TRƯỜNG ĐẠI HỌC PHENIKAA

KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO BÀI TẬP LỚN

MÔN HỌC: LẬP TRÌNH PHÂN TÍCH DỮ LIỆU VỚI PYTHON

Đề tài: Phân tích và Phân loại cảm xúc các bài đăng tiếng Anh trên Twitter (X)

Thành viên nhóm

Nguyễn Quang Nhật	**22010510**	**K16-KHMT-TN**
Nguyễn Ngọc Hảo	**22010115**	**K16-KHMT-TN**

GVHD: Th.S Nguyễn Anh Tuấn

19/07/2025 – Hà Nội

Nội dung

1.Tóm tắt **3**
2.Giới thiệu **3**
3.Kiến thức nền tảng **4**
3.1. TF-IDF (Term Frequency - Inverse Document Frequency) 4
3.2. Bag of Words (BoW) 4
3.3. Support Vector Machine (SVM) 5
3.3.1. Mô hình SVM cơ bản 5
3.3.2. SVM OvsA (One-vs-All) 6
3.3.3. Ưu điểm và nhược điểm 7
3.3. Mô hình hồi quy Logistic (Logistic Regression) 7
3.3.1. Giới thiệu về hồi quy Logistic 7
3.3.2. Các dạng hồi quy logistic hay sử dụng. 8
3.3.3. Ưu điểm và nhược điểm 10
3.4. Mô hình Random Forest. 10
3.4.1. Giới thiệu về mô hình Random Forest 10
3.4.2. Mô hình kết hợp (ensemble model) 10
3.4.3. Lấy mẫu tái hợp (bootstrapping) 11
3.4.4. Quy trình mô hình Random Forest 11
3.4.5. Ưu điểm và nhược điểm 12
4. Thu thập, khai phá và xử lý dữ liệu **12**
4.1. Giới thiệu về bộ dữ liệu 12
4.2. Chi tiết bộ dữ liệu 13
4.3. Tiền xử lý văn bản 15
a, Loại bỏ dấu câu 16
b, Tách từ (Tokenization) 16
c, Loại bỏ từ dừng (Stop Words Removal) 17
d, Lemmatization (Chuẩn hóa từ về dạng gốc) 17
4.4. Khai phá và phân tích dữ liệu 19
5. Thiết kế và triển khai mô hình **23**
5.1. Mô tả các bước xây dựng thuật toán 23
5.2. Chọn feature dữ liệu và các phép xử lý feature 24
5.3. Chia dữ liệu 24
5.4. Xử lý dữ liệu 24
5.5. Chọn mô hình, tham số mô hình 25
6. Kết quả và thảo luận **27**
6.1. Tiêu chí đánh giá: 27
6.2. Kết quả mô hình: 28
6.3. Nhận xét 31
6.4. Demo ứng dụng. 32
7. Tổng kết bài tập lớn **33**
8. Tài liệu tham khảo **34**

Danh sách hình vẽ

Hình 4.1: Load và view dữ liệu từ đường dẫn.

Hình 4.2: Tổng quan về dữ liệu.

Hình 4.3: Kiểm tra rỗng

Hình 4.4: Các nhãn của bộ dữ liệu

Hình 4.5: Số lượng điểm dữ liệu của mỗi nhãn

Hình 4.6: Biểu đồ phân phối của các token.

Hình 4.7: Biểu đồ tỉ trọng và số lượng điểm dữ liệu của mỗi nhãn của bộ dữ liệu.

Hình 4.8: Biểu đồ tỉ trọng và số lượng điểm dữ liệu của mỗi nhãn sau khi downsampling.

Hình 4.9: Biểu đồ thể hiện phân phối độ dài của câu.

Hình 4.10: Biểu đồ thể hiện phân phối độ dài câu của mỗi nhãn.

Hình 4.11: Biểu đồ so sánh tổng số lượng từ và số lượng từ duy nhất.

Hình 4.12: Biểu đồ phân phối từ thuộc nhãn.

Hình 4.13: Biểu đồ thể hiện số lần xuất hiện của từ trong mỗi nhãn.

Hình 5.1: SVM scores.

Hình 5.2: Logistic Regression scores.

Hình 5.3: Random Forest Classifier scores.

Hình 6.1: Thông số Accuracy cho tất cả mô hình.

Hình 6.2: SVM.

Hình 6.3: Logistic Regression.

Hình 6.4. Random Forest.

Hình 6.5. Giao diện sản phẩm

Hình 6.6. Kết quả thử nghiệm

Danh sách bảng biểu

Bảng 3.1: Minh họa Bag of Words.

Bảng 4.1: Text Pre-processing.

Bảng 4.2: Ví dụ các điểm dữ liệu sẽ thành rỗng sau loại bỏ từ dừng.

Bảng 6.1: Ma trận Confusion Matrix.

Bảng 6.2: Thông số Precision cho tất cả mô hình.

Bảng 6.3: Thông số Precision cho tất cả mô hình.

Bảng 7.1: Bảng phân công công việc.

1.Tóm tắt

Cảm xúc là một trong những bản năng cơ bản của con người. Nhận diện cảm xúc đóng vai trò quan trọng trong lĩnh vực phân tích văn bản. Hiện nay, biểu cảm và trạng thái cảm xúc của con người đã trở thành chủ đề nghiên cứu hàng đầu. Trong dự án này, mục tiêu chính của chúng tôi là phát hiện cảm xúc của con người từ văn bản đầu vào bằng một số kỹ thuật học máy.

2.Giới thiệu

Cảm xúc là một trong những bản năng cơ bản của con người. Cảm xúc đề cập đến các trạng thái ý thức hoặc trạng thái tâm trí khác nhau được thể hiện dưới dạng cảm giác. Chúng có thể được bộc lộ qua biểu cảm khuôn mặt, cử chỉ, văn bản và lời nói.

Nhận diện cảm xúc đóng vai trò quan trọng trong lĩnh vực phân tích văn bản. Hiện nay, biểu cảm và trạng thái cảm xúc của con người đã trở thành chủ đề nghiên cứu hàng đầu. Nhận diện và phân loại cảm xúc từ văn bản là những lĩnh vực nghiên cứu mới, có mối liên hệ chặt chẽ với Phân tích cảm xúc.

Phân tích cảm xúc nhằm mục đích phát hiện và nhận diện cảm xúc thông qua biểu đạt trong câu, chẳng hạn như tức giận, ngạc nhiên, vui vẻ, ghê tởm, buồn bã, v.v. Để nhận diện cảm xúc, chúng tôi sẽ sử dụng một số kỹ thuật xử lý ngôn ngữ tự nhiên (NLP) và các thuật toán phân loại trong học máy.

Dự án này nằm trong khuôn khổ môn học Lập trình cho Trí tuệ Nhân tạo, một lĩnh vực quan trọng trong các ứng dụng liên quan đến phân tích dữ liệu. Mục tiêu của bài tập lớn là xây dựng một mô hình có khả năng nhận diện cảm xúc từ văn bản một cách chính xác, giúp giải quyết các bài toán như phân tích tâm lý người dùng, đánh giá phản hồi khách hàng, và cải thiện trải nghiệm tương tác giữa con người và máy tính.

3.Kiến thức nền tảng

3.1. TF-IDF (Term Frequency - Inverse Document Frequency)

TF-IDF (**Term Frequency - Inverse Document Frequency**) (*Tf–idf*, n.d.) là một kỹ thuật thống kê được sử dụng để đánh giá tầm quan trọng của một từ trong một tài liệu so với một tập hợp tài liệu

TF-IDF của một từ t trong tài liệu d được tính bằng:

$$\text{TF-IDF}(t,d) = \text{TF}(t,d) \times \text{IDF}(t)$$

TF (Tần suất xuất hiện của từ) là một thước đo tần suất t xuất hiện trong một tài liệu d.

$$\text{TF}(t,d) = \frac{f_{t,d}}{số\ lượng\ từ\ xuất\ hiện\ trong\ tài\ liệu\ d}$$

với $f_{t,d}$ là số lần t xuất hiện trong tài liệu d. Do đó, mỗi tài liệu và từ sẽ có giá trị TF riêng.

IDF đo lường mức độ phổ biến của từ trong toàn bộ tập tài liệu, mục tiêu là để đo độ hiểm của t trong toàn bộ bộ dữ liệu.

$$\text{IDF}(t) = log(\frac{tổng\ số\ lượng\ bộ\ tài\ liệu}{số\ lượng\ tài\ liệu\ có\ chứa\ t + 1})$$

Từ phổ biến trong một tài liệu nhưng hiếm trong tập dữ liệu sẽ có trọng số TF-IDF cao. Từ xuất hiện ở nhiều tài liệu sẽ có IDF thấp, giảm ảnh hưởng của các từ phổ biến

3.2. Bag of Words (BoW)

Bag of Words (BoW) là một phương pháp phổ biến trong xử lý ngôn ngữ tự nhiên (NLP) để biểu diễn văn bản dưới dạng vector số. Phương pháp này hoạt động bằng cách trích xuất tất cả các từ duy nhất trong tập dữ liệu huấn luyện để tạo thành một từ điển (vocabulary). Sau đó, mỗi văn bản được biểu diễn bằng một vector có kích thước bằng số lượng từ trong từ điển, trong đó giá trị tại mỗi vị trí thể hiện số lần từ đó xuất hiện trong văn bản. (Guide, 2024)

Trong một bộ dữ liệu lớn, có thể có những từ xuất hiện rất nhiều lần nhưng không mang ý nghĩa quan trọng. Để khắc phục vấn đề này, TF-IDF (Term Frequency - Inverse Document Frequency) được sử dụng kết hợp với BoW nhằm đánh giá mức độ quan trọng của từng từ trong một tài liệu so với toàn bộ tập dữ liệu. (Guide, 2024)

Mặc dù BoW là một phương pháp đơn giản và dễ triển khai, nó có một số hạn chế, đặc biệt là việc không nắm bắt được ngữ nghĩa hay thứ tự từ trong văn bản. Điều này có thể dẫn đến việc mất thông tin ngữ cảnh quan trọng. Hơn nữa, khi từ điển trở nên quá lớn, ma trận biểu diễn văn bản có thể rất thưa (sparse) và tốn bộ nhớ. Tuy nhiên, BoW vẫn là một kỹ thuật nền tảng trong nhiều ứng dụng NLP như phân loại văn bản, hệ thống tìm kiếm thông tin, và hệ thống gợi ý nội dung.

Câu	great	is	love	programming	python	i	for
"I love programming"	0	0	1	1	0	1	0
"Python is great for programming"	1	1	0	1	1	0	1
"I love love love Python"	0	0	3	0	1	1	0

Bảng 3.1: Minh họa Bag of Words.

3.3. Support Vector Machine (SVM)

Support Vector Machine là một thuật toán học máy có giám sát, thường được sử dụng cho bài toán phân loại và hồi quy. SVM hoạt động bằng cách tìm một siêu phẳng (hyperplane) tối ưu để phân tách các lớp dữ liệu với khoảng cách (margin) lớn nhất.

