ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

Nguyễn Trường Huy
22022509

BÁO CÁO
PHÂN TÍCH TƯƠNG TÁC VÀ NỘI DUNG CỦA MỘT / NHIỀU TÀI KHOẢN FACEBOOK

Môn: Lập trình xử lý dữ liệu với Python
Ngành: Trí tuệ nhân tạo

Hà Nội – 2023

Mục lục

1. TỔNG QUAN

1.1. Đề bài

Phân tích tương tác và nội dung của một / nhiều tài khoản Facebook.

1.2. Mô tả công việc

- Thu thập dữ liệu từ một / nhiều tài khoản Facebook (bài đăng, bình luận, tương tác…)
- Xử lý dữ liệu thu thập được, phân tích và rút ra những nhận xét.

1.3. Tài khoản Facebook

- Fanpage: EDM Vietnam Community
- Link Fanpage: https://www.facebook.com/edmvco

1.4. Link Github project

Link:
https://github.com/huynguyentruong119/AIT_2023_1_Final_Project.git

2. CÀO DỮ LIỆU VÀ LƯU TRỮ DỮ LIỆU

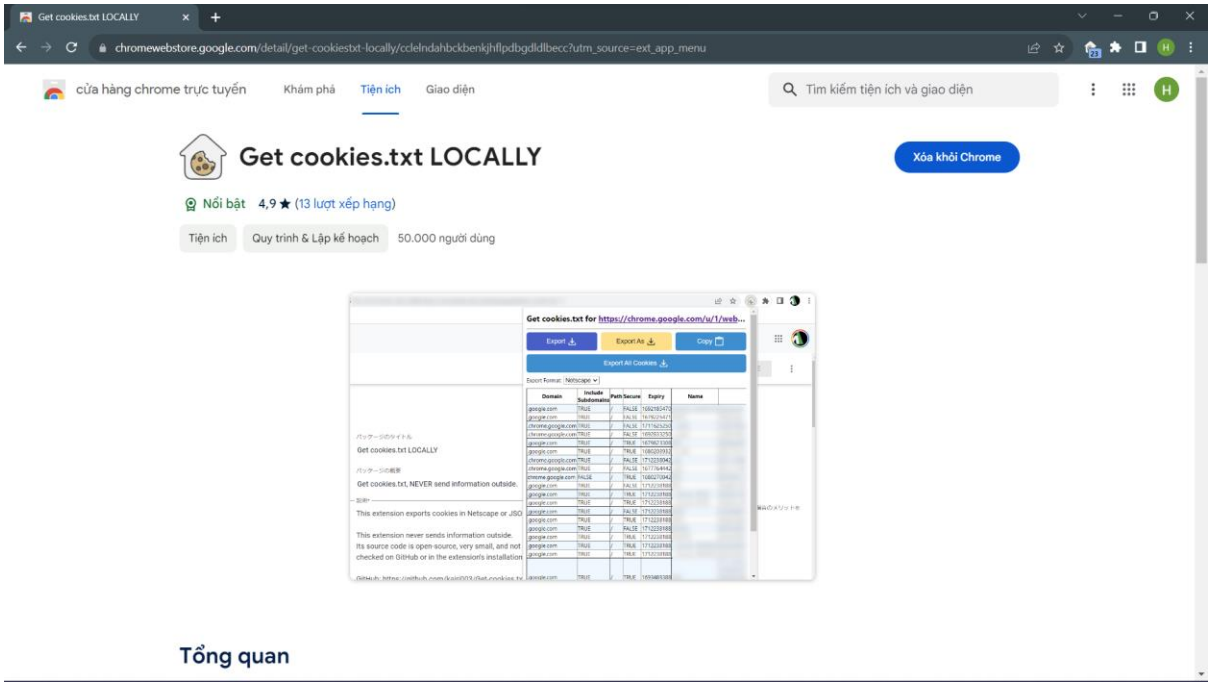
File: Crawl_Data.ipynb

2.1. Cào dữ liệu

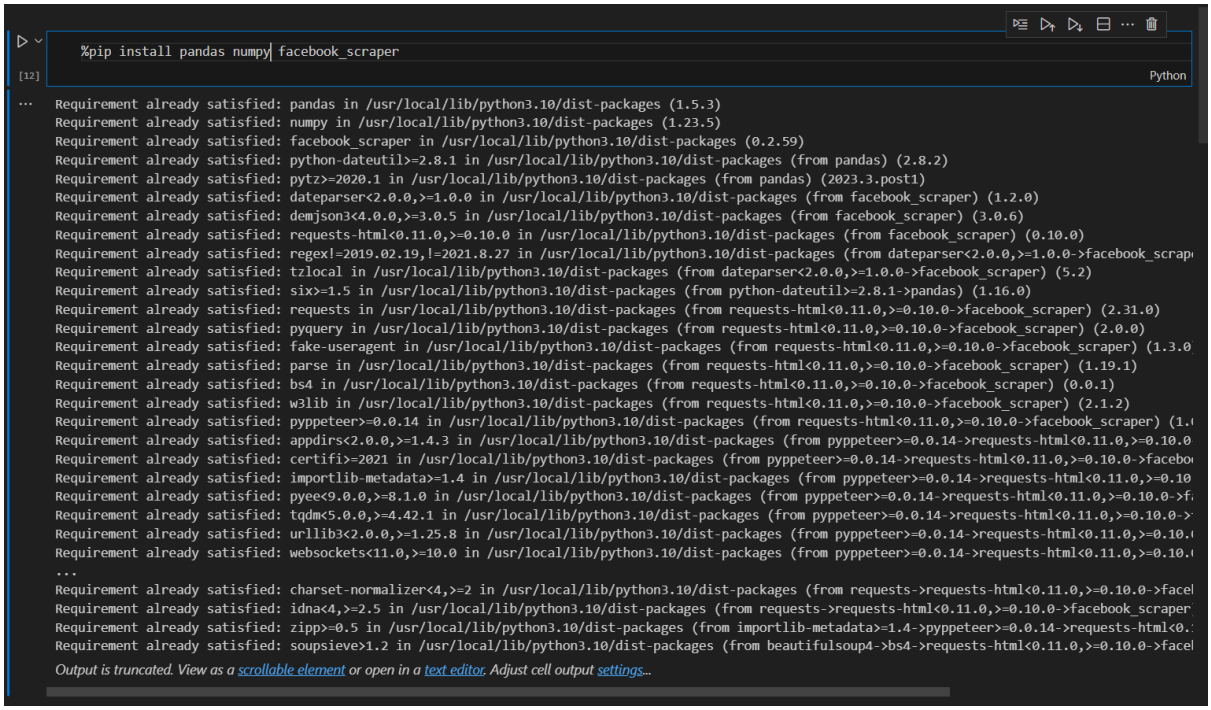
2.1.1. Cài đặt công cụ cần thiết

2.1.1.1. Get cookies.txt LOCALLY

Trên Google tải Extension: Get cookies.txt LOCALLY.



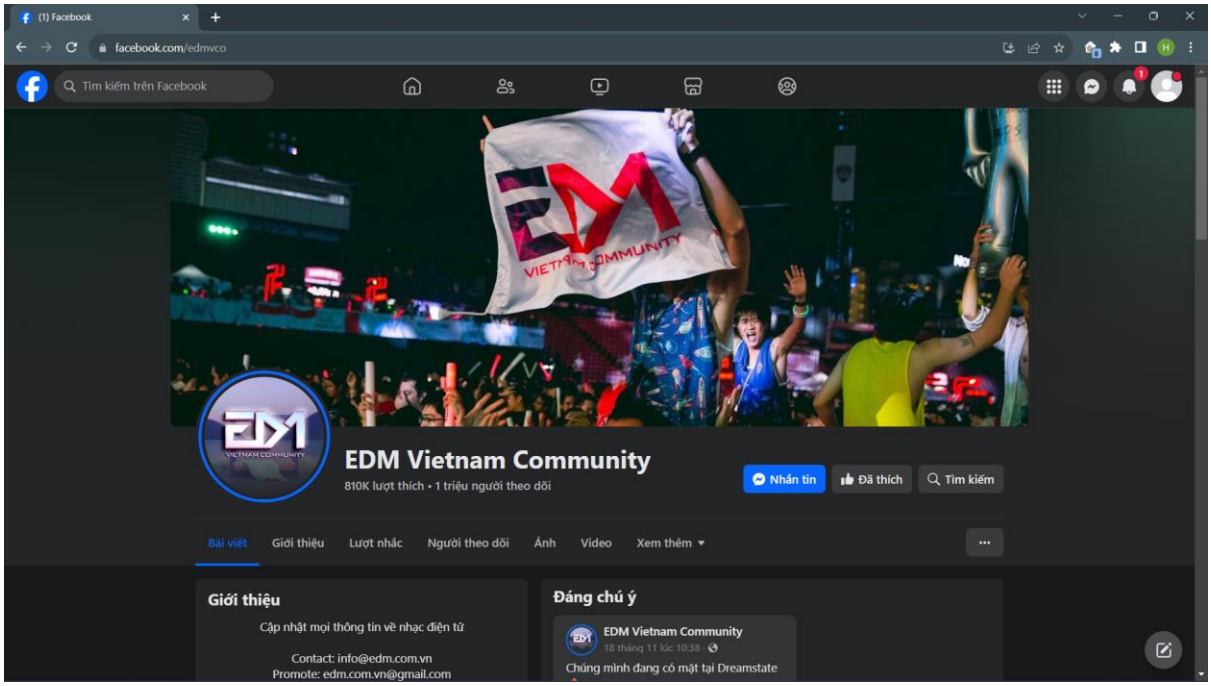
2.1.1.2. Thư viện facebook-scraper, pandas và numpy của Python



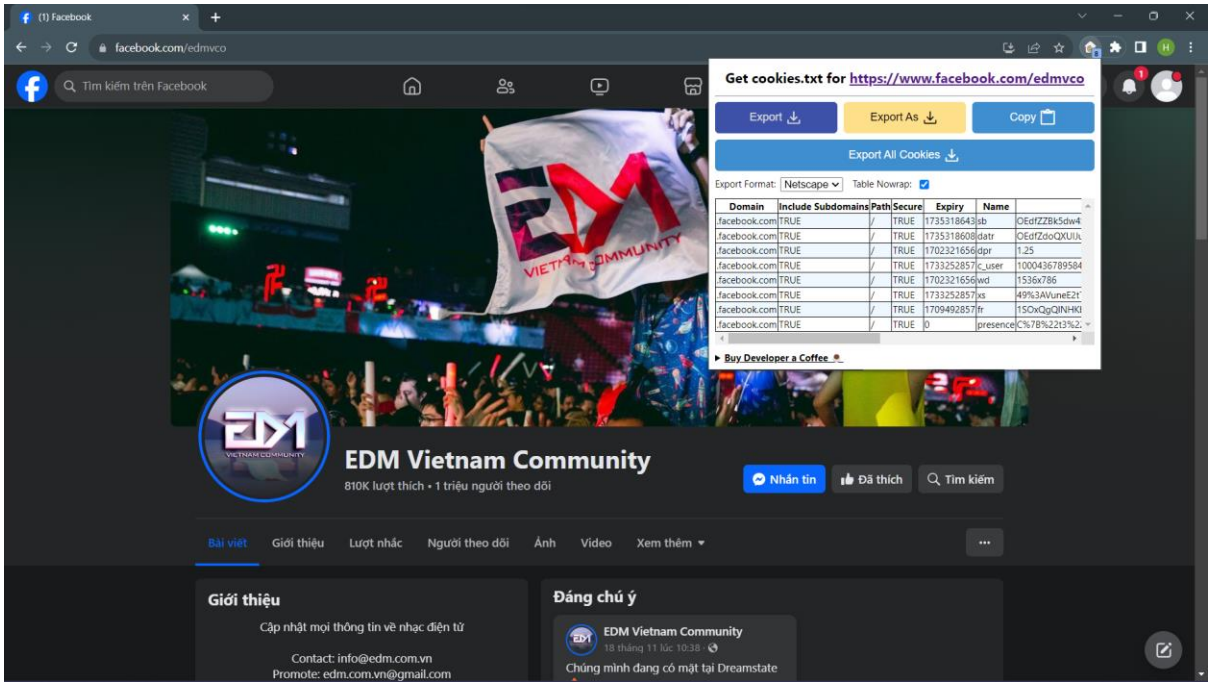
2.1.2. Cào dữ liệu

2.1.2.1. Lấy file cookies

- Bước 1: Mở trang Facebook cần cào dữ liệu.



- Bước 2: Bật Extension Get cookies.txt LOCALLY và nhấn Export để tải file cookies về.



- Bước 3: Tạo thư mục Data và cho file cookies vừa tải vào.
  File cookies: cookies.txt

2.1.2.2. Cào dữ liệu

- Import các thư viện cần thiết

```python
from facebook_scraper import get_posts
import pandas as pd
```

Python

- Cào 150 bài viết từ Fanpage

```python
FANPAGE_LINK = "edmvco"             # Link đến Fanpage
FOLDER_PATH = "Data/"               # Đường dẫn đến thư mục lưu dữ liệu
COOKIE_PATH = "Data/cookies.txt"    # Đường dẫn đến file cookies

PAGES_NUMBER = 15                   # số trang cần cào (1 trang có 10 bài đăng)
```

[14] Python

```python
post_list = []
for post in get_posts(FANPAGE_LINK,
                      options={"comments": True, "reactions": True, "allow_extra_requests": True},
                      extra_info=True, pages=PAGES_NUMBER, cookies=COOKIE_PATH):
    post_list.append(post)
```

[17] Python

Ở đây, chúng ta đã dùng hàm get_posts() của thư viện facebook_scraper để kéo dữ liệu của 150 bài đăng gần đây về và đưa nó vào trong post_list

Output: Xem ở project.

2.2. Lưu dữ liệu vừa thu thập được

```python
post_df_full = pd.DataFrame(columns=post_list[0].keys(), index=range(len(post_list)), data=post_list)

path=FOLDER_PATH + FANPAGE_LINK + ".csv"
post_df_full.to_csv(path, index=False)
```

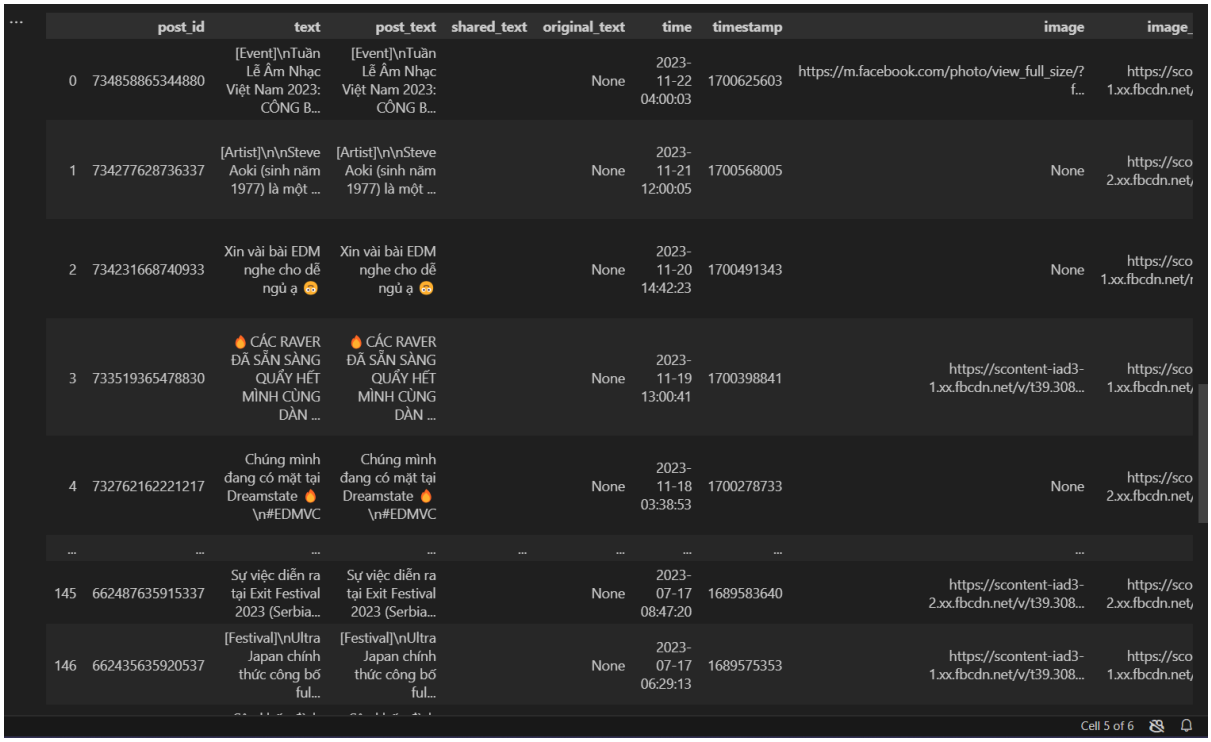
Python

```python
post_df_full
```

[19] Python

Tạo một DataFrame post_df_full để lưu dữ liệu vừa thu thập được và chuyển nó thành file .csv để lưu trữ dữ liệu

