School of Computing
FACULTY OF ENGINEERING AND
PHYSICAL SCIENCES

UNIVERSITY OF LEEDS

Final Report

Machine Learning Models for Probabilistic Football Analysis and Prediction

Alexander Tim Redshaw

Submitted in accordance with the requirements for the degree of
BSC Computer Science: Digital & Technology Solutions

2023-2024

COMP3932 Individual Project

The candidate confirms that the following have been submitted.

Items	Format	Recipient(s) and Date
Final Report	PDF file	Uploaded to Minerva (30/04/24). Supervisor and assessor informed.
Link to online GitHub repository	URL	Sent to supervisor and assessor (30/04/24)

The candidate confirms that the work submitted is their own and the appropriate credit has been given where reference has been made to the work of others.

I understand that failure to attribute material which is obtained from another source may be considered as plagiarism.

(Signature of Student) Alexander Tim Redshaw

Summary

Football has an inherent problem in the scoring mechanism that defines the sport. Goals are ultimately scarce, and as such do not paint the full picture in evaluating team performance. Traditional metrics like shots and possession can be used to make inferences on match events, but fail to fully capture the complex nature of the game. This project aimed to explore the use of machine learning models in building upon the insights of commonly available metrics. This involved an understanding for their capability in analysing the results of events in the past, in real-time and in the future. The project achieved this through the development of a couple of key models and modelling processes.

An probabilistic expected goals (xG) model was developed, to indicate the probability of a shot resulting in a goal based on positional and contextual factors. The positive correlation between the outputs of the model, and actual goals, despite their scarcity, identifies the benefits of using xG as an evaluative measure. By quantifying the goal likelihood from shots, it becomes clearer which team was likely to score more shots, rather than just who took the most shots. The model ultimately introduces a measure of quality between shots, hidden by a traditional shot count.

The effectiveness of this model was then evaluated in analysing previous matches using Monte Carlo Simulation, enabled by the probabilistic nature of the xG values. Simulating all shots in a game for both sides introduced a variance in the combinations of goals scored. Doing this multiple times allowed for calculation of the average points per game, giving the expected points value (xPts) based on the expected goals. This showed a significant, strong correlation with the actual points, and finishing position, a team achieved.

These expected goal and expected points inputs were used to explore the effectiveness of probabilistic metrics on match prediction models. The inclusion of averages and rolling averages for these probabilistic input features had almost no impact on model performance. This indicates that whilst effective in analysing previous performances, the probabilistic metrics are no better at capturing contextual influences for future matches than the traditional metrics. These are the complex dynamics that make match prediction so difficult in the first place.

Overall, the project emphasises the potential performance benefits from the application of machine learning model outputs. Using the data produced to inform decision making is invaluable in tuning the tactical approaches employed in matches.

Acknowledgements

I would like to thank my Supervisor Roy Ruddle for his continued support, guidance and expertise across all stages of the project development lifecycle. I would also like to thank Layik Hama and Liqun Liu for supporting these supervision meetings, and making themselves available to give guidance and bounce ideas off of.

I would also like to thank my Assessor Duygu Sarikaya for offering feedback on the direction of my project at its inception, as well as midway through development.

Lastly I would like to thank my family and friends, for their continued support and role in nurturing my passion for sports analytics.

Contents

Chapter 1

Introduction and Background Research

1.1 Introduction

Football is renowned for its inherent unpredictability. Despite the multitude of variables involved, match outcomes ultimately hinge on a single quantifiable metric - the number of goals scored by each team. The vast and unpredictable nature of these factors at play mean that not all shots are equal, and that the likelihood of a specific shot resulting in a goal is highly nuanced. This has been combated thanks to developments in data capture technology within the sport, helping to increase the possibility of data-informed footballing approaches [10]. Subsequently, new opportunities have arisen for the analysis of match dynamics and their overarching influence on goals and goal-scoring opportunities. Enhanced understanding and optimisation in this area is a problem that ultimately influences a wide-range of stakeholders. Significantly, a 2018 international cross-market survey identified that approximately 43% of people across the 18 international markets studied were 'interested' or 'very interested' in football [20]. This clearly indicates that it is not only the employees of teams that have a vested interest in the practical applications of football data, but also millions of supporters worldwide.

Furthermore, the financial stakes within the sport are substantial. In recent years, the FIFA World Cup alone has brought in approximately 3 billion dollars in media rights income [21]. Domestically, Deloitte reported that the top 20 revenue generating clubs produced 10.5 billion euros across the 2022/23 season [7]. The financial landscape highlights the overarching disparity in wealth distribution, underlining the importance of using data-informed approaches to help clubs gain a competitive edge, therein bridging the gap to the financial powerhouses. In the modern world, the pursuit of understanding this competitive advantage extends beyond club and media revenues to include an ever-growing bookmakers' market, thriving as a consequence of the increased availability of online betting opportunities [27]. The development of sophisticated predictive models is also highly relevant in this area, offering valuable insights to betting customers, whilst allowing bookmakers to optimise their odds and minimise financial risk. Given the market-type, understanding the ethical implications is highly important. This is discussed further in *Appendix A.3*.

This project aims to validate the capability of xG focused machine learning models in building upon the effectiveness of commonly existing metrics, both in understanding and forecasting football matches. This will primarily be done through the development of a regression-style model to predict the probabilities of shots resulting in goals, an Expected Goals (xG) model. These models generate a probability based on their confidence in classifying a shot as belonging to the positive class - a goal (1). This yields a probabilistic value indicating the likelihood of a shot resulting in a goal, falling in the range $0 \leq xG \leq 1$. Refinement of these Expected Goals models will then allow for further examination into the wider benefits of this style of probabilistic metric. Specifically, this entails understanding their effectiveness in conducting analysis on previous match results, as well as understanding their effect on the performance of models designed to predict the outcomes of future fixtures.

Developing a probabilistic approach to football analysis in this way will better capture the nuances that influence goal-scoring and match outcomes, helping to reduce the noise derived from the inherent unpredictability of the sport. In doing so, the models will help to improve strategic planning and analysis, creating new opportunities for further data-informed decision making processes.

1.2 Literature review

1.2.1 Traditional Approaches to Football Analytics

Historically, football analytics has been built on observational subjectivity and primitive, easy to measure, quantitative values. Prior to the early 21st century, any data beyond scores, lineups and attendances likely had to be captured independently [11]. Even then, insights interpreted from these would not have the depth to fully capture a nuanced understanding of the game. Baseline metrics like goals and assists fail to capture the interrelationship of players' actions with one another on the pitch, obfuscating other potential key player impacts in scenarios like defending or build-up play. Team-wide stats like possession percentages are not only slightly more difficult to capture, but give no real indication as to what a team is doing when they have the ball.

At this stage, critical judgement from those with experience of the game was an integral part of football analysis. However, the intrinsic subjectivity, and susceptibility to biases based on past allegiances, created inconsistency. Pundits were well aware of the role tactical theory played in football matches, but the lack of data availability made it difficult to acknowledge the mathematical principles at play [33]. There was no real foundation for them to support their expert understanding with a concrete quantitative approach.

The emergence of modern data capture bodies, such as Opta (Statsperform), has accelerated football's data generation space, with modern wearable Electronic Performance Tracking Systems (EPTS) also opening up new opportunities. EPTS thrive in dynamically capturing aspects related player coordinates and velocity multiple times per second. However, the capture of specific event data, such as the specifics pertaining to passes and shots, is still a manual process. Subsequently, availability of this data is often restricted to only the most popular competitions [38]. So whilst modern technological advancements have built on traditional approaches to football analytics, poor data availability continues to hinder analytical process at lower-levels of the game.

Regardless of lacking data availability at the time, Charles Reep stands out as a pioneering example for the capability of quantitative football analysis. Beginning in 1953, Reep independently recorded game sequence events for over 500 games [11]. This culminated in early observations on goal likelihoods based on different game sequences, with correlation analysis conducted between goals and match events [22]. It is clear that the potential insights that could be gained with greater, more accurate, data collection processes, is huge. Reep and Benjamin built on these original efforts to then lay the groundwork of a foundational probabilistic goals model.

1.2.2 Significance of Probabilistic Metrics

Reep and Benjamin categorised shots into probabilistic contours based on their spatial position to goal. Notably, kicked shots inside the penalty area, inside the contour formed by extending 18 yard lines 45° from the goalpost, were scored in 0.189 instances on average. In contrast, shots originating from inside the penalty area but outside the defined contour has a substantially reduced average conversion rate at 0.014 [23].

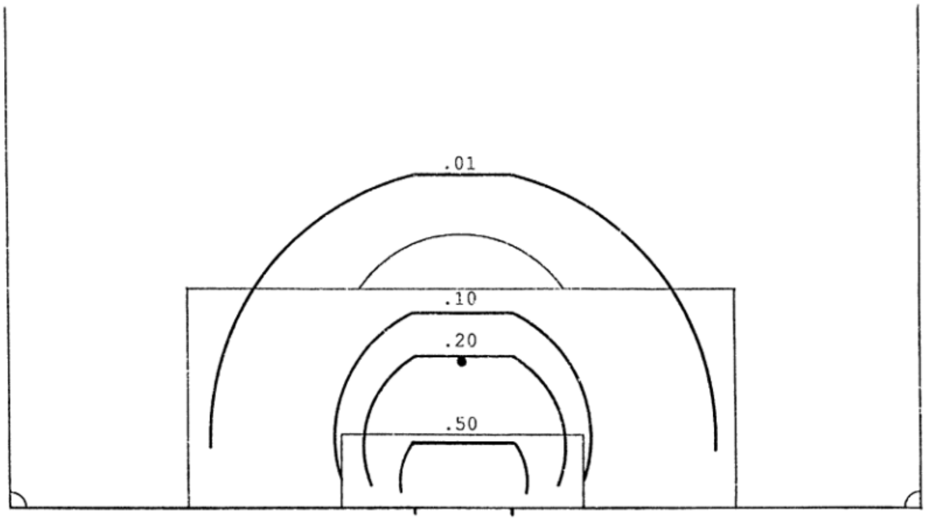

Figure 1.1: Reep & Benjamin's scoring probability contours for kicked shots from open play and from less than 1 yard from the nearest opponent [23].

Insights like this into optimal shooting positions and the different overall yields of specific zones and events changed the tactical landscape. Offering these newfound ideas ultimately encouraged a pressing style of football, which has since seen great success in top leagues across Europe [2]. It is evident that this data availability allows optimisation in developing a style of play, but also has wider benefits in terms of player profiling and recruitment. Given the concepts explored also underpin more sophisticated xG models, there is significant potential. Present-day technology produces an abundance of high-granularity data points, paving the way for new cutting-edge models to interpret the intricacies of the game in new ways, and with greater precision, than before.

While the traditional analytical metrics are capable of identifying broader correlations, they fail to fully explain the causality behind these relationships. Being able to understand this is key in facilitating astute tactical insight. For example, there is a positive correlation between the shots a team takes and the number of goals they score. However, interpretation of correlations without considering wider footballing context can be unreliable [11]. Even though ten shots from the halfway line and ten shots from the penalty spot could both yield the same number of goals, we intuitively know they are highly unlikely to. This is ratified by *Figure 1.1.* In this way, probabilistic values provide a more balanced representation as to the true likelihood of events occurring, factoring in additional contextual constraints that may be underrepresented by the traditional metrics. Using this to understand the areas of the pitch that a shot is most likely to yield a goal from is hugely beneficial. This is because it can be used to alter build-up play and chance creation patterns to maximise the number of chances in these 'higher probability' areas. Furthermore, the models that produce these values can provide transparency and additional domain knowledge into feature importances and variable sensitivity. These capabilities are essential in providing improved clarity over the complex interrelationship of variables and game scenarios, often hidden by traditional metrics. It is in this way that modern machine learning algorithms can improve the specificity of xG values, by breaking from the zonal structure first identified by Reep and Benjamin. The adoption of more sophisticated machine learning methods, such as logistic regression and random forest models, can account for the interrelationships of a wider number of variables on the probability of a shot resulting in a goal. This allows for xG predictions to be made on a more individual level.

Brentford FC are a high-profile example for the effectiveness of these streamlined xG models in shaping style and recruitment in the modern game. Statisticians applied a relative xG model to predict player performance at Brentford, in spite of the current division they were playing in. Using these probabilistic metrics to evaluate player performances has enabled shrewd recruitment to support their promotion to the Premier League. This approach of targeting undervalued players also created significant profit margins in the instance where players were sold [34]. Embracing this probabilistic, data-driven approach enables teams to build upon key tactical insights and consider opportunities for undervalued player acquisition. The all-round combination of practical and financial benefits for this method of evaluating player performance highlights the importance of leveraging machine learning capabilities for analytics in the modern game.

