PremierInsight: Simulating Premier League Standings with ML

Noel Abraham Tiju
2022338
CSE, IIITD
noel22338@iiitd.ac.in

Rahul Ramesh Omalur
2022392
CSAI, IIITD
rahul22392@iiitd.ac.in

Dhruv Sharma
2022170
CSAM, IIITD
dhruv22170@iiitd.ac.in

Vimal Jayant Subburaj
2022571
CSAI, IIITD
vimal22571@iiitd.ac.in

Pratham Sibal
2022374
CSAM, IIITD
pratham22374@iiitd.ac.in

Abstract

Computer-aided sports predictions have become a cornerstone of modern sports analytics, yet accurately predicting match outcomes and standings remains a challenge. This study explores the application of various machine learning models, including logistic regression, support vector machines (SVM), decision trees, and ensemble methods such as random forests, xGBoost and Gradient Boosting alongside the ELO rating system to enhance predictions for English Premier League (EPL) matches. By evaluating these models' performance, we aim to identify the most effective approach for improving accuracy in predicting match results and league standings. The findings promise to benefit fans, analysts, and content creators by providing deeper insights and more reliable predictions. Github Link

1. Introduction

Predicting football match outcomes is a complex task due to the dynamic nature of the game, influenced by factors like player form, team strategies, injuries, and random events. Despite the abundance of data on players, teams, and past performances, accurately forecasting match results and league standings remains challenging. The English Premier League (EPL), known for its competitiveness, serves as an ideal platform to develop and test prediction models.

This project, PremierInsight, aims to enhance match outcome predictions by combining modern machine learning techniques with established metrics such as Elo ratings. By integrating advanced data-driven approaches, the model seeks to predict EPL match results and simulate final league standings, offering valuable insights for fans, analysts, and clubs. The model's performance will be rigorously evaluated against historical data to gauge its accuracy and practical utility.

2. Literature Review

Predicting football match outcomes, especially in the Premier League, remains challenging due to the sport's complex dynamics, which include quantitative and qualitative factors. Machine learning techniques increasingly analyze match statistics, player performance, and other variables for improved forecasts.

Hucaljuk and Rakipović (2011) [1] and Choi et al. (2023) [2] emphasize the complexity of football prediction due to diverse influences such as team form, historical encounters, individual performances, and external factors like injuries and fan presence. The low average number of goals per match (2–3) amplifies unpredictability, making small events decisive. Predicting draws, less frequent than wins or losses, poses further challenges due to class imbalance, which skews machine learning model performance.

Feature selection significantly impacts prediction accuracy. Both studies discuss leveraging domain knowledge and experimental testing to refine features. One began with 30 features, reducing to 20 after trials, focusing on team form, past results, goals, and injuries. Another used feature engineering, employing correlation matrices and boxplots to analyze match statistics like shots, corners, and goals across five previous matches. These refined feature sets consistently enhanced model accuracy.

Classifier selection included evaluating Naive Bayes, k-NN, Random Forest, and Artificial Neural Networks (ANNs) using accuracy and F1 score. Hucaljuk and Rakipović (2011) [1] reported ANNs and LogitBoost achieving up to 68% accuracy, while Naive Bayes underperformed. Choi et al. (2023) [2] highlighted Random Forest's superiority in binary classification and logistic regression's effectiveness for multiclass tasks. Balanced sampling strategies, addressing class imbalance, significantly improved prediction of rare outcomes like draws.

Both studies faced limitations, including small datasets and difficulties in accessing historical player data, which reduced model generalizability. Future research could improve by using larger datasets, advanced feature selection methods, and deep learning models. Incorporating detailed player statistics and enhancing models to evaluate individual performances are also recommended directions.

Limitations include small datasets and difficulties in acquiring detailed historical data, reducing generalizability. Future work should explore larger datasets, advanced feature engineering, and deep learning models, focusing on incorporating granular player performance data for enhanced predictions.

3. Dataset

The dataset utilized in this project integrates features from various sources, including historical match data, Elo ratings, xG statistics, and player performance metrics. Our comprehensive dataset facilitates an in-depth analysis of team performance and match outcomes across multiple English Premier League (EPL) seasons. The final dataset includes the following features: **Home Team, Away Team, Home Team ELO, Away Team ELO, Home xG, Away xG, Home xGA, Away xGA, Home Win Percentage, Home Draw Percentage, Away Win Percentage, Away Draw Percentage, Home Team Form, Away Team Form, Home Team Cumulative Points, Away Team Cumulative Points, Home Team Form Statistics, Away Team Form Statistics, Winner.**

3.1. Feature Sources

3.1.1 Kaggle Datasets

Key match statistics, such as goals scored, teams involved, and match outcomes, were sourced from the following Kaggle datasets:

- Kaggle Dataset 1: EPL match data from 1992 to 2022.
- Kaggle Dataset 2: Match data for the 2023-2024 EPL season.

