



## Data Analyze Page Facebook.

⌚ Created	@November 8, 2023 11:32 AM
⌚ Type	Seminar
☑ Reviewed	<input type="checkbox"/>

# ĐẠI HỌC CÔNG NGHỆ ĐẠI HỌC QUỐC GIA HÀ NỘI. VIỆN TRÍ TUỆ NHÂN TẠO.

-----❖❖❖❖-----

## BÁO CÁO ĐỀ TÀI

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

Giảng viên hướng dẫn: Đặng Trần Bình, Nguyễn Văn Phi.

Sinh viên: Hồ Cảnh Quyền.

ĐẠI HỌC CÔNG NGHỆ ĐẠI HỌC QUỐC GIA HÀ NỘI.  
VIỆN TRÍ TUỆ NHÂN TẠO.

BÁO CÁO ĐỀ TÀI

DATA ANALYZE ABOUT PAGE FACEBOOK.

I. Giới Thiệu.

II. Phân tích và báo cáo.

1. Tổng quan về Page.
2. Thu thập dữ liệu.
3. làm sạch dữ liệu.

Thông tin dữ liệu thô:

Chiến lược crawl hiệu quả.

4. Phân tích dữ liệu.

phân tích reactions\_count thay đổi theo thời gian ?

phân tích các type reaction từ reactions ?

type react được sử dụng nhiều nhất?

Số lượt tương tác trong các bài đăng thay đổi như thế nào?

sự tương quan giữa shares và reactions\_count? và sự tương quan giữa comments với reactions\_count?

Bài viết có lượt tương tác nhiều nhất là ?

Các mốc thời gian trong ngày mà page thường xuyên đăng bài viết?

Sự tương quan giữa thời gian đăng bài và các ngày trong tuần?

Sự thay đổi số lượng bài post thay đổi qua các ngày ?

đâu là các từ khóa xuất hiện nhiều nhất trong bài đăng được thu thập (post\_text) ?

những cái tên tham gia tương tác bài viết chính?

Tổng hợp toàn bộ tên tương tác trong toàn bộ page kể cả bình luận và react bình luận?

những cái tên ảnh hưởng nhiều nhất đến dữ liệu?

Sự tương quan giữa Reaction, Comments, shares và độ dài bài viết?

## DATA ANALYZE ABOUT PAGE FACEBOOK.

### I. Giới Thiệu.

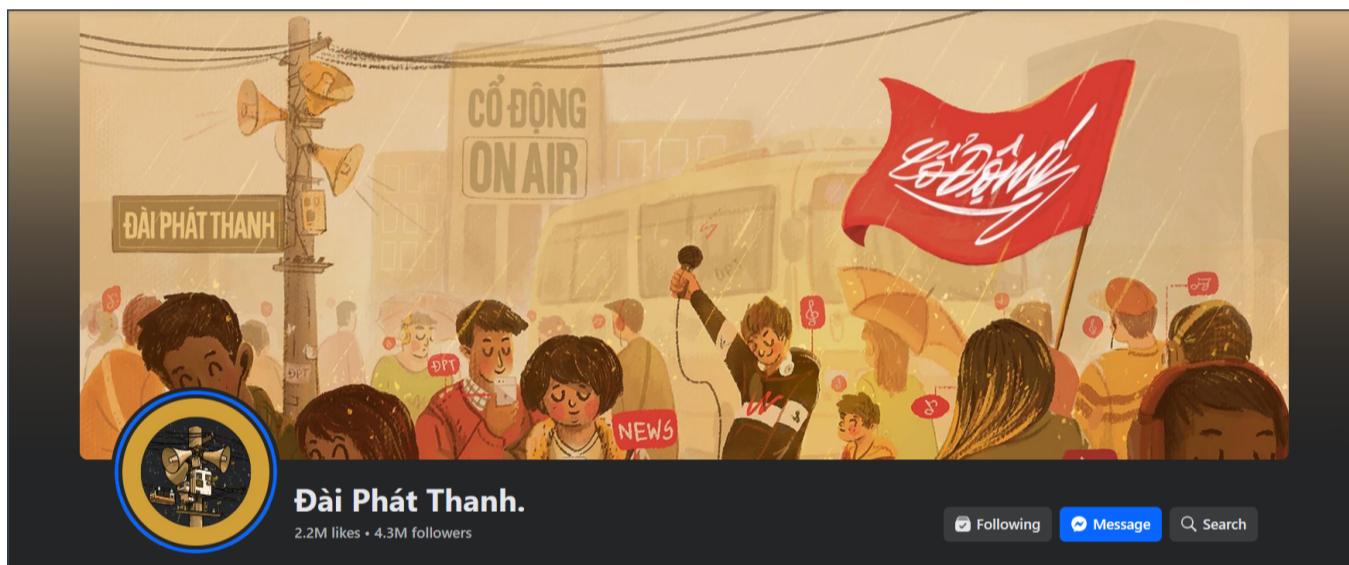
Trong bài báo cáo này, chúng ta sẽ tiến hành một phân tích dữ liệu đối với một trang Facebook cụ thể. Phân tích dữ liệu sẽ giúp chúng ta hiểu rõ hơn về người dùng, hoạt động và tương tác trên trang này. Chúng ta sẽ sử dụng các công cụ phân tích dữ liệu để tìm hiểu các yếu tố quan trọng như số lượng người theo dõi, tương tác và xu hướng hoạt động trên trang Facebook.

**Page** chúng ta phân tích ở đây có tên là “Đài Phát Thanh.”, bạn có thể theo đường dẫn này

[“https://www.facebook.com/daiphatthanh.sound”](https://www.facebook.com/daiphatthanh.sound) để xem page để hiểu hơn về dữ liệu mà chúng tôi sẽ phân tích sắp tới.

### II. Phân tích và báo cáo.

#### 1. Tổng quan về Page.



- Page vào ngày 27/11 có 2,2M likes và 4,3M followers.
- Thể loại page: Trang web tin tức và truyền thông
- Chủ đề : Âm nhạc, Nghệ thuật và Giải trí.

#### 2. Thu Thập dữ liệu.

A screenshot of the GitHub project page for 'facebook-scrapers' version 0.2.59. The page has a blue header with the GitHub logo, a search bar, and navigation links for 'Help', 'Sponsors', 'Log in', and 'Register'. The main title is 'facebook-scrapers 0.2.59'. Below the title, there is a button with the text 'pip install facebook-scrapers' and a download icon. To the right of the download button is a green button with a checkmark and the text 'Latest version'. Below these buttons, the text 'Released: Aug 31, 2022' is displayed. At the bottom of the page, a gray banner states 'Scrape Facebook public pages without an API key'.

Bây giờ chúng ta có thể lấy dữ liệu từ Facebook bằng thư viện `facebook_scrapers`. Chúng tôi sẽ sử dụng chức năng `get_posts` để lấy các bài đăng từ trang fan hâm mộ. Hàm này sẽ trả về một danh sách các từ điển, trong đó mỗi từ điển đại diện cho một bài viết. Chúng tôi sẽ lưu danh sách từ điển này vào một tệp json. Bạn có thể tìm thêm thông tin về những gì bạn có thể làm với thư viện `facebook_scrapers` tại đây: <https://github.com/kevinzg/facebook-scrapers>

Thực hiện crawl data thực hiện chủ yếu qua hai bước chính:

Define variables.

