ĐẠI HỌC KHOA HỌC TỰ NHIÊN
Đại học Quốc gia Thành phố Hồ Chí Minh

BÁO CÁO CUỐI KÌ NHẬP MÔN KHOA HỌC DỮ LIỆU

MÔ HÌNH DỰ ĐOÁN TIN GIẢ (FAKE NEWS)

Giảng viên:
Hà Văn Thảo

Thành viên:
Trần Tiến Đạt - 20280016
Nguyễn Thị Hoa - 20280033
Nguyễn Thị Huyền - 20280048
Nguyễn Văn Sơn - 20280081

Tháng 6 Năm 2022

What is News?

Tin tức là thông tin về các sự kiện hiện tại. Điều này có thể được cung cấp thông qua nhiều phương tiện khác nhau: truyền miệng, in ấn, hệ thống bưu chính, phát thanh truyền hình, liên lạc điện tử, hoặc thông qua lời khai của các quan sát viên và nhân chứng về các sự kiện.

Các chủ đề phổ biến cho các bản tin bao gồm chiến tranh, chính phủ, chính trị, giáo dục, y tế, môi trường, kinh tế, kinh doanh, thời trang và giải trí, cũng như các sự kiện thể thao, sự kiện kỳ quặc hoặc bất thường. Các tuyên bố của chính phủ, liên quan đến các nghi lễ hoàng gia, luật pháp, thuế, sức khỏe cộng đồng và tội phạm, đã được coi là tin tức từ thời cổ đại. Sự phát triển về công nghệ và xã hội, thường được thúc đẩy bởi mạng lưới truyền thông và gián điệp của chính phủ, đã làm tăng tốc độ lan truyền tin tức cũng như ảnh hưởng đến nội dung của nó. Thể loại thời sự như chúng ta biết ngày nay gắn liền với tờ báo.

Trong thời hiện đại, hay như chúng ta đã biết, trong thế hệ máy tính, chúng ta đã chứng kiến một sự chuyển dịch hàng loạt từ chế độ bản cứng sang chế độ trực tuyến nhận thức Tin tức.

What is Fake News?

Tin tức giả là thông tin sai lệch hoặc gây hiểu lầm được trình bày dưới dạng tin tức. Nó thường có mục đích làm tổn hại danh tiếng của một cá nhân hoặc tổ chức hoặc kiếm tiền thông qua doanh thu quảng cáo. Học giả truyền thông Nolan Higdon đã đưa ra một định nghĩa rộng hơn về tin tức giả là "nội dung sai sự thật hoặc gây hiểu lầm được trình bày dưới dạng tin tức và được truyền đạt ở các định dạng bao gồm giao tiếp nói, viết, in, điện tử và kỹ thuật số."

Một khi đã phổ biến trên báo in, sự phổ biến của tin tức giả đã tăng lên cùng với sự gia tăng của các phương tiện truyền thông xã hội, đặc biệt là Facebook News Feed. Sự phân cực chính trị, chính trị hậu sự thật, thiên vị xác nhận và các thuật toán truyền thông xã hội có liên quan đến việc lan truyền tin tức giả mạo. Nó đôi khi được tạo ra và tuyên truyền bởi các tác nhân nước ngoài thù địch, đặc biệt là trong các cuộc bầu cử. Việc sử dụng các trang web tin tức giả mạo được lưu trữ ẩn danh đã gây khó khăn cho việc truy tố các nguồn tin tức giả mạo vì tội phỉ báng. Trong một số định nghĩa, tin tức giả mạo bao gồm các bài báo châm biếm bị hiểu sai là thật và các bài báo sử dụng các tiêu đề giật gân hoặc kích động không được hỗ trợ trong văn bản.

Mục lục

Phần I

GIỚI THIỆU

1. Tổng quan về mô hình phân tích dự đoán

Phân tích dự đoán (Predictive Analytics) là một khái niệm khá phổ biến, theo Google Trends thì nó có mức độ được quan tâm tăng đều đặn qua từng năm. Phân tích dự đoán dựa trên dữ liệu lịch sử, sử dụng thống kê, máy học và trí tuệ nhân tạo để dự đoán những gì sẽ xảy ra trong tương lai. Dữ liệu lịch sử này sẽ được đưa vào một mô hình toán học để xem xét các xu hướng và mẫu chính trong dữ liệu. Các công ty, các ứng dụng kinh doanh và các nhà phân tích có thể sử dụng mô hình này để biết trước liệu một thay đổi có giúp họ giảm thiểu rủi ro, cải thiện hoạt động hay tăng doanh thu hay không. Tóm lại, mô hình phân tích dự đoán sẽ trả lời cho câu hỏi: " *Điều gì có nhiều khả năng xảy ra nhất dựa trên dữ liệu hiện tại của tôi và tôi có thể làm gì để thay đổi kết quả đó?* "

