

Machine Learning in Sports Betting, a Look at the NBA

Austin Fett
Virginia Tech
austinfett@vt.edu

Tyler Wulff
Virginia Tech
tmwulff13@vt.edu

Davin Stephens
Virginia Tech
davincs@vt.edu

Ron Yue
Virginia Tech
ron@vt.edu

The image shows a sports betting board with multiple sections. The left side lists various bets and odds, including 'SCORE A TOUCHDOWN FIRST:', 'MORE RUSHING YARDS:', 'MORE FIRST DOWNS:', 'MORE POINTS:', 'JERSEY #-PLYR SCORE 1st TD:', 'HIGHEST SCORING QUARTER:', and 'LOWEST SCORING QUARTER:'. The right side lists more bets and odds, including 'MORE: ANTETOKNMPO P/NE P', 'MORE: DeROZN P/SHORT MD FG', 'MORE: PR+BS 1QP/AMNDLA RC Y', 'MORE: BOS P/NE RUSH YARDS', 'MORE: IRVING P/NE P', 'MORE: TATUM P/NE 1H P', 'MORE: LAL+OKC P/BRDY G P Y', and 'MORE: INGRAM P/PHI 1H P'. The board is filled with numbers and team names, representing different betting options and their associated odds.

Figure 1: Odds from a physical sports book

ABSTRACT

At its core, sports betting is a data science problem. How many points does a player score? What are the chances of a team winning?

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
Conference'17, July 2017, Washington, DC, USA
© 2023 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...\$15.00
<https://doi.org/10.1145/nnnnnnnn.nnnnnnnn>

This is the focus of sports books. Sports books are not always right and professional sporting events can be extremely volatile. We explore the possibility of utilizing data and machine learning principles to gain an edge over sports books.

KEYWORDS

datasets, neural networks, sportsbooks, spread, over under

ACM Reference Format:

Austin Fett, Davin Stephens, Tyler Wulff, and Ron Yue. 2023. Machine Learning in Sports Betting, a Look at the NBA. In *Proceedings of ACM Conference (Conference'17)*. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/nnnnnnnn.nnnnnnnn>

1 INTRODUCTION

1.1 Motivation

Sports betting has become increasingly popular over the last several years. Thirty-three states now offer some sort of sports betting, whether online or in-person. We can expect more states to pass legislation and implement sports betting in the coming years. Sports betting has also become a solid revenue generator, bringing in 6.56 billion dollars of revenue in 2022 [8].

There are four types of bets that will be focused on in this project. Three of these bets will focus on teams as a whole; moneyline, spread, and over/under. A moneyline bet consists of betting on one team to beat another and your bet will win or lose solely based on the win or loss result of the game. Moneyline bets usually bring different odds than the other bets that will be looked at have. This means that if Team A is expected to beat Team B, you will win less money if you bet on team A than if you bet on Team B. Odds will be looked at more later. Spread betting, unlike moneyline betting, means that you are betting on one team to beat the other by a predetermined number of points. Sportsbooks will set lines before a game starts, indicating how much they expect a team to beat another team.

Looking at NBA lines, on February 28th, the Toronto Raptors played the Chicago Bulls, with the spread being Raptors -4.5. This means that the Raptors were expected to win by 4.5 points. You could either have bet Raptors -4.5, meaning if the Raptors won by more than 4.5 points you won, otherwise, you could have bet Bulls +4.5 and if any other result occurred, you won. The Raptors ended up winning by 6 and "covering" the spread of -4.5. Then, there are over/under bets. When taking an over/under, you are betting on how many points will be scored in the game. Looking back at the previous NBA game, the over/under was set at 219.5, meaning the two teams were expected to combine for 219.5. The score ended 104-98, meaning the under of 219.5 hit. Spread and over/unders will mostly consist of odds of -110. Although this is not always true, it is commonly the case. No matter what side you are on, you will have odds of -110. This means that in order to win 100 dollars, you need to bet 110 dollars. Avoiding a 50/50 payout is a way that sportsbooks are able to bring in such massive profits.

As stated earlier, moneylines are a little different. Once again looking back to the Raptors game, since the Raptors were already expected to win, if you wanted to just bet on Raptors moneyline, the odds were -175; a 175 dollar bet would win you 100 dollars. On the flip side, if you took Bulls moneyline, the odds were +140. If the Bulls had won, a 100 dollar bet would have won 140 dollars. The last type of bet that will be looked at is player props. These bets consist of individual player accomplishments in a game. Points, rebounds, assists, etc. can all be bet on for individual players. An over/under of each of these attributes will be set by the sportsbooks and the better has the ability to bet on whether a player will exceed or stay below the line for the individual attributes.

