1_NLP_Análisis

July 6, 2025

Creado por:

Isabel Maniega

1 Natural Language Processing (NLP)

https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset

El objetivo de este ejercicio: * Los ordenadores trabajan con números, no con letras * así que necesitamos NLP para tranasformar las palabras a números

---

[1]:
```
import warnings
warnings.filterwarnings("ignore")
```

[2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

[3]:
```
from sklearn.naive_bayes import MultinomialNB
```

1.1 Cargar archivo .csv

[4]:
```
# https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset
```

[5]:
```
df = pd.read_csv("spam.csv",
                 sep=",", encoding='ISO-8859-1')
df.head(15)
```

[5]:

	v1	v2	Unnamed: 2	\
0	ham	Go until jurong point, crazy.. Available only …	NaN	
1	ham	Ok lar… Joking wif u oni…	NaN	
2	spam	Free entry in 2 a wkly comp to win FA Cup fina…	NaN	
3	ham	U dun say so early hor… U c already then say…	NaN	
4	ham	Nah I don't think he goes to usf, he lives aro…	NaN	
5	spam	FreeMsg Hey there darling it's been 3 week's n…	NaN	
6	ham	Even my brother is not like to speak with me. …	NaN	
7	ham	As per your request 'Melle Melle (Oru Minnamin…	NaN	

```
8   spam  WINNER!! As a valued network customer you have…        NaN
9   spam  Had your mobile 11 months or more? U R entitle…       NaN
10   ham  I'm gonna be home soon and i don't want to tal…       NaN
11  spam  SIX chances to win CASH! From 100 to 20,000 po…       NaN
12  spam  URGENT! You have won a 1 week FREE membership …       NaN
13   ham  I've been searching for the right words to tha…       NaN
14   ham                I HAVE A DATE ON SUNDAY WITH WILL!!        NaN

   Unnamed: 3 Unnamed: 4
0         NaN        NaN
1         NaN        NaN
2         NaN        NaN
3         NaN        NaN
4         NaN        NaN
5         NaN        NaN
6         NaN        NaN
7         NaN        NaN
8         NaN        NaN
9         NaN        NaN
10        NaN        NaN
11        NaN        NaN
12        NaN        NaN
13        NaN        NaN
14        NaN        NaN
```

[6]:
```
df = df.iloc[:, 0:2]
df.head()
```

[6]:
```
     v1                                                  v2
0   ham  Go until jurong point, crazy.. Available only …
1   ham                      Ok lar… Joking wif u oni…
2  spam  Free entry in 2 a wkly comp to win FA Cup fina…
3   ham  U dun say so early hor… U c already then say…
4   ham  Nah I don't think he goes to usf, he lives aro…
```

1.2 Nombres para las columnas

[7]:
```
df.columns= ["Status", "Message"]
df.head()
```

[7]:
```
  Status                                            Message
0    ham  Go until jurong point, crazy.. Available only …
1    ham                      Ok lar… Joking wif u oni…
2   spam  Free entry in 2 a wkly comp to win FA Cup fina…
3    ham  U dun say so early hor… U c already then say…
4    ham  Nah I don't think he goes to usf, he lives aro…
```

[8]:
```
df.shape
```

[8]:
```
(5572, 2)
```

[9]:
```
len(df)
```

[9]:
```
5572
```

1.3 Vemos si nos faltan algunos datos

[10]:
```
df.Message.isnull().sum()
```

[10]:
```
np.int64(0)
```

[11]:
```
df.describe()
```

[11]:

	Status	Message
count	5572	5572
unique	2	5169
top	ham	Sorry, I'll call later
freq	4825	30

1.4 ¿Cuántos datos de "spam" en nuestros datos?

