ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

Khoa Khoa Học Máy Tính

BÁO CÁO ĐỒ ÁN

PHÂN LOẠI ĐÁNH GIÁ SẢN PHẨM

BẰNG TIẾNG VIỆT

GVHD:

PGS. TS Lê Đình Duy

THS. Phạm Nguyễn Trường An

SVTH:

Nguyễn Đình Vinh 16521582

Phan Đăng Lâm 16521710

Thành phố Hồ Chí Minh – 07/2020

MỤC LỤC

LỜI NÓI ĐẦU

Đầu tiên nhóm em xin gửi lời cảm ơn đến nhà trường và bộ môn đã tạo điều kiện cho bọn em cơ hội học tập và làm việc với môn học này, luôn tạo điều kiện tốt nhất để sinh viên có thể hoàn thành tốt quá trình học tại trường nói chung và trong môn học này nói riêng.

Tiếp theo, nhóm em xin gửi lời cảm ơn chân thành tới cô Nguyễn Thị Anh Thư, giảng viên trực tiếp phụ trách giảng dạy lớp với môn Khai Thác Dữ liệu và Ứng Dụng. Cô đã tận tình hướng dẫn, chỉ bảo với những phân tích định hướng rõ ràng cho nhóm trong suốt quá trình thực hiện đề tài, là tiền đề để nhóm có thể hoàn thành đề tài đúng hạn. Cô cũng tạo điều kiện thuận lợi nhất có thể với các tài liệu cần thiết liên quan, giải đáp thắc mắc tại lớp khi các nhóm gặp khó khăn.

Và cuối cùng, xin cảm ơn tất cả các bạn trong nhóm đã cùng nhau chia sẻ công việc, hoàn thành tốt trách nhiệm của cá nhân trong suốt quá trình thực hiện với sự hướng dẫn của cô và phân công của nhóm trưởng, là yếu tố quan trọng nhất để hoàn thành tốt mục tiêu môn học. Mặc dù đã cố gắng hoàn thành đề tài với tất cả sự cố gắng nhưng tất nhiên chúng em vẫn còn mắc phải những sai sót, khuyết điểm trong đề tài, mong nhận được sự thông cảm của cô và những lời nhận xét để giúp nhóm cải thiện.

XIN CHÂN THÀNH CẢM ƠN

NHẬN XÉT CỦA GIẢNG VIÊN

.....

.....

.....

.....

.....

.....

.....

.....

Chương 1. GIỚI THIỆU CHUNG

1.1. Đặt vấn đề

- Bán hàng online là xu thế công nghệ của ngày nay, tuy nhiên khó có thể mà kiểm định được chất lượng có đảm bảo hay không. Đặc biệt với các mặt hàng đắt tiền được bày bán nhan nhản ở khắp mọi nơi trên internet, và việc lựa chọn mua ở đâu, mua hãng gì cho tốt trở thành mối quan tâm lớn cho người dùng.
- Một trong những cách để quyết định có nên mua hay không là dựa vào đánh giá từ những người đã mua trước, tuy nhiên số lượng đánh giá rất lớn, không có nhân lực để thống kê được hết. Vì thế áp dụng machine learning trong việc phân loại đánh giá của khách hàng là một việc đơn giản và hiệu quả và tiết kiệm chi phí.

1.2. Các bước xây dựng một ứng dụng ML

- Để xây dựng được một ứng dụng ML nhóm em tiến hành theo pipeline ở dưới đây:
    - B1: Thu thập dữ liệu
    - B2: Xử lý dữ liệu
    - B3: Feature Engineering
    - B4: Training
    - B5: Evaluate và tinh chỉnh các Hyperparameter
    - B6: Predict

