BEAT THE BOOKIES

A PREPRINT

Group Name: `Group K`
Department of Computer Science
University College London
London, WC1E 6BT

18 December 2023

1 Introduction

The problem we were tasked with was to train a machine learning model to predict the full-time results (FTRs) of given English Premier League (EPL) football matches. Each prediction needed to be in the form of "H", "A", or "D", indicating a home win, an away win, or a draw respectively. We were given access to an initial dataset of EPL post-match data from August 2000 to October 2023, including for each match the home and away teams playing, the goals scored by each side at half time and full time, shots on target and more.

Our approach began with a review of relevant literature, including papers and competitions covering similar predictive tasks for football, as well as for other sports and more general domains. With the help of selected resources, we compiled information on how similar problems have previously been tackled and formulated our own approach by finding combinations of complementary techniques that have not been explored in the literature. Seeing from our research that more extensive match and player data leads to significantly better results [8], we gathered injury reports, player statistics and weather data at the time of matches. The more complete dataset allowed us to enrich the set of features we would produce for the model but also presented challenges with respect to data pre-processing and conversion to binary features.

Following this, the dataset was reformatted to show each feature broken down by team as opposed to match by match. Using the reformatted table, we explored the data by computing the correlation of features with full-time results (FTR) and creating descriptive statistics such as win percentage or average goals per match for each team. Analysing these results led us to evaluate the (ir)relevancy of certain aspects of the data, such as the diminishing value of 5+ year-old match data or the importance of home advantage. Once we had a clear understanding of the data, we performed feature engineering. Most notably, we constructed the home and away Pi-Ratings of each team from their performance in the training data [2]. To take the Pi-Rating system forward we also created a set of features based on the pairwise rating of teams, where a team has individual and independent ratings against all other teams. Additionally, our approach explores the use of sentiment analysis, performed through a Large Language Model (LLM) accessing a social media API, with the objective of assigning sentiment scores to teams based on prevailing online posts and discussions before kick-off.

Finally, we trained and evaluated four machine learning classifiers on the full dataset as well as taking only the past one, two and five seasons as training data. To determine the accuracy and feature redundancy of each model we used a combination of K-fold cross-validation, precision, recall and F1 scores with the Shapely Additive exPlanations (SHAP) method. After hyperparameter tuning, the highest performing model was found to be the CatBoost Classifier, which achieved 51.06% accuracy on the test set, trained on the first 75% percent of the 2023/24 EPL matches up until 18th December 2023, while the test data consisted of the remaining 25% of matches in the current season.

2 Data Transformation and Exploration

Data transformation is an important aspect of machine learning, used for the purpose of organising and preparing raw data for use in machine learning algorithms. Data exploration, on the other hand, is essential for understanding the specific characteristics of data sets, helping facilitate the exploration of patterns and insights crucial for its analysis.

2.1 Pre-Processing

Throughout our process, several data transformations were implemented in order to prepare the data for modelling. These included normalization to ensure a consistent scale amongst numerical features, pre-processing, feature engineering and feature scaling.

A comprehensive set of pre-processing steps was implemented in our code to clean, transform and organise the raw data into a format suitable for analysis and model implementation. This began with the initialisation of a dictionary (team_scores) to accumulate scores for each team, which was updated based on the goals recorded in home and away matches. Additional dictionaries (team_wins, team_loss, team_draw) were also created to monitor wins, losses, and draws for each team. Additionally, various team-specific statistics were then computed, encompassing metrics including goals scored at half-time, shots on goal, shots on target, yellow and red cards, corners, and fouls. The inclusion of these statistics allowed for better organisation of the data.

Furthermore, a dictionary (team_seasons) was also added in order to track the active seasons for each team. To help with the analysis, two Pandas DataFrames (team_stats_df and team_stats_relative_df) were constructed, serving as an organised repository for the calculated statistics. Additionally, statistics including win percentage, draw percentage, loss percentage, shots on goal per match, etc were calculated. This normalisation allowed for a better understanding of individual teams' performance, accounting for variations in the total number of matches played.

Separately, the FTR feature, which was to be predicted, underwent a one-hot encoding transformation. This process involved systematically converting its original values of "H", "D", or "A" to class representations, specifically 0, 1, or 2, respectively. This application of label encoding provided a numerical representation of the previously categorical data, facilitating its integration into our machine learning models. More effective decision-making and in-depth analysis was enabled by calculating a diverse set of team-specific statistics.

