****

MODELOS PREDICTIVOS DE RESULTADOS DEPORTIVOS: APLICACIÓN DE MACHINE LEARNING EN PLATAFORMAS DE APUESTAS

MODALIDAD DEL TFG: CONVENCIONAL

CONVOCATORIA: ORDINARIA

ALUMNA: ANA ISABEL GONZÁLEZ SAHAGÚN

TUTORA: ICÍAR CIVANTOS GÓMEZ

GRADO: INGENIERÍA DEL SOFTWARE

Contenido

[RESUMEN 4](#_Toc193474881)

[ABSTRACT 4](#_Toc193474882)

[1. INTRODUCCIÓN 5](#_Toc193474883)

[1.1 Motivación y Contexto 5](#_Toc193474884)

[1.2 Planteamiento del Problema 6](#_Toc193474885)

[1.2.1 Definición de Éxito de los Equipos 6](#_Toc193474886)

[1.2.2 Selección del Modelo y Limitaciones 7](#_Toc193474887)

[1.2.3 Necesidad de Definir Métricas Claras para Evaluar el Modelo 8](#_Toc193474888)

[1.2.4 Tema poco desarrollado 8](#_Toc193474889)

[1.3 Objetivos del Trabajo 9](#_Toc193474890)

[2. ESTADO DE LA CUESTIÓN 10](#_Toc193474891)

[2.1 Plataformas de Apuestas Deportivas 10](#_Toc193474892)

[2.1.1 Historia y Evolución de las Apuestas Deportivas 10](#_Toc193474893)

[2.1.2 Funcionamiento de las Apuestas Deportivas 10](#_Toc193474894)

[2.1.3 Datos Estadísticos del Mercado: Impacto Económico y Social 12](#_Toc193474895)

[2.2 El Fútbol y las Apuestas 13](#_Toc193474896)

[2.2.1 Tipos de Apuestas en el Fútbol 13](#_Toc193474897)

[2.2.2 Impacto Económico de las Apuestas en el Fútbol 15](#_Toc193474898)

[2.3 Marco Teórico del Trabajo 16](#_Toc193474899)

[2.3.1 Introducción al Machine Learning 16](#_Toc193474900)

[2.3.2 Machine Learning: Conceptos Básicos 18](#_Toc193474901)

[2.3.3 Machine Learning: Evaluación del Modelo 23](#_Toc193474902)

[2.3.4 Machine Learning: Modelos Lineales 27](#_Toc193474903)

[2.3.5 Machine Learning: Modelos no Lineales 29](#_Toc193474904)

[2.3.6 Machine Learning: Redes Neuronales 34](#_Toc193474905)

[2.4 Trabajos Relacionados 38](#_Toc193474906)

[3. ASPECTOS METODOLÓGICOS 40](#_Toc193474907)

[3.1 Metodología 40](#_Toc193474908)

[3.1.1 Planificación 40](#_Toc193474909)

[3.1.2 Organización y seguimiento del trabajo 41](#_Toc193474910)

[3.1.3 Adaptaciones y ajustes en la metodología 41](#_Toc193474911)

[3.2 Tecnologías Empleadas 42](#_Toc193474912)

[3.2.1 Lenguajes y entornos de desarrollo 42](#_Toc193474913)

[3.2.2 Bibliotecas y Frameworks 42](#_Toc193474914)

[3.2.3 Github 42](#_Toc193474915)

[3.2.4 MLflow 43](#_Toc193474916)

[4. DESARROLLO DEL TRABAJO 45](#_Toc193474917)

[4.1 Adquisición, Análisis y Procesamiento de Datos 45](#_Toc193474918)

[4.1.1 Origen y Descripción de los Datos 46](#_Toc193474919)

[4.1.2 Análisis Exploratorio de los Datos 49](#_Toc193474920)

[4.1.3 Generación de Datasets por Equipos 53](#_Toc193474921)

[5. CONCLUSIONES 54](#_Toc193474922)

[6. REFERENCIAS 55](#_Toc193474923)

[6.1 Bibliografía 55](#_Toc193474924)

[6.2 Referencias Teóricas Consultadas 56](#_Toc193474925)

[6.2 Índices de imágenes 57](#_Toc193474926)

[ANEXOS 59](#_Toc193474927)

# RESUMEN

Breve resumen del TFG en español. Se recomienda describir en pocas palabras, no más de dos párrafos, la temática, el trabajo desarrollado y la conclusión.

# ABSTRACT

Brief summary of the TFG in English. It is recommended to describe in a few words, no more than two paragraphs, the topic, the work developed and the conclusion.

# 1. INTRODUCCIÓN

## 1.1 Motivación y Contexto

El deporte, además de ser una gran fuente de entretenimiento, mueve cifras económicas notables en todo el mundo. Las competiciones de futbol atraen a seguidores de todas las edades y países, generando ingresos millonarios a través de la venta de entradas, derechos de transmisión, patrocinio y publicidad. El deporte se ha convertido en un factor decisivo para el crecimiento económico y el desarrollo social de muchos países. Dentro de este ámbito, el fútbol destaca como el deporte con mayor número de aficionados, impulsando la creación de empleo y fomentando la actividad comercial a su alrededor.

En 2023, la famosa empresa consultora KPMG realizó un estudio sobre el impacto socioeconómico del fútbol profesional en España. Según este informe, el fútbol profesional alcanzó un impacto económico de 18.350 millones de euros durante la temporada 2021/2022, equivalente al 1,44% del PIB nacional [[1](#_6.1_Bibliografía)]. Además, las apuestas deportivas en línea rebasaron los 2.950 millones de euros en ese mismo periodo.

Las plataformas de apuestas se enfocan en anticipar de la forma más precisa posible el resultado de estos encuentros deportivos. Esto se debe a que un cálculo correcto de estas predicciones afecta de manera directa en sus ganancias y en las decisiones de quienes apuestan. Sin embargo, la alta incertidumbre de los eventos deportivos hace complicada la tarea de encontrar la fórmula que se acerque a la predicción perfecta. Existen numerosos estudios como “*Luck is Hard to Beat: The Difficulty of Sports Prediction”* que destacan como la mezcla de habilidad y azar en un encuentro deportivo hace que ningún modelo sea capaz de asegurar un resultado final con total exactitud [[2](#_6.1_Bibliografía)].

Precisamente, este componente impredecible del deporte, y en particular del fútbol, lo convierte en un ámbito interesante para el estudio de Inteligencia Artificial y Machine Learning. Actualmente, la IA es un campo en pleno auge que ha experimentado un crecimiento acelerado en los últimos años. Su aplicación en el ámbito deportivo ha permitido optimizar estrategias tácticas, mejorar el rendimiento de los jugadores y prevenir lesiones mediante el análisis de patrones físicos y niveles de fatiga. Sin embargo, la predicción de resultados sigue siendo un área poco explorada debido a la gran cantidad de factores impredecibles que influyen en un partido, como el estado físico de los jugadores, las decisiones arbitrales o lesiones inesperadas. Estos factores hacen que predecir un marcador sea un desafío complejo pero a la vez un problema interesante que pone a prueba las capacidades actuales de la Inteligencia Artificial.

## 1.2 Planteamiento del Problema

Predecir resultados deportivos es un reto complejo que requiere la capacidad de analizar múltiples elementos que pueden influir en el desarrollo de un partido. Cada partido y cada jugada está influida por una combinación infinita de factores, como la forma física y mental de los jugadores, las estrategias empleadas, las condiciones climatológicas externas, y el azar. Esta tarea representa un gran desafío técnico, pero también una oportunidad con impacto significativo. ¿Es posible que un modelo de Machine Learning sea capaz de detectar patrones ocultos y convertir esa aleatoriedad en algo más predecible? Para desarrollar el problema, se ha utilizado un conjunto de datos históricos de partidos de la liga española desde 2003.

A continuación, se presentan los principales desafíos que enfrenta este problema, organizados en diferentes secciones.

1.2.1 Definición de Éxito de los Equipos

Determinar qué significa el éxito en el fútbol es clave para construir modelos de predicción precisos. La victoria de un encuentro es la medición inmediata más directa para evaluar el éxito de un equipo. Sin embargo, la definición de éxito toma matices específicos según el interés del estudio. Por ejemplo, para medir el rendimiento de un equipo se pueden considerar factores como la clasificación final en una liga, la diferencia de goles, el porcentaje de victorias de una temporada, y otros indicadores relacionados con el desempeño. Existe la necesidad de realizar un análisis previo para determinar cuáles de estas variables resultan más útiles a la hora de desarrollar modelos de predicción e interpretar los resultados. A continuación se presentan dos métricas utilizadas en diferentes estudios para analizar el rendimiento de los equipos.

**Soccer Power Index (SPI)**

El SPI es un sistema de calificación desarrollado por FiveThirtyEight que estima la calidad de los equipos de fútbol a nivel mundial[[1]](#footnote-1). Este índice combina datos históricos con un modelo predictivo que evalúa el rendimiento de los modelos en función de su capacidad ofensiva y defensiva. Se basa en los siguientes componentes:

1. Calificación de ataque: Estima la cantidad de goles esperados que un equipo marcaría contra un oponente promedio.
2. Calificación de defensa: Estima la cantidad de goles que un equipo recibiría contra un oponente promedio.
3. Resultados recientes: Se ponderan más los partidos recientes, pero sin ignorar completamente los datos históricos.

**Expected Goals (xG)**

Es una de las métricas más influyentes en la analítica moderna del fútbol, utilizada en numerosos estudios [[3](#_6.1_Bibliografía)]. En lugar de medir simplemente los coles marcados, el xG evalúa la calidad de cada oportunidad de gol basándose en múltiples factores:

1. Ubicación del disparo: La distancia y el ángulo con respecto a la portería influyen en la probabilidad de éxito
2. Tipo de pase previo: Dependiendo del pase, la probabilidad del gol puede cambiar
3. Tipo de disparo: Disparos con el pie, cabezazos o situaciones uno contra uno. Cada uno tiene probabilidades distintas de éxito.
4. Situación de juego: Los tiros en jugadas a balón parado, penaltis, o en jugadas abiertas tienen distintos valores de xG.

Un xG alto significa que el equipo ha generado oportunidades de alta calidad, incluso si no ha marcado goles. Estudios como *Beyond Expected Goals* del MIT han demostrado que el xG es un predictor más confiable del rendimiento ofensivo futuro en comparación con los goles marcados reales [[4](#_6.1_Bibliografía)]. Sin embargo, la tarea de recopilación de datos para calcular el xG es mucho más compleja, ya que requiere un alto nivel de precisión en la medición de variables como la ubicación del disparo, la velocidad del balón y la presión defensiva en cada jugada.

1.2.2 Selección del Modelo y Limitaciones

Existen diversos enfoques utilizados en la predicción de resultados deportivos, cada uno con sus ventajas y sus limitaciones. Modelos como las Redes Neuronales Recurrentes han mostrado resultados prometedores en la predicción de partidos. Son ampliamente utilizados para calcular probabilidades de victoria en encuentros de fútbol debido a su capacidad para manejar grandes volúmenes de datos y detectar patrones complejos.

A pesar de los avances recientes, los modelos de predicción siguen enfrentando varias limitaciones. Estudios muy recientes siguen destacando estas dificultades [[5](#_6.1_Bibliografía)], entre ellas:

1. Calidad de los datos: La precisión de las predicciones depende en gran medida de la calidad y la cantidad de datos disponibles.
2. Sobreajuste a patrones históricos: Muchos modelos tienden a ajustarse demasiado a los datos pasados, lo que reduce su capacidad de generalizar en situaciones nuevas o imprevistas.
3. Imprevisibilidad inherente de los resultados deportivos: Aspectos como lesiones inesperadas o decisiones arbitrales siguen siendo difíciles de predecir con precisión y pueden influir significativamente en el resultado de un partido.

1.2.3 Necesidad de Definir Métricas Claras para Evaluar el Modelo

Para medir la efectividad de los modelos de predicción es fundamental definir métricas de evaluación adecuadas. Un desafío clave es que ninguna métrica es completamente representativa por sí sola. La combinación de varias medidas permite una validación más rigurosa del modelo. Algunas de las más utilizadas incluyen:

* Accuracy: Porcentaje de predicciones correctas. Sin embargo, no siempre es una métrica adecuada en deportes donde los empates y la variabilidad afectan los resultados.
* Log Loss: Evalúa la probabilidad asignada a cada resultado, penalizando predicciones con alta confianza pero incorrectas.
* Brier Score: Similar a Log Loss, mide la precisión de las probabilidades predichas, siendo útil para modelos probabilísticos.
* Área bajo la curva ROC (AUC-ROC): Mide la capacidad del modelo para distinguir entre diferentes categorías (victoria, empate, derrota).

Es posible identificar problemas en la interpretación de estas métricas cuando se aplican a modelos con alta variabilidad, como los partidos de fútbol.

1.2.4 Tema poco desarrollado

A pesar del creciente interés en el uso de Machine Learning en la analítica deportiva, la predicción de resultados deportivos sigue siendo un área con muchas limitaciones. Hasta la fecha, no existe un modelo ampliamente aceptado que haya demostrado predecir con precisión los resultados de fútbol. Muchos estudios presentan buenos resultados en ligas específicas o temporadas concretas, pero fallan al generalizar en otros contextos. Además, no existe un consenso sobre qué modelo o conjunto de técnicas es el más adecuado para este tipo de predicciones.

## 1.3 Objetivos del Trabajo

Este proyecto tiene como objetivo general explorar y evaluar la capacidad de los modelos de Machine Learning para realizar predicciones en un entorno caracterizado por su alta imprevisibilidad, como un partido de fútbol. En particular, se busca determinar si la inteligencia artificial es capaz de predecir una situación de gran aleatoriedad, y hasta qué punto puede reducir la incertidumbre en los resultados. Para lograrlo, se establecen los siguientes objetivos específicos, que guiarán el desarrollo del trabajo:

* **Mejorar la eficiencia de los modelos:** Trabajar con técnicas para ordenar y extraer características clave, facilitando criterios de clasificación claros.
* **Desarrollo y evaluación de modelos**: Implementar diferentes modelos de Machine Learning, ajustarlos y evaluar su rendimiento en función de métricas previamente definidas.
* **Predicción de resultados multiclase** Determinar si un equipo ganará, perderá, o empatará un partido.
* **Comparar la efectividad de los distintos modelos propuestos:** Ofrecer una comparativa del rendimiento de los distintos modelos propuestos para identificar cuál ofrece la mejor precisión en la predicción de resultados.
* **Definir y aplicar métricas de evaluación**: Establecer métricas claras para medir la efectividad de los modelos y analizar sus resultados.
* **Examinar limitaciones y efectividad**: Evaluar las limitaciones de los modelos propuestos, así como su efectividad para reducir la aleatoriedad inherente al ámbito deportivo.

