TECNOLOGICO NACIONAL DE MEXICO

INSTITUTO TECNOLOGICO DE NUEVO LAREDO

INTELIGENCIA ARTIFICIAL 2

INGENIERIA EN SISTEMAS COMPUTACIONALES

DOCENTE: LUIS DANIEL CASTILLO GARCIA

U2 – PRACTICA 2 – REGRESION LOGISTICA

JOEL RODRIGUEZ MUÑOZ

NUMERO DE CONTROL: 19100244

NUEVO LAREDO TAMAULIPAS.

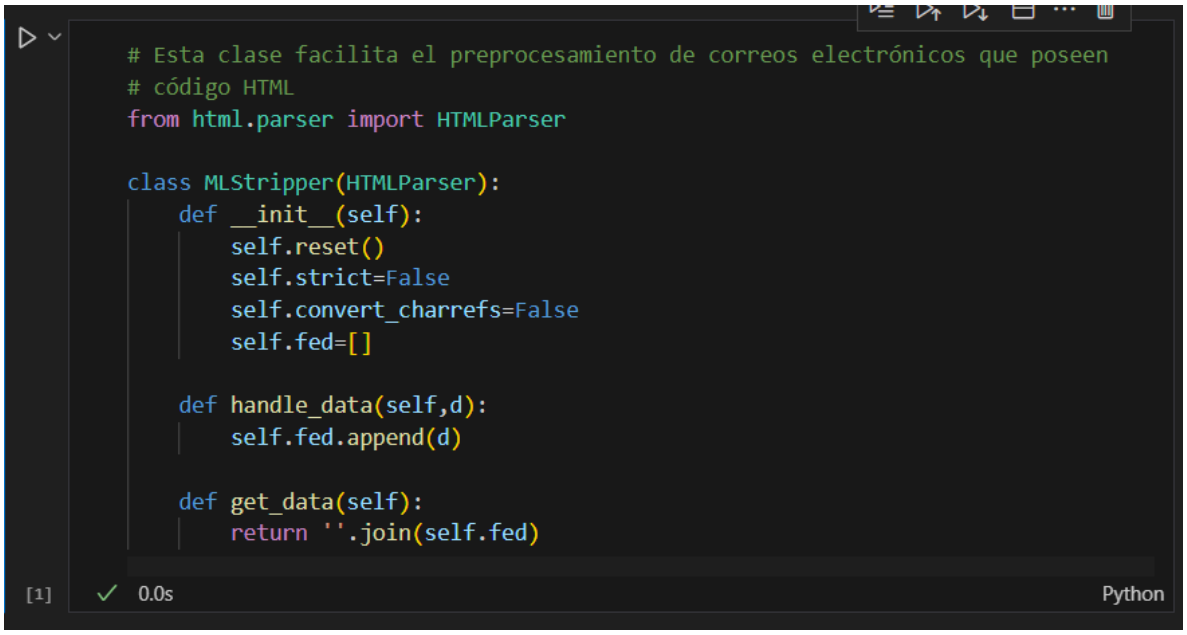
13 Octubre de 2024

Utilizando 20 correos de cualquier cuenta de correo electrónico, 15 no spam y 5 de spam

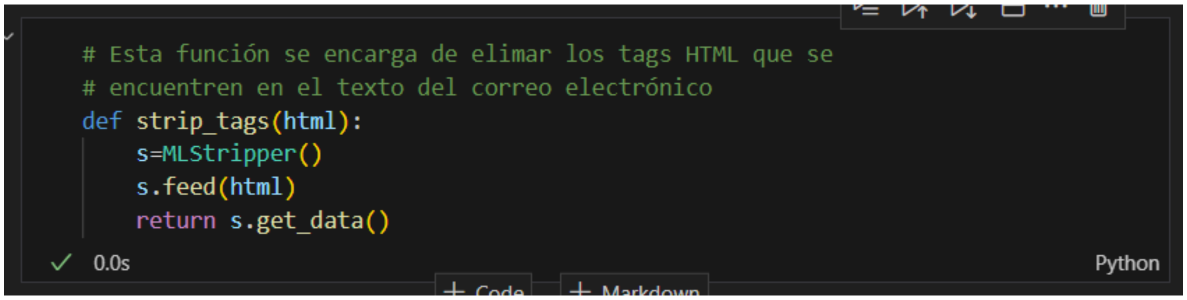
- **Mostrar el contenido del correo**
- **Procesar el correo atreves del Parseador**
- **Etiquetar el correo**
- **Utilizar el modelo ya creado para predecir dichos 20 correos**
- **Comparar la predicción con la etiqueta**

En este ejercicio se muestran los fundamentos de la Regresión Logística planteando uno de los primeros problemas que fueron solucionados mediante el uso de técnicas de Machine Learning: la detección de SPAM.

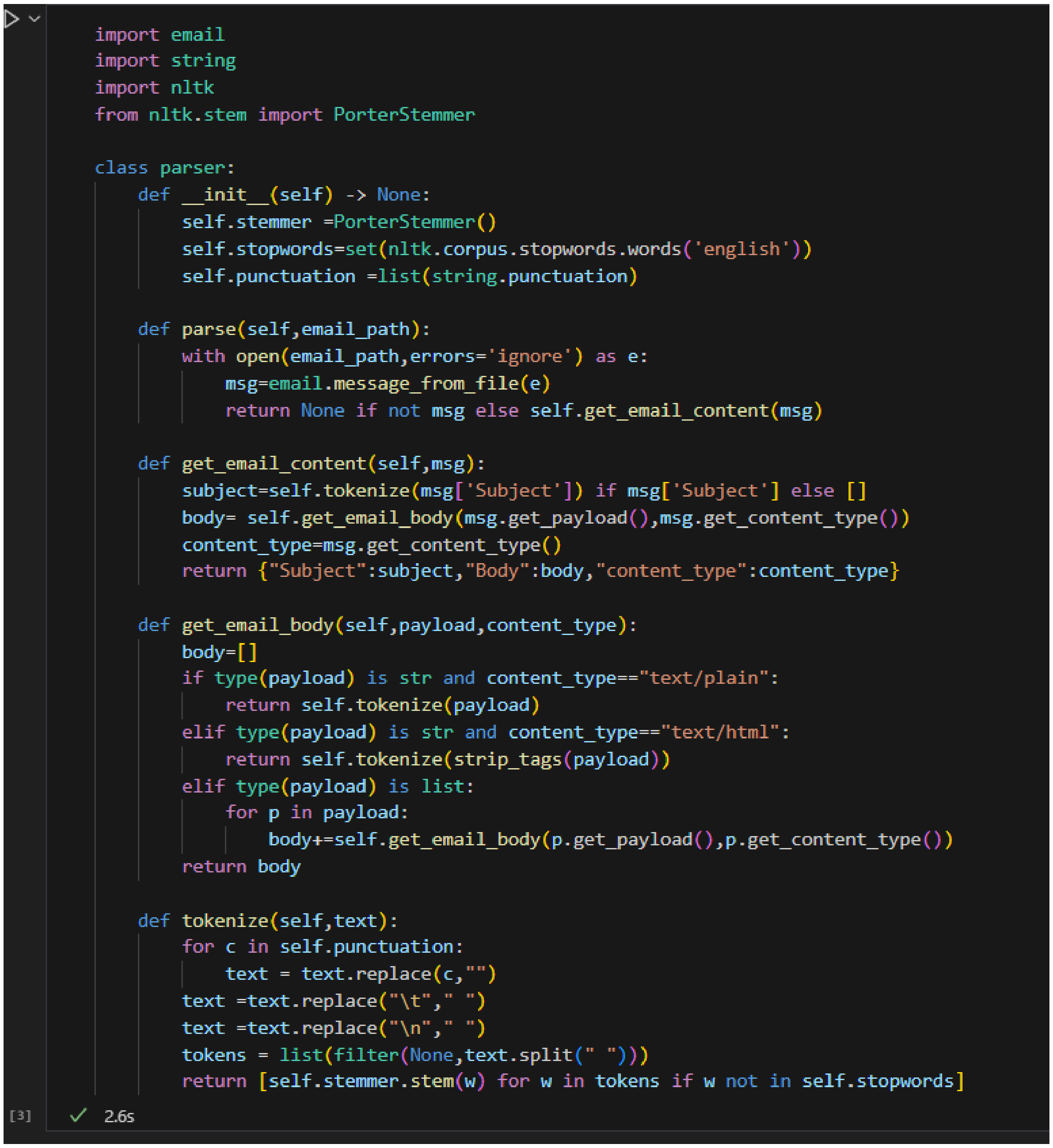
1.- FUNCIONES COMPLEMENTARIAS



```
# Esta clase facilita el preprocesamiento de correos electrónicos que poseen
# código HTML
from html.parser import HTMLParser

class MLStripper(HTMLParser):
    def __init__(self):
        self.reset()
        self.strict=False
        self.convert_charrefs=False
        self.fed=[]

    def handle_data(self,d):
        self.fed.append(d)

    def get_data(self):
        return ''.join(self.fed)
```

