

## PROJECT 01

# Thu thập dữ liệu từ Web

### 1. Thông tin thành viên

20424008  
20424013

Dương Mạnh Cường  
Phạm Nguyễn Mỹ Diễm

### 2. Phân công và kế hoạch thực hiện

STT	Công việc	Thành viên thực hiện	Thời gian thực hiện (Deadline: 4 tuần)	Hoàn thành (%)
1	Xác định chủ đề cần thu thập.	Cường – Diễm	Tuần 1	100
2	Thu thập dữ liệu theo đúng chủ đề từ ít nhất hai website trở lên.	Cường	Tuần 1	100
3	Tạo cấu trúc phù hợp để lưu trữ dữ liệu.	Cường	Tuần 1	100
4	Thiết kế crawler.	Diễm	Tuần 2 – 3	100
5	Tránh các vấn đề xảy ra đối với crawler.	Cường – Diễm	Tuần 2 – 3	100
6	Thể hiện sự phức tạp của crawler như lấy dữ liệu từ trang web động, lấy qua API, độc lập với cấu trúc của website, cho phép người dùng chọn chủ đề, ...	Cường – Diễm	Tuần 2 – 3	100
7	Đo tốc độ của crawler dựa trên thống kê số trang (webpage) xử lý trong đơn vị thời gian, kích thước dữ liệu lấy được theo chủ đề, thời gian lấy, ...	Cường	Tuần 2 – 3	100
8	Đánh giá dữ liệu thu được như thể hiện các thống kê về dữ liệu lấy như kích thước, loại, đặc trưng, phân phối, tính đa dạng của dữ liệu, ...	Diễm	Tuần 2 – 3	100
9	Tiến hành phân tích dữ liệu.	Cường – Diễm	Tuần 4	100
10	Run chương trình và viết báo cáo	Cường – Diễm	Tuần 4	100

### 3. Báo cáo theo các mục ở phần 3

- **Tiêu chí 1:** Trình bày chủ đề, lý do chọn chủ đề, trang web lấy hoặc trang web là hạt giống.
  - + Trong thương mại điện tử, việc liên tục nâng cao chất lượng sản phẩm và dịch vụ để đáp ứng nhu cầu khách hàng nhằm nâng cao uy tín là công việc hàng đầu của các doanh nghiệp khi tham gia sàn thương mại điện tử.

- + Hệ thống hỗ trợ doanh nghiệp phân loại các phản hồi của khách hàng thành hai nhóm: **positive** [nhóm khách hàng tích cực], kí hiệu  $\oplus$  và **negative** [nhóm khách hàng tiêu cực], kí hiệu  $\ominus$  dựa trên dữ liệu đã vào dưới dạng **text** [tài liệu văn bản].
  - + Hệ thống được xây dựng dựa trên lịch sử những đánh giá của khách hàng đã có trước đó, dữ liệu được thu thập từ phần **comment** [bình luận] và **rank** [số sao (điểm) đánh giá cho sản phẩm] của khách hàng ở trang web thương mại điện tử từ một nhóm ngành nào đó.
  - + Hệ thống giúp doanh nghiệp có thể biết được những phản hồi nhanh chóng của khách hàng về sản phẩm, dịch vụ của họ, điều này giúp cho doanh nghiệp có thể hiểu được tình hình kinh doanh, hiểu được ý kiến của khách hàng từ đó giúp doanh nghiệp cải thiện hơn trong dịch vụ, sản phẩm.
  - + Ở đây, trang thương mại điện tử được dùng để crawl dữ liệu là **Shopee Việt Nam**, dữ liệu được crawl về là những bình luận và đánh giá của khách hàng về các sản phẩm thuộc nhóm ngành thời trang.
- **Tiêu chí 2:** Mô tả thuật toán, cấu trúc mã nguồn, các thành phần hệ thống.

#### Mô tả thuật toán và các thành phần của hệ thống:

- + Đối với quá trình data pre-processing sẽ được trình bày chi tiết trong phần **Tiêu chí 6** và **Tiêu chí 7** ở file “**01.pre-processing\_references.ipynb**”, ở đây ta chỉ tập trung vào cách ta crawl dữ liệu.
- + Shopee là một dynamic website, điều này có nghĩa các component của trang sẽ được load lên “khi có sự tương tác” của người dùng, ta cứ hình dung như trang newfeeds của Facebook, ban đầu chỉ hiện một vài bài viết sau đó khi ta scroll để xem hết bài viết thì qua cơ chế AJAX nó sẽ load thêm các bài mới viết mới. Điều này giúp cho giảm tải về dung lượng mạng và thời gian waiting cho người dùng nhưng nó gián tiếp khiến cho việc crawl data khó thực hiện hơn.
- + Toàn bộ thuật toán và chi tiết cách thực hiện nằm trong phần sau.

#### Cấu trúc mã nguồn:

- + Folder “**modules**” chứa các user defined function, class.
- + Folder “**data**” chứa các data mà ta crawl về, những data phát sinh sau bước pre-processing,...
- + Folder “**images**” chứa các hình minh họa.
- + File “**00.intro\_scraping.ipynb**”: là các phần **Tiêu chí 1** đến **Tiêu chí 5** của báo cáo.
- + File “**01.pre-processing\_references.ipynb**”: chứa hai phần **Tiêu chí 6** và **Tiêu chí 7** của báo cáo.
- + File “**modules/crawler.py**”: định nghĩa các hàm dùng để crawl data.

- + File “**modules/processor.py**”: định nghĩa các hàm dùng để tiễn xử lý dữ liệu.
  - + File “**modules/regex\_patterns.py**”: định nghĩa các regular expression để tiễn xử lý dữ liệu.
  - + File “**modules/user\_object\_defined.py**”: định nghĩa các kiểu dữ liệu để tiện cho quá trình code, cho code clear và dễ hiểu.
  - + File “**modules/utils.py**”: chứa các hàm như đọc file txt, ghi file, các hàm chức năng,...
  - + Folder “**modules/dependencies**”: chứa các file data dùng làm sạch dữ liệu, trong đó:
    - “**abbreviate.txt**”: chứa các từ viết tắt và dạng chuẩn của từ viết tắt.
    - “**stopwords.txt**”: các stopword tiếng việt.
    - “**vocabulary.txt**”: các từ đơn trong tiếng việt.
- **Tiêu chí 3:** Các vấn đề xảy ra đối với crawler và phương pháp xử lý.

Vấn đề	Phương pháp xử lí
Hầu hết các trang web thương mại điện tử hiện tại là dynamic website, chúng tiến hành load dữ liệu bằng AJAX. Và để crawl được dữ liệu từ các trang web như thế này thì đòi hỏi ta cần phải giả lập thao tác người dùng.	<ul style="list-style-type: none"><li>- Sử dụng <b>Selenium</b> để tiến hành giả lập thao tác người dùng.</li><li>- Các trang web thương mại điện tử ngày nay có các API để hỗ trợ các lập trình viên có thể nhanh chóng crawl được dữ liệu.</li></ul>
Vấn đề về đường truy cập internet hiện tại đang bị hỏng khiến việc crawl data trở nên lâu và vất vả hơn.	Chưa có biện pháp xử lí.

- **Tiêu chí 4:** Các tính năng phức tạp của crawler.
- + Giả lập cuộn trang.
  - + Cuộn trang và kiểm tra sự xuất hiện của một css selector được chỉ định.
  - + Giả lập click button.
  - + Crawl data bằng API.
    - + Linh hoạt trong thời gian timeout, thay vì ta sử dụng cơ chế là bắt crawler dừng tĩnh trong 3 giây, vậy nếu như ta đã access vào trang thành công trước 3 giây thì ta vẫn phải chờ cho hết 3 giây timeout. Cái ta muốn là tự động kết thúc timeout ngay khi ta truy cập vào trang thành công hoặc hết timeout 3 giây, lúc này ta sử dụng cơ chế **WebDriverWait(<browser driver>, <second waiting>).until()** của Selenium.
    - + Kiểm tra đã kết thúc navigation hay chưa.

