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

In the contemporary era, we find ourselves immersed in a data revolution where every aspect of our society is being transformed by the power of advanced analytics and machine learning. Sport, and particularly football, is no exception to this trend; data has come to play a pivotal role in strategic decision making, not only at the business and management level, but also directly on the field of play. Data-driven optimisation not only offers a deep understanding of the dynamics of the game, but also has the potential to raise the level of competition to unprecedented heights, promoting increasingly sophisticated and effective strategies.

Within this context, customisation of training and team tactics based on individual and collective player characteristics has become a cornerstone for success. Coaches and analysts are continually looking for innovative ways to use data to gain competitive advantage, exploring metrics ranging from biomechanical analysis to real-time performance statistics.

Despite significant progress in this area, there is still vast uncharted territory and great potential to deepen the integration of data science in football. This work arises from the need to explore deeper and more nuanced ways of using data to influence game strategies. It focuses on the development of predictive models that can help optimise player positional allocation and team formation, taking into account a wide range of individual and collective attributes, and contextual factors of the match to be played.

The research presented here focuses on the creation of two machine learning models; the first one, aimed at predicting the possible positions of a player based on his individual

attributes, and the second one, aimed at optimising the team formation for a specific match, considering the attributes of the own and the opposing team's players. Through this dual approach, we aim to provide a powerful and flexible tool that can be a valuable guide for teams in planning their strategies, both in training and in competitions.

In this paper, we will outline the process of building these models, presenting both the methodology employed and the results obtained, in order to highlight not only the feasibility but also the effectiveness of this approach for the optimisation of strategies in football.

2. Data

In this project, data sources from the open data platform Kaggle are used. Kaggle is an online platform where you can compete in data analysis competitions, collaborate on projects, and access a large set of public datasets and educational resources for learning. On this platform, we can find the following data sources:

1 - European Soccer Database is a comprehensive database designed for data analysis and machine learning in the context of European football. The database contains detailed information on more than 25,000 matches and more than 10,000 players from 11 European countries, covering the seasons from 2008 to 2016. It uses dataframes highlighted in Figure 1 below.

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

Figure 1: *Tables from the European Soccer Database*

- In the Player table we have the following variables: api_id, player name, height and weight. It is shown in Figure 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

Figure 2*:Player table information*

- In the Player_*Attributes* table you will find a total of 38 attributes of a player recorded at a certain time of the season. That is, a player usually has several records throughout his career. We can see 5 examples below:

o sprint_speed: Attribute of the player's top speed.

o agility: Attribute of the player's agility.

o reactions: Attribute of the player's reactions.

o balance: Player's balance attribute.

o shot_power: Attribute of the player's shot power.

- In the *Match* table we find data on matches played. The information we are interested in from the following dataset is:

  o Api_id of the teams that played the match.

  o Season in which the match was played.

  o Outcome (home goals and away goals).

  o Api_id of the players of each team that played in the match.

  o Position on the y-axis of each player (height of the field).

- The Team table is only used to transform the api of the home and away team to the real name of the team, as we do not have this information in the previous table.

2 - *FIFA complete player dataset*, where detailed information on the players available in the Career mode of the FIFA videogame versions from FIFA 15 to FIFA 22 is collected and presented in a file called "players_YEAR.csv". The detail of this file allows for multiple comparisons of the same player across the last 8 versions of the game, covering more than 100 individual attributes for each player. Of all the datasets contained in this database, we will use the two that refer to the year 2015 and 2016:

- players_2015.csv
- players_2016.csv

- Within these two dataframes, we are interested in the two variables shown in Figure 3.

