Análisis de estadísticas de NBA

Máster data analytics semipresencial

Bryan Eduardo De Sousa Velásquez

Juan García Bericua Goyanes

Nuria Heras Louzao

Victor Maestro Calvo

Alberto Sánchez Delgado

2020

INDICE

# INTRODUCCIÓN

# ANTECEDENTES

# HIPOTESIS

# METODOLOGIA DE TRABAJO Y VARIABLES

### EXTRACCIÓN DE DATOS

### VARIABLES NUEVAS

VARIABLE VALORACIÓN EQUIPO

# MODELO 1

# MODELO 2

# CONCLUSIONES RESPUESTAS A LAS HIPOTESIS

# REFERENCIAS BIBLIOGRÁFICAS / WEBGRAFÍA

*“La evaluación del jugador de baloncesto en los partidos es un proceso que despierta mucha atención de los entrenadores e investigadores” (Sampaio, 2002).*

# INTRODUCCIÓN

En un mundo en el que hasta ahora hemos basado nuestras decisiones empresariales y personales en experiencias y estadísticas de pequeños conjuntos de información, hemos pasado a un mundo en el que disponemos de gran cantidad de información.

La era actual de tecnologías de la información y comunicación ha permitido el registro de grandes cantidades de datos para posteriormente manipular la información y sacar conclusiones sobre la conducta de los seres humanos, ventajas para empresas y con aplicaciones para todos los sectores de la sociedad.

El análisis de grandes cantidades de datos para convertirlos en información relevante resulta una tarea compleja, pero que se hay ido aplicando a todos los aspectos de la vida, siendo el deporte uno de los que más ventajas se han obtenido, especialmente de deportes de táctica y estadísticas como es el baloncesto.

La NBA está en constante evolución. El estilo de juego, las capacidades de los jugadores, la información de los equipos, son variables que han ido modificando el estilo de juego, hacia un estilo más defensivo y físico, en el que prima la planificación y la innovación, no tiene cabida los jugadores y entrenadores que son absorbidos por la gran cantidad de información que se extrae en cada partido. Dado este paradigma, nos preguntamos, ¿Cómo ha evolucionado el estilo de juego?

Además, dentro de los datos obtenidos de cada equipo, los directores deportivos de cada equipo quieren saber cómo influye cada jugador en el juego global del equipo. Cada equipo hace sus propios análisis de datos para determinar fichajes, estrategias a llevar a cabo con cada contrincante, sacar el mayor partido a cada componente del equipo, determinar minutos en cancha y partidos en los que participará cada jugador

Mediante este trabajo intentaremos dar respuesta a alguna de estas preguntas: ¿Los grandes jugadores siguen teniendo la influencia en el juego que tenían antes? ¿Cómo saben las franquicias cuál es el estilo de jugador que mejor se adapta a las condiciones del equipo?

La búsqueda de una variable que cuantifique esta métrica y tenga en cuenta todos los aspectos que interviene durante un partido será otro de los objetivos del trabajo.

El conocimiento del estilo de juego y de qué jugador es más óptimo para nuestro estilo de juego nos llega a preguntarnos si es posible la proyección de un modelos que nos permite averiguar las probabilidades y derrotas. Esta variable la hemos aplicado junto con las distancias que hay entre los campos de cada equipo y la influencia de los lesionados en esta.

# ANTECEDENTES

Todos los entrenadores e investigadores han buscado la forma de conseguir sacarle el máximo rendimiento a las estadísticas que se recogen durante el tiempo del partido, tanto de sus propios jugadores como del resto de equipos de la NBA. Dentro de los avances obtenidos en el desarrollo y aplicación de nuevos métodos para el análisis cuantitativo de datos deportivos, sobre todo utilizando técnicas estadísticas. Esa acumulación incalculable de datos requiere de conclusiones que nos permitan analizar el estado de nuestro equipo, valorar el impacto de nuestros jugadores dentro del colectivo, prever futuras líneas de actuación y hasta cambiar puntualmente el estilo de juego en función de las estadísticas propias y del equipo rival

Desde la implementación hace más de un lustro de la medición, utilización y comprensión de los datos de estadísticas convencionales como, puntos, rebotes, asistencias nos ha permitido ir conociendo cómo ha evolucionado el juego.

La incorporación de tecnología para recabar datos se ha ido realizado paulatinamente, aunque el mayor avance se produce en el año 2013, con la instalación de cámaras de registro de movimientos, para proporcionar a los entrenadores, jugadores y aficionados la información de todos los partidos.

Los principales estudios se centran en el **ritmo de juego** (se centra en el uso de las faltas, las posesiones, y las jugadas desarrolladas durante el tiempo de partido), los **porcentajes de tiro** (desarrollo de un algoritmo para conocer le valor medio de los tiros en el campo, contando con un valor extra para el lanzamiento de línea de tres) y el intento de conocer mediante el análisis de datos cual es el **mejor ataque y la mejor defensa.**

Siendo estos los temas principales, se han desarrollado varios estudios acerca de las pérdidas de balón y varios estudios sobre las votaciones para los jugadores en el draf y la evolución del juego de triples.

Entre los estudios que destacan por su notoriedad se encuentran, el estudio de Ibáñez y Col y Sampaio y Janeira ambos del 2003 en el que sus conclusiones es que los equipos que anotan más tiros de 2 puntos y recogen más balones debajo del arco obtienen un índice mayor de partidos ganados.

La tesis de Sampaio y Leite, recoge los datos del Eurobasket d 2005, en el que concluyen que las estadísticas más diferenciadora se situaba en la línea d tres, los puntos fallados y las asistencias. Con este estudio se planteaba que la situación ofensiva se concluía con un tiro cercano, con elevadas probabilidades de aciero, para minimizar los riesgos del tiro exterior y escogiendo las mejores opciones de lanzamiento.

un estudio de Hierro (2002), tras analizar la liga ACB y la NBA, concluye que los equipos con mejores porcentajes de tiros de 2 y de 3, mayor número de faltas recibidas, tapones y asistencias, obtenían más victorias; no considerando el número de pérdidas de balón y los tiros libres como aspectos definitivos.

No todos los estudios se centran en las estadisiticas, ya que algunos toman de referencia la edad de los jugadores en la canga, (García, Ibañez y Feu (2007)), así como encontrar la posición más influyente en los partidos. Este estudio (Sampaio et Al, 2009), determinó que los jugadores más importantes eran los bases titulares y los pívots suplentes, reflejándose su aportación en los lanzamientos de 3 puntos, los rebotes y los lanzamientos de 2 puntos.

Otra de las líneas de investigación que hemos encontrado en el análisis de la literatura científica, es el estudio de las victorias y las derrotas en función de la condición del equipo como local o visitante. En la mayoría de los casos, los equipos que juegan en casa, tienen mayor ventaja (Courneya y Carron, 1992; Sánchez et al., 2001; Smith, 2005), debido a factores como viajes realizados, el conocimiento de la cancha de juego, la presión percibida por los árbitros, la familiaridad con el estadio, el público y determinadas variables psicológicas

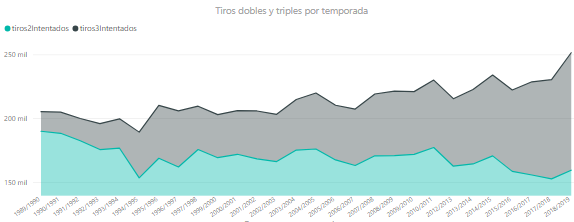
Diversos estudios cuentan con bbdd obtenidas de páginas de datos gratuitas, pero que en nuestro caso, hemos querido sacar nuestras propias variables, para así enriquecer este estudio. El objetico es la creación de un modelo que contenga como base la información más completa de estadísticas de los jugadores a nivel individual y como equipo.

**Cómo ha evolucionado el juego del baloncesto.**

El juego que se desarrolla actualmente en la NBA no es el mismo juego que se realizaba cuando se comenzaron a recoger datos. El análisis de datos de juego ha creado un antes y un después en el mundo del baloncesto. Tanto así que ha dado lugar a cambios importantes en la forma de jugar y las estrategias de los equipos.

La variación en la forma que se están obteniendo los datos y del año en el que comenzó a registrarse información está repercutiendo en el estilo de juego desarrollado por los equipos y en la forma de afrontar los partidos, buscando las estrategias más eficientes para conseguir el mayor número de puntos.

Con un simple vistazo a los datos de tiros por temporada a lo largo de los últimos 30 años, podemos ver que el volumen ha experimentado un gran aumento.