3.3.1. Mô hình SVM cơ bản

Trong trường hợp bài toán phân chia n điểm thành 2 lớp, bài toán có dạng như sau (*Machine Learning Cơ Bản*, 2017):

Cho tập dữ liệu huấn luyện D = $\{(x_i, y_i)\}_{i=1}^{n}$ với, $x_i \in R^d$ là vector đặc trưng của mẫu thứ i và $y_i \in \{-1,1\}$ là nhãn của mẫu đó.

Siêu phẳng phân tách có dạng:

$$w^T x + b = 0$$

trong đó, w là vector trọng số và b là bias.

Điều kiện phân tách đúng của các điểm dữ liệu:

$$y_i (w^T x_i + b) \geq 1, \forall i = 1, 2, \ldots, n.$$

Khoảng cách từ một điểm dữ liệu đến siêu phẳng là:

$$\frac{|w^T x + b|}{||w||}$$

SVM tìm siêu phẳng có khoảng cách lớn nhất giữa hai lớp, tức là tối ưu hóa:

$$max_{w,b} \frac{2}{||w||}$$

hay tương đương với bài toán tối ưu:

$$min_{w,b} \frac{1}{2} ||w||^2$$

với ràng buộc:

$$y_i (w^T x_i + b) \geq 1, \forall i$$

3.3.2. SVM OvsA (One-vs-All)

Với bài toán có nhiều hơn 2 lớp, ta cần giải bài toán tối ưu sau:

$$min_{w_k b_k} \frac{1}{2} ||w_k||^2 + C \sum_{n=1}^{N} max(0, 1 - \overline{y}_n (w_k^T x_n + b_k))$$

trong đó:

- w_k là vector trọng số của bộ phân loại cho lớp k.
- b_k là hệ số điều chỉnh (bias).
- C là siêu tham số kiểm soát sự đánh đổi giữa biên rộng và lỗi phân loại.
- $\overline{y}_n$ là nhãn đã được biến đổi thành nhị phân:

$$\tilde{y}_n = \begin{cases} +1, & \text{nếu } y_n = k \\ -1, & \text{nếu } y_n \neq k \end{cases}$$

Với hàm mất mát hinge loss được bổ sung, giúp kiểm soát lỗi phân loại.

$$\sum_{n=1}^{N} max(0,\ 1 - \overline{y}_n(w_k^T x_n + b_k))$$

Với mỗi mẫu n:

- Nếu mẫu được phân loại đúng và nằm ngoài biên quyết định $\overline{y}_n(w_k^T x_n + b_k) \geq 1$ thì mất mát bằng 0.
- Nếu mẫu nằm trong biên quyết định hoặc bị phân loại sai định $\overline{y}_n(w_k^T x_n + b_k) \geq 1$ thì mất mát tăng theo khoảng cách của mẫu đến biên.

C là hệ số kiểm soát sự đánh đổi giữa:

- Biên rộng (khả năng tổng quát tốt) khi C nhỏ.
- Độ chính xác huấn luyện cao hơn nhưng có thể overfit khi C lớn.

3.3.3. Ưu điểm và nhược điểm

Ưu điểm: VM giúp tránh hiện tượng overfitting, đặc biệt khi dữ liệu có nhiễu, vì nó chỉ quan tâm đến support vectors – những điểm dữ liệu quan trọng nhất. Ngoài ra, SVM có thể giải quyết tốt bài toán phi tuyến bằng cách sử dụng kernel trick, giúp biến đổi dữ liệu vào không gian cao hơn để phân tách tốt hơn. Điều này làm cho SVM trở thành một công cụ mạnh mẽ trong các bài toán phân loại và hồi quy.

Nhược điểm: Độ phức tạp tính toán cao, đặc biệt khi áp dụng trên tập dữ liệu lớn với hàng triệu mẫu, hiệu suất của mô hình phụ thuộc rất nhiều vào lựa chọn hàm kernel. Ngoài ra, với những tập dữ liệu có lớp chồng chéo nhau, SVM có thể cho kết quả không ổn định và không đạt hiệu suất cao

3.3. Mô hình hồi quy Logistic (Logistic Regression)

3.3.1. Giới thiệu về hồi quy Logistic

Hồi quy logistic là một mô hình học máy có giám sát được áp dụng chủ yếu trong các bài toán phân loại. Trong trường hợp số lượng class đầu ra của mô hình phân loại là 2 class thì đó chính là bài toán phân loại nhị phân – binary

classification (*[AI From Scratch][Basic ML] #3 - Logistic Regression*, n.d.). Hồi quy logistic sẽ dự đoán xác suất một sự kiện xảy ra, từ đó phân loại các mẫu dữ liệu vào các lớp tương ứng. Ví dụ trong bài toán phân tích cảm xúc trong văn bản, hồi quy logistic sẽ được dùng để dự đoán một văn bản là mang cảm xúc "tích cực" hay "tiêu cực".

3.3.2. Các dạng hồi quy logistic hay sử dụng.

Mô hình hồi quy logistic trong bài toán phân loại nhị phân:

Phân loại nhị phân là bài toán phân loại có biến mục tiêu gồm hai nhãn 0 và 1. Trong đó nhãn 1 thường là nhãn mang tính tích cực (Positive) và nhãn 0 là tiêu cực (Negative). Nhãn tích cực chính là sự kiện cần phát hiện và tiêu cực là trường hợp không xảy ra sự kiện. Mục tiêu của bài toán phân loại nhị phân là dự báo xác suất một mẫu thuộc về một trong hai lớp tương ứng với nhãn 0 và 1. Tổng của xác suất này bằng 1 (Nguyễn, n.d., #):

$$P(x, w) + P(y = 0|x, w) = 1$$

Trong công thức trên thì $P(x, w)$ chính là xác suất có điều kiện, thể hiện cho xác suất của nhãn tích cực tại một quan sát x ứng với một mô hình cụ thể. Giá trị của xác suất này luôn nằm trong khoảng [0,1]. Chính vì thế mà hàm Sigmoid hay còn được gọi là hồi quy logistic được sử dụng để chuyển đổi một giá trị tuyến tính thành xác suất trong khoảng [0,1]:

$$\sigma(z) = \frac{1}{1+e^{-z}}$$

Trong đó, $z = \beta_0 + \beta_1.x_1 + \beta_2.x_2 + ... + \beta_n.x_n$ là tổ hợp tuyến tính của các đặc trưng x_i và các hệ số β_i. β_0 : là hệ số chặn, còn $\beta_1, \beta_2, ..., \beta_n$ là các trọng số của các đặc trưng.

Ranh giới quyết định: Hàm dự đoán sẽ đưa ra các giá trị xác suất trong khoảng 0 đến 1. Để có thể ánh xạ các xác suất này tới các danh mục nhị phân rời rạc như đúng hay sai thì cần chọn một ngưỡng nếu mà xác suất dự đoán lớn hơn giá trị ngưỡng này thì ta sẽ phân loại thành danh mục đó, còn nếu thấp hơn ta sẽ phân loại vào danh mục còn lại ví dụ với ngưỡng là 0.5 thì $\sigma(z) > 0.5$ mẫu sẽ được phân loại vào lớp 1 và ngược lại nếu $\sigma(z) \leq 0.5$ thì mẫu sẽ thuộc lớp 0.

Để huấn luyện mô hình, hàm mất mát (Loss Function) Cross-Entropy được áp dụng vào bài toán phân loại nhị phân:

$$L(\beta) = -\frac{1}{n}\sum_{i=1}^{n}[y_i.\log log\,(p_i) + \left(1 - p_i\right).\log log\left(1 - p_i\right)]$$

Trong đó:

- n: Số lượng mẫu dữ liệu trong tập huấn luyện.
- y_i: Giá trị thực tế của đầu ra thứ i.
- p_i: Xác suất dự đoán thuộc lớp 1 cho đầu vào thứ i.

Khi bài toán phân loại có nhiều hơn hai lớp thì bài toán hồi quy logistic phân loại nhị phân không còn phù hợp nữa. Thay vào đó, mô hình hồi quy logistic phân loại đa lớp sẽ được áp dụng vào hay còn được gọi là softmax regression.

Thay vì sử dụng hàm Sigmoid để dự đoán xác suất cho mỗi lớp thì bài toán phân loại đa lớp này sử dụng hàm softmax để tính xác suất cho mỗi lớp trong số K lớp (*Machine Learning Cơ Bản*, 2017):

$$P(x) = \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}}$$

Trong đó:

- K: Số lượng lớp.
- $z_k = \beta_{k0} + \beta_{k1}.x_1 + \beta_{k2}.x_2 + ... + \beta_{kn}.x_n$: Hàm tuyến tính cho lớp k.

Hàm sẽ đưa ra dự đoán dựa vào lớp có xác suất cao nhất:

$$\hat{y} = \arg arg \max_k \; P(y = k|x)$$

Hàm mất mát cross-entropy đa lớp được áp dụng:

$$L(\beta) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} y_{i,k}.\log log\,(p_{i,k})$$

Trong đó, $y_{i,k}$ là nếu mẫu I thuộc lớp k, 0 nếu không và $p_{i,k}$ là xác suất dự đoán mẫu i thuộc lớp k.

3.3.3. Ưu điểm và nhược điểm

Ưu điểm: Mô hình hồi quy logistic có cấu trúc rõ ràng với tốc độ huấn luyện nhanh và phù hợp trên tập dữ liệu lớn. Đồng thời tính linh hoạt của mô hình được thể hiện rõ rệt nhờ việc có thể chuyển đổi mô hình phân loại nhị phân sang phân loại đa lớp thông qua softmax regression.

Nhược điểm: Trong trường hợp số lượng mẫu không cân bằng dẫn đến mô hình có thể bị lệch và dự đoán không tốt nếu không được điều chỉnh. Đồng thời mô hình hồi quy logistic chỉ dùng cho bài toán phân loại nhị phân hoặc đa lớp, không thể dự đoán đối với các giá trị liên tục.

3.4. Mô hình Random Forest.

3.4.1. Giới thiệu về mô hình Random Forest

Random Forest là một thuật toán học có giám sát được sử dụng cho cả bài toán phân lớp và bài toán hồi quy. Random Forest kết hợp nhiều cây quyết định để tạo ra một mô hình dự đoán chính xác và ổn định hơn so với việc chỉ sử dụng một cây quyết định đơn lẻ. Mô hình Random Forest áp dụng hai phương pháp học kết hợp (ensemble learning) và lấy mẫu tái lập (bootstrapping) để cải thiện độ chính xác và khả năng chống overfitting.

