# Predicting United States National Basketball Game Spreads using Machine Learning Techniques

Dulani Jayasuriya

Jizhi Liu

Kevin Dow

**Declarations of interest: none**

**Submission of an article implies that the work described has not been published previously**

**This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.**

**Abstract**

We utilize diverse machine learning techniques alongside an OLS regression model as a benchmark to perform out-of-sample forecasts of point spreads in United States (US) National Basketball Association (NBA) games and simulate ensuing profits. Prior literature largely utilizes traditional statistical models to predict NBA game outcomes but fails to make high accuracy predictions due to using limited set of algorithms/ sample size and features. Our findings demonstrate that ML models, particularly Light GBM, outperform traditional linear models in terms of both prediction accuracy and profitability, achieving $150,000 in profit with an initial investment of $100 for the entire year of 2018 NBA sports betting simulations. These insights have crucial implications for both academic researchers, practitioners and other key stakeholders in sports betting, highlighting the potential benefits of using advanced ML models and feature engineering techniques to enhance prediction accuracy and profits.

Keywords: Machine Learning algorithms; Deep Learning; Sports Betting; Outcome Forecasting; Decision Making.

**Acknowledgements:**

Jizhi Liu is a PhD candidate and Dulani Jayasuriya is a Lecturer at the University of Auckland Business School. Kevin Dow is the Dr. Gary Mann Distinguished Professor in Accounting Professor of Accounting Information Systems at Woody L. Hunt College of Business, the University of Texas at El Paso. We would like to express our gratitude to the participants, organizers, and reviewers at the China International Conference in Finance (CICF), the Financial Markets and Corporate Governance Conference (FMCG), the World Finance Conference and various other seminar participants. Any remaining errors are our own.

**Predicting United States National Basketball Game Spreads using Machine Learning Techniques**

## 1. Introduction

Between 150 and 400 billion U.S. dollars annually are invested in the sports betting market, which is one of the fastest growing and innovative industries in the world [3]. Betting on the point spread in basketball games is popular among bettors as it generally provides a higher return than in football games due to low uncertainty [9]. The US NBA is considered as the largest professional basketball league globally, receiving $4.6 billion in revenue per annum, with high economic relevance to the real economy and society [21]. As a result, sports bettors/exchanges, investors, media and even regulators in the sports industry have a vested interest in NBA point spreads and game outcome predictions. However, to date, the majority of bettors are unable to achieve long-term positive returns, especially by following popular Bet Home, Bet Favorite, and Bet Over strategies[1] [21]. Thus, sports betting remains a highly inefficient market with considerable information asymmetry. We choose NBA as opposed to other basketball leagues to avoid match fixing instances. NBA players have massive compensation packages, brand value/endorsements leading to lower incentives for match fixing. Moreover, predicting game outcomes using machine learning techniques with match fixing or other exogenous factors could be an up-hill battle due to their unpredictable nature[2]. Given this setting, we attempt to use NBA betting as an instance of the whole sports betting market on drawing statistical inferences.

[1] The author tests three betting strategies, which are Bet Home, Bet Favorite, and Bet Over. Bet Home involves always betting on home teams due to home-court advantage. Bet Favorite strategy involves betting on a favorite team where the odds are higher than the opponent. Bet Over happens when you bet Over/Under (predict the sum of the total score), and you always choose Over.

2 This issue is prevalent in highly liquid comparatively lower player wages sports. Since, such sports enable low-cost entry points to corrupt players through match fixing. This issue is further enhanced in sports markets rampant with illegal betting, low regulation, monitoring and enforcement [10].

There are three primary methods of betting on an NBA game: the game-winner (Win), total points (Over/under), and point spread (Win against the point spread[3]). We focus on the point spread prediction[4] due to its popularity and potential for a more profitable betting strategy design. Manner [18] implements point spread predictions with simple linear regression models, further refined in subsequent studies, yet are unable to yield sustainable positive returns. Jones [13] extends Manner's [18] work through a linear regression model resulting in marginal improvements in predictive accuracy. Kayhan and Watkins [14] further improve these models and predict point spreads with linear regression models. However, these studies are limited with regard to prediction accuracy, sample size and feature engineering techniques. For example, prior literature uses data from a single game to predict the same game's result or uses eason-averaged data. Bettors are not privy to both these features ex-ante game initiation which is problematic. Moreover, machine learning models are able to identify non-linear relationships between the predicted and input features, unlike linear models, leading to higher accuracy. In addition, ML algorithms efficiently handle large datasets with a myriad of features and are not limited to stringent underlying assumptions suffered by traditional OLS linear regression models. We employ various machine learning (ML) algorithms, including Support Vector Machines (SVM), Decision Trees (DT), Ensemble Methods (XGBoost; LightGBM; Random Forest), Deep Neural Networks (DNN), and Linear Models (OLS, Ridge, Lasso, SGD). For each algorithm, we also test alternative datasets including prior literature data, dimension reduction data, standardlized and normalized data. In sum, we train 165 models in total and select the best

[3] For example, in team A vs team B, team A's point spread is team A's points minus team B's points; team B's point spread is team B's points minus team A's points.

[4]The game-winner prediction means predicting which team will win a game. Total points prediction involves predicting the sum of the final score for both teams. The point spread prediction means predicting the gap in the final score for two teams. The reason we use the point spread prediction is that most bettors invest in point spread betting as opposed to other options [11].

model for each algorithm. The detailed results of our extensive analysis are included in Internet Appendix C.

Subsequently, we pose the following research questions: Which side to bet (Home or Away) based on point spread betting for a single game? To address this question, we apply ML models, which lead to the following sub-research questions: (1.) Does sports betting based on point spread prediction have a positive return[5]? (2.) Could ML algorithms outperform traditional models with regard to prediction accuracy and profit generation? (3.) What new features would be better predictors that increase model accuracy?

The contributions of our study are manifold. Firstly, we implement an extensive collection of the most prominent ML algorithms to predict NBA point spreads[6]. Secondly, we contribute to feature engineering techniques on sports betting, which improves accuracies relative to prior literature. Thirdly, we set a new benchmark in terms of higher accuracies which translates to economic significance. Fourthly, our study aid bettors in making decisions on point spread betting.

Our findings demonstrate that machine learning models, particularly Light GBM, outperform traditional linear models in terms of both prediction accuracy and profitability. Our models achieved 150,000 dollars with an initial investment of 100 dollars for 2018 sports betting simulation. Moreover, our results indicate that employing feature selection, dimension reduction, and feature scaling can further improve model performance. New variables and feature engineering techniques are significantly better than prior literature in terms of improving prediction accuracy. These insights have important implications for both academic researchers and practitioners in sports betting, as they highlight the potential

[5] The point spread is defined as the difference between two teams' points.
[6] This research uses averaged team-based features for the last ten games before a game starts so that bettors can obtain required data and simulate the point spread before a game begins. Another reason for using the last ten games average dataset is that teams' abilities may change during a particular season, hence, a recent dataset would be more appropriate for accurate prediction. However, this number cannot be too small due to opponent bias.

benefits of employing advanced machine learning models and feature engineering techniques to enhance prediction accuracy and profits.

## 2. Literature Review

The primary objective of this study is to determine the optimal side to place bets on by identifying key features and developing robust forecasting models for point spread predictions. This section reviews existing literature on NBA game outcome predictions, which remains relatively scarce. Three principal methods exist for betting in the NBA sports betting market: predicting the winning team, determining whether the total points scored by both teams exceed or fall below a particular threshold set by bookmakers, and ascertaining if the point spread is above or below specific points established by bookmakers. Table 1 provides a summary of the main findings for each betting method.

Previous studies have predominantly focused on predicting either point spreads or game outcomes. A significant limitation of predicting game-winners is the restricted profitability. To elaborate, some teams consistently outperform their opponents, leading the majority of bettors to wager on the superior team. Consequently, the odds for the favored team become overvalued, causing bettors who predict game-winners to consistently bet on overvalued odds, resulting in limited profitability. Moreover, predicting game-winners poses challenges due to the vast number of games available for betting each day. For instance, if there are ten games in a day, bettors cannot simply choose a single game to bet on, as predictions are only based on which team wins or loses and not the margin of victory. In contrast, if bettors know the anticipated margin of victory, they can select the game with the most significant deviation from the line for a safer bet and higher profits. Furthermore, most prior literature relies on single-game data to predict the same game's outcome. As a result, compared to game-winner predictions, point spread predictions can help bettors avoid overvalued odds and select undervalued odds, leading to substantial profits. Additionally, our

study employs point spread predictions, as spreads can be further elucidated by differences in team abilities, as identified by the integration of both player and team-level datasets.

