**TRƯỜNG ĐẠI HỌC THỦY LỢI**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**BÁO CÁO MÔN HỌC KHAI PHÁ DỮ LIỆU**

**CHỦ ĐỀ**

**ỨNG DỤNG PHÂN LOẠI DỮ LIỆU TRONG**

**BÀI TOÁN PHÂN TÍCH CẢM XÚC**

**Giảng viên hướng dẫn: TS.** Lê Thị Tú Kiên

**Nhóm sinh viên thực hiện:** Nhóm 10

1. Trần Duy Tuấn(Nhóm trưởng) - 2251061917
2. Đinh Ngọc Anh - 2251061696
3. Trần Quốc Anh - 2251061713
4. Mông Thị Bích Hạnh - 2251061770
5. Hoàng Minh Tuấn - 2251061917

**Lời nói đầu**

Trước tiên, chúng em xin gửi lời cảm ơn chân thành đến cô Lê Thị Tú Kiên, người đã tận tình hướng dẫn và hỗ trợ chúng em trong suốt quá trình thực hiện dự án này. Nhờ sự chỉ dẫn tận tâm của cô, chúng em đã có cơ hội tiếp cận và tìm hiểu sâu hơn về các phương pháp phân tích dữ liệu, từ đó nâng cao kiến thức cũng như kỹ năng thực hành của bản thân qua đó chúng em có thể thực hiện dự án này.

Chúng ta đang sống trong thời đại công nghệ số, dữ liệu giữ vai trò quan trọng trong mọi lĩnh vực của cuộc sống. Việc thu thập và phân tích dữ liệu một cách hiệu quả giúp chúng ta đưa ra quyết định chính xác, từ đó tạo ra giá trị thực tiễn. Đặc biệt, khi các nền tảng mạng xã hội ngày càng phổ biến, phân tích cảm xúc – một ứng dụng quan trọng của khai phá dữ liệu – trở thành công cụ hữu ích trong việc nghiên cứu hành vi người dùng, hỗ trợ các chiến lược marketing và dự đoán xu hướng thị trường. Ngoài ra, nó còn giúp giám sát dư luận về thương hiệu hoặc các sự kiện quan trọng.

Thông qua dự án này, chúng em mong muốn áp dụng các kỹ thuật phân tích dữ liệu để khám phá những thông tin hữu ích, đồng thời rèn luyện tư duy logic và kỹ năng xử lý dữ liệu để phục vụ cho công việc sau này.

Chúng em hy vọng rằng nội dung nghiên cứu này sẽ mang lại những giá trị hữu ích về mối liên hệ giữa phân tích dữ liệu và phân tích cảm xúc đồng thời làm nổi bật những tiềm năng to lớn của nó và rất mong nhận được sự góp ý từ cô và mọi người để hoàn thiện hơn. Xin chân thành cảm ơn!

**Mục lục**

[**Chương 1: Tổng quan về phương pháp phân tích dữ liệu 6**](#_jnz8cqxwoblv)

[I. Tổng quan về phân loại dữ liệu. 6](#_f3jeh6jj7pch)

[1. Khái niệm về phân loại dữ liệu. 6](#_atd0rp6f1ryr)

[2. Ứng dụng của phân loại dữ liệu. 6](#_slk9khf6m4if)

[3. Phân hoạch các thuật toán phân loại dữ liệu. 7](#_n1fwyhnwef53)

[II. Tổng quan về Sentiment Analyst. 8](#_6tcid6t0kng3)

[1. Khái niệm của Sentiment Analyst. 8](#_8e03hqb91x0f)

[2. Các cấp độ Sentiment Analyst. 8](#_ax9fbixqnqj8)

[3. Quy trình thực hiện Sentiment Analyst. 8](#_x3qg1aco1dko)

[4. Các phương pháp Sentiment Analyst. 9](#_jfdtfd2j5wv5)

[**- Học máy truyền thống: Thích hợp cho tập dữ liệu vừa và lớn, dễ triển khai với các thư viện như scikit-learn. Tuy nhiên, cần đảm bảo dữ liệu đủ nhãn và chất lượng. 9**](#_h4o52jryzk2w)

[**Chương 2: Định nghĩa bài toán và các phương pháp giải quyết bài toán 11**](#_vsxyc895jh7e)

[I. Định nghĩa bài toán. 11](#_ms8pww3oebl3)

[1. Lí do chọn đề tài. 11](#_sv3rtkkq0gv5)

[2. Tổng quan bài toán. 12](#_kb9h2d67vr6s)

[3. Quy trình thực hiện. 13](#_rgli6niln5y4)

[**Chương 3: Thực nghiệm so sánh và đánh giá kết quả 14**](#_kcnctxy5p9qx)

[I. Các phương pháp giải quyết bài toán. 14](#_823uh9e7pme6)

[1. Phương pháp dựa trên quy tắc. 14](#_v825p6i9ihb4)

[2. Học máy truyền thống. 14](#_bjpa0xnqzt26)

[● Support Vector Machine (SVM) 14](#_s0lq8y88q2y3)

[● Logistic Regression 14](#_1mg01ujqj31m)

[3. Kết hợp các phương pháp. 14](#_r9cuzc6zeysd)

[II. Quy trình thực hiện thuật toán. 14](#_gew9mpfz3qzb)

[1. Thu thập và tìm hiểu tập dữ liệu. 14](#_bin5nph1bc7i)

[2. Tiền xử lý dữ liệu. 20](#_dlw3tkupu30l)

[Phân tích: 32](#_lkitf1w7xcr0)

[Kết luận: 33](#_plcbn7kncghu)

[Phân tích: 34](#_6s1mkt1qo6xh)

[Kết luận: 34](#_xxprl1yyyn4o)

[Phân tích: 35](#_ndphr47uftrd)

[Kết luận: 36](#_dsmot7njzkwc)

[- Cảm xúc tích cực (Positive Sentiment): 37](#_yfadz7jtcoqo)

[- Cảm xúc trung lập (Neutral Sentiment): 38](#_qu1y6f1r0g5y)

[- Cảm xúc tiêu cực (Negative Sentiment): 38](#_ndvoqvfvgjub)

[Kết luận: 38](#_13yb7c4ghv7z)

[Phân tích: 39](#_8vblk5nl8v69)

[Kết luận: 40](#_2pv744owwx87)

[3. Trích Xuất Đặc Trưng. 40](#_1huyklu9dnnx)

[4. Huấn Luyện Mô Hình. 43](#_qufdwhpgqcoe)

[4.1. SVM (Support Vector Machine). 43](#_9lm6am8fzm55)

[4.2. Logistic Regression 50](#_xgtk7qqcqgfc)

[- Đầu ra (Outputs) 51](#_ris5cxs3tpxm)

[- Các bước thuật toán 51](#_gme3qu9389be)

[5.Triển Khai. 57](#_g2mgs6y2063u)

[**Chương 4: Kết luận 59**](#_cgg619s2zgot)

[Tương lai và ứng dụng: 60](#_dco91e6n7j0q)

**Đóng góp của các thành viên**

1. **Trần Duy Tuấn - Nhóm trưởng**

* Tiền xử lý dữ liệu
* EDA (Khám phá dữ liệu)
* Demo mô hình dự đoán

1. **Đinh Ngọc Anh**

* Gán nhãn dữ liệu
* Trích xuất đặc trưng

1. **Trần Quốc Anh**

* Thuật toán SVM (Support Vector Machine)

1. **Mông Thị Bích Hạnh**

* Thu thập dữ liệu
* Tìm hiểu bộ dữ liệu

1. **Hoàng Minh Tuấn**

* Thuật toán Logistic Regression

# 

# 

# 

# 

# **Chương 1: Tổng quan về phương pháp phân tích dữ liệu**

## **I. Tổng quan về phân loại dữ liệu.**

### ***1. Khái niệm về phân loại dữ liệu.***

* **Phân loại dữ liệu:**
* Là một kỹ thuật trong khai phá dữ liệu và học máy
* Là một quá trình có hệ thống để phân loại thông tin dựa trên mức độ nhạy cảm, tính quan trọng và tầm quan trọng của thông tin đối với một tổ chức.
* Quá trình này bao gồm việc gán nhãn hoặc thẻ cho các tập dữ liệu, cho phép các tổ chức áp dụng các biện pháp bảo mật phù hợp với giá trị và tính bảo mật của thông tin.
* Đây là một dạng học có giám sát(Supervised Learning)

### **2. *Ứng dụng của phân loại dữ liệu.***

* **Xử lý ngôn ngữ tự nhiên (NLP - Natural Language Processing)**
* **Phân tích cảm xúc (Sentiment Analysis) :**Dùng để phân loại cảm xúc của người dùng từ văn bản, ví dụ: Phân tích đánh giá sản phẩm (tích cực, trung lập, tiêu cực): Theo dõi phản ứng của người dùng trên mạng xã hội (Facebook, Twitter).
* **Phân loại email (Email Classification):** Xác định email là spam (thư rác) hay not spam (hợp lệ).
* **Nhận diện ngôn ngữ (Language Detection):** Phân loại văn bản theo ngôn ngữ (Anh, Việt, Pháp,...). Ứng dụng trong dịch máy (Google Translate, DeepL).
* **Y tế & Chăm sóc sức khỏe**
* **Chẩn đoán bệnh tự động:** Sử dụng dữ liệu y tế để phân loại bệnh nhân có mắc bệnh hay không (ví dụ: chẩn đoán ung thư dựa trên hình ảnh X-ray, MRI).
* **Phát hiện bất thường trong dữ liệu sức khỏe:** Dự đoán nguy cơ đột quỵ, tiểu đường dựa trên hồ sơ bệnh án.
* **Tài chính & Ngân hàng**
* **Phát hiện gian lận tài chính (Fraud Detection):** Nhận diện các giao dịch bất thường, giúp ngăn chặn lừa đảo thẻ tín dụng, rửa tiền. Các thuật toán phổ biến: Random Forest, Neural Networks, Anomaly Detection.
* **Đánh giá tín dụng (Credit Scoring):** Phân loại khách hàng có nguy cơ vỡ nợ hay không, hỗ trợ quyết định cấp khoản vay. Ứng dụng trong các hệ thống chấm điểm tín dụng (FICO, CIC).
* **An ninh mạng & Bảo mật**
* **Phát hiện phần mềm độc hại (Malware Detection):** Sử dụng các thuật toán phân loại để xác định file có chứa mã độc hay không.
* **Nhận diện hành vi xâm nhập (Intrusion Detection):** Phân loại các gói tin mạng để phát hiện hành vi tấn công .
* **Thương mại điện tử & Tiếp thị**
* **Hệ thống gợi ý sản phẩm (Recommendation Systems):** Phân loại khách hàng theo sở thích để cá nhân hóa quảng cáo, gợi ý sản phẩm.
* **Xác định đối tượng khách hàng tiềm năng (Customer Segmentation):** Dùng phân loại để nhóm khách hàng thành các phân khúc (VIP, phổ thông, tiềm năng).
* **Nhận diện hình ảnh & Thị giác máy tính (Computer Vision)**
* **Nhận diện khuôn mặt (Face Recognition):** Phân loại hình ảnh có chứa khuôn mặt của một người cụ thể hay không. Ứng dụng trong bảo mật, mở khóa điện thoại (Face ID).
* **Phân loại vật thể trong ảnh (Object Classification):** Dùng trong xe tự lái, camera giám sát để phân loại đối tượng (người, xe, động vật,...).

### 

### ***3. Phân hoạch các thuật toán phân loại dữ liệu.***

* Cây quyết định (Decision Tree)
* Máy vector hỗ trợ (SVM - Support Vector Machine)
* Naïve Bayes

1 số cách tiếp cận phân loại dữ liệu:

* Phân loại nhị phân (Binary Classification)
* Phân loại đa lớp (Multi-class Classification)
* Phân loại đa nhãn (Multi-label Classification)

## **II. Tổng quan về Sentiment Analyst.**

### ***1. Khái niệm của Sentiment Analyst.***

**Sentiment Analysis (Phân tích cảm xúc):**

* Là một kỹ thuật trong khai phá dữ liệu, sử dụng xử lý ngôn ngữ tự nhiên (NLP), học máy (ML) và khai thác văn bản để xác định và phân loại cảm xúc trong dữ liệu văn bản.
* Nó thường được sử dụng để phân tích ý kiến của người dùng về một sản phẩm, dịch vụ hoặc sự kiện.

### ***2. Các cấp độ Sentiment Analyst.***

* **Phân loại nhị phân (Binary Classification)**: Xác định xem một văn bản có cảm xúc tích cực hay tiêu cực.
* **Phân loại đa cấp độ (Multi-class Classification)**: Xếp loại văn bản theo nhiều mức độ cảm xúc (tích cực, trung lập, tiêu cực).
* **Phân tích theo cảm xúc cụ thể**: Xác định các loại cảm xúc như vui, buồn, giận dữ, lo lắng,...
* **Phân tích cảm xúc theo khía cạnh (Aspect-based Sentiment Analysis - ABSA)**: Phân tích cảm xúc của người dùng theo từng khía cạnh cụ thể của sản phẩm/dịch vụ.

### ***3. Quy trình thực hiện Sentiment Analyst.***

* **Thu thập dữ liệu**: Lấy văn bản từ các nguồn như mạng xã hội, đánh giá khách hàng, blog, v.v.
* **Tiền xử lý dữ liệu**: Làm sạch văn bản bằng cách loại bỏ dấu câu, từ dừng, và chuẩn hóa về chữ thường.
* **Trích Xuất Đặc Trưng**: Chuyển văn bản thành dạng số, sử dụng các phương pháp như Bag of Words (BoW), TF-IDF hoặc nhúng từ (word embeddings).
* **Huấn Luyện Mô Hình**: Sử dụng thuật toán học máy hoặc học sâu để huấn luyện trên dữ liệu có nhãn.
* **Đánh Giá Mô Hình**: Kiểm tra hiệu suất trên tập dữ liệu kiểm tra bằng các chỉ số như độ chính xác, độ chính xác (precision), độ nhạy (recall) và F1-score.
* **Triển Khai**: Áp dụng mô hình để phân tích văn bản mới.

### ***4. Các phương pháp Sentiment Analyst.***

* **Phương pháp dựa trên quy tắc**: Phù hợp khi bạn có tập dữ liệu nhỏ hoặc cần kết quả nhanh, nhưng không hiệu quả với văn bản có ngữ cảnh phức tạp như câu mang tính mỉa mai.
* **VADER (Valence Aware Dictionary and sEntiment Reasoner):** Phù hợp cho văn bản ngắn, đặc biệt là từ mạng xã hội.
* **SentiWordNet:** Sử dụng từ điển cảm xúc để gán giá trị tình cảm cho từng từ.
* **AFINN:** Một từ điển chứa các từ với giá trị cảm xúc từ -5 đến 5.

# **Học máy truyền thống**: Thích hợp cho tập dữ liệu vừa và lớn, dễ triển khai với các thư viện như scikit-learn. Tuy nhiên, cần đảm bảo dữ liệu đủ nhãn và chất lượng.

* **Naive Bayes**: Đơn giản, hiệu quả cho văn bản ngắn.
* **Support Vector Machines (SVM):** Hiệu quả trong việc phân loại văn bản.
* **Decision Trees và Random Forests:** Có thể xử lý các đặc trưng phức tạp.
* **Logistic Regression:** Đơn giản nhưng hiệu quả cho phân loại nhị phân.
* **Học sâu**: Đạt hiệu suất cao nhất, đặc biệt với văn bản dài hoặc phức tạp, nhưng đòi hỏi máy tính mạnh và kinh nghiệm về deep learning. Ví dụ, mô hình BERT từ Hugging Face có thể được tinh chỉnh cho phân tích cảm nghĩ.
* **Recurrent Neural Networks (RNN):** Xử lý tốt chuỗi thời gian của văn bản.
* **Long Short-Term Memory (LSTM):** Cải tiến của RNN, xử lý vấn đề gradient vanishing.
* **Convolutional Neural Networks (CNN):** Phù hợp cho việc phát hiện tính năng cục bộ trong văn bản.
* **Transformers:** Mô hình như BERT, RoBERTa, XLNet, có khả năng hiểu ngữ cảnh và từ vựng rất tốt.
* **Kết hợp các phương pháp:** Kết hợp các phương pháp trên để tăng cường hiệu suất, ví dụ như sử dụng từ điển để tạo ra các đặc trưng bổ sung cho mô hình học máy.
* **Phương pháp khác:**
* **Aspect-based Sentiment Analysis:** Phân tích cảm xúc theo các khía cạnh cụ thể của sản phẩm hoặc dịch vụ.
* **Cross-lingual Sentiment Analysis:** Phân tích cảm xúc trong các ngôn ngữ khác nhau.

