Bài giảng:

Lập trình Python

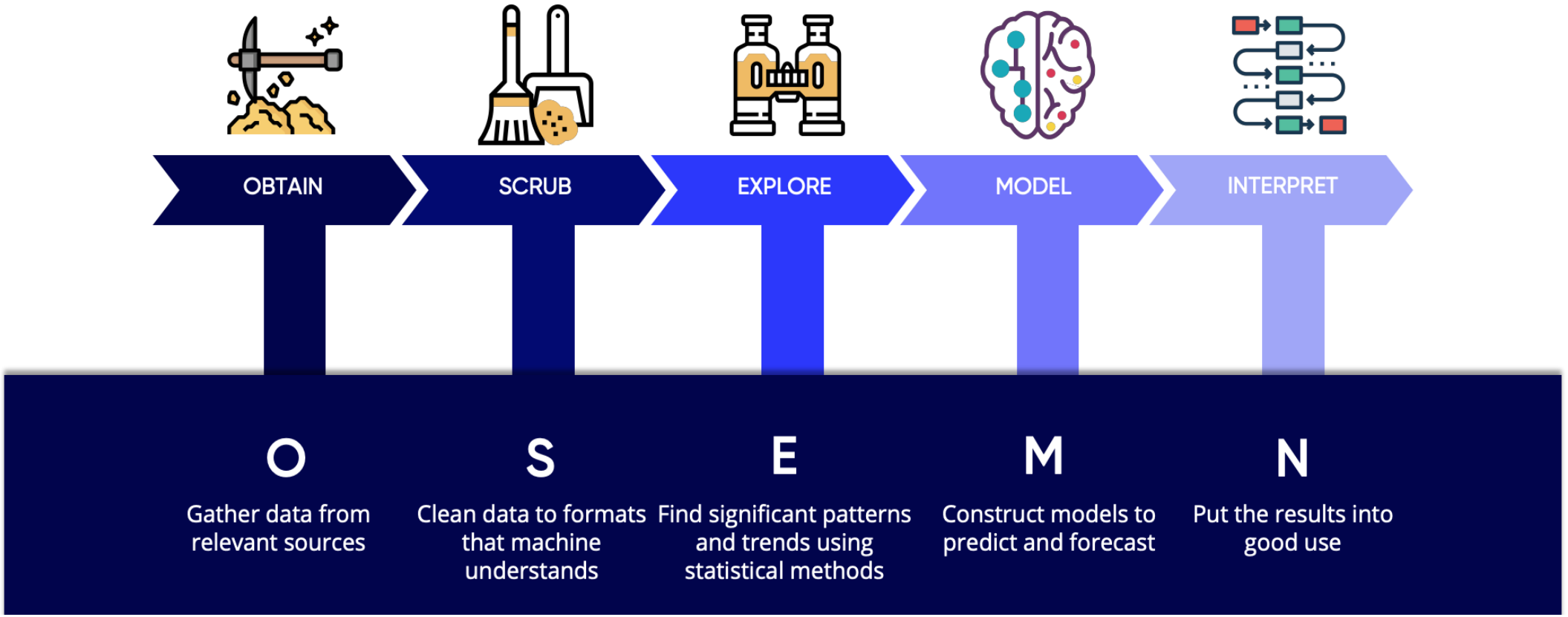
Bài 06:
Làm việc với các nguồn dữ liệu khác nhau

Nội dung bài 6

1. Giới thiệu
2. Đọc các file dữ liệu với Pandas:
   - Giới thiệu Pandas
   - Đọc dữ liệu từ file CSV, Excel, Json
3. Lấy dữ liệu từ API
4. Làm việc với dữ liệu từ MongoDB
5. Sử dụng sklearn đọc tập dữ liệu mẫu
6. Tạo Simulated Dataset với sklearn

1. Giới thiệu

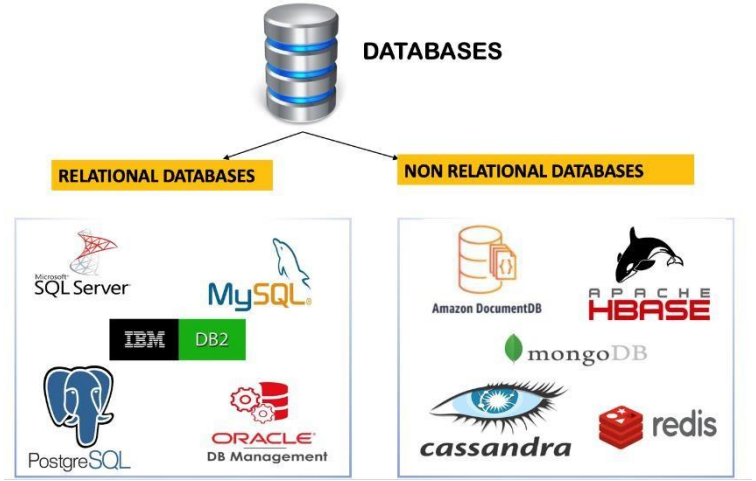
1. Giới thiệu

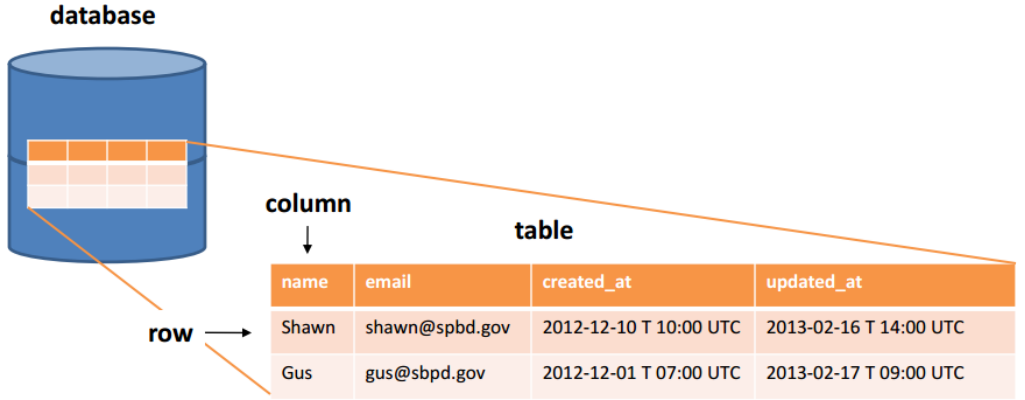


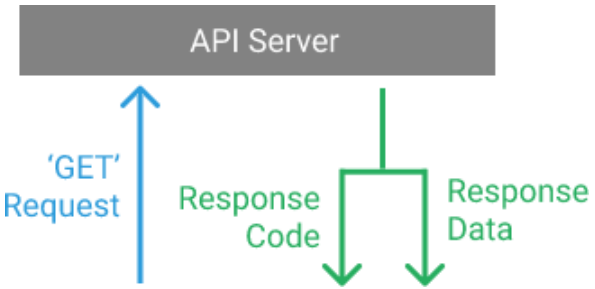
Để có thể phân tích dữ liệu cho bất kỳ mục đích gì, bước đầu tiên bao giờ cũngphải thu thập và đọc dữ liệu từ tất cả các nguồn, các dạng khác nhau

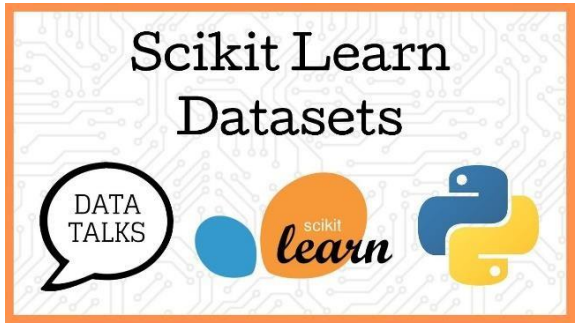
1. Giới thiệu

Dữ liệu tồn tại trong rất nhiều dạng khác nhau, từ dữ liệu trong các file cơ bản như Excel, CSV, text, Json, XML…hay lưu trữ trong các CSDL. Dữ liệu còn có thể được cung cấp thông qua các API.







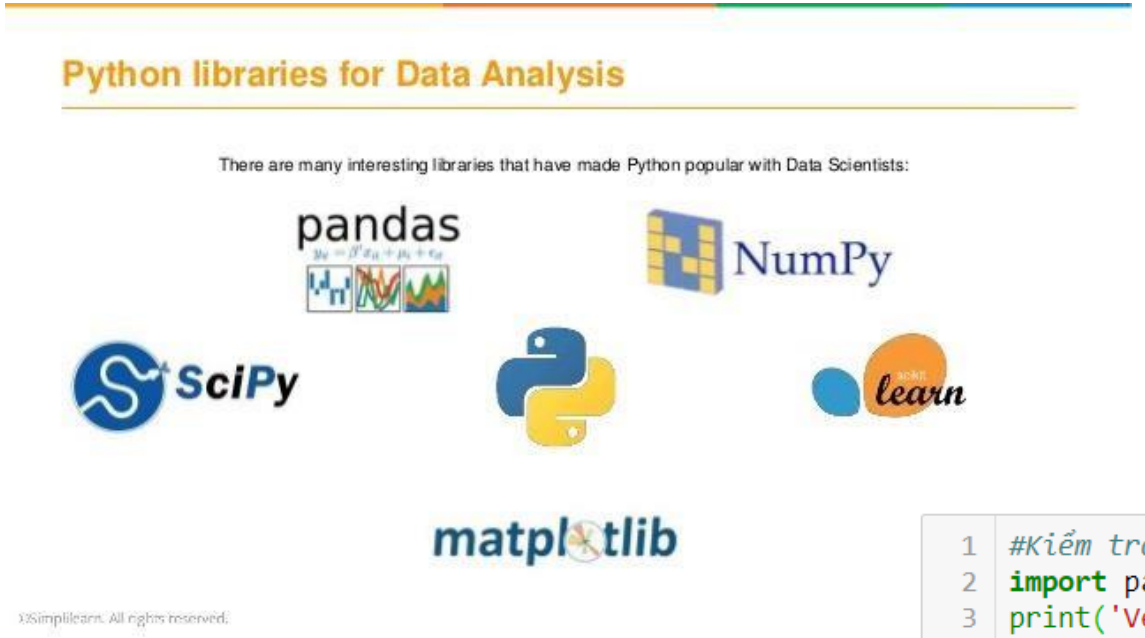


Trong bài học này chúng ta sẽ tìm hiểu cách đọc dữ liệu từ một số dạng cơ bản và phổ biến nhất.

2. Đọc dữ liệu với Pandas

2.1 Giới thiệu Pandas

- Pandas là một thư viện phần mềm viết cho ngôn ngữ lập trình Python. Nó được sử dụngcho thao tác và phân tích dữ liệu. Nó cung cấp cấu trúc dữ liệu đặc biệt và hoạt động chocác thao tác của các bảng số liệu và chuỗi thời gian. Pandas là thư viện miễn phí được phát hành theo giấy phép BSD.
- Python với Pandas được sử dụng trong nhiều lĩnh vực cả lĩnh vực học thuật và thương mại như: Tài chính, kinh tế, thống kê, phân tích, v.v.

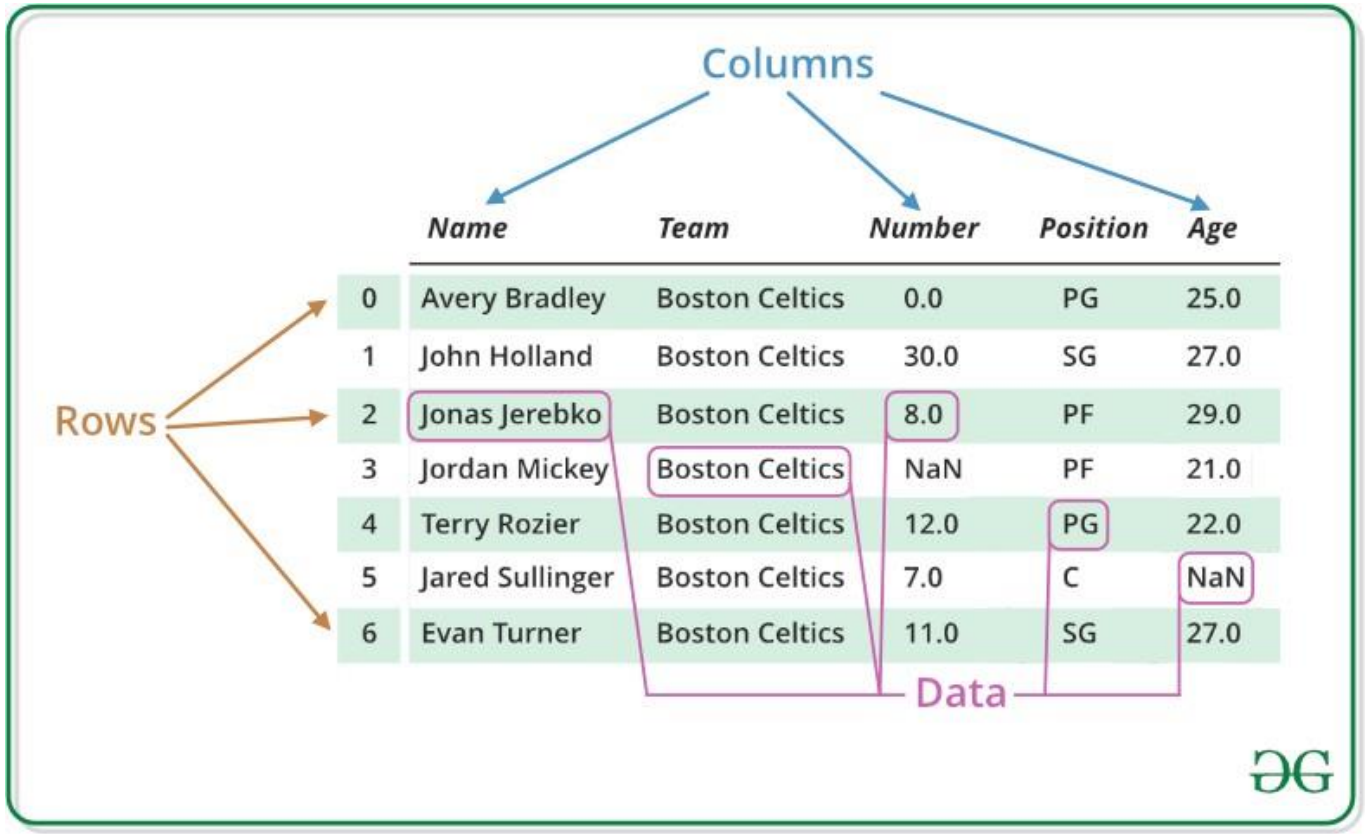



```
#Kiểm tra phiên bản của thư viện Pandas
import pandas as pd
print('Version Pandas: ',pd.__version__)
```

```
Version Pandas:  1.1.1
```

2.1 Giới thiệu Pandas

- Pandas có ba cấu trúc dữ liệu và nó được xây dựng dựa trên thư viện Numpy vậynên chúng hoạt động rất nhanh và hiệu quả: ***Series, DataFrame, Panel***.
- Trong ba kiểu dữ liệu, DataFrame là kiểu dữ liệu được sử dụng rộng rãi nhất

