

# **ESCUELA POLITÉCNICA NACIONAL**

**FACULTAD DE SISTEMAS**

**UNIDAD DE TITULACIÓN**

**EVALUACIÓN Y APLICACIÓN DE ALGORITMOS DE  
INTELIGENCIA ARTIFICIAL EXPLICADA PARA PREDECIR EL  
DESEMPEÑO DE LOS CLUBES DE FÚTBOL BASADO EN LAS  
CARACTERÍSTICAS INDIVIDUALES DE SUS JUGADORES**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DE GRADO DE  
MAGISTER EN SISTEMAS DE INFORMACIÓN MENCION INTELIGENCIA DE  
NEGOCIOS Y ANALÍTICA DE DATOS MASIVOS**

**WALTER ANDRES CARGUA ABRIL**

walter.cargua@epn.edu.ec

**Director: Dr. Edison Loza**

edison.loza@epn.edu.ec

**Quito, marzo 2024**

## **APROBACIÓN DEL DIRECTOR**

Como director del trabajo de titulación: “Evaluación y aplicación de algoritmos de inteligencia artificial explicada para predecir el desempeño de los clubes de fútbol basado en las características individuales de sus jugadores” desarrollado por Walter Andrés Cargua Abril, estudiante de la Maestría en Sistemas de Información Mención en Inteligencia de Negocios y Analítica de Datos Masivos, habiendo supervisado la realización de este trabajo y realizado las correcciones correspondientes, doy por aprobada la redacción final del documento escrito para que prosiga con los trámites correspondientes a la sustentación de la Defensa Oral.

---

**Dr. Edison Loza**  
**DIRECTOR**

## **DECLARACIÓN DE AUTORÍA**

Yo, Walter Andrés Cargua Abril, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y que he consultado las referencias bibliográficas que se incluyen en este documento.

La Escuela Politécnica Nacional puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

---

**Ing. Walter Andrés Cargua Abril**

## **DEDICATORIA**

Con profundo amor y gratitud, dedico este trabajo a Sol y a nuestro precioso hijo Andrés. A Sol, por ser mi fuente de inspiración y mi compañera constante en este viaje. Tu amor, comprensión y apoyo incondicional han sido el faro que me ha guiado a través de los desafíos y logros. A mi hijo, por ser la luz y alegría de cada día, recordándome siempre la importancia de perseguir nuestros sueños con determinación y pasión.

Extiendo esta dedicatoria a mi querida familia: padres, hermanos y seres queridos. Cada uno de ustedes ha desempeñado un papel invaluable en mi vida, proporcionándome el amor, el apoyo y la motivación necesarios para alcanzar este logro.

Este logro es tanto mío como de ustedes, y con este trabajo, quiero honrar todo lo que han hecho por mí. Gracias por ser mi más grande fuente de fuerza y por estar siempre a mi lado.

## **AGRADECIMIENTO**

Primero que nada, deseo expresar mi más sincero agradecimiento a Sol y a mi hijo Andrés, quienes han sido mi fuente de inspiración y apoyo incondicional a lo largo de este viaje. Su amor, paciencia y comprensión han sido esenciales en los momentos más desafiantes de este proceso.

Asimismo, quisiera agradecer a mis padres Lastenia y Fredy, cuya educación y valores han sido la base de mi perseverancia y dedicación. Su ejemplo ha sido mi guía constante y su confianza en mí, el motor que me impulsó a seguir adelante.

Finalmente, no puedo dejar de mencionar a mi director de Edison Loza, cuya guía experta, consejos y críticas constructivas han sido fundamentales para el desarrollo y culminación exitosa de esta investigación. Su compromiso y dedicación han sido una fuente de aprendizaje y motivación constante.

Este logro no sólo es mío, sino también de todos ustedes que han sido parte esencial de este camino. Gracias por creer en mí y acompañarme en cada paso de esta valiosa experiencia.

# ÍNDICE DE CONTENIDO

<b>1. INTRODUCCIÓN</b>	<b>1</b>
1.1. Objetivo General	2
1.2. Objetivos específicos	2
1.3. Alcance	3
1.4. Marco Teórico	3
1.4.1. Aprendizaje de Máquina	3
1.4.2. Logistic Regression	4
1.4.3. Random Forest Classifier	4
1.4.4. Support Vector Machine	5
1.4.5. Artificial Neural Networks	6
1.4.6. Inteligencia Artificial Explicable	8
1.4.7. Trabajos Relacionados	11
1.4.8. Herramientas Utilizadas	12
<b>2. METODOLOGÍA</b>	<b>14</b>
2.1. Descripción General de la metodología CRISP-DM	14
2.2. Comprensión del negocio	16
2.2.1. Determinar los Objetivos del Negocio	16
2.2.2. Evaluación de la Situación	16
2.2.3. Plan del proyecto	17
2.3. Conocimiento de los Datos	18
2.3.1. Recolección de los Datos Iniciales	18
2.3.2. Descripción de los Datos	18
2.3.3. Exploración de los Datos	20
2.3.4. Calidad de los Datos	22
2.4. Preparación de los datos	24
2.4.1. Limpieza de datos	24
2.4.2. Preprocesamiento de Datos	26

<b>2.5. Modelado</b>	<b>40</b>
2.5.1. Selección de técnicas de AI	41
2.5.2. Plan de Pruebas para Evaluación de Algoritmos de Machine Learning	41
2.5.3. Construcción de Modelos de Machine Learning	42
2.5.4. Evaluación de Modelos	44
2.5.5. Implementación del algoritmo de XAI	45
<b>3. RESULTADOS</b>	<b>49</b>
<b>3.1. Modelos de Machine Learning</b>	<b>49</b>
3.1.1. Support Vector Machine	49
3.1.2. Random Forest	50
3.1.3. Red Neuronal Artificial	51
<b>3.2. Resultados de modelos de XAI</b>	<b>54</b>
3.2.1. SHAP	54
3.2.2. LIME	62
<b>4. CONCLUSIONES Y RECOMENDACIONES</b>	<b>66</b>
4.1. CONCLUSIONES	66
4.2. RECOMENDACIONES	68
<b>5. REFERENCIAS BIBLIOGRÁFICAS</b>	<b>69</b>
<b>6. ANEXOS</b>	<b>72</b>
ANEXO I	72
ANEXO II	73
ANEXO III	74

## Índice de Tablas

Tabla 1. Descripción del plan de proyecto.....	17
Tabla 2. Columnas seleccionadas para el estudio.....	19
Tabla 3. Cantidad de jugadores por año.....	22
Tabla 4. Agrupación de atributos de jugados en una sola variable.....	28
Tabla 5. Resultados del rendimiento de los algoritmos de aprendizaje automático para las clases Ganar (G) y No Ganar (NG). ....	44
Tabla 6. Parámetros de la clase "LimeTabularExplainer" .....	45
Tabla 7. Parámetros de la función "explain_instance" para la generación de explicaciones de una instancia determinada.....	46
Tabla 8. Parámetros de la función "shap.Explainer" para la generación de explicaciones. ....	47
Tabla 9. Parámetros para la generación de explicaciones de SHAP de una instancia determinada. ....	47
Tabla 10. Resultado de las métricas Precision, Recall y F1-Score para el modelo de clasificación Support Vector Machine.....	50
Tabla 11. Resultado de las métricas Precision, Recall y F1-Score para el modelo de clasificación Random Forest.....	51
Tabla 12. Resultado de las métricas Precision, Recall y F1-Score para el modelo de clasificación de Red Neuronal Artificial.....	52
Tabla 13. Instancia de la clase Ganar. ....	55
Tabla 14. Instancia de la clase No Ganar.....	56
Tabla 15. Valores medios de cada atributo en el conjunto de datos. ....	57



## Índice de Figuras

Figura 1. Función Sigmoide [7].	4
Figura 2. Random Forest Classifier [9].	5
Figura 3. Clasificador SVM [11].	5
Figura 4. Hiperplano SVM [11].	6
Figura 5. Modelo Neurona [12].	7
Figura 6. Topología típica ANN [12].	8
Figura 7. Funcionamiento de Lime [13].	10
Figura 8. Metodología CRISP-DM, Figura adaptada de [15].	14
Figura 9. Muestra categórica de la clasificación de los resultados del dataset de partidos jugados.	20
Figura 10. Cantidad de jugadores por año que posee el dataset de jugadores.	21
Figura 11. Dataset de jugadores desplegado dentro del sitio Kaggle.	23
Figura 12. Información general del jugador Karim Benzema en el sitio SOFIFA.	23
Figura 13. Información de partidos jugados en la Premier League en el sitio FBREF.	24
Figura 14. Aplicación comando drop.	25
Figura 15. Variable finales del dataset de jugadores.	25
Figura 16. Aplicación comando replace.	26
Figura 17. Variable finales del dataset de partidos jugados.	26
Figura 18. Aplicación comando merge.	27
Figura 19. Aplicación comando drop_duplicates.	27
Figura 20. Aplicación comando mean().	27
Figura 21. Aplicación comando get_dummies.	29
Figura 22. Variables con Valores Nulos.	29
Figura 23. Aplicación comando fillna.	29
Figura 24. Aplicación comando groupby en base al promedio de las variables.	30
Figura 25. Dataset con valores promedios de un solo equipo y los resultados de los partidos.	31
Figura 26. Aplicación comando merge con sufijos para diferenciar los equipos.	31
Figura 27. Dataset con valores promedios de los dos equipos y los resultados de los partidos.	31
Figura 28. Código para generar la matriz de correlación del dataset.	32
Figura 29. Matriz de correlación del dataset.	32
Figura 30. Histogramas Parte 1.	34
Figura 31. Histogramas Parte 2.	35
Figura 32. Características del histograma usando el algoritmo SelectKBest.	37
Figura 33. Muestra categórica de la clasificación de los resultados del dataset final.	38
Figura 34. Bloxpot de la varianza en la distribución de datos de cada atributo.	39
Figura 35. Bloxpot de la varianza en la distribución de datos de cada atributo después de realizar una estandarización.	40
Figura 36. Red Neuronal Artificial para evaluar el resultado de partidos de fútbol.	43
Figura 37. Matriz de confusión de los resultados del modelo de clasificación Support Vector Machine.	50
Figura 38. Matriz de confusión de los resultados del modelo de clasificación Random Forest.	51
Figura 39. Matriz de confusión de los resultados del modelo de clasificación con ANN.	52

Figura 40. Comparación de los resultados de la métrica Recall de los algoritmos implementados. ....	53
Figura 41. Gráfico de decisión de SHAP para realizar explicaciones de la instancia Ganar. ....	58
Figura 42. Gráfico de cascada de SHAP para realizar explicaciones de la instancia Ganar. ....	59
Figura 43. Gráfico de decisión de SHAP para realizar explicaciones de la instancia No Ganar. ....	60
Figura 44. Gráfico de cascada de SHAP para realizar explicaciones de la instancia No Ganar. ....	61
Figura 45. Explicaciones de LIME para una instancia clasificada como "Ganar". ....	62
Figura 46. Explicación detallada de LIME de la clase Ganar. ....	63
Figura 47. Explicaciones de LIME para una instancia clasificada como No Ganar. ....	64
Figura 48. Explicación detallada de LIME de la clase No Ganar. ....	65

## RESUMEN

El ámbito deportivo ha sido uno de los tantos sectores en donde el análisis de datos y la aplicación de modelos de aprendizaje automático han sido implementados para evaluar el desempeño de varias de sus áreas de interés en los últimos años. El presente proyecto analiza la aplicación de algoritmos de Machine Learning (ML) y técnicas de Inteligencia Artificial Explicable (XAI, por sus siglas en inglés) en el contexto de la clasificación de resultados en partidos de fútbol en base a las características de los jugadores de equipos de la Premier League. El texto discute la selección de algoritmos basados en su capacidad para ser analizados mediante herramientas de XAI como SHAP y LIME, y su adecuación al tipo de datos disponibles.

El plan de pruebas para la evaluación de los algoritmos de ML considera el uso de matrices de confusión y las métricas como la Precisión, Recall y F1-Score para una evaluación más minuciosa. Finalmente, se analizaron instancias de los resultados de los partidos de fútbol por medio de las técnicas de XAI y se obtuvieron gráficas explicativas donde se detallan qué atributos de los equipos y jugadores inciden en la ganancia o no ganancia de un juego de fútbol profesional.

**Palabras clave:** Machine Learning, Inteligencia Artificial Explicable, SHAP, LIME, fútbol profesional, análisis predictivo.

## ABSTRACT

The sports sector has been one of the main sectors in which data analysis and the application of machine learning models have been implemented to evaluate the performance of several of its areas of interest in recent years. This project analyzes the application of Machine Learning (ML) algorithms and Explainable Artificial Intelligence (XAI) techniques in the context of soccer match score classification based on player characteristics of Premier League teams. The text discusses the selection of algorithms based on their ability to be analyzed by XAI tools such as SHAP and LIME, and their suitability to the type of data available.

The test plan for the evaluation of ML algorithms considers the use of confusion matrices. and metrics such as Accuracy, Recall and F1-Score for a more thorough evaluation. Finally, instances of soccer match results were analyzed using XAI techniques and explanatory graphs were obtained detailing which attributes of teams and players affect the winning or non-winning of a professional soccer game.

**Keywords:** Machine Learning, Explainable Artificial Intelligence, SHAP, LIME, professional soccer, predictive analysis.

# 1. INTRODUCCIÓN

El deporte, en su papel de elemento fundamental en la vida social y cultural, atrae a masas de seguidores gracias a su inigualable capacidad para cautivar audiencias de todas las edades. En este contexto, el fútbol se ha consolidado como el deporte más apasionante y popular en las sociedades contemporáneas [1], generando flujos económicos significativos a través de la venta de entradas, merchandising, derechos de transmisión televisiva y transferencias de jugadores. Este fenómeno se refleja con claridad en las cifras de audiencia televisiva de los eventos futbolísticos más destacados a nivel global, como la Copa del Mundo o la UEFA Champions League. Por ejemplo, la final de la Copa del Mundo 2022 en Catar alcanzó una audiencia televisiva total de 1.500 millones de espectadores, superando con creces los 1.120 millones de espectadores registrados en 2018 [2].

A medida que el fútbol ha avanzado en su evolución, se ha vuelto un deporte cada vez más tecnificado, generando un vasto volumen de datos relacionados con el desempeño de equipos y jugadores. Sin embargo, la capacidad humana para procesar volúmenes tan colosales de información mediante técnicas estadísticas tradicionales se ve limitada. En contraste, las técnicas de Inteligencia Artificial (IA) brindan la oportunidad de superar estas restricciones, desentrañar patrones de rendimiento y explorar relaciones que, de otra manera, permanecerían inaccesibles.

En el contexto del fútbol, una multitud de características puede estar directamente vinculada al rendimiento de un equipo y sus resultados deportivos. Sin embargo, estas características resultan desafiantes para la comprensión humana al intentar predecir con alta precisión el desenlace de un partido [3]. A pesar de que las reglas del fútbol son relativamente simples, la predicción de resultados de partidos sigue siendo una tarea ardua. La singularidad del fútbol, caracterizada por la escasez de goles y el estrecho margen de victoria en la mayoría de los encuentros [4], lo convierte en un deporte impredecible.

En respuesta a esta complejidad, en los últimos años se ha observado un crecimiento en la aplicación de modelos de Machine Learning (ML) para asistir a las directivas de los

clubes en la toma de decisiones. Estos modelos se emplean en diversas áreas, desde la predicción de lesiones hasta la estimación del rendimiento de los deportistas [1], [5].

En este contexto, en este trabajo se propone desarrollar un modelo basado en Inteligencia Artificial Interpretativa, o explicada, (explainable Artificial Intelligence, XAI) que permita identificar las características físicas y deportivas de los jugadores de fútbol, tanto en ataque como en defensa, que influyen de manera determinante en los resultados de los partidos. La finalidad es proporcionar a las directivas de los clubes herramientas valiosas para la toma de decisiones estratégicas futuras, incluyendo la optimización de sus plantillas o la compra y venta de jugadores. Para lograrlo, se llevará a cabo un estudio de caso centrado en la Premier League, considerando su importancia como la liga de fútbol más seguida a nivel global, su elevado nivel competitivo y la disponibilidad de datos exhaustivos relacionados con el desempeño de los equipos de esta liga y sus características.

## **1.1. Objetivo General**

Evaluar y aplicar algoritmos de inteligencia artificial explicada para predecir el desempeño de los clubes de fútbol basado en las características individuales de sus jugadores.

## **1.2. Objetivos específicos**

- Realizar una revisión de trabajos relacionados con la predicción de desempeño deportivo y algoritmos de inteligencia artificial explicada.
- Diseñar un método de valoración de las capacidades de los equipos de fútbol a partir de las características deportivas de los futbolistas profesionales de la Premier League y los resultados de los partidos jugados, por medio de la extracción de información por web scraping y búsqueda de datos en la red.
- Probar y evaluar, siguiendo la metodología de CRISP-DM, el desempeño de tres diferentes algoritmos de ML supervisados en búsqueda del algoritmo que brinde el mejor rendimiento y menor margen de error con los datos recolectados.
- Aplicar métodos de inteligencia artificial explicada para determinar cuáles son las características deportivas que tienen mayor impacto en los resultados de un partido de fútbol.
- Visualización de los resultados obtenidos.

### **1.3. Alcance**

El alcance del este proyecto abarca la aplicación de la metodología de investigación CRISP – DM junto a algoritmos de ML supervisados y uso de técnicas de XAI para identificar las características principales de los jugadores de los equipos de fútbol profesional que afectan a los resultados de los partidos. Es importante destacar que este proyecto no involucra ni la automatización de la recopilación de datos ni la creación de interfaces que permita su manipulación con el usuario final.

### **1.4. Marco Teórico**

En esta sección, se presentan las definiciones proporcionadas por varios autores sobre los temas desarrollados en este proyecto. Entre estas temáticas se incluyen conceptos como técnicas de ML, Logistic Regression, Random Forest Clasifier, Support Vector Machine, Artificial Neural Networks y técnicas de XAI, Local Interpretable Model-Agnostic Explanation Method (LIME) y Shapley Additive Explanations (SHAP).

#### **1.4.1. Aprendizaje de Máquina**

Los algoritmos de ML tienen la característica de consumir y procesar datos con el fin de identificar patrones relacionados con individuos, transacciones, eventos, etc. Los algoritmos de ML se dividen principalmente en cinco categorías: Aprendizaje supervisado, Aprendizaje no supervisado, Aprendizaje semisupervisado, Aprendizaje Profundo y Aprendizaje por refuerzo [6].

Para este proyecto se utiliza técnicas de aprendizaje automático supervisado ya que el proceso de aprendizaje supervisado implica que el algoritmo debe aprender a mapear las entradas a las salidas observando ejemplos de entrenamiento lo cual es posible por la naturaleza de los datos de este estudio que son resultados de partidos de fútbol.

### 1.4.2. Logistic Regression

La regresión logística es una técnica derivada de la regresión lineal, pero con el objetivo de entrenar un clasificador que pueda tomar una decisión binaria sobre la clase de una nueva observación de entrada. Para tomar esta decisión, se utiliza el clasificador sigmoide, ilustrado en la Figura 1 [7].

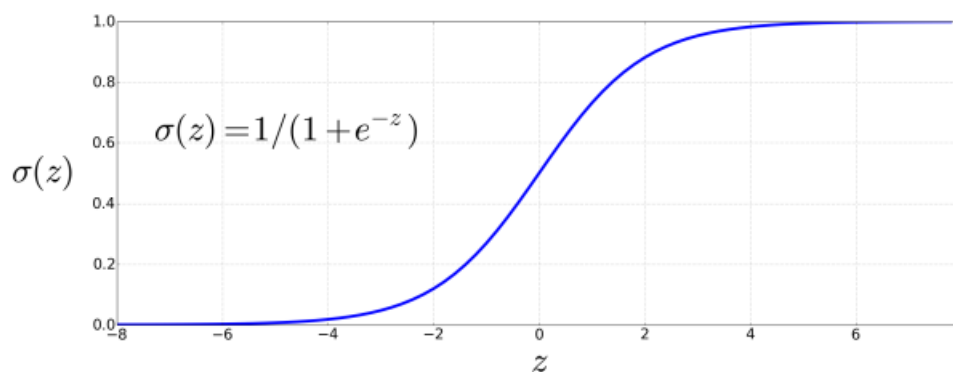


Figura 1. Función Sigmoide [7].

La función sigmoide toma un valor real y lo asigna al intervalo (0,1). Es casi lineal en torno a cero, pero los valores extremos se desplazan hacia 0 o 1.

### 1.4.3. Random Forest Classifier

Random Forest (RF) es un algoritmo de aprendizaje colectivo para clasificación, regresión y otras tareas. Funciona creando un grupo de árboles de decisión en el momento del entrenamiento y produce una clase que es la predicción media de los árboles individuales. El RF tiene una relación directamente proporcional entre el número de árboles del bosque y la precisión resultante [8]. En la Figura 2 se presenta como cada árbol se entrena con un subconjunto de características del dataset y hace su propia predicción para posteriormente determinar la clasificación final mediante una votación mayoritaria entre las predicciones de todos los árboles.



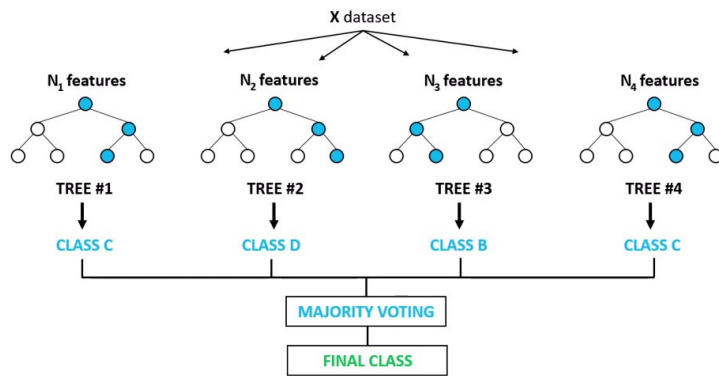


Figura 2. Random Forest Classifier [9].

