



Channel Islands

CALIFORNIA STATE UNIVERSITY

**The Importance of NBA Box Score Statistics
and the Value of Statistical Outbursts**

A Thesis Presented to

The Faculty of the Computer Science Department

In (Partial) Fulfillment

of the Requirements for the Degree

Masters of Science in Computer Science

by

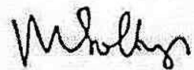
Student Name:
Jack BJ BENSON

Advisor:
Dr. Michael SOLTYS

November 2019

© 2019
Jack BJ Bension
ALL RIGHTS RESERVED

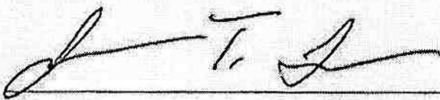
APPROVED FOR MS IN COMPUTER SCIENCE



Dec 10, 2019

Advisor: Dr. Michael Soltys

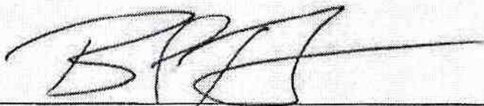
Date



Dec 10, 2019

Dr. Jason Isaacs

Date

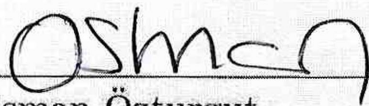


12-10-19

Dr. Brian Thoms

Date

APPROVED FOR THE UNIVERSITY



12/12/19

Dr. Osman Özturgut

Date

Non-Exclusive Distribution License

In order for California State University Channel Islands (CSUCI) to reproduce, translate and distribute your submission worldwide through the CSUCI Institutional Repository, your agreement to the following terms is necessary. The author(s) retain any copyright currently on the item as well as the ability to submit the item to publishers or other repositories.

By signing and submitting this license, you (the author(s) or copyright owner) grants to CSUCI the nonexclusive right to reproduce, translate (as defined below), and/or distribute your submission (including the abstract) worldwide in print and electronic format and in any medium, including but not limited to audio or video.

You agree that CSUCI may, without changing the content, translate the submission to any medium or format for the purpose of preservation.

You also agree that CSUCI may keep more than one copy of this submission for purposes of security, backup and preservation.

You represent that the submission is your original work, and that you have the right to grant the rights contained in this license. You also represent that your submission does not, to the best of your knowledge, infringe upon anyone's copyright. You also represent and warrant that the submission contains no libelous or other unlawful matter and makes no improper invasion of the privacy of any other person.

If the submission contains material for which you do not hold copyright, you represent that you have obtained the unrestricted permission of the copyright owner to grant CSUCI the rights required by this license, and that such third party owned material is clearly identified and acknowledged within the text or content of the submission. You take full responsibility to obtain permission to use any material that is not your own. This permission must be granted to you before you sign this form.

IF THE SUBMISSION IS BASED UPON WORK THAT HAS BEEN SPONSORED OR SUPPORTED BY AN AGENCY OR ORGANIZATION OTHER THAN CSUCI, YOU REPRESENT THAT YOU HAVE FULFILLED ANY RIGHT OF REVIEW OR OTHER OBLIGATIONS REQUIRED BY SUCH CONTRACT OR AGREEMENT.

The CSUCI Institutional Repository will clearly identify your name(s) as the author(s) or owner(s) of the submission, and will not make any alteration, other than as allowed by this license, to your submission.

The Importance of NBA Box score Statistics and the
Title of Item Value of Statistical Outbursts
Machine Learning, NBA Statistics, Decision Tree Classifier
3 to 5 keywords or phrases to describe the item

Jack B. Benson
Author(s) Name (Print)

[Signature]
Author(s) Signature

12/13/19
Date

The Importance of NBA Box Score Statistics and the Value of Statistical Outbursts

Jack BJ Bension

December 9, 2019

Abstract

The Nation Basketball Association (NBA) has embraced the 21st Century by increasing its use of advanced analytics. New and evolving statistics can be used to determine how efficient a player is while he is on the court. However, even though a player is being efficient, his performance may not lead to victories. This paper creates a Decision Tree Classifier model that helps to determine, through game by game statistics, an NBA player's value to a team's chance to win. Players tested in the model demonstrate that having a high *PER* does not always lead to being a great asset for their team. The models created also distinguish what statistics are important for All-Star and Starter level players. The All-Star model favors individually focused, offensive statistics; whereas the Starter model places a higher level of importance on team statistics.

Contents

1	Introduction	1
1.1	Motivation	2
2	Background	3
2.1	History of NBA Analytics	3
2.2	Applications of Advanced Analytics	7
2.2.1	General Managers	8
2.2.2	Coaches	8
2.2.3	Scouts	9
2.3	New Statistics	11
2.4	What is Machine Learning?	24
2.4.1	Different Types of Machine Learning	24
2.4.2	Regression vs Classification Models	27
2.5	Python	32
2.5.1	<i>pandas</i> API	33
2.5.2	<i>NumPy</i> API	35
2.5.3	<i>scikit-learn</i> API	36
3	Contribution	39
3.1	Statistics Analyzed	39
3.2	Creating Models	40
3.3	Gathering and Organizing the Data	40
3.4	Training the Models	42
4	Experiments and Justification	43
4.1	Finding Important Variables	43
4.2	Creating New Models	45
4.3	The Lack of Defense	47
4.3.1	DRtg	47
4.3.2	The Value of a Block	48
4.4	Player Tests	49
4.4.1	Difference in Play Style	51
4.5	Salary vs Data	52
5	Conclusion and Future Work	56

List of Figures

1	This chart demonstrates one of the effects that the 2004-2005 Phoenix Suns had on the NBA. Between their performance and the increased emphasis on analytics, the NBA has seen a dramatic increase in the use of the 3 point shot [19].	5
2	A sample provided by Synergy that lists Kobe Bryant's scoring tendencies [19].	6
3	A sample of Second Spectrum's software. A green bar indicates that the player is wide open and is not currently being contested by a defender. An orange bar indicates that the player is being contested by an opponent but still has room to score. Red indicates that the player is fully guarded and has little to no room to shoot the ball. The percentage over each player shows their accuracy from that range based on data gathered on the player [17].	7
4	A sample of different reports that are produced by Second Spectrum's software [17].	10
5	A graphical representation of clustering. Each circle of data points represents a group of records that have similar traits [14].	26
6	A graphical representation of a supervised learning model [14].	27
7	An example of a linear regression model plot. The predictor variables are represented by x and the response variables are represented by y [14].	29
8	An example of the data splitting process that occurs in a decision tree classifier. Each colored section represents a different category [14].	30
9	A plot of a decision tree. The rules and logic created by the Decision Tree Classifier can be clearly viewed through this image [2].	31
10	The documentation on the <code>read_csv</code> procedure with basic parameter inputs [18].	34
11	An example of <code>concat</code> and its ability to merge different <code>DataFrame</code> structures together [18].	35
12	The <code>isfinite</code> function is a tool that <code>NumPy</code> provides. It is used to identify null data and eliminate it from the <code>DataFrame</code> structures [11].	36

13	The definitions of the procedures and the attribute that will be used to analyze the models [2].	37
14	A section of the code in this study. The <i>fit</i> functions is used to train the model. The <i>score</i> function is then used to determine the accuracy of the test data [2].	38
15	A page of player’s data, game by game, for a whole season. The <i>www.basketball-reference.com</i> website provides several different ways to export the data. This eliminates the need to create a web scrapper for the website [1].	41
16	This table shows the value of importance for all of the tested statistical categories. The results for both the “All-Star” and “Starter” tree are shown. The statistics highlighted in yellow are the most important statistics for Starters, the statistics highlighted in blue are the most important statistics for All-Stars, and the statistics highlighted in green are the shared statistics that are important for both Starters and All-Stars. .	44
17	This table shows the value of importance for all of the selected statistical categories. The categories were chosen based on the top categories in Figure 16. The results for both the ”All-Star” and ”Starter” trees are shown.	46
18	This table shows the fit for players tested against both models. Along with the fit, the average <i>PER</i> for the players over the seasons in which they were tested is listed. Based on the players that were tested, the average All-Star fit was .5378 and the average Starter fit was .5302.	50
19	This table shows the fit for players tested against both models versus the salary the players had during the 2017-2018 season.	53
20	This table shows the <i>PER</i> for players tested against both models versus the salary the players had during the 2017-2018 season.	54

1 Introduction

The NBA is becoming a data driven league. It is no longer a league in which basic statistics can summarize a player's ability to help his team win. Advanced analytics are providing coaches and GMs (general managers) deeper insight into which players are the most efficient on the court. The more efficient a player is on the court, the chance that his team wins increases.

One advanced statistic that is used in determining a player's effectiveness is *PER* (Player Efficiency Rating). *PER* is one of the main statistics that coaches and GMs use to determine the value of a player. However, this statistic may not relate to the player's ability to help their team win. For example, Nikola Vucevic has, at the time this paper is being written, played 59 games in the 2018-2019 season and was selected as an All-Star. He has a *PER* of 25.79, which places him in the top 10 of all the players in the NBA. His team, however, is below .500 in win percentage this year and is not currently in the playoff race. Even though he is being extremely efficient in his play, his high statistical performances do not necessarily increase his team's chance of winning. Is he just putting up big numbers on a bad team? Can certain statistics determine if the player is performing at an All-Star level?

1.1 Motivation

The following research examines a player's statistics game by game in order to determine a player's value. A Decision Tree Classifier machine learning model, in conjunction with Python, is used to explore which statistics separate an All-Star level player from a Starter level player in the NBA. This model will determine whether or not a player's statistical outburst leads to a greater chance for their team to win or if a player is "padding" his statistics in losing efforts. With the information provided by the model, coaches and GMs can determine if a player is worth giving "All-Star" level money to or if the player is simply exceeding on a bad team.

2 Background

2.1 History of NBA Analytics

One of the first models created to help coaches better understand the effectiveness of basketball players was developed by Stanford University professor emeritus and author of *The Art of Computer Programming* Donald Knuth. In 1959, Knuth created “The Electronic Coach”, a program that analyzes game statistics and produces a rating for each player [6]. At this time, the NBA was tracking a minimal amount of statistics, which included points, rebounds, assists, field goals, and free throws [19]. Knuth’s program helped to increase his high school basketball team’s wins from 1 to 14 the next season. Further analysis of a player’s value to a team’s win probability was constructed by Eldon G. Mills and Harlan D. Mills in 1970. This study focused on the Major League Baseball player statistic Player Win Average (PWA), a statistic that helped to identify “winning” players [9]. In the 1970s and early 1980s, the NBA began creating full game logs, which included new statistics: offensive rebounds, steals, blocks, and turnovers [19]. In 1994, basketball analysts conducted research on a basketball statistic that is similar to PWA, Plus-Minus [19]. The Plus-Minus statistic is a count of how many points the player’s team has gained or lost while the player is on the court [1]. There is a weakness in the Plus-Minus statistic, as the statistic is heavily influenced by the player’s teammates and by the player’s opponents.

Due to the potential varying quality of both teammates and opponents, “the marginal Plus-Minus for individual players are inherently polluted” [4].

