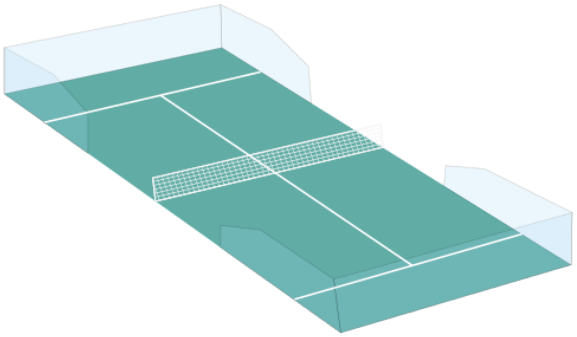
Tipología y Ciclo de vida de los datos

Práctica 1 - Web Scraping

Web Scraping en World Padel Tour



José Ramón Martínez-Carbonell Martín
Miguel Ángel Pérez García

Apartados

1. Contexto - ¿Por qué se ha elegido ese tema/sitio web?

El pádel es un deporte en auge en los últimos años. Aunque inicialmente solo estaba en Argentina, en la década de los 2000 llegó a España, creando el primer circuito profesional de calado. Actualmente se está expandiendo a otros países como Suecia, Italia y Portugal, creando un circuito cada vez más importante y conocido a nivel mundial.

Junto a todo ello, existen multitud de datasets para muchos deportes como tenis, baloncesto o fútbol, pero a día de hoy hay muy pocos para pádel. Creemos por tanto que puede resultar de interés para interesados de este deportes. Y con ello no nos referimos únicamente a aficionados, sino también a los posibles patrocinadores que puedan estar interesados en apostar por diferentes jugadores en base a sus estadísticas o futura progresión, interrogantes que intentaremos responder con este dataset.

Finalmente, otro de los motivos que nos ha hecho escogerlo ha sido que hemos notado un cierto reto técnico, en cuanto que teníamos que aplicar cierta navegabilidad entre URL's para obtener la información.

2. Definir un título para el dataset

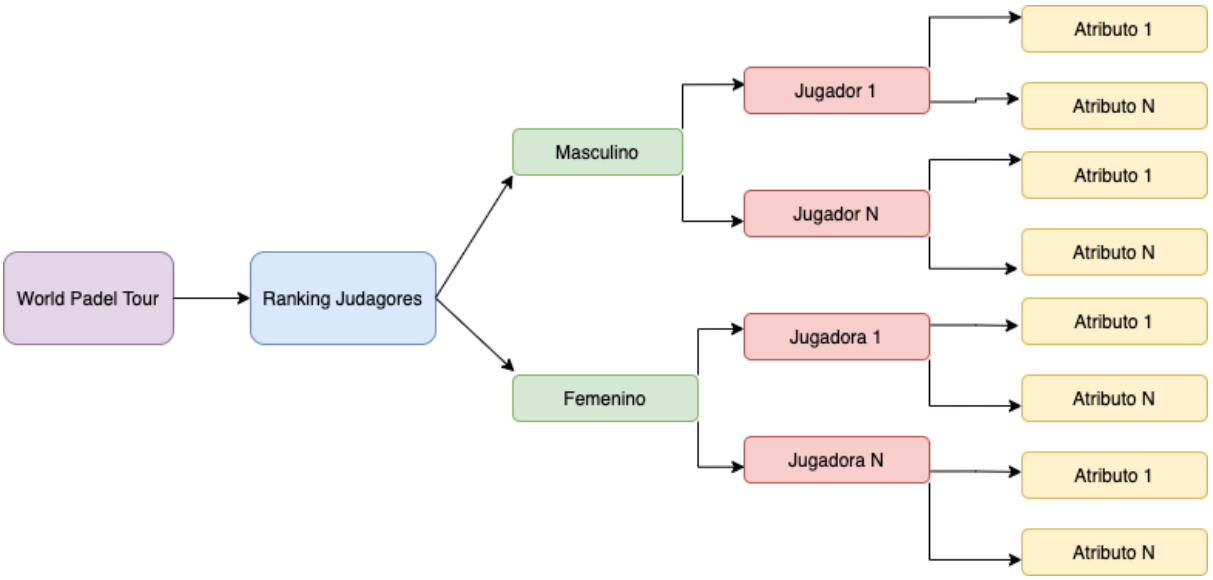
El título elegido para este dataset es: "*Estadísticas de los jugadores del circuito masculino y femenino de World Padel Tour en 2021 y 2020*".