Player props have become an increasingly popular bet with the arrival of mobile sports betting. Bettors can bet on a player to make a certain number of points, assists, blocks, and a variety of other stat lines. This data is widely available and sportsbooks build their own models. If a bettor is able to make a more accurate model, it is possible to win money in the long run.

1.2 Problem Description

As touched on in the background, sportsbooks manipulate their odds so that there is not a complete 50/50 payout for people on both sides of a spread or over/under line. This -110 line may seem like a small difference, but that extra bit means that a better who hits 50 percent of their bets is losing money. If every single bet a person places has odds of -110, they will need to win 52.4 percent of the time to be profitable [3]. Only 3-5 percent of all betters are able to meet or exceed this mark, meaning that the vast majority of sports gamblers are not profitable in the long-run [1]. The goal of our project is to try to build a machine learning model to meet this profitability mark and join that 3-5 percent. We aren't looking for perfection in this project, a success rate of 53 percent or above will be considered a success. Due to its wide availability of data, we will be looking at the NBA to build and train our model. With 30 teams and 82+ games a year for each team, we are looking at a massive amount of data that can be used to train a machine learning model. We will be focusing on spread, over/under, and player props for our model, as we believe it is important to look at different metrics to maximize the probability of meeting our success threshold. In addition, we will experiment with non-traditional datasets like sentiment analysis. These non-traditional datasets may give insight and an edge in predicting outcomes. Not only will we be looking at different types of bets, we also will look at different sportsbooks in different countries to shop for the best odds available. Although most sportsbooks have very similar odds, there is slight variance among the lines and the odds depending on where you look. When the difference between long-term profitability and money-loss is mere percentage points, we believe it will be necessary to find any edge we can find. We also will not be "forcing" plays, meaning that we won't be betting on every game. We will need to stay very strict in regards to the what thresholds we decide to make a play on. If the model seems to think that a bet is a strong one, we must stay consistent betting at that level, switching up when we bet at the same confidence level will lead to skewed results and it will be hard to achieve a consistent win rate.

A model on this topic will not only possibly build a source of passive income for users, but it will also give more insight into how sportsbooks set their lines, and why lines are set the way they are. Although every book may be different in regards to how they set their lines, some kind of model or algorithm is most likely used to set lines as quickly as they are. Different approaches offer different insights into how approaches can be improved. Thus, our model could help sportsbooks sharpen their odds and continue to beat the public and increase profit. Lastly, our model could be used for NBA teams and players to find what aspects of the game affect the likelihood to win a game. We need to find input features that point to positive outcomes for teams, something that organizations would love to have to help improve strategy and playbooks.

1.3 Challenges and Solutions

One of the primary challenges in using machine learning as a bettor in the sports betting market is that highly developed machine learning models are already in use by sportsbooks and are used in helping set betting lines. With the existing accuracy of these models and the need to be right at least 53% of the time to make profit,

simply building a near-accurate model is not enough— although it is a necessary step[3]. Furthermore, the sportsbooks do not reveal their models as they have a business incentive not to, and in general, successful bettors do not either as this would expose and potentially undermine their means of making profit. This means that while the work has already been done within this space to build an accurate model, we do not have access to it and must replicate the process.

Another challenge is the unpredictability of sporting events themselves[4]. Even the most accurate models are limited by the quality and quantity of the data available, as well as the complex and often unpredictable nature of sporting events. While machine learning can help to improve the accuracy of predictions, there will always be a degree of uncertainty and unpredictability that must be taken into account when making bets. In terms of profit, this means that good stretches, in which more money than usual is made, and bad stretches, in which a considerable amount of money is lost, are common. Thus, a model that is deemed successful and ultimately generates a profit will still have significant periods of time in which it loses money. This can throw off confidence in the model and means that testing must occur over an extended period of time, with data to account for it. Data from 5 or 10 year stretch of games must be used to ensure that a model stands the test of time, and this costs significant resources in the form of scraping and then training the model.

In addition to these challenges, there are also ethical considerations that must be taken into account when using machine learning for sports betting. For example, there is a risk that such a model could be successful and thus exploit vulnerable individuals, such as those with poor gambling habits. This risk is not as great as it seems though, since sportsbooks rely on keeping 50/50 odds and then making money based on taking a percentage of winnings, so a winning model would not affect the chances of casual bettors.

