

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

Conor Walsh, Alok Joshi

Abstract—Sports betting’s recent federal legalisation in the USA coincides with the golden age of machine learning. If bettors can leverage data to reliably 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 predictive 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. We show that optimising the predictive model for calibration leads to greater returns than optimising for accuracy, on average (return on investment of +34.69% versus -35.17%) and in the best case (+36.93% versus +5.56%). These findings suggest that for sports betting (or any probabilistic decision-making problem), calibration is a more important metric than accuracy. Sports bettors who wish to increase profits should therefore optimise their predictive model for calibration.

Impact Statement—Sports bettors have an abundance of publicly available data at their disposal. This data can be leveraged to identify profitable betting opportunities, using machine learning. However, bettors risk losing money if the model’s predictions are unreliable. Bettors generally assume the most reliable predictions come from the most accurate model, when in fact they come from the most well-calibrated model. We show that optimising the sports betting model for calibration leads to greater profits than optimising for accuracy. The returns generated by our calibration-optimised betting systems were 69.86% higher on average, and 31.37% higher when comparing best case returns. With no such comparison between calibration and accuracy present in the literature, our results serve as a benchmark for evaluation of future research. In addition to helping bettors generate greater profits, our findings may also suggest well-calibrated models are more useful than highly accurate models for probabilistic decision-making, generally.

Index Terms—Data mining, Decision theory, Machine learning, Rule-based systems, Uncertainty, Visualization of decision and learning models

I. INTRODUCTION

SPORTS betting in the US is conservatively estimated to be a \$150 billion industry [1]. 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 for 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 predictive 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 [2]. 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 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 simplest 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 [3]. 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 [3]. 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 [4], [5]. Taking the inverse of the odds results in the *implied probability* of the outcome occurring [4], [5]. 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 [3]. In practice, the odds are never fair, as the bookmaker wants to ensure they make a profit [6]. The odds 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

Conor Walsh was with the Department of Computer Science, University of Bath, Somerset, UK. (e-mail: conorwalsh206@gmail.com).

Alok Joshi is with the Department of Computer Science, University of Bath, Somerset, UK. (e-mail: aj2151@bath.ac.uk).

bettors by the bookmaker [3]. Unsurprisingly, each possible 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 probability implied by the bookmaker's odds, the expected value of the bet is positive. We define such a bet as a value bet [7]. Value bets arise when the bookmaker misprices the odds. To spot such opportunities, bettors can compare their model's predicted probability to the bookmaker's odds. We follow this approach and implement and evaluate betting systems which aim to identify value bets and capitalise on them. We measure the success of a system by the return on investment (ROI) achieved over the course of an NBA season.

The predictive model is perhaps the most important part of a sports betting system, and there are many algorithms available to bettors [8], [9], [10], [11], [12], [13], [14]. Prominent examples used in research include support vector machines (SVM), logistic regression (LR), Naive-Bayes, K-nearest neighbours (kNN), decision trees and neural networks [15], [16], [17]. 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. Our hypothesis, that calibration is a more suitable metric in this setting, addresses this gap in the literature. We therefore design competing betting systems, optimising the predictive model for accuracy in one system, and calibration in the other. Comparing the returns achieved by each betting system, we can determine whether optimising 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, optimising the predictive model for calibration, rather than accuracy, leads to greater profits.

The remainder of this paper is organised as follows. Related works are discussed in section II. section III involves a statement of the central hypothesis of the paper and explores the reasons behind the authors' arrival at this hypothesis. section IV lays out the design of an experiment to test this hypothesis. section V covers the novel feature engineering carried out ahead of the predictive modelling, which is covered in section VI. section VII discusses the betting experiments and the algorithm used to conduct the betting simulations in detail. In section VIII, results of the experiments are examined. Finally, section IX discusses the implications of our findings and concludes the paper.

II. 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 [18]. 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 [18]. 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 [18]. 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 [18]. Levitt found that bookmakers generally focus on outperforming the average bettor in outcome forecasting [18]. 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 predictive 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 [19]. 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 [19], [20]. In their work, Bunker and Thabtah deemed this appropriate, noting 'classification accuracy is a reasonable measure of evaluation' for the sports outcome prediction problem [2]. Many different classifiers have been used to address this problem with neural networks among the most widely used [19]. Other classifiers commonly used for the sports outcome prediction problem include SVM and kNN, both of which are widespread in baseball forecasting [21], [19]. When it comes to designing a robust sports prediction 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 [19], [22]. 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 [2], [19]. In general, feature selection and feature extraction are also employed to reduce the dimensionality and complexity of the classification problem [19].

Much of this work has focused on basketball, likely due to the abundance of publicly available data [23], [24], [25], [26]. Researchers have made efforts to identify the most important features and best-performing classifiers. 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 [27]. 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 [9]. 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) [9]. 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 [28]. In a separate study, Lin et al discovered that a team's win/loss record is a crucial predictor of victory [29]. 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 [30]. Another common limitation is that many predictive 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 [19]. 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 generating predictions for pre-game betting can be done using batch processing, and resources for batch processing are much more readily available to the average bettor [31].

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* [32]. The criterion identifies the optimal bet size by maximising the expected value of the logarithmic growth of wealth [33]. While its origins lie in the analysis of long-distance telephone signal noise, the equation has found widespread application across many domains [34], [35], [36], [37], [38]. Sports bettors can use this criterion to calculate optimal bet size as a proportion of overall bankroll [39]. 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):

$$k = \frac{pb - q}{b} \quad (1)$$

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. $q = 1 - p$ [34]

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 [33]. 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 [33]. 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 '5th Kelly' strategy earned an ROI of over 98%, while its full-Kelly counterpart crashed to zero [34].

Excluding accuracy, the lack of metrics used to evaluate the performance of sports prediction models is a noticeable gap in the literature [19]. 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 [3]. One of the key findings of their work was that training models under this loss resulted in greater profits than optimising for accuracy [3]. Further, it has been observed that a highly accurate predictive model is useless as long as it coincides with the bookmaker's model [3].

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.

III. CENTRAL HYPOTHESIS

The purpose of the 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. The predictive model may also play a role in deciding how much to bet, e.g. if the bettor makes use of the Kelly criterion to decide bet size. Therefore, for a betting system to be successful, it is critically important that the probability of victory generated by the 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* [40]. 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. 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 $Y \in \mathbb{Y} = \{1, \dots, K\}$ 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} [41]. Perfect calibration occurs if

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

with reference to the probability over the joint distribution [42].

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 [41]. 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):

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

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 $B_{j,i}$ [41]. This error is bounded between 0 and 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 [43]. 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 error for just the positive class, as this is equivalent to the average error of the two classes.