# 2. ESTADO DE LA CUESTIÓN

## 2.1 Plataformas de Apuestas Deportivas

Las apuestas deportivas han ido evolucionando significativamente a lo largo de los años, pasando de ser una actividad informal a una industria multimillonaria impulsada por la digitalización. En este apartado se ofrecerá una breve contextualización del mundo de las apuestas deportivas.

2.1.1 Historia y Evolución de las Apuestas Deportivas

Las apuestas deportivas tienen sus inicios en la antigüedad, con registros que indican que los griegos y romanos apostaban en eventos como en los primeros Juegos Olímpicos y las carreras de carros. Sin embargo, fue en Inglaterra durante el siglo XVIII donde las apuestas comenzaron a formalizarse con la regulación de las carreras de caballos y la aparición de las primeras casas de apuestas oficiales.[[2]](#footnote-2)

En el siglo XX, las apuestas deportivas se expandieron a otros deportes como el fútbol, el boxeo o el baloncesto. Sin embargo, la revolución más significativa ocurrió en los años 90 con la aparición de Internet, lo que permitió la creación de plataformas digitales con apuestas en tiempo real, incluyendo una variedad de mercados mayor.

2.1.2 Funcionamiento de las Apuestas Deportivas

Las casas de apuestas funcionan estableciendo cuotas que reflejan la probabilidad de que ocurra un determinado resultado en un evento deportivo. Las cuotas se calculan utilizando análisis estadísticos y modelos matemáticos que consideran diferentes factores.

Las casas de apuestas aplican un margen de beneficio llamado “*vig*” o “*juice*”, incorporadas en las cuotas ofrecidas a los apostadores para asegurar ganancias a largo plazo. Además, se ajustan las cuotas en función del volumen de apuestas recibidas para equilibrar su exposición y garantizar beneficios independientemente del resultado. Este margen se obtiene ofreciendo cuotas ligeramente inferiores a las probabilidades reales del evento, lo que permite que la suma de las probabilidades implícitas supere el 100%. Además, Las casas incorporan sesgos de mercado, ajustes de liquidez, y consideraciones sobre el volumen total apostado para equilibrar su exposición al riesgo y minimizar pérdidas. Este ajuste se intensifica en apuestas en vivo, donde las cuotas se actualizan en directo según transcurre el evento

**Tipos de Casas de Apuestas**

Existen dos tipos principales de casas de apuestas:

* **Casas de apuestas de contrapartida**: Son las más comunes y operan estableciendo cuotas y aceptando apuestas directamente contra los apostantes. Ejemplos de este tipo incluyen Bet365, William Hill y Bwin.
* **Casas de apuestas de intercambio**: Permiten que los apostantes jueguen entre sí en lugar de hacerlo contra la casa. En este modelo, los usuarios pueden fijar sus propias cuotas y aceptar apuestas de otros jugadores. La casa actúa como intermediaria, obteniendo un beneficio a través de una comisión sobre las ganancias de los jugadores. Un ejemplo destacado es Betfair.

**Ajustes de Cuotas y Margen de Beneficio: Sesgos e Ineficiencias**

Para protegerse de posibles pérdidas, las casas de apuestas incluyen un margen de seguridad en las cuotas. Este margen está basado en la probabilidad estadística de los resultados y en el comportamiento de los apostantes. Existen varios estudios que investigan los sesgos y las ineficiencias de las casas de apuestas [[6](#_6.1_Bibliografía)]. Entre los más importantes se encuentran:

* Longshot Bias: Se refiere a la tendencia de sobrevaloración de las apuestas a resultados poco probables, implicando que la probabilidad real de ganar es subestimada en las cuotas [[7](#_6.1_Bibliografía)].
* Home-Field Advantage Misestimation: Error al estimar la ventaja del equipo local. Tradicionalmente, se considera que el equipo de casa tiene una mayor probabilidad de ganar, lo que se refleja en cuotas más bajas. Pero existen situaciones como la ausencia del público que pueden modificar este efecto.
* Incorporación Tardía de Información: Ocurre cuando las casas de apuestas actualizan de forma lenta las cuotas ante la aparición de nueva información.

**Modelo de rentabilidad**

La rentabilidad de una casa de apuestas se basa en la diferencia entre las cantidades apostadas y los pagos realizados a los ganadores. El margen de beneficio se puede expresar con la formula:

Si el resultado es desfavorable para la casa, su perdida se limita a la cantidad apostada

El modelo de negocio de las casas de apuestas no se basa en acertar los resultados de los eventos, sino de asegurarse ganancias a largo plazo ajustando las cuotas y controlando el riesgo financiero [[8](#_6.1_Bibliografía)].

2.1.3 Datos Estadísticos del Mercado: Impacto Económico y Social

Estudios recientes estiman que el mercado global de apuestas deportivas genera más de 90 millones de dólares anuales. Esta cifra tiene tendencia a seguir aumentando debido a la legalización en varios países y acceso facilitado por plataformas digitales. En regiones como Europa y América del Norte, las apuestas deportivas son un gran porcentaje la industria del entretenimiento y juegos de azar. Se estima que el mercado de las apuestas deportivas previsto para 2030 sea de 608.410 millones USD. [[3]](#footnote-3)

Por otra parte, la accesibilidad de las apuestas en línea ha aumentado el número de casos de adicción al juego, especialmente entre los más jóvenes. Estudios estiman que alrededor de 80 millones de adultos en todo el mundo sufren de adicción al juego.[[4]](#footnote-4) En países como Brasil se ha observado una preocupación creciente por el gasto excesivo en apuestas en línea. Informes indican que los brasileños gastan más de 3.200 millones de euros mensualmente en apuestas, representando el 20% de la masa salarial del país.[[5]](#footnote-5)

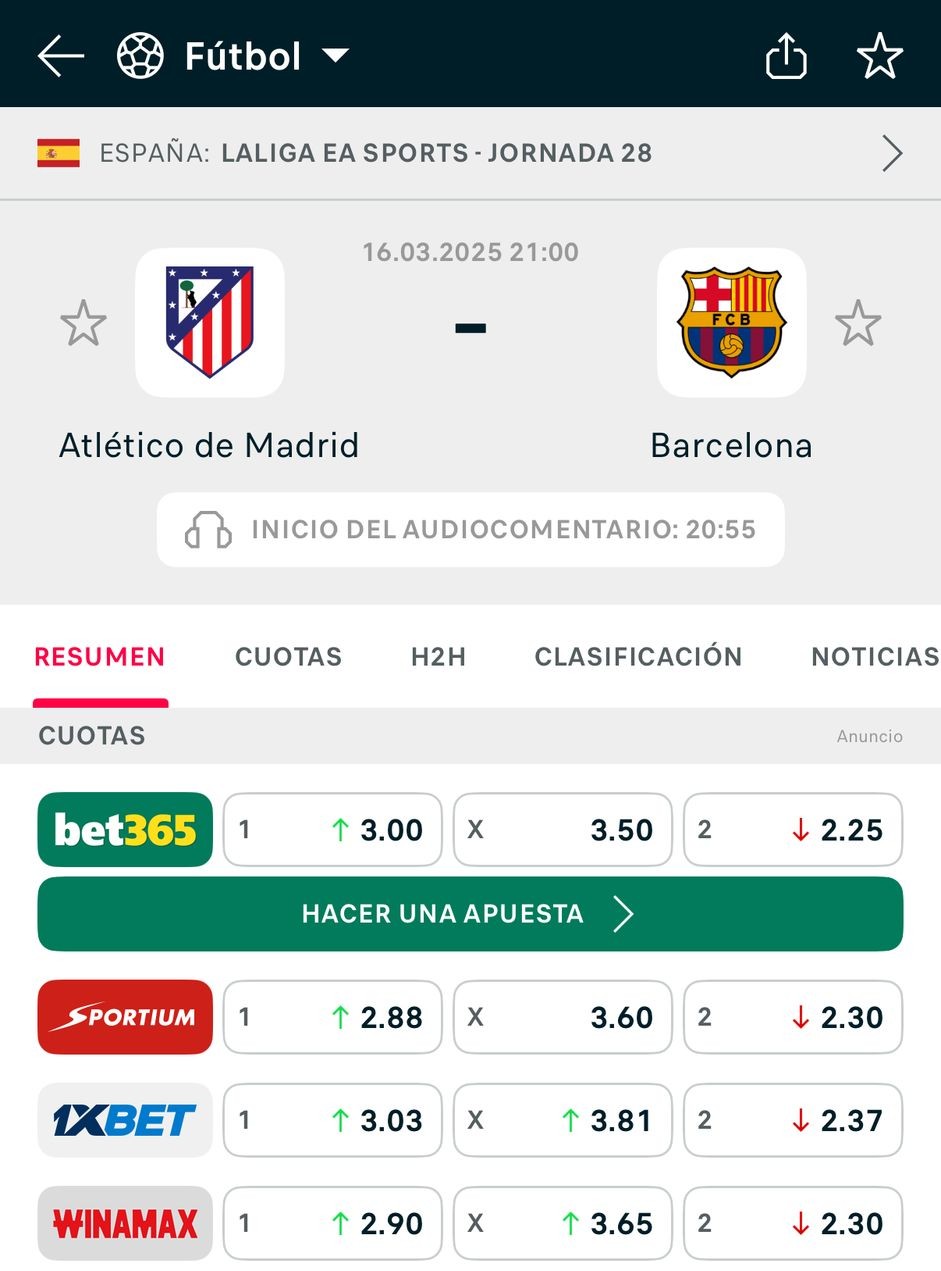
## 2.2 El Fútbol y las Apuestas

El fútbol es el deporte más popular en el ámbito de las apuestas deportivas. Este deporte tiene una gran base global de aficionados y una constante programación de partidos diarios a lo largo del año, representando más del 86% de las actividades de apuestas deportivas en algunos países.[[6]](#footnote-6)

2.2.1 Tipos de Apuestas en el Fútbol

Las apuestas deportivas en el fútbol han evolucionado en las últimas décadas con la digitalización y legalización de plataformas de juego en línea. Aunque existen múltiples modalidades de apuestas, este trabajo se centrará en la siguiente:

**1-X-2:** Apuesta a qué equipo ganará el partido, o si habrá un empate.



[Imagen 1](#_6.2_Índices_de) – Plataforma de resultados Flashscore: Apuestas 1X2

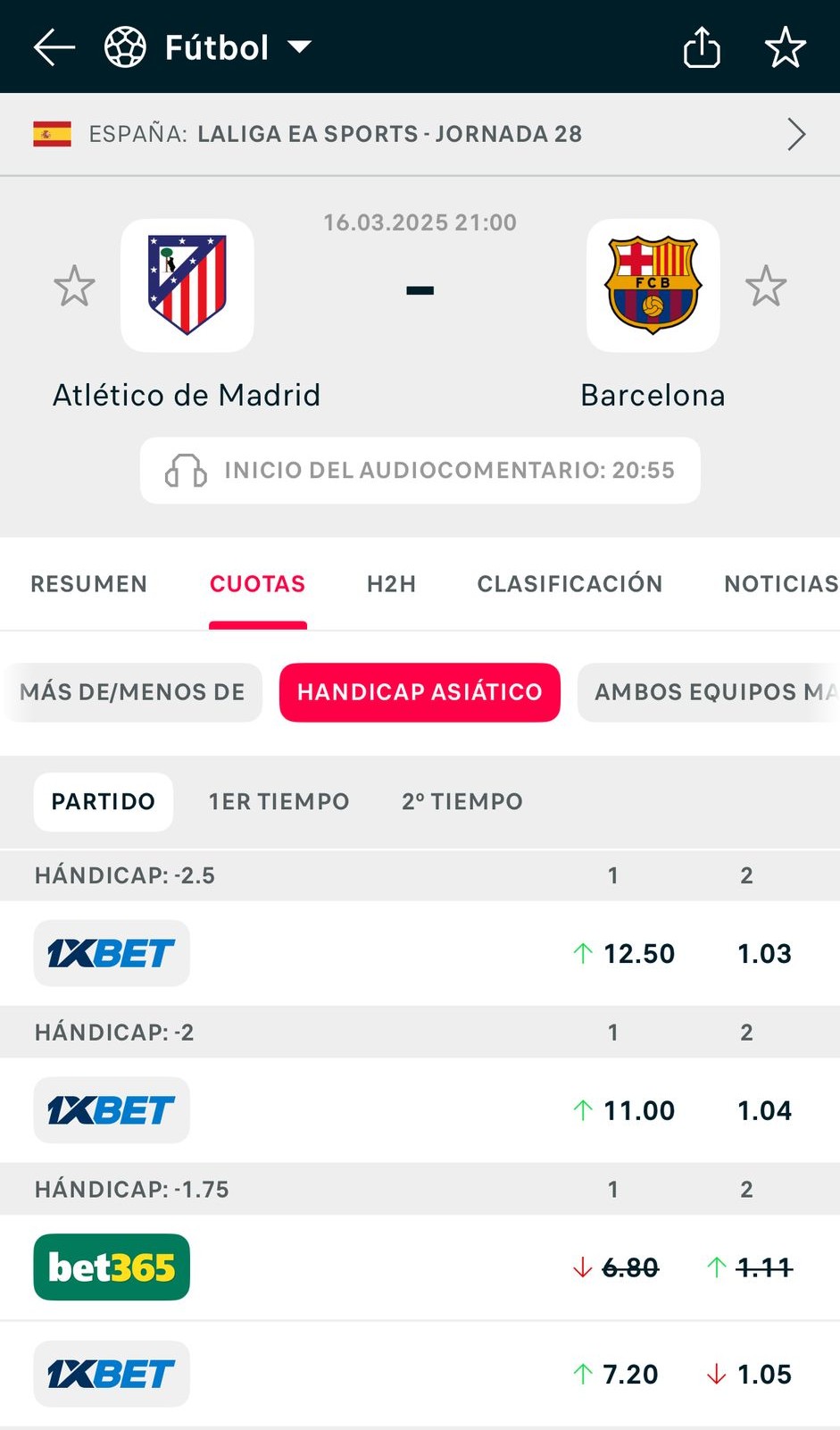
Cada opción tiene una cuota asociada, que representa el pago potencial por cada unidad apostada. Estas cuotas pueden variar entre casas de apuestas y fluctuar en función de diversos factores como estadísticas previas, alineaciones, lesiones o la cantidad de dinero apostado en cada opción.

Si un apostador apuesta 1€ a la victoria del Atlético de Madrid en Bet365 con una cuota de 3.00, y acierta, recibiría 3.00€ en total: 2.00€ de ganancia neta más 1.00€ de la apuesta inicial.

* **Flecha arriba:** Indica que la cuota ha subido, lo que significa que la probabilidad percibida de ese resultado ha disminuido, o bien que hay menos apuestas en esa opción.
* **Flecha abajo**: indica que la cuota ha **bajado**, lo que sugiere que hay más dinero apostado en esa opción y, por lo tanto, se considera más probable.

