**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**Viện Công nghệ thông tin và truyền thông**



**Báo cáo**

**Project I**

**Đề tài: Phân loại tin tức**

**bằng thuật toán Naive Bayes Classifier**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Nhóm thực hiện:   |  |  |  | | --- | --- | --- | | 1. | Nguyễn Hữu Doanh | 20160631 | | 2. | Đinh Hoàng Anh | page1image296027755220160054 |   Giảng viên hướng dẫn:  **PGS. TS Nguyễn Kim Anh** |

*Hà Nội, 5/2019*

# Lời mở đầu

Phân loại văn bản là một vấn đề quan trọng trong lĩnh vực xử lý ngôn ngữ. Nhiệm vụ của bài toán này là gán các tài liệu văn bản vào nhóm các chủ đề cho trước. Đây là một bài toán rất thường gặp trong thực tế điển hình như : một nhà chuyên phân tích thị thường chứng khoán, anh ta cần phải tổng hợp rất nhiều tài liệu, bài viết về thị trường chứng khoán để đọc và đưa ra phán đoán của mình. Tuy nhiên, anh ta không thể đọc tất cả các bài viết, bài báo hay các tài liệu để rồi phân loại chúng đâu là tài liệu chứng khoán sau đó anh ta mới đọc kỹ chúng cho mục đích của anh ta. Lý do của vấn đề này là bởi ví số lượng bào viết, bài báo hiện nay rất nhiều, đặc biệt là trên internet, nếu để đọc hết được tất cả tài liệu đó thì sẽ mất rất nhiều thời gian. Một ví dụ khác trong thực tế là việc phân loại spam mail. Khi một mail được gửi đến hộp thư, nếu để người dùng phải đọc tất cả các mail thì sẽ tốn rất nhiều thời gian vì spam mail rất nhiều. Vì vậy, cần có một hệ thống phân loại đâu là spam mail và đâu là mail tốt.

Để giải bài toán này đã có rất nhiều phương pháp được đưa ra như : thuật toán Naïve Bayes, K-NN (K-Nearest-Neighbor), Cây quyết định (Decision Tree), Mạng Neuron nhân tạo (Artificial Neural Network) và SVM (Support Vector Machine). Mỗi phương pháp đều có ưu nhược điểm riêng cho bài toán này, tuy nhiên trong môn học này chúng em đã lựa chọn và sử dụng phương pháp phân loại văn bản bằng thuật toán Bayes để thực hiện bài toán.

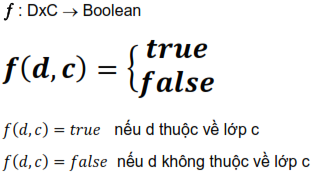
# Giới thiệu

## Giới thiệu đề tài

Ngày nay, sự bùng nổ thông tin do bị tác động bởi sự xuất hiện của các siêu phương tiện và World Wide Web (WWW) đã làm cho không gian dữ liệu gia tăng thường xuyên, điều này tạo ra một thách thức cho các hệ thống truy vấn thông tin sao cho có hiệu quả. Một trong những khó khăn mà các hệ thống thông tin thường phải gặp đó là tần suất cập nhật của các thông tin quá lớn .Phương thức sử dụng giấy trong giao dịch đang dần được số hóa, do nhiều tính năng vượt trội mà phương thức này mang lại, như là có thể lưu trữ lâu dài, cập nhật, sửa đổi, tìm kiếm một cách nhanh chóng.

Do đó số lượng văn bản số hóa ngày nay đang tăng dần theo cấp số nhân, cùng với sự gia tăng của số lượng văn bản, nhu cầu tìm kiếm văn bản cũng tăng theo, khi đó phân loại văn bản tự động là một yêu cầu cấp thiết được đặt ra. Phân loại văn bản giúp sẽ giúp chúng ta tìm kiếm thông tin một cách nhanh chóng hơn thay vì phải tìm lần lượt trong từng văn bản, hơn nữa khi mà số lượng văn bản đang gia tăng một cách nhanh chóng thì thao tác tìm lần lượt trong từng văn bản sẽ mất rất nhiều thời gian, công sức và là một công việc nhàm chán và không khả thi. Chính vì thế nhu cầu phân loại văn bản tự động là thực sự cần thiết.

