Field Wizard: Predicting Positions and Formations for Success on the Soccer Field

Guillem Miralles

Abstract: The project, entitled "Field Wizard: Predicting Positions and Formations for Success on the Soccer Field", represents a comprehensive exploration of football player data and match-related information with the goal of discovering insights, patterns and predictive models in the professional football environment. The project consists of a series of Jupyter notebooks, each focusing on specific aspects of analytics, ranging from exploratory data analysis (EDA) to predictive modelling of football-related phenomena. The main objective of our project is to develop two robust dynamic models aimed at optimising on-field strategies. The first model seeks to predict the possible positions that a player can occupy on the field based on a careful analysis of his individual attributes. Through a meticulous analysis of variables such as speed, stamina, technical skills, among others, this model proposes strategic positions that enhance the individual skills of each player and, at the same time, contribute to the overall scheme of the team. The second model aims to optimise the team's formation for a specific match, taking into account not only the attributes of the team's own players, but also analysing the characteristics of the opposing team. This dual analysis allows for the creation of a formation that not only enhances one's own team's strengths, but can also effectively counteract the opposing team's strengths. When evaluating the developed models, highly satisfactory results were obtained. The model for predicting player positions achieved an accuracy of 0.97 and an F1 score of 0.99. On the other hand, the model for predicting optimal team formation reported an accuracy of 0.783 and an F1 score of 0.782.

Key words: machine learning, deep learning, sports, football

1. Introduction

En la era contemporánea, nos encontramos inmersos en una revolución de datos donde cada aspecto de nuestra sociedad está siendo transformado por el poder de la analítica avanzada y el aprendizaje automático. El deporte, y particularmente el fútbol, no es una excepción a esta tendencia; la data ha pasado a jugar un papel fundamental en la toma de decisiones estratégicas, no solo en el ámbito empresarial y de gestión, sino también directamente en el campo de juego. La optimización basada en datos no solo ofrece un profundo entendimiento de las dinámicas del juego, sino que también tiene el potencial de elevar el nivel de la competencia a alturas sin precedentes, promoviendo estrategias cada vez más sofisticadas y eficaces.

Dentro de este contexto, la personalización del entrenamiento y las tácticas del equipo basadas en las características individuales y colectivas de los jugadores se ha vuelto una piedra angular para alcanzar el éxito. Los entrenadores y analistas buscan continuamente formas innovadoras de utilizar datos para obtener ventajas competitivas, explorando métricas que van desde el análisis biomecánico hasta las estadísticas de rendimiento en tiempo real.

A pesar del progreso significativo en este ámbito, existe aún un vasto terreno inexplorado y un gran potencial para profundizar la integración de la ciencia de datos en el fútbol. Este trabajo surge desde la necesidad de explorar formas más profundas y matizadas de utilizar datos para influir en estrategias de juego. Se centra en el desarrollo de modelos predictivos que pueden ayudar a optimizar la asignación de posiciones de los jugadores y la formación del equipo,

tomando en cuenta una amplia gama de atributos tanto individuales como colectivos, y factores contextuales del partido a enfrentar.

La investigación que aquí se presenta se enfoca en la creación de dos modelos de aprendizaje automático; el primero, orientado a prever las posibles posiciones de un jugador basándose en sus atributos individuales, y el segundo, dirigido a optimizar la formación del equipo para un encuentro específico, considerando los atributos de los jugadores propios y del equipo rival. A través de este enfoque dual, se busca proporcionar una herramienta potente y flexible que pueda ser una guía valiosa para los equipos en la planificación de sus estrategias, tanto en entrenamientos como en competiciones.

En este paper, delinearemos el proceso de construcción de estos modelos, presentando tanto la metodología empleada como los resultados obtenidos, con el fin de destacar no solo la viabilidad sino también la eficacia de esta aproximación para la optimización de estrategias en el fútbol.

2. Datos

En este proyecto, se usan fuentes de datos provenientes de la plataforma de datos abiertos *Kaggle*. Kaggle es una plataforma en línea donde se puede competir en competencias de análisis de datos, colaborar en proyectos, y acceder a un gran conjunto de *datasets* públicos y recursos educativos para aprender. En esta plataforma, podemos encontrar las siguientes fuentes de datos:

1. ***European Soccer Database*** es una base de datos exhaustiva diseñada para análisis de datos y aprendizaje automático en el contexto del fútbol europeo. La base de datos contiene información detallada sobre más de 25,000 partidos y más de 10,000 jugadores de 11 países europeos, abarcando las temporadas de 2008 a 2016. Se utilizan *dataframes* destacados en la siguiente Figura 1:

DATAFRAMES	ROWS	COLS
Player_Attributes	183978	42
Player	11060	8
Match	25979	115
League		
Country		
Team	299	5
Team_Attributes		

