P12

June 6, 2019

1 Reporte de práctica 12: Análisis de texto con nltk y wordcloud

En esta práctica trataremos con cadenas de caracteres. En la base de datos se cuenta con las sinopsis de los videos que se registraron en el concurso.

1.1 Objetivo

- Usar nltk o sklearn para hacer algún tipo de análisis de texto
- Apoyarse en las herramientas de bash para preprocesamiento.

1.2 Lectura de datos

Se aplican comandos del bash de sistema para poder saber cuales son las columnas de los archivos .csv y para poder obtener un ejemplo de los datos de cada archivo.

1.2.1 Columnas de los csv

```
In [8]: !head -n 1 2016.csv | tr , '\n' | grep -v "^$" | nl -v 2

     2        Sinopsis
     3        limpios


In [10]: !head -n 1 2017.csv | tr , '\n' | grep -v "^$" | nl -v 2

     2        Sinopsis
     3        limpios


In [11]: !head -n 1 2018.csv | tr , '\n' | grep -v "^$" | nl -v 2

     2        Categoria
     3        Edad
     4        Pais
     5        Titulo
     6        Genero
     7        Duracion
     8        Marca
     9        Referencia
```

```
    10        Dias
    11        Marcas
    12        Personas
    13        Sinopsis
    14        limpios
```

```
In [12]: !head -n 1 2018mx.csv | tr , '\n' | grep -v "^$" | nl -v 2
```

```
     2        Sinopsis
     3        limpios
```

Se consulta los primeros cinco registros para confirmar que la información que tenemos es la que se necesita.

1.2.2 Ejemplos

```
In [13]: !head -n 5 2016.csv
```

```
Sinopsis,limpios
"Valentina España, una niña preocupada por que sus amigas ya no juegan si no que se la pasan en
"sammy es una pequeña soñadora que espera con ansias un día especial en la playa para jugar y c
es una niña que tiene un sueño lo cual ella se preocupa por lo que soño asi que decide investig
"Es un documental en donde se investiga el punto de vista de una niña de cinco años ante el mun
```

```
In [14]: !head -n 5 2017.csv
```

```
Sinopsis,limpios
"ESTA HISTORIA INICIA CON CADA UNO DE LOS PERSONAJES CAMINANDO POR LA CALLE, ELLOS SE ENCUENTRA
"La historia tratarÍ de un convencional joven, rutinario, que a pesar de estar metido en su vid
"Se quiere resaltar BogotÍ y sus alrededores capturando paisajes y su cotidianidad desde los to
"A lo largo del día se muestra la rutina de Martín, un joven aparentemente comÍžn y corriente c
```

```
In [16]: !head -n 5 2018.csv
```

```
Categoria,Edad,Pais,Titulo,Genero,Duracion,Marca,Referencia,Dias,Marcas,Personas,Sinopsis,limpi
AFICIONADO,24,Colombia,0.5,Drama,300,Motorola,Moto G 2,7,130.0,4,"Jimena, mujer rígidamente est
CRONICAS,26,Belgium,12,Crónica,180,Apple,ad,1,,1,<ksdvnc,ksdvnc
AFICIONADO,34,Colombia,#ATuRitmo,Comedia,65,Huawei,P10 lite,8,245.0,7,"Un día cualquiera en el
AFICIONADO,41,Colombia,#EstamosConPipe,Ficción,300,Apple,Ipad Mini 2 Modelo: MD528E7/A,16,153.0
```

```
In [17]: !head -n 5 2018mx.csv
```

```
Sinopsis,limpios
"Se trata de una niña que pensaba que la última palabra la tenia el culo, pero comprueba que no
una pequeña probada de lo que tenemos planeado para una serie de nuestro pequeño set de produc
"Luzma es una maestra jubilada que recuerda con melancolía sus años de enseñanza, pero una tar
"TRATA SOBRE CUANDO ALGUIEN VIAJA DE NOCHE POR LAS CARRETERAS, SE CUENTAS DIVERSAS LEYENDAS URB
```

