UNIVERSITY OF CALIFORNIA
Los Angeles

Beating the Book:
A Machine Learning Approach to Identifying an Edge in NBA Betting Markets

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Science in Statistics

by

Guy Dotan

2020

ABSTRACT OF THE THESIS

Beating the Book:
A Machine Learning Approach to Identifying
an Edge in NBA Betting Markets

by

Guy Dotan
Master of Science in Statistics
University of California, Los Angeles, 2020
Professor Frederic R. Paik Schoenberg, Chair

With the recent rise of sports analytics, legalization of sports gambling, and increase in data availability to the everyday consumer, the opportunity to close the gap between the bettor and the casino appears more attainable than ever before. Our hypothesis was that one could build a model capable of exploiting the inherent inefficiencies that might exist within the betting marketplace.

Part one of this study required a derivation of the mathematics behind betting odds to determine the true probability a sportsbook places on the outcome of a matchup. Integral to this analysis was to factor in the casino's always-included cut of the betting pool (known as the "vig") that are baked into all wagers to maximize profits.

Part two was the model building process in which we trained on an archive of team-level, NBA box scores dating back to 2007 in order to predict which team in the matchup would win or lose. We aggregated our pace-adjusted box score metrics using two different methodologies, rolling eight-game spans and accumulated year-to-date statistics, and then applied these datasets to four different modeling implementations: logistic regression, random forest, XGBoost, and neural networks. Our results were optimistic, as all models were able to accurately predict the winner of a matchup at a rate of greater than 60%, thus outperforming random chance.

The final part of this research involved seeing how our best model faired versus the real-world betting lines for the 2019-20 NBA season. We used our logistic model to get a win probability for each team in every matchup and compared that result to the probability as defined by the sportsbook odds. This betting edge (the discrepancy between our model and the sportsbook) was used in a variety of betting strategies. Using our best fixed wagering technique, we were able to generate a return on investment of about 5% over the course of the entire season. Using a more complicated wagering method known as the Kelly criterion—a strategy that adjusts the amount of money wagered based on the size of the edge identified—we were able to almost double our investment with a return of about 98%. In summary, building a betting model to gain a gambling edge was determined to be quite achievable.

The thesis of Guy Dotan is approved.

Zili Liu

Vivian Lew

Frederic R. Paik Schoenberg, Committee Chair

University of California, Los Angeles

2020

Dedicated to my parents . . .

for never losing faith that their peculiar son, who,

growing up, would waste his time checking box scores,

playing fantasy sports, and reading electoral college maps,

would eventually find his way.

...also Steph Curry, for bringing basketball back to the Bay Area. Big time.

#strengthinnumbers

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

First and foremost, I'd like to acknowledge UCLA for the eight memorable years I spent on this campus. But more specifically, the UCLA Statistics faculty, staff, and community. I'll never forget the first time I stumbled into 8th floor of the Math-Sciences building as an undergrad, distraught over whether I would ever find my academic passion at UCLA. Fortunately, Glenda Jones was there to great me, support me, and guarantee that she and the statistics department had my back. That I was safe to call this place my home. And she was right.

Next I'd like to thank all of my professors in the MAS program for sacrificing three hours of their evenings, every week, to teach a bunch of tired professionals after their long day at work. Your custom-made curriculums designed directly for our interests and work-related necessities made commuting to class, not a burden, but a privilege. Thank you for your time, attention, and gracious flexibility with deadlines. Extra special thanks to Rick Schoenberg and Vivian Lew for their integral contributions to my academic success.

Lastly, I must acknowledge the companies that have filled in all my knowledge gaps and pushed me further than academics could ever do alone. What I've learned in my experiences at Tixr, STATS, GumGum Sports, and WagerWire has been invaluable and I would not be where I am personally and professionally without each and every one of them.

And of course, love and thanks to both my brothers, for their mentorship every step of the way.

CHAPTER 1

Introduction

The world of sports analytics has progressed rapidly over the past decade and with these advances, the entire professional sports landscape has evolved with it. While general advances in technology and sports organizations' willingness to pursue analytical research has facilitated the evolution, two seminal shifts within the field stand out as reasons for this surge. The first: proliferation and democratization of accessible data. The second, and more recently: the federal legalization of sports gambling within the United States.

The sports industry is one of the newest sectors to be disrupted by the emergence of data-driven decision making challenging preconceived notions from "experts," and its impact has been fast and far reaching. Setting aside the unprecedented shock to the economic ecosystem—specifically within sports and entertainment—from the 2020 outbreak of COVID-19, the sports analytics business has been thriving. "The global sports analytics market is expected to reach a revenue of $4.5 billion by 2024, growing at a CAGR [Compound Annual Growth Rate] of 43.5%." [LLP18]

1.1 Current State of Sports Analytics

To the general public, the most well-known adoption of analytics into the sports universe was within Major League Baseball, thanks largely to Michael Lewis' 2003 book *Moneyball* and subsequent movie blockbuster, starring Brad Pitt, in 2011. The story chronicles the influence of Bill James, the field of empirical baseball research known as Sabermetrics, and the story of the 2002 Oakland A's unlikely success. *Moneyball* has been the poster child of how data can create a competitive edge on the playing field. But sports analytics has

made its impression in far more avenues than just baseball. The field is responsible for the increased emphasis of the three-point shot in basketball, the use of optical player tracking technology in the NFL, and even the statistical optimization of curling game strategy that helped the Swedish women's national team to a gold medal in the 2018 Winter Olympics [Her18], just to name a few.

The integration of data analysts and scientists as crucial members of professional sports organizations appears here to stay, but this acceptance of analytics is not reserved to just a small circle of experts working for teams. The true acceleration of the movement is due to the passion from fans enabled by the increased availability of data. The NFL and NBA hold yearly hackathons to allow anyone the opportunity to dive into their sport's data and present findings to top league officials with prizes and networking at stake. Conferences such as the Sloan Sports Analytics Conference in Boston began as a small gathering of about 100 attendees in 2006, and now, in 2020, attracts over 4,000 people. The conference has gained national recognition, notably hosting former President Barack Obama as the keynote speaker in 2018. The industry's explosion in popularity though, has been aided by communities such as FiveThirtyEight, Retrosheet, Sports-Reference, and league-offered APIs bestowing data democracy to anyone that desires. Sports analytics has largely become open-source and this hive mind has benefited players, teams, and franchises.

1.2 The Legalization of Sports Gambling

On May 14, 2018, the Supreme Court case *Murphy v. National Collegiate Athletic Association* reached a landmark decision regarding the federal government's right to control a state's ability to sponsor sports betting. In a 6-3 decision, the Professional and Amateur Sports Protection Act of 1992 (PASPA) was overturned, thus opening the doors for every state to make its own laws permitting in-state sports wagering.

In just two years since the ruling there are already 18 states with full-scale legalization and another six that have passed legislation that will take effect in the coming year. [Rod20] And as one would expect, bettors in legal states have flocked to sportsbooks, both digital

and brick-and-mortar. (Sportsbooks, or "books", are places, usually part of a greater casino, where bettors can make wagers on all types of sporting events.) Since the overturning of PASPA, Americans have placed over $20 billion of bets which have generated $1.4 billion of revenue in those legal states. [Leg20] Morgan Stanley projects that in just five years, by 2025, almost three-quarters of US states (36) will have legalized sports betting and the U.S. market could see $7 to $8 billion in revenue. [AP19]

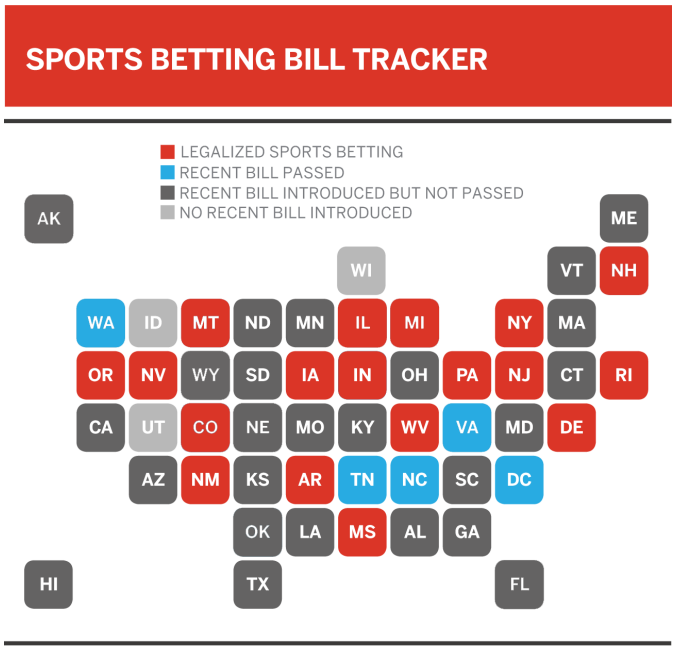

Figure 1.1: U.S. map of the current state of sports gambling legislation (as of May 2020).

1.3 The Intersection of Data and Wagering

Sportsbook operators within casinos have had decades of experience building a complex infrastructure of analytics to help them determine where to set their gambling lines. Their goal is to establish betting odds in a matchup such that there is an even amount of money wagered on both sides of the bet. This allows them to take their cut of the wagers (known in the industry as the "vigorish" or "vig") and thus drive revenue to their casino, no matter which team wins. For the entirety of their existence, sportsbooks have maintained a significant advantage over the majority of bettors. Their leverage was largely based on their access

to data and domain expertise, building models to establish the perfect betting line. It is this statistical edge that has keeps income flowing into sportsbooks. Money that helps build lavish 50-story casinos and hotels that make the Las Vegas strip world-renowned. That said, surprisingly, sports wagering makes just 3% of gaming revenue in Nevada casinos. [Boa20]

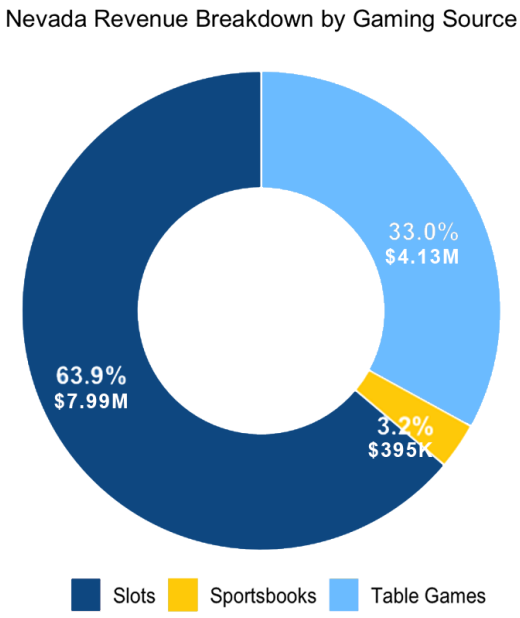

Figure 1.2: Total gaming revenue by the source in Nevada from Mar. 2019 to Feb. 2020.

But now, with sports wagering becoming more commonplace in American society and the proliferation of available sports data to everyday consumers, there is an opportunity to close the gap between casinos and bettors. Similar to how stockbrokers use proprietary projection models to systematically "beat the market," sports wagering has followed suit. It had always been a one-sided battle between the everyday bettor and the house. But in this digital age of data availability, the barrier to enter as lowered, thus leveling the playing field. The war between the book and the bettor is now an arms race where the munition is data, and for the first time in history, it might be winnable for the bettor.

Recall, a sportsbook's objective on each bet is to account for an even amount of money wagered on both sides. Oftentimes a betting line is skewed by the inherent biases of an average sports bettor. For example, if the Los Angeles Lakers (a TV market size of over five million people) were to play the San Antonio Spurs (TV market size of just 900 thousand), we

might expect a sportsbook to place the line so it slightly favors the Spurs. Even if the teams were evenly matched, sportsbooks would anticipate a disproportionate amount of hometown favorite bets supporting the Lakers. Just the smallest marginal edge, demonstrated by this example, could be enough to be exploited by an adept model. A model, when applied to a large enough dataset, could yield a considerable return on investment.

The goal of this study is to determine if applying machine learning methods to vast sports datasets (in this case, within the NBA) can create such a model that would give a bettor the competitive edge over the lines set by a sportsbook.

CHAPTER 2

The Mathematics of Sports Gambling

To understand how to beat the bookmakers, one first needs to understand how to interpret the betting lines they provide. There is a wide array of different types of bets that a person can make at a sportsbook. The three most popular betting styles are: "point spreads", "totals (over/under)", and "moneylines". This chapter will go into detail with descriptions of these types of bets using basketball as an example.

2.1 Point Spreads, Totals and Moneylines

Point spreads are mechanisms used to account for the discrepancy between two unevenly matched teams. Usually notated as: *Warriors (-5) vs. Clippers* or the inverse: *Clippers (+5) vs. Warriors.* The number in the parentheses is called the "spread." Essentially, the sportsbook believes that the Warriors are more likely to win the game, but the book wants to drive an even amount of bettors to wager on the Clippers (even though they are underdogs) as the Warriors. Therefore, the bookmakers suggest placing a spread of five points on the game. So, if you bet on the Warriors they need to win by more than 5 points to win the bet. If you bet on the Clippers, they have to lose by 4 points or fewer or win the game, for you to cash in on the wager. If the game ends with the Warriors winning by 5 points, then this is called a "push" and all bettors get their money back. Sometimes a spread will be listed at 0 points (called a "pick-em"), indicating that this is an even matchup and all you have to do is pick the winner to win the bet.

Similar to point spreads, totals (over/under) involve a specific point amount that a bettor needs to wager on the correct side of. However, in this case, the winner of the game is

irrelevant. All the bettor must do is guess if the combined score between the teams will be greater than or less than the line. For example, *Warriors vs. Clippers (o200).* If you bet the "over 200", you are expecting the combined score between the two teams to reach 201 or more and it does not matter what combination in occurs (Warriors 120-Clippers 81, Clippers 101-Warriors 100, etc.) If the final combined score is exactly 200, again this is called a push and bettors get their money back. For this reason, totals (and spreads, for that matter), oftentimes use fractional lines (+5.5 or o200.5) to prevent the case of a push. Over/under can be offered for single quarters, just the first half, just the second half, or even for a single team's score.

The third type of popular betting type is called "moneylines" and is the subject of this study. In a moneyline bet, a person simply needs to determine which team will win the game. But if one team was the heavy favorite versus the other team, it would not make sense for a sportsbook to pay out an equal amount for choosing the favorite as choosing the underdog. As a result, sportsbooks offer a moneyline, which adjusts the amount you win for having your bet hit based on the likelihood that the team will win the game. Moneylines are notated in various formats: decimal, fractional, and moneyline. The first two are commonly used in Europe. This paper will use the moneyline odds notation since they are most common to the US (and often called "American" odds). Moneylines are written as follows: *Warriors (-235) vs. Clippers*, or conversely, *Clippers (+185) vs. Warriors.*

Essentially, these either positive or negative numbers imply how much money a bettor would profit relative to a $100 bet. +185 means that if a bettor laid $100 on the Clippers, and then they won the game, the bettor would make $185 profit. -235 means that a bettor would have to wager $235 to profit $100 from that game. So if a bettor placed $100 on the Warriors at -235 moneyline odds, and the Warriors indeed won, the bettor makes $42.55 profit. Derivations for these two formulas are shown in Equation 2.1 and 2.2.

Moneyline Profit - Underdog

Let $Profit_{dog} = \Pi_d$ and $Moneyline = ML$

$$\begin{aligned} \frac{ML}{100} &= \frac{\Pi_d}{Risk} \\ 100 \times \Pi_d &= ML \times Risk \\ \Pi_d &= \frac{ML}{100} \times Risk \end{aligned} \tag{2.1}$$

Using our example...

$$\begin{aligned} \Pi_d &= \frac{+185}{100} \times 100 \\ &= 1.85 \times 100 \\ &= \$185 \end{aligned}$$

Moneyline Profit - Favorite

Let $Profit_{fav} = \Pi_f$

$$\begin{aligned} \frac{-ML}{100} &= \frac{Risk}{\Pi_f} \\ \Pi_f \times (-ML) &= Risk \times 100 \\ \Pi_f &= \frac{100}{(-ML)} \times Risk \end{aligned} \tag{2.2}$$

Using our example...

$$\begin{aligned} \Pi_f &= \frac{100}{(-1 \times -235)} \times 100 \\ &= \frac{100}{235} \times 100 \\ &= .4255 \times 100 \\ &= \$42.55 \end{aligned}$$

In summary, a \$100 bet on the Clippers (+185) leads to \$185 profit. A \$100 bet on the Warriors (-235) leads to about \$42 profit. This discrepancy in profits is to discourage enough bettors from taking the favorite Warriors and instead take the potential for upside in profit by betting on the underdog Clippers. Again, the sportsbooks goal is to optimize these moneylines to set it at a line such that an even amount of money is placed on both sides.

2.2 Implied Win Probability

The payout formula for moneylines makes it quite simple to determine how much profit a bettor can make from having their wager hit. Risk-averse bettors tend to take favorites (negative moneylines) despite the lower payouts because there is a higher chance that the team they bet on will win. Risky bettors will seek out underdogs (positive moneylines) with lower probabilities of winning, but they think will surprise the public, win the game, and thus provide a larger profit-margin.

In addition to the payout, however, moneylines can be converted into a win probability known as "implied probability". The implied probability formula is defined as the size of the bettor's wager divided by the return on investment for that wager. Or simply: risk over return. When placing a wager and consequently winning that wager, the bettor receives their initial risk amount back in addition to the profit-margin. Thus, return on investment of a wager is just risk plus return.

$$ImpliedProbability = \frac{Risk}{Return} \tag{2.3}$$

Note: $Return = Risk + Profit$

2.3 Derivation of Implied Win Probability

A universal formula to calculate the implied win probability for any bet (underdog or favorite) can be derived by plugging in the formula for moneyline profit for an underdog (Equation 2.1) and a favorite (Equation 2.2) into the implied probability formula (Equation 2.3).

Implied Probability - Underdog

$$\begin{aligned} IP_{dog} &= \frac{Risk}{Return} \\ &= \frac{Risk}{Risk + \Pi_d} \\ &= \frac{Risk}{Risk + \left(\frac{ML}{100} \times Risk\right)} \\ &= \frac{Risk}{Risk\left(1 + \frac{ML}{100}\right)} \\ &= \frac{1}{1 + \frac{ML}{100}} \\ IP_{dog} &= \frac{100}{100 + ML} \end{aligned} \tag{2.4}$$

Implied Probability - Favorite

$$\begin{aligned} IP_{fav} &= \frac{Risk}{Return} \\ &= \frac{Risk}{Risk + \Pi_f} \\ &= \frac{Risk}{Risk + \left(\frac{100}{-ML} \times Risk\right)} \\ &= \frac{Risk}{Risk\left(1 + \frac{100}{-ML}\right)} \\ &= \frac{1}{1 + \frac{100}{(-ML)}} \\ IP_{fav} &= \frac{(-ML)}{(-ML) + 100} \end{aligned} \tag{2.5}$$

Implied Probability - General Equation

$$\text{IP(ML)} = \begin{cases} \dfrac{100}{ML + 100}, & \text{if ML} \geq 0 \\ \\ \dfrac{(-ML)}{(-ML) + 100}, & \text{if ML} < 0 \end{cases} \tag{2.6}$$

2.4 The Vig or the Juice

As mentioned previously, the goal of a sportsbook is to have an equal amount of money placed on both sides of a wager so that no matter the results, they will make money once they take their cut of the bets. This cut is known as the vigorish, and more colloquially, the "vig" or the "juice." So how does one calculate the juice? Let's use our example from above with the Warriors versus the Clippers.

The Warriors moneyline odds were -235, which after using Formula 2.6, comes out to an implied probability of 70.15%. The Clippers moneyline odds were +185 and therefore an implied probability of 35.09%. Now, the most basic rule of probability states that the sum of all possible outcomes of an event always equals 1. And more specifically, the probability of an event plus the probability of the complement of that event equals 1. In basketball there are only two valid outcomes of a game: win or loss. If we consider the chances that a team wins a game as the probability while the chances they lose is the complement of that probability, we would expect these two events to sum to 1. See below:

Given:

$P(A) + P(A^c) = 1$

Let ...

