

School of Mathematics and Applied Statistics

MTST399 Capstone Project:

The random forest algorithm for NBA win
predictions

Nicholas Hedley

Abstract

This project endeavours to harness the predictive capabilities of the random forest algorithm with independent variables in player data to forecast win outcomes in the National Basketball Association. Focused primarily on the realm of sports data science, specifically in predicting the success or failure of a team, this research employs an extensive dataset encompassing a multitude of player statistics. The main goal of this project is to ultimately develop a robust predictive model capable of discerning the intricate relationships between player performance metrics and overall team success. Through analysis and model refinement, the project seeks to contribute insights into the key determinants influencing game outcomes. The findings have the potential to enhance decision-making processes in team management, aiding coaches, analysts, and stakeholders in optimising strategies via lineup decisions or training adjustments. Overall, this project will provide a base model where further player prediction methods can be applied to this, for more accurate predictions. This project also acts as an extension of my personal learning experience from the pathway subject STAT301: introduction to data science. Testing my knowledge of the subject and pushing me to learn more about the field, especially the random forest algorithm. Finally, this project connects to the broader field of sports analytics showcasing the applicability of machine learning methods in extracting useful information from data sets.

Acknowledgements

I, Nicholas Hedley, declare that this report, submitted in fulfillment for the capstone report for MATH399 is to the best of knowledge my own work and contains no material previously published or written by another person except where due reference is made in the project itself. Any contribution made to the project by others, with whom I have worked at UOW or elsewhere, is explicitly acknowledged in the project. I also declare that the intellectual content of this project is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.

I want to acknowledge and thank, Prof Andrew Zammit-Mangion for his assistance and guidance in the development of this project. Thank you to Prof Glen Wheeler, Prof Mark Nelson and Prof Carole Birrell for providing me with feedback throughout the semester and teaching me the skills necessary to create this project. I appreciate the work and dedication you have put into your craft to ensure that your students have the best opportunity to learn.

Contents

1	Introduction	1
1.1	Overview	4
2	Literature review	6
2.0.1	Research question	6
2.0.2	Introduction	6
2.0.3	Main	8
2.0.4	Conclusion	14
3	Main Body	16
3.1	The Random Forest Algorithm	16
3.1.1	Decision Trees	17
3.1.2	Bagging process	19
3.2	Measures of effectiveness	21
3.3	Statistics in the NBA	22
3.3.1	Simple Statistics	22
3.3.2	Advanced statistics	24
3.4	Carrying out the project	26
3.4.1	The data	26
3.4.2	Fitting the models	28

<i>CONTENTS</i>	iii
3.4.3 Analysing the models	30
3.4.4 Prediction methods	31
3.4.5 Summary	33
4 Conclusions	35
4.1 Bibliography	37
A R-code	41

List of Figures

3.1	Example data set	17
3.2	Decision tree on x_0 and x_1	18
3.3	Decision tree on x_2	19
3.4	Bootstrap samples	19
3.5	Random Forest with 4 decision trees	20
3.6	Random test observation	20
3.7	Random test observation	30

Chapter 1

Introduction

The capstone project is set in the dynamic landscape of the National Basketball Association (NBA), where data-driven decision-making has become increasingly integral to improving team performance. Specifically, the project focuses on leveraging analytics in player data and utilising the power of the random forest algorithm to predict team outcomes using player data. The project's scope is limited to the extensive landscape of NBA player statistics, with a primary focus on predicting team win outcomes using player data. The analysis spans various facets of player performance, investigating the interplay between independent player variables and team success. It involves identifying key performance metrics, exploring non-linear relationships, and developing a predictive model tailored to the unique dynamics of NBA gameplay. This project will look at how given player data can be used to predict wins. It does not encompass a framework for player statistic prediction, which would be a great extension to this project in the future for further prediction refinement.

The project unfolds through several key stages. It commences with a

thematic review of literature, exploring the junction of statistical theory and sports analytics within the NBA and the application of machine learning techniques in predicting team outcomes. Subsequently, the project involves data collection, preprocessing, and feature engineering to prepare the NBA player dataset for analysis. The core of the project involves implementing the random forest algorithm, model training, and rigorous evaluation. The findings are then interpreted, and actionable insights derived from the analysis are discussed. The project concludes with reflections on the implications of the results and potential applications in NBA team management. The primary aim is to construct a predictive model tailored to NBA dynamics that accurately forecasts team win outcomes based on player data. This encompasses the identification of NBA-specific performance metrics and the development of a robust model. The content aims to contribute specifically to the field of NBA sports analytics, offering insights into player dynamics which lead to team success.

The project aims to provide the reader with an advanced understanding of the application of machine learning, particularly the random forest algorithm, in the context of the NBA. Readers will gain insights into the nuanced relationships within NBA player data, the challenges specific to basketball analytics, and the potential impact on team decision-making. The project aims to provide readers with knowledge prerequisite to roles in NBA analytics and sports management. The project is designed to be adaptable, allowing for iterative refinement based on more complex player prediction models and any further research in the field of sports statistics.

To fully grasp the ideas and discussions of the report, readers must have sound statistics knowledge. Content from STAT101, STAT201 and STAT202

provides a solid general understanding of the statistics and techniques used in this report. Hypothesis testing, model fitting and model testing are key aspects of these subjects that are considered pre-requisite for this project. Additionally, an understanding of the game of basketball and basic knowledge of the performance indicators are necessary to comprehend the report. Reading and understanding the R code can be a difficult task if one is unfamiliar with R and even more so if the reader has no programming experience, so it would be advisable for the reader to have a background in programming from any CSIT, CSCI or STAT 200-level subjects. The random forest algorithm or any advanced machine learning techniques is not declared prior knowledge and will be defined in the relevant components of the report. For basketball terminology, I will provide concise definitions and provide context as to why this statistic is relevant.

I picked this as my capstone project for a multitude of reasons. The first is that I thoroughly enjoy data science and am set on taking on a career in this field in the future. Drawing inferences from the large scales of data streams, in my mind, is highly logical and can lead to intriguing discoveries or confirm intuitive thinking. In a world where the collection of data has become so widespread to match such advancement in technology, we should strive to use this data as often as possible. Thus, as a society, we will become more informed about the factors around us and can channel a higher level of comprehensive analysis into the decisions we make. Another reason I chose this as my project is my passion and enthusiasm for the NBA and the basketball world. I have been following the NBA since around 2014 and have been playing basketball since 9 years old. Now in 2023 and being 22, I still cherish both following the league and playing competitively for the university. This project allows me to combine two areas I am passionate

about and create a project which interests me and broadens my knowledge of the NBA and statistics in the subclass of data science.

1.1 Overview

We begin with a Literature Review. It dives into the intricacies of data manipulation in R, emphasising the critical role in preparing NBA player data for modelling. It explores the statistical theory underpinning the random forest algorithm, drawing from seminal works and contemporary perspectives. Then, examines the application and testing of various predictive methods within the context of the NBA. And finally, focuses on the implementation of the random forest algorithm in the unique context of the NBA, addressing challenges and opportunities. We then move on to the main body of the report. First, it provides a detailed explanation of decision trees, a key component of the random forest algorithm. We, explore the bagging process, showcasing how bootstrap sampling and random feature selection contribute to the algorithm's effectiveness. Additionally, we discuss the significance of the 'random' and 'forest' elements in the algorithm, addressing issues like overfitting. Stemming from this, we review some measures of effectiveness and deterministic properties that aid our model selection process. In this section, we also introduce fundamental NBA statistics, and advanced statistics, offering a more sophisticated approach to analysing player performance. After explaining all this theory, we apply our preparation to the 2022-2023 NBA season data set. Executing the model-building and prediction process in a logical and succinct fashion. Lastly, we end with a conclusion that Summarises the aims of the project, highlighting the exploration of the random forest algorithm in predicting NBA game outcomes. Discusses the journey

throughout the project, combining a passion for data science and basketball, and concludes with a statement on the intersection of mathematics, statistics, and sports analytics.

Chapter 2

Literature review

2.0.1 Research question

Using the random forest algorithm with independent variables in player data to predict win outcome

2.0.2 Introduction

The convergence of advanced statistical techniques and sports analytics has paved the way for a new era in the interpretation and prediction of sports outcomes. In the world of professional basketball, the National Basketball Association (NBA), leveraging an extensive and seemingly endless array of player data, has become a pinnacle for the application of cutting-edge statistical technique. Among these methodologies, the random forest algorithm emerges as a formidable tool in the endeavour to discover the metrics influencing game results. This literature review delves into the intricate landscape of employing the random forest algorithm within the realm of NBA analytics, highlighting its multifaceted components and potential applications. Divided

into four sections, this review dissects the aspects of this approach, aiming to provide an overview for analysts, statisticians, and basketball enthusiasts alike.

