Soccer Analytics - Data Science applied to soccer

Gabriel Bortoli[1], **Lucas Maretti**[2], **and Roseli A.P. Romero**[3]

[1]**Universidade de São Paulo - Instituto de Ciencias Matemáticas e Computação**

ABSTRACT

The accurate prediction of expected goals (xG) in soccer matches has gained significant attention in recent years due to its potential applications in various aspects of the game. In this study, we investigate the performance of different classification models, namely Logistic Regression, Random Forest, XGBoost, LightGBM, and LSTM; for developing an xG model using the open data provided by the renowned company Wyscout.
The dataset used in this research contains comprehensive information about various match events, player actions, and contextual features. We preprocess the data and engineer relevant features to enhance the model's predictive capabilities. We then compare the performance of the aforementioned models by evaluating their log-loss scores.
Our experiments reveal that the XGBoost model outperforms other classification models, achieving a log-loss of 0.2793. The XGBoost model demonstrates superior predictive accuracy in estimating the likelihood of a goal-scoring opportunity based on the available match data. The robustness and flexibility of the XGBoost algorithm make it an excellent choice for modeling expected goals in soccer matches.
These findings highlight the potential of machine learning techniques in soccer analytics and provide valuable insights for both researchers and practitioners in the field. The developed xG model can be utilized in various applications, such as player evaluation, team performance analysis, and strategic decision-making during matches.

Keywords: football analytics, probabilistic classification, soccer match data

1 INTRODUCTION

According to Nielsen, a leader in audience analysis, football is the sport with the largest number of fans worldwide. In their *Annual World Report* for 2022 (Nielsen (2022)), which includes data analysis on sports viewership, it is estimated that 40% of the global population considers football as their favorite sport, making it one of the most significant sports in economic terms. In an article also published in 2022, Forbes magazine (Forbes (2022)) listed the Spanish club Real Madrid as the most valuable in the world, with a market value of 5.1 billion euros, closely followed by clubs like Barcelona, valued at 5.1 billion euros, and Liverpool, valued at 4.45 billion euros. The Brazilian scenario is analogous: Flamengo and Palmeiras lead the market evaluation according to a study conducted by the consultancy Sport Value (Exame (2022)), with values of 3.4 billion and 3.1 billion reais, respectively. Furthermore, in Brazil, football transcends mere entertainment and holds significant socio-economic aspects: many boys and girls from low-income backgrounds find in the sport an opportunity for social mobility. Moreover, the success of hosting the 2014 FIFA World Cup and the 2016 Olympics demonstrates that the combination of tourism and sports is an important driving force for the country's economic growth.

The boom of data science in the past 10 years, which has revolutionized sectors of the economy such as e-commerce, retail, and manufacturing, driven by increased computational power and the broader collection of data through IoT devices, has also reached the realm of soccer. Currently, players wear sensor devices that collect data on their movements during matches (also known as tracking data), allowing for the mapping of player coordinates on the soccer field with a granularity of seconds. Furthermore, specialized companies like Wyscout, Statsbomb, and Opta perform event tagging during matches, meticulously recording actions that occur on the field, such as passes, shots, tackles, interceptions, among others, generating event data.

The combination of tracking and event data has led to a true revolution in the way soccer is analyzed through the lens of data science in the past five years. Few European clubs today operate without an Analytics department within their organizational structures. In Brazil, however, this is still an emerging scenario. Among the top clubs in the Brazilian Serie A championship, only Atlético-MG and Red Bull Bragantino have dedicated data analysis structures, with Atlético-MG being the pioneer, starting as recently as 2021, according to the newspaper O Tempo (Tempo (2021)). Therefore, this article aims to contribute to the literature on this topic in the national context.

Since data science started being used to analyze soccer matches, new metrics and concepts have emerged to try to explain the nature of the game, which is inherently random and subject to uncertainties. The main metric among these is the so-called expected goals (xG). Expected goals assign a probability between 0 and 1 to each shot taken by a player in a game (0 indicating no possibility of the shot being a goal and 1 indicating certainty of a goal). This is a better way to deal with the randomness in soccer compared to traditional metrics based solely on the number of goals, as shots are much more frequent occurrences than actual goals. In the words of David Sumpter, a renowned mathematician and author of the book ”Soccermatics,” ”xG is the probability that on a typical day of soccer, a particular shot from a specific location would result in a goal. It is often based on measurements taken from many shots within the same league and season or by aggregating data from different leagues.” (Sumpter (2017))

In the book ”Football Hackers,” author Christoph Biermann tells the story of Ted Knutson, one of the pioneers in using expected goals. Knutson initially worked as a data analyst for a company specializing in sports betting and later founded the company Statsbomb, which has become one of the market leaders in providing data collection and analysis solutions for soccer clubs. Knutson also served as a sports consultant for the English team Brentford.

Brentford is often described as a ”Moneyball” club, referring to the 2011 movie starring Brad Pitt and Jonah Hill that depicts the Oakland Athletics baseball team, which became a pioneer in using data analysis for assembling their roster - a groundbreaking strategy at the time to overcome the club’s revenue limitations (Biermann (2019)).