3.4.2. Mô hình kết hợp (ensemble model)

Giả sử khi xây dựng một hệ thống phân loại nhị phân để phân biệt giữa hai đối tượng. Khi một mẫu dữ liệu gặp khó khăn chẳng hạn như do nhiễu, chất lượng dữ liệu thấp, một mô hình đơn lẻ có thể chỉ cho kết quả dự báo với xác suất không quá cao khiến cho độ tin cậy vào dự đoán đó không quá chắc chắn. Để khắc phục điều đó, có thể áp dụng xây dựng nhiều mô hình khác nhau (chẳng hạn xây dựng 9 mô hình) và sau đó sử dụng một phương pháp gọi là bầu cử đa số để tổng hợp kết quả và đưa ra dự đoán. Mặc dù mỗi mô hình đơn lẻ chỉ đưa ra xác suất không quá cao, nếu phần lớn các mô hình đều ủng hộ một nhãn nhất định (giả sử 7 trên 9 mô hình dự báo ra cùng một nhãn), thì qua bầu cử đa số có thể suy ra nhãn dự báo cuối cùng đó là đúng. Khi độ tin cậy từ dự đoán của một mô hình đơn lẻ không đạt yêu cầu thì việc kết hợp kết quả từ nhiều mô hình sẽ giúp tăng cường độ chắc chắn và sự ổn định đối với kết quả dự đoán cuối cùng.

3.4.3. Lấy mẫu tái hợp (bootstrapping)

Giả định rằng mô hình có tập dữ liệu huấn luyện là một tập hợp $D = \{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}$ bao gồm N quan sát. Random Forest sẽ sử dụng phương pháp lấy mẫu tái lập để tạo thành B tập dữ liệu con. Quá trình lấy mẫu tái lập này còn gọi là bỏ túi (bagging). Cụ thể, quá trình này lặp lại M lần, và mỗi lần ta sẽ thực hiện chọn mẫu ngẫu nhiên (các phần tử được phép lặp lại) từ tập D ban đầu để tạo thành một tập con $B_i = \{(x_1^{(i)}, y_1^{(i)}), (x_2^{(i)}, y_2^{(i)}), ..., (x_M^{(i)}, y_M^{(i)})\}$. Bởi vì quá trình chọn mẫu được phép lặp lại nên một số quan sát trong tập D sẽ không xuất hiện trong tập B_i. Các quan sát đó được gọi là nằm ngoài túi (out of bag). Những quan sát nằm ngoài túi này thường được dùng để đánh giá chất lượng mô hình Random Forest mà không cần tách riêng một tập kiểm tra khác. Với mỗi tập dữ liệu con B_i được tạo ra, mô hình cây quyết định được huấn luyện và đạt được kết quả dự đoán $\hat{y}_j^{(i)} = f_i(x_j)$. Trong đó, $\hat{y}_j^{(i)}$ là dự đoán cho quan sát thứ j từ mô hình thứ i, x_j là vector đặc trưng của quan sát đó còn hàm f_i là hàm dự báo của mô hình thứ i. Sau khi có kết quả dự báo từ tất cả mô hình, ta sẽ tổng hợp lại để đưa ra dự đoán cuối cùng $\hat{y}_j$. Cách thức tổng hợp và đưa ra kết quả dự đoán phụ thuộc vào loại bài toán (*Mô Hình Kết Hợp (Ensemble Model)*, n.d.) với B là tổng số cây quyết định đã huấn luyện:

- Đối với bài toán dự báo (hồi quy) kết quả dự đoán sẽ được tính từ giá trị trung bình các dự đoán:

$$\hat{y}_j = \left(\sum_{i=1}^{B} \hat{y}_j^{(i)}\right)$$

- Đối với bài toán phân loại (nhị phân hoặc đa lớp): Mô hình sẽ trả về xác suất hoặc nhãn cho từng lớp, và kết quả cuối cùng thu được là lớp nhận được nhiều phiếu nhất trong quá trình bầu cử tức là chọn lớp c có tổng xác suất (số phiếu) cao nhất từ tất cả mô hình:

$$\hat{y}_j = \arg arg \max_c \left(\sum_{i=1}^{B} p_i(\hat{y}_j^{(i)} = c)\right)$$

3.4.4. Quy trình mô hình Random Forest

Mô hình Random Forest được tạo thành theo trình tự như sau:

Bước đầu tiên mô hình sẽ khởi tạo số lượng cây ($n\_estimators$) cần xây dựng. Đối với mỗi cây mô hình sẽ thực hiện chọn mẫu ngẫu nhiên có lặp lại từ tập dữ liệu huấn luyện để tạo thành các tập dữ liệu con khác nhau.

Tiếp theo mô hình sẽ chọn ngẫu nhiên một nhóm biến và xây dựng mô hình cây quyết định trên tập dữ liệu con ở bước trên. Mỗi lần chọn sẽ chỉ sử dụng một số biến nhất định (chọn ngẫu nhiên) để huấn luyện cây quyết định thay vì dùng toàn bộ biến. Đồng thời quá trình này được lặp đi lặp lại nhiều lần để có thể tạo ra nhiều mô hình cây quyết định khác nhau.

Cuối cùng mô hình sẽ tổng hợp tức là thực hiện bầu cử hoặc lấy trung bình giữa các cây quyết định để đưa ra dự đoán cuối cùng.

3.4.5. Ưu điểm và nhược điểm

Ưu điểm: Mô hình Random Forest là sự kết hợp của nhiều cây quyết định nên kết quả được dự đoán được tổng hợp từ nhiều mô hình nên thường chính xác hơn. Giảm hiện tượng quá khớp (overfitting) do nhiều cây được huấn luyện từ các tập dữ liệu con giúp hạn chế việc mô hình học quá sát với dữ liệu huấn luyện.

Nhược điểm: Thời gian huấn luyện và dự đoán có thể hơi tốn thời gian và sẽ tỉ lệ thuận với số cây được sử dụng trong quá trình huấn luyện. Không phù hợp với các dữ liệu có cấu trúc đơn giản do mô hình random forest được thiết kế để xử lý các mối quan hệ phi tuyến và phức tạp.

4. Thu thập, khai phá và xử lý dữ liệu

4.1. Giới thiệu về bộ dữ liệu

Bộ dữ liệu được sử dụng trong nghiên cứu này là dair-ai/emotion, một tập dữ liệu phổ biến được lưu trữ và cung cấp thông qua nền tảng Hugging Face (Dair-Ai/emotion · Datasets at Hugging Face, n.d.). Đây là tập dữ liệu gồm các bài viết tiếng Anh được thu thập từ mạng xã hội Twitter, với mục tiêu phục vụ cho các tác vụ phân loại cảm xúc trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP). Tổng cộng, bộ dữ liệu bao gồm 436.809 văn bản ngắn, tất cả đều đã được chuyển về dạng chữ thường (lowercase) để chuẩn hóa dữ liệu đầu vào.

Mỗi văn bản trong bộ dữ liệu được gán một nhãn duy nhất, đại diện cho một trong sáu cảm xúc cơ bản: *anger* (tức giận), *fear* (sợ hãi), *joy* (vui vẻ), *love* (yêu thương), *sadness* (buồn bã) và *surprise* (ngạc nhiên). Các cảm xúc này được lựa chọn dựa trên mô hình phân loại cảm xúc cơ bản trong tâm lý học, nhằm đảm bảo khả năng khái quát hóa tốt trong các bài toán học máy và hiểu cảm xúc của người dùng.

Về phân bố, bộ dữ liệu không hoàn toàn cân bằng giữa các nhãn. Cụ thể, hai cảm xúc *joy* và *sadness* chiếm tỷ lệ lớn hơn đáng kể so với các nhãn còn lại, điều này phản ánh xu hướng biểu đạt cảm xúc thường gặp của người dùng trên mạng xã hội. Sự mất cân đối này là một yếu tố cần được cân nhắc trong quá trình huấn luyện mô hình, bởi nó có thể ảnh hưởng đến hiệu suất phân loại đối với các cảm xúc ít phổ biến hơn như *surprise* hay *love*.

Nhờ tính đa dạng, số lượng lớn và cấu trúc rõ ràng, bộ dữ liệu dair-ai/emotion là một lựa chọn phù hợp cho các nghiên cứu liên quan đến nhận diện cảm xúc, phân tích quan điểm người dùng, hoặc xây dựng các hệ thống AI có khả năng hiểu và phản hồi cảm xúc trong ngôn ngữ tự nhiên.

4.2. Chi tiết bộ dữ liệu

Đầu tiên, chúng ta sẽ thực hiện load và view file data:

	text	label
0	i feel awful about it too because it s my job ...	0
1	im alone i feel awful	0
2	ive probably mentioned this before but i reall...	1
3	i was feeling a little low few days back	0
4	i beleive that i am much more sensitive to oth...	2
...	...	...
416804	that was what i felt when i was finally accept...	1
416805	i take every day as it comes i m just focussin...	4
416806	i just suddenly feel that everything was fake	0
416807	im feeling more eager than ever to claw back w...	1
416808	i give you plenty of attention even when i fee...	0

416809 rows × 2 columns

Hình 4.1: Load và view dữ liệu của hugging face.

Có thể thấy, tập dữ liệu bao gồm 416.809 dòng, với hai cột chính:

- ‘text’ : Chứa các câu văn thể hiện cảm xúc của người viết.
- ‘label’ : Là nhãn cảm xúc dưới dạng số nguyên, tương ứng với một loại cảm xúc nhất định.

Dữ liệu này thuộc dạng xử lý ngôn ngữ tự nhiên (NLP), có thể được sử dụng để huấn luyện mô hình phân loại cảm xúc từ văn bản. Tuy nhiên, trước khi đưa vào huấn luyện, cần kiểm tra xem dữ liệu có tồn tại các giá trị trống, có bị mất cân bằng nhãn.

Từ file dữ liệu chúng ta phải kiểm tra các dạng của biến và xem có dữ liệu nào còn thiếu hay không để có thể xử lý chúng:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 416809 entries, 0 to 416808
Data columns (total 2 columns):
 #   Column  Non-Null Count   Dtype 
---  ------  --------------   ----- 
 0   text    416809 non-null  object
 1   label   416809 non-null  int64 
dtypes: int64(1), object(1)
memory usage: 6.4+ MB
```

Hình 4.2: Tổng quan về dữ liệu.

```
data.isna().sum()
```

	0
text	0
label	0

dtype: int64

```
[6] data.isnull().sum()
```

	0
text	0
label	0

dtype: int64

Hình 4.3: Kiểm tra rỗng

Có thể thấy, bộ dữ liệu rất hoàn thiện khi không có giá trị rỗng nào.

Tiếp đó, ta thực hiện kiểm tra số lượng nhãn cảm và giá trị của các nhãn trong bộ dữ liệu.

