ĐẠI HỌC ĐÀ NẴNG

TRƯỜNG ĐẠI HỌC BÁCH KHOA

THU THẬP VÀ LƯU TRỮ DỮ LIỆU

Chương 4

Khoa Công nghệ thông tin

Thu thập và lưu trữ dữ liệu

- Thu thập và lưu trữ là bước rất quan trọng quan trọng trước và sau khi phân tích dữ liệu
- Nội dung:
  - Đọc và ghi dữ liệu từ file
  - Thu thập dữ liệu từ web hoặc sử dụng các API
  - Tương tác với cơ sở dữ liệu

Đọc và Ghi dữ liệu dạng Text

- Đọc dữ liệu dạng bảng
  - Pandas cung cấp nhiều hàm có thể đọc các dữ liệu dạng bảng dựa vào đối tượng DataFrame

Function	Description
`read_csv`	Load delimited data from a file, URL, or file-like object; use comma as default delimiter
`read_table`	Load delimited data from a file, URL, or file-like object; use tab (`'\t'`) as default delimiter
`read_fwf`	Read data in fixed-width column format (i.e., no delimiters)
`read_clipboard`	Version of `read_table` that reads data from the clipboard; useful for converting tables from web pages
`read_excel`	Read tabular data from an Excel XLS or XLSX file
`read_hdf`	Read HDF5 files written by pandas
`read_html`	Read all tables found in the given HTML document
`read_json`	Read data from a JSON (JavaScript Object Notation) string representation
`read_msgpack`	Read pandas data encoded using the MessagePack binary format
`read_pickle`	Read an arbitrary object stored in Python pickle format
`read_sas`	Read a SAS dataset stored in one of the SAS system's custom storage formats
`read_sql`	Read the results of a SQL query (using SQLAlchemy) as a pandas DataFrame
`read_stata`	Read a dataset from Stata file format
`read_feather`	Read the Feather binary file format

Đọc và Ghi dữ liệu dạng Text

Ví dụ 1

ex1.csv

```
no,a,b,c,d,message
0,1,2,3,4,hello
1,5,6,7,8,world
2,9,10,11,12,foo
```

```
import pandas as pd
df = pd.read_csv('ex1.csv',sep=',',header=None)
print(df)
```

Kết quả

```
  0  1  2  3  4        5
0 no  a  b  c  d  message
1  0  1  2  3  4    hello
2  1  5  6  7  8    world
3  2  9 10 11 12      foo
```

Đọc và Ghi dữ liệu dạng Text

Ví dụ 2

ex2.csv

```
1,2,3,4,hello
5,6,7,8,world
9,10,11,12,foo
```

```
import pandas as pd
names = ['a', 'b', 'c', 'd', 'message']
df = pd.read_csv('ex2.csv',sep=',',names=names, index_col='message')
print(df)
```

Kết quả

```
         b  c  d
message
hello    1  2  3  4
world    5  6  7  8
foo      9 10 11 12
```

Các tham số của hàm read_csv

Argument	Description
path	String indicating filesystem location, URL, or file-like object
sep or delimiter	Character sequence or regular expression to use to split fields in each row
header	Row number to use as column names; defaults to 0 (first row), but should be None if there is no header row
index_col	Column numbers or names to use as the row index in the result; can be a single name/number or a list of them for a hierarchical index
names	List of column names for result, combine with header=None
skiprows	Number of rows at beginning of file to ignore or list of row numbers (starting from 0) to skip.
na_values	Sequence of values to replace with NA.
comment	Character(s) to split comments off the end of lines.
parse_dates	Attempt to parse data to datetime; False by default. If True, will attempt to parse all columns. Otherwise can specify a list of column numbers or name to parse. If element of list is tuple or list, will combine multiple columns together and parse to date (e.g., if date/time split across two columns).
keep_date_col	If joining columns to parse date, keep the joined columns; False by default.
converters	Dict containing column number of name mapping to functions (e.g., {'foo': f} would apply the function f to all values in the 'foo' column).
dayfirst	When parsing potentially ambiguous dates, treat as international format (e.g., 7/6/2012 -> June 7, 2012); False by default.
date_parser	Function to use to parse dates.
nrows	Number of rows to read from beginning of file.
iterator	Return a TextParser object for reading file piecemeal.
chunksize	For iteration, size of file chunks.
skip_footer	Number of lines to ignore at end of file.
verbose	Print various parser output information, like the number of missing values placed in non-numeric columns.
encoding	Text encoding for Unicode (e.g., 'utf-8' for UTF-8 encoded text).
squeeze	If the parsed data only contains one column, return a Series.
thousands	Separator for thousands (e.g., ',' or '.').