# **Chương 2: Định nghĩa bài toán và các phương pháp giải quyết bài toán**

## **I. Định nghĩa bài toán.**

### ***1. Lí do chọn đề tài.***

Trong thời đại công nghệ số, việc khai thác và phân tích dữ liệu một cách hiệu quả giúp đưa ra những quyết định chính xác, từ đó tạo ra giá trị thực tiễn. Đặc biệt là khi mà nhiều nền tảng xã hội mọc lên thì việc phân tích cảm xúc một trong những ứng dụng của phân tích dữ liệu là cần thiết để ta có thể giúp ta nghiên cứu thêm về hành vi của người dùng và Hỗ trợ marketing đồng thời dự đoán xu hướng thị trường. Bài toán đặt ra là việc phân tích cảm xúc của nhóm khách hàng về các sản phẩm công nghệ dựa trên đặc trưng dữ liệu như:

* **Tích cực (Positive)**: Văn bản thể hiện cảm xúc hoặc thái độ lạc quan, vui mừng.
* **Tiêu cực (Negative)**: Văn bản thể hiện sự không hài lòng, phẫn nộ, hoặc chỉ trích.  
  **Trung lập (Neutral)**: Văn bản không thể hiện cảm xúc rõ ràng, mang tính khách quan hoặc thông tin trung lập.

Mục tiêu chính là phát hiện và phân chia khách hàng thành các cục có đặc điểm tương đồng với ý nghĩa:

* Giúp người bán, doanh nghiệp xác định được nhóm khách hàng tiềm năng
* Xây dựng chiến lược marketing phù hợp
* Cá nhân hóa dịch vụ và tối ưu hóa trải nghiệm khách hàng

Qua dự án này sinh viên có thể áp dụng các phương pháp phân tích dữ liệu như phân tích cảm xúc (Sentiment Analyst) vào thực tế, rèn luyện kỹ năng tư duy phân tích và đóng góp vào việc thúc đẩy xu hướng cá nhân hóa trong kỷ nguyên số hóa.

### ***2. Tổng quan bài toán.***

Dataset bao gồm 2550 mẫu dữ liệu bao gồm 9 cột dữ liệu chính

* brand: tên thương hiệu
* subreddit: tên subreddit chứa bài viết
* post\_id: id của bài viết
* post\_title: tiêu đề của bài viết
* comment\_id: id của bình luận
* comment\_body: nội dung bình luận
* comment\_score: số điểm của bình luận
* comment\_author: tác giả của bình luận
* comment\_time: thời gian bình luận

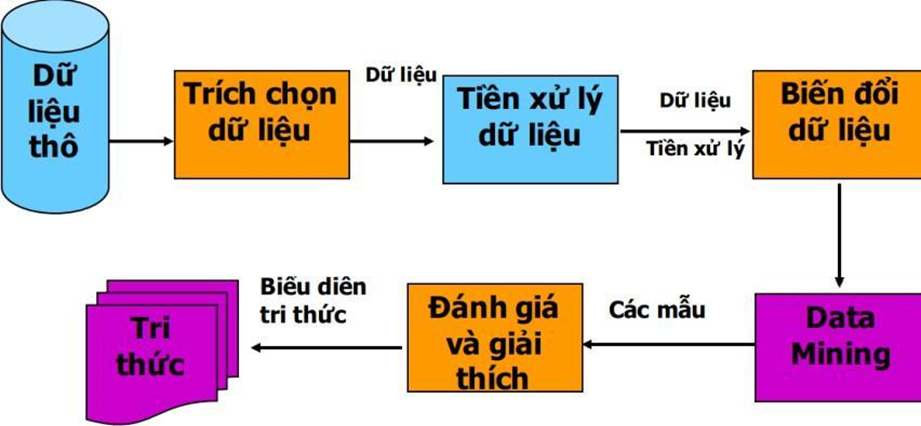
Mỗi dòng trong dataset là thái độ của người dùng,khách hàng đối với sản phẩm thuộc các thương hiệu như iPhone, SamSung, Xiaomi... bao gồm 3 cảm nhận chính:

* **Tích cực**: Người dùng đánh giá cao các sản phẩm của **Apple (iPhone)** với trải nghiệm hệ sinh thái mượt mà, đồng bộ với Mac/iCloud, ổn định và tiện dụng. Một số bình luận về **Samsung** cũng ghi nhận giao diện và ứng dụng tiện lợi, đặc biệt là khả năng tùy chỉnh.
* **Tiêu cực**: Một số bình luận chỉ trích **Facebook**, **Reddit**, hoặc các ứng dụng cụ thể hoạt động không ổn định trên thiết bị. Ngoài ra, người dùng cũng than phiền về **OneDrive**, **Google Calendar** mới cập nhật không tốt, hoặc sự lạm dụng quyền riêng tư từ các app trên iPhone. Samsung đôi khi cũng bị phàn nàn vì các cập nhật phần mềm làm giảm trải nghiệm người dùng.
* **Trung lập**: Nhiều người chỉ đơn giản chia sẻ quan sát, đưa ra gợi ý hoặc giải thích về cách hoạt động của các ứng dụng/hệ thống, mà không thể hiện cảm xúc mạnh mẽ tích cực hay tiêu cực. Ví dụ như nói về iCloud là lựa chọn mặc định trên Mac, hay lý do chọn Samsung Calendar thay vì Google Calendar.

Những cảm nhận trên dựa trên các đặc điểm chính sau về sản phẩm:

* **Trải nghiệm ứng dụng và hiệu suất thiết bị**:Người dùng thảo luận về các lỗi phần mềm, khả năng kết nối, độ ổn định của ứng dụng như Facebook, Reddit, Google Calendar,… kèm theo phản ánh việc cập nhật phần mềm gây ảnh hưởng đến trải nghiệm tổng thể trên thiết bị.
* **So sánh giữa các thương hiệu thiết bị**: iPhone thường được so sánh với Samsung về độ mượt, tính đồng bộ, và sự riêng tư. Người dùng cũng nhắc đến chuyển đổi giữa nền tảng iOS và Android hoặc giữa các ứng dụng mặc định (ví dụ: iCloud vs OneDrive).
* **Vấn đề bảo mật và quyền riêng tư**: TMột số bình luận thể hiện mối lo ngại về quyền truy cập danh bạ của Facebook trên iPhone mà không có cách kiểm soát rõ ràng, làm nổi bật sự lo ngại về quyền riêng tư trên các nền tảng di động.
* **Chính sách và ảnh hưởng của các công ty công nghệ lớn**: Các bình luận đề cập đến **Meta/Facebook**, **Google**, và thậm chí các quyết định của chính phủ như **thuế nhập khẩu** ảnh hưởng đến giá bán thiết bị (ví dụ: iPhone).

### ***3. Quy trình thực hiện.***



* **Bước 1:** Thu thập dữ liệu lấy văn bản từ nguồn reddit .
* **Bước 2 :** Tiền xử lý dữ liệu làm sạch văn bản bằng cách loại bỏ dấu câu, từ dừng, và chuẩn hóa về chữ thường.
* **Bước 3:** Trích Xuất Đặc Trưng chuyển văn bản thành dạng số, sử dụng các phương pháp như Bag of Words (BoW), TF-IDF
* **Bước 4:** Huấn Luyện Mô Hình sử dụng thuật toán học máy hoặc học sâu để huấn luyện trên dữ liệu có nhãn.
* **Bước 5:** Đánh Giá Mô Hình kiểm tra hiệu suất trên tập dữ liệu kiểm tra bằng các chỉ số như độ chính xác, độ chính xác (precision), độ nhạy (recall) và F1-score.
* **Bước 6:** Triển Khai áp dụng mô hình để phân tích văn bản mới.

# **Chương 3: Thực nghiệm so sánh và đánh giá kết quả**

## **I. Các phương pháp giải quyết bài toán.**

### ***1. Phương pháp dựa trên quy tắc.***

* **Bag-of-Words (BoW)**
* **TF-IDF**
* **Đặc trưng thủ công**
* **TF-IDF kết hợp với đặc trưng thủ công**
* **BoW kết hợp với đặc trưng thủ công**

### ***2. Học máy truyền thống.***

### Support Vector Machine (SVM)

### Logistic Regression

### ***3. Kết hợp các phương pháp.***

* **BoW kết hợp với đặc trưng thủ công +** SVM
* **BoW kết hợp với đặc trưng thủ công +** Logistic Regression

## **II. Quy trình thực hiện thuật toán.**

### ***1. Thu thập và tìm hiểu tập dữ liệu.***

1.1. Thu thập dữ liệu.

*1.1.1. Giới thiệu về Reddit và nguồn dữ liệu.*

* Reddit là một trong những nền tảng mạng xã hội lớn nhất với lượng người dùng đông đảo, nơi mà các thành viên có thể chia sẻ và thảo luận về các chủ đề từ công nghệ đến đời sống. Trong nghiên cứu này, chúng ta sẽ sử dụng Reddit làm nguồn dữ liệu để phân tích cảm nghĩ của người dùng về các sản phẩm công nghệ. Các subreddit nổi bật như **r/technology**, **r/gadgets**, **r/smartphones**, **r/apple**, và **r/android** sẽ được sử dụng để thu thập dữ liệu từ các bài viết và bình luận.

1.1.2. Lý do chọn Reddit làm nguồn dữ liệu.

* Reddit là một nguồn dữ liệu phong phú cho phân tích cảm nghĩ vì người dùng thường xuyên thảo luận, đánh giá và chia sẻ kinh nghiệm cá nhân về các sản phẩm công nghệ. Các bình luận trên Reddit có tính tự nhiên và chân thật, giúp phản ánh quan điểm đa dạng và cảm xúc thực tế từ người tiêu dùng.

*1.1.3. Quy trình thu thập dữ liệu.*

* **Bước 1:** Truy cập:<https://www.reddit.com/prefs/apps>.
* **Bước 2:** Điền thông tin theo mẫu và Chọn **"Create App"** hoặc **"Create Another App" :**

***Giao diện quản lý ứng dụng của Reddit (Reddit Developer Portal)***

* Sau khi tạo xong, bạn sẽ thấy:
* client\_id (là dòng chữ phía dưới app name)
* client\_secret (dòng "secret")
* **Bước 3:** Thiết lập kết nối với Reddit:

Đầu tiên, một kết nối tới Reddit được thiết lập thông qua việc cung cấp các thông tin xác thực bao gồm client\_id, client\_secret, và user\_agent. Đây là các thông tin cần thiết để truy cập vào API của Reddit.

| reddit = asyncpraw.Reddit(  client\_id="vawf1ksMlZvt40z5fPQnOg",  client\_secret="xEPH62kQkyHBhIUEL30wIzC1A7rrww",  user\_agent='user sentiment analysis'  ) |
| --- |

* **Bước 4:** Danh sách các từ khóa tìm kiếm (Thương hiệu sản phẩm):

Để lọc dữ liệu liên quan đến các sản phẩm công nghệ, chúng ta tạo một danh sách các từ khóa đại diện cho các thương hiệu phổ biến như: **iPhone**, **Samsung**, **Xiaomi**. Các bình luận chứa các từ khóa này sẽ được thu thập để phân tích cảm nghĩ.

| brands = ['iPhone', 'Samsung', 'Xiaomi'] |
| --- |

* **Bước 5:** Danh sách các subreddit:

Các subreddit được chọn lựa bao gồm các chủ đề liên quan đến công nghệ và sản phẩm điện tử như: **r/technology**, **r/gadgets**, **r/smartphones**, **r/apple**, **r/android**. Việc chọn lựa này đảm bảo thu thập được các bình luận có liên quan đến các sản phẩm công nghệ phổ biến.

| subreddits = ['technology', 'gadgets', 'android', 'apple', 'smartphones'] |
| --- |

* **Bước 6:** Thu thập dữ liệu từ các bài viết và bình luận:

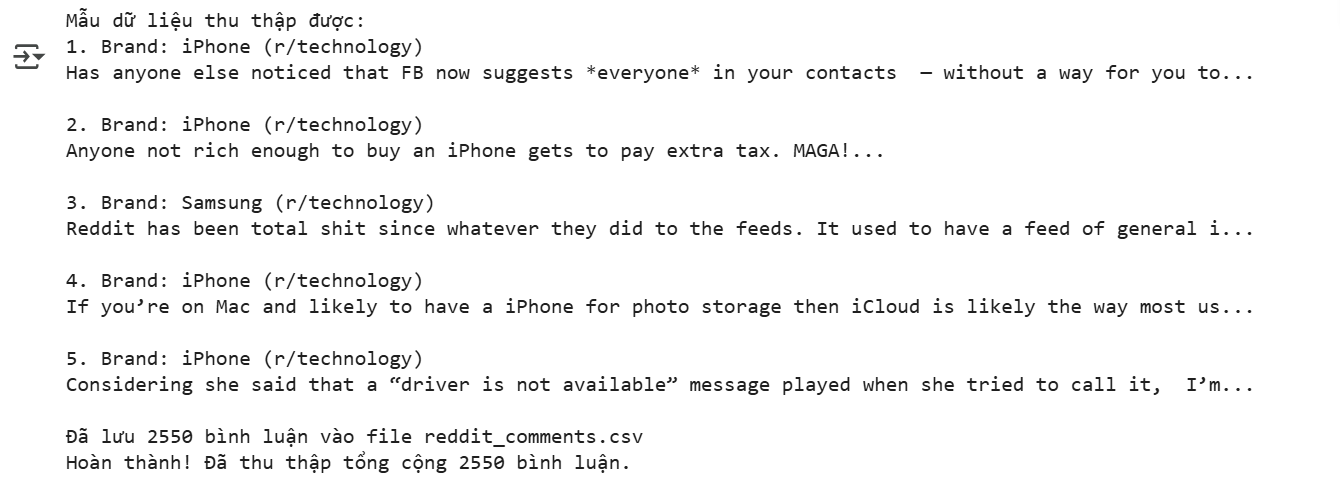
Với mỗi subreddit, ta sẽ thu thập các bài viết **hot** (nổi bật) và lấy tất cả các bình luận từ những bài viết này. Sau đó, chương trình sẽ kiểm tra xem bình luận có chứa bất kỳ từ khóa (thương hiệu) nào trong danh sách hay không. Nếu có, thông tin về bình luận sẽ được lưu lại. Các thông tin thu thập bao gồm:

* **Brand**: Tên thương hiệu được nhắc đến trong bình luận.
* **Subreddit**: Tên subreddit nơi bình luận được đăng.
* **Post ID** và **Post Title**: ID và tiêu đề của bài viết chứa bình luận.
* **Comment ID**, **Comment Body**, **Author Name**, **Comment Score**: Thông tin về bình luận (ID, nội dung, tên tác giả, điểm số).
* **Comment Time**: Thời gian đăng bình luận.

| # Lấy dữ liệu từ subreddit  async **for** post **in** subreddit.hot(limit=100):  ...  # Duyệt qua từng bình luận  **for** comment **in** all\_comments:  **if** hasattr(comment, 'body'):  brand = get\_brand\_from\_text(comment.body, brands)  **if** brand:  comments\_data.append({  'brand': brand,  'subreddit': subreddit\_name,  'post\_id': post.id,  'post\_title': post.title,  'comment\_id': comment.id,  'comment\_body': comment.body,  'comment\_author': author\_name,  'comment\_score': comment.score,  'comment\_time': comment.created\_utc  }) |
| --- |

* **Bước 7:** Lưu trữ và xuất dữ liệu

| # Lưu dữ liệu vào file CSV  df = pd.DataFrame(comments\_data)  df.to\_csv('/content/drive/MyDrive/BTL\_Data Mining/Dataset/reddit\_comments.csv', index=False) |
| --- |

**⇒ Results:** 

***Mẫu kết quả quá trình thu thập dữ liệu từ Reddit***

1.2. Tìm hiểu dữ liệu.

Bộ dữ liệu của tôi gồm 2550 dòng với các thuộc tính:

**Cột brand**: Chứa tên các thương hiệu sản phẩm công nghệ (iPhone, Samsung, Xiaomi), dùng để phân loại bình luận theo thương hiệu.