#### 1.4.4. Support Vector Machine

El método de Máquina de Soporte Vectorial (SVM) es una técnica de clasificación de patrones. A diferencia de los enfoques convencionales que se centran en minimizar los errores de entrenamiento empíricos, SVM adopta una estrategia que busca minimizar el margen de error superior al maximizar la separación entre el hiperplano y los datos de entrenamiento. Las soluciones generadas por SVM son siempre únicas y globalmente óptimas. Esto contrasta con otros métodos de entrenamiento de redes, que requieren optimización no lineal y presentan el riesgo de quedar atrapados en mínimos locales [10].

Para comprender su funcionamiento, se puede imaginar un espacio vectorial en el que se distribuyen dos clases distintas de puntos, representados como  $x_i$ ,  $y_j$ . Estas clases ocupan regiones separadas en el espacio, lo que permite su clasificación mediante un hiperplano separador definido por la ecuación  $w^T x = 0$ , como se muestra en la Figura 3 [11].

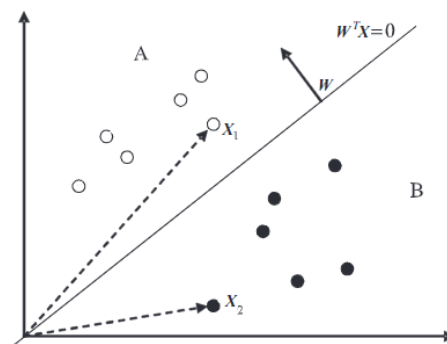


Figura 3. Clasificador SVM [11].

Si se desea determinar la clase de un punto  $x_1$ , se requiere evaluar el ángulo entre el vector  $w$ , asociado al plano y la muestra  $x_1$  para determinar si es mayor o menor a 90 grados. Si el ángulo es menor a 90 grados, la muestra se clasifica como perteneciente a la clase A; de lo contrario, se clasificará como clase B. Esto se logra calculando el producto escalar entre los vectores. Un valor positivo indica que el ángulo es menor a 90 grados, lo que corresponde a la clase A.

Por otro lado, para un vector  $x_2$ , el ángulo será mayor a 90 grados, resultando en un producto escalar negativo y la clasificación como clase B. Las muestras situadas en el plano son ortogonales a  $w$ , lo que da como resultado un producto escalar igual a cero. En esta situación, no podemos determinar su clase. Es en este punto donde se introduce un hiperplano para clasificar los elementos en el plano cartesiano. El procedimiento de SVM implica colocar el hiperplano lo más lejos posible de la muestra más cercana, como se muestra en la Figura 4 [11].

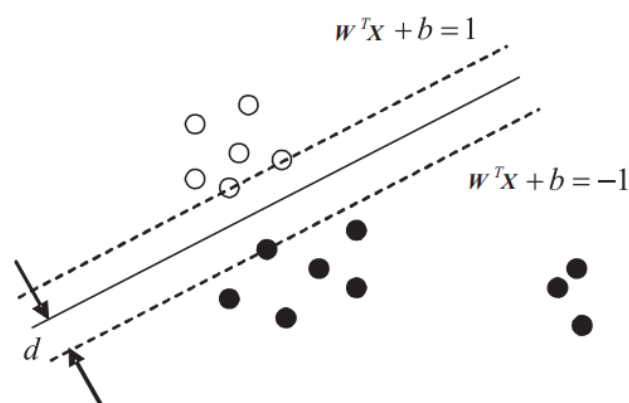


Figura 4. Hiperplano SVM [11].

#### 1.4.5. Artificial Neural Networks

Una red neuronal artificial (ANN, por sus siglas en inglés) es un sistema de procesamiento de información inspirado en las neuronas interconectadas de sistemas biológicos. Dos aplicaciones destacadas de las ANNs incluyen el reconocimiento de patrones y la predicción/forecasting. Una ANN aprende a realizar tareas sin ser programada

explícitamente con reglas específicas de la tarea. En cambio, aprende al ser presentada con datos y modificar los parámetros internos de la ANN para minimizar los errores. Una red neuronal consta de muchas neuronas interconectadas. Cada nodo realiza una computación independiente, unida por conexiones que generalmente tienen pesos ( $w$ ), como se puede observar en la Figura 5. Durante la fase de entrenamiento de una ANN, los pesos de cada conexión se ajustan y afinan en función de los datos presentados.

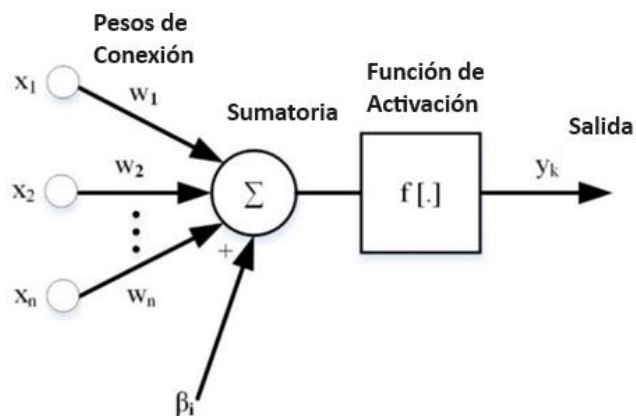


Figura 5. Modelo Neurona [12].

Las funciones de activación habituales abarcan la función identidad, el escalón, la unidad lineal rectificadora, sigmoide, la tangente hiperbólica y la gaussiana. Es importante destacar que existen diversas funciones de activación que se seleccionan en función de los datos y la aplicación específica.

Las ANNs típicamente consisten en tres tipos principales de capas: una capa de entrada, capa(s) oculta(s) y una capa de salida, como se puede observar en la Figura 6. Una ANN puede tener múltiples niveles de capas ocultas. La capa de entrada consta de la(s) variable(s) del regresor que se utilizan para obtener una estimación de la variable objetivo. Las conexiones entre las capas están ponderadas, y los pesos se determinan durante la fase de entrenamiento de la red neuronal [12].

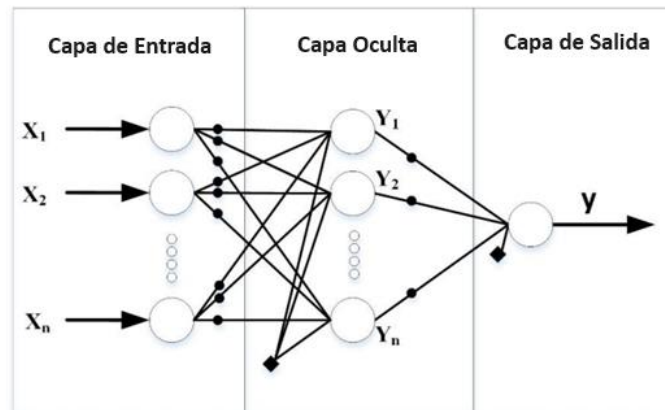


Figura 6. Topología típica ANN [12].

#### 1.4.6. Inteligencia Artificial Explicable

Cuando se evalúan los resultados de un sistema de IA, se tienen diferentes enfoques dependiendo de la complejidad del modelo que se emplea para proporcionar explicaciones [13]. En base a esto, el campo de la XAI generalmente comparte un objetivo fundamental: hacer que la IA sea accesible y comprensible para las personas. Al adoptar una definición pragmática y centrada en el ser humano en su enfoque, se concibe que la XAI engloba una variedad de enfoques técnicos destinados a lograr esta comprensión. Esto incluye métodos como la interpretación directa, la generación de explicaciones o justificaciones, y la provisión de información sobre la transparencia, entre otros [14].

En términos generales, las técnicas de XAI se dividen en dos categorías principales: la elección de modelos directamente interpretables, como árboles de decisión o regresiones lineales simples, y la elección de modelos complejos y opacos (llamados "modelos de caja negra") junto con el uso de técnicas posteriores para generar explicaciones. La elección entre estos enfoques a menudo se discute en términos de "compensación entre rendimiento e interpretabilidad", ya que los modelos opacos tienden a tener un mejor rendimiento en muchas tareas [14].

Sin embargo, en la práctica, a menudo se eligen modelos opacos debido a su ventaja en rendimiento para un conjunto de datos dado, a menudo un menor requisito de esfuerzo humano o la disponibilidad de soluciones listas para usar. En estos casos, es necesario utilizar técnicas de XAI posteriores para hacer que los modelos sean explicables [14].

## Explicación de modelos de caja negra

Una estrategia avanzada en el ámbito de la XAI implica la utilización de modelos de IA complejos, a menudo denominados "cajas negras", que ofrecen un alto nivel de precisión en sus predicciones. Posteriormente, se busca comprender y explicar el razonamiento detrás de las decisiones de estos modelos una vez que han sido entrenados. Estos enfoques se conocen como métodos post-hoc, ya que se aplican después de que el modelo ha completado su fase de entrenamiento [15].

El reto en el ámbito de la XAI en la etapa post hoc es lograr que un modelo de caja negra sea cuantificable en retrospectiva. Para abordar este desafío, se aplican diversos métodos, algunos de los cuales registran información durante el proceso de entrenamiento, mientras que otros revisan exhaustivamente el modelo para su cuantificación [15].

Entre los métodos más comunes utilizados para generar explicaciones post-hoc se incluyen el de LIME y el de Shap, y que son empleados en este trabajo. Estos enfoques son fundamentales para obtener una comprensión más profunda de modelos complejos, lo que resulta esencial en situaciones donde los datos recopilados son de alta dimensionalidad y complejidad.

**LIME:** "Local Interpretable Model-Agnostic Explanation" es un algoritmo versátil que se puede utilizar con diversos modelos de clasificación de ML, lo que le permite proporcionar explicaciones precisas de tipo post-hoc que justifican las predicciones locales. El objetivo de la interpretación local no es describir el comportamiento general del modelo en su totalidad, sino más bien proporcionar una comprensión de por qué el modelo produce resultados específicos para una entrada particular [16]. Para llevar a cabo su tarea, LIME somete al modelo a pruebas ajustando y variando sus entradas para su posterior análisis. Esto se hace con el propósito de comprender cómo cada variable de entrada influye en las diferentes predicciones del modelo [17].

La operación de LIME se puede visualizar de manera gráfica, como se ilustra en la Figura 7. Inicialmente, se generan numerosas muestras que son similares al ejemplo que se busca explicar (cruz roja gruesa). Estas muestras se crean al modificar cada característica de

entrada siguiendo una distribución normal, y se obtiene la salida del modelo para cada una de ellas [13].

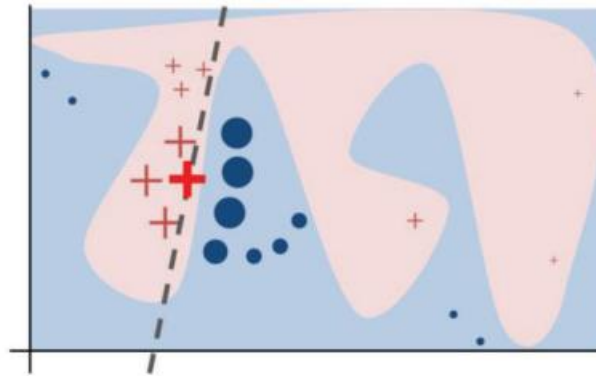


Figura 7. Funcionamiento de Lime [13].

Posteriormente, se evalúa la distancia entre los resultados del ejemplo original y las muestras generadas, y se ordenan las características en función de su cercanía. Finalmente, se aplica la técnica de análisis de regresión Lasso (least absolute shrinkage and selection operator) para identificar las características más relevantes [13].

**SHAP:** “Shapley Additive Explanations” se enfoca en analizar el impacto de cada característica en la salida del modelo. Para lograr esto, emplea un modelo explicativo más simple que el modelo original, con tres características deseables clave: precisión local, lo que significa que proporciona predicciones iguales para un ejemplo específico; mínima pérdida, al simplificar las entradas en un espacio de características más reducido; y, consistencia, que garantiza que, si una característica se mantiene constante o aumenta independientemente de las demás, su importancia no disminuye [13].

La relevancia de cada característica se determina mediante los valores de SHAP, que son los valores de Shapley, derivados de la teoría de juegos, aplicados a una función de expectativa condicional del modelo original.

El método para calcular los valores de Shapley fue propuesto por Lloyd Shapley en con el propósito de medir la contribución de cada jugador en un juego cooperativo [18]. La idea subyacente es que en un juego en el que “n” jugadores trabajan colectivamente para obtener una recompensa “p”, dicha recompensa debe dividirse entre los “n” jugadores de

acuerdo con la contribución individual de cada uno, lo que se conoce como el valor de Shapley.

Para calcular el valor de Shapley de un jugador en este contexto, se deben considerar todos los subconjuntos posibles del conjunto que contiene a todos los jugadores y que incluyen al jugador en cuestión. Para cada uno de estos subconjuntos, se calcula la diferencia entre el beneficio con la presencia del jugador y el beneficio sin su presencia. Esta diferencia se calcula mediante el factor  $\frac{(s-1)!(n-s)!}{n!}$ , donde "n" es el número total de jugadores y "s" representa el número de jugadores en el subconjunto en consideración. Shapley propuso que el valor de un juego representado como  $(N, v)$  para cada jugador  $i \in N$  se calcula mediante la siguiente fórmula [15].

$$\varphi_i(v) = \sum_{s \subseteq N: i \in S} \frac{(s-1)!(n-s)!}{n!} [v(S) - v(S-i)] \text{ donde } n = |N| \text{ y } s = |S|$$

El valor de Shapley se interpreta como la contribución marginal esperada del jugador  $i$ , es decir, el valor promedio de las contribuciones marginales  $[v(S) - v(S-i)]$  del jugador a todas las coaliciones no vacías  $S \in 2^N$ . Se considera que el tamaño de la coalición del jugador ( $1 \leq s \leq n$ ) es equiprobable, y todas las coaliciones de tamaño  $S$  tienen la misma probabilidad [15].

Usando estos valores, la explicación se construye como la suma de las contribuciones individuales. Estas explicaciones se contrastan con un valor de referencia que representa la predicción esperada si no tuviéramos conocimiento de ninguna de las características de la salida obtenida. Este proceso de comparación permite destacar cómo cada característica influye en la predicción y proporciona una valiosa perspectiva sobre la importancia relativa de cada variable en el modelo.

#### 1.4.7. Trabajos Relacionados

Existen varios trabajos en los que se emplea técnicas de aprendizaje automático en el ámbito deportivo profesional a pesar de que en muchas áreas del deporte se siguen

empleando herramientas estadísticas más tradicionales. Uno de los estudios revisados en este proyecto es el que se plantea en [19] el cual explora un enfoque de XAI para predecir los resultados de los partidos en la Liga Brasileña de Voleibol y que en una de sus fases de desarrollo hace el uso de modelos de SVM y de Redes Neuronales Profundas (DNN) para obtener un rendimiento predictivo en los resultados de los partidos de voleibol, y SHAP para estimar la contribución de cada característica en la predicción del modelo. Con respecto a los trabajos que se enfocan en la predictibilidad del rendimiento competitivo de los deportistas basado en sus características morfológicas y pruebas motoras se tiene en caso estudiado en [20], en el cual se incluyó data de 1002 tenistas, tanto hombres como mujeres, y que ayudó a confirmar que los métodos de ML para identificar las variables más prometedoras resultaron ser más exitosos que los de los entrenadores. En el ámbito del fútbol, existen trabajos que utilizan técnicas de ML para prevenir lesiones en los deportistas [21], ayudar en el scouting en clubes de fútbol [22], la predicción de resultados de los partidos de fútbol [23] e incluso se han realizado análisis de anomalías de desempeño utilizando minería de opiniones [24]. Sin embargo, no se ha profundizado en el uso de técnicas de XAI enfocados en el fútbol profesional, de la manera en que se ha hecho por ejemplo en el ámbito de la medicina como por ejemplo en los estudios [15],[16].

Por este motivo este estudio se enfoca en la aplicación de técnicas de XAI para ayudar a los clubes a definir las mejores características de los jugadores para poder ganar un partido.

#### **1.4.8. Herramientas Utilizadas**

El desarrollo de este Proyecto se lo realizó en el lenguaje de programación Python en la plataforma de Google llamada Colab, que por medio del navegador permite programar y ejecutar Python de forma gratuita haciendo uso de las GPUs y TPUs de Google, y poder archivar el avance del proyecto en la nube [25].

Se utilizaron varias librerías disponibles en Python para realizar la lectura, limpieza y manejo de datos, así como librerías que permiten aplicar algoritmos de aprendizaje automático, XAI y realizar gráficas. Entre las más importantes tenemos las siguientes:

**Pandas:** carga, manipulación y análisis de datos.



**NumPy:** realizar cálculos numéricos en datos que suelen estar representados en forma de matrices.

**Matplotlib:** creación de gráficos.

**Requests:** herramienta para realizar solicitudes HTTP a través de la web.

**BeautifulSoup:** herramienta para la extracción de datos en aplicaciones de web scraping y procesamiento de datos web.

**Scikit-learn:** ofrece una amplia variedad de herramientas y algoritmos para tareas de aprendizaje supervisado y no supervisado, así como para la selección y evaluación de modelos.

**TensorFlow:** biblioteca desarrollada por Google utilizada para entrenar modelos de aprendizaje automático, en especial, modelos de redes neuronales.

**Lime:** librería para explicar modelos de aprendizaje automático de forma local e interpretable.

**Shap:** librería utilizada para explicar el modelo de aprendizaje automático mediante la asignación de importancias a las características o variables de entrada.

## 2. METODOLOGÍA

### 2.1. Descripción General de la metodología CRISP-DM

Este estudio se orientará bajo el marco de la metodología CRISP-DM [15] con el propósito de alcanzar los objetivos detallados. Las etapas que conforman esta metodología se presentan en la Figura 8.

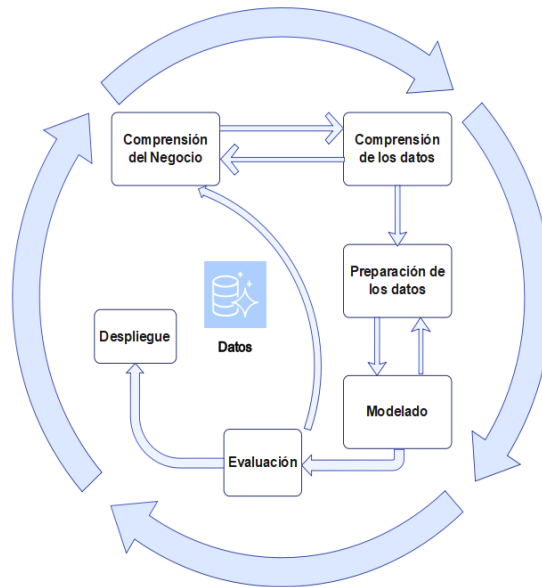


Figura 8. Metodología CRISP-DM, Figura adaptada de [15].

#### **Comprensión del Negocio:**

En esta fase se define claramente el objetivo del proyecto y se establecen los criterios específicos que determinarán su éxito. Se busca entender el problema desde la perspectiva de la organización o del sector al que está dirigido. Es fundamental tener en cuenta las expectativas y necesidades de los usuarios o beneficiarios. La correcta definición en esta etapa sienta las bases para el desarrollo de las fases subsiguientes.

#### **Comprensión de los Datos:**

Durante esta fase, se realiza un análisis preliminar de los datos a disposición. Es importante obtener un panorama general de las características y estructura de los datos.

Paralelamente, se valora su calidad basándose en su potencial para satisfacer los objetivos del proyecto definidos anteriormente.

### **Preparación de los Datos:**

Aquí se realizan todos los procesos necesarios para transformar y limpiar los datos, preparándolos para la fase de modelado. Esto puede incluir la corrección de inconsistencias, el tratamiento de valores faltantes y la selección de características relevantes. La meticulosidad en esta fase es crucial para asegurar la efectividad del modelo.

### **Modelado:**

A lo largo de esta fase, se exploran y aplican diversas técnicas de modelado adaptadas al problema en cuestión. Es fundamental tener en cuenta que ciertas técnicas pueden requerir especificaciones particulares en los datos, lo que podría necesitar reajustes en la etapa de preparación de datos. Las actividades clave incluyen la elección de la técnica de modelado adecuada, el diseño de pruebas para validar la eficacia del modelo y, finalmente, su implementación.

### **Evaluación:**

Una vez desarrollado el modelo, es esencial evaluar su desempeño y eficacia en relación con los objetivos propuestos para el proyecto de minería de datos. Esta etapa permite identificar áreas de mejora y reajustes. Se realiza una revisión exhaustiva para asegurar que todos los elementos cruciales han sido considerados y se contempla la posibilidad de reutilizar el modelo o sus componentes en futuros proyectos.

### **Despliegue:**

Los resultados y modelos se presentan de manera que puedan ser comprendidos y utilizados por los stakeholders o beneficiarios. Esto puede incluir la implementación en un sistema productivo, la formación de usuarios y la documentación pertinente.

## **2.2. Comprensión del negocio**

### **2.2.1. Determinar los Objetivos del Negocio**

El proceso inicial consiste en recopilar datos deportivos de múltiples fuentes. La información adquirida se analizará y organizará de manera que nos permita relacionar los resultados de partidos con las características de los jugadores que participaron en cada contienda.

El objetivo final es analizar las características deportivas que influyen más para que un equipo de fútbol gane un partido de tal manera que esta información ayude a la directiva del club a identificar a los jugadores que más aportarían al equipo o que habilidades deberían ser mejoradas en los jugadores.

### **2.2.2. Evaluación de la Situación**

Actualmente los estudios sobre la XAI son mayormente aplicados al campo de la medicina, en donde los médicos necesitan entender por qué un modelo de IA está haciendo una recomendación particular para poder confiar en ella. Esto se puede apreciar en los estudios realizados en [15][16], donde la XAI apoya a la toma de decisiones médicas en la salud fetal.

En el ámbito deportivo el ML y la XAI han sido empleadas en varios aspectos, la predicción de lesiones en atletas [21], el análisis de anomalías de desempeño en futbolistas usando minería de opiniones [24], o más específicamente como en [19], en donde se aplica el método de SHAP en las predicciones de los resultados de partidos de Volleyball. Sin embargo, actualmente no existen demasiados estudios que hagan uso de técnicas de XAI para evaluar las características más relevantes de jugadores para que los equipos de fútbol puedan ganar un partido.

### 2.2.3. Plan del proyecto

El esquema de trabajo presentado en la Tabla 1 describe las acciones a llevar a cabo en cada una de las 7 etapas del CRISP-DM, junto con la duración estimada de cada tarea. Al concluir la tabla, encontraremos la suma total de horas proyectadas para la finalización integral del proyecto.

