

EII 4220 – Introducción a Data Science

Escuela de Ingeniería Industrial

Predicción de Resultados en Partidos de la Premier League mediante Ciencia de Datos

Por

**Joaquín Catán Vargas 20444854-k
Iván Garrote Valenzuela 20271610-5
Esteban Silva Trujillo 20.183.645-k**

Profesor: Diego Martinez, Javier Maturana.

Noviembre, 2024

Índice

Parte I: Cambios realizados a la propuesta.....	3
Parte II: Avances del proyecto.....	3
1) Discusión ética sobre legalidad de las apuestas:.....	3
2) Muestreo de entrenamiento completo:.....	4
3) Modelo Predictivo Funcional:.....	4
4) Segundo modelo predictivo funcional.....	7
5) Comparación métricas ambos modelos propuestos.....	8
a) Modelo de machine learning de random forest.....	9
b) modelo de machine learning de red neuronal.....	9
6) Desempeño modelo comparado con predicción de probabilidad realizada por página web de apuestas.....	10
7) Recomendaciones claves.....	11
Parte III: Reflexiones.....	11

Parte I: Cambios realizados a la propuesta

ANTES	DESPUÉS
El equipo menciona la relevancia del auge de las apuestas deportivas y los desafíos regulatorios, pero aún faltan referencias sólidas que respalden esta justificación, particularmente en términos de estudios previos o datos que demuestren el impacto real de estas apuestas en la industria.	ACCJ entregó un estudio sobre las apuestas en Chile mencionando que los casinos online presentan apostadores más frecuentes y con mayores niveles de riesgo (“Radiografía de la apuesta en Chile al 2023”). State of Play: The Sports Betting Market” (AGA, 2022) menciona el crecimiento de las apuestas deportivas (en US\$)
Se aprecia una mejora en lo que quieren lograr. En los "Would like to accomplish" aún podrían trabajar en un enfoque más detallado sobre cómo se lograrán en la práctica, ya que algunos de los aspectos más complejos, como la predicción de faltas o tarjetas, no están completamente justificados ni explicados en términos de implementación técnica	Al igual que los goles, de los cuales se consideran los goles históricos, para poder predecir faltas o tarjetas se esperaría utilizar un método similar en el cual se consideren las faltas y tarjetas históricas de cada partido. De esta forma obtener la probabilidad en que puedan ocurrir y que se puedan predecir en cierta manera.

Parte II: Avances del proyecto

En relación a los avances del proyecto están los must deliverables, que son los siguientes:

Generación de discusión ética sobre legalidad de las apuestas, Muestreo de entrenamiento completo, Modelo Predictivo Funcional. Para comenzar, se plantean diversas preguntas relacionadas a la ética sobre legalidad de las apuestas.

1) Discusión ética sobre legalidad de las apuestas:

¿Qué tan efectivas han sido las leyes actuales para controlar el crecimiento del mercado de apuestas deportivas y evitar fraudes o manipulación de resultados o afectar las probabilidades reales de ganar?

Al día de hoy en Chile no existen leyes que regulen las apuestas deportivas en línea, aunque la Superintendencia de Casinos de Juego (SCJ) y el Ministerio de Hacienda han impulsado proyectos para regular esta industria. En marzo de 2022, el Ministerio de Hacienda presentó un proyecto de ley que busca establecer un marco regulador para las apuestas en línea, incluidas las deportivas, exigiendo licencias para los operadores y medidas de transparencia en el uso de datos de los consumidores. Sin embargo, este proyecto aún no ha sido aprobado, por lo que el mercado sigue en una situación de legalidad.

¿Qué papel deberían tener los gobiernos y entidades reguladoras al permitir o restringir el uso de algoritmos predictivos en el ámbito de las apuestas deportivas?

Los gobiernos y las entidades reguladoras juegan un papel crucial en la gestión del uso de algoritmos predictivos en las apuestas deportivas, y su rol debería tener en cuenta tres aspectos clave como lo son la protección del consumidor, la integridad del deporte y el desarrollo de una industria regulada y transparente.

¿Cómo puede el uso de algoritmos predictivos en apuestas deportivas afectar la percepción pública sobre la legalidad y moralidad de las apuestas?