**Cột subreddit**: Chứa tên subreddit (ví dụ: "technology", "gadgets"), xác định cộng đồng mà bình luận thuộc về.

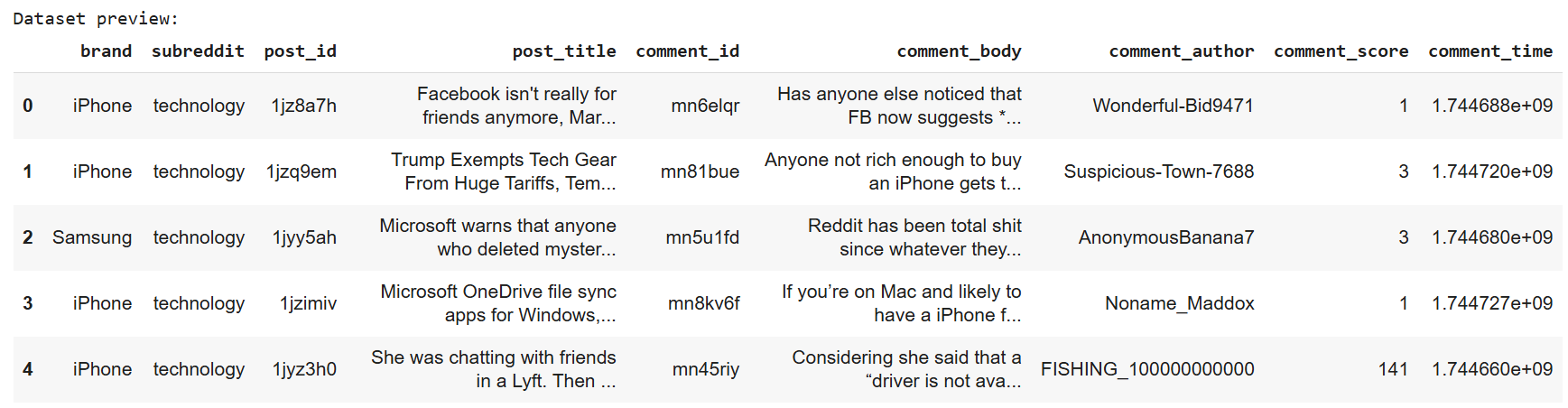
**Cột post\_id và post\_title**: Cung cấp thông tin về bài viết, giúp xác định nguồn gốc của bình luận.

**Cột comment\_id và comment\_body**: Chứa ID và nội dung của bình luận.

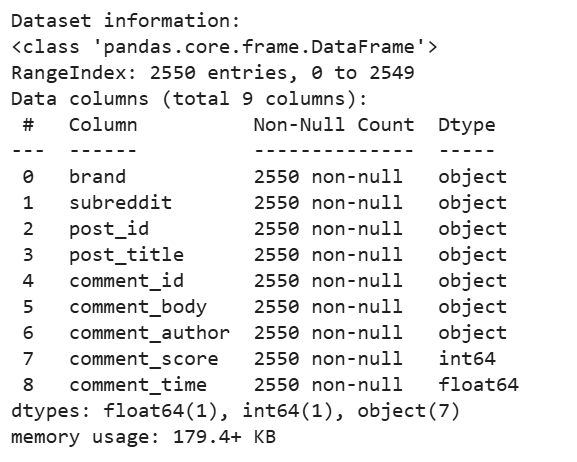
**Cột comment\_author**: Lưu tên tác giả bình luận, hoặc "deleted" nếu tài khoản bị xóa.

**Cột comment\_score**: Điểm số của bình luận, tính từ số upvotes trừ downvotes, thể hiện mức độ tương tác.

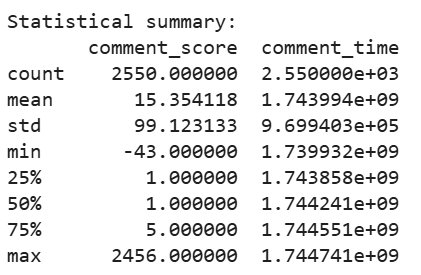
**Cột comment\_time**: Thời gian đăng bình luận dưới dạng timestamp, giúp phân tích sự thay đổi cảm nghĩ theo thời gian.



***Một số bản ghi trong tập dữ liệu***

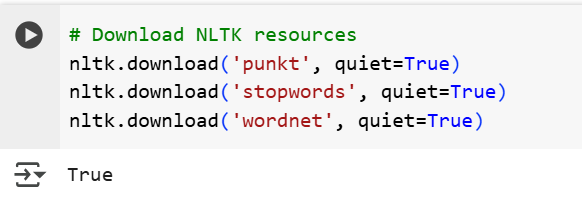


***Thông tin chi tiết về bộ dữ liệu***



***Thống kê mô tả về tập dữ liệu***

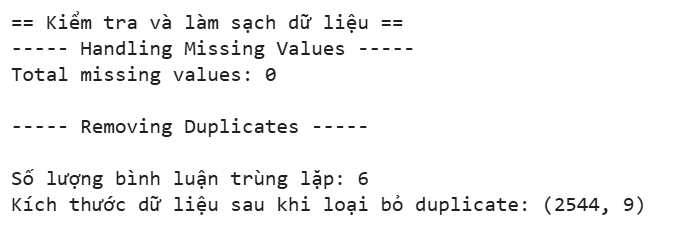
### ***2. Tiền xử lý dữ liệu.***



2.1. Kiểm tra và làm sạch dữ liệu.

Đầu tiên, chúng tôi tiến hành kiểm tra và làm sạch dữ liệu để đảm bảo rằng bộ dữ liệu không chứa các giá trị thiếu (null) hoặc bình luận trùng lặp:

* **Xử lý giá trị thiếu**: Chúng tôi kiểm tra các cột dữ liệu, đặc biệt là cột comment\_body, để tìm các bình luận bị thiếu. Các dòng có bình luận bị thiếu sẽ được loại bỏ.
* **Loại bỏ dữ liệu trùng lặp**: Tiếp theo, chúng tôi kiểm tra và loại bỏ các bình luận trùng lặp trong cột comment\_body, giúp đảm bảo tính đa dạng và độ chính xác của bộ dữ liệu.

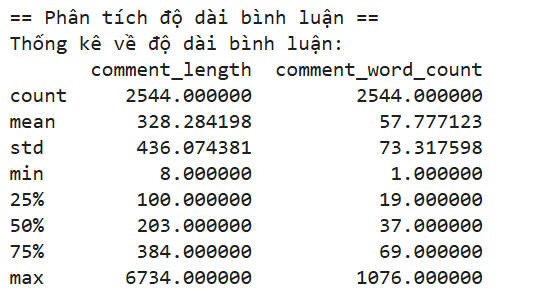


***Kết quả làm sạch dữ liệu***

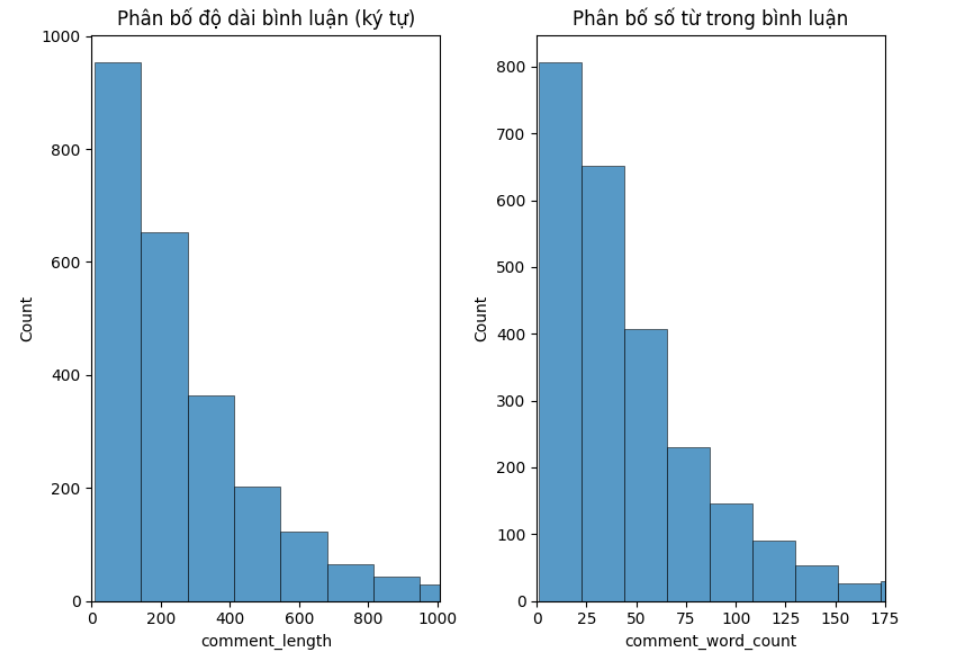
2.2. Phân tích sơ bộ về độ dài bình luận.

Sau khi làm sạch dữ liệu, chúng tôi thực hiện phân tích độ dài các bình luận:

* **Đo lường độ dài**: Chúng tôi thêm hai cột mới vào dữ liệu: comment\_length (số ký tự trong bình luận) và comment\_word\_count (số từ trong bình luận). Sau đó, chúng tôi thống kê về độ dài của các bình luận.
* **Vẽ biểu đồ phân bố độ dài**: Biểu đồ histogram được sử dụng để hiển thị phân bố độ dài bình luận dưới dạng số ký tự và số từ. Điều này giúp phân tích các bình luận có độ dài quá ngắn hoặc quá dài, để có thể điều chỉnh nếu cần.



***Kết quả thống kê về độ dài bình luận***



***Biểu đồ histogram phân bố độ dài***

Phân tích biểu đồ: Biểu đồ phân bố độ dài bình luận (theo số ký tự và số từ) cho thấy:

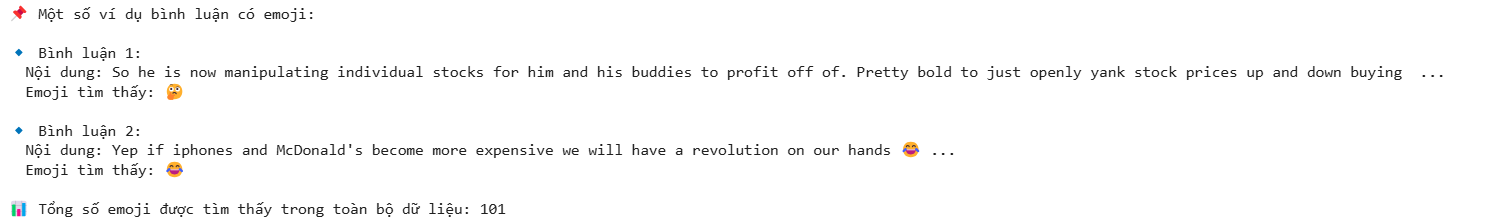
* **Biểu đồ độ dài bình luận (ký tự)**: Phần lớn bình luận có độ dài dưới 200 ký tự, với một số ít bình luận có độ dài dài hơn. Biểu đồ có phân bố không đồng đều với sự tập trung ở phần đầu của trục x.
* **Biểu đồ số từ trong bình luận**: Tương tự, phần lớn bình luận có ít hơn 50 từ. Biểu đồ cũng cho thấy sự phân bố chủ yếu ở các bình luận ngắn, với một số bình luận dài hơn nhưng ít.

Cả hai biểu đồ đều cho thấy rằng phần lớn bình luận trong dữ liệu có độ dài và số từ tương đối ngắn.

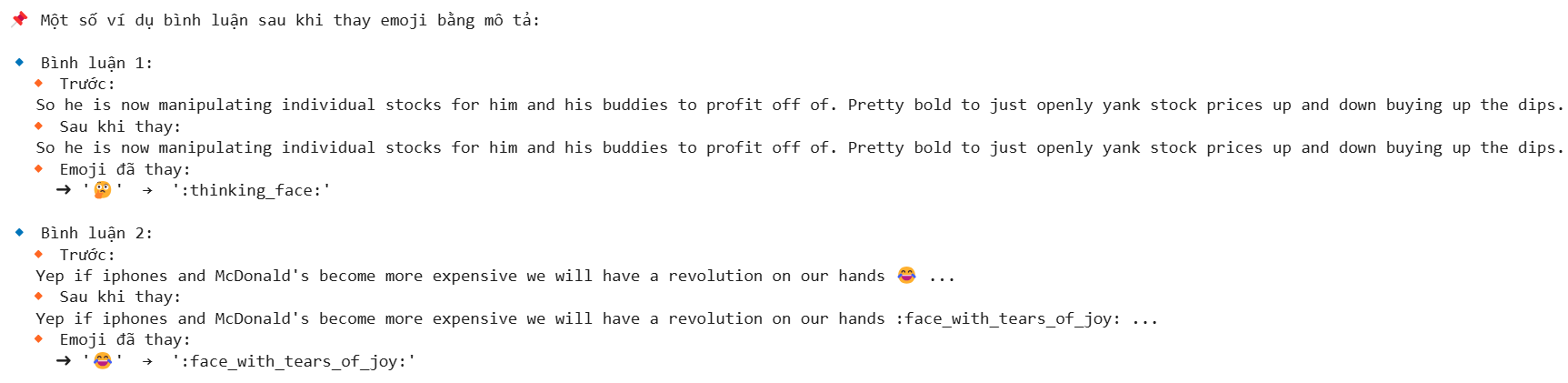
2.3. Chuyển đổi emoji.

Một phần quan trọng trong việc xử lý dữ liệu là chuyển đổi các emoji:

* **Trích xuất emoji**: Đầu tiên, chúng tôi trích xuất các emoji từ các bình luận và lưu chúng dưới dạng danh sách. Điều này giúp dễ dàng quan sát và phân tích các bình luận có emoji.
* **Thay thế emoji bằng tên mô tả**: Để dễ dàng xử lý và phân tích, chúng tôi thay thế các emoji trong bình luận bằng các từ mô tả tương ứng (ví dụ: 😊 → :smiling\_face\_with\_smiling\_eyes:). Quá trình này giúp chuyển emoji thành các yếu tố văn bản, có thể phân tích trong các mô hình cảm xúc.



