**TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP**

**KHOA ĐIỆN TỬ**

**Bộ môn: Công nghệ Thông tin**.

A circular logo with a star and a star in the middle

AI-generated content may be incorrect.

**BÀI TẬP KẾT THÚC MÔN HỌC**

MÔN HỌC

**KHOA HỌC DỮ LIỆU**

Sinh viên: **Đỗ Đình Đức**

MSSV: **K215480106013**

Lớp: **K57KMT**

Giáo viên giảng dạy: **TS. Nguyễn Văn Huy**

Link github: https://github.com/DucK215480106013/BTL\_KHDL

**Thái Nguyên – 2025**

|  |  |
| --- | --- |
| **TRƯỜNG ĐHKTCN** | **CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM** |
| **KHOA ĐIỆN TỬ** | ***Độc lập - Tự do - Hạnh phúc*** |

**BÀI TẬP KẾT THÚC MÔN HỌC**

**MÔN HỌC: KHOA HỌC DỮ LIỆU**

BỘ MÔN : CÔNG NGHỆ THÔNG TIN

Sinh viên: **Đỗ Đình Đức**

Mssv: **K215480106013**

Lớp: **K57KMT** Ngành: **Kỹ thuật máy tính**

Giáo viên hướng dẫn: **TS. Nguyễn Văn Huy**

Ngày giao đề …….**20/5/2025**……. Ngày hoàn thành ….**30/5/2025**….

Tên đề tài : Phân tích comment Facebook để xác định cảm xúc tích cực, tiêu cực

Yêu cầu :

**Đầu vào**: Dữ liệu comment từ Facebook Comments Sentiment - Kaggle

**Đầu ra**: Thống kê tỷ lệ cảm xúc, visualization bằng Pie Chart. Các tính năng: Phân tích ngôn ngữ tự nhiên (CountVectorizer, TF-IDF) Classification (Naive Bayes, Logistic Regression) Trực quan hóa dữ liệu cảm xúc

**Hướng dẫn:** Xử lý văn bản (Pandas và Scikit-Learn). Huấn luyện mô hình sentiment classification. Hiển thị biểu đồ Pie Chart về tỷ lệ cảm xúc.

|  |
| --- |
| **GIÁO VIÊN HƯỚNG DẪN** |
| *(Ký và ghi rõ họ tên)* |

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

### Thái Nguyên, ngày…tháng…năm 20....

## GIÁO VIÊN HƯỚNG DẪN

*(Ký ghi rõ họ tên)*

Mục Lục

[LỜI CẢM ƠN 5](#_Toc199501844)

[MỤC LỤC 6](#_Toc199501845)

[DANH MỤC HÌNH VẼ 6](#_Toc199501846)

[LỜI NÓI ĐẦU 7](#_Toc199501847)

[CHƯƠNG I. GIỚI THIỆU ĐỀ TÀI 1](#_Toc199501848)

[CHƯƠNG II. CƠ SỞ LÝ THUYẾT 3](#_Toc199501849)

[2.1. Thư viện Pandas 3](#_Toc199501850)

[2.2. Thư viện Matplotlib 4](#_Toc199501851)

[2.3. Kỹ thuật xử lý ngôn ngữ tự nhiên 6](#_Toc199501852)

[2.3.1. CountVectorizer 6](#_Toc199501853)

[2.3.2. TF-IDF 7](#_Toc199501854)

[2.4. Mô hình phân loại 9](#_Toc199501855)

[2.4.1. Mô hình Naive Bayes 9](#_Toc199501856)

[2.4.2. Logistic Regression 10](#_Toc199501857)

[CHƯƠNG III. THIẾT KẾ VÀ XÂY DỰNG CHƯƠNG TRÌNH 11](#_Toc199501858)

[**3.1.** **Sơ đồ khối hệ thống** 11](#_Toc199501859)

[**3.2.** **Sơ đồ thuật toán** 13](#_Toc199501860)

[3.2.1. Huấn luyện mô hình học máy 13](#_Toc199501861)

[3.2.2. Lấy dữ liệu người dùng nhập và xử lý 14](#_Toc199501862)

[**Hình 12. Sơ đồ khối chương trình lấy dữ liệu người dùng nhập và xử lý** 14](#_Toc199501863)

[3.2.3. Ứng dụng Web để sử dụng 15](#_Toc199501864)

[15](#_Toc199501865)

[**3.3.** **Xây dựng chương trình** 16](#_Toc199501866)

[3.3.1. Thu thập và xử lý dữ liệu. 16](#_Toc199501867)

[3.3.2. Vecto hoá văn bản 17](#_Toc199501868)

[3.3.3. Mô hình học máy 19](#_Toc199501869)

[**3.4.** **Xây dựng bảng dữ liệu** 22](#_Toc199501870)

[CHƯƠNG IV. THỰC NGHIỆM VÀ KẾT LUẬN 23](#_Toc199501871)

[4.1. Thực nghiệm 23](#_Toc199501872)

[Lựa chọn mô hình học máy 23](#_Toc199501873)

# LỜI CẢM ƠN

Em xin chân thành cảm ơn thầy **TS. Nguyễn Văn Huy** đã tận tình hướng dẫn và hỗ trợ em trong suốt quá trình thực hiện bài tập lớn này. Bên cạnh đó, em xin cảm ơn thầy cô và các bạn đã đóng góp ý kiến để em hoàn thiện sản phẩm. Em hy vọng kết quả đạt được sẽ đáp ứng yêu cầu của bài tập và mang lại trải nghiệm thú vị cho người dùng.

# MỤC LỤC

# DANH MỤC HÌNH VẼ

 **Hình 1:** Sơ đồ khối hệ thống phân tích và giám sát bình luận trên Facebook.

 **Hình 2:** Mô tả cách hoạt động của CountVectorizer.

 **Hình 3:** Mô tả cách hoạt động của TF-IDF.

 **Hình 4:** Mô tả mô hình Naive Bayes.

 **Hình 5:** Mô tả mô hình Logistic Regression.

 **Hình 6:** Sơ đồ khối chương trình huấn luyện mô hình học máy.

 **Hình 7:** Sơ đồ khối chương trình lấy dữ liệu người dùng và xử lý.

 **Hình 8:** Sơ đồ khối chương trình Web mô phỏng.

 **Hình 9:** Phân bổ dữ liệu trong tập train.

 **Hình 10:** Biểu đồ Pie Chart thống kê cảm xúc bình luận.

# LỜI NÓI ĐẦU

Trong thời đại số, mạng xã hội ngày càng đóng vai trò quan trọng trong giao tiếp, truyền thông và quản lý hình ảnh của các tổ chức, doanh nghiệp. Tuy nhiên, việc kiểm soát và xử lý bình luận trên các nền tảng như Facebook vẫn còn nhiều thách thức. Đề tài "Phân tích comment Facebook để xác định cảm xúc tích cực, tiêu cực" được thực hiện nhằm ứng dụng các kỹ thuật xử lý ngôn ngữ tự nhiên (NLP) và học máy để xây dựng một hệ thống phân tích, lọc và giám sát bình luận theo thời gian thực.