- + Các xử lí phức tạp trong tiềnlà xử lí dữ liệu như regex, xử lí noise sample, extracting emoji sẽ được trình bày chi tiết trong phần **Tiêu chí 6** và **Tiêu chí 7**.
- **Tiêu chí 5:** Đánh giá hiệu năng crawler.
  - + Crawler nhìn chung hoạt động tốt, không bị hiện tượng treo do cơ chế **WebDriverWait().until()** đã khắc phục điều này.
  - + Tuy nhiên thời gian crawl lâu, mất từ 3 đến 5 ngày treo máy tính.
- **Tiêu chí 6 - 7:** Mô tả và đánh giá dữ liệu thu thập được & Tiềnlà xử lý dữ liệu thu thập.
  - + Load toàn bộ review vào một *dataframe* duy nhất.

```
In [1]: 1 %load_ext autoreload
2 %autoreload 2

In [2]: 1 import modules.utils as Utils
2 import modules.processor as Processor
3 import numpy as np
4 import pandas as pd
5 import enchant
6 import random
7
8 from sklearn.utils import shuffle

In [3]: 1 # Lấy tất cả các directory path của các lần ta tiến hành crawl data
2 dir_paths = Utils.getAllFolderPath("./data/product_reviews/")
3
4 dir_paths

Out[3]: ['./data/product_reviews/product_reviews_01/',
         './data/product_reviews/product_reviews_02/',
         './data/product_reviews/product_reviews_03/',
         './data/product_reviews/product_reviews_00/']

In [4]: 1 # Đọc toàn bộ các review từ các file csv
2 reviews = Utils.readReviews(dir_paths)

In [5]: 1 reviews.head()

Out[5]:
   raw_comment  rating
0  Minh mua size L. Cổ tay siêu bé, như size S ấy...      1
1  Size S M L XL\n\nForm áo cực kì dễ mang, thiết...      5
2  Áo khá đẹp vừa với dáng giao hàng cực kì nhanh...      5
3  Đẹp rất hài lòng okokokokokokokokokoko...      5
4  Đẹp, ôm dáng, mặc đẹp lắm mà form nhỏ. Mình 60...      5

In [6]: 1 print("Tập dữ liệu có {} bình luận.".format(reviews.shape[0]))
Tập dữ liệu có 278159 bình luận.
```



+ Đếm tần số xuất hiện của từng rating.

```
In [7]: 1 reviews['rating'].value_counts()  
  
Out[7]: 5    260555  
        4    9646  
        3    4194  
        1    2308  
        2    1456  
Name: rating, dtype: int64
```

#### Nhận xét:

- Nhìn chung tuy ta crawl được hơn 200,000 quan sát nhưng có sự chênh lệch lớn giữa các rating.
- Nhìn qua ta thấy đa phần là các rating được đánh giá 5 sao, điều này cũng dễ hiểu vì hệ thống recommend của Shopee sẽ ưu tiên gợi ý cho khách hàng những sản phẩm có đánh giá tốt. Và sẽ hạn chế hoặc thậm chí là không gợi ý các mặt hàng bị đánh giá kém. Nên với địa vị là người đi trộm dữ liệu như chúng ta thì không có cách nào khắc phục điều này.
- Nay giờ, do ta cần chỉ ra hai lớp là negative và positive nên những comment mà  $rating \geq 4$  sẽ được cho vào nhóm positive, ngược lại là nhóm negative.

+ Tiến hành label cho *reviews* với các giá  $rating < 4$  sẽ thuộc nhóm *negative* còn lại là nhóm *positive*.

```
In [8]: 1 reviews = Utils.labelRating(reviews)  
  
In [9]: 1 reviews.head()  
  
Out[9]:  
          raw_comment  rating  label  
0      Minh mua size L. Cổ tay siêu bé, như size S ấy...    1    0  
1      Size S M L XL\n\nForm áo cực kì dễ mang, thiết...    5    1  
2      Áo khá đẹp vừa với dáng giao hàng cực kì nhanh...    5    1  
3      Đẹp rất hài lòng okokokokokokokokokoko...    5    1  
4      Đẹp, ôm dáng, mặc đẹp lắm mà form nhỏ. Mình 60...
```

+ Bây giờ ta sẽ xóa đi feature *rating* vì về sau ta sẽ không cần dùng đến nó nữa, và tiến hành đếm số lượng quan sát của từng nhóm trên *label*.

```
In [10]: reviews = reviews.drop(columns=['rating'])
2
3 Processor.printAfterProcess(reviews)
4 reviews.head()

Shape: (278159, 2)
1    270201
0     7958
Name: label, dtype: int64

Out[10]:
```

	raw_comment	label
0	Mình mua size L. Cổ tay siêu bé, như size S ấy...	0
1	Size S M L XL\n\nForm áo cực kì dễ mang, thiết...	1
2	Áo khá đẹp vừa với dáng giao hàng cực kì nhanh...	1
3	Đẹp rất hài lòng okokokokokokokokokoko...	1
4	Đẹp, ôm dáng, mặc đẹp lắm mà form nhò. Mình 60...	1

+ Một trong những vấn đề đầu tiên và tối quan trọng khi xử lý với dữ liệu văn bản là kiểm tra xem liệu text *a* có cùng cách biểu diễn với text *b* hay không.

+ Một ví dụ dễ hiểu là giả sử ta có biến  $a$  = ‘đẹp’ và biến  $b$  = ‘đẹp’, nhưng khi ta compare hai biến này  $a == b$  thì kết quả sẽ ra False, nguyên nhân là do chúng sử dụng mã hóa unicode khác nhau, có thể  $a$  dùng unicode-8 và  $b$  dùng unicode-16.

+ Vậy điều đầu tiên ta cần làm là phải đưa tất cả các text về cùng một chuẩn duy nhất, ta có thể làm điều này bằng cách sử dụng *unicodedata.normalize()* từ package chuẩn *unicodedata* của Python. (tham khảo thêm tại đây: (<https://www.kite.com/python/docs/unicodedata.normalize>)).

+ Dưới đây là ví dụ cho trường hợp này:

```
In [11]: 1 import unicodedata
2
3 a = 'đẹp, rất hài lòng'
4 b = 'đẹp, rất hài lòng'
5
6 a == b

Out[11]: False

In [12]: 1 ''' Sử dụng chuẩn NFD '''
2 a = unicodedata.normalize('NFD', a)
3 b = unicodedata.normalize('NFD', b)
4
5 print("String a: {}, Type a: {}".format(a, type(a)))
6 print("String b: {}, Type b: {}".format(b, type(b)))

String a: đẹp, rất hài lòng, Type a: <class 'str'>
String b: đẹp, rất hài lòng, Type b: <class 'str'>

In [13]: 1 a == b

Out[13]: True
```

+ Bây giờ, ta sẽ tạo một feature có tên là *normalize\_comment*, trải qua 2 bước:

> *lower()* cho text.

> Chuẩn hóa bằng `unicodedata.normalize()`

```
In [14]: reviews['normalize_comment'] = reviews['raw_comment'].apply(lambda cmt: Processor.normalizeComment(cmt))
2 reviews.head()
3
```

Out[14]:

	raw_comment	label	normalize_comment
0	Minh mua size L. Cổ tay siêu bé, như size S ấy...	0	minh mua size l. cổ tay siêu bé, như siz...
1	Size S M L XL\n\nForm áo cực kì dễ mang, thiết...	1	size s m l xl\n\nform áo cực kì dễ mang,...
2	Áo khá đẹp vừa với dáng giao hàng cực kì nhanh...	1	áo khá đẹp vừa với dáng giao hàng cư...
3	Đẹp rất hài lòng okokokokokokokokokoko...	1	đẹp rất hài lòng okokokokokokokokoko...
4	Đẹp, ôm dáng, mặc đẹp lắm mà form nhò. Minh 60...	1	đẹp, ôm dáng, mặc đẹp lắm mà form nh...

+ Ta cũng sẽ chuẩn hóa cho cột *raw\_comment* nhưng không *lower()* chúng.

```
In [15]: 1 reviews['raw_comment'] = reviews['raw_comment'].apply(lambda cmt: Processor.normalizeComment(cmt, False))
2
3 reviews.head()

Out[15]:   raw_comment    label      normalize_comment
0  Minh mua size L. Cố tay siêu bé, như siz...  0  minh mua size l. cố tay siêu bé, như siz...
1  Size S M L XL\n\nForm áo cực kì dễ mang,...  1  size s m l xl\n\nform áo cực kì dễ mang,...
2  Áo khá đẹp vừa với dáng giao hàng cur...  1  áo khá đẹp vừa với dáng giao hàng cur...
3  Đẹp rất hài lòng okokokokokokokokokok...  1  đẹp rất hài lòng okokokokokokokokokok...
4  Đẹp, ôm dáng, mặc đẹp lắm mà form nho...  1  đẹp, ôm dáng, mặc đẹp lắm mà form nho...
```

+ Chúng ta biết rằng, các comment của các sản phẩm đôi khi sẽ chứa các URL do người bán hàng chèn vào để giúp khách hàng có thể click vào để xem các mặt hàng khác, chúng là các noise sample mà ta cần phải loại bỏ khỏi dataset của chúng ta.

