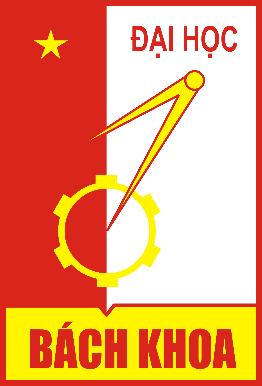
**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

----- 🕮 -----



**ĐỒ ÁN MÔN HỌC**

***Đề tài: Phân loại cảm xúc bình luận  
trên sàn thương mại điện tử***

**Giảng viên:** PGS.TS. Lê Thanh Hương

**Nhóm sinh viên thực hiện:**

Phùng Phú Cường – 20190084

Nguyễn Trần Hiếu Giang – 20191804

Nguyễn Đức Hiếu – 20192845

Vũ Văn Mạnh – 20191563

**Hà Nội, ngày 11 tháng 7 năm 2023**

**MỤC LỤC**

[Chương 1. Giới thiệu 4](#_Toc139921268)

[1.1. Tổng quan 4](#_Toc139921269)

[1.2. Phát biểu bài toán 4](#_Toc139921270)

[1.3. Khó khăn và thách thức 4](#_Toc139921271)

[1.4. Ứng dụng của bài toán 5](#_Toc139921272)

[Chương 2. Các nghiên cứu liên quan 6](#_Toc139921273)

[Chương 3. Các phương pháp phân loại văn bản 7](#_Toc139921274)

[3.1 Biểu diễn dữ liệu văn bản 7](#_Toc139921275)

[3.2 Một số mô hình Học Máy trong bài toán phân lớp 9](#_Toc139921276)

[3.2.1 Mô hình hồi quy Logistic 9](#_Toc139921277)

[3.2.2 Mô hình máy véc-tơ hỗ trợ 10](#_Toc139921278)

[3.2.3 Mô hình phân lớp Naive Bayes 13](#_Toc139921279)

[3.2.4 Mô hình rừng ngẫu nhiên 14](#_Toc139921280)

[3.2.5 Thuật toán tăng cường độ dốc 15](#_Toc139921281)

[3.3 BERT 16](#_Toc139921282)

[Chương 4. Mô hình phân loại cảm xúc văn bản 18](#_Toc139921283)

[4.1. Tiền xử lý dữ liệu 18](#_Toc139921284)

[4.2. Biểu diễn dữ liệu trong không gian vector 19](#_Toc139921285)

[4.2.1. Biểu diễn dữ liệu với TFIDFVectorizer 19](#_Toc139921286)

[4.2.2. Biểu diễn dữ liệu với mô hình fastText 20](#_Toc139921287)

[4.2.3. Biểu diễn với mô hình PhoBERT 20](#_Toc139921288)

[4.3. Mô hình phân lớp 20](#_Toc139921289)

[4.3.1. Các mô hình Học Máy phân lớp 20](#_Toc139921290)

[4.3.2. Mô hình phân loại học sâu PhoBERT 21](#_Toc139921291)

[Chương 5: Kịch bản thực nghiệm 23](#_Toc139921292)

[5.2. Tiêu chí đánh giá 24](#_Toc139921293)

[5.3. Môi trường thực nghiệm 25](#_Toc139921294)

[5.4. Kết quả thực nghiệm 25](#_Toc139921295)

[Chương 6: Kết luận 30](#_Toc139921296)

[Tài liệu tham khảo 31](#_Toc139921297)

# **Chương 1. Giới thiệu**

## **1.1. Tổng quan**

Trên các sàn thương mại điện tử hiện nay, phần bình luận đánh giá sản phẩm hiện đang là một phần không thể thiếu giúp cho các khách hàng có những đánh giá khách quan ban đầu về sản phẩm thông qua những trải nghiệm của người dùng trước, từ đó đưa đến quyết định có nên mua hay không. Chính vì vậy, các đánh giá này có ảnh hưởng khá lớn đến độ tin cậy của một nhãn hàng, từ đó đặt ra một nhu cầu cần phân loại được nhận xét tích cực, tiêu cực nhằm nhanh chóng cải thiện, nâng cao trải nghiệm người dùng và nâng cao doanh số.

Mỗi bình luận đều hướng đến bày tỏ ý kiến tiêu cực, tích cực hoặc trung tính đối với các sản phẩm. Những bình luận chủ quan thường bao gồm những từ ngữ trực tiếp thể hiện trạng thái tiêu cực hoặc tích cực, tuy nhiên do tính chất khách hàng, những từ viết tắt, từ không dấu, sai chính tả hoặc cả những từ mang lớp nghĩa mới theo trend hiện hành xuất hiện rất nhiều.

Vì vậy chúng em lựa chọn đề tài này để có thể nghiên cứu, xây dựng và so sánh các mô hình phân lớp nhằm tìm ra những mô hình hiệu quả trong việc dự đoán xu hướng tiêu cực, tích cực hay trung tính của một bình luận tiếng Việt.

## **1.2. Phát biểu bài toán**

Bài toán **Phân loại cảm xúc bình luận trên sàn thương mại điện tử** được mô tả như sau:

* **Đầu vào**: Một câu bình luận tiếng Việt.
* **Đầu ra**: Y ∈ {-1; 0; 1}, với Y là nhãn của bình luận, -1 là tiêu cực, 0 là trung tính, -1 là tích cực.

Trong project này, đề tài **Phân loại cảm xúc bình luận trên sàn thương mại điện tử** sẽ xây dựng tập dữ liệu huấn luyện trên miền dữ liệu tiếng Việt. Sau đó, sửu dụng các phương pháp biểu diễn văn bản trong không gian véc-tơ kết hợp các mô hình học máy thống kế và học sâu để phân loại bình luận.

## **1.3. Khó khăn và thách thức**

Để thực hiện giải quyết bài toán đã gặp không ít những khó khăn thách thức nhất định như sau:

Khó khăn lớn nhất chính là quá trình xây dựng được tập dữ liệu bình luận đáp ứng đủ yêu cầu của bài toán, không bị quá bias hoặc mất cân bằng, có thể khiến mô hình đưa ra những phân loại chính xác. Những bình luận có thể gán được các nhãn tiêu cực, tích cực, trung tính dựa trên số sao, tuy nhiên, nhiều khi số sao người dung đánh giá lại không giống với những từ ngữ nhận xét trong bình luận đó, do đó để có thể gán nhãn được chính xác phải mất nhiều thời gian ngồi đọc, phân tích bình luận, tìm những bình luận đáp ứng đủ tiêu chí nhằm hạn chế gán nhãn nhầm ảnh hưởng đến mô hình.

Thách thức thứ hai là việc biểu diễn các bình luận trên không gian véc- tơ. Do đặc điểm phong phú, đa dạng trong ngữ nghĩa của ngôn ngữ tiếng Việt, khiến cho việc biểu diễn văn bản tiếng Việt mà vẫn giữ được ý nghĩa của từ, ngữ cảnh của câu trong không gian vec-tơ gặp nhiều khó khăn. Cùng với đó là những từ ngữ viết tắt, không dấu sai chính tả, những icon, hay những từ vô nghĩa , không có tính phân loại gây nhiễu trong quá trình huấn luyện.

Một khó khăn nữa chính là việc lựa chọn mô hình cũng như tham số thích hợp để mô hình phân lớp đạt được kết quả cao nhất với đầu vào là tập dữ liệu bình luận tiếng Việt. Việc sử dụng những mô hình học sâu để giải quyết bài toán cũng có nhiều thách thức do tập dữ liệu lớn, thời gian huấn luyện lâu nên đòi hỏi phần cứng cũng như tốc độ xử lý mạnh mẽ.

## **1.4. Ứng dụng của bài toán**

Ứng dụng của bài toán là phân loại bình luận tiếng Việt, giúp cho các nhãn hàng hiểu hơn về khách hàng dựa trên những phản hồi, từ đó cũng giúp các doanh nghiệm nhanh chóng tìm ra những thiếu xót trong sản phẩm cũng như những trải nghiệm dịch vụ từ đối tác bên thứ ba để có thể cải thiện sớm nhất giảm thiểu các bình luận tiêu cực và nâng cao uy tín cho nhãn hàng.

Ngoài ra bài toán có thể được mở rộng hơn như phân loại số sao bình luận, làm tiền đề phát triển các mô hình phân loại khác trong tương lai.

# **Chương 2. Các nghiên cứu liên quan**

Nguồn gốc của phân tích ngữ nghĩa bắt nguồn từ những năm 1950, khi phân tích ngữ nghĩa chủ yếu được sử dụng trên các tài liệu giấy bằng văn bản. Tuy nhiên, ngày nay, phân tích tình cảm được sử dụng rộng rãi để khai thác thông tin từ nội dung trên In-tơ-nét, bao gồm văn bản, tweet, blog, phương tiện truyền thông xã hội, bài báo, đánh giá và nhận xét. Điều này được thực hiện bằng cách sử dụng một loạt các kỹ thuật khác nhau, bao gồm xử lý ngôn ngữ tự nhiên, thống kê, luật kết hợp và phương pháp học máy.

Những bài toàn phân tích ngữ nghĩa có thể được phân loại thành hai loại: tích cực, tiêu cực, trung tính hoặc vào sử dụng thang điểm n, ví dụ: rất tốt, tốt, tạm, xấu, rất tệ. Bài toán phân loại ngữ nghĩa là bài toán phân loại được thực hiện trên nhiều tập dữ liệu khác nhau, ngôn ngữ văn bản đa dạng và với mỗi bài toán sẽ có đặc trưng riêng, dưới đây là một số nghiên cứu liên quan đến bài toán phân loại ngữ nghĩa:

Đầu tiên là công trình nghiên cứu cho bài toán phân tích ngữ nghĩa cho tập dữ liệu trên mạng xã hội Twitter của nhóm tác giả Vishal A. Kharde và S.S. Sonawane [1]. Dữ liệu thô được lấy từ mạng xã hội Twitter được tiến hành tiền xử lý (loại bỏ từ dừng, ký tự lạ, biểu tượng cảm xúc, số cùng với đó sửa lỗi chính tả, trùng lặp câu, thay thế từ viết tắt thành từ hoàn chỉnh...), sau đó nhóm tiến hành trích xuất từ loại trong câu để tìm ra đặc trưng trong câu như từ phủ định, từ có tác dụng phân loại và biểu diễn câu, đặc trưng của câu trong không gian vec-tơ sử dụng phương pháp biểu diễn dựa trên tần suất từ. Đầu vào của mô hình là vec-tơ câu cùng với vec-tơ trích xuất đặc trưng của câu và mô hình phân loại là mô hình Naive Bayes (độ chính xác là 74.56%) và mô hình vec-tơ hỗ trợ (độ chính xác là 76.68%) để giải quyết bài toán.

Đối với văn bản tiếng Việt, một số công trình được nghiên cứu đã được vào ứng dụng, điển hình là phân tích quan điểm văn bản tiếng Việt của Underthesea (<http://undertheseanlp.com/#!/sentiment>) (một thư viện chuyên về xử lý ngôn ngữ tiếng Việt). Trong bài báo nghiên cứu [2], tác giả Quoc Thai Nguyen và ba cộng sự, đã tiến hành sử dụng nhiều cách tiếp cận khác nhau cho bài toán phân tích quan điểm đánh giá tiếng Việt. Nhóm sử dụng hai phương pháp Glove và fastText để biểu diễn văn bản tiếng Việt trong không gian vec-tơ. Sau đó, sử dụng các mô hình học máy phân loại máy vec-tơ hỗ trợ, xgboost, và các mô hình học sâu như mô hình bộ nhớ dài ngắn hạn (LSTM), mô hình BERT để giải quyết bài toán phân loại. Theo như thực nghiệm của tác giả, mô hình học sâu BERT là mô hình đạt kết quả tốt nhất so với các mô hình phân lớp còn lại.

Một cách tiếp cận khác đến từ tác giả Suong N. Hoang [3] và ba cộng sự cho bài toán phân tích quan điểm của các bài đánh giá sản phẩm điện tử tiếng Việt. Nhóm tác giả sử dụng fastText để biểu diễn văn bản tiếng Việt trong không gian vec-tơ làm đầu vào cho mô hình. Sau đó sử dụng các mô hình học sâu như mô hình bộ nhớ dài ngắn hạn (LSTM), mô hình đơn vị tái phát có cổng (GRUS) và mô hình Transformer để phân loại quan điểm. Theo như đánh giá của tác giả, mô hình Transformer đạt được điểm F1 cao nhất so với hai mô hình còn lại.

