Práctica 1 (25% nota final)

1. Contexto. Explicar en qué contexto se ha recolectado la información. Explique por qué el sitio web elegido proporciona dicha información.

En este proyecto, se ha decidido investigar sobre la plataforma de videos en vivo Twitch. Esta plataforma, trabaja con la participación de usuarios (streamers) que realizan transmisiones de video en tiempo real (stream), principalmente sobre videojuegos. Por otra parte, el resto de los usuarios que no realizan streams sino que consumen el material o, en otras palabras, visualizan los videos, tienen la posibilidad de realizar donaciones económicas a los streamers y, además, deben ver publicidades cada cierto tiempo. Por consiguiente, los streamers reciben remuneraciones por su trabajo, las cuales son proporcionales al número de usuarios que visitan e interactúan con su canal.

Por esta razón, muchas personas han decidido trabajar con esta plataforma. Sin embargo, a pesar de la gran cantidad de público potencial, resulta difícil captar la atención de los usuarios a partir de un nuevo canal. Por lo tanto, resulta conveniente analizar las tendencias de las preferencias actuales de los usuarios para la elección de sus visitas para, así, desarrollar contenido en los nuevos canales a partir de estos resultados. En este caso, la característica más importante a determinar es: ¿Qué juegos son los más demandados por los usuarios?.

En este trabajo, se ha elegido un sitio llamado https://sullygnome.com, el cual es un sitio web que proporciona estadísticas y análisis de los canales más influyentes dentro de la plataforma Twitch. Este sitio fue elegido ya que clasifica a los top streamers y proporciona información relevante sobre sus canales. Esta información resulta clave para la realización del scraping.

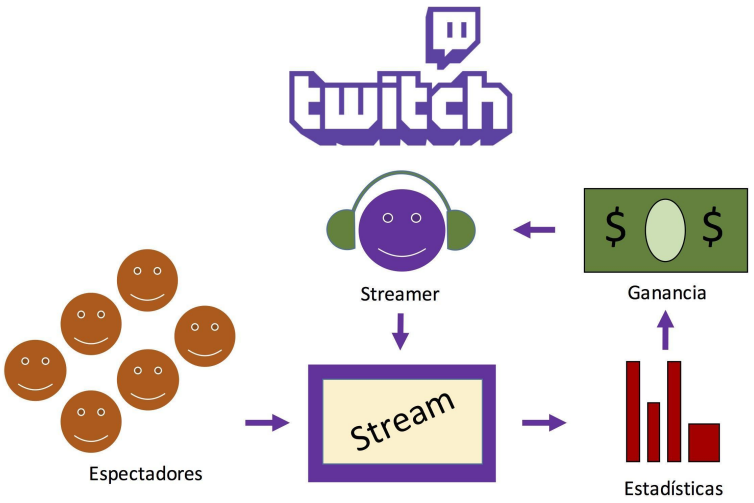
2. Definir un título para el dataset. Elegir un título que sea descriptivo.

Canales más exitosos de transmisión de videojuegos en la plataforma Twitch: los datos más relevantes.

3. Descripción del dataset. Desarrollar una descripción breve del conjunto de datos que se ha extraído (es necesario que esta descripción tenga sentido con el título elegido).

Tal como expresa el título, el dataset está basado sobre los datos más importantes a tomar en cuenta para la evaluación de un cierto canal de Twitch. En este dataset, se presentan dichos datos para cada uno de los top streamers dentro de la plataforma en un periodo de 365 días. Las unidades o magnitudes de las características extraídas son en horas y cantidad según el caso. Los datos no han pasado por un proceso de preprocesado o limpieza, por lo que aún pueden existir inconsistencias y el formato no es necesariamente el más adecuado para un análisis directo. Por ejemplo, hay campos donde los valores enteros se presentan como un string del tipo “XXX horas”, en vez de ser simplemente el valor XXX. En este caso, se extrajo la información para el top 50 de los canales de la plataforma. La descripción de las características extraídas son descritas en las siguientes preguntas. El formato del dataset es un fichero CSV que facilita su visualización y tratamiento.

4. Representación gráfica. Presentar esquema o diagrama que identifique el dataset visualmente y el proyecto elegido



5. Contenido. Explicar los campos que incluye el dataset, el periodo de tiempo de los datos y cómo se ha recogido.

En este dataset, se presentan dichos datos para cada uno de los top streamers dentro de la plataforma en un periodo de 365 días. Las características extraídas son las siguientes:

A. Usuario: Nombre o link del usuario (streamer) analizado.

B. Followers: Número de usuarios que siguen a dicho streamer.
C. Streams: Cantidad de videos realizados por el streamer.
D. AverageViews: Media de visualizaciones anuales al canal.
E. WatchTime: Tiempo acumulado de visualizaciones anuales en horas.
F. PeakViewers: Cantidad máxima de espectadores registrada en el último año .
G. StreamTime: Tiempo acumulado de transmisión en vivo en horas (en el último año).
H. topGameLink: Link con el nombre e información del juego más utilizado por dicho streamer para realizar sus transmisiones.

