ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA HỆ THỐNG THÔNG TIN

ĐỒ ÁN KHAI THÁC DỮ LIỆU

Đề tài: PHÂN LOẠI BÀI BÁO ĐIỆN TỬ

Lớp: IS252.J22

Giảng viên hướng dẫn: PGS.TS Đỗ Phúc

Huỳnh Thiện Ý

Sinh viên thực hiện:

1) Nguyễn Phi Yến 16521484
2) Lê Minh Đức 17520358
3) Trần Đức Phát 17520881

TP. Hồ Chí Minh, Ngày 10- 04- 2019

MỤC LỤC

LỜI CẢM ƠN

Lời đầu tiên, nhóm End Game xin gửi lời cảm ơn chân thành đến quý Thầy Cô trường Đại học Công nghệ thông tin, đặc biệt là quý Thầy Cô Khoa Hệ thống thông tin - những người đã dùng tri thức và tâm huyết của mình để truyền đạt cho chúng em vốn kiến thức vô cùng quý báu trong khoảng thời gian học tập tại trường. Những kiến thức mà Thầy Cô truyền đạt là bước đệm quan trọng giúp chúng em có thể hoàn thành đề tài tốt hơn.

Nhóm End Game xin gửi lời cảm ơn đặc biệt chân thành tới thầy Huỳnh Thiên Ý– giáo viên thực hành môn Khai thác dữ liệu đã tận tình giúp đỡ, trực tiếp chỉ bảo, hướng dẫn nhóm trong suốt quá trình làm đồ án môn học. Nhờ đó, chúng em đã tiếp thu được nhiều kiến thức bổ ích trong việc vận dụng cũng như kỹ năng làm đồ án.

Trải qua thời gian một học kỳ thực hiện đề tài. Với sự hướng dẫn tận tình cùng những đóng góp quý báu của Cô/Thầy và các bạn giúp nhóm End Game hoàn thành tốt hơn báo cáo môn học của mình. Bên cạnh việc vận dụng những kiến thức được học trên lớp đồng thời kết hợp với việc học hỏi và tìm hiểu những kiến thức mới. Từ đó, nhóm đã vận dụng tối đa những gì đã tiếp thu được để hoàn thành một báo cáo đồ án tốt nhất. Tuy nhiên, trong quá trình thực hiện, không tránh khỏi những sai sót. Dó đó, rất mong nhận được những sự góp ý từ Cô/Thầy nhằm giúp nhóm hoàn thiện những kiến thức đã học tập và cũng là hành trang để nhóm thực hiện tiếp các đề tài khác trong tương lai. Xin chân thành cảm ơn quý Thầy Cô và các bạn!

Nhóm sinh viên thực hiện

Nhóm End Game

NHẬN XÉT CỦA GIẢNG VIÊN

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

1. Đặt vấn đề

Thu thập thông tin là một trong những công cụ cơ bản để mở rộng tầm nhìn cho nhà lãnh đạo. Hãy thử hình dung, chúng ta sẽ lãnh đạo tổ chức thế nào nếu không có kiến thức nền và không biết những gì đang xảy ra xung quanh.

Nhà lãnh đạo cần thu thập hai loại thông tin chính, một là thông tin nền tảng và hai là thông tin liên quan đến công việc của mình. Thu thập thông tin nền tảng để xây dựng quan điểm của họ về thế giới mà họ đang sống. Thông tin này đến từ thực tế, từ những xu hướng và những quan điểm mà họ gặp và quan sát hàng ngày. Chất lượng thông tin họ thu thập được càng cao, họ càng xử lý nó hiệu quả hơn, quan điểm của họ với thế giới càng chính xác hơn, phán đoán và cảm nhận của họ cũng tốt hơn. Chẳng hạn, nếu bạn đang có một kế hoạch kinh doanh năm năm thì bạn cần một dự báo phát triển đáng tin cậy từ ngân hàng Nhà nước. Hoặc có thể bạn muốn có thông tin về những thu nhập sẵn có của một nhóm khách hàng xu hướng thị trường lao động....v.v

Ngày nay, với sự phát triển mạnh mẽ của công nghệ thông tin và những ứng dụng của nó trong đời sống. Máy tính điện tử không còn là thứ phương tiện lạ lẫm đối với mọi người mà nó dần trở thành công cụ làm việc, giải trí thông dụng và hữu ích của chúng ta, không chỉ ở công sở mà ngay cả trong gia đình.

