A Comparative Analysis of Classification Algorithms for predicting Football Match Outcomes

Ankita Kumari Donal Loitam
EP21BTECH11008 AI21BTECH11009

Abstract

Football, being one of the most popular sports in the world, has emerged as a prominent domain for sports betting, offering numerous opportunities for predictive analysis and decision-making. Due to its widespread appeal, the perception of football match outcomes holds universal interest for fans, coaches, media, and gamblers, and hence although forecasting football outcomes remains a challenge, the football betting industry is continuously expanding. In this project, we leverage machine learning algorithms, including Logistic Regression, Decision trees, Random Forest AdaBoost, Gaussian Naive Bayes, and XGBoost, and feature engineering techniques such as Feature Creation, Correlation-based Feature Selection, and Recursive Feature Elimination. These techniques utilize multiple statistics from previous matches to predict football match outcomes, with a focus on enhancing model accuracy and exploring the efficacy of various feature engineering methods. Several combinations of prediction models and feature engineering techniques were tested, with the experimental results showing encouraging performance in terms of different performance metrics, where our best model(Random Forest) achieved accuracies close to 81%.

1. Introduction

Football is one of the most popular sports in the world, so the perception of the game and the prediction of results is of general interest to fans, coaches, media and gamblers. Although predicting football results is a very complex task, the football betting business has grown over time. The unpredictability of football results and the growing betting business justify the development of prediction models to support gamblers.

In this project, we develop machine learning methods that take multiple statistics of previous matches and both team's attributes as inputs to predict the outcome of football matches. More specifically, our focus lies in evaluating the effectiveness of different feature engineering techniques and assessing the performance of diverse machine learning algorithms across these techniques employed.Through rigorous experimentation, we have tested several prediction models, yielding promising results. Specifically, our findings indicate a notable improvement in the profitability of football bets, underscoring the potential of our approach.

2. Background: Football Overview

In this section, we will have a quick look at football rules, match events, the domestic league(English Premier league) for which we are doing our analysis etc.

2.1. An overview

Football, known as soccer in some parts of the world, is a widely cherished sport characterized by its simplicity and universal appeal. Played by two teams of 11 players each, the objective of the game is straightforward: to score goals by moving a spherical ball(the football) into the opponent's goal using any part of the body except the hands and arms. The team with the most goals at the end of the regulation time, typically 90 minutes, emerges victorious.

The game's charm lies in its simplicity and the strategies employed by players and teams to outwit their opponents. Teams work together to pass, dribble, and shoot the ball towards the opposing goal while simultaneously defending their own. In the event that both teams score an equal number of goals by the end of regulation time, the match may result in a draw or proceed to extra time, and possibly a penalty shootout, depending on the competition's rules. The anticipation and excitement of scoring goals are central to the essence of football and which keeps fans and players hooked !

However, the **inherent unpredictability of football matches** is what drives bookies and machine learning enthusiasts to predict outcomes eagerly ! With lots of factors influencing each game, the task of forecasting match results presents an intriguing challenge for both bookmakers and those interested in machine learning and hence there are lots of ongoing researches on this topic.

2.2. Match Events and rules

- **Goals:** a goal is scored when the ball enters the opposing team's goal.
- **Shots:** a shot is when a player hits the ball towards the opposing team's goal with the intention of scoring.
- **Passes:** a pass is when a player hits the ball towards another player of his team.
- **Crosses:** a cross is when a player hits the ball from the side of the pitch towards the opposing team's goal with the intention of passing the ball to one of his teammates.
- **Possession:** the possession represents the fraction of the time that a team controls the ball in the match.
- **disciplinary cards:** Yellow card is awarded for small fouls,and means the player is only cautioned and given a warning. Red card is given for serious fouls and means dismissal from the game for that particular player. Two yellow cards in the same match for one player also result in the player's dismissal.
- **Ball In or Out of Play:** The ball is in play when it is inside the field of play and the referee has not stopped play. If the ball rebounds off a goalpost, corner flag, or the referees and remains in the field, it is still in play. The ball is out of play when it has completely crossed the touchlines or the goal lines, whether in air or on ground.
- **Throw-in Rule:** If the ball goes out of play past the touchlines, it results in a throw-in. A throw-in is awarded to the opponents of the player who last touched the ball, deliberately or accidentally. A throw-in is taken by hand.
- **Goal-Kick Rule:** A goal-kick is awarded to the defending team when the ball crosses its goal line, a goal has not been scored, and the last player touch was from the opposition. Although it is generally the goalkeeper, any player may take the goal kick, placing the ball anywhere in the goal area.
- **Corner-Kick Rule:** A corner-kick is awarded to the attacking team when the opposition is last to touch the ball and the ball crosses the goal line without a goal being scored. The attacking team restarts play by placing the ball in one of the 2 corners nearest to where it crossed the goal line.

2.3. Domestics leagues Competition Format

In this project, we are analyzing English Premier League (EPL)matches across 3 seasons(2022-2024). EPL is England's top domestic league and their match statistics are readily available in the internet.

- A domestic league in football refers to a league competition organized within a single country. It involves clubs or teams from various cities or regions within that country.
- There are usually 20 teams in each league, with each team playing the others twice, once at their stadium ('home' match) and once at the opposing team's stadium ('away' match).
- Points are awarded for wins(3 points) and draws(1 point), and the team with the most points at the end of the season is crowned the league champion.

3. Literature Review

The methods used for predicting the result of football matches fall into 3 categories namely statistical models, machine learning algorithms and rating systems have been explored. Our project is based on machine learning algorithms, so we restrict ourselves to the literature review of Machine learning approaches.

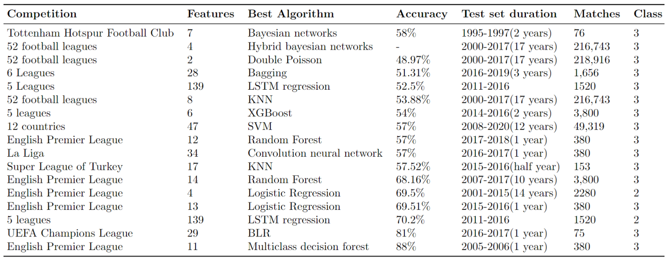
Efforts to enhance football predictability have concentrated on two main areas: employing **advanced machine learning algorithms** and conducting **feature engineering** to create more comprehensive variables. Some studies have pursued both approaches simultaneously. **Figure 1** provides a summary of the primary studies and their attributes in this regard.

3.1. Algorithms-related papers

Researchers have investigated various algorithms for predicting football outcomes, ranging from simpler ones like Linear Regression, K-nearest neighbors, and Decision Trees, to more sophisticated methods such as Random Forests, Support Vector Machines, Artificial Neural Networks, and Boosting. Recently, Deep learning techniques like Convolutional Neural Networks and LSTM have emerged. Studies by Zhang et al. and Malini and Qureshi concluded that LSTM models outperformed traditional machine learning algorithms and artificial neural networks in match prediction.

3.2. Features-related papers

The literature highlights various effective features for predicting football outcomes. These include halftime goal data, first goal team data, individual technical behavior data, ball possession and passing over data, and key player position data, as identified by researchers such as Yekhande et al., Parim et al., Bilek and Ulas, and Joseph et al. Kınalıoğlu and Coskun compared predictive models using different machine learning algorithms and evaluated them with various methods. Beal et al. improved prediction accuracy by incorporating expert and personal opinions, environmental factors, player sentiment, competition, and external factors. However, there's a need for richer predictive features and robust validation in future works, as suggested by researchers. While newer studies tend to reuse existing features, limited proposals have been made for innovating with novel features, primarily focusing on data not available in public datasets such as player transfer data, injuries, expert advice, and psychological data, as noted by researchers. Direct prediction of odds is rare due to bookmaker predictions being embedded within them, requiring a combination of machine learning or statistical methods to use odds for match outcome prediction, as discussed by researchers.



