Escuela Superior
de Ingeniería y Tecnología
Universidad de La Laguna

Trabajo de Fin de Grado

Grado en Ingeniería Informática

Aprendizaje automático en la predicción de ganadores en partidos deportivos

Machine learning in the predicting of winners in sports matches

La Laguna, 04 de junio de 2019

Dª. **Rosa María Aguilar Chinea**, con N.I.F 43.778.956-C profesora Titular de Universidad adscrita al Departamento de Ingeniería de Sistemas y Automática de la Universidad de La Laguna, como tutora.

D. **Jesús Miguel Torres Jorge**, con N.I.F 43.826.207-Y profesor Titular de Universidad adscrito al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como cotutor.

C E R T I F I C A (N)

Que la presente memoria titulada:
"Aprendizaje automático en la predicción de ganadores en partidos deportivos"

ha sido realizada bajo su dirección por D. **Daniel Rodríguez Martín**,
con N.I.F 42.417.921-X.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 04 de junio de 2019.

Agradecimientos

En primer lugar, me gustaría agradecer a mis padres y al resto de familiares por su ayuda durante toda la carrera. Siempre me mostraron su apoyo, su refuerzo y se han esforzado cada día para permitirme obtener la mejor formación posible.

También quiero agradecer a mi tutora Rosa María Aguilar Chinea por guiarme en el desarrollo de este proyecto.

Agradecer a mis amigos y compañeros por apoyarme en todo momento, tanto en las buenas como en las malas.

Por último, a Marta, por todo.

Licencia

Resumen

El deporte constituye actualmente una parte fundamental en la rutina diaria de la mayoría de las personas. Mientras que en Europa el deporte más popular es el fútbol, en los Estados Unidos predominan el béisbol, el fútbol americano y el baloncesto. Durante muchos años, el baloncesto norteamericano se ha considerado el mejor del mundo y, con ello, su máxima competición: la NBA (National Basketball Association).

El objetivo principal de este proyecto es poder predecir y clasificar correctamente qué equipo de la NBA ganará un partido en concreto en la temporada actual 2018/19. Desarrollaremos un sistema basado en Python [1] que recopila los datos de una base de datos abierta y los analiza con diferentes bibliotecas de código abierto. Estas predicciones se realizarían a partir del análisis histórico de datos potencialmente relevantes, lo que permitiría obtener patrones susceptibles de repetirse a lo largo del tiempo. Para evaluar los resultados obtenidos y extraer conclusiones se utilizarán dos técnicas de aprendizaje automatizado (DecisionTree y RandomForest). Además se realizará diversos experimentos adicionales para poder realizar comparaciones. Se llevará a cabo un estudio sobre la importancia de las cuotas de las casas de apuestas en el modelo y se utilizará el mejor clasificador para probar el sistema sobre un conjunto de partidos específico. Para ello se apostará una cantidad de dinero de manera virtual y se comprobará la rentabilidad obtenida en las casas de apuestas con las predicciones del clasificador.

Palabras clave: Deporte, Baloncesto, NBA, Python, Aprendizaje Automático, DecisionTree, RandomForest, Predicción.

Abstract

Sport is an essential part of the daily routine for most of our contemporaries. While soccer is the most popular sport in Europe, baseball, football and basketball predominate in the United States. For many years, American basketball has been considered the best in the world and, with it, its highest ranked championship: the NBA (National Basketball Association).

The main objective of this project is to correctly predict and rank which NBA team will win a given match in the current 2018/19 season. We will develop a Python-based system [1] that collects data from an open database and analyzes it with different open source libraries. These predictions would be made on the basis of historical analysis of potentially relevant data. This would allow guessing patterns with the potential be repeated over time and, therefore, predicted. Two automated learning techniques (DecisionTree and RandomForest) will be used to evaluate the results and draw conclusions. In addition, several additional experiments will be carried out in order to make comparisons. Further insight on the importance of the odds of the bookmakers in the model will be provided and the best classifier will be used to test the system on a specific set of matches. For this purpose, a virtual amount of money will be bet and the profitability obtained in the bookmakers will be compared with the predictions obtained by the classifier

Keywords: Sports, Basketball, NBA, Python, Machine Learning, DecisionTree, RandomForest, Predictions.

Índice general

CAPÍTULO 1 Antecedentes y estado del arte 1
1.1 Baloncesto - NBA 1
1.2 La NBA en las apuestas deportivas 4
1.2.1 Orígenes 4
1.2.2 ¿Cómo funcionan las cuotas? 4
1.2.3 Formato de cuotas 5
1.3 Big Data 6
1.3.1 ¿Qué es Big Data? 6
1.3.2 La importancia de Big Data 6
1.4 Data Mining (minería de datos) 7
CAPÍTULO 2 Objetivos 7
2.1 Objetivos principales 7
2.2 Objetivos específicos 8
2.3 Otros objetivos 8
CAPÍTULO 3 Fases y desarrollo del proyecto 8
3.1 Metodología 8
3.2 Herramientas y lenguaje utilizado 10
3.3 Datos 11
3.3.1 Estructura de los ficheros 11
3.3.1.1 Dataset principal 11
3.3.1.2 Otros datasets utilizados 11
3.4 Primeros pasos 12
3.4.1 Librerías 12
3.4.2 Datasets 12
3.4.3 Experimentación previa 13
3.4.4 Atributos 14
3.4.5 Selección de atributos 16
3.5 Aprendizaje automático 19
3.5.1 Métodos utilizados 19
3.5.1.1 DecisionTree (Árbol de decisión) 19
3.5.1.2 Random Forest (Bosque) 20
3.5.2 Normalización de datos 21
CAPÍTULO 4 Resultados y evaluación de los modelos 21

4.1 Evaluación de resultados 21
4.2 Parámetros de los modelos 23
4.2.1 Parámetros más óptimos elegidos por GridSearchCV para el Random Forest 24
4.3 Experimentación adicional 24
4.3.1 Estudio tasa de acierto a lo largo del tiempo 24
4.3.2 Evolución del RandomForest en comparación al número de árboles 25
4.3.3 Importancia de las cuotas de las casas de apuestas 26
4.3.4 Rentabilidad de los resultados obtenidos en las casas de apuestas 28
CAPÍTULO 5 Página web 29
5.1 Herramientas y lenguajes utilizados 29
5.2 Ficheros y funcionamiento 29
CAPÍTULO 6 Conclusiones 31
CAPÍTULO 7 Líneas futuras 32
CAPÍTULO 8 Bibliografía 34

Índice de figuras

Figura 1: Mercado de apuestas para Toronto Raptors - Milwaukee Bucks **5**

Figura 2: Metodología KDD **9**

Figura 3: Logo Jupyter Notebook **10**

Figura 4: Comparativa porcentaje de ganadores **14**

Figura 5: Atributos **14**

Figura 6: Comparativa atributos DecisionTree vs Forest **19**

Figura 7: Ejemplo de árbol de decisión **20**

Figura 8: Ejemplo de RandomForest **20**

Figura 9: Equipos antes **21**

Figura 10: Equipos después **21**

Figura 11: Ejemplo de código utilizando GridSearchCV para la inicialización de un modelo **24**

Figura 12: Mejores parámetros para el RandomForest **24**

Figura 13: Evolución forest con el paso del tiempo **25**

Figura 14: Evolución Random Forest por número de árboles **26**

Figura 15: Importancia cuotas casas de apuestas **27**

Figura 16: Interfaz Web **30**

Índice de tablas

Tabla 1: Conferencia Este 2

Tabla 2: Conferencia Oeste 3

Tabla 3: Fragmento dataset principal del proyecto 11

Tabla 4: Porcentaje importancia atributos 18

Tabla 5: Comparativa resultados 22

Tabla 6: Resultados con cuotas vs sin cuotas 28

Tabla 7: Rentabilidad en casas de apuestas 28

CAPÍTULO 1 Antecedentes y estado del arte

1.1 Baloncesto - NBA

El baloncesto [4] es un deporte de equipo, jugado entre dos conjuntos de cinco jugadores cada uno durante cuatro períodos o cuartos que oscilan entre diez y doce minutos dependiendo de la competición. Estos equipos compiten por anotar una mayor cantidad de puntos que su rival introduciendo un balón en una canasta situada a 3,05 metros del suelo. La puntuación de cada canasta es de dos o tres puntos, dependiendo de la posición desde la que se efectúa el lanzamiento, o de uno, si se trata de un tiro libre producido por una falta de un jugador contrario.

En este deporte, a diferencia del fútbol, balonmano, fútbol americano, etc., no existe el empate, puesto que si al terminar el tiempo reglamentario el marcador de ambos equipos está igualado, se jugarán prórrogas de cinco minutos cada una hasta que uno de los dos equipos consiga una puntuación mayor que su rival. Como única excepción a esta regla tenemos los partidos a ida y vuelta, en ellos puede haber empate en uno de los partidos, puesto que el vencedor se decide por el resultado combinado de ambos.

La National Basketball Association [5], más conocida por las siglas NBA (en español, Asociación Nacional de Baloncesto) es la principal liga de baloncesto profesional que se disputa en Estados Unidos desde 1946, añadiéndose posteriormente Canadá en los años 1990. Actualmente se compone de 30 equipos, los cuales 29 se encuentran en Estados Unidos y solamente uno de ellos pertenece a Canadá. La organización actual de la liga divide a los equipos en dos conferencias (Este y Oeste) [6], de tres divisiones cada una, las cuales constan a su vez de cinco equipos. La división territorial vigente fue introducida en la temporada 2004-2005, en la cual la mayoría de equipos están en la mitad del este del país: trece conjuntos están en la zona horaria del este, nueve en la zona central, tres en la zona horaria montañosa y cinco en la Pacífico.

La comisión se encarga de que haya el mismo número de franquicias en cada conferencia para mantener una división equivalente. En los casos en los que se produce una reubicación de franquicia, el mapa divisional se reestructura para que cada división cuente con los cinco equipos correspondientes.

Conferencia Oeste	
División	**Equipo**
Noroeste	Denver Nuggets
	Minnesota Timberwolves
	Oklahoma City Thunder
	Portland Trail Blazers
	Utah Jazz
Suroeste	Dallas Mavericks
	Houston Rockets
	Memphis Grizzlies
	New Orleans Pelicans
	San Antonio Spurs
Pacífico	Golden State Warriors
	Los Angeles Clippers
	Los Angeles Lakers
	Phoenix Suns
	Sacramento Kings

Tabla 1: Conferencia Oeste

Conferencia Este	
División	**Equipo**
Atlántico	Boston Celtics
	Brooklyn Nets
	New York Knicks
	Philadelphia 76ers
	Toronto Raptors
Central	Chicago Bulls
	Cleveland Cavaliers
	Detroit Pistons
	Indiana Pacers
	Milwaukee Bucks
Sureste	Atlanta Hawks
	Charlotte Hornets
	Miami Heat
	Orlando Magic
	Washington Wizards

Tabla 2: Conferencia Oeste

Esta competición se divide en dos etapas: fase regular y playoffs. Durante la primera, cada equipo disputa 82 partidos divididos en partes iguales entre encuentros de local y visitante. El calendario no es el mismo para todos, ya que los equipos se enfrentan en cuatro ocasiones con los oponentes de su propia división,

ante los de las otras dos divisiones de su conferencia, entre tres o cuatro veces y contra los de la otra conferencia, dos encuentros al año.

