Explicación código

Nicolas Andres Vera

506211068

```
import numpy as np
import pandas as pd
from pandas_profiling import ProfileReport
#El codigo  importa las bibliotecas de NumPy y Pandas , también importa la biblioteca pandas_profiling y ProfileReport genera un informe de los datos utilizado por las bibliotecas
```

```
[ ] data = pd.read_csv("scopus.csv")
#El codigo carga el csv llamado scopus.csv
```

```
[ ] profile = ProfileReport(data, title="Profiling Report")
#genera el informe utilizando la libreria pandas y le da un titulo
```

```
[ ] profile
#Se hace el llamada para ejecutar lo anterior
```

Summarize dataset: 100% 43/43 [00:06<00:00, 4.73it/s, Completed]

Generate report structure: 100% 1/1 [00:08<00:00, 8.62s/it]

Render HTML: 100% 1/1 [00:01<00:00, 1.56s/it]

Profiling Report

Overview Variables Interactions Correlations Missing values Sample

Number of observations	650
Missing cells	3145
Missing cells (%)	20.2%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	122.0 KiB
Average record size in memory	192.2 B

Numeric	2
Unsupported	2

Variables

```
[ ] import numpy as np
    import pandas as pd

    import requests
    from bs4 import BeautifulSoup

    from PIL import Image
    from wordcloud import WordCloud, ImageColorGenerator
    import matplotlib.pyplot as plt

    import string
    import nltk
    from nltk.corpus import stopwords
    #Basicamente importamos varias librerias que son:
    #numpy y pandas para analizar los datos
    #requests y BeautifulSoup para hacer solicitudes http y analizar un html
    #PIL y WordCloud para crear una imagen de acuerdo a un texto
    #matplotlib.pyplot para hacer graficos en python
    #string y nltk procesar texto
    #Todo esto se utilizara para generar la nube de palabras que veremos despues
```

```
[ ]  nltk.download('stopwords')
     #Se quitan palabras vacias o sin significado
```

```
    [nltk_data] Downloading package stopwords to /root/nltk_data...
    [nltk_data]   Unzipping corpora/stopwords.zip.
    True
```

```
[ ] punctuation=[]
    for s in string.punctuation:
        punctuation.append(str(s))
    sp_punctuation = ["¿", "¡", "“", "”", "…", ":", "—", "»", "«"]

    punctuation += sp_punctuation
    #Se crea una lista de puntuaciones que pueden ser utiles para eliminarlos del texto porque no son relevantes
```

```
[ ] titulos = str
    for titulo in data["Abstract"]:
        titulos = titulo + " " + str(titulos)
    titulos
    #Se concatenan los valores de abstract
```

```
'Objective: To establish a predictive model of injury risk based on anatomical and functional descriptors with knee and ankle joint instability in non-professional basketball players. Design and participants: Non-experimenta
l, descriptive cross-sectional study, directed sampling; 44 university subjects of both sexes, 27 men (61.4%) and 17 women (38.6%) were evaluated; non-professional basketball players aged between 18 - 26 years (22 ± 3.2), be
longing to the sports branch of university basketball in Chile. Results: a) (p = .000) Valgus knee stress establishes a significant association with postural angle in the knee and ankle and functional variables of take off;
level of discrimination based model 82% (CC = .820). b) (p = .000) Valgus ankle stress establishes a significant association with the knee and ankle angle, functional variables of take off; level of discrimination based on m
odel 71.3% (CC = .713). c) (p = .000) General risk establishes a significant association with the knee an…'
```

```
[ ] stop_words = stopwords.words('english')
    #Este código carga la lista de palabras vacías en inglés para enfocarse en las palabras importantes y significativas
```

```
[ ] for p in punctuation:
        clean_texto = titulos.lower().replace(p,"")

    for p in punctuation:
        clean_texto = clean_texto.replace(p,"")
    #Esta compuestos de dos ciclos for, el primero convierte a minusculas y el segundo limpia el texto
```

```
[ ] for stop in stop_words:
        clean_texto_list = clean_texto.split()
        clean_texto_list = [i.strip() for i in clean_texto_list]
        try:
            while stop in clean_texto_list: clean_texto_list.remove(stop)
        except:
            print("Error")
            pass
        clean_texto= " ".join(clean_texto_list)
        #Elimina palabras vacias como los conectores para que no se vean reflejados en el resultado
```

```
[ ] lista_texto = clean_texto.split(" ")

    palabras = []

    for palabra in lista_texto:
        if (len(palabra)>=3 and len(palabra)<18):
            palabras.append(palabra)
            #Se separa el texto y se crea una lista llamada palabras, y el bucle verifica si la longitud de la palabra es mayor o igual a 3 y menor que 18 , si cumple lo agrega a la lista
```

```
[ ] word_count={}
    for palabra in palabras:
        if palabra in word_count.keys():
            word_count[palabra][0]+=1
        else:
            word_count[palabra]=[1]
     #Se crea un diccionario wordl_count , con el bucle se verifica si la palabra esta en word_count, si esta suma y si no se agrega al diccionario
```

```
[ ]
```