Bài tập này giúp sinh viên rèn luyện kỹ năng xử lý dữ liệu văn bản, xây dựng và huấn luyện mô hình học máy, trực quan hóa kết quả, đồng thời áp dụng kiến thức lập trình Python và các thư viện Pandas, Scikit-Learn, Matplotlib để giải quyết vấn đề thực tế. Hệ thống được phát triển hướng đến việc hỗ trợ các Fanpage Facebook tự động nhận diện và xử lý bình luận không phù hợp, góp phần nâng cao chất lượng nội dung và trải nghiệm người dùng.

Trong quá trình thực hiện, mặc dù đã cố gắng hoàn thiện bài tập, nhưng vẫn không tránh khỏi những thiếu sót. Kính mong thầy cô và các bạn đóng góp ý kiến để nhóm có thể hoàn thiện và phát triển sản phẩm tốt hơn.

# CHƯƠNG I. GIỚI THIỆU ĐỀ TÀI

Trong kỷ nguyên số, mạng xã hội đã trở thành một phần không thể tách rời trong chiến lược truyền thông của các doanh nghiệp, tổ chức và cá nhân. Đặc biệt, các Fanpage trên Facebook đóng vai trò như một cầu nối trực tiếp giữa thương hiệu và cộng đồng người dùng. Tuy nhiên, sự gia tăng nhanh chóng của tương tác cũng kéo theo thách thức lớn trong việc kiểm soát nội dung, khi nhiều bình luận có thể mang tính tiêu cực, công kích hoặc không phù hợp.

Việc theo dõi, sàng lọc và phản hồi hàng loạt comment thủ công không chỉ tốn thời gian mà còn khó đảm bảo tính nhất quán và kịp thời. Do đó, nhu cầu ứng dụng công nghệ để tự động phân tích cảm xúc và xử lý bình luận. Đề tài này hướng đến việc xây dựng một hệ thống thông minh, có khả năng nhận diện và lọc bỏ comment tiêu cực.

A screenshot of a social media post

AI-generated content may be incorrect.

Đề tài *"* *Phân tích comment Facebook để xác định cảm xúc tích cực, tiêu cực."* tập trung vào việc phân tích comment để xác định cảm xúc tích cực hoặc tiêu cực nhằm hỗ trợ tự động xử lý các bình luận không phù hợp hoặc tiêu cực. Dữ liệu đầu vào được sử dụng từ bộ dữ liệu Tập dữ liệu comment được thu thập trên Facebook, chứa các bình luận thực tế kèm nhãn cảm xúc.

Ứng dụng được thiết kế nhằm thực hiện phân tích cảm xúc các bình luận trên Fanpage Facebook theo thời gian thực, kết hợp nhiều kỹ thuật xử lý ngôn ngữ tự nhiên và học máy để mang lại hiệu quả cao nhất. Đầu tiên, dữ liệu văn bản từ các comment sẽ được tiền xử lý thông qua hai phương pháp phổ biến là CountVectorizer và TF-IDF, giúp chuyển đổi các câu chữ thành dạng số đặc trưng dễ dàng tiếp nhận bởi các mô hình học máy. Tiếp theo, ứng dụng áp dụng các thuật toán phân loại như Naive Bayes và Logistic Regression để huấn luyện và dự đoán cảm xúc tích cực hoặc tiêu cực cho từng bình luận. Kết quả phân loại sau đó được tổng hợp và trực quan hóa dưới dạng biểu đồ Pie Chart sinh động.

A cartoon of a group of men running away from a person

AI-generated content may be incorrect.

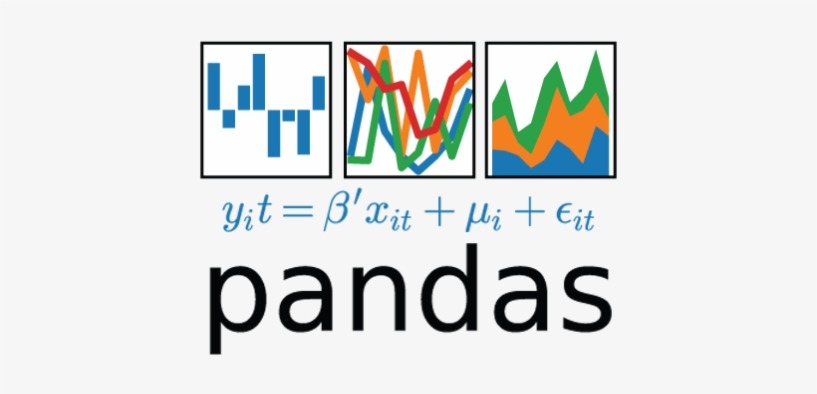
Đối với các Fanpage Facebook – vốn là kênh truyền thông chính của nhiều tổ chức, việc quản lý bình luận hiệu quả đóng vai trò then chốt trong xây dựng và duy trì hình ảnh. Trong bối cảnh đó, việc ứng dụng công nghệ học máy và xử lý ngôn ngữ tự nhiên để phân tích cảm xúc, phát hiện và xử lý comment tiêu cực một cách tự động không chỉ là hướng đi khả thi mà còn là giải pháp cấp thiết nhằm đảm bảo chất lượng nội dung và nâng cao trải nghiệm người dùng.

Đề tài này ra đời với mục tiêu xây dựng một hệ thống giám sát và lọc bình luận tiêu cực theo thời gian thực trên Facebook, đánh giá và kiểm soát nội dung một cách hiệu quả.