The first section details the art of data manipulation using the popular statistical computing language R. Emphasising and defining the critical role of data processing in preparing NBA player data for effective modelling. Building upon this foundation, the following section explores the statistical theory underpinning the random forest algorithm and surrounding information, explaining its core principles and mechanisms

Moving from theory to practice, the third section examines the application and testing of other predictive methods within the context of the NBA. This section discusses the empirical results and outcomes of applying statistical techniques to real-world NBA data sets, assessing its predictive accuracy and interpretation in the potential for actionable insights.

Finally, the fourth section brings the theory and application together, focusing on the implementation of how the random forest algorithm can be used in the unique context of the NBA. It explores the challenges and opportunities of harnessing player data to predict game outcomes, providing a practical perspective on the algorithm's utility within this dynamic sports domain.

2.0.3 Main

Data Manipulation techniques

The foundation of any predictive analysis rests upon the preparation and manipulation of data. In the context of utilising the random forest algorithm to predict NBA game outcomes, becoming familiar with data manipulation techniques in R becomes key. Two sources are examined and used to teach me through this process. Both sources have a slightly different definition of data manipulation. In essence, data manipulation is the process in which raw data is transformed into clean data, ready for analysis and statistical model building. Data manipulation is a critical step in my project, as I must properly prepare the NBA player data before I can run certain R commands on it.

The first source, [Mailund, 2017], offers a pragmatic approach to data manipulation in R. This resource equips practitioners with practical insights and step-by-step guidance, making it a valuable companion for those seeking to navigate data manipulation. Its accessible language and well-defined examples facilitate a smoother transition from theory to practice, ensuring that data manipulation becomes an actionable skillset. Although the examples are easy to understand and the definitions are conducted with far more vernacular language than other sources in the field, it sacrifices theoretical motive for ease of comprehension. Telling you more how to perform methods rather than explaining why you would incorporate them.

Conversely, the second source, [Speegle & Clair, 2021], provides a more formal and theoretical foundation for data manipulation in R. This text

delves into the underlying principles and methodologies, educating from a more theoretical stance on the mechanics of data transformation. While its tone may be more scholarly, its depth of coverage equips analysts with a deeper understanding of the theoretical notions that harbour data manipulation techniques in R. Despite the detail the textbook goes into, it is far less practical than [Mailund, 2017] as often the sheer amount of content in the book acts as a barrier for learning. Without the sufficient background knowledge or another source to assist with the reading, learning the content from this textbook can prove quite the challenge.

These two distinct sources complement each other, offering a well-rounded perspective on data manipulation in R. The practical and simplistic insights from [Mailund, 2017] and the theoretical aspects of [Speegle & Clair, 2021] combine to provide a comprehensive toolkit for preparing NBA player data for predictive modelling in my project.

Statistical Theory

The foundation of our exploration into the random forest algorithms' predictive potential in the context of NBA game outcomes necessitates a critical examination of its statistical underpinnings. Breiman's seminal work in 2001 [Breiman, 2001] undoubtedly marked a pivotal moment in the history of machine learning. However, it's crucial to acknowledge the inherent complexity of this source. Breiman's pioneering efforts, although visionary for their time, are inherently dense and may pose as a challenge to those not well-versed in the intricacies of ensemble learning.

Contrasting to Breiman's work, [Schonlau & You, 2020] offers a more contemporary lens through which to view the random forest algorithm. While it strives to bridge the gap between previous theory and modern practice, it too is not without its challenges. Its depth of coverage may appear daunting, demanding a significant commitment to grasp the nature of applying random forest in today's data-rich landscape. This underlines the evolving nature of machine learning and the ongoing need for analysts to remain attentive to the latest advancements. As a complementary resource, [Rennert, 2018], designed as an introductory lecture, has the merit of accessibility. However, it is key to recognise its limitations in terms of depth. While it provides foundational knowledge, it may not suffice for those seeking a deeper understanding of the statistical theory underpinning the algorithm.

The random forest algorithm is a way of picking the best predictors for a model. In other words, choosing the best combination of independent variables quantifies the most variation in the dependent variable. To very simply and abusively describe the r.f algorithm it achieves this through taking a bootstrap of the data and then randomly selecting different combinations of predictors based on each bootstrap. This process is referred to as bagging the data. Creating decision trees with our bagged data, and comparing how they do to our full data set, will determine which predictor combinations are the best to build the model.

Adding a layer of nuance to our understanding, [Couronne, Probst & Boulesteix, 2018] introduce a comparative perspective, contrasting Random Forest with logistic regression within prediction theory. Logistic regression can be thought of as another technique to create a model from the data. It is particularly useful and relevant as it will give a number between 0 and 1,

which is perfect for quantifying the probability of a win in the NBA data context. The source highlights an essential criterion for algorithm selection—the presence or absence of contextual knowledge regarding predictors’ impact on the prediction variable. This argument outlines the situational utility of random forest, particularly in scenarios where we know almost nothing about which predictors are relevant to the independent variable.

Application and testing of statistical theory

In the realm of applying statistical theory to predict NBA game outcomes, two distinctive sources offer insights into different facets of the process. [Horvat, Logozar & Livada, 2023] highlights the significance of a wide and diverse dataset as the foundation for building effective prediction models. The emphasis here lies in the richness of the data itself, highlighting that the breadth and diversity of information can be instrumental in improving predictive accuracy. However, it is essential to assess this approach, recognising that data breadth alone may not guarantee predictive success. The quality, relevance, and suitability of variables within the data set also play a pivotal role in model performance.

On another front, [Zhao, Du, Tan, 2023] delves into a spectrum of machine learning and model-building techniques applied to NBA game prediction. This source takes a broader view, exploring various methodologies beyond the confines of random forest such as LASSO, logistic regression and graph neural networks. While the exploration of diverse techniques is valuable, it too necessitates analysis. It is crucial to consider the applicability of these techniques in the context of NBA game outcome prediction. Additionally,

it's worth noting that both sources focus their predictions on implementing team data, rather than player data. In fact, across a variety of reading in the area, it appears that there is a large absence of model development in using player data to predict team win percentage over team data. Using a player data model is far more applicable and useful in different situations with the context of the NBA. Teams are ever changing in a dynamic trading and signing market. Using a model which adjusts for new player acquisitions may prove to perform far better in different situations than its team data based counterparts. This is gap in exploration, is one of the major reasonings I adapted my research objective to focus on player data.

In summary, this section showcases two distinct approaches to applying and testing statistical theory in NBA game outcome prediction. While the diversity of data and methodologies presented in these sources offers valuable insights, critical analysis is essential. I believe using a combination of a broad data and different techniques is the key to reaching a repeatable, accurate and theoretically stimulating project. The practical effectiveness and suitability of these approaches within the specific context of NBA games warrant careful consideration, especially as they focus on team data rather than player data

Implementation in an NBA context

To make sense of NBA data in a practical way, we turn to two sources. The first source, [NBA advanced stats, 2023], provides a helpful NBA glossary. This glossary explains what key statistics and advanced NBA numbers mean. It helps us understand the numbers behind the game, but we need to think about how to use this knowledge when predicting game outcomes.

The second source, [Song, Gao & Shi, 2020], looks at real-time predictions for team scoring using betting odds. It simplifies things by figuring out how many points each team is expected to score and uses that to pick a potential game-winner. This approach makes things easier, but we need to consider how reliable betting odds are when it comes to predicting the unpredictable nature of NBA games.

Knowing the context helps identify which predictors or features are meaningful for the prediction. In NBA game prediction, it's vital to observe the significance of statistics such as points per game, field goal percentage, and turnovers concerning game outcomes. Without context, we might include irrelevant predictors, leading to a less effective model. Although the random forest algorithm will sort some of these out, if we include too many irrelevant predictors, the process will become overly computationally expensive. Even more so, the data set obtained is enormous, therefore I must have an extremely thorough understanding of each statistic. It will allow me to swiftly and easily spot mistakes in prediction or mistakes in predictors. Furthermore, it will assist me in making educated assumptions about which predictors contribute and detract from a team's win. However, this is a difficult balance, as I must leave enough predictors in so that the r.f algorithm can effectively do its job and not be limited to my own possible bias constraints. [Song, Gao & Shi, 2020] also opened me up to alternative dependent variables which win percentage can be easily acquitted from, in their paper they create predictions and confidence intervals or team scores, which I believe through an intermediate and preliminary statistical technique you can calculate win percentage.

