TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN CUỐI KÌ

XỬ LÝ DỮ LIỆU LỚN

Phát hiện tin giả tiếng Việt

Người hướng dẫn: **TS. BÙI THANH HÙNG**

Người thực hiện: **NGUYỄN HOÀNG MINH THƯ – 518H0061**

MẠC THUẬN ĐẠT – 518H0606

Nhóm: **17**

Khóa: **22**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2022

ĐỒ ÁN ĐƯỢC HOÀN THÀNH
TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Chúng tôi xin cam đoan đây là sản phẩm đồ án của riêng chúng tôi và được thực hiện dưới sự hướng dẫn của TS. Bùi Thanh Hùng. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kỳ sự gian lận nào chúng tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình. Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do chúng tôi gây ra trong quá trình thực hiện (nếu có).

TP. Hồ Chí Minh, ngày tháng năm

Tác giả

(ký tên và ghi rõ họ tên)

Nguyễn Hoàng Minh Thư

Mạc Thuận Đạt

LỜI CẢM ƠN

Chúng em xin chân thành cảm ơn thầy Hùng đã tận tâm giảng dạy, chia sẻ các kiến thức, tài liệu tốt thầy thu thập được và đồng thời cũng đã hỗ trợ chúng em rất nhiều trong quá trình thực hành các bài tập trong bộ môn này.

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

Phần xác nhận của GV hướng dẫn

_____

_____

_____

_____

_____

_____

_____

TP. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

Phần đánh giá của GV chấm bài

_____

_____

_____

_____

_____

_____

_____

TP. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

MỤC LỤC

DANH MỤC CÁC CHỮ VIẾT TẮT

CÁC CHỮ VIẾT TẮT

TF-IDF	Term Frequency – Inverse Document Frequency
SVD	Singular Value Decomposition
SVM	Support Vector Machines
LSVM	Linear Support Vector Machines
KNN	K-Nearest Neigbor
DT	Decision Tree
SGD	Stochastic Gradient Descent
XGB	eXtreme Gradient Boosting
LR	Logistic Regression
AUC	Area Under the Curve
LGBM	Light Gradient Boosting Machine
AUC	Area Under ROC Curve

DANH MỤC CÁC HÌNH VẼ

DANH MỤC CÁC BIỂU ĐỒ

1. PHÁT HIỆN TIN GIẢ TIẾNG VIỆT

1.1. Giới thiệu về bài toán

Với các xu hướng công nghệ ngày càng phát triển, các thể loại báo điện tử, tin tức ngày càng được chia sẻ một cách rộng rãi và dễ dàng hơn bao giờ hết thông qua mạng xã hội hay các trang web. Việc phân biệt được giữa một trang báo chính thống và một trang báo "lá cải" đã là một nhiệm vụ không dễ gì đối với người đọc. Nhưng hơn đó, khả năng phân biệt được giữa một tin tức khách quan, báo cáo đúng sự thật và một tin tức giả được dàn dựng để lan truyền thông tin sai sự thật, gây hoang mang cho người đọc là một vấn đề khó hơn rất nhiều.

Do đó, việc phát hiện được tin tức giả là một nhiệm vụ quan trong xã hội ngày nay và đã được nghiên cứu trong nhiều lĩnh vực, chẳng hạn như trong các tài liệu, bài báo khoa học, tin tức và các trang mạng xã hội.

Một ví dụ với Facebook, mạng xã hội với hàng triệu người sử dụng, bất kỳ người dùng nào cũng đều có thể đăng nội dung không được kiểm duyệt lên trang cá nhân của mình và công khai chia sẻ chúng. Như trong thời gian dịch bệnh năm vừa qua tại Việt Nam, đã có vô số các thông tin sai sự thật về dịch bệnh COVID-19 được lan truyền với tốc độ nhanh chóng và rộng rãi đến mức khiến nhiều người tin rằng chính những thông tin giả đó mới là sự thật. Việc này gây ra nhiều sự nhầm lẫn, hoang mang và sai lệch kiến thức cho người đọc.

Vì vậy, một thuật toán có thể tự động nhận diện được thông tin giả chính xác trước khi nó được lan truyền đến người đọc sẽ góp phần rất lớn trong việc xây dựng một không gian mạng lành mạnh cùng với các nội dung xác thực.

1.2. Phân tích yêu cầu của bài toán

1.2.1. Yêu cầu của bài toán

Bài toán phát hiện tin giả tiếng Việt, ta sẽ cần bộ dữ liệu về tin tức bao gồm nội dung, thông số cùng với nhãn của các tin tức đó. Trong đó, nội dung của tin tức là thông

tin chủ yếu cần phải có. Sau đó biến đổi dữ liệu đó thành dạng thích hợp và sử dụng các thuật toán Machine Learning hoặc Deep Learning để phân loại tin tức đó là thật hay giả.

