



universidad
de león



Escuela de Ingenierías Industrial, Informática y Aeroespacial

GRADO EN INGENIERÍA INFORMÁTICA

Trabajo de Fin de Grado

**MODELO PREDICTIVO DE MACHINE LEARNING
APLICADO AL DEPORTE**

**PREDICTIVE MODEL OF MACHINE LEARNING APPLIED
TO SPORTS**

Autor: Jorge Corral Losada
Tutor: Manuel Castejón Limas

(Septiembre, 2021)

UNIVERSIDAD DE LEÓN
Escuela de Ingenierías Industrial, Informática y
Aeroespacial

GRADO EN INGENIERÍA INFORMÁTICA
Trabajo de Fin de Grado

ALUMNO: Jorge Corral Losada

TUTOR: Manuel Castejón Limas

TÍTULO: Modelo predictivo de Machine Learning aplicado al deporte

TITLE: Predictive model of Machine Learning applied to sports

CONVOCATORIA: Septiembre, 2021

RESUMEN:


El proyecto planteado en este Trabajo de Fin de Grado es la creación de un modelo predictivo, a través de técnicas de Machine Learning, capaz de seleccionar correctamente el mayor número posible de jugadores que serán elegidos para participar en el All-Star de la NBA. Se ha conseguido crear un clasificador que posee un 95,83% de acierto, prediciendo correctamente 23 jugadores de los 24 seleccionados. La memoria consta de: una introducción, donde se explica y contextualiza el trabajo; el estado del arte, donde se comenta y compara este trabajo con otros similares; los objetivos que han de llevarse a cabo a lo largo del proyecto; la planificación que se ha seguido para la consecución del proyecto; el núcleo del trabajo, donde se explica en detalle cada una de las partes que componen el proyecto; los resultados de todo el trabajo realizado así como las conclusiones obtenidas tras haber realizado el trabajo. Podemos considerar como muy satisfactorio el resultado final del proyecto ya que se ha conseguido obtener un modelo predictivo con una alta tasa de acierto, mejor que cualquiera de los existentes en la actualidad.

ABSTRACT:

The project proposed in this Final Degree Project is the creation of a predictive model, through Machine Learning techniques, capable of correctly selecting the largest possible number of players who will be chosen to participate in the NBA All-Star Game. It has been possible to create a classifier with a 95.83% accuracy rate, correctly predicting 23 players out of the 24 selected. The report consists of: an introduction, where the work is explained and contextualized; the state of the art, where this work is commented and compared with other similar works; the objectives to be carried out throughout the project; the planning that has been followed for the achievement of the project; the core of the work, where each of the parts that make up the project is explained in detail; the results of all the work done as well as the conclusions obtained after having carried out the work. We can consider the final result of the project as very satisfactory, since it has been possible to obtain a predictive model with a high accuracy rate, better than any of the actual existing models.

Palabras clave: Inteligencia Artificial, Machine Learning, modelo predictivo, dataset, baloncesto.

Firma del alumno:



VºBº Tutor/es:

Índice de contenidos

| | |
|--|----|
| Índice de contenidos | 4 |
| Índice de figuras | 5 |
| Índice de cuadros y tablas | 7 |
| Glosario de signos, símbolos, unidades, abreviaturas, acrónimos o términos | 8 |
| 1. Introducción | 10 |
| 2. Estado del arte..... | 12 |
| 3. Objetivos..... | 13 |
| 4. Planificación..... | 15 |
| 5. Legislación y normativa | 16 |
| 6. Cuerpo del trabajo | 17 |
| 6.1. TECNOLOGÍAS UTILIZADAS..... | 17 |
| 6.2. CONJUNTO DE DATOS | 18 |
| 6.3. PREPROCESAMIENTO DE LOS DATOS | 20 |
| 6.4. MODELOS PREDICTIVOS | 30 |
| 6.5. RESULTADOS | 38 |
| 7. Conclusiones y recomendaciones | 55 |
| Lista de referencias bibliográficas | 56 |

Índice de figuras

| | |
|---|----|
| Figura 4.1. – Diagrama de Gantt del proyecto..... | 15 |
| Figura 6.1. – Relación de entradas positivas y negativas del dataset..... | 19 |
| Figura 6.2 – Conjunto de datos de entrenamiento | 19 |
| Figura 6.3. – Conjunto de datos a clasificar | 20 |
| Figura 6.4. – Relación de positivos y negativos en función de su GP%..... | 22 |
| Figura 6.5. – Pesos de las estadísticas en la creación del modelo | 23 |
| Figura 6.6. - Relación de positivos y negativos en función del ranking de su equipo | 24 |
| Figura 6.7. - Relación de positivos y negativos en función del número de victorias | 24 |
| Figura 6.8. - Relación de positivos y negativos en función de sus puntos | 25 |
| Figura 6.9. - Relación de positivos y negativos en función de sus rebotes | 25 |
| Figura 6.10. - Relación de positivos y negativos en función de sus asistencias .. | 25 |
| Figura 6.11. - Relación de positivos y negativos en función de sus robos | 26 |
| Figura 6.12. - Relación de positivos y negativos en función de sus bloqueos..... | 26 |
| Figura 6.13. - Relación de positivos y negativos en función de sus pérdidas | 27 |
| Figura 6.14. - Relación de positivos y negativos en función de sus triples..... | 28 |
| Figura 6.15. - Relación de positivos y negativos en función de su PIE | 28 |
| Figura 6.16. - Relación de positivos y negativos en función de su USG% | 29 |
| Figura 6.17. - Relación de positivos y negativos en función de su TS% | 29 |
| Figura 6.18. - Relación de positivos y negativos en función de su DEFWS | 30 |
| Figura 6.19. – Métricas de los algoritmos calculados por LazyPredict | 31 |
| Figura 6.20. – Ejemplo de LDA | 32 |
| Figura 6.21. – Ejemplo de LDA | 33 |
| Figura 6.22. – Ejemplo de KNN..... | 33 |
| Figura 6.23. – Ejemplo de funcionamiento de un clasificador de tipo Bagging | 34 |
| Figura 6.24. – Ejemplo de Random Forest..... | 35 |
| Figura 6. 25. – Ejemplo de AdaBoost..... | 36 |
| Figura 6.26. – Ejemplo de Gradient Boosting..... | 36 |
| Figura 6.27. – Ejemplo de SVM..... | 37 |
| Figura 6.28. - Ejemplo de red neuronal Multi-Layer Perceptron | 38 |
| Figura 6 29. - Matriz de confusión | 39 |
| Figura 6.30. – Métricas del modelo creado mediante el algoritmo ExtraTrees..... | 41 |
| Figura 6.31. – Pesos de las variables en la creación del modelo..... | 42 |
| Figura 6.32. – Posibilidades de ser All-Star en la conferencia Este | 43 |
| Figura 6.33. – Posibilidades de ser All-Star en la conferencia Este | 44 |
| Figura 6.34. - Posibilidades de ser All-Star en la conferencia Oeste..... | 45 |
| Figura 6.35. - Posibilidades de ser All-Star en la conferencia Oeste..... | 45 |
| Figura 6. 36. Métricas del modelo creado mediante el algoritmo Random Forest | 46 |
| Figura 6.37. - Pesos de las variables en la creación del modelo..... | 47 |
| Figura 6.38. - Posibilidades de ser All-Star en la conferencia Este | 48 |
| Figura 6.39. - Posibilidades de ser All-Star en la conferencia Este | 49 |

| | |
|---|----|
| Figura 6.40. - Posibilidades de ser All-Star en la conferencia Oeste..... | 50 |
| Figura 6.41. - Posibilidades de ser All-Star en la conferencia Oeste..... | 51 |
| Figura 6.42. - Métricas del modelo creado mediante redes neuronales..... | 51 |
| Figura 6.43. - Posibilidades de ser All-Star en la conferencia Este | 52 |
| Figura 6.44. - Posibilidades de ser All-Star en la conferencia Este | 53 |
| Figura 6.45. - Posibilidades de ser All-Star en la conferencia Oeste..... | 54 |
| Figura 6.46. - Posibilidades de ser All-Star en la conferencia Oeste..... | 54 |

Índice de cuadros y tablas

| | |
|---|----|
| Tabla 6.1. – Matriz de confusión del modelo | 41 |
|---|----|

Glosario de signos, símbolos, unidades, abreviaturas, acrónimos o términos

NBA: Liga nacional de baloncesto de Estados Unidos.

All-Star Game (ASG): partido de carácter amistoso que se celebra durante el All-Star Weekend de la NBA, en el que participan solo los mejores jugadores de cada año, divididos en dos equipos que representan a cada conferencia (Este y Oeste).

Conferencia: Cada una de las dos partes en las que está dividida la NBA. Las conferencias que existen son la Este y la Oeste. Cada una de ellas está compuesta por quince equipos y organizadas en tres divisiones de cinco equipos cada una. Los ocho primeros equipos de cada conferencia se clasifican para disputar los playoffs.

Ritmo: Número de posesiones por cada 48 minutos.

GP: Partidos jugados.

W: Victorias.

PTS: Puntos por partido.

REB: Rebotes por partido.

AST: Asistencias por partido.

STL: Robos por partido.

BLK: Bloqueos por partido

TOV: Pérdidas por partido.

3PM: Triples anotados por partido.

FGA: Tiros de campo intentados.

FTA: Tiros libres intentados.

Porcentaje de Tiro Verdadero (TS%): Una forma de medir la eficiencia en el tiro tomando en cuenta los tiros acertados de 2 puntos, 3 puntos y tiros libres dándole su correspondiente importancia a cada uno de ellos.

$$TS\% = \frac{Puntos}{2 * (FGA + 0,44 FTA)}$$

Uso Ofensivo (USG%): Porcentaje de las jugadas ofensivas del equipo en las cuales está involucrado un jugador cuando está en la pista. Se calcula a partir de las jugadas que terminan con tiro, pérdida o falta a favor.

$$USG\% = \frac{FGA + 0,44 FTA + TOV}{Possession}$$

PIE: Es una estimación del impacto que tiene un jugador cuando juega. Es una fórmula en la que se usan todas las estadísticas que un jugador ha generado con respecto a los parámetros globales del partido.

Defensive Win Shares (DEFWS): Es una estadística que otorga valor a los jugadores en función de su capacidad para evitar que los equipos rivales anoten. Se calcula estimando la cantidad de puntos permitidos por cada 100 posesiones defensivas.

Outlier: Es un valor dentro de un conjunto de datos que es muy diferente al resto de los valores.

Intangible: Es aquella acción que proporciona un beneficio al equipo y que no se ve reflejada en las estadísticas oficiales.

Draft: Es el mecanismo por el cual todas las franquicias de la NBA incorporan a sus equipos jugadores procedentes de las universidades de EE. UU. o de ligas nacionales de baloncesto de otros países.

1. Introducción

Desde un primer momento tenía claro que el área de conocimiento sobre la que iba a tratar este trabajo era la Inteligencia Artificial y más concretamente el Machine Learning. Tras haber realizado en el pasado varios trabajos de procesamiento de textos, audios e imágenes, para este proyecto quería explorar algo nuevo y profundizar en la Ciencia de Datos. La decisión final no fue muy complicada, debía juntar dos pasiones como son el deporte y la Estadística.

Finalmente, el deporte seleccionado ha sido el baloncesto, y más concretamente la NBA. El motivo de ello podemos repartirlo entre los estadounidenses, quienes son unos entusiastas de las estadísticas y registran absolutamente todo lo que ocurre en la liga, y el confinamiento, que ha despertado en mí una pasión por el baloncesto que hasta hace un año no existía, y más concretamente, por todo el mundo que rodea a la NBA.

El baloncesto como deporte en general y la NBA en especial, cada vez está utilizando más las estadísticas para tomar sus decisiones tanto dentro como fuera del campo. Pero aún existe una gran cantidad de entidades y cuerpos técnicos que son reticentes a aceptar lo que es una realidad: acercar los datos al deporte puede ser crucial para alcanzar el éxito. Y es que, el (mal) llamado Big Data en el deporte ha cambiado la NBA en los últimos años. Sin ir más lejos, las franquicias vieron que los triples eran mucho más rentables que cualquier otro tiro, ya que un 34% de acierto en tiros de tres resulta más efectivo que un 50% de acierto en tiros de dos, y actualmente más de la mitad de la liga supera ese porcentaje de acierto en los triples.

En el análisis predictivo se utilizan datos históricos junto con métodos matemáticos y técnicas de aprendizaje automático para crear un modelo predictivo, de manera que al usar los datos actuales con él podemos predecir qué es lo que pasará en un futuro.

El objetivo del proyecto ha sido el de crear un modelo predictivo que clasifique a los jugadores en función de si van a ser seleccionados para el All-Star Game. El conocido como partido de las estrellas de la NBA es un partido amistoso

que se celebra durante el All-Star Weekend de la NBA, en el que participan solo los mejores jugadores de lo que va de temporada de cada Conferencia. La selección de dichos jugadores se divide en dos: titulares y suplentes; 5 titulares y 7 suplentes por equipo. La elección de los titulares corre a cargo del público, quien representa un 50% del peso total de los votos para cada posible titular, la prensa y los propios jugadores, quienes representan un 25% cada uno. Los suplentes son elegidos por los entrenadores de cada equipo. En total son seleccionados 24 jugadores, 12 de la conferencia Este y otros tantos de la Oeste.

En primera instancia se necesitará crear un conjunto de datos completo, con todas las entradas clasificadas en All-Star o no All-Star. Tras realizar un procesamiento de los datos, se obtendrá el dataset final con el que entrenar y testear el modelo predictivo. Finalmente, después de utilizar varios clasificadores distintos, se ha conseguido crear un modelo capaz de seleccionar correctamente a 23 jugadores de los 24 totales, es decir, un 95,83% de acierto.

2. Estado del arte

Las técnicas de Machine Learning cada vez son más utilizadas en el mundo del deporte. La estadística avanzada actualmente es usada en muchos deportes y para gran cantidad de ámbitos dentro de estos. La primera aplicación de algo parecido es el conocido Moneyball [1], cuando el gerente de los Oakland Athletics, la franquicia de la liga nacional de béisbol de Estados Unidos, utilizó la estadística avanzada para mejorar el equipo y llevarlos, tras conseguir una racha histórica, a las eliminatorias de la liga.

Cada vez se recogen más datos y estadísticas, repartidos en multitud de fuentes, y muchas de ellas de libre acceso, como son Nba.stats [2] o Basketball Reference [3]. Gracias a la gran cantidad de datos que poseemos en la actualidad, la labor de análisis a través del Machine Learning se ha simplificado y ampliado en gran medida.

Se han encontrado varios proyectos [4][5][6] que utilizan modelos predictivos para calcular diferentes resultados futuros relativos a la NBA. Concretamente dos de ellos, están dedicados al cálculo de All-Stars. El primero [7] ha conseguido, utilizando técnicas tanto de aprendizaje supervisado como no supervisado, predecir 20 jugadores correctamente del total de 24 jugadores seleccionados. El segundo de los modelos [8] utiliza XGBoost, un algoritmo que se basa en los árboles de decisiones y que utiliza la potenciación de gradientes, siendo capaz de acertar 22 jugadores de los 24 totales.

3. Objetivos

- **Objetivo principal 1:**

Obtención de un modelo predictivo funcional capaz de clasificar a los jugadores de la NBA según sus estadísticas, escogiendo aquellos que serán All-Stars.

- Tarea 1:

Conseguir un buen modelo de clasificación que posea un alto porcentaje de éxito en su tarea.

- Tarea 2:

Evaluación del rendimiento final del modelo y análisis de los resultados obtenidos.

- Tarea 3:

Lograr un modelo que posea una tasa de acierto mayor a cualquiera de los que existen actualmente sobre este tema.

- **Objetivo principal 2:**

Aprendizaje y dominio de las herramientas y tecnologías necesarias para llevar a cabo la épica del proyecto.

