BỘ GIÁO DỤC VÀ ĐÀO TẠO
ĐẠI HỌC KINH TẾ TP HỒ CHÍ MINH
TRƯỜNG CÔNG NGHỆ VÀ THIẾT KẾ

✍✍✍

ĐỒ ÁN MÔN HỌC

ĐỀ TÀI:

PHÂN TÍCH HỘI CHỨNG TRẦM CẢM DỰA TRÊN TRẠNG THÁI CHIA SẺ TRÊN TWITTER

Học phần: Xử lý ngôn ngữ tự nhiên
Nhóm Sinh Viên:

1. LÊ THỊ TUYẾT NHUNG
2. HUỲNH VĂN TRINH
3. BẠCH NGỌC MINH TRÚC
4. MAI HẠ VY

Chuyên Ngành: KHOA HỌC DỮ LIỆU
Khóa: K46

Giảng Viên: TS. Đặng Ngọc Hoàng Thành

TP. Hồ Chí Minh, Ngày 15 tháng 12 năm 2022

MỤC LỤC

CHƯƠNG 1. TỔNG QUAN

1.1. Giới thiệu bài toán "Phân tích hội chứng trầm cảm dựa trên trạng thái (status) chia sẻ trên Twitter"

Mạng xã hội không còn quá xa lạ trong thời đại công nghệ phát triển hiện nay. Nó ngày càng thu hút đông đảo người dùng bằng việc tạo tài khoản ở các trang mạng xã hội để theo dõi tin tức, mua sắm, học tập hay chia sẻ những thú vui trong cuộc sống của cá nhân, đặc biệt đối với giới trẻ. Các trang mạng như Facebook, Instagram, Zalo, YouTube, TikTok, … và trong đó có Twitter.

Ra đời vào năm 2006, Twitter là mạng xã hội rất phát triển ở các nước phương Tây, nhưng lại không được sử dụng rộng rãi như Facebook ở Việt Nam. Theo số liệu thống kê của Statcounter, Twitter mặc dù đang đứng sau Facebook về lượng người dùng, nhưng khoảng cách về số lượng người dùng giữa hai nền tảng này lại cách nhau khá xa. Twitter chiếm 14.29% người dùng trên thế giới, còn Facebook thì con số lên đến khoảng 65% người dùng.

Do nhu cầu về các trang mạng xã hội của người dùng ngày càng tăng nhưng Twitter lại không đáp ứng đầy đủ được như Facebook, như việc giới hạn ký tự đăng tải là một trong những nguyên nhân chính ảnh hưởng đến số lượng người dùng và độ phổ biến của trang mạng xã hội này. Bên cạnh đó, Twitter cũng có thế mạnh riêng cho mình khi hỗ trợ người dùng tìm kiếm hình ảnh, thông tin từ hashtag - điều này cũng trở nên phổ biến hơn khi Facebook phát hành hashtag vào năm 2013 bổ sung cho mạng xã hội của mình. Nhưng đi kèm theo đó, người dùng sử dụng hashtag 1 cách bừa bãi dễ dẫn đến nhiễu thông tin thậm chí bị đánh giá là spam.

Mặc dù vậy, Twitter được xếp vào 1 trong những trang mạng xã hội phổ biến nhất hiện nay. Qua đó, người dùng cũng thường xuyên chia sẻ đời sống riêng tư của mình trên mạng xã hội, xem đó là quyển nhật ký điện tử mà bộc lộ cả cảm xúc của mình khi trải nghiệm qua một điều gì đó trong cuộc sống. Một bài toán được đặt ra ở đây là: *liệu rằng mỗi một người dùng tài khoản Twitter, khi họ đăng tải một thông tin trạng thái trên dòng trạng thái của họ thì liệu rằng họ sẽ mang đến những thông tin tích cực hay là tiêu cực cho những người khác.*

Những phần dưới đây sẽ làm các kiểm chứng từ nhóm sẽ giúp tìm ra các nhóm làm ảnh hưởng lên tâm trạng của người sử dụng.

1.2. Lý Do Chọn Lựa Đề Tài

Twitter ngày một nhiều người có được tài khoản cá nhân, và có rất nhiều thông tin dữ liệu được truyền tải. Nhờ sự phát triển đó mà chúng ta không biết rằng Twitter là trang mạng xã hội mang đến cho người sử dụng một trải nghiệm thú vị hay là một trải

nghiệm tồi tệ, có thể một bài đăng của một người về chuyện thất tình, họ có thể bị stress nhưng tùy vào dòng trạng thái của họ có thể xem xét đến mức độ trầm cảm hay không của, hoặc là một người luôn đăng những tấm hình vui vẻ, những cảnh đẹp nhưng họ chỉ có một mình và ít người tương tác, từ đó ta cũng có thể suy xét về việc họ có đang bị trầm cảm hay không?

Có rất nhiều điều cần xem xét trên mạng xã hội Twitter mà người dùng đang sử dụng hàng ngày. Và có thể rút ra được kết luận Twitter là trang mạng đáng để chúng ta trải nghiệm nhất? Nhóm sẽ tiến hành việc kiểm chứng các điều trên thông qua các dữ liệu thu thập được từ Twitter.