As part of the pre-processing step, feature scaling was also involved. Here, that data was normalised to ensure a consistent scale amongst numerical features. This scaling process ensures that the influence of different features on the machine learning algorithms is proportionate, preventing biases.

2.2 Data Exploration

To enhance our understanding of the data and uncover relationships between specific features, correlation matrices were created. Through this, it was important to look for possible relationships and compare the impact of each individual feature with our target variable FTR.

High Correlations between features may indicate redundancy. Therefore understanding the relationships involved can guide the feature selection process and improve the model's interpretability and performance. The correlation values achieved vary from -1 to +1 with values >0 indicating a positive correlation and values <0 indicating one that is negative.

2.2.1 Correlations Discovered:

Positive Correlations:

+ FTHG (Full Time Home Goals) has a strong positive correlation with HTHG (Half Time Home Goals) and a moderate positive correlation with HST (Total Shots on Target by the Home Team).
+ FTAG (Full Time Away Goals) has a strong positive correlation with AST (Total Shots on Target by the Away Team) and a moderate positive correlation with HST.

Moderate Positive Correlations:

+ FTHG has a moderate positive correlation with HS (Total Shots on Goal by the Home Team) and HC (Total Corners by the Home Team).
+ FTAG has a moderate positive correlation with AS (Total Shots on Goal by the Away Team) and AC (Total Corners by the Away Team).

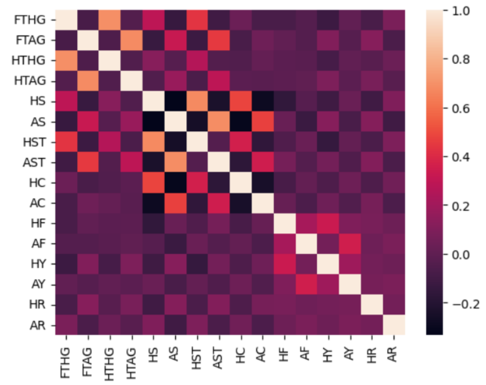

Figure 1: Correlation Matrix Amongst Features.

After producing a detailed correlation matrix for all features in our data set, we directed our focus more specifically to examine the relationships with the target variable, FTR. This aimed to uncover insights into which specific features contributed and influenced predicting the outcome's of future matches. Strong positive correlations between the FTHG and HTHG features with a home win were present, as intuitively expected, with similarly strong positive correlations between FTAG and HTAG with an away win. Other notable findings included that Full Time Home Goals had the strongest negative correlation out of all the features with a Full Time Result of a draw, the home and away teams' shots on target (HST and AST) had stronger correlations with home and away wins respectively than home and away shots on goal, home and away team corners had the weakest correlations with any full time result, and finally red and yellow cards received by the away team appeared to have a lower correlation with any full time result than those received by the home team, with yellow cards especially impacting the result either way for the away team than for the home team.

3 Methodology Overview

To make the most out of the data set we were provided, we went beyond calculating team statistics to engineer our own features. Most importantly, we implemented and extended the Pi-Rating scheme and experimented with external data sources and features such as sentiment analysis or win streak.

Prior to feature engineering and developing the models we conducted a brief literature review of relevant football and other sporting event prediction articles and resources. From our research, we concluded that for football, to generate features that accurately capture a team's current ability and chance against opponents, we should consider:

a The well-known phenomenon of home advantage [3][4][5]

b Emphasising the greater significance of recent results over less recent ones when estimating current ability [6]

c The fact that a win is more important for a team than increasing goal difference

d Developing either a rating scheme for teams or a formula for expected goals (XGs) provides the strongest features [2]

e Individual player statistics incorporated into the model often leads to significantly better accuracy [8]

In addition to feature engineering, we split the original data set into different training sets to also test and evaluate the models' predictive power with respect to the recency of match data. Namely, we segmented the data into four categories: All Seasons, Past Five Seasons, Past Two Seasons and Past Season. While it is well documented that more recent data results in higher accuracy, we wanted to test at what point older data becomes neutral or potentially harmful to the model.