One of the limitations of the original 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 [44]. 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 [41]. To avoid this scenario, we impose a constraint on model predictions - we require at least 80% of the bins (that the predicted probabilities are grouped into) to be non-empty. This ensures that the distribution of predictions 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 predictive 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.

IV. EXPERIMENT DESIGN

In a data-driven sports betting system, optimising the predictive model for calibration should lead to greater profits 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 predictive models (logistic regression, random forest, support vector machines, multi-layer perceptron). Next, we construct a predictive modelling pipeline (see Fig. 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.



Fig. 1. Predictive Modelling Pipeline: Two branches of the pipeline are presented. The first branch optimises the predictive 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 predictive model for calibration through the same processes, selecting the most well-calibrated model. These branches take NBA data and a set of candidate predictive models as inputs. The output of each branch is the optimal predictive 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. We then implement separate

betting systems for each set of predictions. 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 [34]

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.

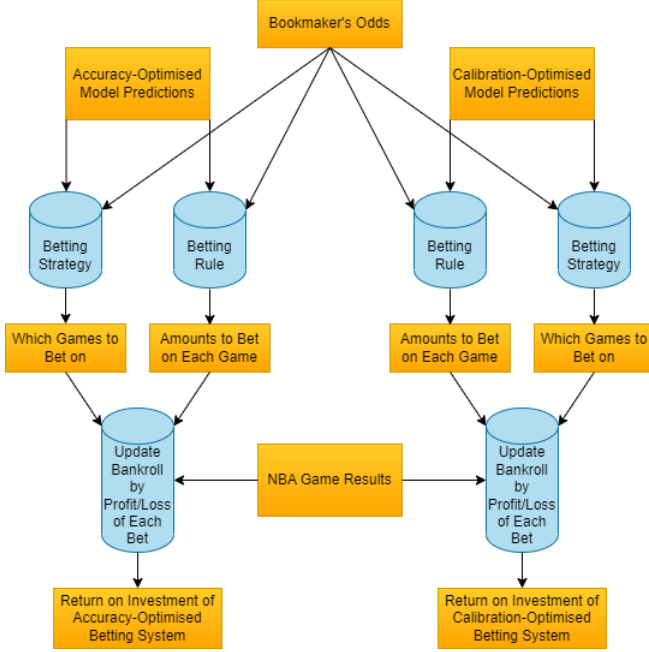


Fig. 2. Betting Simulation Pipeline: Two branches of the pipeline are presented. The first branch represents the simulation for a bettor using a predictive model optimised for accuracy, while the second branch represents the simulation for a bettor using a predictive model optimised for calibration. 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 and colour is given in the caption of Fig. 1.

This experiment is designed to answer the following question: in a data-driven sports betting system, does optimising the predictive model for calibration, rather than accuracy, allow for greater profit generation? To answer this, we examine the ROI achieved by each betting system, to determine whether or not optimising the predictive model for calibration leads to greater profits.

A crucial, preliminary step of the experiment is feature engineering.

V. 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 predicting 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 A (see Tables VIII and IX).

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) [3]. 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 [45]. Therefore, 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 [45]. 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 A (see Tables X and XI).

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 [29]. 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 Winning Percentage’. 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 [3].

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 [46]. 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 [47]. 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 [48]. 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 [49]. 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. The pipeline is discussed in detail in the next section.

VI. PREDICTIVE MODELLING

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

A. Candidate predictive Models

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

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

where $\beta = (\beta_0, \beta_1, \dots, \beta_d)$ are the model parameters to be estimated (with intercept β_0) [50].

The log-likelihood is shown in (5).

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

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) [51]. RF involves the use of bootstrapping to randomly generate various sub-samples of the dataset, before fitting a decision tree to each [52]. To classify a sample, each tree evaluates it individually, and the class which receives the most votes is selected as the final classification result [53].

This is followed by the SVM algorithm. SVMs are supervised learning models which can be used for classification and regression [54]. 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 [55].

For a dataset of size n , the objective function is given by (6) [56].

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

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 [57].

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.

B. Feature Selection

Feature selection (FS) refers to the detection of relevant features and removal of irrelevant, redundant, or noisy data [58]. Various studies have shown the ability of FS to minimise the dimensionality of the problem, maximise the accuracy of classification and prevent overfitting [59], [60], [61]. Of the many FS methods available, we make use of a filter method and a wrapper method in our pipeline. (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 [58]. (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 [58].

(i) Two features are considered highly correlated if their spearman correlation coefficient is greater than 0.7 [62]. 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 Fig. 1), as filter methods are generally considered a pre-processing step [63]. 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. For the calibration branch, the feature subset under which the LR model achieves the lowest classwise-ECE (fitted to an initial training set and evaluated on a validation set) is considered the optimal feature subset for calibration and is denoted as subset B. This subset of features is used for all further modelling along the calibration branch. For the accuracy branch, the feature subset under which the highest accuracy is achieved is considered optimal and is denoted as subset C. Subset C is used for all further modelling along the accuracy branch.

Due to the limitation of the classwise-ECE discussed in section III, we place a constraint on model predictions. 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. To prevent such scenarios, we impose the following constraint on model predictions: if the distribution of bin weights is such that less than 80% of the bins are non-empty, we set the classwise-ECE to 1 (the maximum possible error). This is to ensure that we do not allow models to achieve a low classwise-ECE by generating predictions that are approximately equal to the overall class distribution for all instances. Without imposing this constraint, we could mistakenly identify a sub-optimal feature set as the optimal feature set. This constraint

is also applied to model predictions during the hyperparameter optimisation and model selection processes. Naturally, this constraint does not apply to the accuracy branch.

C. 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 [64]. Many optimisation problems are non-convex or non-differentiable, in which case traditional optimisation techniques may result in a local rather than global optimum [65]. Popular among the non-traditional optimisation techniques that have been used for HPO problems is an iterative algorithm known as bayesian optimisation (BO) [66]. 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 [67]. 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 [20].

For a specified predictive model, along with hyperparameters to be optimised, the search space for each hyperparameter, training and validation data to train and evaluate the model, and an objective function to be minimised, the algorithm identifies the optimal set of hyperparameter values (and returns the corresponding 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 B. HPO is implemented separately along each branch of the predictive modelling pipeline (see Fig. 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 B (see Table XIII).

D. 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 predictive model that 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.

VII. 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 VIII). These two sets of predictions are used to implement competing betting systems, over the given season.