Ghi dữ liệu dạng bảng

- Sử dụng hàm to_csv()

```
import pandas as pd
df.to_csv('out.csv')
```

out.csv

```
message,a,b,c,d
hello,1,2,3,4
world,5,6,7,8
foo,9,10,11,12
```

Ghi dữ liệu thông thường

- Ta có thể sử dụng kết hợp hàm open() và phương thức write()

```
with open('ex2.csv', 'w') as writefile:
    writefile.write("1,2,3,4,hello\n");
    writefile.write("5,6,7,8,world\n");
    writefile.write("9,10,11,12,foo");
```

Dữ liệu Json

Khi ta có dữ liệu Json như sau

```
obj = """
{"name": "Wes",
"places_lived": ["United States", "Spain", "Germany"],
"pet": null,
"siblings": [{"name": "Scott", "age": 30, "pets": ["Zeus", "Zuko"]},
{"name": "Katie", "age": 38,
"pets": ["Sixes", "Stache", "Cisco"]}]
}
"""
```

Ta có thể biến đổi dữ liệu thành đối tượng

```
import json
result = json.loads(obj)
print(result)
```

hoặc ngược lại

```
asjson = json.dumps(result)
```

Thu thập dữ liệu từ web

- Web Scraping
  - Python cung cấp rất nhiều thư viện nhằm mục đích đọc dữ liệu từ các website như
    - lxml
    - Beautiful Soup
    - html5lib
  - Trong phần này giới thiệu sử dụng Beautiful Soup để thu thập dữ liệu

Web crawling với thư viện BeautifulSoup

- Web crawling là gì?
  - ***Web crawling*** là quá trình tự động trích xuất các thông tin từ các trang web và lưu trữ nó dưới một định dạng phù hợp. Chương trình mà thực hiện công việc này gọi là ***web crawler***.
  - Thông thường, khi muốn lấy một số thông tin từ các trang web, chúng ta sẽ dùng các API mà các trang đó cung cấp. Đây là cách đơn giản, tuy nhiên không phải trang web nào cũng cung cấp sẵn API cho chúng ta sử dụng. Do đó chúng ta cần một kĩ thuật để lấy các thông tin từ các trang web đó mà không thông qua API.

Thư viện BeautifulSoup

- Thư viện BeautifulSoup là một thư viện của Python cho phép chúng ta lấy dữ liệu từ HTML đơn giản và hiệu quả.
- Ta có thể sử dụng Python 3 và BeautifulSoup 4 để trích chọn dữ liệu HTML

Demo

- Từ trang chủ của vnexpress.net, hãy lấy tất cả các tiêu đề newfeed của trang đó

Demo

- Trước hết chúng ta cần lấy nội dung của trang web và parse nó như sau:

```
from bs4 import BeautifulSoup
import urllib.request

url =  'https://vnexpress.net'
page = urllib.request.urlopen(url)
soup = BeautifulSoup(page, 'html.parser')
```

Demo

- Xem code html

```
<!DOCTYPE html>
<html lang="vi" xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>VnExpress - Báo tiếng Việt nhiều người xem nhất</title> <meta name="description" content="VnExpress tin tức mới nhất - Thông tin nhanh &amp;amp; chính xác được cập nhật h
<meta name="keywords" content="VnExpress, tin tức, tin the gioi, tin nhanh, tin tuc viet nam, doc bao"/>
<meta name="news_keywords" content="VnExpress, tin tức, tin the gioi, tin nhanh, tin tuc viet nam, doc bao"/>
<meta charset="utf-8">
<meta content="width=device-width, initial-scale=1, minimum-scale=1, maximum-scale=5, user-scalable=1" name="viewport"/>
<meta http-equiv="X-UA-Compatible" content="IE=100"/>
<meta property="fb:app_id" content="1547540628876392"/>
<meta http-equiv="REFRESH" content="1800"/>
<meta name="apple-mobile-web-app-capable" content="yes"/>
<meta name="apple-mobile-web-app-title" content="Vnexpress.net"/>
<meta name="tt_article_id" content="1000000"/>
<meta name="tt_category_id" content="1000000"/>
<meta name="tt_site_id" content="1000000"/>
<meta name="tt_site_id_new" content="1000000"/>
<meta name="tt_list_folder" content="1000000"/>
<meta name="tt_page_type" content="site"/>
<meta name="tt_page_type_new" content="1"/>
<!-- add meta for pvtt3334 -->
<!-- end meta for pvtt -->
<!-- META FOR FACEBOOK -->
<meta property="og:site_name" content="vnexpress.net"/>
<meta property="og:rich_attachment" content="true"/>
<meta property="og:type" content="website"/>
<meta property="og:url" itemprop="url" content="https://vnexpress.net"/>
<meta property="og:image" itemprop="thumbnailUrl" content="https://s1.vnecdn.net/vnexpress/restruct/i/v379/logo_default.jpg"/>
<meta property="og:image:width" content="800"/>
<meta property="og:image:height" content="354"/>
```

