Escola Tècnica Superior d'Enginyeria ETSE-UV

MÀSTER UNIVERSITARI EN CIÈNCIA DE DADES

TREBALL DE FI DE MÀSTER

MÉTODOS DE MACHINE LEARNING APLICADOS AL BALONCESTO

AUTOR:
ANTONIO SEGOVIA GARCÍA

TUTORS:
JORDI MUÑOZ MARÍ
VALERO LAPARRA PÉREZ-MUELAS

DICIEMBRE, 2021

VNIVERSITAT DE VALÈNCIA

Escola Tècnica Superior d'Enginyeria ETSE-UV

MÀSTER UNIVERSITARI EN CIÈNCIA DE DADES

TREBALL DE FI DE MÀSTER

MÉTODOS DE MACHINE LEARNING APLICADOS AL BALONCESTO

AUTOR:
ANTONIO SEGOVIA GARCÍA

TUTORS:
JORDI MUÑOZ MARÍ
VALERO LAPARRA PÉREZ-MUELAS

TRIBUNAL:

PRESIDENT:

VOCAL 1:

VOCAL 2:

DATA DE DEFENSA:

QUALIFICACIÓ:

Presidente del Tribunal Evaluador / *President del tribunal avaluador* via **CCA del Máster** / *CCA del Màster*

Resumen

El principal objetivo de este trabajo es encontrar un modelo de aprendizaje automático capaz de predecir el ganador de un partido de la NBA, basándose únicamente en estadísticas de partidos anteriores. Para ello, se ha obtenido, limpiado y procesado un compendio de estadísticas básicas de partidos de temporada regular entre los cursos 2010-2011 y 2019-2020. A ellas se le añadirán un conjunto de métricas avanzadas, obtenidas mediante procesos de webscraping, así como el Rating ELO. Para establecer una *baseline*, una marca a mejorar, se hará un primer ajuste a varios modelos de aprendizaje automático, tanto los más populares en el ámbito de estudio como la regresión logística o el Random Forest como algoritmos menos utilizados como el k vecinos más próximos o el Extreme Gradient Boosting. Posteriormente, se llevará a cabo un profundo análisis exploratorio, dando importancia no solo a los datos sino a su contexto para, mediante criterios humanos y matemáticos, crear varios conjuntos de datos para tratar de mejorar la marca establecida previamente. Como criterio matemático, se utilizará el algoritmo *Joint Mutual Information Maximization* para la selección de características ya que, tras una revisión bibliográfica, se ha determinado que funciona generalmente mejor que otros métodos en problemas de clasificación. Después de ajustar los datos a los modelos, elaborando un compendio de estadísticas medias de entre 1 y 10 partidos anteriores al encuentro a disputar, se ajustarán todos los conjuntos a los modelos, para acabar obteniendo, mediante un Light Gradient Boosting, un 68.41 % de acierto. Finalmente, se evaluará la robustez de los resultados obtenidos, contemplando métricas más allá de la *accuracy* y observando los estadísticos de un experimento con réplicas bootstrap. Se determinará que, además de haber obtenido resultados robustos, el modelo con mayor robustez para la partición realizada es la máquina de vectores soporte, a pesar de no existir diferencias de rendimiento estadísticamente significativas con otros modelos como el propio Light Gradient Boosting o la regresión logística.

Además, el trabajo cuenta con un experimento en el que se prueban las capacidades de la inteligencia artificial para sacar rédito de las apuestas deportivas, obteniendo un retorno medio sobre la inversión del 42.54 % en escasas diez semanas.

Palabras clave: Aprendizaje automático; NBA; Predicción de resultados; Rating ELO; Algoritmo JMIM.

Índice general

Capítulo 1

Introducción

El *Machine Learning* o aprendizaje automático es uno de los campos cuya popularidad y demanda de profesionales están creciendo a mayor velocidad en los últimos años. En una época en la que la cantidad de datos generados y recolectados crece exponencialmente, aquellas empresas que no basen su toma de decisiones en procesos conducidos por los datos están condenadas a una obsolescencia acelerada e inevitable. Empresas de sectores como banca, seguros, consultoría, farmacia, y un largo etcétera ya cuentan con departamentos enteros de ingeniería, tratamiento y modelizado de datos para optimizar sus decisiones de negocio.

En el sector deportivo también se está llevando a cabo esta transformación tecnológica, aunque más lentamente. A pesar de que solamente algunos de los equipos punteros de los deportes más populares cuenten con departamentos de análisis de datos, los medios ya se están empezando a hacer eco de esta tendencia. Conocidos son los casos del grupo de astrofísicos contratados para formar parte del equipo del famoso entrenador de fútbol Pep Guardiola [13], el exitoso futbolista Kevin de Bruyne, que negoció su última renovación de contrato sin agente, mediante datos e inteligencia artificial [17] y Carolina Marín, mejor jugadora española de bádminton de la historia, que utiliza los datos para optimizar sus entrenamientos y analizar a sus contrincantes [6].

En lo que respecta a este trabajo, que tratará de encontrar un modelo de predicción del ganador en un evento deportivo, existen muchos ámbitos en los que puede resultar útil este estudio. El más obvio, las apuestas deportivas. Más o menos polémicas, es una realidad que la popularidad de estas aumenta vertiginosamente. En este campo, tratar de predecir el ganador de un partido es de vital importancia, tanto para la correcta colocación de las cuotas por parte de las casas de apuestas, como para los jugadores que intentan derrotar a la casa de apuestas.

Dejando esto a un lado, saber quién es el favorito para ganar un partido (en base a estadísticas y datos, no a corazonadas) también resulta muy importante dentro de las propias ligas. Periodistas, organizaciones y obviamente, los equipos participantes, se pueden aprovechar de esta información y de su análisis exhaustivo para tratar de hacer los ajustes que los expertos consideren necesarios previo al partido en cuestión. Es por ello que cada vez más gente, tanto estudiantes como profesionales, han investigado sobre este ámbito, realizando publicaciones desde distintos enfoques que serán comentadas en el capítulo 2.

A pesar de haber remarcado la lentitud de la transición tecnológica en el mundo deportivo, el caso de los deportes norteamericanos es muy distinto. En ellos, sobre todo en los más populares (béisbol, fútbol americano y baloncesto), se lleva bastante más tiempo utilizando tanto los datos como la estadística avanzada para analizar el rendimiento de los equipos y las decisiones de negocio de las franquicias. En este trabajo, nos centraremos en la liga de baloncesto por excelencia, la NBA.

La NBA (*National Basketball Association*), es la tercera liga deportiva que más recaudación genera (7700 millones € en la temporada 2018-2019), solo por detrás de la MLB (*Major League Baseball*) y la NFL (*National Football League*). En ella, 30 equipos se disputan el trofeo Larry O'Brien, otorgado al campeón. Los equipos están divididos en dos conferencias, este y oeste, que a su vez están subdivididas en tres divisiones cada una. Durante la temporada regular, cada equipo disputa 82 partidos[1], 41 como local y 41 como visitante, estructurados de la siguiente manera:

- 4 partidos contra los 4 equipos restandes de la misma división.
- Entre 3 y 4 partidos contra los 10 equipos de la misma conferencia, pero de distinta división.
- 2 partidos contra los 15 equipos de distinta conferencia.

Una vez acabada la temporada regular comienzan los Playoffs, donde los 8 mejores equipos de cada conferencia se enfrentan en eliminatorias al mejor de 7 partidos para decidir los campeones de cada conferencia, que se disputarán el campeonato de la NBA.

En este trabajo, se pretende encontrar un modelo de aprendizaje automático capaz de predecir el ganador del partido con un grado aceptable de fiabilidad en base a las estadísticas medias de los últimos partidos de los equipos

[1]En condiciones habituales, cada equipo disputa 82 partidos durante la temporada regular. No obstante, circunstancias excepcionales como el cierre patronal de la temporada 2011-2012 o la pandemia de la COVID-19 provocaron reestructuraciones del calendario. Así pues, en el curso 2011-2012 se disputaron 66 partidos por equipo durante la temporada regular, en 2020-2021 fueron 72 partidos, y en 2019-2020 cada equipo disputó entre 63 y 75 partidos.

participantes. Para ello contamos con datos de tipo *boxscore*[2] y estadísticas avanzadas de los partidos de temporada regular desde la temporada 2010-2011 hasta la 2019-2020. Se probarán diferentes métodos, tales como regresión logística, algoritmos basados en árboles de decisión, métodos ensemble, algoritmos de boosting y redes neuronales, buscando obtener una precisión cercana al 70 %. Los modelos se probarán en varios conjuntos de distintas características, empleando en algunos el algoritmo *Joint Mutual Information Maximization* para reducir la dimensionalidad. El objetivo de 70 % está marcado por la literatura existente en este campo, estando los mejores resultados de modelos similares ligeramente por encima de esta marca.

La estructura del trabajo es la siguiente: en la revisión de la literatura se contemplan los resultados de algunos de los autores que han investigado en este ámbito. Seguidamente, se exponen los datos, comentando sus fuentes de procedencia, para después realizar un primer ajuste a los modelos, consiguiendo así una marca a batir en las siguientes secciones. En el capítulo 5 se realiza un exhaustivo análisis exploratorio de los datos y se procede a seleccionar las características con las que se crearán nuevos conjuntos para tratar de mejorar los resultados. Posteriormente, se vuelven a ajustar los nuevos conjuntos a los modelos en busca de mejores resultados, que se analizarán en el capítulo 7, además de poner a prueba su robustez. Finalmente, se comentarán los resultados obtenidos en el estudio y se plantearán futuras vías de investigación. Además, este trabajo cuenta con tres anexos. En el primero se definen al detalle las estadísticas avanzadas utilizadas en el estudio. En el segundo se exponen los resultados completos de los modelos y en el tercero se presenta un experimento de testeo en tiempo real llevado a cabo previo a la realización del trabajo.

Para la elaboración de este ensayo se han utilizado mayormente los dos lenguajes más comunes en tareas de ciencia de datos: R y Python. Se ha hecho uso de R para las funciones de importación, extracción, manipulación y visualización de datos, así como para la selección e ingeniería de características, mientras que Python se ha empleado para la implementación, adecuación y evaluación de los modelos de aprendizaje automático. La redacción del texto ha sido realizada en LaTeXmediante la plataforma Overleaf.

[2]Los datos de tipo *boxscore* contienen una tabla estructurada con las estadísticas (de jugadores individuales o del equipo) que resumen los eventos ocurridos durante el transcurso del partido.

Capítulo 2

Revisión de la literatura

Como hemos mencionado anteriormente, la creciente popularidad del aprendizaje automático, la combinación de deportes e inteligencia artificial y la reciente transición tecnológica han provocado un aumento de los estudios sobre el tema a tratar. En ellos, cada autor le da su enfoque particular, bien sea utilizando datos distintos, modificando las variables originales o variando los modelos a ajustar, a veces incluso creando sus propios algoritmos.

Comenzando por aquellos trabajos en los que se han utilizado datos distintos, Cheng *et al*. (2016) [5] utilizaron datos correspondientes a los partidos de Playoffs entre las temporadas 2007-2008 y 2014-2015. Estos partidos se caracterizan por ser algo más fácilmente predecibles que los de temporada regular, ya que en Playoffs, los equipos no dan descanso a ningún jugador importante, incluso los jugadores lesionados fuerzan para recuperarse anticipadamente, llegando a jugar infiltrados o sin haberse recuperado completamente. Todos los partidos de Playoffs son importantes, cosa que no ocurre en temporada regular. Durante una temporada tan larga, se dan casos de equipos jugando sin depositar el máximo esfuerzo en el partido o jugadores importantes que descansan para llegar más frescos a partidos más importantes o al final de la temporada. Esto provoca que los partidos de temporada regular sean más difíciles de predecir que los de Playoffs, en los que se dan menos sorpresas. Volviendo al trabajo que estábamos comentando, Cheng *et al*. (2016) [5] se basaron en el principio de máxima entropía para desarrollar su propio modelo, al que llamaron *NBA Maximum Entropy*. Con él, consiguieron lograr un notable 74.4 % de acierto, mejor que cualquiera de los modelos que utilizan datos de temporada regular.

De todos modos, este no es el mejor resultado obtenido a día de hoy. Hasta un sobresaliente 83 % de acierto se fueron Thabtah *et al*. (2019) [19]. Ellos utilizaron datos de finales de la NBA entre los años 1980 y 2017. Utilizando datos sólo de finales (que duran, como máximo, 7 partidos), es esperable que el conjunto de datos que acabasen utilizando no fuera muy grande. Con-

cretamente, fueron 430 observaciones de 22 variables con las que, mediante el uso de una red neuronal artificial, consiguieron esta marca.

En cuanto a los trabajos que utilizan datos de temporada regular, el mejor resultado lo obtuvo Morate (2016) [15]. Él dotó a su estudio de un enfoque particular al utilizar variables que normalmente no se tenían en cuenta, como lesiones, rachas de los equipos participantes entre sí y las cuotas de apuestas. Utilizando los datos de la temporada regular 2015-2016 y mediante un Random Forest, obtuvo un 74.13 % de acierto que, si bien ha podido resultar algo inflado por el hecho de haber utilizado solamente una temporada, es el mejor resultado entre los trabajos de este tipo.

Cao (2012) [4] realizó un estudio muy teórico, en el que entró muy al detalle sobre la minería de datos en general, en el ámbito deportivo y sobre la teoría de los modelos aplicados. Utilizando los datos de temporadas regulares entre 2006 y 2010 para entrenar una regresión logística, obtuvo un 69.67 % de acierto en los partidos de la temporada 2011, que reservó como conjunto de test.

Combinando predictores convencionales y algunos más novedosos (rachas de los equipos participantes, cuotas de casas de apuestas, similitud entre jugadores), y utilizando partidos de temporada regular entre los meses de enero de 2014 y enero de 2018, Lieder (2018) [12] obtuvo un acierto ligeramente por debajo del 69 % empleando una regresión logística.

Amorim (2013) [1] realiza un análisis de componentes principales para reducir la dimensionalidad de sus datos. El conjunto que utiliza consiste en los partidos de temporada regular entre los cursos 2006-2007 y 2012-2013. Predice sobre temporadas enteras, utilizando todas las observaciones anteriores para entrenar los modelos. Para evaluarlos, calcula la media del acierto obtenido en todas las temporadas. Así, consigue un 68.44 % de acierto utilizando un perceptrón multicapa.

Weiner *et al*. (2021) son los únicos que calcularon el Rating ELO, empleado también en este trabajo. El modelo que mejor les funcionó fue el Random Forest, con el que obtuvieron un 67.15 % de acierto, utilizando las temporadas regulares desde 2008-2009 hasta 2019-2020 como conjunto de datos.

Por debajo del 67 % se quedaron Miljkovic *et al*. (2010) [14], que utilizaron partidos de una sola temporada regular para entrenar y evaluar sus modelos. Concretamente, emplearon un total de 778 partidos de la temporada 2009-2010.

Los resultados más bajos los obtuvieron Villar (2019) [20] y Jones (2016) [10], con un 63.19 % y 62 % de acierto respectivamente. Villar (2019) [20] empleó una de las fuentes de datos utilizadas en este trabajo, el paquete `nbastatR` [3], mediante el cual obtuvo las estadísticas de equipos y de jugadores individuales. Por otra parte, Jones (2016) [10] utilizó muestras estratificadas de

144 partidos de cada equipo seleccionados aleatoriamente entre las temporadas 2008-2009 y 2010-2011. El modelo que mejor les funcionó a ambos fue la regresión logística.

Esta es toda la literatura existente que ha sido revisada previo a la realización de este trabajo. Observamos los resultados resumidos en la Tabla 2.1. Cabe destacar, que contemplamos el modelo de Cao (2012) [4] como resultado a batir, ya que es la mejor marca entre aquellas de trabajos que comparten características generales con este estudio, utilizar datos de temporada regular y emplear más de una temporada.

Tabla 2.1: Resumen de resultados de la literatura existente.

Autor	Acierto	Observaciones
Thabtah *et al*. (2019) [19]	83 %	Datos de finales NBA entre 1980 y 2017 Red Neuronal Artificial 430 observaciones de 22 variables
Cheng *et al*. (2016) [5]	74.4 %	Resultados notables y enfoque novedoso Utilizando el principio de máxima entropía Datos de Playoffs entre 2007-2008 y 2014-2015
Morate (2016) [15]	74.13 %	Solo utiliza temporada regular 2015-2016 Random Forest Datos distintos: lesiones, rachas, cuotas apuestas
Cao (2012) [4]	69.67 %	Datos de temporada regular 2006-2010 train, 2011 test Regresión logística
Lieder (2018) [12]	<69 %	Datos de enero 2014 a enero 2018 Combina predictores convencionales y novedosos Regresión logística
Amorim (2013) [1]	68.44 %	Datos desde 2006-2007 hasta 2012-2013 PCA + Perceptrón multicapa Test sobre una temporada, entrena con anteriores
Weiner *et al*. (2021) [9]	67.15 %	Temporada regular, 2008-2009 hasta 2019-2020 Random Forest Utiliza ELO
Miljkovic *et al*. (2010) [14]	<67 %	Datos de temporada regular 778 partidos de la temporada 2009-2010
Villar (2019) [20]	63.19 %	Datos del paquete nbastatR Estadísticas de equipos y de jugadores individuales Regresión logística
Jones (2016) [10]	62 %	Datos de temporadas entre 2008-2009 y 2010-2011 Muestras estratificadas Regresión logística

Fuente: elaboración propia

Capítulo 3

Los datos

3.1. Fuentes de datos

Para este trabajo vamos a utilizar los registros de todos los partidos de temporada regular entre las temporadas 2010-2011 y 2019-2020. Estos datos provienen de dos fuentes principales. La primera, es el paquete `nbastatR` [3]. Gracias a él, podemos disponer de los datos de tipo *boxscore* para todos los partidos desde la temporada 1946-1947, entre otras muchas funcionalidades. Además, haremos uso de los datos de estadísticas avanzadas para los partidos mencionados, que han sido obtenidos desde la página web oficial de estadísticas de la NBA [18] mediante procesos de webscraping gracias a herramientas como `RSelenium` [8]. Una vez llevados a cabo los procesos pertinentes de manipulación de datos, que serán detallados en el capítulo 5, acabaremos con las siguientes variables.

3.2. Las variables

Tendremos un total de 75 variables, divididas entre estadísticas básicas, estadísticas avanzadas y Rating ELO.

3.2.1. Estadísticas básicas

Son las obtenidas mediante el paquete `nbastatR`. Un conjunto de variables informativas y estadísticas que resumen los eventos ocurridos durante el transcurso del partido. Están agrupadas por equipo y por partido.

Tabla 3.1: Estadísticas básicas.

