AI VIET NAM – COURSE 2023

Data Manipulation and Crawling – Project

Ngày 26 tháng 5 năm 2023

Phần I: Nội dung

1. Cho file bảng dữ liệu **Advertising.csv** (bảng dữ liệu tải tại đây), các bạn hãy sử dụng Python và thư viện Pandas để đọc và thực hiện một số phép tính toán trên bảng dữ liệu này.

	TV	Radio	Newspaper	Sales
1	230.1	37.8	69.2	22.1
2	44.5	39.3	45.1	10.4
3	17.2	45.9	69.3	9.3
4	151.5	41.3	58.5	18.5
5	180.8	10.8	58.4	12.9

Hình 1: 5 hàng đầu tiên của bảng dữ liệu **Advertising.csv**

Đầu tiên, việc đọc một file .csv có thể được thực hiện bằng nhiều cách khác nhau trong Python, song ở project này ta sẽ sử dụng thư viện Pandas. Như vậy, các bạn thực hiện như sau:

(a) **Tải thư viện Pandas:** Ta tải thư viện Pandas sử dụng lệnh pip (nếu sử dụng Google Colab, các bạn không cần thực hiện bước này do Google Colab đã tải sẵn):

```
$ pip install pandas
```

Lưu ý: Dấu **$** trong một số phần code của bài viết tượng trưng cho các câu lệnh chạy trong terminal.

(b) **Import thư viện Pandas:** Để gọi được các hàm trong thư viện Pandas, các bạn cần import thư viện sử dụng lệnh sau:

```
import pandas as pd
```

Ở đây, ta sử dụng keyword `as` trong Python để tạo tên viết tắt cho Pandas nhằm thuận thiện hơn trong việc thực hiện lời gọi các hàm của thư viện này. Cụ thể, thay vì phải ghi `pandas.function()`, ta sẽ ghi thành `pd.function()`.

(c) **Đọc file Advertising.csv:** Bắt đầu thực hiện đọc bảng dữ liệu vào chương trình Python. Tại đây, ta sẽ dùng hàm `pd.read_csv()` (các bạn đọc thêm về hàm này tại đây), kết quả mà hàm này trả về sẽ là một `pd.DataFrame` (một cấu trúc dữ liệu) với nội dung tương đồng với file dữ liệu:

```
advertising_dataset_filepath = './Advertising.csv'
advertising_df = pd.read_csv(advertising_dataset_filepath, index_col=0)
```

Ở hai dòng lệnh trên:

- **Dòng 1:** Khai báo đường dẫn đến vị trí của file .csv trong máy tính. Ở ví dụ này, môi trường sử dụng là Google Colab, vì vậy vị trí mặc định của chúng ta là thư mục `/content`. Khi tải bảng dữ liệu và đặt ngay tại thư mục này, ta hoàn toàn có thể diễn tả đường dẫn file bằng `./Advertising.csv`.
- **Dòng 2:** Truyền đường dẫn file vào hàm `pd.read_csv()`, tham số `index_col=0` dùng để xác định cột làm chỉ mục bảng dữ liệu là cột đầu tiên.

(d) **Chuyển DataFrame sang List of Lists:** Hiện tại, ta sẽ không làm việc với DataFrame. Vì vậy, ta có thể chuyển sang cấu trúc dữ liệu quen thuộc hơn là List of Lists (mỗi List bên trong List lớn là 1 hàng trong bảng dữ liệu):

```
advertising_lst = advertising_df.values.tolist()
```

```
[[230.1, 37.8, 69.2, 22.1],
 [44.5, 39.3, 45.1, 10.4],
 [17.2, 45.9, 69.3, 9.3],
 [151.5, 41.3, 58.5, 18.5],
 [180.8, 10.8, 58.4, 12.9],
 [8.7, 48.9, 75.0, 7.2],
 [57.5, 32.8, 23.5, 11.8],
 [120.2, 19.6, 11.6, 13.2],
 [8.6, 2.1, 1.0, 4.8],
 [199.8, 2.6, 21.2, 10.6],
 [66.1, 5.8, 24.2, 8.6],
```