Khi có được 1 bộ dữ liệu nào đó, đầu tiên ta phải xác định chủ đề mà dữ liệu đó hướng tới. Giải thích được từng thành phần dữ liệu từ đó hiểu được sự tồn tại có ý nghĩa của chúng trong dự hiệu. Cuối cùng dựa vào đó đưa ra định hướng phân tích, dự đoán một mô hình trong tương lai. Để xây dựng một mô hình phân tích và dự đoán:

Đầu tiên ta cần tìm hiểu về dữ liệu: Khi có được 1 bộ dữ liệu nào đó, đầu tiên ta phải xác định chủ đề mà dữ liệu đó hướng tới. Giải thích được từng thành phần dữ liệu từ đó hiểu được sự tồn tại có ý nghĩa của chúng trong dự hiệu. Từ đó đưa ra định hướng phân tích, dự đoán một mô hình trong tương lai.

Tiếp theo là xử lý dữ liệu: Một dữ liệu lớn thông thường sẽ có những lỗi về định dạng, kiểu dữ liệu hay tồn tại dữ liệu rác. Ở bước này, chúng ta có thể cần xây dựng tự động hóa bổ sung để làm cho việc thu thập, làm sạch và chuẩn bị dữ liệu dễ dàng và nhanh hơn. Đây có thể là công việc khó khăn, nhưng đó là công việc quan trọng. Muốn đảm bảo dữ liệu chuẩn bị đủ tốt ở giai đoạn này.

Sau đó xây dựng và đào tạo mô hình: Trong bước này, chúng ta sẽ sử dụng các phương pháp thống kê và học máy khác nhau để tạo ra một mô hình dự đoán. Sau đó lấy một số mô hình ứng cử viên từ bước trước và lặp đi lặp lại, đào tạo chúng bằng cách sử dụng dữ liệu đào tạo. Từ mô hình đó chúng ta sẽ thử nghiệm và đánh giá: Khi mô hình đã được đào tạo, huấn luyện, chúng ta sẽ kiểm tra nó trên dữ liệu mới để xem nó hoạt động tốt như thế nào. Chúng ta có thể tinh chỉnh thêm các mô hình hoặc thậm chí loại bỏ hoàn toàn một trong số chúng tùy thuộc vào kết quả thử nghiệm. Sau khi đã thử nghiệm, kiểm tra hoạt động của mô hình, chúng ta sẽ đưa

ra những nhật xét đánh giá về dữ liệu dựa trên mô hình đó.

Cuối cùng là đưa ra dự đoán phù hợp với nhu cầu của chúng ta.

Từ mô hình đã tạo có thể đưa ra nhận định đánh giá của ta đối với dữ liệu. Nếu ta làm điều này một cách chính xác thì công ty, tổ chức của chúng ta sẽ được hưởng lợi, giảm thiểu được nhiều rủi ro đến từ các phân tích dự đoán.

2. Tin giả (Fake news)

Trong thời đại phát triển như hiện nay, khối lượng thông tin mà chúng ta được tiếp cận, cập nhật mỗi ngày từ các nguồn như mạng xã hội, tờ rơ, báo mạng … là vô cùng lớn. Nhưng bạn có chắc rằng, những luồn tin ấy có thực sự bổ ích và chính xác hay không? Nếu bạn tiếp nhận những thông tin sai sự thật ấy thì đều gì sẽ xảy ra?

Người ta đã chỉ ra rằng việc tuyên truyền tin tức giả đã có ảnh hưởng không nhỏ đến các cuộc bầu cử tổng thống Hoa Kỳ năm 2016. Một vài sự thật về tin tức giả mạo ở Hoa Kỳ:

- 62 công dân Hoa Kỳ nhận được tin tức của họ cho các phương tiện truyền thông xã hội
- Tin tức giả mạo được chia sẻ trên Facebook nhiều hơn tin tức chính thống