+ Hình dưới đây là kết quả cho ra khi ta thử search cụm từ `http` thì nó cho ra hơn 600 mẫu dữ liệu chứa URL. Ta cần loại bỏ các mẫu này.

```
In [16]: 1 reviews['contain_url'] = reviews['normalize_comment'].apply(lambda cmt: Processor.containsURL(cmt))
2
3 Processor.printAfterProcess(reviews, 'contain_url')
4 reviews.head()
```

Shape: (278159, 4)  
0 277589  
1 570  
Name: contain\_url, dtype: int64

	raw_comment	label	normalize_comment	contain_url
0	Minh mua size L. Cổ tay siêu bé, như siz...	0	mình mua size l. cổ tay siêu bé, như siz...	0
1	Size S M L XL\n\nForm áo cực kì dễ mang,...	1	size s m l xl\n\nform áo cực kì dễ mang,...	0
2	Áo khá đẹp vừa với dáng giao hàng cur...	1	áo khá đẹp vừa với dáng giao hàng cur...	0
3	Đẹp rất hời lòng okokokokokokokokok...	1	đẹp rất hời lòng okokokokokokokokok...	0
4	Đẹp, ôm dáng, mặc đẹp lắm mà form nho...	1	đẹp, ôm dáng, mặc đẹp lắm mà form nho...	0

+ Bây giờ chúng ta chỉ sẽ lấy các comment mà không chứa URL.

```
In [17]: 1 reviews = reviews[reviews['contain_url'] == 0]
2 reviews = reviews.drop(columns=['contain_url']).reset_index(drop=True) # xóa cột `contain_url`
3
4 Processor.printAfterProcess(reviews)
5 reviews.head()
```

Shape: (277589, 3)  
1 269633  
0 7956  
Name: label, dtype: int64

	raw_comment	label	normalize_comment
0	Minh mua size L. Cổ tay siêu bé, như siz...	0	mình mua size l. cổ tay siêu bé, như siz...
1	Size S M L XL\n\nForm áo cực kì dễ mang,...	1	size s m l xl\n\nform áo cực kì dễ mang,...
2	Áo khá đẹp vừa với dáng giao hàng cur...	1	áo khá đẹp vừa với dáng giao hàng cur...
3	Đẹp rất hời lòng okokokokokokokok...	1	đẹp rất hời lòng okokokokokokokok...
4	Đẹp, ôm dáng, mặc đẹp lắm mà form nho...	1	đẹp, ôm dáng, mặc đẹp lắm mà form nho...

**Nhận xét:** Đa phần là các bình luận thuộc nhóm positive sẽ chứa các URL, cũng dễ hiểu vì họ quảng cáo mà.

+ Tiếp theo, ta cũng cần xóa các comment mà chứa IN HOA chiếm quá 50% độ dài comment, các comment này khả năng cao cũng là quảng cáo, vì người bán họ muốn làm nổi bật bình luận này lên so với các bình luận còn lại.

```
In [18]: 1 reviews['contain_adv'] = reviews['raw_comment'].apply(lambda cmt: Processor.containsAdvertisement(cmt))
2
3 Processor.printAfterProcess(reviews, 'contain_adv')
4 reviews.head()
```

Shape: (277589, 4)  
0 270936  
1 6653  
Name: contain\_adv, dtype: int64

	raw_comment	label	normalize_comment	contain_adv
0	Minh mua size L. Cổ tay siêu bé, như siz...	0	mình mua size l. cổ tay siêu bé, như siz...	0
1	Size S M L XL\n\nForm áo cực kì dễ mang,...	1	size s m l xl\n\nform áo cực kì dễ mang,...	0
2	Áo khá đẹp vừa với dáng giao hàng cur...	1	áo khá đẹp vừa với dáng giao hàng cur...	0
3	Đẹp rất hời lòng okokokokokokokok...	1	đẹp rất hời lòng okokokokokokokok...	0
4	Đẹp, ôm dáng, mặc đẹp lắm mà form nho...	1	đẹp, ôm dáng, mặc đẹp lắm mà form nho...	0

**Nhận xét:** Các mẫu tiêm năng chứa quảng cáo khá cao, lên đến hơn 3000 sample. Ta có thể xóa chúng.

```
In [19]: reviews = reviews[reviews['contain_adv'] == 0]
reviews = reviews.drop(columns=['contain_adv']).reset_index(drop=True) # xóa cột `contain_adv`
Processor.printAfterProcess(reviews)
reviews.head()

Shape: (270936, 3)
1    263084
0    7852
Name: label, dtype: int64

Out[19]:   raw_comment  label  normalize_comment
0  Minh mua size L. Cổ tay siêu bé, như siz...  0  minh mua size l. cổ tay siêu bé, như siz...
1  Size S M L XL\n\nForm áo cực kì dễ mang,...  1  size s m l xl\nform áo cực kì dễ mang,...
2  Áo khá đẹp vừa với dáng giao hàng cư...  1  áo khá đẹp vừa với dáng giao hàng cư...
3  Đẹp rất hài lòng okokokokokokokokokok...  1  đẹp rất hài lòng okokokokokokokokokok...
4  Đẹp, ôm dáng, mặc đẹp lắm mà form nho...  1  đẹp, ôm dáng, mặc đẹp lắm mà form nho...
```

**Nhận xét:** Lại một lần nữa các comment có khả năng cao là quảng cáo này lại đa phần là thuộc nhóm positive.

+ Nhìn qua các comment, ta sẽ thấy có các comment chứa *emoji* như hình dưới đây:

```
10554 Chắc xài dc may lan deo,1
10555 "Sản phẩm trên cả tuyệt vời"
10556 Cảm ơn shop nhiều ạ 😊😊😊😊
10557 Nên mua nhà ♥
10558 Đẹp lắm ạ",1
10559 "Đông hô`đẹp xuất sắc lun iiiii, dây hơi cứng thoi nha,gia rẻ mà chất lượng n
10560 ói,1
10561 Rất ok,1
10562 Tốt,1
10563 "Đẹp lắm luôn á :((, hơi nhỏ mà nhìn sang trọng lắm nha, phù hợp với giá tiề
10564 Uy tín nha 🌟 vira đé vừa đẹp :v tiề n nào của đây ♥,1
10565 "Giao nhanh, hàng dù, đồ`xinh, shop thân thiện, sẽ tiếp tục ủng hộ vào lần s
10566 xinh lắm mọi người ơi. rất đáng muaaaa ♥♥♥,1
10567 Thời gian giao hàng nhanh,1
10568 Đông hô`đẹp xin xin,1
10569 Ok lắm ạ,1
10570 Đây đông hô`cứng k dc đẹp bị hỏng nữa 😞😞😞,0
10571 Đóng hô`đẹp lắm nha anh shipper đê`thương nứa,1
10572 "Siêu đẹp, nên mua, có giây hướng dẫn nữa mọi người ơi",1
10573 Đẹp nha,1
10574 "Shop phục vụ rất kém, không nên mua.",0
10575 🙄,1
10576 Xin lỗi luôn ạ. Mấy bạn mà da đen như em thì chọn màu tối tối 1 chút là đượ
10577 Đê`thương,1
10578 Đóng hô`đẹp đáng giá tiề,1
10579 "Hàng đẹp như hình ,chạy tốt, dáng tiềnh lắm ,cảm ơn shop",1
10580 "Hàng đúng như trong hình
10581 Giao hàng nhanh",1
10582 "Sản phẩm đẹp, đúng màu",1
10583 Hàng rất vừa ý .Giao hàng cũng ko lâu .{:}{:}{:}{:},0
10584 "Sản phẩm tuyệt vời
10585 Nên mua",1
10586 "Sp chất lượng,giao hàng cảm nhận,nhanh chóng,phù hợp gia tien
10587 Ggfgucgh ucjj hhvjkf hgchjh ghghj ghhv bh hgghjvvhhjhbbjhbjujbvjhbgghhu
10588 đẹp nha,1
10589 Gssjsjhsshdjdjsjdbdbxbdbdhchjfjdfkffkfjcnzn A sbbdbbdbdbdhehwwhhhsgwvv
10590 Tốt nhưng vì đóng hô`màn hình đt của tôi đã vỡ tan tành 😞😞,1
10591 --,1
```

+ Đây là ‘vốn quý’ góp phần làm tăng sức mạnh cho model, nếu ta thực hiện bước loại bỏ các kí tự đặc biệt trước khi ta tách các *emoji* ra, thì ta đã vô tình xóa luôn các *emojis* này, vì các *emojis* thực chất được xây dựng dựa trên các kí tự đặc biệt.

+ Như hình trên, rõ ràng ta thấy được emoji góp phần ta hiểu được một comment là *positive* hay *negative*.

+ Ta sẽ sử dụng một gói của python là *emojis*: pip3 install *emojis*

gói này sẽ giúp ta tách các *emojis* ra khỏi bình luận.