$P(A)$ = Probability that Warriors win

$P(A^c)$ = Probability that Warriors lose (i.e., Clippers win)

$P(A) = 0.70$

$P(A^c) = 0.35$

$P(A) + P(A^c) =$

$$0.70 + 0.35 = 1.05$$

$$1.05 \neq 1$$

So these two implied probabilities are mutually exclusive, compose the entire space of outcomes, and yet sum to over 100%. This summed probability (in this case 105%) that is

greater than 1 is called the "overround" and is how sportsbooks take their cut. By setting the betting lines such that the probabilities result in an overround, the sportsbook effectively ensures that they will gain a profit from this wager. In our example above, a \$100 bet on the Warriors pays out \$42, while a \$100 bet on the Clippers pays out \$185. One can determine how much a sportsbook would expect to pay out from these implied probabilities through the following calculation:

Expected Payout:

Let $E(\Pi)$ = Expected Payout for a Sportsbook

$$E(\Pi_i) = \Pi_i \times P(win_i)$$

$$E(\Pi) = \sum_i \Big(E(\Pi_i)\Big)$$

$$E(\Pi) = \sum_i \Big(\Pi_i \times P(win_i)\Big)$$

$$E(\Pi) = \Pi_A \times P(win_A) + \Pi_{A^C} \times P(win_{A^C})$$

Using our example...

$E(\Pi_A)$ = Expected Payout if Warriors Win

$$E(\Pi_A) = \Pi_A \times P(A)$$

$$E(\Pi_A) = \sim\$42 \times 0.7 = \sim\$30$$

$E(\Pi_{A^C})$ = Expected Payout if Clippers Win

$$E(\Pi_{A^C}) = \Pi_{A^C} \times P(A^C)$$

$$E(\Pi_{A^C}) = \$185 \times 0.35 = \sim\$65$$

$$E(\Pi) = \$30 + \$65$$

$$= \sim\$95$$

As seen above, this overround (105%) of the probabilities creates the vig for the casino. If the sportsbook were to take \$100 of total wagers on this bet, on average, they would expect to pay out just \$95. Thus, ensuring a cut of about \$5 for this wager. Now consider the fact that casinos collect millions of dollars on bets, not \$100, it is easy to see why sports gambling is such a lucrative endeavor for bookkeepers, especially when the optimal moneylines are established (and therefore the juice is optimized as well).

2.5 "Removing the Juice" - Actual Win Probability

To get a clear idea of the sportsbooks' expectations of how likely each of the two teams is to win a matchup, we need to get rid of the guaranteed profit they bake into the lines. The implied probabilities are derived from the betting lines, but the actual is what remains after extracting the vig. Probability theory claims that the sum of all possible events in a sample space should always equate to 1. So to get the true probabilities based on the betting odds, the implied probabilities need to be scaled so that they also sum to 1. The method to remove the vig is simple, just divide the probability by the overround. [Dim20]

$$ActualProbability = \frac{ImpliedProbability}{Overround}$$

$$\text{The actual probability the warriors win} = \frac{70\%}{105\%} = 67\%$$

$$\text{The actual probability the clippers win} = \frac{35\%}{105\%} = 33\%$$

We see that the two probabilities now sum up to 100% and therefore represent the true probability that the sportsbook places on each team's chances of winning. In summary:

Table 2.1: Summary of betting values for the sample wager.

Team	Moneyline Odds	Risk	Profit	Return	Implied Win Probability	Actual Win Probability
Warriors	-235	$100	$42	$142	70%	67%
Clippers	+185	$100	$185	$285	35%	33%

CHAPTER 3

Data Collection

The gambling mathematics discussed in Chapter 2 can be applied to any sport with betting lines. But for this study, we focused specifically on professional basketball betting data. The NBA is a particularly useful test subject for this research for a few reasons. First off, the general popularity of the sport of basketball is on the rise. A 2017 Gallup poll found that basketball has surpassed baseball as America's second-most favorite sport to watch. Although, football is still heavily the favorite: at 37%, versus 11% for basketball. [Gal17]

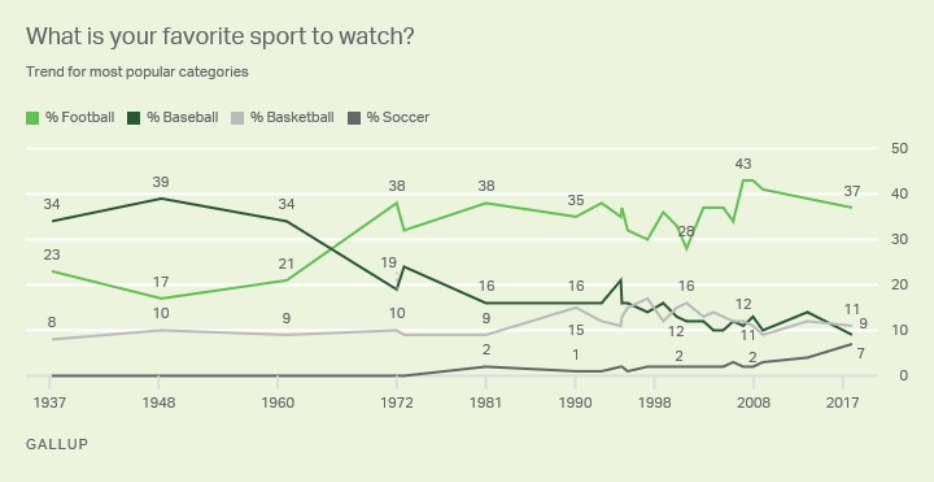

Figure 3.1: 2017 survey from Gallup on Americans and their favorite sport to watch. [Gal17]

But for our purposes, the advantage the NBA has over the NFL, is its sample size. The NBA consists of 30 teams and an 82-game season, as opposed to the NFL which has 32 teams, but teams play just 16 games in the regular season. Additionally, basketball games are high scoring and high possession competitions, hence, within a single basketball game, there are a lot of statistics that accumulate. Much more than a baseball, football, or hockey game. Simply put: lots of games and lots of stats within those games makes for large sample

sizes and great datasets.

With the rise in popularity of the sport so too has the interest in its numbers. Over the last decade, the NBA has experienced arguably the largest increase in the emphasis put on data analytics of the four major sports (NHL — hockey, MLB — baseball, and NFL — football). Because fandom is increasing, and this young, more technologically-savvy fan base is interested in the subject matter, the teams and the league recognize the value of analytics. As teams and the league office hire larger analytics departments, they, in turn, provide data for the fans to dig into. The community takes this data and shares insights on forums, blogs, and repositories which spawn innovative tech companies in the field and, consequently, these new ideas make their way back to NBA. This feedback loop powered by data availability keeps research and excitement high around basketball statistics.

The number of basketball fans continues to grow and likewise the betting markets grow as well. But amongst these waves of new bettors entering the markets there is a subset that aren't just your regular sports fan. These are knowledgeable NBA geeks that have a proclivity for numbers and the tools for adept decision making.

3.1 Betting Lines

The first step to building the dataset required for this study was acquiring betting data for as many NBA games as possible. More specifically, we are looking for moneylines odds since that is what is necessary to convert betting lines into win probabilities. Fortunately, an archive of betting odds for the NFL, NBA, NCAA football, NCAA basketball, MLB, and NHL were all available on one online resource.[1] The relevant data for this study went back to the 2007-08 NBA season and was complete up to the current 2019-20 season. Each file was formatted into one of 13 downloadable Excel files that contained both regular and postseason odds. Each season's data was clean and uniform, which seamlessly merged into one large dataset of 32,952 rows (16,476 games since each game had two rows to represent each team's

[1] https://sportsbookreviewsonline.com/scoresoddsarchives/scoresoddsarchives.htm

odds). There was only one missing moneyline in the dataset (which we ignored), spanning from the first game of the 2007-08 season until the NBA season was abruptly cut short on March 11, 2020. This was a historic day, not only in basketball, but professional sports history. Before tipoff, Rudy Gobert, a center for the Utah Jazz tested positive for COVID-19, prompting commissioner Adam Silver to cancel Utah's game versus the Oklahoma City Thunder and then the rest of the season indefinitely. [ESP20] Recent breaking news is that the NBA will continue it's season in a closed-campus environment at Orlando's *Disneyworld* starting late July.

With a complete set of betting data, the final processing step for the dataset was to convert those odds into win probabilities. Using the derived probability formulas from Chapter 2, R functions were written to implement these formulas and were systematically applied to the entire betting dataset.

3.2 NBA Game Statistics

There were several different approaches for statistics that we could use to build a win probability model to pair alongside our archive of betting odds. The most straightforward approach, and the one used for this research, was to utilize game-by-game, team-level box scores. Game-level detail allowed for the flexibility to aggregate the data in a variety of ways that would be useful for modeling purposes.

Conveniently, the R package *nbastatR* provides a robust interface that easily pulls data from an array of online basketball data resources such as NBA.com/Stats, Basketball Insiders, Basketball-Reference, HoopsHype, and RealGM. [Bre20] One of the functions in this package loads game-logs for each team over any desired seasons. All that was necessary was to input the same 2007-2020 season span that we already had in our betting archive. This gave us the same 16 thousand or so games and over two dozen raw team statistical variables to use in our model. A data dictionary of these variables is shown in Table 3.1.

Joining the betting odds data with the game-log box scores was not a trivial task. Because these datasets did not come from the same source there was no linking key between each box

score, each betting line, and each team. However, through rigorous data cleaning—which included the manual creation of standardized team IDs and game IDs—we were able to combine the two datasets. The merging of the two datasets was a perfect match with team statistics available for every single matchup in which betting lines were available, save two instances. One, the canceled 2020 games due to COVID-19 and two, a March 15, 2013 game between the Boston Celtics and Indiana Pacers which was canceled, and never rescheduled, as a result of the tragic bombing at the Boston Marathon the day before. [Gol13]

Table 3.1: Data dictionary of variables from betting archive and NBA box scores.

Category	Variable	Description
Matchup	`idGame`	(num) Unique game ID
	`slugSeason`	(num) NBA season
	`gameDate`	(date) Date of the game
	`beforeASB`	(T/F) Game is before All-Star Break (T) or after (F)
	`locationGame`	(char) Home or away
	`slugMatchup`	(char) Game matchup
	`slugTeam`	(char) Team abbreviation
	`slugOpponent`	(char) Opponent team abbreviation
	`numberGameTeamSeason`	(num) Team's Xth game of season
	`isB2B`	(T/F) Is the team on a back-to-back (i.e. 0 days rest)
	`isB2BFirst`	(T/F) First game of a back-to-back
	`isB2BSecond`	(T/F) Second game of a back-to-back
	`countDaysRestTeam`	(num) Number of days since the team's last game
	`countDaysNextGameTeam`	(num) Number of days until the team's next game
Outcome	`slugTeamWinner`	(char) Winning team abbreviation
	`slugTeamLoser`	(char) Losing team abbreviation
	`outcomeGame`	(char) Team win or loss
	`isWin`	(T/F) Team win or loss
Betting Odds	`teamML`	(num) Moneyline odds for the team
	`oppML`	(num) Moneyline odds for the opponent
	`teamWinProb`	(num) Win probability for the team
	`oppWinProb`	(num) Win probability for the opponent

Traditional Stats	fgmTeam	(num) Field goals made by the team
	fgaTeam	(num) Field goals attempted by the team
	pctFGTeam	(num) Field goal percentage from team
	fg3mTeam	(num) 3PT field goals made by the team
	fg3aTeam	(num) 3PT field goals attempted by the team
	pctFG3Team	(num) 3PT field goal percentage by the team
	pctFTTeam	(num) Free throw percentage by the team
	fg2mTeam	(num) 2PT field goals made by the team
	fg2aTeam	(num) 2PT field goals attempted by the team
	pctFG2Team	(num) 2PT field goal percentage by the team
	minutesTeam	(num) Minutes played by the team
	ftmTeam	(num) Free throws made by the team
	ftaTeam	(num) Free throws attempted by the team
	orebTeam	(num) Offensive rebounds by the team
	drebTeam	(num) Defensive rebounds by the team
	trebTeam	(num) Total rebounds by the team
	astTeam	(num) Total assists by the team
	stlTeam	(num) Total steals by the team
	blkTeam	(num) Total blocks by the team
	tovTeam	(num) Total turnovers by the team
	pfTeam	(num) Personal fouls committed by the team
	ptsTeam	(num) Total points scored by the team
	plusminusTeam	(num) Team score differential (Own - Opponent points)
Advanced Stats	possessions	(num) True team possessions in a game
	pace	(num) Number of possessions per 48 minutes by a team.

3.3 Advanced Metrics

For the majority of basketball history, analytics were driven by these standard box score metrics currently in our dataset. More recently, however, the industry has been shifting away from these raw values toward a more dynamic approach that can take into account the shifts in game tempo season-to-season and even game-to-game. The solution: looking at the

normal metrics—points, rebounds, assists, etc.—on a per-possession basis, as opposed to per game. "By looking at the game at a per-possession level, it eliminated pace and style of play differences from the equation and put all teams on a level playing field. This way a team that is constantly running and has more possessions each game doesn't have a statistical advantage compared to a team that plays at a slower speed." [Mar18]

Possessions used to be estimated through a rudimentary formula that took into account how many field goals, free throws, turnovers, and rebounds a team got over the course of a game. But with the proliferation of play-by-play data, the NBA now has an exhaustive account of true possessions for each team and each game dating back to the 1996-97 season. The NBA provides advanced data (of which possessions is included) on a per game and per team basis through an open API.[2] Using Python's *requests* package, we were able to tap into the API and then make the call to the appropriate endpoints to return a data dump of advanced metrics for each game. The data came in a JSON format, so all that was left to do was to use Python's *pandas* library to format the JSON response into a data frame so it is easier to work with. Since this data was pulled directly from the NBA's official statistics database, the game IDs and team IDs directly matched those that came from the traditional box score metrics from the *nbastatR* package and the two were successfully joined together.

Our dataset was now complete with betting data (and corresponding win probabilities), matchup data (outcome, home/away, days rest, etc.), traditional box score metrics, and per-game possessions.

[2]Application Programming Interface

CHAPTER 4

Data Processing and Exploratory Analysis

The raw box score statistics provided the backbone for the features that will need to be included in the win probability model. With those box scores, we then applied two types of transformations to our dataset. The first, as mentioned previously, was to scale the raw metrics to account for the variability in game-to-game possessions (the rationale for scaling by possessions is detailed in Sections 4.1 and 4.2). The second, was to aggregate the data to account for the accumulation of statistics each team had built up entering the given matchup.

4.1 Seasonal Trends

People who have been fans of the NBA over the past couple of decades can attest to how different the game is played in 2020 than before. Whether it is due to the rise in analytics, improvements in training and medical advances, or gameplay strategy implemented by coaches, the fact remains: what it takes to win a basketball game has changed. One of the biggest shifts in game style recently has been the tendency to shoot more three pointers. The secret to unlocking this strategy is no real mystery. NBA teams on average shoot about 35% on three pointers, 45% on two pointers, and 75% on free throws. The math breaks down quite simply as follows:

Expected Points on a Certain Shot Attempt:

Expected Points = (Points) $\times$ (Percent changes of making the shot)

$E(3PA) = 3 \times 0.35 = 1.05$

$E(2PA) = 2 \times 0.45 = 0.90$

$E(FTA) = 1 \times 0.75 = 0.75$

There are more subtleties and complexities to these style changes, but the redefining of the ideal shot selection distribution has been a major component. With these types of optimizations occurring on the court, we see it manifesting in season-wide trends on the box score metrics. Figure 4.1 demonstrates these trends.

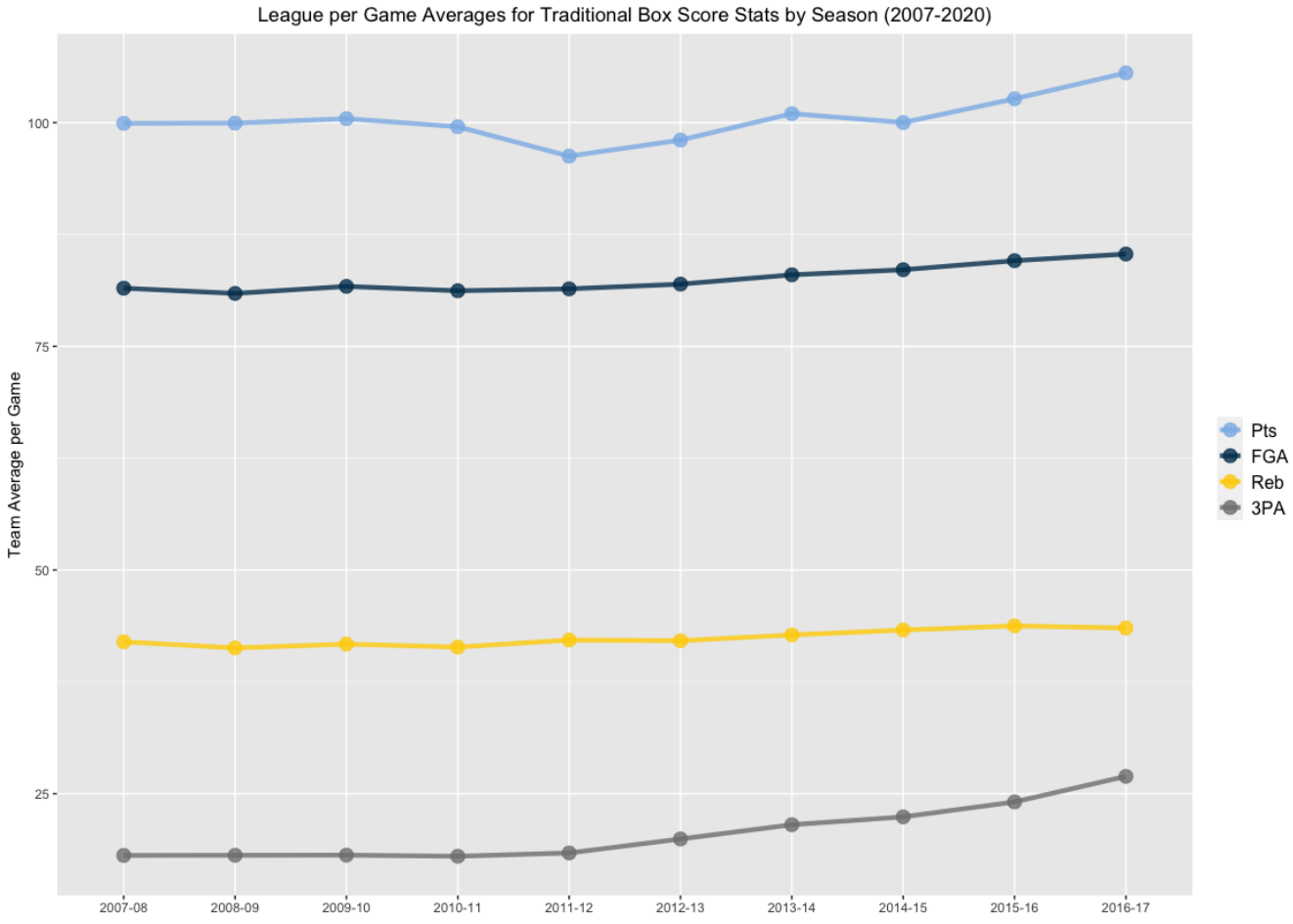

Figure 4.1: Statistical trends by season.

All four of these metrics are trending upwards, with the most drastic increase being three-point attempts. Just to accentuate this point, the team with the fewest three-point attempts per game in 2019-20 (the Indiana Pacers at 27.5 per game), would have ranked first place by almost a full three-point attempt per game more in the 2007-08 season (the Golden State Warriors ranked first with 26.6 per game). The NBA-leading Houston Rockets, who averaged 44.3 threes per game in 2019-20, calculate out to a 145% increase in three-point attempts compared to the league average 18.1 attempts in 2007-08. Going further, 48% of

the Rockets field-goal attempts this year came from behind the arc, compared to the league average of 22% back in 2007-08. The league average in 2019-20 was 38%.

Three pointers are up, shooting attempts are up, scoring is up, rebounding is up, all of today's traditional box score metrics are inflated compared to earlier in the decade. The question that emerges is whether this inflation is due to optimizations in offensive schemes or merely a change in the flow of the game. The eye test suggests that the NBA game is much faster paced today than it was a decade ago. As it turns out, the data backs this belief up.

The more possessions a team gets in a game, the higher tempo that game is being played at. The basketball statistic "pace" is defined as the average of both team's possessions per 48 minutes of combined gameplay.

$$\text{Pace} = 48 \times \frac{(TmPoss + OppPoss)}{2 \times (TmMin/5)}$$

The season-to-season trend in pace verifies our assumption that the game is played at a higher velocity today. Figure 4.2 shows the positive slope on the regression of pace by season demonstrated as both a scatterplot and a series of box plots.