It was not until 2004 that the computation of NBA analytics advanced and gained importance. For example, the Plus-Minus statistic was heavily improved upon by Dan Rosenbaum. Rosenbaum introduced Adjusted Plus-Minus, which took into consideration both the player’s teammates’ and opponents’ value of play [15]. Pace, which is the total number of possessions a team uses in a single game, was a statistic that was undervalued until the 2004-2005 season [1]. During this season, the Phoenix Sun’s “7 seconds or less” offense produced a pace of 95.9 against the league average of 90.9 [1, 19]. The Suns had the best record in the league that year, with their enhanced pace helping them to lead the league in points, 3 pointers made, and 3 pointers attempted. The play style demonstrated by the Suns is the cause of the increase in the 3 point shot, as seen in Figure 1. This increased level of importance placed onto the 3 point shot led to the birth of the Effective Field Goal Percentage ($eFG\%$) statistic; a statistic that takes into account the difficulty and value of a 3 point attempt versus a 2 point attempt [1, 12]. Since the 2004-2005 season, with the help of technological advancements, more advanced data collection and analytics have led to a better understanding of what leads to a successful team and a successful player.

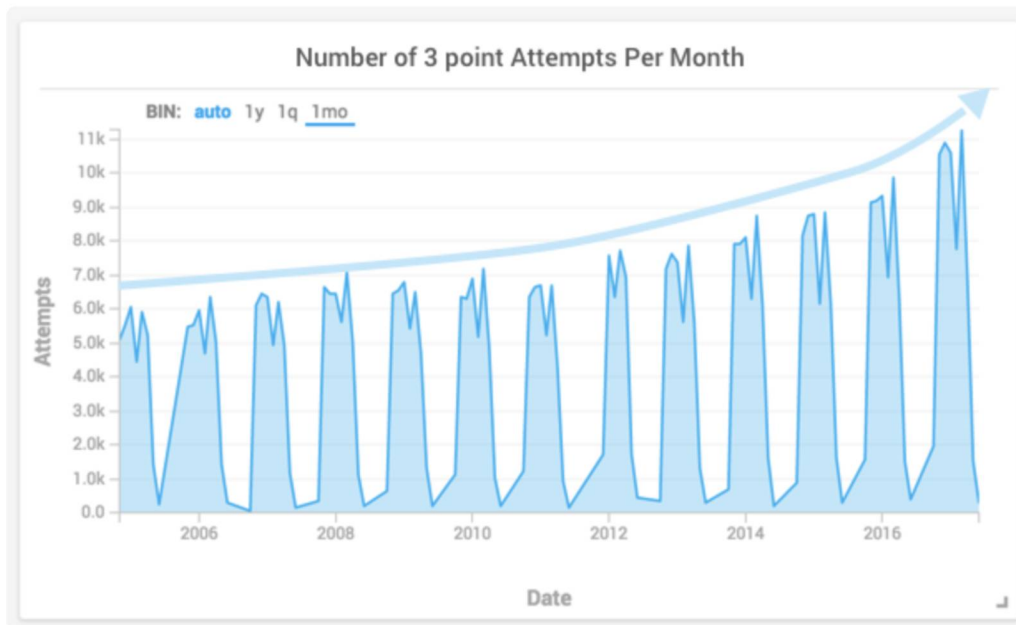


Figure 1: This chart demonstrates one of the effects that the 2004-2005 Phoenix Suns had on the NBA. Between their performance and the increased emphasis on analytics, the NBA has seen a dramatic increase in the use of the 3 point shot [19].

In 2006, Garrick Barr left his assistant coaching position that he held on the Phoenix Suns in order to create Synergy Sports Technology [19]. Synergy created a real-time video-indexing statistical engine that provides play-by-play statistical tracking. It breaks down the play into categories that go beyond the basic box score. Some of the tracked analytics provided by Synergy include play types and their success percentage, from which side of the court a player typically scores on, and what type of move was used

by the player to score (example data in Figure 2). The data provided by Synergy can be helpful in determining a player's tendencies on the court.

**Kobe Bryant's offense comes from:
28% isolation situations,
22% post-ups,
11% as the pick-and-roll ball-handler,
and the rest in an amalgam of transition, cuts, spot-ups, etc.**

Figure 2: A sample provided by Synergy that lists Kobe Bryant's scoring tendencies [19].

The use of cameras and video processing continued to grow, as in 2008 SportVu introduced a video tracking tool for tracking soccer player and referee positioning [19]. In 2010, a small group of NBA teams installed cameras to allow for SportVu's technology to be used for their teams. By 2013, every team in the NBA was required to have cameras installed for this player/referee tracking software. The NBA has since progressed from SportVu and has started to use Second Spectrum and their video processing software. Whereas Synergy provided end of play analytics, Second Spectrum provides dynamic updates to a team's chances to score as the play is occurring. Figure 3 is an example of Second Spectrum video processing software. It uses a combination of basic analytics and player spacing to determine the likelihood the player has to make the shot.



Figure 3: A sample of Second Spectrum’s software. A green bar indicates that the player is wide open and is not currently being contested by a defender. An orange bar indicates that the player is being contested by an opponent but still has room to score. Red indicates that the player is fully guarded and has little to no room to shoot the ball. The percentage over each player shows their accuracy from that range based on data gathered on the player [17].

2.2 Applications of Advanced Analytics

The NBA has taken the sport of basketball and turned it into a multi-billion dollar business. Due to this, there is tremendous pressure on front office members, such as GMs, coaches, and scouts, to increase their team’s chance to win. The increase of advanced analytics has helped front office

members produce more successful and efficient teams.

2.2.1 General Managers

The “new crop of executives” are all salary-cap savvy and embrace all of the data that technology has been able to provide [8]. Analytics are used in all stages of the player acquisition process. Many of the new statistics can help to determine how efficient a player is being. This can help the GM distinguish players who, if given the right amount of minutes, can produce positive results for the team. New models can help GMs view their teams in varying scenarios. Viewing what lineups are effective can help a GM create a roster of players that lead to a greater rate of team success.

2.2.2 Coaches

Coaches utilize the increase in advanced analytics to help their team perform optimally both offensively and defensively. In recent years, coaches have had great success in utilizing optical tracking models. The models produce reports regarding player shooting tendencies. This information can help teach players where their teammates are most efficient on the court. Players can also learn opponents’ tendencies, including shooting and dribbling preferences. Through Second Spectrum, coaches have the ability to quickly pull up video clips for the players [17]. Coaches can utilize these video packages to emphasize strategy on both the offensive and the defensive sides of the court.

2.2.3 Scouts

There are two main roles that scouts can fulfill. One is the role of gathering information on talent that could potentially be added to the team [16]. Scouts typically travel the world going to gyms to view college and foreign professional players. With the emphasis on advanced analytics in recent years, scouts can now spend time scouting players before traveling to see them. This not only saves time, but also reduces travel costs for NBA team organizations. Scouts can also use optical tracking models to better understand how the player affects his current team's offensive and defensive capabilities.

The second role scouts can fulfill is the gathering of information on upcoming opponents. The new advanced statistics can help to determine what aspects of the game in which a player will be successful. If an opposing player shoots well from behind the 3-point line, then the scout can inform the coach to increase the team's pressure on that player. The available models and data gathering techniques help the scouts produce data in a format easily understandable by coaches and players [16]. Figure 4 is an example of formatted data that is used by coaches and players. With the ability to quickly create shooting charts and player tendency graphs of opposing players, scouts can provide coaches with critical information that will influence the team's success.

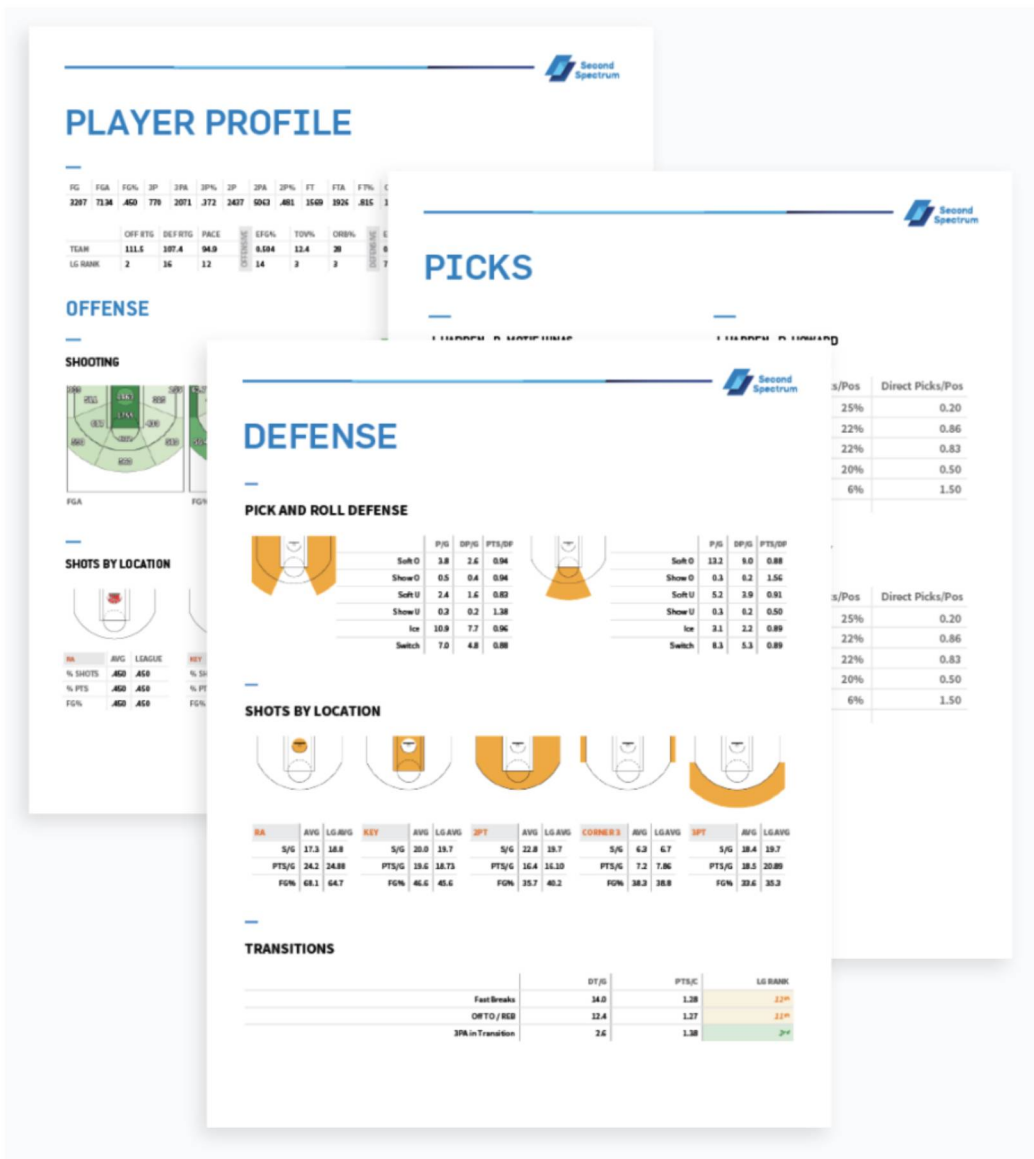


Figure 4: A sample of different reports that are produced by Second Spectrum's software [17].

