Premier League Match Outcome Prediction: Form-Based Features and Model Benchmarking

Yazid Abaroudi
HEC Lausanne
yazid.abaroudi@unil.ch

2026

Abstract

This capstone project investigates whether simple *form-based* features can predict the result of a Premier League match. Historical data from the 2018/19–2023/24 seasons were used to compute rolling averages of points and goal difference for the home and away sides. Several classification algorithms—K-nearest neighbours (KNN), logistic regression, random forests and gradient boosting—were trained on matches up to 2023/24 and evaluated on the full 2024/25 season. A naïve baseline that always predicts a home win provides a reference. The best classifier (KNN) achieved an accuracy of 47.4 %, a modest improvement over the 40.8 % baseline. We analyse why draws remain particularly difficult to predict, compare the form-based approach against richer models from the literature and discuss limitations arising from the absence of variables such as expected goals, squad market value and betting odds. The report concludes with suggestions for future work and an assessment of the maintainability of the codebase.

Keywords: football analytics, match prediction, class imbalance, machine learning, Premier League

Contents

1 Introduction

Football prediction offers commercial and strategic value, but the sport is low-scoring and inherently unpredictable. Even experts rarely exceed a 55 % success rate when converting bookmaker odds into win/draw/lose probabilities [1]. The multi-class nature of match outcomes (home win, draw or away win), the decisive impact of single goals and the prevalence of draws create modelling challenges.

This project asks whether rolling averages of points and goal difference—the simplest expressions of team form—provide enough signal to classify Premier League match outcomes. By focusing on a minimal feature set we isolate the intrinsic predictability of match outcomes and quantify the added value of richer variables in future work.

The remainder of this report reviews related work, describes the data and methods, presents results, discusses findings in the context of existing literature, and concludes with suggestions for future research.

2 Research Question and Literature Review

2.1 Background and related work

Predicting football outcomes has been studied from three main perspectives. *Statistical models* treat goals scored by each team as independent Poisson processes; the classic Dixon–Coles model improves the basic Poisson approach by correcting its systematic underestimation of draws and introducing a time-decay parameter that gives more weight to recent matches [2]. *Rating systems* such as Elo assign each team an underlying strength that is updated after every match; these ratings, combined with a home-advantage parameter and simple adjustments for attack and defence, have been used to produce probabilistic forecasts. *Machine-learning models* combine a wide array of features—expected goals, shot locations, player quality, injuries, betting odds and even natural language sentiment—to train classifiers or regression models. Recent research shows that even with access to rich features and advanced algorithms the maximum pre-match accuracy rarely exceeds 55 % [1].

Expected goals (xG) models estimate the quality of each shot based on location, angle and defensive pressure [3]. When aggregated over a match, xG provides a proxy for how many goals a team *should* score and has become the gold standard for assessing team performance. A bachelor's thesis by Tiippana compared multiple models (xG, shots-on-target, Elo and betting odds) across Europe's top leagues and found that none could accurately predict individual match outcomes; the Poisson-distributed xG model performed best but still exhibited a systematic bias toward predicting too many draws [3]. Other studies achieve around 55 % accuracy by combining player ratings, Elo features and betting odds [4]. These results highlight both the ceiling of predictive accuracy and the importance of comprehensive features.

2.2 Contribution relative to existing work

Compared with the literature, our project deliberately adopts a minimalist approach: we restrict ourselves to form-based features—rolling averages of points and goal difference—and evaluate a suite of off-the-shelf classifiers. The aim is not to compete with state-of-the-art models but to establish a clear baseline and understand where the predictive signal resides. By holding back advanced features we can pinpoint which aspects of the task are inherently hard (e.g. predicting draws) and which can be improved by adding richer data. Throughout the report we benchmark our results against published studies to contextualise our findings.