4.2 Adjusting for Pace

The solution that the basketball analytics community has put in place to combat this phenomenon of statistical inflation is pace-adjusted metrics. Instead of aggregating stats on a per-game basis, we now examine those same stats compared to how many possessions that team had in the game. More specifically, the standardized pace-adjusted metric looks at a team's stats per 100 possessions. Previously, adjusting statistics on a per-48-minute basis was used to account for overtime games, but using per-100 possessions addresses both overtime instances as well as seasonal and daily shifts in gameplay.

When we build our model, it is important to handle this inflation factor as we don't want

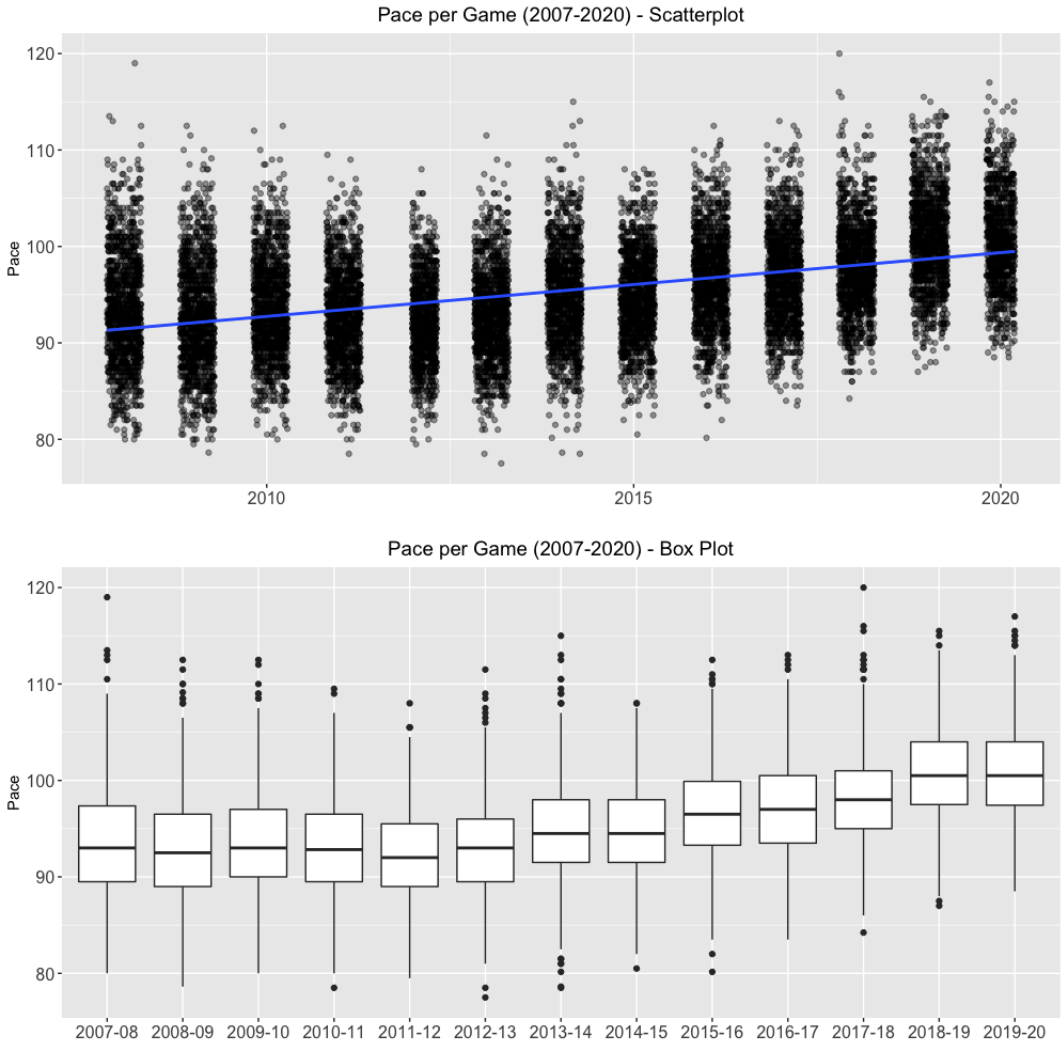

Figure 4.2: Team pace by season.

a "good" team back in 2007 to appear like a "bad" team by 2020 standards. The top two charts in Figure 4.3 show the distribution of scoring and field goal attempts for all games in the 2007-08 season versus the 2019-20 season. As is apparent, the 2019-20 season is shifted to the right verifying this statistical inflation. However, when scaling these two metrics onto a per-100 possessions scale, we see a far greater overlap between the two distributions. The increase in pace does not account for the entirety of the scoring surges, but it does lessen the gap.

For this research, we will scale all of the standard box score metrics on a per-100 possession basis. This methodology will help dramatically when trying to train a dataset that

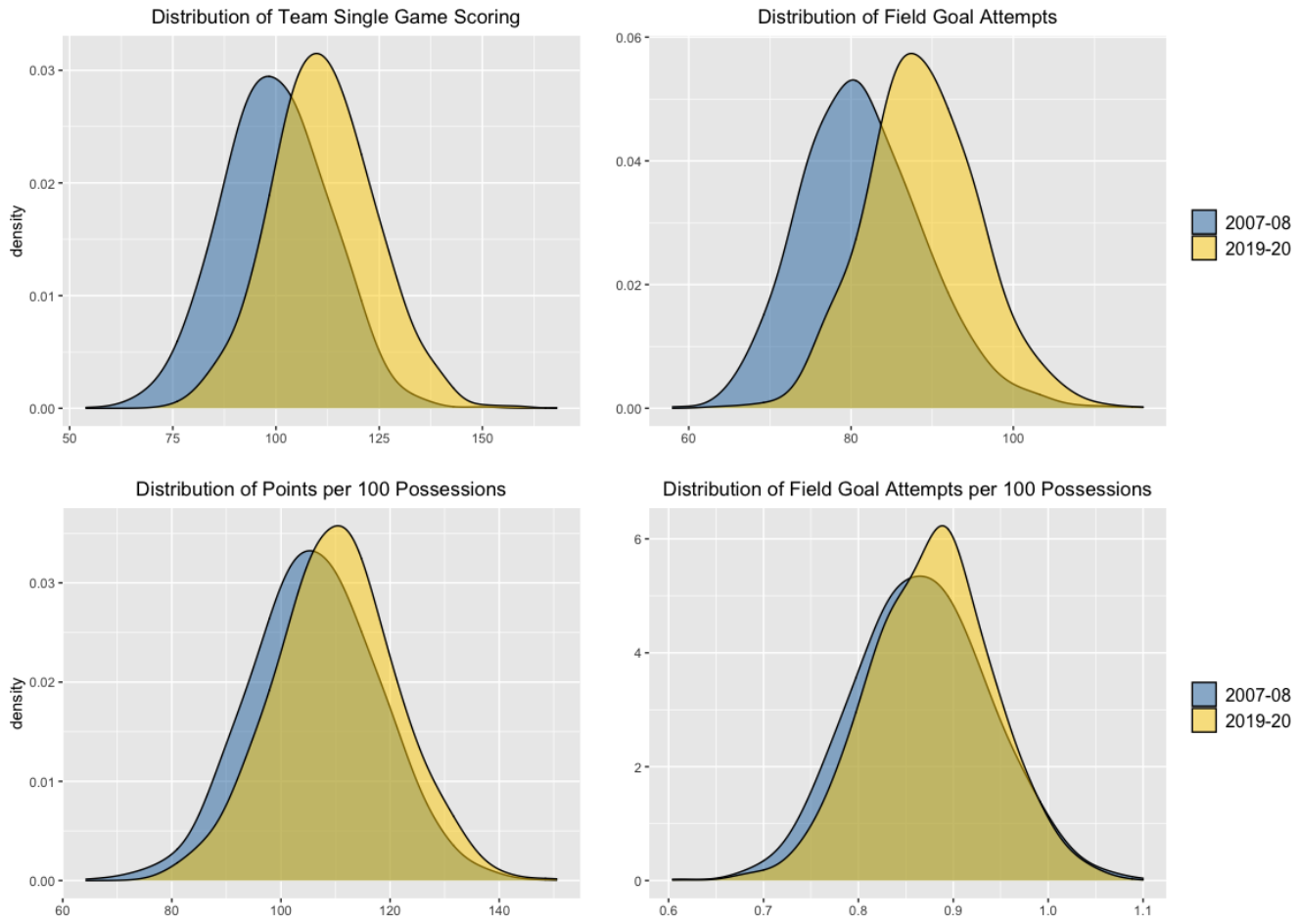

Figure 4.3: Pace adjusted metrics, 2007-08 vs. 2019-20.

spans over a long time frame by creating a standardization across the ever-changing styles of play. Points scored per 100 possessions (Offensive Rating), points allowed per 100 possessions (Defensive Rating), and the difference between the two (Net Rating) are some of the key metrics used by the analytics community to define the quality of a team's performance. We revisit these ratings in the modeling sections later in this study. Figures 4.4, 4.5, 4.6 paint a rather clear picture as to the correlation between the best teams in the league and their performance in those three advanced ratings.

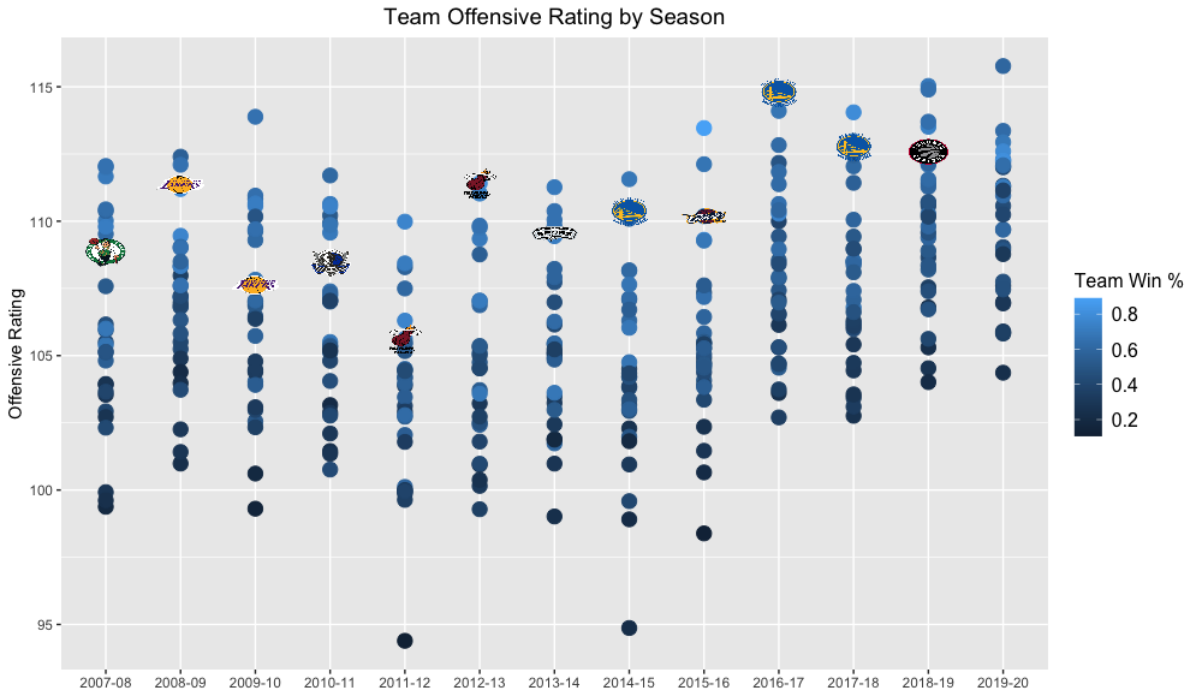

Figure 4.4: Offensive rating by season (NBA champion indicated by team logo).

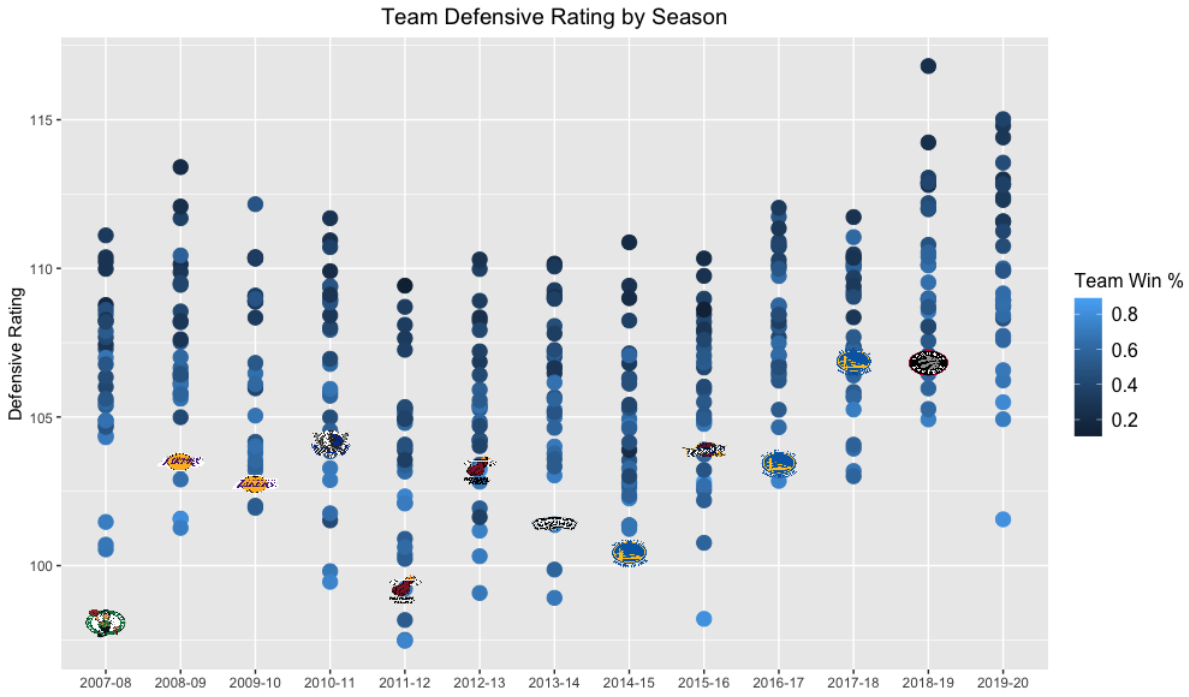

Figure 4.5: Defensive rating by season (NBA champion indicated by team logo).

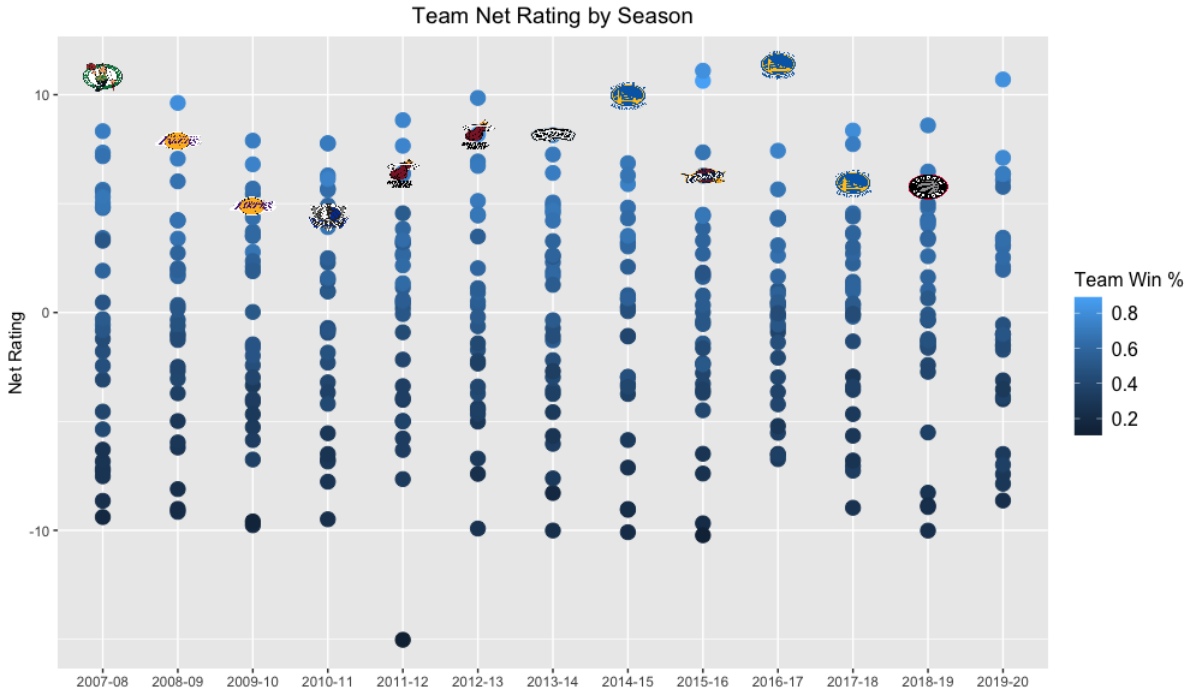

Figure 4.6: Net rating by season (NBA champion indicated by team logo).

4.3 Aggregation

A complete dataset of box scores and matchup details provides the opportunity for a lot of flexibility in how to best aggregate the data. Before aggregating though, there was more pre-processing required to handle all aspects for a team and their matchup. First off, the data had to be joined to itself so that we knew not only the team's box score in that game, but also their opponent's. Access to the opponent's statistics for each matchup makes it much easier to understand the team's performance on both offense and defense. The number of points and offensive rebounds a team gets will be valuable in a model, but equally as valuable is the number of points they give up and offensive rebounds they allow. Also, recall those opponent possessions were part of the formula for pace.

Of course, the statistics that we need to consider when trying to predict which team won a matchup, are not the team's performance in that game, but the games leading up to it. We took two different aggregation methods to our dataset. The first was the team's year-to-date performance entering that game. So, for example, if both teams were on their tenth game of

the season, we had the first nine games worth of data aggregated for each squad to predict on that matchup. If it was the last game of the season for both teams, we had 81 games worth of data for each. The one caveat: if the game was the first of the season for that team, we used the entire previous season's worth of data as our predictors.

The second aggregation method took into account the ebbs and flows of a basketball season. Some teams go on hot streaks or cold streaks so using the entire season's stats to date might not be the best method. As an alternate aggregation option, we looked at the team's performance in the eight-game span entering that game. This rolling aggregation would hopefully be more sensitive than the year-to-date method in reflecting how the team has been performing entering the matchup. Research has shown that eight games are a large enough sample size in an NBA season to determine the quality of the team. [Chr96] The one caveat for this method: if it was the team's eighth game of the season or earlier, the aggregation would roll back to the previous season. So, if it was the team's third game of the year, the aggregation would consist of the first two games of that season and then the six last games of the previous. Again, all of these metrics in both methodologies were scaled on a per-100 possession basis, not as the combined raw totals within the aggregation period.

The final processing step before building the model was required because of the unique response variable for this study. Our goal was to predict the probability that a team would win a game against its opponent, given all the data provided for that matchup. To properly build this model, we cannot treat both teams in that matchup as separate entities. If we did, there would be no way to guarantee that the combined win probabilities for both teams in that matchup would sum to 100%. Therefore, the proper way to transform our dataset would be to join the data to itself once again. By randomly assigning a "1" or "2" to each team in a matchup, we could merge *Team 1's* data to *Team 2's*, such that there is only one row per game. This randomization technique was a better strategy than dividing the dataset into home and away teams because we preserve the `locationGame` variable. History suggests the home team wins about 59% of the time versus the away team and therefore location might be a useful predictor in our dataset. Consequently, this processing step allows us to structure the model so that the response variable *(p)* is the probability that the randomly

defined *Team 1* wins. Then we can simply find the probability that *Team 2* wins by doing *1 - p*.

Thus, our dataset went from about 32,000 rows (one row per each team in the matchup) down to about 16,000 rows (one row per game). The sample size may have halved, but the parameter set quadrupled. Before we had just the team's statistics for each game. After the processing we now had, for each aggregation technique:

1. Team 1's statistics entering each matchup
2. Team 1's opponent's statistics entering each matchup
3. Team 2's statistics entering each matchup
4. Team 2's opponent's statistics entering each matchup

The last alteration to the dataset was to remove postseason games from this particular research. In the NBA postseason, teams play a best-of-seven-game series against the same team. Also, the competition is much higher since only the eight best teams from each conference qualify. Therefore, top caliber teams would be losing more frequently than they normally would when playing the full spectrum of competition. Predicting the winner of postseason games is a critical aspect of a robust betting model, but to narrow the scope of this study, we will stick to the regular season only. With that last step, these two datasets—the eight-game rolling aggregates and the year-to-date statistics—were in a form ready for modeling. While the target response variable, to reiterate, was the binary classification of whether the randomly assigned *Team 1* of the matchup won the game.

