Soccer Outcome Predictions through Graph Theory and Historical Statistics: An Advanced Sports Analytical Approach

Cecilia Alberti, Valentina Varetto
Department of Rice Center for Engineering Leadership
Applied Statistics and Data Science for Engineering Leaders Course
Rice University

Abstract

The aim of this project is to develop a machine learning model to predict the outcome of a soccer match from La Liga based on variables derived from Graph Theory and historical statistics from the teams. The data for the model is collected from StatsBomb, with 868 matches from 'La Liga'. Each match is studied as a graph, where the mean positions of the players are considered as the nodes, and the passes between them as edges. The selected descriptors from Graph Theory are weighted density, diameter, weighted transitivity and average clustering. On the other hand, historic winning, losing and drawing rates for both teams are considered, together with their average goals. Additionally, the model contains descriptors related to the historical outcomes of the matches between the two teams.

Table of contents

1. Introduction

Soccer, globally acclaimed as the preeminent sport, transcends geographical boundaries, cultural diversities and linguistic barriers. Its popularity can be attributed to several factors. One of them is the accessibility and simplicity of the game, which offers an easy comprehension and engagement of the players and fans. Also, being a team sport demands a collective effort and coordination to attain a shared objective, thus fostering a spirit of unity and collaboration.

Moreover, an essential attribute of soccer is its inherent unpredictability. The low scoring nature compared to other team sports such as basketball or American football, makes it more difficult to predict. A single goal can drastically change the outcome of a match, and often games are decided by just one or two key moments in the match. Additionally, factors such as player performance, referee decisions, injuries, substitutions and even weather conditions can play a role in the final outcome.

In the past few years, the use of data analytics in soccer has become increasingly prevalent, with many people attempting to predict match outcomes. However, given the complexity and numerous variables involved, these predictions often remain uncertain.

In light of this unpredictability, the presented study aims to delve deeper into understanding how different variables can influence the outcome of a soccer match. By analyzing aspects such as how collaborative the team plays, how the team is usually displayed in the field, and historical data we develop a comprehensive model. This model not only quantifies the impact of each variable, but also predicts match outcomes with a greater degree of accuracy. Despite the inherent unpredictability of soccer, our approach harnesses the power of data analytics and discrete mathematics through graph theory to offer an accurate prediction of soccer matches.

2. Project Description

2.1 Preliminary Exploration

2.1.1 Visual analysis of a particular player's passes throughout a match

In this first explorative instance, the incidence of a particular player on the result of a match is explored through a visual analysis of his passes throughout a specific match. The player's passes are grouped into three categories: 'Complete', 'Incomplete' and 'Offside'; which can be visually distinguished with the colors green, red and yellow, respectively.

For this visual analysis, a function called *filter_player_passes()* is developed in order to obtain a visual representation of a player's passes by giving the inputs: tournament id, match id, team and player. In this way, for a given match in a specific tournament, the user can select a team and a player and obtain a static visualization of the player's performance that integrates a heatmap with the outcomes of his passes in the match. An example is shown below, with Messi's passes in the Barcelona vs. Celta Vigo match from La Liga 2020/2021.

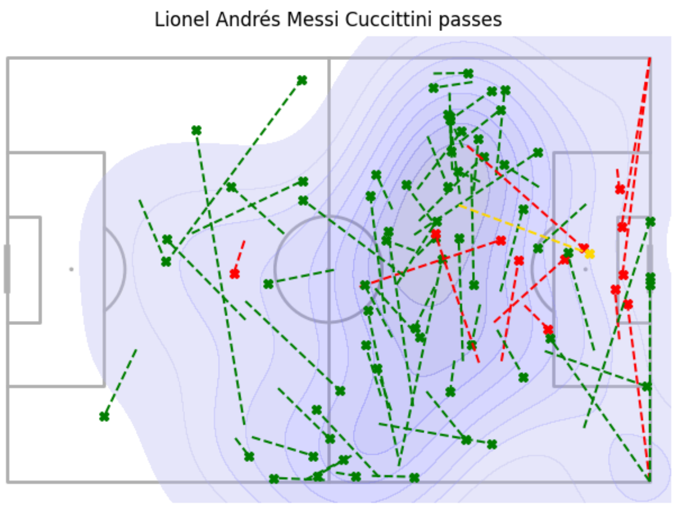


Fig. 1: Heatmap and outcomes of Messi's passes in Barcelona vs. Celta Vigo - La Liga 2020/2021.

2.1.2 Visual analysis of shots and goals of a match

This exploration is aimed to gain visual insights of the ball's positions in shots and goals' instances. For this purpose, a function, *shots_function()*, is designed in order to obtain the positions of shots and goals for a specific match in a tournament. By the inputs tournament id, team and match_id, the user receives a graph that depicts with full-colored circles the positions for goals, and with transparent circles, the positions for shots. An example can be seen in the figure below, which portrays the shots and goals of the Barcelona vs. Celta Vigo match in La Liga 2020/2021.

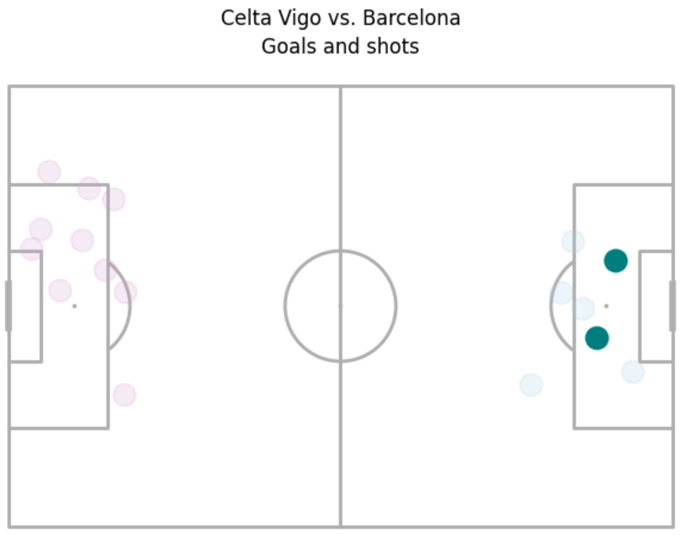


Fig. 2: Visual representation of shots and goals positions in Barcelona vs. Celta Vigo - La Liga 2020/2021.

2.1.3 Visual analysis of the ball's sequence before a goal

In order to obtain more dynamic insights of the ball's position before a goal, a visual analysis of the ball's sequence before a goal is made. The purpose of this exploration is to visually study the incidence of the ball's trajectory before a shot in its resulting outcome.

The first step of this analysis consists of filtering from the dataset of a given match the rows that correspond to goals, and registering the position of the ball in previous observations (no more than 2 minutes earlier) with the intention of recreating the ball's trajectory. These results are stored in a dictionary called *ball_sequence_before_goal*, that has goals' indexes as keys and ball's sequences as values.

The second step is removing from the dictionary those keys that contain less than two values, since it means that for those goals it has not been possible to track the trajectory of the ball and, therefore, can not provide useful information.

Finally, by using the *imageio* library, a gif of the sequence of the ball before a goal is created to replicate the dynamic of the ball before the shot. An example of a ball's sequence can be visualized in the following figure, conveying the trajectory of the ball before the goal made in the 10th minute in the Barcelona vs. Celta Vigo match from La Liga 2020/2021.