Tabla 1. Descripción del plan de proyecto.

<b>Actividad</b>		<b>Duración (horas)</b>
<b>Conocimiento del negocio</b>	Revisar literatura sobre modelos de XAI en el contexto deportivo.	20
	Evaluar los repositorios de datos de EA Sports.	10
	Revisar la información sobre resultados de partidos jugados en la web	10
<b>Conocimiento de los datos</b>	Recolectar y clasificar los datos	20
	Obtener información sobre jugadores y partidos de la Premier League	10
<b>Preparación de los datos</b>	Limpieza y preprocesamiento de datos	20
	Integración de datos	10
	Crear conjuntos de datos para entrenamiento y validación.	10
<b>Modelado</b>	Modelado de tres algoritmos predictivos que permitan identificar el resultado de un partido de fútbol usando técnicas de machine learning supervisado.	35
<b>Evaluación</b>	Análisis de los modelos predictivos aplicados	25
<b>Implementación</b>	Implementación de algoritmos de XAI para determinar cuáles son las características deportivas que más influyen en el resultado de un partido de fútbol.	25
<b>Generación del reporte</b>	Redacción del documento de titulación	25
<b>Total</b>		220

## **2.3. Conocimiento de los Datos**

### **2.3.1. Recolección de los Datos Iniciales**

En el desarrollo de este proyecto, se optó por emplear 2 fuentes primarias de información: SOFIFA<sup>1</sup> y FBREF<sup>2</sup>. SOFIFA es una plataforma en línea que alberga los datos recabados por EA Sports, una reconocida empresa en el mundo de los videojuegos, que, si bien es popular por sus juegos relacionados con el fútbol, también ha diversificado su enfoque hacia otros deportes, como el baloncesto, fútbol americano y béisbol, entre otros. EA Sports invierte considerablemente en recabar datos detallados sobre habilidades y rendimientos de los deportistas para ser reflejados en sus juegos electrónicos. Para acceder a la vasta información de SOFIFA de manera ordenada, se recurrió a un repositorio en Kaggle. La autoría de estos archivos corresponde a Stefano Leone, con una trayectoria de más de seis años en Kaggle y reconocido como un Especialista en Conjuntos de Datos dentro de la plataforma.

Adicionalmente, se examinó una segunda fuente: el portal FBREF, donde se obtuvo por medio de web scraping, usando librerías de Python como BeautifulSoup, información sobre los partidos jugados en la Premier League, durante los últimos 10 años, toda esta información se unió para obtener un dataset que contenga información sobre características de los jugadores e información de los partidos jugados en la Premier League.

### **2.3.2. Descripción de los Datos**

El dataset<sup>3</sup> de Kaggle que posee información de jugadores en el portal SOFIFA contiene registro de los jugadores de clubes de fútbol que han estado activos desde los años 90 y cuenta con un total de 10003590 filas y 110 columnas, y de los cuales 415242 filas pertenecen a jugadores que han participado en la Premier League.

---

<sup>1</sup> <https://sofifa.com/>

<sup>2</sup> <https://fbref.com/en/>

<sup>3</sup> [https://www.kaggle.com/datasets/stefanoleone992/fifa-23-complete-player-dataset?select=male\\_players.csv](https://www.kaggle.com/datasets/stefanoleone992/fifa-23-complete-player-dataset?select=male_players.csv)

Tabla 2. Columnas seleccionadas para el estudio.

<b>Campo</b>	<b>Descripción del contenido</b>
result	Resultado del partido, donde 1 es ganar y 0 es no ganar (empatar o perder)
overall	Refleja una estimación cualitativa del rendimiento global de un jugador
age	Edad del jugador
height_cm	Altura del jugador
weight_kg	Peso del jugador
weak_foot	Habilidad de su pie secundario (1 a 5)
skill_moves	Habilidad de movimientos (1 a 5)
international_reputation	Reputación internacional (1 a 5)
pace	Atributo del jugador que determina la velocidad de un jugador al trotar y correr en el campo
shooting	Atributo de un jugador clasificado de 0 a 99 que define la fuerza y precisión general del tiro del jugador
dribbling	Atributo del jugador que determina la capacidad de un jugador para llevar el balón y superar a un adversario mientras tiene el control
defending	Atributo que determina la capacidad defensiva de un jugador
mentality_composure	Habilidad de mantener la calma y el control de sus emociones durante los partidos
xg	Goles esperados del jugador durante un partido
poss	Posesión del balón del equipo, calculado por el porcentaje de pases intentados
sh	Total de disparos realizados por el jugador en un partido
sot	Tiros a portería realizados por el jugador en un partido
dist	Distancia promedio, en yardas, del total de disparos hechos hacia portería en un partido
pk	Tiros penales ejecutados durante el partido
attacking_skills	Promedio de atributos de ataque (remates, cruce, precisión con la cabeza, pase corto, voleas)
movement_speed_skills	Promedio de atributos de velocidad de movimientos (aceleración, velocidad de carrera, agilidad, reacción, balance)
power_skills	Promedio de atributos de potencia (potencia de disparo, salto, resistencia, fuerza, pases largos)
mentality_aggression_skills	Promedio de atributos de mentalidad de agresión (agresión, interceptión)
mentality_vision_skills	Promedio de atributos de mentalidad de visión (posicionamiento, visión, penales)
defending_skills	Promedio de atributos de defensa (vigilancia defensiva, derribo de pie, derribo por deslizamiento)
unique	Atributo que define si el jugador tiene un cuerpo único o no
home	Define si el equipo jugó de local o de visitante (1 local, 0 visitante)

Por otra parte, el dataset de FBREF contiene información de partidos jugados en la Premier League durante los años 2012 al 2023 y cuenta con un total de 26 columnas. Son un total de 4070 partidos diferentes ya que el dataset tiene para un mismo partido jugado en cada

fecha la información tomando al equipo local como equipo número 1 y al equipo visitante como equipo número 2 y viceversa, al equipo visitante como equipo número 1 y al equipo local como equipo número 2. Esto doblaría el número de partidos al doble con un total de 8140 partidos.

Sin embargo, luego de combinar los dos datasets y realizar un análisis de los datos, se tiene un dataset con información de 8140 partidos (4070 partidos diferentes) con 52 columnas que contienen información de los valores promedios de ciertas características de los jugadores y de los partidos, y que se puede apreciar en la Tabla 2.

Cabe aclarar que para cada partido se tiene los atributos mencionados en la tabla anterior para los dos equipos participante. Por ejemplo, el overall del equipo 1 tiene como nombre overall\_t (team) y del equipo 2 tiene como nombre overall\_o (opponent).

### 2.3.3. Exploración de los Datos

Como se mencionó en la sección anterior se tiene información de 8140 partidos con sus respectivos resultados (ganar = W, perder = L, empatar = D).

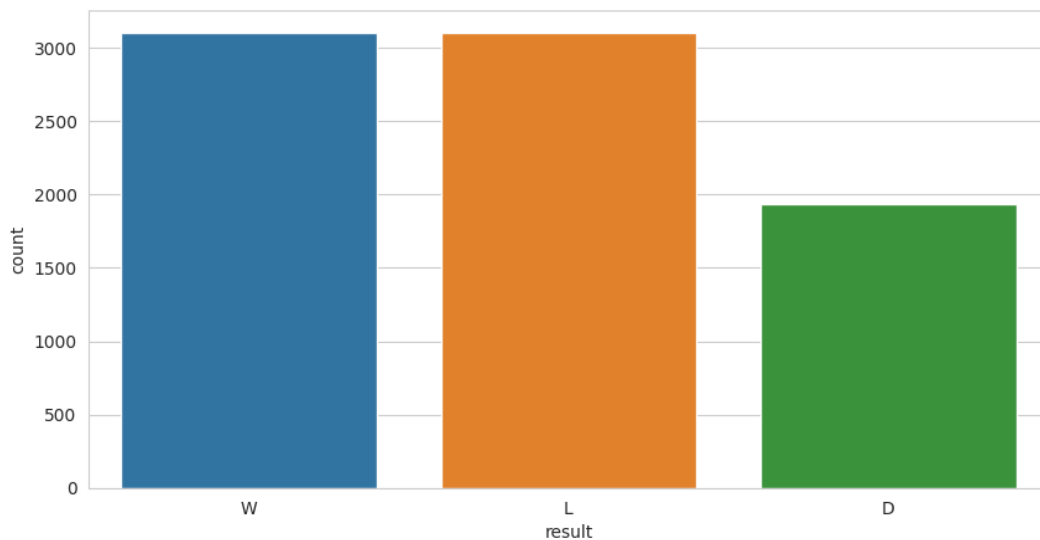


Figura 9. Muestra categórica de la clasificación de los resultados del dataset de partidos jugados.



Los partidos ganados son un total de 3120, se tiene un total de partidos perdidos de 3102 y partidos empatados 1936, como se puede apreciar en la Figura 9.

Por otra parte, la información sobre los jugadores contiene a jugadores que han permanecido activos en sus equipos desde los años 90, como es el caso de Steve Harper portero que comenzó su carrera profesional en 1993 y se retiró en 2016. Estos datos se filtran a futuro en base al año de actualización de FIFA, dado que la información sobre los resultados de los partidos va solo desde el año 2012.

El dataset inicial de jugadores cuenta con información de deportistas que juegan en diferentes posiciones (GK, RB, LB, ST, etc) y tienen diferentes atributos deportivos. El número de jugadores iniciales que se tiene de la Premier League contando la fecha de actualización de SOFIFA, sin contar los jugadores de reserva es de 415242, de estos se tienen el siguiente número de jugadores de los años que son de interés en este proyecto mostrados por medio del gráfico de barras de la Figura 10 y la Tabla 3.

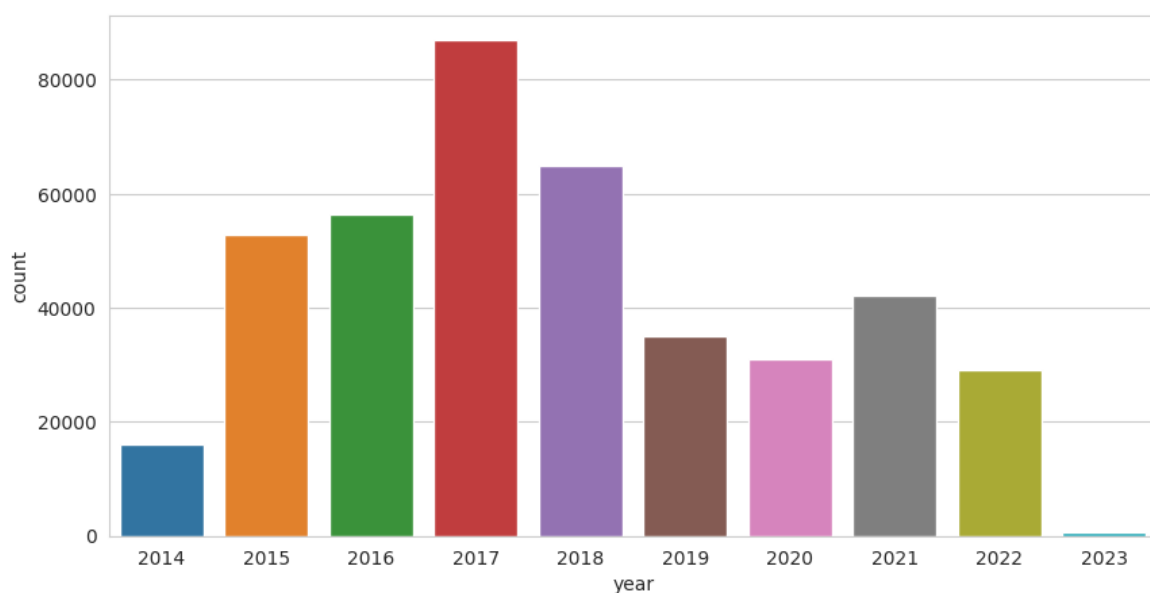


Figura 10. Cantidad de jugadores por año que posee el dataset de jugadores.

Tabla 3. Cantidad de jugadores por año

<b>Año</b>	<b>Número de Jugadores</b>
<b>2014</b>	16146
<b>2015</b>	52888
<b>2016</b>	56500
<b>2017</b>	86965
<b>2018</b>	64836
<b>2019</b>	34976
<b>2020</b>	30958
<b>2021</b>	42228
<b>2022</b>	29170
<b>2023</b>	575
<b>TOTAL</b>	<b>415242</b>

Para completar los años faltantes 2012 y 2013 se tomará en cuenta los jugadores que se han unido a sus equipos durante esos años. Así tenemos que la cantidad de jugadores para el año 2012 es de 29078 y para el año 2013 es de 35274.

#### **2.3.4. Calidad de los Datos**

Los datos de SOFIFA obtenidos por medio de Kaggle, han sido de alta usabilidad. Según información del propio portal este dataset ha sido descargado alrededor de 5000 veces hasta la fecha de creación de este documento.

Observando la Figura 11, es evidente que el portal web facilita la inspección de la información albergada en los archivos CSV. Se puede apreciar que una columna específica del conjunto de datos muestra la dirección web exacta de donde se extrajeron los detalles del futbolista; en este escenario, la URL corresponde a la información de Karim Benzema, la cual nos dirige a la información del jugador en el portal SOFIFA como se puede apreciar en la Figura 12.

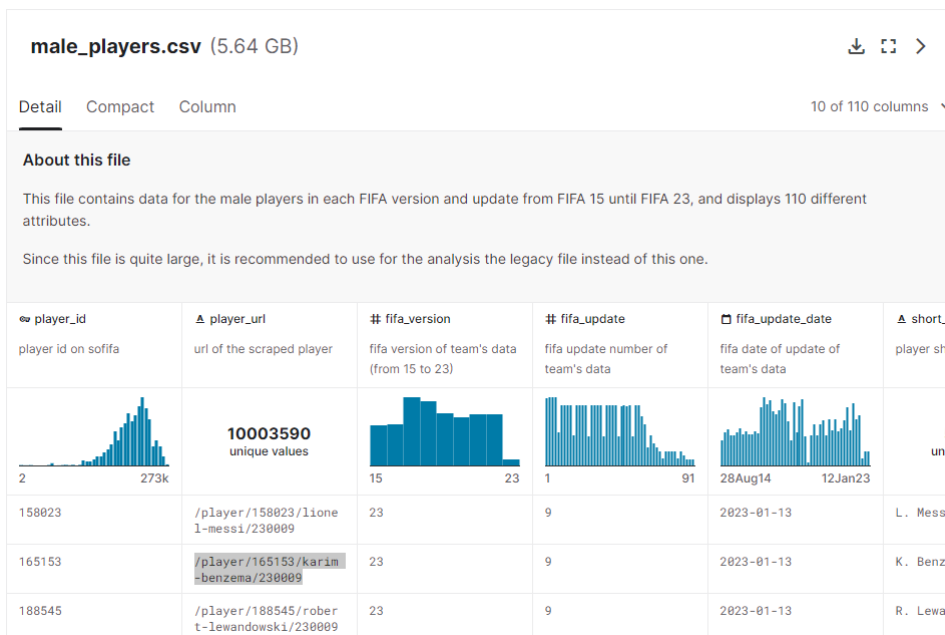


Figura 11. Dataset de jugadores desplegado dentro del sitio Kaggle.

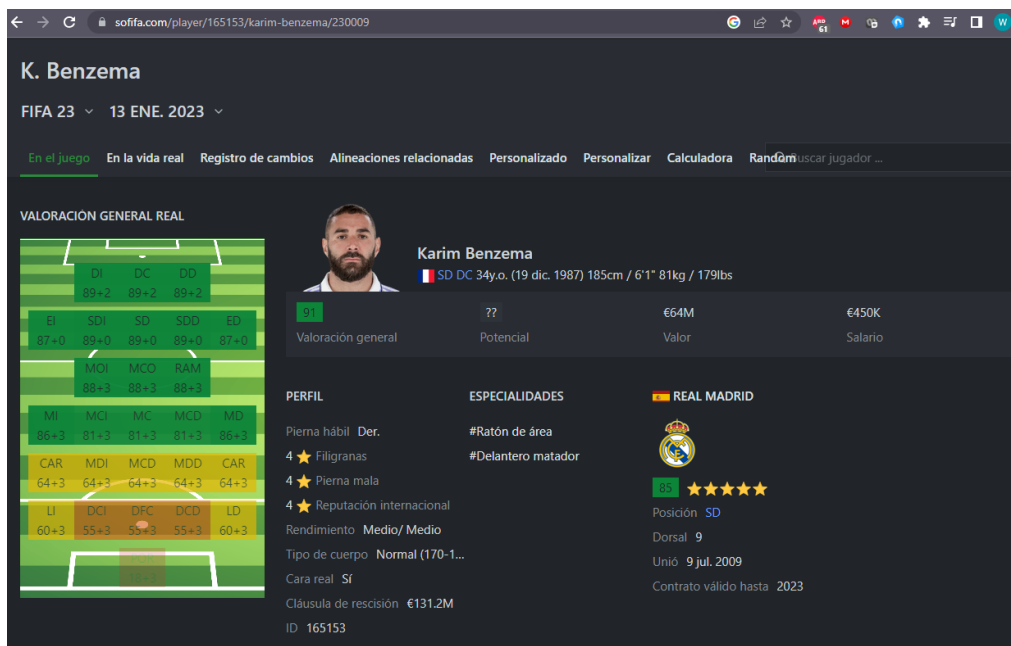


Figura 12. Información general del jugador Karim Benzema en el sitio SOFIFA.

Los datos extraídos de Kaggle cuentan con columnas relacionadas a las características de los jugadores con valores nulos, como por ejemplo la de `goalkeeping_speed`, `player_tags`, etc. Estas columnas y valores nulos serán analizados posteriormente en la sección del procesamiento de datos.

Por otra parte, FBREF.com es una fuente confiable de información ya que es una plataforma en línea dedicada a proporcionar estadísticas detalladas, relatos históricos y los últimos resultados del mundo del fútbol, abarcando más de 100 torneos tanto de equipos masculinos como femeninos, incluyendo clubes y selecciones nacionales como se puede observar en la Figura 13.

Wk	Day	Date	Time	Home	xG	Score	xG	Away	Attendance	Venue	Referee	Match Report	Notes
1	Fri	2022-08-05	20:00 (14:00)	Crystal Palace	1.2	0-2	1.0	Arsenal	25,286	Selhurst Park	Anthony Taylor	<a href="#">Match Report</a>	
	Sat	2022-08-06	12:30 (06:30)	Fulham	1.2	2-2	1.2	Liverpool	22,207	Craven Cottage	Andy Madley	<a href="#">Match Report</a>	
			15:00 (09:00)	Tottenham	1.5	4-1	0.5	Southampton	61,732	Tottenham Hotspur Stadium	Andre Marriner	<a href="#">Match Report</a>	
			15:00 (09:00)	Newcastle Utd	1.7	2-0	0.3	Nott'ham Forest	52,245	St James' Park	Simon Hooper	<a href="#">Match Report</a>	
			15:00 (09:00)	Leeds United	0.8	2-1	1.3	Wolves	36,347	Elland Road	Robert Jones	<a href="#">Match Report</a>	
			15:00 (09:00)	Bournemouth	0.6	2-0	0.7	Aston Villa	11,013	Vitality Stadium	Peter Bankes	<a href="#">Match Report</a>	
			17:30 (11:30)	Everton	0.7	0-1	1.5	Chelsea	39,254	Goodison Park	Craig Pawson	<a href="#">Match Report</a>	
	Sun	2022-08-07	14:00 (08:00)	Leicester City	0.6	2-2	0.8	Brentford	31,794	King Power Stadium	Jarred Gillett	<a href="#">Match Report</a>	
			14:00 (08:00)	Manchester Utd	1.4	1-2	1.5	Brighton	73,711	Old Trafford	Paul Tierney	<a href="#">Match Report</a>	
			16:30 (10:30)	West Ham	0.5	0-2	2.2	Manchester City	62,443	London Stadium	Michael Oliver	<a href="#">Match Report</a>	

Figura 13. Información de partidos jugados en la Premier League en el sitio FBREF.

De igual manera los datos extraídos de FBREF cuentan con columnas de información de partidos con valores nulos, como por ejemplo la columna time, notes, etc. Estos valores nulos se analizarán en la sección del procesamiento de datos.

## 2.4. Preparación de los datos

Durante esta etapa del proceso metodológico, los datos son preparados y ajustados para alinearse con las técnicas de ML que se aplicarán. Este procedimiento incluye elegir el segmento específico de datos relevantes, unir los datos de las características de los jugadores con los datos sobre los partidos jugados, limpiarlos para elevar su precisión y utilidad, y estructurarlos conforme a las especificaciones del sistema de modelado que se usará.

### 2.4.1. Limpieza de datos

Como se mencionó previamente para este proyecto se tiene inicialmente dos diferentes datasets. Uno con las características de los jugadores el cual cuenta con 110 columnas. En este dataset se tuvo que eliminar varios de los atributos, por medio del comando drop,

como los puntajes de posición (LM = 82+2, RW = 74+3) que son valores que ayudan a los jugadores del videojuego a armar su alineación, pero que para nuestro estudio son irrelevantes. De igual manera se eliminaron valores como el player\_id, nacionalidad, fecha de nacimiento, posición, etc, y solo se tomaron en cuenta jugadores que eran titulares y reemplazos, no se tomaron en cuenta los jugadores que son de reserva ya que por lo general su participación en partidos oficiales es muy baja.

```
df1= df1.drop(['ls','st','rs','lw','lf','cf','rf','rw','lam','cam'], axis=1)
```

Figura 14. Aplicación comando drop.