Esto se debe a que cada vez el juego es más rápido y por lo tanto da lugar a un mayor número de jugadas por partido. Y con ello, la oportunidad de anotar una mayor cantidad de tantos en cada encuentro.

Pero, ¿cómo podemos conocer que el juego es más rápido? ¿Se producen más intentos de tiro ahora que hace 20 años?

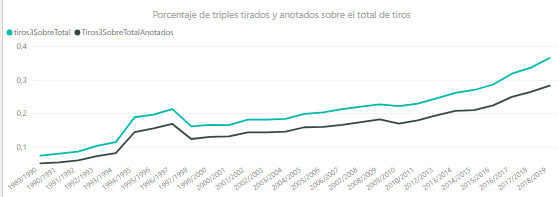
En el siguiente gráfico, comenzamos a analizar los puntos por temporada desde 1989/1990 hasta la temporada 2018/2019. La evolución se produce de forma lineal a lo largo del tiempo, teniendo un margen de crecimiento muy parecido en todos ambas métricas.



Sin embargo, el hecho de tirar más veces a canasta no es lo único que ha favorecido el aumento de puntos por temporada. Ha sido otro de los cambios tácticos importantes que ha dado la analítica de datos lo que más ha influido en este aumento.

Se trata de la importancia que se está dando a los tiros triples sobre los de 2 puntos, lo cual está invirtiendo el porcentaje de estos sobre el total de tiros, tanto en el caso de tiros intentados como en los anotados.

Como podemos ver en el siguiente gráfico, mientras en la temporada de 1989/1990, los triples sólo suponían un 7% sobre el total de tiros realizados y un 5% sobre los anotados, en la temporada de 2018/2019 estos porcentajes suponen un 37% y 28% respectivamente. Sobretodo se observa un aumento mucho más acusado en las últimas 5 temporadas, donde el incremento es de 11 y 7 puntos porcentuales respectivamente.



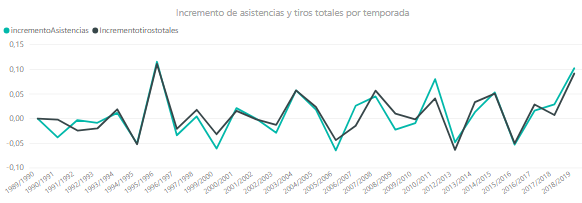
La respuesta a por qué se está llevando a cabo esta sustitución de tiros de dos puntos por tiros triples, se encuentra en la esperanza matemática que supone cada uno de ellos.

Como es evidente, el porcentaje de efectividad es superior en tiros de 2 puntos que en tiros de 3 puntos, -además este dato no ha experimentado grandes variaciones en el tiempo a pesar de que se esté tendiendo a una mayor práctica de tiros triples sobre dobles-, siendo de una media del 50% y del 35% respectivamente. Sin embargo, el hecho de anotar un triple, supone en el marcador un 50% más de puntos que un tiro doble.

Además, también se trata de una cuestión táctica. Al realizar el ataque desde fuera como es el caso de los triples, más tienen que abrirse los rivales. De esta manera resultaría más fácil atacar el aro, y por parte de la defensa, más difícil el apoyo entre los jugadores, ya que tienen que cubrir más terreno.

Una idea que podría surgir del hecho de que se esté tendiendo a un juego más basado en triples y se estén llevando a cabo jugadas más rápidas, es que el tipo de juego esté tendiendo a un modelo más individualista y menos colaborativo. Sin embargo, podemos descartar esa idea con un vistazo rápido a la evolución de los datos de tiros y asistencias en conjunto.

Como podemos observar en el siguiente gráfico, el incremento o disminución porcentual de cada temporada respecto a la anterior sigue una evolución muy similar entre ambas medidas, por lo que no podemos evidenciar un cambio en el juego hacia una tendencia más o menos individualista. Los datos de tiros totales y asistencias evolucionan en la misma dirección y en proporciones similares.



# HIPOTESIS

Con la presentación de este trabajo final de Data analytics nuestro objetivo es la creación de un modelo predictivo que nos ayude a identificar que equipo va a ganar un partido durante el campeonato.

Sabemos que durante el desarrollo de un partido intervienen una gran cantidad de variables que puedan hacer cambiar el resultado final de un partido. Lesiones, acierto de tiros, faltas cometidas son algunas de las variables más importante que pueden llegar a afectar a los partidos.

Dentro de conocer, la importancia para escoger estas métricas hemos desarrollado un modelo que nos explique cual ha sido la evolución del baloncesto en los últimos años, intentado descifrar mediante los datos como ha evolucionado cada una de estas métricas y su influencia en los resultados.

**PENDIENTE MODELO DE BRYAN PARA DARLE FORMA.**

# METODOLOGIA DE TRABAJO Y VARIABLES

Introducción:

Tras una primera tentativa de consumir la información de la API oficial de la NBA (data.nba.net), nos decantamos por emplear scraping sobre diferentes fuentes debido principalmente a dos problemas que nos encontramos:

1. Falta de consistencia de la API: presenta un año de inicio relativamente alto (2017) y deja de estar disponible para los partidos jugados desde el 13 de marzo de 2019, esto nos plantea un gran problema porque, ya que el fichero de entrada final tenía únicamente con unos 2000 registros para entrenar al modelo.
2. Problemas para añadir variables al DataFrame: tendríamos que la información obtenida en la API enriquecer con otras fuentes de datos, ya que algunas de las variables que se pretenden incluir en el estudio no están recogidas, esto nos plantea otro problema: los códigos que utilizan para hacer referencia a los equipos y jugadores no son fácilmente *mapeables*, de hacerlo, sería un proceso manual y costoso.

Esta primera versión de la capa de extracción del proyecto sirvió como prueba de concepto, para delinear nuestros objetivos, pero finalmente resultó insuficiente.

El scraping en cambio nos resulta tremendamente flexible a la hora de realizar las uniones entre las diferentes fuentes de datos, y las fuentes que escogemos resultan ser mucho más consistentes en cuando a su disponibilidad temporal. Solo requerimos de transformaciones menores para adaptar el nombre los jugadores de ESPN al formato de basketball-reference, para adaptar la fecha de basketball-reference al formato de shrpsports y la creación de dos mapas (diccionarios) que unifiquen el nombre del equipo (i.e. Chicago Bulls), el nombre abreviado (CHI) y el nombre de la ciudad (Chicago).

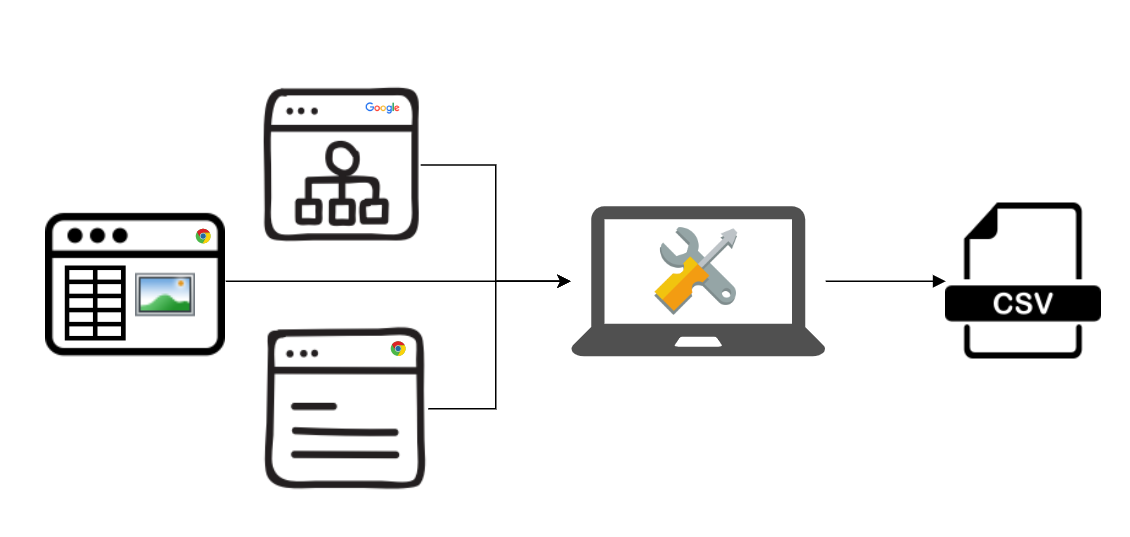
En resumen, habría sido un trabajo enormemente más sencillo consumir un json estructurado, con una única fuente de datos, y preocuparnos únicamente del número de peticiones por hora para evitar ser bloqueados, pero la cantidad de información sería insuficiente. Además, la significancia del proyecto sería menor, ya que existe una gran cantidad de estudios, como hemos comentado, que se limitan a estudiar los Boxscores y clasificaciones que la API suministra. Empleando este método podemos aportar nuevas variables al estudio que no se encuentran recogidas en ésta.

