BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC PHENIKAA**


BÀI TẬP LỚN

# LẬP TRÌNH PYTHON CHO KHOA HỌC DỮ LIỆU

**Đề tài:**

# PHÂN TÍCH TẬP DỮ LIỆU PHẢN HỒI CỦA SINH VIÊN ĐẠI HỌC

Học viên thực hiện: Lê Hoàng Lân - 24800017

Lớp Khai phá dữ liệu và học máy 5-1-24 (N01)

Giảng viên hướng dẫn: TS. Nguyễn Thanh Bình

Hà Nội, tháng 06 năm 2025

# MỤC LỤC

# 1. Giới thiệu

Phân tích cảm xúc (Sentiment Analysis) là một lĩnh vực nghiên cứu trong xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP) nhằm xác định và phân loại các ý kiến, cảm xúc được thể hiện trong văn bản. Với sự gia tăng của các phản hồi trực tuyến, đặc biệt từ sinh viên trên các nền tảng giáo dục, lượng dữ liệu văn bản chứa thông tin về quan điểm và cảm xúc ngày càng phong phú. Điều này mở ra cơ hội lớn để phân tích và khai thác giá trị từ những dữ liệu này, đồng thời đặt ra thách thức về việc xây dựng các mô hình hiệu quả để xử lý tự động.

Trong bối cảnh đó, các thuật toán học máy và học sâu đã được áp dụng để tự động hóa quá trình phân tích cảm xúc và phân loại chủ đề. Trong số đó, Multinomial Naive Bayes nổi bật như một phương pháp học máy đơn giản nhưng hiệu quả, phù hợp cho phân loại văn bản nhờ khả năng xử lý tập dữ liệu lớn và tốc độ huấn luyện nhanh. Ngoài ra, các mô hình như Support Vector Machine (SVC), Long Short-Term Memory (LSTM), và mô hình ngôn ngữ lớn đã được huấn luyện trước như PhoBERT cũng được khám phá để so sánh hiệu quả. Dù Naive Bayes giả định tính độc lập giữa các đặc trưng, nó vẫn mang lại kết quả đáng tin cậy trong nhiều ứng dụng thực tế, trong khi các mô hình phức tạp hơn như PhoBERT tận dụng biểu diễn ngữ cảnh để nâng cao hiệu suất.

Bài tập lớn này tập trung vào việc áp dụng và so sánh các mô hình học máy và học sâu để phân loại chủ đề và cảm xúc từ tập dữ liệu phản hồi sinh viên (UIT-VSFC). Cụ thể, ta xây dựng các mô hình để phân loại phản hồi thành bốn chủ đề (giảng viên, chương trình đào tạo, cơ sở vật chất, các vấn đề khác) và ba mức cảm xúc (tiêu cực, trung tính, tích cực). Quá trình thực nghiệm bao gồm các bước tiền xử lý dữ liệu như phân đoạn từ bằng py_vncorenlp, loại bỏ dấu câu, và chuẩn hóa văn bản, sau đó huấn luyện các mô hình Multinomial Naive Bayes, SVC, LSTM, và PhoBERT. Ngoài ra, ta đã thử nghiệm chia lại dữ liệu với tỷ lệ mới (7:1:2) và đánh giá hiệu suất thông qua các chỉ số như Precision, Recall, F1-score, và Accuracy.

Bài tập lớn được tổ chức thành các phần chính bao gồm: Giới thiệu về bối cảnh và mục tiêu, tổng quan về các mô hình liên quan, chi tiết quá trình thực nghiệm từ tiền xử lý dữ liệu đến huấn luyện và đánh giá mô hình, so sánh hiệu quả giữa các mô hình, và cuối cùng là kết luận cùng các hướng mở rộng trong tương lai. Với các phân tích và kết quả thực nghiệm, nghiên cứu này hy vọng cung cấp cái nhìn sâu sắc về việc áp dụng các thuật toán học máy và học sâu trong phân tích

phản hồi giáo dục, đồng thời đề xuất các hướng cải thiện như phát hiện phản hồi quan trọng hoặc tăng cường dữ liệu để xử lý mất cân bằng.

# 2. Dữ liệu và tiền xử lý dữ liệu

## 2.1 Giới thiệu tập dữ liệu

Bộ dữ liệu được sử dụng trong bài toán là "Vietnamese Students' Feedback Corpus" (UIT-VSFC), được xây dựng bởi nhóm UIT NLP của Đại học Công nghệ Thông tin – ĐHQG TP.HCM [1]. Dữ liệu bao gồm 16.223 phản hồi (feedback) của sinh viên bằng tiếng Việt, thu thập qua khảo sát đánh giá học phần tại trường.

Mỗi phản hồi trong tập dữ liệu là một đơn vị văn bản ngắn, thường bao gồm một đến ba câu, được gán hai nhãn:

- Sentiment: Sắc thái của phản hồi, nhận một trong ba giá trị 0, 1, và 2, tương ứng với các sắc thái tiêu cực, trung tính, và tích cực.
- Topic: Chủ đề của phản hồi, với các giá trị 0 (giảng viên), 1 (chương trình học), 2 (cơ sở vật chất), và 3 (các vấn đề khác).

Tập dữ liệu được gán nhãn thủ công với độ chính xác cao, không chứa trường khuyết dữ liệu.

## 2.2 Tiền xử lý dữ liệu

Trước khi đưa dữ liệu vào mô hình học máy hoặc học sâu, cần thực hiện một số bước tiền xử lý cơ bản để chuẩn hóa văn bản và giảm nhiễu. Mục này sẽ trình bày các bước tiền xử lý dữ liệu văn bản. Kết quả tiền xử lý dữ liệu được minh họa trên Bảng 2.1

### *2.2.1 Chuyển văn bản về dạng chữ viết thường*

Việc chuyển văn bản về dạng chữ viết thường (lowercase) giúp chuẩn hóa dữ liệu, loại bỏ sự khác biệt giữa các ký tự in hoa và thường, từ đó giảm độ phức tạp khi xử lý văn bản. Trong tập dữ liệu UIT-VSFC, các phản hồi của sinh viên có thể chứa các ký tự in hoa không nhất quán (ví dụ: "Giảng viên" hoặc "giảng viên"). Việc chuẩn hóa này đảm bảo rằng các từ giống nhau được xử lý như một thực thể duy nhất, tránh sự phân biệt không cần thiết trong quá trình huấn luyện mô hình.

### *2.2.2 Loại bỏ dấu câu*

Dấu câu (như dấu chấm, dấu phẩy, dấu chấm than, v.v.) thường không mang ý nghĩa ngữ nghĩa quan trọng trong các bài toán phân tích văn bản, đặc biệt là phân loại sắc thái hoặc chủ đề. Việc loại bỏ dấu câu giúp giảm nhiễu, đơn giản hóa văn

**Bảng 2.1 Tập dữ liệu sau khi tiền xử lý**

| ID | Sentence | Topic | Sentiment | Dataset | Tokens |
|---|---|---|---|---|---|
| 0 | slide giáo trình đầy đủ . | 1 | 2 | train | [slide, giáo_trình, đầy_đủ] |
| 1 | nhiệt tình giảng dạy , gần gũi với sinh viên . | 0 | 2 | train | [nhiệt_tình, giảng_dạy, gần_gũi, sinh_viên] |
| 2 | đi học đầy đủ full điểm chuyên cần . | 1 | 0 | train | [đi, học, đầy_đủ, full, chuyên_cần] |
| 3 | chưa áp dụng công nghệ thông tin và các thiết bị hỗ trợ giảng dạy . | 0 | 0 | train | [áp_dụng, công_nghệ, thông_tin, thiết_bị, giảng_dạy] |
| 4 | thầy giảng bài hay , có nhiều bài tập ví dụ ngay trên lớp . | 0 | 2 | train | [thầy, giảng, bài_tập, ví_dụ, lớp] |

bản và tập trung vào các từ mang nội dung chính. Trong tập dữ liệu UIT-VSFC, các phản hồi ngắn thường chứa dấu câu không nhất quán, do đó bước này giúp cải thiện chất lượng dữ liệu đầu vào cho mô hình.

