Predictive Modeling for Sports Betting

Praneeth Bhatt PSBH223@UKY.EDU
Luke Olsen LUKE.OLSEN@UKY.EDU
Nathan Turlington NHTU223@UKY.EDU
University of Kentucky, Lexington, KY, USA

Abstract

There has been a notable increase in sports gambling in recent years, in large part thanks to its rapid legalization across multiple states, coupled with technological advancements and shifting societal attitudes. This rise presents significant addiction and financial risks to users. In this paper, we introduce a neural network model that analyzes NBA games to predict future outcomes and then compare them with money-line odds to determine the best games to place bets on. We aim to empower users with informed betting strategies, thereby promoting responsible gambling practices and mitigating the potential harms associated with excessive sports wagering. We first leverage the NBA API to construct a dataset on team statistics from multiple seasons. From there, we calculate a series of advanced stats to further contextualize the box scores. After training our model with this data, we predict the outcome of the games in the test set with 65% accuracy. Our model was found to be most profitable when placing bets on a subset of games where our model and the implied odds had the highest discrepancy. There is future potential in this project to improve the accuracy by collecting more data and training further. Additionally, an explainability model, such as Shapley values, may be introduced to get a better understanding of the outputs produced.

Keywords: Sports Betting, Neural Network, Predictive Modeling, Responsible Gambling

1. Introduction

The rise of sports gambling presents a high degree of risk to a large population. This not only includes severe financial risk but also the risk of addiction. As such, there is a strong motivation to create a tool to help manage this risk. The challenges involved in this include sourcing and engineering influential features for a game in which an infinite number of factors can impact the result. Existing methods demonstrated very important steps to follow. One work proved the importance of decorrelating one's model from the bookmaker's predictions. Databall (Lane, 2022) provided the architecture to pull historical NBA game data and formulas for calculating advanced stats. Our project combined this knowledge with methods learned throughout our CS460 class to create a tool with an MLP model that would profitably predict the outcomes of NBA games.

2. Methodology

In order to identify money-line bets with a positive expected value, the team's likelihood of winning the game as a percentage is needed. Our proposed solution was to take the output of a multi-layer perceptron, where the sigmoid activation function is applied at the

last stage. The application of the sigmoid function to a real value is standard practice in binary classification problems. (Nwankpa et al., 2020)

The MLP model was trained on data from regular season NBA games, ranging from the 2019 regular season to the 2023 regular season. The stats were pulled through the publicly-available NBA API (Lane, 2022).

To improve model performance, we ran testing on different combinations of optimizers and learning rates. Further, we tested with and without regularization to account for the overfitting of the model.

The model output (winning likelihood percentage) was used in conjunction with the money-line odds, which were collected using Odds API, an online sports betting data retrieval platform. Specifically, we utilized Odds API to pull money-line odds from all games during the 2023-2024 NBA regular season. With these odds (which can be converted to percent likelihoods), we could compare our model predictions with sportsbook predictions to produce an expected value on placing a bet on any NBA game, a function of differential between our predictions and a sportsbook's.

Money-line odds represent the likelihood of a particular outcome in a sports event and indicate how much profit a bettor stands to make on a successful wager. Positive money-line odds indicate the potential profit on a \$100 bet, while negative money-line odds indicate the amount needed to wager to win \$100. To convert money-line odds to an implied win probability, the formula varies depending on whether the odds are positive or negative:

For Positive Odds:

$$Implied\ Probability = 100/(Moneyline + 100)$$

For Negative Odds:

$$Implied\ Probability = Moneyline/(Moneyline + 100)$$

By comparing the difference between the model's predicted win probability and the implied win probability, and sorting by this difference, we find the games where our model disagrees most with the moneyline. We believe that these games will generate the most profit over time.

3. Data

To create a dataset for this project, we leveraged the nba-api repository, which provides a Python package interface with which to access the NBA's official back-end API. From this, we were able to gather information about each game in a season and created a dataset of games from the 2019 regular season to the 2023-2024 regular season, which just recently concluded. In addition to the basic counting stats, we added calculations for various advanced measurements, such as offensive and defensive rating, pace, and effective field goal percentage. The formulas to calculate these were described in a project we referenced (Lane, 2022). Finally, with all of these stats from each game, we grouped the games by team, and set the values as an expanding mean, such that stats given for each game became the averages of all the games for that team prior.

