WIRTSCHAFTSUNIVERSITÄT WIEN
Vienna University of Economics and Business

Bachelor's Thesis

Title of Thesis (English)	Do bookmaker betting odds and a result based form parameter improve the performance of complex machine learning models for football game prediction?
Title of Thesis (German)	Verbessern Buchmacherwettquoten sowie ein performance-basierter Formparameter die Leistung komplexer Machine Learning Modelle zur Vorhersage von Fußballspielen?
Author	Harald Körbel
Student ID	12208935
Degree program	Bachelor of Science in Economics and Social Sciences
Examiner	Dr. Lucas Kook & Dipl.-Ing. Robert Bajons, MSc

I hereby declare that:

1. I have written this Bachelor's thesis myself, independently and without the aid of unfair or unauthorized resources. Whenever content has been taken directly or indirectly from other sources, this has been indicated and the source referenced.

2. While writing this thesis, I used ChatGPT 5.2 for the purpose of debugging and coding assistance for the belonging R files, as well as helping with the creation of entries to the literature library of the corresponding LaTeX file. I carefully reviewed and edited the content as needed. I take full responsibility for the content of my thesis and understand that any unacknowledged but detected use of generative AI will be regarded as academic fraud.

3. This Bachelor's Thesis has not been previously presented as an examination paper in this or any other form in Austria or abroad.

4. This Bachelor's Thesis is identical with the thesis assessed by the examiner.

Vienna on January 14, 2026

Harald Körbel

Do bookmaker betting odds and a result based form parameter improve the performance of complex machine learning models for football game prediction?

Harald KÖRBEL
Vienna University of Economics and Business

ABSTRACT

Football predictions are not only an interesting possibility to evaluate the performance of complex machine learning models, but due to high betting volumes, they are economically relevant too. This project is trying to answer if the inclusion of bookmaker betting odds and a performance based form parameter improve the quality of complex machine learning models for the prediction of football games in the English Premier League. Therefore, extensive data from the 2017/18 to the 2024/25 season of the Premier League is leveraged to build XGBoost and random forest based models. The obtained models are evaluated based on the ranked probability score and accuracy. Furthermore, the predictions of the models are applied to two different betting strategies with the goal of a positive return over the course of one season, trying to beat the bookmakers. Both, betting odds as well as the utilized form parameter increase prediction performance, while a positive return over the course of the test season could be achieved, but with a 95% confidence interval stretching into negative returns on the lower bound.

CONTENTS

1. INTRODUCTION

Predicting the outcome of football games is not new to the world of data analytics. Many analysts have already contributed to the field, resulting in various approaches. Also varying between authors is the motivation behind the attempts of predicting football games. While some are in pursuit of building a model as accurate as possible, aiming at using their predictions for sports betting, trying to beat the bookmakers (Stuebinger et al., 2020), others try their best to understand the sport and its games better for analytical reasons (Verkerk, 2018). Another interest group in football analytics are the teams themselves. The advance of data analytics provides the teams with new opportunities to better understand what factors influence the outcome of their games (Yang, 2023), and therefore decide over their own success and failure.

Especially the occurrence of big international tournaments, like the UEFA Euros or the FIFA World Cup, attracts the interest of data analysts with their abilities in predicting the outcomes (Stuebinger et al., 2020). But not only these tournaments, national leagues and other yearly recurring competitions too, are subject to this field. In a comparison between different approaches predicting the UEFA Euros 2020 Groll et al. (2021) conclude that XGBoost is a very promising approach to model football games. According to the authors' analysis, the comparably small training data set for European Championships has deprived them of the possibility to leverage the full potential of XGBoost, suggesting it to be a method only really unfolding its true potential with more training data. Following this argumentation it seems natural to adopt XGBoost for league competitions, since they encompass more games due to their weekly schedule, resulting in more training data. Therefore, this thesis aims at unfolding the full potential of XGBoost and other complex machine learning algorithms by applying them to the English Premier League.

1.1. Related Work

With scored goals being the deciding variable in football, the game is dependent on discrete, non-negative count variables. Since such can be modeled by a Poisson distribution (Rue and Salvesen, 2001), there have been different approaches in the past, leveraging Poisson models to predict football games (Maher, 1982; Dixon and Coles, 1997; Rue and Salvesen, 2001; Karlis and Ntzoufras, 2003, 2005; Koopman and Lit, 2015; Robberechts and Davis, 2019; Tsokos et al., 2019; Groll et al., 2021, 2024). While Maher (1982) modeled the number of scored goals of two competing teams in a game as independent Poisson distributed variables, Dixon and Coles (1997) discovered the struggle of such Poisson models to predict low scoring game results correctly, leading them to the introduction of an additional parameter to adjust the probability of such results, an approach also adopted by Rue and Salvesen (2001). Karlis and Ntzoufras (2003) further improved the Poisson based method, arguing the goals scored in a game by two competing teams are not independent from one another, therefore utilizing a bivariate Poisson model, an strategy leveraged too by Robberechts and Davis (2019). They further improved their work by applying diagonal inflated bivariate Poisson regression models (Karlis and Ntzoufras, 2005), to account for the underrepresentation of draws experienced by previous models. Koopman and Lit (2015) developed a dynamic bivariate Poisson model utilizing time variant intensity coefficients, allowing teams' attacking and defending abilities to change with time. Tsokos et al.

(2019) modeled football games using a hierarchical Poisson log-linear model, before Groll et al. (2021, 2024) started to work with hybrid models for predictions of the 2020 and 2024 European Championship, combining Poisson distributions with machine learning methods.

In the last years, complex machine learning models have been leveraged to build football prediction models (Constantinou, 2019; Berrar et al., 2019; Hubáček et al., 2019; Baboota and Kaur, 2019; Stuebinger et al., 2020; Groll et al., 2021, 2024). Introducing a model called Dolores, Constantinou (2019) utilizes a combination of two subsystems. The first is a dynamic rating system, extending the pi-rating system (Constantinou and Fenton, 2012), and generating features, by learning from historic game data from 52 leagues from all over the world, which work as an input for the second subsystem, a Hybrid Bayesian Network, delivering probabilities for the three possible game outcomes (home win, draw, away win). Arguing that the integration of domain knowledge and right preparation of data is more important for success than the choice of algorithm, Berrar et al. (2019) developed a competition winning model using k-nearest neighbor learning to classify game outcomes, by searching for the k most similar game constellations in their historic data set. After winning the *2017 Soccer Prediction Challenge* with their model, Berrar et al. (2019) experimented further, ultimately developing an even better model leveraging XGBoost.

Hubáček et al. (2019) analyzed multiple techniques. Among them the, on Relational Dependency Networks function approximators based, RDN-Boost algorithm. Using this approach leverages the objects (e.g., teams and leagues) as well as their relationships to each other (e.g., teams being part of a league or games between teams) together with a set of relational facts to deliver predictions. These facts include game results, scores and affiliation of games to a certain team and league. To incorporate the current and historic strength of the teams, Hubáček et al. (2019) included pi-ratings. Since RDN-Boost is a binary classifier, three separate models, one for each of the three possible game outcome classes, were trained and the resulting probabilities normalized. A second method in use by Hubáček et al. (2019) is a feature-based classification model, based on XGBoost, directly delivering a probability distribution for the possible game outcomes without the need for additional post-processing. This model being the one performing best in the respective paper further strengthens the believe in the XGBoost approach of this project. Other contributions leveraging XGBoost classification include Baboota and Kaur (2019); Groll et al. (2021, 2024). Thirdly, arguing that the standard classification from their second model does not consider the ordinal scale of game outcomes ($win > draw > loss$), Hubáček et al. (2019) implement a feature-based XGBoost regression model. To do so, they assign the classes numerical values ($win = 1,\ draw = 0.5,\ loss = 0$). Because the regression model only outputs one value r, the probability of each game outcome is additionally modeled as a function of the regression output. In the end the three obtained probabilities are normalized to retrieve the final predictions.

With a Gaussian naive Bayes, a purely probabilistic algorithm based on conditional probability, Baboota and Kaur (2019) utilize another ML technology, next to support vector machines (SVMs). Them being a collection of supervised learning algorithms, the authors argue that, when applied to a high-dimensional feature space, SVMs are highly effective. Utilizing the linear and the radial basis kernel, which are functions to identify the best fitting decision boundaries, Baboota and Kaur (2019) computed probabilities for the different classes leveraging five-fold cross-validation. Another method employed by Baboota and Kaur (2019); Groll et al. (2021, 2024) is the random forest. It is able to handle the problem of high variance experienced by single decision trees, by growing a large number of decision trees and trusting on the majority vote, as well as its ability to map non-linear decision boundaries, the authors emphasize the random forest as a robust machine learning algorithm.

A combined approach was applied by Stuebinger et al. (2020), where the authors utilize explanatory variables using player characteristics from domains including ball skills, passing and shooting, to predict the game outcome via the goal difference between the home and the away team. With boosting, support vector machines, a linear regression, a random forest and an additional combination of the four above technologies, in total Stuebinger et al. (2020) tried five different modeling methods. The least absolute shrinkage and selection operator (LASSO) method is another methodology applied by Groll et al. (2024) to predict game outcomes via determining the scored goals per team in dependence on various features. The LASSO model distinguishes itself as a powerful ML tool mainly through two properties. Firstly, it is capable of setting the coefficients of less relevant explanatory variables to zero, enabling good feature selection and learning which covariates have a significant influence on the target variable during training. Secondly, by adding a penalty term, the complexity of the model is controlled, keeping it from memorizing noise of the train data and improving prediction quality of unseen data instead, meaning LASSO is strong when it comes to regularization, avoiding overfitting (Groll et al., 2024).

1.2. The project's contributions

The project's main goal is predicting football games of the English Premier League by leveraging complex machine learning models. For this, three steps were followed: 1) feature extraction, 2) model choice and 3) evaluation. Since bookmakers are recurringly considered as a benchmark for prediction models (Robberechts and Davis, 2019; Groll et al., 2021, 2024), during step 1) the models of this work are trained in different variations, with one of the differences being the inclusion of bookmaker odds as additional explanatory variable only for some of the models, to evaluate their influence on prediction quality. Additionally, a team specific form parameter derived from past game outcomes, previously employed by Baboota and Kaur (2019), who suggest a value of 0.33 for the hyperparameter γ, which is necessary to create the desired form parameter, as the best value, is put to test together with the afore mentioned hyperparameter. Furthermore, a new approach to weighting the game statistics of past games as covariates is tested, by not weighting the values of each explanatory variable for a specified amount of past games with a strict rule like past contributions (Ley et al., 2019; Groll et al., 2021), but feeding the exact values of each explanatory variable from the last 5 games to the built models, to fully embrace the capabilities of complex machine learning algorithms and put to test, if they are capable of delivering good results without feature pre-selection, by training them with all 94 play-specifc variables available. With step 2) a comparison between random forest and XGBoost models is carried out, while step 3) is about the evaluation of the trained models with the ranked probability score and accuracy as chosen metrics. Moreover, to put the models to an economic test, two different betting strategies are applied, trying to beat a bookmaker and realize a positive return over the course of one season.

2. METHODOLOGY

As this thesis aims at clarifying if the inclusion of bookmaker's betting odds and a performance based form parameter have an enhancing effect on the prediction quality of complex machine learning algorithms when predicting football games in the English Premier League. To provide the reader with a better understanding of the methods in question, random forest and XGBoost, they are explained in the following section.

2.1. Random Forest

A random forest is a machine-learning method, which can be applied to regression and classification problems (Breiman, 2001). The models developed for this thesis classify the predicted outcome of football games as either home win, draw or away win. A forest for such a purpose is created by growing multiple decision trees, each one marginally different, and deciding on the class by majority vote of the trees. For regression the prediction would equal the average output of the trees (Liaw and Wiener, 2002a). Numerous trees are needed, because single decision trees can be unstable. Small changes in the data can lead to significant changes for a tree. Furthermore, a single tree is more likely to overfit. By creating an ensemble of trees, a single tree's weaknesses are dealt with.(Breiman, 2001).

To incorporate randomness, different approaches can be leveraged. To give a general understanding, bagging, random split selection and the random subspace method are shortly introduced.

Bagging does not include randomizing the tree algorithm itself, but the data each tree is trained on instead. For each tree, a bootstrap sample is drawn from the original dataset, meaning N training examples with replacement are selected, leading to some examples appearing multiple times and others not at all. A decision tree is then trained on the created bootstrapped sample. Following this technique, several similar, but not identical, datasets are generated, each producing a different tree (Breiman, 1996).