Output: Xem thêm ở project



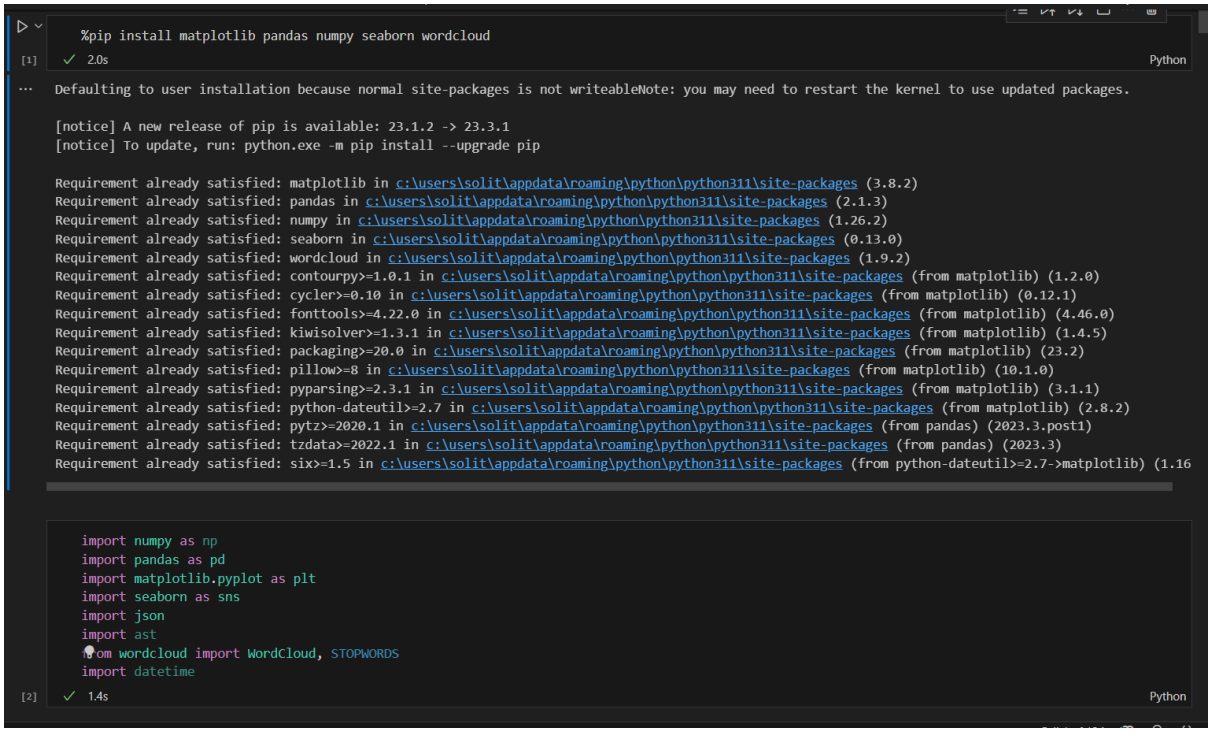
	post_id	text	post_text	shared_text	original_text	time	timestamp	image	image_
0	734858865344880	[Event]\nTuần Lễ Âm Nhạc Việt Nam 2023: CÔNG B...	[Event]\nTuần Lễ Âm Nhạc Việt Nam 2023: CÔNG B...		None	2023-11-22 04:00:03	1700625603	https://m.facebook.com/photo/view_full_size/?f...	https://sco 1.xx.fbcdn.net/
1	734277628736337	[Artist]\n\nSteve Aoki (sinh năm 1977) là một ...	[Artist]\n\nSteve Aoki (sinh năm 1977) là một ...		None	2023-11-21 12:00:05	1700568005	None	https://sco 2.xx.fbcdn.net/
2	734231668740933	Xin vài bài EDM nghe cho dễ ngủ ạ 😳	Xin vài bài EDM nghe cho dễ ngủ ạ 😳		None	2023-11-20 14:42:23	1700491343	None	https://sco 1.xx.fbcdn.net/r
3	733519365478830	🔥 CÁC RAVER ĐÃ SẴN SÀNG QUẨY HẾT MÌNH CÙNG DÀN ...	🔥 CÁC RAVER ĐÃ SẴN SÀNG QUẨY HẾT MÌNH CÙNG DÀN ...		None	2023-11-19 13:00:41	1700398841	https://scontent-iad3-1.xx.fbcdn.net/v/t39.308...	https://sco 1.xx.fbcdn.net/
4	732762162221217	Chúng mình đang có mặt tại Dreamstate 🔥 \n#EDMVC	Chúng mình đang có mặt tại Dreamstate 🔥 \n#EDMVC		None	2023-11-18 03:38:53	1700278733	None	https://sco 2.xx.fbcdn.net/
...	...	...	...	...	...	...	...	...	
145	662487635915337	Sự việc diễn ra tại Exit Festival 2023 (Serbia...	Sự việc diễn ra tại Exit Festival 2023 (Serbia...		None	2023-07-17 08:47:20	1689583640	https://scontent-iad3-2.xx.fbcdn.net/v/t39.308...	https://sco 2.xx.fbcdn.net/
146	662435635920537	[Festival]\nUltra Japan chính thức công bố ful...	[Festival]\nUltra Japan chính thức công bố ful...		None	2023-07-17 06:29:13	1689575353	https://scontent-iad3-1.xx.fbcdn.net/v/t39.308...	https://sco 1.xx.fbcdn.net/

Cell 5 of 6

File dữ liệu thô: edmvco.csv

3. LÀM SẠCH VÀ TIỀN XỬ LÝ DỮ LIỆU

Tải và import các thư viện cần thiết



```
%pip install matplotlib pandas numpy seaborn wordcloud
```

[1] ✓ 2.0s Python

```
Defaulting to user installation because normal site-packages is not writeableNote: you may need to restart the kernel to use updated packages.

[notice] A new release of pip is available: 23.1.2 -> 23.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip

Requirement already satisfied: matplotlib in c:\users\solit\appdata\roaming\python\python311\site-packages (3.8.2)
Requirement already satisfied: pandas in c:\users\solit\appdata\roaming\python\python311\site-packages (2.1.3)
Requirement already satisfied: numpy in c:\users\solit\appdata\roaming\python\python311\site-packages (1.26.2)
Requirement already satisfied: seaborn in c:\users\solit\appdata\roaming\python\python311\site-packages (0.13.0)
Requirement already satisfied: wordcloud in c:\users\solit\appdata\roaming\python\python311\site-packages (1.9.2)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\solit\appdata\roaming\python\python311\site-packages (from matplotlib) (1.2.0)
Requirement already satisfied: cycler>=0.10 in c:\users\solit\appdata\roaming\python\python311\site-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\solit\appdata\roaming\python\python311\site-packages (from matplotlib) (4.46.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\solit\appdata\roaming\python\python311\site-packages (from matplotlib) (1.4.5)
Requirement already satisfied: packaging>=20.0 in c:\users\solit\appdata\roaming\python\python311\site-packages (from matplotlib) (23.2)
Requirement already satisfied: pillow>=8 in c:\users\solit\appdata\roaming\python\python311\site-packages (from matplotlib) (10.1.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\solit\appdata\roaming\python\python311\site-packages (from matplotlib) (3.1.1)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\solit\appdata\roaming\python\python311\site-packages (from matplotlib) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\solit\appdata\roaming\python\python311\site-packages (from pandas) (2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in c:\users\solit\appdata\roaming\python\python311\site-packages (from pandas) (2023.3)
Requirement already satisfied: six>=1.5 in c:\users\solit\appdata\roaming\python\python311\site-packages (from python-dateutil>=2.7->matplotlib) (1.16
```

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import json
import ast
from wordcloud import WordCloud, STOPWORDS
import datetime
```

[2] ✓ 1.4s Python

3.1. Làm sạch

Ta sẽ xem qua bộ dữ liệu thô (Xem thêm Output ở project)

```
raw_df = pd.read_csv('Data/edmvco.csv')
raw_df
```

[3] ✓ 0.1s Python

	post_id	text	post_text	shared_text	original_text	time	timestamp	image	image_
0	734858865344880	[Event]\nTuần Lễ Âm Nhạc Việt Nam 2023: CÔNG B...	[Event]\nTuần Lễ Âm Nhạc Việt Nam 2023: CÔNG B...	NaN	NaN	2023-11-22 04:00:03	1700625603	https://m.facebook.com/photo/view_full_size/?f...	https://sco 1.xx.fbcdn.net/
1	734277628736337	[Artist]\n\nSteve Aoki (sinh năm 1977) là một ...	[Artist]\n\nSteve Aoki (sinh năm 1977) là một ...	NaN	NaN	2023-11-21 12:00:05	1700568005	NaN	https://sco 2.xx.fbcdn.net/
2	734231668740933	Xin vài bài EDM nghe cho dễ ngủ ạ 😡	Xin vài bài EDM nghe cho dễ ngủ ạ 😡	NaN	NaN	2023-11-20 14:42:23	1700491343	NaN	https://sco 1.xx.fbcdn.net/
3	733519365478830	🔥CÁC RAVER ĐÃ SẴN SÀNG QUẨY HẾT MÌNH CÙNG DÀN ...	🔥CÁC RAVER ĐÃ SẴN SÀNG QUẨY HẾT MÌNH CÙNG DÀN ...	NaN	NaN	2023-11-19 13:00:41	1700398841	https://scontent-iad3-1.xx.fbcdn.net/v/t39.308...	https://sco 1.xx.fbcdn.net/
4	732762162221217	Chúng mình đang có mặt tại Dreamstate 🔥 \n#EDMVC	Chúng mình đang có mặt tại Dreamstate 🔥 \n#EDMVC	NaN	NaN	2023-11-18 03:38:53	1700278733	NaN	https://sco 2.xx.fbcdn.net/
...	...	...	...	...	...	...	...	...	
145	662487635915337	Sự việc diễn ra tại Exit Festival 2023 (Serbia...	Sự việc diễn ra tại Exit Festival 2023 (Serbia...	NaN	NaN	2023-07-17 08:47:20	1689583640	https://scontent-iad3-2.xx.fbcdn.net/v/t39.308...	https://sco 2.xx.fbcdn.net/

Cell 1 of 124

```
raw_df.info()
```

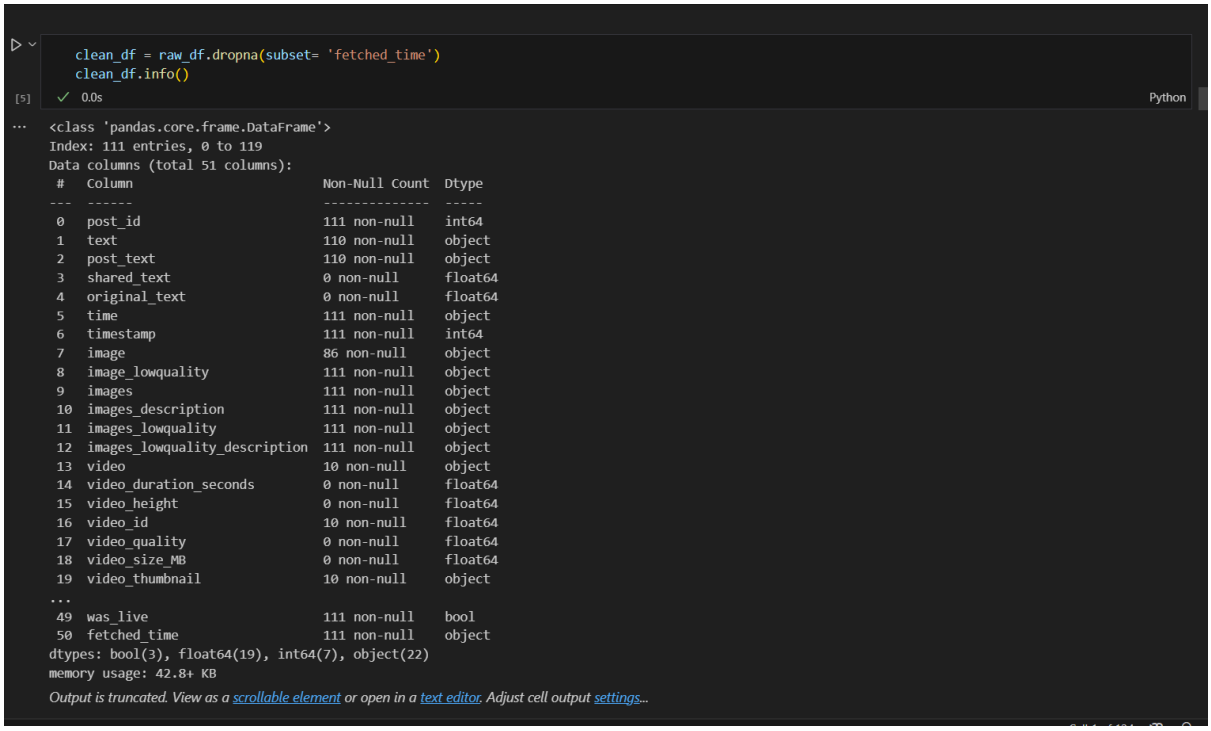
[4] ✓ 0.0s Python

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 51 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   post_id                      150 non-null    int64
 1   text                         146 non-null    object
 2   post_text                    146 non-null    object
 3   shared_text                  0 non-null      float64
 4   original_text                0 non-null      float64
 5   time                         150 non-null    object
 6   timestamp                    150 non-null    int64
 7   image                        119 non-null    object
 8   image_lowquality             150 non-null    object
 9   images                       150 non-null    object
 10  images_description           150 non-null    object
 11  images_lowquality            150 non-null    object
 12  images_lowquality_description  150 non-null  object
 13  video                        14 non-null     object
 14  video_duration_seconds       0 non-null      float64
 15  video_height                 0 non-null      float64
 16  video_id                     14 non-null     float64
 17  video_quality                0 non-null      float64
 18  video_size_MB                0 non-null      float64
 19  video_thumbnail              14 non-null     object
...
 49  was_live                     150 non-null    bool
 50  fetched_time                 111 non-null    object
dtypes: bool(3), float64(19), int64(7), object(22)
memory usage: 56.8+ KB
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

Những bản ghi có cột fetched_time (thời gian tạo requests) có giá trị NaN là do Facebook đã hạn chế quyền truy cập. Vì vậy khi ta nhìn vào tập dữ liệu sẽ thấy các bản ghi như vậy đều bị mất dữ liệu quan trọng. Để đảm bảo tính đúng

đắn cho việc phân tích dữ liệu, ta sẽ loại bỏ các bản ghi này. (Xem thêm Output ở project)



```
clean_df = raw_df.dropna(subset= 'fetched_time')
clean_df.info()
```

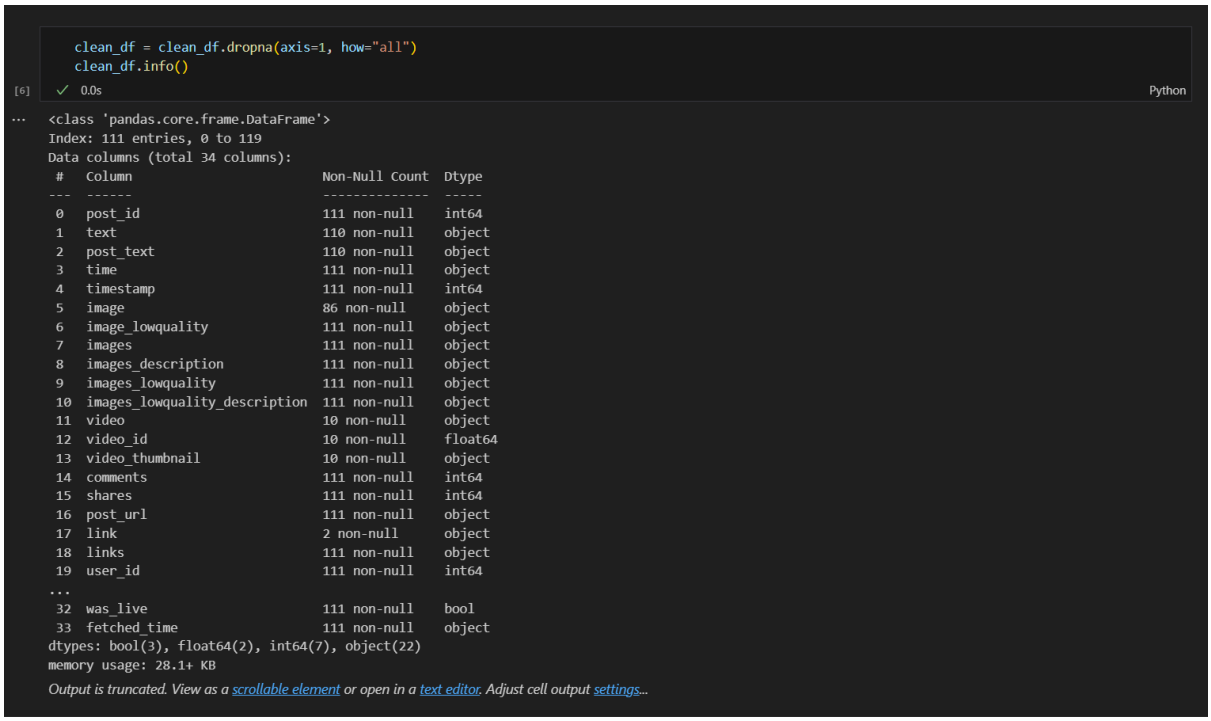
[5] ✓ 0.0s Python

```
<class 'pandas.core.frame.DataFrame'>
Index: 111 entries, 0 to 119
Data columns (total 51 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   post_id                         111 non-null    int64
 1   text                            110 non-null    object
 2   post_text                       110 non-null    object
 3   shared_text                     0 non-null      float64
 4   original_text                   0 non-null      float64
 5   time                            111 non-null    object
 6   timestamp                       111 non-null    int64
 7   image                           86 non-null     object
 8   image_lowquality                111 non-null    object
 9   images                          111 non-null    object
 10  images_description              111 non-null    object
 11  images_lowquality               111 non-null    object
 12  images_lowquality_description   111 non-null    object
 13  video                           10 non-null     object
 14  video_duration_seconds          0 non-null      float64
 15  video_height                    0 non-null      float64
 16  video_id                        10 non-null     float64
 17  video_quality                   0 non-null      float64
 18  video_size_MB                   0 non-null      float64
 19  video_thumbnail                 10 non-null     object
...
 49  was_live                        111 non-null    bool
 50  fetched_time                    111 non-null    object
dtypes: bool(3), float64(19), int64(7), object(22)
memory usage: 42.8+ KB
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

Ta nhận được 111 bài đăng.