Hình 2: Kết quả sau khi chuyển đổi từ DataFrame sang List of Lists

(e) **Thực hiện một số phép tính toán trên bảng dữ liệu:** Tại đây, chúng ta sẽ tiến hành thực hiện một vài phép tính toán trên bảng dữ liệu này, bao gồm:

- **Tách cột:** Để thuận tiện tính toán cho các bước sau, ta sẽ tách các cột thành các list sử dụng **List Comprehension**:

```
tv_lst = [lst[0] for lst in advertising_lst]
radio_lst = [lst[1] for lst in advertising_lst]
newspaper_lst = [lst[2] for lst in advertising_lst]
sales_lst = [lst[3] for lst in advertising_lst]
```

- **Tổng của các cột:**

```
tv_sum = sum(tv_lst)
radio_sum = sum(radio_lst)
newspaper_sum = sum(newspaper_lst)
sales_sum = sum(sales_lst)
```

- **Tổng của các cột tính từ hàng thứ 2 đến hàng thứ 10:** Để lấy ra một tập các hàng trong khoảng nào đó, ta dùng kỹ thuật **List Slicing** (**Lưu ý:** Quy ước số thứ tự hàng trong bảng dữ liệu tính từ 1 nhưng chỉ mục trong list sẽ tính từ 0). Như vậy:

```
start_idx = 1
end_idx = 10

tv_sum = sum([n for n in tv_lst[start_idx:end_idx]])
radio_sum = sum([n for n in radio_lst[start_idx:end_idx]])
newspaper_sum = sum([n for n in newspaper_lst[start_idx:end_idx]])
sales_sum = sum([n for n in sales_lst[start_idx:end_idx]])
```

- **Min, Max của các cột:**

```
tv_min, tv_max = min(tv_lst), max(tv_lst)
radio_min, radio_max = min(radio_lst), max(radio_lst)
newspaper_min, newspaper_max = min(newspaper_lst), max(newspaper_lst)
sales_min, sales_max = min(sales_lst), max(sales_lst)
```

- **Giá trị trung bình của các cột:**

```
tv_avg = sum(tv_lst) / len(tv_lst)
radio_avg = sum(radio_lst) / len(radio_lst)
newspaper_avg = sum(newspaper_lst) / len(newspaper_lst)
sales_avg = sum(sales_lst) / len(sales_lst)
```

- **Giá trị trung vị của các cột:**

```
def median(lst):
    sorted_lst = sorted(lst)

    n = len(lst)
    mid = n // 2
    if n % 2 == 0:
        y = (sorted_lst[mid] + sorted_lst[mid - 1]) / 2
    else:
        y = sorted_lst[mid]

    return y

tv_median = median(tv_lst)
radio_median = median(radio_lst)
newspaper_median = median(newspaper_lst)
sales_median = median(sales_lst)
```

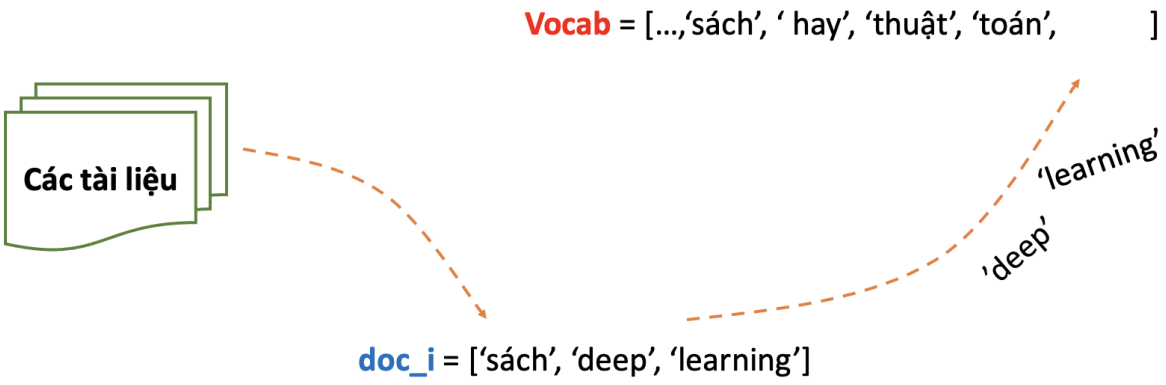
- **Giả sử ngân sách phân bổ cho TV, Radio và Newspaper là 100%, tính tỉ lệ phần trăm phân bổ cho các cột trên:**

```
total_budget = sum(tv_lst) + sum(radio_lst) + sum(newspaper_lst)

tv_percentage = (sum(tv_lst) / total_budget) * 100
radio_percentage = (sum(radio_lst) / total_budget) * 100
newspaper_percentage = (sum(newspaper_lst) / total_budget) * 100
```