Figura 1: *Tablas de la European Soccer Database*

- En la tabla Player tenemos las siguientes variables: *api_id*, *nombre del jugador*, *altura y peso*. Se muestra en la Figura 2:

player_api_id	player_name	height	weight
505942	Aaron Appindangoye	182.88	187
155782	Aaron Cresswell	170.18	146
162549	Aaron Doran	170.18	163

***Figura 2**: Información de la tabla Player*

- En la tabla *Player_Attributes* se encuentra un total de 38 atributos de un jugador registradas en un determinado momento de la temporada. Es decir, un jugador normalmente tiene varios registros a lo largo de su carrera. Podemos ver 5 ejemplos a continuación*:*
    - *sprint_speed: Atributo de velocidad máxima del jugador.*
    - *agility: Atributo de agilidad del jugador.*
    - *reactions: Atributo de reacciones del jugador.*
    - *balance: Atributo de equilibrio del jugador.*
    - *shot_power: Atributo de potencia de tiro del jugador.*
- En la tabla *Match* encontramos datos sobre partidos disputados. La información que nos interessa del siguiente dataset es:
    - *Api_id de los equipos que jugaron el partido*
    - *Temporada en la que se jugó el partido.*
    - *Resultado (goles del local y goles del visitante)*
    - *Api_id de los jugadores de cada equipo que disputaron el partido*
    - *Posición en el eje y de cada jugador (altura del campo)*
- La tabla *Team* solo se utilitza para transformar el api del equipo local y visitante al nombre real del equipo, ya que no contamos con esta información en la tabla anterior.

2. ***FIFA complete player dataset***, donde se reúne información detallada de los jugadores disponibles en el modo Carrera de las versiones del videojuego FIFA desde el FIFA 15 hasta el FIFA 22, presentado en un archivo llamado "players_AÑO.csv". El detalle de este archivo permite realizar múltiples comparaciones de un mismo jugador a través de las últimas 8 versiones del videojuego, abarcando más de 100 atributos individuales para cada jugador. De todos los datasets que contiene esta base de datos, usaremos los dos que hacen referencia al año 2015 y 2016:
    - players_2015.csv
    - players_2016.csv
- Dentro de estos dos dataframes, nos interesan las dos variables que mostramos en la Figura 3

Player	Position
L. Messi	RW, CF
C.Ronaldo	LW, LM
A. Robben	RM, LM, RW

***Figura 3**: Información Relevante de los datasets FIFA*

3. Materiales y Métodos

3.1. Predecir posiciones en base a los atributos de un jugador

3.1.1. Eliminación y Transformación de Datos

En esta fase, se eliminaron los valores incorrectos identificados en las columnas 'defensive_work_rate' y 'attacking_work_rate'. Posteriormente, se categorizaron los valores y se prescindió de la columna 'id'. Para una visión más detallada, los datos fueron agrupados por el nombre del jugador y se procedió a calcular la media de sus atributos por temporada. Se puede intuir, que los jugadores, al largo de su carrera, van variando de atributos y también pueden variar de posición, es por ello que nos quedamos con su media por temporada. Si mantenemos tantas filas como sea posible, podríamos identificar patrones y tendencias en los atributos de un jugador en diferentes contextos (equipos, edad del jugador, etc...). Esto ayudara al modelo a identificar mejor los patrones para predecir la posición.

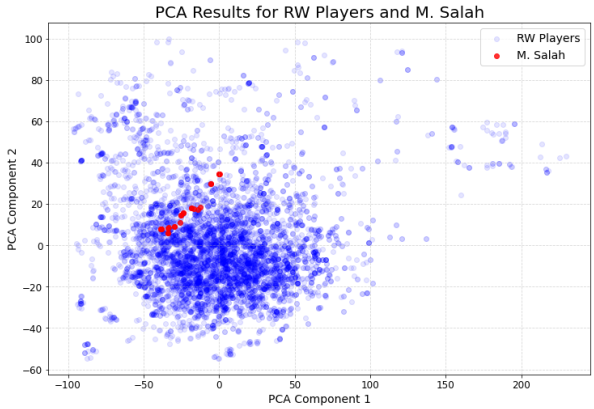

***Figura 4**: 1a y 2a componente principal de los registros del jugador Mohamed Salah*

Se observa como, en la Figura 4, encontramos que M. Salah ha variado su información a lo largo de los registros que tenemos en nuestros datos.

3.1.2. *Integración de Datos de la Base de Datos FIFA*

Un paso substancial es la incorporación de información sobre las posiciones de los jugadores procedente de la base de datos FIFA. Se realizó una transformación de los nombres de los jugadores para coincidir con los registros en el dataset de FIFA, permitiendo una fusión

precisa de los datasets de FIFA 15 y FIFA 16 con el dataset principal. Recordamos que un jugador puede tener associades diferentes posiciones.