CHAPTER 5

Model Building

Before training any preliminary models, an important first step was to see if there was any redundancy in our variable set. A quick way to examine that potential is to look at correlations/multicollinearity within the numerical predictors in the dataset. Figure 5.1 shows the breakdown for all correlations between the single-team box score metrics in the modeling set, including the advanced ones. As one familiar with basketball statistics might expect, there were some quite highly correlated variables.

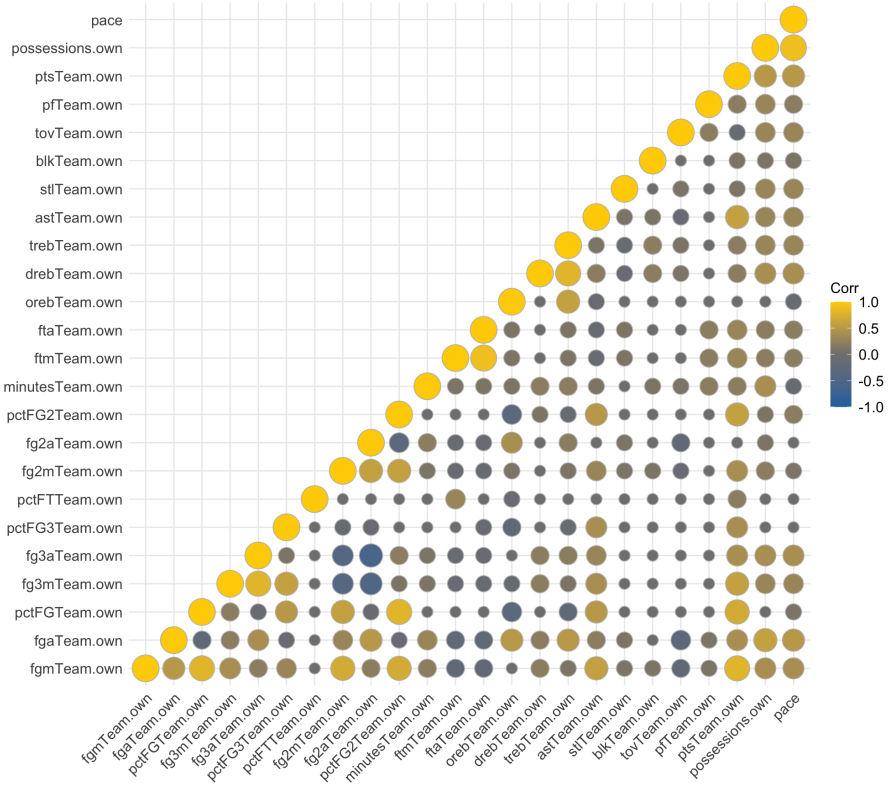

Figure 5.1: Correlation matrix for box score statistics.

For example, we calculated pace earlier in the study as basically a per-48 minute trans-

formation of possessions. And as a result, the correlation between the two variables was 0.9. Field-goal attempts were highly correlated with field goals made (0.8), free-throw attempts were associated with free throws made (0.9), and total rebounds were correlated to defensive rounds (0.8). Additionally, all of the shooting percentage metrics (field-goal percentage, two-point field-goal percentage, three-point percentage), were highly associated with the shooting attempts metrics.

Many of the standard NBA box score statistics are interrelated, which is why the analytics communities have crafted advanced metrics that bake in many of these raws statistics into a catch-call measure. Examples of these are "true shooting percentage" and "defensive rebounding percentage".[1] In further studies, it would be prudent to examine these advanced metrics developed by basketball's analytics community when building probability models similar to the one in this study. Regardless, for this research, we will stick to our available box score metrics and keep in mind the opportunity to eliminate these correlated variables when choosing the best predictors for the model.

Additionally, before any model building, we must examine if there is any apparent need for transformations of our box scores metrics. Correlations indicate whether there might be confounding issues between two different variables, but equally as important to investigate is if there exists skewness within the distribution of a single variable. Ideally, the distribution of numerical predictors in our model would resemble a symmetric distribution and normal curve. Normalized variables perform better in models. The density matrix in Figure 5.2 allows us to examine the distribution of all our numerical predictors.

We see in Figure 5.2 that the variable distributions (even when parsed between wins and losses) do not show any glaring skewness or multi-modality in our variable set. Recall, at the end of Chapter 4, we took these raw box score metrics and scaled them to a per-possession basis. Fortunately, we see the distribution of possessions also appears quite normal, implying that taking the box score statistics and scaling them by possessions will maintain the normality. We can conclude that proceeding without any transformations should suffice.

[1] https://www.basketball-reference.com/about/glossary.html

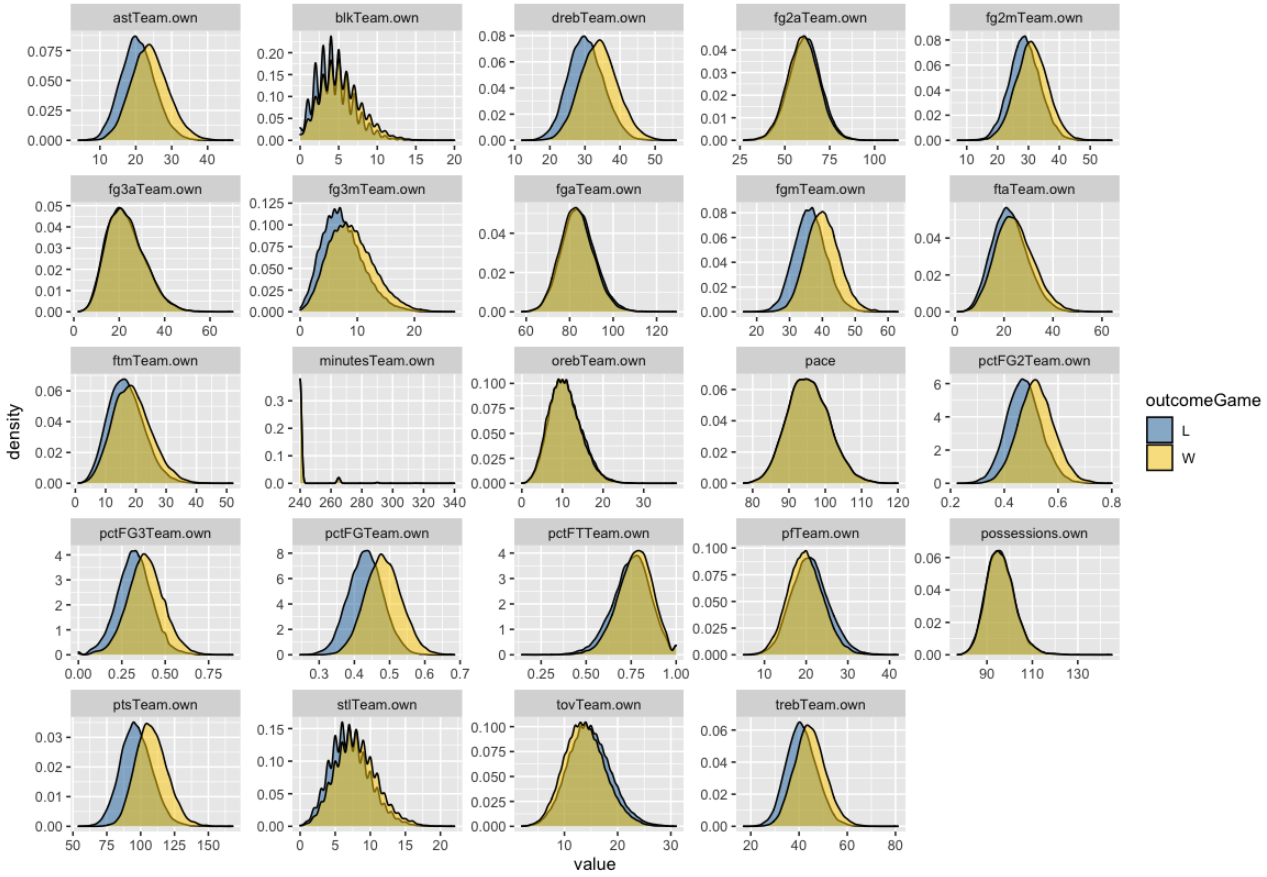

Figure 5.2: Density distribution matrix for box score statistics.

5.1 Logistic Regression

Since our response outcome is a binary variable (team 1 either won or lost) we would be remiss to not begin the model building process with an attempt at a logistic regression. The more common, linear regression, is ideal for scenarios where you are predicting an outcome that has a range of possibilities, usually continuous. For example, how much revenue a product will generate or how long a flight might take. Whereas with a logistic regression the predicted Y values are restricted by asymptotes that exist at 0 and 1. Logistic regression is reserved for models with two outcomes: whether a patient tests positive or negative for a disease, if a person will default on their credit or not, or if a team will win or lose a matchup.

In fact, for our particular study, logistic regression is a perfect model type to explore as it does not output a single binary response. Logistic models return a probability score that

reflects the predicted probability of the occurrence of that event. That probability is then rounded (usually at 50%) to determine whether the model predicts a positive or negative response. Figure 5.3 depicts a logistic regression curve and the range of outcomes that can be outputted from a model. [Jos19]

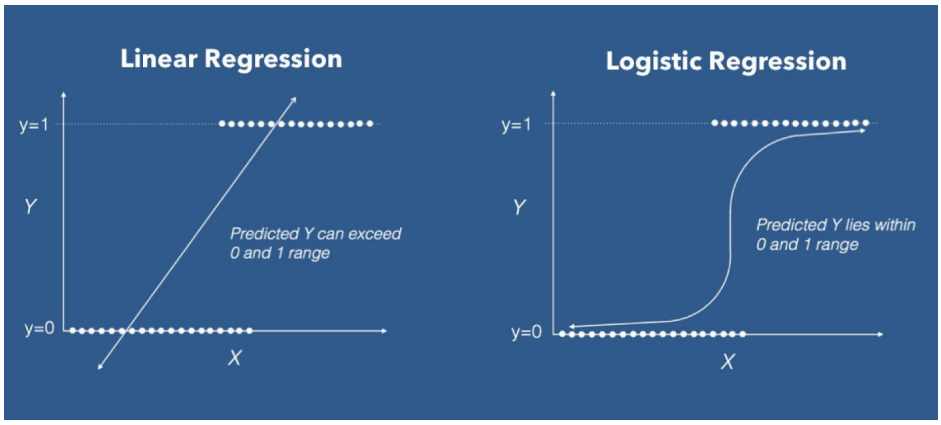

Figure 5.3: Simplified examples of linear regression versus logistic regression. [Jos19]

Logistic models, and most machine learning algorithms, operate best when only the top critical features are included. Reasons for this reality of model building are many: fewer features allow for algorithms to train faster, improves the interpretability of the final result, reduces overfitting, and, when properly selected, fewer features can improve the accuracy of model versus one with far more. We examined two separate methods for feature selection: recursive feature elimination and stepwise selection.

To conduct the model building in this study we used the *caret* package in R. *Caret* is short for "classification and regression training" and contains a variety of tools to prep the data for modeling, build training and testing sets, and has over 200 different machine learning algorithms available. *Caret* also provided the functionality necessary to perform feature selection techniques and hyperparameter tuning. All the modeling and evaluation code can be found on GitHub. [Dot20].

Recursive feature elimination (RFE) finds the best performing feature subset by utilizing a greedy optimization algorithm. Essentially it repeatedly makes model after model and sets aside the best features after going through all iterations. In the end, it ranks the variables

based on importance (the order of elimination). Stepwise feature elimination is a similar concept. Stepwise elimination can be done using "forward" selection or "backwards" selection. In forward selection it begins with a feature-less model, then adds a predictor one by one such that each added variable improves the model performance. This process continues until the model no longer receives additional performance improvements. Backwards selection is just the opposite, starting with a model with all the features and removes them systematically until the optimal final form is reached.

As it turned out, stepwise selection and recursive feature elimination settled on two rather different predictor sets as shown in Figure 5.4. That said, the resulting performance metrics for the two methodologies were quite similar. We then applied the selected feature-set from each selection methodology to both aggregation datasets: year-to-date and trailing eight games. We used a 75% and 25% train/test split for model training based on the games from the 2007-08 season through 2018-19—a total of 10,830 games. Additionally, we used a standard 10-fold cross validation technique to increase the robustness of our model through resampling. Figure 5.5 shows the model performance results for these four logistic models. (Note: accuracy is defined as the ratio of correct responses to the total. Kappa is defined as the observed accuracy compared to the expected accuracy.)

$$Accuracy = \frac{TruePos + TrueNeg}{Total} \qquad Kappa = \frac{ObsAccuracy - ExpAccuracy}{1 - ExpAccuracy}$$

The best of the four models, by a narrow margin, ended up being the one generated through recursive feature selection on the year-to-date dataset. Of the roughly 3,600 games in our test dataset, it yielded an accuracy of 65.9% with a Kappa score of 31.9%. The resulting confusion matrix is shown in Table 5.1.

5.2 Random Forest

The next machine learning technique that we looked to implement was the random forest model. A random forest consists of grouped and randomized decision trees. Each decision

Table 5.1: Confusion matrix results for the most performant logistic regression.

	Actual - Win	Actual - Loss
Predicted - Win	1209	631
Predicted - Loss	604	1181

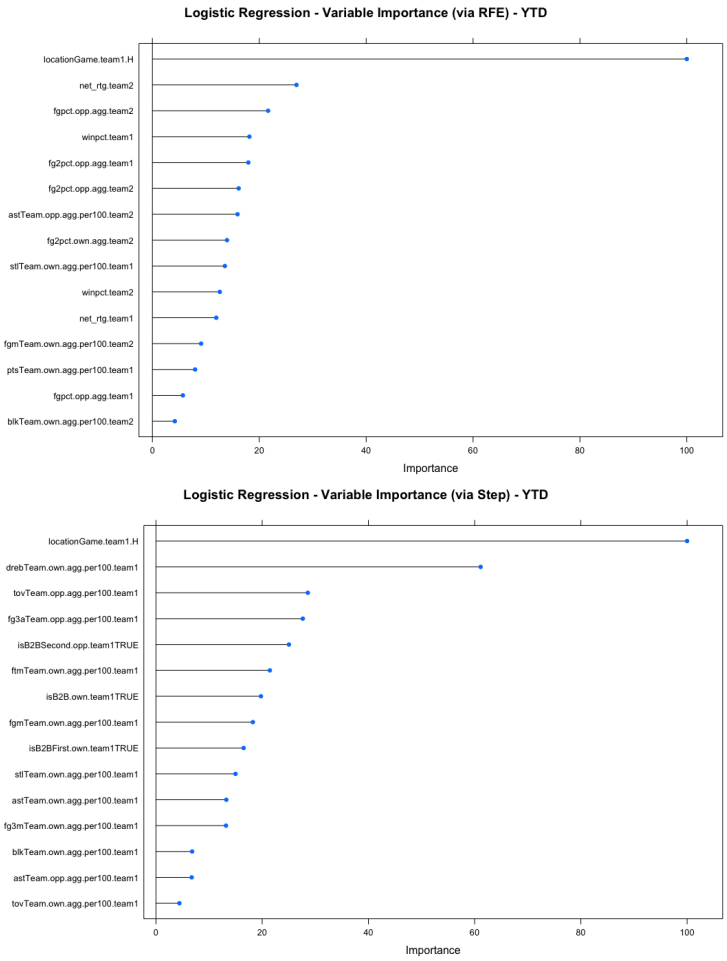

Figure 5.4: Top variables by importance from both selection methodologies.

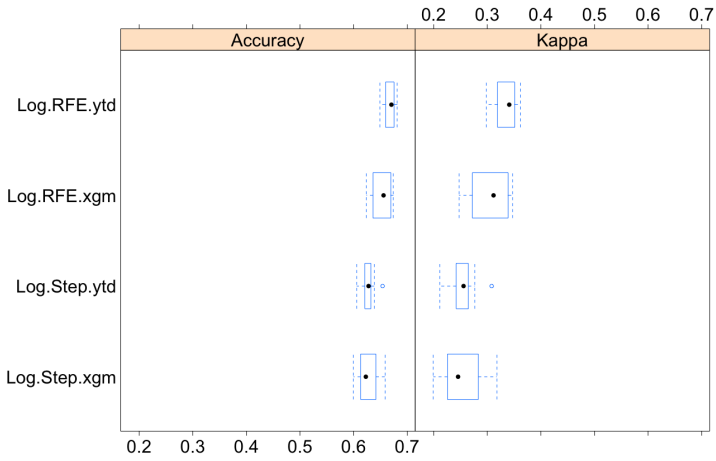

Figure 5.5: Performance of our four different logistic regression models.

tree randomly selects a variable from the predictor set and establishes a certain cutoff range for the logic flow to decide upon. For example: is team 1 the home team? Or, is team 1's field-goal percentage greater than 45%? Each decision tree is built on a random sample of the original data and at each tree node the subset of features selected are is done randomly.

The random forest classification algorithm continuously iterates through this process of generating feature trees. The algorithm's goal is to create an uncorrelated forest of trees whose prediction by committee is more accurate than that of any individual tree. Eventually the model plateaus in regards to the number of trees and splits within those trees such that there are no more performance improvements to warrant continuing tree building.

We used the same model preparation methodology for the random forest model as we did for logistic regression. So we had all games from 2007-08 until 2018-19 as the training set split into a 75%/25% partition. Then, using the same 10-fold cross-validation technique, we built a random forest model on both the year-to-date dataset and eight-game spans dataset. The performance results of the two models shown in Figure 5.6 suggest that, once again, the year-to-date aggregation was marginally better for prediction performance.

The most important variables from the random forest model are shown in Figure 5.7 while the confusion matrix results are shown in Table 5.2. The accuracy of this final model was 64.9% while the kappa score was 29.8%.

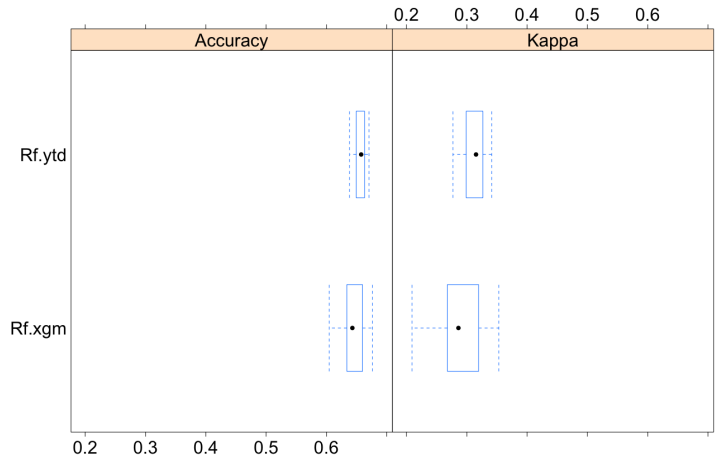

Figure 5.6: Performance of the random forest model on our two aggregation data sets.

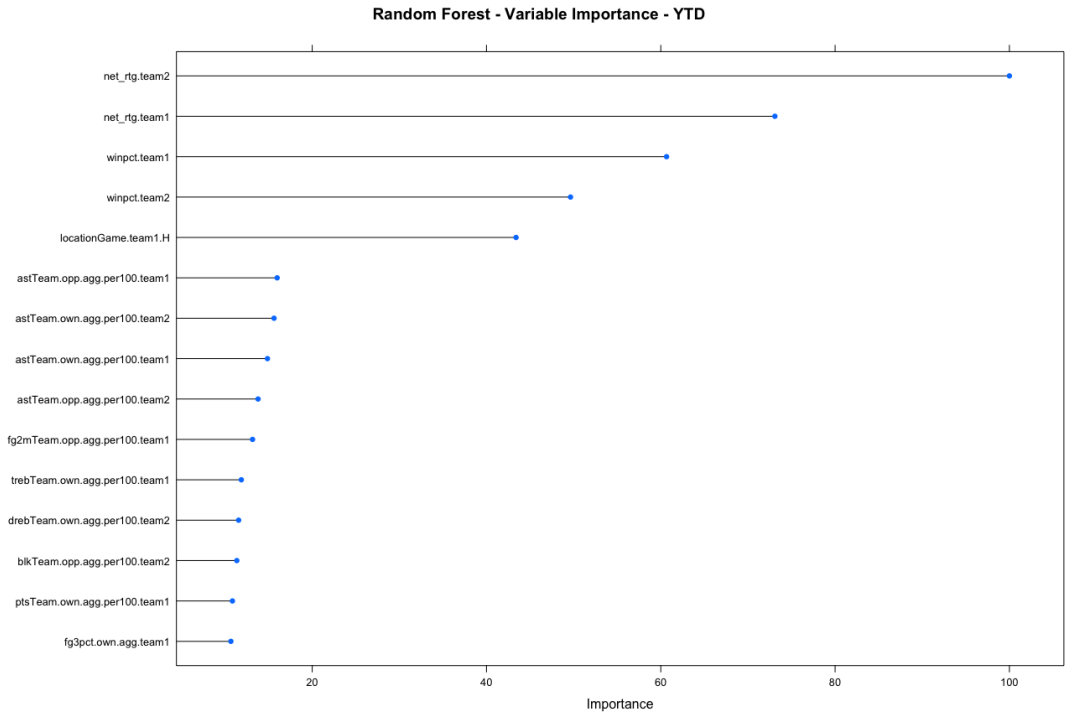

Figure 5.7: Top 20 variables by importance in the year-to-date random forest model.