Nombre de la variable	Descripción
yearSeason	Año de la temporada
slugSeason	Código de la temporada
slugLeague	Código de la liga (NBA)
typeSeason	Tipo de partido (Temp. Regular/Playoffs)
dateGame	Fecha del partido
idGame	Identificador del partido
numberGameTeamSeason	Número de partido en la presente temporada
nameTeam	Nombre del equipo
idTeam	Identificador del equipo
isB2B	Indica si el partido forma parte de un back to back
isB2BFirst	Indica si el partido es el primero de un back to back
isB2BSecond	Indica si el partido es el segundo de un back to back
locationGame	Lugar del partido (en casa/fuera)
slugMatchup	Código del enfrentamiento
slugTeam	Código del equipo
countDaysRestTeam	Días de descanso del equipo antes del partido
countDaysNextGameTeam	Días hasta el próximo partido
slugOpponent	Código del oponente
slugTeamWinner	Código del equipo ganador
slugTeamLoser	Código del equipo perdedor
outcomeGame	Resultado del partido
isWin	Indica si se ha ganado el partido
fgmTeam	Tiros totales anotados
fgaTeam	Tiros totales intentados
pctFGTeam	Porcentaje total de acierto en tiros
fg3mTeam	Triples anotados
fg3aTeam	Triples intentados
pctFG3Team	Porcentaje de acierto en triples
pctFTTeam	Porcentaje de acierto en tiros libres
fg2mTeam	Tiros de dos anotados
fg2aTeam	Tiros de dos intentados
pctFG2Team	Porcentaje de acierto en tiros de dos
minutesTeam	Minutos jugados
ftmTeam	Tiros libres anotados
ftaTeam	Tiros libres intentados
orebTeam	Rebotes en ataque
drebTeam	Rebotes en defensa
trebTeam	Rebotes totales
astTeam	Asistencias
stlTeam	Robos de balón
blkTeam	Tapones
tovTeam	Pérdidas de balón
pfTeam	Faltas personales
ptsTeam	Puntos totales anotados
plusminusTeam	Diferencial de puntos

Fuente: elaboración propia

3.2.2. Estadísticas avanzadas

Estos datos han sido obtenidos mediante herramientas de webscraping de la página web oficial de estadísticas de la NBA. Son un compendio de estadísticas avanzadas sobre los tiros, asistencias, rebotes y ritmo del partido. En el Apéndice A se encuentran explicaciones más detalladas sobre estas variables, así como las fórmulas para calcularlas.

Tabla 3.2: Estadíasticas avanzadas.

Nombre de la variable	Descripción
OFFRTG	Rating ofensivo
DEFRTG	Rating defensivo
NETRTG	Rating neto
AST %	Ratio de asistencias
AST/TO	Ratio asistencias sobre pérdidas
AST RATIO	Ratio de asistencias sobre 100 posesiones
OREB %	Proporción de rebotes ofensivos
DREB %	Proporción de rebotes defensivos
REB %	Proporción total de rebotes
TOV %	Proporción de posesiones acabadas en pérdida
EFG %	Porcentaje efectivo de tiros
TS %	Porcentaje verdadero de tiros
PACE	Ritmo
PIE	Impacto estimado de los jugadores
%FGA 2PT	Proporción de tiros de dos
%FGA 3PT	Proporción de tiros de tres
%PTS 2PT	Proporción de puntos de tiros de dos
%PTS 2PT MR	Proporción de puntos de tiros de dos, media distancia
%PTS 3PT	Proporción de puntos de tiros de tres
%PTS FBPS	Proporción de puntos en contraataque
%PTS FT	Proporción de puntos de tiros libres
%PTS OFF TO	Proporción de puntos después de pérdida rival
%PTS PITP	Proporción de puntos en la pintura
2FGM %AST	Tiros de dos tras asistencia
2FGM %UAST	Tiros de dos sin asistencia
3FGM %AST	Tiros de tres tras asistencia
3FGM %UAST	Tiros de tres sin asistencia
FGM %AST	Tiros totales tras asistencia
FGM %UAST	Tiros totales sin asistencia

Fuente: elaboración propia

Teniendo ya estas variables, solo falta añadir una más, que tendrá un rol muy importante, ya que es una medida del estado de forma de los equipos, el **Rating ELO**.

3.2.3. El Rating ELO

El rating ELO es un método estadístico para calcular la habilidad relativa de equipos o jugadores de deportes. Original del ajedrez, este método fue inventado por el profesor Árpád Élo e implementado por primera vez por la *United States Chess Federation* en el año 1960. Dado su buen funcionamiento, se empezó a adaptar a otras disciplinas, y en la actualidad, se utiliza tanto en deportes de intelecto como el ajedrez o el Go, como en deportes de contacto como el fútbol, el fútbol americano y el baloncesto.

En cuanto al tema que nos concierne, Weiner *et al* (2021) [9] fueron los primeros en incluir los cálculos del ELO como predictor en los modelos. Más adelante observaremos cómo esta variable individual tiene una sorprendente capacidad predictiva.

Cálculo del ELO en la NBA

En la NBA, el ELO está formulado para tener en cuenta el margen de victoria, los resultados inesperados, el lugar del encuentro e incluso posibles movimientos de jugadores (o incluso equipos) entre temporadas. Todos los equipos comienzan con un ELO de 1500, que se va actualizando partido a partido mediante la siguiente fórmula.

$$ELO_{i+1} = k \cdot \left(S_{equipo} - E_{equipo}\right) + ELO_i$$

S_{equipo} indica si el equipo ha ganado o ha perdido el partido. Por lo tanto, tomará un valor de 1 en las victorias y 0 en las derrotas.

E_{equipo} representa la probabilidad esperada de victoria del equipo, calculada en función de su ELO y el de su oponente, de la siguiente manera:

$$E_{equipo} = \frac{1}{1 + 10^{\frac{ELO_{oponente} - ELO_{equipo}}{400}}}$$

k es una constante que depende del margen de victoria y de la diferencia del ELO de los participantes. El primer factor para el cálculo de k varía dependiendo del deporte, para corregir por la estructura de la liga (en caso de haberla) y la cantidad de partidos jugados en una temporada. Fisher-Baum y Silver (2015) [7] demostraron que para la NBA, el valor más apropiado para este primer factor es de 20. Por tanto el cálculo de la constante k se actualiza mediante esta fórmula:

$$k = 20 \cdot \frac{(Margen\ victoria\ + 3)^{0,8}}{7{,}5 + 0{,}006\left(ELO_{ganador} - ELO_{perdedor}\right)}$$

Para actualizar el ELO entre temporadas teniendo en cuenta las posibles modificaciones de equipos durante el verano, se aplica esta fórmula [7].

$$ELO_{t+1} = (ELO_t \cdot 0{,}75) + (1505 \cdot 0{,}25)$$

En la Tabla 3.3 observamos los ELOs que corresponderían aproximadamente con diferentes récords en temporada regular. Más del 90 % de los equipos tienen un ELO entre 1300-1700, pero temporadas históricas (para bien y para mal), pueden caer fuera de este rango.

Tabla 3.3: ELOs y sus récords equivalentes aproximados.

ELO	Récord equivalente	Descripción
1800	67-15	Top 5 % de la historia
1700	60-22	Aspirante
1600	51-31	Playoffs
1500	41-41	Equipo medio
1400	31-51	Lotería del Draft
1300	22-60	Reconstrucción
1200	15-67	Históricamente malo

Fuente: elaboración propia a partir de FiveThirtyEight

Una vez calculado el Rating ELO para todos los partidos disponibles, ya tenemos todos los datos que vamos a utilizar en el trabajo.

Capítulo 4

Modelos Base

Utilizando los datos comentados en el capítulo 3, vamos a crear 3 conjuntos distintos para probar diferentes modelos. Los resultados que obtengamos los consideraremos como *baseline*, y marcarán el registro a mejorar en posteriores secciones. Los conjuntos de datos son un compendio de estadísticas medias de los últimos 10 partidos de los equipos participantes, así como el ELO de los equipos antes del partido correspondiente, y están confeccionados de la siguiente manera:

- **Conjunto 1**: contiene solo las estadísticas básicas comentadas en la sección 3.2.1. Tras realizar las transformaciones pertinentes que se comentarán más adelante[1], las dimensiones del conjunto son de 11714 datos (filas) y 61 características (columnas). Se dividen en conjuntos de entrenamiento y test, utilizando una proporción de división de 0.75, acabando con 8786 observaciones de train y 2928 de test.
- **Conjunto 2**: este conjunto contiene tanto las estadísticas básicas del anterior como las avanzadas, comentadas en la sección 3.2.2. Así pues, se trata de un conjunto bastante más grande, con el mismo número de observaciones pero con 119 características.
- **Conjunto 3**: en este, contamos con estadísticas básicas y avanzadas, pero se ha calculado un diferencial para todas las variables disponibles.

  $$\text{dif} = \text{estadística}_{\text{local}} - \text{estadística}_{\text{visitante}}$$

  De este modo, valores positivos supondrán una superioridad del equipo local en la estadística correspondiente, y valores negativos indicarán que el equipo visitante tiene valores mayores para esa variable. Este conjunto de datos contiene el mismo número de observaciones,

[1] Básicamente, transformar el conjunto para tener cada partido en una observación, con las estadísticas de equipos local y visitante por separado.

pero solo 60 variables, ya que la información se encuentra resumida. La partición en conjunto de entrenamiento y test es la misma para los 3 conjuntos de datos.

A estos, se les han aplicado los siguientes modelos:

- **Regresión logística:** similar a la regresión lineal, que trata de predecir los valores de una variable continua, la regresión logística trata de predecir sobre una variable categórica. Además, se suele utilizar para modelar la probabilidad de un evento en función de un grupo de variables independientes.
- **Regresión logística** (escogiendo el mejor modelo con el método *stepwise*): el método *stepwise* es un método iterativo para la inclusión de variables independientes en una regresión. Basándose en criterios estadísticos, incluye o descarta un predictor en cada iteración y compara los modelos ajustados para decidir qué hacer con dicho predictor. En este caso, la métrica de selección es el criterio de información de Akaike.
- **Random Forest**: se trata de un método ensemble, en el que se combinan árboles de decisión, donde para tomar las decisiones correspondientes en cada nodo de cada árbol se escoge un subconjunto aleatorio de variables menor que el conjunto original. Una de sus principales ventajas es que, a pesar de su simplicidad, los resultados no suelen estar muy lejos de los resultados de algoritmos más sofisticados. Este algoritmo se puede utilizar tanto para clasificación como para regresión.
- **Máquinas de vectores soporte**: este método trata de encontrar el hiperplano óptimo de separación entre clases, entendiendo por esto aquel plano que tenga la máxima distancia posible con los puntos más cercanos.
- **k vecinos más próximos**: en este algoritmo se clasifica las muestras calculando las distancias (se utilizará la distancia euclídea, aunque pueden ser otras) a sus k vecinos más próximos, dando como resultado la clase más frecuente entre estos.
- **AdaBoost**: también es un método ensemble, su nombre viene de *Adaptative Boosting*. El algoritmo combina varios clasificadores débiles (clasificadores que no suelen obtener resultados muy superiores al 50 %) mediante una ponderación con unos pesos que dependen del error de cada clasificador, para tratar que el conjunto de clasificadores de el mejor resultado posible.
- **Perceptrón multicapa**: es el tipo más simple de red neuronal. En él, los datos entran por la capa de entrada, y mediante distintos pesos

y funciones de activación, se van propagando por las capas ocultas del perceptrón. Finalmente, en la capa de salida se devuelven tantas salidas como neuronas tenga esta capa.

- **Light Gradient Boosting**: es un algoritmo basado en árboles de decisión. Inventado recientemente, presume de ser más rápido que el Extreme Gradient Boosting, y de funcionar mejor que algoritmos de boosting más antiguos como el AdaBoost. La principal diferencia respecto al Extreme Gradient Boosting reside en la forma de dividir los áboles. Mientras que este divide los árboles por niveles de profundidad, el Light Gradient Boosting escoge cada división hoja a hoja, basándose en su contribución al error global en vez de al error en cada rama, obteniendo así resultados que, si bien no siempre son mejores, sí que conllevan un coste de computación menor.
- **Extreme Gradient Boosting**: algoritmo algo más antiguo, su invención supuso casi una revolución entre los métodos de aprendizaje automático, dados sus buenos resultados con casi cualquier conjunto de datos.

En todos ellos se ha utilizado valización cruzada[2], se ha realizado un *grid-search* para el tuneado de los hiperparámetros y se ha escogido el mejor modelo en función del acierto en los conjuntos de validación. Los resultados obtenidos figuran en las Tablas 4.1, 4.2 y 4.3. Para comparación, también se incluyen un modelo que siempre prediga victoria del equipo local, y otro que siempre apueste por el equipo con un ELO mayor. Este último modelo, sorprendentemente obtiene una precisión del 65 % en todos los partidos, lo cual no está nada mal teniendo en cuenta que solo se hace uso de una única variable.

Observamos que, para los conjuntos 1 y 2, el modelo que mejor resultados obtiene es la máquina de vectores soporte, cuyo acierto en test es ligeramente superior en el conjunto con todas las estadísticas que en el que contiene únicamente las estadísticas básicas. En cuanto al conjunto 3, el modelo que mejor funciona es el Light Gradient Boosting, obteniendo unos resultados peores que los del conjunto 2, pero mejores que los del primer conjunto. Esta será nuestra marca a mejorar durante el trabajo.

[2]con 3 o 5 folds, dependiendo del tamaño del conjunto de datos

Tabla 4.1: Resultados sobre el conjunto 1.

Modelo	Precisión en validación	Precisión en test	Aciertos en test
Victoria local		58.58	
Mayor ELO		65.00	
Regresión logística	**67.42**	67.52	1977
Reg. log. (+ stepwise)	67.29	67.31	1971
Random Forest	66.70	65.98	1932
SVM	66.73	**67.68**	**1981**
kNN	65.93	66.77	1955
AdaBoost	66.63	67.01	1962
Perceptrón multicapa	66.63	67.55	1978
LightGBM	66.37	66.09	1935
ExtremeGB	66.51	66.57	1920

Fuente: elaboración propia

Tabla 4.2: Resultados sobre el conjunto 2.

Modelo	Precisión en validación	Precisión en test	Aciertos en test
Victoria local		58.58	
Mayor ELO		65.00	
Regresión logística	66.03	66.63	1951
Reg. log. (+ stepwise)	65.61	66.12	1936
Random Forest	66.25	64.96	1902
SVM	66.67	**67.93**	**1989**
kNN	**66.87**	66.80	1956
AdaBoost	66.38	67.21	1968
Perceptrón multicapa	66.64	66.36	1943
LightGBM	66.70	67.04	1963
ExtremeGB	65.57	65.77	1926

Fuente: elaboración propia

Tabla 4.3: Resultados sobre el conjunto 3.

Modelo	Precisión en validación	Precisión en test	Aciertos en test
Victoria local		58.58	
Mayor ELO		65.00	
Regresión logística	66.29	67.01	1962
Reg. log. (+ stepwise)	66.44	67.11	1965
Random Forest	66.21	64.22	1945
SVM	66.67	67.18	1967
kNN	66.25	66.84	1957
AdaBoost	66.48	67.62	1980
Perceptrón multicapa	**67.16**	67.59	1979
LightGBM	66.34	**67.83**	**1986**
ExtremeGB	66.75	67.14	1966

Fuente: elaboración propia

Una vez tenemos definida nuestra *baseline*, vamos a realizar los procesos de análisis exploratorio e ingeniería de características sobre el conjunto con estadísticas básicas y avanzadas, ya que es para el que mejor resultado hemos obtenido. Posteriormente, volveremos a realizar el ajuste de los modelos con los nuevos conjuntos que confeccionaremos, para ver si conseguimos mejorar los resultados.

Capítulo 5

Análisis exploratorio, selección e ingeniería de características

El objetivo de esta sección es confeccionar dos conjuntos de datos con los que seguir adelante en el trabajo. Recordemos que en la sección anterior creamos tres conjuntos distintos, cada uno con diferentes características, para realizar un primer ajuste de los modelos y observar los resultados. Estos no han sido sometidos a ningún tipo de análisis o tratamiento[1], con lo cual, puede existir redundancia, confusión, o variables superfluas que entorpezcan la capacidad predictiva de los modelos. Así pues, en este capítulo se llevará a cabo el análisis exploratorio de las variables para posteriormente confeccionar dos conjuntos de datos, con los que se estudiará la importancia de los predictores y posibles efectos de interacción entre ellos. Uno de los conjuntos se realizará de forma manual, tomando las decisiones de inclusión o eliminación de predictores de forma humana, mientras que el conjunto restante se obtendrá mediante un algoritmo automático de selección de características, concretamente, el algoritmo JMIM o *Joint Mutual Information Maximization*.

5.1. Análisis exploratorio y selección de características

5.1.1. Procesado de los datos

Una vez juntados los datos en un solo conjunto, lo primero que tenemos que hacer es modificar los nombres de algunos equipos que, entre los años 2010 y 2020, cambiaron de nombre o de ciudad. Este es el caso de, por ejemplo, los Charlotte Bobcats, que en 2014 volvieron a llamarse Charlotte Hornets (cuando ese nombre quedó libre, ya que los New Orleans Hornets pasaron a

[1] Más allá de las tareas de adecuación de los datos al formato correcto para poder entrenar los modelos.

ser New Orleans Pelicans) o los New Jersey Nets, que se mudaron a Nueva York, convirtiéndose en los Brooklyn Nets.

Tras realizar estas modificaciones pasamos a analizar los valores faltantes en el conjunto de datos. Observamos que la variable `countDaysNextGameTeam`, es decir, el número de días hasta el siguiente partido del equipo, tiene varios valores faltantes. Fijándonos en los registros, nos damos cuenta de que corresponden a los últimos partidos de las temporadas 2011-2012 y 2019-2020. Estas temporadas no fueron temporadas al uso. El curso 2011-2012 comenzó el día de Navidad, dos meses después de lo habitual, y los equipos disputaron 66 partidos cada uno en vez de los habituales 82. Durante el verano de 2011, la liga y la asociación de jugadores tuvieron varios desacuerdos sobre medidas acerca de los salarios de los deportistas y el límite salarial de los equipos, dando lugar al cuarto cierre patronal o *lockout* de la historia de la NBA. Las negociaciones se alargaron hasta noviembre de 2011, con la liga completamente parada. Durante el cierre, que comenzó el 1 de julio, las franquicias no podían mantener ningún contacto con los jugadores, ni para negociar contratos, ni para entrenar, estando prohibido que los jugadores utilizasen las instalaciones de los equipos para ejercitarse. Al acabar dicha temporada regular, no se sabía con seguridad si la siguiente daría comienzo en las fechas usuales o se llevaría a cabo otra reestructuración del calendario. Por eso, las últimas observaciones de esta variable correspondientes a la temporada 2011-2012 son `NA`s (*not available*).

En cuanto a la temporada 2019-2020, esta fue interrumpida forzosamente el 11 de marzo de 2020, después de que un jugador diese positivo en COVID-19. La suspensión se alargó más de 4 meses, hasta el 30 de julio, cuando se reanudó la liga en una sola sede, y donde solamente participaron aquellos equipos que tenían opciones de clasificarse para la postemporada. Así pues, no todos los equipos disputaron los mismos partidos en temporada regular. Además, la inusual duración de esta temporada provocó una reestructuración del calendario de la siguiente, que se llevó a cabo una vez finalizada la liga ese año. Por lo tanto, cuando la temporada regular llegó a su fin, no se sabía con certeza cuándo iba a ser disputado el próximo partido de cada equipo. Así pues, las observaciones de esta variable correspondientes a los últimos partidos de cada equipo en la temporada 2019-2020, también son `NA`s. Todos los valores faltantes en esta variable serán sustituidos por 0.