Sin embargo, existen diversas modalidades de apuestas en el fútbol, cada una con características específicas que permiten diferentes estrategias de juego. Algunas de las más destacadas son:

1. Resultado exacto: Acertar el resultado final del partido
2. Descanso/Final del Partido: Se debe acertar el resultado al término de la primera parte y al final del encuentro.
3. Doble Oportunidad: Permite apostar a dos posibles desenlaces. Ofrecen cuotas más bajas, pero aumentan las probabilidades de acierto.
4. Apuesta con reembolso en caso de empate: Se apuesta por la victoria de un equipo con la ventaja de que, si el partido termina en empate, se devuelve el dinero invertido.
5. Apuestas sobre goles: Incluyen predicciones sobre la cantidad total de goles en el partido.
6. Apuestas especiales: Se pueden realizar predicciones sobre diversos aspectos del partido, como el número de tarjetas mostradas.
7. Scorecast: Se debe acertar tanto el primer goleador del partido como el resultado final, lo que genera cuotas elevadas debido a su dificultad.
8. Hándicap Asiático: Se asigna una ventaja o desventaja en el marcador a uno de los equipos antes del inicio del partido. Su principal característica es que elimina la posibilidad de empate, ya que en muchos casos se ofrecen líneas con fracciones (como -1.5 o -2.5), lo que asegura que siempre habrá un ganador en la apuesta.



[Imagen 2](#_6.2_Índices_de) – Plataforma de resultados Flashscore: Hándicap Asiático

**Hándicap -2.5 en Atlético de Madrid:** Si se apuesta por el Atlético de Madrid (-2.5) a cuota 12.5, el Atlético debe de ganar por al menos 3 goles de diferencia para que la apuesta sea ganadora.

**Hándicap -2 en Atlético de Madrid:** Él Atlético de Madrid debe de ganar por al menos 3 goles de diferencia para que la apuesta sea ganadora y recibir la cuota de 11.0. SI gana exactamente por 2 goles, la apuesta se reembolsa.

**Hándicap -1.75 en Atlético de Madrid:** Aquí, si el Atlético gana por 2 goles exactos, la mitad de la apuesta se gana y la otra mitad se devuelve.

**Flechas arriba:** Indican que la cuota ha subido, lo que significa menor probabilidad percibida de que ocurra ese resultado.

**Flechas abajo**: Indican que la cuota ha bajado, lo que sugiere mayor confianza en esa opción.

2.2.2 Impacto Económico de las Apuestas en el Fútbol

La Copa Mundial y la UEFA Champions League son los eventos futbolísticos más populares de la actualidad y ambos tienen un gran impacto económico en el mercado de apuestas. Durante estos torneos, se observa un aumento considerable en la actividad de apuestas, lo que genera fluctuaciones económicas notables. Por ejemplo, en España, las apuestas deportivas representan casi el 1% del PIB nacional. [[1](#_6.1_Bibliografía)]

El fútbol no solo tiene un impacto en la industria del juego, sino que también influye en la economía general del país. La actividad económica generada por el futbol repercute en la generación de empleo, con más de 194.381 empleos a jornada completa, incluyendo puestos directos e indirectos relacionados con el deporte.[[7]](#footnote-7)

## 2.3 Marco Teórico del Trabajo

En este apartado se presentan los conceptos teóricos fundamentales para el desarrollo del trabajo. Esta sección se basa en un conjunto de artículos introductorios que explican los fundamentos del Machine Learning, con especial referencia al libro *An Introduction to Statistical Learning*) [[9](#_6.1_Bibliografía)], junto con otros trabajos adicionales. Solo se explican los conceptos básicos necesarios para comprender el desarrollo del trabajo, y todas las fuentes utilizadas para establecer esta base teórica están listadas en el apartado [6.2 Referencias Teóricas Consultadas](#_6.1_Bibliografía).

2.3.1 Introducción al Machine Learning

El Machine Learning (ML) es una subrama de la Inteligencia Artificial que se centra en desarrollar algoritmos capaces de aprender y hacer predicciones a partir de datos, sin ser programados específicamente para ello. Algunas de las aplicaciones del Machine Learning son las siguientes:

* Reconocimiento de imágenes y voz.
* Procesamiento del lenguaje natural.
* Finanzas y predicción de mercados.
* Diagnóstico de enfermedades a partir de imágenes médicas.

**Funcionamiento**

El funcionamiento del Machine Learning se basa en el desarrollo de modelos matemáticos que, a través de la experiencia y la optimización de funciones, pueden mejorar su desempeño en diversas tareas. Estos modelos matemáticos establecen una relación entre las variables de entrada (*features)* y las variables de salida (*targets),* con el objetivo de minimizar una función de error o pérdida. Los modelos aprenden a partir de un conjunto de datos mediante un proceso iterativo de optimización donde se ajustan los parámetros internos para mejorar la precisión.

Matemáticamente, el aprendizaje de los modelos de Machine Learning se puede representar como la búsqueda de una función desconocida:

que mapea un conjunto de entradas a una salida , con objetivo de minimizar (o maximizar) una función objetivo . Esta función objetivo cuantifica la discrepancia entre la predicción y el valor real . Dependiendo del tipo de aprendizaje, los modelos de Machine Learning se clasifican en tres tipos comunes:

1. Aprendizaje supervisado
2. Aprendizaje no supervisado
3. Aprendizaje por refuerzo

**Aprendizaje supervisado**

El aprendizaje supervisado se basa en entrenar modelos utilizando un conjunto de datos etiquetado, es decir, donde cada entrada tiene una salida conocida. Su objetivo es aprender una función de mapeo que relacione las entradas con las salidas correctas para poder hacer predicciones precisas sobre nuevos datos.

En términos matemáticos, este tipo de aprendizaje se basa en la construcción de una función:

que aprende una relación entre un conjunto de características de la entrada y sus respectivas salidas . El modelo se entrena utilizando un conjunto de datos etiquetado:

donde representa un vector de características y cada es la etiqueta correspondiente. El objetivo es minimizar la función de pérdida que cuantifica la discrepancia entre la predicción del modelo y el valor real .

Este es el tipo de aprendizaje que se utilizará para desarrollar este trabajo, dado que tenemos datos históricos de partidos etiquetados con sus respectivos resultados. En los siguientes apartados se detallará con mayor profundidad el funcionamiento de la función de pérdida, los diferentes tipos existentes y su impacto en la precisión del modelo

**Aprendizaje no supervisado**

En el aprendizaje no supervisado, el modelo se entrena con datos que no están etiquetados, es decir, que no hay una salida específica asignada a cada entrada. En lugar de predecir valores específicos, los modelos buscan encontrar agrupaciones o patrones entre los datos.

Matemáticamente, se basa en la optimización de una función de similitud o dispersión, en lugar de minimizar una función de error explícita.

Estos modelos se aplican en tareas como la detección de anomalías y segmentación de clientes, donde no es necesario contar con datos etiquetados. Sin embargo, dado que este proyecto se centra en aprendizaje supervisado, los modelos no supervisados no serán explicados en detalle más adelante.

**Aprendizaje por refuerzo**

El aprendizaje por refuerzo, tambien llamado Reinforcement Learning en inglés, se basa en conseguir que el agente interaccione con un entorno con el objetivo de aprender una política óptima que maximice la recompensa acumulada a lo largo del tiempo. El agente aprende una función de valor que estima la recompensa esperada al tomar una acción en un estado . En términos generales, el agente mejora su toma de decisiones a través de prueba y error, donde recibe retroalimentación en forma de recompensas o penalizaciones para ajustar su comportamiento y maximizar el retorno a largo plazo

Esta técnica de Machine Learning podría resultar interesante para la automatización de estrategias de apuestas deportivas en plataformas en línea. En este caso se entrenaría al agente para decidir cuándo apostar, por qué equipo apostar y cuánto apostar, basándose en probabilidades calculadas a partir de datos históricos, con el objetivo de maximizar las ganancias. Sin embargo este tipo de modelo añade una complejidad adicional significativa al estudio, por lo que no será explicado más adelante.

2.3.2 Machine Learning: Conceptos Básicos

A continuación se presentan los conceptos básicos necesarios para comprender el funcionamiento de los modelos con los que se van a estar trabajando en este Trabajo de Fin de Grado.

**Función de Pérdida**

El objetivo de la función de pérdida es proporcionar una métrica cuantificable para que el algoritmo de optimización pueda minimizar el error y mejorar la precisión del modelo. Esta función mide la diferencia entre la predicción del modelo y el valor real .

Sea un conjunto de entrenamiento etiquetado , donde representa un vector de características y cada es la salida real. El modelo genera una predicción , y la función de pérdida se define como:

donde mide el error de la predicción en una muestra individual.

La elección de la función de pérdida determina como el modelo ajusta los parámetros para minimizar el error. Dependiendo del tipo de modelo, se utilizan diferentes funciones de pérdida.

Para modelos de **clasificación** (predicción de etiquetas discretas) se utilizan las siguientes funciones de pérdida:

**Entropía Cruzada** (Cross-Entropy Loss): Se usa en clasificación, maximizando la probabilidad de la clase correcta.

* Clasificación binaria: dos etiquetas (o clases) posibles
* Clasificación multiclase: varias etiquetas (o clases) posibles

Donde:

es la verdadera clase

es la probabilidad predicha por el modelo para la clase correcta

es el número total de ejemplos en el conjunto de datos

es el número de clases

**Hinge Loss** (para SVM): Se usa en máquinas de soporte vectorial (SVMs) y favorece grandes márgenes de separación entre clases. No se entrará en detalle en este tipo de función de pérdida porque no se utilizará para el desarrollo del trabajo.

Para modelos de **regresión** (predicción de valores continuos) se utilizan las siguientes funciones de pérdida:

* **Error Cuadrático Medio** (MSE - Mean Squared Error): Penaliza los errores grandes más que los pequeños, lo que puede hacer que el modelo sea más sensible a valores atípicos.
* **Error Absoluto Medio** (MAE - Mean Absolute Error): Es menos sensible a valores extremos, pero puede ser menos estable en la optimización.

**Algoritmos de Optimización**

Los algoritmos de optimización son métodos numéricos utilizados para encontrar los valores de los parámetros del modelo que minimizan las funciones de pérdida. En Machine Learning, estos algoritmos ajustan los pesos del modelo a lo largo de las iteraciones del entrenamiento. La optimización eficiente de la función de pérdida es clave para garantizar un entrenamiento estable y efectivo de los modelos.

**Descenso de Gradiente:** El Descenso de Gradiente (Gradient Descent, GD) es el algoritmo más utilizado para optimizar los modelos de Machine Learning. La actualización de los parámetros se realiza con la siguiente regla:

Donde

representa los parámetros del modelo

es la tasa de aprendizaje (Learning Rate)

es el gradiente de la función de pérdida con respecto a

**Tasa de aprendizaje:** La tasa de aprendizaje () es un hiperparámetro en los algoritmos de optimización que controla la magnitud del ajuste de los parámetros del modelo por cada iteración del entrenamiento.

* Valores demasiado grandes pueden causar inestabilidad al modelo
* Valores demasiado pequeños pueden ralentizar la convergencia del modelo

**Variantes del Descenso de Gradiente:** Existen diferentes variantes del algoritmo de optimización del Descenso de Gradiente para mejorar la eficiencia del modelo

* Descenso de Gradiente Estándar (Batch Gradient Descent): Usa todo el conjunto de datos en cada iteración para calcular el gradiente
* Descenso de Gradiente Estocástico (SGD - Stochastic Gradient Descent): Actualiza los parámetros después de cada muestra
* Adam (Adaptive Moment Estimation): Combina SGD con ajustes adaptativos de la tasa de aprendizaje.

**Overfitting y underfitting**

Uno de los objetivos más importantes de los modelos de Machine Learning es la capacidad de generalizar bien a datos no vistos. En este contexto, existen dos problemas comunes:

Gráfico, Gráfico de líneas

Descripción generada automáticamente

[Imagen 3](#_6.2_Índices_de)– Representación gráfica de los conceptos de overfitting y underfitting

**Overfitting (Sobreajuste):** Ocurre cuando el modelo se ajusta demasiado bien a los datos de entrenamiento, incluyendo ruido y patrones irrelevantes. Como resultado, tiene un bajo error en el entrenamiento pero un alto error en datos no vistos.

Soluciones:

* Regularización L1: Agrega una penalización a los parámetros del modelo
* Regularización L2: Reduce la magnitud de los coeficientes del modelo para mejorar la generalización
* Dropout: En redes neuronales, desactiva neuronas aleatoriamente en cada iteración.
* Aumento de datos (Data Augmentation): Genera más datos a partir de los existentes.
* Validación cruzada: Divide los datos en múltiples conjuntos de entrenamiento y prueba.

**Underfitting (Subajuste):** Sucede cuando el modelo es demasiado simple para capturar la estructura de los datos resultando en alto error tanto en el entrenamiento como en la validación.

Soluciones:

* Aumentar la complejidad del modelo (ej., más capas en redes neuronales).
* Agregar más características relevantes en los datos de entrada.
* Entrenar por más épocas para permitir una mejor adaptación a los datos.

**Hiperparámetros**

Los hiperparámetros son valores que se deben definir antes de la optimización del modelo para configurar su entrenamiento.

Ejemplos:

* Tasa de aprendizaje (𝛼): Controla cuánto se actualizan los parámetros en cada iteración.
* Número de árboles en Random Forest: Influye en la estabilidad y precisión del modelo.
* Número de capas y neuronas en Redes Neuronales: Afecta la capacidad de representación del modelo.

Los hiperparámetros pueden ajustarse manual o automáticamente mediante técnicas de optimización. En la búsqueda automatizada, se define un rango de valores posibles para cada hiperparámetro y el modelo realiza múltiples entrenamientos con distintas combinaciones. Finalmente, selecciona la configuración óptima basada en el desempeño obtenido según una métrica de evaluación predefinida.

* Grid Search: Explora todas las combinaciones posibles de hiperparámetros.
* Random Search: Prueba combinaciones aleatorias para encontrar la mejor configuración
* Optuna: Librería de optimización automática de hiperparámetros que utiliza estrategias avanzadas como búsqueda bayesiana y pruning para encontrar la mejor configuración de manera eficiente.

**Reducción de dimensionalidad**

En Machine Learning, manejar conjuntos de datos con muchas variables puede generar redundancia y afectar la eficiencia del modelo. La reducción de dimensionalidad es un proceso que busca representar información con menos variables, eliminando las que son redundantes o tienen poca influencia en la predicción. A continuación se presentan las técnicas más comunes.

**Selección de características**

Consiste en elegir las variables más relevantes y descartar las que aportan poca información o son redundantes. Esta técnica mantiene las variables originales y es útil cuando algunas características son irrelevantes para el problema.

