BÁO CÁO ĐỀ TÀI
TRẠI HÈ LẬP TRÌNH PYTHON-AI
TRƯỜNG CNTT&TT - NĂM 2024

09 tháng 8, 2024

Đề tài: Mô hình Học máy Dự đoán cảm xúc văn bản

Tên nhóm:	F4 SOICT
Ngày thực hiện:	05-07/08/2024

Nhóm tác giả

Phạm Công Hoàng	K69 CTTT: Khoa học Dữ liệu và Trí tuệ Nhân tạo
Nguyễn Đức Dũng	K69 CTTT: Khoa học Dữ liệu và Trí tuệ Nhân tạo
Nguyễn Gia Minh	K69 CTTT: Khoa học Dữ liệu và Trí tuệ Nhân tạo
Nguyễn Danh Bảo	K69 CTTT: Khoa học Dữ liệu và Trí tuệ Nhân tạo

Mục lục

1 Lời nói đầu

Lời đầu tiên, nhóm tác giả chúng em xin cảm ơn các thầy, cô, anh, chị tại Trường Công nghệ Thông tin và Truyền thông, Đại học Bách Khoa Hà Nội đã tổ chức khóa học *AI Adventure* cho chúng em và các tân sinh viên K69 của trường để có thể học hỏi, làm quen và trau dồi kinh nghiệm.

Bài báo cáo nằm trong dự án cuối khóa do 4 bạn tân sinh viên K69 tại trường thực hiện. Do đây là lần đầu chúng em cùng nhau thực hiện một dự án về Học máy nên sản phẩm có thể có một vài sai sót. Chúng em mong các thầy, cô, anh, chị có thể thông cảm. Mọi góp ý, thắc mắc xin quý bạn đọc liên hệ với các thành viên trong nhóm.

- *Phạm Công Hoàng* - Email: hoangphamconglc2212@gmail.com
- *Nguyễn Đức Dũng* - Email: dung11071979@gmail.com
- *Nguyễn Gia Minh* - Email: ngminh209206@gmail.com
- *Nguyễn Danh Bảo* - Email: ndbgm2k6@gmail.com

2 Giới thiệu sản phẩm

2.1 Ý tưởng của sản phẩm

Ý tưởng của ”Dự án học máy dự đoán cảm xúc văn bản” là xây dựng một mô hình có thể xác định cảm xúc (tích cực hoặc tiêu cực) trong các văn bản như bình luận, bài đánh giá, nhận xét,... Hệ thống này sẽ đọc các đánh giá, phân tích nội dung văn bản, và sử dụng các thuật toán học máy để dự đoán cảm xúc của người viết.

2.2 Giới thiệu

Mô hình phân tích cảm xúc văn bản là ứng dụng của trí tuệ nhân tạo, sử dụng phân tích văn bản, ngôn ngữ học tính toán để xác định, trích xuất, định lượng và nghiên cứu các trạng thái cảm xúc và thông tin chủ quan một cách có hệ thống. Phân tích cảm xúc được áp dụng rộng rãi cho các tài liệu tiếng nói của khách hàng như đánh giá và phản hồi khảo sát, truyền thông trực tuyến và mạng xã hội, và tài liệu chăm sóc sức khỏe cho các ứng dụng từ tiếp thị đến dịch vụ khách hàng đến y học lâm sàng.

Mô hình chúng em tạo ra dựa vào tiền xử lý dữ liệu nạp vào, từ đó máy tính có thể phân tích các từ khóa và dự đoán trạng thái cảm xúc là tích cực hay tiêu cực. Mô hình mang lại lợi ích nhờ khả năng xử lý nhiều loại thông tin văn bản một cách chính xác. Miễn là phần mềm được đào tạo với đầy đủ các ví dụ, phân tích cảm xúc có thể dự đoán chính xác sắc thái cảm xúc của tin nhắn.