```
data['label'].unique()
```

```
array([0, 1, 2, 3, 4, 5])
```

Hình 4.4: Các nhãn của bộ dữ liệu

Để dễ dàng cho việc quan sát và đánh giá dữ liệu hơn, ta sẽ thực hiện chuyển đổi label của bộ dữ liệu từ dạng số sang string tương liệu về biểu cảm (được cung cấp trong phần giới thiệu về Huggingface của bộ dữ liệu).Các biểu cảm tương ứng với giá trị số là: sadness (0), joy (1), love (2), anger (3), fear (4), surprise (5).

Với mỗi nhãn có số lượng điểm dữ liệu như sau:

```
data['label'].value_counts()
```

label	count
joy	141067
sadness	121187
anger	57317
fear	47712
love	34554
surprise	14972

Hình 4.5: Số lượng điểm dữ liệu của mỗi nhãn

4.3. Tiền xử lý văn bản

Trước khi trích xuất đặc trưng, chúng tôi cần tiền xử lý dữ liệu văn bản để đảm bảo mô hình đạt được kết quả tốt. Trước khi huấn luyện mô hình, chúng tôi thực hiện các bước tiền xử lý như sau (*Essential Text Pre-Processing Techniques for NLP!*, 2024):

a, Loại bỏ dấu câu

b, Tách từ (Tokenization)

c, Loại bỏ từ dừng (Stop Words)

d, Lemmatization (Chuẩn hóa từ về dạng gốc)

Trong bộ dữ liệu này, các văn bản trong ‘text’ đều đã ở dạng chữ in thường (lowercase) nên chúng ta có thể bỏ qua bước chuyển các văn bản về dạng đồng nhất lowercase hoặc uppercase

a, Loại bỏ dấu câu

Đầu tiên, chúng tôi loại bỏ dấu câu khỏi dữ liệu văn bản. Ví dụ:
Danh sách dấu câu cần loại bỏ: !"#$%&´()*+,-./:;<=>?[ ]_‘{|}

Những ký tự này không giúp ích trong việc nhận diện cảm xúc, vì vậy chúng cần được loại bỏ.

Ví dụ với Doc-1:

- Trước khi loại bỏ dấu câu: *i didn't feel humiliated*
- Sau khi loại bỏ dấu câu: *i didnt feel humiliated*

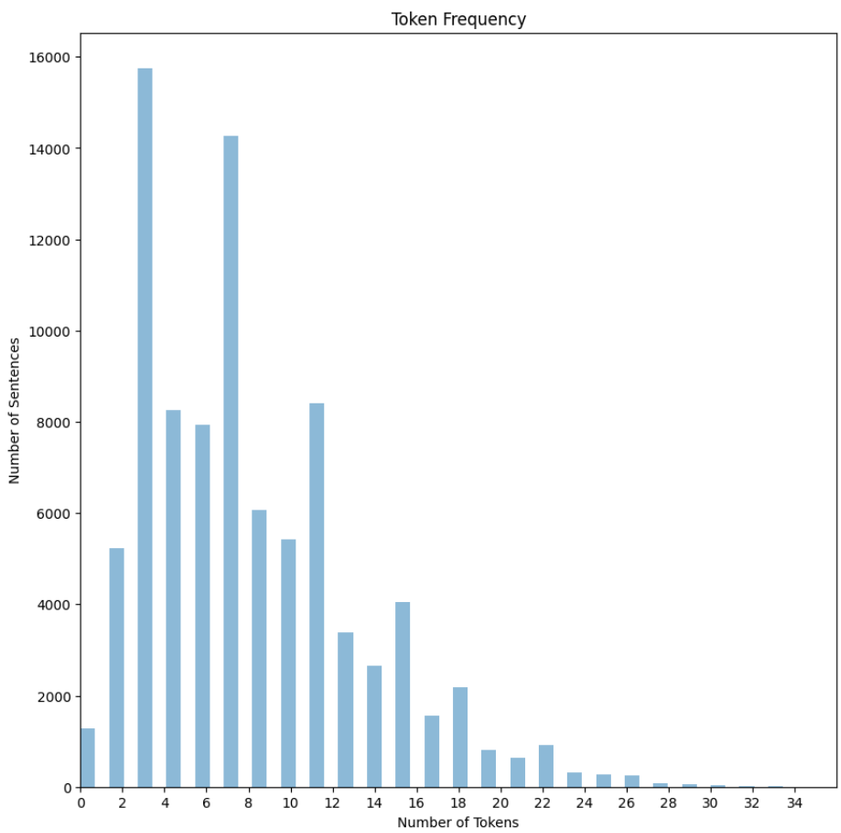
b, Tách từ (Tokenization)

Tokenization là quá trình chia nhỏ một câu thành danh sách các từ.

Ví dụ với Doc-1:

- Trước khi tách từ: *i didnt feel humiliated*
- Sau khi tách từ: [i, didnt, feel, humiliated]

Sau khi hoàn thành quá trình tiền xử lý, chúng tôi phân tích số lượng tổng thể của các token trong toàn bộ tập dữ liệu. Phân phối của các token sau khi tiền xử lý sẽ được hiển thị trong biểu đồ tiếp theo.



Hình 4.6: Biểu đồ phân phối của các token.

c, Loại bỏ từ dừng (Stop Words Removal)

Từ dừng là tập hợp các từ thường xuyên xuất hiện trong ngôn ngữ nhưng không mang nhiều thông tin hữu ích. Ví dụ trong tiếng Anh bao gồm: *"a"*, *"the"*, *"is"*, *"are"*, v.v.

Trong xử lý ngôn ngữ tự nhiên (NLP) và khai thác văn bản (Text Mining), việc loại bỏ từ dừng giúp giảm nhiễu và tập trung vào các từ mang ý nghĩa quan trọng hơn. Ngoài danh sách từ dừng chuẩn, chúng tôi cũng có thể mở rộng danh sách này theo nhu cầu của mô hình.

Ví dụ với Doc-1:

- Trước khi loại bỏ từ dừng: [i, didnt, feel, humiliated]
- Sau khi loại bỏ từ dừng: [humiliated]

d, Lemmatization (Chuẩn hóa từ về dạng gốc)

Lemmatization, còn được gọi là kỹ thuật chuẩn hóa từ, là một phương pháp quan trọng trong xử lý ngôn ngữ tự nhiên. Nó phân tích hình thái từ và chuyển đổi bất kỳ dạng nào của từ về dạng gốc (lemma). Ví dụ:

- *"Playing" → "Play"*
- *"Going" → "Go"*

Trong Doc-1:

- Trước khi lemmatization: [humiliated]
- Sau khi lemmatization: [humiliated]

Mặc dù trong ví dụ này, từ *"humiliated"* không thay đổi, nhưng với các từ khác, lemmatization có thể giúp chuẩn hóa dữ liệu hiệu quả hơn.

Sau khi thực hiện tất cả các bước tiền xử lý, tập dữ liệu của chúng tôi có dạng như sau:

Content	Punctuation Removed	Tokenized Text	Stop word Removed	Lemmatized Text
I didn't feel humiliated	i didnt feel humiliated	['i', 'didnt', 'feel', 'humiliated']	['humiliated']	['humiliated']
I can go from feeling so hopeless to so damned hopeful just from being around someone who cares and is awake	i can go from feeling so hopeless to so damned hopeful just from being around someone who cares and is awake	['i', 'can', 'go', 'from', 'feeling', 'so', 'hopeless', 'to', 'so', 'damned', 'hopeful', 'just', 'from', 'being', 'around', 'someone', 'who', 'cares', 'and', 'is', 'awake']	['go', 'feeling', 'hopeless', 'damned', 'hopeful', 'around', 'someone', 'cares', 'awake']	['go', 'feeling', 'hopeless', 'damned', 'hopeful', 'around', 'someone', 'care', 'awake']
I'm grabbing a minute to post I feel greedy wrong	im grabbing a minute to post i feel greedy wrong	['im', 'grabbing', 'a', 'minute', 'to', 'post', 'i', 'feel', 'greedy', 'wrong']	['grabbing', 'minute', 'post', 'greedy', 'wrong']	['grabbing', 'minute', 'post', 'greedy', 'wrong']
I am ever feeling nostalgic about the fireplace I will know that it is still on the property	i am ever feeling nostalgic about the fireplace i will know that it is still on the property	['i', 'am', 'ever', 'feeling', 'nostalgic', 'about', 'the', 'fireplace', 'i', 'will', 'know', 'that', 'it', 'is', 'still', 'on', 'the', 'property']	['ever', 'feeling', 'nostalgic', 'fireplace', 'know', 'still', 'property']	['ever', 'feeling', 'nostalgic', 'fireplace', 'know', 'still', 'property']