Ngoài những nghiên cứu tiếp cận sử dụng các mô hình học máy, học sâu cho văn bản tiếng Việt ở trên, nhóm tác giả Xuan-Son Vu, Seong-Bae Park [4] đã nghiên cứu và xây dựng SentiWordNet tiếng Việt bằng từ điển Tiếng Việt. Đây là bài toán nhằm xây dựng bộ từ điển để có thể sử dụng phương pháp tiếp cận Lexicon cho bài toán phân loại quan điểm tiếng Việt.

Qua đó, có thể thấy rằng việc nghiên cứu về phân tích ngữ nghĩa văn bản nhằm mục đích phân loại đã xuất hiện từ lâu. Với nhiều cách tiếp cận, hướng đi từ mọi thể loại văn bản cũng như ngôn ngữ trên thế giới bao gồm cả tiếng Việt. Đối với phân tích quan điểm văn bản tiếng Việt một số nghiên cứu đã đạt được kết quả rất tốt, với nhiều hướng đi, phương pháp cũng như cách tiếp cận khác nhau. Phân tích quan điểm văn bản là một bài toán mở rộng cho nhiều loại văn bản và do đó cần sự linh hoạt trong hướng đi, phương pháp để đáp ứng nhu cầu thể loại của văn bản cần phân tích quan điểm.

Trong project này, bài toán xây dựng mô hình phân tích quan điểm của bình luận tiếng Việt sử dụng hai phương pháp biểu diễn véc-tơ dữ liệu văn bản bình luận trong không gian là mô hình fastText và mô hình PhoBert kết hợp với mô hình học máy thống kê SVM ( Support Vector Machine ) .

# **Chương 3. Các phương pháp phân loại văn bản**

## **3.1 Biểu diễn dữ liệu văn bản**

Biểu diễn văn bản là một trong những vấn đề cơ bản trong khai thác văn bản và truy xuất thông tin. Nó nhằm mục đích đại diện bằng số cho các tài liệu văn bản phi cấu trúc để làm cho chúng có thể tính toán được về mặt toán học. Biểu diễn dữ liệu văn bản thường được chia làm hai phương pháp: phương pháp biểu diễn dựa trên tần suất từ và phương pháp dự đoán.

* **Phương pháp biểu diễn dựa trên tần suất từ**: xây dựng véc-tơ từ trong văn bản dựa vào tần suất xuất hiện trong văn bản.

***Mô hình túi từ***: là cách biểu diễn biến văn bản tùy ý thành các véc-tơ có độ dài cố định bằng cách đếm số lần mỗi từ xuất hiện.

Ví dụ: sử dụng mô hình túi từ để biểu diễn véc-tơ các câu trong văn bản đồ án; đồ án môn học.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | đồ | án | môn | học |
| đồ án | 1 | 1 | 0 | 0 |
| đồ án môn học | 1 | 1 | 1 | 1 |

Bảng 3.1: Mô hình túi từ trong văn bản.

Từ bảng 3.1 thu được véc-tơ biểu diễn từ của các câu:

* đồ án: [1, 1, 0, 0]
* đồ án môn học: [1, 1, 1, 1]

*Mô hình túi ký tự n-gram*: Mỗi từ w được thể hiện như là một túi các ký tự n-gram. Với mỗi từ w thêm ’<’ vào đầu và thêm ’>’ vào cuối từ, cùng với đó trong tập hợp ngram cũng bao gồm cả từ w. Ví dụ từ "where" và n-gram = 3: [<wh, whe, her, ere, re>, <where>]. Lưu ý thêm <her> trong từ her sẽ khác với từ her ở trong tri-gram của từ where.

Nhược điểm của mô hình túi từ là khi số lượng dữ liệu lớn, sẽ có nhiều từ có tần suất xuất hiện lớn nhưng lại không cần thiết.

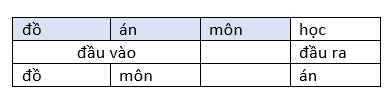
***TF-IDF***: trong đó TF là tần suất xuất hiện của một từ trong văn bản còn IDF là đánh giá mức độ quan trọng của từ trong tập văn bản. Khác với mô hình túi từ chỉ quan đếm đến tần suất xuất hiện của từ trong một văn bản, TF-IDF quan tâm đến tần suất xuất hiện của từ trong toàn bộ tập văn bản. Những từ xuất hiện càng nhiều ở nhiều văn bản (tính phân loại thấp) thì IDF càng nhỏ, dẫn đến kết quả cuối cùng sẽ nhỏ. Nhược điểm của cả hai phương pháp biểu diễn trên chính là chú trọng vào tần suất xuất hiện của từ, mà hầu như không quan tâm vào mặt ngữ cảnh.

* **Phương pháp dự đoán**: xây dựng véc-tơ từ trong văn bản dựa vào dự đoán theo ngữ cảnh.

***Mô hình túi từ liên tục của Mikolov***: học cách biểu diễn một từ bằng cách dự đoán từ theo ngữ cảnh. Ngữ cảnh được định nghĩa là một cửa sổ đối xứng chứa tất cả các từ xung quanh. Giả sử có chuỗi từ , mục tiêu của mô hình túi từ liên tục là tối đa hóa khả năng xảy ra xác suất của các từ được đưa ra xung quanh chuỗi T có thể biểu diễn theo công thức:

Trong đó: là ngữ cảnh của từ thứ . Các từ là cửa sổ ngữ cảnh có kích thước là . Ở trong hình 3.2, mô hình kiến trúc túi từ liên tục có kích thước của tập từ vựng là , kích thước của lớp ẩn là , các nút trên các lớp sử dụng liên kết đầy đủ. Đầu vào của mô hình là các véc-tơ theo dạng one-hot của các từ ngữ cảnh, tức là trong mỗi véc-tơ ngữ cảnh chỉ có một giá trị là 1 các giá khác đêu là 0. Trọng số giữa lớp đầu vào và lớp ẩn có thể được biểu diễn bởi một ma trận là có kích thước . Mỗi hàng của là một biểu diễn của véc-tơ có số chiều là đại diện cho từ . là ma trận kích thước là . Chúng ta có mối liên kết sau:

Trong đó, là véc-tơ đại diện cho từ ở đầu vào.



Hình 3.1: Ảnh minh họa về đầu vào - đầu ra của mô hình túi từ liên tục với cửa sổ ngữ cảnh có kích thước là 2 (c = 1).

Diagram

Description automatically generated

Hình 3.2: Mô hình túi từ liên tục của Mikolov [16].

Từ lớp ẩn đến lớp đầu ra sử dụng ma trận có kích thước là . Sử dụng các trong số trong ma trận có thể tính điểm cho mỗi từ trong từ vựng theo công thức:

Trong đó, là cột thứ trong ma trận . Sau đó, chúng ta có thể sử dụng softmax để có được phân phối sau của các từ, đó là phân phối đa thức.

Từ (3.5), (3.6), (3.7) ta thu được công thức:

Chú ý rằng và là hai véc-tơ đại diện cho từ . đến từ các hàng của ma trận (là ma trận trọng số từ đầu vào -> lớp ẩn), đến từ các cột của ma trận (là ma trận trọng số từ lớp ẩn -> đầu ra).

## **3.2 Một số mô hình Học Máy trong bài toán phân lớp**

***Suy giảm độ dốc***: là một thuật toán tối ưu hóa việc lặp đi lặp lại để tìm được giá trị nhỏ nhất của một hàm. Để tìm được điểm cực tiểu của một hàm, bắt đầu tại một số điểm ngẫu nhiễn và thực hiện các bước di chuyển ngược với dấu của đạo hàm hiện tại.

Thuật toán:

* Bắt đầu với việc khởi tạo ngẫu nhiên tham số:
* Cập nhật bằng cách lặp đi lặp lại công thức dưới đây cho đến khi nhận được kết quả chấp nhận được: . Trong đó, là tỉ lệ học tập có thể điều chỉnh được dùng để thay tối độ hội tụ của hàm mất mát tới điểm cực tiểu toàn cục. là hàm mất mát tại .

***Suy giảm độ dốc ngẫu nhiên***: là một phiên bản của thuật toán **Suy giảm độ dốc**, thuật toán này chọn ngẫu nhiên một điểm dữ liệu từ toàn bộ tập dữ liệu tại mỗi lần lặp lại để giảm các tính toán rất nhiều. Với thuật toán **Suy giảm độ dốc**, thường thì sẽ được cập nhật sau khi duyệt qua hết tập dữ liệu, còn với **Suy giảm độ dốc** **ngẫu nhiên**, sẽ được cập nhật sau khi duyệt qua một điểm (sau đó tập dữ liệu sẽ được xáo trộn).

### **3.2.1 Mô hình hồi quy Logistic**

**Hồi quy Logistic**: không phải là thuật toán hồi quy mà là một thuật toán phân loại được sử dụng để gán giá trị cho tập dữ liệu rời rạc. Ví dụ: việc dự đoán email nào là spam hay không spam hoặc có thể sử dụng để dự đoán khối u bình thường hay là ác tính, ...

Trong hồi quy Logistic, thực chất là mô hình vẫn là một hàm tuyến tính của , chẳng hạn nếu muốn xác định hai nhãn và , chỉ cần tìm một hàm liên tục đơn giản có miền là , giá trị mô hình trả về cho đầu vào gần với giá trị , thì gán nhãn cho là , nếu không gán nhãn cho là . Một hàm có thuộc tính như thế gọi là hàm Logistic tiêu chuẩn hay còn gọi là hàm “**sigmoid**”:

Với là số Euler có giá trị xấp xỉ . cũng có thể được viết là .

Từ (3.5), ta biểu diễn mô hình hồi quy Logistic như sau:

Trong đó, là trọng số; là “bias” giúp cho mô hình tránh việc luôn đi qua gốc tọa độ. Sử dụng một ngưỡng để có thể xác định xem đầu vào thuộc nhãn nào. Chẳng hạn: chọn ngưỡng , những giá trị nào nhỏ hơn thì đầu vào gán nhãn , ngược lại đầu vào được gán nhãn .

Thay cho việc sử dụng tính toán tổn thất bình phương như của hồi quy tuyến tính, thì Logistic tập trung vào việc tìm các bộ tham số sao cho tối đa hóa khả năng mà mô hình với bộ tham số đó sinh ra các điểm dữ liệu gần nhất với các điểm dữ liệu có sẵn.

Trong ví dụ dưới đây, sử dụng cho bài toán phân lớp nhị phân (hai lớp và ), tuy vậy trong thực tế hồi quy Logistic có thể được sử dụng để giải quyết các bài toán đa lớp.

Giả sử rằng, có bộ dữ liệu trong tập đào tạo, lựa chọn một số giá trị cụ thể của và cho tham số của mô hình. Sử dụng mô hình cho sử dụng công thức (3.7) sẽ cho ra giá trị trong đó . Nếu có nhãn là thì (kết quả của mô hình) cũng nhiều khả năng nhãn và theo mô hình nó có giá trị là p. Giống với đó nếu như có nhãn là 0 thì có giá trị là . Do đó, có thể biểu theo công thức xác suất như sau:

Từ (3.8) và (3.9) ta có thể gộp thành công thức:

Với công thức (3.10) nếu là nhãn thì lúc đó công thức (3.10) sẽ chính là công thức (3.8) và ngược lại sẽ chính là công thức (3.9) với là nhãn . Tổng quát hóa (3.10) cho tập dữ liệu rời rạc và , ta cần tìm sao cho biểu thức dưới đây đạt giá trị lớn nhất:

Khi lớn, việc tính tích của số nhỏ hơn sẽ rất đến kết quả bị sai số do tích quá nhỏ. Để khắc phục điều này, sử dụng cho hàm mất mát để chuyển từ tích sang tổng, sau đó lấy ngược dấu để được một hàm mất mát. Lúc này bài toán chuyển từ tìm w cho biểu thức (3.11) lớn nhất sang bài toán tìm w sao cho biểu thức hàm mất mát dưới đây là nhỏ nhất:

Sử dụng thuật toán **Suy giảm độ dốc ngẫu nhiên** cho điểm dữ liệu ta tìm được công thức cập nhật cho (với là tỉ lệ học tập).

### **3.2.2 Mô hình máy véc-tơ hỗ trợ**

**Máy véc-tơ hỗ trợ**: là một thuật toán giám sát dựa trên lý thuyết thống kê và có mục đích xác định vị trí ranh giới (có thể là một đường thẳng, một mặt phẳng hoặc một siêu phẳng) để tạo ra phân tách tối ưu giữa các lớp. Trong một số trường hợp, không thể tách biệt hai nhóm điểm một cách hoàn hảo vì dữ liệu bị nhiễu, lỗi ghi nhãn hoặc giá trị ngoại lệ.