2 RELATED WORK

2.1 Literature Survey

There have been many machine learning models built with the intention of beating the sportsbooks. One example of this comes from the paper "Exploiting sports-betting market using machine learning" by Ondrej Hubacek, Gustav Sourek, and Filip Zelezny. Similar to our project, they chose to focus on the NBA in an effort to predict games. They approached the model using artificial neural networks, allowing them to manipulate the number of features and structure easily. A goal of finding systematic bias within the odds bookmakers were offering was a failure, showing that the bookmakers were pretty unbiased. They also approached their problem from the Modern Portfolio Theory point of view, allowing them to look at expected return and variance to find an optimal betting strategy. Their strategies eventually led to profit in 15 NBA seasons, developing a model that was highly correlated to the odds that the sportsbooks offered. They found that developing a confidence level in their plays led to increased profitability, showing that being conservative is sometimes the better way to go[6].

A different paper by Sascha Wilkens called Sports prediction and betting models in the machine learning age: The case of tennis. This paper doesn't focus on the NBA, but still takes a look at tools and methods that were used to successfully beat sportsbooks in a

different sport, tennis. The focus of their model was to look at the binary relationship between winner and loser and what features led to the prediction of that outcome. Multiple models were run, including linear regression, neural networks and random forest to try to find the best method. Features were then looked at to see how "important" they were in predicting the outcome for the different matches. Something that was interesting was that it was not common for two models to have the same ranking of importance when it came to the features. Thus, each model looked at each of the features at different levels of importance. Using ensemble modeling, the model ended up predicting the outcome of a match around 70 percent of the time, which likely would not end up being profitable in the long-run. Although the model may not have been successful in its current state in the paper, it brings up the ensemble method that could be very important when building a model for this topic[7].

A third paper, by Daniel Pettersson and Robert Nyquist titled Football Match Prediction using Deep Learning, focuses on recurrent neural networks to predict the outcome of football matches. The data they used consisted of both player and team data to generate time series data to be used in the RNNs. They built their neural network using LSTM/GRU cells along with a softmax classifier. Different tuning parameters such as dropout, learning rate, batch size, and embedding dimensions were looked at in depth to try to find the most effective values for each. After training the data, their model came out to be around 46.5 percent accurate when it came to predicting the correct outcome. They came to the conclusion that there wasn't a large enough set of data for them to use, causing the model to not be as effective as hoped. They believe that a bigger network with more data about the individual players would have allowed the model to make better predictions. Although the final model may not have been the best, it still provides great insight into RNNs and the different tuning parameters that need to be used when building one of these models[4].

Cao focused on the NBA as well, with a goal of predicting the winner of games. He took a classification approach, using logistic regression and artificial neural networks to make predictions on games. Data was pulled from Basketball Reference and the NBA website, bringing in about 6 seasons worth of data points. The data consisted of player data, home and away splits, and mostly box score data of both a team and their opponents. Cao focused most of his feature engineering tasks on his and others expertise in the field of basketball. Features consisted of rest days, number of games in the last 5 days, and recent games between the opponents. He then labeled the data as either a W or L, indicating whether a team won or not. After implementing different models, Cao found that the simple logistics model, based on logistic regression, was the most successful, achieving an accuracy of about 67.82 percent. Cao brings up interesting points, specifically with how he went about his feature engineering. Expert opinions can be very useful when trying to figure out what factors are important when building a model. There was a focus only on wins and losses, so he did not have to worry about how much a team won by, but the extraction of features and simple logistic success is still something that needs to be considered when looking at how our model will be built [2].

In a Master's thesis, Guy Dotan focuses on the NBA and how to find an edge, specifically looking deep into the statistics and how

the game has changed over the years. While processing data that he pulled in from different sources, he adjusted the data to account for the change in pace of the game and the differences between three pointers now and even a few seasons ago. By looking at expected value of points depending on three point and two point percentage, he was able to find trends in data over the past seasons, not to make predictions, but to choose viable features that would be useful in a machine learning model. He started with a logistic regression model, resulting in an accuracy of 65.9 percent success rate in terms of predicting which team would win or lose. He also tried an XGBoost model that resulted in an accuracy of 65.9 percent of choosing winners and losers. He was also able to bring in almost two times return on his investment when picking and choosing which games to bet on using his models. By being more selective on what he took, he maximized his ROI. This shows that when we are looking at our final model and what we want to take, we shouldn't try to bet on every game, only ones that we feel confident in, due to either increased amount of data or the model giving us high confidence[5].

2.2 Limitations of Existing Approaches

Although there have been several models built with the intention of beating sportsbooks, very few actually meet the threshold to become profitable. Recreational players bet on their favorite home teams or use "feeling" to make wagers. Others do research and make more informed decisions. However, it is rare for a sports bettor to beat the books as bookies have teams of people whose jobs are to research and build predictive models. It seems that the biggest barrier for bettors is to find which features or statistics are actually important when it comes to the prediction of a game. Looking at correlation between features and just looking at expert opinions on the game are common ways of trying to find which features may result in a team winning or losing.

