


Article

Beyond xG : A Dual Prediction Model for Analyzing Player Performance Through Expected and Actual Goals in European Soccer Leagues

Davronbek Malikov ¹  and Jaeho Kim ^{2,*} 

¹ Department of AI Convergence Engineering, Gyeongsang National University (GNU), Jinjudaero 501, Jinjusi 52828, Republic of Korea; davronbekmalikov96@gmail.com

² Department of AI Convergence Engineering & Department of Software Engineering, Gyeongsang National University (GNU), Jinjudaero 501, Jinjusi 52828, Republic of Korea

* Correspondence: jaeho.kim@gnu.ac.kr

Abstract: Soccer is evolving into a science rather than just a sport, driven by intense competition between professional teams. This transformation requires efforts beyond physical training, including strategic planning, data analysis, and advanced metrics. Coaches and teams increasingly use sophisticated methods and data-driven insights to enhance decision-making. Analyzing team performance is crucial to prepare players and coaches, enabling targeted training and strategic adjustments. Expected goals (xG) analysis plays a key role in assessing team and individual player performance, providing nuanced insights into on-field actions and opportunities. This approach allows coaches to optimize tactics and lineup choices beyond traditional scorelines. However, relying solely on xG might not provide a full picture of player performance, as a higher xG does not always translate into more goals due to the intricacies and variabilities of in-game situations. This paper seeks to refine performance assessments by incorporating predictions for both expected goals (xG) and actual goals (aG). Using this new model, we consider a wider variety of factors to provide a more comprehensive evaluation of players and teams. Another major focus of our study is to present a method for selecting and categorizing players based on their predicted xG and aG performance. Additionally, this paper discusses expected goals and actual goals for each individual game; consequently, we use expected goals per game (xGg) and actual goals per game (aGg) to reflect them. Moreover, we employ regression machine learning models, particularly ridge regression, which demonstrates strong performance in forecasting xGg and aGg , outperforming other models in our comparative assessment. Ridge regression's ability to handle overlapping and correlated variables makes it an ideal choice for our analysis. This approach improves prediction accuracy and provides actionable insights for coaches and analysts to optimize team performance. By using constructed features from various methods in the dataset, we improve our model's performance by as much as 12%. These features offer a more detailed understanding of player performance in specific leagues and roles, improving the model's accuracy from 83% to nearly 95%, as indicated by the R-squared metric. Furthermore, our research introduces a player selection methodology based on their predicted xG and aG , as determined by our proposed model. According to our model's classification, we categorize top players into two groups: efficient scorers and consistent performers. These precise forecasts can guide strategic decisions, player selection, and training approaches, ultimately enhancing team performance and success.

Keywords: machine learning; ridge regression; soccer analytics; expected and actual goals; European soccer leagues



Citation: Malikov, D.; Kim, J. Beyond xG : A Dual Prediction Model for Analyzing Player Performance Through Expected and Actual Goals in European Soccer Leagues. *Appl. Sci.* **2024**, *14*, 10390. <https://doi.org/10.3390/app142210390>

Academic Editor: Luigi Portinale

Received: 2 October 2024

Revised: 6 November 2024

Accepted: 8 November 2024

Published: 12 November 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Soccer, or football as it is known in many countries, is one of the most popular and financially lucrative sports in the world [1,2]. For instance, the broadcast and commercial revenues to clubs in the Premier League for the 2022/23 season alone reached nearly GBP

3 billion [3]. This substantial financial investment and the intense competition among clubs necessitate a focus on improving team performance through innovative methods. As the sport continues to grow globally, the need for advanced strategies and technologies to support team performance and player development also increases.

Recent advances in sports analytics have led to the development of various novel metrics that are transforming the way clubs approach team management and player performance. These metrics, such as expected goals (xG), have become integral to clubs' decision-making processes [4]. However, a comprehensive approach to player and team assessment is still lacking, as existing metrics may not provide a complete view of a player's contributions or a team's effectiveness. This gap exists partly due to the absence of crucial features and the inherent limitations of xG [5]. It is crucial for analysts to recognize the significance of context. Relying solely on one specific metric for statistical analysis, especially when assessing games individually, can be detrimental. While those utilizing xG may appreciate its advantages, it is equally important to acknowledge its limitations. Although the data used in any model play a critical role, how that data are interpreted is equally vital. A comprehensive understanding that encompasses both the strengths and weaknesses of xG is necessary for its fullest potential [6].

In addition to expected goals (xG), actual goals (aG) provide a crucial measure of performance, as they reflect the outcomes of goal-scoring opportunities. By incorporating both xG and aG , we gain a clearer and more comprehensive understanding of a player's effectiveness. In this work, we propose a hybrid approach that predicts both xG and aG to address the limitations of relying solely on xG , offering a more realistic evaluation of player performance. For a more detailed assessment, this paper considers expected goals (xG) and actual goals (aG) on a per-game basis. Therefore, we use the terms expected goals per game (xGg) and actual goals per game (aGg) to represent these metrics. xGg estimates the number of goals a player or team is expected to score in each game based on the quality of chances created, while aGg reflects the actual number of goals scored in a game. This distinction helps provide a clearer evaluation of performance and efficiency. This approach involves constructing models that integrate significant features to improve the accuracy of player and team performance predictions. By leveraging these models, we can offer insights into players' goal-scoring opportunities and team strategies. Additionally, our study explores and compares xGg across six prominent European football leagues: the English Premier League (EPL), German Bundesliga, Spanish La Liga, Italian Serie A, French Ligue 1, and Russian Premier League (RPL). Through our analysis, we identify the top goal scorers from each league and evaluate their performance in relation to the proposed models.

The main contributions of this paper are as follows:

- It proposes a hybrid approach that combines expected goals per game (xGg) and actual goals per game (aGg) to enhance player performance prediction.
- It introduces a new method for feature construction in player performance modeling.
- It compares various machine learning models to evaluate their effectiveness in predicting xGg and aGg , identifying the best algorithms for player performance analysis.
- It categorizes players based on xGg and aGg to provide more detailed insights into player performance and effectiveness.

The remainder of the paper is organized as follows: Section 2 reviews the background and literature, while Section 3 covers data preparation, including data sources, tools, overview, description, feature engineering, and statistical analysis. Section 4 discusses modeling, including model introduction, comparative evaluation, and validation. Section 5 outlines the research findings and results obtained from the analysis, whereas Section 6 offers a discussion and limitations. The conclusion highlights the principal insights and implications drawn from the research.

2. Background and Literature Review

2.1. Literature Review

Data analytics has become a key component in soccer, with most team coaches relying on assistants to help prepare players for games and gather intelligence on opponents. Advanced data modeling techniques are now widely used by professional soccer teams to address challenges such as team selection, personalized training, tactical analysis, and player recruitment. While the use of these advanced techniques is relatively recent, statistical analysis in soccer dates back to the 1950s. Charles Reep, the first soccer analyst to use a notational system to record every event in a match, found that a sequence of three or four consecutive passes significantly increased the chances of scoring a goal [7,8].

Despite soccer's widespread popularity, there has historically been little apparent connection between the sport and statistical data analysis [1,2]. However, the Expected Goals (xG) metric bridges this gap by employing statistical methods to model the probability of a shot resulting in a goal [9]. Since Sam Green's groundbreaking article introducing xG , this metric has become one of the most prevalent and insightful tools in soccer analytics [10]. Numerous studies have explored the application of xG models by incorporating different features, primarily focusing on two key variables: the distance from the goal and the angle of the shot [11–13]. For instance, Rathke applied these variables to data from the 2012/13 Premier League and Bundesliga seasons [14]. His approach involved dividing the football pitch into eight zones and analyzing the probability of scoring from shots taken in each zone. Rathke's findings demonstrated that both distance and angle significantly affect the likelihood of scoring a goal. Similarly, Spearman examined the impact of distance and angle on shooting outcomes [15]. His model, based on event data from a 14-team professional football league during the 2017/18 season, introduced a probabilistic approach to measure off-ball scoring opportunities (OBSSO). This nuanced approach highlights the complexity and potential of xG modeling in soccer analytics. Another important aspect discussed in the literature is the shot type, which provides contextual information about the shot [16–18]. Shot types can be divided into two main categories: the part of the body used to take the shot (e.g., left/right foot or head) and the game situation when the shot occurred. Brechot and Flepp incorporated these features into their model and found that both aspects influence shot outcomes [19]. They observed that shots from free kicks and penalty kicks are more likely to result in goals compared to shots taken in open play with either foot. Conversely, shots from headers tend to be less likely to result in goals.

Selecting the right features is crucial for implementing xG models, as it plays a pivotal role in both past and future research. In addition to shot type and shooting position, other key features identified in previous studies include player ability and home advantage [16,17,20,21]. However, not all earlier studies encompass all the essential features for accurately predicting xG and assessing soccer players' and teams' performance. In our study, we aim to enhance the overall performance of the model by proposing constructed features. These additions seek to provide a more comprehensive and nuanced understanding of the factors that influence xG and overall soccer performance.

Furthermore, most previous studies have utilized xG metrics to predict the performance of individual players and soccer teams. However, relying solely on xG may not provide a comprehensive assessment of player or team performance, as even teams with high xG metrics may not always achieve the desired results. To illustrate the additional insights provided by xG , consider Manchester United's 0-0 draw with Watford in February 2022 [21]. The scoreline alone might suggest that Watford defended well against a stronger team and that Manchester United struggled against a weaker opponent. While the draw was a favorable outcome for Watford, it did not fully reflect the match's dynamics. Manchester United generated 2.9 xG , whereas Watford managed just 0.5. Watford's good fortune allowed them to maintain a clean sheet, but their performance was lacking overall. Conversely, Manchester United's missed opportunities led to a disappointing result, though their performance indicated potential for future success.