3 Mô tả dự án

3.1 Cấu hình, nền tảng, thư viện sử dụng

Để hoàn thiện dự án, chúng em triển khai mô hình trên máy tính cá nhân và nền tảng *Google Colab*. Cụ thể về cấu hình máy tính cục bộ:

Phần cứng

- CPU: Intel Xeon E3-1220 v3 @ 3.10GHz
- GPU: NVDIA GeForce GTX 1050 Ti - 4GB VRAM
- RAM: 16GB DDR3
- SSD: 256GB SATA3

Phần mềm

- OS: Windows 10 Pro 22H2
- IDE: Visual Studio Code 1.92.0
- Python 3.12.4
- Pandas 2.2.2, Seaborn 0.13.2, Matplotlib 3.9.1, Scikit-Learn 1.5.1.
- Mã nguồn của dự án được xuất bản trên Github Repository của nhóm

Các tệp Jupyter Notebook của dự án cũng được triển khai trên Google Colab với 2 mô hình (Tiếng Anh và Tiếng Việt)

3.2 Phân tích bộ dữ liệu văn bản

3.2.1 Bộ dữ liệu Đánh giá phim trên IMDb bằng Tiếng Anh

Bộ dữ liệu Đánh giá phim trên IMDb được nhóm lấy từ nền tảng *Kaggle* tại đây. Đây là bộ dữ liệu với 50.000 đánh giá về các bộ phim trên trang web IMDb.com được đánh nhãn tích cực (*positive*) hay tiêu cực (*negative*). Sau đây, chúng ta sẽ tiến hành phân tích bộ dữ liệu này.

Bộ dữ liệu bao gồm 2 cột là '*review*' ghi lại đánh giá và '*sentiment*' thể hiện cảm xúc tương ứng và 50.000 cột tương ứng là tổng số các đánh giá có trong bộ dữ liệu này.

	review	sentiment
0	One of the other reviewers has mentioned that ...	positive
1	A wonderful little production. The...	positive
2	I thought this was a wonderful way to spend ti...	positive
3	Basically there's a family where a little boy ...	negative
4	Petter Mattei's "Love in the Time of Money" is...	positive
...	...	...
49995	I thought this movie did a down right good job...	positive
49996	Bad plot, bad dialogue, bad acting, idiotic di...	negative
49997	I am a Catholic taught in parochial elementary...	negative
49998	I'm going to have to disagree with the previou...	negative
49999	No one expects the Star Trek movies to be high...	negative

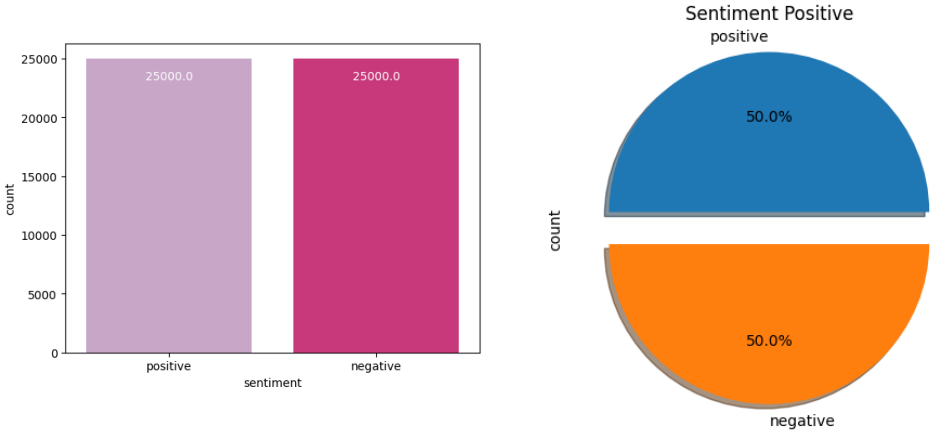
Hình 1: Tổng quan bộ dữ liệu đánh giá phim

Tiếp theo, ta thấy rằng bộ dữ liệu không có dữ liệu bị thiếu, hay tất cả các đánh giá trong đây đều được gán nhãn.

	missing_values	percent_missing %
review	0	0.0
sentiment	0	0.0

Hình 2: Kết quả của hàm tìm dữ liệu thiếu

Hơn nữa, bộ dữ liệu rất cân bằng, tỷ lệ đánh giá tiêu cực và tích cực đều bằng nhau với 25.000 đánh giá mỗi loại.