2.3 New Statistics

In the wake of the Phoenix Suns' 2004-2005 season, the past 15 years have turned the NBA into a data driven league. NBA analysts are always attempting to develop new models and statistics to increase a team's chance to win. Deshpande and Jensen's study focused on estimating a player's impact on his team's chance to win [3]. The approach taken in their study differs from the research presented in this paper, as the emphasis in their study is on player rotations and not statistical performances. New statistics that emphasize a player's ability to perform have been explored as well. One statistic that has been created is *PER*, a statistic that determines the efficiency of a player. Yuanhao Yang, a postgraduate student from University of California Berkeley, has proven that players with a high *PER* lead to a higher team winning percentage [21]. Yang also acknowledges that *PER* is a flawed stat, as both *PER* and the box score of a NBA game are very offensively favored [21]. The typical box score consists of: points, rebounds, assists, steals, blocks, turnovers, made field goals, and missed field goals. Of those statistics, only steals and blocks are defensive categories, with rebounds being split between both offense and defense. Looking deeper at the *PER* formula, explained in Definition 1, steals and blocks are minimal factors compared to the rest of the offensive statistics in the calculation. This bias discredits players like Trevor Ariza, as his defensive ability to lockdown the opposing team's best player will not be noticed in the *PER* calculation.

Definition 1. Player Efficiency Rating: *PER* is a rating developed by John Hollinger. In Hollinger’s words, “The PER sums up all a player’s positive accomplishments, subtracts the negative accomplishments, and returns a per-minute rating of a player’s performance” [1]. This allows for players to be graded on a level playing field, regardless of how many minutes they play or what their team’s pace is. The league average for *PER* is set to 15.00 every year. To calculate *PER*, first *uPER*, or *PER* that has not been adjusted for pace, needs to be calculated:

$$\begin{aligned}
 \mathbf{uPER} = (1/MP) * [& \\
 & 3P+ \\
 & (2/3) * AST+ \\
 & (2 - factor * (teamAST/teamFG)) * FG+ \\
 & (FT * 0.5 * (1 + (1 - (teamAST/teamFG)) + (2/3)* \\
 & (teamAST/teamFG))) - \\
 & VOP * TOV - \\
 & VOP * DRB\% * (FGA - FG) - \\
 & VOP * 0.44 * (0.44 + (0.56 * DRB\%)) * (FTA - FT) + \\
 & VOP * (1 - DRB\%) * (TRB - ORB) + \\
 & VOP * DRB\% * ORB +
 \end{aligned}$$

$$VOP * STL +$$

$$VOP * DRB\% * BLK -$$

$$PF * ((lgFT/lgPF) - 0.44 * (lgFTA/lgPF) * VOP)]$$

Where:

$$\mathbf{factor} = (2/3) - (0.5 * (lgAST/lgFG)) / (2 * (lgFG/lgFT))$$

$$\mathbf{VOP} \text{ (Value of Possession)} = lgPTS / (lgFGA - lgORB + lgTOV + 0.44 * lgFTA)$$

$$\mathbf{DRB\%} = (lgTRB - lgORB) / lgTRB$$

team = prefix indicating of team rather than player

lg = prefix indicating of league rather than player

FG = number of field goals made

FT = number of free throws made

To adjust for pace, *PaceAdj* needs to be calculated:

$$\mathbf{PaceAdj} = 2 * lgPPG / (teamPPG + oppPPG)$$

Then applying the adjustment we get *APER*:

$$\mathbf{APER} = (PaceAdj) * uPER$$

Finally:

$$\mathbf{PER} = APER * (15/\lg APER)$$

There are other statistics besides PER that have been developed that provide a better understanding of how a player performs when they are on the court. Game Score provides a rating for a player’s statistical performance. As seen in its formula in Definition 2, the calculation develops the rating based on basic game statistics without taking into account time of play or pace. More advanced analytics were created by a research group that included Dean Oliver and Kevin Pelton, who are two of the creators of APBRmetrics (Association of Professional Basketball Research Metrics). They define key advanced statistics in basketball, which are defined in Definition 2, and why the statistics are significant [7]. A notable set of statistics are the percentage-based calculations, which have been created in order to take into consideration pace of the team and minutes of the player [7]. Benjamin Morris, who founded *skepticalsports.com* and now works for the sport analytics site *FiveThirtyEight*, explains how steals are the most important statistic in determining a player’s contribution to a team’s chance to win [10]. Specifically Steal percentage ($STL\%$), which is a percentage-based statistic, can possibly be seen as one of the most valuable defensive statistics for this research. This is due to the fact that a steal is “weighted nine times more heavily when predicting a player’s impact than a marginal point” [10].

Definition 2. Basic Statistics: statistics that do not take into account time of play or pace:

MP - Minutes Played

PTS - Points Scored

GmSc - Game Score: a statistic created by John Hollinger to give a rough measure of a player's productivity for a single game by incorporating weights based on the importance of the statistic. The scale is similar to that of points scored, where 40 is an outstanding performance and 10 is an average performance. The statistic is calculated via the following formula: $GmSc = PTS + 0.4 * FG - 0.7 * FGA - 0.4 * (FTA - FT) + 0.7 * ORB + 0.3 * DRB + STL + 0.7 * AST + 0.7 * BLK - 0.4 * PF - TOV$

Advanced statistics: statistics that build upon basic statistics in order to provide more information regarding how efficient the player is while they are playing. These statistics are calculated by incorporating *MP* of both the player and the team or by incorporating other basic statistics to produce a more meaningful statistic. These statistics were created in the mid 2000s by John Hollinger and Dean Oliver:

TS% - True Shooting Percentage: a measure of shooting efficiency that takes into account *FG*, 3-point *FG*, and *FT*. It incorporates *PTS*

and *TSA* (True Shooting Attempts). The formula is as follows:

$$TS\% = PTS / (2 * TSA)$$

ORB% - Offensive Rebound Percentage: an estimate of the percentage of available offensive rebounds a player grabbed while he was on the floor. A player with a high *ORB%* means that while that player is on the court, the player increases the number of opportunities his team has to score. The formula is as follows: $ORB\% = 100 * (ORB * (TeamMP/5)) / (MP * (TeamORB + OppDRB))$

DRB% - Defensive Rebound Percentage: an estimate of the percentage of available defensive rebounds a player grabbed while he was on the floor. A player with a high *DRB%* means that while that player is on the court, the opposing team gets less second chance attempts at scoring the ball. The formula is as follows: $DRB\% = 100 * (DRB * (TeamMP/5)) / (MP * (TeamDRB + OppORB))$

TRB% - Total Rebound Percentage: an estimate of the percentage of available rebounds a player grabbed while he was on the floor. A player with a high *TRB%* means that while that player is on the court, the player eliminates opportunities for the opposing team to gain possession of the ball during a missed shot attempt. The formula is as follows: $TRB\% = 100 * (TRB * (TeamMP/5)) / (MP * (TeamTRB + OppTRB))$

AST% - Assist Percentage: an estimate of the percentage of teammate field goals a player assisted while he was on the floor. A player with a high *AST%* means that while that player is on the court, the offense of the player's team is moving the ball and playing offense efficiently. The formula is as follows: $AST\% = 100 * AST / (((MP / (TeamMP / 5)) * TeamFG) - FG)$

STL% - Steal Percentage: an estimate of the percentage of opponent possessions that end with a steal by the player while he was on the floor. A player with a high *STL%* means that while that player is on the court, the opposing team loses possessions and the player's team gains more possessions/opportunities to score. The formula is as follows: $STL\% = 100 * (STL * (TeamMP / 5)) / (MP * OppPoss)$

BLK% - Block Percentage: an estimate of the percentage of opponent two-point field goal attempts blocked by the player while he was on the floor. A player with a high *BLK%* means that while that player is on the court, the opposing team has more missed field goals and means that the player's team has a better internal defense. The formula is as follows: $BLK\% = 100 * (BLK * (TeamMP / 5)) / (MP * (OppFGA - Opp3PA))$

TOV% - Turnover Percentage: an estimate of turnovers per 100 plays. A player with a low *TOV%* means that while that player is on the

court, the player does not hurt his team by losing control of the ball and giving the opposing team possession of the ball without a shot attempt. The formula is as follows: $TOV\% = 100 * TOV / (FGA + 0.44 * FTA + TOV)$

USG% - Usage Percentage: an estimate of the percentage of team plays used by a player while he was on the floor. A player with a high *USG%* means that while that player is on the court, he is being utilized in the offense and is in control of the ball a large amount of the time. The formula is as follows: $USG\% = 100 * ((FGA + 0.44 * FTA + TOV) * (TeamMP/5)) / (MP * (TeamFGA + 0.44 * TeamFTA + TeamTOV))$

Based on a player's performance on the court, the statistics known as Offensive Rating and Defensive Rating can be calculated (defined in Definition 3 and Definition 4). These statistics, created by Dean Oliver, estimate how many points are produced/allowed by the player per 100 possessions while he is on the court [12]. Defensive Rating may prove to be a key statistic in this research, as it not only accounts for blocks, steals, and defensive rebounds, but it also estimates the number of forced turnovers and missed shots by the player [1].

Definition 3. Offensive Rating (*ORtg*): for players it is points produced per 100 possessions, while for teams it is points scored per 100 possessions

[1, 12]. Developed by Dean Oliver, *ORtg* was created to provide a rating of how well a player's team is performing while they are on the court. It can also be used to determine how well a team's offense is performing. *ORtg* is best used to determine how well certain teams or lineups perform or to rate the offensive abilities of players who have similar *USG%* and roles in the offense. The calculation is adjusted for minutes played and incorporates individual based statistics (*PTS*, *AST*, *FGM*, etc.) and team based statistics (*TeamFGM*, *TeamAST*, *TeamORB*, etc). The first part that needs to be calculated for *ORtg* is the Scoring Possessions Formula (*ScPoss*):

$$\begin{aligned} \mathbf{ScPoss} &= (FGPart + ASTPart + FTPart) * \\ &\quad (1 - (TeamORB/TeamScoringPoss) * \\ &\quad TeamORBWeight * TeamPlay\%) + ORBPart \end{aligned}$$

Where:

$$\mathbf{FGPart} \text{ (FG made due to player activity)} = FGM * (1 - 0.5 * ((PTS - FTM)/(2 * FGA)) * qAST)$$

$$\begin{aligned} \mathbf{qAST} \text{ (Quantity of Assists During Player Activity)} &= \\ &\quad ((MP/(TeamMP/5)) * (1.14 * ((TeamAST - AST)/TeamFGM))) + \\ &\quad (((TeamAST/TeamMP) * MP * 5 - AST)/ \\ &\quad ((TeamFGM/TeamMP) * MP * 5 - FGM)) * \end{aligned}$$

$$(1 - (MP/(TeamMP/5))))$$

ASTPart (Assists accrued while the player was active) =

$$0.5 * (((TeamPTS - TeamFTM) - (PTS - FTM)) / (2 * (TeamFGA - FGA))) * AST$$

FTPart (FT made/attempted while the player was active) = $(1 - (1 -$

$$(FTM/FTA))^2) * 0.4 * FTA$$

TeamScoringPoss (Total amount of scoring possessions by the team)

$$= TeamFGM + (1 - (1 - (TeamFTM/TeamFTA))^2) * TeamFTA * 0.4$$

TeamORBWeight (weight to adjust ORB based on total scoring plays)

$$= ((1 - TeamORB\%) * TeamPlay\%) / ((1 - TeamORB\%) * TeamPlay\% + TeamORB\% * (1 - TeamPlay\%))$$