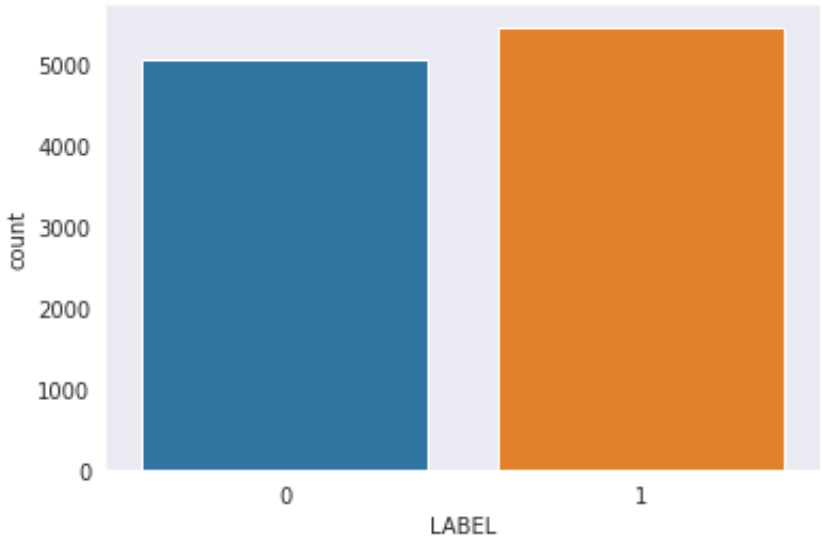
Chương 2. MÔ TẢ DỮ LIỆU

Nguồn dữ liệu được dùng để huấn luyện model được nhóm em tự crawl từ trang https://www.thegioididong.com/ bao gồm:

- 10545 đánh giá của người dùng về sản phẩm điện thoại di động.
- Các thuộc tính của dữ liệu gồm: CONTENT, STAR, LABEL trong đó nhóm em gán nhãn cho 1 sample có nhãn là 0 (negative) nếu có số sao bé hơn hoặc bằng 3, ngược lại gán nhãn là 1 (positive) nếu có số sao lớn hơn 3

	CONTENT	STAR	LABEL
0	Vừa dùng được 3 ngày. Nói chung là khá thất vọ...	3	0
1	Nói tóm lại là xiaomi dạo này dùng chán camera...	1	0
2	Giá con này hơi cao nếu giá này nên chọn s10 l...	1	0
3	toàn chip đời thấp, chip 865 k thấy bán, giá v...	2	0
4	Máy dùng ok, mình dùng được có 13 ngày máy lỗi...	3	0

- Dữ liệu giữa 2 class 0 và 1 là khá cân bằng



- Vì là đánh giá chung của mọi người, không có quy chuẩn chung về mặt ngữ pháp nên phần đánh giá còn chứa nhiều nhiễu như viết tắt, teen code, sai chính tả…

17 Dt mik mua đk 1 năm rồy.. xài rất ổn .. rất mượt.. pin trâu.. mà tự nhiên mấy ngày nay lên androi 10 thấy sao sao á.. xạc pin chậm hơn.. lướt fb hay chơi game hay bị đứng đứng màn hình.. lắc lắc kg đk mượt như trước nữa.. wifi thì bắt kém... giờ mik mún đổi phềm mềm lại ar9 đk hk ạk.. hay là có cách nào để máy xài mượt như lúc đầu hk ạk...

- Về chất lượng dữ liệu: dựa trên nội dụng và số sao mà khách hàng đánh giá, nội dung đánh giá đôi khi không phù hợp với số sao, gây giảm độ chính xác của mô hình. VD một số mẫu có content là positive nhưng có số sao là negative

Màu mới máy đẹp tuyệt vời.hy vọng giá tốt.Chip 7130 mà.có nhầm j k.hy vọng tôi nhầm.	1
sản phẩm thiết kế sang trọng lên mạng nhanh đẹp cầm pin, pin xài lâu hết,cấu hình mạnh	1

Chương 3. TIỀN XỬ LÝ DỮ LIỆU

3.1. Lower case dữ liệu

Vì các đánh giá là dữ liệu văn nói, không quan trọng chữ hoa chữ thường… nên việc chuyển chữ hoa về chữ thường sẽ quy về 1 chuẩn chung và tăng độ chính xác cho mô hình.

