Citation for published version:
Walsh, C & Joshi, A 2023 'Machine learning for sports betting: should forecasting models be optimised for accuracy or calibration?' arXiv. https://doi.org/10.48550/arXiv.2303.06021

DOI:
10.48550/arXiv.2303.06021

Publication date:
2023

Document Version
Early version, also known as pre-print

Link to publication

Publisher Rights
CC BY-NC-ND

University of Bath

Alternative formats
If you require this document in an alternative format, please contact:
openaccess@bath.ac.uk

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 04. Nov. 2024

arXiv:2303.06021v2 [cs.LG] 13 Mar 2023

Machine learning for sports betting: should forecasting models be optimised for accuracy or calibration?

Conor Walsh[a,*], Alok Joshi[a]

[a]*Department of Computer Science, University of Bath, Bath, United Kingdom*

Abstract

Sports betting's recent federal legalisation in the USA coincides with the golden age of machine learning. If bettors can leverage data to accurately predict the probability of an outcome, they can recognise when the bookmaker's odds are in their favour. As sports betting is a multi-billion dollar industry in the USA alone, identifying such opportunities could be extremely lucrative. Many researchers have applied machine learning to the sports outcome prediction problem, generally using accuracy to evaluate the performance of forecasting models. We hypothesise that for the sports betting problem, model calibration is more important than accuracy. To test this hypothesis, we train models on NBA data over several seasons and run betting experiments on a single season, using published odds. Evaluating various betting systems, we show that optimising the forecasting model for calibration leads to greater returns than optimising for accuracy, on average (return on investment of 110.42% versus 2.98%) and in the best case (902.01% versus 222.84%). These findings suggest that for sports betting (or any forecasting problem where decisions are made based on the predicted probability of each outcome), calibration is a more important metric than accuracy. Sports bettors who wish to increase profits should therefore optimise their forecasting model for calibration.

Keywords:
NBA game outcome prediction, Kelly criterion, Probabilistic classifiers, Error measures, Decision making, Model selection, Evaluating forecasts, Value bet

1. Introduction

Sports betting in the US is conservatively estimated to be a $150 billion industry (legalsportsbetting, 2022). As a result of the worldwide interest in US sports, an abundance of data is publicly available. Much research has been done on the use of machine learning (ML) for sports outcome prediction. The success of this research has turned this data into an invaluable commodity to sports bettors around the world. If a bettor can leverage data to accurately estimate the true probability of a sporting outcome, they can identify a bookmaker's mispricing of this outcome - and whether or not there is an opportunity to make a profit. The development of a proficient forecasting model could therefore prove extremely lucrative.

The National Basketball Association (NBA) in North America is the world's premier basketball league. This paper focuses on the development of a data-driven betting system in the case of the NBA. ML for sports outcome prediction has been widely studied, however very little of this research extends to sports betting. In the bulk of this work, ML models are evaluated on accuracy achieved. Accuracy is a suitable model evaluation metric for the sports outcome prediction problem, as the goal is to correctly predict the winning team (Bunker and Thabtah, 2019). However this is not necessarily true for the sports betting problem, where the goal is to estimate the true probability of the sporting outcome to identify and profit from any mispricing of the odds offered on this outcome. As *calibration* is used to estimate how close a model's predicted probabilities are to the true probabilities, we hypothesise that calibration is

*Corresponding author
Email address: conorwalsh206@gmail.com (Conor Walsh)

a more appropriate metric than accuracy for the sports betting problem, and that optimising the predictive model for calibration rather than accuracy leads to greater profit generation.

The most uncomplicated way to test this hypothesis is to consider the most straightforward form of a wager – the moneyline bet. To win a moneyline bet, the bettor must predict the winner of the game (Hubáček et al., 2019). If the bettor correctly predicts the winner, they win back the wager (also called the stake) along with the profit, otherwise, they lose the wager to the bookmaker (Hubáček et al., 2019). The profit is a predetermined quantity that corresponds to the odds. Decimal odds display the total return (stake plus profit) that a wager of a single unit could yield (Cortis, 2015, 2016). Taking the inverse of the odds results in the *implied probability* of the outcome occurring (Cortis, 2015, 2016). Let us assume that π_i represents the bookmaker's implied probability of outcome i. Then the odds are given as $1/\pi_i$.

For fair odds, the implied probability represents the bookmaker's estimate of the probability of that outcome occurring (Hubáček et al., 2019). In practice, the odds are never fair as the bookmaker wants to maximise profits. They deviate from the 'fair price' by the bookmaker's *margin*. This is the absolute difference between 1 and the sum of the implied probabilities, and can be viewed as a commission charged to bettors by the bookmaker (Hubáček et al., 2019). Unsurprisingly, the probability of each outcome of a coin flip is equally likely, so the fair odds for each outcome should be 2.0. A bookmaker might offer odds of 1.90 for each so the sum of implied probabilities would be $(1/1.90) + (1/1.90) \approx 1.0526$, which suggests that the bookmaker has factored in a margin of approximately 5%. When the true probability of an outcome occurring is greater than the implied probability, the expected value of the bet is positive. We define such a bet as a value bet (Edwards, 1955). Value bets arise when the bookmaker misprices the odds. To spot such opportunities, bettors can compare their forecasting model's predicted probability to the bookmaker's odds. We follow this approach and implement and evaluate several betting systems which aim to identify value bets and capitalise on them. We measure the success of each system by the return on investment (ROI) achieved over the course of an NBA season.

The forecasting model is perhaps the most important part of a sports betting system, and there are many algorithms available to bettors (Zdravevski and Kulakov, 2010; Cao, 2012; Zimmermann et al., 2013; Alonso and Babac, 2022; Cheng et al., 2016; Tran, 2016; Pai et al., 2017). Prominent examples used in research include support vector machines (SVM), logistic regression (LR), Naive-Bayes, K-nearest neighbours (kNN), decision trees and neural networks (Hamadani, 2006; Miljković et al., 2010; Loeffelholz et al., 2009). Although many predictive algorithms have already been explored, there has still been a distinct lack of exploration of model evaluation metrics. In the vast majority of this research, models have been optimised for and evaluated based on accuracy alone. As accuracy may not be the most appropriate metric for the sports betting problem, the lack of alternative metrics to evaluate model performance is a key gap in the literature. We therefore design two groups of betting systems, one making use of a predictive model that is optimised for accuracy, and the other group a predictive model that is optimised for calibration. Comparing the returns achieved by each group of betting systems, we can determine whether optimising the predictive model for accuracy or calibration leads to greater profit generation. One of the interesting findings of our work is that, on average, and in the best case scenario, betting systems using a forecasting model that is optimised for calibration earn greater profits than those using a forecasting model that is optimised for accuracy.

The remainder of this paper is organised as follows. Related works are discussed in section 2. Section 3 involves a statement of the central hypothesis of the paper and explores the reasons behind the authors' arrival at this hypothesis. Section 4 lays out the design of an experiment to test this hypothesis. Section 5 covers the novel feature engineering carried out ahead of the predictive modelling, which is covered in section 6. Section 7 discusses the betting experiments and the algorithm used to conduct the betting simulations in detail. In section 8, results of the experiments are examined. Finally, section 9 discusses the implications of our findings and concludes the paper.

2. Related Work

While analysing data from the National Football League (NFL) to understand bettor and bookmaker strategies, Levitt and colleagues identified various approaches used by bookmakers to generate profits (Levitt, 2004). The first approach relies on the bookmaker's ability to anticipate the price which equalises the quantity of money wagered on

each side of the bet (Levitt, 2004). If done successfully, the losers compensate the winners while the bookmaker collects the margin. If the bookmaker doesn't get this right they risk having to reach into their own reserves to compensate the winners. A second approach involves the bookmaker being able to systematically outperform bettors in game outcome prediction. This allows the bookmaker to set the 'correct' price. While the money wagered is not equalised on any given game, this approach sees the bookmaker profit off the margin on average over the course of the season (Levitt, 2004). To increase profits, the bookmaker may combine the previous two approaches and set the 'wrong' price on purpose to exploit bettor preferences. However, if the odds deviate too much from the true price, shrewd bettors aware of the 'correct' price can capitalise on this, and make a profit (Levitt, 2004). Levitt found that bookmakers generally focus on outperforming the average bettor in outcome forecasting (Levitt, 2004). This leaves bettors with an opportunity to generate positive returns if they can identify when the bookmaker's price is wrong.

In order to identify value bets, bettors must first possess a reliable forecasting model. As the aim of sports outcome prediction is to predict the outcome of a sporting event given a finite set of possibilities, it is usually addressed as a classification problem (Horvat and Job, 2020). To quantify the performance of classifiers in such settings, accuracy is generally used as the model evaluation metric, where accuracy refers to the proportion of correctly classified data (Horvat and Job, 2020; Yang and Shami, 2020). In their work, Bunker and Thabtah deemed this appropriate, noting 'classification accuracy is a reasonable measure of evaluation' for the sports outcome prediction problem (Bunker and Thabtah, 2019). Many different classifiers have been used to address this problem with neural networks among the most widely used (Horvat and Job, 2020). Other classifiers commonly used for the sports outcome prediction problem include SVM and kNN, both of which are widespread in baseball forecasting (Zhang, 2000; Horvat and Job, 2020). When it comes to designing a robust sports forecasting model, other key considerations one must take into account include the features to use, as well as methods of evaluating the model's performance. A common approach for model evaluation is chronological data segmentation, i.e. using a training set made up of seasons prior to those in the evaluation set (Horvat and Job, 2020, 2019). For sports prediction, it is critically important to preserve the chronological order of the training data. This is done to ensure that upcoming matches are predicted based on data from past matches only. As cross-validation usually involves shuffling the order of the instances, it is not recommended in the sports prediction setting (Bunker and Thabtah, 2019; Horvat and Job, 2020). In general, feature selection and feature extraction are also employed to reduce the dimensionality and complexity of the classification problem (Horvat and Job, 2020).

Much of this work has focused on basketball, likely due to the abundance of publicly available data (Sports-Reference-LLC, 2022; NBA, 2023; databasketball, 2023; basketballgeek, 2023). Researchers have made efforts to identify the most important features and best-performing classifiers, in addition to investigating whether the use of player or team-level features achieves superior results. Notably, Ivankovic found the most important features to be, in order of diminishing importance, defensive rebounds, two-point and three-point shots, steals, turnovers, offensive rebounds, free-throw shots, blocks and assists (Ivanković et al., 2010). In terms of identifying the best performing classifiers, Cao and colleagues trained several models on NBA data from the 2005/2006-2009/2010 seasons, and evaluated these models on the 2010/2011 season (Cao, 2012). In this study, LR was found to be the best-performing model with an accuracy of 69.97%, outperforming multi-layer perceptron (MLP) and SVM models (which achieved accuracies of 68% and 67.7%, respectively) (Cao, 2012). In contrast, Torres found an MLP model to be superior to LR in the same setting, with accuracies of 68.44% and 67.98%, respectively (Torres and Hu, 2013). In a separate study, Lin et al discovered that a team's win/loss record is a crucial predictor of victory (Lin et al., 2014). Leveraging the impressive predictive capability of neural networks, Hubáček and colleagues constructed a neural network that used a convolutional layer to summarise player-level features into team-level features. By considering only high-confidence predictions, this classifier achieved an accuracy of 84.35%, compared to an accuracy of 80% achieved by a neural network that only used team-level features. Despite the success researchers have achieved, practical implementation of such forecasting systems poses several limitations, as pointed out by Ganguly and Frank, including (*i*) lack of context, (*ii*) no measure of uncertainty of the prediction and (*iii*) lack of benchmark datasets to compare results (Ganguly and Frank, 2018). Another common limitation is that many forecasting models do not take in-game events into account. Events such as the early injury of a star player can have a significant influence on the outcome of a game (Horvat and Job, 2020). Despite this limitation, we focus on pre-game betting based on the closing odds. Naturally, considering in-game events would require the ability to process streaming data, whereas forecasting for pre-game

betting can be done using batch processing, and resources for batch processing are much more readily available to the average bettor (Pfandzelter and Bermbach, 2019).

For a gambler seeking to maximise profits over a series of successive bets (with the opportunity to reinvest the winnings), the size of each bet can be determined using the *Kelly criterion* (Kelly Jr, 2011). The criterion identifies the optimal bet size by maximising the expected value of the logarithmic growth of wealth (Hsieh and Barmish, 2015). While its origins lie in the analysis of long-distance telephone signal noise, the equation has found widespread application across many domains (Dotan, 2020; Rotando and Thorp, 1992; Thorp, 2008, 1975; Barnett, 2010). In essence, sports bettors can use this criterion to calculate optimal bet size as a proportion of overall bankroll. They simply require the bookmaker's odds, and the probability of victory according to their predictive model. Mathematically, the Kelly criterion can be defined as shown in (1):

$$k = \frac{pb - q}{b} \tag{1}$$

where:

- k is the proportion of the bettor's bankroll to wager on the given outcome
- p is the probability of the given outcome occurring
- b represents the potential winnings on a wager of 1 unit i.e. $odds - 1$
- q is the probability of the given outcome not occurring, i.e. $q = 1 - p$ (Dotan, 2020)

Despite the criterion's reputation as the optimal strategy for resource allocation on a set of gambles repeated over time, relying on it to decide bet size has certain limitations (Hsieh and Barmish, 2015). For instance, the criterion often suggests wagering a very large proportion of the overall bankroll on a single game, which is a recipe for disaster in a realm as unpredictable as the world of sport. The Kelly strategy in this form therefore leads to almost sure ruin (Hsieh and Barmish, 2015). In contrast, the *fractional Kelly* is a less risky variation of the strategy. This variation uses the same formula, but here k represents the proportion of a *fraction* of the overall bankroll. Thus, implementing the quarter-Kelly would mean for $k = 0.4$, instead of wagering 40% of their bankroll, the bettor wagers 10%. Recently, Dotan illustrated the utility of the fractional kelly in NBA betting markets. Notably, in a betting simulation spanning a single season, the '5^{th} Kelly' strategy earned an ROI of over 98%, while its full-Kelly counterpart crashed to zero (Dotan, 2020).