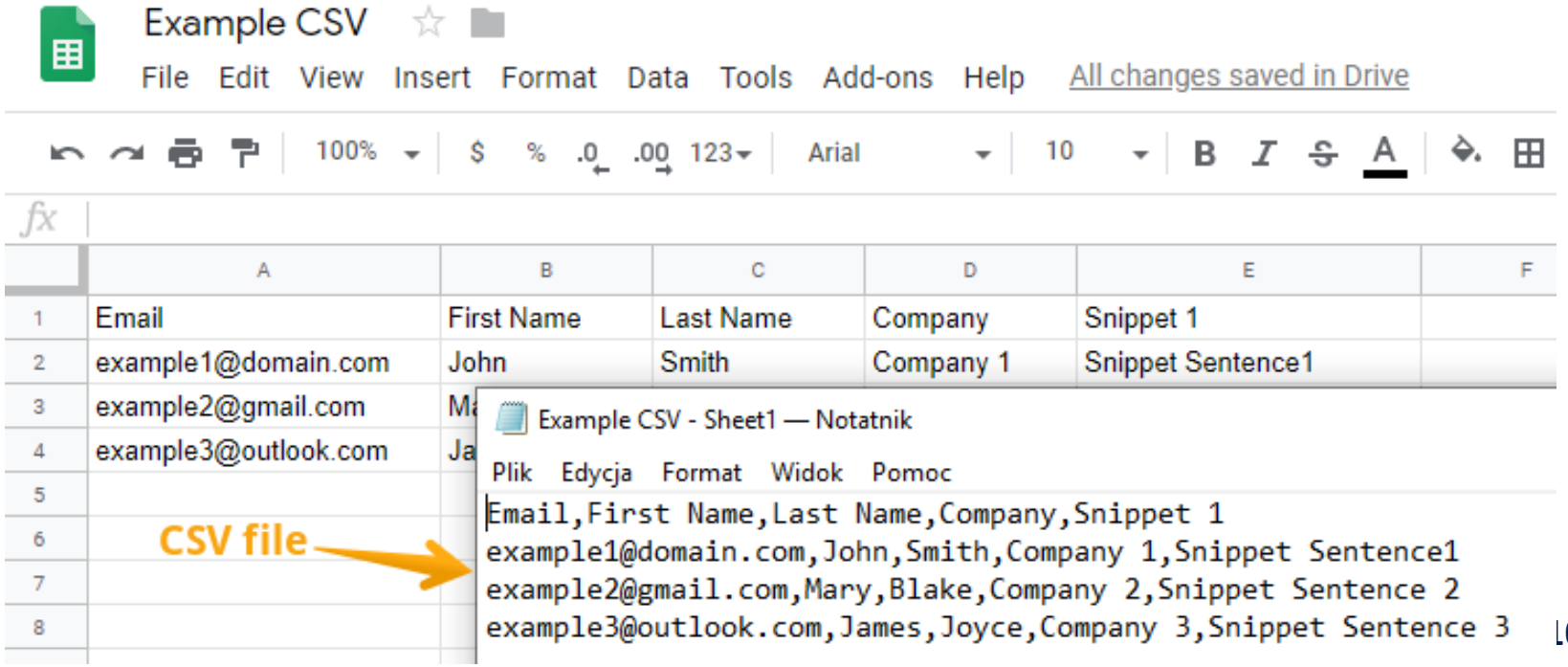


DataFrame: Cấu trúc dạng bảng 2D, kích thước có thể thay đổi được. Dữ liệu một cột là đồng nhất nhưng có thể không đồngnhất giữa các cột

2.2 Đọc dữ liệu từ file CSV

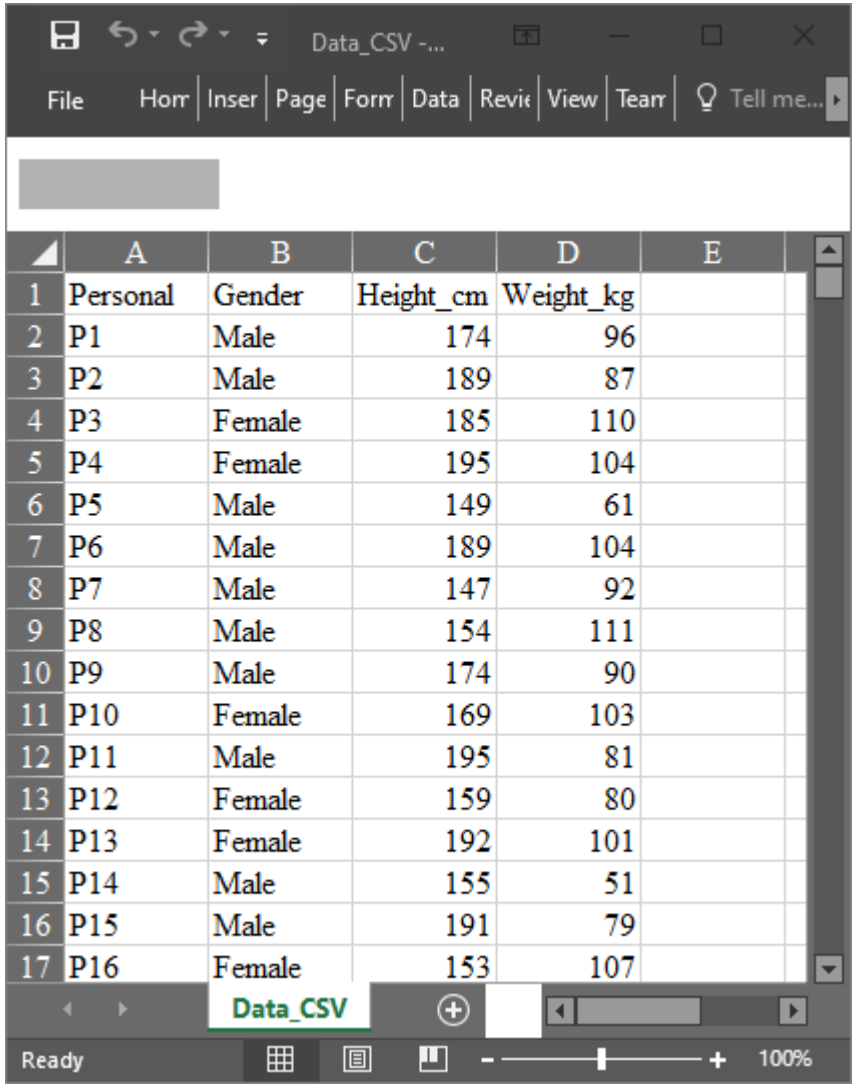
2.2 Đọc file CSV

- CSV là một định dạng dữ liệu văn bản đơn giản có tên đầy đủ là Comma SeparatedValues. Với định dạng CSV này, các giá trị được chia tách với nhau bởi các dấu phẩy. Định dạng CSV phổ biến bởi vì chúng có tính tương thích cao, dễ dàng dichuyển từ phần mềm này sang phần mềm khác để sử dụng mà không lo gặp các xung đột.
- Tài liệu CSV cũng làm một trong những tài liệu phổ biến trên thế giới với khả năng lưu trữ nhỏ nhẹ.



2.2 Đọc file CSV

Sử dụng phương thức read_csv() đọc dữ liệu từ file .CSV



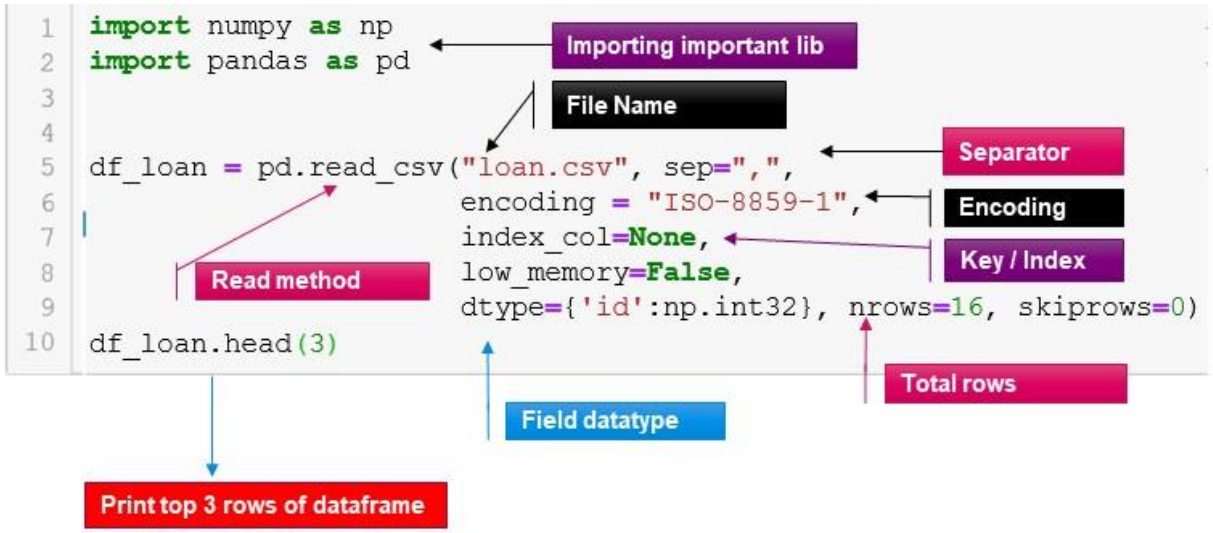
	A	B	C	D	E
1	Personal	Gender	Height_cm	Weight_kg	
2	P1	Male	174	96	
3	P2	Male	189	87	
4	P3	Female	185	110	
5	P4	Female	195	104	
6	P5	Male	149	61	
7	P6	Male	189	104	
8	P7	Male	147	92	
9	P8	Male	154	111	
10	P9	Male	174	90	
11	P10	Female	169	103	
12	P11	Male	195	81	
13	P12	Female	159	80	
14	P13	Female	192	101	
15	P14	Male	155	51	
16	P15	Male	191	79	
17	P16	Female	153	107	

```python
import pandas as pd
path = 'Data_Excersice\CSV\Data_CSV.csv'
#Sử dụng phương thức read_csv
data = pd.read_csv(path)
#Hiển thị thông tin biến Data
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Personal   500 non-null    object
 1   Gender     500 non-null    object
 2   Height_cm  500 non-null    int64
 3   Weight_kg  500 non-null    int64
dtypes: int64(2), object(2)
memory usage: 15.8+ KB
```

2.2 Đọc file CSV

Sử dụng phương thức read_csv() có rất nhiều tham số khác nhau để thiết lập cách thức đọc file .csv



ID	funding_round_type	funded_at	raised_amount_usd	name	category_list	status	country_code
1	VENTURE	1/8/2014	0	DERON	NAN	OPERATING	NaN
2	SEED	1/1/2015	18192	ASYS	CONSUMER ELECTRONICS	OPERATING	USA
3	GRANT	1/10/2013	14851	Â HIZMETLERI TIC	CONSUMER GOODS	OPERATING	NaN

```
Arguments

read_csv(filepath_or_buffer, sep=',',
         delimiter=None, header='infer',
         names=None, index_col=None, usecols=None,
         squeeze=False, prefix=None,
         mangle_dupe_cols=True, dtype=None,
         engine=None, converters=None,
         true_values=None, false_values=None,
         skipinitialspace=False, skiprows=None,
         nrows=None, na_values=None,
         keep_default_na=True, na_filter=True,
         verbose=False, skip_blank_lines=True,
         parse_dates=False,
         infer_datetime_format=False,
         keep_date_col=False, date_parser=None,
         dayfirst=False, iterator=False,
         chunksize=None, compression='infer',
         thousands=None, decimal='.',
         lineterminator=None, quotechar='"',
         quoting=0, escapechar=None, comment=None,
         encoding=None, dialect=None,
         tupleize_cols=False,
         error_bad_lines=True,
         warn_bad_lines=True, skipfooter=0,
         skip_footer=0, doublequote=True,
         delim_whitespace=False,
         as_recarray=False, compact_ints=False,
         use_unsigned=False, low_memory=True,
         buffer_lines=None, memory_map=False,
         float_precision=None)
```

https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.read_csv.html

2.2 Đọc file CSV

Vd1: sử dụng tham số index_col để thiết lập cột index khi đọc file csv

```
#Sử dụng phương thức read_csv()
#Tham số: Thiết lập cột index là cột Personal
data1 = pd.read_csv(path,
                    index_col=0)
data1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 500 entries, P1 to P500
Data columns (total 3 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Gender      500 non-null    object
 1   Height_cm   500 non-null    int64
 2   Weight_kg   500 non-null    int64
dtypes: int64(2), object(1)
memory usage: 15.6+ KB
```

```
#Hiển thị dữ liệu 5 dòng đầu tiên
data1.head()
```

Personal	Gender	Height_cm	Weight_kg
P1	Male	174	96
P2	Male	189	87
P3	Female	185	110
P4	Female	195	104
P5	Male	149	61

2.2 Đọc file CSV

Vd2: Thiết lập tham số chỉ đọc 100 dòng đầu tiên và dữ liệu trong 2 cột Height_cm, Weight_kg

```
#Sử dụng phương thức read_csv()
#Thiết lập số hàng, cột muốn đọc dữ liệu
data2 = pd.read_csv(path,
                    nrows=100,
                    usecols=['Height_cm', 'Weight_kg'])
data2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 2 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Height_cm  100 non-null    int64
 1   Weight_kg  100 non-null    int64
dtypes: int64(2)
memory usage: 1.7 KB
```

```
#Hiển thị dữ liệu 5 dòng đầu tiên
data2.head()
```

	Height_cm	Weight_kg
0	174	96
1	189	87
2	185	110
3	195	104
4	149	61

2.2 Đọc file CSV

Vd3: Thiết lập tham số đọc dữ liệu từ dòng thứ 5 trở đi, và đặt lại tên của từng cộtdữ liệu thành [‘ID’,’Sex’,’H(cm)’,W(kg)’]

```
#Thiết lập tham số đọc dữ liệu từ dòng thứ 5 trong file
#và đặt lại tên của các cột dữ liệu
data3 = pd.read_csv(path,
                    names=['ID','Sex','H(cm)','W(kg)'],
                    skiprows=5)
data3.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 496 entries, 0 to 495
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   ID      496 non-null    object
 1   Sex     496 non-null    object
 2   H(cm)   496 non-null    int64
 3   W(kg)   496 non-null    int64
dtypes: int64(2), object(2)
memory usage: 15.6+ KB
```

```
#Hiển thị 5 dòng dữ liệu đầu tiên
data3.head()
```

	ID	Sex	H(cm)	W(kg)
0	P5	Male	149	61
1	P6	Male	189	104
2	P7	Male	147	92
3	P8	Male	154	111
4	P9	Male	174	90

Thực hành 2.2

Thực hành 2.2

Yêu cầu 2.2.1: Sinh viên đọc dữ liệu dạng CSV lưu trong file csv_Data_Loan.csv với các tham số mặc định

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	loan_amnt	term	int_rate	emp_length	home_ownership	annual_inc	purpose	addr_state	dti	delinq_2yrs	revol_util	total_acc	bad_loan	longest_cre	verification_status
2	5000	36 months	10.65	10	RENT	24000	credit_card	AZ	27.65	0	83.7	9	0	26	verified
3	2500	60 months	15.27	0	RENT	30000	car	GA	1	0	9.4	4	1	12	verified
4	2400	36 months	15.96	10	RENT	12252	small_business	IL	8.72	0	98.5	10	0	10	not verified
5	10000	36 months	13.49	10	RENT	49200	other	CA	20	0	21	37	0	15	verified
6	5000	36 months	7.9	3	RENT	36000	wedding	AZ	11.2	0	28.3	12	0	7	verified
7	3000	36 months	18.64	9	RENT	48000	car	CA	5.35	0	87.5	4	0	4	verified
8	5600	60 months	21.28	4	OWN	40000	small_business	CA	5.55	0	32.6	13	1	7	verified
9	5375	60 months	12.69	0	RENT	15000	other	TX	18.08	0	36.5	3	1	7	verified
10	6500	60 months	14.65	5	OWN	72000	debt_consolidation	AZ	16.12	0	20.6	23	0	13	not verified
11	12000	36 months	12.69	10	OWN	75000	debt_consolidation	CA	10.78	0	67.1	34	0	22	verified
12	9000	36 months	13.49	0	RENT	30000	debt_consolidation	VA	10.08	0	91.7	9	1	7	verified
13	3000	36 months	9.91	3	RENT	15000	credit_card	IL	12.56	0	43.1	11	0	8	verified
14	10000	36 months	10.65	3	RENT	100000	other	CA	7.06	0	55.5	29	1	20	verified
15	1000	36 months	16.29	0	RENT	28000	debt_consolidation	MO	20.31	0	81.5	23	0	4	not verified
16	10000	36 months	15.27	4	RENT	42000	home_improvement	CA	18.6	0	70.2	28	0	13	not verified
17	3600	36 months	6.03	10	MORTGAGE	110000	major_purchase	CT	10.52	0	16	42	0	18	not verified
18	6000	36 months	11.71	1	MORTGAGE	84000	medical	UT	18.44	2	37.73	14	0	8	verified
19	9200	36 months	6.03	6	RENT	77385.19	debt_consolidation	CA	9.86	0	23.1	28	0	10	not verified
20	21000	36 months	12.42	10	RENT	105000	debt_consolidation	FL	13.22	0	90.3	38	1	28	verified
21	10000	36 months	11.71	10	OWN	50000	credit_card	TX	11.18	0	82.4	21	0	26	verified
22	10000	36 months	11.71	5	RENT	50000	debt_consolidation	CA	16.01	0	91.8	17	0	8	not verified
23	6000	36 months	11.71	1	RENT	76000	major_purchase	CA	2.4	0	29.7	7	1	10	not verified
24	15000	36 months	9.91	2	MORTGAGE	92000	credit_card	IL	29.44	0	93.9	31	0	9	verified

Data_Loan

Ready

100%

Thực hành 2.2

Yêu cầu 2.2.2: Đọc dữ liệu từ file Data_Loan.CSV vào 2 biến DataFrame tương ứng.