For random split selection, randomness is injected into the tree algorithm. For every node within a tree, the top K candidate splits are computed. Out of those K candidates, one is chosen at random, leading to the creation of different trees, as they are trained on the same data (Dietterich, 2000).

The random subspace method facilitates randomness by choosing arbitrary subsets of features, then called the random subspace. For each tree grown, such a random subspace of the available variables is selected and the training samples are projected into this reduced subset, building a complete decision tree. This mechanism is repeated independently for each tree. Following this approach, every tree knows different features, while the training data remains constant, letting the trees grow in different sections of the feature space, establishing dissimilar splits (Ho, 1998).

Leveraging the `randomForest` (Liaw and Wiener, 2002b) and `ranger` (Wright and Ziegler, 2017) packages in `R` (R Core Team, 2025), two variations of random forests were trained for this project. To inject randomness, the above described methods bagging and random subspace were facilitated. Bagging is utilized by both packages by default, as long as the parameter `replace` is not set to `FALSE` in the specifications when calling the functions `randomForest` (Liaw and Wiener, 2002b) and `ranger` (Wright and Ziegler, 2017) respectively. For both random forest variations, the random subspace is employed by specifying the parameter `mtry`, were a value of `max(1, floor(sqrt(p)))` was chosen, setting the number of features considered per split to the square root of the available explanatory variables p, while simultaneously ensuring `mtry` never is smaller than 1. For both model variations, the number of trees to grow was set to 500.

Following the Strong Law of Large Numbers, as more trees are appended, random forests do not overfit, an important characteristic. With infinite trees, a forest behaves like the expected value on all possible randomly created trees and variance vanishes, as proven by Breiman (2001).

Furthermore, random forests work well and accurate, because they create strong and uncorrelated trees. Strength representing the average margin of the trees, with margin calculated as the difference between the predicted probability of the right class and the predicted probability of the best wrong class, and correlation indicating how much the different trees tend to make the same mistakes, the stronger and more uncorrelated the trees, the more accurate the forest (Breiman, 2001).

Due to their ability to effectively capture complex, non-linear relationships between different features, tree-based ensemble methods, such as random forests, are well suited for modeling football outcomes, scoring competitive RPS-values compared to bookmakers (Baboota and Kaur, 2019). Additionally, numerical as well as categorical features can be included without any difficulties (Stuebinger et al., 2020) and, as argued above, overfitting is of little concern.

2.2. XGBoost

XGBoost is a scalable machine learning framework designed for tree boosting, with great impact on data mining and machine learning competitions, delivering state-of-the-art performances for various problems, including motion detection, web text classification and store sales prediction (Chen and Guestrin, 2016).

The method utilizes an ensemble of trees, like a random forest, but with the difference of no independence between the single trees. Instead, every tree aims at eliminating or at least reducing the errors the previously grown trees made, such that every new tree progressively improves the model marginally. Many small refinements ultimately lead to a good model (Chen and Guestrin, 2016). This makes XGBoost an iterative boosting algorithm, combining many weak learners into a strong ensemble in additive manner, resulting in high predictive accuracy (Groll et al., 2021).

The adaptive capabilities towards high-dimensional data together with the proficiency to incorporate variable selection in the fitting process count as great asset of statistical boosting algorithms (Groll et al., 2021). Furthermore, leveraging decision trees as learners marked the start of gradient tree boosting. Here, decision trees are recurrently fitted on the errors of the preceding fit, assembling them to a successive ensemble (Friedman, 2001). Further optimizing this approach by integrating supplementary regularization in the objective function, turning the single trees into weak learners, to prevent overfitting, the next tree is cumulatively incorporated into the ensemble after multiplication with a tendentially small learning rate, making the learners even weaker, the method had evolved to extreme gradient boosting, short XGBoost.

This is the popular method known today for great performance in diverse machine learning competitions hosted by organizations such as Kaggle (Chen and Guestrin, 2016). Besides the regularization, XGBoost applies a shrinkage component, meaning every new tree is scaled with a specific learning rate before being added to the ensemble. Through this, optimization is stabilized and many small, controlled improvements are enabled. Additionally, a certain minimal loss-reduction per split is enforced, such that only remunerative splits are accepted and, similar to random forests, on top of row-sub-sampling, column-sub-sampling is leveraged, randomly selecting features per tree or even per split. (Chen and Guestrin, 2016). For this thesis, the package `xgboost` (Chen et al., 2025) was used, which is also the package leveraged by Berrar et al. (2019), who praise it for outstanding performances in various data mining competitions organized by Kaggle. Choosing `objective = "multi:softprob"`, the XGBoost training function is set to deliver probabilities for the classes of the target variable. The learning rate η, indicating the improvements from newly added trees (Berrar et al., 2019), was set to a tendentially small value of 0.05, while `subsample`, indicating the percentage of all rows in the train data utilized by each tree, and `colsample_bytree`, specifying the percentage of feature columns included to grow each tree, equivalent to the random subspace of a random forest, were both set to 0.8, following the recommendations of Berrar et al. (2019). The maximum tree depth, `max_depth`, was set to 6, allowing deeper trees than suggested by Hubáček et al. (2019).

2.3. Feature Engineering

Form parameter Because winning against the bottom teams in the table is arguably easier than winning against the top teams in the table and a team's fixtures could be scheduled in a way such that it has to play against weaker opponents back to back for a couple of weeks, making it easier to score a lot of points, a parameter capturing the recent performance and current strength of teams, considering their past opponents, is needed. Considering only whether or not a team has won, drawn, or lost its' last games and how many points it scored inevitably ignores the importance of taking into account the strength of the opponents in these games. Therefore, Baboota and Kaur (2019) introduce form as a covariate to measure recent performance and current strength. This is the form parameter leveraged and analyzed in this project.

At the beginning of each season, all teams are initialised with a form parameter value of 1, which is iteratively updated after each game, such that higher values indicate better form. Unlike streak, another parameter introduced by Baboota and Kaur (2019), form can incorporate a team's performance relative to their opponents, meaning a weak team beating a strong team results in a greater coefficient update than the opposite case. Additionally, a draw leads to a coefficient decrease for the stronger team and an increase for the weaker team.

In case team h beats team a in game j, the exact form values f_j^h, representing the form of team h after game j, and f_j^a, representing the form of team a after game j, are calculated using γ, which is a stealing parameter $\in (0, 1)$ that can be interpreted as a certain fraction the winning team steals from the loosing team, as follows

$$f_j^h = f_{(j-1)}^h + \gamma \cdot f_{(j-1)}^a,$$
$$f_j^a = f_{(j-1)}^a - \gamma \cdot f_{(j-1)}^a.$$

In case of a draw, the parameters are calculated with these formulas

$$f_j^h = f_{(j-1)}^h - \gamma \cdot \left(f_{(j-1)}^h - f_{(j-1)}^a\right),$$
$$f_j^a = f_{(j-1)}^a - \gamma \cdot \left(f_{(j-1)}^a - f_{(j-1)}^h\right).$$

This leads to the previously weaker team stealing form from the previously stronger team after a draw. Baboota and Kaur (2019) concluded in their research that 0.33 is the best value for γ. Leveraging machine learning algorithms, like random forests and XGBoost, provides the opportunity to calculate team strength parameters and include them as a new covariate in a prediction model, similar to Groll et al. (2021). Following this strategy, the form parameter from Baboota and Kaur (2019) is tested, with all values from 0.05 up to 1 in steps of 0.05 and additionally 0.33 for the hyperparameter γ, in this thesis.

Bookmaker probabilities To create an additional comparison possibility for the models obtained in this project, the odds of a bookmaker too were utilized as prediction model. The betting odds used are those offered by bet365, as the company is the biggest bookmaker for sports bets globally (Ramirez et al., 2023). Since not odds but prediction probabilities were necessary for the different evaluation approaches leveraged in this project, as described below in subsection 2.5, the odds of the bookmaker bet365 had to be transformed to outcome probabilities following two steps.
Step one calls for all three bookmaker outcome odds (home win, draw, away win) to be converted into probabilities by dividing 1 by the bookmaker odds. With bmo_h representing the bookmaker odds for a home win, bmo_d the bookmaker odds for a draw and bmo_a the bookmaker odds for an away win, the probabilities p_h, p_d and p_a, symbolizing the probabilities of a home win, draw and away win occurring, are obtained as follows

$$p_h = \frac{1}{\text{bmo}_h},$$
$$p_d = \frac{1}{\text{bmo}_d},$$
$$p_a = \frac{1}{\text{bmo}_a}.$$

As these probabilities do not sum up to 1 but a higher number, because the bookmaker odds include the win margin of the bookmaker, step two demands the probabilities to be normalized, by dividing them by their total sum. With s_p denoting the sum of the beforehand calculated probabilities, p_h^n, p_d^n and p_a^n signify the normalized probabilities

$$
\begin{aligned}
s_p &= p_h + p_d + p_a, \\
p_h^n &= \frac{p_h}{s_p}, \\
p_d^n &= \frac{p_d}{s_p}, \\
p_a^n &= \frac{p_a}{s_p}.
\end{aligned}
$$

With these normalized probabilities the performance of bet365 odds as predictors could be evaluated.

2.4. Challenges

First k games of the season Forecasting the first k games of a season in a league competition, with k representing the number of previous games used to predict the upcoming game, can be a problem. This problem becomes apparent, for instance, when a model strictly utilizes the last 5 games to predict the following one. Such a model will be operable for the first time for the prediction of the sixth game of a season (Baboota and Kaur, 2019). This is due to the fact, that games from the past season cannot be incorporated into the model to predict games of the current season, since over the break between two seasons, teams, as well as the leagues as a whole, change. On a team level, players leave clubs to join others or retire from the sport entirely, altering a teams' squad (Koopman and Lit, 2015). Additionally, on the league level, clubs are promoted to higher leagues and relegated to lower leagues, transforming the compositions of the competitions every year. Predicting the performance of changed teams against new rivals based on their performance against clubs from a different league from a year before is not suitable.
One possible way to address this issue suggests relying only on the games already played in the season up to the k-th. It follows that with such an approach the games considered must be weighted differently than for the rest of the season, until game number k has been played (Baboota and Kaur, 2019). This enables predictions from the second game of the season onwards. However, these predictions can be expected to be less accurate than the ones following after game week $k + 1$. The models developed for this thesis avoid the first k games of the season problem by starting predictions in gameweek $k + 1$.

Underrepresentation of draws A different problem occurring, especially for models based on a Poisson distribution, is a tendential underrepresentation of draws (Dixon and Coles, 1997; Karlis and Ntzoufras, 2005). Games ending 0-0 or 1-1 are affected the most (Koopman and Lit, 2015). To deal with this issue, Koopman and Lit (2015) utilize diagonal inflation, applying an adjustment term, shifting probability mass away from 0-1 and 1-0 towards 0-0 and 1-1, while Karlis and Ntzoufras (2005) increase the probability of all draws. For this thesis, no similar adjustment has been done in order to fully embrace and not alter the results of complex machine learning. Nevertheless, since Baboota and Kaur (2019) suggest XGBoost models to leap frog other complex machine learning models when predicting draws, the differences in the predicted number of draws between the different models are of interest to this project.

2.5. Evaluation

The decision of which metric to use in order to evaluate a specific problem's outcome should consider the according problem's underlying scale type (Constantinou and Fenton, 2012). E.g., the possible outcomes when rolling a dice give the impression of an ordered set $\{1, 2, ..., 6\}$, but the true scale type is nominal, because in this set, if the true outcome of rolling the dice would be a 3, a predicted 2 is not closer to this true outcome than a predicted 6. For football predictions with the set of possible outcomes {H, D, A}, representing a home win, draw as well as an away win, the scale is ordinal, due to the fact that in case of a home win a predicted draw is closer to the true outcome than a predicted away win. Imagining the home team leading a game by one goal, meaning in this scenario the away team would only need to score one goal to change the outcome to a draw, but two goals to change it to an away game, illustrates this conveniently (Constantinou and Fenton, 2012). Hence, probabilistic football predictions should be evaluated with a metric considering this ordinal scale.