Hình 3: Tương quan giữa đánh giá tích cực và tiêu cực

Tóm lại, đây là một bộ dữ liệu về văn bản khá lớn với sự cân bằng rất tốt về dữ liệu, không có dữ liệu bị thiếu nên giúp chúng em dễ dàng để xử lý và triển khai vào mô hình.

3.2.2 Bộ dữ liệu Đánh giá sản phẩm bằng Tiếng Việt

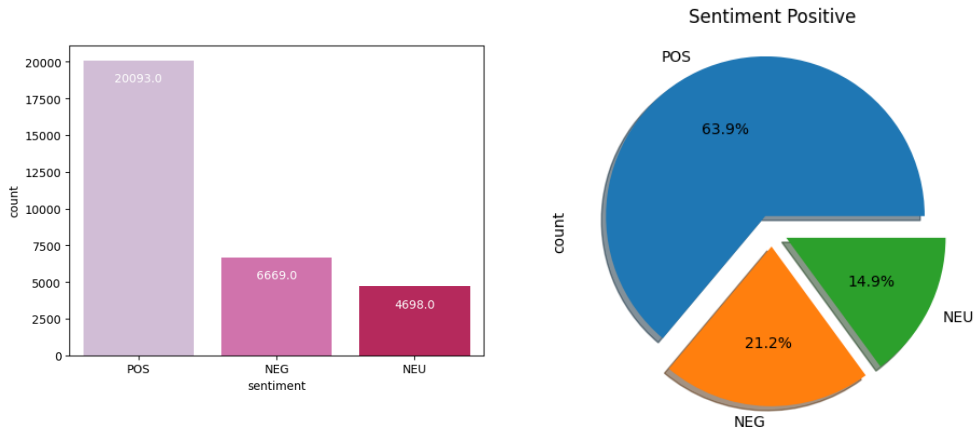
Tiếp theo ta sẽ đi đến phân tích bộ dữ liệu đánh giá sản phẩm, được chúng em lấy trên *Kaggle* tại đây.

	review	sentiment	rate	Unnamed: 3
0	Áo bao đẹp ạ!!	POS	5	NaN
1	Tuyệt vời !	POS	5	NaN
2	2day ao khong giong trong.	NEG	1	NaN
3	Mùi thơm,bôi lên da mềm da.	POS	5	NaN
4	Vải đẹp, dày dặn.	POS	5	NaN
...	...	...	...	...
31455	Không đáng tiền.	NEG	1	NaN
31456	Quần rất đẹp.	POS	5	NaN
31457	Hàng đẹp đúng giá tiền.	POS	5	NaN
31458	Chất vải khá ổn.	POS	4	NaN
31459	áo rất ok nhé , vải mịn , len cao cổ này phối ...	POS	5	NaN

Hình 4: Tổng quan bộ dữ liệu đánh giá sản phẩm

Đánh giá cơ bản đây bộ dữ liệu này không nhiều (31.460 đánh giá) như bộ dữ liệu về đánh giá phim ở trên (50.000 đánh giá). Hơn nữa, các đánh giá ngoài được đánh nhãn tích cực (*POS*) hoặc tiêu cực (*NEG*) thì ở bộ dữ liệu này còn có thêm nhẫn trung lập (*NEU*) thể hiện cảm xúc không nghiêng về mặt tiêu cực hay tích cực. Ngoài ra, ở bộ dữ liệu còn thừa một cột ở cuối không có dữ liệu. Tuy nhiên, bộ dữ liệu vẫn không có dữ liệu thiếu và hơn nữa, có thêm một cột '*rate*' gồm các giá trị nguyên từ 1 đến 5, thể hiện mức độ cảm xúc (1 - rất tiêu cực, 2 - khá tiêu cực, 3 - trung lập, 4 - khá tích cực, 5 - rất tích cực). Tuy nhiên, ở mô hình chúng ta triển khai sẽ không sử dụng cột này do chỉ cần dự đoán được cảm xúc là được.