Como solo tenemos información de FIFA (por tanto de posiciones) de jugadores los años 2015 y 2016, hay jugadores en nuestros datos que no tienen posiciñón associada. Como se puede ver en la Figura 5, el número de jugadores con y sin posicion es parecido; es decir, aproximadamente la mitad de nuestros registros de jugadores no tienen posición y, por tanto, no sabemos su posición de juego en el campo. Esto trataremos de predecirlo mediante un modelo.

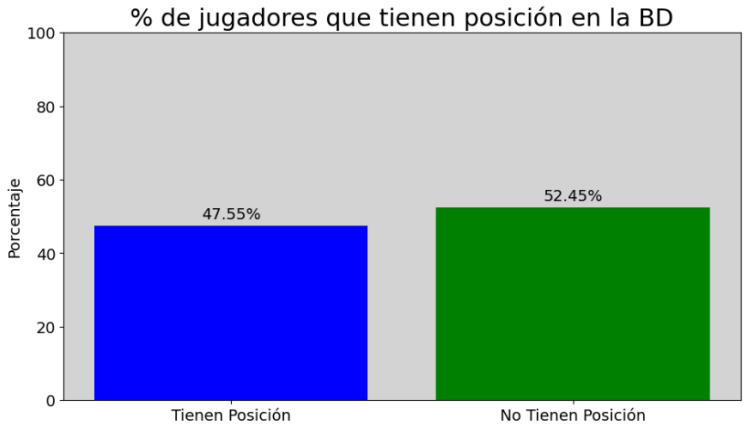

Figura 5: *Porcentaje de jugadores con y sin posición en los datos*

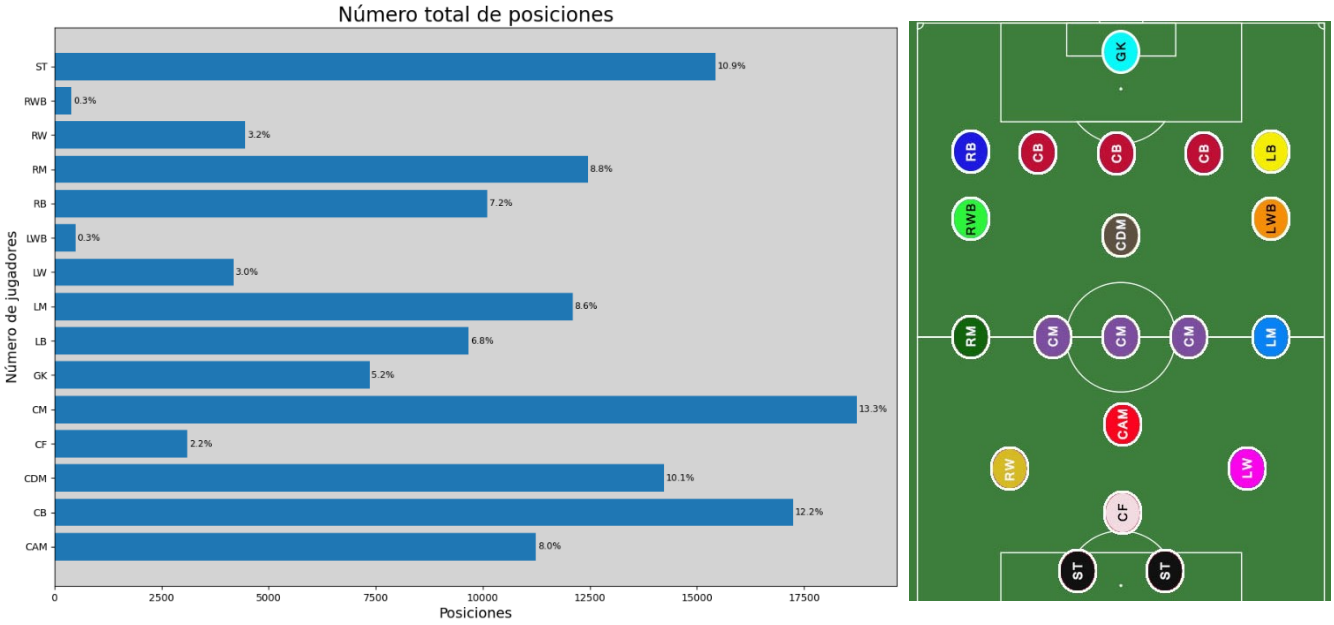

Figura 6: *Porcentaje de cada posición en el campo y su ubicación exacta en el terreno de juego*

En la Figura 6, podemos observar que hay posiciones con un número muy pequeño de apariciones respecto a las otras, como LWB, RWB y CF. Entonces, dada la mejora que se ha visto, se ha cambiado las posiciones "LWB" por "RB", "RWB" por "RB" y "CF" por "ST".