Durante la etapa de los Playoffs se disputan 3 rondas entre los 16 mejores clasificados, repartidos entre la Conferencia Oeste y la Conferencia Este. Los ganadores de la 1º ronda (o cuartos de final de conferencia) avanzan a la 2ª ronda (o Semifinales de conferencia). Posteriormente, a la 3ª ronda (o Finales de Conferencia) y por último, los vencedores, a las Finales de la NBA, disputadas entre los campeones de cada conferencia. Dicha final será al mejor de siete partidos, por lo que el primero que consiga vencer en cuatro ocasiones será finalmente el campeón de la NBA y, por consiguiente, campeón del anillo.

1.2 La NBA en las apuestas deportivas

1.2.1 Orígenes

Las apuestas deportivas [7] es una modalidad en la que se intenta predecir los resultados de una competición deportiva. El país pionero en las apuestas de todo tipo, sobre todo en carreras de caballos y carreras de galgos ha sido el Reino Unido. También son muy populares las apuestas de boxeo profesional en algunas ciudades de Estados Unidos, a las que se han sumado las apuestas en baloncesto, sobretodo, NBA y, en menor medida: NCAA y NCAAB.

Uno de los conceptos que todo apostante debe dominar es el de las cuotas [8]. Las cuotas son lo que una casa de apuestas ofrece al jugador si su apuesta resulta ganadora, y el encargado de fijarlas es la propia casa de apuestas. Según la cuota hay más o menos probabilidades de que una apuesta acabe en verde (acertada) o no, y se pueden expresar en tres formatos: decimal, fraccionario y americano.

1.2.2 ¿Cómo funcionan las cuotas?

A continuación se muestra la interfaz de la casa de apuestas Bet365 para el partido entre Toronto Raptors y Milwaukee Bucks:

Figura 1: Mercado de apuestas para Toronto Raptors - Milwaukee Bucks

Cuanto más alta sea la cuota mayor será el premio que se lleve el apostante por cada euro apostado, aunque también es cierto que las opciones de acertar la apuesta son menores. En cambio, si la cuota es baja las posibilidades de acierto se multiplican pero la recompensa por cada euro jugado no será tan elevada. Para calcular las posibilidades de que la apuesta resulte ganadora se puede dividir 1 entre la cuota y multiplicar el resultado por 100. Por ejemplo, en la ilustración 1 se puede observar que la cuota de ganador para Toronto Raptors es 3.35, por lo que las opciones serían: $1/3.35 = 0.30\,x\,100 = 30\%$. Esto quiere decir que la casa de apuestas le da un 30% de posibilidades de que gane Toronto Raptors. Si calculamos las posibilidades de que gane Milwaukee Bucks de la misma manera: $1/1.34 = 0.75\,x\,100 = 75\%$ y, es que si sumamos estos dos resultados nos da 105%, cuando la lógica nos dice que debería ser de un 100%. Esto se debe a que las casas de apuestas se aseguran un beneficio sea quien sea el ganador final.

Las cuotas son variables, oscilan según la oferta y la demanda. Las casas de apuestas lanzan los mercados a una cuota determinada, pero si muchos jugadores apuestan una gran cantidad de dinero en la misma apuesta su cuota bajará, haciendo que suban las cuotas del resto de opciones. A veces son las propias casas de apuestas las que reajustan los mercados, por ejemplo: ante la presencia de lesiones en la alineación de un conjunto, cuando se configura la parrilla en motos, etc., pero el cambio no afecta a las apuestas hechas antes del reajuste.

1.2.3 Formato de cuotas

El formato más habitual [9] en las casas de apuestas que operan en España, y en casi toda Europa, es el formato decimal, que indica la ganancia bruta por euro apostado. Siguiendo con el ejemplo anterior, si apostamos a Ganador Toronto nos llevaríamos 3.35€ por cada euro apostado, es decir, el euro que hemos apostado y

2.35€ de ganancias. O si apostamos a que gana Boston ganaríamos 1.34€ por euro apostado, es decir, el euro que hemos apostado y 0.34€ de ganancias.

También está el sistema fraccionario, muy utilizado en Reino Unido, en el cual se utiliza una fracción para determinar la cuota, donde el numerador indica el beneficio neto y el denominador la cantidad a jugar. Es decir, en una cuota ¾ tendríamos que jugar 4€ para ganar 3€.

Por último está el formato americano, utilizado principalmente en los países americanos como los Estados Unidos. Utilizan como punto de partida los 100 dólares, y a partir de ahí indican cuánto hay que apostar para ganar esos 100 dólares. Por ejemplo, con se trata de una cuota +150 indica que el jugador ganará 150 dólares por cada 100 dólares apostados; pero si la cuota es -150 nos indica que tendremos que apostar esos 150 dólares para ganar los 100 dólares anteriormente mencionados.

1.3 Big Data

1.3.1 ¿Qué es Big Data?

Cuando hablamos de Big Data [10] nos referimos a conjuntos de datos o combinaciones de conjuntos de datos cuyo tamaño, complejidad y velocidad de crecimiento dificultan su captura, gestión, procesamiento o análisis. Aunque el tamaño utilizado para determinar si un conjunto de datos determinado se considera Big Data no está firmemente definido y sigue cambiando con el tiempo, la mayoría de los analistas y profesionales actualmente se refieren a conjuntos de datos que van desde 30-50 Terabytes a varios Petabytes.

Aunque el término “big data” es relativamente nuevo, la acción de recopilar y almacenar grandes cantidades de información para su posterior análisis se viene realizando desde hace muchos años, cogiendo impulso a partir del año 2000 cuando el analista Doug Laney [11] articuló la definición ahora muy popular del big data como las tres Vs:

- Volumen: Recopilación de datos de diversas fuentes.
- Velocidad: Transmisión de los datos casi en tiempo real.
- Variedad: Los datos vienen en toda clase de formatos, desde datos numéricos estructurados en bases de datos tradicionales hasta documentos de texto no estructurados, correo electrónico, vídeo, audio, etc.

1.3.2 La importancia de Big Data

Lo que hace que Big Data sea tan útil para muchas empresas es el hecho de que proporciona respuestas a muchas preguntas que las empresa ni siquiera sabían que tenían. Con una cantidad tan grande de información, los datos pueden ser

explotados de cualquier manera que la empresa considere oportuna. Al hacerlo, las organizaciones son capaces de identificar los problemas de una forma más comprensible.

La recopilación de grandes cantidades de datos y la búsqueda de tendencias dentro de los datos permiten que las empresas se muevan mucho más rápido, sin problemas y de manera más eficiente. El análisis de Big Data ayuda a las organizaciones a aprovechar sus datos y utilizarlos para identificar nuevas oportunidades, lo que conduce, a su vez, a movimientos de negocios más inteligentes, operaciones más eficientes, mayores ganancias y clientes más satisfechos.

1.4 Data Mining (minería de datos)

El Data Mining o minería de datos [12] es un conjunto de técnicas y tecnologías que permiten explorar grandes bases de datos con el objetivo de encontrar patrones repetitivos que expliquen el comportamiento de estos datos.

La minería de datos surgió con la intención o el objetivo de ayudar a comprender una enorme cantidad de datos y, que estos, pudieran ser utilizados para extraer conclusiones que ayuden en la mejora y crecimiento de las empresas. Por lo tanto, los datos son el medio o la base para inferir conocimiento y transformar estos datos en información relevante para que las empresas puedan abarcar mejoras y soluciones que les ayuden a conseguir sus objetivos.

Las técnicas de minería de datos crean modelos que son predictivos y/o descriptivos. En este caso nos centraremos en los modelos predictivos o aprendizaje supervisado, aquellos que responden preguntas sobre datos futuros. ¿Quién ganará el siguiente partido?

CAPÍTULO 2 Objetivos

2.1 Objetivos principales

Adentrarse dentro del mundo del machine learning (aprendizaje automático) conociendo diversas técnicas de clasificación, diferentes librerías que nos ofrece Python y, sobretodo, familiarizarse con los árboles de decisión (decision trees) y los bosques (random forests).

2.2 Objetivos específicos

Con el desarrollo de este proyecto se pretenden conseguir los siguientes objetivos:

- Identificar, cargar, limpiar y analizar los datos disponibles.
- Estudiar el algoritmo de árbol de decisión.
- Entrenar el predictor para determinar el ganador del partido.
- Mejorar la precisión del clasificador utilizando un método de ensemble: random forests.

2.3 Otros objetivos

Como objetivos secundarios se pretende realizar una página web sencilla en la que se muestre los resultados de las predicciones realizadas previamente. De esta manera todo aquel que lo desee podrá acceder a esta página desde internet y ver qué equipos ganarán en los próximos días según el modelo predictivo realizado en este proyecto.

CAPÍTULO 3 Fases y desarrollo del proyecto

3.1 Metodología

La metodología utilizada en este proyecto está basada en KDD [13] (Knowledge Discovery in Databases). Este modelo se define como “el proceso de identificar patrones válidos, novedosos, potencialmente útiles y principalmente entendibles”. A continuación van a definirse los diferentes pasos y sus adaptaciones a los objetivos del proyecto:

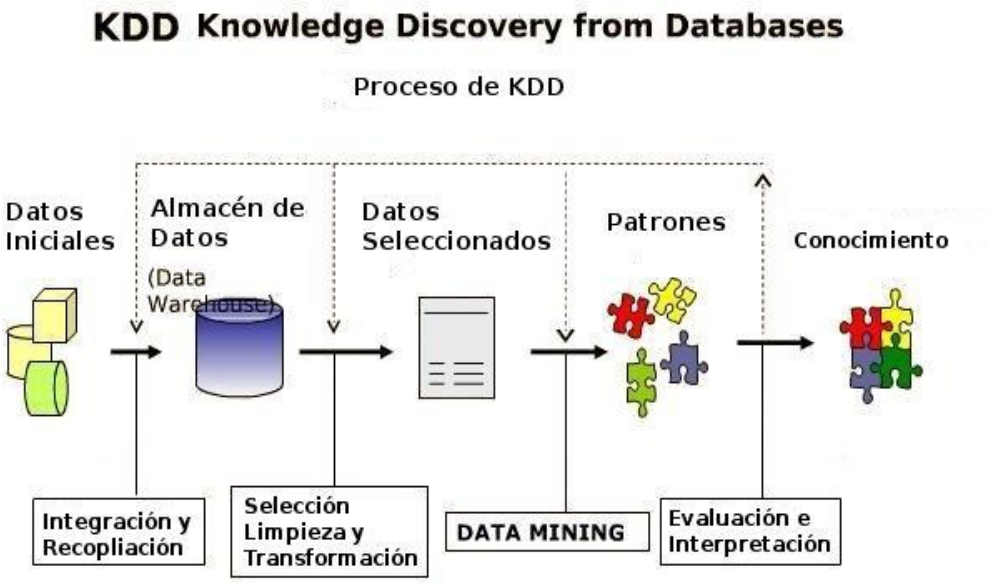

Figura 2: Metodología KDD

1. **Selección de datos.** En esta etapa se determinan las fuentes de datos y el tipo de información a utilizar. Es la etapa donde los datos relevantes para el análisis son extraídos desde la o las fuentes de datos. En este caso, los datos relacionados con ámbitos deportivos se han extraído de la página web de Basketball Reference [14] ya que nos proporciona conjuntos de datos referidos a los partidos de cada temporada, estadísticas de jugadores, clasificaciones, etc.
   En cuanto a las cuotas de las casas de apuestas, se han extraído de la página web de OddsPortal [15], la cual proporciona las cuotas de las casas de apuestas para cada partido desde la temporada 2008/09 hasta la temporada actual

2. **Preprocesamiento.** Esta etapa consiste en la preparación y limpieza de los datos extraídos desde las distintas fuentes de datos. En esta etapa se utilizan diversas estrategias para manejar datos faltantes o en blanco y datos inconsistentes, obteniéndose al final una estructura de datos adecuada para su posterior transformación. Para llevar a cabo este paso se ha hecho uso de Pandas [16], una librería de Python destinada al análisis de datos, que proporciona unas estructuras de datos flexibles y que permiten trabajar con ellos de forma muy eficiente.

