

# CRAWL DỮ LIỆU TỪ FACEBOOK

<b>Phần A : các công cụ được sử dụng .....</b>	<b>3</b>
<b>Phần 1: vấn đề .....</b>	<b>3</b>
<b>Phần 2: công cụ .....</b>	<b>4</b>
1.Crawl .....	4
2.Preprocessing .....	4
3.Analysis .....	6
<b>Phần B : thực thi .....</b>	<b>7</b>
<b>Phần 1 : Crawl dữ liệu .....</b>	<b>7</b>
1.Page dùng để crawl .....	7
2.Phương pháp .....	8
3.Dữ liệu crawl được .....	8
<b>Phần 2 : Phân tích , trực quan hóa dữ liệu .....</b>	<b>10</b>
1.Tiền xử lý dữ liệu .....	10
1.1,Xóa bỏ những cột bị miss dữ liệu .....	10
1.2,Định dạng dữ liệu .....	12
2.Thống kê và trực quan hóa .....	14
2.1, Số lần tương tác, cái nhìn khái quát .....	14
2.1.1, Thời gian thu thập .....	14
2.1.2, Một số dữ liệu đáng chú ý .....	14
2.1.3, Cái nhìn tổng thể .....	15
2.2, Reactions hàng ngày .....	15
2.3, Bức tranh tổng quan .....	17
2.4, Max,min của likes,shares và comments .....	18
2.4.1, Reactions .....	18
2.4.2, Shares .....	20
2.4.3, Comments .....	21
2.5, Regular posting timestamps .....	22
2.6, Tương tác theo giờ .....	23
2.7,Thống kê trong tuần .....	24
2.8,Tương quan thời gian đăng bài viết và lượt tương tác .....	25
3.Phân tích từ khóa trong bài viết .....	26

3.1, Số lượng xuất hiện của một vài từ khóa bất kỳ .....	26
3.2, Tần suất các từ được sử dụng nhiều nhất .....	27
3.3,Tìm mức độ quan trọng của các từ trong bài post .....	27
<b>Phần 3 : Phân tích, đánh giá comments .....</b>	<b>28</b>
1.Tổng số comments .....	28
2.Lọc comments .....	29
3.Wordcloud comments .....	30
4,Phân tích độ tích cực, tiêu cực của các comments .....	31
5.Phân tích người bình luận trong các bài Post .....	31
5.1, Số người dùng comments .....	31
5.2, Phân tích .....	32
5.2.1, Lấy ID người dùng .....	32
5.2.2, Lấy thông tin người dùng .....	32
5.2.3, Phân tích .....	33

## Phần A : các công cụ được sử dụng

### Phần 1: vấn đề

-Việc crawl dữ liệu từ facebook rất cần thiết và được phổ biến rộng rãi.Việc crawl có thể được thực hiện với mục đích nghiên cứu, phân tích thị trường, phát triển sản phẩm, hoặc hiểu rõ hơn về sự tương tác, ý kiến của cộng đồng.Do đó em đã thực hiện việc crawl dữ liệu của 1 page trên Facebook nhằm đánh giá sự tương tác, yêu thích của mọi người đối với 1 bộ truyện của tuổi thơ đó là Doraemon.Bài làm còn nhiều thiếu sót, rất mong nhận được sự đóng góp , nhận xét từ phía thầy.

-Project bao gồm các file dữ liệu:

📁 .git	11/11/2023 10:18 AM	File folder		
WORD ~\$o cáo cuối kì NguyenTongQuan220226...	11/16/2023 4:56 PM	DOCX Document	1 KB	
WORD Báo cáo cuối kì NguyenTongQuan220226...	11/16/2023 2:40 PM	DOCX Document	3,443 KB	
Jupyter Notebook Comments.ipynb	11/15/2023 12:48 PM	Jupyter Source File	433 KB	
Jupyter Notebook Crawl.ipynb	11/15/2023 12:48 PM	Jupyter Source File	5,236 KB	
Excel DoraemonVietnam.xlsx	11/15/2023 10:59 AM	XLSX Worksheet	1,114 KB	
Jupyter Notebook Preprocessing, Statistics and Visualization....	11/15/2023 11:14 PM	Jupyter Source File	943 KB	
INP stop_word.inp	11/8/2023 11:45 PM	INP File	23 KB	
INP stop_word2.inp	11/8/2023 11:48 PM	INP File	4 KB	
Text Document www.facebook.com_cookies.txt	11/4/2023 8:48 AM	Text Document	1 KB	

## Phần 2: công cụ

### 1.Crawl

-Sử dụng scraper cho phép cào dữ liệu từ Facebook:

Install library

```
!pip install facebook_scraper
```

Requirement already satisfied: facebook\_scraper in c:\users\fpt\appdata\local\programs\python\python310\lib\site-packages (0.2.59)  
Requirement already satisfied: dateparser<2.0.0,>=1.0.0 in c:\users\fpt\appdata\local\programs\python\python310\lib\site-packages (from facebook\_scraper) (1.1.8)  
Requirement already satisfied: demjson3<4.0.0,>=3.0.5 in c:\users\fpt\appdata\local\programs\python\python310\lib\site-packages (from facebook\_scraper) (3.0.6)  
Requirement already satisfied: requests-html<0.11.0,>=0.10.0 in c:\users\fpt\appdata\local\programs\python\python310\lib\site-packages (from facebook\_scraper) (0.11.0)  
Requirement already satisfied: python-dateutil in c:\users\fpt\appdata\roaming\python\python310\site-packages (from dateparser<2.0.0,>=1.0.0->facebook\_scraper) (2.8.1)  
Requirement already satisfied: pytz in c:\users\fpt\appdata\local\programs\python\python310\lib\site-packages (from dateparser<2.0.0,>=1.0.0->facebook\_scraper) (2021.3)  
Requirement already satisfied: regex!=2019.02.19,!~2021.8.27 in c:\users\fpt\appdata\local\programs\python\python310\lib\site-packages (from dateparser<2.0.0,>=1.0.0->facebook\_scraper) (2021.4.3)  
Requirement already satisfied: tzlocal in c:\users\fpt\appdata\local\programs\python\python310\lib\site-packages (from dateparser<2.0.0,>=1.0.0->facebook\_scraper) (2021.5.15)  
Requirement already satisfied: requests in c:\users\fpt\appdata\local\programs\python\python310\lib\site-packages (from requests-html<0.11.0,>=0.10.0->facebook\_scraper) (0.11.0)  
Requirement already satisfied: pyquery in c:\users\fpt\appdata\local\programs\python\python310\lib\site-packages (from requests-html<0.11.0,>=0.10.0->facebook\_scraper) (0.3.1)  
Requirement already satisfied: fake-useragent in c:\users\fpt\appdata\local\programs\python\python310\lib\site-packages (from requests-html<0.11.0,>=0.10.0->facebook\_scraper) (0.1.11)  
Requirement already satisfied: parse in c:\users\fpt\appdata\local\programs\python\python310\lib\site-packages (from requests-html<0.11.0,>=0.10.0->facebook\_scraper) (1.2.9)  
Requirement already satisfied: bs4 in c:\users\fpt\appdata\local\programs\python\python310\lib\site-packages (from requests-html<0.11.0,>=0.10.0->facebook\_scraper) (4.9.3)  
Requirement already satisfied: w3lib in c:\users\fpt\appdata\local\programs\python\python310\lib\site-packages (from requests-html<0.11.0,>=0.10.0->facebook\_scraper) (2.4.0)  
Requirement already satisfied: pypeteer>=0.0.14 in c:\users\fpt\appdata\local\programs\python\python310\lib\site-packages (from requests-html<0.11.0,>=0.10.0->facebook\_scraper) (0.0.14)

-Import các thư viện cần thiết :

```
from facebook_scraper import get_posts  
import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt
```

c:\Users\FPT\AppData\Local\Programs\Python\Python310\lib\site-packages\tqdm\auto.py:21: TqdmWarning: IPProgress not found. Please update jupyter and ipywidgets. See https://github.com/tqdm/tqdm#how-to-upgrade  
from .autonotebook import tqdm as notebook\_tqdm

## 2.Preprocessing

-Sử dụng các thư viện để hiện thị,xử lý,phân tích:

## Install library

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

Requirement already satisfied: matplotlib in c:\users\fpt\appdata\local\programs\python\python310\lib\site-packages (3.7.2)  
Requirement already satisfied: pandas in c:\users\fpt\appdata\local\programs\python\python310\lib\site-packages (2.1.0)  
Requirement already satisfied: numpy in c:\users\fpt\appdata\local\programs\python\python310\lib\site-packages (1.25.2)  
Requirement already satisfied: seaborn in c:\users\fpt\appdata\local\programs\python\python310\lib\site-packages (0.13.0)  
Requirement already satisfied: wordcloud in c:\users\fpt\appdata\local\programs\python\python310\lib\site-packages (1.9.2)  
Requirement already satisfied: contourpy>=1.0.1 in c:\users\fpt\appdata\local\programs\python\python310\lib\site-packages (from matplotlib) (1.1.0)  
Requirement already satisfied: cycler>=0.10 in c:\users\fpt\appdata\local\programs\python\python310\lib\site-packages (from matplotlib) (0.11.0)  
Requirement already satisfied: fonttools>=4.22.0 in c:\users\fpt\appdata\local\programs\python\python310\lib\site-packages (from matplotlib) (4.42.1)  
Requirement already satisfied: kiwisolver>1.0.1 in c:\users\fpt\appdata\local\programs\python\python310\lib\site-packages (from matplotlib) (1.4.5)  
Requirement already satisfied: packaging>=20.0 in c:\users\fpt\appdata\roaming\python\python310\site-packages (from matplotlib) (23.1)  
Requirement already satisfied: pillow>=6.2.0 in c:\users\fpt\appdata\local\programs\python\python310\lib\site-packages (from matplotlib) (10.0.0)  
Requirement already satisfied: pyParsing<3.1,>=2.3.1 in c:\users\fpt\appdata\local\programs\python\python310\lib\site-packages (from matplotlib) (3.0.9)  
Requirement already satisfied: python-dateutil>=2.7 in c:\users\fpt\appdata\roaming\python\python310\site-packages (from matplotlib) (2.8.2)  
Requirement already satisfied: pytz>=2020.1 in c:\users\fpt\appdata\local\programs\python\python310\lib\site-packages (from pandas) (2023.3.post1)  
Requirement already satisfied: tzdata>=2022.1 in c:\users\fpt\appdata\local\programs\python\python310\lib\site-packages (from pandas) (2023.3)  
Requirement already satisfied: six>=1.5 in c:\users\fpt\appdata\local\programs\python\python310\lib\site-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)  
Note: you may need to restart the kernel to use updated packages.

