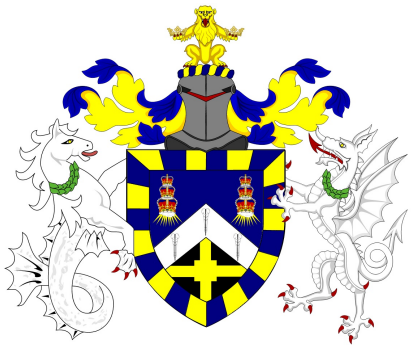
Data Analytics MSc Dissertation MTHM038, 2023/24

Forecasting Team Points and Winner of EPL Using Historical Football Data

Pranav Nigam, ID 230138438

Supervisor: Dr. Hong Qi



A dissertation presented for the degree of
Master of Science in Data Analytics

School of Mathematical Sciences
Queen Mary University of London
September 1, 2024

Declaration of original work

This declaration is made on September 1, 2024.

Student's Declaration: I Pranav Nigam, hereby declare that the work in this dissertation is my original work. I have not copied from any other students' work, work of mine submitted elsewhere, or from any other sources except where due reference or acknowledgment is made explicitly in the text. Furthermore, no part of this dissertation has been written for me by another person, or by AI-assisted technologies.

Referenced text has been flagged by:

1. Using *italic fonts*, and
2. Using quotation marks "...", and
3. Explicitly mentioning the source in the text.

Acknowledgements

I would like to extend my sincere gratitude to my supervisor, Dr. Hong Qi, for their invaluable guidance, support, and encouragement throughout the course of this dissertation. Their expertise and insights played a crucial role in shaping the direction and outcome of my research. I am deeply thankful to my family and friends for their unwavering support and understanding during the more challenging moments of this project. Their patience and belief in my abilities provided the motivation I needed to persevere. I would also like to acknowledge the faculty and staff of the School of Mathematical Sciences at Queen Mary University of London for offering the necessary resources and fostering a supportive learning environment throughout my time in the MSc Data Analytics program. Finally, my appreciation goes out to my fellow students and colleagues, whose advice, shared knowledge, and constructive feedback were invaluable throughout the research process. Your contributions have been greatly appreciated.

Abstract

In this dissertation, I embark on an exploration of predictive modeling to unravel the complexities of English Premier league outcomes. By diving deep into historical match data spanning multiple seasons, I crafted a robust model that merges various team performance metrics and match statistics to **forecast each team's final points** and ultimately **predict the league champion**. The journey began with meticulous feature engineering, transforming raw data into meaningful insights that could fuel predictive accuracy. This foundation set the stage for the deployment of three powerful machine learning models: Logistic Regression, XGBoost, and Random Forest Classifier. Each model was rigorously tested and validated using key performance metrics, with a spotlight on accuracy to ensure precision in predicting final points and league winners. Among the models, the **Random Forest Classifier** emerged as the standout performer, achieving an impressive accuracy rate of **95**. This model consistently demonstrated its prowess by accurately forecasting the final points and pinpointing the league champion. These results not only underscore the potential of machine learning in sports analytics but also provide valuable tools for football clubs, analysts, and enthusiasts aiming to gain a competitive edge in predicting league outcomes. As the field of sports analytics continues to evolve, this dissertation offers a stepping stone for future research, advocating for the integration of additional data sources and the exploration of more sophisticated modeling techniques to push the boundaries of predictive accuracy even further.

Contents

Chapter 1

Introduction

1.1 Background

The integration of data analytics into football has revolutionized the analysis, management, and experience of the sport, with the English Premier League (EPL) leading this transformation. The EPL, known for its intense competition and global audience, functions as a major business entity where data-driven decisions can significantly influence both on-field performance and financial success. Each season generates extensive data—from match outcomes and player statistics to tactical strategies and fan sentiment—offering a wealth of resources for applying advanced analytics and machine learning techniques[1][15]. Traditionally, Premier League predictions relied on the expertise and intuition of pundits, coaches, and analysts. While their insights remain valuable, the vast and complex data available today has enabled the adoption of more scientific forecasting methods. This shift mirrors broader trends in sports where data analytics has become central to understanding performance and strategy. For example, clubs like Liverpool and Manchester City have leveraged analytics to enhance everything from player recruitment to match tactics, contributing to their recent successes[2]. Predictive modeling, in particular, has made a substantial impact on forecasting match

outcomes and league standings. These models now consider various factors such as player injuries, mid-season managerial changes, and external conditions like weather or travel demands, leading to more accurate predictions that give teams a competitive edge. Beyond individual matches, predictive analytics supports long-term planning and decision-making for football clubs, including optimizing squad rotation, assessing the impact of new signings, and forecasting financial outcomes based on team performance[23]. Moreover, the influence of predictive modeling extends beyond the football clubs to industries like betting, where sophisticated algorithms set odds based on a wide array of variables. As the accuracy of these models improves, their impact on the betting market grows, with even small shifts in odds having significant financial implications. The EPL's global reach and diverse fan base further complicate predictive modeling, as clubs face scrutiny both locally and internationally, driving greater investment in analytics[24]. This dissertation investigates the use of predictive modeling to forecast final points for each team and determine the league champion in the Premier League. By analyzing historical data from multiple seasons, the study aims to address the unique challenges of the Premier League, such as competitive balance, and performance variations between home and away matches. The goal is to enhance prediction accuracy and provide actionable insights that benefit football clubs, analysts, and the broader community of Premier League stakeholders.

1.2 Research Objectives

This dissertation aims to develop and evaluate predictive models designed to accurately forecast key outcomes in the English Premier League (EPL). The primary focus is on creating a model capable of estimating the final points for each team by analyzing historical data alongside essential performance metrics. Through this approach, the study seeks to pinpoint the factors that

significantly contribute to a team's success over the course of a season . In addition to predicting final points, the dissertation aims to forecast the eventual league champion. This involves using predictive modeling techniques to aggregate these predictions and determine the overall league standings. The study will also conduct a comparative evaluation of several machine learning models—including Logistic Regression, XGBoost, and Random Forest Classifier—to identify which model delivers the most accurate and reliable predictions for both team points and the league winner . Furthermore, the research will investigate the impact of various performance metrics, such as goals scored, goals conceded, and possession statistics, on the accuracy of these predictions. By examining these factors, the study aims to identify the critical determinants of success in the Premier League . The overarching goal of this research is to offer valuable insights for stakeholders, including football clubs, analysts, and fans, who seek to deepen their understanding of the Premier League through data-driven analysis. This dissertation aspires to contribute to the expanding field of sports analytics by providing a comprehensive framework for predicting league outcomes with enhanced accuracy .

