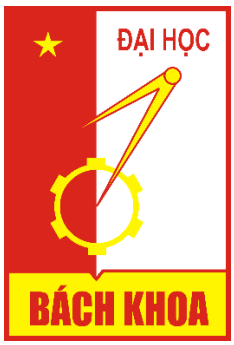
TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

----- □ & □ -----



KHOA HỌC DỮ LIỆU

Đề tài: PHÂN LOẠI PHẢN ÁNH CỦA NGƯỜI DÙNG TRÊN CÁC SÀN THƯƠNG MẠI ĐIỆN TỬ

Giảng viên: **PGS. TS. Thân Quang Khoát**

Nhóm sinh viên thực hiện:

STT	Họ và tên	MSSV
1	Mai Thu Hiền	20200203
2	Trần Thị Như Quỳnh	20205122
3	Nguyễn Duy Khánh	20204992
4	Nguyễn Thùy Dương	20204958
5	Lê Đức Anh Duy	20200111
6	Nguyễn Hà Phú Thịnh	20205131

Hà Nội, năm 2024

MỤC LỤC

MỤC LỤC **2**
PHÂN CHIA CÔNG VIỆC **3**
CHƯƠNG I. Giới thiệu đề tài **4**
1. Giới thiệu 4
2. Mục tiêu 4
CHƯƠNG II. Cách tiếp cận **5**
1. Thu thập dữ liệu 5
2. Tiền xử lí dữ liệu 7
3. Chuẩn hóa 7
   a) Chuẩn hóa unicode 7
   b) Chuẩn hóa dấu câu 8
   c) Đưa về chữ viết thường 8
   d) Chuẩn hóa câu 8
4. Phân tích dữ liệu sau khi tiền xử lý 9
5. Lựa chọn mô hình học máy: 11
   a) Random Forest: 11
   b) Support Vector Machine 13
   c) Logistic Regression 16
6. Lựa chọn mô hình học sâu 19
   a) Long Short Term Memory: 19
   b) Gated Recurrent Unit (GRU) 21
   c) Convolutional Neural Network (CNN) 22
CHƯƠNG III. Tiến hành dự đoán **25**
1. Tiến hành dự đoán trên các mô hình 25
   a) Mô hình học máy 25
   b) Mô hình học sâu 26
2. Đánh giá 27
   a) Mô hình học máy 27
   b) Mô hình học sâu 27
KẾT LUẬN **30**
TÀI LIỆU THAM KHẢO **30**

PHÂN CHIA CÔNG VIỆC

Họ tên	**MSSV**	**Công việc**
Mai Thu Hiền	20200203	Thu thập dữ liệu Tiền xử lý dữ liệu Gán nhãn dữ liệu Viết báo cáo
Trần Thị Như Quỳnh	20205122	Gán nhãn dữ liệu Làm slide Tiền xử lý dữ liệu
Nguyễn Duy Khánh	20204992	Gán nhãn dữ liệu Viết báo cáo
Nguyễn Thùy Dương	20204958	Gán nhãn dữ liệu Làm slide
Nguyễn Hà Phú Thịnh (Nhóm trưởng)	20205131	Gán nhãn dữ liệu Xử lý các model Phát triển mô hình học sâu Viết báo cáo
Lê Đức Anh Duy	20200111	Gán nhãn dữ liệu Triển khai mô hình lên hệ thống Viết báo cáo

CHƯƠNG I. Giới thiệu đề tài

1. Giới thiệu

Ngày nay, với sự phát triển mạnh mẽ của thương mại điện tử, các sàn giao dịch như Shopee, Lazada, và Tiki đã trở thành nơi mua sắm chính của người tiêu dùng. Người tiêu dùng ngày càng có nhiều cơ hội tiếp cận và lựa chọn các sản phẩm trên các nền tảng trực tuyến. Điều này cũng đồng nghĩa với việc người bán cần phải không ngừng nâng cao chất lượng sản phẩm cũng như dịch vụ để có thể giữ được vị trí trên thị trường. Vậy nên việc nắm bắt được những phản hồi của người dùng về sản phẩm là rất quan trọng, đặc biệt là những lời phê bình để giúp họ cải thiện được chất lượng dịch vụ. Các phản hồi của người dùng thường chứa đựng nhiều thông tin quan trọng về chất lượng sản phẩm, dịch vụ khách hàng, cũng như các vấn đề phát sinh trong quá trình mua sắm. Việc phân loại đúng đắn các phản hồi này không chỉ giúp người tiêu dùng có cái nhìn chính xác hơn về sản phẩm trước khi quyết định mua hàng mà còn là cơ sở để cho những người bán hàng cũng như bản thân nền tảng thương mại điện tử dựa vào đó để cải thiện sản phẩm và dịch vụ.

Trên cơ sở đó nhóm chúng em đã phát triển hệ thống này giúp người bán có thể nhận biết thiếu sót của sản phẩm một cách nhanh chóng.

2. Mục tiêu

Mục tiêu của bài tập lớn này là thử nghiệm một số phương pháp nhằm phân tích dữ liệu và thực hiện phân loại phản hồi của người dùng thu thập được từ các trang thương mại điện tử. Nhóm chúng em sẽ tập trung vào những lời phê bình, phàn nàn của người dùng về chất lượng, dịch vụ của từng đơn hàng, vì đây là những thông tin có giá trị đối với người bán hàng online cũng như nền tảng thương mại điện tử. Hệ thống sẽ sử dụng dữ liệu thu thập từ các phàn nàn (dưới 3 sao) của người dùng trên trang thương mại điện tử Tiki. Từ đó, áp dụng các kỹ thuật trong học máy và khai phá dữ liệu để phân loại các bình luận này theo các vấn đề cụ thể như chất lượng sản phẩm, vận chuyển, đóng gói và dịch vụ khách hàng.

CHƯƠNG II. Cách tiếp cận

1. Thu thập dữ liệu

Để có thể thu thập được bình luận của từng sản phẩm, trước hết ta cần thu được id của từng sản phẩm. Sau khi quan sát phần network trong danh mục trên Tiki, nhóm em tìm thấy api của Tiki có phần danh sách "data" trong response JSON, trong đó id là product id của sản phẩm.


```
},
"data": [
  {
    "id": 262731552,
    "sku": "4068266701990",
    "name": "Giường xếp gấp gọn di động SUMIKA 386, khung thép, có bánh xe",
    "url_key": "giuong-xep-gap-gon-di-dong-sumika-386-khung-thep-co-banh-xe-p262731552",
    "url_path": "giuong-xep-gap-gon-di-dong-sumika-386-khung-thep-co-banh-xe-p262731552.html?spid=262731553",
    "type": "",
    "author_name": "",
    "book_cover": null,
    "brand_name": "Sumika",
    "short_description": "",
    "price": 1550000,
    "list_price": 0,
    "badges": [],
    "badges_new": [
```


Dưới đây là api của Tiki.

https://tiki.vn/api/personalish/v1/blocks/listings?**limit=40**&include=advertisement&aggregations=2&version=home-persionalized&trackity_id=50ca9494-98f2-d786-350b-fbaa25c6fe6f&**category=1789**&**page=1**

Trong đó:

- Limit = 40 có ý nghĩa là hiển thị 40 sản phẩm trên 1 trang
- Category = 1789 là mã danh mục của sản phẩm
- Page = 1 là vị trí trang hiện tại

Với mỗi sản phẩm nhóm em sẽ lấy ra trường id và name bằng cách như sau:

```
prod_id = []
for i in range(1, 30):
    params['page'] = i
    response = requests.get('https://tiki.vn/api/personalish/v1/blocks/listings', headers=header, params=params)
    if response.status_code == 200:
        for record in response.json().get('data'):
            prod_id.append({'id': record.get('id'),
                            'name': record.get('name')})
    time.sleep(random.randrange(3, 10))
    if i % 5 ==0:
        df = pd.DataFrame(prod_id)
        df.to_csv('data/dth-tiki/result/product_id_ncds.csv', index=False)
```

Nhóm em sẽ lặp lại qua các trang bằng cách cập nhập params page trên api. Sau khi nhận được response thành công, nhóm em sẽ lấy giá trị id sản phẩm và tên sản phẩm trong danh sách "data" trong response JSON và lưu vào file csv. Kết quả thu được sẽ có dạng như thế này:

```
id,name
173387789,Máy đọc sách All New Kindle Oasis 3 - Hàng nhập khẩu
252586291,Máy đọc sách All New Kindle Paperwhite 5 (11th Gen) - Hàng chính hãng
201067649,Combo máy đọc sách Kindle Paperwhite 5 (11th gen) tặng kèm bao da ( Cover ) - Hàng nhập khẩu
125182567,Máy đọc sách All New Kindle Paperwhite 5 (11th Gen) - Hàng nhập khẩu
217645692,Combo Máy đọc sách All New Kindle Paperwhite 5 (11th Gen) - 16Gb và Bao da - Hàng nhập khẩu
184036446,Apple iPhone 11
184059211,Apple iPhone 13
183330021,Apple iPad 10.2-inch (9th Gen) Wi-Fi 2021
57809866,Điện thoại Xiaomi Redmi 9A (2GB/32GB)
125182567,Máy đọc sách All New Kindle Paperwhite 5 (11th Gen) - Hàng nhập khẩu
43924153,Máy đọc sách Kindle Oasis 3 (2019) - Amazon - Hàng nhập khẩu
217706932,Combo Máy đọc sách All New Kindle Paperwhite 5 (11th Gen) và Bao da - Hàng nhập khẩu
217640502,Máy đọc sách Kindle Scribe - Hàng chính hãng
271362153,"Combo Máy đọc sách All New Kindle Paperwhite 5 (11th Gen) và Bao da FOR KIDS (Bản KIDS, Không Quảng Cáo) - Hàng chính hãng"
273559020,Máy tính bảng Kindle Fire HD10 2023 13th - 32Gb - Hàng chính hãng
120295859,Điện Thoại Nokia C30 (2GB/32GB) - Hàng Chính Hãng
```