Competition	Features	Best Algorithm	Accuracy	Test set duration	Matches	Class
Tottenham Hotspur Football Club	7	Bayesian networks	58%	1995-1997(2 years)	76	3
52 football leagues	4	Hybrid bayesian networks	-	2000-2017(17 years)	216,743	3
52 football leagues	2	Double Poisson	48.97%	2000-2017(17 years)	218,916	3
6 Leagues	28	Bagging	51.31%	2016-2019(3 years)	1,656	3
5 Leagues	139	LSTM regression	52.5%	2011-2016	1520	3
52 football leagues	8	KNN	53.88%	2000-2017(17 years)	216,743	3
5 leagues	6	XGBoost	54%	2014-2016(2 years)	3,800	3
12 countries	47	SVM	57%	2008-2020(12 years)	49,319	3
English Premier League	12	Random Forest	57%	2017-2018(1 year)	380	3
La Liga	34	Convolution neural network	57%	2016-2017(1 year)	380	3
Super League of Turkey	17	KNN	57.52%	2015-2016(half year)	153	3
English Premier League	14	Random Forest	68.16%	2007-2017(10 years)	3,800	3
English Premier League	4	Logistic Regression	69.5%	2001-2015(14 years)	2280	2
English Premier League	13	Logistic Regression	69.51%	2015-2016(1 year)	380	3
5 leagues	139	LSTM regression	70.2%	2011-2016	1520	2
UEFA Champions League	29	BLR	81%	2016-2017(1 year)	75	3
English Premier League	11	Multiclass decision forest	88%	2005-2006(1 year)	380	3

Figure 1. Prediction accuracy of different studies

As can be seen from the table 1, both 3-class classification (W/L/D) and Binary classification(W/not W) are prevalent in the field of football match prediction.

The accuracy of existing studies, as outlined in Table 1, is primarily influenced by the nature of the features utilized. Features generated in real-time during in-play match outcome predictions tend to result in higher accuracies compared to static pre-match features. Studies focusing on in-play predictions differ from those solely relying on pre-match predictions.

Furthermore, the formulation of the predictive problem significantly impacts accuracy. Models predicting simple win or loss scenarios tend to be more accurate than those predicting draws, as complexity increases with the number of categories being predicted.

Dataset size also plays a role in model capability, although the number of features used in football match prediction has a relatively small effect on prediction accuracy.

4. Methodologies Overview

In this section, we provide a concise overview of the methodologies and algorithms utilized in our study.

4.1. Feature Engineering Techniques

Feature engineering is the process of selecting, transforming, or creating input variables (predictive features) to improve the performance and interpretability of machine learning models. A feature also called a dimension, it is an input variable used to generate model predictions. Because model performance largely rests on the quality of data used during training, feature engineering is a crucial preprocessing technique that requires selecting the most relevant aspects of raw training data for both the predictive task and model type under consideration. Following are some of the feature engineering techniques used in this project.

4.1.1 Feature Creation

The process of generating new features based on domain knowledge or by observing patterns in the data. Having and engineering good features helps to accurately represent the underlying structure of the data and therefore create the best model. In our project, based on domain knowledge, we have created new features out of existing features. For example: **Average number of home team wins in their last 5 HOME matches**(which may carry information about the current form of the team) **Average number of away team's AWAY match wins & Goal difference = ’gd’** based off goals for(’gf’) and goals against(’ga’).

$$gd = gf - ga \tag{1}$$

4.1.2 Correlation Based feature Selection

We understand that more features mean more dimensions, training with a big number of features will result in high bias and will lead to overfitting. Therefore having a high dimension is a curse. The **”curse of dimensionality”** refers to the challenges and limitations that arise when dealing with high-dimensional data. As the number of features or dimensions in a dataset increases, the volume of the feature space grows exponentially.

Therefore, we need methods to reduce dimensions without discarding all features. This is where feature selection and feature extraction techniques become essential !

- **Correlation:** Correlation is a statistical measure used to describe the relationship between two or more variables. It indicates the strength and direction of the connection between variables. A correlation coefficient ranges from -1 to 1, where -1 indicates a perfect negative correlation, 1 indicates a perfect positive correlation, and 0 indicates no correlation. In simple terms, it tells us how two variables move together: whether they increase or decrease together (positive correlation), move in opposite directions (negative correlation), or have no apparent relationship (zero correlation).

$$\rho_{X,Y} = \frac{\sum(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum(X_i - \bar{X})^2 \sum(Y_i - \bar{Y})^2}} \tag{2}$$

  where:
  - $\rho_{X,Y}$ is the correlation coefficient between variables X and Y,
  - X_i and Y_i are individual data points,
  - $\bar{X}$ and $\bar{Y}$ are the means of variables X and Y, respectively.
- **Collinearity and Multicollinearity:** Collinearity refers to a situation where two independent variables exhibit a strong linear relationship with each other. In other words, changes in one variable tend to be associated with

changes in the other variable. Multicollinearity is a more specific case of collinearity, involving multiple predictors that are correlated with each other. This often occurs when two or more predictors are highly correlated, with high correlation coefficients, what value is high depends on the context !

Note: Correlation between predictor and target variable is a good thing for a model. However, correlation among the predictors causes multiple problems.

4.1.3 Consequences of multicollinearity

Multicollinearity could result in significant problems during model fitting. It can reduce the overall performance of regression and classification models:

- does not increase bias, but it can increase variance (overfitting)
- make the estimates very sensitive to minor changes in the model i.e a predictor's estimate on the response variable will be less exact and less dependable.
- doesn't affect the predictive power, but individual predictor variable's impact on the response variable could be calculated wrongly.

The result is that the coefficient estimates are unstable and difficult to interpret. Multicollinearity saps the statistical power of the analysis, can cause the coefficients to switch signs, and makes it more difficult to specify the correct model.

4.1.4 Remedies to multicollinearity problem

- **Increase the sample size:** Enlarging the sample will introduce more variation in the data series, which reduces the effect of sampling error and helps increase precision when estimating various properties of the data. Increased sample sizes can reduce either the presence or the impact of multicollinearity, or both.
- **Manual Method :Variance Inflation Factor (VIF):**As the name suggests, a variance inflation factor (VIF) quantifies how much the variance is inflated. But what variance? Recall that we learned previously that the standard errors — and hence the variances — of the estimated coefficients are inflated when multicollinearity exists. A variance inflation factor exists for each of the predictors in a multiple regression model. For example, the variance inflation factor for the estimated regression coefficient b_j —denoted VIF_j —is just the factor by which the variance of b_j is "inflated" by the existence of correlation among the predictor variables in the model. In particular, the variance inflation factor for the j'th predictor is:

$$VIF_j = \frac{1}{1 - R_j^2} \tag{3}$$

- **Automatic Method: Recursive Feature Elimination (RFE):** RFE is popular because it is easy to configure and use and because it is effective at selecting those features (columns) in a training dataset that are more or most relevant in predicting the target variable.

  There are two important configuration options when using RFE: the choice in the number of features to select and the choice of the algorithm used to help choose features. Both of these hyperparameters can be explored, although the performance of the method is not strongly dependent on these hyperparameters being configured well.

- **Feature Elmination using PCA Decomposition:** Principal Component Analysis (PCA) is a dimensionality reduction technique commonly used for feature elimination. It works by transforming the original features into a new set of orthogonal features called principal components. The principal components are ordered in terms of the amount of variance they explain in the data, allowing us to select a subset of the components that capture most of the variability in the data while discarding the rest.

  Mathematically, PCA involves computing the eigenvectors and eigenvalues of the covariance matrix of the original data. The eigenvectors represent the directions of maximum variance in the data, while the eigenvalues indicate the magnitude of variance along these directions. By selecting the top k eigenvectors corresponding to the largest eigenvalues, we can reduce the dimensionality of the data to k dimensions, effectively eliminating less informative features.

4.2. Machine Learning Algorithms

Machine Learning algorithms are transforming football analysis, offering accurate match outcome predictions, insights into player performance, injury prevention, opponent analysis, and scouting. By leveraging historical data and player statistics, these algorithms enable informed decision-making for coaches and clubs, while also enhancing fan engagement through personalized content recommendations. Overall, ML is revolutionizing football analysis, driving innovation and competitiveness in the sport.

4.2.1 Logistic Regression

Logistic regression predicts the probability of a binary outcome based on linear regression. It models the relationship between the independent variables X and the binary dependent variable y using the logistic function

$$\sigma(z) = \tfrac{1}{1+e^{-z}},$$

where z is the linear combination of the independent variables and their coefficients:

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n$$

The logistic function converts the linear combination of predictors into probabilities, which can be interpreted as the likelihood of belonging to a particular class. Logistic regression is particularly effective when there is a strong linear relationship between features and labels. However, it may not perform well in cases with complex interactions or non-linear relationships.

Despite this limitation, logistic regression has several advantages. It is noise-resistant and performs well on small datasets. Additionally, it provides interpretable coefficients, making it useful for understanding the relationship between predictors and the outcome variable.