3.1.3. Preprocesado:

1. **One-Hot Encoding**: Codificamos las posiciones con un vector One-Hot Encoding. Cada posición se convierte en una nueva columna siendo esta uina variable binaria. És un problema multietiqueta

2. **División Train/Test**

Longitud Train = 67009

Longitud Test = 16753

3. **Estandarización MinMaxScaler**

4. **Feature Engeneering**: Dados los 38 atributos que tenemos en nuestros datos y, mediante ingeniería de características, creamos 10 atributos nuevos haciendo un total de 48 atributos. Podemos ver la matriz de confusión de los atributos de los jugadores.

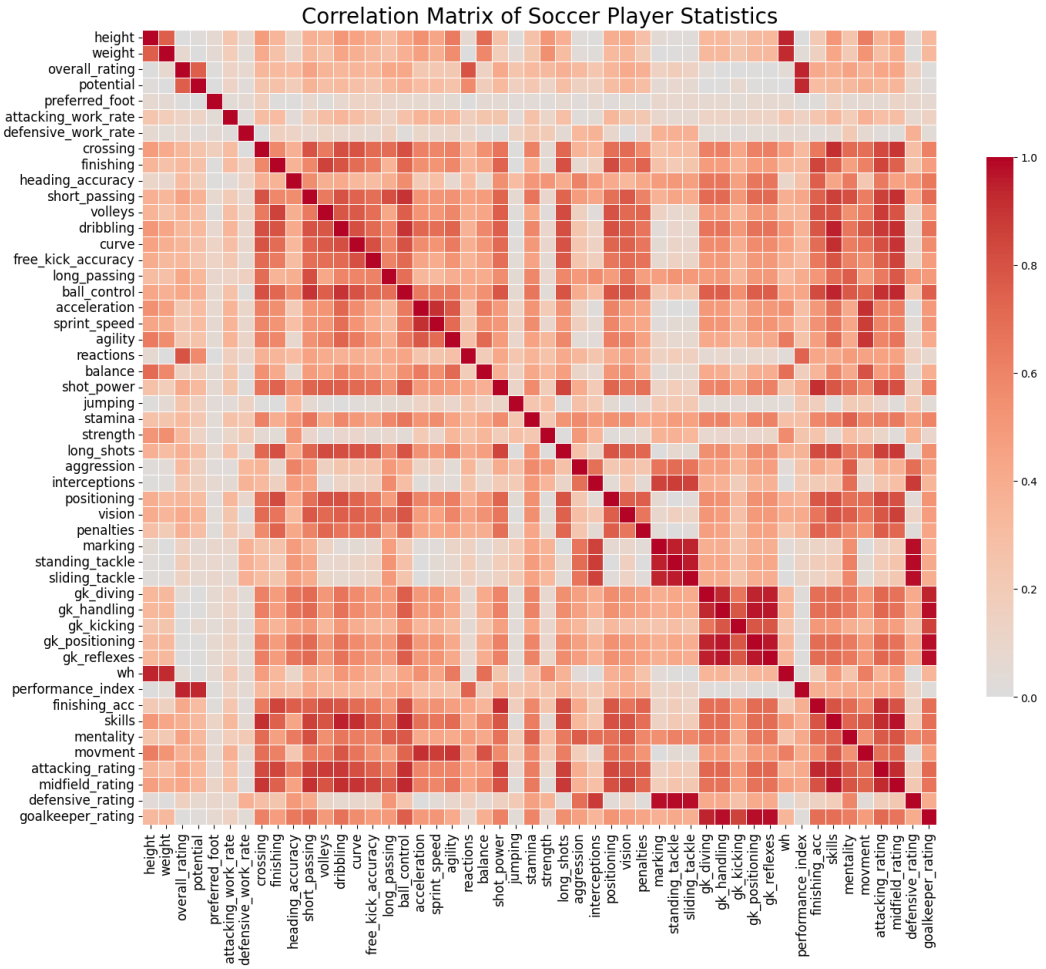

Figura 7: *Matriz de correlación de los atributos de los datos*

3.1.4. Construcción del modelo

Se prueban los siguientes modelos para tratar de estimar las posiciones del jugador en base a sus atributos:

- Regresión Lineal
- Random Forest
- Red Neuronal

3.1.5. Predicciones

Una vez entrenado el modelo, realizamos las predicciones de los jugadores que no tenían posición asociada en nuestra BD. De esta forma, finalmente, exportamos los datos con el nombre del jugador, los atributos y sus posibles posiciones.

