Vinh, Đà Nẵng, Nha Trang** có chứa dữ liệu missing đã bị loại bỏ

2) Thay thế các missing (imputation)

df.fillna(value) → thay thế bằng một giá trị cố định

```
1 #PHƯƠNG PHÁP 2: Thay thế (Imputation)
2 #2.1) Thay thế các dữ liệu mất mát bằng một hằng số cố định
3 value = 25.0
4 #thay thế các giá trị missing bằng một giá trị cố định Value
5 data_new = data_temp.fillna(value)
6 print(data_new)
```

| | time | Ha Noi | Vinh | Da Nang | Nha Trang | Ho Chi Minh | Ca Mau |
|----|--------------|--------|-------|---------|-----------|-------------|--------|
| 0 | 00 15-9-2019 | 25.65 | 24.79 | 24.01 | 25.06 | 25.48 | 24.97 |
| 1 | 01 15-9-2019 | 25.00 | 24.21 | 24.02 | 24.93 | 25.16 | 24.83 |
| 2 | 02 15-9-2019 | 25.05 | 23.73 | 23.89 | 24.79 | 24.80 | 24.55 |
| 3 | 03 15-9-2019 | 24.79 | 23.36 | 23.83 | 25.00 | 24.74 | 24.48 |
| 4 | 04 15-9-2019 | 24.59 | 23.05 | 23.69 | 24.82 | 24.80 | 24.38 |
| 5 | 05 15-9-2019 | 24.40 | 25.00 | 23.52 | 24.79 | 24.87 | 24.40 |
| 6 | 06 15-9-2019 | 24.38 | 22.79 | 23.68 | 25.10 | 24.71 | 24.41 |
| 7 | 07 15-9-2019 | 26.72 | 25.61 | 24.92 | 26.56 | 25.03 | 24.91 |
| 8 | 08 15-9-2019 | 28.84 | 26.93 | 26.51 | 26.53 | 25.75 | 25.85 |
| 9 | 09 15-9-2019 | 30.29 | 28.72 | 27.48 | 26.95 | 26.64 | 26.79 |
| 10 | 10 15-9-2019 | 25.00 | 29.97 | 25.00 | 25.00 | 27.68 | 27.53 |
| 11 | 11 15-9-2019 | 32.05 | 28.93 | 26.86 | 27.38 | 28.43 | 28.98 |
| 12 | 12 15-9-2019 | 31.31 | 28.94 | 26.65 | 27.47 | 28.29 | 29.24 |
| 13 | 13 15-9-2019 | 30.95 | 25.00 | 27.83 | 27.44 | 28.00 | 30.66 |
| 14 | 14 15-9-2019 | 30.56 | 30.62 | 26.49 | 27.16 | 27.67 | 30.97 |
| 15 | 15 15-9-2019 | 31.13 | 30.58 | 26.29 | 26.68 | 27.29 | 30.59 |
| 16 | 16 15-9-2019 | 30.80 | 30.20 | 25.00 | 26.45 | 27.29 | 29.13 |

2) Thay thế các missing (imputation)

df.fillna(method='pad') → thay thế bằng giá trị liền trước

```
1 #PHƯƠNG PHÁP 2: Thay thế (Imputation)
2 #2.2) Thay thế các dữ liệu mất mát bằng giá trị liền trước của nó
3 data_new2 = data_temp.fillna(method='pad')
4 print(data_new2)
```