### *2.2.3 Tách từ*

Trong tiếng Việt, các từ thường không được phân tách bằng dấu cách rõ ràng như tiếng Anh, mà thường là các âm tiết ghép lại thành từ có nghĩa (ví dụ: "giảng viên" là một từ ghép, không phải hai từ riêng lẻ "giảng" và "viên"). Tách từ (word segmentation) là bước quan trọng để chuyển các câu thành danh sách các từ có nghĩa, giúp mô hình học máy hiểu đúng cấu trúc ngữ nghĩa của văn bản. Đối với tập dữ liệu UIT-VSFC, việc tách từ chính xác là cần thiết để đảm bảo các từ như "cơ sở vật chất" được xử lý như một đơn vị hoàn chỉnh.

### *2.2.4 Loại bỏ stopwords*

Stopwords là các từ phổ biến nhưng ít mang ý nghĩa ngữ nghĩa trong văn bản (ví dụ: "là", "của", "và", "rất"). Loại bỏ stopwords giúp giảm kích thước dữ liệu, tập trung vào các từ mang nội dung quan trọng và cải thiện hiệu suất của mô

hình phân loại. Trong tập dữ liệu UIT-VSFC, các phản hồi ngắn có thể chứa nhiều stopwords, do đó việc loại bỏ chúng giúp mô hình tập trung vào các từ khóa liên quan đến sắc thái hoặc chủ đề (như “giảng viên”, “cơ sở vật chất”).

## 2.3 Khám phá dữ liệu

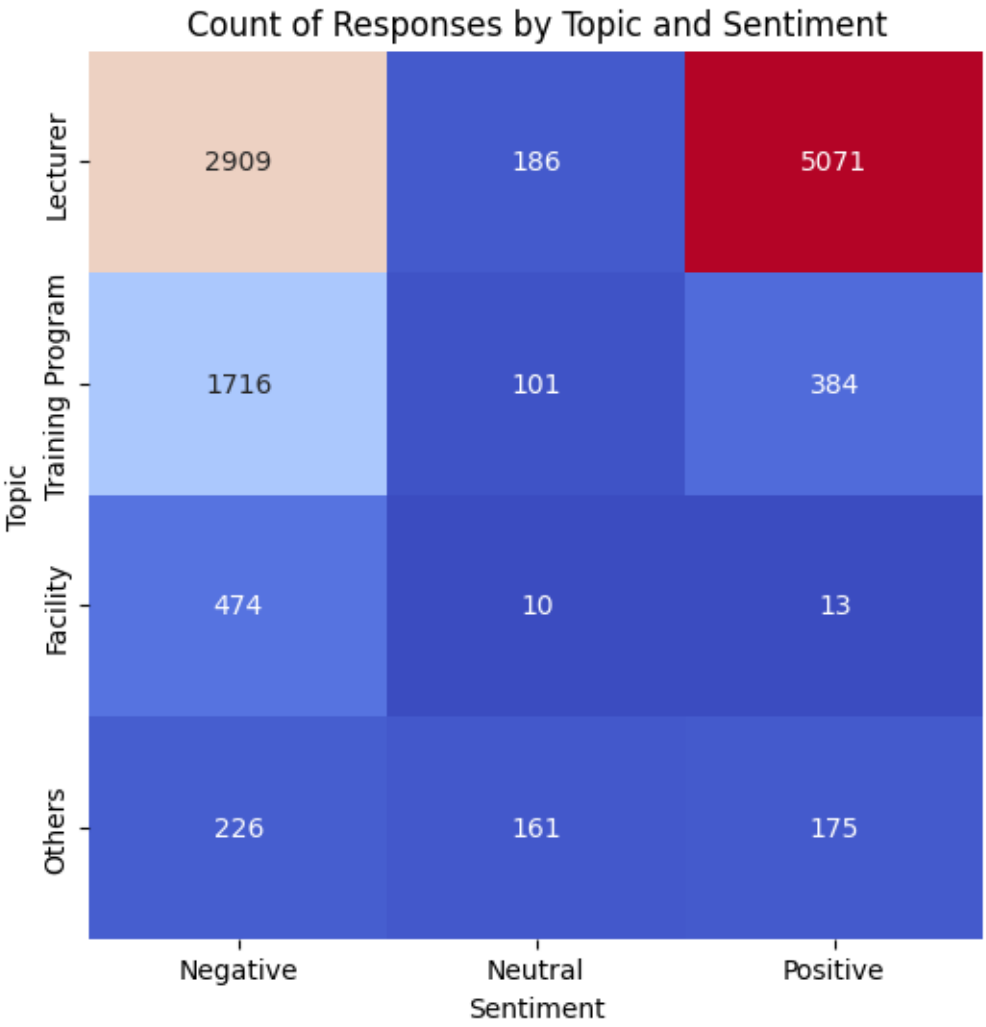
Để hiểu rõ hơn về tập dữ liệu UIT-VSFC sau khi tiền xử lý, chúng ta thực hiện phân tích khám phá dữ liệu (Exploratory Data Analysis - EDA) trên tập huấn luyện. Tập dữ liệu này bao gồm tổng cộng 16.175 mẫu, với 11.426 mẫu trong tập train, 1.583 mẫu trong tập valid và 3.166 mẫu trong tập test. Các bước phân tích dưới đây giúp làm sáng tỏ cấu trúc và đặc điểm của dữ liệu trước khi huấn luyện mô hình.

### *2.3.1 Phân tích phân bố chủ đề và sắc thái*

Để hiểu rõ hơn về phân bố của các chủ đề và sắc thái trong tập dữ liệu huấn luyện, chúng ta thực hiện thống kê số lượng mẫu theo từng chủ đề và sắc thái. Kết quả được thể hiện trong Hình 2.1 dưới đây, cho thấy số lượng phản hồi được phân loại theo bốn chủ đề (giảng viên, chương trình học, cơ sở vật chất, và các vấn đề khác) và ba sắc thái (tiêu cực, trung tính, tích cực).

Từ Hình 2.1, có thể thấy rằng chủ đề "Giảng viên" chiếm số lượng phản hồi lớn nhất, đặc biệt với sắc thái tích cực (5071 mẫu), cho thấy sinh viên có xu hướng đánh giá tích cực về giảng viên. Ngược lại, chủ đề "Cơ sở vật chất" có ít mẫu nhất, đặc biệt là các phản hồi trung tính (10 mẫu) và tích cực (13 mẫu), điều này có thể phản ánh rằng sinh viên ít đề cập đến cơ sở vật chất hoặc các đánh giá về chủ đề này thường mang tính tiêu cực. Sự mất cân bằng này cần được lưu ý khi xây dựng mô hình để tránh thiên lệch.