1.2.3 Machine Learning Libraries

When it comes to the development of these machine learning models, Python has solidified itself as the prevailing language of choice [25]. The range of available libraries, coupled with my familiarity of the language, make it the ideal language to develop in.

Scikit-learn

Scikit-learn stands out as an library well-suited for the use-case of this project. Amongst other functionality, it is capable of completing classification and regression tasks, allowing experimentation with different algorithms with minimal code changes [12]. The efficiency in being able to explore different modelling options is particularly appealing. Given it has become one of the most popular machine learning libraries [12], it has a highly active pool of developers providing a wealth of online support, and a well-documented API. Including my baseline familiarity with development in the library, *scikit-learn* stands out as an excellent option to enable sufficient development of diverse, complex models within the timeframe of the project.

PyTorch, Keras & TensorFlow

Despite the appeal of *scikit-learn*, it is important to consider the potential benefits of using libraries that enable other strains of model development. Other such considerations here are the use of deep-learning libraries, such as *PyTorch*, *Keras* and *TensorFlow*, to implement neural networks. However, these libraries, particularly TensorFlow, tend to have a steeper learning curve [37]. Additionally, their increased complexity reduces the interpretability of model outputs [16]. The capacity to understand model outputs is fundamental in acknowledging the underlying influence of variables on predictions, as harnessing these can further elevate match insights. For these reasons, neural network based approaches are deemed less suitable for the expected goal and outcome models the project aims to deliver.

XGBoost & Statsmodels

Another open-source machine library worth considering is the *XGBoost* library. *XGBoost* here stands for *Extreme Gradient Boost.* In the case of this project, is important to distinguish that this has nothing to do with *expected goals*, which is also abbreviated to *xG*. Boosting seeks to transform weak learners into strong learners through recursive error improvement [39]. Thus, gradient boosting uses a model aggregation method to combine the results of weaker models in order to enable better predictive capabilities [8]. This ensemble approach helps bring the performance of decision trees in line with other modern machine learning methods. Whilst this does make extreme gradient boosting an appealing model option to explore, it is possible to implement this in *scikit-learn*. *Scikit-learn* also enables another form of ensembled decision tree model, the random forest model. It is for this same reason that a library such as *Statsmodels* is discounted. Whilst it has a suite of tools to enable regression model development, it primarily focuses is on statistical and econometric analysis as opposed to predictive machine learning [30]. As such, it makes more sense to develop models under the framework of a single library. Doing so will ensure ease of development, as opposed to dealing with different variations in implementation across multiple libraries.

Considerations for Model Selection

Considering all these factors, it is clear that *scikit-learn* provides the most suitable library for the use-case of this project. The breadth of model options and documentation availability will ensure efficient model development. In-turn, this will allow extensive research into the strengths and weaknesses of different model types, ultimately enabling the development of a better model. It's also important to consider the other libraries that will be used to interact with these models. *Pandas* and *Numpy* provide significant functionality for data analytics procedures, enabling efficient processing even for large complex datasets. It is for these reasons it is often used with analytical and visualisation libraries [17], making it highly suited for use in this project. Whilst there is perhaps more variety in the visualisation libraries to use here, prior experience with *Matplotlib* provides confidence in proceeding with this project. This specifically pertains to experience in developing plots incorporating custom drawings, as will be necessary for visualising data in the context of the football pitch. This combination of data processing, visualisation and machine learning libraries creates a cohesive foundation to develop the project upon.

1.2.4 Machine Learning Algorithms

Having established the key libraries necessary for bringing the project together, it is also worth specifically considering which kinds of underlying models should be used. As previously identified, the use of an XGBoost model has strong performance capabilities. However, this style of model can raise serious concerns over the transparency provided in the feature selection process. This is due to the tendency to underestimate the importance of key features, whilst overestimating the importance of irrelevant features [9]. Interpretable insights are invaluable in further enhancing tactical insights based on the feature contributions. Subsequently, this project will prioritise algorithms with a higher degree of interpretability, namely logistic regression and random forest models.

Logistic Regression

Scikit-learn implements logistic regression as a linear model for classification, where a logistic function is used to model the probabilities that describe the outcome of a given trial [5]. The baseline binary classification approach makes it suitable for both an expected goals model (no-goal vs goal), and a predictive outcome model (win vs not-win). Whilst match outcomes are technically not binary (win, draw or loss), these can still be modelled using a multinomial logistic regression. Adopting a logistic regression approach here is advantageous for model transparency. Access to clear model coefficients allows for analysis into the extent to which different variables alter model predictions. Interpreting the model in this was allows for the abstraction of additional tactical insights. However, the aforementioned linear implementation of the model may be insufficient in capturing the complexity of feature interactions, particularly in the case of the expected goals model.

Random Forests

Given the complexity limitations of implementing a logistic regression model, it is worth considering whether a random forest model is more capable for the problems in this project. Like *XGBoost*, this is a tree-based ensembling approach. By averaging predictions across a randomised set of decision trees, random forest models are able to reduce the susceptibility of ordinary decision trees to overfit to the training data [5]. However, this can sometimes come at the cost of increased model bias. For this reason, it will still be important to carefully tune model hyperparameters using cross-validation techniques. Doing this will validate the replicability of the results produced by the models [19]. Taking this approach for will provide confidence that the models can maintain performance on unseen data samples, and will also be taken for the tuning of any logistic regression models.

1.2.5 Evaluation Methods

Once satisfied with tuned model performance, the models need to be evaluated using contextually appropriate evaluation metrics.

Model Evaluation

Considering the project aims to ultimately produce two different strains of model, it is important to consider the way in which they are evaluated. The predictive outcome model seeks to identify whether a team will win, draw or lose. This classification approach lends itself to metrics expressing the model's performance in terms of true (T) or false (F) positives (P) and negatives (N).

Metric	Measure	Formula
Accuracy	Total correct classifications	$\frac{TP+TN}{TP+TN+FP+FN}$
Precision	Correctness of positive predictions made	$\frac{TP}{TP+FP}$
Recall	Proportion of actual positives found	$\frac{TP}{TP+FN}$
F1-Score	Harmonic mean of precision and recall	$2 \times \frac{Precision \times Recall}{Precision + Recall}$

Table 1.1: Overview of classification metrics [35].

As the predictive outcome model seeks to predict the correct result of games, accuracy is a key metric for measuring model performance here. When one team wins, another team loses, giving an inherent balance to match outcomes. For this reason, precision and recall are not perceived as the primary predictive measure for this style of model. Nevertheless, using these to deepen understanding on model strengths and shortcomings will be beneficial in the tuning process. Given the probabilistic nature of an expected goals model, none of these classification metrics are appropriate in measuring model performance.

Instead, a good way to evaluate model performance will be looking at the size of the errors between the predicted values and the actual outcomes. This is where regression metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) are popular [4].

Metric	Measure	Formula
Mean Absolute Error (MAE)	Average magnitude of errors between the predicted and actual values	$\frac{1}{n}\sum_{i=1}^{n}\lvert y_i - \hat{y}_i\rvert$
Mean Squared Error (MSE)	Average square of errors between the predicted and actual values	$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$
Root Mean Squared Error (RMSE)	Average square of errors between the predicted and actual values, square rooted	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$

Table 1.2: Overview of regression metrics [4].

However, the expected goal value measures the probability of the positive class. Hence, it is more appropriate to use a metric to measure the error between the probability of the positive class and the actual value. This is where the Brier Score is appropriate, measuring the mean squared error between the actual value and the predicted probability, as opposed to the predicted regression value [28].

$$\text{Brier Score} : \frac{1}{N}\sum_{i=1}^{N}(f_i - o_i)^2$$

This will give a clearer indication of the error rate amongst the actual predicted probabilities (the output of the expected goals model). However, this doesn't make other regression metrics redundant. Instead, they are more appropriate in supporting model calibration and refinement.

Evaluating Applied xG

Whilst having a well evaluated model is key to producing sensible xG values, it is also appropriate to evaluate the suitability of their outputs in developing analytical insights. As such, there is a need to affirm the correlation, and the statistical significance, of xG with respect to actual goals.

The aim of the expected goals model is to provide greater statistical transparency over the likelihood of a given shot resulting in goal. As such, there is a need to show a correlation exists between these variables. This can be done using Pearson's Correlation Coefficient. This considers two variables and measures the strength to which they are linearly associated [31]. The p-value can then be used to confirm whether or not the result is statistically significant.

The probabilistic nature of xG values allows them to be utilised in the broader applications aside from a shot-by-shot basis. One such method for this is Monte Carlo Simulations, which conduct repeated random sampling of a probabilistic value [26]. This could be used to randomly sample each shot within a game, simulating the outcome of all shots multiple times for the same game.

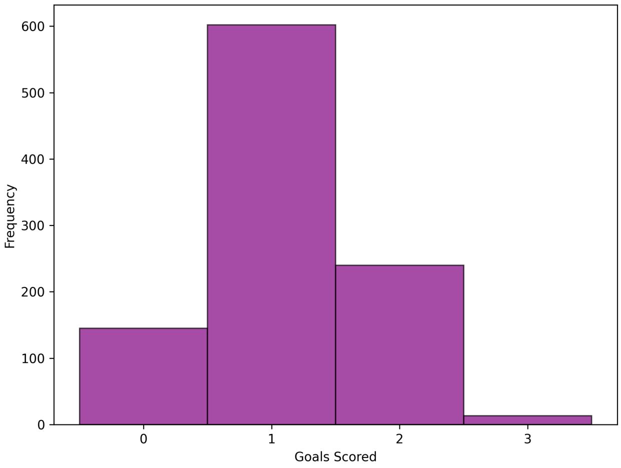

Figure 1.2: Example Monte Carlo simulation (1000 sims) for 3 shots with xG values of 0.3, 0.05 and 0.8 respectively.

Figure 1.2 gives an baseline example for how Monte Carlo Simulation can be applied here. Whilst the total expected goal value is 1.15, the team could score anywhere in the range of 0-3 goals. Simulating the goals for a team and their opponent allows for an average calculation of the number of points expected to be scored from the game. This expected point (xPts) value creates an avenue for a longer-term assessment on the suitability of xG. Correlations between xPts and actual points can then be explored, following a similar methodology to xG and actual goals. Examining this is key in identifying the significance of xG in providing analytical insights on more than just an individual shot basis.

1.3 Project Requirements

The requirements for the project were developed through thorough examination of existing literature in the field of football analytics. Much of this includes the content explores in the *Literature Review (Section 1.2)*. Football's low scoring nature by comparison to other popular sports, necessitates a more nuanced way of evaluating chance quality [3]. A running total of shots alone loses data dimensionality with regards to actual goal likelihood. As such, a machine learning approach is required to provide greater clarity over shot-likelihood values, dependent on other contextual factors. Given the success of current approaches to this, such as the scenario discussed using Brentford FC, it's also important to evaluate exactly why these are beneficial. This includes their capability to improve analysis of past events and the prediction of future events over traditional metrics. These considerations yielded the following requirements:

1. Develop a probabilistic 'expected goals' (xG) model to predict the likelihood of a shot resulting in a goal-scoring event.
2. Analyse how the feature importances of an xG model can be used to alter tactical game-approaches.
3. Evaluate the effectiveness of xG models for analysing previous matches.
4. Develop a baseline model to predict the outcome of upcoming matches.
5. Evaluate the extent to which xG model inputs can improve predictive outcome models.

1.4 Risk Analysis

In order to satisfy these requirements, there are some potential risks that need to be managed:

Risk	**Mitigation**
Reliability of the produced models is largely dependent on the quality of the obtained data	Data will be thoroughly preprocessed to identify any incompleteness or data quality issues. Data will be collated from across multiple different leagues, and numerous data sources will be considered.
The combination of complex data relationships and complex machine learning implementations may lead to poor model interpretability.	Specific machine learning algorithms have been chosen that can provide a more suitable level of interpretability, to boost tactical insights based on underlying model functionality.
Models may fit poorly to the data, providing good performance on training data, but poor adaptation when it comes to unseen data.	Hyperparameters will be optimised using thorough cross-validation tuning processes. Matches that occur after initial data collection will be accumulated separately, allowing model evluation on an additional holdout set.

Table 1.3: Identification of risks and plans for mitigation.