We carry out the betting simulation for a given system as follows. Beginning with an initial bankroll of \$10,000, we iterate through each game in the betting simulation set in chronological order, and for each:

- 1) For each team, we compare the 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
- 2) If the decision is to bet, we determine the stake by the betting rule
- 3) We subtract the stake from the bankroll
- 4) If the bet is successful, we add the stake and the winnings to the bankroll

Both strategy and rule are crucial elements of the betting system. While many possibilities could be explored, this is outside of the scope of this paper. Our goal is to determine whether optimising the predictive model for calibration, rather than accuracy, leads to greater returns, and we do not want differences in ROI achieved by each system to be attributed to other factors such as strategy used, etc. Therefore, we use only the simplest possible strategy and rules.

A. Strategy

We implement the simplest possible strategy: bet on all value bets identified by the model (i.e. bet if a team's predicted probability of victory is greater than the probability implied by the bookmaker's odds).

B. Rules

We implement two simple rules. The first is fixed betting - each time we decide to bet, we set the stake to \$100. The second involves the Kelly criterion - each time we decide to bet, we determine the stake using the eighth-Kelly [33]. As previously discussed, the full-Kelly is considered too aggressive, and leads to almost sure ruin [34], [33]. The eighth-Kelly is a conservative alternative, that recommends betting an eighth of the optimal bet size. This represents a more realistic scenario than fixed betting, as both the odds offered and the bettor's perceived probability of the outcome generally influence the choice of stake [68], [39].

The algorithm used to conduct each betting simulation is shown in Table I.

As discussed, each betting system consists of a strategy and a rule, in addition to a predictive model optimised for either calibration or accuracy. Following the algorithm described

TABLE I
BETTING SIMULATION ALGORITHM

Bankroll = \$10,000

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 $P_h > 1/O_h$:
 $K = (P_h \times \text{Bankroll} - (1 - P_h))/\text{Bankroll}$
 $\text{Stake} = \begin{cases} 1/8 \times K \times \text{Bankroll}, & \text{if Rule} = \text{Eighth-Kelly} \\ \$100, & \text{if Rule} = \text{Fixed Betting} \end{cases}$
 $\text{Bankroll} = \text{Bankroll} - \text{Stake}$
 If home team wins:
 $\text{Winnings} = (\text{Stake} \times O_h) - \text{Stake}$
 $\text{Bankroll} = \text{Bankroll} + \text{Stake} + \text{Winnings}$

Else If $P_a > 1/O_a$:
 $K = (P_a \times \text{Bankroll} - (1 - P_a))/\text{Bankroll}$
 $\text{Stake} = \begin{cases} 1/8 \times K \times \text{Bankroll}, & \text{if Rule} = \text{Eighth-Kelly} \\ \$100, & \text{if Rule} = \text{Fixed Betting} \end{cases}$
 $\text{Bankroll} = \text{Bankroll} - \text{Stake}$
 If away team wins:
 $\text{Winnings} = (\text{Stake} \times O_a) - \text{Stake}$
 $\text{Bankroll} = \text{Bankroll} + \text{Stake} + \text{Winnings}$

Until end of dataset is reached

in Table I, 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.

VIII. RESULTS

To undertake this research, we obtained NBA data from the 2014/2015-2018/2019 seasons from basketball-reference.com [23]. 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 predictive 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 [69].

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 A (see Table IX). Many more features were considered redundant by each of the FS methods employed. The subsets generated by

each of the FS methods, along with the features dropped and features selected by each method are provided in appendix B (see Table XII). Subset B, the optimal feature subset for calibration-optimised predictive modelling, consisted of the following features: 3P, BLK, FT%, ORB, AST%, Previous Season Winning Percentage. Subset C, the optimal feature subset for accuracy-optimised predictive modelling, consisted of the following features: STL%, eFG%, DRB, AST, DRtg, ORtg, 3P, Previous Season Winning Percentage.

Next, HPO was implemented along each branch using the BO-TPE algorithm. The optimal hyperparameter values identified for each model are provided in appendix B (see Tables XIV and XV). 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 II and III.

TABLE II
CLASSWISE-ECE ACHIEVED ON THE TEST SET BY EACH MODEL ALONG THE CALIBRATION BRANCH

| Model | Classwise-ECE |
|------------------------|---------------|
| Logistic Regression | 3.61% |
| Random Forest | 4.39% |
| Support Vector Machine | 3.23% |
| Multi-Layer Perceptron | 3.59% |

The SVM model was identified as the best calibration-optimised model, achieving a classwise-ECE of 3.23% on the test set. This was followed by MLP, with a classwise-ECE of 3.59%, and LR and RF models, with respective classwise-ECEs of 3.61% and 4.39% (see Table II).

TABLE III
ACCURACY ACHIEVED ON THE TEST SET BY EACH MODEL ALONG THE ACCURACY BRANCH

| Model | Accuracy |
|------------------------|----------|
| Logistic Regression | 65.69% |
| Random Forest | 65.34% |
| Support Vector Machine | 66.55% |
| Multi-Layer Perceptron | 65.69% |

The most accurate model was the SVM model with a test set accuracy of 66.55%. The other candidate predictive models achieved accuracies ranging from 65.34% to 65.69% (see Table III). As a result, SVM was identified as the best accuracy-optimised model.

These two final models (calibration-optimised SVM and accuracy-optimised SVM) were used to generate predictions for the 2018/2019 NBA season. Taking as input these predictions along with the bookmaker's odds, we implemented and evaluated our betting systems. Each system was defined by a strategy and a rule. We tested one strategy (bet on all value bets) in combination with two different rules (fixed betting and Kelly betting), as described in section VII. Each simulation was carried out according to the algorithm described in Table I.

A. Fixed Betting

The first rule we tested was the fixed betting rule. In this simulation, for each value bet identified by the model, the

bettor placed a bet of \$100. We compare the performance of the calibration-optimised and accuracy-optimised betting systems in Fig. 3.

In this simulation, both systems identified several losing value bets early in the season (see Fig. 3 games 0-20). These losses were not significant, as under the fixed betting rule, the stake was restricted to \$100 dollars. The systems frequently identified value bets throughout the season, rarely opting not to bet on a game. While the pattern of betting appears similar between the two systems, the success enjoyed by each differed significantly. By the halfway-point of the season, the calibration-optimised system was in the money, up approximately 25% of its initial budget, while its accuracy-optimised counterpart was down approximately 25% (see Fig. 3 circa game 550). This suggests that there was a qualitative difference between the value bets identified by the models. The bets identified by the calibration-optimised model were either successful more often, more profitable, or both, compared to those identified by the accuracy-optimised model. This difference disappeared in the second half of the season, when the systems appeared to consistently place almost identical bets. As a result, the curves resemble mirror images of each other towards the end of the season, and both systems generated positive returns. However, the difference over the first half of the season proved decisive, and the gap between the bankrolls remained significant by season's end. The calibration-optimised betting system ultimately achieved a highly profitable ROI of 32.45%, while the accuracy-optimised system achieved a respectable ROI of 5.56%.