Sau khi thành công thu thập được id sản phẩm, nhóm em quan sát phần network trong trang chi tiết sản phẩm trong Tiki thì nhóm em tìm thấy api của Tiki có phần danh sách "data" trong response JSON chứa các thông tin nhóm em cần: content tương ứng với bình luận, rating tương ứng với đánh giá người mua.


```
  "data": [
    {
      "id": 19681388,
      "title": "Cực kì hài lòng",
      "content": "Quá ưng luôn. \r\nNỗi đau khổ khi bị mất con kindle ppw4 đã mau chóng bay vèo khi nhận được em ppw5. \r\nMàn hình màu ấm rất tuyệt vời, máy màu denim cũng rất đẹp và sang. \r\nShop gửi tặng cái ốp da màu Denim nữa, đỉnh  ghê. Cám ơn shop tinh tế",
      "status": "approved",
      "thank_count": 3,
      "score": 0.02820157,
      "new_score": 0.18053143,
      "customer_id": 1054513,
      "comment_count": 0,
      "rating": 5,
```


Dưới đây là ví dụ một api của Tiki:

https://tiki.vn/api/v2/reviews?**limit=5**&include=comments,contribute_info,attribute_vote_summary&sort=score%7Cdesc,id%7Cdesc,stars%7Call&**page=1**&spid=125182569&**product_id=125182567**

Trong đó:

- Limit = 5 có nghĩa là chỉ hiện 5 bình luận trên 1 trang
- Page = 1 là vị trí trang hiện tại
- Product_id là những giá trị id của sản phẩm mình vừa thu thập bên trên

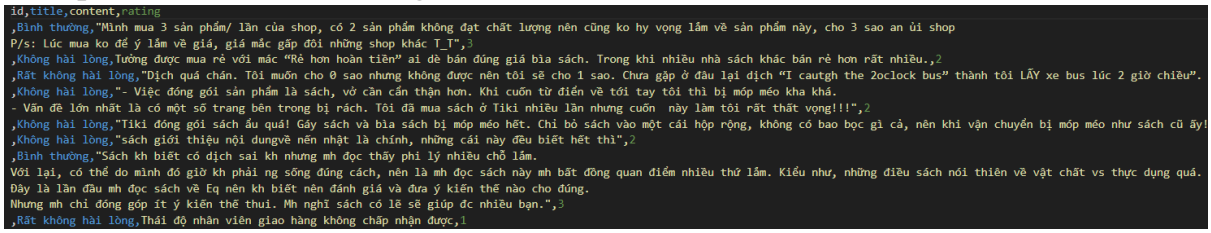
Vậy nên để có thể thu thập được bình luận sản phẩm ta sẽ đọc id sản phẩm từ file csv ta vừa thu thập được bên trên, áp dụng tương tự như cách thu được id sản phẩm ta sẽ được kết quả như sau:

```
id,title,content,rating
,Cực kì hài lòng,"Lần đầu mua máy đọc sách.
Mất rất nhiều thời gian suy nghĩ chọn PPW5 hay OA3.Cuối cùng quyết định mua luôn OA3 bản 32Gb.
Món quà tặng tuyệt vời cho cô em gái mới thi lên lớp 10.

Sản phẩm có vỏ nhôm cao cấp,nhẹ,độ hoàn thiện cao...giao hàng nhanh.Sản phẩm đáng để mua vì đầu tư cho tri thức,cho giáo dục.",5
,Rất không hài lòng,"Mình đặt bản 32gb, màu gold; bên bán hàng giao màu graphite mà không hỏi ý kiến bên mua. Dịch vụ quá tệ.",1
,Cực kì hài lòng,Máy xài thích! Cầm rất thoải mái và chắc tay,5
,Cực kì hài lòng,Hàng mới nguyên seal. Shop tư vấn rất nhiệt tình.,5
,Hài lòng,,4
,Cực kì hài lòng,,5
,Cực kì hài lòng,,5
,Rất không hài lòng,Sp khác review inbox trên YouTube và hướng dẫn chính hãng,1
,Cực kì hài lòng,,5
,Hài lòng,"mua trên tiki đắt hơn mua từ nhà phân phối 200k. Giá tiki ghi là 7350k, giá hóa đơn 7139k",4
,Cực kì hài lòng,,5
,Cực kì hài lòng,Sản phẩm như mô tả!,5
,Cực kì hài lòng,,5
,Cực kì hài lòng,service rat tot. giao hang cung nhanh!!,5
,Cực kì hài lòng,,5
,Cực kì hài lòng,,5
,Cực kì hài lòng,,5
,Cực kì hài lòng,,5
,Cực kì hài lòng,,5
,Cực kì hài lòng,,5
```

2. Tiền xử lí dữ liệu

Sau khi thu thập được các bình luận, nhóm em sẽ chỉ giữ lại những bình luận có rating từ 3 đổ xuống (tương ứng với những bình luận tiêu cực) và các bình luận. Kết quả thu được sẽ có dạng như sau:



```
id,title,content,rating
,Bình thường,"Mình mua 3 sản phẩm/ lần của shop, có 2 sản phẩm không đạt chất lượng nên cũng ko hy vọng lắm về sản phẩm này, cho 3 sao an ủi shop
P/s: Lúc mua ko để ý lắm về giá, giá mắc gấp đôi những shop khác T_T",3
,Không hài lòng,Tưởng được mua rẻ với mác “Rẻ hơn hoàn tiền” ai dè bán đúng giá bìa sách. Trong khi nhiều nhà sách khác bán rẻ hơn rất nhiều.,2
,Rất không hài lòng,"Dịch quá chán. Tôi muốn cho 0 sao nhưng không được nên tôi sẽ cho 1 sao. Chưa gặp ở đâu lại dịch “I cautgh the 2oclock bus” thành tôi LẤY xe bus lúc 2 giờ chiều”.
,Không hài lòng,"- Việc đóng gói sản phẩm là sách, vở cần cẩn thận hơn. Khi cuốn từ điển về tới tay tôi thì bị móp méo kha khá.
- Vấn đề lớn nhất là có một số trang bên trong bị rách. Tôi đã mua sách ở Tiki nhiều lần nhưng cuốn  này làm tôi rất thất vọng!!!",2
,Không hài lòng,"Tiki đóng gói sách ẩu quá! Gáy sách và bìa sách bị móp méo hết. Chỉ bỏ sách vào một cái hộp rộng, không có bao bọc gì cả, nên khi vận chuyển bị móp méo như sách cũ ấy!
,Không hài lòng,"sách giới thiệu nội dungvề nên nhật là chính, những cái này đều biết hết thì",2
,Bình thường,"Sách kh biết có dịch sai kh nhưng mh đọc thấy phi lý nhiều chỗ lắm.
Với lại, có thể do mình đó giờ kh phải ng sống đúng cách, nên là mh đọc sách này mh bất đồng quan điểm nhiều thứ lắm. Kiểu như, những điều sách nói thiên về vật chất vs thực dụng quá.
Đây là lần đầu mh đọc sách về Eq nên kh biết nên đánh giá và đưa ý kiến thế nào cho đúng.
Nhưng mh chỉ đóng góp ít ý kiến thế thui. Mh nghĩ sách có lẽ sẽ giúp đc nhiều bạn.",3
,Rất không hài lòng,Thái độ nhân viên giao hàng không chấp nhận được,1
```

Sau đó nhóm em tiến hành gán nhãn cho dữ liệu. Nhóm quy định sẽ có 4 labels: Quality, Shipping, Packing, Service. Các giá trị sẽ bao gồm 1 – tương ứng với bình luận tích cực về 1 trong 4 nhãn trên, 0 – tương ứng với bình luận tiêu cực về 1 trong 4 nhãn trên, -1 tương ứng với ô rỗng:

	content	Quality	Service	Shipping	Packing	Label
0	dùng chán lắm, mau hết pin	0	-1.0	-1	-1.0	Quality
1	Mình đặt hàng màu trắng shop giao màu đen	-1	0.0	-1	-1.0	Service
5	Không hiệu quả, chuột vẫn vào	0	-1.0	-1	-1.0	Quality
8	Cắm điện đèn sáng nhưng không có sạc usb như t...	0	-1.0	-1	-1.0	Quality
9	Tệ: giao nhầm hàng. Yc giao lại sớm nhất.	-1	0.0	-1	-1.0	Service
10	ổ lỗi vì khi cắm phích vào lỗng lẻo ko chặt. đ...	0	-1.0	-1	-1.0	Quality
13	giao cho tôi găng tay thực phẩm của người nào ...	-1	0.0	-1	-1.0	Service
14	Miếng dán không đúng như hình giới thiệu.	0	-1.0	-1	-1.0	Quality

3. Chuẩn hóa

a) Chuẩn hóa unicode

Chuyển đổi các bình luận về chuẩn NFC của unicode.

Lí do phải chuẩn hóa unicode là có thể người dùng sử dụng bộ mã khác nhau khi gõ tiếng Việt, dẫn đến mặc dù 2 chữ y hệt nhau nhưng máy sẽ dịch thành 2 từ khác nhau. Ví dụ kí tự Å có thể được biểu diễn thành "\u00c5" hoặc chia thành ký tự A và vòng tròn "A\u030A".