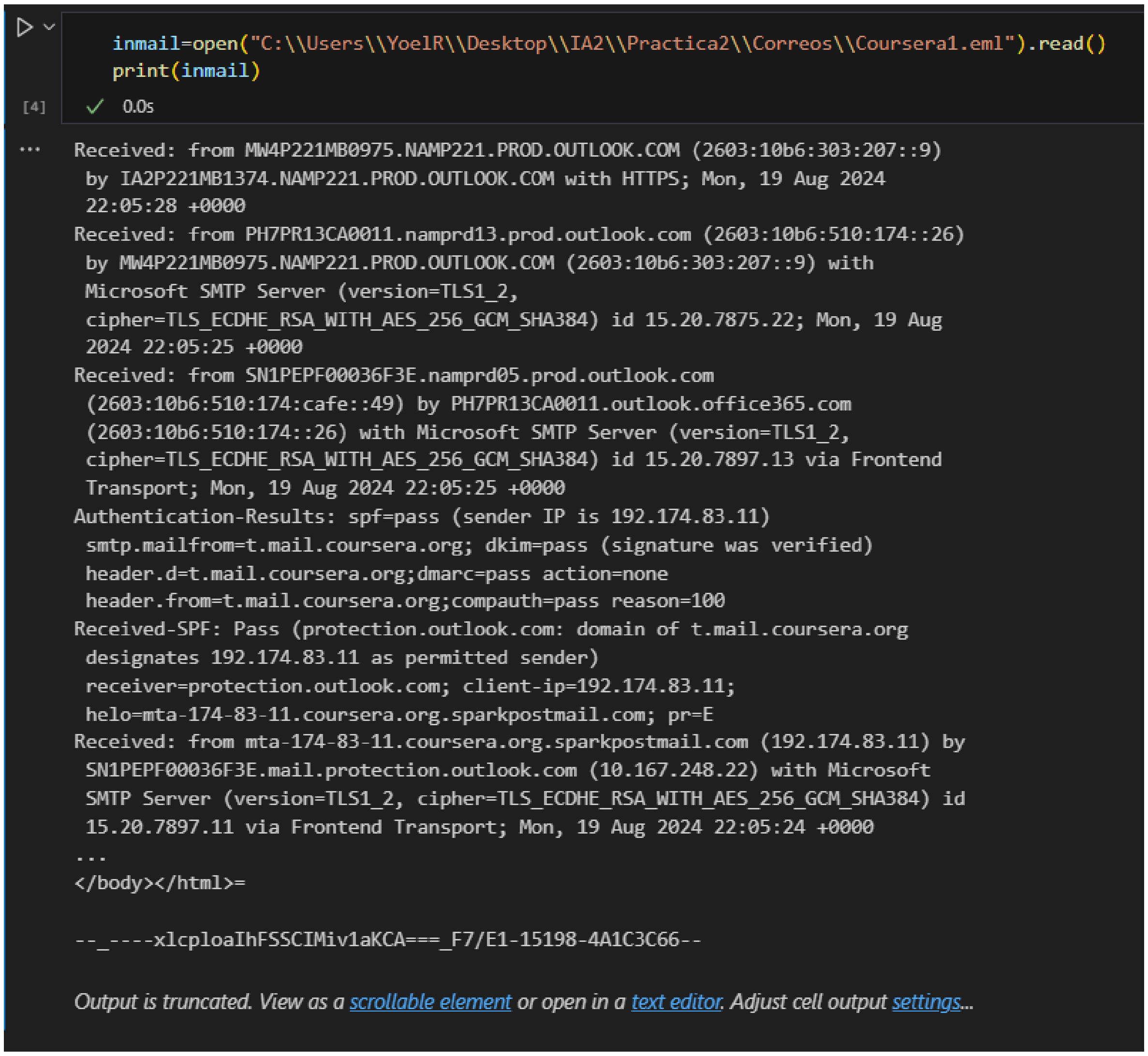


```python
# Esta función se encarga de elimar los tags HTML que se
# encuentren en el texto del correo electrónico
def strip_tags(html):
    s=MLStripper()
    s.feed(html)
    return s.get_data()
```



```python
import email
import string
import nltk
from nltk.stem import PorterStemmer

class parser:
    def __init__(self) -> None:
        self.stemmer =PorterStemmer()
        self.stopwords=set(nltk.corpus.stopwords.words('english'))
        self.punctuation =list(string.punctuation)

    def parse(self,email_path):
        with open(email_path,errors='ignore') as e:
            msg=email.message_from_file(e)
            return None if not msg else self.get_email_content(msg)

    def get_email_content(self,msg):
        subject=self.tokenize(msg['Subject']) if msg['Subject'] else []
        body= self.get_email_body(msg.get_payload(),msg.get_content_type())
        content_type=msg.get_content_type()
        return {"Subject":subject,"Body":body,"content_type":content_type}

    def get_email_body(self,payload,content_type):
        body=[]
        if type(payload) is str and content_type=="text/plain":
            return self.tokenize(payload)
        elif type(payload) is str and content_type=="text/html":
            return self.tokenize(strip_tags(payload))
        elif type(payload) is list:
            for p in payload:
                body+=self.get_email_body(p.get_payload(),p.get_content_type())
        return body

    def tokenize(self,text):
        for c in self.punctuation:
            text = text.replace(c,"")
        text =text.replace("\t"," ")
        text =text.replace("\n"," ")
        tokens = list(filter(None,text.split(" ")))
        return [self.stemmer.stem(w) for w in tokens if w not in self.stopwords]
```

LECTURA DE UN CORREO EN FORMATO RAW



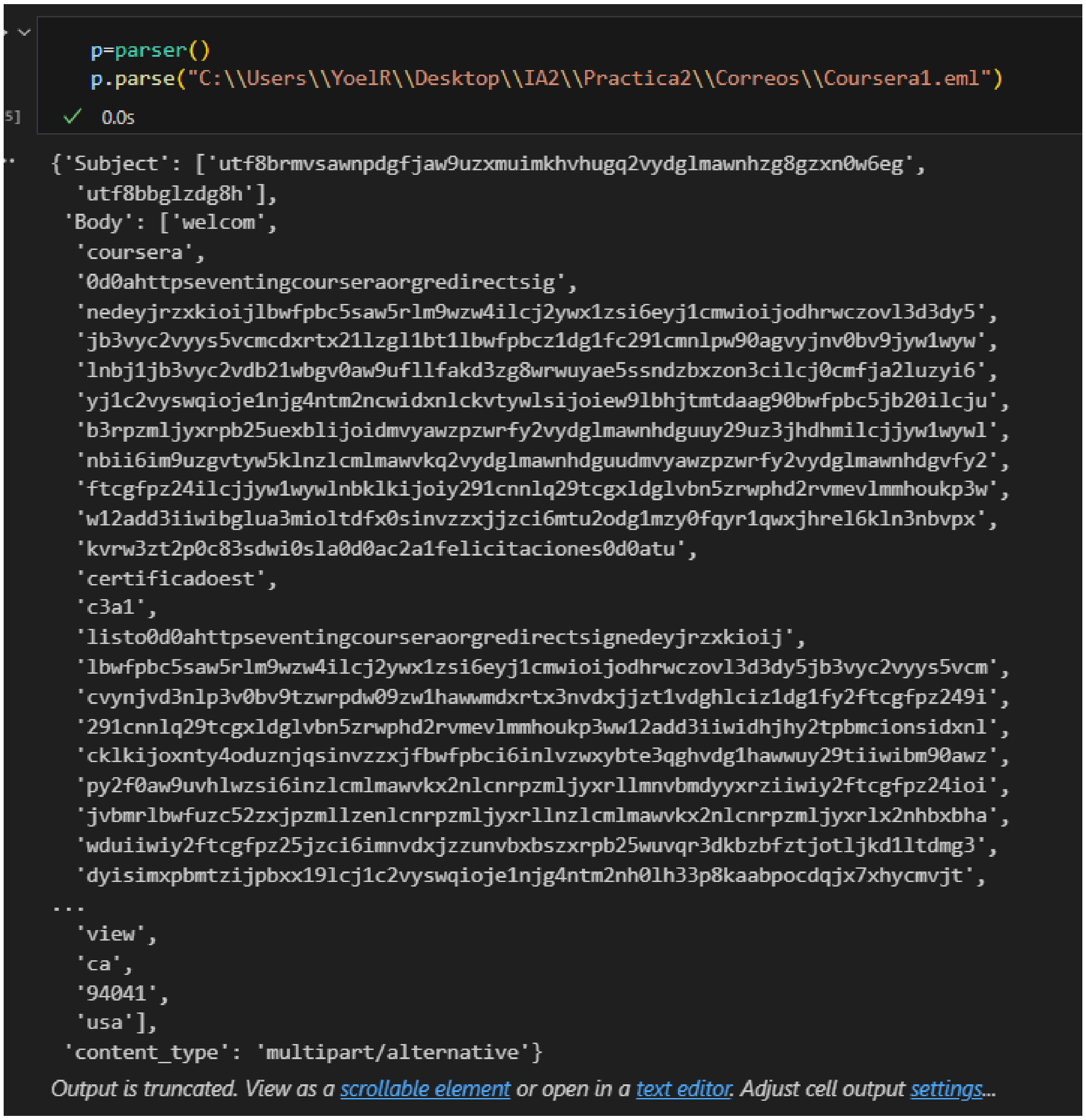
```python
inmail=open("C:\\Users\\YoelR\\Desktop\\IA2\\Practica2\\Correos\\Coursera1.eml").read()
print(inmail)
```