El uso de algoritmos predictivos en apuestas deportivas puede tener un impacto importante en la percepción pública de la legalidad y moralidad de las apuestas, especialmente cuando se percibe que estos algoritmos pueden manipular o influir en los resultados de las apuestas o en el comportamiento de los jugadores.

Se han seleccionado estas 3 preguntas abiertas para realizar una discusión sobre el tema seleccionado.

2) Muestreo de entrenamiento completo:

Luego, como segundo entregable se tiene un Muestreo de entrenamiento. El total de datos en el dataset inicial corresponde a 1560 puntos de datos, luego se obtuvo un conjunto de entrenamiento de 1092 de puntos de datos (70% del total) y finalmente un conjunto de prueba con 468 puntos de datos (30% del total).

```
# División del conjunto de datos en entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X_all_scaled, y_all, test_size=0.3, random_state=4)
```

En esta línea del código se divide el conjunto de datos en dos partes, el conjunto de entrenamiento que denominaremos (x_train y y_train) y el conjunto de prueba (x_test y y_test), especificando además que 30% de los datos será reservado para pruebas, y el 70% para entrenar el modelo.

```
# Entrenamiento del mejor modelo con el conjunto de entrenamiento
best_rf_model.fit(X_train, y_train)
```

El modelo best_rf_model se entrena utilizando el conjunto de entrenamiento x_train y y_train. El modelo aprende a reconocer patrones en los datos que los ayudarán a predecir el resultado de un partido.

```
# Evaluación del modelo en el conjunto de prueba
y_pred = best_rf_model.predict(X_test)
test_accuracy = accuracy_score(y_test, y_pred)
f1_weighted = f1_score(y_test, y_pred, average='weighted')
print('Precisión en el conjunto de prueba:', round(test_accuracy, 3))
print('F1 Score (weighted):', round(f1_weighted, 3))
print(classification_report(y_test, y_pred, target_names=['Away', 'Draw', 'Home']))
```

En esta evaluación accuracy_score mide la precisión general del modelo, calculando el porcentaje de predicciones correctas comparado con el total de predicciones. Aquí realizamos la comparación entre y_pred (predicciones del modelo) con y_test (valores reales en el conjunto de prueba).

Cabe destacar que en esta etapa se hizo una división entre conjunto de entrenamiento y conjunto de prueba, esta división es fundamental para evitar el overfitting o en otras palabras, cuando el modelo aprende demasiados de los datos de entrenamiento y no generaliza bien. y asegurar así que el modelo pueda predecir resultados con precisión en nuevos datos.

3) Modelo Predictivo Funcional:

Por otra parte, tenemos el último must deliverable, que es el Modelo Predictivo Funcional. Se siguió la recomendación del profesor de utilizar un modelo de clasificación de outcome de 3 variables, Para este entregable entonces se implementó el modelo de machine learning de random forest, algoritmo de aprendizaje automático supervisado que se usa para solucionar problemas de clasificación y regresión. Se eligió debido a su capacidad para calcular probabilidades de pertenencia a clase, lo que permite no solo predecir el resultado más probable para cada partido (victoria local, empate o victoria visitante) sino también proporcionar la probabilidad de cada posible resultado. Esta información adicional es

útil para analizar la confianza del modelo en cada predicción, lo cual resulta valioso en contextos de toma de decisiones y análisis deportivo.

Para ser más específicos, vamos a la parte del código en donde se crea el modelo:

```
rf_model = RandomForestClassifier(random_state=4)
grid_search = GridSearchCV(rf_model, param_grid, cv=5, scoring='f1_weighted')
grid_search.fit(X_train, y_train)
best_rf_model = grid_search.best_estimator_

# Resultados de GridSearch
print("Best parameters found by GridSearch:", grid_search.best_params_)

# Validación cruzada para obtener probabilidades
y_pred_cv = cross_val_predict(best_rf_model, X_train, y_train, cv=5, method='predict_proba')
```

Este fragmento de código crea y ajusta un modelo de clasificación de bosque aleatorio (Random Forest) utilizando una búsqueda de hiper parámetros (Grid Search) y validación cruzada para encontrar la mejor configuración del modelo.

