ENPH 455 Final Report

An Improved Approach to Algorithmic Draft Selection for Teams in the National Basketball Association

by

Noah Rowe

A thesis submitted to the
Department of Physics, Engineering Physics and Astronomy
in conformity with the requirements for
the degree of Bachelor of Applied Science

Queen's University
Kingston, Ontario, Canada
April 2021

Executive Summary

Within the National Basketball Association (NBA), it is estimated that on average, a regular season win is worth 3.02 million dollars. Teams are motivated to win through financial incentives and bonuses in the millions of dollars. This contributes to an ultra-competitive climate to build the best clubs. Teams aim to develop rosters through the annual NBA draft, where new recruits are claimed by teams in a structured environment. The NBA draft is a zero-sum process, where teams attempt to secure the best players for themselves, thus negating the possibility of playing against them. Despite the high stakes nature of draft selections teams make critical errors in player selection, often resulting in million dollar losses each year.

This thesis presents an algorithm to automate the draft selection process by making unbiased draft choices. Development of the algorithm is achieved by replicating and extending previous work in this area. Increased performance is achieved through an extended data set, accounting for the impact of existing rosters on draftee performance, and extensive model optimization. This algorithm is limited in its application, as it can be only applied to draftees represented by at least one season of college basketball statistics, thus excluding approximately one third of all draftees. When implemented on the appropriate subset, the automated algorithm is able to provide on average 1.14 additional wins per draft pick. This translates to over 3.4 million dollars in added value per draft pick, or just under an average of 7 million dollars annually.

Acknowledgments

I would like to acknowledge my supervisor and mentor, Dr. Ryan Martin, for his continued guidance and support.

Contents

Executive Summary **i**

Acknowledgments **ii**

Contents **iii**

List of Tables **v**

List of Figures **vi**

Chapter 1: Introduction **1**
1.1 Introduction 1
1.2 Problem Definition 2
1.3 Goals 3
1.4 Motivation 3

Chapter 2: Background **5**
2.1 National Basketball Association 5
2.1.1 NBA Draft 6
2.2 Machine Learning Techniques 6
2.2.1 Neural Network Model 7
2.2.2 K-Nearest-Neighbour 7
2.2.3 Random Forest Model 8
2.2.4 Gradient Boosting Model 8
2.2.5 K-Means Clustering 8
2.2.6 Leave-one-out Training 9
2.2.7 Random Oversampling 10

Chapter 3: Methods **11**
3.1 Data Collection 12
3.2 Data Manipulation 15
3.2.1 Data Scaling 16

3.2.2 Drafting Team Features 16
3.2.3 Model Target 19
3.3 Models 19
3.3.1 Model Validation 20
3.3.2 Design Iteration 20

Chapter 4: Results **28**
4.1 Discussion 30
4.2 Limitations 31
4.3 Future Work 32

Chapter 5: Conclusion **33**

Bibliography **34**

Appendix A: Statement of Work and Contributions **38**

List of Tables

3.1 A summary of the predictive features collected 12

3.2 Features used to create the target variable. 14

3.3 A summary of all engineered features. 15

3.4 A description of the features specific for each drafting team. 17

3.5 Default parameter choices used to begin design optimization. 21

3.6 Base model results for each core model type. 23

3.7 Neural network model performance for different distance metrics. . . 24

3.8 Model results for different scaling techniques. 25

4.1 A summary of the design choices for the final algorithm. 28

4.2 Final model results. 29

List of Figures

2.1 A diagram illustrating the leave-one-out training procedure. 9

3.1 A overview of model steps. 11
3.2 A graphic illustrating the results of k-means clustering. 18
3.3 K-nearest-neighbour model optimization. 22
3.4 Gradient boosting model optimization. 22
3.5 A visual comparison between Euclidean and Manhattan distance metrics. 24
3.6 Model performance as a function of k-means clustering parameters. . 25
3.7 The impact of oversampling on mean squared error. 26
3.8 The impact of oversampling on added wins per draft pick. 27

4.1 A bar graph comparing historical and model-generated wins per draft pick. 29
4.2 A bar graph showing the added wins per draft pick per team generated by the model. 30