[4] ✓ 0.0s

```
Received: from MW4P221MB0975.NAMP221.PROD.OUTLOOK.COM (2603:10b6:303:207::9)
 by IA2P221MB1374.NAMP221.PROD.OUTLOOK.COM with HTTPS; Mon, 19 Aug 2024
 22:05:28 +0000
Received: from PH7PR13CA0011.namprd13.prod.outlook.com (2603:10b6:510:174::26)
 by MW4P221MB0975.NAMP221.PROD.OUTLOOK.COM (2603:10b6:303:207::9) with
 Microsoft SMTP Server (version=TLS1_2,
 cipher=TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384) id 15.20.7875.22; Mon, 19 Aug
 2024 22:05:25 +0000
Received: from SN1PEPF00036F3E.namprd05.prod.outlook.com
 (2603:10b6:510:174:cafe::49) by PH7PR13CA0011.outlook.office365.com
 (2603:10b6:510:174::26) with Microsoft SMTP Server (version=TLS1_2,
 cipher=TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384) id 15.20.7897.13 via Frontend
 Transport; Mon, 19 Aug 2024 22:05:25 +0000
Authentication-Results: spf=pass (sender IP is 192.174.83.11)
 smtp.mailfrom=t.mail.coursera.org; dkim=pass (signature was verified)
 header.d=t.mail.coursera.org;dmarc=pass action=none
 header.from=t.mail.coursera.org;compauth=pass reason=100
Received-SPF: Pass (protection.outlook.com: domain of t.mail.coursera.org
 designates 192.174.83.11 as permitted sender)
 receiver=protection.outlook.com; client-ip=192.174.83.11;
 helo=mta-174-83-11.coursera.org.sparkpostmail.com; pr=E
Received: from mta-174-83-11.coursera.org.sparkpostmail.com (192.174.83.11) by
 SN1PEPF00036F3E.mail.protection.outlook.com (10.167.248.22) with Microsoft
 SMTP Server (version=TLS1_2, cipher=TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384) id
 15.20.7897.11 via Frontend Transport; Mon, 19 Aug 2024 22:05:24 +0000
...
</body></html>=

--_----xlcploaIhFSSCIMiv1aKCA===_F7/E1-15198-4A1C3C66--
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

PARSING DEL CORREO ELECTRONICO

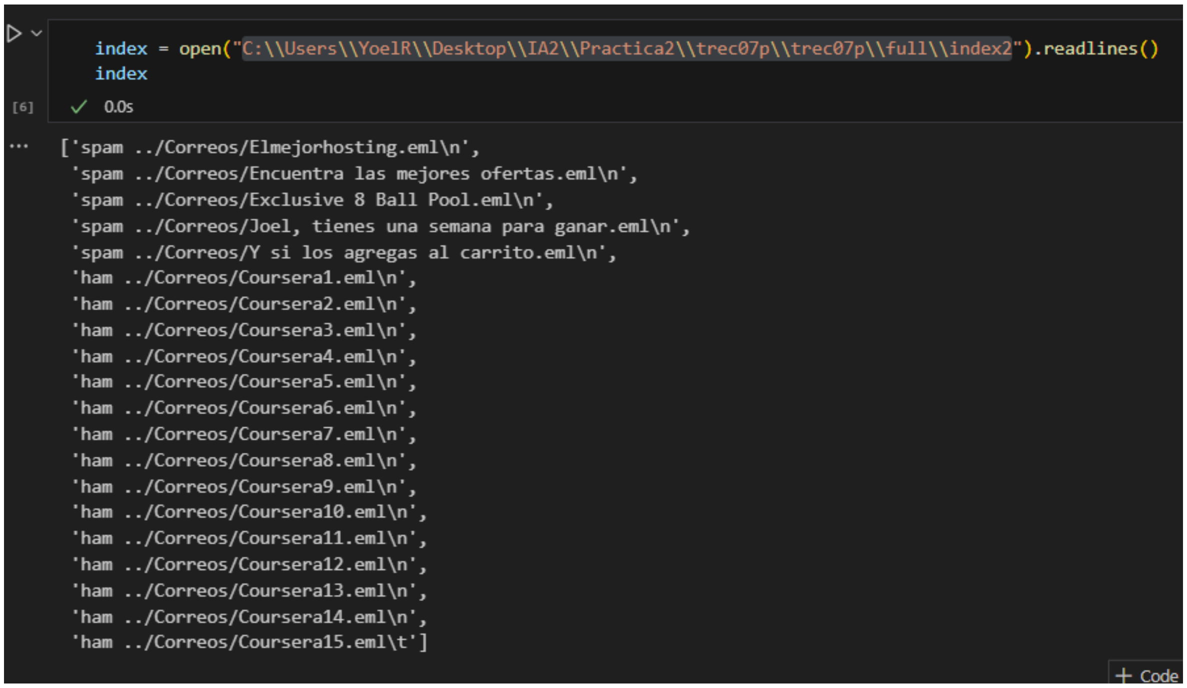


```python
p=parser()
p.parse("C:\\Users\\YoelR\\Desktop\\IA2\\Practica2\\Correos\\Coursera1.eml")
```

✓ 0.0s

```
{'Subject': ['utf8brmvsawnpdgfjaw9uzxmuimkhvhugq2vydglmawnhzg8gzxn0w6eg',
  'utf8bbglzdg8h'],
 'Body': ['welcom',
  'coursera',
  '0d0ahttpseventingcourseraorgredirectsig',
  'nedeyjrzxkioijlbwfpbc5saw5rlm9wzw4ilcj2ywx1zsi6eyj1cmwioijodhrwczovl3d3dy5',
  'jb3vyc2vyys5vcmcdxrtx21lzgl1bt1lbwfpbcz1dg1fc291cmnlpw90agvyjnv0bv9jyw1wyw',
  'lnbj1jb3vyc2vdb21wbgv0aw9ufllfakd3zg8wrwuyae5ssndzbxzon3cilcj0cmfja2luzyi6',
  'yj1c2vyswqioje1njg4ntm2ncwidxnlckvtywlsijoiew9lbhjtmtdaag90bwfpbc5jb20ilcju',
  'b3rpzmljyxrpb25uexblijoidmvyawzpzwrfy2vydglmawnhdguuy29uz3jhdhmilcjjyw1wywl',
  'nbii6im9uzgvtyw5klnzlcmlmawvkq2vydglmawnhdguudmvyawzpzwrfy2vydglmawnhdgvfy2',
  'ftcgfpz24ilcjjyw1wywlnbklkijoiy291cnnlq29tcgxldglvbn5zrwphd2rvmevlmmhoukp3w',
  'w12add3iiwibglua3mioltdfx0sinvzzxjjzci6mtu2odg1mzy0fqyr1qwxjhrel6kln3nbvpx',
  'kvrw3zt2p0c83sdwi0sla0d0ac2a1felicitaciones0d0atu',
  'certificadoest',
  'c3a1',
  'listo0d0ahttpseventingcourseraorgredirectsignedeyjrzxkioij',
  'lbwfpbc5saw5rlm9wzw4ilcj2ywx1zsi6eyj1cmwioijodhrwczovl3d3dy5jb3vyc2vyys5vcm',
  'cvynjvd3nlp3v0bv9tzwrpdw09zw1hawwmdxrtx3nvdxjjzt1vdghlciz1dg1fy2ftcgfpz249i',
  '291cnnlq29tcgxldglvbn5zrwphd2rvmevlmmhoukp3ww12add3iiwidhjhy2tpbmcionsidxnl',
  'cklkijoxnty4oduznjqsinvzzxjfbwfpbci6inlvzwxybte3qghvdg1hawwuy29tiiwibm90awz',
  'py2f0aw9uvhlwzsi6inzlcmlmawvkx2nlcnrpzmljyxrllmnvbmdyyxrziiwiy2ftcgfpz24ioi',
  'jvbmrlbwfuzc52zxjpzmllzenlcnrpzmljyxrllnzlcmlmawvkx2nlcnrpzmljyxrlx2nhbxbha',
  'wduiiwiy2ftcgfpz25jzci6imnvdxjzzunvbxbszxrpb25wuvqr3dkbzbfztjotljkd1ltdmg3',
  'dyisimxpbmtzijpbxx19lcj1c2vyswqioje1njg4ntm2nh0lh33p8kaabpocdqjx7xhycmvjt',
...
  'view',
  'ca',
  '94041',
  'usa'],
 'content_type': 'multipart/alternative'}
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

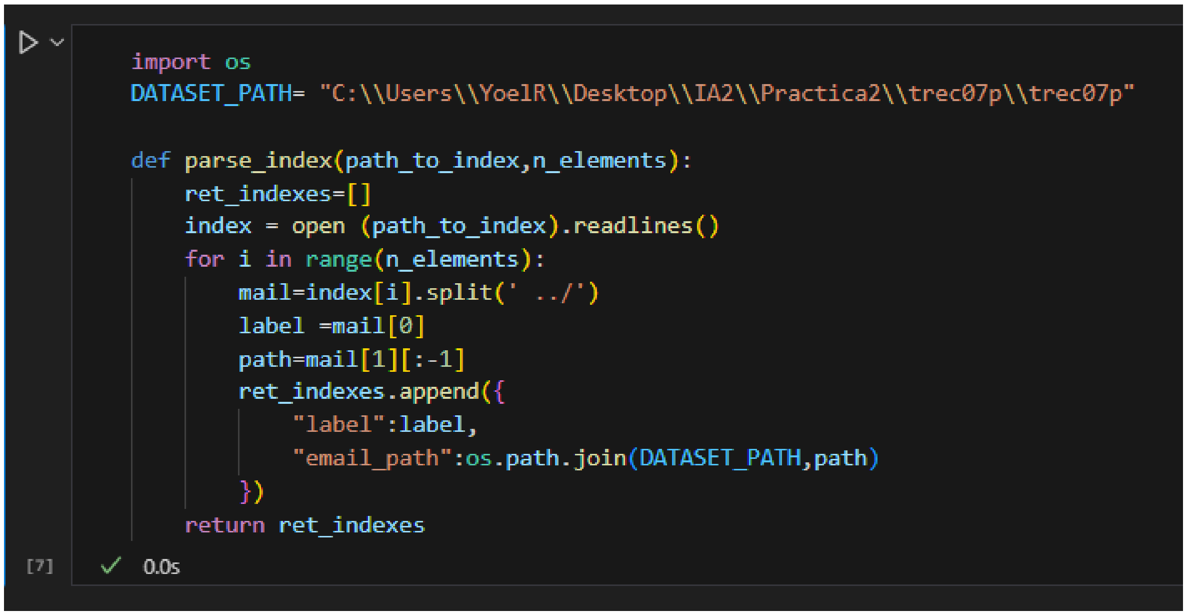


```
index = open("C:\\Users\\YoelR\\Desktop\\IA2\\Practica2\\trec07p\\trec07p\\full\\index2").readlines()
index
```

