

Tipología y ciclo de vida de los datos

Práctica 1: Web scraping

Web scraping de estadísticas de jugadores NBA

Alumnos: Alberto Sánchez Mazarro
Sergio Romero Córdoba

Índice

| | | |
|-----|---|----|
| 1. | Contexto | 3 |
| 2. | Título | 4 |
| 3. | Descripción del dataset | 5 |
| 3.1 | Estadísticas de jugadores | 5 |
| 3.2 | Información de jugadores | 5 |
| 4. | Representación gráfica | 6 |
| 4.1 | Estadísticas de jugadores | 6 |
| 4.2 | Información de jugadores | 7 |
| 5. | Contenido | 8 |
| 5.1 | Estadísticas de jugadores | 8 |
| 5.2 | Información de jugadores por posición | 9 |
| 6. | Agradecimientos | 11 |
| 7. | Inspiración | 12 |
| 8. | Licencia | 13 |
| 9. | Código | 14 |
| 10. | Dataset | 15 |
| 11. | Participación | 16 |

1. Contexto.

La NBA (National Basketball Association) es la liga profesional de Baloncesto estadounidense. Está considerada la mejor liga del mundo, tanto a nivel deportivo como de marketing y organización. Desde su sitio web es posible consultar gran cantidad de información sobre los jugadores y equipos que la componen.

En este trabajo vamos a realizar web scraping sobre ella para obtener información de las estadísticas de los jugadores de un equipo en los principales aspectos del juego (puntos, rebotes, asistencias, recuperaciones, tapones y porcentaje de tiros de campo) en una temporada. Esta información se encuentra dentro de la sección de estadísticas de la página web de cada uno de los equipos de la NBA y es la que utilizaremos para crear nuestro dataset que posteriormente se guardará en formato csv. Además, la página muestra una foto de cada jugador que también será descargada, almacenándose en la carpeta "Pictures" dentro de la carpeta donde se ejecute el script.

Adicionalmente, la página web global, nba.com, cuenta con una sección sobre los jugadores en la que se puede encontrar información de los mismos, tales como el equipo en el que juegan, la posición, la altura o el peso. Esta sección también se recorre en este trabajo, haciendo web scraping de la misma descargando la información de los jugadores agrupados según su posición (guard, forward o center).

2. Título

A fin de intentar abarcar el mayor número de prácticas de web scraping posibles, en este trabajo se obtendrán los siguientes datasets:

- Un dataset que se denominará “Estadísticas de jugadores del equipo *Nombre_equipo* Temporada *Año_temporada*”, donde “*Nombre_equipo*” es el nombre del equipo, que se introduce por parámetro y “*Año_temporada*” la temporada de la que se desean obtener las estadísticas.

Este dataset será el obtenido al ejecutar el fichero main.py

- Tres datasets (Bases.csv, Aleros.csv y Pivots.csv)

Estos datasets se obtienen al ejecutar el archivo “practicaSelenium.py” y corresponden a la descarga de todos los jugadores de la NBA por posición. Como observación, indicar que un jugador puede ocupar diferentes posiciones, por lo que podría aparecer en varios datasets.

3. Descripción del dataset

Como hemos comentado, este trabajo consta de dos datasets diferenciados: el primero con información de estadísticas de jugadores y el segundo con información de los propios jugadores.

A continuación pasamos a describir cada uno de estos datasets.

3.1 Estadísticas de jugadores

Las opciones para descargar este dataset permiten elegir equipo específico de la NBA o bien elegir descargar la información de todos los equipos para una temporada determinada.

En ambos casos, las columnas que contiene son las siguientes:

- **Jugador:** Nombre del jugador
- **Partidos:** Número de partidos disputados
- **Puntos:** Total de puntos anotados
- **FG%:** Porcentaje de acierto en tiros de campo
- **3PT%:** Porcentaje de acierto en tiros de 3 puntos
- **FT%:** Porcentaje de acierto en tiros libres
- **OffReb:** Total de rebotes en ataque capturados
- **DefReb:** Total de rebotes en defensa capturados
- **Rebotes:** Total de rebotes capturados
- **Asistencias:** Total de pases de canasta efectuados
- **Recuperaciones:** Total de recuperaciones logradas
- **Pérdidas:** Total de pérdidas de balón cometidas
- **Faltas:** Total de faltas cometidas

Adicionalmente, la imagen de los jugadores se descarga en una carpeta (carpeta "Pictures" dentro de la carpeta en la que se ejecute) con el nombre del jugador.