Để trực quan hóa phân bố của các chủ đề và sắc thái, biểu đồ cột được sử dụng để thể hiện số lượng mẫu trong tập huấn luyện theo từng danh mục. Biểu đồ bên trái trong Hình 2.2 cho thấy phân bố của các chủ đề, với "Giảng viên" chiếm ưu thế vượt trội, tiếp theo là "Chương trình học", trong khi "Cơ sở vật chất" và "Các vấn đề khác" có số lượng mẫu ít hơn đáng kể. Biểu đồ bên phải thể hiện phân bố sắc thái, với sắc thái tích cực chiếm ưu thế, tiếp theo là tiêu cực, và trung tính có số lượng ít nhất. Sự mất cân bằng trong phân bố sắc thái, đặc biệt là số lượng ít ỏi của các phản hồi trung tính, có thể ảnh hưởng đến khả năng dự đoán của mô hình đối với lớp này, đòi hỏi các kỹ thuật xử lý mất cân bằng dữ liệu như tăng cường dữ liệu hoặc điều chỉnh trọng số lớp trong quá trình huấn luyện.



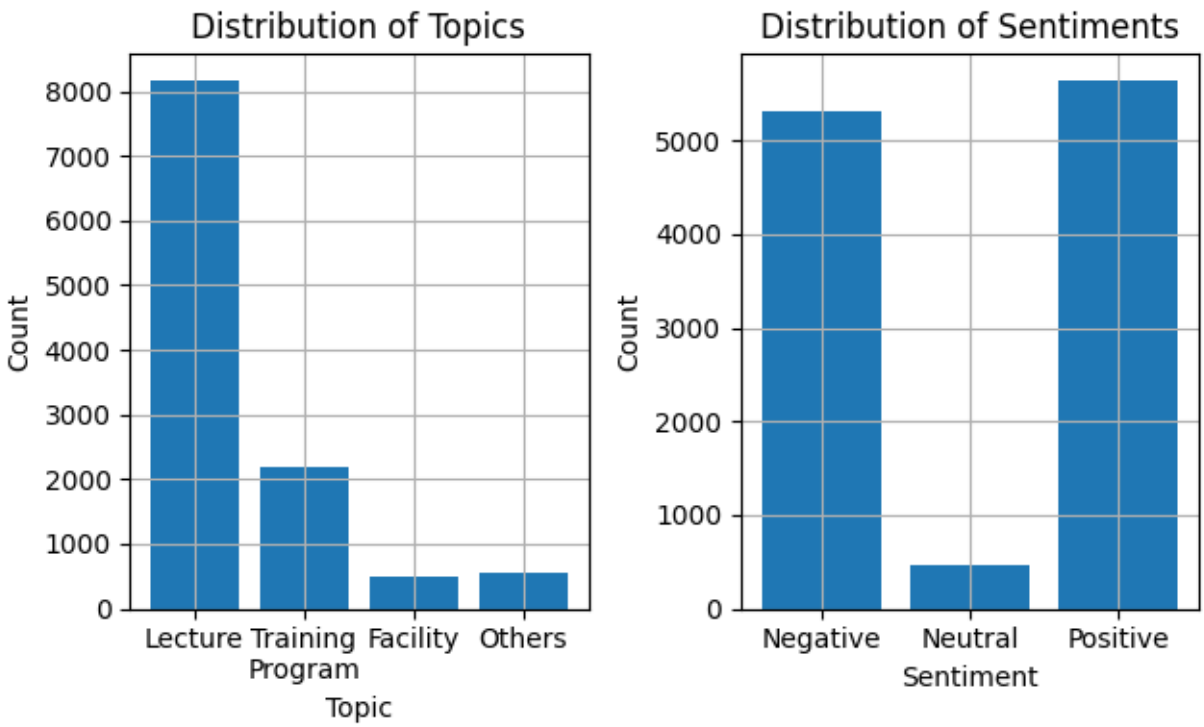

**Hình 2.1 Số lượng phản hồi theo chủ đề và sắc thái**

### *2.3.2 Phân tích tần suất từ*

Để khám phá nội dung các phản hồi trong tập dữ liệu, tần suất xuất hiện của các từ được phân tích trên tập huấn luyện, phân loại theo ba sắc thái: tiêu cực, trung tính và tích cực. Kết quả dưới đây trình bày các từ phổ biến nhất cho mỗi sắc thái, kèm theo nhận xét về ý nghĩa của chúng.

**Phản hồi tiêu cực:**

- Các từ phổ biến: sinh_viên (1314), thầy (1168), học (917), thực_hành (712), dạy (688), giảng_viên (679), bài_tập (624), môn_học (496), giảng (465), lớp (386).
- Nhận xét: Các từ như "sinh_viên", "thầy" và "học" xuất hiện với tần suất cao, cho thấy phản hồi tiêu cực thường liên quan đến trải nghiệm học tập và vai trò của giảng viên. Sự hiện diện của "thực_hành" và "bài_tập" gợi ý rằng sinh




**Hình 2.2 Phân phối số lượng của chủ đề và sắc thái phản hồi**

viên không hài lòng với cách tổ chức các hoạt động thực hành hoặc bài tập, có thể do thiếu hướng dẫn hoặc chất lượng chưa đạt yêu cầu.

**Phản hồi trung tính:**

- Các từ phổ biến: thầy (93), học (47), sinh_viên (34), bài_tập (32), ý_kiến (29), dạy (28), thực_hành (27), lớp (25), giảng_viên (24), môn (20).
- Nhận xét: Tần suất từ trong phản hồi trung tính thấp hơn đáng kể, phản ánh số lượng mẫu hạn chế của lớp này. Từ "ý_kiến" nổi bật, cho thấy các phản hồi trung tính thường mang tính đề xuất hoặc góp ý thay vì đánh giá cảm xúc mạnh mẽ, phù hợp với bản chất trung lập của chúng.

**Phản hồi tích cực:**

- Các từ phổ biến: thầy (1812), nhiệt_tình (1741), dạy (1445), sinh_viên (1332), giảng_viên (906), giảng_dạy (671), giảng (626), kiến_thức (554), tận_tâm (539), bài_tập (392).
- Nhận xét: Các từ như "nhiệt_tình", "tận_tâm" và "kiến_thức" xuất hiện nổi bật, thể hiện sự đánh giá cao về thái độ, sự tận tâm và chất lượng giảng dạy

của giảng viên. Điều này tương ứng với số lượng lớn phản hồi tích cực về chủ đề "Giảng viên" trong Hình 2.2, khẳng định sự hài lòng của sinh viên đối với phương pháp giảng dạy.

Phân tích tần suất từ cho thấy sự khác biệt rõ ràng trong ngôn ngữ sử dụng giữa các sắc thái. Các từ mang tính tích cực như "nhiệt_tình" và "tận_tâm" gần như chỉ xuất hiện trong phản hồi tích cực, trong khi phản hồi tiêu cực thường đề cập đến các vấn đề liên quan đến "thực_hành" và "bài_tập". Những phát hiện này cung cấp thông tin giá trị để lựa chọn đặc trưng văn bản hoặc tối ưu hóa mô hình phân loại, giúp cải thiện khả năng nhận diện sắc thái và chủ đề.

# 3. Các mô hình học máy truyền thống

Phần này trình bày việc sử dụng các mô hình học máy truyền thống để thực hiện hai bài toán phân loại: phân loại chủ đề (topic classification) và phân loại cảm xúc (sentiment classification). Các mô hình được áp dụng bao gồm Multinomial Naive Bayes và Support Vector Machine với kernel tuyến tính, kết hợp với các kỹ thuật vector hóa văn bản để chuyển đổi dữ liệu văn bản thành dạng số phù hợp. Kết quả phân loại được đánh giá trên tập dữ liệu gốc (original splits) bằng các độ đo như precision, recall và f1-score.

