Premier League Match Predictions

Luke Venkataramanan

December 12th, 2025

1 Introduction

Soccer, or football as it will be referred to in this paper, is the world's largest sport. The English Premier League is the most popular sports league in the world, garnering billions of views every year. In recent years, the abundance of football data and advancements in machine learning have revolutionized the use of predictive models for football outcomes. In this paper, we seek to compare the performance of various machine learning models at predicting Premier League match outcomes.

Before selecting and training our models, it is important to consider the inherent difficulty of this prediction task. Due to the nature of the low-scoring sport, luck and fine margins significantly impact results causing match outcomes to be notoriously noisy. Additionally, data sources encapsulating important features may be difficult to access, either due to paywall, difficulty merging datasets or simply no data at all.

This study aims to improve upon prior approaches using classical ML models to predict football results. We believe a predictive system that can predict a winner before the game has started has the most practical value. Many other studies use in-game statistics (i.e. score at half-time, game-possession, etc.) to boost accuracy and inflate reported performance via information leakage. To avoid data leakage and enable a feasible sports-arbitrage system, we aim to develop a model that will make predictions based on pre-game data. The following sections of our paper outline the approaches we take and discuss our results.

2 Data Sources

We use a few data sources for this project. Firstly, we used Football-Data.co.uk (Football-Data.co.uk, 2025) as our primary data source for match statistics. Our independent variables of interest for both the home and away teams were goals scored, shots on target, fouls, win streak, and betting odds.

Furthermore, we also used data from TransferMarkt (Transfermarkt, 2025), a popular website among users that provides the estimated market valuations of teams and players. We hypothesized that a squad's total market value might indicate how strong a team was, so we scraped this data from TransferMarkt.

Finally, we used FootballCritic (FootballCritic, 2025) to scrape possession data. Generally speaking, teams with higher possession rates tend to exercise more control over games, so we included this as a feature.

3 Technologies Used

This project integrated several modern software development tools and libraries.

First, Jupyter Notebook was used for exploratory data analysis and model development, as it allowed us to visualize data, experiment with feature engineering, and document results interactively. These notebooks interacted with reusable modules implemented in the project's source code.

Second, Scikit-learn served as the primary machine learning library. It was selected due to its strong support for machine learning algorithms, ease of integration with Python (our chosen language), and alignment with the techniques taught in the accompanying coursework.

Third, Flask was used to build a lightweight web application that exposes the trained model through a REST API. This inference server receives requests, constructs the appropriate input features, runs the model, and returns predictions to the user over the network.

Finally, Docker was used to containerize the application along with all its dependencies. Containerization ensures consistent execution across different environments and simplifies deployment. The final image was published to Docker Hub for easy distribution.

4 Methodology

4.1 Current State of the Art

Recent work frames football prediction as a three-class classification task with outcomes {home win, draw, away win}. A common finding is that ensemble methods tend to outperform individual models. Heijboer (Heijboer, 2022) analyzes 6,237 Premier League matches (2008–2016) using Random Forests, Gradient Boosting, SVMs, and Logistic Regression under class-imbalance controls. Using strictly pre-match features and chronological splits, the study reports a best accuracy of about 53.7%, slightly exceeding the bookmakers' implied-favorite baseline. Heijboer also shows that differential (home–away) features improve signal, while draws remain the hardest class, with a maximum F_1 of roughly 0.37.

Morgan et al. (Morgan, Grant, & Kim, 2024) provide a broader survey of soft-computing and machine-learning approaches. Three conclusions informed our methodology: (i) feature quality—particularly form, head-to-head history, and betting odds—matters more than model complexity; (ii) heterogeneous ensembles offer small but reliable performance gains; and (iii) chronological splits are critical for avoiding temporal leakage, especially for the imbalanced draw class. These findings shaped our feature engineering choices, use of ensemble methods, and evaluation design.

4.2 Data Pre-Processing

As previously mentioned, Football-Data.co.uk gave us key match statistics — results, goals scored, goals conceded, etc. — but we were also using data from TransferMarkt and FootballCritic. Because we were using features from a variety of datasets, we had to figure out how to merge the data.