3. **Transformación:** Consiste en el tratamiento preliminar de los datos, transformación y generación de nuevas variables a partir de las ya existentes. Para el desarrollo de este proyecto se han generado varias características útiles como, por ejemplo: “Media de puntos anotados por el equipo local” o “Porcentaje de victorias del equipo local”.

4. **Data Mining:** Es la fase de modelamiento, en donde se aplica una serie de métodos con el objetivo de extraer patrones previamente desconocidos e imposible procesar por el humano.

5. **Interpretación y Evaluación:** Se identifican los patrones obtenidos y se realiza una evaluación de los resultados obtenidos.

3.2 Herramientas y lenguaje utilizado

Los datos se han almacenado en hojas de cálculo de Excel, específicamente en formato CSV separados por comas (,) para posteriormente importarlos a la herramienta Jupyter Notebook.

Figura 3: Logo Jupyter Notebook

Jupyter Notebook [17] es un entorno de trabajo interactivo que permite desarrollar código en Python de manera dinámica. En una sola interfaz, los usuarios pueden escribir, documentar y ejecutar código, visualizar datos, realizar cálculos y ver los resultados obtenidos. Concretamente, la fase de prototipado incluye la ventaja de que el código se organiza en celdas independientes, es decir, es posible probar bloques concretos de código de forma individual.

3.3 Datos

3.3.1 Estructura de los ficheros

3.3.1.1 Dataset principal

Como se ha explicado previamente, los datos se han recogido en una hoja de cálculo. A continuación se podrá observar una pequeña muestra del dataset principal del proyecto recién exportado de la página de Basketball Reference:

	A	B	C	D	E	F	G	H	I	J
1	Date	Start (ET)	Visitor/Neut	PTS	Home/Neutr	PTS			Attend.	Notes
2	Tue Oct 29 2(	7:00p	Orlando Mag	87	Indiana Pace	97	Box Score		18165	
3	Tue Oct 29 2(	10:30p	Los Angeles	103	Los Angeles	116	Box Score		18997	
4	Tue Oct 29 2(	8:00p	Chicago Bull:	95	Miami Heat	107	Box Score		19964	
5	Wed Oct 30 2	7:00p	Brooklyn Ne	94	Cleveland Ca	98	Box Score		20562	
6	Wed Oct 30 2	8:30p	Atlanta Hawl	109	Dallas Maver	118	Box Score		19834	
7	Wed Oct 30 2	7:30p	Washington	102	Detroit Pisto	113	Box Score		19258	
8	Wed Oct 30 2	10:30p	Los Angeles	94	Golden State	125	Box Score		19596	
9	Wed Oct 30 2	8:00p	Charlotte Bo	83	Houston Roc	96	Box Score		18083	
10	Wed Oct 30 2	8:00p	Orlando Mag	115	Minnesota T	120	Box Score	OT	17988	
11	Wed Oct 30 2	8:00p	Indiana Pace	95	New Orleans	90	Box Score		17803	
12	Wed Oct 30 2	7:30p	Milwaukee E	83	New York Kn	90	Box Score		19812	
13	Wed Oct 30 2	7:00p	Miami Heat	110	Philadelphia	114	Box Score		19523	
14	Wed Oct 30 2	10:00p	Portland Trai	91	Phoenix Sun	104	Box Score		17208	
15	Wed Oct 30 2	10:00p	Denver Nugg	88	Sacramento	90	Box Score		17317	
16	Wed Oct 30 2	8:30p	Memphis Gri	94	San Antonio	101	Box Score		18581	

Tabla 3: Fragmento dataset principal del proyecto

El formato es simple, se compone de un encabezado que da nombre a cada atributo. Cada fila representa un partido y contiene en cada columna los valores para el atributo que nombra en el encabezado.

3.3.1.2 Otros datasets utilizados

Para obtener ciertos atributos se ha tenido que hacer uso de otros datasets: las clasificaciones de las temporadas y las estadísticas de los jugadores, para así determinar qué equipo de los enfrentados fue más regular en la temporada pasada e intentar predecir qué equipo de los dos tiene mejores jugadores en su plantilla acorde a la valoración individual de cada jugador que compone cada equipo. Para obtener dicha valoración se debe seguir la siguiente fórmula:

Valoración = PTS + ASIS + REB + TAPF + BR + FR + T1C + T2C + T3C - BP - TAPC - FP - T1I - T2I - T3I

Donde:

PTS = PUNTOS, ASIS = ASISTENCIAS, REB = REBOTES, TAPF = TAPONES A FAVOR, BR = BALONES ROBADOS, FR = FALTAS RECIBIDAS, T1C = TIROS LIBRES CONVERTIDOS, T2C = TIROS DE 2 CONVERTIDOS, T3C = TRIPLES CONVERTIDOS, BP = BALONES PERDIDOS, TAPC = TAPONES EN CONTRA,

FP = FALTAS PERSONALES, T1I = TIROS LIBRES INTENTADOS, T2I = TIROS DE 2 INTENTADOS, T3I = TRIPLES INTENTADOS.

Como ya se ha detallado anteriormente, también se ha extraído las cuotas de las casas de apuestas con el objetivo de que estas aporten un valor elevado al modelo e intentar encontrar un patrón que ayude a clasificar el ganador del encuentro.

Estos datasets tienen el mismo aspecto que el principal. Para el de la clasificación: cada fila representa en qué puesto de la clasificación quedó cada equipo en una temporada en concreto, para el de las estadísticas de los jugadores cada fila representa cada jugador y en sus columnas todos aquellos atributos anteriormente mencionados (PTS, ASIS, REB…) y, por último, en el dataset de las cuotas, cada fila representa un partido junto a la cuota del equipo Local y la cuota del equipo Visitante.

3.4 Primeros pasos

3.4.1 Librerías

Para el desarrollo de este proyecto se ha hecho uso de una serie de librerías que ofrece Python para el procesamiento los datos de una manera más fácil. Las librerías elegidas son:

- En primer lugar se ha empleado Pandas para cargar, limpiar y analizar los datos disponibles. La estructura de datos básica de pandas se denomina DataFrame, que es una colección ordenada de columnas con nombres y tipos, parecido a una tabla de base de datos, donde una fila representa un caso (ejemplo) y las columnas representan atributos.

- Usaremos Numpy [18] para todo tipo cálculo numérico que se necesite a lo largo del proyecto. Con esta librería nos resultará más sencillo realizar cualquier operación, ya que cuenta con todo tipo de funciones matemáticas.

- Para los procesos de Machine Learning se ha pensado en el uso de Scikit-Learn [19], ya que cuenta con varios algoritmos de clasificación y regresión como Support Vector Machines (máquinas de vectores de soporte), decision tree(árbol de decisión), random forests (árboles), k-means (k-medias), etc. Cabe destacar que está diseñada para interactuar con bibliotecas numéricas de Python y Numpy.

3.4.2 Datasets

Para entender un poco mejor el contexto del proyecto hay que conocer la estructura de los datos elegidos para su realización. Cabe destacar que se recogerán los datos desde la temporada 2013/14 hasta la temporada actual (2018/19), siendo esta última la cual se va a predecir.

Cada temporada cuenta con aproximadamente 1300 partidos, lo que hacen un total aproximado de 7800 encuentros entre fase regular y playoffs. Sin embargo, en las primeras jornadas los datos relacionados al rendimiento de los equipos no son del todo fiables. Es por esto por lo que se ha decidido omitir los 100 primeros partidos de cada temporada, tanto para el entrenamiento del modelo como para la predicción, ya que se ha considerado que es a partir de 100 encuentros disputados cuando se pueden obtener datos con cierta relevancia.

Por otro lado, en cada temporada participa una media de 500 jugadores, para los cuales Basketball Reference nos ofrece unos 30 atributos de los cuales obtener información sobre su rendimiento. Por último, los conjuntos de datos de las clasificaciones, para así sacar conclusiones de la posición en la que quedó cada equipo la temporada anterior.

Cuando visualizamos por primera vez los datos vemos que tendremos que hacer una pequeña tarea de limpieza en el dataset. Esto se debe a que hay atributos sin relevancia para el desarrollo del proyecto como: si en dicho partido hubo una prórroga (para la mayoría de partidos este campo está en blanco puesto que son muy pocos los partidos que se van a un tiempo extra) y, el número de personas que asistieron de público a ver el partido. También es destacable una columna de Notas en la que todos sus valores están en blanco y, por lo tanto, es un atributo que se debe eliminar del dataset, además de los mencionados anteriormente.

3.4.3 Experimentación previa

Antes de comenzar con las técnicas de aprendizaje automático, resulta interesante realizar pequeñas comprobaciones estadísticas con los datos extraídos.

En primer lugar, se ha comprobado que en un 57.92% de los partidos, el equipo local es el ganador. Por otro lado, en un 60.34%, gana el equipo que tiene un mayor porcentaje de victorias hasta ese momento. Sin embargo, aquellos equipos que tienen mayor valoración media de sus jugadores con respecto a su rival, ganan en un 47.38% y, por último, en un 46.43% se alza con la victoria aquel equipo que tiene una mayor diferencia entre puntos anotados y puntos recibidos hasta el momento del partido.

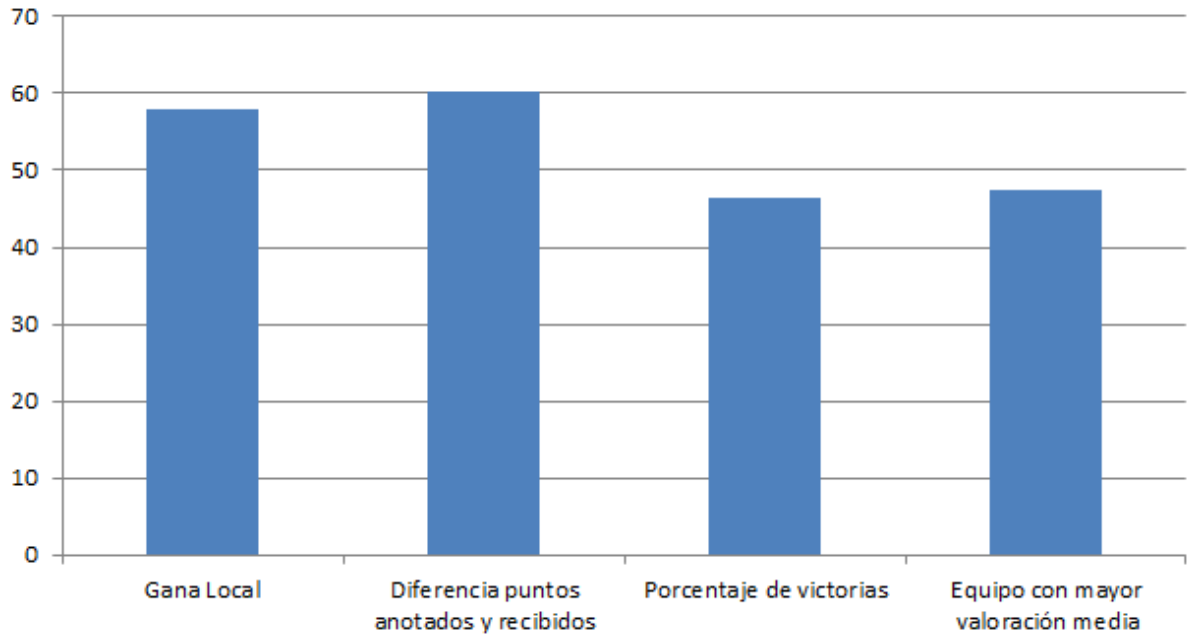

Figura 4: Comparativa porcentaje de ganadores

Con este experimento se intenta obtener una comparativa general de datos a los que tiene acceso cualquier persona para, posteriormente, compararlos con los resultados obtenidos tras el entrenamiento del modelo.