Large sportsbooks utilize machine learning as they have job postings for machine learning engineers. However, they keep all the information secret. In addition, sports bettors that successfully make machine learning models that profitably predicts outcomes are incentivized to keep their model a secret. There is no current state-of-the-art that is openly available online.

3 PROPOSED APPROACH

3.1 Problem Definition (Mathematically)

A working model is one that profitably predicts sport outcomes. It is not enough to pick which team is likely to win. Sportsbooks will give strong odds to long-shot underdogs while providing terrible odds for big favorites. Sportsbooks are effectively data science companies. NBA games are never true 50-50's; one team always has an edge and the goal of the bookmaker is to model this as closely as possible. We evaluate our model based on the win rate - a model that profitably makes decisions is considered a success. A sport like basketball is volatile and even underdog teams have a real chance at victory.

Consider this moneyline example: The Raptors are +172 and The Wizards are -225. If the odds are positive, the formula to calculate EV is:

$$EV = 1/(1 + x/100)$$

If the odds are negative, is formula is:

$$EV = 1/(1 + 100/|x|)$$

Using these examples, the Toronto Raptors will need to win at least 36.8% of the time for a moneyline bet to be profitable. The Wizards will need to win 69.2% of the time for the moneyline bet to be profitable.

Our model must pick plays that are positive in EV gains. It is difficult to measure the variance as true odds are impossible to calculate. Instead we can only rely on actual profit/loss over a large sample size to evaluate our model. Utilizing implied EV helps us make decisions on which side to bet on.

3.2 Data Pre-processing and Feature Engineering

The pre-processing done on this data involved removing any null values for any statistical categories and discounting any games that had such a value. This is important as without accurate information to feed the model, we will help the model develop bad habits. We also transformed the categorical results of spreads and over/unders into values that can be used in regression. The value we would give the spread result is 1. When the favorite covers it will be a positive value, when the underdog covers it will be a negative value, and when it's a push it will be 0. An example of this for the Atlanta Hawks, in the second game this season they were the favorites to win by 9 points, but they ended up winning by 10. For over/unders we used the same method, having values be the difference between the total score and the over/under line.

The feature engineering on the data includes transforming all of the box score values into simple season averages up until that point in the season. An example of this is for the Atlanta Hawks this season, after scoring 117 and 108 in their first two games, their average points when predicting the third game would be 112.5. This approach will allow us to teach our model to predict the outcome of games with the same limited data that it will have when predicting games that have not been played yet. The specific stats that we will be looking at include: OffPoss, Points, FG2M, FG2A, Fg2Pct, FG3M, FG3A, Fg3Pct, NonHeaveFg3Pct, FtPoints, PtsAssisted2s, PtsUnassisted2s, PtsAssisted3s, PtsUnassisted3s, Assisted2sPct, NonPutbacksAssisted2sPct, Assisted3sPct, FG3APct, ShotQualityAvg, EfgPct, TsPct, PtsPutbacks, Fg2aBlocked, FG2APctBlocked, Fg3aBlocked, FG3APctBlocked. We will be looking at these stats for both teams playing, but also looking at the these same stats that the opponents they have faced so far have average. An example of this opponent statistic for the Atlanta Hawks this season is, after giving up 107 and 98 points in the first two games, their opponents average points when predicting the third game would be 102.5.

Not only are we looking at season totals as a whole, we have transformed the dataset into rolling averages of 5 days and 10 days. This means that we take the average of the teams over the last 5 days or the last 10 days. We then train and test our models on the different datasets to see which outputs the most promising results. We are using rolling averages to account for teams getting streaky or getting cold. This also helps us account for longer term that

may occur as we are looking at how teams are playing at certain points of the year. There is a lot of variance in an NBA season, so we wanted to look at smaller segments throughout the year.

Other feature engineering we are looking into involves combining certain less-meaningful personal statistics into useful metrics for performance. An example of this includes assist-to-turnover ratio. This is a very simple stat that divides a player's/team's assists by turnovers to approximate how effectively they pass the ball. This is a misleading stat however as not all turnovers are based off of passes, some can simply be offensive fouls or lost dribbles. In order to effectively measure how efficient a player/team passes the ball, we must eliminate those "game flow" turnovers and focus on only those based off errant passes. This gives a better picture of how a team can score off passing and could be an important metric for the model.