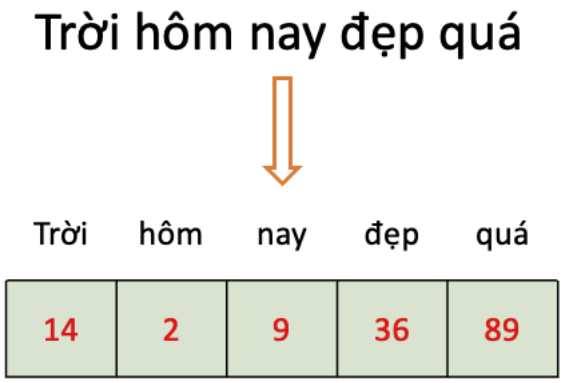
2. Xây dựng hàm biểu diễn văn bản (Text Representation) sử dụng kỹ thuật **Index-based Encoding**. Các bước thực hiện kỹ thuật này bao gồm:

   (a) **Xây dựng bộ từ vựng (Dictionary):** Từ kho ngữ liệu (tập hợp các văn bản), lọc ra toàn bộ các từ mới có trong toàn bộ văn bản để tạo lập thành một bộ từ điển.



Hình 3: Xây dựng bộ từ vựng từ một tập các văn bản.

(b) **Xây dựng hàm biểu diễn văn bản:** Dựa vào bộ từ vựng, ta sẽ biểu diễn lại đoạn string (văn bản) bất kì thành một list các số nguyên. Các số này là chỉ mục (index) của từ tương ứng trong bộ từ vựng.



Hình 4: Hàm biểu diễn văn bản. Ở đây, các từ trong văn bản được đổi thành chỉ mục tương ứng trong bộ từ vựng.

Như vậy, để cài đặt kỹ thuật biểu diễn văn bản này, ta sẽ code theo hướng như sau:

(a) **Xây dựng hàm chuẩn hóa văn bản (Text Normalization):** Để giảm kích thước của bộ từ vựng, ta sẽ giảm đi độ phức tạp của văn bản đầu vào bằng cách chuẩn hóa chúng:

```
import string

remove_characters = '\t' + string.punctuation
def text_normalize(text):
    text = text.lower()
    text = text.strip()
    text = text.replace('\n', ' ')
    for char in remove_characters:
        text = text.replace(char, '')

    return text
```

Trong đó:

- **Dòng 1:** Import thư viện **string** là một module có sẵn trong Python, cung cấp một số tiện ích cho việc tương tác với kiểu dữ liệu string.
- **Dòng 3:** Tạo danh sách các kí tự sẽ loại bỏ ra khỏi văn bản. Danh sách bao gồm: Toàn bộ các dấu câu (punctuations) và kí tự tab (\t).
- **Dòng 4:** Khai báo hàm tên **text_normalize** nhận tham số đầu vào **text** là một string (văn bản).
- **Dòng 5:** Chuyển văn bản sang chữ viết thường.
- **Dòng 6:** Xóa các khoảng trắng ở rìa văn bản.
- **Dòng 7:** Thay thế kí tự xuống hàng (\n) bằng khoảng trắng.
- **Dòng 8, 9:** Xóa toàn bộ các kí tự trong danh sách các từ **remove_characters**.
- **Dòng 11:** Trả về văn bản đã được chuẩn hóa.