TABLE IV
RESULTS OF FIXED BETTING SIMULATIONS

| Model | Final Bankroll | ROI |
|---------------------------|----------------|--------|
| Calibration-optimised SVM | \$13,244.51 | 32.45% |
| Accuracy-optimised SVM | \$10,556.29 | 5.56% |

B. Kelly Betting

Next, we tested a betting rule based on the Kelly criterion. In this simulation, for each bet identified by the model, the bettor calculated the stake as a function of the predicted probability and the odds offered. This was done using a conservative variation of the Kelly criterion known as the eighth-Kelly. Fig. 4 compares the performance of the calibration-optimised and accuracy-optimised betting systems under this rule.

Once again, both systems identified several losing value bets early in the season (see Fig. 4 games 0-20). This was followed by several spikes and dips for each system. Compared to the fixed betting systems, the bankrolls were much more volatile. This is because the loss on a single bet was capped at \$100 in the fixed betting systems, and capped at 12.5% of the bankroll in the eighth-Kelly betting systems. This increase in volatility exacerbated the effect on the bankrolls of the qualitative difference in bets identified by each system in the first half of the season. As a result, by the time the accuracy-optimised system's bankroll had approximately halved in size, the calibration-optimised system's bankroll had

increased by the same amount. (see Fig. 4 circa game 500). Volatility continued to be ever-present, and the calibration-optimised system increased its bankroll by almost \$15,000 (to reach a peak of approximately \$20,000) over the span of approximately 100 games (see Fig. 4 circa game 800). While the models identified almost the exact same value bets over the second half of the season (as shown in Fig. 3), the curves look not at all similar towards the end of the season. In a Kelly-based betting system, simply identifying the same bets is not enough to achieve similar outcomes. In these systems, the size of the bankroll, as well as the predicted probability and the odds offered on each outcome, are crucial factors affecting the returns achieved. This is reflected in Fig. 4. The accuracy-optimised system consistently diminished as the end of the season approached, ultimately achieving a negative ROI of -75.9%. The calibration-optimised system continued to experience high volatility, but mostly remained in the money, and concluded the season with an impressive ROI of 36.93%.

TABLE V
RESULTS OF KELLY BETTING SIMULATIONS

| Model | Final Bankroll | ROI |
|---------------------------|----------------|--------|
| Calibration-optimised SVM | \$13,692.86 | 36.93% |
| Accuracy-optimised SVM | \$2,409.66 | -75.9% |

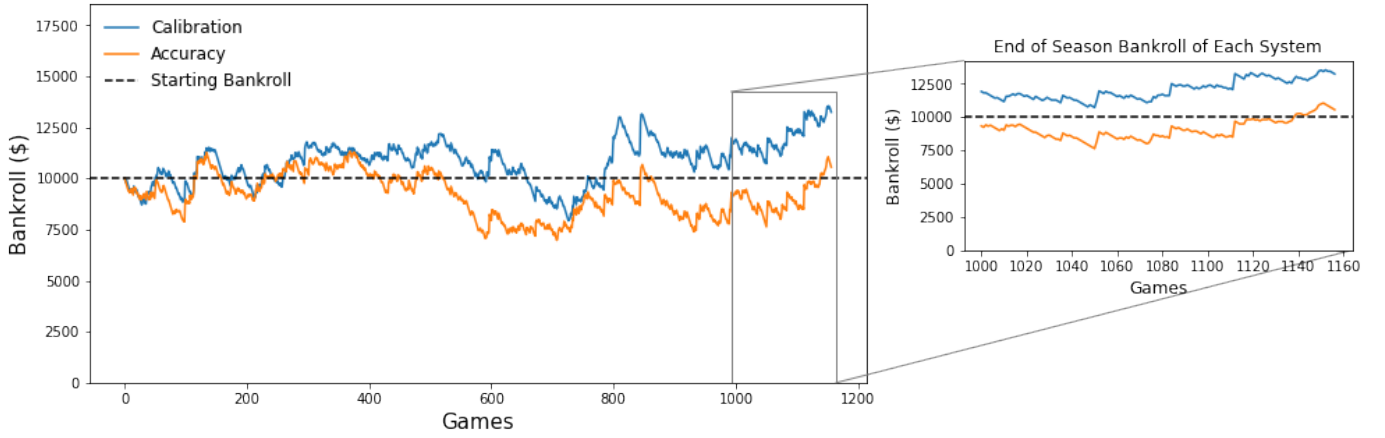


Fig. 3. Bankrolls over the course of the 2018/2019 NBA season under the fixed betting rule. The blue curve represents the bankroll of the betting system using the calibration-optimised predictive model, while the orange curve represents its accuracy-optimised counterpart. The broken black line represents the initial bankroll. Curves which finish above this line earn a profit and are considered successful betting systems.

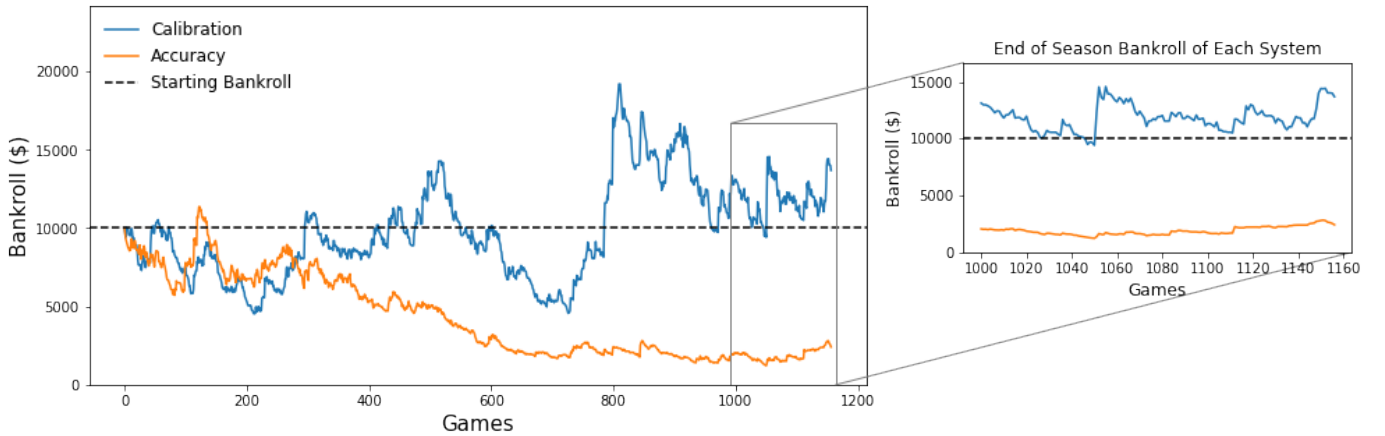


Fig. 4. Bankrolls over the course of the 2018/2019 NBA season under the Eighth-Kelly betting rule. The blue curve represents the bankroll of the betting system using the calibration-optimised predictive model, while the orange curve represents its accuracy-optimised counterpart. The broken black line represents the initial bankroll. Curves which finish above this line earn a profit and are considered successful betting systems.

