LAB01 Detección de pishing

Paula Camila Gonzalez Ortega, 18398

Diana Ximena de León Figueroa, 18607

In [18]:

```
import pandas as pd
import numpy as np
import  re
import  matplotlib.pyplot as plt
import seaborn as sns
from urllib.parse import urlparse, urlencode
from pandas_profiling import ProfileReport
# import pandas_profiling as pp
# from pandas_profiling import ProfileReport
# import sklearn
# from sklearn import metrics, model_selection, tree
```

Exploración de datos

In [3]:

```
df = pd.read_csv('dataset_pishing.csv')
```

In [4]:

```
df.head()
```

Out[4]:

	url	ip	nb_www	nb_com	nb_dslash	http_in_path
0	http://www.crestonwood.com/router.php	0	1	0	0	0
1	http://shadetreetechnology.com/V4/validation/a...	1	0	0	0	0
2	https://support-appleld.com.secureupdate.duila...	1	0	1	0	0
3	http://rgipt.ac.in	0	0	0	0	0
4	http://www.iracing.com/tracks/gateway-motorspo...	0	1	0	0	0

5 rows × 67 columns

In [24]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11430 entries, 0 to 11429
Data columns (total 73 columns):
 #   Column                      Non-Null Count  Dtype  
---  ------                      --------------  -----  
 0   url                         11430 non-null  object 
 1   ip                          11430 non-null  int64  
 2   nb_www                      11430 non-null  int64  
 3   nb_com                      11430 non-null  int64  
 4   nb_dslash                   11430 non-null  int64  
 5   http_in_path                11430 non-null  int64  
 6   punycode                    11430 non-null  int64  
 7   port                        11430 non-null  int64  
 8   tld_in_path                 11430 non-null  int64  
 9   tld_in_subdomain            11430 non-null  int64  
 10  abnormal_subdomain          11430 non-null  int64  
 11  nb_subdomains               11430 non-null  int64  
 12  prefix_suffix               11430 non-null  int64  
 13  random_domain               11430 non-null  int64  
 14  shortening_service          11430 non-null  int64  
 15  path_extension              11430 non-null  int64  
 16  nb_redirection              11430 non-null  int64  
 17  nb_external_redirection     11430 non-null  int64  
 18  length_words_raw            11430 non-null  int64  
 19  char_repeat                 11430 non-null  int64  
 20  shortest_words_raw          11430 non-null  int64  
 21  shortest_word_host          11430 non-null  int64  
 22  shortest_word_path          11430 non-null  int64  
 23  longest_words_raw           11430 non-null  int64  
 24  longest_word_host           11430 non-null  int64  
 25  longest_word_path           11430 non-null  int64  
 26  avg_words_raw               11430 non-null  float64
 27  avg_word_host               11430 non-null  float64
 28  avg_word_path               11430 non-null  float64
 29  phish_hints                 11430 non-null  int64  
 30  domain_in_brand             11430 non-null  int64  
 31  brand_in_subdomain          11430 non-null  int64  
 32  brand_in_path               11430 non-null  int64  
 33  suspecious_tld              11430 non-null  int64  
 34  statistical_report          11430 non-null  int64  
 35  nb_hyperlinks               11430 non-null  int64  
 36  ratio_intHyperlinks         11430 non-null  float64
 37  ratio_extHyperlinks         11430 non-null  float64
 38  ratio_nullHyperlinks        11430 non-null  int64  
 39  nb_extCSS                   11430 non-null  int64  
 40  ratio_intRedirection        11430 non-null  int64  
 41  ratio_extRedirection        11430 non-null  float64
 42  ratio_intErrors             11430 non-null  int64  
 43  ratio_extErrors             11430 non-null  float64
 44  login_form                  11430 non-null  int64  
 45  external_favicon            11430 non-null  int64  
 46  links_in_tags               11430 non-null  float64
 47  submit_email                11430 non-null  int64  
 48  ratio_intMedia              11430 non-null  float64
 49  ratio_extMedia              11430 non-null  float64
 50  sfh                         11430 non-null  int64  
 51  iframe                      11430 non-null  int64  
 52  popup_window                11430 non-null  int64  
 53  safe_anchor                 11430 non-null  float64
 54  onmouseover                 11430 non-null  int64  
 55  right_clic                  11430 non-null  int64  
 56  empty_title                 11430 non-null  int64  
 57  domain_in_title             11430 non-null  int64  
 58  domain_with_copyright       11430 non-null  int64  
 59  whois_registered_domain     11430 non-null  int64  
 60  domain_registration_length  11430 non-null  int64  
 61  domain_age                  11430 non-null  int64  
 62  web_traffic                 11430 non-null  int64  
 63  dns_record                  11430 non-null  int64  
 64  google_index                11430 non-null  int64  
 65  page_rank                   11430 non-null  int64  
 66  status                      11430 non-null  object 
 67  longitud_url                11430 non-null  int64  
 68  longitud_hostname           11430 non-null  int64  
 69  special_characters          11430 non-null  int64  
 70  is_https                    11430 non-null  int64  
 71  ratio_digits_url            11430 non-null  float64
 72  ratio_digits_domain         11430 non-null  float64
dtypes: float64(13), int64(58), object(2)
memory usage: 6.4+ MB
```

Muestre la cantidad de observaciones etiquetadas en la columna status como “legit” y como “pishing”. ¿Está balanceado el dataset?

Esta balanceado

In [6]:

```
df['status'].value_counts()
```

Out[6]:

```
legitimate    5715
phishing      5715
Name: status, dtype: int64
```

Derivación de las características

In [7]:

```
def getDomain(url):
    return urlparse(url).netloc

def getProtocol(url):
    return 1 if urlparse(url).scheme == 'https' else 0

def getSpecialCharacters(url):
    count_characters = 0
    count_characters += url.count('.')
    count_characters += url.count('-')
    count_characters += url.count('@')
    count_characters += url.count('?')
    count_characters += url.count('&')
    count_characters += url.count('|')
    count_characters += url.count('=')
    count_characters += url.count('_')
    count_characters += url.count('~')
    count_characters += url.count('%')
    count_characters += url.count('/')
    count_characters += url.count('*')
    count_characters += url.count(':')
    count_characters += url.count(',')
    count_characters += url.count(';')
    count_characters += url.count('$')
    count_characters += url.count('%20')
    count_characters += url.count(' ')
    return count_characters
```

In [8]:

```
df['longitud_url'] = df['url'].str.len()
df['longitud_hostname'] = df['url'].apply(getDomain).str.len()
df['special_characters'] = df['url'].apply(getSpecialCharacters)
df['is_https'] = df['url'].apply(getProtocol)
df['ratio_digits_url'] = df['url'].str.count('[0-9]') / df['url'].str.len()
df['ratio_digits_domain'] = df['url'].apply(getDomain).str.count('[0-9]') / df['url'].appl
```

In [9]:

```
df['longitud_url']
```

Out[9]:

```
0          37
1          77
2         126
3          18
4          55
         ...
11425      45
11426      84
11427     105
11428      38
11429     477
Name: longitud_url, Length: 11430, dtype: int64
```

In [10]:

```
df['longitud_hostname']
```

Out[10]:

```
0        19
1        23
2        50
3        11
4        15
         ..
11425    17
11426    18
11427    16
11428    30
11429    14
Name: longitud_hostname, Length: 11430, dtype: int64
```

In [11]:

```
df['special_characters']
```

Out[11]:

```
0          7
1          7
2         19
3          5
4         10
          ..
11425      7
11426     16
11427     17
11428      6
11429     99
Name: special_characters, Length: 11430, dtype: int64
```

In [12]:

```
df['is_https']
```

Out[12]:

```
0        0
1        0
2        1
3        0
4        0
        ..
11425    0
11426    0
11427    1
11428    0
11429    0
Name: is_https, Length: 11430, dtype: int64
```

In [13]:

```
df['ratio_digits_url']
```

Out[13]:

```
0        0.000000
1        0.220779
2        0.150794
3        0.000000
4        0.000000
           ...   
11425    0.000000
11426    0.023810
11427    0.142857
11428    0.000000
11429    0.085954
Name: ratio_digits_url, Length: 11430, dtype: float64
```

In [14]:

```
df['ratio_digits_domain']
```

Out[14]:

```
0        0.000000
1        0.000000
2        0.000000
3        0.000000
4        0.000000
           ...   
11425    0.000000
11426    0.000000
11427    0.000000
11428    0.000000
11429    0.785714
Name: ratio_digits_domain, Length: 11430, dtype: float64
```

In [15]:

```
df['url']
```

Out[15]:

```
0                    http://www.crestonwood.com/router.php (http://www.cres
tonwood.com/router.php)
1         http://shadetreetechnology.com/V4/validation/a... (http://shadetre
etechnology.com/V4/validation/a...)
2         https://support-appleld.com.secureupdate.duila... (https://support
-appleld.com.secureupdate.duila...)
3                                       http://rgipt.ac.in (http://rgipt.a
c.in)
4         http://www.iracing.com/tracks/gateway-motorspo... (http://www.irac
ing.com/tracks/gateway-motorspo...)
                                ...
11425         http://www.fontspace.com/category/blackletter (http://www.font
space.com/category/blackletter)
11426     http://www.budgetbots.com/server.php/Server%20... (http://www.budg
etbots.com/server.php/Server%20...)
11427     https://www.facebook.com/Interactive-Televisio... (https://www.fac
ebook.com/Interactive-Televisio...)
11428                http://www.mypublicdomainpictures.com/ (http://www.mypu
blicdomainpictures.com/)
11429     http://174.139.46.123/ap/signin?openid.pape.ma... (http://174.139.
46.123/ap/signin?openid.pape.ma...)
Name: url, Length: 11430, dtype: object
```

Ejemplo

In [29]:

```
df.iloc[:, [0, 67, 68, 69, 70, 71, 72 ]].head()
```

Out[29]:

	url	longitud_url	longitud_hostname	special_charact
0	http://www.crestonwood.com/router.php	37	19	
1	http://shadetreetechnology.com/V4/validation/a...	77	23	
2	https://support-appleld.com.secureupdate.duila...	126	50	
3	http://rgipt.ac.in	18	11	
4	http://www.iracing.com/tracks/gateway-motorspo...	55	15	

Preprocesamiento

In [32]:

```
# Codificacion de variable objetivo
df['status'] = df['status'].replace(to_replace='phishing', value = 1)
df['status'] = df['status'].replace(to_replace='legitimate', value = 0)
```

In [33]:

```
df['status']
```

Out[33]:

```
0        0
1        1
2        1
3        0
4        0
        ..
11425    0
11426    1
11427    0
11428    0
11429    1
Name: status, Length: 11430, dtype: int64
```

In [34]:

```
df.drop(['url'], axis = 1, inplace = True)
```

Visualización de resultados

In [35]:

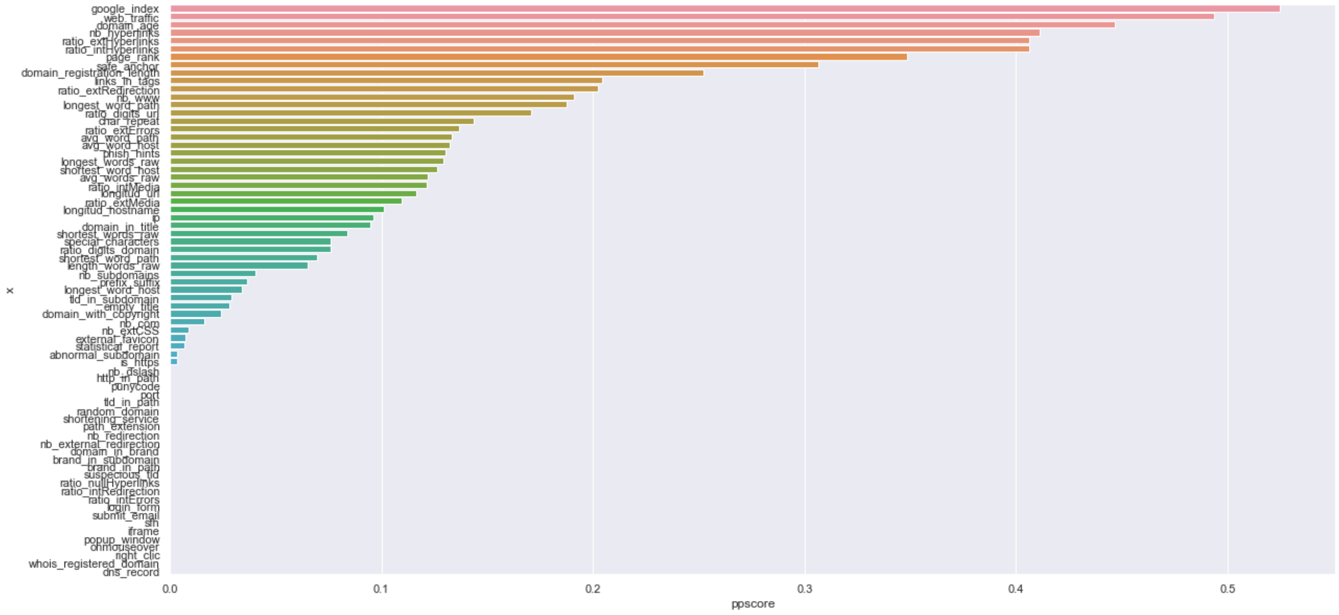
```
profile = ProfileReport(df, title='Reporte Pishing final')
profile.to_file('Reporte Deteccion de Pishing presentacion.html')
```