The use of analytics in decision-making marked a new era for the small London-based team. In the 2013-14 season, the club was competing in the third division of the English league (League One) and had not played in the top division since 1947, nor had they ever won any FA Cup or League Cup titles. Currently, the club is in its second consecutive season in the Premier League, the top division of English soccer, sitting in ninth position in the league. They have achieved remarkable results in the current 2023-2024 season, including victories over Liverpool, Chelsea, and Manchester United.

The data required to build any xG model in soccer (event or tracking data with positional information) is difficult to obtain, as the companies that collect such data typically use it to build their own models. Fortunately, as part of the Soccer Data Challenge initiative (a soccer data analytics event held in Italy in 2019), the oraganization provided what is believed to be the largest publicly available collection of event data ever released (Pappalardo et al. (2019)). The data, which was collected by the company Wyscout, encompasses all events from the top five European leagues (English Premier League, Spanish La Liga, German Bundesliga, Italian Serie A, and French Ligue 1) for the 2017-18 season.

Using this Wyscout data, the objective of this study was to train various classifiers: Logistic Regression, Random Forest, LightGBM, XGBoost, and LSTM—on the data from the five leagues for all available matches for the expected goals (xG) metric. Two main metrics were used to evaluate the models: logarithmic loss (log loss) and the area under the curve (AUC).

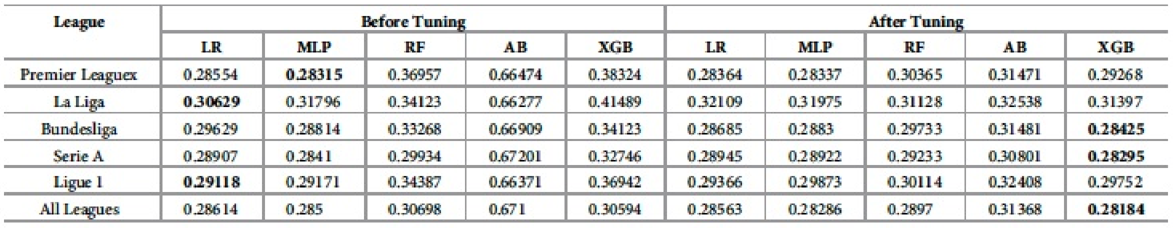
In the following sections, the organization of the data provided by Wyscout will be presented in more detail, as well as the chosen models and metrics for modeling and how the data was processed. This will be followed by a description of the exploratory data analysis and hypothesis generation. Finally, the final modeling results, including training and comparison of the models, will be discussed, leading to conclusions.

2 RELATED WORK

The article by Pappalardo et al. (2019) is crucial to mention as it introduced the database that is now widely used for research in football analytics. In this article, the database is described in detail, providing examples of how a well-performed exploratory analysis can already bring relevant insights to clubs.

On the other hand, Mead et al. (2023) and Umami et al. (2021) specifically worked on modeling expected goals. They tested models such as Logistic Regression, Random Forest, Multilayer Perceptron, Adaboost, and XGBoost, using logarithmic loss and AUC as the main metrics. The following table shows a summary of the results obtained by the authors:

Figure 1. Log-loss results for tested models

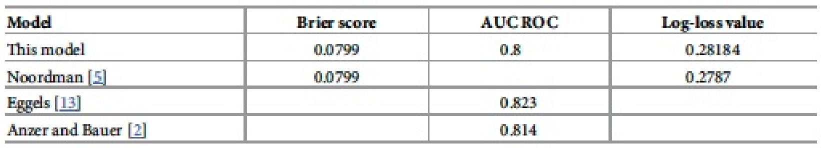


League	Before Tuning					After Tuning				
	LR	MLP	RF	AB	XGB	LR	MLP	RF	AB	XGB
Premier Leaguex	0.28554	**0.28315**	0.36957	0.66474	0.38324	0.28364	0.28337	0.30365	0.31471	0.29268
La Liga	**0.30629**	0.31796	0.34123	0.66277	0.41489	0.32109	0.31975	0.31128	0.32538	0.31397
Bundesliga	0.29629	0.28814	0.33268	0.66909	0.34123	0.28685	0.2883	0.29733	0.31481	**0.28425**
Serie A	0.28907	0.2841	0.29934	0.67201	0.32746	0.28945	0.28922	0.29233	0.30801	**0.28295**
Ligue 1	**0.29118**	0.29171	0.34387	0.66371	0.36942	0.29366	0.29873	0.30114	0.32408	0.29752
All Leagues	0.28614	0.285	0.30698	0.671	0.30594	0.28563	0.28286	0.2897	0.31368	**0.28184**

Source: Mead et al. (2023)

In the same article, the authors provide a comparison with results obtained from other publications on the same topic. The results are summarized and presented in Table 2 below:

Figure 2. Log-loss results and AUC comparing to other publications on the same topic



Model	Brier score	AUC ROC	Log-loss value
This model	0.0799	0.8	0.28184
Noordman [5]	0.0799		0.2787
Eggels [13]		0.823	
Anzer and Bauer [2]		0.814	

Source: Mead et al. (2023)

Umami et al. (2021) focused on the use of Logistic Regression with the AUC metric, obtaining similar results to those mentioned earlier.