Con nuestro método conseguimos finalmente obtener 14000 registros (de 2010 en adelante) en el formato de entrada del modelo, frente a los 2000 de la primera versión, si bien decidimos posteriormente limitar ámbito temporal de nuestro estudio, debido a los cambios que se producen en el baloncesto en los últimos años, que hemos comentado anteriormente.

Metodología:

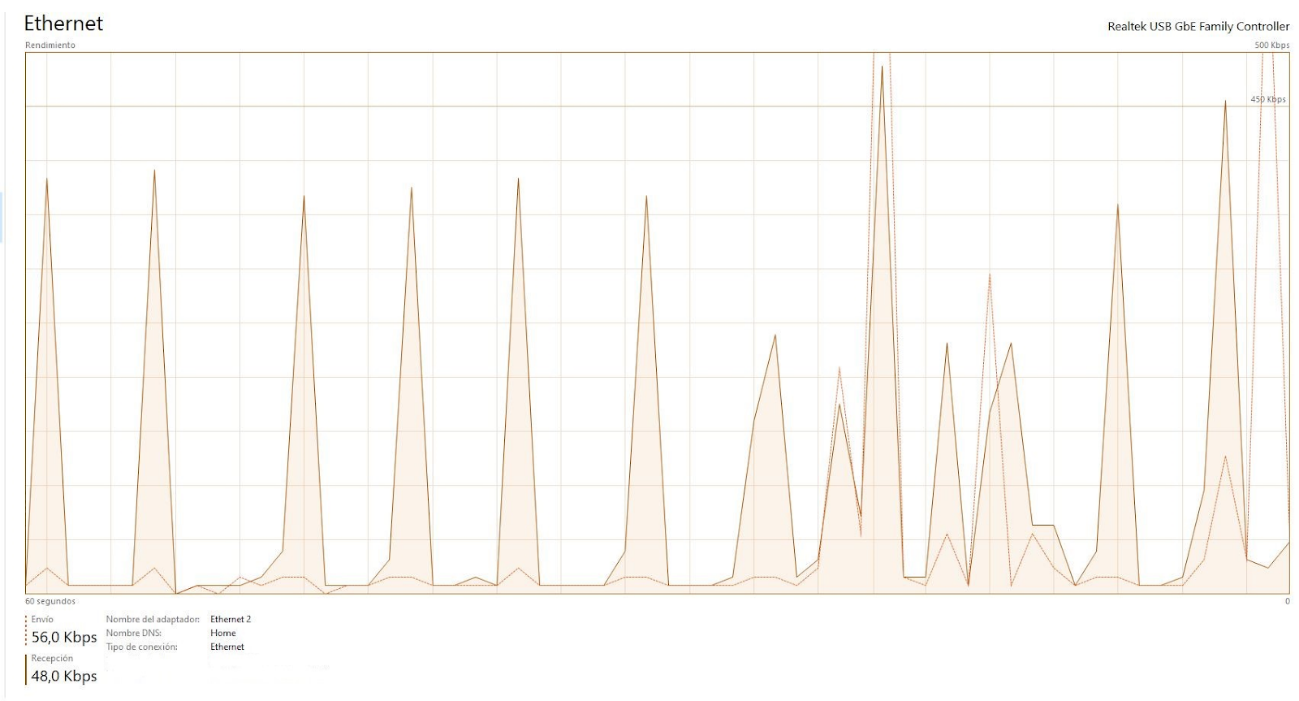
La extracción de los datos se lleva a cabo realizando peticiones GET a los endpoints mediante Python empleando las librerías request, y Pandas para recopilar la información en DataFrames, principalmente.

El proceso se basa en descomponer mediante la función "Split" la respuesta del endpoint para recopilar la información útil y desechar el resto, también se llevan a cabo en estos procesos unas transformaciones mínimas en los datos y uniones de DataFrames cuando es necesario, para dar lugar a ficheros csv que contengan la información que necesitamos para el análisis. Para dividir correctamente la respuesta de la request y generar el DataFrame hay que analizar el texto plano que llega y observar los patrones que repiten, de este modo conseguimos transformar la información no estructurada, o semiestructurada, a información estructurada con la que podemos realizar nuestro análisis.



Este proceso también nos permitirá acceder a una mayor cantidad de información que si empleásemos un DataFrame de los suministrados en portales como Kaggle, también nos permitirá incorporar más variables a nuestro estudio y obtener otra información que utilizaremos como referencia o control a algunas de las medidas de nuestro análisis.

La fuente de datos principal será www.basketball-reference.com. También, como ya hemos ido comentando, consumimos información de data.nba.net, de www.shrpsports.com y www.geodatos.net. Realizaremos las llamadas de forma que no sature a los receptores de las mismas, es decir, de manera secuencial (sin yield y sin emplear la librería scrapy), como podemos ver en la imagen. Como contrapartida nos encontraremos con procesos muy costosos en el tiempo.



Obtendremos mediante los extractores información relativa a:

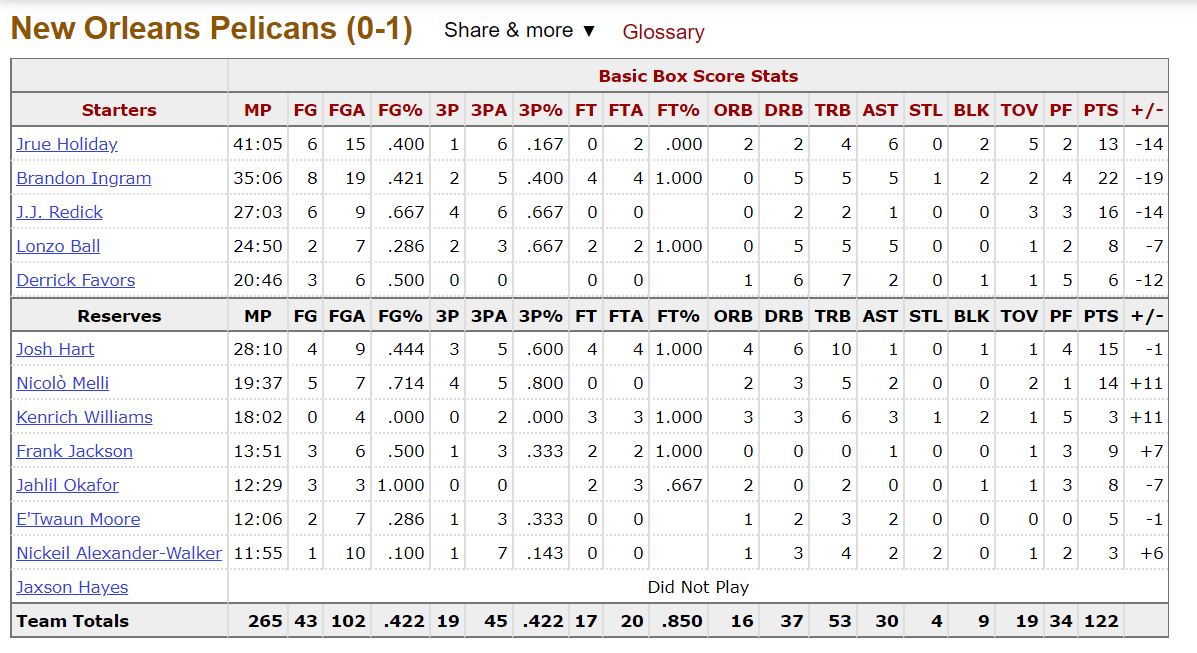
* Convocatoria de los partidos
* Clasificación previa al encuentro
* Estadísticas de los partidos (Boxscore)
* Nominaciones a los premios individuales
* Sueldos
* Distancias entre las ciudades que tienen equipo en la NBA
* Estadísticas al descanso
* Evolución del juego (estadísticas acumuladas de todos los equipos, por temporada)

Extractores:

1. **StatsJugadoresV2.py**: Extrae la información de cada jugador y sus estadísticas en cada partido desde el 2010 a 2020. Para ello se realiza una llamada por equipo y temporada para obtener el calendario del equipo y de manera recursiva realiza llamadas al detalle de cada partido del calendario para obtener las siguientes variables.