Además de los ya comentados, hay dos observaciones donde todas las variables numéricas son 0 o `NA`. Son las correspondientes a un partido del 16 de abril de 2013 entre los Boston Celtics y los Indiana Pacers. Era el penúltimo partido de la temporada regular 2012-2013 para los dos equipos, y ambos tenían sus puestos en Playoffs garantizados, con lo cual no había nada en juego. Un día antes, se había llevado a cabo la conocida maratón de Boston, en la cual se produjeron dos fuertes explosiones cercanas a la línea de meta.

A la luz de los acontecimientos y por la seguridad de sus habitantes, todos los eventos deportivos de la ciudad de Boston fueron suspendidos y, como ambos equipos no se jugaban nada, la NBA decidió no aplazar el partido a otra fecha, sino suspenderlo. Así pues, estas observaciones serán eliminadas. Una vez tenemos nuestros datos libres de valores faltantes, podemos proceder al análisis univariante. Iremos comentando las decisiones que tomamos sobre las variables del conjunto de datos confeccionado a mano a medida que realizamos el análisis.

5.1.2. Análisis univariante

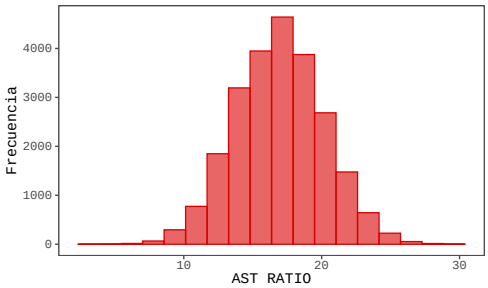
Antes de comenzar, cabe destacar que el conjunto cuenta con un grupo de variables informativas que no serán incluidas en los modelos, ya que se trata de nombres de equipos, identificadores y códigos. Así pues, las variables `yearSeason`, `slugSeason`, `slugLeague`, `typeSeason`, `dateGame`, `idGame`, `nameTeam`, `idTeam`, `slugMatchup`, `slugTeam`, `slugOpponent`, `slugTeamWinner`, `slugTeamLoser`, `outcomeGame` serán descartadas. A partir de los predictores `isWin` y `locationGame` crearemos nuestra variable respuesta, `H_Win`, que tomará valores de 1 si vence el equipo local (*Home*) y 0 si vence el equipo visitante.

En cuanto a la variable respuesta, observamos que el equipo local vence un 58.58 % de los partidos, destacando la importancia del factor cancha en la NBA. Dentro del resto de predictores encontramos únicamente distribuciones unimodales. Además, la mayoría de ellas son razonablemente simétricas, como la distribución de la ratio de asistencias por 100 posesiones (Figura 5.1), aunque también encontramos alguna distribución sesgada a la izquierda (por ejemplo, la distribución del porcentaje de acierto en tiros libres, Figura 5.2) y a la derecha (como es el caso de la proporción de puntos provenientes de contraataques, Figura 5.3).

En base al análisis univariante ya podemos tomar decisiones sobre el descarte de algunas variables. Encontramos predictores que son resultados directos de operaciones entre otros, lo cual puede estar añadiendo redundancia en los modelos. Así pues, de momento, descartaremos `trebTeam` (rebotes totales, suma de `orebTeam`, rebotes ofensivos, y `drebTeam`, rebotes defensivos), `NETRTG` (rating neto, resultado de la diferencia entre `OFFRTG`, rating ofensivo y `DEFRTG`, rating defensivo) y todas las variables del siguiente conjunto:

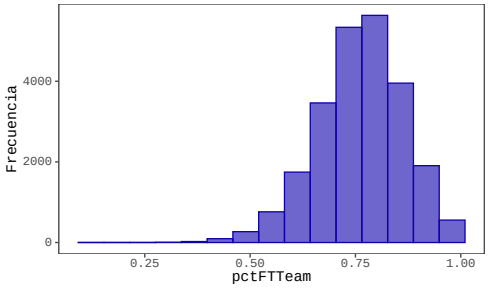
%FGA 2PT, 2FGM %UAST, 3FGM %UAST, FGM %UAST[2].

Figura 5.1: Distribución de la ratio de asistencias por 100 posesiones.



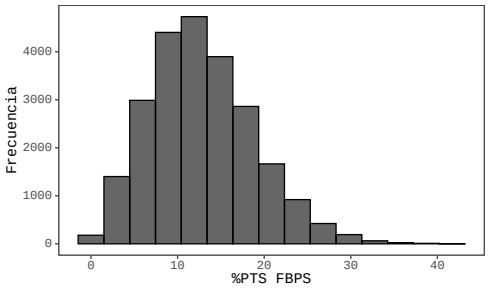
Fuente: elaboración propia

Figura 5.2: Distribución del porcentaje de acierto en tiros libres.



Fuente: elaboración propia

Figura 5.3: Distribución de la proporción de puntos provenientes de contraataques.



Fuente: elaboración propia

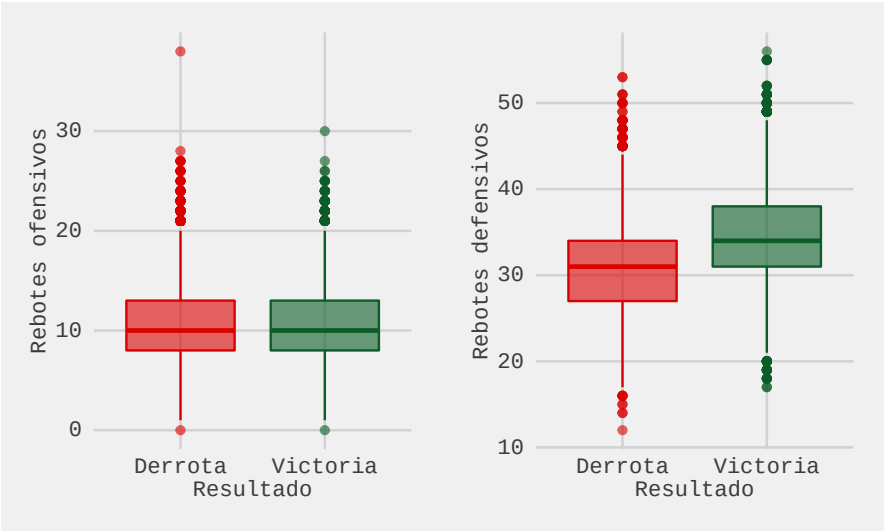
[2]Todas corresponden a una operación de tipo $100 - X$, concretamente:

- $\%FGA\ 2PT = 100 - \%FGA\ 3PT$
- $2FGM\,\%UAST = 100 - 2FGM\,\%AST$
- $3FGM\,\%UAST = 100 - 3FGM\,\%AST$
- $FGM\,\%UAST = 100 - FGM\,\%AST$

5.1.3. Análisis bivariante

Observando la distribución de cada variable frente a la variable respuesta podemos llegar a conclusiones evidentes, como que por norma general gana el equipo que más tiros mete (y con mejor porcentaje), más asistencias reparte, más rebotes captura o menos pérdidas o faltas comete, pero también podemos obtener información más interesante. Por ejemplo, en la Figura 5.4 observamos que, mientras la cantidad de rebotes defensivos capturados (y rebotes totales) sí que parece ser determinante, los rebotes ofensivos no parecen marcar tantas diferencias.

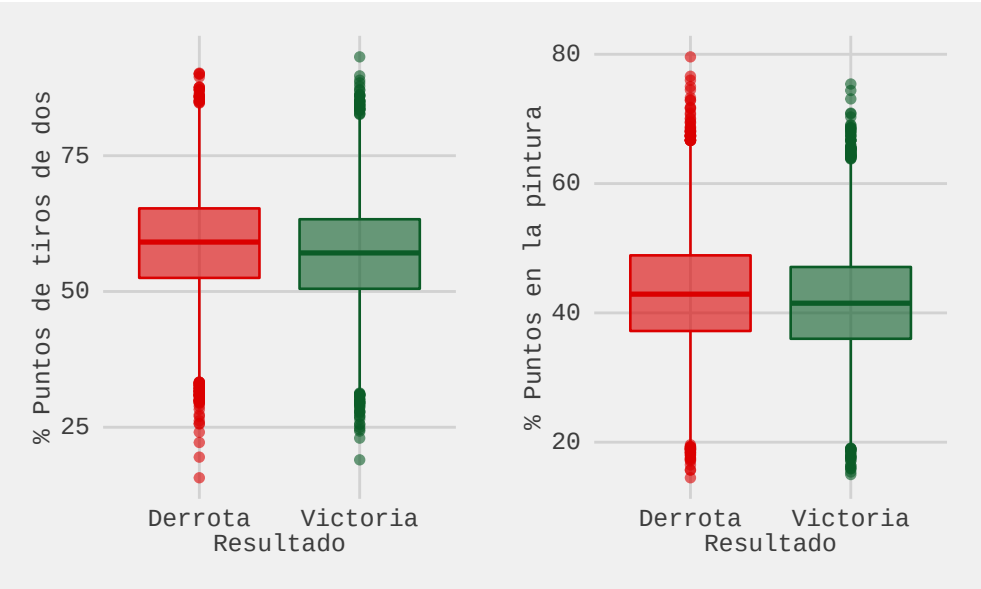
Figura 5.4: Distribución de rebotes ofensivos y defensivos frente al resultado.



Fuente: elaboración propia

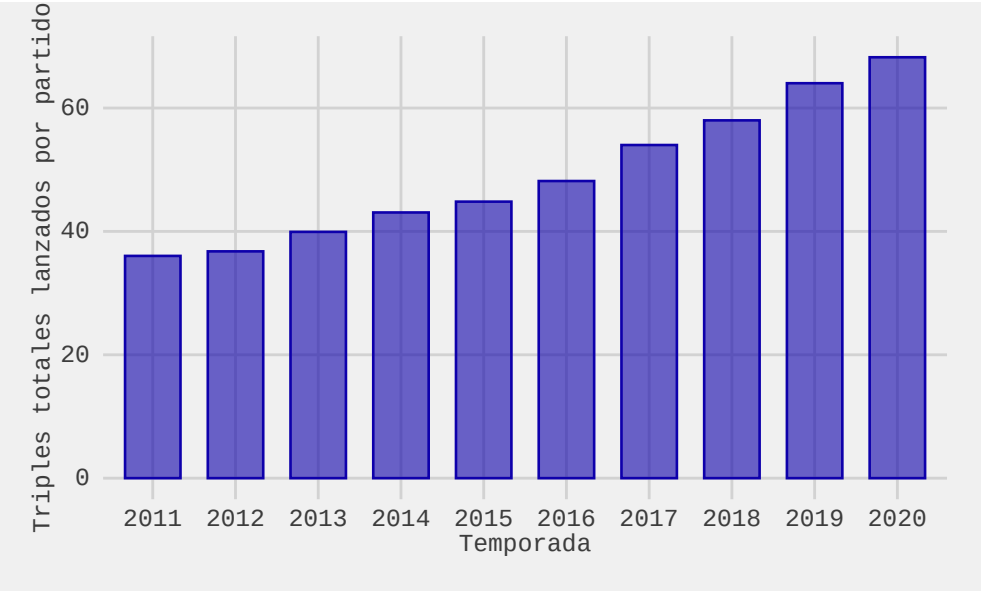
En la Figura 5.5 se aprecia como los equipos que centran su juego en los tiros de dos y en los puntos en la pintura son más proclives a perder el partido, lo que concuerda con la evolución del estilo de juego de los últimos años hacia una dinámica mucho más centrada en el lanzamiento de tiros de tres puntos (Figura 5.6). Esto se debe a lo que los expertos conocen como la eficiencia de los lanzamientos. Como es obvio, cuanto más lejos se produce un lanzamiento, más difícil es que entre, y todos los tiros desde dentro de la zona demarcada por la línea de triple valen lo mismo, dos puntos. En cambio, sí que existe una recompensa extra cuando se sobrepasa dicha línea, pasando a valer los tiros tres puntos. Es por esto por lo que se cree que no merece la pena asumir el riesgo de lanzar de media distancia, cuando estos lanzamientos tienen el mismo valor que los tiros debajo de canasta, lo que ha llevado a que la popularidad de los tiros de media distancia, tan utilizados antaño, haya caído notablemente, evolucionando hacia un juego centrado, sobre todo, en los tiros en la pintura y los triples.

Figura 5.5: Distribución de los puntos de tiros de dos y en la pintura frente al resultado.



Fuente: elaboración propia

Figura 5.6: Distribución de la cantidad de triples lanzados por partido y por temporada.



Fuente: elaboración propia

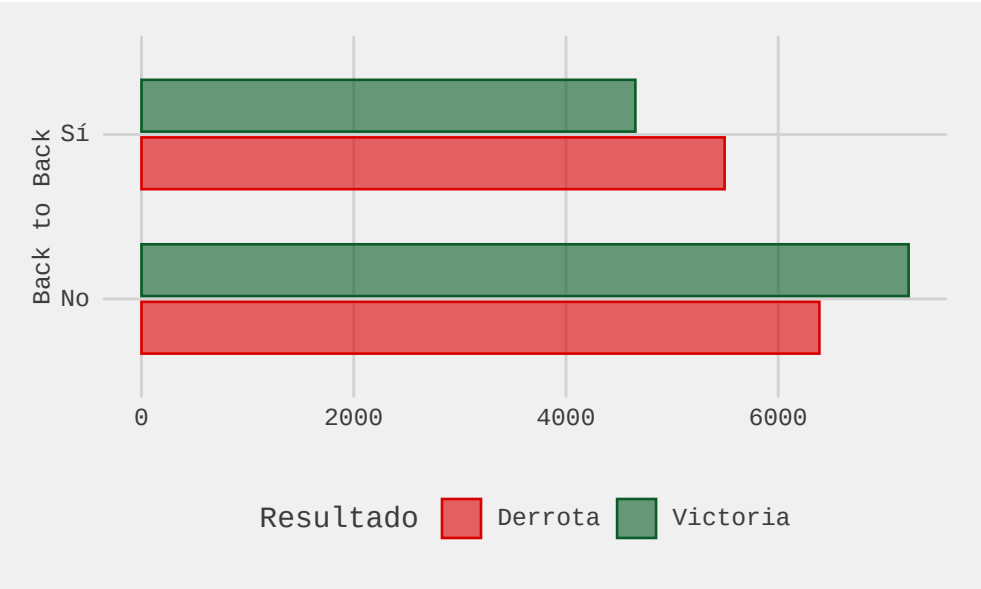
Respecto a las situaciones de *back to back*, o disputa de partidos en noches consecutivas, se aprecia claramente que los equipos en situación de *back to back* suelen tener un peor rendimiento, cosechando bastantes más derrotas que victorias. En cambio, los equipos que no están en esta situación suelen salir victoriosos más frecuentemente (Figura 5.7).

Si nos centramos en los equipos, observamos en la Figura 5.8 que son los San Antonio Spurs quienes más victorias han cosechado entre las temporadas 2010-2011 y 2019-2020. Campeones de la NBA en 2014, los tejanos han sido durante los últimos años un seguro en los Playoffs, mostrándose muy regulares durante la temporada. De hecho, el curso 2019-2020 fue el primero en el que no lograron clasificarse para Playoffs después de 22 años consecutivos. Por otro lado, los Minnesota Timberwolves, cuya última (y única durante el período analizado) aparición en Playoffs fue en 2018, y cuya última victoria en una serie de Playoffs se remonta a 2004, son el equipo que menos victorias ha conseguido durante la temporada regular entre los cursos 2010-2011 y 2019-2020.

De todos modos, el hecho de ser regularmente exitoso durante la temporada no garantiza tener mayor probabilidad de conseguir el campeonato de manera consistente durante unos años. Lo más normal es que durante rachas de 3 a 5 años haya equipos que consigan juntar un buen grupo de jugadores que, salvo sorpresa, les lleven a la pugna por el campeonato. Es el caso, por ejemplo, de los Miami Heat, que disputaron todas las finales entre las temporadas 2010-2011 y 2013-2014, venciendo en dos ocasiones. Ese verano, su estrella, LeBron James, quien acababa contrato, decidió volver a los Cleveland Cavaliers, equipo donde empezó su carrera, y al que llevó a las finales los siguientes 4 años consecutivos, logrando obtener un campeonato en 2016 contra los poderosos Golden State Warriors, quienes ganaron 3 de las 5 finales que disputaron de manera consecutiva entre las temporadas 2014-2015 y 2018-2019.

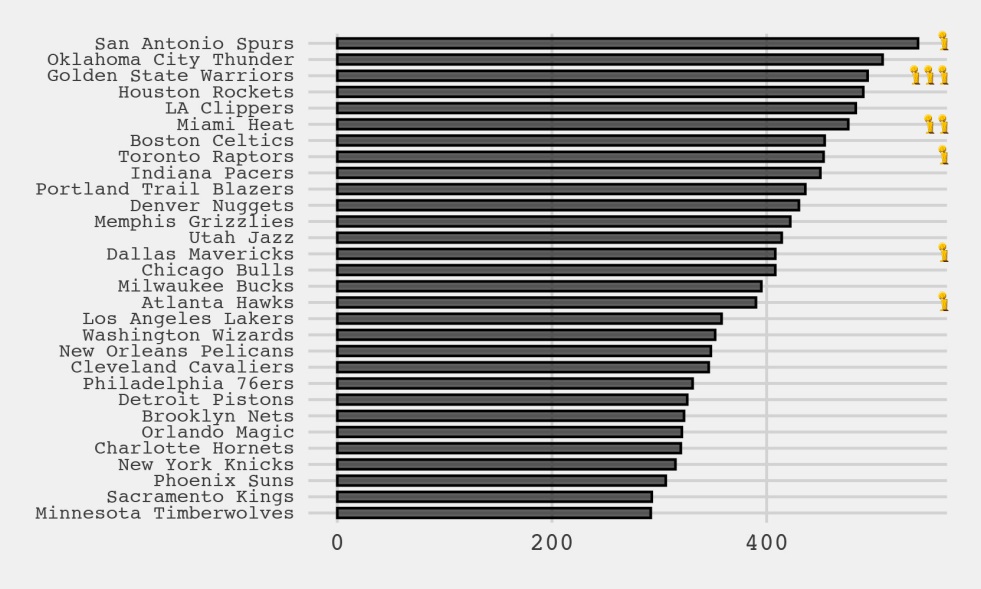
Así pues, aunque aún no hayamos tomado más decisiones acerca de las variables de nuestro conjunto de datos, hemos obtenido algo de información que hasta ahora no sabíamos. Vamos ahora a analizar la correlación lineal entre todos los pares de variables, para tratar de eliminar información duplicada.

Figura 5.7: Cambio en el rendimiento en situaciones de *back to back*.



Fuente: elaboración propia

Figura 5.8: Victorias en temporada regular y campeonatos entre 2011 y 2020.