[notice] A new release of pip is available: 23.2.1 -> 23.3.1  
[notice] To update, run: python.exe -m pip install --upgrade pip

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

### +Matplotlib:

Tác dụng: Matplotlib là một thư viện trực quan hóa dữ liệu, giúp tạo ra đồ thị và biểu đồ chất lượng cao.

Ứng dụng: Hiển thị và trực quan hóa dữ liệu thống kê, phân tích, và mô phỏng trong Python.

### +Pandas:

Tác dụng: Pandas là một thư viện xử lý và phân tích dữ liệu dạng bảng (DataFrame).

Ứng dụng: Đọc, xử lý, và thống kê dữ liệu. Hỗ trợ quản lý và thao tác trên dữ liệu có cấu trúc.

### +NumPy:

Tác dụng: NumPy là một thư viện toán học, cung cấp các cấu trúc dữ liệu mảng và các hàm toán học hiệu quả.

Ứng dụng: Thực hiện các phép toán số học trên mảng, xử lý dữ liệu đa chiều.

### +Seaborn:

Tác dụng: Seaborn là một thư viện trực quan hóa dữ liệu dựa trên Matplotlib, tập trung vào việc tạo ra đồ thị thống kê có thẩm mỹ cao.

Ứng dụng: Trực quan hóa mối quan hệ thống kê phức tạp, phân tích mối quan hệ giữa nhiều biến.

### +WordCloud:

Tác dụng: WordCloud là một thư viện trực quan hóa dữ liệu, tạo ra hình ảnh từ các từ xuất hiện

trong văn bản.

Ứng dụng: Hiển thị các từ quan trọng trong văn bản, đặc biệt thường được sử dụng trong phân tích ý kiến và trực quan hóa từ khóa.

-Sử dụng stopword để tìm keyword:

```
!pip install -U nltk
!pip install rake-nltk

Requirement already satisfied: nltk in c:\users\fpt\appdata\local\programs\python\python310\lib\site-packages (3.8.1)
Requirement already satisfied: click in c:\users\fpt\appdata\local\programs\python\python310\lib\site-packages (from nltk) (7.1.2)
Requirement already satisfied: joblib in c:\users\fpt\appdata\local\programs\python\python310\lib\site-packages (from nltk) (1.3.2)
Requirement already satisfied: regex>=2021.8.3 in c:\users\fpt\appdata\local\programs\python\python310\lib\site-packages (from nltk) (2023.10.3)
Requirement already satisfied: tqdm in c:\users\fpt\appdata\local\programs\python\python310\lib\site-packages (from nltk) (4.66.1)
Requirement already satisfied: colorama in c:\users\fpt\appdata\roaming\python\python310\site-packages (from tqdm->nltk) (0.4.6)

[notice] A new release of pip is available: 23.2.1 -> 23.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip
Requirement already satisfied: rake-nltk in c:\users\fpt\appdata\local\programs\python\python310\lib\site-packages (1.0.6)
Requirement already satisfied: nltk<4.0.0,>=3.6.2 in c:\users\fpt\appdata\local\programs\python\python310\lib\site-packages (from rake-nltk) (3.8.1)
Requirement already satisfied: click in c:\users\fpt\appdata\local\programs\python\python310\lib\site-packages (from nltk<4.0.0,>=3.6.2->rake-nltk) (7.1.2)
Requirement already satisfied: joblib in c:\users\fpt\appdata\local\programs\python\python310\lib\site-packages (from nltk<4.0.0,>=3.6.2->rake-nltk) (1.3.2)
Requirement already satisfied: regex>=2021.8.3 in c:\users\fpt\appdata\local\programs\python\python310\lib\site-packages (from nltk<4.0.0,>=3.6.2->rake-nltk) (2023.10.3)
Requirement already satisfied: tqdm in c:\users\fpt\appdata\local\programs\python\python310\lib\site-packages (from nltk<4.0.0,>=3.6.2->rake-nltk) (4.66.1)
Requirement already satisfied: colorama in c:\users\fpt\appdata\roaming\python\python310\site-packages (from tqdm->nltk) (0.4.6)

[notice] A new release of pip is available: 23.2.1 -> 23.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip

import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk import tokenize
import math
from operator import itemgetter
```

### 3. Analysis

-Sử dụng SentimentAnalyzer để đánh giá cảm xúc, độ tích cực của comments:

```

import nltk
from nltk.sentiment import SentimentIntensityAnalyzer

# Tải dữ liệu cho việc phân tích cảm xúc
nltk.download('vader_lexicon')

# Khởi tạo đối tượng SentimentIntensityAnalyzer
sid = SentimentIntensityAnalyzer()

# Phân tích cảm xúc
sentiment_scores = sid.polarity_scores(text)

# In kết quả
print(sentiment_scores)

# Xác định là tích cực, tiêu cực, hay trung tính dựa trên điểm số
if sentiment_scores['compound'] >= 0.05:
    print("Bình luận của các bài viết mang hướng tích cực")
elif sentiment_scores['compound'] <= -0.05:
    print("Bình luận của các bài viết mang hướng tiêu cực")
else:
    print("Bình luận của các bài viết mang hướng trung tính")

```

[nltk\_data] Downloading package vader\_lexicon to  
[nltk\_data] C:\Users\PTT\AppData\Roaming\nltk\_data...  
[nltk\_data] Package vader\_lexicon is already up-to-date!

## Phần B : thực thi

### Phần 1 : Crawl dữ liệu

#### 1. Page dùng để crawl

-Trang fanpage được chọn là DoraemonVietNam :

## 2.Phương pháp

-FANPAGE\_LINK: Liên kết đến trang fan hâm mộ mà chúng tôi muốn thu thập dữ liệu từ đó.

-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 ta sẽ lưu dữ liệu vào. Thư mục này sẽ được tạo trong cùng thư mục với sổ ghi chép này.

Crawl the data using facebook\_scraper

```
FANPAGE_LINK = "DoraemonVietnam"
FOLDER_PATH = "C:/FinalProject"
COOKIE_PATH = "C:/FinalProject/www.facebook.com_cookies.txt"

PAGES_NUMBER = 13

post_list = []
for post in get_posts(FANPAGE_LINK,
    options={"comments": True, "reactions": True, "allow_extra_requests": True},
    extra_info=True, pages=13, cookies=COOKIE_PATH):
    print(post)
    post_list.append(post)
```

## 3.Dữ liệu crawl được

-Thông tin của 1 post:

```

post_list[0]

{'post_id': '725532749613297',
 'text': 'Chú mèo đèn từ tương lai 🐱\n#Doraemon #doraemovn #doraemon',
 'post_text': 'chú mèo đèn từ tương lai 🐱\n#Doraemon #doraemovn #doraemon',
 'shared_text': '',
 'original_text': None,
 'time': datetime.datetime(2023, 10, 28, 18, 51, 7),
 'timestamp': 1698493867,
 'image': None,
 'image_lowquality': 'https://scontent.fsgn2-11.fna.fbcdn.net/v/t15.5256-10/395654417_1306735223380706_906448791072744133_n.jpg?stp=cp0_dst-jpg_e15_q65_s320x320',
 'images': [],
 'images_description': [],
 'images_lowquality': ['https://scontent.fsgn2-11.fna.fbcdn.net/v/t15.5256-10/395654417_1306735223380706_906448791072744133_n.jpg?stp=cp0_dst-jpg_e15_q65_s320x320'],
 'images_lowquality_description': [None],
 'video': 'https://scontent.fsgn2-10.fna.fbcdn.net/v/t42.1790-2/396398893_862844678504217_8115985833379005781_n.mp4?_nc_cat=109&ccb=1-7&_nc_sid=55d0d3&efg=eyJ2Z',
 'video_duration_seconds': None,
 'video_height': None,
 'video_id': '864495042015331',
 'video_quality': None,
 'video_size_MB': None,
 'video_thumbnail': 'https://scontent.fsgn2-11.fna.fbcdn.net/v/t15.5256-10/395654417_1306735223380706_906448791072744133_n.jpg?stp=cp0_dst-jpg_e15_q65_s320x320',
 'video_watches': None,
 'video_width': None,
 'likes': None,
 'comments': 78,
 'shares': 10,
 ...
 'sharers': None,
 'image_id': None,
 'image_ids': []
}

```

## -Thông tin crawl được:

post_df_full							
	post_id	text	post_text	time	timestamp	image	image_lowquality
0	733451988821373	Phim hoạt hình Đô rê mon từng được lòng tiếng ...	Phim hoạt hình Đô rê mon từng được lòng tiếng ...	2023- 11-11 08:50:06	1699667406	NaN	https://scontent.fhan2- 4.fna.fbcdn.net/v/t15.5...
1	733109305522308	BỘ BOARD GAME DORAEMON ĐẦU TIÊN SẮP XUẤT HIỆN...	BỘ BOARD GAME DORAEMON ĐẦU TIÊN SẮP XUẤT HIỆN...	2023- 11-10 20:01:56	1699621316	https://scontent.fhan2- 4.fna.fbcdn.net/v/t39.3... https://scontent.fhan2- 4.fna.fbcdn.net/v/t39.3... [https://scontent.fhan2-4.fna.fbcdn.net/v/t39.3...	[https://scontent.fhan2-4.fna.fbcdn.net/v/t39.3...
2	732388768927695	Chưa làm được tích sự gi nhưng vừa đến đã ăn l...	Chưa làm được tích sự gi nhưng vừa đến đã ăn l...	2023- 11-09 10:46:39	1699501599	https://m.facebook.com/photo/view_full_size/? f... https://scontent.fhan2- 4.fna.fbcdn.net/v/t39.3... [https://m.facebook.com/photo/view_fu...	[https://m.facebook.com/photo/view_fu...
3	731920548974517	Một phút thật lòng nào :)	Một phút thật lòng nào :)	2023- 11-08 18:46:30	1699443990	NaN	https://scontent.fhan2- 4.fna.fbcdn.net/v/t39.3...

