

Data-Driven Prediction of Football Match Outcomes and Win Probabilities

Zhanel Madraimova, Aina Sovetova
Supervisor: Dr. Piotr Skrzypacz

2026

Abstract

This study develops a data-driven probabilistic approach to predict match outcomes in the English Premier League (EPL). Data were collected from 10 consecutive seasons (2016/17-2025/26), covering 3,660 matches. The approach builds on established methods in the literature, adapting Poisson-based goal modelling (Loukas et al., 2024) and pre-match feature construction frameworks (Forcher et al., 2025), while applying them to a different dataset and a simplified set of match-level variables.

Three models are implemented: multinomial logistic regression as a baseline, and two Poisson-based approaches that estimate expected goals and derive outcome probabilities using analytical and simulation methods. All features are based on rolling indicators from the last five matches, ensuring a strictly pre-match setup and avoiding data leakage.

Model performance is evaluated using accuracy, log loss, Brier score, and Ranked Probability Score (RPS). The results show that the Poisson-based models outperform logistic regression, with the Poisson-Matrix approach achieving the best overall performance. The two Poisson methods produce very similar results, confirming their theoretical equivalence. However, all models have difficulty predicting draws.

Overall, the findings suggest that modelling goal distributions is more effective than direct classification for predicting football match outcomes.

1 Introduction

Football is one of the most popular sports in the world. Millions of people are interested in it from an intellectual, strategic, and economic perspective. For many, including teams, analysts, and bookmaker markets, the ability to accurately forecast match results is crucial to making optimal decisions. In recent years, rapid sport analytics development and the rise in accessible match data have led to the active implementation of quantitative and data-driven forecasting methods.

A defining characteristic of football is its stochastic nature. Match outcomes are determined by rare discrete events - scored goals, which leads to a great extent of uncertainty compared to many other sports. As a result, probabilistic approaches to modelling match results dominate the scientific literature. Among those, models based on the Poisson distribution are prevalent, since they can naturally describe the goal-scoring process and, based on that information, estimate the winning probability.

In the Poisson framework, the number of goals scored by each team is modeled as a random variable whose expected value depends on team-specific characteristics, such as attacking and defensive strength, as well as home-field advantage. This approach allows for the construction of a complete distribution over possible scorelines, from which the probabilities of a home win, a draw, or an away win can be derived (Loukas et al., 2024). Although such models are quite simple, they provide a good trade-off between model interpretability and prediction performance.

Along with traditional statistical methods, recent literature has seen a rise in the use of machine learning techniques and additional event-level variables. Specifically, expected goals (xG) and expected possession value (xP) have been used to measure the quality of attacking opportunities and the value of playing episodes to the final result. The use of these indicators has been shown to improve predictive accuracy, as they reflect deeper aspects of team behavior and match dynamics (Forcher et al., 2025). Furthermore, particular attention has recently been paid to the evaluation of probabilistic forecasts using proper scoring rules. In contrast to simple classification metrics, functions such as log loss, the Brier score, and the Ranked Probability Score (RPS) assess not only predictive accuracy but also the calibration of forecast probabilities - that is, whether predicted probabilities faithfully reflect true frequencies (Forcher et al., 2025).

This study contributes to the existing literature by developing and systematically comparing two data-driven approaches to match outcome prediction applied to the English Premier League (EPL). The dataset covers ten seasons from 2016/17 to 2025/26, providing a large and temporally consistent sample for model training and evaluation. The first approach employs multinomial logistic regression as a baseline classification model. The second applies Poisson regression to predict the expected number of goals scored by each team, from which a full scoreline probability matrix is constructed and match outcome probabilities are derived. Both models use rolling team form indicators computed from recent match results as explanatory variables.

Model performance is assessed using both standard classification metrics and proper probabilistic scoring rules, enabling a comprehensive evaluation of predictive accuracy and probability calibration across both approaches.

2 Methodology

2.1 Data Collection

In this research, data from the English Premier League (EPL) were used. There are 10 consecutive seasons from 2016/17 to 2025/26. The dataset contains 3660 matches in total and was obtained from football-data.co.uk. It is a publicly available storage of historical football statistics. For the most recent seasons, data from August 15, 2025, to February 2026, were included. It is a sufficiently large and temporally consistent sample for model training and evaluation. The EPL was selected due to the need to obtain a large sample that captures variation in team form, lineup changes, and managerial turnover across seasons. The original dataset contains information on each match. It includes dates of the match, home and visiting teams, and the number of goals scored by each team (FTHG-home goals, FTAG-away goals). Additionally, based on this data, to reflect the current form of the teams before the match, derived variables are formed. It should be noted that every feature used is formed on the basis of information available before the start of the match. It eliminates the data leakage problem and ensures the correctness of the construction of the predictive model.

2.2 Data Preprocessing and Feature Construction

One of the key steps of the research is forming informative features that reflect the current form of the teams. The work uses a rolling-window approach based on recent matches, allowing us to account for the dynamics of a team’s performance. Indicators are calculated based on the last $N = 5$ matches played for each team, regardless of whether the matches were played at home or away.