In gathering money-line odds for each team for each game of the 2023-2024 NBA regular season, we provided a GET request using Odds API along with various endpoints. Specifically, we set the MARKET parameter to be 'h2h', which is representative of the money-line market. We chose to collect odds from Draftkings sportsbook for simplicity and consistency, and set the bookmaker parameter to 'draftkings' to pull their money-line odds. Each game along with each team's money-line odds for that game were collected and stored in a .csv file for later testing against our model's predictions.

4. Implementation Details

The training input features, consisting of every NBA regular season game from 2019 to 2022, was of the shape (9198, 60). The 9198 represents the number of games that were pulled by the NBA API. The reason that the number of games is not divisible by 82*15 (number of games in a season multiplied by number of teams / 2) is that the 2019-2020 Season was truncated due to Covid-19. The 60 columns are a combination of basic and advanced stats, for both the home team and the away team of each game.

The test set is the 2023-2024 NBA regular season, which has a shape of (2460, 60). Again, the row dimension is the number of games and the columns represented features. The target/label for both the training and testing sets is a single binary value, indicating if the home team won or lost.

A combination of different hyper-parameters were tested. Values of 0.25, 0.1, and 0.01 were tested for the learning rate. These are further discussed in the Results and Discussion section. The batch size used was 64. 120 Epochs were used. When a train test split was used on single season data, the ratio used was 0.3.

The model was a sequential MLP, with dropout layers included between each hidden layer. The dropout rate was set to 0.3. The number of hidden units in each layer was 128, 64, and 32, respectively. The output was a single real value produced by the sigmoid activation function that would be interpreted as a probability of a given team winning the game.

5. Results and Discussion

Training and Validation accuracies were recorded and compared using the Adam and SGD optimizers on a small range of learning rates. The comparisons were run twice with the 2023 NBA season as split into a training and validation set: one without regularization in Table 1 and one with dropout layers between hiddens layers in Table 2. The corresponding training graphs are in Figure 1 and Figure 2, respectively. Regularization was used in Table 3, where the training data is comprised of the regular seasons from 2019-2023, and the test set is the 2023 regular season. The training is shown in Table 3.

Table 1: Single Season Input Without Regularization

	Learning Rate = 0.25	**Learning Rate = 0.1**	**Learning Rate = 0.01**
Adam	(.4907, .4892)	(.5000, .5108)	(.6591, .6287)
SGD	(.6928, .6003)	(.7178, .6206)	(.7062, .6314)

Table 2: Single Season Input With Regularization

	Learning Rate = 0.25	**Learning Rate = 0.1**	**Learning Rate = 0.01**
Adam	(.5019, .4851)	(.4861, .4851)	(.6614, .6431)
SGD	(.7318, .6409)	(.7451, .6030)	(.7225, .6558)

Table 3: Multi Season Input With Regularization

	Learning Rate = 0.25	**Learning Rate = 0.1**	**Learning Rate = 0.01**
Adam	(.4989, .5000)	(.4993, .5000)	(.6070, .6098)
SGD	(.6335, .6492)	(.6351, .6520)	(.6435, .6488)

Note that Adam was not able to meaningfully train with a learning rate of 0.25 or 0.1, but the rest of the combinations of optimizer and learning rate were successful.

After adding dropout layers, the validation accuracy improved on three of the four combinations that were able to train (Adam with a rate of .01 and every model with SGD). The other two Adam learning rates could not train and are not considered in the evaluation of the regularization. Clearly, the addition of dropout layers improved the model performance.

