Bachelor Thesis

Advancing in player performance analysis for men's professional football: Machine learning approaches for expected Goals

Sina Haghgoo

Student ID: 11909041

Supervisor: Univ.Prof.Dr.Kurt Hornik

Co-Supervisor: Dipl.-Ing.Robert Bajons, MSc (WU)

Department of Statistics and Mathematics, Vienna University of Economics and Business, Welthandelsplatz 1, 1020 Vienna, Austria

Abstract

In recent years, many different metrics have emerged from the world of football analytics, which aim to assess performance on the pitch beyond traditional statistics like possession or shots. One example are expected Goals (xG): This metric quantifies the quality of chances during a game and therefore allows to evaluate both teams and players in terms of creating and converting goal scoring opportunities. The main goal of our thesis is to develop an expected Goals model and conduct a performance analysis on players of the FIFA 2022 World Cup. Our further objective is to analyse whether the biggest standout players from our analysis are also those with significant changes in market value after the end of the tournament. By developing and comparing several suitable classification algorithms, the aim is to identify the most robust xG model for our practical implementation. For that purpose, we utilize publicly available event data from Statsbomb and feature engineer both basic and advanced features for modelling. The knowledge for that process is gained via a comprehensive literature review of existing approaches. Our results indicate that both basic features like distance or angle and advanced variables like the applied pressure indeed have big effect on the assigned goal probability. Furthermore, our player performance analysis shows that standout players from our analysis are as expected, among the ones who received the biggest market value upgrades after the tournament.

Contents

1 Introduction

In the dynamic and constantly evolving world of professional football, the pursuit for new and more advanced methods for player performance analysis has become vital. Big transfer fees for acquiring new players are increasing continuously [1] and this shift in the transfer market is further amplified with the rise of clubs from Saudi Arabia who even exceed the spendings of many European top teams [2]. Due to these developments, the ability to assess and monitor the performance of players has become more important than ever, since teams need to determine whether their current players and potential new transfers are over- or underperforming. Back in the days coaches and analysts primarily used video footage and metrics such as goals, assists or shots to analyse the performance of individual players. However, the randomness and low-scoring nature of football are reasons why traditional evaluation methods often fall short in providing an appropriate assessment of individual performance [3]. In todays AI-driven world, machine learning algorithms can help in the quantification and evaluation of both player and team performance, that go beyond those conventional metrics. Consequently, the development of more sophisticated metrics is continuously pushed forward by football analysts worldwide.

One of the most notable advancements in football analysis is the introduction of the expected Goals (xG) metric. In short, expected Goals are a probabilistic estimation of goal scoring opportunities, based on their likelihood of resulting in a goal. Various factors like distance, angle or opponents in range can influence the assigned probability values significantly. By implementing these predictors into a machine learning model, one can not only predict the xG value of a particular shot but also uncover patterns of goalscoring chances and get strategic insights about the performance of different players and teams. [4, 5, 6]

1.1 Research goals and method

The goal of this thesis is to build and compare various expected Goal models via different machine learning algorithms, in order to identify the most robust and efficient one for assessing player performance. Particularly, one of the main emphasises lies in how the xG probabilities can be utilized for detecting over- and underperformance of individual players and to examine the impact of expected Goals on their market value developments.

We start by conducting a literature review of the metric itself and existing xG-approaches from available research studies, where we investigate the integrated features and techniques. This forms the basis for the development of our own models, which will be developed with publicly available event data from Statsbomb [7]. Ultimately, our most efficient approach will be used for a performance analysis of players who participated in the 2022 World Cup. Consequently, this thesis aims to bridge the gap between theoretical exploration and practical implementation.

2 Overview of expected Goals

As stated earlier, expected Goals are a statistical approach for quantifying the quality of a shot, by calculating the likelihood that it converts into a goal, without accounting for unforeseen factors [4, 5, 6, 8]. The xG-calculations are based on an extensive analysis of historical shots, which were all gathered from various different matches. Since the basic concept of expected Goals calculation can be viewed as a classification problem [9], researchers utilize different classification algorithms and train the models with available shot data. This strategy enables to determine and predict the goal scoring probability of every chance.

A straightforward example to demonstrate the functionality of expected Goals calculation can be illustrated with a penalty: Since all penalties in professional football share the same characteristics, such as fixed distance and angle, a simplified model with those basic types of predictors can be sufficient enough to estimate the goal probability. In practise this results in an expected Goal probability of approximately 0.78 [4, 5, 6]. This means that players have a 78% chance of scoring from the penalty spot, translating to the successful conversion of roughly three out of four penalties. However, it is important to note that other types of shots need a more careful and complex approach, since each of those presents its own unique challenges and can occur under a much wider set of circumstances.

2.1 Relevance for performance analysis

Expected Goals are not only a statistical measure for the quality of chances but in many cases also a better performance indicator, compared to traditional metrics [4]. To demonstrate that we take a comparison from the 2022/23 season between Rúben Neves and Alexis Sanchez, based on their xG-values from Opta [4]: Both of them had 63 total shots in the entire season but Sánchez managed to score twelve goals in comparison to Neves three goal tally. Although it may seem that Sánchez had a significantly more efficient season, the xG-values clearly falsify this statement. In fact Neves expected Goals of 2.8 was much much lower compared to the 10.2 from Alexis Sánchez, indicating the worse quality of chances that he had. Their expected Goals also showcase that even Neves managed to slightly overperform in the 2022/23 season in terms of goal scoring, despite only managing to score three goals from 63 shots.

Additionally, expected Goals can deal much better with the randomness and uncertainty of this low-scoring sport [3]. Due to the unpredictability of football, a match result can contain almost as much uncertainty (noise) as meaningful information about the quality of teams and players (signal) [10]. Consequently, the usage of goals alone as the main performance metric can introduce many random fluctuations that don't reflect the actual performance of players and teams. Expected Goals on the other hand are less influenced by these noises that lay outside a player's control and more reflective of their actual performances.

Furthermore, Michael Caleys [11] research revealed that expected Goals have a higher correlation to future goals than actual goals or other metrics such as shots and therefore being the better predictor for future performance of both teams and players. Especially in the medium run of a season (6-13 games), expected Goals tend to reflect better how well teams and players are performing, in comparison to those traditional indicators [10].

3 Literature review

In the following section we will review existing approaches from the literature, where we mainly focus on the implemented features and the approach that is used to derive the xG probabilities. This analysis serves as the foundation for the development of our own models, which aim to integrate the strengths of the reviewed techniques and mitigate their limitations.

It is important to note that the predictors, which the existing studies integrate, can come from two different types of data: Event and tracking [9]. Event data records all the actions that occur during a specific match such as shots, passes or fouls, thus focusing primarily on the interactions between players and the ball [12]. Each event is documented with the corresponding coordinates of the pitch and the players involved. The relevant data for that is gathered manually by dedicated tagging software [9]. Tracking data on the other hand exceeds the information capability of event-data, by also providing details about the positions of all the entities on the pitch and not only those who are involved in specific actions - capturing details up to 25 times a second [13].

Although there is some debate about the origin of expected Goals, Pollard and Reep were the first authors who calculated an xG-model in their 1997 study for estimating the effectiveness of team possession on the ball [14]: In their approach they used event data from 22 matches of the 1986 world cup, which was obtained manually by tagging discrete events like shots, passes, possession changes or fouls from team possessions. Their ultimate goal was to determine the expected outcome of each of the possession plays. Whenever one of the ball controls resulted in a shot, the goal probability was calculated with a Logistic Regression approach. The authors split the dataset, based on the shot type and created two different models: One for shots with foot and one for headers. In estimating the goal probability for shots with foot, the following features were identified as significant predictors:

- Distance between the middle of the goal line and shot location
- Angle between shot location and goal posts in radians
- Binary variable indicating whether the shot taker was more than 1 yards away from the closest opponent
- Play pattern leading to goal (i.e. set piece or open play)

For headers, the distance, angle and play pattern turned out to have a significant impact on the assigned goal probability. It is important to emphasise, that especially distance and angle serve as fundamental features in nearly all the reviewed models [8, 9, 15, 16, 3, 13, 14, 17, 18], underlying their significant importance.

Brechot and Flepp [3] used event data for over 7000 games, from the top five European leagues (collected between the 2013/14 and 2016/17 season) and developed a Logistic Regression model were beside angle and distance the shot body part was also considered as a predictor. However, unlike Pollard and Reeps [14] approach, the authors developed one model for both headers and kicked shots and created a dummy variable to differentiate between both shot types. Based on the obtained coefficients, it turned out that headers have a significantly negative impact on the goal scoring probability, compared to shoots with foot. This can be attributed to the sample size of headers in their data and the fact that directing headers with precision can be considerably more difficult compared to kicked shots, especially from longer distances.

Lucey et al. [19] primarily focused on strategic features such as positional attributes and the type of play for their Logistic Regression. Concretely they used tracking data for approximately 10.000 shots and analysed a ten second window of play before each shot. By mainly concentrating on spatio-temporal aspects, they concluded that features such as defender-proximity have a significant influence on the goal probability. The method for determining the proximity of opponents is more complex compared to Pollard and Reeps [14] approach and involves calculating the euclidean distance from the shot location to each defender present in the shot area and the two goal posts. This area which is visualized in Figure 1, is a triangle with the baseline being along the goal line and one vertex being on the shot location. A point-in-polygon calculation is used to determine if a defender is within this triangle. The total number of defenders inside this polygon is also considered as a predictor.