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

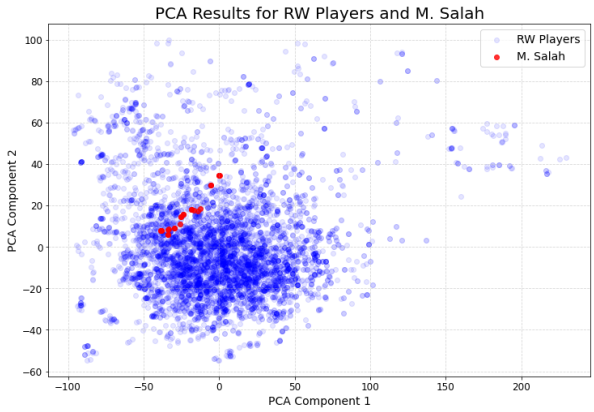
***Figure 3**: Relevant FIFA dataset information*

3. Materials and Methods

3.1. Predicting positions based on a player's attributes

3.1.1. Data Deletion and Transformation

In this phase, incorrect values identified in the columns 'defensive_work_rate' and 'attacking_work_rate' were removed. Subsequently, the values were categorised and the 'id' column was dispensed with. For a more detailed view, the data was grouped by player name and the average of the player attributes per season was calculated. It can be intuited that players, throughout their career, vary in attributes and may also vary in position, which is why we kept their seasonal average. If we keep as many rows as possible, we could identify patterns and trends in a player's attributes in different contexts (teams, age of the player, etc...). This will help the model to better identify patterns to predict position.



***Figur3 4**: 1st and 2nd main component of the records of the player Mohamed Salah*

It can be seen from Figure 4 that M. salah has varied its information throughout the records we have in our data.

3.1.2. *FIFA Database Data Integration*

A substantial step is the incorporation of player position information from the FIFA database. A transformation of the player names was performed to match the records in the FIFA dataset, allowing an accurate merging of the FIFA 15 and FIFA 16 datasets with the main dataset. Remember that a player can have different positions associated with him.

As we only have FIFA information (therefore positions) for players in 2015 and 2016, there are players in our data that do not have associated positions. As can be seen in Figure 5, the number of players with and without positions is similar; that is, approximately half of our player records have no position and, therefore, we do not know their playing position on the field. We will try to predict this using a model.