3 DATA AND METHODS

As described in the introductory section, the data used in this study was provided by Wyscout as part of the Soccer Data Challenge initiative in 2019. This dataset includes all events from the top five European leagues (English Premier League, Spanish La Liga, German Bundesliga, Italian Serie A, and French Ligue 1) for the 2017-18 season. This dataset is particularly valuable as it has become a reference in academic studies on the topic.

Although positional data is not available (thus, some influential features examined in other works cannot be included in the models), this dataset was the most comprehensive and publicly available source found, containing the necessary information to fulfill the objectives of this study.

3.1 About the dataset

The Wyscout data is available in JSON format and is divided into three categories: ”matches”, which contains metadata about the matches such as location, league, stadium name, match result, and more; ”events”, which includes the events (actions that occur in the game, as mentioned before) for each match (the largest file of all); and ”players”, which contains metadata about the players, such as full name, age, preferred foot for kicking, and more.

Next, we will describe the features present in the events file, which is crucial for the modeling process (details about the other files can be found in the Appendix):

Events data (events.json):

- eventId: identifier of the event type. Each eventId is associated with an event name (see the next point);
- eventName: name of the event type. There are seven types of events: pass, foul, shot, duel, free kick, offside, and touch;
- subEventId: identifier of the subevent type. Each subEventId is associated with a subevent name (see the

next point);
- subEventName: name of the subevent type. Each event type is associated with a different set of subevent types;
- tags: list of event tags, each describing additional information about the event (e.g., for a pass event, whether the pass was accurate). Each event type is associated with a different set of tags;
- eventSec: time at which the event occurs (in seconds since the start of the current half of the match);
- id: unique identifier for the event;
- matchId: identifier of the match to which the event refers. The identifier corresponds to the ”wyId” field in the match dataset;
- matchPeriod: period of the match. It can be ”1H” (first half), ”2H” (second half), ”E1” (first extra time), ”E2” (second extra time), or ”P” (penalties);
- playerId: identifier of the player who generated the event. The identifier corresponds to the ”wyId” field in the player dataset;
- positions: origin and destination positions associated with the event. Each position is a pair of coordinates (x, y). The x and y coordinates are always in the range [0, 100] and indicate the percentage of the field in the perspective of the attacking team. In particular, the x-coordinate value indicates the proximity of the event (as a percentage) to the opposing goal, while the y-coordinate value indicates the proximity of the event (as a percentage) to the right side of the field;
- teamId: identifier of the player’s team. The identifier corresponds to the ”wyId” field in the team dataset.

3.2 Tested Models

Obtaining an xG model is a classification problem in machine learning, as we aim to predict the probability of a shot resulting in a goal. With that in mind, five algorithms were tested in this study: Logistic Regression, Random Forest Classifier, LightGBM, XGBoost, and LSTM. The implementation of these models was done using the scikit-learn library.

For Logistic Regression, the main hyperparameters evaluated were:

- C: Inverse of regularization strength; must be a positive number. Smaller values specify stronger regularization. The following values were tested: [0.3,0.5,0.8,1.0]
- Solver type: The algorithm to be used for optimization. In this problem tests were done with two solvers: ’lbfgs’, which uses L2 regularization, and ’newton-cholesky’, which is recommended when there are multiple features with one-hot encoding, as is the case in this project.

For Random Forest, the hyperparameters evaluated were:

- Number of estimators: Equivalent to the number of trees to be trained. The tested values were [30, 40, 60], as higher values tended to result in overfitting;
- Max depth: The maximum depth of the tree. If set to None, nodes will be expanded until all leaves are pure. The tested values were [4, 5, 6, 7, 8, 9, 10];
- Min sample split: The minimum number of samples required to split an internal node. The tested values were [2, 4, 6, 8].

For LightGBM, the hyperparameters evaluated were:

- Number of estimators: Equivalent to the number of trees to be trained. The tested values were [20, 50, 100, 200, 300];
- Max depth: The maximum depth of the tree. If set to None, nodes will be expanded until all leaves are pure. The tested values were [4, 5, 6, 7, 8, 9, 10];
- Number of leaves: Represents the maximum number of leaf nodes in a decision tree. In other words, it determines the complexity or maximum size of the tree. This hyperparameter directly affects the model’s ability to fit the training data. By increasing the value of ”num leaves,” the tree becomes deeper and more complex, allowing the model to better fit the training data. The tested values were [14, 31, 63, 127].