Ở đây nhóm quyết định chuyển thành chuẩn NFC (Canonical Composition):

- Composition: Chuyển đổi các ký tự nhiều mã thành ký tự một mã, như ví dụ trên là "A\u030A" => "\u00c5"
- Canonical: Bảo toàn nguyên dạng, ví dụ "2^5" sau khi chuẩn hóa vẫn sẽ là "2^5"

Vậy sau khi dùng chuẩn NFC thì máy sẽ dịch như sau : (Å) "A\u030A" => (Å) "\u00c5"

b) Chuẩn hóa dấu câu

Trong tiếng Việt có 2 kiểu gõ dấu câu như sau: “hóa” và “hoá”. Do đó, có thể có các từ giống nhau nhưng lại có cách viết khác nhau khi chúng ta sử dụng dữ liệu văn bản vào bài toán học máy. Do đó, ta cần đưa chúng về một bộ gõ tiêu chuẩn (ở đây nhóm chọn tiêu chuẩn là “hóa”).

Để làm được vậy, ta sẽ lặp qua từng ký tự trong từ và xác định vị trí của các nguyên âm và xác định vị trí của các dấu thanh (ấn, sắc, huyền, ngã, nặng) trên các nguyên âm này.

Dựa trên thông tin về vị trí nguyên âm và dấu thanh, ta sẽ cập nhật lại các ký tự nguyên âm với dấu thanh chính xác. Ví dụ, nếu gặp "a" không dấu, nó sẽ được thay thế bằng "à", "á", "ả", "ã" hoặc "ạ" tùy thuộc vào dấu thanh xác định.

Nhưng có những trường hợp đặc biệt như “qu” và “gi”. Trong tiếng Việt, chuỗi "qu" thường biểu thị một âm vị riêng biệt, khác với việc đọc riêng "q" và "u". Trường hợp này ta sẽ coi “qu” như 1 kí tự đơn và thực hiện như bước trên.

c) Đưa về chữ viết thường

Các trường hợp viết hoa sẽ được xử lí thành viết thường. Ví dụ: “BÁO CÁO” hoặc “Báo Cáo” sẽ được chuyển thành “báo cáo”

d) Chuẩn hóa câu

- Thay thế các khoảng trắng liên tiếp bằng một khoảng trắng duy nhất và loại bỏ khoảng trắng ở đầu và cuối chuỗi
- Xóa các ký tự không hợp lệ trong văn bản tiếng Việt
- Loại bỏ các dấu câu !"#$%&'()*+,-./:;<=>?@[\]^_\{|}~` bằng cách thay thế chúng bằng khoảng trắng. Ví dụ khi đưa vào "Hello, world!" đầu ra sẽ là "Hello world"
- Xử lý các từ stopwords. Các từ stopwords là các từ mang ít giá trị ý nghĩa và không khác nhau nhiều trong các văn bản khác nhau, ví dụ là từ “các”, “những”, “rất”,… Để xử lí các từ này nhóm em sẽ duyệt qua file có sẵn những từ stopwords trong tiếng Việt và loại bỏ các từ có trong câu nếu tồn tại trong file.
- Xóa các emoji. Nhóm em xử lí bằng cách tạo 1 mảng chứa các unicode của emoji và xóa các kí tự đó nếu tồn tại trong câu.
- Thay thế các từ viết tắt. Ví dụ “k” thành “không”, “sp” thành “sản phẩm”,… Nhóm em xử lí bằng cách tạo 1 mảng chứa các từ viết tắt thay thế các từ đó nếu tồn tại trong câu.

Dữ liệu sau khi chuẩn hóa sẽ có dạng như thế này:

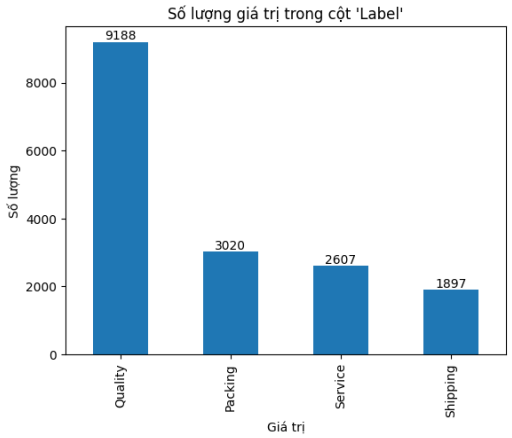
	content	Quality	Service	Shipping	Packing	Label
0	chán lắm mau pin	0	-1.0	-1	-1.0	Quality
1	hàng màu trắng cửa hàng giao màu đen	-1	0.0	-1	-1.0	Service
5	hiệu quả chuột	0	-1.0	-1	-1.0	Quality
8	cắm điện đèn sạc usb mô tả	0	-1.0	-1	-1.0	Quality
9	tệ giao nhầm hàng yc giao nhất	-1	0.0	-1	-1.0	Service
10	ổ lỗi cắm phích lõng lẽo không chặt 1 sao	0	-1.0	-1	-1.0	Quality
13	giao găng thực phẩm cục khuyếch đại wifi mercusy	-1	0.0	-1	-1.0	Service
14	miếng dán hình giới thiệu	0	-1.0	-1	-1.0	Quality
15	sản phẩm kém chất lượng đầu ống đầu vòi không ...	0	-1.0	-1	-1.0	Quality

4. Phân tích dữ liệu sau khi tiền xử lý

Sau khi thu thập dữ liệu, nhóm em nhận thấy có quá ít dữ liệu thuộc các label Packing, Service, Shipping. Cụ thể số lượng dữ liệu cho Quality là 9188, nhưng số lượng cho các label khác chỉ trong khoảng chưa tới 1000 dữ liệu.

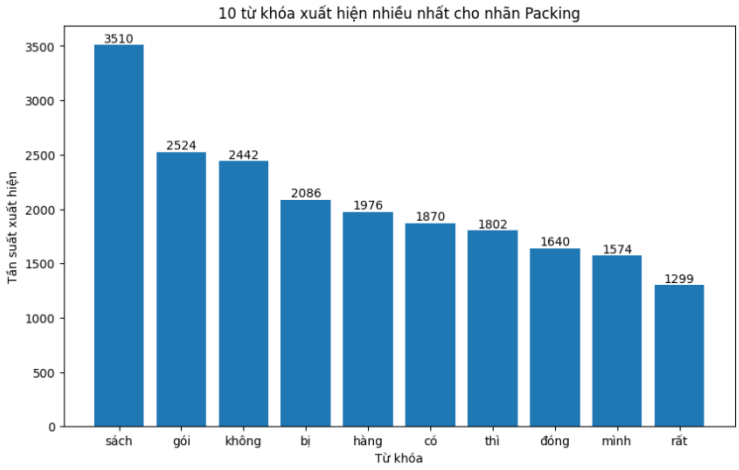
Vậy nên để mở rộng thêm dữ liệu, nhóm em đã sử dụng phương pháp Synonym Replacement để tăng cường tệp dữ liệu. Synonym Replacement là một kỹ thuật trong đó ta thay thế một từ bằng một trong các từ đồng nghĩa của nó. Sau khi thay thế từ, vì bản chất các bình luận đó không đổi nên nó sẽ vẫn giữ label của bình luận gốc ban đầu, nên nhóm em sẽ gán label của bình luận gốc cho các bình luận mới đã được thay thế bằng các từ đồng nghĩa.

Số lượng dữ liệu thu được sau khi mở rộng dữ liệu :

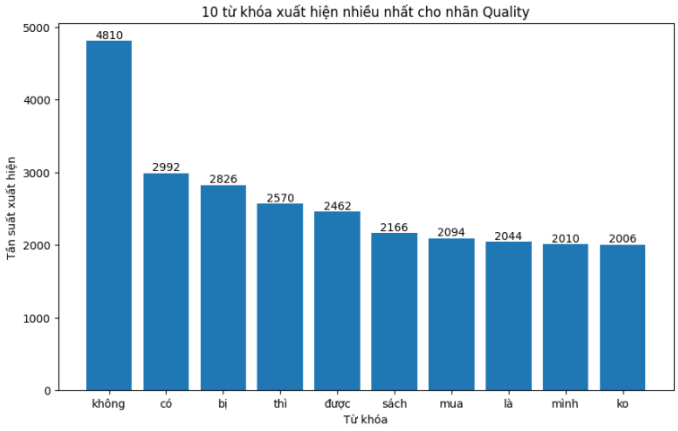


Các từ khóa xuất hiện nhiều nhất của mỗi nhãn :

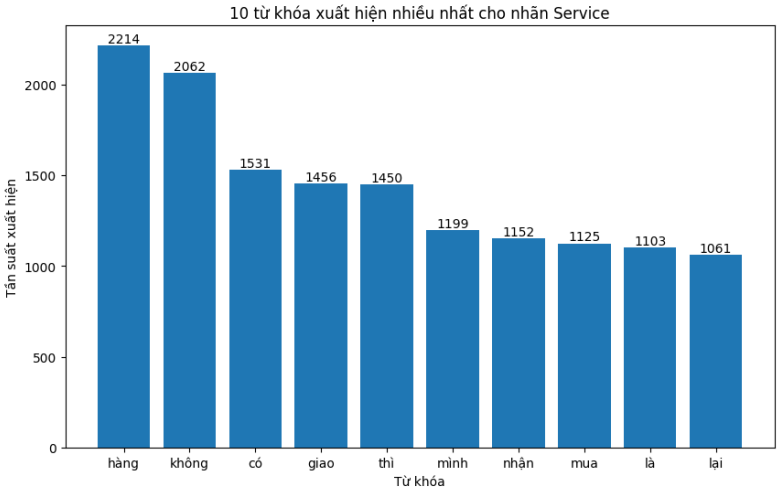
- Packing :