Fuente: elaboración propia

5.1.4. Estudio de correlaciones

Para tratar de eliminar redundancias, vamos a calcular el coeficiente de correlación de Pearson de todos los pares de variables. Pondremos un umbral en $\pm 0{,}8$ y descartaremos variables para tratar que la correlación entre el menor número posible de pares de predictores restantes supere dicho umbral, evitando así posibles problemas de multicolinealidad provenientes de tener información duplicada. Tras el estudio realizado, dejaremos de contar con estas variables:

- `fgmTeam`
- `pctFGTeam`
- `fg3aTeam`
- `ftaTeam`
- `%PTS FT`
- `OREB%`
- `AST RATIO`
- `TOV%`
- `AST%`
- `%PTS 3PT`
- `2FGM %AST`
- `TS%`

Además, también descartaremos la variable `countDaysNextGameTeam`, que indica el número de días que faltan para el próximo partido del equipo. En el conjunto también contamos con `countDaysRestTeam`, los días que ha tenido de descanso el equipo antes del partido. A pesar de que el coeficiente de correlación entre estas variables no supera el umbral, es evidente que ambas variables (para cada equipo en particular) contendrán los mismos valores, pero desplazados una observación[3].

Tras haber eliminado variables informativas y redundantes, obtenemos un conjunto de datos de 39 predictores (en lugar de los 74 que teníamos originalmente) con el que continuaremos estudiando la importancia de las variables y posibles efectos interacción.

[3]Por ejemplo, si los Dallas Mavericks tienen $countDaysNextGameTeam = 2$ en el partido x, en el partido x+1 habrán tenido dos días de descanso, es decir, $countDaysRestTeam = 2$, y así sucesivamente.

5.1.5. Importancia de las variables: conjunto manual

Antes de comenzar a estudiar la importancia de las variables, hay que tener en cuenta ciertos aspectos:

- Las decisiones sobre el descarte de variables ya han sido tomadas, y no se eliminarán más predictores del conjunto manual. El estudio de la importancia de las variables se hace con fines exploratorios y de aprendizaje sobre los predictores.
- Más adelante, en el conjunto automático, emplearemos otro enfoque para calcular la importancia de las variables. La idea detrás de esto es poder comparar ambos resultados.
- Para calcular la importancia de las variables en el conjunto manual se comparará el acierto en la predicción de un modelo nulo (que prediga siempre victoria, tendrá un acierto del 50 %) con el acierto en la predicción de un modelo de regresión lineal con un único predictor. Dicho predictor será cada variable de la que queramos medir la importancia. Por lo tanto, cuanto mejor sean las predicciones de los modelos, más importante será la variable que se esté usando como único predictor. Se calculará la diferencia entre aciertos del modelo con una variable y el modelo nulo, siendo más importantes aquellas variables que obtengan mayores mejoras.
- El conjunto de datos con el que se realizarán estas pruebas es algo distinto al conjunto con el que más adelante se entrenarán los modelos. Este conjunto tiene dos observaciones por cada partido, mientras que en el que usaremos más adelante se han unido los registros de cada partido para obtener una única observación por encuentro. Además, en este conjunto se tienen las estadísticas del partido del que se quiere predecir el resultado, mientras que más adelante no se contará con las estadísticas del mismo partido, sino de partidos anteriores. Aún así, como se van a realizar comparaciones entre resultados obtenidos con el mismo conjunto de datos, la prueba es válida. Utilizamos este conjunto porque, en el conjunto de medias, cada partido no sólo influye en una observación, si no en tantas como encuentros se utilicen para calcular la media. Para evitar esto, y que cada partido sólo influya en una observación, empleamos este conjunto sin procesar.
- La partición entre conjunto de entrenamiento y test se realizará de la siguiente manera: se generará una lista de 8912[4] números impares aleatorios y se incluirán en el conjunto de entrenamiento las observaciones correspondientes a los partidos de esos índices, dejando el resto para predecir.

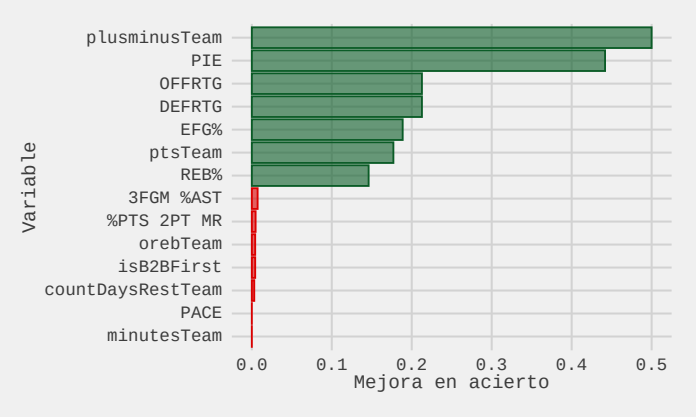
[4]Tenemos 23766 observaciones de 11883 partidos, haremos una partición de 75 % entrenamiento y 25 % test.

Observamos en la Figura 5.9 las 7 variables más importantes y las 7 variables menos importantes según este enfoque. Obviamente, la variable más importante es `plusminusTeam`, ya que si sabemos el diferencial del resultado final vamos a conocer el ganador del partido el 100 % de las ocasiones. Era de esperar también que el impacto estimado de los jugadores del equipo, `PIE`, aportase una mejora importante en la capacidad predictiva del modelo. Se trata de un compendio de estadísticas tanto del equipo como del rival, y en la gran mayoría de ocasiones toma un valor mayor para el equipo vencedor. La inclusión de alguna de estas dos variables hace que el modelo acierte todos (en el caso de `plusminusTeam`) o casi todos los resultados[5].

Después encontramos un conjunto de variables relacionadas con los porcentajes de acierto, los puntos, los rebotes y los ratings que aportan mejoras en la capacidad predictiva de los modelos simples de entre un 12 y un 22 %. Estas son (de mayor a menor importancia): rating ofensivo, rating defensivo, eficacia efectiva de lanzamiento, puntos anotados, proporción de rebotes conseguidos.

En el otro lado tenemos 7 variables que apenas mejoran la capacidad predictiva del modelo simple, siendo la mayor mejora conseguida por alguno de estos predictores del 1 % y habiendo 2 variables que no aportan mejoras en absoluto: los minutos jugados (`minutesTeam`) y el ritmo del partido (`PACE`). El resto de este conjunto de variables menos importantes son: el número de días de descanso antes del partido, los rebotes ofensivos capturados, si se trata el primer partido de un *back to back*, y la proporción de puntos anotados de media distancia.

Figura 5.9: Variables más y menos importantes, conjunto manual.



Fuente: elaboración propia

[5]Volvemos a dejar claro que estamos haciendo "trampasütilizando estadísticas del mismo partido del que se quiere predecir el resultado. Desde este enfoque estamos estudiando qué estadísticas tienen mayor importancia para el resultado de un partido. En los modelos finales no se incluirán estas variables, sino las medias de los últimos partidos de estas estadísticas.

5.1.6. Selección de características e importancia de las variables: conjunto automático

Como hemos mencionado en la introducción de esta sección, para la selección de características vamos a utilizar el algoritmo JMIM o *Joint Mutual Information Maximization*. Después de un estudio bibliográfico se concluyó que, en líneas generales y para problemas de clasificación, este algoritmo presentaba mejores resultados que otros procesos de selección de características. Bennasar *et al.* (2015) [2] utilizaron varios conjuntos de datos del repositorio UCI para demostrar el rendimiento superior de este algoritmo. Lo pondremos en uso con R gracias al paquete `praznik` [11]. Este método, utiliza los conceptos de información mutua y el "máximo del mínimo"para tratar de solventar el problema de sobreestimación de la significatividad de las características (Bennasar *et al.*, 2015 [2]). Mediante la puntuación otorgada por este algoritmo a las diferentes variables, basada en la información mutua entre los predictores y la variable respuesta, obtendremos nuestros datos.

Para este conjunto vamos a utilizar un enfoque ligeramente distinto. Mientras que anteriormente hemos tomado las decisiones de exclusión de variables en base al conjunto con dos observaciones por partido, en este vamos a emplear el algoritmo con los datos procesados y adecuados para su introducción en los modelos[6]. Como el algoritmo es supervisado, es decir, utiliza información de la variable respuesta, lo aplicaremos únicamente al conjunto de entrenamiento, evitando utilizar las muestras de test para tomar decisiones sobre los predictores a incluir. Una vez hayamos decidido qué variables vamos a eliminar, las descartaremos también en el conjunto de test. Procederemos de la siguiente manera. Primero, crearemos un conjunto el mismo número de variables que el conjunto manual, para poder comparar los resultados entre ambos. Además, para estudiar hasta qué punto se consiguen mejorar los resultados reduciendo la dimensionalidad de los datos, crearemos otros tres conjuntos más pequeños, cuyo número de predictores se decidirá al final de este apartado. De este modo, dejaremos de referirnos a un sólo conjunto automático para pasar a tener múltiples conjuntos.

Como el algoritmo otorga puntuaciones a los predictores con ayuda de la información mutua, vamos a emplear esta puntuación como medida de importancia de las variables, cambiando de esta manera el enfoque aplicado para el análisis de los predictores del conjunto manual[7].

Observamos las 7 variables más importantes y las 7 variables menos importantes para el algoritmo en la figura 5.10. A primera vista se aprecia la aparente importancia del número de partido de la temporada, `numberGame-`

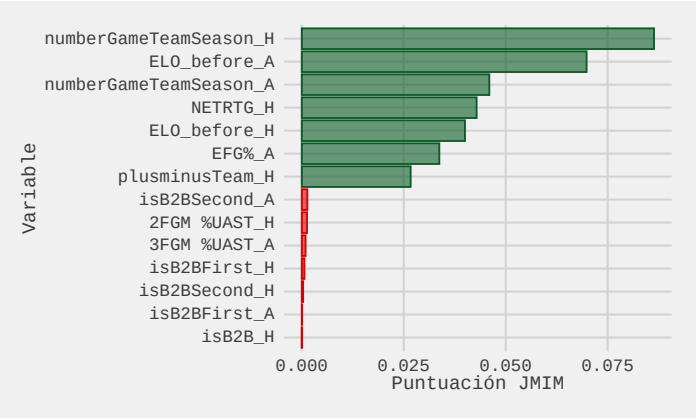
[6] El proceso de preparación y adecuación de los datos se detalla en el próximo capítulo.

[7] Aunque realmente, la esencia sigue siendo la misma, utilizar únicamente la relación entre la variable respuesta y cada uno de los predictores por separado para evaluar la importancia de estos.

`TeamSeason`, que aparece en los primeros puestos. Además, hay algunos predictores que se han determinado importantes bajo ambos enfoques, como es el caso del diferencial de puntos `plusminusTeam` o el rating (en el conjunto manual eran importantes el rating ofensivo y defensivo, mientras que el JMIM considera más importante el rating neto). Completan el grupo de variables más importantes las dos referentes al Rating ELO y el porcentaje efectivo de lanzamiento del equipo visitante, `EFG%_A`.

En cuanto a las variables menos importantes, 5 de las 7 mostradas son referentes a la situación de *back to back*, mientras que las otras dos hacen referencia a la proporción de tiros de 2 y de 3 anotados sin haber recibido previamente una asistencia (`2FGM %UAST_H` y `3FGM %UAST_A`).

Figura 5.10: Variables más y menos importantes, conjunto automático.



Fuente: elaboración propia

Teniendo esto en cuenta procederemos de la siguiente manera: primero crearemos un conjunto con 81 variables para poder comparar su rendimiento con el conjunto manual, de igual número de predictores[8]. Además, estableceremos un umbral en una puntuación de 0.01, incluyendo todas las variables que puntúen por encima de ese umbral, 29 en total. Asimismo, crearemos dos conjuntos más pequeños para realizar más pruebas, uno con 14 (la mitad del número de variables que superan el umbral de importancia establecido) y otro con 10 predictores, un conjunto pequeño para ver cuáles son los resultados con tan pocas variables. Se escogieron 10 porque menos predictores se consideraban demasiado pocos.

[8]Las 39 variables seleccionadas en el conjunto manual, 40 con el Rating ELO, acabarán siendo 80 al llevar a cabo los procesos de adecuación del conjunto a los modelos. Añadiendo la variable respuesta, tendrá 81 predictores en total.

5.2. Ingeniería de características

Tratamos ahora de estudiar si existen efectos de interacción en nuestros conjuntos de datos. Esto ocurre cuando el efecto conjunto de las variables que interactúan es mayor que su efecto por separado. Sólo buscaremos interacciones de dos variables entre sí, ya que Neter *et al.* (1996) [16] demostraron que interacciones de mayor grado ocurren muy poco frecuentemente y no aportan mejoras significativas en la capacidad predictiva. Para ello utilizaremos un enfoque parecido al empleado para analizar la importancia de las variables. Se computarán todas las interacciones posibles entre dos variables y, utilizando una regresión lineal, se compararán los aciertos de la predicción en test de un modelo con las dos variables a estudiar como predictores y de un modelo con la interacción entre estas dos como único predictor, que se modelará multiplicando las variables.

5.2.1. Conjunto manual

Al estudiar este conjunto observamos que hay muy pocas interacciones que mejoren los resultados conseguidos con los modelos con las variables aditivas. Tras realizar varias pruebas en las que se incluían algunas de las mejores interacciones y no se obtenían mejores resultados, se decidió **no** incluir ninguna interacción en este conjunto que, tras el proceso de selección de características, tendrá 39 variables, 40 tras incluir el Rating ELO. En la siguiente tabla 5.2.2 se muestran las interacciones que superan el 0.75 % en mejora de la capacidad predictiva.

5.2.2. Conjuntos automáticos

Al igual que en el conjunto manual, en los conjuntos automáticos no hay ninguna interacción cuya inclusión aporte lo suficiente. También hay muy pocas que mejoren la capacidad predictiva, siendo de un 0.6 % la mayor mejora conseguida. De este modo, tampoco incluiremos ninguna interacción en estos conjuntos, que se quedarán con 81, 29, 14 y 10 variables. En la siguiente tabla observamos las interacciones que superan el 0.5 % de mejora.

Tabla 5.1: Mejores interacciones de ambos conjuntos.

Conjunto manual		Conjuntos automáticos	
Interacción	**Mejora en capacidad predictiva**	**Interacción**	**Mejora en capacidad predictiva**
pfTeam x plusminusTeam	1.16 %	DEFRTG_A x PIE_H	0.61 %
isB2BSecond x fg2aTeam	0.96 %	DEFRTG_H x numberGameTeamSeason_A	0.55 %
isB2BSecond x %FGA 3PT	0.91 %	plusminusTeam_A x pctFGTeam_H	0.55 %
fgaTeam x plusminusTeam	0.79 %	plusminusTeam_A x OFFRTG_H	0.51 %

Fuente: elaboración propia

Capítulo 6

Modelos y resultados

Una vez tenemos nuestros nuevos conjuntos vamos a volver a ajustar los modelos utilizados en el capítulo 4, intentando mejorar la marca de acierto conseguida anteriormente, obtenida por una máquina de vectores soporte con un 67.93 % de precisión en test y 1989 aciertos. Para ello, partimos de los conjuntos generados en la sección anterior. El conjunto manual contiene dos observaciones por partido, mientras que los automáticos ya están listos para introducir en los modelos, ya que se han obtenido a partir de los conjuntos de entrenamiento y evaluación con estadísticas básicas y avanzadas utilizado en el capítulo 4. Los datos con los que entrenaremos los modelos contienen las medias de las estadísticas en los partidos anteriores de los equipos participantes, así como el Rating ELO de cada equipo antes de comenzar el encuentro. Haremos una partición del 75 % para entrenar los modelos, utilizando el 25 % restante para evaluarlos, y emplearemos validación cruzada con 3 folds y un *gridsearch* para tunear los hiperparámetros de los modelos, eligiendo aquellos parámetros con los que se obtenga un mayor acierto en validación. En busca del mejor resultado posible, haremos diferentes tandas de pruebas, empleando en cada una de ellas un conjunto de estadísticas con un número distinto de partidos utilizados para computar la media, desde solo el último partido hasta los últimos 10 partidos. Se espera que, al utilizar los conjuntos reducidos, se consiga simplificar los modelos, resultando en un ajuste más sencillo de los hiperparámetros y, por consiguiente, en una marca superior.

6.1. Adecuación de los datos a los modelos

El conjunto manual tiene 23766 observaciones, es decir, dos observaciones por partido, mientras que los conjuntos automáticos ya tienen las 11718 observaciones que buscamos. Mientras que el primero tiene 39 variables (que acabarán siendo 81 tras su adecuación al modelo), los últimos tienen 10, 14,

29 y 81 variables respectivamente, estando el primero compuesto por las variables rojas, el segundo por las rojas y azules, el tercero por las rojas, azules y violetas y el cuarto por todas las variables que observamos en la siguiente tabla.

Recordemos las variables incluidas en cada conjunto:

Tabla 6.1: Variables en cada conjunto.

Conjunto manual		Conjuntos automáticos	
ELO_before	numberGameTeamSeason	ELO_before_H	numberGameTeamSeason_H
isB2B	isB2BFirst	ELO_before_A	numberGameTeamSeason_A
isB2BSecond	countDaysRestTeam	NETRTG_H	EFG %_A
avgPts	fgaTeam	plusminusTeam_H	EFG %_H
fg3mTeam	pctFG3Team	PIE_H	NETRTG_A
pctFTTeam	fg2mTeam	DEFRTG_H	PIE_A
fg2aTeam	pctFG2Team	plusminusTeam_A	REB %_H
minutesTeam	ftmTeam	OFFRTG_H	pctFG2Team_A
orebTeam	drebTeam	AST/TO_H	OFFRTG_H
astTeam	stlTeam	TS %_A	avgPts_A
blkTeam	tovTeam	pctFGTeam_A	pctFG3Team_H
pfTeam	plusminusTeam	TS %_H	DEFRTG_A
OFFRTG	DEFRTG	fg2aTeam_H	AST RATIO_A
AST/TO	DREB %	pctFG2Team_H	avgPts_H
REB %	EFG %	drebTeam_H	%PTS 2PT MR_H
PACE	PIE	pctFG3Team_A	AST RATIO_H
%FGA 3PT	%PTS 2PT	drebTeam_A	fgmTeam_A
%PTS 2PT MR	%PTS FBPS	AST/TO_A	pctFTTeam_A
%PTS OFF TO	%PTS PITP	fgmTeam_H	ftmTeam_H
3FGM %AST	FGM %AST	blkTeam_A	fg2aTeam_H
		fg3mTeam_A	fg3mTeam_H
		TOV %_A	pfTeam_H
		REB %_A	%FGA 2PT_A
		stlTeam_H	astTeam_H
		fgaTeam_A	astTeam_A
		2FGM %AST_A	%PTS 2PT MR_A
		blkTeam_H	trebTeam_H
		2FGM %AST_H	%PTS 2PT_A
		countDaysNextGameTeam_H	tovTeam_H
		%PTS PITP_A	DREB %_A
		DREB %_H	PACE_A
		pfTeam_A	%PTS FBPS_H
		ftaTeam_H	pctFTTeam_H
		trebTeam_A	fg3aTeam_H
		fg3aTeam_A	%PTS FBPS_A
		%PTS 2PT_H	fg2mTeam_H
		%PTS FT_H	countDaysRestTeam_A
		orebTeam_A	fgaTeam_H
		countDaysNextGameTeam_A	stlTeam_A
		ftmTeam_A	

Fuente: elaboración propia

Para transformar el conjunto manual primero juntamos las observaciones correspondientes al mismo partido, reduciendo el número de registros a la mitad. Así, duplicaremos todas las variables, añadiendo un sufijo a su nombre indicando si se trata del equipo local o del equipo visitante (H: *Home*, A: *Away*). De esta manera, conseguiremos una sola observación por

encuentro. Después, a partir de las variables `isWin` y `locationGame` creamos la que será la variable respuesta, `H_Win`, que tomará el valor 1 si el equipo local gana el encuentro y 0 si el vencedor es el equipo visitante. Lo siguiente que haremos será realizar los cálculos necesarios para incluir el Rating ELO en nuestra base de datos.