For XGBoost model, the hyperparameters evaluated were:

- Number of estimators: Equivalent to the number of trees to be trained. The tested values were [20, 50, 100, 200, 300];
- Max depth: The maximum depth of the tree. If set to None, nodes will be expanded until all leaves are pure. The tested values were [4, 5, 6, 7, 8, 9, 10];
- Col sample by tree: Controls the fraction of features (variables) to be considered in each tree during XGBoost training. It determines the proportion of columns (features) that are randomly sampled in each tree. The tested values were [2, 4, 6, 8];
- Learning rate: Controls the learning rate of the model. It is a factor that multiplies the gradient updates during the training process. A higher learning rate allows the model to adjust more quickly to the data but can also lead to a higher risk of overfitting. The tested values were [0.05, 0.1, 0.3].

For LSTM, the different network architectures studied were::

- Network 1:
- model = Sequential()
- model.add(LSTM(units=2))
- model.add(Dense(units=1))
- model.add(Dense(units=1, activation='sigmoid'))
- model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['binary_crossentropy'])
- epochs = 10 ; batch_size = 32

- Network 2:
- model = Sequential()
- model.add(LSTM(units=4))
- model.add(Dense(units=1))
- model.add(Dense(units=1, activation='sigmoid'))
- model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['binary_crossentropy'])
- epochs = 10 ; batch_size = 32

- Network 3:
- model = Sequential()
- model.add(LSTM(units=8))
- model.add(Dense(units=2))
- model.add(Dense(units=1, activation='sigmoid'))
- model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['binary_crossentropy'])
- epochs = 10 ; batch_size = 32

- Network 4:
- model = Sequential()
- model.add(LSTM(units=12))
- model.add(Dropout(0.2))
- model.add(Dense(units=4))
- model.add(Dense(units=1, activation='sigmoid'))
- model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['binary_crossentropy'])
- epochs = 10 ; batch_size = 32

In addition, a data split of 75% for training and 25% for testing was chosen. As goals are relatively rare (at a ratio of approximately 1 goal for every 10 shots, considering all leagues), the split was done in a way that maintained equivalent goal proportions in both datasets. Scaling of the model attributes was also considered to avoid assigning larger weights to variables with larger values and vice versa. This scaling technique is known as "min-max scaling," and it is performed using the formula:

$$x_{\text{scaled}} = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

where x is the original value of the feature, $x_{\min}$ is the minimum value of the feature in the dataset, $x_{\max}$ is the maximum value of the feature in the dataset and x_{scaled} is the scaled value of the feature ranging from 0 to 1.

Cross-validation was also used during the model training, employing 5 folds in all training processes, including the hyperparameter optimization process using the grid search technique.

3.3 Metrics

For classification algorithms, the standard cost function used is the logarithmic loss. For models with binary output, such as the expected goals model, the log loss function is given by:

$$\mathscr{L}(\hat{y}, y) = -\frac{1}{n}\sum_{i=1}^{n} y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)$$

where $\hat{y}$ is the predicted probability of the positive class, y is the true value (0 or 1) and n is the number of samples in the dataset.

While most evaluation metrics for classification problems involve analyzing the predictive ability of the model in various situations using traditional metrics such as precision and recall, this approach is not ideal for the xG modeling problem. This is because the desired outcome of the model is the probability that a specific shot will result in a goal, rather than a prediction of whether a shot is a goal (i.e., a binary output). For this reason, this study uses the logarithmic loss as the main comparative metric for models, where a lower score indicates a better ability of the classification algorithm to accurately estimate the probability of a goal.

Additionally, due to its mention in some researched works during the literature review, the AUC (area under the curve) was also evaluated as a secondary metric. AUC represents the area under the ROC curve.

4 RESULTS AND DISCUSSION

4.1 Exploratory Analysis

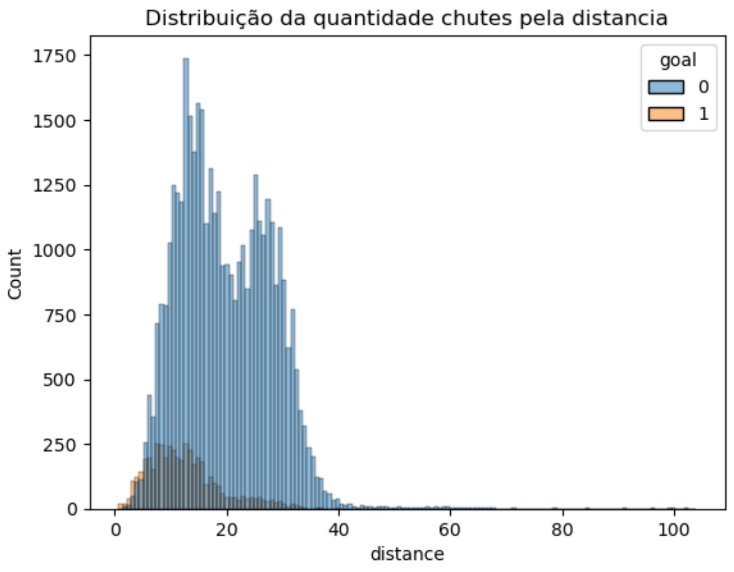
After reading the three JSON files, the first step was to preprocess the data, which involved merging them into a tabular format (Pandas DataFrame).