Our program loads the Football-Data.co.uk data from each season into a large Pandas DataFrame. We then had a script scrape possession data from FootballCritic, store it in a CSV file, and later

merge the data into our original DataFrame based on date, home team, and away team. Because the websites would sometimes have inconsistent team names (i.e. TransferMarkt uses "Manchester United" while Football-Data.co.uk uses "Man United"), we needed to explicitly account for this. In order to merge squad valuation data from TransferMarkt, we took a similar approach, merging the data based on year, home team, and away team.

The data we used was very structured and consistent, so we did not have to use imputation to handle null data or any explicit type conversions. There also weren't any duplicate rows muddying our data. However, for newly promoted teams that lacked historical data, we initialized their features to reasonable baselines rather than zero or NaN. Rolling form statistics were set to previous-season league averages, and Elo was set to the seasonal baseline; these values were then updated as matches progressed. Although this slightly reduced raw accuracy, it improved the model's generalizability to unseen teams.

4.3 Feature Engineering

A team's recent performance, or "form," is a well-established predictor of match outcomes. To quantify form, we introduced a hyperparameter `N_MATCHES` and computed rolling statistics (wins, goals scored, goals conceded, fouls, etc.) over each team's previous `N` games.

To strengthen the feature matrix, we incorporated Elo ratings, which encode historical strength, opponent quality, and recency. Our Elo implementation includes a home-field advantage term and a seasonal regression toward the mean to reflect roster and manager changes and prevent long-term drift.

Bookmaker odds were also added as pre-match features. Because these odds are publicly available prior to kickoff and reflect market expectations, we used the average odds across all bookmakers as a stable signal.

After assembling and cleaning our feature set, we applied Principal Component Analysis (PCA) to evaluate redundancy and reduce dimensionality while preserving most of the dataset's variance.

4.4 Temporal Split

Because our model is meant to capture a team's recent form over the last `N_MATCHES`, we need to split our data into training and testing while retaining chronological information. For a project like this, it makes sense to utilize a temporal training split [2]; we use the first 70% of the dataset for training and the remaining 30% of the data for validation.

Using a random split (e.g. *sklearn.model_selection.train_test_split()*) causes data leakage, since random sampling mixes future matches into training. This is not appropriate for our situation, and it undermines model validity.

4.5 Models

We trained and compared a diverse set of models: Logistic Regression, Random Forest, XGBoost, Support Vector Machine (SVM), Multilayer Perceptron Feedforward Neural Network (MLPFFNN), Naive Bayes, and a Voting Ensemble.

Each model family captures different structures in the data and represents a distinct bias–variance trade-off. Logistic Regression models roughly linear relationships, whereas tree-based methods

like Random Forest and XGBoost capture non-linear interactions and threshold effects. SVMs provide flexible decision boundaries in high-dimensional spaces, while MLPs learn more complex nonlinear functions. Naive Bayes serves as a fast probabilistic baseline.

Comparing these models—and ultimately combining the strongest through a soft-voting ensemble—allowed us to evaluate which decision boundaries generalized best, identify where individual models struggled (particularly with draws), and assess whether ensembling improved robustness across seasons and feature regimes.

4.6 Evaluation

After training our models, we used accuracy as our metric of interest. The imbalance in our dataset is moderate, since home wins are a lot more likely than draws or away wins, but predicting a draw incorrectly is not inherently more costly than predicting a win correctly. As such, we use accuracy as our metric of interest instead of precision, recall, or F1 score.

Furthermore, we simply wanted to know which model was most likely to predict the correct results — exactly what accuracy measures.

5 Results

5.1 Principal Component Analysis

We applied Principal Component Analysis (PCA) to identify the main sources of variance in our feature set and to evaluate which engineered features were most informative. High-impact components were dominated by possession percentage, sportsbook odds, and Elo differentials, together explaining over 90% of the total variance. Form-related statistics (e.g., goals conceded, win streak) contributed moderate variance, while head-to-head metrics and squad values had relatively low impact.

Using `sklearn.decomposition.PCA`, we computed and visualized the principal components, with the top fifteen components ranked by importance in Figure 1.