Chapter 2

Literature Review

2.1 Predictive Modeling in Football

Sports analytics has become an integral part of data science, revolutionizing decision-making processes across various sports, from player recruitment to in-game tactics and post-match assessments. Although the application of analytics in sports is not entirely new, its complexity and influence have significantly expanded due to advances in computational power and the growing availability of detailed data .

Historical Context and Evolution: The roots of sports analytics can be traced back to early statistical practices, where basic metrics like batting averages in baseball or goals-per-game in football were used to assess player performance. However, the modern era of sports analytics emerged prominently in the late 20th and early 21st centuries, particularly with the "Moneyball" strategy in Major League Baseball (MLB). Popularized by Michael Lewis's book Moneyball (2003), this approach demonstrated how data-driven decision-making could upend traditional scouting and player evaluation methods[6]. The Oakland Athletics' success, despite a limited budget, underscored the potential of analytics to offer a competitive edge, even in resource-constrained environments. Following the success of Mon-

eyball, other sports began integrating analytics. In basketball, advanced metrics such as Player Efficiency Rating (PER) and effective field goal percentage became standard tools. In American football, data-driven decisions have increasingly shaped play-calling, player health management, and contract negotiations. This broader acceptance of analytics across various sports laid the foundation for its adoption in football (soccer), where the game's complexity and vast data generation provide rich opportunities for comprehensive analysis[7] .

Sports Analytics in football: Predictive modeling is a cornerstone of sports analytics, enabling stakeholders to forecast outcomes such as match results, player performance, and league standings. The use of predictive modeling in sports is diverse, ranging from simple regression models to sophisticated machine learning algorithms . Early predictive models in sports often utilized linear regression to predict outcomes like match scores based on variables such as team strength, form, and historical results. While these models provided a basic framework for predictions, their accuracy was often limited by their linear assumptions, which failed to capture the non-linear relationships inherent in sports data. The advent of machine learning has revolutionized predictive modeling in sports. Machine learning models, including decision trees, random forests, and neural networks, can process large datasets and identify complex patterns that traditional models might miss. These models are particularly well-suited to the dynamic and unpredictable nature of sports. For instance, Random Forest, an ensemble learning method, has been used to predict football match outcomes by incorporating a wide range of variables, such as team formations, player injuries, and even referee biases . The field of predictive modeling in football has seen considerable growth in recent years, fueled by the surge in detailed match data and advancements in machine learning technologies. Forecasting football outcomes, including match results, league rankings, and player performances, has emerged as a key area of interest in sports analytics, with researchers

employing various methods to enhance prediction accuracy. One of the key studies in this field is by Constantinou et al. (2012), which introduced a Bayesian network model for predicting football match outcomes. This model integrated various factors related to the match, such as team strength, home advantage, and historical head-to-head records, to estimate the likelihood of different outcomes (win, draw, loss). The study showed that incorporating domain-specific insights into predictive models could greatly improve their accuracy compared to traditional statistical methods, especially in accounting for the inherent uncertainty in football matches[25]. Another significant contribution to the literature is the work by Groll et al. (2018), which used a random forest model to predict the outcomes of the FIFA World Cup 2018. This study emphasized the critical role of feature selection and model tuning in predictive modeling, demonstrating that incorporating factors like team strength, player experience, and recent form could enhance prediction accuracy. The random forest model proved particularly effective at managing the complex interactions among these variables, providing a strong approach to forecasting football outcomes[16]. These studies highlight the potential of machine learning techniques, especially those capable of capturing non-linear relationships and interactions between variables, in improving the accuracy of football predictions. As the field advances, future research is likely to explore more advanced models, including deep learning algorithms, to further enhance predictive precision.[3][17].

2.2 Predictive Modelling Techniques

Several techniques are fundamental to sports forecasting due to their ability to handle diverse datasets and generate reliable predictions. This section reviews the three machine learning models utilized in this research: Logistic Regression, XGBoost, and Random Forest Classifier. Each model is examined in detail, focusing on their mathematical foundations, strengths, and

relevance to sports forecasting, particularly in predicting football match outcomes and league standings.

Logistic Regression

Logistic regression is a fundamental statistical method used for binary classification problems, where the outcome variable has two possible states, typically labeled as 0 and 1. In sports forecasting, logistic regression is often applied to predict the likelihood of binary events, such as whether a football team will win or lose.

The logistic regression model works by estimating the probability $P(Y = 1 \mid X)$ based on the input features X. The model relies on the sigmoid function which is a logistic function, that converts any real-valued number into a probability value between 0 and 1, making it ideal for estimating probabilities.

The logistic function is given by:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{1.1}$$

where $z = \beta_0 + \sum_{i=1}^{n} \beta_i X_i$, β_0 is the intercept, β_i are the coefficients for each feature X_i, and e represents the base of the natural logarithm[5][20].

Therefore, the probability of the event $Y = 1$ is expressed as:

$$P(Y = 1 \mid X) = \frac{1}{1 + e^{-(\beta_0 + \sum_{i=1}^{n} \beta_i X_i)}} \tag{1.2}$$

XGBoost

XGBoost, which stands for Extreme Gradient Boosting, is a highly efficient and scalable machine learning algorithm known for its accuracy and speed, especially in structured data problems. XGBoost builds an ensemble of decision trees, where each new tree is added sequentially to correct the errors of the previous trees.

XGBoost operates by optimizing an objective function that combines a loss function L with a regularization term Ω, which penalizes model complexity to prevent overfitting[12][21].

The prediction for a given input X_i is calculated as the sum of the predictions from each of the K trees in the ensemble:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(X_i) \tag{2.1}$$

where f_k represents each tree in the ensemble. The objective function minimized by XGBoost is:

$$\text{Obj}(\Theta) = \sum_{i=1}^{N} L(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{2.2}$$

In this equation, $L(y_i, \hat{y}_i)$ measures the difference between the true values y_i and the predicted values $\hat{y}_i$, while $\Omega(f_k)$ represents the regularization term for each tree. **Gradient Boosting**
XGBoost updates the model at each iteration by adding a new tree that aims to minimize the residuals (errors) of the current model:

$$\hat{y}_i^{(t+1)} = \hat{y}_i^{(t)} + \eta f_{t+1}(X_i) \tag{2.3}$$

where η is the learning rate, and $f_{t+1}(X_i)$ represents the tree trained on the residuals. The learning rate η controls the contribution of each new tree to the overall model.

Random Forest Classifier
Random Forest is an ensemble machine learning technique that constructs multiple decision trees and then combines their predictions to improve accuracy and reduce the risk of overfitting. By introducing randomness in both

the sampling of data and the selection of features for each tree, Random Forest reduces the correlation among the trees and enhances the overall model's robustness.

A Random Forest consists of B decision trees, each of which is built using a bootstrap sample (a sample drawn with replacement) of the training data. Each tree is further diversified by selecting a random subset of features at each split.