| | time | Ha Noi | Vinh | Da Nang | Nha Trang | Ho Chi Minh | Ca Mau |
|----|--------------|--------|-------|---------|-----------|-------------|--------|
| 0 | 00 15-9-2019 | 25.65 | 24.79 | 24.01 | 25.06 | 25.48 | 24.97 |
| 1 | 01 15-9-2019 | 25.65 | 24.21 | 24.02 | 24.93 | 25.16 | 24.83 |
| 2 | 02 15-9-2019 | 25.05 | 23.73 | 23.89 | 24.79 | 24.80 | 24.55 |
| 3 | 03 15-9-2019 | 24.79 | 23.36 | 23.83 | 24.79 | 24.74 | 24.48 |
| 4 | 04 15-9-2019 | 24.59 | 23.05 | 23.69 | 24.82 | 24.80 | 24.38 |
| 5 | 05 15-9-2019 | 24.40 | 23.05 | 23.52 | 24.79 | 24.87 | 24.40 |
| 6 | 06 15-9-2019 | 24.38 | 22.79 | 23.68 | 25.10 | 24.71 | 24.41 |
| 7 | 07 15-9-2019 | 26.72 | 25.61 | 24.92 | 26.56 | 25.03 | 24.91 |
| 8 | 08 15-9-2019 | 28.84 | 26.93 | 26.51 | 26.53 | 25.75 | 25.85 |
| 9 | 09 15-9-2019 | 30.29 | 28.72 | 27.48 | 26.95 | 26.64 | 26.79 |
| 10 | 10 15-9-2019 | 30.29 | 29.97 | 27.48 | 26.95 | 27.68 | 27.53 |
| 11 | 11 15-9-2019 | 32.05 | 28.93 | 26.86 | 27.38 | 28.43 | 28.98 |
| 12 | 12 15-9-2019 | 31.31 | 28.94 | 26.65 | 27.47 | 28.29 | 29.24 |
| 13 | 13 15-9-2019 | 30.95 | 28.94 | 27.83 | 27.44 | 28.00 | 30.66 |
| 14 | 14 15-9-2019 | 30.56 | 30.62 | 26.49 | 27.16 | 27.67 | 30.97 |
| 15 | 15 15-9-2019 | 31.13 | 30.58 | 26.29 | 26.68 | 27.29 | 30.59 |
| 16 | 16 15-9-2019 | 30.80 | 30.20 | 26.29 | 26.45 | 27.29 | 29.13 |

2) Thay thế các missing (imputation)

df.fillna(method='bfill') → thay thế bằng giá trị liền sau

```
1 #PHƯƠNG PHÁP 2: Thay thế (Imputation)
2 #2.3)Thay thế các dữ liệu mất mát bằng giá trị liền sau của nó
3 data_new3 = data_temp.fillna(method='bfill')
4 print(data_new3)
```

| | | time | Ha Noi | Vinh | Da Nang | Nha Trang | Ho Chi Minh | Ca Mau |
|----|----|-----------|--------|-------|---------|-----------|-------------|--------|
| 0 | 00 | 15-9-2019 | 25.65 | 24.79 | 24.01 | 25.06 | 25.48 | 24.97 |
| 1 | 01 | 15-9-2019 | 25.05 | 24.21 | 24.02 | 24.93 | 25.16 | 24.83 |
| 2 | 02 | 15-9-2019 | 25.05 | 23.73 | 23.89 | 24.79 | 24.80 | 24.55 |
| 3 | 03 | 15-9-2019 | 24.79 | 23.36 | 23.83 | 24.82 | 24.74 | 24.48 |
| 4 | 04 | 15-9-2019 | 24.59 | 23.05 | 23.69 | 24.82 | 24.80 | 24.38 |
| 5 | 05 | 15-9-2019 | 24.40 | 22.79 | 23.52 | 24.79 | 24.87 | 24.40 |
| 6 | 06 | 15-9-2019 | 24.38 | 22.79 | 23.68 | 25.10 | 24.71 | 24.41 |
| 7 | 07 | 15-9-2019 | 26.72 | 25.61 | 24.92 | 26.56 | 25.03 | 24.91 |
| 8 | 08 | 15-9-2019 | 28.84 | 26.93 | 26.51 | 26.53 | 25.75 | 25.85 |
| 9 | 09 | 15-9-2019 | 30.29 | 28.72 | 27.48 | 26.95 | 26.64 | 26.79 |
| 10 | 10 | 15-9-2019 | 32.05 | 29.97 | 26.86 | 27.38 | 27.68 | 27.53 |
| 11 | 11 | 15-9-2019 | 32.05 | 28.93 | 26.86 | 27.38 | 28.43 | 28.98 |
| 12 | 12 | 15-9-2019 | 31.31 | 28.94 | 26.65 | 27.47 | 28.29 | 29.24 |
| 13 | 13 | 15-9-2019 | 30.95 | 30.62 | 27.83 | 27.44 | 28.00 | 30.66 |
| 14 | 14 | 15-9-2019 | 30.56 | 30.62 | 26.49 | 27.16 | 27.67 | 30.97 |
| 15 | 15 | 15-9-2019 | 31.13 | 30.58 | 26.29 | 26.68 | 27.29 | 30.59 |
| 16 | 16 | 15-9-2019 | 30.80 | 30.20 | 25.80 | 26.45 | 27.29 | 29.13 |