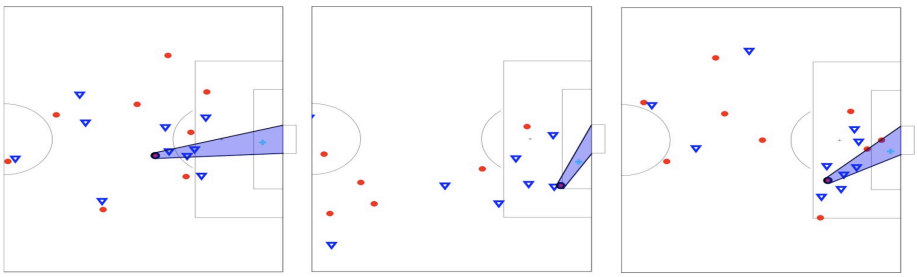

Figure 1: Opponents between Shot Location and Goalposts according to Lucey et al. [19]

In addition to that, the authors also integrated attacking features, including the play pattern and the available space to the assister of the shot. What is particularly interesting are features regarding the speed of play, such as the pace of the attackers towards goal. Along with the previous predictors, all these variables significantly contribute to the predictive ability of the authors model.

One aspect that is also implemented in various xG models are features regarding the quality of individual players. The underlying assumption for including player quality features is based on the idea that they can be used as a proxy for the skill level of a player and their composure in critical situations during a match. Variables from this area are integrated in the study of Mead et al [9]: For their xG-models the authors used event data from the top five European leagues of the 2017/18 season and created several different models including tree-based algorithms like Random Forest, AdaBoost and Extreme Gradient Boosting. The quality indicator which was used as predictor is called "PlayeRank": PlayeRank, first developed by Pappalardo et al. [20], is a performance metric, which is derived by assessing the role-based performance of each outfield player in a match and analysing all their on-the-field actions like passes, shots or interceptions. The cumulative PlayeRank value for each player is derived by summing up the PlayeRank scores of all the previous matches in which the player has participated in [9]. This metric enables a role-based performance comparison for each individual player on the field. Additionally the authors used the market value of all shot takers, which was manually obtained from transfermarkt.com. Beside player quality features, predictors regarding psychological aspects of the game were also included in the study. Among these were features like match attendance and whether the team of the shot taker had home advantage or not. Furthermore, the age was also considered as predictor, in order to investigate if the level of experience affects the likelihood of a goal. By conducting a feature importance analysis and looking at the gain of predictability for each feature on the models output, especially the market value variable turned out to have a big impact on the assigned goal probability. On the other hand, the Attendance, PlayeRank and Player Age features all had a comparably low effect, while the Home/Away variable led to practically no improvements in predictability. Especially the small influence of the PlayeRank predictor could be attributed to the fact that its main strength lies in its ability to evaluate players based on their on-field roles and actions, which go beyond the quantification of goal scoring chances.

The research of Eggels et al. [18] aims to determine the winner of a particular game by aggregating the individual xG values that occur during a game. Concretely this is achieved with the usage of event and tracking data for roughly 12.000 shots and with four different classification algorithms: Logistic Regression, Decision Tree, Random Forest, and an AdaBoost [18]. Variables regarding the quality of individual players are web scraped in form of player-ratings from the FIFA game. Unlike Mead et al [9] models, the authors not only included those player-quality features for the shot takers but also for the goalkeepers, since a better goalkeeper can significantly worsen the goal scoring likelihood of a chance.

In conclusion, we can observe that the spectrum of implemented features ranges from basic predictors like distance, angle or shot body part to more sophisticated ones, including opponent proximity and player quality features like market values and player ratings. Regarding the utilized machine learning algorithms, we primarily notice that most of the examined research studies integrate a logistic regression approach, while some also work with more complex techniques like tree-based algorithms.

4 Methodology

In the following section of this thesis, we start by describing the concrete materials that we use for modelling. Following that, we give an overview and explanation of all the integrated features, including the utilized machine learning algorithms and the rationale behind their inclusion. Lastly, a comprehensive summary of our modelling approach and evaluation technique is provided. Overall, this ensures that our entire procedure is reproducible and transparent to the reader.

4.1 Materials

All the utilized data for this thesis is based on publicly available event data from Statsbomb's github repository [7] and is extracted via their respective Statsbombpy module in Python. What sets their data apart from other options is the inclusion of positional data, which primarily contains information about the location of all relevant players, at the time of a shot and will be utilized for the feature engineering of positional attributes. In order to train our models, we take all available records of the 2015/16 season, across the top five European Leagues:

- Bundesliga
- Premier League
- Ligue 1
- La Liga
- Serie A

Extracting the shots from these matches, result in a dataset with 36510 entries. Throughout the 1446 matches available, the fetched records contain a total tally of 3462 goals, with an average of 2.7 goals per match. Penalties where excluded from this thesis due to their uniform characteristics and fixed expected goal values [4, 5, 6]. It is important to note that no data cleaning steps were required to use the extracted dataset for the next steps.

4.2 Feature selection and engineering

The selection of all features is mainly based on the gained insights from the conducted literature review. However, since some of them are rarer in the expected Goals domain or involve more complex calculations, a brief explanation and rationale behind their inclusion will be provided. In short, the following features are included:

- **Distance (float)**: Euclidean distance between shot location and center of goal.

- **Angle (float)**: Derived by using the arctangent function and calculating the angle between shot location and goal posts.

- **Shot Body Part (categorical)**: Indicates whether the shot was taken with foot or head.

- **Play Pattern leading to goal (categorical)**: Type of play leading to goal (i.e open play, corner or free kick), in order to investigate the influence of different attacking settings on goal probability.

- **Vertical Distance to goalkeeper (float)**: Distance on the y-coordinate of the pitch between shot taker and goalkeeper.

- **Market Value of shot taker (float)**: Estimated worth of shot taker, at the time of shooting and according to transfermarkt.com. Used as proxy for player quality.

- **Opponents in Triangle (int)**: Number of opponents within the triangular area from shot location towards goal posts. Similar to approach in Figure 1.

- **Pressure Front (int) and Pressure Behind (int)**: Number of opponents positioned in a circular area around the shot location, called "pressure radius". The radius of this circle changes, based on the distance to goal. The further a shot is away, the larger the considered radius and vice versa. By counting the opponents who are located inside this pressure radius and in front or behind the shot taker, one can observe how defensive pressure from both sides impacts the goal probability.

- **Strong foot/ big height (bool)**: For kicked shots this variable indicates if it is taken with the strong foot and for headers whether it came from a player with a height above 185cm and who therefore had a physical advantage. The information regarding strong foot and height comes from transfermarkt.com.

- **Position (categorical)**: Indicates whether the shot taker was a defender, midfielder or attacker.

The two most basic features that we included are the distance and the angle, which were fundamental variables in most of the reviewed approaches from the literature [8, 9, 15, 16, 3, 13, 14, 17, 18]. For our models, the distance feature is derived by calculating the euclidean distance between shot location and the center of the goal , while our strategy to compute the shot angle θ is based on the following approach by David Sumpter [21]:

$$\tan(\theta) = \frac{7.32 \times x}{x^2 + y^2 - \left(\frac{7.32}{2}\right)^2}$$

Here x and y represent the coordinates of the shot. Concretely x is the horizontal distance between shot location and goal line, along the edges of the pitch, while y is the vertical distance from the center of the pitch. The constant 7.32 is a standard measure for the width of the goal in meters.

For the Shot Body Part feature our initial idea was to create two separate models: One for headers and one for shots with foot. However, due to the sparsity of headers in the data, we decided to implement both shot types in the same model and use this boolean feature as distinction between the two shot types, similar to Brechots and Flepps [3] approach.

Our literature review revealed the importance of predictors that are based on positional data. Our main goal here is to enhance the predictability, by capturing the impact of defensive pressure and the available space for placing a shot. One feature that does that for us is "Opponents in Triangle". This variable counts the number of opponents who are present inside the triangular area, from the shot location towards the two goalposts, similar to Figure 1. Our approach starts by defining a triangle where the three vertices are formed based on the coordinates of the shot location and the two goal posts. Then the number of opponents in the resulted triangle is counted by using vector cross products. For each opponent we compare his position to all the sides of the triangle. By applying the cross product formula, we get three values for each player:

- Cross Product 1: Compares the opponent position with the triangle edge from vertex 1 to vertex 2
- Cross Product 2: Compares the opponent position with the triangle edge from vertex 2 to vertex 3
- Cross Product 3: Compares the opponent position with the triangle edge from vertex 3 to vertex 1

If all the three cross products have the same sign (positive or negative), then the opponent is assumed to be within the boundaries of the triangle. The idea and practical implementation for counting if a player is inside the triangle, is based on David Sumpters and Alexander Andrzejewskis expected Goals model [22].

The findings of the literature review also reveal the importance of features that reflect the individual quality of shot takers. For that we define the "Market value of shot taker" variable, representing the estimated worth of players, at the time they took the shot. All the information for that is based on a scraped dataset from Transfermarkt, which is publicly available at Kaggle [23]. This predictor is considered to be a proxy for the overall skill-level of a player, since the stated market value mainly shows the perceived quality of a player, based on his past performances.