In essence, the random forest algorithms' effectiveness in predicting NBA

game outcomes hinges on the quality and relevance of the data it uses. Understanding the context of the data is fundamental in ensuring that the predictors chosen are meaningful and that the models' predictions align with the subtleties of the NBA environment.

2.0.4 Conclusion

This review journeyed through the world of NBA game prediction using the random forest algorithm, offering insights for analysts and basketball enthusiasts. We began by learning the practical side of data manipulation in R, balancing practical skills with theoretical depth through [Mailund, 2017] and [Speegle & Clair, 2021]. Moving into the theory, we explored the foundations of random forest, acknowledging its complexity in [Breiman, 2001], and finding a bridge to modern practice with [Schonlau & Zou, 2020]. We delved into applying these theories to NBA games, highlighting the value of diverse datasets in [Horvat, Job, Logozaar & Livada, 2023], and embracing various machine learning techniques with [Zhao, Du, Tan, 2023]. Also pointing out the missing exploration in the field of using player data to predict team outcomes as opposed to team data. Finally, we implemented these theories in an NBA context, using an NBA glossary [NBA advanced stats, 2023], and considering real-time predictions based on betting odds [Song, Gao & Shi, 2020]. Throughout, the importance of context in selecting predictors and evaluating data was clear.

In summary, this review equips researches with practical and theoretical tools, emphasises the consideration for situational algorithm choices, and denotes the importance of understanding the NBA context in predictive analytics. It opens the door and provides my own extension for further exploration

at the intersection of statistics and sports.

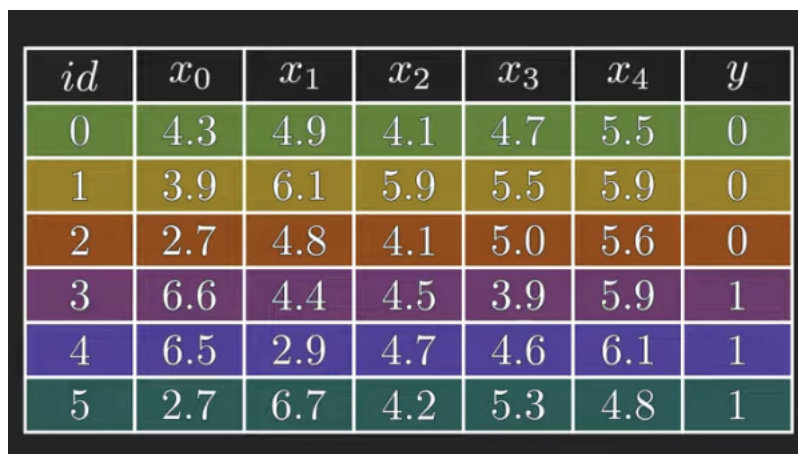
Chapter 3

Main Body

3.1 The Random Forest Algorithm

The random forest algorithm is a supervised machine-learning method which can be used for classification and regression. We will first explore how it works and then move on to why it fits well into the project and the course of STAT301.

It is best to understand the random forest with an example. So let us use Figure 3.1 as our data set for the example.



id	x_0	x_1	x_2	x_3	x_4	y
0	4.3	4.9	4.1	4.7	5.5	0
1	3.9	6.1	5.9	5.5	5.9	0
2	2.7	4.8	4.1	5.0	5.6	0
3	6.6	4.4	4.5	3.9	5.9	1
4	6.5	2.9	4.7	4.6	6.1	1
5	2.7	6.7	4.2	5.3	4.8	1

Figure 3.1: Example data set

3.1.1 Decision Trees

Decision trees are covered after week 5 in the STAT301 course and are a key concept in random forests. A decision tree is also a supervised machine-learning algorithm used for both classification and regression tasks. It recursively splits the dataset into subsets based on the most significant attribute (or specified attributes), creating a tree-like structure where each internal node represents a decision based on a feature, each branch represents an outcome of that decision, and each leaf node represents the final predicted class or value. The algorithm first begins at the root node, which contains the entire data set. The algorithm evaluates each attribute (feature/variable) in the dataset to determine the one that provides the best split. The attribute that results in the most homogenous subsets (purest) is chosen as the splitting criterion. Common measures include Gini impurity for classification and mean squared error for regression. The dataset is split into subsets based on the chosen attribute. Each subset corresponds to a unique value or range of values for the selected attribute. The above steps are recursively applied to

each subset, treating it as a new dataset. This recursive process continues until one of the stopping criteria is met usually, this is meeting a leaf node. For our data above data-set in Figure 3.1, we can make a decision tree based on the features x_0 and x_1 which would look like Figure 3.2

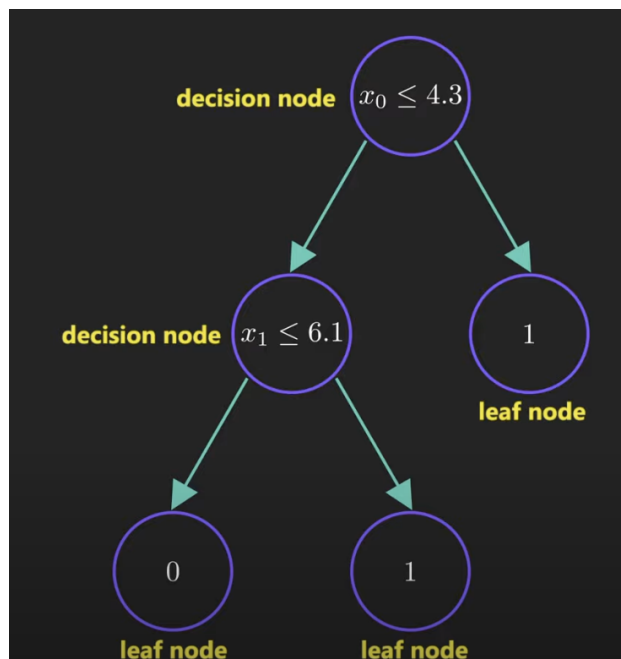
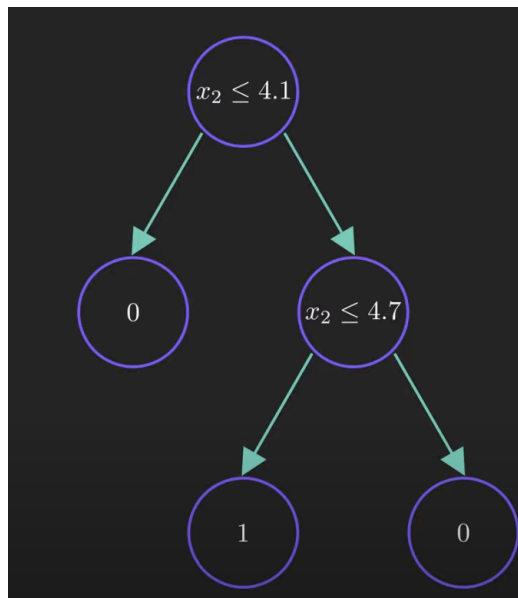


Figure 3.2: Decision tree on x_0 and x_1

Decision trees alone can often have their own drawbacks, for instance, they are prone to overfitting when a tree gets too deep or long. They begin to fit parameters which are specific to the training set rather than creating a rule or generalisation of where that data point should be categorised. This leads to high variance and in the classification case, a high misclassification rate. They also rely very heavily on the variables initially selected, for example. We could instead fit the tree with just the x_2 variable in mind and obtain the tree which looks like Figure 3.3

Figure 3.3: Decision tree on x_2

3.1.2 Bagging process

The first process done in random forest is the bagging of data. First, a bootstrap with replacement is performed on the data set. This is a preliminary concept from STAT202 and will not be explained. After the bootstrap is completed in our example we might end up with the data in the following samples shown in Figure 3.4:

<i>id</i>	<i>id</i>	<i>id</i>	<i>id</i>
2	2	4	3
0	1	1	3
2	3	3	2
4	1	0	5
5	4	0	1
5	4	2	2

Figure 3.4: Bootstrap samples

Next, the features are randomly selected for each sample, we specify that only 2 features are to be selected. Then we create decision trees based on each bootstrap sample and the specified features.