3.2. Loại bỏ stopwords và kí tự đặc biệt, các dấu chấm câu

Các stopwords, dấu chấm câu, kí tự đặc biệt không có ý nghĩa về mặt sentiment nhiều nhưng lại chiếm số lượng lớn trong hầu hết các đánh giá, giảm độ chính xác của mô hình nên nhóm em tiến hành loại bỏ nó.

```
Before:
mình đã mua a70 sau 1 tuần mình xin nhận xét như sau cấu hình máy khoẻ mượt đa
nhiệm tốt pin 4k5 dùng lâu hết nhận diện khuôn mặt rất nhanh có điều là vân tay
chưa được nhanh

After:
mình mua a70 1 tuần mình nhận xét cấu hình máy khoẻ mượt đa nhiệm tốt pin 4k5
dùng lâu hết nhận diện khuôn mặt rất nhanh vân tay chưa được nhanh
```

Nguồn stopwords: https://github.com/stopwords/vietnamese-stopwords

3.3. Chuẩn hoá một số sentiment word, icon

Quy chuẩn các sentiment word và icon này về 1 từ chung để tăng trọng số cũng như độ chính xác của mô hình. VD:

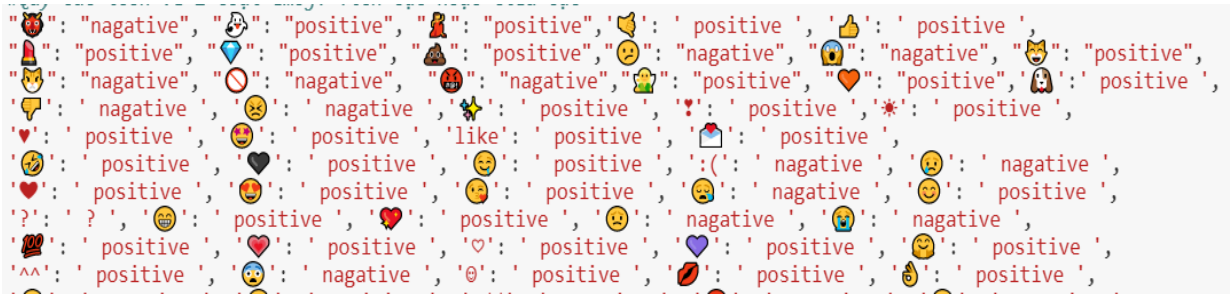
“okie”→ “ok”

“okey”→ “ok”

“kg”→ “không”

“ko”→ “không”

Chuẩn hoá các sentiment icon :



…

Cách quy chuẩn trên được tham khảo tại: https://forum.machinelearningcoban.com/t/chia-se-model-sentiment-analysis-aivivn-com-top-5/4537

3.4. Tách từ

Tách từ trong Tiếng Việt không chỉ đơn giản là tách bởi khoảng trắng vì các từ riêng biệt đi kèm với nhau sẽ mang nghĩa khác nhau. Để tách từ nhóm em sử dụng thư viện underthesea.

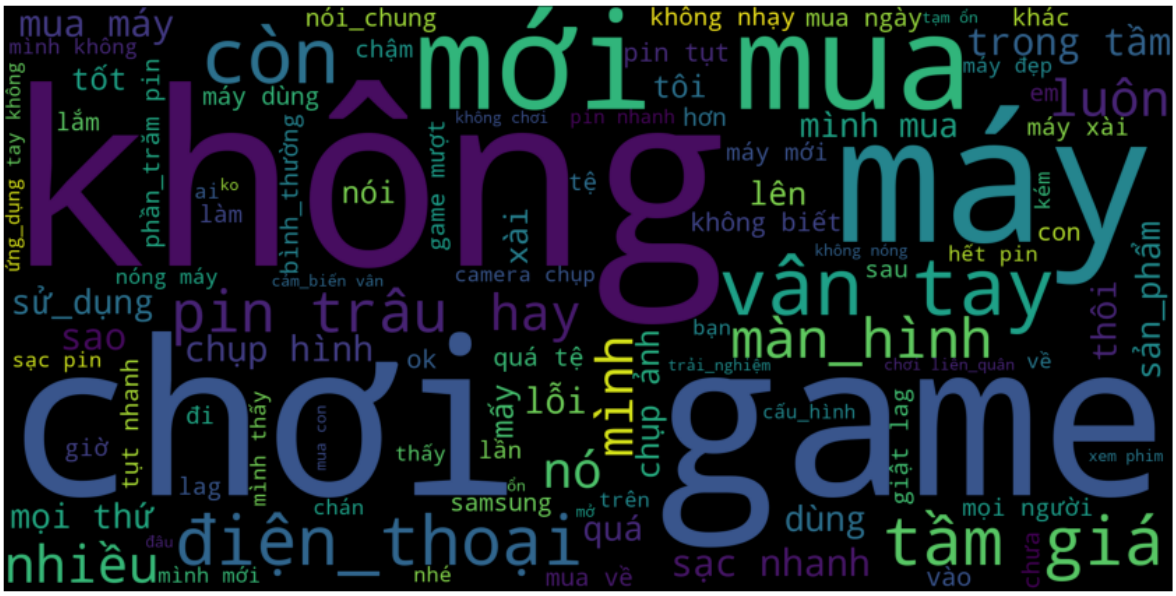
```
Before:
mình mua a70 1 tuần mình nhận xét cấu hình máy khoẻ mượt đa nhiệm tốt pin 4k5
dùng lâu hết nhận diện khuôn mặt rất nhanh vân tay chưa nhanh

After:
['mình', 'mua', 'a70', '1', 'tuần', 'mình', 'nhận_xét', 'cấu_hình', 'máy',
'khoẻ', 'mượt', 'đa_nhiệm', 'tốt', 'pin', '4k5', 'dùng', 'lâu', 'hết',
'nhận_diện', 'khuôn_mặt', 'nhanh', 'vân', 'tay', 'chưa', 'nhanh']
```