3 Methodology

3.1 Data description

The dataset originates from the publicly available repository on `https://www.football-data.co.uk`, which provides match outcomes and scores for major European leagues. We collected Premier League results from the 2018/19 through 2023/24 seasons as the training set and reserved the 2024/25 season for evaluation. Each record lists the teams, final score and result (H, D or A); draws constitute about a quarter of matches, creating a class imbalance.

To capture team form we engineered four features: rolling averages of points and goal difference for the home and away teams, computed over a five-match window. These features smooth short-term fluctuations while reflecting recent performance and use only past matches to avoid data leakage. The target variable has three classes (H, D and A).

3.2 Models and algorithms

We evaluated four classification algorithms in addition to a naive baseline that always predicts the home side to win:

1. **Baseline (home win)**: returns class H for every match. It reflects the empirical observation that home wins occur most frequently in the Premier League.

2. **Logistic Regression (multinomial)**: a linear classifier that estimates class probabilities via a softmax function. Model complexity is $\mathcal{O}(n\,p)$, where n is the number of observations and p the number of features.

3. ***K*-Nearest Neighbours (KNN)**: classifies a match by majority vote among the k closest matches in the training set, using Euclidean distance in feature space. At prediction time the complexity is $\mathcal{O}(n\,p)$ per query; although costly for large n, our dataset (about 2 660 training matches) made KNN tractable.

4. **Random Forest**: an ensemble of decision trees trained on bootstrapped samples with random feature subsets. This non-parametric model can capture non-linear interactions but requires careful tuning to avoid overfitting. Complexity grows with the number and depth of trees; we used 200 trees with maximum depth unconstrained.

5. **Gradient Boosting (XGBoost)**: sequentially fits shallow trees to the residuals of previous trees. Boosting often outperforms bagging on tabular data, though it is sensitive to hyperparameters such as learning rate and number of estimators. We used 300 estimators with a learning rate of 0.1.

Hyperparameters were selected via manual experimentation rather than exhaustive search, reflecting the exploratory nature of the project. Models were trained on matches up to the end of the 2023/24 season and evaluated on the held-out 2024/25 season. Accuracy (proportion of correct predictions) served as the primary metric. To compare models against the baseline we computed two-sided McNemar tests and Wilson 95 % confidence intervals on the difference in accuracy.

4 Implementation, Code Structure and Reproducibility

The project repository follows a modular design. Dedicated scripts handle data loading, feature engineering, model training and evaluation, while a simple command-line interface orchestrates the workflow. Reproducibility is ensured through fixed random seeds and clear documentation: the repository's README explains how to set up the environment, download the raw data and reproduce the results. This structure makes the codebase easy to extend—adding features such as expected goals or Elo ratings requires only minor modifications to the feature engineering module—while maintaining version control and transparency.

5 Results

5.1 Model performance on the 2024/25 season

Table 1 summarises the accuracy achieved by each model on the held-out 2024/25 season. The baseline accuracy corresponds to the proportion of matches won by the home team (40.8 %). The KNN and gradient boosting classifiers achieved the highest accuracies, improving upon the baseline by roughly six percentage points. Logistic regression offered a modest improvement, while the random forest failed to outperform the naive predictor. The p-values reflect the significance of improvements relative to the baseline.

Table 1: Predictive accuracy on the 2024/25 season. Confidence intervals are Wilson 95 % intervals; p-values are two-sided McNemar tests versus the baseline.

Model	Accuracy	95 % CI	p-value vs baseline
Baseline (Home Win)	0.408	$[0.360, 0.458]$	–
KNN ($k = 23$)	0.474	$[0.424, 0.524]$	0.034
Gradient Boosting	0.471	$[0.421, 0.521]$	0.022
Logistic Regression	0.458	$[0.408, 0.508]$	0.151
Random Forest	0.413	$[0.365, 0.463]$	0.936

The Wilson 95 % confidence intervals indicate that performance differences are modest. The intervals for KNN and gradient boosting barely exceed those of the baseline, and their improvements over the naive home-win predictor are statistically significant at the 5 % level (two-sided McNemar tests: p=0.034 and p=0.022, respectively). Logistic regression's improvement is not significant (p=0.151), and the random forest is virtually indistinguishable from the baseline (p=0.936).