In our paper, we propose new methods for predicting player performance by combining expected goals per game (xG) and actual goals per game (aG) in machine learning models across several top European soccer leagues. This approach seeks to offer a more detailed and precise evaluation of player and team performance. Moreover, we consider both expected goals and actual goals as indicators of player performance, with both being derived by game-specific contexts. Expected goals are influenced by various situational factors that help indicate the likelihood of scoring, while actual goals serve as quantifiable outcomes within these contexts. A more detailed discussion of these factors can be found in the subsequent subsections.

2.2. Understanding Expected Goals (xG)

Soccer has undergone a significant transformation in recent years, with an increasing emphasis on tactical battles between coaches. To navigate these strategic complexities, metrics that evaluate the performance of teams and players and assess their contributions have become essential. One of the most prominent metrics in modern soccer is xG , which plays a pivotal role in tactical discussions.

Moreover, xG is a statistical metric that provides insight into the quality of scoring opportunities for both individual players and teams. This measurement evaluates how likely a player or team is to score based on the quality of their chances. As a vital tool in analyzing player efficiency and team strategies, xG offers a deeper understanding of the game beyond traditional goal-scoring metrics [22]. Many data analytics companies use their own approaches to calculating xG , often considering factors such as the distance to the goal post, angle of the shot, body part used to take the shot, and the type of assist or preceding action, all based on historical player data [23]. Furthermore xG assigns a value to each shot, ranging from 0.00 to 1.00, to reflect the probability of it resulting in a goal [22]. For instance, a shot with an xG of 0.01 suggests a 1% probability of scoring, indicating that a goal would be expected once in every 100 attempts—a low-chance scenario. Conversely, a shot with an xG of 0.99 implies it would be scored 99 times out of 100 by an average player, representing an almost certain goal. The xG value can vary significantly based on the position and angle of the shot. For example, there is a large disparity in the xG value between shots taken from inside the penalty box versus those from outside the box. Shots from within the box generally have a higher xG due to their proximity to the goal and better angles. Using xG , these situations can be quantified numerically. For example, if a shooting opportunity is assigned an xG of 0.1, it indicates that a player is expected to score one goal from every ten shots in that specific scenario. If a player has 15 shooting chances from inside the box in a single match, their expected goal total would be 15 times 0.1, resulting in 1.5 xG . This means the player's expected goal value can fluctuate depending on the quality and frequency of the chances they receive [24]. However, relying solely on xG does not provide a complete picture of team and player contributions. When analyzing a single match, xG can sometimes present a skewed interpretation of team performance. For example, if Team A accumulates 3 xG while team B records only 1 xG , it might appear that Team A dominated the game. However, the reality could be different; Team B might have scored three early goals and then chosen to defend their lead, allowing Team A more opportunities to attack. In such a case, Team A may have generated better chances, but Team B's early success would have changed the dynamics of the game. This illustrates the potential discrepancy between what xG suggests and the actual course of events [25].

2.3. Understanding Actual Goals (aG)

Actual goals (aG) is another distinct metric employed to evaluate the performance of players and teams. The aG metric refers to the actual number of goals scored by a player or team in a match. It provides a straightforward measure of a player's or team's scoring efficiency and reflects the real outcomes of their offensive efforts. Comparing xG with aG can offer key insights into performance. For example, if a player has a high xG value but a low aG , it may suggest missed opportunities or inefficient finishing. Conversely, if a player

consistently exceeds their xG , it could indicate superior finishing skills or effective use of limited chances. Similarly, when evaluating team performance, the relationship between xG and aG can shed light on a team's ability to capitalize on scoring opportunities [26].

Table 1 provides a detailed comparison of actual goals (aG) and expected goals (xG) for each team in the Premier League during the 2022/2023 season [27]. The Difference column quantifies the discrepancy between aG and xG with teams ranked in descending order based on this difference. A higher difference indicates a larger deviation between performance and expected outcomes. For instance, Arsenal and Manchester City exhibit the highest positive differences, demonstrating that they scored notably more goals than anticipated.

Table 1. Clubs ranked by the difference between aG and xG in the Premier League for the 2022/2023 season.

Team	aG	xG	Difference
Arsenal	88	73.33	14.67
Manchester City	94	80.47	13.53
Spurs	70	57.81	12.19
Chelsea	38	50.08	−12.08
Everton	34	45.78	−11.78
Man Utd	58	68.74	−10.74
West Ham	42	50.77	−8.77
Fulham	55	47.03	7.97
Wolves	31	37.84	−6.84
Newcastle	68	73.48	−5.48
Brighton	72	74.75	−2.75
Southampton	36	38.41	−2.41
Bournemouth	37	39.46	−2.46
Nottingham Forest	38	39.78	−1.78
Liverpool	75	73.74	1.26
Leeds	48	48.57	−0.57
Leicester	51	51.32	−0.32
Brentford	58	57.89	0.11
Crystal Palace	40	39.95	0.05
Aston Villa	51	50.99	0.01

In contrast, Chelsea and Everton have the largest negative differences, implying that their goal-scoring performance is below expectations due to missed opportunities.

2.4. Defining the Prediction Task

In our study, we predict player performance for the upcoming season using key metrics such as xGg and aGg , with these predictions based on historical data from the 2014/2015 to 2021/2022 seasons. Moreover, the input variables reflect each player's past performance metrics, capturing their historical contributions across multiple seasons, while the target variables are designed to project future performance outcomes. This approach clarifies that the model's input features are derived from previous seasons, whereas the target variables represent the expected results in the forthcoming season. This feature structuring ensures that the model is trained to accurately forecast future performance based on patterns observed in historical data. While predicting performance for an entire career or all past seasons is less practical due to variability in player form, focusing on next-season predictions offers actionable insights. To ensure realistic decision-making conditions, our experiment design simulates real-world scenarios by using only the data available before the predicted season. We also analyze player efficiency by comparing the difference between xGg (chances created per game) and aGg (goals scored per game), which helps assess how effectively players convert chances. We categorize players as either efficient scorers or consistent performers based on their ability to capitalize on opportunities or

maintain steady performance levels. These predictions offer practical value for coaches and teams. For example, when preparing for a match against a strong opponent where scoring chances might be scarce, coaches can rely on efficient scorers who excel at converting lower xG opportunities into goals. This strategic selection helps ensure the team is better prepared for challenging matches. Additionally, these predictions assist teams in evaluating whether a player's future contributions align with their tactical and strategic goals, guiding decisions on recruitment, transfers, and contract renewals.

3. Handling Our Dataset

In this section, we provide a detailed overview of the dataset used for building our model, along with the tools and techniques applied throughout our study. The following Figure 1 outlines the sequence of this section, where we first detail the data sources and tools applied in our research.

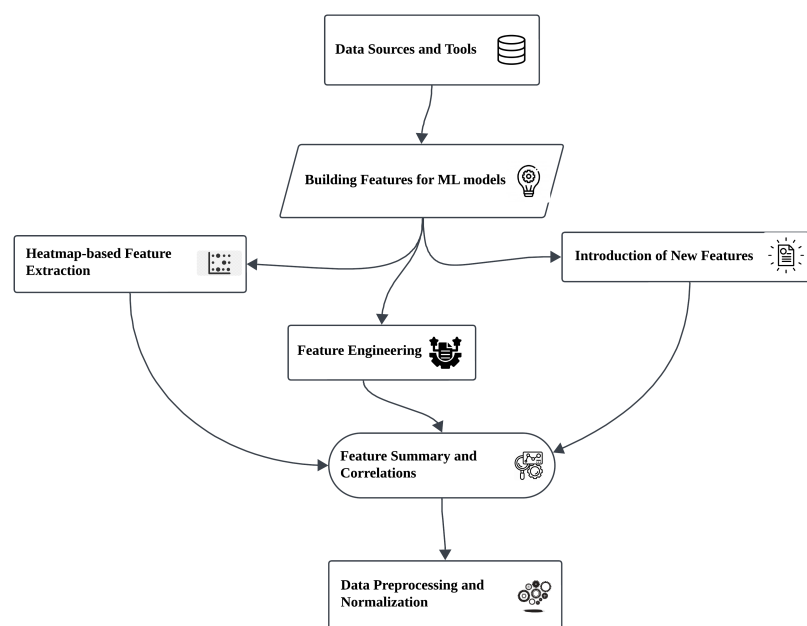


Figure 1. Data handling process overview.

Then, we thoroughly describe the comprehensive methods employed for constructing features, including the use of heatmaps for extracting key features and various feature engineering techniques to create new variables. We also offer an in-depth explanation of the newly introduced features, discussing their relevance and significance in the context of our model. Additionally, this section includes a summary of the features, highlighting their importance, and examines the correlations among them to enhance the model's predictive performance. Moreover, we outline the data preprocessing steps and normalization techniques used to prepare the dataset for modeling. These insights help inform the decisions made in the subsequent stages of our analysis and modeling.