3.2 Información de jugadores por posición

Este dataset contendrá la información de los jugadores en una posición determinada.

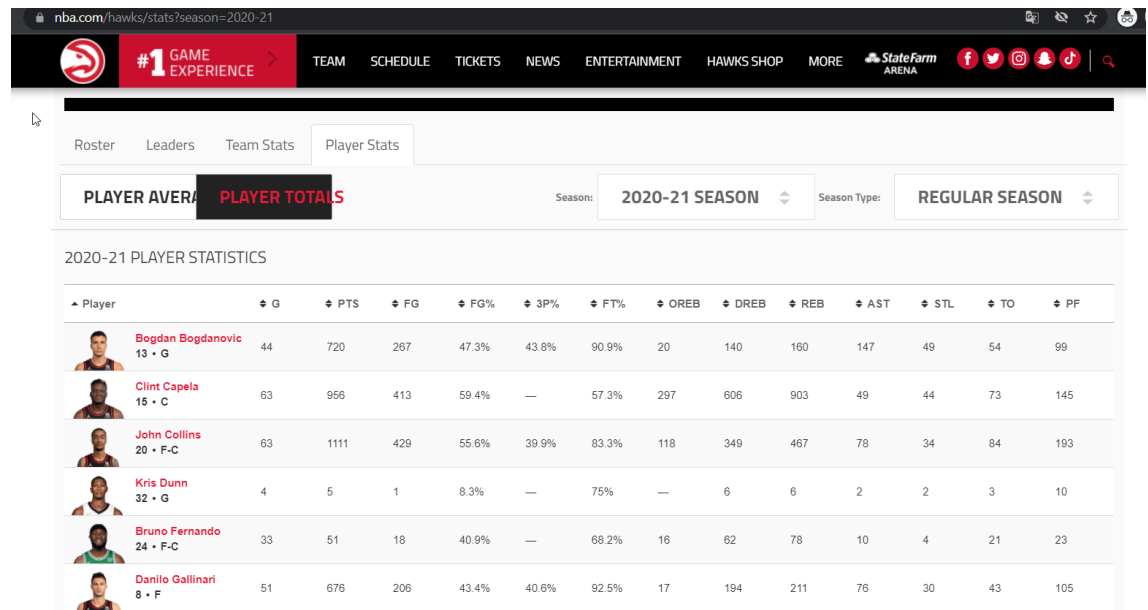
Las columnas que contiene son las siguientes:

- **Jugador:** Nombre del jugador
- **Equipo:** Equipo al que pertenece el jugador
- **Posición:** Posición en la que juega

4. Representación gráfica.

4.1 Estadísticas de jugadores

A continuación ofrecemos una captura de pantalla de la página web de la que hemos obtenido la información:



The screenshot shows the NBA Hawks' 2020-21 Player Statistics page. The page is titled "2020-21 PLAYER STATISTICS" and features a table of player statistics. The table includes columns for Player, G (Games), PTS (Points), FG (Field Goals), FG% (Field Goal Percentage), 3P% (Three-Point Percentage), FT% (Free Throw Percentage), OREB (Offensive Rebounds), DREB (Defensive Rebounds), REB (Total Rebounds), AST (Assists), STL (Steals), TO (Turnovers), and PF (Personal Fouls). The table lists six players: Bogdan Bogdanovic, Clint Capela, John Collins, Kris Dunn, Bruno Fernando, and Danilo Gallinari. The page also includes a navigation bar with links to Roster, Leaders, Team Stats, and Player Stats, and a search bar.