- Tarea 1:

Creación de varios modelos de clasificación mediante Scikit-Learn y comprensión del funcionamiento de los algoritmos utilizados.

- Tarea 2:

Conocer las métricas que pueden ser aplicadas a los modelos del proyecto y ser capaz de seleccionar aquellas que mejores resultados proporcionan.

- Tarea 3:

Creación de un modelo basado en redes neuronales.

- Tarea 4:

Aprendizaje de librerías para la visualización de datos y correcta selección e interpretación de las gráficas creadas.

- **Objetivo principal 3:**

Aprender y conocer los procedimientos, métodos y pasos necesarios para la creación de un modelo predictivo.

- Tarea 1:

Obtención y elaboración de un conjunto de datos completo que contenga todas las estadísticas necesarias para la construcción del clasificador.

- Tarea 2:

Llevar a cabo una correcta limpieza, normalización, transformación y selección de los datos.

- Tarea 3:

Construcción de los modelos predictivos que se aplicarán al conjunto de datos.

- Tarea 4:

Interpretación y análisis de los resultados obtenidos.

5. Legislación y normativa

Debido al carácter público de los datos utilizados y de las técnicas empleadas, el trabajo no se ve afectado por aspectos de confidencialidad ni tampoco por la ley de protección de datos.

Para la realización de este trabajo se ha seguido la siguiente normativa:

- **Normativa de Trabajos de Fin de Grado de la Universidad de León.** [9]
- **UNE 50103:1990** [10]. Documentación. Preparación de resúmenes.
- **UNE 50132:1994** [11]. Documentación. Numeración de las divisiones y subdivisiones en los documentos escritos.
- **UNE 50135:1996** [12]. Documentación. Presentación de informes científicos y técnicos.
- **IEEE Documentation Style.** [13]

6. Cuerpo del trabajo

6.1. TECNOLOGÍAS UTILIZADAS

- **Python** [14]: Es un lenguaje de programación multiparadigma, ya que soporta parcialmente la orientación a objetos, programación imperativa y, en menor medida, programación funcional.
- **Scikit-learn** [15]: Es una librería de código abierto para el aprendizaje automático que soporta aprendizaje supervisado y no supervisado. Además, posee varias herramientas que permiten el ajuste de modelos, el preprocesamiento de datos y la evaluación y selección de modelos entre otras. Incluye varios algoritmos de clasificación, regresión y análisis de grupos. Presenta compatibilidad con otras librerías de Python como son NumPy, Pandas y matplotlib.
- **Pandas** [16]: Es una librería de código abierto escrita como extensión de Numpy que ofrece estructuras de datos y herramientas de análisis de datos de alto rendimiento y fáciles de usar para el lenguaje de programación Python. Pandas permite leer y escribir datos en diferentes formatos, seleccionar y filtrar de manera sencilla tablas de datos en función de posición o valor, insertar y eliminar columnas en estructuras de datos y mezclar y unir datos entre otras funciones.
- **Numpy** [17]: Es una librería de Python especializada en el cálculo numérico y el análisis de datos, especialmente para un gran volumen de datos. Da soporte para crear arrays y matrices que permiten representar colecciones de datos de un mismo tipo en varias dimensiones además de una gran colección de funciones matemáticas de alto nivel muy eficientes para su manipulación.
- **Matplotlib.pyplot** [18]: Es una librería de Python para la creación y visualización de gráficas estáticas, animadas e interactivas.
- **Seaborn** [19]: Es una librería de Python que sirve para la visualización de datos basada en matplotlib y con una fácil integración con estructuras de datos de Pandas. Proporciona una interfaz de alto nivel que permite generar de manera sencilla gráficos estadísticos atractivos e informativos.

- **Lazy Predict** [20]: Es una librería de Python que nos permite automatizar la tarea de encontrar el mejor algoritmo para nuestro modelo predictivo. Ejecuta 29 modelos de aprendizaje automático distintos y nos proporciona una idea de cómo funcionarán cada uno de ellos con nuestro conjunto de datos. Posee tanto un Lazy Classifier para tareas de clasificación como un Lazy Regressor para problemas de regresión. Para problemas de clasificación se proporcionan las siguientes métricas: accuracy, balanced accuracy, ROC AUC, F1 Score y el tiempo requerido para ejecutar cada uno de los modelos.

6.2. CONJUNTO DE DATOS

La obtención del dataset que se vaya a utilizar es el primer paso a la hora de crear un modelo predictivo. Cuantos más y mejores datos recolectemos, mejor será el rendimiento de nuestro modelo.

El conjunto de datos que se ha utilizado para llevar a cabo el modelo predictivo de los NBA All-Stars han sido dos archivos de extensión .csv creados a partir de los datos que se han extraído de las páginas web de la NBA y de Basketball-Reference. Los datos utilizados son aquellos referentes a la primera parte de la temporada, ya que el All-Star Game es jugado a mitad de esta. Las estadísticas que conforman este dataset son las tradicionales como *PTS*, *REB*, *AST*, *STL*, *BLK*, *TOV* y *3PM*, las avanzadas como *TS%*, *USG%* y *PIE*, y defensivas como *DEFWS*. Otras variables que conforman este dataset son el año de dichas estadísticas, así como el ritmo medio de ese año, el número de partidos jugados por el jugador y por el equipo, si el año pasado fue All-Star y la posición que ocupa el equipo de cada jugador en la clasificación de su conferencia, ya que es mucho más difícil ser seleccionado si juegas en un mal equipo, tanto por el récord de este como por la exposición mediática. Respecto a esto último, es importante destacar que juega un gran papel, pues el voto de los aficionados se tiene en cuenta para realizar la selección de jugadores. Sería complicado evaluar el impacto social que tienen los jugadores y no podríamos tener tan siquiera en cuenta el número de seguidores que poseen en las redes sociales ya que este factor es bastante

moderno. Por ello una manera de medir el estatus y la reputación de los jugadores es el número de veces que han sido All-Star anteriormente. Esta variable perjudicará a nuevos jugadores que quizás no sean tan conocidos y recompensará a superestrellas asentadas en la liga, y que año tras año son seleccionados. En el dataset que contiene los datos de entrenamiento se posee, al tratarse de un problema de clasificación, una variable que nos indica si el jugador fue All-Star ese año o no lo fue.

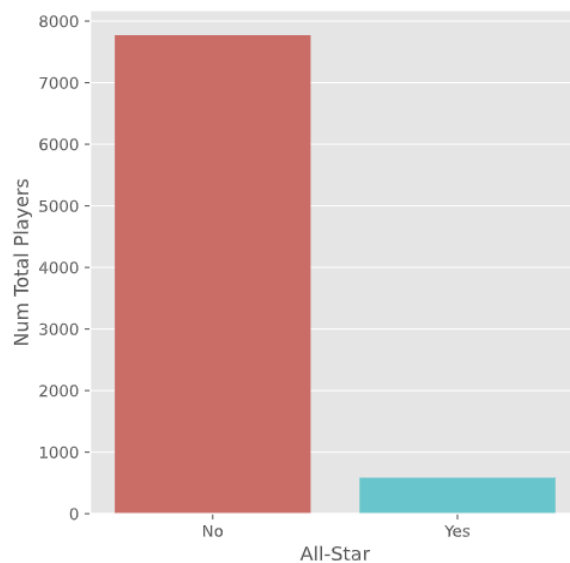


Figura 6.1. – Relación de entradas positivas y negativas del dataset (Fuente: Elaboración propia)

El dataset de entrenamiento posee 10081 filas y 22 columnas, compuesto por las estadísticas anteriormente mencionadas desde la temporada 1996/1997 hasta la 2019/2020.

| | Year | Avg. Pace | PLAYER | TEAM | Team Conference Rank | GP | Team GP | W | PTS | REB | ... | BLK | TOV | TS% | 3PM | DEFMS | USG% | PIE | Prior ASG Appearances | AS Last Year? | Selected? |
|-------|------|-----------|---------------------|------|----------------------|-----|---------|-----|------|------|-----|-----|-----|------|-----|-------|------|------|-----------------------|---------------|-----------|
| 0 | 1996 | 90.1 | Michael Jordan | CHI | 1 | 40 | 40 | 35 | 30.9 | 5.8 | ... | 0.5 | 1.7 | 56.4 | 1.2 | 0.004 | 33.6 | 19.8 | 10 | 1 | 1 |
| 1 | 1996 | 90.1 | Shaquille O'Neal | LAL | 3 | 40 | 41 | 28 | 26.2 | 13.2 | ... | 3.1 | 3.1 | 55.7 | 0.0 | 0.003 | 30.4 | 18.4 | 4 | 1 | 1 |
| 2 | 1996 | 90.1 | Latrell Sprewell | GSW | 7 | 39 | 39 | 16 | 25.9 | 4.9 | ... | 0.8 | 4.0 | 57.1 | 2.2 | 0.001 | 28.2 | 14.5 | 2 | 0 | 1 |
| 3 | 1996 | 90.1 | Karl Malone | UTA | 4 | 40 | 40 | 27 | 25.8 | 10.8 | ... | 0.7 | 3.1 | 57.6 | 0.0 | 0.003 | 31.5 | 20.8 | 9 | 1 | 1 |
| 4 | 1996 | 90.1 | Hakeem Olajuwon | HOU | 1 | 37 | 41 | 28 | 24.1 | 9.4 | ... | 2.2 | 3.7 | 54.8 | 0.1 | 0.003 | 32.2 | 16.3 | 11 | 1 | 1 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 10076 | 2019 | 100.3 | Miye Oni | UTA | 3 | 3 | 43 | 2 | 0.0 | 0.3 | ... | 0.0 | 0.3 | 0.0 | 0.0 | 0.000 | 4.5 | 0.0 | 0 | 0 | 0 |
| 10077 | 2019 | 100.3 | Paul Watson | TOR | 3 | 2 | 43 | 0 | 0.0 | 1.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.003 | 14.3 | -4.7 | 0 | 0 | 0 |
| 10078 | 2019 | 100.3 | Stanton Kidd | UTA | 3 | 4 | 43 | 3 | 0.0 | 0.8 | ... | 0.0 | 0.5 | 0.0 | 0.0 | 0.001 | 13.9 | -7.7 | 0 | 0 | 0 |
| 10079 | 2019 | 100.3 | Talen Horton-Tucker | LAL | 1 | 2 | 43 | 1 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.000 | 10.0 | 14.8 | 0 | 0 | 0 |
| 10080 | 2019 | 100.3 | Zach Norvell Jr. | GSW | 15 | 2 | 45 | 2 | 0.0 | 0.5 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.000 | 7.7 | 0.0 | 0 | 0 | 0 |

10081 rows x 22 columns

Figura 6.2 – Conjunto de datos de entrenamiento (Fuente: Elaboración propia)

El dataset que debemos predecir posee 492 filas y 21 columnas, compuesto por los datos de la última temporada de la NBA.

| | Year | Avg. Pace | PLAYER | TEAM | Team Rank | Conference | GP | Team GP | W | PTS | REB | ... | STL | BLK | TOV | TS% | 3PM | DEFWS | USG% | PIE | Prior ASG Appearances | AS Last Year? | |
|-----------------------|------|-----------|-------------------|-------------------|-----------|------------|-----|---------|-----|-----|-------|-------|-----|------|------|------|-------|-------|------|-------|-----------------------|---------------|-----|
| | 0 | 2020 | 99.20 | Bogdan Bogdanovic | ATL | | 11 | 11 | 36 | 6 | 9.40 | 3.50 | ... | 0.40 | 0.20 | 1.20 | 53.70 | 2.10 | 0.09 | 17.70 | 8.20 | 0 | 0 |
| | 1 | 2020 | 99.20 | Brandon Goodwin | ATL | | 11 | 24 | 36 | 10 | 4.30 | 1.10 | ... | 0.40 | 0.00 | 0.50 | 43.80 | 0.60 | 0.06 | 18.70 | 6.30 | 0 | 0 |
| | 2 | 2020 | 99.20 | Bruno Fernando | ATL | | 11 | 18 | 36 | 10 | 1.70 | 3.30 | ... | 0.10 | 0.10 | 0.80 | 49.10 | 0.00 | 0.04 | 12.50 | 7.60 | 0 | 0 |
| | 3 | 2020 | 99.20 | Cam Reddish | ATL | | 11 | 26 | 36 | 10 | 11.20 | 4.00 | ... | 1.30 | 0.30 | 1.30 | 48.80 | 1.30 | 0.05 | 18.30 | 5.30 | 0 | 0 |
| | 4 | 2020 | 99.20 | Clint Capela | ATL | | 11 | 32 | 36 | 13 | 14.70 | 14.20 | ... | 0.80 | 2.20 | 1.40 | 59.30 | 0.00 | 0.12 | 19.30 | 16.30 | 0 | 0 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 487 | 2020 | 99.20 | Robin Lopez | WAS | | | 12 | 34 | 34 | 14 | 8.30 | 4.30 | ... | 0.30 | 0.60 | 1.10 | 62.90 | 0.10 | 0.07 | 15.20 | 8.60 | 0 | 0 |
| 488 | 2020 | 99.20 | Rui Hachimura | WAS | | | 12 | 27 | 34 | 14 | 12.50 | 5.40 | ... | 0.70 | 0.10 | 1.00 | 54.60 | 0.80 | 0.04 | 17.10 | 7.90 | 0 | 0 |
| 489 | 2020 | 99.20 | Russell Westbrook | WAS | | | 12 | 27 | 34 | 10 | 20.30 | 9.70 | ... | 1.10 | 0.30 | 4.80 | 48.40 | 1.10 | 0.09 | 30.10 | 13.70 | 9 | 1 |
| 490 | 2020 | 99.20 | Thomas Bryant | WAS | | | 12 | 10 | 34 | 2 | 14.30 | 6.10 | ... | 0.40 | 0.80 | 1.10 | 70.40 | 0.90 | 0.04 | 16.30 | 11.10 | 0 | 0 |
| 491 | 2020 | 99.20 | Troy Brown Jr. | WAS | | | 12 | 17 | 34 | 5 | 4.10 | 3.00 | ... | 0.10 | 0.10 | 0.70 | 45.90 | 0.60 | 0.04 | 15.90 | 5.50 | 0 | 0 |
| 492 rows x 21 columns | | | | | | | | | | | | | | | | | | | | | | | |

Figura 6.3. – Conjunto de datos a clasificar (Fuente: Elaboración propia)

6.3. PREPROCESAMIENTO DE LOS DATOS

Una vez obtenido el dataset, el siguiente paso es seleccionar, limpiar y transformar los datos, para conseguir que estos sean útiles y puedan ser empleados para entrenar un modelo. De todas las métricas que posee un dataset, lo más seguro es que haya muchas de ellas que no sean necesarias, ya sea por que aporten poco o ningún valor al modelo, o bien porque se trate de valores duplicados. Es necesario identificar los picos de datos, los datos ausentes y los valores anómalos y eliminarlos de los datos. Una vez tengamos los datos “limpios”, debemos seleccionar aquellos que sean de valor para nuestro modelo predictivo y descartar aquellos que aporten ruido. Este trabajo se suele llamar Feature Engineering. Dentro de esta fase se llevará a cabo la transformación de los datos, a través de la cual se generan nuevos campos basados en los que ya se tienen, y se seleccionarán aquellos campos que consideremos predictores.

Esta parte es la que requiere de un mayor esfuerzo y creatividad de todo el proceso de creación de un modelo predictivo. Además, es de gran valor poseer un amplio conocimiento no solo del campo de estudio sobre el que se está realizando el análisis predictivo, porque nos ayudará a transformar y descartar los datos además de decidir qué campos de datos son los más valiosos para nuestro modelo,

sino también del funcionamiento de los algoritmos, ya que eso nos permitirá saber hacia dónde debemos encaminarnos a la hora de seleccionar uno u otro.