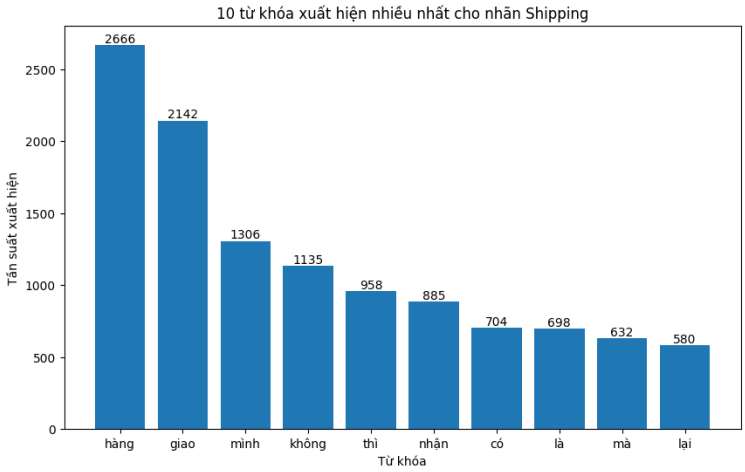
- Quality



- Service



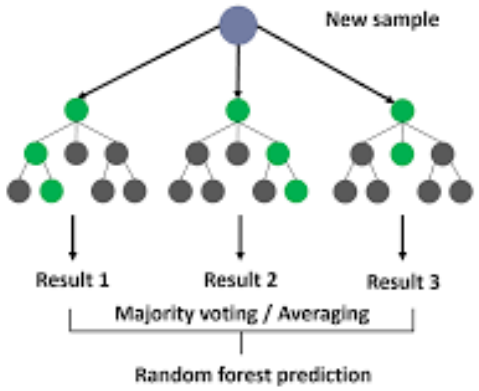
- Shipping



5. Lựa chọn mô hình học máy:

a) Random Forest:

Giới thiệu về Random Forest:



<source: medium.com >

- Random Forest là một phương pháp ensemble learning, tức là kết hợp nhiều mô hình học máy để tạo ra một mô hình mạnh mẽ hơn. Trong trường hợp của Random Forest, mô hình này kết hợp nhiều cây quyết định (decision trees) để đạt được hiệu suất tốt hơn.
  + Bagging (Bootstrap Aggregating): là kỹ thuật chủ đạo đứng sau Random Forest, giúp cho giảm biến động (variance) và cải thiện độ chính xác của mô hình. Quá trình bagging gồm các bước sau: Lấy mẫu dữ liệu và Xây dựng cây quyết định
  + Random Feature Selection : Một yếu tố quan trọng khác trong Random Forest là lựa chọn ngẫu nhiên các đặc trưng (features) tại mỗi nút của cây quyết định. Điều này giúp giảm sự tương quan giữa các cây và làm cho mô hình tổng thể mạnh mẽ hơn. Quá trình này bao gồm: Chọn đặc trưng ngẫu nhiên và chọn ngưỡng phân chia tốt nhất

+ Aggregation of Predictions: Sau khi các cây quyết định được xây dựng, Ramdom Forest tổng hợp dự đoán từ từng cây để đưa ra dự đoán cuối cùng. Đối với bài toán phân loại: Random Forest sử dụng phương pháp majority voting (lớp dự đoán cuối cùng là lớp được dự đoán nhiều nhất từ các cây)

Ưu điểm của Random Forest:

- **Giảm độ biến động và tránh overfitting:** việc sử dụng nhiều cây quyết định và lựa chọn đặc trưng ngẫu nhiên, Random Forest thường ít bị overfitting so với một cây quyết định đơn lẻ.
- **Độ chính xác cao:** Random Forest thường đạt được độ chính xác cao nhờ vào phương pháp tổng hợp..
- **Xử lý dữ liệu mất mát và không cân bằng:** Random Forest có khả năng xử lý tốt các dữ liệu thiếu và không cân bằng nhờ vào phương pháp bootstrap.
- **Ít nhạy cảm với các outliers:** Việc kết hợp nhiều cây quyết định giúp giảm ảnh hưởng của các outliers.

Random Forest trong bài toán:

```
# Huấn luyện Random Forest
rf_classifier = RandomForestClassifier(n_estimators=100, random_state=42)
rf_classifier.fit(X_train_tfidf, y_train)
```

- Mô hình Random
  + n_estimators=100: Số lượng cây quyết định (decision trees) trong rừng.
  + random_state=42: Thiết lập seed cho bộ sinh số ngẫu nhiên để đảm bảo tính tái lập của kết quả
- Huấn luyện mô hình:
  + X_train_tfidf: Dữ liệu huấn luyện đã được chuyển đổi thành ma trận TF-IDF.
  + **y_train**: Nhãn của dữ liệu huấn luyện.
- Đánh giá mô hình
  + Accuracy: Cho biết tỷ lệ phần trăm các dự đoán đúng.
  + Precision: Cho biết tỷ lệ phần trăm các dự đoán đúng trong số các dự đoán được thực hiện cho mỗi lớp.
  + Recall: Cho biết tỷ lệ phần trăm các dự đoán đúng trong số các mẫu thực sự thuộc về mỗi lớp. F1
  + Score: Là trung bình điều hòa của precision và recall, cung cấp một cái nhìn cân bằng giữa hai chỉ số này.

```python
# Đánh giá Random Forest
rf_y_pred = rf_classifier.predict(X_test_tfidf)
rf_accuracy = accuracy_score(y_test, rf_y_pred)
rf_precision = precision_score(y_test, rf_y_pred, average='weighted')
rf_recall = recall_score(y_test, rf_y_pred, average='weighted')
rf_f1 = f1_score(y_test, rf_y_pred, average='weighted')

print("Kết quả Random Forest:")
print(f"Accuracy: {rf_accuracy}")
print(f"Precision: {rf_precision}")
print(f"Recall: {rf_recall}")
print(f"F1 Score: {rf_f1}")
print("\n--------------------\n")
```

```
Kết quả Random Forest:
Độ chính xác: 0.9024096385542169
Độ chính xác (weighted): 0.9021775304094918
Độ thu hồi (weighted): 0.9024096385542169
Điểm F1 (weighted): 0.9021333667772969
```

- Kết luận: Mô hình Random Forest đạt được kết quả đáng kể trong việc dự đoán với độ chính xác cao, đồng thời có khả năng giảm thiểu tình trạng overfitting so với các mô hình cây quyết định đơn lẻ nhờ vào phương pháp tổng hợp (ensemble). Cụ thể, mô hình đã cho thấy:

  + Accuracy: Độ chính xác tổng thể của mô hình là cao, cho thấy mô hình có khả năng phân loại chính xác phần lớn các mẫu trong tập kiểm tra.

  + Precision: Tỷ lệ dự đoán đúng trong số các dự đoán đã thực hiện là tốt, đặc biệt là khi có nhiều lớp.

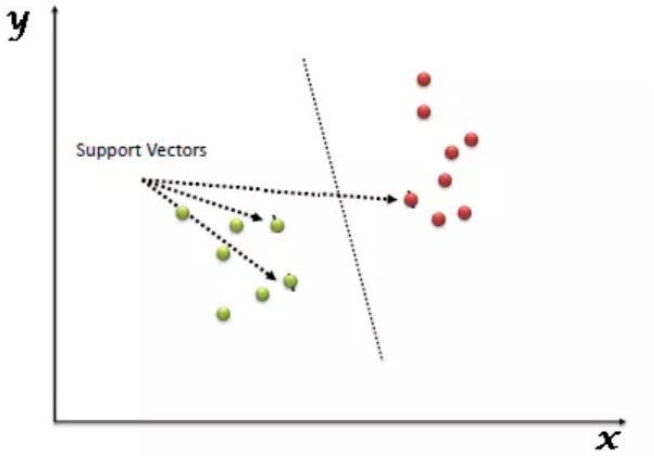
  + Recall: Tỷ lệ dự đoán đúng trong số các mẫu thực sự thuộc về mỗi lớp là khá tốt, cho thấy mô hình ít bỏ sót các mẫu quan trọng.

  + F1 Score: Sự cân bằng giữa precision và recall là tốt, cho thấy mô hình có hiệu suất ổn định và cân bằng.

b) Support Vector Machine

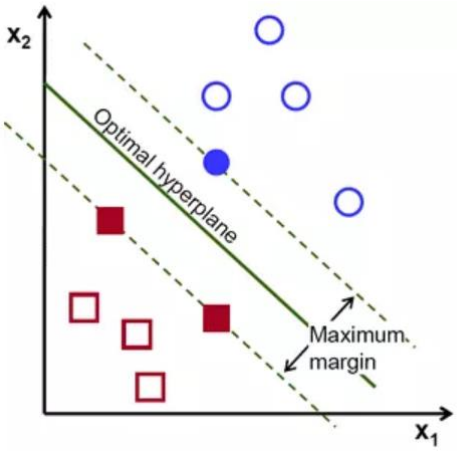
Giới thiệu về SVM

- **SVM** là một thuật toán giám sát được sử dụng chủ yếu cho việc phân loại. Trong thuật toán này, chúng ta vẽ đồ thị dữ liệu là các điểm trong n chiều (ở đây n là số lượng các tính năng) với giá trị của mỗi tính năng sẽ là một phần liên kết. Sau đó chúng ta thực hiện tìm *hyper-plane* phân chia các lớp. Mục tiêu của SVM là tìm ra hyperplane tối ưu nhất để phân tách các lớp.