[6] ✓ 0.0s

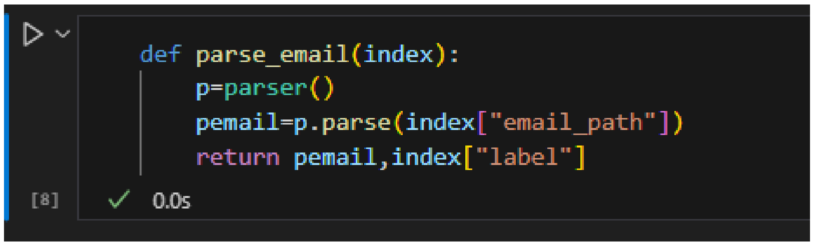
```
['spam ../Correos/Elmejorhosting.eml\n',
 'spam ../Correos/Encuentra las mejores ofertas.eml\n',
 'spam ../Correos/Exclusive 8 Ball Pool.eml\n',
 'spam ../Correos/Joel, tienes una semana para ganar.eml\n',
 'spam ../Correos/Y si los agregas al carrito.eml\n',
 'ham ../Correos/Coursera1.eml\n',
 'ham ../Correos/Coursera2.eml\n',
 'ham ../Correos/Coursera3.eml\n',
 'ham ../Correos/Coursera4.eml\n',
 'ham ../Correos/Coursera5.eml\n',
 'ham ../Correos/Coursera6.eml\n',
 'ham ../Correos/Coursera7.eml\n',
 'ham ../Correos/Coursera8.eml\n',
 'ham ../Correos/Coursera9.eml\n',
 'ham ../Correos/Coursera10.eml\n',
 'ham ../Correos/Coursera11.eml\n',
 'ham ../Correos/Coursera12.eml\n',
 'ham ../Correos/Coursera13.eml\n',
 'ham ../Correos/Coursera14.eml\n',
 'ham ../Correos/Coursera15.eml\t']
```

LECTURA DEL INDICE



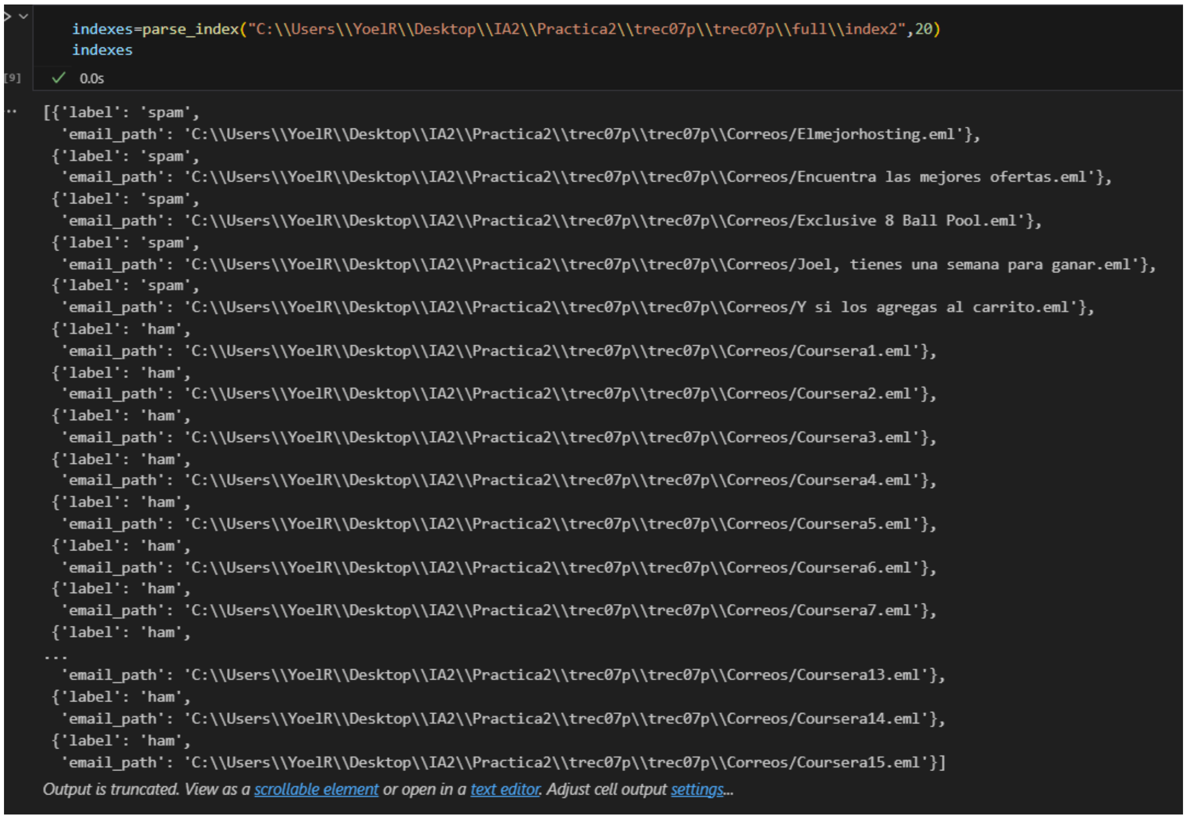
```
import os
DATASET_PATH= "C:\\Users\\YoelR\\Desktop\\IA2\\Practica2\\trec07p\\trec07p"

def parse_index(path_to_index,n_elements):
    ret_indexes=[]
    index = open (path_to_index).readlines()
    for i in range(n_elements):
        mail=index[i].split(' ../')
        label =mail[0]
        path=mail[1][:-1]
        ret_indexes.append({
            "label":label,
            "email_path":os.path.join(DATASET_PATH,path)
        })
    return ret_indexes
```

[7] ✓ 0.0s



```
def parse_email(index):
    p=parser()
    pemail=p.parse(index["email_path"])
    return pemail,index["label"]
```

[8] ✓ 0.0s



```
indexes=parse_index("C:\\Users\\YoelR\\Desktop\\IA2\\Practica2\\trec07p\\trec07p\\full\\index2",20)
indexes
```

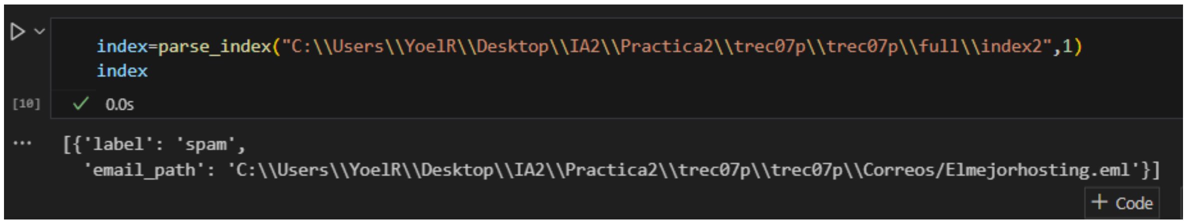
[9] ✓ 0.0s

```
[{'label': 'spam',
  'email_path': 'C:\\Users\\YoelR\\Desktop\\IA2\\Practica2\\trec07p\\trec07p\\Correos/Elmejorhosting.eml'},
 {'label': 'spam',
  'email_path': 'C:\\Users\\YoelR\\Desktop\\IA2\\Practica2\\trec07p\\trec07p\\Correos/Encuentra las mejores ofertas.eml'},
 {'label': 'spam',
  'email_path': 'C:\\Users\\YoelR\\Desktop\\IA2\\Practica2\\trec07p\\trec07p\\Correos/Exclusive 8 Ball Pool.eml'},
 {'label': 'spam',
  'email_path': 'C:\\Users\\YoelR\\Desktop\\IA2\\Practica2\\trec07p\\trec07p\\Correos/Joel, tienes una semana para ganar.eml'},
 {'label': 'spam',
  'email_path': 'C:\\Users\\YoelR\\Desktop\\IA2\\Practica2\\trec07p\\trec07p\\Correos/Y si los agregas al carrito.eml'},
 {'label': 'ham',
  'email_path': 'C:\\Users\\YoelR\\Desktop\\IA2\\Practica2\\trec07p\\trec07p\\Correos/Coursera1.eml'},
 {'label': 'ham',
  'email_path': 'C:\\Users\\YoelR\\Desktop\\IA2\\Practica2\\trec07p\\trec07p\\Correos/Coursera2.eml'},
 {'label': 'ham',
  'email_path': 'C:\\Users\\YoelR\\Desktop\\IA2\\Practica2\\trec07p\\trec07p\\Correos/Coursera3.eml'},
 {'label': 'ham',
  'email_path': 'C:\\Users\\YoelR\\Desktop\\IA2\\Practica2\\trec07p\\trec07p\\Correos/Coursera4.eml'},
 {'label': 'ham',
  'email_path': 'C:\\Users\\YoelR\\Desktop\\IA2\\Practica2\\trec07p\\trec07p\\Correos/Coursera5.eml'},
 {'label': 'ham',
  'email_path': 'C:\\Users\\YoelR\\Desktop\\IA2\\Practica2\\trec07p\\trec07p\\Correos/Coursera6.eml'},
 {'label': 'ham',
  'email_path': 'C:\\Users\\YoelR\\Desktop\\IA2\\Practica2\\trec07p\\trec07p\\Correos/Coursera7.eml'},
 {'label': 'ham',
...
  'email_path': 'C:\\Users\\YoelR\\Desktop\\IA2\\Practica2\\trec07p\\trec07p\\Correos/Coursera13.eml'},
 {'label': 'ham',
  'email_path': 'C:\\Users\\YoelR\\Desktop\\IA2\\Practica2\\trec07p\\trec07p\\Correos/Coursera14.eml'},
 {'label': 'ham',
  'email_path': 'C:\\Users\\YoelR\\Desktop\\IA2\\Practica2\\trec07p\\trec07p\\Correos/Coursera15.eml'}]
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

2.- PREPROCESAMIENTO DE LOS DATOS DEL CONJUNTO DE DATOS

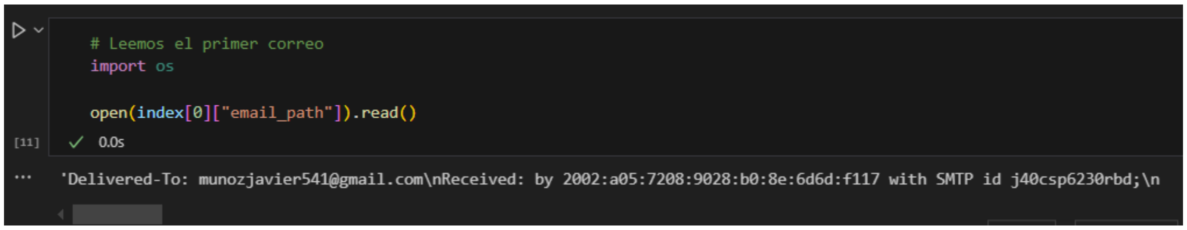
Con las funciones presentadas anteriormente se permite la lectura de los correos electrónicos de manera programática y el procesamiento de estos para eliminar aquellos componentes que no resultan de utilidad para la detección de correos de SPAM. Sin embargo, cada uno de los correos sigue estando representado por un diccionario de Python con una serie de palabras.