* MP: Minutos jugados
* FG: Tiros de campos anotados
* FGA: Tiros de campo intentados
* FG%: Porcentaje de tiros de campo
* 3P: Triples anotados
* 3PA: Triples intentados
* 3P%: Porcentaje de triples
* FT: Tiros libres anotados
* FTA: Tiros libres intentados
* FT%: Porcentaje de tiros libres
* ORB: Rebotes ofensivos
* DRB: Rebotes defensivos
* TRB: Rebotes totales
* AST: Asistencias
* STL: Recuperaciones
* BLK: Tapones
* TOV: Pérdidas
* PF: Faltas personales
* PTS: Puntos
* +/-: Más menos (marcador parcial con el jugador en pista)
* TS% -- True Shooting Percentage
* eFG% -- Effective Field Goal Percentage
* 3PAr -- 3-Point Attempt Rate
* FTr -- Free Throw Attempt Rate
* ORB% -- Offensive Rebound Percentage
* DRB% -- Defensive Rebound Percentage
* TRB% -- Total Rebound Percentage
* AST% -- Assist Percentage
* STL% -- Steal Percentage
* BLK% -- Block Percentage
* TOV% -- Turnover Percentage
* USG% -- Usage Percentage
* ORtg -- Offensive Rating
* DRtg -- Defensive Rating
* BPM -- Box Plus/Minus

En la imagen vemos un ejemplo de un Boxscore a scrapear, en la imagen solo se muestran algunas de las variables (puede verse completo en https://www.basketball-reference.com/boxscores/201910220TOR.html):



1. **StatsJugadoresV2\_h1.py**: Con este proceso buscamos documentar la información de cada jugador y sus estadísticas al descanso en cada partido desde 2010 a 2020. Para ello realizamos una llamada para obtener el calendario del equipo, que nos facilita la URL del Boxscore de cada partido. Esta información viene a nivel de jugador, debemos agrupar por equipo y unir las filas de las variables del equipo local a las del visitante. Recogemos la siguiente información:

* FG: Tiros de campos anotados
* FGA: Tiros de campo intentados
* FG%: Porcentaje de tiros de campo
* 3P: Triples anotados
* 3PA: Triples intentados
* 3P%: Porcentaje de triples
* FT: Tiros libres anotados
* FTA: Tiros libres intentados
* FT%: Porcentaje de tiros libres
* ORB: Rebotes ofensivos
* DRB: Rebotes defensivos
* TRB: Rebotes totales
* AST: Asistencias
* STL: Recuperaciones
* BLK: Tapones
* TOV: Pérdidas
* PF: Faltas personales
* PTS: Puntos

1. **ResultadosClasificacionv2.py:** Extrae la información relativa a la clasificación en todas las fechas posibles desde octubre de 2009 a abril de 2020. Los datos los obtenemos a través de la realización de una serie de llamadas a [www.shrpsports.com](http://www.shrpsports.com). Recogemos los siguientes campos:

* Fecha (mes y día)
* Año
* Temporada
* Equipo
* Posición de conferencia
* Victorias
* Derrotas
* Porcentaje victorias / derrotas
* Diferencia con el líder de conferencia
* Victorias como local
* Derrotas como Local
* Victorias intra-división
* Derrotas intra-división
* Victorias intra-conferencia
* Derrotas intra-conferencia
* Victorias inter-conferencia
* Derrotas inter-conferencia

También hemos realizado las mismas llamadas en el extractor StatsjugadoresV2.py para obtener los mismos datos que obtenemos en este extractor a nivel de equipo, para el local y el visitante.

Para adaptar las nomenclaturas, hemos realizado un diccionario con los nombres de cada equipo en cada fuente (shrsports y basketball-reference,com) para poder realizar la unión de estos datos en un único dataframe.

Es importante indicar en este punto que la unión se realiza de modo que en cada línea del Dataframe se recoja el resultado del partido, las estadísticas de los equipos en el mismo y la clasificación previa al encuentro.

1. **DNP.py:** Incluye la información de qué jugadores (nombre y equipo) no participaron en cada partido durante el campeonato. Para ello realizamos llamadas para obtener el calendario y para obtener el Boxscore de cada partido, recogiendo la información del jugador que no jugó y el ID del partido.
2. **Premios.py:** Recoge la información de los jugadores que han recibido votos para recibir el premio MVP (premio a mejor jugador del año) desde 2010 hasta 2020. Se registra el nombre del jugador, el número de votos en primera posición, los puntos obtenidos, los puntos máximos y el share de éstos.
3. **Distancias.py:** Mediante llamadas a [www.geodatos.net](http://www.geodatos.net/) obtenemos las distancias entre las ciudades de todos los equipos que componen la NBA. Realizamos las llamadas iterando sobre la misma lista de equipos (que se relacionan con la ciudad mediante un diccionario) dos veces, bajo la condición de que el equipo no sea el mismo. Obtenemos así la distancia de Equipo A al Equipo B en kilómetros. Nos servirá para incluir alguna variable que recoja la distancia viajada por el equipo visitante en el modelo.
4. **Evolucion.py:** Recoge las estadísticas totales de la NBA en cada temporada para realizar un análisis sobre la evolución en el baloncesto NBA desde 1990 hasta 2020. La fuente de estos datos es [www.basketball-reference.com](http://www.basketball-reference.com/). Este análisis, como hemos visto anteriormente, nos ayuda a establecer un límite temporal a la información que incluimos en el modelo.

* FG: Tiros de campos anotados
* FGA: Tiros de campo intentados
* FG%: Porcentaje de tiros de campo
* 3P: Triples anotados
* 3PA: Triples intentados
* 3P%: Porcentaje de triples
* FT: Tiros libres anotados
* FTA: Tiros libres intentados
* FT%: Porcentaje de tiros libres
* ORB: Rebotes ofensivos
* DRB: Rebotes defensivos
* TRB: Rebotes totales
* AST: Asistencias
* STL: Recuperaciones
* BLK: Tapones
* TOV: Pérdidas
* PF: Faltas personales
* PTS: Puntos

1. **Sueldos.py:** Este extractor obtiene la información de los sueldos de los jugadores de la NBA de 2010 en delante de www.espn.com/nba/salaries/. Exporta la información a nivel de jugador (nombre del jugador, el equipo, la temporada y el sueldo en dólares). La información en la web viene paginada con un número de páginas variable de un año a otro, de modo que el extractor recoge la información realizando llamadas que incrementan en uno el número de página hasta que se produce una excepción, momento en el que pasa al siguiente año.

Generación del fichero de entrada:

Una vez completado todo el proceso de extracción de datos, tenemos que generar un fichero que sirva de input al modelo predictivo recogiendo toda la información. El fichero final contendrá la siguiente la información.

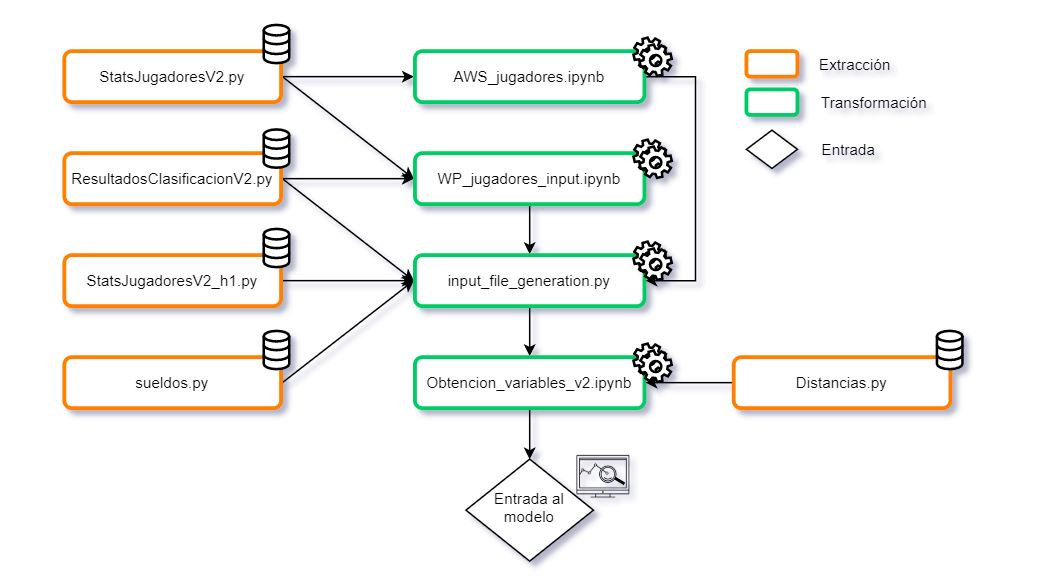
|  |  |
| --- | --- |
| Información | extractor |
| Clasificación antes del partido (posición, rachas, etc) | ResultadosClasificacionV2.py |
| Boxscore al descanso | StatsJugadoresV2\_h1.py |
| Suma de los sueldos del equipo | Sueldos.py |
| AWS | StatsjugadoresV2.py |
| WP | StatsjugadoresV2.py y ResultadosClasificacionV2.py |
| Divisiones y conferencias | - |
| Boxscore medio agrupado antes del encuentro | ResultadosClasificacionV2.py |

Además, durante este proceso, hemos eliminado toda la información duplicada o innecesaria, así como llevado a cabo la adaptación de toda la información al formato local /visitante.