RPS According to Murphy (1970) a metric especially suiting for the evaluation of ordered variables' probability forecasts is the RPS, introduced by Epstein (1969). With its characteristic of higher punishment for predictions that differ more from the actual outcome, due to its sensibility to distance (Constantinou and Fenton, 2012; Robberechts and Davis, 2019; Ley et al., 2019; Baboota and Kaur, 2019; Groll et al., 2021, 2024), additionally to it being a strictly proper probabilistic scoring rule, minimizing its expected value in case the true outcomes are predicted (Tsokos et al., 2019; Groll et al., 2024), which is beneficial since it measures errors, the RPS fits football predictions well. Therefore, it was chosen as evaluation metric for this project.
The RPS, indicating the difference between the predicted and observed probability distributions of the different game outcomes, implies that, in order for a prediction model to be evaluated with the RPS, it needs to deliver probability predictions for the the three possible game outcomes, home win, draw and away win (Baboota and Kaur, 2019; Hubáček et al., 2019). The final value of the RPS is calculated by accumulating the single values from each game, with a smaller RPS indicating a better model.
Constructing a formula from this, with M denoting the total number of games considered, m the current game, P_{H_m} and P_{A_m} the predicted probabilities for a home and away win in game m respectively, y_{H_m} and y_{A_m} the actual outcome of game m, with $y_{H_m} = 1$ in case of a home win and 0 otherwise, in addition to $y_{A_m} = 1$ in case of an away win and 0 otherwise, making an additional term for draws unnecessary due to the ordered structure of the variables, the RPS is calculated as follows (Ley et al., 2019)

$$\text{RPS} = \frac{1}{2M} \sum_{m=1}^{M} \left((P_{H_m} - y_{H_m})^2 + (P_{A_m} - y_{A_m})^2 \right).$$

Accuracy As argued above for the proper evaluation of probabilistic football prediction models, a metric taking the ordinal scale into consideration is needed. Despite accuracy, measuring the share of correctly predicted games, being no such metric, it is still considered in this thesis due to two reasons. First, as betting profits can only be realized if a model delivers accurate predictions, it was desired to test if the models scoring the highest accuracy would return the highest profits too, when applying the below described betting strategies. Second, it was sought to find out whether or not there would be great differences in the below, under section 4, described model variations leading to best RPS and accuracy.

2.6. Betting strategies

Football betting is a hundreds of billion dollar business globally (Baboota and Kaur, 2019). Knowledge about the outcome of games can turn out to be very lucrative. Once one has created a prediction model serving probabilities for the three different possible outcomes of the game, they can engage in different betting approaches. To determine if the models of this thesis are able to create a positive return when applied to football bets, the following two different betting strategies were applied.

Quality Events This approach includes calculating the expected value of the payout of a bet as described by Koopman and Lit (2015). Utilizing $P(A)$, representing the probability of an event A, denoting either a home win, draw, or away win, occurring according to the prediction model in use, $\mathrm{bmo}(A)$, symbolizing the odds offered by the bookmaker for event A, and $\mathbb{E}(p_A)$, the expected value of the payout when betting on event A, with a bet stake of one unit, $\mathbb{E}(p_A)$ builds on $P(A) \cdot \big(\mathrm{bmo}(A) - 1\big)$, for the case event A occurs, and $P(\text{not } A) \cdot (-1)$ equalling $-P(\text{not } A)$ representing the case event A does not occur. $\mathbb{E}(p_A)$ is therefore calculated as follows

$$\mathbb{E}(p_A) = P(A) \cdot \big(\mathrm{bmo}(A) - 1\big) - P(\text{not } A).$$

This can be simplified to

$$\mathbb{E}(p_A) = P(A) \cdot \mathrm{bmo}(A) - P(A) - P(\text{not } A)$$

and since $-P(A) - P(\text{not } A) = -1$, it can be obtained that

$$\mathbb{E}(p_A) = P(A) \cdot \mathrm{bmo}(A) - 1.$$

The simplest betting strategy following this approach would be to bet on all events returning a positive expected value. However, this strategy does not necessarily lead to the highest expected profit. Setting a threshold value τ for $\mathbb{E}(p_A)$ and betting only on events exceeding this threshold, called *quality events*, indicates higher values for τ leading to higher returns, even though the number of bets declines with increasing τ (Koopman and Lit, 2015).
Additionally, Koopman and Lit (2015) bet on long shots, defining such as events with odds higher than 7. With τ set to 0.11, they identified 74 long shots over 2 seasons, from which they predicted 8 right, leading to a net profit of 5.07 units, when betting 0.3 units on each long shot. Nevertheless, their research indicates highest profits for $0.43 \leq \tau \leq 0.45$.
Different safety approaches would be to keep the variance of the expected profit below some limit and place bets in such a way that the expected profit is maximised within this limit or maximising expected profit minus the variance (Rue and Salvesen, 2001).

The quality event approach can, but does not necessarily have to, lead to more than one quality event for a single game. Therefore, it is assessed in two ways. The first envisages betting on maximum one quality event per game. In case more than one quality event is identified for a game, a bet is solely placed on the event with the higher expected value, leading to this method being called *one bet*. Following the other tactic, a bet is placed on every quality event identified, enabling multiple bets per game and leading to the name *multi bet*.
Koopman and Lit (2015) evaluated the performance of their betting strategy and model with the 2010/11 and 2011/2012 Premier League seasons. Since the data sources in use for this thesis unfortunately do not offer as detailed information for these seasons as they do from the 2017/18 season onwards and the models of this project were trained on data and variables, of which some are only available since 2017/18, a comparison with the 2010/11 and 2011/12 betting results from Koopman and Lit (2015) was not possible.

Naive Approach To enable comparisons, a second betting strategy was tested. Due to its simplicity, the strategy is called Naive Approach. Following this strategy simply denotes placing a bet on the outcome of the game a model predicts as the most likely to occur. For all bets and strategies, every bet always has the same stake of 1 unit.

3. DATA

Data was retrieved from two sources. `https://www.football-data.co.uk/` has a tremendous offering for football statistics, spanning over the big five European leagues from England, Spain, France, Italy and Germany, as well as smaller leagues from Europe, Asia, North America and South America, starting in 1993/94 and provided as csv files, in which each row represents one game, directly on the website, granting free and uncomplicated access. Additionally to various game statistics regarding team performances, at `https://www.football-data.co.uk/` betting odds from multiple betting offices are collected. Leveraging the data from this source, a form parameter, as described in subsection 2.3, was created. Additionally, the Full Time Result (home win, draw, away win), names of the teams and the dates of the fixtures are taken from this source. Team names and fixture dates are needed to match the data with the second source.

More exhaustive game statistics were obtained from `https://fbref.com/`, using the R package worldfootballR, a package allowing the collection of various results and statistics from different data sites (Zivkovic, 2022). On `https://fbref.com/` very detailed information for many football games, for the English Premier League especially since the 2017/18 season, is provided. Utilizing the worldfootballR package, three different dataframes per game were scraped from the website. These incorporate numerous variables collecting information about the teams fairness, offense, defense, playmaking, passing and ball carrying. Since some of these variables are included in two or even all three dataframes of a specific game, the dataframes need to be cleaned up before combining all data. Furthermore, the formats for dates differ between the two data sources, and some team names are abbreviated on the dataframes from `https://fbref.com/`. Hence, to match both sources, a common format for date had to be established and some teams had to be renamed.

4. COMPUTATIONAL DETAILS

Data used for the models stems from the seasons 2017/18 – 2024/25 of the English Premier League, because for these seasons the most variables and the most detailed information are available. The first 7 of these 8 seasons are used for training purposes, while the eighth season is used as test dataset. Since complex machine learning models decide on their own on the importance of the variables applied, no variable was omitted beforehand from the models, except for non-team specific variables as the referee or attendance. Instead, all variables available were employed.

Another important feature was not taking into account only the last game the teams played, since teams experiencing subpar performances on occasion, sometimes even across multiple consecutive games, is not an anomaly. Considering only the last game can distort the true abilities of a team.

4.1. Accounting for past games

As argued above, the further back a game lays, the weaker its' representative power over the current strength of a team. One possible approach to addressing this is the Half Period principle. Using this principle would lead to past games loosing influential power on the upcoming prediction with time, with the importance of a game already played reducing to half after a previously defined time period has passed (Ley et al., 2019; Groll et al., 2021). For this project, each teams' last 5 games were included in the models. To fully embrace the capabilities of complex machine learning however, principles like Half Period were not applied. Instead, the machine should learn the influential importance of the past games on the upcoming one on its own. It follows, that the explanatory variables are not fed only once to the models, as could be done if only the last game was to be considered or if the last k games were to be considered in a weighted manner, but they are fed 5 times to the models, one time for each of the last 5 games. For explanatory purposes this mechanism is illustrated using the variable *Goals scored*. This variable, like every other play-specific variable, which are variables representing game relevant statistics, exists in 5 variations in the models. Them being *Goals scored 1 game ago, Goals scored 2 games ago, Goals scored 3 games ago, Goals scored 4 games ago* and *Goals scored 5 games ago*. Using this approach no weights have to be assigned to the according games following intuitions and assumptions, but the machine can learn to attribute importance of the past games on its own. The only exception from this 5-variations per team-specific variable approach is the form parameter variable, which is only fed once and in its most recent form to the models, as this covariate incorporates the current strengths of the teams directly before a game. Furthermore, as the form parameter is calculated iteratively after each game, past games are already accounted for in it automatically. A detailed list of all variables included in the models can be found in the appendix.

4.2. Model variations

At the beginning of this project, one goal formulated was to determine, whether or not including betting odds as covariates into the models improves prediction performance, while the second goal was ascertaining the influence of the form parameter. Not only to find out, if its application leads to performance improvements, but also to identify the best value for the stealing parameter γ, necessary for the computation of the form parameter, as described in subsection 2.3. Literature suggest an optimal γ of 0.33 (Baboota and Kaur, 2019). These goals led to the following model variations.

All Variables (-) Form Parameter (-) Betting Odds
All play-specifc variables (e.g., goals scored, progressive passes, ball carries, etc.) are included and accounted for the last 5 games as described above. The **form parameter** and **betting odds** are **excluded**.

All Variables (-) Form Parameter (+) Betting Odds
All play-specifc variables (e.g., goals scored, progressive passes, ball carries, etc.) are included and accounted for the last 5 games as described above. The **form parameter** is **excluded**, the **betting odds** are **included**.

All Variables (+) Form Parameter (-) Betting Odds
All play-specifc variables (e.g., goals scored, progressive passes, ball carries, etc.) are included and accounted for the last 5 games as described above. The **form parameter** is **included**, the **betting odds** are **excluded**. The form parameter was calculated in 21 different variations as the hyperparameter γ was set to all values from 0.05 up to 1.0, increasing in steps of 0.05, and one additional variation with $\gamma = 0.33$, since this is the suggested best value (Baboota and Kaur, 2019), leading to 21 model sub-variations.

All Variables (+) Form Parameter (+) Betting Odds
All play-specifc variables (e.g., goals scored, progressive passes, ball carries, etc.) are included and accounted for the last 5 games as described above. The **form parameter** and the **betting odds** are **included**. Again, the form parameter was calculated in the same 21 different variations as for **All Variables (+) Form Parameter (-) Betting Odds**, leading to another 21 model sub-variations.

Only Form Parameter Variable
The only explanatory variable deployed is the form parameter in its 21 different variations.

Every time the form parameter is utilized, it is included once for the home team and once for the away team. With betting odds, always the betting odds before the start of a game for home win, draw and away win are meant. Before training the models, expectations were that the inclusion of the betting odds as well as the form parameter would improve the models' performances. All model variations were implemented with one XGBoost model and two random forest models, once with the package `randomforest` (Liaw and Wiener, 2002b) and once with the package `ranger` (Wright and Ziegler, 2017), accumulating to a total of 195 models.

Outputs Goal of the models is predicting the probabilities for a home win, a draw and an away win. With these parameters the performance of the models can be measured with the RPS. Additionally, the models are applied to the betting strategy from Koopman and Lit (2015).

For this too, the probabilities for the 3 possible results of each game were needed, because otherwise identifying quality events, like Koopman and Lit (2015), would not have been possible. Therefore, the models' outputs are probabilities for the three different classes home win, draw and away win. The R files created during the course of this project can be found at `https://github.com/hkoerbel/Football_Predictions_Bachelor_Thesis`.

5. RESULTS

The results section is structured the following. First, the results of the different models concerning the RPS are presented, followed by the results measured with accuracy, before the chapter is closed with an exhibition of the findings of the betting strategies. Where possible, the model building on the bookmakers odds, the best XGBoost, random forest and ranger models respectively as well as the best models of each variation described in section 4 are compared, additionally to the average scores of the different model variations.