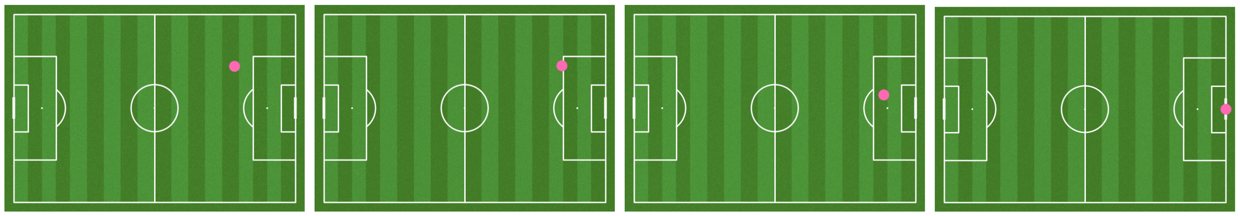


Fig. 3: Ball's sequence before the goal made in the 10th minute of the Barcelona vs. Celta Vigo match - La Liga 2020/2021.

2.1.4 Visual analysis of a team's dynamic throughout a tournament

The objective of this exploration is to generate a static representation of a team's dynamic throughout a match as a network. This exploration is carried out for the Barcelona team in La Liga 2020/2021.

For this purpose, the first step is to calculate each player's mean position at the end of the match in order to locate the player in the field as a node. The second step is to calculate the amount of passes between each pair of players to see the links between the nodes. In order to represent the importance of a link, it is necessary to vary the width of the link according to the amount of passes made by the players at stake in comparison with the maximum amount of passes between any two players throughout the match.

In this way, it is possible to generate a network to visually represent the dynamic of a determined team during a match, where the nodes stand for the players' mean positions and the links, the passes between them.

This visual representation is carried out for 34 of Barcelona's matches in La Liga 2020/2021. One example is shown in the figure below. With the use of the *imageio* library, a gif is created to portray the difference and similarities in Barcelona's team dynamic across the 34 different matches studied.

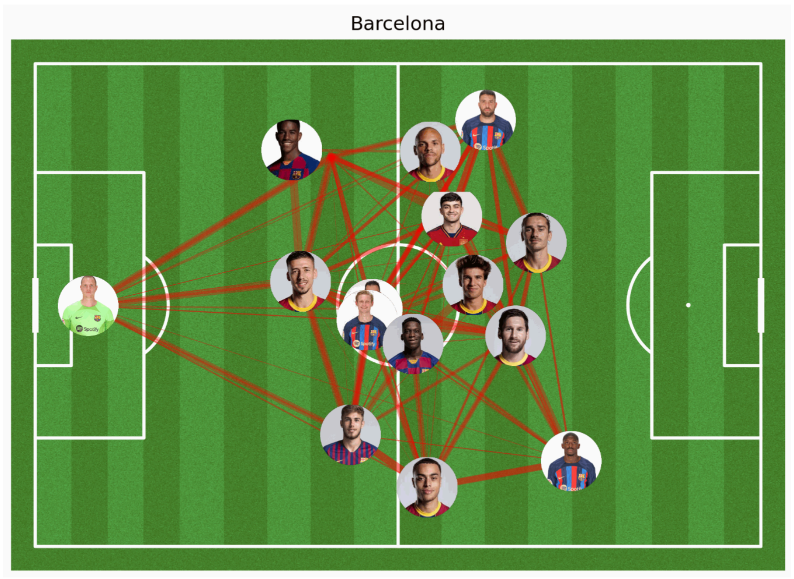


Fig. 4: Visual representation of Barcelona's team dynamic during a match from La Liga 2020/2021.

2.2 Graph Theory applied to Soccer

2.2.1 Representation of a match's dynamic as a graph

Considering the visual analysis performed in the explorative instance together with the literature review, it is decided to study the dynamic of a match with Graph Theory since it allows to generate a static representation of the players' positions and their passes as well.

In discrete mathematics, a graph is a collection of objects called nodes, connected by links called edges. While nodes are individual entities, the edges represent connections or relationships between these entities.

Graphs can be undirected, when edges have no direction, indicating a two-way relationship; or directed, where edges have a direction and each edge can only be transited in that direction. Additionally, a graph can be unweighted, where all edges are the same; or weighted, where edges have their associated values.

When applying Graph Theory to soccer, the players can be represented as the nodes of the graph. The size of the node - that stands for the weight of the node - depends on the participation of the player in the match based on the amount of passes that he was involved in. On the other hand, the edges are given by the passes between the players, and their width - standing for the weight of the edge - expresses the relative importance of the link between the pair of players at stake compared to the maximum number of passes made between two players.

In this scenario, the graph is an undirected graph, since the players can make passes in both ways, and it is a weighted graph, since the edges would have an assigned weight depending on the quantity of passes made between them.

The table below summarizes the interpretation of Graph Theory applied to soccer matches contemplated for this project.

Graph element	**Soccer application**	**Calculation**
$Node_i$ (position)	$Mean\ position\ of\ player_i$	$(xi\ ,\ yi) = (\frac{\sum_{k=1}^{Ni} xk}{Ni}, \frac{\sum_{k=1}^{Ni} yk}{Ni})$, being (xk, yk) the coordinates of player i in the k pass and Ni the total passes of the player.
$Weight\ of\ node_i$	$Importance\ of\ player_i$	$\frac{Ni}{\sum_{i=1}^{P} Ni}$ being Ni the total passes of player i and P the total number of players in the match.
$Edge_{ij}$ (position)	$Pass\ between\ players_{i\,\&j}$	$Xij = (xi,\ xj)$ $Yij = (yi,\ yj)$ being (xi, yi) and (xj, yj) the mean positions of players i and j, respectively.
$Weight\ of\ edge_{ij}$	$Importance\ of\ edge_{i\,j}$	N_{ij}, being Nij the total number of passes made between player i and player j.

Table 1: Interpretation of the elements of Graph Theory applied to soccer.

A function called *graph_theory_match()* generates a visual representation of the resulting graph from a given match by calculating the positions for the nodes and edges as already explained and their corresponding weights. An example can be seen in the figure below.

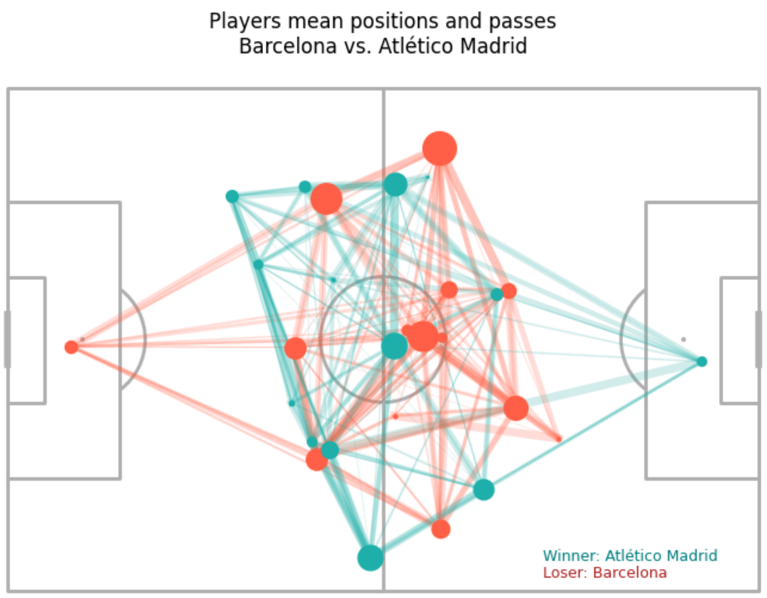


Fig. 5: Visual representation of Graph Theory applied to soccer - Barcelona vs. Atletico Madrid - La Liga 2020/2021

2.2.2 Generation of the data frame with graph's properties

Data set

The data utilized for this project is sourced from StatsBomb. Specifically, we focus on event-level data, which contains various competitions and varies match quantity within each competition.

The following table contains the competitions with their respective number of matches.

