Machine Learning in Sports Betting, a Look at the NBA

Austin Fett
Virginia Tech
austinfett@vt.edu

Davin Stephens
Virginia Tech
davincs@vt.edu

Tyler Wulff
Virginia Tech
tmwulff13@vt.edu

Ron Yue
Virginia Tech
ron@vt.edu

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? 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, 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

1.1 Motivation

Sports betting has become increasingly popular over the last several years. Thirty-three states now offer sports betting, whether online or in-person. We can expect more states to pass legislation and allow 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]. We decided to focus on this space because sports teams rely heavily on statistics, and thus data is widely available. Also, outcomes of games for both teams and bets integrate well into machine learning problems.

There are three types of bets that will be focused on in this project. 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. 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 that 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, moneyline bets 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.

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% of the time to remain profitable [3]. Only 3-5% of all betters are able to meet or exceed this mark throughout their betting history, 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 or exceed this profitability mark and join that 3-5%. Furthermore, we aren't looking for perfection in this project, a success rate of 53% 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 large 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 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. 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. By examining different approaches to determining these lines, we can offer different insights into how approaches can be improved. Thus, our model and others 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 could find useful 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 for helping to set betting lines. With the existing accuracy of these models and the need to be correct 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 we face 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 our predictions, there will always be a degree of uncertainty 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 considerate amount of money is lost, are common. Thus, a model that we would consider 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. It also means that if this model was potentially used to make bets with real money, there is real risk of losing capital before an overall profit is made.

Another challenge with NBA betting specifically is the variability between games in terms of how teams and players perform. There is a major difference between regular season games, in which there is little stakes and the team plays a different team each game, versus playoff games, in which stakes are high and the same team is played in a seven game series. A phenomenon that is becoming more common is star players, with a high impact on the game, sitting out for regular season games. This further emphasizes our previous point– that the NBA is highly unpredictable on a small scale. While this is a challenge for accurate predictions, these changes offer an opportunity for a knowledgeable sports bettor to get an edge on the books.

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 Porfolio 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 Roberty Nyquist title Football Match Prediciton 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.

While there is much data about the statistics from NBA games, most of it is not organized in a way that it can be easily transferred to a machine learning project. This means that existing approaches may not necessarily utilize all of the different statistics possible, such as specific stats for each game played. We hope to gather as much data as possible, so that we have a good overview of what is exactly going on in each game and can make accurate predictions based on this.

Both large sports books and independent machine learning engineers are incentivized to keep their models a secret. This is so they can retain their edge and be profitable. There is no current state-of-the-art that is openly available online.

3 PROPOSED APPROACH

3.1 Problem Definition (Mathematically)

We define a working model as one that is profitable. 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

We pulled data from multiple different sources including PBP stats, Basketball Reference, ESPN, Kaggle, and Discord. PBP stats and Basketball Reference were used to pull box score data over the last 12 seasons. We found a Kaggle dataset that contained historical odds for spreads and over/under lines from 2000-2018. To gather the rest of the odds from 2018 to present, we scraped ESPN to provide us with odds from Caesar's Sportsbook to complete our dataset. We had to combine the odds data and our box score data by date and teams to end with one dataframe that we could train our models on. Discord servers were also scraped to gather the sentiment from fans for games over the previous NBA seasons.

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 averages of the games played by a specific team up to that point. These values also include the averages for the opposing team, as well as a few player stats. 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 averaged. 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.

In addition, we have transformed the dataset into rolling averages of 5 games and 10 games. This means that we take the average box score statistics from the past 5 or 10 games played by a specific team. 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 making predictions over a 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, and this helps our predictions stay relevant throughout the NBA seasons we evaluate.

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 explore 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 and we converted it into a CSV format. We removed punctuation marks, URLs, and made all words lowercase, removed @user mentions, and discord emotes. We also removed low word count messages to discard messages like "OMG" or "ha ha!". We then tokenized the words with NLTK. All data are labeled by the game based on which side the data is referring to. Celtics-Raptors-March-18 would refer to messages in the Celtics server while Raptors-Celtics-March-18 would refer to the Raptors discord server.

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 VADER, which is built into NLTK, for sentiment analysis. Originally we explored TextBlob and BERT but VADER was the most efficient in measuring sentiment via polarity. VADER uutilizes a dictionary to map lexical features. This lets the algorithm calculate emotion intensity scores.

A key part of sentiment analysis is measuring the polarity of sentiment. A phrase can be "somewhat positive" or "extremely negative." This is superior to a binary classifier as emotions are not black and white. We combine the sentiment analysis and traditional data together for our final model. We will touch more on this in our model architecture.