Chương 4. FEATURE ENGINEERING

4.1. Xây dựng tập từ vựng

Sau khi tách từ và chuẩn hoá xong các từ đó, ta tiến hành xây dựng tập từ vựng theo cấu trúc bag-of-word. Mỗi từ sẽ đi kèm với tần số xuất hiện của nó trong toàn bộ tập dữ liệu

Top các từ xuất hiện nhiều nhất trên bộ dataset:



Nhóm em dùng top 2000 từ xuất hiện nhiều nhất để làm feature cho mô hình

4.2. Vectorization

4.2.1. Vectorization cho mô hình LSTM

Trong bài toán xử lý ngôn ngữ (NLP) thì không thể xử lý cả câu được và người ta tách ra từng từ làm input nên chúng em áp dụng mô hình Long Short-Term Memory (LSTM) thích hợp cho bài toán dữ liệu dạng chuỗi (sequence). LSTM có khả năng nhớ các thông tin các từ được tính toán trước đó để học và predict xem đánh giá là positive hay negative

Sau bước chuẩn hoá dữ liệu chúng em thu được mỗi một cmt là một mảng các từ đã được chuẩn hoá.

Example:

```
Before vectorization
 ['i', 'like', 'song', 'i', 'like', 'you']
```

Việc tiếp theo là biến mảng các từ này thành 1 mảng vector bằng cách thay thế các từ này bằng index của chính nó trong tập từ điển

Example:

Tập từ điển:

```
1 check
2 i
3 numbr
4 video
5 song
6 like
7 please
8 subscribe
9 love
10 youtube
11 channel
12 you
13 webaddress
14 get
...
```

```
Before vectorization
 ['i', 'like', 'song', 'i', 'like', 'you']

After vectorization
 [[2], [6], [5], [2], [6], [12]]
```

Sau đó ta tiến hành reshape mảng này thành 1 chiều. Tuy nhiên model LSTM yêu cầu có chung 1 kích thước của input đầu vào mà các cmt thì có độ dài khác nhau dẫn đến các vector đặc trưng có kích thước khác nhau. Cách xử lí là tìm cmt có độ dài lớn nhất và padding 0 cho các vecto bé hơn để chúng có cùng kích thước

Example:

```
Before vectorization & padding
 ['i', 'like', 'song', 'i', 'like', 'you']

After vectorization & padding
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  2  6  5  2  6 12]
```

4.2.2. Vectorization sử dụng TF-IDF

Cách vectorization này áp dụng cho input đầu vào của mô hình linear classifier (SVM) hoặc probabilistic classifier (Naïve bayes) mà chúng em tiến hành cài đặt

TF-IDF (Term Frequency – Inverse Document Frequency) là 1 kĩ thuật sử dụng trong khai phá dữ liệu văn bản. Trọng số này được sử dụng để đánh giá tầm quan trọng của một từ trong một văn bản. Giá trị cao thể hiện độ quan trọng cao và nó phụ thuộc vào số lần từ xuất hiện trong văn bản nhưng bù lại bởi tần suất của từ đó trong tập dữ liệu.