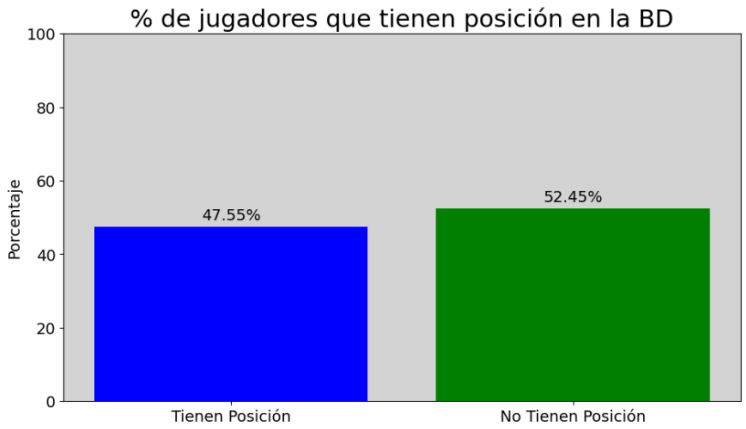


Figure 5: *Percentage of players with and without position in the data*

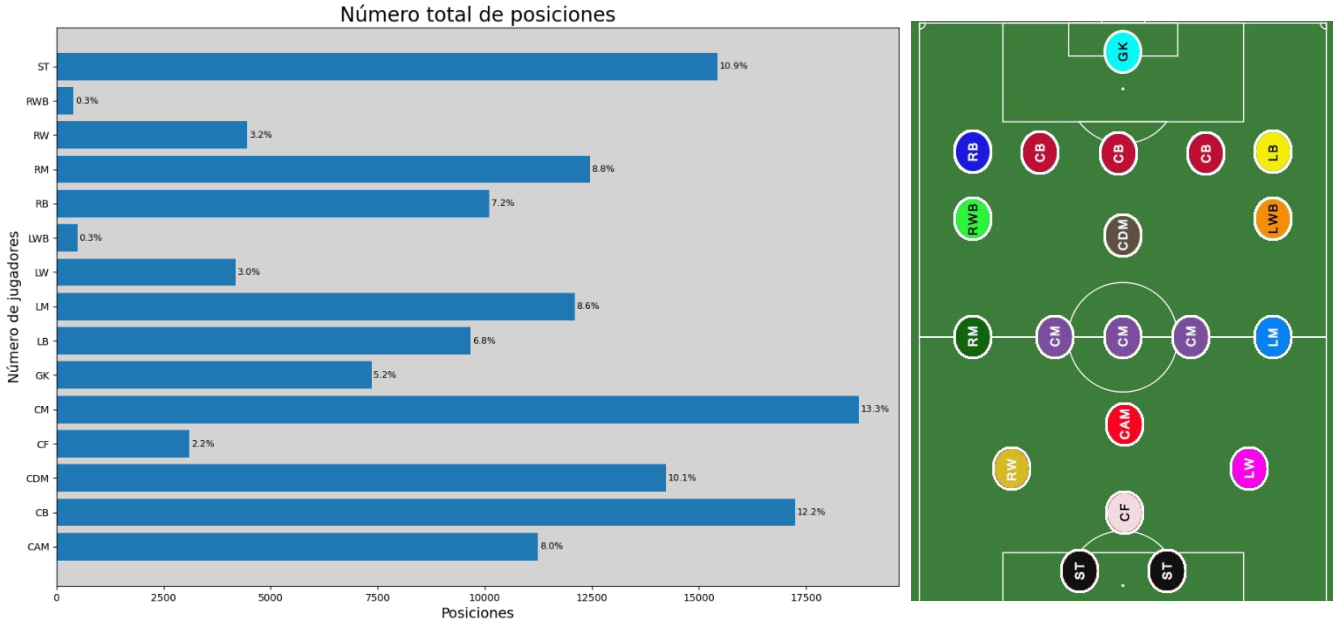


Figure 6: *Percentage of each position on the pitch and its exact location on the pitch*

In Figure 6, we can see that there are positions with a very small number of occurrences compared to the others, such as LWB, RWB and CF. So, given the improvement seen, the positions "LWB" has been changed to "RB", "RWB" to "RB" and "CF" to "ST".

3.1.3. Preprocesado:

1. **One-Hot Encoding**: We encode the positions with a One-Hot Encoding vector. Each position is converted into a new column being a binary variable. This is a multi-label problem.

2. Train/Test Division

Length Train = 67009

Length Test = 16753

3. MinMaxScaler standardisation

4. Feature Engeneering: Given the 38 attributes we have in our data and, through feature engineering, we create 10 new attributes making a total of 48 attributes. We can see the confusion matrix of the player attributes.