Figure 1 visualises the logistic regression coefficients. Positive weights indicate that higher form values increase the probability of a home win, while negative weights favour draws or away wins. The magnitude of the coefficients shows that the home team's form variables contribute slightly more to the decision than the away team's.

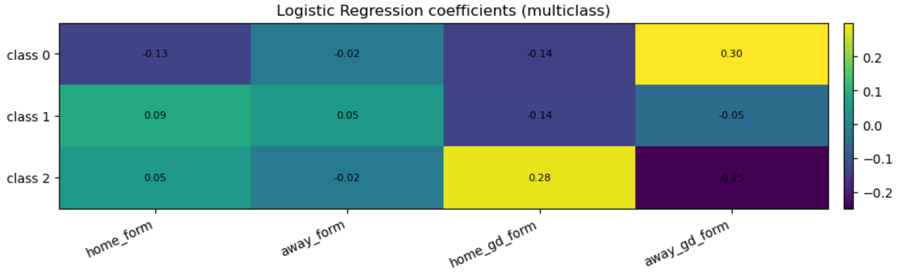

Figure 1: Logistic regression coefficients for the form features. Error bars denote standard errors estimated from the multinomial logistic model. The plot illustrates that home form and goal difference carry positive weight, whereas away form has a negative influence on the probability of a home win.

5.2 League table reconstruction

To understand the practical impact of prediction errors we reconstructed the 2024/25 league table by awarding three points for each predicted win, one point for a predicted draw and zero for a predicted loss. The KNN and gradient boosting models produced inflated win totals and scarcely predicted any draws. As a result, their predicted league tables differ markedly from the actual standings: the top teams reach around 90 points (as seen in the predicted table) and draws are almost absent. This inflation of win totals and underestimation of draws highlights how underestimating draws distorts cumulative performance.

Similarly, predictions for the first fourteen matchweeks of the 2025/26 season exhibit the same distortion: the predicted league table places top teams on roughly 37 points with almost no draws, whereas the actual standings at that stage show lower point totals and several drawn matches. This reinforces the conclusion that under-predicting draws inflates win counts across seasons.

6 Discussion

6.1 Why do form-based models struggle?

The moderate accuracies in Table 1 underscore the limited predictive power of rolling averages. While wins and losses exhibit some dependence on recent performance, draws occur for varied reasons: tactical stalemates, defensive solidity, weather conditions or random events. Because our features do not encode shot quality, possession, squad strength or psychological factors, the models cannot distinguish between an evenly balanced match that ends 1–1 and a mismatch that finishes 0–0 due to luck. The class imbalance further biases classifiers towards predicting home wins; without explicit handling of the minority draw class, even advanced algorithms under-predict draws [1].

Draws are not a stable class; they typically arise when teams of similar strength cancel each other out and occur in only about one quarter of matches, making them relatively rare and noisy. Gradient boosting and KNN models perform best—gradient boosting achieves the highest accuracy (0.471) with KNN close behind (0.474)—because they can capture non-linear relationships in the feature space. However, their improvements are modest: both achieve accuracies below 50 % and fail to correct the under-prediction of draws. The random forest's poor performance (0.413) suggests that with only four features the ensemble overfits noise in the training data; logistic regression's linear decision boundaries capture some trends but cannot model interactions.