Another aspect we are considering is sentiment analysis. We are using a Discord scraper and scraping from Discord servers that are related to NBA basketball. These words are put into JSON objects. We removed punctuation marks, URLs, and made all words lowercase. We then tokenized the words with NLTK. After that, we performed stemming and lemmatization on the tokens to get our final data set. All data are labeled by the game.

We then filtered the phrases. For example: "Go Wizards" from the Wizards discord server is not very useful for analysis, whereas "Fred Vanvleet has been really good on the road recently. LET HIM COOK" is much more useful. We are using Textblob, which is built into NLTK, for sentiment analysis. We will parse the data, tag parts of speech (pos_tag), utilize Name-Entity Recognition, and adapt TF_IDF to attempt to find a correlation between sentiment and game results.

A key part of sentiment analysis is measuring the polarity of sentiment. A phrase can be "somewhat positive" or "extremely negative." This will help build a dynamic weight model for our predictive model. We will touch more on this in our model architecture.

3.3 Model Architecture Description

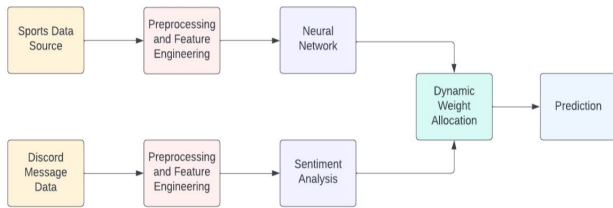


Figure 2: Pipeline Diagram

We approached the problem as seen in Figure 2. Our project involves using a neural network to predict the correct bets for NBA games. As mentioned previously in the challenges section, traditional prediction models have primarily focused on predicting game outcomes, an approach proven to be unprofitable since a profit rate of over 53% cannot be achieved[3]. Therefore, we plan to incorporate additional data, such as published betting lines and percentages, to create a model that purposefully differs from that of the bookmaker, opening up windows for profit.

We have chosen to use a neural network to accomplish our initial task of building a prediction model. Neural networks have been commonly used in past research and have showed positive signs. Furthermore, we will implement a NN using a few machine learning libraries in Python, including TensorFlow and Keras for training the model. We will also test logistic regression, random forest, and linear regression and measure how it compares to our neural network, and then determine which model has the most use for our purposes.

To maximize our chances of making a profit with this project, we will need to decorrelate our model from the sportsbook model, which involves tweaking our neural network model to predict outcomes that differ from the bookmaker's predictions. This difference in prediction is how we can maneuver around the betting lines established by sportsbooks so that occasionally, we are betting on different, and hopefully profit-generating, sides of these lines[6]. Of course, we will have to obtain and maintain a reasonable level of accuracy in predictions throughout the tweaking of our model. We plan to achieve this difference by modifying the loss function, which measures how well the neural network model fits the data, and weight it with the decorrelation term. The measurement of this decorrelation is made possible because we will have the data of odds of different sportsbooks. This part of the process will require a significant amount of trial and error, as we seek to balance the neural network's ability to make profitable decisions against its accuracy in predicting game outcomes.

In addition to our neural network, we plan to explore alternative approaches to solving this problem, one of which is sentiment analysis. We believe that Discord, as a primary source of sports betting talk, will provide a valuable dataset for sentiment analysis. By analyzing tweets and other social media content, we hope to identify the general consensus on the outcome of the games and use that information to bet on one side of the line. We will then compare the odds provided by the bookmaker to those predicted by our neural network model, seeking to identify any discrepancies that could be exploited for profit. A potential solution is to use both the general sentiment and our neural network prediction as decision factors for what games to bet on. As another approach, we will focus on areas where sportsbooks may be less accurate, such as in predicting the performance of individual players or teams in specific situations. It may be possible to identify profitable betting opportunities that are not well-covered by existing models.

The ultimate goal is to utilize sentiment analysis to assist in making betting decisions based on our neural network. We will utilize sentiment analysis data to adjust the weights for each situation and that will influence betting decisions.

4 EXPERIMENTAL EVALUATION

4.1 Dataset Exploration Showing Observed Patterns

When looking through the dataset, there aren't many clear patterns that can be observed, but there are some lessons that we can learn about certain box score stats. Obviously, teams that score a lot of points tend to be bet on to hit the over, but sportsbooks know they score a lot so they can make accurate lines. An example of this is that the Sacramento Kings are averaging 121 points per game this

season, which is the highest in the league, but they have only hit the over in 51.4% of their games. This is because the points total line is set very high. The 2023 Kings also have one of the worst defenses in the league right now and this resulted in a 351 point total game against the Clippers (Typically, point totals are around 240 for the Kings). The sportsbooks are able to adjust and set lines that are nearly equal for both sides.