Each decision tree splits the data recursively based on feature values, aiming to create homogeneous subsets. The quality of a split is often measured using the Gini impurity or entropy. The Gini impurity for a node is calculated as:

$$Gini = 1 - \sum_{j=1}^{C} p_j^2 \tag{3.1}$$

where p_j represents the proportion of instances in class j within the node, and C is the total number of classes. The best split is chosen based on the reduction in Gini impurity (or other criteria).

Random Forest introduces randomness by creating bootstrap samples of the data for each tree and selecting a random subset of features at each split. This approach reduces the correlation between the trees, improving the ensemble's ability to generalize to new data[13][22].

The final prediction of a Random Forest is obtained by aggregating the predictions of all individual trees.

$$\hat{y} = \text{mode}\{T_1(X), T_2(X), \ldots, T_B(X)\} \tag{3.2}$$

where $T_1, T_2, \ldots, T_B$ are the decision trees in the forest.

Chapter 3

Methodology

3.1 Data Collection and Preparation

The dataset utilized in this study was sourced from *FootballData.co.uk*, a well-established provider of historical football statistics. The data encompasses multiple seasons of the English Premier League (EPL), offering a comprehensive and detailed view of team performances, match outcomes, and various other metrics essential for predictive modeling. In this study, match data from the English Premier League (EPL) was collected for eight consecutive seasons, spanning from the 2016-2017 season to the 2023-2024 season. Each season's dataset was stored in a CSV (Comma-Separated Values) file, a format that is widely used for storing and exchanging tabular data. The datasets were carefully curated to include relevant match statistics for each season, providing a comprehensive overview of team performances across multiple years. To manage the data efficiently, a dictionary was created where each key represented a specific season (e.g., "2016-2017"), and the corresponding value was a pandas DataFrame containing the match data for that season. The use of a dictionary allowed for organized storage and easy access to each season's data, facilitating streamlined processing and analysis. The datasets were imported into the analysis environment using the pan-

das library in Python, which offers robust data manipulation capabilities. The process of loading the datasets involved iterating through a list of file paths corresponding to each season's CSV file. The season name was extracted from the file path, which was then used as the key in the dictionary to store the DataFrame. The preparation of data is a critical step in the analytical process, as it ensures the quality and consistency of the data used for modeling. After loading the datasets, the next step involved inspecting the features (columns) contained within each dataset. This inspection was necessary to verify that all datasets had a consistent structure and that all relevant variables were present across all seasons. A thorough examination of the dataset features was conducted to identify any discrepancies, such as missing columns, differences in naming conventions, or changes in data recording methods over the years. This step was essential for ensuring that the data from different seasons could be seamlessly integrated and compared during the analysis phase. The inspection of features provided insights into the uniformity of the datasets, allowing for the identification of any inconsistencies that needed to be addressed. For example, if a particular feature was absent in some seasons, additional steps were taken to either impute the missing data or exclude the feature from the analysis to maintain consistency across the datasets. Ensuring the consistency of features across all datasets is vital because it allows for a uniform approach in the subsequent modeling stages. Consistent data structure ensures that the models developed can be applied across all seasons without the risk of bias or errors due to missing or inconsistent data. This careful attention to data preparation lays the groundwork for robust and reliable model predictions.

3.2 Feature Engineering

Feature engineering is a critical step in the machine learning pipeline, involving the creation and transformation of features to improve model accuracy

and performance. In this study, feature engineering was performed to derive meaningful variables from the raw match data, providing the models with enriched information to accurately predict team performance and league outcomes. This section details the process of generating these features, including the rationale behind each transformation. The primary goal of this project was to predict league outcomes, including which team would win the championship. To achieve this, it was essential to calculate the points each team earned throughout the season, as points directly determine the league standings. In the English Premier League, the point system is straightforward: teams receive 3 points for a win, 1 point for a draw, and no points for a loss. A custom function was created to automate the calculation of points for each match. This function evaluated the match result, stored in the `FTR` (Full-Time Result) column, and assigned points accordingly. **HomePoints**: If the home team won (`FTR` = 'H'), 3 points were assigned to the home team and 0 to the away team. If the match was a draw (`FTR` = 'D'), both teams received 1 point. If the away team won (`FTR` = 'A'), the away team received 3 points, and the home team received none. **AwayPoints**: This column mirrored the logic used for `HomePoints`, ensuring that points were accurately recorded for both teams involved in the match. These newly created features, `HomePoints` and `AwayPoints`, were fundamental in summarizing each team's performance throughout the season, providing a direct measure of their success in terms of match outcomes. A custom function was developed to parse the match date, stored in the `Date` column, and extract the year. The function was designed to handle various date formats that were present in the datasets. The extracted year was stored in a new column, `Year`, which was then used to categorize matches into their respective seasons. Additionally, this step ensured that all subsequent feature engineering tasks were correctly aligned with the specific season each match belonged to, enabling accurate temporal analysis of team performances over multiple seasons.

Home and Away Performance Metrics In football, team performance can vary significantly between home and away matches due to factors such as crowd support, familiarity with the pitch, and travel fatigue. To capture these nuances, separate features were created to quantify a team's performance in home and away games. Home Performance Metrics: The datasets were grouped by the `HomeTeam` column, and several key statistics were aggregated. The season in which the matches were played was recorded, ensuring that the features were correctly aligned with the time frame of the analysis. The total points accumulated by the home team over the season were calculated by summing the `HomePoints` column. The total number of goals scored by the home team in all home matches was aggregated from the `FTHG` (Full-Time Home Goals) column. The total number of goals conceded by the home team was aggregated from the `FTAG` (Full-Time Away Goals) column. The total number of matches won, drawn, and lost by the home team was derived by counting the occurrences of 3 points (win), 1 point (draw), and 0 points (loss) in the `HomePoints` column. Away Performance Metrics: Similarly, performance metrics were calculated for the away team by grouping the datasets by the `AwayTeam` column. The season was recorded to ensure the data was correctly attributed. The total points accumulated by the away team over the season were summed from the `AwayPoints` column. The total number of goals scored by the away team in all away matches was aggregated from the `FTAG` (Full-Time Away Goals) column. The total number of goals conceded by the away team was aggregated from the `FTHG` (Full-Time Home Goals) column. The total number of matches won, drawn, and lost by the away team was derived from the `AwayPoints` column in a similar manner to the home metrics. These features were essential for understanding the strengths and weaknesses of teams in different contexts, providing a nuanced view of their overall performance.