3.4.4 Atributos

```
'Date', 'Visitor Team', 'Home Team', 'HomeWin',
'HomeLastWin', 'VisitorLastWin', 'Home_Days_Last_Game',
'Visitor_Days_Last_Game', 'Home_High_Rank', 'Home_valoration',
'Visitor_valoration', 'PorcentajeTotalVictoriasEquipoLocal',
'PorcentajeTotalVictoriasEquipoVisitante',
'PorcentajeVictoriasEquipoLocalComoLocal',
'PorcentajeVictoriasEquipoVisitanteComoVisitante',
'Cuota Local', 'Cuota Visitante', 'MediaPuntosAnotadosLocal',
'MediaPuntosRecibidosLocal', 'MediaPuntosAnotadosVisitante',
'MediaPuntosRecibidosVisitante', 'PorcTirosCampoLocal',
'PorcTriplesLocal', 'PorcTirosLibresLocal', 'DefLocal',
'RebotesPPLocal', 'AsistenciasPPLocal', 'RobosPPLocal',
'TaponesPPLocal', 'PerdidasLocal', 'FaltasPersonalesLocal',
'PorcTirosCampoVisitante', 'PorcTriplesVisitante',
'PorcTirosLibresVisitante', 'DefVisitante', 'RebotesPPVisitante',
'AsistenciasPPVisitante', 'RobosPPVisitante', 'TaponesPPVisitante',
'PerdidasPPVisitante', 'FaltasPersonalesVisitante',
```

Figura 5: Atributos

Una vez visualizado y entendido los datos, se ha pensado en diferentes atributos que puedan ayudar al modelo a mejorar su porcentaje de acierto. Antes de explicar qué atributos se han escogido y su significado, es necesario recordar y explicar el problema que se quiere resolver. El objetivo principal es clasificar cada partido de la temporada actual (2018/19) como: gana el equipo LOCAL o gana el equipo VISITANTE. Para ello se ha decidido predecir si gana el equipo local (True) o no (False), de esta manera sabríamos cuál de los dos equipos se alza con la victoria.

Posteriormente se detallará cada atributo y qué se espera que aporten al modelo:

- **Date:** Fecha del partido.
- **Visitor Team:** Nombre del equipo visitante.
- **Home Team:** Nombre del equipo local.
- **HomeWin:** Este atributo será el que predecirá el modelo. Sus posibles valores, como bien se ha mencionado anteriormente, son True (gana local) o False (gana visitante).
- **HomeLastWin y VisitorLastWin:** Indican si los equipos ganaron su el último partido que disputaron. De esta manera se intenta saber si los equipos vienen en una buena o mala dinámica y, a su vez, si llegan al partido con una mentalidad positiva o negativa tras una victoria o derrota.
- **Home_Days_Last_Game y Visitor_Days_Last_Game:** Establecen el número de días de descanso que han tenido ambos equipos al llegar a disputar el partido. Con estos atributos se intenta saber si los jugadores de los equipos están cansados o si llegan lo suficientemente frescos como para dar el 100%.
- **Home_High_Rank:** Este atributo nos muestra si el equipo local quedó mejor clasificado la temporada anterior. Es cierto que de una temporada a otra los equipos pueden cambiar bastante pero con esto se intenta buscar si los equipos que fueron mejores en la temporada pasada suelen ganar a los equipos peores clasificados.
- **Home_valoration y Visitor_valoration:** Indican la suma de las valoraciones de todos sus jugadores. De esta manera se podrá saber si, en líneas generales, los equipos con una mayor valoración (lo que se traduce en: mejores jugadores o un buen rendimiento de los mismos) ganan a los equipos con una valoración más baja.
- **PorcentajeTotalVictoriasEquipoLocal y PorcentajeTotalVictoriasEquipoVisitante:** Expresan el porcentaje de victorias totales de ambos equipos hasta la fecha del partido. Se espera que muestren una medición del rendimiento global de los equipos.

- **PorcentajeTotalVictoriasEquipoLocalComoLocal y PorcentajeTotalVictoriasEquipoVisitanteComoVisitante:** Como los nombres propiamente indican, muestra el porcentaje de victorias del local en su estadio y el porcentaje de victorias del visitante a domicilio. Es probable que haya equipos muy fuertes jugando en su campo pero que cuando se desplazan para jugar en otros estadios no son tan fiables. Es esto lo que se intenta encontrar con estas dos características.
- **Cuota Local y Cuota Visitante:** Se establece este atributo para recoger la cuota local previa al partido, y otro para la cuota visitante.
- **MediaPuntosAnotadosLocal y MediaPuntosAnotadosVisitante:** Muestran la media de puntos de cada equipo durante la temporada hasta el día antes del partido. Se pretende observar el potencial ofensivo de ambos equipos.
- **MediaPuntosRecibidosLocal y MediaPuntosRecibidosVisitante:** Muestran la media de puntos recibidos de cada equipo durante la temporada hasta el día antes del partido. Se pretende observar el potencial defensivo de ambos equipos.
- **PocTirosCampoLocal y PorcTirosCampoVisitante:** Representan el potencial ofensivo en tiros de 2 puntos.
- **PorcTriplesLocal y PorcTriplesVisitante:** Representan el potencial ofensivo en tiros de 3 puntos.
- **PorcTirosLibresLocal y PorcTirosLibresVisitante:** Representan el potencial ofensivo en tiros libres.
- **DefLocal y DefVisitante:** Representa para cada equipo, un valor medio sobre su desempeño defensivo.
- **RebotesPPLocal y RebotesPPVisitante:** Representa la capacidad de cada equipo para generar nuevas jugadas.
- **AsistenciasPPLocal y AsistenciasPPVisitante:** Representa para cada equipo el promedio de asistencias, es decir, mide el rendimiento ofensivo.
- **RobosPPLocal y RobosPPVisitante:** Representa para cada equipo el promedio de robos, es decir, mide el rendimiento defensivo.
- **TaponesPPLocal y TaponesPPVisitante:** Representa para cada equipo el promedio de tapones, es decir, mide también el rendimiento defensivo.
- **PerdidasLocal y PerdidasVisitante:** Representa para cada equipo el promedio de pérdidas.
- **FaltasPersonalesLocal y FaltasPersonalesVisitante:** Representa para cada equipo el promedio de faltas personales.

3.4.5 Selección de atributos

Uno de los principales beneficios de la selección de atributos está plasmado por la famosa frase “Menos es más” de Ludwig Mies van der Rohe [20]. Menos

atributos son deseables ya que reduce la complejidad del modelo y, un modelo más simple es más fácil de entender y explicar.

Otros beneficios que nos proporciona una buena selección de atributos son:

- **Reduce el sobreentrenamiento:** Menos datos redundantes significan menos oportunidades para tomar decisiones.
- **Mejora la precisión:** Menos datos engañosos se convierten en una mejora en la exactitud del modelo.
- **Reduce el tiempo de entrenamiento:** Menos datos significa que los algoritmos aprenden más rápido.

A continuación se va a realizar el estudio de selección de atributos tanto para un árbol como para un bosque. Cabe recordar que para esta selección no contará el atributo “HomeWin” ya que esta va a ser la característica a predecir y, por lo tanto, no se utilizará para el entrenamiento del modelo.

La idea es seleccionar los 20 atributos con mayor porcentaje de importancia para cada uno de los dos métodos. Para obtener esta importancia, tanto para un árbol como para un bosque se ha utilizado la función ***“feature_importances_”*** que nos proporcionan ambos modelos de predicción automática.

Atributo	**DecisionTree**	**Forest**
Visitor Team	0	0.00342779
Home Team	0.04478128	0.01043553
HomeLastWin	0	0.00255808
VisitorLastWin	0	0.00062267
Home_Days_Last_Game	0	0.00094085
Visitor_Days_Last_Game	0	0.00084279
Home_High_Rank	0	0.00968487
Home_valoration	0	0.00414899
Visitor_valoration	0	0.00376133
PorcentajeTotalVictoriasEquipoLocal	0	0.10009997
PorcentajeTotalVictoriasEquipoVisitante	0.01397402	0.04533109
PorcentajeTotalVictoriasEquipoLocalComoLocal	0.01932348	0.04962935
PorcentajeTotalVictoriasEquipoVisitanteComoVisitante	0.02498266	0.04925646

Cuota Local	0.14956869	0.15027982
Cuota Visitante	0.42559106	0.17704635
MediaPuntosAnotadosLocal	0	0.01682852
MediaPuntosRecibidosLocal	0.02972805	0.01730298
MediaPuntosAnotadosVisitante	0	0.05422049
MediaPuntosRecibidosVisitante	0.02387943	0.01421896
PorcTirosCampoLocal	0.04234126	0.05844432
PorcTriplesLocal	0	0.01404924
PorcTirosLibresLocal	0	0.00809791
DefLocal	0.0519987	0.00842297
RebotesPPLocal	0	0.01546761
AsistenciasPPLocal	0	0.00482673
RobosPPLocal	0	0.02234039
TaponesPPLocal	0	0.01078376
PerdidasLocal	0	0.0066144
FaltasPersonalesLocal	0.05449755	0.00775457
PorcTirosCampoVisitante	0	0.01905651
PorcTriplesVisitante	0	0.01922818
PorcTirosLibresVisitante	0	0.00415275
DefVisitante	0.09050532	0.02368314
RebotesPPVisitante	0.0288285	0.02116605
AsistenciasPPVisitante	0	0.00727859
RobosPPVisitante	0	0.01703333
TaponesPPVisitante	0	0.00624214
PerdidasVisitante	0	0.00465047
FaltasPersonalesVisitante	0	0.01007001

Tabla 4: Porcentaje importancia atributos

Como se puede observar tras la realización del estudio, para el clasificador DecisionTree solo hay 13 características que reflejen un grado de importancia para el modelo, por lo que estos serán los escogidos para continuar con el experimento. A modo de curiosidad se puede observar la gran importancia que tienen las cuotas de las casas de apuestas para el modelo, tanto para un solo árbol como para un bosque, siendo estos atributos los más relevantes con diferencia en ambos casos.