FANPAGE\_LINK : Liên kết đến trang mà chúng tôi muốn thu thập dữ liệu từ đó. Điều này có thể được tìm thấy bằng cách vào trang fan hâm mộ và sao chép liên kết từ thanh địa chỉ.

**COOKIE\_PATH** : Đường dẫn đến tệp cookie mà chúng tôi sẽ sử dụng để xác thực với Facebook. Tệp cookie này có thể lấy được bằng cách đăng nhập vào Facebook và sao chép cookie từ trình duyệt.

**FOLDER\_NAME** : Tên của thư mục mà chúng tôi sẽ lưu dữ liệu vào. Thư mục này sẽ được tạo trong cùng thư mục với sổ ghi chép này.

Dưới đây là mẫu 1 trong nhiều lần tôi thực hiện crawl dữ liệu về.

## Hình ảnh minh họa.

```
from facebook_scraper import get_posts
import pandas as pd
import numpy as np
FANPAGE_LINK = "daiphatthanh.sound"
FOLDER_PATH = r>Data\
COOKIE_PATH = r"final_project\Data\www.facebook.com_cookies.txt"
PAGES_NUMBER = 100# Number of pages to crawl
```

```
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):
    print(post)
    post_list.append(post)
```

```
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
Traceback (most recent call last):
  File "c:\Users\Admin\miniconda\lib\site-packages\facebook_scrapers\utils.py", line 279, in safe_consume
    for item in generator:
  File "c:\Users\Admin\miniconda\lib\site-packages\facebook_scrapers\extractors.py", line 1140, in extract_comment_replies
    for action in data['payload']['actions']:
                                         ^^^^^^^^^^
TypeError: 'NoneType' object is not subscriptable
Traceback (most recent call last):
  File "c:\Users\Admin\miniconda\lib\site-packages\facebook_scrapers\utils.py", line 279, in safe_consume
    for item in generator:
  File "c:\Users\Admin\miniconda\lib\site-packages\facebook_scrapers\extractors.py", line 1140, in extract_comment_replies
    for action in data['payload']['actions']:
                                         ^^^^^^^^^^
TypeError: 'NoneType' object is not subscriptable
Traceback (most recent call last):
  File "c:\Users\Admin\miniconda\lib\site-packages\facebook_scrapers\utils.py", line 279, in safe_consume
    for item in generator:
  File "c:\Users\Admin\miniconda\lib\site-packages\facebook_scrapers\extractors.py", line 1140, in extract_comment_replies
    for action in data['payload']['actions']:
                                         ^^^^^^^^^^
TypeError: 'NoneType' object is not subscriptable
```

```
# in trong tin để xem dữ liệu cơ bản.  
post_df_full.info()
```

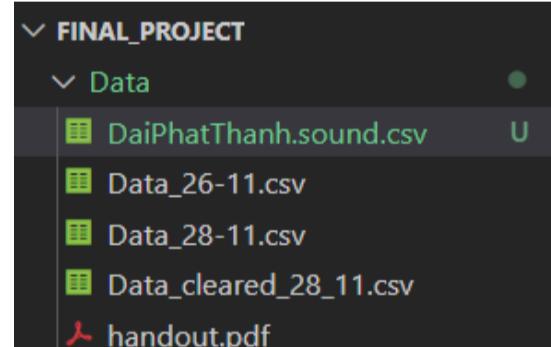
```
-----  
LoginRequired                                     Traceback (most recent call last)  
d:\projects\python\dash_data_analyze\final_project\crawler\data.ipynb Cell 2 line 1  
    7 PAGES_NUMBER = 1000 Number of pages to crawl  
    9 post_list = []  
--> 10 for post in get_posts(FACEBOOK_LINK,  
    11         options={"comments": True, "reactions": True, "allow_extra_requests": True},  
    12         extra_info=True, pages=PAGES_NUMBER, cookies=COOKIE_PATH):  
    13     print(post)  
    14     post_list.append(post)  
  
File c:\Users\Admin\miniconda3\lib\site-packages\facebook_scrapers\facebook_scrapers.py:1114, in FacebookScrapers._generic_get_posts(self, extract_post_fn,  
1111     counter = itertools.count(0) if page_limit is None else range(page_limit)  
1112     logger.debug("Starting to iterate pages")  
--> 1114 for i, page in zip(counter, iter_pages_fn()):  
1115     logger.debug("Extracting posts from page %s", i)  
1116     for post_element in page:  
  
File c:\Users\Admin\miniconda3\lib\site-packages\facebook_scrapers\page_iterators.py:87, in generic_iter_pages(start_url, page_parser_cls, request_fn, **  
85     try:  
86         logger.debug("Requesting page from: %s", next_url)  
--> 87         response = request_fn(next_url)  
88     break  
89 except HTTPError as e:  
...  
944     )  
945     return response  
946 except RequestException as ex:
```

## Convert list of dicts to df

Bây giờ chúng ta có thể chuyển đổi danh sách từ điển thành khung dữ liệu gấu trúc. Chúng tôi sẽ sử dụng thư viện pandas để thực hiện việc này. Chúng tôi cũng sẽ lưu khung dữ liệu vào tệp xlxs hoặc csv.

```
# Initialize dataframe to scrape Facebook post
post_df_full = pd.DataFrame(columns=post_list[0].keys(),
    index=range(len(post_list)), data=post_list)

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



### **3. làm sạch dữ liệu.**



### Thông tin dữ liệu thô:

Ban đầu :

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 205 entries, 0 to 204
Data columns (total 53 columns):

```

	Column	Non-Null Count	Dtype
0	post_id	205	non-null int64
1	text	204	non-null object
2	post_text	204	non-null object
3	shared_text	0	non-null float64
4	original_text	0	non-null float64
5	time	205	non-null object
6	timestamp	205	non-null float64
7	image	146	non-null object
8	image_lowquality	205	non-null object
...			
51	video_ids	20	non-null object
52	videos	20	non-null object

dtypes: bool(3), float64(20), int64(7), object(23)  
memory usage: 80.8+ KB

Khi làm sạch :

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 204 entries, 0 to 203
Data columns (total 11 columns):
```

### **Column Non-Null Count Dtype**

	Column	Non-Null Count	Dtype
0	post_id	204	non-null int64
1	post_text	204	non-null object
2	time	204	non-null object
3	timestamp	204	non-null float64
4	images_description	204	non-null object
5	comments	204	non-null int64
6	shares	204	non-null int64
7	comments_full	204	non-null object
8	reactors	204	non-null object
9	reactions	204	non-null object
10	reaction_count	204	non-null int64

dtypes: float64(1), int64(4), object(6)  
memory usage: 17.7+ KB

### Chiến lược crawl hiệu quả.

Trong quá trình crawl dữ liệu thường hay bị lỗi dữ liệu bởi nhiều nguyên nhân như tốc độ kết nối mạng, độ ổn định của mạng, trình bảo mật của facebook, ... khiến chúng ta gặp vấn đề và khó khăn trong việc lấy dữ liệu dẫn đến dữ liệu không đúng và rời rạc không đầy đủ.

Tôi đã chọn giải pháp đó là crawl dữ liệu nhiều lần và thực hiện xóa những bài viết trùng lặp, lỗi và kết nối những dữ liệu đã bị lỗi với nhau thành một dữ liệu hoàn chỉnh.

Dưới đây là mẫu của 1 trong nhiều lần tôi thực hiện concat để ghép nối dữ liệu .