6.2 Comparison with state-of-the-art models

Studies that incorporate richer features routinely achieve accuracies around 55 % [1]. For example, Muñoz *et al.* used player ratings from the EA FIFA video game and achieved about 55 % accuracy in predicting win/draw/lose outcomes across multiple tournaments [4]. Such models leverage player quality and team strength rather than recent results. The eXplainable Machine Learning study by Ren and Susnjak decomposed matches into categories of predictive difficulty using the Kelly index and combined Elo ratings and betting odds; they report that even the best pre-match forecasts rarely exceed 55 % accuracy [1]. Expected goals models, which estimate the probability of each shot resulting in a goal, provide valuable insight but do not systematically predict results; a comparative study found that the Poisson-distributed xG model still biases towards too many draws and cannot improve accuracy by adding a Poisson distribution [3].

Relative to these studies our form-based models underperform because they omit critical variables. Expected goals incorporate shot location and context; Elo ratings summarise long-term team strength; betting odds aggregate collective wisdom and market information. The absence of such features limits our models' ability to distinguish draws from home or away wins, leading to inflated win totals and unrealistic league tables. Nevertheless, our results establish a transparent baseline: with only rolling averages of points and goal difference one can achieve about 47 % accuracy, a small but significant improvement over a naive home-win predictor. The gap to the 55 % ceiling quantifies the value of richer data.

6.3 Surprising findings

Two observations merit discussion. First, the random forest failed to outperform the baseline. Ensemble methods typically excel on tabular data, but with only four continuous features the decision trees had little structure to exploit and may have overfit the training data. Second, the KNN and gradient boosting models outperformed logistic regression, suggesting that even with a linear relationship between features and outcome there are local patterns that a nearest-neighbour rule or boosting can leverage. However, the improvements are small, underscoring the fundamental difficulty of the task.

6.4 Limitations

This study has several limitations. The feature set is intentionally narrow and does not include key variables such as expected goals, shot counts, squad value, injuries, bookmaker odds or contextual factors like weather and travel. Our evaluation uses only one season (2024/25) as the test set; cross-validation across multiple seasons would yield more robust estimates. Hyperparameters were chosen manually; systematic tuning might improve performance marginally. Finally, the class imbalance was not explicitly addressed; techniques such as weighted loss functions or resampling could mitigate the bias towards home wins.

7 Conclusion and Future Work

7.1 Summary of findings

This capstone project set out to determine whether simple form-based features could predict Premier League match outcomes. We computed rolling averages of points and goal difference for home and away teams and trained four classifiers on data from 2018/19–2023/24. Evaluation on the 2024/25 season showed that the KNN and gradient boosting models achieved accuracies of 0.474 and 0.471, respectively, modestly outperforming a naive home-win predictor (0.408). Logistic regression offered a smaller improvement (0.458), while random forests failed to beat the baseline (0.413). All models struggled to predict draws, leading to unrealistic reconstructed league tables.

7.2 Recommendations

To approach the 55 % accuracy ceiling reported in the literature [1], future work should incorporate richer features. Expected goals provide a granular measure of shot quality and have been shown to improve explanatory power [3]. Elo ratings, squad market values and betting odds capture long-term team strength and market expectations. Player-level data (fitness, position, age) can further refine predictions [4]. Models such as the Dixon–Coles Poisson regression adjust for low-scoring matches and correct the underestimation of draws [2]. Combining these variables with modern machine-learning techniques (e.g. gradient boosting or neural networks) could yield accuracies approaching the theoretical limit.

From a methodological perspective, cross-validation across seasons, systematic hyperparameter tuning and calibration metrics (such as Brier score or log loss) would provide deeper insight into model performance. Handling class imbalance explicitly and exploring ensemble methods that combine generative (Poisson) and discriminative (machine learning) approaches could enhance draw prediction. Finally, open-sourcing reproducible code and documenting datasets and pipelines are essential for advancing research in football analytics.

References

1. Y. Ren and T. Susnjak, *Predicting Football Match Outcomes with eXplainable Machine Learning and the Kelly Index*. This study reports that even with rich features and sophisticated algorithms pre-match prediction accuracies seldom exceed 55 % [1].