Tiếp theo, ta xem xét về tương quan giữa các đánh giá tích cực, tiêu cực và trung lập. Ở biểu đồ dưới dây, ta thấy rằng, số lượng các đánh giá tích cực chiếm tới 63.9% số lượng dữ liệu và chỉ có 21.2% là đánh giá tiêu cực và 14.9% đánh giá tích cực.



Hình 5: Tương quan giữa đánh giá tích cực, tiêu cực và trung lập

Có thể nói đây là một bộ dữ liệu không cân bằng, điều này sẽ ảnh hưởng đến độ hiệu quả của mô hình, cụ thể, mô hình có thể đạt độ chính xác cao khi dự đoán các đánh giá tích cực nhưng lại kém trong dự đoán các đánh giá tiêu cực và trung lập. Tuy nhiên, do các bộ dữ liệu về văn bản Tiếng Việt khá khan hiếm và đây là bộ dữ liệu khá lớn hiếm hoi mà chúng em tìm được. Do đó, chúng em sẽ cố gắng xây dựng mô hình với bộ dữ liệu này.

3.3 Tiền xử lý dữ liệu

Ta sẽ đi vào xử lý cụ thể với mỗi bộ dữ liệu mà ta sử dụng. Đầu tiên, với bộ dữ liệu đánh giá phim bằng tiếng Anh, trước hết ta sẽ chuyển các ký tự viết hoa về ký tự viết thường để đơn giản hóa và tăng tính nhất quán. Sau đó, ta thực hiện loại bỏ các ký tự đặc biệt và các dấu câu trong văn bản. Ta sử dụng thư viện *nltk* để lấy dữ liệu *wordnet* nhằm loại bỏ các *stop word* trong văn bản, đây là các từ được thường xuyên xuất hiện trong tiếng anh nhưng không mang ý nghĩa để xác định cảm xúc. Cuối cùng là chuyển các từ ghép thành từ căn bản (*Lemmatization*), ví dụ running hay *ran* thành *run* bằng cách dùng hàm *WordNetLemmatizer()* trong thư viện *nltk*. Hàm tiền xử lý như sau:

```
nltk.download('wordnet')
def pre_process(text):
    #lower text
    text = text.lower()
    #remove special char
    text = re.sub(r'[^\w\s]', '', text)
    #remove stop word
    stop_words = set(stopwords.words('english'))
    words = text.split()
    filtered_words = [word for word in words if word not in stop_words]
    text = ' '.join(filtered_words)
    #Lemmatization
    lemmatizer = WordNetLemmatizer()
    words = text.split()
    lemmatized_words = [lemmatizer.lemmatize(word) for word in words]
    text = ' '.join(lemmatized_words)
    return text
```

Code 1: Hàm tiền xử lý dữ liệu Tiếng Anh

Tiếp theo là xử lý với bộ dữ liệu đánh giá sản phẩm bằng Tiếng Việt. Ta vẫn tiếp tục chuyển các ký tự in hoa về in thường và loại bỏ các dấu câu, ký tự đặc biệt. Tuy nhiên, chúng em thấy rằng ở nhiều câu bảng thán mang ý nghĩa tích cực thì thường sử dụng dấu nên ta sẽ không loại bỏ dấu này (Điều này theo thực nghiệm làm tăng hiệu quả của mô hình). Tuy thư viện nltk không hỗ trợ xóa stop word bằng tiếng Việt nên ta sẽ tự định nghĩa *stop-word* Tiếng Việt là những từ nối cơ bản thường gặp. Còn với việc *Lemmatization* văn bản tiếng Việt, ta sử dụng thư viện *pyvi* với hàm ViTokenizer.tokenize(). Cuối cùng, hàm tiền xử lý như sau:

```
def pre_process(text):
    #Chuyen ky tu viet thuong
    text = text.lower()
    #Xoa ky tu dau cau dac biet
    text = re.sub(r'[^\w\s!]', '', text)
    stop_words = set([
    'cua', 'la', 'va', 'co', 'de', 'voi', 'theo', 'nhu', 'boi', 'da'])
    words = text.split()
    filtered_words = [word for word in words if word not in stop_words]
    text = ' '.join(filtered_words)
    text = ViTokenizer.tokenize(text)
    return text
```