Demo

- Phân tích

```
<h1 class="title_news">
    <a href="https://vnexpress.net/giai-tri/gia-dinh-ban-be-tien-dua-nhac-si-nguyen-trong-tao-3865800.html" title="Gia đình, bạn bè tiễn đưa nhạc sĩ Nguyễn Trọng Tạo">
    Gia đình, bạn bè tiễn đưa nhạc sĩ Nguyễn Trọng Tạo </a>
</h1>
```

  - Các tiêu đề và link nằm trong thẻ “a” nằm trong <h1 class="title_news">

Demo

- Sử dụng lệnh tìm kiếm
  - Tìm kiếm tất cả
    - findAll()
  - Tìm kiếm 1 phần tử
    - find()

Demo

- Code

```
from bs4 import BeautifulSoup
import urllib.request

url =  'https://vnexpress.net'
page = urllib.request.urlopen(url)
soup = BeautifulSoup(page, 'html.parser')
new_feeds = soup.findAll(class_='title_news')

for nfeed in new_feeds:
        feed = nfeed.find("a")
        title = feed.get('title')
        link = feed.get('href')
        if title==None or title=="" or link==None:
                continue
        print('Title: {} - Link: {}'.format(title, link))
```

Bài tập

- Thu thập trên trang vnexpress.net ít nhất 10 bài báo/1 chủ đề và lưu lại nội dung lại bằng file(mỗi bài báo 1 file).
- Gợi ý
  - Sử dụng
    - f= open(“contest1.txt","w+")
    - f.write()
    - f.close()

Tương tác với APIs

- Ta có thể sử dụng thư viện HTTP request để tương tác với các API thông qua Json

```
import requests
url = 'https://api.github.com/repos/pandas-dev/pandas/issues'
resp = requests.get(url)
result = resp.json()
print(result)
```

Kết nối CSDL

- Cơ sở dữ liệu với Python
- Python cung cấp rất nhiều thư viện để kết nối với CSDL
  - SQL Server
  - PostgreSQL
  - MySQL
  - SQLite

Thư viện sqlite3

- Ta có thể tạo 1 CSDL của SQLite như sau

```
import sqlite3
conn = sqlite3.connect('kuni.db')
print("Opened database successfully");
conn.execute('''
CREATE TABLE IF NOT EXISTS team_data(team text,
                        country text,
                        season integer,
                        total_goals integer);''')

conn.commit()
print("Table created successfully");
conn.close()
```

Thực thi truy vấn bằng hàm execute

```
conn = sqlite3.connect('kuni.db')
conn.execute("INSERT INTO team_data VALUES('Real Madrid', 'Spain', 2019, 53);")
conn.execute("INSERT INTO team_data VALUES('Barcelona', 'Spain', 2019, 47);")
conn.execute("INSERT INTO team_data VALUES('Arsenal', 'UK', 2019, 52);")
conn.execute("INSERT INTO team_data VALUES('Real Madrid', 'Spain', 2018, 49);")
conn.execute("INSERT INTO team_data VALUES('Barcelona', 'Spain', 2018, 45);")
conn.execute("INSERT INTO team_data VALUES('Arsenal', 'UK', 2018, 50 );")

conn.commit()
```

```
conn = sqlite3.connect('kuni.db')

cursor = conn.execute('SELECT * FROM team_data;')

for row in cursor:
  print(row)
conn.close()
```