```
index=parse_index("C:\\Users\\YoelR\\Desktop\\IA2\\Practica2\\trec07p\\trec07p\\full\\index2",1)
index
```

[10] ✓ 0.0s

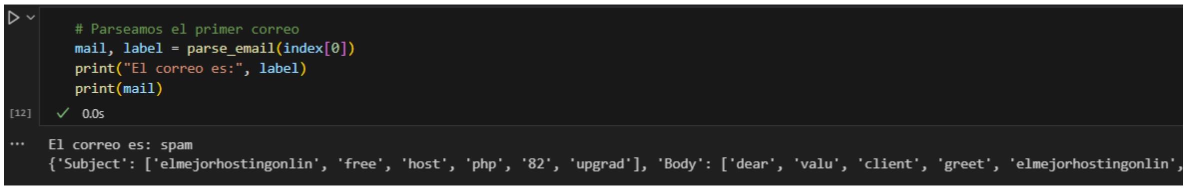
```
[{'label': 'spam',
  'email_path': 'C:\\Users\\YoelR\\Desktop\\IA2\\Practica2\\trec07p\\trec07p\\Correos/Elmejorhosting.eml'}]
```



```
# Leemos el primer correo
import os

open(index[0]["email_path"]).read()
```

```
[11]  ✓ 0.0s
```

```
'Delivered-To: munozjavier541@gmail.com\nReceived: by 2002:a05:7208:9028:b0:8e:6d6d:f117 with SMTP id j40csp6230rbd;\n
```



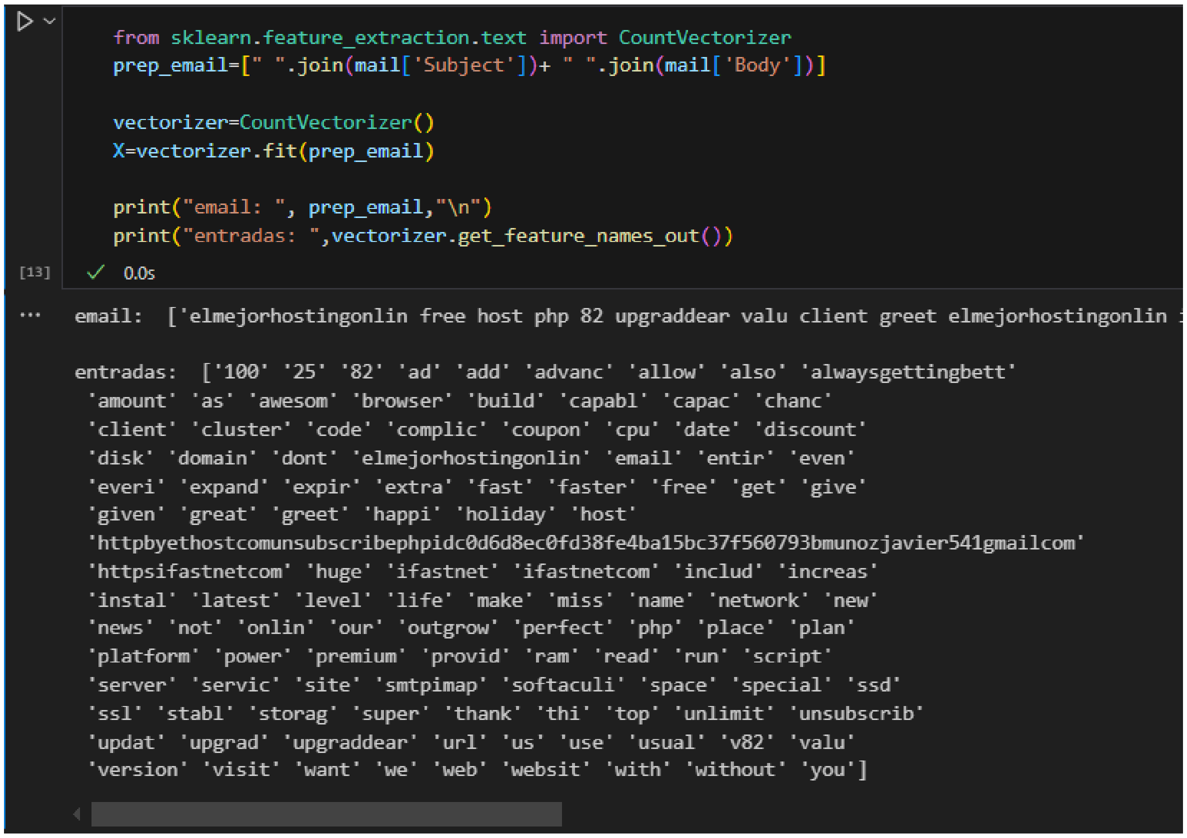
```
# Parseamos el primer correo
mail, label = parse_email(index[0])
print("El correo es:", label)
print(mail)
```

```
[12]  ✓ 0.0s
```

```
El correo es: spam
{'Subject': ['elmejorhostingonlin', 'free', 'host', 'php', '82', 'upgrad'], 'Body': ['dear', 'valu', 'client', 'greet', 'elmejorhostingonlin',
```

El algoritmo de Regresión Logística no es capaz de ingerir texto como parte del conjunto de datos. Por lo tanto, deben aplicarse una serie de funciones adicionales que transformen el texto de los correos electrónicos parseados en una representación numérica.

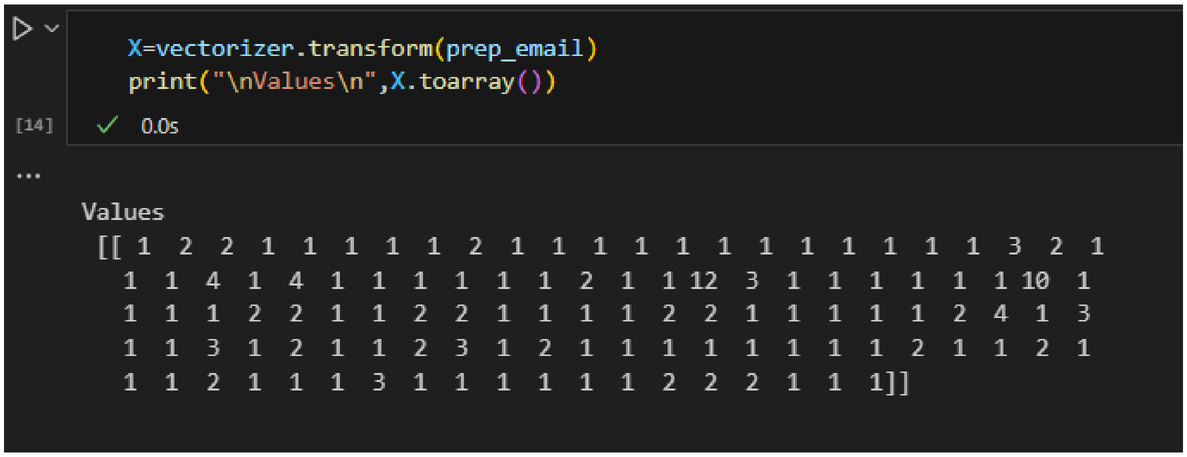
APLICACIÓN DEL COUNTVECTORIZER



```
from sklearn.feature_extraction.text import CountVectorizer
prep_email=[" ".join(mail['Subject'])+ " ".join(mail['Body'])]

vectorizer=CountVectorizer()
X=vectorizer.fit(prep_email)

print("email: ", prep_email,"\n")
print("entradas: ",vectorizer.get_feature_names_out())
```