Tiếp theo, ta sẽ loại bỏ đi các cột có tất cả các giá trị là NaN



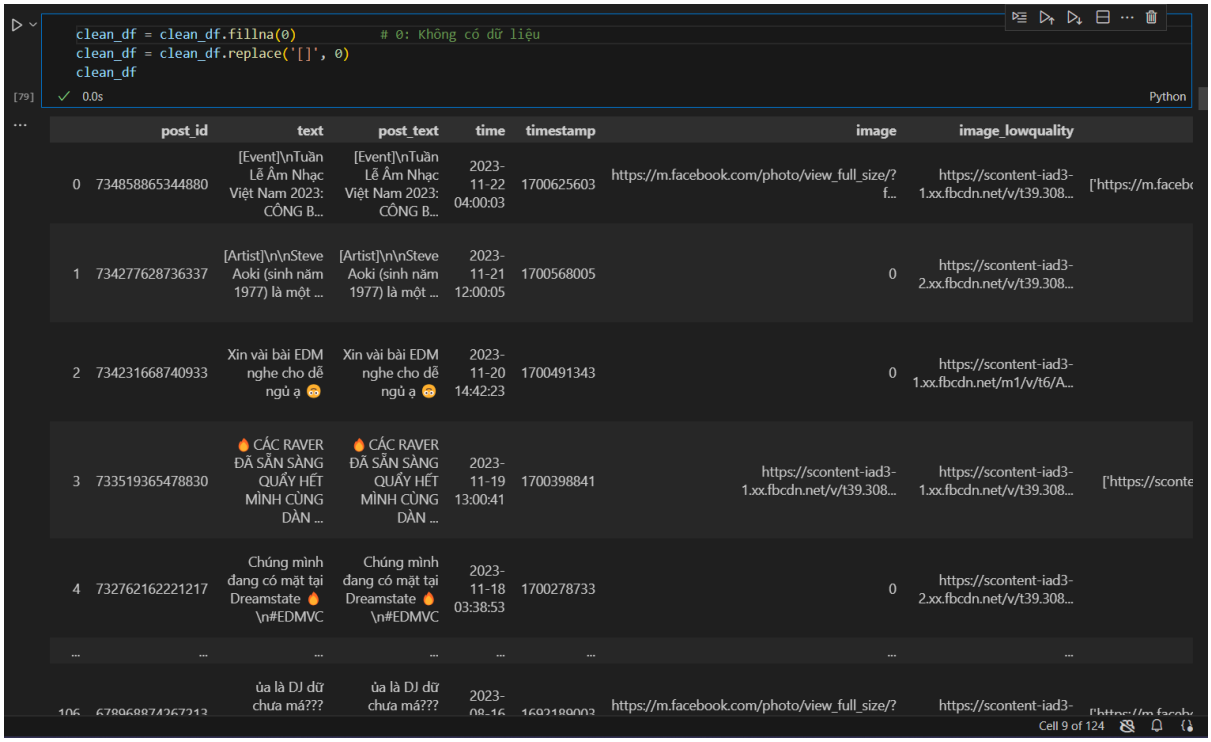
```
clean_df = clean_df.dropna(axis=1, how="all")
clean_df.info()
```

[6] ✓ 0.0s Python

```
<class 'pandas.core.frame.DataFrame'>
Index: 111 entries, 0 to 119
Data columns (total 34 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   post_id                         111 non-null    int64
 1   text                            110 non-null    object
 2   post_text                       110 non-null    object
 3   time                            111 non-null    object
 4   timestamp                       111 non-null    int64
 5   image                           86 non-null     object
 6   image_lowquality                111 non-null    object
 7   images                          111 non-null    object
 8   images_description              111 non-null    object
 9   images_lowquality               111 non-null    object
 10  images_lowquality_description   111 non-null    object
 11  video                           10 non-null     object
 12  video_id                        10 non-null     float64
 13  video_thumbnail                 10 non-null     object
 14  comments                        111 non-null    int64
 15  shares                          111 non-null    int64
 16  post_url                        111 non-null    object
 17  link                            2 non-null      object
 18  links                           111 non-null    object
 19  user_id                         111 non-null    int64
...
 32  was_live                        111 non-null    bool
 33  fetched_time                    111 non-null    object
dtypes: bool(3), float64(2), int64(7), object(22)
memory usage: 28.1+ KB
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

Sau đó, ta sẽ điền vào các ô còn trống giá trị 0 (Không có dữ liệu)



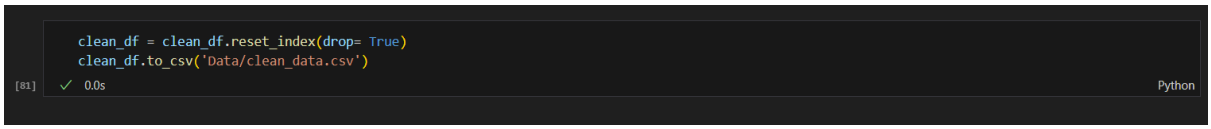
```
clean_df = clean_df.fillna(0)          # 0: Không có dữ liệu
clean_df = clean_df.replace('[]', 0)
clean_df
```

[79] ✓ 0.0s Python

	post_id	text	post_text	time	timestamp	image	image_lowquality	
0	734858865344880	[Event]\nTuần Lễ Âm Nhạc Việt Nam 2023: CÔNG B...	[Event]\nTuần Lễ Âm Nhạc Việt Nam 2023: CÔNG B...	2023-11-22 04:00:03	1700625603	https://m.facebook.com/photo/view_full_size/?f...	https://scontent-iad3-1.xx.fbcdn.net/v/t39.308...	['https://m.faceb
1	734277628736337	[Artist]\n\nSteve Aoki (sinh năm 1977) là một ...	[Artist]\n\nSteve Aoki (sinh năm 1977) là một ...	2023-11-21 12:00:05	1700568005	0	https://scontent-iad3-2.xx.fbcdn.net/v/t39.308...	
2	734231668740933	Xin vài bài EDM nghe cho dễ ngủ ạ 😳	Xin vài bài EDM nghe cho dễ ngủ ạ 😳	2023-11-20 14:42:23	1700491343	0	https://scontent-iad3-1.xx.fbcdn.net/m1/v/t6/A...	
3	733519365478830	🔥CÁC RAVER ĐÃ SẴN SÀNG QUẨY HẾT MÌNH CÙNG DÀN ...	🔥CÁC RAVER ĐÃ SẴN SÀNG QUẨY HẾT MÌNH CÙNG DÀN ...	2023-11-19 13:00:41	1700398841	https://scontent-iad3-1.xx.fbcdn.net/v/t39.308...	https://scontent-iad3-1.xx.fbcdn.net/v/t39.308...	['https://sconte
4	732762162221217	Chúng mình đang có mặt tại Dreamstate 🔥 \n#EDMVC	Chúng mình đang có mặt tại Dreamstate 🔥 \n#EDMVC	2023-11-18 03:38:53	1700278733	0	https://scontent-iad3-2.xx.fbcdn.net/v/t39.308...	
...	...	...	...	...	...	...	...	
106	678968874267213	ủa là DJ dữ chưa má???	ủa là DJ dữ chưa má???	2023-08-16	1692189003	https://m.facebook.com/photo/view_full_size/?	https://scontent-iad3-	['https://m.faceb

Cell 9 of 124

Cuối cùng, ta đánh lại chỉ mục index và chuyển thành file .csv để lưu trữ



```
clean_df = clean_df.reset_index(drop= True)
clean_df.to_csv('Data/clean_data.csv')
```

[81] ✓ 0.0s Python

File data đã được làm sạch: clean_data.csv

3.2. Tiền xử lý dữ liệu

3.2.1. File xử lý dữ liệu reactions: reactions_data.csv

Trường reactions của dữ liệu có chứa số lượng của các loại reactions

```
clean_df['reactions']
```

[218] ✓ 0.0s Python

```
0                      {'thích': 24, 'yêu thích': 8}
1      {'thích': 208, 'yêu thích': 36, 'haha': 24, 'w...
2      {'thích': 426, 'yêu thích': 11, 'haha': 133, '...
3      {'thích': 2599, 'yêu thích': 75, 'haha': 1, 'w...
4      {'thích': 47, 'yêu thích': 16, 'wow': 4, 'thươ...
                             ...
106    {'thích': 225, 'yêu thích': 2, 'haha': 277, 'b...
107    {'thích': 1393, 'yêu thích': 584, 'haha': 1, '...
108    {'thích': 3048, 'yêu thích': 438, 'haha': 7, '...
109    {'thích': 229, 'yêu thích': 88, 'haha': 45, 'w...
110    {'thích': 148, 'yêu thích': 46, 'wow': 1, 'thư...
Name: reactions, Length: 111, dtype: object
```

Ta sẽ tạo một DataFrame để lưu trữ dữ liệu này và làm sạch nó.

```
reactions_df = clean_df[['post_id', 'time', 'reactions', 'reaction_count', 'shares', 'comments']]
reactions_df['time'] = pd.to_datetime(reactions_df['time'])
reactions_df['reactions'] = reactions_df['reactions'].apply(lambda x : dict(eval(x)))
reactions_each_type_df = reactions_df['reactions'].apply(pd.Series)
reactions_each_type_df = reactions_each_type_df.fillna(0)
reactions_df = pd.concat([reactions_df, reactions_each_type_df], axis= 1).drop(columns= 'reactions')
reactions_df
```

[228] ✓ 0.0s Python

```
C:\Users\solit\AppData\Local\Temp\ipykernel_18180\104041726.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  reactions_df['time'] = pd.to_datetime(reactions_df['time'])
C:\Users\solit\AppData\Local\Temp\ipykernel_18180\104041726.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  reactions_df['reactions'] = reactions_df['reactions'].apply(lambda x : dict(eval(x)))
```

	post_id	time	reaction_count	shares	comments	thích	yêu thích	haha	wow	thương thương	buồn	phẫn nộ
0	734858865344880	2023-11-22 04:00:03	32	0	4	24.0	8.0	0.0	0.0	0.0	0.0	0.0
1	734277628736337	2023-11-21 12:00:05	274	4	13	208.0	36.0	24.0	2.0	4.0	0.0	0.0
2	734231668740933	2023-11-20 14:42:23	577	25	335	426.0	11.0	133.0	1.0	2.0	3.0	1.0
3	733519365478830	2023-11-19 13:00:41	2680	2	5	2599.0	75.0	1.0	4.0	0.0	1.0	0.0
4	732762162221217	2023-11-18 03:38:53	72	0	3	47.0	16.0	0.0	4.0	5.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...
106	678968874267213	2023-08-16 12:30:03	508	11	18	225.0	2.0	277.0	0.0	0.0	4.0	0.0
107	676766991154068	2023-08-12 10:12:22	2007	21	30	1393.0	584.0	1.0	0.0	25.0	4.0	0.0

Cell 14 of 125

Các trường còn lại thêm vào để tạo thuận lợi cho việc phân tích.

Chuyển dữ liệu thành file .csv để lưu trữ.

```
reactions_df.to_csv('Data/reactions_data.csv')
```

[229] ✓ 0.0s Python

+ Code + Markdown

3.2.2. File xử lý dữ liệu links: links_data.csv

Trường links trong tập dữ liệu có chứa tất cả đường dẫn được đính kèm trong từng bài đăng như: link web, hashtag, …

```
clean_df['links']
```

[367] ✓ 0.0s Python

```
0      [{'link': '/story.php?story_fbid=pfbid02Ki9kYS...
1      [{'link': '/story.php?story_fbid=pfbid02KY71Kn...
2      [{'link': '/story.php?story_fbid=pfbid0g5b3C7d...
3      [{'link': '/story.php?story_fbid=pfbid0EfoCF7U...
4      [{'link': '/DreamstateUSA?eav=AfbhA1z_HuYesL9l...
                             ...
106    [{'link': '/hashtag/edmvc?__ft__=encrypted_track...
107    [{'link': '/hashtag/edmvc?__ft__=encrypted_track...
108    [{'link': '/story.php?story_fbid=pfbid02sPq6zi...
109    [{'link': '/hashtag/edmvc?__ft__=encrypted_track...
110    [{'link': '/story.php?story_fbid=6722363116071...
Name: links, Length: 111, dtype: object
```

Tạo DataFrame và lưu tất cả link trong các bài đăng

```
links_df = pd.DataFrame()
for link in clean_df['links']:
    link_data = json.dumps(link)
    link_data = json.loads(link_data)
    link_data = eval(link_data)
    link_data_df = pd.DataFrame(link_data)
    links_df = pd.concat([links_df, link_data_df])
links_df
```

[305] ✓ 0.0s Python

	link	text
0	/story.php?story_fbid=pfbid02Ki9kYS1GBUq4azqU8...	Xem thêm
1	https://lm.facebook.com/l.php?u=https%3A%2F%2F...	https://ticketbox.vn/event/vietnam-music-week-...
2	/hashtag/edmvc?refid=17&__ft__=encrypted_trackin...	#EDMVC
3	/hashtag/vietnammusicweek2023?refid=17&__ft__=en...	#VietnamMusicWeek2023
4	/hashtag/vmin?refid=17&__ft__=encrypted_tracking...	#VMIN
...	...	...
0	/hashtag/edmvc?__ft__=encrypted_tracking_data.0A...	#EDMVC
1	/story.php?story_fbid=pfbid0ZydwXGXmsU71nJKDhA...	
2	https://m.facebook.com/photo.php?fbid=67563743...	
0	/story.php?story_fbid=6722363116071368substory...	
1	/photo.php?fbid=6722363116071368id=10006462319...	

510 rows × 2 columns

Dữ liệu trong DataFrame có định dạng string, vì thế ta sẽ chuyển dữ liệu qua dạng json bằng hàm json.dump(), sau đó chuyển dữ liệu dạng json sang đối tượng Python bằng hàm json.load().

Làm sạch và chuyển thành file .csv để lưu trữ

```
links_df = links_df.reset_index(drop= True)
links_df = links_df.replace('', 0)
links_df.to_csv('Data/links_data.csv')
```

[306] ✓ 0.0s Python

3.2.3. File xử lý dữ liệu time: time_post.csv

Trường time của dữ liệu chỉ thời gian đăng bài post

```
clean_df['time']
```

[368] ✓ 0.0s Python

```
0      2023-11-22 04:00:03
1      2023-11-21 12:00:05
2      2023-11-20 14:42:23
3      2023-11-19 13:00:41
4      2023-11-18 03:38:53
              ...
106    2023-08-16 12:30:03
107    2023-08-12 10:12:22
108    2023-08-10 12:00:28
109    2023-08-10 09:10:29
110    2023-08-04 08:50:15
Name: time, Length: 111, dtype: object
```

Ta sẽ tạo DataFrame để lưu trữ ngày và giờ đăng bài.

```
time_post = clean_df[['post_id', 'time']]
time_post['time'] = pd.to_datetime(time_post['time'])
time_post['date'] = time_post['time'].dt.date
time_post['hour'] = time_post['time'].dt.time
time_post
```

[307] ✓ 0.0s Python

```
C:\Users\solit\AppData\Local\Temp\ipykernel_18180\2490160626.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  time_post['time'] = pd.to_datetime(time_post['time'])
C:\Users\solit\AppData\Local\Temp\ipykernel_18180\2490160626.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  time_post['date'] = time_post['time'].dt.date
C:\Users\solit\AppData\Local\Temp\ipykernel_18180\2490160626.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  time_post['hour'] = time_post['time'].dt.time
```

	post_id	time	date	hour
0	734858865344880	2023-11-22 04:00:03	2023-11-22	04:00:03
1	734277628736337	2023-11-21 12:00:05	2023-11-21	12:00:05
2	734231668740933	2023-11-20 14:42:23	2023-11-20	14:42:23
3	733519365478830	2023-11-19 13:00:41	2023-11-19	13:00:41
4	732762162221217	2023-11-18 03:38:53	2023-11-18	03:38:53
...	...	...	...	...
106	67896887426721 3	2023-08-16 12:30:03	2023-08-16	12:30:03

Cell 22 of 128

Chuyển thành file .csv để lưu trữ.