1.2.2. Các phương pháp giải quyết bài toán

Trong nghiên cứu [1], Admed et al. dùng TF-IDF kết hợp với n-gram để trích xuất đặc trưng của nội dung tin tức. Kết quả của nghiên cứu này cho thấy thuật toán LSVM cho độ chính xác (accuracy) cao nhất trong các mô hình được sử dụng với trích xuất đặc trưng TF-IDF với chuỗi 1-gram và 2-gram 10000 đến 50000 giá trị đặc trưng[1].

Trong một nghiên cứu khác [2], Reis et al. sử dụng cách trích xuất đặc trưng về cú pháp, lexical[2], ngữ nghĩa, thái độ, các thông số về tương tác và thông tin của bài viết. Thông qua hai độ đo là AUC và F1, kết quả tốt nhất đạt được từ thuật toán Random Forest và XGB.

Các thuật toán phân loại được sử dụng ở hai nghiên cứu trên là là Support Vector Machines (SVM), Linear Support Vector Machines (LSVM), Logistic Regression (LR), K-Nearest Neigbor (KNN), Decision Tree (DT), Random Forest (RF), Stochastic Gradient Descent (SGD), eXtreme Gradient Boosting (XGB) và Light Gradient Boosting Machine (LGBM).

1.2.3. Phương pháp đề xuất giải quyết bài toán

Do các phương pháp nói trên đã được thực nghiệm và cho kết quả tốt. Chúng em sẽ thực hiện ba phương pháp trích xuất đặc trưng là:

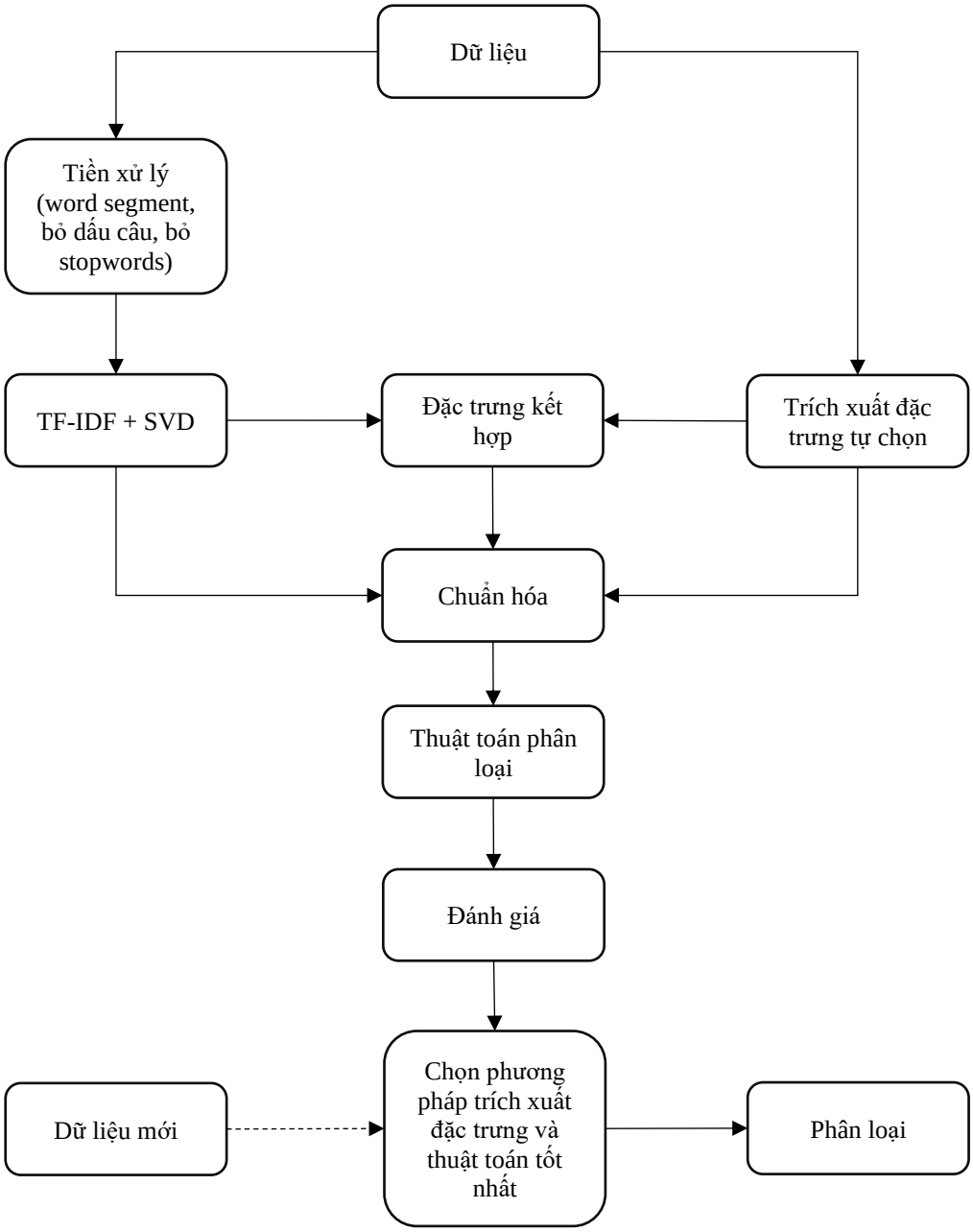
- TF-IDF kết hợp với n-gram
- Các đặc trưng về từ vựng, cú pháp, lexical và các thông số khác
- Kết hợp hai đặc trưng trên