C. Evaluation of Central Hypothesis

The experiment was designed to answer the question “does optimising a sports betting model for calibration, rather than accuracy, allow for greater profit generation?”. To answer this, we examine Table VI.

Both models frequently identified value bets, with calibration-optimised systems betting on 87.55% of games, and accuracy-optimised systems betting on 89.89% of games. Both models also had similar success rates, with the accuracy-optimised systems winning 38.46% of bets placed, and the calibration-optimised systems winning a slightly higher 38.8% of bets. Unsurprisingly, the accuracy-optimised model was the more accurate of the two (accuracy of 64.62% versus 64.27%), while the calibration-optimised model was more well-calibrated (classwise-ECE of 4.46% versus 5.03%). Ultimately, the metric of primary concern to bettors is ROI. Over the two rules (fixed betting and Kelly betting), the maximum ROI achieved by a calibration-optimised betting

system was 36.93% (achieved using the eighth-Kelly betting rule to determine the stake). The maximum ROI achieved by an accuracy-optimised system was 5.56% (under the fixed betting rule). Calibration-optimised betting systems were also more profitable on average, with a highly lucrative average ROI of 34.69%, compared to an ROI of -35.17% as achieved by accuracy-optimised betting systems on average.

In the next section, we reflect on these results and discuss their implications.

TABLE VI
COMPARISON OF CALIBRATION-OPTIMISED AND ACCURACY-OPTIMISED BETTING SYSTEMS

| Model | % Games Bet On | % Bets Won | Accuracy | Classwise-ECE | Maximum ROI | Average ROI |
|---------------------------|----------------|------------|----------|---------------|-------------|-------------|
| Calibration-optimised SVM | 87.55% | 38.8% | 64.27% | 4.46% | 36.93% | 34.69% |
| Accuracy-optimised SVM | 89.89% | 38.46% | 64.62% | 5.03% | 5.56% | -35.17% |

IX. 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 predictive model in a sports betting system, and that betting systems would be more profitable if calibration was used instead.

To test this hypothesis, we optimised one group of predictive 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 betting systems that identified value bets (betting opportunities where the predicted probability of victory was greater than the probability implied by the bookmaker's odds), and used either a fixed betting rule, or a variation of the Kelly criterion 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 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 predictive 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 predictive model for calibration led to profitable betting systems in all cases, with an average ROI of 34.69%, and an ROI of 36.93% in the most profitable system. In contrast, optimising for accuracy led to an ROI of -35.17% on average and 5.56% in the best case, with the worst-performing system (which used the eighth-Kelly betting rule) losing over 75% of its wealth. This reiterates the danger of using a Kelly-based betting system, especially when the model is not well-calibrated. In the case of our best-performing betting system, we showed that bettors can increase their wealth by more than a third over a single season. These exciting findings support our conjecture that in a data-driven sports betting system, optimising the predictive 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 [3].

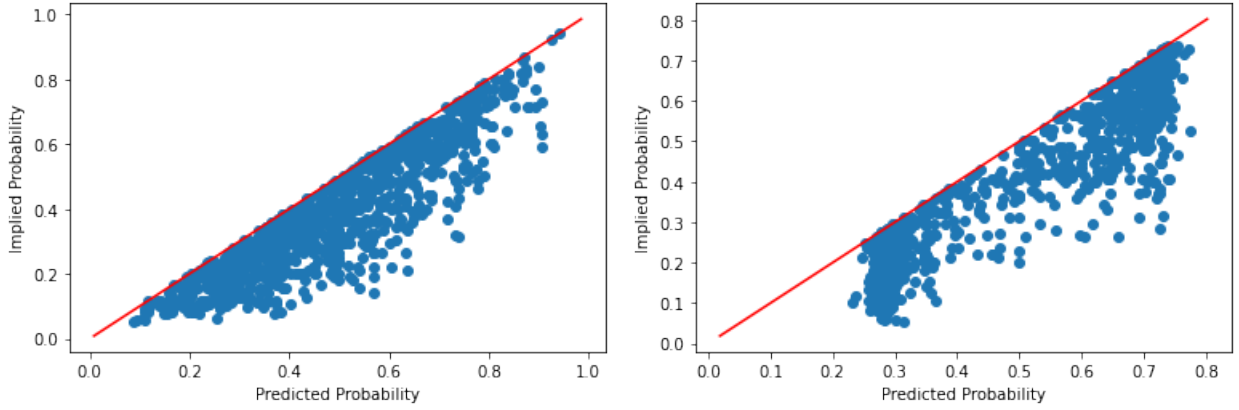
Another interesting finding of our work relates to the

difference in the games identified as value bets by each final model. The accuracy-optimised SVM identified more value bets than the calibration-optimised SVM (betting on 89.89% of games compared to 87.55%) and had a lower success rate (38.46% versus 38.8%). This could suggest a higher number of the value bets identified by the accuracy-optimised model were false positives, i.e. betting opportunities where the odds were not actually favourable. This behaviour suggests the accuracy-optimised model may have been overconfident in its predictions, compared to the calibration-optimised model. This is a well known issue for traditional classification models [70]. It has been shown that in circumstances where models should be uncertain about the label (such as in regions of sparse data), the tendency is to output a more extreme, unrepresentative and overconfident prediction [71]. Examining the distribution of value bets identified by each model, we see evidence of this. Fig. 5 shows a scatter plot of the model's predicted probability versus the probability implied by the bookmaker's odds, for all value bets identified by the calibration-optimised SVM, and the accuracy-optimised SVM, respectively.

We can see from Fig. 5 (a) that the value bets identified by the calibration-optimised model are approximately uniformly distributed (ignoring the extreme ends of the probability range). In contrast, the value bets identified by the accuracy-optimised model are much more unevenly distributed (see Fig. 5 (b)). The bulk of these points lie in one of two regions, forming two clusters towards each end of the probability range, with much fewer value bets found in the middle region. Additionally, this set of predictions exhibits greater variance than its calibration-optimised counterpart (0.036 versus 0.034). We can treat the implied probability as a proxy for the true probability [6]. In this case, the accuracy-optimised model's predictions appear further from the ground truth, compared to the calibration-optimised model. Knowing this, the fact that the accuracy-optimised betting systems lost a higher percentage of bets comes as no surprise, as poorly calibrated models tend to be overconfident on incorrect predictions [72], [42]. This further emphasises the need for a well-calibrated model.