Después de la eliminación de los atributos mencionados anteriormente se tiene una fuente de información con 61 columnas que se muestran a continuación en la Figura 15.

```
#   Column              Non-Null Count  Dtype
-----
0   Unnamed: 0           1523208 non-null float64
1   fifa_update          1523208 non-null float64
2   fifa_update_date     1583874 non-null datetime64[ns]
3   short_name           1583874 non-null object
4   overall              1523208 non-null float64
5   potential            1523208 non-null float64
6   value_eur            1404417 non-null float64
7   wage_eur             1407236 non-null float64
8   age                  1583874 non-null float64
9   height_cm            1523208 non-null float64
10  weight_kg            1583874 non-null object
11  team                  1467479 non-null object
12  club_joined_date     1279416 non-null datetime64[ns]
13  club_contract_valid_until_year  1406813 non-null float64
14  preferred_foot       1523208 non-null object
15  weak_foot            1583874 non-null float64
16  skill_moves          1523208 non-null float64
17  international_reputation  1523208 non-null float64
18  work_rate            1523208 non-null object
19  body_type            1523208 non-null object
20  pace                 1422445 non-null float64
21  shooting             1422445 non-null float64
22  passing              1422445 non-null float64
23  dribbling            1422445 non-null float64
24  defending              1422445 non-null float64
25  physic               1422445 non-null float64
26  attacking_crossing   1583874 non-null float64
27  attacking_finishing  1583874 non-null float64
28  attacking_heading_accuracy  1583874 non-null float64
29  attacking_short_passing  1583874 non-null float64
30  attacking_volleys    1583874 non-null float64
31  skill_dribbling      1583874 non-null float64
32  skill_curve          1583874 non-null float64
33  skill_fk_accuracy    1583874 non-null float64
34  skill_long_passing  1583874 non-null float64
35  skill_ball_control   1583874 non-null float64
36  movement_acceleration  1583874 non-null float64
37  movement_sprint_speed  1583874 non-null float64
38  movement_agility     1583874 non-null float64
39  movement_reactions   1583874 non-null float64
40  movement_balance     1583874 non-null float64
41  power_shot_power     1583874 non-null float64
42  power_jumping        1583874 non-null float64
43  power_stamina        1583874 non-null float64
44  power_strength       1583874 non-null float64
45  power_long_shots     1583874 non-null float64
46  mentality_aggression  1583874 non-null float64
47  mentality_interceptions  1583874 non-null float64
48  mentality_positioning  1583874 non-null float64
49  mentality_vision     1583874 non-null float64
50  mentality_penalties  1583874 non-null float64
51  mentality_composure  1176081 non-null float64
52  defending_marking_awareness  1523208 non-null float64
53  defending_standing_tackle  1583874 non-null float64
54  defending_sliding_tackle  1583874 non-null float64
55  goalkeeping_diving   1583874 non-null float64
56  goalkeeping_handling  1583874 non-null float64
57  goalkeeping_kicking  1523208 non-null float64
58  goalkeeping_positioning  1583874 non-null float64
59  goalkeeping_reflexes  1583874 non-null float64
60  year                  1583874 non-null float64
61  month                 1583874 non-null float64
dtypes: datetime64[ns](2), float64(54), object(6)
memory usage: 761.3+ MB
```

Figura 15. Variable finales del dataset de jugadores.

Esta información será nuevamente modificada una vez que se combine con la segunda fuente de datos y se analice la correlación y la cantidad de valores nulos que contiene cada atributo.

El segundo dataset tiene inicialmente 26 columnas, de las cuales se eliminó columnas como la hora del partido, el día, el nombre de la liga, el nombre del referee, etc, y se modificó el nombre de los equipos para que coincidan con el mismo nombre que tienen en el dataset de los atributos de los jugadores, como se puede observar en el ejemplo mostrado en la Figura 16, en el cual se modificó el nombre “Manchester Utd” por el nombre “Manchester United”, haciendo uso del comando replace.

```
df2 = df2.replace({"Manchester Utd": "Manchester United"})
```

Figura 16. Aplicación comando replace.

Debido a la eliminación de varias columnas, el dataset de partidos jugados se reduce a tener solo 17 columnas que se muestran en la Figura 17.

```
Data columns (total 18 columns):
#   Column      Non-Null Count  Dtype
---  -
0   date         8140 non-null   datetime64[ns]
1   venue        8140 non-null   object
2   result       8140 non-null   object
3   opponent     8140 non-null   object
4   team         8140 non-null   object
5   xg           4340 non-null   float64
6   xga          4340 non-null   float64
7   poss         6620 non-null   float64
8   attendance   5740 non-null   float64
9   formation    8140 non-null   object
10  sh           6620 non-null   float64
11  sot          6620 non-null   float64
12  dist         4337 non-null   float64
13  pk           8140 non-null   float64
14  pkatt        6620 non-null   float64
15  season       8140 non-null   int64
16  year         8140 non-null   int64
17  month        8140 non-null   int64
dtypes: datetime64[ns](1), float64(9), int64(3), object(5)
memory usage: 1.2+ MB
```

Figura 17. Variable finales del dataset de partidos jugados.

## 2.4.2. Preprocesamiento de Datos

Para poder aplicar los algoritmos de ML es necesario combinar la información presente en los dos datasets, para esto se utilizó el comando pd.merge tomando como atributos que deben ser el mismo en ambos datasets el año y el nombre del equipo como se muestra en la Figura 18.

```
joined_df = pd.merge(df1, df2, on=["year", "team"])
```

Figura 18. Aplicación comando merge.

Posteriormente se eliminan los valores duplicados que se crean debido a que en un mismo año puede haber varias diferentes fechas de actualización de fifa (fifa\_update\_date) para un mismo jugador. Esto se logra por medio del comando drop\_duplicates, mostrado en la Figura 19, tomando en cuenta los valores de date (fecha del partido) y short\_name (nombre del jugador).

```
df = joined_df.drop_duplicates(['date', 'short_name'], keep='last')
```

Figura 19. Aplicación comando drop\_duplicates.

Se debe eliminar del dataset con la información de jugadores y los partidos jugados valores que ya no son necesarios para el análisis como lo es fifa\_update\_date, year, month, club\_join\_date, etc. Y ciertas variables se pueden agrupar en un solo atributo por medio de su promedio como se especifica en la Tabla 4, tomando en cuenta información de SOFIFA y haciendo uso del comando mean(), mostrado en la Figura 20.

```
df['mentality_aggression_skills'] = df[['mentality_aggression', 'mentality_interceptions']].mean(axis=1)  
df=df.drop(['mentality_aggression', 'mentality_interceptions'], axis=1)
```

Figura 20. Aplicación comando mean().

Luego de transformar ciertas variables categóricas en valores dummies para que puedan ser analizados posteriormente como las variables formation, work\_rate, body\_type por medio del comando get\_dummies (Figura 21), se realiza una validación de datos nulos (Figura 22) y se los reemplaza por valores promedios de la categoría con el comando fillna(df.mean()) como se muestra en la Figura 23.

Tabla 4. Agrupación de atributos de jugadores en una sola variable.

<b>Variable Final</b>	<b>Atributos que contiene</b>
attacking_skills	attacking_finishing attacking_crossing attacking_heading_accuracy attacking_short_passing attacking_volleys
ball_skill	skill_dribbling skill_curve skill_fk_accuracy skill_long_passing skill_ball_control
movement_speed_skills	movement_acceleration movement_sprint_speed movement_agility movement_reactions movement_balance
power_skills	power_shot_power power_jumping power_stamina power_strength power_long_shots
mentality_aggression_skills	mentality_aggression mentality_interceptions
mentality_vision_skills	mentality_positioning mentality_vision mentality_penalties
defending_skills	defending_marking_awareness defending_standing_tackle defending_sliding_tackle
goalkeeping_skills	goalkeeping_diving goalkeeping_handling goalkeeping_kicking goalkeeping_positioning goalkeeping_reflexes



```
dummies = pd.get_dummies(df['body_type'], drop_first = True)
dummies.head()
```

	Lean (170-185)	Lean (185+)	Normal (170-)	Normal (170-185)	Normal (185+)	Stocky (170-)	Stocky (170-185)	Stocky (185+)	Unique
0	0	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0	0	1
2	0	0	0	0	0	0	0	0	1
3	0	0	0	0	0	0	0	0	1
4	0	0	0	0	0	0	0	0	1

Figura 21. Aplicación comando get\_dummies.

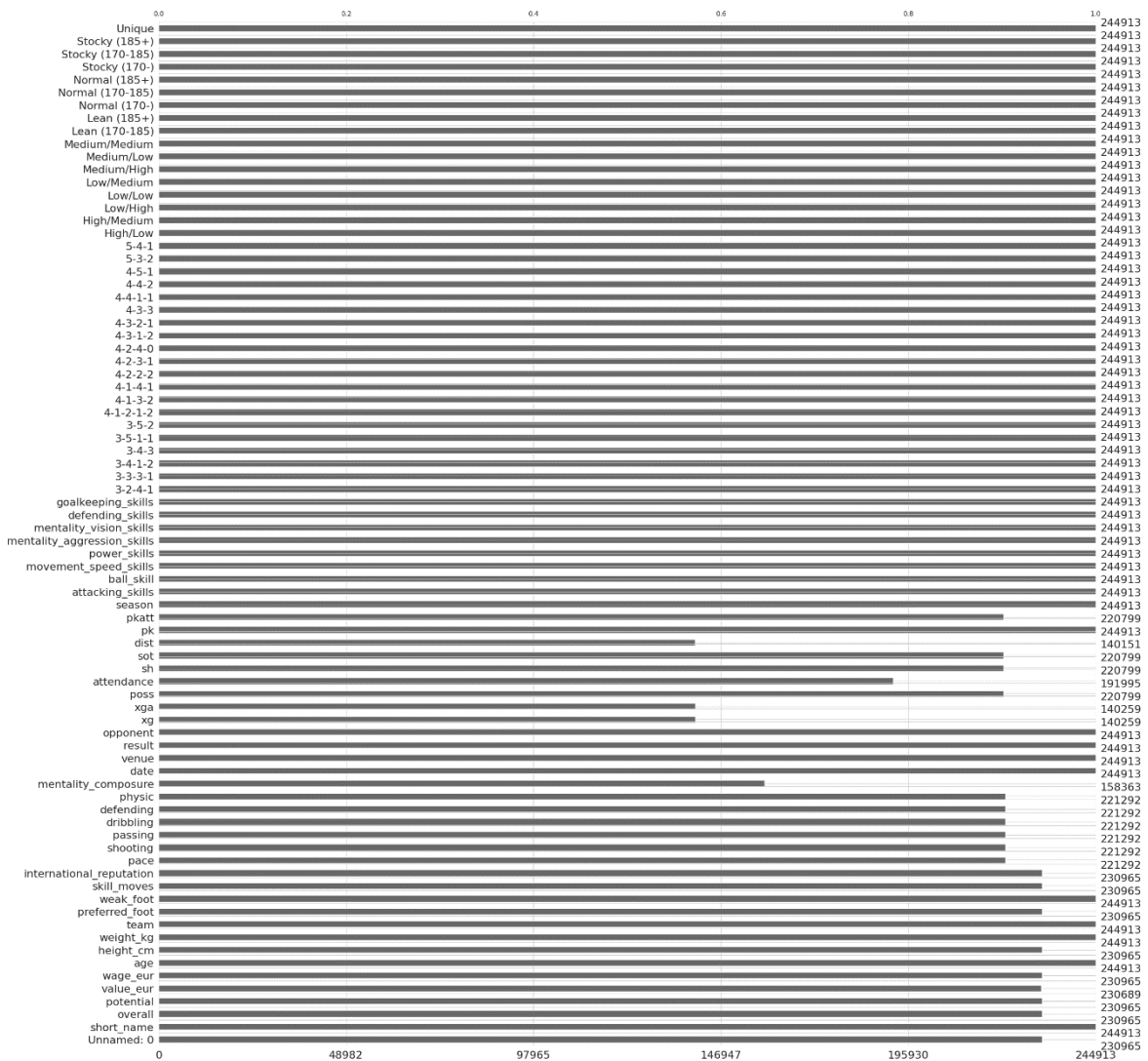


Figura 22. Variables con Valores Nulos.

```
df = df.fillna(df.mean())
```

Figura 23. Aplicación comando fillna.

Toda la información que se tiene hasta este punto sobre los jugadores y los resultados de los partidos es necesaria tenerla en valores promedios, para esto se agrupan los valores en base a sus valores promedios por medio del comando `groupby().mean()`, tomando en cuenta los valores de los cuales no se desea el valor promedio como por ejemplo el nombre del equipo, la temporada, el resultado, sede, el oponente y la fecha del partido, como se observa en la Figura 24.

```
d_grouped=df.groupby(['team', 'season', 'result','venue','opponent','date']).mean()
```

Figura 24. Aplicación comando `groupby` en base al promedio de las variables.

De esta manera el dataset tiene una estructura que se puede apreciar en la Figura 25.

Sin embargo, este dataset está incompleto (Figura 25) ya que solo cuenta con el valor promedio de las características deportivas de un solo equipo, faltaría completarlo con la información del equipo rival y a su vez cambiar los valores categóricos faltantes como sede (`venue`) y resultado (`result`) por valores numéricos. Usando el comando `merge` (Figura 26) y etiquetando la información para el primer equipo con el sufijo `_t` y la información para el segundo equipo con el sufijo `_o`, se obtiene un dataset que puede ser utilizado en el desarrollo de este proyecto (Figura 27).

	team	season	result	venue	opponent	date	overall	potential	value_eur	wage_eur	...	Medium/Medium	Lean (170-185)	Lean (185+)	Normal (170-)	Normal (170-185)	Normal (185+)	Stocky (170-)	Stocky (170-185)	Stocky (185+)	Unique
0	Arsenal	2012	D	Away	Aston Villa	2012-11-24	74.259944	78.688390	1.031533e+07	54528.491546	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.000000	0.000000
1	Arsenal	2012	D	Away	Everton	2012-11-28	74.259944	78.688390	1.031533e+07	54528.491546	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.000000	0.000000
2	Arsenal	2012	D	Away	Manchester City	2012-09-23	74.259944	78.688390	1.031533e+07	54528.491546	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.000000	0.000000
3	Arsenal	2012	D	Away	Southampton	2013-01-01	74.259944	78.688390	1.031533e+07	54528.491546	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.000000	0.000000
4	Arsenal	2012	D	Away	Stoke City	2012-08-26	74.259944	78.688390	1.031533e+07	54528.491546	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.000000	0.000000
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
8135	Wolverhampton Wanderers	2022	W	Home	Liverpool	2023-02-04	75.173913	80.000000	1.490109e+07	58826.086957	...	0.391304	0.217391	0.000000	0.130435	0.217391	0.304348	0.0	0.0	0.000000	0.130435
8136	Wolverhampton Wanderers	2022	W	Home	Nottingham Forest	2022-10-15	72.588235	79.352941	1.215294e+07	49411.764706	...	0.352941	0.205882	0.117647	0.088235	0.235294	0.205882	0.0	0.0	0.029412	0.088235
8137	Wolverhampton Wanderers	2022	W	Home	Southampton	2022-09-03	72.588235	79.352941	1.215294e+07	49411.764706	...	0.352941	0.205882	0.117647	0.088235	0.235294	0.205882	0.0	0.0	0.029412	0.088235
8138	Wolverhampton Wanderers	2022	W	Home	Tottenham Hotspur	2023-03-04	75.173913	80.000000	1.490109e+07	58826.086957	...	0.391304	0.217391	0.000000	0.130435	0.217391	0.304348	0.0	0.0	0.000000	0.130435
8139	Wolverhampton Wanderers	2022	W	Home	West Ham United	2023-01-14	75.173913	80.000000	1.490109e+07	58826.086957	...	0.391304	0.217391	0.000000	0.130435	0.217391	0.304348	0.0	0.0	0.000000	0.130435

8140 rows x 78 columns

Figura 25. Dataset con valores promedios de un solo equipo y los resultados de los partidos.

```
d_t=pd.merge(d_compl, d_compl_2, on=["date","team"],suffixes=("_t", "_o"))
```

Figura 26. Aplicación comando merge con sufijos para diferenciar los equipos.

	result_t	overall_t	potential_t	value_eur_t	wage_eur_t	age_t	height_cm_t	weight_kg_t	preferred_foot_t	weak_foot_t	...	Lean (170-185)_o	Lean (185+)_o	Normal (170-)_o	Normal (170-185)_o	Normal (185+)_o	Stocky (170-)_o	Stocky (170-185)_o	Stocky (185+)_o	Unique_o	Home
0	0	74.259944	78.688390	1.031533e+07	54528.491546	22.083333	182.588505	75.750000	0.238595	3.083333	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	0
1	0	74.259944	78.688390	1.031533e+07	54528.491546	22.083333	182.588505	75.750000	0.238595	3.083333	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	0
2	0	74.259944	78.688390	1.031533e+07	54528.491546	22.083333	182.588505	75.750000	0.238595	3.083333	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	0
3	0	74.259944	78.688390	1.031533e+07	54528.491546	23.666667	182.588505	69.666667	0.238595	3.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	0
4	0	74.259944	78.688390	1.031533e+07	54528.491546	22.083333	182.588505	75.750000	0.238595	3.083333	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
8135	1	75.173913	80.000000	1.490109e+07	58826.086957	24.956522	181.695652	75.304348	0.217391	3.304348	...	0.086957	0.217391	0.000000	0.173913	0.000000	0.0	0.043478	0.043478	0.391304	1
8136	1	72.588235	79.352941	1.215294e+07	49411.764706	24.176471	182.352941	75.382353	0.264706	3.264706	...	0.255319	0.127660	0.000000	0.255319	0.276596	0.0	0.042553	0.042553	0.000000	1
8137	1	72.588235	79.352941	1.215294e+07	49411.764706	24.176471	182.352941	75.382353	0.264706	3.264706	...	0.303030	0.121212	0.000000	0.303030	0.181818	0.0	0.030303	0.030303	0.000000	1
8138	1	75.173913	80.000000	1.490109e+07	58826.086957	24.956522	181.695652	75.304348	0.217391	3.304348	...	0.217391	0.130435	0.000000	0.217391	0.130435	0.0	0.000000	0.043478	0.260870	1
8139	1	75.173913	80.000000	1.490109e+07	58826.086957	24.956522	181.695652	75.304348	0.217391	3.304348	...	0.181818	0.136364	0.045455	0.272727	0.272727	0.0	0.045455	0.000000	0.045455	1

8140 rows x 144 columns

Figura 27. Dataset con valores promedios de los dos equipos y los resultados de los partidos.



De la gráfica anterior se observa que existen varios valores que pueden ser eliminados ya que no existe una correlación importante con las otras variables, entre estos valores están los relacionados a las formaciones (4-4-2, 3-3-3-1, etc) durante los partidos. También se eliminan variables que tienen una correlación igual a 1 con otras variables como es el caso de la variable pkatt (intentos de penaltis ejecutados) que tiene una correlación total con pk (penaltis ejecutados) y por ende se puede tomar en cuenta solo a una de estas variables.

La Figura 30 y la Figura 31 ilustran los histogramas al conjunto de datos, proporcionando una representación visual de la distribución de sus atributos. Al examinar la forma de la distribución de las diferentes variables, se observa que no hay una tendencia uniforme entre ellos. Por ejemplo, la variable relacionada con las habilidades de ataque tiene una distribución aproximadamente simétrica, sugiriendo una distribución normal. En contraste, otros atributos como las habilidades de movimiento y velocidad o edad exhiben gráfica con una asimetría negativa, o el nivel de reputación internacional de los jugadores que tienen asimetría positiva.

Para realizar un análisis preliminar de la variabilidad presente en los datos, resulta esencial observar detenidamente cada uno de los histogramas. Este análisis permite identificar las características particulares de la distribución de cada atributo, como la simetría, el sesgo y la dispersión, ofreciendo una perspectiva inicial sobre la naturaleza de los datos recopilados y su posible influencia en posteriores análisis estadísticos o modelados predictivos.

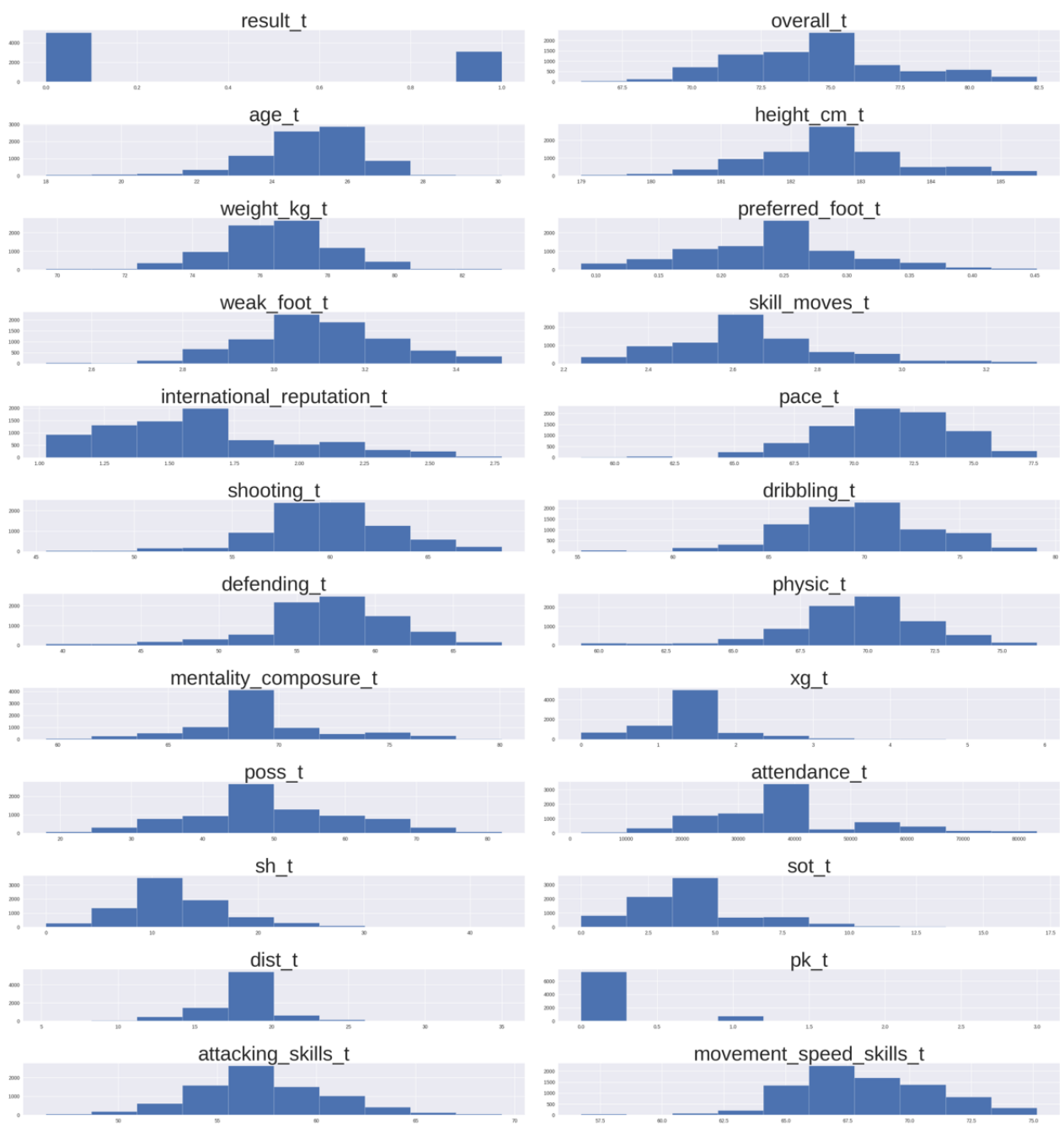


Figura 30. Histogramas Parte 1.