```
time_post.to_csv('Data/time_post.csv')
```

[308] ✓ 0.0s Python

3.2.4. File xử lý dữ liệu reactors: reactors_data.csv

Trường reactors chứa thông tin của những tài khoản đã thả reaction trong từng bài đăng của trang.

```
clean_df['reactors']
```

[369] ✓ 0.0s Python

```
0                                                    0
1      [{'name': 'Kiên Mai Ímе', 'link': 'https://fac...
2      [{'name': 'Bạc Xỉu', 'link': 'https://facebook...
3      [{'name': 'Phat Ngan', 'link': 'https://facebo...
4      [{'name': 'Hungg Le', 'link': 'https://faceboo...
                             ...
106    [{'name': 'Kiên Mai Ímе', 'link': 'https://fac...
107    [{'name': 'Nghĩa Quách', 'link': 'https://face...
108    [{'name': 'ஐ ஹ', 'link': 'https://facebook.c...
109    [{'name': 'Nguyễn Quang', 'link': 'https://fac...
110    [{'name': 'Vũ Quỳnh', 'link': 'https://faceboo...
Name: reactors, Length: 111, dtype: object
```

Ta sẽ tạo DataFrame để lưu trữ dữ liệu này (Làm tương tự như trường links ở trên)

```python
tmp_df = clean_df[clean_df['reactors'] != 0]
reactors_df = pd.DataFrame()
for reactor in tmp_df['reactors']:
    reactor_data = json.dumps(reactor)
    reactor_data = json.loads(reactor_data)
    reactor_data = eval(reactor_data)
    reactor_data_df = pd.DataFrame(reactor_data)
    reactors_df = pd.concat([reactors_df, reactor_data_df])
reactors_df
```

[309] ✓ 0.2s Python

	name	link	type
0	Kiên Mai Íme	https://facebook.com/profile.php?id=1000940985...	like
1	Quinny Ng	https://facebook.com/profile.php?id=1000935124...	like
2	Dũng Lee	https://facebook.com/profile.php?id=1000916250...	like
3	Tuyết Mai	https://facebook.com/profile.php?id=1000913083...	like
4	Nguyễn Thùy Dương	https://facebook.com/profile.php?id=1000906921...	wow
...	...	...	...
45	Best EDM Beats	https://facebook.com/BESTEDMBEATSS?eav=AfZy-a7...	like
46	Nguyễn Trường An	https://facebook.com/profile.php?id=1000718717...	love
47	Canalis Club	https://facebook.com/canalisclub.vn?eav=Afb2mS...	love
48	Ngô Thắng	https://facebook.com/thang.ngo.1612007?eav=Afa...	love
49	TechBeat Records	https://facebook.com/TechBeat.vn?eav=AfblVkg6C...	love

8610 rows × 3 columns

Làm sạch và chuyển thành .csv để lưu trữ

```python
reactors_df = reactors_df.replace({'like' : 'thích', 'love' : 'yêu thích', 'sad' : 'buồn', 'care' : 'thương thương'})
reactors_df = reactors_df.reset_index(drop= True)
reactors_df = reactors_df.replace('', 0)
reactors_df.to_csv('Data/reactors_data.csv')
```

[256] ✓ 0.0s Python

3.2.5. File xử lý dữ liệu comments_full: comments_full.csv

Trường comment_full chứa thông tin của các comment trong từng bài đăng của trang như: id comment, nội dung comment, id người comment...

```python
clean_df['comments_full']
```

[370] ✓ 0.0s Python

```
0      [{'comment_id': '1061064065093152', 'comment_u...
1      [{'comment_id': '865432821690369', 'comment_ur...
2      [{'comment_id': '1002554197493256', 'comment_u...
3      [{'comment_id': '866036398252543', 'comment_ur...
4      [{'comment_id': '374895771637258', 'comment_ur...
                             ...
106    [{'comment_id': '644423690978847', 'comment_ur...
107    [{'comment_id': '319444710530678', 'comment_ur...
108    [{'comment_id': '1011325836665119', 'comment_u...
109    [{'comment_id': '573802094770298', 'comment_ur...
110    [{'comment_id': '1310257489599662', 'comment_u...
Name: comments_full, Length: 111, dtype: object
```

Ta sẽ tạo DataFrame để lưu trữ dữ liệu này (Làm tương tự như trường links ở trên)

```
tmp2_df = clean_df[clean_df['comments_full'] != 0]
comments_full_df = pd.DataFrame()
for comment in tmp2_df['comments_full']:
    comment_data = json.dumps(comment)
    comment_data = json.loads(comment_data)
    comment_data = eval(comment_data)
    comment_data_df = pd.DataFrame(comment_data)
    comments_full_df = pd.concat([comments_full_df, comment_data_df])
comments_full_df
```

[311] ✓ 0.4s Python

	comment_id	comment_url	commenter_id	commenter_url	commenter_name	commenter_meta	com
0	1061064065093152	https://facebook.com/1061064065093152	100064623192769	https://facebook.com/edmvco?eav=AfYeMqh7p8OHnb...	EDM Vietnam Community	Tác giả	T Nhạ Mu
1	746007517389746	https://facebook.com/746007517389746	100004233820245	https://facebook.com/thanhluan10497?eav=Afavm_...	Nguyễn Thành Luân	None	Ngu
2	1587591868715101	https://facebook.com/1587591868715101	100004028093197	https://facebook.com/profile.php?id=1000040280...	Duy Minh	None	
0	865432821690369	https://facebook.com/865432821690369	100064623192769	https://facebook.com/edmvco?eav=AfY-7w2v2_zcL-...	EDM Vietnam Community	Tác giả	Sắp
1	1783294532107184	https://facebook.com/1783294532107184	100005519463021	https://facebook.com/phamtamphong2101?eav=AfZu...	Phạm Tâm Phong	None	anl nén
...	...	...	...	...	...	...	
3	1472344080250863	https://facebook.com/1472344080250863	100010067591455	https://facebook.com/bestedmbeat?eav=AfasRbR8c...	Dang Chi Huong	None	
4	302208355662958	https://facebook.com/302208355662958	100006861136137	https://facebook.com/profile.php?id=1000068611...	Anh Khoa Hoang	None	
5	293122040065458	https://facebook.com/293122040065458	100007760587444	https://facebook.com/flamez.nguyen.0212?eav=Af...	Nguyễn Hoàng Đông Phi	None	Kl 20
6	248371768128047	https://facebook.com/248371768128047	100085238347563	https://facebook.com/profile.php?id=1000852383...	Minh Nhật	None	
7	1016898652773345	https://facebook.com/1016898652773345	100024594242282	https://facebook.com/hoanganh.raver?eav=AfZYo0...	Hoang Anh Pham	None	

1771 rows × 13 columns

Làm sạch và chuyển thành .csv để lưu trữ

```
comments_full_df = comments_full_df.reset_index(drop= True)
comments_full_df['commenter_meta'] = comments_full_df['commenter_meta'].fillna('Người xem')
comments_full_df['comment_time'] = comments_full_df['comment_time'].fillna(comments_full_df['comment_time'].mean())
comments_full_df['comment_time'] = comments_full_df['comment_time'].dt.date
comments_full_df = comments_full_df.fillna(0)
comments_full_df = comments_full_df.replace('[]', 0)
comments_full_df.to_csv('Data/comments_full.csv')
```

[312] ✓ 0.0s Python

4. PHÂN TÍCH DỮ LIỆU

4.1. Thông tin trang

- Tên trang

```
clean_df['username'].loc[0]
```

[313] ✓ 0.0s Python

```
'EDM Vietnam Community'
```

- ID trang

```
clean_df['page_id'].loc[0]
```
[314] ✓ 0.0s Python

```
673611379413932
```

- URL trang

```
clean_df['user_url'].loc[0]
```
[315] ✓ 0.0s Python

```
'https://facebook.com/edmvco?lst=61553217674439%3A100064623192769%3A1700745307&eav=AfbRVPtwo8MdkbYYgJfsdR3XzRHTFDL9t2wY48-ujBuKf8TLOVv2RZq3A8JWhCbwkow
```

4.2. Phân tích reactions, comments, shares

4.2.1. Phân tích shares

- Tổng quan về số lượt share

```
clean_df['shares'].describe()
```
[316] ✓ 0.0s Python

```
count    111.000000
mean      42.504505
std      116.319456
min        0.000000
25%        3.000000
50%        7.000000
75%       24.000000
max      866.000000
Name: shares, dtype: float64
```

Ta có thể thấy được trung bình mỗi bài viết có 42-43 lượt share, bài viết có số lượt share cao nhất là 866 và thấp nhất là 0

- Bài viết có số lượt shares cao nhất (Xem thêm Output ở project)

```
clean_df.nlargest(1, 'shares', keep= 'all')
```
[30] ✓ 0.0s Python

	post_id	text	post_text	time	timestamp	image	image_lowquality	images	images_description	ima
60	699546618876105	Thông tin đã được công bố chính thức bởi Heine...	Thông tin đã được công bố chính thức bởi Heine...	2023-09-22 03:28:12	1695353292	https://scontent-iad3-1.xx.fbcdn.net/v/t39.308...	https://scontent-iad3-1.xx.fbcdn.net/v/t39.308...	['https://scontent-iad3-1.xx.fbcdn.net/v/t39.3...	['Có thể là hình ảnh về 2 người và văn bản']	['h 1.xx.fbc

1 rows × 34 columns

- Tổng số lượt share

```
clean_df['shares'].sum()
[317] ✓ 0.0s                                                    Python
... 4718
```

4.2.2. Phân tích comments

- Tổng quan về số lượt comment

```
clean_df['comments'].describe()
[320] ✓ 0.0s                                                    Python
... count    111.000000
    mean     183.810811
    std      577.739978
    min        1.000000
    25%        9.500000
    50%       23.000000
    75%       85.000000
    max     5243.000000
    Name: comments, dtype: float64
                                        + Code  + Markdown
```

Ta có thể thấy được trung bình mỗi bài viết có 183-184 lượt comment, bài viết có số lượt comment cao nhất là 5243 và thấp nhất là 1

- Bài viết có số lượt comment cao nhất (Xem thêm Output ở project)

```
clean_df.nlargest(1, 'comments', keep= 'all')
[107] ✓ 0.0s                                                    Python
```

	post_id	text	post_text	time	timestamp	image	image_lowquality	images	images_description	ima
60	699546618876105	Thông tin đã được công bố chính thức bởi Heine...	Thông tin đã được công bố chính thức bởi Heine...	2023-09-22 03:28:12	1695353292	https://scontent-iad3-1.xx.fbcdn.net/v/t39.308...	https://scontent-iad3-1.xx.fbcdn.net/v/t39.308...	['https://scontent-iad3-1.xx.fbcdn.net/v/t39.3...	['Có thể là hình ảnh về 2 người và văn bản']	['h 1.xx.fbc

1 rows × 34 columns

- Tổng số lượt comment

```
clean_df['comments'].sum()
[321] ✓ 0.0s                                                    Python
... 20403
```

4.2.3. Phân tích reactions

- Tổng quan về số lượt reaction

```
reactions_df['reaction_count'].describe()
```

```
[324] ✓ 0.0s Python

count      111.000000
mean      2034.054054
std       5217.361811
min         32.000000
25%        227.000000
50%        572.000000
75%       1479.000000
max      37629.000000
Name: reaction_count, dtype: float64
```

Ta có thể thấy được trung bình mỗi bài viết có 2034 lượt reaction, bài viết có số lượt reaction cao nhất là 37629 và thấp nhất là 32

- Bài viết có số lượt reaction cao nhất (Xem thêm Output ở project)

```
clean_df.nlargest(1, 'reaction_count', keep= 'all')
```

```
[152] ✓ 0.0s Python
```

	post_id	text	post_text	time	timestamp	image	image_lowquality	images	images_description	imag
44	707938714703562	Alan Walker vừa đáp xuống sân bay Tân Sơn Nhất...	Alan Walker vừa đáp xuống sân bay Tân Sơn Nhất...	2023-10-06 07:51:43	1696578703	https://scontent-iad3-2.xx.fbcdn.net/v/t39.308...	https://scontent-iad3-2.xx.fbcdn.net/v/t39.308...	['https://scontent-iad3-2.xx.fbcdn.net/v/t39.3...	['Có thể là hình ảnh về 5 người và văn bản']	['ht 2.xx.fbc

1 rows × 34 columns

- Tổng số lượt reaction

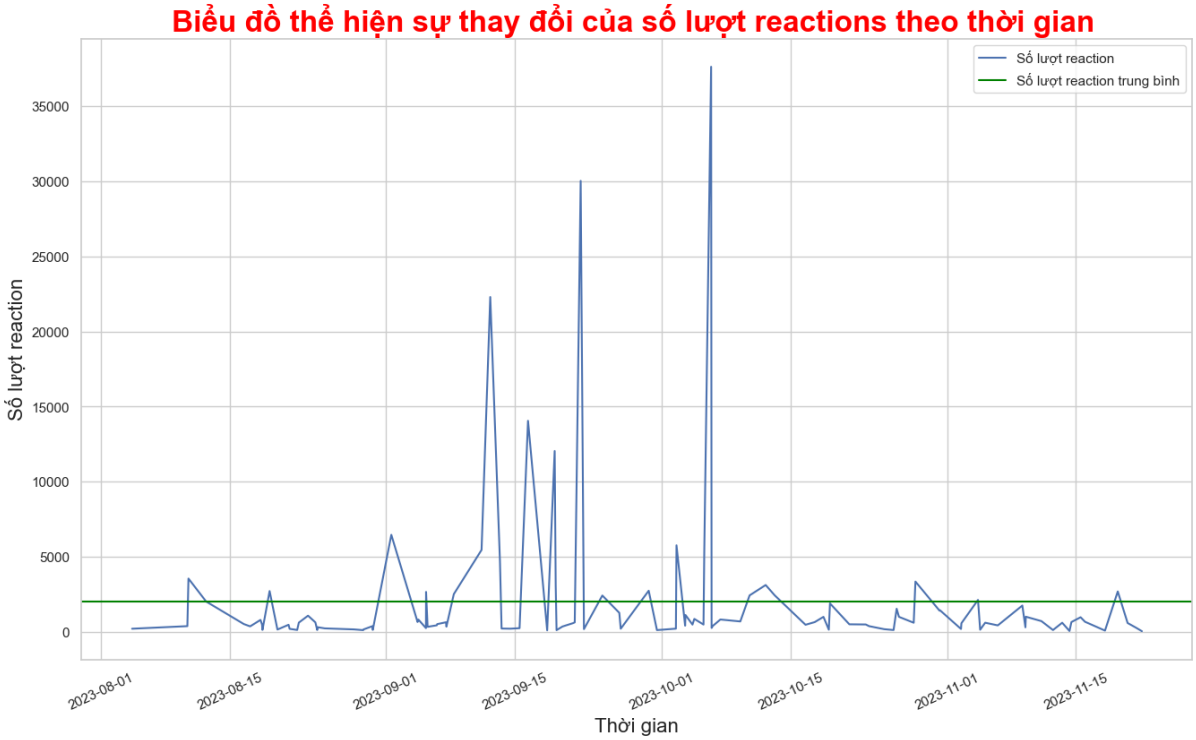
```
reactions_df['reaction_count'].sum()
```

```
[325] ✓ 0.0s Python

225780
```

- Sự thay đổi của số lượt reaction theo thời gian

```
sns.set_theme(style="whitegrid")
plt.subplots(figsize=(16, 9))
sns.lineplot(x = 'time', y = 'reaction_count', data = reactions_df, label= 'Số lượt reaction')
plt.xticks(rotation = 25)
plt.xlabel('Thời gian', fontsize=16)
plt.ylabel('Số lượt reaction', fontsize=16)
plt.title('Biểu đồ thể hiện sự thay đổi của số lượt reactions theo thời gian ', fontsize=24, color='red', fontweight='bold')
plt.axhline(reactions_df['reaction_count'].mean(), color='green', label='Số lượt reaction trung bình')
plt.legend()
```