Máy véc-tơ hỗ trợ cho dữ liêu tách biệt tuyến tính: Khi dữ liệu có thể phân tách tuyến tính, có vô số cách có thể để xây dựng siêu phẳng phân tách tuyến tính giữa các lớp.

Diagram

Description automatically generated

Hình 3.3: Dữ liệu tách biệt tuyến tính [1].

Giả sử có điểm dữ liệu trong tập dữ liệu trong đó là một vectơ hàng có chiều tương ứng với điểm dữ liệu thứ và thuộc là nhãn của điểm dữ liệu thứ .

Khi đó, phương trình của một siêu phẳng được biểu diễn như sau:

Trong đó, là véc-tơ pháp tuyến theo hàng và có chiều; là một đại lượng vô hướng hay còn gọi là “bias”.

Giả định rằng các lớp có thể phân tách tuyến tính, tất cả các điểm dữ liệu với sẽ nằm ở một bên của siêu phẳng thỏa mãn: và ngược lại sẽ nằm ở một bên của siêu phẳng thỏa mãn: .

Những rằng buộc ở (3.14) và (3.15) chưa kết hợp yêu cầu về lề (được tính là khoảng cách gần nhất từ một điểm của một lớp đến siêu phẳng phân tách, điểm còn được gọi là véc-tơ hỗ trợ). Giả định siêu phẳng phân tách nằm ở tâm của hai siêu phẳng chứa các véc-tơ hỗ trợ của hai lớp, sử dụng một tham số để có thể biểu diễn hai siêu phẳng trên như sau:

Có thể giả sử, không làm mất đi tính tổng quát, các biến và được chia tỷ lệ vừa đủ, để giá trị của có thể được đặt thành . Do đó, hai siêu phẳng chứa các véc-tơ hỗ trợ của hai lớp được biểu diễn dưới dạng sau:

Từ (3.18) và (3.19) ta sẽ có được điều kiện tổng quát cho các điểm dữ liệu trong :

Do luôn cùng dấu với phía của nên cùng dấu với :

Gọi là khoảng cách giữa hai siêu phẳng chứa các véc-tơ hỗ trợ. Với là L2-Norm

ta có thể tính được như sau:

Mục tiêu của bài toán cần tim giá trị của và để có thể tối đa hóa giá trị của :

Do việc tối đa hóa cũng tương tự với việc giảm thiểu . Việc tìm và để tối đa hóa trị của cũng tương tự với việc tìm và sao cho biểu thức sau là tối thiểu:

Đến đây có thể sử dụng các công cụ hỗ trợ để tìm nghiệm cho bài toán tối ưu bậc 2 để có thể tìm được , .

**Máy véc-tơ hỗ trợ với lề mềm cho dữ liệu không thể tách rời**: Ở phần bên trên, đã thảo luận về máy véc-tơ hỗ trợ mà các điểm dữ liệu của hai lớp có thể phân tách tuyến tính. Tuy nhiên, khả năng phân tách tuyến tính hoàn hảo thực tế khá phức tạp và các tập dữ liệu thực thường sẽ không đáp ứng thuộc tính này. Nhiều tập dữ liệu thực gần với việc có thể phân tách được, trong đó hầu hết các điểm dữ liệu nằm trên các phía chính xác của các siêu mặt phẳng phân tách được lựa chọn kỹ càng.

Chart

Description automatically generated

Hình 3.4: Dữ liệu gần với dữ liệu hoàn toàn tách biệt tuyến tính [1]. Tuy vậy. vẫn có thể chọn được lề bằng cách bỏ qua một vài điểm dữ liệu (những điểm này rơi vào vùng không an toàn).

Với máy véc-tơ hỗ trợ với lề mềm, có thêm biến đo sự hy sinh . Các điểm nằm trong vùng an toàn thì . Những điểm nằm trong vùng không an toàn nhưng vẫn nằm cùng phía với đường phân chia thì có giá trị· thuộc . Ngược lại những điểm nằm trong vùng không an toàn nhưng nằm khác phía với đường phân chia thì có giá trị .

Chart, diagram, scatter chart

Description automatically generated

Hình 3.5: Các giá trị của trong các vùng an toàn hoặc không an toàn.

Từ bài toán tối ưu của bài toán **Máy véc-tơ hỗ trợ cho dữ liệu tách biệt tuyến tính** (3.25) kết hợp với biến đo sự hy sinh , ta thu được bài toán tối ưu **Máy véc-tơ hỗ trợ với lề mềm**:

Với là một hằng số được sử dụng để điều chỉnh tầm quan trọng giữa lề và sự hy sinh. Đến đây có thể sử dụng các công cụ hỗ trợ để tìm nghiệm cho bài toán tối ưu bậc 2 để có thể tìm được .

**Máy véc-tơ hỗ trợ phi tuyến**: Trong nhiều bài toán, đường ranh giới quyết định không phải là tuyến tính (hình 3.6), khi đó cần sử dụng các phép biến đổi không gian dữ liệu ban đầu sang không gian khác sao cho dữ liệu giữa các lớp ở không gian mới phân biệt tuyến tính hoặc gần như phân biệt tuyến tính thông qua hàm biến đổi (hàm thêm một chiều dữ liệu mới). Tuy nhiên, hàm thường tạo ra chiều không gian mới có số chiều lớn hơn hoặc có thể là vô hạn, do vậy nếu tính toán hàm trực tiếp sẽ tốn rất nhiều bộ nhớ, tốc độ tính toán chậm. Vì vậy, thay vì tính toán trực tiếp hàm , sử dụng hàm "hạt nhân" để tính tích của hai điểm và trong không gian mới mà không cần phải tính chia tiết tọa độ của hai điểm trong không gian mới. Dưới đây là một số hàm hạt nhân phổ biến:  
 *Hạt nhân tuyến tính*: là hàm tính tích vô hướng của hai véc-tơ.

*Hạt nhân đa thức*: Với là một số dương thể hiện bậc của đa thức, là một hằng số tự do:

*Hạt nhân Sigmoid*: Với thường là tham số có giá trị là là một hằng số tự do:

*Hàm cơ sở xuyên tâm*: Với tham số có thể điều chỉnh đóng một vai trò quan trọng trong hiệu suất của hạt nhân.

### **3.2.3 Mô hình phân lớp Naive Bayes**

**Naive Bayes**: là mô hình phân loại dựa trên định lý Bayes cho các xác suất có điều kiện. Định lý này tính xác suất xảy ra của một biến ngẫu nhiên khi biết các sự kiện liên quan đã xảy ra. Giả sử là biến ngẫu nhiên đại diện cho biến lớp và giá trị đặc trưng -chiều . Mục tiêu là tính được xác suất để xác định xem rơi vào lớp nào. Đặt các biến ngẫu nhiên cho các kích thước riêng lẻ của được ký hiệu là . Sau đó, người ta muốn ước tính xác suất có điều kiện . Xác suất này thường khó được tính trực tiếp, sử dụng Bayes có thể suy ra tương đương sau:

Bởi vì mẫu số độc lập với lớp nên ta có công thức (3.32). là xác suất để một điểm rơi vào lớp . Trong trường hợp xấp xỉ của Naïve Bayes, giả định rằng các giá trị là độc lập với nhau có điều kiện dựa trên một giá trị cố định của biến lớp . Điều này được biểu diễn theo công thức như sau:

Kết hợp (3.32) với (3.33):

Việc tính toán phụ thuộc vào dữ liệu và thông thường có ba phân phối được dùng để tính: Gaussian Naive Bayes, Đa thức Naive Bayes, và Bernoulli Naive Bayes.

*Gaussian Naive Bayes*: Với giá trị ở chiều dữ liệu thứ và lớp , phân bố theo phân phối chuẩn có độ kì vọng là và phương sai :

*Đa thức Naive Bayes*: Với là số lần dữ liệu thứ xuất hiện trong lớp , tổng số dữ liệu trong lớp (kể cả dữ liệu trùng): .

Trong đó, là một hằng số dương được thêm vào để tránh việc tử số bằng . Mẫu số cộng với để đảm bảo .

*Bernoulli Naive Bayes*: Mô hình này chỉ áp dụng cho tập dữ liệu nhị phân ( hoặc ).

### **3.2.4 Mô hình rừng ngẫu nhiên**

**Mô hình cây quyết định**: là mô hình phân loại, trong đó quá trình phân loại được mô hình hóa với việc sử dụng một tập hợp các quyết định phân cấp trên các biến đặc trưng, được sắp xếp giống với cấu trúc của cây. Quyết định tại một nút cụ thể của cây, gọi là tiêu chí phân tách, thường là một điều kiện đối với một hoặc nhiều biến trong dữ liệu huấn luyện. Tiêu chí phân tách này chia dữ liệu thành hai hoặc nhiều phần. Trong cây quyết định có hai loại nút, nút lá và nút trong (nằm giữa gốc và nút lá), mỗi nút lá được gắn nhãn lớp trội tại nút đó, mỗi nút bên trong đặc biệt là nút gốc tương ứng với toàn bộ không gian đặc trưng. Thuật toán quy nạp cây quyết định bắt đầu với tập dữ liệu huấn luyện đầy đủ tại nút gốc và phân vùng đệ quy dữ liệu thành các nút cấp thấp hơn dựa trên tiêu chí phân tách.

*Tiêu chí phân tách cây quyết định*: Mục tiêu của tiêu chí phân tách là tối đa hóa sự tách biệt của các lớp khác nhau giữa các nút con. Thiết kế của tiêu chí phân tách phụ thuộc vào bản chất của thuộc tính cơ bản:

1. Thuộc tính nhị phân: Chỉ có thể thực hiện một kiểu phân tách, cây luôn là nhị phân. Mỗi nhánh tương ứng với một trong các giá trị nhị phân.
2. Thuộc tính phân loại: Nếu một thuộc tính phân có n giá trị khác nhau, thì có nhiều cách để phân tách nó nổi bật là sử dụng phép tách n-nhánh, trong đó mỗi nhánh tương ứng với phép tách tương ứng và một giá trị thuộc tính cụ thể.

Do cần phải chọn từ nhiều thuộc tính và từ các lựa chọn thay thế khác nhau có sẵn để phân tách từng thuộc tính. Do đó, cần phải có các định lượng về chất lượng phân chia:

1. Chỉ số Gini cho một tập các điểm dữ liệu phân phối xác suất theo (trong đó: là xác suất một phần tử dữ liệu trong tập thuộc lớp ) được tính theo công thức sau:

Khi phân chia tập theo -nhánh thành các tập con (trong đó là số lượng phần tử của trong tập ), ta có chỉ số Gini tổng thể theo công thức:

Thuộc tính nào có chỉ số Gini-Split nhỏ nhất sẽ được lựa chọn. Thuật toán CART sử dụng chỉ số Gini làm tiêu chí phân tách.

1. Phép đo Entropy được sử dụng trong một trong những thuật toán phân loại sớm nhất, được gọi là ID3. Entropy E(S) cho một tập các điểm dữ liệu S phân phối xác suất theo (trong đó: là xác suất một phần tử dữ liệu trong tập thuộc lớp ) được tính theo công thức sau:

Cũng tương tự với chỉ số Gini, chỉ số Entropy tổng thể được biểu diễn theo công thức:

Giá trị thấp hơn của entropy được mong muốn hơn. Phép đo entropy được sử dụng bởi các thuật toán ID3 và C4.5.

*Tiếu chí dừng cây quyết định:* khi cây quyết định được phát triển cho đến khi mọi nút lá chỉ chứa các trường hợp thuộc một lớp cụ thể, cây quyết định kết quả thể hiện độ chính xác 100% đối với các trường hợp thuộc dữ liệu huấn luyện. Tuy nhiên, nó thường gặp khó với những trường hợp thử nghiệm không có trong tập huấn luyện bởi vì cây quyết định hiện đã quá phù hợp ngay cả với các đặc tính ngẫu nhiên trong các trường hợp huấn luyện. Đây chính là nhược điểm của cây quyết định.

**Rừng ngẫu nhiên**: là thuật toán xây dựng ngẫu nhiên nhiều cây quyết định từ **mô hình cây quyết định**, kết quả dự đoán cuối cùng được tổng hợp từ kết quả của các cây quyết định. Mỗi cây quyết định đều có yếu tố ngẫu nhiên như lấy dữ liệu ngẫu nhiên hoặc lấy thuộc tính ngẫu nhiên, do đó khi kết hợp nhiều cây quyết định ngẫu nhiên sẽ khắc phục được nhược điểm quá khớp với dữ liệu huyến luyện dẫn đến gặp khó với dữ liệu không có có trong tập huấn luyện của thuật toán cây quyết định. Giả sử trong tập huấn luyện có dữ liệu và mỗi dữ liệu có thuộc tính. Lấy ngẫu nhiên dữ liệu từ tập dữ liệu ban đầu (dữ liệu được lấy ra vẫn được giữ lại ở tập dữ liệu ban đầu), lấy ngẫu nhiên thuộc tính từ thuộc tính (). Với dữ liệu ngẫu nhiên kết hợp thuộc tính, sử dụng mô hình cây ngẫu nhiên để xây dựng được một cây ngẫu nhiên. Kết quả cuối cùng sẽ là tổng hợp từ các cây ngẫu nhiên được dựng.