| Player | G | PTS | FG | FG% | 3P% | FT% | OREB | DREB | REB | AST | STL | TO | PF |
|-----------------------------|----|------|-----|-------|-------|-------|------|------|-----|-----|-----|----|-----|
| Bogdan Bogdanovic 13 • G | 44 | 720 | 267 | 47.3% | 43.8% | 90.9% | 20 | 140 | 160 | 147 | 49 | 54 | 99 |
| Clint Capela 16 • C | 63 | 956 | 413 | 59.4% | — | 57.3% | 297 | 606 | 903 | 49 | 44 | 73 | 145 |
| John Collins 20 • F-C | 63 | 1111 | 429 | 55.6% | 39.9% | 83.3% | 118 | 349 | 467 | 78 | 34 | 84 | 193 |
| Kris Dunn 32 • G | 4 | 5 | 1 | 8.3% | — | 75% | — | 6 | 6 | 2 | 2 | 3 | 10 |
| Bruno Fernando 24 • F-C | 33 | 51 | 18 | 40.9% | — | 68.2% | 16 | 62 | 78 | 10 | 4 | 21 | 23 |
| Danilo Gallinari 8 • F | 51 | 676 | 206 | 43.4% | 40.6% | 92.5% | 17 | 194 | 211 | 76 | 30 | 43 | 105 |

Esta página corresponde a un equipo de la NBA. Todos los equipos cuentan con una página idéntica en la que podemos ver las estadísticas de los jugadores de ese equipo para una temporada específica.

Adicionalmente, para el manejo de contenido audiovisual, hemos descargado la foto del jugador que acompaña a sus estadísticas.

El dataset obtenido se descarga en un fichero CSV con el siguiente formato:

| Jugador | Partidos | Puntos | FG% | 3PT% | FT% | OffReb | DefReb | Rebotes | Asistencias | Recuperacio | Perdidas | Faltas |
|-------------------|----------|--------|--------|--------|--------|--------|--------|---------|-------------|-------------|----------|--------|
| Enes Kanter | 2 | 4 | 40% | - | - | 2 | 2 | 4 | - | - | - | 1 |
| Dennis Schröder | 9 | 120 | 36.90% | 32.50% | 83.80% | 10 | 25 | 35 | 54 | 13 | 19 | 22 |
| Marcus Smart | 8 | 70 | 30.60% | 28% | 75% | - | 29 | 29 | 30 | 19 | 17 | 20 |
| Robert Williams | 8 | 74 | 69.60% | - | 66.70% | 29 | 38 | 67 | 11 | 9 | 8 | 13 |
| Payton Pritchett | 8 | 20 | 29.20% | 33.30% | - | 5 | 13 | 18 | 12 | - | 5 | 11 |
| Al Horford | 7 | 94 | 43.60% | 30.30% | 88.90% | 12 | 59 | 71 | 23 | 7 | 9 | 17 |
| Jayson Tatum | 9 | 204 | 37.30% | 27.10% | 74.50% | 9 | 64 | 73 | 31 | 8 | 21 | 27 |
| Bruno Fernando | 3 | - | - | - | - | - | - | - | 1 | - | 2 | 2 |
| Jaylen Brown | 8 | 205 | 49.30% | 39.70% | 78% | 3 | 46 | 49 | 20 | 10 | 23 | 25 |
| Juancho Hernández | 5 | 6 | 33.30% | 33.30% | 50% | 2 | 3 | 5 | - | 2 | 3 | 1 |
| Grant Williams | 9 | 63 | 51.30% | 44% | 85.70% | 6 | 16 | 22 | 11 | 2 | 5 | 16 |
| Aaron Nesmith | 6 | 16 | 31.60% | 26.70% | - | 3 | 4 | 7 | 4 | 1 | 1 | 6 |
| Jabari Parker | 4 | 24 | 62.50% | 66.70% | - | - | 7 | 7 | 1 | 1 | 2 | 5 |
| Romeo Langford | 6 | 37 | 50% | 46.70% | 100% | 6 | 8 | 14 | 4 | 2 | 5 | 13 |
| Josh Richardson | 7 | 47 | 42.90% | 33.30% | 80% | 6 | 12 | 18 | 7 | 4 | 6 | 13 |

No todos los jugadores cuentan con estadísticas en todos los apartados. En estos casos, la celda mostrará un guion.

