

Project_3

January 9, 2022

0.1 Link deploy

Demo cho chương trình: [Link](#)

Bảng phân công công việc

MSSV	Họ tên	Công việc
19120212	Vũ Công Duy	Mô hình hóa
19120297	Đoàn Việt Nam	Tiền xử lý dữ liệu và bonus point
19120328	Võ Trọng Phú	Khám phá dữ liệu
19120389	Tô Gia Thuận	Deploy mô hình
19120602	Hồ Hữu Ngọc	Tiền xử lý dữ liệu

```
[ ]: !pip install vncorenlp
```

```
Requirement already satisfied: vncorenlp in /home/tgt/miniconda3/envs/min_ds-  
env/lib/python3.8/site-packages (1.0.3)  
Requirement already satisfied: requests in /home/tgt/miniconda3/envs/min_ds-  
env/lib/python3.8/site-packages (from vncorenlp) (2.25.1)  
Requirement already satisfied: certifi>=2017.4.17 in  
/home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from  
requests->vncorenlp) (2021.10.8)  
Requirement already satisfied: urllib3<1.27,>=1.21.1 in  
/home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from  
requests->vncorenlp) (1.26.7)  
Requirement already satisfied: chardet<5,>=3.0.2 in  
/home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from  
requests->vncorenlp) (4.0.0)  
Requirement already satisfied: idna<3,>=2.5 in /home/tgt/miniconda3/envs/min_ds-  
env/lib/python3.8/site-packages (from requests->vncorenlp) (2.10)
```

```
[ ]: !pip install wordcloud
```

```
Requirement already satisfied: wordcloud in /home/tgt/miniconda3/envs/min_ds-  
env/lib/python3.8/site-packages (1.8.1)  
Requirement already satisfied: matplotlib in /home/tgt/miniconda3/envs/min_ds-  
env/lib/python3.8/site-packages (from wordcloud) (3.4.1)  
Requirement already satisfied: numpy>=1.6.1 in /home/tgt/miniconda3/envs/min_ds-  
env/lib/python3.8/site-packages (from wordcloud) (1.20.2)
```

Requirement already satisfied: pillow in /home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from wordcloud) (8.4.0)
Requirement already satisfied: pyparsing>=2.2.1 in /home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from matplotlib->wordcloud) (3.0.6)
Requirement already satisfied: python-dateutil>=2.7 in /home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from matplotlib->wordcloud) (2.8.2)
Requirement already satisfied: kiwisolver>=1.0.1 in /home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from matplotlib->wordcloud) (1.3.2)
Requirement already satisfied: cycler>=0.10 in /home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from matplotlib->wordcloud) (0.11.0)
Requirement already satisfied: six>=1.5 in /home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from python-dateutil>=2.7->matplotlib->wordcloud) (1.16.0)

```
[ ]: !pip install nltk
```

Requirement already satisfied: nltk in /home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (3.6.7)
Requirement already satisfied: regex>=2021.8.3 in /home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from nltk) (2021.11.10)
Requirement already satisfied: tqdm in /home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from nltk) (4.62.3)
Requirement already satisfied: click in /home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from nltk) (7.1.2)
Requirement already satisfied: joblib in /home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from nltk) (1.1.0)

```
[ ]: !pip install tensorflow
```

Requirement already satisfied: tensorflow in /home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (2.7.0)
Requirement already satisfied: keras-preprocessing>=1.1.1 in /home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from tensorflow) (1.1.2)
Requirement already satisfied: numpy>=1.14.5 in /home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from tensorflow) (1.20.2)
Requirement already satisfied: flatbuffers<3.0,>=1.12 in /home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from tensorflow) (2.0)
Requirement already satisfied: opt-einsum>=2.3.2 in /home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from tensorflow) (3.3.0)
Requirement already satisfied: libclang>=9.0.1 in /home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from

tensorflow) (12.0.0)
Requirement already satisfied: protobuf>=3.9.2 in
/home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from
tensorflow) (3.19.1)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.21.0 in
/home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from
tensorflow) (0.23.1)
Requirement already satisfied: gast<0.5.0,>=0.2.1 in
/home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from
tensorflow) (0.4.0)
Requirement already satisfied: wheel<1.0,>=0.32.0 in
/home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from
tensorflow) (0.37.0)
Requirement already satisfied: typing-extensions>=3.6.6 in
/home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from
tensorflow) (4.0.0)
Requirement already satisfied: six>=1.12.0 in /home/tgt/miniconda3/envs/min_ds-
env/lib/python3.8/site-packages (from tensorflow) (1.16.0)
Requirement already satisfied: h5py>=2.9.0 in /home/tgt/miniconda3/envs/min_ds-
env/lib/python3.8/site-packages (from tensorflow) (3.6.0)
Requirement already satisfied: astunparse>=1.6.0 in
/home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from
tensorflow) (1.6.3)
Requirement already satisfied: absl-py>=0.4.0 in
/home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from
tensorflow) (1.0.0)
Requirement already satisfied: tensorflow-estimator<2.8,~=2.7.0rc0 in
/home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from
tensorflow) (2.7.0)
Requirement already satisfied: google-pasta>=0.1.1 in
/home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from
tensorflow) (0.2.0)
Requirement already satisfied: keras<2.8,>=2.7.0rc0 in
/home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from
tensorflow) (2.7.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in
/home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from
tensorflow) (1.43.0)
Requirement already satisfied: termcolor>=1.1.0 in
/home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from
tensorflow) (1.1.0)
Requirement already satisfied: tensorboard~=2.6 in
/home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from
tensorflow) (2.7.0)
Requirement already satisfied: wrapt>=1.11.0 in
/home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from
tensorflow) (1.13.3)
Requirement already satisfied: requests<3,>=2.21.0 in

/home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from
 tensorboard~=2.6->tensorflow) (2.25.1)
 Requirement already satisfied: google-auth<3,>=1.6.3 in
 /home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from
 tensorboard~=2.6->tensorflow) (2.3.3)
 Requirement already satisfied: markdown>=2.6.8 in
 /home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from
 tensorboard~=2.6->tensorflow) (3.3.6)
 Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in
 /home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from
 tensorboard~=2.6->tensorflow) (0.6.1)
 Requirement already satisfied: werkzeug>=0.11.15 in
 /home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from
 tensorboard~=2.6->tensorflow) (2.0.2)
 Requirement already satisfied: setuptools>=41.0.0 in
 /home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from
 tensorboard~=2.6->tensorflow) (59.1.1)
 Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in
 /home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from
 tensorboard~=2.6->tensorflow) (0.4.6)
 Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in
 /home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from
 tensorboard~=2.6->tensorflow) (1.8.1)
 Requirement already satisfied: cachetools<5.0,>=2.0.0 in
 /home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from google-
 auth<3,>=1.6.3->tensorboard~=2.6->tensorflow) (4.2.4)
 Requirement already satisfied: pyasn1-modules>=0.2.1 in
 /home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from google-
 auth<3,>=1.6.3->tensorboard~=2.6->tensorflow) (0.2.8)
 Requirement already satisfied: rsa<5,>=3.1.4 in
 /home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from google-
 auth<3,>=1.6.3->tensorboard~=2.6->tensorflow) (4.8)
 Requirement already satisfied: requests-oauthlib>=0.7.0 in
 /home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from google-
 auth-oauthlib<0.5,>=0.4.1->tensorboard~=2.6->tensorflow) (1.3.0)
 Requirement already satisfied: importlib-metadata>=4.4 in
 /home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from
 markdown>=2.6.8->tensorboard~=2.6->tensorflow) (4.8.2)
 Requirement already satisfied: chardet<5,>=3.0.2 in
 /home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from
 requests<3,>=2.21.0->tensorboard~=2.6->tensorflow) (4.0.0)
 Requirement already satisfied: certifi>=2017.4.17 in
 /home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from
 requests<3,>=2.21.0->tensorboard~=2.6->tensorflow) (2021.10.8)
 Requirement already satisfied: urllib3<1.27,>=1.21.1 in
 /home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from
 requests<3,>=2.21.0->tensorboard~=2.6->tensorflow) (1.26.7)
 Requirement already satisfied: idna<3,>=2.5 in /home/tgt/miniconda3/envs/min_ds-