***Hiển thị một vài mẫu cùng emoji tương ứng***

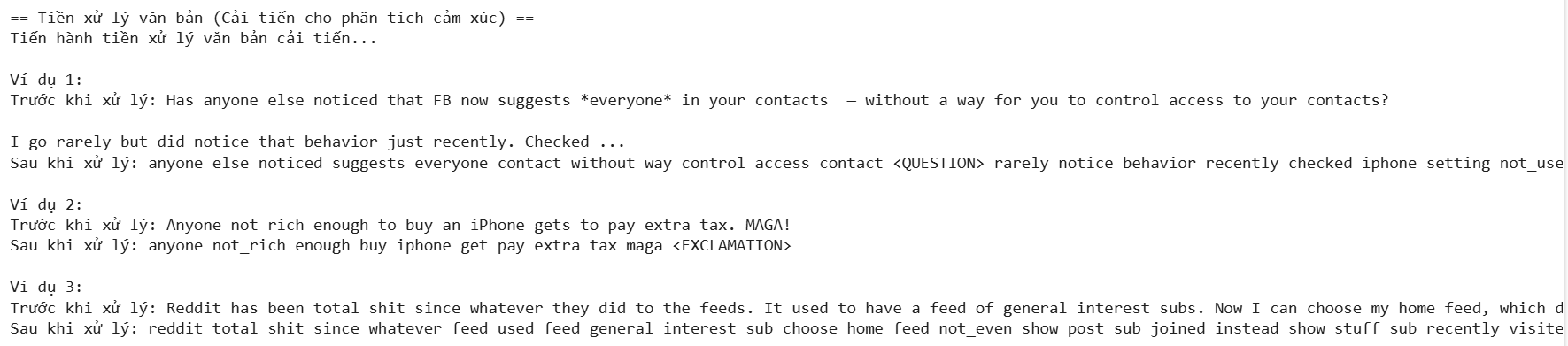


***Hiển thị kết quả trước và sau khi chuyển emoji thành các yếu tố văn bản***

2.4. Tiền xử lý văn bản - ENHANCED FOR SENTIMENT ANALYSIS.

Trong bước này, chúng tôi thực hiện các bước tiền xử lý văn bản để chuẩn bị dữ liệu cho phân tích cảm xúc:

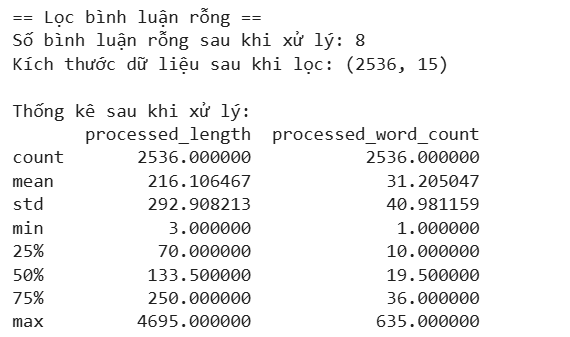
* **Xử lý contraction:** Thêm bước mở rộng các contraction như "can't" → "cannot", "won't" → "will not", "I'm" → "I am"... Điều này giúp giữ nghĩa phủ định tốt hơn.
* **Xử lý từ phủ định**: Các từ phủ định như "not", "no", "never" được giữ lại và kết hợp với từ tiếp theo bằng dấu "\_" (ví dụ: "not good" → "not\_good"). Điều này giúp giữ nguyên nghĩa trong các câu có từ phủ định, giúp mô hình phân tích cảm xúc nhận diện chính xác.
* **Loại bỏ stopwords**: Các từ dừng (stopwords) như "the", "and", "is" thường không mang nhiều ý nghĩa và sẽ bị loại bỏ. Tuy nhiên, các từ phủ định vẫn được giữ lại để không làm mất nghĩa trong câu.
* **Xử lý từ**: Chúng tôi thực hiện **lemmatization**, chuyển các từ về dạng gốc (ví dụ: "running" → "run"), giúp giảm sự đa dạng của từ ngữ trong văn bản và cải thiện độ chính xác trong phân tích cảm xúc.
* **Xử lý các yếu tố đặc biệt**:
* **Loại bỏ URLs**: Các liên kết web (http, https) không mang giá trị cảm xúc và sẽ bị loại bỏ.
* **Loại bỏ tài khoản người dùng**: Tài khoản người dùng (@username) sẽ bị loại bỏ vì không cần thiết cho phân tích cảm xúc.
* **Thay thế số**: Các số sẽ được thay thế bằng <NUM> để tránh phân biệt các giá trị số cụ thể.
* **Loại bỏ HTML tags**: Các thẻ HTML sẽ bị loại bỏ khỏi văn bản.
* **Tokenization**: Văn bản được chia nhỏ thành các từ (tokens), giúp chuẩn bị dữ liệu cho các bước phân tích tiếp theo. Tokenization giúp tách biệt các từ và đơn vị ngữ nghĩa để dễ dàng xử lý.



***Hiển thị kết quả trước và sau khi chuẩn hóa comment***

2.5. Lọc các bình luận rỗng sau khi xử lý.

* Sau khi tiền xử lý, chúng tôi loại bỏ các bình luận rỗng (những bình luận không có nội dung sau khi xử lý) để đảm bảo dữ liệu có chất lượng. Đồng thời, chúng tôi kiểm tra độ dài và số từ của các bình luận còn lại để đảm bảo tính đầy đủ và sẵn sàng cho phân tích cảm xúc.

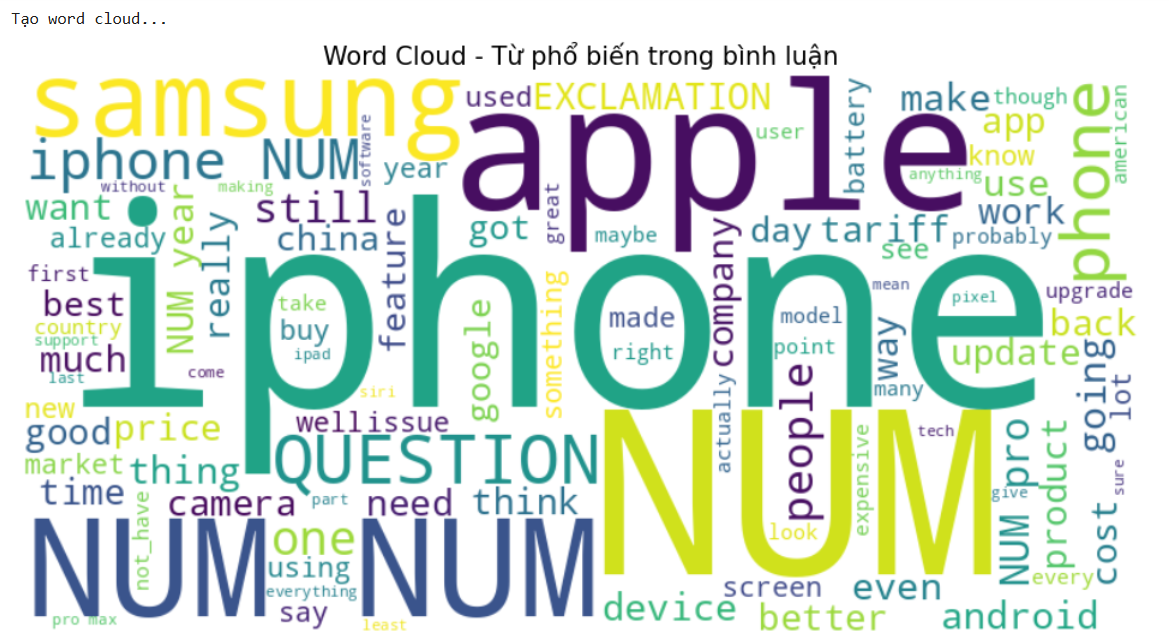


***Thống kê mô tả sau khi loại bỏ bình luận rỗng***

2.6. Phân tích các từ phổ biến.

- Trong bước này, chúng tôi thực hiện phân tích để tìm ra các từ phổ biến nhất trong bộ dữ liệu bình luận:

* **Tính tần suất xuất hiện của các từ**: Chúng tôi tính toán số lần xuất hiện của các từ trong toàn bộ các bình luận đã được xử lý. Dữ liệu này giúp xác định các từ được nhắc đến nhiều nhất.
* **Hiển thị các từ phổ biến**: Chúng tôi liệt kê 20 từ phổ biến nhất cùng với tần suất xuất hiện của chúng trong bộ dữ liệu.
* **Tạo Word Cloud**: Chúng tôi sử dụng **Word Cloud** để trực quan hóa các từ phổ biến, giúp dễ dàng nhận diện các từ quan trọng trong bộ dữ liệu. Word Cloud sẽ hiển thị các từ có tần suất cao với kích thước lớn hơn.



***Biểu đồ Word Cloud hiển thị các từ phổ biến nhất***

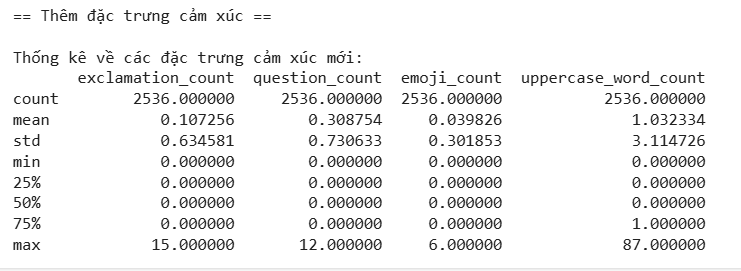
- Phân tích biểu đồ:

* **Các từ nổi bật**: "iphone", "apple", "phone", "samsung", "NUM", "question", "feature", "battery", "price" là những từ xuất hiện nhiều nhất, cho thấy người dùng chủ yếu thảo luận về các sản phẩm như **iPhone** và **Samsung**, các tính năng của điện thoại, giá cả và các vấn đề liên quan đến **pin**.
* **Các từ đặc biệt**: "NUM" xuất hiện nhiều lần, có thể là đại diện cho các giá trị số (có thể là giá, năm, v.v.) trong bình luận.
* **Cảm xúc và chủ đề chính**: Các từ như "question", "need", "think" cho thấy người dùng có thể đang cân nhắc và so sánh các sản phẩm. Các từ "company", "upgrade", "device" liên quan đến các yếu tố kỹ thuật và nâng cấp sản phẩm.

2.7. Thêm các đặc trưng cảm xúc mới.

- Để tăng cường khả năng phân tích cảm xúc, chúng tôi tạo thêm một số đặc trưng mới từ các bình luận:

* **Đếm số dấu chấm than và dấu hỏi**: Đếm số lượng dấu chấm than (!) và dấu hỏi (?) trong mỗi bình luận, vì chúng có thể chỉ ra sự nhấn mạnh hoặc sự nghi vấn trong cảm xúc của người dùng.
* **Đếm số lượng emoji**: Tổng số emoji trong mỗi bình luận được đếm và lưu vào một cột mới, vì emoji có thể mang thông tin cảm xúc quan trọng.
* **Đếm số từ in hoa**: Các từ in hoa được đếm vì chúng có thể thể hiện sự nhấn mạnh trong câu.



2.8. Lưu dữ liệu đã tiền xử lý.

- Cuối cùng, dữ liệu đã được xử lý được lưu lại vào một file CSV mới để sử dụng trong các bước phân tích tiếp theo. File này bao gồm các cột dữ liệu gốc và các cột mới chứa các đặc trưng cảm xúc như số lượng dấu chấm than, dấu hỏi, emoji, và số từ in hoa.

| # Chuẩn bị các cột để lưu  columns\_to\_save = ['brand', 'subreddit', 'post\_id', 'post\_title', 'comment\_id',  'comment\_body', 'comment\_score', 'comment\_time', 'comment\_author',  'processed\_comment', 'processed\_word\_count',  'exclamation\_count', 'question\_count', 'emoji\_count', 'uppercase\_word\_count']    output\_df = df[columns\_to\_save].copy()  # Lưu với tên khác để phân biệt  output\_df.to\_csv('/content/drive/MyDrive/'"BTL\_Data Mining"'/Dataset/preprocessed\_reddit\_comments\_enhanced.csv', index=False)  **print**("\nEnhanced preprocessed data saved to 'preprocessed\_reddit\_comments\_enhanced.csv'") |
| --- |

2.9. Gán nhãn dữ liệu sử dụng VADER.

- Trong bước này, chúng tôi sử dụng công cụ **VADER Sentiment Analyzer** từ thư viện **NLTK** để phân tích cảm xúc của các bình luận và gán nhãn cảm xúc cho mỗi bình luận. Quy trình thực hiện như sau:

* **Tính điểm cảm xúc (Sentiment Score)**:

Chúng tôi sử dụng **VADER SentimentIntensityAnalyzer** để tính điểm cảm xúc (sentiment score) cho mỗi bình luận. Điểm này dao động từ -1 (tiêu cực) đến 1 (tích cực), trong đó 0 thể hiện sự trung tính.

* **Gán nhãn cảm xúc**:  
  Dựa trên điểm cảm xúc, chúng tôi gán nhãn cho mỗi bình luận:  
  + **Positive**: Nếu điểm cảm xúc ≥ 0.05 (tích cực).
  + **Negative**: Nếu điểm cảm xúc ≤ -0.05 (tiêu cực).
  + **Neutral**: Nếu điểm cảm xúc nằm trong khoảng từ -0.05 đến 0.05 (trung tính).
* **Áp dụng cho bộ dữ liệu**:

Hàm phân tích cảm xúc và gán nhãn được áp dụng cho tất cả các bình luận trong cột processed\_comment.

* **Kiểm tra kết quả**:

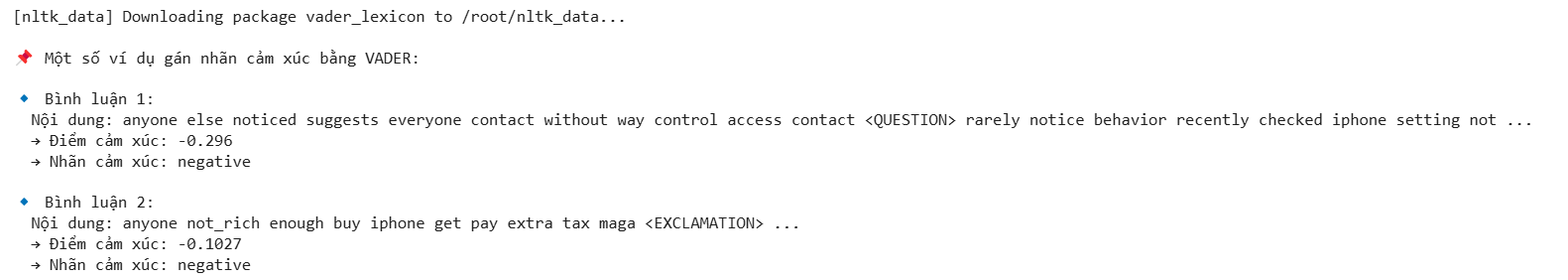
Một số ví dụ về bình luận, điểm cảm xúc và nhãn cảm xúc được in ra để kiểm tra kết quả.

* **Phân bố điểm cảm xúc**:

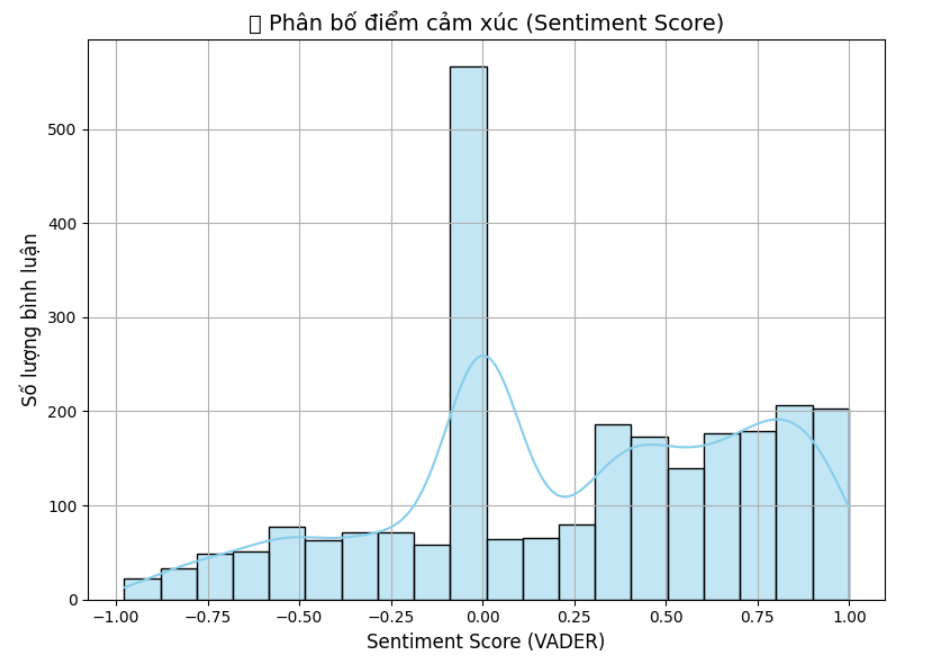
Chúng tôi tạo biểu đồ phân bố điểm cảm xúc (sentiment score) để trực quan hóa sự phân bố của các bình luận theo các mức độ cảm xúc.

* **Lưu dữ liệu đã gán nhãn**:

Sau khi gán nhãn cảm xúc, dữ liệu được lưu vào file CSV mới để sử dụng cho các bước phân tích tiếp theo.