Tin tức giả mạo cũng đã được sử dụng để gây ảnh hưởng đến cuộc trưng cầu dân ý ở Vương quốc Anh cho "Brexit". Tin giả như một loại virus độc hại, nó xâm nhập, gây rối dư luận, gây rối lòng tin, thậm chí làm khủng hoảng niềm tin. Vì vậy việc xử lý,lọc ra những thông tin đáng tin cậy và có độ chính xác cao là một trong những điều cần thiết ngay lúc này. Bài báo cáo này sẽ giúp chúng ta dự đoán được liệu một mục tin tức có phải là giả mạo hay không. Dựa trên bộ dữ liệu Fake - True, chúng ta sẽ có những định hướng phân tích cũng như xây dựng và đào tạo một mô hình. Từ đó đưa ra những đánh giá, nhận biết chính xác hơn về tin giả. Với nội dung này, sinh viên ngành Khoa học dữ liệu không những có được cái nhìn toàn diện hơn về những lượng dữ liệu lớn mà bản thân còn có trách nhiệm sàng lọc, tiếp nhận thông tin chính xác, phủ nhận thông tin xuyên tạc để đưa ra quyết định đúng đắn nhất.

Phần II
PHÂN TÍCH VÀ XÂY DỰNG MÔ HÌNH

1. Tìm hiểu về dữ liệu của đề tài

Dữ liệu của đề tài được lấy từ nguồn dữ liệu uy tín và khá phổ biến (Kaggle), dữ liệu bao gồm hai tập tin Comma Separated Values (CSV): `Fake.csv` và `True.csv`. **Fake.csv** là tập tin chứa thông tin của các bài báo giả mạo (Tin giả) trong khi tập tin **Real.csv** cho ra những bảng tin có độ chính xác cao. Hai tập tin đều là bảng thống kê theo thời gian và có đầy đủ các thuộc tính nhằm giúp cho người đọc có thể dễ dàng hiểu thông tin một cách nhanh chóng.

Mỗi dòng là thông tin của 1 bài báo, bao gồm:

- `title` : tên tiêu đề
- `text`: nội dung bài báo
- `subject`: chủ đề của bài báo
- `date`: ngày công bố

Trong hai bảng dữ liệu, ta thấy rằng có 2 loại dữ liệu :

- Loại text (là dạng ký tự chữ, chuỗi ký tự, không phải là dạng số: title, text
- Loại date (là dạng ngày tháng: date)

Để phân tích và dự đoán một dữ liệu nào đó thì việc làm đầu tiên là gọi vào dữ liệu đó để có thể sử dụng chúng. Ta tiến hành gọi các dữ liệu từ hai tập tin đã chuẩn bị trước đó:

```
1 df_fake = pd.read_csv("Fake.csv")
2 df_true = pd.read_csv("True.csv")
```

Khi nhập dữ liệu vào hai dataframe **df_fake** và **df_true**, ta thêm vào cả hai dataframe đó lần lượt hai cột `class` mang giá trị là **0** và **1** để phân biệt hai bảng thông tin thật và giả. Dữ liệu của cả hai dataframe sẽ cùng tăng thêm một cột so với ban đầu (khi nhập vào) và giữ nguyên số dòng.

2. Xử lý dữ liệu

Tạo dataframe mới có tên **df_fake_manual_testing** từ 10 phần tư cuối cùng của **df_fake** và tương tự cho **df_true_manual_testing** với 10 phần tử cuối cùng của **df_true**. Đồng thời cũng xóa 10 dòng cuối đó ở cả **df_fake** và **df_true**.

```
df_fake_manual_testing = df_fake.tail(10)
for i in range(23480,23470,-1):
    df_fake.drop([i], axis = 0, inplace = True)

df_true_manual_testing = df_true.tail(10)
for i in range(21416,21406,-1):
    df_true.drop([i], axis = 0, inplace = True)
```

Tiếp theo, ta tiến hành gộp **df_fake_manual_testing** và **df_true_manual_testing** thành một dataframe mới có tên **df_manual_testing**, sau đó xuất **df_manual_testing** thành một file có tên manual_testing.csv. Tương tự, ta cũng tạo dataframe có tên **df_merge** từ **df_fake** và **df_true**.

```
df_manual_testing = pd.concat([df_fake_manual_testing, ↩
   df_true_manual_testing], axis = 0)
df_manual_testing.to_csv("manual_testing.csv")
```

Tiếp đến, ta xóa ba cột: tittle, subject và date của **df_merge** và sắp xếp lại dataframe này một cách ngẫu nhiên.

	text	class
34042	the families of the four dead americans you le...	0
41749	the unexpected passing of justice antonin scal...	0
27208	washington reuters the nominee to be the n...	1
20221	during the democrat s debate hillary clinton ...	0
785	how bad is trump for the gop it seems he s wo...	0

Vì được sắp xếp ngẫu nhiên nên chỉ mục ban đầu của df sẽ không theo thứ tự nhất định, lúc này ta cần định dạng lại chỉ mục cho nó để chỉ mục của df được sắp xếp theo thứ tự (bắt đầu từ 0 và tăng dần giống như số thứ tự).

	text	class
0	the families of the four dead americans you le...	0
1	the unexpected passing of justice antonin scal...	0
2	washington reuters the nominee to be the n...	1
3	during the democrat s debate hillary clinton ...	0
4	how bad is trump for the gop it seems he s wo...	0

Để xây dựng một mô hình dựa trên dữ liệu này, chúng ta cần tiến hành làm sạch dữ liệu. Viết hàm wordopt để chuyển văn bản sang định dạng chữ thường, đồng thời xóa các khoảnh cách dư và các kí tự đặc biệt.

