

Optimized Gradient Boosting Models for Predicting Over/Under 2.5 Goals in La Liga

Zihui Deng^{†1,3}, Wei Liu^{†1,3}, Fangtian Liu^{2,3}, Ebenezer Fiifi Emire Atta Mills^{†2,3,*}

¹Department of Computer Science, Wenzhou-Kean University, Wenzhou, 325060, China

²Department of Mathematical Sciences, Wenzhou-Kean University, Wenzhou, 325060, China

³WKU Academy of Interdisciplinary Research for Sustainability (WKU-AIRs), Wenzhou-Kean University, Wenzhou, 325060, China

Abstract

This paper investigates five advanced gradient boosting techniques, namely Adaptive Boosting, Gradient Boosting Machine, eXtreme Gradient Boosting (XGBoost), Categorical Boosting (CatBoost), and Light Gradient Boosting Machine (LightGBM), for predicting over/under 2.5 goals in Spanish La Liga matches. Using data from 2010 to 2024, the models were evaluated through feature engineering, including Principal Component Analysis for dimensionality reduction and the development of performance indicators. Comparisons were made using precision, recall, F1 score, accuracy, and area under the receiver operating characteristic curve scores. XGBoost showed the highest accuracy and robustness, while LightGBM demonstrated efficiency and scalability. CatBoost excelled at handling categorical features. The introduction of the Odds Balance Ratio and UnderOver Ratio significantly improved predictive performance. To verify transferability, the trained models were further evaluated on English Premier League and Turkish Süper Lig data sets. The results show that the models generalize well to the English Premier League, which has similar competitive characteristics to La Liga, but experience a substantial performance decline on the Turkish Süper Lig, highlighting the importance of league-specific adaptation in football prediction. The findings have practical applications in betting strategies and sports management, offering a data-driven approach to accurately predicting match outcomes. This research advances predictive analytics methods in sports, particularly for predicting football match outcomes.

Keywords: Gradient boosting, Football prediction, XGBoost, CatBoost, LightGBM

1. Introduction

Football is one of the most widely watched sports in the world. Its unpredictable outcomes make the forecasting of match results a challenging yet valuable area of study. This research focuses on the Spanish La Liga, renowned for its fierce competition, dynamic style of play, and extensive historical records, all of which contribute to the complexity of predictive modeling [1].

*Corresponding author.

[†]These authors contributed equally to this work and are considered co-first authors.

Email addresses: dengzih@kean.edu (Zihui Deng[†]), lwei@kean.edu (Wei Liu[†]), liufat@kean.edu (Fangtian Liu), eattamil@kean.edu (Ebenezer Fiifi Emire Atta Mills[†])

The primary objective of this paper is to evaluate the performance of gradient boosting models in predicting the number of goals scored in football matches. Five gradient boosting algorithms are examined in depth: Adaptive Boosting (AdaBoost), Gradient Boosting Machine (GBM), eXtreme Gradient Boosting (XGBoost), Categorical Boosting (CatBoost), and Light Gradient Boosting Machine (LightGBM). These models were selected for their strong performance with complex, variable data structures; however, their effectiveness in sports analytics, particularly in forecasting football match outcomes, remains underexplored. The models were trained on seasonal data from 2010 to 2024, covering 5,319 matches. After training, they were used to predict whether La Liga matches would feature more or fewer than 2.5 goals, which is a common threshold in sports betting markets. The findings have direct implications for betting strategies and sports management. The goal is not only to predict match outcomes but also to advance predictive analytics methods in sports.

The remainder of this paper is structured as follows. Section 2 reviews the relevant literature on the prediction of sports outcomes. Section 3 details our methodology, covering both feature engineering and gradient boosting models. The feature engineering subsection explains how new variables were created, and existing ones were transformed to enhance predictive power, including the use of Principal Component Analysis (PCA) to reduce dimensionality and develop performance indicators. The gradient boosting subsection describes each algorithm's distinctive enhancement technique and explains how the models were implemented and optimized for this prediction task. Section 4 discusses the results for each model, highlighting their respective strengths and weaknesses. We also evaluate the models on two additional football leagues to assess their generalization performance. Finally, Section 5 concludes the paper.

2. Related Work

[2] use gradient-boosted tree models, specifically CatBoost, applied to football-specific ratings, such as pi ratings, to achieve high predictive performance for football match outcomes. They propose future directions for integrating more comprehensive data, such as time-tracking and event data, to further refine these predictions and improve the models' interpretability. [3] research tested five different machine learning models: K-nearest neighbor (k-NN), LogitBoost (LB), support vector machine (SVM), Random Forest (RF), and CatBoost. CatBoost also had the highest accuracy at 67.73%. [4] used models with pi ratings to predict the probability of winning, drawing, or losing.

To explore the potential of deep learning in this context, they developed a new model that integrates components from TimesNet time series models, Transformer models, and neural networks. [5] has developed a machine-learning model capable of evaluating detailed match situations that go beyond goals. [6] created a web application to visualize the model's results, further helping to make sense of the complex data. This paper highlights the critical roles of defenders and midfielders in shaping match outcomes, aiming to offer a more balanced view of players' contributions beyond goal-scoring. [6] conducted experiments with k-NN and XGBoost algorithms and obtained the best initial results for a k-NN model trained on a learning set of graded features. [7] uses a variety of machine learning models, including linear regression, XGBoost, MLP, table transformation, and graph convolutional networks, to analyze data collected from European soccer leagues over multiple seasons. [8] present an improved LightGBM model that incorporates comprehensive data analysis from the FIFA database and uses a tree structure Parzen Estimator (TPE) for hyperparameter optimization.

Although [9]’s research focuses on tennis match prediction, his approach of using ARIMA and LightGBM models to accurately predict match results offers valuable insights. This combination allows for detailed visualization of performance trends and fluctuations during a match. Interestingly, [10] present a conceptual framework for a soccer win–lose prediction system (SWLPS), which focuses on passing distribution data using social network analysis (SNA) and gradient boosting. [11] applied Bayesian networks (BNs) to predict football match results, achieving 92% accuracy for Barcelona matches in the 2008-2009 Spanish Football League, though their data set comprised only 38 matches. These findings highlight numerous opportunities for further exploration in this field.

Numerous studies focus on different sports competitions. [11] concentrate on the Spanish Football League, while [12] and [13] have analyzed FIFA World Cup matches. Beyond football, research by [14], [15], and [16] extends to other sports such as American football and baseball. Moreover, machine learning is not only used for predictions but also for facilitating analytics. Works by [17] and [18] analyze football matches, providing valuable insights for coaches and management to adjust team strategies. However, this study aims to explore multiple gradient-boosting models, with a specific focus on predicting the number of goals in a match, and seeks to establish a unified and stable paradigm to address this challenge.