Code 2: Hàm tiền xử lý dữ liệu Tiếng Việt (ký tự trong code vẫn có dấu thanh)

Với sự không cân bằng của bộ dữ liệu Tiếng Việt, chúng em đã sử dụng các giải pháp over-sampling tuy nhiên kết quả của mô hình lại bị giảm đi nên trong dự án này, chúng em vẫn để dữ liệu ở tình trạng ban đầu.

Chúng em đều chia các bộ dữ liệu thành 2 tập *train* và tập *test* với tỷ lệ tương ứng là 4:1 và tiến đến bước vector hóa các dữ liệu. Phương pháp để thực hiện điều đó trong dự án này là *TF-IDF Vectorization*. Bằng việc sử dụng hàm TfidfVectorizer() với tham số max_features thích hợp (ở đây là 54.000 với bộ dữ liệu Tiếng Anh và 5000 với bộ tiếng Việt) vào toàn bộ dữ liệu của mỗi bộ dữ liệu. Lúc này các bộ dữ liệu văn bản đã được chuyển thành các vector, giúp dễ dàng triển khai các mô hình học máy để hoàn thiện dự án.

3.4 Các mô hình học máy với Scikit-Learn

Trong phần này, chúng ta thực hiện xây dựng mô hình với việc sử dụng thư viện nổi tiếng về học máy *Scikit-learn*. Do tập dữ liệu cho mô hình học là *vector* với số chiều khá lớn, 54.000 cho mô hình tiếng Anh và 5000 cho mô hình tiếng Việt, nên trong dự án này, chúng ta sẽ chỉ đi vào xây dựng và so sánh với 2 mô hình cơ bản là *Logistic Regression* và *Naive Bayes* vì tính đơn giản và thời gian huấn luyện ngắn.

3.4.1 Mô hình Logistic Regression

Nhắc lại một chút kiến thức về mô hình *Logistic Regression (LR)* đã học trong khóa *AI Adventure*, mô hình *LR* nhận đầu vào là các biến vector $x = [x_1, x_2, ..., x_n]$ với đầu ra thường là biến nhị phân là 0 hoặc 1 (Tuy nhiên bằng việc kết hợp các mô hình *LR* nhị phân thì đầu ra có thể nhận nhiều hơn 2 giá trị). Cụ thể, từ việc huấn luận, mô hình sẽ cho ra hàm tuyến tính:

$$z = w^T x = w_0 + w_1 x_1 + w_2 x_2 + ... + w_n x_n$$

Ở đây, vector w là vector chứa các trọng số thu được từ việc huấn luyện. Khi sử dụng mô hình, đầu vào của người dùng được chuyển thành vector n chiều là x và được đưa vào hàm *sigmoid*:

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

Đặc điểm của hàm *sigmoid* là nó bị chặn trong khoảng $(0,1)$ và $\lim_{x\to-\infty} \sigma(x) = 0$ và $\lim_{x\to+\infty} \sigma(x) = 1$, hơn nữa đạo hàm của hàm này đơn giản nên được sử dụng rộng rãi. Tùy vào từng loại bài toán mà giá trị của hàm *sigmoid* vượt ngưỡng (thường là 0.5) sẽ đầu ra là 1 và bằng 0 nếu ngược lại.

Trong mã nguồn của dự án, ta sẽ chỉ cần gọi thư viện *scikit-learn* và cho tập huấn luyện đã xử lý vào hàm *LogisticRegression()* là sẽ thu được mô hình.