The following subsections detail our approach to feature engineering, in particular how we extended the Pi-Ratings in [2] to the exponential time-decay adjusted pairwise and weighted pairwise Pi-rating scheme.

3.1 Pi-Ratings

The Pi-Rating system dynamically reflects both current team performance trends and the importance of match outcomes, ensuring a robust and responsive rating mechanism. This approach leads to the generation of comprehensive features, including the home team's home rating (`HT_HomeRating`), the home team's away rating (`HT_AwayRating`), the away team's home rating (`AT_HomeRating`), and the away team's away rating (`AT_AwayRating`), as well as pairwise (`PW` prefix) and weighted pairwise (`WPW` prefix) equivalents, offering a nuanced view of team strengths and weaknesses in both home and away contexts.

3.1.1 Simple Pi

We assign each team an initial rating of 0, which represents an average team relative to the residual teams [2].

One iteration (match) of calculating pi-ratings involves seven steps as follows:

1. Calculate Home and Away Team's Expected Goal Difference:
   Home Team α: $\hat{g}_{DH}$ is the goal difference of team α against the average opponent when playing at home. Here, $R_{\alpha H}$ represents the rating of Team α when playing at home.

$$\hat{g}_{DH} = \begin{cases} 10^{\left|\frac{R_{\alpha H}}{3}\right|} - 1, & \text{if } R_{\alpha H} \geq 0 \\ -\left(10^{\left|\frac{R_{\alpha H}}{3}\right|} - 1\right), & \text{otherwise} \end{cases}$$

- Away Team β: $\hat{g}_{DA}$ is the goal difference of team β against the average opponent when playing away. Here, $R_{\beta A}$ represents the rating of Team β when playing away.

$$\hat{g}_{DA} = \begin{cases} 10^{\left|\frac{R_{\beta A}}{3}\right|} - 1, & \text{if } R_{\beta A} \geq 0 \\ -\left(10^{\left|\frac{R_{\beta A}}{3}\right|} - 1\right), & \text{otherwise} \end{cases}$$

2. Predict the Match Goal Difference: Compute the predicted goal difference $\hat{g}_D$ by subtracting $\hat{g}_{DA}$ from $\hat{g}_{DH}$. A positive value indicates the home team is predicted to score more goals than the away team, while a negative value indicates the opposite.

$$\hat{g}_D = \hat{g}_{DH} - \hat{g}_{DA}$$

3. Record the Actual Goal Difference: Log the actual goal difference (g_D) from the match, which is the difference between the home team's goals (g_H) and the away team's goals (g_A).

$$g_D = g_H - g_A$$

4. Determine Prediction Error e: Calculate the error e as the absolute value of the difference between the predicted goal difference ($\hat{g}_D$) and the actual goal difference (g_D).

$$e = |\hat{g}_D - g_D|$$

5. Exponential Time Decay: Apply a decay factor $\alpha^{t_{\text{diff}}}$ to account for the time elapsed since the match. The time difference t_{diff} is calculated as the difference between the date of the latest match d_{latest} and the date of the current match d_{match}, both expressed in days.

$$t_{\text{diff}} = d_{\text{latest}} - d_{\text{match}}$$
$$\text{decay} = \alpha^{t_{\text{diff}}}$$

6. Weighted Error Calculation: First, calculate the weighted error $\psi(e)$ using a logarithmic function with $c = 3$ [2] as the constant and incorporate the decay. Then, adjust $\psi(e)$ based on prediction accuracy for home ($\psi_H(e)$) and away ($\psi_A(e)$) teams.

$$\psi(e) = \text{decay} \cdot c \cdot \log_{10}(1 + e)$$

$$\psi_H(e) = \begin{cases} \psi(e), & \text{if } \hat{g_D} < g_D \\ -\psi(e), & \text{otherwise} \end{cases}$$

$$\psi_A(e) = \begin{cases} \psi(e), & \text{if } \hat{g_D} > g_D \\ -\psi(e), & \text{otherwise} \end{cases}$$

7. Revise Pi-Ratings: Update team ratings using weighted errors. Hyper-parameters λ (for direct adjustments) and γ (for catch-up adjustments) are used.