Two further developed predictors are the "Pressure Front" and "Pressure Behind" features. For both of these variables we create a radius around the shot location and then count the number of opponents who are positioned inside this

circular area, which we call "pressure radius". If an opponent falls within the pressure radius, we check whether he is in front or behind of the shot taker, so that we can capture the applied pressure from both sides. The considered radius changes based on the shot distance. Concretely, the further a player is away from goal, the bigger the radius gets and vice versa. Our aim is to define a pressure radius that increases first but then plateaus, as the distance gets larger. After an extensive analysis of possible ways to define the pressure radius, we decided to derive it with the following custom function, which gives us the best results:

$$PR = 2 \cdot (\ln(distance + 1))$$

by taking the natural log of the shot distance and then doubling it with the corresponding factor, we get a reasonably big and realistic radius for our approach and the characteristics of Statsbomb's positional data (adding 1 to the shot distance avoids negative and undefined values). The general idea behind the inclusion of these two variables is that a higher defensive pressure can decrease the quality of a chance significantly, since the shot taker has less time to prepare and can be forced to rush in the decision making. The idea for that approach is based on the Statsbomb article from Derrick Yam [24].

4.3 Utilized machine learning algorithms

There are many classification techniques that can be applied to the expected Goals domain. For our analysis we opted for Logistic Regression and two tree based algorithms: Random Forest and Extreme Gradient Boosting (XGBoost). Each of these offers advantages but also comes with its limits.

Logistic Regression is in fact the most basic modelling approach for our goal and also the most frequently used one in the available research studies. It models the probability of our outcome variable *Goal*, by using the sigmoid function as input to transform the linear combination of all predictors into probabilities ranging between 0 and 1. In order to integrate this linear term into the sigmoid function, Logistic Regression models the log odds of observing the outcome *Goal* as linear combination of all the available predictors x [25]:

$$\log\left(\frac{p(Goal)}{1 - p(Goal)}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n$$

Therefore all the coefficients are interpreted in terms of log odds. While β_0 gives us the baseline log-odds when all dependent variables have a value equal to zero, the other coefficients show that by changing their values to the factor of 1, the log-odds for observing *Goal*, changes by the corresponding coefficient. As we can see, the main strength of Logistic Regression lies in its high interpretability. Nevertheless, it is crucial to outline that this classification technique also assumes a linear relationship between predictors and log odds of the outcome,

which is not always the case. Consequently, it is recommended to consider other models as well, which can potentially exceed the predictive ability of Logistic Regression and capture more complex patterns of goal scoring opportunities.

As stated, we selected two tree-based algorithms - Random Forest and XGBoost, which are both ensemble methods. Random Forests are based on combining multiple decision trees and each one of them is built in parallel, by considering a random sample of predictors from the total number of features [25]. With this approach, the correlation between individual trees can be reduced and the Random Forest can capture more patterns of the data [25]. Consequently, a Forest can handle more complex relationships, compared to Logistic Regression. However, these models are therefore also more prone to overfitting, if not tuned properly [25]. Furthermore, they come at the cost of losing the high interpretability of logistic regression.

Unlike Random Forests, boosting algorithms like XGBoost use multiple weak learners (trees) that perform slightly better than random [26] and grow them sequentially, based on the obtained information from previously grown trees [25]. As a result, each subsequent tree learns from the mistakes of the previous ones. This is one of the main reasons why these types of boosting algorithms often tend to outperform a Random Forest [25]. Nevertheless, the reduced interpretability is also a disadvantage for this machine learning algorithm.

4.4 Modelling approach

Each of our models will be trained with 70 percent of the dataset, while 30 percent will be utilized for testing. Evaluating on unseen testing data ensures that the corresponding algorithm learns generalized patterns from the training data, rather than merely memorizing it. Furthermore, we stratify the instances of goals in our training and testing data to ensure that the same ratio of positive and negative classes is maintained in both datasets.

The preprocessing and modelling is done inside a pipeline with the sckit-learn library in Python. These pipelines can help in the automation process of all the cycle steps of a particular machine learning model, including preprocessing and model training and evaluation, so that a completely reproducible workflow can be provided [27]. Our preprocessing pipeline is illustrated in Figure 2 and shows the entire transformation process of our features. The numeric features are imputed with the mean, while for the categorical and boolean variables, we use the most frequent category of the corresponding column. Following that we standard scale the numeric features by subtracting the mean and then diving by the standard deviation. In general, this can be vital for certain machine learning algorithms like the distance based KNN or for specific hyperparameter settings, such as the l1 and l2 regularizers of Logistic Regression. Our categorical columns are transformed into a numeric format with One Hot encoding, while the boolean columns use a FunctionTransformer before imputing, which converts the True and False values into integers of 1 and 0, ensuring that all the input features are in a numeric format.

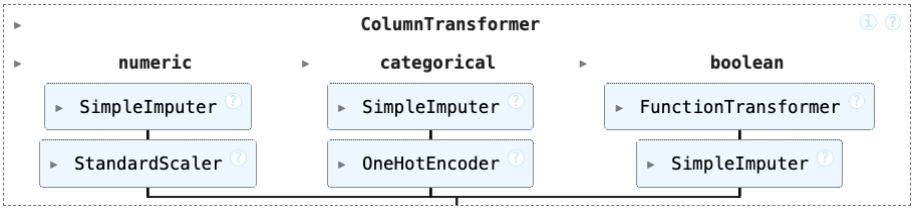

Figure 2: Preprocessing Pipeline with sckit-learn

4.4.1 Hyperparameters

For each of our algorithms we tune a set of chosen hyperparameters, which ensures that our models are set up with the optimal configuration of parameters that are defined prior the training process.

Starting with our Logistic Regression model, we identified the following hyperparameters:

- **penalty** - [l1, l2, None]: Applied technique to penalize complex models and reduce overfitting. L1 (Lasso Regularization) adds the absolute value of the coefficients as penalty, while l2 (Ridge Regularization) adds it‘s square. By also setting a possible parameter value to "None", we test whether the model performs better with no regularization technique.
- **c** - [0.001, 0.01, 0.05, 0.1, 0.5, 1, 5, 10]: Controls the inverse strength of the applied regularization technique (not considered in case of "None" as value for the penalty). Smaller values for c indicate more applied regularization.

Moving on to the Random Forest, we similarly conducted a comprehensive research of possible hyperparameters and selected the most suitable ones for our data and algorithm. Since Random Forests are ensemble methods that are based on multiple decision trees, a different set of tree-based hyperparameters are integrated:

- **n_estimators** - [50, 80, 100, 120, 150, 200, 250, 300]: Number of trees in the forest. While a larger number of trees can improve the performance of the model and lead to a decrease in overfitting (because the predictions of the trees are averaged), it also worsens the computational efficiency.
- **max_depth** - [2, 3, 4, 5, 6, 7, 8]: Maximum level of depth for each tree, to limit the extent of complexity for each tree in the forest and ultimately also the risk of overfitting.
- **min_samples_split** - [1, 2, 3, 4, 5, 6, 7, 10, 15]: Samples required to split an internal node further, which also limits the complexity and overfitting proneness of trees in the forest.

Lastly, our tree boosting algorithm XGBoost, has the following hyperparameter settings:

- **learning_rate** - [0.001, 0.01, 0.10, 0.20, 0.3, 0.50, 1]: Shrinkage size of feature weights. Smaller values limit and slow down the extent of correction made by new trees and therefore prevent overfitting more.
- **max_depth** - [3, 4, 6, 10, 15]: Depth of each sequentially created tree, to control the risk of overfitting.

We search for the optimal combination of different values with the Randomized Search technique. This approach randomly selects a combination of defined values from each of the respective hyperparameters and allows for a more time and computational efficient search approach, compared to other techniques such as Grid Search [28]. During each combination, ten fold cross validation is performed, in order to find the optimal set of tuning parameters.

4.5 Evaluation and feature importance

Once the optimal hyperparameter settings for each algorithm are found, we compare their Log Loss scores in order to find the best performing model and use log loss as the corresponding loss criterion for the evaluation process. Concretely, this metric evaluates how far away the predicted probability is from the actual label of the corresponding observation [29] and is derived with [29]:

$$\text{Logloss} = -\frac{1}{N}\sum_{i=1}^{N}\left(y_i \ln(p_i) + (1-y_i)\ln(1-p_i)\right)$$

with N being the total number of observations, y_i the actual label of the observation and p_i the probability of the positive class. The reason why we opted for Log Loss as evaluation metric is due to its threshold independence. This is particularly useful for the expected Goals domain, since our aim is to focus on the accuracy of the prediction itself. Optimizing the models with threshold dependent techniques such as accuracy, precision or recall could compromise the integrity of the results because only a small proportion of shots are converted into goals [3]. With the optimal hyperparameter settings for each model, the training and test scores help us to understand which algorithm performed best and is therefore most suitable for our player performance analysis.