2) Thay thế các missing (imputation)

df.interpolate() → thay thế giá trị bằng nội suy

```

1  #PHƯƠNG PHÁP 2: Thay thế (Imputation)
2  #2.4)Xử lý các giá trị missing theo phương pháp nội suy
3  #Sử dụng hàm interpolate để thay thế giá trị missing với tham số:
4  #Thuật toán nội suy: Tuyến tính (linear)
5  #Hướng nội suy: Tiến lên (forward)
6  data_new4 = data_temp.interpolate(method='linear', limit_direction='forward')
7  print(data_new4)

```

| | | time | Ha Noi | Vinh | Da Nang | Nha Trang | Ho Chi Minh | Ca Mau |
|----|----|-----------|--------|-------|---------|-----------|-------------|--------|
| 0 | 00 | 15-9-2019 | 25.65 | 24.79 | 24.010 | 25.060 | 25.48 | 24.97 |
| 1 | 01 | 15-9-2019 | 25.35 | 24.21 | 24.020 | 24.930 | 25.16 | 24.83 |
| 2 | 02 | 15-9-2019 | 25.05 | 23.73 | 23.890 | 24.790 | 24.80 | 24.55 |
| 3 | 03 | 15-9-2019 | 24.79 | 23.36 | 23.830 | 24.805 | 24.74 | 24.48 |
| 4 | 04 | 15-9-2019 | 24.59 | 23.05 | 23.690 | 24.820 | 24.80 | 24.38 |
| 5 | 05 | 15-9-2019 | 24.40 | 22.92 | 23.520 | 24.790 | 24.87 | 24.40 |
| 6 | 06 | 15-9-2019 | 24.38 | 22.79 | 23.680 | 25.100 | 24.71 | 24.41 |
| 7 | 07 | 15-9-2019 | 26.72 | 25.61 | 24.920 | 26.560 | 25.03 | 24.91 |
| 8 | 08 | 15-9-2019 | 28.84 | 26.93 | 26.510 | 26.530 | 25.75 | 25.85 |
| 9 | 09 | 15-9-2019 | 30.29 | 28.72 | 27.480 | 26.950 | 26.64 | 26.79 |
| 10 | 10 | 15-9-2019 | 31.17 | 29.97 | 27.170 | 27.165 | 27.68 | 27.53 |
| 11 | 11 | 15-9-2019 | 32.05 | 28.93 | 26.860 | 27.380 | 28.43 | 28.98 |
| 12 | 12 | 15-9-2019 | 31.31 | 28.94 | 26.650 | 27.470 | 28.29 | 29.24 |
| 13 | 13 | 15-9-2019 | 30.95 | 29.78 | 27.830 | 27.440 | 28.00 | 30.66 |
| 14 | 14 | 15-9-2019 | 30.56 | 30.62 | 26.490 | 27.160 | 27.67 | 30.97 |
| 15 | 15 | 15-9-2019 | 31.13 | 30.58 | 26.290 | 26.680 | 27.29 | 30.59 |
| 16 | 16 | 15-9-2019 | 30.80 | 30.20 | 26.045 | 26.450 | 27.29 | 29.13 |

6. Chuyển đổi dữ liệu từ chuỗi sang số (labelEncoder) với pandas

original dataset

| x ₁ | x ₂ | y |
|----------------|----------------|---------|
| 5 | 8 | calabar |
| 9 | 3 | uyo |
| 8 | 6 | owerri |
| 0 | 5 | uyo |
| 2 | 3 | calabar |
| 0 | 8 | calabar |
| 1 | 8 | owerri |

LabelEncoder



```
{  
  "calabar" ---> 0  
  "owerri" ---> 1  
  "uyo"      ---> 2  
}
```

dataset with encoded labels

| x ₁ | x ₂ | y |
|----------------|----------------|---|
| 5 | 8 | 0 |
| 9 | 3 | 2 |
| 8 | 6 | 1 |
| 0 | 5 | 2 |
| 2 | 3 | 0 |
| 0 | 8 | 0 |
| 1 | 8 | 1 |

6) LabelEncoder

Các model chỉ thực hiện trên dữ liệu dạng số. Do đó chúng ta cần phải chuyển đổi các nhãn sang số.