3. Descripción del dataset

Contiene distintos datos relativos a cada uno de los jugadores y jugadoras del World Padel Tour[1]. En concreto, nos muestra información detallada acerca de cada jugador, como su nombre, posición, edad y nacionalidad. Junto a todo ello nos informa de las victorias y derrotas del jugador, así como de los resultados que ha cosechado en los dos últimos años.

4. Representación gráfica

Seguidamente mostraremos un diagrama que ilustra el esquema de conocimiento que hemos encontrado en la web. Cabe destacar que nombramos tanto jugadores como atributos de 1 a N, para indicar que hay varios y están englobados dentro de ese subconjunto. En los apartados posteriores se detalla cada uno de esos atributos.



5. Contenido

El periodo de tiempo de los datos va desde los primeros partidos y torneos en el año 2020 hasta el actual 2021, temporada que acaba de comenzar. Hemos recogido los datos de su página web[1] con Python haciendo uso de las bibliotecas de BeautifulSoup y Selenium, a continuación detallaremos cada uno de los atributos que contiene el conjunto de datos:

Atributo	Descripción	Clasificación	Tipo de datos
Nombre	Nombre completo del jugador o jugadora.	Categórico	Cadena de caracteres
Posición	Hace referencia al lugar que ocupa en la pista. Este puede ser *drive* si juega en la derecha y *revés* si juega en la parte izquierda.	Categórico	Cadena de caracteres
Ranking	Posición que ocupa en el ranking de jugadores.	Discreto	Entero
Compañero	Nombre del jugador con el que comparte pista y es compañero.	Categórico	Cadena de caracteres
Altura	Altura del jugador	Continuo	Decimal
Fecha de nacimiento	Fecha de nacimiento del jugador	Continuo	Fecha
Residencia	Lugar de residencia del jugador actualmente	Categórico	Cadena de caracteres

Lugar de nacimiento	Lugar en el que nace el jugador	Categórico	Cadena de caracteres
Partidos jugados total	Número de partidos totales que ha jugado en su carrera.	Discreto	Entero
Partidos ganados total	Número de partidos ganados en su carrera.	Discreto	Entero
Partidos perdidos total	Número de partidos perdidos en su carrera.	Discreto	Entero
Efectividad total	% de partidos ganados respecto a los partidos disputados.	Continuo	Decimal
Racha de victorias	Número total de victorias seguidas sin conocer la derrota.	Discreto	Entero
Partidos jugados añoX	Número de partidos jugados en el año X	Discreto	Entero
Partidos ganados añoX	Número de partidos ganados en el año X	Discreto	Entero
Efectividad año X	Relación entre los partidos ganados y perdidos en el año X	Continuo	Decimal
campeon año X	Veces que ha sido campeón en el año X	Discreto	Entero
finalista año X	Veces que ha sido finalista en el año X	Discreto	Entero
semifinalista año X	Veces que ha sido semifinalista en el año X	Discreto	Entero
cuartos año X	Veces que se ha quedado en cuartos en el año X	Discreto	Entero
octavos año X	Veces que se ha quedado en octavos en el año X	Discreto	Entero
dieciseisavos año X	Veces que se ha quedado en dieciseisavos en el año X	Discreto	Entero

6. Agradecimientos

World Padel Tour[1] o WPT, como su nombre describe, es el campeonato de Pádel con mayor potencial del mundo. Reúne a los mejores profesionales del Pádel Nacional Español e Internacional. La empresa que organiza todo lo relativo a la competición, y por tanto a la que agradecemos el conjunto de datos anterior para fines académicos (Ir al apartado Contenidos), es Setpoint Events, S.A. con CIF A-66270844 domiciliada en Barcelona. Su socio principal es S.A. DAMM.

No se han encontrado estudios previos de relevancia en este campo, por lo que no citaremos ninguno. De hecho, ese ha sido uno de los puntos por los que nos hemos inclinado en recoger este tipo de datos.