# CHƯƠNG II. CƠ SỞ LÝ THUYẾT

## 2.1. Thư viện Pandas

Pandas là một thư viện Python mạnh mẽ dùng để xử lý và phân tích dữ liệu dạng bảng. Thư viện này cung cấp hai cấu trúc dữ liệu chính là Series (dữ liệu một chiều) và DataFrame (dữ liệu hai chiều gồm hàng và cột), giúp thao tác với dữ liệu trở nên đơn giản và hiệu quả. Với Pandas, người dùng có thể dễ dàng đọc dữ liệu từ các tệp CSV, Excel, cơ sở dữ liệu SQL, sau đó thực hiện các thao tác như lọc, sắp xếp, nhóm, tính toán thống kê và làm sạch dữ liệu. Thư viện này đặc biệt hữu ích trong lĩnh vực khoa học dữ liệu và học máy, nhờ khả năng kết hợp tốt với các thư viện khác như NumPy, Matplotlib và Scikit-learn. Nhờ tính linh hoạt và hiệu quả, Pandas là công cụ không thể thiếu trong quá trình tiền xử lý và phân tích dữ liệu.



1. **Cấu trúc dữ liệu chính trong Pandas**

* Series: Một mảng một chiều tương tự như một cột trong bảng.
* DataFrame: Một bảng hai chiều với hàng và cột, tương tự như bảng tính hoặc SQL table.

1. **Tạo Series và DataFrame**

Pandas cho phép tạo Series và DataFrame từ nhiều nguồn dữ liệu khác nhau như list, dictionary, NumPy array hoặc từ file CSV, Excel.

1. **Đọc và ghi dữ liệu với Pandas**

* Đọc file CSV: pd.read\_csv('tenfile.csv')
* Ghi DataFrame ra CSV: df.to\_csv('tenfile.csv', index=False)
* Hỗ trợ nhiều định dạng khác: Excel, JSON, SQL...

1. **Một số thao tác dữ liệu cơ bản**

* df.head(n): Xem n dòng đầu tiên
* df.tail(n): Xem n dòng cuối cùng
* df.info(): Thông tin tổng quan
* df.describe(): Thống kê dữ liệu số
* df['cot']: Truy cập một cột
* df.loc[index]: Truy cập theo chỉ số dòng

df.iloc[i]: Truy cập theo vị trí dòng

1. **Lọc và xử lý dữ liệu**

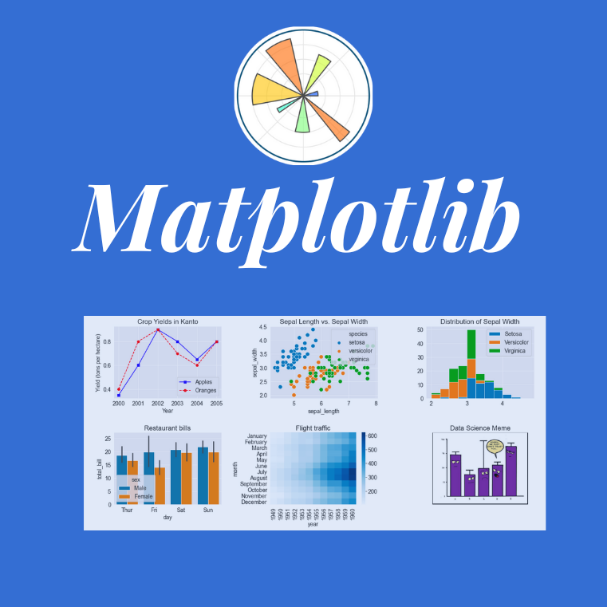
* Lọc theo điều kiện: df[df['Age'] > 20]
* Xử lý thiếu dữ liệu: df.dropna(), df.fillna(value)
* Thêm/xoá cột: df['new'] = ..., df.drop('col', axis=1)

Pandas là công cụ cực kỳ mạnh mẽ và linh hoạt cho xử lý dữ liệu trong Python. Với cú pháp dễ học và khả năng tích hợp tốt với các thư viện khác, Pandas là lựa chọn hàng đầu trong các dự án phân tích dữ liệu và học máy.

Nếu bạn muốn, mình có thể làm ví dụ minh họa thực tế theo đề tài của bạn (ví dụ: đọc file comment, phân tích, lọc comment tiêu cực với Pandas).

## 2.2. Thư viện Matplotlib

**Matplotlib** là một thư viện vẽ đồ thị nổi tiếng và mạnh mẽ trong Python, dùng để trực quan hóa dữ liệu bằng cách tạo các biểu đồ, đồ thị và hình ảnh minh họa một cách linh hoạt. Thư viện này được phát triển với mục tiêu hỗ trợ các nhà khoa học, nhà phân tích và lập trình viên dễ dàng trình bày và hiểu dữ liệu thông qua hình ảnh.



**Hình 4. Thư viện Matpotlib**

Matplotlib cung cấp một giao diện rất giống với MATLAB, cho phép tạo các biểu đồ như: biểu đồ đường (line plot), biểu đồ cột (bar chart), biểu đồ tròn (pie chart), biểu đồ tán xạ (scatter plot), biểu đồ histogram (histogram), v.v. Thư viện hoạt động hiệu quả với NumPy và Pandas – hai công cụ phổ biến để xử lý dữ liệu trong Python. Trong Matplotlib, mô-đun quan trọng nhất là pyplot, thường được import với tên viết tắt là plt. Mô-đun này cho phép người dùng gọi các hàm đơn giản để tạo biểu đồ, đặt tiêu đề, ghi nhãn trục, thêm chú thích, vẽ lưới, điều chỉnh màu sắc, kiểu đường, độ dày, và nhiều tuỳ chỉnh khác.

* Cách vẽ một số loại biểu đồ
* Biểu đồ đường (Line Chart)

|  |
| --- |
| x = [1, 2, 3, 4, 5]  y = [2, 4, 6, 8, 10]  plt.plot(x, y, label="y = 2x", color='blue', linestyle='--', marker='o')  plt.title("Biểu đồ đường")  plt.xlabel("Trục X")  plt.ylabel("Trục Y")  plt.legend()  plt.grid(True)  plt.show() |
| plot(): vẽ biểu đồ đường.  label: nhãn cho đường biểu diễn.  color, linestyle, marker: điều chỉnh màu, kiểu đường, ký hiệu điểm.  title(), xlabel(), ylabel(): tiêu đề và tên trục.  legend(): hiển thị chú thích.  grid(): hiển thị lưới.  show(): hiển thị biểu đồ. |