3. Methodology

3.1. Data Collection

Data for this study were obtained from the Football Data website, which provides comprehensive statistics for various football matches. Specifically, this study focuses on data from the Spanish La Liga from 2010 to 2024. Figure 1 shows the balanced data set for the La Liga, which is very helpful for our model training. We also test on the English Premier League and Turkish League data sets to assess the model’s generalization. The data set includes various features, as detailed in Table 1. These features include odds for different match outcomes, number of goals or fouls, and so on, providing a broad spectrum of dimensions for our predictive models. Because

Table 1. Features collected

Feature	Description
HWOdd, DROdd, AWOdd	Odds for home win, draw, and away win
UnderOdd, OverOdd	Odds for total goals under or over 2.5
Max-UnderOdd, Max-OverOdd	Maximum market odds for total goals under or over 2.5
HTHG, HTAG	Half-time goals by home and away team
HTR	Half-time result: home win, draw, or away win
HST, AST	Shots on target by home and away teams
HS, AS	Shots by home and away teams
HF, AF	Fouls committed by home and away teams
HY, AY	Yellow cards for home and away teams
HR, AR	Red cards for home and away teams

the data set was from 2010, special attention was needed to handle older records with many missing values. We addressed them by partially cleaning them (if some games have too many missing values) and filling in zeros or medians. Before feeding them into the model, all features were

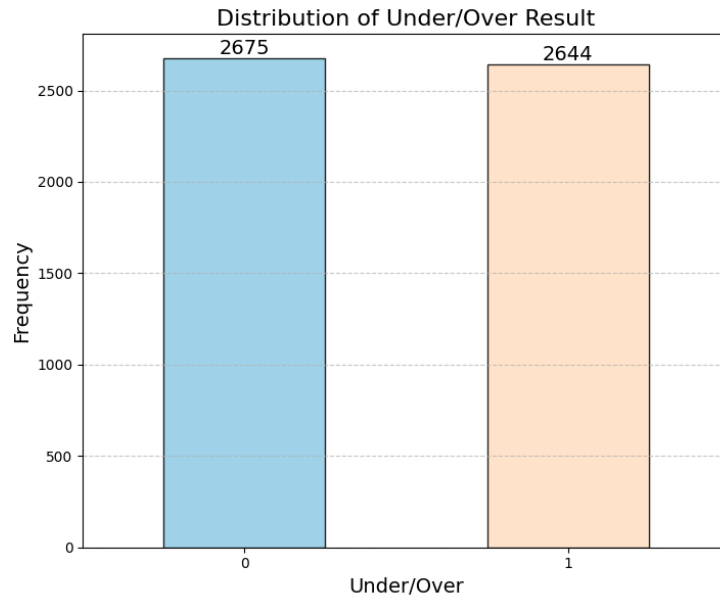


Figure 1. Distribution of Under/Over results

normalized to ensure uniformity. This will help mitigate potential biases or errors arising from data inconsistencies. Next, to generate accurate predictions, we employ a customized prediction approach that tailors the model to each match using features extracted from both historical match data and betting market information. This approach ensures the model adapts to each league's unique characteristics, resulting in more precise predictions. In the following section, we detail the feature engineering process, which plays a crucial role in transforming raw data into meaningful inputs for the predictive model.

3.2. Feature Engineering

Effective feature engineering is essential for improving the predictive power of models. This process involves deriving new variables from raw data using techniques or a uniquely designed formula that captures the underlying patterns of football matches. Our approach focused on developing metrics that reflect historical performance and the betting market's perspective.

We began by calculating historical performance metrics to provide a foundational analysis of each team's capabilities:

- Home Team Metrics: Number of wins ('HW'), draws ('HD'), and losses ('HL') at home.
- Away Team Metrics: Number of wins ('AW'), draws ('AD'), and losses ('AL') away from home.
- Hist-HST, Hist-AST: Historical average of shots on target by home and away teams.
- Hist-HS, Hist-AS: Historical average of shots by home and away teams.
- Hist-HF, Hist-AF: Historical average of fouls committed by home and away teams.

- Hist-HY, Hist-AY: Historical average number of yellow cards for home and away teams.
- Hist-HR, Hist-AR: Historical average number of red cards for home and away teams.

These metrics were accounted for by condensing or shifting past results from every team, which were used as a basis for assessing their performance and for statistical indicators for their genres. We used a sliding window of 5, allowing us to traverse each team's match history in each season within the data set and derive feature values for the most recent 5 matches. We treat these features by setting them to zero for the first five matches of the season and computing them thereafter. This way characterizes the short-term fluctuations within each team and helps address the lack of adequate data in the initial phase of the season. By using the sliding window method, we can generate next-match event predictions, which are the most influential part of the strategy development process and match analysis in general, thereby achieving strong data reliability.

To further synthesize the information and reduce the complexity of our model, we applied PCA. PCA is a statistical technique for dimensionality reduction that preserves as much variability as possible. It transforms the original variables into a new set of uncorrelated variables called principal components. In our case, PCA was applied selectively to a small subset of highly correlated features, primarily to generate new composite variables that capture dominant trends. This preprocessing step does not conflict with the principles of gradient boosting, as it neither distorts the original data distribution nor interferes with the tree-based structure. Rather, it reduces redundancy and noise, allowing the model to converge more efficiently while maintaining its capacity to learn non-linear relationships and feature interactions. Following [19], principal component analysis (PCA) was employed as a dimensionality-reduction procedure to obtain an orthogonal low-dimensional representation of the feature space while preserving the maximum possible variance. Let $\mathbf{X} \in \mathbb{R}^{n \times p}$ denote the data matrix, where n is the number of observations and p is the number of features. The standardized data matrix is defined as

$$\mathbf{Z}_{ij} = \frac{\mathbf{X}_{ij} - \mu_j}{\sigma_j}, \quad (1)$$

where μ_j and σ_j denote the sample mean and standard deviation of the j -th feature, respectively. The empirical covariance matrix of the standardized data is then computed as

$$\mathbf{C} = \frac{1}{n-1} \mathbf{Z}^\top \mathbf{Z}. \quad (2)$$

PCA proceeds by solving the eigenvalue problem

$$\mathbf{C} \mathbf{v}_j = \lambda_j \mathbf{v}_j, \quad (3)$$

where λ_j represents the variance explained by the corresponding eigenvector $\mathbf{v}_j$. The eigenvectors are ordered according to decreasing eigenvalues, and the first k eigenvectors are selected to form the projection matrix

$$\mathbf{W}_k = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k]. \quad (4)$$

The standardized data are then projected onto the reduced k -dimensional subspace as

$$\mathbf{Y} = \mathbf{Z} \mathbf{W}_k. \quad (5)$$

This transformation produces uncorrelated principal components that retain the dominant variance structure of the original feature space while reducing dimensionality and mitigating redundancy among correlated variables.

We separately performed PCA on the sets of home and away performance metrics (HW, HL, HD, AW, AL, AD) to extract principal components that can represent the most significant variance in team performance trends. The resulting principal components, HomeTrend_PCA and AwayTrend_PCA, summarize the historical trends in a single metric for home and away performances, respectively.