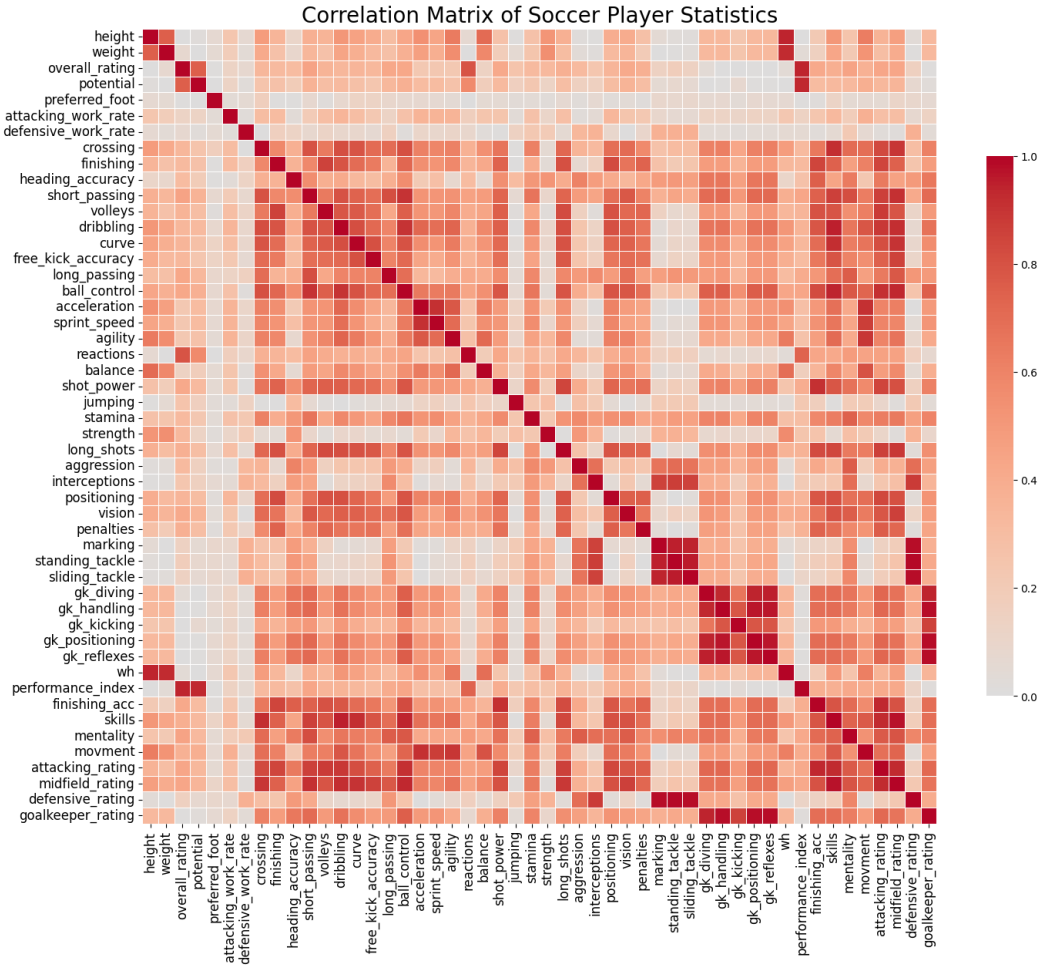


Figure 7: *Correlation matrix of data attributes*

3.1.4. Model construction

The following models are tested in an attempt to estimate player positions based on player attributes:

- Linear Regression

- Random Forest

- Neural Network

3.1.5. Predictions

Once the model was trained, we made the predictions of the players that did not have an associated position in our DB. In this way, finally, we exported the data with the player's name, attributes and possible positions.

3.2. Optimising team building

3.2.1. Create the training dataframe

In the match database, we do not have the formations used by the team in the match. However, we do have the position on the y-axis of each player (height of the pitch).

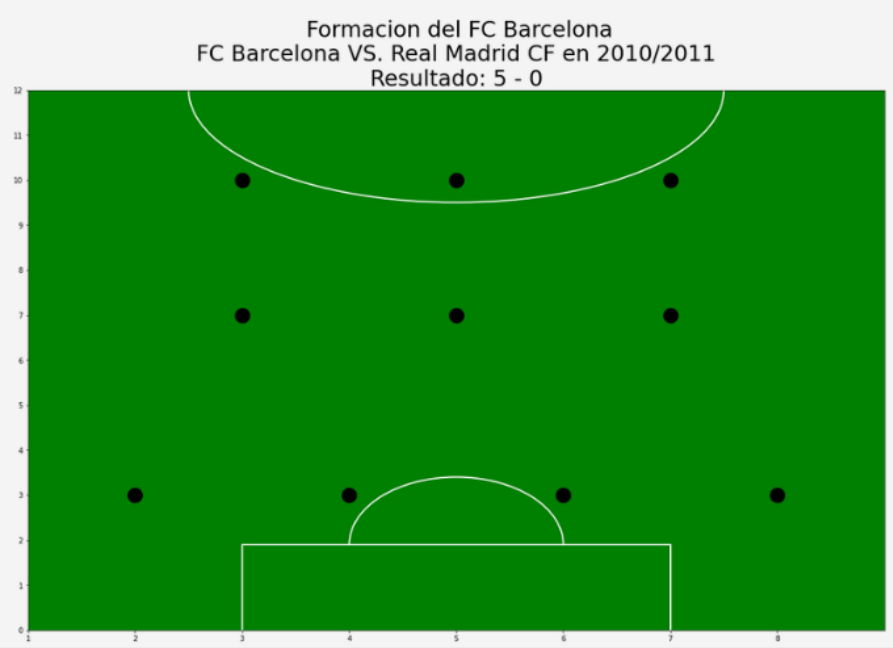


Figure 8. *FC Barcelona vs. Madrid CF line-up in 2010/2011*

So, if we look at Figure 8, we can see that this line-up would be a clear 4-3-3. Then, doing a little more exploratory analysis, we define a function that, given this height and in the field of players, counts the number of players that:

- Height and less than 4

- Height and between 4 and 9

- Height and greater than 9

By doing this and ordering the formations, we obtain that in the Match table there are the formations that can be seen in Figure 10.

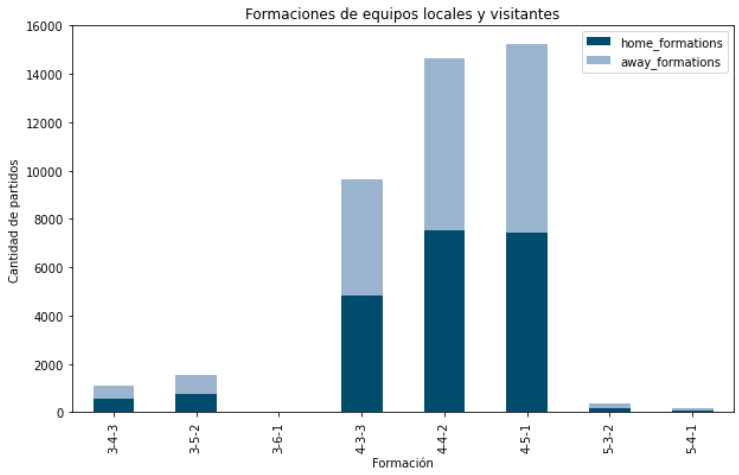


Figure 9: *Team formations in the dataset*

It can be seen in Figure 9 that there are formations such as 3-6-1, 5-3-2 and 5-4-1 that appear in very small numbers compared to the other formations.

3.2.2. Transforming data

First, we exclude matches where the result is a draw. Cases where the match ended in a draw are removed from the original dataframe. We then identify the winner and its formation. Next, a new dataframe is created by iterating over each row of the initial dataframe, where the individual attributes of each player, both winner and loser, are extracted for each match. In other words, the process would be as follows:

- We are left with the formation of the winning team (in case of a tie, data out).
- Given the api_id of the players of the winning team:
    - For each player we put his 42 attributes in that season in which the match is played (winner_player1_stat1 : winner_player_11_stat40).
- Repeated with the losing team

Forming a total of 882 variables for each match.

3.2.3. Pre-processing

1. **One-Hot Encoding**: We encode the positions with a One-Hot Encoding vector. Each position is converted into a new column being a binary variable (multiclass problem).
2. **Train/Test Division**
3. **MinMaxScaler standardisation**
4. The following **transformations** on the data have been tested
    - Oversample of the positions with the lowest number of occurrences.
    - Make PCA (var expl > 97.5%).
    - Select features with the Boruta algorithm.

For each of the transformations, modelling is performed with:

- Linear Regression

- Random Forest

- Neural Network

5. Results and discussion

5.1. Results in predicting player position based on player attributes

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

Table 1: *Results of the models trying to predict the possible positions of each player.*

