Anexo 11

Proyecto 11: Detector de Spam

Mg. Luis Felipe Bustamante Narváez

Este proyecto se desarrollará utilizando una base de datos sintética creada a través de ChatGPT con base en datasets extraídos en inglés de diferentes repositorios. Esto debido a, no tener muchas fuentes de gran tamaño de correos electrónicos en español, y no poder hacer uso de ciertas bases de datos por motivos de seguridad. Además, se utilizará una base de datos tomada de kaggle, donde se muestran correos en inglés, para validar que sin importar el idioma, se cumple con el objetivo.

Para desarrollar este ejercicio, usaremos Naive Bayes, con el fin de clasificar si un correo electrónico es o no, spam.

Librerías

In [...]
```
pip install seaborn -q
```

```
Note: you may need to restart the kernel to use updated packages.
```

In [...]
```
pip install wordcloud -q
```

```
Note: you may need to restart the kernel to use updated packages.
```

In [...]
```
import numpy as np
import pandas as pd
import seaborn as sn
import matplotlib.pyplot as plt
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.metrics import roc_auc_score, f1_score, confusion_matrix
from sklearn.naive_bayes import MultinomialNB
from wordcloud import WordCloud  #Mostrar gráfico de palabras
```

Cargamos los datos

In [...]
```
opcion = int(input('Ingrese 1 para los correos en español y 2 para los correos en inglés:'))

if opcion == 1:
    path = 'data/spam.csv'
    df = pd.read_csv(path, encoding='utf-8')
elif opcion == 2:
    path = 'data/spam_or_not_spam.csv'
    df = pd.read_csv(path, encoding='ISO-8859-1')
    df.rename(columns={'email':'contenido', 'label':'spam'}, inplace=True)
    #encontramos una cadena NAN  df[df.isna().any(axis=1)]
    df['contenido'] = df['contenido'].fillna('')  #La reemplazamos por cadena vacía
else:
    print('Opción no válida')
```

In [...]
```
df
```

Out[…

	contenido	spam
0	date wed NUMBER aug NUMBER NUMBER NUMBER NUMB...	0
1	martin a posted tassos papadopoulos the greek ...	0
2	man threatens explosion in moscow thursday aug...	0
3	klez the virus that won t die already the most...	0
4	in adding cream to spaghetti carbonara which ...	0
...	...	...
2995	abc s good morning america ranks it the NUMBE...	1
2996	hyperlink hyperlink hyperlink let mortgage le...	1
2997	thank you for shopping with us gifts for all ...	1
2998	the famous ebay marketing e course learn to s...	1
2999	hello this is chinese traditional å□ ä»¶ NUM...	1

3000 rows × 2 columns

In […

```
df['contenido'][0][:500]
```

Out[…

```
' date wed NUMBER aug NUMBER NUMBER NUMBER NUMBER NUMBER from chris garrigues cwg dated NUMB
ER NUMBERfaNUMBERd deepeddy com message id NUMBER NUMBER tmda deepeddy vircio com i can t re
produce this error for me it is very repeatable like every time without fail this is the deb
ug log of the pick happening NUMBER NUMBER NUMBER pick_it exec pick inbox list lbrace lbrace
subject ftp rbrace rbrace NUMBER NUMBER sequence mercury NUMBER NUMBER NUMBER exec pick inbo
x list lbrace lbrace subject ftp rbrace '
```

In […

```
df['spam'][0]
```

Out[…

```
0
```

In […

```
# Agrupamos para obtener el total de datos  (0-> ham, 1->spam)
grouped = df.groupby('spam').count()
grouped
```