Bảng 4.1: Text Pre-processing.

4.4. Khai phá và phân tích dữ liệu

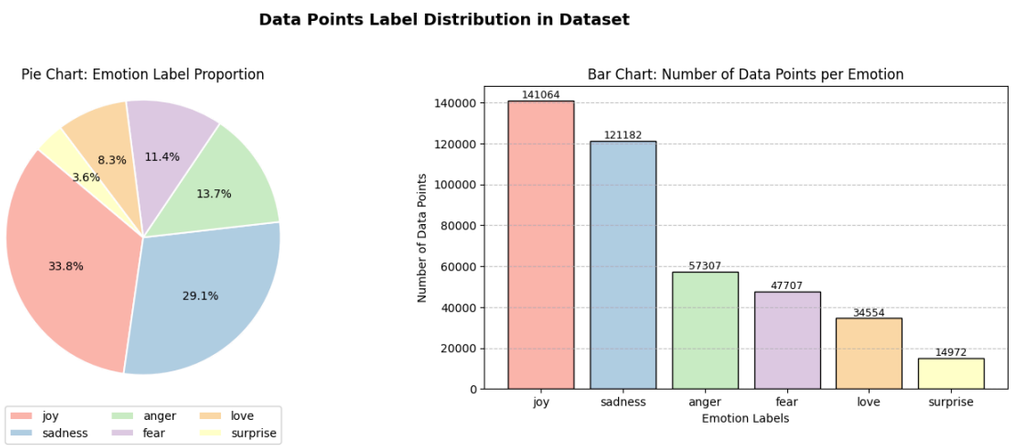
Trong giai đoạn tiền xử lý dữ liệu, việc loại bỏ các từ dừng (stop words) là một bước phổ biến. Tuy nhiên, điều này có thể dẫn đến trường hợp một số câu quá ngắn, và nếu tất cả các từ trong câu đều thuộc danh sách từ dừng, kết quả là trong giữ liệu tiền xử lý sẽ có các câu rỗng, có độ dài bằng 0, ta sẽ thực hiện việc loại bỏ các câu rỗng này khỏi dữ liệu do câu không còn ý nghĩa.

Qua kiểm tra, thấy rằng có 23 điểm dữ liệu sau khi thực hiện lại bỏ từ dùng sẽ trở thành câu rỗng.

Index	Text_tokenized	Label
60210	[after, my]	anger
75246	[one, day]	sadness
77384	[when, i]	anger
127737	[did, very]	joy
128410	[in]	fear
154969	[when, a, man]	anger
170335	[once]	anger
172993	[when, i, was]	joy
180033	[when, a, boy]	anger
208981	[in]	sadness

Bảng 4.2: Ví dụ các điểm dữ liệu sẽ thành rỗng sau loại bỏ từ dừng.

Để trực quan hóa, ta thực hiện biểu diễn số lượng điểm dữ liệu của mỗi nhãn lên biểu đồ tròn và biểu đồ cột.



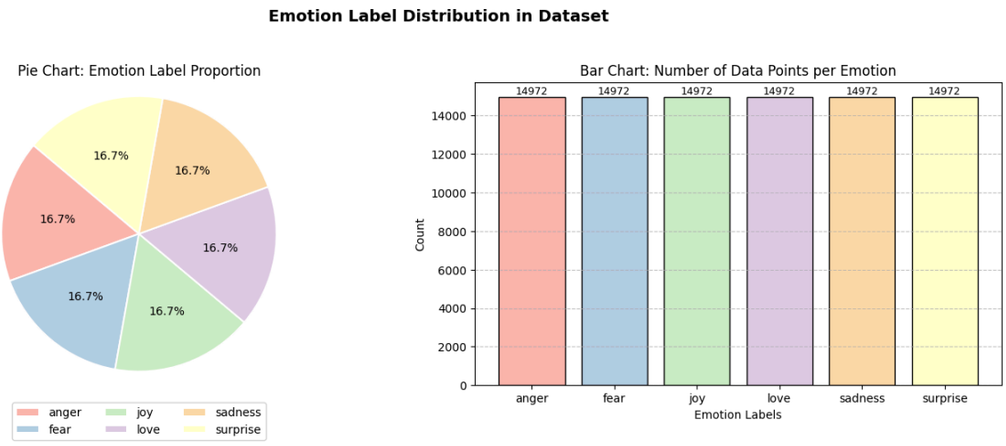
Hình 4.7: Biểu đồ tỉ trọng và số lượng điểm dữ liệu của mỗi nhãn của bộ dữ liệu.

Biểu đồ trên thể hiện số lượng của các nhãn cảm xúc trong tập dữ liệu. Dễ dàng nhận thấy rằng hai cảm xúc "Joy" (vui vẻ) và "Sadness" (buồn bã) có số lượng lớn nhất, trong đó "Joy" chiếm tỷ lệ cao nhất. Điều này cho thấy các cảm xúc này xuất hiện phổ biến hơn trong dữ liệu so với các cảm xúc khác.

Ngược lại, cảm xúc "Surprise" (ngạc nhiên) có số lượng thấp nhất, cho thấy nó ít được thể hiện hơn. Các nhãn "Love" (tình yêu), "Anger" (tức giận) và "Fear" (sợ hãi) có số lượng trung bình, với mức độ chênh lệch không quá lớn.

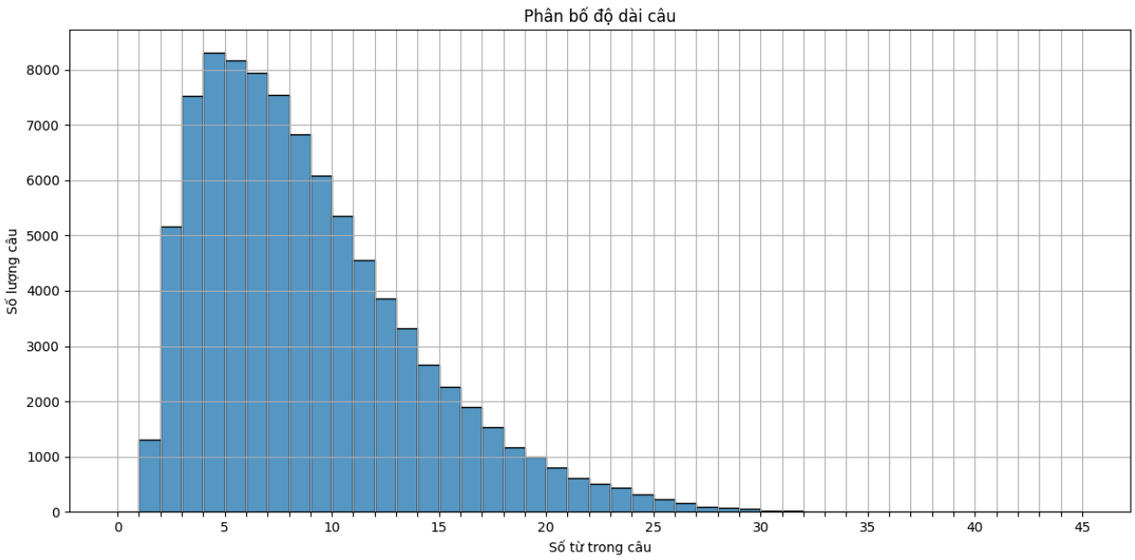
Sự phân bố không đồng đều này có thể dẫn đến mất cân bằng dữ liệu, gây ảnh hưởng đến độ chính xác khi huấn luyện mô hình phân loại cảm xúc. Để khắc phục vấn đề này, chúng tôi sẽ thực hiện cân bằng dữ liệu bằng cách bằng phương pháp Downsampling, thực lấy số lượng mẫu bằng với lớp ít nhất. Ta thực hiện việc này với niềm tin rằng số lượng cách thức để biểu thị cảm xúc bằng văn bản giữa các loại cảm xúc là bằng hoặc gần tương đương nhau.

Ở đây, mẫu ít nhất có 14972, vậy tập dữ liệu mới sẽ có 89832 điểm dữ liệu. Với input và output được lựa chọn cho quá trình huấn luyện mô hình lần lượt sẽ là ‘text’ và ‘label’.



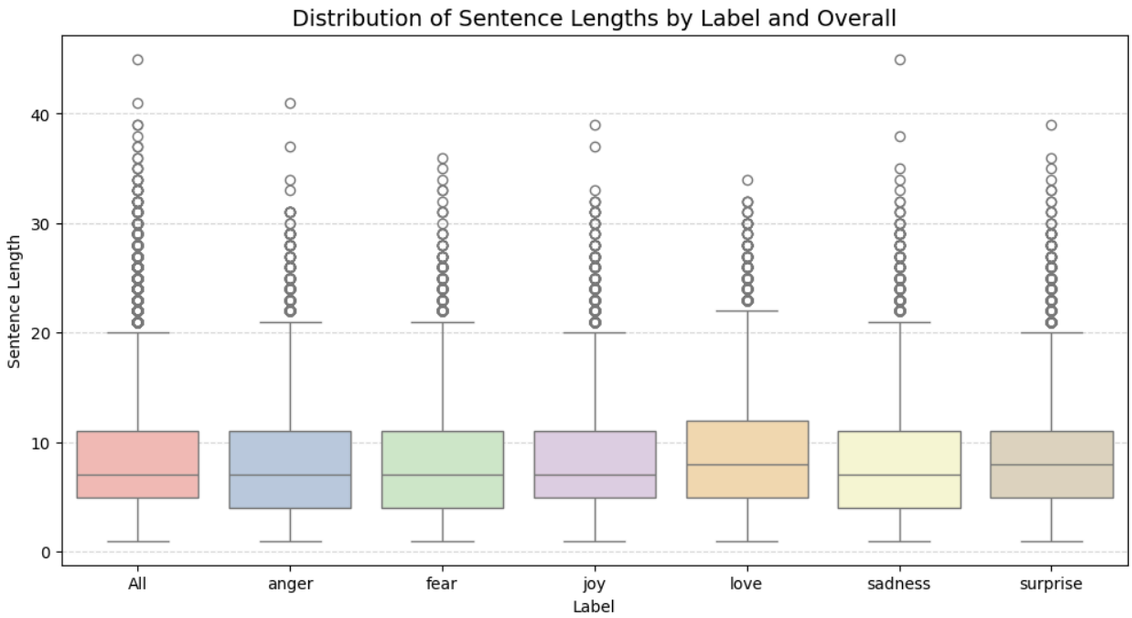
Hình 4.8: Biểu đồ tỉ trọng và số lượng điểm dữ liệu của mỗi nhãn sau khi downsampling

Thực hiện một số thống kê, ta có dữ liệu như sau về độ dài câu:
Trên toàn bộ tập dữ liệu sau khi thực hiện downsample, câu có độ dài ngắn nhất là 1 và câu có độ dài dài nhất là 45 với độ dài trung bình là 8.403 từ.



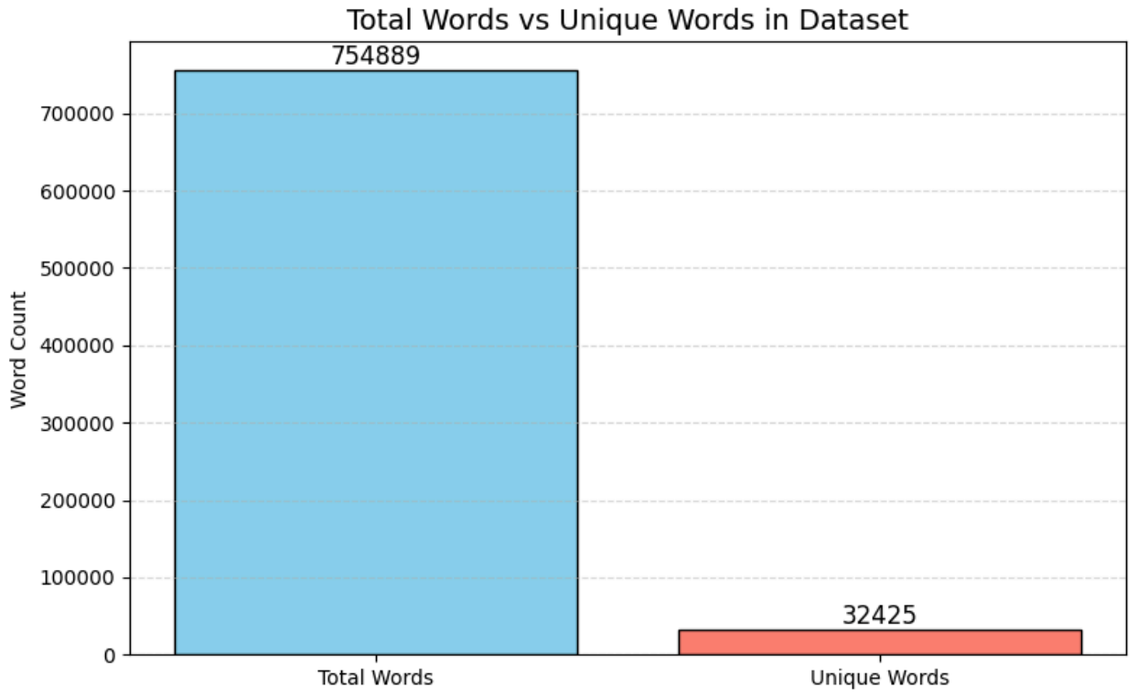
Hình 4.9: Biểu đồ thể hiện phân phối độ dài của câu.

Thực hiện thêm trên các label, ta có kết quả như hình 4.9, không quá sự chênh lệch nhiều trong dữ liệu. Như vậy, độ dài câu sẽ không có hoặc rất ít ảnh hưởng đến ý nghĩa cảm xúc trong câu.



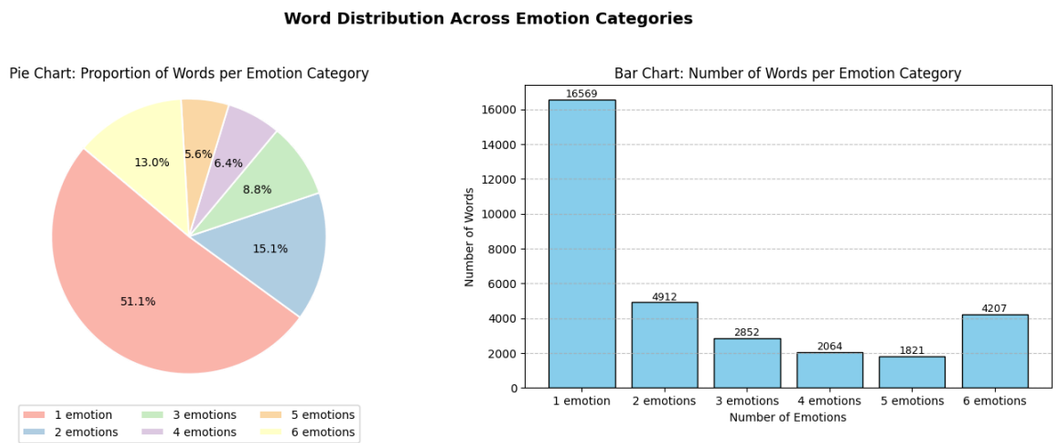
Hình 4.10: Biểu đồ thể hiện phân phối độ dài câu của mỗi nhãn

Kiểm tra tổng tất cả các từ và số từ duy nhất ta thấy có sự cách biệt rất lớn, chứng tỏ có rất nhiều từ được lập lại. Như thông tin của hình 4.6, ta có thể thấy, sẽ có những từ được lặp lại nhiều lần, những từ này có thể ảnh hưởng ít hoặc nhiều đến ý nghĩa cảm xúc của câu hơn tùy thuộc vào tần suất xuất hiện của từ đó qua các category cảm xúc khác nhau. Nếu tần suất xuất hiện của một từ đều tương tự nhau qua các category khác nhau thì sức ảnh hưởng của từ đó đến ý nghĩa cảm xúc của câu là không nhiều. Ngược lại, nếu tần xuất suất hiện của từ đó chỉ tập chung vào một số category cảm xúc, phân phối không đồng đều, thì từ đó sẽ đóng góp nhiều ý nghĩa cảm xúc cho câu.



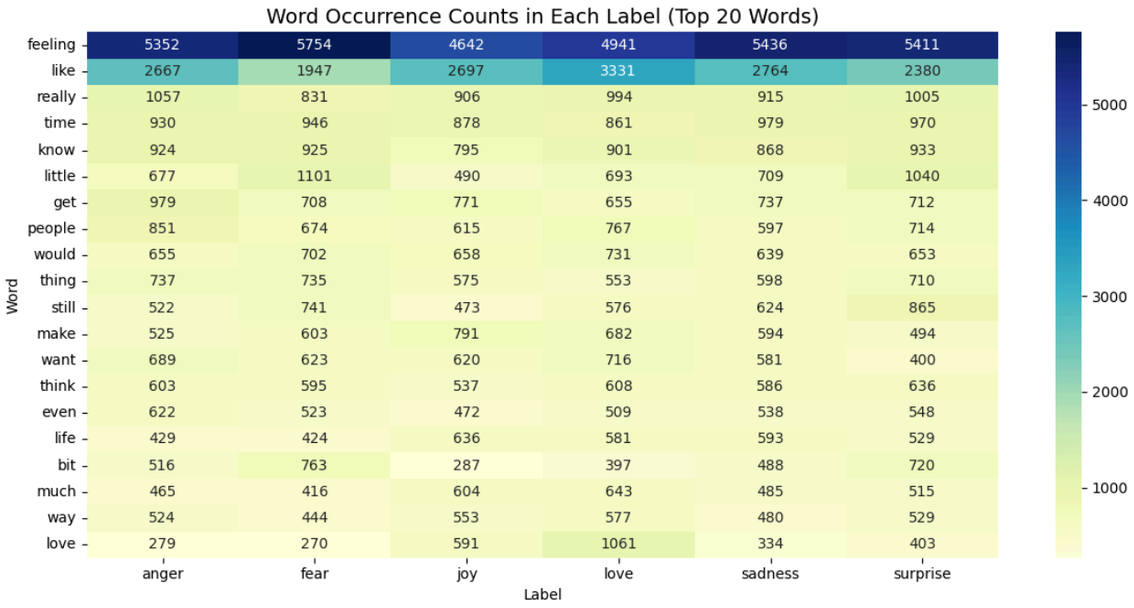
Hình 4.11: Biểu đồ so sánh tổng số lượng từ và số lượng từ duy nhất

Bởi số lượng và tần suất xuất hiện trên cùng một tập dữ liệu tỉ lệ thuận với nhau, vậy nên ở đây chúng em áp dụng các phương pháp thống kê trên số lượng thay vì cho tần suất. Thống kê số lần xuất hiện của một từ trên mỗi emotion category cho thấy, hơn 50% số từ là độc nhất với 1 nhãn, có nghĩa rằng từ đó chỉ xuất hiện trong một nhãn duy nhất, các từ như vậy mang ý nghĩa mạnh mẽ thể hiện cho cảm xúc của câu. Với số lượng các từ nằm trên nhiều hơn 1 nhãn giảm dần, ngoại trừ mục 6 emotion, hay có nghĩa là các từ nằm trong mục này là các từ phổ biến và ít ảnh hưởng đến ý nghĩa cảm xúc của câu. Việc loại bỏ các từ thuộc nhóm này sẽ có thể góp phần tăng độ chính xác của mô hình.



Hình 4.12: Biểu đồ phân phối từ thuộc nhãn