Luego, la función predict_match_result toma como entrada los nombres de dos equipos (local y visitante) y usa un modelo de clasificación ya entrenado para predecir la probabilidad de los posibles resultados (victoria del equipo local, empate o victoria del equipo visitante).

```
# Función para hacer predicciones de partidos específicos
def predict_match_result(home_team, away_team, data, model, scaler, original_data):
    # Normalizar nombres de equipos ingresados
    home_team = normalize_team_name(home_team)
    away_team = normalize_team_name(away_team)

    # Verificar si los nombres normalizados existen en el conjunto original de equipos
    if home_team not in original_data['HomeTeam'].unique():
        print(f"No se encontró el equipo {home_team} en el conjunto de datos.")
        return None
    if away_team not in original_data['AwayTeam'].unique():
        print(f"No se encontró el equipo {away_team} en el conjunto de datos.")
        return None

    # Obtener las columnas dummies correspondientes
    home_team_columns = [col for col in original_data.columns if col.startswith('Home_')]
    away_team_columns = [col for col in original_data.columns if col.startswith('Away_')]

    # Obtener HomeWinRate y AwayWinRate para los equipos
    home_win_rate = original_data[original_data['HomeTeam'] == home_team]['HomeWinRate'].values[0]
    away_win_rate = original_data[original_data['AwayTeam'] == away_team]['AwayWinRate'].values[0]

    # Crear DataFrame con todos los valores en 0
    X_example = pd.DataFrame(0, index=[0], columns=columns_name)

    # Asignar 1 a las columnas correspondientes
    home_team_column = f'Home_{home_team}'
    away_team_column = f'Away_{away_team}'
    if home_team_column in X_example.columns:
        X_example[home_team_column] = 1
    if away_team_column in X_example.columns:
        X_example[away_team_column] = 1

    # Asignar HomeWinRate y AwayWinRate
    X_example['HomeWinRate'] = home_win_rate
    X_example['AwayWinRate'] = away_win_rate

    # Escalar los datos de ejemplo
    X_example_scaled = scaler.transform(X_example)

    # Predecir probabilidades
    pred_probs = model.predict_proba(X_example_scaled)[0]

    # Mostrar las probabilidades
    pie_labels = ['Visita', 'Empate', 'Local']
    print("Probabilidades:")
    for label, prob in zip(pie_labels, pred_probs):
        print(f"{label}: {prob:.2f}")
```

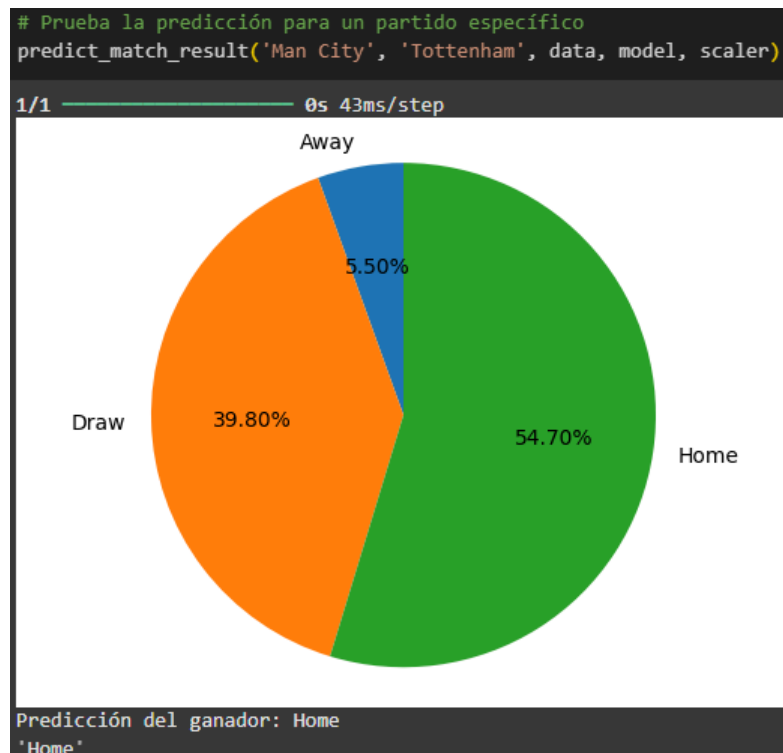
La función está compuesta de 8 puntos que son importantes y se deben destacar:

- Normalización de nombres de equipos (convierte los nombres de los equipos ingresados a un formato normalizado)
- Validación de equipos (Se asegura de que los equipos ingresados están en el conjunto de datos original.)
- Configuración de columnas para los equipos (obtiene las columnas correspondientes a cada equipo en las variables dummy generadas)

Extracción de tasas de victoria de local y visitante (acá se agregan 2 variables importantes y que son utilizadas como referencia en las recomendaciones), luego se construye un ejemplo de entrada con las mismas columnas que el conjunto de entrenamiento, inicializando todos los valores en 0. Después, se escala el ejemplo usando el mismo escalador (scaler) que fue aplicado al conjunto de entrenamiento.