```
[328] ✓ 0.4s Python
```



Theo biểu đồ trên, ta có thể thấy được trong khoảng thời gian từ 01-09-2023 đến 15-10-2023 số lượt reactions tăng đột biến, vượt xa so với mức trung bình.

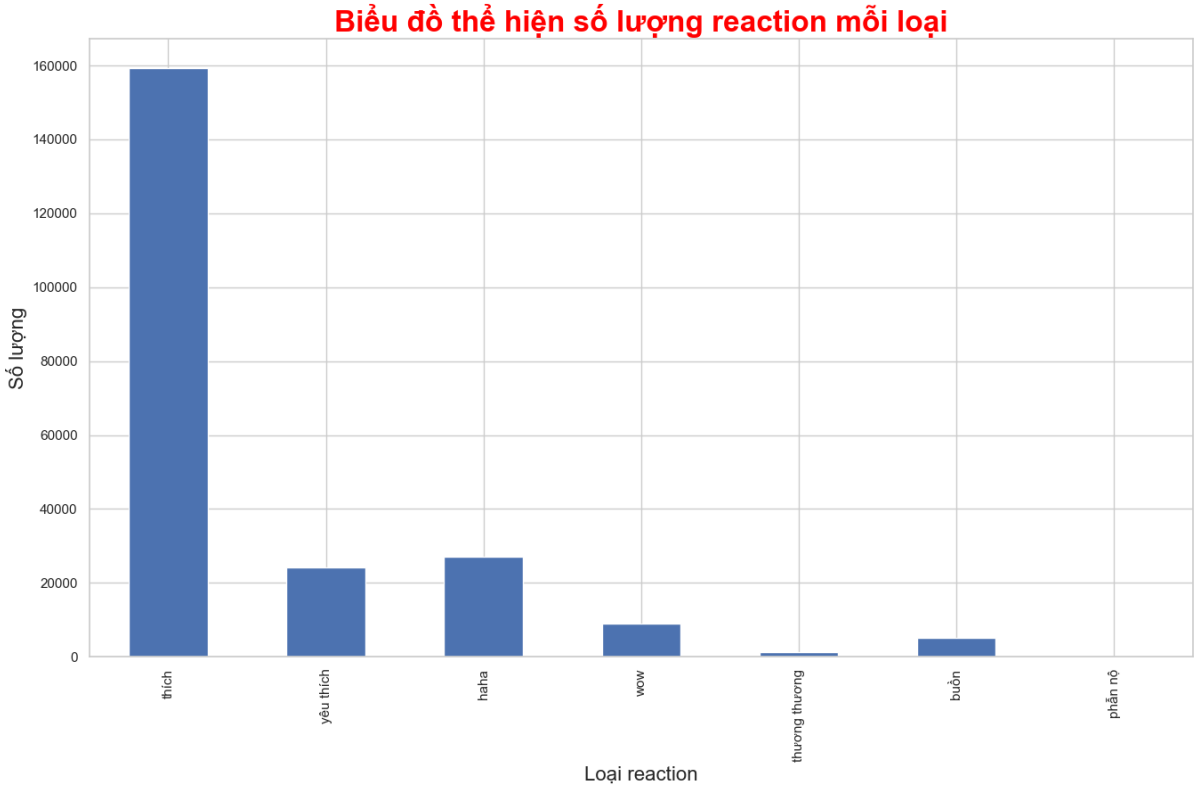
Trong khoảng thời gian này diễn ra lễ hội âm nhạc Heineken Kennation ở Việt Nam với sự tham gia của Alan Walker – DJ/Producer hàng đầu thế giới. Điều này thu hút sự chú ý của đông đảo các fan tại Việt Nam và họ có xu hướng theo dõi những thông tin liên quan đến anh chàng DJ này và lễ hội âm nhạc. Vì vậy mà số lượng reactions mới tăng đột biến như vậy.

- Số lượng reactions mỗi loại

```
sum_of_reaction = pd.Series(reactions_df[['thích', 'yêu thích', 'haha', 'wow', 'thương thương', 'buồn', 'phẫn nộ']].sum())
sum_of_reaction.plot(kind= 'bar', figsize= (16, 9))
plt.xlabel('Loại reaction', fontsize=16)
plt.ylabel('Số lượng', fontsize=16)
plt.title('Biểu đồ thể hiện số lượng reaction mỗi loại', fontsize=24, color='red', fontweight='bold')
sum_of_reaction
```

[153] ✓ 0.2s Python

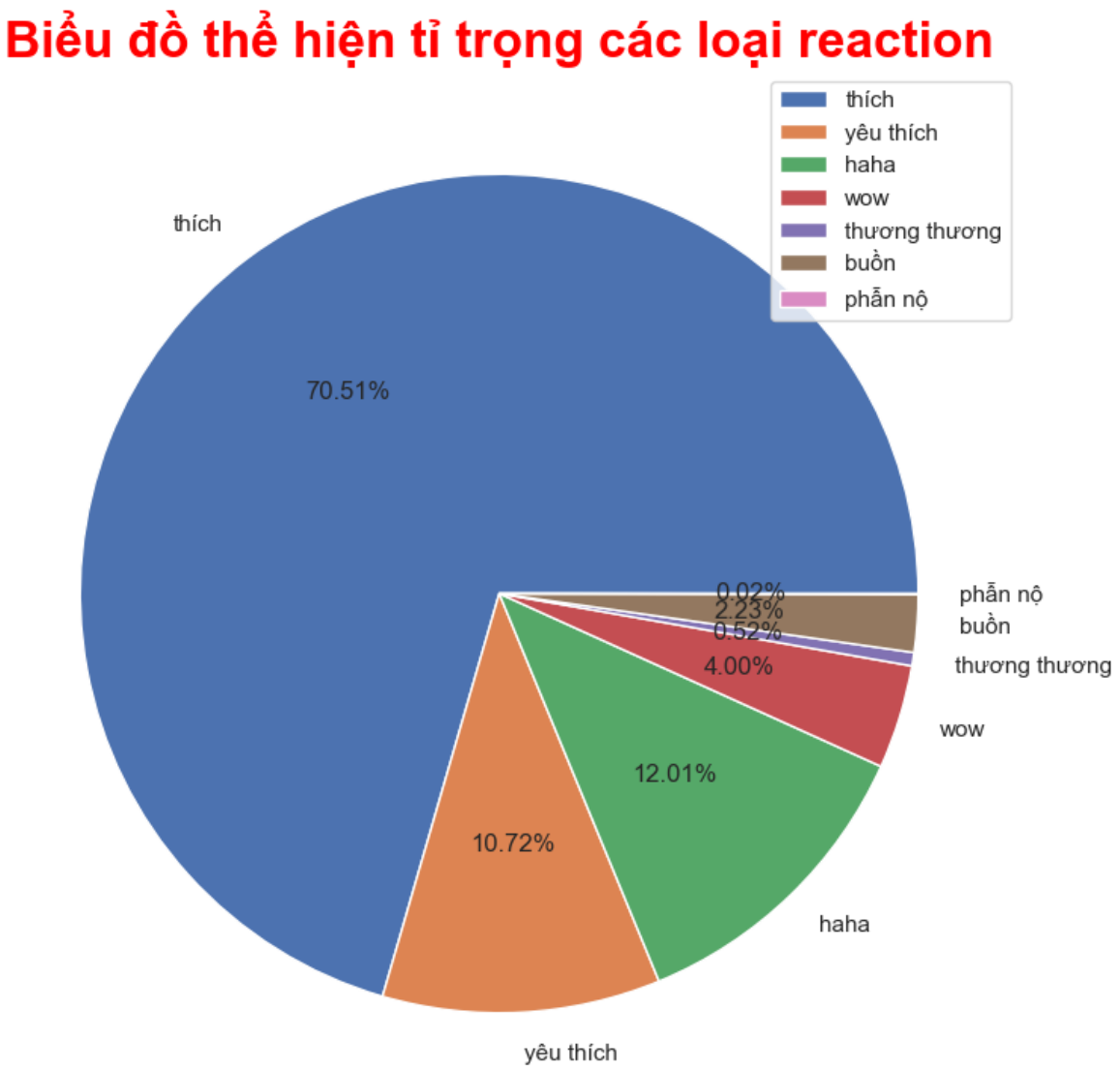
```
thích           159196.0
yêu thích        24203.0
haha             27115.0
wow               9032.0
thương thương     1164.0
buồn              5027.0
phẫn nộ             43.0
dtype: float64
```



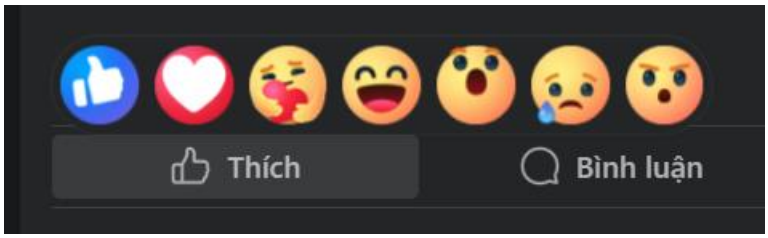

```
sum_of_reaction.plot(kind= 'pie', figsize= (9, 9), autopct='%1.2f%%')
plt.title('Biểu đồ thể hiện tỉ trọng các loại reaction', fontsize=24, color='red', fontweight='bold')
plt.legend()
```

[113] ✓ 0.2s Python

```
<matplotlib.legend.Legend at 0x1e9e6019c10>
```



Ở đây, ta có thể thấy reaction thích có số lượng vượt trội hơn hẳn so với phần còn lại. Nguyên nhân có thể là do nút thích nằm ở ngay đầu thanh reaction và reaction mặc định của Facebook là thích nên người dùng chỉ cần nhấn một lần là được. Còn các nút reaction còn lại thì người dùng phải thao tác kéo thả trên thanh reaction, phải thực hiện nhiều thao tác hơn. Vì vậy người dùng có xu hướng nhấn thích nhiều hơn các reaction còn lại.



- Mức độ tương quan giữa tổng số reaction, số reaction mỗi loại, tổng số comment và tổng số share

Bảng thể hiện mức độ tương quan giữa các trường

```
corr = reactions_df[['reaction_count', 'shares', 'comments', 'thích', 'yêu thích',
                     'haha', 'wow', 'thương thương', 'buồn', 'phẫn nộ']].corr(numeric_only= True)

corr
```

[194] ✓ 0.0s Python

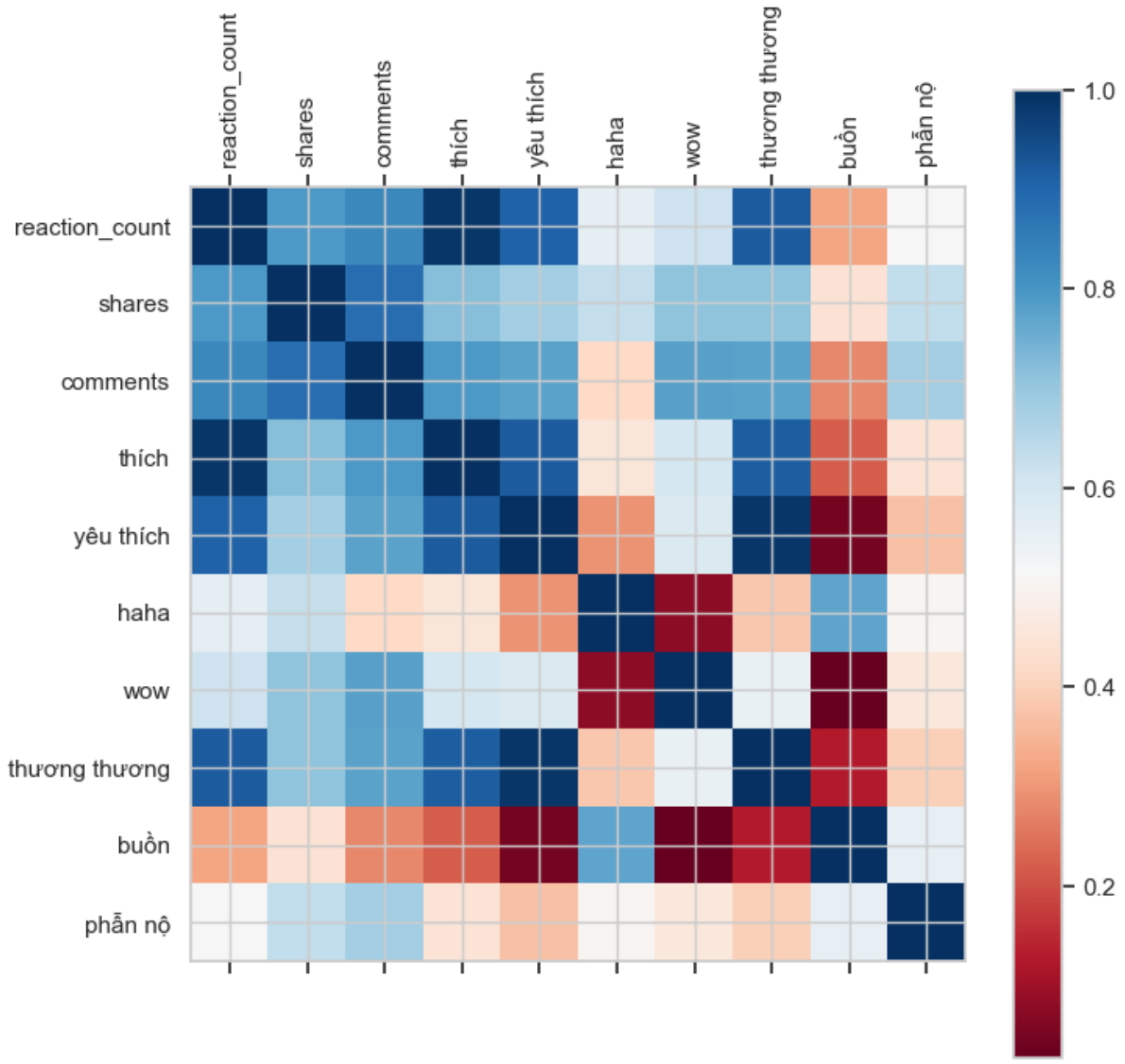
	reaction_count	shares	comments	thích	yêu thích	haha	wow	thương thương	buồn	phẫn nộ
reaction_count	1.000000	0.792502	0.827602	0.988563	0.908291	0.560174	0.613441	0.918958	0.322393	0.513368
shares	0.792502	1.000000	0.882487	0.719802	0.678506	0.630695	0.709115	0.709874	0.442953	0.635160
comments	0.827602	0.882487	1.000000	0.794128	0.772744	0.416061	0.777688	0.774155	0.278643	0.679619
thích	0.988563	0.719802	0.794128	1.000000	0.916741	0.453916	0.597708	0.915305	0.219159	0.446319
yêu thích	0.908291	0.678506	0.772744	0.916741	1.000000	0.294520	0.582363	0.986707	0.048806	0.372521
haha	0.560174	0.630695	0.416061	0.453916	0.294520	1.000000	0.078064	0.381766	0.771532	0.504361
wow	0.613441	0.709115	0.777688	0.597708	0.582363	0.078064	1.000000	0.551108	0.027142	0.461923
thương thương	0.918958	0.709874	0.774155	0.915305	0.986707	0.381766	0.551108	1.000000	0.126406	0.397234
buồn	0.322393	0.442953	0.278643	0.219159	0.048806	0.771532	0.027142	0.126406	1.000000	0.552107
phẫn nộ	0.513368	0.635160	0.679619	0.446319	0.372521	0.504361	0.461923	0.397234	0.552107	1.000000

Biểu đồ thể hiện mức độ tương quan giữa các trường

```
fig = plt.figure(figsize=(8,8))
plt.matshow(corr, cmap='RdBu', fignum=fig.number)
plt.xticks(range(len(corr.columns)), corr.columns, rotation='vertical');
plt.yticks(range(len(corr.columns)), corr.columns);
plt.colorbar()
```

[195] ✓ 0.2s Python