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 (requiring at least 80% of the bins to be non-empty) was somewhat arbitrary, and based on the authors' domain knowledge. Perhaps a better solution could be found to deal with the limitation of the classwise-ECE mentioned in section III.

This research also leaves room for future work. One could experiment to find the optimal betting strategy and rule to max-



(a) Scatter plot of predicted probability versus implied probability for each game identified as a value bet by the calibration-optimised SVM. (b) Scatter plot of predicted probability versus implied probability for each game identified as a value bet by the accuracy-optimised SVM.

Fig. 5. Comparison of scatter plots showing the predicted probability versus implied probability for each value bet identified by the predictive models (calibration-optimised SVM, and accuracy-optimised SVM, respectively). All points lie below the line $y = x$ as a value bet is defined as one for which the model's predicted probability is greater than the probability implied by the bookmaker's odds ($x > y$).

imise profits. 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 $69 \pm 2.5\%$ [3]. 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 predictive 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 used for decision-making. 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. Practical applications of our results are clear - sports bettors can adopt our blueprint 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.

APPENDIX

A. Feature Engineering

1) *Demonstration of construction of features:* To clarify how we calculate the value of features derived from 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 is the home team. 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). The steps involved in these calculations are shown in tables X and XI.

For this hypothetical game, the value of the DRB feature would be $5 - (-1) = 6$ (Boston's averaged difference versus previous opponents in DRB less Chicago's averaged difference versus previous opponents in DRB).

B. Predictive Modelling

TABLE VII
BASIC BOX SCORE STATISTICS AND THEIR DESCRIPTIONS

| Box Score Statistic | Description |
|---------------------|---|
| 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. |
| +/- | 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 VIII
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. |
| FTtr | 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 IX
FEATURES DROPPED PRIOR TO THE FEATURE SELECTION PROCESS AND THE CORRESPONDING REASON FOR THEIR EXCLUSION.

| Feature | Reason for Exclusion from dataset |
|---------|---|
| USG% | Player-level box score statistic with no meaning at team level. |
| BPM | Player-level box score statistic with no meaning at team level. |
| +/- | Player-level box score statistic with no meaning at team level. |
| TRB | Linear combination of other features. |
| FGA | Linear combination of other features. |
| 3PA | Linear combination of other features. |
| FTA | Linear combination of other features. |
| 3P% | Displayed signs of covariate shift as shown by Kolmogorov-Smirnov test. |

TABLE X

HYPOTHETICAL CALCULATION OF AVERAGED DIFFERENCES 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 XI

HYPOTHETICAL CALCULATION OF AVERAGED DIFFERENCES 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 |

TABLE XII

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.8 AND B.9). 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, PREVIOUS SEASON WINNING PERCENTAGE: 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, Previous Season Winning Percentage |
| A | Correlated Feature Removal | TOV%, ORB%, BLK%, TRB%, DRB%, 3PAR, FT, STL, TOV, FTr | FG, FG%, 3P, FT%, ORB, DRB, AST, BLK, PF, TS%, eFG%, AST%, STL%, ORtg, DRtg, Previous Season Winning Percentage |
| B | Correlated Feature Removal and Forward Selection with classwise-ECE as Evaluation Metric | FG, FG%, FT, DRB, AST, STL, TOV, PF, TS%, eFG%, 3PAR, FTr, ORB%, DRB%, TRB%, STL%, BLK%, TOV%, ORtg, DRtg | 3P, BLK, FT%, ORB, AST%, Previous Season Winning Percentage, |
| C | Correlated Feature Removal and Forward Selection with accuracy as Evaluation Metric | FG, FG%, FT, FT%, ORB, STL, BLK, TOV, PF, TS%, 3PAR, FTr, ORB%, DRB%, TRB%, AST%, BLK%, TOV% | STL%, eFG%, DRB, AST, DRtg, ORtg, 3P, Previous Season Winning Percentage |

TABLE XIII
HPO SEARCH SPACE FOR EACH LEARNING ALGORITHM'S HYPERPARAMETERS, GIVEN BY THEIR NAMES IN SKLEARN.

| Algorithm | Hyperparameters | Type | Search Space |
|------------------------|--------------------|-------------|--|
| Logistic Regression | C | Continuous | $C \sim \ln \mathcal{N}(0, 1)$ |
| Random Forest | solver | Categorical | {'liblinear', 'lbfgs'} |
| | n_estimators | Discrete | [10,100] |
| | max_depth | Discrete | [5,50] |
| | criterion | Categorical | {'gini', 'entropy'} |
| Support Vector Machine | min_samples_split | Discrete | [2,11] |
| | min_samples_leaf | Discrete | [1,11] |
| | max_features | Discrete | [1,12] |
| | C | Continuous | [0.1,50] |
| Multi-Layer Perceptron | kernel | Categorical | {'linear', 'poly', 'rbf', 'sigmoid'} |
| | degree | Discrete | [2,4] |
| | 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 XIV
OPTIMAL HYPERPARAMETER VALUES FOR EACH CALIBRATION-OPTIMISED MODEL

| Algorithm | Hyperparameter | Value |
|------------------------|--------------------|-------------|
| Logistic Regression | C | 0.293 |
| Random Forest | solver | 'liblinear' |
| | n_estimators | 35 |
| | max_depth | 10 |
| | criterion | 'gini' |
| Support Vector Machine | min_samples_split | 8 |
| | min_samples_leaf | 4 |
| | max_features | 1 |
| | C | 17.654 |
| Multi-Layer Perceptron | kernel | 'linear' |
| | degree | N/A |
| | hidden_layer_sizes | (4, 6) |
| | solver | 'lbfgs' |
| | activation | 'identity' |
| | learning_rate | 'adaptive' |
| | learning_rate_init | 0.0125 |
| | alpha | 0.0003 |
| | batch_size | 32 |

TABLE XV
OPTIMAL HYPERPARAMETER VALUES FOR EACH ACCURACY-OPTIMISED MODEL

| Algorithm | Hyperparameter | Value |
|------------------------|--------------------|-------------|
| Logistic Regression | C | 1.526 |
| Random Forest | solver | 'liblinear' |
| | n_estimators | 85 |
| | max_depth | 5 |
| | criterion | 'entropy' |
| Support Vector Machine | min_samples_split | 10 |
| | min_samples_leaf | 3 |
| | max_features | 4 |
| | C | 2.01 |
| Multi-Layer Perceptron | kernel | 'rbf' |
| | degree | N/A |
| | hidden_layer_sizes | (5) |
| | solver | 'lbfgs' |
| | activation | 'identity' |
| | learning_rate | 'constant' |
| | learning_rate_init | 0.061 |
| | alpha | 0.028 |
| | batch_size | 64 |

1) Combatting randomness with random seeds:

a) *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 predictive model.

b) *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 predictive 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.

c) *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.