(b) **Xây dựng hàm tạo bộ từ vựng:** Dựa vào danh sách các văn bản (string), tạo bộ từ vựng bằng cách thu thập toàn bộ các từ (không lấy lại từ đã có) trong toàn bộ các văn bản:

```
1  def create_dictionary(corpus):
2      dictionary = []
3      for paragraph in corpus:
4          paragraph = text_normalize(paragraph)
5          tokens = paragraph.split()
6          for token in tokens:
7              if token not in dictionary:
8                  dictionary.append(token)
9
10     return dictionary
```

Trong đó:

- **Dòng 1:** Khai báo hàm có tên **create_dictionary**, nhận đầu vào tham số **corpus** là một list các string tượng trưng cho danh sách các văn bản.
- **Dòng 2:** Khai báo một list dùng để chứa các từ.
- **Dòng 3, 4:** Duyệt qua các văn bản và chuẩn hóa chúng.
- **Dòng 5:** Tách văn bản thành list các từ (còn được gọi là các token) thông qua việc tách theo khoảng trắng dùng phương thức `split()`.
- **Dòng 6, 7, 8:** Duyệt qua các token, nếu token đang xét chưa nằm trong bộ từ vựng thì thêm vào dùng phương thức `append()`.
- **Dòng 10:** Trả về bộ từ vựng.

(c) **Xây dựng hàm biểu diễn văn bản:** Dựa vào bộ từ vựng, ta sẽ xây dựng được hàm biểu diễn văn bản thành list các số nguyên:

```
1  def vectorize(text, dictionary, unknown_token_id):
2      text = text_normalize(text)
3      tokens = text.split()
4      vector = [
5          dictionary.index(token) \
6          if token in dictionary else unknown_token_id \
7              for token in tokens
8      ]
9
10     return vector
```

Trong đó:

- **Dòng 1:** Khai báo hàm có tên **vectorize** nhận tham số đầu vào bao gồm:
  - **text:** Là một string (văn bản).

  - **dictionary:** Là một list các từ (bộ từ vựng).
  - **unknown_token_id:** Là một số nguyên đại diện cho kí tự không tồn tại trong bộ từ vựng.
- **Dòng 2, 3:** Chuẩn hóa văn bản đầu vào và tách thành list các từ (tokens).
- **Dòng 4, 5, 6, 7, 8:** Chuyển đổi các từ trong văn bản **text** thành các số nguyên sử dụng chỉ mục tương ứng của chúng trong **dictionary** dùng phương thức **index()**. Nếu từ đang xét không nằm trong **dictionary** thì đổi từ đó thành **unknown_token_id**.
- **Dòng 10:** Trả về list các số nguyên.

Như vậy, các hàm quan trọng đã được xây dựng. Ta có thể dùng đoạn code sau để kiểm tra:

```
paragraph = 'Hello World. This is a text representation (vectorization) example.'
corpus = [paragraph]
dictionary = create_dictionary(corpus)
text = 'Hello World. My name is AI.'
unknown_token_id = len(dictionary)
vectorize(text, dictionary, unknown_token_id)
```

Các bạn thử xem văn bản **text** trong đoạn code trên sẽ được biểu diễn thành một list có các giá trị như thế nào?

3. **Data Crawling** là một kỹ thuật tự động thu thập dữ liệu từ các trang web. Thông qua việc đọc và trích xuất thông tin từ file HTML của trang web (Ngôn ngữ đánh dấu siêu văn bản - file xây dựng cấu trúc các thành phần của một trang web), ta hoàn toàn có thể thu thập được các thông tin, dữ liệu mong muốn.