Además de los campos obtenidos directamente en el proceso de scraping se realizan transformaciones para obtener los siguientes:

* Ultimos10Victorias: las victorias conseguidas en los últimos diez partidos \* jugados.
* Ultimos10Derrotas: las derrotas conseguidas en los últimos diez partidos jugados.
* Sueldo: La suma del sueldo del equipo en la temporada.
* División: división a la que pertenece el equipo.
* Conferencia: conferencia a la que pertenece el equipo.
* AWS\_MEDIO\_AGRUPADO: La medida AWS será explicada más adelante, una vez calculada para cada jugador, hacemos un rolling mean y un shift, para obtener la media hasta antes del partido del jugador en la temporada, posteriormente se agrupan por equipo y se realiza el join, obteniendo una valoración conjunta de los jugadores que participan en el encuentro por equipo.
* WP\_MEDIO\_AGRUPADO: Se trata de una medida análoga al AWS.

El flujo de la extracción y transformación de datos para generar un fichero de entrada es el siguiente:



### NUEVAS VARIABLES

Con el objetivo de enriquecer nuestro modelo de predicción de resultados, hemos elaborado nuevas variables que complementan la información de los datos extraídos durante el primer proceso.

En un primer lugar obtenemos variables a partir del archivo stats\_jugadores\_27\_04\_20.csv. Este archivo consta de las siguientes variables:

En definitiva, este archivo tiene cada estadística de cada partido por equipo y por jugador a nivel fila siendo las estadísticas cada columna. Es decir, sigue una estructura parecida a la siguiente:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID Partido | Equipo | Jugador | Estadística\_1 | Estadística\_n |
| /boxscores/201510280BOS.html | Boston Celtics | Player1 | 5.0 | 2.0 |
| /boxscores/201510280BOS.html | Boston Celtics | Player2 | 5.0 | 3.0 |
| /boxscores/201510280BOS.html | Boston Celtics | Player3 | 20.0 | 4.0 |
| /boxscores/201510300BOS.html | Boston Celtics | Player1 | 7.0 | 6.0 |
| /boxscores/201510300BOS.html | Boston Celtics | Player2 | 3.0 | 2.0 |

A partir de este archivo se realizarán las transformaciones necesarias para saber qué impacto tiene cada estadística de cada jugador en el equipo mediante un modelo de regresión logística. Con ello, se pretende saber que impacto tiene cada estadística de cada jugador dentro de un equipo y cómo aporta al desempeño del equipo, si cogiendo rebotes, metiendo puntos etc.

Con este planteamiento el objetivo final es tener un dataset por cada equipo con granularidad partido teniendo como columnas cada jugador con cada estadística asociada. Esto quiere decir que como columnas tendremos player1\_estadística1, player1\_estadística2, player2\_estadística1, player2\_estadística2, playern\_estadíticam siendo n el número de jugadores por temporada y m el número de estadísticas finales.

Para conseguir este dataset lo primero que se realiza es la transformación de ciertas variables origen al tipo deseado. Esto se explica a continuación:

1. Se transforman las siguientes variables de float a int: con ello se pretende poder operar de forma correcta con los valores y evitar posibles fallos ya que las variables que se citan posteriormente no tiene ningún sentido que sean decimales.
   * Season
   * Puntos
   * Faltas
   * Perdidas
   * Robos
   * Asistencias
   * Rebotes total
   * Rebotes ofensivos
   * Tiros libres intentado
   * Tiros libres anotados
   * Tiros de tres intentados
   * Tiros de tres anotados
   * Tiros intentados
   * Team Points
   * Opponent Points
   * Tiros anotado
   * Rebotes defensivo
   * Tapones
2. Transformación variable Minutes Played: esta variable viene con el formato mm:ss siendo mm los minutos disputados por un jugador en un determinado partido y ss los segundos disputados por un jugador en un determinado partido. Para poder realizar operaciones con esta variable se transforma a un formato mm.ss de tipo float.
3. Transformación variable ID Partido: esta variable viene con información innecesaria de la extracción que puede generar ruido. Esta variable tiene el siguiente formato:

“/boxcores/fecha\_partido0abreviaturaequipoquejuegalocal.html”

Siendo:

* + fecha\_partido la fecha en la que se juega el partido en formato yyyymmdd
  + abreviaturaequipoquejuegalocal la abreviatura en tres letras del equipo que juega como local.

Con ello se realizan transformaciones para solo obtener el ID Partido con el formato:

“fecha\_partido0abreviaturaequipoquejuegalocal”

1. Transformación variable Date: viene con un formato difícil de operar con el ya que contiene una parte literal para identificar el mes y otra parte numérica para identificar el número.

“Mes\_abreviado día\_partido” 🡪 Ejemplo: Oct 28

Siendo:

* Mes\_abreviado el mes en el que se juega el partido con formato de tres letras.
* Día\_partido: día en el que se juega el partido en formato int.

Se transforma la variable en otra llamada “Complete Date” para que contenga un formato “yyyymmdd”, siendo:

* + yyyy: el año en el que se juega el partido en formato int.
  + mm: el mes en el que se juega el partido en formato int.
  + dd: el día en el que se juega el partido en formato int.

Para ello, se utiliza la función calendar.month\_abbr con la que se puede obtener la equivalencia entre la abreviatura del mes y el número correspondiente a cada mes. Posteriormente se coge la variable year y date y se juntan en una única variable, “Complete Date” en la que se guarda con el formato anteriormente indicado. Este formato es el más común usado en bases de datos ya que permite operar de manera sencilla con las fechas.

Una vez se han formateado las variables se transforma el dataset mediante los pasos siguientes:

1. Se divide el df inicial en tantos dataframes como equipos existan.
2. Cada df por equipo se divide en un dataframe por cada jugador.
3. Se transforma el df por cada jugador para que tenga una estructura. jugador\_estadística1, jugador\_estadística2, jugador\_estadística\_m siendo m todas las posibles estadísticas existentes.
4. Se juntan todos los dataframes por jugador en un único dataframe por equipo. Cada dataframe está almacenado en una posición de un diccionario.

Con ello, se obtiene un diccionario de dataframes que cada uno de ellos para cada temporada tiene aproximadamente 82 filas (una por cada partido) y unas 1052 columnas.

Una vez obtenido este diccionario de dataframes la idea era realizar un modelo que midiera la contribución de cada estadística de cada jugador al global del equipos pero finalmente se decidió no realizar por las siguientes razones:

* Se optaron por utilizar otras medidas ya predefinidas que permiten medir el desempeño global de un jugador y poder compararlos entre ellos.
* Demasiadas pocas filas por dataframe.
* Demasiadas columnas para en comparación con el número de filas.
* Simplemente se iba a ver la influencia de cada estadística sobre el resultado, pero no se iba a poder clasificar que jugadores son mejores que otros.
* Solo iba a servir para hacer reporting y creemos que no aporta el valor suficiente.

**OTRAS VARIABLES**

Además, se calculan más variables para los modelos a desarrollar. Para calcular estas variables se tomarán las variables obtenidas en la extracción.

Para calcular estas nuevas variables se utilizarán tres orígenes:

* Input.csv
* Distancias.csv
* stats\_jugadores\_27\_04\_20.csv

Estos archivos se guardan mediante el comando read\_csv en los siguientes dataframes:

* Input.csv 🡪 stats\_equipos
* Distancias.csv 🡪 distancias
* stats\_jugadores\_27\_04\_20.csv 🡪 stats\_jugadores

Teniendo en cuenta que en el input.csv el máximo nivel de granularidad viene dado por un id, se eliminan todos los posibles duplicados.