Chapter 2

Methods

2.1 Project Management

Following this literary review period, the project was structured into key project phases, data gathering, data exploration and model development, each of which were split into further modular components where appropriate. Organising the project in this way allowed for coherent dependency management between project phases. As such, this is the method used to compartmentalise the project in the GitHub repository, which facilitated version control across the development process. To further ensure a structured approach across project life cycle, bi-weekly supervision meetings were organised. These were used to manage the workload across the end-to-end project process and get regular feedback on code development phases, supporting the project in an agile-style development approach.

2.2 Software Design

Given the focuses of this project on data science and machine learning processes, *Jupyter Notebooks* offered a suitable python development environment [14]. The ability to split up code workflows into text and cell based fragments helps to keep code clear and organised. Additionally, some of the machine learning processes used in this project (e.g. hyperpameter tuning) are likely to be computationally expensive. By splitting these off into separate code cells, access to other code fragments can be maintained, without the repetition of intense tasks. The key libraries used for development in these Jupyter notebooks can be seen in *Table 2.1.*

Library	**Use-case**
Requests	HTTP requests to facilitate data retrieval.
Pandas	Provision of easy and efficient data manipulation.
Numpy	Used in combination with *Pandas*. Also extensively used when performing calculations for augmented spatial features.
Matplotlib	Generate pitch graphics and general visualisations.
Scikit-learn	Implementation of machine learning processes. Includes train-test splitting, model implementation and quantitative evaluation.

Table 2.1: Overview of the key libraries and their use-cases.

Figure 2.1 helps visualise how the software implementation aligns with the defined requirements. Numerous data implementation steps ensure that the models will be built on large quantities of data with the appropriate features. Development of appropriate models ensures a quality resultant expected goals model *(Requirement 1).* With a suitable model in place, confidence can be placed in the respective feature importance dynamics, to draw tactical insights in optimising game-approaches *(Requirement 2).* The effectiveness of using expected goals to analyse previous matches *(Requirement 3)* can be approached using quantitative

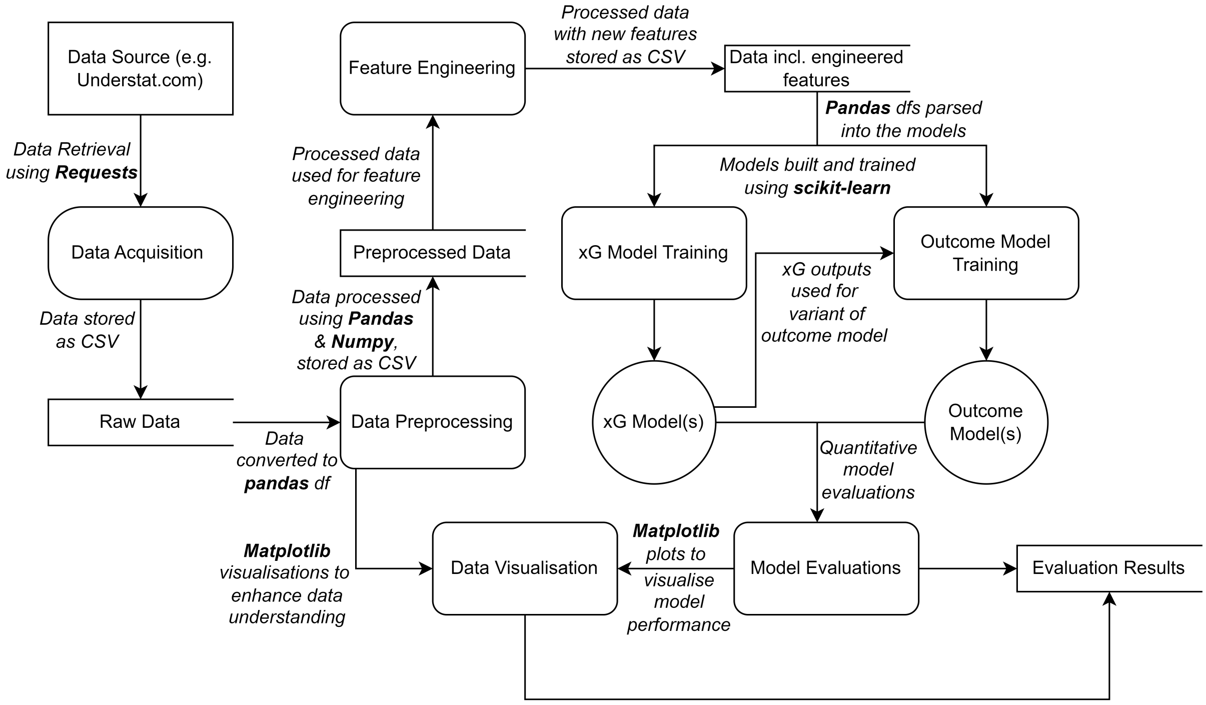
Data Source (e.g. Understat.com)
Data Retrieval using **Requests**
Data Acquisition
Data stored as CSV
Raw Data
Data converted to **pandas** df
Data Preprocessing
Data processed using **Pandas** & **Numpy**, stored as CSV
Preprocessed Data
Processed data used for feature engineering
Feature Engineering
Processed data with new features stored as CSV
Data incl. engineered features
Pandas dfs parsed into the models
Models built and trained using **scikit-learn**
xG Model Training
Outcome Model Training
xG outputs used for variant of outcome model
xG Model(s)
Outcome Model(s)
Quantitative model evaluations
Model Evaluations
Evaluation Results
Matplotlib plots to visualise model performance
Data Visualisation
Matplotlib visualisations to enhance data understanding

Figure 2.1: Planned project data flow diagram.

correlation analysis. Supporting this with Monte Carlo simulation shifts the evaluative lens to a broader context of entire matches and seasons. Using the initial data, a baseline predictive outcome model can be developed *(Requirement 4)*. Here, there is then an opportunity to apply the expected goals model as inputs for the predictive outcome model, such as in the form of averages and rolling averages. At this stage it is suitable to evaluate the effectiveness of xG values for forecasting future events, and building on the baseline predictive outcome model *(Requirement 5)*. With an end-to-end understanding of the processes required to meet the requirements in place, data was now needed in order to kick-start software development.

2.3 Data Gathering

2.3.1 Data Sources

Detailed football data is relatively hard to come by, specifically when it comes to shot data. Datasets like those collected from the publicly available Fantasy Premier League API [1] do provide some overarching seasonal data. However, these are not granular enough to build anything close to an expected goals model, and do not provide data on leagues from a breadth of different countries. As such, it is important to consider data accessible via other methods, such as web scraping.

Sofascore

Sofascore is a site used for live football scores, and provides a host of match data to accomplish this [32]. Further examination of the site identified an API feeding match data, with substantial overarching season information. It also contained highly specific shot data across a number of leagues, capturing a range of contextual features. Whilst the available feature-set from the API made it an appealing source of information for this project, there were a few concerns. Firstly, navigation between endpoints was highly difficult. There was no specific structure or naming convention, meaning UIDs had to be identified for each match. The API offered no convenient way of achieving this. Given the low scoring nature of the sport, maximising the data points available was a key focus. Thus, using this approach to ensure a significant quantity of data capture was deemed infeasible. Furthermore, the data source was not completely consistent, with data availability appearing to evolve in more recent years. This also raised concerns over providing a suitable volume of data for the project.

Understat

Subsequently, *Understat* was considered as a source for information for the project. Understat also provides match across the top leagues in England, France, Spain, Italy and Germany [36]. Some research into using *Understat* as a data source identified an easy method for extracting data as a JSON object from the site's underlying Javascript [6]. Given *Understat's* structured URL structure, this made navigation between match data sources incredibly simple. Whilst the level of contextual information was not quite as extensive as *Sofascore's* data, this made over 9 years of match information across the top 5 European leagues readily available. The combination of data volume, ease of access and feature availability made *Understat* the most suitable data source to pursue for project development.

2.3.2 Data Retrieval

A function was designed to follow the data retrieval process outlined in the previous sub-section. Using the structured URL, the function iteratively searched through each of the 10 available years of fixtures (including the present season for each of the leagues). Here, URLs for all of the fixtures in that league and season were dumped into an array. Subsequently, the function was able to go through each of the fixtures one-by-one, converting the data from a JSON object, and storing it in a CSV file specific to that league. Throttling, i.e. limiting the rate of requests being made to the site, was implemented in order minimise the risk of being blocked or overloading the server [15]. Given data was initially pulled in January 2024, getting data all the way back to 2014, this was a lengthy process. However, eventually this produced a dataset of over $17,300$ matches and $434,000$ shots. This same data retrieval process was used to obtain a smaller sample of data in the matches preceding this period from February to April. These additional data points ultimately made up the holdout set, to allow for unseen model evaluation, as was explored in the literature review.

2.4 Data Exploration

Figure 2.2 shows the distribution of match variables, much of which conforms with what is expected. Neither home nor away goals identify any major outliers in their distributions. Ligue 1 (France) and the Bundesliga (Germany) appear to contain a smaller number of games, but this is also expected, as their leagues contain fewer teams than the rest. Seasonal distribution of matches remains consistent with our expectations. All seasons contain the same number of games, excluding the current season (which is in progress), and the 2019-20 season. This season

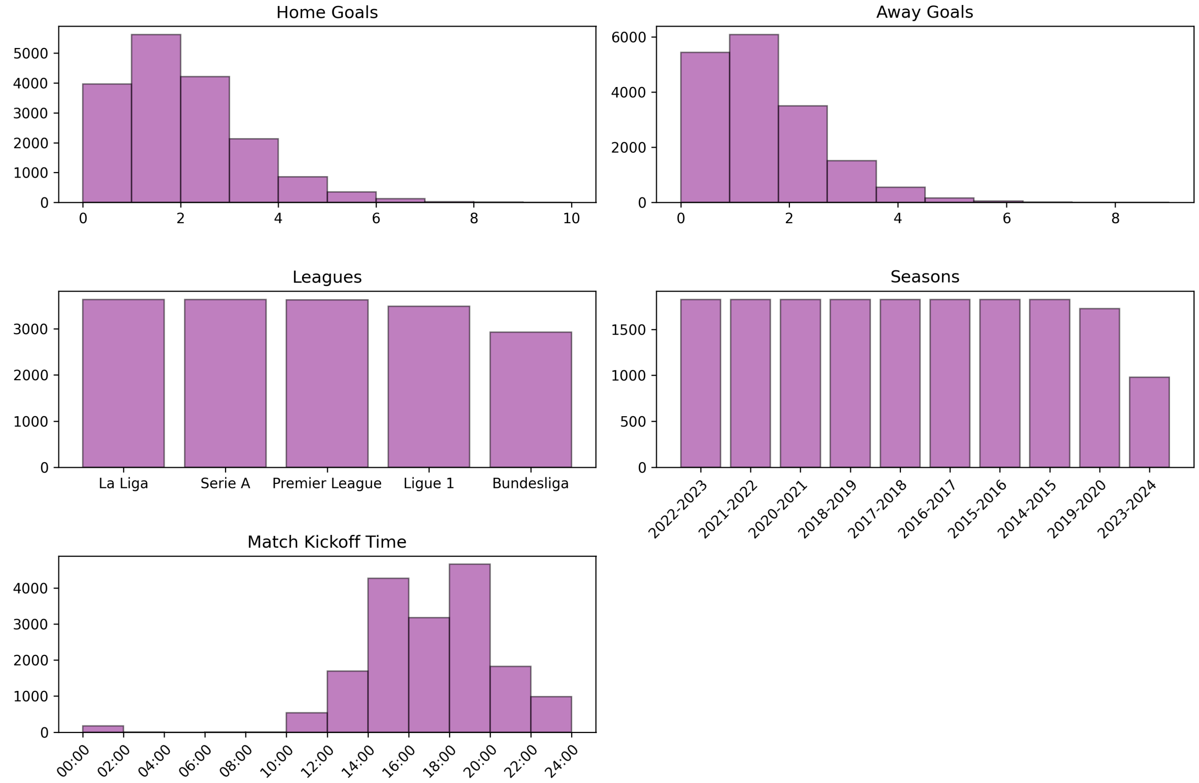

Figure 2.2: Distribution of variables across the matches dataset.

saw an understandable drop in matches, due to the cancellation of matches during the Covid-19 pandemic. The distribution of match kick-off times appears to raise a few questions. Exploring values with a kick-off time between 00:00 and 02:00 provides some explanation surrounding this. Matches with these kick-off times are real matches with existent statistics. However, they did not kick-off at midnight. As such, this appears to be some sort of indication of an unknown or missing match kick-off time. This is important to note if kick-off time makes up a part of the predictive outcome modelling, as these records will be removed. Further manual examination identified some discrepancies in the timings between countries. It did not always appear to be clear, nor necessarily consistent, which timezone these are registered in. As such, it may be best to disregard this when modelling predictive outcomes. Given the shot data concerns spatial values, these can be visualised within the context of a football pitch, using a pitch drawing approach inspired by FC Python [24].