Excluding accuracy, the lack of metrics used to evaluate the performance of sports forecasting models is a noticeable gap in the literature (Horvat and Job, 2020). While it may be suitable for the sports outcome prediction problem, accuracy alone is not a sufficient model evaluation metric for the sports betting problem. To compensate for this, Hubáček and colleagues designed a loss function to penalise correlation with the bookmaker's odds (Hubáček et al., 2019). One of the key findings of their work was that training models under this loss resulted in greater profits than optimising for accuracy (Hubáček et al., 2019). Further, it has been observed that a highly accurate predictive model is useless as long as it coincides with the bookmaker's model (Hubáček et al., 2019).

Combining these findings with the aforementioned gap in the literature leads to the central hypothesis of this paper, which is discussed in the next section.

3. Central Hypothesis

The purpose of the forecasting model is to predict the probability of victory for each team in a given game, so that these probabilities can be compared to the bookmaker's odds to determine if a value bet is on offer for either team. The forecasting model also plays a vital role in deciding how much to bet, as this is a function of the predicted probability and the bookmaker's odds according to the Kelly criterion. Therefore, for a Kelly-based betting system to be successful, it is critically important that the probability of victory generated by the forecasting model is close

to the true probability of victory. This notion, that the probability a classifier assigns to an event should reflect the true frequency of that event, relates to the concept of *calibration* (Kumar et al., 2019). In contrast to calibration, the fundamental problem with accuracy in this setting is that it does not take into account the distance between the predicted probability of victory and the true probability of victory. Accuracy simply measures the proportion of correctly classified data, where for any given game, the team whose predicted probability of victory is greater than 0.5 is predicted to be the winner. As we want our model's predicted probability to be as close as possible to the true probability, for a team with a true probability of victory of 60%, we would favour a model which assigns them a predicted probability of victory of 55% over one which assigns them a probability of victory of 70%. However, using accuracy as a model evaluation metric is equivalent to treating these predictions just the same. The following example serves to illustrate the danger of using accuracy to evaluate model performance when the predicted probability is of more interest than the predicted label. Imagine there exists a set of 100 games in each of which the forecasting model assigns the home team a 60% chance of victory. The official prediction of the model for each game would be that the home team will win. If the home team did in fact win each of those games the forecasting model would achieve an accuracy of 100%. If these predicted probabilities were roughly equivalent to the true probabilities, the chance of this occurring is approximately zero ($0.6^{100} \approx 0$). The fact that it did occur implies that it is unlikely the model's predicted probabilities were reasonable. If accuracy was used to select the best-performing model, this model would be selected, even though its predicted probabilities are unlikely to be close to the true probabilities. This model could be described as perfectly accurate, but *uncalibrated*, as its predicted probabilities were not closely aligned with the true probabilities, despite the fact that it correctly predicted the winner of each game. Instead of accuracy, we desire a metric that provides an indication of the distance between the predicted probabilities and the true probabilities.

To discuss calibration formally, let us consider the problem of multiclass classification. Suppose we have input $X \in \mathbb{X}$ and label $Y \in \mathbb{Y} = \{1, ..., K\}$ which are random variables with ground truth joint distribution $\pi(X, Y) = \pi(Y|X)\pi(X)$. Then our classifier is of the form $f(X) = (\hat{Y}, \hat{P})$, where $\hat{Y}$ is the predicted label and $\hat{P}$ is the probability associated with the prediction. The probabilistic classifier is said to be well-calibrated if, among test instances assigned a predicted probability vector $\hat{P}$, the class distribution is (approximately) distributed as $\hat{P}$ (Kull et al., 2019). In the previously discussed example, the forecasting model would be considered well-calibrated if approximately 60 of the 100 games were won by the home team. Perfect calibration occurs if

$$\mathbb{P}(\hat{Y} = Y|\hat{P} = p) = p, \forall p \in [0, 1] \tag{2}$$

with reference to the probability over the joint distribution (Guo et al., 2017).

Many methods have been proposed to measure calibration. We use the **Classwise Expected Calibration Error** (classwise-ECE), as this variation overcomes some of the limitations of the original ECE (Kull et al., 2019). To calculate the classwise-ECE, the interval $[0, 1]$ is split into M bins of equal length so that the m^{th} bin is the interval $[\frac{m-1}{M}, \frac{m}{M})$. For a given class k, each prediction in the set is grouped into the bin its probability lies within, i.e. we associate $\hat{P}_k(x)$ with the j^{th} bin if $\hat{P}_k(x) \in [\frac{j-1}{M}, \frac{j}{M})$. Let $B_{j,k}$ represent the j^{th} bin for predicted probabilities relating to class k. The classwise-ECE is defined as shown in (3):

$$classwise - ECE = \frac{1}{k} \sum_{i=1}^{k} \sum_{j=1}^{m} \frac{|B_{j,i}|}{n} |y_i(B_{j,i}) - \bar{p}_i(B_{j,i})| \tag{3}$$

where k is the number of classes in the problem, m is the number of bins used, n is the size of the dataset, $|B_{j,i}|$ is the size of the bin (number of class i predictions associated with the j_{th} bin), $y_i(B_{j,i})$ is the actual rate of occurrence of class i in $B_{j,i}$ and $\bar{p}_i(B_{j,i})$ denotes the average probability of class i predictions in $B_{j,i}$ (Kull et al., 2019). This error is bounded with lower limit 0 and upper limit 1, and can be thought of as the percentage by which the model's predicted probability deviates from the true probability, on average. We note that for binary classification, classwise calibration is equivalent to class-specific calibration focusing on the positive class, since the predicted probability of the negative class is determined by the predicted probability of the positive class (Posocco and Bonnefoy, 2021). We can equivalently state that class i's contribution to the classwise-ECE is equal for both classes in the binary case (due to symmetry), and so we can simply calculate the loss for just the positive class, as this is equivalent to the average loss of the two classes.

One of the limitations of the traditional expected calibration error is that it focuses only on the probability of the most likely class, and for each prediction, ignores calibration with respect to the $K - 1$ other classes (Nixon et al., 2019). While the classwise-ECE overcomes this, a concerning limitation still exists - one can obtain almost perfectly calibrated probabilities by predicting the overall class distribution for all instances (Kull et al., 2019). To avoid this scenario, we impose a constraint on model predictions - we require the distribution of weights over the bins to be platykurtic. This ensures that the distribution is not too highly concentrated around the mean.

Having introduced the concept of calibration, we arrive at the central hypothesis of this paper, that it is more important for a classifier to be well-calibrated than highly accurate. Therefore, optimising the forecasting model for calibration rather than accuracy should allow for greater profit generation. We design an experiment to test this hypothesis, as discussed in the next section.

4. Experiment Design

In a data-driven sports betting system, optimising the forecasting model for calibration should lead to greater profit generation than optimising for accuracy. This is the idea at the core of this paper. To test this notion, we design an experiment consisting of the following steps. First, we begin with a set of candidate forecasting models (logistic regression, random forest, support vector machines, multi-layer perceptron). Next, we construct a predictive modelling pipeline (see figure 1) that optimises the models through feature selection and hyperparameter-optimisation, prior to selecting the best-performing model. This pipeline consists of two branches, one optimising for accuracy, the other optimising for calibration. We optimise for accuracy by maximisation of accuracy, we optimise for calibration by minimisation of the classwise-ECE (using 20 bins). Finally, the models are evaluated on a test set to select the best-performing model under each metric. The final output of the pipeline consists of a model from each branch - the most accurate model from the accuracy branch, and the most well-calibrated model from the calibration branch.

Figure 1: Predictive Modelling Pipeline: Two branches of the pipeline are presented. The first branch optimises the forecasting model for accuracy through feature selection and hyperparameter optimisation using accuracy as the evaluation criterion, prior to selecting the most accurate model. The second branch optimises the forecasting model for calibration through the same processes, selecting the most well-calibrated model. These branches take NBA data and a set of candidate forecasting models as inputs. The output of each branch is the optimal forecasting model under the given metric. Here the blue cylinders represent ML processes and the yellow rectangles represent inputs and outputs of these processes.

The two final models are used to generate predictions for each game in an NBA season. Then, several betting systems are implemented. Typically, a betting system has two components: a strategy and a rule.

- The betting strategy decides whether or not to place a bet
- The betting rule decides the size of the bet (Dotan, 2020)

For each game, these decisions are made by comparing the model's predicted probabilities to the bookmaker's odds. Ultimately, the betting systems are evaluated by their ROI, where the ROI is the percentage change in the bettor's initial bankroll by the end of the season.

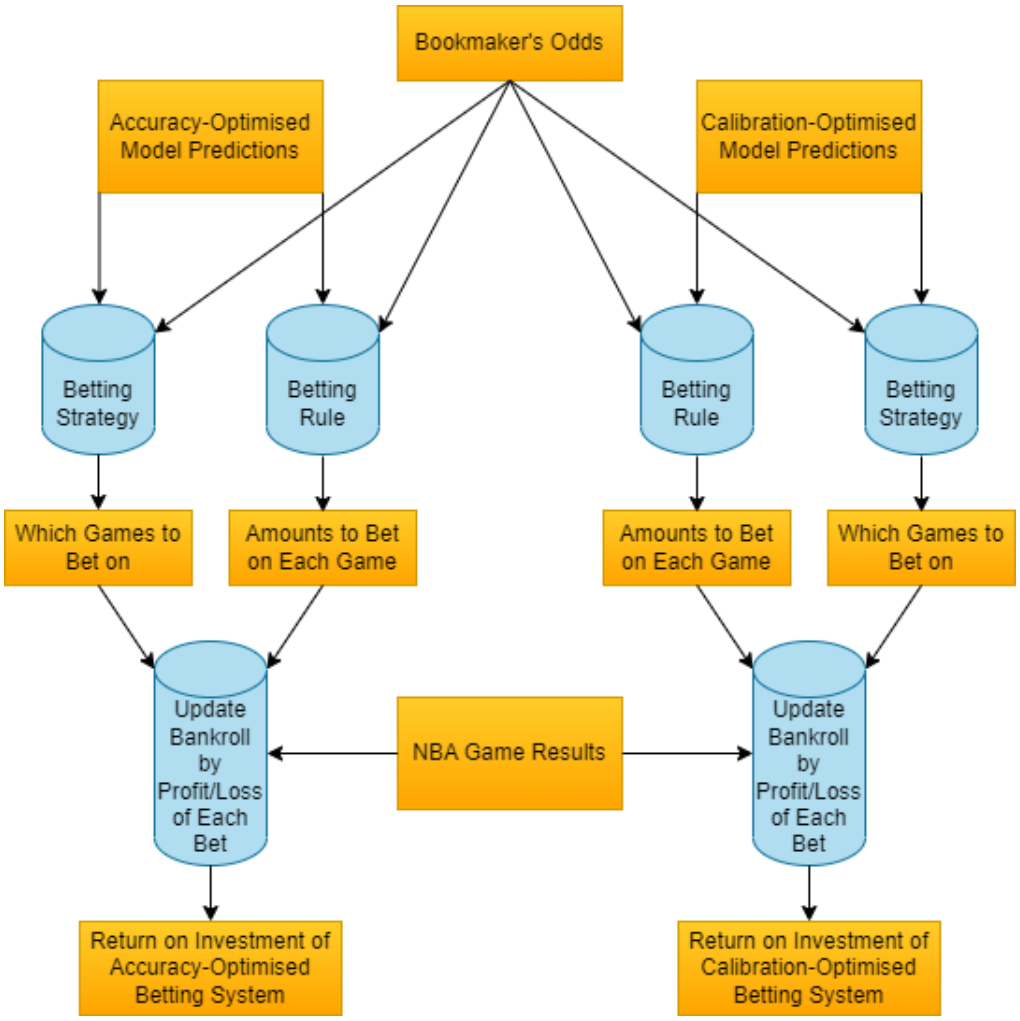

Figure 2: Betting Simulation Pipeline: Two branches of the pipeline are presented. The first branch represents the simulation for a bettor using a forecasting model optimised for accuracy, while the second branch represents the simulation for a bettor using a forecasting model optimised for calibration. These branches take as inputs a set of predictions, the bookmaker's odds and the results of each game in the given NBA season. For a given combination of strategy and rule, the output of each branch is the return on investment achieved by the bettor. The meaning of each shape is given in the caption of figure 1.