env/lib/python3.8/site-packages (from requests<3,>=2.21.0->tensorboard~=2.6->tensorflow) (2.10)
Requirement already satisfied: zipp>=0.5 in /home/tgt/.local/lib/python3.8/site-packages (from importlib-metadata>=4.4->markdown>=2.6.8->tensorboard~=2.6->tensorflow) (3.6.0)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in /home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from pyasn1-modules>=0.2.1->google-auth<3,>=1.6.3->tensorboard~=2.6->tensorflow) (0.4.8)
Requirement already satisfied: oauthlib>=3.0.0 in /home/tgt/miniconda3/envs/min_ds-env/lib/python3.8/site-packages (from requests-oauthlib>=0.7.0->google-auth-oauthlib<0.5,>=0.4.1->tensorboard~=2.6->tensorflow) (3.1.1)

```
[ ]: import pandas as pd
import numpy as np
import vncorenlp
from vncorenlp import VnCoreNLP
import re
import seaborn as sns
import copy
import pickle
from sklearn.pipeline import make_pipeline
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.linear_model import LogisticRegression
import sklearn.ensemble
import sklearn.naive_bayes
import sklearn.svm
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
from sklearn.metrics import plot_confusion_matrix
import wordcloud
from wordcloud import WordCloud
import nltk
from nltk import tokenize
import os
import tensorflow as tf
# tf.get_logger().setLevel('INFO')
```

2022-01-09 13:39:52.960256: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'libcudart.so.11.0'; dlderror: libcudart.so.11.0: cannot open shared object file: No such file or directory
2022-01-09 13:39:52.960270: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.

0.2 Đọc data, stopwords, thư viện java

```
[ ]: data=pd.read_csv("vn_news_223_tdlfr.csv")
      f = open('resources/stopwords.txt', 'r', encoding='UTF-8')
      stopwords = f.read().split('\n')
      annotator = VnCoreNLP("resources/VnCoreNLP-1.1.1.jar",
                             ↪annotators="wseg,pos,ner,parse", max_heap_size='-Xmx2g')
      data.head()
```

```
[ ]:
0 Thủ tướng Abe cúi đầu xin lỗi vì hành động phi...      binhluan.biz
1 Thủ tướng Nhật cúi đầu xin lỗi vì tinh thần ph...      www.ipick.vn
2 Choáng! Cơ trưởng đeo khăn quàng quấy banh nóc...      tintucqpvn.net
3 Chưa bao giờ nhạc Kpop lại dễ hát đến thế!!!\n...      tintucqpvn.net
4 Đại học Hutech sẽ áp dụng cải cách "Tiếng Việt"...    www.gioitreviet.net

label
0      1
1      1
2      1
3      1
4      1
```

1 Tiền xử lý dữ liệu (Data preprocessing)

1.1 1. Xử lý noise

Các noise trong văn bản bao gồm các link, dấu xuống dòng, các kí tự đặc biệt, viết hoa sẽ được xử lý

```
[ ]: def NoiseDefuse(s):
    result = copy.copy(s)
    result = result.str.lower()
    result = result.apply(lambda x: re.sub(r'http\S+', '', x))
    result = result.apply(lambda x: x.replace('\n', ' '))
    result = result.apply(lambda x: re.sub('[^aâãäåäåäåäåäåäåäåä b c d e f g h i j k l m n o p q r s t u v w x y z 0-9]+', '', x))
    return result
```

1.2 2. TokenNize

```
[ ]: #Giảm chiều dữ liệu
def reduce_dim(x):
    return x[0]

#Tách từ
def TokenNize(s):
```

```
return s.apply(annotator.tokenize).apply(reduce_dim)
```

1.3 3.Normalize

```
[ ]: #Hàm chuyển các từ viết tắt về dạng đầy đủ
def normalized1(x):
    contractions={
        'cđv': 'cổ động viên',
        'thcs': 'trung học cơ sở',
        'pgs': 'phó giáo sư ',
        'gs': 'giáo sư ',
        'ts': 'tiến sĩ ',
        'gd đt': 'giáo dục - đào tạo',
        'gd dt': 'giáo dục - đào tạo',
        'gdđt': 'giáo dục - đào tạo',
        'hlv': 'huấn luyện viên',
        'tp': 'thành phố ',
        'hcm': 'Hồ Chí Minh ',
        'đt': 'đội tuyển ',
        'gd': 'giáo dục '
    }
    for k,v in contractions.items():
        x=x.replace(k,v)
    return x
def normalized(s):
    return s.apply(normalized1)
```

1.4 4. Remove stopwords

```
[ ]: #Hàm loại bỏ các stopwords
def remove_stopword(list_word):
    clean_list = []
    for i in range(len(list_word)):
        temp=list_word[i].replace('_', ' ')
        if temp not in stopwords :
            clean_list.append(list_word[i])
    return clean_list
```

1.5 5. Tổng hợp

```
[ ]: def Preprocess(s):
    a= TokenNize(normalized(NoiseDefuse(s)))
    a=a.apply(remove_stopword)
    return a
data['clean_text']= Preprocess(data['text'])
data
```

```
[ ]:                                     text                domain \
0    Thủ tướng Abe cúi đầu xin lỗi vì hành động phi...    binhluan.biz
1    Thủ tướng Nhật cúi đầu xin lỗi vì tinh thần ph...    www.ipick.vn
2    Choáng! Cơ trưởng đeo khăn quàng quấy banh nóc...    tintucqpvnet
3    Chưa bao giờ nhạc Kpop lại dễ hát đến thế!!!\n...    tintucqpvnet
4    Đại học Hutech sẽ áp dụng cải cách "Tiểu Việt"...    www.gioitreviet.net
..    ..    ..
218  "Siêu máy bay" A350 sẽ chở CĐV Việt Nam đi Mal...    dantri.com.vn
219  Thuởng 20.000 USD cho đội tuyển cờ vua Việt Na...    dantri.com.vn
220  Trường Sơn giành HCV tại giải cờ vua đồng đội ...    dantri.com.vn
221  Chuyện về chàng sinh viên Luật - Kiện tướng Lê...    dantri.com.vn
222  Tiền đạo Malaysia: "Tôi đã có cách vượt qua hà...    dantri.com.vn

      label                clean_text
0      1  [thủ_tướng, abe, cúi, đầu, xin_lỗi, hành_động,...
1      1  [thủ_tướng, nhật, cúi, đầu, xin_lỗi, tinh_thần...
2      1  [choáng, cơ_trưởng, đeo, khăn_quàng, quấy, ban...
3      1  [nhạc, kpop, hát, giáo_sư, nguyễn_lân, dũng, t...
4      1  [đại_học, hutech, áp_dụng, cải_cách, tiểu, việ...
..    ...    ..
218    0  [siêu, máy_bay, a350, chở, cổ_động_viên, việt_...
219    0  [thuởng, 20000, usd, đội_tuyển, cờ_vua, việt_n...
220    0  [trường_sơn, giành, hcv, giải, cờ_vua, đồng_độ...
221    0  [chàng, sinh_viên, luật, kiện_tướng, lê_tuấn, ...
222    0  [tiền_đạo, malaysia, hàng, thủ, đội_tuyển, việ...

[223 rows x 4 columns]
```

2 Khám phá dữ liệu

2.1 Dữ liệu có bị thiếu không ? Có cột nào sai kiểu dữ liệu không ?

```
[ ]: data.isnull().values.any()
      #data.dropna()
```

```
[ ]: False
```