3.3 Model Architecture Description

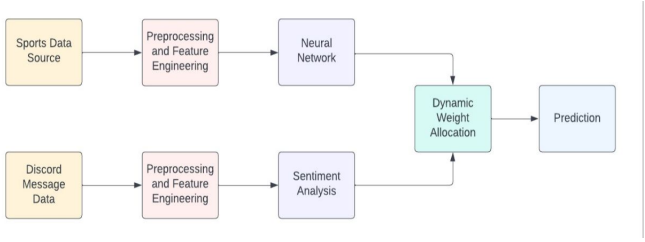

Figure 2: Pipeline Diagram

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 planned 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. Certain types of neural networks have been commonly used in the sports field, based on past research, and have showed positive signs. Furthermore, we implemented a CNN using a few machine learning libraries in Python, including TensorFlow and Keras for training the model. Our sequential model uses three dense layers with the Adam optimizer and a binary cross entropy loss function, which fits our classification problem well. We also tested logistic regression, random forest, and linear regression and measure how it compares to our neural network, and then determined which model has the most use for our purposes.

To maximize our chances of making a profit with this project, we needed to differentiate 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 had to obtain and maintain a reasonable level of accuracy in predictions throughout the tweaking of our model. This is made possible due to us training our model with various betting lines and determining which ones were most successful for our end goal of profit generation. This part of the process required a significant amount of trial and error, as we seeked 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 explored alternative approaches to solving this problem, one of which was sentiment analysis. We believe that Discord, as a primary source of sports betting talk, would be a valuable dataset for sentiment analysis. By analyzing tweets and other social media content, we hoped to identify the general consensus on the outcome of the games and use that information to bet on one side of the line. We then compared 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. Originally, we planned to build a dynamic weight model but sentiment analysis was not a reliable source of data. We instead put the polarity score into our neural network as a feature.

The ultimate goal was to utilize sentiment analysis to assist in making betting decisions based on our neural network. We utilized sentiment analysis data to react to the public's sentiment.

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 decreased 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. The sentiment analysis model is flawed because we have a false premise; discord data is skewed. The average member of a team's discord server is a fan and will produce positive sentiment in their messages. For example, the Pistons is one of the worst teams in the 2022-2023 NBA season. However, the sentiment score of this ultra low ranked team is positive. The fans are happy for any small victory and cheer for their team despite poor performance. This makes sentiment analysis difficult. In the future we could explore data that isn't from a dedicated team server. One possible source is Twitter. Twitter users do not have to join a team's server in order to tweet. This makes opinions one sided and skewed to the positive.

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 to narrow 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 then trained a neural network model on the box scores over the last 10 seasons. We 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 when looking at wins and losses.

Next, we looked at predicting whether a team would cover a spread or not. We focused our efforts on the neural network model, as that showed the most promising results when predicting wins and losses. We used the lines that were previously pulled from Kaggle and ESPN, and predicted 1 if a team covered and 0 if a team failed to cover. Our model fared extremely well, producing a training accuracy of around 62 percent, and a test accuracy of almost 60 percent. Our training data for the spreads consisted of the seasons from 2013-2022 and our test data for the spreads consisted of the entirety of the 2023 season.

We also used linear regression and a random forest regressor to make predictions on spreads. The linear regression did not do well, with a RMSE value of almost 11. In regards to the random forests, we got an RMSE of about 10, which was around the same as what we encountered with the linear regression. More work needs to be done with the data and models to try to correctly what the spread number itself should be.

Building off that, we looked at predicting whether a game would go over or under its total points. We focused our efforts on the neural network model, as that showed the most promising results when predicting wins and losses and spreads. We used the lines that were previously pulled from Kaggle and ESPN, and predicted

1 for a game to go over and 0 for a game to go under. Our model fared extremely well, producing a training accuracy of around 60 percent, and a test accuracy of almost 62 percent. Our training data for the totals consisted of the seasons from 2013-2022 and our test data for the totals consisted of the entirety of the 2023 season.

Lastly, we built an unsupervised sentiment analysis model. We originally utilized Textblob and BERT to classify phrases based on word polarity and subjectivity. However, the output was difficult to combine with the neural network. We utilized VADER from the NLTK library instead.

VADER is able to measure a message's polarity and return a sentiment score. We take the average score over the entire dataset for a game and inverse it based on sides for consistency. This gives us a decimal point number from -1 to 1, where -1 is negative, 0 is neutral, and 1 is positive.