To determine whether or not our hypothesis holds true, there are two key questions to consider. This experiment is designed to answer both. The first question is focused on the theoretical aspect of our hypothesis, while the second addresses practical implications for bettors. The first questions asks: (*i*) In a data-driven sports betting system, does optimising the forecasting model for calibration, rather than accuracy, allow for greater profit generation? To answer this, we compare the ROI achieved by each betting system using a forecasting model optimised for calibration, to the ROI achieved by its accuracy-optimised counterpart. Afterwards, conducting a paired t-test allows us to determine whether or not the mean ROI of the calibration-optimised group of betting systems is greater than the mean ROI of the accuracy-optimised group. The second questions asks: (*ii*) Which betting system generates the greatest returns, and does it use an accuracy-optimised or calibration-optimised forecasting model? To answer this, we compare the ROI achieved by each betting system, and identify the greatest. More often than not, bettors do not care about the philosophical implications of the choice of metric used to evaluate the forecasting model, they simply want to know which betting system will award them the greatest returns. For a bettor deciding upon a single system to use over a season, the answer to this second question reveals whether that system's forecasting model should be optimised for calibration, or accuracy.

We have so far assumed there exists a dataset on which to train and test our models, with readily available features. However, this is not the case. A crucial, preliminary step of the experiment is feature engineering.

5. Feature Engineering

NBA games cannot end in a draw - in the case of a tie, successive overtime periods are played until there is a winner. This makes forecasting the outcome of NBA games a binary classification problem where the aim is to predict the winning team. Many researchers have successfully applied supervised learning to this problem, largely constructing their features using box score statistics. A comprehensive list of basic and advanced box score statistics is provided in appendix B (see tables B.16 and B.17).

A common approach to construction of the feature set is to use box score statistics averaged over the season to date as predictor variables for each game (e.g. average blocks per game) (Hubáček et al., 2019). This approach considers how well a team performs on average in a particular aspect of the sport. However, considering absolute figures like this can leave the data prone to shift, as the characteristics of the league may change over time (Dutta et al., 2017). Therefore, to construct our team-level features, we average the differences in team-total box score statistics versus previous opponents over the season to date. This approach considers the amount by which a team *outperforms* their opponent on average in each particular aspect of the sport, as these relative figures are less prone to shift (Dutta et al., 2017). We then perform feature extraction by taking the difference of the home and away teams' 'average out-performance values' for each box score statistic, to reduce the dimensionality of the data. To demonstrate the construction of such features, we provide a hypothetical calculation using synthetic data in appendix B(see section B.1).

Importantly, these team-level features do not take into account the influence of individual players. For instance, if a star player is injured, it will likely influence the outcome of the game. This is also true for the absence of a strong rebounder, as defensive rebounds is a strong predictor of victory (Ivanković et al., 2010). We therefore include player-level features to account for the influence of star players and strong rebounders. To avoid significantly increasing the dimensionality of the data, we carefully select just two player-level box score statistics to include in the model, and apply a number of transformations to these statistics to produce two novel features that capture the influence of individual players.

To account for star players, we consider the 'Box Plus/Minus' statistic. This is a rating for individual players which is adjusted for the strength of their team, and one in which star players generally score much higher than average players (see table B.17). To account for the influence of strong rebounders, we consider 'Defensive Rebound Percentage'. To include these as features for a given game, for each statistic we find each starting player's average value over the season to date. We then take the mean of these values for each team, before taking the difference between the home and away teams' resulting values. This final figure is then used as the value of the feature. These features allow the model to reflect the impact of changes in the team sheet.

Interestingly, Lin and colleagues found that box score statistics alone may not provide enough information for accurate prediction, and concluded that using a team's win/loss record can improve accuracy (Lin et al., 2014). Therefore, we include another feature in our model - each team's regular season winning percentage from the previous year. For each game, we take the difference between the home and away team's winning percentages from the previous season, and denote this feature 'Previous season record'. To ensure each prediction is sufficiently well-informed, we exclude the first 10 games of each team in each season from the training instances, and use them only in the calculation of later games' features (Hubáček et al., 2019).

Our final feature engineering step is feature standardisation. For each feature, this involves subtracting the mean and dividing by the standard deviation so that it is distributed with a mean of 0 and standard deviation of 1 (Labayen et al., 2020). This is done to ensure all features are on the same scale, as models that are smooth functions of the input are affected by the scale of the input, and we do not want the range of a feature's values to dictate the influence it has on the model (Zheng and Casari, 2018). Generally, both the training and test set features are scaled using the distribution of the training set. This approach assumes these distributions are approximately equal. If this is not the case (a phenomenon known as covariate shift), it may yield inaccurate results (Sugiyama et al., 2007). To detect

covariate shift, we compare the distribution of each feature in a validation set to its distribution in an initial training set which is composed of the seasons prior to the validation set. If these are found to be different, the feature is dropped. To test if two samples are drawn from the same (unknown) distribution, the two-sample Kolmogorov-Smirnov test is used (Pratt and Gibbons, 2012). Using an initial training set and a validation set, we carry out these tests at the 1% level of significance, as strong evidence of covariate shift is necessary for a feature to be dropped. All remaining features are then standardised - using their own distribution for training sets, and using the distribution across all prior seasons for validation, test, or betting simulation sets. With feature set constructed, we have finalised the NBA data required by our predictive modelling pipeline. This represents one half of the input for the pipeline. The other half of the input, as well as all pipeline processes and outputs, are discussed in the next section.

6. Predictive Modelling

The final input component of the predictive modelling pipeline is the set of candidate forecasting models. We explore these probabilistic classifiers in detail below.

6.1. Candidate Forecasting Models

The first candidate forecasting model is LR. Suppose we have a dataset with n samples $D = \{(X_1, y_1), ..., (X_n, y_n)\}$, where each sample has d features i.e. $X_i = (X_{i1}, ..., X_{id})$ and a corresponding response y_i. Given input X_i, LR gives the probability of the sample belonging to class C by (4).

$$P(Y_i = C|X_i) = f(X_i\beta) = \frac{\exp(X_i\beta)}{1 + \exp(X_i\beta)} \tag{4}$$

where $\beta = (\beta_0, \beta_1, ..., \beta_d)$ are the model parameters to be estimated (with intercept β_0) (Liang et al., 2013).
The log-likelihood is shown in (5).

$$L(\beta|D) = -\sum_{i=1}^{n}\{y_i log[f(X_i\beta)] + (1 - Y_i)log[1 - f(X_i\beta)]\} \tag{5}$$

LR learns β that minimises this.

The next candidate is the random forest (RF) algorithm, an ensemble learning method which makes use of bagging to combine multiple decision trees (DT) (Injadat et al., 2018). RF involves the use of bootstrapping to randomly generate various sub-samples of the dataset, before fitting a decision tree to each (scikit learn, 2022). To classify a sample, each tree evaluates it individually, and the class which receives the most votes is selected as the final classification result (Salo et al., 2019).

This is followed by the SVM algorithm. SVMs are supervised learning models which can be used for classification and regression (Smola and Schölkopf, 2004). They work by partitioning data points using hyperplanes as decision boundaries, and often map data into higher-dimensional space to make the points linearly separable (Yang et al., 2018).
For a dataset of size n, the objective function is given by (6) (Zhang et al., 2003).

$$L = argmin_w\left\{\frac{1}{n}\sum_{i=1}^{n} max\{0, 1 - y_i f(x_i)\} + Cw^T w\right\} \tag{6}$$

where w is a normalisation vector and C is a regularisation hyperparameter. $f(x)$ is a function which measures the similarity between two points, known as the kernel. This function can come in the form of a radial basis function (RBF), linear kernel, polynomial kernel, or sigmoid kernel.

The final candidate is the MLP. This is a layered, feed-forward neural network where each layer is composed of nodes with each node connected to every other node in the subsequent layer (Delashmit et al., 2005).

Each of these models takes NBA data as input, and returns the predicted probability of each team winning, for a given game. To ensure reliable predicted probabilities, our features must be well-selected. We discuss the feature selection process below.

6.2. Feature Selection

Feature selection (FS) refers to the detection of relevant features and removal of irrelevant, redundant, or noisy data (Kumar and Minz, 2014). Various studies have shown the ability of FS to minimise the dimensionality of the problem, maximise the accuracy of classification and prevent overfitting (Wah et al., 2018; Li et al., 2017; Kira and Rendell, 1992). Three major families of FS methods exist. (*i*) Filter methods are those used to evaluate the relevance of features that are independent of the learning algorithm, e.g. ranking of features based on correlation with the response variable (Kumar and Minz, 2014). (*ii*) Wrapper methods are those which evaluate candidate feature subsets (under the given evaluation criterion) using a learning algorithm, and select the best-performing subset (Kumar and Minz, 2014). (*iii*) Embedded methods refer to learning algorithms with innate FS mechanisms, e.g. Lasso (L1 penalty), Ridge (L2 penalty) and random forest (Kumar and Minz, 2014).

(*i*) Two features are considered highly correlated if their spearman correlation coefficient is greater than 0.7 (Xiao et al., 2016). For a given group of highly correlated features, we consider all except the feature which is most correlated with the target to be redundant, and so remove them. This initial step is common to both branches of the predictive modelling pipeline (see figure 1), as filter methods are generally considered a pre-processing step (Kotsiantis et al., 2006). The features remaining after application of the filter method are denoted as subset A.

(*ii*) Next, a wrapper method is applied. Sequential forward selection (SFS) is used with LR as the learning algorithm and completion of search as the stopping criterion. This is implemented as a separate process along each branch, and for each, the input to the process is feature subset A. In the calibration branch, the feature subset under which the LR model achieves the lowest classwise-ECE is considered the optimal feature subset according to the SFS algorithm and is denoted as subset B. For the accuracy branch, the feature subset under which the highest accuracy is achieved is considered optimal and is denoted as subset C.

(*iii*) Finally, an embedded method is applied. We capitalise on the innate FS capability of lasso regression, by employing an LR model with the L1-penalty. We consider this an embedded method, as features not associated with the target have their coefficient shrunk to zero (effectively removing them from the model) (Fonti and Belitser, 2017). This step is also common to both branches. Feature subset A is the input to the process, and the output consists of those features which maintain non-zero coefficients. These features comprise subset D.

The FS process culminates in the identification of the optimal feature set for each branch of the pipeline. The candidate feature sets are the full feature set, as well as subsets A, B, C and D. For each of these feature sets, each of the candidate forecasting models is fit to an initial training dataset and evaluated on a validation set, and their scores are recorded under the given metric. For the calibration branch, the feature subset under which the lowest classwise-ECE is achieved on average across the candidate forecasting models is deemed to be the optimal feature set, and is used for all further predictive modelling along this branch of the pipeline. In the case of the accuracy branch, the feature subset under which the highest average accuracy is achieved is deemed optimal, and is used for all further modelling along this branch.

Due to the limitation of the classwise-ECE discussed in section 3, we place a constraint on model predictions. This is to ensure a low classwise-ECE is not achieved by a model generating predictions that are approximately equal to the overall class distribution for all instances. After grouping predictions into the corresponding bins, we look at the distribution of the weights of each bin, where the weight of a bin refers to the proportion of predictions associated with it. We want to ensure the predictions (for a given class) are not concentrated in a small number of bins. We therefore consider the kurtosis of the distribution of bin weights. Although it is misleading to say kurtosis measures the peakedness of a distribution, it is true that as a unimodal distribution approaches being concentrated entirely at its mean, its kurtosis approaches infinity (Darlington, 1970). To prevent such scenarios, we impose the following constraint on model predictions: if the excess kurtosis of the distribution of bin weights (for a given class) is greater than 0, the classwise-ECE achieved is excluded from the calculation of the average classwise-ECE achieved under the feature subset. Further, if the excess kurtosis is positive for the majority of the candidate forecasting models, the feature subset in question is not considered a candidate for the optimal feature subset under calibration. Naturally, these constraints do not apply to the accuracy branch.

6.3. Hyperparameter Optimisation

Model performance is significantly influenced by the choice of hyperparameters, and automating the process of hyperparameter tuning has become the focus of much research in recent years. Automated hyperparameter optimisation (HPO) has several important benefits, including reduction of human effort required for applying ML, improvement in performance of ML models, and improvement in reproducibility and fairness of research (Hutter et al., 2019). Many optimisation problems are non-convex or non-differentiable, in which case traditional optimisation techniques may result in a local rather than global optimum (Luo, 2016). Popular among the non-traditional optimisation techniques that have been used for HPO problems is an iterative algorithm known as bayesian optimisation (BO) (Snoek et al., 2012). BO is considered more efficient than traditional HPO techniques like random search or grid search, because the algorithm decides which points in the hyperparameter search space to evaluate based on previously-obtained results, rather than letting the user specify the points. This involves a careful balance between exploration and exploitation, so that the search primarily examines promising regions of the hyperparameter search space (where the global optimum is likely to be found), but also explores less visited regions of the space, to avoid missing better configurations (Hazan et al., 2017). We implement a form of BO known as BO-TPE, regarded as one of the most suitable HPO techniques for LR, RF, SVM, and MLP classifiers (Yang and Shami, 2020).

To run this algorithm, the user specifies the forecasting model along with the hyperparameters to be optimised, the search space for each hyperparameter, training and validation data to train and score the model, and the objective function to be minimised. The algorithm returns the optimal set of hyperparameter values along with its score on the validation data. Due to the element of inherent randomness in this process, we run the algorithm several times over different random seeds before selecting the optimal set of hyperparameters. The full process is described in appendix C (section C.3.1). HPO is implemented separately along each branch of the predictive modelling pipeline (see figure 1). For the calibration branch, the objective function is the classwise-ECE, while the negative of accuracy is used for the accuracy branch. The hyperparameter search space for each model is given in appendix C (see table C.23).