Para conseguir este dataset lo primero que se realiza es la transformación de ciertas variables origen al tipo deseado. Esto se explica a continuación:

1. Se transforman las siguientes variables de float a int: con ello se pretende poder operar de forma correcta con los valores y evitar posibles fallos ya que las variables que se citan posteriormente no tiene ningún sentido que sean decimales.
   * Season
   * Puntos
   * Faltas
   * Perdidas
   * Robos
   * Asistencias
   * Rebotes total
   * Rebotes ofensivos
   * Tiros libres intentado
   * Tiros libres anotados
   * Tiros de tres intentados
   * Tiros de tres anotados
   * Tiros intentados
   * Team Points
   * Opponent Points
   * Tiros anotado
   * Rebotes defensivo
   * Tapones
2. Transformación variable Minutes Played: esta variable viene con el formato mm:ss siendo mm los minutos disputados por un jugador en un determinado partido y ss los segundos disputados por un jugador en un determinado partido. Para poder realizar operaciones con esta variable se transforma a un formato mm.ss de tipo float.
3. Transformación variable ID Partido: esta variable viene con información innecesaria de la extracción que puede generar ruido. Esta variable tiene el siguiente formato:

“/boxcores/fecha\_partido0abreviaturaequipoquejuegalocal.html”

Siendo:

* + fecha\_partido la fecha en la que se juega el partido en formato yyyymmdd
  + abreviaturaequipoquejuegalocal la abreviatura en tres letras del equipo que juega como local.

Con ello se realizan transformaciones para solo obtener el ID Partido con el formato:

“fecha\_partido0abreviaturaequipoquejuegalocal”

Una vez realizadas estas transformaciones se realiza un proceso de transformación de variables:

1. Obtención día partido: se calcula la columna number\_date para obtener el día en el que se jugará el partido. Esta variable es importante porque permitirá realizar de forma sencilla diversos cálculos.

Esta variable se calcula a partir del ID Partido. Como se ha comentado anteriormente esta variable contiene ruido y sigue el siguiente formato:

“/boxcores/fecha\_partido0abreviaturaequipoquejuegalocal.html”

Siendo:

* + fecha\_partido la fecha en la que se juega el partido en formato yyyymmdd
  + abreviaturaequipoquejuegalocal la abreviatura en tres letras del equipo que juega como local.

Para obtener la fecha se realizan lo siguientes pasos:

Eliminación letras mediante el comando replace sustituyendo las letras por “” es decir, por nada.

De forma análoga se elimina el carácter “.” y “/”.

Al ser las primeras ocho cifras las indicativas de la fecha se realiza un slice para extraerlas y se guardan en la columna “number\_date”.

1. Cálculo día anterior, dos días anteriores, tres días anteriores y previous\_10\_days: se calcula en cada registro que día ha sido el anterior, dos días anteriores y tres días anteriores. Esto se guardará en las variables previous\_date, two\_days\_ago y three\_days\_ago. Esto permitirá realizar transformaciones para obtener el resto de variables que utilizará el modelo.
2. Para calcular estas variables se utiliza timedelta de la librería datetime. A partir de la variable number\_date, anteriormente calculada, se le resta mediante timedelta uno, dos, tres o diez días según la variable que se desee calcular.
3. Una vez ya tenemos las variables calculadas se transforman a formato datetime.
4. Cálculo variable prórroga: se desea calcular si un partido ha tenido prórroga o no. Para ello, se realizan los siguientes pasos:
5. Se crea un dataframe llamado stats\_jugadores\_ordered con solo tres columnas con el ID\_Partido y las variables visitor\_prorrogue y local\_prorrogue. Este dataframe por lo tanto no tendrá como granularidad el trío de columnas jugador/equipo/id\_partido sino que será el id\_partido. Este dataframe, además, será mucho más manejable ya que su número de filas y columnas es menor que al original.
6. En la columna ID\_Partido del dataframe stats\_jugadores\_ordered se añade todos los ID\_Partido únicos es decir, se añaden los id\_partido sin repeticiones ya que en debida a la naturaleza del daframe stats\_jugadores estos estaban repetidos.
7. Se suma todos los minutos de cada jugador por partido y equipo en el dataframe stats\_jugadores. De esta forma se obtendrá los minutos que ha durado el partido.
8. Se recorre el dataframe stats\_jugadores\_ordered y se crea un dataframe en el que se guardan todos los registros del dataframe stats\_jugadores que coindicen con el ID\_Partido del índice correspondiente del dataframe stats\_jugadores\_ordered.
9. Si el índice mínimo del borrador es mayor que 50.0 significa que en ese partido ha habido prórroga por lo que la variable visitor\_prorrogue y local\_prorrogue se pone a 1. Un partido dura 48 minutos y la prórroga 5 minutos lo que suman 53 min, por lo que el umbral para decidir si hay prórroga o no es 50 min por si ha habido algún ligero descuadre en la extracción de los datos.
10. Este dataframe mediante un merge se fusiona con el dataframe stats\_equipos.
11. Creación de variables a calcular: se crean las siguientes variables que se utilizarán en el modelo posterior.

* local\_dif\_between\_previous\_game: se calcula la diferencia de días respecto al útlimo partido. Se calcula esta
* visitor\_dif\_between\_previous\_game: se calcula la diferencia de días respecto al útlimo partido. Se calcula esta
* local\_played\_previous\_date: se calcula si el local ha jugado el día anterior. Se calcula esta variable porque pensamos que el que el local juegue el día anterior puede tener influencia sobre el partido ya que puede estar más cansado.
* visitor\_played\_previous\_date: se calcula si el visitante ha jugado el día anterior. Se calcula esta variable porque pensamos que el que el visitante juegue el día anterior puede tener influencia sobre el partido ya que puede estar más cansado.
* local\_played\_two\_days\_ago: se calcula si el local ha jugado dos días antes. Se calcula esta variable porque pensamos que el que el local juegue dos días antes puede tener influencia sobre el partido ya que puede estar más cansado. Se diferencia del día anterior porque pensamos que la diferencia de significancia puede ser notable.
* visitor\_played\_two\_days\_ago: se calcula si el visitante ha jugado dos días antes. Se calcula esta variable porque pensamos que el que el visitante juegue dos días antes puede tener influencia sobre el partido ya que puede estar más cansado. Se diferencia del día anterior porque pensamos que la diferencia de significancia puede ser notable.
* local\_played\_three\_days\_ago: se calcula si el local ha jugado tres días antes. Se calcula esta variable porque pensamos que el que el local juegue tres días antes puede tener influencia sobre el partido ya que puede estar más cansado. Se diferencia del día anterior y dos días anteriores porque pensamos que la diferencia de significancia puede ser notable.
* visitor\_played\_three\_days\_ago: se calcula si el visitante ha jugado tres días antes. Se calcula esta variable porque pensamos que el que el vistante juegue tres días antes puede tener influencia sobre el partido ya que puede estar más cansado. Se diferencia del día anterior y dos días anteriores porque pensamos que la diferencia de significancia puede ser notable.
* local\_played\_prorrogue\_previous\_date: se calcula si el local ha jugado el día anterior prórroga. Se calcula esta variable porque pensamos que el que el local juegue el día anterior prórroga puede tener influencia sobre el partido ya que puede estar más cansado.
* visitor\_played\_prorrogue\_previous\_date: se calcula si el visitante ha jugado el día anterior prórroga. Se calcula esta variable porque pensamos que el que el visitante juegue el día anterior prórroga puede tener influencia sobre el partido ya que puede estar más cansado.
* local\_played\_prorrogue\_two\_days\_ago: se calcula si el local ha jugado el dos días anteriores prórroga. Se calcula esta variable porque pensamos que el que el local juegue dos días anteriores prórroga puede tener influencia sobre el partido ya que puede estar más cansado. Se diferencia del día anterior porque pensamos que la diferencia de significancia puede ser notable.
* visitor\_played\_prorrogue\_two\_days\_ago: se calcula si el visitante ha jugado el dos días anteriores prórroga. Se calcula esta variable porque pensamos que el que el visitante juegue dos días anteriores prórroga puede tener influencia sobre el partido ya que puede estar más cansado. Se diferencia del día anterior porque pensamos que la diferencia de significancia puede ser notable.
* local\_played\_prorrogue\_three\_days\_ago: se calcula si el local ha jugado tres días anteriores prórroga. Se calcula esta variable porque pensamos que el que el local juegue tres días anteriores prórroga puede tener influencia sobre el partido ya que puede estar más cansado. Se diferencia del día anterior y dos días anteriores porque pensamos que la diferencia de significancia puede ser notable.
* visitor\_played\_prorrogue\_three\_days\_ago: se calcula si el visitante ha jugado tres días anteriores prórroga. Se calcula esta variable porque pensamos que el que el visitante juegue tres días anteriores prórroga puede tener influencia sobre el partido ya que puede estar más cansado. Se diferencia del día anterior y dos días anteriores porque pensamos que la diferencia de significancia puede ser notable.
* local\_played\_local: se calcula si el local jugó como local su último partido. Se piensa que jugar varios partidos como local seguidos puede tener influencia en el resultado del partido actual.
* local\_played\_visitor: se calcula si el local jugó como visitante su último partido. Se piensa que jugar varios partidos como local seguidos puede tener influencia en el resultado del partido actual.
* visitor\_played\_visitor: se calcula si el visitante jugó como local su último partido. Se piensa que jugar varios partidos como local seguidos puede tener influencia en el resultado del partido actual.
* visitor\_played\_local: se calcula si el visitante jugó como local su último partido. Se piensa que jugar varios partidos como local seguidos puede tener influencia en el resultado del partido actual.
* place\_local\_played: se calcula donde jugó el local el último partido. Se calcula esta variable porque nos ayudará a a calcular la distancia que ha recorrido el equipo entre partidos.
* place\_visitor\_played: se calcula donde jugó el visitante el último partido. Se calcula esta variable porque nos ayudará a calcular la distancia que ha recorrido el equipo entre partidos.
* avg\_distance\_place\_visitor\_played: se calcula la distancia que ha recorrido el visitante respecto al último partido jugado por él entre los días que han pasado desde su último partido.
* avg\_distance\_place\_local\_played: se calcula la distancia que ha recorrido el local respecto al último partido jugado por él entre los días que han pasado desde su último partido.