Based on the models that were optimized on Log Loss, we also extract their Area Under the Receiving Operating Curve (AUROC) scores, for the purpose of assessing the models performance from an alternative perspective, specifically its ability to differentiate between goals and misses across different thresholds. In order to derive the AUROC, we first plot the Receiver Operating Characteristic Curve (ROC), which is based on the True Positive Rate (TPR) and False

Positive Rate (FPR) of the corresponding classifier. Both these metrics can be derived with the confusion matrix in Table 1. The TPR measures the proportion of actual positives in the data, which are also correctly classified as positive by the corresponding algorithm and the FPR is defined as the proportion of negative instances, which were wrongly classified as positive:

$$\text{TPR} = \frac{TP}{TP + FN}$$

$$\text{FPR} = \frac{FP}{FP + TN}$$

While the Receiver Operating Characteristic Curve (ROC) plots the True Positive rate of an classifier against the False Positive rate at various different thresholds, the Area Under the ROC curve measures the space beneath that curve [30]. Concretely, it is defined as the probability of ranking a randomly chosen positive instance higher than a negative one, with an AUC of 0.5 corresponding to random guessing and 1.0 to perfect accuracy in terms of differentiating between positive and negative classes [30].

Lastly, the best performing algorithm will be further examined via a feature importance analysis. This helps us to gain an understanding into the most influential factors of goal scoring opportunities, according to the model and consequently to understand how the predictions are derived and whether we can observe unexpected relationships between our variables and their effect on the assigned goal scoring probability. Since our aim is to additionally identify high and low impacting predictors, we utilize SHAP values. SHAP (SHapley Additive exPlanations) is a game theoretical technique to explain the predictions of machine learning models, by quantifying the impact of each feature on the models output [31, 32]. Therefore all the predictors are assigned with an importance score, based on their average impact on the models prediction, with positive values for positively impacting features and vice versa for negatively impacting ones [31, 32].

	Predicted Positive	**Predicted Negative**
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

Table 1: Confusion Matrix

5 Explanatory data analysis (EDA)

In the following section we do an EDA of the described shots dataset, for the purpose of understanding the characteristics of the corresponding data. As stated earlier two of the most frequent predictors for any expected goal model are the distance and the angle of a shot [9]. When looking at the distribution of those features, we can confirm our initial thought: While many shots are taken from a longer range, mainly those with closer distances end up as goals and tighter angles provide a much bigger challenge in terms of goal scoring. By looking at the spatial distribution of misses and goals in Figure 4, we observe that misses are concentrated in comparably bigger distances compared to goals and that the angle of wide shots is also spread wider.

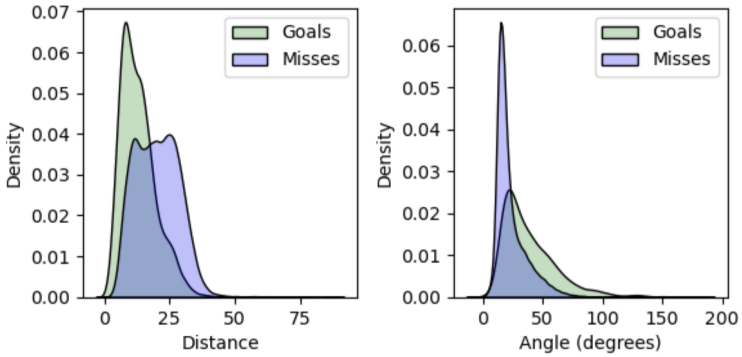

Figure 3: Density of Distance and Angle

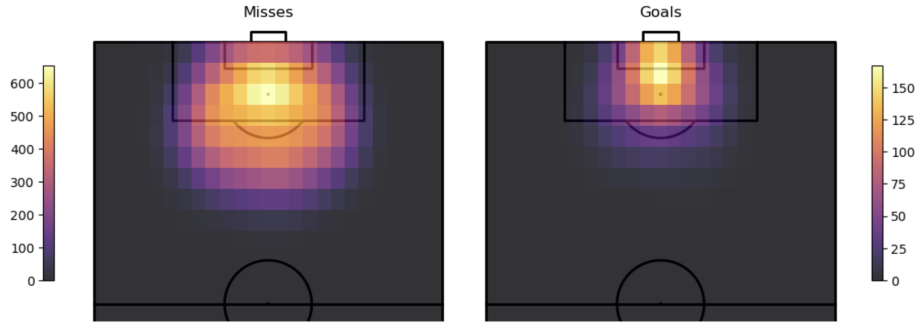

Figure 4: Goals and Misses

Furthermore, the goal conversion rate for both shot types in the data is similar. While 11% of all headers are converted into goals, shots with foot have a success rate of 9%. The violin plot on Figure 5 shows that headers are skewed towards much closer distances to the goal, whereas both goals and misses for kicked shots are spread considerably more. This is very likely due to the higher degree of force and direction that can be applied to these types of shots. We also see that both misses and goals with the foot tend to come from tighter angles, compared to headers.

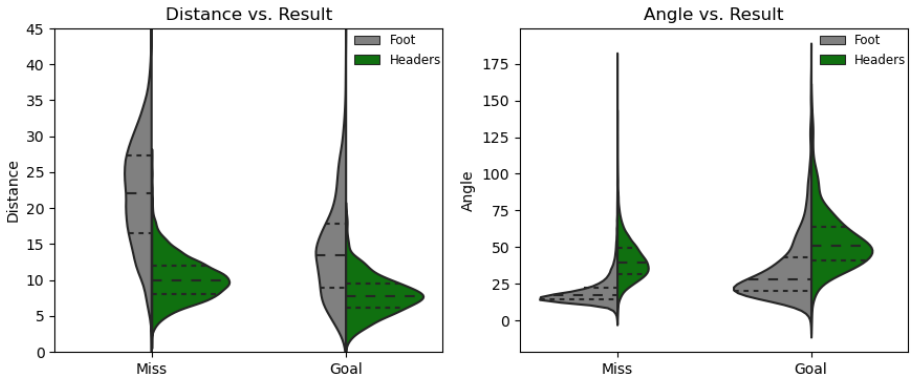

Figure 5: Violin Plot for Angle and Distance of Headers and kicked Shots

In addition to that, the goal conversion rate for shots with the strong foot and headers from players with a height above 185cm also reveals valuable insights: 68% of goals from kicked shots are scored with the strong foot of the shot taker and players above 185cm account for 54% of the entire header goals.

Regarding the market value of shot takers, we observe that players with smaller values have a much bigger shot/goal ratio, compared to the high-valued players. However, it is essential to point out that analysing plots like this, demands cautious interpretation. First, the majority of the players in our dataset has a market value that is closer to 0, compared to the much smaller proportion of high-valued players. In addition to that, this graph doesn't consider player positions, which also affects their responsibilities during a game and consequently their goal-scoring abilities. For example, unlike a forward, a defenders value isn't usually derived based on his goal scoring abilities but rather his defending abilities.

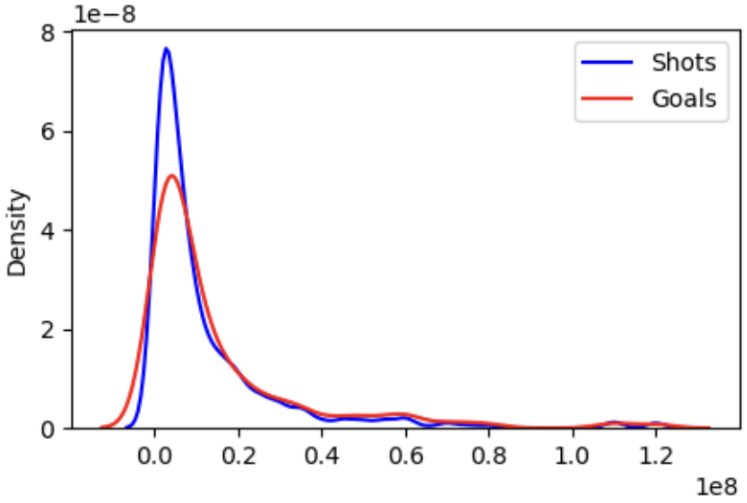

Figure 6: Market Value of Players, at Time of Shot

By investigating the temporal patterns of goals, we observe a clear pattern: When looking at intervals of 15 minutes, the number of goals seems to increase linear and only declines in the interval between minute 61 and 75. What is particularly interesting is the spike in the last interval, where in total 735 were scored, which could very likely be due to the increased pressure that teams apply as the match comes to an end.

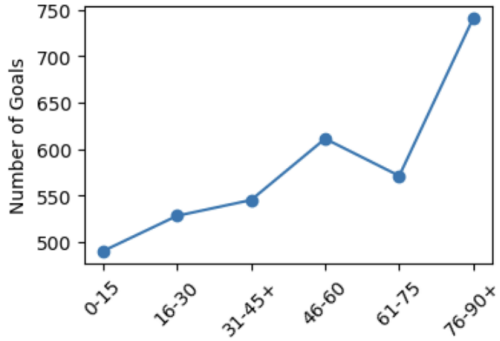

Figure 7: Goals in Intervals of 15 Minutes

Regarding defender proximity, our analysis clearly reveals a tendency for shots to be taken with a fewer number of opponents in range. One mentionable anomaly in Figure 8 is the low number of shots for zero opponents in range between the shot location and goalposts. This is very likely due to the rarity of such opportunities that occur for players during a match.