Se divide en:

* Métodos de filtro: Aplican medidas estadísticas para evaluar la relevancia de cada variable.
* Métodos de envoltura: Evalúan distintos subconjuntos de variables con un modelo predictivo.
* Métodos basados en árboles de decisión: Identifican la importancia de cada variable en la predicción.

**Análisis de componentes principales (PCA)**

PCA es una técnica de reducción de dimensionalidad que transforma las variables originales en un nuevo conjunto de variables no correlacionadas. Estas nuevas variables se llaman **componentes principales**.

Matemáticamente, PCA busca encontrar una base ortogonal de menor dimensión en la que la varianza de los datos se preserve en la mayor medida posible. El proceso implica:

1. Estandarización de las variables para que tengan media cero y varianza unitaria.
2. Cálculo de la matriz de covarianza para identificar relaciones entre variables.
3. Descomposición en valores propios, extrayendo los vectores propios que representan las direcciones de mayor varianza.
4. Selección de los componentes principales, eligiendo los que explican la mayor parte de la varianza.

A diferencia de la selección de características, PCA no elige variables específicas, sino que las reestructura, lo que puede dificultar la interpretación directa de los datos.

2.3.3 Machine Learning: Evaluación del Modelo

Después de entrenar un modelo de Machine Learning, es importante evaluar el desempeño para determinar su capacidad de generalización en datos no vistos. La evaluación del modelo se basa en diversas métricas de rendimiento que dependen del tipo de problema: clasificación o regresión.

**División de los datos de entrenamiento:** El conjunto de datos se debe dividir para obtener una medición precisa del rendimiento del modelo

* Entrenamiento (Training set): Datos utilizados para ajustar los parámetros del modelo.
* Validación (Validation set): Conjunto de datos usado para la selección de hiperparámetros y ajuste del modelo.
* Prueba (Test set): Conjunto de datos separado que evalúa el desempeño final del modelo en datos no vistos.

**Evaluación de Modelos de Clasificación**

En problemas de clasificación (ej., predecir si un equipo ganará o perderá), se utilizan métricas basadas en la matriz de confusión. La matriz de confusión presenta cuatro valores fundamentales:

* Verdaderos Positivos (TP): Casos correctamente clasificados como positivos.
* Falsos Positivos (FP): Casos incorrectamente clasificados como positivos.
* Verdaderos Negativos (TN): Casos correctamente clasificados como negativos.
* Falsos Negativos (FN): Casos incorrectamente clasificados como negativos.

Tabla

Descripción generada automáticamente

[Imagen 4](#_6.2_Índices_de)– Matriz de Confusión

**Métricas de clasificación**

***Accuracy***: Mide el porcentaje de predicciones correctas

* Problema: No es confiable para conjuntos de datos desbalanceados

***Precision***: Indica cuántos de los casos clasificados como positivos son realmente positivos.

* Importante cuando los falsos positivos son costosos (ej., detección de fraudes)

***Recall*** (sensibilidad o tasa de verdaderos positivos): Indica cuántos de los casos positivos reales fueron correctamente identificados.

* Importante cuando los falsos negativos son críticos (ej., detección de enfermedades).

***F1-score***: Es el promedio armónico entre *Precisión* y *Recall*, ofreciendo una métrica balanceada.

* Ideal para conjuntos de datos desbalanceados

**Balanceo de clases**

En problemas de clasificación, es común que las clases no estén representadas de manera equitativa en el conjunto de datos, lo que se conoce como desbalanceo de clases. Cuando una clase es mucho más frecuente que las demás, los modelos de Machine Learning pueden inclinarse hacia la predicción de la clase mayoritaria, afectando la capacidad de generalización y reduciendo la efectividad de las métricas de evaluación.

La **accuracy** (precisión global) es una métrica que se utiliza comúnmente para evaluar modelos de clasificación. Esta métrica mide el porcentaje de predicciones correctas. Sin embargo, en conjuntos de datos desbalanceados, esta métrica suele no ser la más adecuada.

Por ejemplo, si un equipo tiene un 90% de victorias y solo un 10% de empates o derrotas, un modelo que siempre prediga la clase mayoritaria obtendrá un 90% de accuracy, aunque nunca prediga correctamente la clase minoritaria. Para detectar y evaluar correctamente el impacto del desbalance, es recomendable fijarse en otras métricas como precision y recall.

* Precision y Recall: Permiten evaluar cuántos de los casos predichos como positivos son realmente positivos y cuántos casos reales positivos fueron correctamente identificados.

**Estrategias para el balanceo de clases**

Para corregir el desbalance, se pueden aplicar diferentes estrategias:

* Undersampling: Reduce la cantidad de instancias de la clase mayoritaria para equilibrar el conjunto de datos. Su método principal incluye la eliminación de instancias de la clase mayoritaria de manera aleatoria, o basado en heurísticas que identifican los ejemplos menos representativos o redundantes.
* Oversampling: El Oversampling aumenta la cantidad de muestras de la clase minoritaria para equilibrar el conjunto de datos. Algunas técnicas comunes incluyen el duplicado de datos de la clase minoritaria de manera aleatoria o la generación de instancias sintéticas interpolando con ejemplos cercanos.

**Validación Cruzada**

La validación cruzada es una técnica que se utiliza para evaluar el rendimiento de un modelo y garantizar su capacidad de generalización a datos no vistos. En lugar de depender de una única partición de datos, se divide el conjunto en múltiples subconjuntos y se realizan varias iteraciones de entrenamiento y prueba.

**K-Fold Cross Validation:** Es el método más común, donde el conjunto de datos se divide en k subconjuntos (folds). En cada iteración:

1. Se entrena el modelo con k-1 folds y se prueba en el fold restante.
2. Se repite el proceso k veces, alternando el fold de validación.
3. Se promedian las métricas obtenidas para una evaluación más estable.

Los valores más comunes son k=5 o k=10, logrando un buen balance entre precisión y eficiencia. En cada iteración:

**Validación Cruzada Estratificada:** En conjuntos de datos desbalanceados, la validación cruzada estratificada mantiene la proporción de cada clase en cada fold, asegurando una evaluación más representativa. Algunas ventajas de esta técnica son:

* Reduce la dependencia de una única partición.
* Mejora la evaluación del modelo, evitando sobreajuste.
* Asegura una utilización eficiente de los datos disponibles.

Para problemas con clases desbalanceadas, se recomienda combinar validación cruzada estratificada con técnicas de balanceo.

**Evaluación de Modelos de Regresión**

En problemas donde se predicen valores continuos (ej., cantidad de goles en un partido), se utilizan métricas basadas en la diferencia entre valores reales y predicciones.

* ***MSE*** – Mean Squared Error:Penaliza los errores grandes debido a la elevación al cuadrado.
* ***RMSE*** – Root Mean Squared Error: Es la raíz cuadrada del MSE. Permite interpretar el error en la misma escala de los datos.
* ***MAE*** – Mean Absolute Error: Mide el error medio en términos absolutos.
* – Coeficiente de Determinación: Mide qué porcentaje de la variabilidad en los datos es explicado por el modelo.

2.3.4 Machine Learning: Modelos Lineales

Los modelos lineales forman la base del aprendizaje automático debido a su simplicidad y eficacia en muchos contextos. Estos modelos asumen que la variable de salida se explica como una combinación lineal de las variables de entrada. En su forma más básica, la ecuación que define este tipo de modelos es:

donde son los parámetros (o pesos) que el modelo aprende del conjunto de datos. Los dos modelos lineales más utilizados son los siguientes:

* Regresión Logística
* Regresión Lineal

**Regresión Logística**

Un modelo de regresión logística modela la probabilidad de que una instancia pertenezca a una clase determinada. La variable respuesta es categórica (binaria o multinomial).

**Regresión logística para clasificación binaria**

La regresión logística se utiliza para tareas de clasificación. Para el caso binario, , y el modelo busca ajustar:

Donde

es el vector de pesos

es el sesgo (bias) o término independiente

es la función sigmoide dada por

Con esta función sigmoide, el valor de la salida se mantiene en el rango (0,1) interpretándose como una probabilidad.

Gráfico, Gráfico de líneas

El contenido generado por IA puede ser incorrecto.

[Imagen 5](#_6.2_Índices_de)– Representación de la función sigmoide

**Regresión logística para clasificación multiclase**

La regresión logística es conocida tradicionalmente como un modelo para clasificación binaria, pero existen 2 enfoques para extenderla a clasificación multiclase (por ejemplo, para predecir si un equipo gana, pierde, o empata.

* **One-vs-Rest** (OvR): Para *K* clases se entrenan *K* clasificadores de regresión Logística, uno para cada clase versus las *K-1* restantes. Durante la inferencia, se escoge la clase con mayor probabilidad estimada.
* **Softmax Regression** (o Regresión Logística Multinomial): Se entrenan todas las clases simultáneamente, con una única función de costo *cross-entropy* modificada para la clasificación multinomial. Se generaliza la función sigmoide a la función softmax.

El método Softmax tiende a ser más elegante y consistente matemáticamente, ya que maneja en un único paso las *K* categorías.

**Función de costo**

La función de costo es una métrica que cuantifica el error entre las predicciones de un modelo y los valores reales. Para entrenar el modelo se utiliza la función de costo de entropía cruzada, también llamada log-loss.

La diferencia entre la función de pérdida (mencionada anteriormente) y la función de costo es que la función de pérdida mide el error para **una sola muestra** del conjunto de datos, mientras la función de costo es el promedio o suma de las pérdidas de **todas las muestras** en el conjunto de datos. En la práctica, ambos términos se confunden porque muchos algoritmos de optimización aplican la misma lógica tanto para una muestra como para el conjunto completo.

**Entrenamiento**

Para el entrenamiento de la Regresión Logística se hace típicamente mediante gradiente descendiente (o variantes). El objetivo es encontrar que minimicen la función de costo *cross-entropy*

1. Inicialización de los parámetros: Los pesos y el sesgo se inicializan (por ejemplo, en cero o con valores aleatorios).
2. Cálculo de gradientes: Se deriva la función de costo respecto a cada parámetro
3. Actualización de los parámetros según la regla:

donde α es la tasa de aprendizaje.

**Regresión Lineal**

Los modelos de Regresión lineal se utilizan para predecir valores continuos , asumiendo una relación lineal entre las variables de entrada *“x”* y la salida *“y”*. La regresión lineal busca modelar la relación entre un conjunto de variables independientes y una variable dependiente continua “y”. La forma general del modelo es:

**Evaluación del modelo:** Para la evaluación de los modelos de regresión lineal se utilizan las métricas vistas en el apartado [2.3.3. Machine Learning: Evaluación del Modelo](#_2.3_Marco_Teórico)

1. Error Cuadrático Medio (MSE)
2. Raíz del Error Cuadrático Medio (RMSE)
3. Error Absoluto Medio (MAE)
4. Coeficiente de Determinación ()

Dado que el objetivo del trabajo es predecir resultados categóricos (victoria, empate o derrota), el enfoque principal estará en modelos de clasificación en lugar de regresión.

2.3.5 Machine Learning: Modelos no Lineales

Los modelos no lineales de Machine Learning permiten capturar relaciones complejas en los datos que no pueden ser representadas mediante una combinación lineal de las variables independientes. Estos modelos son útiles cuando los datos presentan patrones entre variables que no siguen una tendencia lineal simple. A continuación se describen algunos de los modelos no lineales más utilizados en tareas de clasificación multiclase y regresión.

**K-Nearest Neighbors (KNN)**

KNN es un algoritmo de aprendizaje supervisado que se puede utilizar tanto para clasificación como para regresión. Es un método de aprendizaje perezoso *(lazy learning),* lo que significa que no construye específicamente un modelo durante la fase de entrenamiento, sino que guarda el conjunto de datos completo y realiza los cálculos en el momento de la predicción

**Funcionamiento**

El principio básico es que la clase o el valor de una muestra se predice en función de sus *k* vecinos más cercanos en el espacio de características. La distancia entre dos puntos y se calcula utilizando la distancia euclidiana:

También se pueden utilizar otras métricas como la distancia Manhattan o la distancia de Minkowski.

* Para clasificación: La clase de una muestra se determina por votación mayoritaria entre los 𝑘 vecinos más cercanos.
* Para regresión: La predicción se realiza calculando el promedio de los valores numéricos de los 𝑘 vecinos más cercanos.

**Ventajas y desventajas**

K-NN es sencillo de implementar y eficaz a la hora de identificar relaciones complejas. Sin embargo, su rendimiento puede disminuir con grandes conjuntos de datos debido a la necesidad de calcular distancias para todas las muestras. También es sensible a la escala de las variables y al ruido en los datos. La elección del número de vecinos *k* es crucial para evitar el sobreajuste o la subestimación de patrones importantes.

**Árboles de Decisión**

Los Árboles de Decisión también se pueden utilizar tanto para clasificación como para regresión. La estructura de un árbol de decisión consiste en nodos internos que representan condiciones o preguntas sobre los atributos de entrada, ramas que representan el resultado de estas condiciones, y hojas que representan las predicciones finales (clase o valor).

**Funcionamiento**

El algoritmo construye el árbol dividiendo recursivamente el conjunto de datos en subconjuntos más pequeños basados en características. Estas características proporcionan la mayor ganancia de información o la mayor reducción de impureza. Hay diferentes medidas de impureza. A continuación se presentan las más comunes:

1. Índice Gini para clasificación:

donde es la proporción de instancias de la clase *i* en un nodo y *C* es el número de clases

1. Entropía:
2. Error cuadrático medio (MSE) para regresión

El criterio de división busca minimizar estas métricas de impureza, eligiendo la característica y el umbral que mejor separen los datos.

**Ventajas y desventajas**

Los árboles de decisión son fáciles de interpretar y visualizar. Pueden manejar tanto datos numéricos como categóricos sin necesidad de preprocesamiento. Sin embargo, tienen una alta tendencia al sobreajuste si no se controla la profundidad del árbol o el número mínimo de muestras por hoja. Además, pueden ser inestables frente a pequeñas variaciones en los datos.

**Random Forest**

El Random Forest es un modelo de *ensemble learning* basado en la combinación de múltiples árboles de decisión para mejorar la precisión y reducir el sobreajuste. Fue propuesto en 2001 y se ha convertido en uno de los algoritmos más robustos y utilizados en la práctica.

**Funcionamiento**

El Random Forest crea una "colección" o "bosque" de árboles de decisión, cada uno entrenado en una muestra aleatoria del conjunto de datos original (con reemplazo).

1. Muestreo aleatorio con reemplazo: Para cada árbol, se selecciona una muestra aleatoria del conjunto de entrenamiento.
2. Selección aleatoria de características: En cada nodo del árbol, se selecciona un subconjunto aleatorio de características para decidir la mejor división, lo que introduce diversidad entre los árboles.
3. Agregación de resultados:
   1. Clasificación: Se realiza mediante votación mayoritaria entre todos los árboles.
   2. Regresión: La predicción final es el promedio de las salidas de todos los árboles.