## Phần 2 : Phân tích , trực quan hóa dữ liệu

### 1.Tiền xử lý dữ liệu

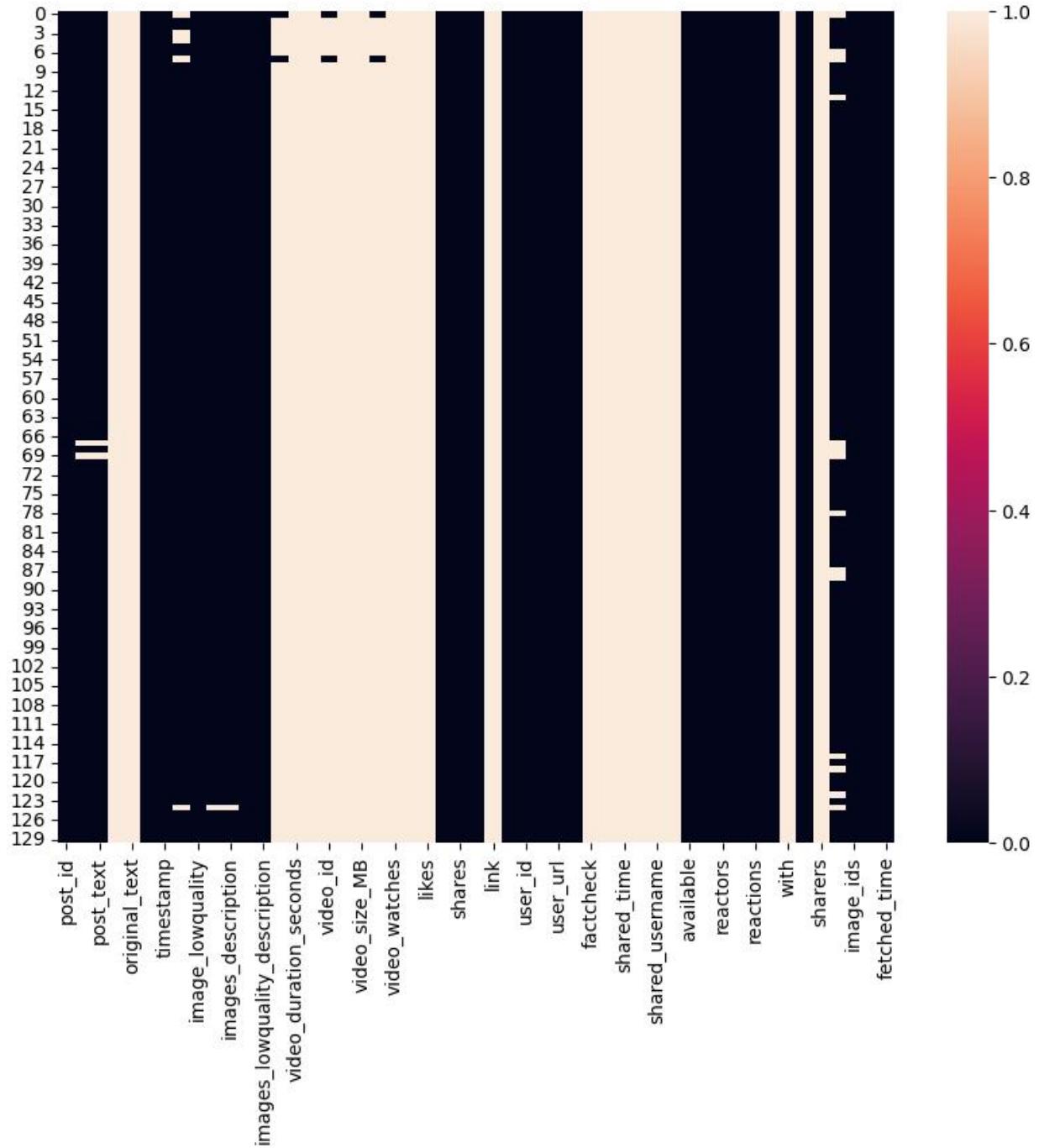
#### 1.1,Xóa bỏ những cột bị miss dữ liệu

-Xác định dữ liệu bị mất:

```
for col in post_df_full.columns :
    miss_data = post_df_full[col].isna().sum()
    miss_per = miss_data / len(post_df_full)
    miss_per = round(miss_per * 100,2)
    print(f"{col} có : {miss_per}% dữ liệu bị mất")
    ✓ 0.0s

post_id có : 0.0% dữ liệu bị mất
text có : 1.54% dữ liệu bị mất
post_text có : 1.54% dữ liệu bị mất
shared_text có : 100.0% dữ liệu bị mất
original_text có : 100.0% dữ liệu bị mất
time có : 0.0% dữ liệu bị mất
timestamp có : 0.0% dữ liệu bị mất
image có : 3.85% dữ liệu bị mất
image_lowquality có : 0.0% dữ liệu bị mất
images có : 0.77% dữ liệu bị mất
images_description có : 0.77% dữ liệu bị mất
images_lowquality có : 0.0% dữ liệu bị mất
images_lowquality_description có : 0.0% dữ liệu bị mất
video có : 98.46% dữ liệu bị mất
video_duration_seconds có : 100.0% dữ liệu bị mất
video_height có : 100.0% dữ liệu bị mất
video_id có : 98.46% dữ liệu bị mất
video_quality có : 100.0% dữ liệu bị mất
video_size_MB có : 100.0% dữ liệu bị mất
video_thumbnail có : 98.46% dữ liệu bị mất
video_watches có : 100.0% dữ liệu bị mất
video_width có : 100.0% dữ liệu bị mất
likes có : 100.0% dữ liệu bị mất
comments có : 0.0% dữ liệu bị mất
shares có : 0.0% dữ liệu bị mất
```

-Hiển thị dữ liệu bị mất dưới dạng biểu đồ:



-Xóa những dữ liệu không đủ để sử dụng:

```

for col in post_df_full.columns:
    miss_data = post_df_full[col].isna().sum()
    miss_per = miss_data / len(post_df_full)
    miss_per = round(miss_per * 100, 2)

    if miss_per > 90:
        post_df_full.drop(col, axis=1, inplace=True)
        print(f"{col} đã bị xóa vì có {miss_per}% dữ liệu bị mất.")

✓ 0.0s

shared_text đã bị xóa vì có 100.0% dữ liệu bị mất.
original_text đã bị xóa vì có 100.0% dữ liệu bị mất.
video đã bị xóa vì có 98.46% dữ liệu bị mất.
video_duration_seconds đã bị xóa vì có 100.0% dữ liệu bị mất.
video_height đã bị xóa vì có 100.0% dữ liệu bị mất.
video_id đã bị xóa vì có 98.46% dữ liệu bị mất.
video_quality đã bị xóa vì có 100.0% dữ liệu bị mất.
video_size_MB đã bị xóa vì có 100.0% dữ liệu bị mất.
video_thumbnail đã bị xóa vì có 98.46% dữ liệu bị mất.
video_watches đã bị xóa vì có 100.0% dữ liệu bị mất.
video_width đã bị xóa vì có 100.0% dữ liệu bị mất.
likes đã bị xóa vì có 100.0% dữ liệu bị mất.
link đã bị xóa vì có 100.0% dữ liệu bị mất.
factcheck đã bị xóa vì có 100.0% dữ liệu bị mất.
shared_post_id đã bị xóa vì có 100.0% dữ liệu bị mất.
shared_time đã bị xóa vì có 100.0% dữ liệu bị mất.
shared_user_id đã bị xóa vì có 100.0% dữ liệu bị mất.
shared_username đã bị xóa vì có 100.0% dữ liệu bị mất.
shared_post_url đã bị xóa vì có 100.0% dữ liệu bị mất.
with đã bị xóa vì có 100.0% dữ liệu bị mất.
sharers đã bị xóa vì có 100.0% dữ liệu bị mất.

```

## 1.2, Định dạng dữ liệu

-Dữ liệu ngày đang ở dạng chuỗi cần phải chuyển sang dạng datetime :

```
post_df_full['time'] = pd.to_datetime(post_df_full['time'])
```

✓ 0.0s

-Tách cột reactions thành những cột cảm xúc riêng biệt để dễ dàng thao tác:

Categorize each emotion

```
# chuyển kiểu
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 )
```

```
C:\Users\FPT\AppData\Local\Temp\ipykernel_14564\1935438685.py:3: FutureWarning: Returning a DataFrame from Series.apply when the supplied function returns a DataFrame is deprecated, and will return a Series by default starting in a future version of pandas.
  post_df_full_reactions = post_df_full1['reactions'].apply(pd.Series )
```



## 2.Thống kê và trực quan hóa

### 2.1, Số lần tương tác, cái nhìn khái quát

#### 2.1.1, Thời gian thu thập

```
start_date = post_df_full["time"].min().strftime('%Y-%m-%d')
end_date = post_df_full["time"].max().strftime('%Y-%m-%d')

print(f"Mốc thời gian thu thập từ {start_date} đến {end_date}")
```

```
Mốc thời gian thu thập từ 2021-10-15 đến 2023-11-11
```

-Do Page đăng bài với số lượng bài viết không quá lớn, nên các số liệu thu thập được đến tận năm 2021

#### 2.1.2, Một số dữ liệu đáng chú ý

-Về lượt reactions, số lượng bày tỏ cảm xúc gần 900k lượt, một con số mơ ước với mỗi FanPage:

```
print("Tổng reaction :",post_df_full["reaction_count"].sum())
```

```
Tổng reaction : 878084
```

```
average_reaction = round(post_df_full["reaction_count"].mean(), 2)
print("Số reaction trung bình:", average_reaction)
```

```
Số reaction trung bình: 6754.49
```

