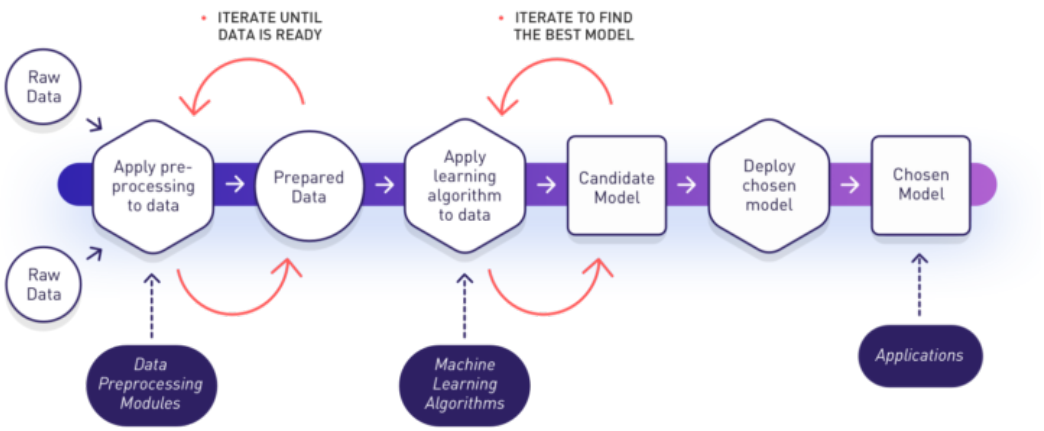
# Đọc và chuyển đổi dữ liệu

## 1 Đọc dữ liệu

### 1.1 Giới thiệu

Trong phần này, chúng ta sẽ học về cách xử lý dữ liệu thô (data preprocessing). Đây là những kỹ thuật dùng để chuyển đổi dữ liệu thô thành dạng hữu ích và dễ phân tích hơn.



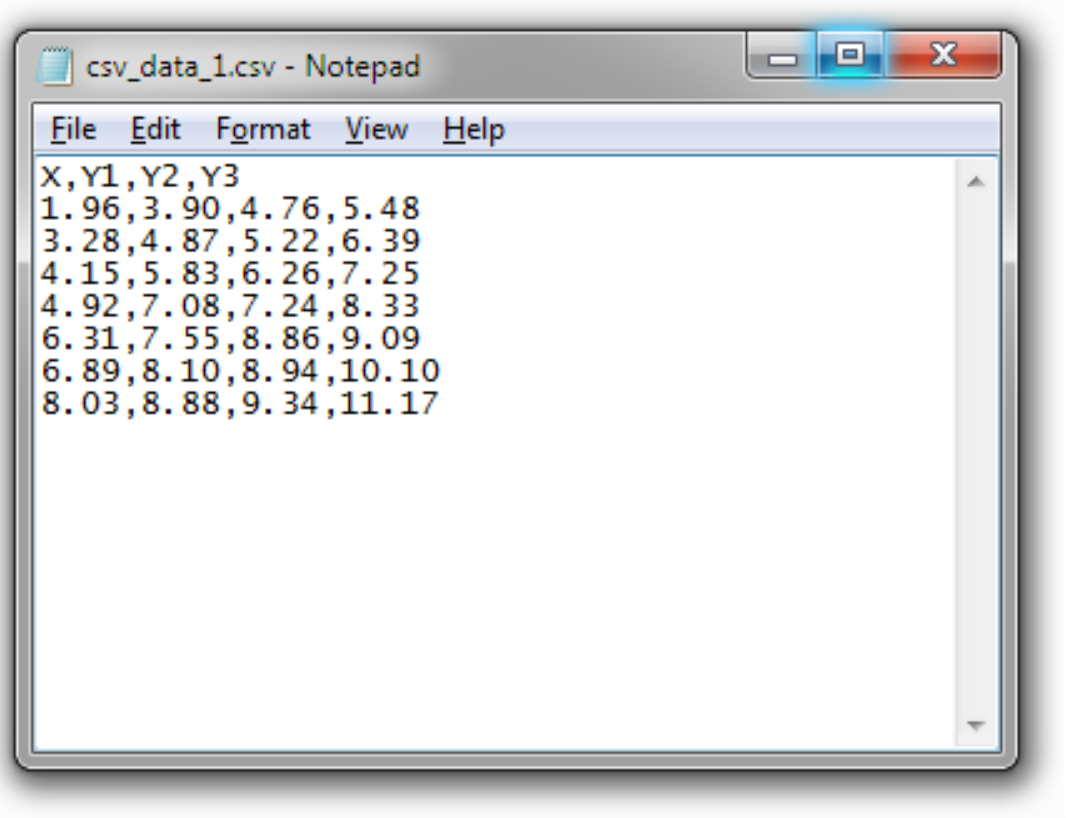

Một số kỹ thuật thường được sử dụng:

- Làm sạch dữ liệu (data cleaning):
  - Dữ liệu bị thiếu (missing data).
  - Dữ liệu nhiễu (noisy data).
- Chuyển đổi dữ liệu (data transformation):
  - Chuẩn hóa dữ liệu (normalization).
  - Rời rạc hóa dữ liệu (discretization).
- Thu gọn dữ liệu (data reduction).