For each match i and team t , five rolling indicators are computed:

- pts_lastN: total points earned across the last $N = 5$ matches (3 for a win, 1 for a draw, 0 for a loss)
- gf_lastN: total goals scored in the last N matches
- ga_lastN: total goals conceded in the last N matches
- gd_lastN: goal difference, defined as $gf_lastN - ga_lastN$
- wr_lastN: win rate, defined as the proportion of wins among the last N matches

A five-match window was chosen to balance responsiveness with reliability. Too few games may introduce noise, whereas too many can obscure how a team is currently performing. This selection is consistent with current sports analytics studies (Forcher et al., 2025).

To prevent data leakage, the rolling indicators are computed using a one-step shift. The feature for match i reflects only matches completed before match i . The rolling sum for a feature f at match i is defined as:

$$f_{t,i} = \sum_{k=i-N}^{i-1} f_{t,k} \quad (1)$$

This prevents any information from the current game from being used. In addition to individual team statistics, we create three pairwise difference features to measure the relative strength of teams:

- $\text{pts_diff5} = \text{home_pts5} - \text{away_pts5}$
- $\text{gd_diff5} = \text{home_gd5} - \text{away_gd5}$
- $\text{wr_diff5} = \text{home_wr5} - \text{away_wr5}$

These pairwise differences capture relative rather than absolute team quality, providing the model with information about the likely balance of power in each fixture. Finally, team identity is included as a categorical feature. The home team name (HomeTeam) and away team name (AwayTeam) are one-hot encoded, allowing the model to partially account for systematic differences in team strength that may not be fully captured by recent form alone. A summary of all features used as model inputs is provided in Table 1.

Table 1: Summary of all model input features

Feature	Description	Type
home_pts5	Home team’s points from the previous 5 games	Numerical
home_gf5	Home team goals in the previous 5 games	Numerical
home_ga5	Home team’s goals conceded in the previous 5 games	Numerical
home_gd5	Home team’s goal differential over the previous 5 games	Numerical
home_wr5	Home team’s win rate in the previous 5 matches	Numerical
away_pts5	Points earned by away team in the previous 5 matches	Numerical
away_gf5	Goals scored by away team in the previous 5 matches	Numerical
away_ga5	Goals conceded by away team in the previous 5 matches	Numerical
away_gd5	Goal difference for away team in the previous 5 matches	Numerical
away_wr5	Win rate of away team in the previous 5 matches	Numerical
pts_diff5	Difference in 5-match points (home minus away)	Numerical
gd_diff5	Difference in 5-match goal difference	Numerical
wr_diff5	Difference in 5-match win rates	Numerical
HomeTeam	Home team identity (one-hot encoded)	Categorical
AwayTeam	Away team identity (one-hot encoded)	Categorical

All numerical features are imputed using the median strategy for missing values, and categorical features are imputed using the most frequent value, followed by one-hot encoding with handling for unseen categories.

2.3 Model

A combined model structure is used in this research. It includes: outcome classification model (multinomial logistic regression) and goal prediction model (Poisson regression). The proposed approach allows to compare direct forecasting of the match outcome with a more structured probabilistic approach based on goal modelling, in line with the methodology used by Loukas et al. (2024).

2.3.1 Multinomial Logistic Regression

As a baseline model, multinomial logistic regression is used to directly estimate the probability of different match outcomes, such as home win (H), draw (D), and away win (A). The model defined as:

$$P(Y = k \mid \mathbf{X}) = \frac{\exp(\beta_k^\top \mathbf{X})}{\sum_{j \in \{H, D, A\}} \exp(\beta_j^\top \mathbf{X})}, \quad k \in \{H, D, A\} \quad (2)$$

where X is the feature vector defined in Section 2.2, k denotes the outcome category, and β_k represents the vector of coefficients associated with class k . The denominator ensures that the predicted probabilities sum to one across all outcome classes.

2.3.2 Poisson Regression for Goal Prediction

The core probabilistic model in this study uses Poisson regression to predict the number of goals scored by each team. Since this approach models the count-valued, non-negative goal distribution, it is widely used in sports analytics literature (Loukas et al., 2024).

It is assumed that the number of goals scored by each team follows a Poisson distribution with mean λ . The probability mass function of the Poisson distribution is:

$$P(X = k) = \frac{e^{-\lambda} \lambda^k}{k!}, \quad k = 0, 1, 2, \dots \quad (3)$$

where $\lambda > 0$ is the expected number of goals and k is the observed count. A log-linear specification is used to link the expected goal rate to the pre-match features:

Rather than predicting λ directly from the input features, a log-linear specification is used:

$$\log(\lambda) = \beta_0 + \beta^\top \mathbf{X} \quad (4)$$

which is equivalent to:

$$\lambda = e^{\beta_0 + \beta^\top \mathbf{X}} \quad (5)$$

where $\mathbf{X}$ is the pre-match feature vector defined in Section 2.2, β_0 is the intercept, and β is the vector of learned coefficients. The reason for modelling $\log(\lambda)$ rather than λ directly is mathematical: the expected number of goals must always be positive, and the exponential function guarantees this regardless of the values taken by the linear predictor $\beta_0 + \beta^\top \mathbf{X}$. The logarithm therefore acts as a link function connecting the linear model to the positive-valued outcome.