TF: Term Frequency(Tần suất xuất hiện của từ) là số lần từ xuất hiện trong văn bản. Vì các văn bản có thể có độ dài ngắn khác nhau nên một số từ có thể xuất hiện nhiều lần trong một văn bản dài hơn là một văn bản ngắn.

$$\mathrm{tf}(t,d) = \frac{\mathrm{f}(t,d)}{\max\{\mathrm{f}(w,d) : w \in d\}}$$

Trong đó:

- tf(t, d): tần suất xuất hiện của từ t trong văn bản d
- f(t, d): Số lần xuất hiện của từ t trong văn bản d
- max({f(w, d) : w ∈ d}): Số lần xuất hiện của từ có số lần xuất hiện nhiều nhất trong văn bản d

IDF: Inverse Document Frequency(Nghịch đảo tần suất của văn bản), giúp đánh giá tầm quan trọng của một từ . Khi tính toán TF , tất cả các từ được coi như có độ quan trọng bằng nhau. Nhưng một số từ như “is”, “of” và “that” thường xuất hiện rất nhiều lần nhưng độ quan trọng là không cao. Như thế chúng ta cần giảm độ quan trọng của những từ này xuống.

$$\mathrm{idf}(t,D) = \log \frac{|D|}{|\{d \in D : t \in d\}|}$$

Trong đó:

- idf(t, D): giá trị idf của từ t trong tập văn bản
- |D|: Tổng số văn bản trong tập D
- |{d ∈ D : t ∈ d}|: thể hiện số văn bản trong tập D có chứa từ t.

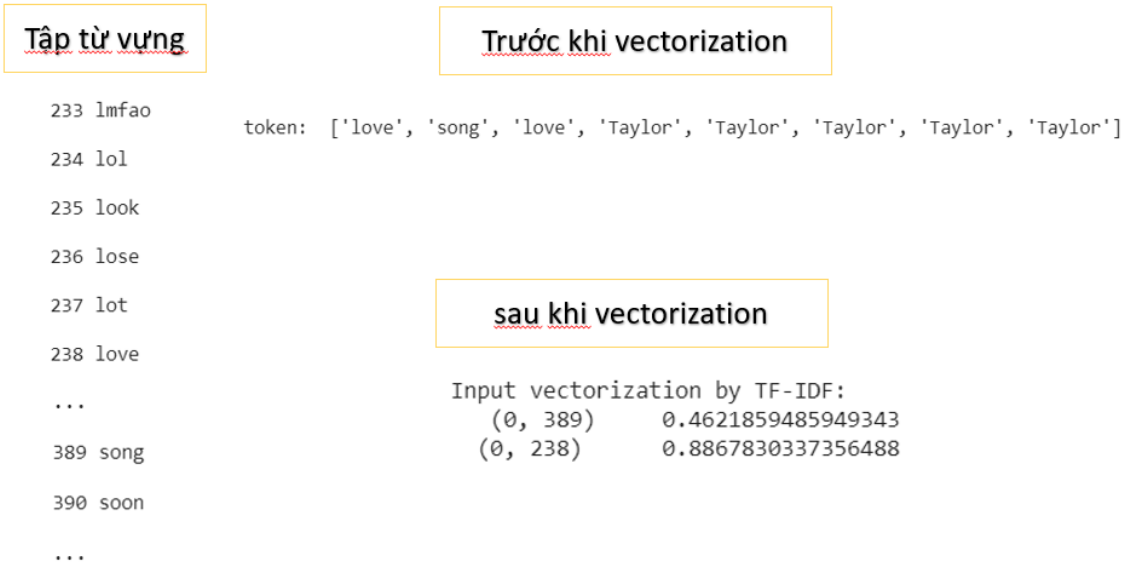
Cụ thể, chúng ta có **công thức tính tf-idf** hoàn chỉnh như sau:

tfidf(t, d, D) = tf(t, d) x idf(t, D)

Khi tính được trọng số từng term theo công thức trên ta tiến hành vectorization theo chuẩn của TF-IDF