Odds Combination Features. In addition to performance metrics, we introduced the Odds Balance Ratio ('ODR'), a new feature calculated with the following steps:

- ODR Formula:
 - $ODR = \left(\frac{\text{Home Win Odds}}{\text{Away Win Odds}} \right) / \text{Draw Odds}$
- Interpretation of ODR:
 - The ratio reflects the bookmakers' perception of the teams' relative strengths. A higher 'ODR' suggests a strong bias toward the home team as the expected winner, with the draw odds serving as a balancing factor.
 - This metric is useful in gauging the market's confidence in the home team's victory over the away team, providing a more granular insight into how odds are set based on historical outcomes, current form, and other strategic factors.
- The 'ODR' provides valuable insights into the dynamics of betting odds, serving as a predictor of match outcomes by revealing underlying market sentiments that might not be apparent through simple odds comparison.

Furthermore, we developed another new metric, the UnderOver Ratio ('UOR'), calculated using the following steps:

- UOR Formula:
 - $UOR = \left(\frac{\text{AvgOverBet}}{\text{MaxOverBet}} + \frac{\text{AvgUnderBet}}{\text{MaxUnderBet}} \right) / 2$
- Interpretation of UOR:
 - If the ratio of AvgOverBet to MaxOverBet is close to 1, it indicates high consistency between the market's average and highest odds. The closer this ratio is to 1, the stronger the consistency.
 - Similarly, the closer the ratio of AvgUnderBet to MaxUnderBet is to 1, the higher the market consistency.
- This feature captures the market's expectation of the total number of goals in a match, thereby providing richer information for predicting match outcomes.

Feature importance is estimated using a Random Forest, which has clear advantages for analyzing feature performance. The ability of this method to capture the tangled patterns of the data by making sample sets smaller through bootstrap sampling and, with that, generating more sub-samples is a great advantage for dealing with complex interactions between sports metrics and the outcome of the game. By implementing the technique that exploits bootstrap aggregation combined with random feature selection at each split, the algorithm inherently evaluates feature

importance, thereby substantially reducing noise in high-dimensional data scenarios typical of sports datasets. This two-fold mechanism, combining ensemble averaging and feature randomization, guarantees stable importance estimates even when features are correlated or samples are diverse.

To quantify team strength in a context-aware manner, we construct two composite indices — the Home Team Performance Index (HTPI) and Away Team Performance Index (ATPI) — that integrate multiple performance metrics with empirically derived weights:

- Home Team Performance Index:

$$\text{HTPI} = w_1 \cdot \text{HWR} + w_2 \cdot \text{HAGS} - w_3 \cdot \text{HAGC}$$

- Away Team Performance Index:

$$\text{ATPI} = w_1 \cdot \text{AWR} + w_2 \cdot \text{AAGS} - w_3 \cdot \text{AAGC},$$

where HWR and AWR represent home and away win rates, HAGS and AAGS denote average goals scored, and HAGC and AAGC indicate average goals conceded for home and away teams, respectively. The weights w_i quantify each metric's discriminative power by reflecting its relative contribution to outcome prediction accuracy, as derived from the Random Forest's feature importance scores. The negative weighting coefficients for HAGC and AAGC reflect their inverse relationship with team competitiveness — teams that concede more goals are inherently weaker, hence the negative contribution to the performance index.

3.3. Boosting Models

AdaBoost was the first boosting algorithm to introduce adaptive sample reweighting: Iteratively increasing the weights of misclassified examples forces subsequent weak classifiers (each learner slightly better than random guessing) to focus on previously misclassified examples. The theoretical significance of this approach lies in its ability to ensure, through a weighted majority vote of weak hypotheses, that the training error decreases exponentially with the number of weak learners, while the generalization error is also bounded [20].

Gradient Boosting Machine provides a theoretically unified framework for optimizing diverse loss functions via functional gradient descent, enabling the optimization of arbitrary differentiable loss functions for regression, classification, and ranking tasks [21]. It trains a new model composed of a sequence of decision trees added one by one, and each new model tries to correct the errors left by the previous one by using negative gradients to move closer to the actual output. However, early implementations faced critical scalability limitations: computing and storing gradients for all training instances at each iteration will cause memory inefficiency, so computational costs grew linearly with feature dimensionality [21]. These limitations restricted the applicability of applications involving high-dimensional data. Later developments in stochastic gradient boosting alleviated these challenges by subsampling instances during tree induction. Randomized sampling reduced memory overhead and detached complexity from feature counts, preserving convergence properties [25]. This innovation expanded the practical utility of gradient-boosted machines without compromising their foundation in gradient-driven optimization.

XGBoost represents a systematic advancement in gradient boosting through three key innovations. First, it introduces a weighted quantile sketch algorithm that approximates optimal feature

split points via histogram-based bucketing, reducing computational complexity from $O(n \log n)$ to $O(n)$. Second, the framework incorporates a regularized objective function that combines l_2 penalties on leaf node weights (λ) with explicit tree complexity control via minimum-loss reduction thresholds (γ), effectively addressing overfitting in high-dimensional scenarios. Third, its sparsity-aware split-finding mechanism automatically determines optimal directions for missing values during tree construction, eliminating the need for manual imputation in sparse data sets [22].

LightGBM employs histogram-based splitting with 255-bin feature discretization, reducing memory consumption by 80%–90% and enabling constant-time split computation through pre-cached gradient statistics [26]. The framework also adopts a leaf-wise growth strategy that prioritizes node expansion based on the maximum reduction in loss per iteration. This approach accelerates training by 3.5 times compared to conventional level-wise methods, while mitigating overfitting through depth constraints. Additionally, LightGBM introduces Exclusive Feature Bundling (EFB), which combines mutually exclusive sparse features into composite dimensions using graph coloring algorithms. This reduces the feature space by 40%–60%, maintaining the integrity of the information [26].

CatBoost addresses the challenges associated with prediction shift and categorical feature handling by introducing several key advancements. The framework incorporates Ordered Target Encoding, which computes target statistics dynamically through progressive permutations during training, effectively mitigating the information leakage commonly observed in traditional encoding methods [24]. In addition, Ordered Boosting ensures unbiased gradient estimation by restricting each tree’s training to previous instances within a randomized permutation sequence, thereby guaranteeing consistency between the distributions of training and inference data [24]. To further optimize computational efficiency, CatBoost employs Oblivious Decision Trees that apply uniform splitting rules across three levels, resulting in a 5-8x increase in prediction throughput compared to conventional gradient-boosted trees while maintaining competitive accuracy [24]. Together with its native support for high-cardinality categorical variables, this approach enables CatBoost to achieve state-of-the-art performance across a wide range of real-world datasets with complex, heterogeneous feature spaces. Please refer to Table 2 for a summary of the formulation of the models. Detailed hyperparameter configurations for each model are provided in A.11.