Table 5.2: Confusion matrix results for the year-to-date random forest model.

	Actual - Win	Actual - Loss
Predicted - Win	1174	634
Predicted - Loss	639	1178

5.3 XGBoost

Finally, to expand upon the random forest model, we examined one of the more popular and powerful tree-based methodologies available in machine learning: XGBoost. XGBoost is an ensemble based algorithm that uses a technique called boosting to improve predictive performance and of the boosting methods out there, XGBoost is maybe the most advanced. Like all machine learning algorithms, gradient boosting makes use of a loss function that goes through an optimization process to guide the direction of the model behavior. Each iteration makes a prediction based on a decision tree (called a weak learner) and then iterates to another one. These weak learner decision trees get bundled together and over time become more accurate. The steps described are similar to how a random forest process works as well. But there are two main differentiators between random forest and gradient boosting.

First, random forests and gradient boosting models are built differently. Random forests build each tree independently while gradient boosting builds one tree at a time. This additive model (ensemble) works in a forward stage-wise manner, introducing a weak learner to improve the shortcomings of existing weak learners.

Random forests and gradient boosting also combine models differently. Random forests combine results at the end of the process (by averaging or "majority rules") while gradient boosting combines results along the way. [Gle19]

Because boosting-based algorithms are constantly improving and "pruning" trees throughout the building process it leads to more refined trees. XGBoost, in particular, is considered the most powerful of the boosting methods due to its considerable complexity. Although simple enough to implement using the *caret* package in R, this technique's highly involved nature can tend to be a victim to overfitting more so than random forest models do. Additionally it is far more intensive computationally. For reference, the random forest model took about one hour to train on our dataset of about 11,000 games while XGBoost took five times longer.

Implementing XGBoost was a similar process as before. Once again, we used our training set of data from 2007-08 to 2018-19 for both the year-to-date and eight-game span datasets.

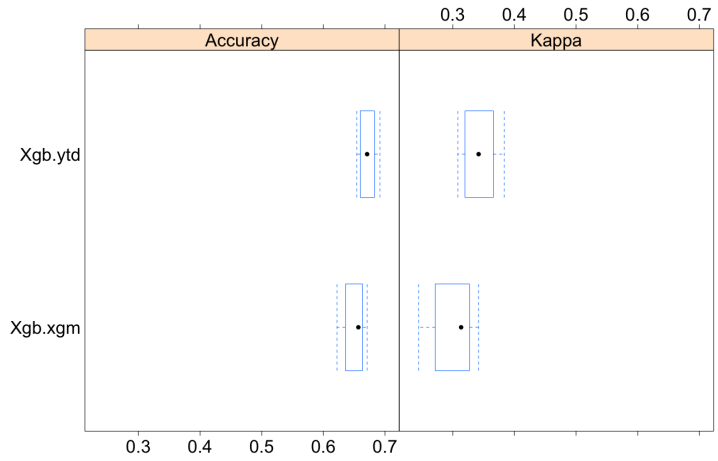

Figure 5.8: Performance of the XGBoost model on our two aggregation data sets.

Table 5.3: Confusion matrix results for the year-to-date XGBoost model.

	Actual - Win	Actual - Loss
Predicted - Win	1191	614
Predicted - Loss	622	1198

But this time, plugging the data into the XGBoost algorithm in the *caret* package while still using the same hyperparameters as the random forest model. Tuning the hyperparameters of the XGBoost model is somewhat of an art, as even the slightest alterations to the number of trees, depth of those trees, cross-validation folds, etc. can lead to vastly improved results. For the simplicity of this research, we kept the hyperparameters consistent from model to model. But if one were truly attempting to build a powerful enough tool to invest their money into some real-world gambling stakes, optimizing the parameters to each appropriate model would be necessary. Regardless, the modeling results from the XGBoost algorithm for each dataset are shown in Figure 5.8. Again the year-to-date model was superior and its feature importance breakdown is shown in Figure 5.9. Lastly, the confusion matrix is in Table 5.3 with an accuracy metric from the testing dataset of 65.9% and a kappa score of 31.8%.

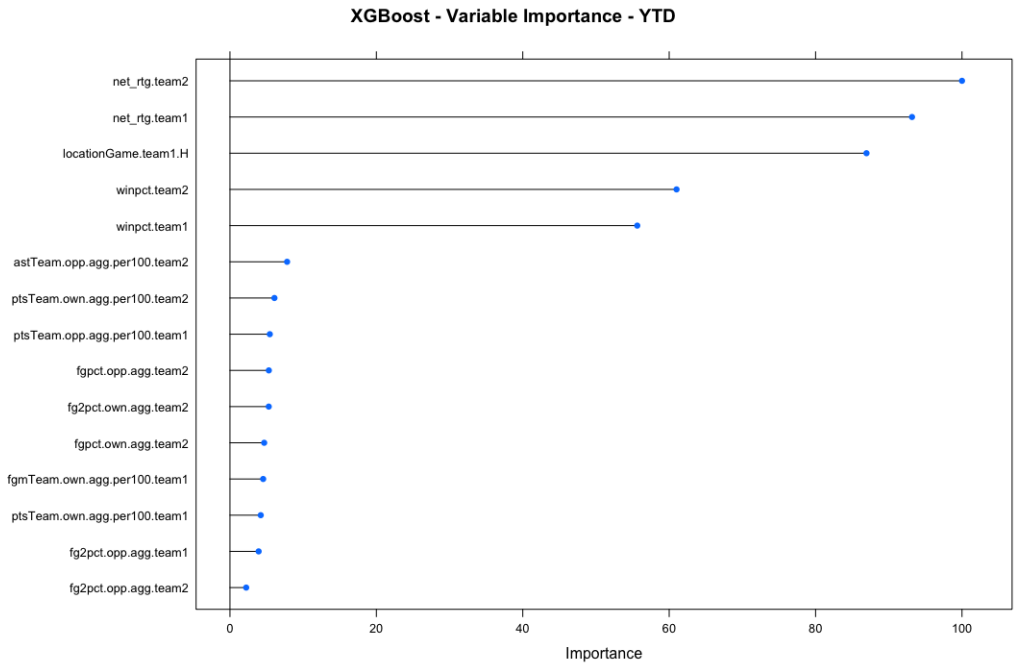

Figure 5.9: Top 20 variables by importance in the year-to-date XGBoost model.

5.4 Modeling Summary

Of the three different models we built—and several variations of those three models—they all performed relatively similarly. Each of them outputted an accuracy rate of around 60% on the test set of about 3,600 randomly selected games throughout the 2007-08 to 2018-19 seasons. All in all, these results were very positive as each model performed better than chance (50% accuracy) by a sizable margin. In summation, we can reach the following conclusions from our modeling experiments.

First of all, certain variables appeared most frequently across our most performance models suggesting these indicators are most critical to predicting win likelihood. Examples of these highly important predictors were, net rating, winning percentage, location (home versus away), points scored per 100 possessions (offensive rating), points allowed per 100 possessions (defensive rating), field goal percentage, and assists per 100 possessions.

Secondly, the year-to-date aggregation method had better results than the eight-game span aggregates. The general premise for using the shorter spans as opposed to full season

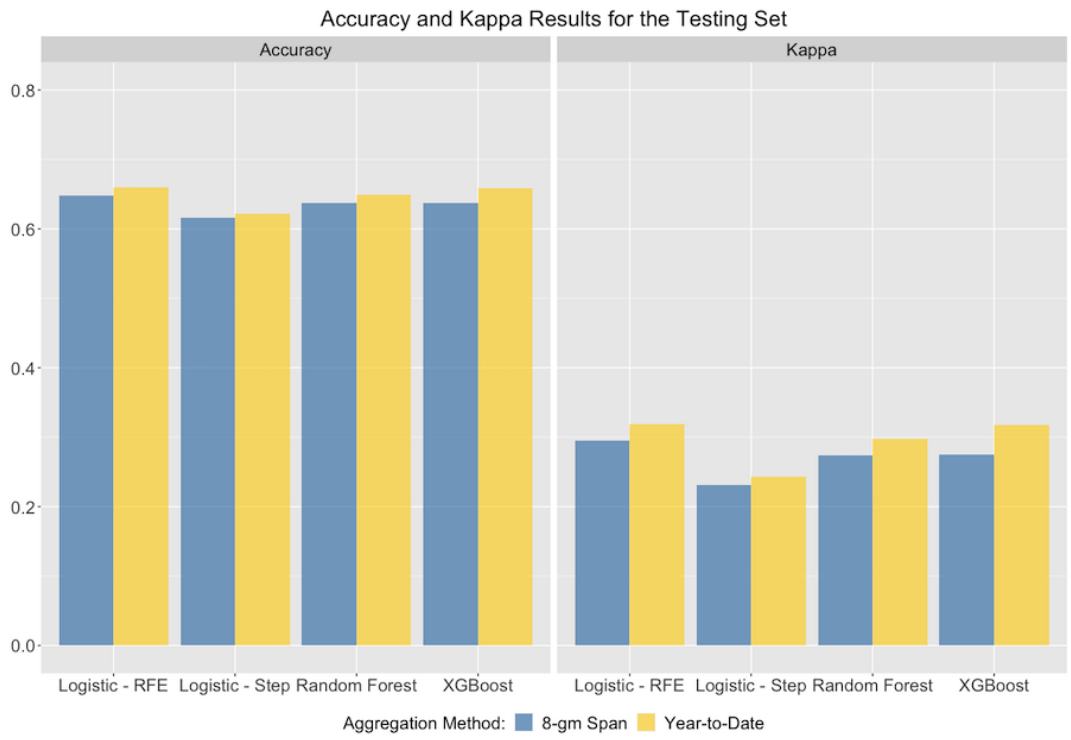

Figure 5.10: Performance results from all model types and aggregation methods in this study.

metrics was to catch the trends that might be occurring over the year. However, it appeared that the larger sample size generated from season-long aggregations provided better predictive power. That said, the difference between the two was not remarkably different and we discuss the implications of their similarities in Section 9.1.

Lastly, we can conclude, of the three model types conducted thus far, the most performant one was the logistic model that used the top 20 most important variables as selected by the recursive feature elimination method. Although not by a large margin, it did yield the highest accuracy in the testing set and therefore was used in Chapter 7 for the betting simulation. Additionally, the logistic, year-to-date model also performed the best in its F1-score. F1 is a measure commonly used for binary response variables as a metric to balance precision (how many of the positive predictions were actually positive) and recall (how many of the actual positives were predicted as positive).

A summary of the performance metrics of all models built is shown in Figure 5.10 and Table 5.4 as well as the precision, recall, and F1 scores in Table 5.5.

Table 5.4: Summary metrics for all model types and aggregation methods on testing dataset.

Model Type	Aggregation Type	Accuracy	Accuracy (95% CI)	Kappa
Logistic - RFE	8-game span	0.647	[0.6316, 0.6631]	0.295
Logistic - RFE	Year-to-date	0.659	[0.6436, 0.6747]	0.319
Logistic - Step	8-game span	0.616	[0.5994, 0.6315]	0.231
Logistic - Step	Year-to-date	0.622	[0.6055, 0.6373]	0.243
Random Forest	8-game span	0.637	[0.6207, 0.6524]	0.273
Random Forest	Year-to-date	0.649	[0.6328, 0.6641]	0.297
XGBoost	8-game span	0.638	[0.6218, 0.6535]	0.275
XGBoost	Year-to-date	0.659	[0.6433, 0.6745]	0.318

$$Precision = \frac{TruePos}{TruePos + FalsePos} \qquad Recall = \frac{TruePos}{TruePos + FalseNeg}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Table 5.5: Extra summary metrics for all models and aggregation methods on test dataset.

Model Type	Aggregation Type	Precision	Recall	F1
Logistic - RFE	8-game span	0.6480	0.6510	0.6490
Logistic - RFE	Year-to-date	0.6570	0.6670	0.6620
Logistic - Step	8-game span	0.6140	0.6286	0.6212
Logistic - Step	Year-to-date	0.6203	0.6271	0.6237
Random Forest	8-game span	0.6353	0.6469	0.6410
Random Forest	Year-to-date	0.6490	0.6460	0.6480
XGBoost	8-game span	0.6373	0.6447	0.6409
XGBoost	Year-to-date	0.6600	0.6570	0.6580

CHAPTER 6

Neural Networks

For one final modeling approach we looked at maybe the most cutting edge tool used in data science and machine learning today: the artificial neural network. Now, one could write an entire dissertation on the application of neural networks in sports wagering predictive models. But for this research we just wanted to briefly broach the subject and test out its capabilities in comparison to the other classification models we built.

6.1 Basic Structure and Theory

As the name suggests, neural networks are inspired by the science behind neurons and how they process and send information in and out of the brain. A neural network model consists of a series of layers composed of nodes. The first layer is the input layer in which information is passed into the model, with each node representing a certain data point. The middle layer is called the hidden layer in which the input data passes through and is processed until it emerges in the final layer, the output layer. The output layer is the response variable, so in our instance, the output layer would consist of two nodes: win and loss. A diagram of a neural network model is shown in Figure 6.1.

The more hidden layers that a model contains, the more complex the processing is. This concept of adding more and more hidden layers is known as "deep learning." One could consider these hidden layer nodes as a bunch of dials that are constantly tuning themselves to optimize to the most accurate possible prediction in the output layer. As the model learns, using a technique called "backpropagation," the nodes within the hidden layers continue to adjust according to the minimization of a loss function. The more data inputted and

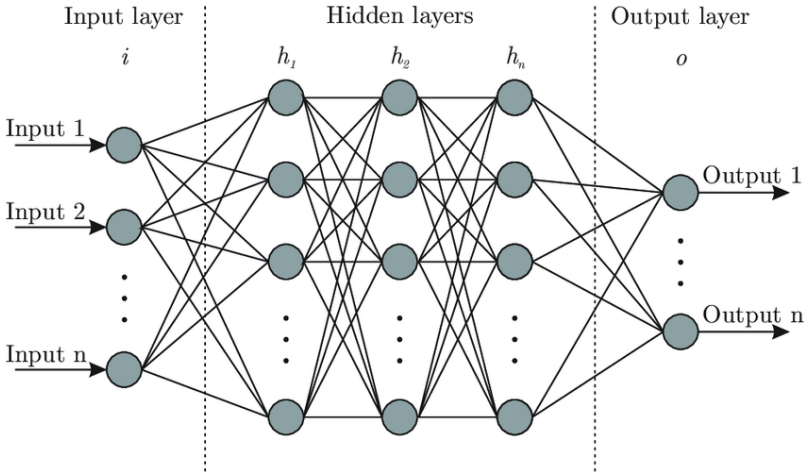

Figure 6.1: Basic diagram of a neural network with three hidden layers. [Shu19]

more complex the tuning, the better the output. However, because of the nature of deep learning and artificial networks, these models are relatively difficult to interpret and operate as somewhat of a "black box." For this reason, neural networks are very common choices for complex systems like image recognition or video game AI decision making, but rarely the first choice in statistical modeling. That said, because of their processing power, our general interest in the subject matter, and the popularity of deep learning, we built a relatively simple neural network to compare against our previous three models.

6.2 Model Processing and Results

As it turns out, the *caret* package provides all the necessary functionality required to carry out a relatively rudimentary neural network model. Two of the most common libraries used in data science today for building neural networks are called *Keras* and *Tensor Flow* and provide a multitude of capabilities that allow as much customization to one's model as desired. The packages allow for tweaking the number of hidden layers, the number of nodes within those hidden layers, the type of activation (optimization) function that is used, and more. But for our purposes, *caret* provided enough capability to build a neural net with our dataset.

One intricacy in neural networks is that the model performance is far better when the

structure of the inputted data has been scaled. Whether it's scaling the data so all values are between 0 and 1, -1 and 1, or something similar, preprocessing the data is a critical part of neural network implementation. For our model we normalized the data, meaning we centered it (subtracted each value it by the mean) and then scaled it (divided each value by the standard deviation). These normalized inputs would perform far better than if we put the raw aggregated box score metrics that were used in the other models. (Note that because neural networks are not highly interpretable, transforming the inputs significantly is not as problematic as doing so would be for standard regressions.)

After scaling the inputs, we used the same 10-fold cross-validation as the previous models but included some other hyperparameters as well. *Caret* allows the specification of two key parameters: size and decay. Size is the number of units in the hidden layer, while decay is a regularization parameter to avoid overfitting. A general rule of thumb in deep learning is to set the size of the hidden layer equal to about two-thirds of the number of inputs. We had 105 total variables so we set a few different hidden layer options ranging from 70 down to 3. We also set a few different decay options for the model to iterate over.

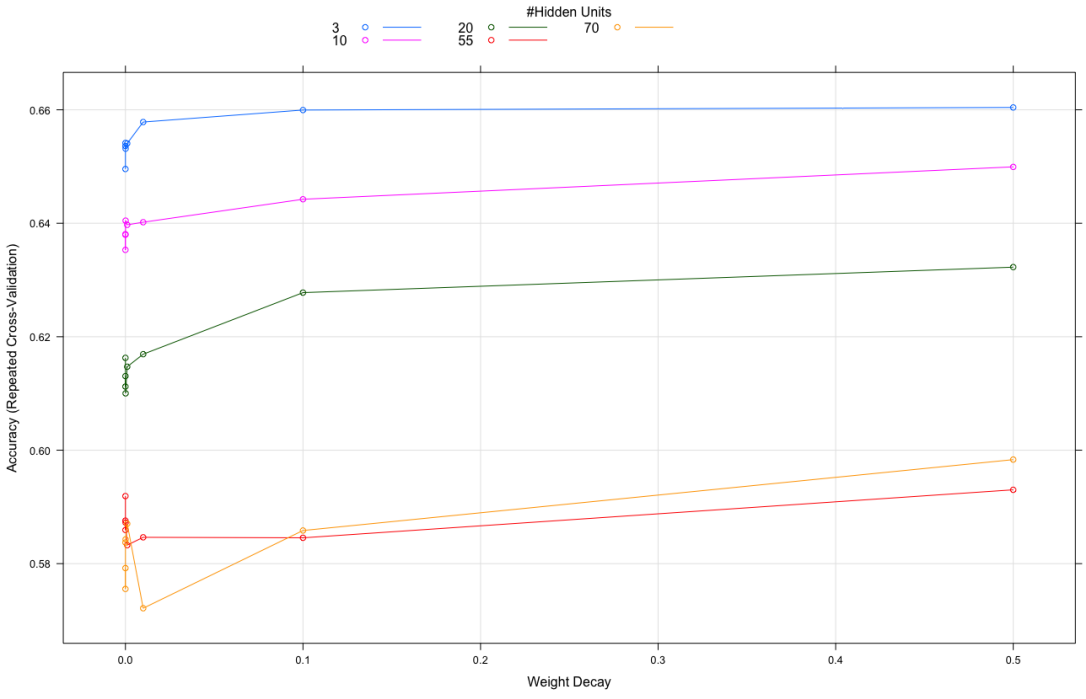

Figure 6.2: Accuracy of each neural network based on each combination of hyperparameters.

In the end, the most performant neural network was surprisingly the one with just three nodes in the hidden layer, Figure 6.2 depicts *caret's* attempts at finding the best neural network configuration through all the hyperparameter permutations. To keep the methodology consistent, we used the same training/testing sets and built a neural network for each aggregation method. The results are shown in Figure 6.3 and, once again, the year-to-date model was the more performant of the two. Applying our model to the testing data set we got the confusion matrix in Table 6.1 which results in an accuracy of 65.5% and kappa score of 31.0%.