6.4. Model Selection

The final stage of the predictive modelling pipeline is model selection. We fit each model to an extended training set (consisting of the initial training data combined with the validation data) under the optimal feature set and hyperparameter values for the given branch, and generate predictions for the test set. We then evaluate these predictions under the given metric. Along the calibration branch, the candidate forecasting model which achieves the lowest classwise-ECE on the test set is deemed to be the best calibration-optimised model. Along the accuracy branch, the model which achieves the highest accuracy on the test set is selected as the best accuracy-optimised model. These two models are the final output of the predictive modelling pipeline, and are used to generate predictions for each game in an NBA season. These predictions (combined with the bookmaker's odds) are the input of our betting experiments, as detailed in the next section.

7. Betting Experiments

We fit each model selected by the pipeline to a final training set, and generate predictions for a betting simulation set. The final training set comprises the extended training and test sets, and the betting simulation set consists of a single NBA season (details of each data set are discussed in section 8). We use these predictions to evaluate several betting systems over the given season.

Beginning with an initial bankroll of \$10,000, we iterate through each game in the betting simulation set in chronological order, and for each:

Step 1. For each team, we compare the forecasting model's predicted probability of victory to the probability implied by the bookmaker's odds, to decide whether or not to place a bet. This decision is determined by the **betting strategy**

Step 2. If the decision is to bet, we determine the stake by the **betting rule**

Step 3. We subtract the stake from the bankroll

Step 4. If the bet is successful, we add the stake and the winnings to the bankroll

As the betting strategy dictates which games to bet on, it is a crucial element of any betting system.

7.1. Betting Strategies

We implement three distinct strategies. In all cases, we wager only on value bets, as they are profitable in expectation. Our strategies are as follows: (*i*) Wager on all value bets. (*ii*) Avoid value bets with a predicted probability of victory that exceeds the implied probability by more than 10%. If the model strays too far from the bookmaker's forecast, it may be a sign that the model is missing important information about the fixture (e.g. the bookmakers may know that a particular player is having difficulty in their personal life which could affect their performance). (*iii*) Wager only on strong favourites. This strategy comes recommended by Hubáček and colleagues, who suggested that probabilistic predictions of around 50% are typically less precise than forecasts of higher confidence (Hubáček et al., 2019). Thus, if the model is indifferent about the favourite, it may be safer not to bet.

7.2. Betting Rules

We test four different rules for each strategy. These are the full-kelly, half-kelly, quarter-kelly and eighth-kelly (Hsieh and Barmish, 2015). These rules represent different levels of risk tolerance (Dotan, 2020).

The algorithm used to conduct each betting simulation is shown in table 1. The algorithm has several parameters to incorporate different strategies and rules. Betting strategy and rule parameters are described in tables 2 and 3, respectively.

Table 1: Betting simulation algorithm

Bankroll = B_0

For each game in the dataset:
- P_h = Predicted probability of victory for home team
- O_h = Bookmaker's odds for home team victory
- P_a = Predicted probability of victory for away team
- O_a = Bookmaker's odds for away team victory

If $L_t < (P_h - 1/O_h) < U_t$ and $P_h > L$:
- K = ($P_h \times$ Bankroll $-(1 - P_h)$)/Bankroll
- Stake = $F \times K \times$ Bankroll
- Bankroll = Bankroll - Stake
- If home team won:
  - Winnings = (Stake $\times O_h$) - Stake
  - Bankroll = Bankroll + Stake + Winnings

Else If $L_t < (P_a - 1/O_a) < U_t$ and $P_a > L$:
- K = ($P_a \times$Bankroll $-(1 - P_a)$)/Bankroll
- Stake = $F \times K \times$Bankroll
- Bankroll = Bankroll - Stake
- If away team won:
  - Winnings = (Stake $\times O_a$) - Stake
  - Bankroll = Bankroll + Stake + Winnings

Until end of dataset is reached

Table 2: Betting strategy parameters and their meanings

Parameter	Description
$L \in [0, 1)$	The value that the predicted probability of victory must exceed for a bet to be placed
$L_t \in [0, 1)$	The value that the difference between predicted probability and implied probability of victory must exceed for a bet to be placed
$U_t \in (0, 1]$	The value that the difference between predicted probability and implied probability of victory must not exceed for a bet to be placed

The parameters for strategies 1, 2 and 3 are as follows.

1. $L = 0$, $L_t = 0$, $U_t = 1$
2. $L = 0$, $L_t = 0$, $U_t = 0.1$
3. $L = 0.8$, $L_t = 0$, $U_t = 1$

Table 3: Betting rule parameters and their meanings

Parameter	Description
$F \in [0, 1]$	Fractional Kelly coefficient: If $F = 1$ the stake is determined by the Kelly criterion, if $F = 0.5$ the half-Kelly criterion is used etc.
$B_0 \in [0, \infty)$	Initial Bankroll

As discussed, each betting system consists of a strategy and a rule, in addition to a forecasting model optimised for either calibration or accuracy. Following the algorithm described in table 1, we simulate each betting system over a single NBA season. Measuring the ROI achieved by each system, we compare the profitability of calibration-optimised systems to their accuracy-optimised counterparts, to test our hypothesis. The results of this experiment are discussed in the next section.

8. Results

To undertake this research, we obtained NBA data from the 2014/2015-2018/2019 seasons from basketball-reference.com (Sports-Reference-LLC, 2022) We used the 2014/2015-2015/2016 seasons as an initial training set. The models were fitted to this data during the FS and HPO processes. The 2016/2017 season was used as a validation set to evaluate model performance during these processes. After the FS and HPO processes were completed, the 2014/2015-2016/2017 seasons were used as an extended training set. The forecasting models were fitted to this data ahead of model selection, during which they were evaluated on a test set consisting of the 2017/2018 season. The best-performing models were then fitted to a final training set spanning the 2014/2015-2017/2018 seasons and used to generate predictions for the 2018/2019 season - which comprised our betting simulation set. A key requirement for the betting simulation was obtaining authentic odds published by a bookmaker. To fulfill this requirement, we obtained the publicly available closing moneyline odds for the 2018/2019 NBA season, published by Las Vegas sportsbook Westgate (sportsbookreviewsonline, 2022).

Certain features were dropped from the dataset prior to FS, as a result of showing signs of covariate shift, being a linear combination of other features, or presenting only null values. A list of these features is provided in appendix B (see table B.18). Many more features were considered redundant by each of the FS methods employed. Table 4 shows the subsets generated by each of the FS methods, along with the features dropped and features selected by each method.

Table 4: Feature subsets resulting from each feature selection method. Full list of basic and advanced box score statistics, and their meaning, is provided in appendix B (see tables B.16 and B.17). FG: Field Goals, FG%: Field Goal Percentage, 3P: 3-Point Field Goals, FT: Free Throws, FT%: Free Throw Percentage, ORB: Offensive Rebounds, DRB: Defensive Rebounds, AST: Assists, STL: Steals, BLK: Blocks, TOV: Turnovers, PF: Personal Fouls, TS%: True Shooting Percentage, eFG%: Effective Field Goal Percentage, 3PAr: 3-Point Attempt Rate, FTr: Free Throw Attempt Rate, ORB%: Offensive Rebound Percentage, DRB%: Defensive Rebound Percentage, TRB%: Total Rebound Percentage, AST%: Assist Percentage, STL%: Steal Percentage, BLK%: Block Percentage, TOV%: Turnover Percentage, ORtg: Offensive Rating, DRtg: Defensive Rating, drbp: as described in section 5, bpm: as described in section 5, Previous Season Record: as described in section 5.

Feature Subset	**Methods Employed**	**Features Dropped**	**Features Selected**
Full	None		FG, FG%, 3P, FT, FT%, ORB, DRB, AST, STL, BLK, TOV, PF, TS%, eFG%, 3PAr, FTr, ORB%, DRB%, TRB%, AST%, STL%, BLK%, TOV%, ORtg, DRtg, drbp, bpm, Previous Season Record
A	Correlated Feature Removal	FG%, FT, AST, STL, TOV, TS%, eFG%, 3PAr, FTr, ORB%, DRB%, TRB%, BLK%, TOV%, ORtg, DRtg	FG, 3P, FT%, ORB, DRB, BLK, PF, AST%, STL%, drbp, bpm, Previous Season Record
B	Correlated Feature Removal and Forward Selection with classwise-ECE as Evaluation Metric	FG, FG%, 3P, FT, DRB, AST, STL, BLK, TOV, PF, TS%, eFG%, 3PAr, FTr, ORB%, DRB%, TRB%, STL%, BLK%, TOV%, ORtg, DRtg, drbp, bpm, Previous Season Record	FT%, ORB, AST%
C	Correlated Feature Removal and Forward Selection with accuracy as Evaluation Metric	FG, FG%, FT, FT%, DRB, AST, STL, BLK, TOV, PF, TS%, eFG%, 3PAr, FTr, ORB%, DRB%, TRB%, STL%, BLK%, TOV%, ORtg, DRtg, Previous Season Record	bpm, 3P, AST%, ORB, drbp
D	Correlated Feature Removal and L1 Penalty	FG%, FT, AST, STL, TOV, TS%, eFG%, 3PAr, FTr, ORB%, DRB%, TRB%, BLK%, TOV%, ORtg, DRtg	FG, 3P, FT%, ORB, DRB, BLK, PF, AST%, STL%, drbp, bpm, Previous Season Record

As a result of applying the filter method, 16 features were removed due to high correlation (e.g. FG%, TS%). Interestingly, no remaining feature was considered redundant by the embedded method, meaning subset A and D were equivalent. Significantly more features were removed by the wrapper method in each branch. SFS under classwise-ECE resulted in an optimal feature set with just three features, while under accuracy it contained only five (subsets B and C respectively, as shown in table 4).

Along each branch, the average score achieved by the models under each feature subset was recorded. These figures are given in appendix C (see tables C.21 and C.22). Along the calibration branch, the optimal feature subset was subset A, with an average classwise-ECE of 5.06% (see table C.21). Along the accuracy branch, subset A was once again the optimal feature subset, with an average accuracy of 66% (see table C.22). Therefore, subset A was used for all further modelling steps.

Next, HPO was implemented along each branch using the BO-TPE algorithm. The optimal hyperparameter values

identified for each model are provided in appendix C (see tables C.24 and C.25). These values were used for all further modelling steps.

Using the optimal feature set and hyperparameter values identified for each branch, the models were evaluated on a test set. Their scores are provided in tables 5 and 6 below.

Table 5: Classwise-ECE achieved on the test set by each model along the calibration branch

Model	**classwise-ECE**
Logistic Regression	5.64%
Random Forest	4.06%
Support Vector Machine	3.49%
Multi-Layer Perceptron	5.71%

The SVM model was identified as the best calibration-optimised model, achieving a classwise-ECE of 3.49% on the test set. This was followed by RF, with a classwise-ECE of 4.06%, and LR and MLP models, with respective classwise-ECEs of 5.64% and 5.71% (see table 5).

Table 6: Accuracy achieved on the test set by each model along the accuracy branch

Model	**Accuracy**
Logistic Regression	69.23%
Random Forest	68.28%
Support Vector Machine	68.28%
Multi-Layer Perceptron	68.02%

The most accurate model was LR with a test set accuracy of 69.23%. The other candidate forecasting models achieved accuracies ranging from 68.02% to 68.28% (see table 6). As a result, LR was identified as the best accuracy-optimised model.

These two final models (calibration-optimised SVM and accuracy-optimised LR) were used to generate predictions for the 2018/2019 season. Taking as input these predictions along with the bookmaker's odds, we implemented and evaluated several betting systems, each consisting of a strategy and a rule. We tested three distinct strategies, in combination with four different rules, as described in sections 7.1 and 7.2. Each simulation was carried out according to the algorithm described in table 1.

8.1. Strategy 1: Wager on all value bets

The first strategy we tested was to wager on all value bets. This strategy was considered high risk due to the possibility of identifying 'value bets' when the model may be missing information, as well as betting on games for which model confidence is low. Our first simulation was that of a bettor following strategy 1 using the calibration-optimised SVM. Figure 3 depicts the evolution of the bettor's bankroll over the course of the 2018/2019 season. This follows a different trajectory for each of the four rules, as shown below.

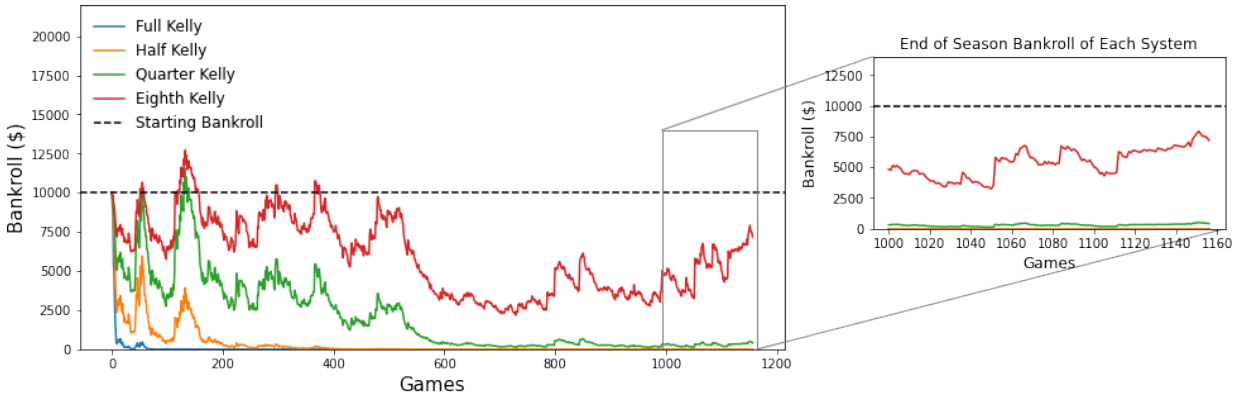

Figure 3: Bettor's bankroll over the course of the 2018/2019 season under strategy 1 in combination with each of the different betting rules, using the calibration-optimised SVM predictions. The blue curve corresponds to the full-Kelly rule, the orange curve corresponds to the half-Kelly, the green curve corresponds to the quarter-Kelly and the red curve corresponds to the eighth-Kelly. The black broken line represents the bettor's initial bankroll ($10,000). Curves lying above this line at the season's end generate a profit, and are considered successful betting systems.