With a unified dataset and based on the literature review, several features that could be relevant for creating an expected goals (xG) model were identified, such as the angle and distance of the shot to the goal, calculated from the "positions" attribute, which provides the x and y coordinates of each shot. Additionally, binary variables were created for pre-shot events, such as the type of pass, whether it was a through ball, a tackle, or at the moment of the shot. For example, whether the shot originated from a free-kick or a corner kick, among others. At this step, several variables that were known to have no impact on the model, such as some match metadata (stadium name, country) and player metadata (date of birth, surname), were excluded. Furthermore, it was found that there were no missing data during this initial stage.

In the end, the dataset used for exploratory analysis and modeling contained 45,284 entries (rows) and 34 features (columns).

The initial stage of exploratory analysis involved evaluating the behavior of features with respect to the target variable, which is the ”goal” column that takes values of 0 (no goal) and 1 (goal). An example of analysis is shown below, which demonstrates the distribution of the number of shots with respect to the distance. It can be observed that shots resulting in goals tend to be taken closer to the opposing goal.

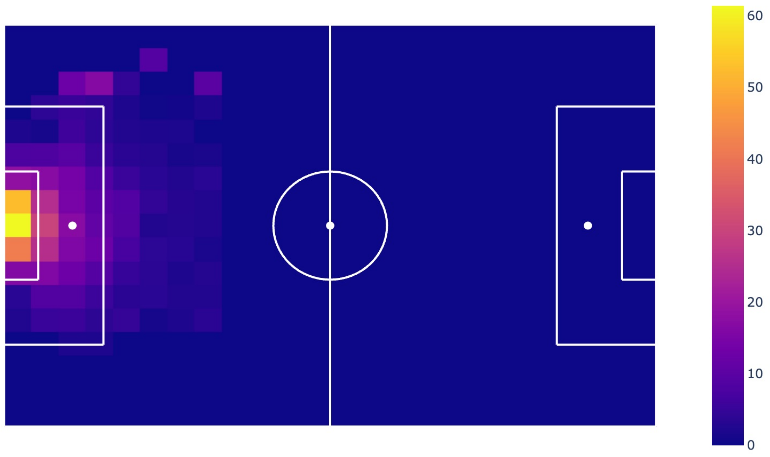
Figure 3. Distribution of the number of goals with respect to the distance of the shot



Source: from author.

Additionally, it is also possible to calculate the probability of scoring a goal based on distance using a heatmap graph. To do this, we divide the field into subsections and for each subsection, we divide the number of goals scored in that region by the number of shots taken to obtain the probability. The resulting graph can be observed in Figure 4, shown below.

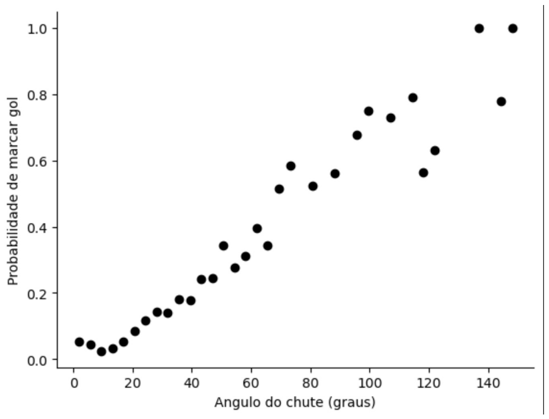
Figure 4. Probability of scoring based on position on the field



Source: from author.

From the initial graphical analyses, two main hypotheses were raised: What is the relationship between the distance and angle of the shot and the probability of scoring a goal? To answer these questions, the graphs shown in Figures 5 and 6 were plotted.

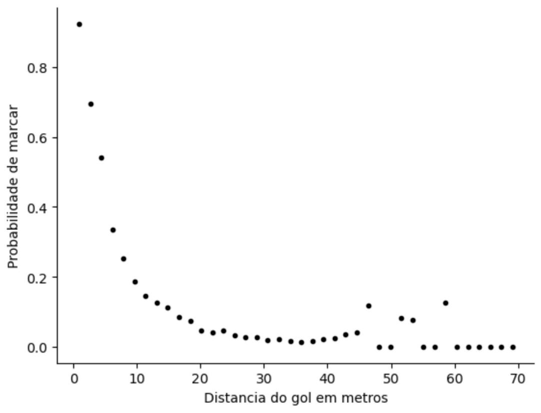
Figure 5. Probability of scoring a goal based on the angle of the shot



Source: from author.