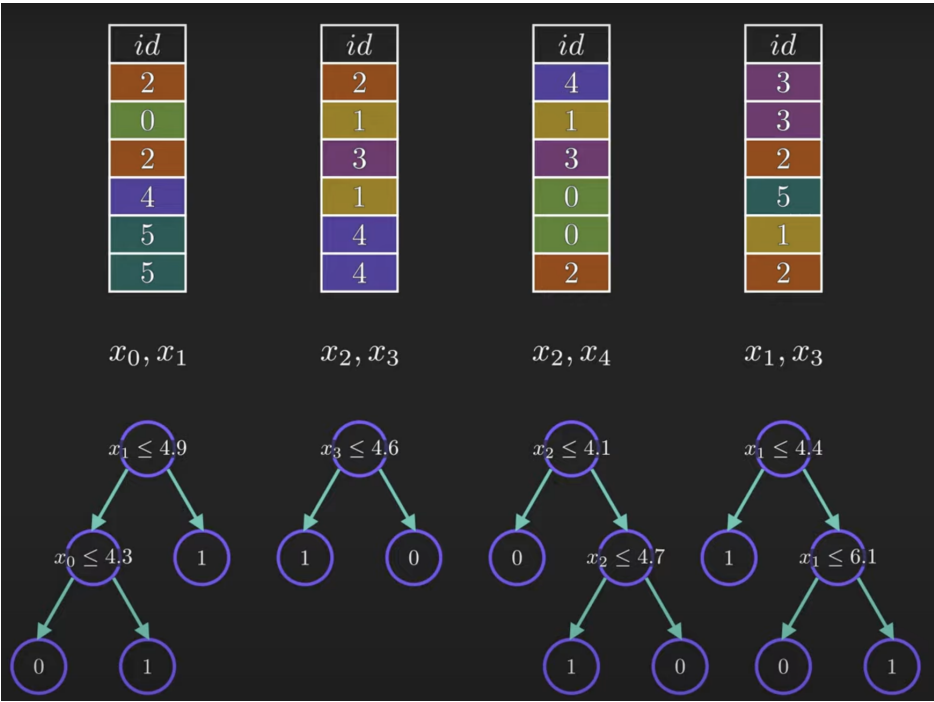


Figure 3.5: Random Forest with 4 decision trees

Note that Figure 3.5 provides an example of the random forest algorithm with 4 trees. Once this has been completed, a new data point, this will be our test data point with y unknown.



Figure 3.6: Random test observation

We classify this data point using all 4 decision trees. Doing this, we obtain a 1, 0, 1 and 1 going through the decision trees from left to right. As this is a classification problem, we choose the majority decision, so we

would classify the data point as 1. This process of combining the results of multiple models is called aggregation. When we perform the random forest algorithm, we are taking the bootstrap and the aggregation, we refer to the combination of these two methods as bagging.

Now, it is evident that the ‘random’ in the random forest comes from the randomness of the bootstrap and the random feature selection. The forest component comes from the creation of multiple decision trees, as a collection of trees makes a forest.

STAT301 in the latter weeks explores classifying methods and decision trees. Random forests are an advanced extension of decision trees which by nature alleviate many of the drawbacks such as the tendency to over-fit. Recall the bootstrapping we used at the beginning of the process, by performing such a method, the model is far less sensitive to the training data, slightly increasing the bias but drastically shrinking the variance. Theoretically, this leads to a lower miss-classification rate.

The random forest is an excellent choice to model player data to win outcomes as we are modelling a classification problem with features we are uncertain as to how they relate to the prediction variable.

3.2 Measures of effectiveness

The measures of effectiveness are covered in the preliminary coursework of STAT301. We will be using, accuracy, miss-classification rate, true positive rate, true negative rate and hypothesis testing.

3.3 Statistics in the NBA

NBA statistics refer to a set of quantitative measures and metrics used to assess and analyse the performance of players. These statistics cover various aspects of the game, providing insights into player abilities, team dynamics, and overall trends within the league. We will define the simple statistics used in this project and what they indicate. All descriptions information obtained via [NBA advanced stats,2023]

3.3.1 Simple Statistics

Points (pts)

The number of total points scored by that player. Scoring more points than the other team wins points is a direct stat of winning.

Assists (ast)

An assist is counted when a player scores a basket from a teammate's pass. The teammate who passed the ball receives the assist. This stat is a good indicator of teamwork and reflects players who are good at setting up their teammates to score.

Rebounds (reb)

Rebounds are counted when a field goal attempt is missed and a player grabs possession of the basketball after bounces off the rim. There are defensive and offensive rebounds, indicating the possession of the player before they grabbed the rebound. TREB stands for total rebounds, and what we will be using in this report. Rebounds are important to a team's win as they

provide more possessions for the winning team, teams with more possessions attempt more shots and as a result, are likely to score more points and win.

Steals (stl)

Steals are a defensive statistic which occurs when a player grabs the ball from the other team and maintains possession of it. As with rebounds, they provide more possession to the team which stole the ball while also taking away a possession from the other team, more steals result in more possessions relative to the other team

Blocks (blk)

A block occurs when a defensive player alters the shot attempt of an opponent by thwarting the ball right after it has left the opponent's hand. Blocks are a defensive statistic which ensures that the opponent's shot is missing, more blocks means the opposing team missed more shots.

Turnovers (tov)

A turnover occurs when a player mistakenly loses possession of the basketball to the opposing team. It is a negative statistic, that is, players with higher turnovers are worse for the team as they lead to fewer possessions.

Field goals

Field goal attempts (FGA) are the number of attempted shots at the basket. Field goals made (FGM) is the number of shot attempts a player has successfully converted. FGM/FGA results in the field goal percentage (FG%), the higher the better. Typically the average NBA FG% sits between 0.46 and 0.47.

3-pointers

3-pointers are a special type of field goal scored when the player makes a basket beyond the three-point line. We have similar stats for these 3PA, 3PM and 3P%. Typically, scoring 3-pointers is more difficult than 2-pointers and hence sits a lower percentage on average at around 33%. 3 point attempts do count as field goal attempts.

Free throws

A player may have the opportunity to shoot free throws throughout the game. Same as the field goal statistics, we have FTA, FTM and FT%. A free throw does not count as a field goal.

3.3.2 Advanced statistics

Often, simple statistics provide a very basic overview of player performance[Tiodorovic, 2022]. Yes, players who score more points may seem like they contribute more to a win, but the game of basketball has evolved far further than this. A player's points are not always an indicator of a good performance, assists, rebounds and all the other simple statistics can provide a better notion of player performance. Yet, in the realm of statistics, these simple indicators of performance lack the inclusion of complex interactions between them. This lack of depth would hinder the model, as it would not possess enough relevant information to create a model which produces reasonable inference. Thus, advanced statistics are introduced. Advanced statistics provide a more sophisticated and data-driven approach to analysing basketball performance. They offer a deeper understanding of the game, inform strategic decisions, and contribute to a more holistic assessment of player and team contributions

on the court

Advanced statistics are complex metrics of player performance which are obtained through calculations of other statistics. In this project, we will observe 3 advanced player statistics and implement them into the model. All formula were obtained via [Tiodorovic, 2022]

Effective field goal percentage (eFG%)

Effective field goal percentage shows how efficient the scorer is based on the fact that perimeter shots are worth three points. It can be calculated via the Equation 3.1.

$$eFG\% = \frac{FGM + (0.5 \times 3PM)}{FGA} \quad (3.1)$$

By multiplying the number of 3-pointers made and adding it to the field goals made, we are accounting for the 3-pointer being worth more than regular shots. By analysing effective field goal percentage, coaches can determine whether or not a player is hurting the team with poor shooting factoring in the fact that 3-pointers are made at a lower rate than regular shots. This stat has become particularly useful in modern basketball where the 3-point shot is attempted far more than ever before.

True Shooting percentage (TS%)

True shooting percentage is a statistic that measures the players' shooting efficiency with the inclusion of factoring in their free throws. This statistic includes all ways a player can score for their team and is calculated via Equation 3.2

$$TS\% = \frac{Pts}{2(FGA + (0.44 \times FTA))} \quad (3.2)$$

Factoring in the free throws is an important distinction as it is regarded as the most efficient shot in basketball. This stat benefits players who have good percentages from the line and get fouled more frequently. Getting to the foul line has been very important in consistently winning games, explaining why true shooting percentage is an integral part of modern basketball statistics.

Efficiency (Eff)

Efficiency is an all-encompassing statistic, commonly regarded as the most significant stat in basketball. Although it does not highlight every aspect of the game, it accounts for enough to accurately derive how beneficial a player has been to their team. Shots made are given certain values based upon average shooting percentages and positive stats add on to it, while shots missed and turnovers give a player a negative score. It can be calculated with the following formula in Equation 3.3

$$Eff = (Pts + Reb + Ast + Stl + Blk - FG_{missed} - FT_{missed} - Tov) \quad (3.3)$$

To keep the scope of this project manageable, we will only be using these three advanced statistics.