**Ventajas y desventajas:** Random Forest ofrece alta precisión y es robusto frente al sobreajuste, gracias a la diversidad introducida en la construcción de los árboles. Además, es resistente al ruido y puede manejar grandes cantidades de datos y características. Sin embargo, su principal desventaja es la menor interpretabilidad en comparación con un único árbol de decisión y su mayor demanda de recursos computacionales y memoria.

**Gradient Boosting**

El Gradient Boosting es otra técnica de *ensemble learning* que mejora el rendimiento de los modelos mediante la combinación secuencial de múltiples árboles de decisión. A diferencia de Random Forest, que construye árboles de manera independiente, el Gradient Boosting entrena los árboles secuencialmente, donde cada nuevo árbol intenta corregir los errores cometidos por el árbol anterior.

**Funcionamiento**

El algoritmo minimiza una función de costo mediante el enfoque de descenso por gradiente.

donde

es el modelo después de iteraciones

es el árbol de decisión entrenado para corregir los errores del modelo anterior

es la tasa de aprendizaje que controla cuánto contribuye cada nuevo árbol al modelo final.

**Entrenamiento**

1. Inicialización: Se empieza con un modelo base simple.
2. Cálculo de residuos: Se calcula el residuo o el gradiente negativo del error para cada observación.
3. Entrenamiento del árbol: Se entrena un nuevo árbol para predecir estos residuos.
4. Actualización del modelo: Se actualiza el modelo sumando la predicción ajustada por la tasa de aprendizaje.
5. Repetición: El proceso se repite hasta alcanzar un número predeterminado de iteraciones o hasta que la mejora en la función de costo sea mínima.

**Variantes de Gradient Boosting**

Existen varias implementaciones del algoritmo clásico de Gradient Boosting que Optimizan el rendimiento y la eficiencia computacional:

* **XGBoost** (Extreme Gradient Boosting): Versión optimizada de Gradient Boosting que incluye mejoras en la velocidad y el rendimiento. Utiliza técnicas de regularización L1 y L2 para reducir el sobreajuste, procesamiento en paralelo y manejo eficiente de los datos faltantes. Implemente un algoritmo llamado **pruning** para eliminar ramas innecesarias en los árboles y mejorar la generalización.
* **LightGBM (**Light Gradient Boosting Machine): Variante que mejora la eficiencia en grandes volúmenes de datos. Utiliza una estrategia de crecimiento de árboles **leaf-wise** en lugar de **level-wise,** lo que permite encontrar divisiones óptimas y rápidas. Es altamente eficiente en memoria y soporta entrenamiento distribuido.

**Ventajas y desventajas**

Gradient Boosting ofrece un buen rendimiento y permite un ajuste preciso a través de múltiples hiperparámetros. Sin embargo, su naturaleza secuencial puede llevar a tiempos más largos de entrenamiento y un riesgo mayor de sobreajuste si no se regula adecuadamente la complejidad del modelo. También requiere una selección de hiperparámetros adecuada para obtener el mejor rendimiento.

2.3.6 Machine Learning: Redes Neuronales

Las Redes Neuronales son un tipo de algoritmo inspirado en la estructura y funcionamiento del cerebro humano. Están compuestas por unidades llamadas neuronas artificiales, organizadas en capas y conectadas entre sí mediante pesos ajustables. En este estudio se van a abordar dos tipos de Redes Neuronales, las Redes Neuronales Artificiales y las Redes Neuronales Recurrentes. Las Redes Neuronales Artificiales (ANN) sirven para la identificación de patrones complejos en los datos, mientras que las Redes Neuronales Recurrentes se utilizan para analizar datos secuenciales o temporales.

**La Neurona Artificial**

Una Neurona Artificial es el componente básico de una red neuronal. Su funcionamiento se basa en un modelo matemático que simula como las neuronas biológicas reciben, procesa y transmiten información. El funcionamiento de una neurona puede representarse de la siguiente forma:

donde

representa las entradas o características del dato

son los pesos sinápticos que modulan la importancia de cada entrada

es un sesgo que permite ajustar el umbral de activación

la combinación lineal de las entradas y los pesos

A continuación, el valor pasa por una función de activación, donde *a* es la salida de la neurona.

**Funciones de activación**

Transformación matemática aplicada a la salida de una neurona en una red neuronal para introducir no linealidad al modelo. Esto permite a la red aprender y representar relaciones complejas en los datos. Las funciones de activación más comunes son las siguientes:

* Función Sigmoide Convierte la salida en un valor entre 0 y 1, útil para problemas de clasificación binaria.
* Función ReLU (Rectified Linear Unit): Introduce no linealidad y es eficiente computacionalmente, evitando problemas de desvanecimiento del gradiente.
* Función Tangente Hiperbólica (tanh): Escala la salida entre -1 y 1, centrándola en cero, para mejorar la convergencia.

**Arquitectura de una ANN (Artificial Neural Network)**

Las ANN están estructuradas en varias capas:

1. Capa de Entrada: Recibe los datos originales
2. Capas Ocultas: Transforman las entradas mediante combinaciones lineales y funciones de activación. El número de capas y neuronas determina la capacidad del modelo para aprender patrones complejos.
3. Capa de Salida: Proporciona la predicción final. En un problema de clasificación binaria, puede tener una sola neurona con activación sigmoide.

Diagrama, Esquemático

El contenido generado por IA puede ser incorrecto.

[Imagen 6](#_6.2_Índices_de)– Red Neuronal Artificial

**Entrenamiento de una ANN – Backpropagation**

El proceso de entrenamiento de una ANN implica ajustar los pesos y sesgos para minimizar una función de pérdida. Esto se realiza mediante un algoritmo conocido como retropropagación (backpropagation) combinado con descenso de gradiente.

1. Cálculo de la pérdida: Se mide la diferencia entre la predicción “” el valor real “*y”* usando una función de pérdida como el Error Cuadrático Medio (**MSE**)
2. Cálculo del gradiente: Se computan las derivadas parciales de la función de pérdida con respecto a cada peso y sesgo.
3. Actualización de los pesos: Se ajustan los pesos en la dirección que minimiza la pérdida:

donde es la tasa de aprendizaje

**Redes Neuronales Recurrentes**

Las Redes Neuronales Recurrentes (RNN) son una extensión de las ANN diseñadas para trabajar con datos secuenciales o temporales. En el contexto de este trabajo, las RNN son útiles para incorporar el historial de rendimiento de los equipos a lo largo del tiempo.

**Funcionamiento de una RNN**

En una RNN, la salida de una neurona no depende solo de la entrada actual, sino también del estado anterior. Esto introduce un mecanismo de memoria que permite a la red mantener información sobre entradas pasadas. Matemáticamente, el estado oculto en el tiempo *t* se define como:

donde

es la entrada en el tiempo *t*

es el estado oculto del paso anterior

son los pesos de entrada y recurrentes

*b* es el sesgo

*f* es la función de activación, comúnmente *tanh* o *ReLU*

**Diagrama, Esquemático

El contenido generado por IA puede ser incorrecto.**

[Imagen 7](#_6.2_Índices_de)– Red Neuronal Recurrente

La salida en cada paso puede ser:

donde *g* es una función de activación adecuada para la tarea, como puede ser *softmax* para clasificación.

**Problemas en RNN**

1. Desvanecimiento de gradiente: Problema causado en secuencias largas donde los gradientes que se calculan para actualizar los pesos pueden hacerse extremadamente pequeños. Esto limita la capacidad de la red para aprender dependencias a largo plazo.
2. Explosión del gradiente: Es lo contrario al desvanecimiento de gradiente. Los gradientes pueden volverse extremadamente grandes haciendo que los pesos crezcan sin control. Esto provoca inestabilidad en el entrenamiento.
3. Dependencias a largo plazo: Las RNN tradicionales son efectivas para capturar relaciones locales o de corto plazo. Sin embargo, fallan al tratar de aprender patrones que dependen de información que ocurrió muchas etapas atrás en la secuencia.

**Variantes de RNN**

1. Long Short-Term Memory (**LSTM**): Red Neuronal Recurrente que aborda específicamente el problema del desvanecimiento del gradiente. La arquitectura de las LSTM introduce una celda de memoria que puede mantener información durante largos periodos de tiempo. Sin embargo, requiere más capacidad computacional.
2. Gated Recurrent Unit **(GRU):** Las GRU son una simplificación de las LSTM. Requiere menos capacidad y complejidad computacional, pero no alcanza la precisión de las LSTM en secuencias largas o complejas.

## 2.4 Trabajos Relacionados

El uso de modelos de aprendizaje automático en la predicción de resultados deportivos ha sido un área de creciente interés en la última década, con aplicaciones que van desde el análisis de rendimiento deportivo hasta la optimización de estrategias en apuestas. A continuación se presentan algunos estudios relevantes que hablan sobre problemáticas similares.

Herbinet (2018) [[10](#_6.1_Bibliografía)], en su trabajo “*Using Machine Learning Techniques to Predict the Outcome of Professional Football Matches”,* explora diferentes métodos de Machine Learning para predecir el marcador de los partidos de fútbol. En su estudio se introduce una métrica llamada *expected goals* (*xG*), que estima el rendimiento de los equipos mediante eventos ocurridos durante el partido. Algunos de los modelos utilizados en este trabajo son:

1. Modelos Generalizados Lineales: para predecir resultados binarios (victoria/derrota).
2. Árboles de Decisión y Random Forest: utilizados para identificar patrones complejos en el rendimiento de los equipos.
3. Redes Neuronales: aplicadas para capturar patrones no lineales en los datos.

Los resultados mostraron que estos modelos alcanzaron precisiones comparables a las utilizadas por las casas de apuestas, mejorando la capacidad predictiva en comparación con métodos estadísticos tradicionales.

Por otro lado, el estudio *"Aplicación de Métodos de Aprendizaje Automático en el Análisis y la Predicción de Resultados Deportivos"* de Soto-Valero (2018) [[11](#_6.1_Bibliografía)] analiza la aplicación de métodos de Machine Learning en el análisis y predicción de resultados deportivos. Este estudio ofrece una revisión exhaustiva del uso de técnicas de Machine Learning en el análisis cuantitativo de datos deportivos. Se destaca el uso de modelos como:

1. Regresión Logística y Árboles de Decisión para la clasificación de resultados competitivos.
2. Redes Neuronales para la evaluación del rendimiento deportivo.
3. Algoritmos de Agrupamiento como K-Means y DBSCAN para la identificación de patrones de comportamiento en el rendimiento de los equipos.

Además, el autor propone una metodología para aplicar estos métodos en el análisis de mercados deportivos, incluyendo las apuestas, lo que resulta directamente relevante para este trabajo.

En un contexto más reciente, Martínez Arias y Marulanda Vélez (2023) realizan una monografía titulada *“Modelo de Clasificación Multiclases para la Predicción de Apuestas Deportivas”* [[12](#_6.1_Bibliografía)]*.* En este estudio los autores se enfocan en la predicción de resultados en la Serie A de Italia. Utilizan diferentes algoritmos de clasificación, evaluando su precisión mediante diversas métricas:

1. Decision Tree Classifier: obtuvo una precisión del 65%, sirviendo como modelo base.
2. Random Forest Classifier: mejoró la precisión al 73%, reduciendo el sobreajuste mediante la combinación de múltiples árboles.
3. Gradient Boosting Classifier y LightGBM: alcanzaron una precisión del 75%, mejorando iterativamente los errores de modelos anteriores.
4. Histogram Gradient Boosting Classifier: logró una precisión del 75%, destacándose por su eficiencia en el procesamiento de grandes volúmenes de datos​.

Los resultados obtenidos evidencian la efectividad de los modelos de ensemble learning (como Random Forest y Gradient Boosting) para mejorar la precisión en la predicción de resultados deportivos.

Los trabajos analizados evidencian el gran potencial del Machine Learning en la predicción de resultados deportivos, permitiendo identificar qué modelos y técnicas ofrecen los mejores resultados. En particular:

* Los modelos de **Ensemble Learning, como Random Forest y Gradient Boosting,** destacan por su alta precisión y robustez frente a datos complejos, siendo capaces de minimizar el sobreajuste y mejorar la generalización.
* Las **Redes Neuronales** demuestran una notable eficacia para capturar patrones no lineales en los datos, aunque requieren mayor capacidad computacional y un cuidadoso ajuste de hiperparámetros.

Estos estudios sirven como base fundamental para el desarrollo del presente TFG que busca evaluar la capacidad de diferentes modelos de Machine Learning en la predicción de resultados deportivos.

# 3. ASPECTOS METODOLÓGICOS

Este capítulo justifica las técnicas empleadas para investigar, escribir y desarrollar la parte práctica del trabajo. Se detallarán las metodologías utilizadas, las razones para su elección y las tecnologías aplicadas. Este apartado también describe las herramientas y tecnologías utilizadas en el desarrollo del trabajo, justificando su selección en base a eficiencia, compatibilidad y experiencia previa.

## 3.1 Metodología

El desarrollo del Trabajo de Fin de Grado (TFG) se organizó en diferentes fases, cada una con objetivos definidos. Esta estructura permitió dividir el trabajo en bloques manejables y facilitar su seguimiento y documentación.

3.1.1 Planificación

Inicialmente, se consideraron varios temas para el TFG. Tras un análisis conjunto con la tutora, se seleccionó este tema por su viabilidad, disponibilidad de datos históricos y posibilidad de aplicar las técnicas de Machine Learning aprendidas durante la carrera. Desde el inicio, se establecieron las siguientes fases:

**1. Planificación:** Se definieron los objetivos del proyecto y se estableció un esquema de trabajo basado en fases. Se llevó a cabo una primera reunión con la tutora para evaluar la viabilidad del tema y se determinó la metodología a seguir.

**2. Investigación:** Se realizó un estudio de trabajos previos y modelos utilizados en la predicción de resultados deportivos mediante Machine Learning. Se identificaron las principales técnicas aplicadas y se seleccionaron aquellas más relevantes para el trabajo.

**3. Adquisición, análisis y procesamiento de datos**: Se recopilaron los datos desde Football-Data.co.uk.[[8]](#footnote-8) Durante esta fase, se realizó una limpieza exhaustiva de los datos para corregir inconsistencias, eliminar valores nulos y estructurar un dataset homogéneo.

**4. Desarrollo del modelo:** Se implementaron y compararon distintos modelos de Machine Learning, iniciando con modelos básicos y avanzando hacia modelos más complejos como Random Forest y técnicas de Boosting. Se utilizaron herramientas como MLflow y Dagshub para registrar experimentos y métricas.