### **3.2.5 Thuật toán tăng cường độ dốc**

**Phương pháp kết hợp mô hình**: là một cách tiếp cận để tăng độ chính xác của dự đoán bằng cách kết hợp các kết quả từ nhiều mô hình lại với nhau (các mô hình không bắt buộc phải khác nhau) nhằm giảm việc mô hình quá khớp hoặc mô hình chưa khớp với tập dữ liệu huấn luyện.

**Phương pháp kết hợp mô hình tăng cường**: mục tiêu chính của của việc kết hợp tăng cường là khắc phục việc mô hình chưa khớp với tập dữ liệu huấn luyện. Sử dụng một chuỗi các mô hình yếu tuần tự (thường là các mô hình giống nhau), sao cho mô hình sau sẽ học cách sửa lỗi và cập nhật lại trọng số từ kết quả của mô hình trước. Điều này tạo thành một chuỗi các mô hình mà mô hình sau lại tốt hơn mô hình trước.

**Tăng cường độ dốc**: là thuật toán sử dụng phương pháp kết hợp tăng cường, mô hình yếu được sử dụng trong thuật toán thường là mô hình cây quyết định (các cây quyết định thường có từ 8 đến 32 lá). Thuật toán sử dụng thuật toán **Suy giảm độ dốc** để các mô hình học cách sửa lỗi và cập nhật lại trọng số từ mô hình trước đó.

## **3.3 BERT**

**Mã hóa Transformers**: có nhiệm vụ học các đặc điểm, thông tin từ câu, sau đó biểu diễn dưới dạng véc-tơ thể hiện được toàn bộ thông tin từ câu đầu vào. Dữ liệu trước khi đưa vào mô hình cần phải được biểu diễn theo các véc-tơ, sau đó được chuyển qua lớp mã hóa vị trí (Positional Encoding) để lưu lại thông tin về vị trí ban đầu của các từ trong câu. Kiến trúc của mã hóa Transformers bao gồm một chồng 6 lớp, mỗi lớp có 2 lớp con. Lớp con đầu tiên là cơ chế chú ý nhiều đầu bao gồm nhiều lớp chú ý chạy song song với nhau (sử dụng cơ chế tự chú ý nhằm tìm ra từ giống với từ đầu vào trong câu), lớp con thứ hai là một mạng chuyển tiếp đầy đủ, giữa hai lớp con là kết nối dư và tiếp theo là lớp chuẩn hóa. Có thể hiểu đầu ra của mỗi lớp con là LayerNorm ( + lớp con ) trong đó lớp con là hàm được thực hiện bởi chính lớp con.

Diagram

Description automatically generated

Hình 3.7: Mô hình mã hóa Transformers [20].

**BERT (Biểu diễn bộ mã hóa hai chiều từ Transformers)**: là một mô hình học sẵn, học ra các véc-tơ đại diện cho văn bản thông qua ngữ cảnh hai chiều (từ trái qua phải, từ phải qua trái) hoặc được dùng để tinh chỉnh cho một nhiệm vụ cụ thể. Mô hình sử dụng kiến trúc gồm nhiều lớp Transformers hai chiều (mã hóa Transformers) như trong hình 3.8. Gọi là số lớp Transformers, là số lớp ẩn và là số lượng đầu của tự chú ý (self-attention). được chia làm 2 kích thước (, Tổng tham số ) và (, Tổng tham số ). Trong khi đào tạo, sử dụng "Mặt nạ LM" và "Dự đoán câu tiếp theo", cả hai đều là dự đoán không giám sát.

Diagram

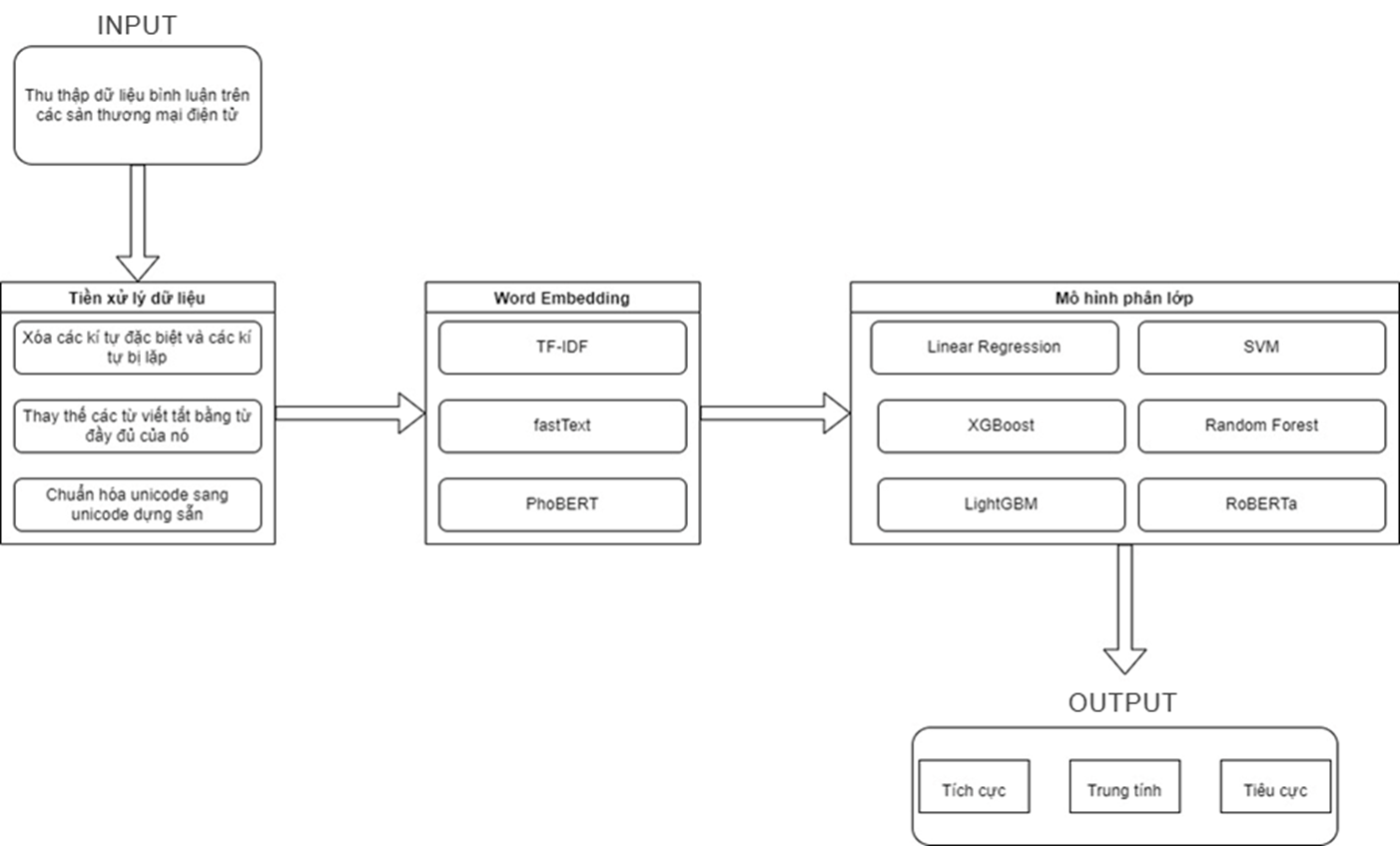
Description automatically generated

Hình 3.8: Mô hình BERT, trong hình "Trm" là mã hóa Transformers [5].

*Mặt nạ LM*: Một mô hình đào tạo thông qua ngữ cảnh hai chiều được đánh giá cao hơn so với việc đào ngữ cảnh từ trái qua phải (hoặc từ phải qua trái). Tuy nhiên, các mô hình đào tạo ngôn ngữ trước đây chỉ có thể đào tạo thông qua ngữ cảnh một chiều bởi vì nếu như đào tạo hai chiều sẽ làm cho một từ có thể gián tiếp nhìn thấy chính nó thông qua ngữ cảnh nhiều lớp. Do đó, để đào tạo hai chiều cần che giấu ngẫu nhiên một số từ, cụm từ được tách từ một từ, một câu hay là một văn bản, sau đó dự đoán các từ, cụm từ bị che giấu đó. Quá trình che giấu này được gọi là "Mặt nạ LM". Không phải lúc nào cũng thay thế các từ "bị che" bằng [Mặt nạ], chọn ngẫu nhiên từ hoặc cụm từ tại vị trí thứ i để dự đoán trong đó: từ hoặc cụm từ được chọn từ vị trí i được che, được thay thế ngẫu nhiên bằng các từ hoặc cụm từ khác và được giữ lại.

*Dự đoán câu tiếp theo*: Để cho mô hình đào tạo có thể biểu diễn được mối quan hệ giữa các câu nên cần xây dựng mô hình có thể dự đoán được câu tiếp theo.

# **Chương 4. Mô hình phân loại cảm xúc văn bản**



Hình 4.1. Luồng xử lý của bài toán xây dựng mô hình phân loại cảm xúc câu bình luận.

Trong chương 4, đồ án tập trung vào việc giải thích, làm rõ những phương pháp dùng cho các khối tiền xử lý dữ liệu, khối biểu diễn dữ liệu văn bản trong không gian véc-tơ (Word Embedding) và khối các mô hình phân lớp dùng cho bài toán (hình trên). Khối tiền xử lý dữ liệu đóng vai trò làm sạch dữ liệu các câu bình luận đã được thu thập, loại bỏ những kí tự lạ, thay thế các từ/cụm từ viết tắt và sai chính tả gây nhiễu giúp việc biểu diễn dữ liệu văn bản trong không gian véc-tơ tốt hơn. Khối biểu diễn dữ liệu thông qua các mô hình TFIDF, fastText và PhoBert có nhiệm vụ chuyển dữ liệu các câu bình luận sau khi đã tiền xử lý sang không gian véc-tơ để đưa vào các mô hình học máy thống kê. Khối mô hình phân lớp sử dụng đầu ra của khối biểu diễn dữ liệu là các véc-tơ đại diện cho các câu bình luận để làm dữ liệu huấn luyện, mục tiêu của khối mô hình là dự đoán một trong ba nhãn (tích cực, tiêu cực, trung tính) cho câu bình luận.

## **4.1. Tiền xử lý dữ liệu**

Dữ liệu dùng cho bài toán xây dựng mô hình phân loại cảm xúc của các câu bình luận trên sàn thương mại điện tử được thu thập từ các sàn thương mại điện tử Tiki và Lazada, bao gồm câu bình luận của khách hàng ở nhiều danh mục sản phẩm như sách báo, đồ điện tử, đồ gia dụng, sản phẩm chăm sóc sức khỏe, … Dữ liệu thu thập được là dữ liệu thô được lưu ở dạng tập tin json và csv với thông tin đi kèm gồm: nội dung câu bình luận và nhãn được gán cho câu bình luận đó.

Bài toán xây dựng mô hình phân tích cảm xúc các câu bình luận Tiếng Việt nhận đầu vào là câu bình luận, đầu ra là một trong các nhãn tích cực/tiêu cực/trung tính. Bước đầu tiền của tiền xử lý sẽ tiến hành loại bỏ những ký tự đặc biệt không phải là chữ (?@#$%!\*-"’), khoảng trống trắng bị thừa (giữa các từ có nhiều hơn 1 khoảng trống " "), các đường link, mã HTML xuất hiện trong câu bình luận và các chữ số bằng cách sử dụng thư viện biểu thức chính quy (regex). Ví dụ như ta có 1 câu “<p class=\"par\">This is an example</p>”, sau khi loại bỏ đi các kí tự đặc biệt và mã HTML xuất hiện trong câu thì ta sẽ được đầu ra là “This is an example”.