3.1. Data Sources and Tools

Data serves as the foundation for any machine learning project, playing a critical role in the accuracy and effectiveness of analysis and predictions. The success of research outcomes is highly dependent on both the quantity and quality of the data used. In this study, we utilized a dataset from Understat, a well-known professional soccer website, with data gathered by analyst Edd Webster, who has publicly shared it on his GitHub repository [28,29]. Our dataset includes match events from the Top 5 European leagues and the Russian Premier League, covering the 2014/2015 to 2021/2022 seasons. It comprises 21,678 observations, offering a detailed perspective on player and team performance over multiple years. As a result of this time frame, individual players can appear multiple times,

leading to a total of 6359 unique players in the dataset. Our dataset consists of 29 features related to individual players, reflecting their performances in matches, constructed using three distinct methods. Initially, we extracted 14 core features from Understat, which included well-established soccer metrics such as goals, assists, and expected goals (xG). Moreover, we developed another 13 features through feature engineering. These metrics are fundamental in evaluating player performance and are widely recognized in soccer analytics [10,23,29]. In addition to the 27 features derived from extraction and engineering, we introduced two new features called *position_weight* and *league_weight* to further refine model accuracy. These new additions were carefully selected to offer additional insights and enhance the overall predictive power of our models. A detailed explanation of the feature construction process is provided in the following sections.

Furthermore, for this research, we primarily use Python (version 3.11.7) due to its user-friendly nature and extensive range of libraries. We leverage several Python libraries, including scikit-learn (version 1.4.2), which provides a wide array of pre-built algorithms and tools for data preprocessing, model training, and evaluation [30]. Additionally, we use Pandas (version 2.1.4) for efficient data manipulation and analysis [31], while NumPy (version 1.26.4) serves as the foundation for numerical computing, supporting multi-dimensional arrays, matrices, and a variety of mathematical functions [32]. For visualization, we utilize Seaborn (version 0.12.2), an advanced visualization library built on top of Matplotlib, which simplifies the creation of informative and aesthetically pleasing statistical visuals such as heatmaps and box plots [33]. Matplotlib (version 3.7.5) itself offers a diverse range of chart and plot options, enabling detailed customization and seamless integration with other libraries for effective data visualization [34]. By combining these tools and libraries, we efficiently process, analyze, and visualize data, thereby enhancing the overall quality and depth of our research findings.

3.2. Building Features for ML models

In this subsection, we present three distinct methods for constructing features to build our machine learning models. To facilitate a clearer understanding of these methods, we provide accompanying tables and graph that illustrate our approaches and highlight the significance of the features constructed through each method.

3.2.1. Heatmap-Based Feature Extraction

To extract features from Understat [29] we utilize a systematic approach that includes leveraging heatmaps to identify additional correlated features. The heatmap shown in Figure 2 visually represents the correlations between the various features selected from our dataset. This visualization helps in understanding the strength and direction of relationships among features, with color intensity indicating the degree of correlation. We set a predefined correlation threshold, typically ± 0.5 , to guide our selection process. Features with coefficients greater than 0.5 were considered to have a significant positive correlation, while those less than -0.5 indicated a significant negative correlation. By applying these criteria, we ensure that only features with strong relationships to the target variable are included, namely xGg and aGg , thereby enhancing our model's predictive power. The selected features include fundamental metrics such as *games* and *goals* which are crucial for assessing player productivity and scoring efficiency. Moreover, xG and *assists* were included for their roles in quantifying scoring opportunities created and converted. Metrics like $xGChain$ and $xGBuildup$ were also chosen to measure a player's involvement in goal-scoring sequences and play-building activities. Furthermore, the player ID serves as a unique identifier within the dataset. Generally, it is not regarded as an informative feature since it does not provide statistical insights into performance or characteristics.

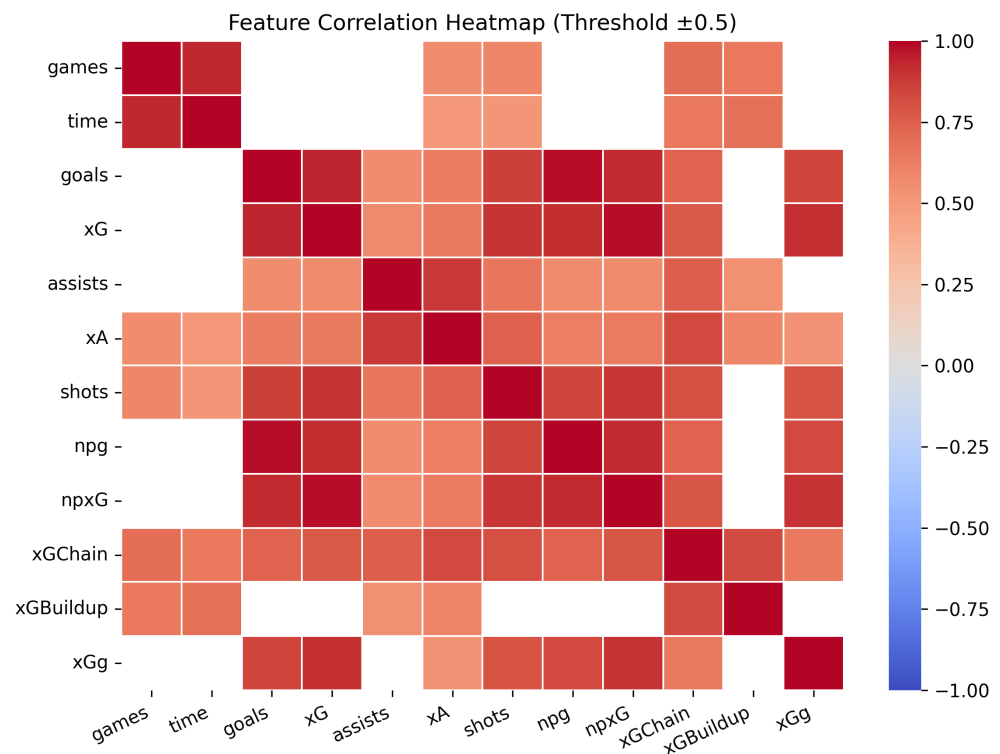


Figure 2. Feature correlation heatmap.

These features were carefully selected based on their logical relevance and statistical significance, aiming to improve both the accuracy and robustness of our predictions. This approach facilitates a deeper understanding of player performance across various soccer leagues, reflecting a nuanced interpretation of the underlying data dynamics.

3.2.2. Feature Engineering

In addition to feature extraction, we implemented feature engineering to introduce 13 additional features categorized by their relevance to player performance: scoring efficiency, playmaking abilities, disciplinary behavior, and advanced metrics. Feature engineering enhances the dataset by creating new features that provide deeper insights into player performance. This process involves transforming raw data into meaningful inputs that improve model performance, predictive accuracy, and the overall understanding of player dynamics. For instance, metrics like *aGg* (actual goals per game) help assess a player's scoring efficiency by indicating the average number of goals scored per game, offering insights into offensive productivity and identifying top-performing players. Another engineered feature, *gpm* (goals per minute), reveals the rate at which goals are scored during game time. This aids in assessing a player's impact throughout the match and enhances the precision of goal-scoring predictions. Moreover, *apg* (assists per game) sheds light on a player's playmaking abilities by showing the average number of assists provided per game. This metric not only highlights individual performance but also contributes to understanding collaborative efforts within the team, thereby enriching insights into team dynamics. In terms of disciplinary behavior, features like *ypg* (yellow cards per game) and *rpg* (red cards per game) provide insights into a player's disciplinary record and aggression level. Monitoring these metrics helps assess their impact on match outcomes and team dynamics, considering potential suspensions or player availability issues. Advanced metrics like *xGdiff* (expected goals difference) further enrich the analysis by evaluating a player's goal-scoring potential relative to the quality of scoring opportunities.

Additionally, Table 2 offers a detailed summary of all features, including those obtained through different methods discussed in the subsequent subsections. This table presents both descriptions of the features and their correlation coefficients with the target variables, as detailed in the analysis.

Table 2. Details of extracted and engineered features of the Understat data. Features 1–14 are extracted using heatmap; features 15–27 are engineered; features 28–29 represent newly introduced features. Input denotes data from past seasons, while input and target indicate metrics that incorporate both past performance and predictive elements.

No	Field	Description	Correlation	Category
1	games	Total games played by the player.	0.55	Input
2	timesplayed	Total minutes played by the player.	0.58	Input
3	goals	Goals scored by the player.	0.85	Input
4	xG	Total expected goals for a season.	0.91	Input
5	assists	Total assists provided by the player.	0.55	Input
6	xA	Expected assists for a season.	0.54	Input
7	shots	Total shots taken by the player.	0.80	Input
8	keypasses	Key passes leading to scoring chances.	0.55	Input
9	yellowcards	Total yellow cards received.	0.52	Input
10	redcards	Total red cards received.	0.51	Input
11	npg	Non-penalty goals scored.	0.83	Input
12	npxG	Non-penalty expected goals.	0.90	Input
13	xGChain	Total xG from every possession involvement.	0.65	Input
14	xGBuildup	Total xG from play buildup, excluding shots.	0.53	Input
15	aGg	Actual goals per game.	0.88	Input and Target
16	gpm	Goals per minute.	0.54	Input
17	apg	Assists per game.	0.55	Input
18	shpg	Shots per game.	0.84	Input
19	shpm	Shots per minute.	0.53	Input
20	kppg	Key passes per game.	0.55	Input
21	kppm	Key passes per minute.	0.52	Input
22	ypg	Yellow cards per game.	0.51	Input
23	ypm	Yellow cards per minute.	0.53	Input
24	rpg	Red cards per game.	0.52	Input
25	rpm	Red cards per minute.	0.51	Input
26	xGdiff	Difference between goals and xG.	0.55	Input
27	xGg	xG per game.	1.00	Input and Target
28	positionweight	Weight based on player's position.	0.55	Input
29	leagueweight	Weight based on the player's league.	0.55	Input

3.2.3. Introduction of New Features

To complement the features obtained through heatmaps and feature engineering, we introduce new features, such as *league_weight*, to further improve the performance of our model. This feature enhances our dataset by capturing the variability in player performance across leagues, allowing the model to make informed predictions within a broader competitive context. Moreover, the *league_weight* feature is used to differentiate between leagues, which is essential for avoiding bias. It is important to account for varying levels of competition, quality, and popularity across leagues. Each league has a unique history of success in international tournaments like the UEFA Champions League, influencing its competitive standard [35].