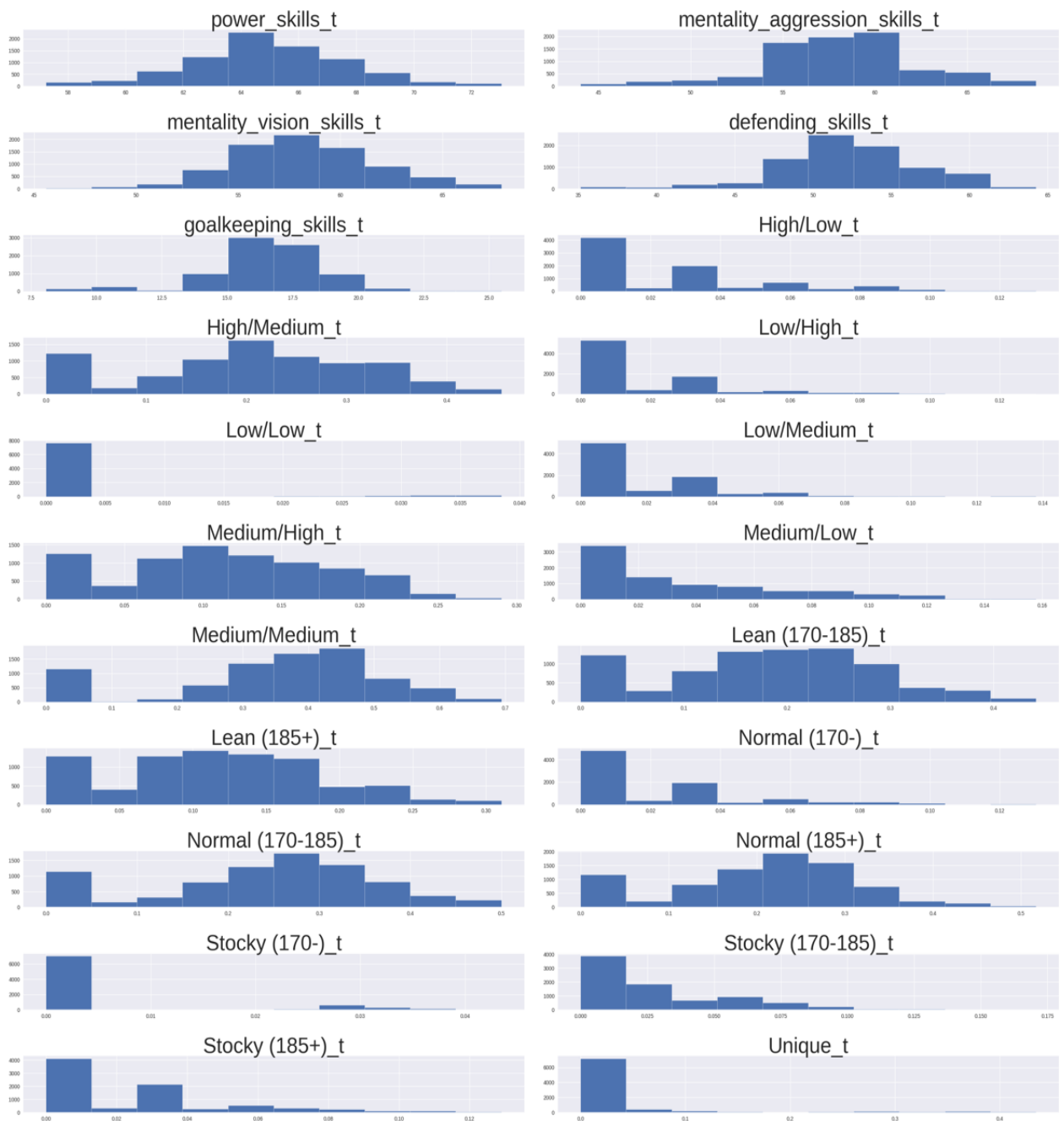


Figura 31. Histogramas Parte 2.

## Selección de Variables por medio del análisis de varianza

El análisis de varianza, conocido como ANOVA, es una técnica estadística usada para determinar si existen diferencias significativas entre los valores de distintas categorías de entrada para una variable de salida específica. Esta metodología es crucial para examinar la relación entre variables dependientes e independientes, permitiendo el análisis de

múltiples grupos de datos. Su objetivo es identificar variaciones tanto entre como dentro de las muestras, proporcionando información valiosa sobre las interacciones entre estas variables.

La función `SelectKBest`, una herramienta de la librería `Scikit-learn`, se caracteriza por su capacidad de seleccionar las características más relevantes de un conjunto de datos mediante la asignación de puntuaciones univariadas. Esta selección se basa en un test específico que se pasa como parámetro a la función. En el escenario descrito, el test elegido es el `f_classif`, una elección motivada por el hecho de que la variable objetivo es una clasificación relacionada con el resultado de los partidos. El test `f_classif` es especialmente útil en contextos donde la variable de salida es una clasificación.

Por otro lado, en situaciones donde la salida es una regresión lineal, se prefiere el uso del test `f_regression`. Ambos métodos, `f_classif` y `f_regression`, tienen el objetivo común de estimar el grado de dependencia lineal entre dos variables aleatorias. Esto se logra al asignar un puntaje específico a cada característica, evaluando así su importancia o relevancia en relación con la variable objetivo. Estos puntajes son fundamentales para entender la influencia de cada característica en la modelización de los datos y ayudan en la toma de decisiones sobre qué variables incluir en el análisis final.

Como se puede ver la Figura 32, existen varias variables que tienen puntuaciones muy bajas en relación con otras como es el caso de la variable `preferred_foot` (pie preferido del jugador), `goalkeeping_skills` (habilidades de portería) o incluso características relacionadas con el tipo de cuerpo de los jugadores. Estas variables son eliminadas del dataset para tener obtener un mejor resultado a la hora de aplicar los algoritmos de ML.



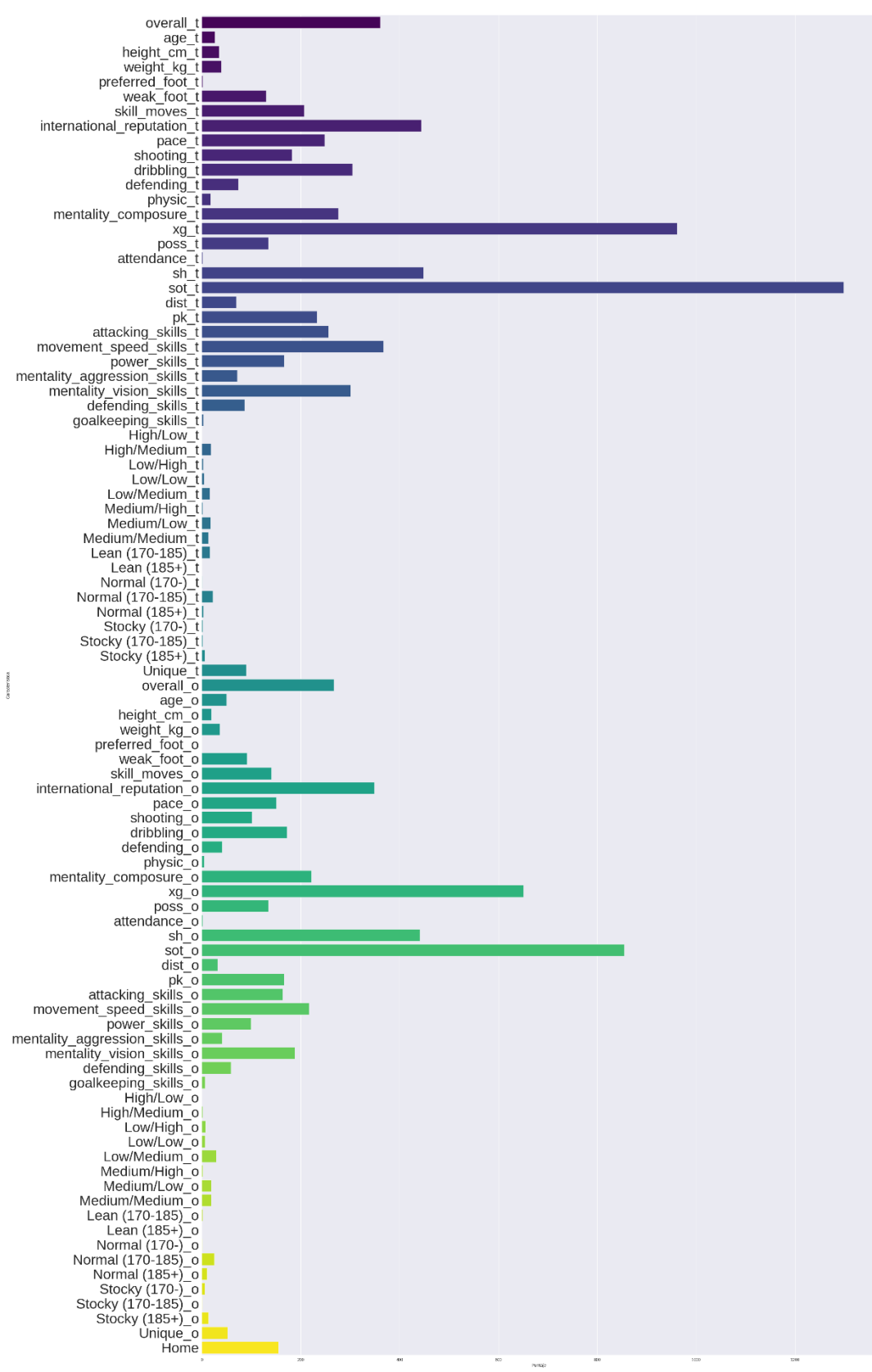


Figura 32. Características del histograma usando el algoritmo SelectKBest.

## Presentación Categórica Final de los Datos

Como se muestra en la Figura 9, las categorías iniciales de los resultados de los partidos estaban divididos en ganar, empatar y perder. Pero para tener un mejor desempeño de los algoritmos de ML que se aplicarán se va a tomar solamente dos categorías GANAR (1) y NO GANAR (0), los cuales están clasificados de la manera mostrada en la Figura 33.

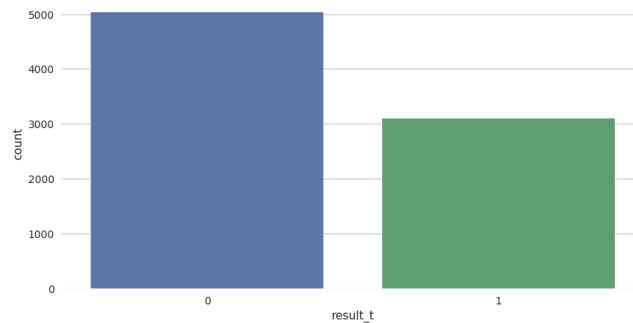


Figura 33. Muestra categórica de la clasificación de los resultados del dataset final.

Como se puede observar en la Figura anterior los datos de No Ganar (5038) son mayores que los de Ganar (3102), esto implica que los datos no están balanceados.

Hay diversas estrategias para alcanzar un equilibrio en las categorías dentro de un conjunto de datos. Sin embargo, para los propósitos de este análisis, donde es crucial mantener la fidelidad de los datos para interpretar y explicar las decisiones tomadas por algoritmos de inteligencia artificial, se ha optado por no aplicar técnicas de balanceo.

Mantener los datos en su forma original es vital para evaluar cómo las características reales influyen en el proceso de toma de decisiones de la IA. Aunque la generación de datos sintéticos puede ser beneficiosa para aumentar la precisión de los modelos de ML, podría comprometer la transparencia y la comprensibilidad, que son esenciales para la XAI.

## Estandarización de Características

Para asegurar la integridad de los modelos de ML, es crucial estandarizar las variables involucradas. Las diferencias significativas en la varianza de las características pueden impedir una comparación efectiva entre ellas y potencialmente sesgar el aprendizaje del modelo. La estandarización permite que cada característica contribuya equitativamente al

modelo, alineando la varianza y centrando los datos en torno a cero. Esto se logra mediante la función StandardScaler de la biblioteca Scikit-learn, que ajusta cada característica para que tenga una media de cero y una varianza de uno.

En la Figura 34, se expone cómo la distribución de las variables de entrada varía notablemente en cuanto a dispersión. Sin embargo, tras la aplicación de StandardScaler, se observa una distribución más homogénea y una reducción de las desviaciones extremas que pueden ser perjudiciales durante el entrenamiento del modelo, como se observa en la Figura 35. Es crucial señalar que la normalización de los datos se llevará a cabo solo en aquellos algoritmos de inteligencia artificial que lo necesiten, subrayando la adaptabilidad de este proceso a las necesidades específicas de cada modelo.

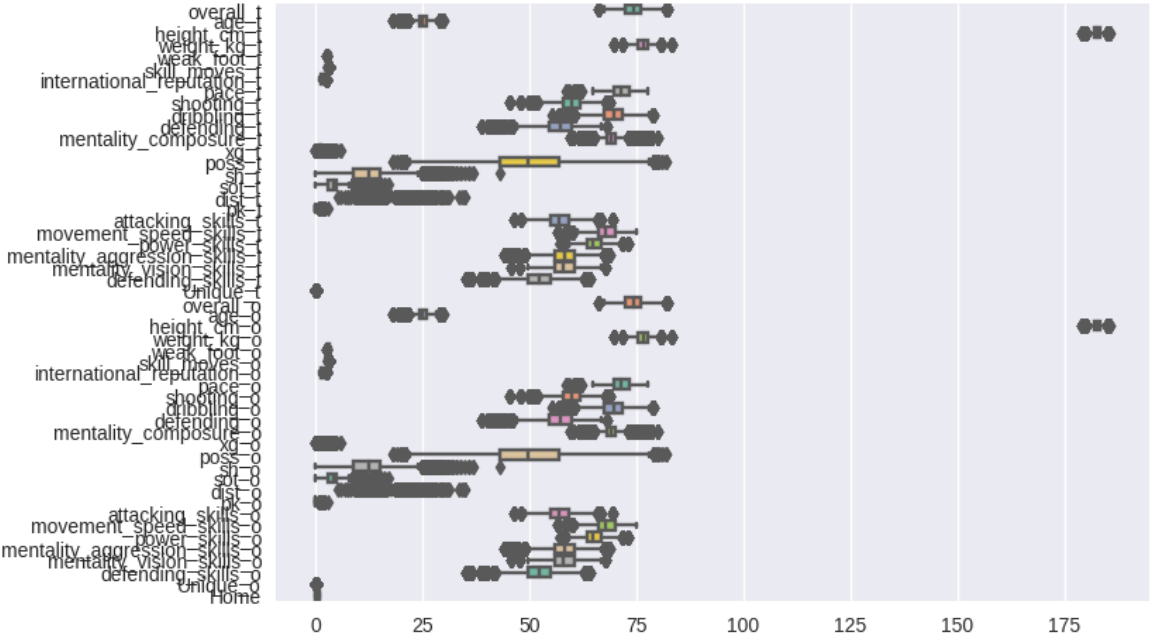


Figura 34. Bloxpot de la varianza en la distribución de datos de cada atributo.



### **2.5.1. Selección de técnicas de AI**

La selección de métodos adecuados para el análisis de datos es crucial y requiere considerar varios aspectos que influirán en la elección de los algoritmos de ML más apropiados para el problema en cuestión. Dada la estructura de los datos, que incluye variables de entrada y salida definidas, se optó por algoritmos que funcionan bajo un paradigma de aprendizaje supervisado.

Un criterio adicional a considerar es la elección de la metodología de XAI que se utilizará. Debido a que SHAP y LIME son técnicas "post-hoc" que se especializan en interpretar modelos complejos después de que han hecho una predicción, se priorizarán los algoritmos considerados como "cajas negras", aquellos cuyo proceso de toma de decisiones no es transparente o directamente interpretable.

En este contexto, se han seleccionado los siguientes algoritmos de ML para ser implementados en este estudio, basados en su capacidad para ser analizados mediante LIME y su adecuación al tipo de datos disponibles:

- SVM, que son efectivas en espacios de alta dimensión.
- RF, que son buenos para manejar un gran número de características y proporcionan estimaciones de la importancia de las variables.
- ANN, que son poderosas para capturar relaciones complejas y no lineales en los datos.

### **2.5.2. Plan de Pruebas para Evaluación de Algoritmos de Machine Learning**

Se realizaron pruebas en condiciones uniformes para evaluar y comparar objetivamente el rendimiento de los diferentes modelos de ML. Se debe considerar el desbalance en los datos durante la evaluación de modelos. Esto implica que ciertas métricas estándar, como la "accuracy", pueden no ser adecuadas debido a su potencial para proporcionar resultados no tan precisos en tales condiciones.

Para abordar los problemas relacionados con datos desbalanceados, se recomienda el uso de matrices de confusión y la métrica F1-Score. Las matrices de confusión ayudan a visualizar el rendimiento de los modelos mostrando el número de predicciones correctas e incorrectas. Esto permite evaluar la capacidad predictiva real del modelo, incluyendo la identificación de errores críticos como falsos positivos y falsos negativos.

La métrica F1-Score, que es un promedio armónico de Recall (sensibilidad) y Precision (exactitud), se propone como una herramienta eficaz para comparar modelos. Esta métrica es especialmente útil en situaciones con datos desbalanceados, ya que tiene en cuenta tanto la capacidad del modelo para identificar correctamente las clases positivas como su precisión al hacerlo.

### **2.5.3. Construcción de Modelos de Machine Learning**

Para la elección de los algoritmos ML, se han descartado aquellos que son intrínsecamente interpretables, como los clasificadores lineales o modelos basados en árboles de decisión. Esta exclusión obedece a la necesidad de utilizar modelos con un enfoque Post-Hoc en XAI que se aplica a modelos de ML considerados como "cajas negras". Basándose en la tendencia de estudios anteriores [15],[16], se han seleccionado algoritmos de ML supervisados que puedan ser aplicados a los algoritmos de XAI como SHAP y LIME.

El primer algoritmo elegido es el SVM el cual es un conjunto de métodos de aprendizaje supervisado utilizados para clasificación, regresión y detección de valores atípicos. Es conocido por su eficacia en el manejo de datos multiclase.

El segundo algoritmo es el RF, el cual se basa en árboles de decisión y ajusta múltiples clasificadores de árboles, submuestreando el conjunto de datos y promediando los resultados, mejorando así la predicción y evitando el sobreajuste.

El último algoritmo seleccionado son las ANN, indispensables en el campo de la AI. Estas forman parte del aprendizaje supervisado y son altamente valoradas en la actualidad, especialmente por su habilidad para procesar grandes volúmenes de datos no estructurados y detectar patrones.

En nuestro estudio de caso, analizamos 51 atributos con una salida de clasificación multiclase representando el resultado de un partido de fútbol profesional (Ganar, No Ganar). A través de diferentes capas neuronales, se realiza el reconocimiento de patrones y la decisión de clasificación. Para una mejor comprensión del proceso de las ANN en nuestro estudio, se puede observar la Figura 36.

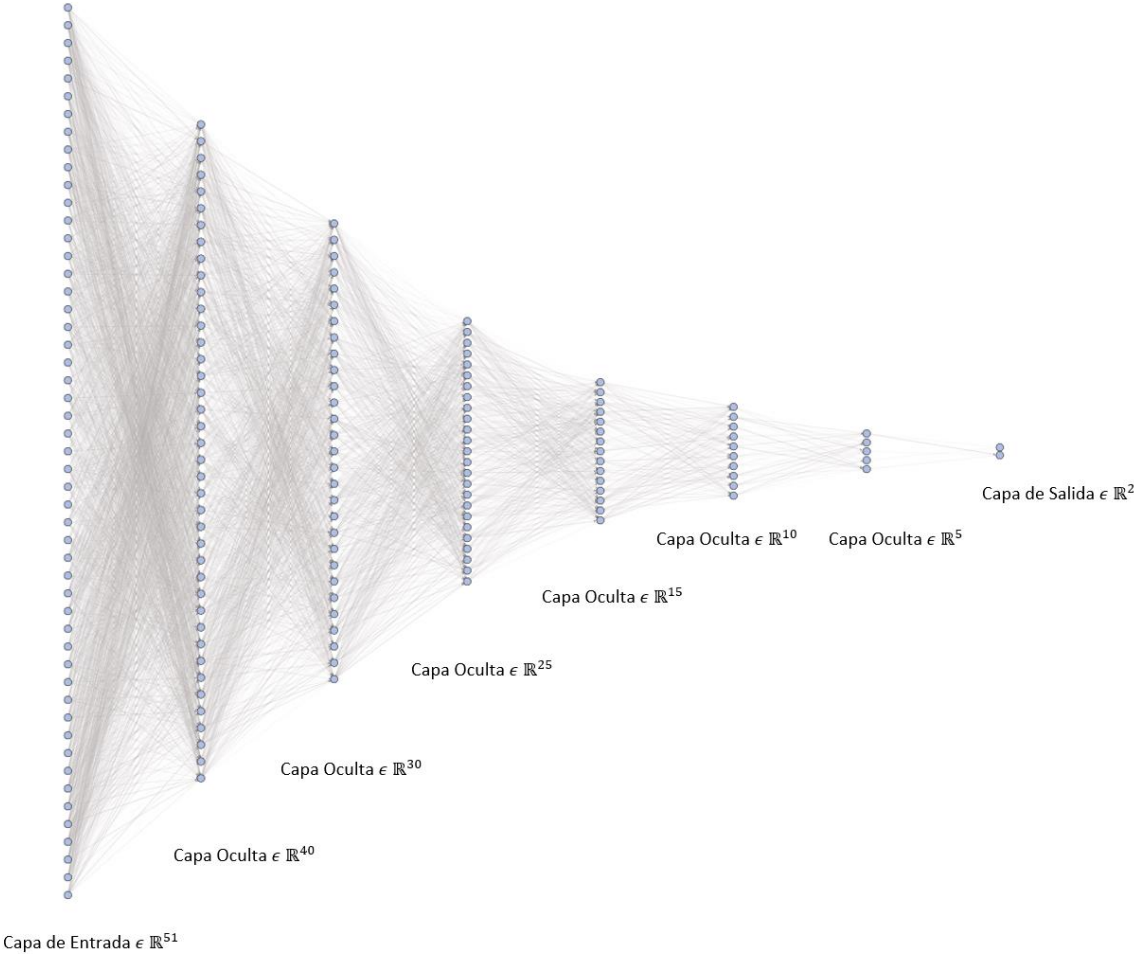


Figura 36. Red Neuronal Artificial para evaluar el resultado de partidos de fútbol.