Hiện tại, có 2 loại mã Unicode phổ biến đó là Unicode tổ hợp và Unicode dựng sẵn, do đó, có thể sẽ dẫn đến tình trạng sau: ví dụ,

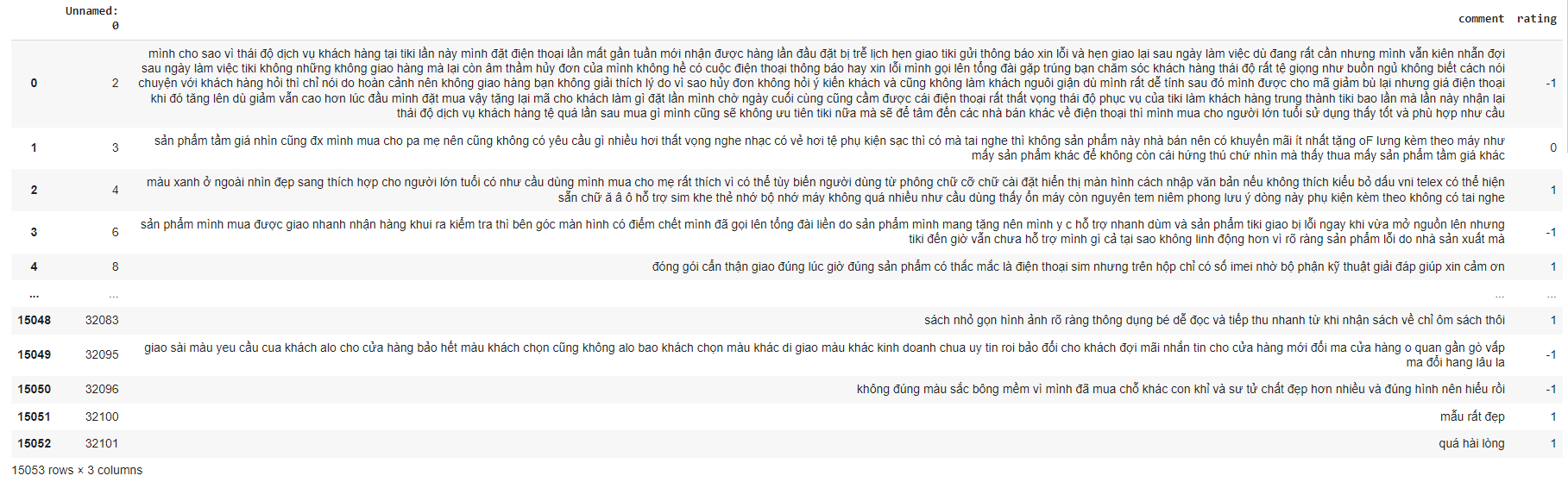
>>> 'hiếu' == 'hiếu'

False

>>> 'hiếu' == 'hiếu'

True

Điều này có thể dẫn đến việc khi xây dựng từ điển thì sẽ có nhiều từ bị lặp lại do dữ liệu được lấy từ nhiều trang thương mại điện tử khác nhau, dẫn đến việc các kí tự có thể được mã hóa Unicode khác nhau. Vì vậy việc chuẩn hóa tất cả các câu bình luận về một kiểu mã hóa là Unicode dựng sẵn là cần thiết để nâng cao chất lượng bộ từ điển. Cuối cùng chuyển tất cả các kí tự chữ cái in hoa về chữ cái thường và lưu lại bộ dữ liệu đã qua các bước tiền xử lý như trên vào tập tin csv, bao gồm 2 thuộc tính là câu bình luận và nhãn được gán cho câu bình luận đó.



Hình 4.2. Dữ liệu sau khi đã được tiền xử lý

## **4.2. Biểu diễn dữ liệu trong không gian vector**

Với các mô hình học máy, sử dụng hai mô hình học sẵn được đào tạo trên tập dữ liệu câu bình luận tiếng Việt là TFIDF, fastText và PhoBert để biểu diễn dữ liệu các câu bình luận tiếng Việt trong không gian véc-tơ. Cột nhãn của bình luận được chuyển thành véc-tơ gồm các giá trị (-1, 0, 1) trong đó: -1 là nhãn tiêu cực, 0 là nhãn trung tính và 1 là nhãn tích cực.

### **4.2.1. Biểu diễn dữ liệu với TFIDFVectorizer**

Trong khai phá dữ liệu văn bản (text mining), thuật ngữ TF-IDF (term frequency – inverse document frequency) là một phương thức thống kê được biết đến rộng rãi nhất để xác định độ quan trọng của một từ trong đoạn văn bản trong một tập nhiều đoạn văn bản khác nhau. Nó thường được sử dụng như một trọng số trong việc khai phá dữ liệu văn bản. TF-IDF chuyển đổi dạng biểu diễn văn bản thành dạng không gian vector (VSM), hoặc thành những vector thưa.

Trong bài tập lớn này, chúng em sẽ thực hiện tính toán TF-IDF dựa vào thư viện scikit-learn trên Python. Áp dụng TfidfVectorizer với việc đặt giới hạn kích thước của từ điển là 100,000 từ, đồng thời sử dụng mô hình ngôn ngữ n-gram để làm giàu thêm kích thước của từ điển (trong bài áp dụng unigram, bigram, trigram và 4-gram). Sự đóng góp về mức độ quan trọng của từ và tần suất xuất hiện của từ trong câu bình luận có mối quan hệ như một hàm phụ tuyến tính, do đó ta sẽ áp dụng sublinear\_tf để giảm giá trị TF của những từ có tần suất lớn. Vector đại diện cho câu sẽ là vector có kích thước bằng kích thước của từ điển và có giá trị là các giá trị TF-IDF của mỗi từ trong câu.

### **4.2.2. Biểu diễn dữ liệu với mô hình fastText**

fastText là một thư viện được dùng để biển diễn từ theo không gian véc-tơ được xây dựng bằng cách kết hợp: đại diện câu với mô hình túi từ, mô hình túi kí tự n-gram và kiến trúc mô hình túi từ liên tục của Mikolov.

Mô hình học sẵn fastText biểu diễn các véc-tơ (kích thước là 300) cho các từ, cụm từ trong câu bình luận tiếng Việt. Để có thể thực hiện việc véc-tơ nhúng, cần phải tách các câu bình luận thành các từ hoặc cụm từ (tokens). Sử dụng hàm tách từ tiếng việt của thư viện undertheseanlp, ví dụ văn bản: "chất lượng sản phẩm tuyệt vời giao hàng nhanh" sau khi tách từ, cụm từ sẽ được một danh sách các từ, cụm từ khác nhau như sau: ['chất lượng', 'sản phẩm', 'tuyệt vời', 'giao', 'hàng', 'nhanh']. Kết hợp các từ, cụm từ sau khi được tách từ dữ liệu các câu bình luận thu được một danh sách tổng hợp các từ, cụm từ của một câu bình luận. Sau đó, sử dụng mô hình fastText để ánh xạ các từ, cụm từ trên thành các véc-tơ có chiều là 300, nếu như từ, cụm từ nào chưa có trong mô hình fastText sẽ được ánh xạ thành một véc-tơ có chiều là 300. Véc-tơ đại diện cho câu bình luận sẽ là một véc-tơ trung bình của tổng các véc-tơ từ, cụm từ.

### **4.2.3. Biểu diễn với mô hình PhoBERT**

Mô hình PhoBERT được đào tạo sẵn là mô hình ngôn ngữ hiện đại cho tiếng Việt, phương pháp đào tạo trước của PhoBERT dựa trên RoBERTa (loại bỏ mục tiêu dự đoán câu tiếp theo, tự động thay đổi mô hình mặt nạ) giúp tối ưu hóa quy trình trước đào tạo của BERT để có hiệu suất mạnh mẽ hơn.

Trong bài toán xây dựng mô hình phân tích quan điểm văn bản báo chí Tiếng Việt sử dụng PhoBERT\_base\_fairse (bao gồm 4 tập tin nhỏ: config.json chứa cấu hình cài đặt của mô hình, model.bin lưu trữ tham số, trọng số được đào tạo trước của mô hình, bpe.codes và dict.txt chứ tập từ vựng sẵn có của PhoBERT) để biểu diễn câu bình luận tiếng Việt trong chiều không gian véc-tơ có kích thước véc-tơ biểu diễn văn bản là 768. Vẫn sử dụng hàm tách từ tiếng Việt của thư viện undertheseanlp, tuy nhiên khác với fastText tách thành các danh sách chứa từ, cụm từ thì ở đây sử dụng một cách biểu diễn từ, cụm từ như sau, chẳng hạn câu: "chất lượng sản phẩm tuyệt vời giao hàng nhanh" sau khi tách từ, cụm từ sẽ thành câu mới như sau: "chất\_lượng sản\_phẩm tuyệt\_vời giao hàng nhanh". Sử dụng thuật toán mã hóa cặp byte từ tập tin bpe.code trong PhoBert để mã hóa các câu sau khi được phân tách ở trên thành danh sách các từ phụ (tách từ theo mức nhỏ hơn từ nhưng lớn hơn ký tự). Sau đó, sử dụng mô hình học sẵn RoBERTa để biểu diễn véc-tơ cho câu. Tuy nhiên do giới hạn độ dài tối đa của một chuỗi đầu vào là 258, do đó văn bản sẽ được chia thành các đoạn nhỏ (có trung bình khoảng 200 từ, cụm từ mỗi đoạn), những đoạn văn bản này được đưa vào mô hình để tìm ra véc-tơ biểu diễn. Véc-tơ đại diện cho văn bản sẽ là một véc-tơ trung bình của tổng các véc-tơ của các đoạn văn bản được chia nhỏ trong văn bản.

## **4.3. Mô hình phân lớp**

### **4.3.1. Các mô hình Học Máy phân lớp**

**Mô hình Học Máy**: Các mô hình học máy đều sử dụng thư viện scikit-learn (là một thư viện học máy bao gồm nhiều thuật toán hồi quy, phân loại, phân cụm dành cho ngôn ngữ Python) với đầu vào là tập dữ liệu các danh sách véc-tơ của các câu bình luận trên sàn thương mại điện tử. Đầu ra là một nhãn được mô hình dự đoán của bài viết được đánh theo số: -1 là tiêu cực, 0 là trung tính, 1 là tích cực.

**Hồi quy Logistic**: là mô hình thường được thiết kế cho các vấn đề hai lớp, mô hình hóa mục tiêu bằng cách sử dụng phân phối xác suất nhị thức. Đầu ra được thể hiện dưới dạng xác suất và được tính theo công thức "sigmoid". Tuy nhiên, bài toán xây dựng mô hình phân tích cảm xúc các câu bình luận Tiếng Việt có ba nhãn: tiêu cực, tích cực và trung tính. Do đó, mô hình Hồi quy Logistic sẽ sử dụng phân bố xác suất đa thức để phân loại nhiều hơn hai nhãn (trong đó: phân phối xác suất nhị phân là trường hợp đặc biệt của phân bố xác suất đa thức). Trong mô hình này, tham số được sử dụng trong mô hình bao gồm: phân loại đa nhãn theo phân bố xác suất đa thức (multi\_class = "multinomial"), thuật toán tối ưu là "newton-cg" cho dữ liệu đa nhãn.

**Mô hình máy Véc-tơ hỗ trợ (SVM)**: có đầu ra là một trong những siêu mặt phẳng tốt nhất (trong một không gian đa chiều) để phân loại hai lớp (trong trường hợp phân loại nhị phân) của một bộ dữ liệu. Để sử dụng mô hình Véc-tơ hỗ trợ cho bài toàn phân loại có nhiều hơn hai nhãn sử dụng phương pháp Một-Một (One-Vs-One). Phương pháp này chia một bộ dữ liệu phân loại nhiều lớp thành các vấn đề phân loại nhị phân. Đối với bài toàn phân loại các câu bình luận có ba nhãn: tiêu cực, tích cực và trung tính sẽ được chia làm 3 bộ dữ liệu phân loại nhị phân như sau: (tiêu cực, tích cực), (tiêu cực, trung tính) và (tích cực, trung tính) sau đó sử dụng mô hình máy véc-tơ hỗ trợ cho bài toán nhị phân. Trong mô hình này, tham số được sử dụng trong mô hình bao gồm: phương pháp Một-Một cho phân loại đa lớp (decision\_function\_shape = "ovo"), hàm hạt nhân (kernel) là "rbf" (hàm cơ bản xuyên tâm).

**Mô hình phân lớp Naive Bayes**: là mô hình phân loại dựa trên định lý Bayes, đầu ra là xác suất điểm dữ liệu X rơi vào lớp C. Trong mô hình này, sử dụng Gaussian Naive Bayes để tính giá trị xác suất.

