Supplementary Materials for "Machine learning for sports betting: should model selection be based on accuracy or calibration?"

Conor Walsh, Alok Joshi

February 7, 2025

1 Introduction

Dear Reader,

Thank you for your interest in our research paper, "Machine learning for sports betting: should model selection be based on accuracy or calibration?". To provide a more in-depth understanding of our study and to present additional data that could not be included within the main text, we have created this supplementary document. The purpose of this document is to further enrich your comprehension of our work and its significance.

This supplementary material includes comprehensive details not present in the primary manuscript due to word limitations and other constraints, which are crucial for thorough understanding or reproduction of our research. Our goal is to ensure full transparency and enable fellow researchers to delve into the fine details should they wish to reproduce our study, extend it, or apply our methodology in their own work.

Herein, you will find additional tables that expand upon the ones included in the main article. These tables encompass further breakdowns of our data and more granular insights, that provide a deeper dive into our results. They are designed to support and extend the data discussed in the main manuscript.

Moreover, we provide additional details of our methodology. We hope this information will help readers understand the nuts and bolts of our research design, and offers an opportunity to delve into the technical depths of our work.

Please note that while this supplementary material contains additional information, it is designed to complement, not replace, the information in the main article. We encourage readers to refer back to the primary manuscript when necessary, for a complete understanding of the context and interpretation of these additional materials.

We hope this supplementary document provides a more nuanced understanding of our work, allows for a comprehensive appreciation of our research, and catalyzes further discussions in this exciting field.

2 Feature Engineering

Table 1: 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.
FG%	Displayed signs of covariate shift as shown by Kolmogorov-Smirnov test.
ORB	Displayed signs of covariate shift as shown by Kolmogorov-Smirnov test.

2.1 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 invovled in these calculations are shown in tables 4 and 5.

Table 2: 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 3: 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

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).

Table 4: Feature subsets resulting from each feature selection method. Full list of basic and advanced box score statistics, and their meaning, is provided above (see tables 1 and 2). FG: Field Goals, 3P: 3-Point Field Goals, 3P%: 3-Point Percentage, FT: Free Throws, FT%: Free Throw Percentage, 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 of the main manuscript.

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

Table 5: 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, 3, 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 6: Optimal hyperparameter values for each calibration-driven model

Algorithm	**Hyperparameter**	**Value**
Logistic Regression	C	0.736
	solver	"liblinear"
Random Forest	n_estimators	95
	max_depth	7
	criterion	"gini"
	min_samples_split	9
	min_samples_leaf	6
	max_features	1
Support Vector Machine	C	0.543
	kernel	"linear"
	degree	N/A
Multi-Layer Perceptron	hidden_layer_sizes	(3)
	solver	"adam"
	activation	"identity"
	learning_rate	"invscaling"
	learning_rate_init	0.0027
	alpha	0.0047
	batch_size	32

Table 7: Optimal hyperparameter values for each accuracy-driven model

Algorithm	**Hyperparameter**	**Value**
Logistic Regression	C	3.021
	solver	"lbfgs"
Random Forest	n_estimators	59
	max_depth	5
	criterion	"gini"
	min_samples_split	6
	min_samples_leaf	6
	max_features	3
Support Vector Machine	C	0.789
	kernel	"rbf"
	degree	N/A
Multi-Layer Perceptron	hidden_layer_sizes	(6)
	solver	"lbfgs"
	activation	"identity"
	learning_rate	"invscaling"
	learning_rate_init	0.003
	alpha	0.0006
	batch_size	64

2.1.1 Combatting randomness with random seeds

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.

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-driven model. Along the accuracy branch, the model which achieves the highest accuracy on the test set is selected as the best accuracy-driven model.

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.