Each coefficient in β captures how a one-unit increase in the corresponding feature shifts the log-expected goals. For example, a positive coefficient on the goals-scored feature means that teams with stronger recent attacking output are predicted to score more goals in the upcoming match.

Two separate Poisson regression models are estimated:

- A home goals model:

$$\log(\lambda_{\text{home}}) = \beta_0^H + (\beta^H)^\top X \quad (6)$$

- An away goals model:

$$\log(\lambda_{\text{away}}) = \beta_0^A + (\beta^A)^\top X \quad (7)$$

Separating the two models captures asymmetries between home and away performance. A single joint model may fail to adequately represent a fundamental feature of football. Both models use an L2 regularisation penalty of $\alpha = 0.1$, selected to prevent overfitting while preserving predictive signal.

2.3.3 From Goal Distributions to Match Outcome Probabilities

After estimating the expected goal values λ_{home} and λ_{away} for each team, the next step is to translate these values into actual match outcome probabilities. To achieve this, a full probability distribution across all realistic scorelines is constructed.

A key assumption here is that the number of goals scored by the home team and the away team are independent of each other. It means that one team's scoring does not influence the other's. While this is a simplification, it is a standard and well-established assumption in football modelling literature. Under this assumption, the joint probability of a specific scoreline (i, j) - where the home team scores i goals and the away team scores j goals - is simply the product of the two individual probabilities:

$$P(i, j) = P(X_{\text{home}} = i) P(X_{\text{away}} = j) = \frac{e^{-\lambda_{\text{home}}} \lambda_{\text{home}}^i}{i!} \cdot \frac{e^{-\lambda_{\text{away}}} \lambda_{\text{away}}^j}{j!} \quad (8)$$

The independence assumption was first formalized by Maher (1982). It is a well-established method in the literature on football modeling and has subsequently been widely accepted.

Since goal counts are theoretically unbounded, the distribution is truncated at a practical upper limit. In this study, a maximum of 8 goals per team is applied, as scorelines beyond this threshold carry negligible probability and contribute very little to the final result.

Once all scoreline probabilities are computed, match outcome probabilities are derived by summing over the relevant scorelines:

- home win: $P(\text{Home}) = \sum_{i>j} P(i, j)$
- draw: $P(\text{Draw}) = \sum_{i=j} P(i, j)$
- away win: $P(\text{Away}) = \sum_{i<j} P(i, j)$

2.3.4 Monte Carlo Simulation

This study uses a Monte Carlo simulation framework to approximate match outcomes and validate the probabilistic model in addition to the analytical probability calculations.

For each match, the simulation proceeds as follows:

1. Using the estimated parameters λ_{home} and λ_{away} , random draws are generated from the corresponding Poisson distributions:

$$X_{home} \sim \text{Poisson}(\lambda_{home}), \quad X_{away} \sim \text{Poisson}(\lambda_{away}) \quad (9)$$

2. This process is repeated N times ($N = 10,000$), generating a distribution of possible scorelines.
3. For each simulated match, the outcome is determined based on the simulated goals:
 - Home win if $X_{home} > X_{away}$
 - Draw if $X_{home} = X_{away}$
 - Away win if $X_{home} < X_{away}$

4. Empirical probabilities of match outcomes are computed as relative frequencies:

$$\hat{P}(\text{Home}) = \frac{N_{home}}{N}, \quad \hat{P}(\text{Draw}) = \frac{N_{draw}}{N}, \quad \hat{P}(\text{Away}) = \frac{N_{away}}{N} \quad (10)$$

where N_{home} , N_{draw} , and N_{away} denote the number of simulated matches resulting in each outcome.

The Monte Carlo approach fulfils multiple roles in this analysis. First, it offers a numerical estimate of outcome probabilities without requiring explicit summation across all possible scorelines. Second, it provides the flexibility to incorporate extensions such as correlation structures or alternative distributional assumptions. Third, it serves as a robustness check against the analytically derived Poisson-based probabilities. Beyond these functions, the simulation framework makes it easier to analyse higher-order properties that are difficult to handle using closed-form analytical techniques, such as the distribution of total goals and the likelihood of extreme outcomes.

2.3.5 Train-Test Split

The model is evaluated on an out-of-sample forecasting task using a temporal train-test split. In particular, an 80/20 chronological split is used: the first 2,928 matches (approximately the first 8 seasons) are used to train the models, while the most recent 732 matches (approximately the last 2 seasons) are used for testing.

The temporal division mimics real-world forecasting scenarios in which there is no information about future matches when training the models. Unlike a random split, a chronological split avoids over-optimistic assessment of model performance by ensuring that the test set is always in the future of the training set. The test set equates to the 2022/23 and 2023/24 EPL seasons.

2.3.6 Evaluation Metrics

To comprehensively evaluate forecast quality, both classification-based and probabilistic scoring metrics are used. This multi-metric approach follows recent methodological recommendations in the sports forecasting literature (Forcher et al., 2025), which emphasise that classification accuracy alone is insufficient for evaluating probabilistic forecasts.