Information about what variables a model utilized are to be found in its name, since names including the abbreviation *av* indicate that the corresponding model utilized all game-specific variables available for this project as covariates. The "+" or "-" in brackets before *fp* reveals if a model leveraged the form parameter, "+" meaning yes and "-" meaning no, while the "+" or "-" in brackets before *bq* indicates the usage of betting odds in the same way. The difference between *bq* in the name of the models to *BO* in the name of the model variations originates in a minor inconsistency, because during the coding process, the idea was to use the term "betting quotes", inspired by the German word "Quoten", giving the models their names, but during the writing process, the word "odds" was used, due to the actually different meaning of the word "quotes". Names of models, depending on the form parameter as the only explanatory variable, start with *only_fp*. If a model name includes a three digit long number, this number indicates the value of the hyperparameter γ utilized to calculate the form parameter.

Models depending on the form parameter as the only covariate had a testset of 379 games, while all other models had a testset of 329 games. The difference of 50 games arises from the fact that models incorporating all available game-specific variables could only start predicting in the sixth round of the test season, in order for the inclusion of the variables of the last five games played to work. Since the English Premier League consists of 20 teams, each round includes 10 games. Multiplying by 5 rounds leads to the difference of 50 games. One game had to be deleted from the test set due to missing data, causing the 379 and 329 test games, instead of 380 and 330.

The true values for home wins, draws and away wins are 155, 92 and 132 for models depending only on the form parameter and 138, 77 and 114 for all other models.

Detailed information on the RPS and accuracy of all 195 models is to be found in the appendix.

5.1. Results RPS

Applying the model based on the bookmaker odds delivered an RPS of 0.2023 which could not be beaten by any of the other models. The score of the best model however, with an RPS of 0.2144, was not far away, giving confidence for complex machine learning models to be suitable for modeling and predicting football games. This model is based on a random forest and leverages both, a form parameter, calculated with a stealing parameter $\gamma = 0.30$, and betting odds. As expected, the inclusion of betting odds and a form parameter improved performance. The best results are achieved by models leveraging both, followed by models employing the betting odds but not the form parameter in front of models applying the form parameter but not the betting odds, before models including all game-specific variables but neither form parameter nor betting odds, with models using only the form parameter performing worst. With the best model building on a form parameter calculated with $\gamma = 0.30$, the suggested best value of 0.33 from Baboota and Kaur (2019) is close. However, since other models achieved similar results with form parameters created through very different values for γ, to conclude on one single specific value for γ to be the best seems premature. Figure 1 shows the trend of the RPS vs. γ for the different models from the variation leveraging the form parameter as well as the betting odds, because this model variation achieved the lowest RPS scores, indicating no clear trend towards a certain best γ-value, while Figure 2 visualizes the same for accuracy vs. γ, although for accuracy it seems that, for models based on the `randomForest` package (Liaw and Wiener, 2002b), γ-values between $0.45 - 0.55$ lead to the worst results. Interestingly, only XGBoost models consistently predict at least a few draws. The two different random forest models almost never predict any draws at all, still they outperformed the XGBoost models when it comes down to the RPS. The most draws are predicted by models leveraging a form parameter as the only explanatory variable. Looking at the rather bad RPS these models scored on average, it gives the impression that these predicted draws are more a product of coincidence rather than a good prediction model.

5.2. Results Accuracy

Utilizing the bookmaker odds based model delivered an accuracy of 53.5% which, again, could not be beaten. With accuracy as the deciding measure, the best model was created with the `ranger` package (Wright and Ziegler, 2017) and achieved an accuracy of 52.89%, which is only 0.61% away from the bookmakers, further increasing confidence for complex machine learning models being suitable for predicting football games. Again, the best results were achieved by models utilizing both a form parameter and betting odds, followed by models including the betting odds and excluding the form parameter and models excluding the betting odds and including the form parameter, before models excluding both, with models only leveraging a form parameter performing worst again. Figure 3 displays these differences of the model variations through the average results of the RPS and accuracy, with the left y-axis belonging to the RPS and the right y-axis belonging to accuracy.

version	name	n games	RPS	pred. H	pred. D	pred. A
Best Bookmakers	`Bookmakers`	329	0.2023	214.00	00.00	115
Best XGBoost	`xgboost_only_fp_010`	379	0.2220	222.00	34.00	123.00
Best random forest	`rf_av_(+)fp_030_(+)bq`	329	0.2144	218.00	00.00	111.00
Best ranger	`ranger_av_(+)fp_055_(+)bq`	329	0.2149	226.00	00.00	103.00
Best AV(-)FP(-)BO	`ranger_av_(-)fp_(-)bq`	329	0.2259	226.00	00.00	103.00
Avg. AV(-)FP(-)BO	`NA`	329	0.2353	226.00	02.67	100.33
Best AV(-)FP(+)BO	`rf_av_(-)fp_(+)bq`	329	0.2166	218.00	00.00	111.00
Avg. AV(-)FP(+)BO	`NA`	329	0.2214	215.67	02.00	111.33
Best AV(+)FP(-)BO	`rf_av_(+)fp_020_(-)bq`	329	0.2227	225.00	00.00	104.00
Avg. AV(+)FP(-)BO	`NA`	329	0.2312	224.33	02.27	102.40
Best AV(+)FP(+)BO	`rf_av_(+)fp_030_(+)bq`	329	0.2144	218.00	00.00	111.00
Avg. AV(+)FP(+)BO	`NA`	329	0.2207	216.75	02.56	109.70
Best Only FP	`xgboost_only_fp_010`	379	0.2220	222.00	34.00	123.00
Avg. Only FP	`NA`	379	0.2502	225.27	40.97	112.76

Table 1: A table representing the results of the best and the average models of the different model variations, measured with the RPS and the number of predicted home wins, draws as well as away wins.

version	name	n games	accuracy	pred. H	pred. D	pred. A
Best Bookmakers	`Bookmakers`	329	53.50	214.00	00.00	115.00
Best XGBoost	`xgboost_av_(+)fp_005_(-)bq`	329	52.58	217.00	03.00	109.00
Best random forest	`rf_av_(+)fp_090_(+)bq`	329	52.28	218.00	01.00	110.00
Best ranger	`ranger_av_(+)fp_015_(+)bq`	329	52.89	220.00	00.00	109.00
Best AV(-)FP(-)BO	`ranger_av_(-)fp_(-)bq`	329	48.33	226.00	00.00	103.00
Avg. AV(-)FP(-)BO	`NA`	329	46.91	226.00	02.67	100.33
Best AV(-)FP(+)BO	`xgboost_av_(-)fp_(+)bq`	329	51.06	207.00	06.00	116.00
Avg. AV(-)FP(+)BO	`NA`	329	50.46	215.67	02.00	111.33
Best AV(+)FP(-)BO	`xgboost_av_(+)fp_005_(-)bq`	329	52.58	217.00	04.00	109.00
Avg. AV(+)FP(-)BO	`NA`	329	48.48	224.33	02.27	102.40
Best AV(+)FP(+)BO	`ranger_av_(+)fp_015_(+)bq`	329	52.89	220.00	00.00	109.00
Avg. AV(+)FP(+)BO	`NA`	329	51.23	216.75	02.56	109.70
Best Only FP (1)	`ranger_only_fp_010`	379	46.97	211.00	42.00	126.00
Best Only FP (2)	`ranger_only_fp_035`	379	46.97	210.00	48.00	121.00
Avg. Only FP	`NA`	379	42.22	225.27	40.97	112.76

Table 2: A table representing the results of the best and the average models of the different model variations, measured with accuracy and the number of predicted home wins, draws as well as away wins.

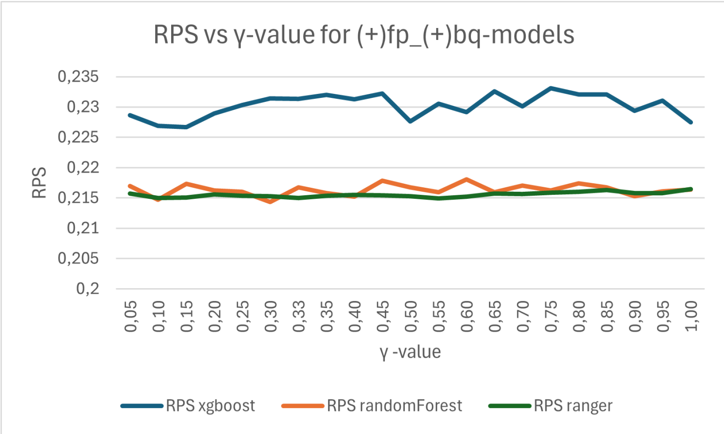

Figure 1: A figure visualizing the trend of the RPS vs. γ from xgboost, randomForest and ranger package based models of the variation leveraging a form parameter as well as betting odds.

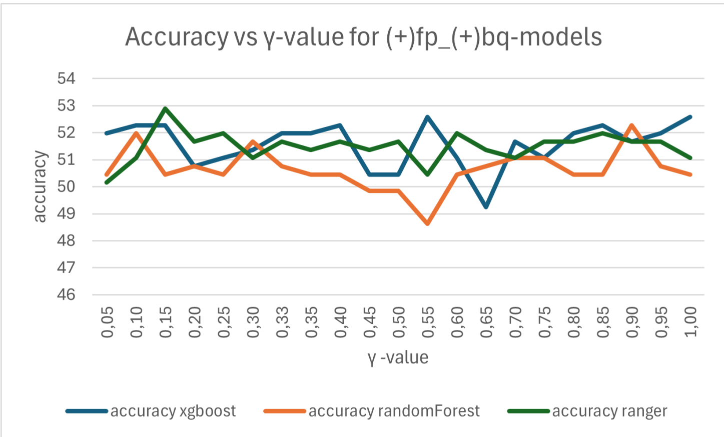

Figure 2: A figure visualizing the trend of accuracy vs. γ from xgboost, randomForest and ranger package based models of the variation leveraging a form parameter as well as betting odds.

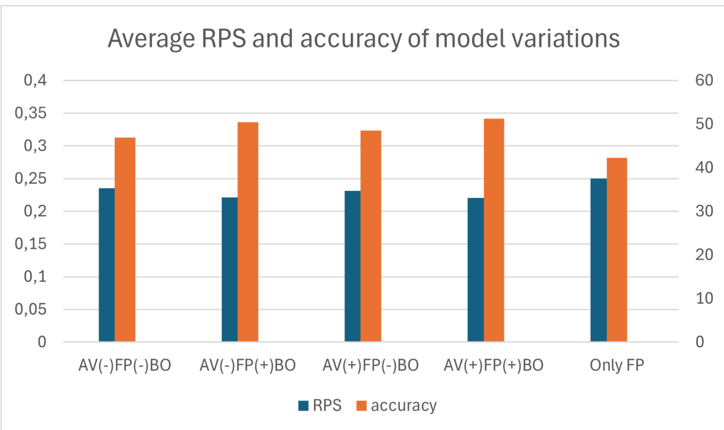

Figure 3: A figure visualizing the average RPS and accuracy scores of all different model variations.

5.3. Results Naive Approach Betting Strategy

Not surprisingly, the naive approach betting strategy did not return profits. All model variations scored a negative net profit and a return factor smaller than 1 on average. It follows that simply betting on the outcome with the highest predicted probability for every game is not a good betting strategy.

version	name	net profit	return factor	n bets
Best Bookmakers	Bookmakers	-19.57	0.95	379
Best XGBoost	xgboost_av_(+)fp_005_(-)bq	-4.97	0.98	329
Best random forest	rf_only_fp_040	10.52	1.03	379
Best ranger	ranger_only_fp_030	14.61	1.04	379
Best AV(-)FP(-)BO	ranger_av_(-)fp_(-)bq	-35.51	0.89	329
Avg. AV(-)FP(-)BO	NA	-41.22	0.87	329
Best AV(-)FP(+)BO	xgboost_av_(-)fp_(+)bq	-27.94	0.92	329
Avg. AV(-)FP(+)BO	NA	-34.81	0.89	329
Best AV(+)FP(-)BO	xgboost_av_(+)fp_005_(-)bq	-4.97	0.98	329
Avg. AV(+)FP(-)BO	NA	-28.95	0.91	329
Best AV(+)FP(+)BO	rf_av_(+)fp_090_(+)bq	-15.44	0.95	329
Avg. AV(+)FP(+)BO	NA	-28.78	0.91	329
Best Only FP	ranger_only_fp_030	14.61	1.04	379
Avg. Only FP	NA	-35.59	0.91	379

Table 3: A table representing the returns of the best and the average models of the different model variations when deploying the naive approach betting strategy.