3.4 Carrying out the project

3.4.1 The data

The NBA dataset we have obtained contains an exhaustive list of player data, containing every player statistic for each game over the 2022-2023 NBA regular season. The NBA consists of 30 teams and 82 games in the regular season. Each team will at a minimum play every other team twice throughout

the season, a game at their home arena and a game at the other team's home arena. The data has been loaded into R via a package called "nbastatR". An NBA team can consist of up to 15 players, yet, the data only includes observations of players with minutes greater than 0. Across all the players and all the games, there are 25,895 observations. Despite this, the data contains 58 different features, most of which are irrelevant and do not include any advanced statistics. Hence the data manipulation and pre-processing must be completed before any model fitting can be conducted.

The first cleaned dataset, cut the variables from 58 to 7 containing the simple statistics of an unknown player, the statistics were

- Points
- Total rebounds
- Assists
- Turnovers
- The game outcome
- Steals
- Blocks

It does not relate players to teams, and they players names are left out so there is no player bias in this data set.

The second cleaned dataset contains all the individual players advanced statistics mentioned earlier in the report and the simple statistics mentioned previously.

The third cleaned dataset contains just the advanced statistics The fourth cleaned data set contains the summation of the efficiency and the averages of the true shooting percentage and effective field goal percentage for a team for that game. Appendix A contains the R code for all these data sets, and how they were obtained. So, the data has been cleaned and 4 different datasets have been obtained, now we move on to fitting the r.f models.

3.4.2 Fitting the models

All r.f models were fitted with 500 trees and 2 variables tried at each split.

Model 1: Simple statistics

First the random forest algorithm on the 1st data set, unsurprisingly, the out-of-bag estimated error rate is 47.13%, a true negative rate 0.6095 and a true positive rate of 0.4144 . Essentially, our data model is marginally better at predicting win outcomes than a coin flip. Therefore, clearly this model is a poor choice for the modelling of win outcome

Model 2: Advanced statistics & simple statistics

The random forest was fit to the second data set containing both simple and advanced player statistics. It too proved to be a poor fit, with an OOB estimated error rate of 45.62% combined with a TNR of 0.5548 and a TPR of 0.5326.

Model 3: Advanced statistics

The random forest model was fit to the third data set containing just the advanced statistics. It produced an estimated error of 45.86% slightly higher

than the previous model but still just as bad. However, it produced a higher sensitivity of 0.6395 and a lower specificity of 0.4534.

Model 4: Total advanced statistics

The fourth data set was implemented into the random forest algorithm. Containing all the advanced statistics as a result of the players performance. The OOB estimate of error rate was 23.13% coupled with a healthy TNR of 0.7659 and a TPR of 0.7715. Thus far this is the best fit we have made.

Model 5: Logistic Regression

Logistic regression is prerequisite content from STAT332 and can be used to model a binomial variable like winning and losing. As with model 4, I utilised the fourth data set. I included logistic regression to test variable significance and observe how the logistic classification would compare to that of the random forest. The logistic model fitted would be in the form

$$p_1(X) = \frac{\exp(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3)}{1 + \exp(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3)}$$

Interestingly, using the hypothesis to determine significance in coefficients

$$H_0 : \beta_1 = \beta_2 = \beta_3 = 0 \text{ vs } H_1 : \text{at least one } \beta_i \neq 0 \text{ for } i = 1, 2, 3$$

at an alpha level of 0.05, determined that only total player efficiency was significant in predicting win outcomes.

So that lead onto the 6th and 7th models which produced the best models as they accounted for the lowest estimated error.

Model 6: Logistic regression and total efficiency

A logistic regression model was fit to the data using only total efficiency as the predictor.

$$p_1(X) = \frac{\exp(\beta_0 + \beta_1 x_1)}{1 + \exp(\beta_0 + \beta_1 x_1)}$$

it produced the estimated model:

$$p_1(\hat{X}) = \frac{\exp(-1.4379702+0.0148511x)}{1+\exp(-1.4379702+0.0148511x)}$$

Model 7: Random forest efficiency model

The final model was the random forest model with just the total team efficiency. All in all teams with higher efficiency were more likely to win. This classification model had an estimated error rate of 21.46% a 0.7585 specificity and 0.8122 sensitivity.

3.4.3 Analysing the models

Let us plot all the random forest models by their estimated accuracy, true positive rate and true negative rate, observe figure 3.7

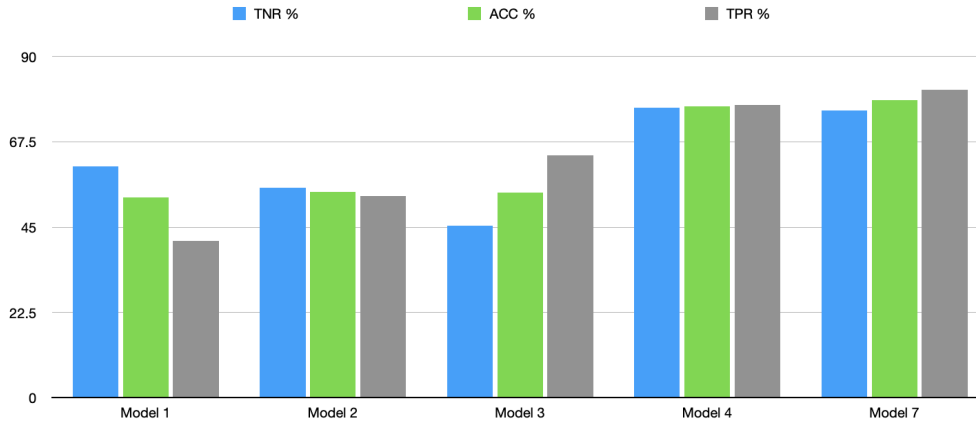


Figure 3.7: Random test observation

Model 7, the efficiency only random forest model proved to have the highest out-of-bag accuracy, the highest true positive rate and the highest true negative rate making it the best model for our data.

Additionally, 10-fold cross validation was ran on model 7. It produced a test accuracy of 78.00%. Which is an incredible fit considering the complex nature of sports prediction. The Kappa statistic, is a measure of inter-rater agreement or classification accuracy that considers the agreement occurring by chance. In the context of your random forest model, a Kappa value of 0.5601626 indicates a moderate level of agreement between the model's predictions and the actual outcomes, beyond what would be expected by chance alone.

3.4.4 Prediction methods

Now we have obtained our best model, we can begin to look at some prediction methods of player data in order to predict win outcomes. It is important we first estimate player data and not team data, as team lineups can change all the time. Missing key players for some games may hinder a teams performance, yet this information is omitted when dealing with just team statistics as opposed to a culmination of player observations. There are many ways to predict a players statistics, this section will highlight a few common methods that could be used to achieve this, however there undoubtedly exist more advanced and accurate methods which serve to improve this prediction process.

Using player averages

According to [source] they are normally distributed, with skew depending on the player. This makes the average the expected value for each statistic, and therefore an estimation of what each player could record for each category.

In R, we observe one team; the New York Knicks. We use the average player efficiency to calculate the total team efficiency using players who

played the majority of the games in the 2022-2023 NBA season. The random forest classifies this as a win. Model 6, the efficiency-only logistic model quantifies this win as a 60.50% chance of happening. If we observe the overall win percentage of the Knicks that year, we obtain 0.57.31% very close to our prediction.

We can conclude that player averages provide an excellent prediction of team a general team win percentage.

Last games against given team

A method of predicting performance whilst incorporating the interactions of other teams is to estimate player efficiency based on the last time those teams played. Take the average from previous games of all the players' efficiency and take the sum to get the estimated team efficiency against the team.

In our example, we look at the Knicks playing against the Atlanta Hawks. We take into account the first 3 games in the matchup to estimate the efficiency of each player. Combining the players that are announced to play/played in that game, we obtain 131.66, inputting this into the random forest model, we classify this as a win, and the logistic regression indicated we have a 0.517429 chance of winning this game. In this case, the Knicks did end up winning this game by a large margin.

Last games in a given time period

The final method of player stat prediction we will analyse is recent player performance. Using recent player performance to predict player stats rather than relying solely on the entire season's data can be motivated by several reasons. Player performance can be significantly influenced by injuries or changes in health status. Recent performance may better reflect a player's current phys-

ical condition and playing form, especially if there have been recent injuries or recovery periods. Basketball, like many sports, is influenced by player form and momentum. Players can go through hot or cold streaks where their performance deviates from their season averages. Analyzing recent games might capture these streaks and provide a more accurate representation of a player's current level of play.

In our example, let us analyse the first 10 games the Knicks played at the beginning of the season. We calculate the efficiency averages of the players in the starting lineup of the next game based on their previous ten performances. We the random forest algorithm predicts that the Knicks will loose the next game, and the logistic regression quantifies the win percentage as 41.29%. In reality, the Knicks did actually end up loosing their 11th game against the Brooklyn nets.