Combining Home and Away Statistics To gain a complete picture of each team's overall performance, the home and away statistics were combined into a single set of features for each season. This involved merging the home and away metrics on the `Team` and `Season` columns, resulting in a comprehensive dataset that included the sum of `Home Points` and `Away Points`, representing the team's total points for the season. The sum of `Goals Scored (Home)` and `Goals Scored (Away)` reflected the team's overall offensive capabilities. The sum of `Goals Conceded (Home)` and `Goals Conceded (Away)` provided insight into the team's defensive strength. The aggregate number of wins, draws, and losses from both home and away matches offered a comprehensive view of the team's consistency and resilience throughout the season. This combined feature set was critical for evaluating each team's performance over the entire season, allowing the models to consider all aspects of their match history when making predictions.

Season Summary and Champion Identification In addition to team-specific features, a summary of each season's overall statistics was created to provide context for the league as a whole. This summary included metrics such as the total number of teams competing in the league each season, the average number of points earned by teams, providing a measure of the league's competitive balance, and the average goals scored and conceded per team, reflecting general offensive and defensive trends within the league. Furthermore, a key feature was the `Champion` indicator, which identified the team with the highest total points at the end of each season. This binary feature, with a value of 1 for the champion and 0 for all other teams, was pivotal in training the models to predict which team would win the league. This feature engineering process transformed raw match data into a well-structured dataset, ready for use in predictive modeling.

3.3 Exploratory Data Analysis

Exploratory Data Analysis (EDA) is a crucial phase in data analysis that focuses on summarizing the key features of the dataset, typically through visual techniques. The main objective of EDA is to uncover insights, identify patterns, detect anomalies, and validate hypotheses. This section highlights the significant findings from the EDA conducted on the Premier League dataset utilized in this study.

Descriptive Statistics Table 3.1 presents the descriptive statistics for key variables in the dataset, including points earned at home and away, goals scored and conceded both at home and away, as well as aggregate statistics like total points, total goals scored, and total wins.

Correlation Matrix The correlation matrix is a unique tool in exploratory data analysis, as it provides the relationships between different numerical features in the dataset. Here, the correlation matrix was computed using the numeric columns from the combined seasons, focusing on key team statistics such as points, goals scored, and goals conceded both at home and away.

Distribution of Wins, Draws, and Losses Understanding the distribution of wins, draws, and losses across teams and seasons provides insight into the competitive nature of the Premier League. In this study, a histogram was plotted to visualize the frequency of total wins, draws, and losses recorded by teams across multiple seasons.

Distribution of Total Points Across all the Seasons The analysis shows the distribution of total points across teams and seasons, highlighting that most teams accumulate points within a middle range, with fewer teams at the extremes. This provides insight into the competitive balance of the league.

Statistic	Points_Home	GoalsScored_Home	GoalsConceded_Home	Wins_Home	Draws_Home	Losses_Home	Points_Away	GoalsScored_Away
Count	160.00	160.00	160.00	160.00	160.00	160.00	160.00	160.00
Mean	30.14	29.72	24.11	8.62	4.29	6.09	22.57	24.11
Std	10.27	11.09	7.66	3.68	1.96	3.27	9.97	8.80
Min	8.00	9.00	9.00	2.00	0.00	0.00	5.00	7.00
25%	23.00	21.75	18.00	6.00	3.00	4.00	15.00	18.00
50%	28.50	27.00	24.00	8.00	4.00	6.00	20.00	23.00
75%	37.00	36.00	28.00	11.00	6.00	8.00	29.25	31.00
Max	55.00	61.00	57.00	18.00	10.00	15.00	50.00	45.00
GoalsConceded_Away	**Wins_Away**	**Draws_Away**	**Losses_Away**	**TotalPoints**	**TotalGoalsScored**	**TotalGoalsConceded**	**TotalWins**	**TotalDraws**
29.72	6.09	4.29	8.62	52.71	53.83	53.83	14.71	8.57
8.42	3.44	1.85	3.43	18.68	18.53	14.23	6.54	2.86
11.00	1.00	0.00	0.00	16.00	20.00	22.00	3.00	2.00
24.00	4.00	3.00	7.00	40.00	40.00	44.00	10.00	7.00
30.00	5.00	4.00	9.00	49.00	51.00	54.00	13.00	8.00
36.00	9.00	5.00	11.00	66.00	66.25	63.25	19.00	10.00
48.00	16.00	9.00	16.00	100.00	106.00	104.00	32.00	15.00
TotalLosses	**Champion**							
14.71	0.05							
6.03	0.22							
1.00	0.00							
11.00	0.00							
15.50	0.00							
19.00	0.00							
29.00	1.00							

Table 3.1: Descriptive Statistics for Key Variables in the Premier League Dataset