TeamORB% (percent of potential ORB that were acquired) =

$$TeamORB / (TeamORB + (OpponentTRB - OpponentORB))$$

TeamPlay% (percent of plays that led to a score) =

$$TeamScoringPoss / (TeamFGA + TeamFTA * 0.4 + TeamTOV)$$

ORBPart (the final adjusted value of ORB) = $ORB * TeamORBWeight *$

$$TeamPlay\%$$

Missed FG and Missed FT Possessions ($FGxPoss$ and $FTxPoss$) are calculated as follows:

$$\mathbf{FGxPoss} = (FGA - FGM) * (1 - 1.07 * TeamORB\%)$$

$$\mathbf{FTxPoss} = ((1 - (FTM/FTA))^2) * 0.4 * FTA$$

Total Possessions ($TotPoss$) are then computed as follows:

$$\mathbf{TotPoss} = ScPoss + FGxPoss + FTxPoss + TOV$$

Now, Individual Points Produced ($PProd$) must also be calculated:

$$\mathbf{PProd} = (PProdFGPart + PProdASTPart + FTM) * (1 - (TeamORB/TeamScoringPoss) * TeamORBWeight * TeamPlay\%) + PProdORBPart$$

Where:

$$\mathbf{PProdFGPart} \text{ (FG that produced points)} = 2 * (FGM + 0.5 * 3PM) * (1 - 0.5 * ((PTS - FTM)/(2 * FGA)) * qAST)$$

$$\mathbf{PProdASTPart} \text{ (assists acquired during possessions that produced points)} = 2 * ((TeamFGM - FGM + 0.5 * (Team3PM - 3PM)) / (TeamFGM - FGM)) * 0.5 * (((TeamPTS - TeamFTM) - (PTS - FTM)) / (2 * (TeamFGA - FGA))) * AST$$

$$\mathbf{PProdORBPart} \text{ (ORB that led to points)} = ORB * TeamORBWeight * TeamPlay\% * (TeamPTS / (TeamFGM + (1 - (1 - (TeamFTM / TeamFTA))^2) * 0.4 * TeamFTA))$$

We can now calculate the player's individual Offensive Rating:

$$\mathbf{ORtg} = 100 * (PProd / TotPoss)$$

Definition 4. Defensive Rating (*DRtg*): for players and teams it is points allowed per 100 possessions. [1, 12]. Developed by Dean Oliver, *DRtg* was created to provide a rating of how well a team is playing defense. This can be used to determine how well a team plays defense or how well a team plays defense while a specific player is on the court. The calculation is adjusted for minutes played and incorporates individual statistics (*STL*, *BLKS*, *DRB*, etc.) and team based statistics (*TeamPF*, *OpponentFGM*, *OpponentFTM*, etc.). To calculate *DRtg*, *Stops* is calculated first. This is done by combining *Stops1* (number of possessions/shots prevented by a single player) and *Stops2* (number of possessions/shots prevented by a team):

$$\mathbf{Stops} = Stops1 + Stops2$$

Where:

$$\mathbf{Stops1} = STL + BLK * FMwt * (1 - 1.07 * DOR\%) + DRB * (1 - FMwt)$$

$$\mathbf{FMwt} = (DFG\% * (1 - DOR\%)) / (DFG\% * (1 - DOR\%) + (1 - DFG\%) * DOR\%)$$

$$\mathbf{Stops2} = (((OpponentFGA - OpponentFGM - TeamBLK) / TeamMP) * FMwt * (1 - 1.07 * DOR\%) + ((OpponentTOV - TeamSTL) / TeamMP)) * MP + (PF / TeamPF) * 0.4 * OpponentFTA * (1 - (OpponentFTM / OpponentFTA))^2$$

DOR% (percentage of offensive rebounds given to the opposing team)
 $= OpponentORB / (OpponentORB + TeamDRB)$

DFG% (opponent field goal percentage) =
 $OpponentFGM / OpponentFGA$

TeamDefensiveRating = $100 * (OpponentPTS / TeamPossessions)$

DPtsperScPoss (opponent points scored per scoring possession) =
 $OpponentPTS / (OpponentFGM + (1 - (1 - (OpponentFTM / OpponentFTA))^2) * OpponentFTA * 0.4)$

Also necessary is the calculation of *Stop%*, which is the rate at which a player forces a defensive stop as a percentage of individual possessions played against an opponent:

$$\mathbf{Stop\%} = (Stops * OpponentMP) / (TeamPossessions * MP)$$

Individual Defensive Rating can now be computed:

$$\mathbf{DRtg} = TeamDefensiveRating + 0.2 * (100 * DPtsperScPoss * (1 - Stop\%) - TeamDefensiveRating)$$

2.4 What is Machine Learning?

In the current age of technology, there is nearly infinite data for people to use and manipulate. Researchers are constantly trying to use the overabundance of data in order to create meaningful applications. Machine Learning, a sub-category of Artificial Intelligence, provides analysts a way to create models based on both structured and unstructured data. Instead of creating rules and model structures by hand, analysts can employ Machine Learning to create models more efficiently and to produce greater predictive, data-driven results [14]. Machine Learning is of great use not only in Computer Science research but also in our everyday lives. Machine Learning is the driving force behind many useful day-to-day utilities, such as e-mail spam filters, Web search engines, and, potentially in the near future, autonomous cars [14].

2.4.1 Different Types of Machine Learning

When constructing Machine Learning algorithms, there are two approaches that are the most popular to “teach” the models. These two approaches are called unsupervised learning and supervised learning.

Unsupervised Learning

Unsupervised learning involves using unlabeled data. This allows for the model to create its own labels for the training data records by forming groups based on characteristics found within the data. An unsupervised learning technique for forming groups out of the data is known as clustering [14]. Clustering allows the model to create groups based on records that have similar characteristics. Clustering is an excellent way for finding hidden meaning and meaningful relationships between data records. Figure 5 is a graphical representation of clustering, with the circles representing different groups.

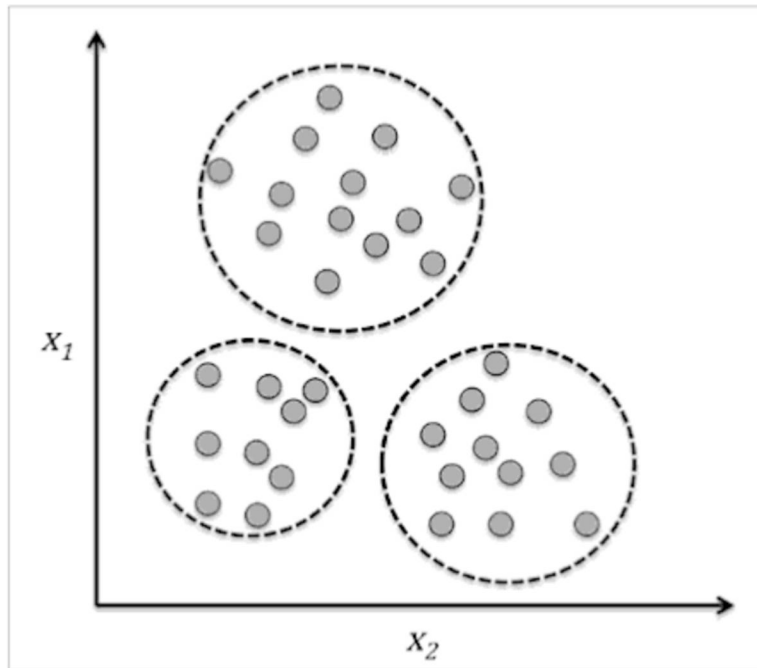


Figure 5: A graphical representation of clustering. Each circle of data points represents a group of records that have similar traits [14].

Supervised Learning

Supervised learning involves labeling the model's training data. Each record of the training data has a predicted result which is represented by the label associated with the record. The word "supervised" is used because the training data records are tested with the outcome being known via the label [14]. This allows the model to learn what data characteristics lead to specific, pre-defined results. A supervised learning model will be used in this study, where the records are player's statistical performances and the labels

are “Win” and “Loss”. The process flow of a supervised learning model is illustrated in Figure 6.

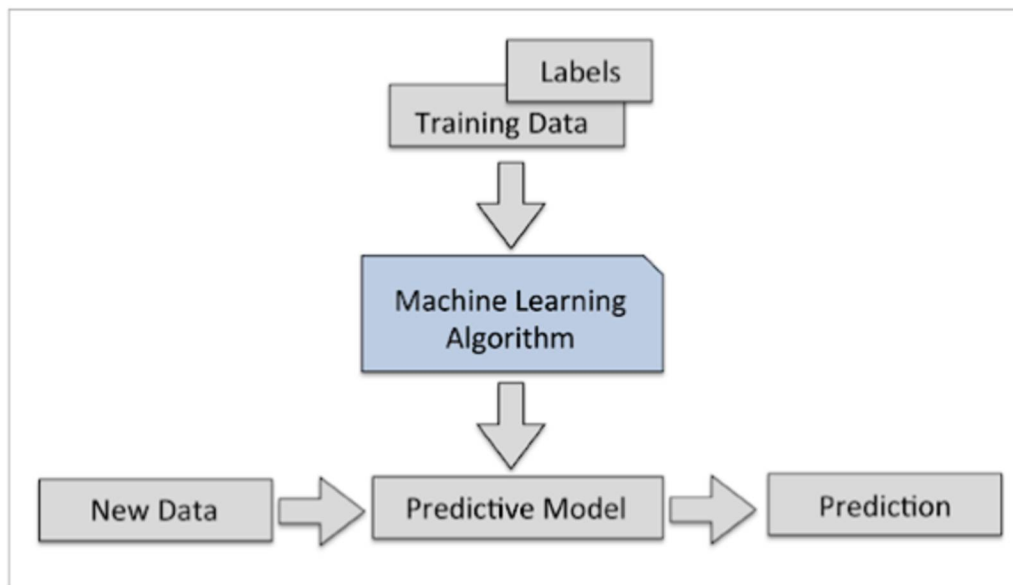


Figure 6: A graphical representation of a supervised learning model [14].

2.4.2 Regression vs Classification Models

Two model techniques used in supervised learning are Regression models and Classification models. Both types of models have their usefulness based on the type of problem that is being addressed.

Benefits of Regression Models

Regression analysis models are used for predicting continuous outcomes. An example of this is a linear regression model. In this type of model, a straight line is developed to fit a set of data containing predictor variables

and response variables (as seen in Figure 7) [14]. The line created minimizes the distance, also known as the average squared distance, between the sample data and the line [14]. The intercept and slope from the fitted line are used to predict outcomes of new data. This type of model would be useful in continuously predicting the outcome of an NBA game as the game is taking place. For example, with 10 minutes left in the game, the model could predict that a team has a 65% chance of winning the game based on the score, time, and other factors. However, with 1 minute left in the game, the classifier could predict a 99% chance of the team winning the game since the team is up by so many points and the time left is minimal.