Vậy là dữ liệu không có giá trị thiếu

```
[ ]: data.dtypes
```

```
[ ]: text                object
      domain            object
      label             int64
      clean_text        object
      dtype: object
```

Các cột dữ liệu đều có giá trị đúng

2.2 Các text có kích thước như thế nào ?

2.2.1 Mô tả kích thước

```
[ ]: copy_data = data
copy_data['text_length'] = [len(a) for a in copy_data['text']]
copy_data['text_length'].describe()
```

```
[ ]: count      223.000000
mean      2539.771300
std      1780.532008
min       309.000000
25%      1286.000000
50%      2009.000000
75%      3300.500000
max      10019.000000
Name: text_length, dtype: float64
```

2.2.2 Kích thước trung bình của mỗi text

```
[ ]: copy_data['text_length'].mean()
```

```
[ ]: 2539.7713004484303
```

2.3 Mỗi miền báo có bao nhiêu bài viết ?

2.3.1 Ta có thể in ra số lượng text của từng domain

```
[ ]: domain_count = data.groupby(['domain'])['text'].count()
domain_count = domain_count.sort_values(ascending=False,)
domain_count
```

```
[ ]: domain
vnexpress.net      73
tinvn.info        68
dantri.com.vn     14
thethao.tuoitre.vn 10
thoibao.today      6
news.zing.vn       6
tuoitre.vn         5
tintucqpvnet      5
thanhnien.vn       5
kinhdoanh.vnexpress.net 3
phapluat.news      3
thegioitre.vn      2
giaitri.vnexpress.net 2
giadinhtiepthi.com 2
doisong.vnexpress.net 2
thoibao.de         2
```

www.ipick.vn	2
www.gioitreviet.net	1
autoxe.net	1
sohoa.vnexpress.net	1
suckhoe.vnexpress.net	1
sorry.vn	1
baoangiang.com.vn	1
laodong.vn	1
https://news.zing.vn	1
haiduong.tintuc.vn	1
dulich.vnexpress.net	1
binhluan.biz	1
baonuocmy.com	1
www.vietgiaitri.com/	1

Name: text, dtype: int64

2.3.2 Miền báo có nhiều bài nhất

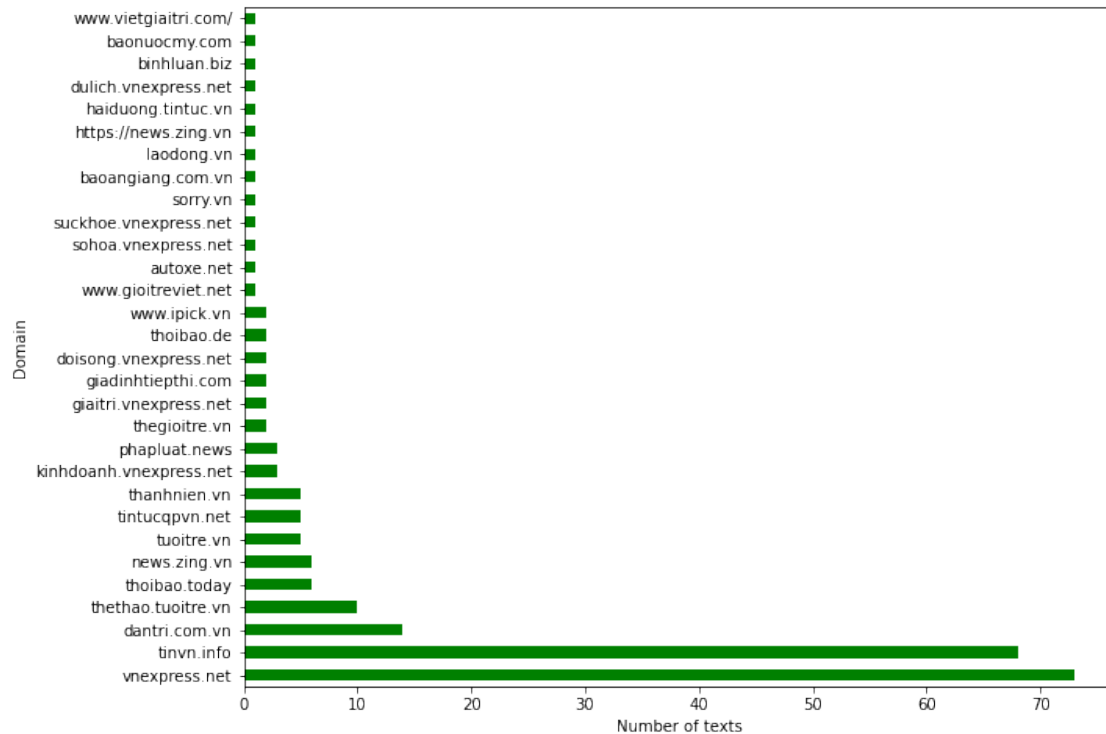
```
[ ]: print("Miền báo có nhiều bài nhất:")
      domain_count.idxmax()
```

Miền báo có nhiều bài nhất:

```
[ ]: 'vnexpress.net'
```

2.3.3 Hmm, sử dụng biểu đồ bar chart để nhìn tổng quát hơn

```
[ ]: domain_count.plot(kind="barh", color = "green", figsize = (10,8))
      plt.xlabel("Number of texts")
      plt.ylabel("Domain")
      plt.show()
```



2.4 Tin giả và tin thật

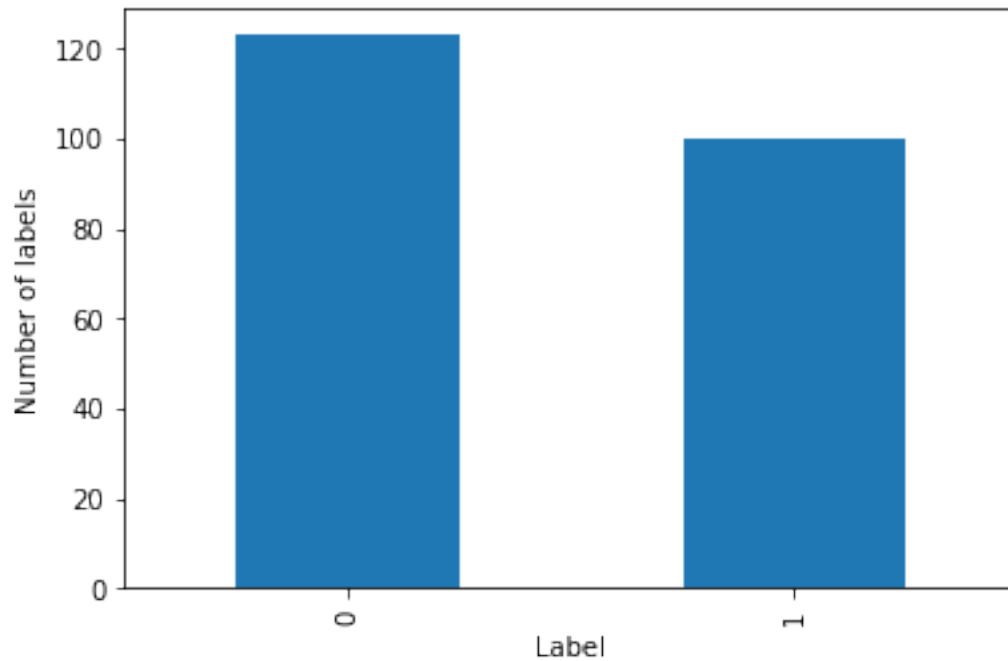
2.4.1 Ta có thể in ra số lượng tin giả (ứng với 1) và tin thật (ứng với 0)

```
[ ]: label_count = data.groupby(['label'])['text'].count()
label_count
```