1.3 La librería nltk

El kit de herramientas de lenguaje natural, o más comúnmente NLTK, es un conjunto de bibliotecas y programas para el procesamiento del lenguaje natural (PLN) simbólico y estadísticos para el lenguaje de programación Python. NLTK incluye demostraciones gráficas y datos de muestra. Se importa la librería al programa.

```
In [1]: import nltk

        nltk.download('punkt')
        nltk.download('stopwords')
        nltk.download('vader_lexicon')
        from nltk.corpus import stopwords
        print(stopwords.words("spanish")[:10])
        from nltk.sentiment.vader import SentimentIntensityAnalyzer
        s = SentimentIntensityAnalyzer() # en inglés hasta podemos distinguir entre palabras p
        print(s.polarity_scores('useless'))
        print(s.polarity_scores('marvelous'))
```

```
[u'de', u'la', u'que', u'el', u'en', u'y', u'a', u'los', u'del', u'se']
{'neg': 1.0, 'neu': 0.0, 'pos': 0.0, 'compound': -0.4215}
{'neg': 0.0, 'neu': 0.0, 'pos': 1.0, 'compound': 0.5994}
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     /home/samataroukami/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]     /home/samataroukami/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package vader_lexicon to
[nltk_data]     /home/samataroukami/nltk_data...
[nltk_data]   Package vader_lexicon is already up-to-date!
```

Ahora se importa un documento para procesar en nltk, el 2018.xslx. Se cuenta con la librería necesaria para poder trabajar con archivos tipo Excel desde Python.

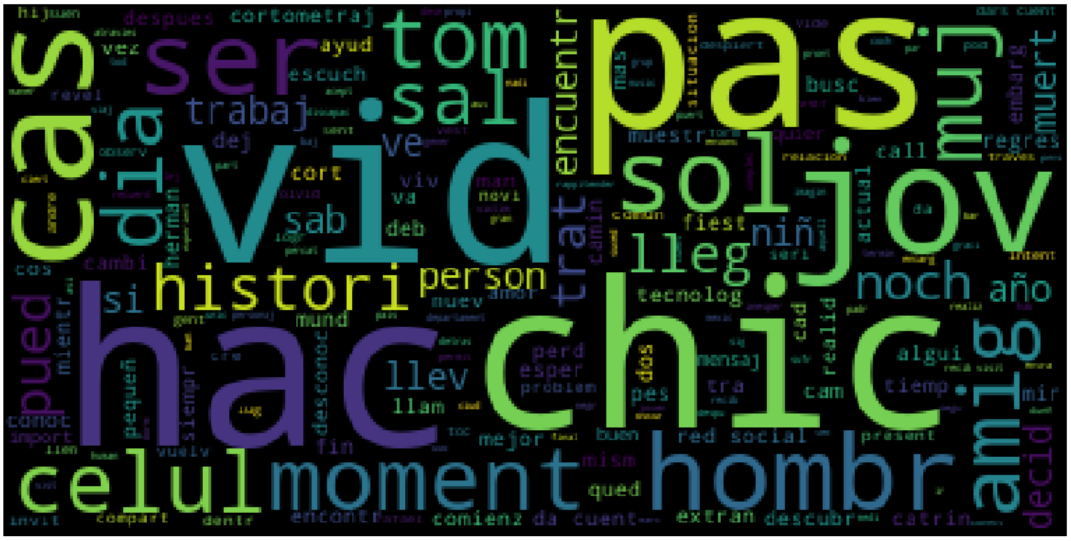
```
In [18]: import pandas as pd
         import matplotlib.pyplot as plt
         from wordcloud import WordCloud
         from nltk import word_tokenize
         from nltk.corpus import stopwords
         from nltk.stem import SnowballStemmer
         from nltk.tokenize import RegexpTokenizer

         d = pd.read_excel('https://raw.githubusercontent.com/SamatarouKami/CIENCIA_DE_DATOS/ma
         d = d[['Categoria','Edad','Pais', 'Titulo','Genero', 'Duracion', 'Marca','Referencia'
         n = len(d)
```

```
spa = stopwords.words("spanish")
stemmer = SnowballStemmer('spanish')
tokenizer = RegexpTokenizer(r'\w+') # para eliminar puntuación
reemplazos = []
for r in range(n):
    original = d.Sinopsis[r]
    reemplazo = ''
    if original != 'SIN_DESCR':
        quedar = [stemmer.stem(p) for p in tokenizer.tokenize(original) if p.lower() 
        reemplazo = ' '.join(quedar)
    reemplazos.append(reemplazo)
d['limpios'] = reemplazos
texto = ' '.join(reemplazos)
nube = WordCloud().generate(texto)
plt.rcParams["figure.figsize"] = [15, 7]
plt.imshow(nube)
plt.axis("off")
plt.show()
d.to_csv("2018.csv",  index=False, encoding="utf-8")
```