```
<matplotlib.colorbar.Colorbar at 0x1e9e754a010>
```

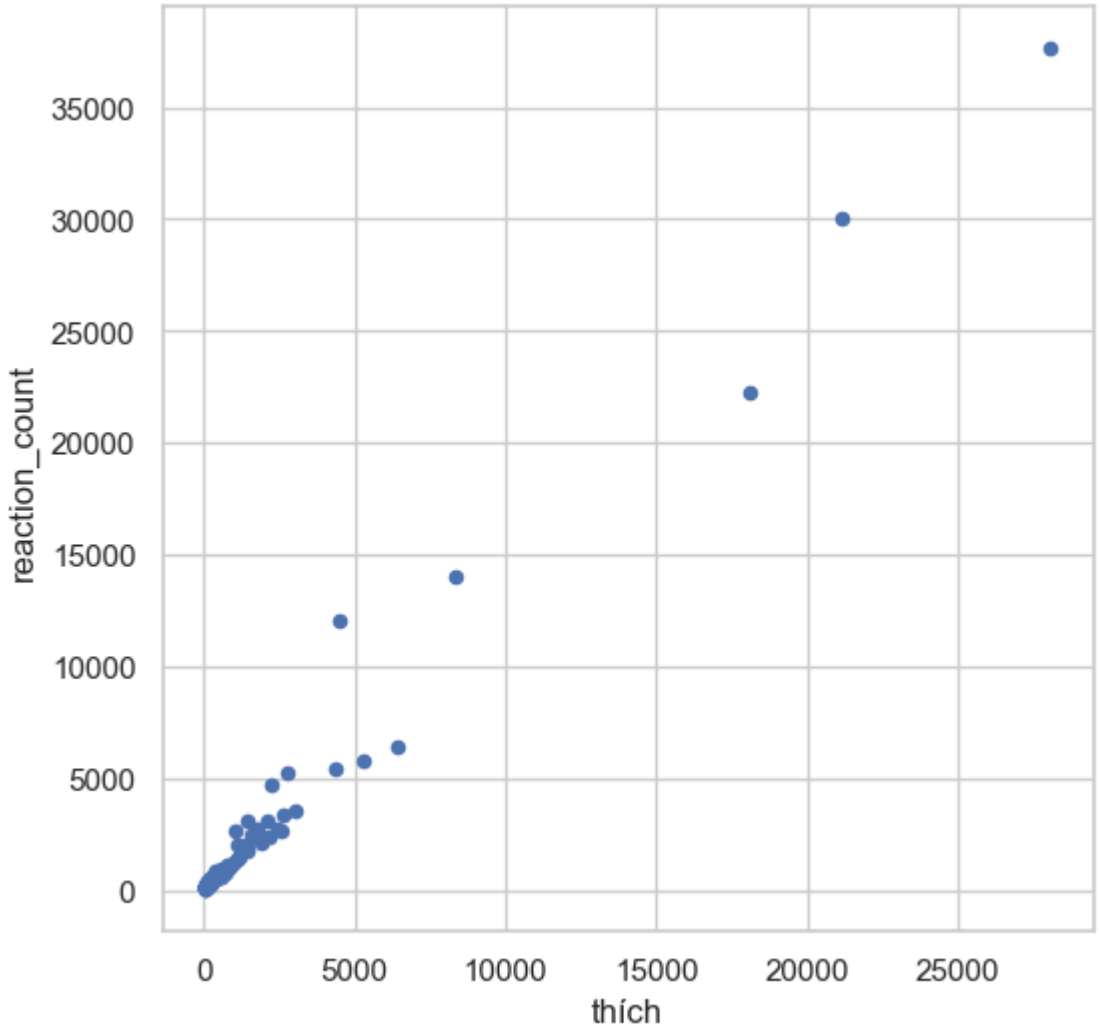


Qua bảng và biểu đồ trên, ta thấy:

- Số reaction có mức độ tương quan tương đối với số comment và share. Điều này là hợp lý vì khi số reaction cao tức là bài post đấy thu hút nhiều sự chú ý, điều đó cũng góp phần làm tăng sự tương tác của người dùng.
- Đặc biệt, số lượt thích có mức độ tương quan gần như tuyệt đối so với số lượng reaction. Điều này dễ hiểu vì số lượt thích chiếm tỉ trọng rất lớn trong tổng số reaction. Vì vậy khi số lượt thích tăng thì số lượt reaction cũng sẽ tăng. Vậy nên giữa số lượt thích và số reaction có mối quan hệ với nhau và phân tích hồi quy tuyến tính có thể phù hợp.

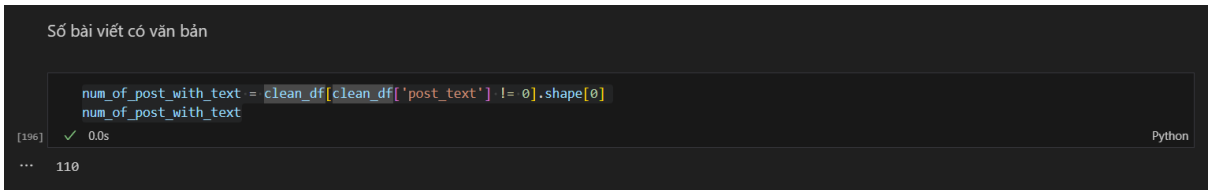
Biểu đồ thể hiện số lượt thích và số reactions:

```
reactions_df.plot(kind='scatter', x='thích', y='reaction_count', figsize=(6,6))
[230] ✓ 0.2s                                                              Python
... <Axes: xlabel='thích', ylabel='reaction_count'>
```



4.3. Phân tích bài đăng

- Số bài viết có chứa văn bản



- Số bài viết có chứa hình ảnh

Số bài viết có hình ảnh

```
num_of_post_with_image = clean_df[clean_df['image'] != 0].shape[0]
num_of_post_with_image
```

[197] ✓ 0.0s Python

```
86
```

- Số bài viết có chứa video

```
num_of_post_with_video = clean_df[clean_df['video'] != 0].shape[0]
num_of_post_with_video
```

[198] ✓ 0.0s Python

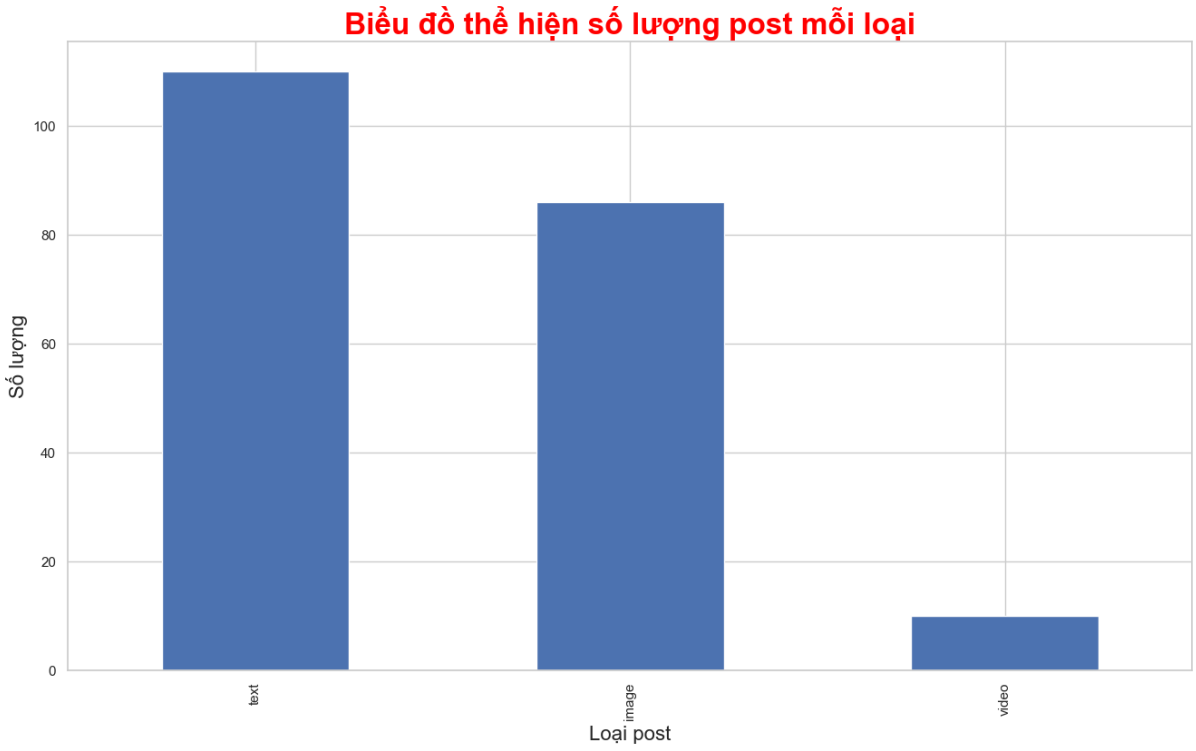
```
10
```

- Biểu đồ thể hiện số lượng bài đăng mỗi loại

```
num_of_post = pd.Series([num_of_post_with_text, num_of_post_with_image, num_of_post_with_video], index=['text', 'image', 'video'])
num_of_post.plot(kind= 'bar', figsize= (16, 9))
plt.xlabel('Loại post', fontsize=16)
plt.ylabel('Số lượng', fontsize=16)
plt.title('Biểu đồ thể hiện số lượng post mỗi loại', fontsize=24, color='red', fontweight='bold')
```

[199] ✓ 0.1s Python

```
Text(0.5, 1.0, 'Biểu đồ thể hiện số lượng post mỗi loại')
```



Ta có thể thấy page có xu hướng đăng những bài post có văn bản và hình ảnh, ít khi đăng video

- Tổng số link đính kèm trong bài post

```
links_df.shape[0]
[201] ✓ 0.0s Python
... 510
```

- Trung bình 1 bài post sec có 4-5 link đính kèm

```
links_df.shape[0] / clean_df.shape[0]
[202] ✓ 0.0s Python
... 4.594594594594595
+ Code + Markdown
```

- Nội dung link

```
links_df['text'].value_counts()
[203] ✓ 0.0s Python
... text
0                              211
#EDMVC                          86
Xem thêm                        46
#AlanWalker                      9
#KENNATION                       8
                              ...
Ravolution Music Festival        1
#TheChainsmokers                 1
+5                               1
#freedvm_films                   1
#CLEARMắtLạnhBạcHà               1
Name: count, Length: 97, dtype: int64
```

Đa số các link đều không có văn bản đính kèm, một vài link được đính kèm dưới dạng hashtag...

- Kiểm tra xem các bài post có đang livestream hay đã từng livestream hay không

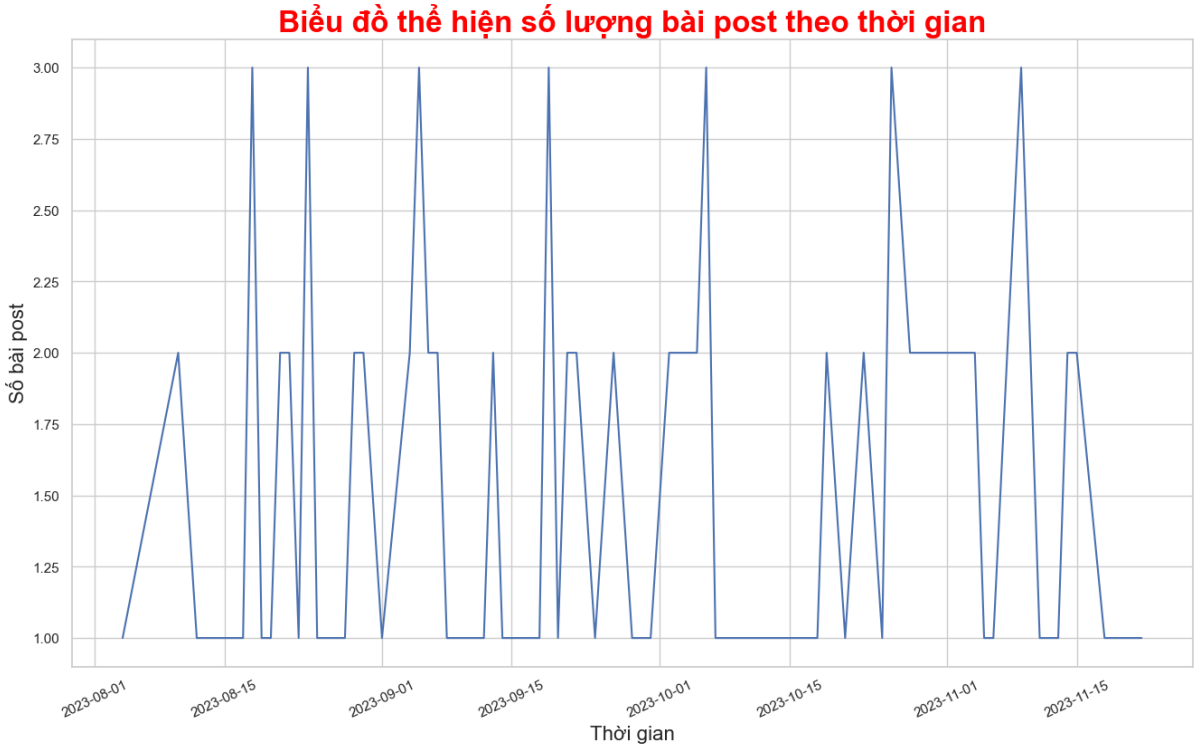
```
clean_df['is_live'].value_counts()
[204] ✓ 0.0s Python
... is_live
False    111
Name: count, dtype: int64

clean_df['was_live'].value_counts()
[205] ✓ 0.0s Python
... was_live
False    111
Name: count, dtype: int64
```

Giá trị của 2 cột này là False hết, tức là trong 111 bài gần đây, không có bài đăng nào là đang hoặc đã từng livestream.

- Sự thay đổi số lượng bài đăng theo thời gian

```
[206] time_post['date'].value_counts()
      ✓ 0.0s                                                    Python

... date
    2023-10-26    3
    2023-08-24    3
    2023-09-19    3
    2023-10-06    3
    2023-09-05    3
                 ..
    2023-11-21    1
    2023-09-28    1
    2023-09-24    1
    2023-09-20    1
    2023-08-04    1
    Name: count, Length: 73, dtype: int64
```

```
    sns.set_theme(style="whitegrid")
    plt.subplots(figsize=(16, 9))
    time_post['date'].value_counts().sort_index().plot(kind= 'line', figsize= (16, 9))
    plt.xticks(rotation = 25)
    plt.xlabel('Thời gian', fontsize=16)
    plt.ylabel('Số bài post', fontsize=16)
    plt.title('Biểu đồ thể hiện số lượng bài post theo thời gian', fontsize=24, color='red', fontweight='bold')
[207] ✓ 0.3s                                                    Python

... Text(0.5, 1.0, 'Biểu đồ thể hiện số lượng bài post theo thời gian')
```



Số lượng bài post tăng giảm không liên tục, ngày nhiều nhất có 3 bài đăng, ít nhất có một bài đăng

- Khoảng thời gian trong ngày mà fanpage hay đăng bài post

```
time_post['hour'].value_counts()
```

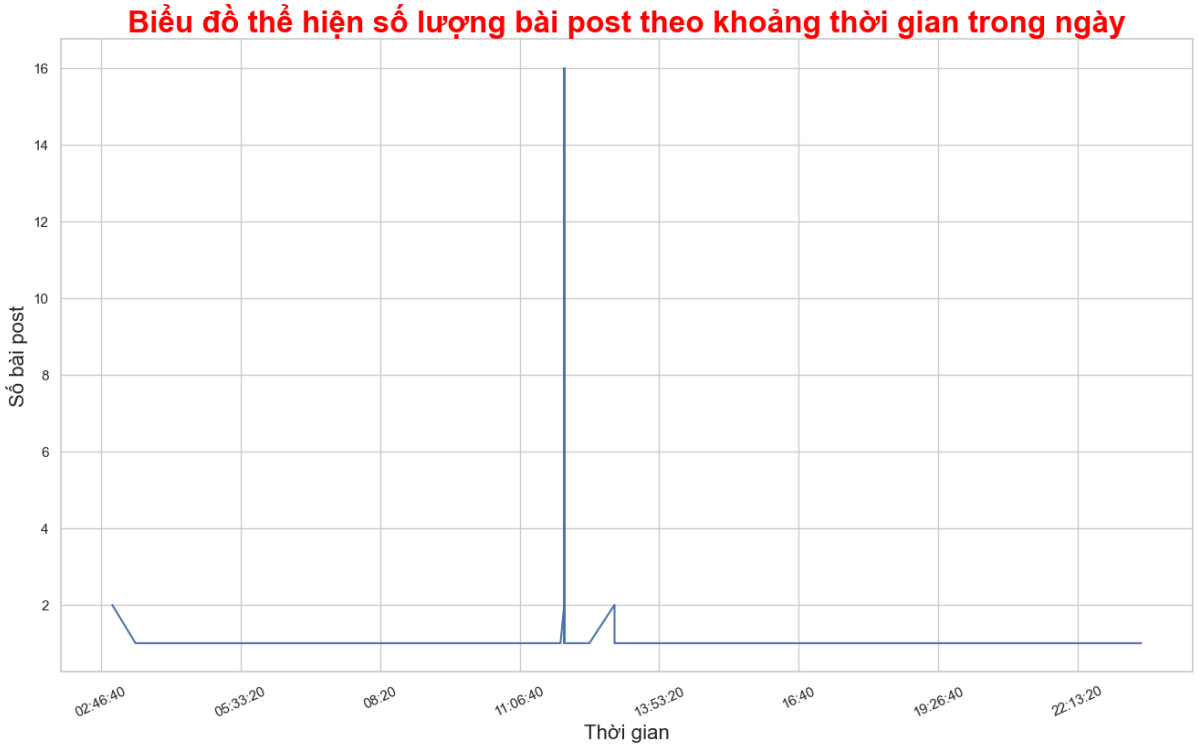
```
[208] ✓ 0.0s                                                    Python
```

```
hour
12:00:01    16
12:00:02     7
13:00:02     2
12:00:05     2
13:00:01     2
            ..
14:14:36     1
06:00:27     1
08:34:22     1
03:27:51     1
08:50:15     1
Name: count, Length: 84, dtype: int64
```

```
sns.set_theme(style="whitegrid")
plt.subplots(figsize=(16, 9))
time_post['hour'].value_counts().sort_index().plot(kind= 'line', figsize= (16, 9))
plt.xticks(rotation = 25)
plt.xlabel('Thời gian', fontsize=16)
plt.ylabel('Số bài post', fontsize=16)
plt.title('Biểu đồ thể hiện số lượng bài post theo khoảng thời gian trong ngày', fontsize=24, color='red', fontweight='bold')
```