```
# dataframe
# dữ liệu ban đầu.
df1 = pd.read_csv(r'Data\Data_26-11.csv')
```

Dữ liệu cần thêm vào và xử lý:

```
# dữ liệu cần xử lý để thêm vào.
df2 = pd.read_csv(r'Data\data28_11_chua_xu_li.csv')
```

```
ls = [] # lưu index cần xóa
for i in range(30,40):
    ls.append(i)
data = pd.DataFrame(data=df2)
res = data.drop(ls, axis='index')
# dữ liệu df2 sau khi xử lý
res
```

```
# concat data.
newData = pd.concat([df1, res], axis=0, ignore_index=True)
newData.info()
```

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 40 entries, 0 to 39

Data columns (total 51 columns):

#	Column	Non-Null Count	Dtype
0	post_id	40 non-null	object
1	text	40 non-null	object
2	post_text	40 non-null	object

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 175 entries, 0 to 174

Data columns (total 53 columns):

#	Column	Non-Null Count	Dtype
0	post_id	175 non-null	int64
1	text	174 non-null	object
2	post_text	174 non-null	object

Sau khi xử lý và concat data.

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 204 entries, 0 to 203

Data columns (total 11 columns):

#	Column	Non-Null Count	Dtype
0	post_id	204 non-null	int64
1	post_text	204 non-null	object
2	time	204 non-null	object
3	timestamp	204 non-null	float64

## 4. Phân tích dữ liệu.

CHUẨN BỊ DỮ LIỆU CHO BÀI PHÂN TÍCH.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv(r'Data\Data_cleared.csv')
```

### phân tích reactions\_count thay đổi theo thời gian ?

Đây là biểu đồ thể hiện lượng reaction qua timestamp. Nhận thấy lượng tương tác reactions thay đổi không đồng đều qua các ngày.

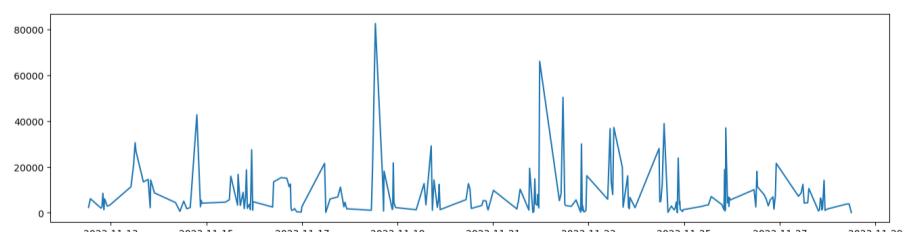
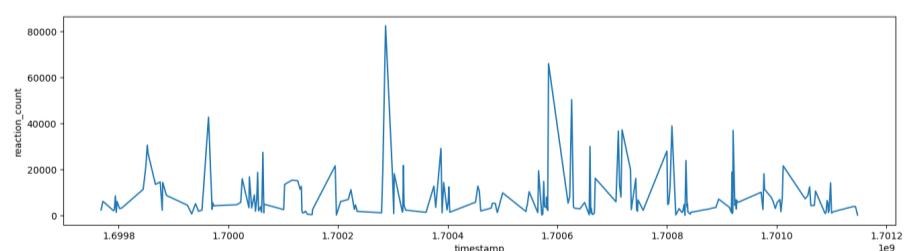
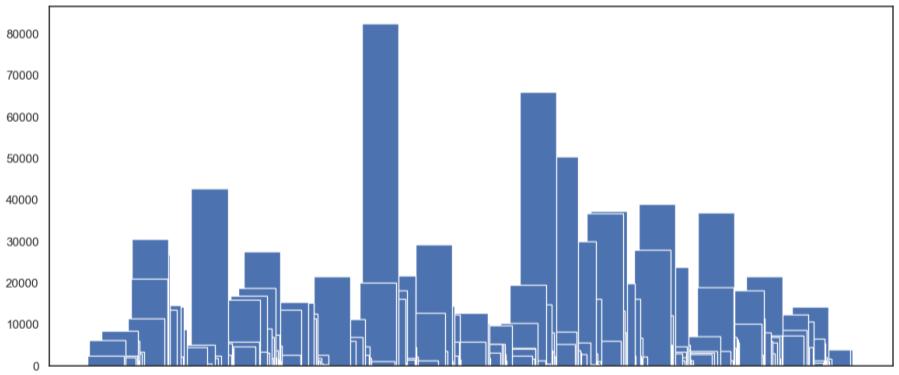
Timestamp là biểu diễn của time đăng tải bài viết được trực quan qua bảng:

	time	timestamp	reaction_count
0	2023-11-28 11:58:32	1.701148e+09	123
1	2023-11-28 10:53:40	1.701144e+09	3845
2	2023-11-28 09:48:24	1.701140e+09	3864
3	2023-11-27 23:59:52	1.701104e+09	1687

Tham chiếu biểu đồ với data dưới đây thì sẽ có cái nhìn trực quan hơn về reaction\_count.

	post_text	time	timestamp	reaction_count
max	THẦY GIÁO NHẬT "NGẠI NGÙNG" KHI LẦN ĐẦU ĐƯỢC T...	2023-11-18 12:52:40	1.700287e+09	82515
min	Nhìn sâu vào chặng đường tìm kiếm bản thân mà ...	2023-11-24 20:31:58	1.700833e+09	64

```
count    204.000000
mean     8372.901961
std      10997.212701
min      64.000000
25%     2091.250000
50%     4471.000000
75%     10734.000000
max     82515.000000
Name: reaction_count, dtype: float64
```



## phân tích các type reaction từ reactions ?

Để phân tích reactions ta thực hiện lấy dữ liệu trong cột reactions và lưu dữ vào 1 data Frame.

```
post_df_full1 = df
post_df_full1['reactions'] = post_df_full1['reactions'].apply(lambda x : dict(eval(x)))
post_df_full_reactions = post_df_full1['reactions'].apply(pd.Series )
post_df_full_reactions
```

	like	love	haha	care	wow	sad	angry
0	88.0	18.0	16.0	1.0	NaN	NaN	NaN
1	2805.0	1004.0	7.0	27.0	2.0	NaN	NaN
2	2877.0	30.0	935.0	5.0	3.0	14.0	NaN
3	1394.0	268.0	17.0	4.0	3.0	1.0	NaN
4	843.0	191.0	4.0	4.0	1.0	1.0	NaN
...	...	...	...	...	...	...	...
199	6042.0	2339.0	10.0	48.0	2.0	51.0	NaN
200	1722.0	714.0	1.0	10.0	NaN	3.0	NaN
201	1704.0	322.0	4.0	6.0	NaN	2.0	NaN
202	4299.0	1770.0	14.0	41.0	3.0	1.0	NaN
203	1863.0	426.0	5.0	16.0	29.0	1.0	NaN