   - **Team α (Home Rating):** Adjust with home error.

   $$\hat{R_{\alpha H}} = R_{\alpha H} + \psi_H(e) \cdot \lambda$$

   - **Team α (Away Rating):** Adjust based on home rating change.

   $$\hat{R_{\alpha A}} = R_{\alpha A} + (\hat{R_{\alpha H}} - R_{\alpha A}) \cdot \gamma$$

   - **Team β (Away Rating):** Adjust with away error.

   $$\hat{R_{\beta A}} = R_{\beta A} + \psi_A(e) \cdot \lambda$$

   - **Team β (Home Rating):** Adjust based on away rating change.

   $$\hat{R_{\beta H}} = R_{\beta H} + (\hat{R_{\beta A}} - R_{\beta A}) \cdot \gamma$$

In consideration of the principles set out in Section 3, the following refined our approach to enhance Pi-Ratings:

a Different home and away ratings are adjusted using a catch-up learning rate γ, balancing the influence of performances both at home and away.[2]

b A learning rate λ governs how much recent goal-based match results influence existing ratings, ensuring that current performances are more significantly represented.[2]

c An exponential time decay $(\alpha)^t$ is employed to moderate the impact of older matches, also prioritising recent performances for a more accurate reflection of current team form.

d We use a logarithmic function $\psi(e) = c \cdot \log_{10}(1 + e)$ to weight goal differences, recognising that in football, a win matters more than the margin of victory. This approach is one solution for diminishing the impact of high score differences.[2]

3.1.2 Pairwise Pi

Pairwise Pi (PW Pi) is an advanced approach that extends Simple Pi. It evaluates how teams fare against each other in both home and away contexts by calculating unique ratings for every possible pair of teams. This takes into account the specific dynamics of each head-to-head encounter.

While the calculations in PW Pi are similar to Simple Pi, thus maintaining similar time complexity, the space complexity is higher. This is due to the need to store unique ratings for every possible pairing among n teams, resulting in $n(n-1)$ unique ratings. This increase in data storage is a reflection of the model's detailed approach.

The PW Pi model offers a more nuanced and targeted understanding of team dynamics. It does this by considering the specific strengths and weaknesses in direct matchups, thereby capturing the unique dynamics and strategies of each encounter. The added storage requirement is a small compromise for the enriched insights.

3.1.3 Weighted Pairwise Pi

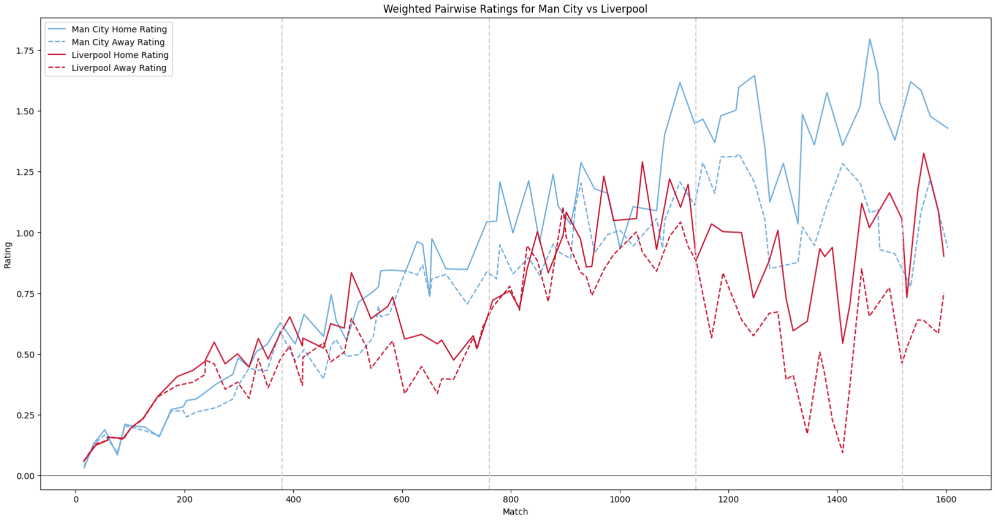

Figure 2: Time-Series Graph of WPW Pi Ratings of Man City and Liverpool

However, PW Pi risks overfitting by potentially interpreting noise or anomalies as significant trends. The model's complexity increases quadratically with the number of teams, leading to a vast number of unique ratings; this can be problematic if the available data for certain pairings is insufficient, as the model might overfit to limited or random fluctuations.