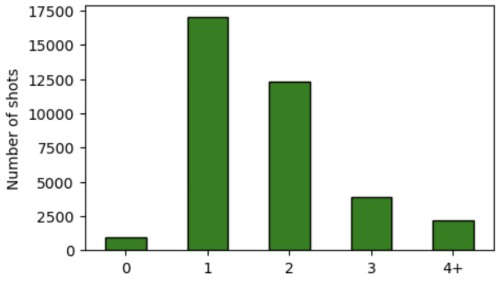

Figure 8: Number of Shots vs. Opponents in Triangle

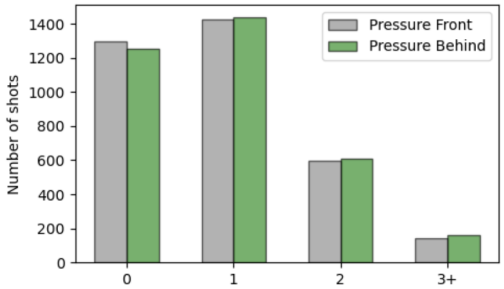

Figure 9: Number of Shots vs. Pressure Front and Behind

6 Evaluation results

As mentioned earlier, we optimize and evaluate our models based on their Log Loss scores and choose the best performing one for our player performance analysis. The final results are summarized in Table 2. In principle, all the models we developed performed well on the data and the resulting differences are only marginal. This implies that the patterns of our available data and features can also be captured by less complex modelling techniques on a very similar level. What can be particularly observed across all models is the consistent performance on both training and testing data, which indicates the lack of overfitting. While the Random Forest managed to outperform the Logistic Regression on the training data, the Logistic Regression achieved a better test score. Considering the small margins of the results, the Extreme Gradient Boosting algorithm managed to outperform both models on training and testing data. Consequently this means that based on our conducted model evaluation, the XGBoost is the performing model in terms of capturing the accuracy of the xG probabilities.

Model	**Log Loss training**	**Log Loss testing**
Logistic Regression	0.258	0.260
Random Forest	0.246	0.261
XGBoost	0.237	0.257

Table 2: Model Performance

With the models that were optimized on Log Loss, we also derived their AUROC. The results indicate that all the models are performing well in terms of distinguishing between goals and misses, across different thresholds. Like for the Log Loss, the findings reveal that that the XGBoost manages to slightly outperform both of the other algorithms with a AUC of 0.798. This means that there is a 80% probability of ranking a randomly chosen instance of "goal" higher than a randomly chosen "miss".

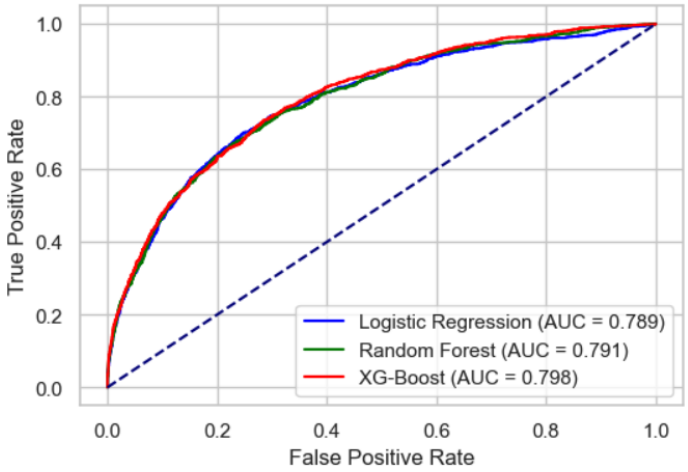

Figure 10: Area under Receiver the Operating Characteristic

By comparing the results of our model evaluation, we can observe that the XGBoost outperforms the other two algorithms in both of the investigated metrics. Consequently, we select this model for our player performance analysis.

7 Feature importance analysis

In the following section we conduct a feature importance analysis for our final model - XGBoost. By utilizing the SHAP values, we can investigate the impact of the relevant features on the expected Goal predictions. The SHAP summary plot in Figure 11 displays the variables, according to their significance for goal prediction, in descending order. Each data point on the graph corresponds to an observation from our training data and the color coding represents the value of the corresponding feature, with red indicating high values and blue low ones [31, 32].

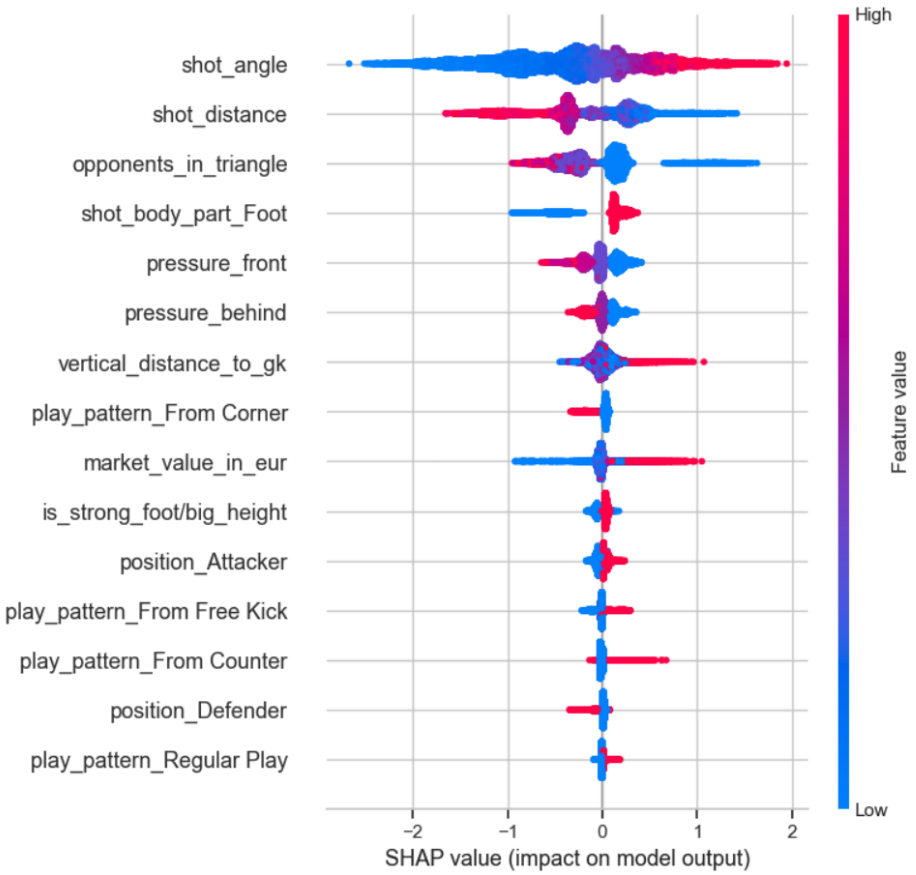

Figure 11: SHAP Summary Plot for XGBoost

Unsurprisingly, the two most predominant features for goal prediction are the angle and distance of the shot. Concretely, our analysis reveals that most of the shots with higher distances and tighter angles have a negative effect on

the assigned goal probability, which perfectly aligns with the principle logic of goal scoring. Interestingly, our model seems to classify the importance of angle higher than distance. This could very likely be due to the reason that even from further distances, wider angles can considerably help a player placing his shot more effectively, compared to closer distances and tighter angles.

Regarding opponent proximity, we observe that the "opponents in triangle" feature is the third most relevant predictor. The SHAP values confirm our initial thought that a lower number of opponents in the in the area between shot location and goal posts increases the xG-probability, while the effect for more opponents is opposite. Upon closer inspection we notice some instances where smaller values lead to a minor decrease in the xG probability. Similar patterns can be found for the pressure front and pressure behind features, with the main difference being that the impact on the models output is weaker, compared to the "opponents in triangle" feature. While both pressure variables have a very similar impact on the predictions of the model, the applied pressure from front is classified as more important, compared to pressure behind. The reason for that could be because pressure from front does not solely force shot takers into quicker decision making (that can lead to less composed finishes) but also obstructs their view of the goal, which consequently limits their ability for a precise shoot precisely.

Furthermore the shot body part seems to be the fourth most impactful feature for the models output. High values for the shot body part "Foot" indicate that the foot was used for the shot, while low values stand for headers, since we only used kicked shots and headers for our analysis. Despite the similar goal conversion rate for both headers and kicked shots (11% for headers and 9% kicked shots), headers seem to negatively influence the assigned goal probability. The sparsity of this shot type in our data and the fact that directing headers with precision from longer distances can be considerably more difficult compared to kicked shots, could be reasons for that.

Interestingly, the market value feature, which we used as proxy for player quality, only has a comparably lower impact in determining the expected Goal values. This limited predictive power could be because the majority of the shot takers in our dataset have a market value on the lower end, compared to the much smaller proportion of high-valued players. Nevertheless, the SHAP values for this variable are as predicted, since smaller market values decrease the assigned goal probability, while higher valued shot takers mainly increase the likelihood of a goal.

In addition to that, we notice that play patterns such as corners and counter attacks seem to be more relevant for the predictions of the model, compared to shots that occur from regular play. Regarding position, we observe that as expected, shots from attackers contribute positively to the assigned xG-probability and vice versa for defenders.

Overall, our SHAP-values show that the way most of our integrated features influence the models behaviour, closely aligns with our initial expectations.