Furthermore, the quality of leagues varies based on player skill levels and team performance, necessitating league-specific adjustments for more accurate insights. These adjustments are essential due to factors such as differing histories of success in international competitions like the UEFA Champions League and varying overall quality levels based on the caliber of players and teams [35]. Additionally, fan engagement and attendance rates

can vary widely from league to league [36]. By applying different ratios to each league, we can adjust for these discrepancies, accurately reflecting each league's actual strength and unique characteristics. This differentiation enables fairer comparisons between players and teams, allowing our models to accommodate variations in league quality and other factors, ultimately leading to more precise and reliable soccer.

The formula for calculating *league_weight* is as follows:

$$league_weight_i = w_{hist} \times hist_score_i + w_{fan} \times fan_score_i, \quad (1)$$

where the variable *i* stands for each league. The weight w_{hist} is assigned to historical information, while w_{fan} is assigned to fan attendance. Moreover, *hist_score* represents the normalized historical score for each league, and *fan_score* is the normalized fan attendance score for each league.

According to our formula, *league_weight* is calculated using two main factors: historical success and fan attendance. By including these two factors, we make sure the analysis considers both the competitive strength and the popularity of each league. A higher historical score shows strong international performance and a higher standard of competition, while a higher fan attendance score reflects more commercial appeal, financial resources, and the ability to attract top players. Each league's score can go up to a maximum of 10 points, with a perfect score indicating the highest levels of fan support and historical success. Using these weights helps us account for differences between leagues, allowing a more accurate representation of each league's true strength and unique features. This approach ensures fairer comparisons between players and teams from different leagues by considering differences in league quality and other key factors. Overall, this method provides more reliable and precise soccer analysis, ensuring our models give valid insights across various levels of competition. In addition, we introduce another feature called *position_weight*. Player position significantly affects a player's *xG* and *aG* as well as the overall impact on the game. Forwards, who focus on scoring goals, usually have higher *xG* compared to defenders. Meanwhile, goalkeepers and defenders have specific duties; goalkeepers concentrate on making saves, while defenders aim to stop the opposition from scoring [37].

Our dataset includes players whose primary positions are goalkeepers, defenders, midfielders, and strikers. Additionally, over the years, certain players have transitioned between various positions on the field. To accurately capture these shifts and their impact, we developed a feature that categorizes players into specific positional groups. We then assign weight to each category based on the players' anticipated contributions to scoring goals and influencing game outcomes. This approach allows us to account for positional versatility and better understand how players' roles evolve and affect their overall performance and team dynamics. For example, we grouped players into categories such as midfielder and striker, defensive striker, and forward, midfielder, and striker, among others. Each group received a weight based on its influence on *xG*. Positions related to scoring goals, such as forward and striker, received higher ratios of 10, highlighting their key role in creating scoring chances. In contrast, positions with defensive or goalkeeping duties, such as defenders and goalkeepers received lower ratios of 4 and 0.1, respectively, reflecting their focus on preventing goals rather than scoring. Ratios for mixed roles, like defender, forward, and midfielder or forward and midfielder were adjusted to account for their diverse responsibilities across different aspects of the game. By incorporating *position_weight*, we adjust our analysis to account for the various responsibilities and expected outcomes of different player positions. This feature offers a more comprehensive understanding of player performance and potential across different roles on the field. It also enables more precise comparisons between players in similar positions, leading to deeper and more meaningful insights into the game.

3.3. Feature Summary and Correlations

In summary, this section presents a comprehensive overview of the features included in our dataset, which were collected using the three methods discussed above. Table 2 shows a comprehensive summary of all features included in the analysis, detailing both descriptions and their correlation coefficients represent the average of the two target variables, xGg and aGg .

The table highlights the various features and their correlations, demonstrating their influence on the model's predictions. Features 1 to 14 were collected using a heatmap, with their respective correlation values directly related to this method. For instance, the *goals* feature shows a strong correlation of 0.85, indicating its significant impact on evaluating player performance. Similarly, xG has a high correlation of 0.91, underscoring its essential role in outcome prediction. Other notable features from this group include *shots* with a correlation of 0.80, playing a crucial role in the model. Conversely, the remaining features, from 15 to 27, were developed through feature engineering, and their correlation values with the target variable are also included in the table. For example, *gpm* (goals per minute) and *shpm* (shots per minute) have correlations of 0.54 and 0.53, respectively, offering valuable context by capturing different aspects of gameplay. Additionally, *league_weight* and *position_weight* exhibit moderate correlations of 0.55 each. While these features may have lower individual importance, they enhance the model by providing supplementary insights that, when combined with other metrics, improve overall predictive accuracy. This table categorizes the model's variables into input and input and target features. Input features reflect a player's past performance, capturing historical data as foundational information. Input and target features, however, combine historical data with predictive relevance, directly contributing to the model's target outcomes. Metrics like aGg and xGg not only summarize past achievements but also enhance predictions, making them essential for performance evaluation and forecasting.

Moreover, Table 3 illustrates the feature importance for both xGg and aGg . This table provides a detailed comparison of how different features contribute to the predictions of xGg (expected goals per game) and aGg (actual goals per game). To determine feature importance, we employed SHAP (SHapley Additive exPlanations) values, which provide an in-depth understanding of how different features influence the model's predictions. In the context of xGg , the most influential features include aGg (goals per game) and *gpm* (goals per minute), which contribute 33.82% and 33.31% to the model, respectively. Other significant features are *shpg* (shots per game) with 6.43%, and *apg* (assists per game) with 4.91%. These features indicate that scoring efficiency and shooting metrics are critical for predicting expected goals per game. Conversely, for aGg , the most impactful feature is xGg (expected goals per game), which has a dominant importance of 60.61%. This suggests that the expected goals metric is highly predictive of the actual goals scored. Other important features for aGg include *gpm* (goals per minute) with 10.90%, and *shpm* (shots per minute) with 10.46%. These contributions highlight the relevance of shooting metrics and goals efficiency in predicting actual goals per game. While some features, such as $xGChain$ and $xGBuildup$ show relatively low importance percentages (below 1% for xGg and below 0.5% for aGg), they still play a role in the overall model. For instance, features like $xGChain$ and $xGBuildup$ capture aspects of play that, despite their smaller individual impact, can affect the prediction accuracy when combined with other features. Table 4 presents a comprehensive overview of player performance metrics, offering valuable insights into the intricate dynamics of soccer. The dataset reveals that players have participated in an average of 3214 games, indicative of their substantial careers and commitment to the sport. This longevity is a testament to the players' resilience and adaptability, essential qualities in the highly competitive environment of professional soccer.

Table 3. Combined feature importance for *xGg* and *aGg*.

Feature	Importance for <i>xGg</i> (%)	Importance for <i>aGg</i> (%)
<i>aGg</i>	33.82	-
<i>gpm</i>	33.31	10.90
<i>rpm</i>	6.72	2.43
<i>shpg</i>	6.43	0.23
<i>apg</i>	4.91	7.35
<i>shpm</i>	4.36	10.46
<i>kppm</i>	3.16	1.55
<i>xG</i>	1.63	0.68
<i>xGdiff</i>	1.49	1.60
<i>rpg</i>	0.96	1.41
<i>kppg</i>	0.87	0.52
<i>ypg</i>	0.73	0.22
<i>ypm</i>	0.64	0.55
<i>shots</i>	0.24	0.01
<i>assists</i>	0.15	0.27
<i>npG</i>	0.13	0.05
<i>goals</i>	0.13	0.91
<i>xGChain</i>	0.10	0.005
<i>xGBuildup</i>	0.08	0.02
<i>npg</i>	0.04	0.14
<i>key_passes</i>	0.04	0.02
<i>position_weight</i>	0.03	0.01
<i>xA</i>	0.01	0.01
<i>league_weight</i>	0.01	0.02
<i>games</i>	0.005	0.03
<i>time</i>	0.005	0.005
<i>xGg</i>	-	60.61

Table 4. Summary statistics of the dataset.

Feature	Mean	Standard Deviation	Range
Number of Games Played (<i>games</i>)	3213.74	2470.25	1–9559
Time Played (<i>time</i>)	18.76	11.13	1–38
Goals Scored (<i>goals</i>)	1317.34	966.07	1–3420
Expected Goals (<i>xG</i>)	1.75	3.40	0–48
Assists (<i>assists</i>)	1.80	3.11	0–39.31
Expected Assists (<i>xA</i>)	1.23	2.02	0–20
Shots (<i>shots</i>)	1.26	1.79	0–20.62
Key Passes (<i>key_passes</i>)	16.58	21.47	0–227
Yellow Cards (<i>yellow_cards</i>)	12.24	15.30	0–146
Shots per Game (<i>shpg</i>)	0.73	0.71	0–7
Expected goals per game (<i>xGg</i>)	0.08	0.11	0–1.12

In terms of on-field activity, players typically contribute around 18.76 min of impactful play per match. This metric underscores the significance of efficiency, as athletes must maximize their performance within the constraints of limited playing time. Notably, the dataset records an average of 1317 goals scored, which encapsulates the myriad moments of skill and achievement that characterize these athletes’ careers. Complementing this, the expected goals (*xG*) statistic, averaging 1.75, indicates players’ proficiency in finding scoring opportunities, highlighting their ability to create potential goal-scoring situations.