Table 2. Technical evolution of boosting models

Model	Core Innovation	Mathematical Formulation
AdaBoost	Adaptive sample reweighting [20]	$w_i^{(t+1)} = w_i^{(t)} \cdot \beta_t^{ h_t(x_i) - y_i }$
GBM	Gradient-driven residual fitting [21]	$h_t^* = \arg \min_h \sum_{i=1}^n [y_i - F_{m-1}(x_i)]^2$
XGBoost	Second-order optimization + regularization [22]	$\mathcal{L}^{(t)} = \sum_{i=1}^n \left[g_i h_t(x_i) + \frac{1}{2} h_i h_t^2(x_i) \right] + \Omega(h_t)$
LightGBM	Gradient-based One-Side Sampling + Exclusive Feature Bundling [23]	$\text{Gain} = \frac{1}{n} \left(\frac{(\sum_{A_l} g_i + \frac{1-a}{b} \sum_{B_l} g_i)^2}{n_l} + \frac{(\sum_{A_r} g_i + \frac{1-a}{b} \sum_{B_r} g_i)^2}{n_r} \right)$
CatBoost	Ordered boosting + Ordered target statistics [24]	$r_{it} = - \frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \Big _{F=F_{t-1}(x_{\sigma_{\epsilon t}})}$

4. Results and Discussion

The results of our analysis are presented below and categorized by the model used. Each model was subjected to 5-fold cross-validation to assess its robustness and generalizability. Metrics such as accuracy, AUC (Area Under the Receiver Operating Characteristic Curve), recall, precision, and F1 score are reported for each fold.

Table 3. Model file size comparison (bytes)

File	AdaBoost	CatBoost	GBM	LightGBM	XGBoost
best_model.joblib	129,444	1,984,660	360,234	2,458,932	119,867
model_preprocessor.joblib	2,886	2,886	2,886	3,605	3,605

The comparative storage analysis reveals variations in model file sizes across different algorithms. As shown in Table 3, CatBoost and LightGBM are the two largest models, while XGBoost demonstrates the most compact representation at 119,867 bytes. Preprocessing components remain relatively lightweight across all implementations; all are nearly 4kb. This storage efficiency table complements the earlier performance metrics, a transparent display of model resource usage differences

4.1. Adaboost Model

As reported in Table 4, AdaBoost demonstrates reliable baseline classification capability, achieving stable cross-validation performance with a mean accuracy of 0.7131 ± 0.0110 and an AUC of 0.8153 ± 0.0100 . While exhibiting exceptional sensitivity for Class 0 (recall: 0.9293 ± 0.0110), it suffers from precision limitations (0.6506 ± 0.0100), indicating substantial false positives. Class 1 predictions show inverse characteristics with moderate recall (0.4947 ± 0.0210) but high precision (0.8739 ± 0.0160), revealing asymmetric learning patterns typical of class-imbalanced scenarios. These results confirm AdaBoost's enduring effectiveness as a baseline ensemble method, though its performance disparities between classes highlight opportunities for threshold optimization and class-weight adjustments.

Table 4. Cross-validation results for AdaBoost

Metric	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Accuracy	0.7096	0.6936	0.7256	0.7190	0.7187
AUC	0.7993	0.8047	0.8275	0.8252	0.8198
Recall Class 0	0.9178	0.9271	0.9458	0.9364	0.9196
Recall Class 1	0.4991	0.4575	0.5028	0.4991	0.5152
Precision Class 0	0.6495	0.6335	0.6580	0.6540	0.6578
Precision Class 1	0.8571	0.8612	0.9017	0.8859	0.8635
F1 Score Class 0	0.7607	0.7527	0.7761	0.7702	0.7670
F1 Score Class 1	0.6308	0.5975	0.6456	0.6385	0.6453

Figure 2 highlights the intrinsic limitations of the AdaBoost model. The features HTHG and HTAG account for almost 85% of the total feature contribution, while the remaining variables

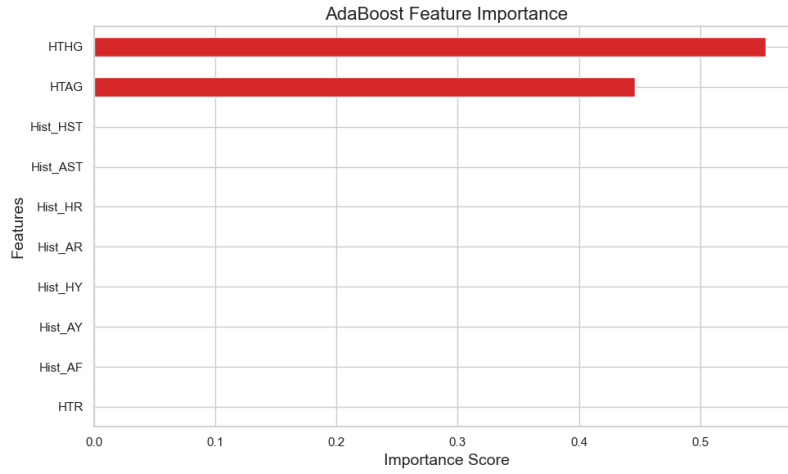


Figure 2. AdaBoost model: top 10 feature importance

exhibit near-zero influence. This pattern suggests that the model fails to effectively capture the broader team-performance feature structure through its boosting mechanism. Such brittleness may stem from AdaBoost’s iterative weight-updating process, which disproportionately amplifies goal-related features during training, thereby increasing model sensitivity and limiting its generalization capacity.

4.2. GBM Model

As shown in Table 5, the GBM model demonstrates a modest improvement in classification performance over the traditional AdaBoost model, achieving a mean accuracy of 0.7261 ± 0.0120 and an AUC of 0.8104 ± 0.0130 across five folds. However, the model does not maintain fully balanced performance for Class 1, with an average recall of 0.6676 ± 0.0310 and an average precision of 0.7541 ± 0.0180 . Although GBM shows progress in class-balance performance compared with earlier boosting implementations, there remains potential for further optimization through hyperparameter refinement or ensemble diversification.

Table 5. Cross-validation results for GBM

Metric	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Accuracy	0.7246	0.7096	0.7472	0.7237	0.7253
AUC	0.7967	0.7885	0.8220	0.8233	0.8217
Recall Class 0	0.8299	0.7738	0.7981	0.7589	0.7589
Recall Class 1	0.6181	0.6446	0.6957	0.6881	0.6913
Precision Class 0	0.6873	0.6877	0.7262	0.7110	0.7135
Precision Class 1	0.7823	0.7381	0.7731	0.7383	0.7389
F1 Score Class 0	0.7519	0.7282	0.7605	0.7342	0.7355
F1 Score Class 1	0.6906	0.6882	0.7323	0.7123	0.7143

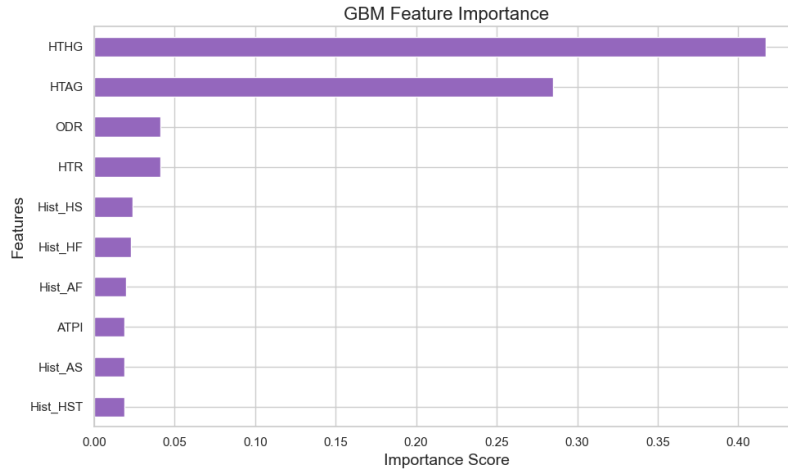


Figure 3. GBM model: top 10 feature importance

The GBM feature importance, as shown in Figure 3, reveals the model’s reliance on Live Match Features such as HTHG and HTAG. However, this situation is not as bad as in Adaboost; other features still show some benefit to the model. The secondary tier of predictive power resides in market-derived indicators, ODR, and HTR, providing quantitative validation that real-time betting market guidance remains intricately linked to in-game performance dynamics.