Bài toán phân loại văn bản có thể được phát biểu như sau : Cho trước một tập văn bản D={d1,d2,…,dn} và tập chủ đề được định nghĩa C={c1,c2,…,cn}. Nhiệm vụ của bài toán là gán lớp di thuộc về cj đã được định nghĩa. Hay nói cách khác, mục tiêu của bài toán là đi tìm hàm f :



## Mục tiêu, phạm vi của đề tài

Vì là môn học project1 (mục tiêu chủ yếu tập trung vào phần lập trình và hiểu rõ code, chưa đặt nặng việc đi sâu vào sử dụng và cái đặt nhiều thuật toán cho một chương trình lớn) nên trong bài toán này chúng em chỉ sử dụng duy nhất thuật toán Naive Bayes và cả thư viện cho việc cài đặt thuật toán, phân loại văn bản.

**Mục đích**:

1. Tìm hiểu thuật toán Navie Bayes và phạm vi ứng dụng của nó trong phân loại văn bản.
2. Cố gắng hiểu rõ từng câu lệnh mà thư viện sử dụng trong việc tiền xử lí dữ liệu và phân loại văn bản.
3. Xây dựng một chương trình có khả năng phân loại văn bản sau khi tìm hiểu thuật toán.
4. Rút ra kinh nghiệm áp dụng cho các đề tài về học máy sau này.

**Phạm vi nghiên cứu**: chương trình thực hiện trong một quy mô nghiên cứu nhỏ với một số lượng văn bản không nhiều vào khoảng 1780 tài liệu.

# Phân loại bài báo

## Tập dữ liệu

Bộ dữ liệu gồm có nguồn gốc từ BBC news, cung cấp để làm tiêu chuẩn cho Machine Learning, dưới dạng chưa xử lý. Bao gồm 1780 tài liệu từ trang web tin tức BBC tương ứng với các câu chuyện trong 5 lĩnh vực chủ đề từ 2004-2005.

Bộ dữ liệu gồm 2 thuộc tính: news và type

5 nhãn của thuộc tính type:

* Business (kinh doanh)
* Entertainment (giải trí)
* Politics (chính trị)
* Sport (thể thao)
* Tech (công nghệ)

## Các thuật toán có thể sử dụng để xử lý phân loại

### Naïve Bayes

#### Giới thiệu định lý Naive Bayes

Định luật Bayes được phát biểu như sau:

Định lý Bayes cho phép tính xác suất xảy ra của một sự kiện ngẫu nhiên A khi biết sự kiện liên quan B đã xảy ra. Xác suất này được ký hiệu là P(A|B), và đọc là “xác suất của A nếu có B”. Đại lượng này được gọi xác suất có điều kiện hay xác suất hậu nghiệm vì nó được rút ra từ giá trị được cho của B hoặc phụ thuộc vào giá trị đó.

Theo định lí Bayes, xác suất xảy ra A khi biết B sẽ phụ thuộc vào 3 yếu tố:

Xác suất xảy ra A của riêng nó, không quan tâm đến B. Kí hiệu là P(A) và đọc là xác suất của A. Đây được gọi là xác suất biên duyên hay xác suất tiên nghiệm, nó là “tiên nghiệm” theo nghĩa rằng nó không quan tâm đến bất kỳ thông tin nào về B.

Xác suất xảy ra B của riêng nó, không quan tâm đến A. Kí hiệu là P(B) và đọc là “xác suất của B”. Đại lượng này còn gọi là hằng số chuẩn hóa (normalising constant), vì nó luôn giống nhau, không phụ thuộc vào sự kiện A đang muốn biết.

Xác suất xảy ra B khi biết A xảy ra. Kí hiệu là P(B|A) và đọc là “xác suất của B nếu có A”. Đại lượng này gọi là khả năng (likelihood) xảy ra B khi biết A đã xảy ra. Chú ý không nhầm lẫn giữa khả năng xảy ra B khi biết A và xác suất xảy ra A khi biết B.

Công thức của định luật Bayes được phát biểu như sau:

#### Ứng dụng của định luật

Định lý Bayes sẽ giúp ta tính ra xác suất xảy ra của một giả thuyết bằng cách thu thập các bằng chứng nhất quán hoặc không nhất quán với một giả thuyết nào đó. Khi các bằng chứng tích lũy, mức độ tin tưởng vào một giả thuyết thay đổi. Khi có đủ bằng chứng, mức độ tin tưởng này thường trở nên rất cao hoặc rất thấp, tức là xác xuất sảy ra giả thuyết sẽ thay đổi thì các bằng chứng liên quan đến nó thay đổi.

Chúng ta có thể ứng dụng Naive Bayes Classification để tính tỷ lệ xác suất với rất nhiều các dạng bài toán khác nhau, với dữ liệu càng nhiều thì độ chính xác của thuật toán sẽ càng cao, và khi dữ liệu thay đổi thì kết quả cũng thay đổi theo.

Thuật toán Naive Bayes Classification được áp dụng vào các loại ứng dụng sau: Real time Prediction, Multi class Prediction, Recommendation System, Text classification/ Spam Filtering/ Sentiment Analysis,...