4.2.2 Decision trees

A decision tree is a decision support hierarchical model that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utilityThe advantage is that it does not require any domain knowledge or parameter assumptions and is suitable for high-dimensional data. It is robustness to Outliers, outliers can not significantly impact the performance of the model since the splitting criterion is based on ranks rather than absolute values.

4.2.3 Random forest

This method involves an ensemble approach that combines multiple Decision Trees. The Random Forest technique merges these Decision Trees, constructing each one independently without backtracking. Each tree in the ensemble utilizes independent sampling and exhibits the same level of randomness in data selection as the others. During the classification process, each tree contributes to a collective decision by voting for the most frequent category, thus assessing the importance of variables in classification. Random Forests typically outperform individual Decision Trees in mitigating overfitting.

The correlation between the features of the model and the predictions on the test data in the Random Forest approach tends to be weak. While this might seem disadvantageous for poorly predicted football match outcomes, it can actually be beneficial in this context. Employing a training dataset with numerous features may lead to a model that excels in predicting some features while performing inadequately on others. Moreover, since each Decision Tree in the Random Forest model has distinct classification criteria, it helps diminish the correlation between them, further enhancing model robustness.

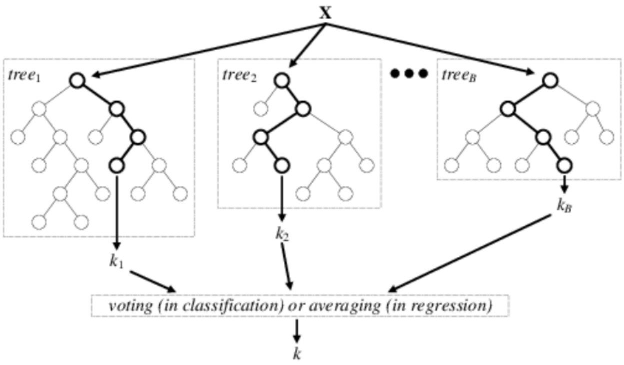


Figure 2. Random forest method

4.2.4 Gradient boosting

Gradient Boosting is a powerful ensemble learning technique that builds a predictive model in a stage-wise fashion by combining the predictions of multiple weak learners. Unlike traditional boosting methods that focus on reducing errors directly, Gradient Boosting optimizes a differentiable loss function, typically using gradient descent, to minimize the errors of the model. This iterative process sequentially adds weak learners to the ensemble, each one focusing on the mistakes made by its predecessors. Gradient Boosting is highly flexible and can be used for both regression and classification tasks, producing accurate predictions even with complex datasets. However, its main drawbacks include potential overfitting, longer training times, and sensitivity to hyperparameters. Despite these challenges, Gradient Boosting remains one of the most popular and effective techniques in machine learning, particularly when coupled with advanced implementations like XGBoost or

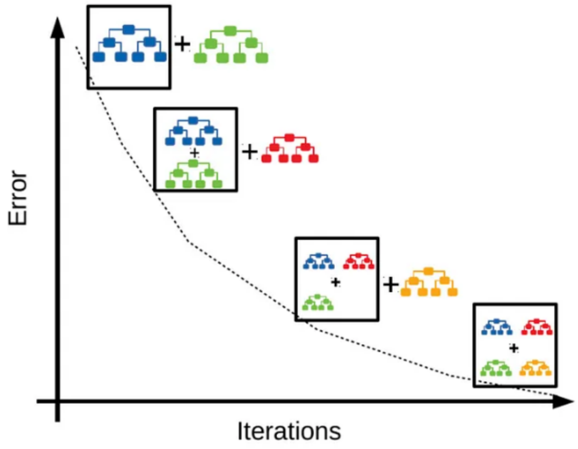


Figure 3. Gradient Boosting: **Iterative addition of weak learners(decision trees)**

LightGBM, offering superior performance and scalability.

4.2.5 Gaussian Naive Bayes

Gaussian Naive Bayes (GNB) is a simple yet effective classification algorithm, particularly suitable for datasets with continuous features. It operates based on Bayes' theorem and assumes that each feature follows a Gaussian (normal) distribution and that features are independent of each other. Mathematically, the probability density function (PDF) of the Gaussian Naive Bayes classifier can be represented as:

$$p(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}\right)$$

where x_i represents a feature value, μ_y is the mean of the feature x_i for class y, σ_y^2 is the variance of feature x_i for class y, and $p(x_i|y)$ is the probability of observing feature x_i given class y.

Using the independence assumption, this leads to the following:

$$P(y|x_1,...,x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{\sum_{y'\in Y} P(y')\prod_{i=1}^{n} P(x_i|y')}$$

where, y is the target class and x_i are the predictor (input) variables

This assumption of feature independence makes GNB computationally efficient and robust to irrelevant features, often resulting in fast training and prediction times. However, GNB's simplicity may limit its ability to capture complex relationships between features, and it is sensitive to the Gaussian distribution assumption. Despite these limitations, GNB remains a valuable tool in machine learning, particularly in scenarios where computational resources are limited and interpretability is crucial.

4.2.6 AdaBoost

AdaBoost, or Adaptive Boosting, is an ensemble learning method that iteratively combines weak learners, typically decision trees, to create a strong learner. It starts by assigning equal weights to all data points and trains a weak learner. It then adjusts the weights of incorrectly classified instances, emphasizing their importance in subsequent iterations. Each new weak learner focuses on the misclassified instances, gradually improving the overall model. However, AdaBoost can be sensitive to noisy data and outliers, and its sequential nature might limit its scalability.

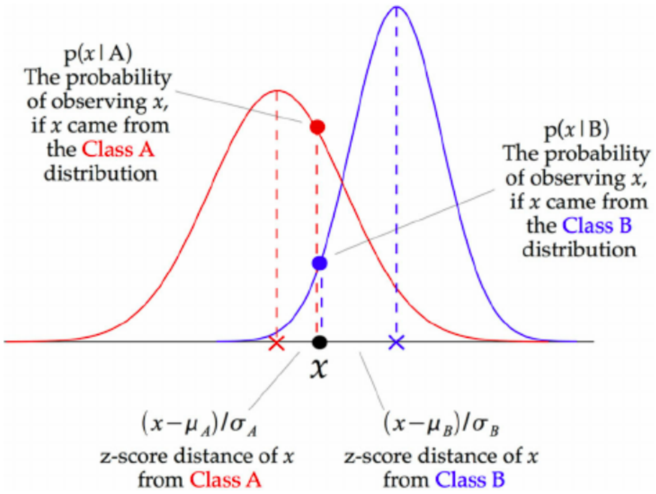

p(x | A)
The probability of observing x, if x came from the Class A distribution
p(x | B)
The probability of observing x, if x came from the Class B distribution
x
$(x-\mu_A)/\sigma_A$
z-score distance of x from Class A
$(x-\mu_B)/\sigma_B$
z-score distance of x from Class B

Figure 4. Gaussian Naive Bayes method

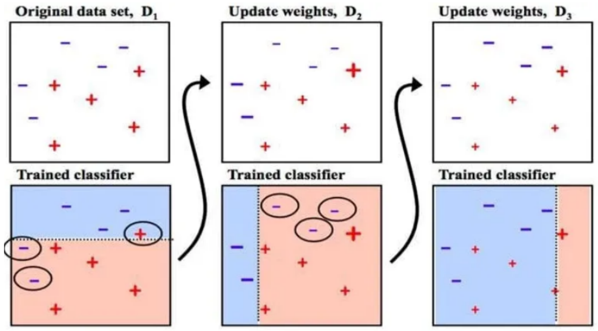


Figure 5. AdaBoost: Iterative Adjusting of weights

4.2.7 XGBoost

XGBoost extends and improves upon the Gradient Boosting. The XGBoost algorithm is faster. It takes advantage of the multi-threading of the CPU based on traditional Boosting and introduces regularisation to control the complexity of the model. Prior to the iteration, the features are pre-ranked for the nodes and by traversing them the best segmentation points are selected, which results in lower complexity of data segmentation. The XGBoost method is popular amongst machine learning researchers. However, the method does not perform well in capturing high-dimensional data, such as images, audio, text, etc. For predicting football matches with based on lower-dimensional data, XGBoost may produce promising models.

5. Dataset

5.1. Data Collection using Webscraping

Web scraping is an automated method used to extract large amounts of data from websites.This data is often unstructured, presented in HTML format, and needs to be converted into structured data, such as a spreadsheet or a database, for use in various applications.