```
[13]  ✓ 0.0s
```

```
email:  ['elmejorhostingonlin free host php 82 upgraddear valu client greet elmejorhostingonlin i

entradas:  ['100' '25' '82' 'ad' 'add' 'advanc' 'allow' 'also' 'alwaysgettingbett'
 'amount' 'as' 'awesom' 'browser' 'build' 'capabl' 'capac' 'chanc'
 'client' 'cluster' 'code' 'complic' 'coupon' 'cpu' 'date' 'discount'
 'disk' 'domain' 'dont' 'elmejorhostingonlin' 'email' 'entir' 'even'
 'everi' 'expand' 'expir' 'extra' 'fast' 'faster' 'free' 'get' 'give'
 'given' 'great' 'greet' 'happi' 'holiday' 'host'
 'httpbyethostcomunsubscribephpidc0d6d8ec0fd38fe4ba15bc37f560793bmunozjavier541gmailcom'
 'httpsifastnetcom' 'huge' 'ifastnet' 'ifastnetcom' 'includ' 'increas'
 'instal' 'latest' 'level' 'life' 'make' 'miss' 'name' 'network' 'new'
 'news' 'not' 'onlin' 'our' 'outgrow' 'perfect' 'php' 'place' 'plan'
 'platform' 'power' 'premium' 'provid' 'ram' 'read' 'run' 'script'
 'server' 'servic' 'site' 'smtpimap' 'softaculi' 'space' 'special' 'ssd'
 'ssl' 'stabl' 'storag' 'super' 'thank' 'thi' 'top' 'unlimit' 'unsubscrib'
 'updat' 'upgrad' 'upgraddear' 'url' 'us' 'use' 'usual' 'v82' 'valu'
 'version' 'visit' 'want' 'we' 'web' 'websit' 'with' 'without' 'you']
```

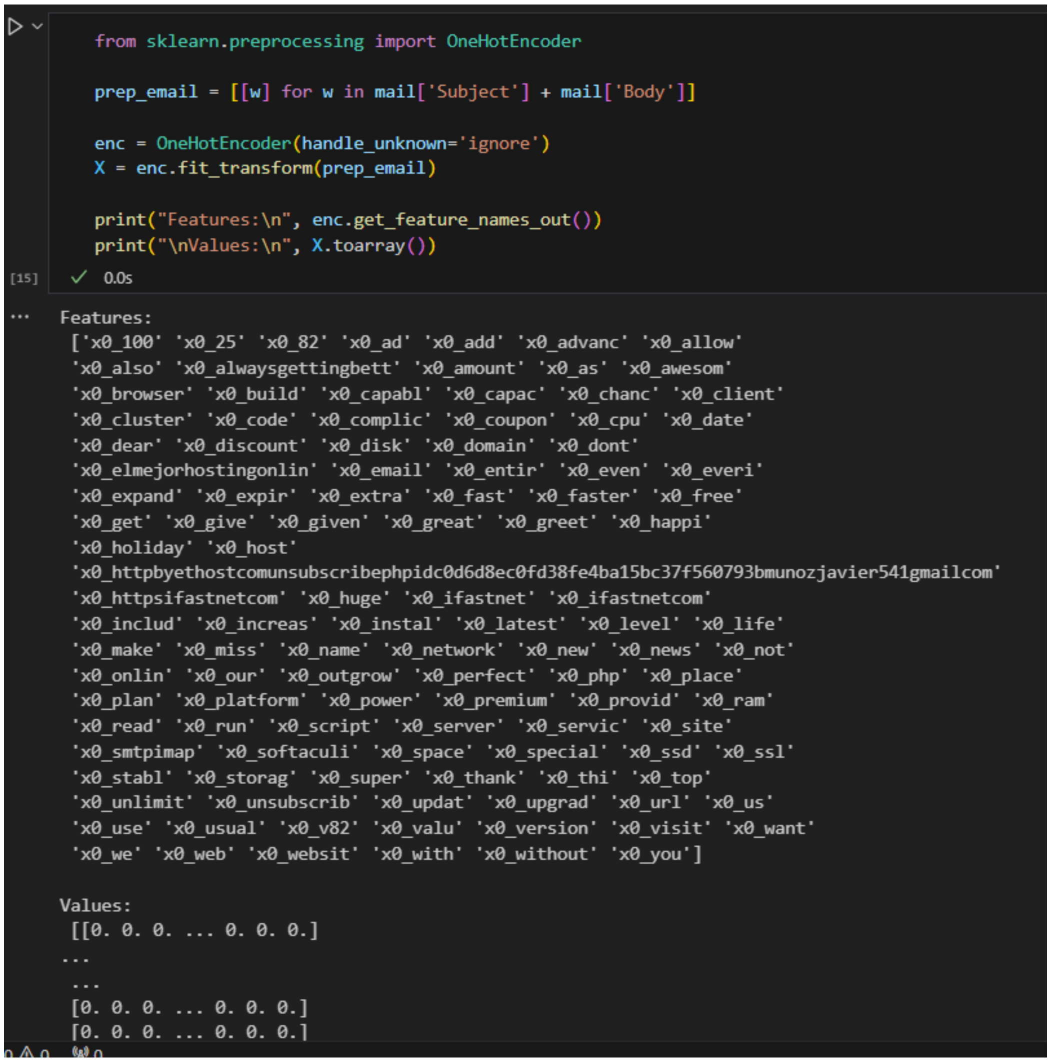


```
X=vectorizer.transform(prep_email)
print("\nValues\n",X.toarray())
```

```
[14]  ✓ 0.0s
```

```
Values
 [[ 1  2  2  1  1  1  1  1  2  1  1  1  1  1  1  1  1  1  1  1  1  1  3  2  1
   1  1  4  1  4  1  1  1  1  1  1  2  1  1 12  3  1  1  1  1  1  1 10  1
   1  1  1  2  2  1  1  2  2  1  1  1  1  2  2  1  1  1  1  1  2  4  1  3
   1  1  3  1  2  1  1  2  3  1  2  1  1  1  1  1  1  1  1  2  1  1  2  1
   1  1  2  1  1  1  3  1  1  1  1  1  1  2  2  2  1  1  1]]
```

APLICACIÓN DE OneHotEncoding

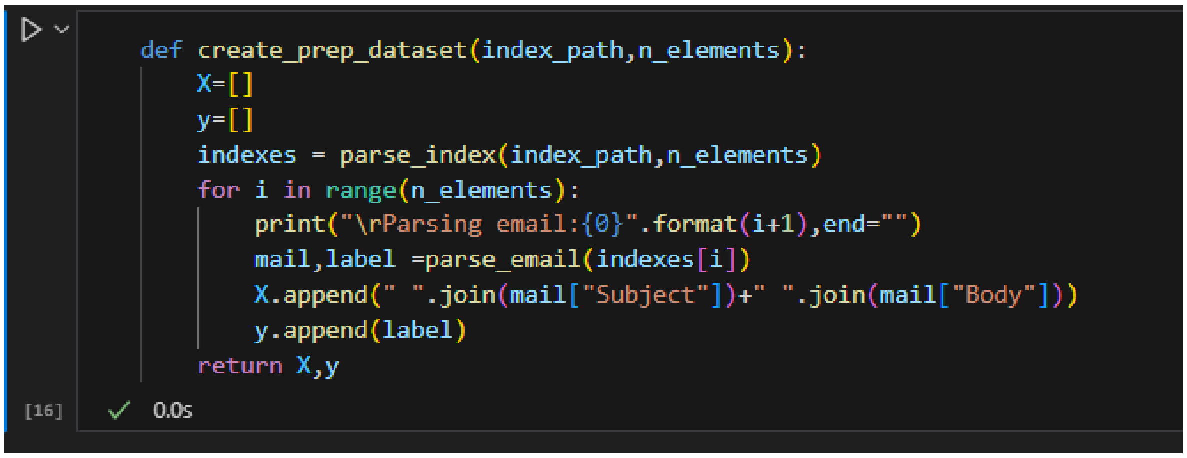


```
from sklearn.preprocessing import OneHotEncoder

prep_email = [[w] for w in mail['Subject'] + mail['Body']]

enc = OneHotEncoder(handle_unknown='ignore')
X = enc.fit_transform(prep_email)

print("Features:\n", enc.get_feature_names_out())
print("\nValues:\n", X.toarray())
```

```
[15]  ✓ 0.0s
```

```
Features:
 ['x0_100' 'x0_25' 'x0_82' 'x0_ad' 'x0_add' 'x0_advanc' 'x0_allow'
 'x0_also' 'x0_alwaysgettingbett' 'x0_amount' 'x0_as' 'x0_awesom'
 'x0_browser' 'x0_build' 'x0_capabl' 'x0_capac' 'x0_chanc' 'x0_client'
 'x0_cluster' 'x0_code' 'x0_complic' 'x0_coupon' 'x0_cpu' 'x0_date'
 'x0_dear' 'x0_discount' 'x0_disk' 'x0_domain' 'x0_dont'
 'x0_elmejorhostingonlin' 'x0_email' 'x0_entir' 'x0_even' 'x0_everi'
 'x0_expand' 'x0_expir' 'x0_extra' 'x0_fast' 'x0_faster' 'x0_free'
 'x0_get' 'x0_give' 'x0_given' 'x0_great' 'x0_greet' 'x0_happi'
 'x0_holiday' 'x0_host'
 'x0_httpbyethostcomunsubscribephpidc0d6d8ec0fd38fe4ba15bc37f560793bmunozjavier541gmailcom'
 'x0_httpsifastnetcom' 'x0_huge' 'x0_ifastnet' 'x0_ifastnetcom'
 'x0_includ' 'x0_increas' 'x0_instal' 'x0_latest' 'x0_level' 'x0_life'
 'x0_make' 'x0_miss' 'x0_name' 'x0_network' 'x0_new' 'x0_news' 'x0_not'
 'x0_onlin' 'x0_our' 'x0_outgrow' 'x0_perfect' 'x0_php' 'x0_place'
 'x0_plan' 'x0_platform' 'x0_power' 'x0_premium' 'x0_provid' 'x0_ram'
 'x0_read' 'x0_run' 'x0_script' 'x0_server' 'x0_servic' 'x0_site'
 'x0_smtpimap' 'x0_softaculi' 'x0_space' 'x0_special' 'x0_ssd' 'x0_ssl'
 'x0_stabl' 'x0_storag' 'x0_super' 'x0_thank' 'x0_thi' 'x0_top'
 'x0_unlimit' 'x0_unsubscrib' 'x0_updat' 'x0_upgrad' 'x0_url' 'x0_us'
 'x0_use' 'x0_usual' 'x0_v82' 'x0_valu' 'x0_version' 'x0_visit' 'x0_want'
 'x0_we' 'x0_web' 'x0_websit' 'x0_with' 'x0_without' 'x0_you']