Competitions	Nr. of matches
Bundesliga	306
Champions League	17
Copa del Rey	3
FA Women's Super League	326
FIFA U20 World Cup	1
FIFA World Cup	147
Indian Super league	115
La Liga	868
Liga Profesional	2
Ligue 1	377
North American League	1
NWSL	36
Premier League	418
Serie A	381
UEFA Euro	51
UEFA Europa League	3
UEFA Women's Euro	31
Women's World Cup	116

Table 2: Quantity of matches per competition in StatsBomb

Considering that ‘La Liga’ is the competition with the highest number of matches, the project is developed with ‘La Liga’ competition.

Graph's properties

For the analysis, each team on the field is considered as a graph. Thus, there are characteristics of each team - and of each graph - that can influence the results. We decide to particularly analyze the properties of graph density, graph transitivity, average clustering, and graph diameter for each team.

The table below summarizes the interpretation of Graph Theory's properties applied to soccer matches contemplated for this project.

Graph element	**Soccer application**	**Calculation**
$Weighted\ density$	$Passing\ pairs\ (weighted)\ to\ total\ possible$	$\frac{\Sigma\ weights\ of\ all\ edges}{\frac{n(n-1)}{2}}$ being n the number of nodes
$Weighted\ transitivity$	$Passing\ triangles\ (weighted)\ to\ total\ possible$	$\#\ triangles\ = \frac{\Sigma nx.triangles(G)}{3}$ $triangle\ weight\ =\ min($*weight of edge* $u-v$, *weight of edge* $v-w$, *weight of edge* $u-w$) $total\ triangle\ weight\ = \Sigma\ triangle\ weigh$ $weighted\ transitivity\ = total\ weight\ /\ \#\ triangles$ being G the graph, u, v and w are nodes, nx.triangles is a function of NetworksX that returns a dictionary where each key is a node and the value is the number of triangles that include that node
$Diameter$	$Longest\ path\ between\ the\ mean\ position\ of\ two\ players_{i\ j}$	$nx.\ diameter(G)$ being G the graph and nx.diameter a function of NetworksX that returns the maximum distance between a pair of nodes that are connected
Average clustering	$Degree\ to\ which\ players\ tend\ to\ form\ tightly-knit\ groups$	$nx.\ average\_clustering(G,\ weight)$ being G the graph and nx.average_clustering(G) a function of NetworksX that returns the average clustering of the graph G; when specifying weight='weight' it considers the weight of the edges

Table 3: Interpretation of graphs' properties in soccer.

Having obtained the corresponding observations of La Liga matches in Stats Bomb and interpreted the application of Graph Theory in soccer, the following section explores how this data is prepared to obtain from the matches, their resulting graph's properties. The following figure visually summarizes the algorithm behind the function *graph_theory_match()*.

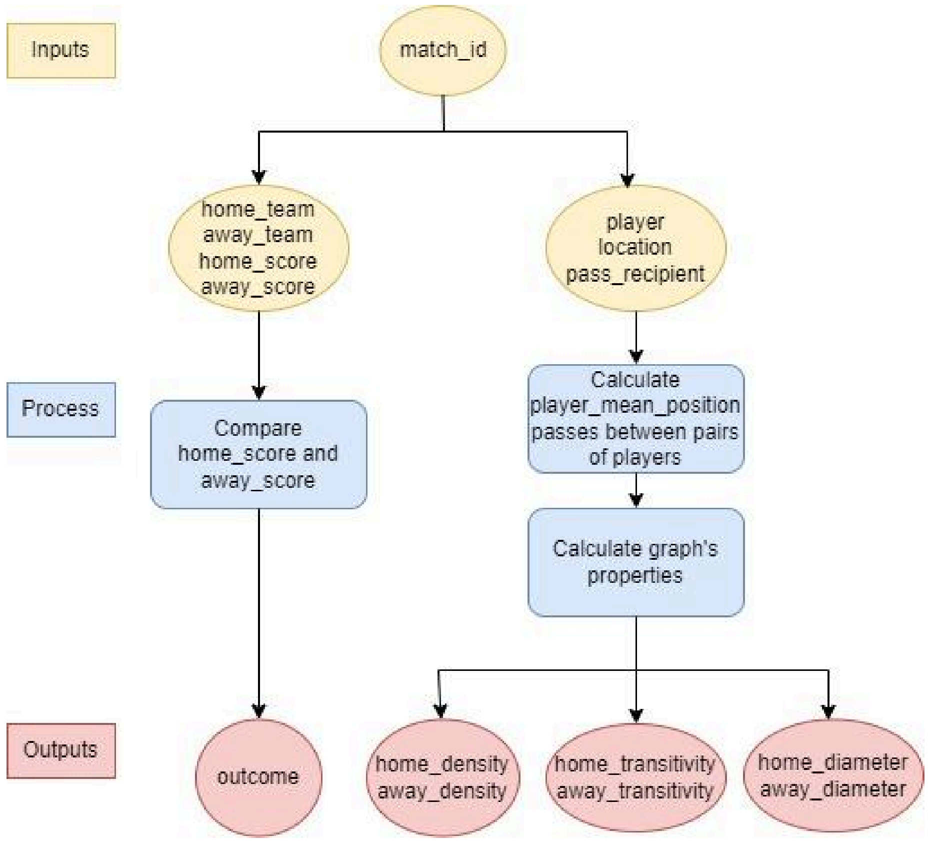


Fig. 6: Algorithm behind the graph_theory_match() function

As it is presented in the figure, the input for the function is the id of the match. From the id of the match, the function looks for the home and the away teams and their respective scores to determine the outcome of a match, which represents a discrete variable that varies between {home_team_winner; away_team_winner; tie}.

On the other hand, with the given id, the function looks for the specific event in Stats Bomb to obtain all the observations registered for that particular match. The function first calculates the mean position of every player (the position of the nodes) and the amount of passes between each pair of players (to locate the edges).

After that, the function proceeds to calculate the resulting graph's properties (density, transitivity and diameter) for each team.

This function is executed for every match in La Liga and the results are stored in a new dataframe, that has the following columns: {home_team; away_team; home_density; home_diameter; home_transitivity; away_density; away_diameter; away_transitivity; home_score; away_score; winner}.

2.3 Historical data in soccer

2.3.1 Historical statistics from a team

A soccer match can be studied as a single event with a result that only depends on the positions of the players and the passes between them (explained by Graph Theory), or it can incorporate historical data to contemplate the history of the players in the team and their historical performance. This is the question that triggers the inclusion of historical data to the model in order to add more explanatory variables.

Although the variables given by graph's properties may explain part of the outcome of a match based on the dynamic of the teams involved, it does not consider if any of the teams has historical outstanding goal scorers, for example. Therefore, studying the historical rates of tie, winning and losing of the teams, together with the average number of goals that the teams have can offer more information that can be explanatory for the outcome.

In order to add this historical statistics to the existing dataframe with the graph's properties, a function *history_team()* is created. This function asks the user for three inputs: the tournament, the team and the match id. From these inputs, the function filters the matches of the tournament that were played before the selected match where the selected team is the home team and where it is the away team. It analyzes the outcome of the matches and the goals made by the team and stores the resulting rates in the dictionaries: winning_history, losing_history, ties_history, goals_history, goals_avg_history.

This function is executed for every match in La Liga and the results are stored in a dataframe called *history_df*, which has the following columns: match_id, winning_history_home, winning_history_away, losing_history_home, losing_history_away, tie_history_home, tie_history_away, avg_goals_home, avg_goals_away. The way of calculating these metrics is presented in the table below.