Over one-third of prior literature does not explicitly detail any feature engineering techniques employed in their analyses. The remaining literature primarily utilizes three techniques:

Employing a single game's features to predict the same game's outcome (e.g., using one game's Rebound, Steal, and Field Goal Percentage to predict the same game's point spread or outcome). The issue with this approach is the difficulty in determining these features' values before a game concludes, rendering real-time, out-of-sample predictions impossible.

Using a single season's average game features to predict a single game's outcome within that season. This method assumes that bettors would have ex-ante access to data or features unavailable prior to a game, including data from subsequent games. Our study addresses this limitation by using features to predict a game that would be available ex-ante, rather than ex-post.

Employing the past five or eight games to create average features, potentially leading to opponent bias. Therefore, there is significant research potential in exploring novel feature engineering techniques that could contribute to sports betting markets.

Regarding datasets, prior literature has used four primary datasets at various levels: player-level, in-game-level, team-level, and combined player and team-level data. However, substantial limitations persist when using only player or team-level data and disregarding other relevant information. Two common drawbacks of in-game datasets include intensity and high costs. Once a game commences, bettors must closely monitor the game and continually adjust inputs to fully utilize the extensive array of in-game data for their models. Moreover, obtaining timely, accurate, and precise in-game data presents challenges in terms

of intensity, cost, and accessibility, particularly for average bettors. Additionally, odds can fluctuate rapidly for in-game betting, while predictions may take time to generate (even with exceptional hardware and resources). Consequently, when bettors eventually receive game outcome predictions from their models, the odds may have already shifted. Lastly, prior literature relies on player and team-level data only for the top three players to measure their abilities, which inadequately represents the rich features and interactions available that contribute to game outcomes. In contrast, our study incorporates both team and player-level data for all players.

We employ various machine learning (ML) algorithms, including Support Vector Machines (SVM), Decision Trees (DT), Ensemble Methods (XGBoost; LightGBM; Random Forest), Deep Neural Networks (DNN), and Linear Models (OLS, Ridge, Lasso, SGD). To the best of our knowledge, our study is the first to implement all these ensemble methods and DNNs for predicting NBA point spreads. As noted by Suk et al. [30], Ensemble Methods, which combine several base ML models, typically outperform individual ML models. Consequently, instead of creating multiple base models and comparing their accuracy, Ensemble Methods aggregate several models and leverage each base model's strengths to construct one superior model [19]. DNNs are also crucial, as they can identify polynomial relationships between dependent and independent variables. DNN architectures create complex models with a clear, simple target and multiple layers. Additional layers allow lower layers to model complex data with fewer units than shallow networks while maintaining similar performance [33]

## 3. System Design

To address our earlier mentioned research questions, we build the NBA Games Forecasting system as shown in Figure 1:

The NBA Games Forecasting System consists of several major components. Firstly, we gather historical games and odds data and conduct feature selection using dimension reduction techniques. Secondly, we transform the data and create two standardized and normalized data sets. Thirdly, we split the data set into train and test sets where hyper-parameters are tuned using a cross-validation method. All prediction results are out-of-sample and based on the test set. Finally, the profits are simulated based on the out-of-sample predictions and Odds data.

## 4. Experimental Design

This section demonstrates our methodology[7]. The daily point spread is collected for each team per game in the NBA from Kaggle,[8] where six seasons are covered (2012 through 2018). This dataset only includes regular seasons and covers all 30 NBA teams. Thus, we compile a comprehensive NBA dataset for 7379 games with an extensive collection of team-level and player-level features[9].

### 4.1 Feature Engineering

Feature engineering is the process of creating features using domain expertise, which is essential for making accurate predictions. We employ two datasets (Player-level and Team-level datasets) to generate relevant features for point spread prediction. The Player-level

[7] For further details of data and methodology, please visit our GitHub page: -- *https://github.com/amazingzhi/PointSpreadPrediction0*
[8] *https://www.kaggle.com/pablote/nba-enhanced-stats.*
[9] Team-level and player-level datasets are combined from *https://www.kaggle.com/pablote/nba-enhanced-stats* and *https://www.kaggle.com/nathanlauga/nba-games*.

dataset records individual players' performance in each game, while the Team-level dataset documents the performance of each team. Predicting the point spread necessitates considering both the team's and players' historical performances. For an in-depth explanation of our Feature Creation process, please refer to Internet Appendix A.

The Odds Data, which chronicles bookies' point spreads with their odds before a game commences, is also used from 2004 to 2018. This dataset is leveraged to simulate profits and the out-of-sample point spread predictions. Figure 2 illustrates our variable of interest and the feature engineering process.

### 4.2 Prior Literature Variables

Prior literature variables are employed to run the same algorithms and compare the results of previous studies. The datasets of this study and the variables from previous studies are juxtaposed to ascertain improvements in prediction accuracy. Results are presented in Table 4. The comparison between prior literature variables and this research's variables is shown in Internet Appendix G.

### 4.3 Feature Selection and Dimension Reduction

Feature selection is advantageous for identifying key features that enhance prediction accuracy. This method ranks the contributions of each feature to the target variable. We then select essential features and input these features to algorithms as an alternative approach compared to inputting all features to algorithms. Optimizing the set of key features and eliminating unnecessary features can significantly improve prediction accuracy. Thus, we identify the key features using the Random Forest algorithm. The selected features are subsequently employed to implement the ML models, and accuracy is compared with all features and PCA features of ML model performance.

Dimension Reduction is another approach for reducing the number of features with similar motivations as feature selection, such as decreasing training time and enhancing

model accuracy. We apply Principal Component Analysis (PCA) to generate the Dimension Reduction dataset. The PCA algorithm diverges from feature selection in that instead of removing features from the original dataset, PCA applies a linear transformation to the original dataset to decrease the number of features. The PCA method's objective is to attempt to reduce the number of features while preserving the explained variance. The explained variance measures how much variance of one dataset can be explained. It is a number between 0 and 1, where 1 indicates that no information is lost from the original dataset after PCA transformation, and 0 implies a complete loss of information.

Following feature selection and Dimension Reduction, we obtain three datasets to be used in ML algorithms: the original dataset, the feature-selected dataset, and the dataset with dimension reduction. The next step involves cleaning and other data pre-processing methods.

### 4.4 Sample Splitting, Feature Scaling

Feature scaling and sample splitting are two crucial data pre-processing methods employed before implementing ML algorithms. Sample splitting is essential for testing the predictive performance of algorithms and tuning hyperparameters based on the validation dataset [25]. Prior literature also indicates that feature scaling improves the performance of ML algorithms [35].

For sample splitting, the data were divided into a training sample (6149 observations) consisting of the first five seasons and a 20% test sample (1230 observations) comprising the last season. This particular sample splitting method was chosen to ensure that the model used historical data to predict unobserved data while maintaining the data's temporal order for prediction purposes. The Empirical Results section presents test accuracy results.

### 4.5 Performance Evaluation

We employ accuracy measures to compare predictions with actual values, including R-squared value, explained variance, MAE, and max error. R-squared value and explained

variance assess the proximity of our predictions to the actual values in percentage terms. MAE measures the average distance between our predictions and the actual values. We consider max error to identify the acceptance of extreme cases. After comparing each model's performance measures, we select the best ML model for point spread predictions and use it for profit simulations.

### 4.6 Hyperparameters tuning

Tuning hyperparameters is beneficial for identifying the best hyperparameters to achieve improved results and regularization procedures [23]. We use Random Search to detect the best hyperparameters for different ML algorithms by randomly selecting combinations of hyperparameters during the tuning process. Grid Search involves utilizing every possible combination of hyperparameters to find the best accuracy, while Random Search involves selecting a subset of hyperparameters randomly and finding the best accuracy. Grid search results in finding the optimal combination of hyperparameters due to utilizing every combination, while Random Search is more efficient in terms of time and resource usage. Accuracy measures used in the tuning process include the best MAE, RMSE, and explained variance.

### 4.7 Linear Algorithms

We first implement the simple linear predictive regression model estimated through OLS estimation. The expected performance of this model is poor for high-dimensional relationships. We use this model as the traditional benchmark model to compare with ML algorithms. Linear models assume [4]**.** that the objective variable *PointSpread* $f(x_{i,t};\theta)$ can be achieved by a linear function $f$ based on the original dataset's features $x_{i,t}$ and the parameter vector, $\theta$.

$$f(x_{i,t};\theta) = x_{i,t}\,\theta \tag{1}$$

This model only allows a linear relationship and does not allow a non-linear relationship between features.

