PROJECT 3 - FAKE NEWS DETECTION FOR VIETNAMESE

Link trang web đã deploy: https://share.streamlit.io/tronglee3107/vietnamese-fake-news-detector/app.py

Bảng phân công công việc:

MSSV	Họ và tên	Công việc
19120057	Lê Quốc Cường	Số hóa dữ liệu
19120124	Võ Thị Cẩm Quỳnh	Khai phá dữ liệu
19120142	Nguyễn Thị Phương Trang	Tiền xử lý dữ liệu
18120429	Phạm Trung Kiên	Mô hình hóa dữ liệu
19120145	Lê Đào Duy Trọng	Triển khai mô hình bằng streamlit

In [1]:
```
import pandas as pd
from underthesea import word_tokenize
import regex as re
import itertools
import numpy as np
from sklearn.linear_model import PassiveAggressiveClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
import seaborn as sns
import pickle
import warnings
warnings.filterwarnings("ignore")
sns.set()
```

I. Tiền xử lý dữ liệu

In [2]:
```
#Đọc dữ liệu từ tập data VNFD gồm 223 record bản tin tiếng Việt với 2 nhãn: 1 (tin giả) và 0 (tin thật)
df=pd.read_csv("vn_news_223_tdlfr.csv")

#Lưu danh sách stopword được cung cấp trong đề
with open("vietnamese-stopwords.txt", encoding = "utf8", errors ='replace') as f_st:
    stop_word=f_st.read().split("\n")
```

In [3]:
```
#Tiền xử lý dữ liệu gồm các bước lowercase, loại stopword, tokenize,...
#word_tokenize sử dụng thư viện underthesea: https://pypi.org/project/underthesea/
def preprocess_line(a):
    a=a.lower()
    a=re.sub(r'[^\w\s]','', a)
    a=a.replace('\n',' ')
    for i in stop_word:
        temp=' '+i+' '
        if temp in a:
            a=a.replace(temp,' ')
    a=word_tokenize(a)
    return a
```

In [4]:
```
#Tiền xử lý dữ liệu truyền vào là danh sách các dòng text
def preprocess(t):
    for i in range(len(t)):
        t[i]=preprocess_line(t[i])
    return t
```

In [5]:
```
preprocess(df.text)
```

Out[5]:
```
0      [thủ tướng, abe, cúi, đầu, lỗi, hành động, phi...
1      [thủ tướng, nhật, cúi, đầu, lỗi, tinh thần, ph...
2      [choáng, trưởng, đeo, khăn quàng, quẩy, banh, ...
3      [chưa, nhạc, kpop, hát, giáo sư, nguyễn lân, d...
4      [đại học, hutech, áp dụng, cải, tiếq, việt, họ...
                             ...
218    [siêu, máy bay, a350, chở, cđv, việt, nam, đi,...
219    [thưởng, 20000, usd, đội tuyển, cờ vua, việt, ...
220    [trường sơn, giành, hcv, giải, cờ vua, đồng độ...
221    [chuyện, chàng, sinh viên, luật, kiện tướng, l...
222    [tiền đạo, malaysia, hàng, thủ, đội tuyển, việ...
Name: text, Length: 223, dtype: object
```

In [6]:
```
#Danh sách các từ khác nhau có nghĩa trong tập dữ liệu
fea = []
for i in df.text:
    for j in i:
        if j not in fea:
            fea.append(j)
```

In [7]:
```
#Số hóa dữ liệu
def numerics(list_text,fea):
    matrix = []
    for i in list_text:
        tmp = []
        for j in fea:
            if j in i:
                tmp.append(1)
            else:
                tmp.append(0)
        matrix.append(tmp)
    df1 = pd.DataFrame(matrix, columns= fea)
    return df1
```

In [8]:
```
df1=numerics(df.text,fea)
df1.head()
```

Out[8]:

	thủ tướng	abe	cúi	đầu	lỗi	hành động	phi	thể thao	tuyển	nhật	...	song song	tiền đạo	norshahrul	idlan	talaha	đtqg	bharian	bổ ích	h45	hlong
0	1	1	1	1	1	1	1	1	1	1	...	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	0	1	...	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