Out[…

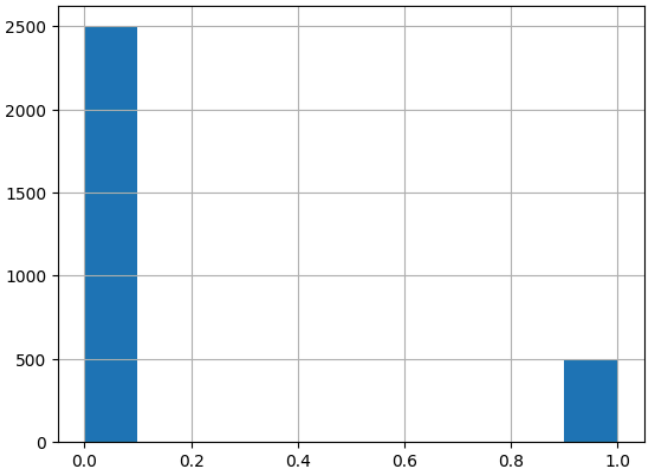
spam	contenido
0	2500
1	500

In […

```
# Gráfico de histograma de la población
df['spam'].hist()
```

Out[…

```
<Axes: >
```



In […

```
# Convertimos la columna spam en un arreglo
Y = df['spam'].to_numpy()
```

In [...]
```
Y
```

Out[...]
```
array([0, 0, 0, ..., 1, 1, 1], dtype=int64)
```

Procesamiento de Datos

Entrenamiento

In [...]
```
# Dividimos los datos en conjuntos de entrenamiento y de prueba
df_train, df_test, Y_train, Y_test = train_test_split(df['contenido'], Y, test_size=0.2)
```

In [...]
```
df_train
```

Out[...]
```
244      zimbabwe has dropped objections to accepting ...
1157    on wed aug NUMBER NUMBER at NUMBER NUMBER ulis...
2418    url URL date NUMBER NUMBER NUMBERtNUMBER NUMBE...
171     david asked my wife noticed something odd the ...
1458    on wed NUMBER sep NUMBER stephane lentz wrote ...
                              ...
294      URL an investigation has been launched after ...
1644     your mail and gives you only the non spam to ...
2648     unlimited web conferencing subscribe to the w...
2599     free adult lifetime membership limited time o...
948      from valdis kletnieks URL date mon NUMBER aug...
Name: contenido, Length: 2400, dtype: object
```

In [...]
```
df_test
```

Out[...]
```
1585    i m taking all my razored mail today and calli...
1495     URL jm URL changed what removed added status ...
855      original message from gary lawrence murphy ga...
1171    help i had gpg working i updated from version ...
2227    url URL date NUMBER NUMBER NUMBERtNUMBER NUMBE...
                              ...
675      help me out here you around barely but don t ...
90      hi dermot if have a look at one of the dists l...
197     hey i has just been given an old toshiba csNUM...
1506    unable to find user matt_relay sbcglobal net p...
354     on wed NUMBER NUMBER NUMBER at NUMBER NUMBER g...
Name: contenido, Length: 600, dtype: object
```

In [...]
```
len(Y_train)
```

Out[...]
```
2400
```

Vectorizamos

In [...]
```
vectores = CountVectorizer(decode_error='ignore')
X_train = vectores.fit_transform(df_train)
X_test = vectores.transform(df_test)
```

In [...]
```
X_train
```

Out[...]
```
<2400x31260 sparse matrix of type '<class 'numpy.int64'>'
        with 284220 stored elements in Compressed Sparse Row format>
```

In [...]
```
X_test
```

Out[...]
```
<600x31260 sparse matrix of type '<class 'numpy.int64'>'
        with 61327 stored elements in Compressed Sparse Row format>
```

Modelo

In [...
```
model = MultinomialNB()
model.fit(X_train, Y_train)
```

Out[...
```
MultinomialNB()
```

In [...
```
# Probamos el modelo con los datos originales
train_accuracy = model.score(X_train, Y_train)
test_accuracy = model.score(X_test, Y_test)
```

In [...
```
# Mostramos la puntuación
print(f'El accuracy de entrenamiento es de {train_accuracy}')
print(f'El accuracy de prueba es de {test_accuracy}')
```

```
El accuracy de entrenamiento es de 0.9954166666666666
El accuracy de prueba es de 0.9983333333333333
```

In [...
```
# Probamos la predicción del modelo
P_train = model.predict(X_train)
P_test = model.predict(X_test)
```

In [...
```
# Mostramos el ajuste del modelo predicho con f1
print(f'Train F1: {f1_score(Y_train, P_train)}')
print(f'Test F1: {f1_score(Y_test, P_test)}')
```

```
Train F1: 0.9864029666254636
Test F1: 0.994535519125683
```

Matriz de Confusión

In [...
```
conf_matrix_train = confusion_matrix(Y_train, P_train)
conf_matrix_train
```