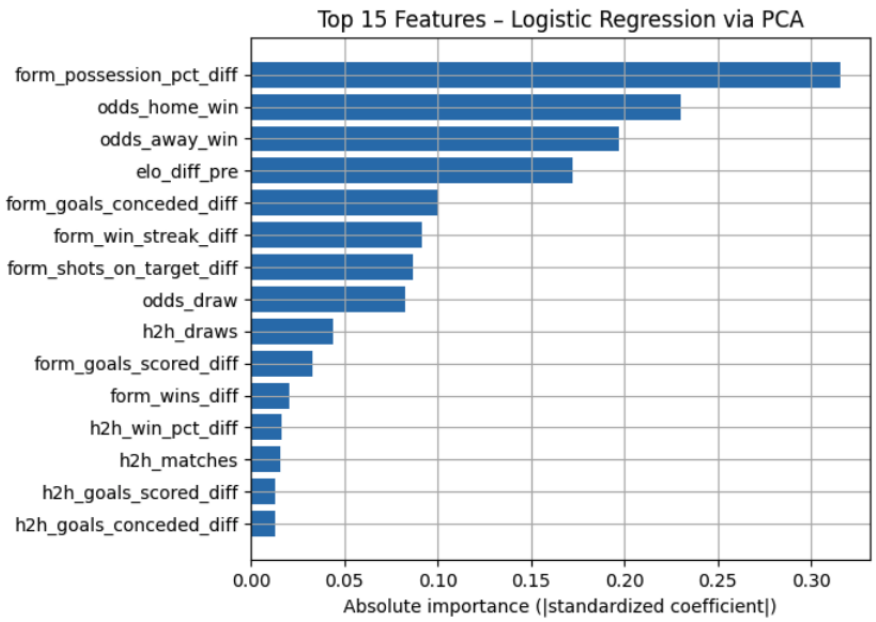


Figure 1: Top 15 Features - Logistic Regression via PCA

We observe that our Receiver Operating Characteristic (ROC) curve bows upward and toward the top left, indicating that the model achieves a high true positive rate while keeping the false positive rate relatively low. Our Area Under the Curve value of 0.729 demonstrates meaningful, though imperfect, predictive power, which is expected for a sport like football.

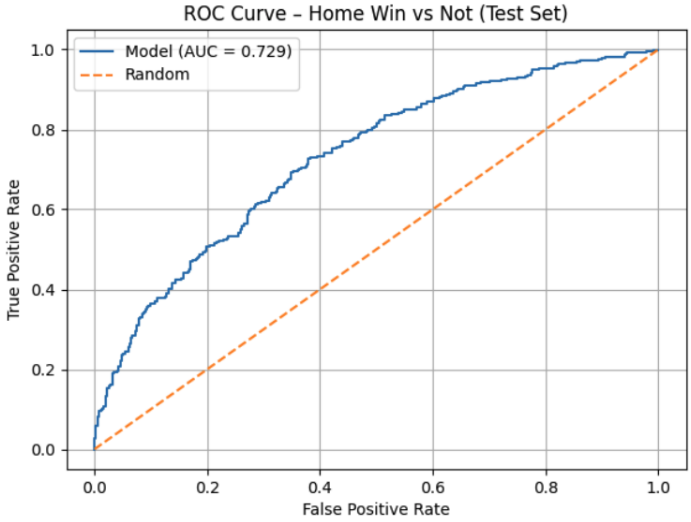


Figure 2: ROC Curve - Home Win vs. Not (Test Set)

PCA also highlighted the advantage of using differential (home–away) features: consolidating home and away statistics allowed the leading components to capture more variance, reducing the effective dimensionality of the feature space. Overall, the analysis clarified which features carried the strongest predictive signal and where simplification was possible.