To mitigate this, Weighted Pairwise (WPW) Pi combines PW Pi's specific matchup ratings with the broader team ratings from Simple Pi. This blend reduces the risk of overfitting by balancing detailed head-to-head data with general team performance.

The model adjusts the weight of specific vs. general ratings to enhance reliability, ensuring a robust, well-rounded analysis of team strengths and weaknesses.

3.1.4 Pi Rating Validation & Hyperparameter Tuning

The aim is to optimise the hyperparameters λ and γ in the Pi-Ratings model. These parameters critically influence the model's ability to accurately predict goal differences in football matches.

We optimise by employing grid search, a systematic approach to test combinations of λ and γ values. This method involves iterating over a predefined list of potential values for each hyperparameter, assessing the effectiveness of each (λ, γ) pair using the mean squared error (MSE) between the predicted goal difference and the observed goal difference calculated by the model.

$$MSE = \frac{1}{n} \sum (\hat{g}_D - g_D)^2$$

The optimal hyperparameters are those that yield the lowest MSE, indicating the most accurate predictions by the model.

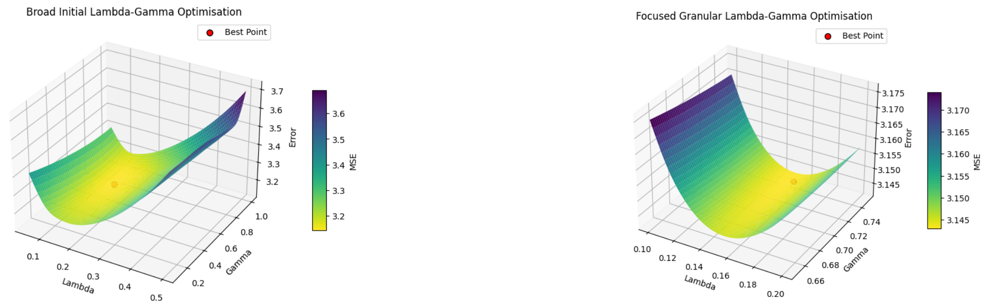

Figure 3: Broad Initial and Focused Granular λ-γ Optimisation

The grid search process is split into two parts, a broad initial optimisation followed by a focused granular optimisation.

Initial Broad Optimisation:

The process began with a broad search across a predefined range to find the optimal settings for λ (from 0.05 to 0.5) and γ (from 0.1 to 1.0), using a step size of 10 to explore the parameter space efficiently.

A 3D surface plot (Figure 3, Left) visualized the MSE across the grid, with colour gradients indicating the error magnitude. The lowest point on this surface—marked in red—represented the best (λ, γ) pair from this initial search. The optimum point found was λ at 0.15 and γ at 0.7, marking the area for a more detailed follow-up search.

Focused Granular Optimisation:

Following the broad optimisation, a finer search was conducted around the initially identified best values for precise pinpointing of the optimum hyperparameters. The ranges were narrowed to λ between 0.1 and 0.2 and γ between 0.65 and 0.75. The step size remained at 10 to maintain granularity.

The second 3D surface plot (Figure 3, Right) yielded the best λ at approximately 0.156, γ at approximately 0.739, and the best error at 3.143.

The granular optimisation honed in on the hyperparameters that minimized the MSE, ensuring that the Pi-Ratings model was finely tuned to predict goal differences with greater accuracy. This two-step optimisation process, starting with a broad sweep and then zeroing in with a granular search, ensured a thorough exploration of the hyperparameter space, balancing computational efficiency with precision.

3.2 Sentiment Analysis

Another feature engineered using external data is the sentiment analysis score. Sentiment analysis involves scrutinizing digital text to identify whether its emotional context leans towards passivity, negativity, or neutrality. In predicting football match outcomes, integrating sentiment analysis, particularly from social media platforms like Twitter, proves pivotal. This method captures sentiments expressed by fans, journalists, and pundits through tweets, encompassing opinions on team performance, football news, and updates—such as injury reports, transfer speculations, and tactical insights.

This amalgamation provides a holistic view for forecasting match outcomes beyond statistical analysis. Deriving sentiment insights allows us to decode the emotional context of the footballing community, enhancing result prediction accuracy. Due to Twitter API limitations, we use Nitter—a third-party front end—to access tweet data. Focused solely on recent and upcoming football matches, our approach avoids sentiment analysis of historical games. By concentrating on pre-match buzz and contemporary views, sentiment analysis captures real-time emotions which enhancing the model's predictive value.