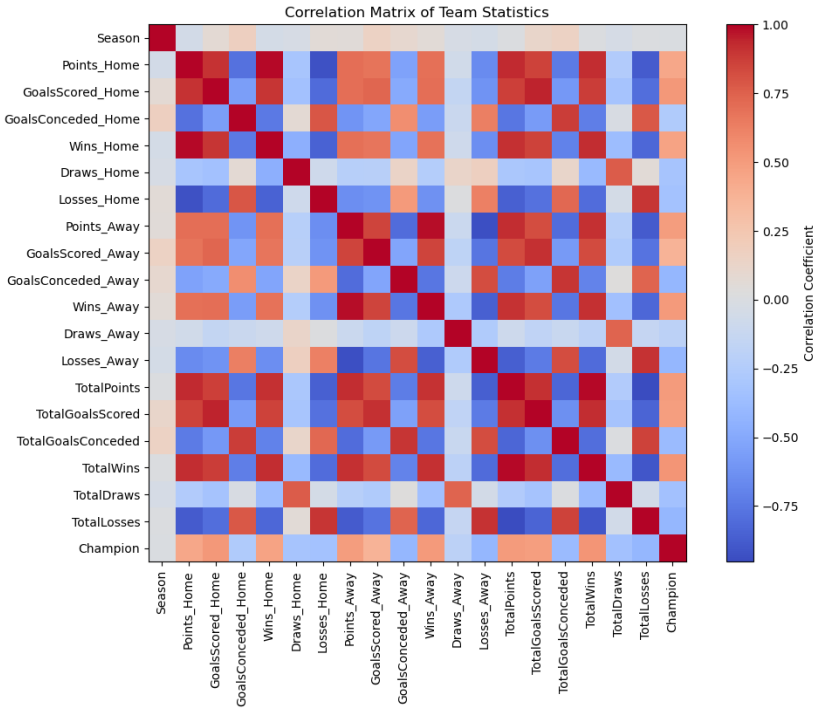


Figure 3.1: Correlation Matrix of Team Statistics

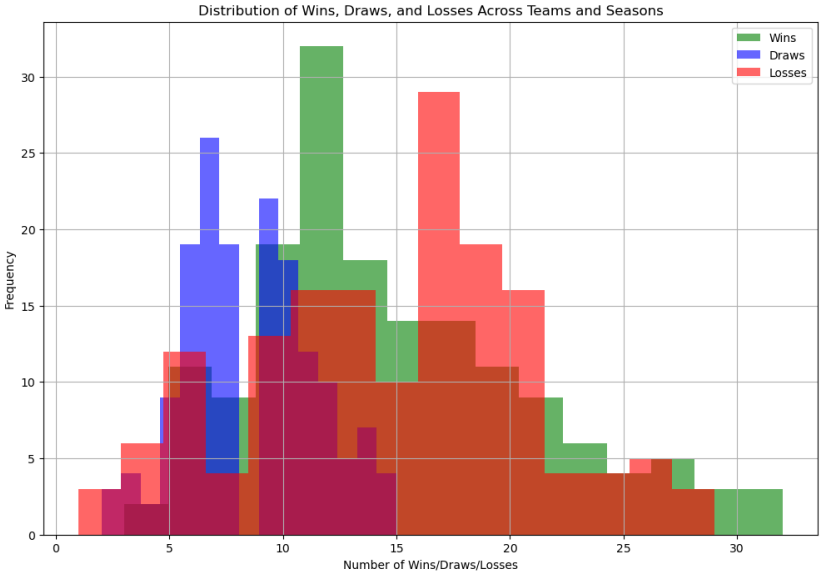


Figure 3.2: Distribution of Wins, Draws, and Losses Across Teams and Seasons

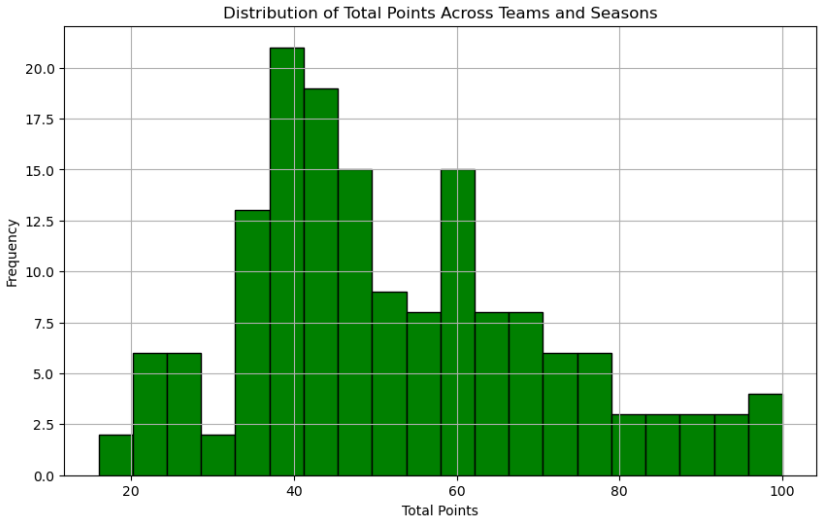


Figure 3.3: Points Distribution

Chapter 4

Data Analysis and Model Implementation

Data Splitting for Training and Testing To effectively train and evaluate the predictive models, the dataset, which covers multiple seasons of the English Premier League from 2016 to 2024, was split into training and testing sets. **The training set consists of data from the 2016-2017 season through to the 2022-2023 season**, while the **testing set is comprised of data from the 2023-2024 season**. This splitting strategy ensures that the models are trained on historical data and tested on the most recent season, thereby providing a realistic assessment of their predictive capabilities.

Feature Selection The features selected for this study represent key performance metrics that are directly related to a team's success in the league. These features were carefully chosen to encapsulate a team's overall performance, both in home and away matches. **Points_Home** represents the total number of points earned by the team in home matches throughout the season, capturing the team's ability to secure wins and draws on home ground. **GoalsScored_Home** is the total number of goals scored by the team in home matches, indicative of the team's offensive strength when playing at home.

GoalsConceded_Home reflects the total number of goals conceded by the team in home matches, highlighting the team's defensive capabilities on home turf. **Wins_Home**, **Draws_Home**, and **Losses_Home** represent the total number of wins, draws, and losses the team experienced in home matches, providing a granular view of the team's consistency and performance at home. **Points_Away** captures the total number of points earned by the team in away matches, highlighting the team's ability to perform under the challenging conditions of away games. **GoalsScored_Away** indicates the total number of goals scored by the team in away matches, demonstrating their offensive prowess in unfamiliar environments. **GoalsConceded_Away** is the total number of goals conceded by the team in away matches, providing insight into the team's defensive resilience while playing away from home. **Wins_Away**, **Draws_Away**, and **Losses_Away** denote the total number of wins, draws, and losses the team recorded in away matches, offering a detailed perspective on their performance across different match settings.

Creation of Target Variables In addition to the features, target variables were created to define the specific outcomes the models were tasked with predicting. Two primary target variables were established. The first target variable, `Champion`, is a binary indicator of whether a team won the league in a given season. This variable is set to 1 for the team with the maximum total points at the end of the season and 0 for all other teams. This target is crucial for binary classification tasks aimed at predicting the league winner. The second target variable, `TotalPoints`, is a continuous variable representing the total points earned by a team during the season. Unlike the binary `Champion` variable, `TotalPoints` is used in regression tasks to predict the exact number of points a team will accumulate by the end of the season. The `Champion` variable was derived by ranking the teams within each season according to their total points. For the `Champion` variable, the team with the highest points was assigned a value of 1, while all other teams were

assigned 0. This ranking and assignment process ensured that the models were trained to identify the league winner accurately.

Extracting Features and Target Variables Once the features and target variables were defined, they were extracted from the training and testing datasets. The training features (`X_train`) and testing features (`X_test`) were selected based on the previously defined feature set, ensuring consistency in the data used across different models. Corresponding target variables for predicting the league champion (`y_train_winner` and `y_test_winner`) and total points (`y_train_points` and `y_test_points`) were also extracted. This structured extraction process provided a clear framework for training and testing the models, enabling precise and consistent evaluation of their performance across different predictive tasks. The careful alignment of features and target variables with the training and testing datasets ensured that the models were equipped with the necessary data to learn from past seasons and apply that knowledge to the prediction of future outcomes.