```
[209] ✓ 0.2s                                                    Python
```

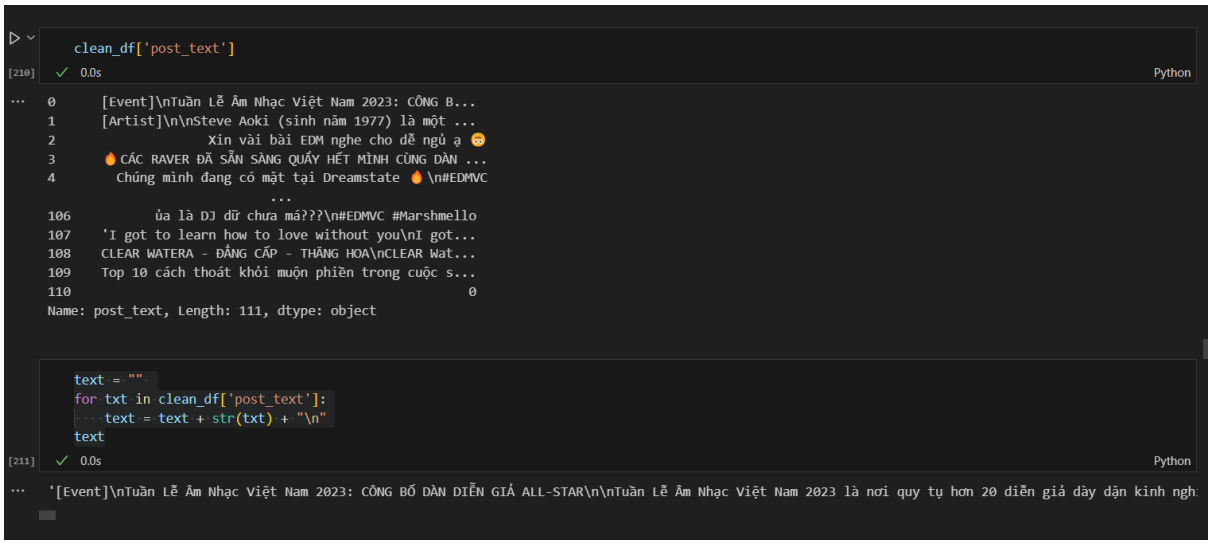
```
Text(0.5, 1.0, 'Biểu đồ thể hiện số lượng bài post theo khoảng thời gian trong ngày')
```



Theo biểu đồ trên, ta có thể thấy được trang hay đăng bài trong khoảng thời gian từ 11h – 14h. Đây là khoảng thời gian nghỉ trưa của đa số người, nên admin fanpage cũng tranh thủ khoảng thời gian rảnh này để đăng bài. Khoảng thời gian này nhiều người dùng Facebook, nên đăng bài vào thời gian này cũng giúp thu hút sự chú ý và đẩy mạnh sự tương tác của mọi người với bài viết hơn là đăng bài vào giờ hành chính, khi mà mọi người vẫn còn đi học, đi làm...

- Phân tích nội dung bài viết

Ta sẽ lấy ra toàn bộ phần văn bản của bài viết và lưu nó thành một chuỗi



```
clean_df['post_text']
```

```
[210] ✓ 0.0s Python

0     [Event]\nTuần Lễ Âm Nhạc Việt Nam 2023: CÔNG B...
1     [Artist]\n\nSteve Aoki (sinh năm 1977) là một ...
2                  Xin vài bài EDM nghe cho dễ ngủ ạ 😳
3     🔥CÁC RAVER ĐÃ SẴN SÀNG QUẨY HẾT MÌNH CÙNG DÀN ...
4       Chúng mình đang có mặt tại Dreamstate 🔥\n#EDMVC
                            ...
106           ủa là DJ dữ chưa má???\n#EDMVC #Marshmello
107    'I got to learn how to love without you\nI got...
108    CLEAR WATERA - ĐẲNG CẤP - THĂNG HOA\nCLEAR Wat...
109    Top 10 cách thoát khỏi muộn phiền trong cuộc s...
110                                                    0
Name: post_text, Length: 111, dtype: object
```

```
text = ""
for txt in clean_df['post_text']:
    text = text + str(txt) + "\n"
text
```

```
[211] ✓ 0.0s Python

'[Event]\nTuần Lễ Âm Nhạc Việt Nam 2023: CÔNG BỐ DÀN DIỄN GIẢ ALL-STAR\n\nTuần Lễ Âm Nhạc Việt Nam 2023 là nơi quy tụ hơn 20 diễn giả dày dặn kinh ngh
```

Sau đó, ta sẽ tạo một list các stopword tiếng Việt từ file.

Stopword hiểu đơn giản là các từ có tần số xuất hiện nhiều như và, của, nên... các từ này thường mang ít giá trị ý nghĩa và không khác nhau nhiều trong các văn bản khác nhau.

File stopword: stopwords.txt

Nguồn: https://github.com/stopwords/vietnamese-stopwords/blob/master/vietnamese-stopwords.txt

Xem thêm Output ở project

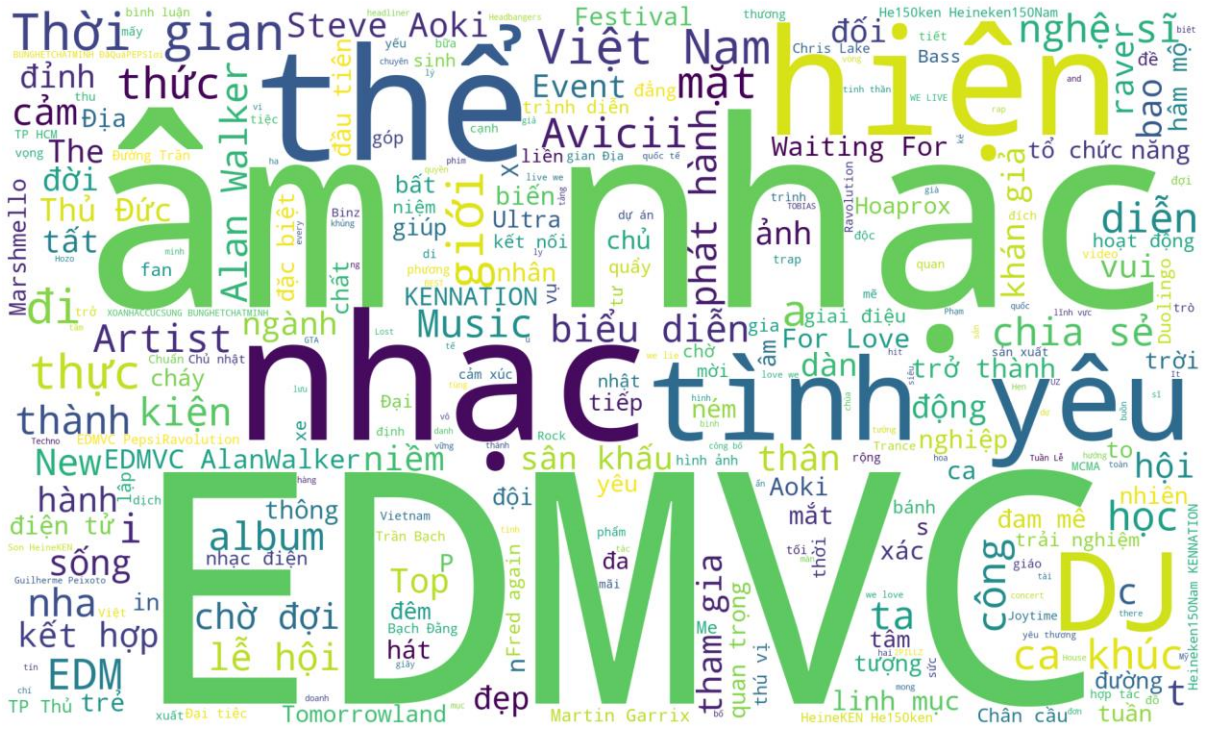
```
def get_stopwords_list(stop_file_path):
    with open(stop_file_path, 'r', encoding="utf-8") as f:
        stopwords = f.readlines()
        stop_set = set(m.strip() for m in stopwords)
        return list(frozenset(stop_set))

STOPWORDS = get_stopwords_list('Data/stopwords.txt')
STOPWORDS
```

```
[212] ✓ 0.0s Python

['quá bộ',
 'trả',
 'điểm chính',
 'tựu trung',
 'lấy xuống',
 'đúng',
 'thứ',
 'tuy vậy',
 'xoành xoạch',
 'dữ',
 'sẽ hay',
 'vâng',
 'nhanh lên',
 'cơ chi',
```

Vẽ hình ảnh đám mây thể hiện tần suất xuất hiện của các từ trong văn bản bằng thư viện WordCloud

```
wordcloud = WordCloud(stopwords=STOPWORDS,
                      background_color='white',
                      max_words=300,
                      width=2000, height=1200
                      ).generate(text)
plt.figure(figsize=(40,20))
plt.clf()
plt.imshow(wordcloud)
plt.axis('off')
plt.show()
```

[213] ✓ 4.7s Python



Độ lớn của từ thể hiện tần suất xuất hiện của các từ lớn hay nhỏ.

Nhìn vào hình trên, ta có thể thấy được các từ như âm nhạc, EDMVC, nhạc, tình yêu... xuất hiện nhiều trong các bài viết. Điều này là hợp lý bởi đây là một Fanpage về âm nhạc, cụ thể là EDM (Electronic dance music). Ta còn có thể thấy tên các DJ/Producer nổi tiếng hay được đề cập đến trong bài viết như Steve Aoki, Avicii, AlanWalker...

Với những người không biết khi nhìn vào hình ảnh này cũng có thể dễ dàng kết luận các bài viết của trang đều xoay quanh chủ đề về âm nhạc.

4.4. Phân tích người dùng

- Thông tin những người đã react bài viết của trang

```
reactors_df
```

[214] ✓ 0.0s Python

	name	link	type
0	Kiên Mai Íme	https://facebook.com/profile.php?id=1000940985...	like
1	Quinny Ng	https://facebook.com/profile.php?id=1000935124...	like
2	Dũng Lee	https://facebook.com/profile.php?id=1000916250...	like
3	Tuyết Mai	https://facebook.com/profile.php?id=1000913083...	like
4	Nguyễn Thùy Dương	https://facebook.com/profile.php?id=1000906921...	wow
...	...	...	...
8605	Best EDM Beats	https://facebook.com/BESTEDMBEATSS?eav=AfZy-a7...	like
8606	Nguyễn Trường An	https://facebook.com/profile.php?id=1000718717...	love
8607	Canalis Club	https://facebook.com/canalisclub.vn?eav=Afb2mS...	love
8608	Ngô Thắng	https://facebook.com/thang.ngo.1612007?eav=Afa...	love
8609	TechBeat Records	https://facebook.com/TechBeat.vn?eav=AfbIVkg6C...	love

8610 rows × 3 columns

- Tên 5 người hay react bài viết của trang nhiều nhất

```
reactors_df['name'].value_counts().nlargest(5)
```

[215] ✓ 0.0s Python

```
name
Lê Uyên Nhựt    42
Ordinary Guy    37
Dani Walker     33
Tùng Vũ         32
Oroka Wanako    30
Name: count, dtype: int64
```

- Phân tích thói quen react của người dung tên Lê Uyên Nhựt

```
reactors_df.loc[(reactors_df['name'] == 'Lê Uyên Nhựt'), 'type'].value_counts()
```

[298] ✓ 0.0s Python

```
type
yêu thích    21
thích        15
haha          6
Name: count, dtype: int64
```

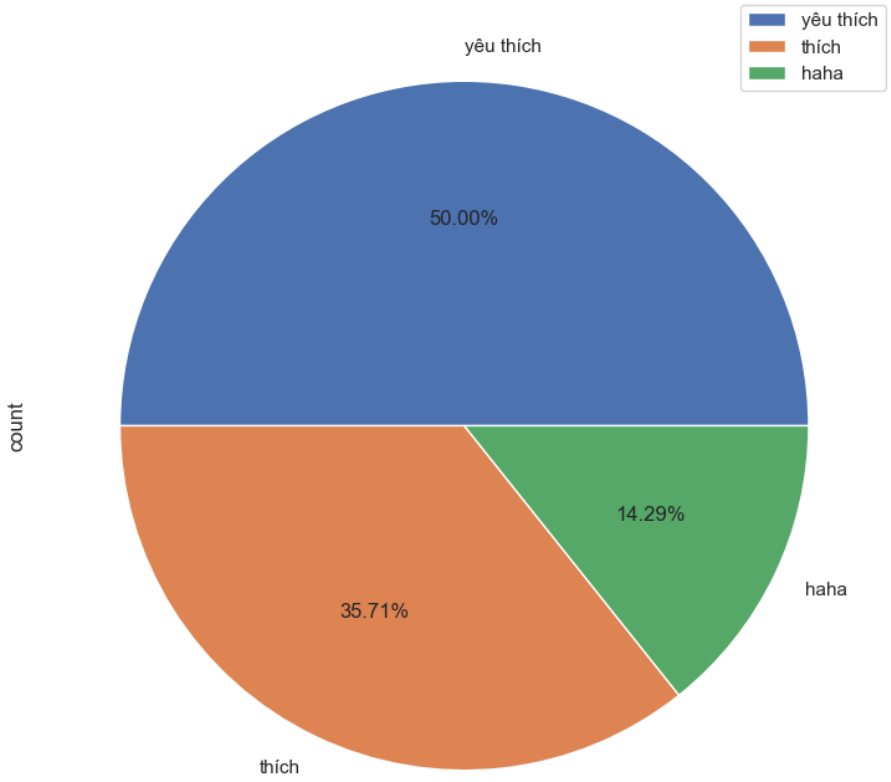
+ Code + Markdown

```
reactors_df.loc[(reactors_df['name'] == 'Lê Uyên Nhựt'), 'type'].value_counts().plot(kind= 'pie', figsize= (9, 9), autopct='%1.2f%%')
plt.title('Biểu đồ thể hiện tỉ trọng các loại reaction của Lê Uyên Nhựt', fontsize=24, color='red', fontweight='bold')
plt.legend()
```

[299] ✓ 0.1s Python

```
<matplotlib.legend.Legend at 0x1e9f39fe1d0>
```

Biểu đồ thể hiện tỉ trọng các loại reaction của Lê Uyên Nhựt



Qua biểu đồ trên, ta có thể thấy người dùng Lê Uyên Nhựt hay thả 3 loại reaction là thích, yêu thích và ha ha.

- Thông tin toàn bộ comment trong bài viết (Xem thêm Output ở project)

```
comments_full_df
```

0.0s Python

	comment_id	comment_url	commenter_id	commenter_url	commenter_name	commenter_meta
0	1061064065093152	https://facebook.com/1061064065093152	100064623192769	https://facebook.com/edmvco?eav=AfYeMqh7p8OHnb...	EDM Vietnam Community	Tác giả
1	746007517389746	https://facebook.com/746007517389746	100004233820245	https://facebook.com/thanhluan10497?eav=Afavm_...	Nguyễn Thành Luân	Người xem
2	1587591868715101	https://facebook.com/1587591868715101	100004028093197	https://facebook.com/profile.php?id=1000040280...	Duy Minh	Người xem
3	865432821690369	https://facebook.com/865432821690369	100064623192769	https://facebook.com/edmvco?eav=AfY-7w2v2_zcL-...	EDM Vietnam Community	Tác giả
4	1783294532107184	https://facebook.com/1783294532107184	100005519463021	https://facebook.com/phamtamphong2101?eav=AfZu...	Phạm Tâm Phong	Người xem
...	...	...	...	...	...	...

- Số comment có đính kèm ảnh

Số comment có đính kèm ảnh

+ Code + Markdown

```
comments_full_df.loc[comments_full_df['comment_image'] != 0].shape[0]
```

[301] ✓ 0.0s Python

```
513
```

- 5 comment có nhiều lượt reaction nhất (Xem thêm Output ở project)

5 comment có nhiều lượt reaction nhất

```
comments_full_df.nlargest(5, 'comment_reaction_count', keep= 'all')
```

[302] ✓ 0.0s Python

	comment_id	comment_url	commenter_id	commenter_url	commenter_name	commenter_meta	comment_t
505	1051885259492078	https://facebook.com/1051885259492078	100064623192769	https://facebook.com/edmvco?eav=AfbxwfNP3F52eh...	EDM Vietnam Community	Tác giả	Anh này tỏ
506	1063671265004244	https://facebook.com/1063671265004244	100064623192769	https://facebook.com/edmvco?eav=AfbxwfNP3F52eh...	EDM Vietnam Community	Tác giả	Nghiện EM được\n nghiện E thì không
1685	255912660596852	https://facebook.com/255912660596852	100005555409381	https://facebook.com/siuly.hien?eav=AfY_NbfNe6...	Hiền Lý Kim	Người xem	Nói anh Frequenci
1601	831662908605119	https://facebook.com/831662908605119	100003788948376	https://facebook.com/profile.php?id=1000037889...	Dyno Mike	Người xem	Có mấy th nhật nó thế thôi ch bac
260	845421870372214	https://facebook.com/845421870372214	100004351771777	https://facebook.com/Truong.rap?eav=AfZhV1gSUX...	Trường Nguyễn	Người xem	ngẫu hứng

- 5 ngày có số lượng người comment nhiều nhất

5 ngày có số lượng người comment cao nhất