ACKNOWLEDGMENT

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 to 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 Prof. KongFatt Wong-Lin from Ulster University for his valuable comments and suggestions.

REFERENCES

- [1] legalsportsbetting, "How much money do americans bet on sports?" 2022. [Online]. Available: <https://www.legalsportsbetting.com/how-much-money-do-americans-bet-on-sports/>
- [2] R. P. Bunker and F. Thabtah, "A machine learning framework for sport result prediction," *Applied computing and informatics*, vol. 15, no. 1, pp. 27–33, 2019.
- [3] O. Hubáček, G. Šourek, and F. Železný, "Exploiting sports-betting market using machine learning," *International Journal of Forecasting*, vol. 35, no. 2, pp. 783–796, 2019.
- [4] D. Cortis, "Expected values and variances in bookmaker payouts: A theoretical approach towards setting limits on odds," *The Journal of Prediction Markets*, vol. 9, no. 1, pp. 1–14, 2015.
- [5] —, "Betting markets: Defining odds restrictions, exploring market inefficiencies and measuring bookmaker solvency," Ph.D. dissertation, University of Leicester, 2016.
- [6] E. Wheatcroft, "Profiting from overreaction in soccer betting odds," *Journal of Quantitative Analysis in Sports*, vol. 16, no. 3, pp. 193–209, 2020.
- [7] W. Edwards, "The prediction of decisions among bets," *Journal of experimental psychology*, vol. 50, no. 3, p. 201, 1955.
- [8] E. Zdravetski and A. Kulakov, "System for prediction of the winner in a sports game," in *ICT Innovations 2009*. Springer, 2010, pp. 55–63.
- [9] C. Cao, "Sports data mining technology used in basketball outcome prediction," 2012.
- [10] A. Zimmermann, S. Moorthy, and Z. Shi, "Predicting college basketball match outcomes using machine learning techniques: some results and lessons learned," *arXiv preprint arXiv:1310.3607*, 2013.
- [11] R. P. Alonso and M. B. Babac, "Machine learning approach to predicting a basketball game outcome," *International Journal of Data Science*, vol. 7, no. 1, pp. 60–77, 2022.
- [12] G. Cheng, Z. Zhang, M. N. Kyebambe, and N. Kimbugwe, "Predicting the outcome of nba playoffs based on the maximum entropy principle," *Entropy*, vol. 18, no. 12, p. 450, 2016.
- [13] T. Tran, "Predicting nba games with matrix factorization," Ph.D. dissertation, Massachusetts Institute of Technology, 2016.
- [14] P.-F. Pai, L.-H. ChangLiao, and K.-P. Lin, "Analyzing basketball games by a support vector machines with decision tree model," *Neural Computing and Applications*, vol. 28, pp. 4159–4167, 2017.
- [15] B. Hamadani, "Predicting the outcome of nfl games using machine learning," URL <http://cs229.stanford.edu/proj2006/BabakHamadani-PredictingNFLGames.pdf>, 2006.
- [16] D. Miljković, L. Gajić, A. Kovačević, and Z. Konjović, "The use of data mining for basketball matches outcomes prediction," in *IEEE 8th international symposium on intelligent systems and informatics*. IEEE, 2010, pp. 309–312.
- [17] B. Loeffelholz, E. Bednar, and K. W. Bauer, "Predicting nba games using neural networks," *Journal of Quantitative Analysis in Sports*, vol. 5, no. 1, 2009.
- [18] S. D. Levitt, "Why are gambling markets organised so differently from financial markets?" *The Economic Journal*, vol. 114, no. 495, pp. 223–246, 2004.
- [19] T. Horvat and J. Job, "The use of machine learning in sport outcome prediction: A review," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 10, no. 5, p. e1380, 2020.
- [20] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," *Neurocomputing*, vol. 415, pp. 295–316, 2020.
- [21] G. P. Zhang, "Neural networks for classification: a survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 30, no. 4, pp. 451–462, 2000.
- [22] T. Horvat and J. Job, "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. vol. 86, pp. 197–202, 2019.
- [23] Sports-Reference-LLC, "Basketball statistics and history," 2022. [Online]. Available: <https://www.basketball-reference.com/>.
- [24] NBA, "Stats," 2023. [Online]. Available: <https://www.nba.com/stats>
- [25] databasketball, "Home," 2023. [Online]. Available: <https://databasketball.com/>
- [26] basketballgeek, "Data," 2023. [Online]. Available: <http://www.basketballgeek.com/>
- [27] Z. Ivanković, M. Racković, B. Markoski, D. Radosav, and M. Ivković, "Analysis of basketball games using neural networks," in *2010 11th International Symposium on Computational Intelligence and Informatics (CINTI)*. IEEE, 2010, pp. 251–256.
- [28] R. A. Torres and Y. Hu, "Prediction of nba games based on machine learning methods," *University of Wisconsin, Madison*, 2013.
- [29] J. Lin, L. Short, and V. Sundaresan, "Predicting national basketball association winners," *CS 229 FINAL PROJECT*, pp. 1–5, 2014.
- [30] S. Ganguly and N. Frank, "The problem with win probability," in *2018 MIT Sloan Sports Analytics Conference*, 2018.