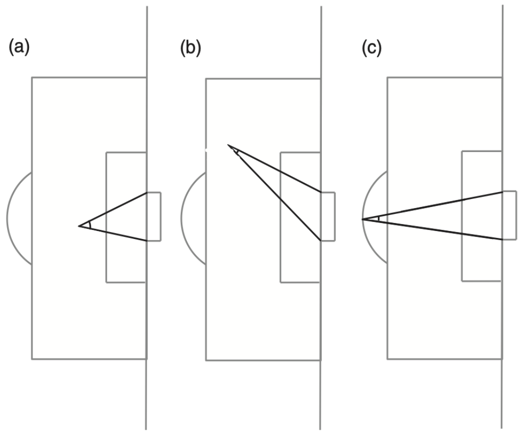
These results indicated that the angle of the shot and the distance from the opposing goal could be features with good predictive power for the desired models. Regarding the angle, the relationship is direct: larger angles favor the probability of scoring a goal. In contrast, the relationship with distance is inverse: the closer to the goal, the better the probability. Figure 7, taken from Sumpter (2017), helps illustrate why this is the case: the angle is calculated based on the triangle formed by the ball and the two goalposts, so larger angles mean that the player has a wider view of the opposing goal. Proximity to the opposing goal also aids in this regard, as seen when comparing diagrams a) and c) in Figure 7.

Figure 6. Probability of scoring a goal based on the distance of the shot



Source: from author.

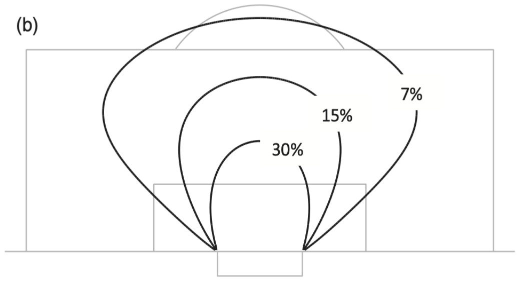
Figure 7. Relationship between angle and distance of the shot



Source: Sumpter (2017)

This type of analysis helps explain another advantage of expected goals: assisting players in decision-making. With the previous analyses and having an xG model, it is possible to create a diagram like Figure 8, which presents the probabilities of scoring within the opponent's penalty area in a contour-style graph. By showing such a graph to a player, it is possible to guide them on where they should aim their shots from within the penalty area to maximize their chances of scoring a goal.

Figure 8. Probabilities of scoring within the opponent's penalty area



Source: Sumpter (2017)

4.2 Modeling

As mentioned in the Data and Methods section, the data was splitted into training and test sets with a ratio of 75% and 25%, respectively. Feature scaling was performed using minimum and maximum values. Cross-validation was applied during the model training using 5 folds for all the experiments, including hyperparameter optimization using grid search.

As a baseline reference for the logarithmic loss metric, a model using only the average expected goals value (10.17%) was used, resulting in a log-loss of 0.333. Next, a Logistic Regression model was trained using all the features with hyperparameter optimization, achieving a log-loss of 0.281. A Random Forest Classifier was also tested, yielding the best result with a log-loss of 0.280. The LGBM model produced a similar result to the Random Forest, followed by XGBoost with a slightly better log-loss of 0.279. Lastly, the LSTM model had a slightly higher log-loss of 0.283. A summary of the results, including the AUC values along with the logarithmic loss, can be found in the following table:

Table 1. Summary of modeling results (RL = Logistic Regression; LGBM = LightGBM, XGB = XGBoost)

Metric/Model	Baseline	RL2	Random Forest	LGBM	XGB	LSTM
Log-loss	0.333	0.281	0.280	0.280	0.279	0.283
AUC	-	76.59%	78.21%	78.39%	78.32%	-

Therefore, the XGBoost algorithm provided the best performance with a log-loss of 0.279. The results obtained closely match those from the literature review, especially the findings reported byMead et al. (2023), whose best XGBoost model achieved a log-loss of 0.28184. For Logistic Regression, the result was nearly identical, as the authors reported a log-loss of 0.285.