**5. Evaluación y análisis de resultados:** Se analizaron las métricas de rendimiento de los modelos seleccionados y se realizaron comparaciones para determinar la solución más efectiva. Se identificaron patrones en los errores y se evaluaron las limitaciones del modelo aplicado.

A medida que se avanzaba en cada fase, se redactaba la memoria correspondiente a esa etapa, asegurando la coherencia y calidad del documento final. Además, se realizaban revisiones continuas de las secciones previas para garantizar la consistencia. Todas estas fases se explicarán en detalle en el apartado [4. Desarrollo del Trabajo](#_4._DESARROLLO_DEL).

3.1.2 Organización y seguimiento del trabajo

El trabajo se organizó a través de objetivos semanales, permitiendo una monitorización continua del progreso y una reevaluación constante de los siguientes pasos. Esta metodología ayudó a mantener un ritmo de trabajo estable y evitar desviaciones del objetivo principal. Para asegurar un seguimiento efectivo, se emplearon las siguientes tecnologías:

* GitHub para el almacenamiento y control de versiones del código
* Dagshub y MLflow para registrar los experimentos y las métricas de rendimiento de los modelos

Las reuniones con la tutora se realizaron aproximadamente cada mes, aumentando la frecuencia a medida que se acercaba la fecha de entrega. Antes de cada reunión, se enviaban los avances para su revisión, lo que permitió recibir feedback estructurado y discutir los próximos pasos de manera efectiva

Estas herramientas facilitaron la documentación del proceso y aseguraron la reproducibilidad del trabajo. Las tecnologías empleadas se explicarán con detalle en el siguiente apartado [3.2 Tecnologías Empleadas](#_3.2_Tecnologías_Empleadas).

3.1.3 Adaptaciones y ajustes en la metodología

Inicialmente, se consideró implementar tanto la predicción de clases (victoria, empate, derrota) como la predicción de resultados exactos. Sin embargo, esto requería explorar un número elevado de modelos y técnicas, por lo que se decidió enfocar todos los esfuerzos en un solo objetivo para optimizar su rendimiento. Además, la predicción de resultados exactos es altamente complicada, ya que requiere información histórica sobre el rendimiento detallado de los equipos y jugadores, la cual no está disponible para todas las temporadas. Adicionalmente, no se disponía de las cuotas exactas para cada resultado específico, lo que impedía realizar una validación precisa en relación con las casas de apuestas.

## 3.2 Tecnologías Empleadas

Este apartado describe las herramientas y tecnologías utilizadas en el desarrollo del trabajo, justificando su selección en base a eficiencia, compatibilidad y experiencia previa.

3.2.1 Lenguajes y entornos de desarrollo

* Python: Lenguaje principal para la implementación de modelos de Machine Learning y procesamiento de datos.
* Jupyter Notebook: Entorno interactivo utilizado para el desarrollo, análisis exploratorio y pruebas de modelos.

3.2.2 Bibliotecas y Frameworks

* Pandas y NumPy: Manipulación y procesamiento de datos.
* Matplotlib: Visualización de datos y análisis exploratorio.
* Scikit-learn: Implementación de modelos de Machine Learning como Regresión Logística, Random Forest y Gradient Boosting.
* Imbalanced-learn (SMOTE, RandomUnderSampler): Técnicas de balanceo de datos.

3.2.3 Github

GitHub es una plataforma utilizada para el control de versiones y la gestión del código fuente. Permite llevar un registro detallado de los cambios en el código, facilitando el seguimiento de la evolución del proyecto y la posibilidad de volver a versiones anteriores en caso de errores o modificaciones no deseadas. Su uso ayuda a mantener la organización del trabajo, evitar la pérdida de información y colaborar de manera eficiente en proyectos de cualquier escala. Además, al ser accesible desde cualquier dispositivo, permite la gestión remota del código y la integración con otras herramientas utilizadas en el desarrollo de software y ciencia de datos.

3.2.4 MLflow

MLflow es una herramienta utilizada para la gestión y seguimiento de experimentos en Machine Learning. Permite registrar ejecuciones de diferentes modelos, comparar sus métricas de rendimiento y organizar los experimentos de manera eficiente. En este proyecto, se utilizó junto con Dagshub para documentar los experimentos y facilitar la comparación entre modelos con distintas configuraciones.

Dagshub es una plataforma especializada en la gestión de versiones y colaboración en ciencia de datos. Funciona de manera similar a GitHub, pero optimizada para proyectos de Machine Learning, permitiendo almacenar y versionar datasets, modelos y experimentos. Una de sus ventajas es que integra MLflow de manera nativa, lo que facilita el almacenamiento y acceso remoto a los experimentos.

En este caso, el repositorio de Dagshub está directamente vinculado al repositorio del proyecto en GitHub, que actúa como el sistema principal de almacenamiento del código. Esta integración permite que cualquier actualización en GitHub se refleje automáticamente en Dagshub, asegurando que los datos, modelos y experimentos estén correctamente sincronizados y versionados.

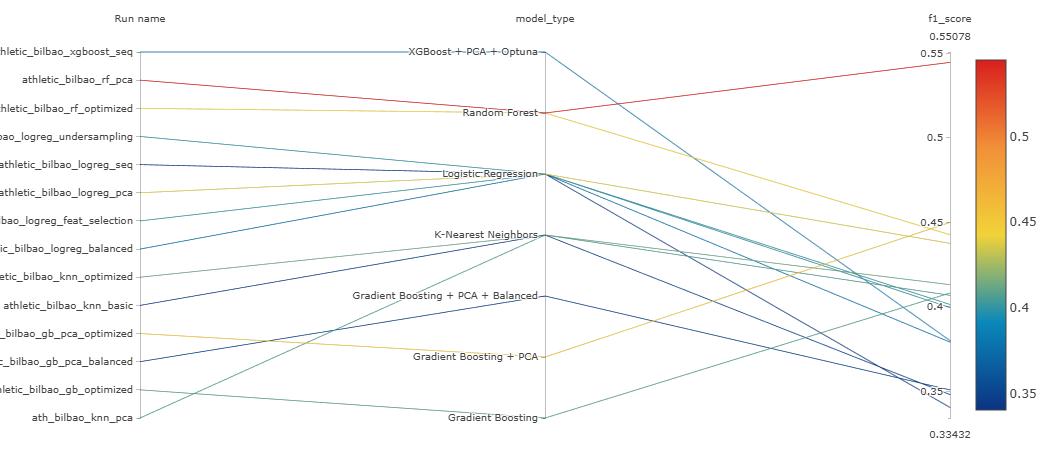
La imagen a continuación muestra la interfaz de MLflow en Dagshub, donde se pueden ver múltiples experimentos registrados. Cada fila representa una ejecución diferente de un modelo con sus respectivas métricas y parámetros. Se asignaron etiquetas a cada experimento para monitorizar el rendimiento por equipos. Permite un registro estructurado de experimentos, evitando pruebas desorganizadas.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

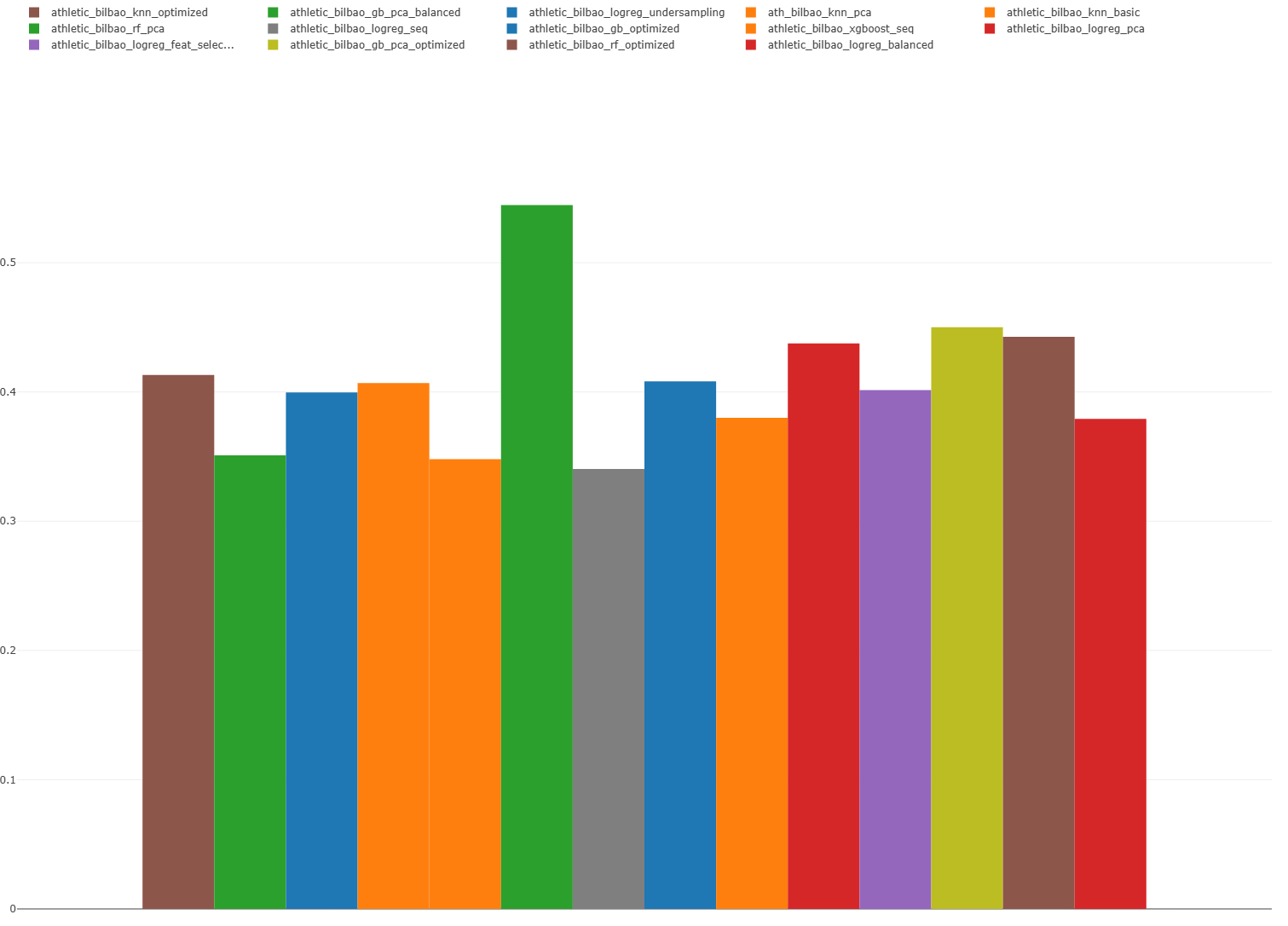
[Imagen 8](#_6.2_Índices_de) – Interfaz de MLflow en Dagshub

MLflow permite realizar comparaciones entre los experimentos seleccionados. Se generan gráficos visuales de las métricas que permiten la comparación del rendimiento de los modelos de manera. La siguiente imagen muestra un diagrama de relaciones entre los modelos y sus métricas.



[Imagen 9](#_6.2_Índices_de) – Grafico de rendimiento que produce MLflow al comparar modelos

MLflow ofrece diferentes tipos de gráficos para realizar la comparación del rendimiento de los modelos. Esta imagen muestra un gráfico de barras del f1\_score de cada uno de los modelos aplicados a un conjunto de datos específico.



[Imagen 10](#_6.2_Índices_de) – Grafico de barras para comparar el rendimiento de los modelos

# 4. DESARROLLO DEL TRABAJO

En este capítulo se detallan las diferentes fases del desarrollo del TFG, describiendo el proceso seguido para alcanzar los objetivos planteados. En particular, en este apartado se abordarán las siguientes fases:

1. **Adquisición, análisis y procesamiento de datos**: Se explicará cómo se obtuvieron los datos necesarios para el desarrollo del modelo, las técnicas empleadas para su limpieza y estructuración, y los criterios utilizados para garantizar la calidad del dataset.
2. **Desarrollo del modelo:** Se describirá la implementación de los distintos modelos de Machine Learning, desde las técnicas más básicas hasta las más avanzadas, así como las herramientas utilizadas para la gestión de experimentos y métricas.

Todo el código desarrollado para este proyecto se encuentra alojado en el repositorio de GitHub: [TFG INSO GitHub](https://github.com/anaigs/tfg_inso_github). Además, los experimentos realizados de cada modelo y sus resultados se pueden consultar en la plataforma de DagsHub: [TFG INSO DagsHub](https://dagshub.com/anaigs/tfg_inso_github).

## 4.1 Adquisición, Análisis y Procesamiento de Datos

En este apartado se describen las distintas etapas seguidas para la adquisición, limpieza y procesamiento de los datos en este trabajo.

Para este proyecto, se recopilaron datos históricos de la liga española LaLiga desde la temporada 2003-04 hasta la última temporada completa, 2023-24. Estos datos fueron obtenidos de la plataforma [Football-Data.co.uk](https://www.football-data.co.uk/spainm.php), una fuente confiable que proporciona información detallada sobre los resultados y estadísticas de múltiples ligas de fútbol en un formato estructurado CSV.

El objetivo de esta fase fue transformar los datos brutos en un conjunto de datos limpio y estructurado, adecuado para el desarrollo del modelo predictivo. Para ello, se llevaron a cabo las siguientes etapas:

1. Recopilación y combinación de datos: Se descargaron múltiples archivos CSV con información de diversas temporadas y se consolidaron en un solo dataset.
2. Exploración de los datos: Se analizaron las características de las variables disponibles, identificando posibles problemas como valores nulos, inconsistencias o redundancias.
3. Limpieza y transformación: Se eliminaron valores irrelevantes, se imputaron datos faltantes y se crearon nuevas variables a partir de las existentes.
4. Generación de datasets específicos: Se crearon datasets centrados en cuatro equipos concretos, filtrando la información relevante para cada uno de ellos.
5. Análisis exploratorio de los datasets específicos: Se examinaron los datasets generados para evaluar su coherencia, distribución y utilidad en el desarrollo del modelo.

En los siguientes apartados se describirá en detalle cada una de estas etapas, comenzando por el origen y descripción de los datos recopilados.

**Notebooks utilizados en esta fase**

Para llevar a cabo el procesamiento y análisis de los datos, se emplearon los siguientes notebooks:

1. clean\_dataset.ipynb: Limpieza y preprocesamiento de datos.
2. analisis\_exploratorio.ipynb: Análisis exploratorio del dataset.
3. generacion\_datasets.ipynb: Creación de datasets específicos por equipo.
4. analisis\_datasets.ipynb: Evaluación de los datasets generados.

Los cuatro datasets generados corresponden a equipos específicos de LaLiga, seleccionados para evaluar su rendimiento de manera individual.