### 1.2 Định dạng `csv`

Trước hết, cần phải đọc dữ liệu thô được lưu trữ và một dạng lưu trữ dữ liệu phổ biến là `csv` (**c**omma **s**eparated **v**alues, dữ liệu phân cách bằng dấu phẩy). Đây là một loại định dạng văn bản đơn giản mà trong đó, các giá trị được ngăn cách với nhau bằng dấu

phẩy. Định dạng CSV thường xuyên được sử dụng để lưu các bảng tính quy mô nhỏ như danh bạ, danh sách lớp, báo cáo.... Thông thường, một file `csv` có đuôi là `.csv`.




## 1.3 Đọc file `csv` thành `DataFrame`

Để đọc file `csv` thành `DataFrame`, có thể dùng `pandas.read_csv()` với cú pháp:

`pandas.read_csv(<đường_dẫn_đến_file>, [các_tham_số_khác])`

Trong đó, `đường_dẫn_đến_file` có thể là đường dẫn một tập tin trong máy (local) hoặc là một url (remote).

[1]:
```
# ví dụ
import pandas as pd
df = pd.read_csv(
    'https://people.math.sc.edu/Burkardt/datasets/csv/addresses.csv'
)
df
```

[1]:
```
              FirstName  "LastName"                  "StreetAddress"  \
0                  John         Doe                   120 jefferson st.
1                  Jack    McGinnis                        220 hobo Av.
2          John "Da Man"      Repici                   120 Jefferson St.
3                Stephen       Tyler  7452 Terrace "At the Plaza" road
4                   NaN    Blankman                                NaN
5  Joan "the bone", Anne         Jet                9th, at Terrace plc
```

```
        "City"  "State"   "ZipCode"
0    Riverside       NJ        8075
1        Phila       PA        9119
2    Riverside       NJ        8075
3     SomeTown       SD       91234
4     SomeTown       SD         298
5  Desert City       CO         123
```

## 1.4 Một số tham số của `pd.read_csv()`

Trong các ví dụ dưới đây, có thể mở file csv bằng trình duyệt để xem cấu trúc file trước khi đọc để hiểu rõ hơn cách hoạt động của các tham số.

### 1.4.1 Tham số `sep`

Tham số này dùng để chỉ ra cách các dữ liệu được phân tách như thế nào. Mặc định là dấu phẩy `','`.

[2]:
```
# ví dụ, không có sep
df1 = pd.read_csv(
    'https://raw.githubusercontent.com/Levytan/MIS.2019/master/Data/
  ↪sample_data_1.csv'
)
df1.head()
```

[2]:
```
  Region;Age;Income;Online Shopper
0                India;49;86400;No
1               Brazil;32;57600;Yes
2                  USA;35;64800;No
3               Brazil;43;73200;No
4                     USA;45;;Yes
```

[3]:
```
# thêm sep = ';'
df2 = pd.read_csv(
    'https://raw.githubusercontent.com/Levytan/MIS.2019/master/Data/
  ↪sample_data_1.csv',
    sep = ';'
)
df2.head()
```

[3]:
```
    Region   Age   Income Online Shopper
0    India  49.0  86400.0             No
1   Brazil  32.0  57600.0            Yes
2      USA  35.0  64800.0             No
3   Brazil  43.0  73200.0             No
4      USA  45.0      NaN            Yes
```

### 1.4.2 Tham số `header`

Tham số này được dùng để chỉ ra những dòng được dùng để làm header (tên cột). Giá trị mặc định là `'infer'` (tự suy ra từ file, thường là lấy dòng đầu tiên làm tên cột).

[4]:
```
# ví dụ, dùng None để báo rằng file .csv không có header
df3 = pd.read_csv(
    'https://raw.githubusercontent.com/Levytan/MIS.2019/master/Data/
  ↪sample_data_2.csv',
    header = None
)
df3.head()
```

[4]:

| | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | India | 49.0 | 86400.0 | No |
| 1 | Brazil | 32.0 | 57600.0 | Yes |
| 2 | USA | 35.0 | 64800.0 | No |
| 3 | Brazil | 43.0 | 73200.0 | No |
| 4 | USA | 45.0 | NaN | Yes |

### 1.4.3 Tham số `index_col`

Tham số này dùng để chỉ ra cột (một hoặc nhiều) dùng làm index. Giá trị mặc định là `None`, nghĩa là không dùng cột nào.