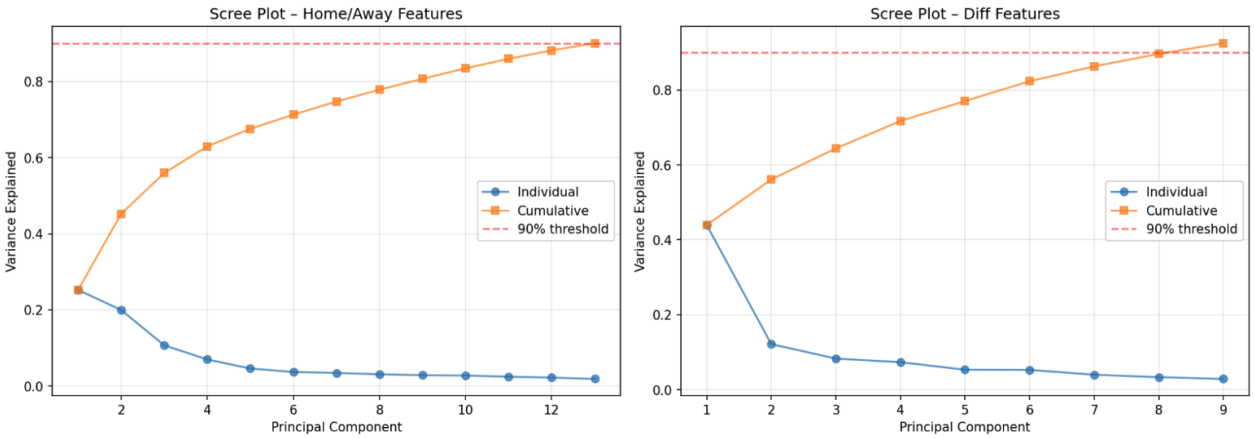


Figure 3: Scree Plots

5.2 Model Performance

Turning to predictive performance, accuracies for all selected models are summarized in Table 1. The overall range of accuracies across the plots below spans roughly 0.15, and consistent performance gaps between models emerged regardless of the chosen value of `N_MATCHES`. The Multilayer Perceptron Feedforward Neural Network (MLPFNN), Naive Bayes, and SVM models consistently performed the worst. XGBoost performed competitively, but Random Forest, Logistic Regression, and the Voting Ensemble clearly outperformed the remaining models, with accuracies approaching 60%.

Model	**Peak Training Accuracy**	**Peak Testing Accuracy**
Logistic Regression	0.58681	0.59071
XGBoost	0.57225	0.57909
Random Forest	0.57072	0.58624
Support Vector Machine	0.52587	0.54424
MLP Feedforward NN	0.50939	0.52011
Naive Bayes	0.47145	0.47364
Voting Ensemble	0.58605	**0.59249**
Heijboer SOTA RF	—	0.537

Table 1: Peak Training and Testing Accuracies Across All Models.

Now that we have discussed model performance, we will explore how tuning hyperparameters impacted model performance. The plots below visualize the testing accuracy of each model versus the value of `N_MATCHES`, for both the models before tuning and the models after tuning.