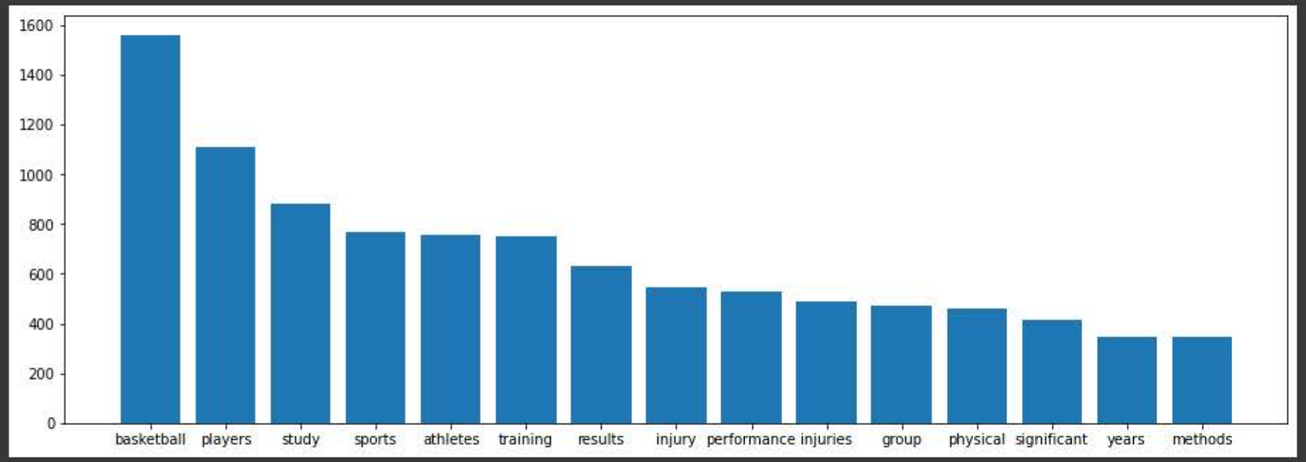
```
[ ] df = pd.DataFrame.from_dict(word_count).transpose()
    df.columns=["freq"]
    df.sort_values(["freq"], ascending=False, inplace=True)
    df.head(10)
    #, el código devuelve las primeras 10 filas del DataFrame panda ordenado, que corresponden a las palabras con mayor frecuencia
```

	freq
basketball	1562
players	1112
study	881
sports	768
athletes	754
training	748
results	631
injury	546
performance	530
injuries	486

```
[ ]
```

```
[ ]  def plot_bar(data=df, top=5):
         fig = plt.figure()
         ax = fig.add_axes([0,0,2,1])
         ax.bar(x =df.iloc[:top,:].index, height = df.iloc[:top,0].values)
         plt.show()
         # Se crea una figura de Matplotlib y se agrega el eje de las barras, ademas se muestra el grafico
```

```
[ ]  plot_bar(data=df, top=15)
     #Llama a la funcion anterior y se hace un grafico de los 15 resultados que mas se repiten
```




```
[ ]  word_cloud = WordCloud(height=800, width=800, background_color='white',max_words=150, min_font_size=5, collocation_threshold=10).generate(clean_texto)
     word_cloud.to_file("n1.png")
     plt.figure(figsize=(10,8))
     plt.imshow(word_cloud)
     plt.axis('off')
     plt.tight_layout(pad=0)
     plt.show()
     #Se genera la nube de palabras teniendo en cuenta la biblioteca wordcloud , se dan parametros como la altura, se crea la imagen y se le dice donde guardarse , este es el resultado cuando ya limpiamos las palabras
     #anteriormente
```


```
data.head()
# muestra las primeras cinco filas del conjunto de datos data
```

	Authors	Author(s) ID	Title	Year	Source title	Volume	Issue	Art. No.	Page start	Page end	...	Affiliations	Authors with affiliations	Abstract	Author Keywords	Index Keywords	Document Type	Publication Stage
0	da Silva G.C.R., Tebar W.R., Saraiva B.T.C., F...	57256954400;57148200400;56979972600;3776124220...	Association of Early Sports Practice with Card...	2023	Sports Medicine - Open	9.0	1	15	NaN	NaN	...	Department of Physical Education, School of Te...	da Silva, G.C.R., Department of Physical Educa...	Background: Sports practice in childhood and a...	Cardiometabolic; Health; Motor behavior; Physi...	accelerometry; adult; Article; athletics; bask...	Article	Final
	Iwatsu J., Yoho Y.		Knee pain in young		BMC Sports							Department of	Iwatsu, J.,	Background: Knee is the	Knee pain;	adolescent;		