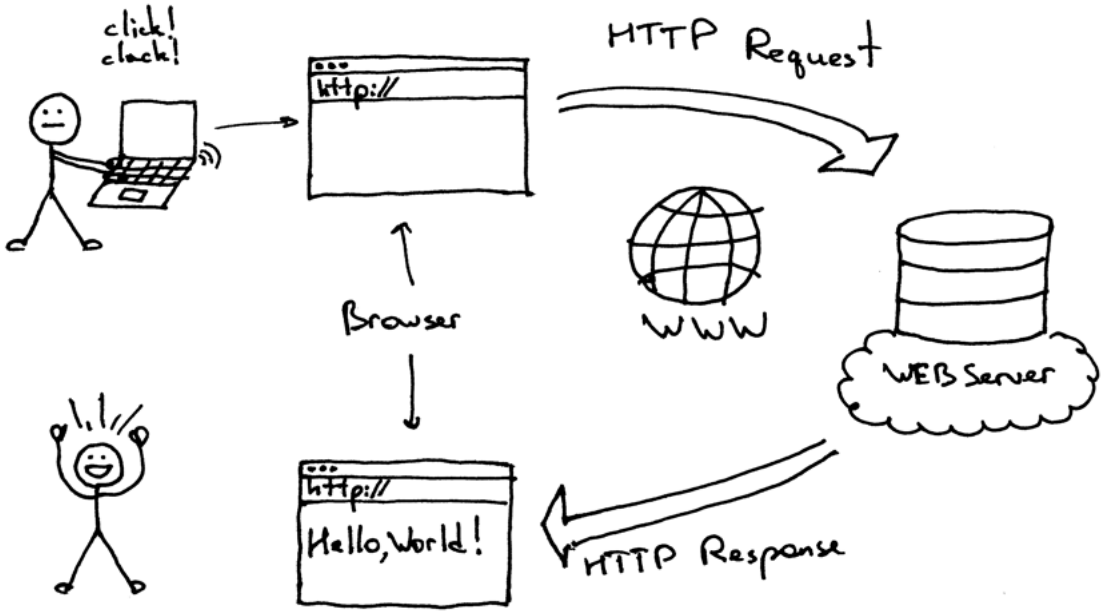
Hình 5: Tổng quan về Data Crawling dưới dạng sơ đồ.

Có rất nhiều thư viện trong Python hỗ trợ tương tác với các website, song ở project này sẽ sử dụng thư viện **selenium**.

Hình 6: Logo của thư viện selenium.

Trong project này, ta sẽ xây dựng một chương trình thu thập các bài báo cùng thuộc một chủ đề nào đó trên trang vietnamnet.vn. Tổng quan nội dung các bước thực hiện bao gồm:

- **Bước 1:** Dựa vào đường dẫn URL của một bài báo, request đến server của trang web nội dung HTML của trang.
- **Bước 2:** Với nội dung HTML nhận được, xác định và tiến hành trích xuất các nội dung của trang báo sử dụng Selenium.
- **Bước 3:** Lưu nội dung của trang báo dưới dạng file văn bản .txt.
- **Bước 4:** Lặp lại 3 bước trên với những trang báo khác.



Hình 7: Ảnh minh họa quá trình truy cập và lấy nội dung HTML của một trang web. **Nguồn:** link.

Phần II: Trắc nghiệm

1. Để đọc một file .csv sử dụng thư viện pandas, ta sử dụng hàm dưới đây?

   (*a*) `pd.read_csv()`
   (*b*) `pd.read_hsv()`
   (*c*) `pd.read_tsv()`
   (*d*) `pd.read_bsv()`

2. Cho `table_lst` là một list of lists. Để lấy ra danh sách các list con từ list thứ `i` đến list thứ `j` cần sử dụng lệnh nào sau đây?

   (*a*) `table_lst[:]`
   (*b*) `table_lst[i:]`
   (*c*) `table_lst[i:j]`
   (*d*) `table_lst[:j]`

3. Cho `table_lst` là một list of lists. Để lấy danh sách chứa tổng của mỗi list con trong `table_lst` cần sử dụng lệnh nào dưới đây?