| | Survived | Pclass | Sex | Age | Fare | Embarked | Title | IsAlone |
|---|----------|--------|--------|------|---------|----------|--------|---------|
| 0 | 0 | 3 | male | 22.0 | 7.2500 | S | Mr | 0 |
| 1 | 1 | 1 | female | 38.0 | 71.2833 | C | Mrs | 0 |
| 2 | 1 | 3 | female | 26.0 | 7.9250 | S | Miss | 1 |
| 3 | 1 | 1 | female | 35.0 | 53.1000 | S | Mrs | 0 |
| 4 | 0 | 3 | male | 35.0 | 8.0500 | S | Mr | 1 |
| 5 | 0 | 3 | male | NaN | 8.4583 | Q | Mr | 1 |
| 6 | 0 | 1 | male | 54.0 | 51.8625 | S | Mr | 1 |
| 7 | 0 | 3 | male | 2.0 | 21.0750 | S | Master | 0 |
| 8 | 1 | 3 | female | 27.0 | 11.1333 | S | Mrs | 0 |
| 9 | 1 | 2 | female | 14.0 | 30.0708 | C | Mrs | 0 |

6) LabelEncoder

Sử dụng pandas.series.map()

```
1 #Chuyển đổi thuộc tính Sex về dạng số nguyên (int)
2 # trong đó: Female = 1; Male = 0
3 data_label['Sex'] = data_label['Sex'].map( {'female': 1, 'male': 0} ).astype(int)
```

```
1 #Hiển thị dữ liệu 10 mẫu đầu tiên sau khi đã chuyển đổi.
2 data_label.head(10)
```

| | Survived | Pclass | Sex | Age | Fare | Embarked | Title | IsAlone |
|---|----------|--------|-----|-----|---------|----------|--------|---------|
| 0 | 0 | 3 | 0 | 22 | 7.2500 | S | Mr | 0 |
| 1 | 1 | 1 | 1 | 38 | 71.2833 | C | Mrs | 0 |
| 2 | 1 | 3 | 1 | 26 | 7.9250 | S | Miss | 1 |
| 3 | 1 | 1 | 1 | 35 | 53.1000 | S | Mrs | 0 |
| 4 | 0 | 3 | 0 | 35 | 8.0500 | S | Mr | 1 |
| 5 | 0 | 1 | 0 | 54 | 51.8625 | S | Mr | 1 |
| 6 | 0 | 3 | 0 | 2 | 21.0750 | S | Master | 0 |
| 7 | 1 | 3 | 1 | 27 | 11.1333 | S | Mrs | 0 |
| 8 | 1 | 2 | 1 | 14 | 30.0708 | C | Mrs | 0 |
| 9 | 1 | 3 | 1 | 4 | 16.7000 | S | Miss | 0 |

6) LabelEncoder

Sử dụng pandas.series.map()

```
1 #Chuyển đổi thuộc tính Embarked về dạng số nguyên (int)
2 # Trong đó: S = 0, C = 1, Q = 2
3 data_label['Embarked'] = data_label['Embarked'].map( {"S": 0, "C": 1, "Q": 2} ).astype(int)
```

```
1 #Hiển thị dữ liệu 10 mẫu cuối cùng sau khi đã chuyển đổi.
2 data_label.tail(10)
```

| | Survived | Pclass | Sex | Age | Fare | Embarked | Title | IsAlone |
|----|----------|--------|-----|-----|----------|----------|--------|---------|
| 15 | 0 | 3 | 0 | 2 | 29.1250 | 2 | Master | 0 |
| 16 | 0 | 3 | 1 | 31 | 18.0000 | 0 | Mrs | 0 |
| 17 | 0 | 2 | 0 | 35 | 26.0000 | 0 | Mr | 1 |
| 18 | 1 | 2 | 0 | 34 | 13.0000 | 0 | Mr | 1 |
| 19 | 1 | 3 | 1 | 15 | 8.0292 | 2 | Miss | 1 |
| 20 | 1 | 1 | 0 | 28 | 35.5000 | 0 | Mr | 1 |
| 21 | 0 | 3 | 1 | 8 | 21.0750 | 0 | Miss | 0 |
| 22 | 1 | 3 | 1 | 38 | 31.3875 | 0 | Mrs | 0 |
| 23 | 0 | 1 | 0 | 19 | 263.0000 | 0 | Mr | 0 |
| 24 | 1 | 3 | 1 | 60 | 7.8792 | 2 | Miss | 1 |