8 Player Performance analysis

To analyse player performance, we take our XGBoost model and apply it on data from the FIFA 2022 World Cup. This is because it is the most recent data that Statsbomb offers at the time of conducting our research and can therefore provide the most valuable insights into the performance of currently active players. Nevertheless, it is important to note that our algorithm was trained on league data from 2015/16 and could therefore affect the results of this analysis! Another important information is that we also considered penalties in our player evaluation and assigned them a xG of 0.78, since this fixed value is considered to be a standard in the expected Goals domain [4, 5, 6].

For our analysis we focus on the top fifteen players with highest expected Goals throughout the tournament and examine who the biggest over- and underperformers are. However, as previously discused the xG metric is more robust in the medium run because it's less influenced by random fluctuations and noise [10], which can be especially a big factor in short knockout-tournaments like the World Cup. Therefore, we refine our analysis and only consider players who participated in at least four games, which corresponds to halfway through the tournament. Since our aim is also to focus on players with an adequate amount of playing time, we limit the performance comparison to players with a minimum average of 30 minutes per match, which allows us to look at both regular starters and substitutes with sufficient enough minutes. It is important to note that the information regarding playing time of each player is note integrated in StatsBomb's data. Consequently, we use an appropriate Kaggle dataset with the desired information [33], which itself is based on data from the football statistics website Fbref. Ultimately, this approach gives us a more representative and reflective performance analysis.

8.1 Players with highest xG

Figure 12 displays the top 15 players with the highest expected Goals and compares these values to the actual goal contributions. It's worth mentioning that all players on the plot are attackers, which is not that surprising, considering their assigned role on the pitch and their close positioning towards the opponents goal, which typically corresponds to higher quality of chances. In general we can observe that most players managed to overperform throughout the tournament and score more goals than their xG suggested, while few underperformed. It becomes clear right away that the two most impactful players are the two superstars Lionel Messi and Kylian Mbappe. In fact their higher number of goals and shots compared to other players, massively contributes to this elevated score. Concretely, each of them recorded almost 30 shots in total, with seven and eight goals respectively, while their closest competitor Oliver Giroud, accounted for 17 attempts and four goals. Interestingly, Messi is one of the few players from the top fifteen, who didn't score more goals than his xG suggested. However he still managed to perform closely to what would be expected, based on the quality of chances he had. In comparison, his main rival Mbappe even managed

to overperform slightly. Interestingly, despite overperforming (according to our model) and having the same amount of goal contributions compared to Lionel Messi, the Frenchman still fell short to him as player of the tournament, which is most likely because Mbappe lost to the Argentinian in the World Cup final. In general, five players on Figure 12 are WC finalists, which is as anticipated due to their larger amount of possible minutes and games. Surprisingly, World Cup winner Lautaro Martinez is the only player who didn't manage to score a single goal, despite being the player with the fourth most shots (14 in total).

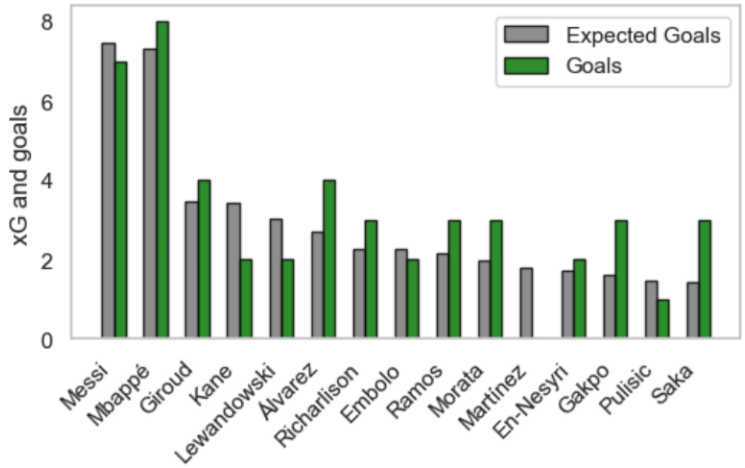

Figure 12: Top 15 players with highest expected Goals and their actual goals

8.2 Over- and underperformers

By examining the differences between xG and actual goals, we can investigate the biggest over- and underperformers. Table 3 displays the top tree most overperforming players, in descending order. Based on our results, English winger Bukayo Saka emerged as the biggest overperformer in terms of expected Goals and actual goals. Concretely he managed to score three goals out of seven shots, with an xG of 1.4 and across 288 minutes. Consequently scoring twice more than his expected tally. Dutch forward Cody Gakpo achieved similar results but with 162 more minutes on the pitch and a smaller difference between xG and goals. Lastly, World Cup winner Julián Álvarez had an xG of 2.7 in 460 minutes and managed to score four goals. However, he also had much more shots and a higher overall chance quality, compared to both Gakpo and Saka.

Player	Minutes	Shots	Goals	xG	xG/90
Bukayo Saka	288	7	3	1.4	0.4
Cody Gakpo	450	5	3	1.6	0.3
Julián Álvarez	464	11	4	2.7	0.5

Table 3: Overperformance Based on Expected Goals in World Cup 2022

In contrast to that, Table 4 displays the tree most underperforming players, in descending order. As mentioned earlier, World Cup winner Lautaro Martínez didn't score a single goal, despite 14 shots on target and a xG of 1.8. Unlike Martínez, Harry Kane and Robert Lewandowski managed to convert two of their ten attempts into goals but also fell short in terms of exceeding their goal expectations.

Player	Minutes	Shots	Goals	xG	xG/90
Lautaro Martínez	241	14	0	1.8	0.7
Harry Kane	402	10	2	3.4	0.8
Robert Lewandowski	360	10	2	3.0	0.8

Table 4: Underperformance Based on Expected Goals in World Cup 2022

8.3 Market value updates vs. expected Goals

By examining how the players estimated worth changed after the World Cup 2022, we can investigate whether the ones from our player performance analysis are also those with significant market value updates. For that purpose, we compare their market value from the start of the tournament, with the first available update after the end of the tournament and look for potential changes. Unfortunately, Transfermarkt only released market value upgrades after the World Cup for the best performing players of the tournament [34]. As a result, we only consider the market value adjustments from this specific update because the next main update would have been 4 months later and this would have skewed our results, since the performances from many club games would have been also considered.

Our final results are displayed on Figure 13. Among the 53 upgrades Transfermarkt made [34], no forward who wasn't involved in our analysis, received any significant upgrades! French superstar Kylian Mbappé managed to cement his position as the most valuable player in the world after his impressive performances and got a market value increase of 20 million. Young striker Julián Álvarez also received a big value increase, which is unsurprising, considering that he was one of the overall most overperforming players, according to our model. The other two players who emerged as most overperforming in our analysis, also got substantial value increases, with Gakpo being upgraded by 15 million euros and Saka by 10 million. In addition to that, we see that Goncalo Ramos and Richarlison, who were among the top fifteen with the highest expected Goals and scored more goals than their xG suggested, also benefited from value growth. Interestingly, Youssef En-Nesyri is the only player who was upgraded, despite not scoring more than the corresponding xG value implied. His increase is most likely attributed to Moroccos impressive run up to the semifinal of the World Cup. Furthermore, the market values from players like Messi or Giroud didn't change, despite good performances, which is most likely due to their high age.

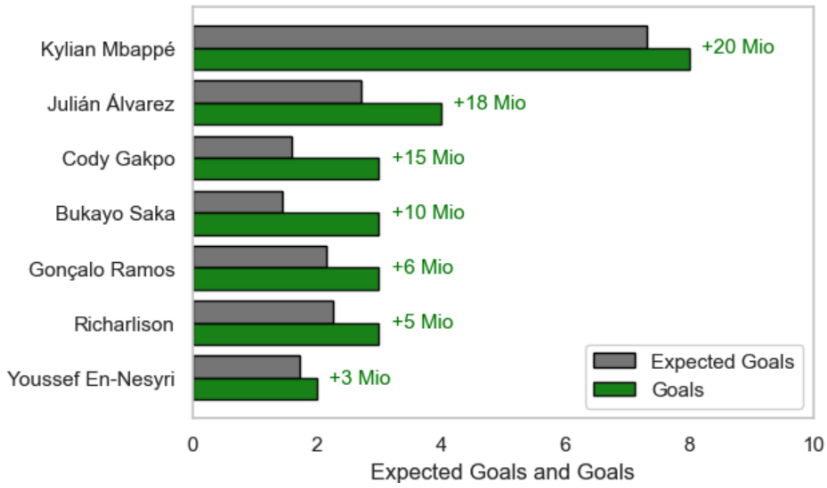

Figure 13: Market Value Adjustments in EUR. vs. Expected and Actual Goals

9 Limitations of expected Goals

Expected Goals provide numerous advantages to football analysis, such as enhanced tactical insights and a deeper understanding of player performances. However, despite all the possibilities that this metrics provides, it certainly has its limitations.