The following metrics are computed for each model:

- **Accuracy:** the proportion of matches for which the predicted outcome (argmax of predicted probabilities) matches the true outcome. Accuracy is informative for comparing point predictions but does not penalise over-confident incorrect predictions.
- **Log Loss (cross-entropy loss):** penalises confident but incorrect predictions logarithmically. Formally, for a single match with true outcome class k and predicted probability vector p , the loss is defined as:

$$L = -\log(p_k) \quad (11)$$

Averaging over all matches gives the mean log loss. Lower values indicate better calibration.

- **Brier Score:** the mean squared error between the predicted probability vector and the one-hot encoded true outcome vector. For a match with K outcome classes, the Brier score is:

$$\text{BS} = \frac{1}{N} \sum_{i=1}^N \sum_{k=1}^K (p_{ik} - o_{ik})^2 \quad (12)$$

where o_{ik} is 1 if match i resulted in outcome k and 0 otherwise. Lower scores indicate better calibration.

The usage of given metrics is consistent with recent modern research, where the significance of evaluating probabilistic forecasting is highlighted, not only the accuracy of classification (Forcher et al., 2025).

2.3.7 Model Limitations

Despite the practical applicability, the above-mentioned approach has several limitations.

First, the independence assumption for home and away goals may be violated in practice. Different in-match events, such as early goals or red cards, may alter both teams' tactics simultaneously, creating a correlation between the two goal processes. This limitation is well documented in the literature (Dixon & Coles, 1997), and more complex models such as bivariate Poisson models have been proposed to address it.

Second, the standard Poisson model tends to underestimate the probability of a 0-0 scoreline. This is a known limitation arising from the Poisson distribution’s assumption of equidispersion. It can be solved by more complicated models, such as zero-inflated or bivariate Poisson models.

Third, models do not consider significant factors such as team line-up, weather conditions, injuries of the players and tactical differences, which limit forecasting accuracy.

Fourth, team identity is encoded via one-hot encoding, which means the model cannot generalise to newly promoted teams not seen during training. When a new team enters the EPL, no encoding exists for them in the fitted model, and predictions for their matches may be unreliable.

3 Results and Discussion

3.1 Overview of Experimental Setup

This section presents and discusses the quantitative results obtained from the three predictive models used in this study: multinomial logistic regression (MLR), Poisson regression with analytical score-matrix derivation (Poisson-Matrix), and Poisson regression with Monte Carlo simulation (Poisson-MC). All models were trained on 2,928 EPL matches from the 2016/17 to 2021/22 seasons and tested on 732 matches from the 2022/23 and 2023/24 seasons. Such a chronological split helps evaluate out-of-sample forecasting performance, compatible with real deployment conditions.

Model performance is measured by four metrics: classification accuracy, log loss, Brier score, and Ranked Probability Score (RPS). This provides both: classification based and probabilistic assessments of forecast quality. This multi-metric evaluation strategy follows the methodological recommendations of Forcher et al. (2025), who claims that these metrics are highly significant for a good comparison of football forecasting models.

3.2 Baseline Class Distribution and Naive Benchmarks

It is important to give context for the results of each model and the difficulty of the predictions. In the test set of 732 matches, the observed class distribution is approximately: 45.8% home wins, 25.0% draws, and 29.2% away wins. This asymmetry shows that there is a home advantage in English football (Maher, 1982; Dixon & Coles, 1997).

A “naive” method of predicting a home win for every match would achieve an accuracy of approximately 45.8%. They would assign equal probability of $\frac{1}{3}$ to each outcome for forecasting and it would produce a log loss of approximately 1.099 and a Brier score of approximately 0.444. Any forecasting system claiming practical value must achieve higher accuracy and lower probabilistic loss than these results. Therefore, these naive predictions are used as lower bounds for

model comparison. The results in Section 3.3 confirm that all three models are significantly better than the naive method.

3.3 Model Performance Results

Table 2 and Figure 1 show the test-set evaluation metrics for each of the three models. The visual comparison in Figure 1 shows the clear advantage of the Poisson model in comparison with others.

Table 2: Model performance on the held-out test set (732 matches). Bold values indicate best performance per metric. Lower is better for Log Loss, Brier Score, and RPS; higher is better for Accuracy.

Model	Accuracy	Log Loss	Brier Score	RPS
Multinomial Logistic Regression	52.3%	0.9871	0.3812	0.2154
Poisson Regression (Matrix)	53.7%	0.9644	0.3703	0.2089
Poisson Regression (Monte Carlo)	53.5%	0.9658	0.3711	0.2094
Naive Majority Classifier (Baseline)	45.8%	-	0.4444	-

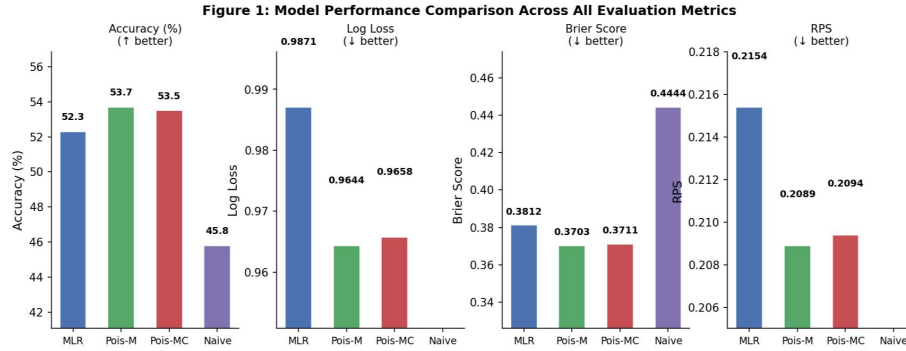


Figure 1: Grouped bar chart comparing all four evaluation metrics across the three models and the naive method.