4.3. Catboost Model

As shown in Table 6, the CatBoost model demonstrates strong negative-class discrimination, with Class 0 recall ranging from 0.8430 in Fold 3 to 0.9159 in Fold 2. This indicates robust true negative detection across the cross-validation folds. However, class 0 precision ranges from 0.6853 (Fold 2) to 0.7080 (Fold 3), suggesting a tendency to misclassify positives as negatives (i.e., higher false positives). In contrast, class 1 shows balanced performance, with recall of 0.5747–0.6484 (Folds 2–3) and precision of 0.8033–0.8711 (Folds 3–2), indicating reliable positive predictions with fewer false alarms.

Table 6. Cross-validation results for CatBoost

Metric	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Accuracy	0.7397	0.7462	0.7462	0.7519	0.7526
AUC	0.7932	0.7987	0.8324	0.8235	0.8297
Recall Class 0	0.8897	0.9159	0.8430	0.8692	0.8991
Recall Class 1	0.5879	0.5747	0.6484	0.6333	0.6042
Precision Class 0	0.6859	0.6853	0.7080	0.7056	0.6971
Precision Class 1	0.8405	0.8711	0.8033	0.8272	0.8552
F1 Score Class 0	0.7746	0.7840	0.7696	0.7789	0.7853
F1 Score Class 1	0.6919	0.6925	0.7176	0.7173	0.7081

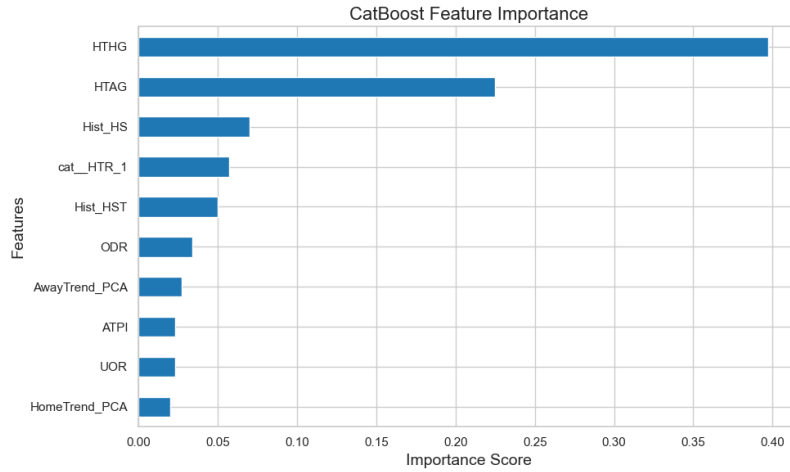


Figure 4. CatBoost model: top 10 feature importance

The Catboost model also shows the importance of real-time game data. The number of goals scored in the first half remains the most important feature, as indicated in Figure 4. It is worth noting that the number of the home team's historical shots is the third most important feature in this model. The historical number of shots on target ranks fifth, and the PCA-compressed value representing the team's performance also ranks in the top ten, indicating that our feature engineering work is effective.

4.4. XGBoost Model

Table 7 indicates that the XGBoost model also performs well, with AUC values ranging from 0.8041 to 0.8399 and accuracy ranging from 0.7359 to 0.7641. The model exhibits high recall for Class 0, ranging from 0.9009 to 0.9271, but comparatively lower precision, ranging from 0.6789 to 0.7006, suggesting the possibility of false positives.

Class 1 shows strong precision (0.8503–0.8904) despite modest recall (0.5614–0.5992), emphasizing its reliability for confirmed predictions. Also, balanced F1 scores (Class 0 and Class 1) highlight stable classification trade-offs.

XGBoost's feature importance (see Figure 5) shows a similar performance distribution to CatBoost. This overlap underscores that gradient-boosting frameworks, despite algorithmic nuances, converge on similar hierarchies when critical predictive signals are strong and well-defined in the data.

4.5. LightGBM Model

Table 8 demonstrates that the LightGBM model has stable performance across folds, with AUC peaking at 0.8326 (Fold 4) and accuracy reaching 0.7566 (Fold 3). Class 0 achieves high recall (0.7888–0.8804), though precision remains moderate (0.6972–0.7240), indicating a tendency for false positives. Class 1 shows improved recall from 0.6163 (Fold 2) to 0.6919 (Fold 4), alongside strong precision (0.7641–0.8359), reflecting enhanced sensitivity without sacrificing reliability. Balanced F1 scores (Class 0: 0.7536–0.7792; Class 1: 0.7050–0.7349) confirm consistent classification trade-offs.

Table 7. Cross-validation results for XGBoost

Metric	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Accuracy	0.7359	0.7415	0.7641	0.7556	0.7528
AUC	0.8041	0.8067	0.8290	0.8284	0.8399
Recall Class 0	0.9009	0.9196	0.9271	0.9196	0.9086
Recall Class 1	0.5690	0.5614	0.5992	0.5898	0.5947
Precision Class 0	0.6789	0.6796	0.7006	0.6939	0.6947
Precision Class 1	0.8503	0.8735	0.8904	0.8789	0.8650
F1 Score Class 0	0.7743	0.7816	0.7981	0.7910	0.7874
F1 Score Class 1	0.6818	0.6835	0.7164	0.7059	0.7048