Logistic Regression Implementation Logistic Regression was selected as the baseline model for this study due to its effectiveness in binary classification tasks, particularly for predicting whether a team would win the league (the `Champion` variable). The simplicity and interpretability of Logistic Regression make it an ideal starting point for understanding the relationships between the selected features and the target variable. The model was implemented using the `LogisticRegression` class from the scikit-learn library in Python. The key parameters included setting the maximum iterations (`max_iter`) to 1000 to ensure convergence and the random state (`random_state`) to 42 for reproducibility. These settings provided a balance between model performance and computational efficiency. The `predict` method was used to generate predictions for the league winner, with results stored in `y_pred_winner_logreg`. The model achieved an accuracy of 90%, correctly predicting the outcome for 9 out of 10 instances. However, the

classification report indicated that while the model performed well in predicting the majority class (teams that did not win the league), it struggled with the minority class (the actual league winner), resulting in low precision and recall for this category. This highlights a common issue in binary classification with imbalanced datasets, where the model may become biased towards the majority class. These findings underscore the limitations of Logistic Regression in this context, particularly its challenges with imbalanced data when predicting the league winner. The insights gained from this analysis informed the selection and tuning of more complex models, such as XGBoost and Random Forest, which were explored in subsequent analyses.

XGBoost Implementation XGBoost (Extreme Gradient Boosting) was selected for this study due to its robust performance in structured data problems and its ability to handle complex interactions between features. XGBoost, an ensemble learning technique, constructs multiple decision trees sequentially, with each new tree correcting the errors of the previous ones. In this research, XGBoost was employed to predict two key outcomes: whether a team would win the league (the `Champion` variable) and the total points a team would accumulate by the end of the season (the `TotalPoints` variable). The versatility of XGBoost, coupled with its capacity to manage both classification and regression tasks, made it an ideal choice for these predictions. The XGBoost models were implemented using the `XGBClassifier` for predicting the league winner and `XGBRegressor` for predicting total points, both from the XGBoost library in Python. Key hyperparameters were set to optimize performance: the number of estimators (`n_estimators`) was set to 50, balancing performance and computational complexity, while the maximum depth (`max_depth`) of 3 was chosen to reduce the risk of overfitting. The evaluation metric (`eval_metric`) for classification was `logloss`, and the random state (`random_state`) was set to 42 to ensure reproducibility. These hyperparameters were carefully selected to provide a balance between model

accuracy and efficiency. The models were trained on the training dataset, with the `xgb_winner` model predicting whether a team would win the league (`y_train_winner`) and the `xgb_points` model predicting the total points a team would earn by the end of the season (`y_train_points`). After training, the models were applied to the test set for the 2023-2024 season, generating predictions stored in `y_pred_winner_xgb` for the league winner and `y_pred_points_xgb` for the total points. These predictions allowed for a comprehensive evaluation of team performance. The XGBoost model achieved an accuracy of 95% for predicting the league winner, indicating a strong overall performance. However, similar to Logistic Regression, it faced challenges in correctly identifying the minority class (the league winner), resulting in lower precision and recall for this category. This issue, common in binary classification with imbalanced datasets, highlighted the need for further optimization. The results demonstrated XGBoost's robustness and its superior ability to handle complex prediction tasks compared to Logistic Regression, although the challenges associated with class imbalance suggest that additional techniques, such as oversampling or different evaluation metrics, could be explored to improve performance further.

Random Forest Classifier Implementation Random Forest is a widely recognized ensemble learning method effective for handling large datasets with numerous features and robust against overfitting. It constructs multiple decision trees during training and outputs either the mode for classification tasks or the mean prediction for regression tasks. This approach improves accuracy by reducing the variance associated with individual decision trees. For this study, Random Forest was chosen to predict two key outcomes: whether a team would win the league (`Champion`) and the total points a team would accumulate by the end of the season (`TotalPoints`). The model's ability to handle non-linear relationships and feature interactions makes it particularly well-suited for these tasks. The Random Forest models were implemented

using the `RandomForestClassifier` and `RandomForestRegressor` classes from the scikit-learn library in Python. Key hyperparameters included 100 estimators (`n_estimators`) and a random state of 42 (`random_state`) to ensure reproducibility. These settings balanced performance and computational efficiency, allowing the model to capture complex patterns without overfitting. The models were trained on the selected features and target variables. The `rf_winner` model was trained to predict the league champion (`y_train_winner`), while the `rf_points` model was trained to predict the total points (`y_train_points`). During training, each tree was built using a different bootstrap sample of the data, with only a random subset of features considered for splitting at each node. This method enhances the model's ability to generalize to unseen data. After training, the models were used to predict the outcomes for the 2023-2024 season. The predictions for the league winner were stored in `y_pred_winner_rf`, and the predicted points were stored in `y_pred_points_rf`. The Random Forest model achieved a high accuracy of 95% for predicting the league winner, demonstrating its effectiveness in capturing non-linear relationships and interactions between features. However, while the precision was high, the recall for the winning class was slightly lower. The model also effectively predicted the total points, offering reliable estimates for the majority of teams. The ability of Random Forest to provide feature importance rankings added valuable insights into the key factors influencing both the league winner and total points predictions.

Chapter 5

Results and Discussion

5.1 Predicted Points and League Winner for 2023-24 Season

Prediction of League Winner

The Random Forest model was employed to predict the winner of the Premier League for the 2023-2024 season. Based on the features and historical data provided, the model predicted that **Manchester City** would be the league winner. This result aligns with Manchester City's recent history of strong performances in the league, reflecting the model's ability to identify consistent high performers based on the input data.

Predicted Winner using Random Forest: **Manchester City**

Prediction of Total Points

The Random Forest model was used to predict the total points that each team would accumulate by the end of the season. The model provided a ranked list of teams based on their predicted points, with **Manchester City**, **Arsenal**,

and **Liverpool** at the top of the table. The predicted points for these teams were as follows:

Table 5.1: Predicted Points for Top Teams

Team	**Predicted Points**
Manchester City	89.72
Arsenal	89.14
Liverpool	81.68

These predictions suggest that Manchester City and Arsenal are expected to have a closely contested battle for the league title, with Liverpool also finishing strongly. Other notable predictions include:

Table 5.2: Summary of Predicted Points for Top 10 Teams

Team	**Predicted Points**
Manchester City	89.72
Arsenal	89.14
Liverpool	81.68
Tottenham Hotspurs	71.29
Aston Villa	64.72
Chelsea	63.71
Manchester United	59.95
Newcastle United	59.09
West Ham United	53.36
Crystal Palace	48.00

These point predictions provide detailed insights into how each team is expected to perform over the season, with **Manchester City** and **Arsenal** emerging as the frontrunners.

Interpretation of Results

The predictions generated by the models provide a compelling forecast for the 2023-2024 Premier League season. The consensus among the models points to Manchester City, Arsenal, and Liverpool as the dominant teams,

with Manchester City being the likely league winner. The predicted points further highlight the competitive nature of the league, with a close contest expected at the top. These results align with recent trends in the Premier League, where these teams have consistently performed at a high level. The accuracy and agreement among the models suggest that these predictions are well-founded, offering valuable insights for analysts, fans, and stakeholders in the sport.