```
Summarize dataset:                                  85/85 [08:08<00:00, 30.41s/it,
100%                                                Completed]

Generate report structure:                                 1/1 [00:29<00:00,
100%                                                       29.25s/it]

Render HTML:                                             1/1 [00:30<00:00,
100%                                                     30.11s/it]

Export report to file:                                     1/1 [00:02<00:00,
100%                                                       2.52s/it]
```

In [42]:

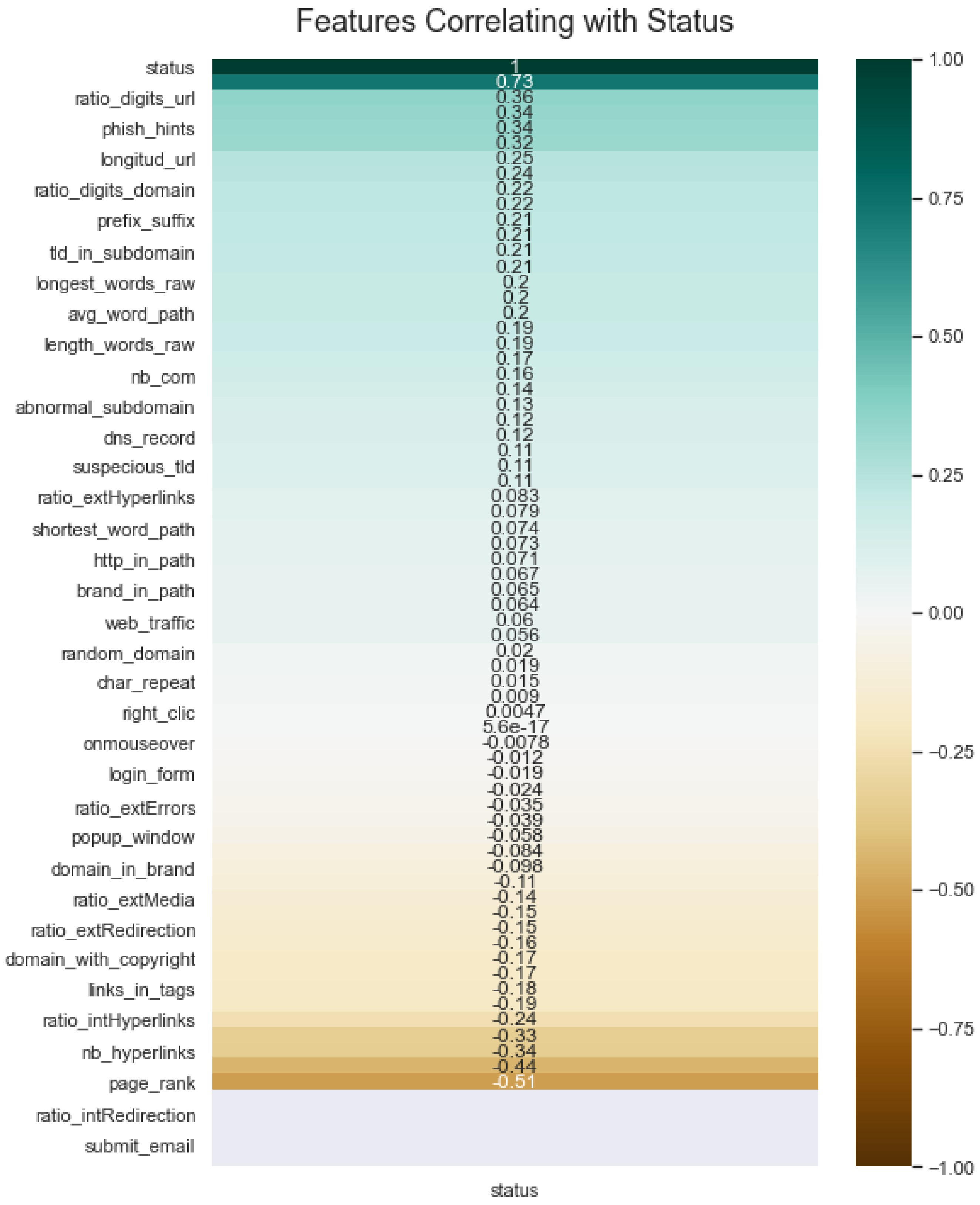
```
from quickda.explore_data import *
from quickda.clean_data import *
from quickda.explore_numeric import *
from quickda.explore_categoric import *
from quickda.explore_numeric_categoric import *
from quickda.explore_time_series import *

eda_numcat(df, "status",
           method = "pps")
```

```
Feature Importance in the prediction of status
```



In [43]:

```
plt.figure(figsize=(8, 12))
heatmap = sns.heatmap(df.corr()[['status']].sort_values(by='status', ascending=False)
heatmap.set_title('Features Correlating with Status', fontdict={'fontsize':18}, pad=1
```



Con el reporte generado fue posible decidir que solo algunas columnas se relacionan mucho con la variable 'status', a continuacion se detallan las mismas: 'nb_com'

,'tld_in_subdomain'
,'abnormal_subdomain'
,'prefix_suffix'
,'longest_words_raw'
,'phish_hints '
,'suspecious_tld '
,'dns_record '
,'longitud_url '
,'ratio_digits_url '
,'ratio_digits_domain'

Selección de características

In [59]:

```
## se guardan solo las columnas importantes y con bastante correlación
new_df = df[['status','nb_com', 'tld_in_subdomain', 'abnormal_subdomain','prefix_suff
```

In [60]:

```
# se eliminan los datos duplicados
new_df.drop_duplicates()
```

Out[60]:

	status	nb_com	tld_in_subdomain	abnormal_subdomain	prefix_suffix	longest_words_r
0	0	0	0	0	0	
1	1	0	0	0	0	
2	1	1	1	0	1	
3	0	0	0	0	0	
4	0	0	0	0	0	
...	...	...	...	...	...	
11423	1	0	0	0	0	
11424	0	1	0	0	0	
11426	1	1	0	0	0	
11427	0	0	0	0	0	
11429	1	0	1	1	0	

7132 rows × 12 columns

Implementación del modelo

In [62]:

```python
import sklearn
from sklearn import metrics, model_selection, tree
```

Separación de datos

In [63]:

```python
target = new_df['status']
feature_matrix = new_df.drop(['status'], axis=1)

print('Final features:', feature_matrix.columns)
feature_matrix.head()

feature_matrix_train, feature_matrix_test, target_train, target_test = model_selection

```

```
Final features: Index(['nb_com', 'tld_in_subdomain', 'abnormal_subdomain',
'prefix_suffix',
       'longest_words_raw', 'phish_hints', 'suspecious_tld', 'dns_record',
       'longitud_url', 'ratio_digits_url', 'ratio_digits_domain'],
      dtype='object')
```

In [71]:

```python
target.to_csv("target_phishing.csv")
```

In [72]:

```python
feature_matrix.to_csv("feature_matrix_phishing.csv")
```

Implementación

In [64]:

```python
clf = tree.DecisionTreeClassifier()
clf = clf.fit(feature_matrix_train, target_train)
```

In [65]:

```
print(feature_matrix_train.count())
```

```
nb_com                 6286
tld_in_subdomain       6286
abnormal_subdomain     6286
prefix_suffix          6286
longest_words_raw      6286
phish_hints            6286
suspecious_tld         6286
dns_record             6286
longitud_url           6286
ratio_digits_url       6286
ratio_digits_domain    6286
dtype: int64
```

In [66]:

```
print(feature_matrix_test.count())
```

```
nb_com                 3429
tld_in_subdomain       3429
abnormal_subdomain     3429
prefix_suffix          3429
longest_words_raw      3429
phish_hints            3429
suspecious_tld         3429
dns_record             3429
longitud_url           3429
ratio_digits_url       3429
ratio_digits_domain    3429
dtype: int64
```

In [67]:

```
target_pred = clf.predict(feature_matrix_test)
```

In [70]:

```python
print(metrics.accuracy_score(target_test, target_pred))
print('Matriz de confusion /n',metrics.confusion_matrix(target_test, target_pred))
print(metrics.classification_report(target_test, target_pred, target_names=['legitimat
```

```
0.7690288713910761
Matriz de confusion /n [[1394  298]
 [ 494 1243]]
              precision    recall  f1-score   support

  legitimate       0.74      0.82      0.78      1692
    phishing       0.81      0.72      0.76      1737

    accuracy                           0.77      3429
   macro avg       0.77      0.77      0.77      3429
weighted avg       0.77      0.77      0.77      3429
```

In [ ]:

```

```