3.4.5 Summary

In this section, we delved into the comprehensive process of utilising the random forest algorithm to analyse NBA player data and predict win outcomes. The exploration began with an exhaustive dataset comprising player statistics from the 2022-2023 NBA regular season, leading to the creation of four cleaned datasets, each tailored for specific modeling purposes. We fitted various random forest models, analysing their performance metrics and comparing their predictive capabilities. Notably, Model 7, focusing on total team efficiency, emerged as the most promising. Beyond model fitting, we investigated different prediction methods, including using player averages, opposing team player performance and recent player performance. These methods provide valuable insights into estimating player statistics and, consequently, predicting team outcomes. The refined models and prediction methods es-

established here will play a pivotal role in constructing future advancements for NBA win outcome predictions, contributing to the overarching goal of employing advanced analytics in sports forecasting

Chapter 4

Conclusions

In conclusion, the aims of this capstone project were to explore the application of the random forest algorithm to predict NBA game outcomes using player data and to contribute to the field of sports analytics, particularly in the context of basketball. Throughout this journey, I delved into the detailed landscape of machine learning, statistics, and the unique dynamics of the NBA. The project unfolded through various stages, encompassing literature review, data collection, preprocessing, feature engineering, model implementation, and evaluation.

In the pursuit of these objectives, I engaged in significant independent learning, researching into advanced statistical concepts, machine learning techniques, and their applications in the realm of sports analytics. My understanding of the random forest algorithm, decision trees, and their adaptation to the unpredictability of NBA player data has deepened considerably. I acknowledge the valuable resources that contributed to this learning, including academic literature, online courses, and collaborative discussions with peers and mentors. These diverse sources enriched my knowledge and provided different perspectives on the intricate balance between statistical theory and

practical application.

Reflecting on this journey, I recognise that a deeper familiarity with certain statistical concepts, especially those related to advanced machine learning techniques, would have facilitated a smoother transition into the project. With this in mind, I've come to appreciate the importance of continuous learning and adaptation in the face of complex and dynamic challenges.

Beyond the direct scope of the project, I acquired insights into effective data manipulation techniques, the nuanced dynamics of NBA gameplay, and the interplay between various statistical metrics. This interdisciplinary exploration not only broadened my understanding of statistics and machine learning but also honed my skills in contextualising these techniques within real-world scenarios.

My journey in this capstone project has been a rewarding experience, blending my passion for data science, basketball, and the pursuit of knowledge. It allowed me to combine two areas of interest, nurturing a more in-depth understanding of the NBA's complex nature and the statistical methods employed in predictive modelling. Through the iterative process of literature review, data analysis, and model refinement, I gained valuable insights into the challenges and opportunities of applying machine learning to sports analytics.

In conclusion, this project has been a testament to the synergy between statistics, machine learning, and the exciting world of professional basketball. The knowledge gained has not only contributed to the field of NBA sports analytics but has also equipped me with a versatile skill set applicable to broader domains within data science. As I embark on future endeavours, I carry forward not only a refined understanding of statistical methodologies but also a passion for uncovering patterns in complex datasets and conclud-

ing relationships in other fields outside of just sports. This project marks the culmination of a chapter in my academic journey being my final year in my bachelors degree, leaving me with a profound appreciation for the possibilities at the intersection of mathematics, statistics, data science and sports analytics.

4.1 Bibliography

[1] Speegle, D, & Clair, B 2021, Probability, Statistics, and Data : A Fresh Approach Using R, CRC Press LLC, Milton. Available from: ProQuest Ebook Central. [29 September 2023].

[2] Mailund, T 2017, Beginning Data Science in R : Data Analysis, Visualization, and Modelling for the Data Scientist, Apress L. P., Berkeley, CA. Available from: ProQuest Ebook Central. [4 September 2023].

The e-textbook was found using google scholar with “Data manipulation using dyplr” as the search condition. As the resource was published fairly recently and contained “Using R” in the title, it seemed like a strong choice. The author wrote this book as a compilation of lecture notes for data science and statistics students. I will be focussing on specific sections in chapter 3 [p58-p73]: Data manipulation

[3] Breiman, L. Random Forests. Machine Learning 45, 5–32 (2001).
<https://doi.org/10.1023/A:1010933404324>

This reference was found by navigating the SpringerLink website. The website is a online literature aggregator for tertiary levels of research and ed-

ucation. After reading a variety of other resources in the random forest field, they all cited this paper as a reference, adding to the over 10,000 citations this paper has received over the past two decades. The research paper was written by Leo Breimen, who at the time, was a statistician perfecting his methods at the University of California. He published the article in 2001 and is credited in the world of statistics as one of the founding fathers of machine learning in statistics.

[4] Schonlau, M., & Zou, R. Y. (2020). The random forest algorithm for statistical learning. *The Stata Journal*, 20(1), 3–29. <https://doi.org/10.1177/1536867X20909688> [6 September 2023]

[5] Rennert, P 2018, *Getting started with machine learning in R*, PACKT Publishing, Birmingham, England.

[6] Couronne, R., Probst, P. Boulesteix, AL. Random forest versus logistic regression: a large-scale benchmark experiment. *BMC Bioinformatics* 19, 270 (2018). <https://doi.org/10.1186/s12859-018-2264-5>

This research article was found using Google scholar with the search terms “Random forests”. This particular paper caught my eye as it contained the term “logistic regression” which I quickly realised would be highly relevant towards my project if I aimed to predict winning chance as a percentage. When deciding how I would quantify my predictions, I had many choices, one of which was logistic regression. This papers sole aim was to conclude which is the better prediction method between random forests and logistic regression.

[7] Horvat, T., Job, J., Logožar, R. & Livada, Č. 2023, "A Data-Driven Machine Learning Algorithm for Predicting the Outcomes of NBA Games", *Symmetry*, vol. 15, no. 4, pp. 798.

[8] Zhao, K.; Du, C.; Tan, G. Enhancing Basketball Game Outcome Prediction through Fused Graph Convolutional Networks and Random Forest Algorithm.

Entropy 2023, 25, 765. <https://doi.org/10.3390/e25050765>

This source was found using the google scholar, with key words of "Random forest", "Prediction", and "NBA" all used in the search criterion. I read the abstract and found that the paper had very similar intentions to those of my own project, and decided to include it. The paper appears to be aimed at a multidisciplinary audience, including researchers, data scientists, sports analysts, and professionals interested in sports analytics, machine learning, and predictive modeling. One key difference is this source has a primary focus of applying graph neural networks to basketball games for outcome prediction.

[9] NBA advanced stats (2023) Stat Glossary — Stats — NBA.com. Available at: <https://www.nba.com/stats/help/glossary> (Accessed: 29 September 2023).

[10] Song, K, Gao, Y & Shi, J 2020, 'Making real-time predictions for NBA basketball games by combining the historical data and bookmaker's betting line', *Physica A*, vol. 547, p. 124411–.

[11] NBA league averages - per game (2023) Basketball Reference. Available at: https://www.basketball-reference.com/leagues/NBA_stats_per_game.html

(Accessed: 13 November 2023).

[12] Staff, NBA. com (2019) NBA frequently asked questions, NBA.com.
Available at:

<https://www.nba.com/news/faq> (Accessed: 13 November 2023).

[13] Tiodorovic, K. (2022) Efficiency- the most important stat in Basketball, BenchBoss. Available at: <https://benchboss.ai/efficiency-the-most-important-stat-in-basketball/> (Accessed: 10 November 2023).