**Mô hình rừng ngẫu nhiên (Random Forest)**: có đầu ra được tổng hợp từ kết quả của các cây quyết định được xây dựng trong quá trình huấn luyện. Trong mô hình này, tham số được sử dụng trong mô hình bao gồm: tiêu chuẩn phân tách của cây quyết định (criterion) là chỉ số "Entropy", số lượng mẫu tối thiểu để có thể phân tách tại nút trong (min\_samples\_split) là 2, độ sâu của cây (max\_depth) không giới hạn (các nút được mở rộng cho đến khi tất cả các lá thuần túy hoặc lá chứa ít hơn số mẫu tối thiểu để phân tách), số mẫu tối thiểu trên các lá là 1.

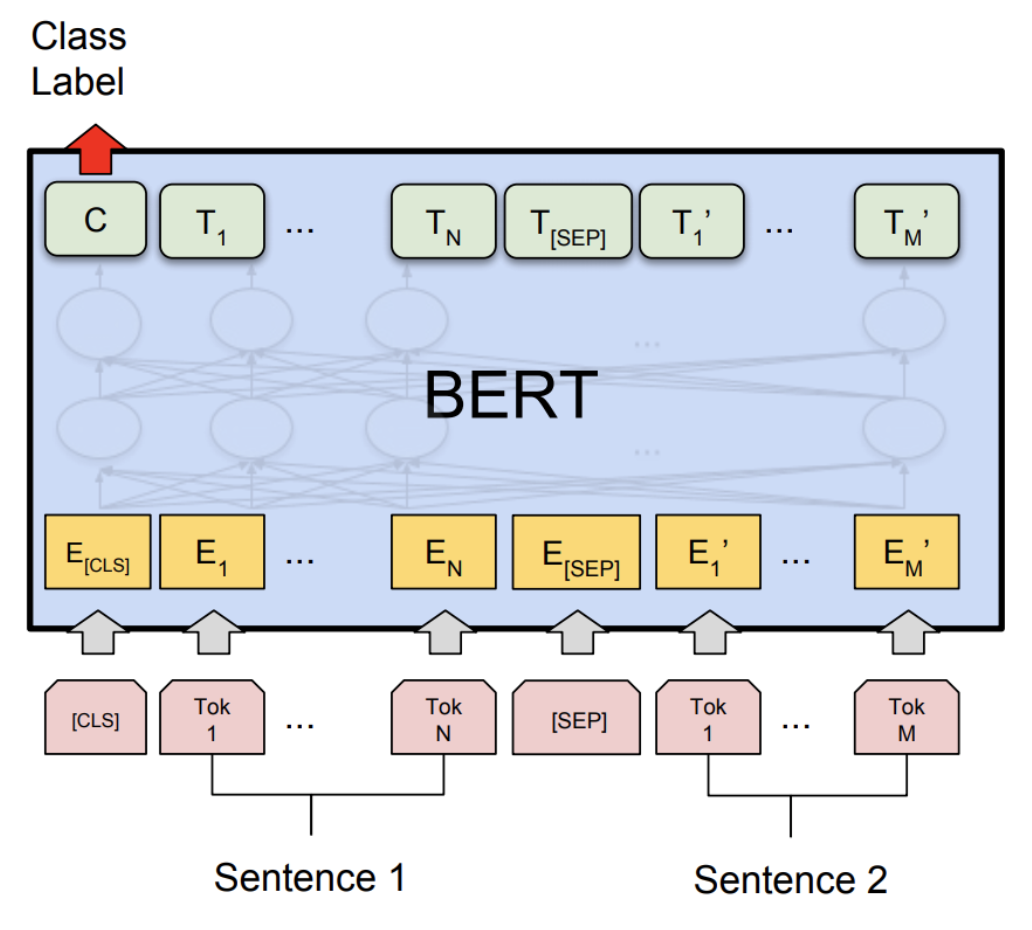
**Thuật toán tăng cường độ dốc cực cao (XGBoost)**: là thuật toán được xây dựng dựa trên thuật toán Tăng cường độ dốc. Đầu ra là một cây quyết định được sửa lỗi và cập nhật lại trọng số từ các mô hình cây quyết định trước đó. Trong mô hình này, tham số được sử dụng trong mô hình bao gồm: độ sâu của cây (max\_depth) là 10 (trong khi huấn luyện đã thử tinh chỉ độ sâu cao hơn nhưng kết quả không cải thiện), tỉ số học tập (learning\_rate) là 0.1, số lượng cây ngẫu nhiên được tạo thành là 100.

**Điều chỉnh tham số tự động sử dụng tìm kiếm lưới (GridSearchCV)**: thực hiện điều chỉnh siêu tham số, tham số để xác định các giá trị tối ưu cho một mô hình nhất định. Tìm kiếm lưới chạy qua tất cả các tham số khác nhau được đưa vào lưới tham số và tạo ra sự kết hợp tốt nhất của các tham số, dựa trên số liệu tính điểm của mô hình. Trong các mô hình trên, cũng được áp dụng tìm kiếm lưới để tìm được tham số tốt nhất cho mô hình. Chẳng hạn sử dụng tìm kiếm lưới với thuật toán tăng cường độ dốc cực cao XGBoost, lúc này tham số truyền vào sẽ như sau: độ sâu của cây (max\_depth) là một danh sách [3, 4, 5, 7], tỉ số học tập (learning\_rate) là danh sách [0.1, 0.01, 0.05], số lượng cây ngẫu nhiên được tạo thành là danh sách [100, 200, 300]. Khi đó tìm kiếm lưới sẽ kết hợp các danh sách lại với nhau và chạy qua các vòng lặp để tìm kiếm được tham số tối ưu nhất cho mô hình.

### **4.3.2. Mô hình phân loại học sâu PhoBERT**

Khác với mô hình PhoBert để biểu diễn câu bình luận trong không gian Véc-tơ, với mô hình RoBERTa phân loại sử dụng PhoBERT\_base\_transformers (bao gồm 4 tập tin nhỏ: config.json chứa cấu hình cài đặt của mô hình, model.bin lưu trữ tham số, trọng số được đào tạo trước của mô hình, bpe.codes và dict.txt chứ tập từ vựng sẵn có của PhoBERT).

Biểu diễn dữ liệu đầu vào mô hình: Bước đầu tiên cũng tương tự với việc biểu diễn dữ liệu bằng PhoBert, cần phải tách các nội dung các câu bình luận thành các từ hoặc cụm từ (tokens) giống như biểu diễn dữ liệu với PhoBert. Bước tiếp theo sử dụng hàm bpe (mã hóa cặp Byte) và hàm vocab (ánh xạ các từ, cụm từ về id của nó trong bộ từ vựng được cung cấp sẵn) của PhoBert để ánh xạ các từ, cụm từ trong một văn bản sang dạng véc-tơ có kích thước là 258 (có thể tùy chỉnh kích thước). Trong đó, những câu bình luận có độ dài nhỏ hơn 258 sẽ được thêm giá trị 0 vào phía cuối của véc-tơ, văn bản có độ dài lớn hơn 258 sẽ được cắt đi cho đủ 258. Tập dữ liệu lúc này thành một danh sách chứa các phần tử là một véc-tơ chứa id của từ, cụm từ của một văn bản trong bộ từ vựng có sẵn. Tạo một danh sách mặt nạ chứa các véc-tơ mặt nạ gồm có giá trị 0 và 1 tương ứng với từng văn bản để xác định phần từ 0 được thêm vào danh sách id từ, cụm từ của văn bản. Cột nhãn của câu bình luận được chuyển thành véc-tơ gồm các giá trị (-1, 0, 1) trong đó: -1 là nhãn tiêu cực, 0 là nhãn trung tính và 1 là nhãn tích cực. Cuối cùng chuyển ba tập dữ liệu (dữ liệu chứa thông tin ánh xạ id của các câu bình luận, dữ liệu mặt nạ, dữ liệu nhãn) sang dạng tensor và sử dụng Dataloader của torch để tạo dataloader cho mô hình.



Hình 4.3. Kiến trúc của mô hình BERT dùng để phân loại

**Xây dựng mô hình**: Tải mô hình RoBERTa phân loại từ tập tin model.bin của PhoBert. Kiến trúc của mô hình RoBERTa dùng để phân loại được chia làm hai nhiệm vụ: tìm véc-tơ thể hiện được toàn bộ thông tin từ câu đầu vào (thông qua mô hình RoBERTa) và phân loại nhãn. Dữ liệu chứa thông tin ánh xạ id của các câu bình luận và dữ liệu mặt nạ được đưa vào mô hình RoBERTa (gồm 12 lớp mã hóa Transformers) để tìm được véc-tơ tốt nhất cho dữ liệu các câu bình luận tiếng Việt. Các véc-tơ vừa tìm được kết hợp với dữ liệu nhãn để huấn luyện mô hình phân loại. Đầu ra của mô hình RoBERTa phân loại là phân phối xác suất do mô hình dự đoán. Trong mô hình này, lựa chọn các tham số cho mô hình như sau: Kích thước của véc-tơ mã hóa từ là 258, số lượng lớp (num\_labels) là 3, số lượng các đầu kết nối trong một lớp tự chú ý là 12, kích thước của lớp mã hóa là 768. Thuật toán Adam Weight Decay được sử dụng làm thuật toán tối ưu cho mô hình huấn luyện dựa trên những ưu điểm của nó so với những thuật toán tối ưu khác.

# **Chương 5: Kịch bản thực nghiệm**

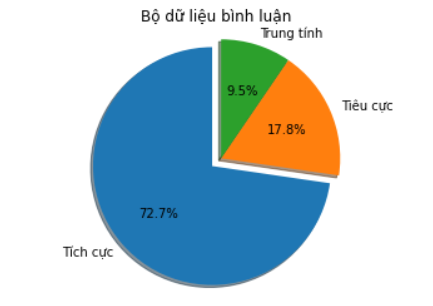
Trong chương này, chúng em tiến hành áp dụng các mô hình véc-tơ hóa dữ liệu, sử dụng các thuật toán học máy khác nhau để phân tích sự quan trọng trong việc biểu diễn dữ liệu và sự cần thiết của việc dán nhãn dữ liệu chính xác.

**5.1. Bộ dữ liệu**

Bộ dữ liệu bao gồm tổng cổng 15053 câu bình luận tiếng Việt được thu thập từ đánh giá trên các sàn thương mại điện tử Lazada, Tiki. Tổng cộng có tất cả 266044 từ, mỗi bình luận có trung bình 12 từ. Sắc thái các câu bình luận được chia làm 3 lớp: Tiêu cực, Trung tính và Tích cực. Phân phối số lượng nhãn cho mỗi lớp trong tập dữ liệu được thống kê trong bảng 5.1

|  |  |  |  |
| --- | --- | --- | --- |
| Nhãn | Tiêu cực | Trung tính | Tích cực |
| Số lượng | 2676 | 1435 | 10942 |

Bảng 5.1. Thống kê số lượng nhãn



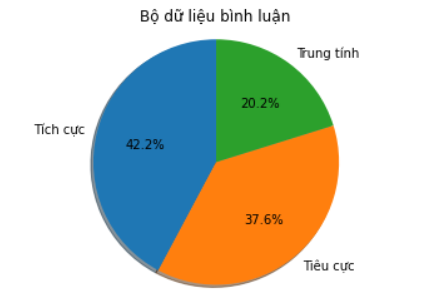
Hình 5.1. [Bộ dữ liệu gốc](https://drive.google.com/file/d/1uy3F6vg-MpdHPbFHJjvrgOkcueQa53SG/view?usp=sharing)

Trong đó, nhãn Tích cực là tổng hợp của các bình luận đánh giá 4 sao và 5 sao; Tiêu cực là tổng hợp của 1 sao và 2 sao; Trung tính là bình luận đánh giá 3 sao.

Theo hình 5.1 có thể thấy bộ dữ liệu có dấu hiệu lệch về nhãn Tích cực, chúng em đã áp dụng phương pháp Undersampling, giảm ngẫu nhiên số mẫu có nhãn Tích cực để tránh sự mất cân bằng dữ liệu (Data Imbalance). Kết quả sau khi thực hiện được cho bởi bảng bên dưới:

|  |  |  |  |
| --- | --- | --- | --- |
| Nhãn | Tiêu cực | Trung tính | Tích cực |
| Số lượng | 2676 | 1435 | 3002 |

Bảng 5.2. Thống kê số lượng nhãn sau khi Undersampling



Hình 5.2. Bộ dữ liệu sau khi áp dụng Undersampling

Như vậy, sau khi áp dụng Undersampling, phân phối bộ dữ liệu đã cân bằng hơn. Điều này là rất quan trọng vì trong bài toán phân loại, chúng em sẽ phân tích lí do ngay sau đây.

## **5.2. Tiêu chí đánh giá**

Độ đo được sử dụng trong bài tập lớn này để đánh giá là: Độ chính xác (Accuracy) và điểm F1 (weighted F1 score).