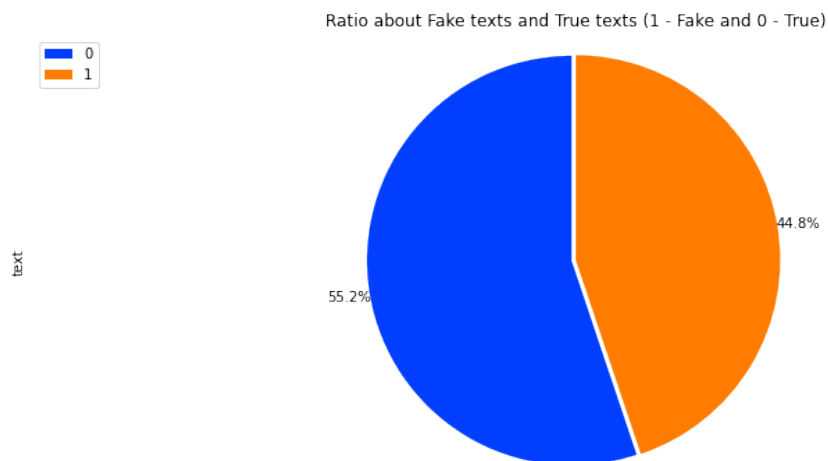
```
[ ]: label
0    123
1    100
Name: text, dtype: int64
```

2.4.2 Một vài biểu đồ

```
[ ]: label_count.plot(kind="bar")
plt.xlabel("Label")
plt.ylabel("Number of labels")
plt.show()
```



```
[ ]: explode_list=[0,0.02]
label_count.plot(kind='pie',figsize=(14,6),autopct="%1.1f%%", startangle=90,
    ↳labels=None, pctdistance=1.1
    , explode=explode_list, colors = sns.
    ↳color_palette('bright')[0:2])
plt.title("Ratio about Fake texts and True texts (1 - Fake and 0 - True)")
plt.axis('equal')
plt.legend(labels = label_count.index, loc='upper left');
```



Tuy có vẻ chênh lệch không nhiều nhưng tin thật vẫn có số lượng lớn hơn tin giả.

Mật độ của tin giả và tin thật khá đều nhau.

2.5 Những từ khóa xuất hiện nhiều nhất

2.5.1 Đối với các tin thật (label = 0) thì những từ khóa nào xuất hiện nhiều nhất trong các bài viết ?

Mình sẽ lấy ra danh sách 20 từ được dùng nhiều nhất trong bộ tin thật

```
[ ]: true_data = data[data["label"] == 0]
true_token_space = tokenize.WhitespaceTokenizer()
true_data_text = true_data['clean_text'].explode()
true_data_text = ' '.join(true_data_text)
true_token_phrase = true_token_space.tokenize(true_data_text)
true_count = nltk.FreqDist(true_token_phrase)
true_data_df = pd.DataFrame({"Word": list(true_count.keys()), "Number of words":
    ↪ list(true_count.values())})
true_data_df = true_data_df.nlargest(columns = "Number of words", n = 20)
```

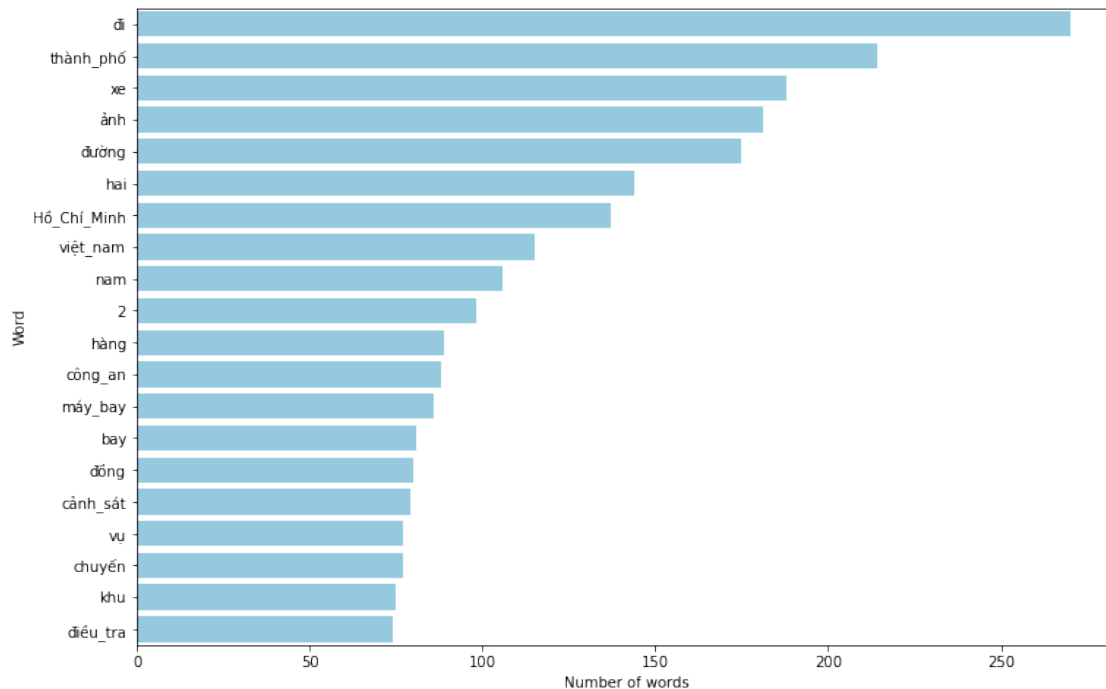
```
[ ]: true_data_df
```

```
[ ]:
      Word  Number of words
217      đi                270
294  thành_phố            214
39       xe               188
58      ảnh               181
30     đường             175
51      hai              144
295  Hồ_Chí_Minh          137
148   việt_nam           115
271      nam             106
101      2                98
252     hàng              89
141   công_an             88
1257  máy_bay             86
293     bay               81
288     đồng              80
1316  cảnh_sát            79
37      vụ               77
1258   chuyển            77
1041    khu              75
155   điều_tra           74
```

Dùng bar chart để nhìn rõ nào

```
[ ]: plt.figure(figsize=(12,8))
```

```
ax = sns.barplot(data = true_data_df, x = "Number of words", y = "Word", color_↵
↵ = 'skyblue')
plt.show()
```



Từ “đi” được dùng nhiều nhất với hơn 250 lần (cụ thể là 270)

2.5.2 Đối với các tin giả (label = 1) thì những từ khóa nào xuất hiện nhiều nhất trong các bài viết ?