Posteriormente creamos un conjunto de datos por equipo, donde calcularemos la media de los últimos n encuentros para todos los partidos disputados por cada equipo. Esto lo haremos por temporada, es decir, si en la temporada presente aún no se han disputado n partidos, se computará la media de todos los que se hayan disputado hasta la fecha en la temporada correspondiente. Esto implica que cuando creemos el conjunto de datos que se introducirá en los modelos **se descartarán todas las observaciones correspondientes a los primeros partidos de los equipos de cada temporada**, al no tener medias de partidos anteriores que poder calcular. Una vez tenemos las medias pasamos a confeccionar el conjunto que introduciremos en los modelos. Con ayuda de las variables informativas averiguamos qué equipos disputan cada partido, quién es el local y quién el visitante, para construir cada observación con las medias de los partidos anteriores de los equipos correspondientes. Cabe recalcar que **no estamos utilizando los eventos del partido que estamos tratando de predecir**, estamos teniendo en cuenta el estado de forma de los equipos en los **encuentros anteriores**. Acabaremos con conjuntos que tendrán 11718 observaciones, con 81 variables en caso del conjunto manual y 10, 14, 29 y 81 variables en los conjuntos automáticos.

6.2. Resultados

6.2.1. Conjunto manual

Con el conjunto manual no hemos logrado superar la marca a batir. El mejor resultado lo hemos obtenido con un AdaBoost utilizando las medias de los 6 y 10 partidos anteriores[1]. En la Tabla 6.1 y la Figura 6.1 se observa un desglose resumido de los resultados, mientras que los resultados completos figuran en el Apéndice B. Apreciamos como es el AdaBoost el algoritmo que mejores resultados da para casi cualquier número de partidos de media, siendo el modelo de k vecinos más próximos el que peor funciona. Además, no parece existir una relación clara entre el número de partidos utilizados para calcular la media y el acierto, aunque sí que se aprecia que, utilizando tanto 5 como 6 y 10 partidos, se obtienen generalmente mejores resultados.

El hecho de no llegar a la marca conseguida anteriormente puede suceder por varias razones. Es posible que la selección de características no haya sido óptima para conseguir mejores resultados (recordemos que la toma

[1]Con ambos se ha obtenido el mismo resultado.

de decisiones ha sido humana, pudiendo esta ser peor que la de un algoritmo matemático sofisticado). También puede ser que se haya eliminado demasiada información, o que simplemente no se hayan encontrado los hiperparámetros idóneos para conseguir una mejor marca en el conjunto de test, ya que en muchas ocasiones los parámetros que maximizaban los resultados en validación funcionaban considerablemente peor en test (a veces incluso obtenían cerca de un 2 % menos de acierto).

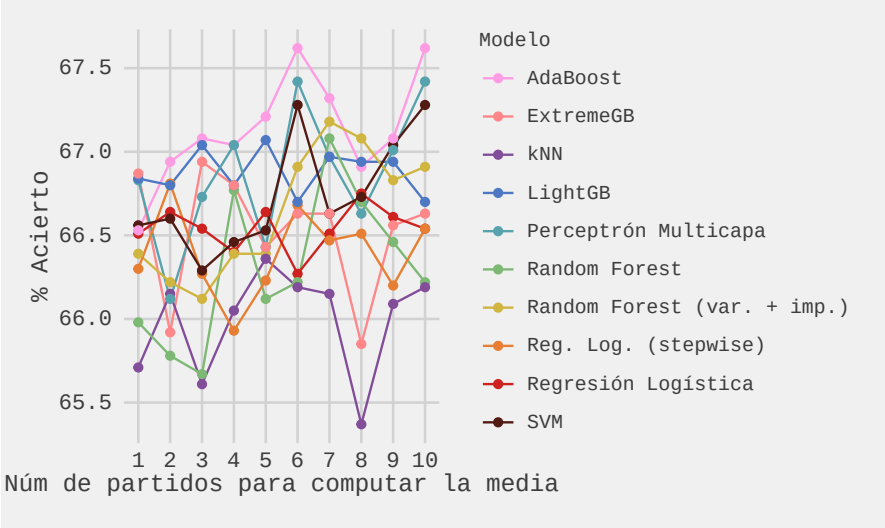
En resumen, el mejor resultado con este conjunto es de un 67.62 % de precisión y 1980 aciertos, 9 menos que el objetivo. A continuación emplearemos los conjuntos automáticos, tratando de mejorar esta marca.

Tabla 6.2: Resumen de los resultados con el conjunto manual.

Modelo	Mayor precisión	Mayor núm. aciertos	Núm. partidos utilizados para computar la media
Regresión Logística	66.75	1955	8
Reg. Log. (stepwise)	66.81	1957	2
Random Forest	67.08	1964	7
Random Forest (variables + imp.)	67.18	1967	7
SVM	67.28	1970	6, 10
kNN	66.36	1943	5
AdaBoost	67.62	1980	6, 10
MLP	67.42	1974	6, 10
LightGBM	67.07	1964	5
ExtremeGB	66.94	1960	3

Fuente: elaboración propia

Figura 6.1: Resultados del conjunto manual.



Fuente: elaboración propia

6.2.2. Conjuntos automáticos

Recordemos que estos conjuntos se han confeccionado con el algoritmo *Joint Mutual Information Maximization*, en el que se utiliza información de la variable respuesta para asignar puntuaciones a cada predictor. Además, en vez de reducir la dimensionalidad antes de adecuar el conjunto a los modelos, hemos aplicado el algoritmo sobre el conjunto ya preparado. A partir de ahí, hemos creado cuatro grupos distintos de variables a incluir basándonos en la puntuación que otorgaba el algoritmo los predictores, quedándonos con 81 (mismo número de predictores que el conjunto manual), 29, 14 y 10 variables respectivamente.

Si comparamos el conjunto manual con el conjunto automático de 81 variables, observamos que los resultados son generalmente mejores en el último. Solamente con el AdaBoost y el perceptrón multicapa se obtienen marcas superiores en el conjunto manual, funcionando el resto de modelos mejor con el conjunto automático.

Con el resto de conjuntos no solo logramos superar los primeros resultados, si no que conseguimos batir la mejor marca obtenida en los modelos base. Utilizando los registros del partido anterior sobre el conjunto con 14 predictores hemos obtenido, mediante un Random Forest, una precisión del 68.03 %, con 1992 aciertos, mejorando así nuestra previa mejor marca en 3 aciertos. Aunque sólo lo hayamos conseguido en una de las muchas pruebas que se han realizado (10 modelos, 10 partidos, 3 conjuntos $\longrightarrow$ 300 pruebas), los resultados con todos los conjuntos automáticos han sido generalmente bastante mejores que los obtenidos con el conjunto manual o los modelos base.

Curiosamente, el Random Forest, algoritmo con el que tanto Morate (2016) [15], como Weiner *et al.* (2021) [9] obtuvieron su registro superior, pero que hasta ahora no había destacado por su rendimiento, ha sido con el que hemos conseguido obtener esta marca, la más alta obtenida hasta el momento. Por otra parte, el algoritmo k vecinos más próximos, que era con el que peores resultados obteníamos en el conjunto manual, ha funcionado medianamente bien con los automáticos. Cabe destacar también que, tanto la máquina de vectores soporte como el perceptrón multicapa se han mantenido bastante regulares en cuanto a su acierto, independientemente del conjunto introducido o del número de partidos utilizados para calcular la media. Además, la regresión logística, modelo con el que la mayoría de autores revisados[2] consiguieron su mejor marca, ha presentado un gran rendimiento con los conjuntos automáticos, con aciertos de entre el 66.87 % y el 67.90 %.

[2]Cao (2012) [4], Lieder (2018) [12], Villar (2019) [20] y Jones (2016) [10] obtuvieron su mejor marca mediante la regresión logística.

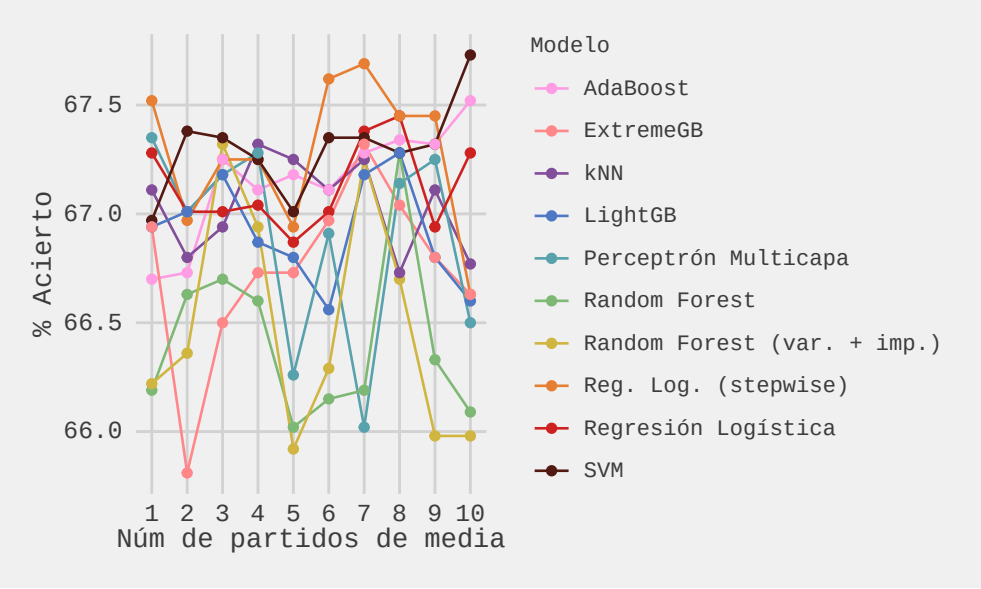
A continuación se observan los resultados resumidos de los conjuntos automáticos, que se muestran más detalladamente en el Apéndice B.

Tabla 6.3: Resumen de los resultados con el conjunto automático (81 vars.).

Modelo	Mayor precisión	Mayor núm. aciertos	Núm. partidos utilizados para computar la media
Regresión Logística	67.45	1975	8
Reg. Log. (stepwise)	67.69	1982	7
Random Forest	67.28	1970	8
Random Forest (variables + imp.)	67.32	1971	3
SVM	67.38	1973	2
kNN	67.32	1971	4
AdaBoost	67.52	1977	10
MLP	67.35	1972	1
LightGBM	67.28	1970	8
ExtremeGB	67.32	1971	7

Fuente: elaboración propia

Figura 6.2: Resultados del conjunto automático (81 vars.).



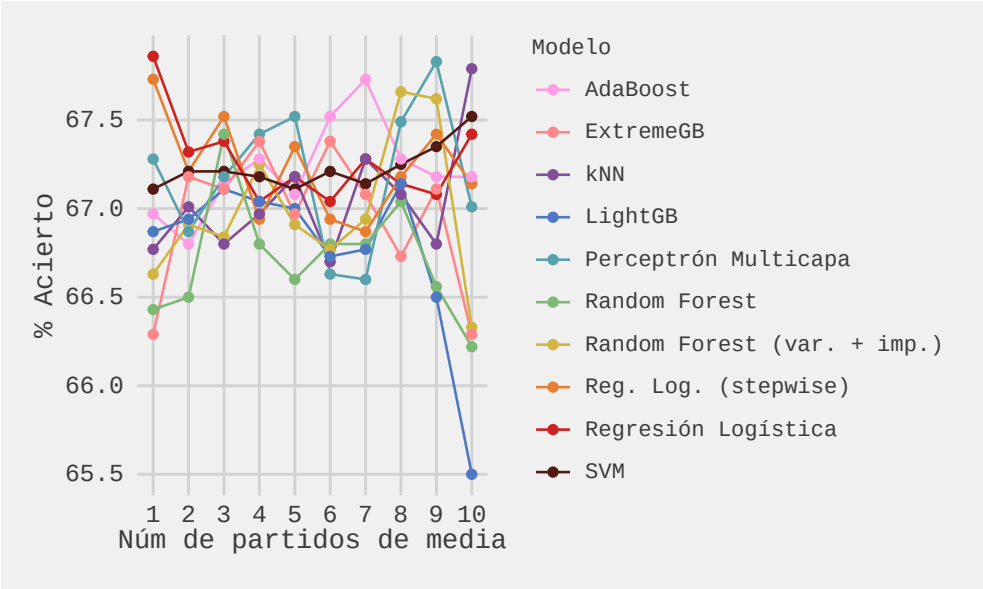
Fuente: elaboración propia

Tabla 6.4: Resumen de los resultados con el conjunto automático (29 vars.).

Modelo	Mayor precisión	Mayor núm. aciertos	Núm. partidos utilizados para computar la media
Regresión Logística	67.86	1987	1
Reg. Log. (stepwise)	67.73	1983	1
Random Forest	67.42	1974	3
Random Forest (variables + imp.)	67.66	1981	8
SVM	67.52	1977	10
kNN	67.79	1985	10
AdaBoost	67.73	1983	7
MLP	67.83	1986	9
LightGBM	67.14	1966	8
ExtremeGB	67.38	1973	4, 6

Fuente: elaboración propia

Figura 6.3: Resultados del conjunto automático (29 vars.).



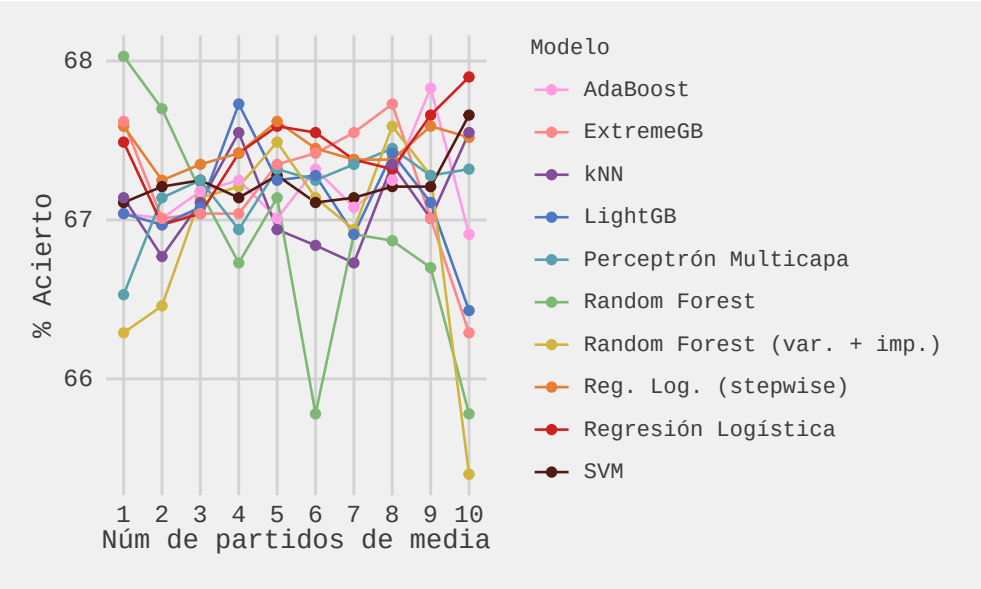
Fuente: elaboración propia

Tabla 6.5: Resumen de los resultados con el conjunto automático (14 vars.).

Modelo	Mayor precisión	Mayor núm. aciertos	Núm. partidos utilizados para computar la media
Regresión Logística	67.90	1988	10
Reg. Log. (stepwise)	67.62	1980	5
Random Forest	68.03	1992	1
Random Forest (variables + imp.)	67.59	1979	8
SVM	67.66	1981	10
kNN	67.55	1978	4, 10
AdaBoost	67.83	1986	9
MLP	67.45	1975	8
LightGBM	67.73	1983	4
ExtremeGB	67.73	1983	8

Fuente: elaboración propia

Figura 6.4: Resultados del conjunto automático (14 vars.).



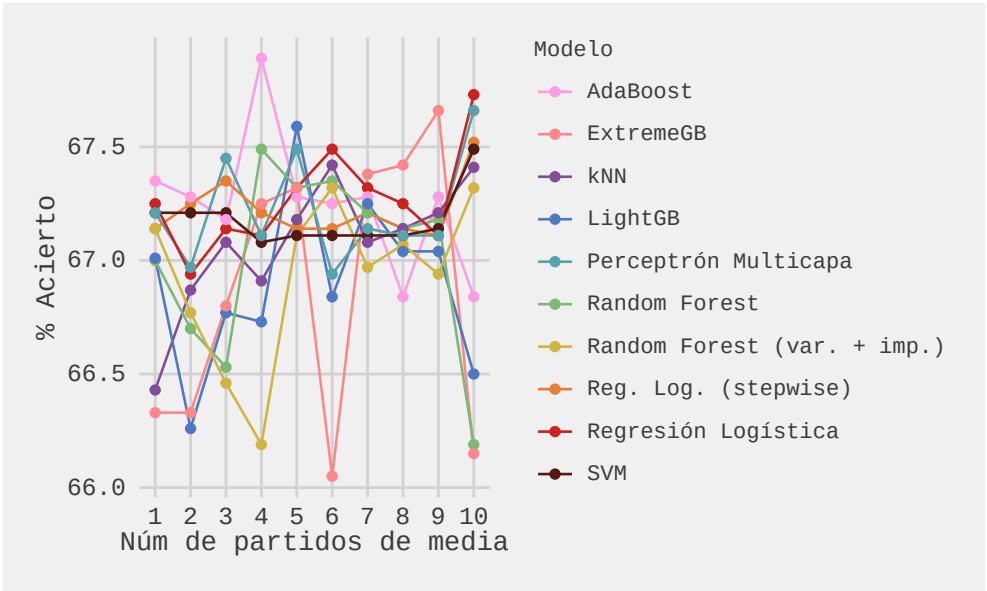
Fuente: elaboración propia

Tabla 6.6: Resumen de los resultados con el conjunto automático (10 vars.).

Modelo	Mayor precisión	Mayor núm. aciertos	Núm. partidos utilizados para computar la media
Regresión Logística	67.73	1983	10
Reg. Log. (stepwise)	67.52	1977	10
Random Forest	67.49	1976	4
Random Forest (variables + imp.)	67.32	1971	6, 10
SVM	67.49	1976	10
kNN	67.42	1974	6, 10
AdaBoost	67.89	1987	4
MLP	67.66	1981	10
LightGBM	67.59	1979	5
ExtremeGB	67.66	1981	9

Fuente: elaboración propia

Figura 6.5: Resultados del conjunto automático (10 vars.).