CHƯƠNG 2. CÁC MÔ HÌNH PHÂN LỚP DỮ LIỆU

2.1. Các Phương Pháp Tiền Xử Lý Dữ Liệu

2.1.1. Làm sạch và loại bỏ các 'Stop words' của tiếng Anh

Quá trình chuyển đổi dữ liệu thành thứ mà máy tính có thể hiểu được gọi là tiền xử lý. Một trong những hình thức tiền xử lý chính là lọc ra những dữ liệu vô ích. Trong xử lý ngôn ngữ tự nhiên, các từ vô ích (dữ liệu), được gọi là các từ dừng (Stop words).

Vậy 'Stop words' là gì? 'Stop words' là một từ thường được sử dụng (chẳng hạn như "the", "a", "an", "in") mà công cụ tìm kiếm đã được lập trình để bỏ qua, cả khi lập chỉ mục các mục để tìm kiếm và khi truy xuất chúng như là kết quả của một truy vấn tìm kiếm.

Chúng ta không muốn những từ này chiếm dung lượng trong cơ sở dữ liệu của mình hoặc chiếm thời gian xử lý nó. Đối với điều này, chúng ta có thể loại bỏ chúng một cách dễ dàng, bằng cách lưu trữ một danh sách các từ mà ta coi như là từ dừng.

Sample text with Stop Words	Without Stop Words
GeeksforGeeks – A Computer Science Portal for Geeks	GeeksforGeeks , Computer Science, Portal ,Geeks
Can listening be exhausting?	Listening, Exhausting
I like reading, so I read	Like, Reading, read

Mô hình này loại bỏ 'Stop words' khỏi văn bản. Loại bỏ các từ dừng rất hữu ích khi người ta chỉ muốn xử lý các từ quan trọng nhất về mặt ngữ nghĩa trong văn bản và bỏ qua các từ hiếm khi liên quan về mặt ngữ nghĩa, chẳng hạn như mạo từ và giới từ. NLTK (Bộ công cụ ngôn ngữ tự nhiên) trong python có danh sách các từ dừng được lưu trữ bằng 16 ngôn ngữ khác nhau.

Tải thư viện NLTK và truy cập vào thư viện để tải hàm 'stopwords':

```
!pip install nltk
import nltk
nltk.download('stopwords')
```

Sau đó truy cập vào hàm 'stopwords' để xuất ra bộ danh sách các từ dừng mà thư viện nltk đã có:

```
stopwords_list = stopwords.words('english')
```

Và tiến hành xóa các từ dừng có trong file dữ liệu.

2.1.2. Làm sạch các ký hiệu đặc biệt trên dữ liệu

Cũng giống như 'Stop words', các ký hiệu đặc biệt có thể chiếm dung lượng của dữ liệu trong quá trình xử lý, có thể làm cho thời gian xử lý không đạt hiệu quả cao và quá trình xử lý có thể sai sót do bị nhiễu. Vì thế mà nhóm quyết định tạo các hàm để xử lý các ký hiệu đặc biệt có trong đoạn text.

- Hàm làm sạch và loại bỏ dấu câu

```
def cleaning_repeating_char(text):
    return re.sub(r'(.)\1+', r'\1', text)
```

- Hàm để dọn dẹp và xóa các ký hiệu '@'

```
def cleaning_email(data):
    return re.sub('@[^\s]+', ' ', data)
```

- Hàm làm sạch và xóa URL's

```
def cleaning_URLs(data):
    return re.sub('((www\.[^\s]+)|(https?://[^\s]+))',' ',data)
```

- Làm sạch và loại bỏ chữ số

```
def cleaning_numbers(data):
    return re.sub('[0-9]+', '', data)
```

Các ký hiệu số không mang ý nghĩa quan trọng trong một đoạn text, đặc biệt là trong phần trạng thái khi đăng trên Facebook, nên cũng cần loại bỏ.

2.1.3. Làm sạch các ký hiệu đặc biệt trên dữ liệu

NLTK có phương pháp đặc biệt này được gọi là *TweetTokenizer()* giúp mã hóa Tweet Corpus thành các mã thông báo có liên quan.

Ưu điểm của việc sử dụng *TweetTokenizer()* so với *word_tokenize* thông thường là khi xử lý các tweet, chúng ta thường bắt gặp các ký tự đặc biệt hoặc biểu tượng cảm xúc, thẻ bắt đầu bằng # cần được xử lý khác đi. Hãy xem một ví dụ để hiệu về phương thức *TweetTokenizer()*:

```
from nltk.tokenize import TweetTokenizer

tweet=u"Snow White and the Seven Degrees #MakeAMovieCold@midnight:)"
tokenizer=TweetTokenizer()
print(tokenizer.tokenize(tweet.lower()))
```

returns,

```
['snow', 'white', 'and', 'the', 'seven', 'degrees', '#makeamoviecold', '@midnight', ':)']
```

Sử dụng *word_tokenize*:

```
['snow', 'white', 'and', 'the', 'seven', 'degrees', '#makeamoviecold', '@midnight', ':)']
```

```
from nltk.tokenize import word_tokenize

s="Snow White and the Seven Degrees #MakeAMovieCold@midnight:)"
print(word_tokenize(s))
```

returns,

```
['Snow', 'White', 'and', 'the', 'Seven', 'Degrees', '#', 'MakeAMovieCold', '@', 'midnight', ':', ')']
```

Các kết quả xuất ra khác nhau, và sử dụng *TweetTokenizer* giúp đoạn text được tối ưu hết mức có thể khi tách các từ.