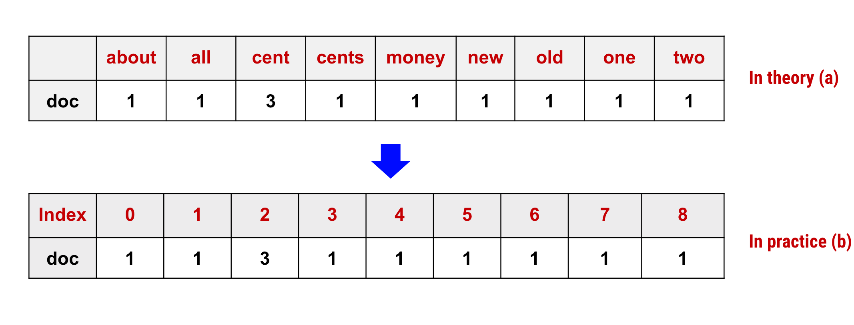
* Biểu đồ tròn (Pie Chart)

|  |
| --- |
| Cách vẽ biểu đồ tròn  categories = ['A', 'B', 'C']  values = [10, 20, 15]  plt.bar(categories, values, color='orange')  plt.title("Biểu đồ cột")  plt.show() |
| pie(): vẽ biểu đồ tròn.  autopct: hiển thị phần trăm.  startangle: xoay góc bắt đầu.  axis('equal'): đảm bảo hình tròn không bị méo. |

## 2.3. Kỹ thuật xử lý ngôn ngữ tự nhiên

### 2.3.1. CountVectorizer

CountVectorizer là một kỹ thuật trong xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP) dùng để chuyển đổi văn bản thành dạng số (vector) nhằm phục vụ cho các thuật toán học máy. Cụ thể, nó thực hiện việc đếm tần suất xuất hiện của từng từ trong tập văn bản và biểu diễn mỗi văn bản dưới dạng một vector đặc trưng.

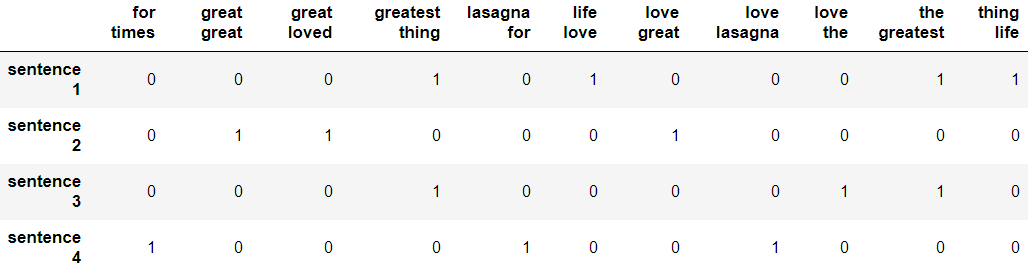


**Hình 5. Mô tả cách hoạt động CountVectorizer**

* Ví dụ về cách hoạt động của CountVectorizer

|  |
| --- |
| Dữ liệu như sau : |
| 1. Học máy là một môn học  2. Tôi rất thích học máy  3. Tôi không thích học |

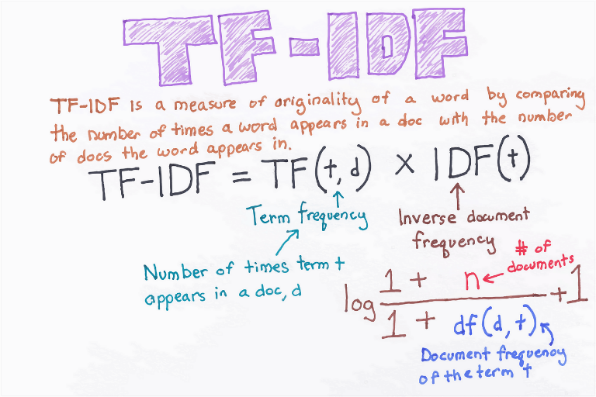
1. **Tách từ** → tạo danh sách từ vựng (vocabulary):  
   → ['tôi', 'thích', 'học', 'máy', 'rất', 'thú vị', 'không']
2. **Đếm số lần xuất hiện của từng từ** trong mỗi câu.
3. **Chuyển thành vector**:



CountVectorizer là một kỹ thuật xử lý ngôn ngữ tự nhiên đơn giản và hiệu quả trong việc chuyển đổi văn bản thành dạng số, giúp các mô hình học máy dễ dàng xử lý dữ liệu ngôn ngữ. Ưu điểm lớn của CountVectorizer là dễ sử dụng, nhanh chóng và cho kết quả tốt đối với các bài toán phân loại văn bản cơ bản. Tuy nhiên, nó cũng có một số hạn chế như không thể nắm bắt được ngữ cảnh hoặc ý nghĩa sâu xa của từ ngữ, đồng thời tạo ra các ma trận đặc trưng có kích thước rất lớn khi xử lý dữ liệu có nhiều từ khác nhau, gây tốn bộ nhớ và thời gian tính toán. Vì vậy, CountVectorizer thường được kết hợp với các kỹ thuật nâng cao hơn để cải thiện hiệu suất và độ chính xác của mô hình.

### 2.3.2. TF-IDF

TF-IDF (Term Frequency-Inverse Document Frequency) là một kỹ thuật phổ biến trong xử lý ngôn ngữ tự nhiên dùng để đánh giá mức độ quan trọng của một từ trong một tài liệu hoặc tập hợp tài liệu. TF (Term Frequency) đo tần suất xuất hiện của từ trong một văn bản cụ thể, trong khi IDF (Inverse Document Frequency) đo mức độ hiếm hoặc phổ biến của từ đó trong toàn bộ tập tài liệu. Bằng cách kết hợp TF và IDF, TF-IDF giúp giảm trọng số của những từ quá phổ biến, ít mang thông tin phân biệt (như các từ nối, từ dừng), đồng thời tăng trọng số của những từ quan trọng, đặc trưng cho nội dung văn bản. Kỹ thuật này giúp mô hình học máy tập trung vào các đặc trưng có ý nghĩa hơn, cải thiện hiệu quả phân loại và tìm kiếm thông tin so với phương pháp đếm từ thuần túy như CountVectorizer. Tuy vậy, TF-IDF vẫn chưa thể hiểu được ngữ cảnh hay mối quan hệ ngữ nghĩa giữa các từ, nên thường được kết hợp với các kỹ thuật xử lý ngôn ngữ sâu hơn.