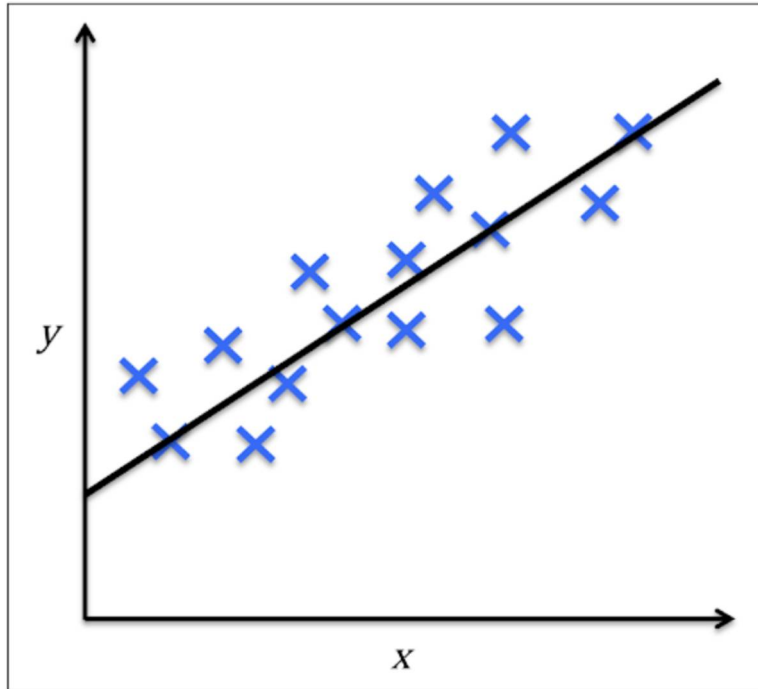


Figure 7: An example of a linear regression model plot. The predictor variables are represented by x and the response variables are represented by y [14].

Benefits of Classification Models

Whereas regression models produce continuous results, classification models provide discrete solutions. The goal of classification models is to predict, based on previous training, which category data represents [14]. An example of a classification model is a decision tree classifier. The classifier divides the feature space into rectangles, with each rectangle representing a different category. In Figure 8, the colored areas of the graph represent different

results based on the test data.

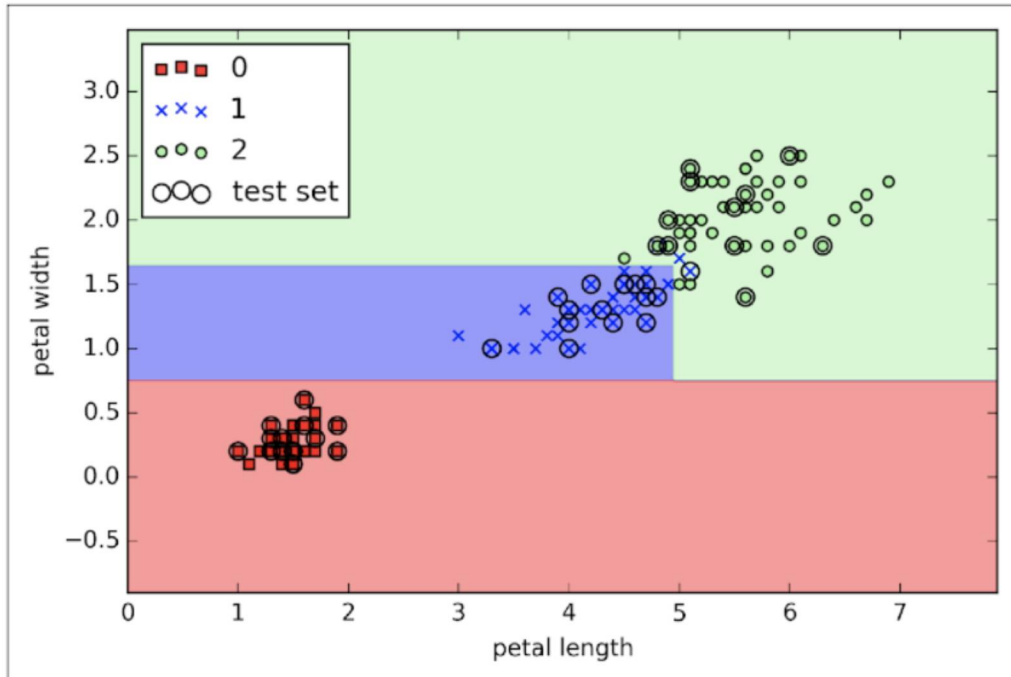


Figure 8: An example of the data splitting process that occurs in a decision tree classifier. Each colored section represents a different category [14].

The colored areas in Figure 8 help to create rules. When new data is run through the decision tree, the model uses these rules to determine what class the data represents. Figure 9 is an illustration of a decision tree and its rules. This allows users to clearly see the logic behind the model.

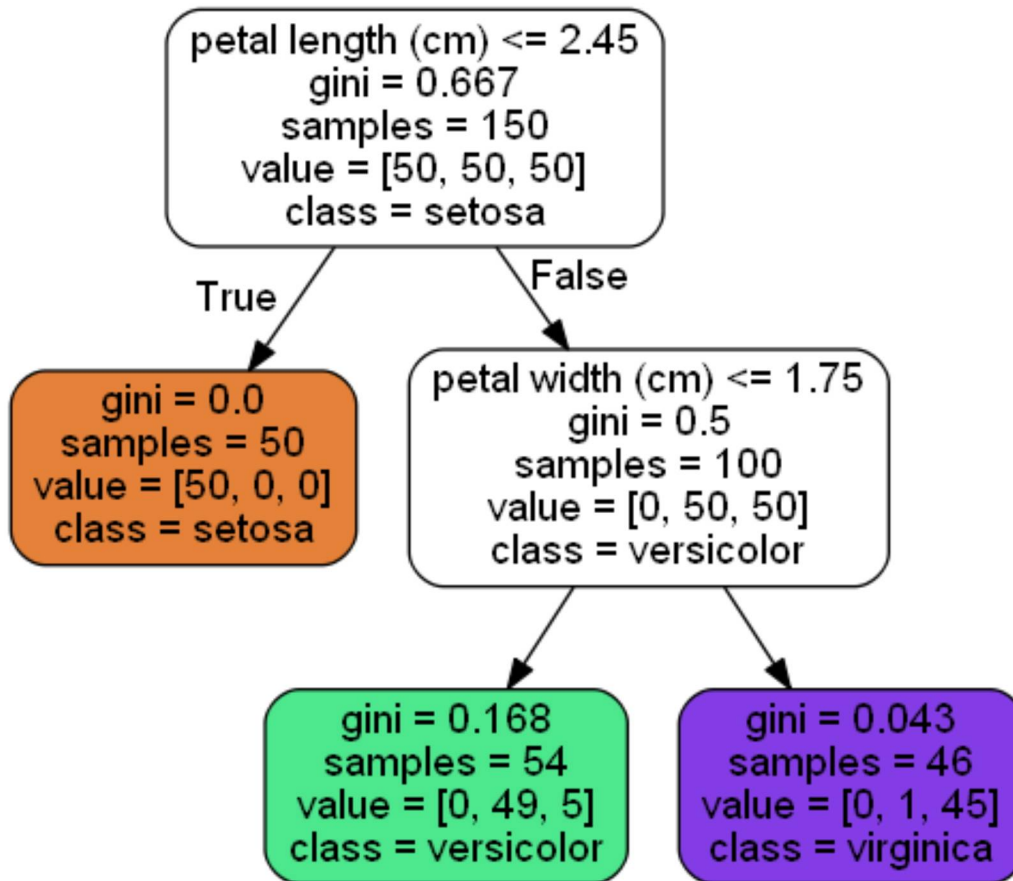


Figure 9: A plot of a decision tree. The rules and logic created by the Decision Tree Classifier can be clearly viewed through this image [2].

A decision tree classifier will be used in this study. The two categories will be “Win” and “Loss” and the test data will be the different statistical performances of the player.

2.5 Python

Python is a very popular programming language among the computer science community. It is a high-level, object oriented language that has syntax that focuses on readability [20]. The ease of readability allows for the developer to quickly understand the code and pick up work in the middle of a project with minimal down-time. Python is also very accessible, as the standard library and interpreter are free and it is usable on all major platforms. Python originally started as a "scripting language" which focus on small, short tasks; however its popularity has grown tremendously and is used for big projects by major companies, including Google [20].

Python's massive support base is a huge factor in its success. With the language being open source and community developed, the language is always seeing improvements to its core functionality. The external API support is also tremendous with the language, allowing for the abilities of the language to grow. Libraries such as *NumPy* and *SciPy* build upon lower level languages such as C to create faster performing programs [14]. There are also great open source libraries for data analysis such as *pandas*. Most importantly, there is a key library for machine learning that will be utilized in this study, which is *scikit-learn*.

2.5.1 *pandas* API

An open source library created and edited by the Python community, *pandas* is a library that will be utilized in this study. The library provides easy to use data structures that are vital in data analysis programs [18]. *DataFrame* is a 2-dimensional labeled structure that can contain columns of different data types [18]. In conjunction with the following commands, the *DataFrame* structure will be used to store and organize player data from csv files.

read_csv

A part of the *pandas* I/O API, *read_csv* is a top level reader function that returns a *DataFrame* structure (Figure 10). This function will be utilized to import the NBA player data into the program.

CSV & Text files

The workhorse function for reading text files (a.k.a. flat files) is `read_csv()`. See the [cookbook](#) for some advanced strategies.

Parsing options

`read_csv()` accepts the following common arguments:

Basic

`filepath_or_buffer` : *various*

Either a path to a file (a `str`, `pathlib.Path`, or `py._path.local.LocalPath`), URL (including http, ftp, and S3 locations), or any object with a `read()` method (such as an open file or `StringIO`).

`sep` : *str, defaults to ',' for read_csv(), \t for read_table()*

Delimiter to use. If `sep` is `None`, the C engine cannot automatically detect the separator, but the Python parsing engine can, meaning the latter will be used and automatically detect the separator by Python's builtin sniffer tool, `csv.Sniffer`. In addition, separators longer than 1 character and different from `'\s+'` will be interpreted as regular expressions and will also force the use of the Python parsing engine. Note that regex delimiters are prone to ignoring quoted data. Regex example: `'\x\t'`.

`delimiter` : *str, default None*

Alternative argument name for `sep`.

`delim_whitespace` : *boolean, default False*

Specifies whether or not whitespace (e.g. `' '` or `'\t'`) will be used as the delimiter. Equivalent to setting `sep='\s+'`. If this option is set to `True`, nothing should be passed in for the `delimiter` parameter.