Mỗi reactions sẽ thuộc về một bài nên thực hiện liên kết bảng data Frame trên với thông tin data frame ban đầu.

```
# gộp các cột lại vào dataframe
post_df_full_with_reactions = pd.concat([post_df_full1, post_df_full_reactions], axis=1).drop('reactions', axis=1)
post_df_full_with_reactions[['post_text','like','love','haha','wow','sad','angry','care',
                           'shares','comments','reaction_count']]
```

	post_text	like	love	haha	wow	sad	angry	care	shares	comments	reaction_count
0	Dù là một nữ idol thì Jiyeon (T-ara) vẫn sẽ là...	88.0	18.0	16.0	NaN	NaN	NaN	1.0	4	4	123
1	ĐỘ TỘC XÂY TRƯỜNG ĐƯỢC RỒI!\\n\\nSáng 25/11, Đoà...	2805.0	1004.0	7.0	2.0	NaN	NaN	27.0	11	20	3845
2	Khi bạn là rapper mainstream nhưng vẫn muốn gi...	2877.0	30.0	935.0	3.0	14.0	NaN	5.0	44	54	3864
3	Thái VG trở lại với Andy Vũ trong MV "Điếc", d...	1394.0	268.0	17.0	3.0	1.0	NaN	4.0	32	17	1687
4	Hoàng Thuỳ Linh và Đen cực tình tứ trong MV mới...	843.0	191.0	4.0	1.0	1.0	NaN	4.0	32	20	1044
...	...	...	...	...	...	...	...	...	...	...	...
199	Mùa 1 The Masked Singer có 'Anh Thương Em Đến ...	6042.0	2339.0	10.0	2.0	51.0	NaN	48.0	730	283	8492
200	Có một Việt Nam cổ kính, hoài niệm cùng chuyện...	1722.0	714.0	1.0	NaN	3.0	NaN	10.0	90	33	2450
201	Nguyễn Hà trở lại với sản phẩm mới mang tên "N...	1704.0	322.0	4.0	NaN	2.0	NaN	6.0	23	7	2038
202	Chị đẹp HyunA "xả ảnh", khoe những món quà đán...	4299.0	1770.0	14.0	3.0	1.0	NaN	41.0	11	14	6128
203	Bác Nguyễn Ngọc Giao là bố của Nodey - chồng S...	1863.0	426.0	5.0	29.0	1.0	NaN	16.0	7	9	2340

204 rows × 11 columns

Xử lý các giá trị Nan và không hợp lệ.

```

for i in range(0, len(post_df_full_with_reactions['like'].values), 1):
    if pd.isna(post_df_full_with_reactions['like'].values[i]):
        post_df_full_with_reactions['like'].values[i] = 0

    if pd.isna(post_df_full_with_reactions['love'].values[i]):
        post_df_full_with_reactions['love'].values[i] = 0

    if pd.isna(post_df_full_with_reactions['haha'].values[i]):
        post_df_full_with_reactions['haha'].values[i] = 0

    if pd.isna(post_df_full_with_reactions['wow'].values[i]):
        post_df_full_with_reactions['wow'].values[i] = 0

    if pd.isna(post_df_full_with_reactions['sad'].values[i]):
        post_df_full_with_reactions['sad'].values[i] = 0

    if pd.isna(post_df_full_with_reactions['angry'].values[i]):
        post_df_full_with_reactions['angry'].values[i] = 0

    if pd.isna(post_df_full_with_reactions['care'].values[i]):
        post_df_full_with_reactions['care'].values[i] = 0

    if pd.isna(post_df_full_with_reactions['shares'].values[i]):
        post_df_full_with_reactions['shares'].values[i] = 0

    if pd.isna(post_df_full_with_reactions['comments'].values[i]):
        post_df_full_with_reactions['comments'].values[i] = 0

post_df_full_with_reactions[['post_text','like','love','haha','wow','sad',
                            'angry','care','shares','comments','reaction_count']]

```

Các giá trị nan sẽ được xử lý và chuyển thành 0. Qua đó ta có thể thấy cái nhìn trực quan và rõ ràng hơn về các dữ liệu liên quan đến bài post.

Dưới đây là bảng thể hiện reactions chi tiết cho từng bài post của page.

	post_text	like	love	haha	wow	sad	angry	care	shares	comments	reaction_count
0	Dù là một nữ idol thì Jieyon (T-ara) vẫn sẽ là...	88.0	18.0	16.0	0.0	0.0	0.0	1.0	4	4	123
1	ĐỘ TỐC XÂY TRƯỞNG ĐÚ/QC RỜI\nSáng 25/11, Đoà...	2805.0	1004.0	7.0	2.0	0.0	0.0	27.0	11	20	3845
2	Khi bạn là rapper mainstream nhưng vẫn muốn gi...	2877.0	30.0	935.0	3.0	14.0	0.0	5.0	44	54	3864
3	Thái VG trở lại với Andy Vũ trong MV "Điếc", d...	1394.0	268.0	17.0	3.0	1.0	0.0	4.0	32	17	1687
4	Hoàng Thúy Linh và Đen cực tình tứ trong MV mớ...	843.0	191.0	4.0	1.0	1.0	0.0	4.0	32	20	1044
...	...	...	...	...	...	...	...	...	...	...	...
199	Mùa 1 The Masked Singer có 'Anh Thương Em Đến ...	6042.0	2339.0	10.0	2.0	51.0	0.0	48.0	730	283	8492
200	Có một Việt Nam cổ kính, hoài niệm cùng chuyện...	1722.0	714.0	1.0	0.0	3.0	0.0	10.0	90	33	2450
201	Nguyễn Hà trở lại với sản phẩm mới mang tên "N...	1704.0	322.0	4.0	0.0	2.0	0.0	6.0	23	7	2038
202	Chị đẹp HyunA "xã ảnh", khoe những món quà đán...	4299.0	1770.0	14.0	3.0	1.0	0.0	41.0	11	14	6128
203	Bác Nguyễn Ngọc Giao là bố của Noddy - chồng S...	1863.0	426.0	5.0	29.0	1.0	0.0	16.0	7	9	2340

Tổng thành phần tương tác của data frame trên.

```

post_df_full_with_reactions[['like','love','haha','wow','sad',
                             'angry','care','shares','comments','reaction_count']].sum()