- **df_number:** Chỉ chứa các cột dữ liệu số
- **df_object:** Chỉ chứa các cột dữ liệu Object

```
#Hiển thị 5 dòng dữ liệu đầu tiên của biến df_number
df_number.head()
```

	loan_amnt	int_rate	emp_length	annual_inc	dti	delinq_2yrs	revol_util	total_acc	bad_loan	longest_credit_length
0	5000	10.65	10.0	24000.0	27.65	0.0	83.7	9.0	0	26.0
1	2500	15.27	0.0	30000.0	1.00	0.0	9.4	4.0	1	12.0
2	2400	15.96	10.0	12252.0	8.72	0.0	98.5	10.0		
3	10000	13.49	10.0	49200.0	20.00	0.0	21.0	37.0		
4	5000	7.90	3.0	36000.0	11.20	0.0	28.3	12.0		

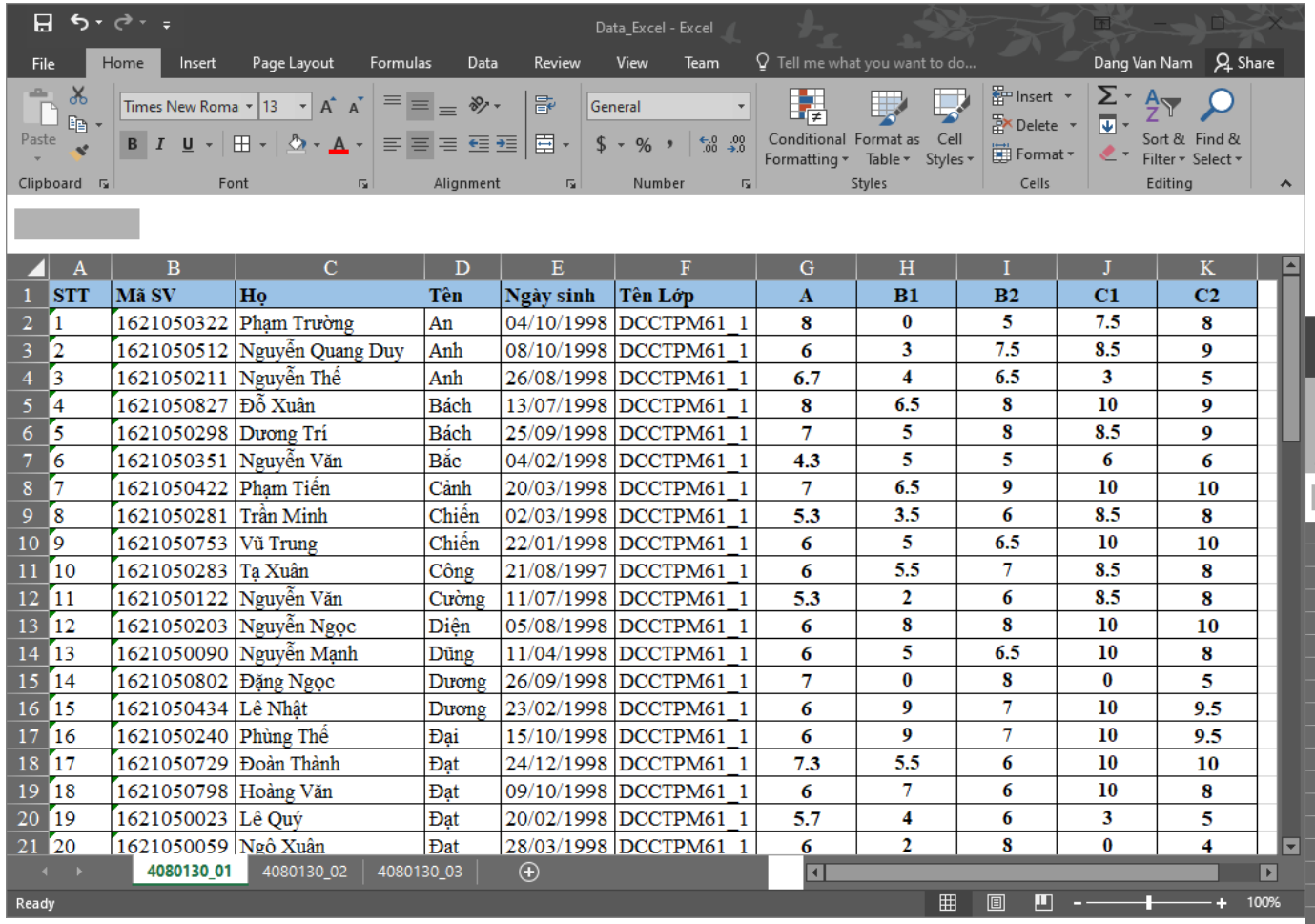
```
#Hiển thị 5 dòng dữ liệu đầu tiên của biến df_object
df_object.head()
```

	term	home_ownership	purpose	addr_state	verification_status
0	36 months	RENT	credit_card	AZ	verified
1	60 months	RENT	car	GA	verified
2	36 months	RENT	small_business	IL	not verified
3	36 months	RENT	other	CA	verified
4	36 months	RENT	wedding	AZ	verified

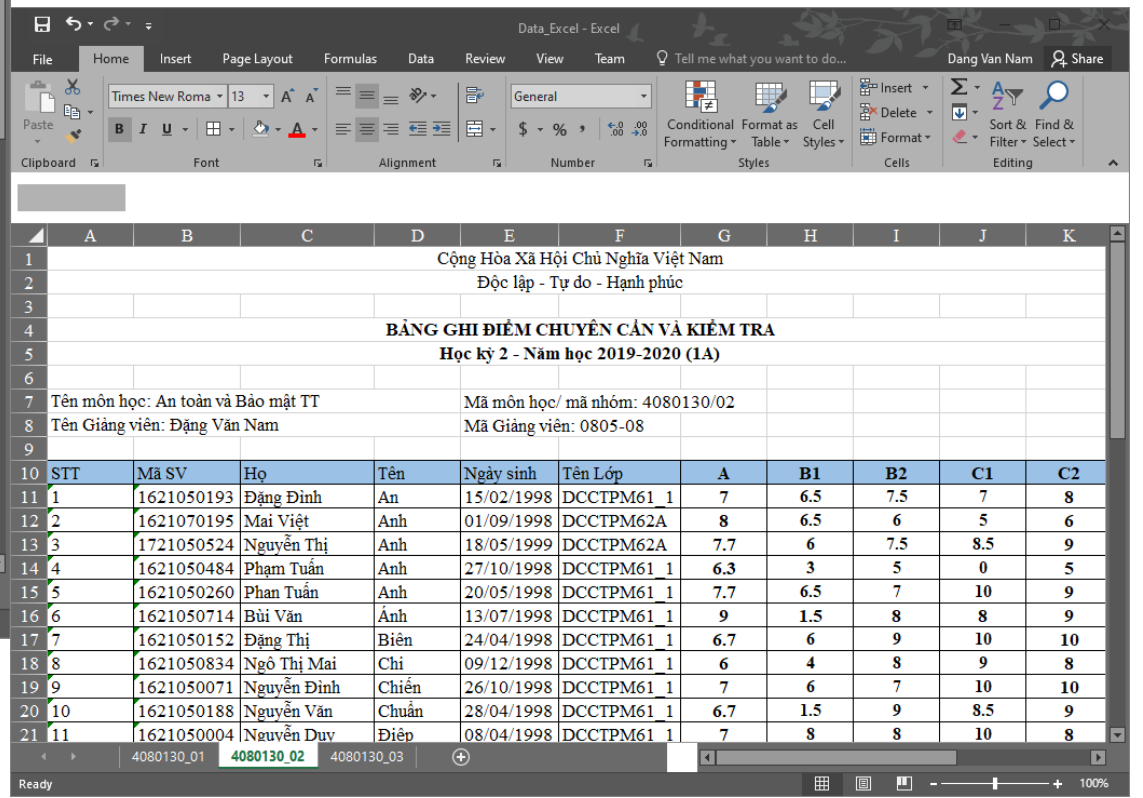
2.3 Đọc dữ liệu từ file Excel

2.3 Đọc file Excel

- File dữ liệu Excel demo gồm 3 sheet:



STT	Mã SV	Họ	Tên	Ngày sinh	Tên Lớp	A	B1	B2	C1	C2
1	1621050322	Phạm Trường	An	04/10/1998	DCCTPM61_1	8	0	5	7.5	8
2	1621050512	Nguyễn Quang Duy	Anh	08/10/1998	DCCTPM61_1	6	3	7.5	8.5	9
3	1621050211	Nguyễn Thế	Anh	26/08/1998	DCCTPM61_1	6.7	4	6.5	3	5
4	1621050827	Đỗ Xuân	Bách	13/07/1998	DCCTPM61_1	8	6.5	8	10	9
5	1621050298	Dương Trí	Bách	25/09/1998	DCCTPM61_1	7	5	8	8.5	9
6	1621050351	Nguyễn Văn	Bắc	04/02/1998	DCCTPM61_1	4.3	5	5	6	6
7	1621050422	Phạm Tiến	Cảnh	20/03/1998	DCCTPM61_1	7	6.5	9	10	10
8	1621050281	Trần Minh	Chiến	02/03/1998	DCCTPM61_1	5.3	3.5	6	8.5	8
9	1621050753	Vũ Trung	Chiến	22/01/1998	DCCTPM61_1	6	5	6.5	10	10
10	1621050283	Tạ Xuân	Công	21/08/1997	DCCTPM61_1	6	5.5	7	8.5	8
11	1621050122	Nguyễn Văn	Cường	11/07/1998	DCCTPM61_1	5.3	2	6	8.5	8
12	1621050203	Nguyễn Ngọc	Diện	05/08/1998	DCCTPM61_1	6	8	8	10	10
13	1621050090	Nguyễn Mạnh	Dũng	11/04/1998	DCCTPM61_1	6	5	6.5	10	8
14	1621050802	Đặng Ngọc	Dương	26/09/1998	DCCTPM61_1	7	0	8	0	5
15	1621050434	Lê Nhật	Dương	23/02/1998	DCCTPM61_1	6	9	7	10	9.5
16	1621050240	Phùng Thế	Đại	15/10/1998	DCCTPM61_1	6	9	7	10	9.5
17	1621050729	Đoàn Thành	Đạt	24/12/1998	DCCTPM61_1	7.3	5.5	6	10	10
18	1621050798	Hoàng Văn	Đạt	09/10/1998	DCCTPM61_1	6	7	6	10	8
19	1621050023	Lê Quý	Đạt	20/02/1998	DCCTPM61_1	5.7	4	6	3	5
20	1621050059	Ngô Xuân	Đạt	28/03/1998	DCCTPM61_1	6	2	8	0	4



Cộng Hòa Xã Hội Chủ Nghĩa Việt Nam

Độc lập - Tự do - Hạnh phúc

BẢNG GHI ĐIỂM CHUYÊN CẦN VÀ KIỂM TRA

Học kỳ 2 - Năm học 2019-2020 (1A)

Tên môn học: An toàn và Bảo mật TT — Mã môn học/ mã nhóm: 4080130/02

Tên Giảng viên: Đặng Văn Nam — Mã Giảng viên: 0805-08

STT	Mã SV	Họ	Tên	Ngày sinh	Tên Lớp	A	B1	B2	C1	C2
1	1621050193	Đặng Đình	An	15/02/1998	DCCTPM61_1	7	6.5	7.5	7	8
2	1621070195	Mai Việt	Anh	01/09/1998	DCCTPM62A	8	6.5	6	5	6
3	1721050524	Nguyễn Thị	Anh	18/05/1999	DCCTPM62A	7.7	6	7.5	8.5	9
4	1621050484	Phạm Tuấn	Anh	27/10/1998	DCCTPM61_1	6.3	3	5	0	5
5	1621050260	Phan Tuấn	Anh	20/05/1998	DCCTPM61_1	7.7	6.5	7	10	9
6	1621050714	Bùi Văn	Ánh	13/07/1998	DCCTPM61_1	9	1.5	8	8	9
7	1621050152	Đặng Thị	Biên	24/04/1998	DCCTPM61_1	6.7	6	9	10	10
8	1621050834	Ngô Thị Mai	Chi	09/12/1998	DCCTPM61_1	6	4	8	9	8
9	1621050071	Nguyễn Đình	Chiến	26/10/1998	DCCTPM61_1	7	6	7	10	10
10	1621050188	Nguyễn Văn	Chuẩn	28/04/1998	DCCTPM61_1	6.7	1.5	9	8.5	9
11	1621050004	Nguyễn Duy	Điệp	08/04/1998	DCCTPM61_1	7	8	8	10	8

2.3 Đọc file Excel

- Sử dụng phương thức **pd.read_excel()** để đọc dữ liệu từ file excel.
  - Lưu ý 2 tham số sheetname=‘’ xác định sheet muốn đọc dữ liệu (Mặc định là sheet đầu tiên)

```
1  import pandas as pd
2  path_excel = 'Data_Excersice\Data_Excel.xlsx'
3  #Đọc dữ liệu từ file excel
4  data_ex = pd.read_excel(path_excel)
5  data_ex.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 66 entries, 0 to 65
Data columns (total 11 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   STT        66 non-null     int64
 1   Mã SV      66 non-null     int64
 2   Họ         66 non-null     object
 3   Tên        66 non-null     object
 4   Ngày sinh  66 non-null     object
 5   Tên Lớp    66 non-null     object
 6   A          66 non-null     float64
 7   B1         66 non-null     float64
 8   B2         66 non-null     float64
 9   C1         66 non-null     float64
 10  C2         66 non-null     float64
dtypes: float64(5), int64(2), object(4)
memory usage: 5.8+ KB
```

```
1  #Hiển thị 5 dòng dữ liệu đầu tiên
2  data_ex.head()
```

	STT	Mã SV	Họ	Tên	Ngày sinh	Tên Lớp	A	B1	B2	C1	C2
0	1	1621050322	Phạm Trường	An	04/10/1998	DCCTPM61_1	8.0	0.0	5.0	7.5	8.0
1	2	1621050512	Nguyễn Quang Duy	Anh	08/10/1998	DCCTPM61_1	6.0	3.0	7.5	8.5	9.0
2	3	1621050211	Nguyễn Thế	Anh	26/08/1998	DCCTPM61_1	6.7	4.0	6.5	3.0	5.0
3	4	1621050827	Đỗ Xuân	Bách	13/07/1998	DCCTPM61_1	8.0	6.5	8.0	10.0	9.0
4	5	1621050298	Dương Trí	Bách	25/09/1998	DCCTPM61_1	7.0	5.0	8.0	8.5	9.0

2.3 Đọc file Excel

- Một vài tham số quan trọng trong phương thức **pd.read_excel()** để đọc dữ liệu từ file excel.