Thử nghiệm trên 7 thuật toán phân loại là: LR, KNN, LSVM, DT, RF, XGB, và LGBM.

[1] Max feature.
[2] Lexical: các đặc trưng về từ vựng. Ví dụ như: số lượng từ, từ khác nhau, dấu câu…

1.3. Phương pháp giải quyết bài toán

1.3.1. Mô hình tổng quát



Hình 1. Mô hình tổng quát qui trình giải quyết bài toán

1.3.2. Đặc trưng của mô hình đề xuất

1.3.2.1. Term Frequency – Inverse Document Frequency (TF-IDF)

TF-IDF là một thuật toán thường được sử dụng để xử lý dữ liệu văn bản trong các bài toán xử lý ngôn ngữ tự nhiên.

Thông thường, trọng số TF-IDF được tính bởi hai giá trị là tần suất xuất hiện của từ ($\boldsymbol{tf}$) và tần suất xuất hiện của từ trong bộ văn bản nghịch đảo (idf).

$$TFIDF(t,d) = tf(t,d) * idf(t)$$

Trong đó, $tf(t,d)$ được tính như sau:

$$tf(t,d) = \frac{số\ lần\ xuất\ hiện\ của\ t\ trong\ d}{tổng\ số\ từ\ trong\ d}$$

và $idf(t)$:

$$idf(t) = 1 + log(\frac{1+n}{1+df(t)})$$

Ký hiệu:

- t là từ đang xét
- d là một văn bản đang xét
- $df(t)$ là số lần xuất hiện của từ đang xét trên toàn bộ các văn bản

1.3.2.2. Singular Value Decomposition (SVD)

Mục đích của SVD được sử dụng trong bài này là để làm giảm chiều dữ liệu của ma trận TF-IDF [3].

1.3.2.3. Logistic Regression (LR)

Logistic regression được sử dụng khi phân loại văn bản dựa trên một bộ đặc tính lớn với kết quả ra kiểu nhị phân(True/False hoặc tin thật/giả). Hàm giả thuyết của Logistic regression là:

$$\mathrm{h_\theta(X)} = \frac{1}{1+e^{-(\beta_0+\beta_1 X)}}$$

Logistic regression sử dụng hàm sigmoid để chuyển kết quả thành một giá trị xác suất; mục tiêu là giảm thiểu hàm cost, hàm được tính như sau:

$$\mathrm{Cost}(h_\theta(x), y) = \begin{cases} \log\left(h_\theta(x)\right), & y = 1, \\ -\log\left(1 - h_\theta(x)\right), & y = 0. \end{cases}$$

1.3.2.4. K-Nearest Neigbor (KNN)

KNN là mô hình học không giám sát sử dụng dữ liệu huấn luyện để quyết định một điểm dữ liệu mới thuộc nhóm điểm dữ liệu nào. Mô hình KNN đo khoảng cách giữa điểm dữ liệu mới với các điểm gần nhất để xác định giá trị của K; nếu giá trị K=1 thì điểm dữ liệu mới chung nhãn với điểm gần nhất. Các phép tính đo khoảng cách là:

$$\text{Khoảng cách Euclidean} = \sqrt{\sum_{i=1}^{k} (x_i - y_i)^2}$$

$$\text{Khoảng cách Manhattan} = \sum_{i=1}^{k} |x_1 - y_i|$$

$$\text{Khoảng cách Minkowski} = \left(\sum_{i=1}^{k} |x_i - y_i|^q \right)^{1/q}$$

1.3.2.5. Linear Support Vector Machines (LSVM)

Support vector machine là mô hình học có giám sát phổ biến dùng để phân loại dữ liệu bằng cách tìm một siêu phẳng (hyperplane) sao cho nó chia bộ dữ liệu thành hai nhóm có đặc tính riêng.