Finalmente, se realiza la predicción de las probabilidades; acá se usa el modelo para predecir las probabilidades de cada resultado posible (visita, empate o local) para posteriormente entregarlas por pantalla.

Finalmente es necesario graficar y observar los resultados obtenidos:



Distribución de probabilidades de ocurrencia de cada evento (Local, Empate, Visita) junto a la predicción del resultado final (categóricamente hablando) en el partido entre Manchester City como local versus Tottenham como visitante.

En resumen, el modelo estima que el partido tiene mayores probabilidades de terminar en victoria para Home (54,70% para Manchester City) que en cualquier otro resultado, basándose en las características históricas y de rendimiento de ambos equipos. A pesar de esta preferencia por la victoria del equipo local, encontramos que las probabilidades de empate de los dos equipos es elevada reflejo de los estados de los últimos años de ambos equipos.

4) Segundo modelo predictivo funcional

Como segundo modelo predictivo funcional se nos hizo necesario seleccionar un modelo que sea capaz de ofrecer mayor capacidad y flexibilidad para abordar la complejidad de los datos y mejorar la precisión en problemas de clasificación. Para este entregable entonces se implementó el modelo de machine learning de red neuronal, este se eligió debido a su capacidad para aprender patrones complejos en los datos y manejar problemas de clasificación multiclase, como predecir el resultado de un partido con tres posibles outcomes (local, empate y visitante). Las redes neuronales son particularmente útiles cuando se trabaja con grandes cantidades de datos y características, ya que pueden capturar relaciones no lineales entre los atributos. En el caso del análisis futbolístico, factores como el rendimiento del equipo como local o visitante, junto con otras estadísticas del partido pueden influir en el resultado de manera no lineal.

La normalización de los atributos fue un paso necesario para mejorar la eficiencia y el rendimiento de la red neuronal. Dado que las redes neuronales son sensibles a los rangos de entrada, la normalización asegura que cada atributo contribuya de manera equitativa al proceso de aprendizaje. La técnica utilizada fue el escalado Min - Max que ajusta los valores de los atributos al rango [0,1].

También se realiza una codificación de los equipos en números, es decir, dado que los nombres de los equipos son valores categóricos y las redes neuronales solo pueden trabajar con entradas numéricas, es necesario convertir estos valores a números. Se utilizó LabelEncoder para la transformación,, permitiendo al modelo comprender la identidad de los equipos sin interpretarlos como valores secuenciales, ya que los nombres de los equipos no siguen ningún orden inherente.

La función para hacer predicciones de partidos específicos se compone de la siguiente manera:

```
# Función para hacer predicciones de partidos específicos
def predict_match_result(home_team, away_team, data, model, scaler):
    dato = data[(data['HomeTeam'] == home_team) & (data['AwayTeam'] == away_team)]
    if dato.empty:
        print(f"No hay datos para el partido entre {home_team} y {away_team}.")
        return None

    X_example = dato[columns_name]
    X_example_scaled = scaler.transform(X_example)

    pred = model.predict(X_example_scaled)
    pred_probs = [float(format(x, '.3f')) for x in pred[0]]
```

La primera línea de código se ubican los nombres de los equipos que juegan el partido, el dataset que contiene los datos de los partidos, el modelo de red neuronal ya entrenado (modelo) y el objeto scaler utilizado para normalizar los atributos. con x_example se seleccionan las columnas relevantes del partido para generar las características que el modelo necesita, luego las características se normalizan usando scaler para que se encuentren en el mismo rango en el que fueron escaladas durante el entrenamiento.

Luego el model.predict, el modelo predice el resultado del partido, generando una lista de probabilidades para cada clase (local, empate o visitante).

Pred_probs, las probabilidades generadas por el modelo se formatean a tres decimales y se almacenan en una lista, cada elemento representa la probabilidad de un resultado específico, con los índices típicamente representando las clases en el orden en el que fueron entrenadas.