Figure 10: The documentation on the `read_csv` procedure with basic parameter inputs [18].

concat

Due to the NBA player data being in several different csv files, *concat*, which is short for concatenate, will be used to merge the data into one *DataFrame* structure. Since all of column headers are the same, *concat* provides an easy and efficient way to merge the data. An example of *concat* and its ability to combine data structures can be found in Figure 11.

df1					Result				
	A	B	C	D		A	B	C	D
0	A0	B0	C0	D0	0	A0	B0	C0	D0
1	A1	B1	C1	D1	1	A1	B1	C1	D1
2	A2	B2	C2	D2	2	A2	B2	C2	D2
3	A3	B3	C3	D3	3	A3	B3	C3	D3
df2					4	A4	B4	C4	D4
	A	B	C	D	5	A5	B5	C5	D5
4	A4	B4	C4	D4	6	A6	B6	C6	D6
5	A5	B5	C5	D5	7	A7	B7	C7	D7
6	A6	B6	C6	D6	df3				
7	A7	B7	C7	D7		A	B	C	D
df3					8	A8	B8	C8	D8
	A	B	C	D	9	A9	B9	C9	D9
8	A8	B8	C8	D8	10	A10	B10	C10	D10
9	A9	B9	C9	D9	11	A11	B11	C11	D11
10	A10	B10	C10	D10					
11	A11	B11	C11	D11					

Figure 11: An example of *concat* and its ability to merge different *DataFrame* structures together [18].

2.5.2 NumPy API

NumPy is a package of scientific computing objects and functions. These objects and functions include multi-dimensional containers of generic data and high-level mathematical operations that work in conjunction with these containers [11]. Figure 12 explains one of the key *NumPy* functions used in this study, *isfinite*. *NumPy* is fundamental to the creation and utilization of several packages, including *pandas* and *scikit-learn*.

```
numpy.isfinite(x, /, out=None, *, where=True, casting='same_kind', order='K', dtype=None, subok=True[, signature, extobj]) =  
<ufunc isfinite>
```

Test element-wise for finiteness (not infinity or not Not a Number).

The result is returned as a boolean array.

Parameters: `x` : *array_like*

Input values.

`out` : *ndarray, None, or tuple of ndarray and None, optional*

A location into which the result is stored. If provided, it must have a shape that the inputs broadcast to. If not provided or *None*, a freshly-allocated array is returned. A tuple (possible only as a keyword argument) must have length equal to the number of outputs.

`where` : *array_like, optional*

This condition is broadcast over the input. At locations where the condition is True, the *out* array will be set to the ufunc result. Elsewhere, the *out* array will retain its original value. Note that if an uninitialized *out* array is created via the default `out=None`, locations within it where the condition is False will remain uninitialized.

****kwargs**

For other keyword-only arguments, see the [ufunc docs](#).

Returns: `y` : *ndarray, bool*

True where `x` is not positive infinity, negative infinity, or NaN; false otherwise. This is a scalar if `x` is a scalar.

Figure 12: The *isfinite* function is a tool that *NumPy* provides. It is used to identify null data and eliminate it from the *DataFrame* structures [11].

2.5.3 *scikit-learn* API

Another library built off of *NumPy*, *scikit-learn* is a library that provides simple and efficient data mining and data analysis tools [2]. There are several different types of machine learning frameworks that *scikit-learn* provides, including classification, regression, and clustering models. From the classification models available in the API, the Decision Tree Classifier model will be used in this study. The statistics from Definition 2, Definition 3, and Definition 4 will be run through the Decision Tree Classifier via the *fit* function, which will build the model based on the data.

The *scikit-learn* library also provides other tools that will be used to test the accuracy of the model.

Predict (X)	<p>Predict class value for X.</p> <p>For a classification model, the predicted class for each sample in X is returned.</p>
Score (X, y)	<p>Returns the mean accuracy on the given test data and labels.</p> <p>In multi-label classification, this is the subset accuracy which is a harsh metric since you require for each sample that each label set be correctly predicted.</p> <p>X = Test samples y = True labels for X</p>
<i>feature_importances</i>	<p>Return the feature importances.</p> <p>The importance of a feature is computed as the (normalized) total reduction of the criterion brought by that feature. It is also known as the Gini importance.</p>

Figure 13: The definitions of the procedures and the attribute that will be used to analyze the models [2].

The code from this study utilizes the functions in Figure 13 in order to quickly and efficiently analyze the model and its results. An example of the code is in Figure 14.

```
1 from sklearn.tree import DecisionTreeClassifier
2 import pandas as pd
3 import numpy as np
4
5 dt = DecisionTreeClassifier()
6
7 dt.fit(X_train, y_train)
8
9 dt.score(X_test, y_test, sample_weight=None)
```

Figure 14: A section of the code in this study. The *fit* functions is used to train the model. The *score* function is then used to determine the accuracy of the test data [2].

Line 5 in Figure 14 creates the Decision Tree Classifier. Line 7 then trains the model by utilizing the *fit* function. The parameters *X_train* and *y_train* represent the input data and expected results respectively. With the model trained, line 9 utilizes what it learned from line 7 to predict results from data in *X_test* and then to compare those results against the actual results in *y_train*. This “score” identifies how well the model performed in predicting the correct results. For the purpose of this study, the higher the score, the more important a player’s statistical performance is to his team’s success.

3 Contribution

3.1 Statistics Analyzed

The goal is to use a Decision Tree Classifier to determine an NBA player's value to a team's chance to win. To do this, the results from the *score* procedure in Figure 13 will be compared to *PER* (which was defined in Definition 1). Another Hollinger statistic called Game Score is employed to look at the players' game by game performances. It encompasses all of the basic statistics that are covered in *PER*; however, there is no bias generated for pace or time of play.

Though Game Score is a viable way to judge a player's basic statistical performance, it does not consider how much time the player spends on the court. Some players, such as a more defensive, team-based player like Tyson Chandler, affect the game just by being present on the court. Due to this, minutes played will be added to the model. Also, to give all players' statistics equal value no matter how many minutes they play, the percentage-based statistics will be used in the model instead of the raw, non-minutes based statistics. To round out the statistical categories, points scored will also be added to the model. Finally, in order to expand upon the individual-focused statistics noted above, the model will also take into consideration other team-focused efficiency metrics. The two team-focused metrics that will be added to the model are Offensive Rating and Defensive Rating, which focus on the

player's impact on their team's offensive and defensive performances while the player is on the court.

3.2 Creating Models

All of the variables in Definition 2, Definition 3, and Definition 4 will be tested through the *scikit-learn* Decision Tree Classifier's *feature_importances* function specified in Figure 13. Using these results, I will be able to determine the difference in play styles between an All-Star and a Starter. The test data will be split into 2 different trees: an All-Star tree and a Starter tree. The All-Star tree will be built with game data from players who were perennial All-Stars between the 2014 and 2018 seasons. The Starter tree will be built on players that were starters on their respective teams and played at least 30 minutes a game. By creating two different trees based on two different categories of players and by studying the results from *feature_importances*, we can identify the differences between a All-Star and a Starter. The two models will also provide a way to determine if a certain play style equates to more wins for a given player's team.

3.3 Gathering and Organizing the Data

The data was gathered from *www.basketball-reference.com*. The website is a database of basketball player and team data [1]. The database on the website provides all of the statistics that will be needed for the study and pro-

vides easy data extraction methods evident in Figure 15. Data was gathered from players based on the criteria previously stated.

2016-17 Regular Season

Share & more [Glossary](#)

Modify & Share Table
 Embed this Table
 Get as Excel Workbook (experimental)
 Get table as CSV (for Excel)
 Strip Mobile Formatting
 Copy Link to Table to Clipboard
 About Sharing Tools
 Video: SR Sharing Tools & How-to
 Video: Stats Table Tips & Tricks

Rk	G	Date	Age	Tm	Opp	GS	MP	FG	FGA	FG%	3P	3PA	3P%	FT	FTA	FT%	ORB	DRB	TRB	AST	STL	BLK	TOV	PF	PTS	GmSc	+/-	
1	1	2016-10-25	31-300	CLE	NYK												00	3	8	11	14	0	1	4	3	19	22.2	+27
2	2	2016-10-28	31-303	CLE	TOR												14	2	6	8	7	0	0	5	2	21	14.1	0
3	3	2016-10-29	31-304	CLE	ORL												36	1	5	6	9	1	1	2	23	20.4	+1	
4	4	2016-11-01	31-307	CLE	HOU												00	3	10	13	8	0	0	4	19	16.5	+15	
5	5	2016-11-03	31-309	CLE	BOS												00	1	6	7	12	1	0	2	2	30	28.5	+11
6	6	2016-11-05	31-311	CLE	PHI												57	1	7	8	14	2	1	5	1	25	22.0	+8
7	7	2016-11-08	31-314	CLE	ATL												00	2	7	9	5	3	0	1	3	23	22.1	+7
8	8	2016-11-11	31-317	CLE	WAS												67	0	10	10	5	2	2	6	1	27	20.3	+7
9	9	2016-11-13	31-319	CLE	CHO												00	0	8	8	8	1	1	3	0	19	13.8	+12
10	10	2016-11-15	31-321	CLE	TOR	W (+4)	1	38:04	10	15	.667	2	5	.400	6	10	.600	0	9	9	15	1	0	5	2	28	28.3	+1
11		2016-11-16	31-322	CLE	IND	L (-10)											Did Not Dress											
12	11	2016-11-18	31-324	CLE	DET	W (+23)	1	28:03	9	14	.643	1	3	.333	2	3	.667	0	3	3	3	0	1	2	1	21	15.7	+21
13	12	2016-11-23	31-329	CLE	POR	W (+12)	1	37:30	11	21	.524	2	3	.667	7	8	.875	1	9	10	13	3	0	3	0	31	32.8	+17
14	13	2016-11-25	31-331	CLE	DAL	W (+38)	1	29:03	6	13	.462	2	5	.400	5	5	1.000	0	5	5	11	1	0	4	1	19	18.1	+33
15	14	2016-11-27	31-333	CLE	PHI	W (+4)	1	41:30	9	19	.474	1	4	.250	7	9	.778	2	8	10	13	1	0	5	0	26	24.4	+2
16	15	2016-11-29	31-335	CLE	MIL	L (-17)	1	32:53	8	16	.500	3	8	.375	3	7	.429	0	4	4	4	0	0	7	3	22	8.2	-12
17	16	2016-12-01	31-337	CLE	LAC	L (-19)	1	33:37	5	14	.357	0	2	.000	6	11	.545	1	4	5	5	2	0	5	2	16	7.8	-20
18	17	2016-12-02	31-338	CLE	CHI	L (-6)	1	44:40	13	22	.591	1	3	.333	0	0		0	5	5	13	0	0	8	2	27	18.6	-3
19	18	2016-12-05	31-341	CLE	TOR	W (+4)	1	42:01	12	26	.462	2	7	.286	8	9	.889	3	5	8	7	2	0	1	2	34	28.9	+9
20	19	2016-12-07	31-343	CLE	NYK	W (+32)	1	32:27	7	10	.700	1	2	.500	10	14	.714	2	4	6	7	1	2	4	2	25	24.3	+32
Rk	G	Date	Age	Tm	Opp	GS	MP	FG	FGA	FG%	3P	3PA	3P%	FT	FTA	FT%	ORB	DRB	TRB	AST	STL	BLK	TOV	PF	PTS	GmSc	+/-	
21	20	2016-12-09	31-345	CLE	MIA	W (+30)	1	37:19	12	22	.545	1	3	.333	2	4	.500	2	6	8	8	3	0	3	2	27	23.6	+21
22	21	2016-12-10	31-346	CLE	CHO	W (+11)	1	42:27	17	24	.708	5	10	.500	5	9	.556	1	8	9	10	3	1	5	3	44	40.0	-2
23	22	2016-12-13	31-349	CLE	MEM	W (+17)	1	36:29	9	17	.529	0	4	.000	5	8	.625	1	5	6	8	3	0	6	1	23	17.9	+8
24		2016-12-14	31-350	CLE	MEM	L (-8)											Not With Team											
25	23	2016-12-17	31-353	CLE	LAL	W (+11)	1	38:55	9	18	.500	2	5	.400	6	11	.545	1	6	7	9	2	2	2	1	26	24.8	+17
26	24	2016-12-20	31-356	CLE	MIL	W (+6)	1	47:29	12	25	.480	5	9	.556	5	6	.833	3	9	12	7	1	1	2	2	34	29.5	+8
27	25	2016-12-21	31-357	CLE	MIL	W (+11)	1	34:25	12	24	.500	4	7	.571	1	1	1.000	1	8	9	6	0	0	2	0	29	22.3	+26

Figure 15: A page of player’s data, game by game, for a whole season. The *www.basketball-reference.com* website provides several different ways to export the data. This eliminates the need to create a web scrapper for the website [1].