Initial attempts to plot this appeared to show an empty pitch plot. However, examination of the shot values identified normalised values, with x and y values lying between 0 and 1. Given a pitch is rectangular, it was clear these were normalised values. Whilst not an original consideration, this is actually an advantageous feature of this dataset, as regulation pitch sizes can differ [13]. However, as pitch features must remain proportional with respect to pitch size, all shots can be plotted proportionally in spite of this. As such, the pitch was drawn using dimensions akin to Wembley Stadium (105 x 70m), and shot values were scaled to reflect this. Plotting all shots in the dataset produced the shotmap seen in *Figure 2.3*.

Given the sheer quantity of shots plotted on the visualisation, the data is incredibly clustered,

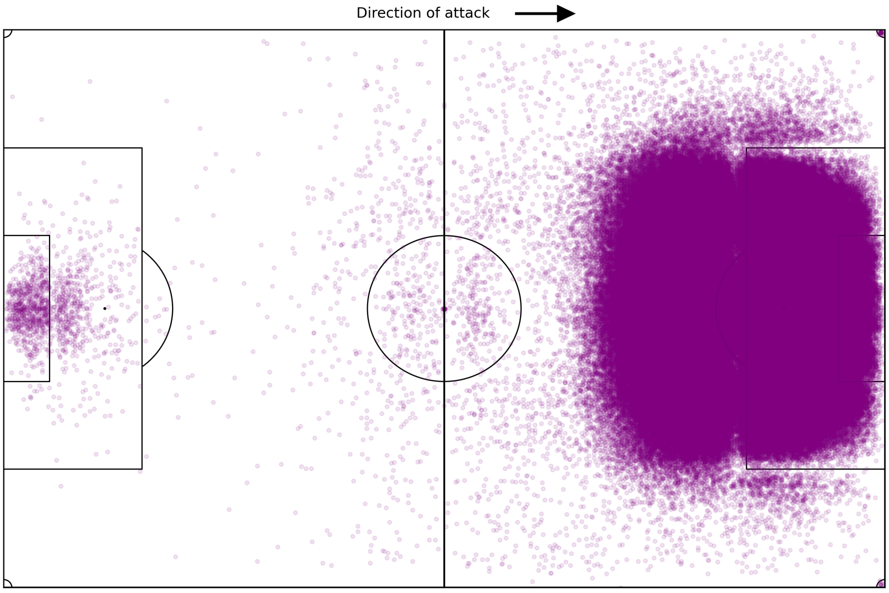

Figure 2.3: Visualising the location of all shots in the shot dataset.

particularly in the vicinity of the opposition penalty area. This follows the expected distribution. However, a shot hotspot appears in a team's own penalty area. Initially, it was presumed that these were mishit goal-kicks that turned into shots, but they did not appear distributed as expected for that scenario. Further exploration identified that the dataset includes own-goals as a shot-based scenario, hence the clustering in this area. By mapping all shot results into three categories: no-goal (0), goal (1) and own-goal (-1), managing shots in the dataset became far simpler, and own goals could be removed. *Figure 2.4* visualises the distribution of these.

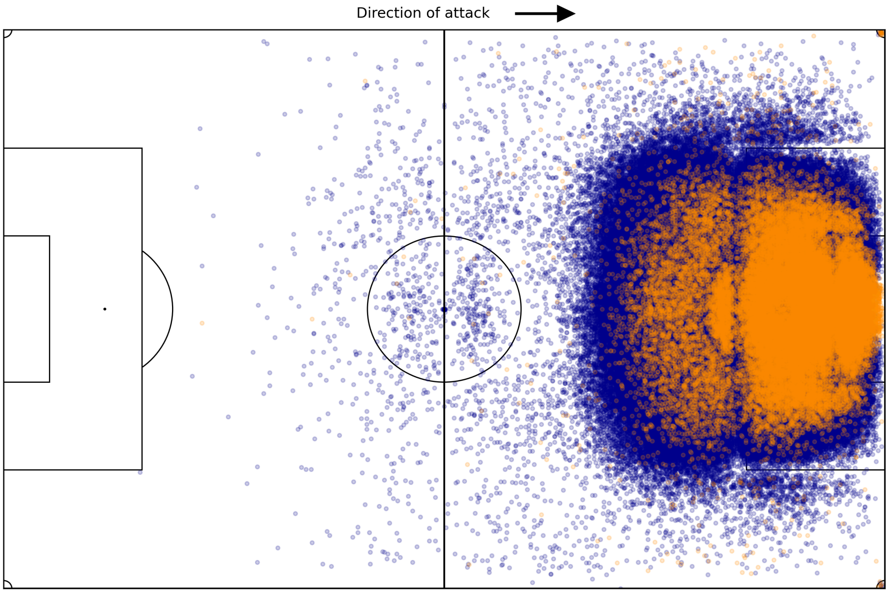

Figure 2.4: A shotmap for shots (excluding own goals) in the dataset. Non-goals are encoded in blue; goals are encoded in orange.

The shot dataset also contains other key contextual fields that support the coordinate information. The possible values for the most relevant fields are listed in *Table 2.2*.

Feature Name	**Feature Description**	**Possible Values**
`situation`	Indicates the scenario that the shot originates from	`OpenPlay, FromCorner, SetPiece, DirectFreeKick, Penalty`
`shot_type`	Indicates the part of the body used for the shot	`RightFoot, LeftFoot, Head, OtherBodyPart`
`last_action`	Indicates the last match playing action before the shot occured.	41 possible values (including `null`)

Table 2.2: Possible values for contextual shot fields.

Whilst training a model using all contextual values would be good, the `last_action` values created significant difficulties. The range of values meant that many actions were underrepresented. 12 of the 40 actions appeared in less than 20 shot scenarios. Additionally, actions were sometimes ambiguous, appearing to have overlap between values. Finally, many shot values had a value of null for this feature. Whilst this could be presumed to be no prior action, the range of possible actions makes this improbable. Further exploration confirmed these were missing values. Subsequently, it was decided this should not be used as a feature for model development.

2.4.1 Data Preprocessing

Having mapped goals into the three categories discussed, own goals were fully removed from the dataset. Their scarcity, coupled with their high inherent unpredictability, meant they did not fit within the scope of this project. Additional exploration was conducted to identify any duplicate shot values, although none were discovered. Finally, any rows containing null values for any of the variables were disregarded. All of the null values appeared in columns that had already been disregarded, such as the `last_action` column, or in columns that would not be suitable for building a predictive model, such as `assisting_ player`. This is because data is inherently too sparse to tune predictive models based on specific shot takers and assist makers. Finally, x and y coordinates were scaled based on the pitch dimensions previously outlined. This would ease the process of engineering additional spatial features.

The preprocessing stage also required consideration for dealing with anomalous values. Given the nature of the game, unlikely shot outcomes do happen, so initially it was decided these should be factored into the probabilities provided. Nonetheless, some experimentation was conducted with regards to model performance by removing different levels of anomalous values. The existence of coordinates in a 2-dimensional space allowed for identification for the other 100 closest shots by euclidean distance. Once identified, and their results calculated, if the shot's outcome was only seen in $\leq n$ neighbours, it could be removed. This necessitated the use of a holdout set.

2.4.2 Feature Engineering

Existence of shots in this 2-dimensional space where also beneficial in allowing the augmentation of additional features. This includes the distance from specific parts of the pitch, such as the nearest touchline, the goal itself, and the width from the centre of the goal. This was simply performed by calculating the distance of x and y coordinates from the fixed pitch boundaries. Additionally, knowing the coordinates for the goalposts, it is possible to trigonometrically calculate the angle between a shot and the goalposts. This indicates the visibility of a shot to goal. Using this should be beneficial to model performance, as the greater the site of goal from a shooting location, the greater the chance of the shot being on target [29]. *Figure 2.5* indicates how this can be done.

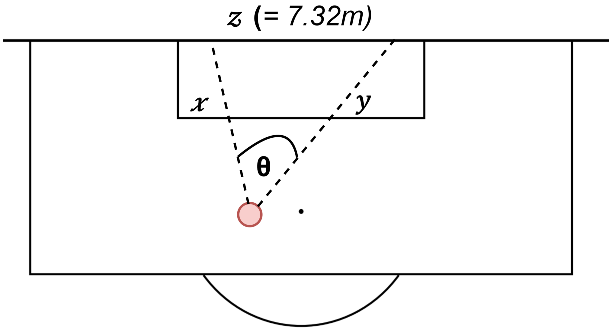

Figure 2.5: Understanding the calculation of the shot angle.

Values for x and y can be simply calculated using Pythagoras' Theorem ($a^2 + b^2 = c^2$), whilst z (the goal width) remains constant at 7.32m. Based on the law of cosines [18]:

$$z^2 = x^2 + y^2 - 2xy\cos(\theta)$$

$$\cos(\theta) = \frac{x^2 + y^2 - z^2}{2xy}$$

Given all values for x, y, and z are known, it is possible to solve the equation, and take the inverse cosine to calculate the angle in radians. Multiplying by $\frac{180}{\pi}$ gives the value in degrees.

$$\theta = \arccos(\cos(\theta)) \cdot \frac{180}{\pi}$$

Finally, contextual features (`situation` and `shot_type`) needed to be engineered into binary values, as logistic regression modelling is unable to deal with continuous features. Pandas provides an easy way to perform this on dataframes, with the *pd.get_ dummies()* function. Here, 1 indicates a true value (e.g. `situation_OpenPlay`), and 0 a false value.

This feature engineering process was far simpler for the match data used in the Monte Carlo Simulation. As the Monte Carlo simulations required xG model outputs, a model was chosen and applied to the shots in the dataset. All shots in a game could then be linked together using the `match_id`. Running 100 simulations each game could then be done, capturing the variance in shot outcomes, and producing an expected points figure.

For the predictive outcome models, pandas could be used to calculate simple averages and (6 game) rolling averages for a team and their opponent in a certain season. This included match expected goal calculations based on the developed xG model, and expected points values from the Monte Carlo simulation. Rolling averages were not rolled over at the beginning of a new season, due to the impact of promotion, relegation and transfer windows. As such, the first 6 games for each team in a season needed to be removed from the dataset, as their rolling average values were `null`. Following these preprocessing steps, all datasets were fully prepared to begin model training.

2.5 Model Training

As previously discussed, based on the literature review, both strains of model were implemented using *scikit-learn*. In order to tune hyperparameters, models used *GridSearhCV* to perform k-fold cross-validation. This included development of a custom hyperparameter tuning process for the xG model, centering optimisation around the Brier score. This did require use of the *joblib* library to enable parallel implementation to reduce the computational expense, which was an interesting challenge in itself.

2.5.1 Expected Goals (xG) Models

The expected goals model was developed using the preprocessed shot dataset. As discussed in the literature review, there was a desire for a degree of interpretability, but also wanting to ensure sufficient performance. As such, both logistic regression and random forest models were developed. Both would be suitable for predicting goal probabilities, allowing a quantitative comparison of the pros and cons of each model. The features ultimately used to train the model were: X, Y, `avg_distance_to_goal, angle_of_shot, width_from_goal_centre` and `proximity_to_touchline` for the basic coordinate based model. The extended model, including extra context, also included the flattened columns for `situation` and `shot_type`.

2.5.2 Predictive Outcome Models

The initial predictive outcome model was developed using contextual match data information, with total averages and (6 game) rolling averages for performance. This process was used for goals, shots, expected goals, points and expected points, both for and against. Opposition stats for these same parameters were also included. Managing the features in this way allowed for a representation of the average, and form-based, attacking and defensive performance of the teams. This allowed for binomial logistic regression classification, such as 'win' vs 'not-win', and multinomial logistic regression, such as 'win', 'draw' or 'loss'. A random forest model was also developed as a point of comparison. Ultimately, the features used in the baseline model were: `goals, goals_conceded, shots_taken, shots_conceded` and `points`. The model including probabilistic features also included `shots_taken_xG, shots_conceded_xG` and `xPts`. Each of these then had a variant prefaced with `opponent_` to indicate oppositions teams performance in these areas. This could finally then be used to produce a `difference` column for each of these variables, indicating the magnitude in performance difference for a team with respect to the feature.

