Tarea 8

NLP

Descripción: Dados 3 textos construir la matriz término-documento y determinar la similitud coseno.

Alumno: Miguel Angel Soto Hernandez

Importaciones necesarias

```
# tratamiento de datos
import numpy as np
import pandas as pd
import string
import re

# preprocesado
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Datos

```
# lectura de datos
url = 'https://raw.githubusercontent.com/JoaquinAmatRodrigo/Estadistica-con-R/master/datos/'
tweets_elon   = pd.read_csv(url + "datos_tweets_@elonmusk.csv")
tweets_edlee  = pd.read_csv(url + "datos_tweets_@mayoredlee.csv")
tweets_bgates = pd.read_csv(url + "datos_tweets_@BillGates.csv")

print('Número de tweets @BillGates: ' + str(tweets_bgates.shape[0]))
print('Número de tweets @mayoredlee: ' + str(tweets_edlee.shape[0]))
print('Número de tweets @elonmusk: ' + str(tweets_elon.shape[0]))
```

```
Número de tweets @BillGates: 2087
Número de tweets @mayoredlee: 2447
Número de tweets @elonmusk: 2678
```

```
tweets_elon.head()
```

	screen_name	user_id	created_at	status_id	text	retweet_count	favorite_count	is_quote_status	qu
0	elonmusk	44196397	2017-11-09T17:28:57Z	9.286758e+17	"If one day, my words are against science, cho...	49919	104722	False	
1	elonmusk	44196397	2017-11-09T17:12:46Z	9.286717e+17	I placed the flowers\n\nThree broken ribs\nA p...	5940	33725	False	
2	elonmusk	44196397	2017-11-08T18:55:13Z	9.283351e+17	Atatürk Anıtkabir https://t.co/al3wt0njr6	34752	104823	False	
3	elonmusk	44196397	2017-11-07T19:48:45Z	9.279862e+17	@Bob_Richards One rocket, slightly toasted	415	7247	False	
4	elonmusk	44196397	2017-10-28T21:36:18Z	9.243894e+17	@uncover007 500 ft so far. Should be 2 miles l...	207	2128	False	

```
# se unen los dos dataframes en uno solo
tweets = pd.concat([tweets_elon, tweets_edlee, tweets_bgates], ignore_index=True)

# se seleccionan y renombran las columnas de interés
tweets = tweets[['screen_name', 'created_at', 'status_id', 'text']]
tweets.columns = ['autor', 'fecha', 'id', 'texto']
```

```
# parseo de fechas
tweets['fecha'] = pd.to_datetime(tweets['fecha'])
tweets.head()
```

	autor	fecha	id	texto
0	elonmusk	2017-11-09 17:28:57+00:00	9.286758e+17	"If one day, my words are against science, cho...
1	elonmusk	2017-11-09 17:12:46+00:00	9.286717e+17	I placed the flowers\n\nThree broken ribs\nA p...
2	elonmusk	2017-11-08 18:55:13+00:00	9.283351e+17	Atatürk Anıtkabir https://t.co/al3wt0njr6
3	elonmusk	2017-11-07 19:48:45+00:00	9.279862e+17	@Bob_Richards One rocket, slightly toasted
4	elonmusk	2017-10-28 21:36:18+00:00	9.243894e+17	@uncover007 500 ft so far. Should be 2 miles l...

Limpieza y tokenizacion

```
def limpiar_tokenizar(texto):
    '''
      Esta función limpia y tokeniza el texto en palabras individuales.
      El orden en el que se va limpiando el texto no es arbitrario.
      El listado de signos de puntuación se ha obtenido de: print(string.punctuation)
      y re.escape(string.punctuation)
    '''

    # se convierte todo el texto a minúsculas
    nuevo_texto = texto.lower()

    # eliminación de páginas web (palabras que empiezan por "http")
    nuevo_texto = re.sub('http\S+', ' ', nuevo_texto)

    # eliminación de signos de puntuación
    regex = '[\\!\\"\\#\\$\\%\\&\\\'\\(\\)\\*\\+\\,\\-\\.\\/\\:\\;\\<\\=\\>\\?\\@\\[\\\\\\]\\^_\\`\\{\\|\\}\\~]'
    nuevo_texto = re.sub(regex , ' ', nuevo_texto)