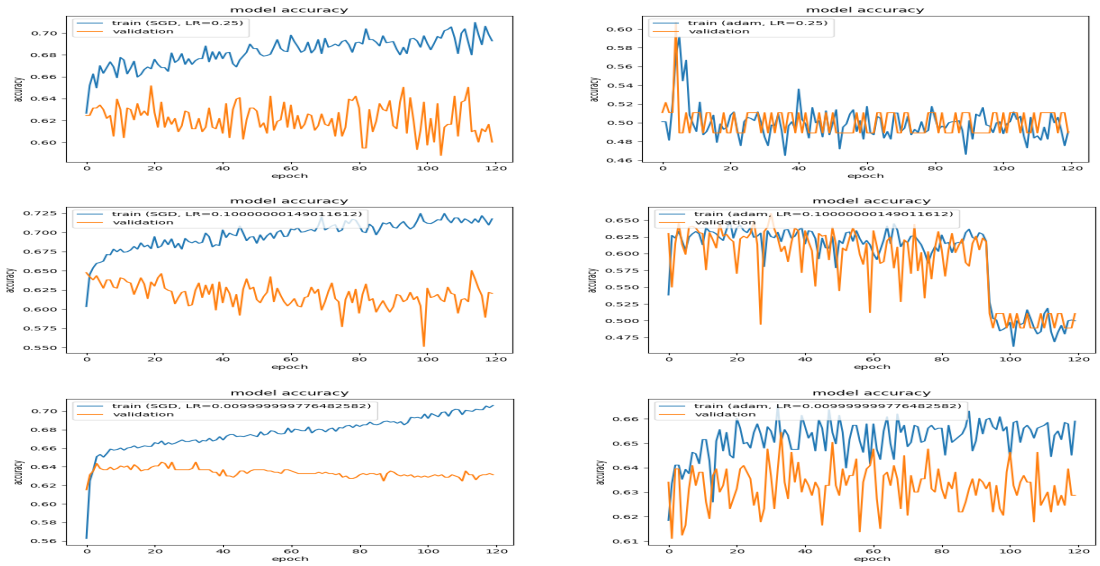

Figure 1: Single Season Training No Regularization

In Table 1 and Table 2, the highest validation rates were achieved with SGD and a learning rate of 0.01. In Table 3, the highest validation rate was achieved with the optimizer as SGD but a learning rate of 0.1. As a smaller learning rate performed better in most cases, and the performance difference in Table 3 is negligible, we decided the best optimizer to use for the final model is SGD, LR = 0.01.

The output of this final model was then compared to the implied win percentage. We then tested what our profit would have been, had we placed $100 bets on every game with a