Ta có thể thấy từ hình 4.13, phần lớn các từ nằm trong top 20 từ xuất hiện nhiều nhất trong các nhãn đều là các từ phổ biến và hầu hết đều không mang nhiều ý nghĩa cảm xúc.



Hình 4.13: Biểu đồ thể hiện số lần xuất hiện của từ trong mỗi nhãn

5. Thiết kế và triển khai mô hình

5.1. Mô tả các bước xây dựng thuật toán

Thuật toán trong bài toán phát hiện cảm xúc từ văn bản được triển khai theo các bước chính sau:

1. **Tiền xử lý dữ liệu**: Văn bản đầu vào được làm sạch bằng cách loại bỏ dấu câu, ký tự đặc biệt, chữ số và chuyển về chữ thường.
2. **Trích xuất đặc trưng**: Sử dụng các phương pháp như Bag of Words (BoW), TF-IDF để biến đổi văn bản thành dạng vector số học có thể sử dụng làm đầu vào cho mô hình.
3. **Chia tập dữ liệu**: Dữ liệu được chia thành tập huấn luyện và tập kiểm tra theo tỷ lệ thích hợp.
4. **Huấn luyện mô hình**: Áp dụng các mô hình học máy như SVM, Random Forest, Logistic Regression, v.v.
5. **Đánh giá hiệu năng**: Sử dụng các độ đo như accuracy, precision, recall, F1-score để đánh giá độ chính xác của mô hình.

5.2. Chọn feature dữ liệu và các phép xử lý feature

Dữ liệu đầu vào là các văn bản chứa cảm xúc, được biểu diễn dưới dạng văn bản thuần. Quá trình xử lý feature bao gồm:

Làm sạch dữ liệu: Loại bỏ dấu câu, ký tự đặc biệt, chữ số, và chuyển đổi văn bản về chữ thường.

Vector hóa văn bản: Sử dụng hai phương pháp chính:

- Bag of Words (BoW): Chuyển văn bản thành ma trận số lượng từ.
- TF-IDF (Term Frequency-Inverse Document Frequency): Đánh trọng số cho từ dựa trên tần suất xuất hiện trong tập dữ liệu.

Chuyển đổi dữ liệu: Sử dụng TfidfTransformer để biến đổi ma trận CountVectorizer thành dạng chuẩn hóa.

5.3. Chia dữ liệu

Dữ liệu được chia thành hai phần chính: Tập huấn luyện (training set): Dùng để huấn luyện mô hình. Tập kiểm tr**a** (test set): Dùng để đánh giá hiệu suất của mô hình trên dữ liệu chưa thấy trước đó. sử dụng hàm train_test_split() của sklearn với tỉ lệ 70/30 với 70% tập dữ liệu cho tập huấn luyện và 30% tập dữ liệu cho quá trình testing.

5.4. Xử lý dữ liệu

Đầu tiên, sử dụng CountVectorizer để chuyển đổi văn bản của các tập train và tập test thành ma trận số, trong đó mỗi cột tương ứng với một từ và giá trị là số lần từ đó xuất hiện trong câu, hay nói cách khác là đưa các văn bản về dạng Bag of Words (BoW).

Apply một hàm tiền xử lý văn bản (loại bỏ dấu câu, Tokenization, Stop Words remove, Lemmatization) vào CountVectorizer để thực hiện việc tiền xử lý trên các văn bản trước khi xây dựng thành BoW.

Sử dụng transform() tập huấn luyện để chuyển đổi tập kiểm tra thành dạng vector số. Ở đây không gọi fit_transform() trên X_test để tránh "nhìn thấy" dữ liệu kiểm tra trong quá trình huấn luyện.

TfidfTransformer() được sử dụng để tính toán giá trị TF-IDF từ ma trận BoW (countVector1). fit_transform(countVector1) giúp chuyển đổi dữ liệu huấn luyện từ số lần xuất hiện từ thành trọng số TF-IDF nhằm giảm bớt ảnh hưởng của các từ xuất hiện quá thường xuyên nhưng không mang nhiều ý nghĩa.

5.5. Chọn mô hình, tham số mô hình

Trong bài toán phát hiện cảm xúc từ văn bản, các mô hình học máy được lựa chọn bao gồm Support Vector Machine (SVM), Random Forest Classifier, và Logistic Regression. Đây là những mô hình phổ biến trong bài toán phân loại văn bản nhờ vào khả năng tổng quát hóa tốt và hiệu suất cao trên nhiều tập dữ liệu khác nhau.

a, Support Vector Machine (SVM)

Mục tiêu của SVM là tìm một siêu phẳng tối đa hóa sự phân tách giữa các điểm dữ liệu theo các lớp thực của chúng trong không gian n chiều. Ở dạng cơ bản nhất, SVM không hỗ trợ phân loại đa lớp. Để thực hiện phân loại đa lớp, nguyên tắc tương tự được áp dụng bằng cách chia bài toán phân loại đa lớp thành các bài toán con nhỏ hơn, tất cả đều là bài toán phân loại nhị phân, chẳng hạn như:

- Phương pháp Một chọi Một (One vs One - OVO)
- Phương pháp Một chọi Tất cả (One vs All - OVA)
- Phương pháp Đồ thị Có Hướng Không Chu trình (Directed Acyclic Graph - DAG)

Accuracy: 91.358
Precision: 91.488
Recall: 91.336
F1-score: 91.303

	precision	recall	f1-score	support
anger	0.93	0.90	0.92	4466
fear	0.89	0.87	0.88	4417
joy	0.94	0.86	0.90	4460
love	0.90	0.97	0.93	4548
sadness	0.95	0.89	0.92	4574
surprise	0.88	0.98	0.93	4485
accuracy			0.91	26950
macro avg	0.91	0.91	0.91	26950
weighted avg	0.91	0.91	0.91	26950

Hình 5.1: SVM scores.

b, Logistic Regression

Phân loại đa lớp được thực hiện bằng cách huấn luyện nhiều bộ phân loại hồi quy logistic, mỗi bộ cho một trong số K lớp của tập dữ liệu huấn luyện. Sau khi huấn luyện tất cả các bộ phân loại, chúng ta có thể sử dụng chúng để dự đoán lớp mà dữ liệu kiểm tra thuộc về.

Đối với dữ liệu đầu vào kiểm tra, chúng ta tính toán "xác suất" rằng nó thuộc về mỗi lớp bằng cách sử dụng các bộ phân loại hồi quy logistic đã được huấn luyện, sau đó chọn lớp có xác suất cao nhất làm lớp dự đoán cho điểm dữ liệu đó.