Naive Bayes Classifiers (NBC) thường được sử dụng trong các bài toán Text Classification.

NBC có thời gian training và test rất nhanh. Điều này có được là do giả sử về tính độc lập giữa các thành phần, nếu biết class.

Nếu giả sử về tính độc lập được thoả mãn (dựa vào bản chất của dữ liệu), NBC được cho là cho kết quả tốt hơn so với SVM và logistic regression khi có ít dữ liệu training.

NBC có thể hoạt động với các feature vector mà một phần là liên tục (sử dụng Gaussian Naive Bayes), phần còn lại ở dạng rời rạc (sử dụng Multinomial hoặc Bernoulli).

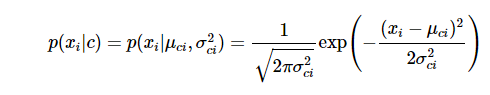
#### Áp dụng vào bái toán news classification

NBC cũng rất thích hợp cho các hệ thống phân loại văn bản hay ngôn ngữ tự nhiên vì tính chính xác của nó lớn hơn các thuật toán khác. Ngoài ra các hệ thống chống thư rác cũng rất ưu chuộng thuật toán này. Và các hệ thống phân tích tâm lý thị trường cũng áp dụng NBC để tiến hành phân tích tâm lý người dùng ưu chuộng hay không ưu chuộng các loại sản phẩm nào từ việc phân tích các thói quen và hành động của khách hàng.

**Các phân phối thường dùng:**

Gaussian Naïve Bayes:

Mô hình này được sử dụng chủ yếu trong loại dữ liệu mà các thành phần là các biến liên tục.

Với mỗi chiều dữ liệu và một class  ,  tuân theo một phân phối chuẩn có kỳ vọng và phương sai :

Multinomial Naive Bayes

Mô hình này chủ yếu được sử dụng trong phân loại văn bản mà feature vectors được tính bằng Bags of Words. Lúc này, mỗi văn bản được biểu diễn bởi một vector có độ dài d chính là số từ trong từ điển. Giá trị của thành phần thứ i trong mỗi vector chính là số lần từ thứ i xuất hiện trong văn bản đó.

Khi đó,  tỉ lệ với tần suất từ thứ  (hay feature thứ cho trường hợp tổng quát) xuất hiện trong các văn bản của class . Giá trị này có thể được tính bằng cách:

Trong đó:

* là tổng số lần từ thứ  xuất hiện trong các văn bản của class , nó được tính là tổng của tất cả các thành phần thứ của các feature vectors ứng với class .
* là tổng số từ (kể cả lặp) xuất hiện trong class . Nói cách khác, nó bằng tổng độ dài của toàn bộ các văn bản thuộc vào class . Có thể suy ra rằng , từ đó .

Bernoulli Naive Bayes

Mô hình này được áp dụng cho các loại dữ liệu mà mỗi thành phần là một giá trị binary - bẳng 0 hoặc 1. Ví dụ: cũng với loại văn bản nhưng thay vì đếm tổng số lần xuất hiện của 1 từ trong văn bản, ta chỉ cần quan tâm từ đó có xuất hiện hay không.

Khi đó được tính bằng :

Với có thể hiểu là xác suất từ thứ xuất hiện trong văn bản của class

### Support Vector machines (SVM)

Support Vector Machine là bài toán đi tìm mặt phân cách sao cho margin tìm được là lớn nhất, đồng nghĩa với việc các điểm dữ liệu an toàn nhất so với mặt phân cách.

### Random Forest Classifier

Như chúng ta biết, Random Forest là một tập hợp mô hình (ensemble). Mô hình Random Forest rất hiệu quả cho các bài toán phân loại vì nó huy động cùng lúc hàng trăm mô hình nhỏ hơn bên trong với quy luật khác nhau để đưa ra quyết định cuối cùng. Mỗi mô hình con có thể mạnh yếu khác nhau, nhưng theo nguyên tắc “wisdom of the crowd” , ta sẽ có cơ hội phân loại chính xác hơn so với khi sử dụng bất kì một mô hình đơn lẻ nào.

## Thư viện sử dụng

### re

re.sub (pattern, repl, string, count=0, flags=0): Trả về string có được bằng cách thay thế bằng cách tìm các pattern xuất hiện trong tham số string.Nếu không tìm thấy pattern, string trả về không thay đổi, tham số repl có thể là một string hoặc là một hàm.