To obtain recent and relevant data for football matches, we utilized web scraping techniques. Specifically, we employed the Python library called **BeautifulSoup** for this purpose. BeautifulSoup is a powerful tool for parsing HTML and extracting data from web pages.

5.1.1 Advantages of BeautifulSoup

- **Performance**: BeautifulSoup is generally faster and less resource-intensive compared to other scraping tools like Selenium. It parses static HTML content directly, resulting in quicker and more efficient data extraction.
- **Ease of Use**: BeautifulSoup provides a simple and intuitive interface for navigating and manipulating HTML documents, making it ideal for web scraping tasks.

5.1.2 Data Source: FBREF

For this project, we scraped data from **FBREF**, a website known for providing a wealth of open-source football data that can be analyzed to unlock valuable insights. FBREF offers the highest level of coverage from partner Opta for over 20 competitions, including major European leagues, Champions League, World Cup and top leagues in various countries.

The Reasons for Choosing FBREF are :-

- FBREF provides numerous pages containing various types of football data, making it an ideal source for scraping a wide range of features.
- Feature engineering is a crucial task in this project due to the large number of features available, which need to be carefully processed to extract meaningful insights.

5.2. Data features

Given in table 1 are the features which were scraped from the Fbref website :-

The features gathered from scraping the Fbref website offer valuable insights for our machine learning classification models. It's essential to acknowledge that certain features, such as **"gf" (goals for) and "ga" (goals against)** and many more, are only accessible for past matches. Consequently, these features are not available for the current match when conducting predictions.Therefore, **careful data pre-processing** becomes essential to ensure the correctness and relevance of our predictions.

5.3. Data Pre-processing

In this section, we conduct various data cleaning, transformation, and feature engineering tasks to prepare the dataset for subsequent analysis and modelling.

Table 1. Description of Football Match Features

Feature	**Description**
date	Date when the match took place.
time	Time of day when the match started.
comp	Competition or league of the match.
round	Round or stage of the competition.
day	Day of the week when the match occurred.
venue	Stadium or location of the match.
result	Outcome of the match (win, loss, draw).
gf	Number of goals scored by the team.
ga	Number of goals conceded by the team.
opponent	Opposing team in the match.
xg	Expected goals created by the team.
xga	Expected goals conceded by the team.
poss	Percentage of possession by the team.
captain	Designated captain for the team.
formation	Tactical formation used by the team.
sh	Total shots taken by the team.
sot	Shots on target by the team.
sot%	Percentage of shots on target.
g/sh	Goals per shot ratio.
g/sot	Goals per shot on target ratio.
dist	Average distance of shots taken by the team.
fk	Number of free kicks taken by the team.
pk	Number of penalties scored by the team.
pkatt	Number of penalties attempted by the team.
cmp	Number of completed passes by the team.
att	Number of attempted passes by the team.
cmp%	Completion percentage of passes.
totdist	Total distance covered by the team.
prgdist	Progressive distance towards the opponent's goal.
kp	Key passes leading to shots by teammates.
1/3	Passes completed in the final third of the pitch.
ppa	Passes per defensive action by the team.
crspa	Crosses completed into the penalty area.
prgp	Progressive passes moving the ball forward.
sca	Shot-creating actions leading to a shot.
gca	Goal-creating actions leading to a goal.
crs	Number of crosses attempted by the team.
int	Number of interceptions made by the team.
tklw	Number of tackles won by the team.
recov	Number of recoveries made by the team.
season	Football season to which the match belongs.
team name	Name of the football team.

5.3.1 Data Cleaning and Transformation

- **Viewing the DataFrame Head and Dropping Insignificant Columns:**
  To initiate the data preprocessing phase, we begin by examining the head of the DataFrame to gain a preliminary understanding of its structure and contents. Subsequently, we identify and drop any columns deemed insignificant for our analysis. This step ensures that our dataset contains only relevant information essential for our predictive modeling tasks.
- **Renaming Entries in the ”Team Name” Column**
  Standardizing the entries in the **”Team Name”** column is crucial for consistency and comparability across different analyses. To achieve this, we review and rename certain entries to align with corresponding entries in the **”Opponent”** column. This harmonization facilitates seamless integration of team-related data and simplifies subsequent data manipulation steps.
- **Handling NaN Values and Checking Column Data Types**
  NaN (Not a Number) values and inconsistent data types can significantly impede the integrity and accuracy of our analyses. Therefore, we meticulously inspect our dataset for any missing values and assess the data types of each column. Fortunately for us, there were missing data in only one of the columns namely ”g/sot”, which was subsequently removed after careful feature engineering.
- **Encoding Categorical Variables**
  Categorical variables, such as the ”Venue” feature, ”Opponent” feature require transformation into numerical format to facilitate their incorporation into machine learning models. Initially, we employ label encoding to convert categorical values into numeric representations. Additionally, we explore the potential benefits of utilizing one-hot encoding to further enhance the interpretability and performance of our predictive models.
- **Creating the Target Feature**
  In preparation for our binary classification task, we derive the target feature based on the outcomes of matches. By categorizing match results as either wins (1) or draws/losses (0), we transform our prediction problem into a binary classification scenario.

.

5.3.2 Home Team Results Visualization

- **General Home Team Performance**
  Visualizing the distribution of wins and losses for home teams provides valuable insights into overall home team performance trends. The concept of **”home advantage”** underscores the phenomenon where teams tend to perform better when playing in familiar surroundings. This inherent bias towards home conditions has significant implications in sports analysis, particularly in football. It is, therefore, essential to consider the**”venue”** as a feature in our predictive models.

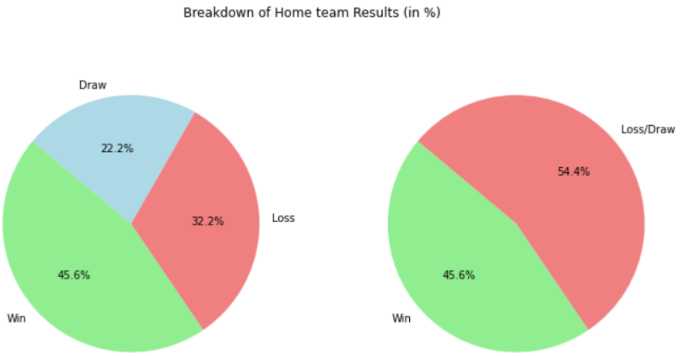


Figure 6. Visualization of Home Team Performance

- **Team-Specific Analysis**
  - **Home Win Percentage vs. Away Win Percentage**
    A comparative analysis of home and away win percentages across different teams reveals nuanced performance variations. By visualizing these disparities, we identify teams that exhibit distinct home or away dominance, informing the creation of predictive features that capture such behavioral patterns.

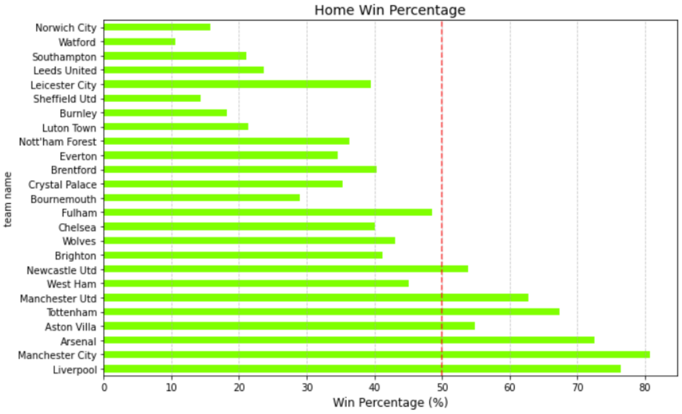


Figure 7. Team-wise Visualization of Home Wins

  - **Home Goals Scored/Conceded vs. Away Goals Scored/Conceded**
    Exploring differences in goals scored and conceded by home and away teams unveils strategic insights into offensive and defensive capabilities. By dissecting goal-related metrics, we uncover performance differentials that inform feature engineering efforts aimed at capturing team-specific scoring and defensive trends.