2. D. Sheehan, "Predicting Football Results With Statistical Modelling: Dixon–Coles and Time–Weighting," blog post, 2018. The Dixon–Coles model introduces a correction for the Poisson model's underestimation of draws and applies time weighting to give more importance to recent matches [2].

3. T. Tiippana, "How Accurately Does the Expected Goals Model Reflect Goalscoring and Success in Football?" Bachelor's thesis, Aalto University, 2020. The thesis finds that expected goals models provide valuable insight but cannot systematically predict match outcomes and tend to overpredict draws [3].

4. O. R. Muñoz Gómez, I. C. Pimentel and L. F. G. Pacheco, "Prediction of Football Match Results Using Virtual Data," preprint, 2023. By using player ratings as features, the authors achieve about 55 % accuracy and highlight the importance of player-level data [4].

A Additional Figures

Figure 2 reproduces the model comparison plot used in the main report.

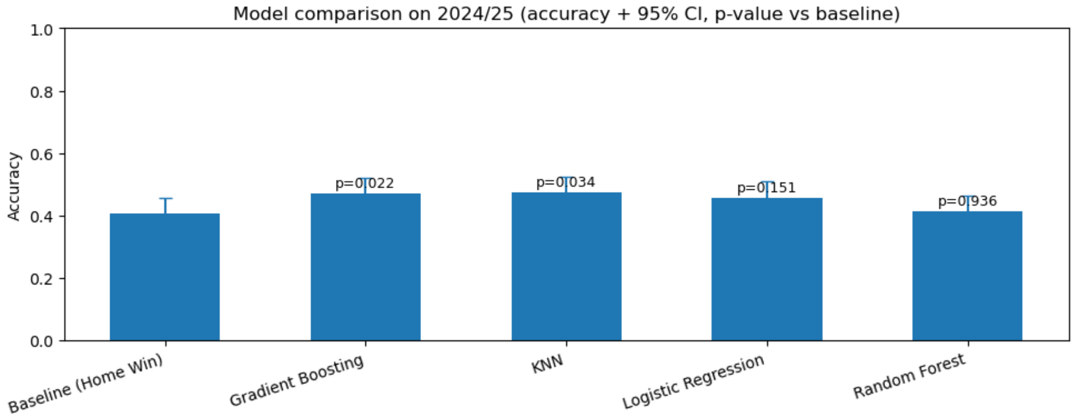

Figure 2: Comparison of baseline and machine-learning models on the 2024/25 season. Error bars represent Wilson 95 % confidence intervals and p-values denote the significance of each model's improvement over the baseline.