2.6 Model Evaluation

The evaluation of the expected goals models and predictive outcome models built upon the evaluation approaches explored in great detail in the literature review. Due to the differing outputs of the model, the metrics used to optimise and assess both types of models differed.

2.6.1 Expected Goals (xG) Models

The xG models required a regression-based approach, meaning performance metrics needed to focus on error sizes as opposed to classification performance. This necessitates the use of metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) for calibration, and the Brier score for ultimate evaluation. The goal of the model is to produce a probability, and the Brier score considers the difference between the probability itself and the actual outcome. Using the Brier score quantifies the average squared deviation between predicted values and actual outcomes, where predicted xG values fall in the range $0 \leq xG \leq 1$, and a shot has a binary outcome of 0 (missed) or 1 (scored). Hence, the lower the Brier score, the better the predictive accuracy of the model. In terms of evaluating the longer-term analytical capabilities of the final xG model, outputs will be used to perform Monte Carlo simulation on previous games. This will allow for a comparison of the estimated points probabilities (xPts) based on these simulations and team's actual points and league standing. The statistical significance, and correlation coefficient for the relationship between expected points and points (as well as xG and goals) can be also be affirmed using Pearson's Correlation Coefficient. Exploring the correlative performance of the model across an entire season enables a more nuanced understanding for the suitability of xG in conducting analysis on previous events. Hence, evaluation of the expected goals model is not limited to the suitability of the model itself, but also its wider applications. This is relevant for both past analysis, as well as predictive capabilities.

2.6.2 Predictive Outcome Models

In contrast to xG, the predictive outcome models require a classification-based approach, where the objective is to correctly classify the result of a game into an appropriate category. This necessitates a focus traditional classification metrics, such as accuracy, precision and recall. The primary result of any match is the number of points earned as a result of a team winning, drawing or losing. As such, developing a model that priorities accurate classification provides a more suitable predictive objective than attempting specific scoreline forecasting. Development of a predictive model with and without probabilistic inputs will enable a quantitative comparison for the benefits or drawbacks of their use in predictive models. As a result of the game's low-scoring nature, it is also imperative to consider the classification approach used. Models can be developed to predicts whether a team wins, draws or loses, but may be more successful in predicting whether a team wins or does not win (including draws). As such, it is sensible to explore the influence of probabilistic inputs for each of the classification approaches.

Chapter 3

Results

3.1 Software Implementation & Testing

The code implementations were ultimately were split into a number of different Jupyter Notebook files, enhancing the overall maintainability of the code-base. Within these files, many elements made use of a modular design approach to increase code reusability. This was particularly effective in the development of functions for the xG models' custom hyperparameter tuning, performance comparisons on the evaluation set, Monte Carlo simulations and the preprocessing steps of the predictive outcomes models.

A number of approaches were taken to testing individual functions, model outputs and overall applicability. Mathematical processes used as a part of feature augmentation were tested in a unit fashion, including methods such as the calculation of shot angle. Testing values in this fashion ensured that augmented features were consistently correct across all data points. Additionally, integration testing explored the interconnection of the different stages of model development. This ensured that each element of the model pipeline fitted together, producing results roughly aligning with pre-conceived expectations.

In terms of model testing, holdout data was used to allow for an equal comparison of all models. Whilst this did provide a level playing field for model comparison, it also ratified the results for models with no 'anomalies' removed. All of these models saw at most ±0.01 fluctuation in Brier scores between test and holdout set performance, indicating the models were well fitted for unseen data generalisation. The outputs of the expected goals model could also be visualised as a pitch heatmap, allowing for qualitative evaluation. The combination of quantitative and qualitative evaluation processes was key in understanding differences between model outputs.

The Monte Carlo simulations acted as the foundation for previous result analysis, through expected points modelling, and were also tested for robustness. This involved running the entire simulation process multiple times, to ensure there was no great fluctuation in average deviations. Across this repeated process, the average deviation between the expected league position and actual league position only fluctuated by ±0.15 between simulations. This was based on 294 different 34-38 game seasons per simulation. The lack of variance between simulations instilled confidence in the robustness of this simulation approach.

3.2 Expected Goals (xG) Model Performance

3.2.1 Initial Model Performance

The performance metrics (*Table 3.1*) identify minimal deviation in performance between differing xG model types for the 'basic' and 'extended' variations. The marginal improvement seen in the extended models highlight the role that wider shot context plays in reducing the error of predictive probabilities. For example, headers have significantly reduced probabilities

Basic (coordinate-based) Models

Model	Brier Score	MAE of xG Value	RMSE of xG Value
Logistic Regression	0.080	0.159	0.282
Random Forest	0.079	0.158	0.281
Extended (situation and shot type context) Models			
Logistic Regression	0.077	0.153	0.277
Random Forest	0.076	0.153	0.276

Table 3.1: Performance metrics (3sf) for basic and extended xG models (lower values are better).

over right or left footed shots, due to the difficulty of generating enough power to beat the goalkeeper. However, this is actually also encouraging in highlighting the benefits of using a basic model. Whilst offering slightly sub-optimal performance, it requires less contextual match data, which is often scarce in data sources. This likely makes it a more accessible model for games with less publicly available data, such as lower-division games.

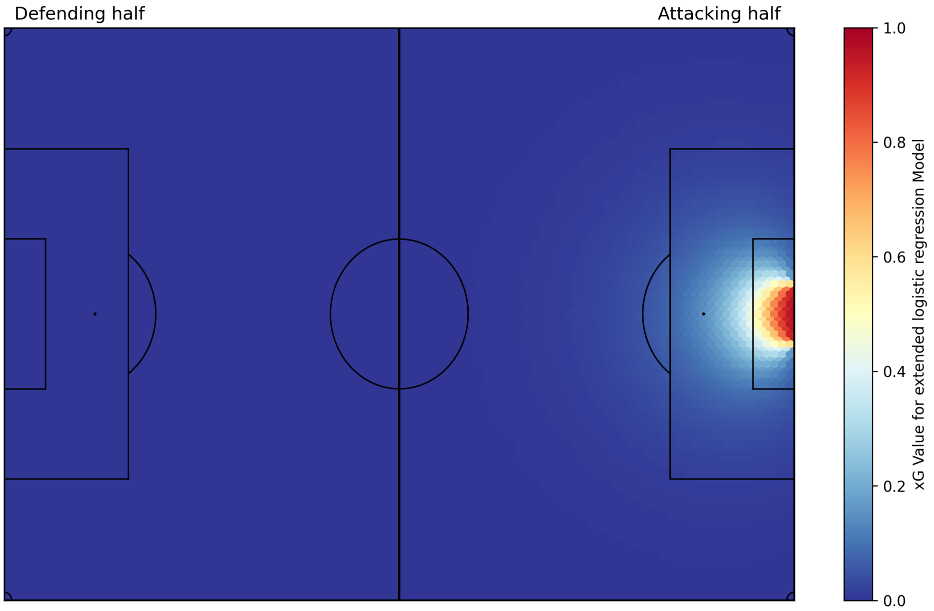

Figure 3.1: **Extended logistic regression -** A heatmap indicating goal probabilities for right footed shots from open play.

For applications inside this project however, an optimal xG model is desired, and so a variation of the extended model will be used to maximise performance. The hyperparameters of the tuned extended models are shown in *Table 3.2*

Model (Extended)	**Hyperparameters**
Logistic Regression	{ ’C’: 10, ’fit_intercept’: True, ’solver’: ’saga’ }
Random Forest	{ ’max_depth’: 10, ’min_samples_leaf’: 4, ’min_samples_split’: 10, ’n_estimators’: 300 }

Table 3.2: Hyperparameters for extended models based on Brier score based tuning.

Given the possible combinations of situations and shot types for these extended models, it is

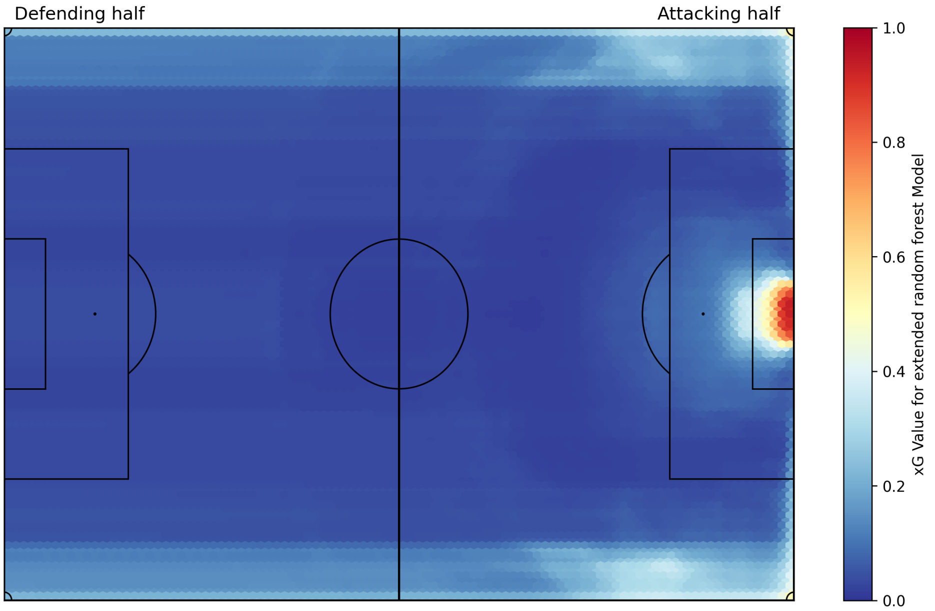

Figure 3.2: **Extended Random Forest -** A heatmap indicating goal probabilities for right footed shots from open play.

not possible to visualise all xG heatmaps. *Figures 3.1* & *3.2* plot an example combination, where shot type is 'Right Foot' and situation is 'Open Play'. Despite their similarities in terms of quantitative performance, qualitative evaluation of their heatmaps show significant underlying differences between the models. The decrease in xG probabilities for the logistic regression model follows a smooth concentric circle around the goal mouth. Meanwhile, the random forest displayed significant hotspots of higher xG, particularly in wing-related areas, appearing to be areas of overfitting. Whilst not fully conclusive, some investigation into the dataset indicates this may be due to the way shot values are captured. Shots in close proximity to either touchline are infrequent. However, a mis-hit cross or a free kick that results in a goal would count as a shot from this area. Had these not resulted in goals, it is unlikely they would've been registered as shots. Subsequently, a high proportion of 'shots' from this area are scored, and this is contributing to the overfitting seen within the random forest model. It is probable that this is superficially boosting the Brier score of the random forest. As such, consideration of anomalous values seemed a potential counteractive measure to this.

Exploration for anomalous value removal was done by examining the results of the nearest n neighbouring shots, and removing shots that fell below the similarity threshold. This was done for values of $n = 2, 5$ & 8, focusing primarily on tackling the overfitting of the random forest model. On the test set data, these models produced significantly improved Brier scores, as low as 0.124. However, as this data excluded anomalous shots, this is expected. For equal comparison, these models needed to be compared with the original models on a holdout set.

3.2.2 Holdout Set Performance

The holdout set used shot data from matches from the 5 leagues previously discussed, whose games occurred between 30/01/24 - 09/04/24. This gives all models an equal dataset to be compared against, with no potentially anomalous results removed. Metrical performance for

Extended (additional context) Models

Model	Neighbors Threshold	Brier Score	MAE of xG Value	RMSE of xG Value
Logistic Regression	**-**	0.078	0.160	0.280
Random Forest	**-**	0.078	0.160	0.280
Random Forest	**2**	0.078	0.160	0.280
Random Forest	**5**	0.078	0.153	0.280
Random Forest	**8**	0.078	0.145	0.280

Table 3.3: Extended model performance metrics (3sf) on holdout set (lower is better).

these models on the holdout set is shown in *Table 3.3.* The performance here is once again incredibly similar across the board. Whilst the 'neighbours' models did make smaller errors on average, maintenance of 0.78 in the Brier score implies a greater distribution when it comes to significant errors. Inspection of their heatmaps does indicate marginal, but imperfect, improvements in the identified wing-hotspots. For this reason, despite offering lower absolute error values on average, concerns over error distribution between points in these random forest models mean they will not be considered. A possible ensembling of models was also explored, but this yielded no performance benefit, at the cost of extra computation time. Hence, this was also dismissed.