5 rows × 8151 columns

II. EDA - Khai phá tập dữ liệu:

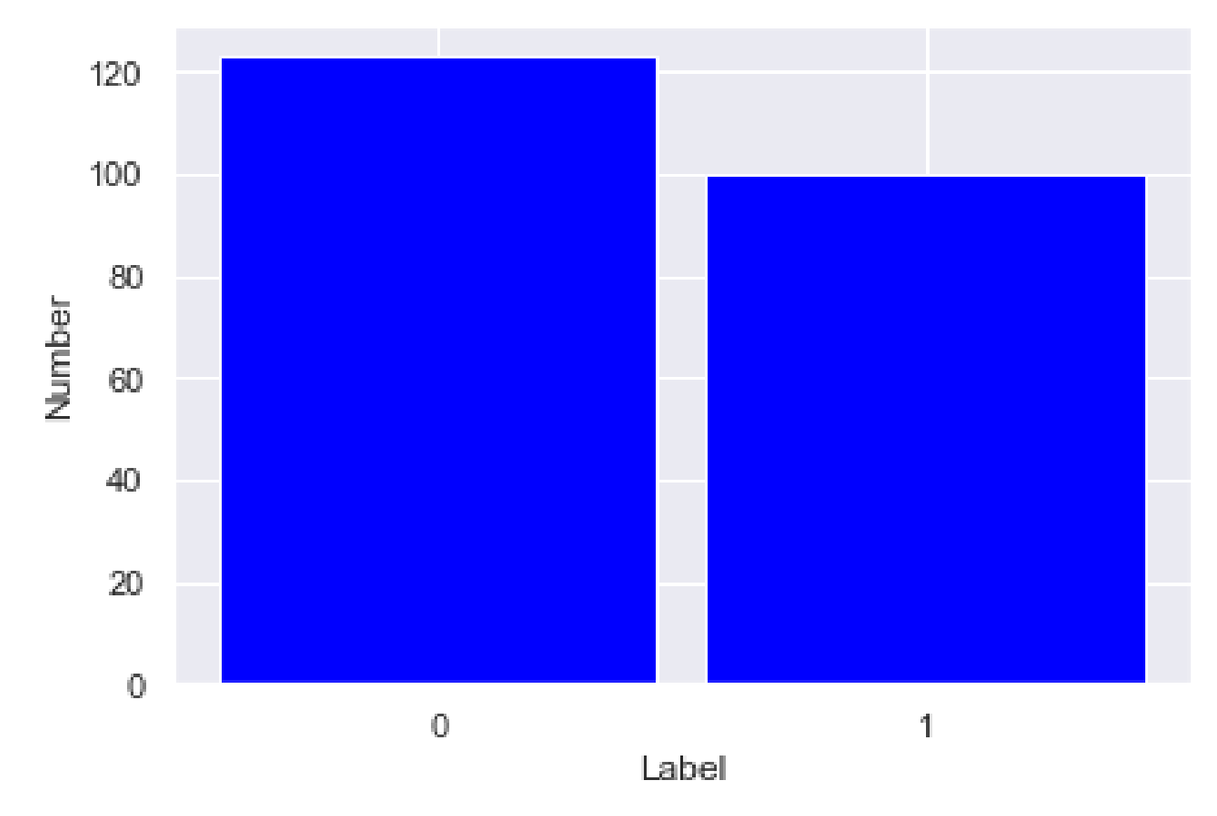
In [20]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 223 entries, 0 to 222
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   text    223 non-null    object
 1   domain  223 non-null    object
 2   label   223 non-null    int64
dtypes: int64(1), object(2)
memory usage: 5.4+ KB
```

Dữ liệu không thiếu và không tồn tại kiểu dữ liệu bị sai

Phân bố của các nhãn

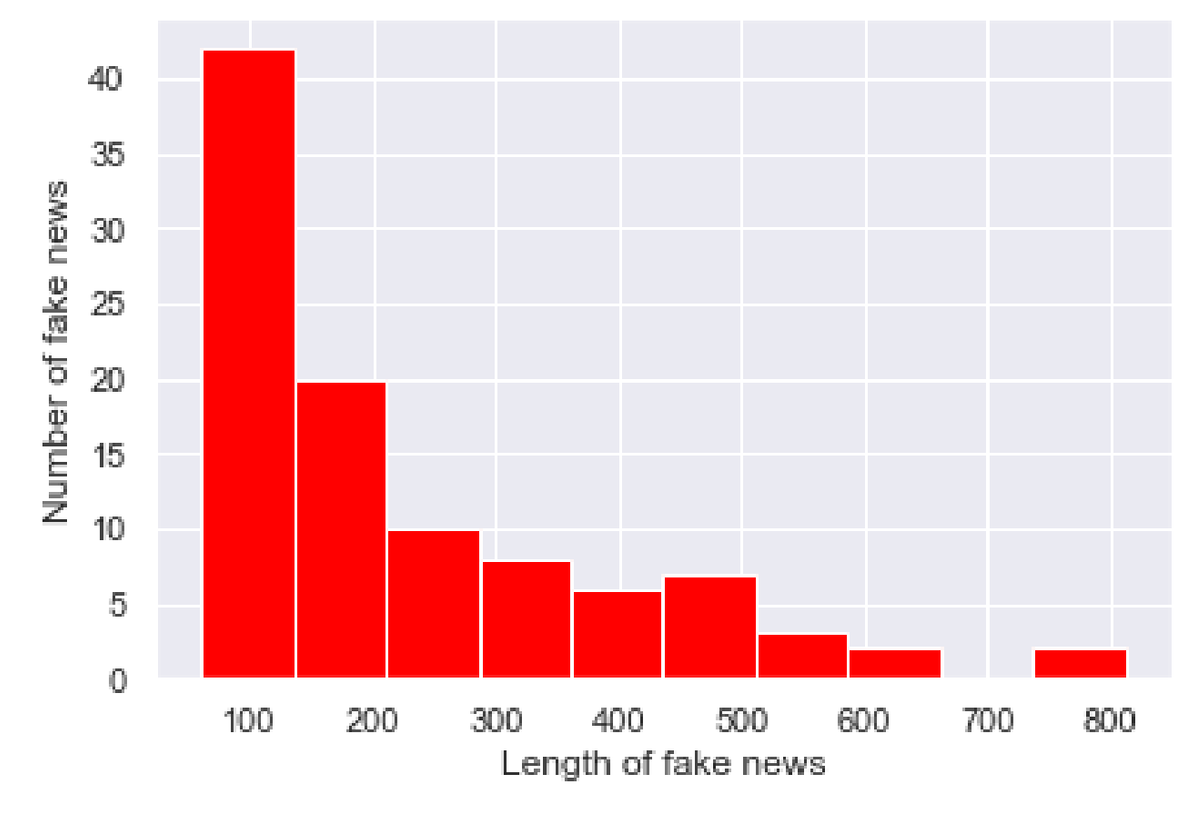
In [10]:
```
plt.bar([0,1], df.groupby(['label']).size().values, 0.9,  color="blue")