```
Accuracy: 91.165
Precision: 91.191
Recall: 91.143
F1-score: 91.124
              precision    recall  f1-score   support

       anger       0.93      0.91      0.92      4466
        fear       0.90      0.86      0.88      4417
         joy       0.91      0.88      0.89      4460
        love       0.90      0.95      0.93      4548
     sadness       0.94      0.90      0.92      4574
    surprise       0.89      0.96      0.93      4485

    accuracy                           0.91     26950
   macro avg       0.91      0.91      0.91     26950
weighted avg       0.91      0.91      0.91     26950
```

Hình 5.2: Logistic Regression scores.

c, Random Forest Classifier

Rừng ngẫu nhiên có thể được sử dụng cho cả phân loại và hồi quy. Rừng ngẫu nhiên tạo ra nhiều cây quyết định trên các mẫu dữ liệu được chọn ngẫu nhiên, sau đó lấy dự đoán từ từng cây và chọn giải pháp tốt nhất bằng phương pháp bỏ phiếu đa số. Tham số siêu n_estimators là số lượng cây mà thuật toán xây dựng trước khi thực hiện bỏ phiếu tối đa hoặc lấy trung bình của các dự đoán.

```
Accuracy: 89.050
Precision: 89.077
Recall: 89.033
F1-score: 89.009
              precision    recall  f1-score   support

       anger       0.89      0.91      0.90      4466
        fear       0.85      0.85      0.85      4417
         joy       0.90      0.85      0.87      4460
        love       0.89      0.95      0.92      4548
     sadness       0.93      0.87      0.90      4574
    surprise       0.88      0.92      0.90      4485