## 3.1 Vectorization trong học máy

Trong học máy, đặc biệt là với các bài toán (NLP), việc biểu diễn dữ liệu văn bản dưới dạng số là một bước quan trọng để các mô hình có thể xử lý. Quá trình này được gọi là *vectorization*, chuyển đổi văn bản thô thành các vector số mà các thuật toán học máy có thể hiểu và xử lý. Lý do vectorization cần thiết là vì các mô hình học máy yêu cầu đầu vào là các giá trị số để thực hiện các phép tính toán học, trong khi văn bản thô thường ở dạng chuỗi ký tự không thể sử dụng trực tiếp. Quá trình vectorization được thực hiện như sau:

1. **Ghép các token đã qua xử lý**: Dữ liệu đầu vào là các câu văn bản đã được tiền xử lý, bao gồm việc phân tách thành các token (từ hoặc cụm từ có ý nghĩa). Các token này được ghép lại thành một chuỗi văn bản. Kết quả được lưu trong cột.

2. **Sử dụng vectorizer từ scikit-learn**: Hai phương pháp vectorization được áp dụng là CountVectorizer và TfidfVectorizer từ thư viện scikit-learn.

    - CountVectorizer chuyển đổi văn bản thành ma trận tần số, trong đó mỗi từ (hoặc token) được biểu diễn bằng tần số xuất hiện của nó trong câu. Ma trận này có kích thước $m \times n$, với $m$ là số lượng câu và $n$ là số lượng từ trong từ vựng.
    - TfidfVectorizer tạo ra các vector dựa trên chỉ số TF-IDF (Term Frequency -Inverse Document Frequency), trong đó trọng số của mỗi từ được tính dựa trên tần số xuất hiện trong câu (TF) và nghịch đảo tần số xuất hiện trong toàn bộ tập dữ liệu (IDF). Phương pháp này giúp làm nổi bật các từ quan trọng và giảm ảnh hưởng của các từ phổ biến nhưng ít mang ý nghĩa phân biệt.

Việc vectorization không chỉ giúp mô hình học máy xử lý dữ liệu văn bản mà còn ảnh hưởng trực tiếp đến hiệu suất của mô hình, tùy thuộc vào phương pháp vectorization được chọn.

## 3.2 Giới thiệu các mô hình được sử dụng

Hai mô hình học máy truyền thống được sử dụng trong notebook này là Multinomial Naive Bayes (MultinomialNB) và Support Vector Machine (SVC) với kernel tuyến tính. Dưới đây là mô tả chi tiết về từng mô hình và lý do lựa chọn phương pháp vectorization tương ứng.

### *3.2.1 Multinomial Naive Bayes*

Multinomial Naive Bayes là một thuật toán phân loại dựa trên định lý Bayes, phù hợp cho các bài toán phân loại văn bản với dữ liệu rời rạc. Mô hình này giả định rằng các đặc trưng (từ hoặc token) tuân theo phân phối đa thức (multinomial distribution), và nó thường được sử dụng với dữ liệu là số nguyên, chẳng hạn như tần số xuất hiện của từ. Do đó, CountVectorizer được chọn để vector hóa dữ liệu đầu vào cho mô hình này, vì CountVectorizer tạo ra ma trận tần số với các giá trị nguyên, phù hợp với yêu cầu của MultinomialNB.

### *3.2.2 Support Vector Machine*

Support Vector Machine (SVC) với kernel tuyến tính là một thuật toán phân loại mạnh mẽ, tìm cách tối ưu hóa một siêu phẳng phân tách các lớp dữ liệu với khoảng cách lớn nhất đến các điểm dữ liệu gần nhất (margin). SVC hoạt động tốt với dữ liệu dạng số thực, đặc biệt là các vector có trọng số như TF-IDF, vì chúng cung cấp thông tin về mức độ quan trọng của các từ trong văn bản. Do đó, TfidfVectorizer được sử dụng để vector hóa dữ liệu cho SVC, vì nó tạo ra các vector số thực phản ánh mức độ quan trọng của các từ, giúp cải thiện hiệu suất phân loại của mô hình.

Sự kết hợp giữa MultinomialNB với CountVectorizer và SVC với TfidfVectorizer được thiết kế để tận dụng điểm mạnh của từng mô hình và phương pháp vectorization, đảm bảo phù hợp với đặc điểm của dữ liệu đầu vào và yêu cầu của bài toán phân loại (topic và sentiment

## 3.3 Kết quả

Phần này trình bày kết quả đánh giá hiệu suất của hai mô hình Multinomial Naive Bayes và Support Vector Machine trên tập dữ liệu gốc (original splits) cho

hai bài toán phân loại: phân loại chủ đề (topic) và phân loại cảm xúc (sentiment). Kết quả được đánh giá bằng các độ đo precision, recall, f1-score, cùng với số lượng mẫu (support) cho từng lớp. Các kết quả này được trình bày dưới dạng bảng, tương tự như định dạng của classification report trong Python.

### *3.3.1 Kết quả phân loại chủ đề*

Bảng 3.1 và 3.2 trình bày kết quả phân loại chủ đề của Multinomial Naive Bayes với CountVectorizer và SVC với TfidfVectorizer trên tập test.

**Bảng 3.1 Kết quả phân loại chủ đề với Multinomial Naive Bayes**

| Lớp | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 0 | 0.86 | 0.94 | 0.90 | 2290 |
| 1 | 0.67 | 0.59 | 0.63 | 572 |
| 2 | 0.91 | 0.93 | 0.92 | 145 |
| 3 | 0.60 | 0.02 | 0.04 | 159 |
| Accuracy | | 0.83 | | 3166 |
| Macro Avg | 0.76 | 0.62 | 0.62 | 3166 |
| Weighted Avg | 0.81 | 0.83 | 0.81 | 3166 |

**Bảng 3.2 Kết quả phân loại chủ đề với SVC**

| Lớp | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 0 | 0.88 | 0.93 | 0.90 | 2290 |
| 1 | 0.66 | 0.66 | 0.66 | 572 |
| 2 | 0.89 | 0.90 | 0.89 | 145 |
| 3 | 0.64 | 0.13 | 0.22 | 159 |
| Accuracy | | 0.84 | | 3166 |
| Macro Avg | 0.77 | 0.66 | 0.67 | 3166 |
| Weighted Avg | 0.83 | 0.84 | 0.83 | 3166 |

Kết quả cho thấy SVC với TfidfVectorizer đạt độ chính xác tổng thể (accuracy) cao hơn (0.84) so với Multinomial Naive Bayes với CountVectorizer (0.83). Tuy nhiên, cả hai mô hình đều gặp khó khăn trong việc phân loại lớp 3 (các vấn đề khác), với recall và f1-score rất thấp, đặc biệt là Multinomial Naive Bayes (recall 0.02, f1-score 0.04). Nguyên nhân là lớp có số lượng mẫu hạn chế, bên cạnh đó việc lớp này là tổng hợp các vấn đề nhỏ thay vì một chủ đề thống nhất cũng gây

khó khăn cho phân loại.

### *3.3.2 Kết quả phân loại cảm xúc*

Bảng 3.3 và 3.4 trình bày kết quả phân loại cảm xúc của Multinomial Naive Bayes với CountVectorizer và SVC với TfidfVectorizer trên tập test.