Argument	Description
io	A string containing the pathname of the given Excel file.
sheet_name	The Excel sheet name, or sheet number, of the data you want to import. The sheet number can be an integer where 0 is the first sheet, 1 is the second, etc. If a list of sheet names/numbers are given, then the output will be a dictionary of DataFrames. The default is to read all the sheets and output a dictionary of DataFrames.
header	Row number to use for the list of column labels. The default is 0, indicating that the first row is assumed to contain the column labels. If the data does not have a row of column labels, None should be used.
names	A separate Python list input of column names. This option is None by default. This option is the equivalent of assigning a list of column names to the columns attribute of the output DataFrame.
index_col	Specifies which column should be used for row indices. The default option is None, meaning that all columns are included in the data, and a range of numbers is used as the row indices.
usecols	An integer, list of integers, or string that specifies the columns to be imported into the DataFrame. The default is to import all columns. If a string is given, then Pandas uses the standard Excel format to select columns (e.g. "A:C,F,G" will import columns A, B, C, F, and G).
skiprows	The number of rows to skip at the top of the Excel sheet. Default is 0. This option is useful for skipping rows in Excel that contain explanatory information about the data below it.

https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.read_excel.html

2.3 Đọc file Excel

- Sử dụng phương thức **pd.read_excel()** với một số tham số cơ bản.

```
#Ví dụ:
#Đọc dữ liệu tại sheet đầu tiên,
#Chỉ lấy dữ liệu cột Mã SV và các cột điểm
#Thiết lập cột đầu tiên làm index
data_ex1 = pd.read_excel(path_excel,
                         sheet_name=0,
                         usecols=[1,6,7,8,9,10],
                         index_col=0)
data_ex1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 66 entries, 1621050322 to 1621050013
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   A       66 non-null     float64
 1   B1      66 non-null     float64
 2   B2      66 non-null     float64
 3   C1      66 non-null     float64
 4   C2      66 non-null     float64
dtypes: float64(5)
memory usage: 3.1 KB
```

```
#Hiển thị 5 dòng dữ liệu đầu tiên
data_ex1.head()
```

Mã SV	A	B1	B2	C1	C2
1621050322	8.0	0.0	5.0	7.5	8.0
1621050512	6.0	3.0	7.5	8.5	9.0
1621050211	6.7	4.0	6.5	3.0	5.0
1621050827	8.0	6.5	8.0	10.0	9.0
1621050298	7.0	5.0	8.0	8.5	9.0

2.3 Đọc file Excel

- Sử dụng phương thức **pd.read_excel()** với một số tham số cơ bản.
  - Đọc dữ liệu sheet 2 [‘4080130_02’], từ dòng 9.

```
#Ví dụ 3:
#Đọc dữ liệu tại sheet 2,từ dòng 9
data_ex3 = pd.read_excel(path_excel,
                         sheet_name='4080130_02',
                         skiprows=9)
data_ex3.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 39 entries, 0 to 38
Data columns (total 11 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   STT        39 non-null     int64
 1   Mã SV      39 non-null     int64
 2   Họ         39 non-null     object
 3   Tên        39 non-null     object
 4   Ngày sinh  39 non-null     object
 5   Tên Lớp    39 non-null     object
 6   A          39 non-null     float64
 7   B1         39 non-null     float64
 8   B2         39 non-null     float64
 9   C1         39 non-null     float64
 10  C2         39 non-null     float64
dtypes: float64(5), int64(2), object(4)
memory usage: 3.5+ KB
```

```
#Hiển thị 5 dòng dữ liệu đầu tiên
data_ex3.head()
```

	STT	Mã SV	Họ	Tên	Ngày sinh	Tên Lớp	A	B1	B2	C1	C2
0	1	1621050193	Đặng Đình	An	15/02/1998	DCCTPM61_1	7.0	6.5	7.5	7.0	8.0
1	2	1621070195	Mai Việt	Anh	01/09/1998	DCCTPM62A	8.0	6.5	6.0	5.0	6.0
2	3	1721050524	Nguyễn Thị	Anh	18/05/1999	DCCTPM62A	7.7	6.0	7.5	8.5	9.0
3	4	1621050484	Phạm Tuấn	Anh	27/10/1998	DCCTPM61_1	6.3	3.0	5.0	0.0	5.0
4	5	1621050260	Phan Tuấn	Anh	20/05/1998	DCCTPM61_1	7.7	6.5	7.0	10.0	9.0

2.3 Đọc file Excel

- Sử dụng phương thức **pd.read_excel()** với một số tham số cơ bản.
  - Đọc dữ liệu sheet 3 [‘4080130_03’], không có dòng header

```
#Ví dụ 4
#Đọc dữ liệu từ sheet: '4080130_03'
#Dữ liệu không chứa dòng header
data_ex4 = pd.read_excel(path_excel,
                         sheet_name='4080130_03',
                         header=None)
data_ex4.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 39 entries, 0 to 38
Data columns (total 11 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   0       39 non-null     int64
 1   1       39 non-null     int64
 2   2       39 non-null     object
 3   3       39 non-null     object
 4   4       39 non-null     object
 5   5       39 non-null     object
 6   6       39 non-null     float64
 7   7       39 non-null     float64
 8   8       39 non-null     float64
 9   9       39 non-null     float64
 10  10      39 non-null     float64
dtypes: float64(5), int64(2), object(4)
memory usage: 3.5+ KB
```

```
#Hiển thị 5 dòng dữ liệu đầu tiên
data_ex4.head()
```

	0	1	2	3	4	5	6	7	8	9	10
0	1	1621050041	Đào Tuấn	Anh	22/10/1998	DCCTPM61_1	6.7	9.0	5.5	8.5	8.0
1	2	1621050262	Vũ Thị Lan	Anh	26/09/1998	DCCTPM61_1	6.7	7.0	9.0	8.5	6.0
2	3	1621050083	Trịnh Như	Bình	06/04/1998	DCCTPM61_1	7.3	8.5	9.5	10.0	9.0
3	4	1621050113	Trần Văn	Cương	19/06/1998	DCCTPM61_1	5.7	5.0	6.0	10.0	5.0
4	5	1621050384	Nguyễn Sỹ	Dũng	02/10/1998	DCCTPM61_1	7.0	0.0	7.5	8.5	9.0

2.3 Đọc file Excel

- Sử dụng phương thức **pd.read_excel()** với một số tham số cơ bản.
  - Đọc dữ liệu sheet 3 [‘4080130_03’],
  - Không có dòng header
  - Chỉ lấy dữ liệu cột 1,6,7,8,910
  - Đặt tên cho các cột lần lượt là [‘Mã SV’, ‘A’, ‘B1’,’B2’,’C1’,’C2’]
  - Thiết lập cột đầu tiên làm Index

```
5  data_ex41 = pd.read_excel(path_excel,
6                            sheet_name='4080130_03',
7                            header=None,
8                            usecols=[1,6,7,8,9,10],
9                            names=['Mã SV','A','B1','B2','C1','C2'],
10                           index_col=0)
11 data_ex41.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 39 entries, 1621050041 to 1621050034
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   A       39 non-null     float64
 1   B1      39 non-null     float64
 2   B2      39 non-null     float64
 3   C1      39 non-null     float64
 4   C2      39 non-null     float64
dtypes: float64(5)
memory usage: 1.8 KB
```

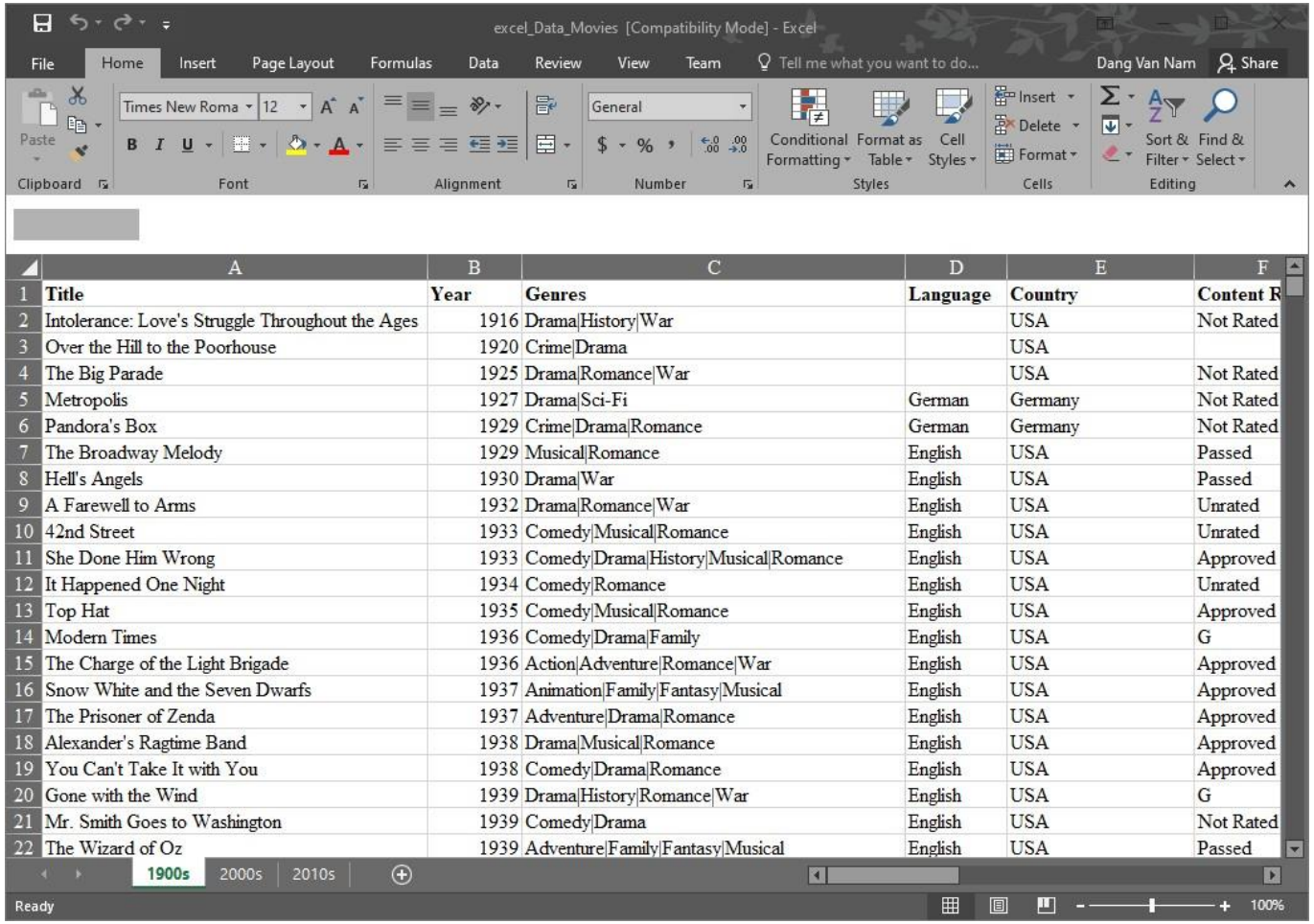
```
1 #Hiển thị 5 dòng dữ liệu đầu tiên
2 data_ex41.head()
```

Mã SV	A	B1	B2	C1	C2
1621050041	6.7	9.0	5.5	8.5	8.0
1621050262	6.7	7.0	9.0	8.5	6.0
1621050083	7.3	8.5	9.5	10.0	9.0
1621050113	5.7	5.0	6.0	10.0	5.0
1621050384	7.0	0.0	7.5	8.5	9.0

Thực hành 2.3

Thực hành 2.3

Yêu cầu: Sinh viên đọc dữ liệu dạng excel lưu trong file excel_Data_Movies.xls theo từng sheet



	A	B	C	D	E	F
1	**Title**	**Year**	**Genres**	**Language**	**Country**	**Content R**
2	Intolerance: Love's Struggle Throughout the Ages	1916	Drama|History|War		USA	Not Rated
3	Over the Hill to the Poorhouse	1920	Crime|Drama		USA	
4	The Big Parade	1925	Drama|Romance|War		USA	Not Rated
5	Metropolis	1927	Drama|Sci-Fi	German	Germany	Not Rated
6	Pandora's Box	1929	Crime|Drama|Romance	German	Germany	Not Rated
7	The Broadway Melody	1929	Musical|Romance	English	USA	Passed
8	Hell's Angels	1930	Drama|War	English	USA	Passed
9	A Farewell to Arms	1932	Drama|Romance|War	English	USA	Unrated
10	42nd Street	1933	Comedy|Musical|Romance	English	USA	Unrated
11	She Done Him Wrong	1933	Comedy|Drama|History|Musical|Romance	English	USA	Approved
12	It Happened One Night	1934	Comedy|Romance	English	USA	Unrated
13	Top Hat	1935	Comedy|Musical|Romance	English	USA	Approved
14	Modern Times	1936	Comedy|Drama|Family	English	USA	G
15	The Charge of the Light Brigade	1936	Action|Adventure|Romance|War	English	USA	Approved
16	Snow White and the Seven Dwarfs	1937	Animation|Family|Fantasy|Musical	English	USA	Approved
17	The Prisoner of Zenda	1937	Adventure|Drama|Romance	English	USA	Approved
18	Alexander's Ragtime Band	1938	Drama|Musical|Romance	English	USA	Approved
19	You Can't Take It with You	1938	Comedy|Drama|Romance	English	USA	Approved
20	Gone with the Wind	1939	Drama|History|Romance|War	English	USA	G
21	Mr. Smith Goes to Washington	1939	Comedy|Drama	English	USA	Not Rated
22	The Wizard of Oz	1939	Adventure|Family|Fantasy|Musical	English	USA	Passed