- **Margin trong SVM**

Margin là khoảng cách giữa hyperplane đến 2 điểm dữ liệu gần nhất tương ứng với các phân lớp. Phương pháp SVM luôn cố gắng cực đại hóa margin này, từ đó thu được một siêu phẳng tạo khoảng cách xa nhất so với 2 quả táo và lê. Nhờ vậy, SVM có thể giảm thiểu việc phân lớp sai (misclassification) đối với điểm dữ liệu mới đưa vào



SVM trong bài toán

Tìm nghiệm cho SVM nhóm em sử dụng trực tiếp SVC trong thư viện ***sklearn***.

```
# Huấn luyện SVM
svm_classifier = SVC(kernel='linear', random_state=42)
svm_classifier.fit(X_train_tfidf, y_train)
```

```
svm_y_pred = svm_classifier.predict(X_test_tfidf)
svm_accuracy = accuracy_score(y_test, svm_y_pred)
svm_precision = precision_score(y_test, svm_y_pred, average='weighted')
svm_recall = recall_score(y_test, svm_y_pred, average='weighted')
svm_f1 = f1_score(y_test, svm_y_pred, average='weighted')

print("Kết quả SVM:")
print(f"Accuracy: {svm_accuracy}")
print(f"Precision: {svm_precision}")
print(f"Recall: {svm_recall}")
print(f"F1 Score: {svm_f1}")
print("\n--------------------\n")
```

- Huấn luyện mô hình:
  + kernel='linear': Đây là tham số xác định loại kernel (hạt nhân) sẽ sử dụng. Ở đây, nhóm em sử dụng kernel tuyến tính (linear kernel). Có nghĩa là SVM sẽ cố gắng tìm siêu phẳng phân chia các lớp trong không gian đặc trưng.
  + random_state=42: Giữ cho kết quả huấn luyện có thể tái lặp lại bằng cách cố định hạt giống ngẫu nhiên. Thông số này giúp đảm bảo rằng mỗi lần chạy mô hình với cùng một tập dữ liệu sẽ nhận được cùng một kết quả.
  + svm_classifier.fit(X_train_tfidf, y_train): Huấn luyện mô hình trên dữ liệu huấn luyện đã được chuyển đổi thành ma trận TF-IDF.
  + X_train_tfidf: Ma trận TF-IDF của dữ liệu huấn luyện.
  + y_train: Nhãn của dữ liệu huấn luyện.
- Đánh giá mô hình:
  + Dự đoán trên tập kiểm tra:
  svm_y_pred = svm_classifier.predict(X_test_tfidf): Dự đoán nhãn của dữ liệu kiểm tra sử dụng mô hình SVM đã huấn luyện.
  X_test_tfidf: Ma trận TF-IDF của dữ liệu kiểm tra.
  + Tính toán các giá trị chỉ số đánh giá:
  Accuracy: Tính tỷ lệ phần trăm các dự đoán đúng trên tổng số dự đoán.
  Precision: Tính tỷ lệ phần trăm các dự đoán đúng trong số các dự đoán được thực hiện. Ở đây, chúng ta sử dụng trung bình có trọng số để tính toán precision.
  Recall: Tính tỷ lệ phần trăm các dự đoán đúng trong số các mẫu thực sự thuộc về mỗi lớp. Chúng ta cũng sử dụng trung bình có trọng số để tính toán recall.

F1 Score: Tính trung bình điều hòa giữa precision và recall. Chúng ta sử dụng trung bình có trọng số để tính toán F1 Score

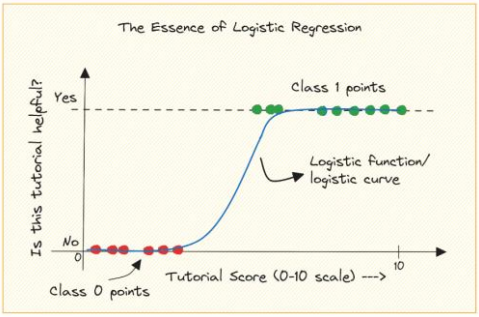
Kết quả:

```
Kết quả SVM:
Độ chính xác: 0.901884534848938
Độ chính xác (weighted): 0.9018189076707184
Độ thu hồi (weighted): 0.901884534848938
Điểm F1 (weighted): 0.9005963660948356
```

Kết luận: SVM là một phương pháp hiệu quả cho bài toán phân lớp dữ liệu. Nó là một công cụ đắc lực cho bài toán phân tích văn bản.

c) Logistic Regression

Giới thiệu về Logistic Regression:



- Logistic Regression là một trong những mô hình phân loại cơ bản trong machine learning, được sử dụng rộng rãi trong các bài toán phân loại hai lớp. Mặc dù có từ "regression" trong tên gọi, nhưng Logistic Regression thực ra là một mô hình phân loại, không phải là mô hình hồi quy. Mô hình này dự đoán xác suất một mẫu thuộc vào một lớp cụ thể dựa trên các biến đầu vào.
- Cách hoạt động của Logistic Regression sử dụng hàm logistic (hay sigmoid function) để biểu diễn xác suất thuộc vào một lớp cụ thể, được tính dựa trên tổ hợp tuyến tính của các biến đầu vào. Hàm sigmoid được biểu diễn như sau:

$$P(y = 1|x) = \frac{1}{1+e^{-z}}$$

Ưu điểm của Logistic Regression:

- Đơn giản và dễ hiểu: Logistic Regression có cách hoạt động rõ ràng và dễ hiểu, phù hợp cho những người mới bắt đầu với machine learning.

- Ít tài nguyên tính toán: Logistic Regression cần ít tài nguyên tính toán so với các mô hình phức tạp hơn như Neural Networks hay Random Forests, nên nó phù hợp cho các tập dữ liệu lớn.
- Tính linh hoạt trong việc điều chỉnh tham số: Logistic Regression cho phép điều chỉnh các tham số như hệ số điều chỉnh (regularization) để kiểm soát overfitting.
- Khả năng hiển thị tầm quan trọng của đặc trưng: Bằng cách xem xét hệ số của mỗi biến đầu vào, ta có thể đánh giá được mức độ ảnh hưởng của từng biến đến kết quả phân loại.
- Cho kết quả xác suất: Logistic Regression không chỉ dự đoán lớp của một mẫu, mà còn cung cấp một ước lượng xác suất cho dự đoán của nó.

Logistic Regression trong bài toán:

```
# Huấn luyện Logistic Regression
lr_classifier = LogisticRegression(random_state=42)
lr_classifier.fit(X_train_tfidf, y_train)
```

- Huấn luyện mô hình:
  + lr_classifier.fit(X_train_tfidf, y_train): Huấn luyện mô hình trên dữ liệu huấn luyện đã được chuyển đổi thành ma trận TF-IDF.
  + X_train_tfidf: Ma trận TF-IDF của dữ liệu huấn luyện.
  + y_train: Nhãn của dữ liệu huấn luyện.

```
# Đánh giá Logistic Regression
lr_y_pred = lr_classifier.predict(X_test_tfidf)
lr_accuracy = accuracy_score(y_test, lr_y_pred)
lr_precision = precision_score(y_test, lr_y_pred, average='weighted')
lr_recall = recall_score(y_test, lr_y_pred, average='weighted')
lr_f1 = f1_score(y_test, lr_y_pred, average='weighted')

print("Kết quả Logistic Regression:")
print(f"Accuracy: {lr_accuracy}")
print(f"Precision: {lr_precision}")
print(f"Recall: {lr_recall}")
print(f"F1 Score: {lr_f1}")
```

```
Kết quả Logistic Regression:
Độ chính xác: 0.8755608734669459
Độ chính xác (weighted): 0.8769734638837902
Độ thu hồi (weighted): 0.8755608734669459
Điểm F1 (weighted): 0.8725429330133847
```

- Đánh giá mô hình:
  + Dự đoán trên tập kiểm tra:

lr_y_pred = lr_classifier.predict(X_test_tfidf): Dự đoán nhãn của dữ liệu kiểm tra sử dụng mô hình Logistic Regression đã huấn luyện.
X_test_tfidf: Ma trận TF-IDF của dữ liệu kiểm tra.
+ Tính toán các giá trị chỉ số đánh giá:
Accuracy: Tính tỷ lệ phần trăm các dự đoán đúng trên tổng số dự đoán.
Precision: Tính tỷ lệ phần trăm các dự đoán đúng trong số các dự đoán được thực hiện. Ở đây, chúng ta sử dụng trung bình có trọng số để tính toán precision.
Recall: Tính tỷ lệ phần trăm các dự đoán đúng trong số các mẫu thực sự thuộc về mỗi lớp. Chúng ta cũng sử dụng trung bình có trọng số để tính toán recall.
F1 Score: Tính trung bình điều hòa giữa precision và recall. Chúng ta sử dụng trung bình có trọng số để tính toán F1 Score.