Algorithms used in this paper include OLS, Ridge, Lasso, and SGD.

**4.8 Support Vector Machine Regressor (SVMR)**

Although the benchmark OLS model can measure how close observations fit an expected line or curve, the linearity of the relationship between the inputs and outputs is an underlying assumption of the model [22]. Some ML algorithms enable the identification of non-linear relationships, which may result in higher accuracy.

The first ML algorithm we implement is SVMR applied to a linear or non-linear regression problem [32] to identify the relationship between the target variable and features. Compared to OLS, SVMR minimizes the l2-norm of the coefficient vector instead of the squared error term. By this error estimation, SVMR generates a relationship between the target variable and other features with the acceptance of extreme values which can easily affect a linear regression [1]. The acceptance level is called the soft margin which could be set to a specific threshold. SVMR applies the cross-validation method to select the minimum error's soft margin on average as the best soft margin.

Below is a formula that calculates the l2-norm, and the target is to minimize it [38].

$$\text{Minimize } J(\beta)=1/2\ \beta'\beta+A\ (\xi_n+\xi_{*n}), \tag{2}$$

Subject to:

$$\forall n : y_n - (x_n'\beta+b) \leq e + \xi_n$$

$$\forall n: (x_n'\beta + b) - y_n \leq e + \xi_{*n}$$

$$\forall n: 0 \leq \xi_{*n}$$

$\forall n$: $0 \leqslant \xi_n$.

The original version of the l2-norm is calculated by 1/2 $\beta'\beta$. However, to allow for errors, this function always adds the $A$ ( $\xi_n$+$\xi_{*n}$) slack. The constant $A$ is a box constraint, which is positive, used to control the penalty imposed on the sample applied outside the soft margin $e$, and aids in stopping overfitting. $y_n$ and $x_n$ are the target value and training sample. The inner product plus intercept $x_n'\beta$+$b$ is the prediction for that sample [8].

**4.9 Decision Tree (DT) Regression**

DT is a versatile ML algorithm useful for classification and regression problems [34] and can fit complex datasets [34]. DT is also the fundamental component of Random Forests, implemented later in this study. We apply DT as a regression tool to investigate the relationship between the target variable and features. The DT algorithm calculates the entropy of the target variable for each feature to identify features that reduce entropy to a minimum level. The information gain (IG) negatively correlates with entropy, and the objective is to maximize the IG for each feature. After DT considers all features, the IG accumulates to a high level leading to optimal prediction accuracy.

The DT regression algorithm treats features one by one. For each continuous feature, DT gives thresholds (values in the middle of this continuous feature) to calculate the IG of the target variable. We call these thresholds division points of a feature. DT links all these features together to make a tree. Therefore, for each of the division points, we have a parent node (division point itself), left child node (value below this threshold), and right child node (value above this threshold). By going over each feature, the IG of the target variable keeps increasing to maximize it.

An objective function is determined to optimize the DT algorithm to divide all nodes for more features' IG. The objective function maximizes the IG at every split (for each feature), calculated as follows [24].:

$$IG\ (Z_a, q) = W(Za) - (X_{left}/X_a * W(Z_{left}) + X_{right}/X_a * W(Z_{right})) \quad (3)$$

$q$ represents the feature at the division point, $Za$, $Z_{left}$, and $Z_{right}$ are the datasets of the parent and baby nodes, $W$ is the impurity measure, $X_a$ is the observations' number at the parent node, and $X_{left}$ and $X_{right}$ are observations' number in the baby nodes.

**4.10 Ensemble Learning Algorithms (XG Boost, Light GBM, and Random Forest)**

In most instances, scholars suggest that the combined answer with discussion provides better performance than one answer from one expert [15,17]. Similarly, several predictors are utilized in Ensemble Learning to improve prediction accuracy. This is implemented by combining several models [28] instead of a single model [28].

Different Ensemble methods are used in our study. The Random Forest algorithm fits Decision Trees on different subsets of the dataset and then averages predictions for all subsets [31]. XGBoost and LightGBM belong to Boosting Ensemble Learning ML algorithms, where Boosting includes sequentially adding ensemble members that correct prior model predictions and produce a weighted average of the predictions [31]. Similar to the OLS, we measure the performance of our models developed by Ensemble Learnings, as explained in section 4.4.

**4.11 Deep Neural Networks (DNN) Algorithm**

The final algorithm used is the Neural Networks (NN). NN, also called deep learning, shows flexibility by twisting numbers of contracting layers of non-linear relationships between dependent and independent variables [40]. The advantage of using NN is that it can imitate any relationship between features and the target variable by training models iteratively [41]. Model performance measurement is explained in section 4.4.

**Model.** We implement and apply the conventional 'feed-forward' neural network. This network includes an input layer that feeds features into several hidden layers that identify complex relationships between independent variables and numbers of 'neurons' in different hidden layers. The output layers represent the dependent variables linked to the last layer of the hidden layers.

Model Explanation.

$$Bias\_Neuron_1 + W5*In_5 + W_4*In_4 + W_3*In_3 + W_2*In_2 + W_1*In_1 = Z_1 \quad (4)$$

Sigmoid($Z_1$) = $Neuron_1$ Activation

The general version of the model between the front (m neurons) and the next layers (n neurons) is shown as follows:

$$[W_{mn}] @ [X_m] + [Bias] = [Z]$$

### 4.12. Profits Simulation

After we train models for each ML algorithm, we select the best model for each ML algorithm. We use these selected models to predict point spreads for our out-of-sample test dataset (NBA 17-18 season)[10]. These predictions are used to simulate profits using Odds Data as well. Our Odds data include odds over and under specific point spreads provided by bookmakers. Both Odds over or under are very close to 1.9, which means betting one dollar would result in 1.9 dollars being returned (win this bet) or losing this one dollar (lose this bet). In this case, the probability of winning a bet or breaking even is 1/1.9, equaling 52.63%. Therefore, any ML algorithm that generates predictions providing over 52.63% winning probability would be recognized as a profitable ML algorithm.

We apply the Kelly Criterion [36] to maximize our profits, which determines the optimal theoretical size for a bet. This size can be calculated as:

[10] Team ranks do not vary considerably at the end of season which means that their winnings or losses will not be a decisive factor on whether they make playoffs or not. Teams that cannot make the playoff may intentionally lose games to get better new players in the next season's NBA draft. As a result, we don't use our predictions at the end of 17-18 season.

$$f = p + (p - 1) / b \quad (5)$$

where $f$ is the fraction of the current bankroll to wager,

$p$ is the probability of a win,

$b$ is the odds minus one.

For each ML algorithm, we calculate its probability of a win as $p$ and use 0.9 as $b$ since the odds are very close to 1.9. We calculate $f$ by substituting $p$ and $b$ in equation (5). We start our bankroll of 100 dollars and keep using f as the fraction of the current bankroll to wager and track our bankroll/funds.

Another factor to consider is which game to bet on. There are several games to bet on daily. Therefore, selecting the safest one would be the least risky option. We identify the safest game of a given day by calculating the difference between our predictions and bookmakers' spreads and identify the game with the largest margin as the best option to bet. Traditional investment theory argues that higher risk would be followed by higher returns [2]. However, this may not always be the case in point spread betting. No matter how far your prediction is away from the bookmaker's point spread, the return odds would always be 1.9. Finally, we add always bet on the home team algorithm to compare our models' performance relative to the traditional betting strategy.

## 5. Empirical Results

### 5.1 Descriptive Statistics

Table 2 summarizes the original numerical attributes for the top ten important features, including the number of observations, mean, standard deviation, minimum, and maximum values. The mean *pointspread* equals 2.69, suggesting a home team advantage of

2.69 points per game. Summary statistics between the train, validation and test datasets are compared to identify whether they share similar attributes[11].

Table 3 summarizes the Odds data depicting the number of observations, mean, standard deviation, minimum, and maximum values. Although the mean of odds is smaller than 1.9, for all percentiles, they are all 1.90909. The reason is that some odds are recognized as 0 due to missing values, which influences the mean.

### 5.2 Feature Selection and Principal Component Transmission

Feature importance is another significant result for all three datasets. We identify the feature importance based on normalization, standardization, and prior literature datasets. However, applying different ML algorithms may result in different features being selected as being important. These results are omitted for brevity and are available in the Internet Appendix section F. Thus, only tree-based algorithms' top ten feature importance values and prior literature are presented in our study.