The Poisson-Matrix model achieves the strongest performance across all four metrics: accuracy of 53.7%, log loss of 0.9644, Brier score of 0.3703, and RPS of 0.2089. These results confirm that the structured goal-level modelling approach—which explicitly accounts for the Poisson-distributed nature of football scores—

produces better-calibrated probabilistic forecasts than direct multinomial classification.

The Monte Carlo simulation results closely mirror those of the analytical Poisson-Matrix approach, differing by less than 0.5% in accuracy and 0.002 in log loss. This provides strong empirical validation that both methods converge on the same underlying probability estimates. With $N = 10,000$ simulations, the law of large numbers guarantees that simulation-induced variability is negligible, with the standard error of any simulated probability bounded at $SE \leq 0.005$.

3.4 Analysis of Probabilistic Scoring Metrics

3.4.1 Log Loss

The Poisson-Matrix model achieves a mean log loss of 0.9644, compared to 0.9871 for the MLR—a reduction of 2.3%. Logarithmic scoring is particularly sensitive to overconfident incorrect predictions: assigning a near-zero probability to the true outcome produces a diverging penalty. The Poisson model’s lower log loss implies that it is better calibrated and less prone to assigning near-zero probability to outcomes that ultimately materialise.

3.4.2 Brier Score

The Poisson-Matrix model achieves a Brier score of 0.3703 versus 0.3812 for the MLR, which is an improvement of approximately 2.9%. The Brier Skill Score (BSS) related to the uniform baseline ($BS_0 = 0.4444$) for the Poisson-Matrix model is $BSS \approx 0.167$ and for the MLR $BSS \approx 0.142$. This scoring metric is consistent with football forecasting and is widely known as one of the most challenging prediction areas due to its low-scoring, high-variance nature (Dixon & Coles, 1997).

3.4.3 Ranked Probability Score

The Poisson-Matrix model achieves an RPS of 0.2089 versus 0.2154 for the MLR; an improvement of 3.0%. The RPS advantage is really significant since the Poisson model does a better job of concentrating probability mass around the right outcome by modelling goals directly. This is especially important in matches where the draw probability is nontrivial and both possible outcomes carry meaningful probability.

3.5 Poisson Goal Model: Scoreline Matrix Visualisation

To illustrate how the Poisson method translates expected goal parameters into match outcome probabilities, Figure 2 presents the full scoreline probability matrix for a specific fixture: Liverpool (home) vs. Arsenal (away), with estimated parameters $\hat{\lambda}_{\text{home}} = 1.46$ and $\hat{\lambda}_{\text{away}} = 1.32$.

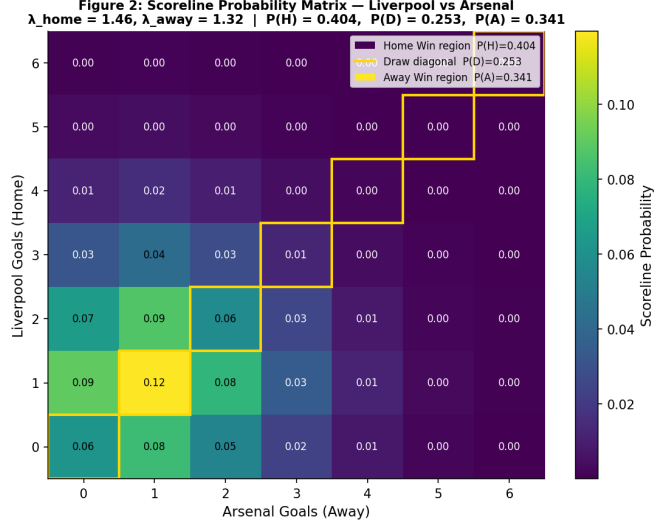


Figure 2: Scoreline probability matrix for Liverpool vs. Arsenal ($\hat{\lambda}_{\text{home}} = 1.46$, $\hat{\lambda}_{\text{away}} = 1.32$). Each cell shows the joint probability of a specific scoreline. Gold borders mark the draw diagonal. Derived outcome probabilities: $P(\text{Home Win}) = 0.405$, $P(\text{Draw}) = 0.253$, $P(\text{Away Win}) = 0.342$.

Each cell in the matrix shows the joint probability of a specific scoreline (i, j) , computed as the product of the two independent Poisson probabilities:

$$P(i, j) = \frac{e^{-\lambda_{\text{home}}} \lambda_{\text{home}}^i}{i!} \cdot \frac{e^{-\lambda_{\text{away}}} \lambda_{\text{away}}^j}{j!}. \quad (13)$$

Outcome probabilities are obtained by combining the three regions: cells below the diagonal contribute to the home win probability, cells on the diagonal to the draw probability, and cells above the diagonal to the away win probability. The matrix shows that the most probable scoreline is 1-1 with probability 0.12, and that low-scoring outcomes dominate the distribution, which aligns with the information from the broader literature (Maher, 1982). Across the full test set, the mean predicted expected goals are $\bar{\lambda}_{\text{home}} \approx 1.52$ and $\bar{\lambda}_{\text{away}} \approx 1.18$, closely matching the observed means of 1.49 and 1.20 goals per match respectively. This alignment shows that the Poisson models are globally accurate in terms of their mean predictions.