Es necesario graficar y observar los resultados obtenidos, para lo cual se tiene el siguiente ejemplo:



Distribución de probabilidades de ocurrencia de cada evento (Local, Empate, Visita) junto a la predicción del resultado final (categóricamente hablando) en el partido entre Manchester City como local versus Liverpool como visitante.

	Date	HomeTeam	AwayTeam	FTHG	FTAG	FTR	HTHG	HTAG	HTR	Referee	...	HST	AST	HF	AF	HC	AC	HY	AY	HR	AR
496	2020-11-08	Man City	Liverpool	1	1	D	1	1	D	C Pawson	...	2	3	19	11	1	2	3	1	0	0
1108	2022-04-10	Man City	Liverpool	2	2	D	2	1	H	A Taylor	...	5	4	9	11	4	1	1	4	0	0
1450	2023-04-01	Man City	Liverpool	4	1	H	1	1	D	S Hooper	...	8	1	9	12	7	1	1	1	0	0

La tabla muestra los últimos partidos entre Manchester City y Liverpool. De estos, dos terminaron en empate, mientras que el otro tuvo como resultado una contundente victoria de Manchester City. Las otras columnas incluyen otras estadísticas como goles en el primer tiempo, tarjetas amarillas y rojas, faltas y otras variables que el modelo usa para realizar la predicción.

5) Comparación métricas ambos modelos propuestos

Con `classification_report` buscamos realizar una evaluación detallada para cada clase presente, es decir, "Away", "Draw", "Home". Para cada clase de cada modelo se muestra:

a) Modelo de machine learning de random forest

Precisión en el conjunto de prueba: 0.927				
F1 Score (weighted): 0.925				
	precision	recall	f1-score	support
Away	0.92	0.97	0.94	156
Draw	0.94	0.78	0.85	128
Home	0.93	0.99	0.96	184
accuracy			0.93	468
macro avg	0.93	0.91	0.92	468
weighted avg	0.93	0.93	0.93	468

-Precisión: La proporción de predicciones correctas entre todas las predicciones para esa clase

-Recall: La proporción de predicciones correctas entre todos los casos verdaderos de esa clase.

-F1 score: El promedio armónico de precisión y recall para cada clase.

De los resultados de prueba, tenemos que el modelo predijo correctamente el resultado en el 92,7% de los casos.

El valor de F1 score obtenido de 0,925 nos indica una buena métrica para datasets desbalanceados, ya que toma en cuenta la precisión y el recall de cada clase de forma equilibrada.

b) modelo de machine learning de red neuronal

Precisión en el conjunto de prueba: 0.953				
F1 Score (weighted): 0.952				
	precision	recall	f1-score	support
Away	0.97	1.00	0.99	156
Draw	0.99	0.84	0.91	128
Home	0.92	0.99	0.95	184
accuracy			0.95	468
macro avg	0.96	0.94	0.95	468
weighted avg	0.96	0.95	0.95	468

Tenemos que el modelo predijo correctamente el resultado en el 95,3% de los casos.

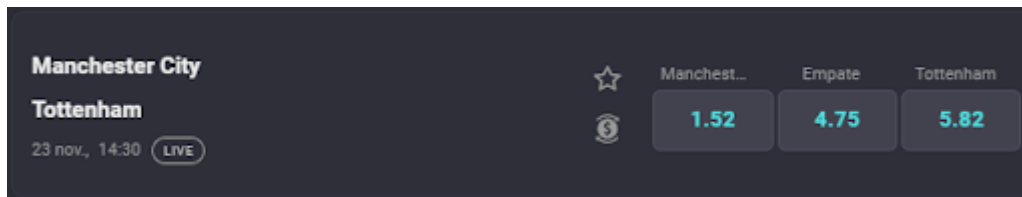
El F1 score obtenido, con un valor de 0.952, representa una métrica adecuada para conjuntos de datos desbalanceados, ya que combina de manera equilibrada la precisión y el recall de cada clase, proporcionando una visión más completa del desempeño del modelo.