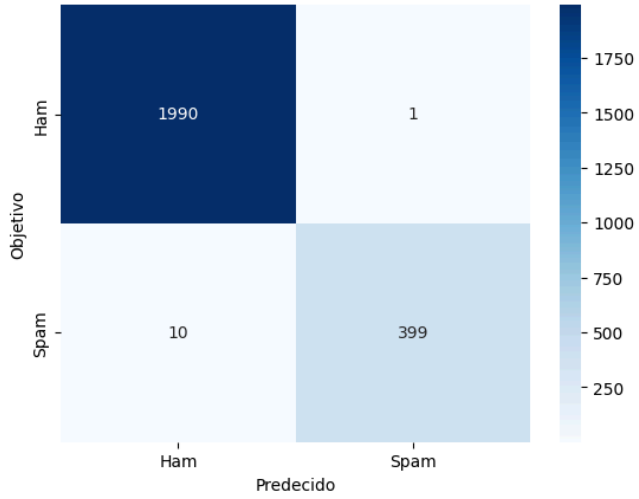
Out[...
```
array([[1990,    1],
       [  10,  399]], dtype=int64)
```

In [...
```
conf_matrix_test = confusion_matrix(Y_test, P_test)
conf_matrix_test
```

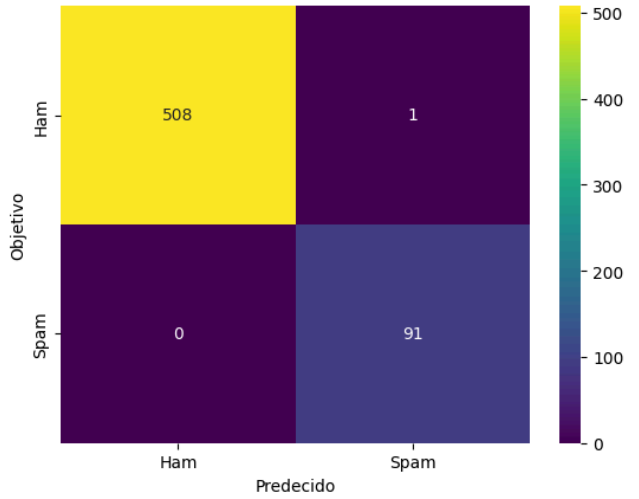
Out[...
```
array([[508,   1],
       [  0,  91]], dtype=int64)
```

In [...
```
# Gráfico de la matriz de confusión
def plot_conf_matrix(c_m, color):
    classes = ['Ham', 'Spam']
    df_cm = pd.DataFrame(c_m, index=classes, columns=classes)
    ax = sn.heatmap(df_cm, annot=True, fmt='g', cmap=color)
    ax.set_xlabel('Predecido')
    ax.set_ylabel('Objetivo')
```

In [...
```
color = 'Blues'  #coolwarm / viridis / Blues / Greens / Reds / magma / cividis
plot_conf_matrix(conf_matrix_train, color)
```



In [...]
```
color = 'viridis'
plot_conf_matrix(conf_matrix_test, color)
```

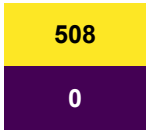


La matriz de correlación, permite visualizar la cantidad de correos que se analizaron en el grupo de entrenamiento y de prueba, y se puede interpretar de la siguiente manera:

- La fila **0**, habla de los correos esperados que **spam**.

- La fila **1**, habla de los correos esperados que son **spam**.

- La columna **0**, habla de los correos predichos que **no spam**.



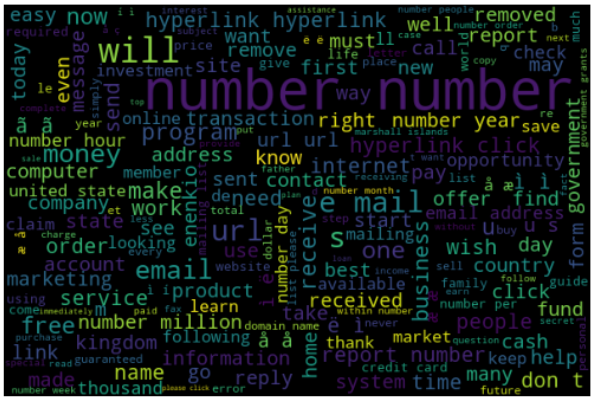
- La columna **1**, habla de los correos predichos que son **spam**.