+ Ta sẽ chứa toàn bộ *emojis* của một comment qua cột tương ứng là *emojis*.

```
In [20]: reviews['emoji'] = reviews['raw_comment'].apply(lambda cmt: Processor.extractEmoji(cmt))
```

	raw_comment	label	normalize_comment	emoji
0	Mình mua size L. Cổ tay siêu bé, như size...	0	mình mua size l. cổ tay siêu bé, như size...	
1	Size S M XL\n\nForm áo cực kì dễ mang....	1	size s m xl\n\nform áo cực kì dễ mang....	
2	Áo khá đẹp vừa với dáng giao hàng cù...	1	áo khá đẹp vừa với dáng giao hàng cù...	
3	Đẹp rất hài lòng okokokokokokokokok...	1	đẹp rất hài lòng okokokokokokokokok...	
4	Đẹp, ôm dáng, mặc đẹp lắm mà form nho...	1	đẹp, ôm dáng, mặc đẹp lắm mà form nho...	

+ Tiếp theo, ta sẽ loại bỏ dấu câu, ký tự đặc biệt.

```
[22]: reviews['normalize_comment'] = reviews['normalize_comment'].apply(lambda cmt: Processor.removeSpecialLetters(cmt.lower()))
reviews.head()
```

	raw_comment	label	normalize_comment	emoji
0	Mình mua size L. Cổ tay siêu bé, như size...	0	mình mua size l cổ tay siêu bé như size ...	
1	Size S M L XL\nForm áo cực kì dễ mang,...	1	size s m l xl form áo cực kì dễ mang thi...	
2	Áo khá đẹp vừa với dáng giao hàng cư...	1	áo khá đẹp vừa với dáng giao hàng cư...	
3	Đẹp rất hài lòng okokokokokokokokok...	1	đẹp rất hài lòng okokokokokokokok...	
4	Đen ôm dáng mặc đèn lồng mà form phô	1	đen ôm dáng mặc đèn lồng mà form phô	

+ Tiếp theo, ta cần chuẩn lại các từ bị duplicate như: chòiiiiiii oiyyyyy, xinhhhhhh quá, đẹp xíuuuuuuuuuuuu thành chòi oi, xinh quá, đẹp xiu.

+ Tuy nhiên có một vấn đề xảy ra, giả sử trong comment có các từ tiếng anh như “feedback”, thì nó sẽ thành “fedback”, nên ta sẽ thực hiện bước này ở phần sau:

```
reviews['normalize_comment'] = reviews['normalize_comment'].apply(lambda cmt: Processor.removeDuplicateLetters(cmt))
```

+ Tiếp theo, chúng ta sẽ chuẩn lại một vài từ viết tắt cơ bản.

+ File “**modules/dependencies/abbreviate.txt**” chứa các từ viết tắt cơ bản mà giới trẻ hay dùng comment, ta có thể bổ sung theo thời gian.

```
[23]: # xây dựng dictionary cho các từ viết tắt
abbreviate = Utils.buildDictionaryFromFile("./modules/dependencies/abbreviate.txt")

# test
abbreviate['okela']

[23]: 'ok'

[24]: reviews['normalize_comment'] = reviews['normalize_comment'].apply(lambda cmt: Processor.replaceWithDictionary(cmt, abbreviate))

reviews.head()

[24]:   raw_comment  label      normalize_comment  emoji
0   Minh mua size L. Cỗ tay siêu bé, như siz...  0       minh mua size l cỗ tay siêu bé như size ...
1   Size S M L XL\n\nForm áo cực kì dễ mang,...  1       size s minh l xl form áo cực kì dễ mang...
2   Áo khá đẹp vừa với dáng giao hàng cư...  1       áo khá đẹp vừa với dáng giao hàng cư...
3   Đẹp rất hời lòng okokokokokokokokokok...  1       đẹp rất hời lòng okokokokokokokokok...
4   Đẹp, ôm dáng, mặc đẹp lắm mà form nho...  1       đẹp ôm dáng mặc đẹp lắm mà form nhò ...
```

+ Bây giờ ta sẽ tiến hành xóa các từ vô nghĩa trong comment, ví dụ như hình dưới đây:

```
10710  "Hàng đẹp nhà mình, chất lượng tốt, giao hàng nhanh, giá cả rất chỉn chu. Vừa mua
10711  Cũng đẹp mà giao tới hết pin rồi,1
10712  "Đông hô`đê~thương, mang nhẹ nhàng, hy vọng bền.",1
10713  Đẹp i hình sě ứng hô dài dài.1
10714  Dndndkkdkdkddkdkddkkdkdmmmdmdmddmnddnndndnddj 1
10715  Sản phẩm đẹp đúng như hình shop phục vụ rất tốt mìn mua hai lân r vân rất ưng giao
10716  "👉 Xét cho cùng, muôn gia đình hp thì cả vk và ck đều phải biết yêu thương nhau v
10717
10718  😊 Là vợ nêu ck than thở bận rộn, mệt mỏi không có thời gian chăm sóc mình thì hãy
10719  Đẹp,1
10720  .1
10721  "Chất lượng sản phẩm tốt, giao hàng nhanh, đóng gói chắc chắn.",1
10722  Đã nhận dc hàng cái thứ 3 trong tháng. Đã nghe thư! Âm thanh khá ổn (nếu đừng nghe
10723  sản phẩm okee lầm ạ,1
10724  Đẹp,1
10725  "Cứ nghĩ giá thành rẻ sp không ok. Nhưng lại ok quá mức cho phép. Mình bt rất ngại
10726  "Chất lượng hàng tốt , giao nhanh, nhưng cái màu hông thì cộng dây ko biêt bị gì n
10727  "hàng đẹp ,chất lượng tốt ,... ",1
10728  Đóng hô`vào nuoc gio có đôi đc ko shop,1
10729  dây kim phứt nó hì lồng đeo lên nó cứ quay vòng vòng sao sài đc shop,1
10730  Dycgokbgolbdszhoibhkklmbcssayokbxkpmfzsujxgokcxhkvxgmkvxkfvxkcknxdkvxgkn,1
10731  Sản phẩm túi giá rẻ,1
10732  "Rất ưng ý
10733  Về`giá cả lân sản phẩm",1
```

nhiều bình luận này mặc dù thuộc lớp positive nhưng nó là các noise sample, có thể các comment này dùng để comment cho có để nhận shopee xu khi đánh giá sản phẩm.

+ Kế tiếp, ta nên xóa các sample mà khả năng cao không là tiếng việt, vì sao ta làm bước này, đơn giản thôi đây là shopee vietnam, và các comment cõ ý bằng tiếng anh, tiếng hàn, tiếng trung của các “thánh làm màu” sẽ là các noise sample khiến model ta bị giảm hiệu năng.

+ Nhưng làm sao ta có thể thực hiện điều này, cách đơn giản nhất là ta có thể sử dụng các package như **textblob**, **googletrans**,... các package này chứa các function giúp ta detect language cho text, tuy nhiên hạn chế là chúng chỉ cho tối đa khoảng 200 request một ngày thôi, và số mẫu của chúng ta hiện tại là quá lớn. Ở đây ta có file “**modules/dependencies/vocabulary.txt**” chứa hơn 17000 từ đơn phổ biến của tiếng việt.

+ Vậy cách đơn giản hơn là ta có thể xây dựng một dictionary chứa các từ đơn của tiếng việt, với mỗi comment, nếu số lượng từ không tìm thấy trong dictionary này lớn hơn số từ được tìm thấy trong dictionary thì khả năng cao đây là một comment làm màu.

+ Tuy nhiên, vẫn có một vài từ tiếng anh mà ta cần giữ lại như shipper, ta sẽ sử dụng package **enchant** để check một từ có phải là từ tiếng anh hay không.

`pip3 install pyenchant`

+ Ở các bước phía trên, ta đã đề cập đến việc xóa các từ bị duplicate kí tự, ta sẽ thực hiện nó ở trong bước này.

```
[25]: # hơn 17 ngàn từ đơn trong tiếng việt
vocabularies = Utils.buildDictionaryFromFile('./modules/dependencies/vocabulary.txt', True)
english_voca = enchant.Dict('en_US') # english if a word is english
```

```
[26]: reviews['normalize_comment'] = reviews['normalize_comment'].apply(lambda cmt: Processor.removeNoiseWord(cmt, vocabularies, english_voca))
reviews.head()
```

	raw_comment	label	normalize_comment	emoji
0	Mình mua size L. Cổ tay siêu bé, như size...	0	mình mua size l cổ tay siêu bé như size ...	
1	Size S M L XL\nForm áo cực kì dễ mang,...	1	size s m l xl form áo cực kì dễ mang...	
2	Áo khá đẹp vừa với dáng giao hàng cư...	1	áo khá đẹp vừa với dáng giao hàng cư...	
3	Đẹp rất hời lồng okokokokokokokokok...	1	đẹp rất hời lồng	
4	Đẹp, ôm dáng, mặc đẹp lắm mà form nh...	1	đẹp ôm dáng mặc đẹp lắm mà form nh ...	