5.4. Results Quality Events Betting Strategy

Koopman and Lit (2015) suggest an optimal threshold of $0.43 \leq \tau \leq 0.45$ for betting on quality events. To evaluate this strategy properly, all models were run with the one bet and the multi bet approach and values for τ ranging from 0 to 4, increasing in steps of 0.01. As this resulted in almost 80 000 outcomes, only the most relevant are presented here, them being highest absolute profit, highest relative profit and most τ values with a positive return. Since the latter two are achieved by the same model, only two different models are included in the following table. The highest absolute profit accumulated to 42.6 units over the course of one season, while the highest relative profit resulted in a return factor of 12. However, since this very high return factor resulted from a scenario where the threshold τ was set to 3.88, leading to only 1 bet being placed, this scenario is not accounted for the best return factor. Instead a second threshold of at least 30 bets placed was set, resulting in a highest return factor of 1.65, meaning a plus of 65% on the total stakes, achieved by the same model, which had before achieved the return factor of 12. Both results, highest absolute and highest relative profit, originate from models only using the form parameter as covariate. As described above, these models tendentially predict more draws than other models. It gives the impression that predicting more draws leads to higher returns, which seems plausible, since draws are underrepresented in all models, but while often not a single draw is predicted in models building on all game-specific variables, those only utilizing the form parameter all predict a double-digit number of draws. Conclusively, models using all game-specific variables tendentially take themselves away the possibility to make a profit from games ending in a draw, instead scoring a monetary loss almost certainly with each draw. Koopman and Lit (2015) scored a plus of 50% over the combined time of the 2010/11 and 2011/12 Premier League seasons with their bivariate Poisson model. Still, the 90% confidence interval for the mean return of their model was not all positive, making it not risk safe. The best model of this thesis scored a plus of 65% over the course of a single season. However, this model too has a confidence interval with the lower bound stretching into the negative area consistently. Even though there is the small difference, that this is a 95% confidence interval, in contrast to the 90% of Koopman and Lit (2015), because stricter controls were desired, this still not makes this combination of model and betting strategy advisable to pursue. The following graphs illustrate these findings, with Figure 4 showing the return factor as a function of τ for model ranger_only_fp_005, which was the model scoring the best return factor, and Figure 5 illustrating the net profit.

name	selection reason	variant	net profit	best τ	return factor	n bets
`rf_only_fp_033`	highest absolute profit	one bet	42.6	0.17	1.13	334
`rf_only_fp_033`	highest absolute profit	multi bet	19.48	0.24	1.06	333
`ranger_only_fp_005`	highest relative profit and most positive τ	one bet	23.93	1.25	1.65	37
`ranger_only_fp_005`	highest relative profit and most positive τ	multi bet	23.93	1.25	1.65	37

Table 4: A table representing the different variants of the models with the highest returns when deploying the quality events betting strategy.

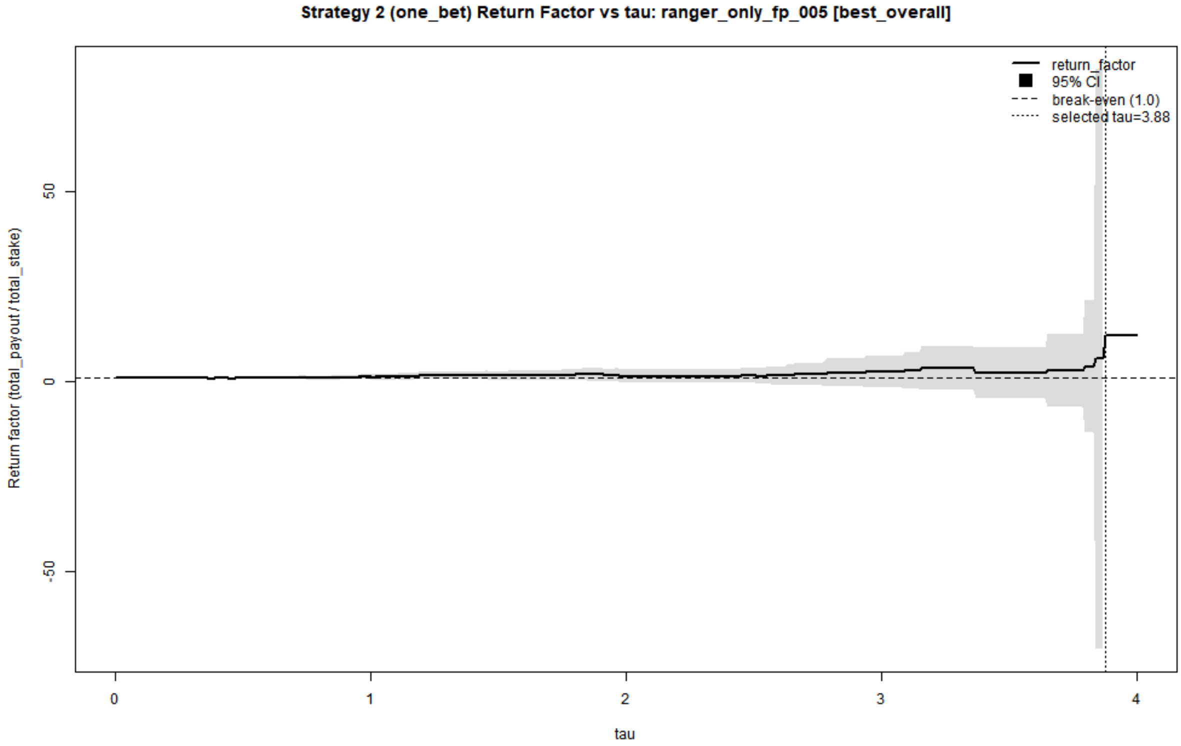

Figure 4: A figure visualizing the return factor of model ranger_only_fp_005 with its 95% confidence interval as a function of τ using betting strategy quality events.

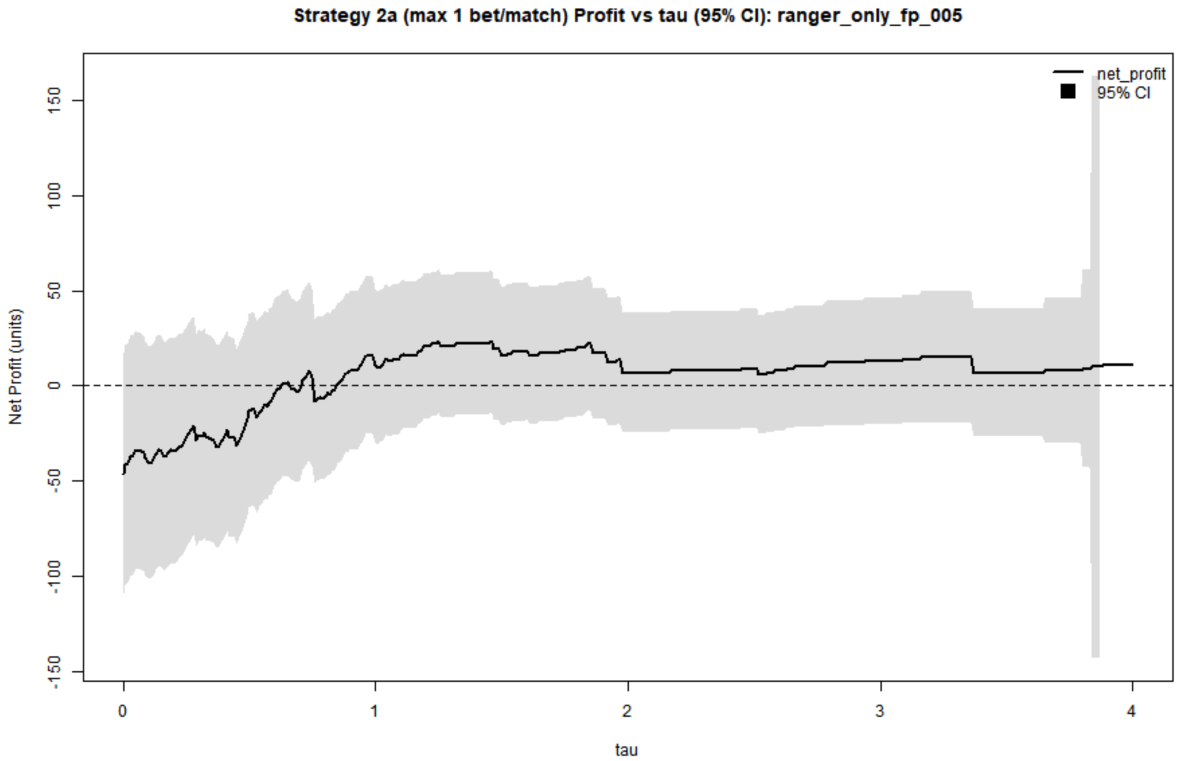

Figure 5: A figure visualizing the net profit of model ranger_only_fp_005 with its 95% confidence interval as a function of τ using betting strategy quality events

6. DISCUSSION

In answering the question if the inclusion of bookmaker betting odds and a performance based form parameter as covariates for complex machine learning models improves their prediction performance, the results of this project indicate they do. With the RPS as chosen evaluation metric, the best scores, both the single best and the best average, were achieved by models leveraging both bookmaker odds as well as a form parameter. When investigating if the suggested best value of 0.33 for the hyperparameter γ of the form parameter truly delivers the best results, the suggestion from Baboota and Kaur (2019) cannot be supported, as no clear trend towards a best value of γ could be identified.

It could be shown that models, based on the packages `xgboost` (Chen et al., 2025), `randomForest` (Liaw and Wiener, 2002b) and `ranger` (Wright and Ziegler, 2017), are capable of delivering good models for the purpose of predicting football games without the need for feature pre-selection, as the achieved RPS scores are satisfying, especially in comparison to a bookmaker.

One major challenge of complex machine learning football prediction models remaining is the underrepresentation of draws. While `xgboost` (Chen et al., 2025) based models at least predict a small amount, `randomForest` (Liaw and Wiener, 2002b) and `ranger` (Wright and Ziegler, 2017) based models almost never do, while still achieving lower RPS values.

REFERENCES

R. Baboota and H. Kaur. Predictive analysis and modelling football results using machine learning approach for english premier league. *International Journal of Forecasting*, 35:741–755, 2019. doi:10.1016/j.ijforecast.2018.01.003.

D. Berrar, P. Lopes, and W. Dubitzky. Incorporating domain knowledge in machine learning for soccer outcome prediction. *Machine Learning*, 108:97–126, 2019. doi:10.1007/s10994-018-5747-8.

L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996. doi:10.1023/A:1018054314350.

L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001. doi:10.1023/A:1010933404324.

T. Chen and C. Guestrin. Xgboost: a scalable tree boosting system. In *KDD '16: proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 785–794. Association for Computing Machinery, 2016. doi:10.1145/2939672.2939785.

T. Chen, T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho, K. Chen, R. Mitchell, I. Cano, T. Zhou, M. Li, J. Xie, M. Lin, Y. Geng, Y. Li, and J. Yuan. *xgboost: Extreme Gradient Boosting*, 2025. URL `https://CRAN.R-project.org/package=xgboost`. R package version 1.7.11.1.

A. C. Constantinou. Dolores: a model that predicts football match outcomes from all over the world. *Machine Learning*, 108:49–75, 2019. doi:10.1007/s10994-018-5703-7.

A. C. Constantinou and N. E. Fenton. Solving the problem of inadequate scoring rules for assessing probabilistic football forecast models. *Journal of Quantitative Analysis in Sports*, 8 (1), 2012. doi:10.1515/1559-0410.1418.

T. G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization. *Machine Learning*, 40:139–157, 2000. doi:10.1023/A:1007607513941.

M. J. Dixon and S. G. Coles. Modelling association football scores and inefficiencies in the football betting market. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 46:265–280, 1997. doi:10.1111/1467-9876.00065.

E. S. Epstein. A scoring system for probability forecasts of ranked categories. *Journal of Applied Meteorology*, 8:985–987, 1969. doi:10.1175/1520-0450(1969)008¡0985:ASSFPF¿2.0.CO;2.

J. H. Friedman. Greedy function approximation: a gradient boosting machine. *The Annals of Statistics*, 29:1189–1232, 2001. doi:10.1214/aos/1013203451.

A. Groll, L. M. Hvattum, C. Ley, F. Popp, G. Schauberger, H. Van Eetvelde, and A. Zeileis. Hybrid machine learning forecasts for the uefa euro 2020, 2021. arXiv preprint arXiv:1906.01131.

A. Groll, L. M. Hvattum, C. Ley, J. Sternemann, G. Schauberger, and A. Zeileis. Modeling and prediction of the uefa euro 2024 via combined statistical learning approaches, 2024. arXiv preprint arXiv:2410.09068.