7. Inspiración

Dada la carencia de conjuntos de datos cuyo principal objetivo sea almacenar información interesante de los torneos y estadísticas de Pádel, como se menciona en el apartado Contexto, lo tomamos como objetivo actual y abordaremos durante esta práctica la implementación de un agente para capturar los datos y construir un conjunto con los datos de los torneos y los jugadores de Pádel, la estructura del dataset está definida en el apartado Contenido.

Algunas de las preguntas que nuestro dataset puede responder:

- **¿Cuál es el jugador que mejor puntuación ha sacado en toda la clasificación?** Este dato en concreto se sacará teniendo en cuenta el momento de extracción de los datos, dado que va cambiando conforme avanza el campeonato y para la entrega de la práctica, la temporada 2021 ya habrá empezado.

- **¿Es la altura un factor determinante para cosechar éxitos?** En los últimos años se está extendiendo un cambio de juego en el que cada vez los jugadores más “físicos” parece que cosechan mejores resultados. Sin embargo, si observamos el dataset se pueden sacar conclusiones reales.

- **Cuando un jugador es joven, ¿llega más lejos en los torneos?** La vida útil de un jugador de pádel puede extenderse hasta los 35 años (aunque hay jugadores con 40). La madurez es un grado, y en este deporte se muestra de manera clara. Intentaremos responder esa pregunta en base a los datos que tenemos.

- **¿A qué edad un jugador consigue mejores resultados?** Esta pregunta está relacionada con la anterior. Sin embargo, se diferencia de ella en que estamos intentando obtener la edad óptima.

8. Licencia

La licencia que elegimos para nuestro dataset es Creative Commons BY-NC-ND[3]. Se utiliza la licencia más restrictiva porque según la web de WPT “Los Derechos de Propiedad Intelectual e Industrial sobre toda la información contenida en sitios web de WPT, entre otros, diseño gráfico, dibujos, logotipos, imágenes, índices, textos, audios, videos, software, diseño y signos identificativos, así como sus códigos fuente, estructura de navegación, bases de datos, y todos los elementos en ellos contenidos, son titularidad de WPT excepto aquellos contenidos que pertenecen a terceros al que el Usuario puede acceder a través de hipervínculos.” En nuestro caso, vamos a utilizar los recursos de la web con fines académicos.