Lo primero que se ha llevado a cabo para tratar el dataset de los NBA All-Stars ha sido eliminar los outliers que poseía. Han sido eliminados aquellos jugadores que fueron All-Star pero que no debieron ser seleccionados [21]. Las elecciones de los titulares para el All-Star Game se realizan mediante las votaciones de los fans, los periodistas y los propios jugadores, mientras que los suplentes son seleccionados por los entrenadores de la liga. Antiguamente el peso del voto de los aficionados era aún mayor que en la actualidad, por lo que podía darse el caso de que participaran jugadores cuyo rendimiento a lo largo de la temporada hubiera sido pésimo o que hubiesen jugado muy pocos partidos a consecuencia de las lesiones.

Los casos que se han considerado como outliers debido a que fueron seleccionados gracias al voto de los fans pero que cuyas temporadas no debieron ser premiadas son:

- Las dos últimas temporadas de la carrera de Allen Iverson, cuyas elecciones fueron muy controvertidas.
- Las temporadas 96/97 y 97/98 de Penny Hardaway, quien estuvo lastrado por las lesiones en ambas temporadas.
- La elección de Grant Hill en el año 2000/01, quien habiendo jugado únicamente 4 partidos (10% del total) fue seleccionado, ya que era uno de los favoritos del público quienes le consideraban el próximo Michael Jordan.
- La elección de Yao Ming en su última temporada en la NBA, que habiendo jugado solo 5 partidos (11% del total) fue votado en masa por la comunidad china aun estando lesionado para este partido.
- En la temporada 2013/14 Kobe Bryant jugó 6 partidos antes al All-Star Game y llegó lesionado al ASG, pero aun así fue votado por los fans y seleccionado como titular.
- La temporada de retiro de Dirk Nowitzki, al que se le incluyó a modo de homenaje.

El siguiente paso que se ha realizado ha sido la transformación de los datos. Se ha creado la variable *GP%* a partir de los partidos jugados por el jugador y su equipo, y que nos indicará el porcentaje de partidos en los que ha participado el jugador con su equipo. Además, se han eliminado a todos aquellos jugadores hayan jugado menos del 33% de los partidos de su equipo, ya que se considera que no ha participado en suficientes partidos como para poder evaluar su rendimiento. Como se observa en la figura 6.4, en el dataset de entrenamiento muy pocos jugadores han sido seleccionados habiendo jugado menos de la mitad de los partidos.

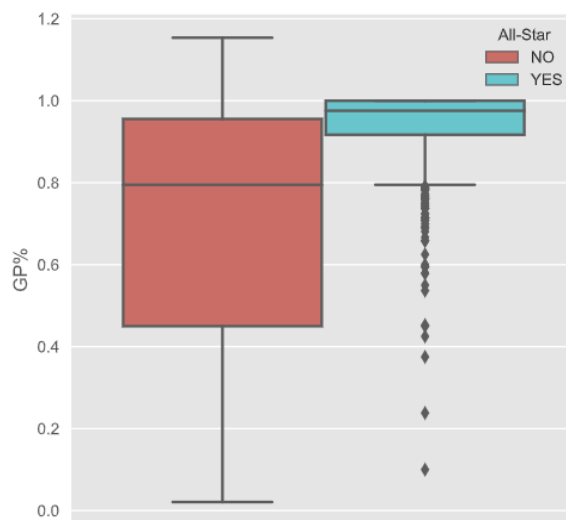


Figura 6.4. – Relación de positivos y negativos en función de su GP% (Fuente: Elaboración propia)

El baloncesto ha cambiado con el paso de los años, y no tendría sentido comparar baloncesto de épocas diferentes, por ello se han dividido las variables entre el ritmo de la liga en cada temporada. Hay que tener en cuenta que no es lo mismo promediar las mismas estadísticas ahora que se juega a 100 posesiones por partido de media, que en los años 90 y 2000 cuando se jugaban unas 90 posesiones de media por partido.

Respecto a las estadísticas que presenta el dataset, hay algunas de ellas que aportan valor al clasificador y otras que se ha decidido descartar. Una de las variables que se han decidido descartar han sido los 3PM, que como podemos ver en la figura 6.5, es una estadística a la que el clasificador no le da demasiada importancia a la hora de su creación. La otra variable que se ha decidido descartar

han sido las pérdidas de balón. Esta si es una variable que en la creación del modelo adquiere cierto peso, concretamente un 6%, pero que ha sido descartada por las razones que se explican más adelante.

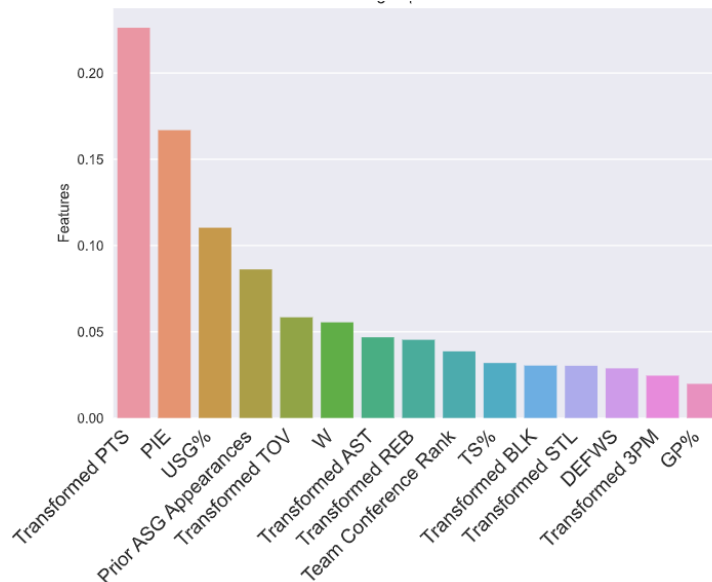


Figura 6.5. – Pesos de las estadísticas en la creación del modelo (Fuente: Elaboración propia)

Para empezar, está claro que la posición que ocupa el equipo en la clasificación influye positivamente. No solamente porque cuanto más arriba se encuentra el equipo mejor juega y mejores números poseen sus jugadores, sino también porque mayor exposición mediática tendrán sus jugadores y más oportunidades de ser seleccionados, ya que de los mejores equipos es más factible que sea seleccionado más de un único jugador para el ASG. La figura 6.6 nos muestra como la mayoría de los jugadores seleccionados se encuentran en equipos clasificados para los playoffs, es decir entre los ocho primeros. Las victorias es otra estadística que en realidad va ligada a la posición del equipo en la clasificación, pero que ayuda a diferenciar a aquellos jugadores que pertenecen a un buen equipo y no han sido completamente partícipes de dicho récord.

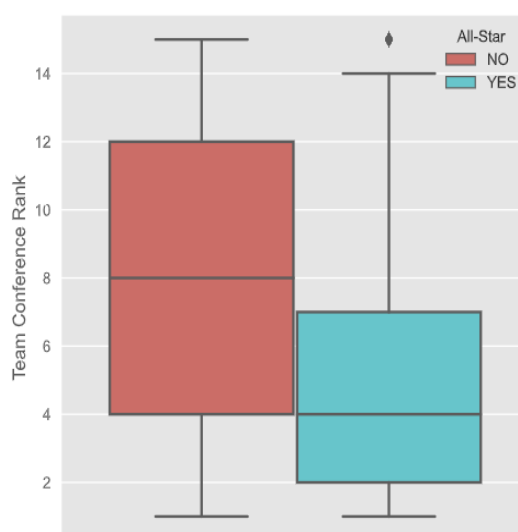


Figura 6.6. - Relación de positivos y negativos en función del ranking de su equipo (Fuente: Elaboración propia)

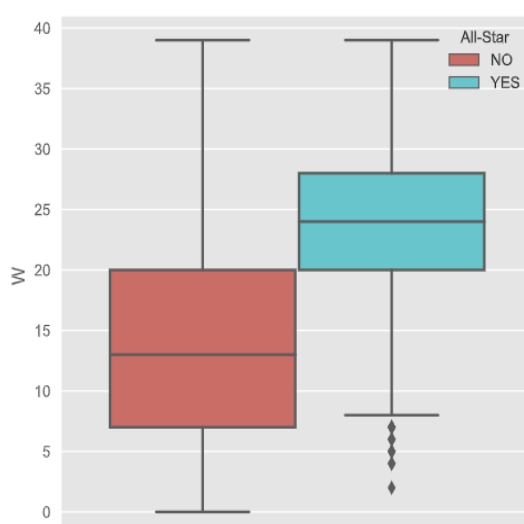


Figura 6.7. - Relación de positivos y negativos en función del número de victorias (Fuente: Elaboración propia)

Los números que promedian los jugadores en estadísticas básicas y tradicionales como son los puntos, rebotes, asistencias, robos y bloqueos, aunque esta última en menor medida ya que es más propia de jugadores grandes y/o defensivos, son las estadísticas principales a través de las cuales podemos valorar el rendimiento de los jugadores y como se puede apreciar claramente en las siguientes figuras, en la gran mayoría de los casos estas estadísticas poseen valores mayores en los jugadores seleccionados que en los no seleccionados.

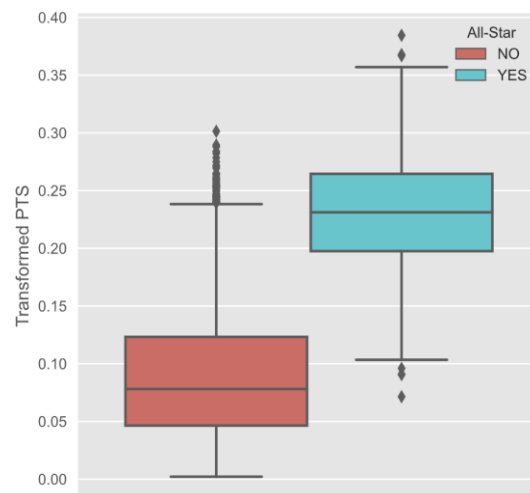


Figura 6.8. - Relación de positivos y negativos en función de sus puntos (Fuente: Elaboración propia)

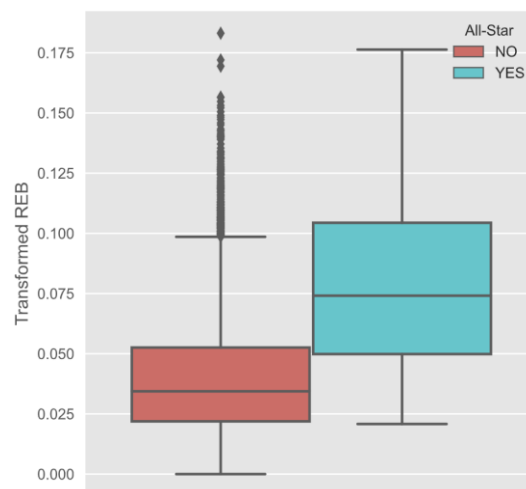


Figura 6.9. - Relación de positivos y negativos en función de sus rebotes (Fuente: Elaboración propia)

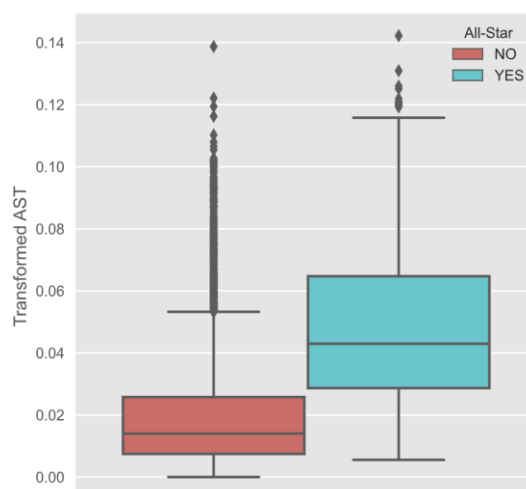


Figura 6.10. - Relación de positivos y negativos en función de sus asistencias (Fuente: Elaboración propia)

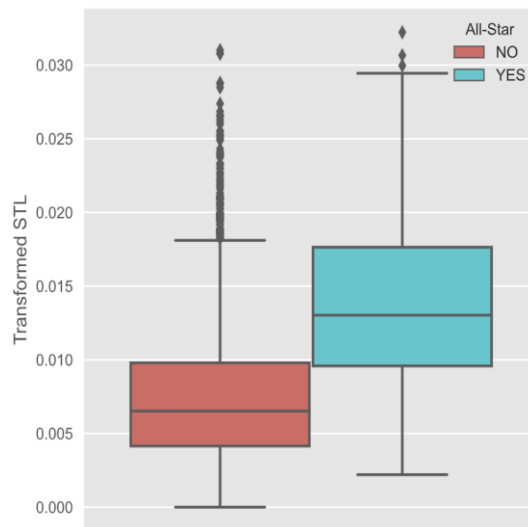


Figura 6.11. - Relación de positivos y negativos en función de sus robos (Fuente: Elaboración propia)

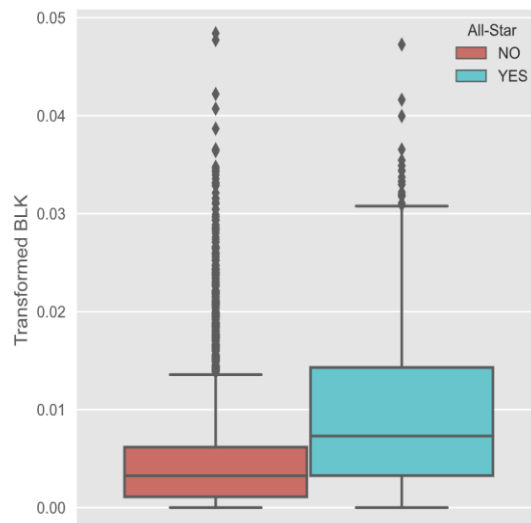


Figura 6.12. - Relación de positivos y negativos en función de sus bloqueos (Fuente: Elaboración propia)

Además, se ha decidido descartar algunas variables por considerar que no iban a aportar valor al modelo. Se ha eliminado la variable que nos decía si el jugador había sido seleccionado el año anterior, pues todos los años hay lesiones que impiden estar dos años seguidos o empiezan a despuntar nuevos jugadores en la liga que no lo habían hecho antes. Sin ir más lejos, en el All-Star Game del año pasado, jugadores como Stephen Curry, Kevin Durant o Klay Thompson no pudieron participar debido a lesiones de larga duración que no les permitieron jugar en toda la temporada y otros grandes jugadores como Luka Dončić, Donovan Mitchell o Jayson Tatum consiguieron su primera aparición.

El número de pérdidas por partido (TOV) que tiene un jugador también se ha descartado porque, aunque podamos pensar que los grandes jugadores son los que menos balones pierden, no es ni mucho menos siempre así, ya que las estrellas de la liga también son las que más acaparan el balón y más se arriesgan y por tanto tienen más posibilidades de perderlo. A pesar de que en la figura 6.13 parece existir una clara diferencia respecto a esta variable entre las entradas clasificadas como positivas y negativas, es una estadística un tanto ambigua a la hora de clasificar, ya que al final los jugadores que más pérdidas de balón cometen son los mejores, pero también los peores.