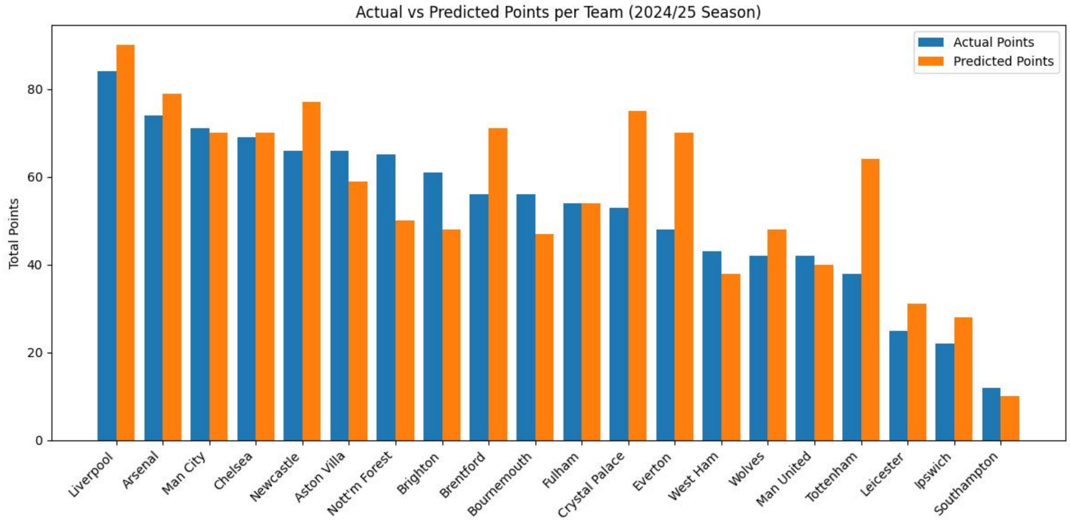

Figure 3: Actual versus predicted points for the top 10 teams in the 2024/25 season. The bar chart compares the actual league points with those implied by the form-based model predictions, highlighting inflation of win totals and scarcity of draws.

Comment. This figure highlights that predicted point totals exceed actual values for most teams, illustrating how under-prediction of draws by the form-based models inflates wins and distorts the reconstructed table.

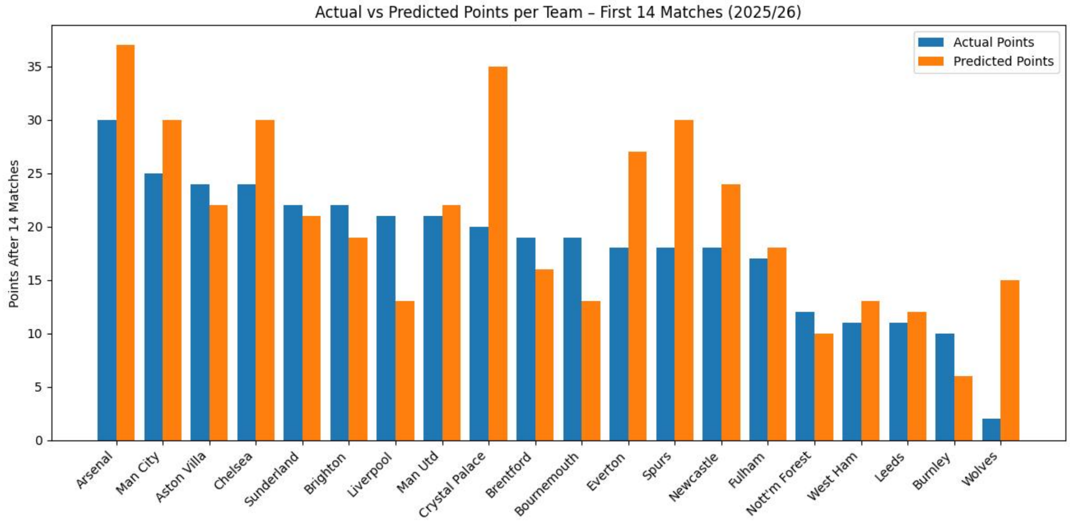

Figure 4: Actual versus predicted points for the top 10 teams after the first 14 matchweeks of the 2025/26 season. The comparison confirms that the pattern observed in 2024/25 persists early in the following season: predicted win totals are inflated and draws are under-represented.

Comment. Consistent with earlier results, this figure shows that the model's early-season predictions overestimate points and underestimate draws, reinforcing the need for richer features to improve draw calibration.

B Code Repository

GitHub Repository: `https://github.com/yaz212id/Project_Premier_League_Prediction`

The repository follows a clear structure:

- **src/**: contains modules for data loading, feature engineering, model definitions and evaluation.
- **notebooks/**: Jupyter notebooks for exploratory analysis and reproducing results.
- **requirements.txt**: lists all Python dependencies with pinned versions.
- **results/**: stores generated tables and figures.

Instructions in the README explain how to install dependencies, download the raw data, run the training scripts and reproduce the figures and tables. A continuous integration workflow (e.g. using GitHub Actions) can be added to automatically test data loading and model training on new commits.

Helper Tools

ChatGPT: Used for language refinement and structural guidance only. ChatGPT was also used to assist with code writing and debugging. The analytical contributions, data processing and modelling decisions were made by the authors.