+ Tiếp theo ta sẽ remove stopword, chúng ta sẽ sử dụng stopword trong file “**modules/dependencies/stopwords.txt**”. Ta không nên sử dụng các stopword được build sẵn trên mạng nhất là cho tiếng việt, vì chưa chắc các từ này đã hợp với dữ liệu hiện tại của chúng ta.

+ Ví dụ như stopword set loại bỏ từ “**nhung**”, tuy nhiên từ này khả năng cao là quan trọng, giả sử ta có câu này: “shop giao hàng chậm **nhung** giao đúng hàng, ủng hộ shop”, thì nhờ từ “**nhung**” này mà model ta có khả năng phân biệt được nó là positive hay negative.

+ Ngoài ra với một file txt như vậy, ta có thể bổ sung stopword sau này.

```
[27]: stopwords = Utils.buildListFromFile("./modules/dependencies/stopwords.txt")  
  
[28]: reviews['normalize_comment'] = reviews['normalize_comment'].apply(lambda cmt: Processor.removeStopwords(cmt, stopwords))  
reviews.head()
```

```
[28]:
```

	raw_comment	label	normalize_comment	emoji
0	Mình mua size L. Cổ tay siêu bé, như siz...	0	mua size l cổ tay siêu bé như size s ấy...	
1	Size S M L XL\n\nForm áo cực kì dễ mang,...	1	size s l xl form áo cực kì dễ mang thiê...	
2	Áo khá đẹp vừa với dáng giao hàng cư...	1	áo khá đẹp vừa dáng giao hàng cực kì...	
3	Đẹp rất hài lòng okokokokokokokokok...	1	đẹp rất hài lòng	
4	Đẹp, ôm dáng, mặc đẹp lắm mà form nho...	1	đẹp ôm dáng mặc đẹp lắm mà form nhò ...	

+ Tiếp theo, ta loại bỏ các empty và duplicate “normalize\_comment”.

```
[29]: reviews = Processor.removeEmptyOrDuplicateComment(reviews)  
  
Processor.printAfterProcess(reviews)  
reviews.head()
```

Shape: (216044, 4)

1 208950

0 7094

Name: label, dtype: int64

```
[29]:
```

	raw_comment	label	normalize_comment	emoji
0	Mình mua size L. Cổ tay siêu bé, như siz...	0	mua size l cổ tay siêu bé như size s ấy...	
1	Size S M L XL\n\nForm áo cực kì dễ mang,...	1	size s l xl form áo cực kì dễ mang thiê...	
2	Áo khá đẹp vừa với dáng giao hàng cư...	1	áo khá đẹp vừa dáng giao hàng cực kì...	
3	Đẹp rất hài lòng okokokokokokokok...	1	đẹp rất hài lòng	
4	Đẹp, ôm dáng, mặc đẹp lắm mà form nho...	1	đẹp ôm dáng mặc đẹp lắm mà form nhò ...	

+ Train test split, ta thấy rằng giữa hai nhóm positive và negative có chênh lệnh lớn, nên tập train data của ta sẽ bằng **0.8 \* min(size(positive), size(negative)) \* 2**

```
[30]: half_min_size = min(reviews['label'].value_counts())  
  
half_min_size
```

7094

```
[31]: reviews_positive = reviews[reviews['label'] == 1]  
reviews_negative = reviews[reviews['label'] == 0]  
  
reviews_positive = shuffle(reviews_positive)  
reviews_positive = reviews_positive.reset_index(drop=True)
```

```
[32]: positive_index = random.sample(range(0, reviews_positive.shape[0]), half_min_size)  
  
positive_index[:10]
```

[32]: [164968, 5741, 44157, 200197, 98770, 56108, 204221, 135907, 74696, 26040]



```
[33]: reviews_positive2 = reviews_positive.iloc[positive_index,:]  
reviews_positive2.head()
```

	raw_comment	label	normalize_comment	emoji
164968	Có áo mầu ko giống nhg vẫn đẹp	1	có áo mầu không giống nhưng vẫn đẹp	
5741	Tóc đẹp lắm ai	1	tóc đẹp lắm ai	
44157	Áo đẹp lắm shop đường may chắc chắn...	1	áo đẹp lắm đường may chắc chắn khô...	
200197	Áo cute lắm mọi người, mặc ôm vào n...	1	áo cute lắm mọi người mặc ôm vào ng...	
98770	Khá là oke mà mink mua trúng đợt k có t...	1	khá là mà mink mua trúng đợt không có ...	😂

+ Đây là tập data mà hai nhóm positive và negative cân bằng nhau.

```
[34]: normalize_reviews = pd.concat([reviews_negative, reviews_positive2], axis=0)  
normalize_reviews = normalize_reviews.reset_index(drop=True)  
  
Processor.printAfterProcess(normalize_reviews)  
normalize_reviews.head()
```

	raw_comment	label	normalize_comment	emoji
0	Minh mua size L. Cổ tay siêu bé, như siz...	0	mua size l cổ tay siêu bé như size s ấy...	
1	bị chật liên hệ. shop để đổi lại sh...	0	chật liên hệ đổi không nghe máy nhá...	
2	hàng 1 lớp, chất vài k ok, sz S mà ngư...	0	hang lớp chất vài không size s mà ngư...	
3	Tiền nào của nấy, thất vọng	0	tiền nào của nấy thất vọng	
4	Màu của giày quá là khác luôn	0	màu của giày quá là khác luôn	

+ Ghi ra file.

```
[35]: normalize_reviews.to_csv("./data/normalize_reviews.csv", index=False)
```

+ Bây giờ ta sẽ ghi phàn bù còn lại của “review\_positive” vào file, ta có thể dùng nó cho việc evaluate model sau này.

```
[36]: reviews_positive3 = reviews_positive[~reviews_positive.index.isin(positive_index)]
reviews_positive3 = reviews_positive3.reset_index(drop=True)

reviews_positive3.head()
```

	raw_comment	label	normalize_comment	emoji
0	Áo xinh giã man ^^❤ shop đóng gói kĩ	1	áo xinh giã man đóng gói kĩ	❤
1	Shop đóng hàng rất đẹp. Giao hàng siêu ...	1	đóng hàng rất đẹp giao hàng siêu nhanh ...	😊
2	Sản phẩm hợp lý với giá tiền.	1	sản phẩm hợp lý giá tiền	
3	Giao hàng nhanh, đóng gói đẹp và chắc ch...	1	giao hàng nhanh đóng gói đẹp chắc chắn...	
4	Sản phẩm chất lượng tốt, vài mát, ...	1	sản phẩm chất lượng tốt vài mát đe...	

```
[37]: reviews_positive3.to_csv("./data/complement_positive_reviews.csv", index=False)
```

- **Tiêu chí 8:** Báo cáo rõ ràng các mục đã thực hiện, có thể hiện mức độ hoàn thiện của từng công việc.

STT	Tiêu chí	Hoàn thành (%)
1	Trình bày chủ đề, lý do chọn chủ đề, trang web lấy hoặc trang web là hạt giống.	100
2	Mô tả thuật toán, cấu trúc mã nguồn, các thành phần hệ thống.	100
3	Các vấn đề xảy ra đối với crawler và phương pháp xử lý.	100
4	Các tính năng phức tạp của crawler.	100
5	Đánh giá hiệu năng crawler.	100
6	Mô tả và đánh giá dữ liệu thu thập được.	100
7	Tỉ lệ xử lý dữ liệu thu thập.	100
8	Báo cáo rõ ràng các mục đã thực hiện, có thể hiện mức độ hoàn thiện của từng công việc.	100

## 4. Hướng dẫn cách thức biên dịch và chạy chương trình

### 4.1 Environment, editor (IDE), programming language & dependent packages

- **OS:** Ubuntu 20.04 LTS, macOS Big Sur Version 11.5.2
- **Programming language:** Python 3.6.8
- **Python's dependent packages:**
  - pip3 install selenium==3.141.0
  - pip3 install emojis==0.6.0
  - pip3 install pyenchant==3.2.1
  - pip3 install numpy==1.19.5
  - pip3 install pandas==1.2.4



- pip3 install requests==**2.22.0**
- pip3 install scikit-learn==**0.24.2**

Một vài package khác có thể sẽ yêu cầu cài thêm trong quá trình install các package phía trên nên sẽ không liệt kê trong đây.