- [31] T. Pfandzelter and D. Bermbach, "Tot data processing in the fog: Functions, streams, or batch processing?" in *2019 IEEE International conference on fog computing (ICFC)*. IEEE, 2019, pp. 201–206.
- [32] J. L. Kelly Jr, "A new interpretation of information rate," in *The Kelly capital growth investment criterion: theory and practice*. World Scientific, 2011, pp. 25–34.
- [33] C.-H. Hsieh and B. R. Barmish, "On kelly betting: Some limitations," in *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2015, pp. 165–172.
- [34] G. Dotan, *Beating the Book: A Machine Learning Approach to Identifying an Edge in NBA Betting Markets*. University of California, Los Angeles, 2020.
- [35] L. M. Rotando and E. O. Thorp, "The kelly criterion and the stock market," *The American Mathematical Monthly*, vol. 99, no. 10, pp. 922–931, 1992.
- [36] E. O. Thorp, "The kelly criterion in blackjack sports betting, and the stock market," in *Handbook of asset and liability management*. Elsevier, 2008, pp. 385–428.
- [37] —, "Portfolio choice and the kelly criterion," in *Stochastic optimization models in finance*. Elsevier, 1975, pp. 599–619.
- [38] T. Barnett, "Applying the kelly criterion to lawsuits," *Law, Probability & Risk*, vol. 9, no. 2, pp. 139–147, 2010.
- [39] B. P. Jacot and P. V. Mochkovitch, "Kelly criterion and fractional kelly strategy for non-mutually exclusive bets," *Journal of Quantitative Analysis in Sports*, 2023.
- [40] A. Kumar, P. S. Liang, and T. Ma, "Verified uncertainty calibration," *Advances in Neural Information Processing Systems*, vol. 32, 2019.
- [41] M. Kull, M. Perello Nieto, M. Kängsepp, T. Silva Filho, H. Song, and P. Flach, "Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration," *Advances in neural information processing systems*, vol. 32, 2019.
- [42] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *International conference on machine learning*. PMLR, 2017, pp. 1321–1330.
- [43] N. Posocco and A. Bonnefoy, "Estimating expected calibration errors," in *International Conference on Artificial Neural Networks*. Springer, 2021, pp. 139–150.
- [44] J. Nixon, M. W. Dusenberry, L. Zhang, G. Jerfel, and D. Tran, "Measuring calibration in deep learning," in *CVPR Workshops*, vol. 2, no. 7, 2019.
- [45] S. Dutta, S. H. Jacobson, and J. J. Sauppe, "Identifying ncaa tournament upsets using balance optimization subset selection," *Journal of Quantitative Analysis in Sports*, vol. 13, no. 2, pp. 79–93, 2017.
- [46] V. Labayen, E. Magaña, D. Morató, and M. Izal, "Online classification of user activities using machine learning on network traffic," *Computer Networks*, vol. 181, p. 107557, 2020.
- [47] A. Zheng and A. Casari, *Feature engineering for machine learning: principles and techniques for data scientists*. O'Reilly Media, Inc., 2018.
- [48] M. Sugiyama, M. Krauledat, and K.-R. Müller, "Covariate shift adaptation by importance weighted cross validation," *Journal of Machine Learning Research*, vol. 8, no. 5, 2007.
- [49] J. W. Pratt and J. D. Gibbons, *Concepts of nonparametric theory*. Springer Science & Business Media, 2012.
- [50] Y. Liang, C. Liu, X.-Z. Luan, K.-S. Leung, T.-M. Chan, Z.-B. Xu, and H. Zhang, "Sparse logistic regression with a $l_{1/2}$ penalty for gene selection in cancer classification," *BMC bioinformatics*, vol. 14, no. 1, pp. 1–12, 2013.
- [51] M. Injadat, F. Salo, A. B. Nassif, A. Essex, and A. Shami, "Bayesian optimization with machine learning algorithms towards anomaly detection," in *2018 IEEE global communications conference (GLOBECOM)*. IEEE, 2018, pp. 1–6.
- [52] scikit learn, "Randomforest classifier," 2022. [Online]. Available: <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>
- [53] F. Salo, M. Injadat, A. Moubayed, A. B. Nassif, and A. Essex, "Clustering enabled classification using ensemble feature selection for intrusion detection," in *2019 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2019, pp. 276–281.
- [54] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and computing*, vol. 14, no. 3, pp. 199–222, 2004.
- [55] L. Yang, R. Muresan, A. Al-Dweik, and L. J. Hadjileontiadis, "Image-based visibility estimation algorithm for intelligent transportation systems," *IEEE Access*, vol. 6, pp. 76 728–76 740, 2018.
- [56] J. Zhang, R. Jin, Y. Yang, and A. Hauptmann, "Modified logistic regression: An approximation to svm and its applications in large-scale text categorization," 2003.
- [57] W. H. Delashmit, M. T. Manry *et al.*, "Recent developments in multilayer perceptron neural networks," in *Proceedings of the seventh Annual Memphis Area Engineering and Science Conference, MAESC*, 2005.
- [58] V. Kumar and S. Minz, "Feature selection: a literature review," *SmartCR*, vol. 4, no. 3, pp. 211–229, 2014.
- [59] Y. B. Wah, N. Ibrahim, H. A. Hamid, S. Abdul-Rahman, and S. Fong, "Feature selection methods: Case of filter and wrapper approaches for maximising classification accuracy," *Pertanika Journal of Science & Technology*, vol. 26, no. 1, 2018.
- [60] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Feature selection: A data perspective," *ACM computing surveys (CSUR)*, vol. 50, no. 6, pp. 1–45, 2017.
- [61] K. Kira and L. A. Rendell, "A practical approach to feature selection," in *Machine learning proceedings 1992*. Elsevier, 1992, pp. 249–256.
- [62] C. Xiao, J. Ye, R. M. Esteves, and C. Rong, "Using spearman's correlation coefficients for exploratory data analysis on big dataset," *Concurrency and Computation: Practice and Experience*, vol. 28, no. 14, pp. 3866–3878, 2016.
- [63] S. B. Kotsiantis, D. Kanellopoulos, and P. E. Pintelas, "Data preprocessing for supervised learning," *International journal of computer science*, vol. 1, no. 2, pp. 111–117, 2006.
- [64] F. Hutter, L. Kotthoff, and J. Vanschoren, *Automated machine learning: methods, systems, challenges*. Springer Nature, 2019.
- [65] G. Luo, "A review of automatic selection methods for machine learning algorithms and hyper-parameter values," *Network Modeling Analysis in Health Informatics and Bioinformatics*, vol. 5, no. 1, pp. 1–16, 2016.
- [66] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," *Advances in neural information processing systems*, vol. 25, 2012.
- [67] E. Hazan, A. Klivans, and Y. Yuan, "Hyperparameter optimization: A spectral approach," *arXiv preprint arXiv:1706.00764*, 2017.
- [68] U. Matej, Š. Gustav, H. Ondřej, and Ž. Filip, "Optimal sports betting strategies in practice: an experimental review," *IMA Journal of Management Mathematics*, vol. 32, no. 4, pp. 465–489, feb 2021. [Online]. Available: <https://arxiv.org/abs/2107.08827>
- [69] sportsbookreviewsonline, "NBA Odds Archives," 2022, <https://www.sportsbookreviewsonline.com/scoresoddsarchives/nba/nbaoddsarchives.htm>
- [70] J. T.-Y. Kwok, "The evidence framework applied to support vector machines," *IEEE Transactions on Neural Networks*, vol. 11, no. 5, pp. 1162–1173, 2000.
- [71] D. J. MacKay, "The evidence framework applied to classification networks," *Neural computation*, vol. 4, no. 5, pp. 720–736, 1992.
- [72] R. Krishnan and O. Tickoo, "Improving model calibration with accuracy versus uncertainty optimization," *Advances in Neural Information Processing Systems*, vol. 33, pp. 18 237–18 248, 2020.