3.2.3 Model Selection

Given the similarities in performance for the extended logistic regression and random forest models (with no neighbours removed) the decision was taken to use the logistic regression model for further exploration. Concerns over the overfitting of the random forest model to discrepancies in the method of data capture created uncertainty. It cannot be certain it would generalise as appropriately to other unseen data from a different source. Meanwhile, the combination of quantitative and qualitative evaluation gives confidence that the logistic regression model is the most well-rounded model across all areas of the pitch. Additionally, it offers improved interpretability over the random forest models. As such, making sense of feature importances will make it easier to uncover additional tactical insights.

Feature Importances

The feature importance values from the model (seen in *Figure 3.3*) indicate the change in the log-odds of a shot ending in a goal from a one-unit increase to the feature. The model places a vast emphasis on working shots into close-range, central positions. The feature importances of `angle_of_shot` (0.451) and `avg_distance_to_goal` (-1.010) exemplify this. This indicates that greater shot angles improve the odds of a goal, but greater distances to goal drastically hinder goalscoring likelihood. Tactically, this indicates the need for teams to focus on patient build-up play, creating an opening in these dangerous spaces. Moreover, `situation_DirectFreeKick` had the second strongest positive feature importance at 0.193. The potential gains here highlight the importance of exploring proper free-kick routines as a part of training. This can be considered in tandem with `shot_type_Head` which had the second highest negative feature importance (-0.288). Based on these considerations, it is likely to be

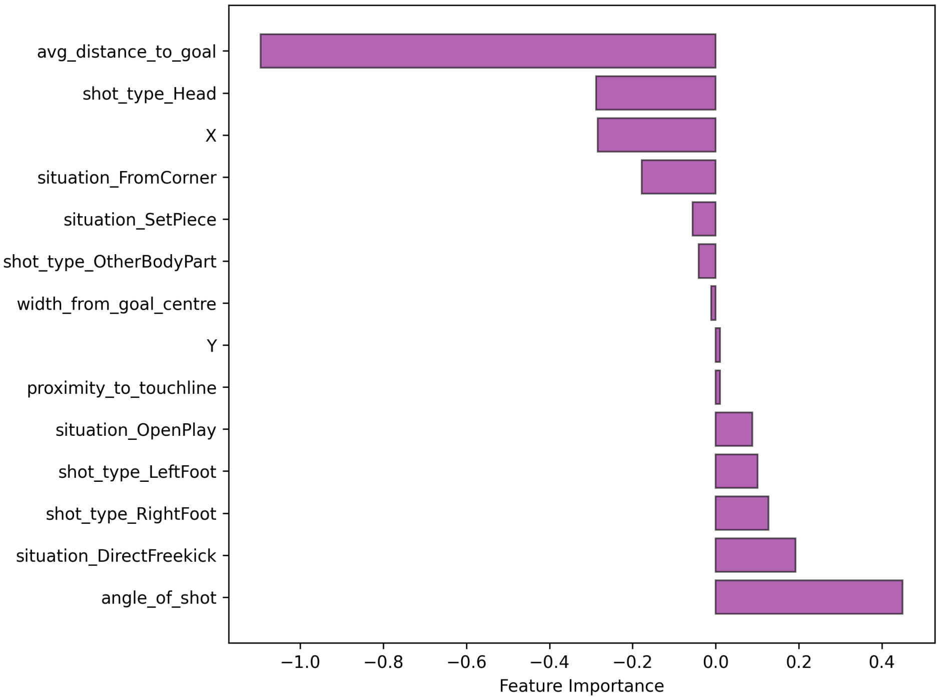

Figure 3.3: Comparion of feature importance values for the extended logistic regression xG model.

beneficial to focus free-kick routines around shooting opportunities, as opposed to crossing into the box for a headed opportunity. It is also worth noting, `situation_penalty` events were ultimately removed from the dataset. Expected goal values for this were calculated independently based on over 5400 penalties, producing an xG value of 0.78. The circumstances surrounding these mean they hugely correlated with goals. Understanding the impact of feature dynamics on goal likelihood is useful, but it's also important to understand the overarching correlation between shots and xG.

The Pearson Correlation Coefficient provides a quantitative evaluation over the correlation between xG and goals. The correlation coefficient between the two was found to be 0.370. The null hypothesis states that there is no correlation between the two, with a p-value threshold set at 0.05. The p-value for this correlation coefficient was found to be 0.0. As $0.0 \leq 0.05$, the null hypothesis can be rejected. As such, the moderately positive correlation between xG and goals is statistically significant. Given the scarcity of goals that undermines football, this highlights the value of xG in quantifying goalscoring opportunities. This creates a solid foundation to examine the capability of xG in analysing the results of previous games.

3.3 Monte Carlo Simulation for Previous Result Analysis

Given xG is a probabilistic outcome, there is always going to be variance in actual shot outcomes. Sometimes unlikely shots are scored, and sometimes likely shots are missed. Monte Carlo simulation provides an avenue to simulate all shot outcomes in a game. This accumulates to a simulated result. Simulated many times, 100 for this scenario, match outcomes should have a distribution converging to xG sums for the game. Doing this provides a total for the number of wins (worth 3 points), draws (worth 1 point) and losses (worth 0 points) in the 100 simulated scenarios. From this, it is possible to calculate an 'expected points' (xPts) value:

$$\text{Expected Points} = \frac{(3 \times \text{Number of wins}) + (1 \times \text{Number of draws})}{100}$$

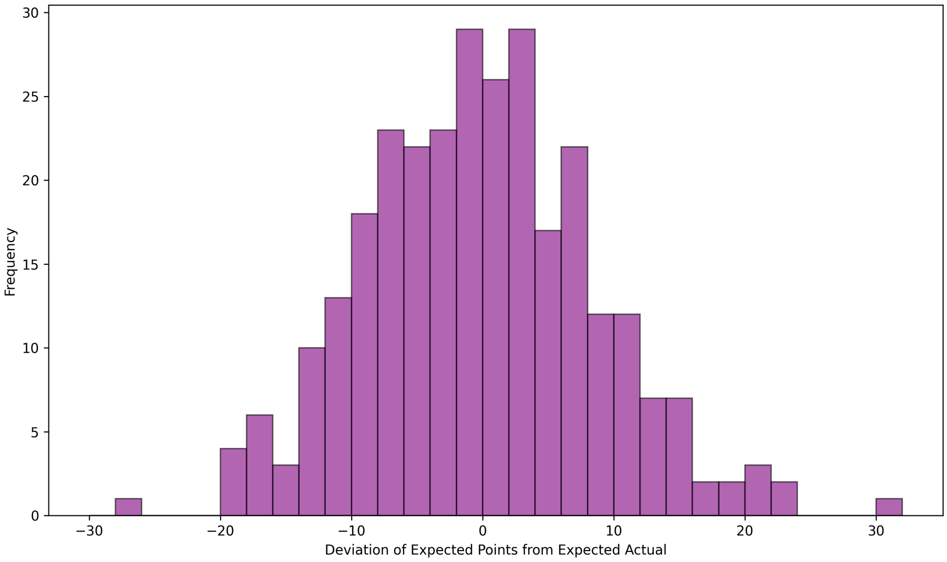

Figure 3.4: A histogram showing the distribution of the differences between actual points and xPts values.

For each season and league, a team's xPts values can be summed, to give an expected league table. This is a league table that quantifies teams in terms of the chances they created, allowing cross-team comparison for a full season. This works based entirely on the probabilistic xG values. Given the computational expensiveness of running the simulations, the project used 3 seasons of data (2016-17, 2018-19 and 2021-22) for all 5 leagues. This gave 15 seasons of expected point data to analyse, covering 294 different teams' seasons. *Figure 3.4* shows the distribution for the deviation of xPts values from points values for the 294 cases.

The distribution of point differences represents a roughly Gaussian distribution. Given the position of the central tendency, it is clear that, on average, the xPts a team accumulates based on the xG match chances closely reflects the actual points they score. This central tendency is also positive in reinforcing the stability of the model, with deviations not drastically skewed in either the positive or the negative direction. The Pearson Correlation Coefficient can once again be used to corroborate this. The correlation between points and xPts has a coefficient of 0.880. Given the infinitesimally small p-value (7.60×10^{-95}), the null hypothesis that there is no correlation between points and xPts can be rejected. Not only is this statistically significant, but the correlation is overwhelmingly positive. This indicates that the use of xG in analysing the performance of teams over a season is hugely reflective of actual results. It effectively captures the long-term quality of teams, the chances they create and their defensive capabilities. This enables quantitative performance-based analysis over result-based analysis to be conducted on an individual team-by-team basis. Correlative insights on these can pave the way for justification and implementation of tactical modifications.

Not only are xPts reflective on an individual team basis, but they are also hugely reflective of a team's performance comparative to the other teams in their division. The absolute average value of the deviation between a team's actual finishing position, and their finishing position in a table based only on xPts, was 2.44 positions. This is inclusive of a minority of extreme outliers. The quantification of overall xPts prediction effectiveness in predicting league positions is seen in *Table 3.4*.

Difference between Actual & Expected Finishing Position	Cumulative Proportion
±0	15.7%
±1	72.5%
±2	80.3%
±3	87.1%

Table 3.4: The cumulative proportion of finishing positions covered by different ranges of actual vs expected finishing position differences. Expected finishing positions are calculated based on each league and season's xPts table.

This effectively underlines the long-run comparative capabilities of using expected goals to develop a simulated league table. 213 of the 294 seasonal cases were within one position of their actual league position in the expected points table. This rose to 236 of the 294 cases within 2 places of their actual league position. The strength of the correlation between actual and expected finishing position (0.831) and significance of the correlation ($p = 3.10 \times 10^{-76}$) highlight its capability for long-term past performance analysis. With this, quantification of the respective over/under-performance of teams allows for nuanced comparative performance, even across different leagues and seasons. Understanding deviations here helps to portray the overarching sustainability of a team's performance. Observing the trends in this space can also allow for the identification of performance influencing factors for a single team. This might include factors such as coaching and player quality. All in all, the use of a probabilistic goals model as the foundation for analysing past game performance is hugely beneficial. The provision of the expected points framework is robust, creating new opportunities for tactical, data-informed, decision making approaches, on the basis of previous performance patterns.

3.4 Predictive Outcome Modelling

3.4.1 Baseline Metric-based Model

The predictive outcome modelling aimed to explore the predictive capabilities of past probabilistic performance (e.g. xG and xPts) in improving models designed to predict match outcomes. Given a model can only use information known prior to the game, the feature-space was primarily made up of averages and (6 game) rolling averages. The features are shown in *Table 3.5*.

As discussed in the literature review, due to this classification style of approach, the models used traditional classification metrics, including accuracy, precision and recall. It was anticipated that a logistic regression model would again be pursued. In the end, random forest models were also developed as a point of performance reference against the logistic regression models when using probabilistic inputs (*Section 3.4.2*).

Features		Targets
team_goals_avg	team_conceded_avg	win_notwin
shots_taken_avg	shots_conceded_avg	or
points_avg	team_goals_avg_6	win_draw_loss
team_conceded_avg_6	shots_taken_avg_6	
shots_conceded_avg_6	points_avg_6	
opponent_goals_avg	opponent_conceded_avg	
opponent_shots_taken_avg	opponent_shots_conceded_avg	
opponent_points_avg	opponent_goals_avg_6	
opponent_conceded_avg_6	opponent_shots_taken_avg_6	
opponent_shots_conceded_avg_6	opponent_points_avg_6	
goals_avg_diff	conceded_avg_diff	
shots_taken_avg_diff	shots_conceded_avg_diff	
points_avg_diff	goals_avg_diff_6	
conceded_avg_diff_6	shots_taken_avg_diff_6	
shots_conceded_avg_diff_6	points_avg_diff_6	
is_home	is_away	

Table 3.5: The features and possible target variables for the baseline predictive model.

Win-Draw-Loss Classification

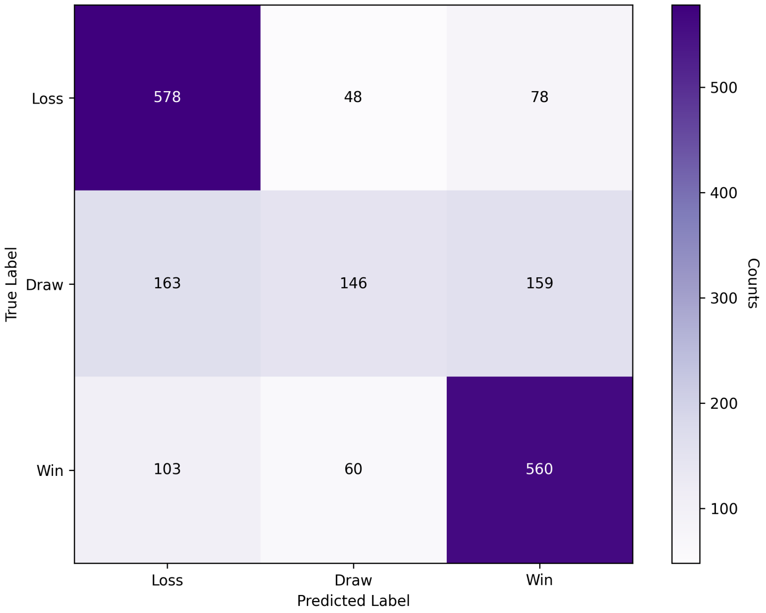

Figure 3.5: Confusion matrix for the baseline W-D-L predictive model.