2.1.4. Chuyển các từ có nghĩa giống nhau mà được chia ở dạng khác trong Tiếng anh về cùng một từ (gọi là xử lý Stemming)

Stemming là quá trình tạo ra các biến thể hình thái của từ gốc. Các chương trình gốc thường được gọi là các thuật toán gốc hoặc các chương trình gốc. Thuật toán tạo gốc rút gọn các từ "chocolates", "chocolatey", và "choco" sẽ chuyển về từ gốc, "chocolate" và "retrieval", "retrieved", "retrieves" cũng sẽ được chuyển về từ gốc "retrieve".

Stem(root) là một phần của từ mà bạn thêm các phụ tố thay thế (thay đổi/từ gốc) chẳng hạn như (-ed,-ize, -s,-de,mis). Vì vậy, bắt đầu một từ hoặc câu có thể dẫn đến những từ không phải là từ thực tế. Các gốc được tạo bằng cách loại bỏ các hậu tố hoặc tiền tố được sử dụng với một từ.

Truy cập vào thư viện nltk để gọi phương thức PorterStemmer

```
from nltk import PorterStemmer
```

Tiến hành chuyển các từ về stem

```
st = nltk.PorterStemmer()
def stemming_on_text(data):
    text = [st.stem(word) for word in data]
    return data

data['text']= data['text'].apply(lambda x: stemming_on_text(x))
```

2.1.5. Làm sạch các ký hiệu đặc biệt trên dữ liệu

Lemmatization là quá trình nhóm các dạng biến cách khác nhau của một từ lại với nhau để chúng có thể được phân tích thành một mục duy nhất. Lemmatization tương tự như *stemming* nhưng nó mang lại ngữ cảnh cho các từ. Vì vậy, nó liên kết các từ có nghĩa tương tự thành một từ.

Tiền xử lý văn bản bao gồm cả stemming cũng như *Lemmatization*, nhiều khi mọi người thấy hai thuật ngữ này khó hiểu. Một số coi hai điều này là như nhau. Trên thực tế, *Lemmatization* được ưa thích hơn Stemming vì từ vựng hóa thực hiện phân tích hình thái học của các từ.

Truy cập vào thư viện nltk để gọi phương thức PorterStemmer

```
# import these modules
from nltk.stem import WordNetLemmatizer
```

Tiến hành chuyển các từ về Lemmatization

```
lm = nltk.WordNetLemmatizer()
def lemmatizer_on_text(data):
    text = [lm.lemmatize(word) for word in data]
    return data

data['text'] = data['text'].apply(lambda x: lemmatizer_on_text(x))
```

2.2. Quy Trình Phân Lớp Dữ Liệu

Mô hình phân lớp dữ liệu là quá trình phân một đối tượng dữ liệu vào một hay nhiều lớp (loại) đã cho trước nhờ một mô hình phân lớp.

Mô hình này được xây dựng dựa trên một tập dữ liệu đã được gán nhãn trước đó (thuộc về lớp nào).

Quá trình gán nhãn (thuộc lớp nào) cho đối tượng dữ liệu chính là quá trình phân lớp dữ liệu.

2.2.1. Mô Hình Phân Lớp Naive Bayes

Trong học máy, phân loại Naive Bayes là một thành viên trong nhóm các phân loại có xác suất dựa trên việc áp dụng định lý Bayes khai thác mạnh giả định độc lập giữa các hàm, hay đặc trưng.

Một phân loại Naive Bayes dựa trên ý tưởng nó là một lớp được dự đoán bằng các giá trị của đặc trưng cho các thành viên của lớp đó. Các đối tượng là một nhóm (group) trong các lớp nếu chúng có cùng các đặc trưng chung. Có thể có nhiều lớp rời rạc hoặc lớp nhị phân.

a) Định lý Bayes

Định lý Bayes được sử dụng rất rộng rãi trong các phân tích để ra quyết định. Xác suất tiên nghiệm thường được phán đoán chủ quan bởi người ra quyết định. Sau đó, các thông tin thu thập được từ việc chọn mẫu và các xác suất hậu nghiệm được tính để làm cơ sở cho việc đưa ra các quyết định tốt nhất.

Tính lại xác suất hậu nghiệm bằng định lý Bayes:

Xác suất tiên nghiệm → Thông tin mới → Áp dụng định lý Bayes → Xác suất hậu nghiệm.

Công thức:

$$P(A_i \mid B) = \frac{P(A_i)P(B \mid A_i)}{P(A_1)P(B \mid A_1) + P(A_2)P(B \mid A_2) + \ldots + P(A_n)P(B \mid A_n)}$$

Trong đó:

- P(A|B) là xác suất xảy ra của một sự kiện ngẫu nhiên A khi biết sự kiện liên quan B đã xảy ra.
- P(B|A) là xác suất xảy ra B khi biết A xảy ra
- P(A) là xác suất xảy ra của riêng A mà không quan tâm đến B.
- P(B) là xác suất xảy ra của riêng B mà không quan tâm đến A.

b) Một số kiểu mô hình Naive Bayes

- Multinomial Naive Bayes
- Bernoulli Naive Bayes
- Gaussian Naive Bayes

c) Ứng dụng: Thuật toán Naive Bayes Classification được áp dụng vào các loại ứng dụng sau