Tham số count là số lần tối đa pattern được thay thế, count phải là một số không âm, nếu tham số này bị bỏ qua hoặc bằng 0, tất cả các pattern sẽ được thay thế

### os

* os.listdir(path): Trả về một list chứa tên của tất cả các file có trong thứ mục được cho bởi path. Thứ tự tùy ý. Không chưa cá file đặc biệt như “.” Hoặc “..”. Tham số truyền vào dưới dạng string là đường dẫn tới thư mục
* os.isfile(path): Giá trị trả về là True nếu đường dẫn path là tồn tại một file.
* os.join(path, \*paths): Có thể tham giá một hoặc nhiều path. Giá trị trả về là một string là sự kết hợp (nối) tham số path với các tham số \*path

### Numpy

* Numpy.array (object, dtype=None, copy=True, order='K', subok=False, ndmin=0): Hàm này có chức năng tạo một mảng.

object: Có thể là một mảng, bất kì đối tượng nào có dạng mảng, hoặc bất kì phương thức trả về một mảng, hoặc các mảng lồng nhau.

dtype = None: (datatype) kiểu dữ liệu mong muốn cho mảng, nếu không được truyền vào

### Pandas

* read\_csv(filepath\_or\_buffer,sep=',',delimiter=None,header='infer', names=None,index\_col=None,usecols=None,squeeze=False,prefix=None,mangle\_dupe\_cols=True,dtype=None,engine=None,converters=None,true\_values=None,false\_values=None,skipinitialspace=False,skiprows=None,skipfooter=0, nrows=None, na\_values=None,keep\_default\_na=True,na\_filter=True,verbose=False,skip\_blank\_lines=True,parse\_dates=False,infer\_datetime\_format=False,keep\_date\_col=False,date\_parser=None,dayfirst=False,iterator=False,chunksize=None, compression='infer',thousands=None,decimal=b'.',lineterminator=None, quotechar='"',quoting=0,doublequote=True,escapechar=None,comment=None, encoding=None,dialect=None,tupleize\_cols=None,error\_bad\_lines=True, warn\_bad\_lines=True,delim\_whitespace=False,low\_memory=True, memory\_map=False, float\_precision=None): đọc một file .csv (comma-separated values) thành DataFrame (Data frame là một tập dữ liệu, trong đó mỗi cột là một biến, mỗi dòng là một quan sát)

filepath\_or\_buffer: được truyền dưới dạng string, là đường dẫn tới file cần đọc

sep = “, ”: Dấu phân cách được dùng trong file .csv, mặc định là dấu phẩy (“,”).

delimiter : tên bí danh cho sep,truyền vào dưới dạng None, mặc định là None.

Header = “infer”: số thứ tự hàng dùng để làm tên cho cột, và là vị trí bắt đầu cho data. Tham số mặc định là infer, tức dòng hàng đầu tiên của cột dữ liệu sẽ làm header.

### Textblob

TextBlob là một thư viện Python (cả 2 và 3) để xử lí dữ liệu văn bản. Nó cung cấp một API đơn giản đê đi sâu vào các tác vụ xử lý ngôn ngữ tự nhiên (NLP) phổ biến như gắn thẻ một phần lời nói, trích xuất cụm danh từ, phân loại, dịch thuật,...

Textblob.words: chia Textblob thành các từ hoặc câu

Word.lematize(): hàm thực hiện việc bổ sung, khô phục các từ gốc

### Sklearn

#### Model\_selection

* sklearn.feature\_extraction.text.TfidfVectorizer(input=’content’, encoding=’utf-8’,decode\_error=’strict’,strip\_accents=None, lowercase=True, preprocessor=None,tokenizer=None,analyzer=’word’,stop\_words=None,token\_pattern=’(?u)\b\w\w+\b’,ngram\_range=(1,1),max\_df=1.0,min\_df=1,max\_features=None,vocabulary=None,binary=False,dtype=<class‘numpy.float64’>,norm=’l2’,use\_idf=True,smooth\_idf=True,sublinear\_tf=False): Chuyển một tập dữ liệu thô thành một ma trận TF-IDF đặc trưng.

input = “content”: Kiểu truyền vào là string, có thể nhận 3 loại giá trị là filename, file, và content. Giá trị mặc định là content

encoding = “utf-8”: Kiểu truyền vào là string, giá trị mặc định là utf-8, tham số này dùng để mã hóa input.

lowercase = True: Kiểu truyền vào là boolean, giá trị mặc định là True. Chuyển toàn bộ ký tự thành chữ thường.

stop\_word: Tham số truyền vào là string, list, hoặc None. Giá trị mặc định là None. Nếu là một chuỗi, nó sẽ kiểm tra \_check\_stop\_list và trả về danh sách stopword thích hợp. “English” là ngôn ngữ duy nhất được hỗ trợ. Nếu được truyền vào một list, danh sách đó là danh sách chứa các stopword, và tất cả các từ trong list sẽ được xóa ở kết quả. Nếu truyền vào giá trị None, không có stop\_word nào được loại bỏ.