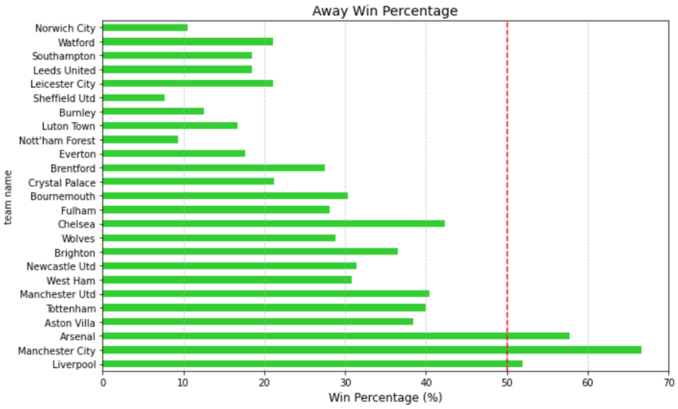


Figure 8. Team-wise Visualization of Away Wins

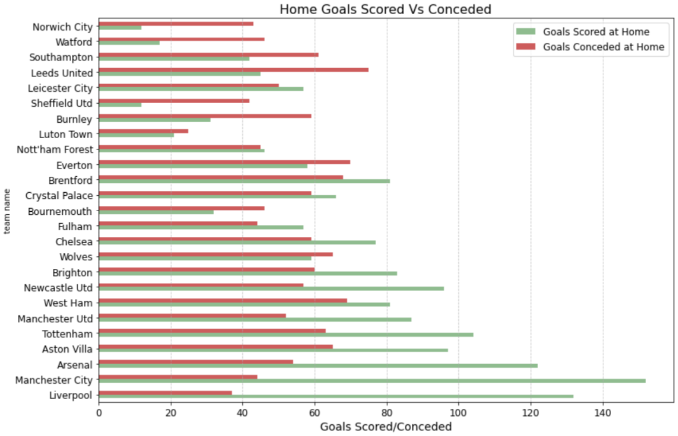


Figure 9. Team-wise Home Goals scored/conceded

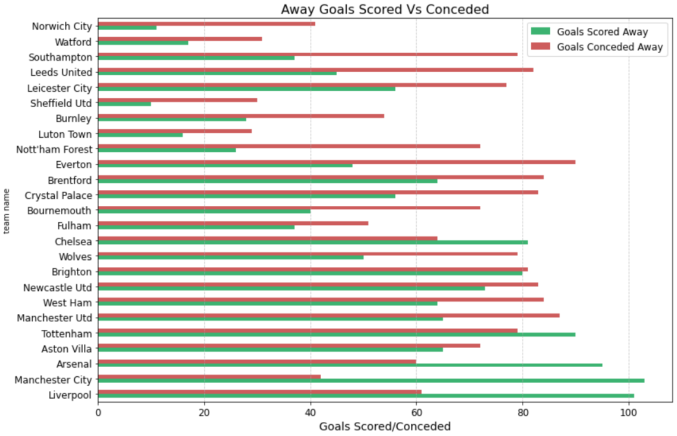


Figure 10. Team-wise Away Goals scored/conceded

5.3.3 Point-Biserial Correlation Analysis

The point-biserial correlation analysis serves as a diagnostic tool to assess the relationship between continuous variables and the binary target feature. By quantifying the strength and direction of associations, we identify potential predictors of match outcomes and guide subsequent feature selection strategies.

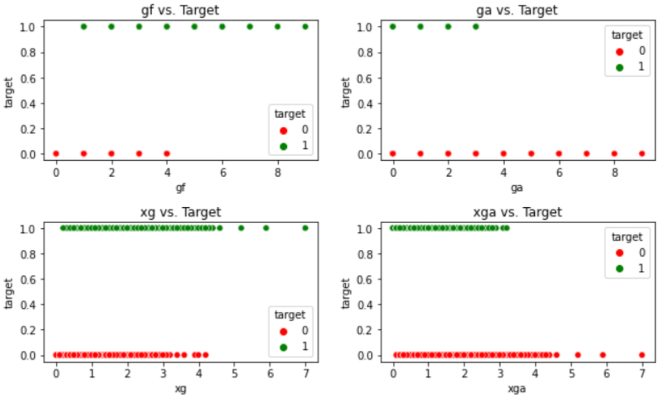


Figure 11. Point Biserial plot between features

- **Feature Selection**
  Leveraging correlation-based feature selection techniques, we identify variables with high predictive utility while mitigating multicollinearity risks. By prioritizing features with significant correlations with the target variable and minimal intercorrelation, we enhance predictive accuracy.
- **Multicollinearity Assessment**
  Detecting and addressing multicollinearity among predictor variables is paramount to ensuring model robustness and interpretability. Through methodologies such as Variance Inflation Factor (VIF) analysis and correlation matrix examination, we pinpoint redundant features and implement corrective measures to optimize model performance.

5.4. Rolling Averages and Recursive Feature Elimination

In football match prediction, utilizing rolling averages of previous match statistics allows us to capture temporal trends and exploit historical performance patterns.By calculating rolling averages of metrics like goals scored, conceded, shots on target, and possession etc. percentage over a specific number of past matches, we gain valuable insights into team dynamics and performance trajectories. For instance, observing an upward trend in goal-scoring over a team's last ten matches may indicate an increase in offensive potency or tactical adjustments enhancing attacking prowess.

Recursive Feature Elimination (RFE) techniques play a crucial role in refining our feature set and enhancing predictive power. RFE systematically evaluates each

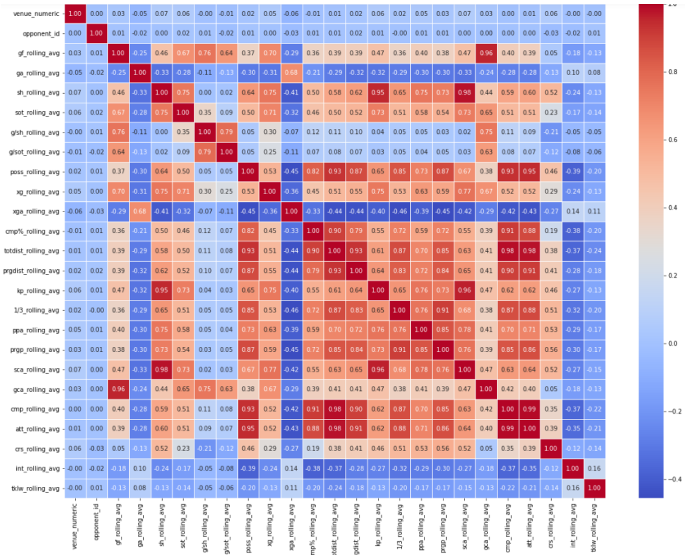


Figure 12. **Correlation matrix of predictor**

feature's importance by iteratively training models with subsets of features and selecting the most informative ones based on their contribution to predictive performance. In football match prediction, RFE helps identify salient predictors influencing match outcomes while reducing the impact of irrelevant or redundant variables.

Integrating rolling averages of historical match statistics and leveraging RFE to select pertinent features improves predictive accuracy and generalization capabilities. These techniques enable capturing nuanced temporal dynamics, exploiting predictive signals inherent in historical data, and generating actionable insights for informed decision-making in football match prediction scenarios.

6. Design and Implementation

6.1. Model components

In this section, we will outline the key components of our predictive model. These components include:

6.2. Model choices

Classification algorithms can be categorized based on their sensitivity to hyperparameters, which are parameters external to the model that affect its behavior and performance. This categorization helps in understanding the nuances of algorithm selection and tuning for predictive modeling tasks.

6.2.1 Algorithms Less Sensitive to Hyperparameters

Algorithms falling into this category include KNN (K-Nearest Neighbors), Decision Trees, AdaBoost, and Gaussian Naive Bayes. These algorithms exhibit a relatively lower sensitivity to hyperparameters compared to others. They typically have a limited number of hyperparameters that significantly impact their performance. For instance, in KNN, the choice of the number of neighbors (K) is a crucial hyperparameter affecting the model's performance. Similarly, decision trees' hyperparameters such as maximum depth and minimum samples split can influence the tree's complexity and generalization ability.

AdaBoost, a popular ensemble learning method, relies on a series of weak learners (e.g., decision trees) and assigns higher weights to misclassified instances, thus focusing on areas of the data that are difficult to classify. Its hyperparameters, such as the number of estimators (the number of weak learners), can impact the overall performance and convergence rate.

Gaussian Naive Bayes is a probabilistic classifier based on Bayes' theorem with the assumption of independence among predictors. While it has fewer hyperparameters compared to other algorithms, the smoothing parameter for handling zero probabilities and the prior probabilities can still influence its classification performance.

6.2.2 Algorithms Very Sensitive to Hyperparameters

In contrast, algorithms such as SVM (Support Vector Machines), Gradient Boosting, and Random Forest belong to this category. These algorithms exhibit a heightened sensitivity to hyperparameters, where even minor adjustments can lead to notable differences in performance outcomes.