Sau khi có được hàm wordopt ta áp dụng nó cho cột text của dataframe - df ta thấy dữ liệu của cột text đã được xử lí đưa về dạng chữ thường, không còn khoảng trống dư và các kí tự đặc biệt.

Để tiến hành xây dựng và đào tạo mô hình, ta tạo vector x và y lần lượt chứa nội dung của cột text và cột class của bảng df, sau đó sử dụng train_test_split() từ thư viện scikit-learning để chia dữ liệu vào các tập đào tạo và tập kiểm tra với test_size = 0.25 (kích thước của tập kiểm tra chiếm 25%) dữ liệu sẽ được chia một cách ngẫu nhiên và đưa vào các biến x_test, y_test, x_train, y_train. Trong đó:

```
def wordopt(text):
    text = text.lower()
    text = re.sub('\[.*?\]', '', text)
    text = re.sub("\\W"," ",text)
    text = re.sub('https?://\S+|www\.\S+', '', text)
    text = re.sub('<.*?>+', '', text)
    text = re.sub('[%s]' % re.escape(string.punctuation), '', text)
    text = re.sub('\n', '', text)
    text = re.sub('\w*\d\w*', '', text)
    return text
```

- x_test : chứa 25% dữ liệu của x
- y_test : chứa 25% dữ liệu của y

- x_train : chứa 75% dữ liệu của x
- y_train : chứa 75% dữ liệu của y

Để xây dựng mô hình dự đoán tin giả (Fake News), chúng ta sẽ tập trung vào các phương pháp truyền thống được sử dụng trong xử lý ngôn ngữ tự nhiên điển hình là **Naive Bayes**. Điều đầu tiên cần làm khi xử lí văn bản là thực hiện nhúng các từ và văn bản để sử dụng các thuật toán học máy học. Văn bản cần được biểu diễn theo cách cung cấp nhiều thông tin có ý nghĩa hơn là một chuỗi bit đơn giản, có những nhược điểm đối với một số từ nhất định, chuỗi bit biểu diễn nó phụ thuộc vào mã hóa. Mã hóa đơn giản là việc chúng ta chia đoạn văn thành các câu văn, các câu văn thành các từ. Trong mã hóa thì từ là đơn vị cơ sở. Chúng ta cần một bộ tokenizer có kích thước bằng toàn bộ các từ xuất hiện trong văn bản hoặc bằng toàn bộ các từ có trong từ điển. Các bộ tokernizer sẽ khác nhau cho mỗi một ngôn ngữ khác nhau. Ở đây, ta sẽ sử dụng CountVectorizer.

CountVectorizer là một công cụ tuyệt vời được cung cấp bởi thư viện scikit-learning bằng Python. Nó được sử dụng để chuyển một văn bản nhất định thành một vectơ trên cơ sở tần suất (số lượng (count)) xuất hiện của mỗi từ trong toàn bộ văn bản. Điều này rất hữu ích khi chúng ta có nhiều văn bản như vậy và chúng ta muốn chuyển đổi từng từ trong mỗi văn bản thành vectơ (để sử dụng trong phân tích văn bản sâu hơn).

Chúng ta sử dụng các túi từ (bags of words) để tạo ra một vector có độ dài bằng độ dài của tokenizer và mỗi phần tử của túi từ sẽ đếm số lần xuất hiện của một từ trong câu và sắp xếp chúng theo một vị trí phù hợp trong vector.

Dưới đây là 1 ví dụ nhằm xem xét một vài văn bản mẫu từ một tài liệu (mỗi văn bản là một phần tử danh sách):

```
from sklearn.feature_extraction.text import CountVectorizer
corpus = [
'This is the first document.',
'This document is the second document.',
'And this is the third one.',
'Is this the first document?']
countvectorizer = CountVectorizer()
terms = countvectorizer.fit_transform(corpus)
print(countvectorizer.get_feature_names())
print("Sparse Matrix form of test data : \n")
print(terms.todense())
```

CountVectorizer sẽ tạo một ma trận (matrix) trong đó mỗi từ duy nhất được biểu thị bằng một cột (với cột đầu tiên: “and”, cột thứ 2: “document”, ...) của ma trận và mỗi mẫu văn bản

từ tài liệu là một hàng (row) trong ma trận. Giá trị của mỗi ô (cell) là số lượng từ trong mẫu văn bản cụ thể đó.