En conclusión la decisión de elegir el modelo de Random Forest radica en que aunque el modelo de redes neuronales presentó un rendimiento superior, con una mayor precisión global en el conjunto de prueba (0.953 frente a 0.927 del Random Forest), su naturaleza de "caja negra" dificulta la comprensión del proceso interno de toma de decisiones. En contraste, el modelo de Random Forest, aunque ligeramente menos preciso, nos permite analizar directamente las características más importantes y entender cómo se llega a las predicciones. Esta transparencia es crucial para mejorar el modelo y justificar sus resultados, especialmente en un contexto académico y práctico donde la trazabilidad de las decisiones tiene un alto valor.

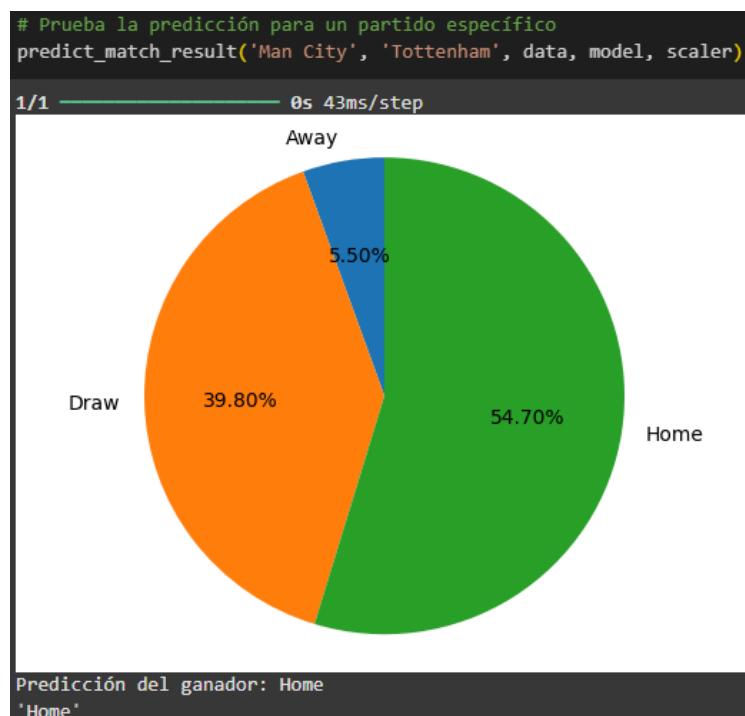
Además, consideramos que un modelo más interpretable facilita el ajuste futuro y la generación de confianza en las predicciones, lo que resulta esencial al tratar con datos dinámicos y resultados que pueden influir en áreas como las apuestas deportivas.

6) Desempeño modelo comparado con predicción de probabilidad realizada por página web de apuestas

A continuación realizaremos una comparación entre el resultado de un partido en particular por parte del modelo diseñado, frente a los resultados que ofrece una de las casas de apuestas de la web.



Cuotas de página de apuestas Online.(Para obtener la probabilidad es necesario dividir **100 en $100 \cdot \text{cuota}$**).



Prueba de predicción para un partido específico por modelo Random forest.

Al comparar las probabilidades obtenidas por el modelo de Random Forest con las probabilidades implícitas de las casas de apuestas para el partido entre Manchester City y Tottenham, se pueden destacar las siguientes conclusiones:

Las casas de apuestas asignan una probabilidad implícita del 65.79%, mientras que el modelo predice un 54.70%. Esto indica que el modelo es más conservador respecto a la victoria del equipo local en comparación con las apuestas.

El modelo predice un 39.80% de probabilidad de empate, una cifra más elevada que el 21.05% implícito en las cuotas de apuestas.

Tanto las apuestas 17.18% como el modelo 5.50% coinciden en que la probabilidad de victoria de Tottenham es baja. Sin embargo, el modelo es mucho más pesimista respecto a las posibilidades del equipo visitante.

En general, aunque el modelo proporciona resultados interesantes, hay margen para optimizar su rendimiento y lograr mayor congruencia con las probabilidades observadas en las apuestas.

7) Recomendaciones claves

Como equipo, buscamos plantear una serie de recomendaciones basadas en los hallazgos obtenidos a lo largo del estudio. Estas sugerencias están diseñadas para potenciar el impacto y la eficacia de las acciones derivadas del análisis realizado. Cada propuesta se ha elaborado con el propósito de atender puntos clave identificados durante la investigación y ofrecer directrices que contribuyan a la implementación de mejoras estratégicas.