Support Vector Machines (SVM) aim to find the optimal hyperplane that separates data points into different classes. Its hyperparameters, such as the choice of kernel function, regularization parameter (C), and kernel coefficient (gamma), profoundly affect its ability to generalize and classify unseen data accurately.

Gradient Boosting is an ensemble technique that builds a series of weak learners (often decision trees) sequentially, with each subsequent learner focusing on the mistakes made by the previous ones. Its performance is highly dependent on hyperparameters such as learning rate, tree depth, and the number of estimators. Subtle variations in these hyperparameters can significantly impact the model's predictive power and generalization ability.

Random Forest, another ensemble method, constructs multiple decision trees during training and outputs the mode of the classes (classification) or the mean prediction (regression) of the individual trees. Hyperparameters like the number of trees, tree depth, and the minimum number of samples required to split a node influence the forest's overall performance and computational efficiency.

Understanding the sensitivity of classification algorithms to hyperparameters is crucial for effectively tuning and optimizing their performance for specific datasets and prediction tasks. By selecting appropriate algorithms and fine-tuning their hyperparameters, practitioners can build robust and accurate predictive models tailored to their application domains.

6.3. General Pipeline

In any football match prediction using machine learning algorithms, we follow a general pipeline consisting breifly of the following steps (**all of which have been discussed thoroughly in different subsections**) :-

- Data Collection: Gather relevant data sources such as match statistics, player attributes, and team performance metrics.
- Data Cleaning and Preprocessing: Handle missing values, normalize or scale features, and encode categorical variables.
- Feature Engineering: Apply feature creation, correlation-based feature selection, and recursive feature elimination techniques to enhance the predictive power of the model.
- Model Selection: Utilize various machine learning algorithms including Logistic Regression, Decision Trees, Random Forests, Gradient Boosting, Gaussian Naive Bayes, AdaBoost, and XGBoost.
- Hyperparameter Tuning: Fine-tune model parameters using techniques like RandomizedSearchCV to optimize model performance.
- Evaluation: Assess model performance using appropriate metrics such as accuracy, precision, recall, and F1-score.

We aim to demonstrate the efficacy of different models in the following scenarios:

1. No Feature engineering applied
2. Feature Creation
3. Correlation-based Feature Selection(no feature creation)
4. Correlation-based Feature Selection + Feature creation
5. Feature creation + Correlation-based selection + Recursive Feature Elimination
6. Feature creation + Recursive Feature Elimination

For each scenario, we evaluate the performance of the following classification models:

- Logistic Regression
- Decision Trees
- Random Forests
- Gradient Boosting
- Gaussian Naive Bayes
- AdaBoost
- XGBoost

We analyze the impact of feature engineering techniques on predictive accuracy and model interpretability across these scenarios.

For complete implementation, please refer https://github.com/Donal-08/Data-Science-Final-Project.

7. Experimentation & Optimisation

7.1. Testing

In the context of our football match prediction project, the testing phase is crucial for evaluating the performance of our machine learning models on unseen data. Our testing approach revolves around assessing the performance of our machine learning models in predicting football match outcomes. Here's a breakdown of our method:

- **Test Data**: As mentioned, the test dataset comprises match data from the second half of the third season onwards until the most recent available matches. This division allows us to simulate real-world scenarios where we predict outcomes for matches that occur after the training period. Our test data consists of 160 observations, ensuring a representative sample for evaluation.
- **Model Evaluation**: After training each model on the training data, we subject our models to rigorous evaluation using a dedicated test dataset. This evaluation entails employing the trained models to make predictions on the test data and comparing these predictions against the actual match outcomes.
- **Performance Assessment**: To quantify the effectiveness of our models, we employ various performance metrics. These metrics, including accuracy, precision, recall, F1 score, and the confusion matrix, provide valuable insights into the models' predictive capabilities and identify areas of strength and improvement.
- **Analysis and Refinement**: Following the computation of performance metrics, we conduct a comprehensive analysis to interpret the results. This analysis informs us about the efficacy of each model and guides our efforts to refine the predictive accuracy. We may iterate on feature selection, adjust model training techniques, or explore alternative algorithms to enhance performance.

7.2. Hyper-parameter tuning

When you're training machine learning models, each dataset and model needs a different set of hyperparameters, which are a kind of tunable parameter. The only way to determine these is through multiple experiments, where you pick a set of hyperparameters and run them through your model. This is called hyperparameter tuning. In essence, you're training your model sequentially with different sets of hyperparameters. **RandomizedSearchCV** method from the scikit-learn library was used to select the most effective parameter values.

Hyperparameter tuning is an important and computationally intensive process. Hyperparameter tuning is a vital step in maximizing model performance, as there are no fixed rules for selecting optimal hyperparameters. Experimentation is key to identifying the best configuration for achieving desired outcomes.

Whatever method you use, you'll have to apply some form of statistical analysis, such as the loss function, to determine which set of hyperparameters gives the best result.The criterion for finding the best-fit hyperparameters is set to be **ROC-AUC** (Receiver Operating Characteristic - Area Under the Curve). This choice is motivated by:

- **Robustness to Class Imbalance:** In the context of our analysis i.e binary prediction of football match outcomes, the class distribution is often imbalanced, with wins (40%) being relatively less frequent compared to non-wins(60%) (draws or losses). Using accuracy as the metric may lead to misleading interpretations, especially if the majority class (non-wins) dominates.ROC AUC provides a more balanced assessment by considering the trade-off between true positive rate (proportion of correctly predicted wins) and false positive rate (proportion of incorrectly predicted non-wins).

By adopting ROC AUC as the evaluation metric and categorizing algorithms based on their sensitivity to hyperparameters, this approach ensures a systematic and robust comparison of classification algorithms for predicting football match outcomes, taking into account the specific characteristics of the task, such as class imbalance.

7.3. Testing metrics

In the context of binary classification tasks, the terms TP, TN, FP, and FN are commonly used to evaluate the performance of predictive models. These terms represent different outcomes of predictions compared to the ground truth.

- **TP (True Positive)**: Instances that are correctly predicted as positive by the model.
- **TN (True Negative)**: Instances that are correctly predicted as negative by the model.
- **FP (False Positive)**: Instances that are incorrectly predicted as positive by the model. It occurs when the model predicts positive, but the ground truth is negative.
- **FN (False Negative)**: Instances that are incorrectly predicted as negative by the model. FN occurs when the model predicts negative, but the ground truth is positive.

These terms are essential for constructing the confusion matrix and computing various performance metrics such as accuracy, precision, recall, and F1-score, which provide insights into the model's classification performance.

7.3.1 Accuracy

Accuracy measures the proportion of correctly classified instances out of the total instances.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Accuracy is a common evaluation metric for classification problems. While it is a commonly used metric, it may not be suitable for imbalanced datasets, where the classes are not equally represented.

7.3.2 Precision

Precision measures the proportion of true positive predictions among all positive predictions made by the model.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Precision is useful when the cost of false positives is high.

7.3.3 Recall (Sensitivity)

Recall measures the proportion of true positive predictions among all actual positive instances in the data.

$$\text{Recall} = \frac{TP}{TP + FN}$$

Recall is important when the cost of false negatives is high. When we have a class imbalance, accuracy can become an unreliable metric for measuring our performance. For this, we use precision and recall instead of accuracy.

7.3.4 F1-score

The F1-score is the harmonic mean of precision and recall, providing a balance between the two metrics.

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Why harmonic mean? Since the harmonic mean of a list of numbers skews strongly toward the least elements of the list, it tends (compared to the arithmetic mean) to mitigate the impact of large outliers and aggravate the impact of small ones. F1-score is useful when you want to seek a balance between precision and recall.

7.3.5 Confusion Matrix

The confusion matrix is a tabular representation of the model's predictions compared to the actual class labels. It consists of four components: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

Table 2. Confusion Matrix

Actual/Predicted	Positive	Negative
Positive	TP	FN
Negative	FP	TN

7.3.6 ROC Curve (Receiver Operating Characteristic Curve)

ROC curve is a graphical representation of the true positive rate (sensitivity) against the false positive rate (1 - specificity) for different threshold values. It helps to visualize the trade-off between sensitivity and specificity. AUC (Area Under the Curve) of the ROC curve summarizes the performance of the model across various thresholds.

7.3.7 ROC AUC (Receiver Operating Characteristic - Area Under the Curve)

ROC AUC provides a single scalar value representing the area under the ROC curve. A higher ROC AUC value indicates better discrimination between positive and negative classes.