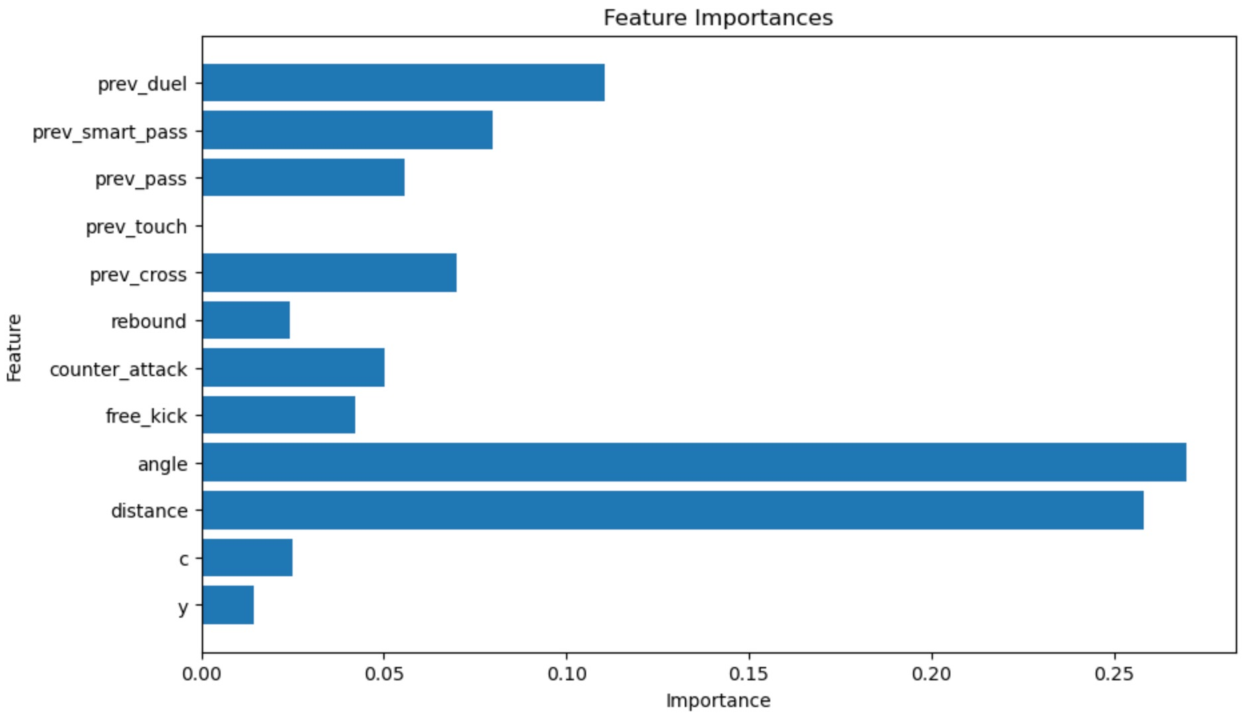
4.3 Features' Importance

For the best model (XGBoost), the features that had the most impact on the model's outcome were evaluated, and the results corroborated what was observed in the exploratory analysis: angle and distance to the goal are the features that have the greatest impact on the final model outcome. To obtain this information, the "feature_impotances_" attribute of the trained model was used. This method provides a score for each feature, indicating how much each feature contributed to improving the model's performance during the training process. This score is calculated based on the statistics of information gain or precision gain obtained when splitting the nodes of the tree during model construction.

The calculation of feature importance in XGBoost is based on decision trees. Each time a feature is used to split a node in a tree, the model calculates the resulting information gain or precision gain from that split. Information gain measures the reduction in entropy after the split, while precision gain represents the improvement in classification accuracy.

XGBoost calculates the feature importance by summing up the information gains or precision gains for each feature across all the trees in the model. It then normalizes these scores relative to the total sum, so that the importance of all features adds up to 1 or 100%. Thus, the resulting score represents the proportion of contribution of each feature to the overall performance of the model.

Figure 9. Most important features for the XGBoost model



Source: from author.

5 CONCLUSION

This project aimed to contribute to the limited set of research on expected goals in the Portuguese language (original language in which the article was written) and compare the performance of the models with previous studies in other countries, using the data provided by Wyscout, which has become a reference for this type of analysis. The optimal model developed in this project (XGBoost) demonstrated competitiveness in relation to the results of existing research in the literature. The most important variables were identified as the angle and distance of the shot, revealing interesting insights into how set-piece plays and counterattacks can increase the probability of a shot resulting in a goal.

Despite the interesting results obtained in this study, it is important to highlight its limitations, which can be addressed in future research. One of the main limitations is related to feature engineering. There are several variables that can be created and calculated to complement the dataset, in addition to those already used. For example, the player's value according to the TransferMarket website could be considered.

Furthermore, the modeling did not include positional data due to the unavailability of this information in the dataset used. Otherwise, it would be possible to create additional features, such as the goalkeeper's position at the time of the shot, and evaluate the presence of defenders in the shot trajectory, further enriching the analysis. Despite these limitations, the results of this study confirm that expected goals can bring significant value to analysts, coaching staff, and fans by providing a new perspective on the game, which is subject to uncertainties and randomness. The main benefits of this approach are the predictive ability of team performance, which goes beyond the simple number of goals scored, mainly due to the fact that the expected goals model is based on shots, which occur much more frequently than goals during matches. Additionally, a good expected goals model can assist players in their decision-making process regarding when to take a shot, by informing them about the areas of the field where the probability of scoring is higher, as illustrated in Figures 7 and 8.

These results highlight the importance of the expected goals model as a valuable tool for football analysis, offering a more comprehensive and informative perspective on goal-scoring probabilities. We hope that this study inspires future research to further improve expected goals models by exploring new variables and analysis techniques, and also encourages the practical application of these models in real scenarios, benefiting teams, players, and football enthusiasts.