**Bảng 3.3 Kết quả phân loại cảm xúc với Multinomial Naive Bayes**

| Lớp | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 0 | 0.78 | 0.79 | 0.79 | 1409 |
| 1 | 0.00 | 0.00 | 0.00 | 167 |
| 2 | 0.79 | 0.86 | 0.82 | 1590 |
| Accuracy | 0.78 | | | 3166 |
| Macro Avg | 0.52 | 0.55 | 0.54 | 3166 |
| Weighted Avg | 0.74 | 0.78 | 0.76 | 3166 |

**Bảng 3.4 Kết quả phân loại cảm xúc với SVC**

| Lớp | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 0 | 0.76 | 0.85 | 0.80 | 1409 |
| 1 | 0.65 | 0.08 | 0.14 | 167 |
| 2 | 0.83 | 0.82 | 0.82 | 1590 |
| Accuracy | 0.79 | | | 3166 |
| Macro Avg | 0.75 | 0.58 | 0.59 | 3166 |
| Weighted Avg | 0.79 | 0.79 | 0.78 | 3166 |

Tương tự như trên, SVC với TfidfVectorizer đạt độ chính xác tổng thể (accuracy) cao hơn (0.79) so với Multinomial Naive Bayes với CountVectorizer (0.78). Cả hai mô hình đều gặp khó khăn trong việc phân loại lớp 1 (cảm xúc trung tính), với recall và f1-score rất thấp, đặc biệt là Multinomial Naive Bayes (recall 0.00, f1-score 0.00). Lớp này có số lượng mẫu hạn chế so với hai lớp còn lại (167 mẫu), dẫn đến sự mất cân bằng dữ liệu, gây khó khăn cho việc học và dự đoán chính xác.

# 4. Mô hình học sâu

Phần này trình bày việc sử dụng mô hình học sâu, cụ thể là mô hình LSTM (Long Short-Term Memory), để thực hiện các bài toán phân loại chủ đề (topic classification) và phân loại cảm xúc (sentiment classification) trên tập dữ liệu đã tiền xử lý. Các bước chuẩn bị dữ liệu, xây dựng từ điển, tạo Dataset và DataLoader, cùng với kiến trúc mô hình được mô tả chi tiết.

## 4.1 Giới thiệu LSTM

LSTM là một biến thể của mạng nơ-ron hồi quy (Recurrent Neural Network - RNN) được thiết kế để xử lý hiệu quả các chuỗi dữ liệu, đặc biệt là trong các bài toán NLP. Mô hình này phổ biến nhờ khả năng ghi nhớ dài hạn, khắc phục vấn đề mất mát gradient (vanishing gradient) của RNN truyền thống. LSTM sử dụng các cổng để kiểm soát luồng thông tin, cho phép mô hình lưu giữ và cập nhật thông tin quan trọng qua các chuỗi dài. Trong NLP, LSTM được ưa chuộng vì khả năng nắm bắt ngữ cảnh và mối quan hệ giữa các từ trong câu, giúp cải thiện hiệu suất trong các bài toán như phân loại văn bản.

## 4.2 Chuẩn bị dữ liệu

Để áp dụng mô hình LSTM, dữ liệu văn bản cần được chuyển đổi thành định dạng số phù hợp với các thư viện học sâu như PyTorch. Phần này mô tả quá trình tạo từ điển, mã hóa dữ liệu, và xây dựng Dataset cùng DataLoader để huấn luyện mô hình.

### *4.2.1 Tạo từ điển*

Quá trình tạo từ điển (vocabulary) là bước quan trọng để ánh xạ các token đã qua tiền xử lý thành các chỉ số số nguyên (input IDs). Cách thực hiện bao gồm:

1. Lấy tất cả các token duy nhất từ cột tokens của tập huấn luyện, sử dụng hàm explode() để trải phẳng danh sách token.

2. Thêm hai token đặc biệt: <PAD> (chỉ số 0) để đệm các chuỗi có độ dài khác nhau thành cùng kích thước, đảm bảo đồng bộ trong quá trình huấn luyện; và <UNK> (chỉ số 1) để biểu diễn các token hiếm hoặc không xuất hiện trong tập huấn luyện, giúp xử lý các trường hợp chưa biết trong tập kiểm tra hoặc xác thực.

3. Gán một chỉ số số nguyên duy nhất cho mỗi token, bắt đầu từ 2, và lưu vào

một từ điển.

Kích thước từ điển được tính là tổng số token duy nhất trong tập huấn luyện cộng với hai token đặc biệt. Trong bài này, kích thước từ điển là 3320.

### *4.2.2 Tạo Dataset và DataLoader*

Trong PyTorch, Dataset và DataLoader là hai thành phần quan trọng để quản lý và cung cấp dữ liệu cho mô hình học sâu.

- **Dataset**: Lớp TextDataset được định nghĩa để lưu trữ các chuỗi input IDs (đã được mã hóa từ token) và nhãn tương ứng (topic hoặc sentiment). Các chuỗi được đệm (padding) bằng token <PAD> sử dụng hàm pad_sequence để đảm bảo tất cả chuỗi có cùng độ dài, phù hợp cho xử lý batch.
- **DataLoader**: DataLoader được sử dụng để chia dữ liệu thành các batch (kích thước 32 trong bài), tự động xáo trộn dữ liệu huấn luyện (shuffle=True) và cung cấp dữ liệu theo từng batch cho quá trình huấn luyện và đánh giá. Điều này giúp tối ưu hóa việc huấn luyện mô hình trên phần cứng, đặc biệt khi xử lý lượng dữ liệu lớn.

Việc sử dụng Dataset và DataLoader đảm bảo dữ liệu được tổ chức hiệu quả, giảm thiểu chi phí tính toán và hỗ trợ huấn luyện mô hình một cách mượt mà.

## 4.3 Kiến trúc mô hình

Mô hình LSTM được sử dụng trong bài tập này là LSTMClassifier, được xây dựng dựa trên thư viện PyTorch. Kiến trúc bao gồm các thành phần chính:

- **Embedding**: Lớp nhúng (embedding) chuyển các input IDs thành các vector số thực có chiều 128. Đây là cách vector hóa hiện đại, thay thế cho các phương pháp truyền thống như CountVectorizer hay TfidfVectorizer, vì nó học được biểu diễn vector dựa trên ngữ cảnh, giúp mô hình nắm bắt tốt hơn ý nghĩa của từ. Lớp nhúng cũng sử dụng <PAD> để bỏ qua các token đệm trong tính toán.
- **LSTM**: Lớp LSTM với kích thước ẩn (hidden dimension) là 64, nhận các vector nhúng làm đầu vào và xử lý chuỗi để tạo ra biểu diễn ngữ cảnh. LSTM sử dụng cơ chế cổng để giữ thông tin quan trọng qua các bước thời gian.
- **Fully Connected Layer**: Một lớp tuyến tính (linear) ánh xạ đầu ra của LSTM thành số lớp đầu ra (4 cho phân loại chủ đề, 3 cho phân loại cảm xúc).

- **Dropout**: Lớp dropout với tỷ lệ 0.1 được thêm để ngăn chặn hiện tượng quá khớp (overfitting).

Mô hình được huấn luyện với hàm mất mát CrossEntropyLoss (chuyên dùng cho bài toán phân loại đa lớp), tối ưu hóa bằng Adam và sử dụng cơ chế dừng sớm (early stopping) để dừng quá trình huấn luyện mô hình khi kết quả không còn được cải thiện.

## 4.4 Kết quả

Bảng 4.1 và 4.2 trình bày kết quả phân loại chủ đề và phân loại cảm xúc của mô hình LSTM trên tập test.