1.3.2.6. Decision Tree (DT)

Decision trees là một mô hình học có giám sát thường được sự dụng để giải bài toán phân loại. Decision trees có cấu trúc cây với mỗi điểm thể hiện một đặc tính dữ liệu, mỗi nhánh thể hiện một quy luật và mỗi lá thể hiện kết quả.

1.3.2.7. Random Forest (RF)

Random forest là dạng cải tiến hơn của decision trees (DT) và cũng là loại mô hình học có giám sát. Random forest gồm nhiều cây decision trees hoạt dộng độc lập để dự đoán nhãn dựa trên kết quả của các cây.

1.3.2.8. eXtreme Gradient Boosting (XGB)

XGBoost là thư viện cho gradient boosting học bằng giải thuật theo kiểu cây, được thiết kế để có hiệu suất cao.

1.3.2.9. Light Gradient Boosting Machine (LGBM)

LightGBM là thư viện cho gradient boosting học bằng giải thuật theo kiểu cây. LightGBM phát triển cây dựa trên leaf-wise, trong khi các giải thuật khác dựa trên level-wise. Leaf-wise giúp giảm nhiều loss hơn giải thật level-wise.

1.4. Thực nghiệm

1.4.1. Dữ liệu

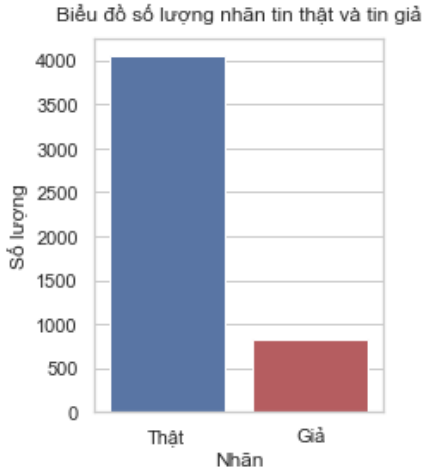
Bài làm sử dụng dữ liệu được chia sẻ qua workshop VLSP2020[3] thông qua cuộc thi ReINTEL[4]. Bộ dữ liệu bao gồm 4372 bản ghi là thông tin bài viết được đăng trên các mạng xã hội người Việt thường dùng như Facebook, Zalo và Lotus. Mỗi bản ghi bao gồm thông tin về người đăng, thời gian, nội dung, lượng tương tác của bài viết và nhãn tương ứng.

	id	user_name	post_message	timestamp_post	num_like_post	num_comment_post	num_share_post	label
0	1	389c669730cb6c54314a46be785cea42	THĂNG CẤP BẬC HÀM ĐỐI VỚI 2 CÁN BỘ, CHIẾN SỸ H...	1585945439	19477	378	173.0	0
1	2	775baa6d037b6d359b229a656eaeaf08	<URL>	1588939166.0	11	5	3	0
2	3	b9f3394d2aff86d85974f5040c401f08	TƯ VẤN MÙA THI: Cách nộp hồ sơ để trúng tuyển ...	1591405213	48	5	19.0	0
3	4	808e278b22ec6b96f2faf7447d10cd8e	Cơ quan Cạnh tranh và Thị trường Anh quyết địn...	1592023613	3	0	0.0	0
4	5	f81bdd6d8be4c5f64bb664214e47aced	Thêm 7 ca tại Quảng Nam liên quan đến hành khá...	1583737358	775	0	54.0	0