***Một vài mẫu minh họa cho điểm cảm xúc và nhãn tương ứng***



***Biểu đồ phân bố điểm cảm xúc (sentiment score)***

- Phân tích biểu đồ: Biểu đồ phân bố điểm cảm xúc cho thấy:

* **Đỉnh ở 0**: Phần lớn bình luận có cảm xúc **trung tính** (gần 0), cho thấy sự chia sẻ quan điểm mà không có cảm xúc mạnh mẽ.
* **Phân bố tích cực và tiêu cực**: Các bình luận tích cực và tiêu cực có số lượng ít hơn, phân bố đều sau điểm 0.
* **Tần suất bình luận**: Hầu hết bình luận có điểm cảm xúc gần 0, với ít bình luận rõ ràng tích cực hoặc tiêu cực.

**⇒ Kết luận:** Biểu đồ này cho thấy phần lớn các bình luận trong bộ dữ liệu có cảm xúc **trung tính** với ít bình luận rõ ràng tích cực hoặc tiêu cực. Điều này có thể phản ánh một sự phân bố cảm xúc cân bằng, nơi người dùng chia sẻ quan điểm mà không có cảm xúc quá mạnh mẽ.

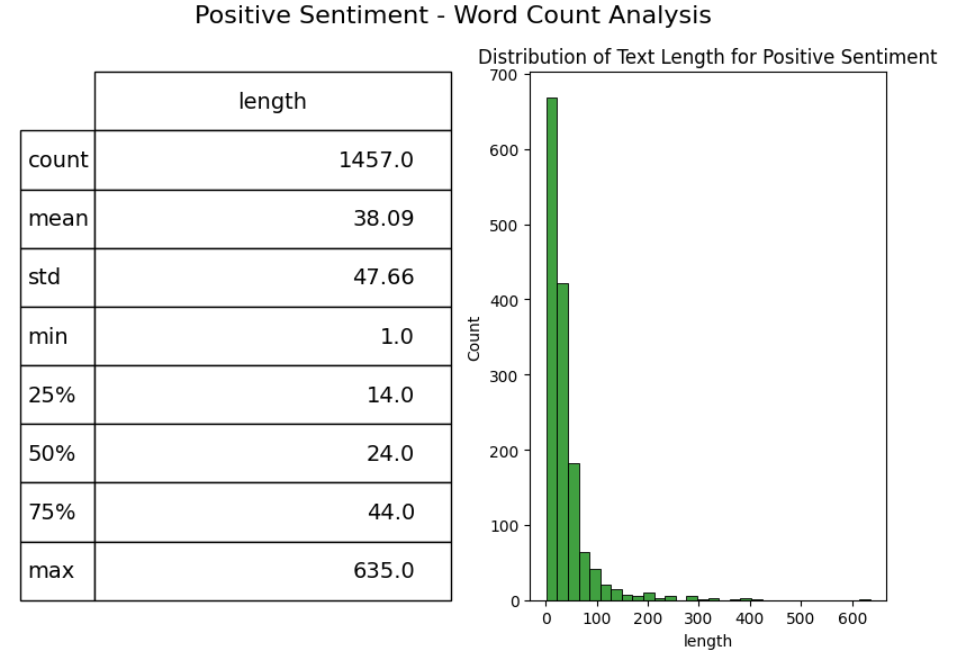


***Số lượng bình luận của mỗi nhãn tương ứng***

⇒Lưu dữ liệu gán nhãn

| # Lưu dữ liệu đã gán nhãn ra file CSV mới  df.to\_csv('/content/drive/MyDrive/'"BTL\_Data Mining"'/Dataset/reddit\_comments\_with\_sentiment.csv', index=False, encoding='utf-8-sig')  **print**("✅ Đã lưu dữ liệu đã gán nhãn vào file reddit\_comments\_with\_sentiment.csv") |
| --- |

2.10. EDA.



***Biểu đồ phân phối độ dài văn bản cho dữ liệu tình cảm tích cực***

### **Phân tích:**

* **Tần suất xuất hiện:**

Phần lớn các bình luận có độ dài rất ngắn, với **hơn 600 bình luận** có độ dài chỉ khoảng 10 từ hoặc ít hơn. Điều này cho thấy người dùng thường có xu hướng chia sẻ ý kiến ngắn gọn khi có cảm xúc tích cực.

* **Độ dài bình luận**:

**Trung bình**: Độ dài trung bình của các bình luận tích cực là khoảng **38 từ**, nhưng độ dài có sự phân tán khá lớn, với độ lệch chuẩn là **47.66**, cho thấy có những bình luận rất ngắn và cũng có những bình luận rất dài.

* **Phân phối không đều**:

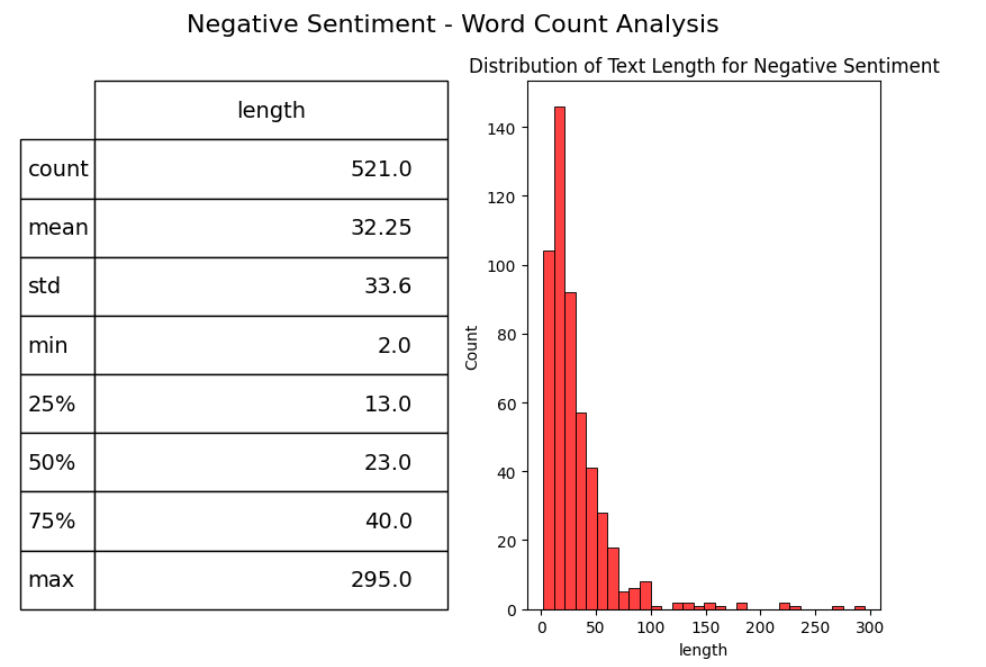
Biểu đồ có dạng phân phối lệch, với một đỉnh lớn ở phía bên trái, nghĩa là hầu hết các bình luận tích cực có độ dài ngắn. Sự phân bố này cho thấy người dùng có thể phản hồi nhanh hoặc đưa ra những cảm xúc tích cực một cách ngắn gọn.

* **Giá trị ngoại lệ**:

Một số bình luận có độ dài rất lớn, lên đến **635 từ** (theo giá trị **max**), nhưng số lượng này là rất ít, chỉ một vài bình luận.

### **Kết luận:**

Biểu đồ cho thấy phần lớn các bình luận tích cực trong dữ liệu có độ dài ngắn, nhưng có một số ít bình luận rất dài. Điều này có thể phản ánh xu hướng người dùng đưa ra nhận xét ngắn gọn và tích cực về sản phẩm hoặc chủ đề, mặc dù cũng có những bình luận dài để mô tả chi tiết cảm xúc tích cực của họ.



***Biểu đồ phân phối độ dài văn bản cho dữ liệu tình cảm tiêu cực***

### **Phân tích:**

* **Tần suất xuất hiện**:

Phần lớn các bình luận tiêu cực có **độ dài ngắn**, với khoảng **140 bình luận** có độ dài dưới 20 từ. Điều này chỉ ra rằng người dùng có xu hướng phản hồi ngắn gọn khi có cảm xúc tiêu cực.

* **Độ dài bình luận**:

**Trung bình**: Độ dài trung bình của các bình luận tiêu cực là **32.25 từ**, với độ lệch chuẩn là **33.6**, cho thấy một số bình luận rất ngắn và một số khác khá dài.

* **Phân phối không đều**:

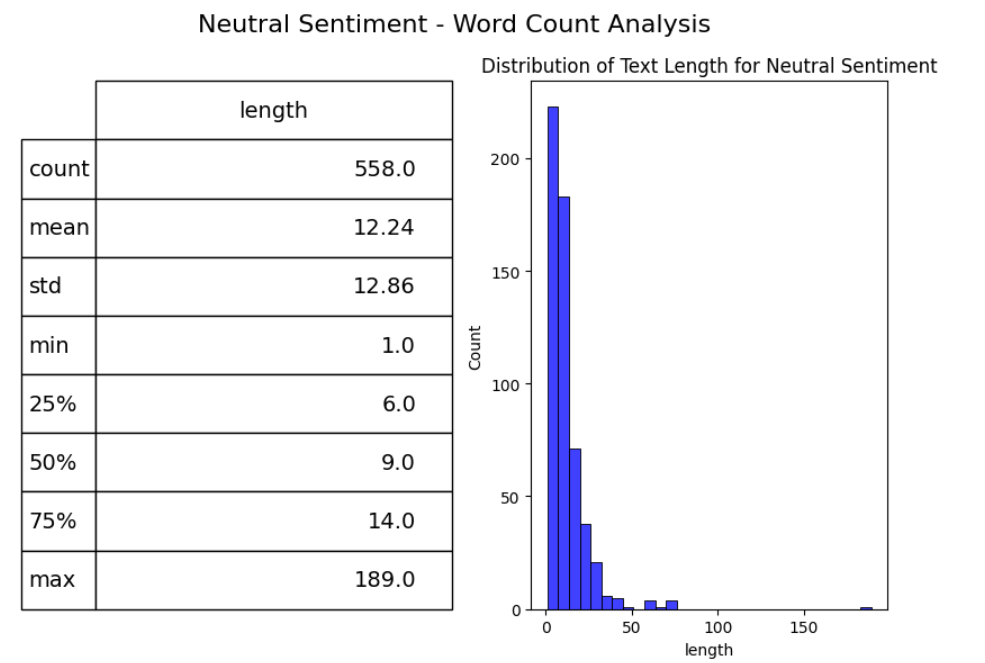
Biểu đồ có sự phân bố lệch, với đa số các bình luận có độ dài ngắn, nhưng vẫn có một số bình luận có độ dài lớn hơn, lên tới **295 từ** (theo giá trị **max**), mặc dù số lượng này rất ít.

* **Giá trị ngoại lệ**:

Một số bình luận có độ dài rất lớn, nhưng số lượng này không đáng kể so với phần còn lại.

### **Kết luận:**

Biểu đồ cho thấy phần lớn các bình luận tiêu cực có độ dài ngắn, nhưng vẫn tồn tại một số bình luận dài hơn. Điều này có thể phản ánh rằng người dùng thường chia sẻ cảm xúc tiêu cực ngắn gọn, mặc dù có những bình luận dài khi cần giải thích chi tiết cảm xúc của họ.



***Biểu đồ phân phối độ dài văn bản cho dữ liệu tình cảm trung lập***

### **Phân tích:**

* **Tần suất xuất hiện**:

Phần lớn các bình luận trung lập có **độ dài ngắn**, với khoảng **200 bình luận** có độ dài dưới 10 từ. Điều này chỉ ra rằng người dùng thường có xu hướng chia sẻ quan điểm trung lập một cách ngắn gọn.

* **Độ dài bình luận**:

**Trung bình**: Độ dài trung bình của các bình luận trung lập là **12.24 từ**, với độ lệch chuẩn là **12.86**, cho thấy phần lớn bình luận có độ dài khá ngắn nhưng cũng có sự phân tán về độ dài.

* **Phân phối không đều**:

Biểu đồ cho thấy một đỉnh lớn ở phần độ dài ngắn (dưới 50 từ) và ít bình luận dài. Sự phân bố này cho thấy hầu hết các bình luận trung lập rất ngắn gọn.

* **Giá trị ngoại lệ**:

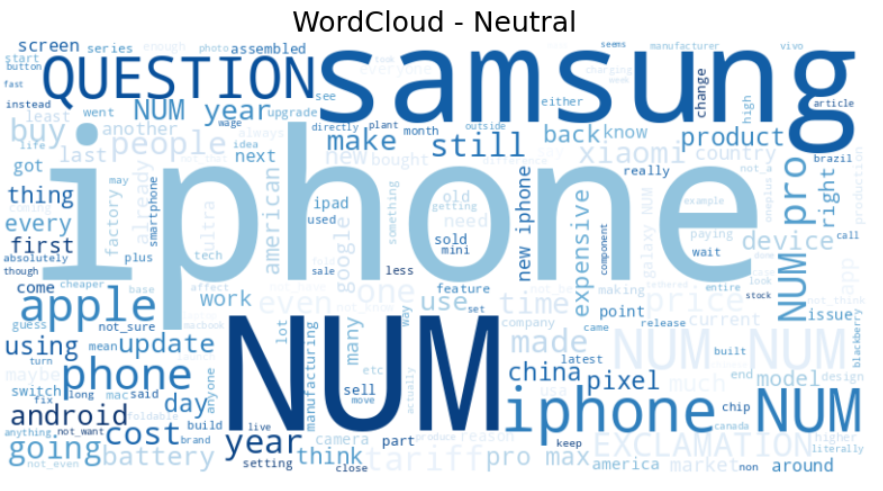
Một số bình luận có độ dài rất lớn (**189 từ**), nhưng đây là các giá trị ngoại lệ, ít gặp.

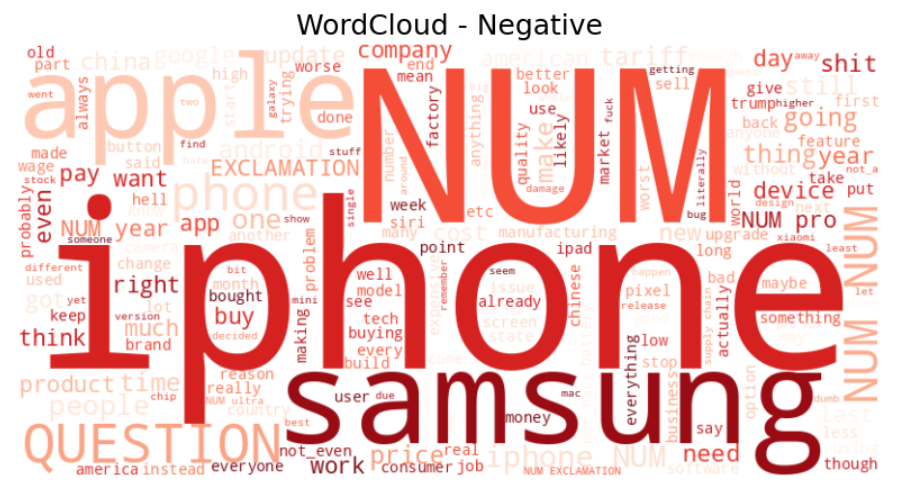
### **Kết luận:**

Biểu đồ cho thấy phần lớn các bình luận trung lập trong dữ liệu có độ dài ngắn, với một số ít bình luận dài. Điều này phản ánh xu hướng người dùng chia sẻ quan điểm trung lập một cách nhanh chóng và ngắn gọn.

Plot WordCloud







**Phân tích:**

### **Cảm xúc tích cực (Positive Sentiment):**

* **Từ phổ biến**: "iPhone", "apple", "phone", "samsung", "battery" là những từ xuất hiện nhiều nhất, cho thấy người dùng đang thảo luận về các sản phẩm **iPhone**, **apple** và các tính năng như **pin**.
* **Cảm xúc**: Các từ như "good", "great", "love" (yêu thích) thể hiện cảm xúc tích cực rõ ràng, cho thấy người dùng chủ yếu có cảm nhận tốt về các sản phẩm.