1. Cálculos variables: para calcular las variables se recorre el dataframe stats\_equipos. Para realizar el cálculo de estas variables se realizan los siguientes pasos:
2. Primero, se calculan los siguientes dataframes sobre los que se aplicarán diferentes cálculos:
   * Jugo\_dia\_anterior: se guardan todos los registros que coinciden con la variable previous\_date del registro en curso. De forma análoga se crean dos dataframes que son jugo\_dos\_dias\_antes y jugo\_tres\_dias\_antes pero con las variables two\_days\_ago y three\_days\_ago
   * Se calculan los dataframes visitor\_jugo\_dia\_anterior y local\_jugo\_dia\_anterior. Estos dataframes se calculan cogiendo el dataframe jugo\_dia\_anterior y viendo si coincide el visitante o el local están en la variable local\_team o visitor\_team con el local\_team o el visitor\_team del registro del dataframe stats\_equipo. De forma análoga se hace para los dataframes visitor\_jugo\_dos\_dias\_antes, local\_jugo\_dos\_dias\_antes, visitor\_jugo\_tres\_dias\_antes y local\_jugo\_tres\_dias\_antes.
3. Una vez calculados estos dataframes se realizan las siguientes comprobaciones:
   * + Si local\_jugo\_dia\_anterior no está vacío la variable local\_played\_previous\_date se pone a uno. Si, además, ese registro de local\_jugo\_dia\_anterior tiene local\_prorrogue a uno la variable local\_played\_prorrogue\_previous\_date se pone a uno.
     + Si visitor\_jugo\_dia\_anterior no está vacío la variable visitor\_played\_previous\_date se pone a uno. Si, además, ese registro de visitor\_jugo\_dia\_anterior tiene visitor\_prorrogue a uno la variable visitor\_played\_prorrogue\_previous\_date se pone a uno.
     + Para las variables local\_played\_two\_days\_ago, local\_played\_prorrogue\_two\_days\_ago, visitor\_played\_two\_days\_ago, visitor\_played\_prorrogue\_two\_days\_ago, visitor\_played\_three\_days\_ago, visitor\_played\_prorrogue\_three\_days\_ago local\_played\_three\_days\_ago y local\_played\_prorrogue\_three\_days\_ago, se trabaja de forma parecida al punto anterior pero con los dataframes visitor\_jugo\_dos\_dias\_antes, local\_jugo\_dos\_dias\_antes, visitor\_jugo\_tres\_dias\_antes y local\_jugo\_tres\_dias\_antes.
4. Posteriormente, para calcular el resto de variables se realizan los siguientes pasos:
   * Se calcula un dataframe diez\_dias\_previos con todos los registros de stats\_equipos que están entre la fecha anterior del partido y diez días antes. Así, acotamos en una determinada línea temporal el análisis y eliminamos casos extremos como principio de temporada, vacaciones o posibles parones donde han ocurrido acontecimientos hace más de diez días pero que consideramos que no influyen en el resultado.
   * Si el dataframe diez\_dias\_previos está vacío sus variables mantienen el valor por defecto. Si no está vacío se crean cuatro dataframes que son local\_jugo\_local, local\_jugo\_visitante, visitante\_jugo\_local y visitante\_jugo\_visitante. Estos dataframes se calculan a partir del dataframe diez\_dias\_previos comprobando si el local jugó como local o como visitante y comprobando si el visitante jugó su último partido como local o visitante.
   * Si local\_jugo\_local o local\_jugo\_visitante no están vacíos quiere decir que el local jugó los diez anteriores.
     + Si el registro con mayor fecha pertenece a local\_jugo\_local quiere decir que place\_local\_played es el campo del local, que avg\_distance\_between\_previous\_date es cero y además se resta el number\_date del registro actual menos la fecha del último partido que jugó como local para saber la diferencia de días con su último partido jugado.
       - Si, además, local\_played\_local es mayor que -1 se suma uno a la variable local\_played\_local del registro actual de stats\_equipo y se pone visitor\_played\_local a cero.
       - Si no es mayor que -1 local\_played\_local es uno y local\_played\_visitor cero.
     + Si el registro con mayor fecha pertenece a local\_jugo\_visitante quiere decir que place\_local\_played es el campo del local de su útlimo partido. Además se resta el number\_date del registro actual menos la fecha del último partido que jugó como visitante para saber la diferencia de días con su último partido jugado.
       - Si, además, local\_played\_visitor es mayor que -1 se suma uno a la variable local\_played\_visitor del registro actual de stats\_equipo y se pone local\_played\_local a cero.
       - Si no es mayor que -1 local\_played\_visitor es uno y local\_played\_visitor cero.
     + Si local\_jugo\_local no está vacío pero si está vacío local\_jugo\_visitante se hace un desarrollo parecido al anterior pero sin comparar si el local jugó como local o visitante su último partido ya que jugó como local.
     + Si local\_jugo\_visitante no está vacío pero si está vacío local\_jugo\_local se hace el mismo desarrollo que en el punto iii. Pero sin comparar si el local jugó como visitante o local su último partido ya que jugó como visitante.
     + Para las variables place\_visitor\_played, visitor\_dif\_between\_previous\_game, visitor\_played\_local y visitor\_played\_visitor se hace el mismo desarrollo que en los puntos iii, iv y v pero con los dataframes visitante\_jugo\_local y visitante\_jugo\_visitante.
5. Una vez realizado el bucle anterior nos queda calcular la variable avg\_distance\_place\_visitor\_played y la variable avg\_distance\_place\_local\_played cuando el local no jugó su último partido como local. Para ello, se realizan los siguientes pasos:
   * + Se cambia los nombres del dataframe distancias por Equipo A = local\_team y Equipo B = visitor\_team y se realiza un merge entre stats\_equipos y distancias por la clave local\_team y place\_local\_played. Con ello, ya sabemos la distancia que recorrió el equipo local respecto a su último encuentro que jugó en el caso de que lo hiciera como visitante.
     + Se cambia los nombres del dataframe distancias por Equipo A = visitor\_team y Equipo B = local\_team y se realiza un merge entre stats\_equipos y distancias por la clave visitor\_team y place\_visitor\_played. Con ello, ya sabemos la distancia que recorrió el visitante respecto a su último partido.
     + Se calculan las variables avg\_distance\_place\_visitor\_place dividiendo la distancia recorrida respecto a su último partido entres visitor\_dif\_between\_previous\_date y avg\_distance\_place\_local\_place dividiendo la distancia recorrida respecto a su último partido entres local\_dif\_between\_previous\_date

Por último, se eliminan las columnas previous\_10\_days, previous\_date, two\_days\_ago, three\_days\_ago, visitor\_prorrogue y local\_prorrogue y se guarda el dataframe en un archivo csv llamado input\_extendido.

### VARIABLE VALORACIÓN EQUIPO

Con el objetivo de enriquecer nuestro modelo de predicción de resultados, con nuevas variables que puedan aportar información sobre la calidad de los equipos que se van a enfrentar en el mismo, se incluye a las ya existentes, una medida de valoración de los jugadores que van a participar en el encuentro, tanto en la posición de local como en la de visitante.