```

like	1039037.0
love	264056.0
haha	332023.0
wow	5447.0
sad	53632.0
angry	111.0
care	13766.0
shares	27968.0
comments	53150.0
reaction_count	1708072.0

dtype: float64

## type react được sử dụng nhiều nhất?

```

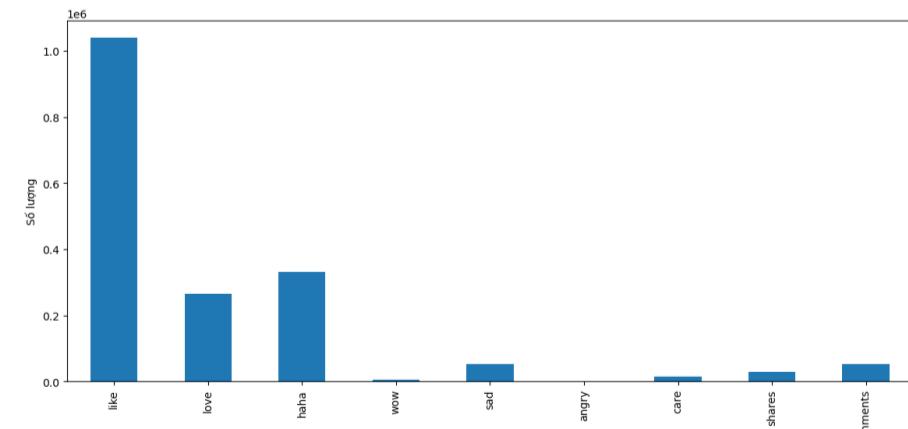
df_pie_reactions = post_df_full_with_reactions[['like','love','haha','wow',
                                                'sad','angry','care','shares','comments']]
df_pie_reactions.sum().plot(kind='pie', figsize=(14,6))

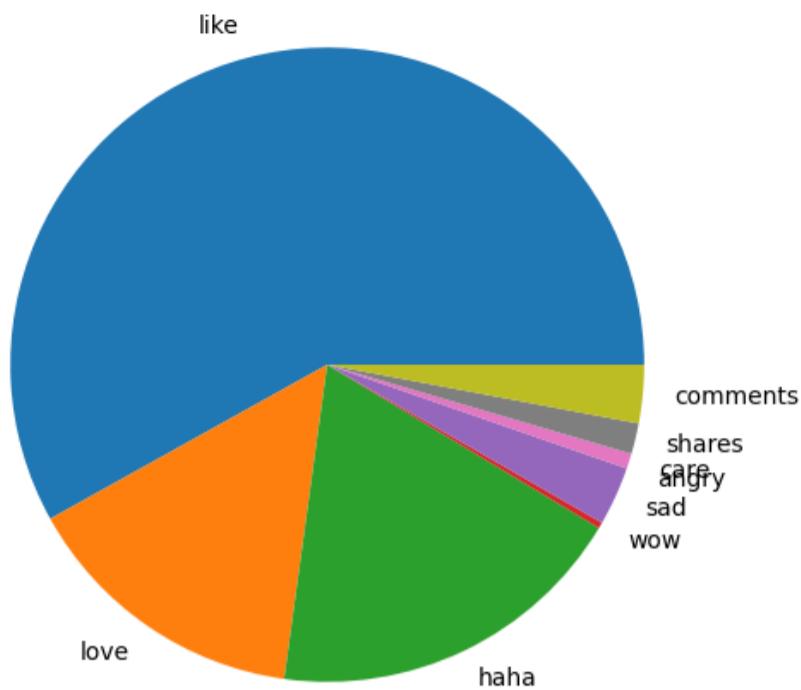
```

```

df_bar_reactions = post_df_full_with_reactions[['like','love','haha','wow',
                                                'sad','angry','care','shares','comments']]
ax = df_bar_reactions.sum().plot(kind='bar', figsize=(14,6))
ax.set_ylabel('Số lượng')

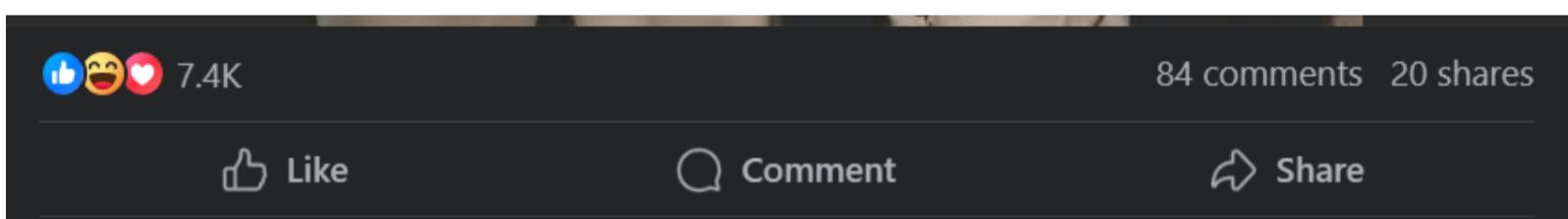
```





Cả hai biểu đồ trên cho thấy kiểu reactions được sử dụng nhiều nhất là like. Chứng tỏ phần lớn người dùng Facebook đều đồng tình thích thú với bài viết. Lý giải tại sao lượng tương tác like lại thường nhiều hơn các tương tác khác là vì

- Thứ nhất : Facebook để react là like làm đại diện nút reactions khi ta tương tác với bài viết và xu hướng lượt tin qua nhanh là xu thế của đa số người dùng mạng hiện nay nên người dùng sẽ dùng react dễ dàng sử dụng và mang sắc thái chung nhất để thả cho bài viết họ đọc qua.



- Thứ hai : nội dung của các bài đăng trong page này thường theo trend và xu hướng không gây ý kiến trái chiều hay kích động nên cũng dễ hiểu vì sao lượng tương tác angry gần như là con số không trong mỗi bài viết và chiếm tỉ lệ rất thấp.

