PHÂN TÍCH SỰ QUAN TÂM CỦA NGƯỜI DÙNG FACEBOOK ĐỐI VỚI MỘT SẢN PHẨM

Link youtube:

+ phân tích reactions: <https://youtu.be/0O_1evgVUHk>

+ phân tích comments: <https://youtu.be/32hDdZeS7Go>

**THÔNG TIN NHÓM 11**

|  |  |
| --- | --- |
| Họ và tên thành viên | Đóng góp và tự đánh giá |
| 1. Lý Gia Hưng – 1050080053 | - Đóng góp: Cài đặt mô hình, thực nghiệm  - Tự đánh giá quá trình: 8  - Tự đánh giá cuối kì: 8 |
| 1. Hoàng Phúc Lộc - 1050080058 | - Đóng góp: Cài đặt mô hình, thực nghiệm  - Tự đánh giá quá trình: 8  - Tự đánh giá cuối kì:8 |

|  |  |
| --- | --- |
| 1. Nguyễn Khắc Thiên – 1050080075 | - Đóng góp: Cài đặt mô hình, thực nghiệm  - Tự đánh giá quá trình: 7.5  - Tự đánh giá cuối kì: 8 |

PHỤ LỤC

[TÓM TẮT BÁO CÁO 4](#_Toc149698683)

[1. Lý do chọn đề tài 4](#_Toc149698684)

[2. Tầm quan trọng của đề tài 4](#_Toc149698685)

[3. Nội dung báo cáo 5](#_Toc149698686)

[BÁO CÁO CHI TIẾT 6](#_Toc149698687)

[1. Giới thiệu Big Data 6](#_Toc149698688)

[2. Bài toán 7](#_Toc149698689)

[3. Công nghệ sử dụng 8](#_Toc149698690)

[4. Mô hình 11](#_Toc149698691)

[5. Thực nghiệm và demo 13](#_Toc149698692)

[5.1. Lấy Token có quyền Facebook Graph API để lấy được thông tin cần thiết của bài viết trên facebook 14](#_Toc149698693)

[5.2. Lấy dữ liệu comments của bài viết bằng API Graph Facebook 17](#_Toc149698694)

[5.2. Lấy thông tin của người comment bài viết 22](#_Toc149698695)

[5.3. Thu thập dữ liệu reactions bằng facebook\_scaper 29](#_Toc149698696)

[5.4. Xử lý – làm sạch dữ liệu 32](#_Toc149698697)

# TÓM TẮT BÁO CÁO

## 1. Lý do chọn đề tài

Hiểu sâu về người dùng: Phân tích dữ liệu từ Facebook giúp doanh nghiệp hiểu rõ hơn về đối tượng tiềm năng, sở thích, và nhu cầu của họ. Điều này có thể giúp cải thiện chiến dịch tiếp thị và phát triển sản phẩm.

Đánh giá sự quan tâm: Dữ liệu từ Facebook có thể giúp đánh giá mức độ quan tâm của người dùng đối với sản phẩm, dự đoán sự thành công hoặc thất bại của sản phẩm trước khi ra mắt.

Tối ưu hóa chiến dịch tiếp thị: Dựa trên thông tin từ Big Data, doanh nghiệp có thể tối ưu hóa chiến dịch tiếp thị, chọn đúng phương tiện truyền thông xã hội và thời điểm phù hợp để tạo sự quan tâm đối với sản phẩm.

Đối phó với phản hồi của người dùng: Big Data cũng giúp doanh nghiệp phát hiện sớm phản hồi và phàn nàn của người dùng, giúp họ có cơ hội giải quyết vấn đề và cải thiện sản phẩm.

Tóm lại, đề tài này đem lại giá trị to lớn cho việc phát triển sản phẩm và chiến dịch tiếp thị thông qua sử dụng Big Data để nắm bắt thông tin về sự quan tâm của người dùng trên mạng xã hội.

## 2. Tầm quan trọng của đề tài

"Phân tích sự quan tâm của người dùng Facebook về một sản phẩm" trong lĩnh vực Big Data không thể bỏ qua. Việc thu thập, xử lý và hiểu sâu về thông tin mà người dùng chia sẻ trên mạng xã hội giúp doanh nghiệp hiểu rõ hơn về nguy cơ và cơ hội liên quan đến sản phẩm của họ.

## 3. Nội dung báo cáo

- Mô tả bài toán

- Các công nghệ sử dụng

- Mô hình

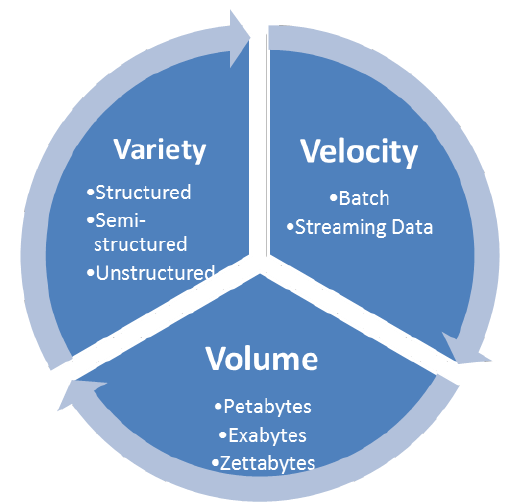
- Thực nghiệm và demo  
- Đánh giá

- Tài liêu tham khảo

# BÁO CÁO CHI TIẾT

## 1. Giới thiệu Big Data

Big Data là một thuật ngữ thường được sử dụng để mô tả lượng dữ liệu lớn và phức tạp, vượt xa khả năng của công cụ và phương pháp truyền thống để quản lý, xử lý và phân tích. Dữ liệu này có thể bao gồm thông tin từ nhiều nguồn khác nhau như máy tính cá nhân, thiết bị di động, cảm biến, mạng xã hội, dữ liệu giao dịch, hình ảnh, video và nhiều nguồn dữ liệu khác. Dữ liệu Big Data thường được đánh giá dựa trên ba yếu tố chính, được gọi là "3V":