While we implemented and experimented with sentiment analysis, these features are not part of our final model. As we have to predict the outcomes of matches in February we are not able to take the sentiment of those matches in December. The second limitation was the fact that sentiment analysis might indirectly encode the odds set by bookmakers. While we did not pursue this avenue further, the code and reasoning behind using sentiment to capture information from a wide range of data sources could be taken forward in the future.

3.3 External Data Sources

Based on our literature review, extensive match and player data is often directly correlated with better predictive power. Therefore, to develop our model further, we collected sample data from injury reports, player statistics and more granular match data, for example, possessions percentages.

Our approach was to evaluate the relevancy of the new data sets in the context of our simpler models. Through a correlation analysis and the Shapely Additive exPlanations (SHAP) analysis that is detailed under 4.2.1 we concluded that for our classifiers individual player statistics brought diminishing improvements on FTR predictions. This is likely the result of our focus on the Pi-rating scheme which proved difficult to merge with external data sources due to its rigid structure.

4 Model Training & Validation

In our endeavour to predict future matches in the English Premier League we segmented our data into varying season lengths to best determine the size of our dataset for optimal results. Through the lens of these datasets we generated synthetic features such as Pi ratings, win streaks semantic analysis scores to form the foundation of our modelling. This section provides a detailed account of the model training process shedding light on the models selected - XGBoost, CatBoost, RandomForest, and Logistic Regression - and the rationale guiding their choice.

4.1 Model Training

Within our classification framework, where the output labels 0, 1 and 2 pertain to win, lose, or draw respectively, we employed an array of training methodologies for each model. The choice of classification models is driven by the inherent nature of the prediction class, that is to categorize match outcomes into one of the three distinct classes, namely, whether a match will result in a win, loss, or draw.

4.1.1 XGBoost Classifier

The XGBoost, a classification algorithm, utilizes ensemble learning by encompassing multiple weaker models to enhance predictions. In general, it initialises the model with a single decision tree that predicts all instances as the average of the target variable, then enters a loop where new decision trees are iteratively built. This model is then used to make a prediction for a given input by aggregating the predictions of all decision trees in the ensemble. Our implementation creates an instance of the classifier with the multi:softmax objective parameter and a seed to ensure reproducibility. The classifier was trained on match-by-match data enriched with our features and then employed to predict the FTR for the test data.

4.1.2 CatBoost Classifier

The CatBoost Classifier improves predictive power by combining multiple models. Starting with a baseline decision tree, it iteratively builds new trees by computing the loss function gradient, stopping at a predefined threshold. Determining the model's depth is crucial and depends on factors like dataset complexity and feature interactions. A higher depth captures complex patterns but risks overfitting, so we balance this with combined cross-validation and grid search. Bayesian Optimization further enhances hyperparameter optimization by approximating the objective function using a surrogate model and an acquisition function, allowing us to pinpoint the best hyperparameters for our model.

4.1.3 Random Forest Classifier

Random Forest Classifiers (RFC), an ensemble machine learning algorithm, leverage multiple uncorrelated decision trees built on subsets of data and features. Each tree predicts the class for a given input, and their predictions are aggregated to form the final model prediction. This ensemble approach reduces overfitting and enhances model accuracy. We instantiated an RFC object, configuring the class weights to be balanced. This decision was prompted by the observation that draw classes were underrepresented in the dataset. By dynamically adjusting the weights of the classes based on their frequency, we aimed to mitigate potential bias in the data and ensure a better representation of all outcome categories. The model was then trained given the training set and used to make predictions on the given test set.

4.1.4 Logistic Regression

Logistic Regression (LR) is generally employed for multi-class classification and is particularly effective in binary and multi-class scenarios. It finds the optimal coefficients that maximise the likelihood of observing specific target labels given the input features. Before utilization, we conducted preprocessing on the football match prediction data, employing feature scaling - specifically, the 'StandardScaler' method - to ensure uniformity. The LR model is then trained using the scaled training data and corresponding target labels which involves iteratively optimising coefficients to enhance the model's predictive capabilities.