Trong nền kinh tế hiện nay, với xu thế toàn cầu hóa nền kinh tế thế giới, mọi mặt của đời sống xã hội ngày càng được nâng cao, đặc biệt là nhu cầu tìm kiếm và nắm bắt thông tin của nỗi cá nhân mỗi tập thể ngày càng được chú trọng, nắm bắt được điều này nhóm chúng em đã đưa ra một giải pháp ứng dụng Công nghệ thông tin và cụ thể trong lĩnh vực Khai thác dữ liệu để đó là crawl dữ liệu từ các trang báo điện tử lớn như Báo Dân Trí, Báo Mới, VnExpress (nguồn cung cấp thông tin đáng tin cậy) từ đó huấn luyện để máy móc có thể đưa ra dự đoán về nội dung văn bản. Dựa vào đó ta có thể năm bắt 1 cách dễ dang nhưng thông tin mà ta đang quan tâm hỗ giúp nhà đầu tư hay doanh nghiệp có thể xem xét và đánh giá trước khi đưa ra các quyết định cũng như các giải pháp định hướng trong tương lai

2. Mục tiêu

- Xây dựng hệ thống dữ liệu về ngôn ngữ tự nhiên, sử dụng máy học để huấn luyện máy móc có thể đưa ra những thông tin , những dự đoán có độ tin cậy cao phục vụ con người.
- Dự đoán thể loại bài báo, giúp tự động hóa dần các công việc làm thủ công thường ngày, tiết kiệm thời gian cho người dùng

3. Công cụ sử dụng

Trong quá trình thực hiện, nhóm đã sử dụng một số phần mềm phục vụ cho việc tìm

hiểu và xây dựng đề tài:

- Phần thu thập và phân tích thông tin sử dụng thư viện và ngôn ngữ lập trình python
- Dữ liệu: Các bài báo trên trang báo điện tử VnExpress

Tất cả các phần mềm trên được nhóm cài đặt và sử dụng trên Hệ điều hành Microsoft Windows 10. Việc tương thích các phần mềm trên với các hệ điều hành khác không nằm trong phạm vi nghiên cứu của đề tài này.

CHƯƠNG 2: DỮ LIỆU SỬ DỤNG ĐỂ THỰC NGHIỆM

1. Giới thiệu thư viện Scrapy

- Scrapy là một framework được viết bằng Python, nó cấp sẵn 1 cấu trúc tương đối hoàn chỉnh để thực hiện việc crawl và extract data từ website một cách nhanh chóng và dễ dàng. Bạn muốn lấy dữ liệu từ các website nhưng dữ liệu đó quá lớn để copy rồi paste vào database của bạn, scrapy hỗ trợ bạn làm điều đó. Việc lấy dữ liệu website hoàn toàn tự động nhanh chóng và việc sử dụng scrapy cũng rất đơn giản giúp bạn tiếp kiệm được nhiều thời gian và công sức.

- Trang chủ scrapy: https://scrapy.org

- Lệnh cài đặt:

```
C:\WINDOWS\system32>pip install scrapy
```

1.1. Thành phần

- Scrapy Engine có trách nhiệm kiểm soát luồng dữ liệu giữa tất cả các thành phần của hệ thống và kích hoạt các sự kiện khi một số hành động xảy ra

- Giống như một hàng đợi (queue), scheduler sắp xếp thứ tự các URL cần download

- Thực hiện dowload trang web và cung cấp cho engine

- Spiders là class được viết bởi người dùng, chúng có trách nhiệm bóc tách dữ liệu cần thiết và tạo các url mới để nạp lại cho scheduler qua engine.

- Những dữ liệu được bóc tách từ spiders sẽ đưa tới đây, Item pipeline có nhiệm vụ xử lý chúng và lưu vào cơ sở dữ liệu

- Là các thành phần nằm giữa Engine với các thành phần khác, chúng đều có mục đích là giúp người dùng có thể tùy biến, mở rộng khả năng xử lý cho các thành phần. VD: sau khi dowload xong url, bạn muốn tracking, gửi thông tin ngay lúc đó thì bạn có thể viết phần mở rộng và sửa lại cấu hình để sau khi Dowloader tải xong trang thì sẽ thực hiện việc tracking.