2.4 Đọc dữ liệu từ file Json

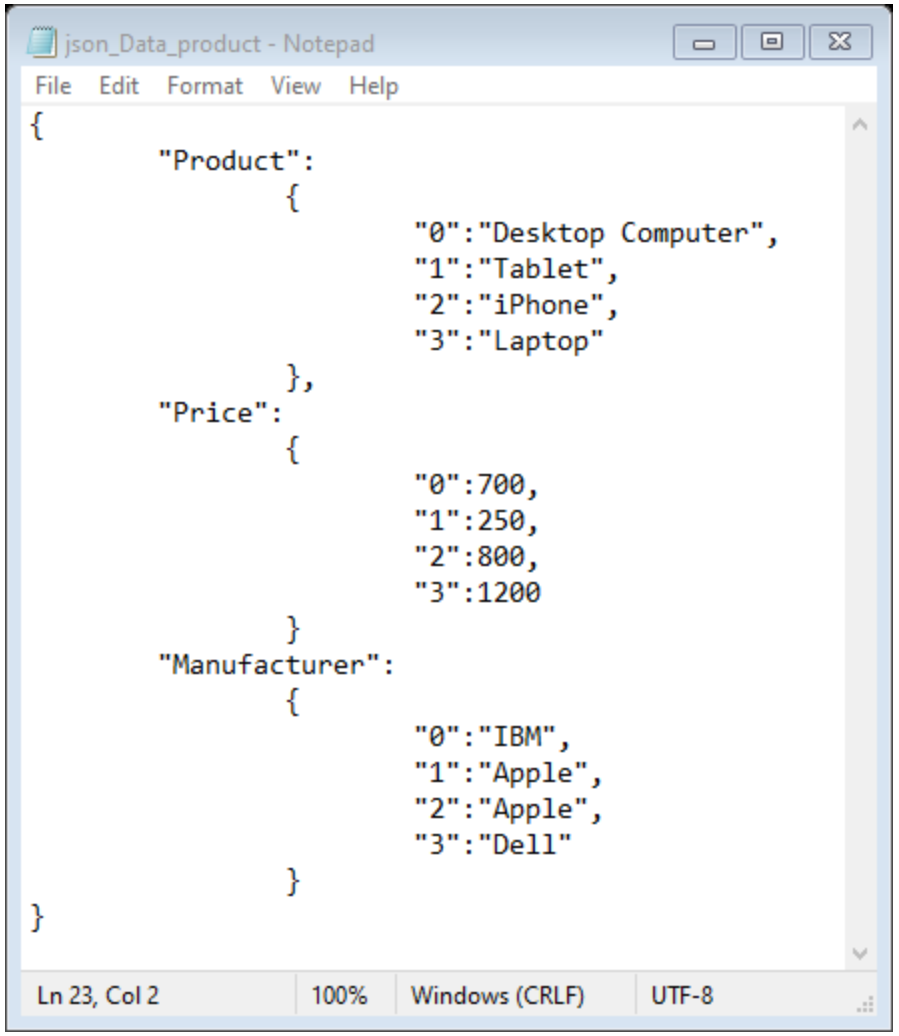
2.4 Đọc file Json

- **JSON** là chữ viết tắt của **J**ava**s**cript **O**bject **N**otation, đây là một dạng dữ liệu tuân theo một quy luật nhất định mà hầu hết các ngôn ngữ lập trình đều có thể đọc được.
- JSON lưu trữ các dữ liệu theo cặp **khóa (key)** và **giá trị (value)**. So với XML thìJSON có định dạng đơn giản, dễ sử dụng và nhẹ hơn.


```
{
    "orders": [
        {
            "orderno": "748745375",
            "date": "June 30, 2088 1:54:23 AM",
            "trackingno": "TN0039291",
            "custid": "11045",
            "customer": [
                {
                    "custid": "11045",
                    "fname": "Sue",
                    "lname": "Hatfield",
                    "address": "1409 Silver Street",
                    "city": "Ashland",
                    "state": "NE",
                    "zip": "68003"
                }
            ]
        }
    ]
}
```

2.4 Đọc file Json

- Sử dụng phương thức **pd.read_json()** để đọc dữ liệu từ file json vào dataframe.




```
1 import pandas as pd
2 path_json = 'Data_Excersice\json_Data_product.json'
3 #Đọc dữ liệu từ file Json vào biến DataFrame
4 data_json = pd.read_json(path_json)
5 data_json
```

	Product	Price	Manufacturer
0	Desktop Computer	700	IBM
1	Tablet	250	Apple
2	iPhone	800	Apple
3	Laptop	1200	Dell

```
1 #Đọc dữ liệu từ file Json vào biến DataFrame
2 #Sử dụng tham số orient='index'
3 data_json1 = pd.read_json(path_json, orient='index')
4 data_json1
```

	0	1	2	3
Product	Desktop Computer	Tablet	iPhone	Laptop
Price	700	250	800	1200
Manufacturer	IBM	Apple	Apple	Dell

https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.read_json.html

2.4 Đọc file Json

- Có thể sử dụng **thư viện json** để làm việc với file có định dạng json.

```
#Hoặc Sử dụng thư viện json làm việc với dữ liệu json
import json
with open(path_json, 'r') as myfile:
    data=myfile.read()

# Đọc dữ liệu vào biến obj
obj = json.loads(data)
type(obj)
```

```
dict
```

```
#Lấy giá trị của key = 'Product'
obj['Product']
```

```
{'0': 'Desktop Computer', '1': 'Tablet', '2': 'iPhone', '3': 'Laptop'}
```

```
#Lấy giá trị của key = 'Price'
obj['Price']
```

```
{'0': 700, '1': 250, '2': 800, '3': 1200}
```

Thực hành 2.4

Thực hành 2.4

Yêu cầu: Sinh viên đọc dữ liệu dạng json lưu trong file json_Data_flights.json vào trong DataFrame

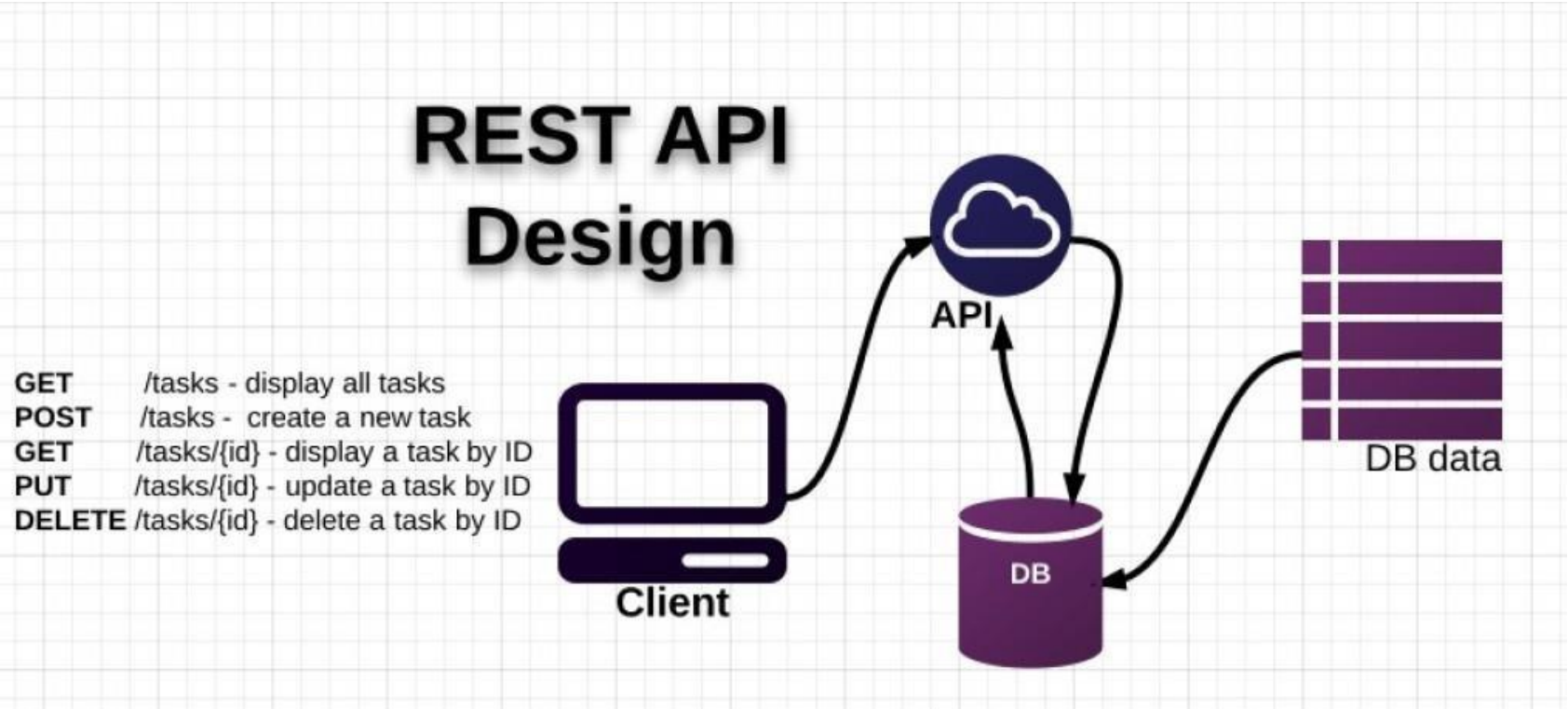
```
1 data_json.head()
```

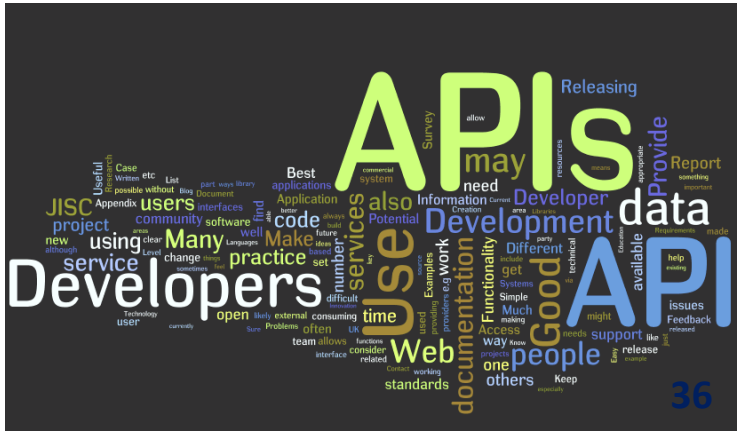
	_id	dofW	carrier	origin	dest	crsdephour	crsdeptime	depdelay	crsarrtime	arrdelay	crselapsedtime	dist
0	ATL_BOS_2017-01-01_11_DL_1204	7	DL	ATL	BOS	11	1141	0	1409	0	148	946
1	ATL_BOS_2017-01-01_13_WN_3677	7	WN	ATL	BOS	13	1335	0	1600	0	145	946
2	ATL_BOS_2017-01-01_14_DL_1208	7	DL	ATL	BOS	14	1416	0	1644	0	148	946
3	ATL_BOS_2017-01-01_16_DL_980	7	DL	ATL	BOS	16	1616	15	1849	0	153	946
4	ATL_BOS_2017-01-01_18_WN_461	7	WN	ATL	BOS	18	1845	0	2110	0	145	946

3. Lấy dữ liệu từ API

3.1 Giới thiệu REST API

- REST API là một ứng dụng chuyển đổi cấu trúc dữ liệu có các phương thức để kếtnối với các thư viện và ứng dụng khác.
- Các trang web ngày nay thường sử dụng REST API để cho phép kết nối dữ liệu



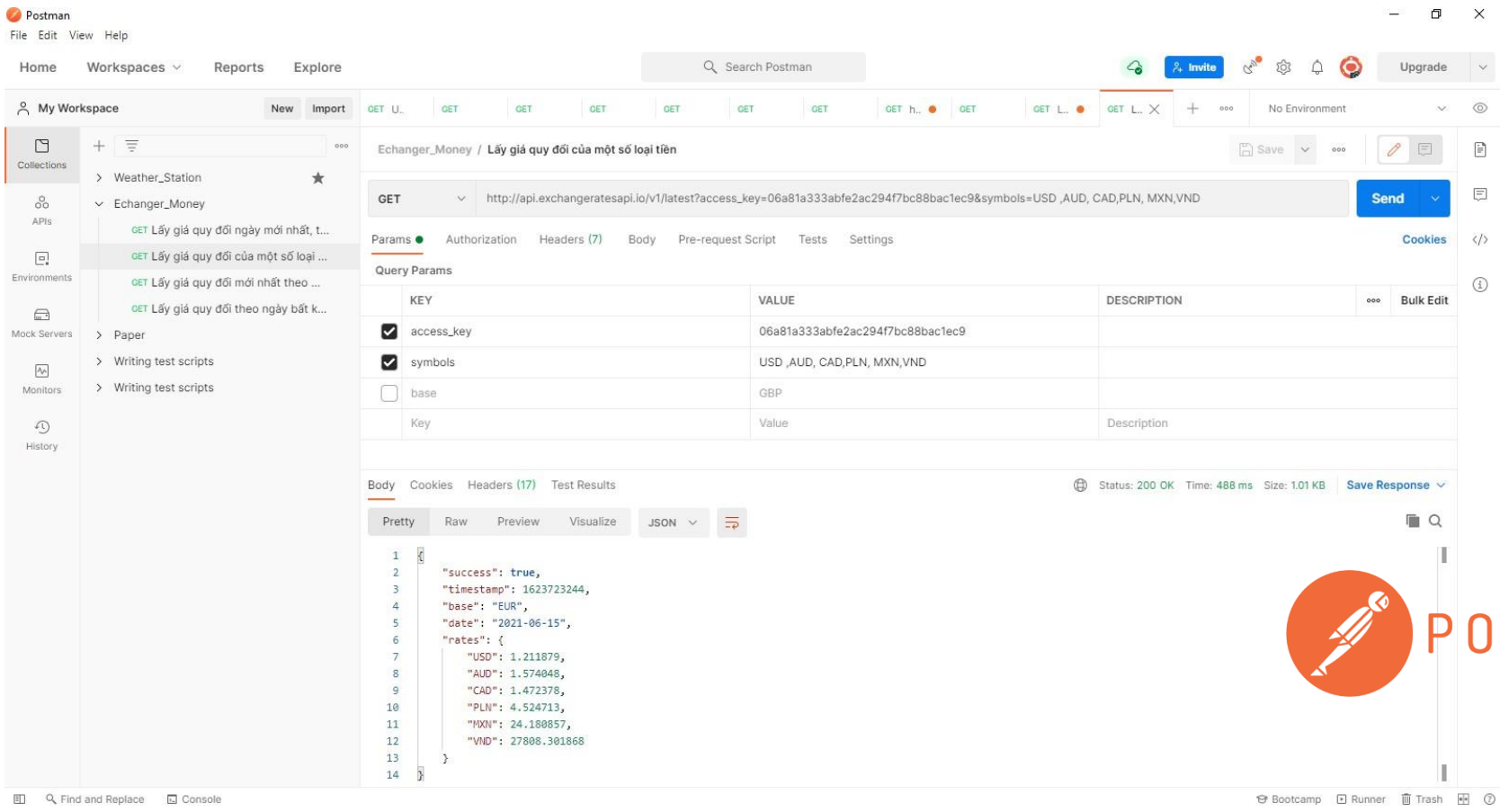


3.1 Giới thiệu REST API

- Vd API lấy dữ liệu tỷ giá ngoại tệ:
  - Lấy tỷ giá ngày cuối cùng, theo EUR:

http://api.exchangeratesapi.io/v1/latest?access_key=06a81a333abfe2ac294f7bc88bac1ec9

Mã NT	Tên Ngoại Tệ
USD	ĐÔ LA MỸ
AUD	ĐÔ LA ÚC
CAD	ĐÔ CANADA
CHF	FRANCE THỤY SĨ
EUR	EURO
GBP	BẢNG ANH
JPY	YÊN NHẬT
SGD	ĐÔ SINGAPORE
THB	BẠT THÁI LAN
MYR	RINGGIT MÃ LAY
DKK	KRONE ĐAN MẠCH
HKD	ĐÔ HONGKONG
INR	RUPI ẤN ĐỘ
KRW	WON HÀN QUỐC