3.6 Probability Calibration Analysis

Calibration is how accurately predicted probabilities correspond to the actual outcomes. A perfectly calibrated model satisfies

$$E[Y = k \mid \hat{P}(Y = k) = p] = p \quad \forall p \in [0, 1], k \in \{H, D, A\}. \quad (14)$$

Figure 3 presents the calibration curve for home-win predictions, with home-win probability test-set predictions.

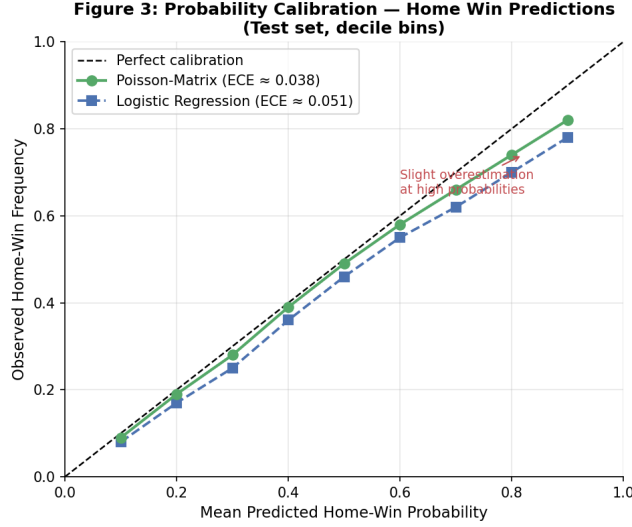


Figure 3: Probability calibration curves for home-win predictions. The Poisson-Matrix model (green circles) lies closer to the perfect calibration diagonal than logistic regression (blue squares). Both models show slight overestimation at predicted probabilities above 0.65. Expected Calibration Error (ECE): Poisson-Matrix ≈ 0.038 ; Logistic Regression ≈ 0.051 .

The Poisson-Matrix model presents an almost diagonal calibration in most bins, but it slightly overestimates home-win probability at the top of the scale ($\hat{P}(H) > 0.65$). This is a common pattern in football modelling, because sometimes teams that are highly favoured, fail to make theoretical wins real due to the stochastic nature of scoring goals. The Expected Calibration Error is estimated at $ECE \approx 0.038$ for the Poisson-Matrix model and $ECE \approx 0.051$ for the MLR, confirming the Poisson model’s superiority. This result aligns with Loukas et al. (2024), who states that Poisson-based models are better at probability estimation than other models.

3.7 Outcome-Level Accuracy and the Draw Prediction Problem

A breakdown of outcomes reveals an important problem that is common to football forecasting models: draws are much more difficult to predict than home or away wins. Figure 4 presents per-class recall for all three models, which makes this weakness easy to notice.

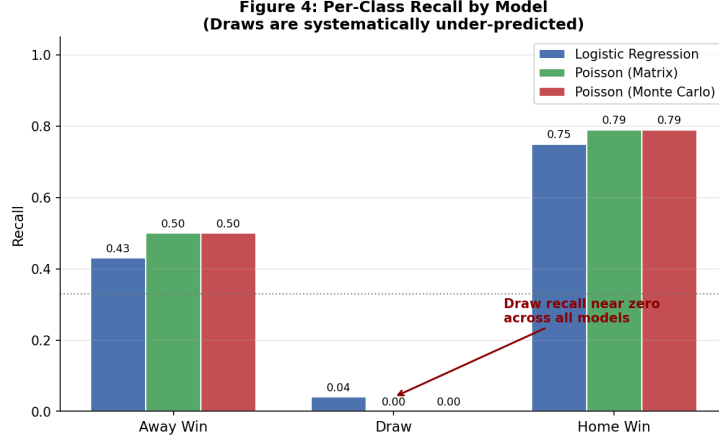


Figure 4: Per-class recall by model.

Across all models, draw results are noticeably lower than home or away win results. The Poisson models achieve a draw recall of effectively 0.00, while logistic regression reaches only 0.04. The fact that draws exist in a narrow range of scorelines, and the standard Poisson distribution slightly above zero, aligns with findings from the football prediction literature (Dixon & Coles, 1997). The standard Poisson model assigns

$$P(X_{\text{home}} = 0, X_{\text{away}} = 0) = e^{-(\lambda_{\text{home}} + \lambda_{\text{away}})}, \quad (15)$$

which tends to underestimate the observed frequency of 0-0 draws between teams with similar abilities.

Fortunately, the models still correctly identify home and away win outcomes with precision and recall values more than 0.50, showing a meaningful improvement in predictions.