- **Editor:** VS-Code (*recommend*), Jupyter Notebook, Jupyter Lab,...
- **Download Firefox drivers** (chọn file phù hợp với OS máy) tại link:

<https://selenium-python.readthedocs.io/installation.html>

## 1.5. Drivers

Selenium requires a driver to interface with the chosen browser. Firefox, for example, requires [geckodriver](#), which needs to be installed before the below examples can be run. Make sure it's in your *PATH*, e. g., place it in */usr/bin* or */usr/local/bin*.

Failure to observe this step will give you an error *selenium.common.exceptions.WebDriverException: Message: 'geckodriver' executable needs to be in PATH*.

Other supported browsers will have their own drivers available. Links to some of the more popular browser drivers follow.

<b>Chrome:</b>	<a href="https://sites.google.com/a/chromium.org/chromedriver/downloads">https://sites.google.com/a/chromium.org/chromedriver/downloads</a>
<b>Edge:</b>	<a href="https://developer.microsoft.com/en-us/microsoft-edge/tools/webdriver/">https://developer.microsoft.com/en-us/microsoft-edge/tools/webdriver/</a>
<b>Firefox:</b>	<a href="https://github.com/mozilla/geckodriver/releases">https://github.com/mozilla/geckodriver/releases</a>
<b>Safari:</b>	<a href="https://webkit.org/blog/6900/webdriver-support-in-safari-10/">https://webkit.org/blog/6900/webdriver-support-in-safari-10/</a>

For more information about driver installation, please refer the [official documentation](#).

- + **Đối với OS Window:** Đặt file geckodiver cùng với file notebook, chứa trong thư mục “Source”.
- + **Đối với macOS:** Đặt file geckodiver ở tại đường dẫn “**/usr/local/bin**”
- + **Đối với OS Ubuntu:** Đặt file geckodiver ở tại đường dẫn “**/usr/local/bin**”

geckodriver-v0.29.1-linux32.tar.gz	2.73 MB
geckodriver-v0.29.1-linux32.tar.gz.asc	833 Bytes
geckodriver-v0.29.1-linux64.tar.gz	2.59 MB
geckodriver-v0.29.1-linux64.tar.gz.asc	833 Bytes
geckodriver-v0.29.1-macos-aarch64.tar.gz	1.67 MB
geckodriver-v0.29.1-macos.tar.gz	1.81 MB
geckodriver-v0.29.1-win32.zip	1.37 MB
geckodriver-v0.29.1-win64.zip	1.44 MB
Source code (zip)	
Source code (tar.gz)	

## 4.2 Demo và chi tiết về cách thực hiện

- Khi truy cập vào trang chủ Shopee Việt Nam tại địa chỉ <https://shopee.vn>, khi kéo xuống một chút ta sẽ thấy được nhóm ngành thời trang như dưới đây:

The screenshot shows the Shopee Vietnam homepage. At the top, there's a navigation bar with links for 'Thông Báo', 'Hỗ Trợ', 'Đăng Ký', and 'Đăng Nhập'. Below the navigation is the Shopee logo and a search bar. A banner at the top features various products and brands like 'DA KHỎE SẠCH...', 'DELI VIETNAM', and 'SÁCH AZ GIẢM ...'. The main content area is titled 'DANH MỤC' and displays a grid of categories. The categories 'Thời Trang Nam' and 'Thời Trang Nữ' are highlighted with red boxes. Other categories shown include Điện Thoại & Phụ Kiện, Thiết Bị Điện Tử, Máy Tính & Laptop, Máy Ảnh & Máy Quay Phim, Đồng Hồ, Giày Dép Nam, Thiết Bị Điện Gia Dụng, Thể Thao & Du Lịch, Ô Tô & Xe Máy & Xe Đạp, Mẹ & Bé, Nhà Cửa & Đời Sống, Sắc Đẹp, Sức Khỏe, Giày Dép Nữ, Túi Ví Nữ, Phụ Kiện & Trang Sức Nữ, Bách Hóa Online, and Nhà Sách Online. Below the categories, there's a 'FLASH SALE' section featuring various products from brands like innisfree with discounts ranging from -50% to -30%.

- Vùng red square là những nhóm hàng mà ta sẽ tập trung crawl cũng như xây dựng model về sau.
- Một câu hỏi đặt ra là tại sao chúng ta không tạo ra một model mà nó có thể phân lớp cho toàn bộ tất cả các nhóm ngành trên trang thương mại điện tử này. Có một vài hạn chế như sau:
  - + Việc chúng ta có gắng nhất nhết toàn bộ các comment của các nhóm ngành khác nhau và bắt máy tính phải học một đồng này sẽ khiến cho quá trình học trở nên phức tạp, khó khăn và tốn thời gian, đồng thời nếu có xây dựng được model thì chất lượng nó cũng sẽ không tốt khi ta evaluate nó hoặc ứng dụng vào thực tế về sau.
  - + Các nhóm ngành khác nhau có những keyword khác nhau, ví dụ nhóm ngành thời trang sẽ có những keyword điển hình như: *vải xấu, áo mỏng, đố lông,...* Nhưng nếu trong nhóm ngành điện tử sẽ có những keyword như: *máy nóng, sạc không vô, chai pin,...*, nhưng giữa hai nhóm ngành thời trang và điện tử lại có những keyword chung như: *hàng không giống ảnh, giao sai màu, giao hàng chậm,...* và điển hình ở các comment tích cực thì việc các keyword này overlap lên nhau thì càng nhiều hơn, ví dụ: *giao nhanh, sản phẩm tốt, chất lượng sản phẩm tuyệt vời,...* Các comment tích cực hay có một xu hướng chung chung như vậy và không đề cập quá chi tiết về nhóm hàng mình đang đánh giá.
  - + Và nếu ta muốn một hệ thống có thể ứng dụng được trên toàn bộ hệ thống các nhóm hàng, thì lúc này ta có thể làm như sau:
    - Giả sử ta là Shopee, thì ta biết rõ comment này thuộc sản phẩm nào và sản phẩm này thuộc nhóm hàng nào dựa vào các label, tag của sản phẩm, từ đó ta sẽ sử dụng model tương ứng cho nhóm hàng này để dự đoán.
    - Nếu ta không là Shopee, ta có thể xây dựng thêm một model-1 với input là comment của khách hàng, output là nhóm hàng mà comment này khả năng cao thuộc về. Sau đó ta mới bắt đầu đưa comment này vào model-2 tương ứng với nhóm hàng mà model-1 đã xuất và đánh giá comment này. Đây là một vài cách mà ta có thể ứng dụng. Thực tế thì các hệ thống này có khả năng cao phức tạp hơn nhiều, nhưng ở đây ta chỉ chú tâm vào nhóm hàng thời trang thôi.
  - + Giả sử ta cần crawl data từ nhóm **Thời Trang Nam**, ta có thể click vào nó:

The screenshot shows the Shopee Vietnam website. At the top, there's a navigation bar with links like 'Kênh Người Bán', 'Trở thành Người bán Shopee', 'Tải ứng dụng', 'Kết nối', 'Thông Báo', 'Hỗ Trợ', 'Đăng Ký', and 'Đăng Nhập'. Below the header is the Shopee logo and a search bar. A banner at the top features several products with their prices: Váy (đ139.000), Áo Phòng (đ99.000), Bóng Tẩy Trang (đ160.000). To the right of the banner are logos for 'DA KHỎE SẠCH...', 'DELI VIETNAM', and 'SÁCH AZ GIẢM...'. Below the banner is a 'DANH MỤC' (Category) grid with two rows of icons. The first row includes 'Thời Trang Nam' (highlighted with a red box), 'Điện Thoại & Phụ Kiện', 'Thiết Bị Điện Tử', 'Máy Tính & Laptop', 'Máy Ảnh & Máy Quay Phim', 'Đồng Hồ', 'Giày Dép Nam', 'Thiết Bị Điện Gia Dụng', 'Thể Thao & Du Lịch', and 'Ô Tô & Xe Máy & Xe Đạp'. The second row includes 'Thời Trang Nữ', 'Mẹ & Bé', 'Nhà Cửa & Đời Sống', 'Sắc Đẹp', 'Sức Khỏe', 'Giày Dép Nữ', 'Túi Ví Nữ', 'Phụ Kiện & Trang Sức Nữ', 'Bách Hóa Online', and 'Nhà Sách Online'. Below the category grid is a 'FLASH SALE' banner featuring various products from brands like innisfree with discounts ranging from 20% to 30%. A button 'Xem Tất Cả >' is visible on the right side of the banner.