3.2. Optimización de la formación de un equipo

3.2.1. Crear el dataframe de formaciones

En la base de datos de los partidos (*Match*), no tenemos las formaciones utilizadas por el equipo en el partido. En cambio, si que tenemos la posición en el eje y de cada jugador (altura del campo).

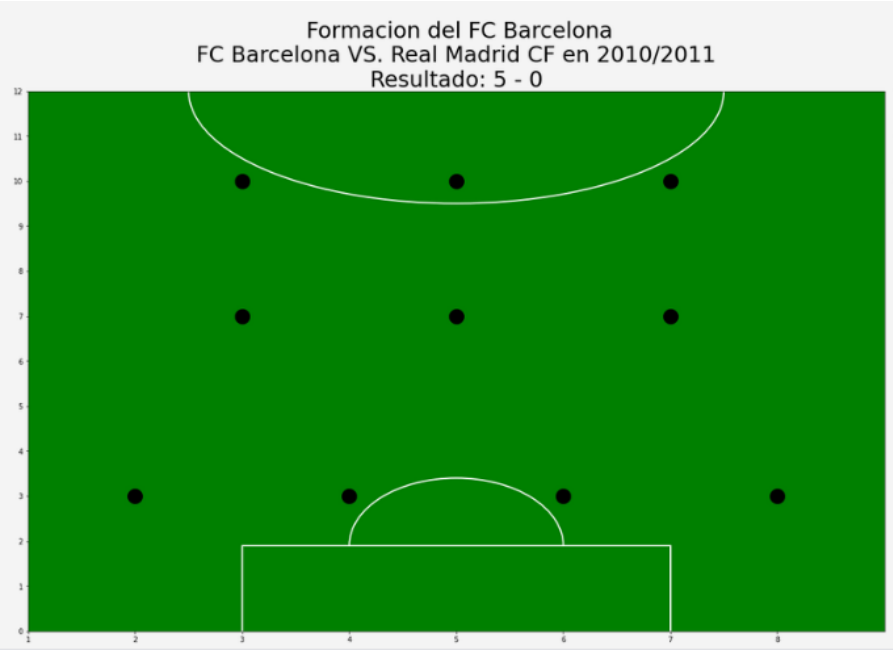

Figura 8. *Alineación del FC Barcelona vs. Madrid CF en 2010/2011*

Entonces, si nos fijamos en la Figura 9, podemos ver que esta alineación seria un claro 4-3-3. Entonces, realizando un análisis exploratorio un poco mas amplio, se define una función que, dada esta altura y en el campo de los jugadores, cuenta el número de jugadores que:

- Altura y inferior a 4
- Altura y entre 4 y 9
- Altura y mayor a 9

Haciendo esto y ordenando las formaciones, obtenemos que en la tabla Match existen las formaciones que se pueden ver en la Figura 10.

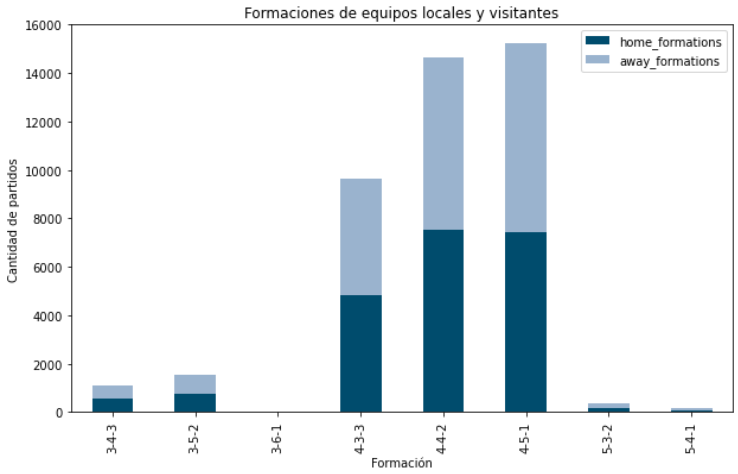

Figura 9: *Formaciones de equipos en el dataset*

Se aprecia en la Figura 10 que hay formaciones como la 3-6-1, la 5-3-2 y la 5-4-1 que aparecen un número muy pequeño en comparación a las demás formaciones.