Tokenizer:

Min\_df: Giá trị truyền vào có thể là kiểu int hoặc kiêt float (trong khoảng 0.0 đến 1.0). Giá trị mặc định là 1. Khi xây dựng từ điển thì tham số này sẽ loại bỏ hết các từ có tần số thấp hơn ngưỡng cho phép.

Max\_df: Giá trị truyền vào có thể là kiểu int hoặc kiêt float (trong khoảng 0.0 đến 1.0). Giá trị mặc định là 1. Khi xây dựng từ điển thì tham số này sẽ loại bỏ hết các từ có tần số cao hơn ngưỡng cho phép (cụ thể là các stopword).

Lowercase = True:

Norm = “l2”: Có thể nhận giá trị “l1”, “l2” hoặc None. Mặc định là l2.

User\_idf = True: Cho phép đặt trọng số idf.

Smooth\_idf=True:Trọng số làm mượt idf bằng cách thêm 1 đơn vị vào tần số document, được coi như là một document có chứa mọi từ trong tập. Tránh trường hợp chia cho 0. Giá trị mặc định là Trụe.

* sklearn.model\_selection.train\_test\_split(\*arrays, \*\*options): Tách arrays hoặc ma trận thành các tập con train và test một cách ngẫu nhiên.

\*arrays: Cho phép truyền vào lists, numpy arrays, ma trận thưa hoặc một DataFrame

test\_size: có thể truyền vào dưới dạng float, int hoặc None, mặc định là None. Nếu tham số truyền vào là float (có giá trị trong khoảng 0.0 tới 1.0) đại diện cho tỉ lệ của tập test. Nếu tham số truyền vào là int sẽ là số lượng tập test được chia. Nếu là None, số lượng tập test sẽ là phần bù của tập train, nếu train\_size cũng mang giá trị None, có sẽ được đặt mặc định là 0.25.

train\_size: có thể truyền vào dưới dạng float, int hoặc None, mặc định là None. Nếu tham số truyền vào là float (có giá trị trong khoảng 0.0 tới 1.0) đại diện cho tỉ lệ của tập train. Nếu tham số truyền vào là int sẽ là số lượng tập train được chia. Nếu là None, số lượng tập train sẽ là phần bù của tập test, nếu test\_size cũng mang giá trị None, có sẽ được đặt mặc định là 0.25.

random\_state: tham số truyền vào có thể dưới dạng int, RandomState Instane hoặc None, giá trị mặc định của tham số là None.Nếu tham số được truyền vào là int, cách phân chia sẽ là như nhau ở mọi lần chương trình được chạy. Nếu là None, ngược lại, mọi tập sẽ được chia một cách ngẫu nhiên.

#### Naive\_bayes

sklearn.naive\_bayes.**MultinomialNB**(*alpha=1.0*,*fit\_prior=True*,*class\_prior=None*)

Trong đó:

alpha = 1.0: giá trị laplace Smoothing được thêm vào

class\_prior = None : Xác suất tiên nhiệm của lớp, giá trị None là các lớp đã được điều chỉnh theo giá trị tiên nhiệm

fit\_prior = True: Có học lớp xác suất trước hay không. True ở đây là có học.

#### Metrics

sklearn.metrics.**accuracy\_score**(*y\_true*, *y\_pred*, *normalize=True*, *sample\_weight=None*)

Hàm này tính toán độ chính xác của tập hợp con (bộ nhãn được dự đoán cho một mẫu phải khớp chính xác với bộ nhãn tương ứng cho y\_true) trả về tỉ lệ mẫu được phân loại chính xác.

Trong đó:

Y\_true: là các giá trị (mảng một chiều, mảng nhãn, hoặc ma trận) nhãn thật

Y\_pred: là các giá trị (mảng một chiều, mảng nhãn, hoặc ma trận) nhãn dự đoán

Normalize (bool, default=True)

## Phân loại bài báo

### Tiền xử lý dữ liệu

Dữ liệu ban đầu được chia thành 5 folder có tên là nhãn của các thuộc tính: Business, entertainment, politics, sport, tech. Mỗi folder chứa các bài báo dưới dạng text. Chúng ta tiến hành chuyển toàn bộ dữ liệu file .txt từ các folder vào file .csv.