[5]:
```
# ví dụ,
df4 = pd.read_csv(
    'https://raw.githubusercontent.com/Levytan/MIS.2019/master/Data/
  ↪sample_data_3.csv',
    index_col = 'ID'
)
df4
```

[5]:

| ID | Method | a1 | a2 | a3 | a4 | a5 | a6 | a7 | a8 | a9 | a10 | a11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Soil | 2 | 3 | 4 | 5 | 0 | 3 | 6 | 4 | 8 | 0 | 0 |
| 2 | Soil | 6 | 9 | 2 | 7 | 0 | 4 | 3 | 4 | 4 | 0 | 0 |
| 3 | Soil | 5 | 2 | 4 | 9 | 0 | 1 | 1 | 2 | 3 | 0 | 0 |
| 1 | Soil | 5 | 0 | 0 | 0 | 4 | 7 | 8 | 2 | 1 | 3 | 4 |
| 2 | Soil | 4 | 0 | 0 | 0 | 6 | 3 | 8 | 7 | 2 | 2 | 1 |

### 1.4.4 Các tham số khác

Có thể đọc về các tham số khác của `pandas.read_csv()` tại đây.

# 2 Chuyển đổi kiểu dữ liệu

Do file csv chỉ lưu trữ dữ liệu dưới dạng văn bản, nên một số kiểu dữ liệu (thường là kiểu thời gian) khi đọc lên chỉ được hiểu như chuỗi. Vì thế, cần phải thực hiện việc chuyển đổi những chuỗi này sang kiểu phù hợp để có thể thực hiện những tính toán cần thiết.

Vài dạng chuyển đổi thường gặp là: - Chuyển đổi chuỗi ngày tháng thành thời gian. - Chuyển đổi chuỗi số thành số. - Chuyển đổi chuỗi thành phân loại.

[6]:
```
# ví dụ
vnm = pd.read_csv(
    'https://raw.githubusercontent.com/Levytan/MIS.2019/master/Data/
  ↪vnm.csv',
    header = None,
    names = ['Date', 'Close', 'Volume', 'PercentChange']
)
vnm.head()
```

[6]:

| | Date | Close | Volume | PercentChange |
|---|---|---|---|---|
| 0 | 2018-06-01 | 136.1k | 1051610 | ?% |
| 1 | 2018-06-04 | 141.6k | 965230 | 4.04% |
| 2 | 2018-06-05 | 144.4k | 879160 | 1.98% |
| 3 | 2018-06-06 | 142.8k | 463720 | -1.11% |
| 4 | 2018-06-07 | 144.7k | 401900 | 1.33% |

[7]:
```
# kiểu của các cột
vnm.dtypes
```

[7]:
```
Date             object
Close            object
Volume            int64
PercentChange    object
dtype: object
```

Trong ví dụ trên, có thể thấy cột **Date** không có kiểu ngày tháng.

## 2.1 Chuyển đổi chuỗi ngày tháng thành ngày tháng

Nhắc lại, có thể dùng `pd.to_datetime()` để chuyển đổi một danh sách chuỗi ngày tháng thành danh sách ngày tháng. Kết hợp với phương thức `.assign()` là hoàn toàn có thể thực hiện việc chuyển đổi như ví dụ

[8]:
```
# ví dụ
vnm = vnm.assign(
    Date = pd.to_datetime(vnm.Date)
)
print(vnm.Date.dtype)
```

```
datetime64[ns]
```

Lưu ý, `pd.to_datetime()` có một tham số là `errors` để chỉ cách xử lý trong trường hợp không chuyển đổi được giá trị nào đó. Tham số này gồm các giá trị: -

- `'raise'`: báo lỗi (mặc định).
- `'ignore'`: giữ nguyên giá trị lỗi.
- `'coerce'`: giá trị lỗi chuyển thành `NaT`.

Các tham số khác của `pd.to_datetime()` có thể xem tại đây.

## 2.2 Chuyển đổi chuỗi số thành số

Để thực hiện việc này, `pandas` cung cấp một hàm là `pd.to_numeric()`. Có thể xem một số ví dụ tại đây.

[9]:
```
# ví dụ, chuyển đổi cột PercentChange của vnm sang kiểu số
# bước 1, bỏ đi ký tự %
vnm = vnm.assign(
    PercentChange = vnm.PercentChange.str.replace('%', '')
)
# bước 2, chuyển đổi từ chuỗi số sang số
vnm = vnm.assign(
    PercentChange = pd.to_numeric(
        vnm.PercentChange,
        errors = 'coerce'
    )
)
# kiểm tra
print(vnm.PercentChange.dtype)
```

```
float64
```

## 2.3 Chuyển đổi chuỗi thành phân loại

### 2.3.1 Kiểu phân loại

Kiểu phân loại có đặc điểm:

- Có các giá trị cố định.
- Số lượng ít.
- Không hoặc gần như không thay đổi số lượng giá trị.

Ví dụ: giới tính, nhóm máu, xếp loại học tập, ...

### 2.3.2 Chuyển đổi

Để thực hiện việc chuyển đổi, có thể dùng `pd.Categorical()` với cú pháp:

```
pd.Categorial(
    <danh_sách_chuyển_đổi>,
    categories = <phân_loại_muốn_dùng>
```

```
)
```

[10]:
```
# ví dụ chuyển đổi, các giá trị không nằm trong phân loại sẽ thành NaN
pd.Categorical(
    ['a', 'b', 'd', 'a', 'd'],
    categories = ['a', 'b', 'c']
)
```

[10]:
```
['a', 'b', NaN, 'a', NaN]
Categories (3, object): ['a', 'b', 'c']
```