When we simulated a bettor implementing a calibration-optimised SVM system under strategy 1, we saw the bankroll instantly crash for each of the four rules (see figure 3 games 0-10). The full-Kelly system wagered nearly the entire budget on an initial bet, crashing almost to zero and never recovering. The half-kelly system likewise never recovered from the initial dip, soon after depleting to zero. For the quarter-Kelly and eighth-Kelly, several spikes and dips followed the initial crash. This tells us that plenty of value bets were identified, but many of them were unsuccessful. Approximately halfway through the season, the quarter-Kelly tapered off towards zero. The eighth-Kelly (the most conservative system) recovered to some extent, ultimately achieving an ROI of -28.18%. While myriad value bets were identified, many were unsuccessful. Owing to its conservative nature, the eighth-Kelly was the best-performing of these systems. Where others bet and lost large sums, the eighth-Kelly bet and lost in smaller amounts. Nonetheless, each of these systems ultimately achieved negative returns, as shown in the magnified portion of figure 3. The large sums lost on individual bets are the result of large differences between the predicted and implied probabilities of victory. As these were losing bets, it is possible the predicted probability of victory was much higher than it should have been in these cases. This kind of bet, in which the predicted probability of victory is considerably greater than the implied probability (and perhaps also considerably greater than the true probability!), is exactly the kind of bet strategy 2 is designed to avoid by not betting when the predicted probability of victory exceeds the implied probability by more than 10%.

When simulating a bettor applying an accuracy-optimised LR system under strategy 1, we saw a similar trend, as shown in figure 4.

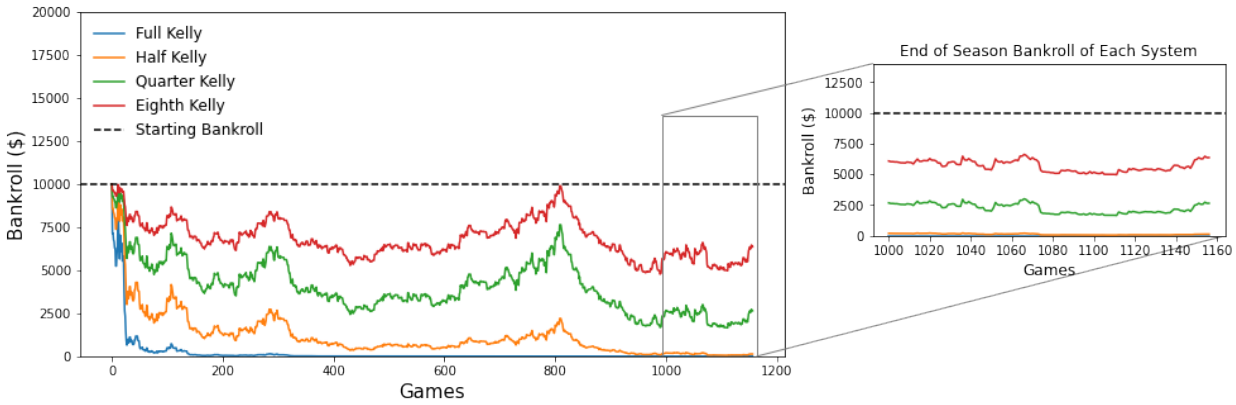

Figure 4: Bettor's bankroll over the course of the 2018/2019 season under strategy 1 in combination with each of the different betting rules, using the accuracy-optimised LR predictions. Meaning of each curve described in the caption of figure 3.

The full-Kelly plummeted to zero, and while the half-Kelly experienced a few spikes, it ultimately also went to zero. The quarter and eighth-Kelly systems experienced mostly steady dips but also several sharp spikes, however, not enough to reach the original level of the bankroll (as shown in magnified portion of figure 4). These systems achieved respective ROIs of -73.59% and -36.42%. One might speculate that the failure of these systems owes to the same reasons as that of their calibration-optimised counterparts. Table 7 provides the results for each strategy 1 betting simulation.

Table 7: Results of betting simulations under strategy 1.

Model	**% Games Bet On**	**% Bets Won**	**Betting Rule**	**Final Bankroll**	**ROI**
Calibration-optimised SVM	90.58%	34.54%	Full Kelly	$0	-100%
			Half Kelly	$0	-100%
			Quarter Kelly	$418.20	-95.82%
			Eighth Kelly	$7,182.26	-28.18%
Accuracy-optimised LR	77.44%	44.64%	Full Kelly	$0	-100%
			Half Kelly	$128.71	-98.71%
			Quarter Kelly	$2,641.44	-73.59%
			Eighth Kelly	$6,358.27	-36.42%

As is evident from table 7, no betting system generated a profit under strategy 1. Interestingly, calibration-optimised systems opted to bet on 90.58% of games, winning 34.54% of the time, while accuracy-optimised systems bet on 77.44% of games and won 44.64% of the time.

8.2. Strategy 2: Avoid value bets with a predicted probability of victory which exceeds the implied probability by more than 10%

The second strategy we tested involved betting on all value bets except for those in which the predicted probability of victory exceeded the implied probability by more than 10%. Our expectation was that this strategy would avoid some of the big losses that might result from betting on games for which the model is missing key information. Figure 5 shows the results of simulating a bettor applying a calibration-optimised SVM system under strategy 2, for each betting rule.

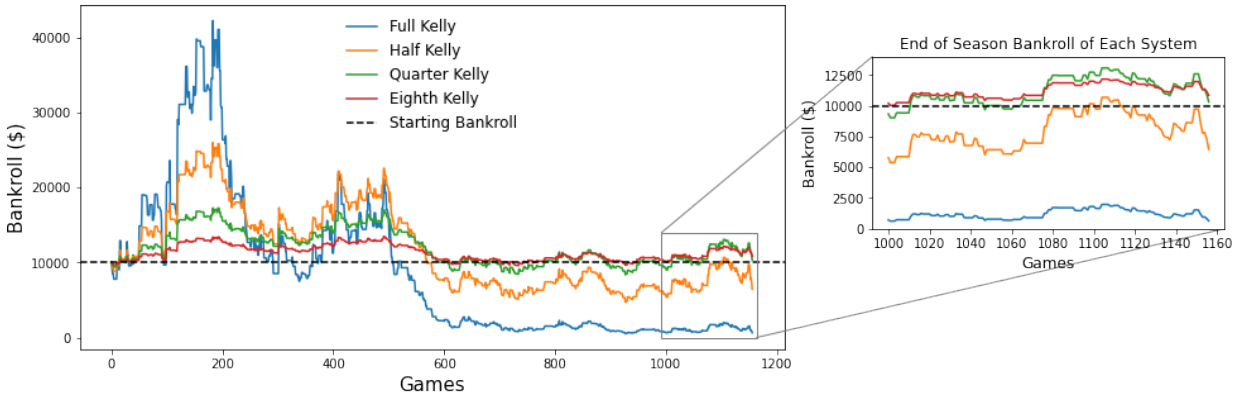

Figure 5: Bettor's bankroll over the course of the 2018/2019 season under strategy 2 in combination with each of the different betting rules, using the calibration-optimised SVM predictions. Meaning of each curve described in the caption of figure 3.

Under strategy 2, the calibration-optimised systems identified several successful value bets early in the season, as shown in figure 5. The full-Kelly system capitalised on these, reaching a peak of over $40,000. It lost big soon thereafter and found itself below the starting bankroll, ultimately tapering off to almost zero. The half-Kelly followed a less exaggerated but similar trend, also ultimately unsuccessful, with an ROI of -35.37%. The quarter and eighth-Kelly systems remained steadier throughout, both benefiting less from early wins and suffering less from later losses. Ultimately, these two systems were profitable, achieving respective ROIs of 3.2% and 8.46%.

The early success (in stark contrast to strategy 1) was likely a result of the constraint on the difference between the predicted and implied probabilities of victory. Avoiding wagers when the predictive model strays too far from the bookmaker's forecast (possibly because the model is missing important information which the bookmaker possesses) appears a smart move, as avoiding these early losses allowed the systems to win big early. However, unsuccessful value bets later in the season proved costly, and the systems generally achieved negative returns.

Simulating a bettor employing an accuracy-optimised LR system under strategy 2 saw positive returns achieved for each betting rule, as shown in figure 6 below.

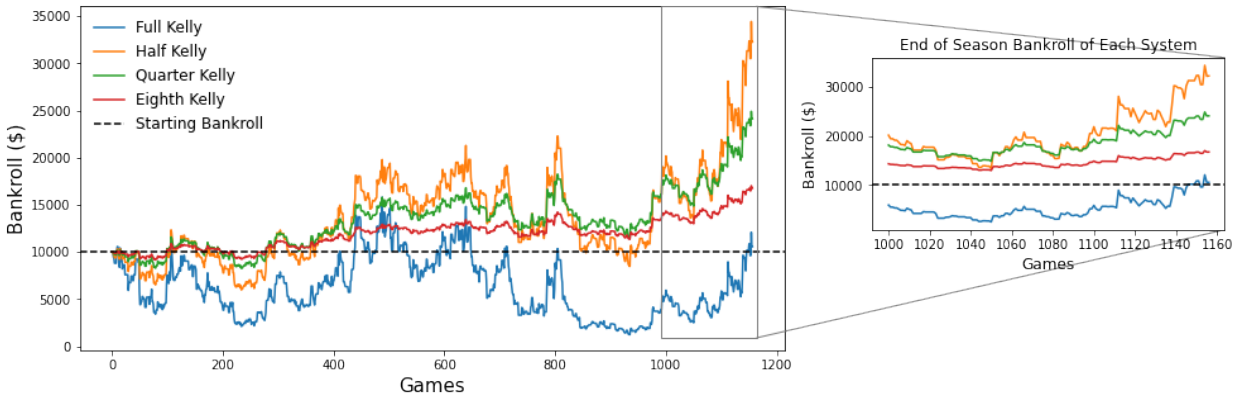

Figure 6: Bettor's bankroll over the course of the 2018/2019 season under strategy 2 in combination with each of the different betting rules, using the accuracy-optimised LR predictions. Meaning of each curve described in the caption of figure 3.

Several unsuccessful value bets were identified by the accuracy-optimised systems early in the season. The full-Kelly system had lost approximately half its initial budget early in the season, but was profitable by the halfway point (see figure 6 circa game 600). Experiencing further sharp spikes and dips, by season's end it was in the money, achieving an ROI of 5.71%. The half-Kelly was subjected to less severe early dips. It gained momentum by winning

some big bets over the season. By a large margin the most profitable of these systems, it skyrocketed towards the tail end of the season and achieved an ROI of 222.84%. The quarter and eighth-Kelly systems followed a less exaggerated but similar trend and generated respective ROIs of 141.29% and 67.54%.

The percentage of games bet on dropped from 77.44% under strategy 1 to 62.58% here for these accuracy-optimised systems, while the percentage of winning bets increased from 44.64% to 47.1%. Strategy 2's constraint appears to have prevented these systems from losing large sums on supposed 'value' bets in cases where the model may be missing information. This resulted in positive returns on average and an ROI of 222.84% in the best case. Table 8 provides the results for each strategy 2 betting simulation.

Table 8: Results of betting simulations under strategy 2.

Model	**% Games Bet On**	**% Bets Won**	**Betting Rule**	**Final Bankroll**	**ROI**
Calibration-optimised SVM	39.5%	40.26%	Full Kelly	\$638.68	-93.61%
			Half Kelly	\$6,462.96	-35.37%
			Quarter Kelly	\$10,319.70	3.2%
			Eighth Kelly	\$10,846.39	8.46%
Accuracy-optimised LR	62.58%	47.1%	Full Kelly	\$10,571.25	5.71%
			Half Kelly	\$32,283.78	222.84%
			Quarter Kelly	\$24,129.38	141.29%
			Eighth Kelly	\$16,754.49	67.54%

In comparison to strategy 1, strategy 2 systems were much more conservative in their betting. Calibration-optimised systems wagered on 39.5% of games, winning 40.26% of the time, with two systems generating a profit. Accuracy-optimised systems wagered on 62.58% of games, winning 47.1% of the time, with all rules generating a profit. The half-Kelly system proved the most profitable of these, with an ROI of 222.84%. Across the board, strategy 2 systems were much more profitable than strategy 1 systems.

8.3. Strategy 3: Wager only on strong favourites

The final strategy we tested focused on value bets where the favourite's predicted probability of victory was greater than 80%. Under this approach, we expected the percentage of value bets identified to be low, but counterbalanced by a high percentage of successful bets. Simulating this approach for a bettor using the calibration-optimsed SVM led to extremely lucrative returns under each betting rule, as shown in figure 7.