- Real time Prediction
- Multi class Prediction
- Text classification/ Spam Filtering/ Sentiment Analysis
- Recommendation System

2.2.2. Mô Hình Phân Lớp kNN

a) Định nghĩa

Trong Machine Learning, KNN(K-Nearest Neighbor) là một trong những thuật toán được sử dụng cho các bài toán Classification và Regression. Ý tưởng của thuật toán này là trong một không gian, những dữ liệu tương tự sẽ tồn tại gần nhau, từ đó tìm k điểm gần với dữ liệu cần kiểm tra nhất, tức là phương pháp chỉ xác định loại mẫu được chia theo loại của một hoặc một số mẫu gần nhất trong quyết định phân loại. Thông thường, k là 1 số nguyên không lớn hơn 20.

b) Các bước trong kNN

1. Ta có D là tập các điểm dữ liệu đã được gắn nhãn và A là dữ liệu chưa được phân loại.
2. Đo khoảng cách (Euclidian, Manhattan, Minkowski, Minkowski hoặc Trọng số) từ dữ liệu mới A đến tất cả các dữ liệu khác đã được phân loại trong D.
3. Chọn K (K là tham số mà bạn định nghĩa) khoảng cách nhỏ nhất.
4. Kiểm tra danh sách các lớp có khoảng cách ngắn nhất và đếm số lượng của mỗi lớp xuất hiện.
5. Lấy đúng lớp (lớp xuất hiện nhiều lần nhất).
6. Lớp của dữ liệu mới là lớp mà bạn đã nhận được ở bước 5.

c) Trong bài toán Classification

Label của một điểm dữ liệu mới được suy ra trực tiếp từ K điểm dữ liệu gần nhất trong training set. Label của một test data có thể được quyết định bằng major voting (bầu chọn theo số phiếu) giữa các điểm gần nhất, hoặc nó có thể được suy ra bằng cách đánh trọng số khác nhau cho mỗi trong các điểm gần nhất đó rồi suy ra label.

d) Trong bài toán Regression

Đầu ra của một điểm dữ liệu sẽ bằng chính đầu ra của điểm dữ liệu đã biết gần nhất (trong trường hợp K=1), hoặc là trung bình có trọng số của đầu ra của những điểm gần nhất, hoặc bằng một mối quan hệ dựa trên khoảng cách tới các điểm gần nhất đó.

2.2.3. Đánh Giá Tính Hiệu Quả

a) Mô hình phân lớp Naive Bayes

Ưu điểm:

- Dễ sử dụng và nhanh khi cần đoán nhãn của dữ liệu test. Thực hiện khá tốt trong multi class prediction (test later).
- Khi giả định rằng các feature của dữ liệu là độc lập với nhau thì Naive Bayes chạy tốt hơn so với các thuật toán khác như logistic regression và cũng cần ít dữ liệu hơn.

Nhược điểm:

- Độ chính xác của Naive Bayes nếu so với các thuật toán khác thì không được cao.
- Trong thế giới thực, hầu như bất khả thi khi các feature của dữ liệu test là độc lập với nhau, nhưng điều này khó xảy ra trong thực tế làm giảm chất lượng của mô hình.

Ứng dụng:

- Được sử dụng rộng rãi trong lĩnh vực máy học và nhiều lĩnh vực khác như trong các công cụ tìm kiếm, các bộ lọc mail. Mục đích chính là làm sao tính được xác suất Pr(Cj, d'), xác suất để tài liệu d' nằm trong lớp Cj.
- Được áp dụng vào các loại ứng dụng sau:
    - Real time Prediction
    - Multi class Prediction
    - Text classification/ Spam Filtering/ Sentiment Analysis
    - Recommendation System

b) Mô hình phân lớp kNN

Ưu điểm

- Thuật toán đơn giản, dễ dàng triển khai.
- Độ phức tạp tính toán nhỏ.
- Xử lý tốt với tập dữ liệu nhiễu

Nhược điểm:

- Với K nhỏ dễ gặp nhiễu dẫn tới kết quả đưa ra không chính xác
- Cần nhiều thời gian để thực hiện do phải tính toán khoảng cách với tất cả các đối tượng trong tập dữ liệu.
- Cần chuyển đổi kiểu dữ liệu thành các yếu tố định tính.
- Ngoài ra, việc lựa chọn k có thể sẽ ảnh hưởng đến độ chính xác của thuật toán. Với tập dữ liệu training đủ lớn và có thể đưa ra số k lớn và hợp lý độ chính xác sẽ tăng lên rất nhiều.

Ứng dụng:

- Được ứng dụng nhiều trong ngành đầu tư, bao gồm dự đoán phá sản, dự đoán giá cổ phiếu, phân bổ xếp hạng tín dụng trái phiếu doanh nghiệp, tạo ra chỉ số vốn và trái phiếu tùy chỉnh.