Kết luận: Logistic Regression là một mô hình phân loại đơn giản nhưng hiệu quả trong machine learning. Mặc dù có cấu trúc đơn giản, nhưng mô hình này vẫn mang lại kết quả tốt trên nhiều loại dữ liệu. Tuy nhiên, cũng có một số hạn chế cần được xem xét: Hạn chế của Logistic Regression:

- Giả định tuyến tính: Logistic Regression giả định mối quan hệ giữa biến đầu vào và đầu ra là tuyến tính. Điều này có thể là hạn chế khi mối quan hệ thực tế không phải lúc nào cũng tuyến tính.
- Khả năng mô hình hóa mối quan hệ phức tạp: Logistic Regression không thể mô hình hóa mối quan hệ phức tạp giữa các biến đầu vào và đầu ra như các mô hình phức tạp hơn như Neural Networks hay Random Forests.

Hướng giải quyết để cải thiện hiệu quả của mô hình Logistic Regression:

- Thử nghiệm các cấu trúc mô hình khác nhau: Khám phá và thử nghiệm các biến thể của Logistic Regression như Regularized Logistic Regression (ví dụ: Ridge hoặc Lasso Regression) để kiểm soát overfitting hoặc Polynomial Logistic Regression để mô hình hóa mối quan hệ phi tuyến.
- Sử dụng các kỹ thuật tiên tiến hơn: Sử dụng các kỹ thuật như Support Vector Machines hoặc Neural Networks để xem xét khả năng mô hình hóa mối quan hệ phức tạp hơn.
- Điều chỉnh tham số cho các thành phần của mô hình: Tìm kiếm tham số tối ưu cho mô hình Logistic Regression bằng cách sử dụng các kỹ thuật như Grid Search hoặc Random Search để cải thiện hiệu suất của mô hình.

6. Lựa chọn mô hình học sâu

Nhóm lựa chọn 3 mô hình **Long Short Term Memory (LSTM), GRU, CNN**

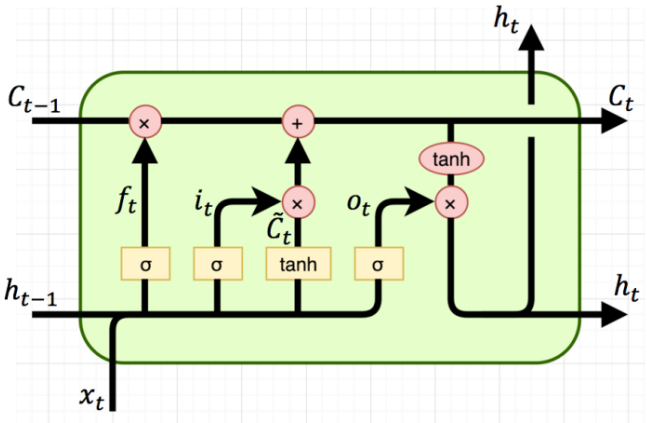
a) Long Short Term Memory:

Giới thiệu về LSTM:

- Mạng nơ-ron hồi quy dài hạn (LSTM) là một kiến trúc mạng nơ-ron nhân tạo (ANN) được thiết kế để xử lý các chuỗi dữ liệu phụ thuộc thời gian. LSTM được phát triển bởi Hochreiter và Schmidhuber vào năm 1997 để giải quyết vấn đề mất mát đạo hàm (gradient vanishing) trong các mạng nơ-ron hồi quy (RNN) truyền thống

Ưu điểm của mạng LSTM so với mạng RNN truyền thống:

- **Khả năng học tập các chuỗi dài:** LSTM có thể học tập các chuỗi dài hơn nhiều so với RNN truyền thống do khả năng lưu trữ thông tin lâu dài của nó.
- **Khả năng xử lý các chuỗi phức tạp:** LSTM có thể xử lý các chuỗi phức tạp hơn so với RNN truyền thống do khả năng học tập các mối quan hệ phụ thuộc thời gian phức tạp.
- **Khả năng chống nhiễu:** LSTM có khả năng chống nhiễu tốt hơn so với RNN truyền thống do khả năng quên đi thông tin không quan trọng.



hình ...: Mô hình LSTM

<Source: medium.com >

Mạng LSTM sử dụng trong bài toán:

```
model = tf.keras.Sequential([
    # This is how you need to set the Embedding layer when using pre-trained embeddings
    tf.keras.layers.Embedding(vocab_size+1, embedding_dim, input_length=maxlen, trainable=False),
    tf.keras.layers.Dropout(0.3),
    tf.keras.layers.Conv1D(64, 5, activation='relu'),
    tf.keras.layers.MaxPooling1D(pool_size=4),
    tf.keras.layers.LSTM(64),
    tf.keras.layers.Dense(4, activation='softmax')

])

model.compile(loss="sparse_categorical_crossentropy",
              optimizer='adam',
              metrics=['accuracy'])
```

- Lớp nhúng (Embedding Layer): Lớp này chuyển đổi các từ trong văn bản thành các vectơ biểu diễn, giúp mô hình có thể hiểu và xử lý thông tin ngữ nghĩa của văn bản.
- Lớp tích chập 1D (1D Convolutional Layer): Lớp này sử dụng các bộ lọc để trích xuất các đặc trưng cục bộ từ chuỗi dữ liệu đầu vào. Các đặc trưng này là những thông tin quan trọng giúp mô hình học được mối quan hệ giữa các từ trong văn bản.
- Lớp pooling 1D (1D Pooling Layer): Lớp này giảm kích thước của chuỗi dữ liệu đầu vào bằng cách thực hiện phép toán lấy giá trị tối đa (max pooling). Việc giảm kích thước giúp mô hình tiết kiệm tài nguyên tính toán và giảm thiểu hiện tượng quá khớp.
- Lớp kết nối đầy đủ (Fully Connected Layer): Lớp này kết nối tất cả các nơ-ron trong lớp LSTM với tất cả các nơ-ron trong lớp đầu ra.

Cách thức mô hình được huấn luyện và đánh giá:

- Optimizer (Trình tối ưu hóa): Adam: Hiệu quả cao, thường được sử dụng cho các bài toán LSTM.
- Loss (Hàm mất mát): Đo lường mức độ sai lệch giữa dự đoán của mô hình và giá trị thực tế.
- Sparse_Categorical_Crossentropy: Sử dụng cho các bài toán phân loại đa lớp
- Metrics (Chỉ số đánh giá): Đo lường hiệu suất của mô hình trên tập dữ liệu đánh giá.
- Accuracy: Tỷ lệ dự đoán chính xác

Kết luận:

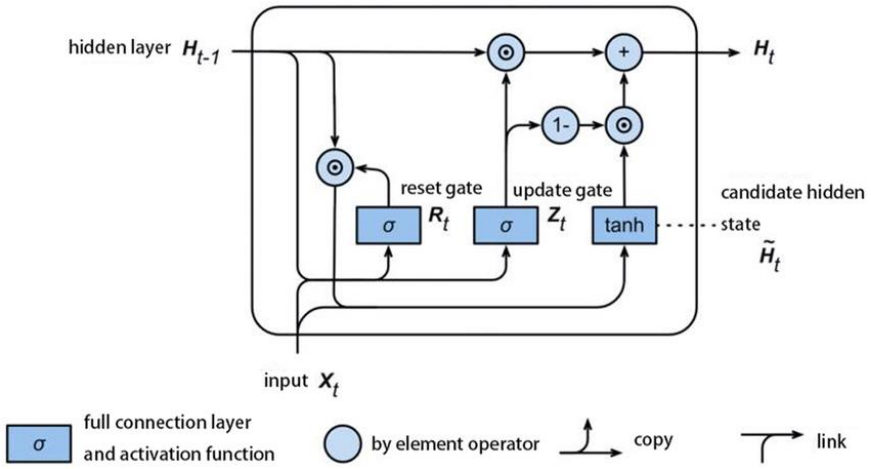
- Nhìn chung, kiến trúc mạng LSTM này có cấu trúc tương đối đơn giản nhưng hiệu quả. Tuy nhiên, mô hình này vẫn có một số hạn chế như việc lựa chọn tham số cho các thành phần của mô hình có thể ảnh hưởng đến

hiệu suất của mô hình. Hướng giải quyết tương lai để cải thiện hiệu quả của mô hình: Thử nghiệm các cấu trúc mô hình khác nhau, Sử dụng các kỹ thuật học sâu tiên tiến hơn, Điều chỉnh tham số cho các thành phần của mô hình.

b) Gated Recurrent Unit (GRU)

Giới thiệu về mạng GRU

- Mạng GRU (Gated Recurrent Unit) là một biến thể của mạng LSTM (Long Short-Term Memory) được phát triển để khắc phục một số hạn chế của LSTM, đặc biệt là vấn đề biến mất gradient trong các mạng RNN (Recurrent Neural Network) sâu. GRU có cấu trúc đơn giản hơn LSTM nhưng vẫn có khả năng học được các phụ thuộc thời gian dài trong dữ liệu.



Cách thức hoạt động của GRU:

- Cổng cập nhật (Update Gate): Xác định mức độ thông tin từ trạng thái ẩn trước đó được giữ lại trong trạng thái ẩn hiện tại.
- Cổng tái thiết lập (Reset Gate): Xác định mức độ thông tin từ trạng thái ẩn trước đó được xóa bỏ trước khi cập nhật.
- Tạo trạng thái ứng cử viên (Candidate State): Tạo một trạng thái ẩn mới dựa trên trạng thái ẩn trước đó và đầu vào hiện tại.
- Cập nhật trạng thái ẩn: Trạng thái ẩn hiện tại được tính toán bằng cách kết hợp trạng thái ẩn trước đó và trạng thái ứng cử viên theo tỷ lệ được xác định bởi cổng cập nhật.