3.4.2 Mô hình Naive Bayes

Đây là mô hình dựa trên định lý *Bayes* với giả định ngây thơ *(naive)* rằng các đặc trưng là độc lập với nhau trong mỗi lớp, vì trong hầu hết bài toán thì chúng luôn có liên quan đến nhau. Đầu vào của mô hình là *vector n chiều x*, mô hình sẽ phân loại x vào một trong C nhãn cho trước, bằng cách tính xác suất x rơi vào nhãn nào là lớn nhất. Tức là x được gán nhãn c nếu:

$$c = \underset{c\in\{1,\ldots,C\}}{\arg\max}\ p(c \mid x)$$

Theo định lý *Bayes* và việc $p(x)$ không phụ thuộc vào c ta viết lại thành:

$$c = \underset{c}{\arg\max}\ p(c \mid x) = \underset{c}{\arg\max}\ \frac{p(x \mid c)p(c)}{p(x)} = \underset{c}{\arg\max}\ p(x \mid c)p(c)$$

Giá trị của $p(c)$ được tính bằng tỷ lệ của dữ liệu nhãn c trong toàn bộ tập huấn luyện, còn $p(x \mid c$ theo giả định *naive* của ta thì cho bởi:

$$p(x|c) = p(x_1,\ldots,x_n|c) = \prod_{i=1}^{n} p(x_i|c)$$

Cả hai giá trị này được xác định khi huấn luyện. Trên đây là những ý cơ bản của *Naive Bayes*, tuy giả định *naive* có phần ngây ngô nhưng lại mang đến kết quả khá khả quan trong nhiều bài toán học máy. Trong dự án này, ta cũng chỉ cần gọi hàm *MultinomialNB()* vào tập huấn luyện là hoàn thiện mô hình này.

3.4.3 So sánh, đánh giá

Thấy rằng, hiệu quả của mỗi mô hình sẽ được tính bằng chỉ số *accurancy*, tức là tỷ số các dự đoán đúng trên tổng số dự đoán, do vai trò của mỗi nhãn tiêu cực, tích cực hay trung lập là như nhau. Sau đây là kết quả của 2 mô hình với bài toán dự đoán cảm xúc đánh giá phim bằng Tiếng Anh:

LOGISTIC REGRESSION

Accuracy Score : 90.25999999999999 %

Classification Report :

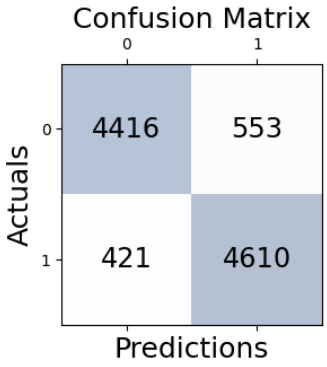
	precision	recall	f1-score	support
negative	0.91	0.89	0.90	4969
positive	0.89	0.92	0.90	5031
accuracy			0.90	10000
macro avg	0.90	0.90	0.90	10000
weighted avg	0.90	0.90	0.90	10000

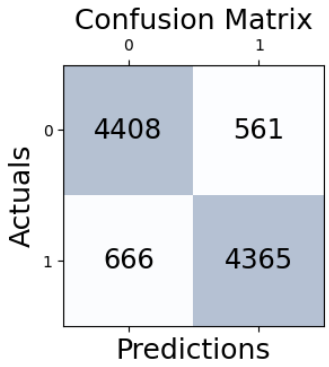
NAIVE BAYES

Accuracy Score : 87.72999999999999 %

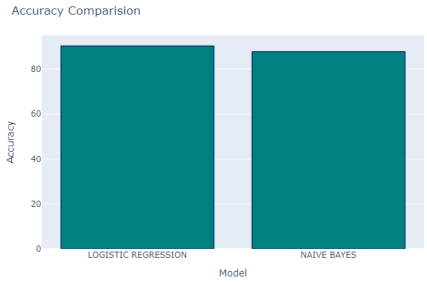
Classification Report :