1.2. Luồng dữ liệu

1. Khi bắt đầu crawl một website, Engine sẽ xác định tên miền và tìm vị trí của spider đó và yêu cầu spider đó tìm các urls đầu tiên để crawl
2. Engine nhận danh sách các urls đầu tiên từ spider, gửi cho Scheduler để sắp xếp
3. Engine yêu cầu danh sách cách urls tiếp theo từ Scheduler
4. Engine nhận danh sách các url tiếp theo từ Scheduler vào gửi đến Dowloader (requests)
5. Downloader nhận request và thực hiện việc tải trang, sau khi tải xong sẽ tạo một response và gửi lại Engine
6. Respone từ Dowloader sẽ được Engine đẩy qua Spiders để xử lý
7. Tại Spiders, khi nhận được response, chúng bóc tách thông tin từ response (tilte, content, author, date publish...) và những url có khả năng để crawl và đẩy lại cho Engine (requests)
8. Ở bước này, Engine nhận được kết quả từ Spiders sẽ thực hiện 2 công việc: đẩy những dữ liệu đã được bóc tách tới Item Pipeline để xử lý và lưu vào Databases, đẩy những url mới (requests) mới về Scheduler và quay về bước 3

2. Mô tả tập dữ liệu:

- Nhóm tiến hành crawl dữ liệu từ trang báo điện tử VnExpress về để thực nghiệm

- Link trang báo điện tử: https://vnexpress.net/

- Cấu trúc dataset:

Tên cột	Mô tả	Kiểu dữ liệu
Label	Thể loại của các bài báo	String
Title	Tiêu đề bài báo	String
Content	Nội dung bài báo	String

3. Tiền xử lý dữ liệu

3.1. Hợp nhất dữ liệu:

- Vì thuộc tính Title và Content tương đương nhau nên nhóm tiến hành gộp giá trị hai thuộc tính này thành 1 thuộc tính content để dễ dàng xử lý và phân lớp hơn

3.2. Làm sạch dữ liệu:

- Mục đích bước này là loại bỏ nhiễu trong dữ liệu hiện có. Càng ít nhiễu thì kết quả thu được càng tốt hơn

```
def clean_text(document):

    #Xóa các kí tự đơn lẻ
    document = re.sub(r'\s+[a-zA-Z]\s+', ' ', document)
    #Xóa các kí tự nháy đơn
    document=re.sub(r'\'', ' ', document)
```

```
    #Xóa các kí tự "\"
    document=re.sub(r'\\', ' ', document)
    #Xóa các kí tự đơn lẻ ở đầu văn bản
    document = re.sub(r'^[a-zA-Z]\s+', ' ', document)
    #Thay thế nhiều khoảng trắng đứng cạnh nhau thành 1 khoảng trắng
    document = re.sub(r'\s+', ' ', document, flags=re.I)
    #Chuyển chữ hoa thành chữ thường
    document = document.lower()
    return document
```

3.3. Tách từ:

- Trong tiếng Việt, dấu cách (space) không được sử dụng như 1 kí hiệu phân tách từ, nó chỉ có ý nghĩa phân tách các âm tiết với nhau. Vì thế, để xử lý tiếng Việt, công đoạn tách từ (word segmentation) là 1 trong những bài toán cơ bản và quan trọng bậc nhất.

- Ở đây, nhóm em sử dụng thư viện underthesea* để tiến hành tách từ, lệnh cài đặt:

```
C:\WINDOWS\system32>pip install underthesea
```

```
from underthesea import word_tokenize
def word_tokenizer(document):
    document=word_tokenize(document, format="text")
    return document
```

3.4. Loại bỏ stopwords:

- StopWords là những từ xuất hiện nhiều trong ngôn ngữ tự nhiên, tuy nhiên lại không mang nhiều ý nghĩa. Ở tiếng việt StopWords là những từ như: để, này, kia...

- Có rất nhiều cách để loại bỏ StopWords nhưng có 2 cách chính là:

- Dùng từ điển
- Dựa theo tần suất xuất hiện của từ

- Nhóm em sử dụng từ điển để loại bỏ stopwords 1 cách nhanh chóng và thuận tiện hơn

```
file=pd.read_csv('../TienXuLyDuLieu/vietnamese-stopwords.txt')
stopwords=[]
for i in range(0,len(file)):
    stopwords.append(str(file.values[i,0]))
def clean_stopword(document):
    sent=[]
    for word in document.split(" "):
        if (word not in stopwords) :
            if ("_" in word) or (word.isalpha() == True):
                sent.append(word)
    return sent
```

CHƯƠNG 3: SỬ DỤNG THUẬT TOÁN KHAI THÁC DỮ LIỆU

1. Giới thiệu thư viện sklearn

- Scikit-learn (viết tắt là sklearn) là một thư viện mã nguồn mở dành cho máy học - một ngành trong trí tuệ nhân tạo, rất mạnh mẽ và thông dụng với cộng đồng Python, được thiết kế trên nền Numpy và Scipy. Scikit-learn chứa hầu hết các thuật toán machine learning hiện đại nhất, đi kèm với documentations, luôn được cập nhật.