4.2 Model Evaluation & K-fold Cross Validation

After training, we conducted K-fold Cross Validation and evaluated our models based on the following metrics: Accuracy, F1 Score, Precision and Recall. Each of these metrics holds distinctive significance in gauging the effectiveness and reliability of the models in the predictive context.

Accuracy, as our fundamental metric, provided an overall measure of the correctness of the model's predictions by assessing the ratio of correctly predicted games to the total number of them. Simultaneously, the F1 Score, the harmonic mean of precision and recall, offers a balanced assessment of the model's performance and signifies the model's ability to identify positive cases while minimizing false positives & negatives.

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{1}$$

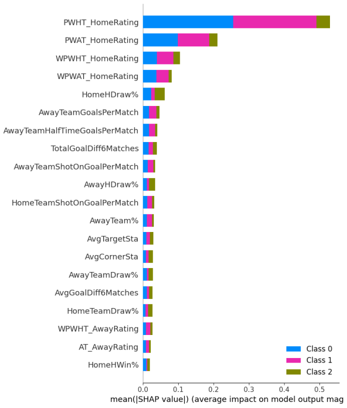

Figure 4: Graph of SHAP Values per Feature.

4.2.1 Feature Importance Selection with SHAP

To further improve the accuracy of the model, a decision was made to perform a more thorough feature selection method. For this, Shapely Additive exPlanations [1] (SHAP) was used. SHAP works by scoring each feature in terms of its direct impact on the final outcome. Through an iterative process, features with lower SHAP values are gradually removed from the data set, resulting in a remodeled data set without low-impact features. This not only improves model interpretability but also enhances efficiency and predictive power accuracy.

This methodology was applied to the (CatBoost) model which had demonstrated the highest accuracy among those explored. SHAP values were computed on (`X_train`) using a SHAP Tree Explainer, providing insights into the specific impacts of each feature on model predictions. Figure 2 highlights the power and robustness of our newly developed pairwise and weighted pairwise as they contribute to the model's output by orders of magnitude more than the original Pi-rating or descriptive statistics. A list of columns (`cols_to_drop`) was identified for removal based on SHAP values. This results in the (`X_train_refined`) data set which includes only the selected features deemed to have the most significant impact on predictive accuracy. From this, the eight least significant features were then removed, and the model was re-evaluated, achieving an even greater accuracy.

The SHAP-based feature selection led to a notable improvement in model training accuracy from 0.5011 to 0.5106, reflecting the effectiveness of identifying and removing redundant features. This approach therefore not only fine-tuned the model but also provided insights into the importance of each feature in outcome prediction.

5 Results

To report the most realistic results we incorporated the EPL matches between the release of the coursework and the 18th of December in our final evaluation. We intentionally did not expose the models to these approximately 60 matches to avoid overfitting. When these matches were added to the test set we experienced an average of 4.31% drop in the accuracy of the models. This indicated that overfitting was present to a certain extent, however, that was also apparent by our best model reporting a 56.4% accuracy which would have beaten the bookies. To keep the testing environment as realistic as possible, we did not split the datasets randomly into train and test sets. While this is a good and common practice in machine learning to avoid patterns and bias in data, in the case of sporting events where the model only ever has to predict match outcomes in the future, we decided on a sequential split.

The final results are reported in Table 1 for each model's accuracy, F1 score, precision and recall on all four data splits we trained on. The highest accuracy, 51.06% was reached with the CatBoost classifier on the past five seasons of EPL match data, while the mean accuracy of the models is 45.38%.

Model	Seasons	Accuracy	F1 Score	Precision	Recall
xgboost	All	0.458	0.456	0.456	0.458
	Five	**0.464**	**0.471**	0.489	**0.464**
	Two	0.433	0.445	**0.518**	0.433
	Past	0.400	0.407	0.436	0.400
catboostclassifier	All	0.501	0.450	0.451	0.510
	Five	**0.510**	0.504	**0.531**	0.528
	Two	0.433	0.443	0.493	0.433
	Past	0.50	**0.524**	0.514	**0.560**
randomforestclassifier	All	0.442	0.435	0.452	0.426
	Five	0.4494	0.441	0.468	0.429
	Two	**0.483**	0.504	0.527	0.500
	Past	0.360	**0.545**	**0.547**	**0.560**
logisticregressionmodel	All	0.485	0.449	0.451	0.486
	Five	0.437	0.436	0.448	0.437
	Two	**0.508**	**0.494**	**0.502**	**0.508**
	Past	0.440	0.436	0.435	0.440