En el diseño de las ANN para tareas de clasificación, es común emplear diferentes funciones de activación en distintas capas para optimizar el rendimiento del modelo. Para las capas de entrada e intermedias, se utiliza la función de activación lineal rectificada, conocida como ReLU. Esta función transforma los datos entrantes de manera que se

vuelven linealmente separables, lo cual es beneficioso para la clasificación ya que facilita la diferenciación entre las categorías.

La capa de salida, por otro lado, a menudo se utilizan funciones sigmoideas o “softmax”. En este caso se utilizó la función “softmax” la cual es particularmente útil en los modelos de clasificación de múltiples clases donde las categorías son mutuamente excluyentes, como es el caso del resultado de una competencia deportiva (GANAR o NO GANAR). La función softmax es adecuada para la capa de salida porque convierte las salidas del modelo en probabilidades que suman 1, permitiendo que se asigne claramente cada instancia a la clase con la mayor probabilidad.

Es importante evitar el uso de softmax en las capas intermedias debido a problemas potenciales como el desvanecimiento del gradiente, un fenómeno donde los gradientes se hacen tan pequeños que el modelo deja de aprender durante el entrenamiento. Este tipo de consideraciones en la selección de funciones de activación son cruciales para el desarrollo de modelos de ANN robustos y efectivos.

#### 2.5.4. Evaluación de Modelos

En esta etapa se debe valorar los diversos algoritmos de aprendizaje automático desplegados para discernir el más eficiente. Para ello, se recurre al esquema de pruebas previamente establecido. Aquí, se aplican indicadores claves como Precision, Recall y F1-Score para establecer un análisis comparativo del desempeño de los distintos algoritmos.

Tabla 5. Resultados del rendimiento de los algoritmos de aprendizaje automático para las clases Ganar (G) y No Ganar (NG).

	Precision		Recall		F1-Score	
	G	NG	G	NG	G	NG
<b>SVM</b>	0,74	0,78	0,60	0,87	0,66	0,83
<b>RF</b>	0,69	0,75	0,54	0,85	0,60	0,80
<b>ANN</b>	0,78	0,71	0,85	0,61	0,82	0,66

La Tabla 5 detalla los rendimientos de los algoritmos evaluados, aplicando las métricas de desempeño previamente mencionadas para cada una de las categorías de clasificación posibles. Destacan los clasificadores basados en los modelos SVM y ANN, se podrían



considerar cualquiera de los dos algoritmos de ML para la siguiente fase de este trabajo, sin embargo, se eligió el SVM ya que sus valores de Recall y F1-Scores son ligeramente superiores a los de la técnica de ANN y su implementación requiere menos costo computacional.

Las decisiones tomadas por este modelo serán posteriormente analizadas utilizando un algoritmo de XAI, que será implementado para esclarecer su funcionamiento.

### 2.5.5. Implementación del algoritmo de XAI

Para interpretar el desempeño del modelo SVM se emplearán los algoritmos LIME (Local Interpretable Model-agnostic Explanations) y SHAP (SHapley Additive exPlanations).

La implementación práctica del primer algoritmo LIME se facilita mediante el uso del módulo lime\_tabular, que proporciona funcionalidades de explicación para modelos que operan con datos en formato tabular.

Tabla 6. Parámetros de la clase "LimeTabularExplainer"

<b>Parámetro</b>	<b>Valor</b>	<b>Descripción</b>
training_data	np.array(X_train)	La función recibe como parámetro los datos de entrenamiento en forma de array.
feature_names	X_train.columns.values.tolist()	Se registran los nombres de las columnas de los datos de entrenamiento.
class_names	No Ganar, Ganar	Se asigna un nombre a las clases de salida.
verbose	True	Imprime los valores de predicción locales del modelo de ajuste lineal.
mode	classification	Se indica que es un caso de clasificación (no de regresión).

Dentro de este módulo, se hace uso de la clase LimeTabularExplainer. Esta clase se encarga de crear una instancia del explicador configurando varios parámetros para adaptar el proceso de explicación a las necesidades específicas del conjunto de datos y del modelo en uso. La configuración del LimeTabularExplainer puede incluir parámetros como el conjunto de datos de entrenamiento, discretización de características continuas, y parámetros para definir cómo se generan los datos de muestra alrededor de una instancia

para la localización de la explicación. La implementación concreta de estos parámetros se ajustará en función de las características y requisitos del conjunto de datos del proyecto.

La configuración detallada previamente sienta las bases para realizar interpretaciones locales utilizando LIME. No obstante, cabe señalar que aún no se ha especificado cuál instancia concreta del conjunto de datos se desea analizar, ni se ha escogido el modelo de aprendizaje automático cuyas predicciones serán objeto de estudio.

El proceso continúa con la elección aleatoria de una muestra del conjunto de datos de prueba, que se convertirá en un arreglo para su análisis. Se procede entonces a invocar el método `explain_instance` del objeto `LimeTabularExplainer` ya instanciado, lo que facilita la generación de explicaciones detalladas sobre la clasificación efectuada en la muestra seleccionada. Los argumentos que se pasan a `explain_instance` definen aspectos cruciales de cómo se llevará a cabo la explicación y se explicarán en detalle en la Tabla 7.

Tabla 7. Parámetros de la función "explain\_instance" para la generación de explicaciones de una instancia determinada.

<b>Parámetro</b>	<b>Valor</b>	<b>Descripción</b>
<code>data_row</code>	<code>instance</code>	Instancia de entrada que será interpretada.
<code>predict_fn</code>	<code>svm.predict_proba</code>	Función de predicción del modelo de aprendizaje automático cuyas decisiones serán evaluadas, SVM en este caso.
<code>num_features</code>	30	Número de características más influyentes que aparecerán en la explicación.
<code>labels</code>	[0,1]	Clases de salida que aparecerán en la explicación.

En el caso del algoritmo de SHAP se hace uso de la clase `shap.Explainer` (Tabla 8) que se utiliza para crear el objeto explicador. Este objeto se utiliza para calcular valores SHAP, que son medidas que ayudan a entender cómo cada característica en los datos contribuye a la predicción del modelo.

El proceso para la explicación de SHAP continúa con la interpretación de una instancia al azar (Tabla 9) para cada una de las categorías en base a los datos de test, `X.test.iloc[[1500]]` para la categoría Ganar y `X.test.iloc[[600]]` para la categoría No Ganar.

Tabla 8. Parámetros de la función "shap.Explainer" para la generación de explicaciones.

Parámetro	Valor	Descripción
predict	svm.predict	Representa la función de predicción del modelo. En este caso, es un clasificador SVM.
training_data	scaler.transform(X_train)	Este parámetro representa los datos de entrenamiento transformados que se utilizan en el explainer.
feature_perturbation	interventional	Indica el método de perturbación de las características utilizado para calcular los valores SHAP. En el modo "interventional", se manipulan las características de manera independiente.
data	scaler.transform(X_train)	Proporciona un conjunto de referencia que SHAP utiliza para calcular los valores base.

Tabla 9. Parámetros para la generación de explicaciones de SHAP de una instancia determinada.

Parámetros	Componente	Descripción
shap_values = explainer(scaler.transform(X_test.iloc[[1500]]))	Selección de instancia	Calcula los valores SHAP para esos datos. Explica el impacto de cada característica en la predicción del modelo para la instancia específica de datos proporcionada.
shap.initjs()	Inicialización de JavaScript	Prepara el entorno del notebook para visualizar gráficos interactivos SHAP con JavaScript.
shap.force_plot(shap_values.base_values, shap_values.values[0], X_test.iloc[1500], matplotlib=True)	Creación de gráfico de fuerza	Genera una visualización interactiva que muestra cómo cada característica afecta la predicción del modelo desde un valor base hasta la predicción final para la instancia específica.

Los resultados de las técnicas de ML y de XAI se detallan en el siguiente capítulo, donde se presentan gráficamente los resultados de cada etapa.

## **3. RESULTADOS**

En la presente sección, se expone un análisis detallado de los resultados derivados de la aplicación de la metodología propuesta. De manera particular, se presentan los hallazgos producto de la ejecución del esquema de pruebas diseñado para evaluar los algoritmos de aprendizaje automático. Además, se examinan las salidas proporcionadas por el algoritmo de XAI, las cuales se manifiestan en el formato de explicaciones comprensibles y accesibles para el usuario. Estas explicaciones buscan esclarecer las decisiones del modelo, permitiendo un entendimiento profundo de las características influyentes y su impacto en las predicciones generadas.

### **3.1. Modelos de Machine Learning**

En el esquema de evaluación establecido en el capítulo anterior, se definieron criterios específicos que facilitarán la selección del algoritmo más eficiente para resolver la problemática de clasificación planteada. Se expone, para cada uno de los algoritmos de aprendizaje automático examinados, los resultados obtenidos en las matrices de confusión, expresadas en términos porcentuales para cada categoría. Adicionalmente, se presentan las métricas de rendimiento clave: Precisión (Precision), Sensibilidad (Recall) y la Puntuación F1 (F1-Score), que juntas ofrecen una perspectiva integral sobre la efectividad de los modelos.

#### **3.1.1. Support Vector Machine**

A continuación, en la Figura 37, se muestra la matriz de confusión correspondiente a los resultados proporcionados por el algoritmo SVM:

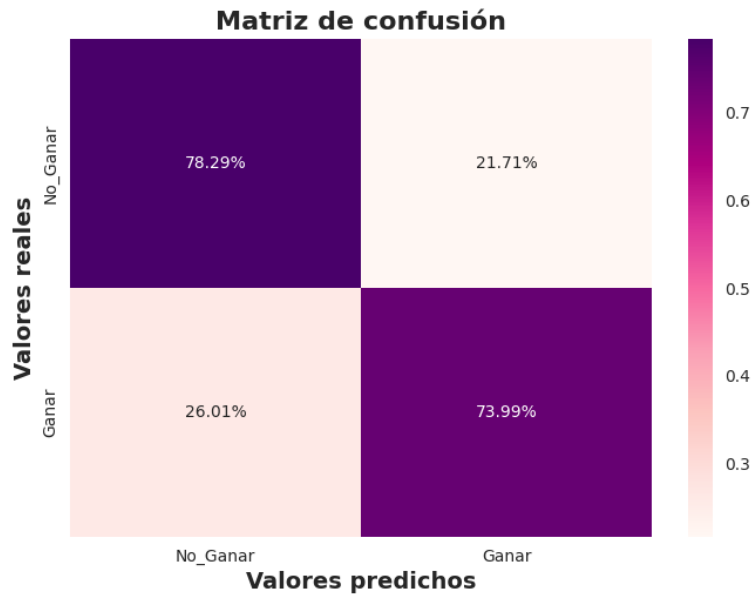


Figura 37. Matriz de confusión de los resultados del modelo de clasificación Support Vector Machine.

Las métricas de precisión (Precision), sensibilidad (Recall) y la puntuación F1 (F1-Score) correspondientes al modelo SVM se detallan en la Tabla 10.

Tabla 10. Resultado de las métricas Precision, Recall y F1-Score para el modelo de clasificación Support Vector Machine.

<b>Support Vector Machine</b>			
	<b>Precision</b>	<b>Recall</b>	<b>F1-Score</b>
<b>No_Ganar</b>	0.78	0.87	0.83
<b>Ganar</b>	0.74	0.60	0.66

### 3.1.2. Random Forest

A continuación, en la Figura 38, se muestra la matriz de confusión correspondiente a los resultados proporcionados por el algoritmo RF:

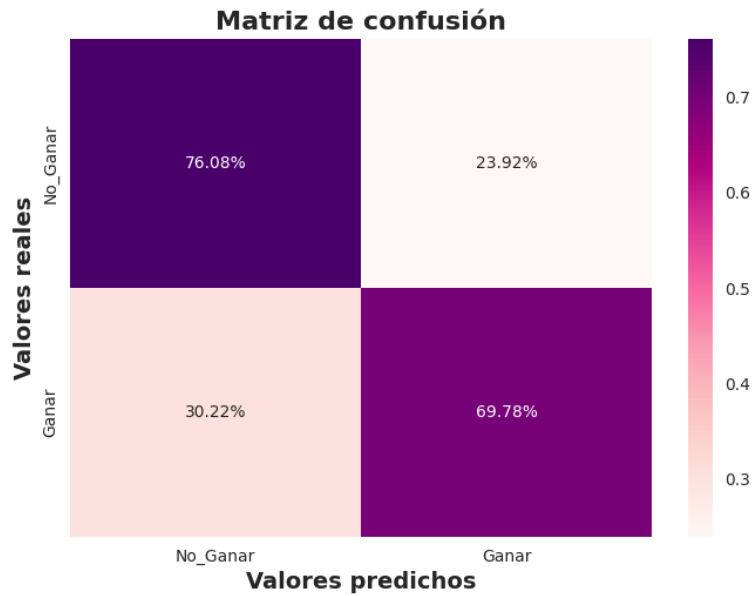


Figura 38. Matriz de confusión de los resultados del modelo de clasificación Random Forest.

Las métricas de Precision, Recall y F1-Score del modelo RF se visualizan en la Tabla 11.

Tabla 11. Resultado de las métricas Precision, Recall y F1-Score para el modelo de clasificación Random Forest.

<b>Random Forest</b>			
	<b>Precision</b>	<b>Recall</b>	<b>F1-Score</b>
<b>No_Ganar</b>	0.75	0.85	0.80
<b>Ganar</b>	0.69	0.54	0.60

### 3.1.3. Red Neuronal Artificial

A continuación, en la Figura 39, se muestra la matriz de confusión de los resultados proporcionados por ANN:

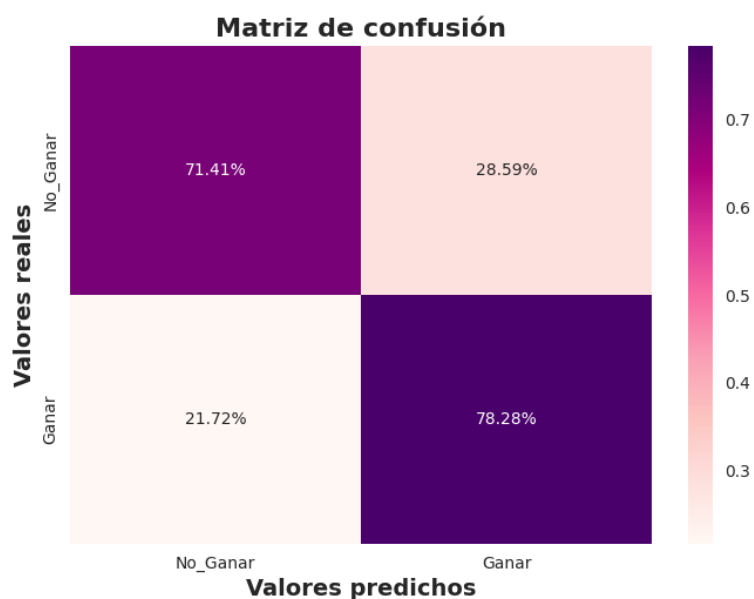


Figura 39. Matriz de confusión de los resultados del modelo de clasificación con ANN.

Los valores de Precision, Recall y F1-Score del modelo de red neuronal artificial se visualizan en la Tabla 12.

Tabla 12. Resultado de las métricas Precision, Recall y F1-Score para el modelo de clasificación de Red Neuronal Artificial.

Red Neuronal Artificial			
	Precision	Recall	F1-Score
No_Ganar	0.71	0.61	0.66
Ganar	0.78	0.85	0.82

Las matrices de confusión en el análisis de resultados de partidos de fútbol presentan los datos de manera porcentual, basándose en el número total de predicciones para cada posible resultado: Ganar o No Ganar. Para mejorar la comprensión, estas matrices utilizan una escala de colores donde la intensidad aumenta con el valor de cada celda.

Visualmente, se puede observar que los tres algoritmos de predicción ofrecen un rendimiento aceptable en cuanto a la precisión de sus predicciones (valores en la diagonal de la matriz), con ninguno cayendo por debajo del 69% de precisión total. Sin embargo, mirar solo la diagonal para determinar el mejor algoritmo es insuficiente; es necesario analizar los resultados en el contexto del problema específico del fútbol.



En este caso, los algoritmos están diseñados para predecir los resultados de los partidos de fútbol. Cada tipo de error en la predicción tiene su importancia, especialmente considerando la influencia de estas predicciones en la contratación y mejora de jugadores para un club deportivo profesional.

En términos prácticos, predecir un empate o una derrota cuando en realidad el equipo gana (falso negativo), podría llevar a subestimar las capacidades de un equipo. Por otro lado, predecir una victoria cuando el equipo realmente empata o pierde (falso positivo) podría llevar a una sobreestimación no deseada, en ambos casos esto provoca una visión poco clara de las habilidades de los jugadores.

Por lo tanto, en este contexto, queremos minimizar los falsos positivos en las predicciones. La métrica más relevante aquí es el Recall, que indica la proporción de predicciones correctas de un resultado específico entre todas las predicciones hechas para ese resultado.

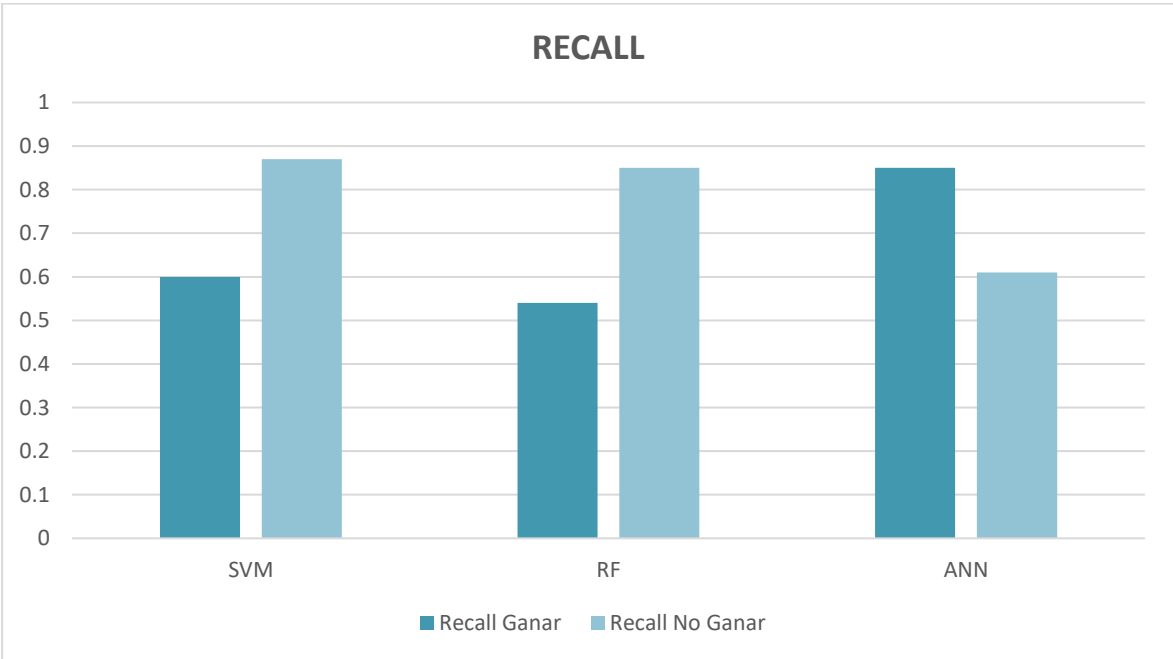


Figura 40. Comparación de los resultados de la métrica Recall de los algoritmos implementados.

La Figura 40 nos indica que los algoritmos con mayor puntaje de Recall son los de SVM para la categoría no ganar y ANN para la categoría ganar, por ese motivo se podría considerar que cualquiera de estos dos algoritmos podría ser útiles para la implementación

de las técnicas de XAI, sin embargo, dado que el algoritmo SVM tiene mayor valor de Recall en una de sus categorías (No Ganar) que el ANN en la categoría (Ganar) y que adicionalmente su valor de F1-Score es ligeramente mayor (0.82 en No Ganar), se decidió que el algoritmo SVM es el indicado.

## **3.2. Resultados de modelos de XAI**

### **3.2.1. SHAP**

En el contexto de nuestra investigación, hemos adoptado el método SHAP de XAI, que satisface la necesidad de interpretación de los modelos. SHAP ofrece un marco para proporcionar explicaciones locales para diversos modelos de machine learning, incluidos aquellos que no son exclusivamente árboles de decisión, lo que justifica nuestra elección de no limitarnos a un TreeExplainer.

En su lugar, hemos implementado un `shap.Explainer` que automáticamente selecciona el mejor explicador disponible para el modelo con el que se está trabajando, además desde la versión 0.37 de SHAP, `shap.Explainer` actúa como una API de alto nivel que encapsula varios explicadores más específicos como `KernelExplainer`, `TreeExplainer`, `DeepExplainer`, además es eficiente computacionalmente para modelos específicos, ya que utiliza explicadores optimizados para diferentes tipos de modelos, optimizando la eficiencia y la exactitud de las explicaciones.

El uso de `shap.Explainer` ofrece una solución eficiente y flexible para interpretar las predicciones de modelos de ML. Aprovechando su capacidad para seleccionar automáticamente el tipo de explicador más adecuado para el modelo en cuestión, podemos generar explicaciones locales detalladas para instancias individuales sin incurrir en una carga computacional excesiva.