- Lệnh cài đặt:

```
C:\WINDOWS\system32>pip install numpy
```

```
C:\WINDOWS\system32>pip install scipy
```

```
C:\WINDOWS\system32>pip install sklearn
```

2. Thuật toán TFIDF

2.1. Cơ sở lý thuyết

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. Một vài biến thể của tf-idf thường được sử dụng trong các hệ thống tìm kiếm như một công cụ chính để đánh giá và sắp xếp văn bản dựa vào truy vấn của người dùng.

- 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. Như vậy, term frequency thường được chia cho độ dài văn bản(tổng số từ trong một văn bản).

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

- 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.

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

- 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.

2.2. Lý do lựa chọn thuật toán

- Trong hầu hết các ngôn ngữ, có một số từ có xu hướng xuất hiện thường xuyên như trong tiếng anh có "is", "the"... tương tự tiếng việt có các từ như "là", "của", "cứ"... Chính vì vậy nếu chỉ xét theo tần số xuất hiện của từng từ thì việc phân loại văn bản rất có thể cho kết quả sai dẫn tỷ lệ chính xác sẽ thấp. Vậy giải pháp là gì ?

- Phương pháp phổ biến là sử dụng một phương pháp thống kê có tên là TF-IDF, giá trị TF-IDF của một từ là một con số thu được qua thống kê thể hiện mức độ quan trọng của từ này trong một văn bản, mà bản thân văn bản đang xét nằm trong một tập hợp các văn bản.

2.3. Cài đặt thuật toán

```
from sklearn.feature_extraction.text import TfidfVectorizer
tfidfconverter = TfidfVectorizer(max_features=2000, min_df=5,
max_df=0.7)
X = tfidfconverter.fit_transform(word_data).toarray()
```

3. Thuật toán Naïve Bayes

3.1. Cơ sở lý thuyết

- 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:

P(A|B) = P(B|A)*P(A)/P(B)

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.

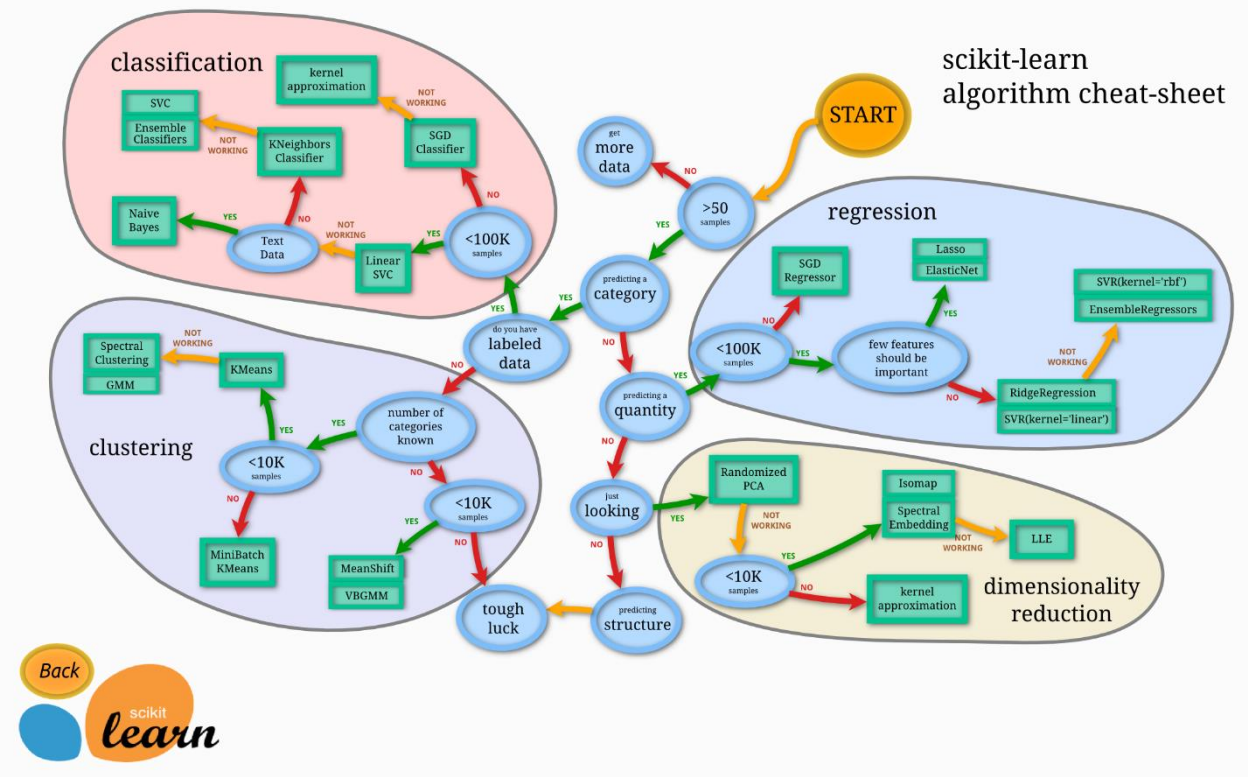
– P(B|A) là xác suất xảy ra B khi biết A xảy ra

– P(A) là xác suất sảy 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.