Forma 1

[12]:
```
df.head()
```

[12]:

	Status	Message
0	ham	Go until jurong point, crazy.. Available only …
1	ham	Ok lar… Joking wif u oni…
2	spam	Free entry in 2 a wkly comp to win FA Cup fina…
3	ham	U dun say so early hor… U c already then say…
4	ham	Nah I don't think he goes to usf, he lives aro…

[13]:
```
df.Status.value_counts()
```

[13]:
```
Status
ham     4825
spam     747
Name: count, dtype: int64
```

Forma 2

[14]:
```
df.iloc[:,0].value_counts()
```

[14]:
```
Status
ham     4825
spam     747
```

```
Name: count, dtype: int64
```

Forma 3

[15]:
```
df_spam = df[df.Status == "spam"]
len(df_spam)
```

[15]:
```
747
```

Forma 4

[16]:
```
data = df[df.iloc[:,0] == "spam"]
len(data)
```

[16]:
```
747
```

1.5 spam == 1 (True); ham == 0 (False)

Método 1

[17]:
```
df["Status"] = df["Status"].map({"ham": 0, "spam": 1})
df.head()
```

[17]:

	Status	Message
0	0	Go until jurong point, crazy.. Available only …
1	0	Ok lar… Joking wif u oni…
2	1	Free entry in 2 a wkly comp to win FA Cup fina…
3	0	U dun say so early hor… U c already then say…
4	0	Nah I don't think he goes to usf, he lives aro…

[18]:
```
df.shape
```

[18]:
```
(5572, 2)
```

[19]:
```
X = df.Message
```

[20]:
```
y = df.Status
```

1.6 Train, Test split

[21]:
```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0,␣
 ↪test_size=0.2)
```

1.7 Método 1: CountVectorizer

[22]:
```
from sklearn.feature_extraction.text import CountVectorizer
```

[23]:
```
cv = CountVectorizer()
```

[24]:
```
X_train = cv.fit_transform(X_train)
X_test = cv.transform(X_test)
```

[25]:
```
y_train = y_train.astype("int")
y_test = y_test.astype("int")
```

[26]:
```
y_train = np.array(y_train)
y_test = np.array(y_test)
```

[27]:
```
X_train
```

[27]:
```
<Compressed Sparse Row sparse matrix of dtype 'int64'
        with 58826 stored elements and shape (4457, 7612)>
```

[28]:
```
X_test
```

[28]:
```
<Compressed Sparse Row sparse matrix of dtype 'int64'
        with 13975 stored elements and shape (1115, 7612)>
```

[29]:
```
y_train
```

[29]:
```
array([0, 0, 0, …, 0, 0, 0], shape=(4457,))
```

[30]:
```
y_test
```

[30]:
```
array([0, 0, 0, …, 0, 0, 0], shape=(1115,))
```

1.8 Un poco de Machine Learning

[31]:
```
clf = MultinomialNB()
```

[32]:
```
clf.fit(X_train, y_train)
```

[32]:
```
MultinomialNB()
```

[33]:
```
y_pred = clf.predict(X_test)
y_pred
```

[33]:
```
array([0, 0, 0, …, 0, 0, 0], shape=(1115,))
```

[34]:
```
from sklearn.metrics import accuracy_score
acc = accuracy_score(y_pred, y_test)
print(acc * 100)
```

```
98.7443946188341
```

[35]:
```
clf.score(X_test, y_test)
```

[35]: 0.9874439461883409

[36]:
```
aciertos = 0

for i in range(len(y_pred)):
    if y_pred[i] == y_test[i]:
        aciertos += 1
aciertos
```

[36]: 1101

[37]:
```
(aciertos/len(y_pred))*100
```

[37]: 98.7443946188341

1.9 Calcular la matriz de confusión

[38]:
```
len(y_train)
```

[38]: 4457

Falsos Positivos

[39]:
```
FP = 0

for i in np.arange(len(y_test)):
    if y_test[i] == 0 and y_pred[i] == 1:
        FP += 1
FP
```

[39]: 2

Falsos Negativos

[40]:
```
FN = 0

for i in np.arange(len(y_test)):
    if y_test[i] == 1 and y_pred[i] == 0:
        FN += 1
FN
```

