



UNIVERSIDAD ALFONSO X EL SABIO

GRADO EN INGENIERÍA MATEMÁTICA

*Utilizando IA para predecir
dentro del campo de fútbol*

Alba Bernal Rodríguez

Marzo 2024

Índice

1	Introducción	3
2	Metodología	3
2.1	Recopilación y preparación de datos	4
2.1.1	Web scraping	4
2.1.2	Tratamiento datos	8
2.2	Análisis y gráficas	9
2.2.1	Metodología	10
2.2.2	Resultados	10
2.2.3	Discusión y futuras direcciones para el Modelo Predictivo	13
2.2.4	Conclusiones	13
2.3	Machine Learning	14
2.3.1	Aprendizaje supervisado	15
2.3.1.1	Regresion Lineal	15
2.3.1.2	Clasificación	17
2.3.2	Aprendizaje no supervisado	22
2.3.2.1	Clustering	22
2.3.3	Aprendizaje por refuerzo	24
2.3.3.1	Cadena de Markov	24
2.3.4	Aprendizaje profundo	25
2.3.4.1	CNN (Red neuronal convolucional)	25
2.3.4.2	TL (Transferencia de aprendizaje)	28
2.3.4.3	DNN (Red neuronal profunda)	29
3	Bibliografía:	30

Índice de figuras

1	Ejemplo de histórico de jugadores	4
2	Ejemplo descripcion general equipos	6
3	Ejemplo histórico equipos	7
4	Ejemplo partidos	8
5	Gráfica goles vs puntos	11
6	Gráfica goles en contra por partido de los porteros destacados . .	11
7	Gráfica equipos con más apariciones en la final	12
8	Porcentaje de Victorias de Local, Visitante y Empates	12
9	Machine Learning	14
10	Matriz correlación	16
11	Diagrama dispersion	17
12	Histograma Residuos	17
13	Matriz confusión	18
14	Matriz confusión mejorada	19
15	Variables mayor impacto Random Forest	21

16	Clusters equipos	22
17	Clusters equipos	23
18	Ejemplo imagenes usadas en entrenamiento modelo	26
19	Evaluacion modelo CNN	27
20	Evaluacion modelo CNN con transferencia	29

1. Introducción

El presente proyecto es parte de la asignatura de Inteligencia Artificial y representa una oportunidad para aplicar y ampliar los conocimientos adquiridos en el ámbito del aprendizaje automático y la inteligencia artificial. En este proyecto, me enfocaré en un desafío apasionante: la predicción de resultados en uno de los eventos deportivos más emocionantes del mundo, la UEFA Champions League.

La predicción de resultados en el fútbol es un campo de estudio fascinante y desafiante, que combina la emoción del deporte con la complejidad de los datos y la incertidumbre inherente a cada partido. En este trabajo, me propongo analizar cómo podemos utilizar una variedad de técnicas de modelado, desde métodos tradicionales como la regresión lineal hasta enfoques más avanzados como las redes neuronales convolucionales (CNN), para predecir con precisión los resultados de los partidos de la Champions League.

Comenzaré mi análisis con un enfoque riguroso basado en la regresión lineal, utilizando datos históricos de partidos anteriores y una amplia gama de características relevantes, como el rendimiento del equipo, la calidad de los jugadores y otros factores influyentes. A partir de este punto de partida, exploraré cómo podemos mejorar y refinar nuestras predicciones mediante técnicas más avanzadas de aprendizaje automático.

A lo largo de este proyecto, también consideraré la importancia del preprocesamiento de datos, la selección de características y la evaluación del rendimiento del modelo. Estos aspectos son fundamentales para construir sistemas de predicción robustos y confiables que puedan capturar con precisión la complejidad y la variabilidad presentes en los datos del fútbol.

Al finalizar este proyecto, no solo habré adquirido un conjunto valioso de habilidades y conocimientos en el campo del aprendizaje automático y la inteligencia artificial, sino que también habré contribuido al avance del conocimiento en el emocionante y competitivo mundo del fútbol profesional. Espero que este trabajo no solo sea un hito en mi trayectoria académica, sino también una contribución significativa al campo de la predicción de resultados en el deporte más popular del mundo.

2. Metodología

En esta sección, describiré detalladamente el enfoque metodológico que seguiré en mi proyecto para abordar el desafío de predecir los resultados de la UEFA Champions League. Este enfoque se divide en varias etapas clave, que incluyen la recopilación y preparación de datos, la selección y entrenamiento

de modelos de aprendizaje automático, la evaluación del rendimiento de los modelos y la interpretación de los resultados obtenidos.

2.1. Recopilación y preparación de datos

2.1.1 Web scraping

En primer lugar, comenzamos la recopilación de datos para nuestro análisis de la UEFA Champions League, una tarea que resultó ser desafiante debido a las restricciones de acceso en fuentes confiables como el sitio web oficial de la UEFA y plataformas de estadísticas conocidas como SofaScore. Ante estas limitaciones, exploramos diversas fuentes en línea y desarrollamos técnicas de web scraping personalizadas utilizando Python y bibliotecas como BeautifulSoup y Selenium. Esto nos permitió recopilar una amplia gama de datos sobre partidos pasados, equipos participantes, resultados y otras variables relevantes en un formato estructurado que pudiera ser utilizado en nuestro análisis posterior. Además, Selenium fue fundamental para extraer información precisa de tablas dinámicas, las cuales son actualizadas en tiempo real o se cargan dinámicamente mediante JavaScript, asegurando así la integridad de nuestros datos.

Para nuestro modelo recolectamos datos sobre los jugadores, de los partidos y uno que es general, que contine los partidos pero tambien jugador destacado de cada uno.

Jugadores:

Estadísticas estándar del jugador 2017-2018 Champions League

Las estadísticas no incluyen rondas clasificatorias. ☒ Ocultar a los no clasificados para las estadísticas de calificación. [Glosario](#) [Cambiar a estadísticas "por90"](#)

[Desplázate hacia la derecha para obtener más estadísticas y; cambiar a vista de pantalla panorámica](#)














						Tiempo Jugado			Rendimiento							Expectativa				Progresión			Porcentaje										
RL	Jugador	País	Posc	Equipo	Edad	Nacimiento	PJ	Titular	Min	90 s	Gls.	Ass.	G+A	G-TP	TP	TPint	TA	TR	xG	npG	xAG	npAG	xG+	npG+	xAG+	npAG+	Gls.	Ass.	G+A	G-TP	G+A-TP	Porcentaje	
1	Vincent Aboubakar		CMR	DL		Porto	25	1992	6	6	495	5.5	5	2	7	5	0	0	0	2.8	2.8	1.0	3.8	4	7	38	0.91	0.36	1.27	0.91	1.27	0.91	
2	Marcos Acuña		ARG	DL,DF		Sporting CP	25	1991	5	5	433	4.8	0	1	1	0	0	2	0	0.0	0.0	0.5	0.5	0.5	12	13	19	0.00	0.21	0.21	0.00	0.21	0.21
3	Tosin Adarabioye		ENG	DF		Manchester City	19	1997	2	1	91	1.0	0	0	0	0	0	0	0	0.0	0.0	0.0	0.0	0.0	0	7	0	0.00	0.00	0.00	0.00	0.00	0.00
4	Adriano		BRA	DF		Besiktas	32	1984	6	5	486	5.4	0	0	0	0	0	1	0	0.0	0.0	0.1	0.2	14	25	11	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	Luz Adriano		BRA	DL,CC		Spartak Moscow	30	1987	6	6	540	6.0	1	0	1	1	0	0	1	0	1.0	1.0	0.9	1.9	8	16	35	0.17	0.00	0.17	0.17	0.17	0.17
6	Anis Agallo		ALB	DF		Qarabağ FK	34	1982	4	4	360	4.0	0	0	0	0	0	0	0	0.0	0.0	0.1	0.2	6	12	14	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	Sergio Agüero		ARG	DL		Manchester City	29	1988	7	5	440	4.9	4	1	5	3	1	2	0	3.9	2.4	0.4	2.8	17	6	37	0.82	0.20	1.02	0.61	0.82	0.82	0.82
8	Karim El Ahmadi		MAR	CC		Feversonod	32	1985	4	4	360	4.0	0	0	0	0	0	3	0	0.1	0.1	0.3	0.4	1	13	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	Valon Ahmed		ALB	CC,DL		NK Maribor	22	1994	4	3	226	2.5	0	0	0	0	0	0	0	0.2	0.2	0.5	0.8	9	6	12	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Figura 1: Ejemplo de un histórico de jugadores del 2017-2018. Fuente: fbref.com

Las columnas que nos proporciona esta página son:

1. 'Temporada': Representa la temporada en la que se jugó el partido.
2. ' ': Representa el número de jugador.
3. 'Jugador': Representa el nombre del jugador.
4. 'País': Representa el país del jugador.
5. 'Posc': Representa la posición del jugador.