Sau khi đã chuyển được dữ liệu sang file .csv. Chúng ta thực hiện tiến hành bước đầu tiên của giai đoạn tiền xử lý dữ liệu là làm sạch dữ liệu, bước này sẽ thực hiện những công việc như sau:

* Loại các ký tự như ‘s, ‘ve, ‘t ,…
* Loại bỏ các dấu câu
* Loại bỏ các khoảng trắng ở đầu và cuối dòng
* Tất cả các từ cũng đã được đưa về dạng ký tự thường (không phải HOA)
* Những từ có cùng ‘gốc’ được đưa về cùng loại. Ví dụ, ‘include’, ‘includes’, ‘included’ đều được đưa chung về ‘include’
* Loại bỏ các ký tự tab và ký tự xuống dòng

Công việc trên sẽ được thực hiện nhờ hàm dưới đây:



Trong đó: Strip

Hàm sử dụng regular expression để thực hiện làm sạch dữ liệu: loại bỏ các ký tự có đuôi ‘s, ‘ve, ‘t, ‘re, ‘d, ‘ll, loại bỏ hết tất cả các dấu câu như dấu chấm (“.”), dấu phẩy (“,”), dấu chấm than(“!”),...

Những từ có cùng ‘gốc’ được đưa về cùng loại. Ví dụ, ‘include’, ‘includes’, ‘included’ đều được đưa chung về ‘include’. Tất cả các từ cũng đã được đưa về dạng ký tự thường (không phải HOA).



Trong đó:

Modul Word trong thư viện textblob sẽ xác định các từ trong đoạn

Hàm lemmatize() sẽ nhóm các từ cùng loại đó thành một từ

Tiếp theo chúng ta sẽ tách từ trong câu, chuẩn hóa từ, và loại bỏ stopwords (Số, dấu câu, ký tự ‘tabs’, ký tự ‘xuống dòng’ đã được loại bỏ.)



Sử dụng hàm TfidfVectorizer trong thư viện sklearn để chuyển bài báo thành ma trận các từ đặc trưng

Trong đó:

Stop\_words = ‘english’ : loại bỏ các stop words có trong bài báo

Các stop word bao gồm: ‘ourselves’, ‘hers’, ‘between’, ‘yourself’, ‘but’, ‘again’, ‘there’, ‘about’, ‘once’, ‘during’, ‘out’, ‘very’, ‘having’, ‘with’, ‘they’, ‘own’, ‘an’, ‘be’, ‘some’, ‘for’, ‘do’, ‘its’, ‘yours’, ‘such’, ‘into’, ‘of’, ‘most’, ‘itself’, ‘other’, ‘off’, ‘is’, ‘s’, ‘am’, ‘or’, ‘who’, ‘as’, ‘from’, ‘him’, ‘each’, ‘the’, ‘themselves’, ‘until’, ‘below’, ‘are’, ‘we’, ‘these’, ‘your’, ‘his’, ‘through’, ‘don’, ‘nor’, ‘me’, ‘were’, ‘her’, ‘more’, ‘himself’, ‘this’, ‘down’, ‘should’, ‘our’, ‘their’, ‘while’, ‘above’, ‘both’, ‘up’, ‘to’, ‘ours’, ‘had’, ‘she’, ‘all’, ‘no’, ‘when’, ‘at’, ‘any’, ‘before’, ‘them’, ‘same’, ‘and’, ‘been’, ‘have’, ‘in’, ‘will’, ‘on’, ‘does’, ‘yourselves’, ‘then’, ‘that’, ‘because’, ‘what’, ‘over’, ‘why’, ‘so’, ‘can’, ‘did’, ‘not’, ‘now’, ‘under’, ‘he’, ‘you’, ‘herself’, ‘has’, ‘just’, ‘where’, ‘too’, ‘only’, ‘myself’, ‘which’, ‘those’, ‘i’, ‘after’, ‘few’, ‘whom’, ‘t’, ‘being’, ‘if’, ‘theirs’, ‘my’, ‘against’, ‘a’, ‘by’, ‘doing’, ‘it’, ‘how’, ‘further’, ‘was’, ‘here’, ‘than’

min\_df = 2 : (min document frequency )Loại bỏ các từ có ít hơn 2 lần xuất hiện trong tất cả các bài báo

Fit\_transform : Học từ vựng và idf, trả về ma trận term document.

Đây là ví dụ của một bài báo trước khi xử lý:

Ad sales boost Time Warner profit

Quarterly profits at US media giant TimeWarner jumped 76% to $1.13bn (£600m) for the three months to December, from $639m year-earlier.

The firm, which is now one of the biggest investors in Google, benefited from sales of high-speed internet connections and higher advert sales. TimeWarner said fourth quarter sales rose 2% to $11.1bn from $10.9bn. Its profits were buoyed by one-off gains which offset a profit dip at Warner Bros, and less users for AOL.