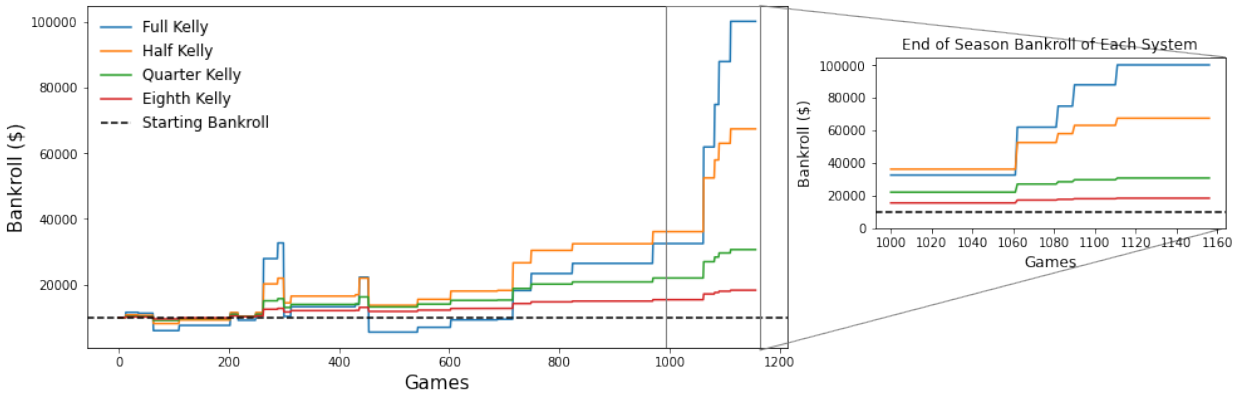

Figure 7: Bettor's bankroll over the course of the 2018/2019 season under strategy 3 in combination with each of the different betting rules, using the calibration-optimised SVM predictions. Meaning of each curve described in the caption of figure 3.

These calibration-optimised systems also started the season with several losing bets. The full-Kelly system found itself below its initial budget at the halfway point of the season. By identifying several successful value bets, it

rebounded late in the season, before skyrocketing to achieve an unprecedented ROI of 902.01%. The half-Kelly system experienced a short depression early on and recovered soon after. It also skyrocketed towards the end of the season and achieved an incredible ROI of 574.41%. The quarter and eighth-Kelly systems fluctuated much less and increased consistently for the majority of the season. These systems achieved respective ROIs of 206.82% and 83.12%. The curves in figure 7 almost appear to exhibit a different resolution when compared to earlier figures. This may be because the number of games the systems bet on was severely reduced under strategy 3. With calibration-optimised systems wagering on just 2.16% of games under strategy 3, a value bet on a strong favourite appears to be a rare find. This was expected, as the bookmakers ought not to offer value on highly probable outcomes if they wish to remain in business! Boasting an 80% success rate, this wagering strategy appears sound. Notably, every calibration-optimised system was profitable under strategy 3, with some achieving enormous returns.

In comparison to these calibration-optimised systems, simulating a bettor using the accuracy-optimised LR under strategy 3 was not as fruitful. Figure 8 shows the returns achieved in this simulation under each betting rule.

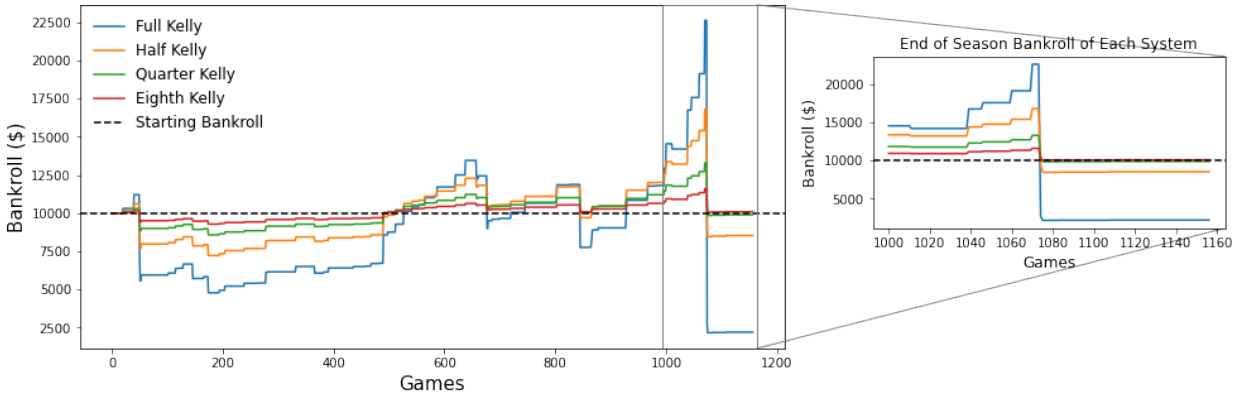

Figure 8: Bettor's bankroll over the course of the 2018/2019 season under strategy 3 in combination with each of the different betting rules, using the accuracy-optimised LR predictions. Meaning of each curve described in the caption of figure 3.

The accuracy-optimised systems also identified several losing bets early in the season under strategy 3. The full-Kelly system lost half the initial budget before ascending to a peak of approximately $22,500 as the end of the season approached. Staking almost the entire bankroll on the final bet of the season and losing, this system ultimately achieved an ROI of -77.97%. The half, quarter, and eighth-Kelly systems all repeated this trend, with each deviating from the starting bankroll less than their predecessor. They achieved respective ROIs of -14.67%, -1.13% and 0.84%. With only 6.4% of games seeing bets placed, finding value under this strategy was once again rare. An 85.14% success rate was still not enough for these systems to be profitable across the board. While appearing extremely lucrative approaching the season's end, one final, unsuccessful bet saw these systems lose their profitability. Two of the systems lost considerable sums of money, while the others approximately broke even. These results illustrate the danger of strategy 3 - placing so few bets allows one big loss to have an outsized impact on overall returns. Table 9 provides the results for each strategy 3 betting simulation.

Table 9: Results of betting simulations under strategy 3.

Model	% Games Bet On	% Bets Won	Betting Rule	Final Bankroll	ROI
Calibration-optimised SVM	2.16%	80%	Full Kelly	$100,200.96	902.01%
			Half Kelly	$67,440.79	574.41%
			Quarter Kelly	$30,682.16	206.82%
			Eighth Kelly	$18,311.76	83.12%
Accuracy-optimised LR	6.4%	85.14%	Full Kelly	$2,203.20	-77.97%
			Half Kelly	$8,532.73	-14.67%
			Quarter Kelly	$9,886.71	-1.13%
			Eighth Kelly	$10,084.05	0.84%

As expected, calibration-optimised systems bet on just a fraction of games, at 2.16%. These were wisely chosen, as 80% of bets were successful, and all systems generated positive returns. The full-Kelly system under strategy 3 was the single best-performing system tested in our experiments, achieving an ROI of 902.01%. Accuracy-optimised systems similarly bet on few games, just 6.4%, and won 85.14% of bets. Two of these systems approximately broke even, while the others lost considerable sums of money. Strategy 3 systems outperformed strategy 1 systems across the board, and were more successful than strategy 2 systems on average.

8.4. Evaluation of Strategies and Rules

To formally compare the performance of each of our strategies (wager on all value bets, avoid bets where the predicted probability of victory exceeds the implied probability by more than 10%, and wager only on strong favourites), we examine the returns achieved under each strategy, on average, as well as in the best case. Table 10 displays these results.

Table 10: Head-to-head comparison of the betting strategies. Best model and best rule refer to the forecasting model and betting rule under which the maximum ROI of a given strategy was achieved.

Strategy	Average ROI	Maximum ROI	Best Model	Best Rule
Strategy 1	-79.09%	-28.18%	Calibration-optimised SVM	Eighth-Kelly
Strategy 2	40.01%	222.84%	Accuracy-optimised LR	Half-Kelly
Strategy 3	209.18%	902.01%	Calibration-optimised SVM	Full-Kelly

Restricting wagers to strong favourites proved the most profitable strategy, achieving an average ROI of 209.18%, as shown in table 10. This was also the strategy used in the single most profitable betting system, which combined strategy 3 with the full-Kelly betting rule and used predictions generated by the calibration-optimised SVM to achieve an ROI of 902.01%.

To compare the performance of each rule, we examine the same metrics in table 11.

Table 11: Head-to-head comparison of the betting rules. Best model and best strategy refer to the forecasting model and betting strategy under which the maximum ROI of a given rule was achieved.

Rule	Average ROI	Maximum ROI	Best Model	Best Strategy
Full-Kelly	89.36%	902.01%	Calibration-optimised SVM	Strategy 3
Half-Kelly	91.41%	574.41%	Calibration-optimised SVM	Strategy 3
Quarter-Kelly	30.13%	206.82%	Calibration-optimised SVM	Strategy 3
Eighth-Kelly	15.9%	83.12%	Calibration-optimised SVM	Strategy 3

The half-Kelly was the best-performing betting rule on average, achieving an average ROI of 91.41%, and a best-case ROI 574.41%, as shown in table 11. The full-Kelly was a close second, with an average ROI of 89.36%, and an unprecedented best-case ROI of 902.01%. For each rule, the combination of forecasting model and strategy which generated the greatest returns was the calibration-optimised SVM under strategy 3.

8.5. Evaluation of Central Hypothesis

As discussed in section 4, the experiment was designed to answer two key questions. The first focused on the theoretical aspect of our hypothesis, and asked whether optimising a sports betting model for calibration or accuracy leads to greater profit generation. The second question related to the practical implication for bettors, and asked which system a bettor should employ to generate the greatest returns, and whether the forecasting model used in this system is optimised for calibration or accuracy.

8.5.1. Does optimising a sports betting forecasting model for calibration, rather than accuracy, allow for greater profit generation?

To test the hypothesis that optimising a sports betting forecasting model for calibration, rather than accuracy, leads to greater profit generation, we conducted an upper-tailed, paired t-test at the 10% level of significance (Snedecor and Cochran, 1989). Let C_i^j equal the ROI of the betting system using the calibration-optimised forecasting model under strategy i and rule j, and let A_i^j equal the ROI of the betting system using the accuracy-optimised forecasting model under strategy i and rule j. We tested the hypothesis that the mean ROI of calibration-optimised betting systems is greater than the mean ROI of accuracy-optimised betting systems. That is, for $D_i^j = C_i^j - A_i^j$, our null hypothesis was $H_0 = \mu_D = \mu_C - \mu_A = 0$, where μ_D represents the population mean of the differences. Our alternative hypothesis was $H_1 = \mu_D > 0$. The results are displayed in table 12.

Table 12: Testing the hypothesis that the mean ROI of calibration-optimised betting systems is greater than that of accuracy-optimised betting systems at the 10% level of significance, using a paired t-test.

Null Hypothesis	**Alternative Hypothesis**	**Test statistic**	**P-Value**
$H_0 : \mu_C = \mu_A$	$H_1 : \mu_C > \mu_A$	1.075	0.153

With a test statistic of 1.075, although the mean ROI of calibration-optimised betting systems was greater than that of accuracy-optimised betting systems, the difference was not significant. As the p-value of 0.153 is greater than 0.1, we fail to reject the null hypothesis that the mean ROI of each group is equal, at the 10% level of significance.

We also conducted upper-tailed, paired t-tests with this hypothesis restricted to each strategy. Here the null hypotheses were $H_0 = \mu_{D_i} = 0$ with corresponding alternative hypotheses $H_1 = \mu_{D_i} > 0$ for $i = 1, 2, 3$. Table 13 displays the results.

Table 13: Testing the hypotheses that, for a given strategy, the mean ROI of calibration-optimised betting systems is greater than that of accuracy-optimised betting systems at the 10% level of significance, using a paired t-test

Null Hypothesis	**Alternative Hypothesis**	**Test statistic**	**P-Value**
$H_0 : \mu_{C_1} = \mu_{A_1}$	$H_1 : \mu_{C_1} > \mu_{A_1}$	-0.589	0.701
$H_0 : \mu_{C_2} = \mu_{A_2}$	$H_1 : \mu_{C_2} > \mu_{A_2}$	-3.226	0.976
$H_0 : \mu_{C_3} = \mu_{A_3}$	$H_1 : \mu_{C_3} > \mu_{A_3}$	2.293	0.053 *

We also fail to reject the null hypothesis for strategy 1 and strategy 2. However, when restricting our test to betting systems under strategy 3, as the p-value is less than 0.1 (p=0.053), we reject the null hypothesis at the 10% level of significance. This suggests that under strategy 3, the mean ROI of calibration-optimised betting systems is greater than that of accuracy-optimised betting systems.

8.5.2. Should bettors optimise the forecasting model for accuracy or calibration?

The main concern for sports bettors is identifying the single most profitable betting system. We probe further, and ask whether the forecasting model in this system is optimised for calibration or accuracy. To find out, we review some summary statistics in table 14.

Table 14: Head-to-head comparison of the performances of each final forecasting model's betting systems. Best strategy and best rule refer to the strategy and rule under which the maximum ROI was achieved for a given forecasting model.

Model	**Average ROI**	**Maximum ROI**	**Best Strategy**	**Best Rule**
Calibration-optimised SVM	110.42%	902.01%	Strategy 3	Full-Kelly
Accuracy-optimised LR	2.98%	222.84%	Strategy 2	Half-Kelly

Betting systems that utilised the calibration-optimised SVM outperformed those which utilised the accuracy-optimised LR on average, as well as in the best case. The difference in average ROIs was greater than 107%. The difference between best-case ROIs was almost 680%. The system that generated the greatest returns utilised a calibration-optimised forecasting model (SVM). Under strategy 3, and using the full-Kelly betting rule, this system generated an incredible ROI of 902.01%. In the next section, we reflect on these results and discuss their implications.