plt.xticks([0,1], [0,1])
plt.xlabel('Label')
plt.ylabel('Number')
plt.show()
```

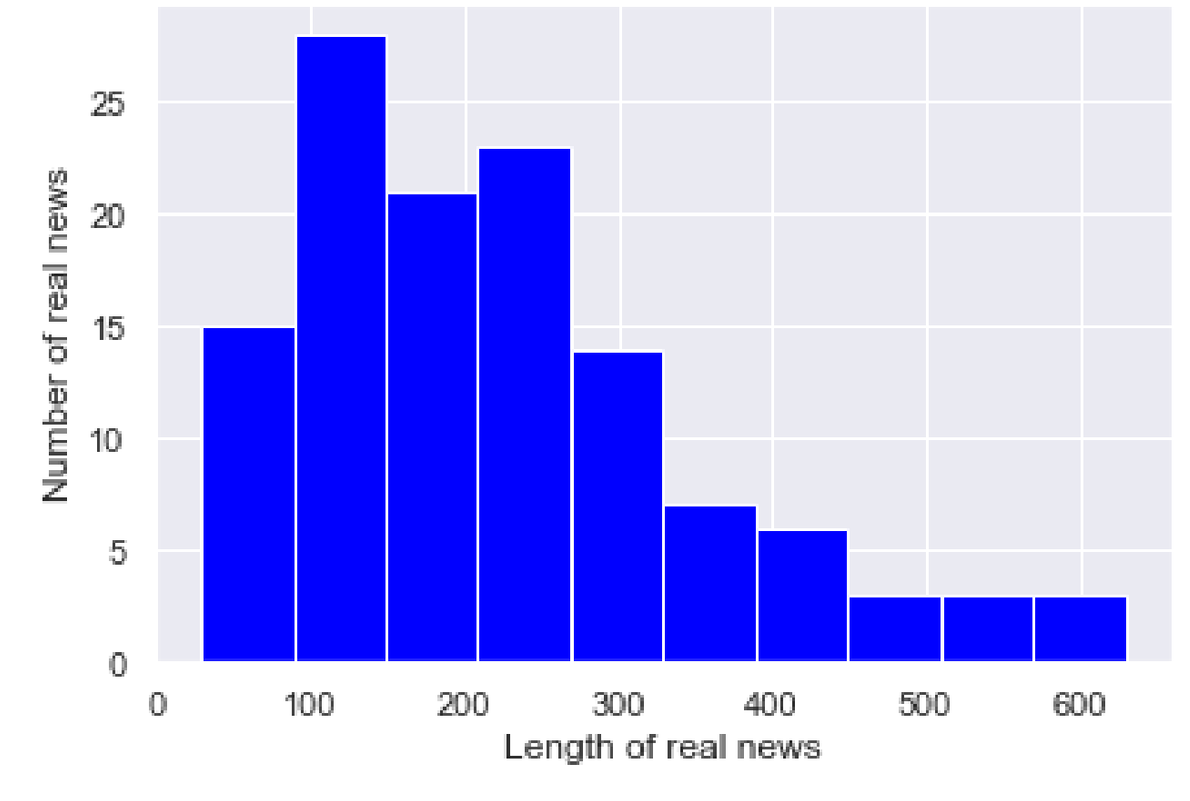


Số lượng giữa tin giả và tin thật khá đồng đều nhau.

Tương quan độ dài của text giữa hai nhãn 0 và 1

In [11]:
```
#Sự phân bố của tin giả
fake_news=df[df['label']==1]['text']
plt.hist([len(i) for i in fake_news],color='red')
plt.xlabel('Length of fake news')
plt.ylabel('Number of fake news')
plt.show()
#Sự phân bố của tin thật
real_news=df[df['label']==0]['text']
plt.hist([len(i) for i in real_news],color='blue')