Table 1: Performance of trained models

6 Final Predictions

To make our final predictions the `epl-test.csv` dataset was cleaned. Firstly, the date strings were converted such that Pandas could recognise them as dates. Secondly, the team names were checked to exactly match the naming in the training sets. Lastly, the test set consisting of Date, HomeTeam and AwayTeam columns was fed through the same function (`construct_table`) as the training data to construct the same set of features.

Taking an initial look at the FTR predictions in Table 2, they are in line with the current performance of teams which gives us hope that predictions will live up to the 50% accuracy we observed on our own test sets. It was promising to see that the model predicted Draw outcome which is known to be an underrepresented class and the pitfall of bookmakers.

Date	HomeTeam	AwayTeam	FTR
2024-02-03	Bournemouth	Nott'm Forest	A
2024-02-03	Arsenal	Liverpool	A
2024-02-03	Brentford	Man City	A
2024-02-03	Brighton	Crystal Palace	D
2024-02-03	Burnley	Fulham	A
2024-02-03	Chelsea	Wolves	H
2024-02-03	Everton	Tottenham	A
2024-02-03	Man United	West Ham	H
2024-02-03	Newcastle	Luton	A
2024-02-03	Sheffield United	Aston Villa	A

Table 2: Final Predictions

7 Conclusion

In conclusion, we successfully carried out the task of predicting the full-time result of EPL matches. We began by gaining an understanding of the data set we were provided which led to experimentation with external data sets on player statistics as well as sentiment analysis to capture information such as player transfers, injuries or match stakes that otherwise would have required niche or impossible to acquire data. Even though these approaches were not used in the final models, they give a sensible starting point for future work. Perhaps the greatest strength of this coursework is the originality of the exponential time-decay adjusted pairwise and weighted pairwise Pi-rating scheme that provide a new, robust and accurate rating of teams. Lastly, we employed several machine learning classifiers to predict match results and reached a 51.06% accuracy that is on par with the 53% "gold standard".

References

[1] Yiming Ren and Teo Susnjak, *"Predicting Football Match Outcomes with eXplainable Machine Learning and the Kelly Index,"* School of Mathematical and Computational Sciences, Massey University, Auckland, New Zealand, November 30, 2022. [Online]. Available: `https://arxiv.org/pdf/2211.15734.pdf`

[2] Anthony Constantinou and Norman Fenton, *"Determining the level of ability of football teams by dynamic ratings based on the relative discrepancies in scores between adversaries," Journal of Quantitative Analysis in Sports*, vol. 9, pp. 37-50, 2013. [Online]. Available: `https://doi.org/10.1515/jqas-2012-0036`.

[3] Clarke and Norman, *"Home Ground Advantage of Individual Clubs in English Soccer."*, vol. 44, No. 4 (1995) [Online]. Available: `https://www.jstor.org/stable/2348899`

[4] Hirotsu and Wright, *"An evaluation of characteristics of teams in association football by using a Markov process model."*, 19 November 2003 [Online]. Available: `https://rss.onlinelibrary.wiley.com/doi/abs/10.1046/j.0039-0526.2003.00437.x`

[5] Damien Poulter, *"Home advantage and player nationality in international club football."*, 23 Jun 2009 [Online]. Available: `https://www.tandfonline.com/doi/abs/10.1080/02640410902893364`

[6] Constantinou et al., *"A Bayesian network model for forecasting Association Football match outcomes. Knowledge-Based Systems."*, December 2012 [Online]. Available: `https://www.researchgate.net/publication/236944355_pi-football_A_Bayesian_network_model_for_forecasting_Association_Football_match_outcomes_Knowledge-Based_Systems_36_322-339`

[7] Scott M. Lundber and Su-In Lee, *"A Unified Approach to Interpreting Model Predictions."*, 2017 [Online]. Available: `https://papers.nips.cc/paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf`

[8] Benjamin Holmes, Ian G. McHale, *"Forecasting football match results using a player rating based model"*, 2023 [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S016920702300033X`