With the data from the website, the difference between the game’s statistics and the player’s averages for the season were calculated for 11 of the 14 variables. This puts all players on an even playing field, allowing for a

player who plays fewer minutes to not have his lower average numbers be outweighed by an All-Star's higher numbers. *ORtg*, *DRtg*, and *GmSc* will all be based on a player's per game value and will not be adjusted by the player's averages.

3.4 Training the Models

With the data set formed, the data was run through the Decision Tree Classifier and the *Predict* and *Score* procedures and *feature_importances* attribute will be used to test the models' capabilities. *Predict* and *Score* will be used in conjunction with one another, and the final result of *Score* will be the metric for comparison between the players. The players with lower results will be considered "stat-stuffers" whose statistical outbursts do not increase their team's chance to win. Players with higher results help to increase their team's chance to win due to their statistical outbursts. The *feature_importances* attribute will show what stats are more important for All-Star and Starter level players. With this knowledge, new All-Star/Starter models can be created that only use the most important features.

4 Experiments and Justification

4.1 Finding Important Variables

As stated previously, Decision Trees were constructed with data on player's classified as "All-Stars" and "Starters". The importance of the variables for the tree are recorded in Figure 16.

Statistic	Starters	All-Star
ORtg	0.066652	0.063120
DRtg	0.079895	0.065049
GmSc	0.043595	0.053426
MP	0.052812	0.062342
PTS	0.066330	0.067737
TS%	0.065155	0.064165
ORB%	0.047024	0.048669
DRB%	0.075715	0.063076
TRB%	0.074129	0.084044
AST%	0.108538	0.110549
STL%	0.092278	0.087273
BLK%	0.059503	0.073403
TOV%	0.080634	0.076504
USG%	0.087733	0.080635

Figure 16: This table shows the value of importance for all of the tested statistical categories. The results for both the “All-Star” and “Starter” tree are shown. The statistics highlighted in yellow are the most important statistics for Starters, the statistics highlighted in blue are the most important statistics for All-Stars, and the statistics highlighted in green are the shared statistics that are important for both Starters and All-Stars.

There are several variables that both trees distinguish as important. The common variables include *TRB%*, *AST%*, *STL%*, *TOV%*, and *USG%*. If we look at the top 8 traits for each tree, we can see that there are 3 differences between the 2 trees. The All-Star tree finds *PTS*, *MP*, and *BLK%* as important statistics, whereas the Starter tree finds *ORtg*, *DRtg*, and *DRB%* to be important. It is interesting that *USG%* is a common trait between the 2 trees. It makes sense that All-Stars value this trait, since All-Stars are usually one of the top 2 players on their teams; however, it appears that the amount a player is “used” on the court is important for Starters as well. *PTS* and *MP* prove to be more important, as expected, for an All-Star. Typically, an All-Star is one of the leading scorers on their team. That player also typically plays a large amount of minutes. Therefore, the more minutes a player spends on the court and the more points that player scores, the more likely the team is to win. The Starter tree favors more team-oriented statistics, as the overall performance of the team’s offense and defense are more important in determining the team’s chance to win.

4.2 Creating New Models

With the knowledge obtained from the models created above, I can create new models that focus on the 8 most important characteristics for both Starters and All-Stars. The results are presented in Figure 17.

Statistic	Starters	All-Star
ORtg	0.126644	-
DRtg	0.088067	-
MP	-	0.084473
PTS	-	0.151753
DRB%	0.110250	-
TRB%	0.126919	0.133230
AST%	0.157469	0.142062
STL%	0.117590	0.105271
BLK%	-	0.071229
TOV%	0.134536	0.150639
USG%	0.138521	0.161340

Figure 17: This table shows the value of importance for all of the selected statistical categories. The categories were chosen based on the top categories in Figure 16. The results for both the "All-Star" and "Starter" trees are shown.

After narrowing down the variables to the important characteristics, the defense-focused statistics become less and less important. *BLK%* is the least important variable for All-Stars, while *DRtg* follows suit and is the least important for Starters. Not only is the focus on offensive statistics evident, but one can also determine the difference between the team-focused Starter,

with $AST\%$ as the most important statistic, and the individual-focused All-Star, where $USG\%$ is the most important statistic.

4.3 The Lack of Defense

4.3.1 DRtg

As seen in Definition 4, the $DRtg$ statistic utilizes many different statistics in its formula, including basic player statistics (steals and blocks), team statistics (team steals, team blocks, and team defensive rebounds), and opposing team statistics (opponent offensive rebounds, opponent defensive rebounds, opponent turnovers, and opponent field goal attempts and makes). Besides $DRB\%$ and $STL\%$, the results in Figure 16 and Figure 17 demonstrate the lack of importance high statistical numbers in the these categories play in a team's chance to win. Since $DRtg$ is largely based on these unimportant defensive statistics, the $DRtg$ statistic itself has a low level of importance. The increased analysis of other defensive statistics could help the $DRtg$ statistic increase its value. Statistics that could be utilized in the $DRtg$ statistic or potentially be included in the model as separate variables include deflections and contested shots. Usually seen as points of emphasis by coaches, deflection and contested shots could help to better understand the true importance of a player and their impact on their team.

4.3.2 The Value of a Block

Amongst the common important statistics between All-Stars and Starters, *BLK%* is the only main box score statistic not present. The reason cannot be due to the heavy offensive statistical bias, as *STL%* is seen as important by both models. Taking a deeper look into the results of a blocked shot demonstrates why *STL%* is far more important than *BLK%*.

Typically blocked shots occur around the key within 5-8 feet of the hoop. This means that an offensive player has moved past his defender and is close to the hoop, which increases the likelihood of the shot going in. If a player is recording a large amount of blocked shots, it could be due to his team's inability to play good defense. Though yes, the player is doing his job and protecting the key from the opponent, the large amount of blocks could be pointing to a bigger problem in the team's lack of defensive prowess and effectiveness.

The end result of a blocked shot is not as certain as a steal. When a steal occurs, that player's team gains possession of the ball. When a block occurs, change of possession is not guaranteed. The ball is not in any team's possession after the block and the outcome is uncertain. Since many blocks occur around the basket, it is highly possible that the player who shot the ball regains possession and is able to quickly put up another shot attempt before the defender has a chance to react again. Due to the uncertainty of the play that leads to the block and of the end result of the block, it is

understandable why blocks are not valued as much as steals.

4.4 Player Tests

Now that the models are created and optimized, one can distinguish which player's high statistical performances affect their team's ability to win. A player's "fit" is determined by using the *score* function defined in Figure 14. A player with a higher fit means the player's performance has more value to a team's chance to win. Whereas a lower fit means that the player's performance has minimal impact on the team's chance to win.

Player	Starer Fit	All-Star Fit	PER
Russell Westbrook	0.5357	0.5649	27.9
Paul Millsap	0.5117	0.4814	19.8
Al Horford	0.5738	0.5805	19.0
Thaddeus Young	0.4703	0.5032	15.7
Nicolas Batum	0.5319	0.5319	14.8
Darren Collison	0.4921	0.5039	16.8
Paul George	0.4723	0.4893	19.9
Kyrie Irving	0.6115	0.5961	22.4
Klay Thompson	0.58	0.606	17.4
Giannis Antetokounmpo	0.5234	0.5276	24
Nikola Vucevic	0.4772	0.4469	22.7

Figure 18: This table shows the fit for players tested against both models. Along with the fit, the average *PER* for the players over the seasons in which they were tested is listed. Based on the players that were tested, the average All-Star fit was .5378 and the average Starter fit was .5302.

As seen above in Figure 18, a player with a high *PER* doesn't necessarily mean that the player's high statistical performances lead to a greater chance to win. Paul George and Paul Millsap have high *PERs* of nearly 20, but according to the low fit calculated by the models, their high statistical performances do not lead to wins. Nikola Vucevic also boasts a high *PER*,

followed by the lowest fit of all of the tested players. Vucevic is putting up big numbers, but his performances appear to have a minimal impact on his team's ability to win games. On the contrary, Klay Thompson, who has a *PER* of 17.4, greatly increases his team's chance to win when he performs better statistically. The same can be seen with Al Horford. However, the difference between Horford and Thompson is that Horford is known as a borderline All-Star player, whereas Thompson is a perennial All-Star.

4.4.1 Difference in Play Style

There is also a difference between the player's fit as an All-Star versus the player's fit as a Starter. Russell Westbrook, a perennial All-Star and former MVP (Most Valuable Player), is seen as a better fit to the All-Star branch than the Starter branch. This difference in results between the 2 models shows Westbrook's ability to fill up the stat sheet (which includes high *TRB%* and *AST%*) is not what gives his team the best chance to win. In order to win more games, Westbrook's team needs him to score more points, decrease his *TOV%*, and increase his *USG%*. It is interesting to see this case with Westbrook, since he is known for averaging a triple-double (which occurs when the player averages 10 or more in 3 different statistical categories).

Though there is not as large of a difference between the 2 trees, Kyrie Irving could also see his team improve via a change in focus. Irving is known as one of the best, if not the best, ball handler in the game today. He

has a tremendous ability to create his own shooting opportunity and score on any defense. However based on the models, instead of Irving looking for his own scoring opportunity, he should be attempting to create opportunities for his teammates. Even though he is one of the best ball handlers in the league, a decrease in $USG\%$ could be what his team needs.

4.5 Salary vs Data

With the fit values for all of the players that were tested against the models, further analysis can be done by comparing the results to the players' salaries.