These metrics provide different perspectives on the performance of classification algorithms. The choice of metrics depends on the specific requirements of the problem and the relative importance of correctly predicting positive and negative instances. In the context of predicting football match outcomes, where class imbalance is prevalent, metrics like precision, recall, F1-score, and ROC AUC are particularly useful for evaluating model performance.

8. Evaluation

8.1. Absolute Results

In this subsection, we assess the efficacy of different classification algorithms across various feature engineering techniques. Our objectives are two-fold:

1. **Comparison of Classification Algorithms:** We compare the performance of different classification algorithms across the six cases. By examining metrics such as accuracy, precision, recall, and F1-score, we aim to identify the algorithm that consistently performs well for this classification task and data characteristics, irrespective of the feature engineering technique used.
2. **Evaluation of Feature Engineering Techniques:** Additionally, we explore the efficacy of different feature engineering techniques by analyzing the performance metrics obtained across the six cases. By comparing the performance with and without feature engineering and assessing the impact on model accuracy and interpretability, we gain insights into the effectiveness of each technique in enhancing model performance.

To facilitate comparison and analysis, we will look at the average performance metrics obtained across the six cases for each classification algorithm. This allows us to both identify the most suitable algorithm for the classification task and assess the relative importance of different feature engineering techniques too !

Table 3. CASE 1: No feature engineering applied.

	Acc.	Prec.	Rec.	F1 Score
Logistic Regression	0.68	0.68	0.40	0.5
Decision Tree	0.57	0.46	0.5	0.48
Gradient Boosting	0.68	0.66	0.40	0.49
Gaussian Naive Bayes	0.66	0.57	0.60	0.58
AdaBoost	0.65	0.59	0.41	0.49
XGBoost	0.69	0.68	0.43	0.53
Random Forest	0.69	0.67	0.43	0.52

Table 4. CASE 2: Feature creation (to incorporate recent team performance).

	Acc.	Prec.	Rec.	F1 Score
Logistic Regression	0.71	0.62	0.65	0.63
Decision Tree	0.61	0.51	0.54	0.53
Gradient Boosting	0.76	0.73	0.63	0.68
Gaussian Naive Bayes	0.68	0.59	0.62	0.60
AdaBoost	0.71	0.61	0.68	0.64
XGBoost	0.75	0.71	0.62	0.66
Random Forest	0.75	0.75	0.57	0.65

Table 5. CASE 3: Correlation-based Feature Selection (without feature creation).

	Acc.	Prec.	Rec.	F1 Score
Logistic Regression	0.67	0.60	0.46	0.52
Decision Tree	0.56	0.44	0.44	0.44
Gradient Boosting	0.7	0.67	0.48	0.55
Gaussian Naive Bayes	0.67	0.58	0.58	0.58
AdaBoost	0.61	0.5	0.39	0.44
XGBoost	0.69	0.65	0.50	0.56
Random Forest	0.72	0.68	0.52	0.59

Table 6. CASE 4: Correlation-based Feature Selection (with feature creation).

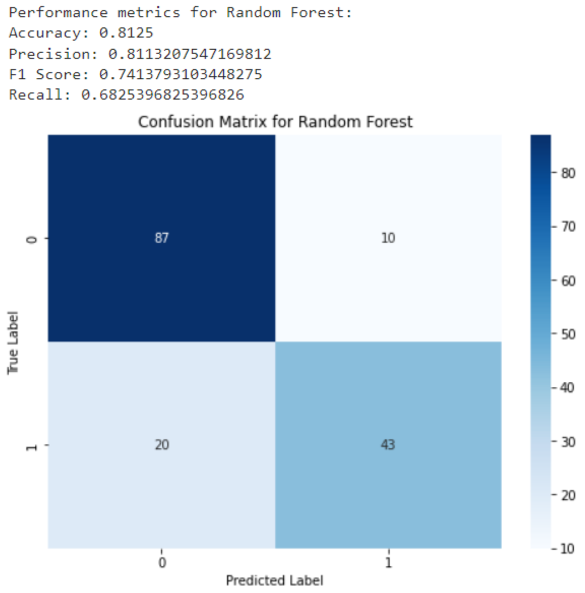
	Acc.	Prec.	Rec.	F1 Score
Logistic Regression	0.79	0.81	0.62	0.70
Decision Tree	0.71	0.65	0.59	0.62
Gradient Boosting	0.79	0.75	0.68	0.72
Gaussian Naive Bayes	0.78	0.73	0.69	0.72
AdaBoost	0.78	0.77	0.65	0.71
XGBoost	0.79	0.76	0.67	0.71
Random Forest	0.81	0.81	0.68	0.74

Table 7. CASE 5: Recursive Feature Elimination (with reduced features).

	Acc.	Prec.	Rec.	F1 Score
Logistic Regression	0.78	0.75	0.65	0.69
Decision Tree	0.71	0.65	0.56	0.59
Gradient Boosting	0.78	0.75	0.67	0.71
Gaussian Naive Bayes	-	-	-	-
AdaBoost	0.8	0.77	0.69	0.73
XGBoost	0.72	0.66	0.63	0.60
Random Forest	0.77	0.75	0.62	0.68

Table 8. CASE 6: Recursive Feature Elimination (with all features).

	Acc.	Prec.	Rec.	F1 Score
Logistic Regression	0.77	0.74	0.63	0.68
Decision Tree	0.52	0.39	0.40	0.39
Gradient Boosting	0.77	0.74	0.63	0.68
Gaussian Naive Bayes	-	-	-	-
AdaBoost	0.69	0.0.64	0.48	0.55
XGBoost	0.75	0.73	0.60	0.66
Random Forest	0.76	0.76	0.56	0.64



```
Performance metrics for Random Forest:
Accuracy: 0.8125
Precision: 0.8113207547169812
F1 Score: 0.7413793103448275
Recall: 0.6825396825396826
```

Figure 13. Confusion matrix for the best performing model(**Random Forest: CASE 4)**

8.2. Analysis of the results

This analysis provides insights into the performance of different classification models and feature engineering techniques in predicting football match outcomes using machine learning algorithms. It underscores the importance of experimentation and highlights the strengths and limitations of various approaches in optimizing predictive accuracy and model interpretability. Here is a breif analysis of the absolute results tabulated above :

8.2.1 Performance Across Models:

When analyzing individual model performance, it's evident that Random Forest, Gradient Boosting, and XGBoost consistently outshine Logistic Regression, Decision Trees, and AdaBoost across all feature engineering cases. These tree-based ensemble methods leverage multiple decision trees to significantly enhance prediction accuracy and effectively manage the complexities within football match outcome data.

Surprisingly, Logistic Regression demonstrated competitive performance despite its simplicity and interpretability.. However, Decision Trees fell short, achieving notably lower accuracy (62%) compared to other models, all of which boasted accuracy rates surpassing 70%. This difference in performance could be attributed to Decision Trees' inherent susceptibility to overfitting, especially when dealing with datasets containing a multitude of features. Additionally, decision trees are weak learners which may have hindered its ability to capture intricate relationships within the data effectively.

Overall, these findings underscore the critical importance of selecting models tailored to the dataset's characteristics and problem domain. They also highlight the necessity of thorough experimentation to identify the most effective combination of models and feature engineering techniques. Despite Logistic Regression and AdaBoost showcasing competitive performance, the consistently high accuracy rates achieved by Random Forest, Gradient Boosting, and XGBoost reaffirm their superiority in handling the complexities inherent in football match outcome prediction tasks.

8.2.2 Performance Across Feature Engineering Cases:

In analyzing the performance of the six classification models across different feature engineering cases, it was observed that Case 4, involving both Correlation-based Feature Selection and Feature Creation, consistently yielded the highest accuracy. This can be attributed to the combined use of these two techniques, which effectively enhance the model's predictive power by selecting relevant features manually after creating new informative ones.