**Bảng 4.1 Kết quả phân loại chủ đề với LSTM**

| Lớp | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 0 | 0.92 | 0.89 | 0.90 | 2290 |
| 1 | 0.57 | 0.73 | 0.64 | 572 |
| 2 | 0.98 | 0.57 | 0.72 | 145 |
| 3 | 0.21 | 0.18 | 0.19 | 159 |
| Accuracy | 0.81 | | | 3166 |
| Macro Avg | 0.67 | 0.59 | 0.61 | 3166 |
| Weighted Avg | 0.82 | 0.81 | 0.81 | 3166 |

**Bảng 4.2 Kết quả phân loại cảm xúc với LSTM**

| Lớp | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 0 | 0.65 | 0.63 | 0.64 | 1409 |
| 1 | 0.00 | 0.00 | 0.00 | 167 |
| 2 | 0.67 | 0.77 | 0.71 | 1590 |
| Accuracy | 0.66 | | | 3166 |
| Macro Avg | 0.44 | 0.46 | 0.45 | 3166 |
| Weighted Avg | 0.63 | 0.66 | 0.64 | 3166 |

Kết quả cho thấy mô hình LSTM đạt hiệu suất khá kém, với độ chính xác tổng thể là 0.81 cho phân loại chủ đề và chỉ 0.66 cho phân loại cảm xúc. Đặc biệt, hiệu suất trên lớp 3 (các vấn đề khác) trong phân loại chủ đề rất thấp (recall 0.18, f1-score 0.19), và lớp 1 (cảm xúc trung tính) trong phân loại cảm xúc không

được dự đoán đúng (recall 0.00, f1-score 0.00). Nguyên nhân chính là do kiến trúc mạng LSTM được sử dụng có cấu trúc tương đối đơn giản, với chỉ một lớp nhúng 128 chiều, một lớp LSTM 64 chiều, và một lớp tuyến tính. Mặc dù kiến trúc này giúp giảm chi phí tính toán, nhưng nó không đủ khả năng nắm bắt các đặc trưng phức tạp của dữ liệu văn bản. Để đạt hiệu suất cao hơn, cần sử dụng các mô hình phức tạp hơn, ví dụ như các mạng nhiều lớp hơn. Tuy nhiên, việc huấn luyện các mô hình phức tạp từ đầu (không sử dụng pretraining) đòi hỏi tài nguyên tính toán lớn và lượng dữ liệu dồi dào, điều không phù hợp với bài tập lớn này do hạn chế về dữ liệu, thời gian và tài nguyên.

# 5. Fine-tune mô hình PhoBERT

Phần này trình bày quá trình tinh chỉnh (fine-tuning) mô hình PhoBERT, một mô hình ngôn ngữ lớn được huấn luyện trước cho tiếng Việt, để thực hiện các bài toán phân loại chủ đề và phân loại cảm xúc trên tập dữ liệu phản hồi của sinh viên. Các bước chuẩn bị dữ liệu, cấu hình mô hình, huấn luyện và đánh giá kết quả được mô tả chi tiết, nhấn mạnh hiệu quả của việc sử dụng mô hình đã được huấn luyện trước trong xử lý ngôn ngữ tự nhiên (NLP).

## 5.1 Giới thiệu PhoBERT

PhoBERT là mô hình dựa trên kiến trúc RoBERTa, được huấn luyện trước trên tập dữ liệu lớn tiếng Việt, bao gồm các văn bản từ báo chí và mạng xã hội. PhoBERT sử dụng kỹ thuật tự học (self-supervised learning) để học biểu diễn ngữ cảnh của từ, giúp nó nắm bắt tốt ngữ pháp và ngữ nghĩa tiếng Việt. Mô hình này đặc biệt hiệu quả cho các bài toán NLP tiếng Việt như phân loại văn bản, nhờ vào khả năng hiểu ngữ cảnh sâu sắc. Trong bài tập này, phiên bản PhoBERT-base-v2 được sử dụng để tinh chỉnh cho hai bài toán phân loại.

## 5.2 Chuẩn bị dữ liệu

Quá trình chuẩn bị dữ liệu bao gồm phân đoạn từ, chuyển đổi sang định dạng phù hợp cho mô hình PhoBERT, và tổ chức dữ liệu sử dụng các công cụ từ thư viện Transformers.

Mặc dù tập dữ liệu đã có các câu được phân đoạn từ bằng thư viện pyvi (như trong cột tokens của tập dữ liệu gốc), các câu gốc cần được phân đoạn lại bằng py_vncorenlp, vì PhoBERT yêu cầu định dạng phân đoạn từ của VnCoreNLP để đảm bảo tính nhất quán với dữ liệu huấn luyện trước của nó. Py_vncorenlp thực hiện phân đoạn từ và câu, ví dụ: câu "Ông Nguyễn Khắc Chúc đang làm việc tại Đại học Quốc gia Hà Nội" được phân đoạn thành "Ông Nguyễn_Khắc_Chúc đang làm_việc tại Đại_học Quốc_gia Hà_Nội .". Việc phân đoạn lại này đảm bảo các token đầu vào phù hợp với từ điển của PhoBERT. Các bước tiền xử lý văn bản khác (như loại bỏ dấu câu, chuẩn hóa chữ thường) thường không cần thiết cho các mô hình ngôn ngữ lớn như PhoBERT, vì chúng đã được huấn luyện để xử lý văn bản thô và có khả năng tự động học các đặc trưng ngữ nghĩa từ dữ liệu đầu vào.

Dữ liệu được tổ chức bằng lớp Dataset và DatasetDict của thư viện Transformers. DatasetDict được sử dụng để lưu trữ ba tập dữ liệu: train (11426 mẫu),

validation (1583 mẫu), và test (3166 mẫu), với các cột topic, sentiment, và segmented_text. Mỗi mẫu được chuyển đổi thành định dạng Dataset, giúp dễ dàng xử lý và tích hợp với các công cụ huấn luyện của Transformers.

Để chuẩn hóa dữ liệu, các văn bản phân đoạn được mã hóa (tokenized) bằng AutoTokenizer của PhoBERT, chuyển thành input_ids và attention_mask. Quá trình mã hóa sử dụng các tham số: padding='max_length', truncation=True, và max_length=128, đảm bảo tất cả chuỗi có độ dài cố định và phù hợp với yêu cầu của mô hình. Các cột không cần thiết (như segmented_text) được loại bỏ, và cột nhãn (topic hoặc sentiment) được đổi tên thành labels để tương thích với Trainer.

## 5.3 Cấu hình mô hình, tham số huấn luyện và Trainer

Mô hình PhoBERT-base-v2 được tải bằng AutoModelForSequenceClassification từ thư viện Transformers, với số nhãn là 4 cho phân loại chủ đề và 3 cho phân loại cảm xúc. Mô hình được chuyển sang thiết bị GPU (cuda) để tăng tốc huấn luyện.

Tham số huấn luyện (TrainingArguments) được cấu hình như sau:

- Đầu ra lưu tại thư mục ./results.
- Tắt ghi log bên ngoài (report_to=[]).
- Đánh giá và lưu mô hình sau mỗi epoch
- Tốc độ học (learning_rate) là 2e-5, kích thước batch là 32 cho cả huấn luyện và đánh giá.
- Huấn luyện trong 15 epoch với weight_decay=0.01 để tránh quá khớp.
- Chọn mô hình tốt nhất dựa trên f1 macro

Trainer được khởi tạo với mô hình, tham số huấn luyện, tập train và validation, tokenizer, và hàm compute_metrics để tính accuracy, precision, recall, và f1 macro.

## 5.4 Kết quả

Bảng 5.1 và 5.2 trình bày kết quả phân loại chủ đề và phân loại cảm xúc của mô hình PhoBERT trên tập test.