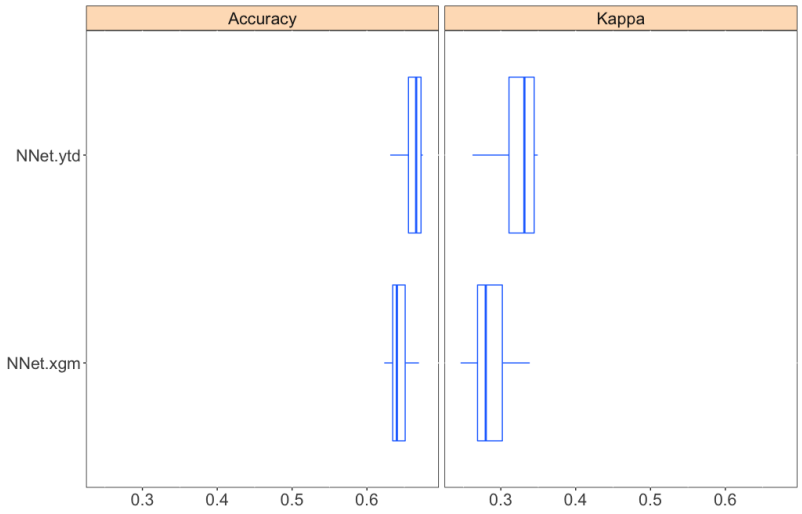

Figure 6.3: Performance of the neural network model on our two aggregation data sets.

Table 6.1: Confusion matrix results for the year-to-date neural network model.

	Actual - Win	Actual - Loss
Predicted - Win	1176	614
Predicted - Loss	637	1198

These outcomes are quite impressive as we were able to get as performant, if not better, results with a neural network than we did with the other modeling techniques. The neural network outperformed almost all the models even though we did not need to perform any feature selection. We merely scaled all the data and inputted the entirety of the dataset

into the network. A limitation of the *caret* package is that it only allows the user to build a network with a single hidden layer. One could imagine that by using the more robust machine learning packages mentioned earlier, and incorporating deep learning methodologies with multiple hidden layers, the performance of this network could improve even further. It's evident why neural networks are all the rage in data science and computing; their power is impressive to say the least. But for the rest of this study, we will proceed with our more interpretable logistic model. Despite being far less complex and using fewer predictors, the logistic regression results were within a margin of similarity of all the models and even up-to-par with the involved neural networks.

CHAPTER 7

Beating the Book

We've demonstrated various highly performant models with statistically significant accuracy metrics, but the intention of this research was not merely to conduct a series of modeling experiments, determine the the most performant one, and then stop there. Our model evaluation process concluded that the logistic model performed best on the training and testing sets, but the true worth of this prognostication tool would be in its application to the real world: the betting markets. We had two hypotheses related to building a win probability model for NBA games. One, that the availability of data could allow anyone with the toolset to create a prediction system that could emulate what the sportsbook experts do on daily in establishing their betting lines. Two, that inherent biases might exist within these lines based on factors outside of pure win expectancy that could lead to edges that were ripe for exploiting by an objective model.

Now in the world of professional sports betting, it is not necessary to perform significantly better than the sportsbook line setters. There also does not need to be a considerable edge from irrational bettors to capitalize on. The magic number in sports betting is not 100% as it is to perfectionists. Nor is it 99.7%, 95%, or 68% as it is to the "empirically-ruled" Gaussians. In sports wagering, statistical significance for bettors (and a number every serious sports gambler has engrained in their mind) lies at the all-important benchmark of 52.4%. [Cul18]

7.1 Why 52.4% Matters

Back in Chapter 2, we broke down the mathematics and economics of sports betting. We derived the formula to convert odds to both expected profits as well as win probabilities. From Equation 2.6, we can determine the moneyline that one would expect in a matchup between two evenly matched teams; that is, each team's chances of winning is 50%.

Expected Moneyline for an Even Matchup:

$$
\begin{aligned}
IP_{fav}(ML) &= \frac{-ML}{-ML+100} \\
0.50 &= \frac{-ML}{-ML+100} \\
0.50(-ML+100) &= -ML \\
\frac{-ML}{2} + 50 &= -ML \\
\frac{ML}{2} &= -50 \\
ML &= -100
\end{aligned}
$$

The same result occurs (but +100 moneyline) if we set *I(P)* equal to 0.50 in the formula for the implied probability of an underdog. And these results make sense. If you were betting $100 on a 50/50 coin flip, you would expect to get $100 profit if your side landed face up. But of course, as referenced previously, sportsbooks always take their share of the betting pool—their vig. Typically, for an even-odds game, a sportsbook will take a 10% cut of the betting margin. This means that you'd have to put in $110 on a team to win $100 (essentially a -110 moneyline). So to find the break-even point on a -110 moneyline payout we simply plot the return on investment percentage as your win probability increases, as shown in Figure 7.1.

Win %	EV	ROI
48%	$ (9.20)	-8.36%
49%	$ (7.10)	-6.45%
50%	$ (5.00)	-4.55%
51%	$ (2.90)	-2.64%
52%	$ (0.80)	0.73%
53%	$ 1.30	1.18%
54%	$ 3.40	3.09%
55%	$ 5.50	5.00%

Figure 7.1: Break-even point based on expected return on investment for a -110 moneyline bet.

Expected Return on Investment vs. Win Probability:

$$
\begin{aligned}
ExpectedROI &= \frac{ExpectedProfit}{Risk} \\
EP &= P(W) \times Profit - P(L) \times Risk \\
EP &= P(W) \times \$100 - P(L) \times \$110 \\
EP &= P(W) \times \$100 - (1 - P(W)) \times \$110 \\
EP &= \$210P(W) - 110
\end{aligned}
\tag{7.1}
$$

The break-even point is defined as the point where your ROI at a certain win probability goes from an expected loss to an expected gain. Solving Equation 7.1 after setting EP equal to 0 gets you to the magic number of 52.4% and this point is clearly shown in Figure 7.1. So as long as sportsbooks continue to take their 10% cut and place coin-flip games at -110 odds, bettors will have to win 52.4% of the time to keep their bankroll from slowly draining. Another quick way to get to this break-even point is by plugging a -110 moneyline into our

implied probability formula.

Implied Win Probability of a -110 moneyline:

$$IP(ML) = \frac{(-ML)}{(-ML) + 100}$$
$$IP(ML) = \frac{(-1 \times -110)}{(-1 \times -110) + 100}$$
$$IP(ML) = \frac{110}{110 + 100}$$
$$IP(ML) = .5238$$

In short, bettors who win more than 52.4% of their bets will "beat the market" and thusly their portfolio increases over time. Those that fall below the threshold will trend toward the negative.

Quick note: the table in Figure 7.1 shows an ROI of -4.55% at a win probability of 50%. This essentially defines the casino's vig when placing their cut at 10%. Given an even-odds matchup (50%), a sportsbook, on average, will take a 4.55% vig from the betting pool.

7.2 Fixed Bet Implementation

This magic 52.4% metric for sports bettors is most applicable when wagering on spreads, over/unders, or even-money games. These make up the most common wagers placed at sportsbooks and therefore that magic number is critical for high volume gamblers. But for this study, however, we are analyzing moneylines which have varying profits based on the extent each team is a favorite or underdog. So to best gauge the success of our model, beating that 52.4% win percentage is not the important metric to evaluate our betting performance. In the end, what matters is our betting return on investment. How much profit was realized versus how much money was wagered? We used a variety of betting strategies to test and simulate our model's predictive power.

The first approach is the most intuitive, but potentially not the most lucrative. Option one is to look at the percent chances our model predicts a given team has to win the game and always pick the favorite. If the model predicts a win probability greater than 50%, then

we bet on that team. This is also, more or less, how a casual bettor approaches making a wager. Upon looking at a matchup, one tries to predict which team is more likely to win (or that they want to root for) and then proceed to bet on them, regardless of where the betting odds lie.

The second option looks not just at who is more likely to win, but directly compares the chances that the team will win versus the chances the sportsbook gives that team. A casual bettor might only have an inkling one way or the other, but because our model outputs a numerical win probability, we can put an exact amount on the discrepancy between the book and the model. For example, if Las Vegas thinks a team is a 90% favorite (moneyline of -900), but our model predicts the team only has a 75% likelihood of winning, we would take the chance and bet on the opposing team. Although betting on a team that might not win is a risky proposition, the payout is higher for an underdog and that increased profit potential justifies the strategy. If the model was accurate enough, over time, exploiting this discrepancy between betting odds and model odds could be the edge a bettor needs to drive up their portfolio value.

Figure 7.2 depicts the logic flow for these two betting options when applied to a matchup between a hypothetical team (a favorite at a -235 moneyline) and its opponent (an underdog at a +185 moneyline).

7.3 Betting Results on the Validation Set

To test our model's performance in a real-world betting environment, we applied our wagering methodologies to the 2019-20 NBA season. Since our data was trained based on a dataset from 2007-08 to 2018-19, the best way to get a true gauge of how it would perform in the wild was to throw the model at an entirely new dataset. For these fixed rate betting strategies, we place a $10 wager on every single matchup based on the betting logic as defined in Figure 7.2. There are 971 games available to bet on in our dataset so the most one could lose in this experiment is $9,710. Figure 7.3 indicates the results of all the different betting strategies we implemented.

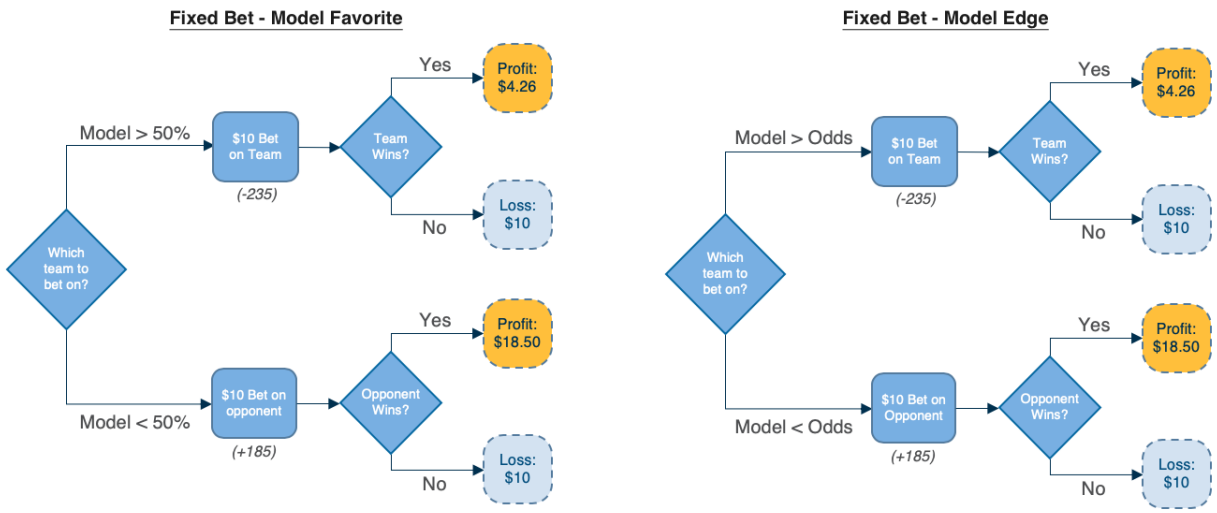

Figure 7.2: Our two fixed bet wagering implementations.

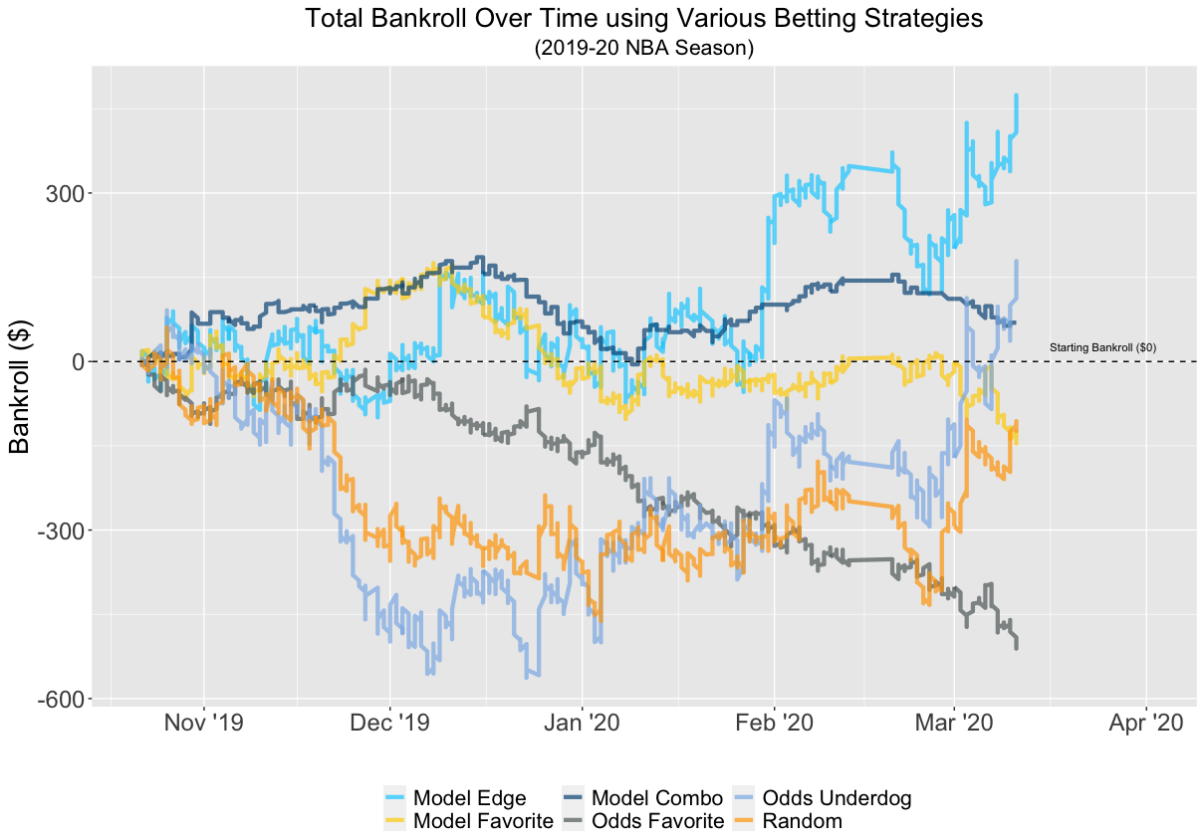

Figure 7.3: Bankroll over time for betting results on odds from the 2019-20 NBA season.

As an additional betting method, we decided to create somewhat of an ensembled approach that was a combination of the two strategies. Even if a bettor believes that a team is being undervalued in a matchup, it does not necessarily mean he should bet on that team. Let's say the sportsbook gives the team a 5% chance of winning (+1900 moneyline) and our model predicts a 25% chance. Even though this discrepancy is large, and there appears to be a considerable edge to exploit, the model still predicts that the team will lose three times out of four. So even with the large profit for picking such an underdog, a risk-averse bettor might still avoid betting on that team. As a third betting method we used a "model combo" strategy where it will only place a wager if (1) the model predicts the existence of an edge and (2) the model predicts a greater than 50% chance of winning. If there is no edge or if the predicted win probability is below 50% it will choose not to place a wager on the matchup.

In addition to comparing the three betting methodologies against each other ("model edge", "model favorite", and "model combo"), it seemed worthwhile to see how these three strategies faired against a betting approach that was devoid of any recommendations of a predictive model. We looked at three different naive betting strategies. The first, was to always take the betting line favorite in a matchup, that is, a win probability greater than 50%. We see that this betting strategy loses a considerable amount of money throughout the season. One would think that betting on the Vegas favorite (and winning about two-thirds of the time) would lead to a net profit. However, because of the structure of a moneyline, the profit from betting on a favorite can never be larger than the original risk. So each loss brings the bankroll down by an amount equal to the amount wagered, while each win can only bring the bankroll up by a smaller margin than the wager. Thus, winning over two-thirds of the games was not enough to overcome these conservative returns. We see a similar effect occurring with the "model combo" method as the gains are minimal while the losses are consistent.

The second naive strategy was the opposite of the "odds favorite" approach and we appropriately called it the "odds underdog" strategy. We bet on whichever team has the least likely chances of winning hoping that the occasional upsets would lead to some large returns. Quite surprisingly, this method yielded a positive ROI and the second-largest final

bankroll of any of the betting strategies. The 2019-20 NBA season had some high-valued upsets throughout of the season such that winning less than a third of matchups still yielded a profit.

Lastly, no proper scientific experiment would be complete without a control group. So we implemented a randomized guesser that just picks one of the two teams to win at random, regardless of what the model odds or betting odds suggest. The random model returned a negative ROI by the season's end.

In the end, our fixed betting results were extremely positive. The savviest strategy, the "model edge" method, finished the season up \$464, an ROI of almost 5%! All these profits were achieved even though it got fewer than 45% of the games correct. Next, the "model favorite" approach correctly guessed the winners at a rate almost as high as the sportsbook did (66.4% versus 67.6%). And despite being about a percent less accurate, our model ended the season with a much higher bankroll, although still negative, than if one was to blindly trust the odds' favorites. Finally, the "model combo" technique we developed proved to be extremely conservative such that it only bet on 23% of the 971 games. That said, this conservative method accurately predicted over 62% of the games it did wager on and finished the season in the black. Table 7.1 summarizes the results from all six betting methodologies discussed for the 2019-20 NBA dataset.

Table 7.1: Summary results between fixed betting methodologies.

	Win	Loss	No Bet	Win Pct	Final Bankroll	ROI	Bankroll SD*
Model Edge	430	541	-	44.3%	$464.17	4.8%	438.72
Model Favorite	645	326	-	66.4%	$-144.77	-1.5%	244.15
Model Combo	142	84	745	62.8%	$69.23	3.1%	129.20
Odds Favorite	645	309	17	67.6%	$-509.81	-5.3%	213.68
Odds Underdog	309	645	17	32.4%	$168.79	1.8%	507.96
Random	474	497	0	48.8%	$-116.14	-1.2%	292.30

*Final bankroll standard deviation calculated by running a simulation 1,000 times using a randomized game outcome based on the betting odds win probability as the 'true probability" for each game.

CHAPTER 8

The Kelly Criterion

To push our model evaluation even further and determine its most effective application, we researched more complex betting methodologies than simply wagering a fixed amount on each game. The more wagers you win, the more money you have to spend, and if you keep making smart bets the profits can compound quickly. Also we know intuitively, that not all matchups are created equal. If a bettor was very certain about the outcome of a game, he should not wager the same amount as he would on a matchup he was less confident in. Again, the goal of a robust betting model would be to identify opportunities to gain an edge over the betting line. If our model predicted a team had a 80% chance to win a game, but the sportsbook predicted the team had only a 40% chance, this would be a prime opportunity to capitalize on this probability discrepancy and bet big. Research into a type of wagering methodology that could account for optimizations based on betting confidence led straight to the famous scientific gambling strategy known as the *Kelly criterion.*

The formula was discovered in the mid-1950's by scientist J. L. Kelly, Jr. who was researching in the analysis of long-distance telephone signal noise. However, its application was then adopted by economists and horse gamblers as a way to maximize portfolio profit and diversification. Over time, and more recently, Kelly's strategy has become a part of mainstream investment theory and used by successful investors including notable billionaires such as Berkshire Hathaway's Warren Buffett and Charlie Monger and bond trader, Bill Gross. In brief, the Kelly criterion is the mathematical formula for investors and gamblers that calculates what percentage of their budget they should allocate to each investment or bet. [Kue20] By using the Kelly criterion a bettor would know exactly how much more to wager on bets that come with a high degree of confidence and the extent to avoid ones with

low confidence, thus resulting in limiting losses and maximizing gains.

Although we already had lucrative results with our fixed wagering strategy, our betting model provides a textbook use case for the application of the Kelly criterion. We have a set of implied probabilities defined by the betting odds, another set of prediction probabilities from our model, and the goal of maximizing profits based on the discrepancy between the two. As a result, for the final evaluation of our model and betting methodologies, it seemed worthwhile to take a brief tangent into applying the Kelly criterion on our 2019-20 NBA dataset to see if we could improve upon the results from Chapter 7's fixed bet strategies. But before utilizing the Kelly methods, we first need to understand its derivation.

8.1 Deriving the Kelly Criterion

To calculate the Kelly criterion, the following pieces of information are required for the formula:

- p: the true probability of an event
- q: probability the event does not occur *(q = 1 - p)*
- b: given odds of the event occurring (in the format of *"b to 1"*)

The Kelly criterion (k) is defined using those three variables in the following formula:

$$k = \frac{pb - q}{b} \tag{8.1}$$

If k is positive, you make the size of your wager equal to k percentage of your entire bankroll. If k is negative, you do not make the wager. So where does this formula come from? The answer, which makes sense in context, is derived from finance and the equations for a compounding bankroll model and compound growth rate. [Vau14]

8.1.1 The Compounding Bankroll Model

The economics behind the compounding bankroll model can be applied to any type of bankroll growth whether it's the stock market or gambling. All that is needed is some

form of binary outcomes that can be defined as a "win" versus a "loss". The equation essentially takes your initial bank amount and increases the bankroll for each win and decreases it for every loss. The degree to which it goes up or down is a function of how much you risk and how likely a win is versus a loss. See Equation 8.2.