* Volume (Lượng dữ liệu): Big Data thường chứa lượng dữ liệu lớn. Điều này có thể bao gồm terabytes, petabytes hoặc thậm chí exabytes dữ liệu.
* Velocity (Tốc độ): Dữ liệu này thường được tạo ra và cập nhật với tốc độ nhanh, đặc biệt là trong các ngữ cảnh như Internet of Things (IoT) hoặc trên mạng xã hội.
* Variety (Đa dạng): Big Data đến từ nhiều nguồn, và nó có thể tồn tại ở nhiều định dạng khác nhau, bao gồm văn bản, hình ảnh, video, âm thanh và dữ liệu không cấu trúc.

Big Data có nhiều ứng dụng quan trọng, bao gồm:

* Phân tích dự đoán: Sử dụng Big Data để dự đoán xu hướng tương lai, ví dụ như dự đoán sự quan tâm của người dùng đối với một sản phẩm.
* Quản lý dữ liệu khách hàng: Doanh nghiệp sử dụng Big Data để hiểu rõ hơn về khách hàng của họ và cải thiện dịch vụ hoặc sản phẩm.
* Nghiên cứu khoa học: Big Data hỗ trợ nghiên cứu trong nhiều lĩnh vực, từ y học đến khoa học xã hội.
* An toàn mạng: Big Data có vai trò quan trọng trong phát hiện và ngăn chặn các mối đe dọa an ninh trực tuyến.
* Dự đoán và tối ưu hóa vận hành: Các ngành công nghiệp như logistics và sản xuất sử dụng Big Data để tối ưu hóa quy trình vận hành và dự đoán cần tài nguyên gì và khi nào.

Big Data đã trở thành một phần quan trọng của nhiều lĩnh vực và mang lại nhiều cơ hội và thách thức cho doanh nghiệp và tổ chức trên khắp thế giới.

## 2. Bài toán

Bài toán: Các doanh nghiệp đăng bài về sản phẩm của họ lên faebook dưới dạng một bài post. Để đánh giá sản phẩm đó, chúng ta không thể nào ngồi đếm từng lượt tương tác (reactions) của người dùng facebook cũng như không thể ngồi đọc từng comment họ nói về sản phẩm được.

Bài toán đặt ra cho chủ đề này là phải thu thập dữ liệu từ một bài viết

* Thu thập về lượt reaction: thích, thương thương, yêu thích, bất ngờ, tức giận, buồn bã.
* Thông tin của comment
* Thông tin của người dùng tương tác với bài viết: tuổi, giới tính, nơi sống, quê quán,..

Sau khi có được dữ liệu, ta phải xử lý dữ liệu rồi phân tích sự quan tâm từ số liệu mới thu được rồi cuối cùng là đưa ra kết quả phân tích.

A diagram of a phone

Description automatically generated with medium confidence

## 3. Công nghệ sử dụng

- Môi trường thực nghiệm (IDE): Google Colab

Google Colab là một môi trường phát triển tích hợp trực tuyến (Integrated Development Environment - IDE) dựa trên nền tảng Jupyter Notebook. Môi trường này đã trở thành công cụ quan trọng trong lĩnh vực học máy và phân tích dữ liệu. Google Colab cho phép các nhà nghiên cứu và nhà phát triển làm việc trên cùng một tài liệu, chia sẻ dự án, và sử dụng tài nguyên tính toán mạnh mẽ, bao gồm cả tài nguyên GPU và TPU, mà không cần phải cài đặt bất kỳ phần mềm nào trên máy tính cá nhân.

- Ngôn ngữ sử dụng: Python

Python là một trong những ngôn ngữ lập trình phổ biến nhất trên thế giới, và nó đã thống trị lĩnh vực học máy và phân tích dữ liệu. Sự phổ biến của Python đến từ cú pháp đơn giản, cộng đồng lớn mạnh, và các thư viện phong phú như NumPy, pandas, scikit-learn, và TensorFlow, giúp người dùng thực hiện các tác vụ phức tạp một cách dễ dàng.

- Cloud : Googe Driver

Google Drive là một dịch vụ lưu trữ đám mây được tích hợp chặt chẽ với Google Colab. Nó cho phép bạn lưu trữ và chia sẻ dự án và dữ liệu dễ dàng, giúp bạn truy cập dữ liệu từ mọi nơi với kết nối internet. Sự kết hợp giữa Google Colab và Google Drive làm cho việc làm việc trên dự án cùng với đồng nghiệp trở nên dễ dàng hơn bao giờ hết.

- Facebook Graph API

Facebook Graph API là một tập hợp các giao diện lập trình ứng dụng (API) do Facebook cung cấp, cho phép các nhà phát triển truy cập và tương tác với dữ liệu trên mạng xã hội Facebook. Các ứng dụng sử dụng Facebook Graph API có thể trích xuất thông tin về người dùng, bài viết, trang, nhóm, và nhiều loại dữ liệu khác từ Facebook. Điều này làm cho nó trở thành một công cụ quan trọng trong việc nghiên cứu và phân tích dữ liệu xã hội.