Para nuestra evaluación, hemos identificado una instancia representativa de cada clase de clasificación en salud fetal (Ganar, No Ganar) dentro de nuestro conjunto de datos. Las instancias seleccionadas son:

- Instancia 1500 para la clase Ganar
- Instancia 600 para la clase No Ganar

Estas instancias han sido elegidas para ilustrar cómo el modelo de SVM determina sus predicciones, mejorando así la transparencia y la comprensión del razonamiento detrás del proceso de toma de decisiones del modelo. Con shap.Explainer, podemos explorar el impacto y la relevancia de cada característica en la decisión final, proporcionando una visión integral y accesible de la lógica del modelo.

### Instancia Ganar

La Tabla 13 muestra los atributos con sus respectivos valores de una instancia que representan la clase Ganar.

Tabla 13. Instancia de la clase Ganar.

<b>Parámetros</b>	<b>Team (T)</b>	<b>Opponent (O)</b>
Overall Team	75.85	71.81
Age	27.42	25.54
Height (cm)	184.61	184.27
Weight (kg)	80.25	77.43
Weak Foot	2.89	2.95
Skill Moves	2.71	2.73
International Reputation	1.61	1.24
Pace	67.56	70.61
Shooting	60.24	57.86
Dribbling	70.28	69.02
Defending	59.17	56.48
Mentality/Composure	69.32	64.68
Expected Goals (xG)	2.80	0.60
Possession (%)	49.00	51.00
Shots	19.00	11.00
Shots on Target	5.00	2.00
Distance (yd)	17.80	21.80
Penalty Kicks	1.00	0.00
Attacking Skills	59.34	54.74
Movement/Speed Skills	66.45	66.19
Power Skills	66.04	63.49
Mentality/Aggression Skills	61.20	56.11
Mentality/Vision Skills	59.98	55.90
Defending Skills	55.65	50.34
Unique	0.00	0.00
Home	1.00	0.00

## Instancia No Ganar

La Tabla 14 muestra los atributos con sus respectivos valores de una instancia que representan la clase No Ganar.

Tabla 14. Instancia de la clase No Ganar.

<b>Parámetros</b>	<b>Team (T)</b>	<b>Opponent (O)</b>
Overall	69.79	77.22
Age	24.31	25.07
Height (cm)	181.40	184.57
Weight (kg)	74.17	79.23
Weak Foot	3.11	3.37
Skill Moves	2.47	2.61
International Reputation	1.24	2.24
Pace	71.17	75.11
Shooting	55.31	65.16
Dribbling	64.77	71.86
Defending	51.68	58.64
Mentality/Composure	69.16	69.16
Expected Goals (xG)	1.30	1.30
Possession (%)	49.68	49.68
Shots	12.41	12.41
Shots on Target	4.11	4.11
Distance (yd)	17.65	17.65
Penalty Kicks	0.00	0.00
Attacking Skills	54.81	61.92
Movement/Speed Skills	67.97	70.27
Power Skills	62.62	68.15
Mentality/Aggression Skills	54.87	59.55
Mentality/Vision Skills	53.71	62.86
Defending Skills	48.75	53.51
Unique	0.00	0.00
Home	1.00	0.00

## Interpretación de Cada Instancia

Para comprender profundamente las interpretaciones proporcionadas por el modelo SHAP, es crucial analizar la media de cada atributo en el conjunto de datos global (Tabla 15). Esta media actúa como un punto de referencia que nos permite evaluar los valores de atributos individuales en cada instancia específica. Al comparar los valores de una instancia con estas medias, podemos identificar cómo las variaciones en los atributos influyen en las

decisiones del modelo. Si un atributo en una instancia particular se desvía significativamente de la media, esto puede indicar que el atributo juega un rol importante en la clasificación realizada por el modelo. Por ejemplo, un valor muy superior o inferior al promedio podría ser un indicador clave de por qué una instancia ha sido asignada a una clase particular.

Tabla 15. Valores medios de cada atributo en el conjunto de datos.

<b>Atributo</b>	<b>Team (T)</b>	<b>Opponent (O)</b>
Overall Team	74.42	74.48
Age	25.01	25.00
Height (cm)	182.58	182.59
Weight (kg)	76.51	76.51
Weak Foot	3.10	3.10
Skill Moves	2.64	2.64
International Reputation	1.62	1.63
Pace	71.44	71.42
Shooting	59.92	59.82
Dribbling	69.77	69.77
Defending	56.98	57.11
Mentality/Composure	69.26	69.29
Expected Goals (xG)	1.31	1.31
Possession (%)	50.06	49.86
Shots	12.56	12.51
Shots on Target	4.17	4.20
Distance (yd)	17.59	17.64
Penalty Kicks	0.10	0.11
Attacking Skills	57.28	57.27
Movement/Speed Skills	68.25	68.24
Power Skills	64.94	64.94
Mentality/Aggression Skills	58.07	58.23
Mentality/Vision Skills	58.24	58.21
Defending Skills	52.21	52.39
Unique	0.02	0.02
Home	0.50	0.00

### **Explicaciones con el algoritmo SHAP correspondientes la clase Ganar.**

En el contexto de explicaciones locales, es habitual emplear gráficos de contribución de características. Estos gráficos ilustran cómo la influencia de cada atributo individual contribuye a la predicción final del modelo. A medida que se analiza el valor de cada atributo, la gráfica muestra cómo estos influyen en la inclinación hacia la clase predicha, en este caso, Ganar.

Para interpretar adecuadamente el Gráfico SHAP anterior, Figura 41, se debe comenzar examinando la línea de base establecida por el valor medio de las predicciones del modelo para todo el conjunto de datos, que en este caso es 0,36, simbolizando  $E[f(x)]$  como se muestra en la Figura 42, este el punto de partida que representa la predicción promedio o esperada para el modelo antes de tomar en cuenta cualquier característica. El valor final a la derecha indica la predicción del modelo después de considerar todas las características.

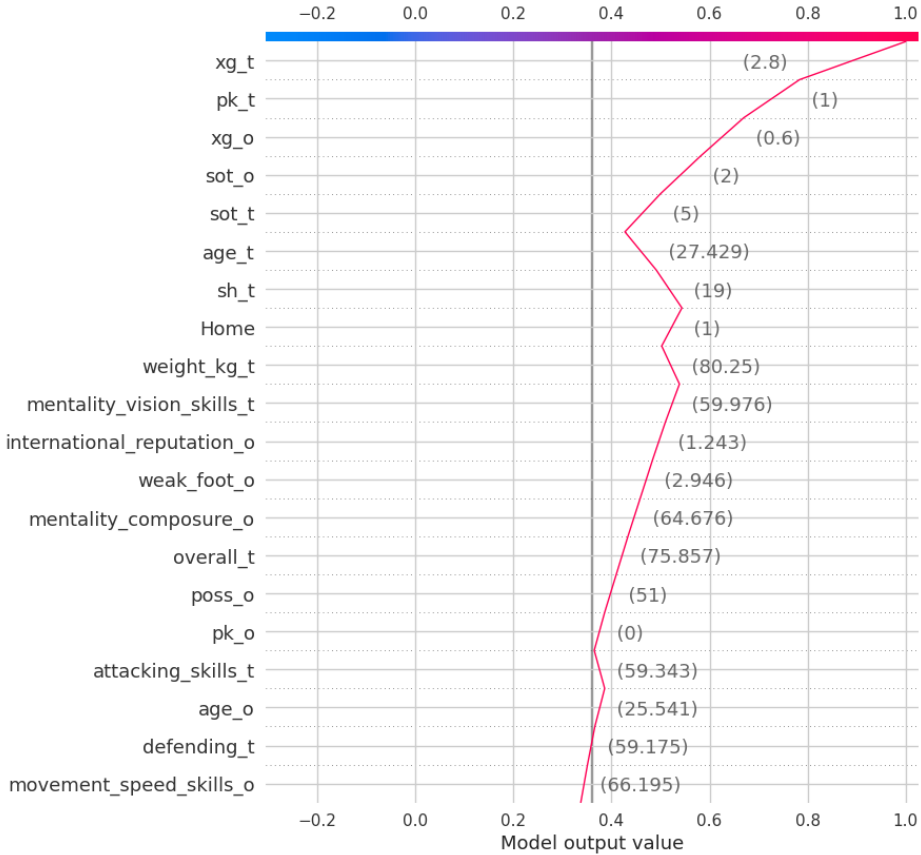


Figura 41. Gráfico de decisión de SHAP para realizar explicaciones de la instancia Ganar.

Podemos ver que algunas características como xg\_t (Expected Goals del equipo) y overall\_t (Calificación general del equipo) tienen un impacto positivo fuerte en la predicción, ya que la línea roja se desplaza hacia la derecha. Por otro lado, características como weight\_kg\_t (Peso del equipo) tienen un impacto negativo ya que la línea se mueve hacia la izquierda.

La Figura 42 es un gráfico de cascada de SHAP, que es utilizado para interpretar la contribución de cada característica a una predicción específica de un modelo de machine learning. Las barras rojas (positivas) empujan la predicción hacia la predicción final (ganar), mientras que las barras azules (negativas) la alejan de esa clase. Podemos ver nuevamente que `xg_t` (Expected Goals del equipo) y `Home` (equipo local) son atributos que aportan para que el equipo gane.

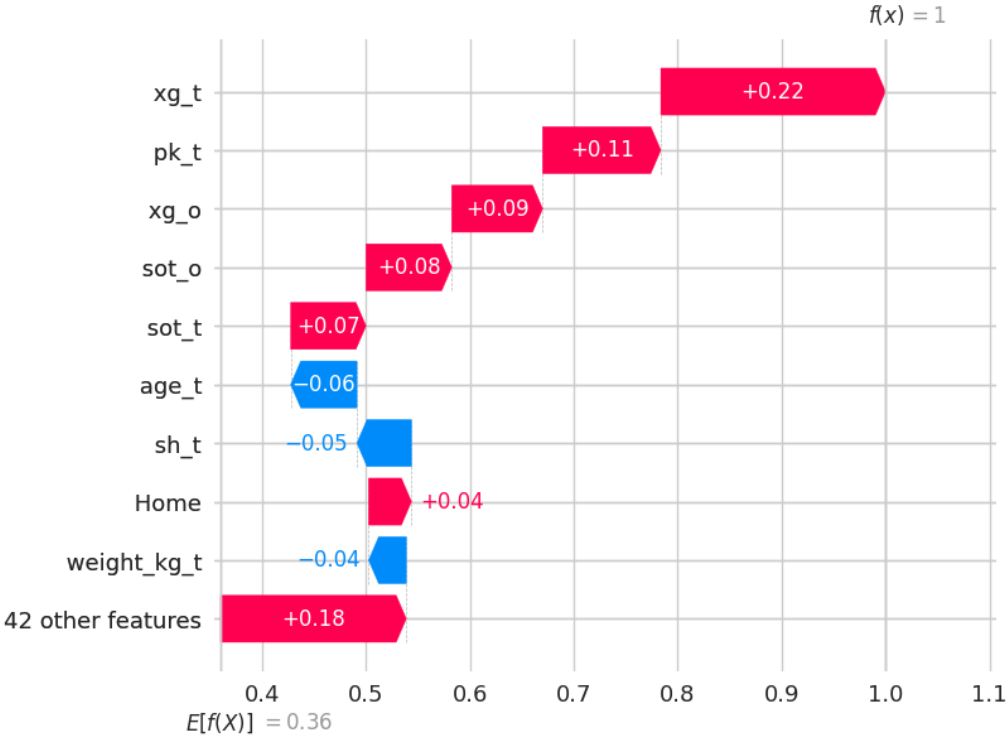


Figura 42. Gráfico de cascada de SHAP para realizar explicaciones de la instancia Ganar.

**Explicaciones con el algoritmo SHAP correspondientes a la clase No Ganar.**

En la Figura 43, se puede observar el gráfico de impacto en la salida del modelo en la categoría No Ganar dependiendo de las variables listadas en el eje vertical. En este caso, la característica en la parte superior, que es la "reputación internacional del oponente" (international\_reputation\_o), tiene la contribución más significativa, otra característica que aporta para que un equipo no gane es el "overall del oponente" los cuales aportan para que la línea se dirija hacia el valor cero (No Ganar), mientras que características como Home (equipo local) y height\_cm\_t (altura de jugadores) aportan a que la línea se dirija hacia 1 (Ganar).

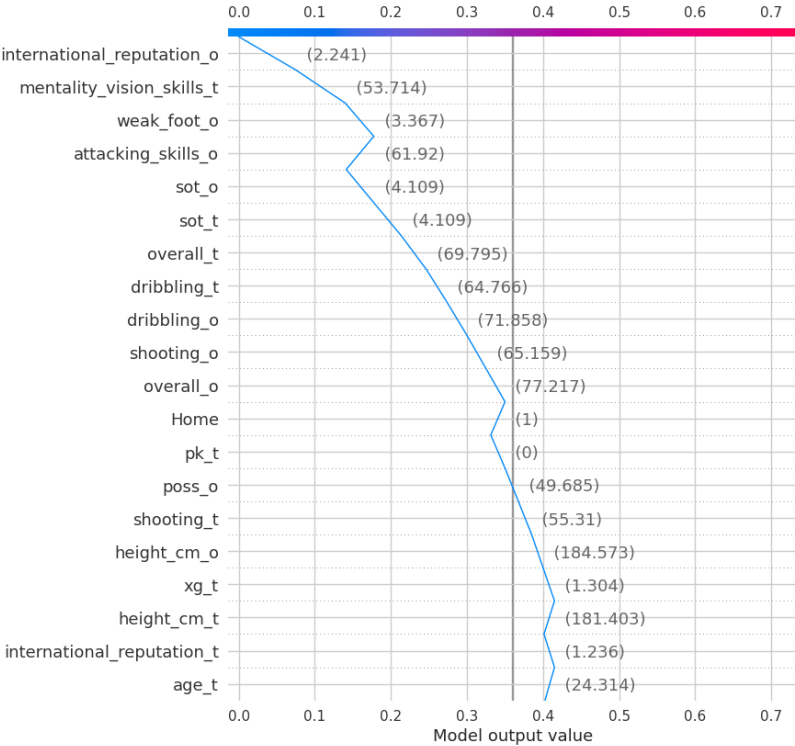


Figura 43. Gráfico de decisión de SHAP para realizar explicaciones de la instancia No Ganar.

El gráfico de cascada de SHAP (Figura 44) muestra que, para esta instancia, las características tienen en su mayoría una influencia negativa en la predicción, alejándola de la categoría GANAR teniendo mayor peso la variable "reputación internacional del oponente" (international\_reputation\_o) y la mentalidad de visión (posicionamiento, visión, penales) del equipo. En el eje vertical se puede apreciar en la parte inferior la etiqueta "42



other features”, lo cual indica que hay otras características que también contribuyen al modelo, pero no se muestran individualmente en el gráfico para evitar la sobrecarga de información.

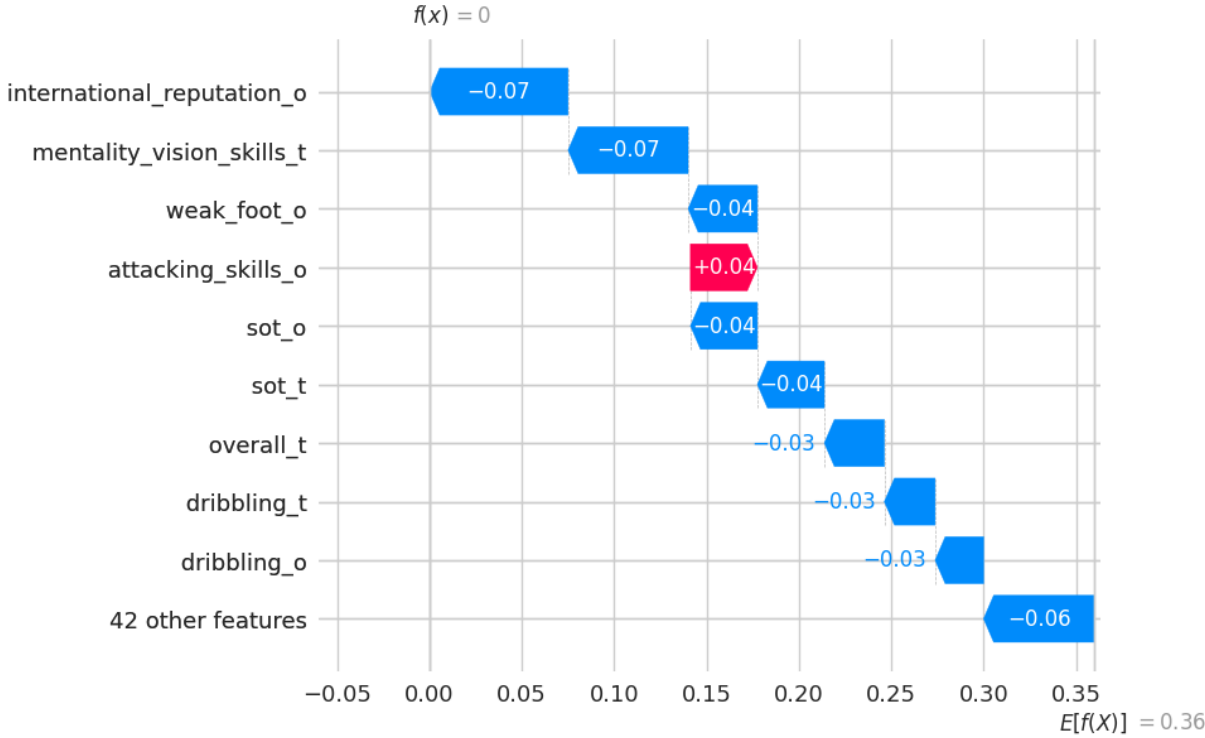


Figura 44. Gráfico de cascada de SHAP para realizar explicaciones de la instancia No Ganar.

### 3.2.2. LIME

#### Explicaciones con el algoritmo LIME correspondientes a la clase Ganar.

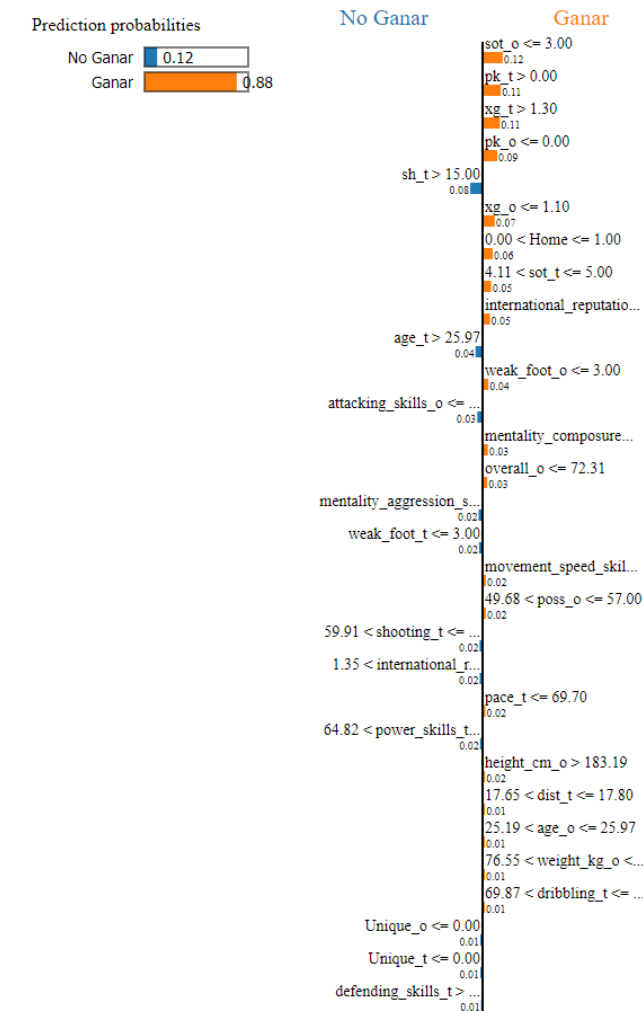


Figura 45. Explicaciones de LIME para una instancia clasificada como "Ganar".

En la Figura 45, se observa dos barras que representan las "probabilidades de predicción" para dos clases: No Ganar y Ganar. La clase Ganar tiene una probabilidad mucho mayor (0.88) que la clase No Ganar (0.12), lo que indica que el modelo predice que el resultado más probable es Ganar, la cual es la instancia de interés en este punto.

A la derecha e izquierda, hay listas de características con valores asociados. Estos valores pueden ser interpretados como los pesos o la importancia que el modelo local asigna a cada característica para esta predicción en particular. Cada característica tiene una barra

que se extiende hacia la derecha o hacia la izquierda. La longitud de la barra indica la magnitud de la contribución de esa característica a la clase correspondiente. Las barras que se extienden hacia la derecha (naranja) aumentan la probabilidad de la clase Ganar, y las que se extienden hacia la izquierda (azul) aumentan la probabilidad de la clase No Ganar.

Entre las características que contribuyen a la predicción Ganar podemos observar nuevamente variables como la Home, International\_reputation\_o, overall\_o, entre otras que favorecen a la clase Ganar, pero ahora se presentan dentro de un rango de valores. Estas explicaciones se observan de manera más detallada en la Figura 46.