-Về lượt shares cũng khá cao gần 13k lượt:

```
print("Tổng shares :",post_df_full["shares"].sum())
✓ 0.0s
```

```
Tổng shares : 12702
```

```
average_shares = round(post_df_full["shares"].mean(), 2)
print("Số shares trung bình mỗi bài post:", average_shares)
```

```
✓ 0.0s
```

```
Số shares trung bình mỗi bài post: 97.71
```

-Về lượt comments:

```

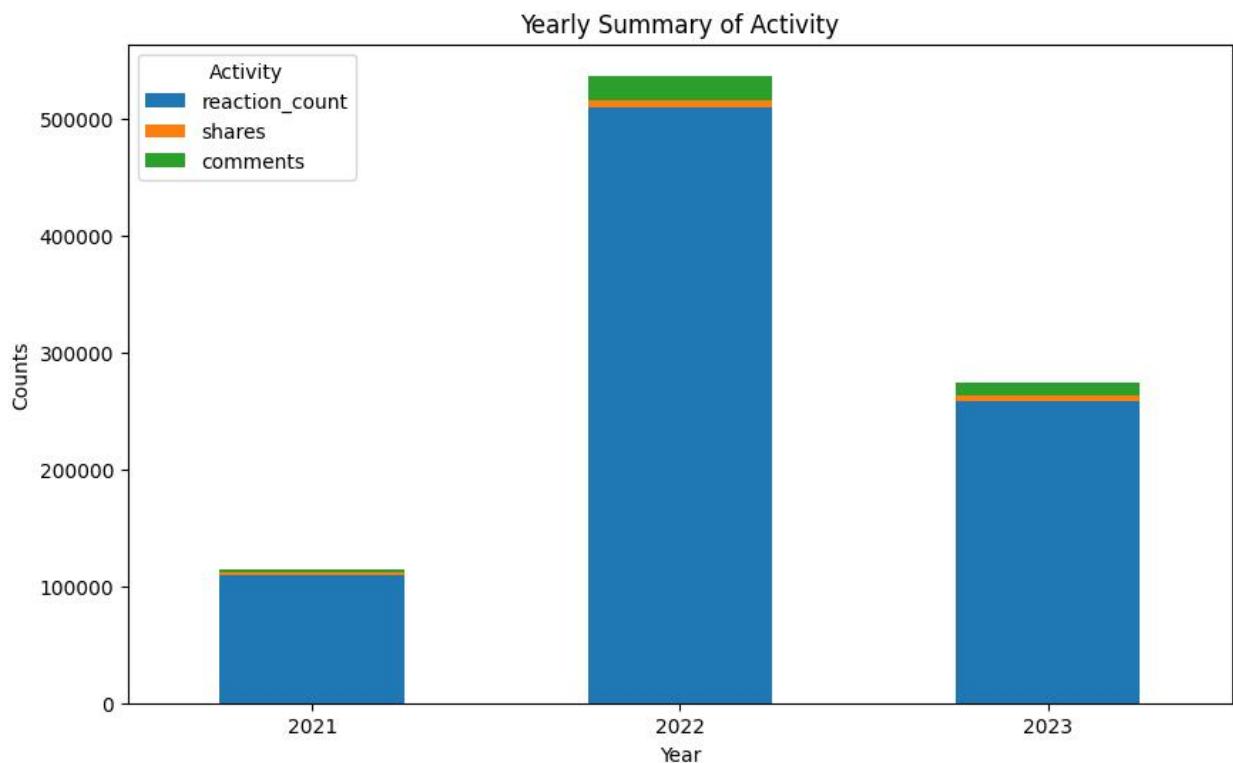
print("Tổng số Comments : ", post_df_full["comments"].sum())
✓ 0.0s
Tổng số Comments : 34101

average_cmt = round(post_df_full["comments"].mean(), 2)
print("Số comments trung bình mỗi bài post:", average_cmt)
✓ 0.0s
Số comments trung bình mỗi bài post: 262.32

```

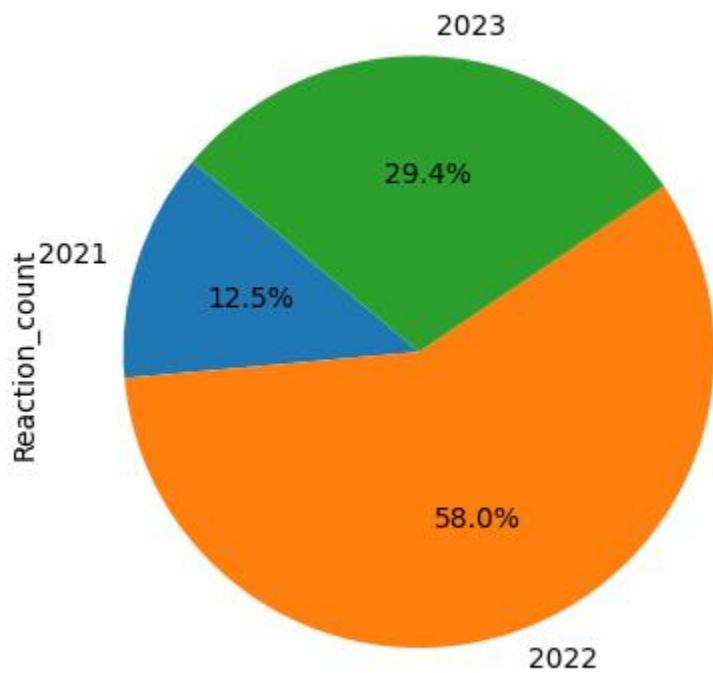
### 2.1.3, Cái nhìn tổng thể

-Do sự chênh lệch về thời gian thu thập nên biểu đồ của năm 2022 vượt trội hơn cả:

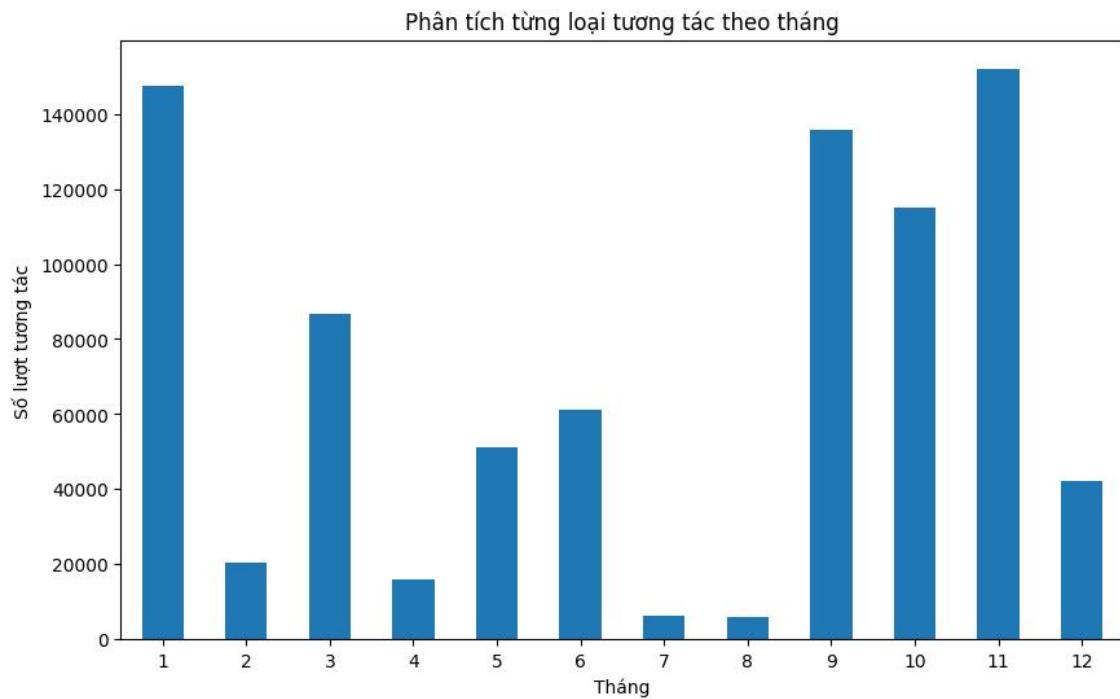


### 2.2, Reactions hàng ngày

-Số lượng tương tác được tính theo năm :

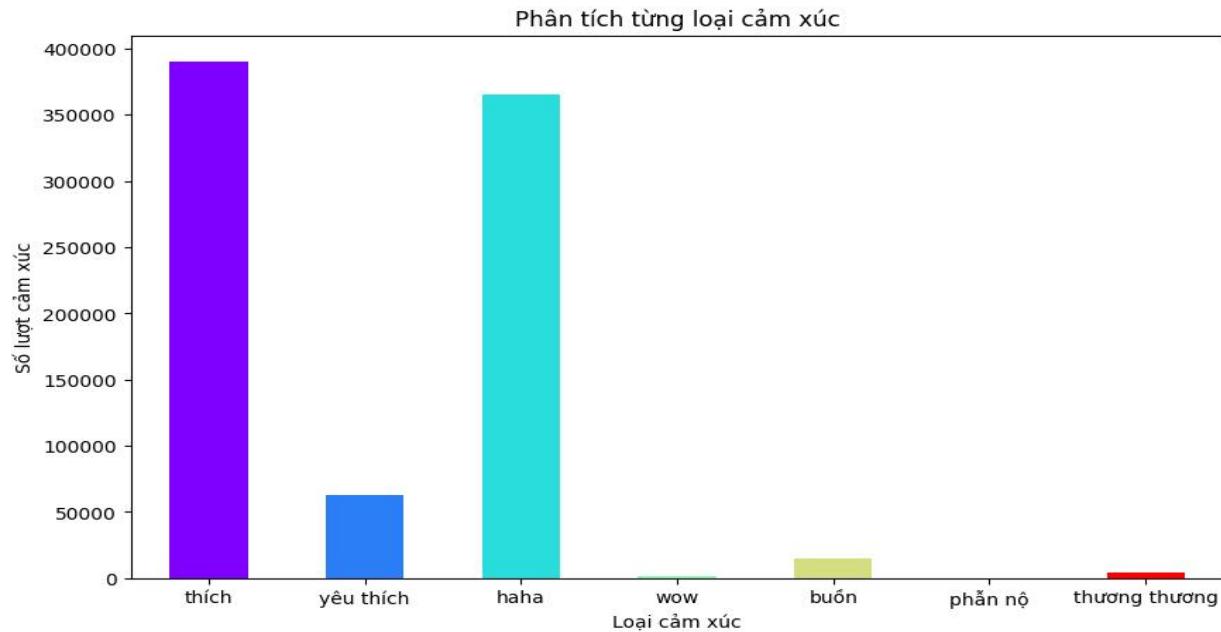


-Số lượng tương tác được tính theo tháng,lượt tương tác vào những tháng mùa Thu là nổi trội hơn cả:



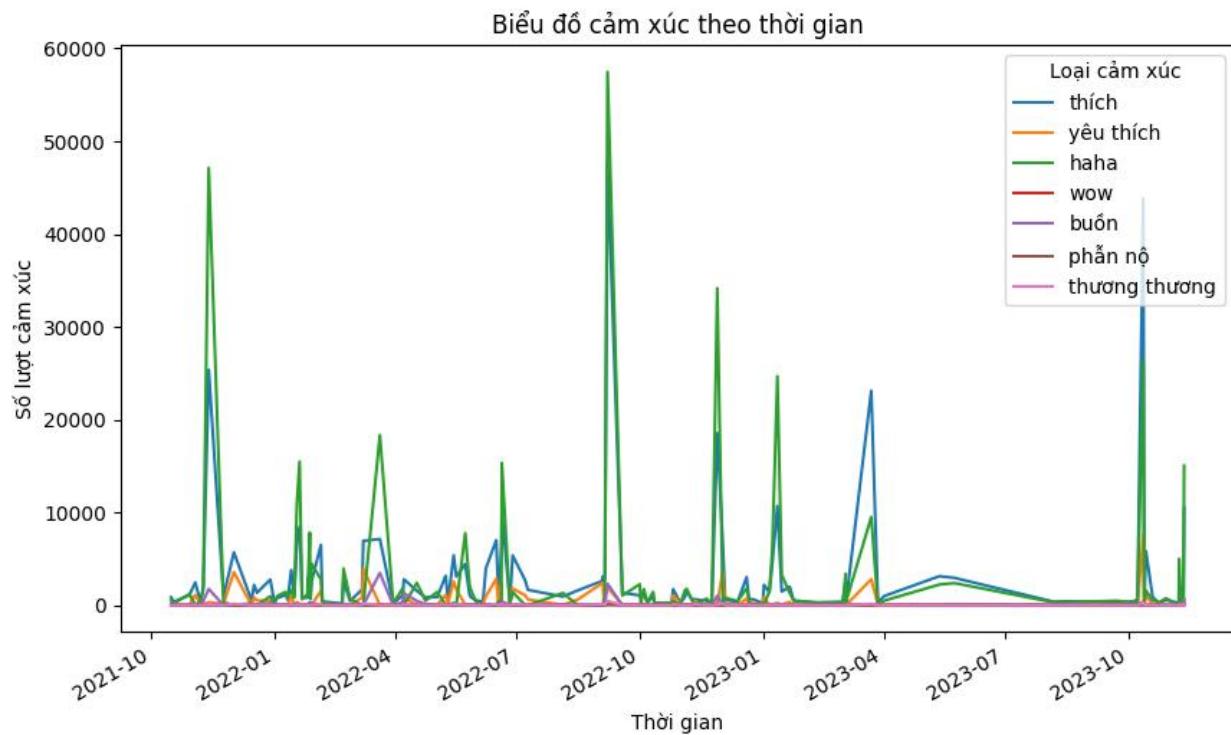
### 2.3, *Bức tranh tổng quan*

-Biểu đồ thống kê cảm xúc, những lượt cảm xúc “Wow” hay là “Phẫn nộ” khá là kén người dùng:



-Chúc ta thấy được nội dung của những bài đăng mang thiên hướng vui vẻ nên nhận được số lượng “Haha” rất lớn

-Biểu đồ cảm xúc theo thời gian:



## 2.4, Max,min của likes,shares và comments

### 2.4.1, Reactions

-Chênh lệch rất lớn giữa số bài viết có tương tác nhiều nhất, và bài viết nhận được ít sự quan tâm nhất:

```
print("Số lượt tương tác nhiều nhất : ",post_df_full["reaction_count"].max())
```

```
Số lượt tương tác nhiều nhất : 110201
```

+Thông tin về bài viết có số lượng reactions lớn nhất:

```

# Find the row with the maximum reaction_count
max_reaction_row = post_df_full.loc[post_df_full["reaction_count"].idxmax()]

max_reaction_id = max_reaction_row["post_id"]
max_reaction_text = max_reaction_row["post_text"]
max_reaction_time = max_reaction_row["time"]
reaction_max = max_reaction_row["reactions"]

# Convert the dictionary to a formatted string with line breaks
formatted_reactions = "\n".join([f"{reaction_type}: {count}" for reaction_type, count in reaction_max.items()])

print(f"Post Id: {max_reaction_id}")
print(f"Post Text: {max_reaction_text}")
print(f"Time: {max_reaction_time}")
print(f"Reactions: \n{formatted_reactions}")

```

✓ 0.0s

Post Id: 2708273079303120  
Post Text: Ghét của nào trót trao của đây  
Time: 2022-09-06 18:30:30  
Reactions:  
thích: 47712  
yêu thích: 1907  
haha: 57465  
wow: 299  
thương thương: 454  
buồn: 2313  
phản nô: 34

+Bài viết có số lượt reactions ít nhất:

```
print("Số lượt tương tác ít nhất : ", post_df_full["reaction_count"].min())
```

Số lượt tương tác ít nhất : 39

```

max_reaction_row = post_df_full.loc[post_df_full["reaction_count"].idxmin()]

max_reaction_id = max_reaction_row["post_id"]
max_reaction_text = max_reaction_row["post_text"]
max_reaction_time = max_reaction_row["time"]
reaction_max = max_reaction_row["reactions"]

formatted_reactions = "\n".join([f"{reaction_type}: {count}" for reaction_type, count in reaction_max.items()])

print(f"Post Id: {max_reaction_id}")
print(f"Post Text: {max_reaction_text}")
print(f"Time: {max_reaction_time}")
print(f"Reactions: \n{formatted_reactions}")

```

Python

Post Id: 733109305522308  
Post Text: |BỘ BOARD GAME DORAEMON ĐẦU TIÊN SẮP XUẤT HIỆN!!|  
Mới nghe nói EverJoy sắp trình làng bộ game Doraemon được chính chủ bản quyền Fujio bên Nhật cấp phép, chắc chắn sẽ tạo ra một cơn sốt lớn. Có thể chúng ta đã từng nghe qua đến cái tên EverJoy. EverJoy Publishing là công ty hoạt động trong lĩnh vực giải trí tương tác, đầu tư vào các thương hiệu hàng đ. Trong năm 2023, ngoài việc phát triển các board game truyền thống, thương hiệu phát hành Board Game VN đã kết hợp với các IP nổi tiếng toàn cầu, đặc biệt có Thám t Xuất phát điểm là những người chơi và người sáng tạo đầy nhiệt huyết, EverJoy góp phần phát triển cho các studio và nhà phát hành board game Việt có thể giới thiệu cùng chờ đón các hình ảnh tiếp theo từ bộ trò chơi này trên fanpage Doraemon nhé!  
#Doraemon #Boardgame #EverJoy  
Time: 2023-11-10 20:01:56  
Reactions:  
thích: 26  
yêu thích: 12  
wow: 1

-Những bài viết khác nhận được sự quan tâm lớn:



```

max_reaction_row = post_df_full.loc[post_df_full["shares"].idxmin()]

max_reaction_id = max_reaction_row["post_id"]
max_reaction_text = max_reaction_row["post_text"]
max_reaction_time = max_reaction_row["time"]
reaction_shares = max_reaction_row["shares"]

print(f"Post Id: {max_reaction_id}")
print(f"Post Text: {max_reaction_text}")
print(f"Time: {max_reaction_time}")
print(f"Shares: {reaction_shares}")

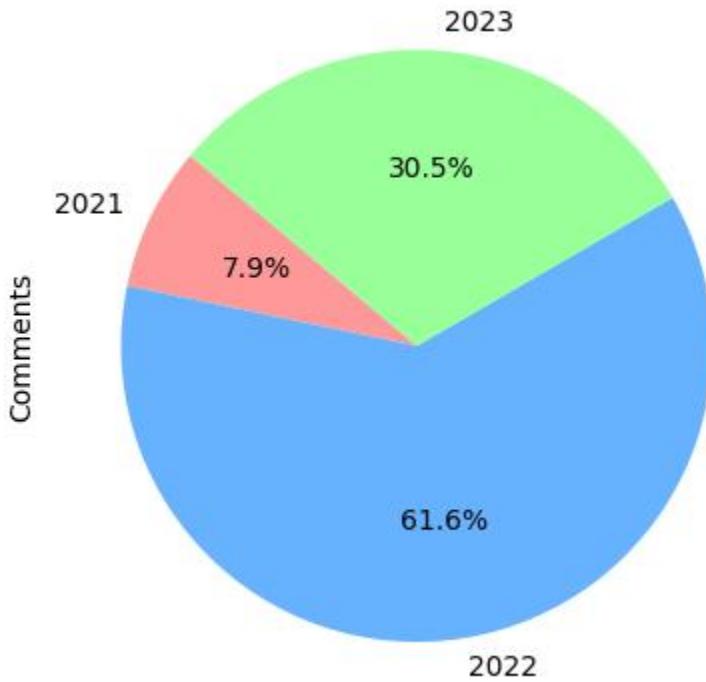
✓ 0.0s

Post Id: 681378290695410
Post Text: Cao thủ gấp thách đấu :)))
Cre: Gia đình ngộ
Time: 2023-08-15 14:02:41
Shares: 1