- Có 9 từ trong corpus, được biểu diễn dưới dạng các cột của bảng : and document first is one second the third this
- Có 4 mẫu văn bản trong corpus, mỗi mẫu được biểu thị dưới dạng các hàng của bảng.
- Mỗi ô trong ma trận chứa một số, đại diện cho số lượng xuất hiện từ đó trong văn bản cụ thể đó.
- Tất cả các từ đã được chuyển đổi thành chữ thường.
- Các từ trong các cột đã được sắp xếp theo thứ tự bảng chữ cái.

Biểu diễn thực tế đã được hiển thị trong bảng dưới đây:

```
['and', 'document', 'first', 'is', 'one', 'second', 'the', 'third', 'this']
Sparse Matrix form of test data :

[[0 1 1 1 0 0 1 0 1]
 [0 2 0 1 0 1 1 0 1]
 [1 0 0 1 1 0 1 1 1]
 [0 1 1 1 0 0 1 0 1]]
```

Cách biểu diễn này được gọi là ma trận thưa (sparse matrix).

Ta sẽ áp dụng phương pháp vào nội dung x_train của bài của mình.

```
vectorization = CountVectorizer()
xv_train = vectorization.fit_transform(x_train)
xv_test = vectorization.transform(x_test)
print(xv_train.shape)
```

```
(35902, 97296)
```

Thu được ma trận có với **35902** hàng đại diện cho số lượng text và **97299** cột đại diện cho số từ khác nhau (duy nhất) trong **x_train**.

Sau đó xuất ra mảng các từ khác nhau xuất hiện trong **x_train** để kiểm tra

```
print(vectorization.get_feature_names())
print(xv_train.toarray())
```

3. Xây dựng và đào tạo mô hình từ dữ liệu

Mô hình được sử dụng để phân loại văn bản (biến rời rạc) biểu diễn dưới dạng ma trận vector trong mô hình này là **Multinomial Naive-Bayes**. Đây là một mô hình học máy xác suất được sử dụng cho nhiệm vụ phân loại. Lấy ý tưởng cơ bản của mô hình Naive-Bayes, tất cả các tính năng đều độc lập với nhau. Đây là một giả thuyết đặc biệt được ứng dụng trong trường hợp phân loại văn. Cho một phần tử của lớp c và vectơ của đối tượng $\mathbf{X} = (x_1, \ldots, x_n)$. Xác suất của lớp cho rằng vectơ đó được xác định là:

$$P(c|\mathbf{X}) = \frac{P(c) * P(\mathbf{X}|c)}{P(\mathbf{X})} \quad (1)$$

Dựa vào giả định về tính độc lập có điều kiện nên:

$$P(x_i|c, x_1, \ldots, x_{i-1}, x_i, \ldots, x_n) = P(x_i|c) \quad (2)$$

Áp dụng quy tắc Bayes:

$$P(c|x_1, \ldots, x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i|c)}{P(x_1, \ldots, x_n)} \quad (3)$$

Vì $P(x_1, \ldots, x_n)$ không đổi nên ta có quy tắc phân loại:

$$\hat{c} = argmax P(c)\prod_{i=1}^{n} P(x_i|c) \quad (4)$$

Với mô hình**Multinomial Naive-Bayes**, 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, $P(x_i|c)$ tỉ lệ với tần suất từ thứ i xuất hiện trong các văn bản của lớp c. Giá trị này có thể được tính bằng cách:

$$\lambda_{ci} = p(x_i|c) = \frac{N_{ci}}{N_c} \quad (5)$$

Trong đó:

- N_{ci} là tổng số lần từ thứ i xuất hiện trong các văn bản của lớp c

- N_c là tổng độ dài của toàn bộ các văn bản thuộc vào lớp c.

  Có thể suy rằng: $N_c = \sum_{i=1}^{d} N_{ci}$, từ đó $\sum_{i=1}^{d} \lambda_{ci} = 1$

Cách tính này có một hạn chế là nếu có một từ mới chưa bao giờ xuất hiện trong lớp c thì biểu thức trên sẽ bằng 0, điều này dẫn đến vế phải của (4) bằng 0 bất kể các giá trị còn lại có lớn thế nào. Việc này sẽ dẫn đến kết quả không chính xác.