6. 'Equipo': Representa el equipo al que pertenece el jugador.
7. 'Edad': Representa la edad del jugador.
8. 'Nacimiento': Representa la fecha de nacimiento del jugador.
9. 'PJ': Representa el número de partidos jugados por el jugador.
10. 'Titular': Indica si el jugador fue titular en el partido.
11. 'Mín 90 s': Representa los minutos jugados por el jugador.
12. 'Gls.': Representa el número de goles marcados por el jugador.
13. 'Ass': Representa el número de asistencias realizadas por el jugador.
14. 'G+A': Representa la suma de goles y asistencias del jugador.
15. 'G-TP': Representa el número de goles marcados desde el punto de penalti por el jugador.
16. 'TP': Representa el número de penaltis lanzados por el jugador.
17. 'TPint': Representa el número de penaltis convertidos por el jugador.
18. 'TA': Representa el número de tarjetas amarillas recibidas por el jugador.
19. 'TR': Representa el número de tarjetas rojas recibidas por el jugador.
20. 'xG': Representa el valor esperado de goles del jugador.
21. 'npxG': Representa el valor esperado de goles sin contar los goles de penalti del jugador.
22. 'xAG': Representa el valor esperado de goles en contra del jugador.
23. 'npxG+xAG': Representa la suma del valor esperado de goles sin contar los goles de penalti y el valor esperado de goles en contra del jugador.
24. 'PrgC': Representa el número de pases completados por el jugador.
25. 'PrgP': Representa el número de pases intentados por el jugador.
26. 'PrgR': Representa el porcentaje de pases completados por el jugador.
27. 'Gls90.': Representa el número de goles por cada 90 minutos jugados por el jugador.
28. 'Ast90': Representa el número de asistencias por cada 90 minutos jugados por el jugador.
29. 'G+A90': Representa la suma de goles y asistencias por cada 90 minutos jugados por el jugador.
30. 'G-TP90': Representa el número de goles marcados desde el punto de penalti por cada 90 minutos jugados por el jugador.

31. 'G+A-TP90': Representa la suma de goles y asistencias sin contar los goles de penalti por cada 90 minutos jugados por el jugador.
32. 'xG90': Representa el valor esperado de goles por cada 90 minutos jugados por el jugador.
33. 'xAG90': Representa el valor esperado de goles en contra por cada 90 minutos jugados por el jugador.
34. 'xG+xAG90': Representa la suma del valor esperado de goles y el valor esperado de goles en contra por cada 90 minutos jugados por el jugador.
35. 'npxG90': Representa el valor esperado de goles sin contar los goles de penalti por cada 90 minutos jugados por el jugador.
36. 'npxG+xAG90': Representa la suma del valor esperado de goles sin contar los goles de penalti y el valor esperado de goles en contra por cada 90 minutos jugados por el jugador.
37. 'Partidos': Representa el número de partidos jugados por el jugador.

General:

Estadísticas estándar del jugador 2017-2018 Champions League

Las estadísticas no incluyen rondas clasificatorias. ☒ Ocultar a los no clasificados para las estadísticas de calificación. [Glosario](#) [Cambiar a estadísticas "por90"](#)

Desplázate hacia la derecha para obtener más estadísticas y: [cambiar a vista de pantalla panorámica](#) ▶











						Tiempo Jugado		Rendimiento								Expectativa				Progresión			Porcentaje								
RL	Jugador	País	Posc	Equipo	Edad	Nacimiento	PJ	Titular	Min	90 s	Gls.	Ass.	G+A	G-TP	TP	TA	TR	xG	npxG	xAG	npxG+xAG	PrgC	PrgP	PrgR	Gls.	Ass	G+A	G-TP	G+A-TP		
1	Vincent Aboubakar		DL		25	1992	6	6	495	5.5	5	2	7	5	0	0	0	0	2.8	2.8	1.0	3.8	4	7	38	0.91	0.36	1.27	0.91	1.27	
2	Marcos Acuña		DL,DF		25	1991	5	5	433	4.8	0	1	1	0	0	0	2	0	0.0	0.0	0.5	0.5	12	13	19	0.00	0.21	0.21	0.00	0.21	
3	Tosin Adarabioye		DF		19	1997	2	1	91	1.0	0	0	0	0	0	0	0	0	0.0	0.0	0.0	0.0	0	7	0	0.00	0.00	0.00	0.00	0.00	
4	Adriano		DF		32	1984	6	5	486	5.4	0	0	0	0	0	0	1	0	0.0	0.0	0.1	0.1	0.2	14	25	11	0.00	0.00	0.00	0.00	0.00
5	Luz Adrian		DL,CC		30	1987	6	6	540	6.0	1	0	1	1	0	0	1	0	1.0	1.0	0.9	1.9	8	16	35	0.17	0.00	0.17	0.17	0.17	
6	Anis Agall		DF		34	1982	4	4	360	4.0	0	0	0	0	0	0	0	0	0.0	0.0	0.1	0.1	0.2	6	12	14	0.00	0.00	0.00	0.00	0.00
7	Sergio Agüero		DL		29	1988	7	5	440	4.9	4	1	5	3	1	2	0	0	3.9	2.4	0.4	2.8	17	6	37	0.82	0.20	1.02	0.61	0.82	
8	Karim El Ahmadi		CC		32	1985	4	4	360	4.0	0	0	0	0	0	0	3	0	0.1	0.1	0.3	0.4	1	13	1	0.00	0.00	0.00	0.00	0.00	
9	Valon Ahmed		CC,DL		22	1994	4	3	226	2.5	0	0	0	0	0	0	0	0	0.2	0.2	0.5	0.8	9	6	12	0.00	0.00	0.00	0.00	0.00	

Figura 2: Ejemplo descripción general rendimiento Champions League 2017-2018. Fuente: fbref.com

Esta tabla cuenta con las siguientes columnas:

1. 'Season': Temporada del equipo.
2. 'Rk': Clasificación del equipo en la temporada.
3. 'Squad': Nombre del equipo.
4. 'MP': Partidos jugados.
5. 'W': Partidos ganados.
6. 'D': Partidos empatados.
7. 'L': Partidos perdidos.
8. 'GF': Goles a favor.

9. 'GA': Goles en contra.
10. 'GD': Diferencia de goles (GF - GA).
11. 'Pts': Puntos obtenidos.
12. 'Attendance': Asistencia promedio en los partidos.
13. 'Top Team Scorer': Máximo goleador del equipo.
14. 'Goalkeeper': Portero del equipo.
15. 'Top Team Scorer Goals': Cantidad de goles anotados por el máximo goleador del equipo.

Equipos:

Scores & Fixtures 2017-2018 Champions League

Glossary

All Rounds	Qualifying rounds	Group stage	Knockout phase											
Round	Wk	Day	Date	Time	Home	xG	Score	xG	Away	Attendance	Venue	Referee	Match Report	Notes
Group stage	1	Tue	2017-09-12	19:45 (20:45)	Chelsea	2.0	6-0	0.3	Qarabag FK	41,150	Stamford Bridge	Tasos Sidiropoulos	Match Report	
				19:45 (20:45)	Celtic	0.4	0-5	2.5	Paris S-G	57,562	Celtic Park	Daniele Orsato	Match Report	
				19:45 (20:45)	Manchester Utd	3.4	2-0	0.6	Basel	73,854	Old Trafford	Ruddy Buquet	Match Report	
				19:45 (20:45)	Benfica	2.1	1-2	1.6	CSKA Moscow	38,323	Estádio do Sport Lisboa e Benfica	Alberto Undiano	Match Report	
				20:45	Roma	0.5	0-0	2.4	Atlético Madrid	36,064	Stadio Olimpico	Milorad Mažić	Match Report	

Figura 3: Ejemplo histórico de equipos participantes en la Champions League 2017-2018. Fuente: fbref.com

Las columnas que tenemos son:

1. 'Season': Temporada del equipo.
2. 'Round': Ronda de la competición.
3. 'Day': Día en el que se juega el partido.
4. 'Date': Fecha del partido.
5. 'Time': Hora del partido.
6. 'Home': Equipo local.
7. 'Score': Puntuación (goles).
8. 'Away': Equipo visitante.
9. 'Attendance': Asistencia promedio en los partidos.
10. 'Venue': Lugar donde se lleva a cabo el partido.
11. 'Referee': Árbitro del partido.
12. 'Match Report': Reporte del partido.
13. 'Notes': Notas adicionales.

Partidos:

Round	Wk	Day	Date	Time	Home	xG	Score	xG	Away	Attendance	Venue	Referee	Match Report
Group stage	1	Tue	2023-09-19	18:45	Milan 	1.9	0-0	0.3	Newcastle Utd 	65,695	Stadio Giuseppe Meazza	José Sánchez	Match Report
				18:45	Young Boys 	0.5	1-3	2.6	RB Leipzig 	31,500	Stadion Wankdorf	Enea Jorgji	Match Report
				20:00 (21:00)	Manchester City 	4.1	3-1	0.8	Red Star 	50,204	Etihad Stadium	João Pinheiro	Match Report
				21:00	Paris S-G 	2.4	2-0	0.7	Dortmund 	47,379	Parc des Princes	Jesús Gil	Match Report

Figura 4: Ejemplo de los equipos participantes en la Champions League que se enfrentan en el 2023-2024. Fuente: fbref.com

Las columnas que tenemos son:

1. 'Season': que muestra la temporada
2. 'Round': Fase de la competición
3. 'Date': Día de la semana en la que se jugó el partido
4. 'Time': Hora en la que comenzó el partido
5. 'Home': Equipo local
6. 'Score': Puntuación
7. 'Away': Equipo visitante
8. 'Attendance': espectadores que acudieron al estadio
9. 'Venue': Estadio donde tuvo lugar el partido
10. 'Referee': Árbitro
11. 'Math Report': un reportaje del partido
12. 'Notes': notas adicionales del partido

2.1.2 Tratamiento datos

Una vez recopilados los datos, nos enfrentamos al desafío de limpiarlos para prepararlos para la fusión con otros conjuntos de datos. Nuestro objetivo principal fue convertir un conjunto de datos brutos y desordenados en uno coherente y listo para el análisis, lo que incluyó realizar ajustes en columnas y filas para garantizar la estabilidad de la fuente de datos. Esto nos permitió integrar nuestros datos de manera efectiva con otros conjuntos de datos para análisis más amplios y completos.