Figure 3 compares the difference between prior literature's top ten important features and our study. We applied the Random Forest algorithm to rank these features. This algorithm shows each feature's importance to the variable of interest (point spread) by measuring point spread's entropy decreases for each feature's addition to the model. Thus, we differentiate between features ranked high as being used in prior literature or a new addition from our study. Subsequently, following new variables such as player ability, team efficiency, point spread averaged over the last ten games, and defensive or offensive ratings, a significantly stronger relationship is found than traditional variables such as points or field goals. Moreover, as explained earlier, feature selection could lead to higher model accuracy. Therefore, we select 75 (number of features in feature selected dataset) out of 108 (number of

[11] For all variables' summary statistics for the normalized and standardized datasets are provided in Internet Appendix section B.

features in original dataset) features to compile our final feature selected dataset, which is used to train our models.

Another dataset utilized in our study involves the PCA dataset, where we apply linear transformations of our original dataset to reduce the original number of features based on PCA. Figure 4 shows the relationship between explained variance (Y-axis) and the number of components (X-axis). Our objective is to reduce the number of components while maintaining the explained variance close to 1. We decide to use 40 PCA identified features where the explained variance was kept at 0.95. This dataset is also used to train our models.

### 5.3 OLS and Machine Learning Algorithm Results

Although OLS aids in finding a linear relationship between the dependent and independent variables, it fails to identify non-linear relationships or perform well with a large number of features. As a result, ML algorithms could be an alternative set of algorithms that can supplement further analysis as explained in or study. Firstly, we compare ML algorithms' results versus the OLS benchmark model results, and secondly, we compare ML algorithms' results with prior literature results regarding different feature engineering techniques.

Table 4 summarizes the best model's accuracy results in terms of OLS and all ML algorithms with prior literature's features to compare OLS and ML model performance. Column one shows algorithms used and prior literature's variables, while other columns record accuracies.

Two points are of particular interest for each algorithm and are worth mentioning during model training. Firstly, we train our models based on various targets: MAE, RMSE, and Explained Variance. Secondly, we further train our models based on many features'

datasets due to feature selection, dimension reduction and feature scaling techniques[12]. For details of the accuracy results for all these options, please refer to Internet Appendix section C:

According to Table 4, it can be observed that there is no significant difference between the accuracies between linear regressions and ML algorithms. However, each algorithm has different performance dynamics for different feature sets as expected. Moreover, Table 4 shows that the accuracy measures of this study's dataset perform better than prior literature's variables dataset indicating that our set of features increases prediction accuracy. Although, none of these models performs well in terms of R squared[13], the key to making profits is not about how close your predictions are to the actual point spread, but how better your predictions can be than other bettors. Hence the popular saying, "Don't play against the house, play against the other player." Thus, we use these models identified as best in terms of accuracy for our profit simulation. For best model visualizations, please refer to the online Internet Appendix section D:

## 6. Discussion

### 6.1 Residual Check

Another way to assess the quality of predictions is by examining their residuals. Figure 5 presents the point spread predictions of different algorithms in density plots, with blue lines representing the residual density lines and black lines indicating normal

[12] In sum, we have six datasets for each algorithm. They are normalized original, feature selected, PCA datasets, as well as standardized original, feature selected, PCA datasets.

[13] We compare the mean of our train and test dataset's target variable. The mean of train target variable is 2.81 while the mean of test target variable is 2.11. we further calculate $\sum(yi-\bar{y})2/n$ for train and test target variable which are 179.115 and 186.3049. The calculation of R square value is $1-\sum(yi-\hat{y}i)2/\sum(yi-\bar{y})2$. Therefore, the R square value is relatively low for our test dataset as the test set is considerably different from the train set further cementing that our results are truly out of sample and similar to real world situations.

distributions. Figure 5 suggests that most of our models' predictions closely resemble a normal distribution.

### 6.2 Betting in Reality

We simulate our profits according to the Kelly Criterion and match our test data with Odds data based on game identifications. As a result, there are some missing values resulting from Odds data[14]. Figure 6 shows the cumulative sum of our profits for all algorithms in our test dataset (01/01/2018 – 12/31/2018). The initial stakes of all models start from 100 dollars. Results indicate that ML algorithms achieve better profits than the benchmark Linear Models. Our models provide significant profits in 2018 NBA games. The best model happens to be Light GBM which achieves 150,000 dollars from only 100 dollars investment.

## 7. Implications

Our findings shed light on informational asymmetries, inefficiencies/processes and underlying interactions between gamblers and bookmakers in NBA games. Casual social and professional gamblers are identified as the two most significant types [6]. This study has several important implications for both academic researchers and practitioners in the field of sports betting. Firstly, our research contributes to the existing literature by providing a comprehensive analysis of various machine learning algorithms and feature engineering techniques in predicting point spreads for NBA games. This work highlights the potential benefits of utilizing machine learning models over traditional linear regression models for predictive tasks in sports betting.

For practitioners, our findings underscore the importance of selecting the most appropriate machine learning model and feature set to maximize prediction accuracy and subsequent profits. By employing advanced techniques such as feature selection, dimension

[14] From June of 2018 to October of 2018 are break dates.

reduction, and feature scaling, bettors can optimize their models to better capture complex relationships in the data and ultimately enhance their betting strategies.

The outcomes of our study hold significant consequences for the knowledge distribution and the construction of efficient Management Information Systems (MIS) within the domain of sports betting, which has not been thoroughly investigated.This in turn would ultimately enhance the decision-making processes of stakeholders and improve overall sports betting market efficiency. By providing a robust and accurate ensemble model for NBA game outcome prediction, our research contributes towards efficient dissemination of information among sports bettors, analysts, coaches, and other stakeholders. As the sports betting industry continues to grow, the development of innovative MIS solutions based on our findings can play a crucial role in shaping the future landscape of this sector.

## 8. Conclusion

Our study presents important implications for the management of information and data in the context of sports betting using advanced machine learning algorithms to predict outcomes and enhance strategies. Unlike prior literature, our study employs a wide range of features and ML algorithms, resulting in higher accuracy predictions. Our models, including SVM, DTR, Stacking models, LightGBM, XGBoost, Random Forest, and DNN, utilize team-level characteristics and player-level features such as team scores, rebounds, assists, steals, and PLUS_MINUS. The simulation results indicate that our models achieved a significant profit of $150,000 with an initial investment of $100 for the 2018 NBA sports betting market.

We conclude that ML models outperform benchmark OLS regression models out-of-sample, and point spread prediction is significantly profitable compared to traditional betting strategies. Furthermore, we identify new features that are more significant and achieve better accuracy than prior literature. Our profit simulation suggests that sports betting markets can

provide significant returns at lower risk levels, which has social, economic, and behavioral implications for bettors, houses, and market efficiencies in the NBA sports betting market.

However, it is important to recognize that sports betting carries inherent risks, and individual game outcomes can be unpredictable. Therefore, while the models and techniques presented in this study may help bettors improve their predictions, they should be utilized with caution and in conjunction with other risk management strategies.

**References**

[1] Alazab, M., Awajan, A., Mesleh, A., Abraham, A., Jatana, V., and Alhyari, S. "COVID-19 prediction and detection using deep learning." International Journal of Computer Information Systems and Industrial Management Applications 12 (June 2020): 168-181.

[2] Amit, R., and Livnat, J. "Diversification and the risk-return trade-off." Academy of Management Journal 31, no. 1 (1988): 154-166.

[3] Barnes, R. "Will supreme court open a 'dam burst' of legalized sports betting?" The Washington Post, November 26, 2017. https://www.washingtonpost.com/politics/courts_law/will-supreme-court-open-a-dam-burst-of-legalized-sports-betting/2017/11/26/9f988aaa-cf9d-11e7-a1a3-0d1e45a6de3d_story.html

[4] Bun, M. J., and Harrison, T. D. "OLS and IV estimation of regression models including endogenous interaction terms." Econometric Reviews 38, no. 7 (2019): 814-827.

[5] Cai, Weihong, Ding Yu, Ziyu Wu, Xin Du, and Teng Zhou. "A hybrid ensemble learning framework for basketball outcomes prediction." Physica A: Statistical Mechanics and its Applications 528 (2019): 121461.

[6] Carver, A. B., and McCarty, J. A. "Personality and psychographics of three types of gamblers in the United States." International Gambling Studies 13, no. 3 (2013): 338-355.

[7] Cheng, Bryan, Kevin Dade, Michael Lipman, and Cody Mills. "Predicting the Betting Line in NBA Games." (2013).

[8] Cortes, C., and Vapnik, V. "Support-vector networks." Machine Learning 20, no. 3 (1995): 273-297.