Mạng GRU sử dụng trong bài toán:

```
    model_gru = tf.keras.Sequential([
    tf.keras.layers.Embedding(vocab_size+1, embedding_dim, input_length=maxlen),
    tf.keras.layers.Dropout(0.3),
    tf.keras.layers.Bidirectional(tf.keras.layers.GRU(32)),
    tf.keras.layers.Dense(6, activation='relu'),
    tf.keras.layers.Dense(4, activation='softmax')
])

# Set the training parameters

    model_gru.compile(loss='sparse_categorical_crossentropy',
                  optimizer='adam',
                  metrics=['accuracy'])
```

- Lớp nhúng (Embedding Layer): Lớp này chuyển đổi các từ trong văn bản thành các vectơ biểu diễn, giúp mô hình có thể hiểu và xử lý thông tin ngữ nghĩa của văn bản.
- Lớp GRU (Gated Recurrent Unit Layer): Lớp này là lớp chính của mô hình GRU, giúp mô hình học được các phụ thuộc thời gian dài trong văn bản.
- Lớp kết nối đầy đủ (Fully Connected Layer): Lớp này kết nối tất cả các nơ-ron trong lớp GRU với tất cả các nơ-ron trong lớp đầu ra.

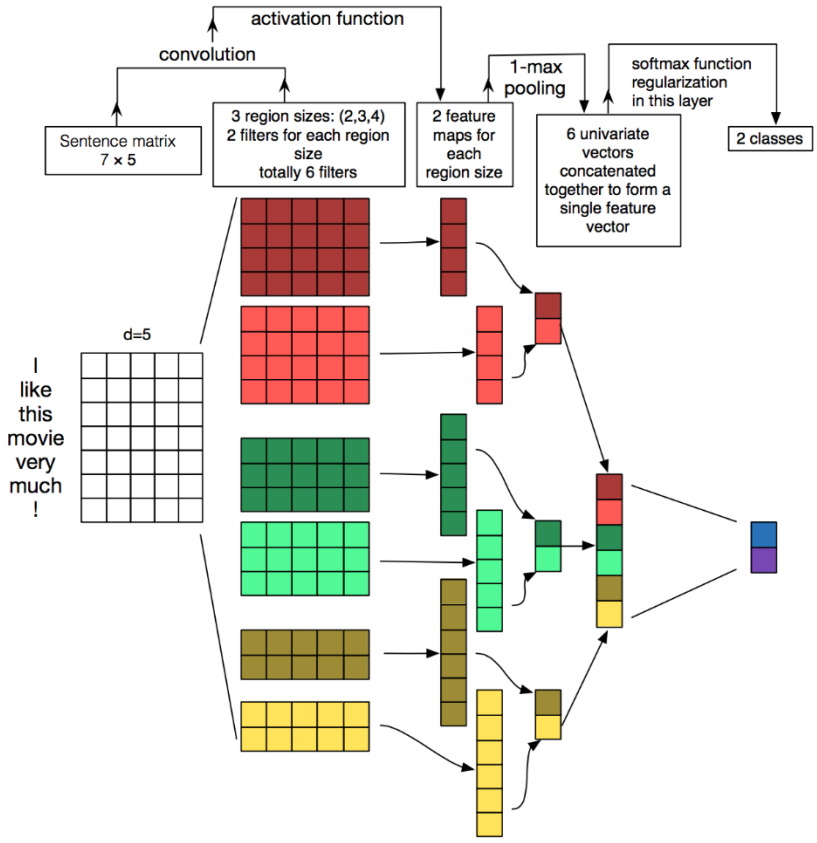
Kết luận:

Kiến trúc mạng GRU có cấu trúc tương đối đơn giản nhưng hiệu quả. Việc sử dụng vectơ nhúng được đào tạo trước giúp cải thiện hiệu quả của mô hình và giảm thời gian huấn luyện. Việc sử dụng lớp GRU giúp mô hình học được các phụ thuộc thời gian dài trong văn bản. Việc sử dụng lớp kết nối đầy đủ giúp mô hình tạo ra dự đoán cuối cùng cho bài toán.

c) Convolutional Neural Network (CNN)

Giới thiệu về mạng CNN:

- Mạng nơ-ron tích chập (Convolutional Neural Network - CNN) là một loại mạng nơ-ron nhân tạo (artificial neural network) được sử dụng phổ biến trong lĩnh vực xử lý hình ảnh và thị giác máy tính. Tuy nhiên, trong những năm gần đây, CNN cũng được ứng dụng hiệu quả trong các bài toán xử lý ngôn ngữ tự nhiên (natural language processing – NLP)



Mạng CNN sử dụng trong bài toán:

```
model = tf.keras.Sequential([
    # This is how you need to set the Embedding layer when using pre-trained embeddings
    tf.keras.layers.Embedding(vocab_size+1, embedding_dim, input_length=maxlen, trainable=False),
    tf.keras.layers.Dropout(0.3),
    tf.keras.layers.Conv1D(128, 5, activation='relu'),
    tf.keras.layers.GlobalMaxPooling1D(),
    tf.keras.layers.Dense(6, activation='relu'),
    tf.keras.layers.Dense(4, activation='softmax')

])

model.compile(loss="sparse_categorical_crossentropy",
              optimizer='adam',
              metrics=['accuracy'])
```

- Lớp nhúng (Embedding Layer): Lớp này chuyển đổi các từ trong văn bản thành các vectơ biểu diễn, giúp mô hình có thể hiểu và xử lý thông tin ngữ nghĩa của văn bản.
- Lớp tích chập 1D (1D Convolutional Layer): Lớp này sử dụng các bộ lọc để trích xuất các đặc trưng cục bộ từ chuỗi vectơ biểu diễn văn bản. Mỗi bộ lọc có kích thước kernel (kích thước cửa sổ) và số lượng kênh bằng với số kênh của ma trận đầu vào.

- Lớp pooling 1D (1D Pooling Layer): Lớp này giảm kích thước của chuỗi vectơ đầu ra bằng cách thực hiện phép toán lấy giá trị tối đa (max pooling) hoặc giá trị trung bình (average pooling) trong một cửa sổ nhất định. Việc giảm kích thước giúp mô hình tiết kiệm tài nguyên tính toán và giảm thiểu hiện tượng quá khớp.
- Lớp kết nối đầy đủ (Fully Connected Layer): Lớp này kết nối tất cả các nơ-ron trong lớp pooling cuối cùng với tất cả các nơ-ron trong lớp đầu ra.

CHƯƠNG III. Tiến hành dự đoán

1. Tiến hành dự đoán trên các mô hình

a) Mô hình học máy

```
# Chuyển đổi comment thành ma trận TF-IDF sử dụng vectorizer đã huấn luyện
new_comment_tfidf = vectorizer.transform(new_comment)

print(f"Comment: {new_comment[0]}")
# Sử dụng mô hình Random Forest đã huấn luyện để dự đoán nhãn
predicted_label = rf_classifier.predict(new_comment_tfidf)
predicted_proba = rf_classifier.predict_proba(new_comment_tfidf)
# In ra nhãn dự đoán và xác suất tương ứng
print(f"Random Forest")
print(f"Predicted Label: {predicted_label[0]}")

# Sử dụng mô hình SVM đã huấn luyện để dự đoán nhãn
predicted_label = svm_classifier.predict(new_comment_tfidf)
predicted_proba = svm_classifier.predict_proba(new_comment_tfidf)
# In ra nhãn dự đoán và xác suất tương ứng
print(f"SVM")
print(f"Predicted Label: {predicted_label[0]}")

# Sử dụng mô hình Logistic Regression đã huấn luyện để dự đoán nhãn
predicted_label = lr_classifier.predict(new_comment_tfidf)
predicted_proba = lr_classifier.predict_proba(new_comment_tfidf)
# In ra nhãn dự đoán và xác suất tương ứng
print(f"Logistic Regression")
print(f"Predicted Label: {predicted_label[0]}")
```

Sử dụng công cụ Vectorizer để chuyển đổi bình luận mới thành ma trận TF-IDF. Ma trận TF-IDF sẽ đại diện cho từ ngữ trong bình luận dưới dạng các giá trị số, phản ánh tần suất từ đó xuất hiện trong tập dữ liệu nhưng giảm trọng số của những từ quá thông dụng. Sau đó ta đưa bình luận vào mô hình để mô hình dự đoán nhãn của bình luận. Kết quả được hiển thị dưới đây

```
Comment: giao hàng chậm
Random Forest
Predicted Label: Shipping
SVM
Predicted Label: Shipping
Logistic Regression
Predicted Label: Shipping
```

```
Comment: chất lượng sản phẩm quá tệ
Random Forest
Predicted Label: Quality
SVM
Predicted Label: Shipping
Logistic Regression
Predicted Label: Quality
```

b) Mô hình học sâu

```
print("Comment: ", new_comment)
cleaned_comment = clean_data(new_comment)
print("Cleaned comment: ", cleaned_comment)
new_comment_seq = tokenizer.texts_to_sequences([cleaned_comment])
new_comment_padded = pad_sequences(new_comment_seq, maxlen=MAXLEN, padding='post')

prediction_LSTM = model_LSTM.predict(new_comment_padded)
predicted_label_LSTM = np.argmax(prediction_LSTM, axis=1)[0]
predicted_label_name = inverse_label_map[predicted_label_LSTM]
print("Nhãn dự đoán với mạng LSTM: \n", predicted_label_name)

prediction_GRU = model_GRU.predict(new_comment_padded)
predicted_label_GRU = np.argmax(prediction_GRU, axis=1)[0]
predicted_label_name = inverse_label_map[predicted_label_GRU]
print("Nhãn dự đoán với mạng GRU: \n", predicted_label_name)

prediction_CONV = model_conv.predict(new_comment_padded)
predicted_label_CONV = np.argmax(prediction_CONV, axis=1)[0]
predicted_label_name = inverse_label_map[predicted_label_CONV]
print("Nhãn dự đoán với mạng CONV: \n", predicted_label_name)
```