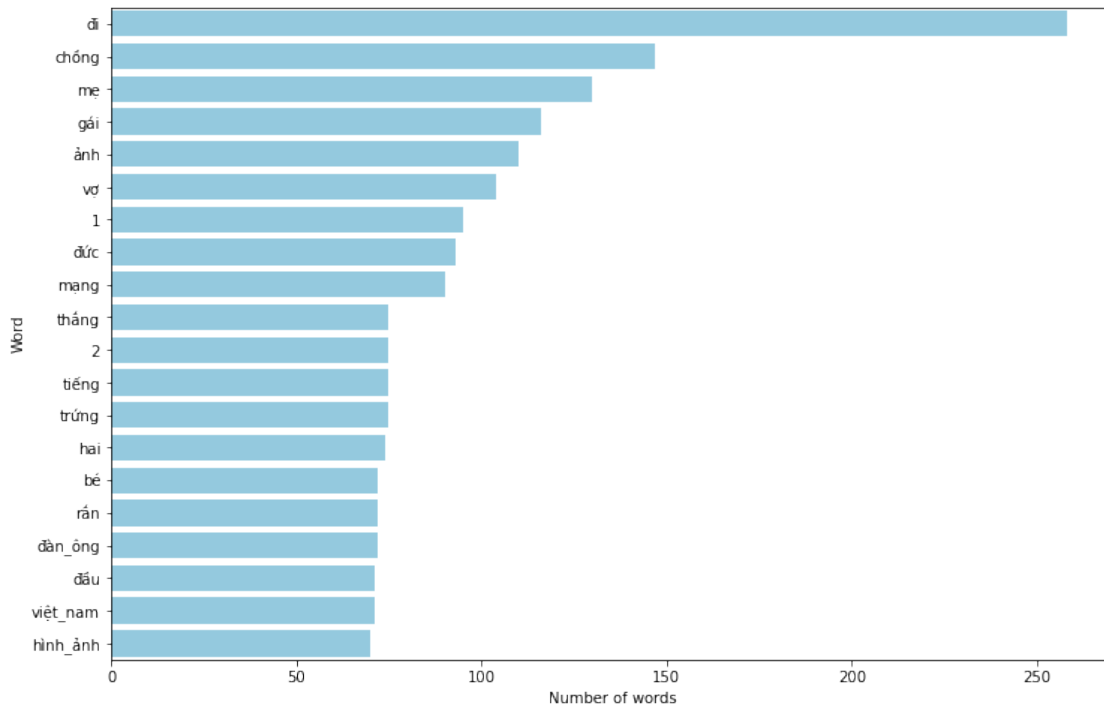
Tương tự, Mình sẽ lấy ra danh sách 20 từ được dùng nhiều nhất trong bộ tin giả và vẽ biểu đồ tương ứng.

```
[ ]: fake_data = data[data["label"] == 1]
fake_token_space = tokenize.WhitespaceTokenizer()
fake_data_text = fake_data['clean_text'].explode()
fake_data_text = ' '.join(fake_data_text)
fake_token_phrase = fake_token_space.tokenize(fake_data_text)
fake_count = nltk.FreqDist(fake_token_phrase)
fake_data_df = pd.DataFrame({"Word": list(fake_count.keys()), "Number of words":
↵ list(fake_count.values())})
fake_data_df = fake_data_df.nlargest(columns = "Number of words", n = 20)
```

```
[ ]: fake_data_df
```

```
[ ]:      Word  Number of words
69      đi      258
1796    chồng    147
648     mẹ      130
882     gái      116
433     ảnh      110
2230    vợ       104
89      1         95
2186    đức       93
227     mạng      90
73     thắng      75
91      2         75
220    tiếng      75
2000   trứng      75
118     hai       74
717     bé       72
1558    rắn       72
2191   đàn_ông    72
3       đầu       71
354   việt_nam    71
45    hình_ảnh    70
```

```
[ ]: plt.figure(figsize=(12,8))
ax = sns.barplot(data = fake_data_df, x = "Number of words", y = "Word", color_
↪ = 'skyblue')
plt.show()
```



Từ “đi” được dùng nhiều nhất với hơn 250 lần (cụ thể là 258)

Vậy là dù cho tin giả hay tin thật thì từ “đi” vẫn được dùng nhiều nhất

2.6 Mô hình đám mây dữ liệu

```
[ ]: #fake news
fake_news = data[data['label']==1]
fake_news_text = fake_news['clean_text'].explode()
fake_news_text = ' '.join(fake_news_text)

wordcloud = WordCloud(width= 800, height= 600,
                       max_font_size = 110,
                       collocations = False).generate(fake_news_text)

plt.figure(figsize=(10,10), facecolor = None)
plt.imshow(wordcloud, interpolation = 'bilinear')
plt.title('Fake News')
plt.axis("off")
plt.show();
```


Fake News



```
[ ]: # real news
real_news = data[data['label']==0]
real_news_text = real_news['clean_text'].explode()
real_news_text = ' '.join(real_news_text)

wordcloud = WordCloud(width= 800, height= 600,
                        background_color='white',
                        max_font_size = 110,
                        collocations = False).generate(real_news_text)

plt.figure(figsize=(10,10), facecolor = None)
plt.imshow(wordcloud, interpolation = 'bilinear')
plt.title('Real News')
plt.axis("off")
plt.show();
```



3 Xây dựng mô hình

Bạn sẽ dùng phương thức `CountVectorizer` để vector hóa văn bản. Vì vậy, bạn sẽ gộp token sau khi được làm sạch lại thành 1 văn bản để có thể sử dụng `CountVectorizer()`.

3.1 1. Tiền xử lý cho mô hình

```
[ ]: def fullPreprocess(s):
      return Preprocess(s).apply(lambda x: " ".join(x))
```

```
[ ]: data['joined_clean_text'] = fullPreprocess(data['text'])
      data.head()
```

```
[ ]:
      text                                     domain \
0  Thủ tướng Abe cúi đầu xin lỗi vì hành động phi...  binhluan.biz
1  Thủ tướng Nhật cúi đầu xin lỗi vì tinh thần ph...  www.ipick.vn
2  Choáng! Cơ trưởng đeo khăn quàng quấy banh nóc...  tintucqpvnet
3  Chưa bao giờ nhạc Kpop lại dễ hát đến thế!!!\n...  tintucqpvnet
4  Đại học Hutech sẽ áp dụng cải cách "Tiếng Việt"...  www.gioitreviet.net
```

	label	clean_text	text_length \
0	1	[thủ_tướng, abe, cúi, đầu, xin_lỗi, hành_động,...	2244
1	1	[thủ_tướng, nhật, cúi, đầu, xin_lỗi, tinh_thần...	884
2	1	[choáng, cơ_trưởng, đeo, khăn_quàng, quấy, ban...	1232
3	1	[nhạc, kpop, hát, giáo_sư, nguyên_lân, dừng, t...	3093
4	1	[đại_học, hutech, áp_dụng, cải_cách, tiếq, việ...	8804

	joined_clean_text
0	thủ_tướng abe cúi đầu xin_lỗi hành_động phi th...
1	thủ_tướng nhật cúi đầu xin_lỗi tinh_thần phi t...
2	choáng cơ_trưởng đeo khăn_quàng quấy banh sân_...
3	nhạc kpop hát giáo_sư nguyên_lân dừng thành_đạ...
4	đại_học hutech áp_dụng cải_cách tiếq việt năm_...

3.2 2. Chia tập test và validation

Vì đây là mô hình dự đoán dựa trên text nên bọn em sẽ chỉ xét cột 'joined_clean_text' tức cột text sau khi qua nhiều bước làm sạch và xử lý (bỏ qua cột domain) làm tập X và cột 'label' làm tập y.

Bọn em sẽ chia tập train và validation ra theo tỉ lệ 7/3.

```
[ ]: X_train, X_val, y_train, y_val = \
    ↪ train_test_split(data['joined_clean_text'], data['label'], random_state=212, test_size=0.
    ↪ 3)
```

3.3 3. Cho chạy các mô hình

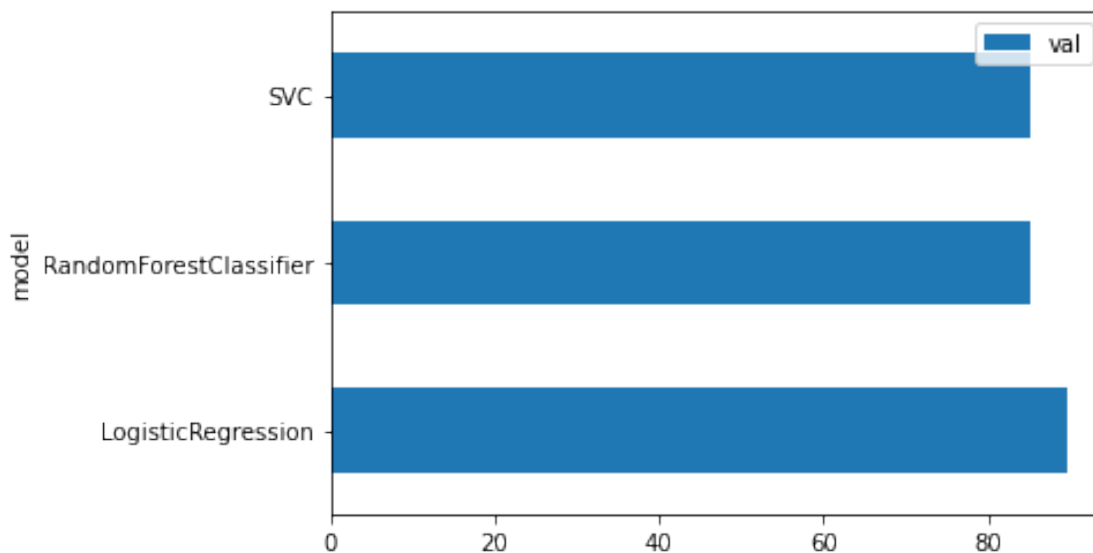
```
[ ]: models = [LogisticRegression(),
sklearn.ensemble.RandomForestClassifier(),
#sklearn.naive_bayes.GaussianNB()
sklearn.svm.SVC()
]
acc_max = 0
accs = []
cm = []
pipelines = []
for model in models:
    full_pipeline = make_pipeline(
        CountVectorizer(),
        model
    )
    pipelines.append(full_pipeline)
    full_pipeline.fit(X_train, y_train)
    y_pred = full_pipeline.predict(X_val)
    accs.append(accuracy_score(y_val, y_pred)*100)
    cm.append(confusion_matrix(y_val, y_pred, normalize='all'))
```