3.2.2. Transformar los datos

En primer lugar, excluimos los partidos cuyo resultado sea empate. Se elimina del dataframe original los casos en los que el partido terminó en empate. Posteriormente se identifica el ganador y su formación. A continuación, se crea un nuevo dataframe iterando sobre cada fila del dataframe inicial, donde se extraen los atributos individuales de cada jugador, tanto ganador como perdedor, para cada partido. Es decir, el proceso seria el siguiente:

- Nos quedamos con la formación del equipo ganador (en caso de empate, datos fuera).
- Dado los *api_id* de los jugadores del equipo ganador:
    - De cada jugador se ponen sus 42 atributos en esa temporada en la que se juega el partido. (winner_player1_stat1 : winner_player_11_stat40)
- Se repite con el equipo perdedor

Formando un total de 882 variables para cada partido

3.2.3. Preprocesado

1. **One-Hot Encoding**: Codificamos las posiciones con un vector One-Hot Encoding. Cada posición se convierte en una nueva columna siendo esta uina variable binaria (problema multiclasse)
2. **División Train/Test**
3. **Estandarización MinMaxScaler**
4. Se han probado las siguientes **transformaciones** sobre los datos
    a. Oversample de las posiciones con menor número de apariciones.
    b. Hacer PCA (var expl > 97.5%)

c. Seleccionar características con el algoritmo Boruta.

Para cada una de las transformaciones, se realiza un modelado con:

- Regresión Lineal
- Random Forest
- Red Neuronal

5. Resultados y discusión

5.1. Resultados en la predicción de la posición del jugador en base a sus atributos

MODELO	*ACCURACY*	*F1-SCORE*
Linear Regresion	0.32	0.58
Random Forest	0.94	0.97
Neural Network	0.97	0.99

Tabla 1: *Resultados de los modelos tratando de predecir las posibles posiciones de cada jugador*

Se aprecia a la Tabla 1, que los mejores resultados obtenidos se obtienen con la Red Neuronal. Esta red neuronal ha sido diseñado tal y como se muestra a la Figura 10.

```
# Build a neural network to predict the positions
Zlatanizer = tf.keras.Sequential([
    Dense(256, activation='elu', input_dim=X_train.shape[1]),
    Dropout(0.15),
    Dense(512, activation='elu'),
    Dropout(0.15),
    Dense(256, activation='elu'),
    Dropout(0.15),
    Dense(128, activation='elu'),
    Dropout(0.1),
    Dense(64, activation='elu'),
    Dropout(0.1),
    Dense(y_train.shape[1], activation='sigmoid')
])


# Compilar el modelo
Zlatanizer.compile(optimizer=tf.keras.optimizers.Nadam(learning_rate=0.001),
    loss='binary_crossentropy',
    metrics=['accuracy'])

# Early Stopping
early_stopping = EarlyStopping(monitor='val_loss', patience=250, verbose=1, mode='min', restore_best_weights=True)

# Entrenar el modelo
history = Zlatanizer.fit(X_train, y_train, epochs=20000, batch_size=512, validation_data=(X_test, y_test),
                         callbacks=[early_stopping])
```

Figura 10: *Diseño de la red neuronal que obtiene las mejores métricas prediciendo las posiciones de cada jugador en base a sus atributos.*