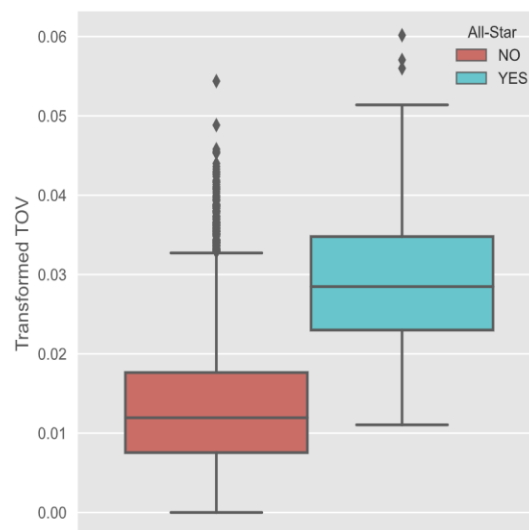


Figura 6.13. - Relación de positivos y negativos en función de sus pérdidas (Fuente: Elaboración propia)

La última característica que ha sido descartada ha sido los triples(3PM), y ha sido debido a varias razones. Como se ha comentado con anterioridad el baloncesto ha cambiado drásticamente en los últimos años. La cantidad de tiros de tres que se anotan en la actualidad es muy superior a los que se anotaban hace tan solo 10 años. Además, la anotación de triples es una faceta muy específica del juego, y que no posee una gran relación con el hecho de ser un jugador All-Star o no serlo. Existen jugadores especialistas en esta parte del juego, que son grandísimos anotadores de tres pero que tienen carencias en el resto de los aspectos del juego, además de los jugadores interiores que, si bien cada vez más están desarrollando su tiro de larga distancia, no ha sido tradicionalmente un modelo de jugador que asumiera este tipo de tiros. Podemos ver un claro ejemplo en Shaquille O'Neal, quien fue 15 veces All-Star y que únicamente anotó un triple

a lo largo de su carrera. Como podemos ver en la figura 6.14, esta variable no arroja una diferencia entre los jugadores All-Star y los no All-Star.

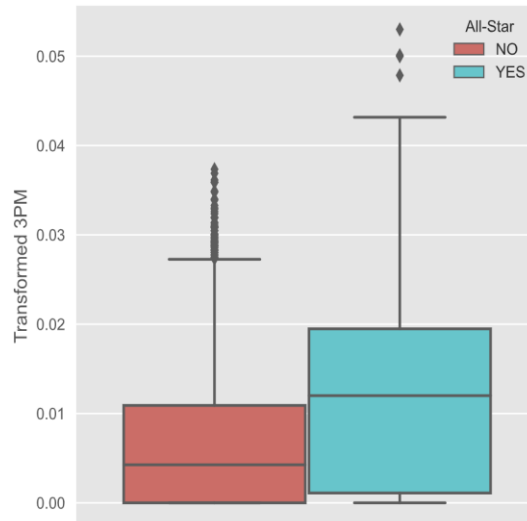


Figura 6.14. - Relación de positivos y negativos en función de sus triples (Fuente: Elaboración propia)

Estadísticas avanzadas como el PIE o el USG%, que miden el impacto que tiene un jugador en el partido y en su equipo, son variables mucho más altas en las estrellas de la liga y que por tanto muestran una clara diferencia entre los jugadores All-Star y los no All-Star como podemos ver en las figuras 6.15 y 6.16. En la figura 6.17 podemos observar cómo el TS%, que es otra estadística avanzada que nos ofrece una lectura sobre la eficiencia en los tiros de cada jugador, no proporciona una clara diferencia entre los jugadores seleccionados y los no seleccionados.

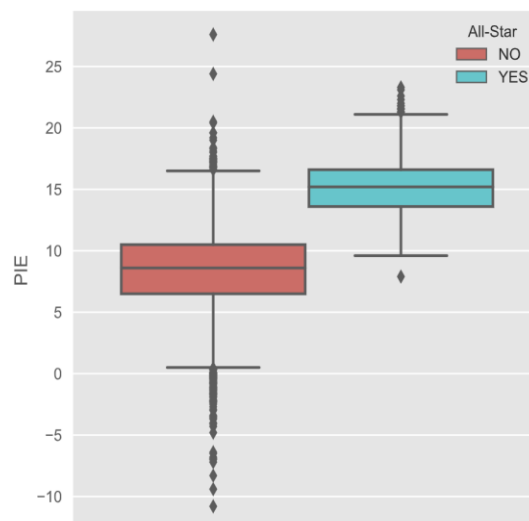


Figura 6.15. - Relación de positivos y negativos en función de su PIE (Fuente: Elaboración propia)

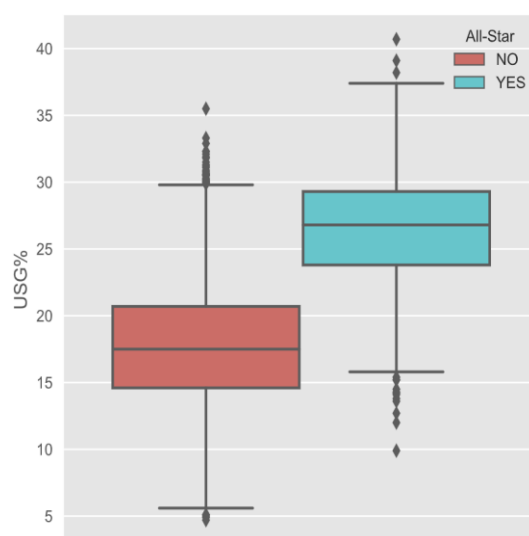


Figura 6.16. - Relación de positivos y negativos en función de su USG% (Fuente: Elaboración propia)

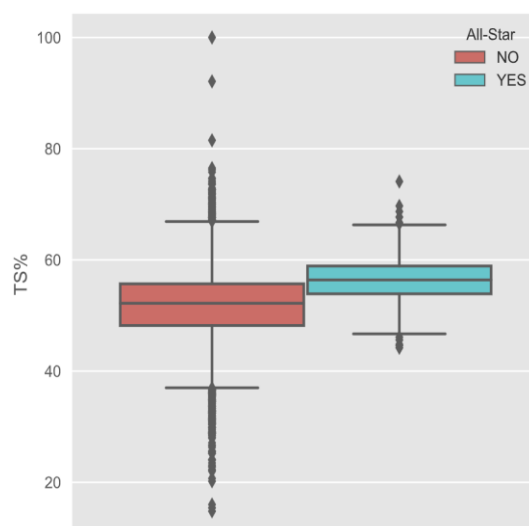


Figura 6.17. - Relación de positivos y negativos en función de su TS% (Fuente: Elaboración propia)

El DEFWS es una estadística defensiva en la cual los jugadores eminentemente anotadores y ofensivos no destacan. Ayudará a recompensar a aquellos jugadores que aporten un valor significativo en defensa dentro de sus equipos, aunque no sean baluartes ofensivos. Es una variable que aun así muestra una diferencia entre jugadores seleccionados y no seleccionados, siendo superior por lo general en los primeros.

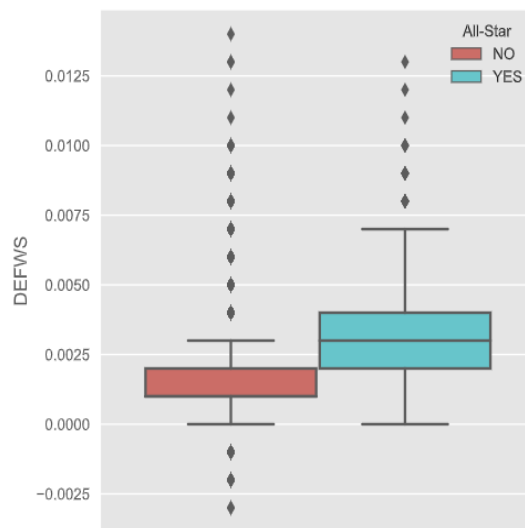


Figura 6.18. - Relación de positivos y negativos en función de su DEFWS (Fuente: Elaboración propia)

6.4. MODELOS PREDICTIVOS

El modelado predictivo utiliza métodos matemáticos que nos sirven para pronosticar resultados a futuro en función de los datos de entrada que reciben. El entrenamiento del modelo es el siguiente paso tras haber realizado una limpieza y preprocesado en los datos. Necesitaremos un dataset en el que cada una de las entradas se encuentre clasificada junto con su resultado, de manera que nos servirá para crear, entrenar y validar el modelo. Para ello se dividen los datos en 4 partes, “X_train”, “X_test”, “Y_train”, “Y_test”. Las variables independientes(X) son las características que queremos usar para predecir algún valor dado de “y”, mientras que la variable dependiente(Y) es aquella característica objetivo que estamos tratando de predecir. Los archivos “train” son los que utilizaremos para entrenar el modelo, mientras que los “test” son los que se van a usar para validar dicho modelo. Una vez validado nuestro modelo, el siguiente y último paso será aplicar el mismo a nuestro conjunto de datos aún sin clasificar para predecir cuál será el resultado.

La tecnología utilizada para la creación del modelo ha sido Scikit-Learn. La separación del conjunto de datos en train y test se ha realizado mediante la función `train_test_split()` [22] que incorpora el paquete `sklearn.model_selection` [23]. Gracias a Lazy Predict he podido comparar de una manera rápida una gran variedad de clasificadores. Lo primero es crear una instancia de `LazyClassifier` con

el parámetro “predictions” igual a “True”. Tras esto, usaremos la función *fit()* pasando como parámetro “X_train”, “X_test”, “Y_train” e “Y_test” y recibiendo como resultado un dataframe con los resultados de cada uno de los modelos.

El modelo predictivo de All-Stars es un modelo de clasificación, esto es que responde de forma binaria o con “sí” y “no”. El objetivo será clasificar a los jugadores en All-Star o no All-Star y seleccionar a los 12 con mayor probabilidad de serlo de cada una de las dos conferencias. Antes de crear el modelo se ha utilizado Lazy Predict para conocer cuáles son los modelos que mejor funcionan con el dataset de los All-Star para enfocarse en trabajar con ellos.

| Model | Accuracy | Balanced Accuracy | ROC AUC | F1 Score | Time Taken |
|-------------------------------|----------|-------------------|---------|----------|------------|
| GaussianNB | 0.92 | 0.95 | 0.95 | 0.93 | 0.03 |
| NearestCentroid | 0.90 | 0.94 | 0.94 | 0.92 | 0.02 |
| QuadraticDiscriminantAnalysis | 0.93 | 0.93 | 0.93 | 0.94 | 0.02 |
| PassiveAggressiveClassifier | 0.96 | 0.92 | 0.92 | 0.96 | 0.02 |
| Perceptron | 0.96 | 0.89 | 0.89 | 0.96 | 0.02 |
| LinearSVC | 0.97 | 0.89 | 0.89 | 0.97 | 0.05 |
| LogisticRegression | 0.97 | 0.89 | 0.89 | 0.97 | 0.05 |
| SGDClassifier | 0.97 | 0.88 | 0.88 | 0.97 | 0.04 |
| CalibratedClassifierCV | 0.97 | 0.88 | 0.88 | 0.97 | 0.20 |
| SVC | 0.97 | 0.88 | 0.88 | 0.97 | 0.15 |
| BernoulliNB | 0.88 | 0.87 | 0.87 | 0.90 | 0.03 |
| XGBClassifier | 0.97 | 0.86 | 0.86 | 0.97 | 0.30 |
| LinearDiscriminantAnalysis | 0.95 | 0.86 | 0.86 | 0.95 | 0.05 |
| LGBMClassifier | 0.96 | 0.85 | 0.85 | 0.96 | 0.27 |
| KNeighborsClassifier | 0.97 | 0.85 | 0.85 | 0.97 | 0.40 |
| AdaBoostClassifier | 0.96 | 0.85 | 0.85 | 0.96 | 0.39 |
| ExtraTreesClassifier | 0.97 | 0.84 | 0.84 | 0.97 | 0.35 |
| RandomForestClassifier | 0.97 | 0.84 | 0.84 | 0.97 | 0.63 |
| BaggingClassifier | 0.96 | 0.83 | 0.83 | 0.96 | 0.28 |
| LabelSpreading | 0.96 | 0.83 | 0.83 | 0.96 | 3.02 |
| LabelPropagation | 0.96 | 0.83 | 0.83 | 0.96 | 2.04 |
| ExtraTreeClassifier | 0.95 | 0.82 | 0.82 | 0.95 | 0.02 |
| DecisionTreeClassifier | 0.94 | 0.80 | 0.80 | 0.95 | 0.06 |
| RidgeClassifier | 0.95 | 0.73 | 0.73 | 0.95 | 0.02 |
| RidgeClassifierCV | 0.95 | 0.73 | 0.73 | 0.95 | 0.03 |
| DummyClassifier | 0.87 | 0.48 | 0.48 | 0.87 | 0.02 |

Figura 6.19. – Métricas de los algoritmos calculados por LazyPredict (Fuente: Elaboración propia)

Gracias a Scikit-Learn, que incluye multitud de algoritmos de clasificación, se han creado y evaluado varios modelos y, posteriormente se ha seleccionado el mejor de todos ellos gracias a las clases Pipeline del módulo `sklearn.pipeline` [24] y `GridSearchCV` [25] del módulo `sklearn.model_selection`. Los clasificadores utilizados son los descritos a continuación:

El Análisis Discriminante Lineal o LDA [26] es un clasificador de aprendizaje supervisado con un límite de decisión lineal y usando teorema de Bayes [27]. El modelo asume que todas las clases comparten la misma matriz de covarianza. LDA calcula la probabilidad de que una observación pertenezca a cada una de las clases existentes. La observación será asignada a aquella clase para la que la probabilidad sea mayor. El algoritmo simple de Naive Bayes consiste en una ecuación que describe la relación de probabilidades condicionales. Vincula la probabilidad de A dado B con la probabilidad de B dado A.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (6.1)$$

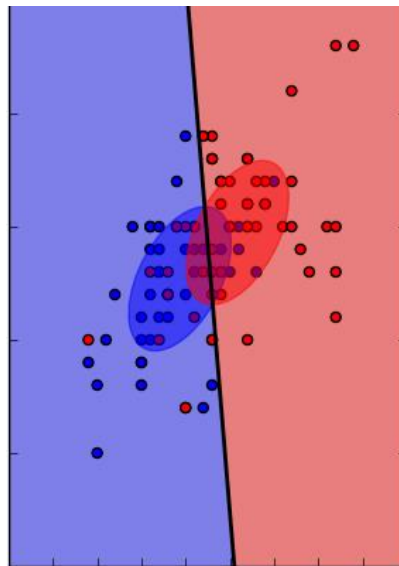


Figura 6.20. – Ejemplo de LDA (Fuente: scikit-learn.org)

El Análisis Discriminante Cuadrático o QDA [28] es igual que LDA con la diferencia de que ajusta una densidad gaussiana para cada clase, cada una tiene su propia matriz de covarianza. QDA genera límites de decisión curvos por lo que puede aplicarse a situaciones en las que la separación entre grupos no es lineal.

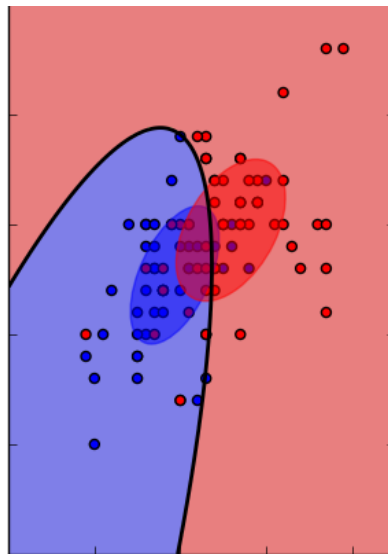


Figura 6.21. – Ejemplo de LDA (Fuente: scikit-learn.org)

El siguiente algoritmo empleado es el conocido KNN (k vecinos más cercanos) [29]. Este modelo clasifica valores buscando los puntos de datos “más similares” (por cercanía) aprendidos en la etapa de entrenamiento y haciendo conjeturas de nuevos puntos basado en esa clasificación. Es un algoritmo supervisado, es decir, que tenemos etiquetado nuestro conjunto de datos, y basado en instancia, esto quiere decir que no aprende explícitamente de un modelo, sino que memoriza las instancias de entrenamiento que son usadas como base para la fase de predicción. En la figura podemos ver un ejemplo de KNN.