- facebook\_scaper

facebook\_scraper là một thư viện Python giúp thu thập dữ liệu từ Facebook thông qua web scraping. Thư viện này cho phép bạn trích xuất thông tin từ bài viết, bình luận, và nhiều thành phần khác trên trang Facebook một cách tự động. Nó cung cấp một phương tiện thuận tiện để thu thập dữ liệu từ Facebook cho các nghiên cứu và dự án liên quan đến mạng xã hội.  
- Matplotlib

Matplotlib là một thư viện Python mạnh mẽ dùng để vẽ biểu đồ và trực quan hóa dữ liệu. Nó cung cấp các công cụ để tạo các biểu đồ đường, biểu đồ cột, biểu đồ phân tán, biểu đồ hình tròn, và nhiều loại biểu đồ khác. Matplotlib giúp người dùng hiển thị kết quả phân tích dữ liệu một cách trực quan và dễ hiểu.

- Mô hình Naive Bayes

Mô hình Naive Bayes là một mô hình học máy dựa trên nguyên tắc xác suất Bayes và giả định "ngây thơ" (naive) rằng tất cả các đặc trưng đầu vào độc lập với nhau. Mô hình này thường được sử dụng trong các tác vụ phân loại và phân tích dữ liệu. Một trong những ứng dụng phổ biến nhất của Naive Bayes là phân loại văn bản, ví dụ như phân loại email thành email thường và email rác (spam).

+ Ưu điểm của Mô hình Naive Bayes:

Dễ triển khai và hiệu quả: Mô hình Naive Bayes dễ triển khai và hoạt động tốt trên các tập dữ liệu lớn. Nó cần ít dữ liệu huấn luyện so với một số mô hình phức tạp khác.

Xử lý dữ liệu văn bản tốt: Naive Bayes thường hiệu quả trong việc phân loại văn bản, ví dụ: phân tích cảm xúc từ các bài đăng trên mạng xã hội, phân loại tin tức, hoặc phát hiện spam email.

Khả năng xử lý đặc trưng lớn: Mô hình Naive Bayes có khả năng xử lý một lượng lớn các đặc trưng (features) mà không gây hiện tượng quá khớp (overfitting).

Không đòi hỏi nhiều tài nguyên tính toán: Mô hình Naive Bayes cần ít tài nguyên tính toán so với một số mô hình phức tạp khác như các mạng nơ-ron sâu (deep neural networks).

+ Nhược điểm của Mô hình Naive Bayes:

Giả định ngây thơ không thực tế: Giả định rằng tất cả các đặc trưng đầu vào độc lập với nhau có thể không phù hợp với thực tế trong một số tình huống. Tuy nhiên, mô hình này vẫn hoạt động tốt trong nhiều trường hợp thực tế.

Hiệu suất có thể bị hạn chế: Mô hình Naive Bayes có thể có hiệu suất thấp hơn so với một số mô hình phức tạp khác, đặc biệt trong trường hợp dữ liệu phức tạp và không tuân theo giả định ngây thơ.

Khả năng xử lý các tương tác phức tạp: Mô hình này không thể mô hình hóa các tương tác phức tạp giữa các đặc trưng. Điều này có nghĩa rằng nó có thể không phù hợp cho các tác vụ yêu cầu mô hình hóa các mối quan hệ phức tạp giữa các biến đầu vào.

## 4. Mô hình

- Mô hình:

A diagram of a company

Description automatically generated

A diagram of a company

Description automatically generated

## 5. Thực nghiệm và demo

- Bài viết thực nghiệm và demo sẽ lấy bài viết này: [https://www.facebook.com/phuclongcoffeeandtea/posts/pfbid0MuhMCqxy5gVdvPNTNsTh96F4r8iNnZJZUAehjMtRxrCJi3ndszwq61aGEkX9KUJxl](https://www.facebook.com/phuclongcoffeeandtea/posts/pfbid0MuhMCqxy5gVdvPNTNsTh96F4r8iNnZJZUAehjMtRxrCJi3ndszwq61aGEkX9KUJxl%20) A screenshot of a social media post

Description automatically generated

### 5.1. Lấy Token có quyền Facebook Graph API để lấy được thông tin cần thiết của bài viết trên facebook

- Tạo một tài khoản <https://developers.facebook.com/>

- Lấy full quyền acess token thong qua ứng dụng. Ở đây chúng ta sẽ dùng thông qua Instagram

A screenshot of a computer

Description automatically generated

- Sao chép đường dẫn sau dán vào trình duyệt: view-source:https://www.facebook.com/dialog/oauth?client\_id=124024574287414&redirect\_uri=https://www.instagram.com/accounts/signup/&&scope=email&response\_type=token



- Sau đó ta được đoạn token có kí tự đầu là “EAAB”

- Copy accessToken như trên sau đó vào link dưới để check quyền:  
<https://developers.facebook.com/tools/debug/accesstoken/>

A screenshot of a computer

Description automatically generated  
  
- Lấy cookies từ accessToken trên

https://graph.facebook.com/me?access\_token=YOUR\_ACCESS\_TOKEN\_HERE

Điền access\_token =

Nhấn F12 và load lại trang.

- Bấm vào tab network -> status: 200 -> Headers -> Request Headers -> Copy Cookie  
A screenshot of a computer

Description automatically generated

- Sau bước trên thì ta đã có 2 loại mã là token và cookie.