Desafíos y Soluciones

Durante el proceso de limpieza de datos, nos encontramos con varios obstáculos que exigían atención y resolución inmediata:

Al inicio, nos enfrentamos a la presencia de valores nulos en columnas y filas, así como a datos incompletos. Estas inconsistencias fueron tratadas mediante la identificación y eliminación de registros vacíos o su llenado apropiado.

Una de las irregularidades más notables fue la variabilidad en los formatos de puntuaciones de equipos. En lugar del estándar "1-2", encontramos formatos como "(2)1-2(4)". Para normalizar estos datos, aplicamos técnicas de procesamiento de texto, eliminando caracteres no numéricos y estandarizando la representación de los resultados.

Además, la presencia de banderas junto a los nombres de los equipos dificultaba la estructura de los datos. Optamos por transformar las banderas en prefijos de países, lo que permitió la creación de una nueva columna llamada `Country` para indicar la ubicación geográfica de cada equipo.

Por último, algunos datos se presentaban en forma categórica, lo que requería su conversión a valores numéricos para el análisis estadístico. Utilizamos métodos de codificación para realizar esta conversión de manera adecuada, garantizando la coherencia y la utilidad de los datos para análisis posteriores.

Datos limpios

Después de completar el proceso de limpieza, obtenemos tres conjuntos de datos coherentes y listos para su análisis:

1. `'datos_champions_limpio.csv'`: Dataframe con información general sobre la Champions League de distintos años.
2. `'overall_limpio.csv'`: Dataframe con información adicional que contine datos de equipos y partidos.
3. `'jugadores_limpio.csv'`: Dataframe que contiene datos específicos sobre jugadores.
4. `'partidos_limpio.csv'`: Dataframe que contiene datos específicos sobre los partidos históricos, mostrando los quipos que se han enfrentado durante años.
5. `'partidos_2023-2024_limpio.csv'`: Dataframe que contiene los datos de los partidos de la temporada actual.

2.2. Análisis y gráficas

En este apartado, se realiza análisis preliminar detallado utilizando datos históricos y de la última temporada de la Champions League, con el objetivo de

identificar patrones y estadísticas claves. Este análisis sienta las bases para futuras aplicaciones de modelos predictivos que buscan estimar el equipo ganador del torneo.

2.2.1 Metodología

Para llevar a cabo este análisis he utilizado herramientas de programación en **Python**, incluyendo las bibliotecas **Pandas** para el manejo de datos y **Matplotlib/Seaborn** para visualizaciones. Los resultados de este análisis se utilizarán para seleccionar las variables más relevantes que alimentarán los modelos predictivos.

Se investigaron tres áreas clave:

- **Rendimiento de Equipos en la Última Temporada:** Evaluación de la efectividad ofensiva y defensiva mediante el análisis de puntos y la diferencia de goles.
- **Tendencias Históricas de las Fases Finales:** Identificación de equipos con consistencia en alcanzar etapas avanzadas del torneo.
- **Desempeño de Jugadores Clave:** Análisis del impacto de goleadores y porteros en el éxito del equipo.

2.2.2 Resultados

Los análisis preliminares sugieren que el Manchester City, el Bayern Munich y el Real Madrid poseen atributos significativos que podrían contribuir a su éxito en la competición. Estos hallazgos son cruciales para la selección de variables en el modelado predictivo.

Algunas gráficas obtenidas en el análisis son:

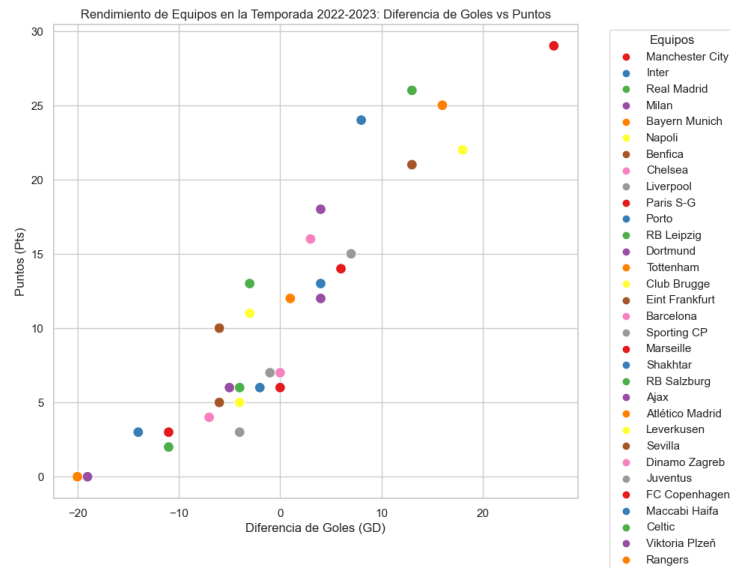


Figura 5: Gráfica que muestra la diferencia de goles y puntos obtenidos por cada equipo temporada 2022-2023. Fuente: Creacion propia

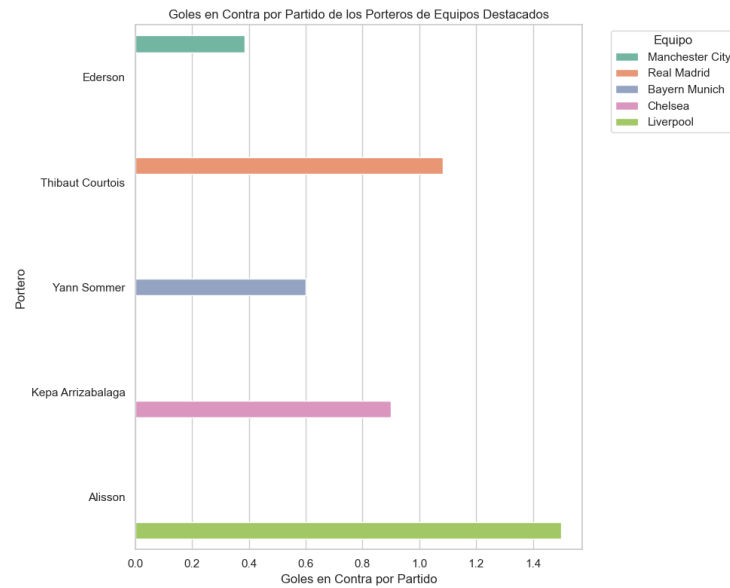


Figura 6: Gráfica que muestra cuántos goles por partido se han marcado a cada portero. Fuente: Creación propia

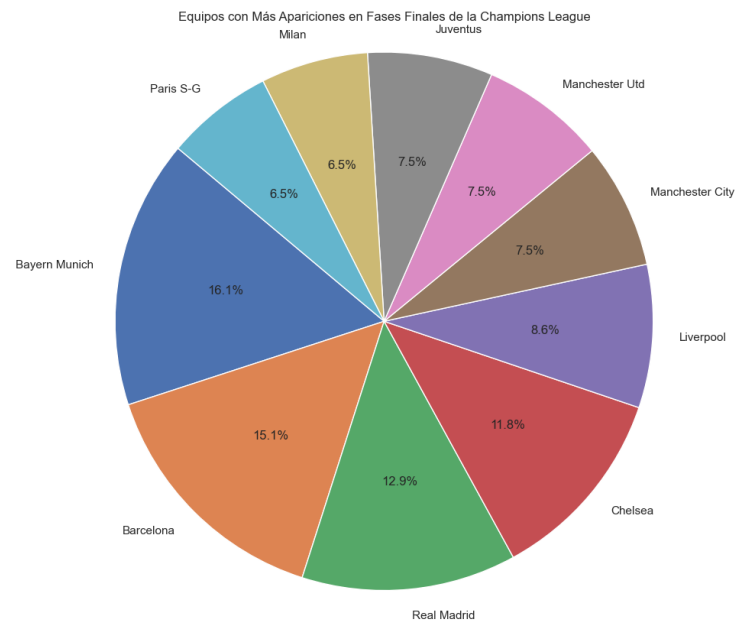


Figura 7: Gráfica que muestra los equipos que han parecido con mayor frecuencia en la final. Fuente: Creación propia

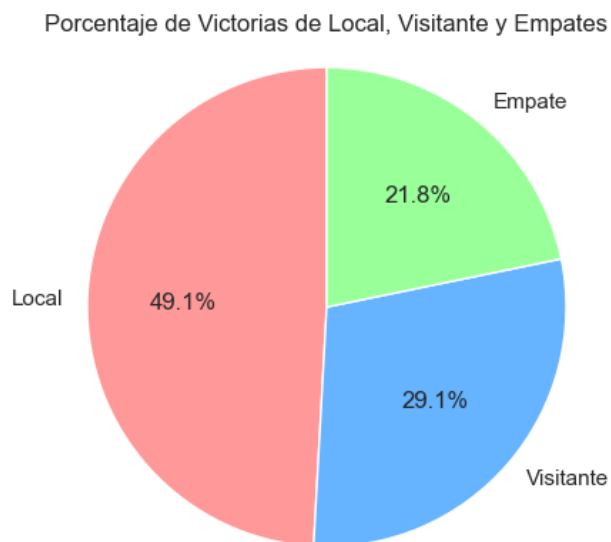


Figura 8: Gráfica que muestra Porcentaje de Victorias de Local, Visitante y Empates. Fuente: Creación propia

2.2.3 Discusión y futuras direcciones para el Modelo Predictivo

Los resultados obtenidos de este análisis preliminar indican factores clave que influyen en el éxito de los equipos en la Champions League. Estos factores incluyen no solo el rendimiento en la última temporada y la experiencia en fases finales, sino también la contribución de jugadores individuales.