CHƯƠNG 3. CÁC KẾT QUẢ THỰC NGHIỆM

3.1. Bộ Dữ Liệu

3.1.1 Tên bộ dữ liệu

EDA: Sentiment Analysis Using

3.1.2 Nguồn dữ liệu

https://www.kaggle.com/kazanova/sentiment140

3.1.3 Giới thiệu bộ dữ liệu

Với sự phát triển của công nghệ thông tin và các trang mạng xã hội, ngày càng nhiều người sử dụng những trang web đó để bộc lộ cảm xúc của họ. Thông qua đó, các doanh nghiệp cũng dựa trên kết quả phân tích dựa trên những dữ liệu này để đánh giá mức độ hài lòng của khách hàng. Tweet là từ để chỉ những người dùng Twitter sử dụng mạng xã hội để đăng suy nghĩ, cảm xúc, tin nhắn bản thân trên hồ sơ của họ, có giới hạn là 140 ký tự. Đối với NLP, việc phân tích tình cảm sẽ sử dụng những kỹ thuật như phân loại tweet ra thành các lớp positive và negative, làm sạch và xóa các Stopword của tiếng anh, làm sạch và loại bỏ dấu chấm câu ở bước tiền xử lý dữ liệu.

Sau đó, xây dựng mô hình dựa trên NLTK, sử dụng bộ dữ liệu tình cảm 140 và chia dữ liệu đó thành 80% cho training và 20% cho testing. Sau khi đào tạo về mô hình, chúng ta sẽ đánh giá mô hình để đánh giá hiệu suất của mô hình được đào tạo.

Việc đánh giá mô hình dựa trên các chỉ số Accuracy (Độ chính xác), Confusion matrix with plot (Ma trận nhầm lẫn), ROC Curve.

a) Thông tin về bộ dữ liệu

- Số cột: 6 cột
- Số dòng: 1599999

Tên cột	Giá trị	Kiểu dữ liệu	Giải thích
Label	0, 4	int64	target (0 là negative, 2 là neutral, 4 là positive)
Time	Nhỏ nhất: 1467810369 Lớn nhất: 2329205794	int64	Thời gian đăng tweet
Date	Nhỏ nhất: Wed May 27 07:27:38 PDT 2009 Lớn nhất: Fri Apr 17 20:30:31 PDT 2009	object	Ngày đăng tweet

Query	NO_QUERY	object	Nếu không có, hiển thị NO_QUERY
Username		object	Tên người dùng
Text		object	Nội dung của Tweet

b) Mục tiêu

Để đưa ra một trình phân tích tình cảm để xác định tình cảm trong tweet (positive, negative) từ đó phân tích mức độ trầm cảm dựa trên những tweet đó.

3.2. Phân chia dữ liệu

3.2.1 Chuẩn bị các tính năng đầu vào cho các model

- Chúng ta chuyển đổi các từ văn bản thành dạng mảng.
- Tối đa 500 tính năng/từ được chọn để máy học. 500 từ này sẽ được chọn dựa trên tầm quan trọng sẽ phân biệt giữa các tweet tích cực và các tweet tiêu cực.

```
X=data.text
y=data.label
```

```
max_len = 500
tok = Tokenizer(num_words=2000)
tok.fit_on_texts(X)
sequences = tok.texts_to_sequences(X)
sequences_matrix = pad_sequences(sequences,maxlen=max_len)
```

Tạo thành một ma trận đã được chuyển thành các số như sau:

```
sequences_matrix
```

```
array([[   0,    0,    0, ...,  138,  297,  100],
       [   0,    0,    0, ...,   75,  200,  240],
       [   0,    0,    0, ...,  114, 1789, 1081],
       ...,
       [   0,    0,    0, ...,  359,   51,  394],
       [   0,    0,    0, ...,  121,   56,  408],
       [   0,    0,    0, ...,  315,  169,  794]], dtype=int32)
```

Vậy chúng ta có thể thấy rằng có tổng số 40000 tweet và số từ/tính năng là 500.

3.2.2 Tách dữ liệu 80% cho dữ liệu huấn luyện và 20% cho dữ liệu thử nghiệm

Bước 1: Đầu vào cho mô hình là 500 từ vì đây là số tính năng/từ mà chúng ta đã trích xuất ở trên từ văn bản của các tweet.

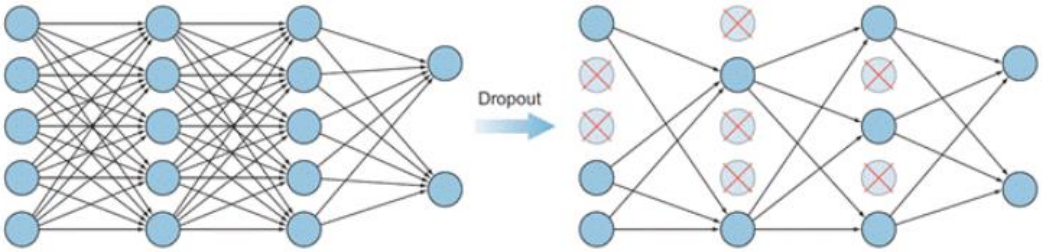
Bước 2: Các phần cung cấp cách trình bày các từ và nghĩa tương đối của chúng. Giống như trong trường hợp này, chúng ta đang cung cấp giới hạn của các từ tối đa, độ dài của các từ đầu vào và đầu vào của lớp trước.

Bước 3: LSTM (trí nhớ dài hạn) lưu các từ và dự đoán các từ tiếp theo dựa trên các từ trước đó. LSTM là một công cụ dự đoán trình tự của các từ tiếp theo.