3.4 4. Nhận xét các mô hình

3.4.1 Độ chính xác trên tập validation giữa các mô hình

```
[ ]: model_name = [type(x).__name__ for x in models]
plot_df = pd.DataFrame({'model':model_name,'val':accs}).set_index('model')
print(plot_df)
plot_df.plot.barh();
```

	val
model	
LogisticRegression	89.552239
RandomForestClassifier	85.074627
SVC	85.074627



So sánh độ chính xác giữa các mô hình với nhau, ta thấy không có chênh lệch quá lớn.

Nếu xét khoảng cách giữa mô hình có độ chính xác lớn nhất và thấp nhất thì vẫn có sự chênh lệch đáng xem xét.

Mô hình với độ chính xác cao nhất là :

```
[ ]: best_model = models[accs.index(max(accs))]
best_model, max(accs)
```

```
[ ]: (LogisticRegression(), 89.55223880597015)
```

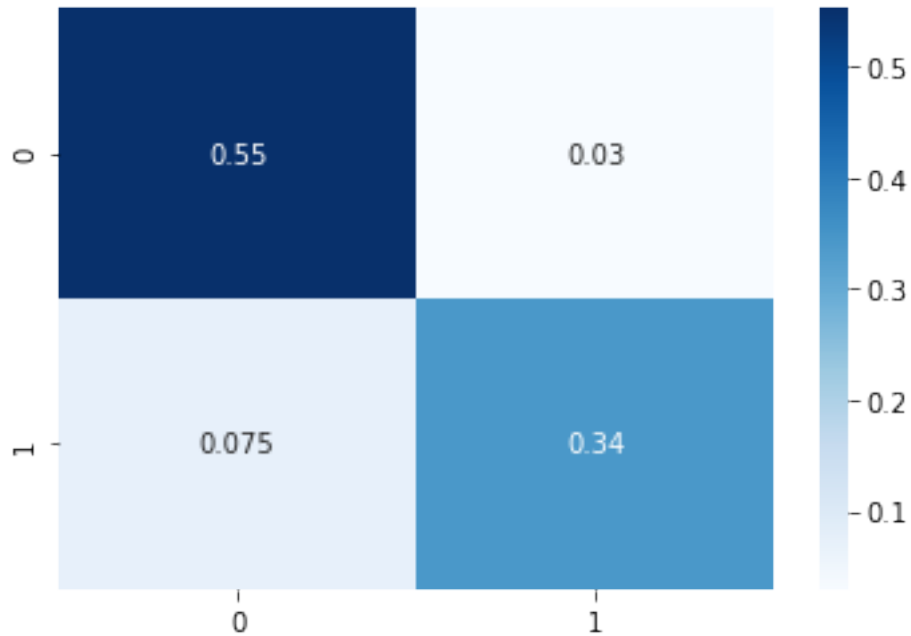
3.4.2 Ma trận lỗi của các mô hình

Một tiêu chí để đánh giá mô hình phân lớp trong những bài toán như thế này là dùng ma trận lỗi.

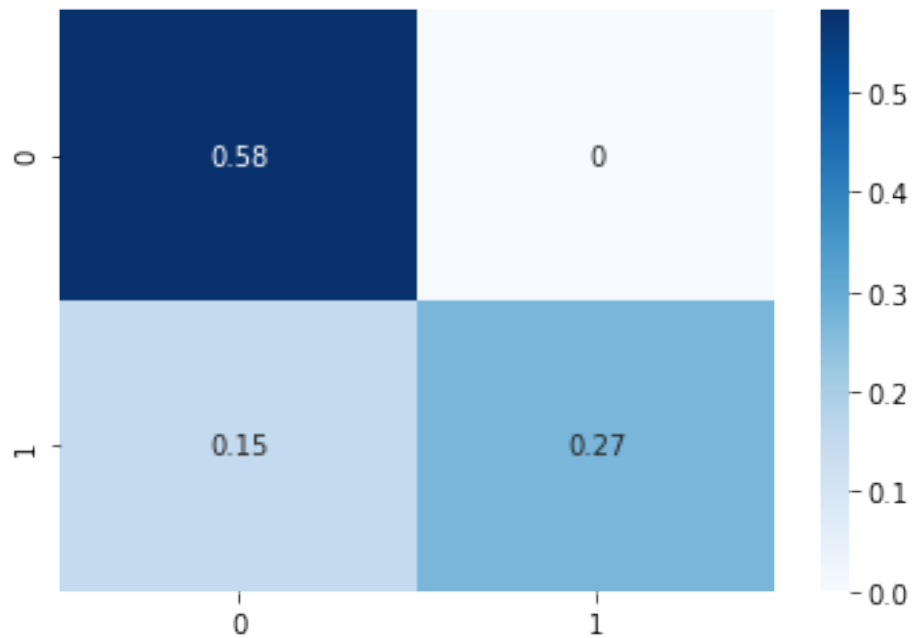
Ta sẽ quan tâm tới ô (0,1) - tỉ lệ fake news nhưng bị dự đoán là real news và ô (1,0) tỉ lệ real news bị dự đoán là fake news.

Rõ ràng tỉ lệ fake news bị dự đoán là real news là tỉ lệ đáng được quan tâm hơn bởi khi thông tin giả được cho là thông tin thật thì sẽ được tiếp nhận vào gây ra sự nhầm lẫn. Còn khi thông tin thật bị gán là thông tin giả thì ta chỉ có thể bị thiếu thông tin (không biết còn đỡ biết biết những thứ sai).