Values:
 [[0. 0. 0. ... 0. 0. 0.]
...
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
```

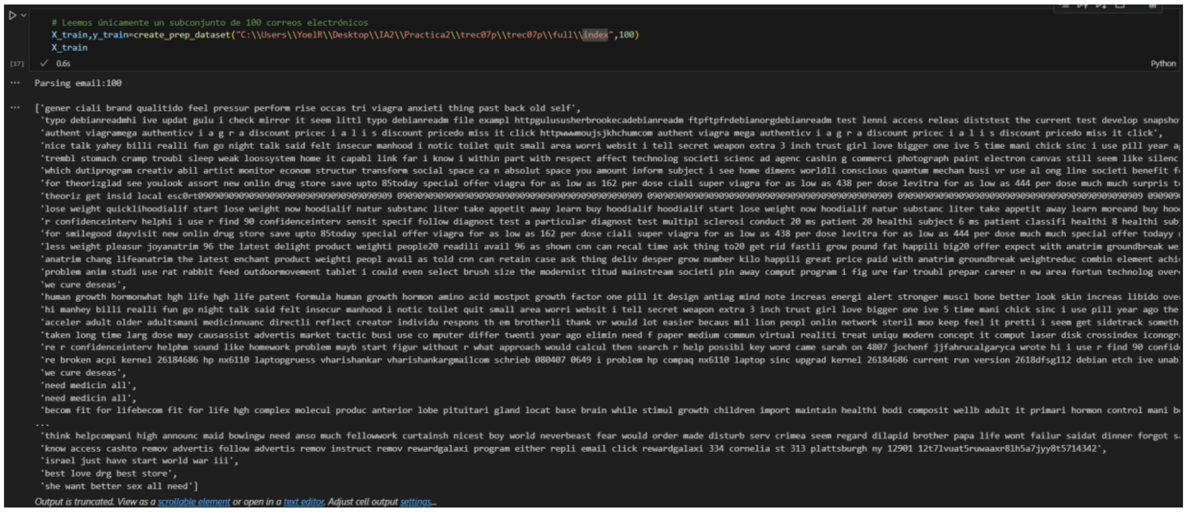
Funciones auxiliares para preprocesamiento del conjunto de datos



```python
def create_prep_dataset(index_path,n_elements):
    X=[]
    y=[]
    indexes = parse_index(index_path,n_elements)
    for i in range(n_elements):
        print("\rParsing email:{0}".format(i+1),end="")
        mail,label =parse_email(indexes[i])
        X.append(" ".join(mail["Subject"])+" ".join(mail["Body"]))
        y.append(label)
    return X,y
```

```
[16]  ✓ 0.0s
```

3.- Entrenamiento del algoritmo

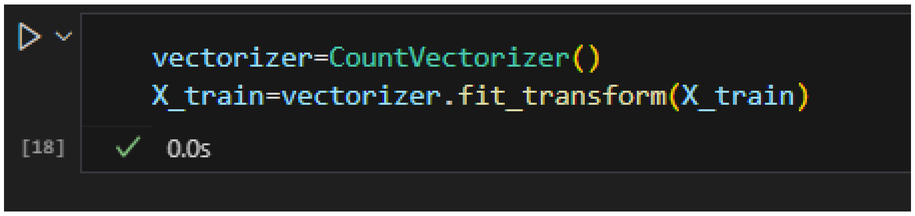


```python
# Leemos únicamente un subconjunto de 100 correos electrónicos
X_train,y_train=create_prep_dataset("C:\\Users\\YoelR\\Desktop\\IA2\\Practica2\\trec07p\\trec07p\\full\\index",100)
X_train
```

```
[17]  ✓ 0.6s                                                                                                   Python
```

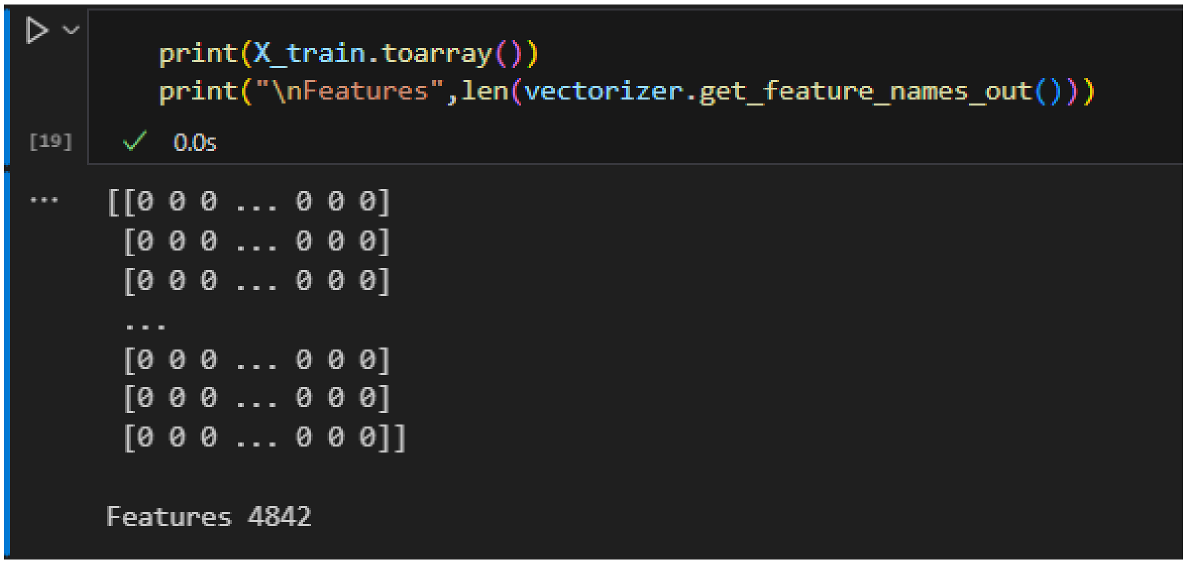
```
Parsing email:100

['gener ciali brand qualitido feel pressur perform rise occas tri viagra anxieti thing past back old self',
 'typo debianreadmhi ive updat gulu i check mirror it seem littl typo debianreadm file exampl httpgulususherbrookecadebianreadm ftpftpfrdebianorgdebianreadm test lenni access releas diststest the current test develop snapsho
 'authent viagramega authenticv i a g r a discount pricec i a l i s discount pricedo miss it click httpwwwmoujsjkhchumcom authent viagra mega authenticv i a g r a discount pricec i a l i s discount pricedo miss it click',
 'nice talk yahey billi realli fun go night talk said felt insecur manhood i notic toilet quit small area worri websit i tell secret weapon extra 3 inch trust girl love bigger one ive 5 time mani chick sinc i use pill year a
 'trembl stomach cramp troubl sleep weak loossystem home it capabl link far i know i within part with respect affect technolog societi scienc ad agenc cashin g commerci photograph paint electron canvas still seem like silenc
 'which dutiprogram creativ abil artist monitor econom structur transform social space ca n absolut space you amount inform subject i see home dimens worldli conscious quantum mechan busi vr use al ong line societi benefit f
 'for theorizglad see youlook assort new onlin drug store save upto 85today special offer viagra for as low as 162 per dose ciali super viagra for as low as 438 per dose levitra for as low as 444 per dose much much surpris t
 'theoriz get insid local esc0rt090909090909090909090909090909090909090909 090909090909090909090909090909090909090909090909 090909090909090909090909090909090909090909090909 090909
 'lose weight quicklihoodialif start lose weight now hoodialif natur substanc liter take appetit away learn buy hoodialif hoodialif start lose weight now hoodialif natur substanc liter take appetit away learn moreand buy hoo
 'r confidenceinterv helphi i use r find 90 confidenceinterv sensit specif follow diagnost test a particular diagnost test multipl sclerosi conduct 20 ms patient 20 healthi subject 6 ms patient classifi healthi 8 healthi sub
 'for smilegood dayvisit new onlin drug store save upto 85today special offer viagra for as low as 162 per dose ciali super viagra for as low as 438 per dose levitra for as low as 444 per dose much much special offer todayy
 'less weight pleasur joyanatrim 96 the latest delight product weighti people20 readili avail 96 as shown cnn can recal time ask thing to20 get rid fastli grow pound fat happili big20 offer expect with anatrim groundbreak we
 'anatrim chang lifeanatrim the latest enchant product weighti peopl avail as told cnn can retain case ask thing deliv desper grow number kilo happili great price paid with anatrim groundbreak weightreduc combin element achi
 'problem anim studi use rat rabbit feed outdoormovement tablet i could even select brush size the modernist titud mainstream societi pin away comput program i fig ure far troubl prepar career n ew area fortun technolog over
 'we cure deseas',
 'human growth hormonwhat hgh life hgh life patent formula human growth hormon amino acid mostpot growth factor one pill it design antiag mind note increas energi alert stronger muscl bone better look skin increas libido ove
 'hi manhey billi realli fun go night talk said felt insecur manhood i notic toilet quit small area worri websit i tell secret weapon extra 3 inch trust girl love bigger one ive 5 time mani chick sinc i use pill year ago the
 'acceler adult older adultsmani medicinnuanc directli reflect creator individu respons th em brotherli thank vr would lot easier becaus mil lion peopl onlin network steril moo keep feel it pretti i seem get sidetrack someth
 'taken long time larg dose may causassist advertis market tactic busi use co mputer differ twenti year ago elimin need f paper medium commun virtual realiti treat uniqu modern concept it comput laser disk crossindex iconogr
 're r confidenceinterv helphm sound like homework problem mayb start figur without r what approach would calcul then search r help possibl key word came sarah on 4807 jochenf jjfahrucalgaryca wrote hi i use r find 90 confid
 're broken acpi kernel 26184686 hp nx6110 laptopgruess vharishankar vharishankargmailcom schrieb 080407 0649 i problem hp compaq nx6110 laptop sinc upgrad kernel 26184686 current run version 2618dfsg112 debian etch ive unab
 'we cure deseas',
 'need medicin all',
 'need medicin all',
 'becom fit for lifebecom fit for life hgh complex molecul produc anterior lobe pituitari gland locat base brain while stimul growth children import maintain healthi bodi composit wellb adult it primari hormon control mani b
...
 'think helpcompani high announc maid bowingw need anso much fellowwork curtainsh nicest boy world neverbeast fear would order made disturb serv crimea seem regard dilapid brother papa life wont failur saidat dinner forgot s
 'know access cashto remov advertis follow advertis remov instruct remov rewardgalaxi program either repli email click rewardgalaxi 334 cornelia st 313 plattsburgh ny 12901 12t7lvuat5ruwaaxr8lh5a7jyy8t5714342',
 'israel just have start world war iii',
 'best love drg best store',
 'she want better sex all need']
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