Se propone la utilización de modelos predictivos, tales como regresión lineal, árboles de decisión o redes neuronales, para estimar las probabilidades de victoria de los equipos. Las variables seleccionadas en este análisis preliminar servirán como entradas para estos modelos. El enfoque futuro se centrará en la implementación y validación de estos modelos predictivos, con el objetivo de desarrollar un sistema robusto para predecir el ganador de la Champions League.

2.2.4 Conclusiones

Este análisis preliminar ha identificado variables críticas que serán fundamentales en la construcción de modelos predictivos para estimar el ganador de la Champions League. El siguiente paso consistirá en la aplicación de técnicas de modelado predictivo para validar la importancia de estas variables y mejorar la precisión de las predicciones.

2.3. Machine Learning

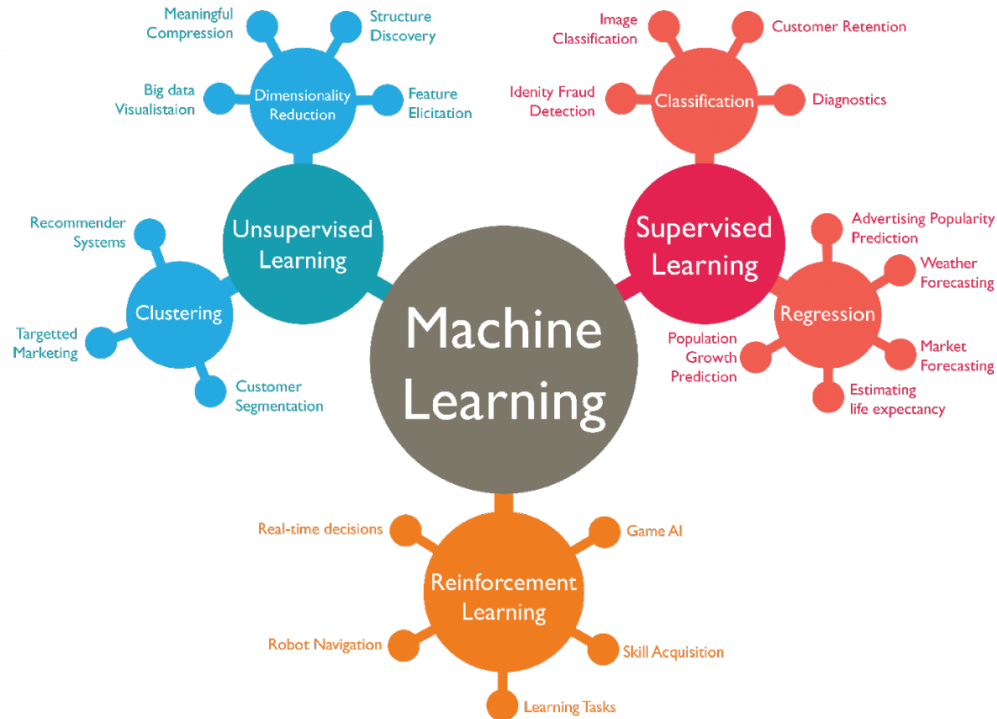


Figura 9: Imagen que muestra los distintos tipos de machine learning. Fuente: <https://mpost.io/es/glossary/machine-learning>

Tras abordar detalladamente la exhaustiva recolección, limpieza y análisis de datos, es hora de sumergirse en la fase de preparación de los modelos predictivos. En este punto, es crucial familiarizarse con el concepto de aprendizaje automático (machine learning) y explorar las diversas categorías que existen.

El **Machine Learning** es una disciplina del campo de la Inteligencia Artificial que, a través de algoritmos, dota a los ordenadores de la capacidad de identificar patrones en datos masivos y elaborar predicciones (análisis predictivo). Este aprendizaje permite a los computadores realizar tareas específicas de forma autónoma, es decir, sin necesidad de ser programados.

Existen varios tipos de aprendizaje automático, cada uno con sus propias características y aplicaciones:

1. **Aprendizaje Supervisado:** En este enfoque, el modelo se entrena utilizando **datos etiquetados**, es decir, datos que tienen una respuesta co-

nocida. El objetivo es aprender una función que mapee las entradas a las salidas deseadas. Ejemplos de algoritmos de aprendizaje supervisado incluyen regresión lineal, regresión logística, y los algoritmos de clasificación como Support Vector Machines (SVM) y árboles de decisión.

2. **Aprendizaje No Supervisado:** Aquí, el modelo se entrena utilizando **datos sin etiquetar**, y el objetivo es encontrar patrones o estructuras intrínsecas en los datos. Los algoritmos de aprendizaje no supervisado incluyen la clustering (agrupamiento), donde el objetivo es dividir los datos en grupos homogéneos, y la reducción de dimensionalidad, que busca representar los datos en un espacio de menor dimensión manteniendo la mayor cantidad posible de información.
3. **Aprendizaje por Refuerzo:** Este tipo de aprendizaje implica que un agente interactúe con un ambiente dinámico en el que debe aprender a tomar decisiones secuenciales. El agente recibe retroalimentación en forma de recompensas o castigos en función de las acciones que realiza. El objetivo es aprender una política óptima que maximice la recompensa acumulada a lo largo del tiempo.

Cada tipo de aprendizaje automático tiene sus propias ventajas y desafíos, y la elección del enfoque adecuado depende de la naturaleza del problema y de los datos disponibles. En nuestro análisis de la Champions League, exploraremos cómo aplicar estos conceptos para predecir resultados de partidos, identificar patrones de juego y entender mejor el rendimiento de los equipos participantes.

2.3.1 Aprendizaje supervisado

2.3.1.1 Regresión Lineal

El proceso de modelado comienza con la carga de los datos históricos de los equipos participantes utilizando pandas, una biblioteca de Python que facilita la manipulación y análisis de datos. Esta etapa inicial es crucial para asegurar que la base de datos esté correctamente estructurada para los análisis subsiguientes.

Tras la carga de datos, se realiza un Análisis Exploratorio de Datos (EDA) para investigar las relaciones entre las variables. Un componente esencial del EDA es la construcción de una matriz de correlación, que se visualiza mediante un mapa de calor usando seaborn y matplotlib. Esta matriz revela cómo variables como goles a favor (GF), goles en contra (GA), diferencia de goles (GD) y puntos acumulados (Pts) están interrelacionadas, proporcionando insights para seleccionar las características más influyentes para el modelo predictivo.

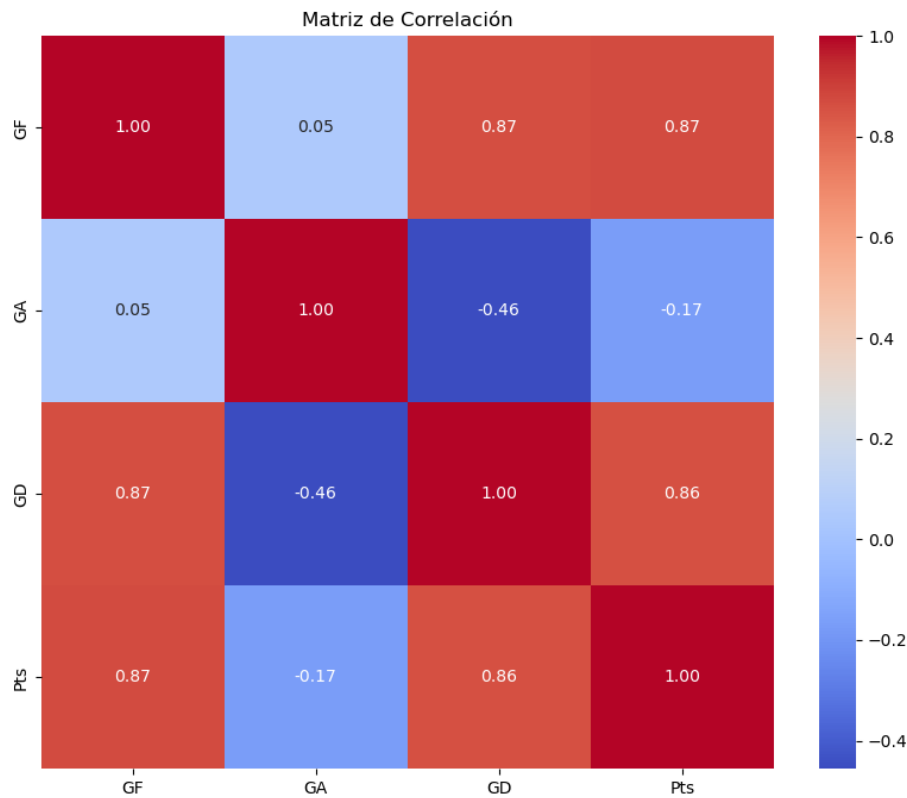


Figura 10: Matriz de correlación que muestra como se relacionan las distintas variables. Fuente: Creación propia

La selección de características basada en la matriz de correlación guía la construcción del modelo de regresión lineal con sklearn, una biblioteca de aprendizaje automático de Python. Se divide el dataset en un conjunto de entrenamiento y otro de prueba, lo que permite no solo entrenar el modelo de manera eficiente sino también evaluar su precisión y generalización fuera de la muestra de entrenamiento. La regresión lineal se elige por su capacidad para modelar relaciones lineales y su interpretabilidad, que son adecuadas para entender y comunicar cómo las variables de entrada afectan a los puntos de un equipo.