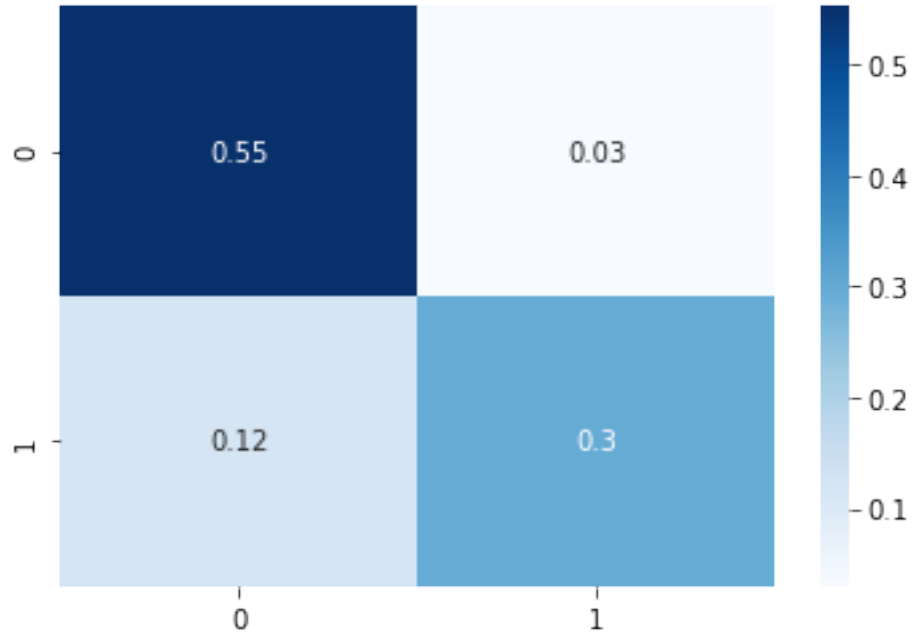
```
[ ]: sns.heatmap(cm[0],annot=True,cmap='Blues');
```



```
[ ]: sns.heatmap(cm[1],annot=True,cmap='Blues');
```



```
[ ]: sns.heatmap(cm[2],annot=True,cmap='Blues');
```



Có thể thấy mô hình LogisticRegression cho ra kết quả khá tốt khi α 0,1 chiếm tỉ lệ thấp nhất trong ba mô hình và đồng thời tổng tỉ lệ lỗi cũng là thấp nhất.

4 Pipeline cho việc deploy mô hình + save mô hình

Tiến hành xây dựng pipe cho phần deploy.

Tiến hành huấn luyện các mô hình đã chọn trên tập dữ liệu đầy đủ (chứ không chia train và validation như trên) :

```
[ ]: models_name = ['LogisticRegression', 'RandomForestClassifier', 'SVC']

for i in range(len(pipelines)):

    pipelines[i].fit(data['joined_clean_text'], data['label'])

    #save model
    with open('models_saved/' + models_name[i] + '.pkl', 'wb') as f:
        pickle.dump(pipelines[i], f)
```

Test thử các pipeline khi chạy trên chính tập huấn luyện (dù điều này khá vô nghĩa, nhưng cứ chạy thử)

```
[ ]: for i in range(len(pipelines)):
        y_pred = pipelines[i].predict(fullPreprocess(data['text']))
        print(accuracy_score(y_pred, data['label']))
```

```
1.0
1.0
1.0
```

5 Thử với một đoạn tin tức trước khi deploy

Test một bài báo lá cải có nội dung giật gân "nam thanh niên điển trai bỗng trở thành **người rắn**". Với kết quả quá rõ ràng như vậy thì cả ba mô hình đều dự đoán là fake news theo kỳ vọng.

```
[ ]:
```

```

text_test = "Chỉ mới 3 tháng phát bệnh, nam thanh niên điển trai bỗng trở thành
↳"người rần" với làn da nứt toát đen xỉ. Hoàn cảnh nghèo túng lại mặc cảm vì
↳bệnh tật, chàng trai 24 tuổi không chịu ăn uống nên thân hình càng gầy còm
↳đáng sợ.\n \n Sinh ra trong hoàn cảnh nghèo khổ, mẹ bệnh tật nằm viện nhiều
↳năm trời, chàng trai 24 tuổi với bao hoài bão lập nghiệp để có khả năng chữa
↳bệnh cho mẹ. Vậy mà, tất cả đã sụp đổ khi Sáng bất ngờ bị căn bệnh kì lạ,
↳quái ác.\n \n Hoàn cảnh đáng thương của chàng trai 24 tuổi mắc bệnh "người
↳rần" ở ấp Vinh Thanh, thị trấn Ngã Giao, huyện Châu Đức, Bà Rịa Vũng Tàu
↳khiến nhiều người xót xa.\n \n Từ chàng trai khỏe mạnh hoạt bát.\n \n Mới 3
↳tháng, Sáng đã ốm yếu, da dẻ nứt toát như da rần.\n \n Cách đây 3 tháng,
↳Sáng vẫn là một thanh niên điển trai, khỏe mạnh. Thế nhưng, Sáng bất ngờ
↳phát hiện là da trở nên xanh xao lạ thường. Vốn tưởng mình mắc bệnh thiếu
↳máu thông thường nên chủ quan, không ngờ dần dần cả cơ thể dần nứt toác, đen
↳xỉ như da rần.\n \n Cũng vì hoàn cảnh khó khăn, Sáng chỉ nằm viện vài ngày
↳rồi ở nhà. Mặc cảm bệnh tật khiến chàng trai 24 tuổi hoạt bát ngày nào không
↳dám tiếp xúc với người lạ, cũng không chịu ăn uống nên cơ thể ngày càng gầy
↳còm. Xót xa hơn, chàng trai sợ căn bệnh của mình làm khổ gia đình nên đã
↳buông tay, không còn ý chí chữa bệnh.\n \n \n -\n \n Gia cảnh quá khó khăn
↳nên chàng trai 24 tuổi buông xuôi, không muốn tiếp tục chữa bệnh.\n \n Biết
↳được số phận đáng thương của Sáng, nhiều nhà hảo tâm đã kêu gọi giúp đỡ và
↳động viên em vượt qua căn bệnh hiểm nghèo.\n \n Ảnh: FB\n \n Nghi Ân (Tổng
↳Hợp)\n \n http://vi.blogtamsu.video/
↳thanh-nien-24-bong-hoa-nguoi-ran-sau-1-dem-da-de-nut-toac-den-xi-ai-nhin-cung-hai.
↳html"

print("Nội dung bài báo :")
print(text_test[0:200]+"...")
print("Kết quả 3 mô hình dự đoán : ")
for pipeline in pipelines:
    print(pipeline.predict(fullPreprocess(pd.Series([text_test]))))

```

Nội dung bài báo :

Chỉ mới 3 tháng phát bệnh, nam thanh niên điển trai bỗng trở thành "người rần" với làn da nứt toát đen xỉ. Hoàn cảnh nghèo túng lại mặc cảm vì bệnh tật, chàng trai 24 tuổi không chịu ăn uống nên thân ...

Kết quả 3 mô hình dự đoán :

[1]

[1]

[1]

6 Bonus point

6.0.1 Tiền xử lý

```

[ ]: Countfit=CountVectorizer().fit(X_train)
X_train1=Countfit.transform(X_train)
X_train1 = X_train1.toarray()
X_val1=Countfit.transform(X_val)

```



```
X_val1 = X_val1.toarray()
```

Khởi tạo model

```
[ ]: model=tf.keras.models.Sequential()
```

```
2022-01-09 13:40:25.007444: I
tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:939] successful NUMA node
read from SysFS had negative value (-1), but there must be at least one NUMA
node, so returning NUMA node zero
2022-01-09 13:40:25.007895: W
tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load
dynamic library 'libcudart.so.11.0'; dlerror: libcudart.so.11.0: cannot open
shared object file: No such file or directory
2022-01-09 13:40:25.007947: W
tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load
dynamic library 'libcublas.so.11'; dlerror: libcublas.so.11: cannot open shared
object file: No such file or directory
2022-01-09 13:40:25.007979: W
tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load
dynamic library 'libcublasLt.so.11'; dlerror: libcublasLt.so.11: cannot open
shared object file: No such file or directory
2022-01-09 13:40:25.008016: W
tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load
dynamic library 'libcufft.so.10'; dlerror: libcufft.so.10: cannot open shared
object file: No such file or directory
2022-01-09 13:40:25.008046: W
tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load
dynamic library 'libcurand.so.10'; dlerror: libcurand.so.10: cannot open shared
object file: No such file or directory
2022-01-09 13:40:25.008073: W
tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load
dynamic library 'libcusolver.so.11'; dlerror: libcusolver.so.11: cannot open
shared object file: No such file or directory
2022-01-09 13:40:25.008102: W
tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load
dynamic library 'libcusparse.so.11'; dlerror: libcusparse.so.11: cannot open
shared object file: No such file or directory
2022-01-09 13:40:25.008129: W
tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load
dynamic library 'libcudnn.so.8'; dlerror: libcudnn.so.8: cannot open shared
object file: No such file or directory
2022-01-09 13:40:25.008134: W
tensorflow/core/common_runtime/gpu/gpu_device.cc:1850] Cannot dlopen some GPU
libraries. Please make sure the missing libraries mentioned above are installed
properly if you would like to use GPU. Follow the guide at
https://www.tensorflow.org/install/gpu for how to download and setup the
required libraries for your platform.
```

Skipping registering GPU devices...