Table 3.6 shows the classification metrics for the win-draw-loss classification approach using this baseline model. The model scores approximately 0.67 across all of accuracy, precision and recall. This indicates a relative level of success in classification approach, with double the performance of randomly guessing. Subsequently, the similarity in values indicates a balanced performance in the classification of true and false positives and negatives. However, identification of the confusion matrix (*Figure 3.5*) can allow for the identification of performance with respect of specific classes.

The confusion matrix paints a clearer picture of the relative strengths and weaknesses of the

Baseline Win-Draw-Loss Predictive Model			
Accuracy	*Precision*	*Recall*	*F1 Score*
0.6776	0.6645	0.6776	0.6584

Table 3.6: Classification metrics for the baseline Win-Draw-Loss predictive model.

model. It performs well at correctly predicting the outcomes of wins and losses, but struggles at correctly predicting draws. Only 31% of draws were predicted correctly, vs 82% and 78% of losses and wins respectively. This is again the product of the scarcity of goals in the game, where margins between drawing a game are often incredibly tight. As such, a model focused around predicting 'win' vs 'not-win' may provide better classification accuracy.

Win vs Not-Win Classification

Baseline Win vs Not-Win Predictive Model			
Accuracy	*Precision*	*Recall*	*F1 Score*
0.791	0.790	0.791	0.787

Table 3.7: Classification metrics for the baseline 'Win vs Not-Win' predictive model.

Table 3.7 indicates the performance increase of a binary classification approach. The model is able to correctly classify more correct cases (accuracy). It also shows improvement in the proportion of true wins out of predicted wins (precision) and actual wins (recall). Whilst model performance does improve, it loses granularity with regards to the inability to distinguish between draws and losses as match outcomes. As such, despite improved performance, it is not appropriate if prediction of all 3 possible match outcomes is a necessity. As such, it is important to explore whether the probabilistic inputs can improve both strains of model.

3.4.2 Model Including Probabilistic Inputs

Additional Probabilistic Input Features	
shots_taken_xG_avg	shots_conceded_xG_avg
xPts_avg	shots_taken_xG_avg_6
shots_conceded_xG_avg_6	xPts_avg_6
opponent_shots_taken_xG_avg	opponent_shots_conceded_xG_avg
opponent_xPts_avg	opponent_shots_taken_xG_avg_6
opponent_shots_conceded_xG_avg_6	opponent_xPts_avg_6
shots_taken_xG_avg_diff	shots_conceded_xG_avg_diff
xPts_avg_diff	shots_taken_xG_avg_diff_6
shots_conceded_xG_avg_diff_6	xPts_avg_diff_6

Table 3.8: Additional probabilistic features added to the baseline model.

Table 3.8 indicates the additional features used to develop the models based on the availability of probabilistic input features. Respective model performance is shown in *Table 3.9.*
This helps to observe that opting for a logistic regression model was a sound approach, as differences in performance were not substantial between model types. However, more interestingly, the classification metrics highlight a lack of classification performance, and sometimes a regression in performance, across the board following the introduction of these

Performance Metrics for Models Including Probabilistic Features

Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression (Win-Draw-Loss)	0.674	0.661	0.674	0.656
Random Forest (Win-Draw-Loss)	0.645	0.612	0.645	0.600
Logistic Regression (Win vs Not Win)	0.784	0.782	0.784	0.781
Random Forest (Win vs Not Win)	0.788	0.785	0.788	0.784

Table 3.9: Performance metrics (3sf) for models using probabilistic features.

probabilistic inputs. Given the success of using probabilistic values in conducting previous game analysis, this came as a surprise. The feature importances of the binary logistic regression (*Figure 3.6*) classifier provides an insight into the reasoning behind this.

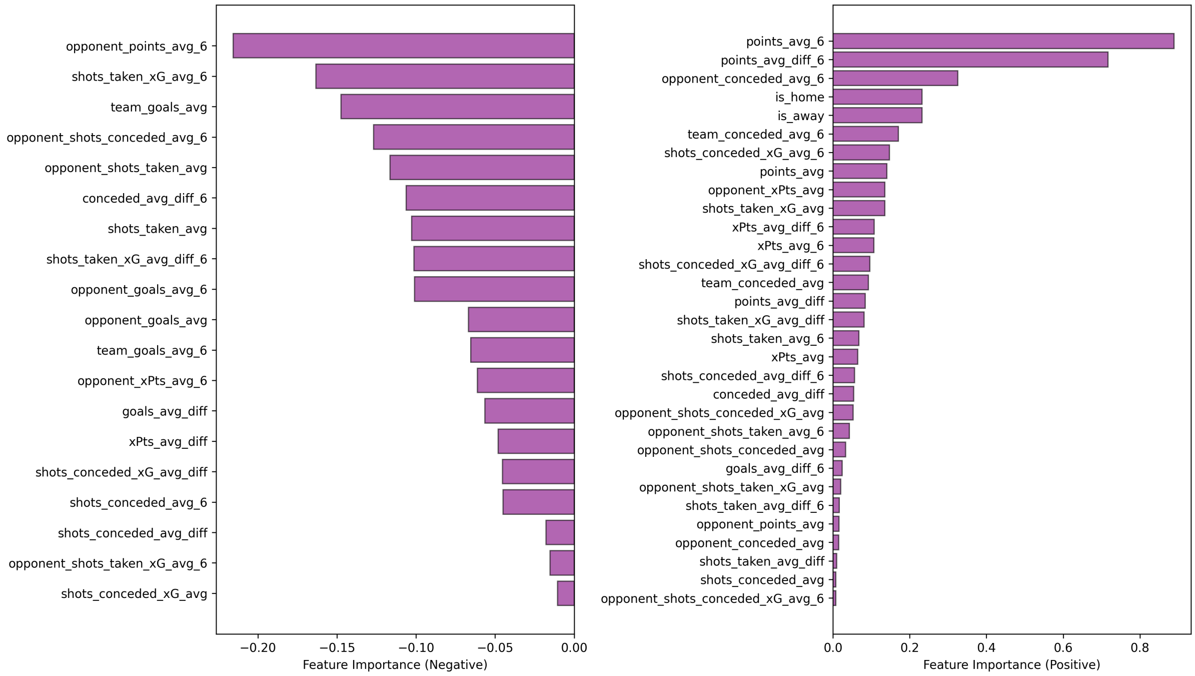

Figure 3.6: The strength of the positive and negative feature importances (including probabilistic values) on 'win' vs 'not win' classification.

Here, the value of and differences between the rolling average of points scored is by far the strongest predictive influence for match wins. Whereas, expected points rolling averages hold approximately 8x less weight in classifying a winning result. This is significant in highlighting a key issue concerning these probabilistic metrics. Whilst effective at analysing past results, they do not have the capacity to effectively capture contextual influences on match outcomes. Raw performances and results can significantly alter a team's mentality and overall momentum, which is not quantifiable using probabilistic values. The same can be said for unpredictable aspects like injuries. These undermine the inherent challenge that makes match prediction a difficult task. Nonetheless, this is significant in proving that probabilistic metrics are no better at capturing contextual indicators of future team performance than raw performance metrics, due to contextual unpredictability. This is not the same case when evaluating previous games, as this context is indirectly captured and considered within the traditional metrics.

Chapter 4

Discussion

4.1 Conclusions

4.1.1 Key Findings

The project successfully validated the suitability of machine learning models for analytical and predictive purposes, through exploration of the derived requirements. The initial use of machine learning models to devise goal predictions was effective. A statistical significance was found in the positive correlations between these predicted values and goals. Even amongst purely coordinate based models, expected goal values still provided valuable, albeit minorly sub-optimal, performance. Following a combination of quantitative and qualitative analysis, it was decided a logistic regression style of model was the most balanced model for wider application. This was in part due to its overall smoothness. It also did not overfit to potential discrepant data values, which may not be present in other datasets, which also a driving factor behind this. This meant the logistic regression model was more likely applicable to a wider source of data points. Finally, pursuing this style of model improved overall interpretability, allowing additional tactical insights to be derived from feature importance interactions. These highlighted the benefits of patient, build-up based approach play, to optimise chances with higher goal likelihoods, as well as exploring the benefit of data-informed approaches to set-piece routines.

The use of Monte Carlo simulation to simulate the results of previous matches and deduce the number of points expected from a game was hugely effective. The correlations between points and xPts, as well as actual and expected league position, were overwhelmingly positive, and statistically significant. The absolute average deviation between a teams expected and actual finishing position was only 2.44 places. When considering the absolute average deviation between points and xPts, a similar observation was made, at 6.95 points across a season. This equates to an error rate of $0.18 - 0.2$ xPts per game, dependent on the number of teams in the league. This highlights the overwhelming effectiveness of using probabilistic models in order to provide insights on past performances, on both a short and long-term basis.

Despite the effectiveness of these probabilistic inputs in analysing past events, there was a lack of improvement in predictive models when using probabilistic inputs. This indicates they are more well suited to past analysis than future predictions. This ultimately boils down to their inability to capture new information on the wider contextual landscape over existing metrics, which is challenging to capture full stop. Subsequently, past analytical predictive modelling approaches are not indicative of future raw or probabilistic performance. This is in part due to the difficulty to add contextual value with regards to aspects like momentum, morale and team availability. As such, in order to improve these predictive models, the priority should be on expanding contextual data capture opportunities.

4.1.2 Limitations & Implications

It is also important to consider the wider drawbacks of all facets of the software developed. All other processes were built on top of the expected goals model, and so are reliant on well-tuned model outputs. Whilst model outputs do provide low error rates on the whole, the limited feature-space, driven by data availability, means model outputs are not fully optimised. Additionally, the failure to consider aspects like own goals reduces data availability for the Monte Carlo simulations, potentially producing some discrepant results. Even though probabilistic inputs didn't improve predictive outcome models, their design was also prone to an impossible predictive outcome, despite improved results over random guessing. Matches were predicted twice, from the perspective of each team vs their opponent. This meant that, for a small minority of fixtures, it was predicted that both teams would win or both teams would lose. Ultimately, engineering a method where each match is predicted once, giving an outcome for each team, would help to avoid this impossible prediction case, even if infrequent.

Nevertheless, the overall findings of the project underline the effectiveness of using xG as a performance measure for past match events. This includes on both individual shots, and wider seasonal performance. Embracing a machine learning based, data-informed approach allows for a quantitative understanding of performance, for both entire teams and individual players. A broad scale adoption of these analytical approaches increases explainability in a sport riddled with unpredictability. In doing so, this provides the groundwork for wide-scale benefits from data-informed tactical approaches, on a team-by-team, game-by-game basis.

4.2 Ideas for future work

The scope of this project also opens the door for numerous future work opportunities. Having highlighted the lack of accessibility to xG data in lower divisions, it is worth considering how models could be opened up to smaller clubs. Development of simple xG plotting software could be used to allow anybody to track xG for a given match, even down to a grassroots level. This also lends itself to exploration over the extent to which changes in quality, of both goalkeepers and finishers, may alter goal probabilities moving down the football pyramid. Moreover, ongoing developments in data capture technology will extend the feature space of existing models, enabling further fine-tuning of models. This also presents the opportunity for an exploration of the benefits of dynamic models dependent on specific players or match states, which could provide more refined, individualised outputs. Aside of development of xG models themselves, there is scope to explore other strands of predictive model. Given the growing demand of the sport, with ever-increasing game frequencies, the development of a predictive injury model could be used to quantify and balance player workloads. Similar implementation approaches could also be explored for optimising the scouting processes, which would likely lend itself to data-driven approaches. Finally, it is also worth considering the role of machine learning football models on the sports betting market. This would necessitate an exploration of the tuning capabilities and predictive implementations of machine-learning models in real-time. This provides an avenue to explore how bookmakers can optimise their odds-provision process to maximise profits. The sheer range of these future work opportunities highlights the wide-scale implications of applying machine learning approaches in a footballing context.