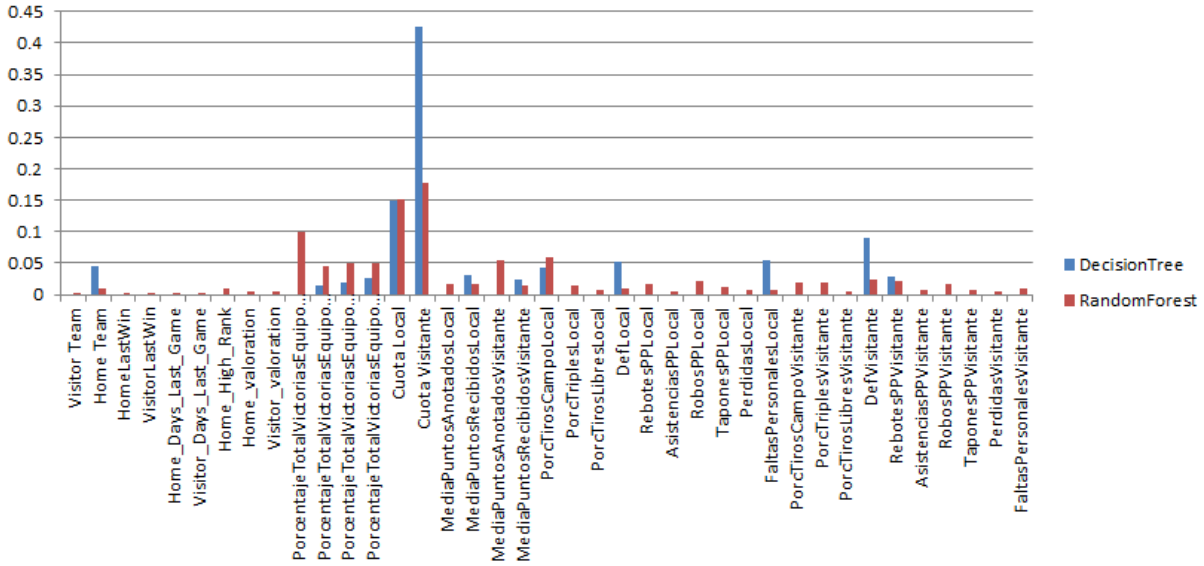

Figura 6: Comparativa atributos DecisionTree vs Forest

3.5 Aprendizaje automático

3.5.1 Métodos utilizados

3.5.1.1 DecisionTree (Árbol de decisión)

Un árbol de decisión es una forma gráfica y analítica de representar todas las alternativas disponibles a la hora de tomar una decisión. Estos árboles nos ayudan a tomar la decisión “más acertada” desde un punto de vista probabilístico, ante un abanico de posibles soluciones.

Un árbol de decisión generalmente comienza con un solo nodo, que se bifurca en posibles resultados. Cada uno de estos resultados conduce a nodos adicionales, que se ramifican hacia otras posibilidades. Esto le da una forma arborescente.

Estos árboles están compuestos por:

- **Nodo de decisión:** Indica que una decisión necesita tomarse en ese punto del proceso. Normalmente se representa mediante un cuadrado.

- **Nodo de probabilidad:** Muestra las probabilidades de ciertos resultados. Se suele representar con un círculo.
- **Rama:** Nos muestra los distintos caminos que se pueden seguir cuando tomamos una decisión. Se representa con una flecha.
- **Nodo terminal:** Muestra el resultado definitivo de un camino de decisión.

Y tienen la siguiente forma:

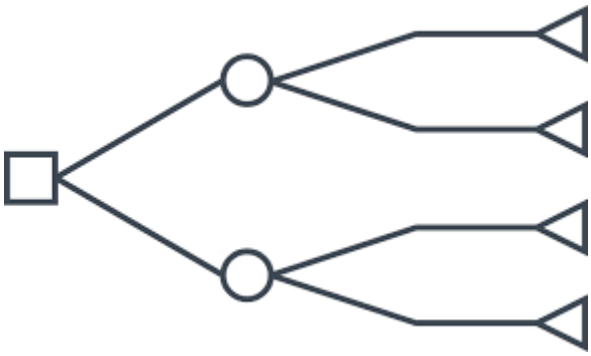

Figura 7: Ejemplo de árbol de decisión

3.5.1.2 Random Forest (Bosque)

Un bosque es un método que combina una gran cantidad de árboles de decisión independientes probados sobre conjuntos de datos aleatorios con igual distribución. Cada árbol se evalúa de forma independiente y la predicción del bosque será la media de los n árboles que componen el bosque. Es decir, la proporción de árboles que toman una misma respuesta se interpreta como la probabilidad de la misma.

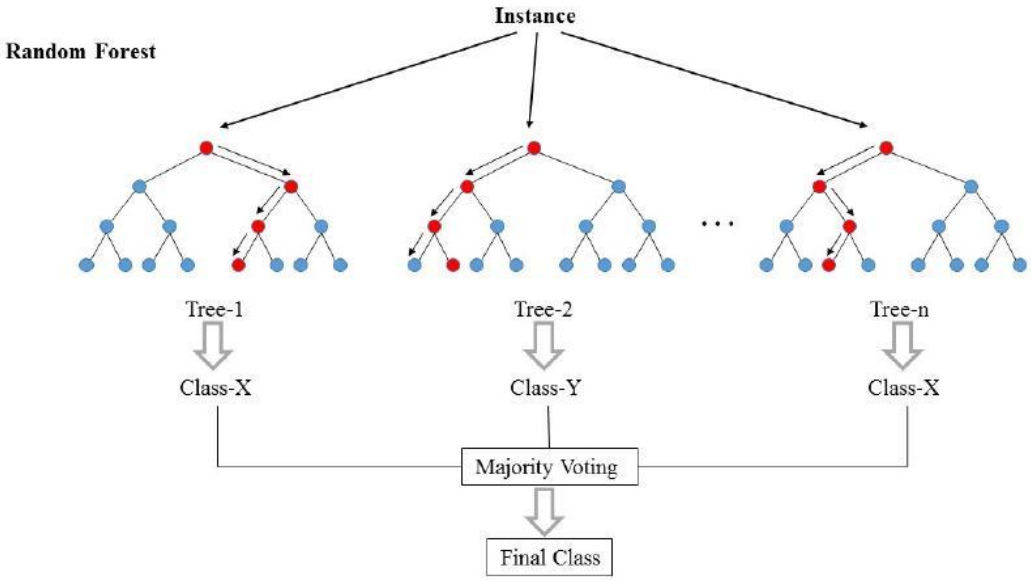

Figura 8: Ejemplo de RandomForest

3.5.2 Normalización de datos

Una vez se han extraído todos los atributos anteriormente mencionados, antes de empezar a entrenar el modelo conviene transformar los datos categóricos o aquellos que contengan texto en datos numéricos, ya que el modelo predictivo no entiende datos que contengan texto. De esta manera se procede a normalizar los nombres de los equipos mediante la clase Label Encoder [21] de sklearn, una clase creada para la normalización de los datos, la cual asigna valores entre 0 y nclases-1.

A continuación se mostrará un ejemplo de cómo ha quedado esta normalización:

Visitor Team
Orlando Magic
Los Angeles Clippers
Chicago Bulls
Brooklyn Nets
Atlanta Hawks
Washington Wizards
Los Angeles Lakers
Charlotte Bobcats
Orlando Magic
Indiana Pacers
Milwaukee Bucks
Miami Heat

Figura 9: Equipos antes

Visitor Team
30
5
11
8
14
24
13
29
30
9
18
1

Figura 10: Equipos después

Como se puede observar, al equipo Orlando Magic se le ha asignado el número 30, a Los Ángeles Clippers el número 5 y así sucesivamente hasta completar todos los nombres de los equipos con su número correspondiente.

CAPÍTULO 4 Resultados y evaluación de los modelos

4.1 Evaluación de resultados

En este apartado se muestran los resultados de aplicar ambos algoritmos descritos anteriormente entrenando con diferentes conjuntos de temporadas pasadas, con el fin de realizar un análisis que ofrezca conclusiones relevantes.

Porcentajes de acierto		
Temporadas con las que entrenar	**Decision Tree**	**Random Forest**
2017/18	66.41%	69.6%
2016/17 - 2017/18	64.06%	66’4%
2015/16 - 2016/17 - 2017/18	65.66%	67.8%
2014/15 - 2015/16 - 2016/17 - 2017/18	64.67%	67.1%
2013/14 - 2014/15 - 2015/16 - 2016/17 - 2017/18	63.69%	67.2%

Tabla 5: Comparativa resultados

Como podemos observar, la técnica Random Forest siempre obtiene mejores resultados que el método Decision Tree. En este caso, resulta curioso que el Random Forest, entrenando exclusivamente con la temporada previa 2017/18, ha sido el que ha ofrecido el porcentaje de acierto más elevado alcanzando un 69.6%, por lo que nos centraremos en este caso durante el resto del proyecto.

Este resultado nos hace pensar que el motivo de que se obtenga un mejor resultado entrenando simplemente con los datos de una temporada que con el conjunto de 5 temporadas, es posible que sea por la inestabilidad de los equipos en la NBA de una temporada a otra debido a los límites salariales. Este límite es la cantidad total de dinero que una franquicia puede gastar para pagar a sus jugadores. Esta cantidad varía cada temporada, y se calcula dependiendo de lo sucedido la temporada anterior. Este límite se ha establecido para evitar que los equipos con grandes excedentes de beneficios se puedan hacer con todos los mejores jugadores disponibles, facilitando de esa manera que se mantenga una igualdad dentro de un orden. Es por esto por lo que es muy frecuente ver equipos ganándolo todo durante varios años y, unos años más tarde, verlos en mitad de tabla o incluso en los puestos bajos durante 4-5 años seguidos.

Cabe destacar que aunque lograr un 69.6% de acierto supone acertar alrededor de 910 partidos de 1300 que componen la temporada, la explotación del modelo en una casa de apuestas no garantiza el éxito. Los beneficios dependerán de qué tipo de partidos son los que el clasificador está acertando y cuáles los que está fallando. Si el modelo solo es capaz de acertar aquellos partidos con una probabilidad muy alta (baja cuota), es probable que se acabe perdiendo dinero.

4.2 Parámetros de los modelos

Para obtener el mejor estimador se ha hecho uso de la clase GridSearchCV [22] de scikit-learn, la cual permite evaluar y seleccionar de forma sistemática los parámetros de un modelo. Indicándole un modelo y los parámetros a probar, puede evaluar el rendimiento del primero en función de los segundos mediante validación cruzada [23].

En este caso, los parámetros a evaluar han sido:

- **min_samples_leaf:** Mínimo número de muestras requeridas para estar en un nodo hoja. Un punto de división en cualquier profundidad solo se considerará si deja al menos min_samples_leaf muestras de entrenamiento en cada una de las ramas izquierda y derecha.
- **max_depth:** Máxima profundidad que puede alcanzar el árbol.
- **min_samples_split:** Mínimo número de muestras necesarias para dividir un nodo interno.
- **n_estimators (sólo en el modelo Random Forest):** Número de árboles que componen el bosque.

La validación cruzada es una técnica con la que se puede identificar la existencia de diferentes problemas durante el entrenamiento de los modelos, como la aparición de sobreajuste [24]. Permitiendo así obtener modelos más estables. En la validación cruzada el conjunto de datos de entrenamiento se divide en grupos de igual tamaño y, una vez realizada dicha partición, se procede a entrenar el modelo una vez por cada uno de los grupos. Utilizando todos los grupos menos el de la iteración para entrenar y este para validar los resultados.