Historic metric of team	**Calculation**
$Winning\ rate\ team\ i\ before\ match\ j$	$\frac{\#\ Winning\ outcomes\ before\ match\ j}{\#\ Total\ matches\ played\ by\ team\ i\ before\ match\ j}$
$Losing\ rate\ team\ i\ before\ match\ j$	$\frac{\#Losing\ outcomes\ before\ match\ j}{\#\ Total\ matches\ played\ by\ team\ i\ before\ match\ j}$
$Tie\ rate\ team\ i\ before\ match\ j$	$\frac{\#Tie\ outcomes\ before\ match\ j}{\#\ Total\ matches\ played\ by\ team\ i\ before\ match\ j}$
$Average\ goals\ team\ i\ before\ match\ j$	$\frac{\#Goals\ made\ by\ team\ i\ before\ match\ j}{\#\ Total\ matches\ played\ by\ team\ i\ before\ match\ j}$

Table 4: Calculation of historic metrics for a specific team.

For the first matches, when the teams do not have any history, the outcomes rates are completed as the default probabilities of the outcomes (33%). For the average goals, the mean value of the column is used to complete the fields of those teams that at the moment of the match do not present any previous records.

2.3.2 Historical statistics from a pair of teams

When studying the historical statistics for a determined team, the interaction between the pair of teams involved in a match is not being contemplated. The teams' historic performance is being analyzed as an independent variable that explains the performance of the team throughout different matches, without taking into account its opponent in the match that is being analyzed.

In order to contemplate the historic interaction between the two teams at stake in a match, it is decided to add to the historical data the metrics that result from analyzing all the matches that were played by these two teams.

For this purpose, a function called *calculate_head_to_head_stats()* is developed. This function requires the inputs: tournament and match id. The function registers the home and away teams involved in the match and looks for their historical encounters prior to the specified match. It loops throughout the matches from the given tournament and finds the matches played by the two given teams. It registers the outcomes of these matches and, finally, it calculates the win, loss and tie rates. The calculations for these metrics are shown in the table below.

Historic metric of pair of teams	**Calculation**
$Pair\ win\ rate\ team\ i\ (teams\ i\ \&\ j,\ match\ k)$	$\frac{\#\ Winning\ outcomes\ team\ i - matches\ against\ team\ j\ prior\ to\ match\ k}{\#\ Total\ matches\ played\ by\ teams\ i\ \&\ j\ before\ match\ k}$
$Pair\ win\ rate\ team\ j\ (teams\ i\ \&\ j,\ match\ k)$	$\frac{\#\ Losing\ outcomes\ team\ i - matches\ against\ team\ j\ prior\ to\ match\ k}{\#\ Total\ matches\ played\ by\ teams\ i\ \&\ j\ before\ match\ k}$
$Pair\ tie\ rate\ (teams\ i\ \&\ j,\ match\ k)$	$\frac{\#Tie\ outcomes - matches\ between\ teams\ i\ \&\ j\ prior\ to\ match\ k}{\#\ Total\ matches\ played\ by\ teams\ i\ \&\ j\ before\ match\ k}$

Table 5: Calculation of historic metrics for a pair of teams.

These metrics are calculated for every match from La Liga and the results are stored in a dataframe named as *pair_df.* This dataframe contains the following columns: match_id, home_team, away_team, pair_win_home, pair_win_away, pair_tie.

As for the historical metrics for a determined team, the default probability of any outcome (33%) is given for those pairs of teams i and j that do not have any prior records to the match k.

2.4 Hypothesis

A soccer match can be represented as a graph, where the nodes stand for the players and the edges represent the passes between them. The study of the resulting properties of the graph can describe the dynamics of the team in the match based on the players' positioning and their interaction through passes.

The dynamics of the teams involved in a match, represented by the explanatory variables given by Graph Theory, have influence on the outcome of the match.

However, analyzing matches as events that are only affected by the positions and passes between players is not contemplating the historical performance of the teams and the historical encounters between these two teams.

For this project, the Machine Learning model implemented will try to predict the outcome of a match based on the properties of the graphs that the teams present and the historical statistics from each team and their interactions.

2.4.1 Definition of variables

The following independent variables are selected as descriptors that will be used to predict the outcome of a match.

Variables from Graph Theory

$Density$: $[density_{home\ team}\ ;\ density_{\ away\ team}]$
$Transitivity$: $[transitivity_{home\ team}\ ;\ transitivity_{\ away\ team}]$
$Diameter$: $[diameter_{home\ team}\ ;\ diameter_{\ away\ team}]$
$Average\ clustering$: $[average\ clustering_{home\ team}\ ;\ average\ clustering_{away\ team}]$

Variables from historical data of a team

$Historical\ winning\ rate$: $[winning\ history_{home\ team};\ winning\ history_{away\ team}]$
$Historical\ losing\ rate$: $[losing\ history_{home\ team};\ losing\ history_{away\ team}]$
$Historical\ tie\ rate$: $[tie\ history_{home\ team};\ tie\ history_{away\ team}]$
$Historic\ average\ goals$: $[goals\ average_{\ home\ team};\ goals\ average_{away\ team}]$

Variables from historical data of the pair of teams

$Historical\ pair\ outcomes$: $[pair\ win_{home\ team};\ pair\ win_{away\ team};\ tie]$

The target will be given by the discrete variable given by the outcome of the match, which can have the following values:

$Outcome\ = \{winner_{home\ team};\ winner_{away\ team};\ tie\}$

2.4.2 Descriptive statistics

A sample is constituted by 868 matches from La Liga. The data frame for this sample contains one column for each of the variables described before. From this data frame, a function called *assign_properties()* iterates through each row in order to divide home and away teams into two groups: winners and losers.

A new data frame with both historical and graph's properties for winners and losers is created, obtaining a sample of a size of 868 for each variable. Ties are not considered for this preliminary study, the properties for those matches that resulted in a tie are equally divided into losers and winners as the average values for the home and away teams' properties.

The distribution of some of the variables are presented below.

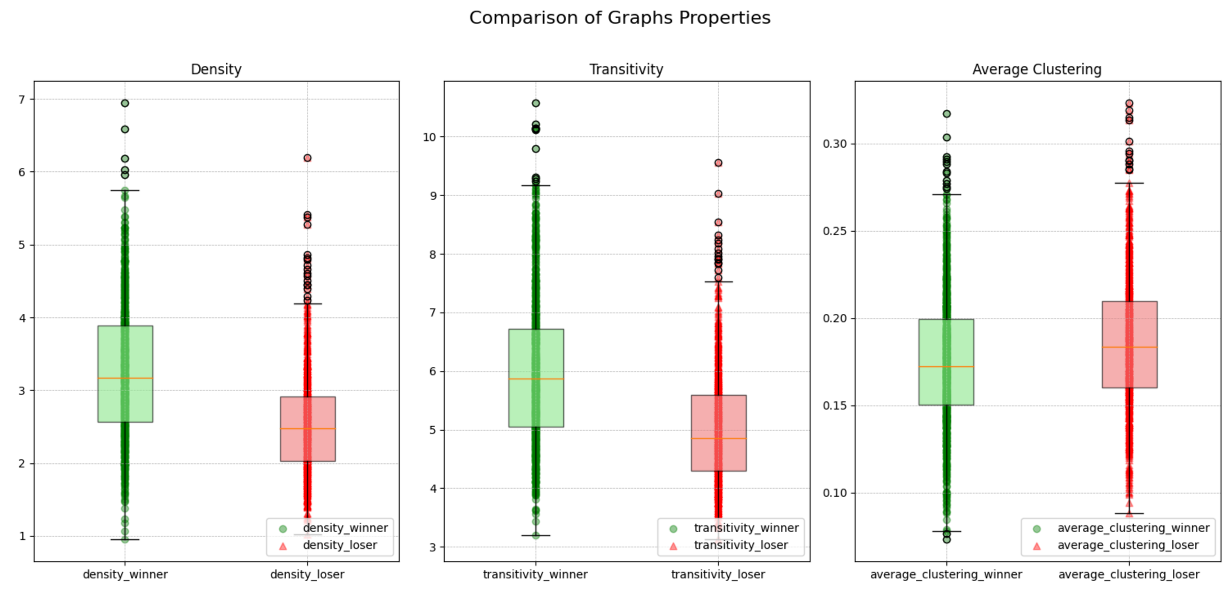


Fig. 7: Boxplots for graph's properties variables for winners and losers.