References

[1] V. Anand. FPL Historical Dataset. https://github.com/vaastav/Fantasy-Premier-League/, January 2024.

[2] C. Anderson and D. Sally. *The Numbers Game: Why Everything You Know About Football is Wrong*. Penguin Books, 2013.

[3] G. Anzer and P. Bauer. A goal scoring probability model for shots based on synchronized positional and event data in football (soccer). *Frontiers in Sports and Active Living*, 3:624475, 2021.

[4] A. Botchkarev. Performance metrics (error measures) in machine learning regression, forecasting and prognostics: Properties and typology. *arXiv preprint arXiv:1809.03006*, 2018.

[5] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

[6] D. Carrion. Scraping understat dataset. https://github.com/douglasbc/scraping-understat-dataset.git, January 2024. GitHub repository.

[7] Deloitte Sports Business Group. *Deliotte Football Money League 2024*. Deloitte UK, 2024.

[8] A. Didavi, R. Agbokpanzo, and M. Agbomahena. Comparative study of decision tree, random forest and xgboost performance in forecasting the power output of a photovoltaic system. In *2021 4th International Conference on Bio-Engineering for Smart Technologies (BioSMART)*, pages 1–5. IEEE, December 2021.

[9] J. Dunn, L. Mingardi, and Y. D. Zhuo. Comparing interpretability and explainability for feature selection. *arXiv preprint arXiv:2105.05328*, 2021.

[10] P. Gamble, L. Chia, and S. Allen. The illogic of being data-driven: reasserting control and restoring balance in our relationship with data and technology in football. *Science and Medicine in Football*, 4(4):338–340, 2020.

[11] I. Graham. *Routledge Handbook of Elite Sport Performance*, chapter Using Data in Football. Routledge, 1st edition, 2019.

[12] G. Hackeling. *Mastering Machine Learning with scikit-learn*, pages 15–19. Packt Publishing Ltd., 2017.

[13] IFAB. *Laws of the Game*, chapter The Field of Play. International Football Association Board, 2023/2024 edition, 2023.

[14] Jupyter. Jupyter notebook documentation. `https://jupyter-notebook.readthedocs.io/en/latest/notebook.html`, March 2024.

[15] R. Lawson. *Web scraping with Python.* Packt Publishing Ltd, 2015.

[16] P. Linardatos, V. Papastefanopoulos, and S. Kotsiantis. Explainable ai: A review of machine learning interpretability methods. *Entropy*, 23(1):18, 2020.

[17] W. McKinney. *Python for data analysis: Data wrangling with Pandas, NumPy, and IPython*, chapter 5. " O'Reilly Media, Inc.", 2012.

[18] J. Molokach. Law of cosines—a proof without words. *The American Mathematical Monthly*, 121(8):722–722, 2014.

[19] K. Morin and J. L. Davis. Cross-validation: What is it and how is it used in regression? *Communications in Statistics-Theory and Methods*, 46(11):5238–5251, 2017.

[20] Nielsen Sports. *World Football Report 2018*, chapter The Fans. The Nielsen Company, 2018.

[21] Nielsen Sports. *World Football Report 2018*, chapter The Media & The Sponsors. The Nielsen Company, 2018.

[22] R. Pollard and C. Reep. Skill and chance in association football. *Journal of Royal Statistical Society Series A (General)*, 131(4):581–585, 1968.

[23] R. Pollard and C. Reep. Measuring the effectiveness of playing strategies at soccer. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 46(4):541–550, 1997.

[24] F. Python. Drawing a pass map in python. `https://fcpython.com/visualisation/drawing-pass-map-python`, January 2024.

[25] S. Raschka. *Python Machine Learning*, pages 13–15. Packt Publishing Ltd., 2015.

[26] S. Raychaudhuri. Introduction to monte carlo simulation. In *2008 Winter simulation conference*, pages 91–100. IEEE, 2008.

[27] P. Rodríguez, B. Humphreys, and R. Simmons. *The Economics of Sports Betting.* Edward Elgar Publishing, 2017.

[28] K. Rufibach. Use of brier score to assess binary predictions. *Journal of clinical epidemiology*, 63(8):938–939, 2010.

[29] E. Schulze, B. Mendes, N. Maurício, B. Furtado, N. Cesário, S. Carriço, and T. Meyer. Effects of positional variables on shooting outcome in elite football. *Science and Medicine in Football*, 2(2):93–100, 2018.

[30] S. Seabold and J. Perktold. Statsmodels: econometric and statistical modeling with python. *SciPy*, 7:1, 2010.

[31] P. Sedgwick. Pearson's correlation coefficient. *Bmj*, 345, 2012.

[32] Sofascore. Sofascore: Football - scores, schedule & odds. `https://www.sofascore.com/`, January 2024.

[33] D. Sumpter. *Soccermatics: Mathematical Adventures in the Beautiful Game*. Bloomsbury Publishing, 2016.

[34] A. Takvorian. *The Beautiful (Computer) Game: How Data Science will Revolutuionize the World's Most Popular Sport*. PhD thesis, The University of Texas at Austin, 2021.

[35] A. Tharwat. Classification assessment methods. *Applied computing and informatics*, 17(1):168–192, 2020.

[36] Understat. Understat: Stats for teams and players from the top european leagues. `https://www.Understat.com/`, January 2024.

[37] I. Vasilev, D. Slater, G. Spacagna, P. Roelants, and V. Zocca. *Python Deep Learning: Exploring deep learning techniques and neural network architectures with PyTorch, Keras, and TensorFlow*, pages 29–83. Packt Publishing Ltd., 2019.

[38] F. Vidal-Codina, N. Evans, B. E. Fakir, and J. Billingham. Automatic event detection in football using tracking data. *Sports Engineering*, 25(18), 2022.

[39] C. Wade and K. Glynn. *Hands-On Gradient Boosting with XGBoost and scikit-learn: Perform accessible machine learning and extreme gradient boosting with Python*, pages 3–5. Packt Publishing Ltd., 2020.

Appendix A

Self-appraisal

A.1 Critical self-evaluation

The development of my project allowed for a thorough, hands-on experience with data analytics and machine learning practices. This covered the development process of a machine learning model from the initial data gathering and cleansing phases, all the way through to the evaluation and application of model outputs. This accumulated in the development of a well-performing expected goals model, with the applicability for analysing previous results being particularly encouraging. Even though the use of probabilistic inputs was not successful in improving the capability of predictive models, it ultimately encouraged valuable insights into why there was a discrepancy in analytical vs predictive power on wider matches. Additionally, the overarching transparency provided in model feature importances also helps to drive tactical decision making processes. When used in combination with the outputs of these models, the opportunity for nuanced, individualised tactical reflection and exploration is overwhelming.

A.2 Personal reflection and lessons learned

The overall development process helped me to hone and develop a number of different skills, both in and outside of a technical domain.

Embracing the development of an end-to-end project allowed for significant opportunities to hone my skills with already familiar libraries. This inherently supplemented my development experience from modules such as Info Vis, Data Science and Machine Learning, which also made use of libraries including matplotlib, pandas and scikit-learn. Lots of my code development took a trial-and-error approach to exploring how these libraries best fitted together. Upon reflection, whilst this was effective in giving me hands-on experience, the literature review for these libraries should've been conducted before any stages of code development. This would've allowed for a more nuanced understanding of library interoperability and overall capabilities, improving the efficiency of project development. Nevertheless, the experience from this built upon my wider approach to project management. This ultimately ensured a more structured approach towards the end of the programming lifecycle, and during the report writing phase of the project.

Also with regards to project management, the use of GitHub was essential in managing the code-development process. Numerous occasions arose where experimentation had broken sections of code, or rendered them no longer useful. The frequency and relevance of commits provided a solid foundation for reversion to a sound project implementation from unstable experimental processes. Additionally, using GitHub for version management allowed for cross-platform development, enabling me to work from both my Desktop PC and Laptop, and keep up to date with code-base edits on the other device. As such, understanding how to use GitHub effectively was key for me in managing the software development cycle, and in providing confidence for exploratory development. This has since standardised my use of GitHub, for personal projects and even small coursework pieces.

As referenced through the exploratory approach, development of the project did come with its challenges. Hyperparameter tuning initially focused on error metrics between the shot classification and actual classification, which failed to ultimately evaluate the error between the actual model output, being the predicted value. This then necessitated the use of a custom hyperparameter tuning function. Implementation for this went relatively smoothly, but was incredibly computationally expensive. Subsequently, this led me to explore the use of parallel frameworks to speed up computation time. Whilst challenging, with a steep learning curve, this was a hugely effective way of getting to grips with parallel methodologies. Doing so made clear the opportunities for performance benefit through the consideration of parallel programming techniques. Additionally, investigation into the discrepancies between the random forest and logistic regression xG heatmaps was a tough, time-consuming process. This required manual evaluation of specific shot points, player positions and scenarios to identify why the models were behaving in slightly different ways, in spite of the hyperparameter tuning. Whilst a frustrating process, it did underline the importance of ensuring a full and comprehensive understanding of the data, as well as the quality of the data source.

While all of these aspects of the project were significant in developing my project management and programming skills, this report-writing process was perhaps where the most value came. Understanding how to fully formulate the development process into well-structured, logical report was difficult. On reflection, a greater proportion of the report-writing process should've been done in tandem with code development. This would have provided a stronger foundation to develop code upon, and enabled the tuning of written sections whilst the code implementation was still fresh and familiar from development.

Ultimately, the opportunity to apply my degree to sports analytics, a field I am greatly passionate about, has been invaluable. Despite highs and lows in project development, pursing something I am specifically engaged in has kept me eager to continue work on the project. This has included taking foundational steps to explore some of the ideas laid out in the 'Ideas for Future Work', with plans to fully build-upon this in the summer. The learning process of this project has not only been invaluable from a programming perspective, but also in understanding how to effectively plan and manage a large-scale project.

A.3 Legal, social, ethical and professional issues

A.3.1 Legal issues

Given the use of data, it is was important for me to consider the regulations laid out by legislation such as the data protection act. The data used in this project is widely-used and openly available, meaning the project ultimately had no concerns relating to the use of personal data.

A.3.2 Social issues

Given the discussion for the wide-ranging possibilities for these machine learning approaches, it is important to understand the way in which they are applied. Models used in evaluating player performance, or scouting players for a club, can have a significant impact on people's lives, namely the employment of the players. As such, it is imperative that the models developed are evaluated for overall fairness, ensuring no exacerbation of underlying biases are present in model outputs. This is an underlying consideration for references to model transparency throughout the project.

Additionally, the references to potential use-cases in the sports-betting industry raises another key social issue. Gambling addiction is not uncommon, and the financial consequences can be devastating and ruin livelihoods. It is absolutely essential to understand that predictive model outputs are just that, they are predictions. If they were to be used in a gambling context, they should only be used as additional context to facilitate responsible betting practices. The predictive models are based purely on historical data, and provide no certainty over the occurrence of future events. As predictive models like the ones developed in this project continue to become more mainstream, it is essential they are supplemented with the awareness and education related to the continued risks of gambling.

A.3.3 Ethical issues

The social issues described around gambling are also relevant here. It is important to raise awareness in the uncertainty around model outputs, as a result of the inherent unpredictability of the sport.

Aside from this, data was collected using web scraping techniques from Understat.com. Whilst this was not forbidden, it was imperative to respect the site's server load during the heavy data retrieval process. As such, this retrieval process used rate limiting to restrict the frequency at which requests were made to the site. This helped ensure the data retrieval process did not contribute to destabilisation the site's servers.

Additionally, it's important to maintain transparency about the development process of these models. The project is clear in outlining the different datasets and their features used throughout implementation, such that if somebody were to adopt the model, they could comprehensively understand the process behind the model outputs.

A.3.4 Professional issues

As a machine learning/data science focused problem, it was essential to undergo full data exploration, cleansing and preprocessing steps. This ensured confidence in the data the models were built upon, enabling honest development of the best performing model possible. The capability of using model feature importances for tactical insights was also relevant here, as it further supported the interpretability of models and their decision making process. Finally, the application of contextually relevant evaluation metrics ensured that the models produced were to the highest possible standard. In doing so, this makes clear that the models were in no way designed to mislead, but rather to act as a support mechanism for conducting football analysis.

Appendix B

External Material

The data used was sourced from Understat.com [36]: `https://understat.com/`

The process used to retrieve data from the site was inspired based on a public GitHub repository, from user *douglasbc* [6]:
`https://github.com/douglasbc/scraping-understat-dataset`