   (*a*) `[min(lst) for lst in table_lst]`
   (*b*) `[max(lst) for lst in table_lst]`
   (*c*) `[sum(lst) for lst in table_lst]`
   (*d*) `[len(lst) for lst in table_lst]`

4. Trong bảng dữ liệu **Advertising.csv**, tổng của cột **Radio** tính từ hàng thứ 10 đến hàng thứ 20 là (lấy phần nguyên)?

   (*a*) 275
   (*b*) 276
   (*c*) 277
   (*d*) 278

5. Trong bảng dữ liệu **Advertising.csv**, giá trị trung vị của cột **Sales** tính từ hàng thứ 30 đến hàng thứ 45 là?

   (*a*) 13.5
   (*b*) 13.6
   (*c*) 13.7
   (*d*) 13.8

6. Trong phương pháp biểu diễn văn bản Index-based encoding, các từ sẽ được biểu diễn bằng?

   (*a*) Tần suất xuất hiện trong văn bản.
   (*b*) Chỉ mục tương ứng trong bộ từ vựng.
   (*c*) Giá trị đối chiếu định nghĩa thủ công.
   (*d*) Sử dụng công thức toán học riêng.

7. Trong phương pháp biểu diễn văn bản Index-based encoding, khi gặp một từ không tồn tại trong bộ từ vựng thì phải làm thế nào?

   (*a*) Bỏ qua từ này.
   (*b*) Xóa từ này ra khỏi văn bản.
   (*c*) Copy giá trị của từ trước đó.
   (*d*) Sử dụng một số đặc biệt tự định nghĩa.

8. Lý do nào sau đây không phải lợi ích khi thực hiện chuẩn hóa văn bản trong việc biểu diễn văn bản?

(a) Giảm độ phức tạp của văn bản.
(b) Giảm kích thước của bộ từ vựng.
(c) Tăng độ dài của list biểu diễn.
(d) Đồng nhất hóa cấu trúc văn bản.

9. Cho đoạn code sau:

```python
def text_normalize(text):
    text = text.upper()
    text = text.replace(',', '')
    text = text.replace('.', '')

    return text

text = 'Hello, what is your name?'
print(text_normalize(text))
```

Kết quả in ra màn hình sau khi thực thi đoạn code trên là?

(a) `HELLO WHAT IS YOUR NAME?`
(b) `HELLO WHAT IS YOUR NAME`
(c) `hello what is your name?`
(d) `hello what is your name`

10. Cho đoạn code sau:

```python
corpus = ['']
dictionary = create_dictionary(corpus)
text = 'Hello World'
print(vectorize(text, dictionary, 0))
```

Giả sử với hàm `create_dictionary()` và `vectorize()` trong phần mô tả. Kết quả in ra màn hình sau khi thực thi đoạn code trên là?

(a) `[0, 0]`
(b) `[0, 14]`
(c) `[14, 2]`
(d) `[14, 0]`

11. Cho đoạn code sau:

```python
def create_dictionary(corpus):
    dictionary = []
    for paragraph in corpus:
        tokens = paragraph.split()
        for token in tokens:
            if token not in dictionary:
                dictionary.append(token)

    return dictionary

corpus = [
    'Hello, how are you?',
    'Hello World',
]
dictionary = create_dictionary(corpus)
text = 'hello, are you ok?'
print(vectorize(text, dictionary, 0))
```

Giả sử với hàm `vectorize()` trong phần mô tả không sử dụng normalization. Kết quả in ra màn hình sau khi thực thi đoạn code trên là?

(a) `[0, 2, 3, 0]`

(b) `[0, 2, 0, 0]`

(c) `[0, 0, 0, 0]`

(d) `[0, 3, 2, 0]`

12. Trong kỹ thuật Data Crawling, để có thể trích xuất nội dung hiển thị trong một trang web, ta cần dựa vào điều gì?

(a) Địa chỉ IP của trang web.

(b) Đường dẫn URL của trang web.

(c) File HTML của trang web.

(d) Không xác định.

- Hết -