Cần lưu ý khi sử dụng phương pháp này thì tập từ vựng sẽ được sắp sếp lại theo thứ tự alphabet chứ không phải theo tần số xuất hiện của từng từ trong toàn bộ dữ liệu

Example



Chương 5. TRAINING VÀ THỰC NGHIỆM

5.1. Training và thực nghiệm với LSTM model

Split train và test theo tỉ lệ 70:30

Add các layer cho model LSTM

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_2 (Embedding)      (None, 455, 64)           192000
_________________________________________________________________
lstm_2 (LSTM)                (None, 96)                61824
_________________________________________________________________
dense_3 (Dense)              (None, 150)               14550
_________________________________________________________________
batch_normalization_2 (Batch (None, 150)               600
_________________________________________________________________
dropout_2 (Dropout)          (None, 150)               0
_________________________________________________________________
dense_4 (Dense)              (None, 1)                 151
_________________________________________________________________
activation_2 (Activation)    (None, 1)                 0
=================================================================
Total params: 269,125
Trainable params: 268,825
Non-trainable params: 300
_________________________________________________________________
```

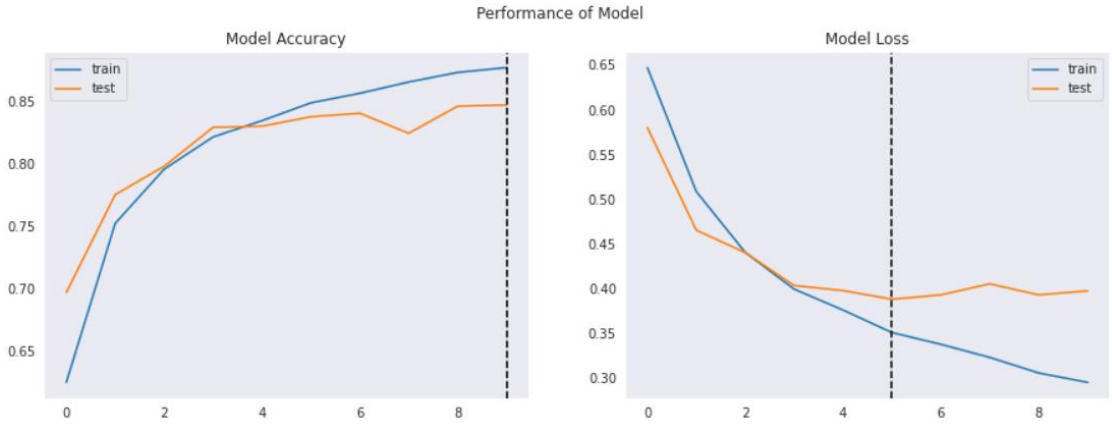
Cách build model được tham khảo từ: https://www.kaggle.com/wflazuardy/sarcasm-detection-with-keras-preprocessing

Chúng em dùng hàm optimizer là Adam với learning rate = 1e-4

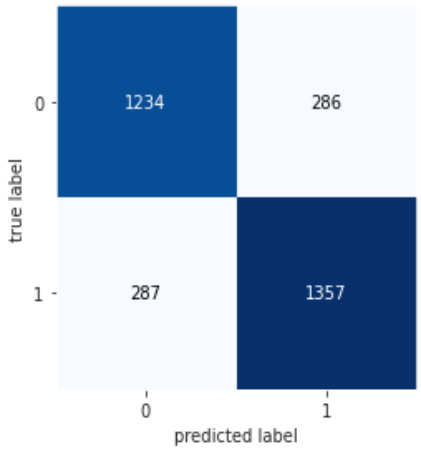
Hàm Activation là sigmoid

Training với 10 epochs

Evaluate



Confusion matrix:



Classification report:

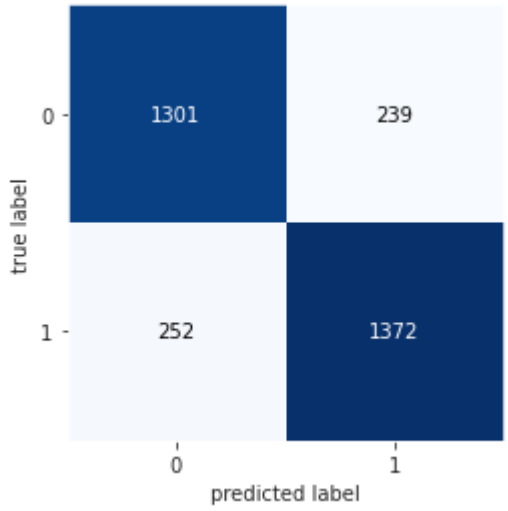
	precision	recall	f1-score	support
0	0.89	0.91	0.90	1508
1	0.92	0.90	0.91	1656
accuracy			0.91	3164
macro avg	0.91	0.91	0.91	3164
weighted avg	0.91	0.91	0.91	3164

5.2. Training và thực nghiệm với SVM clasifer

Training