T. K. Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:832–844, 1998. doi:10.1109/34.709601.

O. Hubáček, G. Šourek, and F. Železný. Learning to predict soccer results from relational data with gradient boosted trees. *Machine Learning*, 108:29–47, 2019. doi:10.1007/s10994-018-5704-6.

D. Karlis and I. Ntzoufras. Analysis of sports data by using bivariate poisson models. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 52:381–393, 2003. doi:10.1111/1467-9884.00366.

D. Karlis and I. Ntzoufras. Bivariate poisson and diagonal inflated bivariate poisson regression models in r. *Journal of Statistical Software*, 14:1–36, 2005. doi:10.18637/jss.v014.i10.

S. J. Koopman and R. Lit. A dynamic bivariate poisson model for analysing and forecasting match results in the english premier league. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 178:167–186, 2015. doi:10.1111/rssa.12042.

C. Ley, T. Van de Wiele, and H. Van Eetvelde. Ranking soccer teams on the basis of their current strength: a comparison of maximum likelihood approaches. *Statistical Modelling*, 19: 55–73, 2019. doi:10.1177/1471082X18817650.

A. Liaw and M. Wiener. Classification and regression by randomforest. *R News*, 2:18–22, 2002a.

A. Liaw and M. Wiener. Classification and regression by randomForest. *R News*, 2(3):18–22, 2002b. URL `https://CRAN.R-project.org/doc/Rnews/`.

M. J. Maher. Modelling association football scores. *Statistica Neerlandica*, 36:109–118, 1982. doi:10.1111/j.1467-9574.1982.tb00782.x.

A. H. Murphy. The ranked probability score and the probability score: a comparison. *Monthly Weather Review*, 98:917–924, 1970. doi:10.1175/1520-0493(1970)098¡0917:TRPSAT¿2.3.CO;2.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2025. URL `https://www.R-project.org/`.

P. Ramirez, J. J. Reade, and C. Singleton. Betting on a buzz: mispricing and inefficiency in online sportsbooks. *International Journal of Forecasting*, 39:1413–1423, 2023. doi:10.1016/j.ijforecast.2022.07.011.

P. Robberechts and J. Davis. Forecasting the fifa world cup – combining result- and goal-based team ability parameters. In *Machine learning and data mining for sports analytics (MLSA 2018)*, volume 11330 of *Lecture notes in computer science*, pages 16–30. Springer, Cham, 2019. doi:10.1007/978-3-030-17274-9_2.

H. Rue and O. Salvesen. Prediction and retrospective analysis of soccer matches in a league. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 49:399–418, 2001. doi:10.1111/1467-9884.00243.

J. Stuebinger, B. Mangold, and J. Knoll. Machine learning in football betting: prediction of match results based on player characteristics. *Applied Sciences*, 10, 2020. doi:10.3390/app10010046.

A. Tsokos, S. Narayanan, I. Kosmidis, G. Baio, M. Cucuringu, G. Whitaker, and F. Király. Modeling outcomes of soccer matches. *Machine Learning*, 108:77–95, 2019. doi:10.1007/s10994-018-5741-1.

B. Verkerk. Beating the bookie in the football betting market without using extensive match data, 2018. Research Paper, Amsterdam University.

M. N. Wright and A. Ziegler. ranger: A fast implementation of random forests for high dimensional data in C++ and R. *Journal of Statistical Software*, 77(1):1–17, 2017. doi:10.18637/jss.v077.i01.

Y. Yang. Research on the winning factors of football matches based on machine learning. *Academic Journal of Mathematical Sciences*, 4:51–56, 2023. doi:10.25236/AJMS.2023.040408.

J. Zivkovic. *worldfootballR: extract and clean world football (soccer) data*. CRAN, 2022.

7. APPENDIX

Variables included in the models

Which variables were utilized for the models is described in the tables below. Play-specific variables are variables, which were fed for all of the last 5 games to the models.

variable name	variable description	play-specific variable
Date	Date a game was played on.	No
Home_Team	Name of the home team in a specific game.	No
Away_Team	Name of the away team in a specific game.	No
FTR	Full Time Result of a game (either H in case of a home win, D in case of a draw or A in case of an away win).	No
Home_Form	Form parameter of the home team in advance of a specific game.	No
Away_Form	Form parameter of the away team in advance of a specific game.	No
B365H	Odds for a home win from bet365.	No
B365D	Odds for a draw from bet365.	No
B365A	Odds for an away win from bet365.	No
Home_Score	Goals scored by the home team.	Yes
Home_xG	Goals expected to be scored by the home team.	Yes
Home_Yellow_Cards	Yellow Cards received by the home team.	Yes
Home_Red_Cards	Red Cards received by the home team.	Yes
Home_Ast	Assists from the home team (the number of goals and assists can differ, because goals scored through a penalty or direct free kick do not have an assist).	Yes
Home_PK	Penalty kicks converted by the home team.	Yes
Home_PKatt	Penalty kicks attempted by the home team	Yes
Home_Sh	Shots by the home team.	Yes
Home_SoT	Shots on target by the home team.	Yes
Home_Touches	Number of ball touches by the home team.	Yes
Home_Tkl	Tackles done by the home team.	Yes
Home_Int	Interceptions done by the home team.	Yes
Home_Blocks	Shots blocked by the home team.	Yes
Home_npxG_Expected	Expected non-penalty goals for the home team.	Yes
Home_xAG_Expected	Expected assisted goals for the home team.	Yes
Home_SCA_SCA	Home team Shot-Creating-Actions (Number of actions that lead to a shot).	Yes

Table 5: Table 1/5 explaining the variables in use in the prediction models.

variable name	variable description	play-specific variable
Home_GCA_SCA	Home team Goal-Creating-Actions (Number of offensive actions immediately before a goal; can be greater then the number of goals scored).	Yes
Home_Att_Passes	Number of passes attempted by the home team.	Yes
Home_Cmp_percent_Passes	Percentage of completed passes by the home team.	Yes
Home_PrgP_Passes	Number of progressive passes by the home team.	Yes
Home_Carries_Carries	Number of ball carries by home team.	Yes
Home_PrgC_Carries	Number of progressive ball carries by the home team.	Yes
Home_Att_Take_Ons	Number of attempted take-ons by the home team (1 vs. 1 dribbles).	Yes
Home_Def Pen_Touches	Number of touches in the own penalty box by the home team.	Yes
Home_Def 3rd_Touches	Number of touches in the defending third by the home team.	Yes
Home_Mid 3rd_Touches	Number of touches in the middle third by the home team.	Yes
Home_Att 3rd_Touches	Number of touches in the attacking third by the home team.	Yes
Home_Att Pen_Touches	Number of touches in the opponents penalty box by the home team.	Yes
Home_Succ_percent_Take_Ons	Percentage of succesful take-ons by the home team (1 vs. 1 dribbles).	Yes
Home_Tkld_percent_Take_Ons	Percentage of take-ons by the home team that ended with the player dribbling being tackled.	Yes
Home_TotDist_Carries	Total distance the home team covered while carrying the ball.	Yes
Home_PrgDist_Carries	Progressive distance the home team covered while carrying the ball.	Yes
Home_Final_Third_Carries	Carries into the final third by the home team.	Yes
Home_CPA_Carries	Carries into the opponent's penalty area by the home team.	Yes
Home_Mis_Carries	Number of missed (failed) carries by the home team.	Yes
Home_Dis_Carries	Number of dispossessed carries by the home team (player dribbling was tackeld and the opponent took the ball).	Yes

Table 6: Table 2/5 explaining the variables in use in the prediction models.

variable name	variable description	play-specific variable
Home_Possession	Percentage of ball possession for the home team.	Yes
Home_Fouls	Number of fouls committed by the home team.	Yes
Home_Corners	Number of corners for the home team.	Yes
Home_Crosses	Number of cross balls played by the home team.	Yes
Home_Tackles	Number of tackles done by the home team.	Yes
Home_Aerials_Won	Number of aerial duels won by the home team.	Yes
Home_Clearences	Number of clearences by the home teams keeper.	Yes
Home_Offsides	Number of offsides committed by the home team.	Yes
Home_Home_Goal_Kicks	Number of goal kicks for the home team.	Yes
Home_Throw_Ins	Number of throw-ins for the home team.	Yes
Home_Long_Balls	Number of long balls played by the home team.	Yes
Away_Score	Goals scored by the away team.	Yes
Away_xG	Goals expected to be scored by the away team.	Yes
Away_Yellow_Cards	Yellow Cards received by the away team.	Yes
Away_Red_Cards	Red Cards received by the away team.	Yes
Away_Ast	Assists from the away team (the number of goals and assists can differ, because goals scored through a penalty or direct free kick do not have an assist).	Yes
Away_PK	Penalty kicks converted by the away team.	Yes
Away_PKatt	Penalty kicks attempted by the away team	Yes
Away_Sh	Shots by the away team.	Yes
Away_SoT	Shots on target by the away team.	Yes
Away_Touches	Number of ball touches by the away team.	Yes
Away_Tkl	Tackles done by the away team.	Yes
Away_Int	Interceptions done by the away team.	Yes
Away_Blocks	Shots blocked by the away team.	Yes
Away_npxG_Expected	Expected non-penalty goals for the away team.	Yes
Away_xAG_Expected	Expected assisted goals for the away team.	Yes
Away_SCA_SCA	Away team Shot-Creating-Actions (Number of actions that lead to a shot).	Yes
Away_GCA_SCA	Away team Goal-Creating-Actions (Number of offensive actions immediately before a goal; can be greater then the number of goals scored).	Yes

Table 7: Table 3/5 explaining the variables in use in the prediction models.

variable name	variable description	play-specific variable
Away_Att_Passes	Number of passes attempted by the away team.	Yes
Away_Cmp_percent_Passes	Percentage of completed passes by the away team.	Yes
Away_PrgP_Passes	Number of progressive passes by the away team.	Yes
Away_Carries_Carries	Number of ball carries by away team.	Yes
Away_PrgC_Carries	Number of progressive ball carries by the away team.	Yes
Away_Att_Take_Ons	Number of attempted take-ons by the away team (1 vs. 1 dribbles).	Yes
Away_Def Pen_Touches	Number of touches in the own penalty box by the Away team.	Yes
Away_Def 3rd_Touches	Number of touches in the defending third by the Away team.	Yes
Away_Mid 3rd_Touches	Number of touches in the middle third by the Away team.	Yes
Away_Att 3rd_Touches	Number of touches in the attacking third by the Away team.	Yes
Away_Att Pen_Touches	Number of touches in the opponents penalty box by the Away team.	Yes
Away_Succ_percent_Take_Ons	Percentage of succesful take-ons by the Away team (1 vs. 1 dribbles).	Yes
Away_Tkld_percent_Take_Ons	Percentage of take-ons by the Away team that ended with the player dribbling being tackled.	Yes
Away_TotDist_Carries	Total distance the Away team covered while carrying the ball.	Yes
Away_PrgDist_Carries	Progressive distance the Away team covered while carrying the ball.	Yes
Away_Final_Third_Carries	Carries into the final third by the Away team.	Yes
Away_CPA_Carries	Carries into the opponent's penalty area by the Away team.	Yes
Away_Mis_Carries	Number of missed (failed) carries by the Away team.	Yes
Away_Dis_Carries	Number of dispossessed carries by the Away team (player dribbling was tackeld and the opponent took the ball).	Yes
Away_Possession	Percentage of ball possession for the away team.	Yes
Away_Fouls	Number of fouls committed by the away team.	Yes
Away_Corners	Number of corners for the away team.	Yes

Table 8: Table 4/5 explaining the variables in use in the prediction models.

variable name	variable description	play-specific variable
Away_Crosses	Number of cross balls played by the away team.	Yes
Away_Tackles	Number of tackles done by the away team.	Yes
Away_Aerials_Won	Number of aerial duels won by the away team.	Yes
Away_Clearences	Number of clearences by the away teams keeper.	Yes
Away_Offsides	Number of offsides committed by the away team.	Yes
Away_Away_Goal_Kicks	Number of goal kicks for the away team.	Yes
Away_Throw_Ins	Number of throw-ins for the away team.	Yes
Away_Long_Balls	Number of long balls played by the away team.	Yes

Table 9: Table 5/5 explaining the variables in use in the prediction models.