### 5.2. Lấy dữ liệu comments của bài viết bằng API Graph Facebook

- Tạo một phiên làm việc với Google Colab

- Nhập lệnh sau :

from google.colab import drive

drive.mount("/content/drive/")

import requests

import json

import csv

def get\_all\_comments(post\_id, access\_token):

    comments\_data = []

    next\_url = f"https://graph.facebook.com/{post\_id}/comments?access\_token={access\_token}"

    while next\_url:

        response = requests.get(next\_url)

        data = response.json()

        if "data" in data:

            comments\_data.extend(data["data"])

        if "paging" in data and "next" in data["paging"]:

            next\_url = data["paging"]["next"]

        else:

            next\_url = None

    return comments\_data

# Thay thế POST\_ID và ACCESS\_TOKEN bằng giá trị thích hợp của bạn

POST\_ID = 'pfbid02RTLY81bKvyjp3sDsu1owZ98vkmZbNkMa5eSo6VjUxrw4QZY1TCx5cmYsWYMac6VNl'

ACCESS\_TOKEN = 'EAABwzLixnjYBO4k1s2TYKguBgwapSsTo4uG4NNVIO1gKlnXZBS36VABgcBJk3rBZBUUMt1IDv2QHP5YNSO9KrDpqoGXwLB0w31padMUT6JlWMJMISjrl7ND2ZCMm7SxOiEDoCr7UZCndXS6S8cvBePOxGduSGkR49oWh42ZAstlg7k9cL4chTu31JUWMokxCDPZCaAAfaa8IAZD'

comments\_data = get\_all\_comments(POST\_ID, ACCESS\_TOKEN)

if comments\_data:

    # Ghi vào tệp CSV

    csv\_path = '/content/drive/MyDrive/project\_facebook\_bigdata/all\_data\_file/comments.csv'

    with open(csv\_path, 'w', encoding='utf-8', newline='') as csv\_file:

        csv\_writer = csv.writer(csv\_file)

        csv\_writer.writerow(['message', 'created\_time', 'from\_name', 'from\_id'])  # Tiêu đề cột

        for comment in comments\_data:

            csv\_writer.writerow([comment.get('message', ''), comment.get('created\_time', ''), comment.get('from', {}).get('name', ''), comment.get('from', {}).get('id', '')])

    # Ghi vào tệp JSON

    json\_path = '/content/drive/MyDrive/project\_facebook\_bigdata/all\_data\_file/comments.json'

    with open(json\_path, 'w', encoding='utf-8') as json\_file:

        json.dump(comments\_data, json\_file, ensure\_ascii=False, indent=4)

    print("Dữ liệu bình luận đã được ghi vào comments.csv và comments.json trên Google Drive")

    print("Đường dẫn tới tệp CSV:", csv\_path)

    print("Đường dẫn tới tệp JSON:", json\_path)

else:

    print("Không tìm thấy thông tin về bình luận cho bài viết này.")

>> Dữ liệu bình luận đã được ghi vào comments.csv và comments.json trên Google Drive

Đường dẫn tới tệp CSV: /content/drive/MyDrive/project\_facebook\_bigdata/all\_data\_file/comments.csv

Đường dẫn tới tệp JSON: /content/drive/MyDrive/project\_facebook\_bigdata/all\_data\_file/comments.json

Vậy là ta đã thu được dữ liệu của comments về 2 dạng file json và csv

A screenshot of a computer program

Description automatically generatedA screenshot of a computer

Description automatically generated

### 5.2. Lấy thông tin của người comment bài viết

- Tạo một phiên làm việc mới với google colab

from google.colab import drive

drive.mount("/content/drive")

import json

import csv

# Đường dẫn đến tệp JSON trên Google Drive

file\_json\_data = '/content/drive/MyDrive/project\_facebook\_bigdata/all\_data\_file/comments.json'

# Đọc tệp JSON

with open(file\_json\_data, 'r') as file:

   data = json.load(file)

data

A computer screen shot of a computer code

Description automatically generated

Chúng ta sẽ thấy trong dữ liệu comments bài viết sẽ có một document dạng 'from': {'name': 'Nguyễn Hồng Tâm', 'id': '100010678567302'},

Dựa vào id ta sẽ dùng API graph Facebook để lấy thông tin user

# Đọc tệp JSON

with open(file\_json\_data, 'r') as file:

    data = json.load(file)

# Tạo danh sách rỗng để lưu trữ thông tin name và id

names\_and\_ids = []

# Lặp qua các đối tượng trong tệp JSON và trích xuất name và id

for item in data:

    if 'from' in item and 'name' in item['from'] and 'id' in item['from']:

        name = item['from']['name']

        user\_id = item['from']['id']

        names\_and\_ids.append({'name': name, 'id': user\_id})

print(len(names\_and\_ids))

names\_and\_ids

A screenshot of a computer

Description automatically generated

import json

import requests

import csv

ACCESS\_TOKEN = 'EAABwzLixnjYBO4k1s2TYKguBgwapSsTo4uG4NNVIO1gKlnXZBS36VABgcBJk3rBZBUUMt1IDv2QHP5YNSO9KrDpqoGXwLB0w31padMUT6JlWMJMISjrl7ND2ZCMm7SxOiEDoCr7UZCndXS6S8cvBePOxGduSGkR49oWh42ZAstlg7k9cL4chTu31JUWMokxCDPZCaAAfaa8IAZD'

FIELDS = 'id,name,gender,location,hometown,birthday'

# # Đọc danh sách người dùng từ tệp JSON

# input\_json\_file = '/content/name\_id.json'  # dẫn tới file json có id và name

# with open(input\_json\_file, 'r', encoding='utf-8') as json\_file:

#     user\_data = json.load(json\_file)

# Mở tệp CSV để ghi

output\_csv\_file = '/content/drive/MyDrive/project\_facebook\_bigdata/all\_data\_file/user\_data.csv'

with open(output\_csv\_file, 'w', newline='', encoding='utf-8') as csv\_file:

    # Tạo một đối tượng CSV writer

    csv\_writer = csv.writer(csv\_file)

    # Ghi tiêu đề (header) của các trường

    header = FIELDS.split(',')

    csv\_writer.writerow(header)

    # Lặp qua danh sách người dùng và lấy thông tin từ Facebook Graph API

    for user in names\_and\_ids:

        user\_id = user['id']

        url = f'https://graph.facebook.com/{user\_id}?fields={FIELDS}&access\_token={ACCESS\_TOKEN}'

        response = requests.get(url)

        user\_info = response.json()

        # Tạo một danh sách chứa giá trị của các trường

        row = [user\_info.get(field, '') for field in header]

        csv\_writer.writerow(row)

        print(f"Đã lấy dữ liệu cho {user\_info.get('name')}")

print("Hoàn thành việc lấy dữ liệu và lưu vào tệp user\_data.csv")

print("Số data user lấy được là:", len(names\_and\_ids))

A screenshot of a computer

Description automatically generated

Chúng ta đã lấy được dữ liệu của những user comment bài viết gồm có các trường như gender, location,hometwon,birthday,..