    # eliminación de números
    nuevo_texto = re.sub("\d+", ' ', nuevo_texto)

    # eliminación de espacios en blanco múltiples
    nuevo_texto = re.sub("\\s+", ' ', nuevo_texto)

    # tokenización por palabras individuales
    nuevo_texto = nuevo_texto.split(sep = ' ')

    # eliminación de tokens con una longitud < 2
    nuevo_texto = [token for token in nuevo_texto if len(token) > 1]

    return(nuevo_texto)
```

```
# prueba
test = '#HéroesSinCapa | El pasado 13 de marzo los oficiales Juárez, Morales y Castañeda adscritos a la #PBI, auxiliaron a los pacient
print(f'Ejemplo: {test}')
print(f'\nTexto limpio y por tokens: {limpiar_tokenizar(texto=test)}')
```

```
Ejemplo: #HéroesSinCapa | El pasado 13 de marzo los oficiales Juárez, Morales y Castañeda adscritos a la #PBI, auxiliaron a los p

Texto limpio y por tokens: ['héroessincapa', 'el', 'pasado', 'de', 'marzo', 'los', 'oficiales', 'juárez', 'morales', 'castañeda',
```

```
# se aplica la función de limpieza y tokenización a cada tweet
tweets['texto_tokenizado'] = tweets['texto'].apply(lambda x: limpiar_tokenizar(x))
tweets.head()
```

	autor	fecha	id	texto	texto_tokenizado
0	elonmusk	2017-11-09 17:28:57+00:00	9.286758e+17	"If one day, my words are against science, cho...	[if, one, day, my, words, are, against, scienc...
1	elonmusk	2017-11-09 17:12:46+00:00	9.286717e+17	I placed the flowers\n\nThree broken ribs\nA p...	[placed, the, flowers, three, broken, ribs, pi...
2	elonmusk	2017-11-08 18:55:13+00:00	9.283351e+17	Atatürk Anıtkabir https://t.co/al3wt0njr6	[atatürk, anıtkabir]
3	elonmusk	2017-11-07	9.279862e+17	@Bob_Richards One rocket, slightly	[bob, richards, one, rocket, slightly,

```
# transformando los arreglos en un solo string por fila
```

```
# transformando los arreglos en un solo string por fila
# explode: transformar cada elemento de una lista en una fila,
# replicando los valores del índice
tweets_tidy = tweets.explode(column='texto_tokenizado')
tweets_tidy = tweets_tidy.drop(columns='texto')
tweets_tidy = tweets_tidy.rename(columns={'texto_tokenizado':'token'})
tweets_tidy.head()
```

	autor	fecha	id	token
0	elonmusk	2017-11-09 17:28:57+00:00	9.286758e+17	if
0	elonmusk	2017-11-09 17:28:57+00:00	9.286758e+17	one
0	elonmusk	2017-11-09 17:28:57+00:00	9.286758e+17	day
0	elonmusk	2017-11-09 17:28:57+00:00	9.286758e+17	my
0	elonmusk	2017-11-09 17:28:57+00:00	9.286758e+17	words

```
# obtención de listado de stopwords del inglés
palabras_auxiliares = list(stopwords.words('english'))

# se añade la stoprword: amp, ax, ex
palabras_auxiliares.extend(("amp", "xa", "xe"))
print(f'Palabras auxiliares: {palabras_auxiliares[:20]}')
```

```
Palabras auxiliares: ['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd",
```

```
# limpiar palabras auxiliares
tweets_tidy = tweets_tidy[~(tweets_tidy['token'].isin(palabras_auxiliares))]
tweets_tidy.head()
```

	autor	fecha	id	token
0	elonmusk	2017-11-09 17:28:57+00:00	9.286758e+17	one
0	elonmusk	2017-11-09 17:28:57+00:00	9.286758e+17	day
0	elonmusk	2017-11-09 17:28:57+00:00	9.286758e+17	words
0	elonmusk	2017-11-09 17:28:57+00:00	9.286758e+17	science
0	elonmusk	2017-11-09 17:28:57+00:00	9.286758e+17	choose

Frecuencia de palabras

```
# palabras totales por usuario
print('Palabras totales por usuario')
tweets_tidy.groupby(by='autor')['token'].count()
```