```

#### 2.4.3, Comments

-Do thời gian thu thập chênh lệch nhau, nên số comments cũng chênh lệch khá lớn, biểu đồ chỉ là sự thống kê của số comments đã được thu thập:

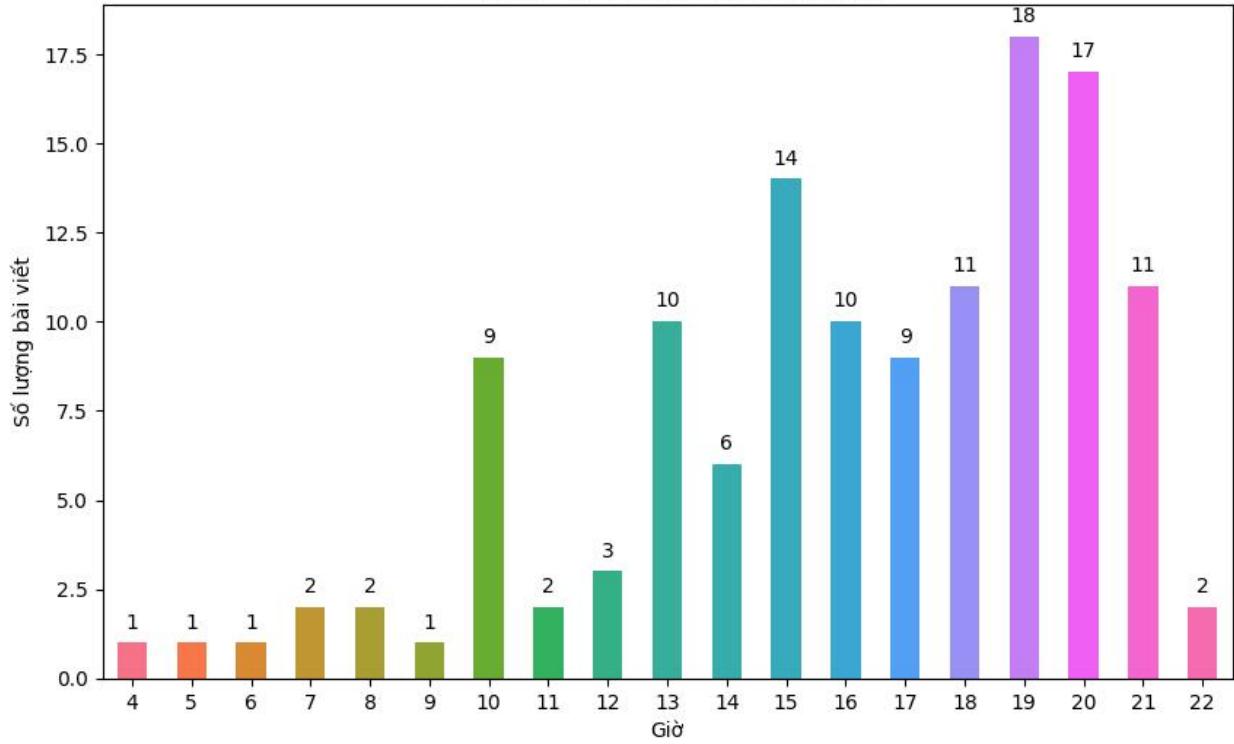


-Những bài viết nhận được nhiều lượt reactions , thì cũng nhận được nhiều phản hồi bằng comments cũng như là lượt chia sẻ rất cao:

post_id	post_text	time	reactions	reaction_count	shares	comments
0 733451988821373	Phim hoạt hình Đô rê mon từng được lòng tiếng ...	2023-11-11 08:50:06	{'thich': 10524, 'yêu thích': 607, 'haha': 150...}	27027	2397	3933
14 714647757368463	Có hẳn phòng riêng cho chuột cớ :))	2023-10-11 20:59:04	{'thich': 43674, 'yêu thích': 7652, 'haha': 26...	78610	916	2014
41 2807155129414914	Độc là Vũng Tàu :)))\n\nAnh: Giang Hoàng Đan	2022-12-19 15:05:34	{'thich': 3036, 'yêu thích': 721, 'haha': 1820...	5700	128	1153
45 2786271731503254	Nuôi ăn miễn phí rồi để nó tìm mẹ trong thùng ...	2022-11-27 17:04:17	{'thich': 18542, 'yêu thích': 228, 'haha': 341...	54060	331	1510
64 2708273079303120	Ghét cửa nào trời trao cửa đấy	2022-09-06 18:30:30	{'thich': 47712, 'yêu thích': 1907, 'haha': 57...	110184	921	8125
123 2461153630681734	Là ma nhưng xem phim ma vẫn sợ :))	2021-11-12 21:48:30	{'thich': 25386, 'yêu thích': 332, 'haha': 471...	74790	436	1304

## 2.5, Regular posting timestamps

Số lượng bài viết theo giờ trong ngày



-Những giờ nghỉ ngơi, số bài viết được đăng rất nhiều

-Những khung giờ đêm khuya và rạng sáng thì không xuất hiện những bài đăng nào

### Time frame with the most and fewest posts

```
max_hour = x_h.idxmax()
max_post_count = x_h.max()

print(f"Thời gian có nhiều bài post nhất vào khoảng: {max_hour} giờ, số lượng: {max_post_count} bài post")
```

Thời gian có nhiều bài post nhất vào khoảng: 19 giờ, số lượng: 18 bài post

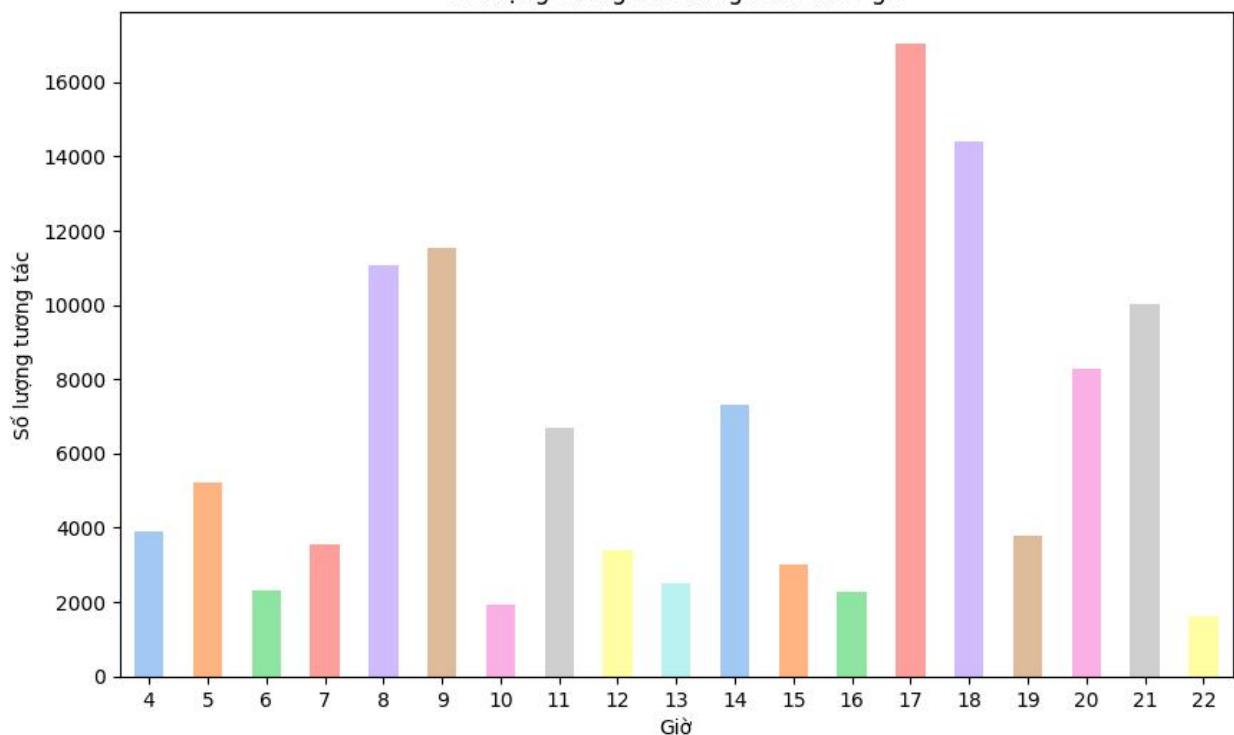
```
min_hour = x_h.idxmin()
min_post_count = x_h.min()

print(f"Thời gian có ít bài post nhất vào khoảng: {min_hour} giờ, số lượng: {min_post_count} bài post")
```