It can be seen from Table 1 that the best results are obtained with the Neural Network. This neural network has been designed as shown in Figure 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])
```

Figure 10: *Design of the neural network that obtains the best metrics by predicting the positions of each player based on their attributes.*

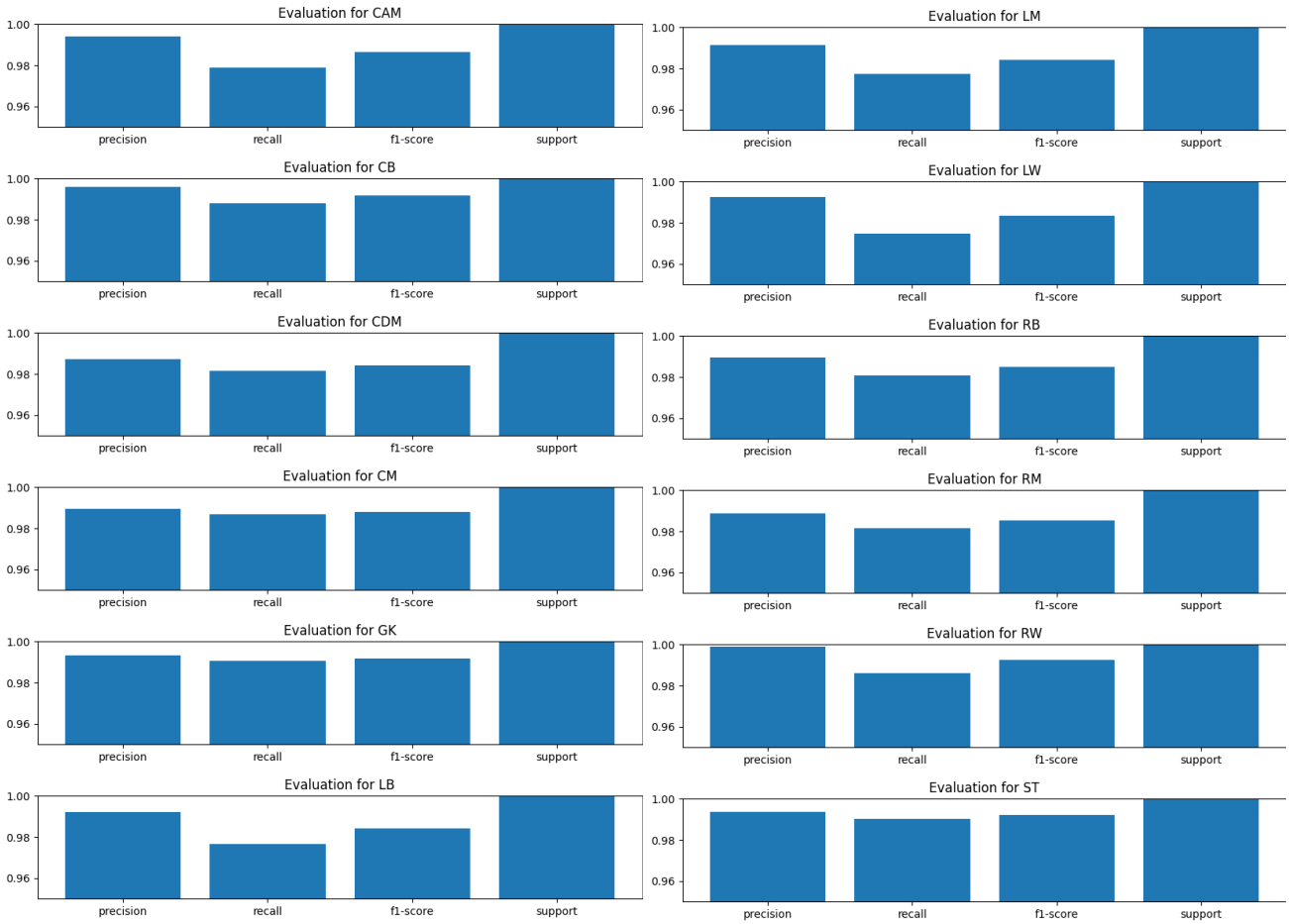


Figure 11: *Metrics obtained by the neural network at each position*

Figure 11 shows how the network obtains fairly balanced results in the prediction of all positions. It is worth noting that the best results have been obtained without oversampling the minority classes.

A number of examples can be defined for which this model can be interesting, some of which are described here:

- Talent identification: The model could help coaches and teams to identify players with versatile skills, able to play in multiple positions.

- Discovering new positions: The model could be used to identify, by further studying the model and threshold results, new positions in which a player can play.

- Assessing injured players: The model could help coaches assess which positions could be covered by other players while an injured player recovers.

5.2. Results in optimising line-up

The best results in this section are achieved by oversampling.

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

Table 2: *Results obtained by the three models by oversampling the data.*

From this table, it may appear that the Random Forest performs better than the neural network in the estimation of the formations; however, as can be seen in Figure 12, the neural network manages to balance its predictions better between the different formations.

RANDOM FOREST *NEURAL NETWORK*

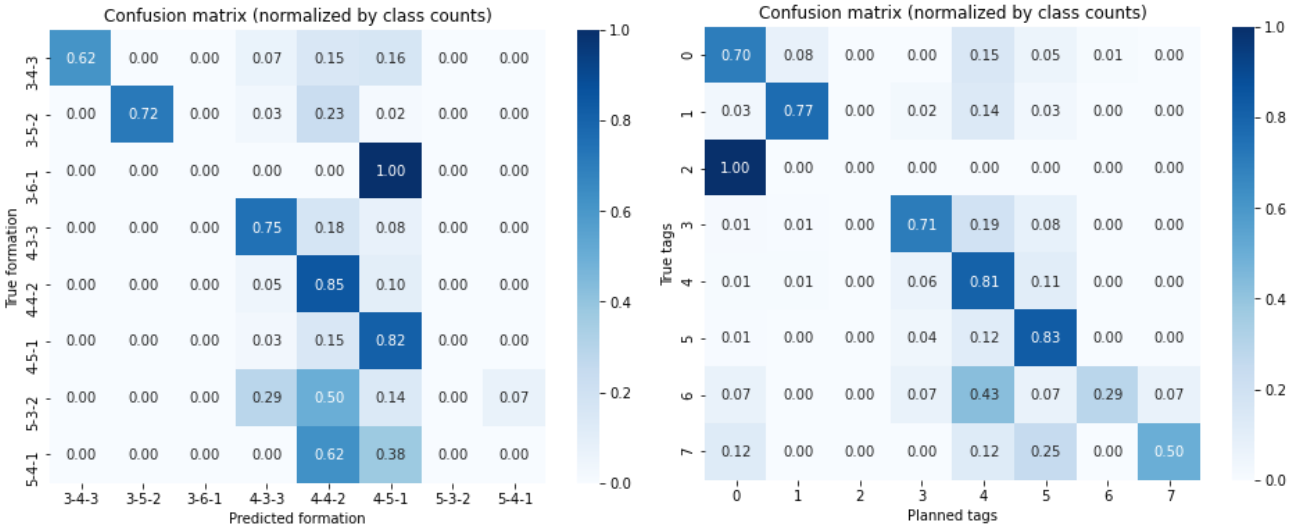


Figure 12: *Matriz de confusión de los resultados del Random Forest y RNN estimando la mejor formación*

Figure 13 below shows the neural network design that obtained these results.