**Hình 6. Mô tả cách hoạt động TF- IDF**

* **Term Frequency (TF):** Đây là tần suất xuất hiện của một từ trong một tài liệu. TF đo lường số lần từ đó xuất hiện trong tài liệu so với tổng số từ trong tài liệu đó. Công thức tính TF cho một từ ttt trong tài liệu ddd như sau:

A black text on a white background

AI-generated content may be incorrect.

* **Inverse Document Frequency (IDF):** Đây là yếu tố phản ánh mức độ quan trọng của từ trong toàn bộ tập hợp tài liệu. Các từ xuất hiện nhiều trong nhiều tài liệu sẽ có IDF thấp, vì chúng không giúp phân biệt tài liệu này với tài liệu khác. Công thức tính IDF cho từ ttt trong tập tài liệu gồm N tài liệu và nt​ là số tài liệu chứa từ t:

A mathematical equation with numbers and symbols

AI-generated content may be incorrect.

* Trong đó, việc cộng thêm 1 vào mẫu số nhằm tránh trường hợp chia cho 0 khi từ không xuất hiện trong bất kỳ tài liệu nào.
* Công thức TF-IDF: Khi kết hợp TF và IDF, ta có chỉ số TF-IDF thể hiện trọng số của từ ttt trong tài liệu ddd:

A mathematical equation with black text

AI-generated content may be incorrect.

* Ý nghĩa của TF-IDF:Từ có tần suất cao trong một tài liệu nhưng xuất hiện ít trong các tài liệu khác sẽ có giá trị TF-IDF lớn, thể hiện nó mang nhiều thông tin đặc trưng cho tài liệu đó. Ngược lại, các từ phổ biến như “và”, “là”, “của” sẽ có giá trị TF-IDF thấp do xuất hiện ở hầu hết các tài liệu, không giúp phân biệt nội dung.

## 2.4. Mô hình phân loại

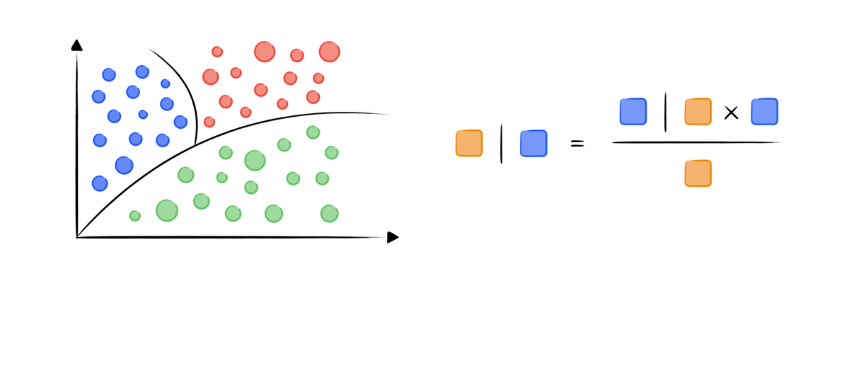
### 2.4.1. Mô hình Naive Bayes

Naive Bayes là một thuật toán phân loại dựa trên định lý Bayes trong xác suất thống kê, thường được sử dụng trong các bài toán phân loại văn bản và xử lý ngôn ngữ tự nhiên. Thuật toán này giả định rằng các đặc trưng (features) đầu vào độc lập với nhau – nghĩa là sự xuất hiện của một đặc trưng không ảnh hưởng đến sự xuất hiện của đặc trưng khác, dù trong thực tế điều này không luôn đúng.

Công thức cơ bản của Naive Bayes dựa trên định lý Bayes:

A white paper with black text

AI-generated content may be incorrect.



**Hình 7. Mô tả mô hình Naive Bayes**

Do P(X) là hằng số đối với tất cả các lớp, Naive Bayes sẽ tính giá trị P(X∣C)×P(C) cho mỗi lớp rồi chọn lớp có giá trị lớn nhất làm dự đoán. Ưu điểm của Naive Bayes là nhanh chóng, đơn giản và hiệu quả với dữ liệu lớn và nhiều đặc trưng. Thuật toán hoạt động tốt khi giả định độc lập gần đúng hoặc khi các đặc trưng ít phụ thuộc nhau. Nhược điểm là nếu giả định độc lập bị vi phạm mạnh, hiệu quả phân loại có thể giảm. Naive Bayes có nhiều biến thể như Gaussian Naive Bayes (cho dữ liệu liên tục), Multinomial Naive Bayes (thường dùng cho phân loại văn bản, xử lý các đặc trưng đếm số lần xuất hiện), và Bernoulli Naive Bayes (cho dữ liệu nhị phân). Trong xử lý ngôn ngữ tự nhiên, Multinomial Naive Bayes rất phổ biến vì phù hợp với đặc trưng đếm tần suất từ trong văn bản.

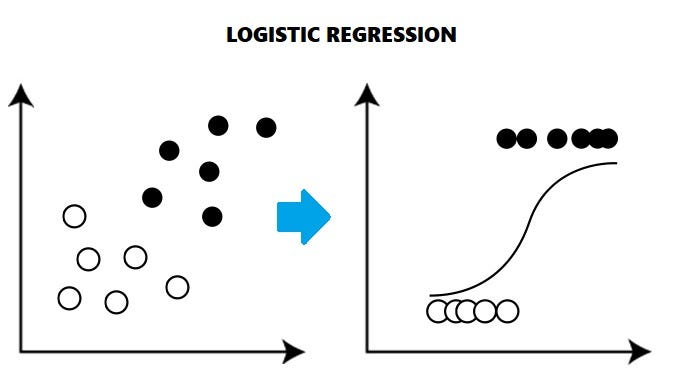
### 2.4.2. Logistic Regression

Logistic Regression là một thuật toán phân loại phổ biến trong học máy, dùng để dự đoán xác suất của một biến nhị phân (hai lớp) dựa trên một hoặc nhiều biến đầu vào.

Ý tưởng chính của Logistic Regression là mô hình hóa xác suất một mẫu thuộc về lớp đích (ví dụ lớp 1) bằng hàm logistic (hay hàm sigmoid), một hàm có giá trị đầu ra nằm trong khoảng từ 0 đến 1, phù hợp làm xác suất:

A math equations and numbers

AI-generated content may be incorrect.