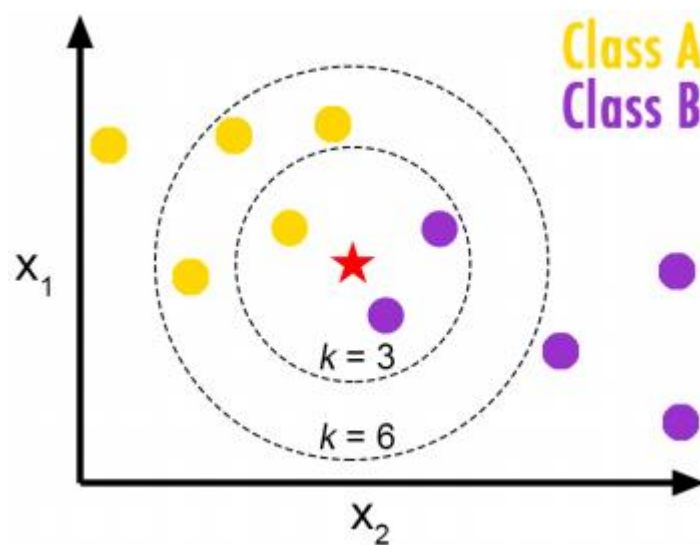


Figura 6.22. – Ejemplo de KNN (Fuente: pythondiario.com)

Un clasificador de tipo Bagging [30] es aquel que crea en paralelo múltiples clasificadores base a partir de subconjuntos de datos aleatorios del dataset original de entrenamiento y que luego agrega las predicciones individuales, ya sea por votación o por promedio, a la predicción final. Suele ser usado como forma de reducir la varianza de los clasificadores de caja negra como los árboles de decisión. Si la selección de los subconjuntos se realiza con reemplazo, entonces es conocido como Bagging, mientras que si se realiza sin reemplazo es conocido como Pasting.

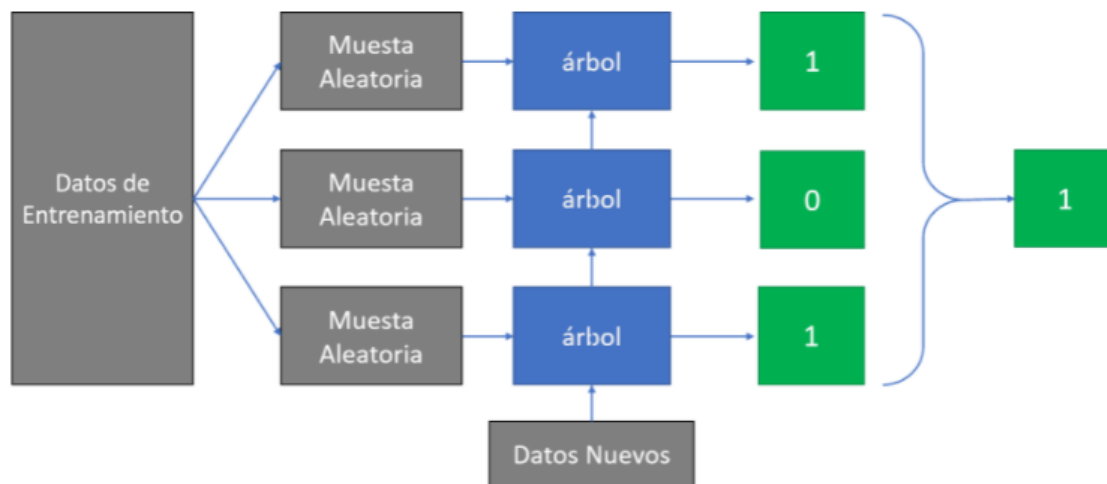


Figura 6.23. – Ejemplo de funcionamiento de un clasificador de tipo Bagging (Fuente: iartificial.net)

Un método que se ha utilizado y que está basado en Bagging es el Random Forest Classifier [31], el cual está formado por un conjunto de árboles de decisión [32]. Es un tipo de método de aprendizaje por conjuntos, donde un grupo de modelos débiles se combinan para formar un modelo poderoso. Uno de los problemas que tienen los árboles de decisiones individuales es que tienden al sobreajuste; pero en este caso, al existir multitud de árboles de decisión, cada uno creado con un conjunto de datos distinto, y utilizar el promedio de todos ellos para determinar el resultado, se acaba mitigando este problema. El funcionamiento es el siguiente: se selecciona un subconjunto de datos de manera aleatoria del dataset de entrenamiento, y se desarrolla el árbol de decisión para ese subconjunto. Este proceso se realiza varias veces. De esta manera cada árbol conoce solo una parte de los datos de entrenamiento y se entrena con ellos. Cada árbol dará una clasificación como resultado, y la que más veces se repita será la que se arroje como resultado final del algoritmo Random Forest.

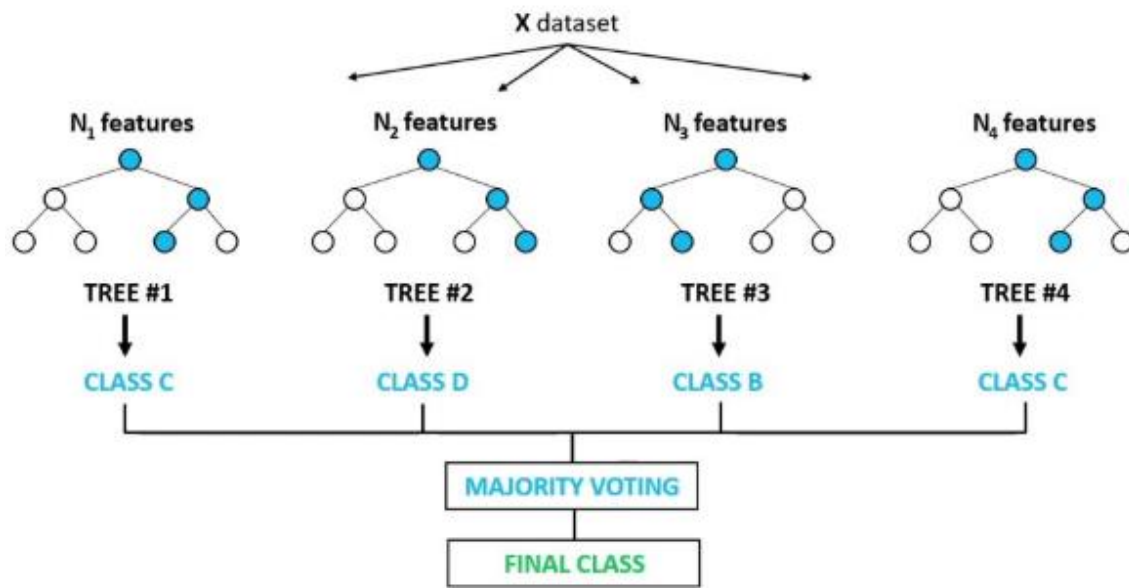


Figura 6.24. – Ejemplo de Random Forest (Fuente: rpubs.com)

Un clasificador parecido a Random Forest que también ha sido usado ha sido el Extra Trees Classifier [33], el cual utiliza un número aleatorio de árboles de decisión, cada uno creado con un subconjunto de datos distinto. Además, en lugar de elegir la mejor selección en cada nodo del árbol, elige también aleatoriamente el camino a seguir en el árbol. En este clasificador se utilizan los promedios para mejorar la precisión y controlar el sobreajuste.

El clasificador AdaBoost [34] pertenece a los clasificadores de tipo boosting. Esta familia de algoritmos, a diferencia de los bagging, se basan en la creación de un conjunto de clasificadores de baja precisión para crear un clasificador de alta precisión de una manera iterativa. Este tipo de clasificadores se ven menos afectados por problemas de sobreajuste. Concretamente en AdaBoost se comienza ajustando un clasificador con el conjunto de datos original y luego ajusta copias adicionales del clasificador con el mismo conjunto de datos, pero éstos van recibiendo pesos que dependen de los errores cometidos por cada aprendiz.

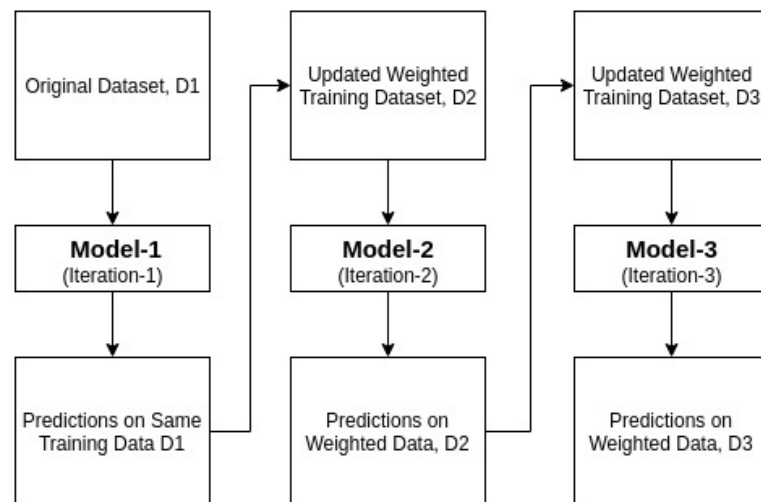


Figura 6. 25. – Ejemplo de AdaBoost (Fuente: datacamp.com)

El Gradient Boosting Classifier [35] es un clasificador, como el anterior de tipo boosting, que está formado por árboles de decisión individuales. De esta manera cada nuevo árbol creado intenta mejorar los errores cometidos en los árboles anteriores. El resultado predictivo final de este modelo se obtiene juntando las predicciones de todos los árboles individuales. La diferencia que existe con AdaBoost, es que Gradient Boosting entrena a cada clasificador de baja precisión no en los datos de entrenamiento, sino en los errores que han sido cometidos por el clasificador que le precede, es decir, en la diferencia entre los valores de la variable objetivo y las predicciones del aprendiz anterior.

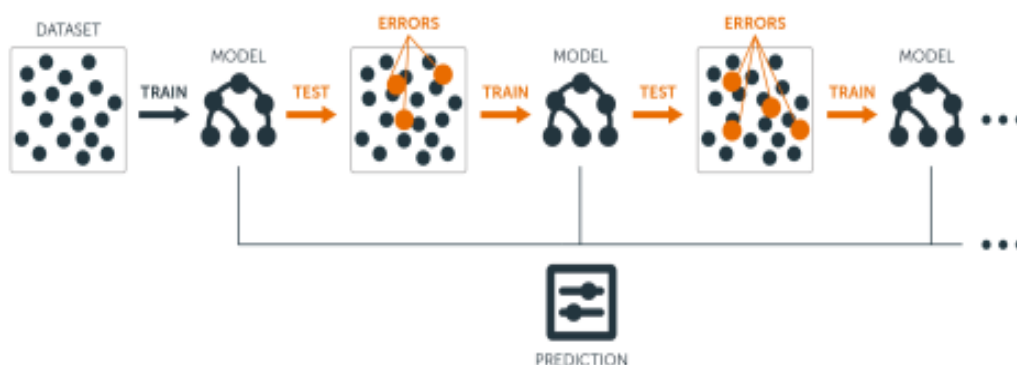


Figura 6.26. – Ejemplo de Gradient Boosting (Fuente: uc-r.github.io/)

El último clasificador que se probó fue el SVM [36], que son las máquinas de vectores de soporte, las cuales permiten encontrar la forma óptima de clasificar entre varias clases. Conocidos los datos (los puntos en la figura), el clasificador genera el hiperplano (la línea roja de la figura), que separa las etiquetas. Dependiendo de donde caiga el dato lo clasificará de una manera o de otra. La clasificación óptima se realiza maximizando el margen de separación entre las clases. Los vectores que definen el borde de esta separación son los vectores de soporte.

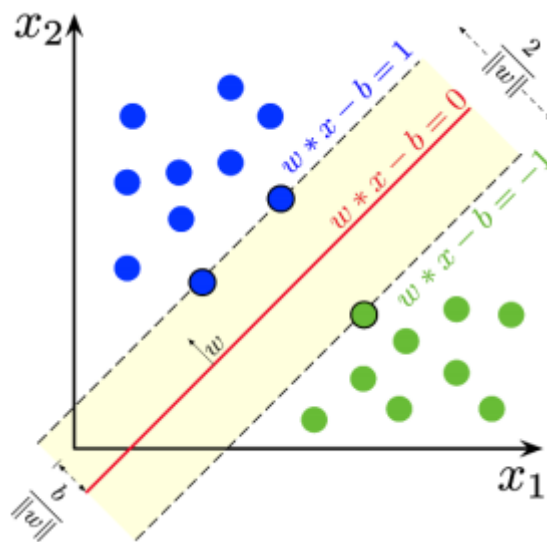


Figura 6.27. – Ejemplo de SVM (Fuente: albertotb.com)

Se ha creado un modelo de redes neuronales con MLPClassifier [37] de Scikit-Learn. MLP significa Multi-layer Perceptron, que es una red neuronal formada por múltiples capas. En la primera capa, la de entrada, cada neurona recibe un valor de entrada y no se realiza ningún procesamiento. A continuación, tenemos una o más capas ocultas, que reciben como entrada todos los valores de salida de las neuronas de la capa de entrada. Finalmente, nos encontramos con la capa de salida, formada por una o varias neuronas y cuyos valores de salida se corresponden con las salidas de toda la red.

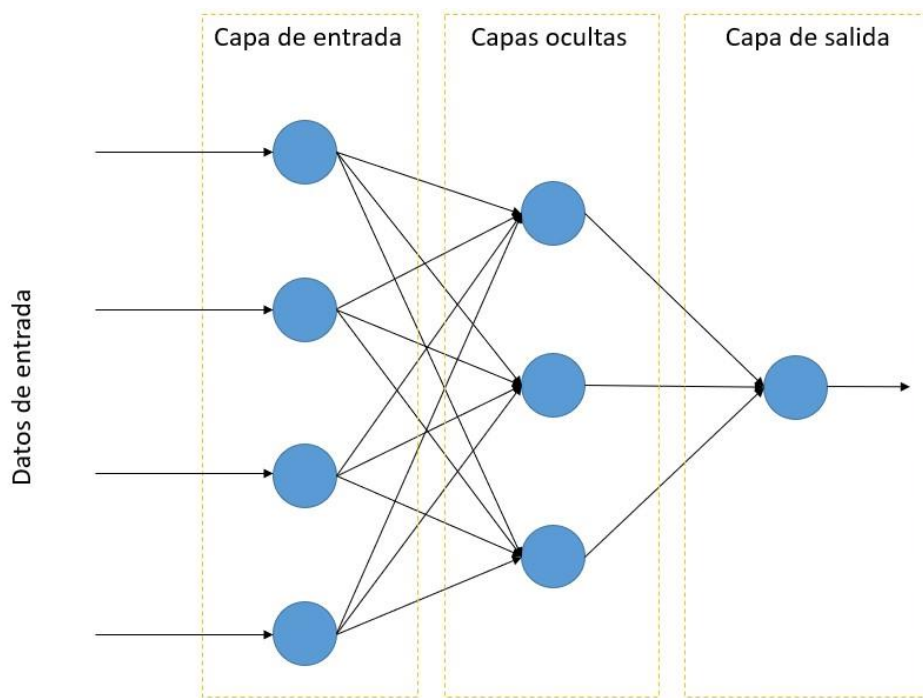


Figura 6.28. - Ejemplo de red neuronal Multi-Layer Perceptron (Fuente: interactivechaos.com)

6.5. RESULTADOS

La forma de medir el rendimiento de un modelo de clasificación con un dataset de entrenamiento, para el cual conocemos qué valores son verdaderos, es mediante una matriz de confusión [38]. En una matriz de confusión, como la que podemos ver en la figura 6.29, los cuadrantes true negative (TN) y true positive (TP) representan las observaciones que han sido predichas de manera correcta, mientras que los cuadrantes false negative (FN) y false positive (FP) computan las predicciones realizadas de manera incorrecta.

True negative hace referencia a aquellas observaciones en las que su clase real es 0 y el valor de la clase predicha es también 0. True positive son los valores en los que tanto la clase predicha como la clase real de la observación es 1.

False negative hace referencia cuando la clase real es 1 pero se ha clasificado como 0, mientras que los true negative ocurren cuando la clase real del valor es 1 pero se predice como 0.