### **Cảm xúc trung lập (Neutral Sentiment):**

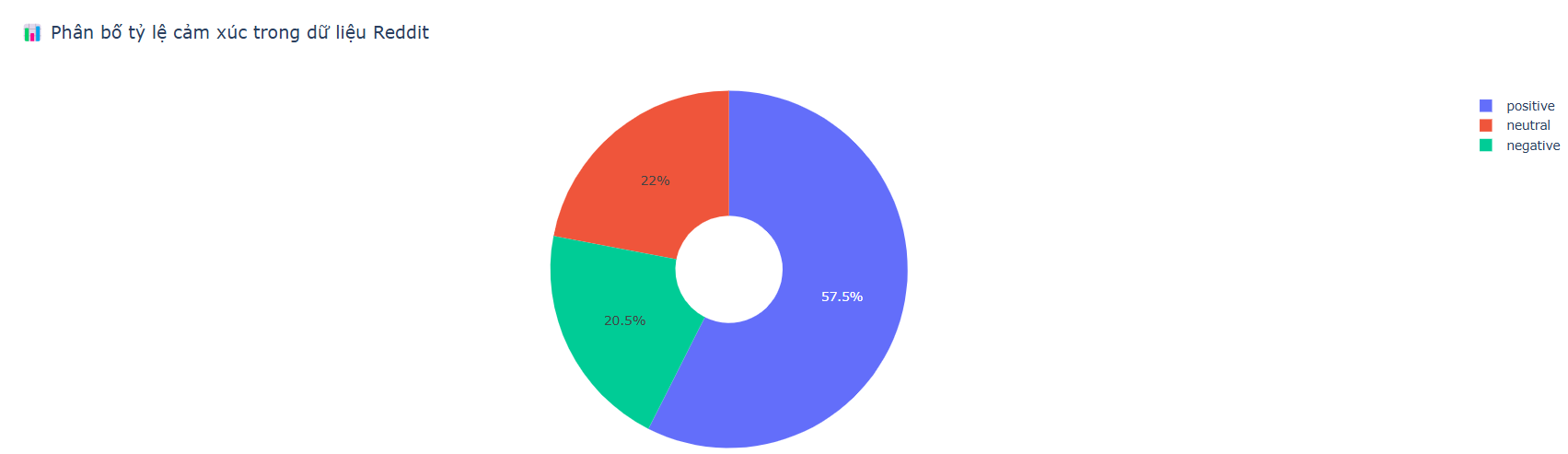
* **Từ phổ biến**: "iPhone", "samsung", "question", "model", "phone" đều xuất hiện nổi bật, với từ "question" cho thấy người dùng đang thảo luận, so sánh hoặc có thắc mắc về sản phẩm.
* **Cảm xúc**: Các từ như "maybe", "expensive", "buy" phản ánh sự phân vân hoặc cân nhắc, cho thấy các bình luận chủ yếu là không có cảm xúc mạnh mẽ và chứa đựng những suy nghĩ, câu hỏi.

### **Cảm xúc tiêu cực (Negative Sentiment):**

* **Từ phổ biến**: "iPhone", "samsung", "phone", "apple", "buy" xuất hiện nhiều, nhưng cũng có từ như "shit", "worse", "problem", "need" cho thấy những cảm xúc tiêu cực liên quan đến các sản phẩm.
* **Cảm xúc**: Các từ như "bad", "worst", "disappointed" phản ánh sự không hài lòng và các vấn đề mà người dùng gặp phải, cho thấy các bình luận mang tính chỉ trích và phàn nàn.

### **Kết luận:**

* **Cảm xúc tích cực**: Chủ yếu liên quan đến sự hài lòng với các sản phẩm như **iPhone** và **Apple**.
* **Cảm xúc trung lập**: Chủ yếu là thảo luận và so sánh giữa các sản phẩm, không có cảm xúc rõ ràng.
* **Cảm xúc tiêu cực**: Liên quan đến sự thất vọng và chỉ trích các sản phẩm, đặc biệt là **iPhone** và **Samsung**.



***Biểu đồ hình tròn về các cảm nhận khác nhau của dữ liệu***

### **Phân tích:**

* **Cảm xúc tích cực (Positive)**:
* **Tỷ lệ**: **57.5%** của các bình luận thuộc loại cảm xúc tích cực.
* **Nhận xét**: Đây là phần lớn bình luận, cho thấy người dùng trên Reddit có xu hướng chia sẻ các cảm xúc tích cực về sản phẩm hoặc chủ đề đang thảo luận.
* **Cảm xúc trung lập (Neutral)**:
* **Tỷ lệ**: **20.5%** của các bình luận có cảm xúc trung lập.
* **Nhận xét**: Các bình luận trung lập chiếm một tỷ lệ khá nhỏ, cho thấy người dùng ít khi đưa ra những quan điểm hoàn toàn trung lập mà không có cảm xúc rõ ràng.
* **Cảm xúc tiêu cực (Negative)**:
* **Tỷ lệ**: **22%** của các bình luận mang cảm xúc tiêu cực.
* **Nhận xét**: Mặc dù tỷ lệ này nhỏ hơn cảm xúc tích cực, nhưng vẫn có một phần người dùng phàn nàn hoặc không hài lòng về sản phẩm hoặc dịch vụ.

### **Kết luận:**

Biểu đồ cho thấy phần lớn các bình luận trên Reddit có cảm xúc tích cực, với một tỷ lệ nhỏ cảm xúc tiêu cực và trung lập. Điều này có thể phản ánh xu hướng chung của người dùng khi tham gia vào các thảo luận trực tuyến về sản phẩm hoặc chủ đề cụ thể.

### ***3. Trích Xuất Đặc Trưng.***

Trong bước này, chúng tôi đã sử dụng các phương pháp khác nhau để trích xuất đặc trưng từ dữ liệu văn bản nhằm xây dựng các mô hình phân tích cảm xúc:

1. **Label Encoding**:

Đầu tiên, chúng tôi sử dụng **LabelEncoder** để mã hóa nhãn cảm xúc (positive, neutral, negative) thành các giá trị số (0, 1, 2) để dễ dàng xử lý trong mô hình.

1. **Bag of Words (BoW)**:

**BoW** được sử dụng để trích xuất đặc trưng từ văn bản, bằng cách chuyển đổi mỗi bình luận thành một vector có độ dài cố định, biểu diễn tần suất xuất hiện của các từ trong bộ dữ liệu. Chúng tôi giới hạn số lượng từ tối đa là **3000 từ**.

1. **TF-IDF (Term Frequency-Inverse Document Frequency)**:

**TF-IDF** giúp đánh giá tầm quan trọng của mỗi từ trong bình luận, với các từ xuất hiện ít nhưng quan trọng trong tài liệu được ưu tiên hơn. Chúng tôi cũng sử dụng **3000 từ** tối đa cho phương pháp này.

1. **Handcrafted Features (Đặc trưng thủ công)**:

Ngoài các đặc trưng văn bản, chúng tôi còn tạo ra một số đặc trưng thủ công từ dữ liệu, bao gồm:

* **Length**: Độ dài của bình luận (số từ).
* **Number of Exclamation Marks**: Số dấu chấm than (<EXCLAMATION>) trong bình luận.
* **Number of Question Marks**: Số dấu hỏi (<QUESTION>) trong bình luận.
* **Number of Numeric Tokens**: Số token đại diện cho số (<NUM>).

1. **Kết hợp các đặc trưng**:

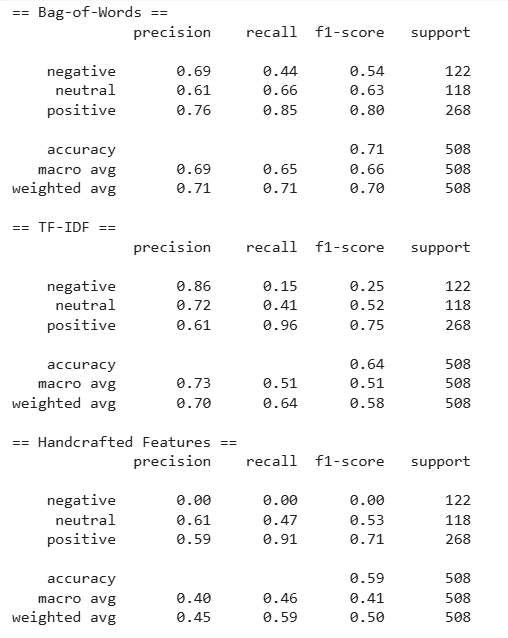
Cuối cùng, chúng tôi kết hợp các đặc trưng từ **TF-IDF** và **Handcrafted Features** vào một ma trận đặc trưng duy nhất, cũng như kết hợp **BoW** và **Handcrafted Features**.

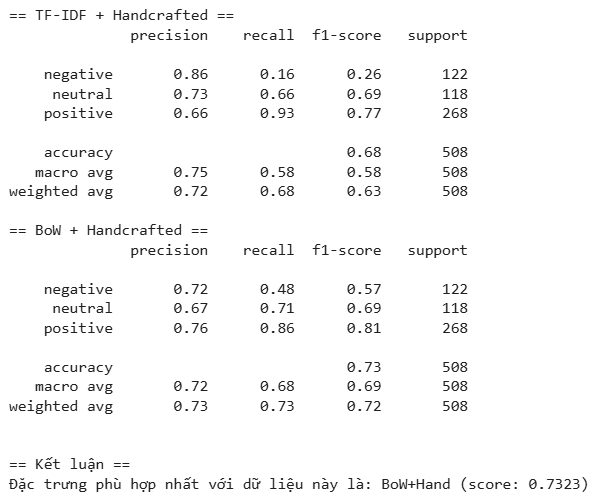
1. **Đánh giá mô hình**:

Sau khi trích xuất các đặc trưng, chúng tôi đã đánh giá các mô hình phân loại sử dụng **Logistic Regression** với từng loại đặc trưng:

* **Bag-of-Words (BoW)**
* **TF-IDF**
* **Đặc trưng thủ công**
* **TF-IDF kết hợp với đặc trưng thủ công**
* **BoW kết hợp với đặc trưng thủ công**

7. **Kết quả**:





⇒ Sau khi so sánh các phương pháp trích xuất đặc trưng, chúng tôi nhận thấy rằng **BoW kết hợp với các đặc trưng thủ công (BoW + Handcrafted)** là phương pháp hiệu quả nhất cho dữ liệu này, với **độ chính xác đạt 73.23%**. Phương pháp này mang lại sự cân bằng giữa việc trích xuất đặc trưng từ văn bản (BoW) và các đặc trưng thủ công, giúp cải thiện khả năng phân loại cảm xúc của mô hình.

### ***4. Huấn Luyện Mô Hình.***

#### 4.1. SVM (Support Vector Machine).

Ý tưởng thuật toán: Support Vector Machine (SVM) là một thuật toán học máy dùng để phân loại và hồi quy. Đặc biệt, SVM phân loại bằng cách tìm ra **siêu phẳng (hyperplane)** tối ưu phân chia dữ liệu thành các lớp khác nhau. Mục tiêu của SVM là **tìm siêu phẳng có khoảng cách lớn nhất** từ các điểm dữ liệu gần nhất (gọi là **các vector hỗ trợ - support vectors**). Điều này giúp mô hình phân loại có độ chính xác cao và có khả năng tổng quát tốt.

SVM có thể được mở rộng để xử lý các vấn đề phi tuyến bằng cách sử dụng các **kernel**, giúp chuyển dữ liệu vào không gian cao chiều, nơi các lớp có thể phân chia tuyến tính.

* Đầu vào:
* Một tập hợp các điểm dữ liệu có đặc trưng (features) và nhãn (labels).
* Dữ liệu có thể được biểu diễn dưới dạng một ma trận X với mỗi hàng là một điểm dữ liệu và mỗi cột là một đặc trưng.
* Mỗi điểm dữ liệu có một nhãn y (thường là 1 hoặc -1 trong phân loại nhị phân).
* Đầu ra:
* Một mô hình phân loại có các tham số trọng số w và bias b, có thể được sử dụng để dự đoán nhãn của các điểm dữ liệu mới.
* Các tham số này sẽ xác định **siêu phẳng tối ưu** trong không gian đặc trưng, phân chia các lớp dữ liệu.

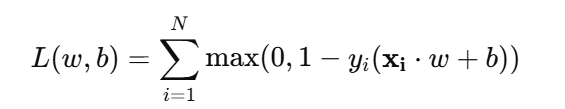
Quy trình thuật toán:

**Bước 1: Khởi tạo**

**Trọng số và bias**: Bắt đầu với giá trị ngẫu nhiên cho các trọng số **w** và bias **b**. Trọng số **w** xác định hướng của siêu phẳng, và bias **b** điều chỉnh vị trí của siêu phẳng đó.

**Bước 2: Tính toán hàm mất mát (Loss Function)**

**Hàm mất mát Hinge Loss** cho SVM được tính như sau:



Trong đó:

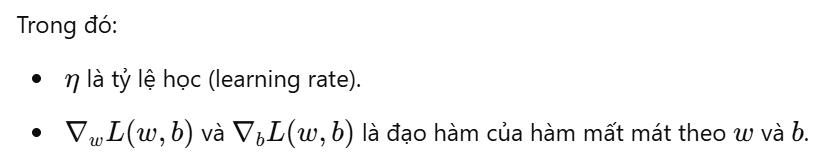
* yi​ là nhãn của điểm dữ liệu thứ i (giá trị 1 hoặc -1).
* xi​ là điểm dữ liệu thứ i
* w là trọng số của mô hình.
* b là bias.

Hàm mất mát này đảm bảo rằng các điểm dữ liệu gần siêu phẳng sẽ có mất mát lớn, và những điểm xa siêu phẳng sẽ không ảnh hưởng nhiều đến hàm mất mát.

**Bước 3: Tối ưu hóa hàm mất mát**

* Để tối ưu hóa hàm mất mát, SVM sử dụng **phương pháp tối ưu hóa** (thường là **Gradient Descent** hoặc **Quadratic Programming**).
* **Gradient Descent** sẽ tính toán gradient của hàm mất mát theo trọng số www và bias bbb, sau đó cập nhật chúng bằng cách di chuyển theo hướng của gradient với một bước học nhất định.  
    
   Cập nhật trọng số và bias theo gradient descent:



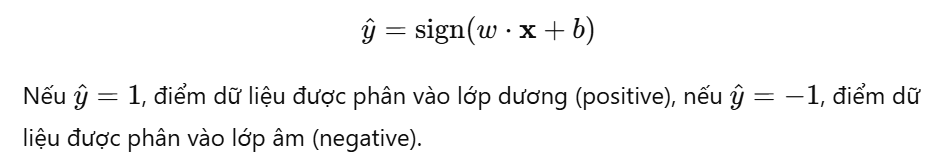


**Bước 4: Điều chỉnh trọng số và bias**

**Điều chỉnh w** và **bias b** liên tục cho đến khi đạt được điều kiện dừng, ví dụ như:

* Mất mát (loss) không thay đổi nhiều sau mỗi vòng lặp.
* Đạt số vòng lặp tối đa.