For the density, it can be seen that the interquartile range for *density_winner* is wider than for *density_loser.* Moreover, the median seems to be higher for *density_winner.* There are several outliers for both distributions. These first observations may indicate that winners tend to have higher density, implying more teamwork and passes.

With regards to the transitivity, the variable *transitivity_winner* has a wider interquartile range than *transitivity_loser*, and its median is also higher. Both distributions present outliers. As well as the density, the transitivity gives an idea of how the team is behaving, and it seems that winners tend to form more triangles than losers.

For the average clustering, both distributions seem to have a similar interquartile range, while *average_clustering_loser* has a higher mean. There are many outliers for both distributions. The average clustering indicates how close players position themselves. The visual analysis obtained from the boxplots might suggest that losers tend to play more closely to one another than winners because they present higher average clustering values.

The comparison of the distribution for the historical variables are presented in the following figure.

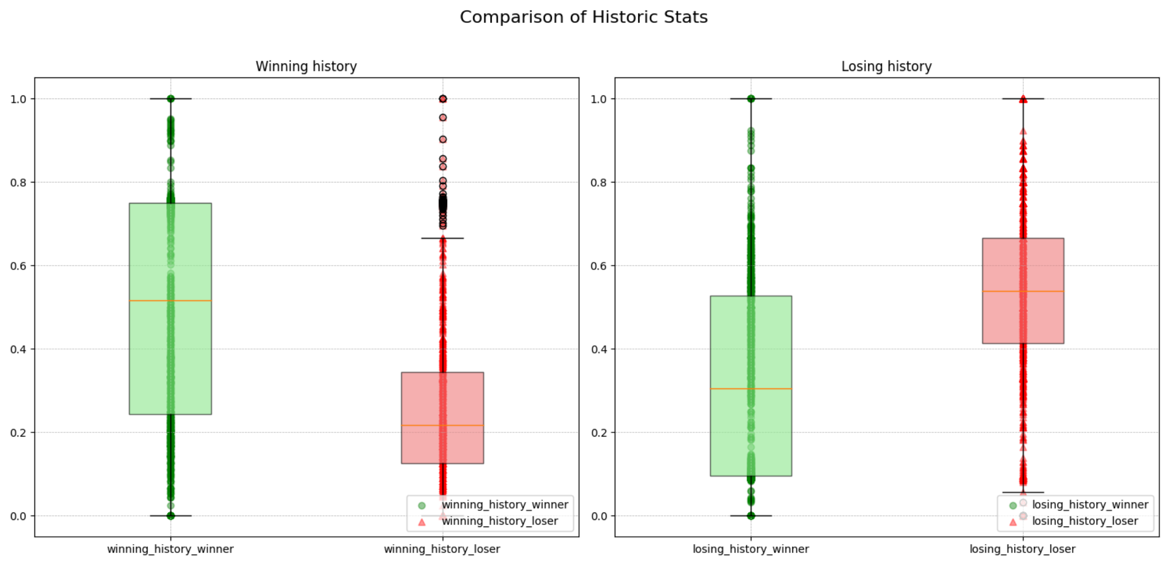


Fig. 8: Boxplots for historical properties variables for winners and losers.

Regarding winning history, the interquartile range for *winning_history_winners* is much wider than the range for *winning_history_loser.* The median for the winners is considerably higher, around 55%, while it is around 23% for the losers. The results from the boxplots may suggest that there is a dependency between the outcome of the match and the historical winning rate of the teams.

Similar observations can be made from the analysis of the boxplots given by *losing_history_winner* and *losing_history_loser.* Losers are more likely to have higher historical losing rates. The median for the winners is close to 30%, while it is around 55% for losers.

2.5 Machine Learning model

In this instance, considering the previous analysis and preliminary results, the hypothesis will be tested by developing a machine learning model to predict the outcome of a match based on the graph's properties and historical metrics. The accuracy of the model will suggest if the studied variables have actual influence in explaining matches' results.

2.5.1 General structure of the model

As already explained, the model will use as descriptors variables corresponding to graphs' properties, variables related to historical rates from the teams involved, and variables associated with the historical outcomes of matches played by these two teams prior to the match.

Considering that the properties of the graph that results from the players' mean positions and passes throughout the match would not be known data before the match, the following approach is considered. Instead of working with the graph's properties that correspond to the match at stake, the model will consider the variables given by the graph's properties of the three most recent matches of each team. In this way, the model will be able to predict the outcome before knowing the actual positioning and passes from each team.

The general structure of the model is presented in the following figure.