2022-01-09 13:40:25.008379: I tensorflow/core/platform/cpu_feature_guard.cc:151]

This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations: AVX2 AVX512F FMA

To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.

Thêm 4 lớp hidden với activation là “relu”

```
[ ]: model.add(tf.keras.layers.Dense(10,input_shape=(X_train1.  
    ↪shape[1],),activation="relu"))  
model.add(tf.keras.layers.Dense(8,activation="relu"))  
model.add(tf.keras.layers.Dense(6,activation="relu"))  
model.add(tf.keras.layers.Dense(8,activation="relu"))
```

Thêm lớp output với 1 unit có hàm activation là “sigmoid” (sigmoid trả về xác suất từ 0->1)

```
[ ]: model.add(tf.keras.layers.Dense(1,activation="sigmoid"))
```

Biên dịch mô hình

```
[ ]: model.compile(  
    optimizer="adam",  
    loss="binary_crossentropy",  
    metrics=["accuracy"]  
)
```

Fit model (30 lần)

```
[ ]: model.fit(X_train1, y_train, epochs=40)
```

Epoch 1/40

5/5 [=====] - 0s 2ms/step - loss: 0.6916 - accuracy: 0.5000

Epoch 2/40

5/5 [=====] - 0s 1ms/step - loss: 0.6510 - accuracy: 0.6667

Epoch 3/40

5/5 [=====] - 0s 1ms/step - loss: 0.6087 - accuracy: 0.6859

Epoch 4/40

5/5 [=====] - 0s 1ms/step - loss: 0.5599 - accuracy: 0.7372

Epoch 5/40

5/5 [=====] - 0s 1ms/step - loss: 0.5012 - accuracy: 0.7628

Epoch 6/40

5/5 [=====] - 0s 1ms/step - loss: 0.4537 - accuracy: 0.7436

Epoch 7/40
5/5 [=====] - 0s 1ms/step - loss: 0.4087 - accuracy:
0.8718
Epoch 8/40
5/5 [=====] - 0s 1ms/step - loss: 0.3761 - accuracy:
0.8910
Epoch 9/40
5/5 [=====] - 0s 1ms/step - loss: 0.3491 - accuracy:
0.9231
Epoch 10/40
5/5 [=====] - 0s 1ms/step - loss: 0.3226 - accuracy:
0.9679
Epoch 11/40
5/5 [=====] - 0s 1ms/step - loss: 0.3043 - accuracy:
0.9679
Epoch 12/40
5/5 [=====] - 0s 1ms/step - loss: 0.2775 - accuracy:
0.9744
Epoch 13/40
5/5 [=====] - 0s 1ms/step - loss: 0.2523 - accuracy:
0.9808
Epoch 14/40
5/5 [=====] - 0s 1ms/step - loss: 0.2274 - accuracy:
0.9872
Epoch 15/40
5/5 [=====] - 0s 1ms/step - loss: 0.1988 - accuracy:
1.0000
Epoch 16/40
5/5 [=====] - 0s 1ms/step - loss: 0.1727 - accuracy:
1.0000
Epoch 17/40
5/5 [=====] - 0s 1ms/step - loss: 0.1432 - accuracy:
1.0000
Epoch 18/40
5/5 [=====] - 0s 1ms/step - loss: 0.1192 - accuracy:
1.0000
Epoch 19/40
5/5 [=====] - 0s 1ms/step - loss: 0.0967 - accuracy:
1.0000
Epoch 20/40
5/5 [=====] - 0s 1ms/step - loss: 0.0773 - accuracy:
1.0000
Epoch 21/40
5/5 [=====] - 0s 1ms/step - loss: 0.0613 - accuracy:
1.0000
Epoch 22/40
5/5 [=====] - 0s 2ms/step - loss: 0.0487 - accuracy:
1.0000

Epoch 23/40
5/5 [=====] - 0s 1ms/step - loss: 0.0389 - accuracy: 1.0000

Epoch 24/40
5/5 [=====] - 0s 2ms/step - loss: 0.0314 - accuracy: 1.0000

Epoch 25/40
5/5 [=====] - 0s 1ms/step - loss: 0.0253 - accuracy: 1.0000

Epoch 26/40
5/5 [=====] - 0s 1ms/step - loss: 0.0209 - accuracy: 1.0000

Epoch 27/40
5/5 [=====] - 0s 1ms/step - loss: 0.0175 - accuracy: 1.0000

Epoch 28/40
5/5 [=====] - 0s 1ms/step - loss: 0.0147 - accuracy: 1.0000

Epoch 29/40
5/5 [=====] - 0s 2ms/step - loss: 0.0127 - accuracy: 1.0000

Epoch 30/40
5/5 [=====] - 0s 1ms/step - loss: 0.0109 - accuracy: 1.0000

Epoch 31/40
5/5 [=====] - 0s 1ms/step - loss: 0.0096 - accuracy: 1.0000

Epoch 32/40
5/5 [=====] - 0s 2ms/step - loss: 0.0086 - accuracy: 1.0000

Epoch 33/40
5/5 [=====] - 0s 1ms/step - loss: 0.0076 - accuracy: 1.0000

Epoch 34/40
5/5 [=====] - 0s 1ms/step - loss: 0.0068 - accuracy: 1.0000

Epoch 35/40
5/5 [=====] - 0s 1ms/step - loss: 0.0062 - accuracy: 1.0000

Epoch 36/40
5/5 [=====] - 0s 2ms/step - loss: 0.0056 - accuracy: 1.0000

Epoch 37/40
5/5 [=====] - 0s 1ms/step - loss: 0.0052 - accuracy: 1.0000

Epoch 38/40
5/5 [=====] - 0s 1ms/step - loss: 0.0048 - accuracy: 1.0000

```
Epoch 39/40
5/5 [=====] - 0s 1ms/step - loss: 0.0044 - accuracy:
1.0000
Epoch 40/40
5/5 [=====] - 0s 1ms/step - loss: 0.0040 - accuracy:
1.0000
```

```
[ ]: <keras.callbacks.History at 0x7fc6384c7d00>
```

Đánh giá mô hình

```
[ ]: model.evaluate(X_val1,y_val,verbose=0)
```

```
[ ]: [0.35917341709136963, 0.8358209133148193]
```

Dự đoán

```
[ ]: y_val_predict=model.predict(X_val1)
y_val_predict=y_val_predict[:,0]
y_val_predict=np.round(y_val_predict).astype('int')
y_val_predict
```

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

Độ chính xác

```
[ ]: accuracy_score(y_val,y_val_predict)
```

```
[ ]: 0.835820895522388
```

6.1 Link deploy

Demo cho chương trình: [Link](#)

6.2 Nguồn tham khảo

<https://github.com/stopwords/vietnamese-stopwords/blob/master/vietnamese-stopwords.txt>

https://maelfabien.github.io/machinelearning/NLP_1/#i-what-is-preprocessing

https://github.com/SushwanthReddy/Fake-News-Detection-using-Machine-Learning/blob/master/FakeNewsDetection.ipynb?fbclid=IwAR1G2t9F6YJ3Wu6WPIpOX6GKV8oMF7PXRiUNo_uHw_wKORCmedg

<https://www.kaggle.com/rodolfooluna/fake-news-detector?fbclid=IwAR31X9TZK68xc-G0nndQ8eF0YJXd0oRa-tj1levZvflUhm5V4clmMBErmA>