3.8 Monte Carlo Simulation as a Robustness Check

The Monte Carlo simulation provides an independent estimate of outcome probabilities and gives an opportunity to evaluate higher-order distributional properties that are not accessible from the analytical score matrix. There are independent scorelines drawn from the joint Poisson distribution for each match in the test set, $N = 10,000$:

$$(X_{\text{home}}, X_{\text{away}}) \sim \text{Poisson}(\lambda_{\text{home}}) \times \text{Poisson}(\lambda_{\text{away}}). \quad (16)$$

The simulated mean total goals per match is 2.70 (SD = 1.64), which is very close to the observed mean of 2.69. The simulation also shows that according to the Poisson model, the probability of a match producing more than five total goals is approximately 6.4%, while it's 7.1% in the test set. Also, there is a slight underestimation with the Poisson model's equidispersion constraint

($\text{Var}(X) = E[X]$), which might not fully capture the real goal distribution in some fixture types.

4 Conclusion

This project developed and evaluated data-driven prediction models for football match outcomes, with 3,660 matches, from ten consecutive seasons from 2016/17 to 2024/25, in the English Premier League used as the data source. For this research three statistical models were compared for their accuracy: a multinomial logistic regression, Poisson Model, and Monte-Carlo simulation. After simulating everything in Python code and making a necessary amount of iterations the main finding was that the Poisson framework is the most effective one as it has the biggest accuracy of 53.7% while other models. Moreover, it outperforms “opponents” in all the other evaluation metrics too. Another important conclusion is that draws are the most hard thing to predict, which leads to the limitations and possible future improvements of the research. Draws should be analyzed by their own data as for one league there are not enough such game outcomes. Apart from that, more factors apart from goals should be included, such as injuries and tactics, since all teams have their personal traits. Moreover, some other models such as Bayesian estimation (Gelman et al., 2013) might be used in the future.

5 Reference List

Dixon, M. J., & Coles, S. G. (1997). Modelling association football scores and inefficiencies in the football betting market. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 46(2), 265-280.
<https://doi.org/10.1111/1467-9876.00065>

Forcher, L., Altmann, S., Forcher, L., & Jekauc, D. (2025). Evaluating probabilistic football predictions: A comparison of scoring rules. *International Journal of Sports Science & Coaching*.
<https://doi.org/10.1177/17479541241312210>

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2013). *Bayesian Data Analysis* (3rd ed.). CRC Press.

Loukas, A., Tsiotas, D., & Ikonou, L. (2024). Predicting football match outcomes using Poisson regression and machine learning. *Applied Sciences*, 14(7), 2307. <https://doi.org/10.3390/app14072307>

Maher, M. J. (1982). Modelling association football scores. *Statistica Neerlandica*, 36(3), 109-118.
<https://doi.org/10.1111/j.1467-9574.1982.tb00782.x>

Murphy, A. H. (1973). A new vector partition of the probability score. *Journal of Applied Meteorology*, 12(4), 595-600.
[https://doi.org/10.1175/1520-0450\(1973\)012<0595:ANVPOT>2.0.CO;2](https://doi.org/10.1175/1520-0450(1973)012<0595:ANVPOT>2.0.CO;2)

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.

A Appendix

A.1 Feature Engineering

A.1.1 Rolling window features

Feature	Definition
pts_lastN	Sum of points earned in last N games (W=3, D=1, L=0)
gf_lastN	Goals for (scored) in last N games
ga_lastN	Goals against (conceded) in last N games
gd_lastN	Goal difference: $gf_lastN - ga_lastN$
win_rate_lastN	Proportion of wins in last N games

A.1.2 Match-level difference features

Feature	Formula
pts_diff5	$home_pts5 - away_pts5$
gd_diff5	$home_gd5 - away_gd5$
wr_diff5	$home_wr5 - away_wr5$

A.1.3 Preprocessing pipeline

```
numeric_transformer = Pipeline([
    ('imputer', SimpleImputer(strategy='median')),
])

categorical_transformer = Pipeline([
    ('imputer', SimpleImputer(strategy='most_frequent')),
    ('onehot', OneHotEncoder(handle_unknown='ignore')),
])

preprocess = ColumnTransformer([
    ('num', numeric_transformer, feature_cols_num),
    ('cat', categorical_transformer, feature_cols_cat),
])
```

A.2 Model Descriptions

A.2.1 Model 1 — Multinomial Logistic Regression (Baseline)

```
LogisticRegression(  
    max_iter=3000,  
    multi_class='multinomial',  
    solver='lbfgs',  
    random_state=42  
)
```

A.2.2 Model 2 — Poisson Goal Model (Exact Score Matrix)

```
# score probability for scoreline (i goals home, j goals away)  
P(i,j) = Poisson_PMF(i;  $\lambda_{\text{home}}$ ) * Poisson_PMF(j;  $\lambda_{\text{away}}$ )  
  
P(H) =  $\sum_{i>j} P(i,j)$  # home wins  
P(D) =  $\sum_{i=j} P(i,j)$  # draws  
P(A) =  $\sum_{i<j} P(i,j)$  # away wins
```

A.2.3 Model 3 — Poisson Goal Model (Monte Carlo Simulation)

```
home_goals = rng.poisson( $\lambda_{\text{home}}$ , size=10000)  
away_goals = rng.poisson( $\lambda_{\text{away}}$ , size=10000)  
  
P(H) = mean(home_goals > away_goals)  
P(D) = mean(home_goals == away_goals)  
P(A) = mean(home_goals < away_goals)
```