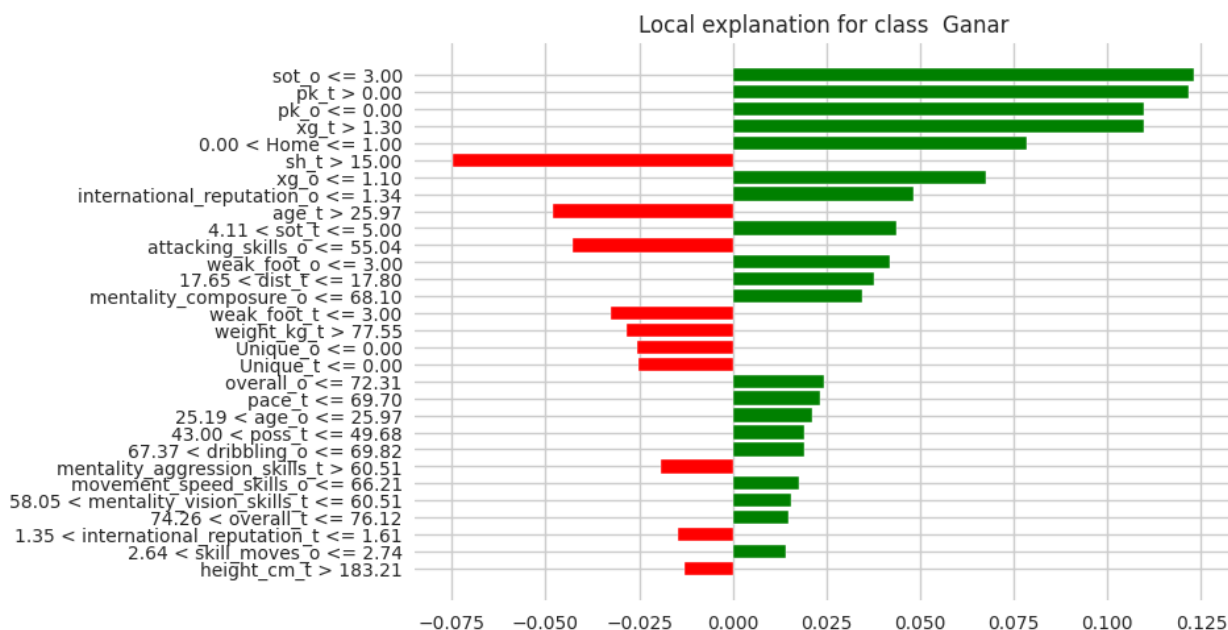


Figura 46. Explicación detallada de LIME de la clase Ganar.

La longitud de la barra se relaciona con el peso de la evidencia que cada característica aporta a la predicción. La etiqueta en cada barra muestra el nombre de la característica y un rango o condición, de esta manera se observa que las características que mayor aportan a la clase Ganar, son las de promedio de disparos a portería del oponente (sot\_o), tiros penales (pk\_t, pk\_o), goles esperados (xg\_t) y si el equipo juega de local (Home).

También se observa el nombre de las características mentales o físicas y un rango o condición, por ejemplo, "mentality\_composure\_o <= 68.1" significa que si el oponente tiene

un valor en su habilidad de mantener la calma y el control de sus emociones durante los partidos igual o menor a 68,1 está contribuyendo a la predicción de Ganar.

**Explicaciones con el algoritmo LIME correspondientes a la clase No Ganar.**

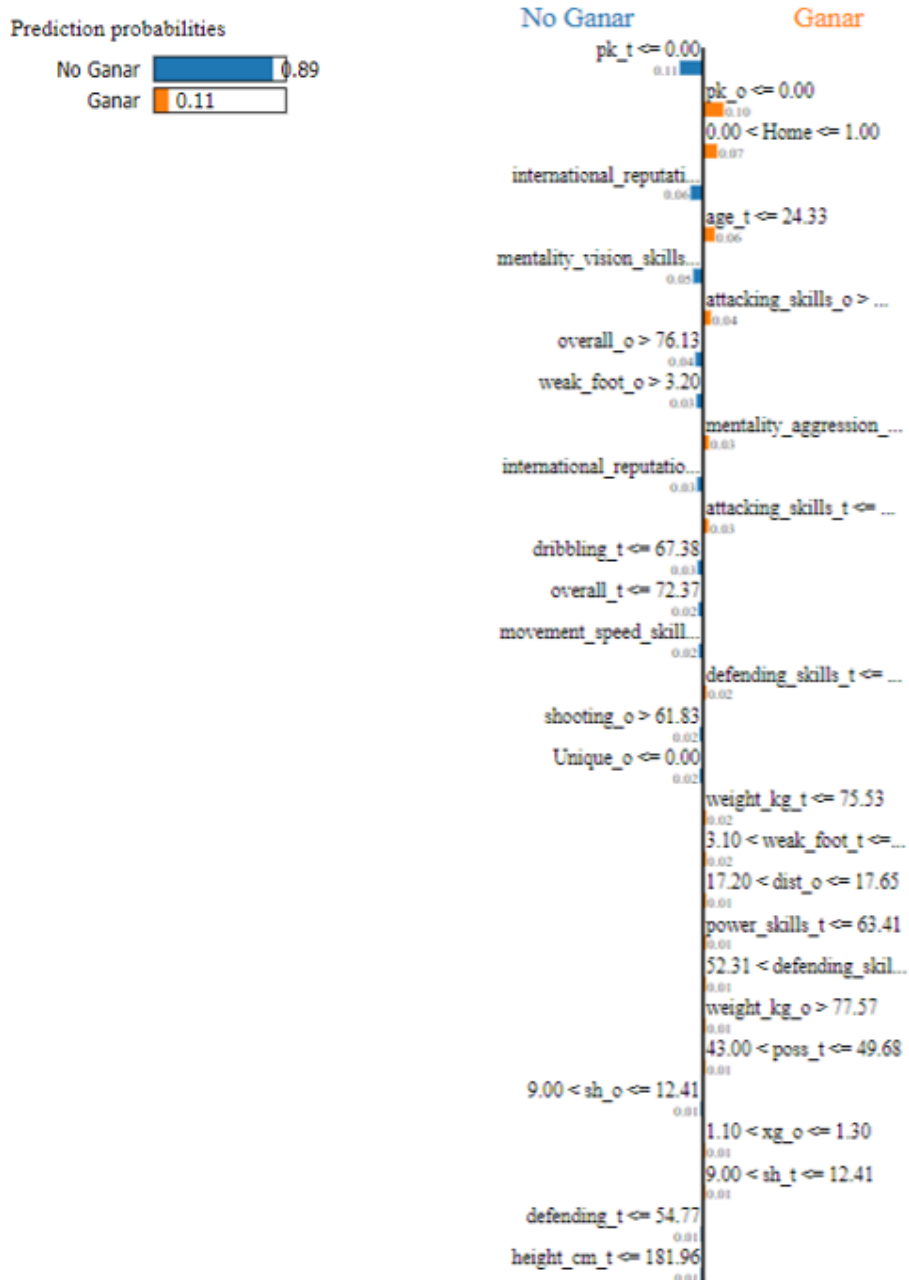


Figura 47. Explicaciones de LIME para una instancia clasificada como No Ganar.

De manera similar que, en la sección anterior, la barra en la parte superior indica que la clase "No Ganar" tiene una alta probabilidad de 0.89 en comparación con una probabilidad baja de 0.11 para la clase Ganar.

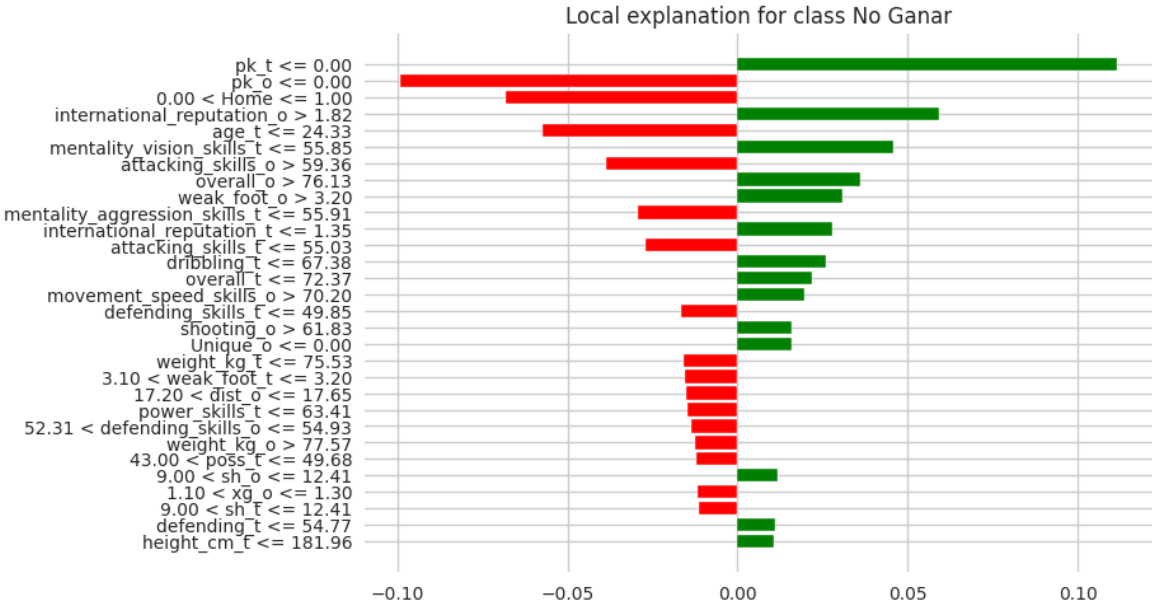


Figura 48. Explicación detallada de LIME de la clase No Ganar.

La Figura 48 nos muestra que entre las características que mayor aportan para que un equipo no gane un partido (empate o pérdida) es si un equipo tiene un valor de cero penales promedios ejecutados durante sus partidos. De igual manera se tiene que recalcar que valores altos de reputación internacional del contrincante (international\_reputacion\_o), overall del oponente (overall\_o) o mayor habilidad con su pie secundario de los jugadores oponentes (weak\_foot\_o) hacen que el equipo pierda.

## **4. CONCLUSIONES Y RECOMENDACIONES**

### **4.1. CONCLUSIONES**

El segmento más exigente en términos de inversión de tiempo en este proyecto ha sido la fase de análisis y entendimiento de los grupos de datos, seguida de cerca por el proceso de la calidad de los datos. Dado que inicialmente se tenían fuentes diferentes de los datos (resultados de partidos, características de jugadores) que debían combinarse y limpiar. Este periodo crítico ha constituido la mayor parte del esfuerzo dedicado al proyecto, culminando en la creación de un conjunto de datos fidedigno y organizado que sienta las bases para avanzar en las etapas subsiguientes de la investigación.

En la ejecución de este proyecto se aplicó la metodología CRISP-DM, que proporcionó una estructura de referencia clara para cada fase de la investigación y un esquema detallado para la documentación y la entrega de información. Esto resultó ser importante en la comprensión de los datos, así como en su relevancia dentro del estudio y comprensión de los resultados.

Dentro de la aplicación de algoritmos de aprendizaje automático: SVM, RF y ANN se concluyó que los algoritmos SVM y ANN son los que mejores resultados presentan en cuanto a la clasificación de resultados deportivos teniendo valores similares en sus métricas de Recall y F1-score en las diferentes categorías de resultados de los partidos.

Los algoritmos de XAI de SHAP y de LIME entregaron resultados significativos para determinar la contribución que tiene cada característica de los equipos que compiten en una liga profesional de fútbol tanto para que ganen o no un partido. Sin embargo, se puede concluir que el algoritmo que da una explicación más detallada de los valores de las características es el de LIME, ya que sus gráficas muestran dichos atributos en forma de rangos que debe tener para aportar o no a cada clase.

Los dos algoritmos de XAI presentan atributos que aportan para que un equipo pueda ganar o no un partido. Entre las características que más aportan para que un equipo gane un partido se encuentran los goles esperados que anoten los equipos y si el equipo juega de local, sin embargo si se toman los valores entregados por LIME se puede detallar que las

características como una baja reputación de los jugadores del equipo rival ( $\text{international\_reputation\_o} < 1.34$ ), la poca habilidad del pie secundario de los jugadores del equipo rival ( $\text{week\_foot\_o} \leq 3$ ) y la poca habilidad de mantener la calma y control de los jugadores del equipo rival ( $\text{mentality\_composure\_o} \leq 68.1$ ) provocan que un equipo pueda ganar el equipo. En otras palabras, equipos que se enfrentan con rivales con jugadores de baja reputación, con pocos jugadores ambidiestros y temperamentales o que pierden el control fácilmente tienen mayor oportunidad de ganar.

Dentro de las características entregadas por SHAP y LIME que atribuyen para que un equipo no gane (empate o pérdida) se encuentra una alta reputación internacional del oponente lo cual concuerda con mencionado en la conclusión anterior, así como un alto promedio en general del oponente ( $\text{overall\_o} > 76.13$ ). Pero basándose más en los resultados entregados por LIME se tiene que características mentales y físicas como una baja mentalidad de los jugadores en su posicionamiento y visión para realizar jugadas o penales ( $\text{mentality\_vision\_skills\_t} \leq 55.85$ ), una baja estatura promedio de sus jugadores ( $\text{height\_cm\_t} \leq 181.96$ ), valores promedio de edad altos ( $\text{age\_t} > 24.33$ ) y peso promedio de sus jugadores altos ( $\text{weight\_kg\_t} > 77.55$ ) tiene impacto en que un equipo pierda un partido.

## 4.2. RECOMENDACIONES

Este estudio, enfocado en un sector concreto del ámbito deportivo, destaca cómo la inteligencia artificial puede aportar significativas mejoras. Sin embargo, la efectividad del proyecto está intrínsecamente ligada a la calidad del conjunto de datos empleado. Un desafío notable en este contexto es el desbalanceo en la disponibilidad de datos; no todas las variables contaban con un número equitativo de datos. Se optó por no realizar un balance forzado de los datos para preservar la veracidad de las conclusiones extraídas por medio de los algoritmos de XAI. Por ese motivo se recomienda que para futuros trabajos se cuenta con una mayor cantidad de datos originales balanceados que puedan ayudar a obtener mejores resultados en las métricas de evaluación de algoritmos como la precisión general.

Uno de los resultados que son más difíciles de predecir en el ámbito deportivo son los empates, por ese motivo en este estudio se unieron los datos de empates y derrotas en una sola categoría. Se sugiere que para próximos trabajos se tome en cuenta de forma independiente la categoría empate teniendo los datos lo más balanceados posibles en las tres categorías.

A pesar del buen desempeño de los algoritmos de XAI implementados en este trabajo, existen ciertas variables como el tipo de cuerpo de los jugadores, o el porcentaje de posesión de balón de los jugadores, distancia de los disparos a la portería, entre otras, que deberían ser analizadas a mayor profundidad por expertos del área deportiva ya que tienen influencia en el resultado final de una competencia.

Se recomienda la aplicación de algoritmos de XAI, en especial el de LIME ya que dio los mejores resultados, con información de varias ligas profesionales de fútbol, ya que esto mejoraría la calidad de los resultados obtenidos por los algoritmos y una mejor comprensión del negocio y características mentales y físicas que debe tener el equipo ideal de fútbol.



## 5. REFERENCIAS BIBLIOGRÁFICAS

- [1] M. del P. Malagón Selma, "Machine learning en el mundo del fútbol," UPV, <https://riunet.upv.es/handle/10251/129491> (accedido May 7, 2023).
- [2] "La Copa Mundial de la FIFA catar 2022TM," FIFA, <https://www.fifa.com/es/fifa-world-cup-qatar-2022-in-numbers> (accedido May 7, 2023).
- [3] G. Fialho, A. Manhães, and J. P. Teixeira, "Predicting sports results with Artificial Intelligence – A proposal framework for soccer games," *Procedia Computer Science*, vol. 164, pp. 131–136, 2019. doi:10.1016/j.procs.2019.12.164.
- [4] W. Dubitzky, P. Lopes, J. Davis, and D. Berrar, "The Open International Soccer Database for Machine Learning," *Machine Learning*, vol. 108, no. 1, pp. 9–28, 2018. doi:10.1007/s10994-018-5726-0.
- [5] A. Rossi, L. Pappalardo, and P. Cintia, "A narrative review for a machine learning application in sports: An example based on injury forecasting in soccer," *Sports*, vol. 10, no. 1, p. 5, 2021. doi:10.3390/sports10010005.
- [6] I. H. Sarker, "Machine Learning: Algorithms Real-World Applications and Research Directions," *SN Computer Science*, vol. 2, no. 160, Mar. 2021. doi:10.1007/s42979-021-00592-x.
- [7] D. Jurafsky and J. H. Martin, "Speech and Language Processing," 3rd ed., [Online]. Available: <https://web.stanford.edu/~jurafsky/slp3/>. (accedido: May. 30, 2023).
- [8] Z. Noshad, N. Javaid, T. Saba, Z. Wadud, M. Q. Saleem, M. E. Alzahrani, and O. E. Sheta, "Fault Detection in Wireless Sensor Networks through the Random Forest Classifier," 1 April 2019.
- [9] P. Banerjee, "Random Forest Classifier Tutorial," Kaggle, 2019. [Online]. Available: <https://www.kaggle.com/code/prashant111/random-forest-classifier-tutorial>. (accedido: May. 30, 2023).
- [10] M. Shao, X. Wang, Z. Bu, X. Chen, y Y. Wang, "Prediction of energy consumption in hotel buildings via support vector machines," en *Sustainable Cities and Society*, vol. 57, no. 102128, 2020.
- [11] A. F. Reinoso Quijo, "Generación de un modelo predictivo que permita identificar el rendimiento académico de estudiantes de la EPN en base a su nivel de acceso a TICs y factores socioeconómicos para la toma estratégica de decisiones de autoridades de la EPN," Tesis de Magíster en Sistemas de Información, Mención Inteligencia de Negocios y Analítica

- de Datos Masivos, Facultad de Sistemas, Escuela Politécnica Nacional, Quito, enero 2023.
- [12] J. Runge y R. Zmeureanu, "Forecasting Energy Use in Buildings Using Artificial Neural Networks: A Review," Centre for Net-Zero Energy Building Studies, Department of Building, Civil, and Environmental Engineering, Concordia University, Montreal, QC, Canadá, publicado el 23 de agosto de 2019.
- [13] A. Megina Gonzalo, "Evaluación de Modelos de Inteligencia Artificial Explicable. Caso de Uso con Clasificadores de Licitaciones Públicas a partir de Textos," Trabajo Fin de Grado, Grado en Ingeniería Informática, Escuela Técnica Superior de Ingenieros Informáticos, Universidad Politécnica de Madrid, Madrid, Junio 2022.
- [14] Q. V. Liao y K. R. Varshney, "Human-Centered Explainable AI (XAI): From Algorithms to User Experiences," arXiv:2110.10790v5 [cs.AI], 19 Abril 2022.
- [15] B. D. Ortuño Barrera, "Evaluación y Aplicación de Algoritmos de Inteligencia Artificial Explicada para Apoyar la Toma Decisiones Médicas en la Salud Fetal," Trabajo de Integración Curricular, Escuela Politécnica Nacional, Facultad de Ingeniería de Sistemas, Quito, Febrero 2022.
- [16] I. S. Rivas Hidalgo, "Evaluación y Aplicación de Algoritmos de Inteligencia Artificial Explicada para Apoyar la Toma de Decisiones Médicas en la Salud Fetal," Trabajo de Integración Curricular, Escuela Politécnica Nacional, Facultad de Ingeniería de Sistemas, Marzo de 2022.
- [17] G. Vaquero de Miguel, "Aproximaciones a la Explicación de Decisiones Algorítmicas: Inteligencia Artificial Explicable," Trabajo Fin de Grado, Grado en Ingeniería Informática, Escuela Técnica Superior de Ingenieros Informáticos, Universidad Politécnica de Madrid, Madrid, Julio de 2020.
- [18] V. P. Singh, A. Ahmad, y K. M. Jagtap, "Weighted Shapley Value: A Cooperative Game Theory for Loss Allocation in Distribution Systems," Department of Electrical Engineering, National Institute of Technology, Srinagar, Hazratbal, Srinagar, India, 2023.
- [19] A. Lalwani, A. Saraiya, A. Singh, A. Jain, y T. Dash, "Machine Learning in Sports: A Case Study on Using Explainable Models for Predicting Outcomes of Volleyball Matches," arXiv:2206.09258v1 [cs.LG], 18 Junio 2022.
- [20] A. Panjan, N. Šarabon, y A. Filipčič, "Prediction of the Successfulness of Tennis Players with Machine Learning Methods," *Kinesiology*, vol. 42, no. 1, pp. 98-106, 2010.
- [21] A. Rossi, L. Pappalardo, y P. Cintia, "A Narrative Review for a Machine Learning Application in Sports: An Example Based on Injury Forecasting in Soccer," en *Sports*, vol. 10, no. 5, 2022.

doi:10.3390/sports10010005. Publicado el 24 de diciembre de 2021.

- [22] M. Organista Ayala y S. Muñoz Benítez, "Sistema experto: Elección de jugadores de fútbol a partir de unas determinadas características seleccionadas," Universidad Carlos III de Madrid, Madrid-España.
- [23] G. Fialho, A. Manhães, y J. P. Teixeira, "Predicting Sports Results with Artificial Intelligence: Framework for Soccer Games – A Proposal," en *Procedia Computer Science*, vol. 164, pp. 131–136, 2019. doi: 10.1016/j.procs.2019.12.164.
- [24] D. C. Albuja González, "Desarrollo de un modelo predictivo para el análisis de anomalías de desempeño en futbolistas profesionales utilizando minería de opiniones," Tesis Máster en Sistemas de Información, Mención en Inteligencia de Negocios y Analítica de Datos Masivos, Facultad de Ingeniería de Sistemas, Escuela Politécnica Nacional, Quito, Ecuador, Dic. 2021.
- [25] Google Colab, "Colaboratory," [En línea]. Disponible en: <https://colab.research.google.com/?hl=es>. [Accedido: Abril, 2023].

## 6. ANEXOS

### ANEXO I

Enlace dirigido al código para la extraer la información de los resultados de los partidos de fútbol de la Premier League.

**<https://colab.research.google.com/drive/1R7Zsw65MLwf38JVniCMMGf3gOlwSXAdx#scrollTo=f1u5bH4PaaBL>**

## **ANEXO II**

Enlace dirigido al código para unir la información de partidos jugados de la Premier League y la información de las características de los jugadores.

**[https://colab.research.google.com/drive/1B\\_UYRr7Qir9vZPNGoVt5wpX5xWAeKjFo#scrollTo=PgaOd3UQnfP3](https://colab.research.google.com/drive/1B_UYRr7Qir9vZPNGoVt5wpX5xWAeKjFo#scrollTo=PgaOd3UQnfP3)**

## **ANEXO III**

Enlace dirigido al código de aplicación de algoritmos de Inteligencia Artificial Explicada (XAI)

**[https://colab.research.google.com/drive/1yovTGwUzVsulKQbd\\_oicTZZvf\\_ue2Miz#scrollTo=V7EDjcc0zIZf](https://colab.research.google.com/drive/1yovTGwUzVsulKQbd_oicTZZvf_ue2Miz#scrollTo=V7EDjcc0zIZf)**