3.2. Lý do lựa chọn thuật toán

- Naive Bayes Classifiers (NBC) là phương pháp cổ điển nhưng vẫn rất hữu dụng với các bài toán nhất định như phân loại văn bản, email…
- NBC với công thức tính toán đơn giản nên dễ cài đặt (sử dụng thư viện sklearn), thời gian training và test nhanh, phù hợp với bài toán data lớn.



3.3. Cài đặt thuật toán

```
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.3, random_state=0)
#Tạo model dùng thuật toán Naive Bayes
model = GaussianNB()
#Train cho model
model.fit(X_train,y_train)
#Test thử model
y_pred = model.predict(X_test)
#Tính xác suất thuật toán phân lớp Naive Bayes
print(classification_report(y_test,y_pred))
print("Xác suất:",metrics.accuracy_score(y_test, y_pred))
```

CHƯƠNG 4: KẾT QUẢ ĐẠT ĐƯỢC

1. Kết quả đạt được

1.1. Kết quả của thực nghiệm

- Dữ liệu huấn luyện: 8287 mẫu

- Dữ liệu kiểm thử: 3552 mẫu

- Kết quả:

	precision	recall	f1-score	support
Giáo dục	0.74	0.72	0.73	386
Giải trí	0.82	0.77	0.80	417
Khoa học	0.76	0.63	0.69	425
Kinh doanh	0.86	0.79	0.82	469
Pháp luật	0.70	0.86	0.77	385
Sức khỏe	0.78	0.75	0.77	387
Thế giới	0.81	0.76	0.79	387
Thể thao	0.76	0.93	0.83	390
Thời sự	0.63	0.65	0.64	306
micro avg	0.76	0.76	0.76	3552
macro avg	0.76	0.76	0.76	3552
weighted avg	0.77	0.76	0.76	3552

Xác suất: 0.7640765765765766

1.2. Ý nghĩa

- Xây dựng được hệ thống dự đoán được thể loại các bài báo.
- Xây dựng từ điển các từ tiếng Việt để phục vụ tốt hơn.
- Áp dụng thành công 2 thuật toán TF-IDF để sắp xếp văn bản theo tần số và độ quan trọng, NBC để tìm ra những từ xuất hiện và ảnh hưởng nhiều tới thể loại của bài viết.
- Loại bỏ được stopwords.

2. Khó khăn và hạn chế

- Để có được mô hình tốt thì cần một lượng dữ liệu lớn và tin cậy.
- Phải chấp nhận xác suất Bayes, nên có thể một số trường hợp cho ra kết quả không chính xác
- Giao diện ứng dụng còn đơn giản, chỉ sử dụng trên máy tính, không phải là trang web

3. Hướng phát triển

- Có thể áp dụng cho việc tìm những bài báo với thể loại và yêu cầu cần tìm.
- Áp dụng những giải thuật tốt hơn và thay đổi phạm vi crawl không chỉ là các bài báo mà còn là dữ liệu trên mạng xã hội,…
- Phát triển ứng dụng phù hợp với người dùng

PHỤ LỤC 1: BẢNG PHÂN CÔNG CÔNG VIỆC

Công việc	Yến	Đức	Phát
Crawler dữ liệu	X		
Tiền xử lý dữ liệu		X	

Cài đặt thuật toán TFIDF	X		
Cài đặt thuật toán Naïve Bayes			X
Thiết kế + xử lý giao diện	X		
Viết báo cáo		X	X

PHỤ LỤC 2: TÀI LIỆU THAM KHẢO

1. Giáo trình khai phá dữ liệu (Data Mining) – PGS.TS.Đỗ Phúc, NXB Đại học Quốc gia TP.Hồ Chí Minh

2. https://docs.scrapy.org

3. https://scikit-learn.org/stable/documentation.html

4. https://underthesea.readthedocs.io/en/latest/readme.html

5. https://docs.python.org/2/library/tkinter.html