Teamwork is also crucial in soccer, as evidenced by the average of 1.80 assists per player. This statistic reflects the collaborative nature of the sport, showcasing players’ capabilities to facilitate scoring opportunities for their teammates. Furthermore, an impressive average of 16.58 key passes per player illustrates the role of playmakers who significantly influence the game’s outcome through strategic ball distribution and offensive orchestration. However,

the competitive nature of soccer presents its challenges. The data reveal an average of 12.24 yellow cards per player, suggesting the intense nature of matches where discipline can be as vital as scoring. The variability observed in metrics, including shots per game (0.73) and expected goals per game (0.08), encapsulates soccer's unpredictable essence, where individual talents often shine through the complexities of game scenarios

Moreover, Table 5 shows a sample from the dataset. The table summarizes compiled data without additional calculations.

Table 5. Final sample dataset including all features and target variables.

id	Games	Time	Goals	xG	Assists	xA	Shots	Key_PASSES	Yellow_Cards	shpg	xGg
619	33	2551	26	25.27	8	5.57	148	33	...	4.48	0.77
647	34	2589	21	17.16	4	3.92	112	27	...	3.29	0.50
802	26	2111	20	15.22	3	4.55	76	41	...	2.92	0.59
848	35	3078	18	17.88	5	2.55	131	23	...	3.74	0.51
498	35	2967	16	13.45	8	8.49	122	82	...	3.49	0.38

It includes various performance metrics for soccer players, such as a unique identifier for each player, the number of games played, total minutes on the field, goals scored, and assists provided. The dataset also features advanced statistics like expected goals (xG), which measures the quality of scoring chances, and expected assists (xA), which estimates the likelihood of a pass leading to a goal. Additional metrics include the total number of shots taken, key passes made (those that lead to a shot), yellow cards received, and averages for shots per game ($shpg$) and expected goals per game (xGg).

3.4. Data Preprocessing and Normalization

Before applying machine learning models, it is crucial to ensure that the dataset is properly prepared, as well-prepared data are essential for achieving accurate and reliable results. In our study, we implement several key preprocessing steps to align the data with the requirements of the algorithms. The key steps are outlined below:

- **Data cleaning:** We address missing values and carefully examine and adjust outliers where necessary to maintain the consistency and integrity of the dataset, preventing any anomalies from affecting the accuracy of model predictions.
- **Normalization/standardization:** We standardize all numerical features, transforming them to have a mean of 0 and a standard deviation of 1. Since many algorithms are sensitive to the scale of input data, this normalization process ensures that each feature contributes proportionately to the model, preventing features with larger ranges from dominating the learning process. As a result, the models perform more consistently and accurately, improving generalization and ensuring robust predictions across different types of algorithms.
- **Data splitting:** We divide the dataset into training and test sets using an 80/20 split. This allows the models to be trained on the majority of the data and evaluated on a separate test set to assess their generalization performance.

4. Comparison of ML Models

The objective of this section is to compare machine learning models and select the most suitable one for developing an effective and accurate solution for our study. To achieve this, we present a comparative evaluation of different machine learning models, which allows us to identify and select the optimal model for our purposes. This evaluation process ensures that the chosen model best meets the study's goals and provides robust and reliable results. Initially, we introduce the machine learning models with concise explanations of each approach. These descriptions highlight the strengths and unique aspects of each model, providing context for their application in our study. Following this, a comparative evaluation of the models guide our selection process, allowing us to identify the model that

best aligns with our objectives and yields the most reliable and robust results. Additionally, we check the correctness of our model using the K-fold cross-validation method to ensure its accuracy.

4.1. Model Introduction

In our study, we apply supervised machine learning models to generate predictions for continuous metrics like xGg and aGg using supervised regression models. This technique aligns with the characteristics of our target variables, which are continuous numerical values, making regression the best option for our analysis. We compare 10 regression-based machine learning models, as shown in Table 6. These models encompass a range of approaches, from traditional linear models to more complex tree-based and neural network models.

Table 6. Key parameters, implementations, and domain parameters for various machine learning models.

Model	Key Parameters	Description	Implementation (Scikit-Learn)	Domain Parameters
Linear Regression	None	A fundamental model that establishes a linear relationship between the input variables and the target variable.	<code>linear_model.LinearRegression</code>	None
Ridge regression	α (regularization strength)	A variant of linear regression that incorporates a regularization term to handle multicollinearity and improve model generalization.	<code>linear_model.Ridge</code>	$\alpha = 1.0$ (default), tune based on cross-validation
Lasso regression	α (regularization strength)	Similar to ridge regression, but uses L1 regularization, which can result in feature selection by penalizing certain features.	<code>linear_model.Lasso</code>	$\alpha = 0.1$ (default), tune based on feature importance
Elastic net	α (regularization strength), $l1_ratio$	Combines the L1 and L2 regularization from lasso and ridge, offering a balanced approach to feature selection and penalty.	<code>linear_model.ElasticNet</code>	$\alpha = 0.5$, $l1_ratio = 0.5$ for balanced regularization
Decision Tree	<code>max_depth</code> , <code>min_samples_split</code>	A tree-based model that splits the data based on specific conditions, offering intuitive, non-linear decision-making.	<code>tree.DecisionTreeRegressor</code>	<code>max_depth = 5</code> , <code>min_samples_split = 20</code>
Random forest	<code>n_estimators</code> , <code>max_depth</code>	An ensemble learning method that uses multiple decision trees to improve accuracy and prevent overfitting.	<code>ensemble.RandomForestRegressor</code>	<code>n_estimators = 100</code> , <code>max_depth = 10</code>
Gradient boosting	<code>n_estimators</code> , <code>learning_rate</code>	An ensemble method that builds trees sequentially, each correcting the errors of the previous one to minimize loss and improve performance.	<code>ensemble.GradientBoostingRegressor</code>	<code>n_estimators = 100</code> , <code>learning_rate = 0.1</code>
AdaBoost	<code>n_estimators</code>	An ensemble learning method that combines the predictions of several weak models to create a stronger overall model.	<code>ensemble.AdaBoostRegressor</code>	<code>n_estimators = 50</code>
K-nearest neighbors	<code>n_neighbors</code>	A simple, distance-based algorithm that classifies data points based on the proximity of their features to known data points.	<code>neighbors.KNeighborsRegressor</code>	<code>n_neighbors = 5</code>
Multi-layer perceptron	<code>hidden_layer_sizes</code> , <code>activation</code>	A neural network model with multiple layers that can capture complex, non-linear relationships within the data.	<code>neural_network.MLPRegressor</code>	<code>hidden_layer_sizes = (100,)</code> , <code>activation = "relu"</code>

Additionally, the table summarizes essential parameters, concise descriptions, and their corresponding implementations in scikit-learn for each model featured in our analysis. It

highlights how models like linear regression, ridge, lasso, and elastic net use regularization techniques to enhance generalization and facilitate feature selection. In contrast, non-linear models such as decision trees, random forests, and gradient boosting rely on parameters like `max_depth` and `n_estimators` to boost predictive accuracy. Models like AdaBoost and K-nearest neighbors emphasize the integration of predictions from multiple models or proximity-based classification.

Meanwhile, the multi-layer perceptron (MLP) adeptly identifies intricate patterns through hidden layers and activation functions. The inclusion of example parameter values offers valuable guidance for optimizing model performance, while the implementation column provides direct references to the specific classes in the scikit-learn library used for each model.

Moreover, the domain parameters column further enriches the table by offering example parameter values commonly used in real-world applications. These values serve as starting points for model tuning and help establish reproducibility in the experimentation process. They are selected based on typical practices in machine learning tasks, providing practitioners with a useful reference for initial configurations and optimization.

4.2. Comparative Evaluation

We employ a comparative evaluation method to identify the most suitable model for our dataset based on its performance metrics. Moreover, we assess all 10 machine learning models using the metrics of mean squared error (MSE), mean absolute error (MAE), median absolute error (MedAE), explained variance score, and R-squared. Additionally, this experiment aims to demonstrate the impact of features constructed using various techniques on the overall model accuracy. To this end, we employ two comparisons: First, we used the GitHub dataset [28] and evaluate it with all 10 machine learning models. Second, we incorporate features that we construct using three methods and replicated the first approach. Our primary objective is to select the most efficient and dependable model for our study, striking an optimal balance of performance across various factors while reducing potential biases.

To achieve this, we conducted a comparative analysis of all 10 machine learning models. Below is a brief explanation of the metrics we used and what results are considered acceptable:

- Mean squared error (MSE): Measures the average of the squares of the errors between the predicted and actual values. A lower MSE indicates better model performance.
- Mean absolute error (MAE): Calculates the average of the absolute differences between the predicted and actual values. A lower MAE is desirable as it signifies a model's accuracy in predictions.
- Median absolute error (MedAE): Finds the median of the absolute errors, offering a robust measure of central tendency that is less affected by outliers. Lower values indicate more accurate predictions.
- Explained variance score (EVS): Indicates the proportion of variance in the target variable that is explained by the model's predictions. A higher explained variance score is preferable, as it suggests better model performance.
- R-squared (R^2): Represents the proportion of variance in the target variable that is explained by the model. R-squared ranges from 0 to 1, with higher values indicating better model fit.
- Root mean squared error (RMSE): It quantifies the difference between the predicted values generated by the model and the actual values observed in the dataset. A lower RMSE indicates better model performance.