**Bảng 5.1 Kết quả phân loại chủ đề với PhoBERT**

| Lớp | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 0 | 0.93 | 0.94 | 0.94 | 2290 |
| 1 | 0.76 | 0.76 | 0.76 | 572 |
| 2 | 0.94 | 0.94 | 0.94 | 145 |
| 3 | 0.61 | 0.54 | 0.57 | 159 |
| Accuracy | | 0.89 | | 3166 |
| Macro Avg | 0.81 | 0.79 | 0.80 | 3166 |
| Weighted Avg | 0.88 | 0.89 | 0.89 | 3166 |

**Bảng 5.2 Kết quả phân loại cảm xúc với PhoBERT**

| Lớp | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 0 | 0.94 | 0.97 | 0.96 | 1409 |
| 1 | 0.73 | 0.48 | 0.58 | 167 |
| 2 | 0.95 | 0.96 | 0.96 | 1590 |
| Accuracy | | 0.94 | | 3166 |
| Macro Avg | 0.88 | 0.80 | 0.83 | 3166 |
| Weighted Avg | 0.94 | 0.94 | 0.94 | 3166 |

Kết quả cho thấy mô hình PhoBERT đạt hiệu suất vượt trội so với các mô hình trước, với độ chính xác tổng thể là 0.89 cho phân loại chủ đề và 0.94 cho phân loại cảm xúc. Đối với phân loại chủ đề, lớp 3 (các vấn đề khác) vẫn có hiệu suất thấp (recall 0.54, f1-score 0.57) do số lượng mẫu hạn chế và tính không thống nhất của lớp này. Tương tự, trong phân loại cảm xúc, lớp 1 (cảm xúc trung tính) có recall thấp (0.48) và f1-score là 0.58, do sự mất cân bằng dữ liệu. Tuy nhiên, hiệu suất tổng thể của PhoBERT cao hơn đáng kể so với LSTM (0.81 cho chủ đề, 0.66 cho cảm xúc), nhờ vào kiến trúc phức tạp và dữ liệu huấn luyện trước phong phú. Việc tinh chỉnh PhoBERT là phù hợp cho bài tập lớn này, vì nó tận dụng được biểu diễn ngữ cảnh mạnh mẽ mà không yêu cầu huấn luyện từ đầu, tiết kiệm tài nguyên tính toán.

# 6. So sánh

Phần này so sánh hiệu quả của bốn mô hình đã được áp dụng cho bài toán phân loại chủ đề và phân loại cảm xúc: PhoBERT, SVC (với TfidfVectorizer), Multinomial Naive Bayes (với CountVectorizer), và LSTM. Hai bảng dưới đây trình bày các chỉ số đánh giá, theo sau là nhận xét ngắn gọn về hiệu suất.

## 6.1 So sánh hiệu quả các mô hình

Bảng 6.1 và 6.2 trình bày các chỉ số Accuracy, F1 (macro), và F1 (micro) của các mô hình trên hai bài toán phân loại chủ đề và phân loại cảm xúc trên tập test.

**Bảng 6.1 So sánh hiệu quả phân loại chủ đề**

| Mô hình | Accuracy | F1 (macro) | F1 (micro) |
|---|---|---|---|
| PhoBERT | 0.89 | 0.80 | 0.89 |
| SVC | 0.84 | 0.65 | 0.84 |
| Naive Bayes | 0.83 | 0.64 | 0.83 |
| LSTM | 0.81 | 0.61 | 0.81 |

**Bảng 6.2 So sánh hiệu quả phân loại cảm xúc**

| Mô hình | Accuracy | F1 (macro) | F1 (micro) |
|---|---|---|---|
| PhoBERT | 0.94 | 0.83 | 0.94 |
| SVC | 0.79 | 0.62 | 0.79 |
| Naive Bayes | 0.78 | 0.61 | 0.78 |
| LSTM | 0.66 | 0.45 | 0.66 |

Kết quả cho thấy PhoBERT vượt trội trên cả hai bài toán, với Accuracy và F1 cao nhất (0.89 và 0.80 cho chủ đề, 0.94 và 0.83 cho cảm xúc), nhờ kiến trúc phức tạp và dữ liệu huấn luyện trước. SVC và Naive Bayes có hiệu suất tương đương nhau, nhưng thấp hơn PhoBERT, với SVC nhỉnh hơn một chút (0.84 so với 0.83 cho chủ đề, 0.79 so với 0.78 cho cảm xúc). LSTM có hiệu suất kém nhất (0.81 và 0.61 cho chủ đề, 0.66 và 0.45 cho cảm xúc), do kiến trúc đơn giản và thiếu huấn luyện trước. Sự khác biệt này nhấn mạnh lợi thế của các mô hình đã được huấn luyện trước như PhoBERT trong các bài toán NLP với tài nguyên hạn chế.

## 6.2 Train lại mô hình với cách chia dữ liệu mới

Dù PhoBERT là mô hình hiệu quả nhất dựa trên kết quả trước, việc huấn luyện lại mô hình này đòi hỏi nhiều thời gian và tài nguyên tính toán do kiến trúc phức tạp. Để đơn giản hóa và tiết kiệm tài nguyên trong bài tập lớn này, hai mô hình SVC (với TfidfVectorizer) và Multinomial Naive Bayes (với CountVectorizer) được chọn để train lại với cách chia dữ liệu mới.

### *6.2.1 Chia lại dữ liệu*

Trong cách chia dữ liệu gốc, tỷ lệ train : valid : test xấp xỉ 7 : 1 : 2. Để kích thước các tập dữ liệu được chia lại tương đương với dữ liệu gốc, dữ liệu được chia lại bằng cách thực hiện train-test split hai lần: lần đầu chia train_valid và test với test size = 0.8, sau đó chia train_valid thành train và valid với test size = 0.125. Sau khi chia lại, các tập dữ liệu mới có kích thước như sau:

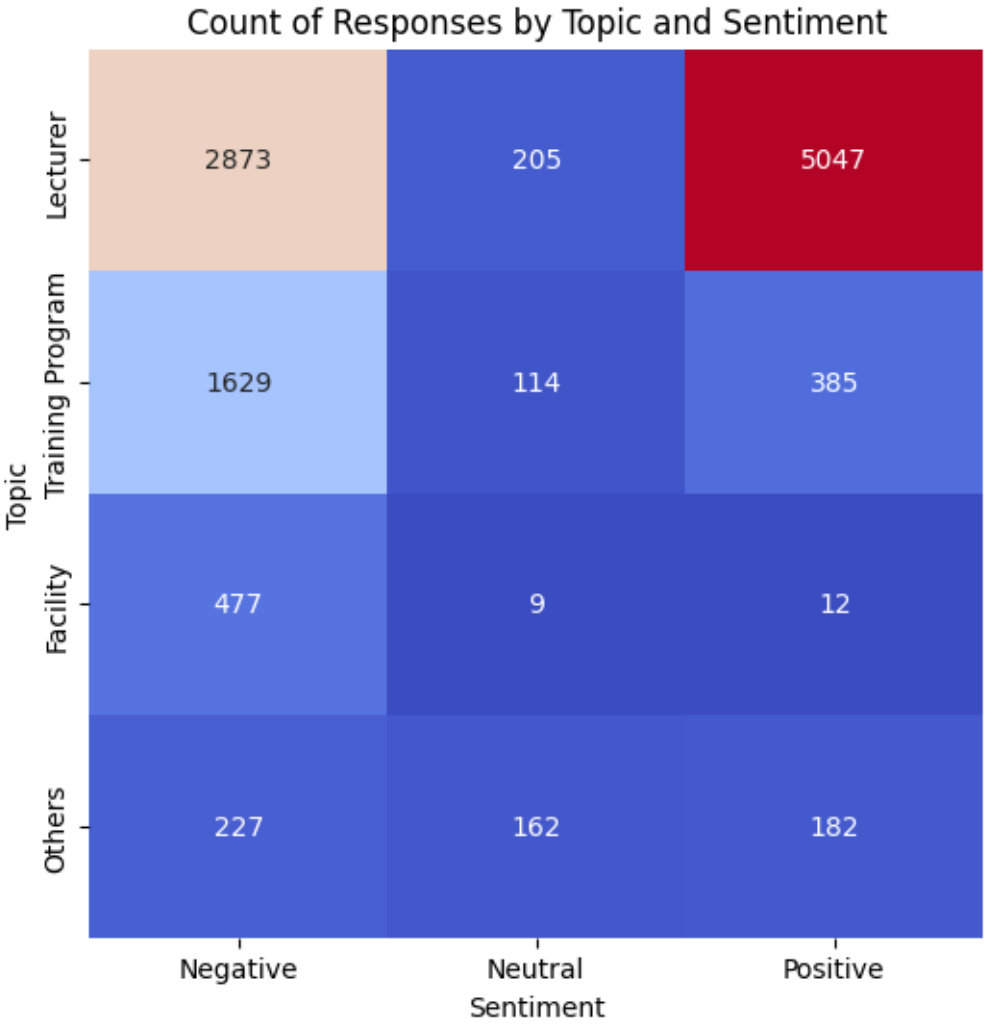
- New Train: 11322
- New Valid: 1618
- New Test: 3235