6 APPENDIX

6.1 Description of matches and players data

Match data (matches.json)

- competitionId: the identifier of the competition to which the match belongs to. It is a integer and refers to the field ”wyId” of the competition document;
- date and dateutc: the former specifies date and time when the match starts in explicit format (e.g., May 20, 2018 at 8:45:00 PM GMT+2), the latter contains the same information but in the compact format YYYY-MM-DD hh:mm:ss;
- duration: the duration of the match. It can be ”Regular” (matches of regular duration of 90 minutes + stoppage time), ”ExtraTime” (matches with supplementary times, as it may happen for matches in continental or international competitions), or ”Penalities” (matches which end at penalty kicks, as it may happen for continental or international competitions);
- gameweek: the week of the league, starting from the beginning of the league;
- label: contains the name of the two clubs and the result of the match (e.g., ”Lazio - Internazionale, 2 - 3”);
- roundID: indicates the match-day of the competition to which the match belongs to. During a competition for soccer clubs, each of the participating clubs plays against each of the other clubs twice, once at home and once away. The matches are organized in match-days: all the matches in match-day i are played before the matches in match-day i + 1, even tough some matches can be anticipated or postponed to facilitate players and clubs participating in Continental or Intercontinental competitions. During a competition for national teams, the ”roundID” indicates the stage of the competition (eliminatory round, round of 16, quarter finals, semifinals, final);
- seasonId: indicates the season of the match;
- status: it can be ”Played” (the match has officially finished), ”Cancelled” (the match has been canceled for some reason), ”Postponed” (the match has been postponed and no new date and time is available yet) or ”Suspended” (the match has been suspended and no new date and time is available yet);
- venue: the stadium where the match was held (e.g., ”Stadio Olimpico”);
- winner: the identifier of the team which won the game, or 0 if the match ended with a draw;
- wyId: the identifier of the match, assigned by Wyscout;
- teamsData: it contains several subfields describing information about each team that is playing that match: such as lineup, bench composition, list of substitutions, coach and scores: - hasFormation: it has value 0 if no formation (lineups and benches) is present, and 1 otherwise;
- score: the number of
goals scored by the team during the match (not counting penalties); - scoreET: the number of goals scored by the team during the match, including the extra time (not counting penalties);
- scoreHT: the number of goals scored by the team during the first half of the match;
- scoreP: the total number of goals scored by the team after the penalties; - side: the team side in the match (it can be ”home” or ”away”);
- teamId: the identifier of the team;
- coachId: the identifier of the team’s coach;
- bench: the list of the team’s players that started the match in the bench and some basic statistics about their performance during the match (goals, own goals, cards);
- lineup: the list of the team’s players in the starting lineup and some basic statistics about their performance during the match (goals, own goals, cards);
- substitutions: the list of team’s substitutions during the match, describing the players involved and the minute of the substitution.

Players data (players.json)

- birthArea: geographic information about the player’s birth area;
- birthDate: the birth date of the player, in the format ”YYYY-MM-DD”;
- currentNationalTeamId: the identifier of the national team where the players currently plays;
- currentTeamId: the identifier of the team where the player plays for. The identifier refers to the field ”wyId” in a team document;
- firstName: the first name of the player;

- lastName: the last name of the player;
- foot: the preferred foot of the player;
- height: the height of the player (in centimeters);
- middleName: the middle name (if any) of the player;
- passportArea: the geographic area associated with the player's current passport;
- role: the main role of the player. It is a subdocument containing the role's name and two abbreviations of it;
- shortName2: the short name of the player;
- weight: the weight of the player (in kilograms);
- wyId: the identifier of the player, assigned by Wyscout.

REFERENCES

Biermann, C. (2019). *Football Hackers: The Science and Art of a Data Revolution*. Blink Publishing.

Exame (2022). Estudo aponta os clubes mais valiosos do brasil. acessado em 06-05-2023 em https://exame.com/esporte/estudo-aponta-os-clubes-mais-valiosos-do-brasil-veja-o-ranking/.

Forbes (2022). The world's most valuable soccer teams 2022: Real madrid, worth 5.1 billion, is back on top. acessado em 05-05-2023 em https://www.forbes.com/sites/mikeozanian/2022/05/26/the-worlds-most-valuable-soccer-teams-2022-real-madrid-worth-51-billion-back-on-top/.

Mead, J., O'Hare, A., and McMenemy, P. (2023). Expected goals in football: Improving model performance and demonstrating value. *PLoS ONE*, 18(4).

Nielsen (2022). The 2022 world football report. acessado em 05-05-2023 em https://www.nielsen.com/wp-content/uploads/sites/2/2022/07/nielsen-world-football-report-2022.pdf.

Pappalardo, L., Cintia, P., and Rossi, A. (2019). A public data set of spatio-temporal match events in soccer competitions. *Scientific Data*, 236(6).

Sumpter, D. (2017). *Soccermatics: Mathematical Adventures in the Beautiful Game*. Bloomsbury Sigma.

Tempo, J. O. (2021). Galo é pioneiro e lança setor analytics, que 'transforma' dados em conhecimento, jornal o tempo. acessado em 06-05-2023 em https://www.otempo.com.br/sports/atletico/galo-e-pioneiro-e-lanca-setor-analytics-que-transforma-dados-em-conhecimento-1.2486872.

Umami, I., Gutama, D. H., and Hatta, H. R. (2021). Implementing the expected goal (xg) model to predict scores in soccer matches. *International Journal of Informatics and Information Systems*, 4(1).