Table 7 presents the results of two comparisons. The first comparison focuses on evaluating the machine learning models' performance using only the baseline feature set, while the second comparison includes the extended feature set. This allows us to observe the impact of incorporating the additional features on model performance. In the table, 'without' is used as a shorthand for the baseline feature set, and 'with' represents the extended feature set. This choice helps keep the table concise and more readable without sacrificing

the clarity of the presented comparisons. In examining the results across various models, we see a spectrum of performance changes influenced by the introduction of constructed features. For instance, gradient boosting shows only marginal improvements, with its mean squared error (MSE) decreasing slightly from 0.0568 to 0.0563. This minimal change indicates that the model’s complexity and regularization may have limited its responsiveness to the added features. Conversely, some models like AdaBoost and K-nearest neighbors (KNN) experience significant declines in performance after incorporating new features. The MSE of AdaBoost increases from 0.5739 to 0.6821, indicating a detrimental effect from the added complexity, while the performance of KNN also worsens, indicating that these models may not effectively handle the enriched dataset. Notably, ridge regression stands out as the most effective model in this context. After integrating the constructed features, ridge regression’s MSE decreases from 0.0019 to 0.0006, a significant reduction of 0.0013. This improvement is echoed in its mean absolute error (MAE), which drops from 0.0225 to 0.0096, representing a decrease of 0.0128. The model also achieves impressive gains in the explained variance score (EVS) and R-squared, increasing by approximately 12%, highlighting its enhanced ability to explain the variance in the target variable. These results collectively illustrate that while some models struggled with added features, ridge regression’s robust performance underscores its capability to leverage additional data effectively, making it a superior choice for this analysis. This performance disparity emphasizes the importance of selecting the right model based on its adaptability to new features.

Table 7. Comparison of evaluation metrics for various machine learning models.

Metric	LR	RR	Lasso	EN	DT	RF	GB	Ada	KNN	MLP
MSE (Without)	0.0019	0.0019	0.0066	0.4753	0.1036	0.0651	0.0568	0.5739	4.0126	0.2136
MSE (With)	0.0006	0.0006	0.0072	0.5323	0.1260	0.0707	0.0563	0.6821	4.0238	1.8178
MSE (Difference)	−0.0013	−0.0013	+0.0006	+0.0570	+0.0224	+0.0056	−0.0004	+0.1082	+0.0112	+1.6042
MAE (Without)	0.0225	0.0225	0.0531	0.3815	0.0701	0.0684	0.0969	0.6353	1.1145	0.3009
MAE (With)	0.0096	0.0096	0.0542	0.3905	0.0761	0.0716	0.0944	0.7073	1.1159	1.2135
MAE (Difference)	−0.0128	−0.0128	+0.0011	+0.0090	+0.0060	+0.0032	−0.0026	+0.0720	+0.0014	+0.9125
MedAE (Without)	0.0131	0.0131	0.0412	0.1858	7.5×10^{-5}	0.0002	0.0289	0.6443	0.5996	0.1766
MedAE (With)	0.0034	0.0034	0.0435	0.2808	8.0×10^{-5}	0.0002	0.0265	0.7293	0.6004	1.1662
MedAE (Difference)	−0.0097	−0.0097	+0.0023	+0.0950	$+5.3 \times 10^{-6}$	+0.00002	−0.0024	+0.0850	+0.0008	+0.9900
EVS (Without)	0.8318	0.8318	0.396	0.89	0.8893	0.8933	0.8941	0.8692	0.5895	0.8799
EVS (With)	0.9469	0.9477	0.423	0.92	0.8870	0.8927	0.8942	0.8666	0.5884	0.8639
EVS (Difference)	+0.1151	+0.1159	+0.027	+0.04	−0.0023	−0.0006	+0.0001	−0.0026	−0.0011	−0.0160
R ² (Without)	0.8318	0.8318	0.396	0.89	0.8893	0.8933	0.8941	0.8692	0.5853	0.8799
R ² (With)	0.9469	0.9477	0.423	0.92	0.8870	0.8927	0.8942	0.8666	0.5841	0.8639
R ² (Difference)	+0.1152	+0.1159	+0.027	+0.04	−0.0023	−0.0006	+0.0001	−0.0026	−0.0012	−0.0160
RMSE (Without)	0.024541	0.024312	0.0814	0.067995	0.009250	0.006527	0.012070	0.056705	0.082561	0.385213
RMSE (With)	0.023415	0.023098	0.0827	0.067421	0.009456	0.006803	0.012034	0.058123	0.082065	0.382731
RMSE (Difference)	−0.001126	−0.001214	+0.0013	−0.000574	+0.000206	+0.000276	−0.000036	+0.001418	−0.000496	−0.002482

4.3. Model Validation

Ensuring the accuracy and effectiveness of machine learning models is essential for reliable and precise predictions. Model validation techniques play a crucial role in this process by providing valuable insights into a model’s performance and its ability to generalize to new data. One such technique is K-fold cross-validation, which divides the dataset into K subsets and trains the model K times, each time using a different subset as the validation set and the remaining subsets as the training set [38]. This approach enables a thorough evaluation of the model’s performance across diverse subsets of the data, minimizing the risk of overfitting and yielding a more dependable estimate of its predictive accuracy.

In our study, we use K-fold cross-validation, a widely used method for model validation to evaluate our model’s accuracy and effectiveness [39]. We chose the value of K as 5 since it is a commonly accepted standard in machine learning and statistical analysis,

offering a good balance between computational efficiency and validation accuracy [38,39]. Additionally, through cross-validation, we obtained scores for all machine learning models by evaluating performance metrics. In Table 8, the average performance metrics of the machine learning models are presented. The implementation of K-fold cross-validation ensures that the results are robust and reliable, forming a solid foundation for the analysis. Furthermore, it is essential to note that the metrics reported are derived from models that incorporate the extended feature set which includes additional variables that enhance the predictive capabilities of the models, thereby providing a more comprehensive understanding of the factors influencing player performance. Ridge regression emerged as a standout model, achieving a mean squared error (MSE) of 0.0005 and a mean absolute error (MAE) of 0.0094. This performance underscores its effectiveness in aligning predicted outcomes with actual player performance, showcasing its capacity to minimize prediction errors. Notably, the R^2 score for ridge regression reached an impressive 0.953, indicating that approximately 95% of the variance in player metrics can be explained by this model. This strong performance is on par with linear regression, which also achieved an MSE of 0.0005 and an R^2 score of 0.953. Such results highlight the reliability of both models in providing valuable insights into player performance. Conversely, lasso regression fell short with an MSE of 0.0066 and a significantly lower R^2 score of 0.396, indicating its limitations in capturing the complexity of player metrics. Similarly, models like K-nearest neighbors (KNN) and multi-layer perceptron (MLP) demonstrated less favorable outcomes, with MSE values of 0.0065 and a staggering 1.5914, respectively. These discrepancies illustrate how some models struggled to generalize effectively, especially in a domain characterized by inherent variability. Interestingly, the performance of decision trees and random forests also revealed a nuanced narrative. While decision trees recorded a low MSE of 0.0002 and an R^2 score of 0.985, random forests outperformed them with an even lower MSE of 0.0001 and an R^2 score of 0.995. However, their predictive performance did not consistently translate into practical application, as their complexity can lead to overfitting, which is a significant concern in sports analytics where data variability is high.

Table 8. Average performance metrics of machine learning models (calculated using K-fold cross-validation).

Model	MSE	MAE	MedAE	R^2	EVS	RMSE
LR	0.0005	0.0094	0.0035	0.953	0.9534	0.0226
Ridge	0.0005	0.0094	0.0035	0.953	0.9534	0.0226
Lasso	0.0066	0.0531	0.0412	0.396	0.3958	0.0814
EN	0.0044	0.0418	0.0307	0.596	0.5960	0.0666
DT	0.0002	0.0027	0.0002	0.985	0.9835	0.0124
RF	0.0001	0.0014	0.0002	0.995	0.9953	0.0071
GB	0.0002	0.0064	0.0031	0.986	0.9859	0.0124
AB	0.0033	0.0517	0.0514	0.703	0.9095	0.0569
KNN	0.0065	0.0507	0.0318	0.404	0.4093	0.0809
MLP	1.5914	0.8949	0.7981	−146.530	−6.6540	1.0822

5. Evaluation and Research Findings

Before proceeding to this section, it is essential to present a thorough overview of the testing process. Figure 3 visually depicts the steps involved in this testing procedure, showcasing the methodology we utilized to assess the model's performance. This flowchart acts as a roadmap for comprehending the systematic approach we adopted during testing, emphasizing the key stages from model selection to performance evaluation. To implement this prediction, we employed the widely-used Python library, scikit-learn. This library provides robust tools for machine learning tasks, including ridge regression, which is well-suited for handling correlation and reducing overfitting in our predictive models. Using scikit-learn, we first trained the ridge regression model on our dataset, leveraging its regularization capabilities to enhance prediction stability. Subsequently, we applied this

trained model to predict xGg and aGg values based on relevant input features derived from our data preprocessing and feature engineering steps.