3.2 Lấy dữ liệu từ API

- sử dụng **thư viện requests, phương thức get()** để lấy dữ liệu theo url
- Lấy tỷ giá mới nhất (chuyển đổi theo EUR (Default))

```python
#Lấy dữ liệu tỷ lệ đổi tiền
import pandas as pd
import requests as rq
url_api = 'http://api.exchangeratesapi.io/v1/latest?access_key=06a81a333abfe2ac294f7bc88bac1ec9'
#lấy dữ liệu theo url_api
result_money = rq.get(url_api)

#check status của request
print(result_money.status_code)
```

```
200
```

```python
#Chuyển đổi dữ liệu sang kiểu string
data_text = result_money.text
#chuyển đổi dữ liệu sang kiểu json
data_json = result_money.json()
print('Text: ', data_text)
print('-----------------------------------------------------')
print('Json:', data_json)
```

```
Text:  {"success":true,"timestamp":1623723244,"base":"EUR","date":"2021-06-15","rates":{"AED":4.45146
9,"AFN":95.687275,"ALL":122.985167,"AMD":630.187794,"ANG":2.175393,"AOA":778.026383,"ARS":115.43133
6,"AUD":1.574048,"AWG":2.181988,"AZN":2.059938,"BAM":1.955651,"BBD":2.447085,"BDT":102.770573,"BGN":
1.955849,"BHD":0.456871,"BIF":2397.491166,"BMD":1.211879,"BND":1.607786,"BOB":8.380966,"BRL":6.13222
```

```python
#Chuyển đổi dữ liệu Json sang kiểu DataFrame
df = pd.DataFrame(data_json)
print(df.head(10))
```

	success	timestamp	base	date	rates
AED	True	1623723244	EUR	2021-06-15	4.451469
AFN	True	1623723244	EUR	2021-06-15	95.687275
ALL	True	1623723244	EUR	2021-06-15	122.985167
AMD	True	1623723244	EUR	2021-06-15	630.187794
ANG	True	1623723244	EUR	2021-06-15	2.175393
AOA	True	1623723244	EUR	2021-06-15	778.026383
ARS	True	1623723244	EUR	2021-06-15	115.431336
AUD	True	1623723244	EUR	2021-06-15	1.574048
AWG	True	1623723244	EUR	2021-06-15	2.181988
AZN	True	1623723244	EUR	2021-06-15	2.059938

https://2.python-requests.org/en/master/user/quickstart/#make-a-request

3.2 Lấy dữ liệu từ API

- Truyền tham số cho url: params={‘key’:’values’}
- Chỉ lấy Lấy tỷ giá một số loại tiền nhất định:

Mã NT	Tên Ngoại Tệ
USD	ĐÔ LA MỸ
AUD	ĐÔ LA ÚC
CAD	ĐÔ CANADA
CHF	FRANCE THỤY SĨ
EUR	EURO
GBP	BẢNG ANH
JPY	YÊN NHẬT
SGD	ĐÔ SINGAPORE
THB	BẠT THÁI LAN
MYR	RINGGIT MÃ LAY
DKK	KRONE ĐAN MẠCH
HKD	ĐÔ HONGKONG
INR	RUPI ẤN ĐỘ
KRW	WON HÀN QUỐC

```
1  #Truyền tham số cho url
2  #Lấy tỷ giá của một số ngoại tệ:
3  #1.GBP (Bảng Anh)
4  #2.JPY (Yên Nhật)
5  #3.MYR (Ringgit - Malaysia)
6  #4.THB (Bạt Thái Lan),
7  #5.KRW (Won - Hàn Quốc)
8  #6.VND (Việt Nam Đồng)
9
10 result_money1 = rq.get(url_api,params={'symbols':'USD, JPY, THB, VND, MYR, GBP, KRW'})
11 data_json1 = result_money1.json()
12 df1 = pd.DataFrame(data_json1)
13 print(df1.head(10))
```

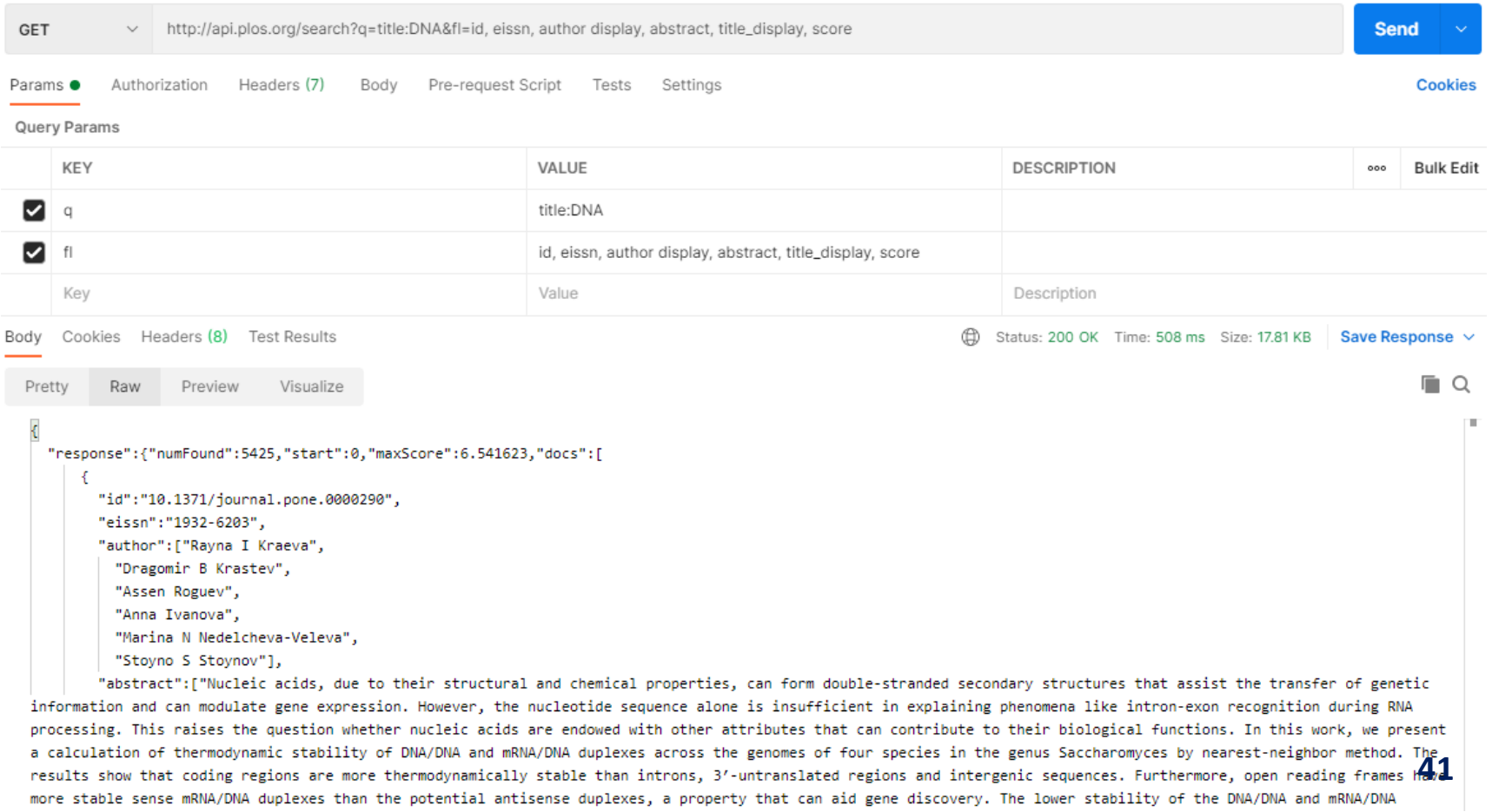
```
     success   timestamp base        date          rates
GBP     True  1623723244  EUR  2021-06-15       0.859289
JPY     True  1623723244  EUR  2021-06-15     133.460542
KRW     True  1623723244  EUR  2021-06-15    1355.171091
MYR     True  1623723244  EUR  2021-06-15       4.985063
THB     True  1623723244  EUR  2021-06-15      37.749351
USD     True  1623723244  EUR  2021-06-15       1.211879
VND     True  1623723244  EUR  2021-06-15   27808.301868
```

Thực hành 2.5

Thực hành 2.5

Yêu cầu: Sinh viên lấy dữ liệu các bài báo trong tiêu đề có từ Viruscung cấp từ API có địa chỉ: http://api.plos.org/search?q=title:VIRUS

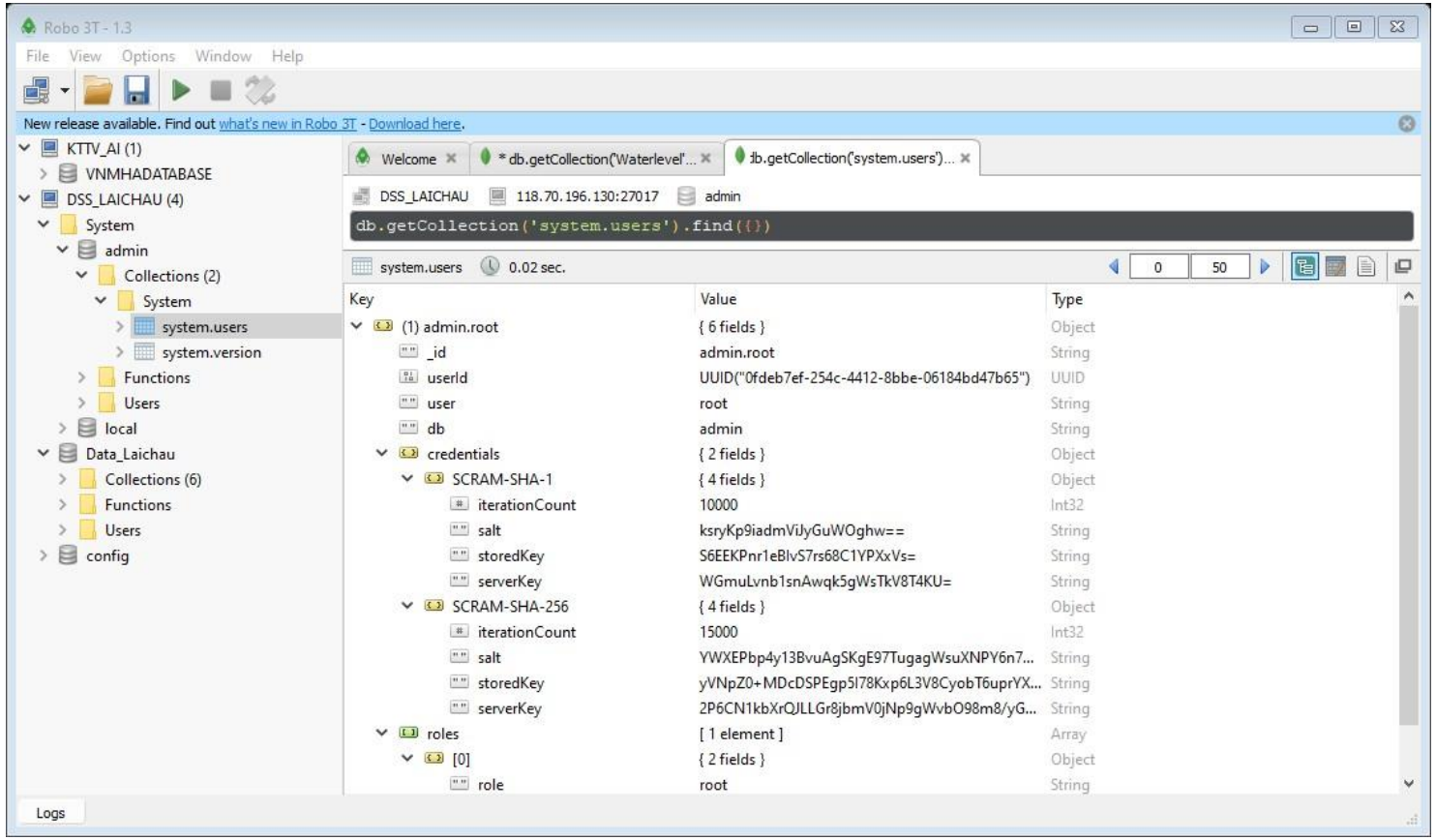
- Tạo DataFrame bao gồm các thông tin: **id, eissn, author display, abstract, title_display, score.**
- Lưu dữ liệu từ Dataframe ra file Paper.csv



4. Lấy dữ liệu từ MongoDB

4.1 CSDL NoSQL

- Cơ sở dữ liệu NoSQL sử dụng nhiều mô hình dữ liệu để truy cập và quản lý dữ liệu. Cácloại cơ sở dữ liệu này được tối ưu hóa dành riêng cho các ứng dụng yêu cầu mô hình dữliệu linh hoạt có lượng dữ liệu lớn và độ trễ thấp, có thể đạt được bằng cách giảm bớt mộtsố hạn chế về tính nhất quán của dữ liệu của các cơ sở dữ liệu khác.




```
{
    "_id" : "admin.root",
    "userId" : UUID("0fdeb7ef-254c-4412-8bbe-06184bd47b65"),
    "user" : "root",
    "db" : "admin",
    "credentials" : {
        "SCRAM-SHA-1" : {
            "iterationCount" : 10000,
            "salt" : "ksryKp9iadmViJyGuWOghw==",
            "storedKey" : "S6EEKPnrleBIvS7rs68C1YPXxVs=",
            "serverKey" : "WGmuLvnblsnAwqk5gWsTkV8T4KU="
        },
        "SCRAM-SHA-256" : {
            "iterationCount" : 15000,
            "salt" : "YWXEPbp4yl3BvuAgSKgE97TugagWsuXNPY6n7Q==",
            "storedKey" : "yVNpZ0+MDcDSPEgp5I78Kxp6L3V8CyobT6uprYXX55k=",
            "serverKey" : "2P6CN1kbXrQJLLGr8jbmV0jNp9gWvbO98m8/yGZeJQI="
        }
    },
    "roles" : [
        {
            "role" : "root",
            "db" : "admin"
        }
    ]
}
```

4.2 Truy cập dữ liệu Mongo

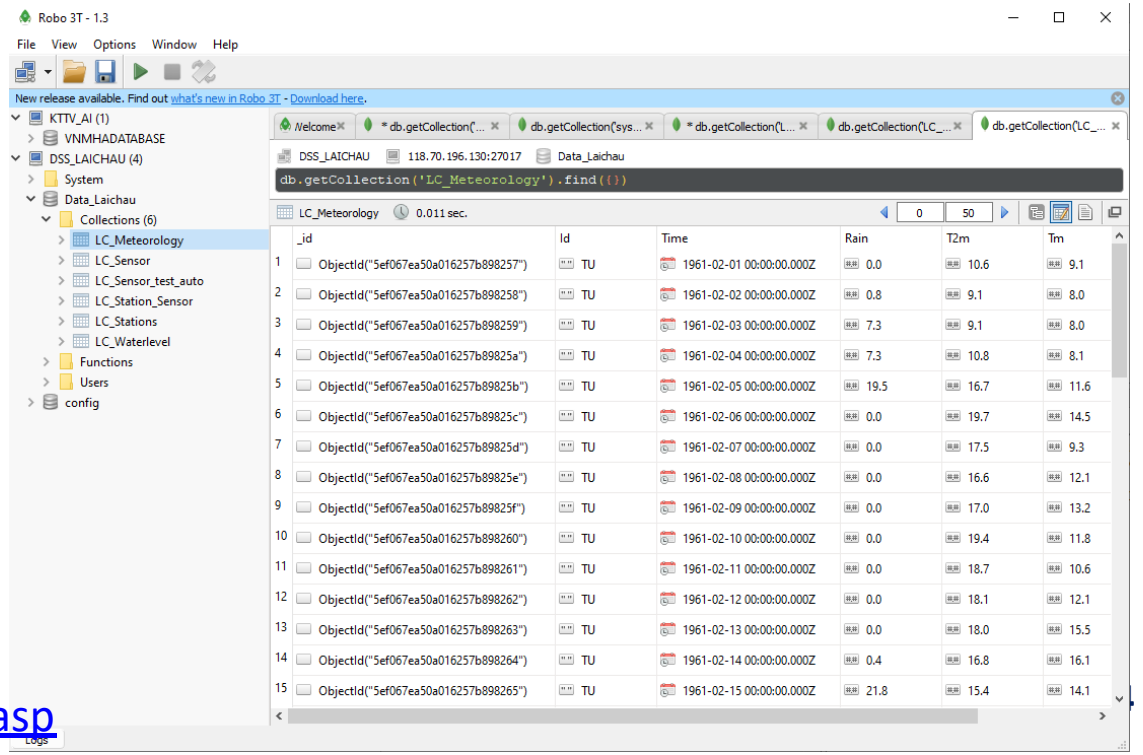
- Để kết nối tới CSDL MongoDB sử dụng thư viện **pymongo.**

```
#Kiểm tra thư viện pymongo
import pymongo
print('Version: ',pymongo.version)
```

```
Version:  3.7.2
```