4.2 Información de jugadores por posición

A continuación se muestra una captura de pantalla de la página web que tiene la información de los jugadores:

[NBA 75](#)
[Games](#)
[Schedule](#)
[News](#)
[Watch](#)
[Stats](#)
[Standings](#)
[Teams](#)
[Players](#)
[Fantasy](#)
[NBABet](#)
[NBA TV](#)
[League Pass](#)
[Store](#)
[Tickets](#)

[Sign In](#)

[Players](#)
[Home](#)
[Player Stats](#)
[Starting Lineups](#)
[Free Agent Tracker](#)
[Transactions](#)

All Players

All Players

All Teams

All Positions

All Colleges

All Countries

Show Historic

504 Rows • Page 1 of 11

| PLAYER | TEAM | NUMBER | POSITION | HEIGHT | WEIGHT | LAST ATTENDED | COUNTRY |
|--|---------------------|--------|----------|--------|---------|-----------------|-------------|
| Precious Achiuwa | TOR | 5 | F | 6-8 | 225 lbs | Memphis | Nigeria |
| Steven Adams | MEM | 4 | C | 6-11 | 265 lbs | Pittsburgh | New Zealand |
| Bam Adebayo | MIA | 13 | C-F | 6-9 | 255 lbs | Kentucky | USA |
| Santi Aldama | MEM | 7 | F-C | 6-11 | 215 lbs | Loyola-Maryland | Spain |
| LaMarcus Aldridge | BKN | 21 | C-F | 6-11 | 250 lbs | Texas-Austin | USA |
| Nickell Alexander-Walker | NOP | 6 | G | 6-5 | 205 lbs | Virginia Tech | Canada |
| Grayson Allen | MIL | 7 | G | 6-4 | 198 lbs | Duke | USA |
| Jarrett Allen | CLE | 31 | C | 6-10 | 243 lbs | Texas-Austin | USA |

La información se descarga agrupada por posición de los jugadores, por lo que se obtienen tres datasets que se descargan cada uno en un fichero CSV con el siguiente formato:

| PLAYER | TEAM | POSITION |
|---------------|------|----------|
| Nickeil Alex | NOP | G |
| Grayson Alle | MIL | G |
| Jose Alvarad | NOP | G |
| Kyle Anders | MEM | F-G |
| Cole Anthon | ORL | G |
| D.J. Augustir | HOU | G |
| Joel Ayayi | WAS | G |
| LaMelo Ball | CHA | G |
| Lonzo Ball | CHI | G |
| Desmond Ba | MEM | G |
| RJ Barrett | NYK | F-G |
| Will Barton | DEN | G |
| Nicolas Batu | LAC | G-F |
| Kent Bazem | LAL | G-F |
| Bradley Beal | WAS | G |
| Malik Beasle | MIN | G |
| DeAndre' Be | BKN | G-F |
| Patrick Beve | MIN | G |
| Eric Bledsoe | LAC | G |

5. Contenido.

5.1 Estadísticas de jugadores

Como hemos comentado en apartados anteriores, este dataset permite almacenar la información estadística sobre el rendimiento de los jugadores NBA en una temporada.

Para ello, nuestro script de Python comienza pidiendo el nombre de un equipo y la temporada de la que se quieren obtener los datos.

En la NBA los equipos son conocidos por el nombre de la ciudad a la que representan más un “apodo” por el que son conocidos. Este “apodo” es el que habrá que introducir como parámetro cuando se solicite el nombre del equipo.

Así, por ejemplo, si queremos obtener las estadísticas de Boston Celtics deberemos introducir “Celtics”. Y si queremos obtener las de los jugadores de Los Ángeles Lakers deberemos introducir “Lakers”.

El script también permite introducir el valor “all” para el nombre del equipo en cuyo caso recorrerá las páginas de todos los equipos para la descarga de las estadísticas.

Para la temporada, el formato requerido es “20xx-xx+1”; es decir, si queremos obtener los resultados de la temporada 2020-2021 introduciremos “2020-21”.

Desde el punto de vista técnico, hemos implementado dos ficheros:

- **scraper.py.**

Este fichero implementa la clase `NBAStatsScraper` que va a ser la encargada de realizar el web scraping propiamente dicho.

Esta clase tiene dos parámetros para su inicialización: el nombre del equipo y la temporada para los cuales queremos descargar las estadísticas. Además, cuenta con un atributo adicional, `teamStats`, un array en el que vamos a ir guardando las estadísticas de un jugador determinado. Cada elemento de este array corresponde por tanto a estadísticas de un jugador.