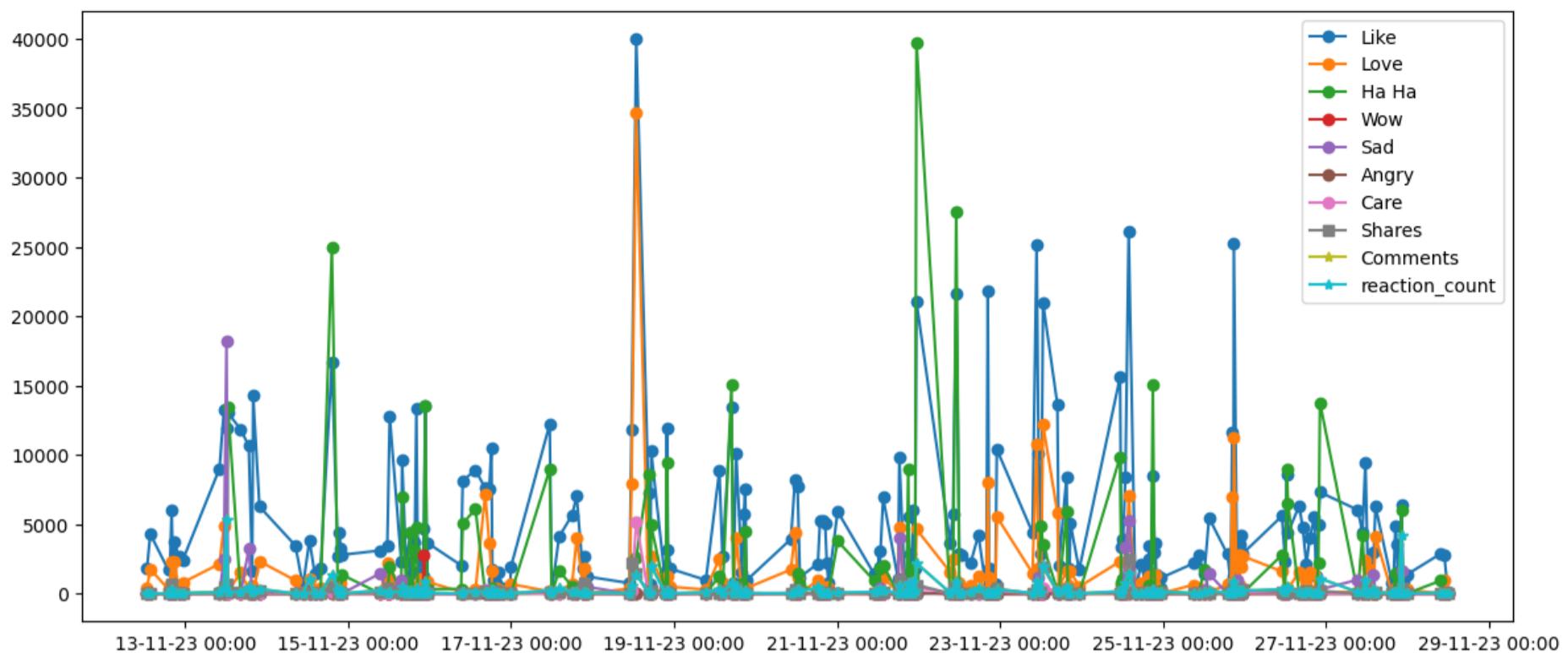
## Số lượt tương tác trong các bài đăng thay đổi như thế nào?

```

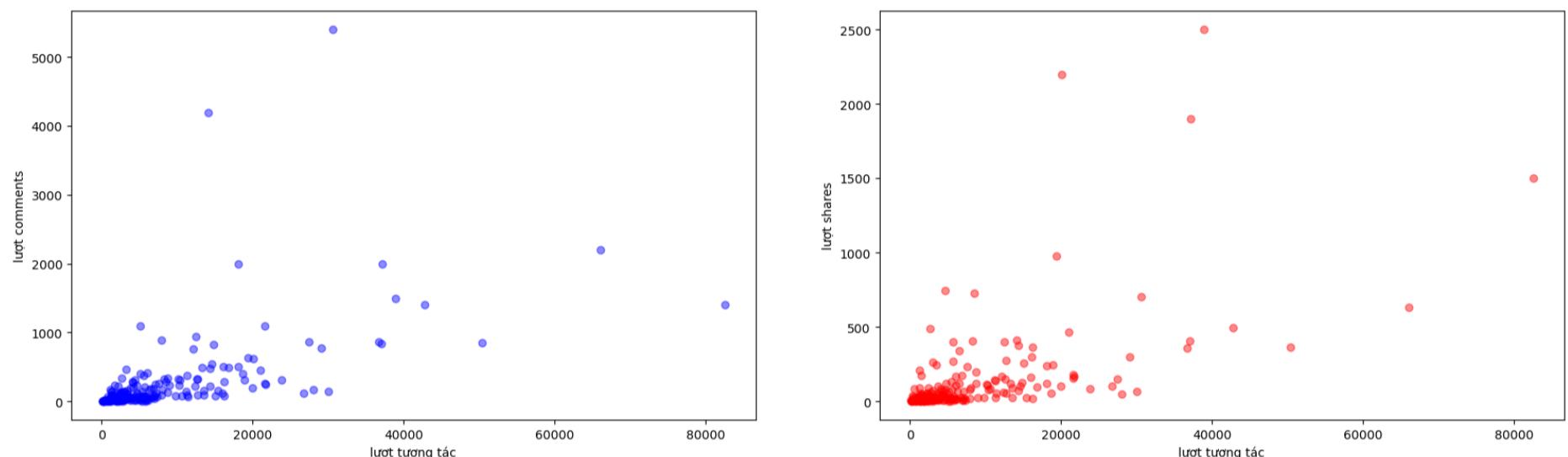
fig, ax = plt.subplots(figsize=(14, 6))
post_df_full_with_reactions['time'] = pd.to_datetime(post_df_full_with_reactions['time'])
ax.plot(post_df_full_with_reactions['time'], post_df_full_with_reactions['like'], label = "Like", marker="o")
ax.plot(post_df_full_with_reactions['time'], post_df_full_with_reactions['love'], label = "Love", marker="o")
ax.plot(post_df_full_with_reactions['time'], post_df_full_with_reactions['haha'], label = "Ha Ha", marker="o")
ax.plot(post_df_full_with_reactions['time'], post_df_full_with_reactions['wow'], label = "Wow", marker="o")
ax.plot(post_df_full_with_reactions['time'], post_df_full_with_reactions['sad'], label = "Sad", marker="o")
ax.plot(post_df_full_with_reactions['time'], post_df_full_with_reactions['angry'], label = "Angry", marker="o")
ax.plot(post_df_full_with_reactions['time'], post_df_full_with_reactions['care'], label = "Care", marker="o")
ax.plot(post_df_full_with_reactions['time'], post_df_full_with_reactions['shares'], label = "Shares", marker="s")
ax.plot(post_df_full_with_reactions['time'], post_df_full_with_reactions['comments'], label = "Comments", marker="*")
ax.plot(post_df_full_with_reactions['time'], post_df_full_with_reactions['reaction_count'], label = "reaction_count", marker="*")

plt.legend()
from matplotlib.dates import DateFormatter
ax.xaxis.set_major_formatter(DateFormatter('%d-%m-%y %H:%M'))

```



sự tương quan giữa shares và reactions\_count? và sự tương quan giữa comments với reactions\_count?



#### NHẬN XÉT:

- Hai biểu đồ phân tán này nhìn tương tự nhau thể hiện xu hướng comments và shares của người dùng có điểm tương đồng với nhau.
- Có rất ít bài mà có lượng comments, react, shares nhiều đồng thời.