6) LabelEncoder

Sử dụng pandas.series.map()

```
1 #Chuyển đổi dữ liệu thuộc tính Title:  
2 #chuyển sang dạng số, với các giá trị tương ứng (Mr=1, Miss=2, Mrs=3, Master=4, Rare=5)  
3 #Có thể sử dụng một biến kiểu Dictionary để chuyển đổi  
4 title_mapping = {"Mr": 1, "Miss": 2, "Mrs": 3, "Master": 4, "Rare": 5}  
5 data_label['Title'] = data_label['Title'].map(title_mapping).astype(int)
```

```
1 #Hiển thị dữ liệu 10 mẫu đầu tiên sau khi đã chuyển đổi.  
2 data_label.head(10)
```

| | Survived | Pclass | Sex | Age | Fare | Embarked | Title | IsAlone |
|---|----------|--------|-----|-----|---------|----------|-------|---------|
| 0 | 0 | 3 | 0 | 22 | 7.2500 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 38 | 71.2833 | 1 | 3 | 0 |
| 2 | 1 | 3 | 1 | 26 | 7.9250 | 0 | 2 | 1 |
| 3 | 1 | 1 | 1 | 35 | 53.1000 | 0 | 3 | 0 |
| 4 | 0 | 3 | 0 | 35 | 8.0500 | 0 | 1 | 1 |
| 5 | 0 | 1 | 0 | 54 | 51.8625 | 0 | 1 | 1 |
| 6 | 0 | 3 | 0 | 2 | 21.0750 | 0 | 4 | 0 |
| 7 | 1 | 3 | 1 | 27 | 11.1333 | 0 | 3 | 0 |
| 8 | 1 | 2 | 1 | 14 | 30.0708 | 1 | 3 | 0 |
| 9 | 1 | 3 | 1 | 4 | 16.7000 | 0 | 2 | 0 |

Tham khảo thêm ***sklearn.preprocessing.LabelEncoder()***

Thực hành

Bài 19: Làm việc với Pandas



Mô tả file dữ liệu: Bai19_Personal.csv

- File dữ liệu chứa thông tin của 300 bệnh nhân bị bệnh tim mạch

| | A | B | C | D | E | F | G | H | I |
|----|----|------|----------|------------------|---------|-------------|---------|-------------|--------|
| 1 | id | tuoi | gioitinh | loai | huyetap | cholesterol | nhiptim | thalassemia | Ketqua |
| 2 | 1 | 63 | Male | Typical angina | 145 | 233 | 150 | 6 | 0 |
| 3 | 2 | 67 | Male | Asymptomatic | 160 | 286 | 108 | 3 | 1 |
| 4 | 3 | 67 | Male | Asymptomatic | 120 | 229 | 129 | 7 | 1 |
| 5 | 4 | 37 | Male | Non-anginal pain | 130 | 250 | 187 | 3 | 0 |
| 6 | 5 | 41 | Female | Atypical angina | 130 | 204 | 172 | | 0 |
| 7 | 6 | 56 | Male | Atypical angina | 120 | 236 | 178 | 3 | 0 |
| 8 | 7 | 62 | Female | Asymptomatic | 140 | 268 | 160 | 3 | 1 |
| 9 | 8 | 57 | Female | Asymptomatic | 120 | 354 | 163 | 3 | 0 |
| 10 | 9 | 63 | Male | Asymptomatic | 130 | 254 | 147 | 7 | 1 |
| 11 | 10 | 53 | Male | Asymptomatic | 140 | 203 | 155 | 7 | 1 |
| 12 | 11 | 57 | Male | Asymptomatic | 140 | 192 | 148 | 6 | 0 |
| 13 | 12 | 56 | Female | Atypical angina | 140 | 294 | 153 | 3 | 0 |
| 14 | 13 | 56 | Male | Non-anginal pain | 130 | 256 | 142 | 6 | 1 |

Bai19_Personal

Ready

145%

Bài 19: Làm việc với Pandas



Chi tiết như sau:

- **Id:** Mã của bệnh nhân (số)
- **Tuoi:** Tuổi của bệnh nhân (số)
- **Gioitinh:** Giới tính của bệnh nhân (chuỗi: Male – Female)
- **Loại:** Cho biết loại triệu chứng đau ngực mà bệnh nhân này mắc phải, với 4 giá trị: (Typical angina, Atypical angina, Non-anginal pain, Asymptomatic)
- **Huyetap:** Huyết áp của bệnh nhân – đơn vị: mmhg (số)
- **Cholesterol:** Chỉ số cholesterol của bệnh nhân – đơn vị: mg/dl (số)
- **Nhiptim:** Thông số nhịp tim của bệnh nhân – đơn vị: lần/phút (số)
- **Thalassemia:** Chỉ số Thalassemia của bệnh nhân chỉ gồm 3 giá trị (3: Bình thường | 4: Khiếm khuyết cố định | 7: Khiếm khuyết có thể đảo ngược)
- **Ketqua:** Cho biết bệnh nhân có bị bệnh tim hay không? (0: Không bị bệnh tim mạch | 1: Bị bệnh tim mạch)

Bài 19: Làm việc với Pandas



Yêu cầu 1:

- **Đọc dữ liệu từ file .csv vào biến kiểu dataframe**
- **Hiển thị thông tin của 20 bệnh nhân đầu tiên và 30 bệnh nhân cuối cùng của tập dữ liệu.**
- **Sử dụng phương thức .describe cho biết:**
 - Tuổi trung bình của các bệnh nhân trong tập dữ liệu
 - Tuổi của bệnh nhân trẻ nhất
 - Tuổi của bệnh nhân già nhất
 - Bao nhiêu bệnh nhân nam (Male)



Bài 19: Làm việc với Pandas



Yêu cầu 2:

- Cho biết những cột nào trong dữ liệu có chứa missing data và số lượng missing là bao nhiêu.
- Liệt kê danh sách các bệnh nhân bị missing dữ liệu cột 'loai', cột 'thalassemia'

| | id | tuoi | gioitinh | loai | huyetap | cholesterol | nhiptim | thalassemia | Ketqua |
|-----|-----|------|----------|------|---------|-------------|---------|-------------|--------|
| 205 | 206 | 58 | Male | NaN | 128 | 259 | 130 | 7.0 | 1 |
| 218 | 219 | 59 | Male | NaN | 138 | 271 | 182 | 3.0 | 0 |
| 250 | 251 | 58 | Male | NaN | 146 | 218 | 105 | 7.0 | 1 |
| 270 | 271 | 66 | Male | NaN | 160 | 228 | 138 | 6.0 | 0 |
| 292 | 293 | 63 | Male | NaN | 140 | 187 | 144 | 7.0 | 1 |

| | id | tuoi | gioitinh | loai | huyetap | cholesterol | nhiptim | thalassemia | Ketqua |
|-----|-----|------|----------|-----------------|---------|-------------|---------|-------------|--------|
| 4 | 5 | 41 | Female | Atypical angina | 130 | 204 | 172 | NaN | 0 |
| 20 | 21 | 64 | Male | Typical angina | 110 | 211 | 144 | NaN | 0 |
| 35 | 36 | 42 | Male | Asymptomatic | 140 | 226 | 178 | NaN | 0 |
| 240 | 241 | 41 | Female | Atypical angina | 126 | 306 | 163 | NaN | 0 |
| 265 | 266 | 52 | Male | Asymptomatic | 128 | 204 | 156 | NaN | 1 |
| 277 | 278 | 57 | Male | Atypical angina | 154 | 232 | 164 | NaN | 1 |
| 293 | 294 | 63 | Female | Asymptomatic | 124 | 197 | 136 | NaN | 1 |

Bài 19: Làm việc với Pandas



Yêu cầu 3:

- Xử lý dữ liệu missing ở cột '**loai**' bằng cách thay thế các giá trị missing bằng giá trị cố định là một chuỗi: 'Asymptomatic'
- Xử lý dữ liệu missing ở cột '**thalassemia**' bằng cách thay thế các giá trị missing bằng số 3

Yêu cầu 4:

- Chuyển đổi dữ liệu chuỗi (label) ở 2 cột **gioitinh** và **loai** sang dạng số

Yêu cầu 5:

- Lưu dataframe sau khi xử lý ở trên ra file:
Bai19_personal_finish.csv