Bước 4: Lớp dày đặc giảm đầu ra bằng cách lấy đầu vào từ lớp Làm phẳng. Lớp dày đặc sử dụng tất cả các đầu vào của các nơ-ron lớp trước đó và thực hiện các phép tính và gửi 256 đầu ra.

Bước 5: Hàm kích hoạt là nút được đặt ở cuối tất cả các lớp của mô hình mạng nơ-ron hoặc ở giữa các lớp mạng nơ-ron. Chức năng kích hoạt giúp quyết định nơ-ron nào sẽ được thông qua và nơ-ron nào sẽ kích hoạt. Vì vậy, chức năng kích hoạt của nút xác định đầu ra của nút đó với một đầu vào hoặc tập hợp các đầu vào.

Bước 6: Lớp Droupout loại bỏ một số nơ-ron từ các lớp trước đó. tại sao chúng tôi áp dụng điều này? Chúng ta áp dụng điều này để tránh các vấn đề overfitting. Trong trang bị quá mức, mô hình cho độ chính xác tốt về thời gian đào tạo nhưng không tốt về thời gian thử nghiệm.



Vì chúng ta đã chuẩn bị tất cả các tweet, bây giờ chúng tôi đang tách/tách các tweet thành dữ liệu đào tạo và dữ liệu thử nghiệm.

- 80% tweet sẽ được sử dụng trong khóa đào tạo
- 20% tweet sẽ được sử dụng để kiểm tra hiệu suất của mô hình.

```
X_train, X_test, Y_train, Y_test = train_test_split(sequences_matrix, y, test_size=0.2, random_state=2)
```

3.3. Huấn luyện dữ liệu

3.3.1 Naive Bayes

- Sau khi chia bộ dữ liệu thành 2 bộ con, lấy bộ dữ liệu huấn luyện (training) để đưa vào mô hình, tiến hành máy học.

```
from sklearn.naive_bayes import GaussianNB
model_NB = GaussianNB()
model_NB.fit(X_train, Y_train)
```

```
GaussianNB()
```

- Đưa bộ dữ liệu huấn luyện vào mô hình xong, nhóm tiếp tục đánh giá Độ chính xác khi phân lớp thành 2: trạng thái tích cực và trạng thái tiêu cực

```
from nltk.tag.api import accuracy
yHat = model_NB.predict(X_test)
print(f'Độ chính xác = {accuracy_score(Y_test,yHat)}')
```

- Cuối cùng là đánh giá mô hình bằng phương pháp ROC

```
fpr, tpr, thresholds = roc_curve(Y_test, yHat)
roc_auc = auc(fpr, tpr)
plt.figure()
plt.plot(fpr, tpr, color='darkorange', lw=1, label='ROC curve (area = %0.2f)' % roc_auc)
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC CURVE')
plt.legend(loc="lower right")
plt.show()
```

3.3.2 K-NN

- Sau khi chạy model Naive Bayes nhóm thấy chỉ số accuracy vẫn chưa được cao, nên tiếp tục lấy bộ dữ liệu huấn luyện (training) để đưa vào mô hình k-NN, tiến hành máy học.

```
k = 7
model_kNN = KNeighborsClassifier(n_neighbors=k)
model_kNN.fit(X_train, Y_train)

KNeighborsClassifier(n_neighbors=7)
```

```
KNeighborsClassifier(n_neighbors=7)
```

```
from sklearn.ensemble import RandomForestClassifier
text_classifier = RandomForestClassifier(n_estimators=200, random_state=0)
text_classifier.fit(X_train, Y_train)
```

```
RandomForestClassifier(n_estimators=200, random_state=0)
```

```
prediction = text_classifier.predict(X_test)
prediction
```

```
array([0, 0, 1, ..., 1, 0, 0])
```

- Tính tiếp tục các thông số cần cho để đánh giá mô hình

```
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score

print(confusion_matrix(Y_test,prediction))
print(classification_report(Y_test,prediction))
print(accuracy_score(Y_test,prediction))
```

- Vẽ ROC để đánh giá mô hình

```
fpr, tpr, thresholds = roc_curve(Y_test, yHat)
roc_auc = auc(fpr, tpr)
plt.figure()
plt.plot(fpr, tpr, color='darkorange', lw=1, label='ROC curve (area = %0.2f)' % roc_auc)
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC CURVE')
plt.legend(loc="lower right")
plt.show()
```