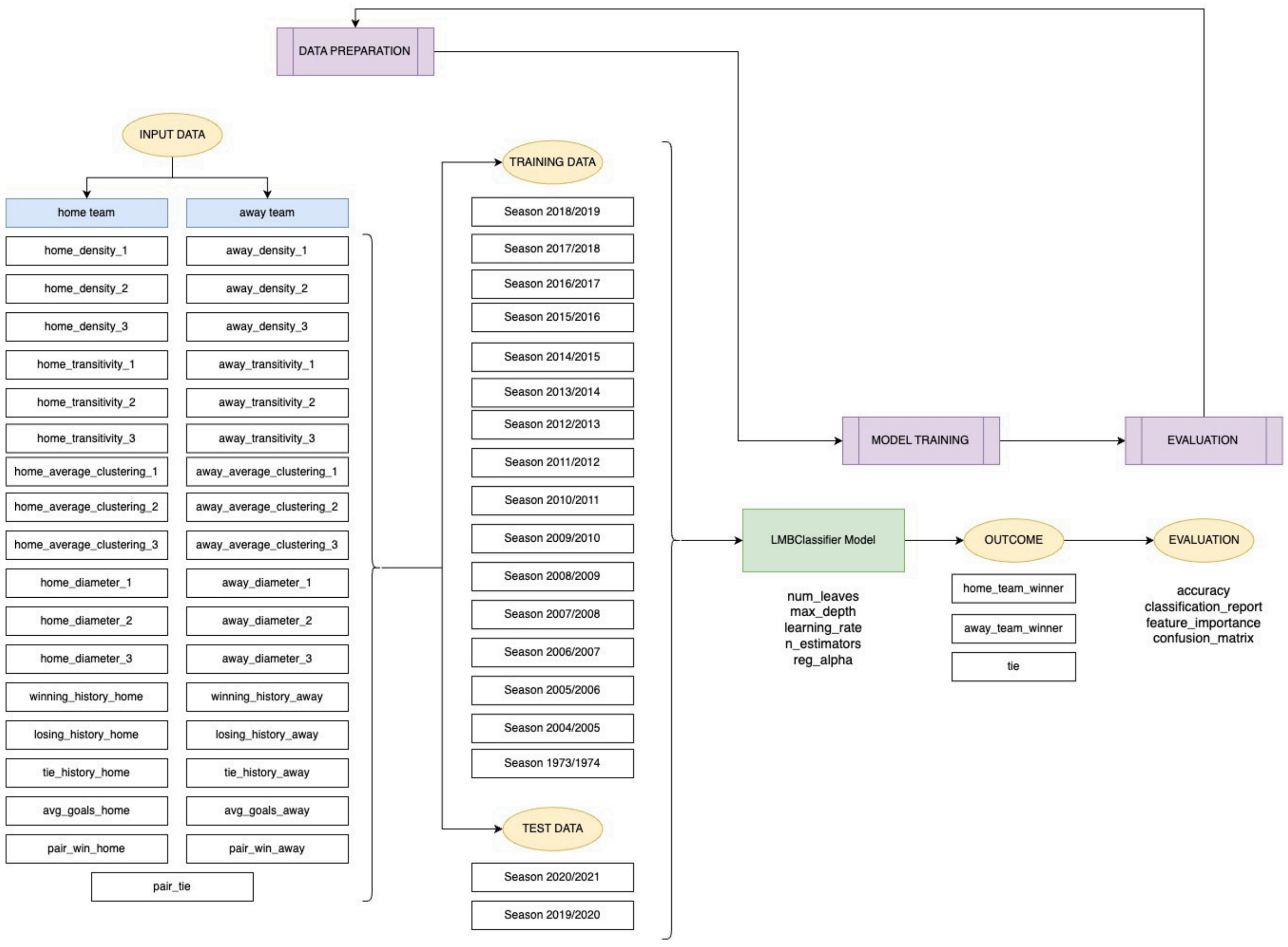


Fig. 9: General structure of the model

2.5.2 Data frame

The data frame used for the model consists of the integration of three data frames: *df_graph_properties, df_history* and *df_pair*. The following schemes show the information that is contained in each of the data frames.

The data frame *df_graph_properties* contains for each match, the graphs' properties of three previous matches for each team.

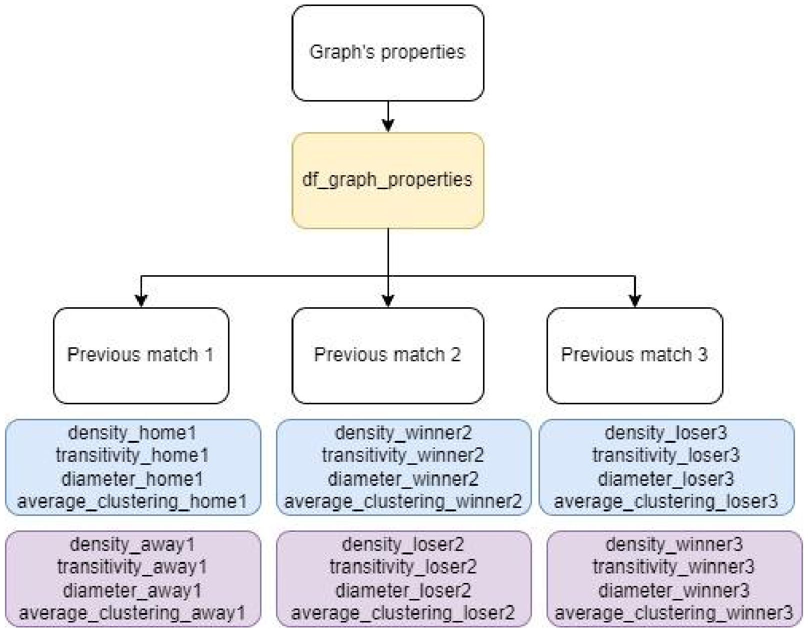


Fig. 10: Columns from df_graph_properties

The data frame *df_history* contains the historical statistics of winning, losing and tie rates for both teams until the moment of the match, with the average scored goals per match as well.

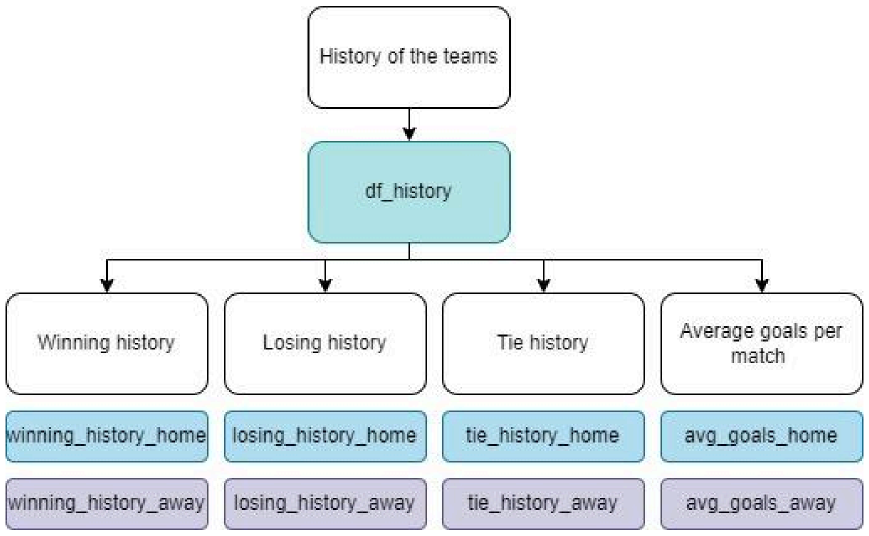


Fig. 11: Columns from df_history

The data frame *df_pair* contains the historical statistics from the encounters of the two teams prior to the match: the winning rate of the home team, winning rate of the away team and tie rate.

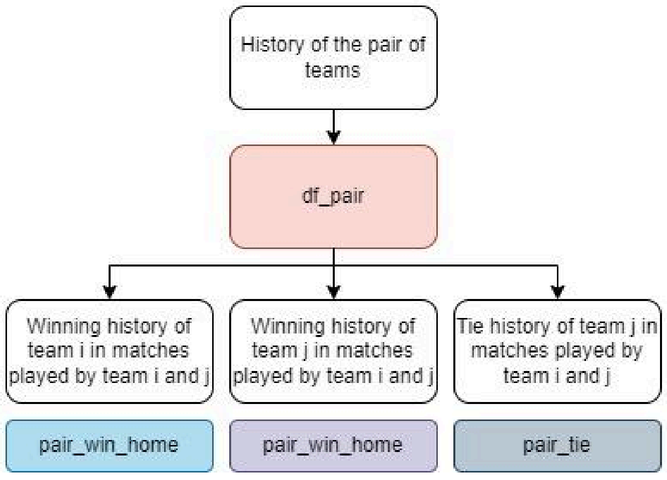

History of the pair of teams
df_pair
Winning history of team i in matches played by team i and j
Winning history of team j in matches played by team i and j
Tie history of team j in matches played by team i and j
pair_win_home
pair_win_home
pair_tie

Fig. 12: Columns from df_pair

These three data frames are merged by using the key ‘match_id’. After a process of data cleaning, the resulting data frame finally contains the descriptors’ variables and the target variables.

2.5.3 Selection of the Machine Learning model

Training and testing datasets

The testing dataset for the machine learning model will correspond to the last two seasons found in StatsBomb of La Liga, 68 matches. This means that the training dataset will be composed of 800 matches from previous seasons.

Selection of the model

The model to implement is selected by using the library ‘lazypredict’ and its module ‘LazyClassifier’. This tool is designed to automate the process of fitting machine learning models to a dataset, obtaining a list of various machine learning models and their results in terms of accuracy, time taken, F1 score, among other performance metrics.

From the report obtained, the model chosen is LGBM Classifier. The results for this model given by ‘LazyClassifer’ are presented in the table below.

Model	Accuracy	Balanced accuracy	F1 Score	Time taken
LGBMClassifier	0.68	0.56	0.61	1.65

Table 6: Performance of LGBM model by ‘LazyClassifier’

Light Gradient Boosting Machine

LGBM Classifier is a gradient boosting framework that uses tree-based learning algorithms. The objective of this classifier is to minimize a loss function that is calculated by resting the differences between the actual and predicted values. For this purpose, the model builds sequential decision trees and each tree is trained to correct the mistakes of the previous ones.

The decision trees that the model uses are binary trees, where each node represents a decision. The final prediction results from the outcomes of the predictions from the individual trees combined. Moreover, LGBM Classifier uses GOSS (Gradient-based-One-Side-Sampling), which means that the model focuses on instances with larger gradients and samples these instances with more frequency.