These datasets provided foundational statistics, such as **Season, Home Team, Away Team, Home Goals, Away Goals, and Winner (FTR)**.

3.1.2 Elo and xG Ratings

Elo ratings and Expected Goals (xG) data were combined to dynamically assess team performance. Below, we outline the steps and formulas used.

Initialization: All Premier League teams start with an Elo rating of 1500, with adjustments for newly promoted teams in later seasons.

Expected Outcome Calculation: The expected probabilities for the home and away teams are:

$$E_A = \frac{1}{1 + 10^{(R_B - R_A)/400}}, \quad E_B = \frac{1}{1 + 10^{(R_A - R_B)/400}}$$

where R_A and R_B are the pre-match Elo ratings.

Outcome and Rating Update: After the match, ratings are updated as:

$$R'_A = R_A + K \times (S_A - E_A), \quad R'_B = R_B + K \times (S_B - E_B)$$

where S_A and S_B are the actual results (1 for a win, 0.5 for a draw, 0 for a loss), and $K = 32$.

Expected Goals (xG) data was obtained from Understat for matches from the 2014-15 season onward. Adjustments to xG and xGA were made to account for goal difference and home advantage using the following formulas:

$$\text{xG}_{\text{home}} = \text{round}\left(\text{xG}_{\text{home}} \times (1 + 0.05 \times \text{factor} \times \text{GD})\right)$$

$$\text{xGA}_{\text{home}} = \text{round}\left(\text{xGA}_{\text{home}} \times (1 - 0.05 \times \text{factor} \times \text{GD})\right)$$

where GD is the goal difference, and the factor used is 32.

Metrics:

- **Home xG, Away xG**: Expected goals for home and away teams.
- **Home xGA, Away xGA**: Expected goals against for home and away teams.

3.1.3 Player Statistics

Detailed player performance metrics were scraped using BeautifulSoup from FBREF, covering attributes such as goals, assists, yellow cards, and tackles from 1992 to 2024. Goalkeeper-specific metrics, such as saves and clean sheets, were also included. A custom formula for integrating these metrics into team-level statistics is under development:

Player Statistics Formulas

Team Performance Formula:

$$\text{season}(n) = 0.6 \cdot \text{season}(n-1) + 0.3 \cdot \text{season}(n-2) + 0.1 \cdot \text{season}(n-3)$$

Penalty Adjustments:

$$\text{Penalty} = \begin{cases} 5 & \text{if season}(n-1) \notin \text{Premier League}, \\ 3 & \text{if season}(n-2) \notin \text{Premier League}, \\ 1 & \text{if season}(n-3) \notin \text{Premier League}. \end{cases}$$

Feature Weights for Team Aggregation:

$$\text{Score} = 20 \cdot \text{GSCR} + 15 \cdot \text{AvgPens} + 10 \cdot \text{CleanSheets} - 0.5 \cdot \text{Saves} - 1 \cdot \text{SOT} + 10 \cdot \text{Save\%}$$

Formula Descriptions:

- **Team Performance Formula:** Weights recent seasons more heavily (60%, 30%, 10%) to reflect current form.
- **Penalty Adjustments:** Applies penalties for missing Premier League seasons, with higher penalties for more recent absences.
- **Feature Weights for Team Aggregation:** Combines player metrics into a team score, prioritizing offense (GSCR, penalties) and defense (clean sheets, save percentage), while penalizing weaknesses (e.g., low saves, high shots faced).

3.2. Final Dataset

The final dataset combines the above features, organized seasonally and exported in CSV format, enabling machine learning-based predictive analysis. It is publicly accessible at Google Drive.

4. Methodology and Model Details

To evaluate the performance of different models in predicting football match outcomes (home wins, away wins, and draws), we compared the accuracy, F1 score, and the percentage of correct predictions for each class (home wins, away wins, and draws). Among the various models tested, our final choices were the XGBoost Classifier and a custom ensemble model combining Random Forest, Gradient Boosting, and XGBoost classifiers for enhanced predictive performance.

4.1. Final Model: XGBoost Classifier

Extreme Gradient Boosting (XGBoost) is a powerful ensemble method known for its scalability and regularization techniques, which help prevent overfitting. For our implementation, we used the `XGBClassifier` with $n\_estimators = 2000$ and $learning\_rate = 0.01$ to ensure gradual, stable learning over 2,000 boosting rounds. We set $max\_depth = 7$ to capture complex patterns while minimizing overfitting and applied $colsample\_bytree = 0.8$ and $subsample = 0.8$ for feature and instance sampling to enhance generalization. Additionally, $gamma = 1$ was used to regularize tree splitting, and $eval\_metric =$ 'mlogloss' evaluated multi-class log loss during training. A $random\_state = 16$ ensured reproducibility.