Phân phối dữ liệu mới được thể hiện qua Hình 6.1, cho thấy sự mất cân bằng vẫn tồn tại, đặc biệt ở các lớp có số lượng mẫu ít như topic 3 (các vấn đề khác) và sentiment 1 (cảm xúc trung tính).

### *6.2.2 Kết quả*

Bảng 6.3 và 6.4 trình bày kết quả phân loại chủ đề, trong khi Bảng 6.5 và 6.6 trình bày kết quả phân loại cảm xúc trên tập test mới.

So sánh với cách chia dữ liệu cũ (train : valid : test xấp xỉ 7 : 1 : 2), kết quả mới với tỷ lệ 7 : 1 : 2 (sau khi chia lại) cho thấy hiệu suất không thay đổi nhiều. Đối với phân loại chủ đề, MultinomialNB giảm nhẹ từ 0.83 xuống 0.83, SVC từ 0.84 xuống 0.85 (hầu như không đổi). Với phân loại cảm xúc, MultinomialNB giảm từ 0.78 xuống 0.80, SVC từ 0.79 xuống 0.80 (cũng không đáng kể). Sự khác biệt nhỏ này cho thấy cách chia dữ liệu mới không ảnh hưởng lớn đến hiệu suất, nhưng vẫn phản ánh vấn đề mất cân bằng dữ liệu ở các lớp thiểu số (như topic 3 và sentiment 1).




**Hình 6.1 Phân phối dữ liệu theo cách chia mới**

**Bảng 6.3 Kết quả phân loại chủ đề với MultinomialNB**

| Lớp | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 0 | 0.86 | 0.94 | 0.90 | 2321 |
| 1 | 0.69 | 0.60 | 0.64 | 608 |
| 2 | 0.88 | 0.88 | 0.88 | 143 |
| 3 | 0.42 | 0.03 | 0.06 | 163 |
| Accuracy | | 0.83 | | 3235 |
| Macro Avg | 0.71 | 0.61 | 0.62 | 3235 |
| Weighted Avg | 0.80 | 0.83 | 0.81 | 3235 |

**Bảng 6.4 Kết quả phân loại chủ đề với SVC**

| Lớp | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 0 | 0.87 | 0.94 | 0.91 | 2321 |
| 1 | 0.72 | 0.65 | 0.69 | 608 |
| 2 | 0.85 | 0.87 | 0.86 | 143 |
| 3 | 0.75 | 0.15 | 0.25 | 163 |
| Accuracy | | 0.85 | | 3235 |
| Macro Avg | 0.80 | 0.65 | 0.67 | 3235 |
| Weighted Avg | 0.84 | 0.85 | 0.83 | 3235 |

**Bảng 6.5 Kết quả phân loại cảm xúc với MultinomialNB**

| Lớp | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 0 | 0.81 | 0.79 | 0.80 | 1488 |
| 1 | 0.43 | 0.02 | 0.04 | 139 |
| 2 | 0.79 | 0.87 | 0.83 | 1608 |
| Accuracy | | 0.80 | | 3235 |
| Macro Avg | 0.68 | 0.56 | 0.56 | 3235 |
| Weighted Avg | 0.79 | 0.80 | 0.78 | 3235 |

**Bảng 6.6 Kết quả phân loại cảm xúc với SVC**

| Lớp | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 0 | 0.78 | 0.84 | 0.81 | 1488 |
| 1 | 0.77 | 0.07 | 0.13 | 139 |
| 2 | 0.82 | 0.83 | 0.83 | 1608 |
| Accuracy | | 0.80 | | 3235 |
| Macro Avg | 0.79 | 0.58 | 0.59 | 3235 |
| Weighted Avg | 0.80 | 0.80 | 0.79 | 3235 |

# 7. Kết luận

## 7.1 Các vấn đề đã giải quyết

Bài tập này đã thành công trong việc áp dụng và so sánh hiệu quả của các mô hình học máy và học sâu để phân loại chủ đề và cảm xúc trong tập dữ liệu phản hồi của sinh viên Việt Nam (UIT-VSFC). Cụ thể, các mô hình như Multinomial Naive Bayes, SVC, LSTM, và PhoBERT đã được huấn luyện và đánh giá trên các tập dữ liệu đã được tiền xử lý, bao gồm phân đoạn từ, loại bỏ dấu câu, và chuẩn hóa văn bản. Kết quả cho thấy PhoBERT đạt hiệu suất cao nhất (0.89 cho chủ đề, 0.94 cho cảm xúc), trong khi các mô hình đơn giản hơn như SVC và Naive Bayes cũng cho kết quả khả quan (0.84 và 0.83 cho chủ đề, 0.79 và 0.78 cho cảm xúc). Ngoài ra, bài tập đã thử nghiệm việc chia lại dữ liệu với tỷ lệ mới ($7:1:2$) và đánh giá lại hiệu suất, cho thấy sự ổn định của các mô hình trong các điều kiện khác nhau. Phân tích phân phối dữ liệu và mất cân bằng cũng được thực hiện để hiểu rõ hơn về tập dữ liệu.

## 7.2 Các vấn đề có thể được mở rộng

Bài tập này có thể được mở rộng theo nhiều hướng để nâng cao hiệu quả và ứng dụng thực tiễn. Trước hết, một yêu cầu quan trọng là xây dựng mô hình để phát hiện các phản hồi có mức độ quan trọng, cần sự quan tâm hoặc xử lý từ phía nhà trường, đặc biệt với phản hồi tiêu cực (ví dụ: khiếu nại về giảng dạy hoặc cơ sở vật chất), và những phản hồi tích cực đáng được truyền thông hoặc nhân rộng (ví dụ: khen ngợi giảng viên hoặc chương trình đào tạo). Tuy nhiên, việc triển khai mô hình này hiện tại chưa rõ cách tiếp cận, có thể đòi hỏi phân tích sâu hơn về mức độ nghiêm trọng hoặc mức độ lan tỏa của phản hồi. Ngoài ra, một hướng mở rộng khác là cải thiện hiệu suất của các mô hình bằng cách áp dụng kỹ thuật tăng cường dữ liệu (data augmentation), chẳng hạn như tạo thêm dữ liệu tổng hợp cho các lớp thiểu số (như topic 3 hoặc sentiment 1) để giảm mất cân bằng và nâng cao độ chính xác tổng thể.

# TÀI LIỆU THAM KHẢO

[1] K. V. Nguyen, V. D. Nguyen, P. X. V. Nguyen, T. T. H. Truong, and N. L.-T. Nguyen, "Uit-vsfc: Vietnamese students' feedback corpus for sentiment analysis," in *2018 10th International Conference on Knowledge and Systems Engineering (KSE)*, 2018, pp. 19–24.