Appendix A

R-code

Capstone_appendix

Nicholas Hedley

2023-11-13

```
# devtools::install_github("abresler/nbastatR")
# http://asbcllc.com/nbastatR/reference/index.html
# (Gl multi package)
Sys.setenv(VROOM_CONNECTION_SIZE=500072)
library("nbastatR")

## Warning: replacing previous import 'curl::handle_reset' by 'httr::handle_reset'
## when loading 'nbastatR'

## Warning: replacing previous import 'httr::timeout' by 'memoise::timeout' when
## loading 'nbastatR'

## Warning: replacing previous import 'magrittr::set_names' by 'purrr::set_names'
## when loading 'nbastatR'

## Warning: replacing previous import 'jsonlite::flatten' by 'purrr::flatten' when
## loading 'nbastatR'

## Warning: replacing previous import 'curl::parse_date' by 'readr::parse_date'
## when loading 'nbastatR'

## Warning: replacing previous import 'purrr::invoke' by 'rlang::invoke' when
## loading 'nbastatR'

## Warning: replacing previous import 'purrr::flatten_raw' by 'rlang::flatten_raw'
## when loading 'nbastatR'

## Warning: replacing previous import 'purrr::flatten_dbl' by 'rlang::flatten_dbl'
## when loading 'nbastatR'

## Warning: replacing previous import 'jsonlite::unbox' by 'rlang::unbox' when
## loading 'nbastatR'

## Warning: replacing previous import 'purrr::flatten_lgl' by 'rlang::flatten_lgl'
## when loading 'nbastatR'

## Warning: replacing previous import 'purrr::flatten_int' by 'rlang::flatten_int'
## when loading 'nbastatR'

## Warning: replacing previous import 'purrr::%@%' by 'rlang::%@%' when loading
## 'nbastatR'

## Warning: replacing previous import 'purrr::flatten_chr' by 'rlang::flatten_chr'
## when loading 'nbastatR'

## Warning: replacing previous import 'purrr::splice' by 'rlang::splice' when
## loading 'nbastatR'

## Warning: replacing previous import 'purrr::flatten' by 'rlang::flatten' when
## loading 'nbastatR'
```



```

## Warning: replacing previous import 'readr::guess_encoding' by
## 'rvest::guess_encoding' when loading 'nbastatR'

## Warning: replacing previous import 'magrittr::extract' by 'tidyr::extract' when
## loading 'nbastatR'

## Warning: replacing previous import 'rlang::as_list' by 'xml2::as_list' when
## loading 'nbastatR'

library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##   filter, lag

## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union

library(tidyr)
library("randomForest")

## randomForest 4.7-1.1
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':
##
##   combine

gamedata2023 <- game_logs(2023)

## Acquiring NBA basic player game logs for the 2022-23 Regular Season

## Warning: The `x` argument of `as_tibble.matrix()` must have unique column names if
## `.name_repair` is omitted as of tibble 2.0.0.
## i Using compatibility `.name_repair`.
## i The deprecated feature was likely used in the nbastatR package.
## Please report the issue at <https://github.com/abresler/nbastatR/issues>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

## Warning: All elements of `...` must be named.
## Did you want `data = -c(slugLeague, typeResult, slugSeason, yearSeason)`?

## Warning: `cols` is now required when using unnest().
## Please use `cols = c(dataTables)`

set.seed(6895980)

#Data cleaning

all_clean_2023 <- select(gamedata2023, outcomeGame, treb, ast, stl, blk, pts, tov)

advanced_2023 <- mutate(gamedata2023, EffFG = (fgm +(0.5*fg3m))/fga,

```

```

TrueSh = pts/(2*(fga +(0.44*fta))),
EFF = pts+treb+ast+stl+blk-(fga-fgm)-(fta-ftm)-tov)

advanced_clean_2023 <- select(advanced_2023, outcomeGame, EffFG, TrueSh, EFF)

all_stats <- select(advanced_2023,outcomeGame, treb, ast, stl, blk, pts, tov, EffFG, TrueSh, EFF )

knicks_games_2023 <- filter(advanced_2023, slugTeam == "NYK")
knicks_games_2023 <- select(knicks_games_2023, outcomeGame, EffFG, TrueSh, EFF, slugMatchup, namePlayer)

total_adv_clean<- advanced_2023 %>%
  group_by(dateGame, slugTeam) %>%
  summarize(
    avg_EffFG = mean(EffFG, na.rm = TRUE),
    avg_TrueSh = mean(TrueSh, na.rm = TRUE),
    total_EFF = sum(EFF,na.rm = TRUE),
    outcomeGame = outcomeGame
  )

```

```

## Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
## dplyr 1.1.0.
## i Please use `reframe()` instead.
## i When switching from `summarise()` to `reframe()`, remember that `reframe()`
## always returns an ungrouped data frame and adjust accordingly.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

```

```

## `summarise()` has grouped output by 'dateGame', 'slugTeam'. You can override
## using the `.groups` argument.

```

```

total_adv_clean<- distinct(total_adv_clean, dateGame, slugTeam, .keep_all = TRUE)

```

#Fitting models

```

rf_model_simple <- randomForest(as.factor(outcomeGame) ~., data = all_clean_2023)
rf_model_simple

```

```

##
## Call:
## randomForest(formula = as.factor(outcomeGame) ~ ., data = all_clean_2023)
##           Type of random forest: classification
##           Number of trees: 500
## No. of variables tried at each split: 2
##
##           OOB estimate of  error rate: 47.36%
## Confusion matrix:
##           L      W class.error
## L 7926 5023    0.3879064
## W 7240 5706    0.5592461

rf_model_all <- randomForest(as.factor(outcomeGame) ~., data = all_stats, na.action = na.omit)
rf_model_all

```

```

##

```

```
## Call:
## randomForest(formula = as.factor(outcomeGame) ~ ., data = all_stats, na.action = na.omit)
##           Type of random forest: classification
##           Number of trees: 500
## No. of variables tried at each split: 3
##
##           OOB estimate of  error rate: 45.73%
## Confusion matrix:
##      L      W class.error
## L 6869 5526  0.4458249
## W 5759 6523  0.4688976

rf_model_adv <- randomForest(as.factor(outcomeGame)~.,
                             data =advanced_clean_2023, na.action = na.omit)
rf_model_adv

##
## Call:
## randomForest(formula = as.factor(outcomeGame) ~ ., data = advanced_clean_2023, na.action = na.
##           Type of random forest: classification
##           Number of trees: 500
## No. of variables tried at each split: 1
##
##           OOB estimate of  error rate: 45.78%
## Confusion matrix:
##      L      W class.error
## L 7725 4670  0.3767648
## W 6627 5655  0.5395701

rf_model_adv

##
## Call:
## randomForest(formula = as.factor(outcomeGame) ~ ., data = advanced_clean_2023, na.action = na.
##           Type of random forest: classification
##           Number of trees: 500
## No. of variables tried at each split: 1
##
##           OOB estimate of  error rate: 45.78%
## Confusion matrix:
##      L      W class.error
## L 7725 4670  0.3767648
## W 6627 5655  0.5395701

rf_total_adv <- randomForest(as.factor(outcomeGame)~. -slugTeam -dateGame,
                             data=total_adv_clean)
rf_total_adv

##
## Call:
## randomForest(formula = as.factor(outcomeGame) ~ . - slugTeam - dateGame, data = total_adv_clean
##           Type of random forest: classification
##           Number of trees: 500
## No. of variables tried at each split: 1
##
##           OOB estimate of  error rate: 23.29%
```

```

## Confusion matrix:
##      L    W class.error
## L 940 290   0.2357724
## W 283 947   0.2300813

rf_Eff <- randomForest(as.factor(outcomeGame)~total_EFF,
                      data=total_adv_clean)

rf_Eff

##
## Call:
## randomForest(formula = as.factor(outcomeGame) ~ total_EFF, data = total_adv_clean)
##              Type of random forest: classification
##              Number of trees: 500
## No. of variables tried at each split: 1
##
##              OOB estimate of  error rate: 21.46%
## Confusion matrix:
##      L    W class.error
## L 933 297   0.2414634
## W 231 999   0.1878049

total_adv_clean <- mutate(total_adv_clean, outcomeGame= as.numeric(outcomeGame == "W"))

logistic_model <- glm(outcomeGame~. -slugTeam -dateGame, data=total_adv_clean)
summary(logistic_model)

##
## Call:
## glm(formula = outcomeGame ~ . - slugTeam - dateGame, data = total_adv_clean)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.47884  -0.32452  -0.00576   0.32859   1.03168
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.3835426   0.0637079  -21.717  <2e-16 ***
## avg_EffFG    -0.5957907   0.3367839   -1.769   0.077 .
## avg_TrueSh    0.2833715   0.3520901    0.805   0.421
## total_EFF     0.0156561   0.0004844   32.319  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.1593958)
##
##      Null deviance: 615.00  on 2459  degrees of freedom
## Residual deviance: 391.48  on 2456  degrees of freedom
## AIC: 2469.7
##
## Number of Fisher Scoring iterations: 2

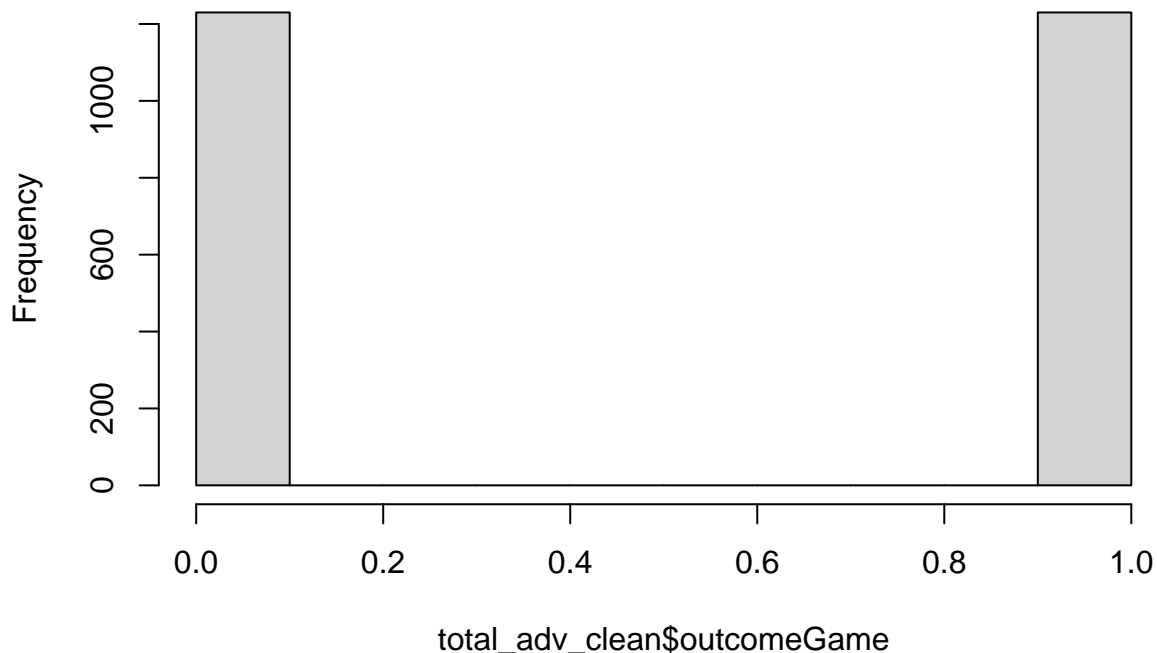
EFF_logistic <- glm(outcomeGame ~ total_EFF, data = total_adv_clean)

summary(EFF_logistic)