El overfitting o sobreajuste de un modelo hace referencia a que este tiende a adaptarse a los datos de entrenamiento, lo que supone un gran porcentaje de acierto con estos datos, mientras que una vez se le pasan los datos de validación (datos nuevos para el modelo) se obtiene una baja notable en el porcentaje de acierto. Una solución a este problema es la poda del árbol [25], haciendo uso del atributo max_depth que se mencionaba anteriormente, especificando la profundidad máxima que se puede alcanzar a la hora de recorrer el árbol, de manera que no se permita llegar a los nodos finales del mismo.

```
num_folds = 10
seed = 7
parameter_space = {
    "n_estimators": [100,200,300],
    "min_samples_leaf": [2,4,6],
    "max_depth": [2,3,4,5,6,7,8,9,10],
    "criterion": ["gini","entropy"],
    "min_samples_split": [100,200,300,400,500,600,700,800,900,1000,1100,1200,1300,1400,1500]}
kfold = KFold(n_splits = num_folds, random_state=seed)
clf = RandomForestClassifier(random_state=14)
grid2 = GridSearchCV(clf, parameter_space, scoring='accuracy', cv = kfold )
grid_result2 = grid2.fit(train[predictorsforest], train[target])
forest = grid2.best_estimator_
forest.fit(train[predictorsforest],train[target])
```

Figura 11: Ejemplo de código utilizando GridSearchCV para la inicialización de un modelo

4.2.1 Parámetros más óptimos elegidos por GridSearchCV para el Random Forest

Una vez realizado y obtenido los resultados del experimento, podemos observar que los parámetros elegidos por la clase GridSearchCV y que, por lo tanto, nos brindan el mejor resultado para el modelo son:

- **max_depth:** 7
- **min_samples_leaf:** 4
- **min_samples_split:** 100
- **n_estimators:** 150

```
Accuracy: 69.6% using  {'max_depth': 7, 'min_samples_leaf': 4, 'min_samples_split': 100,
 'n_estimators': 150}
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                       max_depth=7, max_features='auto', max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=4, min_samples_split=100,
                       min_weight_fraction_leaf=0.0, n_estimators=150,
                       n_jobs=None, oob_score=False, random_state=14, verbose=0,
                       warm_start=False)
```

Figura 12: Mejores parámetros para el RandomForest

4.3 Experimentación adicional

En este apartado se va a realizar una serie de experimentos alternativos.

4.3.1 Estudio tasa de acierto a lo largo del tiempo

En el primero se va a comprobar cómo evoluciona la tasa de acierto a lo largo que avanza la temporada. Tiene como objetivo estudiar la evolución del rendimiento de ambos modelos en el tiempo. Para ello se ha dividido el conjunto total de partidos en 8 partes, en relación a los meses en los que se disputan los partidos:

1. Mes de octubre
2. Mes de noviembre.
3. Mes de diciembre.
4. Mes de junio.
5. Mes de febrero.
6. Mes de marzo.
7. Mes de abril.
8. Meses de mayo y junio, debido a que en estos meses se disputan los playoffs y, en comparación, se disputan menos encuentros.

Se espera que los resultados muestren fuertes variaciones al comienzo de la temporada y, conforme vayan pasando las jornadas, el modelo muestre una mayor precisión a la hora de clasificar el ganador.

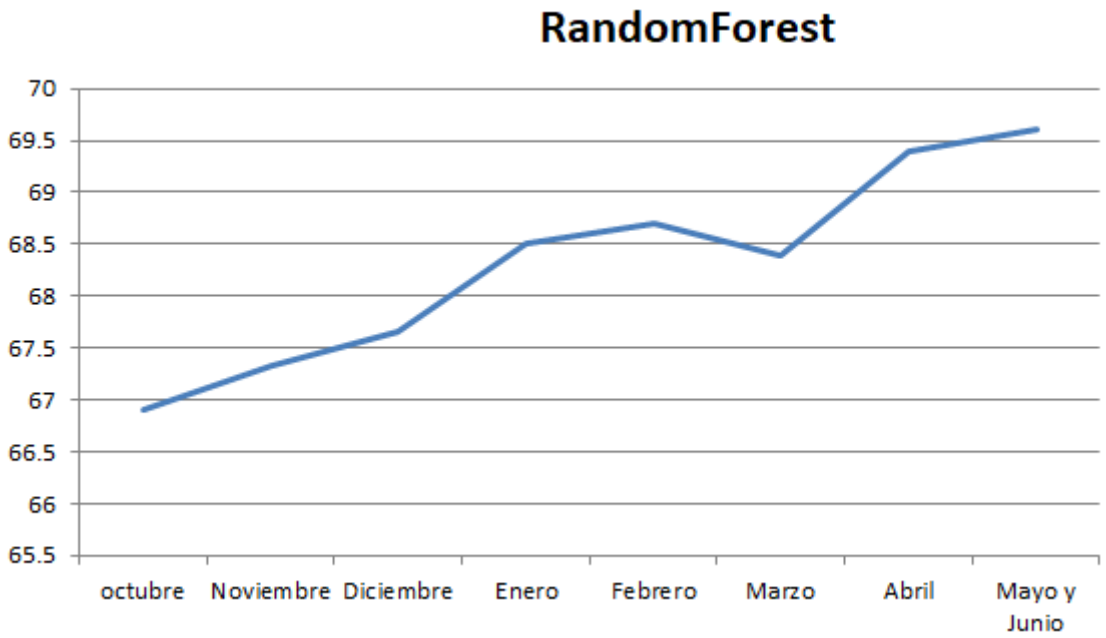

Figura 13: Evolución forest con el paso del tiempo

Tal y como se había pronosticado, conforme avanza la temporada, el clasificador es más preciso. Es cierto que la diferencia no es tan exagerada, ya que el margen entre el pico más bajo y el más alto es de un 1.5% aproximadamente pero, de esta gráfica podemos pensar que si las temporadas fueran más largas, las tasas de acierto incrementarían notablemente con el paso del tiempo.

4.3.2 Evolución del RandomForest en comparación al número de árboles

A continuación se va a comprobar qué tal se comporta el modelo a medida que va aumentando el número de árboles que lo componen. Para ello se ha establecido como valor mínimo de árboles 2 y máximo 600.

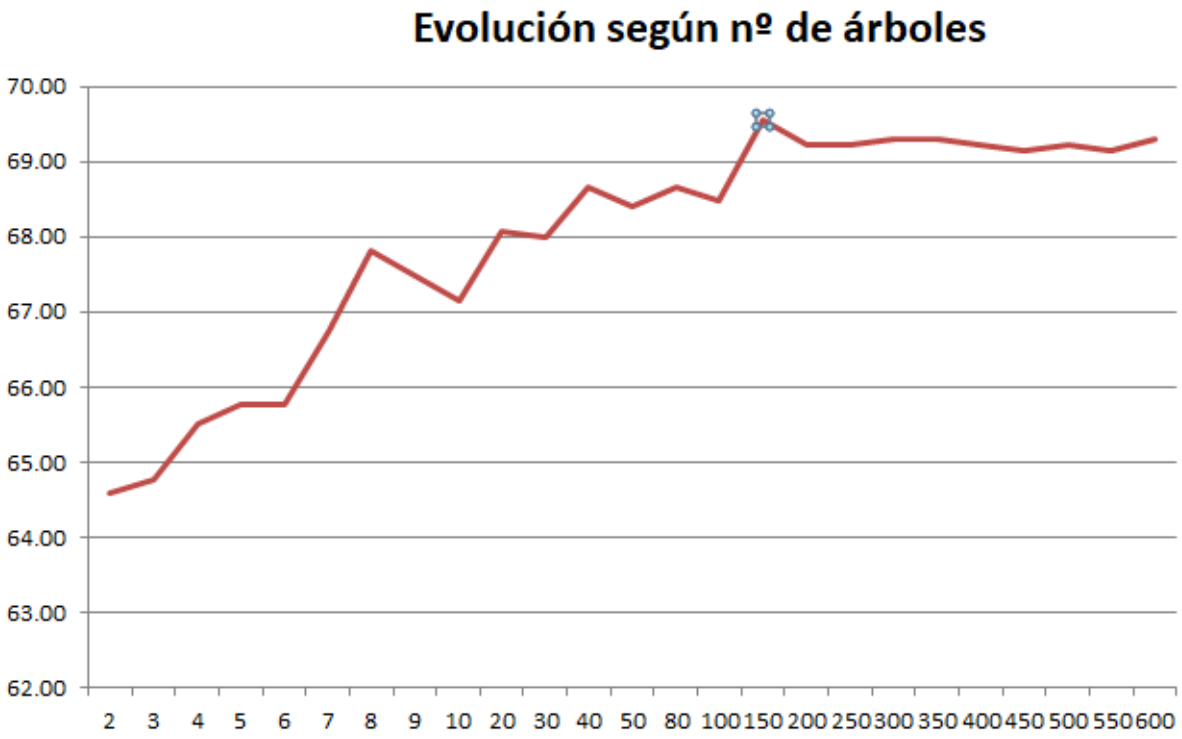

Figura 14: Evolución Random Forest por número de árboles

Tras la realización del experimento, se puede comprobar una tendencia realmente positiva conforme aumenta el número de árboles del bosque pero, una vez sobrepasa los 200 árboles el clasificador no continúa mejorando, por lo que no conviene sobrepasar esta cantidad. Tal y como se puede apreciar y, tal y como se adelantaba anteriormente, el pico más elevado de acierto se alcanza cuando el número de estimadores es 150, con un 69.6%.

4.3.3 Importancia de las cuotas de las casas de apuestas

En este apartado se va a estudiar qué tan acertadas son las casas de apuestas a la hora de establecer una cuota a cada equipo y, por lo tanto, qué importancia cobran estas cuotas en el modelo. A continuación, se va a mostrar un gráfico en el que el eje X representa la cuota de las casas de apuestas del equipo local y el eje Y el porcentaje de victorias del equipo local en su propio estadio. Las marcas azules corresponden a los partidos ganados por el equipo local y las rojas a los partidos ganados por el equipo visitante.

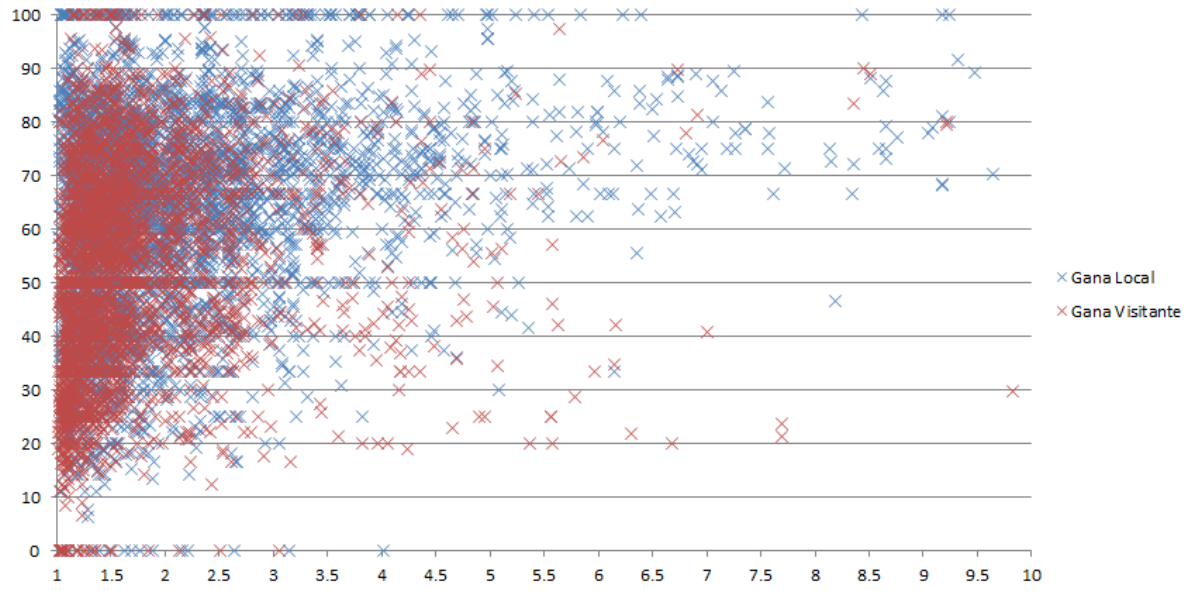

Figura 15: Importancia cuotas casas de apuestas

A través de esta distribución se puede observar la gran relevancia que tienen las cuotas de las casas de apuestas y obtener las siguientes conclusiones:

- Los partidos en los que la cuota del equipo local era muy baja (alto porcentaje de probabilidad) y mayor es el porcentaje de victorias del equipo local en su estadio suelen ser ganados por el local. Como se explicó anteriormente, el equipo local gana aproximadamente un 60% de los partidos, por ello resulta normal que la tendencia se extienda hasta los equipos que poseen cerca de un 40% de victorias como local.
- Otro dato a tener en cuenta es que cuanto mayor es el porcentaje de victorias en su campo, los partidos suelen ser ganados por estos equipos aunque la cuota sea muy elevada. Por otro lado, cuanto menor es el porcentaje de victorias del local y mayor es su cuota, los partidos suelen ser ganados por el equipo visitante.