[40]: 12

True Positives

[41]:
```
TP = 0

for i in np.arange(len(y_test)):
```

```
    if y_test[i] == 1 and y_pred[i] == 1:
        TP += 1
TP
```

[41]: 154

True Negative

[42]:
```
TN = 0

for i in np.arange(len(y_test)):
    if y_test[i] == 0 and y_pred[i] == 0:
        TN += 1
TN
```

[42]: 947

[43]:
```
confusion_matrix = np.array([[TN, FP],
                             [FN, TP]])
confusion_matrix
```

[43]:
```
array([[947,   2],
       [ 12, 154]])
```

[44]:
```
((TN + TP) / (TN+TP+FP+FN)) *100
```

[44]: 98.7443946188341

Forma con Sklearn

[45]:
```
from sklearn.metrics import confusion_matrix

cm = confusion_matrix(y_test, y_pred)
cm
```

[45]:
```
array([[947,   2],
       [ 12, 154]])
```

1.10 Ahora con: TfidfVectorizer

[46]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0,␣
↪test_size=0.2)
```

[47]:
```
X_train
```

[47]:
```
1114    No no:)this is kallis home ground.amla home to…
3589    I am in escape theatre now. . Going to watch K…
3095    We walked from my moms. Right on stagwood pass…
1012       I dunno they close oredi not… ÌÏ v ma fan…
```

```
3320                                Yo im right by yo work
                                  …
4931                Match started.india  &lt;#&gt;  for 2
3264    44 7732584351, Do you want a New Nokia 3510i c…
1653    I was at bugis juz now wat… But now i'm walk…
2607    :-) yeah! Lol. Luckily i didn't have a starrin…
2732    How dare you stupid. I wont tell anything to y…
Name: Message, Length: 4457, dtype: object
```

[48]:
```
X_test
```

[48]:
```
4456    Aight should I just plan to come up later toni…
690                                     Was the farm open?
944     I sent my scores to sophas and i had to do sec…
3768    Was gr8 to see that message. So when r u leavi…
1189    In that case I guess I'll see you at campus lodge
                                  …
2906                                                ALRITE
1270    Sorry chikku, my cell got some problem thts y …
3944    I will be gentle princess! We will make sweet …
2124    Beautiful Truth against Gravity.. Read careful…
253     Ups which is 3days also, and the shipping comp…
Name: Message, Length: 1115, dtype: object
```

[49]:
```
y_train
```

[49]:
```
1114    0
3589    0
3095    0
1012    0
3320    0
       ..
4931    0
3264    1
1653    0
2607    0
2732    0
Name: Status, Length: 4457, dtype: int64
```

[50]:
```
y_test
```

[50]:
```
4456    0
690     0
944     0
3768    0
1189    0
       ..
```

```
2906    0
1270    0
3944    0
2124    0
253     0
Name: Status, Length: 1115, dtype: int64
```

[51]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
```

[52]:
```
tv = TfidfVectorizer(stop_words = "english")
tv
```

[52]:
```
TfidfVectorizer(stop_words='english')
```

[53]:
```
X_train = tv.fit_transform(X_train)
X_test = tv.transform(X_test)
```

[54]:
```
y_train = y_train.astype("int")
y_test = y_test.astype("int")
```

[55]:
```
y_train = np.array(y_train)
y_test = np.array(y_test)
```

2 Creamos el algoritmo

[56]:
```
clf = MultinomialNB()
```

[57]:
```
clf.fit(X_train, y_train)
```

[57]:
```
MultinomialNB()
```

[58]:
```
y_pred = clf.predict(X_test)
y_pred
```

[58]:
```
array([0, 0, 0, …, 0, 0, 0], shape=(1115,))
```

[59]:
```
from sklearn.metrics import accuracy_score
acc = accuracy_score(y_pred, y_test)
print(acc * 100)
```

```
96.59192825112108
```

[60]:
```
from sklearn.metrics import confusion_matrix

cm = confusion_matrix(y_test, y_pred)
cm
```

```
[60]: array([[949,   0],
             [ 38, 128]])
```

Creado por:

Isabel Maniega