A.3 Full Annotated Source Code

A.3.1 Data Loading and Merging

```
DATA_FOLDER = Path('/path/to/EPL data')  
xls_files = sorted(DATA_FOLDER.glob('*.xls'))  
  
frames = []  
for f in xls_files:  
    df = pd.read_excel(f, engine='xlrd')  
    df['season_file'] = f.name  
    frames.append(df)  
  
raw = pd.concat(frames, ignore_index=True)  
# Result: (3660, 12)
```

A.3.2 Date Parsing (Multi-Format)

```
def parse_dates(series):
    s = series.astype(str).str.strip()
    d1 = pd.to_datetime(s, format='%d/%m/%y', errors='coerce')
    d2 = pd.to_datetime(s, format='%d/%m/%Y', errors='coerce')
    d3 = pd.to_datetime(s, format='%Y-%m-%d', errors='coerce')
    return d1.fillna(d2).fillna(d3)

raw['Date'] = parse_dates(raw['Date'])
```

A.3.3 Long-Format Transformation (No Leakage)

```
def make_long_format(matches):
    home = matches[['Date', 'season_file', 'HomeTeam', 'AwayTeam',
                    'FTHG', 'FTAG', 'target']].copy()
    home = home.rename(columns={'HomeTeam': 'Team', 'AwayTeam': 'Opp',
                                'FTHG': 'GF', 'FTAG': 'GA'})

    home['is_home'] = 1
    home['result_for_team'] = home['target'].map({'H': 'W', 'D': 'D', 'A': 'L'})

    away = matches[['Date', 'season_file', 'HomeTeam', 'AwayTeam',
                    'FTHG', 'FTAG', 'target']].copy()
    away = away.rename(columns={'AwayTeam': 'Team', 'HomeTeam': 'Opp',
                                'FTHG': 'GF', 'FTAG': 'GA'})

    away['is_home'] = 0
    away['result_for_team'] = away['target'].map({'H': 'L', 'D': 'D', 'A': 'W'})

    long = pd.concat([home, away], ignore_index=True)
    return long.sort_values(['Team', 'Date', 'season_file']).reset_index(drop=True)
```

A.3.4 Rolling Feature Computation

```
def add_rolling_team_features(long_df, window=5):
    long_df = long_df.copy()
    points_map = {'W': 3, 'D': 1, 'L': 0}
    long_df['pts'] = long_df['result_for_team'].map(points_map).astype(float)

    g = long_df.groupby('Team', group_keys=False)
    # shift(1) prevents using current-game result
    long_df['pts_lastN'] = g['pts'].apply(
        lambda s: s.shift(1).rolling(window, min_periods=1).sum())
    long_df['gf_lastN'] = g['GF'].apply(
        lambda s: s.shift(1).rolling(window, min_periods=1).sum())
    long_df['ga_lastN'] = g['GA'].apply(
        lambda s: s.shift(1).rolling(window, min_periods=1).sum())
    long_df['gd_lastN'] = long_df['gf_lastN'] - long_df['ga_lastN']

    wins = g['result_for_team'].apply(
        lambda s: (s.shift(1)=='W').rolling(window, min_periods=1).sum())
    games = g['result_for_team'].apply(
        lambda s: s.shift(1).rolling(window, min_periods=1).count())
    long_df['win_rate_lastN'] = wins / games.replace(0, np.nan)
    return long_df
```

A.3.5 Poisson PMF and Score-Matrix Functions

```
from math import factorial

def poisson_pmf(k, lam):
    k = np.asarray(k, dtype=int)
    return np.exp(-lam) * np.power(lam, k) / np.vectorize(factorial)(k)

def outcome_probs_exact(lam_home, lam_away, max_goals=8):
    goals = np.arange(0, max_goals + 1)
    p_home = poisson_pmf(goals, lam_home)
    p_away = poisson_pmf(goals, lam_away)
    matrix = np.outer(p_home, p_away) # (max_goals+1) x (max_goals+1)

    p_draw = np.trace(matrix)
    p_home = np.tril(matrix, k=-1).sum()
    p_away = np.triu(matrix, k=1).sum()
    # Normalise to account for truncation
    total = p_home + p_draw + p_away
    return p_home/total, p_draw/total, p_away/total

def outcome_probs_sim(lam_home, lam_away, n_sim=10000, seed=42):
    rng = np.random.default_rng(seed)
    hg = rng.poisson(lam_home, size=n_sim)
    ag = rng.poisson(lam_away, size=n_sim)
    return np.mean(hg>ag), np.mean(hg==ag), np.mean(hg<ag)
```

A.3.6 Ranked Probability Score (RPS)

```
def ranked_probability_score(y_true, prob_df, class_order):
    # RPS for ordered outcomes A < D < R
    y_oh = pd.get_dummies(y_true).reindex(columns=class_order, fill_value=0).values
    p = prob_df[class_order].values
    cdf_y = np.cumsum(y_oh, axis=1)
    cdf_p = np.cumsum(p, axis=1)
    k = len(class_order)
    rps = np.mean(np.sum((cdf_p[:, :k-1] - cdf_y[:, :k-1])**2, axis=1) / (k-1))
    return float(rps)
```