[9] Davis, J. L., and Krieger, K. "Preseason bias in the NFL and NBA betting markets." Applied Economics 49, no. 12 (2017): 1204-1212.

[10] Forrest, D., and McHale, I. G. "Using statistics to detect match fixing in sport." IMA Journal of Management Mathematics 30, no. 4 (2019): 431-449.

[11] Humphreys, B. R. "Point spread shading and behavioral biases in NBA betting markets." Rivista di Diritto ed Economia dello Sport 6, no. 1 (2010): 13-26.

[12] Jain, S., and Kaur, H. "Machine learning approaches to predict basketball game outcome." In 2017 3rd International Conference on Advances in Computing, Communication & Automation (ICACCA)(Fall), pp. 1-7. IEEE, September 2017.

[13] Jones, E. S. Predicting Outcomes of NBA Basketball Games. Doctoral dissertation, North Dakota State University, 2016.

[14] Kayhan, V. O., and Watkins, A. "Predicting the point spread in professional basketball in real time: a data snapshot approach." Journal of Business Analytics 2, no. 1 (2019): 63-73.

[15] Krawczyk, B., Minku, L. L., Gama, J., Stefanowski, J., and Woźniak, M. "Ensemble learning for data stream analysis: A survey." Information Fusion 37 (2017): 132-156.

[16] Lin, Jasper, Logan Short, and Vishnu Sundaresan. "Predicting national basketball association winners." CS 229 FINAL PROJECT (2014): 1-5.

[17] Liu, B., Wang, S., Long, R., and Chou, K. C. "iRSpot-EL: identify recombination spots with an ensemble learning approach." Bioinformatics 33, no. 1 (2017): 35-41.

[18] Manner, H. "Modeling and forecasting the outcomes of NBA basketball games." Journal of Quantitative Analysis in Sports 12, no. 1 (2016): 31-41.

[19] Mendes-Moreira, J., Soares, C., Jorge, A. M., and Sousa, J. F. D. "Ensemble approaches for regression: A survey." ACM Computing Surveys (CSUR) 45, no. 1 (2012): 1-40.

[20] Milkovich, D., Gajić, L., Kovačević, A., and Konjović, Z. "The use of data mining for basketball matches outcomes prediction." In IEEE 8th International Symposium on Intelligent Systems and Informatics, pp. 309-312. IEEE, September 2010.

[21] Moskowitz, T. J. "Asset pricing and sports betting." Chicago Booth Research Paper, no. 15-26 (2015).

[22] Portugal, I., Alencar, P., and Cowan, D. "The use of machine learning algorithms in recommender systems: A systematic review." Expert Systems with Applications 97 (2018): 205-227.

[23] Probst, P., Wright, M. N., and Boulesteix, A. L. "Hyperparameters and tuning strategies for random forest." Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 9, no. 3 (2019): e1301.

[24] Rathore, S. S., and Kumar, S. "A decision tree regression-based approach for the number of software faults prediction." ACM SIGSOFT Software Engineering Notes 41, no. 1 (2016): 1-6.

[25] Rinaldo, A., Wasserman, L., and G'Sell, M. "Bootstrapping and sample splitting for high-dimensional, assumption-lean inference." The Annals of Statistics 47, no. 6 (2019): 3438-3469.

[26] Rodrıguez, J. A. " This Game Is In The Fridge: Predicting NBA Game Outcomes.

[27] Ryan, S. M., and Alameda-Basora, E. "Application of Bayesian Network to Total Points in NBA Games."

[28] Sagi, O., and Rokach, L. "Ensemble learning: A survey." Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 8, no. 4 (2018): e1249.

[29] Shi, Jian, and Kai Song. "A discrete-time and finite-state Markov chain based in-play prediction model for NBA basketball matches." Communications in Statistics-Simulation and Computation 50, no. 11 (2021): 3768-3776.

[30] Suk, H. I., Lee, S. W., Shen, D., and Alzheimer's Disease Neuroimaging Initiative. "Deep ensemble learning of sparse regression models for brain disease diagnosis." Medical Image Analysis 37 (2017): 101-113.

[31] Sun Yin, H. H., Langenheldt, K., Harlev, M., Mukkamala, R. R., and Vatrapu, R. "Regulating cryptocurrencies: a supervised machine learning approach to de-anonymizing the bitcoin blockchain." Journal of Management Information Systems 36, no. 1 (2019): 37-73.

[32] Suthaharan, S. "Support vector machine." In Machine learning models and algorithms for big data classification, pp. 207-235. Springer, Boston, MA, 2016.

[33] Szegedy, C., Toshev, A., and Erhan, D. "Deep neural networks for object detection." In Advances in Neural Information Processing Systems, pp. 2553-2561, 2013.

[34] Tanha, J., van Someren, M., and Afsarmanesh, H. "Semi-supervised self-training for decision tree classifiers." International Journal of Machine Learning and Cybernetics 8, no. 1 (2017): 355-370.

[35] Thara, D. K., PremaSudha, B. G., and Xiong, F. "Auto-detection of epileptic seizure events using deep neural network with different feature scaling techniques." Pattern Recognition Letters 128 (2019): 544-550.

[36] Thorp, E. O. "Portfolio choice and the Kelly criterion." In Stochastic Optimization Models in Finance, pp. 599-619. Academic Press, 1975.

[37] Torres, Renato Amorim. "Prediction of NBA games based on Machine Learning Methods." University of Wisconsin, Madison (2013).

[38] Vatnik, V. N. "The nature of statistical learning." Theory (1995).

[39] Valenzuela, Russell. Predicting national basketball association game outcomes using ensemble learning Techniques. California State University, Long Beach, 2018.

[40] Van Girven, M., and Bohte, S. "Artificial neural networks as models of neural information processing." Frontiers in Computational Neuroscience 11 (2017): 114.

[41] Van Osch, W., and Steinfield, C. W. "Strategic visibility in enterprise social media: Implications for network formation and boundary spanning." Journal of Management Information Systems 35, no. 2 (2018): 647-682.

[42] Zhang, Yasi, Sicheng Zhou, Xi Zheng, Yuyu Wang, and Minrui Liang. "Modeling and Predicting the Outcomes of NBA Basketball Games." In 2021 2nd European Symposium on Software Engineering, pp. 94-99. 2021.

## Appendix for Tables and Figures

Table 1. Summary of Previous Studies.

| Papers | Variable of Interest | Data Used and Number of Variables | Feature Engineering | Method | Accuracy | Limitations |
|---|---|---|---|---|---|---|
| Miljković et al. (2010) [20] | Game-Winner and Point spread | 18 team-level basketball variables and 14 team standing variables | Doesn't mention, but can see that some of features are season averaged | Classification: Decision Tree, K nearest neighbours, Naïve Bayes, and SVM. Regression: not mentioned. | 65% for classification but no results for regression. | Only team-level data and the number are limited.<br>Season averaged data are not accessible.<br>Accuracy is not high enough for profitability.<br>Test dataset is randomly split not at the end of the period. |
| Rodrıguez (2019) [26] | Game-Winner | 18 Player-level variables | Use top 3 players stats as features | Logistic Regression, 3 layers NN | 59.8% | Use one game features to predict same game's outcome.<br>No team-level stats.<br>Other players also contribute to game outcome.<br>Only two algorithms.<br>Not accurate.<br>Test dataset is randomly split not at the end of the period. |
| Jones (2016) [13] | Game-Winner | Select 144 games from 3 NBA seasons 2008-2011. 33 team-level variables. 50 games as test dataset. | No feature engineering | Linear Regression | 88-94%<br>R-squared 0.91 | Small sample size.<br>No player-level data.<br>Use one game features to predict same game's outcome.<br>Only applied linear regression.<br>Size of datasets are small.<br>Test dataset is randomly split not at the end of the period. |