**Hình 8. Mô tả mô hình Logistic Regression**

Giá trị này đại diện cho xác suất mà đối tượng có nhãn lớp 1. Để dự đoán nhãn lớp, ta thường dùng ngưỡng (threshold) 0.5: nếu xác suất lớn hơn hoặc bằng 0.5 thì dự đoán lớp 1, ngược lại là lớp 0. Logistic Regression được huấn luyện bằng cách tối ưu hàm mất mát (loss function) gọi là hàm log-loss hoặc cross-entropy, nhằm tìm các trọng số β sao cho dự đoán gần đúng nhãn thực tế nhất. Ưu điểm của Logistic Regression là đơn giản, dễ hiểu, hiệu quả trên dữ liệu tuyến tính và cho kết quả giải thích được (các trọng số cho biết mức độ ảnh hưởng của từng đặc trưng). Tuy nhiên, nếu dữ liệu không tuyến tính hoặc có quan hệ phức tạp, mô hình này có thể không phù hợp và cần các kỹ thuật hoặc mô hình phức tạp hơn.

Logistic Regression thường được dùng rộng rãi trong các bài toán phân loại nhị phân như phân tích cảm xúc, chẩn đoán y khoa, phân loại thư rác, ...

# CHƯƠNG III. THIẾT KẾ VÀ XÂY DỰNG CHƯƠNG TRÌNH

* 1. **Sơ đồ khối hệ thống**

A flowchart of a computer algorithm

AI-generated content may be incorrect.

**Hình 10. Sơ đồ khối hệ thống**

***Mô tả sơ đồ hệ thống***

Hệ thống phân tích và giám sát bình luận trên nền tảng web được tổ chức theo các khối chức năng như sau:

* **Người dùng (Giao diện web):** Người dùng nhập bình luận thông qua giao diện web do hệ thống xây dựng.
* **Flask Backend:** Dữ liệu từ giao diện người dùng được gửi về máy chủ thông qua Flask, chịu trách nhiệm tiếp nhận và điều phối luồng xử lý.
* **Tiền xử lý văn bản:** Văn bản được làm sạch, chuẩn hóa và tách từ sử dụng các thư viện như Gensim, Pyvi và Pandas để đảm bảo định dạng phù hợp cho mô hình học máy.
* **Vector hóa văn bản:** Văn bản sau tiền xử lý được chuyển đổi thành dạng vector bằng phương pháp CountVectorizer hoặc TF-IDF để phục vụ cho việc phân loại.
* **Mô hình học máy:** Mô hình đã được huấn luyện trước (Naive Bayes hoặc Logistic Regression) được sử dụng để dự đoán cảm xúc của bình luận.
* **Xử lý kết quả:** Nếu bình luận được phân loại là tiêu cực, hệ thống sẽ gắn nhãn cảnh báo. Dữ liệu sẽ được lưu vào cơ sở dữ liệu MySQL.
* **Trực quan hóa và báo cáo:** Dữ liệu đã xử lý được hiển thị thông qua dashboard web (Flask), cho phép theo dõi thống kê cảm xúc, tỷ lệ tích cực/tiêu cực và các thông tin chi tiết khác.

Hệ thống lọc comment thời gian thực hoạt động theo quy trình khép kín từ thu thập dữ liệu qua dữ liệu người dùng nhập, xử lý văn bản bằng các kỹ thuật NLP (CountVectorizer, TF-IDF), đến phân loại cảm xúc với mô hình học máy như Naive Bayes và Logistic Regression. Các comment tiêu cực được tự động phát hiện và xử lý, đồng thời tỷ lệ cảm xúc được trực quan hóa bằng biểu đồ Pie Chart giúp quản trị viên dễ dàng theo dõi. Hệ thống góp phần nâng cao chất lượng nội dung, bảo vệ hình ảnh thương hiệu và quản lý hiệu quả hơn.

* 1. **Sơ đồ thuật toán**
     1. Huấn luyện mô hình học máy

A diagram of a flowchart

AI-generated content may be incorrect.

**Hình 11. Sơ đồ khối chương trình huấn luyện mô hình**

* + 1. Lấy dữ liệu người dùng nhập và xử lý

A diagram of a flowchart

AI-generated content may be incorrect.

**Hình 12. Sơ đồ khối chương trình lấy dữ liệu người dùng nhập và xử lý**

* + 1. Ứng dụng Web để sử dụng

A diagram of a flowchart

AI-generated content may be incorrect.

**Hình 13. Sơ đồ khối chương trình Web**

* 1. **Xây dựng chương trình**
     1. Thu thập và xử lý dữ liệu.

Dữ liệu được thu thập và đánh nhãn dựa trên dữ liệu comment tiếng việt thực tế ở trên Facebook. Tập dữ liệu đã được chia thành 3 phần : tập train, tập valid và tập test.

A close-up of a computer code

AI-generated content may be incorrect.

Vì đây là dữ liệu thực tế, dữ liệu sẽ bao gồm cả những kí tự đặc biệt như icon, … Chính vì vậy phải tiền xử lý dữ liệu.

Dữ liệu trước khi tiền xử lý

A screenshot of a computer

AI-generated content may be incorrect.

Dữ liệu sau khi đã tiền xử lý

A white text with black and blue text

AI-generated content may be incorrect.

**Xử lý dữ liệu đầu vào:**

1. **Loại bỏ ký tự đặc biệt, biểu tượng (icon), emoji và các dấu hiệu không phải chữ:** Ở bước này, tất cả các ký tự không phải là chữ hoặc số sẽ được loại bỏ để tránh gây nhiễu cho mô hình. Ví dụ như dấu chấm câu, dấu ngoặc, ký tự đặc biệt, các biểu tượng cảm xúc, icon,… đều sẽ bị lọc ra nhằm giữ lại chỉ phần nội dung chữ thuần túy.
2. **Chuyển toàn bộ văn bản về chữ thường:** Việc chuẩn hóa toàn bộ văn bản về dạng chữ thường giúp giảm bớt sự phân tán dữ liệu do khác biệt về kiểu viết hoa – viết thường. Ví dụ: "Vất Vả" và "vất vả" sẽ được chuyển thành "vất vả", giúp mô hình học tốt hơn và thống nhất hơn khi phân tích.
3. **Xử lý đặc thù tiếng Việt bằng cách ghép các từ có ý nghĩa liên quan thành một đơn vị duy nhất (token hóa nâng cao):** Trong tiếng Việt, nhiều từ hoặc cụm từ khi đứng riêng lẻ có thể gây hiểu nhầm hoặc không rõ nghĩa. Từ **"mất"** và **" ngủ"** nếu đứng riêng rẽ thì khó thể hiện đúng ý nghĩa. Nhưng khi nối lại thành **"mất\_ngủ"** thì cụm từ này có ý nghĩa rõ ràng: rối loạn giấc ngủ, khó ngủ.