Để giải quyết vấn đề này, một kỹ thuật được gọi là **Laplace smoothing** được áp dụng:

$$\hat{\lambda}_{ci} = \frac{N_{ci} + \alpha}{N_c + d\alpha} \quad (6)$$

Với α là một số dương (thường bằng 1 để tránh trường hợp tử số bằng 0) thì mẫu số được cộng với d để đảm bảo tổng xác suất $\sum_{i=1}^{d} \hat{\lambda}_{ci}$

Như vậy, mỗi lớp c sẽ được mô tả bởi bộ các số dương có tổng bằng 1:

$$\hat{\lambda}_c = \{\hat{\lambda}_{c1}, \ldots \hat{\lambda}_{cd}\} \quad (7)$$

Xét ví dụ

Giả sử trong tập training có các văn bản $d1, d2, d3, d4$ như trong bảng sau. Mỗi văn bản này thuộc vào 1 trong 2 lớp: B *(Bắc)* hoặc N *(Nam)*. Hãy xác định lớp của văn bản $d5$.

Dựa trên tần số xuất hiện của mỗi lớp trong tập training. Ta sẽ có:

	Document	**Content**	**Class**
Training	d1	hanoi pho chaolong hanoi	B
	d2	hanoi buncha pho omai	B
	d3	pho banhgio omai	B
	d4	saigon hutiu banhbo pho	N
Testing	d5	hanoi hanoi buncha hutiu	?

$$P(B) = \frac{3}{4} \qquad P(N) = \frac{1}{4}$$

Tập hợp toàn bộ các từ trong văn bản là:

$V =$ {hanoi, pho, chaolong, buncha, omai, banhgio, saigon, hutiu, banhbo}

Tổng cộng số phần tử trong từ điển là $|V| = 9$

Sử dụng Laplace smoothing với $\alpha = 1$

Sau đây là minh hoạ quá trình Training và Test cho bài toán này khi sử dụng **Multinomial Naive Bayes**:

TRAINING

class = B

	hanoi	pho	chaolong	buncha	omai	banhgio	saigon	hutiu	banhbo	
d1: $\mathbf{x}_1$	2	1	1	0	0	0	0	0	0	
d2: $\mathbf{x}_2$	1	1	0	1	1	0	0	0	0	
d3: $\mathbf{x}_3$	0	1	0	0	1	1	0	0	0	$d = \lvert V \rvert = 9$
Total	3	3	1	1	2	1	0	0	0	$\Rightarrow N_{\mathrm{B}} = 11$
$\Rightarrow \hat{\lambda}_{\mathrm{B}}$	4/20	4/20	2/20	2/20	3/20	2/20	1/20	1/20	1/20	$(20 = N_{\mathrm{B}} + \lvert V \rvert)$

class = N

d4: $\mathbf{x}_4$	0	1	0	0	0	0	1	1	1	$\Rightarrow N_{\mathrm{N}} = 4$
$\Rightarrow \hat{\lambda}_{\mathrm{N}}$	1/13	2/13	1/13	1/13	1/13	1/13	2/13	2/13	2/13	$(13 = N_{\mathrm{N}} + \lvert V \rvert)$

TEST

d5: $\mathbf{x}_5 = [2, 0, 0, 1, 0, 0, 0, 1, 0]$

$$p(\mathrm{B}|\mathrm{d5}) \propto p(\mathrm{B}) \prod_{i=1}^{d} p(x_i|\mathrm{B})$$

$$= \frac{3}{4} \left(\frac{4}{20}\right)^2 \frac{2}{20} \frac{1}{20} \approx 1.5 \times 10^{-4}$$

$$p(\mathrm{N}|\mathrm{d5}) \propto p(\mathrm{N}) \prod_{i=1}^{d} p(x_i|\mathrm{N})$$

$$= \frac{1}{4} \left(\frac{1}{13}\right)^2 \frac{1}{13} \frac{2}{13} \approx 1.75 \times 10^{-5}$$

$\Rightarrow p(\mathbf{x}_5|\mathrm{B}) > p(\mathbf{x}_5|\mathrm{N}) \Rightarrow \mathrm{d5} \in \mathrm{class(B)}$

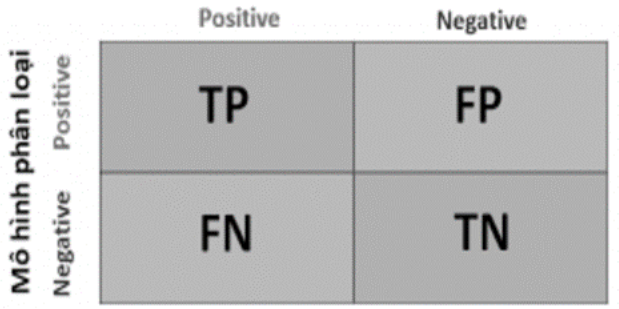
Hai giá trị tìm được 1.5×10^{-4} và 1.75×10^{-5} không phải là hai xác suất cần tìm mà chỉ là hai đại lượng tỉ lệ thuận với hai xác suất đó. Vậy nên có thể dự đoán rằng $d5$ thuộc lớp Bắc.