Hình 2. Truy xuất 5 dòng đầu của bộ dữ liệu

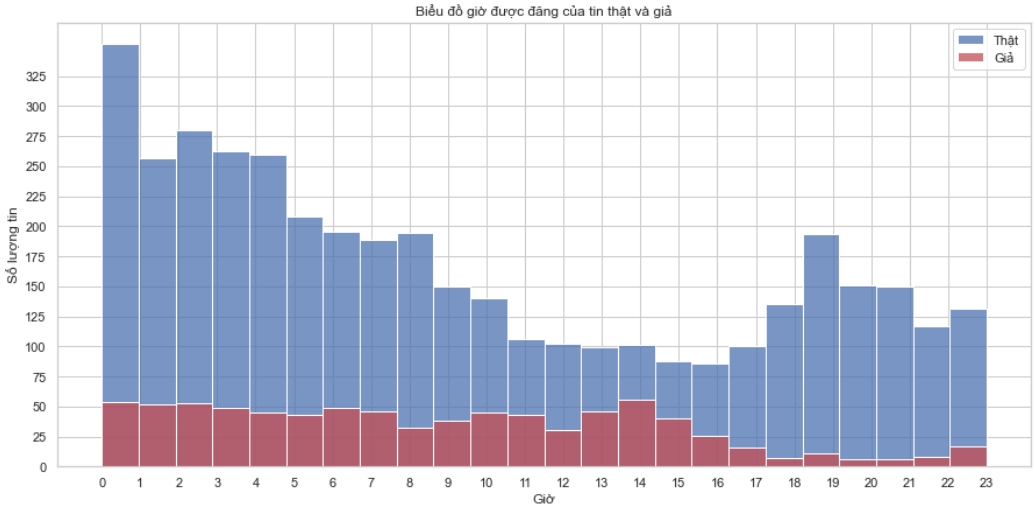
Trong số 4868 dữ liệu sau khi được tiền xử lý cơ bản (xem ở phần 1.4.2), có 4050 dữ liệu với nhãn tin thật và 818 dữ liệu nhãn tin giả, dễ dàng thấy là dữ liệu này khá không cân bằng.

[3] Vietnamese Language and Speech Processing.
[4] Reliable Intelligence Identification on Vietnamese SNSs.



Biểu đồ 1. Số lượng nhãn tin thật và giả

Thông qua *Biểu đồ 2* dưới, ta thấy được thời gian nhiều tin giả được đăng là từ khoảng 0 giờ đến 15 giờ, còn tin thật thì thường được đăng ở hai khoảng là 0 giờ đến 8 giờ và 19 giờ đến 21 giờ.



Biểu đồ 2. Thời gian đăng bài của tin thật và giả

Trong bài toán phát hiện tin giả thì nội dung của nó là một thông tin quan trọng. Wordcloud sẽ biểu diễn các từ thông dụng được dùng trong nội dung của các dữ liệu. Do dữ liệu được thu thập vào năm 2020, đa số các từ thông dụng được dùng cả trong tin giả và thật đều liên quan đến dịch bệnh Covid-19.

Hình 3. Wordcloud nội dung tin thật

Hình 4. Wordcloud nội dung tin giả

1.4.2. Xử lý dữ liệu

Loại bỏ các dữ liệu có nội dung tin tức là rỗng và các dữ liệu trùng lặp, ta có 5172 xuống còn 4868 dữ liệu.

- Đối với trích xuất đặc trưng TF-IDF

Thực hiện tiền xử lý dữ liệu nội dung của tin bằng cách tokenize[5], word segment[6], bỏ tất cả dấu câu và bỏ các token là stopword[7].

```
    # remove stopwords in news text
def clean_stopwords(text_list):
```

[5] Tách từ.
[6] Phân đoạn từ.
[7] Từ dừng: là những từ xuất hiện nhiều trong ngôn ngữ tự nhiên.

```
  ''' sent_list: already tokenized text in list type
  '''
  clean_text = []
  for word in text_list:
    if word not in stopwords:
      clean_text.append(word.lower())
  return clean_text

def tokenize(text):
  text = text.translate(str.maketrans('', '', string.punctuation)) # clean punctuation
  text = word_tokenize(text.lower()) #tokenize, word segment
  return clean_stopwords(text)
```

Sử dụng TF-IDF để trích xuất đặc trưng của nội dung tin. Thực hiện giảm chiều dữ liệu với kỹ thuật SVD để giảm khối lượng tính toán. Cuối cùng thực hiện bước chuẩn hóa dữ liệu với MinMaxScaler.

```
# get tfidf vectors with n_gram = 2
vectorizer = TfidfVectorizer(tokenizer=tokenize, ngram_range=(1, 2), max_features=50000)
X_tfidf = vectorizer.fit_transform(df['post_message'])

# reduce dimensionality of tfidf vector to 100 dimensions as recommended for tfidf
svd = TruncatedSVD(n_components=100, algorithm='arpack')
X_tfidf_svd = svd.fit_transform(X_tfidf)
```

```
# scale data to (0, 1) range as we've performed dimension reduction
scaler1 = MinMaxScaler()
X1 = scaler1.fit_transform(X_tfidf_svd)
```