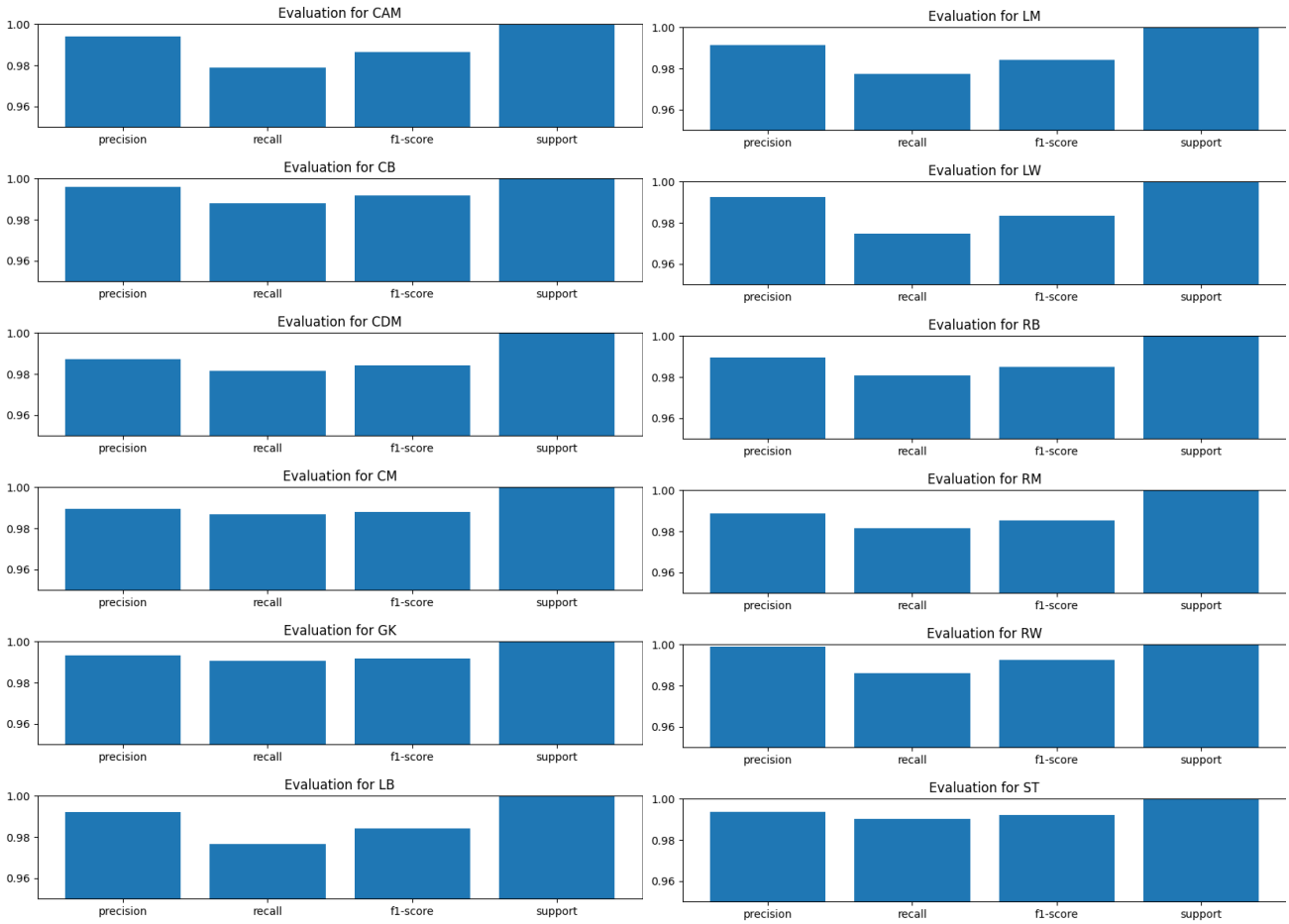

Figura 11: *Métricas obtenidas por la red neuronal en cada posición*

Se observa en la Figura 11 como la red obtiene unos resultados bastante equilibrados en la predicción de todas las posiciones. Cabe destacar que los mejores resultados se han obtenido sin hacer *oversample* de las clases minoritarias.

Se puede definir una serie de ejemplos por las que este modelo puede ser interesante, aquí se describe algunos:

- Identificación de talentos: El modelo podría ayudar a los entrenadores y los equipos a identificar jugadores con habilidades versátiles, capaces de jugar en múltiples posiciones.

- Descubrir nuevas posiciones: El modelo podría utilizarse para identificar, estudiando en mayor medida los resultados del modelo y el umbral, nuevas posiciones en las que un jugador puede jugar.

- Evaluación de jugadores lesionados: El modelo podría ayudar a los entrenadores a evaluar qué posiciones podrían ser cubiertas por otros jugadores mientras un jugador lesionado se recupera.

5.2. Resultados en la optimización de la formación

Los mejores resultados obtenidos en este apartado se consiguen haciendo oversample.

MODELO	ACCURACY	F1-SCORE
Linear Regresion	0.7043	0.7188
Random Forest	0.7958	0.7933
Neural Network	0.7829	0.7823

Tabla 2: *Resultados obtenidos por los tres modelos haciendo oversample de los datos*

En esta tabla, puede parecer que el Random Forest obtiene unos mejores resultados que la red neuronal en la estimación de las formaciones; sin embargo, tal y como se puede ver en la Figura 12, la red neuronal consigue balancear mejor sus predicciones entre las diferentes formaciones.