Strengths and Weaknesses of Each Model Each model exhibited unique strengths and weaknesses. Logistic Regression was simple and interpretable, but it struggled with the imbalanced dataset, leading to poor performance in predicting the minority class, such as the league winner. XGBoost demonstrated strong overall performance with higher accuracy, but it still faced challenges in correctly identifying the minority class. Its ability to handle both classification and regression tasks makes it a versatile choice. Random Forest achieved high accuracy and strong predictive power, particularly in capturing non-linear relationships, showcasing its effectiveness in handling complex interactions between features.

Chapter 6

Conclusion and Future Scope

Conclusion This study's findings underscore the remarkable potential of ensemble methods, particularly Random Forest and XGBoost, in the realm of football league predictions. These models not only excelled in accuracy and reliability but also proved their worth in capturing the complex dynamics of league outcomes. Among them, **Random Forest stood out as the best**, consistently delivering precise predictions for league winners and final points. However, the journey revealed some challenges, particularly in managing class imbalance and forecasting outcomes for teams at the lower end of the table. These hurdles open exciting avenues for future exploration and refinement, offering opportunities to enhance the models' robustness and accuracy further. In the grand scheme, the insights generated by these predictive models are invaluable, providing analysts, teams, and stakeholders with a powerful, data-driven lens through which to view and anticipate league dynamics. As the field of sports analytics continues to evolve, the potential of these models is bound to grow, especially with the integration of richer data sources and more advanced techniques. What we have now is just the beginning—a glimpse into a future where data-driven predictions could redefine how we understand and engage with the beautiful game.

Challenges and Limitations Despite the strong performance of the ensemble models, several challenges and limitations were noted. Firstly, the class imbalance in predicting league winners posed difficulties for all models, particularly Logistic Regression, which struggled to correctly identify the minority class. Even with the application of more advanced models like XGBoost and Random Forest, recall for the league winner category remained lower than desired, suggesting that additional techniques, such as oversampling or the use of alternative evaluation metrics, could be explored to enhance model performance. Moreover, while the models provided accurate predictions for the top teams, their performance may vary for lower-ranked teams, where the data is often more variable and less predictable. The predicted points for these teams, while informative, should be interpreted with caution, especially when considering external factors such as injuries, managerial changes, and mid-season transfers that were not explicitly accounted for in the models.

Future Scope The future scope of this research offers numerous opportunities for enhancing predictive models in sports analytics. One primary recommendation is the integration of more comprehensive and granular data sources. While the current study focused on team-level metrics, incorporating player-level statistics, such as individual performance metrics, injury histories, and fitness levels, could provide a more detailed and accurate prediction model. Additionally, including real-time match data, such as in-game events and player positioning, could enhance the model's ability to predict outcomes by capturing the dynamic nature of football matches. Addressing the challenge of class imbalance, particularly in predicting league winners, is another crucial area for future research. Techniques such as the Synthetic Minority Over-sampling Technique (SMOTE) or the use of customized loss functions that penalize misclassification of minority classes more heavily could be explored. These methods could improve the model's abil-

ity to correctly identify underrepresented outcomes, such as predicting the actual league winner. There is also potential for exploring more advanced machine learning techniques beyond Random Forest and XGBoost, which performed well in this study. Deep learning models, such as neural networks, could be investigated for their ability to capture complex, non-linear relationships within the data. Moreover, hybrid models that combine multiple algorithms, such as ensemble learning techniques with deep learning, could offer improved accuracy and robustness in predictions. The advancement of real-time data processing capabilities presents another exciting opportunity: developing real-time predictive models that provide up-to-the-minute forecasts based on live match data. Such models could be invaluable for in-game decision-making by coaches and analysts, offering dynamic insights that adjust as matches progress. Finally, the methodologies and models developed in this study, while focused on the English Premier League, could be adapted and applied to other football leagues or even different sports. Exploring how these models perform in different contexts could provide valuable insights into their generalizability and potential for broader applications in sports analytics. In conclusion, the recommendations outlined above highlight several promising directions for future research in sports analytics. By expanding the scope of data, employing more sophisticated modeling techniques, and addressing current limitations such as class imbalance, future studies can build on the findings of this research to develop even more accurate and insightful predictive models. These advancements will not only enhance the understanding of football dynamics but also provide valuable tools for teams, analysts, and stakeholders in making data-driven decisions[14].

Appendix

Additional Data and Calculations The following section details the intermediate calculations used in the predictive models:

- **Calculation 1: Expected Goals (xG)**
  - Formula: $xG = \sum_{i=1}^{n} p_i$
  - Where p_i represents the probability of scoring for each shot taken.
  - For Team A, the xG for the 2020 season was calculated as follows: $xG = 0.45 + 0.32 + 0.67 + \cdots = 23.45$.
- **Calculation 2: Points Per Game (PPG)**
  - Formula: $PPG = \frac{\text{Total Points}}{\text{Number of Games}}$
  - For Team B, PPG was calculated as follows: $PPG = \frac{45}{38} = 1.18$.

Calculations for Predictive Models

Here we present additional calculations that were performed as part of the predictive modeling process:

- **Model Accuracy Calculation**
  - The accuracy of the Random Forest model was calculated using the following formula:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \times 100$$

– For the 2023-2024 season, the model made 20 predictions, 18 of which were correct, yielding an accuracy of 90%.

Source Code

```
# Load the datasets
datasets = {}
file_paths = [
    '2016-2017.csv',
    '2017-2018.csv',
    '2018-2019.csv',
    '2019-2020.csv',
    '2020-2021.csv',
    '2021-2022.csv',
    '2022-2023.csv',
    '2023-2024.csv'
]

for file_path in file_paths:
    # Extracting season name from the file path
    season = file_path.split('/')[-1].split('.')[0]
    datasets[season] = pd.read_csv(file_path)

# Checking the columns (features) in each dataset
features = {season: data.columns.tolist() for season,
   data in datasets.items()}
print(features)
```

Listing 1: Loading the datasets

Feature Engineering

```
# Define the function to calculate points based on match
   result
def calculate_points(row):
    if row['FTR'] == 'H':
        return 3, 0
    elif row['FTR'] == 'A':
```

```
6           return 0, 3
7       else:
8           return 1, 1
9
10  # Apply this function to each dataset and create new
        columns
11  for season in datasets:
12      datasets[season][['HomePoints', 'AwayPoints']] =
            datasets[season].apply(calculate_points, axis=1,
            result_type='expand')
```