- Đối với trích xuất đặc trưng tự chọn

Chỉ thực hiện bước tiền xử lý là tokenize, word segment và chuẩn hóa dữ liệu. Bài làm trích xuất 19 các đặc trưng là: số lượng từ, dấu câu, token có 1 từ, token có 2 từ, token có >3 từ, chữ số, stopword, từ viết hoa, từ có chứa 1 chữ cái viết hoa, từ khác nhau, hashtag, lượt thích, bình luận, chia sẻ, giờ, thứ, ngày, tháng và số lượng các đường link có tag là url.

```
# extract custom features
X_cf = np.array([extract_feature(df.iloc[i, :]) for i in range(df.shape[0])])
scaler2 = MinMaxScaler()
X2 = scaler2.fit_transform(X_cf)
```

Ví dụ với một tin có nội dung: *Trong giờ học Thể dục do thầy giáo Nguyễn Văn Quân phụ trách, em D. đã bị thầy Quân "đi đường quyền" lên người dẫn đến việc bị ngất tại trường và sau đó đã được Ban giám hiệu nhà trường đưa đi cấp cứu để điều trị.*

Các đặc trưng được trích xuất sẽ là:

	Total tokens	Punctuation	1 word	2 words	>=3 words	Numeric	Stopwords	CAPITAL	1 CAPITAL	Distinct	Hashtags	Likes	Comments	Shares	Hour	Weekday	Month	Day	URL
0	52	2	46	4	2	0	22	1	6	46	0	2	1	0	4	1	5	26	0

Hình 5. Ví dụ trích xuất đặc trưng tự chọn

- Đối với đặc trưng kết hợp

Kết hợp dữ liệu trước khi chuẩn hóa của hai đặc trưng kể trên và cũng chuẩn hóa sử dụng một scaler khác.

```
# combine tf-idf and custom features
X_cb = np.append(X_tfidf_svd, X_cf, axis=1)
```

```
scaler3 = MinMaxScaler()
X3 = scaler3.fit_transform(X_cb)
```

Chia các dữ liệu của mỗi đặc trưng thành 2 tập train và test với tỉ lệ 8/2.

```
X1_train, X1_test, y_train, y_test = train_test_split(X1, y, test_size=0.2, random_state=42)
X2_train, X2_test, y_train, y_test = train_test_split(X2, y, test_size=0.2, random_state=42)
X3_train, X3_test, y_train, y_test = train_test_split(X3, y, test_size=0.2, random_state=42)
```

Sau khi chia dữ liệu ta có tập train với 3894 dữ liệu (3241 thật và 653 giả), tập test có 974 (809 thật và 165 giả).

1.4.3. Công nghệ sử dụng

- Ngôn ngữ lập trình sử dụng: Python phiên bản 3.9.7.
- Các thư viện sử dụng: numpy, pandas, vncorenlp và sklearn.
- Công cụ sử dụng: Jupyter Notebook.

1.4.4. Cách đánh giá

Kết quả phân loại sẽ được đánh giá bằng hai độ đo là **Accuracy** và **AUC** (Area Under ROC Curve).

- Accuracy là giá trị thể hiện độ chính xác với số trường hợp dữ liệu được phân loại đúng nhãn trên tổng số dữ liệu.
- Độ đo AUC

AUC có giá trị từ khoảng 0 đến 1, điểm AUC càng cao thì thuật toán có khả năng phân biệt càng tốt. Với điểm AUC là 0.5, nó thể hiện rằng thuật toán không thể phân biệt được các sự khác nhau giữa hai nhãn. Và khi AUC là 0 thì cho thấy là thuật toán phân biệt tất cả các nhãn thật thành giả hoặc tất cả các nhãn giả thành thật.