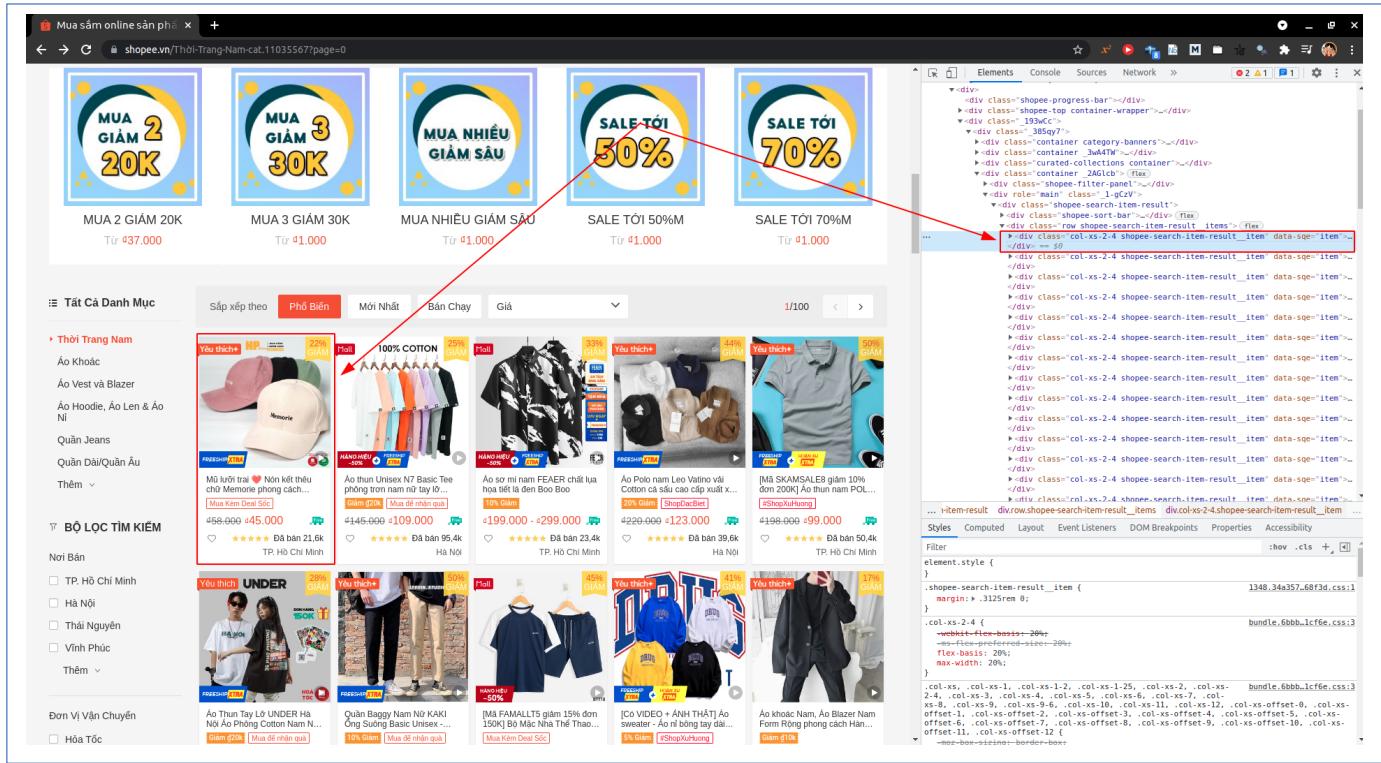
+ Hãy chú ý vào các vùng khoanh đỏ trong hình dưới đây, ta thấy rằng trong URL có một thuộc tính là `page = 0` và trong UI ta thấy nó đang là trang `1/100`. Vậy nếu ta thay giá trị `page` này từ `[0, 99]` thì ta có thể truy cập tương ứng vào các `page` từ `[1:100]`

The screenshot shows a Shopee search results page for men's fashion. At the top, there are five promotional filters: "MUA 2 GIÁM 20K" (From ₫37,000), "MUA 3 GIÁM 30K" (From ₫1,000), "MUA NHIỀU GIÁM SÂU" (From ₫1,000), "SALE TỐI 50% M" (From ₫1,000), and "SALE TỐI 70% M" (From ₫1,000). Below these filters, there is a sidebar for filtering by category (e.g., Thời Trang Nam, Thời Trang Nữ, etc.) and location (e.g., TP. Hồ Chí Minh, Hà Nội). The main area displays a grid of men's clothing items, each with a price, discount percentage, and a "Mua Kém Deal Sốc" button. A red arrow points from the URL bar at the top to the "MUA 3 GIÁM 30K" filter, and another red arrow points from the URL bar to the page number "1/100" in the top right corner.

+ Hãy thử right-click vào mặt hàng đầu tiên và chọn **inspect**, ta có thể thấy được các sản phẩm này được nằm trong một HTML tag element là:

```
<div class="col-xs-2-4 shopee-search-item-result__item" data-sqe="item">
  ...
</div>
```

+ Chúng ta thấy rằng các sản phẩm được bọc trong thẻ các tag `<div>` mà có class là `shopee - search - item - result - item`.



+ Khi ta drop-down tag `< div >` này xuống, ta có thể thấy được tag `< a >` chứa hyper-link đến trang landing-page của sản phẩm này trong attribute `href`.

+ Sơ bộ là vậy, bây giờ chúng ta sẽ tiến hành lấy tất cả các hyperlink dẫn đến các trang landing-page này.

+ Do quá trình crawl data là một quá trình đòi hỏi tốn nhiều thời gian, vì thế nên chỉ có thể tiến hành demo các bước nhỏ chứ không thể báo cáo toàn bộ quá trình crawl trên một paper được, điều này càng khó khăn hơn khi crawl trên một dynamic website.

+ Bây giờ ta sẽ tiến hành lấy các URL của các sản phẩm để có thể truy cập vào trang riêng của sản phẩm đó.

```
[4]: %load_ext autoreload
%autoreload 2

The autoreload extension is already loaded. To reload it, use:
%reload_ext autoreload

[15]: import modules.crawler as Crawler

[6]: product_urls = Crawler.getProductURLS("https://shopee.vn/Th%E1%BB%9Di-Trang-Nam-cat.11035567?page=", [0, 10], "div.shopee-search-item-result__item > a")
```

```
[7]: product_urls[:5]

[7]: ['https://shopee.vn/Qu%E1%BA%A7n-jogger-nam-kaki-NPV-ki%E1%BB%83u-d%C3%A1ng-th%E1%BB%83-thao-qu%E1%BA%A7n-d%C3%A0i-nam-ch%E1%BA%A5t-li%E1%BB%87u-kaki-co-gi%C3%A3n-4-m%C3%A0u-i.195541001.5928995027?position=480',
      'https://shopee.vn/B%E1%BB%99-%C4%91%E1%BB%93ng-ph%E1%BB%A5c-th%E1%BB%83-thao-trung-h%E1%BB%8Dc-Fukurodani-h%C3%B3a-trang-nh%C3%A2n-v%E1%BA%ADt-Akaashi-Keiji-Bokuto-Koutarou-trong-anime-Haikyuu!!-i.254592742.7641350377?position=481',
      'https://shopee.vn/Qu%E1%BA%A7n-Jogger-kaki-kho%C3%A1-k%C3%A9o-%C3%A9o-1.142208247.2904777654?position=482',
      'https://shopee.vn/Qu%E1%BA%A7n-d%C3%A0i-nam-Qu%E1%BA%A7n-b%C3%82-jean-%E1%BB%91ng-c%C3%82-Slim-H%C3%82-C3%80NG-CA0-C%E1%BA%A4P-th%E1%BB%9Di-trang-phong-c%C3%82-%C3%A1ch-%C3%A2u-l%E1%BB%8Bch-l%C3%A3m-n%C4%83ng-%C4%91%E1%BB%99ng-KK3-i.89289068.6489798097?position=483',
      'https://shopee.vn/Face-shield-k%C3%AD%C3%ADnh-ph%C3%82-%C3%82ng-h%E1%BB%99-ch%E1%BB%91ng-gi%E1%BB%8Dt-b%E1%BA%AFn.-N%C3%82-C3%80n-Ch%E1%BB%91ng-D%E1%BB%8Bch-B%E1%BB%A5i-c%C3%82-B3-g%E1%BB%8Dng-cao-c%E1%BA%A5p-ph%C3%82-%C3%82-B9-h%E1%BB%A3p-m%E1%BB%8D-i-l%E1%BB%A9a-tu%E1%BB%95i-i.25104965.7744121028?position=484']
```

### Crawl comment từ hai sản phẩm với URL mà ta đã bắt được từ bước trên bằng Selenium

+ Vậy tóm lại các bước thực hiện là như sau:

- Đầu tiên ta vào trang chủ, có thể search mặt hàng mà ta muốn crawl hoặc chọn các gợi ý có sẵn, copy đường dẫn về.
- Tiếp theo, với mỗi URL như vậy, mặc định sẽ có 100 trang, ta sẽ crawl về mọi URL của các sản phẩm từ 100 trang này.
- Vào URL của từng sản phẩm:
  - Đi qua từng review navigation và crawl về toàn bộ.
  - Nhấn nút next navigation page và quay lại bước trên.
  - Nếu unable clicking, thì đã hết review và dừng lại quá trình crawl.

```
[9]: product_urls = [
    "https://shopee.vn/-M605-SET-B%E1%BB%98-TRANG-PH%E1%BB%A4C-L%E1%BB%8ACH-S%E1%BB%80-SANG-TR%E1%BB%8CNG-CH0-NG%C6%AF%E1%BB%9CI-TRUNG-NI",
    "https://shopee.vn/B%E1%BB%99-Qu%E1%BA%A7n-%C3%81o-Nam-Tay-Ng%E1%BA%AFn-C%E1%BB%95-B%E1%BA%BB-%C3%81o-Khuy-C%C3%A0i-Qu%E1%BA%A7n-Shor"
]

[10]: product_reviews = [] # chứa các Review object

for idx, product_url in enumerate(product_urls): # đi qua từng URL của sản phẩm
    new_reviews = Crawler.getProductReviews(product_url) # lấy tất cả review của sản phẩm này
    Crawler.writeToCsv(f"./tmp/product_reviews_00/product_{idx}.csv", new_reviews) # ghi mọi review của sản phẩm này ra file
    product_reviews += new_reviews # thêm vào để in kết quả (bước này kiểm tra cho cell dưới, thực tế ko sao)
```



```
[12]: for review in product_reviews[:10]:
```

```
    print(f" {review.iRating} - {review.iComment}")
```

5 – Áo siêu mát luôn í, chất đẹp lắm mua cho bà mà ưnggg hêt súcccc!!!!!!Nên mua nhé mọi người rất hợp với ng già hoặc mua cho mẹ cũng hợp luôn😊  
5 – Đồ đẹp sang vài mát mìn nhẹ mè mình khen đẹp nhìn trè hản ra giao hàng cũng nhanh  
5 – Shop giao hàng rất nhanh và tư vấn rất nhiệt tình. Nhận dc hàng mình thấy khá ưng ý.  
Clip chỉ mang tính nhận xu ạ  
5 – Chất mát, nhẹ, mềm mịn mặc lên cảm giác siêu nhẹ mà mát lắm luôn 👍👍  
5 – Áo đẹp, chất lượng  
Giao hàng nhanh  
Đóng hàng kĩ  
(Bừng đê ý đến ảnh và video chỉ mang tính chất nhận xu😊)  
5 – Mình mua tặng mẹ sinh nhật nên ib shop trả lời rất nhiệt tình, shop còn tặng mẹ mình thiệp sinh nhật cơ, đồ khá được, mè mình thích đáng của quần lắm <3  
5 – Hàng đẹp y hình, rất đáng mua nhé!  
5 – Chất vải đẹp, mua làm quà tặng rất hợp lý, mè mình khen tấm tắc, còn dặn mình đặt hộ tặng mấy cô bạn nữa. Yêu lắm.  
5 – Shop tư vấn nhiệt tình hỗ trợ mình đổi sz 🎉 siêu ưng luôn ý chắc chắn sẽ quay lại ủng hộ shop  
5 – Chất lượng sp tuyệt vời nha

```
[14]: print("Vậy ta có tổng cộng {} comment được crawl về từ {} sản phẩm trên.".format(len(product_reviews), len(product_urls)))
```

Vậy ta có tổng cộng 329 comment được crawl về từ 2 sản phẩm trên.

## Crawl comment từ hai sản phẩm với URL mà ta đã bắt được từ bước trên bằng API

+ Điều hạn chế khi crawl bằng API là ta cần biết hai thông tin là ID của shop bán hàng và ID sản phẩm, câu hỏi đặt ra là làm sao để ta có được 2 thông tin này. Khi ta tiến hành crawl URL của các sản phẩm bằng Selenium, hãy nhìn vào một URL cụ thể như hình dưới đây (chú ý vùng khoanh đỏ), mọi product's URL đều có cái này:

+ Ta có một URL's attribute là **i.34880242.2341969918**, đây chính là identifier cho sản phẩm này, với số:

- **34880242**: chính là ID của shop bán hàng.
- **2341969918**: chính là ID của sản phẩm.

- + Vậy ta đã có đủ thông tin để crawl data bằng API, hãy xem hàm **Crawler.getProductReviewsAPI()** bên dưới để hiểu cách nó hoạt động.
- + Vậy tóm lại quá trình crawl data bằng API.

- Ta cần product URL mà ta crawl được bằng Selenium, đây là bắt buộc để có được shop id và product ID.
- Khi ta có shop ID và Product ID, chỉ cần bỏ 2 attribute này vào request của API và nhận review về.

```
[17]: product_reviews = []

for idx, product_url in enumerate(product_urls): # duyệt qua các product's url
    new_reviews = Crawler.getProductReviewsAPI(product_url) # lấy tất cả reviews của product này thông qua API
    Crawler.writeToCsv(f"./tmp/product_reviews_{idx}.csv", new_reviews) # ghi mọi review của sản phẩm này ra file

product_reviews += new_reviews # thêm vào để kiểm tra (thực tế ko sao cái này, ignore nó)
```

**Nhận xét:** Việc ta crawl bằng API tiết kiệm thời gian crawl đáng kể hơn là bằng Selenium.

```
[18]: for review in product_reviews[:10]:
    print(f"{review.irating} - {review.icomment}")

5 - Áo siêu mát luôn luôn i, chất đẹp lắm mua cho bà mà ưnggg hết sứccc!!!!!!Nên mua nhé mọi người rất hợp với ng già hoặc mua cho mẹ cũng hợp luôn😊
5 - Đồ đẹp sang vài mát mịn nhẹ mềm khen đẹp nhìn trẻ hẳn ra giao hàng cũng nhanh
5 - Shop giao hàng rất nhanh và tư vấn rất nhiệt tình. Nhận dc hàng mình thấy khá ưng ý.
Clip chỉ mang tính nhận xét
5 - Chất mát, nhẹ, mềm mịn mặc lên cảm giác siêu nhẹ mà mát lắm luôn 🙌🙌🙌
5 - Áo đẹp, chất lượng
Giao hàng nhanh
Đóng hàng kín
(Đừng để ý đến ảnh và video chỉ mang tính chất nhận xét)
5 - Mình mua tặng mẹ sinh nhật nên ib shop trả lời rất nhiệt tình, shop còn tặng mẹ mình thiệp sinh nhật cơ, đồ khá được, mẹ mình thích đáng của quần lắm <3
5 - Hàng đẹp y hình, rất đáng mua nhé!
5 - Chất vải đẹp, mua làm quà tặng rất hợp lý, mẹ mình khen tấm tắc. còn dặn mình đặt hộ tặng mấy cô bạn nữa. Yêu lắm.
5 - Shop tư vấn nhiệt tình hỗ trợ mình đổi sz 🎉 siêu ưng luôn ý chắc chắn sẽ quay lại ủng hộ shop
5 - Chất lượng sp tuyệt vời nha

[19]: print("Vậy ta có tổng cộng {} comment được crawl về từ {} sản phẩm trên.".format(len(product_reviews), len(product_urls)))
```

Vậy ta có tổng cộng 305 comment được crawl về từ 2 sản phẩm trên.

## 5. Tài liệu tham khảo

- [Web Scraping with Python: Collecting Data from the Modern Web 1st Edition]  
(<https://www.amazon.com/Web-Scraping-Python-Collecting-Modern/dp/1491910291>)
- [Applied Text Analysis with Python: Enabling Language-Aware Data Products with Machine Learning 1st Edition]  
(<https://www.amazon.com/Applied-Text-Analysis-Python-Language-Aware/dp/1491963042>)
- [Book: Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit 1st Edition]  
(<https://bitly.com.vn/96w8t2>)



**fit@hcmus**

VNUHCM-UNIVERSITY OF SCIENCE  
FACULTY OF INFORMATION TECHNOLOGY

---

## 6. Mã nguồn

Link Google Drive:

[https://drive.google.com/drive/folders/1\\_a2MD-t5LFV7c0xQpwIP\\_Q7HY6lL3RNR?usp=sharing](https://drive.google.com/drive/folders/1_a2MD-t5LFV7c0xQpwIP_Q7HY6lL3RNR?usp=sharing)