Sau khi hiểu về **Multinomial Naive-Baye** ta tiến hành áp dụng để xây dựng mô hình dự đoán thông tin thật – giả với sự hỗ trợ từ thư viện *sklearn.naive_bayes*. Ta tiến hành đào tạo tập dữ liệu với **xv_train** và **y_train** đã nói ở trên :

```
from sklearn.naive_bayes import MultinomialNB
classifier = MultinomialNB().fit(xv_train,y_train)
```

Ma trận nhầm lẫn

Ma trận nhầm lẫn (confusion matrix) Là một phương pháp đánh giá kết quả của những bài toán phân loại với việc xem xét cả những chỉ số về độ chính xác và độ bao quát của các dự đoán cho từng lớp. Một confusion matrix gồm 4 chỉ số sau đối với mỗi lớp phân loại: TP, TN, FP, FN.



- **TP (True Positive)**: Các giá trị thực sự Positive và được dự đoán là Positive.
- **FP (False Positive)**: Các giá trị thực sự là Negative nhưng được dự đoán sai là Positive. Còn được gọi là Lỗi loại I.
- **FN (False Negative)**: Các giá trị thực sự là Positive nhưng được dự đoán sai là Negative. Còn được gọi là Lỗi loại II.
- **TN (True Negative)**: Các giá trị thực sự Negative và được dự đoán là Negative.

Từ 4 chỉ số này, ta có 2 con số để đánh giá mức độ tin cậy của một mô hình:

- **Accuracy:** : Nó được tính bằng cách chia tổng số dự đoán đúng cho tất cả các dự đoán

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

- **Precision:** : Trong tất cả các dự đoán Positive được đưa ra, bao nhiêu dự đoán là chính xác? Chỉ số này được tính theo công thức:

$$Precison = \frac{TP}{TP + FP}$$

- **Recall:** :Trong tất cả các trường hợp Positive, bao nhiêu trường hợp đã được dự đoán chính xác? Chỉ số này được tính theo công thức:

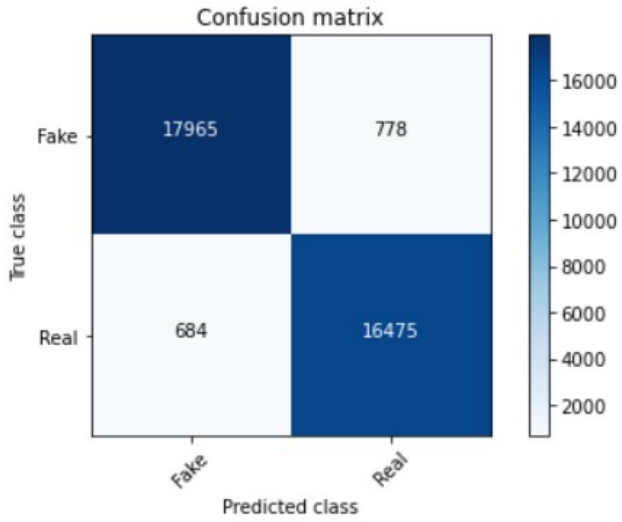
$$Recall = \frac{TP}{TP + FN}$$

- **F1 - score:** : số dung hòa Recall và Precision giúp ta có căn cứ để lựa chọn model. F1 càng cao càng tốt.