9. Código

Nuestro repositorio, el cual contiene el código fuente y el CSV con los datos, se puede encontrar en https://github.com/miguel-a-ngel/WebScraper. Igualmente se adjunta en este documento.

```
import requests
from bs4 import BeautifulSoup
from selenium import webdriver
import time
import csv
import re
import unidecode

#Driver de selenium

driver = webdriver.Firefox(executable_path = '..\geckodriver.exe')

#Pagina web de World Padel Tour en la que nos aparece un listado con todos los jugadores, tanto
#del ranking masculino como del ranking femenino

link_players = 'https://www.worldpadeltour.com/jugadores/'
index = 'https://www.worldpadeltour.com'

#Definimos cabecera distinta a la por defecto para establecer nuestro user agent.
headers = {
"Accept": "text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,\
*/*;q=0.8",
"Accept-Encoding": "gzip, deflate, sdch, br",
"Accept-Language": "en-US,en;q=0.8",
"Cache-Control": "no-cache",
"dnt": "1",
"Pragma": "no-cache",
"Upgrade-Insecure-Requests": "1",
```

```
"User-Agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_3) AppleWebKit/5\
37.36 (KHTML, like Gecko) Chrome/56.0.2924.87 Safari/537.36"
}

#Lista de atributos de jugadores, divididos en 3 arrays distintos por simplicidad, dado que cada uno
#de los arrays aparece en un punto determinado de la página
player_attributes_one = ["name","ranking", "puntos",
"partidos_jugados","partidos_ganados","partidos_perdidos","efectividad","racha_victorias"]
player_attributes_two = ["compañero", "posicion", "lugar nacimiento", "fecha nacimiento", "altura",
"residencia"]
statistics_attributes = ["partidos jugados 2021", "partidos ganados 2021", "efectividad 2021",
"campeon 2021", "finalista 2021", "semifinalista 2021", "cuartos 2021", "octavos 2021",
"dieciseisavos 2021", "partidos jugados 2020", "partidos ganados 2020", "efectividad 2020",
"campeon 2020", "finalista 2020", "semifinalista 2020", "cuartos 2020", "octavos 2020",
"dieciseisavos 2020"]
years = ["2021", "2020"]


#---------------------------------- Funciones de formateo URL --------------------------------

#Función que elimina los acentos de un cadena de caracteres, así como de convertir algunas
#cadenas en otras que nos interesen
def remove_accents(raw_text):
    raw_text = re.sub(u"[àáâãäå]", 'a', raw_text)
    raw_text = re.sub(u"[èéêë]", 'e', raw_text)
    raw_text = re.sub(u"[ìíîï]", 'i', raw_text)
    raw_text = re.sub(u"[òóôõö]", 'o', raw_text)
    raw_text = re.sub(u"[ùúûü]", 'u', raw_text)
    raw_text = re.sub(u"[ýÿ]", 'y', raw_text)
    raw_text = re.sub(u"[ª]", 'maria', raw_text)
    raw_text = re.sub(u"[ñ]", 'n', raw_text)
    return raw_text

#Método que separa la cadena de NombreApellido1Apellido2 en Nombre, Apellido 1 y Apellido 2
# y que convierte a letra minuscula para la posterior construcción de la URL
def camel_case_split(name):
        splitted = re.sub('([A-Z][a-z]+)', r' \1',re.sub('([A-Z]+)', r' \1', name)).split()
        converted_list = [x.lower() for x in splitted]
        return converted_list

#Función que compone la URL específica del recurso jugador con '-' para concordar con el formato
de URL que utiliza
#el sitio web
def compose_url(array_name):
        new_url = '-'.join(array_name)
        return remove_accents(new_url)

#Función principal que construye la url final de cada jugador, añadiendo el prefijo de World Padel
Tour
#para todos los jugadores
def build_url(name):
        compound_url = link_players + compose_url(camel_case_split(name)) + '/'
```

```
        return compound_url

#-------------------------------------------------------------------------------------------
#----------------------------------Extracción de imágenes-----------------------------------
def load_requests(source_url):
    r = requests.get(source_url, stream=True, headers = headers)
    if r.status_code == 200:
        aSplit = source_url.split('/')
        ruta = "/Users/JRamon/imgs_wpt/" + aSplit[len(aSplit)-1]
        output = open(ruta,"wb")
        output.write(r.content)
        output.close()
    else:
        print("Error processing web")

def get_img(content_player):
    img_url_div = content_player.find_all('div', class_='u-img-cropped')
    img_url = img_url_div[1].get('style')
    formatted_img_url = img_url.replace('background-image: url('').replace(');'')
    load_requests(formatted_img_url)

#-------------------------------------------------------------------------------------------

#Método que itera sobre la página de cada jugador para extraer sus atributos.
# Tiene 3 bucles en base a la estructura de la página, para iterar sobre los distintos
# componentes que interesan
def get_attributes(url_player):
    web_player = requests.get(url_player, headers = headers)
    player_list_one = []
    player_list_two = []
    player_list_statistics = []
    if(web_player.status_code == 200) :
        content_player = BeautifulSoup(web_player.content, "lxml")
        #get_img(content_player) <-- Descomentar para guardar las img
        #Nombre del jugador
        player_list_one.append(content_player.find('h1', class_='c-ranking-header__title').text)
        print(url_player)
        i = 1
        for data in content_player.find_all('div', class_='c-ranking-header__data-box'):
            new_data = data.find('p', class_='c-ranking-header__data').text
            print(player_attributes_one[i] + " : "  + new_data)
            player_list_one.append(new_data)
            i+=1
        j = 0
        for more_data in content_player.find_all('li', class_='c-player__data-item'):
            item = more_data.find('p').text
            print(player_attributes_two[j] + " : "  + item)
            player_list_two.append(item)
            j+=1
        statistics_attributes_index = 0
        year_index = 0
        count_statistics = 0
```

```
        for statistics in content_player.find_all('span', class_='c-flex-table__item-data'):
            if(count_statistics == 9): #Si hemos llegado a las 9 estadísticas
                year_index += 1
                statistics_attributes_index = 0
                count_statistics = 0
            if( year_index == 2):
                break


            print(years[year_index] + " " + statistics_attributes[statistics_attributes_index] + " : " +
statistics.text)
            player_list_statistics.append(statistics.text)
            statistics_attributes_index+=1
            count_statistics+=1
    else:
            print('Error processing webpage : ', url_player)
    return player_list_one + player_list_two + player_list_statistics

#Procedimiento que persiste a un jugador en un fichero CSV
def persist(player):
            with open('statistics_players.csv', 'a', newline='') as csvfile:
                        storer = csv.writer(csvfile, delimiter=',', quotechar='"',
quoting=csv.QUOTE_MINIMAL, dialect='excel')
                        storer.writerow(player)

#Procedimiento que prepara todo lo necesario para el procesamiento de un jugador:
# Llama a las funciones que construyen la url.
# A la que obtiene los atributos que queremos almacenar en el fichero csv.
# A la que almacena la fila del jugador en el csv.
def process_player(url_player):
            #url_player = build_url(url_player)
            print("Procesando: ", url_player)
            player = get_attributes(url_player)
            print(player)
            if player != []:
                        persist(player)

#Procedimiento principal que llama a todas las funciones definidas.
# Realiza un scroll down con Selenium en la página de resumen
# donde se muestran los jugadores, para principalmente obtener todos los jugadores de Padel.
def scroll_down(driver, link):
            driver.get(link)
            driver.execute_script("window.scrollTo(0, document.body.scrollHeight);")
            print("Cargando datos.")
            time.sleep(20)

            web = driver.page_source
            content = BeautifulSoup(web, "lxml")
            count = 1 #Player counter

            for player in content.find_all('li', class_='c-player-card__item'):
                        name = player.find('div', class_='c-player-card__name').text
```

```
                url = player.find('a', class_='c-trigger')
                process_player(url['href'])
                time.sleep(10)
                count += 1

        print("Contador de jugadores ", count)
        driver.close()

################################  main  ################################
persist(player_attributes_one + player_attributes_two + statistics_attributes)#añadimos cabecera al csv
scroll_down(driver, link_players)
```