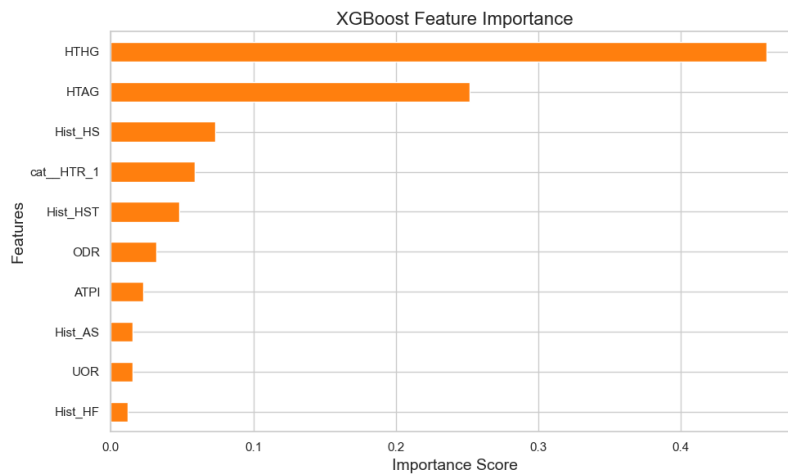


Figure 5. XGBoost model: top 10 feature importance

Table 8. Cross-validation results for LightGBM

Metric	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Accuracy	0.7397	0.7491	0.7566	0.7406	0.7406
AUC	0.7994	0.8039	0.8321	0.8326	0.8295
Recall Class 0	0.8523	0.8804	0.8336	0.7888	0.7966
Recall Class 1	0.6257	0.6163	0.6786	0.6919	0.6837
Precision Class 0	0.6972	0.6988	0.7240	0.7214	0.7189
Precision Class 1	0.8073	0.8359	0.8013	0.7641	0.7681
F1 Score Class 0	0.7670	0.7792	0.7750	0.7536	0.7558
F1 Score Class 1	0.7050	0.7095	0.7349	0.7262	0.7234

Notably, as seen in Figure 6, this model challenges the conventional dominance of goal-related metrics by identifying ODR as the second-most-critical predictor. Historical performance

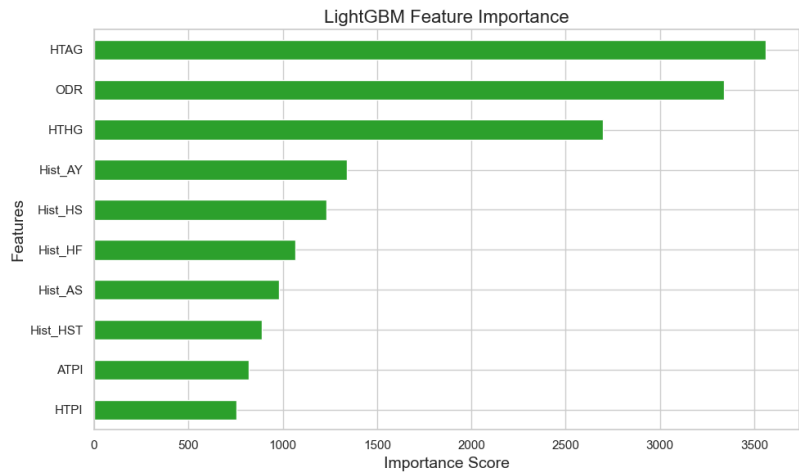


Figure 6. LightGBM model: top 10 feature importance

indicators—including 'Hist_AY' (historical away team yellow cards), 'Hist_HS' (historical home team shots), and 'Hist_AS' (historical away team shots)—emphasize the importance of temporal behavioral patterns in predictive modeling.

4.6. Overall Assessment

Table 9 and Figure 7 provide a comprehensive view of the overall performance metrics averaged across 5 validation folds for each model, simplifying the view by combining class-specific performance indicators into single metrics for accuracy, recall, precision, F1, and AUC score.

Table 9. Comprehensive performance metrics across models with AUC

Model	Accuracy	AUC	Recall	Precision	F1 Score
AdaBoost	0.7133	0.8152	0.7120	0.7622	0.6985
GBM	0.7261	0.8103	0.7257	0.7297	0.7248
CatBoost	0.7473	0.8154	0.7466	0.7680	0.7420
XGBoost	0.7500	0.8216	0.7490	0.7806	0.7425
LightGBM	0.7453	0.8195	0.7448	0.7537	0.7429

XGBoost performs best among five models, achieving the highest accuracy (0.7500), precision (0.7806), and F1 score (0.7425). CatBoost follows closely with a close performance. The accuracy is 0.7473, the precision is 0.7680, and the F1 score is 0.7420, combined with its lightness, making it ideal for scenarios prioritizing computational efficiency without significant performance trade-offs. Regarding LightGBM, while marginally trailing in accuracy (0.7453), it delivers the highest F1 (0.7537), which reflects that the model has achieved an efficient balance between "reducing false positives" and "capturing true positive examples". In contrast, AdaBoost exhibits the lowest accuracy (0.7133) but, notably, achieves a surprisingly high precision (0.7622), suggesting strong specificity in identifying true high-scoring matches despite its

overall limitations. The GBM model outperforms AdaBoost with 0.7261 accuracy and a more balanced precision-recall profile (0.7297 for precision, 0.7248 for F1), illustrating the evolutionary improvements in modern gradient-boosting frameworks like XGBoost and LightGBM, which enhance accuracy by 2-3% while maintaining computational tractability. This experiment underscores the need to select a model based on deployment priorities: raw predictive power (XGBoost), speed-stability balance (LightGBM), or categorical feature integration (CatBoost).

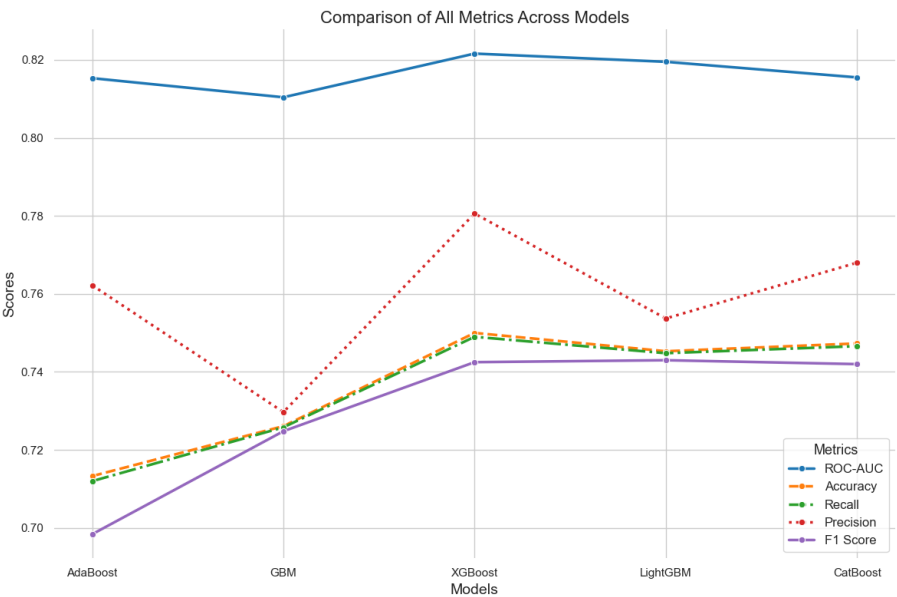


Figure 7. Boosting models’ performance comparison

As depicted in Figure 8, the ROC curves for the models show very tight performance across AdaBoost, GBM, CatBoost, XGBoost, and LightGBM, with AUC scores of 0.8152, 0.8103, 0.8154, 0.8216, and 0.8195, respectively. These scores are very close, indicating that all models have a similar ability to rank positive and negative samples. The AUC values, all above 0.81, demonstrate high model performance in this binary classification task.