Độ chính xác (accuracy) được đo bằng số câu bình luận được gán nhãn đúng chia tổng số câu bình luận

Độ chính xác = Số bình luận gán nhãn đúng/ Tổng số bình luận

Điểm F1 là trung bình điều hòa của Điểm chính xác (Precision - P) và Độ hồi tưởng (Recall - R). Trong đó, P và R được cho bởi công thức sau:

P = Tổng số bình luận được phân loại đúng của một nhãn / Tổng số bình luận được phân loại nhãn đó

R = Tổng số bình luận được phân loại đúng của một nhãn / Tổng số bình luận thực sự thuộc nhãn đó

Điểm khác nhau giữa P và R ở chỗ P được đánh giá trên nhãn do máy phân loại, còn R được đánh giá trên nhãn do người thu thập từ trước.   
Điểm F1 là trung bình điều hòa (harmonic mean) và được xác định bởi công thức sau:

Lí do chúng em sử dụng thêm Điểm F1 để đánh giá mô hình là vì điểm Độ chính xác (accuracy) có thể không phù hợp trên bộ dữ liệu bị mất cân bằng. Độ chính xác chỉ là trung bình số mẫu được gán nhãn đúng chia cho tổng số mẫu. Trong trường hợp mô hình phân loại được lớp đa số đúng nhưng lớp thiểu số tồi thì vẫn có được độ chính xác cao. Điều này là rất nguy hiểm, chẳng hạn trong bài toán phân loại bệnh nhân có bị nhiễm trùng huyết hay không trong số những bệnh nhân được theo dõi ở khoa hồi sức tích cực, số mẫu bị bệnh rất nhỏ (thường chiếm ~5%) trong khi các mẫu không bị bệnh rất lớn (thường chiếm 95%). Vì vậy, chỉ cần mô hình phân loại tất cả bệnh nhân vào lớp không bị bệnh là đã đạt được accuracy 95% trong khi lớp mà chúng ta quan tâm là lớp bị bệnh (lớp thiểu số). Do đó, trong các bài toán phân loại, tiêu chí Điểm F1 cần được sử dụng thêm bên cạnh Độ chính xác để đánh giá độ hiệu quả của mô hình.

## **5.3. Môi trường thực nghiệm**

Môi trường thực nghiệm của đồ án này được thực hiện trên môi trường Google Colab. Phiên bản Python được sử dụng là 3.7 cùng một số thư viện đi kèm như sau: *sklearn, numpy, pandas, underthesea, transformer, torch, fasttext,…*

Chúng em sử dụng 3 phương pháp Word Embedding là: fastText, PhoBERT, Tf-Idf. Các thuật toán AI khác nhau được cài đặt cho bài toán phân loại này với đầu vào là các vector sinh ra từ các phương pháp Embedding kể trên.

## **5.4. Kết quả thực nghiệm**

Kết quả thực nghiệm của bài toán Phân loại cảm xúc bình luận trên sàn thương mại điện tử được trình bày trong 2 bảng 5.3 và 5.4 với tập dữ liệu đầu vào của 2 bảng là khác nhau.

**Phân tích kết quả**: Đồ án này sử dụng dữ liệu đã được tiền xử lý bằng các phương pháp đã được trình bày ở mục 4. Dữ liệu được chia sử dụng 2 phương pháp đó là Hold-out,Repeated Hold-out và KFold. Trong đó 90% dữ liệu làm tập Training và 10% dữ liệu làm tập Validation. Dữ liệu trong 2 bảng là dữ liệu trước và sau khi thực hiện giảm số mẫu Tích cực (lớp đa số). Trên mỗi tập dữ liệu, chúng em cài đặt các thuật toán bao gồm cả học máy và học sâu trên các kịch bản Word Embedding khác nhau: fastText, PhoBERT và Tf-Idf.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Phương pháp Embedding dữ liệu đã qua tiền xử lý | fastText | | PhoBERT | | [Tf-Idf](https://colab.research.google.com/drive/17aokwgrOb11pPxwW4I2EzrVgnPtMa5gq?usp=sharing) | |
| Mô hình | Độ chính xác (%) | Điểm F1 (%) | Độ chính xác (%) | Điểm F1 (%) | Độ chính xác (%) | Điểm F1 (%) |
| [Nhóm các thuật toán học máy](https://colab.research.google.com/drive/1IRMQxZ1bcZazkvnN03MysddRaNOTjjFu?authuser=1#scrollTo=wc1GE8wsOMzX) | | | | | | |
| Hồi quy Logistic | 76.0 | 71.0 | 80.7 | **78.4** | **81.1** | 77.5 |
| Máy véc-tơ hỗ trợ (SVM) | **77.2** | **72.2** | **80.8** | 77.0 | 80.8 | 76.6 |
| XGBoost | 75.3 | 71.0 | 79.0 | 76.1 | 80.1 | 75.2 |
| Random Forest | 75.2 | 69.1 | 79.2 | 75.8 | 78.4 | 73.1 |
| LightGBM | 76.5 | 72.1 | 80.4 | 77.5 | 80.6 | 77.9 |
| [Thuật toán học sâu](https://colab.research.google.com/drive/1FdT8UArxFH1qHTncLhQodtEgB1hjAPVa?authuser=1#scrollTo=oMB0y7nzwKtN) | | | | | | |
| RoBERTa  (3 epochs) | - | - | 80.1 | 75.6 | - | - |

Bảng 5.3. Kết quả chạy các thuật toán trên bộ dữ liệu gốc

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Phương pháp Embedding dữ liệu đã qua tiền xử lý | fastText | | PhoBERT | | Tf-Idf | |
| Mô hình | Độ chính xác (%) | Điểm F1 (%) | Độ chính xác (%) | Điểm F1 (%) | Độ chính xác (%) | Điểm F1 (%) |
| [Nhóm các thuật toán học máy](https://colab.research.google.com/drive/1eCXAvhtXFmRFm_SxsgJPry1EBBV2bNu2?authuser=1#scrollTo=yWLUREQ8P70Z) | | | | | | |
| Hồi quy Logistic | 67.9 | 63.5 | 69.8 | **67.9** | 73.9 | 71.0 |
| Máy véc-tơ hỗ trợ (SVM) | **69.2** | **64.6** | 70.5 | 67.6 | **74.0** | 70.0 |
| XGBoost | 63.9 | 61.4 | 65.3 | 63.0 | 71.5 | 68.7 |
| Random Forest | 65.6 | 61.2 | 68.8 | 64.3 | 70.6 | 66.5 |
| LightGBM | 67.0 | 63.3 | 69.2 | 66.8 | 73.3 | **72.0** |
| [Thuật toán học sâu](https://colab.research.google.com/drive/1RNU8_aauZOPZO6bskM9KumGrJX8kMaeG?authuser=1#scrollTo=oMB0y7nzwKtN) | | | | | | |
| RoBERTa  (4 epochs) | - | - | **71.8** | 67.8 | - | - |

Bảng 5.4 Kết quả chạy các thuật toán trên bộ dữ liệu đã Undersampling

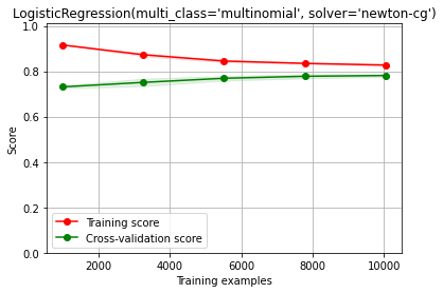
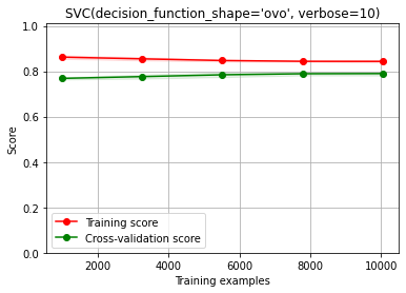
*Chú thích: Đường liên kết tới mã nguồn Google Colab đã được đính kèm trong mỗi bảng*

Với mỗi cột trong bảng, kết quả của thuật toán học máy tốt nhất sẽ được bôi đậm. Từ 2 bảng trên, nhận thấy phương pháp biểu diễn văn bản bình luận tiếng Việt trong không gian véc-tơ đóng vai trò không nhỏ để mô hình có được kết quả tốt. Mô hình PhoBERT và Tf-Idf đem lại kết quả tốt hơn đáng kể so với phương pháp biểu diễn theo mô hình fastText.

Mô hình BERT hoạt động hiệu quả trong các trường hợp cần hiểu rõ ngữ cảnh của văn bản như trong dịch máy, phân loại văn bản dài, tóm tắt văn bản,… Tuy nhiên, trong bài toán phân loại sắc thái bình luận, các câu bình luận ngắn và không mang nhiều đặc trưng về ngữ cảnh nên sử dụng BERT chưa cho kết quả độ chính xác tốt nhất. Việc biểu diễn bình luận trong không gian véc-tơ có thể sử dụng Tf-Idf hiệu quả hơn vì phương pháp này dựa trên tần suất xuất hiện của từ chứ không quan tâm về ngữ cảnh, phù hợp hơn cho bài toán phân loại các câu bình luận ngắn, lặp lại, xuất hiện các từ đặc trưng để phân loại cảm xúc. Hơn nữa thời gian chạy thuật toán véc-tơ hóa của mô hình Tf-Idf nhanh hơn rất nhiều so với 2 mô hình còn lại, điều này là do mô hình Tf-Idf chỉ yêu cầu tính toán trên các chỉ số đơn giản.

Mặc dù vậy, mô hình hóa bình luận bằng PhoBERT vẫn cho kết quả tốt trong một số trường hợp. Cụ thể, từ bảng 2 ta có thể thấy điểm Độ chính xác tốt nhất của PhoBERT là 80.8%, không khác biệt quá nhiều so với Độ chính xác tốt nhất của Tf-Idf là 81%. Tuy nhiên, điểm F1 của PhoBERT là 78.4%, tốt hơn so với Tf-Idf là 77.9%. Ngoài ra, điểm F1 của PhoBERT cũng tốt hơn Tf-Idf và fastText trên nhiều thuật toán khác nhau được thể hiện trong bảng 2. Điều này có thể chứng minh rằng việc học ngữ cảnh của PhoBERT hoạt động hiệu quả trên bộ dữ liệu mất cân bằng, giúp phân loại các câu vào đúng nhãn, kể cả lớp thiểu số và đa số.

Câu hỏi đặt ra là tại sao mô hình đơn giản như Tf-Idf lại chiếm ưu thế và phoBERT lại tỏ ra chưa có gì vượt trội rõ rệt so với Tf-Idf? Giả thuyết của chúng em đặt ra cho hiện tượng này đó là do đặc tính của bộ dữ liệu. Có thể là vì bộ dữ liệu còn nhỏ nên dẫn đến hiện tượng overfit nhanh chóng khi áp dụng các thuật toán học sâu với hàng triệu tham số. Vì vậy chúng em đã phân tích điểm huấn luyện và kiểm tra khi kích thước mẫu tăng dần.

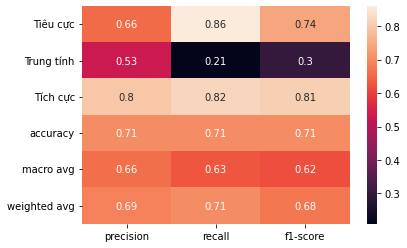
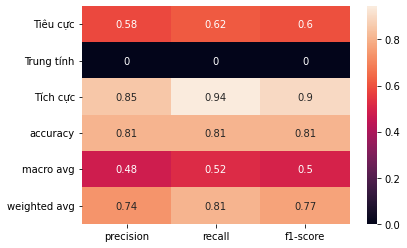


Hình 5.3. Điểm huyến luyện và kiểm tra chéo khi kích thước mẫu tăng dần

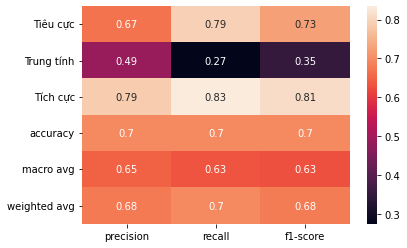
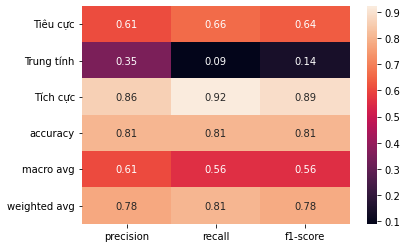
Kết quả cho thấy, lợi ích nhận được khi tăng số mẫu là không đáng kể. Vì vậy, việc bộ dữ liệu nhỏ không phải vấn đề chính, giả thuyết trên có thể được bác bỏ. Lí do sẽ được phân tích kĩ hơn ở phần Phân tích lỗi.