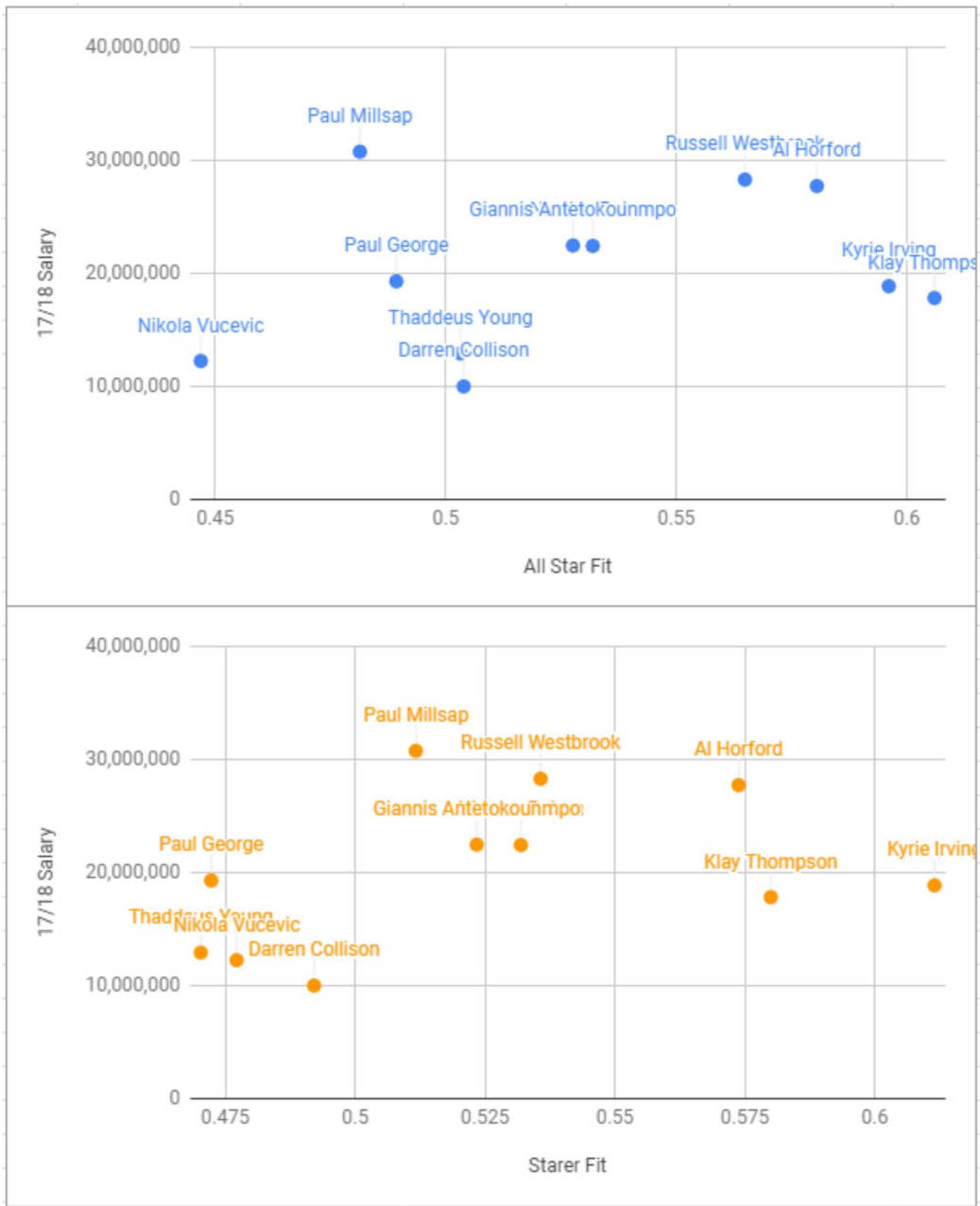


Figure 19: This table shows the fit for players tested against both models versus the salary the players had during the 2017-2018 season.

Figure 19 has the potential to show which players are deserving of the money they are making on their respective teams. For example, Al Horford and Paul Millsap both made nearly \$30,000,000 during the 2017/2018 season. However, according to the charts, Horford was far more effective in producing wins for his team when he put up large statistical numbers. Paul Millsap had the opposite effect on his team, as his high statistical performances do not appear to have a great effect on his team's ability to win. As stated previously, *PER* is a statistic that is commonly used in determining a player's level of play. How does *PER* compare to these players' salaries?

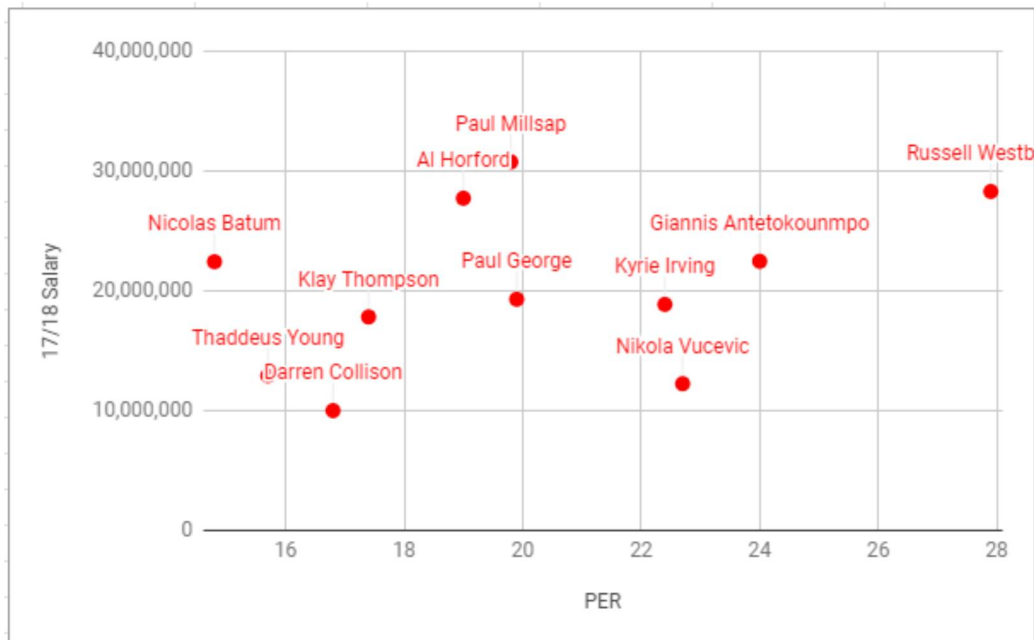


Figure 20: This table shows the *PER* for players tested against both models versus the salary the players had during the 2017-2018 season.

With the help of Figure 20, it can be seen that Horford's *PER* does not show his worth to his team. Though he does not has a high *PER*, it can be seen in Figure 19 that he can affect the game in a positive manner for his team when he is more involved. Other conclusions can be made by comparing the results between Figure 19 and Figure 20 as well. For example, Players like Nikola Vucevic, who have a high *PER* and have a low fit, may be better suited on another team. His current team should not look to re-sign the player in the off season, as he does not produce wins with his high numbers. But with his high *PER*, he could be used as a vital trade asset to his team. Vucevic has the potential to bring wins to another team with his play, but his high statistical performances are wasted in his current situation.

5 Conclusion and Future Work

Statistics do not tell us everything about an NBA player, but they can give us a better idea of what makes the best stand out from the rest. This research has shown that All-Star level players distinguish themselves from other players based on how many points they score and how much the player is being utilized on the court. Starter level players excel in their ability to increase their team's efficiency (as seen through *ORtg* and *DRTg*). Also, the lack of defensive statistics is evident, as only *STL%* is a dominant statistic for both categories of players. It would be interesting to insert deflections or some of the other "hustle" statistics into the tree; however, these statistics are not included in the average or advanced box scores. Better tracking and analysis of these numbers on a player by player basis may help to balance the heavily offensive focused frameworks. This may in turn give us better analysis of the players' abilities to help their team win.

The trees focus on the player's individual statistical performances, using the data to determine if their team will win or lose. However, due to numerous external variables that could affect a team's chance to win, it may be beneficial to add some weighted variables to the trees' calculations. These variables could include overall level of abilities of the player's team, the overall level of abilities of the opposing team, and the final score of the game. Such variables could increase the accuracy of the models.

Overall, the models are excellent additional tools for determining how

valuable a player is to a team's success. However, the models should not be the only tools used when attempting to determine a player's value. *PER* is a key statistic for determining a player's contribution on the court. For example, according to the trees, Paul George had a low fit. These results would conclude that George is not vital to his team's success; however analysts, scouts, and coaches would all agree that he is nothing less than that of an All-Star. Based on *PER* alone, he is a very efficient and useful player. These trees are much better suited for determining if a player who has a high *PER*, but playing on a bad team, is actually increasing his team's chance to win. If he does increase the team's chance to win, then GMs and coaches can use that data when they determine how much money to offer that player in a year of free agency. Thus, the models will become vital tools for GMs and coaches in evaluating current and perspective players.

References

- [1] Basketball Reference: Glossary. (n.d.). Retrieved from <https://www.basketball-reference.com/about/glossary.html>
- [2] Decision Trees. (n.d.). Retrieved from <https://scikit-learn.org/stable/modules/tree.html#tree>
- [3] Deshpande, S. K., and Jensen, S. T. (2016). Estimating an NBA player's impact on his team's chances of winning. *Journal of Quantitative Analysis in Sports*, 12(2), 51-72.
- [4] Gramacy, R. B., Jensen, S. T., and Taddy, M. (2013). Estimating Player Contribution in Hockey with Regularized Logistic Regression. *Journal of Quantitative Analysis in Sports*, 97-111.
- [5] Hollinger, J. (2005). Pro basketball forecast: 2005-2006. *Dulles, VA: Potomac*.
- [6] Knuth, D. E. (2011). *Selected papers on fun and games*. Stanford, CA: CSLI Publications.
- [7] Kubatko, J., Oliver, D., Pelton, K., and Rosenbaum, D. T. (2007). A starting point for analyzing basketball statistics. *Journal of Quantitative Analysis in Sports*, 3 (3).

- [8] Maese, R. (2013, October 25). NBA embraces advanced analytics as Moneyball movement sweeps pro basketball. Retrieved from <https://www.washingtonpost.com/sports/wizards/nba-embraces-advanced-analytics-as-moneyball-movement-sweeps-pro-basketball/>
- [9] Mills, E. G., and Mills, H. D. (1970). *Player win averages: a computer guide to winning baseball players*. AS Barnes.
- [10] Morris, B. (2014, March 25). The Hidden Value of the NBA Steal. Retrieved from <https://fivethirtyeight.com/features/the-hidden-value-of-the-nba-steal/>
- [11] NumPy. (n.d.). Retrieved from <https://www.numpy.org/>
- [12] Oliver, D. (2004). *Basketball on paper: rules and tools for performance analysis*. Potomac Books, Inc.
- [13] PyCharm: The Python IDE for Professional Developers by JetBrains. (n.d.). Retrieved from <https://www.jetbrains.com/pycharm/>
- [14] Raschka, S. (2015). *Python machine learning*. Packt Publishing Ltd.
- [15] Rosenbaum, D. (2004). Measuring How NBA Players Help Their Teams Win.
- [16] Solieman, O. K. (2006). Data mining in sports: A research overview. *Dept. of Management Information Systems*.

- [17] The Next Way of Seeing Things. (n.d.). Retrieved from <https://www.secondspectrum.com/index.html>
- [18] The pandas Project. (n.d.). Retrieved from <https://pandas.pydata.org/about.html>
- [19] Ugur, S. (2018). Analytics Movement. Retrieved from <https://www.nbastuffer.com/analytics101/nba-analytics-movement/>
- [20] What is Python? (n.d.). Retrieved from <https://www.pythonforbeginners.com/learn-python/what-is-python/>
- [21] Yang, Y. S. (2015). Predicting Regular Season Results of NBA Teams Based on Regression Analysis of Common Basketball Statistics. *Undergraduate Thesis, UC Berkeley.*