Considering both the ROC curve and comprehensive performance metrics, the choice between these models might hinge on specific project requirements beyond AUC and traditional performance metrics. XGBoost is recommended for its slightly better performance in accuracy, precision, and F1 score, making it particularly suitable for applications where the highest possible accuracy is critical. However, LightGBM and CatBoost are also strong options, providing similar discriminative capabilities as evidenced by the AUC values. Given these insights, the decision on which model to deploy should consider other factors such as model training time, resource efficiency, ease of integration into existing systems, and the specific nuances of the prediction task in the Spanish football league context. This comprehensive analysis ensures that the model selection aligns with both the performance goals and the requirements of the predictive task.

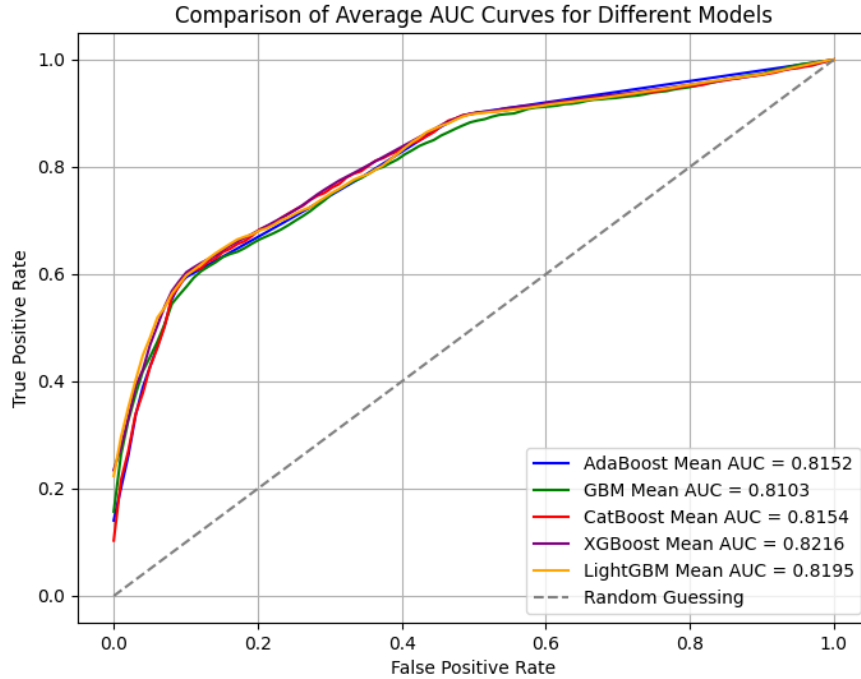


Figure 8. Mean ROC curves for five boosting models across 5-Fold

4.7. Transferability Verification Experiments

To validate the model's generalization capabilities, we conducted additional evaluations across two leagues: the English Premier League (EPL) and the Turkish Süper Lig (TSL). The selection strategy followed a controlled experiment: EPL shares comparable competitive levels, tactical frameworks, and data distribution patterns with the training domain (Spanish La Liga) as both belong to Europe's "Big Five" elite leagues. Conversely, TSL represents a geographically and competition-level distinct league characterized by divergent playing styles, variable defensive-aggressive rhythm metrics, and statistically dissimilar player performance profiles. Using data from these two leagues can better reflect the model's generalization performance.

Experimental results, as shown in Table 10, reveal a minimal performance gap between EPL testing and initial evaluations, with all metrics very close, and CatBoost achieves the highest performance (F1 score: 0.7392, Accuracy: 0.7399) among the five models. For complete hyperparameter specifications used in these transferability experiments, see A.11. In stark contrast, TSL testing exhibited a sharp performance decline, showing an average 20%-30% reduction across all evaluation metrics. Interestingly, the worst model in the Premier League, AdaBoost, has the highest accuracy in TSL. This situation indicates that, in a different feature environment, we should test more models to find the best one. This marked performance dichotomy highlights the domain adaptation challenge in cross-league predictive modeling: While models maintain effectiveness when target league data align with training distributions, they suffer severe

Table 10. Test of english premier league and turkish league data set using models trained in the spanish la liga

Model	Data	Accuracy	Recall	Precision	F1 Score
CatBoost	English Premier League	0.7399	0.7479	0.7545	0.7392
	Turkish League	0.5745	0.5188	0.5178	0.5177
XGBoost	English Premier League	0.7333	0.5900	0.8725	0.7040
	Turkish League	0.5718	0.5112	0.5113	0.5117
LightGBM	English Premier League	0.7342	0.7372	0.7364	0.7341
	Turkish League	0.5807	0.5402	0.5367	0.5358
GBM	English Premier League	0.7160	0.7116	0.7152	0.7124
	Turkish League	0.5121	0.4994	0.4994	0.4868
AdaBoost	English Premier League	0.6837	0.7000	0.7379	0.6749
	Turkish League	0.5985	0.5209	0.5220	0.5211

degradation when confronted with divergent feature patterns that exceed the learned parameters. The findings substantiate the imperative for league-specific domain adaptation frameworks in sports prediction systems, indicating the necessity for individual training or more powerful domain adaptation methods tailored to league-specific characteristics when deploying cross-league prediction systems.

5. Conclusion

This study presents a systematic investigation of gradient boosting models for goal outcome prediction in professional football matches. Through a comprehensive analysis of 5,320 La Liga matches spanning 14 seasons from 2010 to 2024, it was found that modern boosting algorithms, combined with strategic feature engineering, can achieve strong predictive performance. This paper establishes a robust feature processing framework through systematic feature engineering that incorporates novel metrics such as Odds Balance Ratio and UnderOver Ratio, along with historical performance analysis and PCA-based dimensionality reduction. Empirical results demonstrate XGBoost's superior predictive accuracy (0.7500) and discriminative capability ($AUC = 0.8216$), positioning it as the optimal choice for applications that prioritize prediction precision. CatBoost performs very similarly to XGBoost and shows the strongest robustness when tested across leagues of the same level. LightGBM emerges as the benchmark for efficiency, achieving comparable performance while offering greater computational scalability and lower memory usage.

In addition, this study highlights the complex nature of predictive modeling in football, where external factors, such as league characteristics and competition structure, can significantly affect model performance. The model performs respectably in the Premier League, where the level of competition and league intensity are comparable. However, when applied to the substantially different Turkish Süper Lig, a significant decline in performance is observed, with a 20-30% degradation compared with the English Premier League. The substantial performance disparity observed during cross-league validation underscores the critical importance of domain-specific

adaptation in sports analytics models. These findings demonstrate a comprehensive and effective feature engineering workflow, validated through experimentation. Based on this workflow, highly accurate predictive models for football matches have been developed. The methodology also demonstrates adaptability and scalability. Given the similarities in competitive data across sports, this workflow provides a strong foundation for football analytics and holds significant potential for extension to other sports, such as basketball and American football, in order to improve prediction accuracy and support strategy development.