El método `scraper` es el que realiza el scraping de la página. Para ello hace uso de las librerías `requests` y `BeautifulSoup`. La tabla que contiene toda la información que queremos descargar la buscamos por el nombre de la clase, `season-totals`. Una vez que tenemos esta table, recorreremos las filas buscando en cada una de ellas la celda correspondiente a cada una de las estadísticas. Mostramos a continuación un ejemplo de la estructura de esta tabla correspondiente a un equipo (hemos suprimido el código html que no es relevante, en el sentido de que no se ha utilizado para el scraping, para facilitar la legibilidad):


```

<table class="stats-table" >
  <tr>
    <td class="player_name">
      <div>
        
        <span class="playerInfo">
          <span class="playerName">
            <a href="/hawks/roster/lou-williams/101150" >Lou Williams</a>
          </span>
        </span>
      </div>
    </td>
    <td class="gp">6</td>
    <td class="pts">32</td>
    <td class="fgm">11</td>
    <td class="fg_pct">31.4%</td>
    <td class="fg3_pct">20%</td>
    <td class="ft_pct">100%</td>
    <td class="oreb">1</td>
    <td class="dreb">9</td>
    <td class="reb">10</td>
    <td class="ast">8</td>
    <td class="stl">4</td>
    <td class="tov">2</td>
    <td class="pf">4</td>
  </tr>
</table>

```

Adicionalmente, descargamos la foto de cada jugador en la carpeta “pictures”.

- **main.py.**

Es el fichero principal que pide los datos al usuario para generar el dataset y descargar las imágenes. Este fichero importa la clase NBAStatsScraper implementada en el fichero scraper.py.

5.2 Información de jugadores por posición

Para la implementación de este subapartado hemos utilizado la librería de Selenium a fin de simular la ejecución de un navegador web.

Ejecutando el programa **practicaSelenium.py** se abre un navegador web (Chrome), que simula la ejecución de un usuario en el navegador realizando los siguientes pasos:

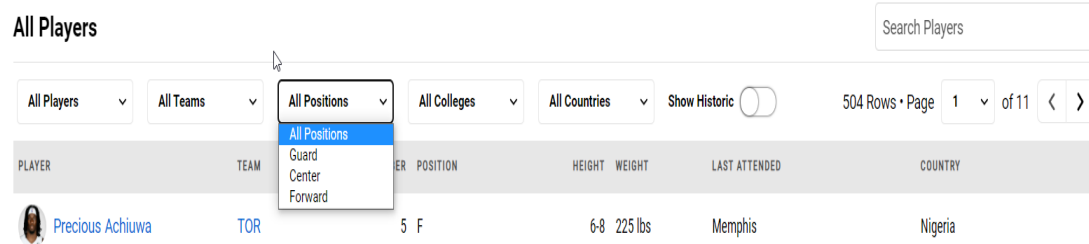
1. Abrir la página web <https://www.nba.com/players>
2. Aceptar las cookies en el popup que se levanta al realizar la ejecución (si accedemos directamente por la url en el navegador no salta, pero en la ejecución del script sí)
3. Filtrar los jugadores por una posición (inicialmente “bases”)
4. Cambiar la paginación para mostrar todos los jugadores que cumplen los requisitos (por defecto solo salen 50 y es preciso mostrar la opción de paginación “ALL”)
5. Recorrer el listado html de los jugadores y extraer su nombre, posición y equipo (se podrían obtener más datos pero hemos pensado que no aportan demasiado al objetivo de la práctica).

6. Exportar los datos anteriores a un dataset
7. Filtrar los jugadores por otra posición (**IMPORTANTE:** hemos detectado que al cambiar el filtro de “posición”, pese a que el combo de paginación sigue mostrando “ALL”, en realidad no se muestran todos los jugadores de dicha posición, sino solo la primera página. Es por ello que nos hemos visto obligados a forzar manualmente el cambio en la paginación cada vez, mostrando momentáneamente solo la primera página para luego volver a seleccionar la opción “ALL” para mostrar todos los jugadores y recorrerlos
8. Volvemos a exportar los nuevos datos al dataset de la posición seleccionada igual que en el punto 6.

Desde el punto de vista del código, hemos descargado el fichero chromedriver.exe para poder realizar la navegación con Selenium. Asimismo, se incluyen las librerías de Selenium para poder manejar los objetos obtenidos mediante la navegación.

A partir de ahí, hemos creado un fichero **practicaSelenium.py** en el que hemos desarrollado dos funciones:

- def muestraPosicion(combo, posicion). Esta función muestra todos los jugadores que pueden jugar en una determinada posición. El parámetro “combo” hace referencia al desplegable de “Position” de la página web y posición a la posición que queremos seleccionar.