plt.xlabel('Length of real news')
plt.ylabel('Number of real news')
plt.show()
```

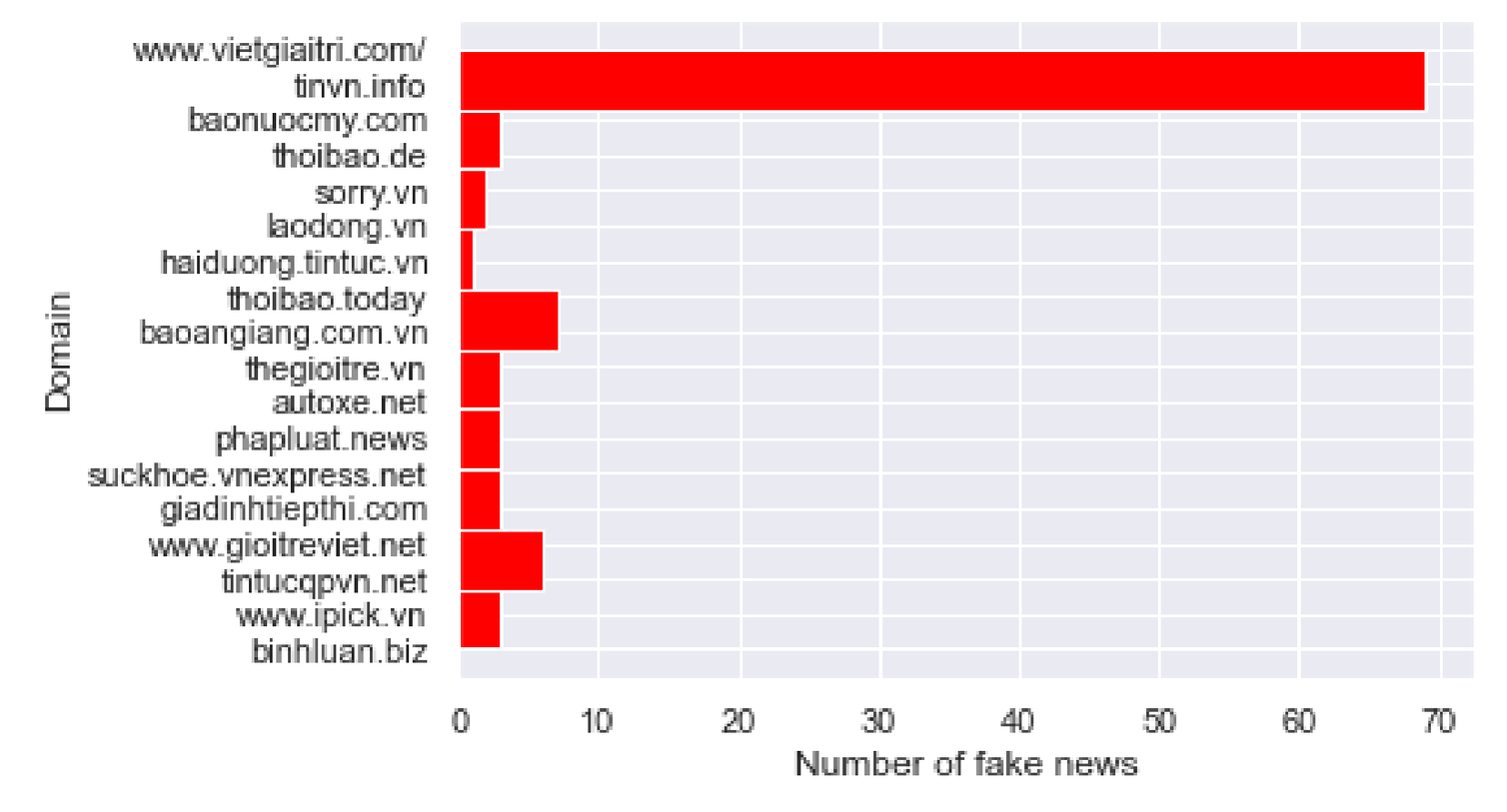


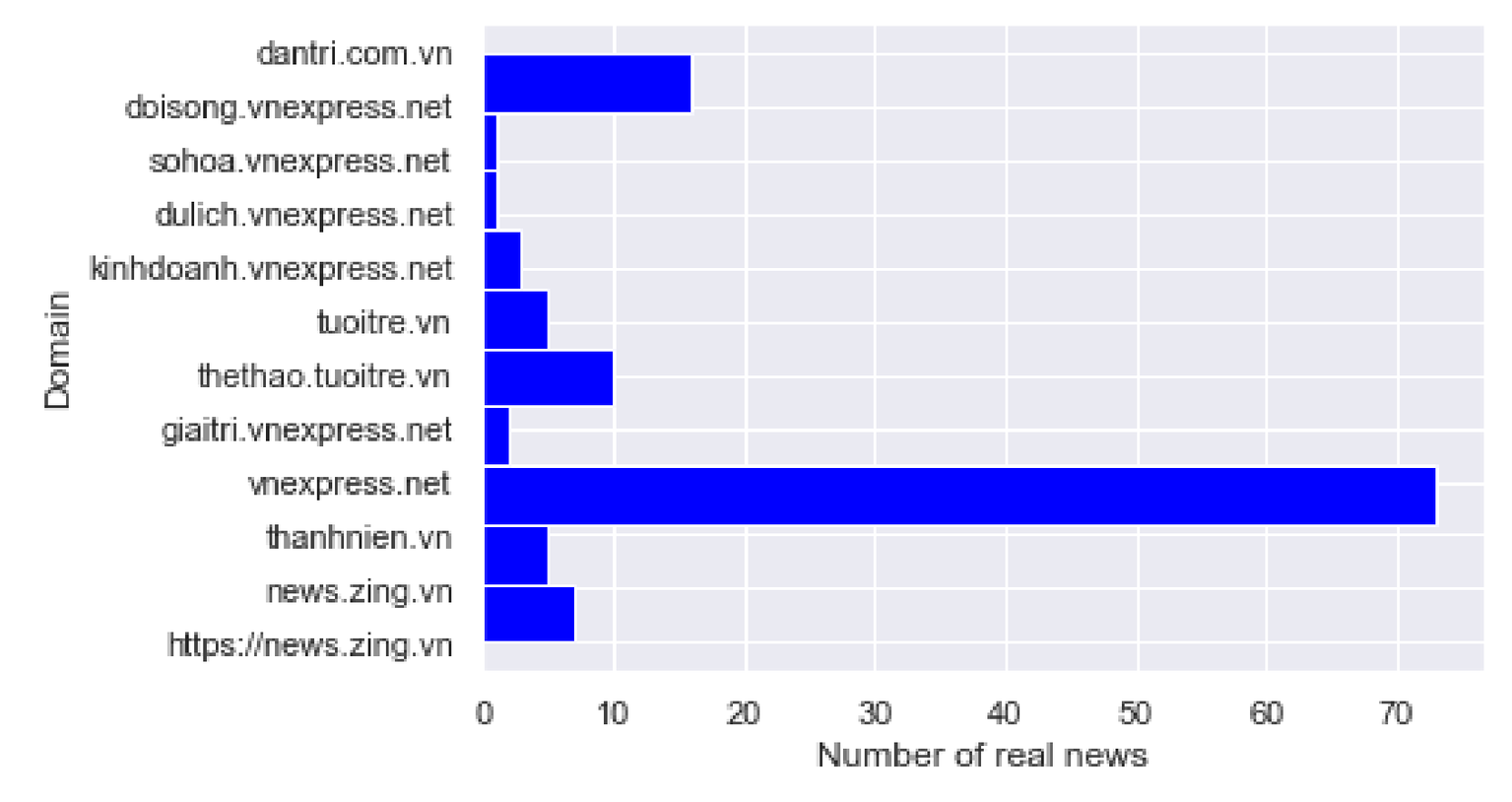


Nhìn chung, tin thật có nhiều tin ngắn hơn tin giả.

Số lượng tin giả, thật ở các domain

In [12]:
```
#Số lượng tin giả ở các domain
fake_news=df[df['label']==1]['domain']
fake_news.hist(color = "red",orientation='horizontal')
plt.xlabel('Number of fake news')
plt.ylabel('Domain')
plt.show()
#Số lượng tin thật ở các domain
real_news=df[df['label']==0]['domain']
real_news.hist(color = "blue",orientation='horizontal')
plt.xlabel('Number of real news')
plt.ylabel('Domain')
plt.show()
```





Từ biểu đồ ta có thể thấy tên miền www.vietgiaitri.com có nhiều tin giả nhất, vnexpress.net cung cấp nhiều tin thật nhất.

III. Mô hình hóa dữ liệu

In [13]:
```
#Chia tập dữ liệu
x_train,x_test,y_train,y_test=train_test_split(df1, df.label, test_size=0.3)
```

1. Mô hình Passive Aggressive Classifier

In [14]:
```
#Khởi tạo PassiveAggressiveClassifier
model_pac = PassiveAggressiveClassifier(C = 0.5, random_state = 5)
model_pac.fit(x_train, y_train)