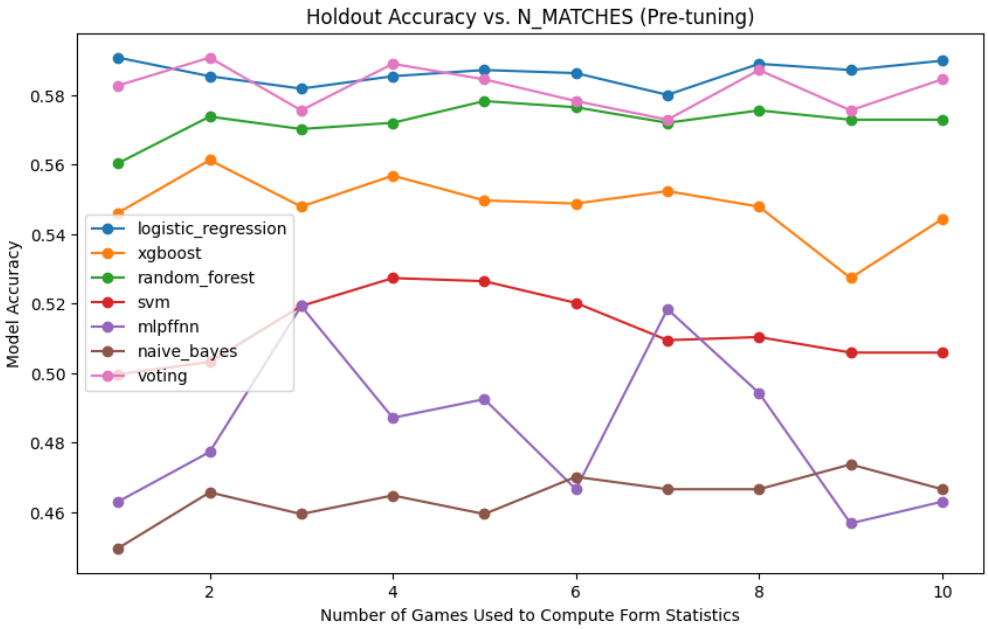


Figure 4: Accuracy vs. `N_MATCHES` (pre-tuning)

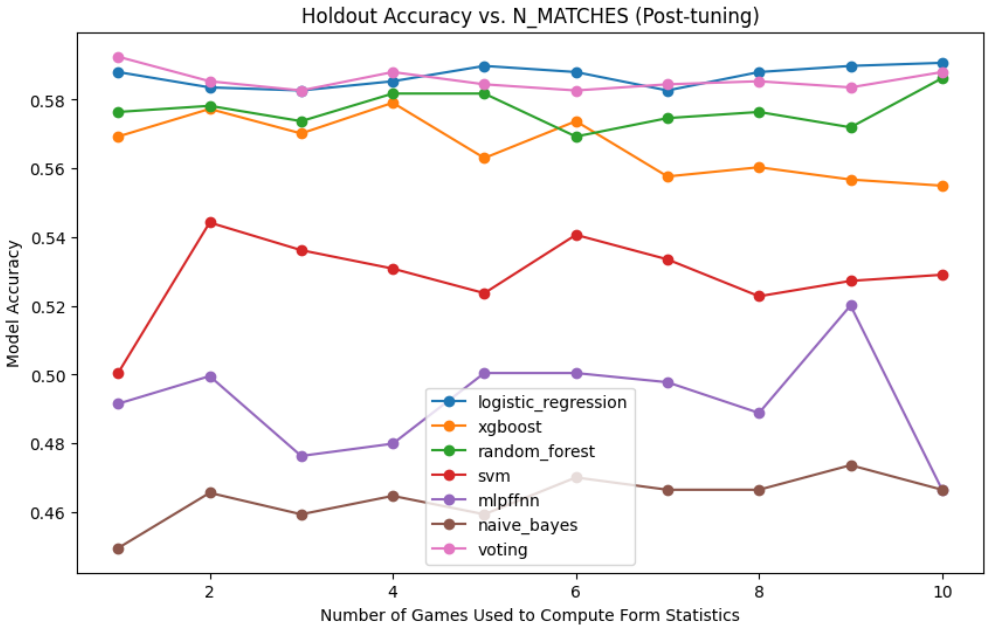
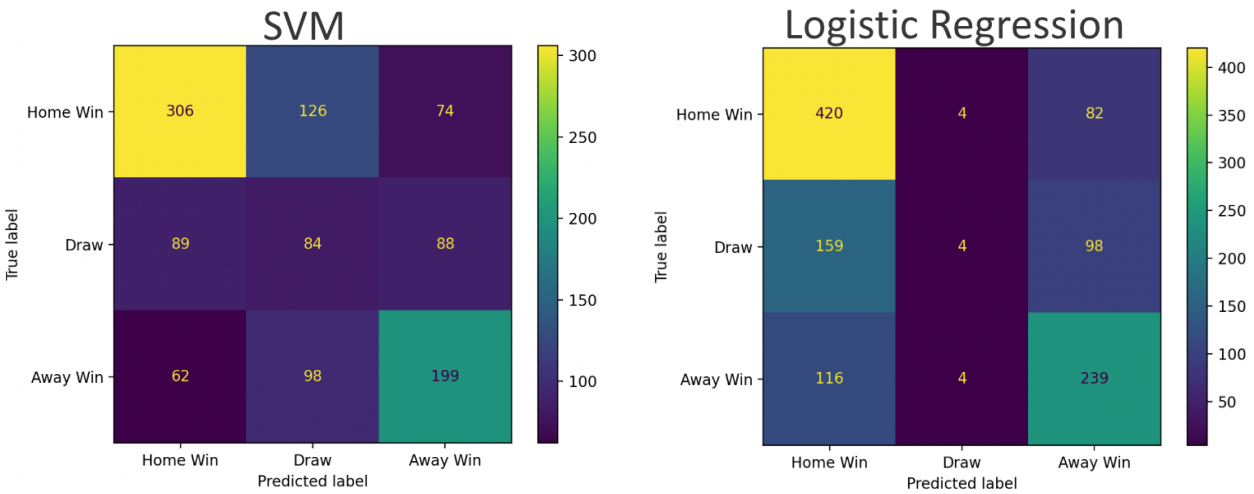