Time Warner said on Friday that it now owns 8% of search-engine Google. But its own internet business, AOL, had has mixed fortunes. It lost 464,000 subscribers in the fourth quarter profits were lower than in the preceding three quarters. However, the company said AOL’s underlying profit before exceptional items rose 8% on the back of stronger internet advertising revenues. It hopes to increase subscribers by offering the online service free to TimeWarner internet customers and will try to sign up AOL’s existing customers for high-speed broadband. TimeWarner also has to restate 2000 and 2003 results following a probe by the US Securities Exchange Commission (SEC), which is close to concluding.

Time Warner’s fourth quarter profits were slightly better than analysts’ expectations. But its film division saw profits slump 27% to $284m, helped by box-office flops Alexander and Catwoman, a sharp contrast to year-earlier, when the third and final film in the Lord of the Rings trilogy boosted results. For the full-year, TimeWarner posted a profit of $3.36bn, up 27% from its 2003 performance, while revenues grew 6.4% to $42.09bn. “Our financial performance was strong, meeting or exceeding all of our full-year objectives and greatly enhancing our flexibility,” chairman and chief executive Richard Parsons said. For 2005, TimeWarner is projecting operating earnings growth of around 5%, and also expects higher revenue and wider profit margins.

TimeWarner is to restate its accounts as part of efforts to resolve an inquiry into AOL by US market regulators. It has already offered to pay $300m to settle charges, in a deal that is under review by the SEC. The company said it was unable to estimate the amount it needed to set aside for legal reserves, which it previously set at $500m. It intends to adjust the way it accounts for a deal with German music publisher Bertelsmann’s purchase of a stake in AOL Europe, which it had reported as advertising revenue. It will now book the sale of its stake in AOL Europe as a loss on the value of that stake.

Và sau khi đã được xử lý:

ad sale boost time warner profit quarterly profit at u medium giant timewarner jumped to for the three month to december from year earlier the firm which is now one of the biggest investor in google benefited from sale of high speed internet connection and higher advert sale timewarner said fourth quarter sale rose to from it profit were buoyed by one off gain which offset a profit dip at warner bros and le user for aol time warner said on Friday that it now owns of search engine google but it own internet business aol had ha mixed fortune it lost subscriber in the fourth quarter profit were lower than in the preceding three quarter however the company said aol underlying profit before exceptional item rose on the back of stronger internet advertising revenue it hope to increase subscriber by offering the online service free to timewarner internet customer and will try to sign up aol existing customer for high speed broadband timewarner also ha to restate and result following a probe by the u security exchange commission sec which is close to concluding time warner fourth quarter profit were slightly better than analyst expectation but it film division saw profit slump to helped by box office flop alexander and catwoman a sharp contrast to year earlier when the third and final film in the lord of the ring trilogy boosted result for the full year timewarner posted a profit of up from it performance while revenue grew to our financial performance wa strong meeting or exceeding all of our full year objective and greatly enhancing our flexibility chairman and chief executive richard parson said for timewarner is projecting operating earnings growth of around and also expects higher revenue and wider profit margin timewarner is to restate it account a part of effort to resolve an inquiry into aol by u market regulator it ha already offered to pay to settle charge in a deal that is under review by the sec the company said it wa unable to estimate the amount it needed to set aside for legal reserve which it previously set at it intends to adjust the way it account for a deal with german music publisher bertelsmann purchase of a stake in aol europe which it had reported a advertising revenue it will now book the sale of it stake

in aol europe a a loss on the value of that stake,business

đưa các bài báo đã được tiền xử lý vào file .csv để thuận tiện cho quá trình sử dụng sau này:



Tách dữ liệu từ tập dữ liệu ban đầu:



Bằng cách sử dụng hàm train\_test\_split trong thư viện sklearn.model\_selection để chia tập dữ liệu thành 2 phần là tập train và tập test, là tập dữ liệu huấn luyện và tập dữ liệu sử dụng để đánh giá hiệu quả sau này. Trong đó các biến và các tham số truyền vào được định nghĩa như sau:

X\_train: là tập các bài báo sử dụng trong quá trình huấn luyện.

X\_test: là tập bài báo sử dụng để phân loại nhãn trong việc đánh giá hiệu quả của mô hình.

Y\_train: là tập nhãn được sử dụng trong quá trình huấn luyện

y\_test: là tập nhãn được sử dụng trong việc đánh giá hiệu quả của mô hình.