10. Publicación en Zenodo

El enlace a Zenodo es: https://zenodo.org/record/4682067#.YHSvexMzYWo con el DOI: 10.5281/zenodo.4682067

11. Decisiones de diseño

- El código se ha creado en un único fichero, dado que por el carácter y tamaño de la práctica no hemos visto necesario modularizar..

- En un primer momento, se optó por obtener la URL del jugador a partir de su nombre, por lo que hay varios métodos que cumplen esta función. Sin embargo, la construcción de dicha URL por parte de los desarrolladores de World Padel Tour no sigue una convención de manera estricta, con lo que nos quedamos sin obtener satisfactoriamente la URL de varios jugadores. Por este motivo optamos por acceder a dicha URL tomando la dirección del atributo href del componente que nos interesaba.

- La extracción de archivo multimedia se ha realizado tomando la URL del recurso, la cual estaba “escondida” en el estilo del div. Se deja comentado en el código la llamada al método encargado de realizar dicha acción. Si el usuario quisiera utilizarlo debería configurar la ruta en la que quiere que se almacenen los archivos que se generan.

- A pesar de los intentos para no obtener un 403, en estos últimos días han cambiado el servidor, limitando las peticiones, a pesar de separarlas en el tiempo con timeouts y probando distintas IP.

Contribuciones

Contribuciones	Firma
Análisis e investigación	José R. y Miguel A.
Documentación	José R. y Miguel A.
Desarrollo	José R. y Miguel A.
Pruebas	José R. y Miguel A.
Referencias	José R. y Miguel A.

Referencias

1. World Padel Tour. (2020). *World Padel Tour*. Recuperado de https://www.worldpadeltour.com/

2. Icon-Icons. (2020). *Court padel sport Icon*. Recuperado de https://icon-icons.com/icon/court-padel-sport/141849

3. Creative Commons. (2020). *Sobre las licencias*. Recuperado de https://creativecommons.org/licenses/

4. StackOverflow. (2020). *StackOverflow*. Recuperado de https://es.stackoverflow.com/

5. Python Doc. (2020). *The Python Standard Library*. Recuperado de https://docs.python.org/3/library/