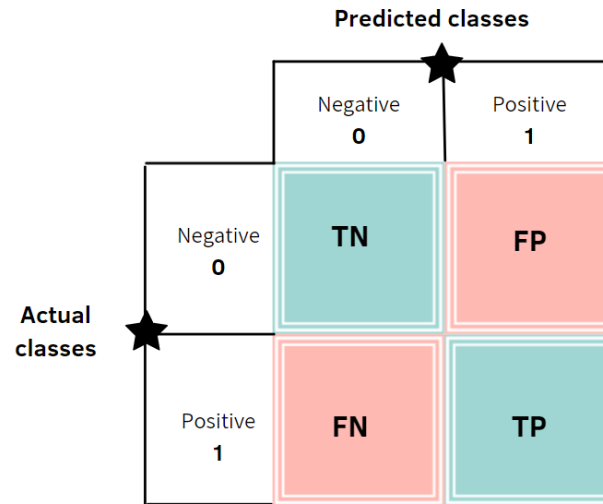


Figura 6 29. - Matriz de confusión (Fuente: towardsai.net)

Una vez conocidos los valores de la matriz de confusión, a partir de ellos podemos calcular varias métricas [39] como son accuracy, precisión, recall y F1 score.

El accuracy o exactitud [40] mide el porcentaje de casos que el modelo ha acertado sobre el total de ellos. Es una de las métricas más utilizadas, pero hay que tener en cuenta que es mucho más fiable cuando las clases están balanceadas y los valores de falsos positivos y falsos negativos son muy parecidos.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (6.2)$$

La precisión [41] mide la relación entre las entradas que han sido clasificadas correctamente como positivas y el total de entradas clasificadas como positivas. Nos dice qué porcentaje de las observaciones que han sido clasificadas positivamente realmente lo son. Esta métrica habla sobre la calidad del modelo en tareas de clasificación y es una buena medida para tener en cuenta cuando el coste de tener falsos positivos es alto, es decir, cuando no queremos una sobreestimación en el clasificador, que no haya falsas alarmas. Un ejemplo es el caso de la detección de spam, donde lo interesante es que no haya correos deseados(positivos) que sean enviados a la carpeta de spam por haber sido clasificados como negativos.

$$Precision = \frac{TP}{TP+FP} \quad (6.3)$$

El recall [42] mide la relación entre las entradas que han sido clasificadas correctamente como positivas y el total de entradas que conocemos que son positivas. Nos dice qué porcentaje de las observaciones que realmente son positivas han sido clasificadas exitosamente. Esta métrica se usa cuando existe un alto coste asociado a la existencia de falsos negativos, es decir, cuando no es recomendable que exista una subestimación en la clasificación. Un ejemplo muy claro es el que hemos sufrido con enfermedades contagiosas como la COVID-19, donde se busca que exista el menor número posible de falsos negativos, ya que clasificar a una persona infectada(positivo) como negativo puede acarrear un alto riesgo.

$$Recall = \frac{TP}{TP+FN} \quad (6.4)$$

El F1 Score [43] es la media ponderada de la precisión y el recall. Esta métrica tiene en cuenta tanto los falsos positivos como los falsos negativos y es muy útil cuando queremos que estén balanceados y buscar que ambas métricas, tanto la precisión y el recall, tengan la misma importancia. El F1 Score suele ser una buena medida en aquellos casos en los que existe una distribución de clases desigual.

$$F1\ Score = 2 * \frac{Recall * Precision}{Recall + Precision} \quad (6.5)$$

Todos los modelos explicados en el apartado 6.4 han sido creados, y aquel que mejores resultados ha dado ha sido el ExtraTreesClassifier. Las métricas correspondientes a este clasificador han obtenido resultados muy positivos. Como se puede observar en la imagen 6.30, este modelo ha obtenido prácticamente un 97% de accuracy, lo que es un muy buen resultado, pero no se puede confiar totalmente en este valor puesto que las clases que conforman el dataset están bastante desbalanceadas; hay muchos más jugadores clasificados negativamente que positivamente. Este modelo tiene una precisión del 81,2%, un recall del 70,6% y un F1 Score del 75,5%.

| Puntuación | |
|------------|----------|
| Métrica | |
| Accuracy | 0.969543 |
| Precision | 0.812500 |
| F1 Score | 0.755814 |
| Recall | 0.706522 |

Figura 6.30. – Métricas del modelo creado mediante el algoritmo ExtraTrees (Fuente: Elaboración propia)

Tanto los falsos positivos como los falsos negativos están bastante balanceados, siendo 54 y 30 respectivamente, como podemos ver en la tabla 6.1, por lo que quizás de estas métricas la que más información puede aportarnos es el F1 Score, ya que además se considera que el coste de la existencia de falsos positivos como de falsos negativos es el mismo. Los falsos negativos pueden verse engordados por jugadores que estén clasificados como positivos sin tener estadísticas para ello pero que fueran votados por el público o jugasen en un equipo con muy buen récord.

| | Predicted = 0 | Predicted = 1 |
|----------|---------------|---------------|
| Real = 0 | 2544 | 30 |
| Real = 1 | 54 | 130 |

Tabla 6.1. – Matriz de confusión del modelo (Fuente: Elaboración propia)

De los jugadores que fueron seleccionados para el All-Star Game 2021, este modelo ha predicho correctamente 23 de los 24 jugadores que fueron seleccionados. Este modelo le otorga una gran importancia (más de un 20% del total) a los puntos anotados por los jugadores. No es de extrañar que el PIE y el USG% sean las dos siguientes características a las que más valor otorga el modelo, pues nos ayudan a reconocer a los jugadores que más destacan y marcan la diferencia en los partidos. El historial de elecciones en el ASG de cada uno de los jugadores es otro de los valores más importantes. A falta de otra variable que mida la repercusión mediática, esta otorga valor extra a los jugadores que más veces han sido seleccionados con anterioridad. El porcentaje de partidos jugados en cambio tiene un valor ínfimo, apenas un 2%, debido probablemente a que a pesar de que el GP% sea bajo, si el jugador es una estrella de la liga lo más seguro sea

que sea seleccionado igualmente si el resto de las estadísticas acompañan. Con el porcentaje de tiro verdadero ocurre lo mismo. Los jugadores cuyos tiros son mayoritariamente interiores, es decir, cercanos a la canasta, poseen unos porcentajes superiores al resto; en cambio, la mayoría de los mejores jugadores de la liga asumen un alto porcentaje de tiros exteriores, bajando así sus prestaciones en este apartado. Estadísticas de carácter defensivo como el DEFWS o los bloqueos realizados también tienen una importancia inferior al 5%, esto se puede explicar porque los jugadores de corte defensivo no suelen ser tan susceptibles de ser seleccionados como sí lo son los jugadores ofensivos.

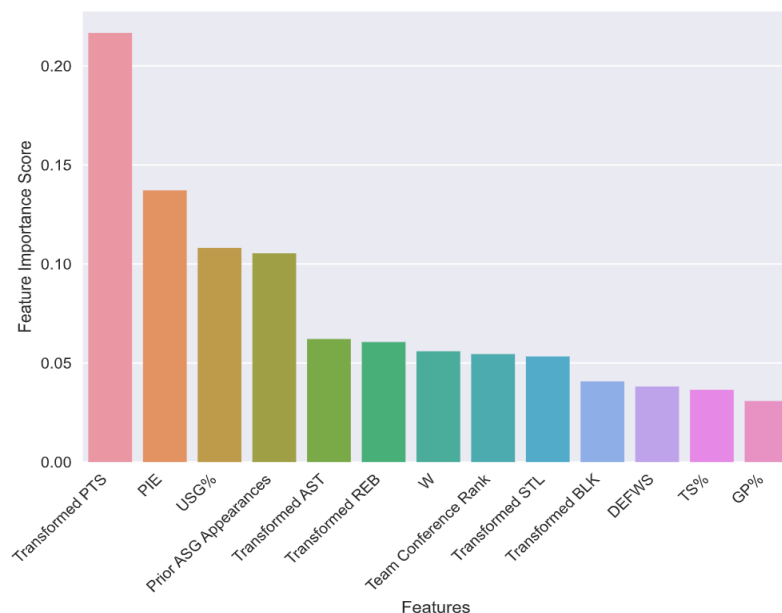


Figura 6.31. – Pesos de las variables en la creación del modelo (Fuente: Elaboración propia)

De los 12 jugadores con mayor probabilidad de ser seleccionados en el este según el clasificador, solamente Trae Young no lo fue All-Star. Como vemos en la figura 6.32, el clasificador le otorga un 69% de posibilidades de ser elegido, mientras que el jugador que realmente participó en este partido fue Ben Simmons, a quien el modelo le coloca en decimocuarta posición con un 57% de posibilidades de ser seleccionado. Ben Simmons es un jugador que destaca en el apartado defensivo y que es una de las estrellas del mejor equipo de la conferencia este, pero sin embargo es poco anotador, nulo de hecho en el tiro exterior, y con un uso del balón muy pobre para ser un jugador All-Star (en la media del resto de jugadores de su equipo). En conclusión, destaca en los apartados a los que el clasificador le

otorga menor importancia y es un jugador promedio en el resto, es por ello por lo que se le otorga la etiqueta de All-Star, pero con tan solo un 57% de posibilidades y no estando entre los 12 primeros.

| | PLAYER | TEAM | PROBABILITY |
|----|-----------------------|------|-------------|
| 1 | Giannis Antetokounmpo | MIL | 0.98 |
| 2 | Joel Embiid | PHI | 0.96 |
| 3 | Bradley Beal | WAS | 0.88 |
| 4 | James Harden | BKN | 0.86 |
| 5 | Kyrie Irving | BKN | 0.79 |
| 6 | Julius Randle | NYK | 0.77 |
| 7 | Jayson Tatum | BOS | 0.75 |
| 8 | Nikola Vucevic | ORL | 0.75 |
| 9 | Trae Young | ATL | 0.69 |
| 10 | Kevin Durant | BKN | 0.66 |
| 11 | Zach LaVine | CHI | 0.65 |
| 12 | Jaylen Brown | BOS | 0.60 |
| 13 | Khris Middleton | MIL | 0.58 |
| 14 | Ben Simmons | PHI | 0.57 |
| 15 | Jimmy Butler | MIA | 0.54 |
| 16 | Bam Adebayo | MIA | 0.52 |

Figura 6.32. – Posibilidades de ser All-Star en la conferencia Este (Fuente: Elaboración propia)

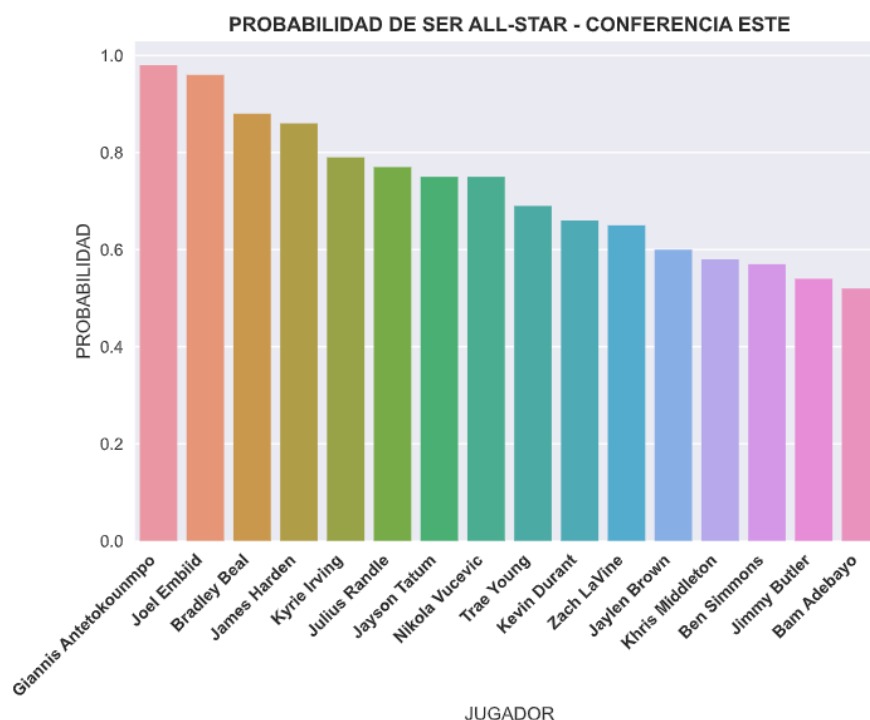


Figura 6.33. – Posibilidades de ser All-Star en la conferencia Este (Fuente: Elaboración propia)

De los jugadores clasificados como All-Stars por el modelo, todos ellos fueron seleccionados en la realidad. En esta conferencia aparecen 13 jugadores clasificados positivamente debido a que Anthony Davis, el noveno jugador de la lista y quien cuenta con un 75% de posibilidades de ser elegido, se lesionó una vez había sido seleccionado y por tanto entró otro jugador le sustituyó en el ASG; ese jugador fue Devin Booker, quien este clasificador le otorga un 72% de posibilidades y que en la realidad entraba en todas las quinielas para ser seleccionado de inicio. Antes de jugar el partido, Devin Booker se lesionaría y el encargado de suplirle sería Mike Conley.

| | PLAYER | TEAM | PROBABILITY |
|----|------------------|------|-------------|
| 1 | LeBron James | LAL | 0.99 |
| 2 | Nikola Jokic | DEN | 0.96 |
| 3 | Luka Doncic | DAL | 0.94 |
| 4 | Stephen Curry | GSW | 0.94 |
| 5 | Damian Lillard | POR | 0.94 |
| 6 | Kawhi Leonard | LAC | 0.91 |
| 7 | Paul George | LAC | 0.84 |
| 8 | Donovan Mitchell | UTA | 0.82 |
| 9 | Anthony Davis | LAL | 0.75 |
| 10 | Devin Booker | PHX | 0.72 |
| 11 | Rudy Gobert | UTA | 0.72 |
| 12 | Chris Paul | PHX | 0.63 |
| 13 | Zion Williamson | NOP | 0.57 |

Figura 6.34. - Posibilidades de ser All-Star en la conferencia Oeste (Fuente: Elaboración propia)

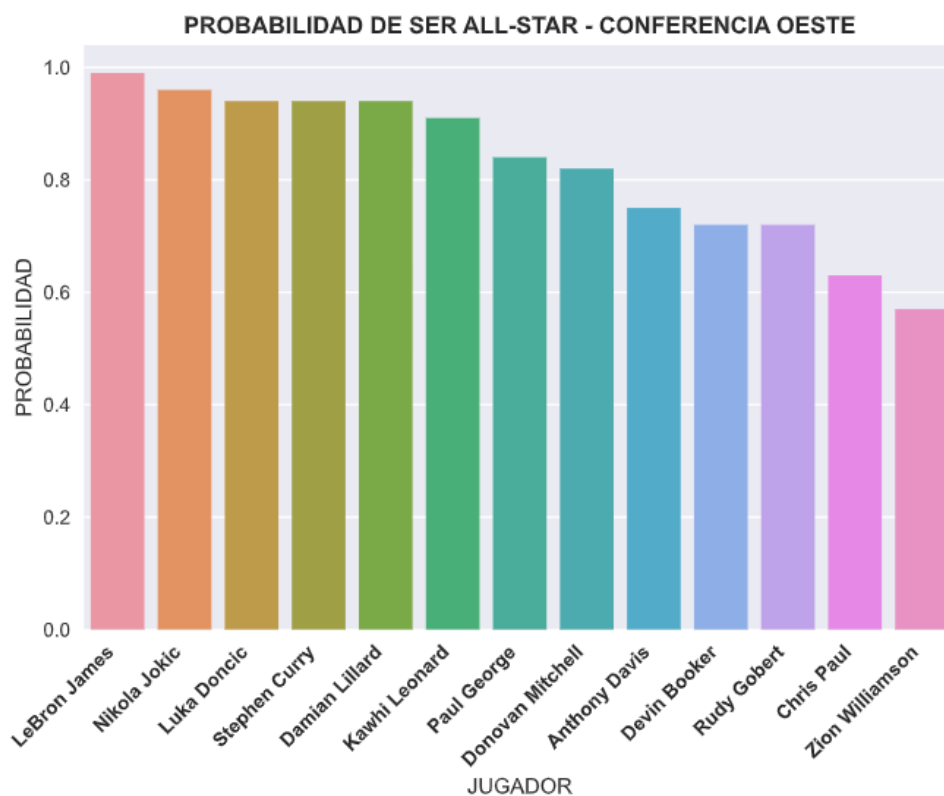


Figura 6.35. - Posibilidades de ser All-Star en la conferencia Oeste (Fuente: Elaboración propia)

El otro modelo que mejores resultados ha obtenido ha sido el RandomForestClassifier. Los resultados de las métricas son muy similares a las del modelo que utiliza ExtraTreesClassifier. Como se ve en la figura 6.36, este modelo ha obtenido un 97% de accuracy, una precisión del 81%, un recall del 70% y un F1 Score del 75%.

| Puntuación | |
|------------|------|
| Métrica | |
| Accuracy | 0.97 |
| Precision | 0.81 |
| F1 Score | 0.75 |
| Recall | 0.70 |

Figura 6.36. Métricas del modelo creado mediante el algoritmo Random Forest (Fuente: Elaboración propia)

Este modelo ha clasificado correctamente a 22 jugadores del total de 24 que fueron seleccionados para el All-Star Game. Este modelo les otorga una gran importancia a los puntos anotados por cada jugador y al PIE. También destaca, al igual que en el anterior modelo el USG% y el número de veces que el jugador ha sido elegido All-Star previamente. Al igual que en el anterior modelo el porcentaje de partidos jugados y el de tiro verdadero tienen una importancia muy baja, además de las estadísticas defensivas y la clasificación del equipo en su conferencia, todas ellas por debajo del 5% de importancia para este modelo.