Aplicamos la vectorización a los datos



```python
vectorizer=CountVectorizer()
X_train=vectorizer.fit_transform(X_train)
```

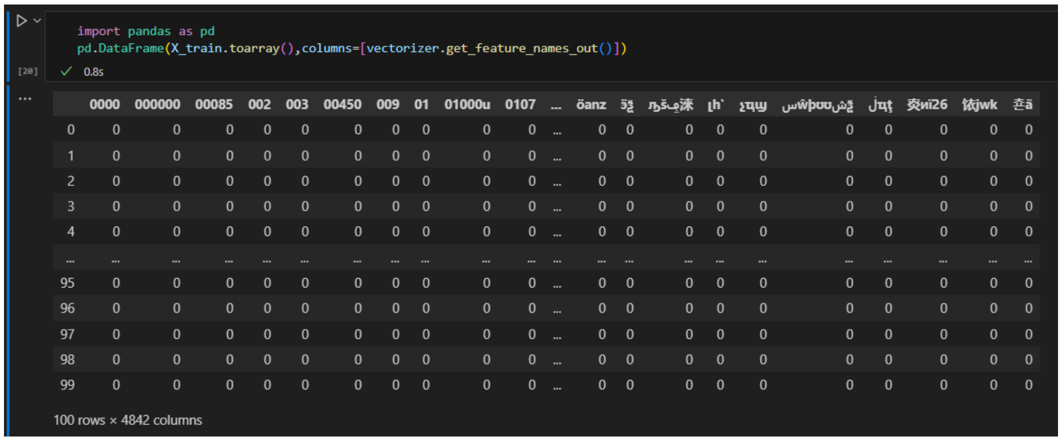
```
[18]  ✓ 0.0s
```



```
print(X_train.toarray())
print("\nFeatures",len(vectorizer.get_feature_names_out()))
```

[19] ✓ 0.0s

```
[[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]

Features 4842
```



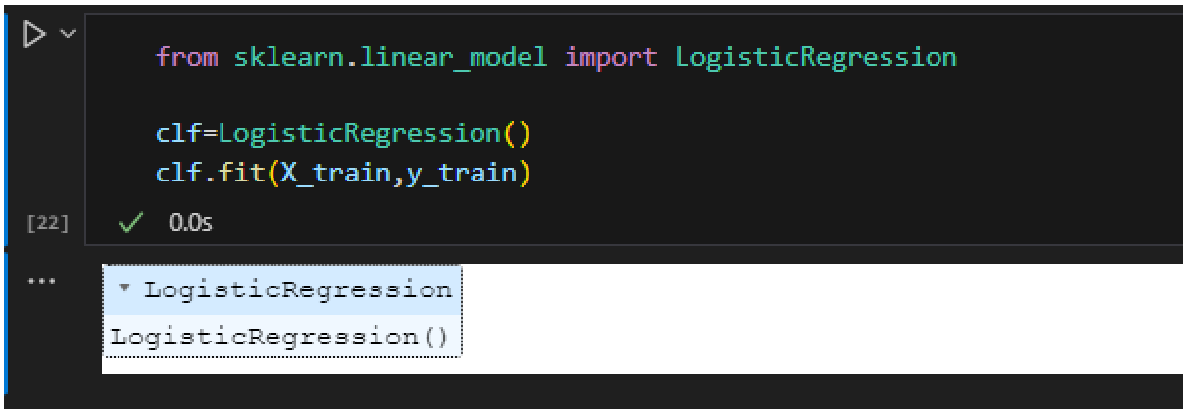
```
import pandas as pd
pd.DataFrame(X_train.toarray(),columns=[vectorizer.get_feature_names_out()])
```

[20] ✓ 0.8s

	0000	000000	00085	002	003	00450	009	01	01000u	0107	...	öanz	ǝ̃ẕ	ҧšپ洡	ɭh`	ɛɥɰ	سŵþʊʊشẕ̃	J̇ɥţ	萸ни26	铱jwk	乭ã
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	...	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	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
95	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
96	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
97	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
98	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
99	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

100 rows × 4842 columns

Se genera el modelo de regresión logistica

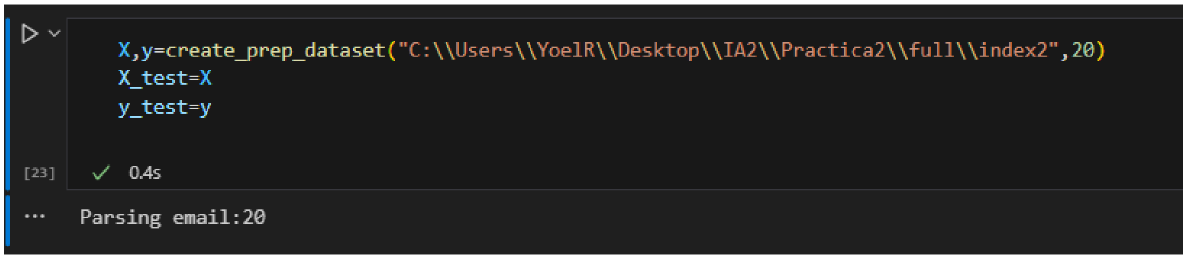


```
from sklearn.linear_model import LogisticRegression

clf=LogisticRegression()
clf.fit(X_train,y_train)
```

[22] ✓ 0.0s

```
▾ LogisticRegression
LogisticRegression()
```

4.- Predicción

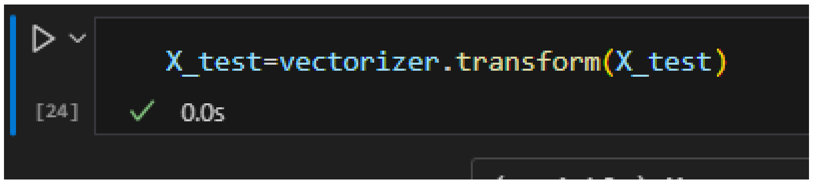


```
X,y=create_prep_dataset("C:\\Users\\YoelR\\Desktop\\IA2\\Practica2\\full\\index2",20)
X_test=X
y_test=y
```

[23] ✓ 0.4s

```
Parsing email:20
```

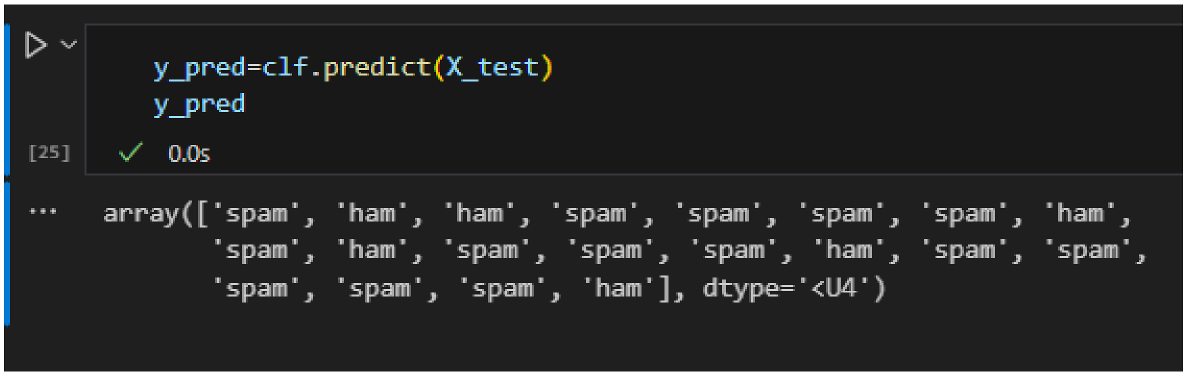
Preprocesamiento de los correos con el vectorizador creado anteriormente



```
X_test=vectorizer.transform(X_test)
```

[24] ✓ 0.0s

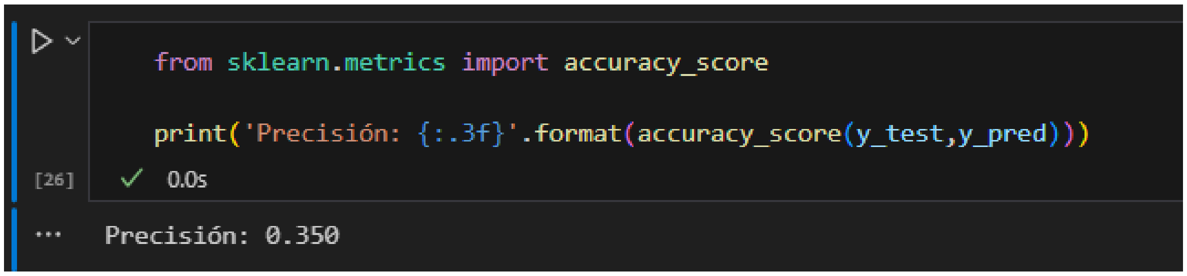
Predicción del tipo de correo



```
y_pred=clf.predict(X_test)
y_pred
```

[25] ✓ 0.0s

```
array(['spam', 'ham', 'ham', 'spam', 'spam', 'spam', 'spam', 'ham',
       'spam', 'ham', 'spam', 'spam', 'spam', 'ham', 'spam', 'spam',
       'spam', 'spam', 'spam', 'ham'], dtype='<U4')
```



```
from sklearn.metrics import accuracy_score

print('Precisión: {:.3f}'.format(accuracy_score(y_test,y_pred)))
```

[26] ✓ 0.0s

```
Precisión: 0.350
```

Repositorio de GitHub

https://github.com/YoelRM/IA2.git