A screenshot of a computer

Description automatically generated

### 5.3. Thu thập dữ liệu reactions bằng facebook\_scaper

- Tạo một phiên làm việc mới trên google colab

- Thực hiện cài đặt công cụ facebook\_scaper

from google.colab import drive

drive.mount("/content/drive")

%pip install facebook\_scraper

import facebook\_scraper as fs

import pandas as pd

POST\_ID = "pfbid02RTLY81bKvyjp3sDsu1owZ98vkmZbNkMa5eSo6VjUxrw4QZY1TCx5cmYsWYMac6VNl"

COOKIE\_PATH= "/content/drive/MyDrive/test scap/facebook.com\_cookies.txt"

MAX\_COMMENTS = True

# get the post (this gives a generator)

gen = fs.get\_posts(

    post\_urls=[POST\_ID],

    options={"reactions": True, "allow\_extra\_requests": True}, # option có những gì thì đọc docs, nhiều quá không viết hết được https://github.com/kevinzg/facebook-scraper

   cookies=COOKIE\_PATH) # ở đay chỉ lấy data về reactions thôi nhé

# take 1st element of the generator which is the post we requested

post = next(gen)

A screenshot of a computer

Description automatically generated

Chúng ta sẽ có thông tin về reacion như trên, giờ thì chúng ta sẽ lưu lại:

import json

# Tạo một từ điển chứa các trường bạn muốn lưu

data\_to\_save = {

    "reactions": post['reactions'],

    "reaction\_count": post['reaction\_count'],

    "comments": post['comments'],

    "shares": post['shares']

}

# Mở tệp JSON để lưu dữ liệu

with open("/content/drive/MyDrive/project\_facebook\_bigdata/all\_data\_file/reactions\_data.json", "w") as json\_file:

    json.dump(data\_to\_save, json\_file)

### 5.4. Xử lý – làm sạch dữ liệu

- Đầu tiên ta vẫn tạo một phiên làm việc mới với google colab và lấy file dữ liệu comments mới lấy ở mục 5.2 lên sử dụng

from google.colab import drive

drive.mount("/content/drive")

import json

import csv

# Đường dẫn đến tệp JSON trên Google Drive

file\_json\_data = '/content/drive/MyDrive/project\_facebook\_bigdata/all\_data\_file/comments.json'

# Đọc tệp JSON

with open(file\_json\_data, 'r') as file:

    data\_comments = json.load(file)

data\_comments

- Chúng ta xử lý loại bỏ icon và kí tự đặc biệt ở trong comments như sau

import re

# Sử dụng danh sách để lưu trữ tất cả các giá trị "message"

message\_list = []

# Duyệt qua danh sách JSON và thu thập giá trị "message"

for comment in data\_comments:

    if 'message' in comment:

        message = comment['message']

        # Loại bỏ các biểu tượng hoặc ký hiệu không phải chữ cái, số hoặc khoảng trắng

        message = re.sub(r'[^\w\s]', '', message)

        # Loại bỏ khoảng trắng không cần thiết ở đầu và cuối của message

        message = message.strip()

        # Đóng message trong dấu ngoặc kép (")

        message = f'"{message}"'

        message\_list.append(message)

# Ghép các giá trị "message" lại thành một chuỗi, cách nhau bởi dấu phẩy

combined\_messages = ', '.join(message\_list)

# In ra chuỗi kết quả

print(f"Tổng số bình luận:", len(data\_comments))

print(combined\_messages)

- Chúng ta sẽ có kết quả

A close-up of a document

Description automatically generated

- Sau khi xử lý xong ,chúng ta sẽ xuất ra file định dạng excel   
import re

import pandas as pd

# Sử dụng danh sách để lưu trữ tất cả các giá trị "message"

message\_list = []

# Duyệt qua danh sách JSON và thu thập giá trị "message"

for comment in data\_comments:

    if 'message' in comment:

        message = comment['message']

        # Loại bỏ các biểu tượng hoặc ký hiệu không phải chữ cái, số hoặc khoảng trắng

        message = re.sub(r'[^\w\s]', '', message)

        # Loại bỏ khoảng trắng không cần thiết ở đầu và cuối của message

        message = message.strip()

        message\_list.append(message)

# Tạo DataFrame từ danh sách message

df = pd.DataFrame(message\_list, columns=["Message"])

# Lưu DataFrame thành tệp Excel

df.to\_excel("/content/drive/MyDrive/project\_facebook\_bigdata/all\_data\_file/comments\_message.xlsx", index=False)

print("Dữ liệu đã được xuất thành công vào tệp 'comments\_message.xlsx'.")

Vậy là chúng ta đã xử lý và làm sạch nội dung comments.