Se puede llegar a la conclusión de que los atributos que recogen las cuotas de las casas de apuestas para los partidos son un buen indicador global que ya recoge el rendimiento de ambos equipos. Para comprobar la importancia de estos atributos se realizará a continuación el mismo experimento de predicción, pero esta vez se excluirán los atributos referentes a las cuotas, realizando el estudio con atributos estrictamente deportivos.

Porcentajes de acierto				
Temporadas con las que entrenar	**DecisionTree**		**RandomForest**	
	Con cuotas	**Sin cuotas**	**Con cuotas**	**Sin cuotas**
2017/18	66.41%	63.28%	69.6%	67.8%
2016/17 - 2017/18	64.06%	63.52%	66’4%	66%
2015/16 - 2016/17 - 2017/18	65.66%	63.65%	67.8%	66.6%
2014/15 - 2015/16 - 2016/17 - 2017/18	64.67%	63.86%	67.1%	66.7%
2013/14 - 2014/15 - 2015/16 - 2016/17 - 2017/18	63.69%	64.5%	67.2%	67%

Tabla 6: Resultados con cuotas vs sin cuotas

Tal y como se esperaba, los resultados obtenidos tras entrenar el modelo sin hacer uso de las cuotas de las casas de apuestas son un poco más bajos, alcanzando un máximo de un 67.8%% frente al 69.6% que se obtenía anteriormente utilizando dichas tasas.

4.3.4 Rentabilidad de los resultados obtenidos en las casas de apuestas

En este apartado se va a realizar un pequeño estudio en el cual se va a comprobar cuál es la rentabilidad de las predicciones del modelo en las casas de apuestas. Para ello se va a recoger todas las predicciones obtenidas y apostar 10€ de forma virtual al ganador de cada encuentro. Para ello, como ya se ha explicado anteriormente, se multiplicará 10€ por la cuota del equipo ganador.

Total de partidos	Dinero total apostado	Ganancias	Beneficio neto
1213	12130€	7776.60€	-4353.40€

Tabla 7: Rentabilidad en casas de apuestas

Como se puede comprobar, se pierde casi 4.500€ durante la temporada apostando 10€ a cada partido. Esto quiere decir que, aunque el modelo acierte cerca de un 70% de los partidos, la gran mayoría de encuentros que acierta, las casas de apuestas ya le daban un alto porcentaje de probabilidad de que gane y, por lo tanto, la cuota que se le otorga es realmente baja.

Es por esto por lo que a la larga se pierde bastante dinero ya que, a modo de ejemplo, si el modelo acierta 7 cuotas 1.30 (por ejemplo) = 91€ y le restamos los 70€ que se han apostado, hacen un total de 21€ de beneficio neto. Seguidamente el clasificador falla solamente 3 partidos y esto ya ha provocado 30€ de pérdidas. Por lo tanto, con este ejemplo, se ha acertado 7 partidos de 10 y se ha perdido 9€.

CAPÍTULO 5 Página web

Se ha procedido a realizar una página web para facilitar a los usuarios a acceder a las predicciones que nos facilita el modelo. De esta manera, todo aquel que lo desee podrá informarse sobre qué equipos se espera que ganen en los próximos días.

5.1 Herramientas y lenguajes utilizados

- **Visual Studio Code [26]:** Es un editor de programación multiplataforma desarrollado por Microsoft.
- **Html [27]:** Lenguaje de marcado para la elaboración de páginas web.
- **Css [278]:** Lenguaje utilizado para definir el estilo de las páginas web.
- **Django [29]:** Es un framework de aplicaciones web gratuito y de código abierto (open source) escrito en Python, de esta manera nos facilita la integración de nuestro código de predicciones ya previamente realizado con el lenguaje de programación Python.
- **Heroku [30]:** Es una plataforma que nos ofrece servicios de servidores y redes en donde se pueden alojar aplicaciones de diferentes lenguajes de programación como Python, Java, PHP, etc.
- **PhpMyAdmin[31]:** Herramienta con la que podremos manejar y administrar las bases de datos MySQL. Se pueden crear, eliminar y modificar bases de datos así como gestionar las tablas de las mismas.

5.2 Ficheros y funcionamiento

- **index.py:** Archivo principal en el cual se aloja todo el funcionamiento de la página web referido con la realización de las consultas a la base de datos y la distribución y manejo de las distintas rutas que componen la página.
- **layout.html:** Es una plantilla para el diseño de las dos rutas que componen la página web. Una cosa muy buena que tiene Django es la extensión de plantillas. Esto quiere decir que se puede reusar partes del código HTML para diferentes páginas del sitio web. De esta manera, no tienes que repetir el mismo código una y otra vez y si quieres cambiar algo no tienes que hacerlo en cada página, solo en una.
- **home.html:** Página en la cual se puede visualizar las predicciones del modelo.

- **nosotros.html:** Página de “bienvenida” en la que especifica quién ha realizado esta página y su propósito.
- **scrap.py:** Se ha creado un programa para facilitar la recogida de datos para el funcionamiento de la página. Este programa accede a la página web de Basketball Reference mediante el uso de BeautifulSoup [32] y se trae todos aquellos partidos de las temporadas que se le especifique. Esta técnica se denomina “Web Scraping” [33], la cual es muy útil para extraer información de sitios web.

  La idea es traer los partidos de todas las temporadas previas una sola vez, almacenarlas en una base de datos y solamente actualizar los partidos de la temporada actual, ya que en la NBA, prácticamente cada noche se disputa una serie de encuentros. Por lo tanto, se ha programado en Heroku para que se ejecute cada mañana, una vez se hayan actualizado los resultados de la noche anterior. De esta manera, cada mañana podrá realizar las predicciones con todos los datos actualizados y almacenar dichas predicciones en una base de datos.

  De esta base de datos que se mencionaba anteriormente, la página web recogerá las predicciones que el usuario desee visualizar. Para ello podrá hacer uso de un calendario para así poder buscar los partidos por fecha de encuentro. Seguidamente se le mostrará una tabla en la que se especifica la fecha del partido, quiénes son los equipos local y visitante y, cuál de estos dos conjuntos se alzará con la victoria.

A continuación se mostrará la interfaz de la página web realizada para el proyecto:

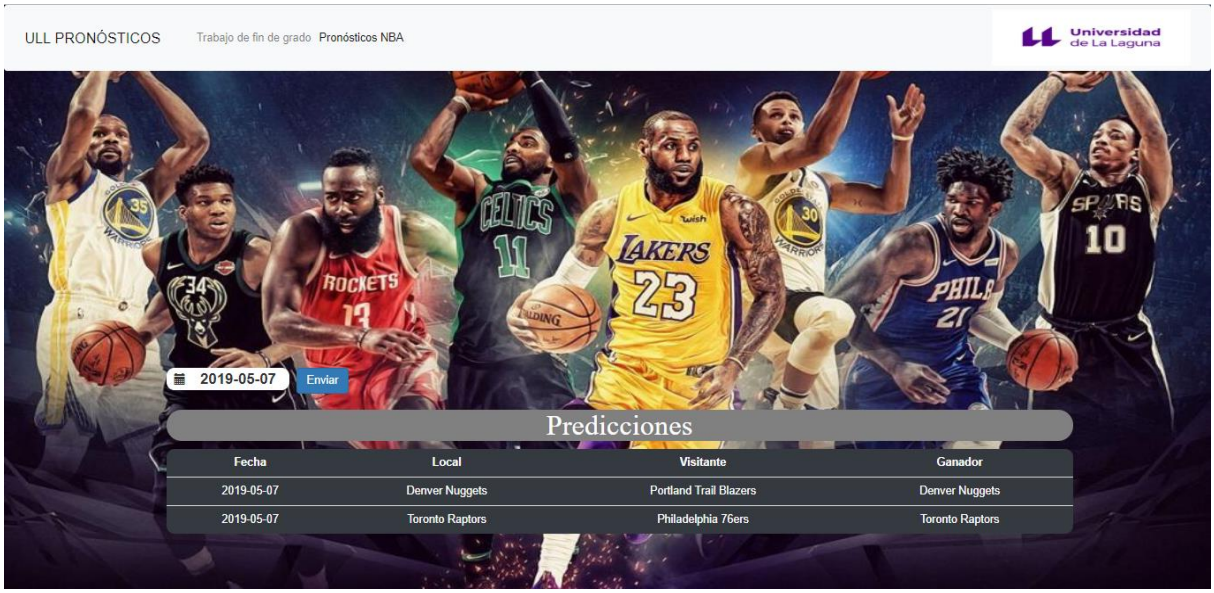

Figura 16: Interfaz Web

CAPÍTULO 6 Conclusiones

Mediante el uso de técnicas de aprendizaje automático se ha conseguido dar respuesta a la pregunta principal de este proyecto: ¿Quién ganará el siguiente partido? Durante el desarrollo y una vez finalizado el mismo, se ha podido extraer las siguientes conclusiones.

En primer lugar, tras recopilar y pre-procesar todos los datos desde la temporada 2013/14 hasta la temporada actual (2018/19), se ha establecido un conjunto de datos compuesto por 7280 partidos. Se ha utilizado la temporada 2017/18 para el entrenamiento del modelo (1212 instancias) y la temporada 2018/19 para realizar el testeo (1212 encuentros).

De ambas técnicas empleadas, mediante la técnica Random Forest es a través de la cual se ha obtenido mayor porcentaje de acierto, llegando al 69.6%. Es decir, de los 1212 partidos que se han comprobado, 843 fueron clasificados correctamente y 369 incorrectamente. La configuración óptima del modelo es con 150 árboles y una máxima profundidad de 7 niveles. Sin embargo, se ha podido comprobar que una vez el bosque sobrepasa los 200 árboles, el clasificador no continúa mejorando.

Este porcentaje de acierto, comparado con otros modelos de aprendizaje automático, es realmente bajo. No obstante, hay que tener en cuenta que se está tratando con el rendimiento de deportistas, lo que implica un peso importante del factor humano y dificulta la predicción.

Como se ha comprobado anteriormente, las cuotas de las casas de apuestas juegan un papel fundamental a la hora de ayudar a predecir nuevos resultados. Por lo tanto, incluso si solo se toman atributos estrictamente deportivos, se debe dar un paso más allá e incluir atributos de obtención más compleja, como los jugadores lesionados en cada partido, los rendimientos ofensivos de cada jugador en función de los defensores contra los que se enfrentan, la racha de victorias de un equipo contra otro (debido a que hay equipos especialmente efectivos contra rivales específicos), etc.

Una de las conclusiones más importantes de este proyecto es que por más datos que se le proporcione al modelo para el entrenamiento, no necesariamente va a mejorar. Los mejores resultados han sido obtenidos entrenando únicamente con la temporada previa y, como ya se ha explicado anteriormente, es posible que se deba a la inestabilidad de las plantillas de la NBA debido a los límites salariales. En deportes con plantillas más estables como el fútbol el entrenamiento con conjuntos masivos de datos podría ayudar a la predicción.