Figure 5: Accuracy vs. `N_MATCHES` (post-tuning)

As shown in Figures 4 and 5, hyperparameter tuning substantially improved model performance. XGBoost rose to match the accuracy of Random Forest and Logistic Regression, while tuning the voting weights elevated the ensemble to the top-performing method, achieving an accuracy of 0.59249. Relative to naive guessing (approximately 33% across win/loss/draw) and to the state-of-the-art benchmark from Heijboer—whose best Random Forest model reached an accuracy of

0.537—our tuned models demonstrate a notable improvement over existing publicized approaches for predicting soccer match outcomes.



(a) Confusion Matrix for Support Vector Machine

(b) Confusion Matrix for Logistic Regression

Figure 6: Confusion Matrices

The results also highlight why draws were far harder to predict than wins or losses. As shown in Figure 6, Logistic Regression almost never predicts a draw, while nonlinear models such as SVM, XGBoost, and Random Forest assign noticeably more probability mass to the draw class. This difference arises from both the severe class imbalance and the overlap between draw features and those of wins and losses.

Because draws are rare and poorly separated in feature space, linear models with a single global decision boundary tend to default to predicting win or loss. Nonlinear models, however, can form more complex and localized decision regions, allowing them to capture subtle patterns indicative of draw outcomes and achieve higher draw recall.

5.3 Conclusion & Outlook

Overall, our model improved on Heijboer's state of the art by over 5%, accomplishing the goals of the experiment: not only beating guessing odds, but creating a margin of accuracy gain sufficient to enable a second phase stemming from this research to use the models for sports betting. Crucially, our models only utilized past or forecasted data, no in-game statistics, making these predictions generalizable beyond on our dataset (most notably, for new games). With our significant gains in accuracy, the top models in this study will yield profitable results in time after many bets are placed. Looking forward, model accuracy could perhaps still be improved by integrating more complex or expensive data sources such as weather or injury data, but as-is, in its current state, this study could be leveraged for sportsbook arbitrage.

Overall, our model improves on Heijboer's state of the art by more than 5%, achieving the primary goals of the experiment: demonstrating a meaningful gain in predictive accuracy relative to established baselines. Importantly, all models in this study rely exclusively on pre–match information—past performance, market valuations, odds, and engineered form statistics—rather

than in-game events. This ensures that the approach generalizes beyond the matches in our dataset and can, in principle, be used to generate predictions for future fixtures.

References

Football-Data.co.uk. (2025). *Historical football results and betting odds data archive.* Retrieved from https://football-data.co.uk/englandm.php (Accessed: December 1, 2025)

FootballCritic. (2025). *Premier league statistics and match data.* Retrieved from https://www.footballcritic.com/premier-league/season-2025-2026/2/76035 (Accessed: December 1, 2025)

Heijboer, M. (2022). *Predicting football match outcomes with classical machine learning methods: A comparative study across european leagues* (Master's thesis, Tilburg University). Retrieved from https://arno.uvt.nl/show.cgi?fid=160932

Morgan, K. A., Grant, E. S., & Kim, E. (2024). Survey of soft computing methods to predict soccer win/loss probability. In *Proceedings of the 2024 international conference on information system and data mining (icisdm '24)* (pp. 1–7). ACM.

Transfermarkt. (2025). *Premier league data administration.* Retrieved from https://www.transfermarkt.com/intern/faq (Accessed: December 1, 2025)

Appendix

GitHub Repository

Presentation Slides

Video Demonstration