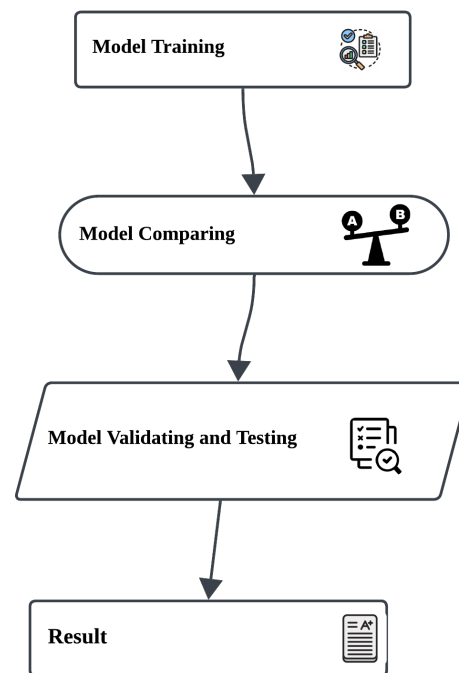


Figure 3. Testing process overview.

In this section, we present our results on predicted xGg and predicted aGg . In the previous section, we identified ridge regression (RR) as the optimal model for our dataset through comparative evaluation. Consequently, we utilized the RR model to predict both xGg and aGg . This choice was informed by its demonstrated superiority in handling our dataset's characteristics and achieving accurate predictions.

First, we identified the top five players in our dataset who excelled according to their aGg as illustrated in Table 9. Moreover, in this study, we opted to use player IDs instead of names for several reasons. Player IDs provide a unique identifier for each player, minimizing confusion in cases where players share similar or common names. Additionally, using player IDs preserves the anonymity of players, especially in research involving sensitive data or personal information. This approach ensures the integrity and confidentiality of the data while allowing for efficient analysis.

Table 9. Top players with maximum aGg (in descending order).

Player ID	Actual Goals per Game (aGg)
227	1.41
2371	1.37
2098	1.14
2097	1.13
2099	1.10

We analyze the differences in performance between the players based on their actual goals per game (aGg) values.

- Player ID 227: This player stands out with the highest actual goals per game value of 1.41. This is a significant achievement compared to the other players in the table.
- Player ID 2371: This player follows closely behind with a actual goals per game value of 1.37, which is just 0.04 lower than Player ID 227. The small difference indicates comparable performance levels between the two players.

- Player ID 2098: The actual goals per game value for this player is 1.14, which is noticeably lower than the top player (227) by 0.27 and lower than Player ID 2371 by 0.23. This suggests a moderate decrease in goal-scoring efficiency compared to the top performers.
- Player ID 2097: With an actual goals per game value of 1.13, this player is close to Player ID 2098, only 0.01 lower. However, compared to the top player (227), the difference is more pronounced, amounting to 0.28.
- Player ID 2099: This player has the lowest actual goals per game value in the table at 1.10. Compared to the top player, the difference is 0.31, indicating a larger gap in goal-scoring efficiency.

Overall, the numerical analysis shows a range of goals per game values among the top players, with player ID 227 leading by a considerable margin. While the other players also have high *aGg* values, there is a noticeable decline in efficiency as we move down the list. This analysis highlights the goal-scoring capabilities and differences in performance among the top players.

Moreover, our second objective is to forecast *xGg* to assess player performance. Table 10 presents data on the top five players based on *xGg*. Notably, the same five players who were identified as the top performers in actual goals per game also rank highly in *xGg* according to our model's predictions. This alignment suggests consistency in performance across different metrics, highlighting the model's accuracy in identifying top talent.

Table 10. Top players with maximum predicted *xGg* (in descending order).

Player ID	Expected Goals per Game (<i>xGg</i>)
2371	1.13
2098	1.09
227	1.09
2097	0.99
2099	0.76

We analyze the differences in performance between the players based on their predicted expected goals per game (*xGg*) values.

- Player ID 2371: This player leads the table with a predicted *xGg* per game value of 1.13. This indicates a consistent ability to generate high-quality scoring chances and outperforms other players in the table.
- Player ID 2098: This player has an *xGg* value of 1.09, identical to Player ID 227, and is only 0.04 lower than Player ID 2371. This small difference indicates a similar level of performance in generating expected goals per game.
- Player ID 227: With an *xGg* of 1.09, identical to Player ID 2098, this player's value is also slightly lower than Player ID 2371 by 0.04. This minor gap reflects a comparable ability to create scoring opportunities.
- Player ID 2097: This player's *xGg* value of 0.99 shows a more noticeable drop from the top three performers, falling 0.10 behind Player ID 227. This decrease may suggest a moderate reduction in efficiency for generating quality chances.
- Player ID 2099: This player has the lowest predicted *xGg* per game value in the table at 0.76. The difference from Player ID 2097 is 0.23, showing a significant gap in efficiency compared to the top players in terms of creating quality scoring opportunities.

Overall, the numerical analysis of predicted *xGg* values demonstrates a range of performance levels among the top players. Player ID 2371 leads with the highest *xGg* value, showing a consistent ability to generate high-quality scoring opportunities. While Player IDs 2098 and 227 follow closely behind, their slight differences suggest comparable efficiency in creating *xGg*. As we move further down the list, the efficiency in generating quality chances decreases noticeably, with Player IDs 2097 and 2099 exhibiting more significant gaps compared to the top performers. This analysis highlights the differences

in players' capabilities to create scoring opportunities, emphasizing the variations in performance across the top players.

Moreover, when a player or team consistently scores more goals than their xGg , it typically signifies outstanding finishing skills and effectiveness in capitalizing on scoring chances [25]. To capture these variations in performance, we suggest classifying the top players into two distinct groups based on the difference between their aGg and xGg as shown in the differences column of Table 11. According to our model, we categorize players with a difference greater than 30% as efficient scorers, while those with a difference under 30% are identified as consistent performers.

- Efficient scorers:
 - Player ID 2099: Displays a difference of 0.34. This substantial difference indicates outstanding efficiency in converting chances into goals.
 - Player ID 227: Exhibits a difference of 0.32. This player demonstrates exceptional efficiency and finishing ability, significantly outperforming their xGg .
- Consistent performers:
 - Player ID 2371: Has a difference of 0.24. This player is close to the threshold between the two groups, suggesting a reasonable level of consistency in generating and converting scoring opportunities.
 - Player ID 2097: Shows a difference of 0.14. This player demonstrates steady and reliable performance in line with their xGg .
 - Player ID 2098: Displays a difference of 0.05. This minimal difference indicates a close alignment between aGg and xGg , suggesting consistent performance.

Table 11. Top players with maximum predicted actual goals per game (aGg) and predicted expected goals per game (xGg), sorted by difference ($aGg - xGg$).

Player ID	aGg	xGg	$(aGg - xGg)$
227	1.41	1.09	0.32
2371	1.37	1.13	0.24
2097	1.13	0.99	0.14
2099	1.10	0.76	0.34
2098	1.14	1.09	0.05

Overall, the analysis shows varying levels of efficiency in generating and converting scoring chances among the top players. Players categorized as efficient scorers such as Player ID 227 and Player ID 2099 exhibit remarkable finishing ability and significantly outperform their xGg . Meanwhile, the consistent performers such as Player ID 2371, Player ID 2097, and Player ID 2098 maintain a closer alignment between aGg and xGg , providing steady performance across the board.

6. Discussion and Limitations

Our analysis reveals valuable insights into player performance through the predictions of actual goals per game (aGg) and expected goals per game (xGg). The ridge regression model proved effective in predicting these metrics, showcasing its capability in managing our dataset's characteristics and avoiding overfitting. The results indicate that Player ID 227 stands out with the highest aGg , reflecting exceptional goal-scoring efficiency. On the other hand, Player ID 2099, while having the lowest aGg among the top players, shows remarkable efficiency in relation to their predicted xGg . The alignment between predicted xGg and actual aGg for top players highlights the model's accuracy in identifying consistently high performers. The classification of players into efficient scorers and consistent performers provides actionable insights for teams and coaches. Efficient scorers, who significantly exceed their predicted xGg , can be valuable targets for enhancing finishing skills and refining game strategies. Conversely, consistent performers offer reliability and stability, contributing to overall team consistency. By focusing on these insights, teams can better

strategize player recruitment, training, and in-game tactics to optimize performance and achieve better results.

Even though our analysis provides valuable insights into player performance and predictive accuracy, several limitations should be acknowledged. Firstly, the dataset used is limited in scope and may not fully represent the diversity of leagues and player conditions. Factors such as injuries, changes in team dynamics, and variations in playing conditions are not accounted for, which could impact the accuracy of predictions. Another significant limitation is that the model's performance could be improved with a more extensive dataset, particularly if it includes historical performance data from individual teams. If a specific soccer team were to collect and utilize detailed historical data on its players, it could enhance the model's accuracy and predictive power, providing more tailored insights for that team. Lastly, the focus on expected goals per game (xGg) does not encompass other critical aspects of player performance, such as defensive contributions and teamwork. A more comprehensive model incorporating these elements could provide a fuller picture of player effectiveness. Addressing these limitations in future research could enhance the robustness and applicability of predictive models in soccer analytics.

7. Conclusions

This study presents a new approach to assess player performance in the top five European soccer leagues and Russian Premier League. We enhance the accuracy of our predictive models by integrating actual goals per game (aGg) and expected goals per game (xGg) metrics with constructed features and estimating continuous metrics through supervised regression that align with our target variables. Regression models are the optimal choice for this analysis due to their ability to handle continuous numerical values.