4.1.1 Origen y Descripción de los Datos

Para este trabajo, se ha utilizado como fuente principal la plataforma [Football-Data.co.uk](https://www.football-data.co.uk/spainm.php), que proporciona información detallada sobre los partidos disputados en distintas temporadas, incluyendo resultados, estadísticas y cuotas de apuestas. Esta plataforma ofrece datos tanto de primera división como de segunda división, pero en este estudio solo se analizarán los partidos de primera división (LaLiga).

Se ha elegido la temporada 2003-04 como punto de partida porque es la primera que incluye datos históricos sobre las cuotas de apuestas, una variable clave en este estudio. Estas cuotas reflejan la probabilidad implícita que los mercados de apuestas asignan a los posibles resultados de un partido. En este contexto, se utilizan como una fuente externa de información que puede mejorar la precisión de los modelos de Machine Learning en la predicción de resultados de los partidos.

A continuación, se describen las variables incluidas en los archivos descargados de [Football-Data.co.uk](https://www.football-data.co.uk/spainm.php). Es importante destacar que algunas variables solo están disponibles en ciertas temporadas, por lo que su presencia en el dataset puede variar.

**Información básica del partido:** Contiene datos esenciales que describen el contexto del encuentro.

* Div: División (SP1 = LaLiga)
* Date: Fecha del partido
* Time: Hora del partido
* HomeTeam: Equipo local
* AwayTeam: Equipo visitante

**Resultados del partido:** Incluye información sobre el resultado final y parcial del partido.

* FTHG: Goles del equipo local (Full Time Home Goals)
* FTAG: Goles del equipo visitante (Full Time Away Goals)
* FTR: Resultado del partido (H = Victoria local, D = Empate, A = Victoria visitante)
* HTHG: Goles del equipo local al descanso (Half Time Home Goals)
* HTAG: Goles del equipo visitante al descanso (Half Time Away Goals)
* HTR: Resultado al descanso (H, D, A)

**Estadísticas avanzadas del partido:** Se incluyen diferentes métricas sobre el desempeño de los equipos en el partido. Esta información no está disponible en todas las temporadas.

* HS, AS: Disparos del equipo local y visitante
* HST, AST: Disparos a puerta del equipo local y visitante
* HC, AC: Saques de esquina del equipo local y visitante
* HY, AY: Tarjetas amarillas del equipo local y visitante
* HR, AR: Tarjetas rojas del equipo local y visitante

**Cuotas de apuestas:** Las cuotas de apuestas indican la valoración del mercado sobre los posibles resultados del partido. No todas estan disponibles para cada partido.

1X2 (ganador del partido):

* B365H, B365D, B365A: Cuotas de Bet365 para local, empate y visitante
* PSH, PSD, PSA: Cuotas de Pinnacle
* WHH, WHD, WHA: Cuotas de William Hill
* AvgH, AvgD, AvgA: Cuotas promedio del mercado

Over/Under (Total de goles en el partido):

* B365>2.5, B365<2.5: Cuotas para más/menos de 2.5 goles en Bet365
* Max>2.5, Max<2.5: Máximas cuotas del mercado para más/menos de 2.5 goles
* Avg>2.5, Avg<2.5: Cuotas promedio del mercado

Hándicap asiático:

* B365AHH, B365AHA: Cuotas de Bet365 para hándicap asiático
* BbMxAHH, BbMxAHA: Máximas cuotas del mercado
* AvgAHH, AvgAHA: Cuotas promedio del mercado

Además, existen otras cuotas de diferentes casas de apuestas como **Sporting Odds, Ladbrokes, Gamebookers, Stanleybet, entre otras**. Sin embargo, las mencionadas anteriormente son las más comunes y están presentes en la mayoría de las temporadas. Para más información sobre las variables de los datasets, se puede consultar el archivo de referencia de la plataforma: [Football-Data.co.uk./notes.txt](https://www.football-data.co.uk/notes.txt)

**Estructura y procesamiento del dataset**

Los datos fueron recopilados en archivos **CSV**, con un archivo separado por cada temporada. Para su análisis, se realizó un proceso de integración y limpieza de los datos, que incluyó las siguientes etapas:

* Lectura de archivos de cada temporada desde la carpeta correspondiente.
* Unificación de los datos en un único dataset.
* Eliminación de valores irrelevantes
* Corrección de formatos
* Creación de la variable Season, que permite identificar a qué temporada pertenece cada partido.

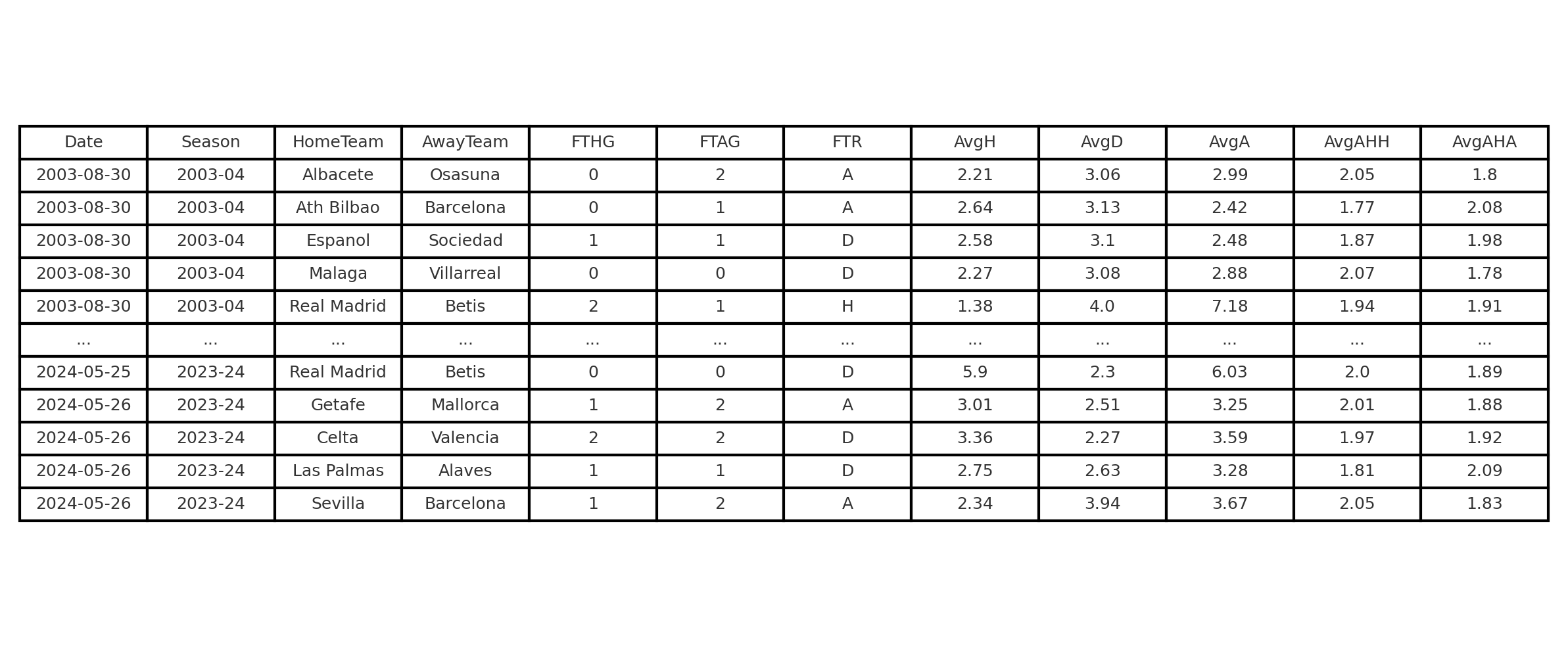
Una vez consolidado el dataset, se aplicaron diversas técnicas de procesamiento para garantizar su calidad y adecuación para el modelo predictivo:

* Selección de variables clave: No se incluyeron estadísticas del partido como remates, posesión o faltas debido a la inconsistencia en la disponibilidad de estos datos, ya que solo algunas temporadas contaban con esta información. Se optó por priorizar variables más consistentes, como los resultados finales de los partidos y las cuotas de apuestas.
* Se eliminaron las variables que contenían información sobre el resultado al descanso (HTHG, HTAG, HTR), ya que la intención del proyecto es predecir el resultado del partido antes de que comience. Estas variables proporcionaban información que solo se conoce una vez iniciado el partido, por lo que no eran adecuadas para la fase de predicción.
* Promedio de cuotas de apuestas: Se calcularon valores promedio (AvgH, AvgD, AvgA) a partir de distintas casas de apuestas, ya que algunos partidos tenían cuotas de ciertas casas y otros de otras, lo que generaba datos incompletos.

Además, se tomó la decisión de incluir solo las cuotas promedio (AvgH, AvgD, AvgA, AvgAHH, AvgAHA) y no todas las cuotas individuales de cada casa de apuestas. Esto se hizo para evitar una sobrecarga de variables en el dataset, reducir la dimensionalidad y enfocarse en las cuotas representativas del mercado en lugar de valores individuales de cada operador.

* Uso de datos de BetBrain: Para algunos partidos, no se contaba con información de todas las casas de apuestas, sino solo con los valores proporcionados por BetBrain, una plataforma de comparación de cuotas que calcula valores promedio basados en múltiples operadores. En estos casos, se usaron los datos de BetBrain para rellenar los valores faltantes.

Finalmente, tras la limpieza y preprocesamiento, el dataset quedó listo para el análisis exploratorio, el cual se abordará en el siguiente apartado. La imagen siguiente muestra la estructura final del dataset tras la limpieza y el preprocesamiento:



[Imagen 8](#_6.2_Índices_de) – Dataset completo de partidos desde la temporada 03/04 hasta 23/24

4.1.2 Análisis Exploratorio de los Datos

Antes de construir los modelos de predicción, se llevó a cabo un análisis exploratorio de los datos para comprender mejor la estructura del dataset, identificar patrones relevantes y evaluar posibles variables que puedan influir en los resultados de los partidos.

**Información general del dataset**

El dataset utilizado en este estudio cubre un total de 21 temporadas de LaLiga, desde 2003-04 hasta 2023-24. Durante este período, LaLiga ha mantenido un formato consistente con **20 equipos por temporada**, lo que equivale a **380 partidos** jugados por temporada (19 partidos de ida y 19 de vuelta para cada equipo). En total, el dataset contiene **7.980 partidos** analizados.

Además, en este periodo han competido un total de **41 equipos** en LaLiga, aunque algunos han tenido una presencia más estable que otros. Equipos como Real Madrid, Barcelona, Atlético de Bilbao y Valencia han disputado todas las temporadas incluidas en el análisis, mientras que otros como Albacete, Xerez o Córdoba han tenido participaciones más limitadas debido a ascensos y descensos.

**Clasificación final por temporada**

Para enriquecer el análisis y mejorar la capacidad predictiva del modelo, se ha generado un dataset con la clasificación final de cada equipo en cada temporada. Esta información permite conocer la posición relativa de los equipos en la competición y puede ser útil para detectar patrones de rendimiento a lo largo del tiempo. El conocimiento de la posición final de los equipos en temporadas anteriores puede ser un factor clave para la predicción de resultados, ya que refleja la fortaleza histórica de un equipo y su desempeño en la liga.

**Análisis de los datos mediante visualizaciones**

A continuación, se presentan los gráficos generados en el análisis exploratorio

**Distribución de goles por partido**: El análisis de los goles anotados en los partidos permite comprender la dinámica de los encuentros en LaLiga.

Estos gráficos muestran la frecuencia de los diferentes resultados en términos de goles anotados en un partido. Se observa que la mayoría de los encuentros terminan con entre 1 y 3 goles, siendo los partidos con 0 goles relativamente poco frecuentes. El boxplot permite visualizar la dispersión de los goles y la presencia de valores atípicos. Se observan algunos partidos con marcadores inusuales (más de 8 goles), pero en general, la mayoría de los encuentros se mantiene en un rango de 1 a 4 goles.

Gráfico, Histograma

El contenido generado por IA puede ser incorrecto.Gráfico, Gráfico de cajas y bigotes

El contenido generado por IA puede ser incorrecto.

[Imagen 9](#_6.2_Índices_de) – Distribución de goles por partido

**Relación entre goles a favor y goles en contra:** El rendimiento ofensivo y defensivo de los equipos a lo largo de las temporadas permite identificar patrones de juego.

Este gráfico muestra la relación entre los goles marcados y los goles recibidos por equipo. Se observa que equipos como Real Madrid y Barcelona tienen un alto número de goles a favor, mientras que equipos más débiles tienden a recibir más goles de los que marcan.

Gráfico, Gráfico de dispersión

El contenido generado por IA puede ser incorrecto.

[Imagen 10](#_6.2_Índices_de) – Relación entre goles a favor y goles en contra

**Rendimiento en casa vs. fuera:** El análisis de los partidos jugados en casa y fuera permite evaluar la influencia del factor local en los resultados.

Para evaluar la influencia del factor local, se analizaron varias métricas clave. Un gráfico de barras apiladas muestra que los equipos ganan más partidos cuando juegan en casa en comparación con los encuentros como visitantes. Esto sugiere que el factor local puede ser determinante en los resultados de los partidos. Además, un gráfico de comparación de goles en casa vs. fuera revela que la mayoría de los equipos marcan más goles en casa que fuera, aunque algunos equipos presentan un rendimiento equilibrado en ambos escenarios. Por último, un diagrama de dispersión analiza la relación entre la cantidad de victorias en casa y las victorias fuera. Se observa que, si bien algunos equipos son sólidos en ambos escenarios, otros dependen más del rendimiento en su estadio. Estos hallazgos refuerzan la importancia de considerar la localía en la construcción del modelo.

Gráfico, Gráfico de barras

El contenido generado por IA puede ser incorrecto.Gráfico, Gráfico de dispersión

El contenido generado por IA puede ser incorrecto.

Gráfico, Gráfico de barras, Histograma

El contenido generado por IA puede ser incorrecto.

[Imagen 11](#_6.2_Índices_de) - Rendimiento en casa vs. fuera

**Variabilidad del rendimiento entre temporadas:** El rendimiento de los equipos varía con el tiempo, ya sea por cambios en la plantilla, estrategias o rendimiento general en la liga.

Este gráfico muestra la cantidad de equipos que han cambiado de posición en la clasificación de una temporada a otra. Se observa que la mayoría de los equipos cambian en promedio 3.32 posiciones por temporada, lo que indica un nivel moderado de estabilidad en LaLiga.

Gráfico, Histograma

El contenido generado por IA puede ser incorrecto.

[Imagen 12](#_6.2_Índices_de) – Distribución de cambios de posición entre temporadas en LaLiga

4.1.3 Generación de Datasets por Equipos

Después de haber limpiado y transformado los datos, se creó una serie de datasets específicos por equipo. La finalidad de esta etapa es disponer de conjuntos de datos enfocados en equipos concretos, permitiendo un análisis más detallado y la construcción de modelos predictivos adaptados a cada club.