La evaluación del modelo se realiza a través del coeficiente de determinación R^2 , que cuantifica cuánta variabilidad en los puntos puede ser explicada por el modelo a través de las variables seleccionadas. Además, se visualizan las predicciones del modelo en comparación con los valores reales a través de un diagrama de dispersión, y se examina la distribución de los errores o residuos utilizando un histograma para evaluar si los errores se distribuyen normalmente alrededor de cero, lo cual es un indicador de un buen ajuste del modelo.

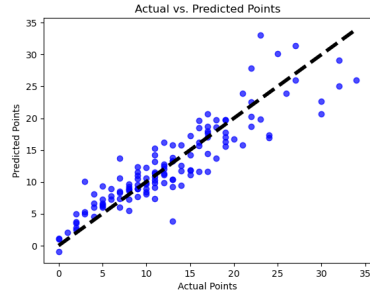


Figura 11: Diagrama dispersión.
Fuente: Creación propia

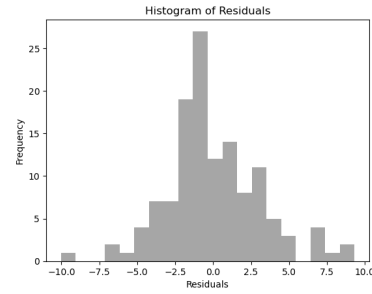


Figura 12: Histograma Residuos.
Fuente: Creación propia

Finalmente, se aplica el modelo entrenado para hacer predicciones sobre la temporada actual de la Champions League, demostrando la utilidad práctica del modelo en un entorno real. Los resultados predichos se integran en el conjunto de datos actual, permitiendo una comparación directa entre los puntos proyectados y los puntos reales obtenidos por los equipos, lo que puede ser útil para las partes interesadas para la toma de decisiones basadas en datos.

Squad	Pts	Predicted Points
Real Madrid	24.0	20.673648
Bayern Munich	23.0	18.228985
Dortmund	18.0	15.549854
Paris S-G	17.0	17.618680
Manchester City	26.0	25.656073
Atlético Madrid	20.0	21.190854
Barcelona	19.0	18.135887
Arsenal	17.0	19.497866
Porto	15.0	16.442898
Inter	15.0	11.836310

Cuadro 1: Equipos y Puntos

2.3.1.2 Clasificación

Tras realizar el modelo de Regresión Lineal pasamos ahora al de Clasificación. Inicialmente, realicé la carga y el preprocesamiento de los datos utilizando Python y bibliotecas como pandas, con datos provenientes de archivos como *jugadores_limpio.csv* y *overall_limpio.csv*.

Al inspeccionar los datos, procedí a transformar la columna *Rk* en *overall_df* en una variable objetivo binaria, clasificando a los equipos que alcanzaron las

finales ('W' y 'F') con un 1, y los demás con un 0. Además, seleccioné variables predictoras relevantes como 'MP', 'W', 'D', 'L', 'GF', 'GA', 'GD', y 'Pts'.

Implementé un modelo de regresión logística utilizando *scikit-learn*, dividiendo los datos en conjuntos de entrenamiento y prueba en una proporción de 70:30 y entrenando el modelo con un máximo de 1000 iteraciones. Los resultados iniciales mostraron una precisión del 96.35 %, evaluada mediante exactitud, matriz de confusión, y la métrica AUC-ROC, indicando una alta capacidad del modelo para predecir correctamente los resultados de los partidos.

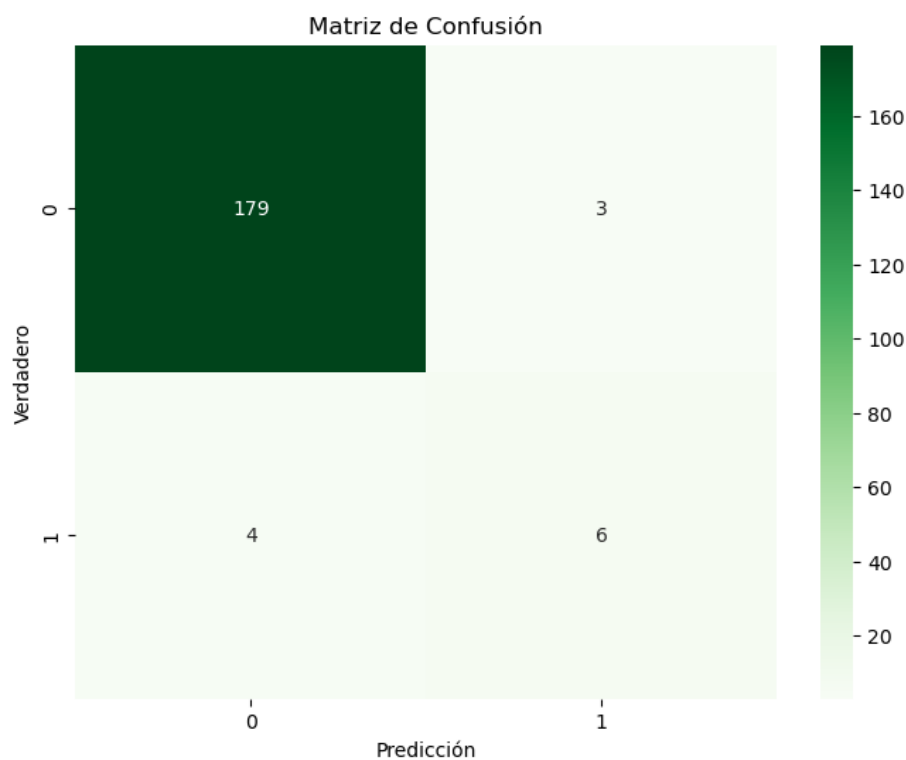


Figura 13: Matriz de Confusión Fuente: Creación propia

Para mejorar el modelo, realicé una optimización de hiperparámetros con *GridSearchCV*, ajustando el coeficiente de regularización C y el algoritmo de solución, lo que mejoró la precisión hasta el 96.86 %. Aplicando el modelo optimizado a los datos de la temporada 2023-2024, predije las probabilidades de que los equipos alcanzaran la final basándome en su rendimiento hasta la fecha, reflejando la incertidumbre competitiva.

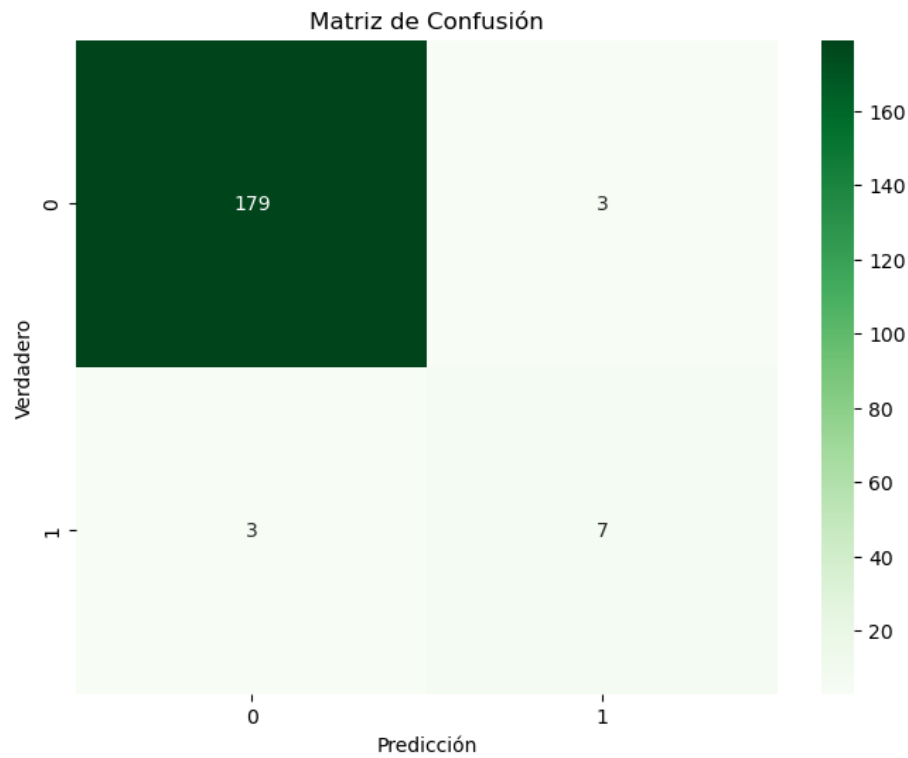


Figura 14: Matriz de Confusión con la mejora aplicada Fuente: Creación propia

Adicionalmente, desarrollé un modelo de Random Forest, combinando datos de rendimiento del equipo con estadísticas de jugadores. Este modelo fue reentrenado con solo datos de equipo debido a problemas de dimensionalidad con nuevos datos, alcanzando una precisión del 88 %. Las predicciones para la temporada actual y el análisis de la importancia de las características revelaron que los puntos y las victorias eran los factores más determinantes del éxito en las competiciones.