Fuente: elaboración propia

6.3. Revisión de los modelos base

Considerando que solo se han mejorado los resultados conseguidos en el capítulo 4 en 3 aciertos, y teniendo en cuenta que en dicho capítulo no se realizó un tuneado fino de hiperparámetros, si no que fue más bien básico, hemos creído apropiado revisar los modelos base que mejor han funcionado y, mediante una búsqueda más exhaustiva, tratar de encontrar los hiperparámetros concretos que nos permitan mejorar los resultados (si es que existen).

Cabe repetir que estamos realizando validación cruzada, es decir, para escoger los hiperparámetros con los que luego evaluaremos los modelos nos estamos basando en los resultados conseguidos en validación.

En los tres conjuntos probados en el capítulo 4, los modelos que mejor funcionan son la máquina de vectores soporte y el Light Gradient Boosting, obteniendo la primera mejores resultados con el conjunto más complejo, con estadísticas básicas y avanzadas. Así pues, con ayuda de estos dos modelos (el primero con el conjunto de estadísticas básicas y avanzadas y el segundo con el conjunto de diferencias) se tratará de mejorar la marca obtenida anteriormente de 1992 aciertos y una precisión del 68.03 %.

6.3.1. Máquina de vectores soporte

La marca conseguida en el capítulo 4 con este modelo era de 1989 aciertos y un 67.93 % de precisión. Con los parámetros adecuados hemos conseguido superarla, obteniendo 1992 aciertos e igualando el resultado obtenido previamente en este capítulo. Dichos parámetros son los siguientes:

Tabla 6.7: Parámetros de la máquina de vectores soporte.

Parámetro	Valor	Significado
kernel	"rbf"	Tipo de kernel utilizado por el algoritmo
C	0.9	Parámetro de regularización Fuerza inversamente proporcional al valor del parámetro
gamma	"scale"	Coeficiente kernel $gamma = \frac{1}{(num\_características \cdot var\_X)}$

Fuente: elaboración propia

6.3.2. Light Gradient Boosting

Aplicando este modelo al conjunto de diferencias conseguimos superar, por primera vez, la barrera de los 2000 aciertos. Concretamente se obtiene una precisión del 68.41 %, acertando 2003 partidos. Los parámetros aplicados para conseguir esta marca son:

Tabla 6.8: Parámetros del Light Gradient Boosting.

Parámetro	Valor	Significado
n_estimators	22	Número de árboles
max_depth	6	Máxima profundidad para los árboles
colsample_bytree	0.9	Tamaño de la submuestra de columnas para cada árbol
min_split_gain	0.3	Mínima reducción de la función de coste para hacer otra partición
num_leaves	11	Número máximo de hojas
reg_alpha	1.2	Regularización L1
reg_lambda	1.1	Regularización L2
subsample	0.7	Ratio de submuestreo del conjunto de entrenamiento

Fuente: elaboración propia

Teniendo en cuenta que hemos conseguido nuestra mejor marca con el conjunto de diferencias, y considerando la mejora en rendimiento obtenida sobre los conjuntos automáticos después de aplicar el algoritmo JMIM, el siguiente paso lógico es comprobar si, reduciendo la dimensión del conjunto de diferencias, obtenemos mejores resultados. Tras realizar las pruebas pertinentes[3] podemos afirmar que no ha sido así, siendo la mejor marca obtenida de este modo de un 67.73 % de precisión y 1983 aciertos. El desglose de los resultados de esta prueba también se encuentra en el Apéndice B.

Hemos logrado mejorar la marca obtenida en los modelos base, rozando los 2000 aciertos en varias ocasiones e incluso superándolos con algún modelo. A continuación, procedemos a realizar un estudio acerca de la robustez de los resultados obtenidos.

[3]Se aplicó el algoritmo JMIM al conjunto de diferencias, manteniendo las 14 variables con mayor puntuación (mismo número de predictores que con el que se obtuvo la mejor marca de los conjuntos automáticos), y se ajustaron todos los modelos con medias de entre 1 a 10 partidos.

Capítulo 7

Análisis y validación de los resultados

En este capítulo vamos a comprobar la robustez de los resultados obtenidos hasta ahora. A pesar de haber conseguido marcas considerablemente buenas, el azar puede haber jugado un factor importante en el buen funcionamiento de los modelos sobre los conjuntos de test. Para verificar si ha sido así o si, por el contrario, nuestros resultados son robustos (para nuestra partición entrenamiento/test), vamos a poner los tres modelos con los que mejores resultados hemos obtenido frente a frente. A partir de los conjuntos con los que cada modelo ha conseguido su mejor marca, vamos a generar 10 réplicas bootstrap tanto del subconjunto de entrenamiento como del de test. Para cada réplica, buscaremos los parámetros que mejor funcionen con el modelo correspondiente, y evaluaremos en el subconjunto de test. Con los 10 resultados que obtendremos, comprobaremos si nuestras mejores marcas son robustas o si han sido fruto de un golpe de suerte. En la siguiente tabla observamos los tres modelos con los que mejores marcas hemos obtenido y las especificaciones de cada caso. Cabe destacar, que la marca obtenida con la máquina de vectores soporte también se consiguió con un Random Forest y el conjunto automático de 14 variables con medias de un partido. Sin embargo, no vamos a incluir este modelo en la sección, ya que, a excepción de ese buen resultado, en el resto de casos funcionaba bastante mal, lo que lleva a pensar que ese resultado sí que habría sido cuestión de suerte. Además, ya tenemos un modelo basado en árboles de decisión en esta sección, el Light Gradient Boosting.

Tabla 7.1: Mejores resultados obtenidos.

Pos.	Modelo	Precisión	Aciertos	Conjunto
1	Light Gradient Boosting	68.41	2003	Conjunto diferencias 60 variables Medias de 10 partidos
2	SVM	68.03	1992	Conjunto completo 119 variables Medias de 10 partidos
3	Regresión logística	67.90	1988	Conjunto automático 14 variables Medias de 10 partidos

Fuente: elaboración propia

Antes de volver a ajustar los modelos a las réplicas bootstrap, vamos a analizar los resultados obtenidos, observando las métricas y matrices de confusión correspondientes. Nos daremos cuenta de lo siguiente: los modelos funcionan realmente mal con las derrotas del equipo local, no llegando siquiera al 50 % de acierto en el mejor de los casos. Como las clases están desbalanceadas, saliendo victorioso el equipo local en más del 58 % de partidos de nuestra muestra, y sumado esto al hecho de que la métrica de evaluación más utilizada en los estudios de este campo es la precisión (en inglés *accuracy*, suma de predicciones correctas entre el total de observaciones, no confundir con *precision*, verdaderos positivos entre predicciones positivas), los modelos se esfuerzan más en predecir mejor las victorias locales, funcionando razonablemente bien para estas, a costa de reducir su rendimiento para las derrotas. Debido a esto, para analizar los resultados utilizaremos las siguientes métricas:

- ***Precision.*** Mide el número de verdaderos positivos entre el total de positivos predichos por el modelo. La utilizaremos para cada clase por separado:
  $\text{Precision} = \frac{Verdaderos\ Positivos}{Verdaderos\ Positivos\ + Falsos\ Positivos}$

- ***Recall.*** Mide el número de verdaderos positivos entre el total de positivos reales en el conjunto de datos. También la utilizaremos para cada clase por separado:
  $\text{Recall} = \frac{Verdaderos\ Positivos}{Verdaderos\ Positivos\ + Falsos\ Negativos}$

- **Área bajo la curva ROC (AUC).** La curva ROC (*Receiver Operating Characteristic*) se utiliza para representar el rendimiento de un clasificador binario frente a un modelo aleatorio. Se muestra el espacio definido por la ratio de falsos positivos y la ratio de verdaderos positivos. Como métrica para resumir el rendimiento del modelo, se suele emplear el área bajo esta curva, siendo mejores los valores más cercanos a 1.

Resultados: Light Gradient Boosting

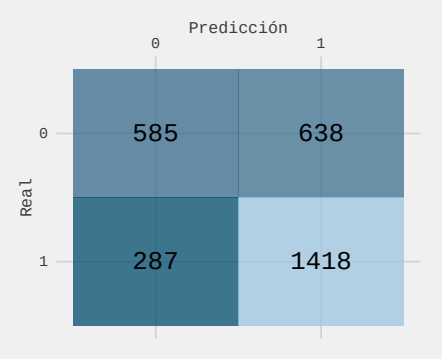
Observamos en la matriz de confusión de este modelo lo que va a ser la tónica habitual de esta sección: los modelos no funcionan bien con las derrotas del equipo local. Este, concretamente, sólo predice correctamente un 47.83 % de las derrotas. En cambio, con las victorias su rendimiento es muy bueno, acertando un 83.17 %. Recordemos que este es el modelo con el que habíamos obtenido la mejor marca hasta ahora.

Tabla 7.2: Métricas del modelo Light Gradient Boosting.

Clase	*Precision*	*Recall*	*Accuracy*	AUC
0	0.6709	0.4783	0.6841	0.7169
1	0.6897	0.8317		

Fuente: elaboración propia

Figura 7.1: Matriz de confusión del modelo Light Gradient Boosting.



Fuente: elaboración propia

Resultados: SVM

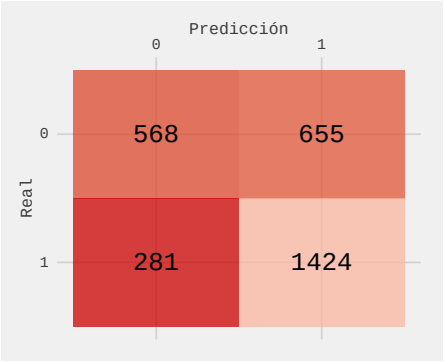
Las métricas son muy parecidas al modelo anterior, aunque este funciona ligeramente peor en las derrotas, pero ligeramente mejor en las victorias. Además, la AUC de este modelo es algo superior a la del Light Gradient Boosting, a pesar de tener peor *accuracy*.

Tabla 7.3: Métricas de la máquina de vectores soporte.

Clase	*Precision*	*Recall*	*Accuracy*	AUC
0	0.6690	0.4644	0.6803	0.7197
1	0.6849	0.8352		

Fuente: elaboración propia

Figura 7.2: Matriz de confusión de la máquina de vectores soporte.



Fuente: elaboración propia

Resultados: Regresión logística

En cuanto a *accuracy*, la regresión logística presenta la menor de los cuatro modelos. A pesar de ello, es el modelo que mejor AUC tiene, y que mejor funciona en las derrotas. En contraposición, su rendimiento con las victorias también es el menor de los cuatro.

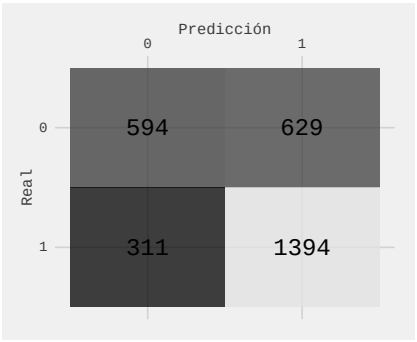
Con este análisis, dejamos clara la magnitud del problema que ha presentado el desbalanceo de clases en nuestro estudio, y planteamos como futuras metas llevar a cabo procesos de balanceo de clases para tratar de encontrar modelos que funcionen mejor en las derrotas, sin perder precisión en las victorias.

Tabla 7.4: Métricas de la regresión logística.

Clase	*Precision*	*Recall*	*Accuracy*	AUC
0	0.6564	0.4857	0.6790	0.7212
1	0.6891	0.8136		

Fuente: elaboración propia

Figura 7.3: Matriz de confusión de la regresión logística.



Fuente: elaboración propia

Réplicas bootstrap

Tras ajustar de nuevo los modelos a las réplicas bootstrap, observamos que generalmente, los resultados obtenidos son robustos, ya que el rango intercuartílico de estos no supera el 1.5 % en todos los casos. Además, las marcas obtenidas están entre la precisión mínima y máxima en todos los casos. El Light Gradient Boosting no ha conseguido obtener una marca superior a la del capítulo anterior con ninguna de las réplicas, consiguiendo, como mucho, igualarla. La regresión logística funciona algo mejor, acercándose más la mediana de los resultados a la precisión obtenida anteriormente, aunque el modelo más robusto es la máquina de vectores soporte. Ha superado los 2000 aciertos con 4 de las 10 réplicas y, a pesar de tener la mayor desviación típica, es el modelo que mejores resultados ha obtenido regularmente. Además, el principal causante de una desviación tan alta es un único resultado especialmente pobre, estando el resto por encima del 66.80 %. En la Figura 7.4 y la Tabla 7.6 se muestran los resultados de esta validación.

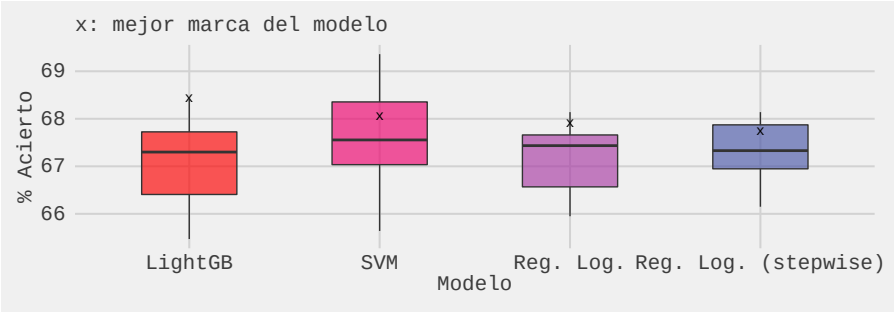
Para constatar la significatividad de las diferencias en los resultados obtenidos, vamos a llevar a cabo un test t sobre la diferencia de las medias de los dos mejores resultados, obtenidos con la máquina de vectores soporte y la regresión logística empleando el método stepwise para la elección del modelo. Esta prueba requiere de unos requisitos previos para poder llevarse a cabo, entre los que se encuentran la normalidad de las muestras y la igualdad de sus varianzas. Para comprobar el primero, se realizó el test de normalidad de Shapiro-Wilk, en el que se tiene como hipótesis nula la normalidad de la población de la que proviene la muestra. Este test se llevó a cabo con los resultados de ambos modelos por separado, estando ambos p-valores por encima de cualquier nivel típico de significatividad, con lo que no podemos rechazar la hipótesis nula, así que asumiremos que las muestras siguen una distribución normal.

Mediante un test de Levene contrastamos la hipótesis nula de igualdad de varianzas. Nuevamente, un p-valor superior a cualquier nivel de significatividad típico nos impide rechazar la hipótesis nula, por lo que asumiremos varianzas iguales. Ya podemos realizar el test t de igualdad de medias. Una vez más, el p-valor es demasiado grande como para rechazar la hipótesis nula, que en este caso supone igualdad de medias, con lo cual, no podemos afirmar que la diferencia entre los resultados obtenidos mediante la máquina de vectores soporte y la regresión logística sea significativa. Los resultados de los tests realizados para llegar a esta conclusión se encuentran resumidos en la Tabla 7.6.

Además, tras realizar pruebas a todos los pares de resultados distintos, podemos afirmar que no existen diferencias significativas en los resultados de los modelos sobre las réplicas bootstrap. Así pues, pese a ser la máquina de vectores soporte el modelo más robusto, sus resultados no son signifi-

cativamente distintos los demás modelos probados en esta sección. Asimismo, retomando lo comentado anteriormente, es, entre estos tres modelos, el que peores resultados ofrece en las derrotas locales, aunque dado el hecho de que los estudios de este campo están muy centrados en obtener la mayor *accuracy* posible, aceptaremos también las limitaciones que esto supone. Por todo esto, acuñamos como mejor marca conseguida en este trabajo la del 68.41 % de *accuracy* y 2003 aciertos, obtenida mediante el modelo Light Gradient Boosting, y como modelo más robusto para nuestra partición en conjunto de entrenamiento y test la máquina de vectores soporte, aunque tanto el Light Gradient Boosting como la regresión logística son igualmente válidos, no presentando diferencias significativas en sus resultados.

Figura 7.4: Resultados del ajuste a las réplicas bootstrap.



Fuente: elaboración propia

Tabla 7.5: Estadísticos de los resultados en las réplicas bootstrap.

Modelo	Min.	Media	Mediana	Máx.	Desviación típica
Light Gradient Boosting	65.47 (1917)	67.11 (1965)	67.30 (1971)	68.41 (2003)	0.91 (27)
SVM	65.64 (1922)	67.67 (1981)	67.56 (1978)	69.36 (2031)	1.10 (32)
Regresión logística	65.95 (1931)	67.17 (1967)	67.44 (1975)	68.14 (1995)	0.76 (22)
Reg. log. (stepwise)	66.15 (1937)	67.29 (1970)	67.33 (1971)	68.14 (1995)	0.70 (20)

Fuente: elaboración propia

Tabla 7.6: Pruebas realizadas para constatar la no significatividad de la diferencia entre resultados, SVM y Reg. Log. (stepwise).

Test	H0	P-valor	Decisión
Shapiro-Wilk Resultados SVM	Normalidad	0.965	No rechazamos H0 Asumimos normalidad
Shapiro-Wilk Resultados RL (step.)	Normalidad	0.414	No rechazamos H0 Asumimos normalidad
Levene Resultados SVM + RL (step.)	Igualdad de varianzas	0.289	No rechazamos H0 Asumimos igualdad
Test t Resultados SVM + RL (step.)	Igualdad de medias	0.360	No rechazamos H0 Asumimos igualdad

Fuente: elaboración propia

Capítulo 8

Conclusiones y futuros trabajos

8.1. Conclusiones

Este estudio se ha centrado en encontrar un modelo de aprendizaje automático capaz de predecir el ganador de un partido de la NBA con una fiabilidad aceptable. Debido a la creciente popularidad de la inteligencia artificial y de su aplicación en los deportes, son varios los autores que han investigado en este campo, desarrollando sus propias versiones de los modelos para alcanzar la mejor marca posible, mediante diferentes datos, temporadas, modelos y técnicas. El hecho de que cada estudio tenga un enfoque particular hace que la comparación entre ellos resulte algo injusta si se utiliza únicamente la métrica de precisión (*accuracy)* para resumir y clasificar todo el trabajo como mejor o peor. De todos se puede aprender y extraer conclusiones muy interesantes que, sin duda, han influido a la hora de realizar este trabajo.

Para la elaboración de este estudio, se han recopilado datos de tipo *boxscore* de partidos de temporada regular de NBA entre los cursos 2010-2011 y 2019-2020, tanto de estadísticas básicas como avanzadas. Estos han sido procesados, analizados y adecuados para elaborar diferentes conjuntos de estadísticas medias de los últimos partidos disputados por cada equipo, que se han utilizado para entrenar y evaluar distintos modelos de aprendizaje automático en busca del mejor resultado posible. Durante este proceso, se ha llegado a las siguientes conclusiones:

- Se han seguido los pasos apropiados para la realización de un trabajo de este tipo: utilizar una base de datos representativa (suficientemente grande), establecer una *baseline* como marca a mejorar, realizar un análisis exhaustivo centrado no sólo en los puros datos, si no también en el conocimiento del campo de investigación, probar varios conjuntos con distintos métodos de selección de características, y analizar y validar los resultados, utilizando métricas más allá de la *accuracy*, y probando además la robustez de estos.