| | | | | | | |
|---|---|---|---|---|---|---|
| Jain and Kaur (2017) [12] | Game-Winner | Not specified | Not specified | SVM and HFSVM | 60-65% | Not profitable against the odds.<br>Only two algorithms.<br>Low accuracy.<br>Test dataset is randomly split not at the end of the period. |
| Ryan and Alameda-Basora (2019) [27] | Game-winner | 2014-2018 NBA season data. 54 In-game team-level data from end of each quarter. | In game data at the end of each quarter. | Bayesian Network | 51.85% to 78.26% | Test datasets sizes are small, only 97 to 100 games, leading to not stable as can be seen from accuracy range.<br>In-game odds change quickly leading too hard to bet with prediction because prediction also takes time.<br>No player-level data.<br>Only one algorithm, cannot compare with others.<br>Cannot find how they get in-game data.<br>Test dataset is randomly split not at the end of the period. |
| Manner (2015) [18] | Point spread | Not mentioned | Not mentioned | Linear Regression | MAE around 9 | Cannot find variable used or feature engineering techniques.<br>Only one algorithm.<br>Test dataset is randomly split not at the end of the period. |
| Kayhan and Watkins (2019) [14] | Point spread | 2009-2016 NBA season data. In-game data. | In game data at the end of each quarter. | Snapshot approach with LSTM | MAE 11 before game starts. | Low accuracy<br>Cannot find In-game data from provided website: **http://stats.nba.com**<br>Only one algorithm.<br>Error is big. |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | | | Test dataset is randomly split not at the end of the period. |
| Torres (2013) [37] | Game-winner | 2007-2013 NBA regular seasons. 8 Past win-loss variables. | Past 8 games. | Linear regression, Maximum Likelihood Classifier, Multi-Layer Perceptron | 63.98% to 68.44% | Could have more variables.<br>No player-level data.<br>Test dataset is randomly split not at the end of the period. |
| Lin, Short, and Sundaresan (2014) [16] | Game-winner | 92-98 NBA seasons. 17 team-level variables. | Recent games. | Logistic regression, SVM, AdaBoost, Random Forest. | 60% to 65% | Data is old.<br>No player-level data.<br>Team-level variables could be more.<br>Not accurate to be profitable.<br>Test dataset is randomly split not at the end of the period. |
| Cheng et al. (2013) [7] | Spread Over/Under classification | 02-12 NBA seasons game-level and player-level data. | Not mentioned | SVM, Naïve Bayes. | 52.78% | Data is old.<br>Only two algorithms.<br>Not accurate enough to be profitable.<br>Test dataset is randomly split not at the end of the period. |
| Shi and Song (2021) [29] | Point spread | 13-17 NBA seasons in-game data. | In game data, use data in every minute to feed forward. | Markov chain | MAE of 10.8 before game starts. | Only team-level data.<br>Only one algorithm.<br>Error is relatively large.<br>Test dataset is randomly split not at the end of the period. |
| Cai et al. (2019) [5] | Game-winner | 380 games of 20 teams from 16-17 CBA regular season. | Last 5 games | SVM with Bagging, Naïve Bayes, Logistic Regression, | 80% | only use 12 variables, didn't consider player-level variables.<br>Sample size is too small, only 280 in total, so their accuracy |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | 12 team-level variables | | Neural Networks. | | can be biased. Test dataset is randomly split not at the end of the period. |
| Valenzuela (2018) [39] | Game-winner | 07-17 NBA seasons 13 team-level in-game variables. | In game data at the end of each quarter. | Logistic regression, Naïve Bayes, SVM, Neural Networks, Random Forest, Model stacking, Adaboost. | 60% to 70% of each season. | Number of variables could be more.<br>In-game odds change fast.<br>Model predictability decreases with year increases.<br>Test dataset is randomly split not at the end of the period. |
| Zhang et al. (2021) [42] | Point spread | 17-21 NBA seasons, regular team-level data with injuries and salaries of player-level data. | Not mentioned | Linear regression | 13.1 RMSE<br>65% game winner accuracy | only linear regression.<br>only limited team-level data.<br>No test dataset. |

Table 2. Summary Statistics for normalized top ten features dataset.

| **Variable** | **Obs** | **Mean** | **Std.Dev** | **Min** | **Max** |
|---|---|---|---|---|---|
| *PointSpread* | 7379 | 2.69 | 13.43 | -51.00 | 61.00 |
| *TeamEifficiencyDifferenceHome* | 7379 | 0.62 | 0.09 | 0.00 | 1.00 |
| *TeamEifficiencyDifferenceAway* | 7379 | 0.39 | 0.09 | 0.00 | 1.00 |
| *PointSpreadLastTenGamesAverageHome* | 7379 | 0.61 | 0.08 | 0.00 | 1.00 |
| *PointSpreadLastTenGamesAverageAway* | 7379 | 0.39 | 0.08 | 0.00 | 1.00 |
| *PointSpreadLastFiveGamesAverageAway* | 7379 | 0.50 | 0.14 | 0.00 | 1.00 |
| *SumPlayerAbilityLastTenGamesHome* | 7379 | 0.51 | 0.09 | 0.00 | 1.00 |
| *SumPlayerAbilityLastTenGamesAway* | 7379 | 0.41 | 0.08 | 0.00 | 1.00 |
| *FreeThrowHome* | 7379 | 0.56 | 0.08 | 0.00 | 1.00 |
| *FreeThrowAway* | 7379 | 0.61 | 0.08 | 0.00 | 1.00 |
| *DefensiveReboundAway* | 7379 | 0.54 | 0.08 | 0.00 | 1.00 |

Table 3. Summary Statistics for odds dataset.

| Variable | Obs | Mean | Std.Dev | Min | Max |
|---|---|---|---|---|---|
| *spread1* | 131690 | 3.09 | 6.43 | -22.50 | 23.00 |
| *spread2* | 131690 | -3.09 | 6.43 | -23.00 | 22.50 |
| *price1* | 131690 | -104.06 | 32.48 | -138.00 | 856.00 |
| *price2* | 131690 | -105.37 | 28.53 | -1100 | 118.00 |
| *odds1* | 131690 | 1.87 | 0.29 | 0.00 | 2.00 |
| *odds2* | 131690 | 1.88 | 0.25 | 0.00 | 2.00 |
| *pointspread* | 131690 | 2.98 | 13.35 | -58.00 | 61.00 |

Table 4. All model comparisons with prior literature features.

| Algorithms | RMSE | Max Error | MAE | $R^2$ |
|---|---|---|---|---|
| **OLS** | 13.1 | 58.31 | 10.18 | 0.17 |
| **Lasso** | 12.84 | 56.23 | 9.97 | 0.2 |
| **Ridge** | 12.86 | 55.84 | 9.98 | 0.2 |
| **SGD** | 12.92 | 56.45 | 10.02 | 0.19 |
| **DT** | 13.14 | 55.6 | 10.14 | 0.16 |
| **SVM** | 12.94 | 56.15 | 10.06 | 0.19 |
| **RF** | 12.93 | 56.55 | 10.04 | 0.19 |
| **XGBoost** | 13.01 | 54.95 | 10.11 | 0.18 |
| **LightGBM** | 12.88 | 55.91 | 9.97 | 0.19 |
| **DNN** | 12.91 | 55.95 | 9.99 | 0.19 |
| **Prior Literature** | 14.15 | 61.38 | 10.19 | 0.12 |

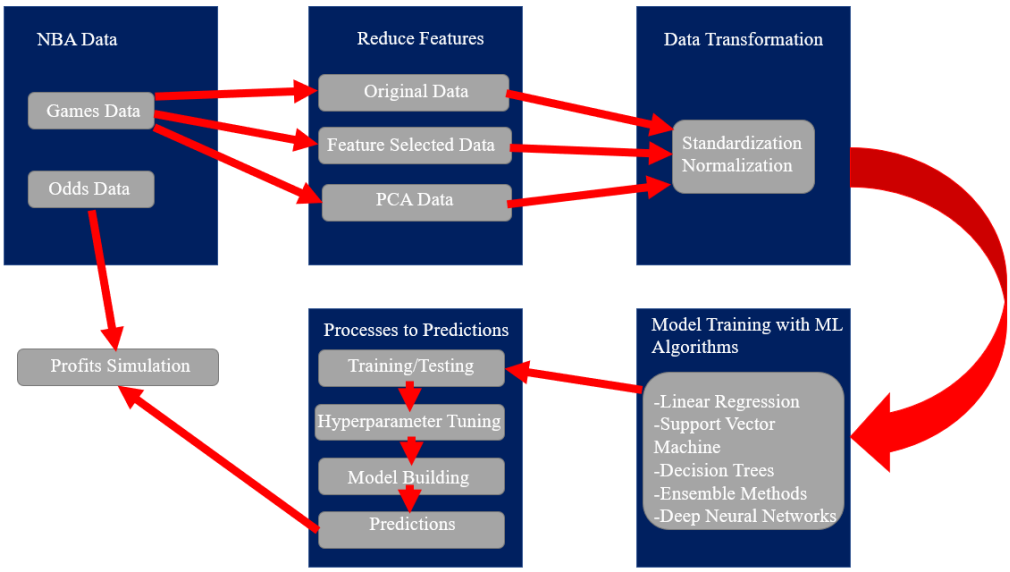



Fig. 1. NBA Games Forecasting System.