Chapter 1

Introduction

1.1 Introduction

The National Basketball Association (NBA) is the second most profitable professional sports league in North America, generating $10.7 billion dollars in revenue over the 2018-2019 season [1]. This is supported by dividing total player annual salaries by the number of wins in a typical season, revealing that NBA teams pay approximately 3.02 million USD per win (using 2020 figures) [2]. The League provides monetary incentive the winningest teams. For example, following their championship-winning year, the total worth of the Toronto Raptors increased by 25%, as compared to the League average of 14% [1]. This highlights a strong financial incentive for each private club to create a winning team. Teams strive to optimize all aspects of operations to secure the best players and build the best possible team. A notable component of team development and refinement is the annual NBA player draft, where prospective players are recruited through a structured process involving all teams. This is a competitive and zero-sum process, with teams attempting to secure the best players. The productivity of draft selection choices is a major factor in determining future

success of NBA teams. Player personnel decisions have been cited as the primary function of League general managers [3].

It is understood that data-driven, objective analysis is required by NBA general managers to optimize player personnel choices; however, it is often difficult to exclude emotion and biases from the decision making process. For example, Berri, Brook, and Fenn [4] find that front offices disproportionately focus on offensive performance, giving rise to suboptimal player evaluation. This practice highlights the need for a data-driven approach to player recruitment, and helps to explain an influential trend in the NBA today. Specifically, what can be considered a data analytics revolution is unfolding across the League, with almost every team now hosting a data analytics department focused on this type of analysis [5].

Published work related to optimizing player recruitment decisions is currently available. For example, Maymin outlines a system for automating all aspects of player acquisition, including draft selections, free agent signings, and player trades [6]. This model is used as a foundation to develop an improved approach, and Maymin's results will serve in this work as a useful baseline to assess final model effectiveness.

1.2 Problem Definition

The final model described in this paper should provide managerial staff on an NBA team with the ability to indiscriminately assess player acquisition opportunities in the absence of bias and human emotion. This can be achieved through the elimination of biases driven by non-relevant features (i.e. former NBA-player parents, physical features/appearance, sexual orientation, religion, etc.).

Based on data availability, this paper focuses only on player who played a minimum of one season of college-level basketball and at least three seasons in the NBA. This represents approximately two thirds of all draftees.

1.3 Goals

The goal of this thesis is to develop an algorithm to identify the highest potential draftees from a group of draft candidates. The final algorithm should be able to recommend the most suitable player for a given team, while considering their existing roster. A fundamental goal of this work is to replicate the results described by Maymin in [6], and incorporate existing roster data to improve upon these results. Success will be measured based on additional wins anticipated with use of the proposed model, as opposed to historical choices. A beneficial feature of the model may include the ability for the user to understand how the model reaches decisions regarding player recommendations. Model interpretability would promote credibility of the algorithm, as large financial considerations are contingent on the accuracy of model output.

1.4 Motivation

This paper attempts to provide an algorithmic solution to address draft selection habits currently adopted by the management of various NBA teams. Utilization of this model is expected to position teams to better improve their rosters, thereby preventing strong draft candidate players as serving as future competitors.

In addition, draft selection choices often define a club for many years. That is, drafting teams hold privileged rights over their draft picks for up to four years [7]. Teams will also be able to leverage these contracts in trading scenarios, allowing for

further team development. All of this supports the need for a reliable process to secure talented draft picks.

Lastly, as mentioned, there is a growing acceptance of data analytics within the sports world, including the NBA. Teams are using statistical methods for player recruitment, contract evaluation, injury prevention, and more [8]. Lagging NBA teams with no analytics departments are at risk of missing out on the benefits, allowing the competition a sizable advantage.

Chapter 2

Background

This section introduces key concepts related to NBA structure and operations, with a specific focus the drafting process. Core machine learning algorithms and training styles are also discussed with reference to their application within this work.

2.1 National Basketball Association

The NBA consists of 30 privately owned teams across North America, which compete in a regular season of 82 games to earn a spot in the playoffs. The top performing eight teams from each of the two conferences (Eastern and Western) are then paired in a tournament style structure, for a best of seven series. The winner of each series moves on to play other series winners, and losing teams are disqualified. Internally, teams often provide monetary incentive for players and management to perform well in the playoffs, and the League offers teams increasing amounts of money for each playoff series win [9]. For example, the Toronto Raptors were given 5.6 million dollars by the NBA for winning the 2019 NBA playoffs [9].