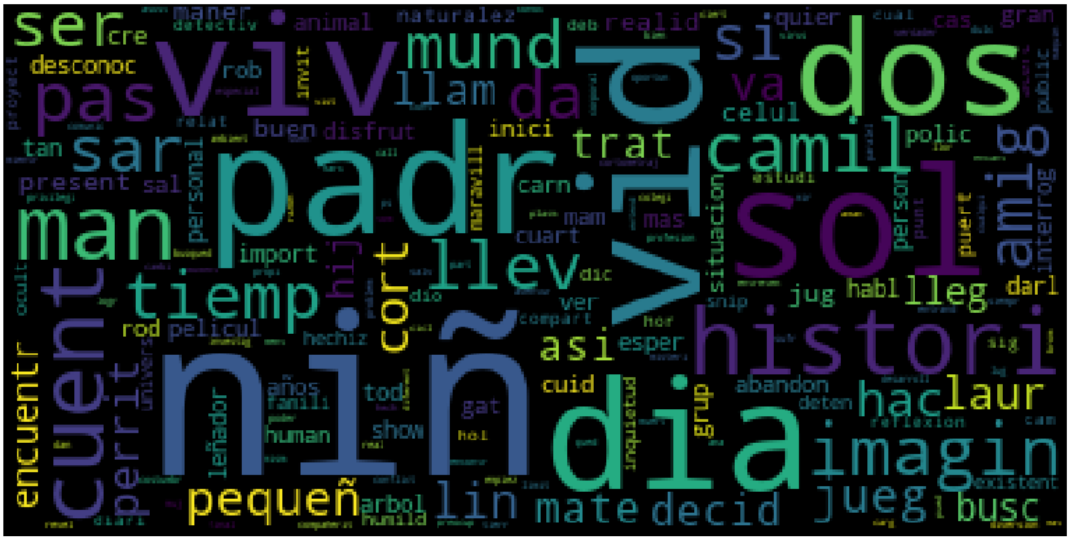
In [3]:
```
d = pd.read_excel('https://raw.githubusercontent.com/SamatarouKami/CIENCIA_DE_DATOS/mas
d = d[['Sinopsis']]
n = len(d)
spa = stopwords.words("spanish")
stemmer = SnowballStemmer('spanish')
tokenizer = RegexpTokenizer(r'\w+') # para eliminar puntuación
reemplazos = []
for r in range(n):
```

```
        original = d.Sinopsis[r]
        reemplazo = ''
        if original != 'SIN_DESCR':
            quedar = [stemmer.stem(p) for p in tokenizer.tokenize(original) if p.lower() no
            reemplazo = ' '.join(quedar)
        reemplazos.append(reemplazo)
    d['limpios'] = reemplazos
    texto = ' '.join(reemplazos)
    nube = WordCloud().generate(texto)
    plt.rcParams["figure.figsize"] = [15, 7]
    plt.imshow(nube)
    plt.axis("off")
    plt.show()
    d.to_csv("2018mx.csv",  index=False, encoding="utf-8")
```