**Bước 5: Dự đoán nhãn cho dữ liệu mới**

Sau khi huấn luyện mô hình, bạn có thể sử dụng siêu phẳng đã tìm được để dự đoán nhãn cho các điểm dữ liệu mới. Dự đoán được tính bằng cách sử dụng công thứ

Triển khai code:

* Chuẩn bị Dữ liệu và Kiểm Tra Phân Phối Lớp:

| # Đảm bảo dữ liệu đầu vào là ndarray, không phải pandas Series  **if** isinstance(y\_train, pd.Series):  y\_train = y\_train.to\_numpy()  **if** isinstance(X\_train, pd.DataFrame):  X\_train = X\_train.to\_numpy()  **if** isinstance(X\_test, pd.DataFrame):  X\_test = X\_test.to\_numpy()  **if** isinstance(y\_test, pd.Series):  y\_test = y\_test.to\_numpy()    # Chuẩn hóa dữ liệu - quan trọng cho SVM  **def** normalize\_data(X):  X\_norm = X.copy()  **for** i **in** range(X.shape[1]):  feature\_max = np.max(np.abs(X[:, i]))  **if** feature\_max > 0:  X\_norm[:, i] = X[:, i] / feature\_max  **return** X\_norm    # Kiểm tra phân phối các lớp trong tập huấn luyện  unique\_classes, class\_counts = np.unique(y\_train\_sample, return\_counts=True)  **for** cls, count **in** zip(unique\_classes, class\_counts):  **print**(f"Lớp {cls}: {count} mẫu ({count/len(y\_train\_sample)\*100:.1f}%)") |
| --- |

* Định nghĩa Các Kernel cho SVM

| **def** linear\_kernel(x1, x2):  **return** np.dot(x1, x2)    **def** rbf\_kernel(x1, x2, gamma):  **return** np.exp(-gamma \* np.linalg.norm(x1 - x2)\*\*2)    **def** polynomial\_kernel(x1, x2, degree=3):  **return** (np.dot(x1, x2) + 1) \*\* degree |
| --- |

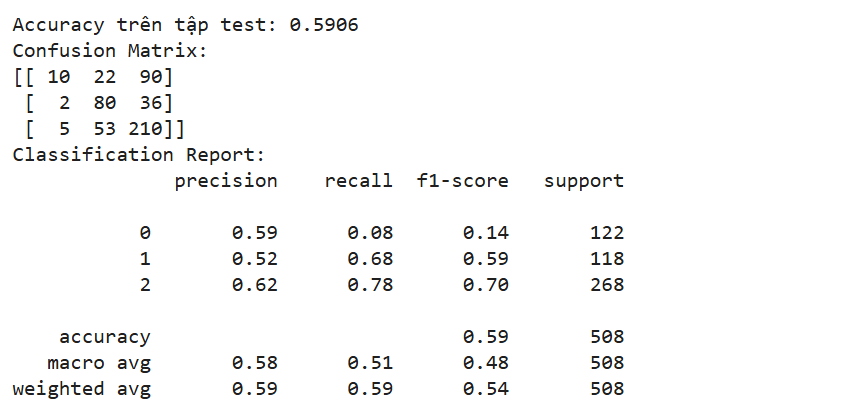
* Lớp SVM\_Manual\_Fast - Huấn Luyện và Dự Đoán:

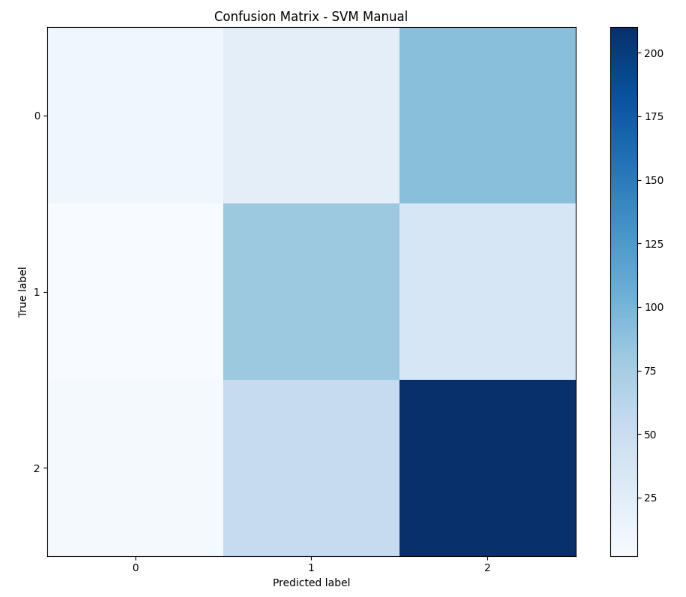
| **class** SVM\_Manual\_Fast:  **def** \_\_init\_\_(self, kernel='linear', C=1.0, gamma=0.1, degree=3, max\_iter=100, tol=1e-3, class\_weight=None):  self.kernel = kernel  self.C = C  self.gamma = gamma  self.degree = degree  self.max\_iter = max\_iter  self.tol = tol  self.classes = None  self.models = []  self.class\_weight = class\_weight # Thêm class\_weight    **def** \_kernel\_function(self, x1, x2):  **if** self.kernel == 'linear':  **return** linear\_kernel(x1, x2)  **elif** self.kernel == 'rbf':  **return** rbf\_kernel(x1, x2, self.gamma)  **elif** self.kernel == 'poly':  **return** polynomial\_kernel(x1, x2, self.degree)  **else**:  **raise** ValueError(f"Kernel không hợp lệ: {self.kernel}")    **def** fit(self, X, y):  # Huấn luyện mô hình SVM với One-vs-Rest approach  self.classes = np.unique(y)  n\_classes = len(self.classes)    # Tính toán kernel matrix một lần để tăng tốc  K = self.\_compute\_kernel\_matrix(X)    # Lặp qua từng lớp và huấn luyện  **for** i, c **in** enumerate(self.classes):  binary\_y = np.where(y == c, 1, -1)  # Cập nhật alpha và bias trong vòng lặp SMO  alphas = np.zeros(X.shape[0])  b = 0  # Thực hiện SMO cho việc huấn luyện  self.\_optimize(X, y, binary\_y, alphas, b, K)    **return** self    **def** predict(self, X):  # Dự đoán nhãn cho tập dữ liệu mới  decision\_values = np.zeros((X.shape[0], len(self.classes)))  **for** i, model **in** enumerate(self.models):  decision\_values[:, i] = self.\_decision\_function(X, model)    **return** self.classes[np.argmax(decision\_values, axis=1)] |
| --- |

* Tìm Các Tham Số Tối Ưu (Hyperparameter Tuning):

| **def** find\_best\_parameters\_fast(X\_train, y\_train, X\_val, y\_val):  best\_acc = 0  param\_grid = [  {'kernel': 'linear', 'C': 1.0, 'gamma': 0.1},  {'kernel': 'rbf', 'C': 1.0, 'gamma': 0.1},  {'kernel': 'rbf', 'C': 10.0, 'gamma': 0.01},  {'kernel': 'rbf', 'C': 100.0, 'gamma': 0.001}  ]    # Thử nghiệm với các tham số khác nhau  **for** params **in** param\_grid:  svm = SVM\_Manual\_Fast(  kernel=params['kernel'],  C=params['C'],  gamma=params['gamma'],  )  svm.fit(X\_train, y\_train)  y\_pred = svm.predict(X\_val)    # Tính toán độ chính xác và F1 score  acc = accuracy\_score(y\_val, y\_pred)  **print**(f"Accuracy: {acc:.4f}")  **if** acc > best\_acc:  best\_acc = acc  best\_params = params    **return** best\_params |
| --- |

**Kết quả:**



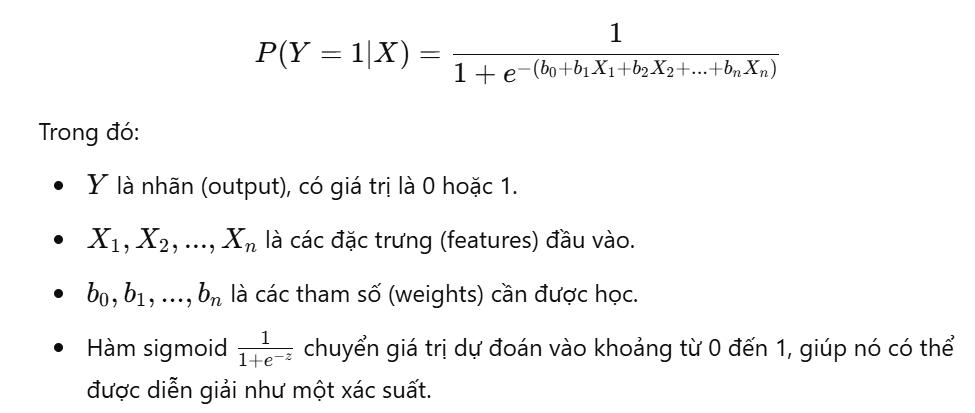


**- Đánh giá tổng quan:**

* **Điểm mạnh**:  
  + Mô hình làm tốt trên lớp 2 với **Recall** cao (78%) và **F1-Score** khá (0.70).
  + Có một số khả năng nhận diện lớp 1, với **Recall** khá cao (68%).
* **Điểm yếu**:  
  + **Lớp 0** có **Recall** rất thấp (8%), điều này cho thấy mô hình gặp khó khăn trong việc phân loại các điểm thuộc lớp này chính xác. Mô hình cần cải thiện khả năng nhận diện lớp này.
  + **Precision** của mô hình trên tất cả các lớp đều khá thấp, đặc biệt là lớp 1 (52%), điều này đồng nghĩa với việc mô hình phân vào các lớp không chính xác trong nhiều trường hợp.
  + **Macro average** và **F1-Score tổng thể** cho thấy mô hình không hoàn hảo và có thể cần cải thiện sự phân biệt giữa các lớp.

#### 4.2. Logistic Regression

* **Ý tưởng thuật toán**: Mục tiêu của Logistic Regression là dự đoán xác suất một đối tượng thuộc vào một trong hai lớp (0 hoặc 1). Các lớp này có thể đại diện cho bất kỳ khái niệm phân loại nào như có bệnh/không có bệnh, chấp nhận/không chấp nhận một yêu cầu, v.v.
* Trong Logistic Regression, mô hình học được một hàm số xác suất dưới dạng:



* Đầu vào: **Dữ liệu đầu vào**: Một tập hợp các đặc trưng (features) X=[X1,X2,...,Xn] cho mỗi đối tượng. Đây là các giá trị số đại diện cho các đặc điểm hoặc thuộc tính của đối tượng (ví dụ: chiều cao, cân nặng, tuổi, thu nhập...).
* **Nhãn (Label)**: Y, là giá trị cần dự đoán, thường có giá trị là 0 hoặc 1. Đây là lớp phân loại của đối tượng.

### **Đầu ra (Outputs)**

* **Xác suất**: Mô hình sẽ xuất ra một giá trị xác suất (từ 0 đến 1) cho mỗi đối tượng, thể hiện khả năng đối tượng thuộc lớp 1. Ví dụ, nếu đầu ra là 0.8, điều đó có nghĩa là có 80% xác suất đối tượng thuộc lớp 1.
* **Phân loại**: Dựa trên xác suất đầu ra, mô hình sẽ phân loại đối tượng vào một trong hai lớp. Thông thường, nếu xác suất lớn hơn 0.5, đối tượng sẽ được phân vào lớp 1, còn nếu nhỏ hơn 0.5, đối tượng sẽ được phân vào lớp 0.

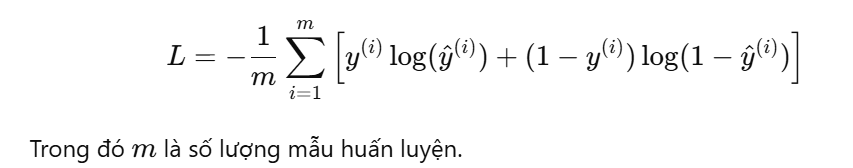
### **Các bước thuật toán**

**1. Khởi tạo tham số**: Khởi tạo các tham số (weights) b0,b1,...,bn​ ngẫu nhiên hoặc bằng 0.

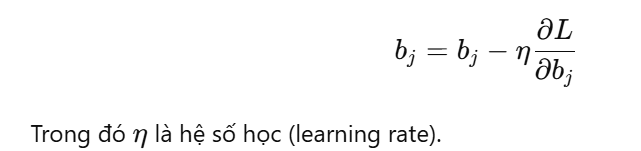
**2. Tính toán dự đoán**: Dựa trên các tham số và đặc trưng đầu vào X, tính toán giá trị dự đoán



**3. Tính toán hàm lỗi (Loss Function)**: Hàm lỗi trong Logistic Regression thường là hàm **Cross-Entropy Loss**:



**4. Cập nhật tham số**: Sử dụng thuật toán Gradient Descent để tối ưu hóa các tham số b0,b1,...,bn​. Cập nhật các tham số theo công thức:



**5. Lặp lại**: Tiếp tục quá trình tính toán và cập nhật tham số cho đến khi hàm lỗi đạt giá trị tối thiểu hoặc số vòng lặp đã được định sẵn.

Triển khai code:

* Tiền xử lý và chuẩn hóa dữ liệu:

| # Đảm bảo dữ liệu đầu vào là ndarray, không phải pandas Series  **if** isinstance(y\_train, pd.Series):  **print**("Chuyển đổi y\_train từ pandas Series sang numpy array")  y\_train = y\_train.to\_numpy()  **if** isinstance(X\_train, pd.DataFrame):  **print**("Chuyển đổi X\_train từ pandas DataFrame sang numpy array")  X\_train = X\_train.to\_numpy()  **if** isinstance(X\_test, pd.DataFrame):  **print**("Chuyển đổi X\_test từ pandas DataFrame sang numpy array")  X\_test = X\_test.to\_numpy()  **if** isinstance(y\_test, pd.Series):  **print**("Chuyển đổi y\_test từ pandas Series sang numpy array")  y\_test = y\_test.to\_numpy() |
| --- |

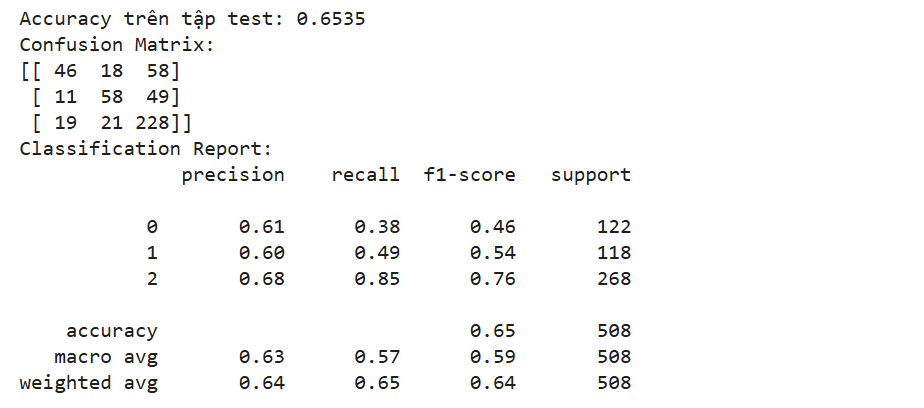
* Huấn luyện mô hình Logistic Regression (Mini-batch Gradient Descent)