- Thực hiện kết nối và lấy dữ liệu từ collection: **LC_Meteorology** với các tham số:

address="118.70.196.130:27017"
user='root'
pas='Humg@123mg'
auth='SCRAM-SHA-1'
database = 'Data_Laichau'
coll_name='LC_Meteorology'



https://www.w3schools.com/python/python_mongodb_getstarted.asp

4.2 Truy cập dữ liệu Mongo

- Kết nối tới CSDL MongoDB.

```
#Kết nối tới CSDL Mongo
import pymongo as mg

#Thiết lập Tham số tới server CSDL
address="118.70.196.130:27017"
user='root'
pas='Humg@123mg'
auth='SCRAM-SHA-1'
database = 'Data_Laichau'
coll_name='LC_Meteorology'
#----------------------------------------
#Thực hiện kết nối tới CSDL
client = mg.MongoClient(address,username=user,password=pas,authMechanism=auth)
db=client[database] #truy cập đến CSDL: Data_Laichau
col = db[coll_name] #truy cập đến collection: LC_Meteorology
```

```
#Thực hiện truy vấn lấy dữ liệu
#Thiết lập câu truy vấn thực hiện:
#1. Lấy dữ liệu từ trạm có mã TU
#2. Sắp xếp dữ liệu theo thứ tự thời gian giảm dần
stationid = 'TU'
data_mg = col.find({"Id":stationid}).sort([('Time',-1)])
print(type(data_mg))
```

```
<class 'pymongo.cursor.Cursor'>
```

4.2 Truy cập dữ liệu Mongo

- Chuyển đổi kết quả từ câu truy vấn về dữ liệu DataFrame.

```
#lấy dữ liệu chuyển vào các biến kiểu list
dt_time,dt_rain,dt_t2m,dt_tm,dt_tx = [],[],[],[],[]
for i in data_mg:
    dt_time.append(str(i['Time']))
    dt_rain.append(i['Rain'])
    dt_t2m.append(i['T2m'])
    dt_tm.append(i['Tm'])
    dt_tx.append(i['Tx'])
```

```
#Kết hợp các biến list lại
ziplist_water = list(zip(dt_time,dt_rain,dt_t2m, dt_tm,dt_tx))
#chuyển về dữ liệu DataFrame
import pandas as pd
df = pd.DataFrame(ziplist_water,
                  columns = ['Time' , 'Rain','T2m','Tm','Tx'],
                  index=None)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21143 entries, 0 to 21142
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Time    21143 non-null  object
 1   Rain    21112 non-null  float64
 2   T2m     21142 non-null  float64
 3   Tm      21142 non-null  float64
 4   Tx      21142 non-null  float64
dtypes: float64(4), object(1)
memory usage: 826.0+ KB
```

```
#hiển thị 5 dòng dữ liệu đầu tiên
df.head()
```

	Time	Rain	T2m	Tm	Tx
0	2018-12-31 00:00:00	2.6	9.7	8.3	11.5
1	2018-12-30 00:00:00	13.9	10.0	9.4	11.6
2	2018-12-29 00:00:00	1.6	14.1	11.5	18.4
3	2018-12-28 00:00:00	0.0	17.2	12.6	24.8
4	2018-12-27 00:00:00	0.0	17.0	12.0	25.8

Thực hành 4

Thực hành 4

Yêu cầu 4.1: Sinh viên tạo DataFrame đọc dữ liệu trong Collection LC_Meteorolory trạm có Id = 'LC' sắp xếp theo thứ tự thời gian giảm dần, chỉ lấy 1000 dòng dữ liệu đầu tiên và chỉ lấy dữ liệu các cột Time, Rain, T2m.

```
db.getCollection('LC_Meteorology')
.find({'Id':'LC'},{'Time':1,'Rain':1,'T2m':1})
.sort({'Time':-1})
.limit(1000)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Time    1000 non-null   object
 1   Rain    1000 non-null   float64
 2   T2m     1000 non-null   float64
dtypes: float64(2), object(1)
memory usage: 23.6+ KB
```

```
#Hiển thị 10 hàng dữ liệu đầu tiên
df_lc.head(10)
```

	Time	Rain	T2m
0	2018-12-31 00:00:00	1.0	14.4
1	2018-12-30 00:00:00	11.9	15.1
2	2018-12-29 00:00:00	3.0	20.1
3	2018-12-28 00:00:00	0.0	19.3
4	2018-12-27 00:00:00	0.0	18.5
5	2018-12-26 00:00:00	0.0	17.8
6	2018-12-25 00:00:00	0.0	18.8
7	2018-12-24 00:00:00	0.0	18.9
8	2018-12-23 00:00:00	0.0	18.4
9	2018-12-22 00:00:00	0.0	19.2

Thực hành 4

Yêu cầu 4.2: Sinh viên tạo DataFrame df_station đọc toàn bộ dữ liệu trong Collection LC_Stations.

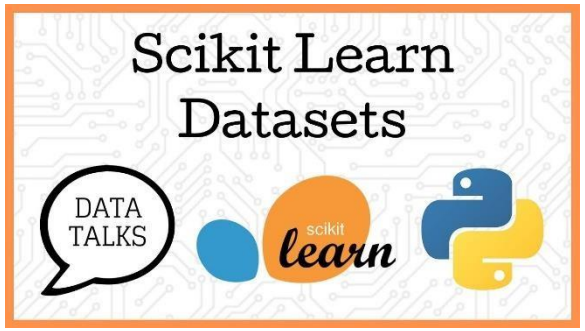
Idstation	Namestation	Timestart	Timeend
LC	Lai Chau	1961-01-01 00:00:00.000Z	2018-12-31 00:00:00.000Z
TU	Than Uyen	1961-02-01 00:00:00.000Z	2018-12-31 00:00:00.000Z
MT	Muong Te	1961-01-01 00:00:00.000Z	2018-12-31 00:00:00.000Z
SH	Sin Ho	1961-01-01 00:00:00.000Z	2018-12-31 00:00:00.000Z
TD	Tam Duong	1975-01-01 00:00:00.000Z	2018-12-31 00:00:00.000Z

5. Tải dữ liệu mẫu với Sklearn

Dữ liệu mẫu trong Sklearn

- Trong thư viện sklearn cung cấp một số bộ dữ liệu mẫu, kích thước nhỏ, rất hữu ích cho việc demo các thuật toán học máy.
- Sử dụng dữ liệu mẫu: **import sklearn.datasets**

`load_boston`(*[, return_X_y])	Load and return the boston house-prices dataset (regression).
`load_iris`(* [, return_X_y, as_frame])	Load and return the iris dataset (classification).
`load_diabetes`(* [, return_X_y, as_frame])	Load and return the diabetes dataset (regression).
`load_digits`(* [, n_class, return_X_y, as_frame])	Load and return the digits dataset (classification).
`load_linnerud`(* [, return_X_y, as_frame])	Load and return the physical excercise linnerud dataset.
`load_wine`(* [, return_X_y, as_frame])	Load and return the wine dataset (classification).
`load_breast_cancer`(* [, return_X_y, as_frame])	Load and return the breast cancer wisconsin dataset (classification).



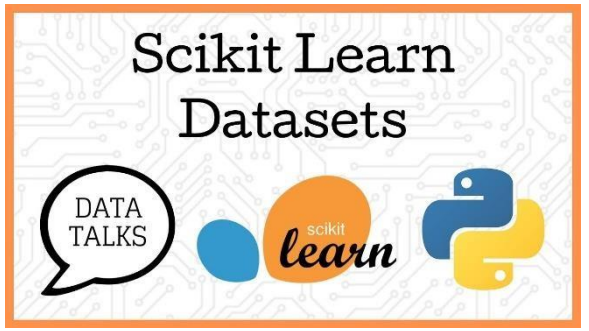
Read more: https://scikit-learn.org/stable/datasets/index.html#

1) Boston house-prices dataset

- Tập dữ liệu về giá nhà tại thành phố Boston.

Data Set Characteristics:

Number of Instances:	506
Number of Attributes:	13 numeric/categorical predictive. Median Value (attribute 14) is usually the target.
Attribute Information (in order):	• CRIM per capita crime rate by town • ZN proportion of residential land zoned for lots over 25,000 sq.ft. • INDUS proportion of non-retail business acres per town • CHAS Charles River dummy variable (= 1 if tract bounds river; 0 otherwise) • NOX nitric oxides concentration (parts per 10 million) • RM average number of rooms per dwelling • AGE proportion of owner-occupied units built prior to 1940 • DIS weighted distances to five Boston employment centres • RAD index of accessibility to radial highways • TAX full-value property-tax rate per $10,000 • PTRATIO pupil-teacher ratio by town • B 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town • LSTAT % lower status of the population • MEDV Median value of owner-occupied homes in $1000's
Missing Attribute Values:	None
Creator:	Harrison, D. and Rubinfeld, D.L.



Samples total	506
Dimensionality	13
Features	real, positive
Targets	real 5. - 50.

```
#Đọc tập dữ liệu Boston House Prices Dataset
import sklearn.datasets as datask
X, y = datask.load_boston(return_X_y=True)

print(type(X))
print('Kích thước dữ liệu đầu vào (features):', X.shape)
print('Kích thước dữ liệu đầu ra (target):', X.shape)
```

```
<class 'numpy.ndarray'>
Kích thước dữ liệu đầu vào (features): (506, 13)
Kích thước dữ liệu đầu ra (target): (506, 13)
```

2) The iris dataset

- Tập dữ liệu về thông số chiều rộng, chiều dài của lá hóa và cánh hóa của 3 loại hoa Lan

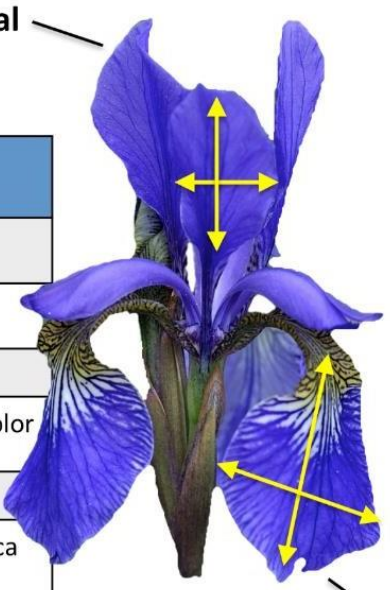
Iris Versicolor | Iris Setosa | Iris Virginica

Iris Dataset

Classes	3
Samples per class	50
Samples total	150
Dimensionality	4
Features	real, positive

Samples
(instances, observations)

	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
			...		
50	6.4	3.5	4.5	1.2	Versicolor
			...		
150	5.9	3.0	5.0	1.8	Virginica

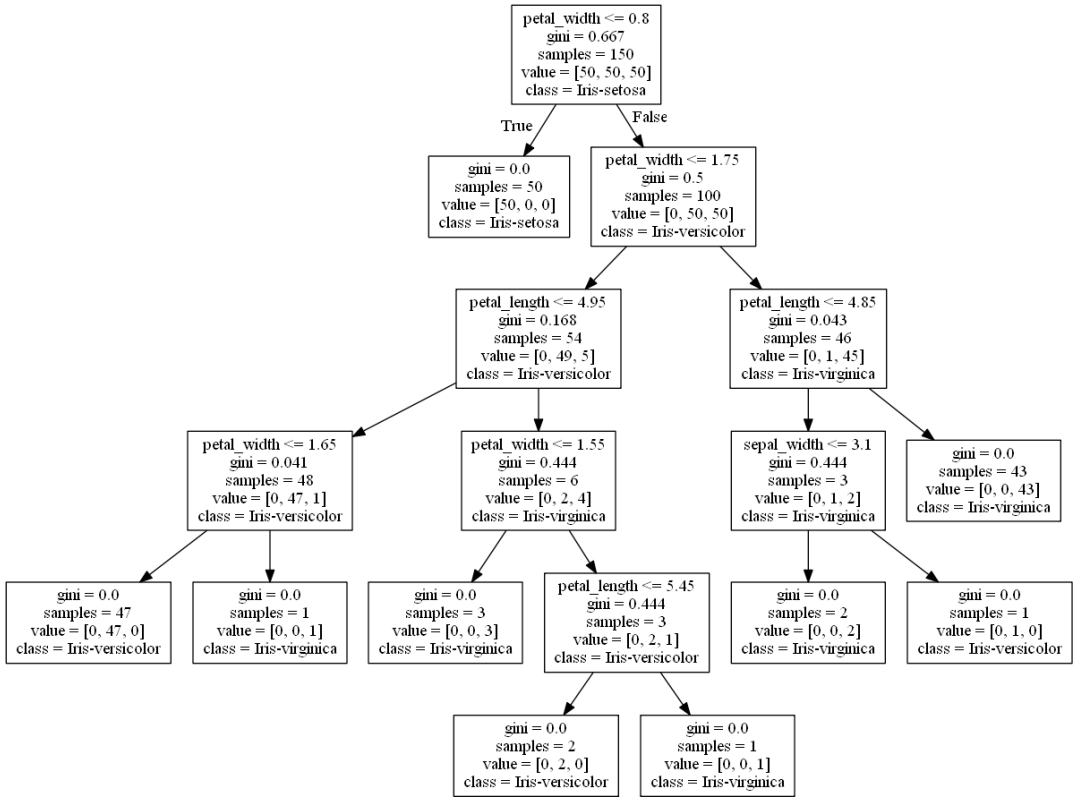


Features
(attributes, measurements, dimensions)

Class labels
(targets)

2) The iris dataset

- Tham số:
  - **return_X_y=True**: Dữ liệu được đọc vào các biến array.
  - **as_frame = True**: Dữ liệu được đọc vào các biến DataFrame




```python
#Đọc tập dữ liệu Iris Dataset
import sklearn.datasets as datask
X_iris, y_iris = datask.load_iris(return_X_y=True)

print(type(X_iris))
print('Kích thước dữ liệu đầu vào (features):', X_iris.shape)
print('Kích thước dữ liệu đầu ra (target)   :', y_iris.shape)
print('Bộ dữ liệu 1)  ', X[1,:], '--',y[1])
print('Bộ dữ liệu 55) ', X[55,:], '--',y[55])
print('Bộ dữ liệu 111)', X[111,:], '--',y[111])
```

```
<class 'numpy.ndarray'>
Kích thước dữ liệu đầu vào (features): (150, 4)
Kích thước dữ liệu đầu ra (target)   : (150,)
Bộ dữ liệu 1)   [4.9 3.  1.4 0.2] -- 0
Bộ dữ liệu 55)  [5.7 2.8 4.5 1.3] -- 1
Bộ dữ liệu 111) [6.4 2.7 5.3 1.9] -- 2
```

- Minh họa xây dựng mô hình học máy sử dụng thuật toán Decision Tree cho bài toán phân lớp hoa lan sử dụng bộ dữ liệu mẫu để huấn luyện

Thực hành 5

Thực hành 5

Yêu cầu 5.1: Sinh viên tìm hiểu tập dữ liệu mẫu nhận dạng các loại rượu (Wine recognition dataset).