Thời gian có ít bài post nhất vào khoảng: 4 giờ, số lượng: 1 bài post

### 2.6. Tương tác theo giờ

Số lượng tương tác trung bình theo giờ



-Vào những khung giờ nghỉ ngơi lượng tương tác khá là ít, và đổ dồn vào những khung giờ còn lại trong ngày

```

max_hour = x_a.idxmax()
max_post_count = x_a.max()

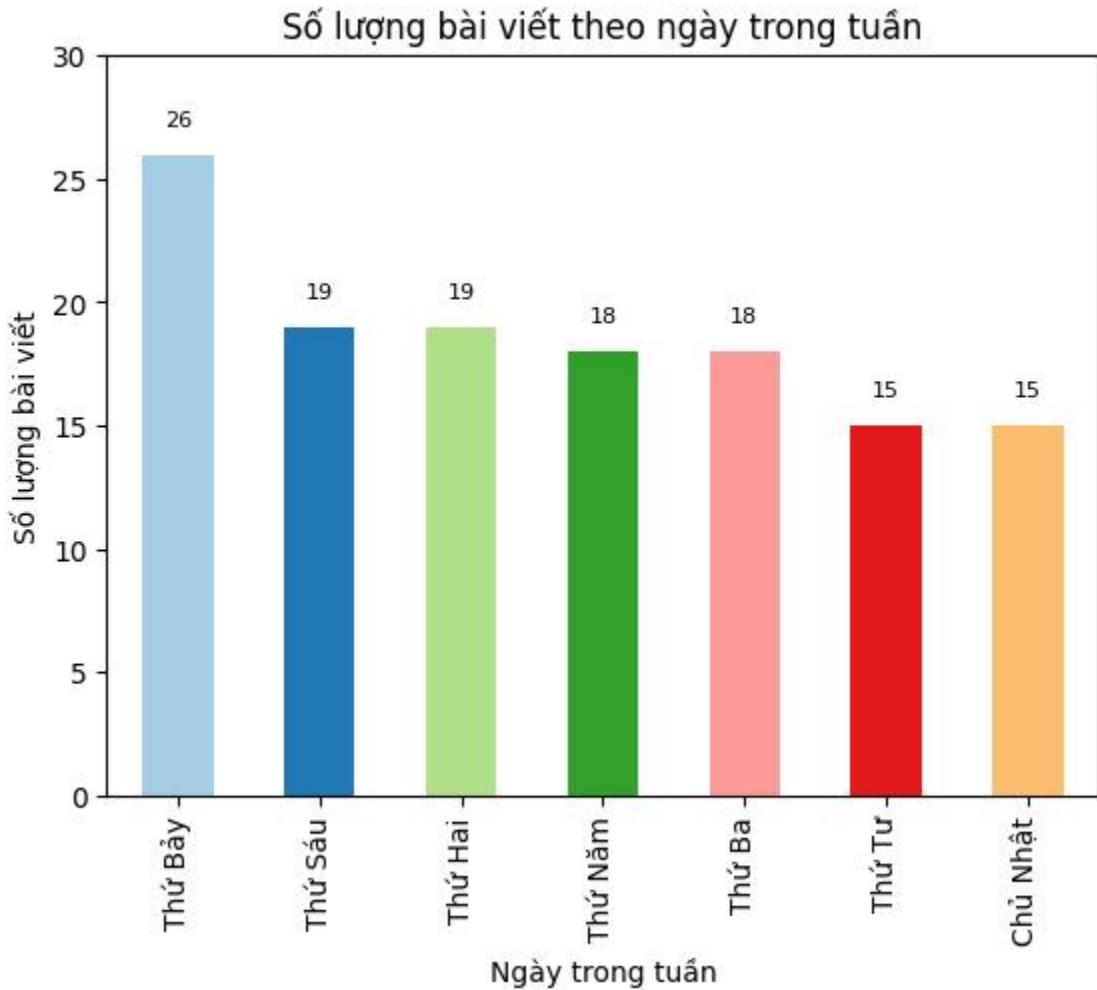
print(f'Lượt tương tác trung bình của bài viết cao nhất vào lúc:{max_hour} giờ,số lượng: {max_post_count} tương tác trên 1 bài post')
✓ 0.0s
Lượt tương tác trung bình của bài viết cao nhất vào lúc:8 giờ,số lượng: 19053 tương tác trên 1 bài post

max_hour = x_a.idxmin()
max_post_count = x_a.min()

print(f'Lượt tương tác trung bình của bài viết thấp nhất vào lúc:{max_hour} giờ,số lượng: {max_post_count} tương tác trên 1 bài post')
✓ 0.0s
Lượt tương tác trung bình của bài viết thấp nhất vào lúc:22 giờ,số lượng: 1609 tương tác trên 1 bài post

```

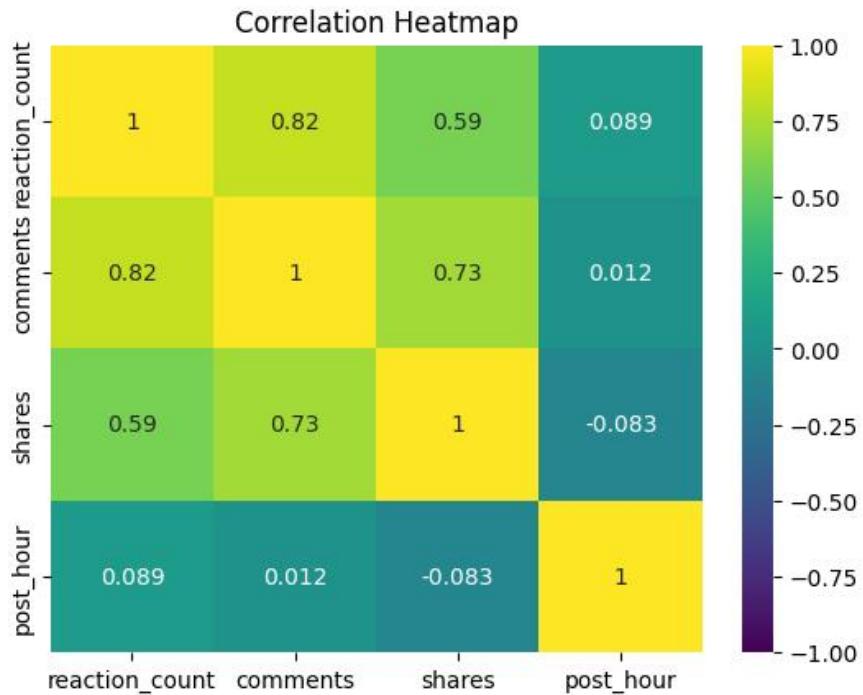
## 2.7, Thống kê trong tuần



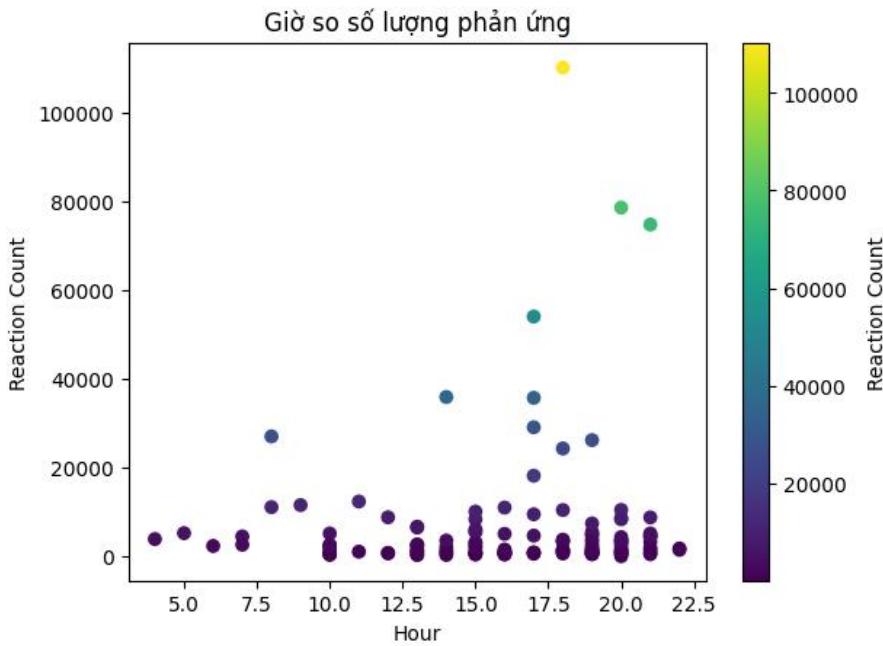
-Không quá ngạc nhiên khi thứ 7 chính là ngày mà số bài viết được đăng lên nhiều nhất, thứ 7 là thứ mà hầu hết mọi người đều nghỉ ngơi thư giãn sau 1 tuần làm việc mệt mỏi

## 2.8, Tương quan thời gian đăng và viết và lượt tương tác

-Bản đồ nhiệt về sự tương quan giữa các trường dữ liệu:



-Thời gian đăng bài so với số lượng reactions



### 3.Phân tích từ khóa trong bài viết

#### 3.1, Số lượng xuất hiện của một vài từ khóa bất kỳ

-Bộ truyện xoay quanh cuộc sống của nhân vật Nobita và bạn thân là Doraemon nên chúng ta xem số lần xuất hiện của hai từ khóa này trong các bài đăng của Fanpage:

```
a = post_df_full_with_reactions['post_text'].tolist()
print("Tổng số bài viết:", len(a))

res = []
for i in a:
    if isinstance(i, str) and "Doraemon" in i:
        res.append(i)

print(len(res), "bài viết chứa từ khóa 'Doraemon'")

✓ 0.0s

Tổng số bài viết: 130
19 bài viết chứa từ khóa 'Doraemon'
```

#### Number of articles containing the keyword "Nobita"

```
a = post_df_full_with_reactions['post_text'].tolist()
print("Tổng số bài viết:", len(a))

res = []
for i in a:
    if isinstance(i, str) and "Nobita" in i:
        res.append(i)

print(len(res), "bài viết chứa từ khóa 'Nobita'")

✓ 0.0s

Tổng số bài viết: 130
10 bài viết chứa từ khóa 'Nobita'
```

### 3.2, Tần suất các từ được sử dụng nhiều nhất



-Từ khóa Doraemon là 1 trong từ khóa xuất hiện gần như là nhiều nhất trong các bài post không quá ngạc nhiên vì Page này được lập ra là dành cho chú mèo máy cũng như bộ truyện cùng tên

-Bên cạnh đó những từ khóa như : “Bộ”,”Hình”,”Movie”,... là những từ khóa liên quan đến bộ phim cũng xuất hiện khá nhiều

### 3.3,Tìm mức độ quan trọng của các từ trong bài post

-Số từ và câu xuất hiện trong các bài post của page,những bài post khá là ngắn sau khi chúng ta thấy được số từ và số câu mà Page đã sử dụng:

```
# tìm số từ xuất hiện trong tài liệu
total_words = text.split()
total_word_length = len(total_words)
print(total_word_length)

#tìm số câu xuất hiện trong tài liệu
total_sentences = tokenize.sent_tokenize(text)
total_sent_len = len(total_sentences)
print(total_sent_len)

✓ 0.0s
```