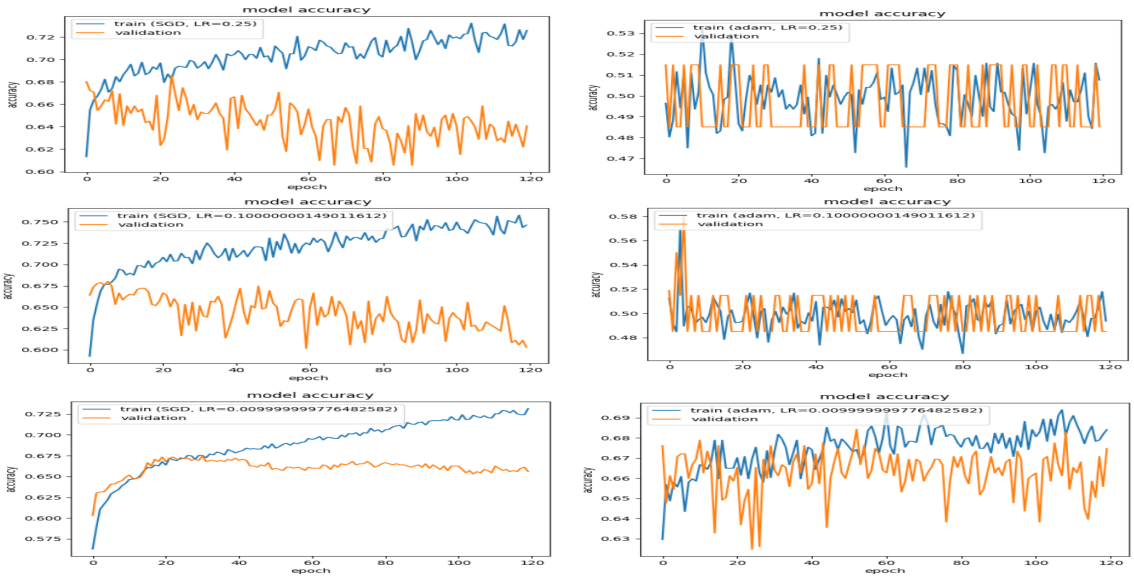

Figure 2: Single Season Training With Regularization

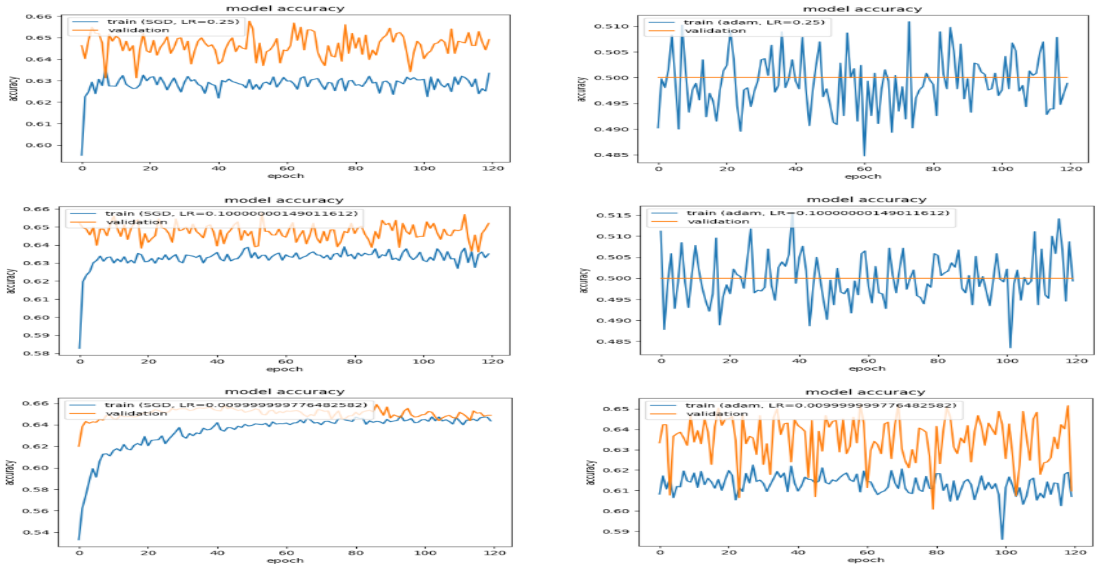

Figure 3: Multi-Season Training With Regularization

difference greater than 20%, as well as the set of the games with the 100 greatest differences. These resulted in profits of $9004 and $6105, respectively.

6. Conclusions

In conclusion, our project introduces a neural network model aimed at addressing the growing concerns surrounding the rise of sports gambling. With the increase of sports betting facilitated by legalization and technological advancements, there is a pressing need for tools

that promote responsible gambling practices and mitigate potential harms. Leveraging the NBA API and advanced statistical analyses, our model demonstrates promising results in predicting game outcomes and identifying profitable betting opportunities. By comparing our model's predictions with money-line odds, we offer users an informed approach to sports wagering, emphasizing strategies that prioritize risk management. Our model exhibits a 65% accuracy in predicting game outcomes, allowing users to more confidently predict game outcomes and thus place wiser wagers on NBA games. Further enhancements and refinements are still possible, including the incorporation of explainability models for deeper insights into model outputs. Despite challenges such as data acquisition and model optimization, our project successfully highlighted the potential of machine learning in fostering responsible gambling behaviors and illustrates the importance of ongoing research in this domain.

7. Future Direction and Impact

7.1. Future Work:

As model calibration plays a larger role in profitability than absolute accuracy, the calibration will be a focus of further developments on the model. (Walsh and Joshi, 2024)

During this project, we wanted to use a Shapley explainer (Lundberg and Lee, 2017) to understand the impact of different features on the model. Initially, we tried to generate the variable shap_values using `explainer.shap_values(features_test, nsamples=1000)`, but this caused VSCode to crash. To make the load less intensive, we attempted to run the Shapley explainer with nsamples = 100, but this progressed only to 8% after running overnight. Given the constraints on time and compute power, we were not able to locally execute this. We considered running on platforms like Google Colab, but we did not have enough computing units to execute the explainer in a reasonable amount of time. The usage of cloud computing services is cost-prohibitive for our purposes, as well.

As such, explainability with Shapley or through another vehicle is left as future work.

There are possible future additions to the model and data, as well. Our subscription to the odds API limited us to pulling the money-line odds, so more advanced options such as point spread, over/under, or betting on individual player stats were not available. A solution to this could be the development of a scraper for these betting options. This would exponentially increase the complexity of the data and model, though. Rather than maintaining data regarding each NBA teams, data would also need to be kept regarding all of the players on each team.

7.2. Impacts:

The usage of the model can help bettors make more informed decisions and minimize the loss of money. The model also illustrates to potential gamblers the thin margins provided by sports bookmakers and may dissuade them from gambling entirely.

References

Kevin Lane. DataBall. DataBall, 2022. URL https://klane.github.io/databall/.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf.

Chigozie Nwankpa, W. Ijomah, Anthony Gachagan, and Stephen Marshall. Activation functions: Comparison of trends in practice and research for deep learning. 12 2020.

Conor Walsh and Alok Joshi. Machine learning for sports betting: should model selection be based on accuracy or calibration?, 2024.

Acknowledgements

We asked general questions regarding overfitting/regularization to Dr. Qiang Ye, but he did not see any of our code or data. He did not write any of our code, either. Generative AI was not used for the development of the project.