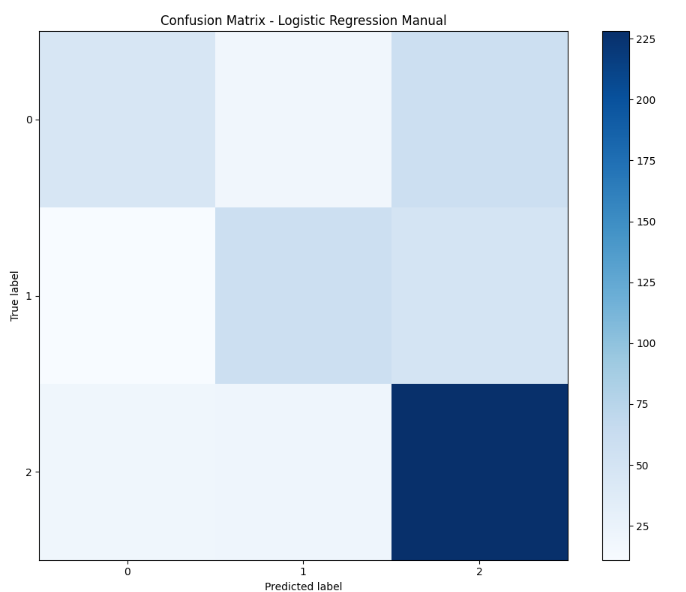
| **class** LogisticRegression\_Manual\_Fast:  **def** \_\_init\_\_(self, learning\_rate=0.01, max\_iter=500, tol=1e-4, C=1.0, batch\_size=128):  self.learning\_rate = learning\_rate  self.max\_iter = max\_iter  self.tol = tol  self.C = C # Regularization parameter  self.batch\_size = batch\_size # Mini-batch size  self.weights = None  self.bias = None  self.classes = None  self.feature\_means = None # Lưu trung bình của các đặc trưng  self.feature\_stds = None # Lưu độ lệch chuẩn của các đặc trưng    **def** sigmoid(self, z):  z = np.clip(z, -500, 500)  **return** 1 / (1 + np.exp(-z))    **def** softmax(self, z):  z = np.clip(z, -500, 500)  exp\_z = np.exp(z - np.max(z, axis=1, keepdims=True))  **return** exp\_z / np.sum(exp\_z, axis=1, keepdims=True)    **def** fit(self, X, y):  n\_samples, n\_features = X.shape  self.classes = np.unique(y)  n\_classes = len(self.classes)    # Lưu thông số chuẩn hóa  self.feature\_means = np.mean(X, axis=0)  self.feature\_stds = np.std(X, axis=0)    # Khởi tạo weights và bias  self.weights = np.zeros((n\_features, n\_classes))  self.bias = np.zeros(n\_classes)    # One-hot encoding cho y  y\_one\_hot = np.zeros((n\_samples, n\_classes))  **for** i, c **in** enumerate(self.classes):  y\_one\_hot[:, i] = (y == c).astype(int)    # Mini-batch Gradient Descent  n\_batches = max(1, n\_samples // self.batch\_size)  prev\_loss = float('inf')    **for** iteration **in** range(self.max\_iter):  indices = np.random.permutation(n\_samples)  X\_shuffled = X[indices]  y\_one\_hot\_shuffled = y\_one\_hot[indices]    total\_loss = 0    # Duyệt qua các mini-batch  **for** batch **in** range(n\_batches):  start\_idx = batch \* self.batch\_size  end\_idx = min((batch + 1) \* self.batch\_size, n\_samples)    X\_batch = X\_shuffled[start\_idx:end\_idx]  y\_batch = y\_one\_hot\_shuffled[start\_idx:end\_idx]  batch\_size = end\_idx - start\_idx    # Forward pass  linear\_output = np.dot(X\_batch, self.weights) + self.bias  y\_pred = self.softmax(linear\_output)    # Tính loss  batch\_loss = -np.mean(np.sum(y\_batch \* np.log(y\_pred + 1e-15), axis=1))  batch\_loss += (self.C / (2 \* batch\_size)) \* np.sum(self.weights \*\* 2)  total\_loss += batch\_loss \* batch\_size    # Backward pass - tính gradient  error = y\_pred - y\_batch    # Update weights và bias  dw = (1 / batch\_size) \* np.dot(X\_batch.T, error) + (self.C / batch\_size) \* self.weights  db = (1 / batch\_size) \* np.sum(error, axis=0)    self.weights -= self.learning\_rate \* dw  self.bias -= self.learning\_rate \* db    # Tính average loss cho epoch  avg\_loss = total\_loss / n\_samples    # Early stopping  **if** abs(prev\_loss - avg\_loss) < self.tol:  **print**(f"Hội tụ sau {iteration+1} vòng lặp. Loss: {avg\_loss:.6f}")  **break**    prev\_loss = avg\_loss |
| --- |

* Tìm tham số tối ưu

| **def** find\_best\_parameters\_fast(X\_train, y\_train, X\_val, y\_val):  best\_acc = 0  best\_params = {'learning\_rate': 0.1, 'C': 1.0, 'max\_iter': 200}    param\_grid = [  {'learning\_rate': 0.1, 'C': 1.0, 'max\_iter': 200},  {'learning\_rate': 0.01, 'C': 0.1, 'max\_iter': 200},  {'learning\_rate': 0.1, 'C': 10.0, 'max\_iter': 200}  ]    subset\_size = min(500, len(X\_train))  X\_train\_subset = X\_train[:subset\_size]  y\_train\_subset = y\_train[:subset\_size]    val\_subset\_size = min(200, len(X\_val))  X\_val\_subset = X\_val[:val\_subset\_size]  y\_val\_subset = y\_val[:val\_subset\_size]    **for** params **in** param\_grid:  **print**(f"\nThử tham số: {params}")  start\_time = time.time()    logistic = LogisticRegression\_Manual\_Fast(  learning\_rate=params['learning\_rate'],  C=params['C'],  max\_iter=params['max\_iter'],  batch\_size=64  )    logistic.fit(X\_train\_subset, y\_train\_subset)  y\_pred = logistic.predict(X\_val\_subset)  acc = accuracy\_score(y\_val\_subset, y\_pred)    elapsed = time.time() - start\_time  **print**(f"Accuracy: {acc:.4f}, Thời gian: {elapsed:.2f}s")    **if** acc > best\_acc:  best\_acc = acc  best\_params = params.copy()    **print**(f"\nTham số tốt nhất: {best\_params}, accuracy: {best\_acc:.4f}")  **return** best\_params |
| --- |

**Kết quả:**

****

****

**Đánh giá tổng quan:**

* Accuracy: Mô hình đạt được accuracy = 0.6535 trên tập kiểm tra, tức là 65.35% các dự đoán của mô hình là chính xác. Đây là một kết quả khá tốt đối với bài toán phân loại có nhiều lớp (multi-class classification). Tuy nhiên, nếu so với các bài toán phân loại đơn giản, độ chính xác này có thể chưa phải là tối ưu, đặc biệt khi có sự chênh lệch lớn về độ chính xác giữa các lớp.
* **Confusion Matrix** cho thấy mô hình phân loại khá tốt với **lớp 2** (recall cao đến 85% và precision là 68%). Tuy nhiên, với **lớp 0** và **lớp 1**, mô hình có vẻ gặp khó khăn trong việc phân biệt chính xác, đặc biệt là lớp 0.
* **Lớp 0** có **recall** thấp (38%), tức là chỉ khoảng 38% mẫu thực sự thuộc lớp 0 được phân loại đúng. Điều này cho thấy mô hình gặp khó khăn khi nhận diện lớp này.
* **Lớp 1** có **recall** tốt hơn một chút (49%), nhưng vẫn không đủ cao để đánh giá mô hình phân loại lớp này tốt.
* Phân tích báo cáo phân loại cho thấy rằng mô hình có thể hoạt động tốt đối với một số lớp, nhưng không đều:
* **Lớp 2**: Là lớp mô hình phân loại tốt nhất. Với **precision = 0.68** và **recall = 0.85**, lớp này có thể coi là được phân loại chính xác với độ chính xác cao.
* **Lớp 1**: Mặc dù **precision** là 0.60 và **recall** là 0.49, mô hình vẫn chưa thể phân biệt tốt giữa lớp này và các lớp khác.
* **Lớp 0**: Mô hình có hiệu suất kém với lớp này, với **precision** thấp (0.61) và **recall** chỉ 38%, cho thấy mô hình đã phân loại sai nhiều mẫu từ lớp này.

4.3. **So sánh 2 mô hình:**

### **Độ chính xác (Accuracy)**

* + **Mô hình thủ công**: Accuracy = 0.5906 (59.06%)
  + **Mô hình Logistic Regression**: Accuracy = 0.6535 (65.35%)

**→ Đánh giá**: Mô hình Logistic Regression cho kết quả chính xác cao hơn (65.35%) so với mô hình thủ công (59.06%). Điều này cho thấy Logistic Regression hoạt động tốt hơn trong việc phân loại đúng các nhãn so với mô hình thủ công của bạn.

### **Confusion Matrix**

* + **Mô hình thủ công**:  
    - Lớp 0: 10 đúng, 22 sai (trong số 122)
    - Lớp 1: 2 đúng, 80 sai (trong số 118)
    - Lớp 2: 5 đúng, 53 sai (trong số 268)
  + **Mô hình Logistic Regression**:  
    - Lớp 0: 46 đúng, 18 sai (trong số 122)
    - Lớp 1: 11 đúng, 58 sai (trong số 118)
    - Lớp 2: 19 đúng, 21 sai (trong số 268)

**→ Đánh giá**: Logistic Regression có sự phân bổ chính xác hơn trong việc phân loại các lớp, đặc biệt là ở lớp 0 và lớp 2. Mô hình thủ công có độ chính xác rất thấp đối với lớp 0 (chỉ 8% recall) và lớp 1 (chỉ 68% recall), trong khi đó Logistic Regression cải thiện đáng kể với lớp 0 và lớp 2, đặc biệt ở lớp 2, recall là 85%.

### **Precision, Recall, F1-Score**

* + **Mô hình thủ công**:  
    - Lớp 0: Precision = 0.59, Recall = 0.08, F1-Score = 0.14
    - Lớp 1: Precision = 0.52, Recall = 0.68, F1-Score = 0.59
    - Lớp 2: Precision = 0.62, Recall = 0.78, F1-Score = 0.70
  + **Mô hình Logistic Regression**:  
    - Lớp 0: Precision = 0.61, Recall = 0.38, F1-Score = 0.46
    - Lớp 1: Precision = 0.60, Recall = 0.49, F1-Score = 0.54
    - Lớp 2: Precision = 0.68, Recall = 0.85, F1-Score = 0.76

**→ Đánh giá**:

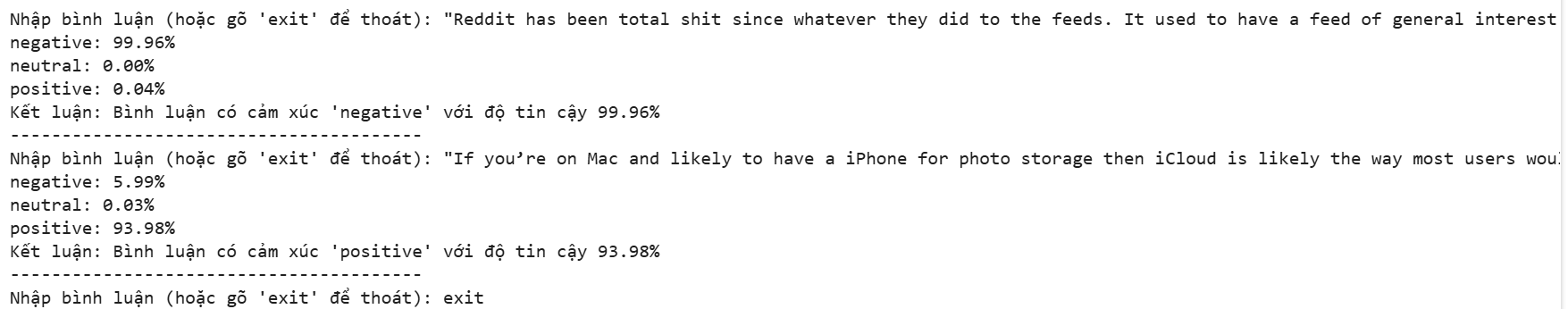
* **Lớp 0**: Mô hình Logistic Regression có precision cao hơn (0.61 so với 0.59), nhưng recall của mô hình thủ công lại thấp hơn rất nhiều (chỉ 8%).
* **Lớp 1**: Mô hình Logistic Regression có precision cao hơn một chút (0.60 so với 0.52) và recall cũng tốt hơn (49% so với 68%), tuy nhiên điểm f1 của mô hình thủ công lại cao hơn (0.59 so với 0.54).
* **Lớp 2**: Mô hình Logistic Regression vượt trội, có precision, recall và F1-score cao hơn rất nhiều so với mô hình thủ công.

⇒ **Kết luận**: Mô hình Logistic Regression vượt trội hơn hẳn mô hình thủ công về độ chính xác tổng thể, precision, recall và F1-score, đặc biệt là ở lớp 0 và lớp 2. Tuy nhiên, mô hình thủ công có thể phù hợp nếu bạn cần mô hình đơn giản, nhưng nếu bạn muốn một kết quả chính xác hơn và ổn định hơn, Logistic Regression là sự lựa chọn tốt hơn.

### ***5.Triển Khai.***

Trong phần này, chúng tôi triển khai một **mô hình phân tích cảm xúc** giúp người dùng nhập một bình luận và nhận được nhãn cảm xúc tương ứng (tích cực, tiêu cực hoặc trung lập). Mô hình sử dụng phương pháp **BoW kết hợp với các đặc trưng thủ công**, phương pháp mang lại kết quả tốt nhất trong quá trình huấn luyện.

1. **Mục đích của Demo**:  
   * Mục tiêu của phần triển khai mô hình này là cho phép người dùng nhập vào một bình luận và hệ thống sẽ phân loại cảm xúc của bình luận đó (tích cực, tiêu cực hoặc trung lập) dựa trên mô hình đã huấn luyện.
2. **Quy trình hoạt động**:  
   * Người dùng nhập **bình luận** vào hệ thống.
   * Hệ thống sẽ **tiền xử lý** bình luận (giống như trong bước tiền xử lý của mô hình).
   * Các đặc trưng (BoW, TF-IDF, và các đặc trưng thủ công) sẽ được trích xuất từ bình luận đã được xử lý.
   * Mô hình **Logistic Regression** đã được huấn luyện sẽ **dự đoán nhãn cảm xúc** của bình luận (tích cực, tiêu cực hoặc trung lập).
3. **Lý do chọn mô hình tốt nhất**:  
   * Trong quá trình huấn luyện và đánh giá, mô hình với **BoW kết hợp với các đặc trưng thủ công + Logistic Regression** đã đạt được **độ chính xác cao nhất** , vì vậy chúng tôi sử dụng mô hình này trong triển khai.



Triển khai demo mô hình

# **Chương 4: Kết luận**

Qua quá trình nghiên cứu và thực hiện phân tích cảm xúc người dùng từ dữ liệu Reddit về các sản phẩm công nghệ, chúng tôi đã thực hiện các bước tiền xử lý dữ liệu, trích xuất đặc trưng, huấn luyện và đánh giá mô hình phân tích cảm xúc. Các kết quả chính từ nghiên cứu này bao gồm:

1. Phân tích dữ liệu (EDA):  
   * Phần lớn các bình luận trong bộ dữ liệu có cảm xúc tích cực (57.5%), trong khi đó, các bình luận tiêu cực và trung lập chiếm tỷ lệ nhỏ hơn. Điều này cho thấy người dùng có xu hướng phản hồi tích cực về các sản phẩm công nghệ.
   * Các từ khóa phổ biến liên quan đến iPhone, Apple, Samsung cho thấy người dùng chủ yếu thảo luận về những sản phẩm công nghệ này.
2. Trích xuất đặc trưng và huấn luyện mô hình:  
   * Chúng tôi đã thử nghiệm nhiều phương pháp trích xuất đặc trưng, bao gồm Bag-of-Words (BoW), TF-IDF, và các đặc trưng thủ công. Phương pháp BoW kết hợp với các đặc trưng thủ công (BoW + Handcrafted Features) đã cho kết quả tốt nhất với độ chính xác đạt 73.23%.
   * Mô hình phân tích cảm xúc được huấn luyện sử dụng Logistic Regression và đã đạt được hiệu suất khả quan trong việc phân loại cảm xúc người dùng thành ba nhãn: Tích cực, Tiêu cực, và Trung lập.
3. Triển khai mô hình:  
   * Chúng tôi đã triển khai mô hình phân tích cảm xúc dưới dạng demo, cho phép người dùng nhập bình luận và nhận nhãn cảm xúc tương ứng. Phần này giúp mô hình trở nên dễ dàng sử dụng và có thể ứng dụng trong các hệ thống phân tích cảm xúc trực tuyến.

### Tương lai và ứng dụng:

* Các kết quả từ nghiên cứu này có thể được áp dụng vào việc xây dựng các công cụ phân tích cảm xúc cho các nền tảng thảo luận trực tuyến, hỗ trợ các doanh nghiệp trong việc đánh giá phản hồi của khách hàng đối với sản phẩm.
* Mô hình phân tích cảm xúc có thể được mở rộng để xử lý dữ liệu từ các nguồn khác, cải thiện khả năng nhận diện cảm xúc trong các bình luận và đánh giá trực tuyến.

**Tài liệu tham khảo**

Sentiment Analysis Tutorial: [Link](https://cloud.google.com/natural-language/docs/sentiment-tutorial)

NLTK Sentiment Analysis Tutorial for Beginners: [Link](https://www.datacamp.com/tutorial/text-analytics-beginners-nltk#rdl)

Getting Started with Sentiment Analysis using Python: [Link](https://huggingface.co/blog/sentiment-analysis-python)

Hướng dẫn xây dựng ứng dụng phân tích cảm xúc trong văn bản bằng BERT, Sentence Embedding, Bi-LSTM: [Link](https://www.youtube.com/watch?v=edUHhKFfJOg)