```
Palabras totales por usuario
autor
BillGates     19445
elonmusk      21719
mayoredlee    26676
Name: token, dtype: int64
```

```
# palabras unicas por usuario
print('Palabras distintas por usuario')
tweets_tidy.groupby(by='autor')['token'].nunique()
```

```
Palabras distintas por usuario
autor
BillGates     4718
elonmusk      6496
mayoredlee    5644
Name: token, dtype: int64
```

Relacion coseno o correlacion

```
# pivotado de datos o matriz termino-documento
tweets_pivot = tweets_tidy.groupby(["autor","token"])["token"] \
                .agg(["count"]).reset_index() \
```

```
            .pivot(index = "token" , columns="autor", values= "count")
tweets_pivot.columns.name = None
tweets_pivot = tweets_pivot.fillna(0)
tweets_pivot[1000:1030]
```

token	BillGates	elonmusk	mayoredlee
beliefs	0.0	0.0	3.0
believe	12.0	14.0	4.0
believed	1.0	1.0	0.0
believen	0.0	0.0	9.0
believing	0.0	3.0	0.0
bell	0.0	0.0	1.0
bellevue	0.0	0.0	1.0
bells	0.0	1.0	1.0
belltower	0.0	1.0	0.0
belongs	1.0	0.0	1.0
beloved	0.0	0.0	2.0
belovedrevol	0.0	2.0	0.0
benbenwilde	0.0	1.0	0.0
bencam	0.0	1.0	0.0
benchmark	0.0	1.0	0.0
bending	2.0	0.0	0.0
benedictevans	0.0	1.0	0.0
benefical	0.0	1.0	0.0
benefit	1.0	2.0	8.0
benefiting	1.0	0.0	0.0
benefits	13.0	0.0	10.0
benefitting	0.0	0.0	1.0
benefi…	0.0	0.0	1.0
benfeldman	0.0	1.0	0.0
benioff	0.0	0.0	6.0
benjamin	0.0	0.0	1.0
benjamincoop	0.0	1.0	0.0
benjaminrphoto	0.0	1.0	0.0
benmacy	0.0	1.0	0.0
benny	0.0	1.0	0.0

```
# relación coseno por el uso y frecuencia de palabras
from scipy.spatial.distance import cosine

def similitud_coseno(a,b):
    distancia = cosine(a,b)
    return 1-distancia

tweets_pivot.corr(method=similitud_coseno)
```

	BillGates	elonmusk	mayoredlee
BillGates	1.000000	0.413110	0.279001
elonmusk	0.413110	1.000000	0.197927
mayoredlee	0.279001	0.197927	1.000000