Basándonos en la retroalimentación entregada, se determinaron las siguientes recomendaciones al sistema en estudio:

La comparación del sistema con los valores entregados por las casas de apuestas

Incorporar características adicionales (clima, estadísticas avanzadas de jugadores) que puedan mejorar las predicciones o nuevas características que puedan crearse en base a las que ya se tienen

Ampliar el espectro de ligas a predecir (incorporar una liga más, por ejemplo LaLiga).

Parte III: Reflexiones

Para finalizar, se desarrolló una parte de reflexión que consta de las siguientes preguntas:

1. ¿Creen que es necesario hacer cambios en los roles del equipo? (por ejemplo, en el spokesperson).

Como grupo, creemos que no hace falta cambiar los roles que ya tenemos. Cada uno ha cumplido bien con su parte, y eso nos ha ayudado a mantener el proyecto en buen camino. Desde el comienzo, hemos tenido una buena dinámica de trabajo en la que todos sabemos qué hacer y en qué enfocarnos, y esto nos ha permitido avanzar sin problemas. La forma en que repartimos los roles está funcionando bien ya que todos aportamos desde nuestras áreas. Gracias a la organización que llevamos, hemos podido resolver las dificultades que han ido surgiendo a lo largo del proyecto, especialmente en la parte de programación del modelo de predicción. Sabemos que podemos seguir avanzando y alcanzar los objetivos del proyecto, así que preferimos mantener los roles tal como están.

2. Reflexionen sobre cómo afectó la incertidumbre en un proyecto como este y cómo la enfrentaron, junto con qué decisiones tomaron y qué aprendieron de la experiencia.

La incertidumbre en nuestro proyecto surgió principalmente durante la programación del código de predicción para calcular las probabilidades de que un equipo gane, pierda o empate, más que en la obtención de los datos. Esto debido a que al inicio del proyecto seleccionamos una base de datos específica que consideramos sólida y confiable para nuestras necesidades, lo que nos permitió partir con una base estable. De todas formas hay que tener en cuenta que realizamos una “limpieza” de datos en donde mantuvimos los que como grupo consideramos necesarios para el proyecto y eliminando los que de cierta forma entorpecían o sobraban. Sin embargo, el desafío apareció cuando comenzamos a implementar los modelos de predicción y ajustar el código. Es por ello que el código se trabajó en componentes más pequeños, lo cual facilitó un poco el proceso con tal de poder encontrar errores en

el camino y nos permitió identificar rápidamente las partes que necesitaban ajustes. Este enfoque ayudó a abordar cada parte del proceso de predicción de manera más estructurada. Probamos diferentes configuraciones de modelos y parámetros para encontrar los que mejor se adaptaran a los datos específicos que estábamos usando. Al no contar con datos externos para comparar nuestros resultados, utilizamos pruebas internas y ajustes basados en observación y análisis de patrones que como grupo consideramos pertinentes. Cuando enfrentamos problemas específicos en el código, recurrimos a la inteligencia artificial con la finalidad de resolver dudas y encontrar errores que no comprendíamos. Esto nos permitió superar barreras en el desarrollo y mejorar la precisión del modelo con base en las mejores prácticas.

3. ¿Cómo afectó la incertidumbre para el desarrollo del proyecto la elección de la base de datos? Reflexione al respecto de esto.

La elección de la base de datos nos generó algo de incertidumbre al principio del proyecto, ya que encontrar una que realmente se ajustara a lo que necesitábamos no fue fácil. Una vez que encontramos una base de datos que parecía buena y provenía de un sitio oficial (confiamos en la veracidad de los datos), nos dimos cuenta de que no todos los datos eran útiles para la finalidad de nuestro proyecto y tuvimos que "limpiarlos" o eliminar los que no servían para que el paso posterior no fuera tan engorroso y así evitar errores. Este proceso de limpieza fue bastante importante porque había muchos datos irrelevantes o incompletos, y si no los hubiéramos eliminado, podrían haber afectado la precisión de nuestras predicciones y terminar molestando al momento de programar el código. La incertidumbre vino de no saber si podríamos depurar esos datos correctamente y si la base que habíamos elegido nos daría la información necesaria de forma confiable o si el código como tal estaba procesando bien los datos. Al reflexionar como grupo, nos dimos cuenta de que elegir una base de datos no se trata solo de encontrar una con mucha información sino de asegurarse de que los datos sean claros, completos y útiles para lo que queremos hacer.