Se seleccionaron cuatro equipos de LaLiga para este estudio:

* Real Madrid
* Barcelona
* Valencia
* Athletic Club de Bilbao

Estos equipos fueron elegidos por su participación constante en LaLiga a lo largo de las temporadas y por el interés de analizar sus tendencias de rendimiento a lo largo del tiempo. Además, el rendimiento de estos equipos en LaLiga varía entre unos y otros. Real Madrid y Barcelona suelen liderar las clasificaciones con un alto porcentaje de victorias, mientras que Valencia y Athletic Club de Bilbao han tenido resultados más variables a lo largo de las temporadas, lo que permite evaluar diferentes tipos de desempeño en el modelo.

**Estructura del dataset generado por equipo**

Cada dataset creado contiene información detallada sobre los partidos disputados por el equipo en cuestión, considerando tanto su rendimiento general como el de su rival. A continuación, se describen las variables incluidas en estos datasets

**Información general del partido**: Contexto general del partido y de los equipos:

* season: Temporada en la que se jugó el partido.
* date: Fecha en la que se jugó el partido.
* team: Nombre del equipo analizado.
* rival\_team: Nombre del equipo rival.
* home\_adv: Indica si el equipo analizado jugó como local (1) o visitante (0).
* last\_season\_team: Posición final del equipo analizado en la temporada anterior.
* last\_season\_rival: Posición final del equipo rival en la temporada anterior.

**Rendimiento reciente del equipo analizado:** Rendimiento del equipo analizado en los últimos 10 partidos de la temporada:

* pct\_wins: Porcentaje de victorias en los últimos 10 partidos de la misma temporada.
* avg\_goals\_scored: Promedio de goles anotados en los últimos 10 partidos.
* avg\_goals\_received: Promedio de goles recibidos en los últimos 10 partidos.
* goal\_difference: Diferencia de goles en los últimos 10 partidos.

**Rendimiento reciente del equipo rival:** Rendimiento del equipo rival en sus últimos 10 partidos de la temporada:

* pct\_wins\_rival: Porcentaje de victorias del equipo rival en los últimos 10 partidos.
* avg\_goals\_scored\_rival: Promedio de goles anotados por el equipo rival en los últimos 10 partidos.
* avg\_goals\_received\_rival: Promedio de goles recibidos por el equipo rival en los últimos 10 partidos.
* goal\_difference\_rival: Diferencia de goles del equipo rival en los últimos 10 partidos.

**Historial entre ambos equipos:** Rendimiento del equipo analizado frente al rival en los últimos 5 enfrentamientos directos:

* pct\_wins\_vs\_rival: Porcentaje de victorias del equipo analizado contra ese rival en los últimos 5 enfrentamientos.
* avg\_goals\_scored\_vs\_rival: Total de goles anotados al rival en los últimos 5 enfrentamientos.
* avg\_goals\_received\_vs\_rival: Total de goles recibidos del rival en los últimos 5 enfrentamientos.
* goal\_difference\_vs\_rival: Diferencia de goles en los últimos 5 enfrentamientos.

**Cuotas de apuestas:** Probabilidad implícita de los resultados según el mercado de apuestas:

* AvgWin: Promedio de cuotas para la victoria del equipo analizado.
* AvgLoss: Promedio de cuotas para la derrota del equipo analizado.
* AvgDraw: Promedio de cuotas para el empate.
* AvgAHWin: Promedio de cuotas para la victoria del equipo analizado con hándicap asiático.
* AvgAHLoss: Promedio de cuotas para la derrota del equipo analizado con hándicap asiático.

**Información del resultado del partido**: Resultado del partido desde la perspectiva del equipo analizado:

* goals\_team: Número de goles anotados por el equipo analizado.
* goals\_rival: Número de goles anotados por el equipo rival.
* result: Resultado del partido desde la perspectiva del equipo analizado:
  + Victoria: 1
  + Empate: 0
  + Derrota: -1

**Proceso de generación de los datasets por quipo**

La generación de datasets específicos por equipo permite analizar el rendimiento individual de cada club y facilita la creación de modelos predictivos personalizados. Gracias a la inclusión de estadísticas de rendimiento reciente, enfrentamientos directos y cuotas de apuestas, estos datasets proporcionan información clave que puede mejorar la precisión en la predicción de los resultados de los partidos.

Para construir estos datasets, se siguieron los siguientes pasos:

1. Filtrado de partidos
   1. Se seleccionaron los partidos donde el equipo analizado participó, ya sea como local o visitante.
2. Asignación de información clave
   1. Se definieron las variables principales, incluyendo la temporada, el equipo analizado, el rival y si el equipo jugaba en casa o fuera.
3. Cálculo de estadísticas de rendimiento
   1. Se calcularon las métricas de los últimos 10 partidos para el equipo y su rival, asegurando que solo se tomaran en cuenta partidos anteriores a la fecha del partido analizado.
   2. Si no se habían jugado aún 10 partidos en la temporada actual, se utilizaron los partidos disponibles, evitando datos de temporadas anteriores para minimizar el impacto de cambios en la plantilla y reflejar el rendimiento real del equipo en ese mismo año.
   3. También se calcularon las estadísticas de enfrentamientos directos entre ambos equipos en los últimos 5 partidos disputados.
4. Ajuste de las cuotas de apuestas
   1. Se aseguraron de que las cuotas siempre correspondieran a la perspectiva del equipo analizado, independientemente de si jugaba como local o visitante.
5. Imputación de valores faltantes
   1. Algunas cuotas de apuestas presentaban valores nulos, por lo que se estableció un valor neutral (1.0) para evitar la pérdida de información.

**Importancia de estas métricas**

Las métricas seleccionadas en la construcción de estos datasets juegan un papel clave en la predicción de los resultados de los partidos:

* El rendimiento reciente del equipo y del rival ayuda a identificar tendencias y evaluar si un equipo atraviesa una racha positiva o negativa.
* El historial entre ambos equipos proporciona información sobre si hay un patrón en los enfrentamientos previos. Algunos equipos pueden tener un historial favorable contra ciertos rivales, independientemente de su desempeño general en la liga.
* Las cuotas de apuestas ofrecen una estimación de la probabilidad de cada resultado desde la perspectiva del mercado, lo que puede mejorar la precisión del modelo al incorporar información externa basada en análisis de expertos.

4.1.4 Análisis de Datasets por Equipos

# 5. CONCLUSIONES

Junto con el capítulo de Introducción, al que debe dar respuesta, este capítulo debería poder leerse con autonomía del resto del documento, por lo que debe ser claro, sintético y didáctico. Debe incluir sugerencias sobre trabajos futuros que pudiesen continuar con el trabajo iniciado en el TFG. Una conclusión personal final no está fuera de lugar (¿qué ha significado el TFG para el alumno/a?).

# 6. REFERENCIAS

Todas las imágenes, citas al pie y referencias del TFG quedan registradas en este apartado siguiendo la normativa APA7.

## 6.1 Bibliografía

[1] KPMG Asesores S.L. (2023). Impacto socioeconómico del fútbol profesional en España. KPMG.

[2] Aoki, R. Assunção, R. Vaz de Melo, P. (2017) Luck is Hard to Beat: The Dificulty of Sports Prediction. Department of Computer Science UFMG.

[3] Cavus, M. Biecek, P. (2022). Explainable Expected Goal Models for Performance Analysis in Football Analytics. Warsaw University of Technology. Eskisehir Technical University.

[4] Spearman, W. (2028). Beyond Expected Goals. Sports Analytics Conference. MIT Sloan, Boston MA.

[5] Manassé Galekwa, R. Tshimula, J. M. Tajeuna, E. G. Kyandoghere, K. Systematic Review of Machine Learning in Sports Betting: Techniques, Challenges, and Future Directions.

[6] Lora Gutiérrez, J. D., & Macias Burbano, Y. A. (2022). Ineficiencias en el mercado de apuestas del fútbol: Los efectos del COVID-19 [Tesis de grado, Universidad Icesi].

[7] Van Raaij, V. (2019). Favorite-longshot bias in European Football betting market: Differences between popular and non-popular football competitions. Radboud Universiteit.

[8] Martín Domínguez, D. (2013). Análisis de resultados deportivos y estimación implícita de probabilidades: Fútbol. UC3M

[9] A. Jung,“Machine Learning: The Basics,” Springer, Singapore, 2022.

[10] Herbinet, C. (2018). Using machine learning techniques to predict the outcome of professional football matches [Undergraduate project report, Imperial College London]. Imperial College London Repository.

[11] Soto-Valero, C. (2018). Aplicación de métodos de aprendizaje automático en el análisis y la predicción de resultados deportivos. Retos: Nuevas Tendencias en Educación Física, Deporte y Recreación.

[12] Martínez Arias, L. M., & Marulanda Vélez, S. (2023). Modelo de clasificación multiclases para la predicción de apuestas deportivas [Trabajo de grado especialización, Universidad de Antioquia]. Repositorio Institucional Universidad de Antioquia.

## 6.2 Referencias Teóricas Consultadas

Neupert, T., Fischer, M. H., Greplova, E., Choo, K., & Denner, M. M. (2021). Introduction to Machine Learning for the Sciences.

Ghojogh, B., & Crowley, M. (2019). The theory behind overfitting, cross validation, regularization, bagging, and boosting: Tutorial.

Montgomery, D. C., Peck, E. A., & Vining, G. G. (2012). Introduction to Linear Regression Analysis. John Wiley & Sons.

Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). Applied Logistic Regression (Vol. 398). John Wiley & Sons.

Izza, Y. Ignatiev, A. Marques-Silva, J. ANITI, (2020). On Explaining Decision Trees. Univ. Toulouse

Cunningham, P. Delany, S.J. (2020). K-Nearest Neighbour Classifiers 2nd Edition (with Python examples). University College Dublin. Technological University Dublin

Louppe, G. (2015). Understanding Random Forests. University of Liège

He, T., Zhao, H., Chu, X., Liu, Z., Liu, X., & Chen, G. (2019). Gradient Boosting: A Survey. Point Zero One Technology. Fordham University. Imperial College London. Massachusetts Institute of Technology.

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting System.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... & Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. Advances in Neural Information Processing Systems

Sherstinski, A. (2023) Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network. Elsevier Journal. Physica D: Nonlinear Phenomena. (Vol. 404)

Schmidt, R. M. (2019) Recurrent Neural Networks (RNNs): A gentle Introduction and Overview. Eberhard-Karls-University Tübingen.

Grossi, E., Buscema, M. (2008) Introduction to artificial neural networks. European Journal of Gastroenterology & Hepatology.

Qamar, R., Zardari, B. A. (2023). Artificial Neural Networks: An Overview. Vol. (2023). Mesopotamian Journal of Computer Science

## 6.2 Índices de imágenes

[1] Captura de pantalla. Aplicación Móvil. Plataforma de resultados Flashscore: Apuestas 1X2

[2] Captura de pantalla. Aplicación Móvil. Plataforma de resultados Flashscore: Hándicap Asiático

[3] Bagnato, J. I. (2017, 12 diciembre). Representación gráfica de los conceptos de overfitting y underfitting. Aprende Machine Learning. https://www.aprendemachinelearning.com/que-es-overfitting-y-underfitting-y-como-solucionarlo/

[4] Izco, F. (s.f.). Matriz de confusión. BDC-POC. https://bookdown.org/f\_izco/BDC-POC/metricas.html

[5] Díaz Sosa, M. L. (s.f.). Representación de la función sigmoide. GeoGebra. https://www.geogebra.org/m/MpJeZcMB

[6] Qamar, R., & Zardari, B. A. (2023). Artificial Neural Networks: An Overview. Mesopotamian Journal of Computer Science, 2023. https://doi.org/10.58496/MJCSC/2023/015

[7] Qamar, R., & Zardari, B. A. (2023). Artificial Neural Networks: An Overview. Mesopotamian Journal of Computer Science, 2023. <https://doi.org/10.58496/MJCSC/2023/015>

[8] Dataset completo de partidos desde la temporada 03/04 hasta 23/24. Generado en:

Repositorio: <https://github.com/anaigs/tfg_inso_github>

Ruta: code/data\_cleaning/2-analisis\_exploratorio.ipynb

[9] Distribución de goles por partido. Generado en:

Repositorio: <https://github.com/anaigs/tfg_inso_github>

Ruta: code/data\_cleaning/2-analisis\_exploratorio.ipynb

[10] Relación entre goles a favor y goles en contra. Generado en:

Repositorio: <https://github.com/anaigs/tfg_inso_github>

Ruta: code/data\_cleaning/2-analisis\_exploratorio.ipynb

[11] Rendimiento en casa vs. fuera . Generado en:

Repositorio: <https://github.com/anaigs/tfg_inso_github>

Ruta: code/data\_cleaning/2-analisis\_exploratorio.ipynb

[12] Distribución de cambios de posición entre temporadas en LaLiga. Generado en:

Repositorio: <https://github.com/anaigs/tfg_inso_github>

Ruta: code/data\_cleaning/2-analisis\_exploratorio.ipynb

# ANEXOS

En los anexos se adjuntará todo el material citado que ocupe demasiado espacio para ser incluido en el cuerpo de texto o que pueda desviar la atención del lector. Cualquier gráfico, estadística, línea de código, texto de referencia, entrevista o escolio técnico que no esté directamente vinculado con el desarrollo y las conclusiones del TFG pero que se quiera añadir como material de consulta es susceptible de incluirse en un anexo.

1. ESPN staff (2014, 11 junio). Soccer Power Index explained. https://www.espn.com/ [↑](#footnote-ref-1)
2. La evolución histórica de las apuestas deportivas: de los juegos antiguos a la era digital. (2023, 21 junio). Fox Sports. https://www.foxsports.com. [↑](#footnote-ref-2)
3. Fernández, R. (2025, 13 febrero). Las apuestas y los juegos de azar en el mundo: Datos estadísticos. https://es.statista.com/ [↑](#footnote-ref-3)
4. Mouzo, J. (2024, 24 octubre). La amenaza de llevar un casino en el bolsillo: 80 millones de adultos sufren adicción al juego. El País. https://elpais.com [↑](#footnote-ref-4)
5. Zuppello, M. (2024, 30 septiembre). Adicción a las apuestas online en Brasil. Infobae. https://www.infobae.com [↑](#footnote-ref-5)
6. (2024, abril) El Pilón. Los favoritos de los aficionados: Los deportes más populares para apostar. https://elpilon.com.co [↑](#footnote-ref-6)
7. Villar, G. (2023, 13 octubre). La cara B de la riqueza que genera el fútbol: el 43% del gasto de los aficionados va a las apuestas online. Relevo. https://www.relevo.com [↑](#footnote-ref-7)
8. Football-Data.co.uk. (2024). Spain football data files. https://www.football-data.co.uk/spainm.php [↑](#footnote-ref-8)