A statistic that does a decent job at predicting wins and losses is whether a team is playing at home or on the road. In our dataset over the last several years, the home team wins about 55 percent of the time. This percentage has changed over the years, but it does a good job at predicting better than a flip of a coin.

One interesting statistic is offensive rebounds. Now general sports fans tend to think that the more offensive rebounds a team gets, the better they are playing however, they are not taking into account that you can only get an offensive rebound if you are missing a shot. As your offensive rebounds go up, so do your missed shots. Although it is generally a good thing to get the ball back after missing a shot, getting many offensive rebounds implies that your team is missing many shots.

As for sentiment analysis data, we see that it is usually very one-sided. We have to measure both polarity and subjectivity but ultimately, sports bettors are emotionally attached to their teams. This is not a hindrance as they are quick to be critical to the flaws of their team. Opinions are usually one-sided; how we interpret that data for our model is something we will have to answer. We will measure the polarity and intensity of sentiment to aid in dynamically weighing our neural network. This will enable us to fine tune decisions and finalize bets.

4.2 A predictive model demonstrating what has been discovered in the data

Initially, we built a logistic regression model to predict the outcome of NBA games as winners or losers. We started by doing this to try narrowing down the features that we believe are important to winning or losing a basketball game in the NBA. We used the patterns that we took notice of from the previous section to start building our models and looked at how they fared on our test data. We trained our data on the seasonal averages, 5 day average, and 10 day averages. For the season average dataset, we got an accuracy of about 56 percent for our test set. This is relatively low compared to what we got with our 10 day and 5 day averages. The 10 day averages generated an accuracy of about 65 percent, while the 5 day averages generated an accuracy of almost 70 percent. A random forest classifier was also trained and tested on our dataset, producing an accuracy of about 68 percent, just slightly worse than the logistic regression. A 70 percent accuracy for predicting wins and losses over the last 10 or so seasons is extremely encouraging.

We also used linear regression and a random forest regressor to make predictions on the data that we had with the available gambling lines that we pulled from ESPN and Kaggle. The linear regression fared terribly, with a RMSE value of almost 13. This was trained and tested on a smaller dataset because we have not been able to successfully gather gambling data from all of the past seasons yet. In regards to the random forests, we got an RMSE of

about 3, which was significantly better than what we received from the linear regression using the same dataset.

We have an initial neural network that we have trained on the box scores over the last 10 seasons. We have started with the classification task of predicting which team will win or lose, not worrying about the spread or over under yet. Our neural network had an accuracy of about 67 percent for 10 day averages and an accuracy of almost 70 percent for 5 day averages. Figure 3 shows how random forest, logistic regression, and the neural network fared against one another.

We are also building a unsupervised sentiment analysis model. We will utilize Textblob to classify phrases based on word polarity and subjectivity. Since sports talks have semantic meanings outside of discord messages, we will utilize GloVe as word embedding. We may consider using transformers using BERT depending on the success of LSTMs.

We will combine the neural network and sentiment analysis model. By making a dynamic weight allocator to make decisions on the fly based on real-time data, we are able to make better decisions.

4.3 Midterm Result Analysis

The midterm results were encouraging for each of our models, including neural neural network, random forest, and logistic regression. Our accuracy in predicting the winner of an NBA for each has increased around 15-20% for each since the initial result findings from milestone 1, with our neural network model topping out near 70%. This accuracy number is on par with models discovered in our research, and we hope to increase it further as we add in the capability to predict over/under betting odds. The increase in our models' performance stems from a greater focus on feature extraction within our box score data set. Mainly, each data point was converted from statistics of that individual game to a rolling average of what occurred in the previous 5 or 10 games and whether that resulted in a current win. This change allows us to predict a winner for a game that has not been played, which takes us in the right direction since eventually, we want to predict how to bet on a future game. Below is a plot of the current accuracy for each of our models, with training results in blue.

The neural network model using TensorFlow will be our main focus going forward. Although, our regression and random forest models are showing improvement as we continue to minimize the RMSE. Shown below in figure 4 is a confusion matrix of our win/loss predictions using our sequential model with 3 dense layers. The test set encompasses the games from about the past two seasons, 2020-21 and 2021-22, with training on previous 8 seasons before that. We also have ran initial testing on just the 2022-23 season alone, which has just been completed, and our results have been around 65%. This small dip in accuracy could mean the model is overfitting to past seasons, and we may need a way to make recent seasons more relevant.