$$AUC = \frac{\sum(n_0 + n_1 + 1 - r_i) - n_0(n_0 + 1)/2}{n_0 n_1}$$

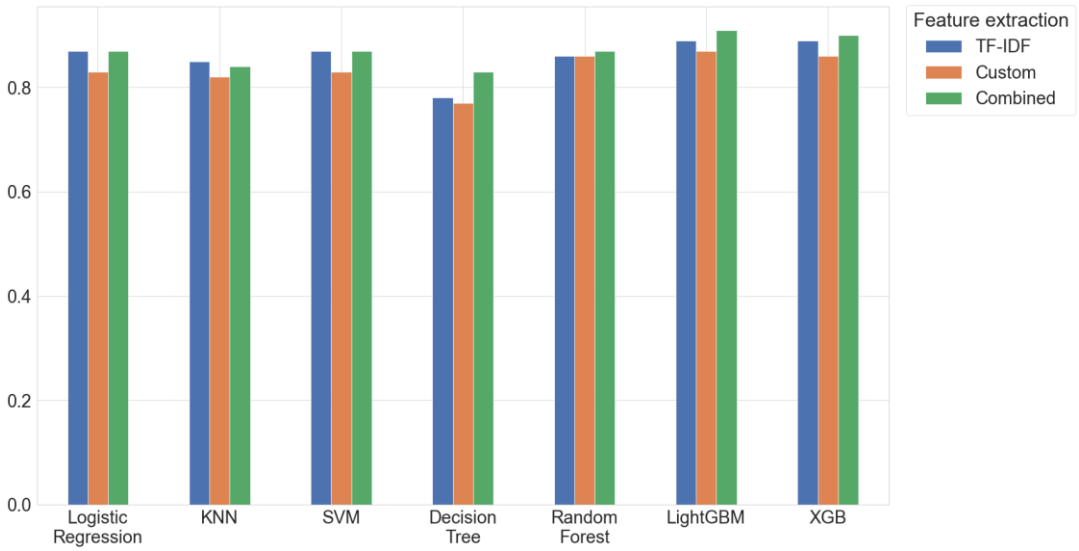
trong đó, r_i là rank (hạng) của tin giả thứ i và n_0 và n_1 là số lượng tin thật và giả.

1.5. Kết quả đạt được

Qua huấn luyện 3 trích xuất đặc trưng với 7 thuật toán khác nhau, ta có bảng độ đo accuracy và AUC như sau:

	TF-IDF	Custom	Combined
Logistic Regression	0.87	0.83	0.87
KNN	0.85	0.82	0.84
SVM	0.87	0.83	0.87
Decision Tree	0.78	0.77	0.83
Random Forest	0.86	0.86	0.87
LightGBM	0.89	0.87	0.91
XGB	0.89	0.86	0.90

Hình 6. Bảng so sánh accuracy



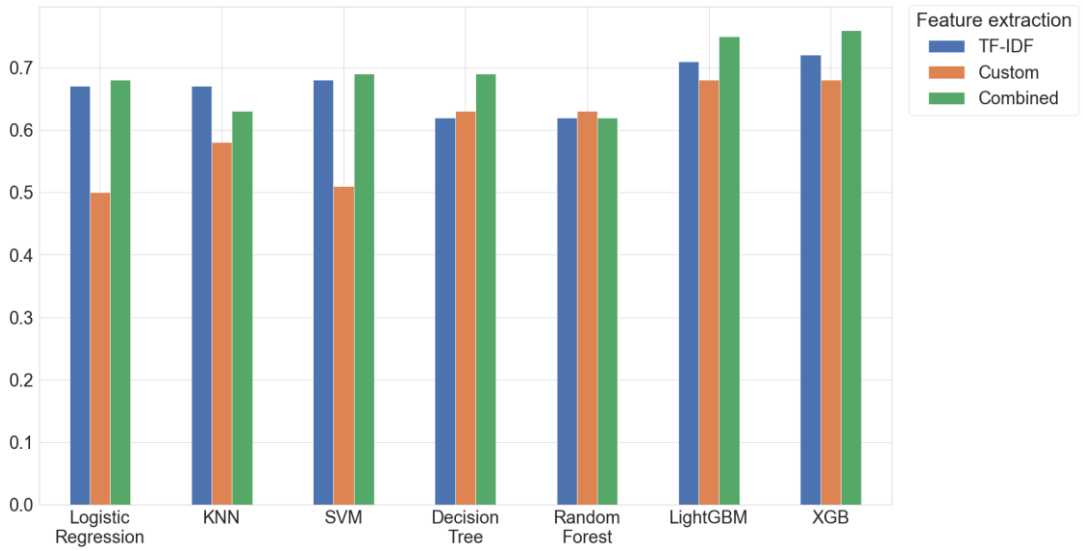
Biểu đồ 3. Biểu đồ so sánh accuracy

Mặc dù accuracy khá cao nhưng cũng chưa thể chứng minh được là thuật toán có hoạt động tốt hay không. Do dữ liệu không cân bằng giữa hai nhãn, ta có tập test với 809 dữ liệu nhãn thật và 165 là giả. Ví dụ trong trường hợp thuật toán phân loại tất cả các dữ liệu test là nhãn thật, thì ta vẫn sẽ có được accuracy là 0.83.

Để có thể đánh giá thêm ở một góc nhìn tốt hơn, ta dùng độ đo AUC để đánh giá khả năng phân loại giữa nhãn thật và giả của thuật toán.