4.2. Final Model: Custom Ensemble Model

The custom ensemble model combines the strengths of Random Forest, Gradient Boosting, and XGBoost classifiers to achieve robust predictive performance. Each model was fine-tuned for optimal accuracy and complementary behavior in the ensemble.

Random Forest Classifier: Configured with $n\_estimators = 1000$ for a large ensemble of trees, $max\_depth = 10$ to limit tree growth and prevent overfitting, and $bootstrap = False$ for sampling without replacement. It uses $min\_samples\_leaf = 1$ and $min\_samples\_split = 2$ to enable fine-grained splits.

Gradient Boosting Classifier: Configured with $n\_estimators = 1000$ for extensive boosting rounds, $learning\_rate = 0.5$ for controlled updates, and $max\_depth = 4$ to capture significant patterns while avoiding overfitting. A $random\_state = 16$ ensures consistent results.

XGBoost Classifier: Employs $n\_estimators = 2000$ and $learning\_rate = 0.01$ for gradual, stable learning over many boosting rounds. It utilizes $max\_depth = 7$ for balance between complexity and overfitting, along with $colsample\_bytree = 0.8$ and $subsample = 0.8$ for improved generalization. The $gamma = 1$ parameter adds regularization for tree splits, and $eval\_metric =$ 'mlogloss' evaluates multi-class log loss during training. $random\_state = 16$ ensures reproducibility.

4.3. Models Tested and Rejected

We also explored several other models during the evaluation phase, including Logistic Regression, Support Vector Machines (SVMs), Decision Trees, Random Forest, and Gradient Boosting as standalone classifiers. While these models provided valuable insights and baseline comparisons, their performance did not surpass that of the XGBoost Classifier or the custom ensemble model. Below is a brief overview of the models tested:

- **Logistic Regression:** Extended for multi-class classification using a 'one vs rest' approach, employing the 'saga' solver and L2 regularization for convergence and overfitting prevention.
- **Support Vector Machines (SVMs):** Utilized the squared hinge loss function with L2 regularization, focusing on maximizing the margin between class boundaries.
- **Decision Trees:** Leveraged the Gini criterion for splitting but lacked robustness compared to ensemble methods.
- **Artificial Neural Network:** It has a higher accuracy than simple models but does not out perform our final models.
- **Multi-Layer Perceptrons:** Eventhough is has a higher accuracy than the ANN, it does not perform well enough.

5. Results

5.1. Tested Models and their Accuracies

Model	Accuracy
Logistic Regression	54.34%
Support Vector Machine (SVM)	56.97%
Decision Trees	61.57%
Artifical Neural Network	66.84%
Multi-Layer Perceptron	69.73%
Random Forest Classifier	73.55%
xGBoost	74.07%
Custom Ensemble	75.52%
Gradient Boosting Classifier	76.84%

Table 1. Model Accuracies for Predicting Football Match Outcomes

5.2. Classwise accuracy of the final models

Model	Away (%)	Draw (%)	Home (%)
Random Forest	62.14%	72.67%	78.93%
XGBoost	63.37%	75.78%	82.58%
Gradient Boosting	67.49%	78.88%	82.58%
ELO Insight	65.43%	75.78%	82.30%

Table 2. Classwise Accuracy of Different Models (in Percentages)

5.3. Predicted vs Actual Standings

The models (xGBoost, Custom Ensemble, and Gradient Boosting) showed varying accuracy in predicting the 2023-2024 Premier League standings as in Figure 1. All models correctly identified Manchester City as champions and Liverpool as a top contender, but Arsenal was underestimated, particularly by xGBoost and Gradient Boosting. Aston Villa and Newcastle were generally well-predicted, though their exact positions varied. For mid-table teams like Tottenham, Chelsea, and Brighton, xGBoost performed best, while Custom Ensemble and Gradient Boosting struggled. Relegation zone predictions were less accurate, with Burnley and Sheffield United overestimated by Custom Ensemble and Gradient Boosting. Overall, xGBoost was most consistent for top and mid-table teams, while improvements in feature selection and model tuning are needed for better accuracy, particularly for relegation candidates.