```
# número de palabras comunes
palabras_elon = set(tweets_tidy[tweets_tidy.autor == 'elonmusk']['token'])
palabras_bill = set(tweets_tidy[tweets_tidy.autor == 'BillGates']['token'])
```

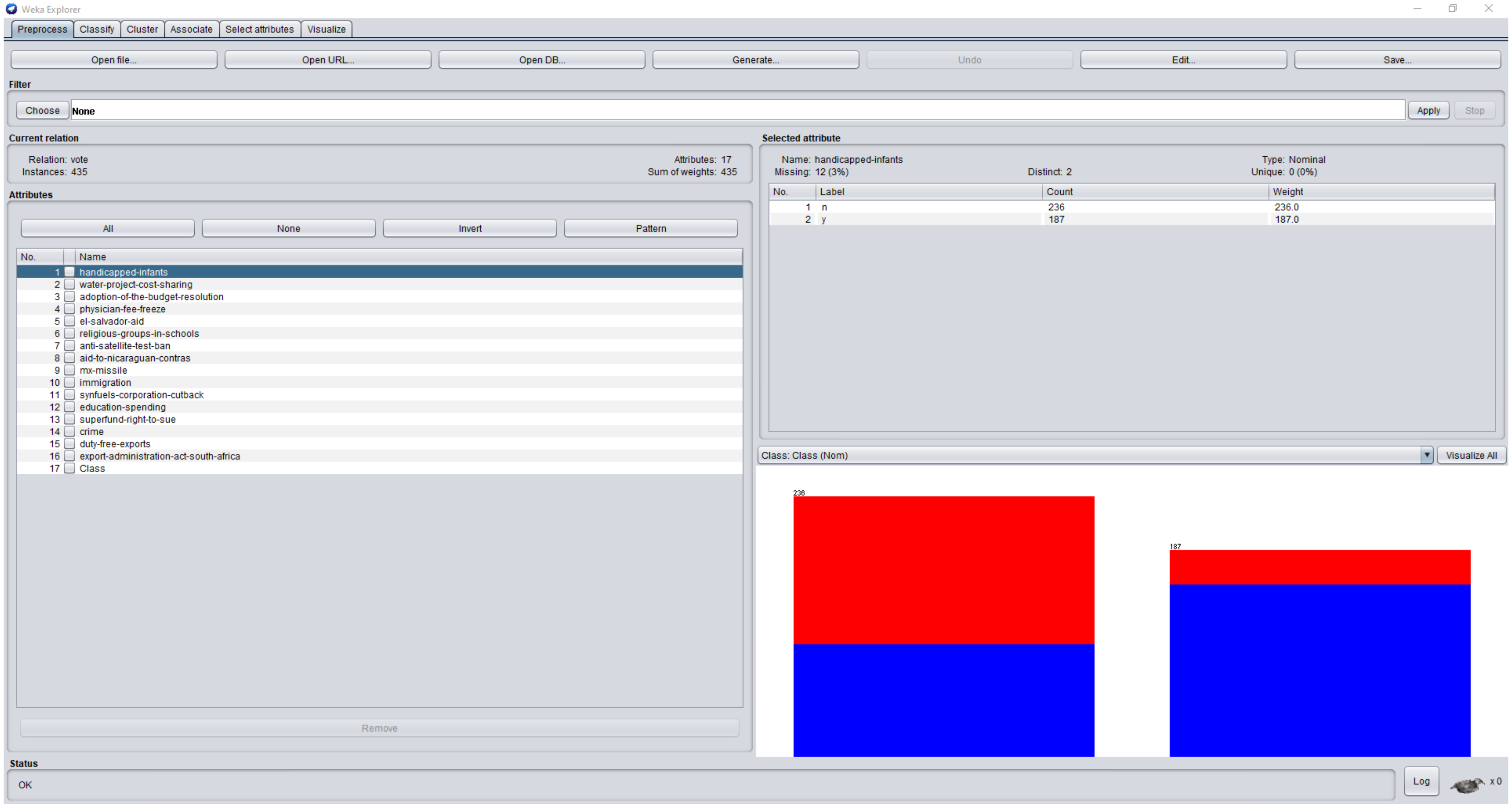
```
palabras_edlee = set(tweets_tidy[tweets_tidy.autor == 'mayoredlee']['token'])

print(f"Palabras comunes entre Elon Musk y Ed Lee:\
      {len(palabras_elon.intersection(palabras_edlee))}")
print(f"Palabras comunes entre Elon Musk y Bill Gates:\
      {len(palabras_elon.intersection(palabras_bill))}")
print(f"Palabras comunes entre Bill Gates y Ed Lee:\
      {len(palabras_bill.intersection(palabras_edlee))}")
```

```
Palabras comunes entre Elon Musk y Ed Lee:      1760
Palabras comunes entre Elon Musk y Bill Gates:      1758
Palabras comunes entre Bill Gates y Ed Lee:      1717
```

0s completed at 8:05 PM

Weka

Dataset: vote.arff



Clasificador: ZeroR

```
=== Run information ===

Scheme:       weka.classifiers.rules.ZeroR
Relation:     vote
Instances:    435
Attributes:   17
              handicapped-infants
              water-project-cost-sharing
              adoption-of-the-budget-resolution
              physician-fee-freeze
              el-salvador-aid
              religious-groups-in-schools
              anti-satellite-test-ban
              aid-to-nicaraguan-contras
              mx-missile
              immigration
              synfuels-corporation-cutback
              education-spending
              superfund-right-to-sue
              crime
              duty-free-exports
              export-administration-act-south-africa
              Class
Test mode:    10-fold cross-validation
```

```
=== Classifier model (full training set) ===

ZeroR predicts class value: democrat

Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         267               61.3793 %
Incorrectly Classified Instances       168               38.6207 %
Kappa statistic                          0
Mean absolute error                      0.4742
Root mean squared error                  0.4869
Relative absolute error                100      %
Root relative squared error            100      %
Total Number of Instances              435