En la literatura existe una cantidad ingente de estudios al respecto, en la que profesionales de distintas disciplinas académicas (economistas, periodistas, ingenieros, psicólogos…) contribuyen a la creación de numerosas formas de valorar de manera más objetiva el desempeño de cada jugador.

Por lo que, teniendo en cuenta las características de nuestro modelo y los datos de los que disponemos, tras una revisión de varios sistemas de valoración -tanto los que se utilizan de manera oficial en las ligas profesionales, como aquellos empleados por webs y analistas especializados en este ámbito-, hemos seleccionado los dos que mejor podrían encajar con nuestro estudio y mayor respaldo o consistencia teórica muestran respecto a las críticas observadas en la literatura.

El tipo de datos del que se basan las medidas de valoración de jugadores, podría diferenciarse en dos principalmente. Información recogida de las estadísticas de cada partido, que se resumen en el “box-score”, o información más detallada basada en cada jugada que se produce en un encuentro, detallada en el “play by play”.

Teniendo en cuenta el objetivo de nuestro modelo de predicción, y las variables de las que disponemos para su estimación, la variable que queremos generar deberá estar basada en datos a nivel de encuentro, y por lo tanto nos serviremos de la información recogida en el  “box-score” para su cálculo.

Por otro lado, ya que el objetivo de esta medida de valoración será calcular el desempeño de cada uno de los jugadores, independientemente de la posición de estos, esta deberá basarse en una evaluación combinada, es decir, teniendo en cuenta cuestiones del juego tanto ofensivas como defensivas.

La primera de las medidas que se han seleccionado sería el Alternate Win Score (AWS) creada por Daniel Lewin y Dan Rosenbaum como alternativa al Win Score de David Berri, con pequeñas modificaciones en los pesos atribuidos a algunas de las métricas.

AWS= Pts + BR +BP + 0,5 (As + TF – FPC) + 0,7(RO – CCI) + 0,3(RD – CCC) – 0,35C1I – 0,15C1C

La otra medida seleccionada sería el Wins Produced (WP) detallada en el libro Wages of Wins and Stumbling on Wins por David Berri, Martin Schmidt y Stacey Brook. Se trata de un  índice que relaciona las victorias, y la eficiencia ofensiva y defensiva de los equipos. Se construye a través de un modelo de regresión previo, que determina el peso de cada una de las variables, en función a su influencia sobre la victoria de los partidos.

Esta medida se construye en varios pasos. Primeramente, y una vez calculados los pesos por los que se va a ponderar cada una de las variables, se obtiene una primera medida a nivel de jugador con las variables del box-score, tanto del jugador en concreto como del equipo oponente. Posteriormente se realiza un ajuste a la medida, añadiendo variables a nivel de equipo. Y por último, un ajuste extra con variables calculadas a nivel de liga.

Teniendo en cuenta la necesidad concreta de este estudio, y el tipo de modelo que queremos estimar, la medida que nosotros vamos a calcular siguiendo este sistema, no contemplará el último ajuste a nivel de liga, ya que las variables de nuestro modelo se van  construyendo de acuerdo a los partidos que han tenido lugar desde el comienzo de la temporada hasta el partido en cuestión.

Algunas de las premisas que nos han llevado a seleccionar estos sistemas de valoración para la creación de una nueva variable para nuestro modelo, es que aunque está basado en ponderaciones lineales como otras técnicas que hemos valorado, Win Score, Alternate Win Score o Wins Produced, premia a los jugadores a partir de un cierto rango de

eficiencia. En este sentido, se han realizado críticas hacia los sistemas de valoración Eficiencia y  PER, por premiar la “ineficiencia” de los jugadores, ya que con unos porcentajes de acierto bajos pueden obtener valores altos en esos índices.

Estos sistemas  utilizan  un  criterio  basado  en  la conversión de la producción del jugador a  las victorias del equipo. Wins Produced  y Win Score, ambas creadas por David Berri, tienen una base econométrica. Sin embargo, la segunda de estas, recibió críticas sobre el planteamiento de la función de producción del jugador, dando con la creación del Alternate Win Score por parte de Lewin y Rosenbaum, quienes teniendo en cuenta que no todos los lanzamientos fallados son pérdidas de posesión, modifican la ponderación del valor de esos lanzamientos, y también de los rebotes ofensivos y defensivos, de manera que se diferencie entre estas dos opciones de posesión.

Para la creación de nuestra medida de valoración conjunta de los jugadores, se han calculado dos medidas de acuerdo a los dos sistemas comentados anteriormente. Siguiendo con la idea de valorar cómo están desarrollando la temporada los jugadores, la medida de valoración consistirá en la media -para cada uno de los jugadores- del desempeño que han realizado en cada encuentro en el que han participado. Posteriormente, para cada encuentro en cuestión, esta medida constará de la suma, de esas medias obtenidas para cada uno de los jugadores que va a participar en él.

Por lo tanto, el primer paso sería valorar el desempeño de cada uno de los jugadores en cada encuentro en el que han participado. En el caso de la variable AWS, esta valoración se obtendría aplicando la fórmula creada por lo autores y comentada anteriormente, con las estadísticas generadas por cada uno de los jugadores en cada partido.

De esta manera, teniendo el resumen al final del partido, de las variables que intervienen en el cálculo, podemos obtener una valoración del desempeño de cada jugador en el mismo.

En el caso del sistema de valoración WP, primero debemos hacer una regresión de las variables del box-score contra el porcentaje de victorias del equipo. Para ello disponemos de un dataframe donde tenemos los resultados de todas estas medidas para cada uno de los encuentros de los últimos años. De esta manera, obtendremos la influencia de cada una de esas variables sobre la victoria.

Posteriormente, aplicaremos la fórmula del WP, ponderando cada una de las variables que la componen por la “influencia” obtenida en dicha regresión.

Al igual que en el caso del AWS, esta medida se calcula para cada jugador en cada uno de los encuentros en los que ha participado, para poder valorar su desempeño en cada uno de ellos.

Para poder introducir estas dos variables en el modelo como medidas de valoración de la calidad de juego que están desempeñando los jugadores en cada momento de la temporada, y poder medir con ello el nivel de los equipos que se enfrentan en cada encuentro, -como se comentó anteriormente- una vez obtenido el valor de ambas para cada uno de los jugadores en cada encuentro en el que participó, se realiza una media con los partidos jugados desde el comienzo de la temporada hasta el partido en cuestión, y se suman esos valores obtenidos, de todos los jugadores que participan en el encuentro.

# MODELO 1

# MODELO 2

# CONCLUSIONES RESPUESTAS A LAS HIPOTESIS

# REFERENCIAS BIBLIOGRÁFICAS / WEBGRAFÍA

Renato Amorim Torres. ***Prediction of NBA games based on Machine Learning Methods.***(2013) <http://homepages.cae.wisc.edu/~ece539/fall13/project/AmorimTorres_rpt.pdf>.

Matthew Beckler, Hongfei Wang. **NBA Oracle** (<http://www.mbeckler.org/coursework/2008-2009/10701_report.pdf> )

<https://towardsdatascience.com/predicting-2018-19-nbas-most-valuable-player-using-machine-learning-512e577032e3>

<https://www.basketball-reference.com/>

SERRANO y BAKER, Basketball Analytics, 2013, Advanced Metrics, Lake St. Louis

<https://www.basketball-reference.com/leagues/NBA_2014.html#all_team_stats>

<http://sportsmadeinusa.com/baloncesto/nba/ha-cambiado-el-modo-de-jugar-nba/>

García, J. Ibáñez, S. J., Feu, S. (2010). Estadísticas de juego que discriminan las selecciones participantes en los Campeonatos del Mundo de 2006 de Baloncesto, en función del nivel y género de los equipos. Revista Kronos, IX, 17, 57-63.

Sampaio, J., Fraga, F., Silva, T. (2004). La evolución de las estadísticas de los tiros libres en partidos de baloncesto de formación. III Congreso de la Asociación Española de Ciencias del Deporte. Valencia, 11-13 de marzo.

Sampaio, J., Leite, N. (2006). ¿Por qué ganaron o perdieron los partidos de baloncesto los equipos que participaron en el Eurobasket 2005? Revista Kronos, nº9, pp.67-73.

Sampaio, J., Lorenzo, A. Gómez Ruano, M. A., Matalarranha, J., Ibáñez, S. I., Ortega, E. (2009). Análisis de las estadísticas discriminantes en jugadores de baloncesto según su puesto específico,