```
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])
```

Figure 13: *Design of the Neural Network that estimates the best formations based on the attributes of all players.*

6. Conclusions and future projection

Through the project "Field Wizard: Predicting Positions and Formations for Success on the Soccer Field", significant light has been shed on optimal on-field strategies in professional football, providing a robust and dynamic analytical framework. The project has shed significant light on optimal on-field strategies in professional football by providing a robust and dynamic analytical framework. The development consisted of two main models meticulously articulated to optimise both the allocation of individual positions and the formation of the team as a whole. The first model exhibited remarkable proficiency in achieving an accuracy of 0.97 and an F1 score of 0.99, proving to be a powerful and accurate tool for the analysis of position allocation based on individual attributes. The second model also manifested considerable robustness, allowing the optimisation of team formations through a dual assessment including both own and opposing team attributes, supported by an accuracy of 0.783 and an F1 score of 0.782. These remarkable results underline not only the feasibility but also the high efficacy of using a data-driven approach for on-field strategies in football.

A promising area for future research may be the incorporation of additional variables into the models, such as weather data and field conditions, to provide an even more holistic and accurate view. Another avenue for future expansion would be the development of an intuitive graphical interface that makes it easier for coaches and analysts to use these models in a real-time environment, providing agile, data-driven strategy recommendations during matches.