Furthermore, Case 5, which incorporates Recursive Feature Elimination in addition to Correlation-based Feature Selection and Feature Creation, demonstrated the 2nd best performance. The iterative nature of Recursive Feature Elimination allows it to select the most important fea-

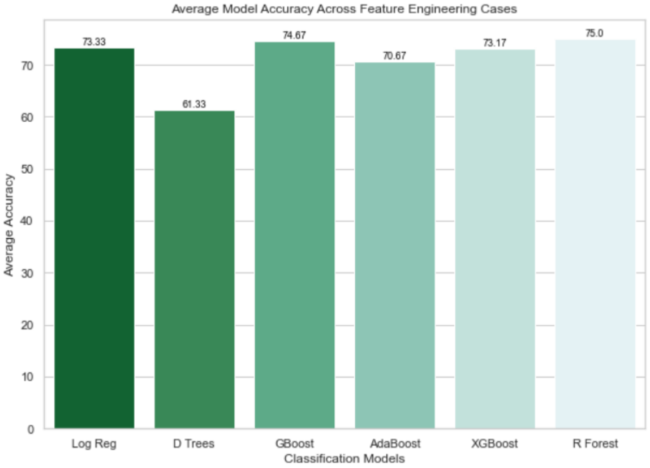


Figure 14. **Average model accuracy** across different feature engineering cases

tures while discarding less relevant ones, contributing to the model's enhanced accuracy.

Conversely, Cases 2 and 6 exhibited comparatively moderate performance(71% accuracy). Case 2 solely relies on Feature Creation without considering feature selection or elimination, potentially resulting in the inclusion of redundant or irrelevant features, thus hindering predictive accuracy. On the other hand, Case 6 combines Feature Creation with Recursive Feature Elimination, but this lacks manual correlation-based feature selection. This undermines the importance of manual correlation-based feature selection, which is important when the number of features are more. Case 1(No feature engineering) and Case 3(no feature creation) performs bad which suggests that creating features is very crucial for the models to perform well !

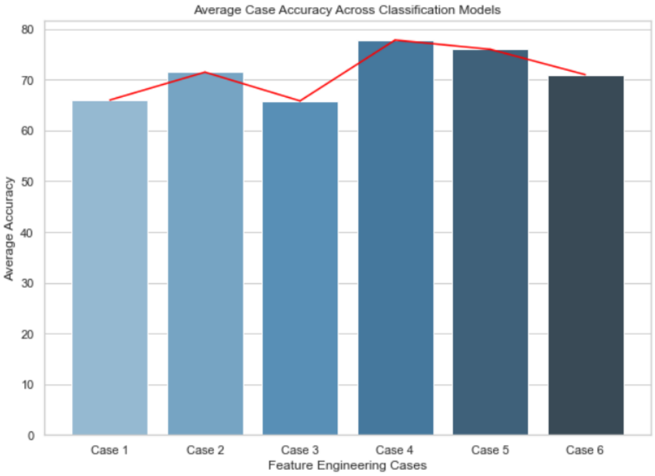


Figure 15. **Average case accuracy** across classification models

9. Conclusion

In this project, we set out to predict football match outcomes using machine learning algorithms and various feature engineering techniques. Football match data of the EPL(English Premier League) spanning 2022-2024 seasons were scraped using Beautiful Soup from the Fbref website to gain as much features as possible. Through extensive experimentation and analysis, we gained valuable insights into the effectiveness of different models and methodologies in handling the complexities inherent in football match prediction tasks.

Our findings revealed that certain feature engineering cases, particularly those combining Correlation-based Feature Selection with Feature Creation, consistently yielded the highest accuracy. This underscores the importance of leveraging synergistic techniques to enhance predictive power by selecting relevant features and generating new informative ones.

Furthermore, our analysis highlighted the superior performance of tree-based ensemble methods such as Random Forest, Gradient Boosting, and XGBoost, which effectively captured complex relationships within the data. Our best model i.e the Random forest of Case 4 achieved high accuracy(81%) and high recall, precision and F1 score as well, which is quite impressive for a complex task like football match prediction. However, one thing which helped this was because we limited our analysis to one league and used data from recent years(2022-2024) .

In conclusion, our project contributes to the growing body of knowledge in football match prediction and underscores the significance of data-driven approaches in sports analytics. Our project emphasizes the critical importance of thorough experimentation with different models and feature engineering techniques to identify the optimal combination for maximizing predictive accuracy and model interpretability. By harnessing the power of machine learning and feature engineering, we can gain deeper insights into the dynamics of football matches and make more informed decisions in sports management and betting contexts.

10. Future Work

It's exciting to see the progress we've made in achieving 80% accuracy with our current models, but there's always room for improvement. Here are some areas we plan to focus on to enhance our football match prediction system:

10.1. Team profiles

Currently, our models do not fully account for the impact of team skill and strategy on match outcomes. In reality, factors such as **team division, in-game tactics and even twitter sentiments** play a significant role. To address this, we plan to pre-compute additional team-specific features, such as division ranking, home/away performance, and halftime strategy tendencies. By incorporating these features into our models, we can better capture the nuances of team dynamics and improve prediction accuracy.

10.2. Player-centred models

Another aspect we aim to integrate into our prediction system is player-specific information. The presence of star players or key injuries can greatly influence a team's performance. By incorporating data on team rosters, player skills, and past performances, we can develop player-centered models that provide more granular insights into match outcomes.

10.3. Live Prediction and Betting Odds

Moving forward, we plan to explore the integration of real-time data and various betting odds as additional features in our prediction models. Live match statistics, such as possession, shots on goal, and fouls, can provide valuable insights into in-game dynamics and help refine our predictions. Additionally, considering various betting odds, such as home/draw/away (HDA) odds, over/under odds, and Asian handicap odds, can further enhance our models' predictive capabilities by incorporating market sentiments and expert opinions.

10.4. Betting analysis

Another potential future extension to this project could be to look at betting odds to see whether our model could recommend good value bets and profitable betting strategies that generate a profit on the long run

10.5. Explore more feature selection methods

To optimize our prediction models, I intend to explore various feature selection methods, including univariate correlation analysis (e.g., KBest) and dimensionality reduction techniques such as Principal Component Analysis (PCA). These methods can help identify the most relevant features while reducing model complexity and improving interpretability.

By incorporating these enhancements, we aim to develop a more robust and accurate football match prediction system that accounts for team dynamics, player performance, and real-time match conditions. These future directions represent exciting opportunities to further advance the state-of-the-art in sports analytics and provide valuable insights for coaches, analysts, and betting enthusiasts alike.

11. Github Link

For complete implementation details, please refer: https://github.com/Donal-08/Data-Science-Final-Project.

References

[1] D. Prasetio et al. Predicting football match results with logistic regression. In *2016 International Conference On Advanced Informatics: Concepts, Theory And Application (ICAICTA)*, pages 1–5. IEEE, 2016. **[Regression]**

[2] J. Brooks, M. Kerr, and J. Guttag. Using machine learning to draw inferences from pass location data in soccer. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 9(5):338–349, 2016. **[KNN]**

[3] B. F. YILDIZ. Applying decision tree techniques to classify european football teams. *Journal of Soft Computing and Artificial Intelligence*, 1(2):86–91, 2020. **[Decision Trees]**

[4] A. T. Oluwayomi, A. O. Olajide, A. A. Adetayo, A. O. Gabriel, O. A. Okunola, and O. T. Gabriel. Evaluation of team's false '9' for match winner prediction. 2022. **[SVM]**

[5] O. Hubá˘cek, G. Šourek, and F. Železny. Learning to predict soccer results from relational data with gradient boosted trees. *Machine Learning*, 108(1):29–47, 2019. **[Boosting]**

[6] A. Groll, C. Ley, G. Schauberger, and H. Van Eetvelde. A hybrid random forest to predict soccer matches in international tournaments. *Journal of quantitative analysis in sports*, 15(4):271–287, 2019. **[Random Forest]**

[7] Y.-C. Hsu. Using convolutional neural network and candlestick representation to predict sports match outcomes. *Applied Sciences*, 11(14):6594, 2021. **[CNN]**

[8] Q. Zhang, X. Zhang, H. Hu, C. Li, Y. Lin, and R. Ma. Sports match prediction model for training and exercise using attention-based lstm network. *Digital Communications and Networks*, 2021. **[LSTM]**

[9] P. Malini and B. Qureshi. A deep learning framework for temperature forecasting. In *2022 7th International Conference on Data Science and Machine Learning Applications (CDMA)*, pages 67–72. IEEE, 2022.

[10] R. P. Bunker and F. Thabtah. A machine learning framework for sport result prediction. *Applied computing and informatics*, 15(1):27–33, 2019. **[ANN]**

[11] Jason Brownlee. "RFE Feature Selection in Python." Machine Learning Mastery. Available online: https://machinelearningmastery.com/rfe-feature-selection-in-python/.

[12] Towards Data Science. "Feature Extraction using Principal Component Analysis - A Simplified Visual Demo." Available online: https://towardsdatascience.com/feature-extraction-using-principal-component-analysis-a-simplified-visual-demo-e5592ced100a.