- Tiếp theo chúng ta sẽ xử lý dữ liệu của những người dùng comments vào bài viết

import csv

user\_data = "/content/drive/MyDrive/project\_facebook\_bigdata/all\_data\_file/user\_data.csv"

# Khởi tạo một danh sách trống để lưu trữ dữ liệu từ tệp CSV

user\_data\_list = []

# Mở tệp CSV và đọc nội dung bằng csv.reader

with open(user\_data, 'r', encoding='utf-8') as csv\_file:

    csv\_reader = csv.reader(csv\_file)

    for row in csv\_reader:

        # Mỗi hàng trong tệp CSV sẽ là một danh sách (list)

        user\_data\_list.append(row)

user\_data\_list

A screenshot of a computer

Description automatically generated

- Sau khi chúng ta có dữ liêu danh sách người dùng và những trường cần thiết, bây giờ chúng ta sẽ lọc và lấy thông tin những trường đó để đi phân tích

import csv

from datetime import datetime

csv\_file\_path = '/content/drive/MyDrive/project\_facebook\_bigdata/all\_data\_file/user\_data.csv'

output\_csv\_file\_path = '/content/drive/MyDrive/project\_facebook\_bigdata/fresh\_data\_clean/user\_age.csv'

def calculate\_age(birthday):

    if not birthday or birthday.strip() == '':

        return 'không xác định'

    else:

        try:

            date\_obj = datetime.strptime(birthday, "%m/%d/%Y")

            current\_year = datetime.now().year

            age = current\_year - date\_obj.year

            return age

        except ValueError:

            return 'không xác định'

# Khởi tạo danh sách trống để lưu trữ tuổi

age\_list = []

# Mở tệp CSV và đọc nội dung

with open(csv\_file\_path, 'r', encoding='utf-8') as csv\_file:

    csv\_reader = csv.DictReader(csv\_file)

    for row in csv\_reader:

        # Truy xuất thông tin birthday từ dòng CSV và tính tuổi

        birthday = row['birthday']

        age = calculate\_age(birthday)

        age\_list.append(age)

# Ghi danh sách tuổi vào tệp CSV mới

with open(output\_csv\_file\_path, 'w', newline='', encoding='utf-8') as csv\_file:

    csv\_writer = csv.writer(csv\_file)

    csv\_writer.writerow(['age'])  # Ghi tiêu đề cho cột tuổi

    for age in age\_list:

        csv\_writer.writerow([age])

print(f'Dữ liệu tuổi đã được ghi vào tệp {output\_csv\_file\_path}')

Dữ liệu tuổi đã được ghi vào tệp /content/drive/MyDrive/project\_facebook\_bigdata/fresh\_data\_clean/user\_age.csv

import csv

csv\_file\_path = '/content/drive/MyDrive/project\_facebook\_bigdata/all\_data\_file/user\_data.csv'

output\_csv\_file\_path = '/content/drive/MyDrive/project\_facebook\_bigdata/fresh\_data\_clean/user\_gender.csv'

# Khởi tạo danh sách trống để lưu trữ thông tin giới tính

gender\_list = []

# Mở tệp CSV và đọc nội dung

with open(csv\_file\_path, 'r', encoding='utf-8') as csv\_file:

    csv\_reader = csv.DictReader(csv\_file)

    for row in csv\_reader:

        # Trích xuất giá trị giới tính từ mỗi hàng và kiểm tra

        gender = row['gender']

        if gender == 'male' or gender == 'female':

            gender\_list.append(gender)

        else:

            gender\_list.append('Không xác định')

# Ghi danh sách giới tính vào tệp CSV mới

with open(output\_csv\_file\_path, 'w', newline='', encoding='utf-8') as csv\_file:

    csv\_writer = csv.writer(csv\_file)

    csv\_writer.writerow(['gender'])  # Ghi tiêu đề cho cột giới tính

    for gender in gender\_list:

        csv\_writer.writerow([gender])

print(f'Dữ liệu giới tính đã được ghi vào tệp {output\_csv\_file\_path}')

Dữ liệu giới tính đã được ghi vào tệp /content/drive/MyDrive/project\_facebook\_bigdata/fresh\_data\_clean/user\_gender.csv

import csv

import json

csv\_file\_path = '/content/drive/MyDrive/project\_facebook\_bigdata/all\_data\_file/user\_data.csv'

output\_csv\_file\_path = '/content/drive/MyDrive/project\_facebook\_bigdata/fresh\_data\_clean/user\_location.csv'

# Mở tệp user\_data.csv

with open(csv\_file\_path, 'r') as file:

    # Đọc nội dung của tệp CSV

    csv\_reader = csv.DictReader(file)

    # Khởi tạo danh sách để lưu các giá trị 'name' từ cột 'location'

    names\_in\_location = []

    # Lặp qua từng dòng trong tệp CSV

    for row in csv\_reader:

        # Kiểm tra xem chuỗi 'location' có đúng định dạng JSON hay không

        try:

            location\_data = json.loads(row['location'].replace("'", "\""))

            if 'name' in location\_data:

                names\_in\_location.append(location\_data['name'])

            else:

                names\_in\_location.append("không xác định")

        except json.JSONDecodeError:

            names\_in\_location.append("không xác định")

# Ghi danh sách giá trị 'name' vào tệp CSV mới

with open(output\_csv\_file\_path, 'w', newline='', encoding='utf-8') as csv\_file:

    csv\_writer = csv.writer(csv\_file)

    csv\_writer.writerow(['Tên'])

    for name in names\_in\_location:

        csv\_writer.writerow([name])

print(f'Dữ liệu đã được ghi vào tệp ',output\_csv\_file\_path)

Dữ liệu đã được ghi vào tệp /content/drive/MyDrive/project\_facebook\_bigdata/fresh\_data\_clean/user\_location.csv

Vậy là vừa rồi chúng ta đã thu được dữ liệu tuổi, giới tính và nơi ở của ngưởi sử comments bài viết facebook.