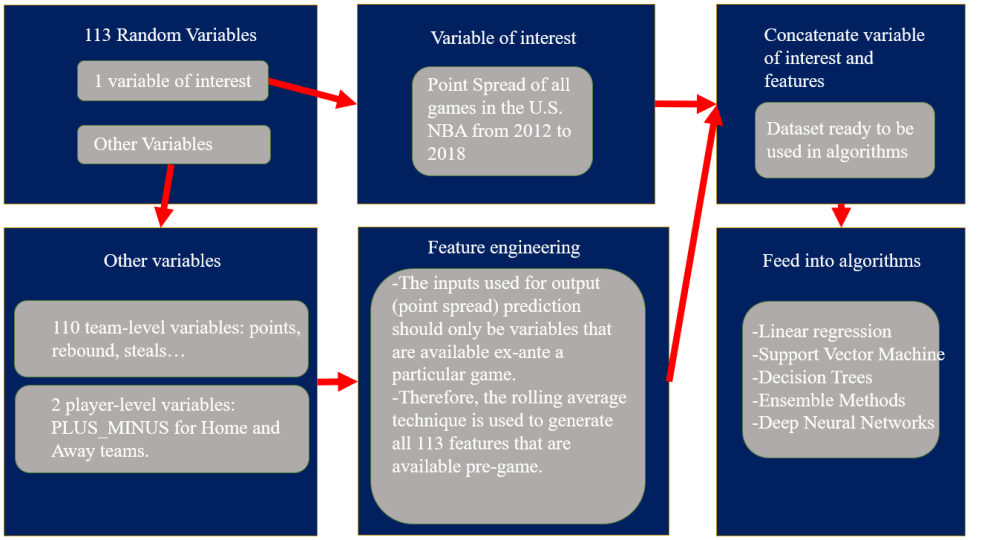



Fig. 2. Variable of Interest and Feature Engineering.

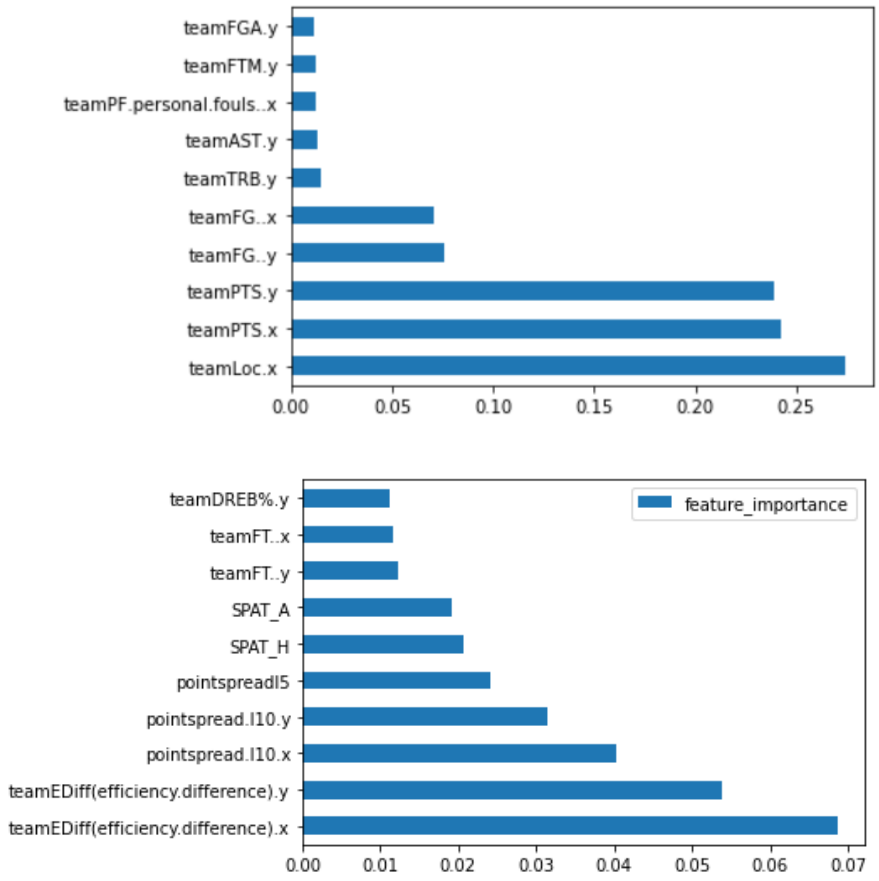



Fig. 3. Feature importance comparison.

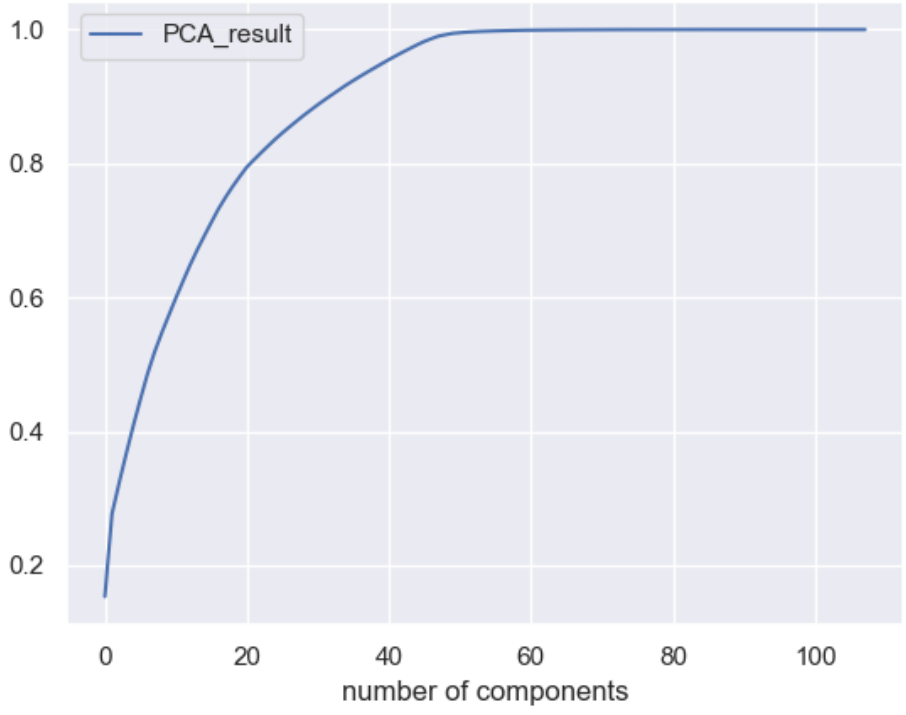
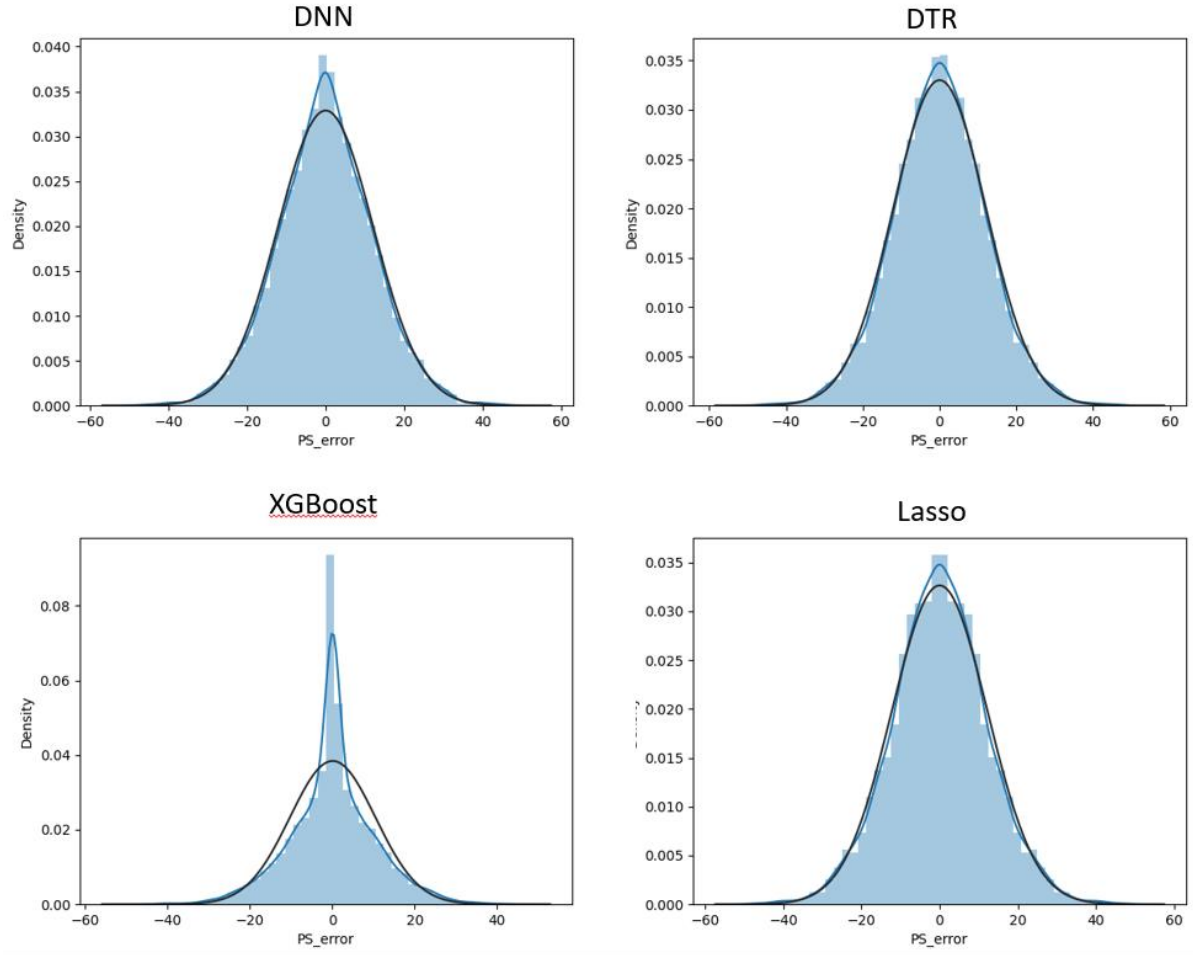