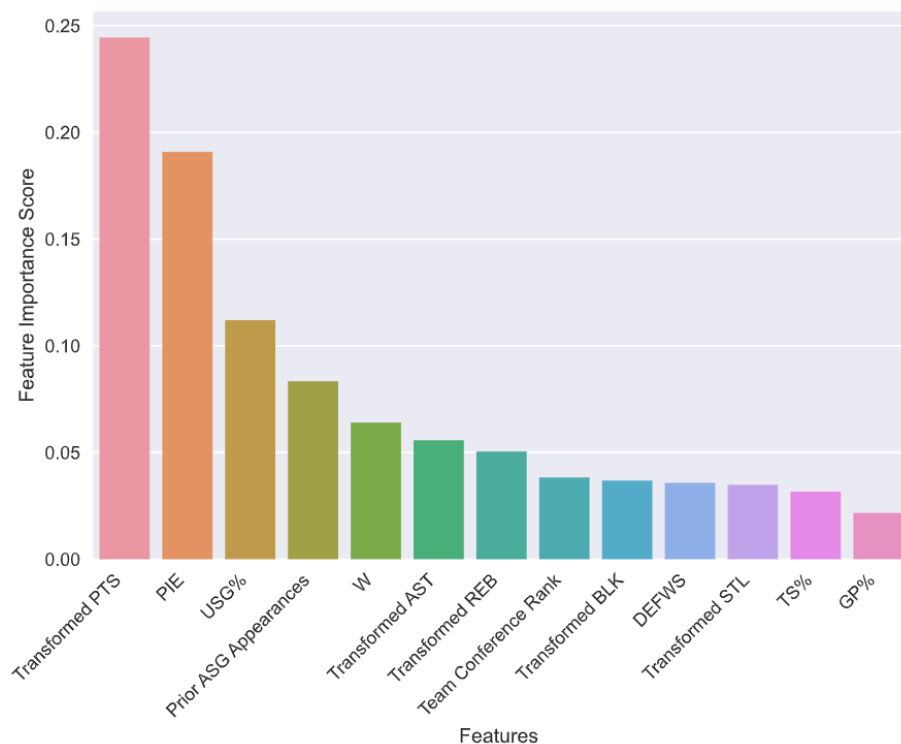


Figura 6.37. - Pesos de las variables en la creación del modelo (Fuente: Elaboración propia)

En la conferencia este, el modelo ha clasificado correctamente 10 de los 12 jugadores a los que le ha asignado un mayor porcentaje de posibilidades de ser seleccionado. Como en el anterior modelo, Trae Young está entre los mejores clasificados, con un 84% en este caso, cuando en la realidad no fue All-Star. Es un jugador que cumple con el prototipo de All-Star y que ambos clasificadores consideran que tiene que ser seleccionado. El otro jugador que ha sido clasificado erróneamente es Khris Middleton, quien ha recibido un 60% de posibilidades de ser positivo colocándose en duodécima posición, por el 52% que recibe Jaylen Brown quien sí que fue seleccionado y aparece en la posición número 15. Ben Simmons, el otro jugador que realmente fue elegido, ni siquiera aparece en la lista, ya que el clasificador le ha otorgado menos de un 50% de posibilidades. Las razones las mismas que las explicadas en el anterior clasificador, pero obteniendo menor número posibilidades en este debido a que variables en las que destacaba como el ranking de su equipo o los robos realizados tienen menos peso, mientras que otras en las que no destaca como los puntos o el PIE adquieren mucho más peso en este modelo.

Como se puede apreciar en la figura 6.38, este clasificador, a pesar de acertar un jugador menos que el anterior entre los 12 primeros, otorga unos porcentajes de selección mucho mayores a los 11 primeros jugadores que a los jugadores que vienen a continuación (se pasa de un 77% de Jayson Tatum en la posición número 11 a un 60% de Khris Middleton en la posición número 12). El modelo tiene claro que los once primeros jugadores que aparecen en la imagen 6.39 han de ser seleccionados y les otorga porcentajes más sólidos, mientras que la elección número 12 podría caer a cualquiera de los siguientes jugadores. Esto encaja con lo que ocurrió en la realidad, y es que la última posición de esta conferencia fue otorgada finalmente a Ben Simmons, pero bien podría haber sido seleccionado cualquiera de estos otros jugadores. De hecho, un jugador como Jimmy Butler, quien tiene un 57% de posibilidades, tenía posibilidades reales de ser elegido, pero insinuó públicamente que, si su compañero de equipo Bam Adebayo no era elegido también, él no estaba interesado en participar.

| | PLAYER | TEAM | PROBABILITY |
|----|-----------------------|------|-------------|
| 1 | Joel Embiid | PHI | 1.00 |
| 2 | Giannis Antetokounmpo | MIL | 0.99 |
| 3 | James Harden | BKN | 0.97 |
| 4 | Bradley Beal | WAS | 0.97 |
| 5 | Julius Randle | NYK | 0.91 |
| 6 | Nikola Vucevic | ORL | 0.87 |
| 7 | Trae Young | ATL | 0.84 |
| 8 | Zach LaVine | CHI | 0.82 |
| 9 | Kevin Durant | BKN | 0.81 |
| 10 | Kyrie Irving | BKN | 0.78 |
| 11 | Jayson Tatum | BOS | 0.77 |
| 12 | Khris Middleton | MIL | 0.60 |
| 13 | Jimmy Butler | MIA | 0.57 |
| 14 | Russell Westbrook | WAS | 0.57 |
| 15 | Jaylen Brown | BOS | 0.52 |
| 16 | Bam Adebayo | MIA | 0.51 |

Figura 6.38. - Posibilidades de ser All-Star en la conferencia Este (Fuente: Elaboración propia)

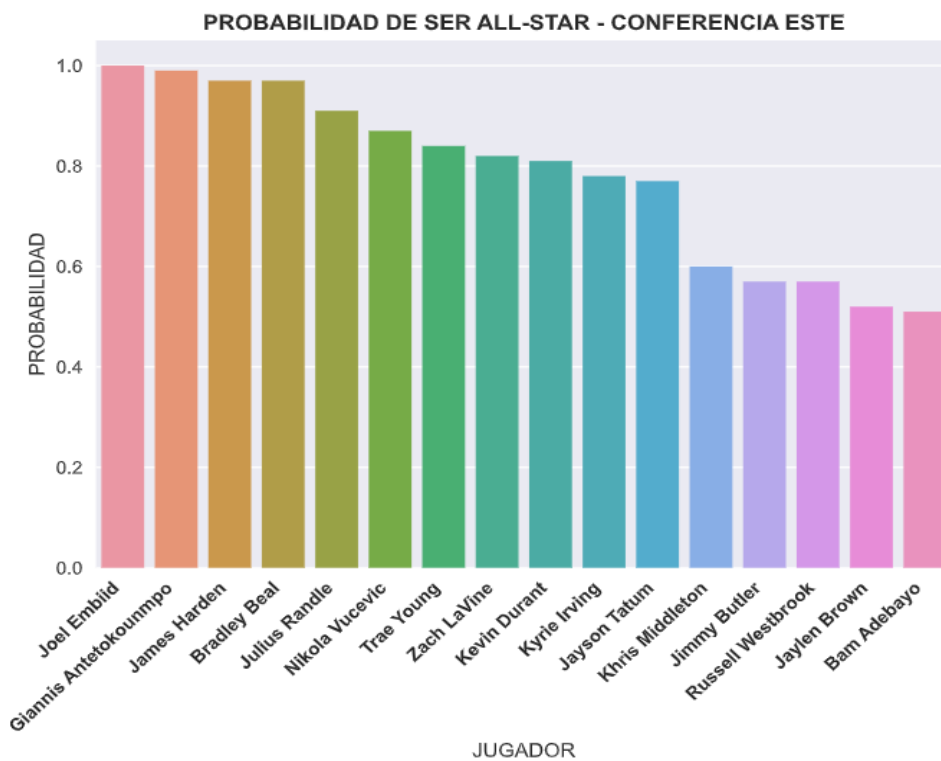


Figura 6.39. - Posibilidades de ser All-Star en la conferencia Este (Fuente: Elaboración propia)

En el caso de la conferencia oeste, de nuevo, los 13 primeros jugadores de la lista de clasificados como positivos son los mismos. De nuevo como en los clasificados en el este, este modelo otorga unos porcentajes más sólidos y elevados a la gran mayoría de los jugadores seleccionados. Como podemos observar en la figura, hasta la posición número 9 se están otorgando valores superiores al 87%, bajando al 78% en la siguiente posición. Está claro que el modelo les otorgaría claramente a estos jugadores la etiqueta de All-Star, y no se equivocaría con respecto a lo que ocurrió en la realidad. Únicamente las siguientes posiciones sufren un claro descenso de probabilidades, a pesar de que aun así los jugadores que ocupan las posiciones 11, 12 y 13 están clasificados correctamente. Estos jugadores son Rudy Gobert, Chris Paul y Zion Williamson. El primero de ellos es el mejor defensor del campeonato, por lo que, como es de esperar, destaca en la parcela defensiva y no en el ataque, y quien además juega en el mejor equipo de la liga. Como hemos visto, estas estadísticas a pesar de tener valor para el modelo son a las que menos importancia se les otorga. El siguiente es Chris Paul, un veterano de la liga, con muchas apariciones previas en el ASG, que, si bien es uno de los jugadores más importantes de su equipo, es alguien que no destaca por

tener grandes cifras en su hoja de estadísticas, que al final es en lo que se puede basar este clasificador, sino por hacer jugar bien a su equipo, su inteligencia, liderazgo y los llamados intangibles, que son variables que este clasificador no toma en cuenta. El último es Zion Williamson, el cual no posee malas estadísticas, pero a quien que este modelo no mida la repercusión mediática más allá de las veces que ha sido seleccionado con anterioridad, que en su caso es ninguna, le ha perjudicado. Zion fue elegido en el número 1 del Draft del año 2019 con muchísima claridad, y es uno de los jugadores favoritos de los fans y de la propia liga. Es un jugador muy exuberante, que posee grandes cualidades físicas y que está considerado por la gran mayoría de la gente como uno de los futuros mejores jugadores del mundo. De ahí que sus elecciones para el All-Star Game, a excepción de que sufra una lesión, comiencen a ser una cosa recurrente año tras año a poco que las estadísticas le acompañen.

| | PLAYER | TEAM | PROBABILITY |
|----|-------------------------|------|-------------|
| 1 | LeBron James | LAL | 0.99 |
| 2 | Nikola Jokic | DEN | 0.99 |
| 3 | Kawhi Leonard | LAC | 0.98 |
| 4 | Stephen Curry | GSW | 0.97 |
| 5 | Anthony Davis | LAL | 0.94 |
| 6 | Damian Lillard | POR | 0.94 |
| 7 | Donovan Mitchell | UTA | 0.92 |
| 8 | Luka Doncic | DAL | 0.91 |
| 9 | Paul George | LAC | 0.87 |
| 10 | Devin Booker | PHX | 0.78 |
| 11 | Rudy Gobert | UTA | 0.66 |
| 12 | Chris Paul | PHX | 0.64 |
| 13 | Zion Williamson | NOP | 0.63 |
| 14 | CJ McCollum | POR | 0.62 |
| 15 | Shai Gilgeous-Alexander | OKC | 0.60 |
| 16 | Karl-Anthony Towns | MIN | 0.58 |
| 17 | Christian Wood | HOU | 0.56 |

Figura 6.40. - Posibilidades de ser All-Star en la conferencia Oeste (Fuente: Elaboración propia)

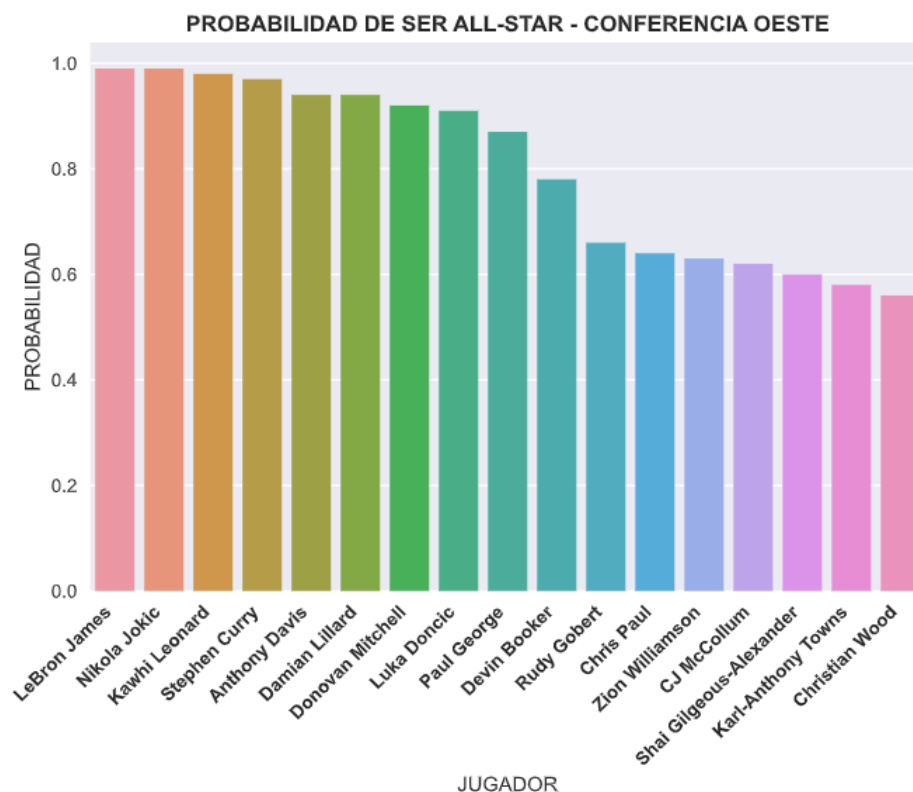


Figura 6.41. - Posibilidades de ser All-Star en la conferencia Oeste (Fuente: Elaboración propia)