2.5.4 Execution of the Machine Learning model

From the 'lightgbm' library, the module 'LGBMClassifier' is imported to generate the model. Also, from 'scikit-learn', the 'classification_report' and the 'accuracy_score' are imported to analyze the results of the model.

First, the model is defined as $lgbm = LGBMClassifier()$. Afterwards, the model is fit to the training dataset (X_{train}; y_{train}). Finally, the model is asked to predict the outcome y_{pred} with the given testing inputs stored in X_{test}.

The accuracy of the model results of comparing the predicted values y_{pred} with the actual values y_{test} by using 'accuracy_score- from 'scikit-learn'.

2.5.5 Optimization of the model: hyperparameters

The objective of this instance is optimizing the hyperparameters from LGBM Classifier to obtain the highest accuracy possible and to avoid overfitting. For this purpose, the module 'GridSearchCV' is imported from 'scikit-learn'.

The hyperparameters of the model chosen to optimize the model are: the number of leaves, the maximum depth of a tree, the learning rate or step size, the number of boosting iterations or number of trees and the regularization of L1 penalty. The table below expresses the hyperparameters' variation that is studied, which can be also seen inside the *param_grid* dictionary in the code.

Hyperparameter	**Variable**	**Settings to try**
Number of leaves in a tree	num_leaves	[15, 31, 45, 60]
Maximum depth of the trees	max_depth	[5, 10, 20, -1], where -1 means without limit
Learning rate	learning_rate	[0.01, 0.05, 0.1, 0.15]
Number of boosting iterations or trees	n_estimators	[50, 100, 200]
Regularization of L1	reg_alpha	[0, 0.1, 0.5, 1, 1.5, 2]

Table 7: Hyperparameters' variation

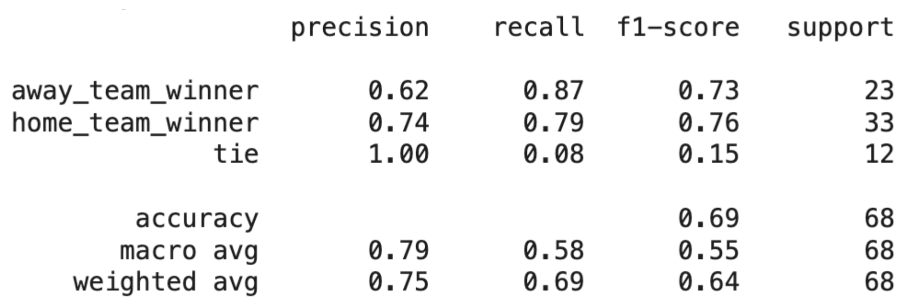
The results obtained from the hyperparameters' variation suggest that the optimal values for the model are the following.

Hyperparameter	**Variable**	**Optimal value**
Number of leaves in a tree	num_leaves	31
Maximum depth of the trees	max_depth	-1
Learning rate	learning_rate	0.1
Number of boosting iterations or trees	n_estimators	100
Regularization of L1	reg_alpha	0

Table 8: Optimal values for the hyperparameters of the LGBM Classifier

2.5.6 Results of the model

The overall accuracy of the model is 69%, which indicates that it correctly predicts the outcome of a game almost 7 out of 10 times.
The model shows to be good at predicting home_team_winner and away_team_winner, with a precision of 0.62 and 0.74 respectively. The recall value is also high, 0.87 and 0.79 respectively, suggesting that the model is better at predicting home_team_wins than away_team_wins. Regarding ties, predictions have a precision of 1, which means that all the predictions made are correct. However, it has a very low recall of 0.08, indicating that the model rarely predicts a tie.
The macro average shows a high precision but lower recall, indicating that it is not as good at identifying all true outcomes. As stated, the model usually does not predict ties.
The weighted average shows that the model's performance is better when analyzing home_team_wins, which is the most common outcome. Thus, the weighted average is higher than the macro average.



```
                  precision    recall  f1-score   support

away_team_winner       0.62      0.87      0.73        23
home_team_winner       0.74      0.79      0.76        33
             tie       1.00      0.08      0.15        12

        accuracy                           0.69        68
       macro avg       0.79      0.58      0.55        68
    weighted avg       0.75      0.69      0.64        68
```

Fig. 13: Classification report

A confusion matrix is developed to visualize the performance of the model on the testing dataset.

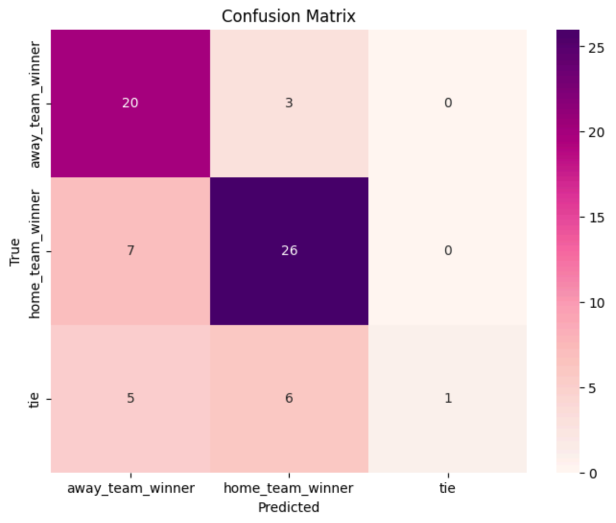


Fig. 14: Confusion matrix

From the confusion matrix, it can be seen that the model fails when predicting ties. This might be due to the size of the observations that resulted in a tie in the training dataset. However, it seems to be very effective to predict winners and losers.

In order to understand the importance of the features studied inside the model, a feature importance visualization is generated.

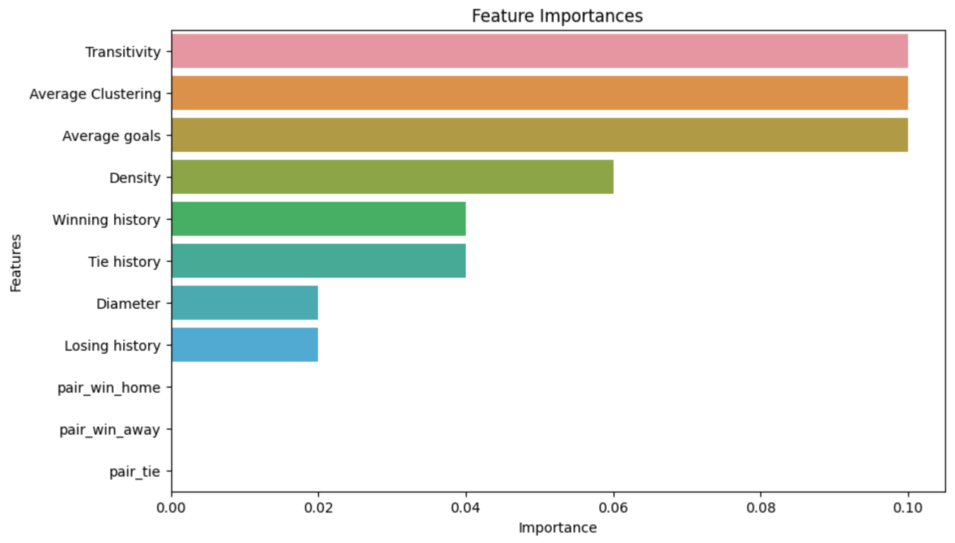
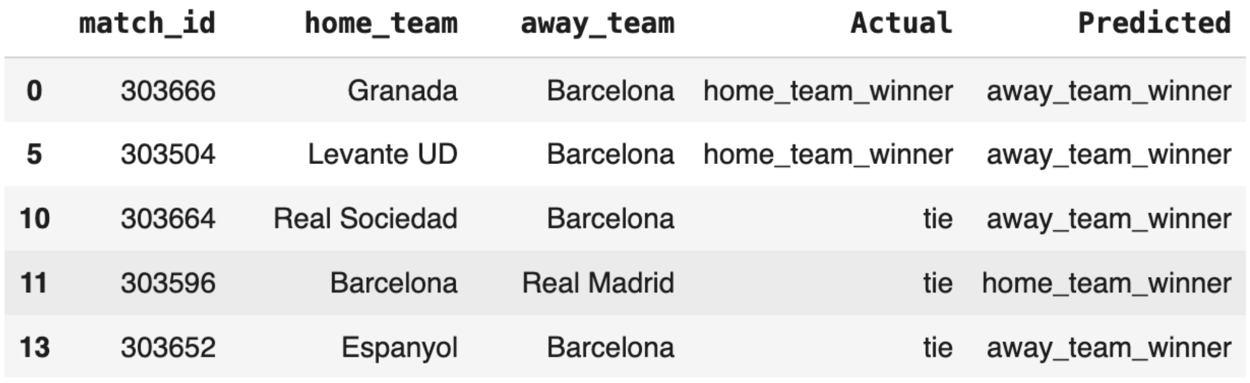