```
SVM = svm.SVC(C=1.0, kernel='linear', verbose=True)
SVM.fit(Train_X_Tfidf,Train_Y)
```

Confusion maxtrix:



Classification report:

	precision	recall	f1-score	support
0	0.84	0.84	0.84	1540
1	0.85	0.84	0.85	1624
accuracy			0.84	3164
macro avg	0.84	0.84	0.84	3164
weighted avg	0.84	0.84	0.84	3164

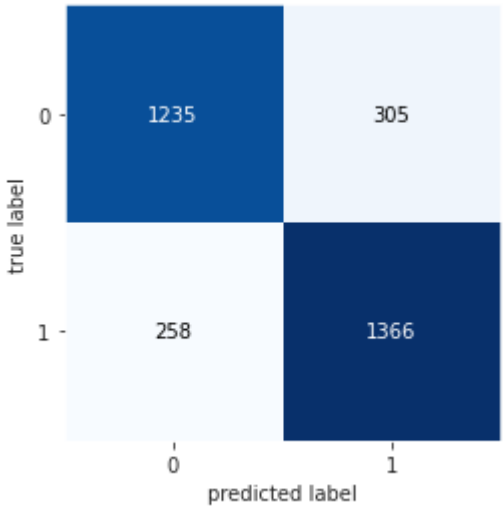
5.3. Training và thực nghiệm với Naïve Bayes clasifer

Training