$$f_1 - score = \frac{2 * Precision * Recall}{Precision + Recall}$$

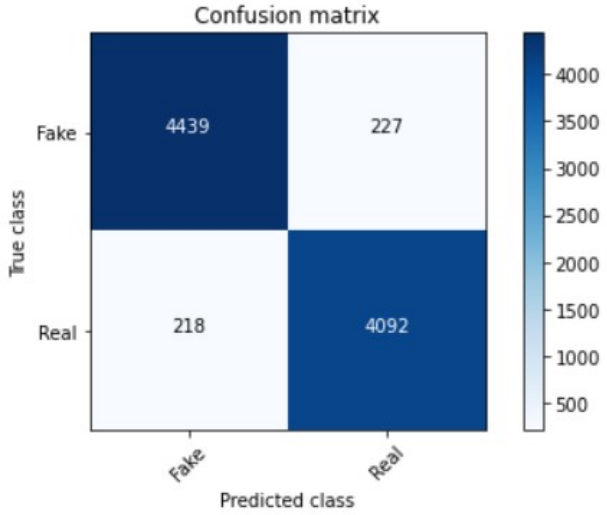
Áp dụng nội dung của ma trận nhầm lần vào `y_train` ta được:

	precision	recall	f1-score	support
0	0.96	0.96	0.96	18743
1	0.95	0.96	0.96	17159
accuracy			0.96	35902
macro avg	0.96	0.96	0.96	35902
weighted avg	0.96	0.96	0.96	35902



Áp dụng nội dung của ma trận nhầm lần vào `y_test` ta được:

	precision	recall	f1-score	support
0	0.95	0.95	0.95	4666
1	0.95	0.95	0.95	4310
accuracy			0.95	8976
macro avg	0.95	0.95	0.95	8976
weighted avg	0.95	0.95	0.95	8976



4. Thử nghiệm và dự đoán

Trước hết, viết hàm output_label có tham số n, với giá trị trả về là **"Fake News"** nếu n=0 và ngược lại **"Not a Fake News"** khi n=1 để đưa ra kết quả.

```
def wordopt(text):
    def output_lable(n):
    if n == 0:
        return "Fake News"
    elif n == 1:
        return "Not A Fake News"
```

Tiếp theo, viết hàm manual_tesing có tham số truyền vào là news nhằm kiểm nghiệm thông tin và đưa ra kết quả. Trong đó, tạo biến testing_news có giá trị bằng với tham số news được truyền vào ở dạng text. Sau đó, đưa testing_news về dạng dataframe có tên new_def_test. Tiến hành áp dụng hàm wordopt để làm sạch dữ liệu cho cột "text" của new_def_test. Lưu các giá trị cột text vào biến new_x_test và tiến hành vector hóa nó với tên new_xv_test. Áp dụng model mà ta đã xây dựng, đào tạo ở trên vào new_xv_test để đứa ra kết quả dự đoán thông qua hàm output_label.

```
def manual_testing(news):
    testing_news = {"text":[news]}
    new_def_test = pd.DataFrame(testing_news)
    new_def_test["text"] = new_def_test["text"].apply(wordopt)
    new_x_test = new_def_test["text"]
    new_xv_test = vectorization.transform(new_x_test)
    pred = classifier.predict(new_xv_test)
    return print(" \nMultinomial Naive Bayes : {}".format(↩
       output_lable(pred[0])))
```

Phần III
KẾT LUẬN

Trong bài báo cáo này, chúng ta đã sử dụng tập dữ liệu Tin tức giả mạo có sẵn công khai của Kaggle, bao gồm 44.898 tin bài, trong đó 21.417 tin tức là thật và 23.481 tin tức giả được gắn nhãn là 0 và 1. Với bộ phân loại học máy thường được sử dụng là **Multinomial Naive-Baye** để phân tích tập dữ liệu và sử dụng tính năng **CountVectorizer**. Nhờ sử dụng các tính năng và phân loại nói trên, xây dựng một mô hình phát hiện tin tức thật - giả có kết quả thử nghiệm cho thấy, phương pháp đề xuất của chúng ta đạt độ chính xác: 95%, precision: 95% , recall: 95% và F1- score: 95%. Đánh giá xác nhận rằng mô hình phân tích dự đoán này cho kết quả chính xác cao tới 95%.

Tin thật - giả luôn là một trong các chủ đề đáng quan tâm và lo ngại. Nó ảnh hưởng trực tiếp đến cuộc sống sinh hoạt hằng ngày của mỗi chúng ta. Việc tiếp nhận những thông tin ấy một cách tích cực hay tiêu cực là phụ thuộc vào nhận thức của mỗi người. Với việc xây dựng mô hình phân tích và dự đoán trên, chúng tôi hi vọng sẽ phần nào giúp mỗi người chúng ta có thể phân biệt được tin tức nào là giả, tin tức nào là thật để tin tức mỗi ngày chúng ta được cập nhật chính xác hơn

Phần IV

TÀI LIỆU THAM THẢO

[1] Fake - Real News Dataset (Kaggle)

[2] Exercise 6: Naive Bayes - Machine Learning - Andrew Ng

[3] sklearn.naive_bayes

[4] 6 Easy Steps to Learn Naive Bayes Algorithm (with code in Python)