Compounding Bankroll Model

$$A_n = A_0(1 + bk)^W(1 - k)^L \tag{8.2}$$

- A_n = amount of money (in dollars) after n bets
- A_0 = starting amount of money (initial bankroll)
- n = total number of bets, with $n = W + L$
- W = number of winning bets
- L = number of losing bets
- k = fraction of bankroll to be wagered
- b = current odds against (stated as *"b to 1"*)

The following is a brief example of how to utilize Equation 8.2. Say you have \$100 in your account (i.e. $A_0 = 100$). The given odds of winning are 2:1 ($b = 2$) and you intend to wager 10% of your bankroll on each bet (k = 10%). Find the rate of change of your bankroll.

$$A_n = A_0(1 + bk)^W(1 - k)^L$$
$$A_n = 100(1 + 2(0.1))^W(1 - 0.1)^L$$
$$A_n = 100(1.2)^W(0.9)^L$$

Therefore, if you bet 10% of your bankroll at the given 2:1 odds, with each "win" the bankroll will increase by 20% and with each "loss" it will decrease by 10%. If you made four wagers and had two wins and two losses, it would result in a final balance of about \$117 which is a 17% rate of return.

$$A_n = 100(1.2)^2(0.9)^2$$
$$A_n = 100(1.44)(0.81)$$
$$A_n = 116.64$$

8.1.2 Compounding Growth Rate

To maximize a compounding bankroll, we need to optimize the rate in which that bankroll increases. The way to calculate this rate of return is with an equation for compounding growth rate per bet. Equation 8.3 is the standard formula for compound growth rate applied to any type of investment or financial return. Usually, n is notated as t to reflect the amount of time the return in accumulating over, but for this application, we use n to indicate how many wagers the growth accumulated over.

Compounding Growth Rate

$$G = \left(\frac{A_n}{A_0}\right)^{\frac{1}{n}} - 1 \tag{8.3}$$

Again, a brief example of the application of this formula. Say a bettor made four wagers and his bankroll increased from \$100 to \$150 by the final bet. Find the growth rate over this time period.

$$A_0 = 100, A_4 = 150, n = 4$$
$$G = \left(\frac{150}{100}\right)^{\frac{1}{4}} - 1$$
$$G = (1.5)^{\frac{1}{4}} - 1$$
$$G = 1.1067 - 1$$
$$G = 10.67\%$$

So this bettor had a 50% return on investment ($\frac{150-100}{100}$) with a compound growth rate of 10.67% per bet.

8.1.3 Maximizing growth rate

Finally, to get to our goal of deriving the Kelly criterion we need to put these two concepts together and optimize for the per bet rate of return (G). This would tell us, based on all the given information about the likelihood of winning or losing a bet, the optimal fraction of

the bank to wager. To get this value we plug the compound growth rate equation into the compound bankroll model. Then, as we do with all maximizations, we take the derivative and set it equal to 0.

Optimizing for Growth Rate

Goal: seek a value for k such that it will maximize the per bet rate of return (G)
Given: G $= \left(\frac{A_n}{A_0}\right)^{\frac{1}{n}} - 1$ and $A_n = A_0(1+bk)^W(1-k)^L$

$$\begin{aligned} A_n &= A_0(1+bk)^W(1-k)^L \\ \frac{A_n}{A_0} &= (1+bk)^W(1-k)^L \\ \left(\frac{A_n}{A_0}\right)^{\frac{1}{n}} &= \left[(1+bk)^W(1-k)^L\right]^{\frac{1}{n}} \\ &= (1+bk)^{\frac{W}{n}}(1-k)^{\frac{L}{n}} \end{aligned}$$

Let p $= \frac{W}{n}$ be the true probability of winning
Let q $= \frac{L}{n}$ be the true probability of losing

$$\left(\frac{A_n}{A_0}\right)^{\frac{1}{n}} = (1+bk)^p(1-k)^q$$

Basic algebra allows us to declare that $(1+bk)^p(1-k)^q = e^{ln[(1+bk)^p(1-k)^q]}$. Therefore, $\left(\frac{A_n}{A_0}\right)^{\frac{1}{n}}$ is maximized when $ln[(1+bk)^p(1-k)^q]$ is maximized as well.

$$\begin{aligned} \text{Let } f(k) &= ln[(1+bk)^p(1-k)^q] \\ &= ln(1+bk)^p + ln(1-k)^q \\ &= p \times ln(1-bk) + q \times ln(1-k) \\ f'(k) &= p \times \frac{1}{1+bk} \times b + q \times \frac{1}{1-k} \times -1 \\ &= \frac{pb}{1+bk} + \frac{-q}{1-k} \end{aligned}$$

Then, to find our critical point...

$$
\begin{aligned}
f'(k) &= \frac{pb}{1+bk} - \frac{q}{1-k} = 0 \\
\frac{pb}{1+bk} &= \frac{q}{1-k} \\
pb(1-k) &= q(1+bk) \\
pb - pbk &= q + qbk \\
pb - q &= pbk + qbk \\
pb - q &= k(pb + qb) \\
k &= \frac{pb-q}{(pb+qb)} \\
k &= \frac{pb-q}{b(p+q)} \qquad \text{... since } p+q = 1 \\
k &= \frac{pb-q}{b}
\end{aligned}
$$

Finally, to ensure this critical point is indeed a maximum, and not a minimum, we take the second derivate and set equal to 0...

$$
\begin{aligned}
f'(k) &= \frac{pb}{1+bk} - \frac{q}{1-k} \\
f''(k) &= \frac{-pb^2}{(1+bk)^2} - \frac{q}{(1-k)^2}
\end{aligned}
$$

We know the denominator on both terms will always be positive. Also we know p and q are always positive. Therefore we have a negative term minus a positive term which will always result in a negative number. So we can conclude $f''(k) < 0$ and therefore $f'(k)$ is indeed a maximum of our function.

8.1.4 An example application of the Kelly Criterion

So the formula is now derived for the optimal amount to wager such that it maximizes the rate of return of our portfolio: $k = \frac{pb-q}{b}$. Applying this equation to an example wagering scenario is straightforward as we now have all the variables required to test this formula.

Let's say you wanted to bet on a team with -110 moneyline odds (recall that comes out to the 52.4% implied probability discussed earlier). Now say our model believes the true win

percentage for that team is 55%. Find the optimal wager amount (the Kelly criterion) as a fraction of your total bankroll.

First we need to convert the moneyline into odds (b) so that it is formatted as *"b to 1"*. Converting moneylines to odds is simple:

Moneyline to Odds Conversion - General Equation

$$\text{Odds} = \begin{cases} \dfrac{ML}{100}, & \text{if ML} \geq 0 \\ \\ \dfrac{100}{-ML}, & \text{if ML} < 0 \end{cases} \tag{8.4}$$

So for our example problem we have:

- p: the true probability of an event = 55%
- q: probability the event does not occur = 45%
- b: given odds of the event occurring = $\frac{100}{110} = .909$

Input these variables into the Kelly criterion formula and we get...

$$\begin{aligned} k &= \frac{pb - q}{b} \\ k &= \frac{0.55 \times 0.909 - 0.45}{0.909} \\ k &= \frac{0.04995}{0.909} \\ k &= 0.05495 \end{aligned}$$

Therefore, the optimal wager would be to place about 5.5% of your bankroll on this bet to ensure the maximum long term growth rate. We can even see what the expected compound bankroll growth would be if, for example, we wagered 20 times, winning 11 and losing 9.

$$\begin{aligned} A_n &= 100(1 + bk)^W (1 - K)^L \\ &= 100(1 + (0.909)(0.05495))^{11}(1 - 0.05495)^9 \\ &= 100(1.050)^{11}(.945)^9 \\ &= 102.79 \end{aligned}$$

The portfolio went from a starting bankroll of $100 to $102.79 throughout the 20 bets. Even though this growth rate seems small at 2.79%, we achieved this growth winning only one more game than we lost. A consistent growth rate of almost 3% from a win percentage of just 55% percent could lead to hefty returns throughout a large sample size of bets.

One final note on the definition of the Kelly criteria. We could rewrite the formula as such:

$$k = \frac{pb - q}{b}$$
$$k = \frac{p(b) - q(1)}{b}$$

The numerator *p(b) - q(1)* can be translated to simply the "expected value on a $1 bet." This expression breaks down as follows: the probability of winning, multiplied by the betting odds, minus the probability of losing, multiplied by the betting risk. We've been referring to the concept of trying to find *the betting edge* throughout this research. So formally, within the gambling sphere, this edge is defined as the expected value on a $1 bet. Essentially, over time, how much do you expect to profit (or lose if it equates to a negative value) from this wager. Thusly, one could also refer to the formula for the Kelly criterion as:

$$k = \frac{\textit{expected value on a \$1 bet}}{\textit{odds}} = \frac{\textit{edge}}{\textit{odds}}$$

8.2 Applying the Kelly Strategy to our Data

The implementation of the Kelly criterion to our dataset was similar to the "model edge" fixed betting strategy conducted in Chapter 7 as they both involved identifying the discrepancy between the model's predicted win probability and the implied probability from the betting odds. The logic flow to utilize the Kelly criterion is the same as the "Fixed Bet - Model Edge" defined in Figure 7.2, but instead of placing a $10 stake on either the team or the opponent, that wager would be variable.

First, we had to define an initial bankroll amount, which we set at $10,000. We chose this quantity because it would provide a comparable investment to how much was wagered in the

fixed betting method ($10 per game x 971 games = $9,710). Then, for every matchup, we compared the model probability to the team's win probability as defined by its moneyline. If the model probability was higher than the betting odds, we would wager on that team, it it was worse, then we would wager on the opponent. Next was the calculation of how much to wager based on the Kelly criteria. If the model recommended betting on the team, we'd plug the model win probability in for p, the model's lose probability in for q, and the moneyline converted betting odds in for b. This gave us the value, k, which defined the fraction of the present bankroll to wager. If the model probability was worse than the betting probability for the team, then we'd just apply the same technique, but instead do so to the opponent's variables. Again using the same 971 games from the 2019-20 NBA season, we can see how the initial $10,000 investment fluctuated over the series of bets.

One of the common critiques of the Kelly wagering strategy is that it is oftentimes too aggressive. Although mathematically the derivation does suggest that the technique, over a large enough sample size, does trend toward an optimal expected value. The issue, however, is that the model is so heavily reliant on the values inputted as the "true probability" (p). So the strategy can only be as effective as the model is accurate. Additionally, Kelly-based strategies are extremely high variance. If the identified edge is very large, the Kelly criterion will recommend almost the entire bankroll. This is a great strategy to rapidly compound your portfolio if the prediction comes to be true. But an unforeseen circumstance leading to an instance of bad fortune can cripple the bankroll with one large failed wager. And because the Kelly wagering method is a function of your total bankroll, once that account is extremely low, it takes exponentially longer to drive it back upwards.

As a result, remedies have been developed to counteract the high variance troubles associated with such an aggressive betting strategy. One of the recommendations to smooth out the wild oscillations common to Kelly-based wagering is to use the concept of "fractional Kellys". [Tho97] For example, if the formula outputted the optimal wager for a certain matchup was 40% of the present bankroll, a bettor might be wary about such an aggressive bet. So instead, a fractional Kelly would recommend betting half of the formula's result or just a quarter of it (20% of the bankroll or 10%, respectively). Wagering with a fractional

Kelly still allows for the bettor to increase or decrease their wager amount systematically based on their confidence in the existence of a betting edge, but doing so with a more tempered approach. Apparently, in the betting community, bettors who implement the Kelly criterion as part of their strategy rarely wager more aggressively than a quarter Kelly. Figure 8.1 illustrates the results from using four variations of the Kelly criteria on our validation data set from the 2019-20 NBA season. Ranging from most aggressive to most conservative, we utilized a "Full Kelly", "Quarter Kelly", "5th Kelly", and "8th Kelly".

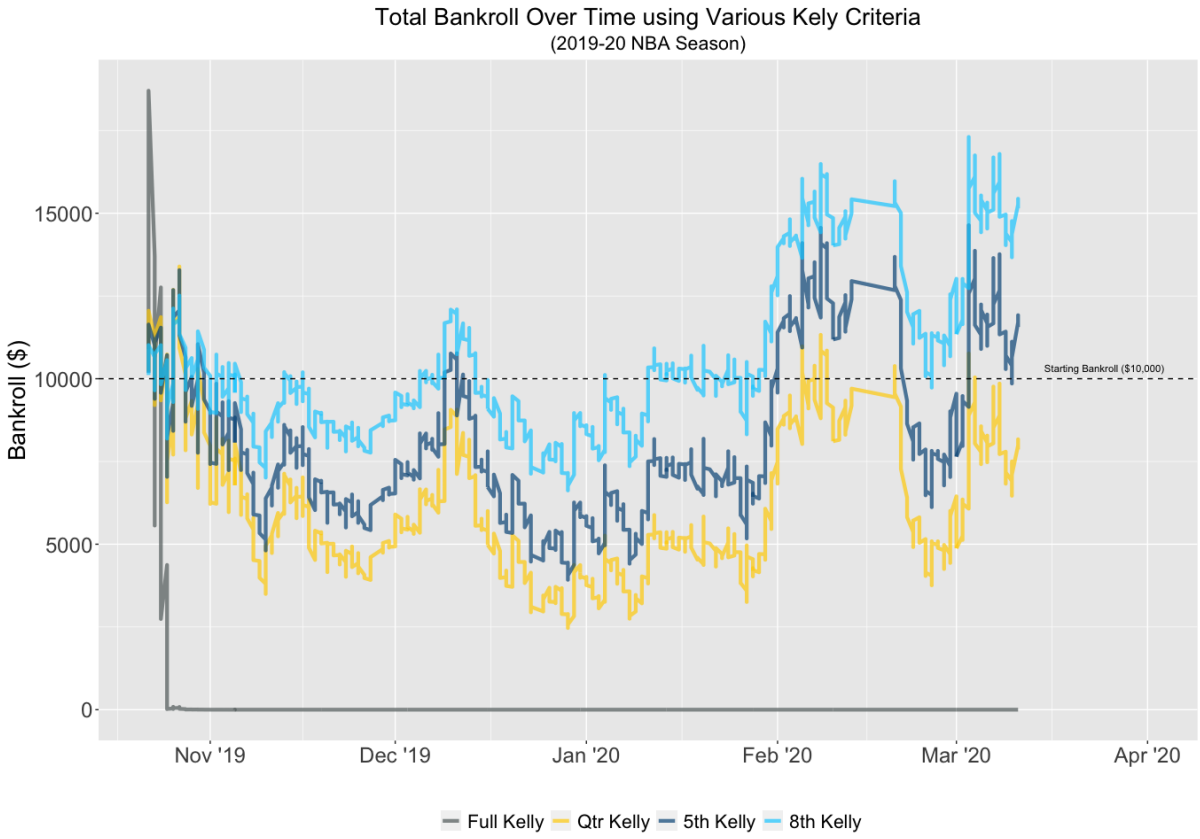

Figure 8.1: Bankroll over time using various Kelly criteria for the 2019-20 NBA season.

There were quite drastic differences between the Kelly criteria we implemented. Right out of the gate, the full Kelly strategy almost doubled the initial bankroll with just a single wager. But just as quickly is at skyrocketed, it came crashing down and ended up plateauing the bankroll to \$0 before the first month of the season was complete. The less aggressive fractional Kellys fluctuated with the same ebbs and flows which was to be expected since they were operating off the same model probabilities. We see in Figure 8.1 that compared to

our starting $10,000 bankroll, the more conservative approaches finished with a net positive gain, while the two more aggressive methods finished with a net negative.

We also see that all betting strategies had the same win percentage as the "model edge" method from Chapter 7, since they were applying the same betting logic (just with different wager amounts). We were quite pleased with the "model edge" method yielding a solid 5% rate of return. However, using the same predictions, the "5th Kelly" and "8th Kelly" finished the season with an 15.6% ROI and 51.5% ROI, respectively! Even more interesting, the "Qtr Kelly" method ended up decreasing the bankroll by 21.3% while the "Full Kelly" hit rock bottom. In this instance, just the slightest downtick in aggressiveness, from 1/4 to 1/5, resulted in an over 35% swing in ROI and was the difference between making money and losing it. A summary of the four Kelly criteria is in Table 8.1.

Table 8.1: Summary results between our fractional Kelly criteria.

	Win	Loss	No Bet	Win Pct	Final Bankroll	ROI	Bankroll SD*
Full Kelly	430	541	-	44.3%	$0.00	-100%	<0.01
Qtr Kelly	430	541	-	44.3%	$7,865.77	-21.3%	5.557.94
5th Kelly	430	541	-	44.3%	$11,565.00	15.6%	5,662.36
8th Kelly	430	541	-	44.3%	$15,150.70	51.5%	4,897.09

*Final bankroll standard deviation calculated by running a simulation 1,000 times using a randomized game outcome based on the betting odds win probability as the 'true probability" for each game.

With all the successes from the model edge strategy as well as the 5th Kelly and 8th Kelly, we could have stopped there. But we were greedy and because we were trying to be as "realistic" as possible with our betting experiment, there was one last tweak to our model that was worth pursuing. We referenced in Section 4.3 that, historically, one needs about eight games worth of data to have a sufficient sample size to gauge the quality of the team. And since there is no requirement for us to bet on every single game of the season, from the very first to the very last, we decided as one last approach, to be patient before making our first wager. Since we were using the year-to-date aggregation methodology, and this

technique did not roll back to the previous season, any wagers placed in the first few games of the year were operating with a less than ideal sample size. Thus, it would be prudent to let all teams play at least eight games before we placed our first wager.

And the results were phenomenal. Besides the full Kelly strategy which again went to zero, the other three methods each yielded extremely high returns. Quarter Kelly gave us a 76.4% return, 5th Kelly gave us 98.2%, while 8th Kelly performed second best with a 91.9% ROI. Just by eliminating the first few games of the season and waiting until we had sufficient data, our best results were able to generate over $4,000 more than the best results from the full-season experiment and almost doubled our initial bankroll! Apparently patience is not just a virtue, but also a lucrative investment strategy.

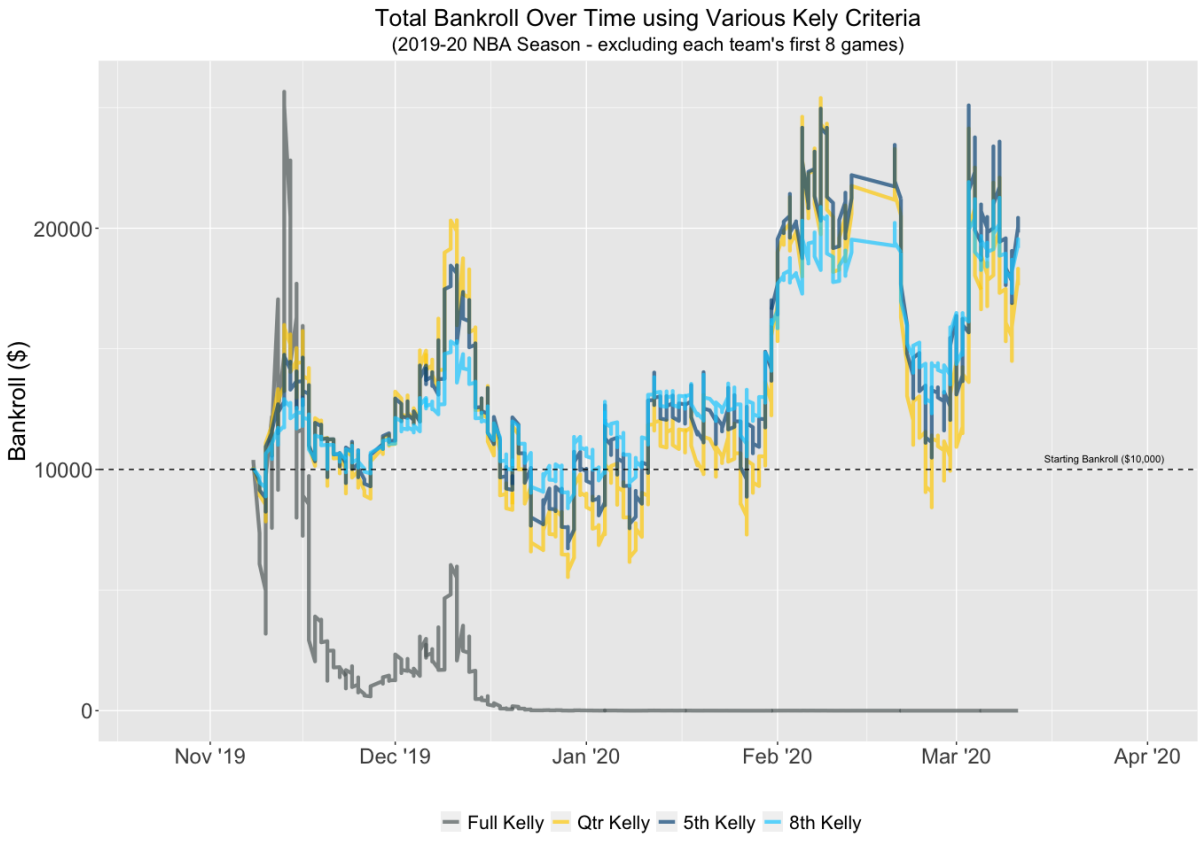

Figure 8.2: Bankroll over time using various Kelly criteria after excluding each team's first eight games.