    accuracy                           0.89     26950
   macro avg       0.89      0.89      0.89     26950
weighted avg       0.89      0.89      0.89     26950
```

Hình 5.3:Random Forest Classifier scores.

Các mô hình này được sử dụng với tham số mặc định mà không qua bước tối ưu hóa tham số do sau khi thực hiện chạy mô hình và kiểm tra trên tập test mô hình đã cho ra kết quả tốt.

6. Kết quả và thảo luận

6.1. Tiêu chí đánh giá:

Bài tập lớn bọn em đã sử dụng bộ dữ liệu gồm 436,809 đoạn văn bản mẫu và 6 nhãn cảm xúc tương ứng bao gồm: buồn, vui, yêu thích, tức giận, sợ hãi và ngạc nhiên. Ban đầu trong dữ liệu gốc các nhãn cảm xúc chỉ là các con số 0,1,2,3,4,5 nhưng trong quá trình xử lý thì bọn em đã ánh xạ các con số đó thành các nhãn cảm xúc tương ứng. Các mô hình như: máy vectơ hỗ trợ (Support Vector Machine), hồi quy logistic (Logistic Regression) và rừng ngẫu nhiên (Random Forest) đã được sử dụng để phát hiện được cảm xúc trong văn bản. Kết quả phân loại của hệ thống so với dữ liệu thực tế sẽ được trình bày dưới dạng ma trận Confusion Matrix với các tham số được mô tả chi tiết ở bảng 3.1:

Kết quả	**Nhãn**		
		POSITIVE	NEGATIVE
	TRUE	TP (True Positive)	TN (True Negative)
	FALSE	FP (False Positive)	FN (False Negative)

Bảng 6.1: Ma trận Confusion Matrix.

Các tham số trong bảng được hiểu như sau:

- TP (True Positives): là tổng số văn bản được dự đoán đúng với nhãn cảm xúc P (cảm xúc cần nhận diện).
- FP (False Positives): là tổng số văn bản mà có nhãn cảm xúc là N (Không phải cảm xúc cần nhận diện) nhưng lại dự đoán nhầm là nhãn cảm xúc P.
- TN (True Negatives): là tổng số văn bản mà có cảm xúc là N (Không phải cảm xúc cần nhận diện) và được dự đoán đúng là N.
- FN (False Negatives): là tổng số văn bản mà thực sự có nhãn cảm xúc là P nhưng lại dự đoán nhầm thành N.

Các mô hình đã được đánh giá dựa trên các performance metrics sau độ chuẩn xác(precision), độ phủ (recall), độ chính xác (accuracy) và f1-score đã được tính toán nhờ các công thức sau:

Trong đó:

- Precision: Đo lường độ tin cậy của nhãn dự đoán P, tức trong số các văn bản dự đoán là P, tỷ lệ bao nhiêu thực sự là P.
- Recall: Đo lường khả năng nhận diện chính xác các văn bản có nhãn là P, trên toàn bộ các văn bản được dự đoán là P.
- Accuracy: Độ chính xác, đo lường tỷ lệ các nhãn được nhận diện đúng (cả P và N) trên tổng số văn bản. Từ những thông số trên, nhận thấy các tham số precision, recall, accuracy càng cao thì hệ thống nhận diện càng tốt.
- F1-score: Đo lường độ cân bằng giữa Precision và Recall, giúp đánh giá hiệu quả tổng thể của mô hình trong việc nhận diện nhãn P.

6.2. Kết quả mô hình:

- Precision:

Kết quả của các thông số đánh giá Precision của các mô hình được thống kê trong các bảng dưới đây:

	Support Vector Machine	Logistic Regression	Random Forest
anger	93.451	92.966	89.407
fear	89.208	89.865	84.544
joy	93.593	90.628	89.682
love	89.648	90.499	89.488
sadness	94.561	93.766	92.916
surprise	88.466	89.424	88.424

Bảng 6.2: Thông số Precision cho tất cả mô hình.

- Recall:

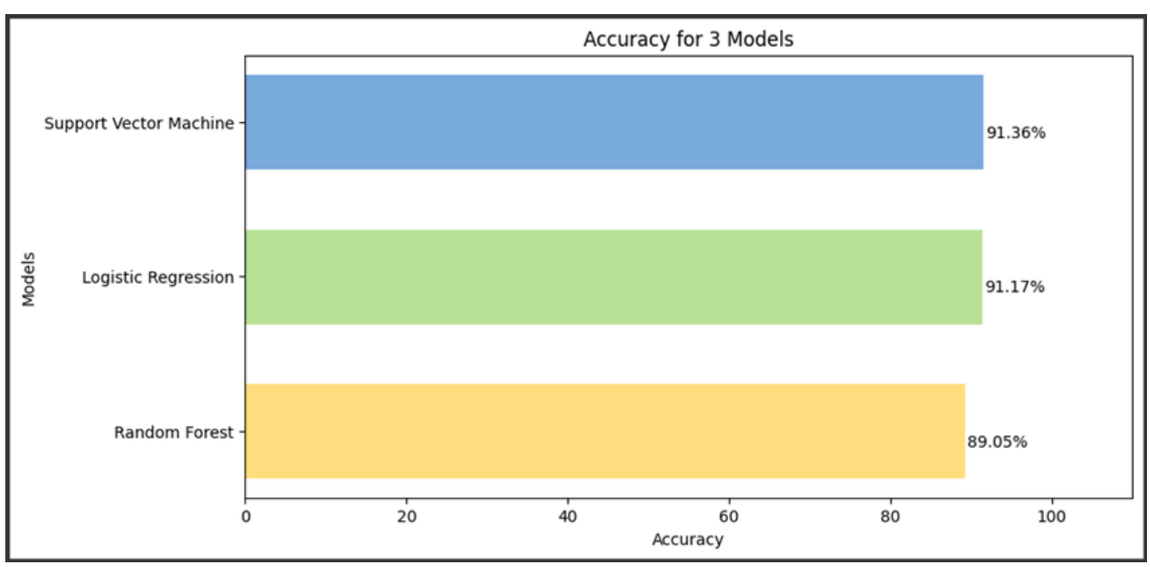
Kết quả của các thông số đánh giá Recall của các mô hình được thống kê trong các bảng dưới đây:

	Support Vector Machine	Logistic Regression	Random Forest
anger	90.416	91.446	91.088
fear	87.209	86.122	84.831
joy	86.143	88.027	85.359
love	96.922	94.877	94.525
sadness	89.331	90.424	86.598
surprise	97.993	95.964	91.795

Bảng 6.3: Thông số Precision cho tất cả mô hình.

- Accuracy:

Kết quả thông số đánh giá Accuracy của các mô hình:



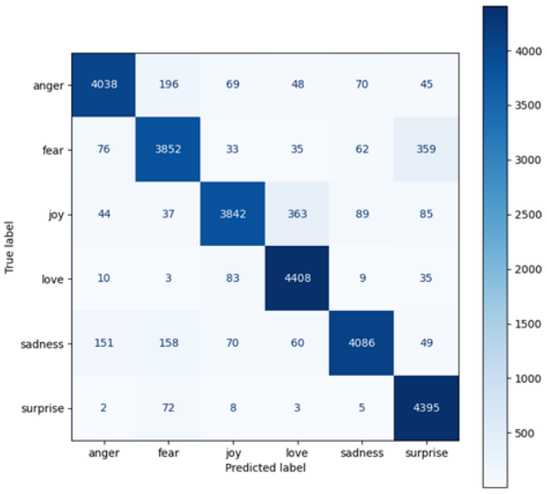
Hình 6.1: Thông số Accuracy cho tất cả mô hình.

- F1-score:

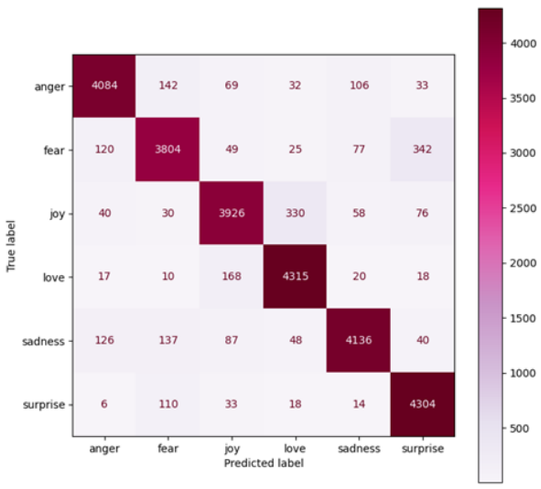
Kết quả F1-score của 3 mô hình: Support Vector Machine (SVM), Logistic Regression, Random Forest Classifier đạt được lần lượt là 91.303%, 91.124%, 89.009%.

- Confusion Matrix:

Confusion Matrix của mô hình Support Vector Machine và Logistic Regression:

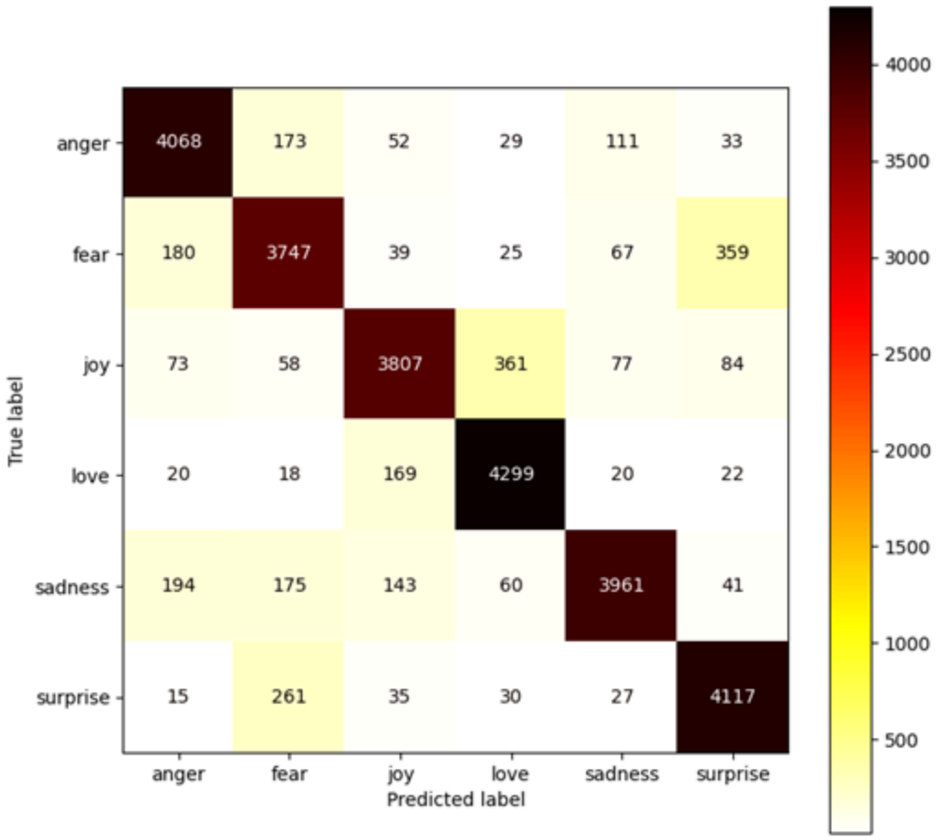


Hình 6.2: SVM.



Hình 6.3: Logistic Regression.

Confusion Matrix của mô hình Random Forest Classifier:



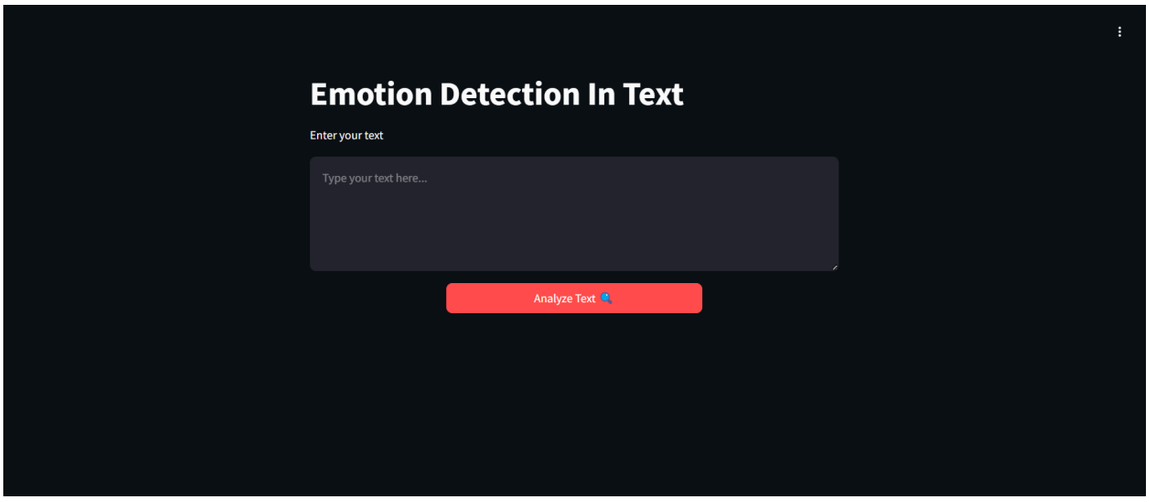
Hình 6.4. Random Forest.

6.3. Nhận xét

- Ưu điểm:
    - Hiệu quả: Mô hình SVM và Logistic Regression đạt độ chính xác xấp xỉ 92% phù hợp trên tập dữ liệu văn bản có độ phức tạp trung bình.
    - Xử lý đa dạng cảm xúc: Mô hình phân loại được nhiều nhãn cảm xúc (6 nhãn) với độ chính xác khá cao.
    - Có demo sử dụng: Dễ tiếp cận và dễ sử dụng.
- Hạn chế:
    - Thiếu ngữ cảnh sâu: TF-IDF không nắm bắt được ngữ cảnh dài hạn hoặc yếu tố trái nghĩa dẫn đến sai số với các câu đơn giản như: “I’m not happy” có thể dễ bị phân loại nhầm thành “joy”.
    - Phụ thuộc vào tiền xử lý: Chưa xử lý triệt để các yếu tố như viết tắt, có biểu tượng cảm xúc hoặc các từ địa phương riêng biệt.

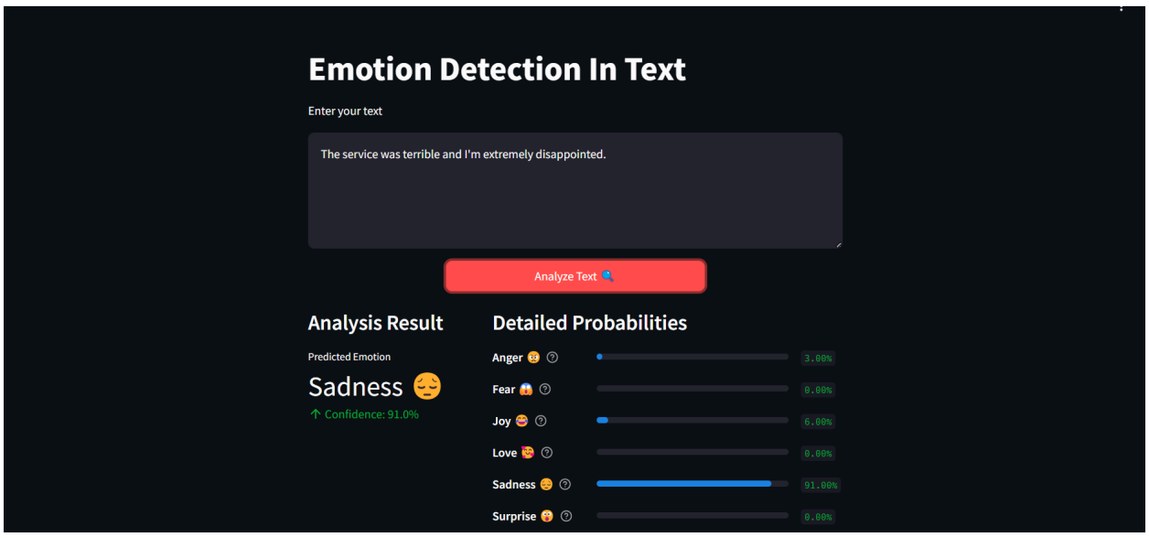
6.4. Demo ứng dụng.

Giao diện ứng dụng ban đầu: Bọn em sử dụng thư viện Streamlit để hoàn thiện giao diện đồng thời tạo space trên hugging face để có đường link cụ thể:



Hình 6.5. Giao diện sản phẩm

Giao diện khi đưa vào phân tích và dự đoán cảm xúc cho văn bản:



Hình 6.6. Kết quả thử nghiệm

Đường link demo: https://huggingface.co/spaces/HaoHao2915/Detect_Emotion_In_Text

7. Tổng kết bài tập lớn

Báo cáo đã tổng kết quá trình thực hiện đề tài nhận diện cảm xúc trong văn bản.

Trưởng nhóm: Nguyễn Quang Nhật.

Tên	Nhiệm vụ	Phân công	Mức độ hoàn thành
Nguyễn Quang Nhật + Nguyễn Ngọc Hảo	- Nghiên cứu cơ sở lý thuyết. - Thiết kế các mô hình. - Lập trình mô hình. - Kiểm thứ ứng dụng. - Viết báo cáo. - Thiết kế slide.	Chia đều (50% mỗi người)	50% mỗi người
Nguyễn Quang Nhật	- Tìm kiếm dữ liệu. - Xử lý dữ liệu - Khai phá dữ liệu	100%	100%
Nguyễn Ngọc Hảo	- Đánh giá mô hình. - Phát triển ứng dụng từ mô hình.	100%	100%

Bảng 7.1: Bảng phân công công việc

8. Tài liệu tham khảo

[AI From Scratch][Basic ML] #3 - Logistic Regression. (n.d.). Viblo. Retrieved March 12, 2025, from https://viblo.asia/p/ai-from-scratchbasic-ml-3-logistic-regression-GrLZDJDw5k0

dair-ai/emotion · Datasets at Hugging Face. (n.d.). Hugging Face. Retrieved March 12, 2025, from https://huggingface.co/datasets/dair-ai/emotion

Essential Text Pre-processing Techniques for NLP! (2024, October 16). Analytics Vidhya. Retrieved March 12, 2025, from https://www.analyticsvidhya.com/blog/2021/09/essential-text-pre-processing-techniques-for-nlp/

Guide, S. (2024, November 11). *Introduction to Bag of Words (BoW) Model*. Analytics Vidhya. Retrieved March 12, 2025, from https://www.analyticsvidhya.com/blog/2020/02/quick-introduction-bag-of-words-bow-tf-idf/

Machine Learning cơ bản. (2017, February 17). Machine Learning cơ bản. Retrieved March 12, 2025, from https://machinelearningcoban.com/2017/02/17/softmax/#-ham-mat-mat-cho-softmax-regression

Machine Learning cơ bản. (2017, April 9). Machine Learning cơ bản. Retrieved March 12, 2025, from https://machinelearningcoban.com/2017/04/09/smv/

Mô hình kết hợp (ensemble model). (n.d.). https://phamdinhkhanh.github.io/deepai-book/ch_ml/RandomForest.html#mo-hinh-ket-hop-ensemble-model

Nguyễn, T. V. (n.d.). *Week 12 - Logistic Regression*.

tf–idf. (n.d.). Wikipedia. Retrieved March 12, 2025, from https://en.wikipedia.org/wiki/Tf%E2%80%93idf