5.5. Phân tích reactions – biểu diễn biểu đồ

from google.colab import drive

drive.mount("/content/drive")

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import numpy as np

import json

# Đường dẫn đến tệp JSON

json\_file\_path = "/content/drive/MyDrive/project\_facebook\_bigdata/all\_data\_file/reactions\_data.json"

# Đọc dữ liệu từ tệp JSON

with open(json\_file\_path, "r") as json\_file:

    data = json.load(json\_file)

print(data)

import json

# Đường dẫn đến tệp JSON

json\_file\_path = "/content/drive/MyDrive/project\_facebook\_bigdata/all\_data\_file/reactions\_data.json"

# Đọc dữ liệu từ tệp JSON

with open(json\_file\_path, "r") as json\_file:

    data = json.load(json\_file)

print(data)

>> {'reactions': {'thích': 2815, 'yêu thích': 186, 'haha': 649, 'wow': 17, 'thương thương': 201, 'buồn': 9, 'phẫn nộ': 1}, 'reaction\_count': 3878, 'comments': 1521, 'shares': 188}

import matplotlib.pyplot as plt

data = {

    'reactions': {

        'thích': 2815,

        'yêu thích': 186,

        'haha': 649,

        'wow': 17,

        'thương thương': 201,

        'buồn': 9,

        'phẫn nộ': 1

    }

}

# Trích xuất các loại phản ứng và số lượng tương ứng từ dữ liệu

reactions = data['reactions']

labels = list(reactions.keys())

values = list(reactions.values())

# Vẽ biểu đồ tròn

plt.figure(figsize=(8, 8))

plt.pie(values, labels=None, autopct='%1.1f%%', startangle=140)

plt.title('Biểu đồ phản ứng trên bài viết')

plt.axis('equal')  # Đảm bảo biểu đồ tròn

# Tạo bảng chú thích riêng và điều chỉnh vị trí

legend\_labels = [f'{label}: {value}' for label, value in zip(labels, values)]

legend = plt.legend(legend\_labels, loc='center left', bbox\_to\_anchor=(1, 0.5))

# Hiển thị biểu đồ

plt.show()

>>

A pie chart with different colored circles

Description automatically generated

5.6. Phân tích người dùng bình luận – biểu diễn biểu đồ

- Tạo một phiên làm việc google colab mới

from google.colab import drive

drive.mount("/content/drive")

import csv

from datetime import datetime

age\_path = "/content/drive/MyDrive/project\_facebook\_bigdata/fresh\_data\_clean/user\_age.csv"

gender\_path = "/content/drive/MyDrive/project\_facebook\_bigdata/fresh\_data\_clean/user\_gender.csv"

location\_path = "/content/drive/MyDrive/project\_facebook\_bigdata/fresh\_data\_clean/user\_location.csv"

- Tiếp theo đây chúng ta sẽ phân tích dữ liệu và biểu thị kết quả cho giới tính người comments bài viết

# Mở tệp CSV và đọc nội dung

gender\_list = []

with open(gender\_path, 'r', encoding='utf-8') as csv\_file:

    csv\_reader = csv.DictReader(csv\_file)

    for row in csv\_reader:

        # Trích xuất giá trị giới tính từ mỗi hàng và kiểm tra

        gender = row['gender']

        if gender == 'male' or gender == 'female':

            gender\_list.append(gender)

        else:

            gender\_list.append('Không xác định')

total\_gender\_count = len(gender\_list)

male\_count = gender\_list.count('male')

female\_count = gender\_list.count('female')

undefined\_count = gender\_list.count('Không xác định')

import matplotlib.pyplot as plt

# Dữ liệu về phân phối giới tính

gender\_labels = ['Male', 'Female', 'Undefined']

gender\_counts = [male\_count, female\_count, undefined\_count]

# Màu sắc cho các phần trong biểu đồ

colors = ['blue', 'pink', 'gray']

# Vẽ biểu đồ tròn

plt.figure(figsize=(6, 6))

plt.pie(gender\_counts, labels=gender\_labels, colors=colors, autopct='%1.1f%%', startangle=140)

plt.title('Phân Phối Giới Tính')

plt.axis('equal')  # Đảm bảo biểu đồ tròn

plt.show()

>>

A blue and pink pie chart

Description automatically generated

- Phân tích và biểu diễn kết quả của độ tuổi những người bình luận bài viết

age\_list = []

# Mở tệp CSV và đọc nội dung

with open(age\_path, 'r', encoding='utf-8') as csv\_file:

    csv\_reader = csv.DictReader(csv\_file)  # Sử dụng DictReader để đọc dữ liệu dưới dạng từ điển

    for row in csv\_reader:

        age = row.get("age")  # Trường "age" trong mỗi hàng

        if age:  # Kiểm tra xem trường "age" có giá trị không

            age\_list.append(age)

# Chia vùng tuổi

under\_18\_count = 0

age\_18\_to\_29\_count = 0

age\_30\_to\_39\_count = 0

age\_40\_to\_49\_count = 0

age\_50\_to\_59\_count = 0

over\_60\_count = 0

undefined\_count = 0

for age\_str in age\_list:

    try:

        age = int(age\_str)  # Chuyển đổi thành số nguyên

        if age < 18:

            under\_18\_count += 1

        elif 18 <= age < 30:

            age\_18\_to\_29\_count += 1

        elif 30 <= age < 40:

            age\_30\_to\_39\_count += 1

        elif 40 <= age < 50:

            age\_40\_to\_49\_count += 1

        elif 50 <= age < 60:

            age\_50\_to\_59\_count += 1

        else:

            over\_60\_count += 1

    except ValueError:

        undefined\_count += 1

import matplotlib.pyplot as plt

# Dữ liệu về phân phối độ tuổi

age\_labels = ['Dưới 18', '18-29', '30-39', '40-49', '50-59', 'Trên 60', 'Không xác định']

age\_counts = [under\_18\_count, age\_18\_to\_29\_count, age\_30\_to\_39\_count, age\_40\_to\_49\_count, age\_50\_to\_59\_count, over\_60\_count, undefined\_count]