El modelo de redes neuronales ha obtenido un accuracy similar, con casi un 97%. En las otras métricas, este modelo obtiene mejores resultados en el recall respecto a los anteriores, con casi un 80%, aunque baja al 75% en la precisión.

| Puntuación | |
|------------|----------|
| Métricas | |
| Accuracy | 0.969181 |
| Precision | 0.753846 |
| F1 Score | 0.775726 |
| Recall | 0.798913 |

Figura 6.42. - Métricas del modelo creado mediante redes neuronales (Fuente: Elaboración propia)

Para los All-Stars de la conferencia este, el modelo clasifica claramente a 12 jugadores por encima del resto, con más de un 85% de posibilidades. De estos jugadores, todos ellos menos Trae Young y Russell Westbrook fueron realmente seleccionados.

| | PLAYER | TEAM | PROBABILITY |
|----|-----------------------|------|-------------|
| 1 | Giannis Antetokounmpo | MIL | 0.999098 |
| 2 | Joel Embiid | PHI | 0.998619 |
| 3 | James Harden | BKN | 0.995272 |
| 4 | Kevin Durant | BKN | 0.971702 |
| 5 | Bradley Beal | WAS | 0.962670 |
| 6 | Nikola Vucevic | ORL | 0.961715 |
| 7 | Trae Young | ATL | 0.958938 |
| 8 | Kyrie Irving | BKN | 0.947594 |
| 9 | Zach LaVine | CHI | 0.929598 |
| 10 | Jayson Tatum | BOS | 0.896109 |
| 11 | Julius Randle | NYK | 0.880363 |
| 12 | Russell Westbrook | WAS | 0.850534 |
| 13 | Jimmy Butler | MIA | 0.646535 |
| 14 | Bam Adebayo | MIA | 0.630445 |
| 15 | Khris Middleton | MIL | 0.624967 |
| 16 | Jaylen Brown | BOS | 0.598467 |
| 17 | Tobias Harris | PHI | 0.578404 |
| 18 | Domantas Sabonis | IND | 0.541074 |

Figura 6.43. - Posibilidades de ser All-Star en la conferencia Este (Fuente: Elaboración propia)

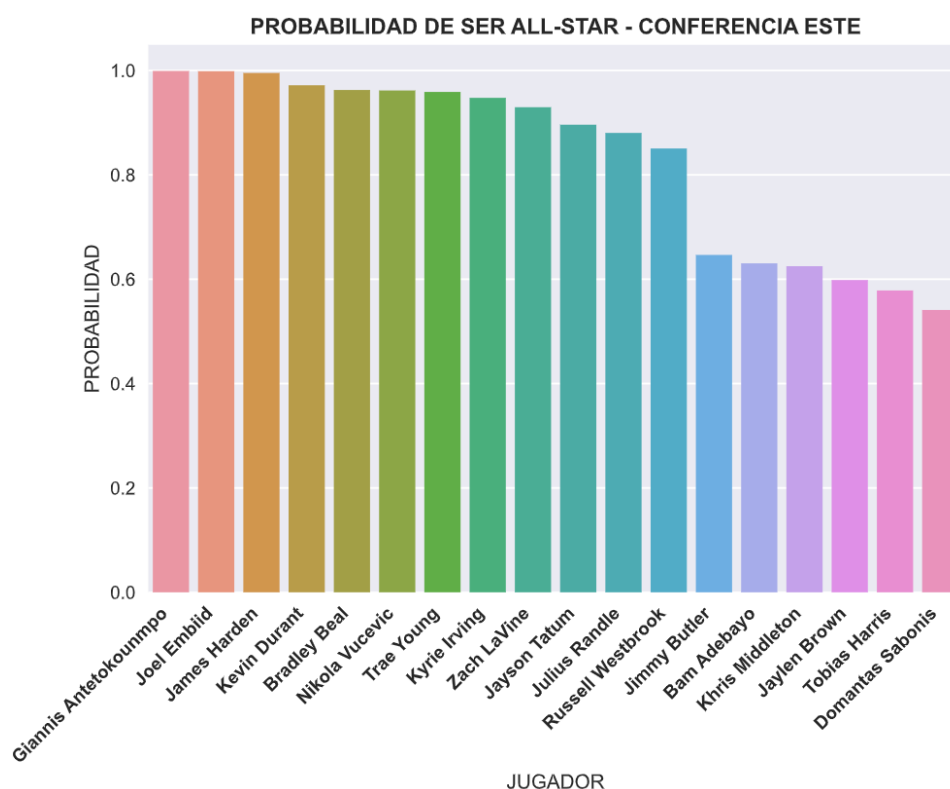


Figura 6.44. - Posibilidades de ser All-Star en la conferencia Este (Fuente: Elaboración propia)

Respecto a los jugadores seleccionados en de la conferencia oeste, el modelo otorga más de un 93% de probabilidades a los 11 primeros jugadores, quienes además fueron realmente seleccionados. Este modelo selecciona a Shai Gilgeous-Alexander, con un 81% de probabilidades en el lugar de Chris Paul y Zion Williamson, quienes fueron realmente seleccionados y obtienen un 67% y un 64,5% de posibilidades de ser seleccionados por este modelo respectivamente.

| | PLAYER | TEAM | PROBABILITY |
|----|-------------------------|------|-------------|
| 1 | Stephen Curry | GSW | 0.994011 |
| 2 | LeBron James | LAL | 0.993953 |
| 3 | Damian Lillard | POR | 0.990794 |
| 4 | Kawhi Leonard | LAC | 0.990555 |
| 5 | Nikola Jokic | DEN | 0.990305 |
| 6 | Luka Doncic | DAL | 0.977752 |
| 7 | Donovan Mitchell | UTA | 0.975492 |
| 8 | Anthony Davis | LAL | 0.961551 |
| 9 | Paul George | LAC | 0.953050 |
| 10 | Devin Booker | PHX | 0.946826 |
| 11 | Rudy Gobert | UTA | 0.930778 |
| 12 | Shai Gilgeous-Alexander | OKC | 0.813040 |
| 13 | CJ McCollum | POR | 0.766559 |
| 14 | Chris Paul | PHX | 0.671641 |
| 15 | Brandon Ingram | NOP | 0.660684 |
| 16 | Zion Williamson | NOP | 0.645123 |
| 17 | DeMar DeRozan | SAS | 0.591153 |
| 18 | Karl-Anthony Towns | MIN | 0.539876 |
| 19 | Christian Wood | HOU | 0.505147 |

Figura 6.45. - Posibilidades de ser All-Star en la conferencia Oeste (Fuente: Elaboración propia)

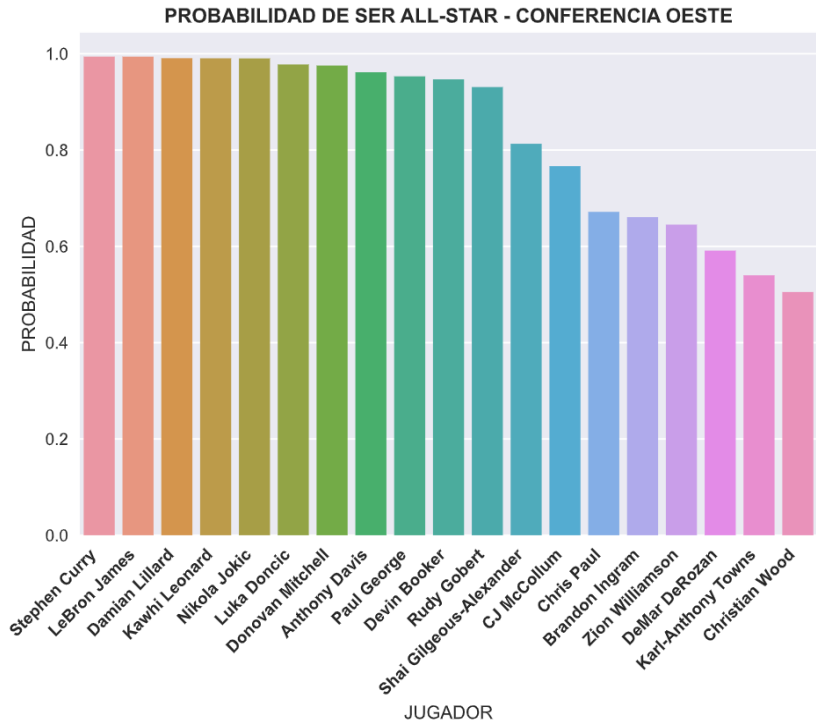


Figura 6.46. - Posibilidades de ser All-Star en la conferencia Oeste (Fuente: Elaboración propia)

7. Conclusiones y recomendaciones

Como se ha expuesto en el apartado 6.5, se ha logrado crear varios clasificadores con altos porcentajes de acierto, teniendo el mejor de ellos un éxito superior al 95%, más que cualquiera de los que existen actualmente en internet. Supone una altísima tasa de acierto, fallando únicamente un jugador del total de los seleccionados. Además, todos los objetivos que fueron propuestos al principio del proyecto han sido cumplidos con éxito.

De cara a mejorar el modelo, se podrían añadir características que midieran la repercusión social de los jugadores, ya que una parte de las selecciones pasa por la votación de los fans. También se podría tratar de crearse otros modelos que fuesen capaces de calcular los jugadores que conformarán el mejor equipo del año o los ganadores de los premios individuales de la NBA.

Lista de referencias bibliográficas

- [1] M. Lewis. *Moneyball: The Art of Winning an Unfair Game*. W W Norton & Co, 2003.
- [2] "Nba.stats" [Online]. Available: <https://www.nba.com/stats/>
- [3] "Basketball Reference" [Online]. Available: <https://www.basketball-reference.com/>
- [4] L. Winter. "Predicting NBA All-Stars" [Online]. Available: <https://betterprogramming.pub/predicting-nba-all-stars-e03655021f63>
- [5] T. Yoon. "Predicting the 2020 NBA Champion with Machine Learning" [Online]. Available: <https://towardsdatascience.com/predicting-the-2020-nba-champion-with-machine-learning-32100f6b253d>
- [6] R. Alterman. "Predicting NBA Win Percentage" [Online]. Available: <https://towardsdatascience.com/predicting-nba-win-percentage-84148ae8d3e6>
- [7] G. Malim. "Predicting the NBA All-Stars and NBA Awards with Machine Learning" [Online]. Available: https://github.com/gmalim/NBA_analysis
- [8] C. Porteous. "Using machine learning to predict NBA All-Stars" [Online]. Available: <https://towardsdatascience.com/using-machine-learning-to-predict-nba-all-stars-part-1-data-collection-9fb94d386530>
- [9] Universidad de León, "Reglamentos de Trabajos de Fin de Grado" [Online]. Available: https://ingenierias.unileon.es/wp-content/uploads/2021/02/20210125_Reglamento-TFG_EIIIA-aprobado-JE.pdf
- [10] AENOR, "UNE 50103:1990" [Online]. Available: <https://www.une.org/encuentra-tu-norma/busca-tu-norma/norma/?c=N0005072>
- [11] AENOR, "UNE 50132:1994" [Online]. Available: <https://www.une.org/encuentra-tu-norma/busca-tu-norma/norma/?c=N0005095>

- [12] AENOR, "UNE 50135:1996" [Online]. Available: <https://www.une.org/encuentra-tu-norma/busca-tu-norma/norma?c=N0005098>
- [13] IEEE, "How to Cite References: IEEE Documentation Style" [Online]. Available: <https://iee-dataport.org/help/how-cite-references-ieee-documentation-style>
- [14] G. v. Rossum, "Python" [Online]. Available: <https://www.python.org/>
- [15] D. Cournapeau, "Scikit-Learn" [Online]. Available: <http://scikit-learn.org/stable/>
- [16] W. McKinney, "Pandas" [Online]. Available: <http://pandas.pydata.org/>
- [17] T. Oliphant, "Numpy" [Online]. Available: <http://www.numpy.org/>
- [18] J. D. Hunter, "Matplotlib" [Online]. Available: <https://matplotlib.org/>
- [19] M. Waskom, "Seaborn" [Online]. Available: <https://seaborn.pydata.org/>
- [20] B. Silva, "Lazy Predict" [Online]. Available: <https://lazypredict.readthedocs.io/en/latest/>
- [21] Drafteados, "Estas estrellas NBA no merecían ir al All-Star" [Online]. Available: <https://www.youtube.com/watch?v=9h03NAdPi6w>
- [22] D. Cournapeau, "Scikit-Learn – Train Test Split" [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split
- [23] D. Cournapeau, "Scikit-Learn – Model Selection" [Online]. Available: https://scikit-learn.org/stable/modules/classes.html#module-sklearn.model_selection
- [24] D. Cournapeau, "Scikit-Learn - Pipeline" [Online]. Available: <https://scikit-learn.org/stable/modules/classes.html?highlight=pipeline#module-sklearn.pipeline>
- [25] D. Cournapeau, "Scikit-Learn - GridSearchCV" [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV
- [26] D. Cournapeau, "Scikit-Learn – Linear Discriminant Analysis" [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.discriminant_analysis.LinearDiscriminantAnalysis.html

[learn.org/stable/modules/generated/sklearn.discriminant_analysis.LinearDiscriminantAnalysis](https://scikit-learn.org/stable/modules/generated/sklearn.discriminant_analysis.LinearDiscriminantAnalysis)

[27] T. Bayes, "Teorema de Bayes" [Online]. Available: https://es.wikipedia.org/wiki/Teorema_de_Bayes

[28] D. Cournapeau, "Scikit-Learn – Quadratic Discriminant Analysis" [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.discriminant_analysis.QuadraticDiscriminantAnalysis

[29] D. Cournapeau, "Scikit-Learn – Nearest Neighbors Classification" [Online]. Available: <https://scikit-learn.org/stable/modules/neighbors.html#nearest-neighbors-classification>

[30] D. Cournapeau, "Scikit-Learn – Bagging" [Online]. Available: <https://scikit-learn.org/stable/modules/ensemble.html#bagging-meta-estimator>

[31] D. Cournapeau, "Scikit-Learn – Decision Trees" [Online]. Available: <https://scikit-learn.org/stable/modules/tree.html>

[32] D. Cournapeau, "Scikit-Learn – Random Forest" [Online]. Available: <https://scikit-learn.org/stable/modules/ensemble.html#random-forests>

[33] D. Cournapeau, "Scikit-Learn – Extra Trees" [Online]. Available: <https://scikit-learn.org/stable/modules/ensemble.html#extremely-randomized-trees>

[34] D. Cournapeau, "Scikit-Learn – AdaBoost" [Online]. Available: <https://scikit-learn.org/stable/modules/ensemble.html#adaboost>

[35] D. Cournapeau, "Scikit-Learn – Gradient Boosting" [Online]. Available: <https://scikit-learn.org/stable/modules/ensemble.html#gradient-tree-boosting>

[36] D. Cournapeau, "Scikit-Learn – SVM" [Online]. Available: <https://scikit-learn.org/stable/modules/svm.html>

[37] D. Cournapeau, "Scikit-Learn – MLP" [Online]. Available: https://scikit-learn.org/stable/modules/neural_networks_supervised.html#multi-layer-perceptron

- [38] D. Cournapeau, "Scikit-Learn – Confusion matrix" [Online]. Available: https://scikit-learn.org/stable/auto_examples/model_selection/plot_confusion_matrix.html#confusion-matrix
- [39] D. Cournapeau, "Scikit-Learn – Metrics" [Online]. Available: <https://scikit-learn.org/stable/modules/classes.html#sklearn-metrics-metrics>
- [40] D. Cournapeau, "Scikit-Learn – Accuracy" [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score
- [41] D. Cournapeau, "Scikit-Learn – Precision" [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_score
- [42] D. Cournapeau, "Scikit-Learn – Recall" [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.recall_score
- [43] D. Cournapeau, "Scikit-Learn – F1 Score" [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html#sklearn.metrics.f1_score