To select the best-fit model and evaluate the accuracy of our predictions, we conducted two experimental approaches. First, we carry out our analysis without incorporating constructed features, providing a baseline for our model's performance. Second, we include constructed features in our models to examine their impact on prediction accuracy. This comparative evaluation allowed us to measure the improvement in model performance after adding features. Among the machine learning models evaluated, ridge regression consistently demonstrated superior performance, which led us to use it to predict aGg and xGg . One of the study's most compelling contributions is the categorization of players into two groups: efficient scorers and consistent performers. Players with a difference greater than 30% between aGg and xGg are classified as efficient scorers, while those with a difference under 30% are identified as consistent performers.

The analysis highlights the varying levels of efficiency in generating and converting scoring opportunities among the top players. Efficient scorers, such as player IDs 227 and 2099, exhibit remarkable finishing ability and significantly outperform their xGg . Meanwhile, consistent performers such as Player IDs 2371, 2097, and 2098 maintain a closer alignment between aGg and xGg , providing steady and reliable performance. Our study offers valuable insights for soccer coaches and team scouts, aiding in the selection and acquisition of players based on their performance metrics. This approach can enhance team strategies and decision-making processes by leveraging data-driven insights into players' strengths and potential impact on the game.

Author Contributions: Conceptualization, D.M., J.K.; methodology, D.M., J.K.; data curation, D.M.; validation D.M.; writing—original draft preparation, D.M.; formal analysis D.M.; writing—review and editing, D.M., J.K.; visualization, D.M., J.K.; supervision, J.K.; funding acquisition, J.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work supported by the research grant of the Gyeongsang National University in 2023.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Acknowledgments: We would like to thank to our lab members for their constructive discussions and valuable feedback which greatly contributed to the development of this work. Special thanks to my family and friends for their unwavering patience and encouragement throughout this work.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Fédération Internationale de Football Association (FIFA). More than Half the World Watched Recordbreaking 2018 World Cup. p. e0282295. Available online: <https://inside.fifa.com/tournaments/mens/worldcup/2018russia/media-releases/more-than-half-the-world-watched-record-breaking-2018-world-cup> (accessed on 13 December 2023).
2. Wood, R. World's Most Popular Sports by Fans. 2008. Available online: <https://www.topendsports.com/world/lists/popular-sport/fans.htm> (accessed on 20 January 2024)
3. News, S. Premier League Central Payments to Clubs 2022/23. 2024. Available online: <https://www.premierleague.com/news/3676561> (accessed on 20 January 2024).
4. Mead, J.; O'Hare, A.; McMenemy, P. Expected goals in football: Improving model performance and demonstrating value. *PLoS ONE* **2023**, *18*, e0282295. [[CrossRef](#)] [[PubMed](#)]
5. Zeng, Z.; Pan, B. A Machine Learning Model to Predict Player's Positions based on Performance. In Proceedings of the icSPORTS, Valletta, Malta, 28–29 October 2021; pp. 36–42.
6. Pinnacle. Understanding the Limitations of Expected Goals. 2023. Available online: <https://www.pinnacle.com/betting-resources/en/soccer/understanding-the-limitations-of-expected-goals/k4gjc1wx3vs6mwvk> (accessed on 1 December 2023).
7. Larson, O. Charles Reep: A major influence on British and Norwegian football. *Soccer Soc.* **2001**, *2*, 58–78. [[CrossRef](#)]
8. Pollard, R. Charles Reep (1904–2002): Pioneer of notational and performance analysis in football. *J. Sport. Sci.* **2002**, *20*, 853–855. [[CrossRef](#)]
9. Bening, S. Data Science and Soccer: The xG Phenomenon. 2023. Available online: <https://annals.yonsei.ac.kr/news/articleView.html?idxno=10988> (accessed on 5 November 2023).
10. Statsperform. EXPECTED GOALS IN CONTEXT. Available online: <https://www.statsperform.com/resource/expected-goals-in-context/> (accessed on 10 October 2023).
11. Anzer, G.; Bauer, P. A goal scoring probability model for shots based on synchronized positional and event data in football (soccer). *Front. Sport. Act. Living* **2021**, *3*, 624475. [[CrossRef](#)] [[PubMed](#)]
12. Eggels, H.; Van Elk, R.; Pechenizkiy, M. Explaining soccer match outcomes with goal scoring opportunities predictive analytics. In Proceedings of the 3rd Workshop on Machine Learning and Data Mining for Sports Analytics (MLSA 2016), Riva del Garda, Italy, 19 September 2016. Available online: <https://ceur-ws.org/> (accessed on 1 December 2023).
13. Lucey, P.; Bialkowski, A.; Monfort, M.; Carr, P.; Matthews, I. Quality vs Quantity: Improved Shot Prediction in Soccer Using Strategic Features from Spatiotemporal Data. 2015. Available online: <https://www.sloansportsconference.com/research-papers/quality-vs-quantity-improved-shot-prediction-in-soccer-using-strategic-features-from-spatiotemporal-data> (accessed on 24 January 2024).
14. Rathke, A. An examination of expected goals and shot efficiency in soccer. *J. Hum. Sport Exerc.* **2017**, *12*, 514–529. [[CrossRef](#)]
15. Spearman, W. Beyond expected goals. In Proceedings of the 12th MIT Sloan Sports Analytics Conference, Boston, MA, USA, 23–24 February 2018; pp. 1–17.
16. Kharrat, T.; McHale, I.G.; Peña, J.L. Plus–minus player ratings for soccer. *Eur. J. Oper. Res.* **2020**, *283*, 726–736. [[CrossRef](#)]
17. Madrero Pardo, P. Creating a Model for Expected Goals in Football Using Qualitative Player Information. Master's Thesis, Universitat Politècnica de Catalunya, Barcelona, Spain, 2020.
18. Schulze, E.; Mendes, B.; Maurício, N.; Furtado, B.; Cesário, N.; Carriço, S.; Meyer, T. Effects of positional variables on shooting outcome in elite football. *Sci. Med. Footb.* **2018**, *2*, 93–100. [[CrossRef](#)]
19. Brechot, M.; Flepp, R. Dealing with randomness in match outcomes: How to rethink performance evaluation in European club football using expected goals. *J. Sport. Econ.* **2020**, *21*, 335–362. [[CrossRef](#)]
20. Joseph, A.; Fenton, N.E.; Neil, M. Predicting football results using Bayesian nets and other machine learning techniques. *Knowl.-Based Syst.* **2006**, *19*, 544–553. [[CrossRef](#)]
21. Tait, J. Just How Accurate is Expected Goals? 2022. Available online: <https://jacktait.substack.com/p/just-how-accurate-is-expected-goals> (accessed on 14 March 2024).
22. Garratt-Stanley, F. What is Expected Goals (xG)? 2022. Available online: <https://jobsinfootball.com/blog/what-is-expected-goals-xg/> (accessed on 14 March 2024).
23. Statsbomp. What Are Expected Goals (xG)? Available online: <https://statsbomb.com/soccer-metrics/expected-goals-xg-explained/> (accessed on 14 March 2024).
24. Opta Analyst. Opta Analyst: Expected Goals (xG). 2023. Available online: <https://theanalyst.com/eu/2023/08/what-is-expected-goals-xg/> (accessed on 10 December 2023).
25. Tweedale, A. Expected Goals: Explained. 2022. Available online: <https://www.coachesvoice.com/cv/expected-goals-xg-explained/> (accessed on 10 December 2023).

26. Chappas, C. Comparing Actual and Expected Goals. 2014. Available online: <https://statsbomb.com/articles/soccer/comparing-actual-and-expected-goals/> (accessed on 15 December 2023).
27. Statistics. 2022/23 in numbers: Who did best for Expected Goals? 2023. Available online: <https://www.premierleague.com/news/3533343> (accessed on 10 November 2024).
28. Webster, E. Soccer Analytics. 2022. Available online: https://github.com/eddwebster/football_analytics/tree/master/data/understat/raw/metadata (accessed on 15 September 2023).
29. Website:Understat Professional Soccer Website. 2022. Available online: <https://understat.com/> (accessed on 15 September 2023).
30. Scikit. Machine Learning in Python. Available online: <https://scikit-learn.org/stable/> (accessed on 7 February 2024).
31. Pandas. Manipulation with Python. Available online: <https://pandas.pydata.org/> (accessed on 7 February 2024).
32. Numpy. Statistics with Python. Available online: <https://numpy.org/> (accessed on 7 February 2024).
33. Seaborn. Statistical Data Visualization in Python. Available online: <https://seaborn.pydata.org/> (accessed on 7 February 2024).
34. Matplotlib. Visualization with Python. Available online: <https://matplotlib.org/> (accessed on 7 February 2024).
35. UEFA. Club Stats. 2024. Available online: <https://www.uefa.com/uefachampionsleague/statistics/clubs/> (accessed on 15 December 2023).
36. Statista. Average Attendance of the Big Five Soccer Leagues in Europe from 2013/14 to 2022/23, by League. 2023. Available online: <https://www.statista.com/statistics/261213/european-soccer-leagues-average-attendance/> (accessed on 15 December 2023).
37. Jannik Lindner. Statistics About the Most Important Position in Soccer. 2024. Available online: <https://gitnux.org/most-important-position-in-soccer/> (accessed on 12 March 2024).
38. Brownlee, J. A Gentle Introduction to k-fold Cross-Validation. 2023. Available online: <https://machinelearningmastery.com/k-fold-cross-validation/> (accessed on 10 June 2024).
39. Kohavi, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI), Montreal, QC, Canada, 20–25 August 1995; Volume 2, pp. 1137–1143.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.