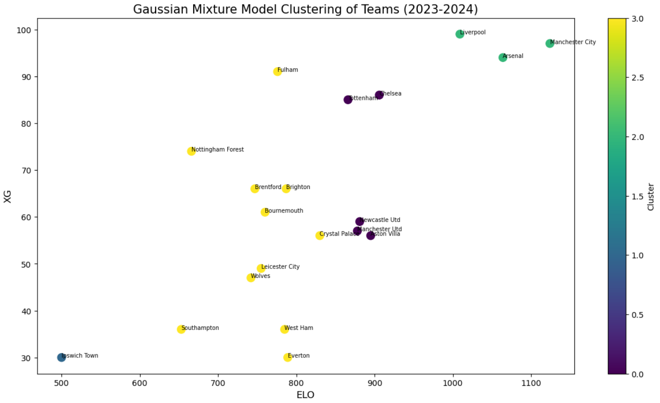
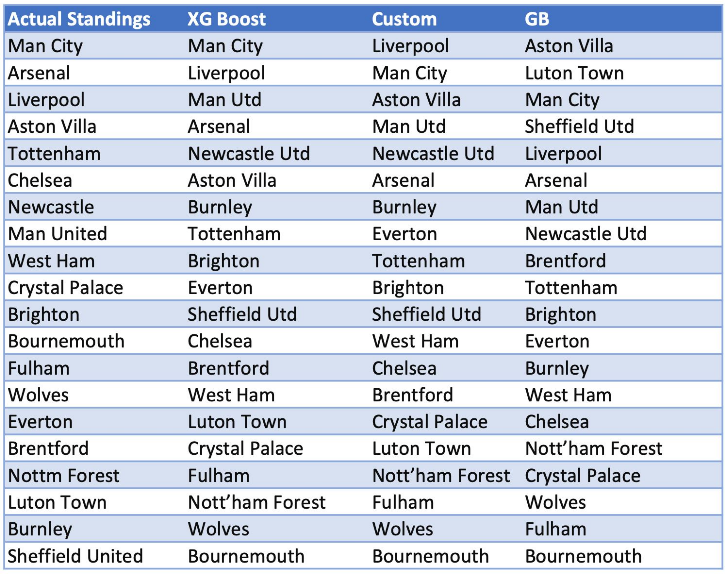


Figure 1. GMM Clusters for segmenting teams into Top-Table,Mid-Table and Relegation teams



Actual Standings	XG Boost	Custom	GB
Man City	Man City	Liverpool	Aston Villa
Arsenal	Liverpool	Man City	Luton Town
Liverpool	Man Utd	Aston Villa	Man City
Aston Villa	Arsenal	Man Utd	Sheffield Utd
Tottenham	Newcastle Utd	Newcastle Utd	Liverpool
Chelsea	Aston Villa	Arsenal	Arsenal
Newcastle	Burnley	Burnley	Man Utd
Man United	Tottenham	Everton	Newcastle Utd
West Ham	Brighton	Tottenham	Brentford
Crystal Palace	Everton	Brighton	Tottenham
Brighton	Sheffield Utd	Sheffield Utd	Brighton
Bournemouth	Chelsea	West Ham	Everton
Fulham	Brentford	Chelsea	Burnley
Wolves	West Ham	Brentford	West Ham
Everton	Luton Town	Crystal Palace	Chelsea
Brentford	Crystal Palace	Luton Town	Nott'ham Forest
Nottm Forest	Fulham	Nott'ham Forest	Crystal Palace
Luton Town	Nott'ham Forest	Fulham	Wolves
Burnley	Wolves	Wolves	Fulham
Sheffield United	Bournemouth	Bournemouth	Bournemouth

Figure 2. Comparison between final models prediction and the actual standings 2023-2024

6. Conclusion

In conclusion, the evaluation of machine learning models for predicting football match outcomes highlighted the XGBoost Classifier and a custom ensemble model as top performers. XGBoost achieved an accuracy of 74.07%, while the ensemble model showed strengths in specific scenarios. Gradient Boosting and Random Forest also performed competitively, with accuracies of 76.84% and 73.55%, respectively (Table 1). Gradient Boosting excelled in predicting away wins (67.49%) and draws (78.88%), while XGBoost and Gradient Boosting were equally effective for home wins (82.58%) (Table 2). Top-performing teams, such as Manchester City and Liverpool, were consistently predicted well, showcasing the models' strength for dominant teams. However, challenges persisted with relegation candidates, particularly Burnley and Sheffield United. Adding features like betting odds and player data could improve predictions for lower-performing teams, laying a solid foundation for further advancements in football match prediction.

Contributions: Noel, Rahul, Vimal worked on models and features, Dhruv and Pratham worked on web scraping, getting player statistics and certain features.

References

[1] Bing Choi, Lee-Kien Foo, and Sook-Ling Chua. Predicting football match outcomes with machine learning approaches. *MENDEL*, 29:229–236, 12 2023.

[2] Josip Hucaljuk and Alen Rakipović. Predicting football scores using machine learning techniques. In *2011 Proceedings of the 34th International Convention MIPRO*, pages 1623–1627, 2011.