Listing 2: Feature Engineering

Model Training

```
1
2   # Splitting the dataset into training and testing sets
3
4   # Training on data from 2016-2023
5   train_data = seasons_combined[seasons_combined['Season']
        < 2023]
6
7   # Testing on 2023-2024 season
8   test_data = seasons_combined[seasons_combined['Season']
        == 2023]
9
10  # Define features
11  features = [
12      'Points_Home', 'GoalsScored_Home', '
            GoalsConceded_Home',
13      'Wins_Home', 'Draws_Home', 'Losses_Home',
14      'Points_Away', 'GoalsScored_Away', '
            GoalsConceded_Away',
15      'Wins_Away', 'Draws_Away', 'Losses_Away'
16  ]
17
18  # Define target variables
19  target_winner = 'Champion'
```

```
target_points = 'TotalPoints'

# Extract features and target variables for training and
    testing
X_train = train_data[features]
y_train_winner = train_data[target_winner]
y_train_points = train_data[target_points]

X_test = test_data[features]
y_test_winner = test_data[target_winner]
y_test_points = test_data[target_points]
```

Listing 3: Model Training

Model Implementation

```
# Logistic Regression for League Winner Prediction
from sklearn.linear_model import LogisticRegression

logreg_winner = LogisticRegression(max_iter=1000,
    random_state=42)
logreg_winner.fit(X_train, y_train_winner)

y_pred_winner_logreg = logreg_winner.predict(X_test)
accuracy_winner_logreg = accuracy_score(y_test_winner,
    y_pred_winner_logreg)
print("Logistic Regression Accuracy for League Winner:",
    accuracy_winner_logreg)

from xgboost import XGBClassifier, XGBRegressor
# Train the XGBoost model for League Winner
xgb_winner = XGBClassifier(n_estimators=50, max_depth=3,
    use_label_encoder=False, eval_metric='logloss',
    random_state=42)
xgb_winner.fit(X_train, y_train_winner)

y_pred_winner_xgb = xgb_winner.predict(X_test)
```

```
accuracy_winner_xgb = accuracy_score(y_test_winner,
    y_pred_winner_xgb)
print("XGBoost Accuracy for League Winner:",
    accuracy_winner_xgb)

from sklearn.ensemble import RandomForestClassifier

# Train the Random Forest model for League Winner
rf_winner = RandomForestClassifier(n_estimators=100,
    random_state=42)
rf_winner.fit(X_train, y_train_winner)

y_pred_winner_rf = rf_winner.predict(X_test)
accuracy_winner_rf = accuracy_score(y_test_winner,
    y_pred_winner_rf)
print("Random Forest Accuracy for League Winner:",
    accuracy_winner_rf)
```

Listing 4: Logistic Regression and Random Forest

Points Prediction Model Implementation

```
from xgboost import XGBRegressor
from sklearn.ensemble import RandomForestRegressor

# Define features and target for points prediction
features = [
    'Points_Home', 'GoalsScored_Home', '
        GoalsConceded_Home',
    'Wins_Home', 'Draws_Home', 'Losses_Home',
    'Points_Away', 'GoalsScored_Away', '
        GoalsConceded_Away',
    'Wins_Away', 'Draws_Away', 'Losses_Away'
]

target_points = 'TotalPoints'

# Split the data into training and testing sets
```

```
X_train = train_data[features]
y_train_points = train_data[target_points]
X_test = test_data[features]
y_test_points = test_data[target_points]

# Train the Random Forest classifier for Points prediction
# Initialize the RandomForestRegressor
rf_model_points = RandomForestRegressor(random_state=42)

# Train the model on the training data
rf_model_points.fit(X_train, y_train_points)

# Make predictions on the test data
y_pred_points_rf_df = rf_model_points.predict(X_test)

# Combine predictions with the team names for display
predicted_points_rf_df = pd.DataFrame({
    'Team': test_data['Team'],
    'Predicted Points': y_pred_points_rf
})

# Display the predicted points for each team, sorted by points
predicted_points_rf_df = predicted_points_rf_df.sort_values(by='Predicted Points', ascending=False)
print(predicted_points_rf_df)

# Combine predictions with the team names for display
predicted_points_df = pd.DataFrame({
    'Team': test_data['Team'],
    'Predicted Points': y_pred_points_rf_df
})
```

Listing 5: Random Forest Points Prediction

Bibliography

1. Baboota, R., & Kaur, H. (2019). Predictive analysis and modelling football results using machine learning approach for English Premier League. *International Journal of Forecasting*, 35(1), 28-43.
2. Carnolli, N., Micarelli A., Sansonetti, G., & Andrea De Angellis, H. (2019). Hybrid machine learning approaches for football match predictions. *Journal of Sports Analytics*, 5(4), 273-284.
3. Bunker, R., & Thabtah, F. (2019). A Machine Learning Framework for Sport Result Prediction. *Applied Computing and Informatics*, 15(1), 27-33.
4. Herold, M., Brechot, F., & Memmert, D. (2020). Match outcome prediction in football: Which match characteristics contribute most to success? *International Journal of Sports Science & Coaching*, 15(2), 239-248.
5. Goddard, J. (2005). Regression models for forecasting goals and match results in association football. *International Journal of Forecasting*, 21(2), 331-340.
6. Lewis, M. (2003). Moneyball: The Art of Winning an Unfair Game. W. W. Norton Company.
7. Kelner, M. (2018). How data is helping football teams win games. *BBC Sport*.

8. Hubáček, O., Šourek, G., & Železný, F. (2019). Forecasting outcomes in the Premier League: Can Machine Learning Models Predict Match Results and Final League Positions? *IEEE Access*, 7, 164334-164346.

9. Berrar, D., Lopes, P., & Dubitzky, W. (2019). Incorporating domain knowledge in machine learning for football outcome prediction. *IEEE Transactions on Knowledge and Data Engineering*, 31(4), 663-675.

10. Herbinet C. Predicting Football Using Machine Learning Techniques

11. Soccer match outcome prediction with random forest and gradient boosting models.

12. Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785-794.

13. Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), 5-32.

14. Article on SpringerLink

15. Smith, J. (2020). Machine Learning in Sports: Decoding Complexity. *Journal of Sports Analytics*.

16. Groll, A., Ley, C., Schauberger, G., & Van Eetvelde, H. (2018). Prediction of the FIFA World Cup 2018—A Random Forest Approach with an Emphasis on Estimated Team Ability Parameters. *Journal of Quantitative Analysis in Sports*, 14(1), 29-38.

17. Nguyen, P., & Zhang, T. (2019). Ensemble Methods in Sports Predictions: A Comprehensive Review. *Applied Computing and Informatics*.

18. Wikipedia Premier League
19. Wikipedia Sports Analytics
20. Wikipedia Logistic Regression
21. Wikipedia XGBoost
22. Wikipedia Random Forest
23. Wikipedia Football Prediction
24. J. James Reade, Carl Singleton and Leighton Vaughan Williams
25. Constantinou, A. C., Fenton, N. E., Neil, M. (2012). Pi-football: A Bayesian network model for forecasting Association Football match outcomes