After the completion of the two separate models (neural network and sentiment analysis model), we combined the two by adding the sentiment scores as a feature in our neural network. We came to this decision after seeing the success that our neural network had. Our hope was for the sentiment to be another attribute that could add to our accuracy and help account for players not playing, a factor that our neural network did not previously take into account.

4.3 Result Analysis

Our results were encouraging, specifically for our Neural Network model. Our accuracy in predicting the winner of an NBA for each increased around 15-20 percent after switching from whole season averages to 5 day rolling averages, with our neural network model topping out near 70%. 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. The 5 day rolling averages also allow us to take momentum into account, which can be an extremely vital part of how an NBA game plays out. Below is a plot of the current accuracy for each of our models, with training results in blue.

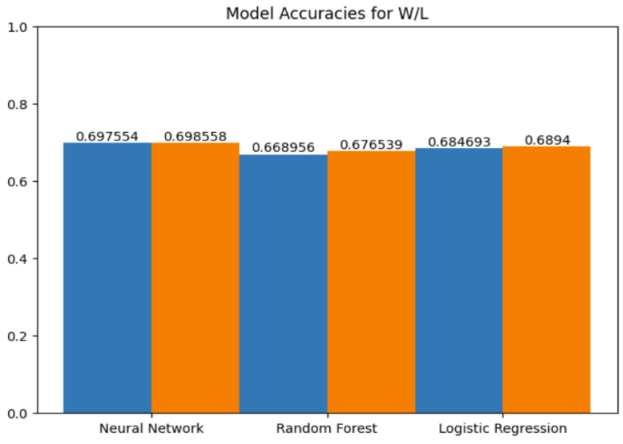

Figure 3: Model comparison (Training sets in orange, testing sets in blue)

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.

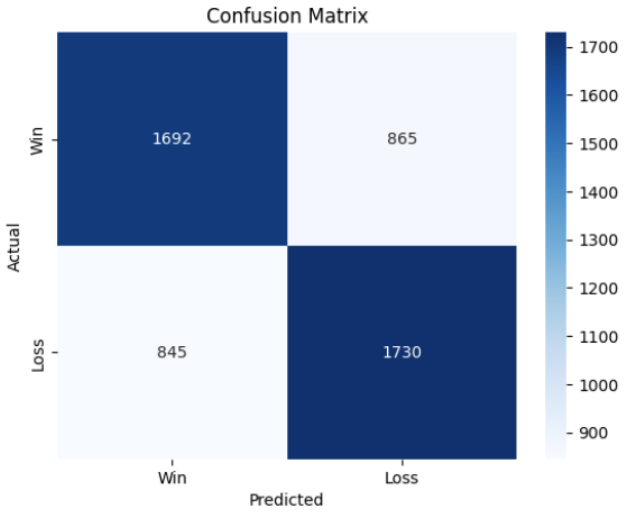

Figure 4: NN confusion matrix

We also ran tests on just the 2022-23 season alone, which resulted in an accuracy just 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.

Moving onto the prediction of spreads, the success of the Neural Network (~60% accuracy) was extremely encouraging. This percentage blows the 52.4 percent marker out of the park, meaning our model is profitable in the long run. If a bettor had placed a 10 dollar bet on every bet during the 2022-2023 regular season, they would have profited almost 4000 dollars, as can be seen in the profit graph below (Figure 5).

As you can see, there are small downturns at points, but they always rebound back up. This is why bankroll management is important and why you should never bet too much. Although the model is successful, it will still incorrectly predict games.

As for the prediction of totals, the Neural Network results were very similar to that of the spreads model (~60% accuracy). This is also well above the 52.4% threshold for profits and should be considered a massive success.

As stated beforehand, sentiment analysis is not a reliable measure of a team's success as fans in a Discord server will trend towards positive messages. This means it is wildly inaccurate for low seeded teams. Curiously enough, the model worked well for high seeded teams. We believe this is because players only express negative sentiment with injuries. However, when the model is backtested on low, middling, and high seeded teams, it produced a slight loss of 3%. This is a big loss as odds are typically pretty fair and a 10% juice along with 3% net loss means 13% total loss. The sentiment analysis succeeded specifically with big market teams that were