X: là trọng số tf-idf của document-term matrix

Y : là mảng các nhãn được sử dụng trong tập dữ liệu ban đầu

test\_size = 0.2: chia dữ liệu thành 2 tập với tỉ lệ tập test là 20%

random\_state = 42: là chỉ số dùng để cố định đoạn mã, để sau mỗi lần chạy, hàm sẽ luôn lấy lượng dữ liệu như lần đầu, do đó sẽ không có sự thay đổi về kết quả

### Phân loại bài báo sử dụng Multinomial Naive Bayes



clf.fit(X\_train, y\_train): fit X\_train theo y\_train dựa theo phân loại Multinomial Naive Bayes

Chúng ta có:



Trong đó:

alpha = 1.0: giá trị laplace Smoothing được thêm vào

class\_prior = None : Xác suất tiên nhiệm của lớp, giá trị None là các lớp đã được điều chỉnh theo giá trị tiên nhiệm

fit\_prior = True: Có học lớp xác suất trước hay không. True ở đây là có học



Clf.predict(X\_test): Thực hiện phân loại theo vector X\_test, giá trị trả về là một mảng các giá trị dự đoán cho X

Y\_predict : Biến lưu trữ và dùng để so sánh với y\_test

### Đánh giá



Thực hiện đánh giá theo 2 giá trị y\_test và y\_predict chúng ta sẽ có kết quả tỷ lệ phân loại bài báo:

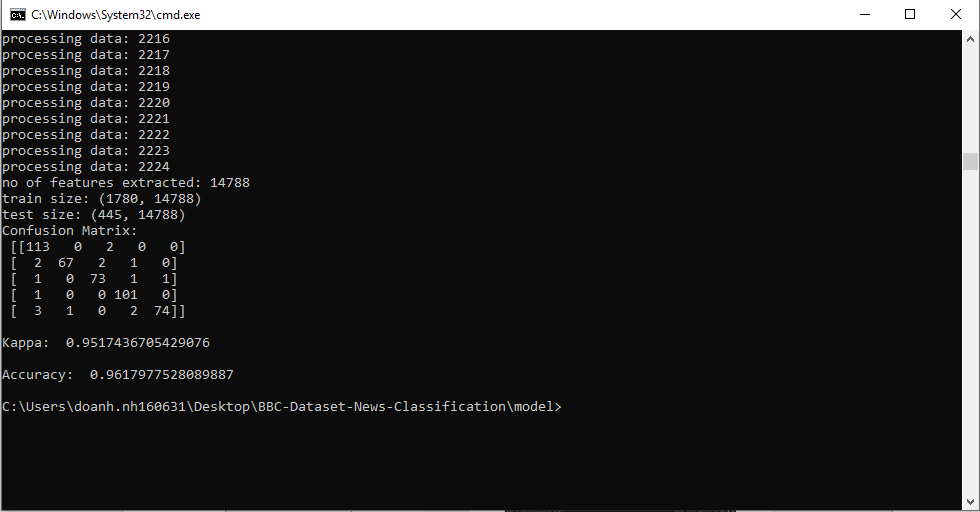


Với tập dữ liệu training là 1780 bản ghi thì thực hiện phân loại có độ chính xác tới 96.63%

# Kết luận

Hiện tại, với tập dữ liệu training là 1780 bài báo và tập dữ liệu test là 445 bài báo và thuật toán sử dụng là Naive Bayes Clustering thì mang lại tỉ lệ chính xác 96.63%

Khi so sánh với cùng một tập dữ liệu training và test với thuật toán Random Forest thì nhóm nhận thấy tỉ lệ chính xác là 96.17%, thấp hơn so với Naive Bayes Clustering. Vậy có thể thấy so với tập dữ liệu nhỏ, thì thuật toán Naive Bayes Clustering làm việc rất tốt và hiệu quả.



Qua project này, nhóm đã đạt được:

* Hiểu được cách viết một chương trình, thành thạo ngôn ngữ python và biết các sử dụng các thư viện hỗ trợ của python về marchine learning
* Hiểu và áp dụng được quy trình của một bài toán marchine learning.
* Hiểu và áp dụng được thuật toán Naive Bayes vào phân loại bài báo.
* Hoàn thành được chương trình phân loại bài báo đơn giản.

Do nhóm còn chưa có kinh nghiệm về ngôn ngữ, thuật toán và chưa thống nhất được một số vấn đề nên còn chậm trễ trong việc hoàn thành deadline và chưa hoàn thành được tất cả nhiệm vụ được giảng viện giao phó.

# Tài liệu tham khảo

* <https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html>
* <https://nlp.stanford.edu/IR-book/html/htmledition/text-classification-and-naive-bayes-1.html>
* <https://machinelearningcoban.com/2017/08/08/nbc/>
* <https://github.com/yoonkim/CNN_sentence/blob/master/process_data.py>