1694

57

-Sử dụng file stop\_word và file stop\_word1 được lấy từ stopword Việt nam để tìm mức độ quan trọng của các từ so với bài viết:

```
# Lấy N từ quan trọng nhất
def get_top_n(dict_elem, n):
    result = dict(sorted(dict_elem.items(), key = itemgetter(1), reverse = True)[:n])
    return result

from operator import itemgetter
get_top_n(tf_idf_score, 10)
✓ 0.0s

{':)': 0.026221494608014668,
 '#on': 0.010432401142498483,
 'v': 0.010324320309196125,
 'Ành': 0.009546756240459387,
 '☺': 0.009410000850229019,
 '<3': 0.009410000850229019,
 'mèo': 0.00927324545999865,
 'Việt': 0.008717009164100734,
 'Doon': 0.008680611828377968,
 'Cre:': 0.008278218159633383}
```

## Phân 3 : Phân tích, đánh giá comments

### 1.Tổng số comments

Sum comments

+ Code + Markdown

```
x = post_df_full["comments"].sum()
print("Tổng số comments là :",x)
✓ 0.0s
```

Tổng số comments là : 34101

## 2. Lọc comments

### Filter comments

```
post_cmt = post_df_full["comments_full"].tolist()

arr = []
for i in range(len(post_cmt)):
    post_cmt[i] = post_cmt[i].replace("[","")
    post_cmt[i] = post_cmt[i].replace("]", "")

input_str = post_cmt[i]

# Xóa dấu ngoặc đối với các thành phần và chia chuỗi thành một danh sách
list_result = [item.strip('{}').split(',') for item in input_str.split('}, {')]

arr.append(list_result)
```

### GET TEXT

```
list_cmt = []
for i in range(len(arr)) :
    ss = ""
    for j in range(len(arr[i])) :
        if len(arr[i][j]) > 6:
            modified_str = arr[i][j][6].replace(":", "")
            ss = ss + modified_str + "\n"
    list_cmt.append(ss)

for i in range(len(list_cmt)) :
    list_cmt[i] = list_cmt[i].replace("comment_text","");
    list_cmt[i] = list_cmt[i].replace("'", "")
```

-Hiển thị toàn bộ comments sau khi đã được lọc và làm sạch:



## 4.Phân tích độ tích cực, tiêu cực của các comments

-Những comments được đánh giá là tích cực sau khi phân tích bằng SentimentAnalyzer:

```
import nltk
from nltk.sentiment import SentimentIntensityAnalyzer

# Tải dữ liệu cho việc phân tích cảm xúc
nltk.download('vader_lexicon')

# Khởi tạo đối tượng SentimentIntensityAnalyzer
sid = SentimentIntensityAnalyzer()

# Phân tích cảm xúc
sentiment_scores = sid.polarity_scores(text)

# In kết quả
print(sentiment_scores)

# Xác định là tích cực, tiêu cực, hay trung tính dựa trên điểm số
if sentiment_scores['compound'] >= 0.05:
    print("Bình luận của các bài viết mang hướng tích cực")
elif sentiment_scores['compound'] <= -0.05:
    print("Bình luận của các bài viết mang hướng tiêu cực")
else:
    print("Bình luận của các bài viết mang hướng trung tính")

[nltk_data] Downloading package vader_lexicon to
[nltk_data]     C:\Users\FPT\AppData\Roaming\nltk_data...
[nltk_data]     Package vader_lexicon is already up-to-date!
{'neg': 0.006, 'neu': 0.968, 'pos': 0.026, 'compound': 0.9999}
Bình luận của các bài viết mang hướng tích cực
```

•Các chỉ số được tính toán ra là:

NEG:0.006

NEU:0.968

POS:0.026

## 5.Phân tích người bình luận trong các bài Post

### 5.1, Số người dùng comments

```
#số người đã comments
print(f"Trong tổng số {x} comments được comments từ {len(list_data)} người dùng Facebook")
✓ 0.0s
Trong tổng số 34101 comments được comments từ 3032 người dùng Facebook
```

-Từ 34101 bình luận thì có 3032 sử dụng tính năng bình luận, tính ra mỗi người đã comments hơn 10 lần trong bài viết, một con số khá tương đối.Chứng tỏ độ tương tác qua lại của người dùng khá là cao, thân thiện với nhau

## 5.2, Phân tích

### 5.2.1, Lấy ID người dùng

-Sử dụng những kỹ năng xử lý chuỗi đã học được để tách ID ra :

```
#chuyển id thành danh sách để tiện thao tác
list_data = []
for i in range(len(list_id)) :
    string_data = list_id[i]
    cleaned_string = string_data.strip(',')
    list_dt = cleaned_string.split(',')
    for j in range(len(list_dt)) :
        list_data.append(list_dt[j])

```

-ID sau khi đã tách ra được :

```
list_data
✓ 0.0s
['1002788703720354',
 '1003909156922620',
 '1005109770773283',
 '1005197783886193',
 '1006304883800272',
 '1006603740558047',
 '1006760720015816',
 '1010047566320722',
 '1011169006769782',
 '1012870949958212',
 '1014347043203471',
 '1015130299884766',
 '1015573366191934',
 '1018089106099318',
 '1018879976017016',
 '1020169688606018',
 '1021495538479623',
 '1024794831544262',
 '1025749608073461',
 '1028437801087382',
 '1028512611607373',
 '1029529937617128',
 '1029537927744081',
 '1029712427745769',
 '1030307091501403',
 ...
 '2514345108695919',
 '2514365158693914',
 '2514449085352188',
 '2514453485351748',
 ...]
```

### 5.2.2, Lấy thông tin người dùng

-Chúng ta sử dụng hàm get\_profile để lấy thông tin của người dùng, do thông tin người dùng là bảo mật, nên chúng ta chỉ lấy được một số dữ liệu hạn chế:

## GET INFO

```
commenter = []
for i in range(len(list_data)):
    try:
        x = int(list_data[i])
        user = get_profile(str(x))
        commenter.append(user)
    except:
        continue
```

-Những thông tin thu thập được :

```
commenter
✓ 0.2s
[{'Friend_count': None,
 'Follower_count': None,
 'Following_count': None,
 'cover_photo': 'https://scontent.fhan14-3.fna.fbcdn.net/v/t39.30808-6/397909431_6693415904028563_6465052904712400856_n.jpg?stp=cp1_d_e3s_3x3_0_0',
 'profile_picture': 'https://scontent.fhan14-3.fna.fbcdn.net/v/t39.30808-1/386516634_6593269764043178_8109638575334085891_n.jpg?stp=cp0_d_e3s_3x3_0_0',
 'id': '100000806580700',
 'Name': 'Võ Anh Hiền (ヴォ アインヒエン)',
 'Công việc': '株式会社ガイアフィールド\nArchitect\n1 tháng 9, 2018 - Hiện tại\nKawasaki-shi, Kanagawa, Japan',
 'Nơi từng sống': 'Nha Trang\nQuê quán',
 'Trích dẫn yêu thích': 'Hãy đặt ra mục tiêu chứ đừng ước mơ !!!'},
 {'Friend_count': None,
 'Follower_count': None,
 'Following_count': None,
 'cover_photo': 'https://scontent.fhan14-3.fna.fbcdn.net/v/t39.30808-6/329120872_587764266540352_5648901694147592340_n.jpg?stp=cp0_d_e3s_3x3_0_0',
 'profile_picture': 'https://scontent.fhan14-3.fna.fbcdn.net/v/t39.30808-1/328048646_529790815650891_6775877252311805013_n.jpg?stp=cp0_d_e3s_3x3_0_0',
 'id': '100005919630625',
 'Name': 'Như Quỳnh',
 'Học vấn': 'Bishop McNally High School\nTrường trung học\n5 tháng 9, 2017 - Hiện tại\nTrường Tiểu học - THPT Bùi Thị Xuân Biên',
 'Nơi từng sống': 'Calgary\nTỉnh/Thành phố hiện tại\nBiên Hòa\nQuê quán'},
 {'Friend_count': None,
 'Follower_count': None,
 'Following_count': None,
 'cover_photo': 'https://scontent.fhan14-3.fna.fbcdn.net/v/t1.18169-9/11896199_834467459982857_3836894243525342025_n.jpg?stp=cp0_dst_e3s_3x3_0_0',
 'profile_picture': 'https://scontent.fhan14-3.fna.fbcdn.net/v/t39.30808-1/361349114_6321771347919080_4207761367201768874_n.jpg?stp=cp0_d_e3s_3x3_0_0',
 'id': '100002588330501'},
```

### 5.2.3, Phân tích

-Sau khi lấy được 1 số thông tin về những người comments, chúng ta cùng qua một vài bước xử lý và lọc ra được thành phố của họ đang sinh sống :

```

●  from unidecode import unidecode
    import re
✓  0.0s

# Khởi tạo dictionary để lưu thông tin thành phố và số lượng người
ds_tp = {}

# Duyệt qua danh sách người bình luận
for i in range(len(ds)):
    try:
        x = ds[i]['Nơi từng sống'].split('\n')
        y = x[0].split(',')

        # Xác định thành phố và tăng số lượng người sống ở thành phố đó
        if len(y) == 1:
            city = unidecode(y[0].lower())
        elif len(y) == 2:
            if y[1] == 'Vietnam':
                city = unidecode(y[0].lower())
            else:
                city = unidecode(y[1].lower())
        else:
            city = unidecode(y[-2].lower())

        # Xóa các ký tự đặc biệt
        city = re.sub(r'\W+', '', city)

        # Tăng số lượng người sống ở thành phố
        ds_tp[city] = ds_tp.get(city, 0) + 1

    except Exception as e:
        continue

```

-Thông tin về những thành phố đã được tổng hợp :

	Thành phố	Số lượng người
14	thanhphochiminh	107
4	hanoi	85
9	danang	14
15	haiphong	13
39	hungyen	9
...	...	...
59	thuanthanh	1
1	calgary	1
56	caibe	1
55	hibac	1
115	dalat	1

-Qua số liệu trên chúng ta thấy được Page thu hút rất nhiều những người đến từ các thành phố khác nhau, có cả những người đang làm việc và sinh sống tại nước ngoài.Nhưng số lượng đông nhất vẫn đến từ 2 thành phố lớn nhất Việt Nam đó là “Thành Phố Hồ Chí Minh” và “Hà Nội”

Phân phối số lượng người theo thành phố