Sau khi chuẩn hóa bình luận như phần trên đã đề cập bằng hàm cleaned_comment, tokenizer.texts_to_sequences() sẽ chuyển đổi bình luận đã làm sạch thành dãy số dựa trên tokenizer đã được huấn luyện từ trước. pad_sequences() thực hiện padding cho dãy số để tất cả các dãy có cùng độ dài MAXLEN. Điều này cần thiết để đảm bảo mỗi đầu vào có cùng kích thước khi đưa vào mô hình. Ta đưa bình luận đã được xử lý vào mô hình và mô hình sẽ dự đoán nhãn của bình luận. Kết quả sẽ được hiển thị dưới đây:

```
Comment:  Giao hàng chậm
Cleaned comment:  giao hàng chậm
1/1 [==============================] - 0s 25ms/step
Nhãn dự đoán với mạng LSTM:
 shipping
1/1 [==============================] - 0s 31ms/step
Nhãn dự đoán với mạng GRU:
 shipping
1/1 [==============================] - 0s 21ms/step
Nhãn dự đoán với mạng CONV:
 shipping
```

```
Comment:  chất lượng sản phẩm quá tệ
Cleaned comment:  chất sản phẩm tệ
1/1 [==============================] - 0s 27ms/step
Nhãn dự đoán với mạng LSTM:
 quality
1/1 [==============================] - 0s 29ms/step
Nhãn dự đoán với mạng GRU:
 quality
1/1 [==============================] - 0s 21ms/step
Nhãn dự đoán với mạng CONV:
 quality
```

2. Đánh giá

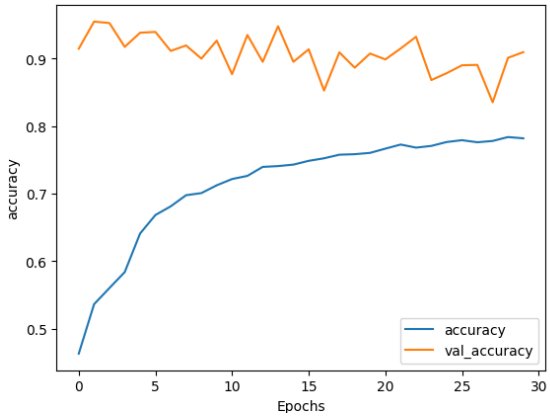
a) Mô hình học máy

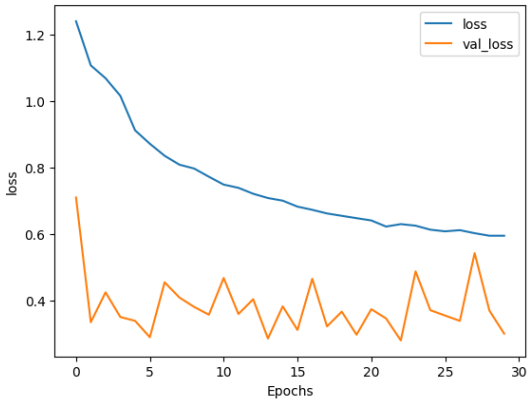
Model	Accuracy	Precision	Recall	F1 Score
Random Forest	**0.95871**	**0.95866**	**0.95871**	**0.95858**
SVM	0.90188	0.90181	0.901884	0.90059
Logistic Regression	0.87556	0.87697	0.87556	0.87254

Nhận xét: Random Forest rõ ràng là mô hình hiệu quả nhất trong ba mô hình được so sánh này. Với độ chính xác 95.87%, Precision, Recall và F1 Score đều xấp xỉ với giá trị này, Random Forest có khả năng phân loại rất tốt. Các chỉ số Precision và Recall đều cao và gần bằng nhau, cho thấy sự cân bằng tốt giữa khả năng dự đoán đúng và khả năng không bỏ sót các mẫu quan trọng.

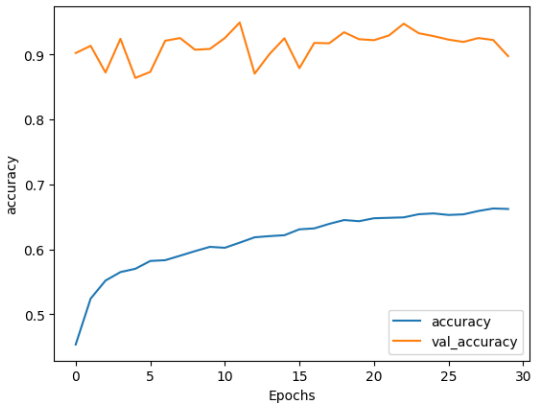
b) Mô hình học sâu

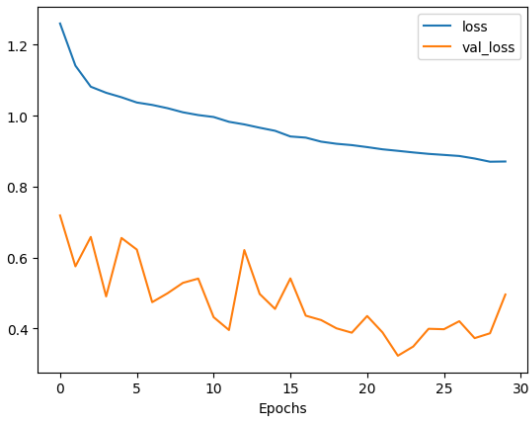
LSTM:



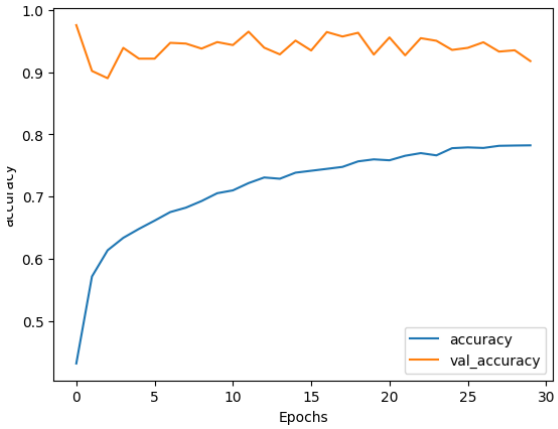

loss
val_loss
loss
Epochs

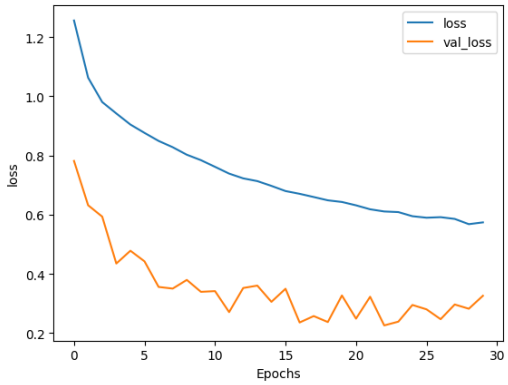
GRU:


accuracy
val_accuracy
accuracy
Epochs


loss
val_loss
Epochs

CNN:





Mô hình	Accuracy Train	Accuracy Test
LSTM	0.7819	0.9094
GRU	0.6622	0.8974
CNN	0.7824	0.9177

Nhận xét: Dựa trên bảng so sánh mô hình học máy, có thể thấy rằng mô hình CNN có hiệu suất tốt nhất với độ chính xác Train và Test cao nhất, đồng thời có giá trị Loss Train và Loss Test thấp nhất. Tuy nhiên, cả ba mô hình LSTM, GRU và CNN đều có hiệu suất tốt với độ chính xác Train và Test trên 0.89, cho thấy khả năng học tập và dự đoán hiệu quả.

KẾT LUẬN

Qua thực hiện đề tài "PHÂN LOẠI PHẢN ÁNH CỦA NGƯỜI DÙNG ĐỐI VỚI SẢN PHẨM TRÊN CÁC SÀN THƯƠNG MẠI ĐIỆN TỬ", nhóm em đã đạt được một số kết quả quan trọng:

- Nhóm em đã mô tả quá trình thu thập dữ liệu và thực hiện các bước tiền xử lý để chuẩn bị dữ liệu cho mô hình học máy.
- Các mô hình học máy như LSTM, GRU và CNN đã được thử nghiệm và so sánh hiệu suất. Kết quả cho thấy mô hình CNN đạt hiệu suất tốt nhất trong việc phân loại phản ánh của người dùng.

Với kết quả đạt được, đề tài đã góp phần vào việc ứng dụng công nghệ học máy vào phân tích dữ liệu người dùng, một lĩnh vực đang nhận được sự quan tâm ngày càng cao trong thời đại số hiện nay.

Tuy nhiên, đề tài cũng còn một số hạn chế cần được cải thiện, như mở rộng quy mô dữ liệu, thử nghiệm các mô hình học sâu phức tạp hơn và so sánh với các phương pháp phân loại khác. Những bước tiếp theo sẽ góp phần nâng cao độ chính xác và hiệu quả của hệ thống phân loại phản ánh của người dùng.

TÀI LIỆU THAM KHẢO

1 Chuẩn hóa: https://nguyenvanhieu.vn/xu-ly-tieng-viet-trong-python/