Evidently Warren Buffet is onto something with this Kelly criterion. But maybe the Oracle of Omaha isn't so much divinely prescient as he is statistically rational. In theory, this formula has the potential for rapidly compounding dividends if the model was particularly performant. But it is also very fickle. In practice, the Kelly method is only as powerful as the quality of its parameters. What we learned is that these high-variance strategies should be reserved for opportunities with a more robustly built and reliable model to use as the "true probability" input into the Kelly criteria formula. And when we used a Kelly criterion larger than one-quarter, we found ourselves into the red quite quickly. But by taking a more conservative approach, with smaller fractional Kellys, our "true probability" did not have to be full-proof, and provided enough margin of error to yield some impressive returns on investment. ROIs that were significantly higher than what resulted from the fixed betting technique used in Chapter 7.

Table 8.2: Fractional Kelly results after excluding each team's first eight games.

	Win	Loss	No Bet	Win Pct	Final Bankroll	ROI	Bankroll SD*
Full Kelly	375	471	-	44.3%	$0.02	-100%	2.96
Qtr Kelly	375	471	-	44.3%	$17,369.64	76.4%	7,067.40
5th Kelly	375	471	-	44.3%	$19,824.89	98.2%	6,301.83
8th Kelly	375	471	-	44.3%	$19,186.95	91.9%	4,755.52

*Final bankroll standard deviation calculated by running a simulation 1,000 times using a randomized game outcome based on the betting odds win probability as the 'true probability" for each game.

CHAPTER 9

Conclusion

9.1 Further Research

Despite the successes of this research, along the way there were many avenues for improvements that would be necessary before offering this tool to bettors or truly trusting the model's predictions with any personal financial investment.

As mentioned earlier in the study, many of the variables in our dataset were highly correlated. There is an entire genre of advanced "Sabermetric-like" stats that could provide insights beyond what's available in the box score. Particularly cutting edge right now in the NBA are tracking metrics that use cameras in every arena to trace the movement of the ball and all ten players on the court at all times. These stats can tell you not only whether a shot went in the basket, but also how far the player was from the hoop, how close the defender was from the shooter, whether he took any dribbles before taking the shot, and more. Tracking metrics are the next wave of basketball analytics and could provide major improvements upon the basic box score stats. Additionally, a crucial matchup variable that was not included in our model was the strength of the schedule. Throughout an entire season, the difficulty of a team's opponents usually evens out compared to all the other teams. But specifically with our eight-game span aggregation method, the quality of their opponent in those eight games is critical to the evaluation. If a team was coming off a five game road trip against playoff-caliber teams, they would likely show a dip in their performance metrics that was directly correlated to who they were playing, not necessarily how good they are.

On the topic of improving our data aggregation inputs, an analysis of our two methods—eight-game spans and year-to-date—could very well warrant a research project of its own.

Across the board, in all three modeling types plus the neural networks, the accuracy metrics in both the test set and training set were higher using the year-to-date aggregation data. Clearly sample size plays a role in improving the predictive power, as once we reached the ninth game of the year or later, the year-to-date method had more data to work with. But what is particularly interesting is not just that the year-to-date method was better, but the fact that it was only *marginally* better. The fact that we were able to generate 60-65% accuracy using just the prior eight games of data is noteworthy and provides validity to the belief that much of the variance of an entire season can be derived from just an eight-game sample. A compelling future study would be to dive deeper into these "X-game spans" and see where the optimal point lies between sample size and high predictability. Maybe you get the same predictive power at only five games? Maybe at 10 games, performance is even better than year-to-date? A highly performant model that only needs a couple weeks of NBA data as inputs would be quite the impressive tool.

A rather glaring omission from our dataset, particularly for a star-driven league like the NBA, was the inclusion of individual player performance. In researching existing NBA win likelihood models in the analytics community, a majority of them are player-based models. Instead of training on team data as a whole, these player systems will treat each player as a model and project an expected playing time and subsequent production level that get combined for each team to determine the likely outcome of the game. Players like Steph Curry and LeBron James are so influential on their team's success that their absence, injury, or playing at less than full capacity would drastically change the expected outcome. Our model was blind to this type of player variance.

Lastly, there is a world of betting strategies that could be factored into a data-driven wagering system. One of the challenges of working with moneylines, like we were in our study, is that even if you believe a team is undervalued, they still have to win the game to realize any profit. If Vegas sets a team at a 5% chance of winning, and our model believes the true probability is closer to 25%, we would still expect to lose money betting on that team three out of four times. Wagering on spreads, on the other hand, allow a bettor to identify an edge in a matchup, but then only need the team to overcome expectations, keep

the game close, and cover the point-spread to win the bet. Of course, it would no longer suffice to have a model with a binary predictor of "win" versus "loss;" you would now have to output a score for each team, which is an entirely different challenge.

Beyond the type of bet to wager on, there are other key indicators that professional sports gamblers are acutely tuned in to. We did not incorporate the actual betting odds into our model, but in a real-world situation this would be an available data point that a bettor would consider in their betting system. Beginning any model with the actual betting line would ground the bet immediately with a high importance variable since we know sportsbooks are good at what they do. If one's model suggests a team has an 80% chance to win, but the betting line says 15%, then most likely, there is a critical piece of intel that the sportsbook line setters are aware of that drastically affected the win likelihood. Secondly, as discussed previously, a casino's goal is not necessarily to predict the outcome of a game, but instead, to place the line such that an even amount of people bet both sides. Generally, this tends to a fair market value but historically we know inherent biases in bettors leads them to bet the favorites, expect high scoring games, and root for the well-known teams, players, and larger markets. Expert bettors (known as "sharps"), will have access to, not just the sportsbook odds of a matchup, but also the number of bets on each side as well as the total amount of money bet on each side. The combined forces of the odds, bet volume, and a machine learning model provide everything a sharp would need to consistently beat the betting market, just as a day trader or hedge fund manager would with the stock market. (Some insider trading in the likes of having a source that is close to the coaching and training staffs helps too.)

9.2 Final Thoughts

On the whole, the results from this study were overwhelmingly positive. The pursuit of an edge over the gambling market odds provided a perfect backdrop and motivation for research. The available tools required to compile a large enough dataset of betting odds and NBA statistics were open source, yet thorough. The output from our betting models resulted

in a relatively accurate win probability prognosticator which was seamlessly implemented into a profitable, systematic wagering strategy for the 2019-20 NBA season.

The mission was ambitious, but in the end, the simplest of all the modeling techniques proved to be powerful enough to accurately predict over 65% of games in a season's worth of data that it had never seen before. And when applying this model to our betting simulation, we realized nearly a 2x return on investment, in the best case scenario. Be it the elegance of the logistic regression or the brute force of a neural network, predicting the outcome of an NBA game was not just a compelling task, but an achievable one.

As the aftereffects of COVID-19 hopefully fade away and live sports trickle back into our society in a much needed return to normalcy, the betting markets will rise from the ashes as well. We can take comfort knowing, that although formidable, these sportsbooks that live behind the walls of casinos fortifying the Las Vegas strip, and now cities all across the US, are not impenetrable. When one places a wager it may seem that your opponent is the team on the other side of the field or court. In reality, though, the real foe is the house. This study suggests that sports betting might not be "gambling" at all, but truly a game of skill. And with enough data access and modeling proficiency it is, in fact, quite possible to "beat the book."

CHAPTER 10

Appendix

Table 10.1: Record for matchup favorites based on the sportsbook betting odds.

Season	Games	Win	Loss	Even Odds	Win Pct
2007-08	1229	846	363	20	0.700
2008-09	1230	859	358	13	0.706
2009-10	1230	863	365	2	0.703
2010-11	1230	842	383	5	0.687
2011-12	990	677	311	2	0.685
2012-13	1229	841	380	8	0.689
2013-14	1230	841	378	11	0.690
2014-15	1230	863	364	3	0.703
2015-16	1230	851	367	12	0.699
2016-17	1230	810	403	17	0.668
2017-18	1230	837	383	10	0.686
2018-19	1230	818	394	18	0.675
2019-20	971	645	309	17	0.676

Figure 10.1: Distribution of moneylines for both teams by season from 2007-2020.

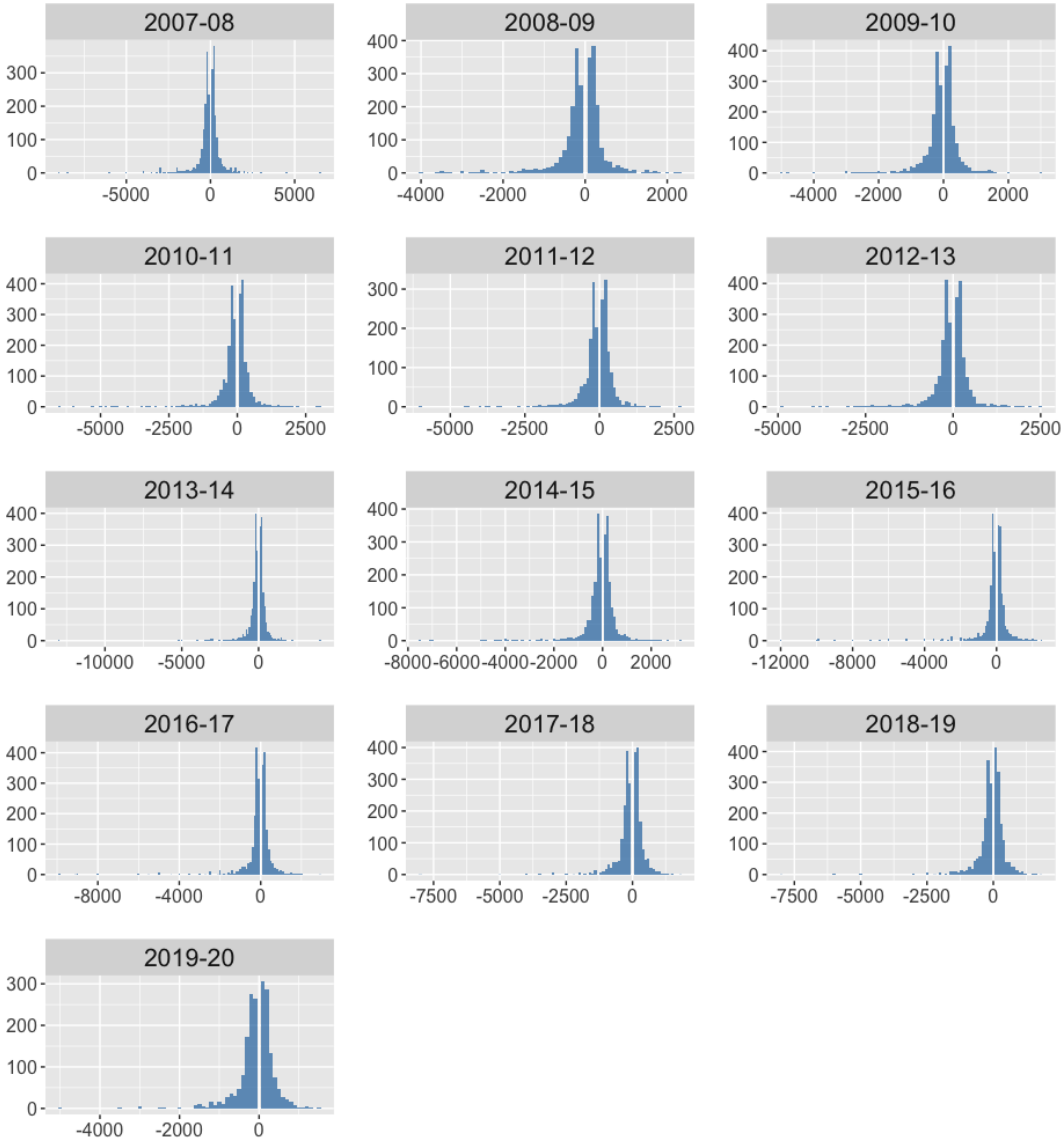

Table 10.2: League average stats per game by season.

season	pace	fga	fgpct	fg3a	fg3pct	fg2a	fg2pct	fta	ftpct	ppg	dreb	oreb	ast	tov	stl	blk	pf	ortg
2007-08	93.5	81.5	0.457	18.1	0.362	63.4	0.484	24.9	0.755	99.9	30.8	11.2	21.8	14.1	7.3	4.7	21.0	106.2
2008-09	92.8	80.9	0.459	18.1	0.367	62.8	0.485	24.7	0.771	100.0	30.3	11.0	21.0	14.0	7.3	4.8	21.0	107.0
2009-10	93.6	81.7	0.461	18.1	0.355	63.6	0.492	24.5	0.759	100.4	30.8	11.0	21.2	14.2	7.2	4.9	20.9	106.6
2010-11	92.9	81.2	0.459	18.0	0.358	63.2	0.487	24.4	0.763	99.6	30.5	10.9	21.5	14.3	7.3	4.9	20.7	106.3
2011-12	92.3	81.4	0.448	18.4	0.349	63.0	0.477	22.5	0.752	96.3	30.8	11.4	21.0	14.6	7.7	5.1	19.6	103.5
2012-13	93.0	82.0	0.453	19.9	0.359	62.0	0.483	22.2	0.753	98.1	30.9	11.2	22.1	14.5	7.8	5.1	19.8	104.8
2013-14	94.8	83.0	0.454	21.5	0.360	61.5	0.488	23.6	0.756	101.0	31.8	10.9	22.0	14.6	7.7	4.7	20.7	105.7
2014-15	94.7	83.6	0.449	22.4	0.350	61.2	0.485	22.8	0.750	100.0	32.4	10.9	22.0	14.4	7.7	4.8	20.2	104.7
2015-16	96.6	84.6	0.452	24.1	0.354	60.5	0.491	23.4	0.757	102.7	33.3	10.4	22.3	14.4	7.8	5.0	20.3	105.6
2016-17	97.0	85.4	0.457	27.0	0.358	58.4	0.503	23.1	0.772	105.6	33.4	10.1	22.6	14.0	7.7	4.7	19.9	108.2
2017-18	98.0	86.1	0.460	29.0	0.362	57.1	0.510	21.7	0.767	106.3	33.8	9.7	23.2	14.3	7.7	4.8	19.9	107.8
2018-19	100.7	89.2	0.461	32.0	0.355	57.2	0.520	23.1	0.766	111.2	34.8	10.3	24.6	14.1	7.6	5.0	20.9	109.7
2019-20	100.7	88.8	0.460	33.9	0.357	54.9	0.523	22.9	0.771	111.4	34.7	10.1	24.3	14.5	7.7	4.9	20.6	109.9

Table 10.3: Final logistic model from the year-to-date dataset used for betting implementation.

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.8167  -1.0183  -0.1463   1.0149   2.7019

Coefficients:
                               Estimate Std. Error z value             Pr(>|z|)
(Intercept)                   -0.664874   1.159278  -0.574             0.566290
locationGame.team1.H          -0.896211   0.042425 -21.125 < 0.0000000000000002 ***
net_rtg.team2                  0.077939   0.013344   5.841       0.00000000519 ***
net_rtg.team1                 -0.037474   0.013882  -2.700             0.006944 **
winpct.team1                  -1.170826   0.292961  -3.997       0.00006428004 ***
winpct.team2                   0.807796   0.284718   2.837             0.004551 **
ptsTeam.own.agg.per100.team1 -0.025395   0.013572  -1.871             0.061314 .
ptsTeam.own.agg.per100.team2  0.006667   0.012497   0.534             0.593668
stlTeam.own.agg.per100.team1 -0.066672   0.021957  -3.037             0.002393 **
blkTeam.own.agg.per100.team2  0.026313   0.024454   1.076             0.281905
fgpct.own.agg.team2           -2.127449   2.769143  -0.768             0.442327
fgpct.own.agg.team1            2.451518   2.760752   0.888             0.374547
fg2pct.own.agg.team2           5.463032   1.750985   3.120             0.001809 **
fg2pct.own.agg.team1          -1.790129   1.751866  -1.022             0.306856
fg2pct.opp.agg.team2          -7.814259   2.185005  -3.576             0.000348 ***
fgmTeam.own.agg.per100.team1 -0.004568   0.022912  -0.199             0.841967
astTeam.opp.agg.per100.team2 -0.051764   0.014653  -3.533             0.000411 ***
fgmTeam.own.agg.per100.team2 -0.048245   0.022915  -2.105             0.035253 *
fgpct.opp.agg.team1           -4.496422   3.229357  -1.392             0.163814
fg2pct.opp.agg.team1           8.613108   2.177498   3.956       0.00007637280 ***
fgpct.opp.agg.team2           14.625446   3.092135   4.730       0.00000224647 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 15076  on 10874  degrees of freedom
Residual deviance: 13195  on 10854  degrees of freedom
AIC: 13237
```

REFERENCES

[AP19] AP. "Sports betting market expected to reach $8 billion by 2025." *MarketWatch - Associated Press*, November 2019.

[Boa20] Nevada Gaming Control Board. "Monthly Revenue Report." Technical report, February 2020.

[Bre20] Alex Bresler. *nbastatR: R's interface to NBA data*, 2020. R package version 0.1.1503.

[Chr96] Nicholas Christenfeld. "What makes a good sport?" *Nature*, October 1996.

[Cul18] John Culver. "Why 52.4% is the most important percentage in sports gambling." *Medium*, October 2018.

[Dim20] Sports Betting Dime. "Beating the Juice: Removing the Vig from Sports Betting Lines." *Sports Betting Dime*, February 2020.

[Dot20] Guy Dotan. "UCLA MAS Thesis - 2020.", 2020. https://github.com/guy-dotan/uclathesis.

[ESP20] ESPN. "NBA suspends season until futher notice after player tests positive for the coronavirus." March 2020.

[Gal17] Gallup. "Sports." 2017. https://news.gallup.com/poll/4735/sports.aspx.

[Gle19] Stephanie Glen. "Decision Tree vs Random Forest vs Gradient Boosting Machines: Explained Simply." *Data Science Central*, 2019.

[Gol13] Ben Golliver. "NBA cancels game between Celtics and Pacers after Boston Marathon blasts." *Sports Illustrated*, April 2013.

[Her18] Devin Heroux. "Curling may finally be having its Moneyball moment." *ESPN*, April 2018.

[Jos19] Maithili Joshi. "A Comparison between Linear and Logistic Regression." *Medium*, April 2019.

[Kue20] Justin Kuepper. "Using the Kelly Criterion for Asset Allocation and Money Management." *Investopedia*, March 2020.

[Leg20] LegalSportsBetting.com. "Legal Sports Betting Revenue Tracker." March 2020.

[LLP18] Infoholic Research LLP. "Sports Analytics Market Forecasts up to 2024." Technical report, November 2018.

[Mar18] Brian Martin. "NBA.com/Stats Switches To True Possessions: What It Means and Why It's Important." *NBA.com*, October 2018.

[Rod20] Ryan Rodenberg. "United States of sports betting: An updated map of where every state stands." *Canadian Broadcasting Corporation*, May 2020.

[Shu19] Lavanya Shukla. "Designing Your Neural Networks." *Towards Data Science*, September 2019. https://towardsdatascience.com/designing-your-neural-networks-a5e4617027ed.

[Tho97] Edward O. Thorp. "The Kelly Criterion in Blackjack Sports Betting and the Stock Market." *Finding the Edge: Mathematical Analysis of Casino Games*, 1997.

[Vau14] Christopher Vaughen. "Mathematics of Gambling: the Kelly Formula.", 2014.