Los datos fueron recogidos a través de web scraping en lenguaje Python sobre las páginas individuales de cada streamer dentro del sitio web. Para ello, primero se extrajo la información sobre los canales que estuvieran en el top de visitas. Luego, de forma automática, se recorre cada una de las páginas individuales donde se encuentra la información relevante. Sobre cada página, se aplica el scraping para recolectar la información. Finalmente, se guardan los datos extraídos en un fichero CSV.

6. Agradecimientos. Presentar al propietario del conjunto de datos. Es necesario incluir citas de análisis anteriores o, en caso de no haberlas, justificar esta búsqueda con análisis similares.

El propietario del sitio web y del conjunto de datos es David (aka SullyGnome), quien facilita las estadísticas y análisis para cada uno de los canales investigados. Él se basa en la API de Twitch para recoger la información, la cual sondea cada 15 minutos y es presentada en el sitio (el cual se actualiza diariamente). El propietario puede ser contactado a través de su mail (contact@sullygnome.com) o su Twitter (@SullyGnome).

Debido a que el propietario expresa en su página web la petición de que no se le realice web scraping a su sitio, se ha procedido a su contacto para pedir su consentimiento. Previo a la realización del scraping, se contactó al propietario para obtener su permiso para realizar el proceso sobre su sitio web. El propietario accedió con la condición de no programar un bot recurrente sobre su sitio sino que se tratase de una sola experiencia.

Existen diversos análisis similares fácilmente encontrados en internet. Ejemplo de estos son:

- https://www.catalystmints.com/gaming-news/124-top-games-to-stream-on-twitch-and-youtube-keep-audience-engaged
- https://twitchstrike.com/what
- https://www.tomsguide.com/us/pictures-story/1491-best-games-to-stream-on-twitch.html

Sin embargo, se trata de análisis que explican simplemente los resultados y no su procedimiento. Además, no permiten trabajar sobre ellos para realizar otros análisis diferentes.

7. Inspiración. Explique por qué es interesante este conjunto de datos y qué preguntas se pretenden responder. Es necesario comparar con los análisis anteriores presentados en el apartado 6.

El interés de analizar este conjunto de datos es que, gracias al crecimiento exponencial de consumidores de multimedia en internet, se han abierto nuevas puertas de autoempleo a través de estas plataformas. Sin embargo, como fue mencionado anteriormente, resulta bastante complicado iniciar con estas plataformas y poder recibir algún beneficio. Por esta razón, es importante analizar las tendencias de los usuarios para, de esta manera, tomar mejores decisiones del contenido inicial a presentar que permita un crecimiento más eficiente dentro de la plataforma, en este caso: Twitch.

Concretamente, las preguntas que se pretenden responder son: ¿Quiénes son los streamers más influyentes?, ¿Cuántos espectadores atraen?, ¿Cuanto pueden ganar aproximadamente?, ¿Qué juegos son los más buscados por los usuarios?

Tal y como se mencionó anteriormente, los análisis anteriores no permiten trabajar sobre los datos obtenidos por lo que no tienen un valor añadido equiparable al de este proyecto. Por otra parte, una de las principales limitaciones de este Dataset es que los datos pueden variar con mucha frecuencia por lo que el tiempo de vida es relativamente corto al tratarse de una lista que depende tanto de moda como de tendencias dentro de la red social. Sin embargo, el código permite actualizar este Dataset cada vez que sea ejecutado siempre y cuando la estructura del sitio web no cambie.

8. Licencia. Seleccione una de estas licencias para su dataset y explique el motivo de su selección:
    - Released Under CC0: Public Domain License

- Released Under CC BY-NC-SA 4.0 License
- Released Under CC BY-SA 4.0 License
- Database released under Open Database License, individual contents under Database Contents License
- Other (specified above)
- Unknown License

Una posible licencia para este conjunto de datos puede ser CC BY-SA 4.0 License. La elección se basa en la idoneidad de las cláusulas que en ella se presentan en relación con el trabajo realizado en donde:

- Se provee el nombre del creador del conjunto de datos generado y se indican los cambios realizados sobre este. De esta manera, se reconoce el trabajo de terceros y en qué medida se realizaron aportaciones con respecto al trabajo original.
- Se permite su uso comercial, lo cual incrementa las posibilidades de que empresas puedan interesarse en los datos generados, permitiendo así, la realización de nuevos proyectos que reporten un reconocimiento al autor original.
- Las nuevas contribuciones deben ser publicadas bajo la misma licencia, lo que permite que se le reconozca al autor original en todo momento y bajo los mismo términos que fueron planteados por él.