Model Performances: RPS and accuracy

The following tables present the performance results all models achieved, measured with the RPS and accuracy. The column *form parameter* states if a model leveraged a form parameter. If it did, the column γ indicates the value of the hyperparameter γ used to derive the associated form parameter. The column *betting odds* indicates if a model incorporated betting odds as covariates. The three columns *predicted H*, *predicted D* and *predicted A* present the number of predicted home wins, draws and away wins of each model respectively.

model name	model type	form parameter	γ	betting odds	predicted H	predicted D	predicted A	accuracy	RPS
xgboost_av_(-)fp_(-)bq	xgboost	No	NA	No	218	8	103	47.1125	0.2520
rf_av_(-)fp_(-)bq	randomForest	No	NA	No	234	0	95	45.2888	0.2279
ranger_av_(-)fp_(-)bq	ranger	No	NA	No	226	0	103	48.3283	0.2259
xgboost_av_(-)fp_(+)bq	xgboost	No	NA	Yes	207	6	116	51.0638	0.2299
rf_av_(-)fp_(+)bq	randomForest	No	NA	Yes	218	0	111	50.4559	0.2166
ranger_av_(-)fp_(+)bq	ranger	No	NA	Yes	222	0	107	49.8480	0.2179
xgboost_av_(+)fp_005_(-)bq	xgboost	Yes	0.05	No	217	3	109	52.5836	0.2343
rf_av_(+)fp_005_(-)bq	randomForest	Yes	0.05	No	229	0	100	48.6322	0.2247
ranger_av_(+)fp_005_(-)bq	ranger	Yes	0.05	No	229	0	100	49.2401	0.2235
xgboost_av_(+)fp_010_(-)bq	xgboost	Yes	0.10	No	221	5	103	50.4559	0.2372
rf_av_(+)fp_010_(-)bq	randomForest	Yes	0.10	No	232	0	97	48.3283	0.2248
ranger_av_(+)fp_010_(-)bq	ranger	Yes	0.10	No	230	0	99	48.6322	0.2232
xgboost_av_(+)fp_015_(-)bq	xgboost	Yes	0.15	No	220	5	104	51.0638	0.2361
rf_av_(+)fp_015_(-)bq	randomForest	Yes	0.15	No	228	0	101	47.7204	0.2262
ranger_av_(+)fp_015_(-)bq	ranger	Yes	0.15	No	227	0	102	50.7599	0.2233
xgboost_av_(+)fp_020_(-)bq	xgboost	Yes	0.20	No	214	7	108	48.9362	0.2361
rf_av_(+)fp_020_(-)bq	randomForest	Yes	0.20	No	225	0	104	49.5441	0.2262
ranger_av_(+)fp_020_(-)bq	ranger	Yes	0.20	No	226	0	103	49.8480	0.2233
xgboost_av_(+)fp_025_(-)bq	xgboost	Yes	0.25	No	214	9	106	48.0243	0.2363
rf_av_(+)fp_025_(-)bq	randomForest	Yes	0.25	No	224	0	105	48.3283	0.2255
ranger_av_(+)fp_025_(-)bq	ranger	Yes	0.25	No	231	0	98	49.2401	0.2241
xgboost_av_(+)fp_030_(-)bq	xgboost	Yes	0.30	No	220	2	107	49.2401	0.2413
rf_av_(+)fp_030_(-)bq	randomForest	Yes	0.30	No	226	0	103	48.9362	0.2246
ranger_av_(+)fp_030_(-)bq	ranger	Yes	0.30	No	233	0	96	47.7204	0.2236
xgboost_av_(+)fp_033_(-)bq	xgboost	Yes	0.33	No	211	5	113	50.1520	0.2430
rf_av_(+)fp_033_(-)bq	randomForest	Yes	0.33	No	229	0	100	49.8480	0.2242
ranger_av_(+)fp_033_(-)bq	ranger	Yes	0.33	No	223	0	106	50.1520	0.2242
xgboost_av_(+)fp_035_(-)bq	xgboost	Yes	0.35	No	225	7	97	47.4164	0.2448
rf_av_(+)fp_035_(-)bq	randomForest	Yes	0.35	No	230	0	99	47.4164	0.2256
ranger_av_(+)fp_035_(-)bq	ranger	Yes	0.35	No	220	0	109	50.5449	0.2243

Table 10: Table 1/7 displaying the settings and prediction results of all models, leveraging the number of predicted home wins, draws, away wins, accuracy and RPS for the results.

model name	model type	form parameter	γ	betting odds	predicted H	predicted D	predicted A	accuracy	RPS
xgboost_av_(+)fp_040_(-)bq	xgboost	Yes	0.40	No	223	6	100	48.6322	0.2426
rf_av_(+)fp_040_(-)bq	randomForest	Yes	0.40	No	222	0	107	50.1520	0.2257
ranger_av_(+)fp_040_(-)bq	ranger	Yes	0.40	No	223	0	106	49.5441	0.2252
xgboost_av_(+)fp_045_(-)bq	xgboost	Yes	0.45	No	217	5	107	49.5441	0.2425
rf_av_(+)fp_045_(-)bq	randomForest	Yes	0.45	No	235	0	94	47.7204	0.2256
ranger_av_(+)fp_045_(-)bq	ranger	Yes	0.45	No	225	0	104	49.5441	0.2253
xgboost_av_(+)fp_050_(-)bq	xgboost	Yes	0.50	No	217	8	104	48.6322	0.2451
rf_av_(+)fp_050_(-)bq	randomForest	Yes	0.50	No	230	0	99	48.3283	0.2235
ranger_av_(+)fp_050_(-)bq	ranger	Yes	0.50	No	233	0	96	48.3283	0.2250
xgboost_av_(+)fp_055_(-)bq	xgboost	Yes	0.55	No	215	5	109	49.8480	0.2472
rf_av_(+)fp_055_(-)bq	randomForest	Yes	0.55	No	222	0	107	47.7204	0.2251
ranger_av_(+)fp_055_(-)bq	ranger	Yes	0.55	No	221	0	108	49.5441	0.2249
xgboost_av_(+)fp_060_(-)bq	xgboost	Yes	0.60	No	212	8	109	49.2401	0.2454
rf_av_(+)fp_060_(-)bq	randomForest	Yes	0.60	No	238	1	90	47.1125	0.2265
ranger_av_(+)fp_060_(-)bq	ranger	Yes	0.60	No	231	0	98	48.3283	0.2254
xgboost_av_(+)fp_065_(-)bq	xgboost	Yes	0.65	No	219	6	104	47.7204	0.2446
rf_av_(+)fp_065_(-)bq	randomForest	Yes	0.65	No	225	0	104	48.9362	0.2241
ranger_av_(+)fp_065_(-)bq	ranger	Yes	0.65	No	224	0	105	47.1125	0.2261
xgboost_av_(+)fp_070_(-)bq	xgboost	Yes	0.70	No	215	9	105	48.9362	0.2441
rf_av_(+)fp_070_(-)bq	randomForest	Yes	0.70	No	228	0	101	47.1125	0.2261
ranger_av_(+)fp_070_(-)bq	ranger	Yes	0.70	No	229	0	100	48.6322	0.2254
xgboost_av_(+)fp_075_(-)bq	xgboost	Yes	0.75	No	220	10	99	46.5046	0.2465
rf_av_(+)fp_075_(-)bq	randomForest	Yes	0.75	No	227	0	102	48.9362	0.2272
ranger_av_(+)fp_075_(-)bq	ranger	Yes	0.75	No	229	0	100	49.2401	0.2256
xgboost_av_(+)fp_080_(-)bq	xgboost	Yes	0.80	No	224	6	99	48.9362	0.2436
rf_av_(+)fp_080_(-)bq	randomForest	Yes	0.80	No	227	0	102	48.3283	0.2268
ranger_av_(+)fp_080_(-)bq	ranger	Yes	0.80	No	224	0	105	48.9362	0.2255
xgboost_av_(+)fp_085_(-)bq	xgboost	Yes	0.85	No	211	9	109	47.4164	0.2497
rf_av_(+)fp_085_(-)bq	randomForest	Yes	0.85	No	225	0	104	47.1125	0.2271
ranger_av_(+)fp_085_(-)bq	ranger	Yes	0.85	No	226	0	103	48.6322	0.2263

Table 11: Table 2/7 displaying the settings and prediction results of all models, leveraging the number of predicted home wins, draws, away wins, accuracy and RPS for the results.

model name	model type	form parameter	γ	betting odds	predicted H	predicted D	predicted A	accuracy	RPS
xgboost_av_(+)fp_090_(-)bq	xgboost	Yes	0.90	No	228	7	104	48.6322	0.2462
rf_av_(+)fp_090_(-)bq	randomForest	Yes	0.90	No	227	0	102	49.2401	0.2267
ranger_av_(+)fp_090_(-)bq	ranger	Yes	0.90	No	226	0	103	48.3283	0.2260
xgboost_av_(+)fp_095_(-)bq	xgboost	Yes	0.95	No	230	7	92	47.1125	0.2484
rf_av_(+)fp_095_(-)bq	randomForest	Yes	0.95	No	227	0	102	47.4164	0.2271
ranger_av_(+)fp_095_(-)bq	ranger	Yes	0.95	No	229	0	100	49.8480	0.2264
xgboost_av_(+)fp_100_(-)bq	xgboost	Yes	1.00	No	215	13	101	48.6322	0.2428
rf_av_(+)fp_100_(-)bq	randomForest	Yes	1.00	No	231	0	98	46.5046	0.2273
ranger_av_(+)fp_100_(-)bq	ranger	Yes	1.00	No	229	0	100	49.2401	0.2261
xgboost_av_(+)fp_005_(+)bq	xgboost	Yes	0.05	Yes	202	9	118	51.9757	0.2286
rf_av_(+)fp_005_(+)bq	randomForest	Yes	0.05	Yes	215	0	114	50.4559	0.2169
ranger_av_(+)fp_005_(+)bq	ranger	Yes	0.05	Yes	223	0	106	50.1520	0.2157
xgboost_av_(+)fp_010_(+)bq	xgboost	Yes	0.10	Yes	204	8	117	52.2796	0.2269
rf_av_(+)fp_010_(+)bq	randomForest	Yes	0.10	Yes	224	0	105	51.9757	0.2147
ranger_av_(+)fp_010_(+)bq	ranger	Yes	0.10	Yes	218	0	111	51.0638	0.2150
xgboost_av_(+)fp_015_(+)bq	xgboost	Yes	0.15	Yes	204	10	115	52.2796	0.2267
rf_av_(+)fp_015_(+)bq	randomForest	Yes	0.15	Yes	213	0	116	50.4559	0.2174
ranger_av_(+)fp_015_(+)bq	ranger	Yes	0.15	Yes	220	0	109	52.8875	0.2151
xgboost_av_(+)fp_020_(+)bq	xgboost	Yes	0.20	Yes	210	8	111	50.7599	0.2289
rf_av_(+)fp_020_(+)bq	randomForest	Yes	0.20	Yes	223	0	106	50.7599	0.2162
ranger_av_(+)fp_020_(+)bq	ranger	Yes	0.20	Yes	220	0	109	51.6717	0.2156
xgboost_av_(+)fp_025_(+)bq	xgboost	Yes	0.25	Yes	205	5	119	51.0638	0.2304
rf_av_(+)fp_025_(+)bq	randomForest	Yes	0.25	Yes	216	0	113	50.4559	0.2160
ranger_av_(+)fp_025_(+)bq	ranger	Yes	0.25	Yes	217	0	112	51.9757	0.2154
xgboost_av_(+)fp_030_(+)bq	xgboost	Yes	0.30	Yes	214	6	109	51.3678	0.2314
rf_av_(+)fp_030_(+)bq	randomForest	Yes	0.30	Yes	218	0	111	51.0638	0.2153
ranger_av_(+)fp_030_(+)bq	ranger	Yes	0.30	Yes	221	0	108	51.0638	0.2153
xgboost_av_(+)fp_033_(+)bq	xgboost	Yes	0.33	Yes	212	6	111	51.9757	0.2314
rf_av_(+)fp_033_(+)bq	randomForest	Yes	0.33	Yes	219	0	110	50.7599	0.2167
ranger_av_(+)fp_033_(+)bq	ranger	Yes	0.33	Yes	224	0	105	51.6717	0.2150

Table 12: Table 3/7 displaying the settings and prediction results of all models, leveraging the number of predicted home wins, draws, away wins, accuracy and RPS for the results.