- La mejor marca obtenida ha sido de una precisión del 68.41 %. Esta se ha conseguido mediante el modelo Light Gradient Boosting, cuyos hiperparámetros se especifican en el capítulo 6.
- El algoritmo de selección de características *Joint Mutual Information Maximization* ha resultado ser de mucha utilidad, mejorando los resultados base obtenidos. Con esto, se subraya la importancia de la reducción de la dimensionalidad para mejorar el rendimiento de los modelos.
- Desde el inicio del planteamiento del trabajo, se ha decidido no utilizar redes neuronales profundas, considerándose inapropiadas para el número de observaciones con las que se contaba. El perceptrón multicapa, que sí que ha sido probado, no ha destacado especialmente por su buen rendimiento.
- Los modelos que mejores resultados han obtenido de manera regular han sido: la máquina de vectores soporte, el Light Gradient Boosting y la regresión logística.
- Tras validar los mejores resultados, se ha comprobado que estos son robustos, siendo la máquina de vectores soporte el modelo más robusto para nuestra partición de datos, a pesar de no existir diferencias estadísticamente significativas en los resultados de los distintos modelos.
- El factor cancha juega un papel importante, tanto en la NBA como en este trabajo. El mayor número de victorias locales ha provocado que las clases estén desbalanceadas. Esto llevaba a la mayoría de nuestros modelos a rendir muy pobremente en las derrotas locales. Como en este campo, la métrica utilizada para evaluar los modelos es la *accuracy*, tenemos que aceptar esta limitación, ya que los modelos que funcionaban mejor en las derrotas presentaban peores valores de *accuracy*.
- Mediante el experimento de testeo en tiempo real de la temporada 2020-2021 (Apéndice C) se ha demostrado que se puede utilizar la inteligencia artificial y el aprendizaje automático para derrotar a las casas de apuestas, consiguiendo un retorno medio de aproximadamente un 42 % sobre la inversión inicial.

8.2. Futuros trabajos:

Como futuras vías de investigación, se plantean las siguientes:

- **Incorporación de nuevas características**: Una de las limitaciones de este trabajo puede haber sido el hecho de haber incorporado únicamente características basadas en estadística básicas. No se ha tenido

en cuenta información como lesiones, rachas de los equipos (más allá de lo que se ve reflejado en el ELO), si el partido tenía alguna implicación especial, etc. Un dato interesante a incluir podría haber sido las cuotas de casas de apuestas, ya que estas tienen en cuenta cualquier acontecimiento que suceda antes del partido, ya sean lesiones, estados recientes de forma o confianza que le dan los apostadores a cada equipo.

- **Balanceo de clases**: Ya se ha comentado el problema que supone tener las clases desbalanceadas. Por eso, podría ser interesante probar a realizar algún tipo de balanceo de clases, ya sea eliminar observaciones correspondientes a victorias locales o generar nuevas muestras de derrotas para ver si se consigue mejorar el rendimiento de los modelos, sobre todo para las derrotas del equipo local.
- **Incorporación de nuevas métricas de evaluación**: Como ya hemos comentado, el hecho de que la métrica para evaluar los modelos más utilizada en este campo sea la *accuracy*, limita mucho a la hora de intentar ajustar modelos que predigan mejor en las derrotas. De utilizarse otra métrica objetivo, como el *f1-score*, que no está tan influida por las clases desbalanceadas, se podría dar un enfoque novedoso e interesante al problema estudiado.
- **Predicción de estadísticas concretas**: Pese a que, previsiblemente, sería una tarea mucho más complicada, podría resultar muy interesante tratar de predecir, en vez del ganador de un partido, estadísticas concretas. Los puntos que meterá un equipo, las asistencias del base estrella, los minutos que va a jugar la joven promesa, se abren muchísimas puertas nuevas para investigar acerca de esta liga.
- **Investigación de otros mercados**: Aunque a veces parezca que no, el baloncesto también existe más allá de Norteamérica. Tanto las competiciones europeas como las ligas domésticas son mercados muy interesantes a investigar, no solo porque también generan mucho interés y tienen un elevado número de aficionados, sino también porque son ámbitos en los que se han realizado muchos menos estudios de este tipo. El siguiente paso lógico sería realizar este mismo trabajo para la Euroliga, la mayor competición de baloncesto europeo. Surgirían nuevos problemas, como el hecho de que, al no ser una liga completamente cerrada, cada temporada participan varios equipos distintos, o que no hay tantos datos y estadísticas disponibles como los hay para la NBA, pero sería, sin duda, un reto muy interesante.

Apéndice A

Glosario de estadísticas avanzadas

OFFRTG

- **Descripción**: Mide los puntos anotados por un equipo por 100 posesiones.
- **Fórmula**: $\frac{Puntos}{Posesiones} \cdot 100$

DEFRTG

- **Descripción**: Mide los puntos permitidos por un equipo por 100 posesiones.
- **Fórmula**: $\frac{Puntosoponente}{Posesionesoponente} \cdot 100$

NETRTG

- **Descripción**: Mide el diferencial de puntos por 100 posesiones.
- **Fórmula**: $OFFRTG - DEFRTG$

AST %

- **Descripción**: Mide la proporción de tiros anotados que fueron asistidos.
- **Fórmula**: $\frac{Asistencias}{Tirosanotados} \cdot 100$

AST/TO

- **Descripción**: Mide la ratio de asistencias sobre pérdidas de balón.
- **Fórmula**: $\frac{Asistencias}{Pérdidas}$

AST RATIO

- **Descripción**: Mide la ratio de asistencias por 100 posesiones.
- **Fórmula**: $\frac{Asistencias}{Posesiones} \cdot 100$

OREB %

- **Descripción**: Mide la proporción de rebotes ofensivos del equipo sobre el total de rebotes ofensivos disponibles.
- **Fórmula**: $\frac{Rebotes\ ofensivos\ equipo}{Rebotes\ ofensivos\ equipo\ +\ Rebotes\ defensivos\ oponente} \cdot 100$

DREB %

- **Descripción**: Mide la proporción de rebotes defensivos del equipo sobre el total de rebotes defensivos disponibles.
- **Fórmula**: $\frac{Rebotes\ defensivos\ equipo}{Rebotes\ defensivos\ equipo\ +\ Rebotes\ ofensivos\ oponente} \cdot 100$

REB %

- **Descripción**: Mide la proporción de rebotes totales del equipo sobre el total de rebotes disponibles.
- **Fórmula**: $\frac{Rebotes\ equipo}{Rebotes\ equipo\ +\ Rebotes\ oponente} \cdot 100$

TOV %

- **Descripción**: Mide la proporción de posesiones del equipo que acaban en pérdida de balón.
- **Fórmula**: $\frac{Pérdidas}{Posesiones} \cdot 100$

EFG %

- **Descripción**: Métrica de eficacia de lanzamiento con un factor corrector para incorporar el valor añadido del triple sobre el tiro de dos.
- **Fórmula**: $\frac{Tiros\ totales\ anotados\ +\ 0{,}5 \cdot Triples\ anotados}{Tiros\ totales\ intentados} \cdot 100$

TS %

- **Descripción**: Métrica de eficacia de lanzamiento con un factor corrector sobre los tiros de 3 y los tiros libres.
- **Fórmula**: $\frac{Puntos\ anotados}{2 \cdot (Tiros\ totales\ intentados\ +\ 0{,}44 \cdot Tiros\ libres\ intentados)} \cdot 100$

PACE

- **Descripción**: Cantidad de posesiones de las que dispone un equipo durante un partido.

PIE

- **Descripción**: Mide el impacto estimado de los jugadores del equipo con respecto a las estadísticas totales del encuentro.
- **Fórmula**: $\frac{Estadisticas\ totales\ equipo}{Estadisticas\ totales\ partido} \cdot 100$

  $Estadisticas\ totales\ equipo = PTS + TA + TLA - TI - TLI + REBD + 0{,}5 \cdot REBO + AST + ROB + 0{,}5 \cdot TAP - FAL - PER(Equipo)$

  $Estadisticas\ totales\ partido = PTS + TA + TLA - TI - TLI + REBD + 0{,}5 \cdot REBO + AST + ROB + 0{,}5 \cdot TAP - FAL - PER(Totales\ de\ ambos\ equipos)$

%FGA 2PT

- **Descripción**: Mide la proporción de tiros de dos intentados sobre el total de tiros intentados.
- **Fórmula**: $\frac{Tiros\ de\ dos\ intentados}{Tiros\ totales\ intentados} \cdot 100$

%FGA 3PT

- **Descripción**: Mide la proporción de tiros de tres intentados sobre el total de tiros intentados.
- **Fórmula**: $\frac{Triples\ intentados}{Tiros\ totales\ intentados} \cdot 100$

%PTS 2PT

- **Descripción**: Mide la proporción de puntos provenientes de tiros de dos sobre el total de puntos anotados.
- **Fórmula**: $\frac{Puntos\ de\ tiros\ de\ dos}{Puntos\ totales\ anotados} \cdot 100$

%PTS 2PT MR

- **Descripción**: Mide la proporción de puntos provenientes de tiros de media distancia sobre el total de puntos anotados.
- **Fórmula**: $\frac{Puntos\ de\ tiros\ de\ media\ distancia}{Puntos\ totales\ anotados} \cdot 100$

%PTS 3PT

- **Descripción**: Mide la proporción de puntos provenientes de tiros de tres sobre el total de puntos anotados.
- **Fórmula**: $\frac{Puntos\ de\ tiros\ de\ tres}{Puntos\ totales\ anotados} \cdot 100$

%PTS FBPS

- **Descripción**: Mide la proporción de puntos provenientes de contraataques sobre el total de puntos anotados.
- **Fórmula**: $\frac{Puntos\ en\ contraataque}{Puntos\ totales\ anotados} \cdot 100$

%PTS FT

- **Descripción**: Mide la proporción de puntos provenientes de tiros libres.
- **Fórmula**: $\frac{Puntos\ de\ tiros\ libres}{Puntos\ totales\ anotados} \cdot 100$

%PTS OFF TO

- **Descripción**: Mide la proporción de puntos anotados tras pérdida de balón del rival.
- **Fórmula**: $\frac{Puntos\ tras\ pérdida\ rival}{Puntos\ totales\ anotados} \cdot 100$

%PTS PITP

- **Descripción**: Mide la proporción de puntos anotados en la pintura (zona más cercana a la canasta).
- **Fórmula**: $\frac{Puntos\ en\ pintura}{Puntos\ totales\ anotados} \cdot 100$

2FGM %AST

- **Descripción**: Mide la proporción de tiros de dos anotados tras asistencia sobre el total de tiros de dos anotados.
- **Fórmula**: $\frac{Tiros\ de\ dos\ tras\ asistencia}{Tiros\ de\ dos\ anotados\ totales} \cdot 100$

2FGM %UAST

- **Descripción**: Mide la proporción de tiros de dos anotados sin asistencia sobre el total de tiros de dos anotados.
- **Fórmula**: $\frac{Tiros\ de\ dos\ sin\ asistencia}{Tiros\ de\ dos\ anotados\ totales} \cdot 100$

3FGM %AST

- **Descripción**: Mide la proporción de tiros de tres anotados tras asistencia sobre el total de tiros de tres anotados.
- **Fórmula**: $\frac{Tiros\ de\ tres\ tras\ asistencia}{Tiros\ de\ tres\ anotados\ totales} \cdot 100$

3FGM %UAST

- **Descripción**: Mide la proporción de tiros de tres anotados sin asistencia sobre el total de tiros de tres anotados.
- **Fórmula**: $\frac{Tiros\ de\ tres\ sin\ asistencia}{Tiros\ de\ tres\ anotados\ totales} \cdot 100$

FGM %AST

- **Descripción**: Mide la proporción de tiros totales anotados tras asistencia sobre el total de tiros anotados.
- **Fórmula**: $\frac{Tiros\ anotados\ tras\ asistencia}{Tiros\ anotados\ totales} \cdot 100$

FGM %UAST

- **Descripción**: Mide la proporción de tiros totales anotados sin asistencia sobre el total de tiros anotados.
- **Fórmula**: $\frac{Tiros\ anotados\ sin\ asistencia}{Tiros\ anotados\ totales} \cdot 100$

Apéndice B

Resultados completos de los modelos

Tabla B.1: Precisión de los modelos en el conjunto manual.

Modelo	Número de partidos utilizados para calcular la media									
	1	2	3	4	5	6	7	8	9	10
Regresión Logística	66.51	66.64	66.54	66.40	66.64	66.27	66.51	66.75	66.61	66.54
Reg. Log. (stepwise)	66.30	66.81	66.27	65.93	66.23	66.68	66.47	66.51	66.20	66.54
Random Forest	65.98	65.78	65.67	66.77	66.12	66.22	67.08	66.70	66.46	66.22
Random Forest (variables + imp.)	66.39	66.22	66.12	66.39	66.39	66.91	67.18	67.08	66.83	66.91
SVM	66.56	66.60	66.29	66.46	66.53	67.28	66.63	66.73	67.04	67.28
kNN	65.71	66.15	65.61	66.05	66.36	66.19	66.15	65.37	66.09	66.19
AdaBoost	66.53	66.94	67.08	67.04	67.21	67.62	67.32	66.91	67.08	67.62
MLP	66.83	66.12	66.73	67.04	66.43	67.42	66.97	66.63	67.01	67.42
LightGBM	66.84	66.80	67.04	66.80	67.07	66.70	66.97	66.94	66.94	66.70
ExtremeGB	66.87	65.92	66.94	66.80	66.43	66.63	66.63	65.85	66.56	66.63

Fuente: elaboración propia

Tabla B.2: Aciertos de los modelos en el conjunto manual.

Modelo	Número de partidos utilizados para calcular la media									
	1	2	3	4	5	6	7	8	9	10
Regresión Logística	1948	1952	1949	1945	1952	1941	1948	1955	1951	1949
Reg. Log. (stepwise)	1942	1957	1941	1931	1940	1953	1947	1948	1939	1949
Random Forest	1932	1926	1923	1955	1936	1939	1964	1953	1948	1939
Random Forest (variables + imp.)	1944	1939	1936	1944	1944	1959	1967	1964	1957	1959
SVM	1949	1950	1941	1946	1948	1970	1951	1954	1963	1970
kNN	1924	1937	1921	1934	1943	1938	1937	1923	1935	1938
AdaBoost	1948	1960	1964	1963	1968	1980	1971	1959	1964	1980
MLP	1957	1936	1954	1963	1945	1974	1961	1951	1962	1974
LightGBM	1957	1956	1963	1956	1964	1953	1961	1960	1960	1953
ExtremeGB	1958	1930	1960	1956	1955	1951	1951	1928	1949	1951

Fuente: elaboración propia

Tabla B.3: Precisión de los modelos en el conjunto automático de 81 variables.

	Número de partidos utilizados para calcular la media									
Modelo	1	2	3	4	5	6	7	8	9	10
Regresión Logística	67.28	67.01	67.01	67.04	66.87	67.01	67.38	67.45	66.94	67.28
Reg. Log. (stepwise)	67.52	66.97	67.25	67.25	66.94	67.62	67.69	67.45	67.45	66.63
Random Forest	66.19	66.63	66.70	66.60	66.02	66.15	66.19	67.28	66.33	66.09
Random Forest (variables + imp.)	66.22	66.36	67.32	66.94	65.92	66.29	67.25	66.70	65.98	65.98
SVM	66.97	67.38	67.35	67.25	67.01	67.35	67.35	67.28	67.32	67.73
kNN	67.11	66.80	66.94	67.32	67.25	67.11	67.25	66.73	67.11	66.77
AdaBoost	66.70	66.73	67.25	67.11	67.18	67.11	67.28	67.34	67.32	67.52
MLP	67.35	67.01	67.18	67.28	66.26	66.91	66.02	67.14	67.25	66.50
LightGBM	66.94	67.01	67.18	66.87	66.80	66.56	67.18	67.28	66.80	66.60
ExtremeGB	66.94	65.81	66.50	66.73	66.73	66.97	67.32	67.04	66.80	66.63

Fuente: elaboración propia

Tabla B.4: Aciertos de los modelos en el conjunto automático de 81 variables.

	Número de partidos utilizados para calcular la media									
Modelo	1	2	3	4	5	6	7	8	9	10
Regresión Logística	1970	1962	1962	1963	1958	1962	1973	1975	1960	1970
Reg. Log. (stepwise)	1977	1961	1969	1969	1960	1980	1982	1975	1975	1951
Random Forest	1938	1951	1953	1950	1933	1937	1938	1970	1942	1935
Random Forest (variables + imp.)	1939	1943	1971	1960	1930	1941	1969	1953	1932	1932
SVM	1961	1973	1972	1969	1962	1972	1972	1970	1971	1983
kNN	1965	1956	1960	1971	1969	1965	1969	1954	1965	1955
AdaBoost	1953	1954	1969	1965	1967	1965	1970	1972	1971	1977
MLP	1972	1962	1967	1970	1940	1959	1933	1966	1969	1947
LightGBM	1960	1962	1967	1958	1956	1949	1967	1970	1956	1950
ExtremeGB	1960	1927	1947	1954	1954	1961	1971	1963	1956	1942

Fuente: elaboración propia

Tabla B.5: Precisión de los modelos en el conjunto automático de 29 variables.

	Número de partidos utilizados para calcular la media									
Modelo	1	2	3	4	5	6	7	8	9	10
Regresión Logística	67.86	67.32	67.38	67.04	67.18	67.04	67.28	67.14	67.08	67.42
Reg. Log. (stepwise)	67.73	67.21	67.52	66.94	67.35	66.94	66.87	67.18	67.42	67.14
Random Forest	66.43	66.50	67.42	66.80	66.60	66.80	66.80	67.04	66.56	66.22
Random Forest (variables + imp.)	66.63	66.91	66.84	67.25	66.91	66.77	66.94	67.66	67.62	66.33
SVM	67.11	67.21	67.21	67.18	67.11	67.21	67.14	67.25	67.35	67.52
kNN	66.77	67.01	66.80	66.97	67.18	66.70	67.28	67.08	66.80	67.79
AdaBoost	66.97	66.80	67.14	67.28	67.08	67.52	67.73	67.28	67.18	67.18
MLP	67.28	66.87	67.18	67.42	67.52	66.63	66.60	67.49	67.83	67.01
LightGBM	66.87	66.94	67.11	67.04	67.00	66.73	66.77	67.14	66.50	65.50
ExtremeGB	66.29	67.18	67.11	67.38	66.97	67.38	67.08	66.73	67.11	66.29

Fuente: elaboración propia

Tabla B.6: Aciertos de los modelos en el conjunto automático de 29 variables.

	Número de partidos utilizados para calcular la media									
Modelo	1	2	3	4	5	6	7	8	9	10
Regresión Logística	1987	1971	1973	1963	1967	1963	1970	1966	1964	1974
Reg. Log. (stepwise)	1983	1968	1977	1960	1972	1960	1958	1967	1974	1966
Random Forest	1945	1947	1974	1956	1950	1956	1956	1963	1949	1939
Random Forest (variables + imp.)	1951	1959	1957	1969	1959	1955	1960	1981	1980	1942
SVM	1965	1968	1968	1967	1965	1968	1966	1969	1972	1977
kNN	1955	1962	1956	1961	1967	1953	1970	1964	1956	1985
AdaBoost	1961	1956	1966	1970	1964	1977	1983	1970	1967	1967
MLP	1970	1958	1967	1974	1977	1951	1950	1976	1986	1962
LightGBM	1958	1960	1965	1963	1962	1954	1961	1966	1947	1918
ExtremeGB	1941	1967	1965	1973	1961	1973	1964	1954	1965	1941

Fuente: elaboración propia

Tabla B.7: Precisión de los modelos en el conjunto automático de 14 variables.