Moreover, this feature engineering framework and predictive approach have cross-domain applications. For example, in the financial sector, similar techniques can be applied to stock market prediction and risk management, while in healthcare, comparable data analysis methods could be used to predict patient outcomes and evaluate treatment efficacy. In this way, data-driven predictive models may continue to spread across various industries, advancing decision-making and intelligent management in multiple fields.

A limitation of this study is that the available data features are somewhat limited in scope. Variables such as team valuation, player injuries, and weather conditions are not considered in the current model. Including these factors could further improve predictive performance and provide a more comprehensive representation of the dynamic nature of football matches.

Future research should focus on the automatic collection of player data in real time using advanced technologies. Another direction is to improve generalization by increasing league diversity, collecting a comprehensive, large-scale data set to support training, and using federated learning to manage each league's data separately while assigning weights accordingly.

Funding

The authors acknowledge funding from the WKU Academy of Interdisciplinary Research for Sustainability (WKU-AIRs), Wenzhou-Kean University. The authors acknowledge support from the Student Partnering with Faculty/Staff Research Program at Wenzhou-Kean University. Grant ID: WKUSPF202434.

References

- [1] H. Sarmiento, A. Pereira, N. Matos, J. Campaniço, T. M. Anguera, and J. Leitão, "English Premier League, Spain's La Liga, and Italy's Serie A – What's different?," *International Journal of Performance Analysis in Sport*, vol. 13, no. 3, pp. 773–789, 2013.
- [2] R. Bunker, C. Yeung, and K. Fujii, "Machine learning for soccer match result prediction," in *Artificial Intelligence, Optimization, and Data Sciences in Sports*, Springer, 2025, pp. 7–49.
- [3] M.-C. Malamatinas, E. Vrochidou, and G. A. Papakostas, "On predicting soccer outcomes in the Greek league using machine learning," *Computers*, vol. 11, no. 9, Art. no. 133, 2022.
- [4] C. Yeung, R. Bunker, R. Umemoto, and K. Fujii, "Evaluating soccer match prediction models: A deep learning approach and feature optimization for gradient-boosted trees," *Machine Learning*, vol. 113, no. 10, pp. 7541–7564, Aug. 2024.
- [5] C. Juhász, "Soccer action evaluation using machine learning algorithms," M.S. thesis, Dept. Telecommun. Media Inform., Budapest Univ. Technol. Econ., 2021.
- [6] D. Berrar, P. Lopes, and W. Dubitzky, "Incorporating domain knowledge in machine learning for soccer outcome prediction," *Machine Learning*, vol. 108, no. 1, pp. 97–126, 2018.
- [7] C. Sheng, "Goal-oriented forecasting: Predicting soccer match outcomes with deep learning," M.S. thesis, Dept. Comput. Sci., Univ. California, 2024.
- [8] H. Lee, B. A. Tama, and M. Cha, "Prediction of football player value using Bayesian ensemble approach," *arXiv preprint*, arXiv:2206.13246, Jun. 2022.
- [9] T. Wang, "A study on visualization of tennis players' game performance based on wavelet analysis and ARIMA-LightGBM model," *Highlights in Science, Engineering and Technology*, vol. 98, pp. 362–371, 2024.

- [10] Y. Cho, J. Yoon, and S. Lee, "Using social network analysis and gradient boosting to develop a soccer win-lose prediction model," *Engineering Applications of Artificial Intelligence*, vol. 72, pp. 228–240, 2018.
- [11] F. Owsramipour, P. Eskandarian, and F. Moznab, "Football result prediction with Bayesian network in Spanish League-Barcelona team," *International Journal of Computer Theory and Engineering*, vol. 5, no. 5, pp. 812–815, 2013.
- [12] L. Hervet-Escobar, T. I. Matis, and N. Hernandez-Gress, "Prediction learning model for soccer matches outcomes," in *Proc. 17th Mexican International Conference on Artificial Intelligence (MICAI)*, Guadalajara, Mexico, Oct. 2018, pp. 63–69.
- [13] J. Shin and R. Gasparyan, "A novel way to soccer match prediction," Stanford Univ., Tech. Rep. CS-2014-01, 2014.
- [14] M. Amor and W. Griffiths, "Modeling the behaviour and performance of Australian football tipsters," Univ. Melbourne, Dept. Econ., Working Paper 2003-05, 2003.
- [15] T. Y. Yang and T. Swartz, "A two-stage Bayesian model for predicting winners in Major League Baseball," *Journal of Data Science*, vol. 2, no. 1, pp. 1–15, 2004.
- [16] J. Martinich, "College football rankings: Do the computers know best?," *Interfaces*, vol. 32, no. 5, pp. 85–94, 2002.
- [17] J. M. Falter and C. Perignon, "Demand for football and intra-match winning probability: An essay on the glorious uncertainty of sports," *Appl. Econ.*, vol. 32, no. 12, pp. 1757–1765, 2000.
- [18] D. Forrest and R. Simmons, "Outcome uncertainty and attendance demand in sport: The case of English soccer," *Statistician*, vol. 51, no. 2, pp. 291–301, 2002.
- [19] F. Kherif and A. Latypova, "Principal component analysis," in *Machine Learning: Methods and Applications*, 1st ed. Amsterdam, Netherlands: Elsevier, 2020.
- [20] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.
- [21] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.
- [22] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, CA, USA, 2016, pp. 785–794.
- [23] G. Ke et al., "LightGBM: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems*, vol. 30, Long Beach, CA, USA, 2017, pp. 3146–3154.
- [24] L. Prokhorenkova et al., "CatBoost: Unbiased boosting with categorical features," in *Advances in Neural Information Processing Systems*, vol. 31, Montréal, Canada, 2018, pp. 6638–6648.
- [25] J. H. Friedman, "Stochastic gradient boosting," *Computational Statistics & Data Analysis*, vol. 38, no. 4, pp. 367–378, 2002.
- [26] G. Ke et al., "LightGBM: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems*, vol. 30, Long Beach, CA, USA, 2017, pp. 3146–3154.

Appendix A. Model Hyperparameters

Model	Hyperparameter	Value
AdaBoost	n_estimators	200
	learning_rate	0.02
	algorithm	SAMME.R
	base_estimator	DecisionTreeClassifier(max_depth=3)
	loss	deviance
GradientBoosting	n_estimators	200
	learning_rate	0.1
	loss	deviance
	criterion	friedman_mse
	max_depth	3
	min_samples_split	2
	min_samples_leaf	1
XGBoost	n_estimators	100
	learning_rate	0.015
	max_depth	3
	subsample	0.9
	colsample_bytree	1
	booster	dart
	tree_method	hist
	sampling_method	uniform
	grow_policy	depthwise
	enable_categorical	true
LightGBM	n_estimators	1000
	learning_rate	0.001
	num_leaves	21
	max_depth	15
	reg_lambda	3
	subsample	0.95
	objective	binary
	metric	binary_logloss
CatBoost	iterations	50
	learning_rate	0.0005
	depth	8
	l_2 _leaf_reg	5
	bootstrap_type	Bernoulli
	subsample	0.9
	one_hot_max_size	4

Table A.11. Model hyperparameters