CHAPTER 6 Conclusions

The answer to the question "*Who will win the next match?* " has been tried to be elucidated through the use of automated learning techniques. The following conclusions have been drawn.

After collecting and pre-processing all data from the 2013/14 season to the current season (2018/19), a dataset of 7280 matches has been established. The 2017/18 season was used for training the model (1212 matches) and the 2018/19 season for testing (1212 matches).

The Random Forest technique is the one with the highest percentage of success, reaching 69.6%. From the 1212 matches verified, 843 were correctly classified while 369 wer incorrectly classified. The optimal configuration of the model is with 150 trees and a maximum depth of 7 levels. However, it has been proven that once the forest exceeds 200 trees, the classifier does not continue to improve.

This success rate is rather low compared to other automatic learning models. However the performance of sportspeople bears a strong human factor burden, thus making prediction difficult.

As previously stated, bookmaker odds play a key role in the outcome prediction. Therefore, even only considering strictly sports-related issues, more elaborated winning attributes should be taken into account, such as the players injured in each match, the offensive performances of each player in relation with the defenders they face, the winning streak of one team against another (because there are teams particularly well-fit to play against another specific one), and so on.

One of the most important conclusions that the improvement of the training model is not necessarily related to the amount of data provided. The best results have been obtained using data only form the the previous season and, as explained above, this may be related to the instability of the NBA squads due to salary limits. Sports such as football, where sporting staffs are more stable, training with massive data sets may increase predictability.

CAPÍTULO 7 Líneas futuras

Como se ha explicado anteriormente, este experimento dispone de un amplio margen de mejora, pero se ve envuelto por un dominio repleto de atributos y comportamientos humanos impredecibles. Asimismo, el desarrollo del proyecto se ha visto dificultado por la inexperiencia con técnicas de aprendizaje automático.

Para la realización de este proyecto se ha asumido que los datos referentes a los rendimientos medios de los equipos son un buen estimador para realizar las oportunas comparaciones entre equipos. Sin embargo, un análisis profundo de los rendimientos individuales puede aportar nuevas características al modelo. Sería

interesante recolectar información referida a zonas habituales del campo donde los jugadores efectúan sus lanzamientos a canasta o, por ejemplo, la altura de sus jugadores, debido a que puede haber equipos con jugadores con una media de altura baja que no les resulte fácil jugar contra equipos con una plantilla alta.

Por otro lado, tal y como se mencionaba anteriormente, sería muy interesante poder recoger los jugadores que son baja en un equipo para un partido en concreto. Esto es realmente complicado, debido a que en la NBA solamente una o dos horas antes del comienzo del partido cada equipo anuncia mediante Twitter los jugadores que no estarán disponibles para esa fecha. Para recoger dicha información se podría intentar utilizar la API Tweepy [34] ; aunque esta búsqueda de información conllevaría mucho tiempo y sólo permitiría obtener predicciones con muy poca antelación.

Por último, considero oportuno mencionar que este tipo de problema podría obtener grandes mejoras mediante el uso de redes neuronales. Keras [35] es una librería de TensorFlow que nos proporciona una serie de capas interesantes para dichas redes. Considero la más interesante para este problema la capa LSTM (Long Short Term Memory), aplicable sobre todo en el campo del análisis de series temporales. A través de dicha capa se podría obtener, por ejemplo, una valoración de las dinámicas previas de los equipos que se enfrentan en un partido.

CHAPTER 7 Future lines

As previously stated, this project deals with an environment heavily influenced by human behaviors and somehow unpredictable human attributes and. Therefore, the room for improvement is rather wide. The development of the project has also been hampered by inexperience with automatic learning techniques.

In order to carry out this project, it has been assumed that data related to the average performance of the teams are a good estimator in order to make the straightforward comparisons between teams. However, an in-depth analysis of the individual performances may widen the scope of the model. It would be interesting to collect information referring to habitual zones of the field where players try their essays or the average height of the players, as teams employing players with lower average stature may be in disadvantage when playing against teams with a taller squad.

On the other hand, as mentioned above, it would be very interesting to gather information about handicapped players before a particular match. This is really complicated, as this information is released via Tweeter only a couple of hours before the NBA game. The Tweepy API [34] could be useful for gathering such information, although this query would be time-consuming and only allow very short-range predictions.

Finally, it is worth mentioning that this kind of problem could be effectively approached through the use of neural networks. Keras [35] is a TensorFlow library that provides with a series of interesting layers for these networks. I consider the best suited for this problem the LSTM (Long Short Term Memory) layer, strongly applicable in the field of time series analysis. Through this layer we could obtain, for example, an assessment of the previous dynamics of the teams that are facing each other in a given match.

CAPÍTULO 8 Bibliografía

[1] "Welcome to Python.org," *Python.org*. [Online]. Available: https://www.python.org/.

[2] "1.10. Decision Trees," *scikit*. [Online]. Available: https://scikit-learn.org/stable/modules/tree.html.

[3] "3.2.4.3.1. sklearn.ensemble.RandomForestClassifier," *scikit*. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html.

[4] "METODOLOG," *Google Libros*. [Online]. Available: https://books.google.es/books?hl=es&lr=&id=GuQNvW3Te1sC&oi=fnd&pg=PA5&dq=baloncesto&ots=1n9W3Pkr9X&sig=gncLsbg6QD0BkUM5ntRHlONawIE#v=onepage&q=baloncesto&f=false.

[5] "Reglamento de baloncesto comentado," *Google Libros*. [Online]. Available: https://books.google.es/books?hl=es&lr=&id=yTEZ9lMqzH4C&oi=fnd&pg=PA37&dq=reglamento+nba&ots=00YljxeHJO&sig=oZBeVxEM_3RVRHJLjSYt6ZTmC4A#v=onepage&q=reglamento nba&f=false.

[6] "Clasificación NBA por conferencias y reglas desempate," *nbamaniacs*.

[Online]. Available: https://www.nbamaniacs.com/tabla-clasificacion-nba/.

[7] “Los orígenes de las Apuestas Deportivas,” *Casas de Apuestas*, 28-Aug-2015. [Online]. Available: https://www.casasdeapuestas.com/los-origenes-de-las-apuestas-deportivas/.

[8] Diario16, “¿Cómo funcionan las casas de apuestas?,” *Diario16*, 03-Oct-2018. [Online]. Available: https://diario16.com/funcionan-las-casas-apuestas/.

[9] “Apuestas deportivas online,” *Google Libros*. [Online]. Available: https://books.google.es/books?hl=es&lr=&id=cCzoPnf9wTwC&oi=fnd&pg=PT3&dq=formato+de+cuotas+casas+de+apuestas&ots=s6djJfKUOy&sig=pYzmTRrXOo-8d2-WBvqxclm9wMo#v=onepage&q=formato de cuotas casas de apuestas&f=false.

[10] “Big data : la revoluci,” *Google Libros*. [Online]. Available: https://books.google.es/books?id=uO9FbEcaMpkC&printsec=frontcover&dq=big+data&hl=es&sa=X&ved=0ahUKEwjR5_C5gdDiAhU8AGMBHcGKCC4Q6AEIKDAA#v=onepage&q=big data&f=false.

[11] “Mastering Predictive Analytics with R,” *Google Libros*. [Online]. Available: https://books.google.es/books?id=SJZGDwAAQBAJ&pg=PA383&dq=doug+laney+big+data+three+v's&hl=es&sa=X&ved=0ahUKEwj-uNDngdDiAhUE-hQKHdk2AOoQ6AEIKzAA#v=onepage&q=doug laney big data three v's&f=false.

[12] “Miner,” *Google Libros*. [Online]. Available: https://books.google.es/books?hl=es&lr=&id=wz-D_8uPFCEC&oi=fnd&pg=PR4&dq=mineria+de+datos&ots=TiY6zl4v3G&sig=U

F1cAuwHlSXHwjzfny6oJhYahWY#v=onepage&q=mineria de datos&f=false.

[13] *Javier Landa*. [Online]. Available: http://fcojlanda.me/es/ciencia-de-los-datos/kdd-y-mineria-de-datos-espanol/.

[14] “Basketball Statistics and History,” *Basketball*. [Online]. Available: https://www.basketball-reference.com/.

[15] “OddsPortal,”. [Online]. Available: https://www.oddsportal.com/

[16] “powerful Python data analysis toolkit,” *pandas*. [Online]. Available: https://pandas.pydata.org/pandas-docs/stable/.

[17] “Project Jupyter,” *Project Jupyter*. [Online]. Available: https://jupyter.org/.

[18] “NumPy,” *NumPy*. [Online]. Available: https://www.numpy.org/.

[19] “learn,” *scikit*. [Online]. Available: https://scikit-learn.org/stable/.

[20] B. de Arquitectura, “Menos es más, las mejores frases de Mies van der Rohe,” *Menos es ms, las mejores frases de Mies van der Rohe - Noticias de Arquitectura - Buscador de Arquitectura*. [Online]. Available: http://noticias.arq.com.mx/Detalles/17325.html#.XPaDCYhKjIU.

[21] “sklearn.preprocessing.LabelEncoder,” *scikit*. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html.

[22] “sklearn.model_selection.GridSearchCV,” *scikit*. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html.

[23] “3.1. Cross-validation: evaluating estimator performance,” *scikit*. [Online]. Available: https://scikit-learn.org/stable/modules/cross_validation.html.

[24] C. Ochoa, “Over-fitting: el enemigo de las buenas predicciones,” *Netquest*. [Online]. Available: https://www.netquest.com/blog/es/blog/es/over-fitting-enemigo-buenas-predicciones.

[25] F. S. Caparrini, “Aprendizaje Inductivo: Árboles de Decisión,” *Aprendizaje Inductivo: Árboles de Decisión - Fernando Sancho Caparrini*. [Online]. Available: http://www.cs.us.es/~fsancho/?e=104.

[26] “Visual Studio Code - Code Editing. Redefined,” *RSS*, 14-Apr-2016. [Online]. Available: https://code.visualstudio.com/.

[27] “HTML5 Tutorial,” *HTML Tutorial*. [Online]. Available: https://www.w3schools.com/html/default.asp.

[28] *CSS Tutorial*. [Online]. Available: https://www.w3schools.com/css/.

[29] “Django,” *Django*. [Online]. Available: https://www.djangoproject.com/.

[30] “Heroku,” *Cloud Application Platform*. [Online]. Available: https://www.heroku.com/.

[31] phpM. A. contributors, “phpMyAdmin,” *phpMyAdmin*. [Online]. Available: https://www.phpmyadmin.net/.

[32] “Beautiful Soup Documentation,” *Beautiful Soup Documentation - Beautiful Soup 4.4.0 documentation*. [Online]. Available: https://www.crummy.com/software/BeautifulSoup/bs4/doc/.

[33] “Web Scraping with Python,” *Google Libros*. [Online]. Available: https://books.google.es/books?hl=es&lr=&id=TYtSDwAAQBAJ&oi=fnd&pg=PT8&dq=web+scraping&ots=y0y5yFshcl&sig=W6ACixUr9HlxfW1V9WzBsENPUfo#v=onepage&q=web scraping&f=false.

[34] “Tweepy,” *Tweepy*. [Online]. Available: https://www.tweepy.org/.

[35] “Keras | TensorFlow Core | TensorFlow,” *TensorFlow*. [Online].

Available: https://www.tensorflow.org/guide/keras.