2.1.1 NBA Draft

Each season, the NBA hosts a drafting process that allows clubs to sign individuals from a pool of perspective players. The drafting order is designed to promote fairness and to provide advantage to losing teams. Increased odds to get lower draft picks are given to teams lower in the standings, therefore giving them better recruitment options. For example, the lowest three teams in the standings have a 14% chance to get the first draft pick, while the 4^{th} lowest team has a 12.5% chance [10]. After the first four picks are decided, the remaining selections are distributed according to the inverse of placement in the standings.

Each team is assigned 2 picks per draft, resulting in 60 player selections. However, teams often trade draft picks for players, cash considerations, or future picks, meaning that not every draft consists of each team selecting twice.

For eligibility purposes, all draftees are required to be 19 years of age or older and at least one year removed from high school [11].

2.2 Machine Learning Techniques

The core predictor behind the algorithm introduced in this thesis will be selected from one of four machine learning statistical models: neural networks, standard random forests, k-nearest-neighbour, and gradient boosted random forests. Machine learning models are an ideal solution to this problem given the availability of data and the opportunity to make data-focused decisions, excluding non-relevant factors. Each of theses approaches have unique strengths and weaknesses, including computational demands, model accuracy, and the interpretability of results.

2.2.1 Neural Network Model

Neural network models are a type of machine learning model designed to replicate the behaviour of a biological human brain [12]. These networks are initialized with neurons connected within various layers that perform mathematical operations, transforming an input vector to a final output. Neural networks have a complex parameter space and performance is highly dependent on the structure of the model. Neural network training can be computationally intensive, depending on model size. Neural networks are a type of black-box model that does not provide justification of its decisions. Despite these unfavorable attributes [12], neural networks are explored in this thesis given their high levels of accuracy.

2.2.2 K-Nearest-Neighbour

The k-nearest-neighbour (KNN) algorithm generates predictions based on the N closest entries in the training data set, where N is a parameter chosen by the user [13]. KNN is considered in this work given its fast application speed, as no training period is required. Also, due to the nature of the algorithm, KNN results are not subject to randomness. This eliminates any need for repeated runs to verify results. KNN also provides results that are easily interpreted, as it is simply an average of the N most similar data points in the training data set. Unlike more complex models, KNN optimization typically only involves optimizing one parameter, N, allowing for easy deployment.

2.2.3 Random Forest Model

Random forest models are a type of ensemble predictor, meaning that multiple individual models are combined to form a single output [14]. This project investigates random forest models implemented with decision trees as the base predictor. The random forest algorithm is considered due to its general applicability and ability to provide feature importance, allowing for a more accessible model. Random forest models also have a relatively simple parameter space and a computational speed that can be easily scaled by increasing or reducing the number of individual predictors [14].

2.2.4 Gradient Boosting Model

Gradient boosting (GB) models are an enhanced version of random forest models that train individual models in order to address model weaknesses. GB models have a more complicated parameter space, but can often achieve better accuracy than standard random forest models [15]. Like random forest models, GB models can provide feature importance [16]. For these reasons, GB models are investigated for use within the model presented in this work.

2.2.5 K-Means Clustering

K-means clustering is an unsupervised machine learning algorithm that classifies data entries into k different categories, where k is selected by the user[17]. Within this application, k-means clustering is used to identify players similar to one another, according to their feature vectors. K-means clustering was chosen because of its computational speed and its capacity to specify the number of clusters in advance

of implementation. It is also necessary to choose an algorithm which allows for the classification of new data without restructuring the original categorization model, ensuring that clustering remains consistent across all players.

2.2.6 Leave-one-out Training

When training a machine learning model, it is important that no entries in the testing data set are present in the training data; thus preventing the possibility of training bias, as all testing data points have never been seen by the model. Due to the relatively small amount of data used in this thesis, further limitation of the data size greatly reduces model accuracy. Leave-one-out (LOO) training is a type of training that seeks to maximize training data, while preventing testing exposure. For each entry in the training data, a unique model is trained on all other data entries. This model can generate predictions on the excluded data point with maximal training exposure, while preventing training bias. Figure 2.1 summarizes this process.