Trong mô hình fastText, việc biểu diễn véc-tơ từng từ, cụm từ sẽ dựa trên ngữ cảnh của các từ lân cận của nó, trong khi PhoBert học ý nghĩa cả câu để biểu diễn véc-tơ (các từ giống nhau nhưng khi xuất hiện ở các ngữ cảnh khác nhau sẽ có véc-tơ khác nhau). Do đó, kết quả thực hiện trên mô hình PhoBert tốt hơn đáng kể so với mô hình fastText.

Sau đây, chúng em sẽ xem xét kết quả của các thuật toán đối với việc phân loại trên từng nhãn thông qua các biểu đồ ma trận lỗi. Trong đó, mỗi hình thể hiện kết quả một thuật toán khi thực hiện trên bộ dữ liệu gốc (bên trái) và bộ dữ liệu sau khi áp dụng Undersampling (bên phải).



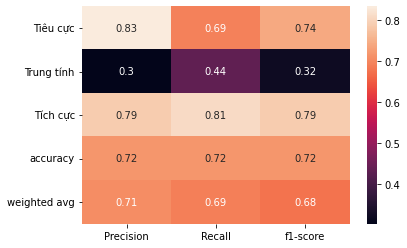
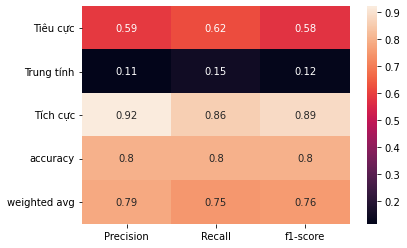
Hinh 5.4. SVM



Hinh 5.5. Logistic Regression

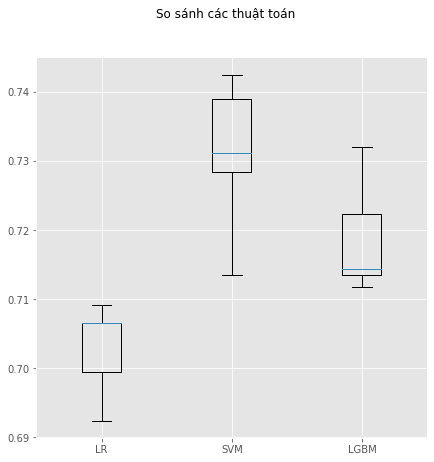
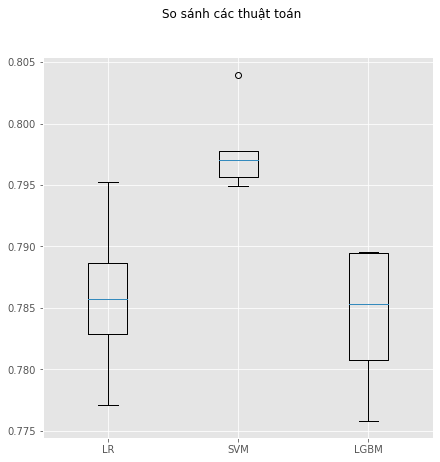
Hai giải thuật học máy đạt kết quả tốt nhất là SVM và Hồi quy Logistic, trong đó SVM cho độ chính xác tốt hơn các thuật toán khác trong nhiều trường hợp. Tuy nhiên, nhược điểm của mô hình SVM là kết quả phân loại lớp thiểu số rất nhỏ. Điều này được thể hiện trong hình 5.4 phía bên trái, SVM bỏ qua hoàn toàn việc phân loại nhãn Trung tính vì đây là nhóm thiểu số. Mô hình học máy Logistic và học sâu RoBERTa thì cho kết quả tốt hơn trên lớp Trung tính. Ngoài ra, chúng em đã áp dụng thêm phương pháp Dropout trên mô hình RoBERTa, giúp tránh hiện tượng overfitting thường gặp với các thuật toán học sâu trên bộ dữ liệu nhỏ. Trong đó, mô hình hồi quy Logistics cho thấy kết quả tốt trên mọi lớp trong các kịch bản khác nhau.

Một điểm đáng chú ý khác, trên bộ dữ liệu mất cân bằng, các thuật toán có xu hướng bỏ qua lớp thiểu số (Trung tính) và nâng cao hiệu quả các lớp còn lại. Vì vậy, việc tái cân bằng tập dữ liệu là cần thiết. Kết quả cho thấy, sau khi áp dụng phương pháp Undersampling thì điểm F1 của các thuật toán trên các lớp thiểu số đều tăng đáng kể. Cụ thể, đối với SVM điểm F1 của nhãn Trung tính tăng từ 0 lên 0.3, điểm F1 của nhãn Tiêu cực tăng từ 0.6 lên 0.74, điểm precision của nhãn Trung tính tăng từ 0 lên 0.53. Điều này cũng đúng trên các thuật toán còn lại. Tuy nhiên, vấn đề gặp phải của phương pháp này đó là độ chính xác của thuật toán sẽ giảm, do độ chính xác của lớp đa số - lớp đóng góp chính giảm. Cụ thể, điểm F1 của SVM giảm từ 0.9 xuống 0.81 trên nhãn Tích cực. Điều này có thể do việc phân loại nhầm giữa nhãn Tích cực với nhãn Trung tính khi ta đưa số mẫu của 2 nhãn về gần nhau hơn. Như vậy, việc nâng cao độ chính xác khi phân loại lớp thiểu số mà vẫn đảm bảo được độ chính xác của lớp đa số là sự đánh đổi (trade-off).



Hình 5.6. RoBERTa

Ngoài ra, chúng em so sánh kết quả của 3 thuật toán học máy cho điểm F1 và accuracy tốt nhất khi áp dụng KFold: Hồi quy Logistic, SVM và LightGBM



Hình 5.7. Sử dụng KFold với k = 5

Hình 5 so sánh độ chính xác (accuracy) 3 thuật toán hồi quy Logistic, SVM và LightGBM trên 2 bộ dữ liệu sử dụng KFold Cross Validation và mô hình PhoBERT để Embedding. Kết quả cho thấy SVM tốt hơn so với 2 thuật toán còn lại.

**Phân tích lỗi:** Qua việc chạy thực nghiệm, chúng em đã nhận ra những trường hợp phân tích quan điểm mà mô hình phân loại sai như sau: Lỗi phân loại văn bản nhầm lẫn giữa nhãn Trung tính và nhãn Tiêu cực. Điều này xuất phát từ việc dãn nhãn dữ liệu ở bộ dữ liệu gốc. Do chi phí đọc, dãn nhãn hơn 10000 câu bình luận rất lớn nên chúng em đã chia nhãn từ các câu bình luận được đánh giá từ 1 đến 5 sao thành các câu được dán nhãn Tích cực, Trung tính và Tiêu cực như sau:  
Tích cực: 5 sao, 4 sao

Trung tính: 3 sao

Tiêu cực: 2 sao, 1 sao

Điều này có thể gây ra nhầm lẫn giữa nhãn Tiêu cực và Trung tính bởi vì không có sự phân biệt quá rõ ràng giữa các nhãn 1 sao, 2 sao và 3 sao. Chẳng hạn, bình luận được đánh giá 3 sao trên thực tế có thể mang nghĩa tiêu cực nhưng lại được dãn nhãn Trung tính. Hoặc ngược lại, bình luận được đánh giá 2 sao có thể mang nghĩa Trung tính (vừa khen vừa chê) nhưng lại được dán nhãn Tiêu cực.

Ngoài ra, có một số bình luận được đánh số sao không đúng với nội dung, chẳng hạn “sản phẩm ok” nhưng chỉ đánh giá 3 sao và được cho vào lớp Trung tính thay vì lớp Tích cực. Hoặc “ship chậm hàng hơi bị móp” nhưng vẫn đánh giá 4 sao và được cho vào lớp Tích cực thay vì lớp Tiêu cực. Việc dán nhãn còn mang tính tự phát và phụ thuộc nhiều vào cảm quan của người dùng sàn thương mại điện tử nên nhiều nhãn bị sai lệch. Chất lượng bộ dữ liệu chưa đủ tốt dẫn đến việc các thuật toán học máy học sai lệch theo bộ dữ liệu đào tào và phụ thuộc vào việc phân chia giữa tập đào tạo và tập kiểm tra.

Bên cạnh đó, một số câu bình luận vừa mang nghĩa tích cực, vừa mang nghĩa tiêu cực sẽ gây khó khăn cho việc phân loại, chẳng hạn câu “tiki giao hàng nguyên tem đóng gói kỹ sản phẩm chính hãng chỉ có cái là chất lượng vỏ máy hoàn thiện kém ọp ẹp” (Tích cực) thì vế “tiki giao hàng nguyên tem đóng gói kỹ sản phẩm chính hãng” mang nghĩa tích cực, nhưng vế sau lại mang nghĩa tiêu cực. Việc quyết định đây nhãn là Tích cực hay Tiêu cực là rất khó khăn, kể cả đối với con người.

Vì dữ liệu được dán nhãn chưa chính xác dẫn đến các thuật toán học dựa trên ngữ nghĩa, ngữ cảnh sẽ hiểu sai ý nghĩa văn bản và mất ổn định trong việc dự đoán. Ngoài ra, một số bình luận có thể vô nghĩa, lặp lại nhiều lẫn dẫn đến việc áp dụng các mô hình véc-tơ hóa từ hiểu ngữ cảnh, ngữ nghĩa như PhoBERT hay fastText có thể không tốt bằng mô hình dựa trên tần suất như Tf-Idf thể hiện trong một số kịch bản thực nghiệm.

# **Chương 6: Kết luận**

Đồ án đã nghiên cứu những phương pháp, hướng giải quyết khác nhau để giải quyết bài toán **Phân loại cảm xúc bình luận trên sàn thương mại điện tử.** Bài toán được thực hiện trên dữ liệu văn bản bình luận thu thập từ các sàn thương mại điện tử như Tiki, Lazada và sử dụng những tiêu chí khác nhau để thực hiện việc gán nhãn tiêu cực, tích cực hoặc trung tính cho từng văn bản. Sau khi có được tập dữ liệu phù, dữ liệu sẽ được biểu diễn dưới dạng véc-tơ thông qua các phương pháp véc-tơ hóa khác nhau. Sau đó, các thuật toán học máy và học sâu được áp dụng và cho thấy kết quả khả quan với điểm độ chính xác và điểm F1 đều lớn hơn 70%, trong đó, mô hình đạt điểm cao nhất là hồi quy Logistic với điểm F1 trên phương pháp véc-tơ hóa phoBERT là 78.4% và điểm độ chính xác trên phương pháp véc-tơ hóa Tf-Idf là 81%.

Sau quá trình xây dựng mô hình và tiến hành hiệu chỉnh và đánh giá có thể thấy việc quay lại xử lí dữ liệu, xây dựng lại mô hình, thay đổi tham số có vai trò rất quan trọng trong việc có được một mô hình tốt. Do đó chúng ta luôn cần phải học hỏi, phát triển mô hình, khai thác dữ liệu để có thể có được những kết quả tối ưu.

Do bộ dữ liệu còn hạn chế và chưa được dán nhãn thật sự đúng đắn, độ chính xác của các mô hình học máy trên tập validation vẫn còn chưa ưu tú và cần được cải thiện. Trong tương lai, để có thể đưa ra train được những model tốt hơn thì sẽ cần gán nhãn dữ liệu nhiều hơn, chuẩn xác hơn. Ngoài ra còn có thể thử nghiệm thêm nhiều thuật toán để có thể tìm ra thuật toán tối ưu nhất cho bài toán này.

# **Tài liệu tham khảo**

|  |  |
| --- | --- |
| [1] | V. Kharde, P. Sonawane et al., “Sentiment analysis of twitter data: a survey of techniques,” arXiv preprint arXiv:1601.06971, 2016. |
| [2] | Q. T. Nguyen, T. L. Nguyen, N. H. Luong, and Q. H. Ngo, “Fine-tuning bert for sentiment analysis of vietnamese reviews,” in 2020 7th NAFOSTED Conference on Information and Computer Science (NICS). IEEE, 2020, pp. 302–307. |
| [3] | S. N. Hoang, L. V. Nguyen, T. Huynh, and V. T. Pham, “An efficient model for sentiment analysis of electronic product reviews in vietnamese,” in International Conference on Future Data and Security Engineering. Springer, 2019, pp. 132–142. |
| [4] | X.-S. Vu and S.-B. Park, “Construction of vietnamese sentiwordnet by using vietnamese dictionary,” arXiv preprint arXiv:1412.8010, 2014. |