The sentiment analysis model is accurate and has increased by 15% since the last report. We improved by implementing BERT to give encoder data. This allows us to compare similarity scores more accurately and sort by how many comments are of similar intensities.

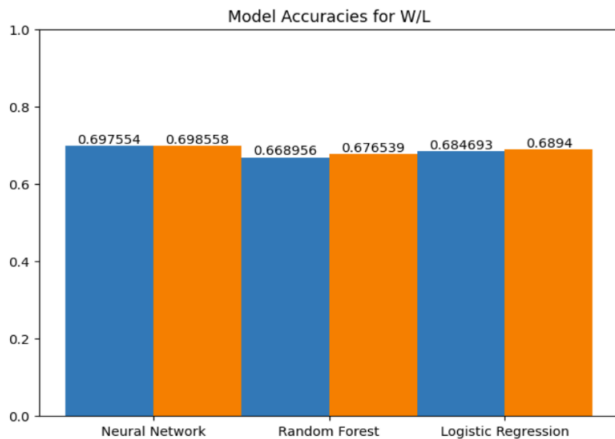


Figure 3: Model comparison (Training sets in orange, testing sets in blue)

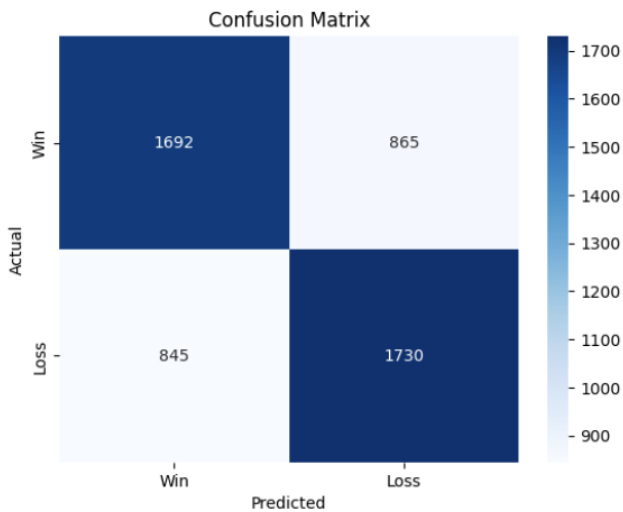


Figure 4: NN confusion matrix

5 FUTURE WORK

As we have seen significant progress in the model accuracy, we will finalize those models and alter the predictions from win/loss to also forecasting spread bets. Furthermore, we need a way to continually bring in data from the current games so that we can keep our predictions up to date. This process includes formatting the new data standard to our model, which makes predictions based off past team results. Also, our team will continue working on a regression approach where we predict the spread or over under instead, so our model can be used to choose a side that is being offered by a sportsbook. In doing so, we continue to look at different ways to extract information about spreads and over/unders from the sportsbooks so we can ensure we are getting the most accurate

lines and the best value for our plays. This is done through scraping various betting sites and adding their lines to our training sets.

We are confident that are success in predicting wins and losses will translate at least partially to spreads. Our current challenges consist of finding a source to get past spreads, since online sports betting just recently became widespread. We currently have a dataframe with about 70 percent of our data matched up with a spread, but we need that to approach 100. We will continue working on applying this dataset to our different models.

When the betting functionality is fully implemented, we can track the profit of bets over time based on our model's predictions. To reiterate our goal, we need to make a profit over 53% of the time. With the current accuracy of our neural network combined with our sentiment analysis model, it is likely that this goal can be met. The main next step for us is to link the two models together to show the confidence in our bets based on a weighting system between them. The idea is that these two models should show a strong indication that a over or under line is going to be achieved in order for us to bet on it.

Sentiment analysis is working very well; however, we still need to combine the neural network model with the sentiment analysis model. We have implemented BERT which has increased performance. Our main goal is to combine them by making a model that adjusts the weight dynamically. We are still experimenting with methods of combining the two models together.

We are working on making the website. Right now, we have not gotten started on this portion yet. Our plan is to utilize flask as the framework since our team is familiar with Python. We plan on utilizing the MVC design for the website to view statistics and betting results.