model name	model type	form parameter	γ	betting odds	predicted H	predicted D	predicted A	accuracy	RPS
xgboost_av_(+)fp_035_(+)bq	xgboost	Yes	0.35	Yes	208	9	112	51.9757	0.2320
rf_av_(+)fp_035_(+)bq	randomForest	Yes	0.35	Yes	219	0	110	50.4559	0.2158
ranger_av_(+)fp_035_(+)bq	ranger	Yes	0.35	Yes	223	0	106	51.3678	0.2153
xgboost_av_(+)fp_040_(+)bq	xgboost	Yes	0.40	Yes	206	5	118	52.2796	0.2313
rf_av_(+)fp_040_(+)bq	randomForest	Yes	0.40	Yes	216	0	113	50.4559	0.2152
ranger_av_(+)fp_040_(+)bq	ranger	Yes	0.40	Yes	222	0	107	51.6717	0.2155
xgboost_av_(+)fp_045_(+)bq	xgboost	Yes	0.45	Yes	215	7	107	50.4559	0.2323
rf_av_(+)fp_045_(+)bq	randomForest	Yes	0.45	Yes	226	0	103	49.8480	0.2178
ranger_av_(+)fp_045_(+)bq	ranger	Yes	0.45	Yes	220	0	109	51.3678	0.2154
xgboost_av_(+)fp_050_(+)bq	xgboost	Yes	0.50	Yes	210	10	109	50.4559	0.2276
rf_av_(+)fp_050_(+)bq	randomForest	Yes	0.50	Yes	224	0	105	49.8480	0.2167
ranger_av_(+)fp_050_(+)bq	ranger	Yes	0.50	Yes	219	0	110	51.6717	0.2153
xgboost_av_(+)fp_055_(+)bq	xgboost	Yes	0.55	Yes	213	6	110	52.5836	0.2305
rf_av_(+)fp_055_(+)bq	randomForest	Yes	0.55	Yes	220	1	108	48.6322	0.2159
ranger_av_(+)fp_055_(+)bq	ranger	Yes	0.55	Yes	226	0	103	50.4559	0.2149
xgboost_av_(+)fp_060_(+)bq	xgboost	Yes	0.60	Yes	217	8	104	51.0638	0.2292
rf_av_(+)fp_060_(+)bq	randomForest	Yes	0.60	Yes	215	0	114	50.4559	0.2180
ranger_av_(+)fp_060_(+)bq	ranger	Yes	0.60	Yes	223	0	106	51.9757	0.2152
xgboost_av_(+)fp_065_(+)bq	xgboost	Yes	0.65	Yes	214	7	108	49.2401	0.2326
rf_av_(+)fp_065_(+)bq	randomForest	Yes	0.65	Yes	223	0	106	50.7599	0.2159
ranger_av_(+)fp_065_(+)bq	ranger	Yes	0.65	Yes	217	1	111	51.3678	0.2157
xgboost_av_(+)fp_070_(+)bq	xgboost	Yes	0.70	Yes	214	8	107	51.0638	0.2171
rf_av_(+)fp_070_(+)bq	randomForest	Yes	0.70	Yes	218	0	111	51.0638	0.2157
ranger_av_(+)fp_070_(+)bq	ranger	Yes	0.70	Yes	223	0	106	51.0638	0.2331
xgboost_av_(+)fp_075_(+)bq	xgboost	Yes	0.75	Yes	207	9	113	51.0638	0.2331
rf_av_(+)fp_075_(+)bq	randomForest	Yes	0.75	Yes	219	0	110	51.0638	0.2163
ranger_av_(+)fp_075_(+)bq	ranger	Yes	0.75	Yes	222	0	107	51.6717	0.2159
xgboost_av_(+)fp_080_(+)bq	xgboost	Yes	0.80	Yes	209	7	113	51.9757	0.2321
rf_av_(+)fp_080_(+)bq	randomForest	Yes	0.80	Yes	223	0	106	50.4559	0.2174
ranger_av_(+)fp_080_(+)bq	ranger	Yes	0.80	Yes	219	0	110	51.6717	0.2160

Table 13: Table 4/7 displaying the settings and prediction results of all models, leveraging the number of predicted home wins, draws, away wins, accuracy and RPS for the results.

model name	model type	form parameter	γ	betting odds	predicted H	predicted D	predicted A	accuracy	RPS
xgboost_av_(+)fp_085_(+)bq	xgboost	Yes	0.85	Yes	206	6	117	52.2796	0.2321
rf_av_(+)fp_085_(+)bq	randomForest	Yes	0.85	Yes	224	0	105	50.4559	0.2167
ranger_av_(+)fp_085_(+)bq	ranger	Yes	0.85	Yes	219	0	110	51.9757	0.2163
xgboost_av_(+)fp_090_(+)bq	xgboost	Yes	0.90	Yes	209	7	113	51.6717	0.2293
rf_av_(+)fp_090_(+)bq	randomForest	Yes	0.90	Yes	218	1	110	52.2796	0.2153
ranger_av_(+)fp_090_(+)bq	ranger	Yes	0.90	Yes	221	0	108	51.6717	0.2158
xgboost_av_(+)fp_095_(+)bq	xgboost	Yes	0.95	Yes	213	10	106	51.9757	0.2311
rf_av_(+)fp_095_(+)bq	randomForest	Yes	0.95	Yes	223	0	106	50.7599	0.2161
ranger_av_(+)fp_095_(+)bq	ranger	Yes	0.95	Yes	221	0	108	51.6717	0.2158
xgboost_av_(+)fp_100_(+)bq	xgboost	Yes	1.00	Yes	210	7	112	52.5836	0.2274
rf_av_(+)fp_100_(+)bq	randomForest	Yes	1.00	Yes	217	0	112	50.4559	0.2163
ranger_av_(+)fp_100_(+)bq	ranger	Yes	1.00	Yes	222	0	107	51.0638	0.2165
xgboost_only_fp_005	xgboost	Yes	0.05	No	234	25	120	46.4380	0.2246
rf_only_fp_005	randomForest	Yes	0.05	No	208	51	120	43.5356	0.2487
ranger_only_fp_005	ranger	Yes	0.05	No	225	30	124	45.3826	0.2321
xgboost_only_fp_010	xgboost	Yes	0.10	No	222	34	123	46.4380	0.2220
rf_only_fp_010	randomForest	Yes	0.10	No	197	59	123	45.3826	0.2443
ranger_only_fp_010	ranger	Yes	0.10	No	211	42	126	46.9657	0.2297
xgboost_only_fp_015	xgboost	Yes	0.15	No	251	22	106	45.6464	0.2320
rf_only_fp_015	randomForest	Yes	0.15	No	210	51	118	43.2718	0.2540
ranger_only_fp_015	ranger	Yes	0.15	No	219	46	114	44.8549	0.2363
xgboost_only_fp_020	xgboost	Yes	0.20	No	243	31	105	46.7018	0.2284
rf_only_fp_020	randomForest	Yes	0.20	No	203	70	106	41.4248	0.2505
ranger_only_fp_020	ranger	Yes	0.20	No	223	44	112	44.0079	0.2315
xgboost_only_fp_025	xgboost	Yes	0.25	No	224	30	125	44.8549	0.2282
rf_only_fp_025	randomForest	Yes	0.25	No	186	63	130	43.2718	0.2516
ranger_only_fp_025	ranger	Yes	0.25	No	199	45	135	43.7995	0.2365
xgboost_only_fp_030	xgboost	Yes	0.30	No	225	28	126	45.6464	0.2283
rf_only_fp_030	randomForest	Yes	0.30	No	177	71	131	43.5356	0.2512
ranger_only_fp_030	ranger	Yes	0.30	No	193	50	136	46.7018	0.2312

Table 14: Table 5/7 displaying the settings and prediction results of all models, leveraging the number of predicted home wins, draws, away wins, accuracy and RPS for the results.

model name	model type	form parameter	γ	betting odds	predicted H	predicted D	predicted A	accuracy	RPS
xgboost_only_fp_033	xgboost	Yes	0.33	No	243	22	114	45.1187	0.2301
rf_only_fp_033	randomForest	Yes	0.33	No	200	59	120	43.5356	0.2477
ranger_only_fp_033	ranger	Yes	0.33	No	209	46	124	46.1741	0.2321
xgboost_only_fp_035	xgboost	Yes	0.35	No	230	31	118	44.0633	0.2291
rf_only_fp_035	randomForest	Yes	0.35	No	198	58	123	45.3826	0.2459
ranger_only_fp_035	ranger	Yes	0.35	No	210	48	121	46.9657	0.2324
xgboost_only_fp_040	xgboost	Yes	0.40	No	238	23	118	44.8549	0.2357
rf_only_fp_040	randomForest	Yes	0.40	No	209	50	120	43.0079	0.2554
ranger_only_fp_040	ranger	Yes	0.40	No	220	38	121	45.6464	0.2410
xgboost_only_fp_045	xgboost	Yes	0.45	No	233	29	117	40.6332	0.2378
rf_only_fp_045	randomForest	Yes	0.45	No	213	54	112	40.8971	0.2671
ranger_only_fp_045	ranger	Yes	0.45	No	219	38	122	43.0079	0.2449
xgboost_only_fp_050	xgboost	Yes	0.50	No	226	28	125	38.5224	0.2524
rf_only_fp_050	randomForest	Yes	0.50	No	188	73	118	35.3562	0.2707
ranger_only_fp_050	ranger	Yes	0.50	No	200	57	122	35.8839	0.2488
xgboost_only_fp_055	xgboost	Yes	0.55	No	229	26	124	42.7441	0.2409
rf_only_fp_055	randomForest	Yes	0.55	No	194	64	121	35.8839	0.2690
ranger_only_fp_055	ranger	Yes	0.55	No	207	56	116	40.1055	0.2462
xgboost_only_fp_060	xgboost	Yes	0.60	No	252	26	101	43.5356	0.2380
rf_only_fp_060	randomForest	Yes	0.60	No	191	61	127	38.7863	0.2645
ranger_only_fp_060	ranger	Yes	0.60	No	199	48	132	41.6887	0.2458
xgboost_only_fp_065	xgboost	Yes	0.65	No	223	29	127	42.7441	0.2409
rf_only_fp_065	randomForest	Yes	0.65	No	202	49	128	40.3694	0.2739
ranger_only_fp_065	ranger	Yes	0.65	No	219	31	129	41.9525	0.2546
xgboost_only_fp_070	xgboost	Yes	0.70	No	260	23	96	39.5778	0.2515
rf_only_fp_070	randomForest	Yes	0.70	No	224	46	109	39.0501	0.2685
ranger_only_fp_070	ranger	Yes	0.70	No	239	37	103	39.5778	0.2553
xgboost_only_fp_075	xgboost	Yes	0.75	No	251	26	102	37.9947	0.2497
rf_only_fp_075	randomForest	Yes	0.75	No	229	44	106	39.3140	0.2776
ranger_only_fp_075	ranger	Yes	0.75	No	233	39	107	38.5224	0.2596

Table 15: Table 6/7 displaying the settings and prediction results of all models, leveraging the number of predicted home wins, draws, away wins, accuracy and RPS for the results.

model name	model type	form parameter	γ	betting odds	predicted H	predicted D	predicted A	accuracy	RPS
xgboost_only_fp_080	xgboost	Yes	0.80	No	250	24	105	40.8971	0.2454
rf_only_fp_080	randomForest	Yes	0.80	No	203	63	113	36.9393	0.2722
ranger_only_fp_080	ranger	Yes	0.80	No	234	43	102	39.3140	0.2552
xgboost_only_fp_085	xgboost	Yes	0.85	No	245	28	106	40.1055	0.2498
rf_only_fp_085	randomForest	Yes	0.85	No	224	42	113	41.1609	0.2599
ranger_only_fp_085	ranger	Yes	0.85	No	231	33	115	41.6887	0.2429
xgboost_only_fp_090	xgboost	Yes	0.90	No	236	30	113	41.9525	0.2443
rf_only_fp_090	randomForest	Yes	0.90	No	214	53	112	41.9525	0.2677
ranger_only_fp_090	ranger	Yes	0.90	No	221	39	119	44.3272	0.2468
xgboost_only_fp_095	xgboost	Yes	0.95	No	252	32	95	39.8417	0.2490
rf_only_fp_095	randomForest	Yes	0.95	No	207	75	97	40.3694	0.2733
ranger_only_fp_095	ranger	Yes	0.95	No	234	52	93	42.7441	0.2549
xgboost_only_fp_100	xgboost	Yes	1.00	No	336	0	43	41.9525	0.2409
rf_only_fp_100	randomForest	Yes	1.00	No	338	0	41	41.6887	0.4313
ranger_only_fp_100	ranger	Yes	1.00	No	338	0	41	41.6887	0.2370

Table 16: Table 7/7 displaying the settings and prediction results of all models, leveraging the number of predicted home wins, draws, away wins, accuracy and RPS for the results.