Fig.15: Feature importance

The analysis of the feature importance visualization suggests that the most explanatory features in the model are related to graphs' properties, like average clustering and transitivity. The next most important feature is related with the average goals of each team, followed by the density and then the historical statistics of each team and the diameter. The history between the pair of teams does not seem to have importance on the model.

Concerning the model's predictions, a data frame with the predictions and actual outcomes for each match is created to gain a better understanding of the model. In particular, the data frame is filtered so it shows the wrong predictions. This filtered data frame is called 'not_predicted'.



	match_id	home_team	away_team	Actual	Predicted
0	303666	Granada	Barcelona	home_team_winner	away_team_winner
5	303504	Levante UD	Barcelona	home_team_winner	away_team_winner
10	303664	Real Sociedad	Barcelona	tie	away_team_winner
11	303596	Barcelona	Real Madrid	tie	home_team_winner
13	303652	Espanyol	Barcelona	tie	away_team_winner

Fig. 16: Dataframe not_predicted first results

The majority of the models' mistaken predictions were grounded in reasonable expectations. In Barcelona matches, the model usually predicts victories for Barcelona, which is one of the teams with a more formidable track record. It stands to reason that the model would predict a higher probability of winning for historically stronger teams in the competition.

3. Conclusion

The primary objective of this project is to design a machine learning model to predict the outcome of a soccer match from La Liga based on variables derived from Graph Theory and historical statistics from the teams, as stated in the ‘Abstract’ section. At the project start, the hypothesis held is that graph theory properties such as transitivity, average clustering, density and diameter can influence the match outcome. Our results affirm that hypothesis, demonstrating that these characteristics are of notable significance to the model, as shown in the feature importance graph. Thus, it is evident that the application of graph theory to soccer proves beneficial in predicting match outcomes.

Moreover, the incorporation of historical statistics improves the accuracy of the model, which means that these variables are explanatory for the outcome and therefore are useful to predict results. These observations help to prove that the result of a soccer match can not be analyzed only as a single event that depends on players’ positioning and passes; but it also relies on historical performance of the teams.

However, there are aspects to be improved in the model. According to the results, the model is effective at predicting home and away wins but struggles with predicting ties, likely due to the imbalanced dataset. Only 20% of the matches of the dataset are ties, thus, the model rarely predicts a tie, and most actual ties are missed by the model.

Concerning the model’s predictions, the majority of them were grounded in reasonable expectations. For instance, the model often anticipated victories for Barcelona - one of the strongest teams in La Liga - when the actual outcomes were draws of defeats. Given Barcelona’s formidable track record, it stands to reason that the model would predict a higher probability of winning for such a dominant team.

4. Recommendations for future improvements

To further improve the performance of the predictive model, the first aspect to consider is to gather more data to address the imbalance classes in the model - particularly the model needs more ties data to learn to better predict ties.

Collecting additional features that could influence match outcomes - such as players data, game expectations of the public, possession of the team in previous matches, among others - could also improve performance, by providing more context to make informed predictions.

Additionally, to simplify the model, it would be beneficial to remove non-important features, such as pair history. Removing these features could improve the performance by reducing noise in the data, would be beneficial to training speed and would decrease the chances of overfitting.

Implementing these changes involves challenges, as it requires careful consideration and testing. Each adjustment offers the potential to improve the model performance, but it should be evaluated with validation techniques to ensure that it contributes positively to the model performance.

5. References

[1] Fátima Rodrigues. (2022, September 10). *Prediction of football match results with Machine Learning*. Procedia Computer Science. https://www.sciencedirect.com/science/article/pii/S1877050922007955

[2] Stübinger, J., Mangold, B., & Knoll, J. (2019, December 19). *Machine learning in football betting: Prediction of match results based on player characteristics*. MDPI. https://www.mdpi.com/2076-3417/10/1/46

[3] E. Arriaza-Ardiles, Chen, X., Passos, P., Yamamoto, Y., Cotta, C., Guerra, Y. D. D. S., Fagiolo, G., Greihaine, J. F., & Lago-Peñas, C. (2017, September 21). *Applying graphs and complex networks to football metric interpretation*. Human Movement Science. https://www.sciencedirect.com/science/article/pii/S0167945717306280

[4] Martínez Meneses, M. R. (2013, January 1). Sistema inteligente para pronósticos de partidos de fútbol de la liga española basado en Redes Neuronales Artificiales. Repositorio Académico UPC. https://upc.aws.openrepository.com/handle/10757/314999

[5] Massimo Marchiori, Halvorsen, P., Perin, C., Rehman, A., Janetzko, H., Schlipsing, M., Bialkowski, A., Shafizadeh, M., Clemente, F. M., Castellano, J., Rein, R., Marchiori, M., & Gregson, W. (2019, November 19). *Secrets of soccer: Neural network flows and game performance*. Computers & Electrical Engineering. https://www.sciencedirect.com/science/article/abs/pii/S0045790618332518

[6] General network analysis of national soccer teams in FIFA World Cup ... (n.d.). https://www.tandfonline.com/doi/abs/10.1080/24748668.2015.11868778

[7] Laschober, J., & Harsy, A. (n.d.). *Analysis of passing networks in soccer*. Mathematics and Sports. https://libjournals.unca.edu/OJS/index.php/mas/article/view/7

[8] Applying networks and graph theory to match analysis. (n.d.). https://www.cienciadeporte.com/images/congresos/caceres_2/Rendimiento/aecd2014_submission_16.pdf

[9] View of analysis of passing networks in soccer. (n.d.). https://libjournals.unca.edu/OJS/index.php/mas/article/view/7/3

[10] How to use graph theory to scout soccer. KDnuggets. (n.d.). https://www.kdnuggets.com/2022/11/graph-theory-scout-soccer.html

[11] Lagos. (2020, April 15). El Uso de Grafos en el Fútbol. Medium.

https://medium.com/@jesslm_48641/el-uso-de-grafos-en-el-f%C3%BAtbol-47637b7089d2

[12] Lagos, J. (2019, June 10). Uso de Redes (grafos) en el fútbol. Medium. https://medium.com/@jesslm_48641/uso-de-redes-grafos-en-el-f%C3%BAtbol-655d2dfd8cb1

[13] Manejo de Grafos Con NetworkX en Python. https://www.ellaberintodefalken.com/2020/02/grafos-con-networkx.html

[14] Prashant. (2020, July 21). *LightGBM classifier in Python*. Kaggle. https://www.kaggle.com/code/prashant111/lightgbm-classifier-in-python