N.º	Equipo	Rango Predicho
0	Manchester City	W
1	Real Madrid	QF
2	Bayern Munich	QF
3	Atlético Madrid	R16
4	Arsenal	R16
5	Barcelona	R16
6	Dortmund	R16
7	Paris S-G	R16
8	Porto	R16
9	Inter	R16
10	RB Leipzig	R16
11	Lazio	R16
12	Real Sociedad	R16
13	Napoli	R16
14	PSV Eindhoven	R16
15	FC Copenhagen	GR
16	Shakhtar	GR
17	Milan	GR
18	Lens	GR
19	Feyenoord	GR
20	Newcastle Utd	GR
21	Galatasaray	GR
22	Manchester Utd	GR
23	Benfica	GR
...		
28	Antwerp	GR
29	Union Berlin	GR
30	Sevilla	GR
31	Red Star	GR

Cuadro 2: Clasificación predicha de los equipos

Con los datos actualizados, nuestra predicción cambió y obtuvimos lo siguiente:

Equipo	Probabilidad de Ganar
Real Madrid	0.04
Bayern Munich	0.01
Dortmund	0.00
Paris S-G	0.00

Cuadro 3: Predicción de ganar de varios equipos de fútbol obtenido el 24/04/2023. Fuente: Creación propia

Como vemos hay una incoherencia en nuestra predicción que es que presenta a posibles ganadores a dos equipos que antes de la fase final se enfrentan entre sí lo que quiere decir que uno queda eliminado y por lo tanto su probabilidad entonces de ganar debería ser igual a 0. Esto se debe a que el modelo con el que estamos trabajando no tiene en cuenta los quipos que se enfrentan entre sí. Por lo tanto, es una mejora que se podría incluir.

Finalmente vimos cuales son las características más importantes y de mayor impacto en el modelo de Random Forest

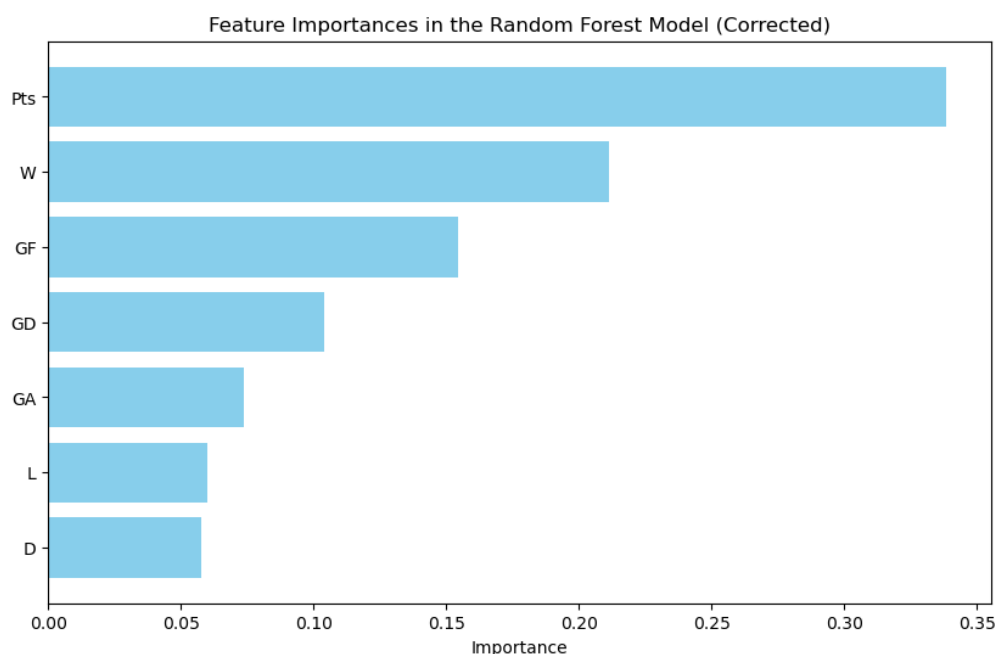


Figura 15: Gráfica que muestra las variables con mayor impacto dentro del modelo de Random Forest. Fuente: Creación propia

Esta visualización nos ayuda a entender qué factores considera el modelo más predictivos del éxito de un equipo en la Champions League. Los puntos y las victorias, que son indicadores directos del éxito en los partidos, naturalmente lideran en importancia, seguidos de medidas de rendimiento ofensivo y defensivo como los goles a favor y la diferencia de goles.

2.3.2 Aprendizaje no supervisado

2.3.2.1 Clustering

Para realizar este modelo hemos realizado dos distintos: uno con equipos y otro con los jugadores, con el objetivo de agrupar a los jugadores en cuatro clusters basados en sus características de juego (portero, defensa, central, delantero) y a los equipos en tres clusters basados en su rendimiento general, permitiendo así un análisis detallado del rendimiento individual y colectivo

Para el clustering de jugadores, comenzamos cargando y explorando los datos para asegurarnos de que estén limpios y listos para el análisis. Seleccionamos las columnas numéricas relevantes y normalizamos los datos para preparar la aplicación del algoritmo K-means. Determinamos el número óptimo de clusters utilizando métodos como el codo (elbow method). Una vez determinados los clusters, visualizamos los resultados aplicando PCA (Análisis de Componentes Principales) para reducir la dimensionalidad y facilitar la visualización en gráficos de dispersión. Estos gráficos nos permiten ver cómo se distribuyen los jugadores en los diferentes clusters, identificando patrones y características comunes.

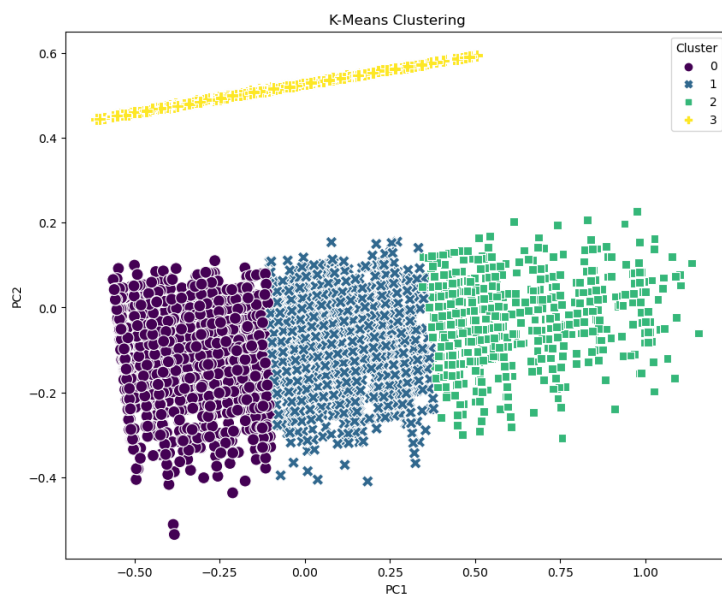


Figura 16: Clusters de los jugadores. Fuente: Creación propia

Por otro lado, en el análisis de los equipos de la Champions League, cargamos los datos correspondientes y realizamos una selección similar de características numéricas relevantes, como partidos jugados, victorias, empates, derrotas, goles a favor, goles en contra, diferencia de goles y puntos. Normalizamos estos datos

y aplicamos nuevamente el algoritmo K-means, esta vez para agrupar a los equipos en tres clusters. Utilizamos PCA para visualizar estos clusters en un espacio bidimensional. Los resultados mostraron tres clusters distintos: el primer cluster agrupa a equipos con el menor rendimiento general, con menos partidos ganados y una diferencia de goles negativa; el segundo cluster representa a los equipos con el mejor desempeño, con el mayor número de victorias y una diferencia de goles positivamente alta; el tercer cluster incluye equipos con un rendimiento intermedio.

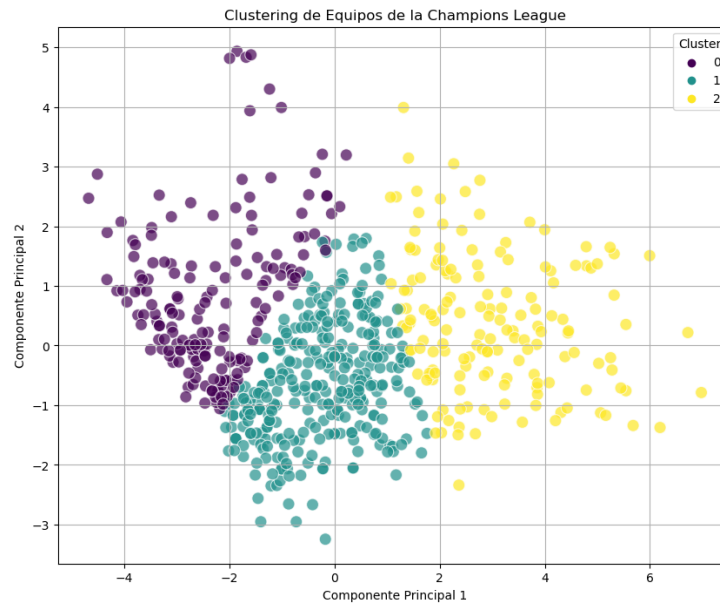


Figura 17: Clusters de los equipos. Fuente: Creación propia

Los gráficos generados incluyen el gráfico de codo para determinar el número óptimo de clusters, gráficos de dispersión para visualizar la distribución de los clusters y diagramas de caja para comparar las distribuciones de características específicas entre los clusters. También se pueden incluir matrices de confusión y métricas de evaluación si se emplean técnicas de validación cruzada o se comparan diferentes métodos de clustering.

En resumen, este análisis proporciona una visión integral del rendimiento en la competición, tanto a nivel individual como colectivo, al agrupar y analizar las características de los jugadores y equipos. Esto ofrece insights valiosos que pueden ser útiles para el análisis del rendimiento y la toma de decisiones estratégicas en el contexto deportivo.