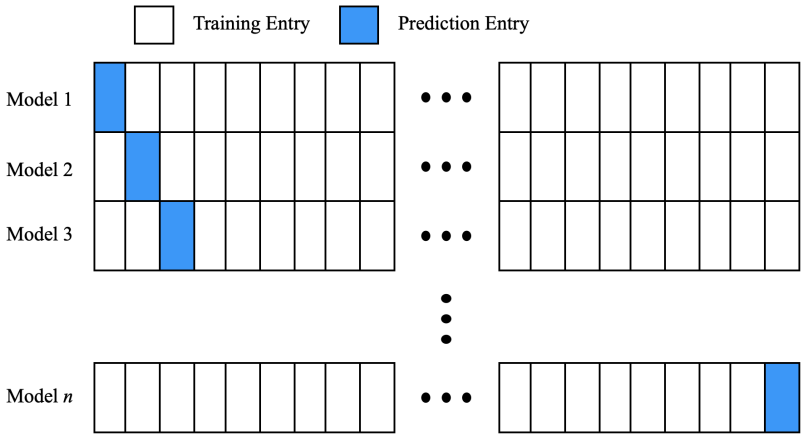

Figure 2.1: A diagram illustrating the leave-one-out training procedure. Each row represents an independent training session, resulting in n different models.

2.2.7 Random Oversampling

It is well understood that machine learning models tend to struggle when applied to imbalanced data [18]. A data set is considered imbalanced when specific types of data entries are underrepresented, as compared to the dominant type. A common method to address this is to artificially duplicate random elements of the minority subset until a balance is achieved. This technique is known as random oversampling, and compels model training to focus on underrepresented data points. In this application, random oversampling is applied to the highest performing faction of draftees in the data. By increasing the model's exposure to the most influential data entries, it is better equipped to recognize them when applied to unseen data [18].

Chapter 3

Methods

This chapter provides a detailed outline of how individual components of the model are chosen and optimized, as well as a depiction of how these discrete units are combined. Figure 3.1 explains how the two raw data sets are processed, combined, predicted upon, and evaluated using a flow-style diagram.

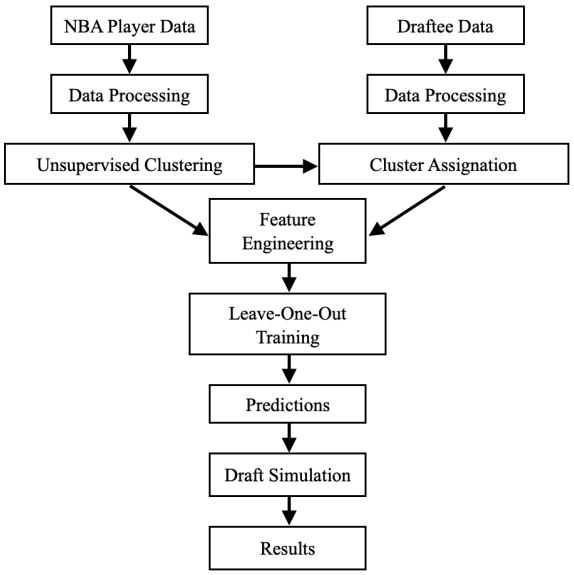

Figure 3.1: A visual summary of the steps taken to transform the raw data into feature vectors and generate model output.

3.1 Data Collection

Data for this project were collected using web-scraping software, as well as from published data sets. All data scraping was conducted using Octoparse, a commercial software [19]. Data related to draftees was collected from multiple sources [20, 21, 22, 23, 24], and historical NBA data was downloaded from the Kaggle data set "NBA Players stats since 1950" [25], containing player stats from 1950 to 2017. Table 3.1 shows the list of features used to create the input vectors, and Table 3.2 outlines the statistics used to develop a target variable. Draftee data spans from 2000 to 2017, and all NBA data prior to 2000 is excluded from the analysis, as it is deemed to be less relevant as compared to more recent data. Removing players with fewer than 3 years of data resulted in a data set containing 655 draftees. Mean imputation was used to address missing values.