9. Discussion

In this paper, we aimed to devise a data-driven approach to sports betting. Focusing on the NBA, we set out to show that it is possible to leverage data to make a profit over a single season. Identifying a gap in the literature, we hypothesised that accuracy is not the most appropriate metric to evaluate the performance of the forecasting model in a sports betting system, and that calibration is more useful in this setting.

To test this hypothesis, we optimised one group of forecasting models for calibration (through feature selection and hyperparameter tuning), and an identical group of models for accuracy. Selecting the best model from each group and generating predictions for an NBA season, we implemented several betting systems, testing out different strategies to determine which games to bet on, and rules to determine the size of each bet. Measuring the returns achieved by each betting system, we were able to compare the profitability of calibration-optimised systems and accuracy-optimised systems. To the authors' knowledge, this work represents the first attempt to study the effect on profit generation of optimising a sports betting system's forecasting model for calibration, as opposed to the traditional approach of maximising accuracy. Another novelty of this work comes in the form of the features used in the forecasting model. While a common approach for NBA game outcome prediction is to use box score statistics averaged over the season to date, we show that averaging differences in box score statistics versus opponents over the season to date can result in similar success.

Optimising the forecasting model for calibration led to an average ROI of 110.42%, and an ROI of 902.01% in the most profitable system, while optimising for accuracy led to an ROI of 2.98% on average and 222.84% in the best case. We found that the most profitable betting strategy, on average and in the best case, is to bet only on value bets when the model is highly confident. This echoes the results of Hubáček and colleagues, who showed that profits can be improved by avoiding betting on games where the model is indifferent about the favourite (Hubáček et al., 2019). We found that the half-Kelly was the most profitable betting rule on average. This is also in agreement with the literature, the common consensus being that the full-kelly is too aggressive (Dotan, 2020; Hsieh and Barmish, 2015). Conducting statistical tests to examine the ground truth of our hypothesis, we showed that optimising the forecasting model for calibration rather than accuracy does indeed allow for greater profit generation, under the strategy of betting only when the model is highly confident. This represents one of the first attempts to compare the performance of sports betting systems by means of a statistical test. In the case of our best-performing betting system, we showed that bettors can increase their wealth ten-fold over a single season, using a calibration-optimised forecasting model. These exciting findings support our conjecture that in a data-driven sports betting system, optimising the forecasting model for calibration, rather than accuracy, leads to greater profit generation. This reiterates the findings of Hubáček

and colleagues, who showed that optimising for decorrelation with the bookmaker's odds leads to greater returns than optimising for accuracy (Hubáček et al., 2019).

While the results are encouraging, a few critiques can be made. The most obvious is the use of a single season for the betting experiments, as no guarantee can be made that systems which were successful in this particular season would have been similarly profitable over other seasons. Further, the constraint imposed on predictions along the calibration branch of the predictive modelling pipeline (enforcing a platykurtosis on the distribution of bin weights) was somewhat arbitrary, and perhaps a better solution could be found to deal with the limitation of the classwise-ECE mentioned in section 3.

This research also leaves room for future work. One could experiment with completely different betting rules. For example, instead of basing rules on the Kelly criterion, reinforcement learning could be used to determine the stake. Another interesting idea would be to investigate the relationship between calibration and accuracy. Historically, bookmakers' accuracy in predicting NBA game winners is in the region of $69 \pm 2.5\%$ (Hubáček et al., 2019). This begs the question, is there a limit to the accuracy that one can consistently achieve in predicting NBA game outcomes? If it exists, what is the limit that accuracy tends to, as the classwise-ECE tends to zero?

We have established a blueprint for developing a data-driven sports betting system, and shown that when evaluating forecasting models for the sports betting problem, calibration is a more useful metric than accuracy. Beyond the realm of sports betting, calibration may be a more important metric than accuracy in any setting where the predicted probability is more important than the predicted outcome. This applies to many problems, such as weather forecasting and diagnosis of disease. Modellers who spend countless hours trying to increase the accuracy of probabilistic classifiers may be focusing on the wrong metric, and could be better served by trying to minimise the classwise-ECE instead. Practical applications of our results are clear - sports bettors can adopt our blueprint, or modify existing systems to optimise the forecasting model for calibration rather than accuracy, to increase their wealth. Finally, our findings can help bookmakers too. Before setting the odds, the bookmaker generates their own predictions. They can use the classwise-ECE to reveal how far from the true probability their predictions lie. This could be an immensely valuable asset for their risk management team.

10. Acknowledgements

This research builds upon work carried out as part of a dissertation for the degree of Master of Science in Data Science at the University of Bath. The authors would like to extend thanks to Dr. Alessio Guglielmi from the University of Bath for his support and guidance during this process. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors. The authors would like to thank Sports Reference LLC for making NBA data available for researchers and basketball fanatics alike, and the American sportsbook Westgate for making their NBA odds data available on the website sportsbookreviewsonline.com. Finally, the authors are grateful to Dr. KongFatt Wong-Lin from Ulster University for his valuable comments and suggestions.

References

Alonso, R.P., Babac, M.B., 2022. Machine learning approach to predicting a basketball game outcome. International Journal of Data Science 7, 60–77.

Barnett, T., 2010. Applying the kelly criterion to lawsuits. Law, Probability & Risk 9, 139–147.

basketballgeek, 2023. Data. URL: http://www.basketballgeek.com/.

Bunker, R.P., Thabtah, F., 2019. A machine learning framework for sport result prediction. Applied computing and informatics 15, 27–33.

Cao, C., 2012. Sports data mining technology used in basketball outcome prediction .

Cheng, G., Zhang, Z., Kyebambe, M.N., Kimbugwe, N., 2016. Predicting the outcome of nba playoffs based on the maximum entropy principle. Entropy 18, 450.

Cortis, D., 2015. Expected values and variances in bookmaker payouts: A theoretical approach towards setting limits on odds. The Journal of Prediction Markets 9, 1–14.

Cortis, D., 2016. Betting Markets: Defining odds restrictions, exploring market inefficiencies and measuring bookmaker solvency. Ph.D. thesis. University of Leicester.

Darlington, R.B., 1970. Is kurtosis really "peakedness?". The American Statistician 24, 19–22.

databasketball, 2023. Home. URL: https://databasketball.com/.

Delashmit, W.H., Manry, M.T., et al., 2005. Recent developments in multilayer perceptron neural networks, in: Proceedings of the seventh Annual Memphis Area Engineering and Science Conference, MAESC.

Dotan, G., 2020. Beating the Book: A Machine Learning Approach to Identifying an Edge in NBA Betting Markets. University of California, Los Angeles.

Dutta, S., Jacobson, S.H., Sauppe, J.J., 2017. Identifying ncaa tournament upsets using balance optimization subset selection. Journal of Quantitative Analysis in Sports 13, 79–93.

Edwards, W., 1955. The prediction of decisions among bets. Journal of experimental psychology 50, 201.

Fonti, V., Belitser, E., 2017. Feature selection using lasso. VU Amsterdam research paper in business analytics 30, 1–25.

Ganguly, S., Frank, N., 2018. The problem with win probability, in: 2018 MIT Sloan Sports Analytics Conference.

Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q., 2017. On calibration of modern neural networks, in: International conference on machine learning, PMLR. pp. 1321–1330.

Hamadani, B., 2006. Predicting the outcome of nfl games using machine learning. URL http://cs229. stanford. edu/proj2006/BabakHamadani-PredictingNFLGames. pdf .

Hazan, E., Klivans, A., Yuan, Y., 2017. Hyperparameter optimization: A spectral approach. arXiv preprint arXiv:1706.00764 .

Horvat, T., Job, J., 2019. Importance of the training dataset length in basketball game outcome prediction by using naive classification machine learning methods. Elektrotehniški vestnik-Journal of Electrical Engineering and Computer Science, sv 86, 197–202.

Horvat, T., Job, J., 2020. The use of machine learning in sport outcome prediction: A review. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 10, e1380.

Hsieh, C.H., Barmish, B.R., 2015. On kelly betting: Some limitations, in: 2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton), pp. 165–172. doi:10.1109/ALLERTON.2015.7447000.

Hubáček, O., Šourek, G., Železnỳ, F., 2019. Exploiting sports-betting market using machine learning. International Journal of Forecasting 35, 783–796.

Hutter, F., Kotthoff, L., Vanschoren, J., 2019. Automated machine learning: methods, systems, challenges. Springer Nature.

Injadat, M., Salo, F., Nassif, A.B., Essex, A., Shami, A., 2018. Bayesian optimization with machine learning algorithms towards anomaly detection, in: 2018 IEEE global communications conference (GLOBECOM), IEEE. pp. 1–6.

Ivanković, Z., Racković, M., Markoski, B., Radosav, D., Ivković, M., 2010. Analysis of basketball games using neural networks, in: 2010 11th International Symposium on Computational Intelligence and Informatics (CINTI), IEEE. pp. 251–256.

Kelly Jr, J.L., 2011. A new interpretation of information rate, in: The Kelly capital growth investment criterion: theory and practice. World Scientific, pp. 25–34.

Kira, K., Rendell, L.A., 1992. A practical approach to feature selection, in: Machine learning proceedings 1992. Elsevier, pp. 249–256.

Kotsiantis, S.B., Kanellopoulos, D., Pintelas, P.E., 2006. Data preprocessing for supervised leaning. International journal of computer science 1, 111–117.

Kull, M., Perello Nieto, M., Kängsepp, M., Silva Filho, T., Song, H., Flach, P., 2019. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. Advances in neural information processing systems 32.

Kumar, A., Liang, P.S., Ma, T., 2019. Verified uncertainty calibration. Advances in Neural Information Processing Systems 32.

Kumar, V., Minz, S., 2014. Feature selection: a literature review. SmartCR 4, 211–229.

Labayen, V., Magaña, E., Morató, D., Izal, M., 2020. Online classification of user activities using machine learning on network traffic. Computer Networks 181, 107557.

scikit learn, 2022. Randomforest classifier. URL: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html.

legalsportsbetting, 2022. How much money do americans bet on sports? URL: https://www.legalsportsbetting.com/how-much-money-do-americans-bet-on-sports/.

Levitt, S.D., 2004. Why are gambling markets organised so differently from financial markets? The Economic Journal 114, 223–246.

Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R.P., Tang, J., Liu, H., 2017. Feature selection: A data perspective. ACM computing surveys (CSUR) 50, 1–45.

Liang, Y., Liu, C., Luan, X.Z., Leung, K.S., Chan, T.M., Xu, Z.B., Zhang, H., 2013. Sparse logistic regression with a l1/2 penalty for gene selection in cancer classification. BMC bioinformatics 14, 1–12.

Lin, J., Short, L., Sundaresan, V., 2014. Predicting national basketball association winners. CS 229 FINAL PROJECT , 1–5.

Loeffelholz, B., Bednar, E., Bauer, K.W., 2009. Predicting nba games using neural networks. Journal of Quantitative Analysis in Sports 5.

Luo, G., 2016. A review of automatic selection methods for machine learning algorithms and hyper-parameter values. Network Modeling Analysis in Health Informatics and Bioinformatics 5, 1–16.
Miljković, D., Gajić, L., Kovačević, A., Konjović, Z., 2010. The use of data mining for basketball matches outcomes prediction, in: IEEE 8th international symposium on intelligent systems and informatics, IEEE. pp. 309–312.
NBA, 2023. Stats. URL: https://www.nba.com/stats.
Nixon, J., Dusenberry, M.W., Zhang, L., Jerfel, G., Tran, D., 2019. Measuring calibration in deep learning., in: CVPR Workshops.
Pai, P.F., ChangLiao, L.H., Lin, K.P., 2017. Analyzing basketball games by a support vector machines with decision tree model. Neural Computing and Applications 28, 4159–4167.
Pfandzelter, T., Bermbach, D., 2019. Iot data processing in the fog: Functions, streams, or batch processing?, in: 2019 IEEE International conference on fog computing (ICFC), IEEE. pp. 201–206.
Posocco, N., Bonnefoy, A., 2021. Estimating expected calibration errors, in: International Conference on Artificial Neural Networks, Springer. pp. 139–150.
Pratt, J.W., Gibbons, J.D., 2012. Concepts of nonparametric theory. Springer Science & Business Media.
Rotando, L.M., Thorp, E.O., 1992. The kelly criterion and the stock market. The American Mathematical Monthly 99, 922–931.
Salo, F., Injadat, M., Moubayed, A., Nassif, A.B., Essex, A., 2019. Clustering enabled classification using ensemble feature selection for intrusion detection, in: 2019 International Conference on Computing, Networking and Communications (ICNC), IEEE. pp. 276–281.
Smola, A.J., Schölkopf, B., 2004. A tutorial on support vector regression. Statistics and computing 14, 199–222.
Snedecor, G., Cochran, W., 1989. Statistical methods, 8th edn. iowa city.
Snoek, J., Larochelle, H., Adams, R.P., 2012. Practical bayesian optimization of machine learning algorithms. Advances in neural information processing systems 25.
Sports-Reference-LLC, 2022. Basketball statistics and history. URL: https://www.basketball-reference.com/.
sportsbookreviewsonline, 2022. *NBA Odds Archives*. Https://www.sportsbookreviewsonline.com/scoresoddsarchives/nba/nbaoddsarchives.htm.
Sugiyama, M., Krauledat, M., Müller, K.R., 2007. Covariate shift adaptation by importance weighted cross validation. Journal of Machine Learning Research 8.
Thorp, E.O., 1975. Portfolio choice and the kelly criterion, in: Stochastic optimization models in finance. Elsevier, pp. 599–619.
Thorp, E.O., 2008. The kelly criterion in blackjack sports betting, and the stock market, in: Handbook of asset and liability management. Elsevier, pp. 385–428.
Torres, R.A., Hu, Y., 2013. Prediction of nba games based on machine learning methods. University of Wisconsin, Madison .
Tran, T., 2016. Predicting NBA games with matrix factorization. Ph.D. thesis. Massachusetts Institute of Technology.
Wah, Y.B., Ibrahim, N., Hamid, H.A., Abdul-Rahman, S., Fong, S., 2018. Feature selection methods: Case of filter and wrapper approaches for maximising classification accuracy. Pertanika Journal of Science & Technology 26.
Xiao, C., Ye, J., Esteves, R.M., Rong, C., 2016. Using spearman's correlation coefficients for exploratory data analysis on big dataset. Concurrency and Computation: Practice and Experience 28, 3866–3878.
Yang, L., Muresan, R., Al-Dweik, A., Hadjileontiadis, L.J., 2018. Image-based visibility estimation algorithm for intelligent transportation systems. IEEE Access 6, 76728–76740.
Yang, L., Shami, A., 2020. On hyperparameter optimization of machine learning algorithms: Theory and practice. Neurocomputing 415, 295–316.
Zdravevski, E., Kulakov, A., 2010. System for prediction of the winner in a sports game, in: ICT Innovations 2009, Springer. pp. 55–63.
Zhang, G.P., 2000. Neural networks for classification: a survey. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 30, 451–462.
Zhang, J., Jin, R., Yang, Y., Hauptmann, A., 2003. Modified logistic regression: An approximation to svm and its applications in large-scale text categorization .
Zheng, A., Casari, A., 2018. Feature engineering for machine learning: principles and techniques for data scientists. " O'Reilly Media, Inc.".
Zimmermann, A., Moorthy, S., Shi, Z., 2013. Predicting college basketball match outcomes using machine learning techniques: some results and lessons learned. arXiv preprint arXiv:1310.3607 .