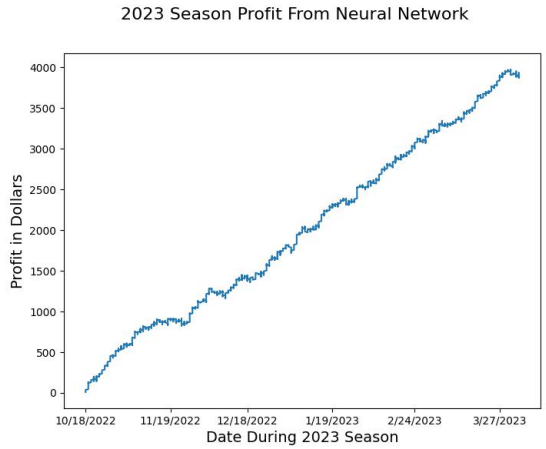

Figure 5: Profit from our Neural Network over the 2022-2023 Season

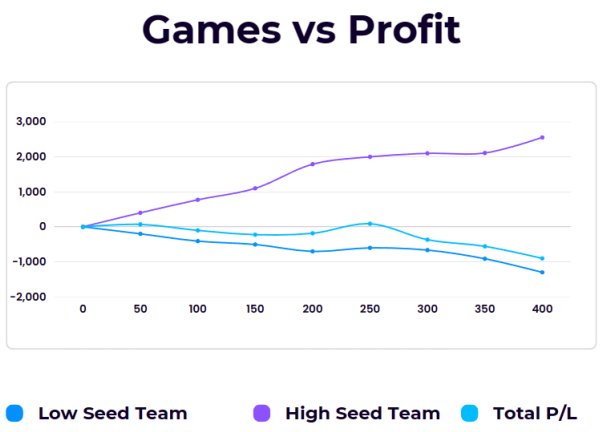

Figure 6: Profit from our Sentiment Analysis model over the 2022-2023 Season

high seeded, but struggled with smaller market teams, due to the lack of data. The model ended up losing about 22 cents per ever 10 dollars wagered over 433 previous NBA games. With more data, we believe that sentiment could be an important factor in predicting the outcome of NBA games.

Adding the sentiment as a feature in our neural network had very minimal effects on the accuracy of our model, but as stated earlier, we believe that the sentiment can be used as an important factor when looking at individual players and big market teams. More testing needs to be done with this feature to see where it can be beneficial. Thus, our final model ended with an accuracy of about 61.7 percent over the entirety of the 2023 season when predicting spreads and an accuracy of over 70 percent when predicting wins and losses, a huge success.

5 FUTURE WORK

Our final models were able to make betting predictions with an accuracy we were hoping for on the current season, making a hypothetical profit. However, before our model can be deemed trustworthy to predict future games, there is still work to be done. This work includes fine tuning, improving ease of use, and adjusting for the playoffs.

As of now, the model has no information about what is happening in a current game: it does not know the importance of the game, if there are any injuries, and how star players are performing. It relies solely on the two teams' averages for the games leading up to it. If we were able to include more data, then the model could be more accurate but more importantly, more reliable on a game-by-game basis. This way, a profit is more guaranteed in the short-term over a few games, which is more suitable to the average bettor. Another feature that could be added to convert this from a research project to a usable tool is automating predictions for games to be played in the future. We can do this manually, giving it information about the two teams playing and calculating their averages, but it is not feasible to do this for every single game in order to make a prediction. If this step was automated so that the model's predictions could be easily outputted, then it could become a useful tool for us and anyone who is inclined to use it to gain an edge on the sportsbooks.

In the opinion of our team, the NBA playoffs are much more significant than regular season games, and knowing the outcome of these playoff games would be valuable. However, our model takes into account more regular season games than playoff by a great margin. It would be interesting to focus more on the prediction of playoff games, but currently we do not have that functionality, and thus our model does not perform that well on them. This is something we could work on in the future.

6 CONCLUSION

We found great success approaching our project as a classification problem. Using rolling averages for our box score dataset increased our accuracy greatly, showing that momentum can be an important factor in the NBA. We somewhat surprisingly found that some of the most important factors in predicting a team to cover a spread or win the game are assists, defensive rebounds, and field goals given up over the last five games, not necessarily home or away as we suspected at the beginning. We found logistic regression and neural networks to perform slightly better than random forests, generating almost a 70% success rate when predicting wins and losses and a 60% success rate when looking at spreads and totals over the 2023 season. Our Neural Network was able to generate a profit of almost 4000 dollars betting on every game spread with a unit size of 10 dollars. Our model will have to continue to train on incoming data as the game continues to change season over season. We also need to continue looking at playoff games and how those differ from regular season games, as obviously playing a team at least four times in a row will likely generate different results. Overall, our results were a success, and we believe that our model provides a possible edge on NBA betting.

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 inefficiences 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.