```

```
##
## Call:
## glm(formula = outcomeGame ~ total_EFF, data = total_adv_clean)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.50255  -0.32931  -0.01247   0.33158   1.02711
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.4379702  0.0526321  -27.32  <2e-16 ***
## total_EFF    0.0148511  0.0003986   37.26  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.1598913)
##
##      Null deviance: 615.00  on 2459  degrees of freedom
## Residual deviance: 393.01  on 2458  degrees of freedom
## AIC: 2475.4
##
## Number of Fisher Scoring iterations: 2
hist(total_adv_clean$outcomeGame)
```

Histogram of total_adv_clean\$outcomeGame



```
hist(total_adv_clean$total_EFF)
```

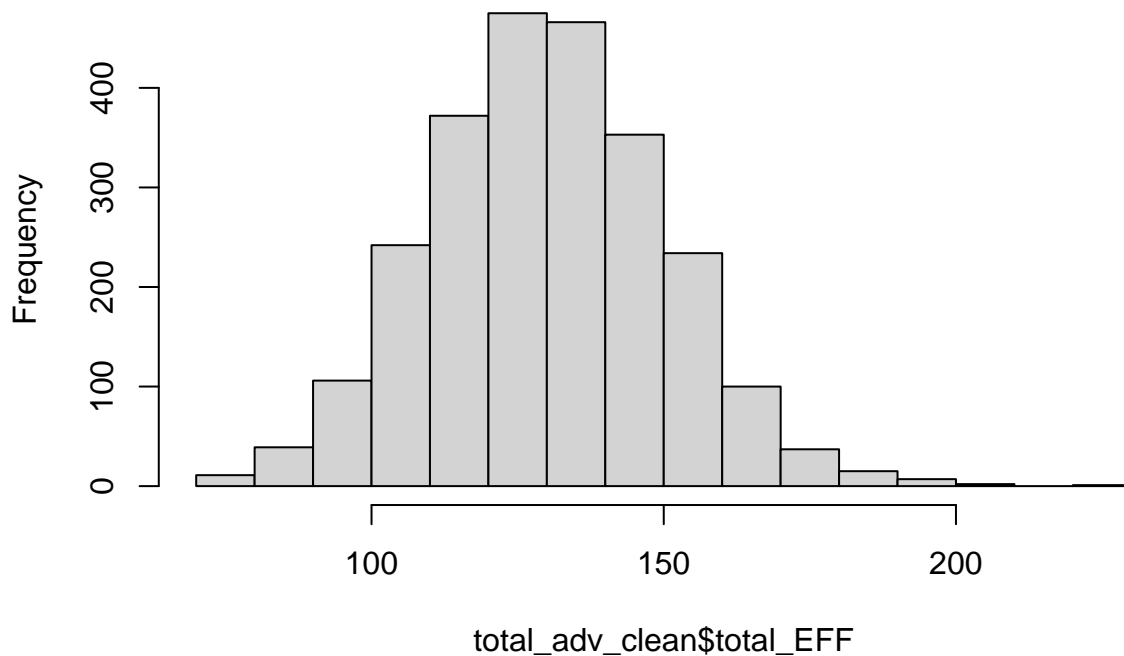
#Interestingly, the best model is the efficiency only model.

#Cross validation

```
library(caret)
```

```
## Loading required package: ggplot2
##
## Attaching package: 'ggplot2'
## The following object is masked from 'package:randomForest':
##
##     margin
## Loading required package: lattice
```

Histogram of total_adv_clean\$total_EFF



```
ctrl <- trainControl(method = "cv", number = 10)
train_model <- train(as.factor(outcomeGame)~total_EFF, data=total_adv_clean, method = "rf", trControl =
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid mtry:
## reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid mtry:
## reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid mtry:
## reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid mtry:
## reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid mtry:
## reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid mtry:
```

```
## reset to within valid range

## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid mtry:
## reset to within valid range

## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid mtry:
## reset to within valid range

## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid mtry:
## reset to within valid range

## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid mtry:
## reset to within valid range

## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid mtry:
## reset to within valid range
```

```
train_model
```

```
## Random Forest
##
## 2460 samples
##    1 predictor
##    2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 2214, 2214, 2214, 2214, 2214, 2214, ...
## Resampling results:
##
##   Accuracy   Kappa
##  0.7800813  0.5601626
##
## Tuning parameter 'mtry' was held constant at a value of 2
```

```
logistic_cv <- train(as.factor(outcomeGame)~total_EFF, data=total_adv_clean, method = "glm", trControl =
logistic_cv
```

```
## Generalized Linear Model
##
## 2460 samples
##    1 predictor
##    2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 2214, 2214, 2214, 2214, 2214, 2214, ...
## Resampling results:
##
##   Accuracy   Kappa
##  0.7833333  0.5666667
```

```
#Predictions
# Player averages.
```

```
average_stats <- select(advanced_2023, namePlayer, EFF, slugTeam)
```

```

average_stats <- average_stats %>%
  group_by(namePlayer, slugTeam) %>%
  summarize(
    avg_EFF = mean(EFF, na.rm = TRUE)
  )

## `summarise()` has grouped output by 'namePlayer'. You can override using the
## `.groups` argument.

Knicks_EFF <- data.frame(total_EFF = 22.8529412 + 25.5844156 + 18.1200000 + 17.1525424 + 15.5185185 +
  15.2602740 + 11.8591549 + 11.2195122)

predictions <- predict(rf_Eff, Knicks_EFF, type = "response")

predictions

## 1
## W
## Levels: L W

predictions_logistic <- predict(EFF_logistic, Knicks_EFF, type = "response")

predictions_logistic

##          1
## 0.6050609
# Last Games against opponents

against_hawks <- filter(knicks_games_2023, slugMatchup == "NYK @ ATL" | slugMatchup == "NYK vs. ATL" )

total_against_hawks<- against_hawks %>%
  group_by(dateGame) %>%
  summarize(
    total_EFF = sum(EFF,na.rm = TRUE),
  )

est_against_hawks <- data.frame(total_EFF = ((112+140+143)/3))

predictions <- predict(rf_Eff, est_against_hawks , type = "class")
predictions

## 1
## W
## Levels: L W

predictions_logistic <- predict(EFF_logistic, est_against_hawks)
predictions_logistic

##          1
## 0.517429
# Recent Games

```



```

last_10 <- filter(knicks_games_2023, dateGame <= "2022-11-07")
last_10 <- last_10 %>%
  group_by(namePlayer) %>%
    summarise(
      avg_EFF =mean(EFF,na.rm = TRUE)
    )

last_10_eff <- data.frame(total_EFF = sum(last_10$avg_EFF)-13.625)

predictions <- predict(rf_Eff, last_10_eff , type = "response")
predictions

## 1
## L
## Levels: L W

predictions_logistic <- predict(EFF_logistic, last_10_eff, type = "response")
predictions_logistic

##          1
## 0.412976

```