Appendix A. Python Packages

Table A.15: Python packages used throughout this work and their purpose.

Package	**Use(s)**
scipy	Kolmogorov-Smirnov tests, calculating kurtosis, two-sample t-tests
sklearn	Predictive modelling
Hyperopt	Automated Hyperparameter Optimisation

Appendix B. Feature Engineering

Table B.16: Basic box score statistics and their descriptions

Box Score Statistic	Description
MP	Minutes Played: Number of minutes played.
FG	Field Goals: Combined number of 2-point and 3-point baskets scored.
FGA	Field Goal Attempts: Number of attempted shots at the basket in-play.
FG%	Field Goal Percentage: Percentage of field goal attempts made (Field Goals/Field Goal Attempts).
3P	3-Point Field Goals: Number of 3-point field goals made.
3PA	3-Point Field Goal Attempts: Number of 3-point field goals attempted.
3P%	3-Point Field Goal Percentage: Percentage of 3-point field goal attempts made (3-Point Field Goals/3-Point Field Goal Attempts).
FT	Free Throws: Number of free throws made. Free throws are primarily awarded if a player is fouled in the process of shooting. A converted free throw is worth one point.
FTA	Free Throw Attempts: Number of free throws attempted.
FT%	Free Throw Percentage: Percentage of free throws converted (Free Throws Made/Free Throws Attempted).
TRB	Total Rebounds: Number of offensive and defensive rebounds collected. A rebound occurs when a player recovers the basketball after a missed field goal or free throw attempt.
ORB	Offensive Rebounds: Number of rebounds collected while playing offense.
DRB	Defensive Rebounds: Number of rebounds collected while playing defense.
AST	Assists: Number of assists made. An assist refers to a pass to a teammate that leads directly to a score.
STL	Steals: Number of steals made. A steal occurs when a player dispossesses an opposition player of the basketball leading to their own team gaining possession of the basketball.
BLK	Blocks: Number of blocks made. A block occurs when a defensive player deflects an offensive player's shot, preventing them from scoring.
TOV	Turnovers: Number of turnovers committed. A turnover occurs when a player loses possession of the basketball to the opposing team.
PF	Personal Fouls: Number of personal fouls committed. A personal foul occurs when a player makes illegal personal contact with an opponent.
PTS	Points: Number of points scored.
+/-	Plus/Minus: Total point differential over the time that a given player was on the court. Seeks to measure a specific player's influence on the game.

Table B.17: Advanced box score statistics and their descriptions

Box Score Statistic	Description
TS%	True Shooting Percentage: A measure of shooting efficiency that takes into account 2-point field goals, 3-point field goals, and free throws.
eFG%	Effective Field Goal Percentage: This statistic adjusts for the fact that a 3-point field goal is worth one more point than a 2-point field goal.
3PAr	3-Point Attempt Rate: Percentage of field goal attempts taken from 3-point range.
FTr	Free Throw Attempt Rate: Number of free throw attempts per field goal attempt.
TRB%	Total Rebound Percentage: An estimate of the percentage of available rebounds a player grabbed while they were on the court.
ORB%	Offensive Rebound Percentage: An estimate of the percentage of available offensive rebounds a player grabbed while they were on the court.
DRB%	Defensive Rebound Percentage: An estimate of the percentage of available defensive rebounds a player grabbed while they were on the court.
AST%	Assist Percentage: An estimate of the percentage of teammate field goals a player assisted while they were on the court.
STL%	Steal Percentage: An estimate of the percentage of opposition possessions that ended with a steal by the player while they were on the court.
BLK%	Block Percentage: An estimate of the percentage of opposition 2-point field goal attempts blocked by the player while they were on the court.
TOV%	Turnover Percentage: An estimate of the number of turnovers committed per 100 plays.
USG%	Usage Percentage: An estimate of the percentage of team plays used by a player while they were on the court.
ORtg	Offensive Rating: An estimate of points produced (players) or scored (teams) per 100 possessions.
DRtg	Defensive Rating: An estimate of points allowed per 100 possessions.
BPM	Box Plus/Minus: A box score estimate of the points per 100 possessions a player contributed above a league average player, translated to an average team.

Table B.18: Features dropped prior to the feature selection process and the corresponding reason for their exclusion.

Feature	Reason for Exclusion from dataset
USG%	This box score statistic only applies to players and has no meaning at the team level. All instances of this feature had 'None' as value.
TRB	Linear combination of other features (ORB + DRB). Therefore supplied redundant information.
3P%	Displayed signs of covariate shift as shown by Kolmogorov-Smirnov test.

Appendix B.1. Demonstration of construction of features

To clarify how we calculate the value, for any given game, of those features derived from team total box score statistics, we consider a hypothetical match between the Boston Celtics and the Chicago Bulls. Let us assume this is the third match of the season for each team, and Boston are at home. To calculate the value for the feature 'DRB' (defensive rebounds), we take the difference between Boston's averaged differences versus previous opponents in DRB and Chicago's averaged differences versus previous opponents in DRB (over the season to date).

Table B.19: Hypothetical calculation of averaged difference in DRB versus previous opponents over season to date for Boston after two games.

Game	**Boston Raw DRB**	**Opponent Raw DRB**	**Difference**	**Averaged Difference**
Game 1	20	18	20-18=2	2/1=2
Game 2	24	16	24-16=8	(2+8)/2=5

Table B.20: Hypothetical calculation of averaged difference in DRB versus previous opponents over season to date for Chicago after two games.

Game	**Chicago Raw DRB**	**Opponent Raw DRB**	**Difference**	**Averaged Difference**
Game 1	16	18	16-18=-2	-2/1=-2
Game 2	20	20	20-20=0	(-2+0)/2=-1

For this hypothetical game, the value of the DRB feature would be $5-(-1)=6$.

Appendix C. Predictive Modelling

Appendix C.1. Feature Selection

Table C.21: Average classwise-ECE achieved under each feature subset

Subset	**classwise-ECE**
Full	5.61%
A	5.06%
B	5.79%
D	5.06%

Table C.22: Average accuracy achieved under each feature subset

Subset	**Accuracy**
Full	64.76%
A	66%
C	65.81%
D	66%

Table C.23: HPO search space for each learning algorithm's hyperparameters, given by their names in sklearn.

Algorithm	**Hyperparameters**	**Type**	**Search Space**
Logistic Regression	C	Continuous	$C \sim \ln \mathcal{N}(0,\ 1)$
	solver	Categorical	{'liblinear', 'lbfgs'}
Random Forest	n_estimators	Discrete	[10,100]
	max_depth	Discrete	[5,50]
	criterion	Categorical	{'gini','entropy'}
	min_samples_split	Discrete	[2,11]
	min_samples_leaf	Discrete	[1,11]
	max_features	Discrete	[1,12]
Support Vector Machine	C	Continuous	[0.1,50]
	kernel	Categorical	{'linear','poly','rbf','sigmoid'}
	degree	Discrete	[2,4]
Multi-Layer Perceptron	hidden_layer_sizes	Discrete	{(3),(4),(5),(6), (3,3),(3,4),(3,5),(3,6), (4,3),(4,4),(4,5),(4,6), (5,3),(5,4),(5,5),(5,6), (6,3),(6,4),(6,5),(6,6) }
	solver	Categorical	{'lbfgs', 'sgd', 'adam'}
	activation	Categorical	{'identity', 'logistic', 'tanh', 'relu'}
	learning_rate	Categorical	{'constant','invscaling','adaptive'}
	learning_rate_init	Continuous	$lr_0 \sim \ln \mathcal{U}(\ln(0.001),\ \ln(0.1))$
	alpha	Continuous	$\alpha \sim \ln \mathcal{U}(\ln(0.0001),\ \ln(0.1))$
	batch_size	Discrete	{32,64,128}

Table C.24: Optimal hyperparameter values for each calibration-optimised model

Algorithm	**Hyperparameter**	**Value**
Logistic Regression	C	0.009
	solver	'liblinear'
Random Forest	n_estimators	62
	max_depth	38
	criterion	'gini'
	min_samples_split	11
	min_samples_leaf	7
	max_features	1
Support Vector Machine	C	23.36
	kernel	'rbf'
	degree	N/A
Multi-Layer Perceptron	hidden_layer_sizes	(6)
	solver	'adam'
	activation	'logistic'
	learning_rate	'invscaling'
	learning_rate_init	0.03
	alpha	0.077
	batch_size	32

Table C.25: Optimal hyperparameter values for each accuracy-optimised model

Algorithm	Hyperparameter	Value
Logistic Regression	C	9.908
	solver	'liblinear'
Random Forest	n_estimators	53
	max_depth	5
	criterion	'gini'
	min_samples_split	8
	min_samples_leaf	1
	max_features	11
Support Vector Machine	C	42.932
	kernel	'linear'
	degree	N/A
Multi-Layer Perceptron	hidden_layer_sizes	(5)
	solver	'lbfgs'
	activation	'relu'
	learning_rate	'invscaling'
	learning_rate_init	0.014
	alpha	0.056
	batch_size	64

Appendix C.3. Combatting randomness with random seeds

Appendix C.3.1. Hyperparameter Optimisation

There is an element of inherent randomness in the BO-TPE process. This means each time the algorithm is run, a different set of 'optimal' hyperparameter values may be found. To combat this, we run the algorithm 10 times (over different random seeds) and record the set of optimal hyperparameters it returns each time. For each of these sets of hyperparameters, we fit the model to the training data over 10 different random seeds and record its score on the validation data each time. The set of hyperparameters under which the model achieves the lowest average score over the 10 runs is deemed to be the optimal set of hyperparameters for the given forecasting model.

Appendix C.3.2. Model Selection

For model selection, we fit each model to an extended training set (consisting of the initial training data combined with the validation data) under the optimal feature set and hyperparameter values for the given branch, over ten different random seeds. A set of predictions is generated for a test set each time. For each data point in the test set, we take the average of the predicted probabilities (across the 10 seeds) as the final predicted probability of the given model for that data point. We then evaluate these predictions under the given metric. Along the calibration branch, the candidate forecasting model which achieves the lowest classwise-ECE on the test set is deemed to be the best calibration-optimised model. Along the accuracy branch, the model which achieves the highest accuracy on the test set is selected as the best accuracy-optimised model.

Appendix C.3.3. Generating predictions for the betting experiments

We fit the models to a final training set over 10 different random seeds, generating predictions for the betting simulation data each time. For each data point, we take its average predicted probability over the 10 seeds as the model's final prediction for that data point.

Appendix D. Hypothesis Testing

The two-sample t-test is used to determine if two population means are equal. A common application is to test if a new process or treatment is superior to the current version. If there is a one-to-one correspondence between the values in the two samples, i.e. if the two samples are $\{X_1, X_2, ..., X_n\}$ and $\{Y_1, Y_2, ..., Y_n\}$ with X_i corresponding to Y_i, then a

paired t-test is conducted by calculating the difference $D_i = X_i - Y_i$. The null hypothesis is $H_0 = \mu_D = \mu_X - \mu_Y = 0$ and the alternative hypothesis is $H_1 = \mu_D > 0$ for an upper-tailed test to see if the new process is superior to the old process. The test statistic is given by:

$$t = \frac{\bar{d} - 0}{s_D / \sqrt{n}}$$

where $\bar{d}$ is the mean of the sample of differences of corresponding points, s_D is the sample standard deviation and n is the sample size.