```
Naive = naive_bayes.MultinomialNB()
Naive.fit(Train_X_Tfidf,Train_Y)
```

```
MultinomialNB()
```

Confusion maxtrix:

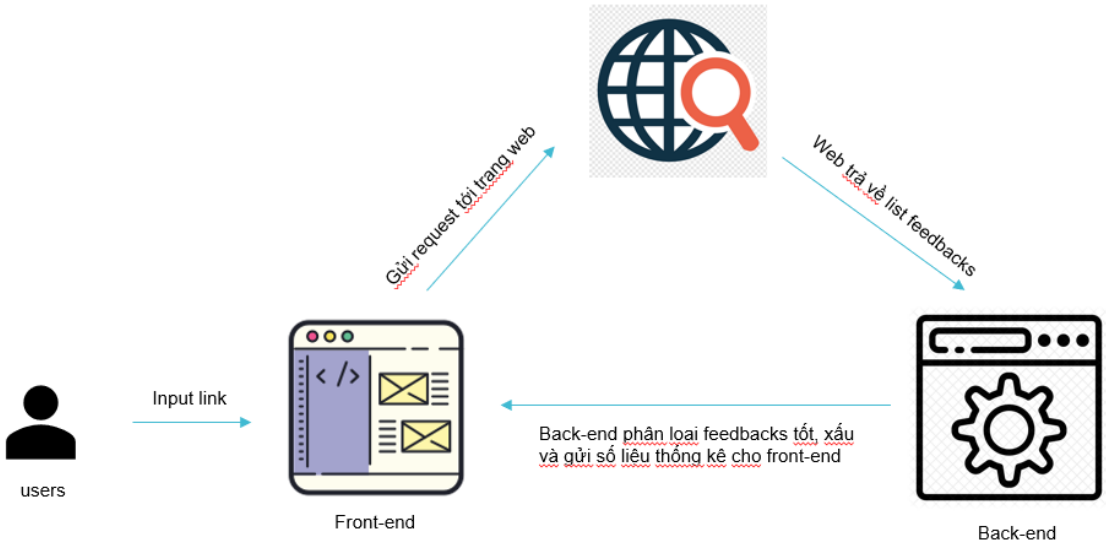


Classification report:

	precision	recall	f1-score	support
0	0.83	0.80	0.81	1540
1	0.82	0.84	0.83	1624
accuracy			0.82	3164
macro avg	0.82	0.82	0.82	3164
weighted avg	0.82	0.82	0.82	3164

Chương 6. XÂY DỰNG ỨNG DỤNG GỢI Ý MUA SẢN PHẨM

Từ các model đã train ở trên chúng em tiến hành xây dựng một web app hỗ trợ người dùng mua điện thoại di động dựa vào các đánh giá của sản phẩm đó. Nếu số lượng đánh giá positive chiếm hơn 50% toàn bộ đánh giá thì gợi ý cho người dùng mua hàng



Demo ở: https://github.com/ndvinh98/Phone-Recommendation-System

Tuy nhiên chỉ dựa vào đánh giá của người dùng là chưa đủ: Nhóm sẽ tiếp tục phát triển bằng cách kết hợp thêm các dữ liệu khác như: Lịch sử dao động giá, Giá tiền so với các trang tmđt khác, các đánh giá các trang tmđt….

Chương 7. KẾT LUẬN

Đã làm được:

- Nhóm đã tự biết cách triển khai một ứng dụng ML từ khâu chuẩn bị dữ liệu cho đến xây dựng một ứng dụng hoàn chỉnh.
- Kết quả của model chấp nhận được (Accuracy: Naïve Bayes 82%, SVM 84%, LSTM 91%)
- Hiểu rõ quá trình training và chọn các parameter phù hợp
- Thực nghiệm trên nhiều model khác nhau và thử các cách vectorization khác nhau

Khó khăn:

- Chưa xử lý được vấn đề phủ định: TF-IDF không xử lý được vấn đề phủ định trong bài toán sentiment. Ví dụ: *Cái áo này rất đẹp* và *Cái áo này chẳng đẹp* sẽ không khác nhau nhiều khi chọn feature tf-idf.
- Các đánh giá không dấu cho kết quả dự đoán thiếu chính xác
- Việc xử lý dữ liệu còn gặp nhiều khó khăn: Vì các đánh giá là dữ liệu văn nói, không có một quy chuẩn chung, nên sẽ có nhiều nhiễu: Viết tắt, teencode, sai chính tả, không dấu…. Khó có thể tổng quát hoá toàn bộ về 1 chuẩn chung được
- Dữ liệu sai nhãn: Phần nội dung đánh giá là positive nhưng lại đánh giá số sao là negative

Hướng phát triển:

- ❖ Giải quyết vấn đề phủ định bằng cách dùng từ điển tâm lý hoặc từ điển phủ định…
- ❖ Augmentation data bằng cách thêm vào các sample của chính tập train nhưng không dấu để giải quyết các bình luận không dấu.
- ❖ bổ sung vào tập train các sample mới lấy từ chính 2 từ điển positive và negative. Các từ vựng trong từ điển tích cực gán nhãn 0, các từ vựng từ từ điển tiêu cực gán nhãn 1 để tăng độ chính xác cho model
- ❖ Sử dụng Error Analysis để gán lại nhãn

Các giải pháp tăng độ chính xác mô hình tham khảo ở:
https://forum.machinelearningcoban.com/t/chia-se-model-sentiment-analysis-aivivn-com-top-5/4537

TÀI LIỆU THAM KHẢO

https://nttuan8.com/bai-13-recurrent-neural-network/

https://machinelearningcoban.com/2017/04/09/smv/

https://medium.com/@batnamv/ml-t%C3%ACm-hi%E1%BB%83u-v%E1%BB%81-naive-bayes-classification-ph%C3%A2n-lo%E1%BA%A1i-bayes-%C4%91%C6%A1n-gi%E1%BA%A3n-c4cf84e7733a

https://github.com/stopwords/vietnamese-stopwords

https://www.kaggle.com/wflazuardy/sarcasm-detection-with-keras-preprocessing

Machine Learning — Word Embedding & Sentiment Classification using Keras

Vietnamese Sentiment Analysis

Bag-of-words model

TF-IDF