	Número de partidos utilizados para calcular la media									
Modelo	1	2	3	4	5	6	7	8	9	10
Regresión Logística	67.49	66.97	67.04	67.42	67.59	67.55	67.38	67.32	67.66	67.90
Reg. Log. (stepwise)	67.59	67.25	67.35	67.42	67.62	67.45	67.38	67.38	67.59	67.52
Random Forest	68.03	67.70	67.18	66.73	67.14	65.78	66.91	66.87	66.70	65.78
Random Forest (variables + imp.)	66.29	66.46	67.14	67.21	67.49	67.14	66.94	67.59	67.28	65.40
SVM	67.11	67.21	67.25	67.14	67.28	67.11	67.14	67.21	67.21	67.66
kNN	67.14	66.77	67.11	67.55	66.94	66.84	66.73	67.35	67.01	67.55
AdaBoost	67.04	67.01	67.18	67.25	67.01	67.32	67.08	67.25	67.83	66.91
MLP	66.53	67.14	67.25	66.94	67.32	67.25	67.35	67.45	67.28	67.32
LightGBM	67.04	66.97	67.08	67.73	67.25	67.28	66.91	67.42	67.11	66.43
ExtremeGB	67.62	67.01	67.04	67.04	67.35	67.42	67.55	67.73	67.01	66.29

Fuente: elaboración propia

Tabla B.8: Aciertos de los modelos en el conjunto automático de 14 variables.

	Número de partidos utilizados para calcular la media									
Modelo	1	2	3	4	5	6	7	8	9	10
Regresión Logística	1976	1961	1963	1974	1979	1978	1973	1971	1981	1988
Reg. Log. (stepwise)	1979	1969	1972	1974	1980	1975	1973	1973	1979	1977
Random Forest	1992	1953	1967	1954	1966	1926	1959	1958	1953	1926
Random Forest (variables + imp.)	1941	1946	1966	1968	1976	1966	1960	1979	1970	1915
SVM	1965	1968	1969	1966	1970	1965	1966	1968	1968	1981
kNN	1966	1955	1965	1978	1960	1957	1954	1972	1962	1978
AdaBoost	1963	1962	1967	1969	1962	1971	1964	1969	1986	1959
MLP	1948	1966	1969	1960	1971	1969	1972	1975	1970	1971
LightGBM	1963	1961	1964	1983	1969	1970	1959	1974	1965	1945
ExtremeGB	1980	1962	1963	1963	1972	1974	1978	1983	1962	1941

Fuente: elaboración propia

Tabla B.9: Precisión de los modelos en el conjunto automático de 10 variables.

	Número de partidos utilizados para calcular la media									
Modelo	1	2	3	4	5	6	7	8	9	10
Regresión Logística	67.25	66.94	67.14	67.11	67.32	67.49	67.32	67.25	67.11	67.73
Reg. Log. (stepwise)	67.14	67.25	67.35	67.21	67.14	67.14	67.21	67.14	67.11	67.52
Random Forest	67.00	66.70	66.53	67.49	67.32	67.35	67.21	67.14	67.18	66.19
Random Forest (variables + imp.)	67.14	66.77	66.46	66.19	67.11	67.32	66.97	67.07	66.94	67.32
SVM	67.21	67.21	67.21	67.08	67.11	67.11	67.11	67.11	67.14	67.49
kNN	66.43	66.87	67.08	66.91	67.18	67.42	67.08	67.14	67.21	67.41
AdaBoost	67.35	67.28	67.18	67.89	67.28	67.25	67.28	66.84	67.28	66.84
MLP	67.21	66.97	67.45	67.11	67.49	66.94	67.14	67.11	67.11	67.66
LightGBM	67.01	66.26	66.77	66.73	67.59	66.84	67.25	67.04	67.04	66.50
ExtremeGB	66.33	66.33	66.80	67.25	67.32	66.05	67.38	67.42	67.66	66.15

Fuente: elaboración propia

Tabla B.10: Aciertos de los modelos en el conjunto automático de 10 variables.

	Número de partidos utilizados para calcular la media									
Modelo	1	2	3	4	5	6	7	8	9	10
Regresión Logística	1969	1960	1966	1965	1971	1976	1971	1969	1965	1983
Reg. Log. (stepwise)	1966	1969	1972	1968	1966	1966	1968	1966	1965	1977
Random Forest	1962	1953	1948	1976	1971	1972	1968	1966	1967	1938
Random Forest (variables + imp.)	1966	1955	1946	1938	1965	1971	1961	1964	1960	1971
SVM	1968	1968	1968	1964	1965	1965	1965	1965	1966	1976
kNN	1945	1958	1964	1959	1967	1974	1964	1966	1968	1974
AdaBoost	1972	1970	1967	1987	1970	1969	1970	1957	1970	1956
MLP	1968	1961	1975	1965	1976	1960	1966	1965	1965	1981
LightGBM	1962	1940	1955	1954	1979	1957	1969	1963	1963	1947
ExtremeGB	1942	1942	1956	1969	1971	1934	1973	1974	1981	1937

Fuente: elaboración propia

Tabla B.11: Precisión de los modelos en el conjunto de diferencias de 14 variables.

	Número de partidos utilizados para calcular la media									
Modelo	1	2	3	4	5	6	7	8	9	10
Regresión Logística	66.91	66.91	66.87	66.67	66.77	67.04	66.84	66.97	66.80	67.34
Reg. Log. (stepwise)	66.63	66.50	66.91	66.50	66.70	66.80	66.97	67.28	66.73	67.28
Random Forest	66.63	66.53	66.77	66.46	65.68	65.98	66.70	66.91	66.70	67.38
Random Forest (variables + imp.)	66.84	66.33	66.33	66.67	66.05	66.63	66.56	66.70	66.77	67.73
SVM	67.08	66.87	67.04	67.01	67.04	67.04	67.18	67.18	67.14	67.08
kNN	66.63	66.77	66.77	66.60	66.09	66.29	66.56	66.15	66.70	67.66
AdaBoost	66.12	65.54	65.30	66.02	65.47	65.57	65.64	65.37	65.23	67.11
MLP	65.92	66.56	66.70	66.26	66.73	66.77	65.37	66.84	66.33	67.35
LightGBM	67.25	66.56	66.73	66.97	66.87	66.39	66.46	66.33	67.14	67.73
ExtremeGB	66.67	66.60	66.46	67.18	66.36	66.77	66.39	67.04	66.43	67.32

Fuente: elaboración propia

Tabla B.12: Aciertos de los modelos en el conjunto de diferencias de 14 variables.

	Número de partidos utilizados para calcular la media									
Modelo	1	2	3	4	5	6	7	8	9	10
Regresión Logística	1959	1959	1958	1952	1955	1963	1957	1961	1956	1972
Reg. Log. (stepwise)	1951	1947	1959	1947	1953	1956	1961	1970	1954	1970
Random Forest	1951	1948	1955	1946	1923	1932	1953	1959	1953	1973
Random Forest (variables + imp.)	1957	1942	1942	1952	1934	1951	1949	1953	1955	1983
SVM	1964	1958	1963	1962	1963	1963	1967	1967	1966	1964
kNN	1951	1955	1955	1950	1935	1941	1949	1937	1953	1981
AdaBoost	1936	1919	1912	1933	1917	1920	1922	1914	1910	1965
MLP	1930	1949	1953	1940	1954	1955	1914	1957	1942	1972
LightGBM	1969	1949	1954	1961	1958	1944	1946	1942	1966	1983
ExtremeGB	1952	1950	1946	1967	1943	1955	1944	1963	1945	1971

Fuente: elaboración propia

Apéndice C

Test en tiempo real, temporada 2020-2021

Introducción

Una de las primeras ideas al empezar este estudio fue probar si realmente este trabajo podía servir para predecir sobre partidos que aún no habían sido disputados. Es decir, en vez de tener una base de datos de partidos pasados, dividirla en conjunto de entrenamiento y conjunto de test y ajustar los modelos y hacer predicciones sobre estos conjuntos, utilizar los partidos que se habían disputado hasta la fecha en la presente temporada para entrenar y predecir sobre los partidos que aún no habían sido disputados. En vez de hacer una división aleatoria de entrenamiento/test, hacer esta partición de manera cronológica.

Así pues, este experimento dio comienzo el 31 de marzo de 2021. Hasta entonces, se habían disputado 695 partidos de los 1080 totales de la temporada, es decir, aproximadamente un 65 %. Cabe destacar que la temporada 2020-2021 vio su calendario reducido a causa de la pandemia por COVID-19.

Los datos

Con respecto a los datos, se utilizaron solamente las estadísticas básicas obtenidas mediante el paquete `nbastatR` [3], a las que se le aplicó una media móvil de los últimos diez partidos. Añadiendo el Rating ELO, las dimensiones del conjunto de entrenamiento fueron de 695 observaciones de 48 variables, 49 contando la variable respuesta. Sobre el conjunto de test, cabe destacar que se tenía que ir creando a diario con los partidos a disputar esa noche, ya que de no haber sido así, no se habrían tenido en cuenta las estadísticas actualizadas, habiendo sido las mismas para todos los partidos de cada equipo dentro de este conjunto. Mediante esta actualización diaria, se tenían en cuenta todos los partidos para la creación del conjunto de test, no

solo los partidos anteriores al 31 de marzo.

Los modelos

En cuanto a los modelos, se realizó una prueba similar con los datos de la temporada 2018-2019[1] y se concluyó que los dos modelos que mejor funcionaban eran la regresión logística y la máquina de vectores soporte con kernel lineal. Así pues, estos fueron los modelos utilizados en el experimento.

Además, para añadir una forma distinta de evaluación de los modelos, se comprobó si se podían utilizar para vencer a las casas de apuestas. Para ello, se comenzó con 10 000 puntos[2] y se elaboraron dos estrategias de apuesta. Con la regresión logística, se utilizaron las probabilidades esperadas de victoria que se otorgaba a cada equipo para apostar una cantidad proporcional a la confianza del modelo en los equipos. Así pues, para probabilidades de victoria entre el 50 y el 62.5 %, se apostaban 100 puntos. Para probabilidades de victoria entre el 62.5 y el 75 %, 200 puntos, y 300 puntos para probabilidades superiores al 75 %. Para la máquina de vectores soporte, se utilizó un método lineal en el que se apostaba 150 puntos a todas las predicciones del modelo, independientemente de las probabilidades. Las cuotas se incorporaban diariamente de manera manual, a la vez que se actualizaban los modelos, y se obtenían de las páginas web de resultados deportivos FlashScore y Sofascore.

Resultados

Los resultados fueron incluso mejor de lo esperado. Ambos modelos consiguieron superar la marca obtenida en el estudio principal, consiguiendo la regresión logística una precisión del 68.57 % y la máquina de vectores soporte un impresionante precisión del 72.47 %. Curiosamente, fue el modelo que peor funcionó el que más puntos obtuvo, consiguiendo unas ganancias netas de 4721 puntos frente a los 3786 puntos obtenidos por el modelo de precisión superior, lo que subraya la importancia de tener una buena estrategia de apuestas. En cualquier caso, retornos sobre inversión del 47.21 y 37.86 % en escasos dos meses y medio son cifras muy superiores a las de cualquier inversión al uso. El hecho de haber obtenido precisiones tan altas puede haber sido por las siguientes razones.

- La *upset ratio*, la proporción de partidos ganados por el equipo no favorito, ha sido especialmente bajo durante los partidos del conjunto de test. A largo plazo, esta ratio suele tomar valores de entre un 31 y

[1] Se utilizó esta temporada en vez de la 2019-2020 ya que el transcurso de esta se vio interrumpido por el COVID-19

[2] En ningún momento se ha utilizado esto para apostar dinero real, esta prueba se ha realizado únicamente por fines experimentales, no apoyo a las casas de apuestas de ninguna manera.

32 %, mientras que en el periodo de evaluación de los modelos, esta ratio fue del 29.35 %, es decir, los favoritos han ganado partidos más regularmente, haciendo que los resultados fueran algo más previsibles.

- Haber utilizado sólo una temporada para el experimento y haber hecho la división de conjunto de entrenamiento y conjunto de test de manera cronológica en lugar de aleatoria ha podido ser diferencial para obtener esta precisión superior ya que, de haber existido tendencias favorables o desfavorables en los equipos durante el periodo considerado para el conjunto de entrenamiento, estas se han visto reflejadas en los datos con los que se han entrenado los modelos. Además, al utilizar solo una temporada hemos logrado evitar grandes cambios de nivel de los equipos debido a incorporación o despedida de jugadores importantes durante la postemporada.
- El cambio en el sistema de clasificación para Playoffs provocó un descenso en el número de partidos sin importancia al final de la temporada. Hasta la temporada 2019-2020, los 8 primeros clasificados de cada conferencia disputaban los Playoffs. Esto provocaba que, al final de la temporada, tanto equipos que ya tenían su puesto en el torneo asegurado, como equipos matemáticamente eliminados, tendían a restarle importancia al partido, dando descanso a jugadores importantes o incluso haciendo lo posible para perder el partido[3]. Desde entonces, se ha instaurado un torneo clasificatorio, o *Play-in tournament*, en el que los equipos en 7º, 8º, 9º y 10º lugar de cada conferencia se baten para obtener dos puestos en Playoffs, mientras que del primero al sexto de cada conferencia se clasifican automáticamente. Este cambio en la estructura ha provocado que, hasta el final de la temporada haya habido partidos importantes, ya sea para conseguir ascender en la clasificación, o para intentar llegar al torneo clasificatorio, haciendo estos últimos partidos de la temporada ligeramente menos impredecibles. Esta también es una de las razones por las que la *upset ratio* ha sido tan baja durante estos partidos.

En la Tabla C.1 podemos observar un breve resumen de los resultados. Para más detalles, se realizó un cuadro de mando para dar soporte al experimento, que se puede encontrar en el siguiente link:

`https://7antoniosegovia.shinyapps.io/NBA_predictions/`

[3]Este fenómeno, conocido como *tanking*, se produce cuando equipos matemáticamente eliminados de Playoffs intentan perder los partidos para caer en la clasificación, ya que cuanto peor sea el resultado del equipo, más probabilidades tiene de conseguir una elección superior en el *Draft* del verano siguiente.

Tabla C.1: Resumen de los resultados del experimento.

Modelo	Precisión	Ganancias	Máxima cuota acertada	% Apuestas a no favorito (Precisión)
Regresión Logística	68.57	4721 ptos.	4.2	17.92 (46.38)
SVM	72.47	3786 ptos.	3.4	10.65 (60.98)

Fuente: elaboración propia

Conclusiones

Considerando que fue uno de los primeros experimentos que llevé a cabo cuando comencé a investigar este campo, a la vista está que los resultados superan cualquier expectativa. Además, esta prueba sirvió para descubrir el potencial en este ámbito de la regresión logística y la máquina de vectores soporte, cuyo rendimiento ha estado entre los mejores durante todo el estudio. También se ha demostrado que se puede batir a las casas de apuestas con el Machine Learning y la inteligencia artificial, dándole una aplicación real al experimento. Como último apunte, destacar que me sirvió de empujón para acabar de decidirme a realizar este trabajo, y recordar que el experimento se puede encontrar en el siguiente link:

`https://7antoniosegovia.shinyapps.io/NBA_predictions/`

Bibliografía

[1] Renato Amorim Torres. Prediction of NBA games based on Machine Learning methods. 2013.

[2] Mohamed Bennasar, Yulia Hicks, and Rossitza Setchi. Feature selection using joint mutual information maximisation. *Expert Systems with Applications*, 42(22):8520–8532, 2015.

[3] Alex Bresler. *nbastatR: R's interface to NBA data*, 2021. R package version 0.1.1505.

[4] Chenjie Cao. Sports Data Mining Technology Used in Basketball Outcome Prediction. 2012.

[5] Ge Cheng, Zhenyu Zhang, Moses Ntanda Kyebambe, and Nasser Kimbugwe. Predicting the Outcome of NBA Playoffs Based on the Maximum Entropy Principle. *Entropy*, 18(12), 2016.

[6] Sara Cuesta Torrado. Carolina Marín y el poder del 'big data' en el deporte. *El País Semanal*. Consultado el 10 de junio de 2021, en https://elpais.com/eps/2021-04-04/carolina-marin-la-reina-del-big-data.html.

[7] Reuben Fischer-Baum and Nate Silver. How we calculate NBA ELO Ratings. *FiveThirtyEight*. Consultado el 25 de marzo de 2021, en https://fivethirtyeight.com/features/how-we-calculate-nba-elo-ratings/.

[8] John Harrison. *RSelenium: R Bindings for 'Selenium WebDriver'*, 2020. R package version 1.7.7.

[9] Jackson Joffe, Jack Rosener, and Josh Weiner. Predicting the outcome of NBA games with Machine Learning. *Towards Data Science*. Con-

sultado el 25 de marzo de 2021, en https://towardsdatascience.com/predicting-the-outcome-of-nba-games-with-machine-learning-a810bb768f20.

[10] Eric Scot Jones. Predicting outcomes of NBA basketball games. B.S. thesis, 2016.

[11] Miron B. Kursa. *praznik: Tools for Information-Based Feature Selection*, 2020. R package version 8.0.0.

[12] Nachi Lieder. Can Machine-Learning Methods Predict the Outcome of an NBA Game? 2018.

[13] Redacción Marca. El Manchester City se refuerza con ¡astrofísicos! *Diario Marca*. Consultado el 10 de junio de 2021, en https://www.marca.com/futbol/premier-league/2021/03/23/6059a2cc268e3ea4758b460a.html.

[14] Dragan Miljković, Ljubiša Gajić, Aleksandar Kovačević, and Zora Konjović. The use of data mining for basketball matches outcomes prediction. In *IEEE 8th International Symposium on Intelligent Systems and Informatics*, pages 309–312, 2010.

[15] Jorge Morate Vázquez. Predicción de equipo ganador en el baloncesto. B.S. thesis, 2016.

[16] John Neter, Michael H Kutner, Christopher J Nachtsheim, William Wasserman, et al. Applied linear statistical models. 1996.

[17] Alberto Pérez Sierra. Kevin de Bruyne lidera la revolución de los contratos. *Diario As*. Consultado el 10 de junio de 2021, en https://as.com/futbol/2021/04/17/internacional/1618647579_178025.html.

[18] NBA Stats. Nba Advanced Stats, 2021. Datos obtenidos de varios enlaces dentro de la página, último acceso el 4 de junio de 2021, en https://www.nba.com/stats/.

[19] Fadi Thabtah, Li Zhang, and Neda Abdelhamid. Nba game result prediction using feature analysis and machine learning. *Annals of Data Science*, 6(1):103–116, 2019.

[20] Albert Villar Ortiz and Ramon Baldrich i Caselles. Machine Learning para la predicción de eventos en la NBA.