RANDOM FOREST ***NEURAL NETWORK***

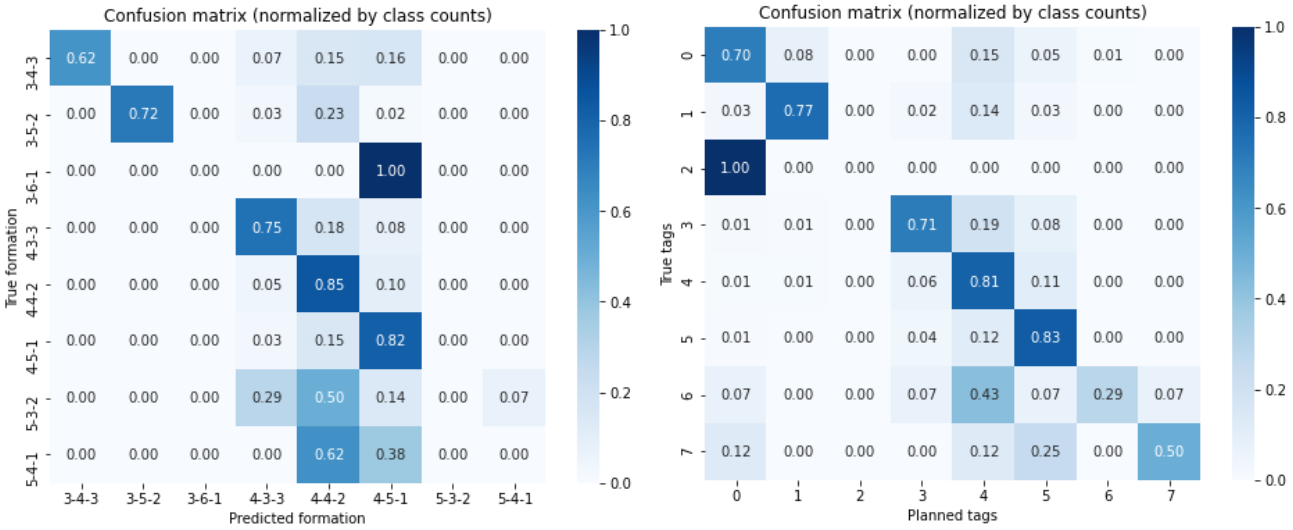

Figura 12: *Matriz de confusión de los resultados del Random Forest y RNN estimando la mejor formación*

En la siguiente Figura 13 se puede observar cual ha sido el diseño de la red neuronal que ha obtenido estos resultados.

```
bAIcelona = models.Sequential([
    layers.Dense(1024, activation='elu', input_dim=X_train_resampled.shape[1]),
    layers.BatchNormalization(),
    layers.Dropout(0.5),

    layers.Dense(2048, activation='elu'),
    layers.BatchNormalization(),
    layers.Dropout(0.6),

    layers.Dense(1024, activation='elu'),
    layers.BatchNormalization(),
    layers.Dropout(0.5),

    layers.Dense(512, activation='elu'),
    layers.BatchNormalization(),
    layers.Dropout(0.4),

    layers.Dense(256, activation='elu'),
    layers.BatchNormalization(),
    layers.Dropout(0.25),

    layers.Dense(128, activation='elu'),
    layers.BatchNormalization(),
    layers.Dropout(0.15),

    layers.Dense(y_resampled.shape[1], activation='softmax')
])

optimizer = tf.keras.optimizers.Adamax(learning_rate=0.0075)
early_stopping = EarlyStopping(monitor='val_loss', patience=35, verbose=1, mode='min', restore_best_weights=True)
reduce_lr = tf.keras.callbacks.ReduceLROnPlateau(monitor='val_loss', factor=0.2, patience=10, verbose=1)
bAIcelona.compile(optimizer=optimizer, loss='categorical_crossentropy', metrics=['accuracy'])

history = bAIcelona.fit(X_train_resampled, y_resampled, epochs=1000, batch_size=1024,
                    validation_data=(X_test_scaled, y_test),
                    callbacks=[early_stopping, reduce_lr])
```

Figura 13: *Diseño de la Red Neuronal que estima las mejores formaciones en base a los atributos de todos los jugadores.*

6. *Conclusiones y futura proyección*

A través del proyecto "Field Wizard: Predicting Positions and Formations for Success on the Soccer Field", se ha logrado arrojar luz significativa sobre las estrategias óptimas en el terreno del fútbol profesional, proporcionando un marco analítico sólido y dinámico. El desarrollo consistió en dos modelos principales articulados meticulosamente para optimizar tanto la asignación de posiciones individuales como la formación del equipo en su conjunto. El primer modelo exhibió una notable destreza al alcanzar una exactitud del 0.97 y un puntaje F1 de 0.99, demostrando ser una herramienta poderosa y precisa para el análisis de la asignación de posiciones con base en atributos individuales. Por su parte, el segundo modelo también manifestó una robustez considerable, permitiendo la optimización de las formaciones del equipo a través de una evaluación dual que incluye tanto los atributos propios como los del equipo contrario, respaldado por una exactitud de 0.783 y un puntaje F1 de 0.782.Estos resultados notables subrayan no solo la viabilidad sino también la alta eficacia de utilizar un enfoque basado en datos para estrategias en el campo de fútbol.

Una área prometedora para futuras investigaciones puede ser la incorporación de variables adicionales en los modelos, como datos meteorológicos y condiciones del terreno, para ofrecer una visión aún más holística y precisa. Otra vía para la expansión futura sería el desarrollo de una interfaz gráfica intuitiva que facilite a los entrenadores y analistas la utilización de estos modelos en un entorno en tiempo real, proporcionando recomendaciones estratégicas ágiles y basadas en datos durante los partidos.