- La celda **(0, 0)**, muestra los correos que **no spam**, que se esperaban y que se predijeron correctamente.
- La celda **(1, 1)**, muestra los correos que son **spam**, que se esperaban y que también se predijeron correctamente.
- La celda **(0, 1)**, muestra los correos que se esperaban como **spam** y que la predicción los arrojó como **no spam**, es decir, los correos tipo **spam** que se lograron colar como correos buenos.
- La celda **(1, 0)**, muestra los correos que se esperaban como **no spam** y que la predicción los arrojó como **spam**, es decir, los falsos **spam** positivos.

WordCloud

In [...
```
def visualize(label):
    words = ''
    for msg in df[df['spam'] == label]['contenido']:
        msg = msg.lower()
        words += msg + ' '
    wordcloud = WordCloud(width=600, height=400).generate(words)
    plt.imshow(wordcloud)
    plt.axis('off')
    plt.show()
```

In [...
```
visualize(1)  #enviamos un 1, ya que la columna spam es 1 si el correo es spam
```



Explicación

Este generador de nube de palabras, se crea a partir de una cadena **words** vacía, la cual se va llenando cada vez que al recorrer el ciclo **for**, se notan con mayor frecuencia ciertas palabras en los mensaje clasificados como **spam**. Se convierten las palabras a minúsculas y se muestran utilizando el método **WordCloud** de la librería que lleva su mismo nombre.

Entre más se repite una palabra, más grande se ve en la nube de palabras.

Identificación de Falsos Spam

In [...
```
# Vectorizamos la columna contenido
X = vectores.transform(df['contenido'])
# Cremos la columna de predicciones
df['predicciones'] = model.predict(X)
df
```

Out[…

	contenido	spam	predicciones
0	date wed NUMBER aug NUMBER NUMBER NUMBER NUMB...	0	0
1	martin a posted tassos papadopoulos the greek ...	0	0
2	man threatens explosion in moscow thursday aug...	0	0
3	klez the virus that won t die already the most...	0	0
4	in adding cream to spaghetti carbonara which ...	0	0
...	...	...	...
2995	abc s good morning america ranks it the NUMBE...	1	1
2996	hyperlink hyperlink hyperlink let mortgage le...	1	1
2997	thank you for shopping with us gifts for all ...	1	1
2998	the famous ebay marketing e course learn to s...	1	1
2999	hello this is chinese traditional å□ ä»¶ NUM...	1	1

3000 rows × 3 columns

In […

```python
# identificación de los falsos positivos
falso_spam = df[(df['predicciones'] == 1) & (df['spam'] == 0)]['contenido']
falso_ham = df[(df['predicciones'] == 0) & (df['spam'] == 1)]['contenido']
if falso_spam.empty:
    print('No se encontraron falsos Spam')
else:
    print('**Falsos Spam**\n')
    for msg in falso_spam:
        print(msg[:300])

if falso_ham.empty:
    print('\n\nNo se encontraron falsos Ham')
else:
    print('\n\n**Falsos Ham**\n')
    for msg in falso_ham:
        print(msg[:10])
```

```
**Falsos Spam**

 with our telecoms partner bumblebee don t get ripped off by expensive hotel payphone and mob
ile charges save save save on international calls with ryanair s phone partner you ll save up
to NUMBER on international phone calls when you use our online phone card you can use the car
d from any phone in
url URL date not supplied detailed guidelines for vaccinating all NUMBER million citizens wit
hin five days of an outbreak are being dispatched to every state


**Falsos Ham**

 NUMBER NU
 x m a h c
 this URL
 r v r f i

 r v r f i
 r v r f i

 this URL
```

Conclusiones

Se realizó un modelo basado en datos de correos electrónicos en español e inglés, obteniendo resultados diferentes pero con alta probabilidad de clasificación. Esto permite identificar que los modelos de Naive Bayes, en este caso el Multinomial, permiten realizar un óptimo proceso para separar correos basura.

Mg. Luis Felipe Bustamante Narváez