9. Código. Adjuntar el código con el que se ha generado el dataset, preferiblemente en Python o, alternativamente, en R.

El código fue hecho en lenguaje Python con la implementación de la librería Selenium. El código fuente se encuentra dentro de la carpeta Git.

10. Dataset. Publicación del dataset en formato CSV en Zenodo (obtención del DOI) con una breve descripción.

El DOI al dataset en formato CSV en Zenodo.

Contribuciones	Firma
Investigación previa	XX
Redacción de las respuestas	XX
Desarrollo código	XX

Código

```
from selenium import webdriver
import requests
import os

#First we want to check the robots.txt file of our objective
website
#A function get_robot_txt is constructed to check any url
def get_robot_txt(url):
    if url.endswith('/'):
        path = url
    else:
        path = url + '/'

    req = requests.get(path + "robots.txt",data =None)
    return req.text

#Objective website
URL = "https://sullygnome.com"

#Read robots.txt file
print(get_robot_txt(URL))

#Setting options for the webdriver
option = webdriver.ChromeOptions()
option.add_argument("--incognito") #open incognito mode
option.add_argument("user-agent=AcademicCrawler")#set our UserAgent
name, in this case AcademicCrawler

#Getting current folder path
My_path = os.path.dirname(os.path.abspath(__file__))

#Delay/Pause of download Throttling
```

```
TimeOut = 5 #sec

#Looking for the chromedriver file (Download from
http://chromedriver.chromium.org/downloads)
browser = webdriver.Chrome(executable_path=My_path +
'/chromedriver.exe', chrome_options=option)

#Get content from objective website
browser.get("https://sullygnome.com/channels/365/mostfollowers")

#Apply delay
browser.implicitly_wait(TimeOut);

#Check if our UseraAgent is OK
agent = browser.execute_script("return navigator.userAgent")
print(agent)

#Search all the urls for the top 50 streamers
elements =
browser.find_elements_by_xpath("""//*[@id="tblControl"]/tbody/tr/td
[1]/a""")
links = []
for element in elements:
    links.append(element.get_attribute("href"))

#Navigate through the links
dictlist = []
for link in links:
    browser.get(link)
    browser.implicitly_wait(TimeOut);
    userStatsDict={}

    #Data for Scraping
        followers = browser.find_elements_by_css_selector("body >
div.RightContent > div.MainContent > div.InfoStatPanelContainerTop
```

```
> div > div.InfoStatPanelWrapper.InfoStatPanelSpacerLeft > div > div > div.InfoStatPanelTL > div")
    streams = browser.find_elements_by_css_selector("#combinedPanel > div > div.InfoPanelCombinedBottom > a")
     averageViews = browser.find_elements_by_css_selector("body > div.RightContent > div.MainContent > div.InfoStatPanelContainerTop > div > div.InfoStatPanelWrapper.InfoStatPanelSpacerLeft > div > div > div.InfoStatPanelTL > div")
      watchTime = browser.find_elements_by_css_selector("body > div.RightContent > div.MainContent > div.InfoStatPanelContainerTop > div > div:nth-child(2) > div > div > div.InfoStatPanelTL > div")
     peakViewers = browser.find_elements_by_css_selector("body > div.RightContent > div.MainContent > div.InfoStatPanelContainerTop > div > div:nth-child(5) > div > div > div.InfoStatPanelTL > div")
      streamTime = browser.find_elements_by_css_selector("body > div.RightContent > div.MainContent > div.InfoStatPanelContainerTop > div > div.InfoStatPanelWrapper.InfoStatPanelSpacerMiddle > div > div > div.InfoStatPanelTL > div")
                                                  topGame = browser.find_elements_by_xpath("""//*[@id="combinedPanel"]/div/div[2]/div/div[2]/div[8]/div[1]/a""")

    #Saving variables into a dictionary
    userStatsDict['User']=link
    userStatsDict['Followers']=followers[0].text
    userStatsDict['Streams']=streams[0].text
    userStatsDict['AverageViews']=averageViews[0].text
    userStatsDict['WatchTime']=watchTime[0].text
    userStatsDict['PeakViewers']=peakViewers[0].text
    userStatsDict['StreamTime']=streamTime[0].text
    userStatsDict['topGameLink']=topGame[0].get_attribute("href")

    dictlist.append(userStatsDict)

# Overwrite to the specified file.
```

```
# Create it if it does not exist.
filename = "/TwitchData.csv"
file = open(My_path + filename, "w+")

#Get the keys of the dictionary
keys = []
for key in userStatsDict:
    keys.append(key)

# Dump all the data with CSV format
for i in range(len(keys)):
    file.write(str(keys[i]) + ";");
file.write("\n");

for i in range(len(dictlist)):
    for j in range(len(keys)):
        file.write(str(dictlist[i][keys[j]]) + ";");
    file.write("\n");

file.close()

browser.quit()
```