6 CONCLUSION

Our team knew prior to this endeavor in making a profit in sports betting that it would be a challenging space to enter. This is due to the fact that we are competing with already established prediction models used by betting agencies that are inaccessible to the public. Our initial findings confirmed this challenge, however, with our improvement since milestone 1 and neural network accuracy of near 70%, we believe that our goal is reachable. We have been successful in gathering a large amount of relevant data for our models to be trained on, from various websites with box score information such as ESPN and Basketball Reference. So far, we have explored a few different models that can be trained on the data from previous seasons, and we also worked on the sentiment analysis model. Mainly, we use a TensorFlow sequential model with dense layers to predict the winner, and it is trained on average team statistics over the past few games. This takes advantage of the rapidly changing environment of the NBA and allows us to predict current game winners and over/under betting lines, assuming we add the betting data to our training set. For the rest of the semester, we plan on linking our neural network and sentiment analysis model and making predictions based on the odds data, so that we can start to make bets. The sentiment analysis model will also be expanded upon so that it can be used for betting confidence alongside our neural network. Then, once the models are interlinked, we can track profit and determine if our project is successful.

7 PROJECT MANAGEMENT

7.1 Milestones Description

Our midterm goal was to have a functioning model that can predict the winner, whether to bet over/under, and player props over previous seasons. We achieved part of this goal, which is predicting the winner, however we do not have the betting prediction incorporated yet. For our next milestone, we plan to have this and also to have begun linking our neural net and sentiment analysis models. For our final goal, we hope to have a model that operates with current games and has a successful hit rate for the current season as games continue to be played. We also make weekly milestones when we meet to we stay on track. These have consisted of bringing data sets together, cleaning the data, building a basic model, and now making current predictions.

7.2 Project Timeline

We now have a few models complete with an accuracy level that can be finalized for predictions on past seasons, and we plan to transition them to making current predictions. By milestone 2 on April 12, we hope to have this done as well as adding predictions for betting over or under on current games. We will then merge together and plan to have our interlinked neural network and sentiment analysis models successful during the current NBA season by April 25th. These finished models should aim to make a profit, or at least predict the winner or score of a game with confidence, and help our team place bets.

7.3 How the team collaborates

We meet every Monday at 7:30 over zoom to update each other on the work that has been completed over the week. We also communicate with each other during the week over text, or a meeting should we feel it is necessary. The project has been split into two parts, with Ron working mostly on the NLP sentiment analysis portion of the project, while the rest of us work with the traditional box score data. Davin, Tyler, and Austin have split up the work of the traditional box score data, collaborating on what features they feel are important and in what aspects they should be used. We have come together more as the data has been scraped and pulled in, with each of us grabbing data from different sources (PBP, ESPN, and Basketball Reference). Davin scraped data from basketball reference and Austin focused on scraping data from ESPN as Tyler focused more on building the baseline regression tests that were talked about earlier. We use google colab and have been cleaning the data together and have collaborated on what we feel is the best way to build the neural network model.

7.4 Libraries and Tools

We are currently using google colab to work together to make changes to our datasets and machine learning models. We have a google drive folder containing our colab files and a document with what we feel to be relative references and sources. We also are currently using github to hold our datasets (csv files) so that we can pull them into colab with ease.

In regards to the libraries that have been used, we are using sklearn for the linear and logistic regression and are using tensorflow and keras for the neural network at the moment. BeautifulSoup was used to scrape the data from both Basketball Reference and ESPN. We also use various portions of NLTK and transformers from Hugging Face.

REFERENCES

- [1] Aaron Bruce. 2021. What Percentage of Sports Bettors Win. <https://sitpicks.com/what-percentage-of-sports-bettors-win/>.
- [2] Chenjie Cao. 2012. *Sports Data Mining Technology Used in Basketball Outcome Prediction*. Master's Dissertation. Technological University Dublin.
- [3] John V. Culver. 2018. Why 52.4 is the most important percentage in sports gambling. <https://medium.com/the-intelligent-sports-wagerer/why-52-4-is-the-most-important-percentage-in-sports-gambling-16ade8003c04>.
- [4] Robert Nyquist Daniel Pettersson. 2017. *Football Match Prediction using Deep Learning*. Master's Thesis. Chalmers University of Technology.
- [5] Guy Dotan. 2020. *Beating the Book: A Machine Learning Approach to Identifying an Edge in NBA Betting Markets*. Master's Thesis. University of California, Los Angeles.
- [6] Gustav Sourek Ondrej Hubacek. 2017. *Exploiting betting market inefficiencies with machine learning*. Master's Thesis. Czech Technical University in Prague.
- [7] Sascha Wilkens. 2021. Sports prediction and betting models in the machine learning age: The case of tennis. *Journal of Sports Analytics* 7, 2 (August 2021), 99–117.
- [8] Will Yakowicz. 2023. U.S. Set Gambling Record in 2022 With more Than 54.9 Billion In Revenue. <https://www.forbes.com/sites/willyakowicz/2023/01/13/us-set-gambling-record-in-2022-with-more-than-549-billion-in-revenue/?sh=5f9ae8a667cc>.