Fig. 4. PCA explained variance.



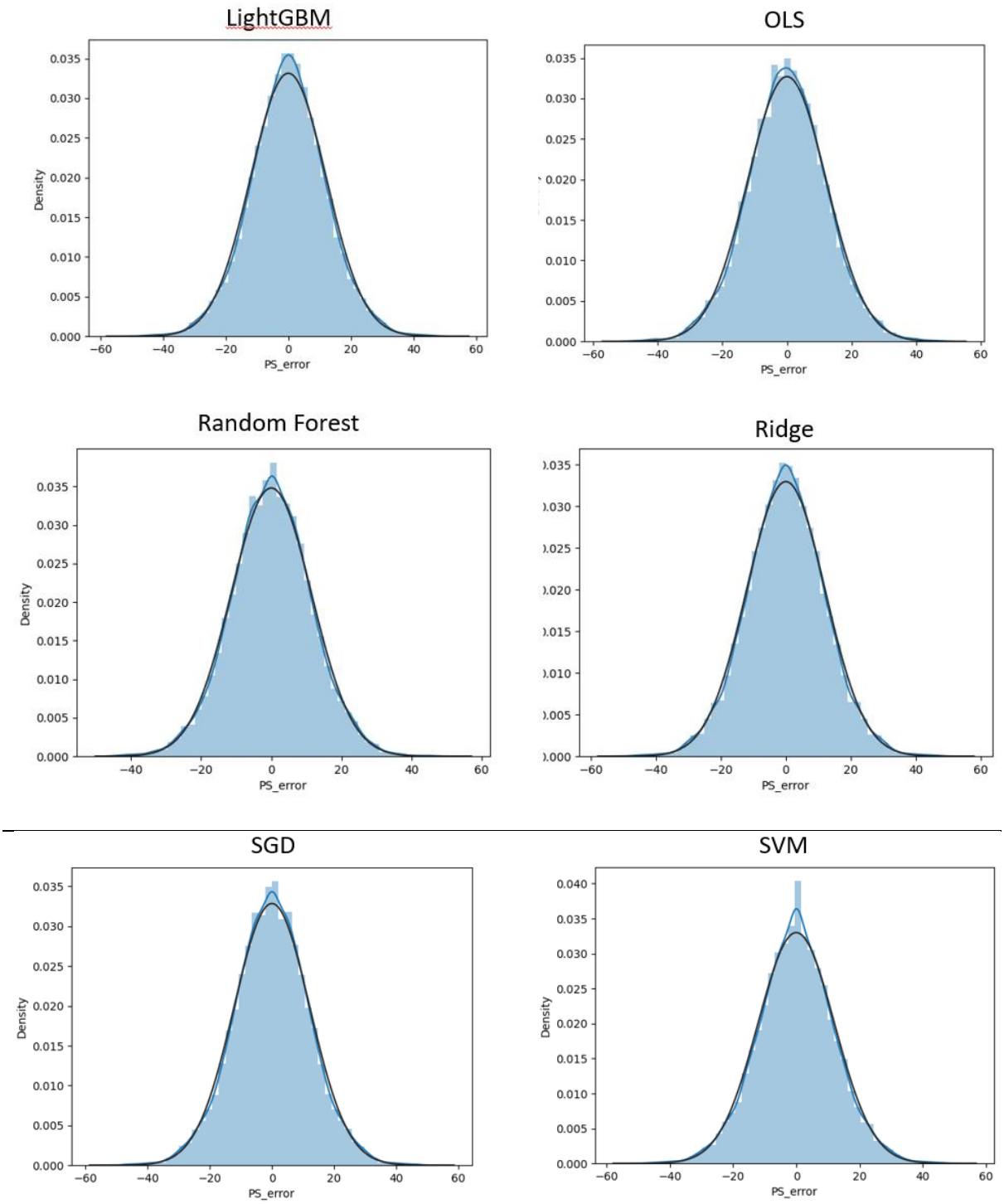



Fig. 5. Point spread comparison.

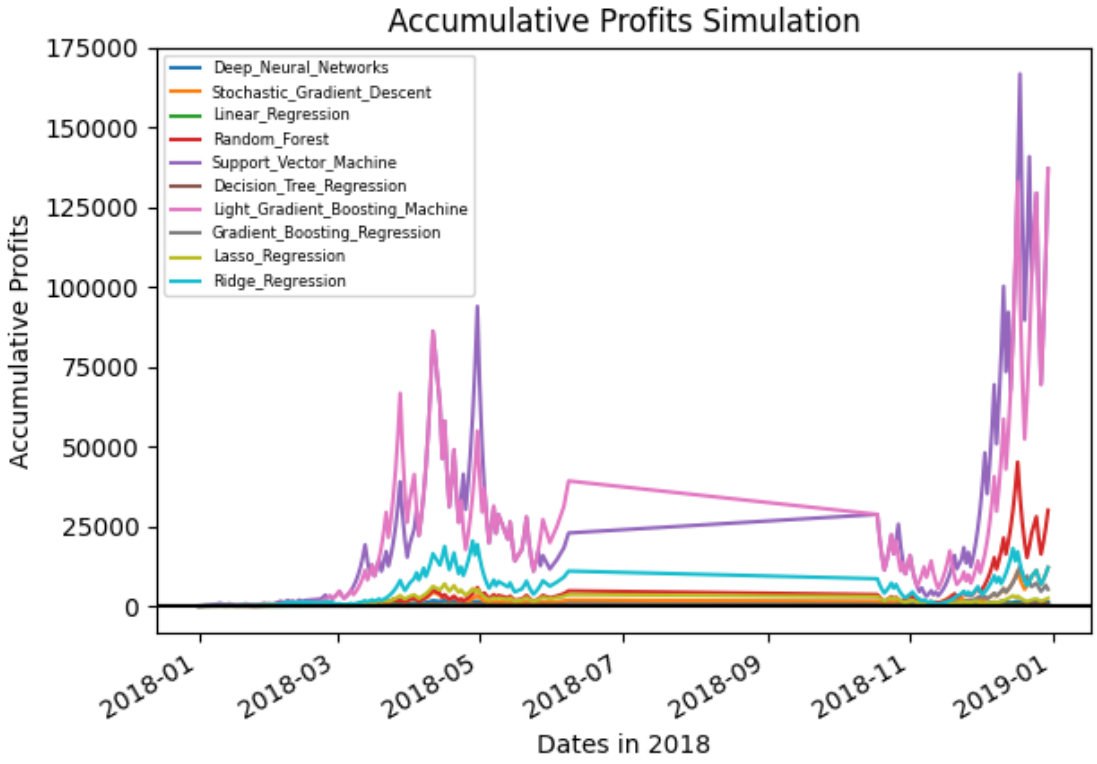



Fig. 6. Profit Simulation.