3.4. Các kết quả

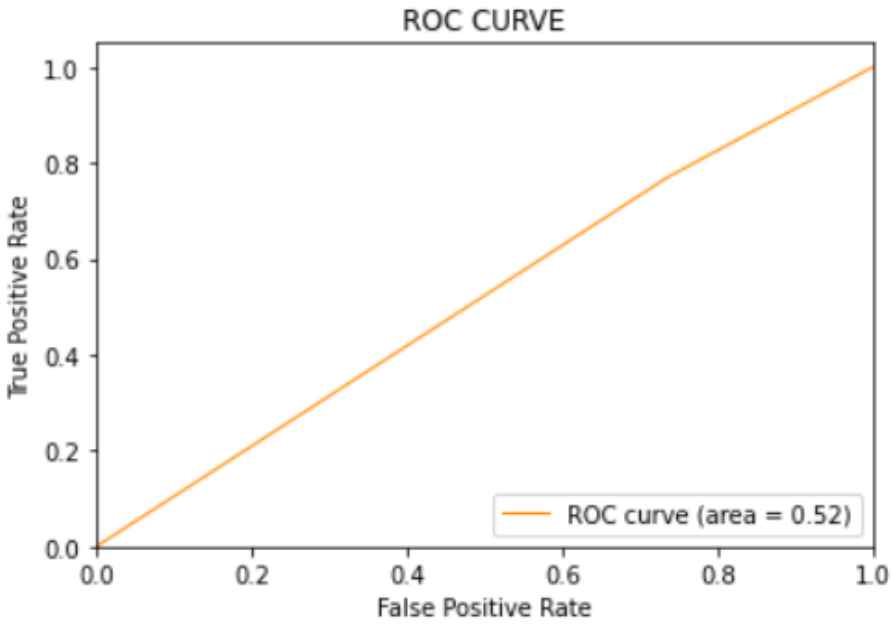
3.4.1 Naive Bayes

- Kết quả của quá trình phân lớp trên bằng phương pháp Naive Bayes bằng 52%, kết quả này vẫn chưa phù hợp khi phân lớp dữ liệu

```
from nltk.tag.api import accuracy
yHat = model_NB.predict(X_test)
print(f'Độ chính xác = {accuracy_score(Y_test,yHat)}')
```

```
Độ chính xác = 0.51725
```

- Kết quả của quá trình đánh giá mô hình



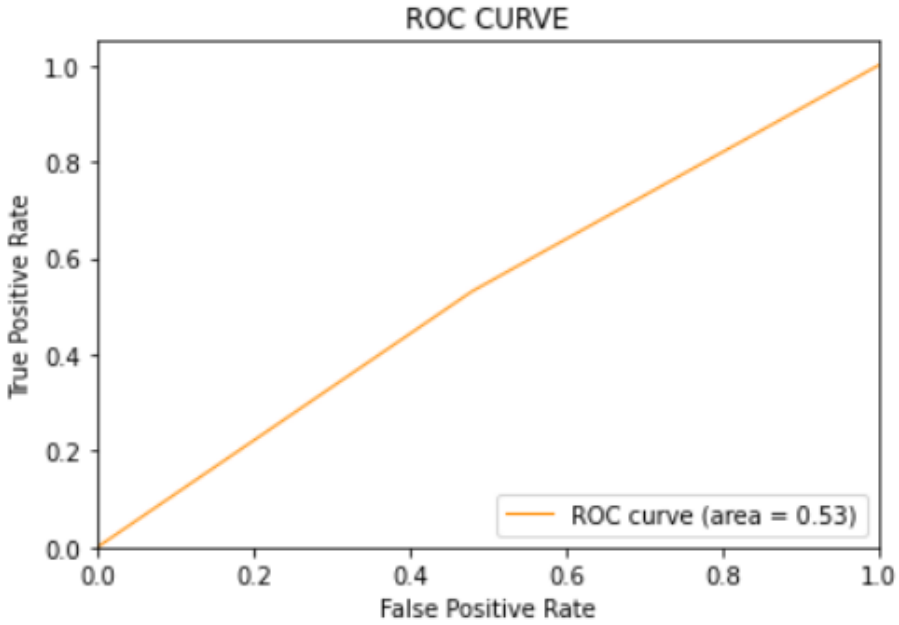
3.4.2 K-NN

- Kết quả của quá trình phân lớp trên bằng pp k-NN bằng 53%, kết quả này vẫn chưa phù hợp khi phân lớp dữ liệu.

```
yHat = model_kNN.predict(X_test)
print(f'Độ chính xác = {accuracy_score(Y_test, yHat) * 100:.2f}%')
```

```
Độ chính xác = 52.56%
```

- Kết quả của quá trình đánh giá mô hình



- Tính tỷ số accuracy

```
[[2475 1523]
 [1819 2183]]
              precision    recall  f1-score   support

           0       0.58      0.62      0.60      3998
           1       0.59      0.55      0.57      4002

    accuracy                           0.58      8000
   macro avg       0.58      0.58      0.58      8000
weighted avg       0.58      0.58      0.58      8000

0.58225
```

3.5. Phân tích và đánh giá

Từ các kết quả mà nhóm làm ở trên, cho thấy tỷ lệ chính xác **Accuracy** (Accuracy là tỉ lệ giữa số điểm được phân loại đúng và tổng số điểm) ở hai mô hình đều rất thấp (xấp xỉ 50%). Và vì đây là bộ dữ liệu có sẵn các nhãn (label) dùng để phân lớp các triệu chứng trầm cảm khi đăng trạng thái trên Twitter. Và trong thông tin bộ dữ liệu có giới thiệu về cột label gồm 3 giá trị (0 là negative, 2 là neutral, 4 là positive), nhưng trong dữ liệu nhóm chỉ thấy có 2 giá trị 0 và 4, như hình bên dưới:

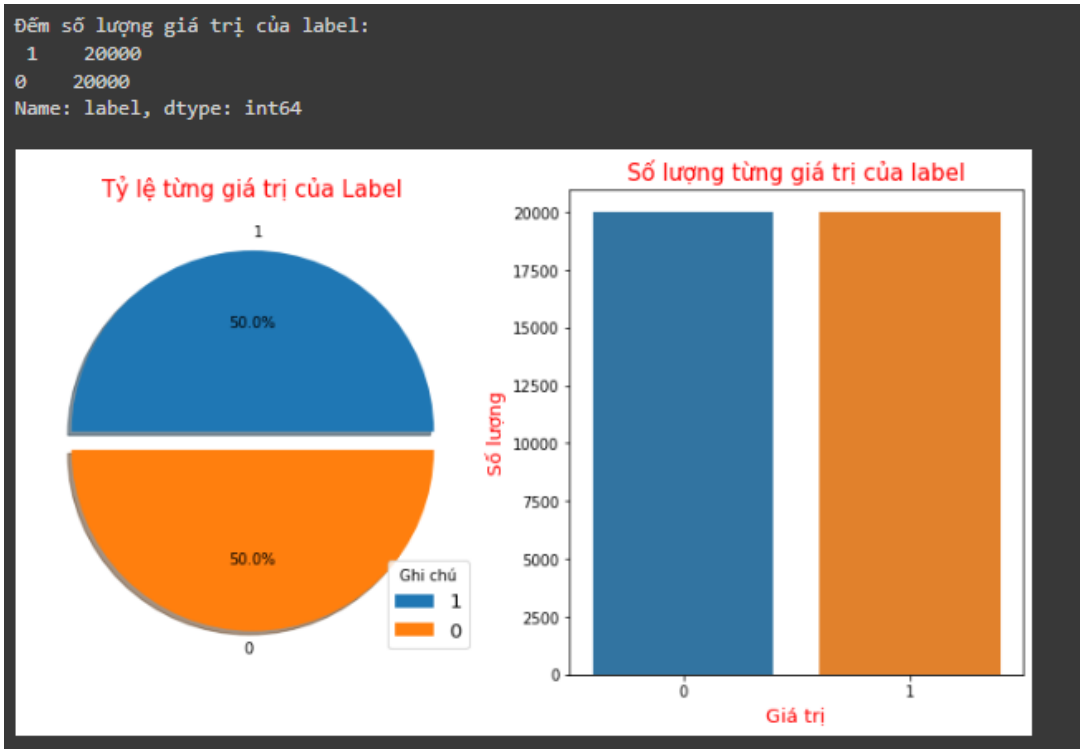
```
data['label'].unique()

array([0, 4])
```

Nên hai mô hình trên dùng để đánh giá sự chính xác của các nhãn đã được phân từ trước. Vậy các kết quả của hai mô hình cho thấy các nhãn được phân chỉ đúng có xấp xỉ 50% so với toàn bộ dữ liệu. Như vậy các label chưa phù hợp với dữ liệu cần thực hiện lại quá trình phân lớp để được kết quả chính xác.

CHƯƠNG 4. TỔNG KẾT

4.1. Các Kết Quả Đạt Được



Đây là hình ảnh trực quan về tỉ lệ cũng như số lượng của các giá trị label. Trong biểu đồ cho nhóm thấy:

- Chỉ có ⅔ label tồn tại (0 là negative, 1 là positive) so với 3 giá trị ban đầu mà bộ dữ liệu mong muốn hướng tới (0 là negative, 2 là neutral, 4 là positive).
- Trọng số của của 2 giá trị trên là xấp xỉ 50:50. Chúng có độ tương đồng khá giống nhau.

Tóm lại, nhóm nhận thấy rằng, 2 giá trị (0 là negative, 1 là positive) có mức ý nghĩa là ngang nhau hoặc thậm chí là không thể đánh giá, kết luận lẫn nhau. Vì vậy không thể suy đoán, dự báo mức độ tin cậy của biểu đồ cũng như là không thể kết luận rằng: liệu ***"Trạng thái của người dùng trên Twitter có thực sự phản ánh đến mức độ trầm cảm thực tế của họ"*** hay không.

4.2. Những tồn tại và thiếu sót

- Hạn chế khách quan: do bộ data này bị gắn nhãn thiếu.
- Hạn chế chủ quan:
  - Nhóm chưa liên kết Pipeline của sklearn với NLTK được.
  - Nhóm chưa chuẩn hóa dữ liệu để cho ra được chỉ số cao hơn.
  - Nhóm sẽ nghiên cứu thêm và hoàn thiện mong thầy góp ý.

TÀI LIỆU THAM KHẢO

1. Mã nguồn: *https://github.com/maihavy/NLP*

2. Lý thuyết về NAIVE BAYES và ứng dụng phân loại tài liệu tiếng việt trong thư viện số

NGHIÊN CỨU LÝ THUYẾT NAIVE BAYES VÀ ỨNG DỤNG PHÂN LOẠI TÀI LIỆU TIẾNG VIỆT TRONG THƯ VIỆN SỐ

3. Tài liệu môn học Khoa học dữ liệu

4. Tổng quan về thuật toán phân lớp Naive Bayes Classification

Tổng quan về thuật toán phân lớp Naive Bayes Classification (NBC) - HocTrucTuyen123.NET

5. Mô hình phân lớp Naive Bayes

Mô hình phân lớp Naive Bayes

6. K-nearest neighbors

Bài 6: K-nearest neighbors

Giới thiệu về thuận toán K láng giềng (K nearest neighbor) trong machine learning

KNN (K-Nearest Neighbors) #1