#### Bài viết có lượt tương tác nhiều nhất là ?

	post_id	post_text	time	timestamp	images_description	comments	shares	comments_full	reactors	reactions	reaction_count
135	909089207250135	THẦY GIÁO NHẬT "NGÀI NGÙNG" KHI LẦN ĐẦU ĐƯỢC T...	2023-11-18 12:52:40	1.700287e+09		1400	1500	[{"comment_id": "1460715988059959", "name": "Minh Hằng", "link": "https://facebook.com/1460715988059959", "comment_u..."}]	{'like': 39963, 'love': 34694, 'haha': 2593, ...}		82515

	post_id	post_text	time	timestamp	images_description	comments	shares	comments_full	reactors	reactions	reaction_count	Day_name
52	912335590258830	Nhìn sâu vào chặng đường tìm kiếm bản thân mà ...	2023-11-24 20:31:58	1.700833e+09		4	6	[{"comment_id": "3221295438180288", "name": "Phạm Ngọc Minh Quân", "link": "http://facebook.com/3221295438180288", "comment_u..."}]	{"like": 50, "love": 14}		64	Friday

	post_text	time	timestamp	reaction_count
max	THẦY GIÁO NHẬT "NGẠI NGÙNG" KHI LẦN ĐẦU ĐƯỢC T...	2023-11-18 12:52:40	1.700287e+09	82515
min	Nhin sâu vào chặng đường tìm kiếm bản thân mà ...	2023-11-24 20:31:58	1.700833e+09	64

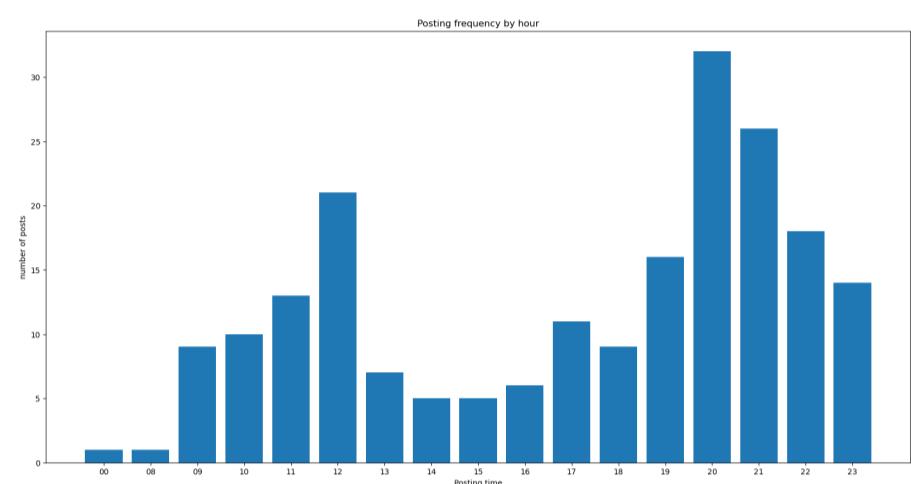
NHẬN XÉT:

- bài đăng nhiều tương tác nhất có ngày gần ngày 20/11 là một ngày lễ lớn để tri ân thầy cô giáo của việt nam.
- Nội dung bài đăng liên quan đến thầy giáo 1 phần nào đó phù hợp với điều kiện ngày lễ 20/11 sắp đến nên được quan tâm nhiều.

### Các mốc thời gian trong ngày mà page thường xuyên đăng bài viết?

NHẬN XÉT:

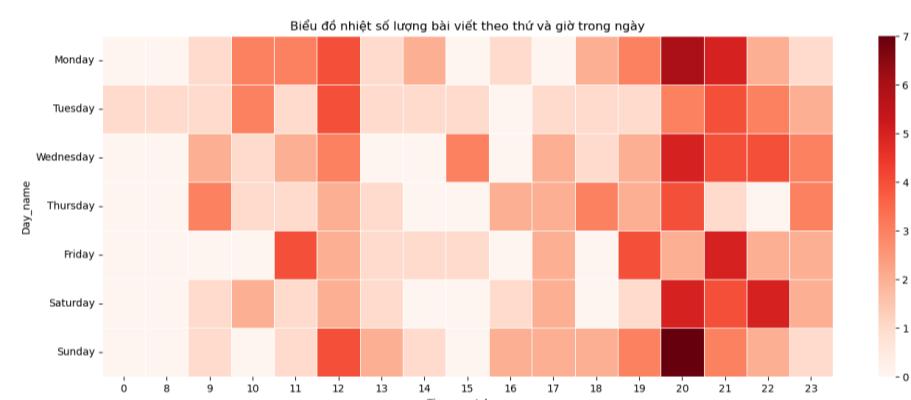
- các mốc thời gian này rải rác trong ngày và tập trung vào các khung giờ chủ yếu từ 9 giờ sáng đến 12 giờ trưa và 19 giờ tối đến 23 giờ tối.
- Đây là các khung giờ phổ biến mà con người ta sinh hoạt nên tin tức của page dễ dàng được chúng ta tiếp cận hơn.



### Sự tương quan giữa thời gian đăng bài và các ngày trong tuần?

NHẬN XÉT:

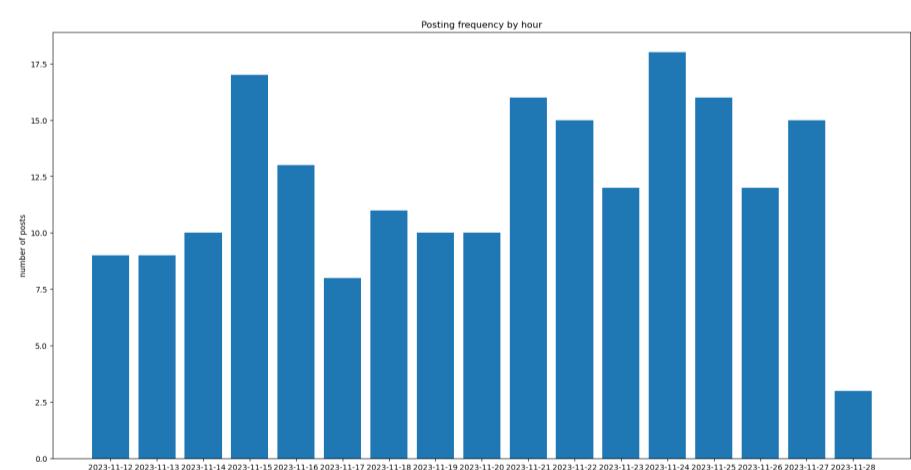
- khi nhìn tổng quan về số lượng và tần suất đăng bài trong tuần thì những khoảng thời gian được đăng nhiều chủ yếu rơi vào 20 giờ và 21 giờ.



### Sự thay đổi số lượng bài post thay đổi qua các ngày ?

NHẬN XÉT:

- số lượng bài đăng hầu hết đều nhiều hơn 7 bài trên ngày.
- một số ngày đăng ít hơn có thể là do chủ page có việc bận đột xuất không sắp xếp được công việc.



### đâu là các từ khóa xuất hiện nhiều nhất trong bài đăng được thu thập (post\_text) ?

- Vì trong top các chữ phổ biến nhất page không làm về một đối tượng hay một nhóm đối tượng cụ thể nào mà là trải đều rất nhiều chủ đề.
  - page như cái tên là đài phát thanh và nội dung page thì trong top 300 từ ta chú ý thì có thể thấy rõ các từ liên quan đến âm nhạc .

## KẾT LUẬN:

- page không đi lệch hướng và chủ đề là 'Chủ đề : Âm nhạc, Nghệ thuật và Giải trí.'