2.3.3 Aprendizaje por refuerzo

2.3.3.1 Cadena de Markov

En el proyecto de investigación, se implementó un modelo de Cadena de Markov para predecir el ganador de la Champions League utilizando datos históricos de partidos y la temporada actual. Inicialmente, se cargaron y exploraron los datos de temporadas anteriores para identificar los equipos, fechas, y resultados de partidos. El preprocesamiento de estos datos implicó la extracción de goles de cada equipo de la columna 'Score' y la clasificación de los resultados en tres estados posibles: victoria local, empate y victoria visitante.

Para construir la matriz de transición, se definieron estos estados y se calculó la probabilidad de transición de un resultado a otro basándose en la frecuencia de resultados consecutivos. La matriz resultante representó las probabilidades de pasar de un estado a otro, lo que es fundamental en la aplicación de Cadenas de Markov, donde la probabilidad de un evento futuro depende únicamente del estado actual.

-Matriz transición

$$\begin{bmatrix} 0,224 & 0,512 & 0,264 \\ 0,20212766 & 0,5177305 & 0,28014184 \\ 0,24096386 & 0,42771084 & 0,3313253 \end{bmatrix}$$

Para la temporada actual, se cargaron los datos de los partidos jugados y los pendientes. Utilizando la matriz de transición, se simularon los resultados de los partidos no jugados, asumiendo el último resultado conocido como el estado inicial para cada equipo en un partido no jugado. Posteriormente, se simuló la progresión del torneo desde cuartos de final hasta la final, utilizando los resultados simulados y reales para avanzar equipos.

-Tabla de predicciones

Home	Away	Simulated_Result
Bayern Munich	Real Madrid	2
Dortmund	Paris S-G	1
Paris S-G	Dortmund	2
Real Madrid	Bayern Munich	0

Cuadro 4: Simulated Football Results

He procesado los resultados de los partidos para determinar si el equipo local ganó (representado por 1), si el partido resultó en empate (representado por 0), o si ganó el equipo visitante (representado por 2). Con esta información, ya podemos avanzar hacia la definición y construcción de la matriz de transición para la Cadena de Markov.

La final se simuló basándose en los resultados de las semifinales, y se identificó al equipo ganador. En este caso con este modelo: **'Dortmund'**

2.3.4 Aprendizaje profundo

2.3.4.1 CNN (Red neuronal convolucional)

Antes de adentrarnos en el proceso de creación y entrenamiento de nuestro modelo, es crucial entender qué es una red neuronal convolucional (CNN) y por qué optamos por utilizarla. Las CNN son una categoría especializada de redes neuronales profundas, particularmente eficaces para procesar datos que adoptan una forma de cuadrícula, como las imágenes. Diseñadas para detectar patrones y variaciones complejas en grandes conjuntos de datos visuales, las CNN son ideales para aplicaciones de visión computarizada como el reconocimiento y clasificación de imágenes.

La estructura de una CNN incluye capas convolucionales que aplican filtros a la entrada para extraer mapas de características, capas de agrupamiento que reducen la dimensionalidad de estos datos mientras conservan los aspectos más importantes, y capas de normalización y activación que transforman las salidas de manera no lineal. Esta configuración permite a las CNN capturar eficientemente estructuras visuales a varios niveles de abstracción, esencial para aprender de imágenes que son altamente dimensionales y complejas, como los logos de equipos deportivos.

Elegimos una CNN para nuestro proyecto debido a su capacidad superior para reconocer y clasificar imágenes con gran precisión, y por su habilidad para manejar variaciones en forma, tamaño y color, características típicas de los logos de equipos.

El objetivo principal de este proyecto es desarrollar un modelo capaz de identificar con precisión el equipo de fútbol a partir de imágenes de sus logos, lo que implica reconocer y clasificar correctamente cada imagen en una de las categorías de equipos que participan en la Champions League de este año. Para ello, comenzamos el proceso recolectando imágenes de los logos de los equipos utilizando la extensión de Google "Download All Images", que nos permitió descargar eficientemente todas las imágenes relevantes de una página web.

Con las imágenes recolectadas, procedimos a configurar el entorno y preparar los datos para el entrenamiento del modelo utilizando TensorFlow y Keras. Algunas imágenes que se usaron fueron las siguientes:

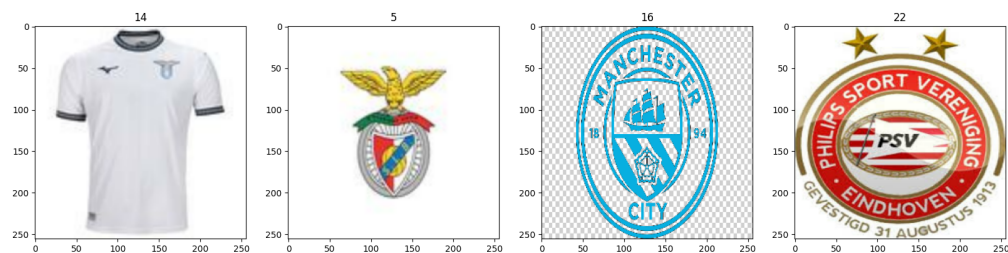


Figura 18: Pequeña muestra de imágenes aplicadas en el entrenamiento. Fuente: Creación propia

A continuación, definimos manualmente la arquitectura del modelo, implementando capas convolucionales, de agrupamiento, de normalización y de activación para adaptarse a nuestra tarea específica de clasificación multiclase de los logos de equipos. También implementamos un proceso de aumento de datos para mejorar la capacidad del modelo de generalizar a nuevas imágenes no vistas durante el entrenamiento, y normalizamos las imágenes dividiendo los valores de los píxeles por 255 para mejorar la estabilidad durante el entrenamiento.

Los datos se dividieron en conjuntos de entrenamiento, validación y prueba, cada uno con un propósito específico para asegurar que el modelo sea robusto y efectivo al enfrentarse a datos nuevos y desconocidos. Durante el entrenamiento, utilizamos técnicas como la normalización por lotes y el abandono para optimizar la estabilidad y minimizar el sobreajuste.

Finalmente, el modelo se entrenó utilizando el optimizador Adam y se monitoreó mediante TensorBoard, lo que nos permitió realizar ajustes en tiempo real y asegurar un aprendizaje efectivo.

Época	Pérdida	Precisión	Pérdida (validación)	Precisión (validación)
1	1.5926	0.5528	1.5461	0.5893
2	1.4957	0.5828	1.7260	0.5536
3	1.4483	0.5895	1.7745	0.5357
4	1.3750	0.6030	1.6500	0.5610
5	1.2646	0.6356	1.7303	0.5491
6	1.2523	0.6360	1.5802	0.5685
7	1.1535	0.6630	1.5754	0.5580
8	1.0481	0.6862	1.6043	0.5759
9	1.0830	0.6841	1.4749	0.6161
10	1.0031	0.7014	1.5427	0.6146

Cuadro 5: Resultados del entrenamiento del modelo

Evaluamos el rendimiento del modelo a través de gráficos que mostraban la precisión y la pérdida durante el entrenamiento, proporcionando una visión clara del comportamiento del modelo y garantizando que el resultado final fuera preciso y robusto, listo para identificar correctamente los equipos a partir de sus logos en aplicaciones reales.

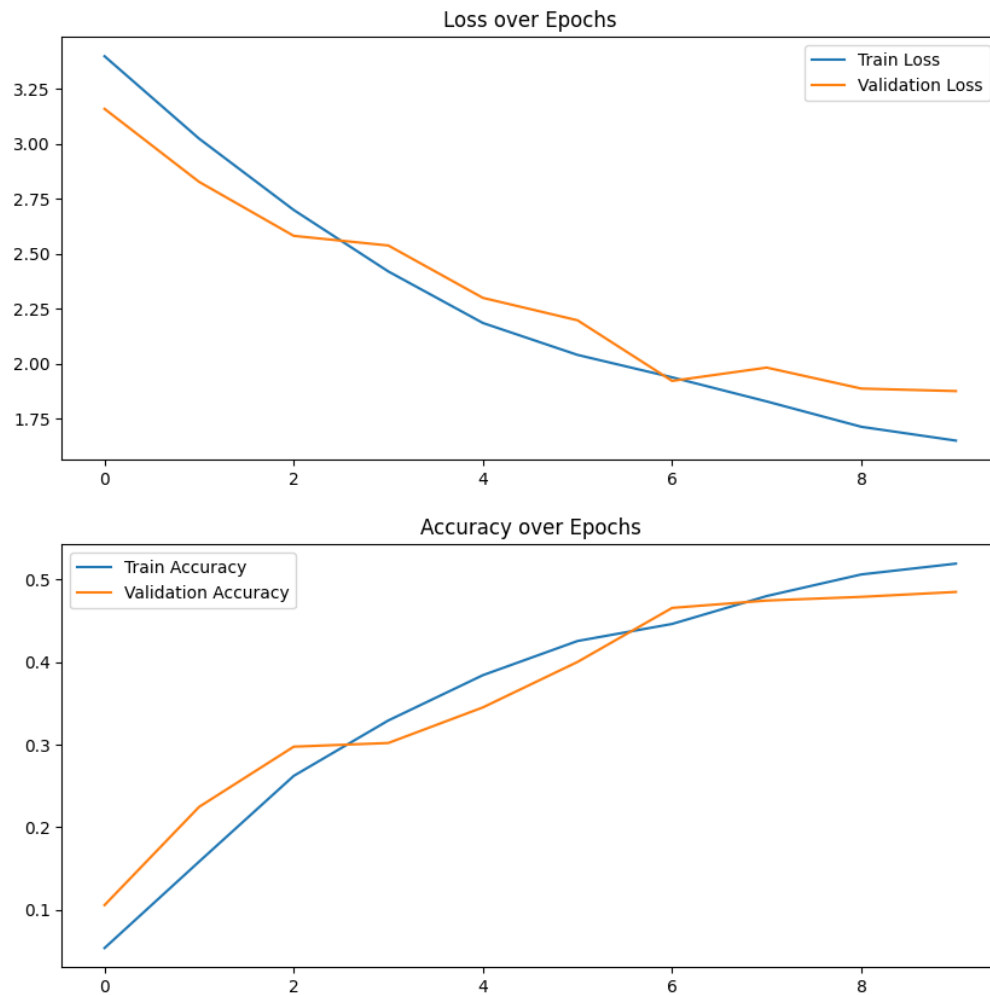


Figura 19: Gráficas que muestran la precisión y pérdida del modelo durante el entrenamiento. Fuente: Creación propia