First, the xG metric should not be used as an universal player performance indicator due to its lack in considering player roles and their corresponding responsibilities during a game. For example a defenders worth to a team is usually not derived by the amount of threat he provides in attacking play or in front of goal but rather due to his level of defensive awareness and stability. In section 3 we mentioned PlayeRank [20] as a metric, which is supposed to asses player performance, based on their individual roles on the pitch. Metrics like this exceed the predictive capabilities of expected Goals as a more general performance indicator because they are not solely based on shots and therefore more expressive for performance assessment of non attacking players. In addition to that, just like actual goals, xG can involve a lot of noise in the first few games and could therefore skew the results of an analysis and evaluation [4]. Although in the medium run of a season, expected Goals tend to better reflect performance of teams and players compared to goals, it changes for the later stages of a season. According to David Sumpter [4], the noise in measurement for both metrics becomes only marginally different as the season advances and more games are played. Since goals are true and not a statistical model with errors, one should rather use them as performance indicator in the long run. These are just a few examples why the metric also comes with its drawbacks and should be used carefully in specific situations, despite the many advantages that it offers, like it's enhanced interpretability or simplicity.

10 Conclusion and future work

The main goal of our thesis was to develop a robust expected Goals model, based on the gained insights from our literature review and to conduct a player performance analysis with the corresponding model. For that purpose, we developed and evaluated several classification algorithms and identified the XGBoost as our most performant algorithm. Regarding integrated features, we saw that especially basic features like distance and angle are fundamental for the predictions of our best model and that positional variables like the opponents in front of the shot taker or the applied pressure from front and behind, also have an important impact. Lastly, the results of our player performance analysis shows that indeed the most outstanding players were the ones who received big market value upgrades.

For future work, several aspects of our current approach can be improved and fine tuned, in order to enhance the predictive capabilities of our analysis. First, training the models with more comprehensive data and further advanced predictors, could provide deeper insights into other important aspects that determine the quality of a chance, such as player movements or more detailed capture of applied pressure from opponents. Furthermore, the training and testing process of the algorithms could be done with more recent data, since the dynamics of the game change frequently over the time and can therefore have a big influence on the corresponding results. In addition to that, the basic concept of expected Goals can be applied and integrated into other more advanced areas of football analysis. One example are action value models, which aim to understand the impact each players makes, by quantifying their actions on the pitch. Here a metric like expected Goals can be integrated and calculate the probability of scoring, based on different types of actions, like certain passes for example. As we can see, there are many ways how our approach can be expanded and improved for an even more comprehensive and reflective analysis.

Appendix

A.1 Python code for selection of feature engineered variables

In this section we present the code for selected variables, which we explained more detailed in section 4.2 and involve more complex calculations. The entire code for this thesis was created with the programming language Python.

A.1.1 Opponent in triangle

After extracting the number of opponents in each shot freeze frame, we start by defining a triangle where the three points are formed via the shot location and the two goal posts. Then the number of opponent players in the resulted triangle is counted by using the cross product formula in relation to each vertex of the triangle. The idea for counting whether a player is inside the triangle is based on David Sumpters and Aleksander Andrzejewskis approach [22].

```
def opponents_triangle(shot, shots_df):
    freeze_frame_data = shots_df.loc[shots_df['id'] == shot["id"],'
        shot_freeze_frame'].values

    # Extracting opposition player locations
    player_positions = [player['location'] for player in
        freeze_frame_data[0] if not player['teammate']]

    # triangle vertices
    x1, y1 = 120, 36 # Top right post
    x2, y2 = 120, 44 # Bottom right post
    x3, y3 = shot["location"]

    # Counting players inside the triangle by calculating cross product
        of vectors that are formed by the vertices of the triangle
    count = 0
    for xp, yp in player_positions: # xp and yp are the player
        coordinates of the player positions
        c1 = (x2 - x1) * (yp - y1) - (y2 - y1) * (xp - x1) # 1) vertices
            of triangle
        c2 = (x3 - x2) * (yp - y2) - (y3 - y2) * (xp - x2) # 2) vertices
            of triangle
        c3 = (x1 - x3) * (yp - y3) - (y1 - y3) * (xp - x3) # 3) vertices
            of triangle
        if ((c1 <= 0) & (c2 <= 0) & (c3 <= 0)) or ((c1 >= 0) & (c2 >= 0)
            & (c3 >= 0)): # all the cross products have the same sign:
            player is inside triangle
            count += 1

    return count
```

A.1.2 Pressure front and behind

For defining the final pressure radius of the circle, we tried several different approaches and opted with the following technique: We logarithmize the shot distance with the natural logarithm and then multiple it with a suitable factor, in order to get a reasonable big and not too small radius for our approach. With this method we achieve a comprise between a realistic pressure radius that decreases with shot distance and has an appropriate size for the characteristics of Statsbomb's data. To check whether an opponent falls within the circle, we calculate the distance of the opponents from the center of the circle (shot location) and then compare it to the radius of the circle. If the distance from a defender is less or equal to the radius, then the point is inside the circle, otherwise it is outside. If an opponent is in the circle, we check whether he is behind or in front of the defender. The idea is based on the Statsbomb article from Derrick Yam [24].

```
def opponents_in_circle(shot, shots_df):
    freeze_frame_data = shots_df.loc[shots_df['id'] == shot["id"], '
        shot_freeze_frame'].values[0]
    player_positions = [player['location'] for player in
        freeze_frame_data if not player['teammate']]

    x0, y0 = shot["location"]
    r = np.clip((2*np.log(shot["shot_distance"] + 1)))
                                    # approach for getting the radius of
                                        the cycle
                                    # adding 1 to the distance incase the
                                         distance is 0 or below 1
    upper_count, lower_count = 0, 0
    for xp, yp in player_positions:
        # Check if a player falls within the circle
        if (xp - x0) ** 2 + (yp - y0) ** 2 <= r ** 2:
            if yp > y0:
                upper_count += 1 # Player is in the upper half
            elif yp =< y0:
                lower_count += 1 # Player is in the lower half

    return upper_count, lower_count
```

A.2 Further plots and tables from EDA

The plots and tables in this section are also based on our shots dataframe, which was used for training and testing the models.

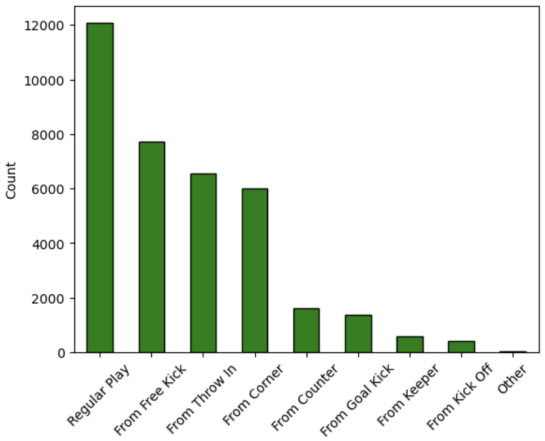

Figure 14: Number of Play Patterns of Shots

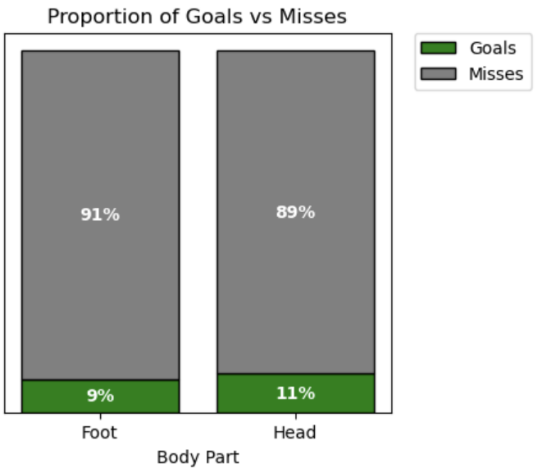

Figure 15: Proportion of Goals and Misses, per Shot Body Part

Interval	Goals
0-15	486
16-30	527
31-45+	541
46-60	606
61-75	567
76-90+	735

Table 5: Number of Goals per 15 Minute Interval

A.3 Results of hyperparameter tuning

The results of our hyperparameter tuning for the parameters which we identified in section 4.4.1 are as follow:

algorithm	hyperparameter	value
Logistic Regression	penalty	l1
Logistic Regression	regularization (C)	0.1
Random Forest	n_estimators	100
Random Forest	min_samples_split	4
Random Forest	max_depth	7
XGBoost	learning_rate	0.1
XGBoost	max_depth	4

Table 6: Results of Hyperparameter Tuning

A.4 Average impact of features on output of XGBoost

In section 7 we did a feature importance analysis for our best performing model, which is the XGBoost and examined the SHAP summary plot. In Figure 16 the average impact of features on the models output is also illustrated and looks as follows:

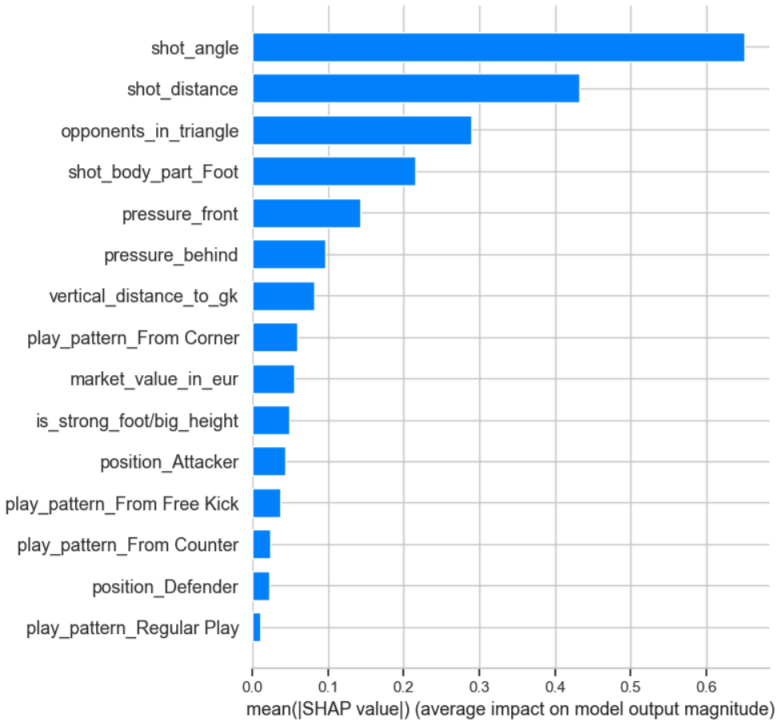

Figure 16: Average Impact of Features on Output of XGBoost

A.5 Expected Goals vs. players with most goals

Based on our conducted player performances analysis in section 8, where we focused on players who participated in at least 4 games of the tournament, with a minimum average of 30 minutes per match, these are the top 10 players with the biggest amount of goals:

player	minutes	xG	goals
Kylian Mbappé	598	7.3	8
Lionel Messi	690	7.5	7
Olivier Giroud	419	3.5	4
Julián Álvarez	464	2.7	4
Álvaro Morata	185	2.0	3
Cody Gakpo	450	1.6	3
Bukayo Saka	288	1.4	3
Gonçalo Ramos	152	2.2	3
Richarlison	323	2.3	3
Robert Lewandowski	360	3.0	2

Table 7: Expected Goals vs. Players with most Shots

A.6 Expected Goals vs. players with most shots

In contrast to the players with the most goals, these are the top 10 players with the biggest amount of shots:

player	minutes	xG	shots
Kylian Mbappé	598	7.3	29
Lionel Messi	690	7.5	27
Olivier Giroud	419	3.5	17
Lautaro Martínez	241	1.8	14
Youssef En-Nesyri	542	1.7	11
Julián Álvarez	464	2.7	11
João Félix	331	1.2	11
Cristiano Ronaldo	290	1.4	10
Marco Asensio	229	1.1	10
Robert Lewandowski	360	3.0	9

Table 8: Expected Goals vs. Players with most Shots

References

[1] Raffaele Poli, Loïc Ravenel, and Roger Besson. Inflation in the football players' transfer market (2013/14-2022/23). *CIES Football Observatory*, February 2023.

[2] Saudi Pro League transfer spending ranks second in world | Premier League smashes new records. Sky Sports, September 2023.

[3] Marc Brechot and Raphael Flepp. Dealing With Randomness in Match Outcomes: How to Rethink Performance Evaluation in European Club Football Using Expected Goals. *Journal of Sports Economics*, January 2020. URL https://doi.org/10.1177/1527002519897962.

[4] Jonny Whitmore. What Is Expected Goals (xG)? The Analyst, August 2023. URL https://theanalyst.com/eu/2023/08/what-is-expected-goals-xg/.

[5] What is xG? How is it calculated? StatsBomb Inc. URL https://statsbomb.com/soccer-metrics/expected-goals-xg-explained/. Accessed: January 3, 2024.

[6] Was sind eigentlich Expected Goals? Eine Erklärung des xGoals-Modells. DFL GmbH, July 2021. URL https://www.bundesliga.com/de/bundesliga/news/expected-goals-xgoals-fussball-analyse-statistik-3760/.

[7] StatsBomb: Football data, 2022. URL https://github.com/statsbomb/open-data. Accessed: 2024-01-01.

[8] David Sumpter and Aleksander Andrzejewski. Introducing expected goals. Soccermatics, 2022. URL https://soccermatics.readthedocs.io/en/latest/lesson2/introducingExpectedGoals.html.

[9] James Mead, Anthony O'Hare, and Paul McMenemy. Expected goals in football: Improving model performance and demonstrating value. *Public Library of Science San Francisco, CA USA*, April 2023. URL https://doi.org/10.1371/journal.pone.0282295.

[10] David Sumpter. Should you write about real goals or expected Goals? Medium, November 2017. URL https://soccermatics.medium.com/should-you-write-about-real-goals-or-expected-goals-a-guide-for-journalists-2cf0c7ec6bb6.

[11] Michael Caley. What is the best method of predicting goals? Putting xG to the test, February 2017. URL https://cartilagefreecaptain.sbnation.com/2014/2/28/5452786/shot-matrix-tottenham-hotspur-stats-analysis-expected-goals.

[12] Lars van Hoeve. How Geoinformation Enhances Professional Football. *GIM-International*, November 2017. URL https://www.gim-international.com/content/article/geovisual-football-analytics.

[13] Gabriel Anzer and Pascal Bauer. A goal scoring probability model for shots based on synchronized positional and event data in football (soccer). *Frontiers in Sports and Active Living*, page 53, March 2021.

[14] Richard Pollard and Charles Reep. Measuring the effectiveness of playing strategies at soccer. *Journal of the Royal Statistical Society Series D: The Statistician*, April 1997.

[15] James H. Hewitt and Oktay Karakuş. A machine learning approach for player and position adjusted expected goals in football (soccer). *Franklin Open*, September 2023. URL https://www.sciencedirect.com/science/article/pii/S2773186323000282.

[16] Alex Rathke. An examination of expected goals and shot efficiency in soccer. *Journal of Human Sport and Exercise*, January 2017.

[17] Izzatul Umami, Deden Hardan Gautama, and Heliza Rahmania Hatta. Implementing the Expected Goal (xG) model to predict scores in soccer matches. *International Journal of Informatics and Information Systems*, March 2021.

[18] Harm Eggels, Ruud Van Elk, and Mykola Pechenizkiy. Explaining Soccer Match Outcomes with Goal Scoring Opportunities Predictive Analytics. *Eindhoven University of Technology*, 2016.

[19] Patrick Lucey, Alina Bialkowski, Mathew Monfort, Peter Carr, and I. Matthews. "Quality vs Quantity": Improved Shot Prediction in Soccer using Strategic Features from Spatiotemporal Data. In *MIT Sloan Sports Analytics Conference.*, 2015. URL https://api.semanticscholar.org/CorpusID:110900298.

[20] Luca Pappalardo, Paolo Cintia, Paolo Ferragina, Emanuele Massucco, Dino Pedreschi, and Fosca Giannotti. PlayeRank: data-driven performance evaluation and player ranking in soccer via a machine learning approach. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2019. URL https://doi.org/10.1145/3343172.

[21] David Sumpter. The Geometry of Shooting. Medium, January 2017. URL https://soccermatics.medium.com/the-geometry-of-shooting-ae7a67fdf760.

[22] David Sumpter and Aleksander Andrzejewski. Expected Goals including player positions, 2017. URL https://soccermatics.readthedocs.io/en/latest/gallery/lesson7/plot_xG_tracking.html.

[23] Football Data from Transfermarkt. Kaggle. URL https://www.kaggle.com/datasets/davidcariboo/player-scores/data. Update frequency: Weekly. Accessed: 2024-01-10.

[24] Derrick Yam. Closing Down: How Defensive Pressure Impacts Shots. Statsbomb Inc., September 2018. URL https://statsbomb.com/articles/soccer/closing-down-how-defensive-pressure-impacts-shots/.

[25] James Gareth, Witten Daniela, Hastie Trevor, and Tibshirani Robert. *An introduction to statistical learning: with applications in R*. Springer, 2013.

[26] Dishant Kharkar. About Boosting and Gradient Boosting Algorithm... . Medium, July 2023. URL https://medium.com/@dishantkharkar9/about-boosting-and-gradient-boosting-algorithm-98dd4081ec18.

[27] Hannes Hapke and Catherine Nelson. *Building machine learning pipelines*. O'Reilly Media, 2020.

[28] Arindam Banerjee. Hyperparameter Tuning Using Randomized Search, November 2022. URL https://www.analyticsvidhya.com/blog/2022/11/hyperparameter-tuning-using-randomized-search/.

[29] Gaurav Dembla. Intuition behind Log-loss score. Medium, November 2020. URL https://towardsdatascience.com/intuition-behind-log-loss-score-4e0c9979680a.

[30] Rahul Agarwal and Brennan Whitfield. ROC Curves and AUC: The Ultimate Guide. Built In, March 2024. URL https://builtin.com/data-science/roc-curves-auc.

[31] Christoph Molnar. *Interpretable machine learning*. Lulu. com, 2020.

[32] Abid Ali Awan. Introduction to SHAP Values: Machine Learning Interpretability. DataCamp, June 2023. URL https://www.datacamp.com/tutorial/introduction-to-shap-values-machine-learning-interpretability.

[33] FIFA WC 2022 Player Data. Kaggle. URL https://www.kaggle.com/datasets/swaptr/fifa-world-cup-2022-player-data?select=player_stats.csv. Accessed: 2024-02-05.

[34] 53 new World Cup market values: Mbappé back to the top - Saka joins €100m club. Transfermarkt, December 2022. URL https://www.transfermarkt.com/53-new-world-cup-market-values-mbappe-back-to-the-top-saka-joins-euro-100m-club/view/news/415282.

List of Figures

List of Tables