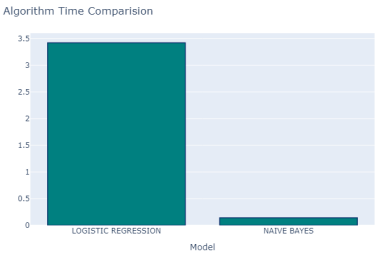
	precision	recall	f1-score	support
negative	0.87	0.89	0.88	4969
positive	0.89	0.87	0.88	5031
accuracy			0.88	10000
macro avg	0.88	0.88	0.88	10000
weighted avg	0.88	0.88	0.88	10000





Hình 6: Kết quả 2 mô hình với bài toán dự đoán tiếng Anh





Hình 7: So sánh giữa Accurancy và thời gian thực hiện 2 mô hình

Qua kết quả này, ta thấy rằng mô hình *Logistic Regression* thể hiện hiệu quả tốt với tỷ lệ dự đoán đúng là 90.25% và hơn 2.53% so với *Naive Bayes* mặc dù thời gian thực hiện lâu hơn nhưng không đáng kể với một bài toán học máy đơn giản.

Đối với bài toán dự đoán cảm xúc bằng tiếng Việt, kết quả vẫn nghiên về mô hình *LR* với tỷ lệ dự đoán đúng là 77.51% ở mức có thể chấp nhận và đã được dự tính từ trước.

LOGISTIC REGRESSION

Accuracy Score : 77.51112523839797 %

Classification Report :

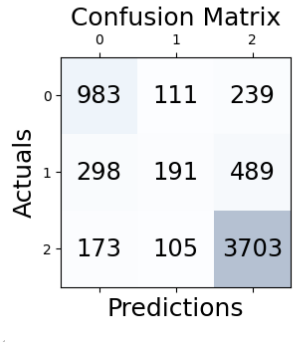
	precision	recall	f1-score	support
NEG	0.68	0.74	0.71	1333
NEU	0.47	0.20	0.28	978
POS	0.84	0.93	0.88	3981
accuracy			0.78	6292
macro avg	0.66	0.62	0.62	6292
weighted avg	0.74	0.78	0.75	6292

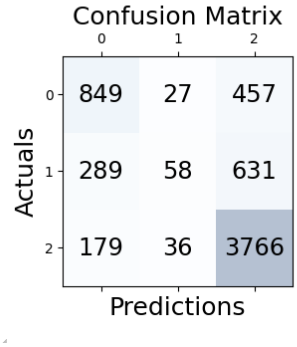
NAIVE BAYES

Accuracy Score : 74.26891290527654 %

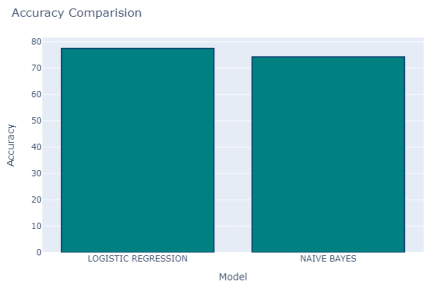
Classification Report :

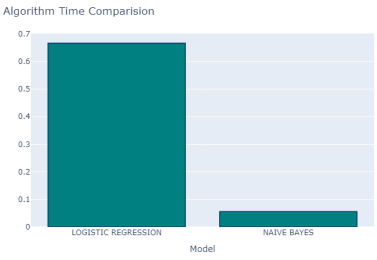
	precision	recall	f1-score	support
NEG	0.64	0.64	0.64	1333
NEU	0.48	0.06	0.11	978
POS	0.78	0.95	0.85	3981
accuracy			0.74	6292
macro avg	0.63	0.55	0.53	6292
weighted avg	0.70	0.74	0.69	6292





Hình 8: Kết quả 2 mô hình với bài toán dự đoán tiếng Việt





Hình 9: So sánh giữa Accurancy và thời gian thực hiện 2 mô hình

4 Đánh giá, mở rộng mô hình

4.1 Ưu điểm và nhược điểm của mô hình

1. **Ưu điểm**

    - Độ chính xác của mô hình xử lý văn bản tiếng Anh khá cao ($\approx$ 90%)
    - Có thể đánh giá các văn bản ở nhiều ngữ cảnh khác nhau
    - Sử dụng *Logistic Regression* và *Naive Bayes* giúp hiệu quả với nguồn dữ liệu nhỏ nhưng đem lại tính chính xác cao, tăng tốc độ tính toán, hiệu quả trong việc xử lý các lớp dữ liệu không cân bằng.