#Dự đoán trên test set và tính toán độ chính xác
test_pred_pac = model_pac.predict(x_test)
print(f"Test Set Accuracy : {accuracy_score(y_test, test_pred_pac) * 100}%")
```

```
Test Set Accuracy : 86.56716417910447%
```

Mô hình PassiveAggressiveClassifier sau khi huấn luyện đạt được độ chính xác khá cao trên tập dữ liệu test.

In [15]:
```
#Dự đoán sample_1
pred_1=model_pac.predict([x_test.iloc[10]])
print("Pred: ",pred_1)
print("Truth: ",y_test.iloc[10])
#Dự đoán sample_2
pred_2=model_pac.predict([x_test.iloc[40]])
print("Pred: ",pred_2)
print("Truth: ",y_test.iloc[40])
```

```
Pred:  [0]
Truth:  0
Pred:  [1]
Truth:  1
```

2. Mô hình Logistic Regression

In [16]:
```
#Khởi tạo Logistic Regression
model_lrc = LogisticRegression(max_iter=50)
model_lrc.fit(x_train, y_train)

#Dự đoán trên test set và tính toán độ chính xác
test_pred_lrc = model_lrc.predict(x_test)
print(f"Test Set Accuracy : {accuracy_score(y_test, test_pred_lrc) * 100}%")
```

```
Test Set Accuracy : 91.04477611940298%
```

Mô hình Logistic Regression sau khi huấn luyện đạt được độ chính xác trên tập dữ liệu test khá cao cao hơn mô hình 1. Tuy nhiên do tập dữ liệu nhỏ nên độ chính xác khi dự đoán còn phụ thuộc nhiều vào cách chia tập train, test (ngẫu nhiên) ở lần chạy đó.

In [17]:
```
#Dự đoán sample_1
pred_1=model_lrc.predict([x_test.iloc[10]])
print("Pred: ",pred_1)
print("Truth: ",y_test.iloc[10])
#Dự đoán sample_2
pred_2=model_lrc.predict([x_test.iloc[40]])
print("Pred: ",pred_2)
print("Truth: ",y_test.iloc[40])
```

```
Pred:  [0]
Truth:  0
Pred:  [1]
Truth:  1
```

IV. Lưu mô hình

In [18]:
```
#Lưu feature
with open('fea.pkl','wb') as file:
    pickle.dump(fea,file)
#Lưu model 1
model_1 = 'PassiveAgressive.pkl'
pickle.dump(model_pac, open(model_1, 'wb'))
#Lưu model 2
model_2 = 'LogicalRegressive.pkl'
pickle.dump(model_lrc, open(model_2, 'wb'))
```