	TF-IDF	Custom	Combined
Logistic Regression	0.67	0.50	0.68
KNN	0.67	0.58	0.63
SVM	0.68	0.51	0.69
Decision Tree	0.62	0.63	0.69
Random Forest	0.62	0.63	0.62
LightGBM	0.71	0.68	0.75
XGB	0.72	0.68	0.76

Hình 7. Bảng so sánh độ đo AUC



Biểu đồ 4. Biểu đồ so sánh độ đo AUC

Hai thuật toán LightGBM và XGB có điểm AUC cao nhất trong cả 3 cách trích xuất đặc trưng giữa 7 thuật toán này.

Áp dụng thuật toán XGB để phát hiện một tin tức giả.

```
# choose xgb as it shows the best performance with both 2 scores amongst
def detect(data):
  # TF-IDF feature
  tfidf = vectorizer.transform([' '.join(tokenize(data[2]))])
  tfidf_svd = svd.transform(tfidf)
  cf = extract_feature(data)
  feature = scaler3.transform([np.append(tfidf_svd, cf)])
  pred = model3_lgbm.predict(feature)
  return pred
```

Kết quả dự đoán là:

```
Máy bay Vietnam Airlines bị dọa bắn hạ trên vịnh Tokyo. Một máy bay của Vietnam Airlines đã phải chuyển hướng hạ cánh sau
cuộc điện thoại có nội dung đe dọa bắn hạ máy bay trên vịnh Tokyo.

Nhãn dự đoán là: [1]
```

1.6. Kết luận

Mặc dù trung bình điểm cả hai độ đo qua đánh giá chưa được tốt nhưng nó vẫn cho thấy được là việc kết hợp hai trích xuất đặc trưng TF-IDF và các đặc trưng tùy chọn với nhau có cải thiện khả năng phân loại của thuật toán lên một mức độ nhất định.

1.7. Hướng phát triển

Ta có thể thử nghiệm thêm ở phần trích xuất đặc trưng bằng cách kết hợp với các phương pháp khác như Word2Vec, GloVe, BERT... Hoặc sử dụng các neural network làm mô hình phân loại cho bài toán.

TÀI LIỆU THAM KHẢO

1. Ahmed, H., Traore, I., & Saad, S. (2017). Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques. Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments, 127–138. doi:10.1007/978-3-319-69155-8_9
   Link
2. Reis, J. C. S., Correia, A., Murai, F., Veloso, A., Benevenuto, F., & Cambria, E. (2019). Supervised Learning for Fake News Detection. IEEE Intelligent Systems, 34(2), 76–81. doi:10.1109/mis.2019.2899143
   Link
3. Kadhim, Ammar & Cheah, Yu-N & Hieder, Inaam & Ali, Rawaa. (2017). Improving TF-IDF with Singular Value Decomposition (SVD) for Feature Extraction on Twitter. 10.23918/iec2017.16.
   Link
4. Iftikhar Ahmad, Muhammad Yousaf, Suhail Yousaf, Muhammad Ovais Ahmad, "Fake News Detection Using Machine Learning Ensemble Methods", Complexity, vol. 2020, Article ID 8885861, 11 pages, 2020.
   Link

5. N. -D. Pham, T. -H. Le, T. -D. Do, T. -T. Vuong, T. -H. Vuong and Q. -T. Ha, "Vietnamese Fake News Detection Based on Hybrid Transfer Learning Model and TF-IDF," 2021 13th International Conference on Knowledge and Systems Engineering (KSE), 2021, pp. 1-6, doi: 10.1109/KSE53942.2021.9648676.
   Link
6. Aldwairi, M., & Alwahedi, A. (2018). Detecting Fake News in Social Media Networks. Procedia Computer Science, 141, 215–222. doi:10.1016/j.procs.2018.10.171
   Link
7. Jiawei, Zhang & Dong, Bowen & Yu, Philip. (2020). FakeDetector: Effective Fake News Detection with Deep Diffusive Neural Network. 1826-1829. 10.1109/ICDE48307.2020.00180.
   Link

TỰ ĐÁNH GIÁ

(Với nhóm có 2 thành viên)

Câu	Nội dung	Điểm chuẩn	Tự chấm	Ghi chú
1 (8.5)	**1.1. Giới thiệu về bài toán**	0.5	**0.5**	
	1.2. Phân tích yêu cầu của bài toán	1.0	**1.0**	
	1.3. Phương pháp giải quyết bài toán	1.5	**1.0**	
	1.4. Thực nghiệm	4.0	**3.5**	
	1.5. Kết quả đạt được	1.0	**0.5**	
	1.6. Kết luận	0.5	**0.5**	
2	**Điểm nhóm**	0.5	**0.5**	
3	**Báo cáo (**chú ý các chú ý 2,3,4,6 ở trang trước, nếu sai sẽ bị trừ đ iểm nặng**)**	1.0	**1.0**	
Tổng điểm			**8.5**	