2.3.4.2 TL (Transferencia de aprendizaje)

En este modelo, al introducir transferencia de aprendizaje con MobileNetV2, nuestra CNN experimenta una mejora significativa en el rendimiento. Aprovechamos las características aprendidas por MobileNetV2 en un conjunto de datos masivo como ImageNet, lo que permite a nuestro modelo entender mejor las características complejas de las imágenes de los equipos de fútbol. Esta transferencia de conocimiento nos permite alcanzar una mayor precisión y generalización en la tarea de clasificación en comparación con nuestro modelo de CNN básico, lo que resulta en una mejor capacidad para distinguir entre los diferentes logotipos de equipos con mayor precisión y eficacia.

Época	Pérdida	Precisión	Pérdida (validación)	Precisión (validación)
1	2.9694	0.2838	1.8754	0.5312
2	1.6561	0.5528	1.4304	0.6339
3	1.3195	0.6351	1.3177	0.6458
4	1.1688	0.6778	1.2062	0.6801
5	1.0481	0.7128	1.0417	0.7336
6	0.9417	0.7331	1.1833	0.6801
7	0.9017	0.7542	1.0902	0.7039
8	0.8311	0.7631	1.1481	0.7039
9	0.7365	0.7893	0.9648	0.7515
10	0.7269	0.7914	1.0727	0.7262

Cuadro 6: Resultados del entrenamiento del modelo con transferencia

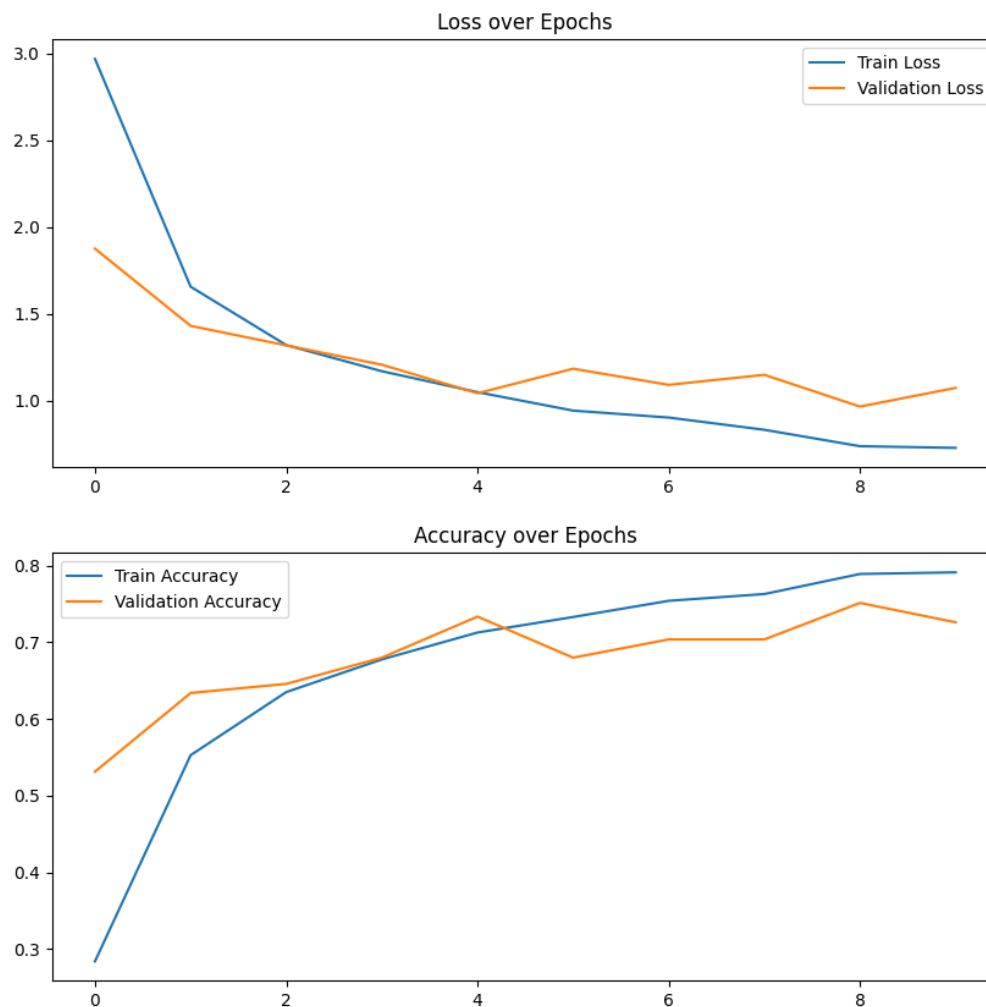


Figura 20: Gráficas que muestran la precisión y pérdida del modelo durante el entrenamiento. Fuente: Creación propia

2.3.4.3 DNN (Red neuronal profunda)

En este estudio, hemos implementado un modelo de red neuronal profunda utilizando las bibliotecas TensorFlow y Keras, con el objetivo de predecir los resultados de partidos deportivos. El modelo diseñado es un ensamblaje secuencial de capas densas que procesan características derivadas de datos históricos de partidos.

Inicialmente, los datos se extrajeron de un conjunto de datos recolectado sobre los partidos historicos utilizando la biblioteca pandas para la carga y

manipulación inicial. El preprocesamiento de estos datos involucró la transformación de la columna Score, que contiene los resultados de los partidos, en etiquetas binarias que identifican al ganador. Este paso es crucial para convertir resultados numéricos en una forma que el modelo pueda utilizar para aprender patrones de victoria o derrota.

Posteriormente, se aplicaron técnicas de codificación OneHot para convertir variables categóricas en un formato numérico adecuado para el procesamiento por la red neuronal. Utilizamos ColumnTransformer de scikit-learn para esta tarea, lo que asegura que cada categoría única se represente mediante un vector binario, eliminando así cualquier sesgo ordinal que pudiera introducirse en el modelo.

El diseño del modelo incluye múltiples capas densas; cada una con una función de activación ReLU, seleccionada por su capacidad para introducir no linealidades en el modelo sin afectar significativamente la velocidad de convergencia durante el entrenamiento. La compilación del modelo se realizó configurando el optimizador Adam, conocido por su eficiencia en la actualización de pesos y manejo de tasas de aprendizaje dinámicas, junto con la función de pérdida de entropía cruzada binaria, que es adecuada para problemas de clasificación binaria.

El entrenamiento se llevó a cabo dividiendo el conjunto de datos en porciones de entrenamiento y prueba. Esto no solo permite ajustar los pesos del modelo en base a los datos de entrenamiento sino también evaluar su capacidad de generalización en datos no vistos mediante el conjunto de prueba.

Este enfoque metodológico asegura que el modelo aprenda a predecir el ganador de los partidos basado en patrones y tendencias contenidos en los datos históricos, proporcionando una herramienta analítica que podría ser extendida a otros tipos de predicciones basadas en datos temporales o secuenciales.

3. Bibliografía:

- Lao, R. (2018, 6 junio). A Beginner's Guide to Machine Learning - Randy Lao - Medium. Medium. <https://medium.com/@randylaosat/a-beginners-guide-to-machine-learning-dfad19f6caf>
- Yalalov, D. (2022, 17 agosto). Aprendizaje automático (Machine learning & LLM). Metaverse Post. <https://mpost.io/es/glossary/machine-learning/>
- Corporativa, I. (s. f.). Descubre los principales beneficios del Machine Learning. Iberdrola. <https://www.iberdrola.com/innovacion/machine-learning-aprendizaje-automatico>

- Huanca, L. A. R. (2021, 30 diciembre). CLUSTERING - aprendizaje no supervisado - medium. Medium. <https://medium.com/aprendizaje-no-supervisado/clustering-cee49ad0061f>
- Modelado de cadenas de Markov predicción de resultados futuros con la simulación del modelo de cadenas de Markov - FasterCapital. (s. f.). FasterCapital. <https://fastercapital.com/es/contenido/Modelado-de-cadenas-de-Markov-prediccion-de-resultados-futuros-con-la-simulacion-del-modelo-de-cadenas-de-Markov.html>
- Silva, M. (2021, 10 diciembre). Aprendizaje por Refuerzo: Procesos de Decisión de Markov — Parte 1. Medium.
- Azevedo, D. (2022, 26 diciembre). Player Detection using Deep Learning - Analytics Vidhya - Medium. Medium. <https://medium.com/analytics-vidhya/player-detection-using-deep-learning-492122c3bf9>