2. **Nhược điểm**

    - Chưa xử lý được đa dạng các ngôn ngữ: Mô hình này chỉ phân loại được cảm xúc của văn bản tiếng Anh và tiếng Việt, thậm chí với văn bản tiếng Việt thì có độ chính xác khá thấp ($\approx$ 78%) do hạn chế và số lượng dữ liệu nạp vào.
    - Một số văn bản dài, phức tạp có khả năng đánh giá sai
    - Chưa học được *icon*, tiếng lóng, viết tắt, *teencode*,...

4.2 Định hướng phát triển

Sau khi phân tích mặt tốt và hạn chế của sản phẩm, chúng em sẽ có một số định hướng phát triển và cải tiến sản phẩm như sau:

- Cải thiện *accuracy* của văn bản tiếng Việt bằng cách tăng lượng dữ liệu nạp vào.
- Đào tạo mô hình hiểu được thêm nhiều thứ tiếng, một số kí hiệu đặc biệt,...
- Đào tạo mô hình có thể sử dụng được trong đa dạng lĩnh vực hơn.
- Nâng cấp mô hình có thể phân loại được tính khách quan và chủ quan của văn bản.

5 Kết luận

Báo cáo này đã mô tả quá trình phát triển và đánh giá mô hình học máy để phân tích cảm xúc văn bản. Chúng em đã thực hiện các bước từ tiền xử lý dữ liệu đến xây dựng và xây dựng một số mô hình khác nhau để đem tới hiệu suất tối ưu. Quá trình tiền xử lý giúp tối ưu hóa dữ liệu đầu vào, đồng thời các kỹ thuật xây dựng mô hình khác như *Logistic Regression* và *Naive Bayes* đã được áp dụng để phân tích ngữ nghĩa và cảm xúc của văn bản. Hơn nữa, mô hình đạt được mức độ chính xác cao, cho phép nhận diện cảm xúc với độ tin cậy đáng kể. Tuy nhiên, một số thách thức vẫn tồn tại, như khả năng xử lý cảm xúc văn bản phức tạp còn hạn chế và thiếu đa dạng ngữ cảnh và ngôn ngữ. Những vấn đề này chỉ ra rằng vẫn còn nhiều điểm để có thể phát triển và nâng cấp mô hình.

Kết quả nghiên cứu không chỉ chứng minh tính khả thi của mô hình phân tích cảm xúc mà còn mở ra hướng phát triển cho các ứng dụng thực tiễn trong nhiều lĩnh vực khác. Nhìn chung, báo cáo này đóng góp một cái nhìn sâu sắc vào lĩnh vực phân tích cảm xúc văn bản và cung cấp nền tảng cho các nghiên cứu và ứng dụng tiếp theo trong tương lai.

Và cuối cùng, chúng em xin cảm ơn thầy cô, các anh các chị đã dành thời gian theo dõi báo cáo về mô hình của nhóm chúng em. Bản báo cáo có thể còn một số sai sót bị bỏ lại, mong ban giám khảo và bạn đọc có thể thông cảm cho chúng em.

Xin chân thành cảm ơn.

Tài liệu

[1] https://en.wikipedia.org/wiki/Sentimentanalysis

[2] https://scikit-learn.org/stable/sklearn.linearmodel.LogisticRegression.html

[3] https://scikit-learn.org/stable/modules/naivebayes.html

[4] https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews

[5] https://www.kaggle.com/datasets/linhlpv/vietnamese-sentiment-analyst

[6] https://machinelearningcoban.com/