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    1.000    0.614      1.000    0.761      ?        0.491     0.609     democrat
                 0.000    0.000    ?          0.000    ?          ?        0.491     0.382     republican
Weighted Avg.    0.614    0.614    ?          0.614    ?          ?        0.491     0.521

=== Confusion Matrix ===

   a   b   <-- classified as
 267   0 |   a = democrat
 168   0 |   b = republican
```

Clasificador: PART

```
=== Run information ===

Scheme:       weka.classifiers.rules.PART -C 0.25 -M 2
Relation:     vote
Instances:    435
Attributes:   17
              handicapped-infants
              water-project-cost-sharing
              adoption-of-the-budget-resolution
              physician-fee-freeze
              el-salvador-aid
              religious-groups-in-schools
              anti-satellite-test-ban
              aid-to-nicaraguan-contras
              mx-missile
              immigration
              synfuels-corporation-cutback
              education-spending
              superfund-right-to-sue
              crime
              duty-free-exports
              export-administration-act-south-africa
              Class
Test mode:    10-fold cross-validation
```

```
=== Classifier model (full training set) ===

PART decision list
------------------

physician-fee-freeze = n: democrat (253.41/3.75)

synfuels-corporation-cutback = n AND
education-spending = y: republican (125.78/1.29)

immigration = y AND
adoption-of-the-budget-resolution = n: republican (14.32/0.25)

crime = y AND
anti-satellite-test-ban = y: republican (13.08/1.63)

crime = y AND
adoption-of-the-budget-resolution = y AND
superfund-right-to-sue = y: democrat (7.05/0.07)

crime = y: republican (19.07/4.94)

: democrat (2.29/0.03)

Number of Rules  :      7


Time taken to build model: 0.03 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         412               94.7126 %
Incorrectly Classified Instances        23                5.2874 %
Kappa statistic                          0.8879
Mean absolute error                      0.071
Root mean squared error                  0.2114
Relative absolute error                 14.9765 %
Root relative squared error             43.4245 %
Total Number of Instances              435
```

```
=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.966    0.083    0.949      0.966    0.957      0.888    0.949     0.924     democrat
                 0.917    0.034    0.945      0.917    0.931      0.888    0.949     0.937     republican
Weighted Avg.    0.947    0.064    0.947      0.947    0.947      0.888    0.949     0.929

=== Confusion Matrix ===

   a   b   <-- classified as
 258   9 |   a = democrat
  14 154 |   b = republican
```

Clasificador: DesitionTable

```
=== Run information ===

Scheme:       weka.classifiers.rules.DecisionTable -X 1 -S "weka.attributeSelection.BestFirst -D 1 -N 5"
Relation:     vote
Instances:    435
Attributes:   17
              handicapped-infants
              water-project-cost-sharing
              adoption-of-the-budget-resolution
              physician-fee-freeze
              el-salvador-aid
              religious-groups-in-schools
              anti-satellite-test-ban
              aid-to-nicaraguan-contras
              mx-missile
              immigration
              synfuels-corporation-cutback
              education-spending
              superfund-right-to-sue
              crime
              duty-free-exports
              export-administration-act-south-africa
              Class
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

Decision Table:

Number of training instances: 435
Number of Rules : 40
Non matches covered by Majority class.
        Best first.
        Start set: no attributes
        Search direction: forward
        Stale search after 5 node expansions
        Total number of subsets evaluated: 125
        Merit of best subset found:   96.322
Evaluation (for feature selection): CV (leave one out)
Feature set: 3,4,12,16,17

Time taken to build model: 0.17 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         413               94.9425 %
Incorrectly Classified Instances        22                5.0575 %
Kappa statistic                          0.8929
Mean absolute error                      0.0995
Root mean squared error                  0.208
Relative absolute error                 20.9752 %
Root relative squared error             42.7262 %
Total Number of Instances              435

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.966    0.077    0.952      0.966    0.959      0.893    0.981     0.990     democrat
                 0.923    0.034    0.945      0.923    0.934      0.893    0.981     0.958     republican
Weighted Avg.    0.949    0.061    0.949      0.949    0.949      0.893    0.981     0.977

=== Confusion Matrix ===

   a   b   <-- classified as
 258   9 |   a = democrat
  13 155 |   b = republican
```