A yellow and pink circle with a number of percentages

AI-generated content may be incorrect.

**Hình 16. Phân bổ dữ liệu trong tập train**

Sử dụng biểu đồ Pie Chart (Biểu đồ tròn) để trực quan hoá dữ liệu đầu vào huấn luyện trong tập train.

* + 1. Vecto hoá văn bản

Ngôn ngữ của con người là ngôn ngự tự nhiên, máy móc không thể hiểu được. Chính vì vậy, phải biến đổi những chữa cái về dạng số 0, 1,.. Để máy móc có thể hiểu được. Trong bài này, em sẽ sử dụng 2 cách biến đổi là CountVectorizer và TF-IDF để so sánh sự khác nhau sẽ đạt được

* CountVectorizer

|  |
| --- |
| *# --- 2. Vector hóa văn bản ---* vectorizer = CountVectorizer() X\_train\_vec = vectorizer.fit\_transform(X\_train) X\_valid\_vec = vectorizer.transform(X\_valid) X\_test\_vec = vectorizer.transform(X\_test) |

|  |
| --- |
| **vectorizer = CountVectorizer()**   * Tạo một đối tượng CountVectorizer mặc định. * Tham số mặc định bao gồm:   + lowercase=True: Tự động chuyển văn bản thành chữ thường.   + stop\_words=None: Không loại bỏ từ dừng (như “the”, “is”, “in”...).   + max\_features=None: Không giới hạn số lượng từ.   + ngram\_range=(1,1): Chỉ lấy unigram (1 từ một lần).   + token\_pattern=r'(?u)\b\w\w+\b': Token là các từ dài ít nhất 2 ký tự trở lên. |
| **X\_train\_vec = vectorizer.fit\_transform(X\_train)**   * **fit\_transform()**:   + fit: học từ điển (vocabulary) từ tập X\_train.   + transform: chuyển từng văn bản trong X\_train thành vector (dựa trên từ điển vừa học). * Kết quả: X\_train\_vec là một **ma trận sparse** (dạng ma trận thưa) có kích thước (số mẫu, số từ). |
| **X\_valid\_vec = vectorizer.transform(X\_valid)**  **X\_test\_vec = vectorizer.transform(X\_test)**   * **transform()**:   + Sử dụng **cùng từ điển học được từ X\_train** để chuyển X\_valid và X\_test thành vector.   + Điều này đảm bảo không có "leakage" (rò rỉ dữ liệu từ validation/test sang train). |

* TF-IDF

|  |
| --- |
| *# --- 2. Vector hóa bằng TF-IDF ---* vectorizer = TfidfVectorizer() X\_train\_vec = vectorizer.fit\_transform(X\_train) X\_valid\_vec = vectorizer.transform(X\_valid) X\_test\_vec = vectorizer.transform(X\_test) |
| **vectorizer = TfidfVectorizer()**   * Tạo một đối tượng TfidfVectorizer từ thư viện sklearn.feature\_extraction.text. * TfidfVectorizer = Term Frequency – Inverse Document Frequency:   + **TF**: Tần suất xuất hiện của từ trong văn bản.   + **IDF**: Giảm trọng số của những từ phổ biến trên toàn bộ tập dữ liệu (như “là”, “và”...). * Mục tiêu: **Ưu tiên những từ phân biệt nội dung giữa các văn bản.** |
| **X\_train\_vec = vectorizer.fit\_transform(X\_train)**   * fit: học từ điển và tính IDF từ tập X\_train. * transform: biến đổi từng văn bản thành vector TF-IDF dựa trên từ điển và trọng số IDF vừa học được. |
| **X\_valid\_vec = vectorizer.transform(X\_valid)**  **X\_test\_vec = vectorizer.transform(X\_test)**   * Dùng **cùng từ điển và IDF học từ tập huấn luyện** để biến đổi các văn bản mới (validation/test). * Tránh "data leakage" (rò rỉ dữ liệu). |

* + 1. Mô hình học máy

Trong bài này, em sẽ sử dụng hai mô hình Naive Bayes và Logistic Regression để thử nghiệm va đánh giá xem mô hình nào tốt hơn so với bộ dữ liệu của mình.

* Naive Bayes

|  |
| --- |
| *# --- 3. Huấn luyện mô hình Naive Bayes ---* nb\_model = MultinomialNB() nb\_model.fit(X\_train\_vec, y\_train) |
| **MultinomialNB()**   * Là mô hình Naive Bayes phù hợp với **dữ liệu rời rạc (discrete)** như số lần xuất hiện từ, hoặc trọng số TF-IDF. * Đặc biệt hiệu quả trong bài toán phân loại văn bản (spam, cảm xúc, chủ đề...).   **.fit(X\_train\_vec, y\_train)**   * Huấn luyện mô hình trên dữ liệu vector hóa (X\_train\_vec) và nhãn (y\_train).   **Ưu điểm:**   * Nhanh, hiệu quả với dữ liệu lớn. * Ít bị overfitting, dễ triển khai.   **Nhược điểm:**  Giả định độc lập mạnh giữa các đặc trưng |

* Logistic Regression

|  |
| --- |
| *# --- 4. Huấn luyện Logistic Regression ---* lr\_model = LogisticRegression(max\_iter=1000) lr\_model.fit(X\_train\_vec, y\_train) |
| **LogisticRegression(max\_iter=1000)**   * Mô hình phân loại tuyến tính, thường dùng cho **phân loại nhị phân hoặc đa lớp**. * max\_iter=1000: tăng số vòng lặp tối đa để đảm bảo thuật toán hội tụ (do dữ liệu văn bản thường rất nhiều chiều → cần nhiều vòng lặp hơn mặc định 100).   **.fit(X\_train\_vec, y\_train)**   * Huấn luyện Logistic Regression trên dữ liệu văn bản đã vector hóa. |

|  |
| --- |
| *# --- 6. Lưu mô hình và vectorizer ---* joblib.dump(nb\_model, 'nb\_model.joblib') joblib.dump(lr\_model, 'lr\_model.joblib') joblib.dump(vectorizer, 'tfidf\_vectorizer.joblib') |

Tương tự khi lấy comment từ trên Facebook về chúng ta cũng phải thực hiện các bước tiền xử lý dữ liệu thì mới đưa vào mô hình để dự đoán được, mô hình sẽ dự đoán và trả về giá trị 1 – tích cực, 0 – tiêu cực

* + 1. Xây dựng Web

Xây dựng một Web mô phỏng như ứng dụng Facebook, đăng một bài bất kì mọi người có thể vào và bình luận nếu lời bình luận là tích cực sẽ được gửi và lưu lại vào cơ sở dữ liệu MySQL, còn lời đó lả tiêu cực thì sẽ cảnh báo vi phạm.