- **Đọc dữ liệu từ dataset mẫu theo 2 dạng vào biến ndarray - dataframe.**
- **Hiển thị một số mẫu rượu trong tập dữ liệu**

Data Set Characteristics:

Number of Instances:	178 (50 in each of three classes)
Number of Attributes:	13 numeric, predictive attributes and the class
Attribute Information:	• Alcohol • Malic acid • Ash • Alcalinity of ash • Magnesium • Total phenols • Flavanoids • Nonflavanoid phenols • Proanthocyanins • Color intensity • Hue • OD280/OD315 of diluted wines • Proline

- **class:**
  - class_0
  - class_1
  - class_2

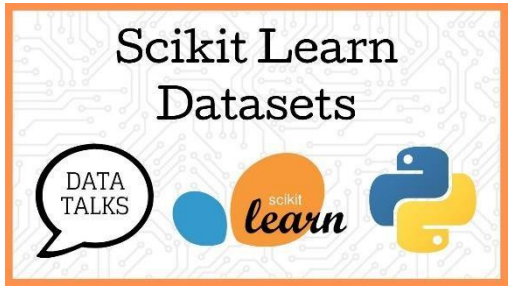
Thực hành 5

Yêu cầu 5.2: Sinh viên tìm hiểu tập dữ liệu mẫu còn lại trong bộ Datasets của thư viện Sklearn

`fetch_olivetti_faces`(* [, data_home, ...])	Load the Olivetti faces data-set from AT&T (classification).
`fetch_20newsgroups`(* [, data_home, subset, ...])	Load the filenames and data from the 20 newsgroups dataset (classification).
`fetch_20newsgroups_vectorized`(* [, subset, ...])	Load the 20 newsgroups dataset and vectorize it into token counts (classification).
`fetch_lfw_people`(* [, data_home, funneled, ...])	Load the Labeled Faces in the Wild (LFW) people dataset (classification).
`fetch_lfw_pairs`(* [, subset, data_home, ...])	Load the Labeled Faces in the Wild (LFW) pairs dataset (classification).
`fetch_covtype`(*[, data_home, ...])	Load the covertype dataset (classification).
`fetch_rcv1`(* [, data_home, subset, ...])	Load the RCV1 multilabel dataset (classification).
`fetch_kddcup99`(* [, subset, data_home, ...])	Load the kddcup99 dataset (classification).
`fetch_california_housing`(* [, data_home, ...])	Load the California housing dataset (regression).

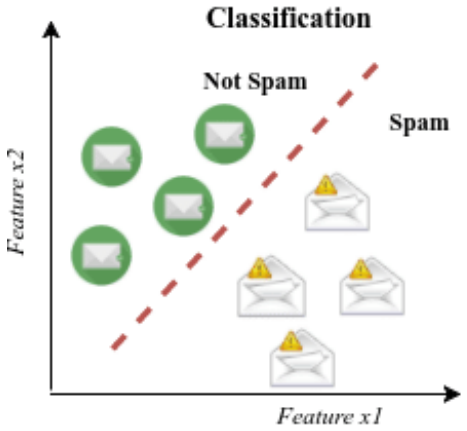
6. Tạo Simulated Dataset

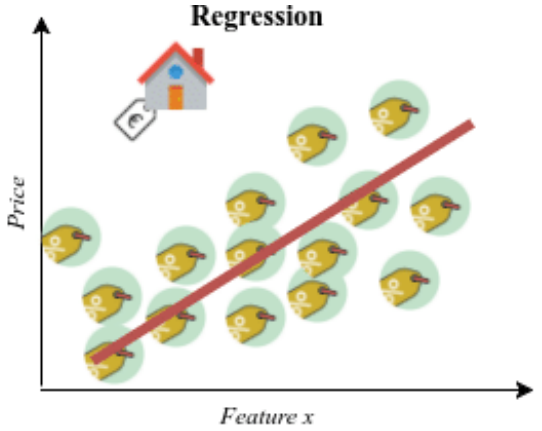
Simulated Dataset

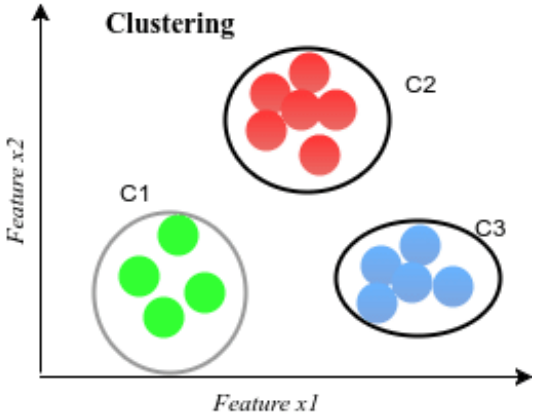


- Sklearn cho phép tạo ra các bộ dữ liệu mô phỏng theo các tham số thiết lập để chạy cho các thuật toán:

- Sklearn cho phép tạo các bộ dữ liệu cho 3 bài toán cơ bản trong học máy:
  - **Phân lớp: sklearn.datasets.make_classification**
  - **Hồi quy: sklearn.datasets.make_regression**
  - **Phân cụm: sklearn.datasets.make_blobs**







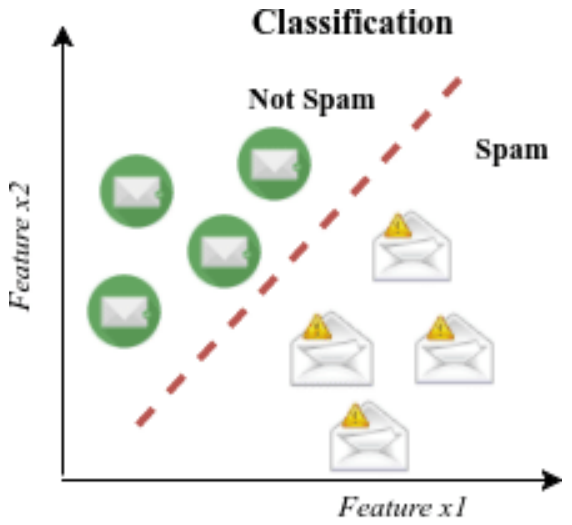
Simulated Dataset

- Tạo dữ liệu mẫu cho bài toán phân lớp:
  - **sklearn.datasets.make_classification**

```
#load module datasets trong thư viện sklearn
import sklearn.datasets as datask

#Tạo dữ liệu mẫu cho bài toán PHÂN LỚP
#Với 200 mẫu, đầu vào 5 thuộc tính, phân thành 2 lớp
X_features,y_target = datask.make_classification(n_samples=200,
                                                 n_features=5,
                                                 n_classes=2)
print(X_features[:10,:])
print(y_target[:10])
```

```
[[ 0.48748837  0.18518267 -0.16275634 -0.80735756  0.59954885]
 [ 0.35201265 -0.28948479 -0.51352122 -1.38507706 -1.26633695]
 [ 1.45849046  0.2428483  -0.28191709 -1.29481816  0.75690151]
 [-0.89887946 -0.64428871 -0.29329826 -0.15400215 -2.45431292]
 [-0.82031678  0.68686289 -0.32193918 -2.02339423  2.34453115]
 [-1.20629474 -0.16067933 -0.1907809  -0.44390306 -0.66249345]
 [ 2.14333379 -0.06069032 -0.39889965 -1.29430544 -0.39029573]
 [-0.31829191 -0.61415345 -0.29747469 -0.20848376 -2.34718624]
 [-1.02502387  0.07738804  0.40022813  1.27667389  0.45121483]
 [-0.94362914 -0.1619073  -0.40262302 -1.17250383 -0.75771508]]
[0 0 0 0 0 0 0 1 1 0]
```



Read more: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_classification.html

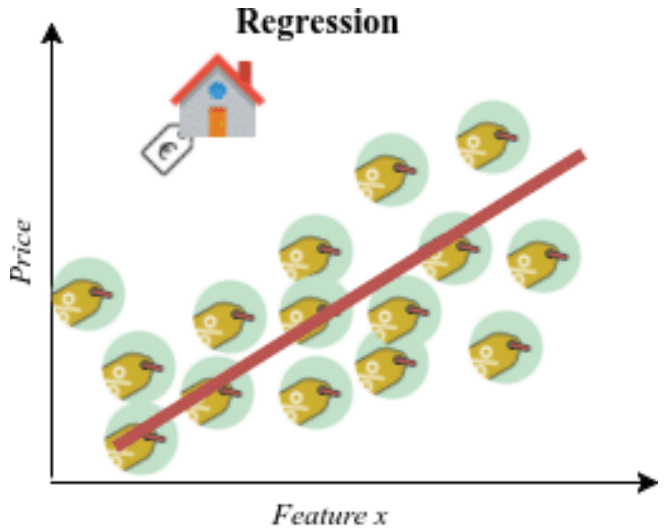
Simulated Dataset

- Tạo dữ liệu mẫu cho bài toán hồi quy:
  - **sklearn.datasets.make_regression**

```python
#load module datasets trong thư viện sklearn
import sklearn.datasets as datask

#Tạo dữ liệu mẫu cho bài toán HỒI QUY
X_features, y_target, coff= datask.make_regression(n_samples=200,
                                                   n_features=5,
                                                   n_informative=3,
                                                   n_targets=1,
                                                   noise=0.0,
                                                   coef=True,
                                                   random_state=1)
print(X_features[:10,:])
print(y_target[:10])
```

```
[[-0.36956243  0.4972691  -2.14444405  0.2373327  -1.88141087]
 [ 0.40890054 -0.03869551  1.12141771 -1.61577235 -1.31228341]
 [ 0.0607502  -0.43750898  0.92145007  0.09542509 -0.78191168]
 [-1.02188594  1.76795995  0.47761018 -0.47537288 -1.42655542]
 [ 0.6210827  -0.79726979 -0.82609743  0.28267571 -0.13597733]
 [-1.57915629  0.63019567  0.45194604 -0.4148469  -0.10534471]
 [-0.18695502 -2.277298    0.35387043 -0.06962454 -0.11644415]
 [ 0.44136444 -2.43483776  2.18697965  1.0388246  -0.31011677]
 [ 1.15528789 -0.70584051 -0.23794194  1.19268607  0.9561217 ]
 [ 0.7147896   2.13782807 -1.75592564 -0.785534    0.56438286]]
[ -42.26048415  -93.80258805  -49.39514348   19.83790815  -37.36573177
   16.42912647 -118.52143931 -105.57802562   32.55783001  107.22348747]
```



Read more: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_regression.html

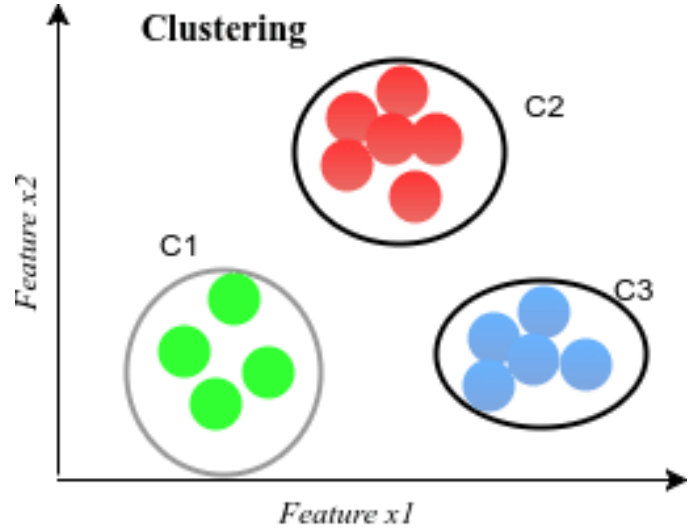
Simulated Dataset

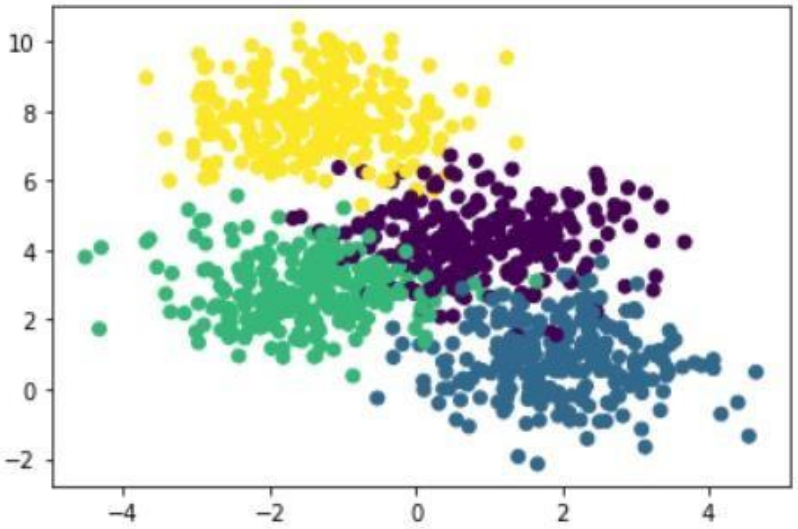
- Tạo dữ liệu mẫu cho bài toán PHÂN CỤM:
  - **sklearn.datasets.make_blobs**

```
#load module datasets trong thư viện sklearn
import sklearn.datasets as datask

#Tạo dữ liệu mẫu cho bài toán PHÂN CỤM
X_features, y_target = datask.make_blobs(n_samples=1000,
                                         centers=4,
                                         n_features=2,
                                         random_state=0)
print(X_features[:10,:])
print(y_target[:10])
```

```
[[-2.50804312  7.86408052]
 [ 0.4666179   3.86571303]
 [-2.94062621  8.56480636]
 [-2.89994656  1.85014025]
 [ 2.34438803  1.31053448]
 [ 0.87305123  4.71438583]
 [ 0.84032038  5.44067869]
 [ 1.66287852 -2.14847939]
 [ 0.85810603  3.62360912]
 [-0.72183574  4.6910678 ]]
[3 0 3 2 1 0 0 1 0 0]
```





Read more: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_blobs.html

Thực hành 6

Thực hành 6

Yêu cầu 6.1: Sinh viên tạo một tập dữ liệu mẫu cho bài toán phân lớp theo yêu cầu sau:

- **Dataset mẫu bao gồm: 1000 samples, 8 features, 2 classes.**
- **Xây dựng một mô hình ML theo theo thuật toán Decision Tree với Dataset tạo được.**

Yêu cầu 6.2: Sinh viên tạo một tập dữ liệu mẫu cho bài toán phân cụm theo yêu cầu sau:

- **1. Dataset mẫu bao gồm: 10000 samples, 2 features, 5 centers, cluster_std = 0.5.**
- **2. Trực quan hóa tập Dataset ở trên lên biểu đồ scatter.**
- **3. Thay đổi tham số cluster_std trong khoảng từ [0-1.0], nhận xét kết quả thu được**

Thank you!