```
comments_full_df['comment_time'].value_counts().nlargest(5)
```

[304] ✓ 0.0s Python

```
comment_time
2023-09-23    684
2023-08-23    441
2023-10-23    364
2023-11-09     79
2023-11-02     72
Name: count, dtype: int64
```

- Tên 5 người hay comment bài viết của trang nhiều nhất

Tên 5 người hay comment bài viết của trang nhiều nhất

```
comments_full_df['commenter_name'].value_counts().nlargest(5)
```

[303] ✓ 0.0s Python

```
commenter_name
EDM Vietnam Community    75
Hiền Lý Kim              19
Dang Chi Huong           14
Nguyễn Anh               11
Huỳnh Nhung              10
Name: count, dtype: int64
```

Ta có thể thấy admin của fanpage comment vào bài viết của trang nhiều nhất. Làm việc đó có thể là để tương tác với người xem được nhiều hơn.

- Thói quen comment của admin

Ta sẽ lưu nội dung comment của admin thành một chuỗi và vẽ ảnh đám mây

```
comment_text1 = ""
for txt in comments_full_df.loc[(comments_full_df['commenter_name'] == 'EDM Vietnam Community'), 'comment_text']:
    comment_text1 = comment_text1 + str(txt) + "\n"
comment_text1
```

✓ 0.0s Python

```
'Tuần Lễ Âm Nhạc Việt Nam - Vietnam Music Week 2023: Hội thảo Kinh doanh âm nhạc Việt Nam & Biểu diễn nhạc sống\n Thời gian: 20 - 24.11.2023\n Đị
```

```
wordcloud = WordCloud(stopwords=STOPWORDS,
                      background_color='white',
                      max_words=300,
                      width=2000, height=1200
                      ).generate(comment_text1)
plt.figure(figsize=(40,20))
plt.clf()
plt.imshow(wordcloud)
plt.axis('off')
plt.show()
```

✓ 5.1s Python

Nhìn vào ảnh trên, ta có thể nhận ra rằng admin fanpage hay comment vào bài đăng các đường link của trang web.

- Thói quen comment của người dùng tên Hiền Lý Kim

Làm tương tự như trên

```
comment_text2 = ""
for txt in comments_full_df.loc[(comments_full_df['commenter_name'] == 'Hiền Lý Kim'), 'comment_text']:
    comment_text2 = comment_text2 + str(txt) + "\n"
comment_text2
```

[307] ✓ 0.0s Python

```
'Thấy đáng mà trời 😊\nThế Anh Bùi thấy hội thảo và showcase hợp với em\nQuân Võ 😊😊\nChiến quá chiến Thanh Huyen\nMua vé chưa Thanh Huyen :))\nVide
```

```
wordcloud = WordCloud(stopwords=STOPWORDS,
                      background_color='white',
                      max_words=300,
                      width=2000, height=1200
                      ).generate(comment_text2)
plt.figure(figsize=(40,20))
plt.clf()
plt.imshow(wordcloud)
plt.axis('off')
plt.show()
```

[308] ✓ 1.9s Python

Nhìn vào ảnh trên, ta có thể nhận ra rằng người dùng này hay tag tên của người khác vào trong comment. Người dùng cũng hay viết tắt đại từ xưng hô (có thể là tôi, tao, tớ...) là t

- Thói quen comment của người dùng tên Dang Chi Huong

Làm tương tự như trên

```
comment_text3 = ""
for txt in comments_full_df.loc[(comments_full_df['commenter_name'] == 'Dang Chi Huong'), 'comment_text']:
    comment_text3 = comment_text3 + str(txt) + "\n"
comment_text3
```

[309] ✓ 0.0s Python

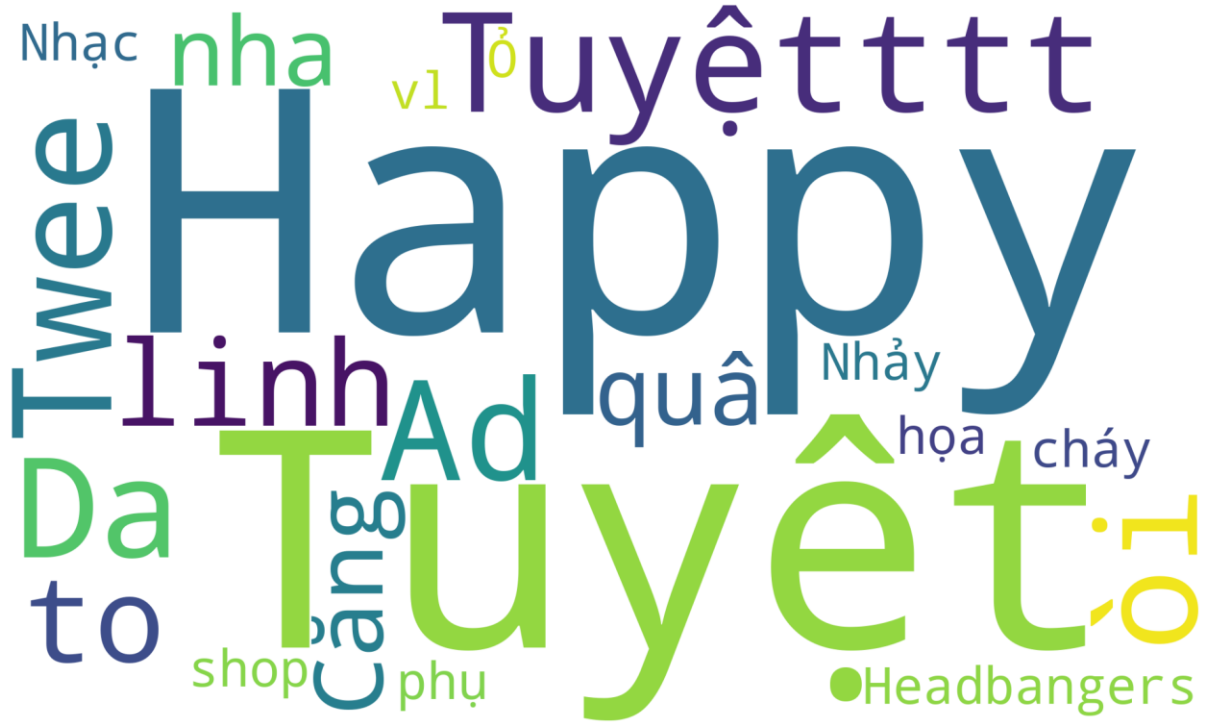
```
'Tuyệtttt\nAnh Da Twee đâu òi 😡\nAd lên giờ linh :))\nNói to quá nha\nHappy happy\nCàng thế Headbangers 🔥🔥\nỏ :))\n👀\nVậy mà mình chưa có :))\nN
```

+ Code + Markdown

Add Code Cell

```
wordcloud = WordCloud(stopwords=STOPWORDS,
                      background_color='white',
                      max_words=300,
                      width=2000, height=1200
                      ).generate(comment_text3)
plt.figure(figsize=(40,20))
plt.clf()
plt.imshow(wordcloud)
plt.axis('off')
plt.show()
```

[310] ✓ 1.6s Python



Nhìn vào ảnh trên, ta có thể nhận ra rằng người dùng này hay dùng những từ ngữ biểu cảm như tuyệt, tuyệtttt, òi, ỏ...

4.5. Dự đoán số lượt thích dựa trên số lượt reaction bằng phương pháp hồi quy tuyến tính (linear regression)

- Công thức:

$$thích = \beta_0 + \beta_1 * reaction\_count + \epsilon$$

- Tải và import các công cụ cần thiết

```
%pip install scikit-learn
```
[738] ✓ 1.3s Python

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: scikit-learn in c:\users\solit\appdata\roaming\python\python311\site-packages (1.3.2)
Requirement already satisfied: numpy<2.0,>=1.17.3 in c:\users\solit\appdata\roaming\python\python311\site-packages (from scikit-learn) (1.26.2)
Requirement already satisfied: scipy>=1.5.0 in c:\users\solit\appdata\roaming\python\python311\site-packages (from scikit-learn) (1.11.4)
Requirement already satisfied: joblib>=1.1.1 in c:\users\solit\appdata\roaming\python\python311\site-packages (from scikit-learn) (1.3.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\solit\appdata\roaming\python\python311\site-packages (from scikit-learn) (3.2.0)
Note: you may need to restart the kernel to use updated packages.

[notice] A new release of pip is available: 23.1.2 -> 23.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```
[739] ✓ 0.0s Python

- Đầu tiên, ta sẽ tạo mảng để lưu giá trị số lượt reaction và số lượt thích

```
X = reactions_df['reaction_count'].values

y = reactions_df['thích'].astype('int64').values
```
[737] ✓ 0.0s Python

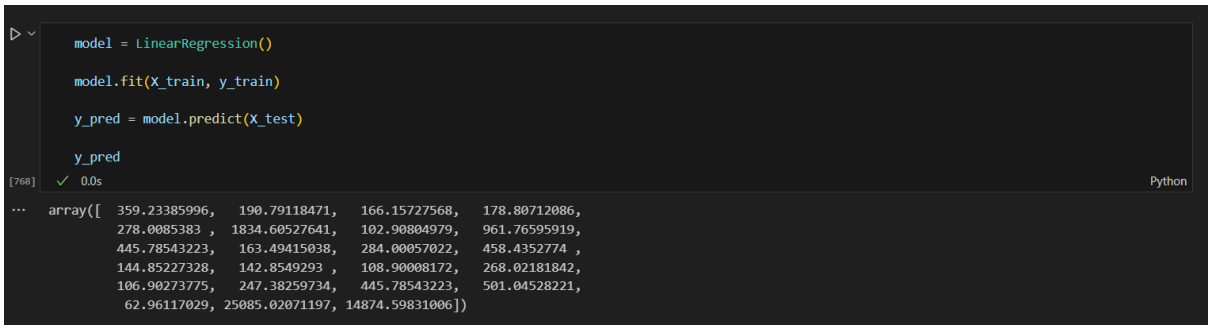
- Tiếp theo, chia bộ dữ liệu trên thành 2 tập: 1 tập dùng để huấn luyện và một tập dùng để test

```
print(X.shape)
print(y.shape)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=.2)

X_train = np.array(X_train).reshape(-1, 1)
X_test = np.array(X_test).reshape(-1, 1)

print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
```
[767] ✓ 0.0s Python

```
(111,)
(111,)
(88, 1)
(23, 1)
(88,)
(23,)
```

Ở đây, ta đã dùng hàm train_test_split() trong thư viện scikit-learn. Hàm này có tác dụng hỗ trợ việc chia dữ liệu một cách ngẫu nhiên và có thể tùy chỉnh tỷ lệ phần trăm của tập huấn luyện và tập kiểm tra.

- Sau đó đưa tập huấn luyện x_train, y_train vào trong mô hình linear regression



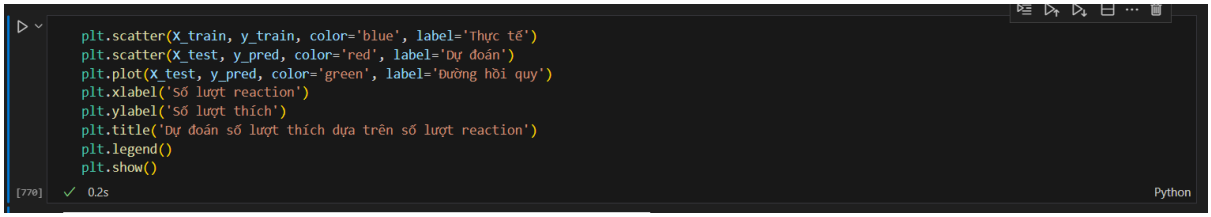
```
model = LinearRegression()

model.fit(X_train, y_train)

y_pred = model.predict(X_test)

y_pred
```

[768] ✓ 0.0s Python

```
array([  359.23385996,   190.79118471,   166.15727568,   178.80712086,
         278.0085383 ,  1834.60527641,   102.90804979,   961.76595919,
         445.78543223,   163.49415038,   284.00057022,   458.4352774 ,
         144.85227328,   142.8549293 ,   108.90008172,   268.02181842,
         106.90273775,   247.38259734,   445.78543223,   501.04528221,
          62.96117029, 25085.02071197, 14874.59831006])
```

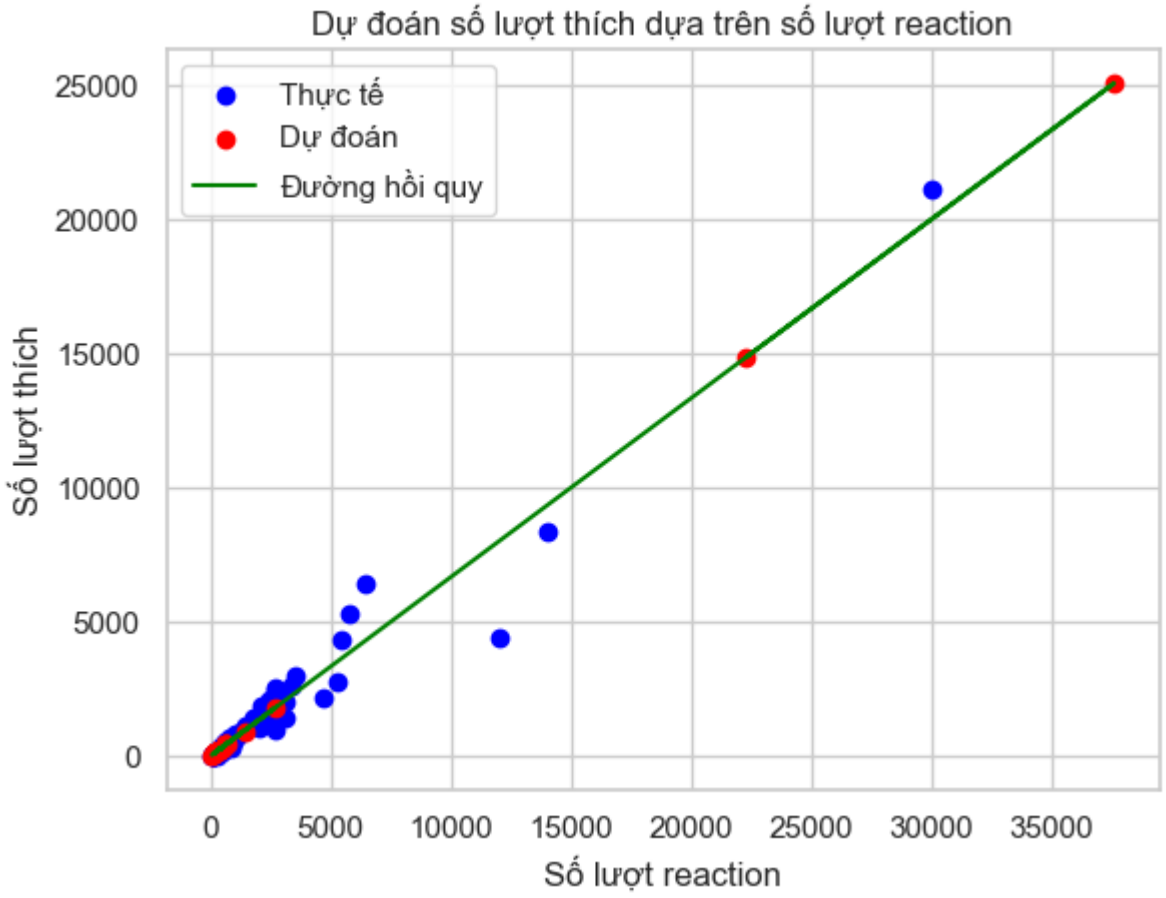
Ở đây, ta đã tạo một đối tượng LinearRegression của thư viện scikit-learn để huấn luyện mô hình. Mô hình được huấn luyện qua hàm fit() và dự đoán kết quả bằng hàm predict()

- Cuối cùng, ta sẽ vẽ biểu đồ để thấy rõ hơn kết quả dự đoán



```
plt.scatter(X_train, y_train, color='blue', label='Thực tế')
plt.scatter(X_test, y_pred, color='red', label='Dự đoán')
plt.plot(X_test, y_pred, color='green', label='Đường hồi quy')
plt.xlabel('Số lượt reaction')
plt.ylabel('Số lượt thích')
plt.title('Dự đoán số lượt thích dựa trên số lượt reaction')
plt.legend()
plt.show()
```

[770] ✓ 0.2s Python



Nhìn biểu đồ trên, ta thấy được các điểm dữ liệu thực tế nằm gần so với đường thẳng hồi quy. Điều này có thể cho thấy mô hình hồi quy tuyến tính phù hợp với dữ liệu.

Đường thẳng hồi quy hướng lên, chứng tỏ số lượt reaction và số lượt thích có sự tương quan dương, tức là số lượt reaction tăng thì số lượt thích cũng tăng và ngược lại. Điều này phù hợp với phân tích ở phần 4.2.3

----------*Hết*----------