**những cái tên tham gia tương tác bài viết chính?**

Thực hiện truy cập reactor và tạo data frame lưu trữ tổng tất cả các thông tin những người là reactors.

```

# tạo datafram.
reactor_df_full = pd.DataFrame([{'name':'','link':'','type':''}])
for i in df['post_id']:
    reactor = df[df['post_id']== i]
    # chuyển thành list và xử lý dấu không cần thiết .
    reactor_list = reactor['reactors'].to_list()
    data_cleaned = [eval(d) for d in reactor_list]
    # chuyển thành list 1 chiều.
    data = sum(data_cleaned, [])
    # tạo dataframe lưu thông tin tất cả các người react trong bài post.
    temp = pd.DataFrame(data)
    reactor_df_full = pd.concat([reactor_df_full,temp],axis=0,ignore_index=True)
reactor_df_full

```

	name	link	type
0			
1	Lê Ngọc Phát	<a href="https://facebook.com/profile.php?id=1000953685...">https://facebook.com/profile.php?id=1000953685...</a>	like
2	진진	<a href="https://facebook.com/profile.php?id=1000952785...">https://facebook.com/profile.php?id=1000952785...</a>	like
3	Trong-Huy Nguyen	<a href="https://facebook.com/profile.php?id=1000951571...">https://facebook.com/profile.php?id=1000951571...</a>	love
4	Nguyễn Vănn	<a href="https://facebook.com/profile.php?id=1000949636...">https://facebook.com/profile.php?id=1000949636...</a>	like
...	...	...	...
17814	Dương Nguyễn	<a href="https://facebook.com/profile.php?id=1000901116...">https://facebook.com/profile.php?id=1000901116...</a>	like
17815	Nam Hoàng	<a href="https://facebook.com/profile.php?id=1000899741...">https://facebook.com/profile.php?id=1000899741...</a>	love
17816	Phan Quốc Việt	<a href="https://facebook.com/profile.php?id=1000913207...">https://facebook.com/profile.php?id=1000913207...</a>	love
17817	Hoang Ky Anh	<a href="https://facebook.com/profile.php?id=1000906789...">https://facebook.com/profile.php?id=1000906789...</a>	like
17818	Thúy Black	<a href="https://facebook.com/profile.php?id=1000907047...">https://facebook.com/profile.php?id=1000907047...</a>	like

Tổng hợp toàn bộ tên tương tác trong toàn bộ page kể cả bình luận và react bình luận?

Tổng số những người thực hiện tương tác với page.

danh sách những người tương tác với page

```
# tổng hợp text của 3 list.  
text = ""  
for i in name_list1:  
    text = text + i + "\n"  
for i in name_list2:  
    text = text + i + "\n"  
for i in name_list3:  
    text = text + i + "\n"  
len(text)
```

```

print(text)
✓ 0.0s

Lê Ngọc Phát
진진
Trong-Huy Nguyen
Nguyễn Vănn
Nguyễn Trần Hà Linh
Junie Tran
Lê Hiếu
Nguyễn Khánh Huyền
Thanh Thuy Le
Vũ Hoàng
Nguyễn Khánh Huyền
Minh Anh
Cam Tu
Đặng Tiến Thanh
Nguyễn Như
Phạm Ngọc Minh Quân
Lê Văn Hưởng
Ryan
Diệu Linh
Hạnh Nguyễn
Đạo Nguyễn
Minhh Hanhh
Emma Emma
Quynh Anh
...
Như Thảo
Liên Tạ
Nguyễn Quang Thành

```

### những cái tên ảnh hưởng nhiều nhất đến dữ liệu?

Ta nhận thấy trong tất cả các cột có tồn tại 1 key word là 'name'. Ta thực hiện tập hợp tất cả các 'name' xuất hiện trong dữ liệu và thực hiện phân tích tìm ra 10 keyword ảnh hưởng nhiều nhất đến dữ liệu.

Những từ xuất hiện trong bảng có thể là họ, tên hoặc tên đệm của name được phân tích. Theo kích cỡ của các chữ và ý nghĩa thì ta đưa ra bảng kết luận sau.

Ta đối chiếu với kết quả trên google.

About 125,000,000 results (0.48 seconds)

Các họ phổ biến của người Việt

Thứ tự	Họ	Tỉ lệ dân số
1	Nguyễn	31.5%
2	Trần	10.9%
3	Lê	8.9%
4	Phạm	5.9%

thì kết quả cho thấy những họ phổ biến nhất sẽ xuất hiện trong bảng bên trái .

Vậy những chữ cái còn lại thì sẽ là những cái tên tương tác với page này nhiều nhất

#### KẾT LUẬN:

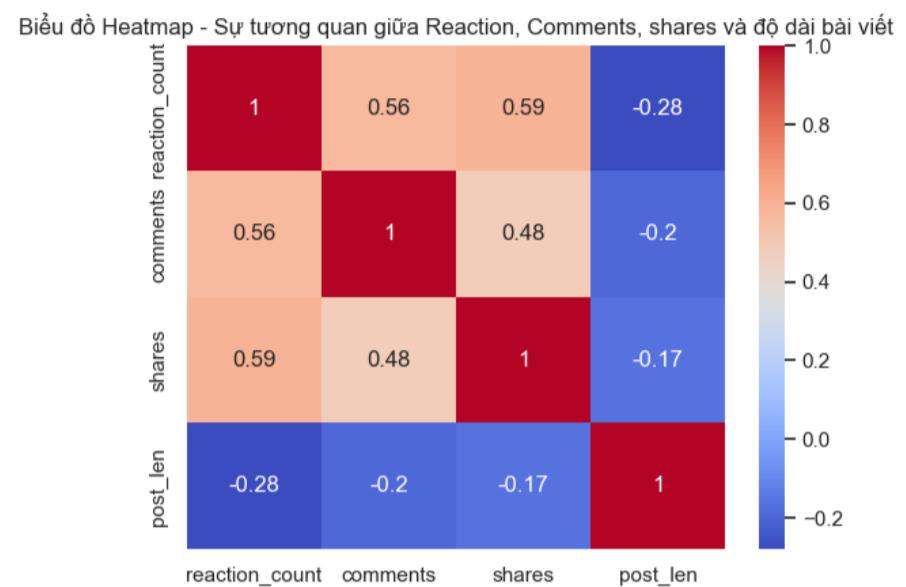
- Tuy cơ cấu dòng họ và tên tuổi của nước ta có ảnh hưởng đến page nhưng ta có thể đánh giá được là những người họ “NGUYỄN, TRẦN, LÊ, PHẠM” sẽ là những họ chính trong thành phần tương tác nhiều nhất với page. Những người họ này thường có xu hướng cởi mở nhiều trên mạng xã hội.
- Những cái tên “Hoàng, Thanh, Anh, Minh, Linh” là những cái tên thường xuất hiện trong page thì có sự quan tâm đặc biệt đến nội dung page đăng tải.

#### Sự tương quan giữa Reaction, Comments, shares và độ dài bài viết?

- dựa vào biểu đồ cho thấy độ tương quan giữa độ dài bài viết với reaction\_count, comments shares là độ dài bài viết càng ngắn thì lượng tương tác càng tăng lên.
- Reaction, Comments, shares tỉ lệ thuận với nhau và cùng tỉ lệ nghịch với độ dài bài viết.

#### KẾT LUẬN:

- người dùng có xu hướng ưa thích các bài viết ngắn gọn, xúc tích hơn.
- người dùng thường thực hiện Reaction, Comments, shares kèm theo nhau như là một thói quen thể hiện sự quan tâm của họ.



# Thank you