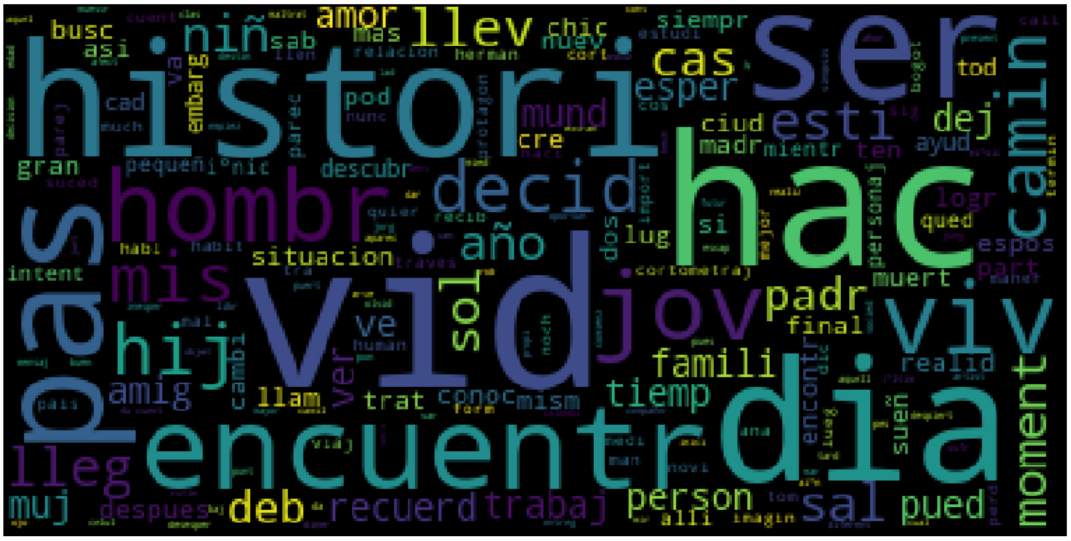
In [4]:
```
    d = pd.read_excel('https://raw.githubusercontent.com/SamatarouKami/CIENCIA_DE_DATOS/mas
    d = d[['Sinopsis']]
    n = len(d)
    spa = stopwords.words("spanish")
    stemmer = SnowballStemmer('spanish')
    tokenizer = RegexpTokenizer(r'\w+') # para eliminar puntuación
    reemplazos = []
    for r in range(n):
        original = d.Sinopsis[r]
        reemplazo = ''
        if original != 'SIN_DESCR':
            quedar = [stemmer.stem(p) for p in tokenizer.tokenize(original) if p.lower() no
            reemplazo = ' '.join(quedar)
```

```
        reemplazos.append(reemplazo)
    d['limpios'] = reemplazos
    texto = ' '.join(reemplazos)
    nube = WordCloud().generate(texto)
    plt.rcParams["figure.figsize"] = [15, 7]
    plt.imshow(nube)
    plt.axis("off")
    plt.show()
    d.to_csv("2016.csv",  index=False, encoding="utf-8")
```




```
In [5]: d = pd.read_excel('https://raw.githubusercontent.com/SamatarouKami/CIENCIA_DE_DATOS/mas
        d = d[['Sinopsis']]
        n = len(d)
        spa = stopwords.words("spanish")
        stemmer = SnowballStemmer('spanish')
        tokenizer = RegexpTokenizer(r'\w+') # para eliminar puntuación
        reemplazos = []
        for r in range(n):
            original = d.Sinopsis[r]
            reemplazo = ''
            if original != 'SIN_DESCR':
                quedar = [stemmer.stem(p) for p in tokenizer.tokenize(original) if p.lower() no
                reemplazo = ' '.join(quedar)
            reemplazos.append(reemplazo)
        d['limpios'] = reemplazos
        texto = ' '.join(reemplazos)
        nube = WordCloud().generate(texto)
        plt.rcParams["figure.figsize"] = [15, 7]
```

```
plt.imshow(nube)
plt.axis("off")
plt.show()
d.to_csv("2017.csv", index=False, encoding="utf-8")
```



1.4 Conclusión

Al usar la librería nltk se obtivieron imagenes con las palabras más utilizadas en las sinopsis de los videos. Las palabras que se ven presentes en todos gráficos son:

- vid
- dia
- viv
- histori
- ser

> La librería es muy util ya que se puede corregir los errores ortográficos desde codigo Python.
> --06 de junio 2019-- Luis Angel Gutiérrez Rodríguez 1484412