A diagram of a diagram

AI-generated content may be incorrect.

* 1. **Xây dựng bảng dữ liệu**

Để tạo ra những báo cáo về số lượt comment tích cực, tiêu cực... để người quản trị đánh giá thì phải lưu trữ những dữ liệu này.

Bảng comments

A screenshot of a computer

AI-generated content may be incorrect.

# CHƯƠNG IV. THỰC NGHIỆM VÀ KẾT LUẬN

4.1. Thực nghiệm

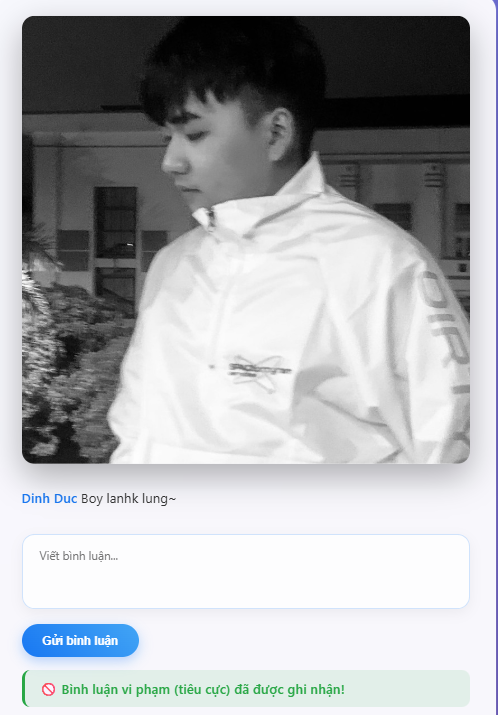
### Lựa chọn mô hình học máy

| Vectorizer | Mô hình | Tập | Accuracy | Precision (Tiêu cực) | Recall (Tiêu cực) | F1 (Tiêu cực) | Nhận xét chính |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TF-IDF | Naive Bayes | Validation | 0.813 | 0.86 | 0.19 | 0.31 | Bỏ sót nhiều comment tiêu cực |
|  |  | Test | 0.839 | 0.87 | 0.18 | 0.30 |  |
| TF-IDF | Logistic Regression | Validation | 0.852 | 0.80 | 0.45 | 0.58 | Cân bằng hơn ở cả 2 lớp |
|  |  | Test | 0.868 | 0.78 | 0.43 | 0.55 |  |
| CountVectorizer | Naive Bayes | Validation | 0.845 | 0.67 | 0.61 | 0.64 | Đã cải thiện recall rõ rệt |
|  |  | Test | 0.858 | 0.64 | 0.60 | 0.62 |  |
| CountVectorizer | Logistic Regression | Validation | 0.866 | 0.78 | 0.55 | 0.65 | Hiệu suất tốt nhất |
|  |  | Test | 0.873 | 0.74 | 0.51 | 0.61 | Tốt và ổn định |

**Kết luận : Logistic Regression + CountVectorizer (cân bằng, hiệu quả, độ chính xác cao).**

**4.2. Ứng dụng thực tế**

Tích hợp bộ lọc nhúng vào một trang Web mô phỏng trang Facebook cho phép người khác comment, nếu không vi phạm sẽ được hiển thị.

****

Xem báo cáo tỉ lệ comments :

A screenshot of a graph

AI-generated content may be incorrect.

4.2. Kết luận

Trong quá trình triển khai bài toán phân loại cảm xúc từ các bình luận, nhóm đã đạt được nhiều thành tựu nổi bật cả về mặt kỹ thuật lẫn ứng dụng thực tiễn.

a) Thành quả đã đạt được

* Hoàn thiện hệ thống phân loại bình luận thành hai nhóm tích cực và tiêu cực, sử dụng các bước tiền xử lý dữ liệu tiếng Việt, kết hợp kỹ thuật vector hóa TF-IDF và CountVectorizer cùng với các mô hình học máy như Naive Bayes và Logistic Regression.
* Ứng dụng mô hình vào thực tế thông qua công cụ lọc bình luận tiêu cực, có khả năng tích hợp trực tiếp vào fanpage để hỗ trợ tự động kiểm duyệt nội dung.
* Xây dựng website giám sát bình luận theo thời gian thực dựa trên Flask và MySQL, cung cấp giao diện trực quan để thống kê và theo dõi các chỉ số quan trọng.

b) Những kiến thức thu nhận được

* Nắm vững quy trình xử lý dữ liệu văn bản tiếng Việt, bao gồm các bước chuẩn hóa, loại bỏ ký tự đặc biệt, từ dừng và biểu tượng cảm xúc.
* So sánh hiệu quả của các phương pháp vector hóa TF-IDF và CountVectorizer, cũng như đánh giá sự khác biệt giữa các mô hình phân loại Naive Bayes và Logistic Regression.
* Thành thạo các tiêu chí đánh giá mô hình như accuracy, precision, recall và F1-score, đặc biệt trong trường hợp dữ liệu bị mất cân bằng nhãn.
* Có kinh nghiệm triển khai hệ thống hoàn chỉnh từ thu thập dữ liệu, xử lý, huấn luyện mô hình đến phát triển hệ thống giám sát trực tuyến.

c) Định hướng phát triển trong tương lai

* Nâng cao độ chính xác của mô hình, đặc biệt với lớp “bình luận tiêu cực” bằng cách áp dụng các mô hình mạnh hơn như Random Forest, SVM hoặc fine-tune các mô hình ngôn ngữ tiên tiến như BERT/PhoBERT.
* Áp dụng các kỹ thuật xử lý mất cân bằng dữ liệu hiệu quả hơn như SMOTE hoặc focal loss để cải thiện khả năng nhận diện.
* Cải tiến hệ thống web thành nền tảng cảnh báo tự động theo thời gian thực, giúp quản trị viên có thể phản ứng nhanh chóng trước sự gia tăng của các bình luận tiêu cực.
* Phát triển thêm các tính năng phân tích xu hướng cảm xúc theo thời gian, hỗ trợ đo lường và đánh giá hiệu quả các chiến dịch truyền thông một cách toàn diện hơn.