- def data2csv(filename, rows). Esta función crea un csv con el nombre indicado en el parámetro “filename” a partir del parámetro “rows”, que son las filas con la información de los jugadores seleccionados.

En el código adjunto a la entrega se incluyen comentarios para comprender mejor el funcionamiento del script. Se notará que en algunos puntos se han incluido sentencias “sleep” para asegurar que se dispone del siguiente tiempo para ejecutar las acciones antes de realizar un cambio en la navegación.

6. Agradecimientos

La NBA, además de ser deportivamente la mejor liga de baloncesto del mundo, está considerada como una de las mejores marcas deportivas a nivel de publicidad y difusión.

En su página web es posible encontrar una cantidad ingente de información sobre sus jugadores y equipos (tanto actuales como anteriores).

Aprovechando esto, nos ha parecido interesante realizar la práctica de web scraping sobre las estadísticas acumuladas de los jugadores de una plantilla en una temporada. Hemos revisado el archivo “robots.txt” (<https://www.nba.com/robots.txt>) y aunque se incluyen algunas directivas “disallow”, no afectan a las estadísticas por equipo de los jugadores.

Adjuntamos el contenido del fichero donde se puede comprobar que no hay ningún impedimento para acceder al contenido a partir de /equipo/stats



Buscando trabajos similares realizados anteriormente, hemos encontrado análisis de estadísticas de baloncesto desde la web <https://www.basketball-reference.com/>, que almacena información no solo de la NBA sino de otras ligas de baloncesto.

En nuestro caso, hemos utilizado información pública de la web de la NBA para realizar nuestro dataset.

7. Inspiración

En la actualidad, en el mundo del baloncesto se le da una tremenda importancia al análisis del rendimiento de los jugadores.

Desde hace unos años se está utilizando en todos los equipos el concepto de “estadística avanzada”, que analiza a muy bajo nivel de detalle el rendimiento de los jugadores.

Así, los equipos disponen de información referente a desde qué punto de la pista es más efectivo un jugador en sus lanzamientos o si determinado jugador suele fallar más tiros por exceso o defecto de fuerza (lo cual da lugar a que el rebote del balón en el aro pueda salir en una u otra dirección).

Nuestro web scraping no resulta tan ambicioso, pero sí que permite considerar una cantidad de datos que pueden dar lugar a un análisis más exhaustivo del que permite un simple vistazo a las estadísticas generales.

De este modo, la extracción de estos datos nos podría permitir crear columnas “adicionales” en nuestro dataset.

Pongamos que nos interesa ver el ratio de puntos anotados por minuto de juego disputado, ya que no tiene la misma dificultad anotar 10 puntos disputando 40 minutos que disputando 20. Al disponer de la información en un dataset resulta trivial añadir una nueva columna para calcular este cociente

La parte de utilización de Selenium tiene un componente más didáctico que práctico, si bien puede resultar interesante tener almacenados todos los jugadores de la liga en función de la posición que ocupan en el campo.

8. Licencia

El tipo de licencia de nuestro dataset sería “Released Under CC BY-NC-SA 4.0 License”.

Obtenemos los datos de la NBA. Según las condiciones de uso de la web de la NBA se especifica que las estadísticas pueden ser utilizadas para uso no comercial.

9. Código

El código generado para la realización de este ejercicio se puede encontrar en https://github.com/AlberSM83/NBA_UOC_Sergio_Alberto

10. Dataset

Se han publicado los cuatro datasets de Zenodo en la siguiente dirección:

<https://zenodo.org/record/5650757#.YYZiXmDMJPY>

11. Participación

La práctica ha sido realizada conjuntamente. Si bien se ha realizado una lógica repartición de las tareas, ambos integrantes hemos participado en todos los puntos, tanto investigación, consensuar los datos a sacar, desarrollo de código, publicación y redacción de este documento.

| CONTRIBUCIONES | FIRMA |
|-----------------------------|---------------------------------|
| Investigación previa | Sergio Romero y Alberto Sánchez |
| Desarrollo de código | Sergio Romero y Alberto Sánchez |
| Redacción de las respuestas | Sergio Romero y Alberto Sánchez |
| | |