# Vẽ biểu đồ đường

plt.figure(figsize=(10, 6))

plt.plot(age\_labels, age\_counts, marker='o', linestyle='-', color='b', markersize=8, linewidth=2)

plt.xlabel('Độ Tuổi')

plt.ylabel('Số Lượng')

plt.title('Phân Phối Độ Tuổi (Biểu đồ Đường)')

plt.xticks(rotation=45, ha='right')  # Xoay nhãn trục x để dễ đọc

plt.grid(True, linestyle='--', alpha=0.6)

plt.show()

A graph with a line

Description automatically generated

- Phân tích và biểu diễn top 10 thành phố xuất hiện nhiều nhất có trong danh sách nơi ở của người comments

A graph with blue bars

Description automatically generated

5.5. Phân tích dữ liệu comments text – Mô hình Naive Bayes

- Trong phần này chúng ta sẽ có 2 chương trình như sau:

* Chương trình 1: tập trung vào việc huấn luyện một mô hình phân loại Naive Bayes từ dữ liệu huấn luyện và sau đó sử dụng mô hình này để dự đoán nhãn cho các bình luận mới.
* Chương trình thứ hai: đánh giá hiệu suất của mô hình sử dụng các chỉ số đánh giá như độ chính xác, precision, recall và F1-score, trong khi chương trình đầu tiên chỉ thực hiện dự đoán nhãn cho bình luận mới mà không đánh giá mô hình trên tập kiểm tra.

Chương trình 1:

Để làm được chương trình này đầu tiên ta phải xử lý dữ liệu, gán nhãn thủ công cho các comment. Ta chia làm 3 thang điểm 1,2 và 3.

* 1 – đại diện cho bình luận tích cực
* 2 – bình luận trung lập
* 3 – bình luận tiêu cực

A screenshot of a computer

Description automatically generated

Sau khi gán nhãn xong chúng ta sẽ up file lên google drive kết nối với colab để xử lý công việc tiếp theo,..

Chương trình chạy như sau:

from google.colab import drive

drive.mount("/content/drive")

import json

import csv

!pip install unidecode

import json

import unidecode

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.naive\_bayes import MultinomialNB

# Đọc dữ liệu từ tệp JSON

with open('/content/drive/MyDrive/project\_facebook\_bigdata/fresh\_data\_clean/messgae\_commnet.json', 'r', encoding='utf-8') as json\_file:

    data = json.load(json\_file)

# Chuẩn bị dữ liệu

comments = []

labels = []

for entry in data:

    if "Message" in entry and "Labels" in entry and entry["Labels"] is not None:

        comments.append(entry["Message"])

        labels.append(int(entry["Labels"]))

# Tiền xử lý dữ liệu

def preprocess(text):

    if text is not None and len(text) > 0:

        text = unidecode.unidecode(text)

        text = ''.join(e for e in text if (e.isalpha() or e.isspace()))

        text = text.lower()

        return text

    else:

        return ""

preprocessed\_comments = [preprocess(comment) for comment in comments]

# Biểu diễn số hóa dữ liệu sử dụng TF-IDF

vectorizer = TfidfVectorizer()

X = vectorizer.fit\_transform(preprocessed\_comments)

# Xây dựng mô hình Naive Bayes

clf = MultinomialNB()

clf.fit(X, labels)

# Dự đoán nhãn cho các bình luận mới

new\_comments = ["Bình luận Tích cực","Bình luận Trung lập","Bình luận Tiêu cực"]

preprocessed\_new\_comments = [preprocess(comment) for comment in new\_comments]

X\_new = vectorizer.transform(preprocessed\_new\_comments)

predicted\_labels = clf.predict(X\_new)

print("Predicted Labels for New Comments:", predicted\_labels)

>>

>> Predicted Labels for New Comments: [1 1 1]

Chương trình 2:

from sklearn.metrics import accuracy\_score, precision\_score, recall\_score, f1\_score, confusion\_matrix

from sklearn.model\_selection import train\_test\_split

# Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, labels, test\_size=0.2, random\_state=42)

# Train the Multinomial Naive Bayes classifier on the training data

clf = MultinomialNB()

clf.fit(X\_train, y\_train)

# Predict labels on the test data

y\_pred = clf.predict(X\_test)

# Evaluate the model

accuracy = accuracy\_score(y\_test, y\_pred)

precision = precision\_score(y\_test, y\_pred, average=None)  # Use average=None for multiclass

recall = recall\_score(y\_test, y\_pred, average=None)  # Use average=None for multiclass

f1 = f1\_score(y\_test, y\_pred, average=None)  # Use average=None for multiclass

confusion = confusion\_matrix(y\_test, y\_pred)

print("Accuracy:", accuracy)

print("Precision:", precision)

print("Recall:", recall)

print("F1-Score:", f1)

print("Confusion Matrix:\n", confusion)

>> Accuracy: 0.5672514619883041

Precision: [0.54744526 0.63636364 1. ]

Recall: [0.92592593 0.33870968 0.03571429]

F1-Score: [0.68807339 0.44210526 0.06896552]

Confusion Matrix:

[[75 6 0]

[41 21 0]

[21 6 1]]

### 6. Đánh giá thực nghiệm

- Việc gán nhãn thủ công tại bước phân tích comments text với mô hình Naive Bayes còn tốn thời gian.

- Dữ liệu về nơi ỏ của người comment chưa thật sự được làm sạch khi vẫn trùng lặp nhau về địa điểm.

- Google Colab hạn chế kết : chỉ có phép tối đa 3 phiên làm việc kết nối vào google driver cùng một lúc

- Chương trình phải thực hiện tuần tự, không linh động

7. Tài liệu tham khảo

<https://github.com/kevinzg/facebook-scraper>

<https://github.com/18520339/facebook-data-extraction>

<https://github.com/huyinit/Craw-Data-Facebook-Easy>