Table 3.1: A summary of the predictive features collected. Values denoted with * are also contained within the Kaggle data set "NBA Player stats since 1950" and represent the same statistic, but applied to NBA games. All college basketball statistics are averaged over each season played.

Data Point	Description	Source
Mock	Drafting order from mock drafts performed by ESPN.com	ESPN.com
RSCI	Ranking of the player when they graduated high school	Draftexpress.com
Awards	The number of NCAA awards won by the player	Basketball.realgm.com
SOS	A measure of how difficult the player's college schedule was	Basketball.realgm.com
Position*	Player position	NBA.com

Table 3.1 – continued from previous page

Statistic	Description	Source
Wingspan	Player wingspan (NBA combine metric)	NBA.com
Reach	Player standing reach (NBA combine metric)	NBA.com
Body fat %	Player body fat percentage	NBA.com
Hand width	Player hand width	NBA.com
Hand length	Player hand length	NBA.com
Height*	Player height	NBA.com
Weight*	Player weight	NBA.com
Leap	Maximum vertical leap (NBA combine metric)	NBA.com
Speed	Sprint speed performance (NBA combine metric)	NBA.com
Agility	Lane agility performance (NBA combine metric)	NBA.com
BPR	Bench press repetitions performed (NBA combine metric)	NBA.com
GP	Total games	Basketball-reference.com
MP	Total minutes	Basketball-reference.com
P*	Total points	Basketball-reference.com
A*	Total assists	Basketball-reference.com
STL*	Total steals	Basketball-reference.com
BLK*	Total blocks	Basketball-reference.com
TOV*	Total turnovers	Basketball-reference.com
OBR*	Total offensive rebounds	Basketball-reference.com
DBR*	Total defensive rebounds	Basketball-reference.com
eFG%*	Effective field goal percentage	Basketball-reference.com
ORB%*	Offensive rebound percentage	Basketball-reference.com
DRB%*	Defensive rebound percentage	Basketball-reference.com
AST%*	Assist percentage	Basketball-reference.com
TOV%*	Turnover percentage	Basketball-reference.com
STL%*	Steal percentage	Basketball-reference.com
BLK%*	Block percentage	Basketball-reference.com

Table 3.1 – continued from previous page

Statistic	Description	Source
USG%*	Usage percentage	Basketball-reference.com
PER	Player efficency rating	Basketball-reference.com
OR	Dean Oliver's offensive rating	Basketball-reference.com
DR	Dean Oliver's defensive rating	Basketball-reference.com
OWS*	Offensive win shares	Basketball-reference.com
DWS*	Defensive win shares	Basketball-reference.com
3PA*	Total three point shot attempts	Basketball-reference.com
3P%*	Three point shot percentage	Basketball-reference.com
FTA*	Total free throw attempts	Basketball-reference.com
FT%*	Free throw percentage	Basketball-reference.com
FGA*	Total field goals attempted	Basketball-reference.com
PF*	Total personal fouls	Basketball-reference.com
Combine	Whether the player completed the NBA combine	Basketball-reference.com

Table 3.2: Statistics used to generate the model target variable. See source locations for variable derivations.

Statistic	Description	Source
Win shares	Win shares earned by the player per NBA season	Basketball-reference.com
Wins produced	Estimated wins produced by the player per NBA season	boxscoregeeks.com
Estimated wins added	Estimated wins added by the player per NBA season	insider.espn.com

3.2 Data Manipulation

Draftee specific data and historic NBA data are separately used to create player-specific features. The raw draftee statistics were used to build each feature vector and new features were derived to help a model identify important measures. The position feature was separated into three columns: Center, Forward, and Guard, indicating what positions a draftee can play. The mock draft feature represents 6 different columns, each denoting different mock evaluations. The statistics shown in Table 3.1, along with the engineered features described in Table 3.3, comprise a length 55 base feature vector unique for each draftee in the data set.

Table 3.3: A summary of the derived statistics, generated from the original features. These features were generated for both the NBA data set and the draftee data.

Derived Statistic	**Description**
Points per field goal attempt	Total points divided by total field goal attempts
Points per game	Total points divided by total games played
Points per minute played	Total points divided by total minutes played
Minutes played per personal foul	Total minutes played divided by total personal fouls
Free throw attempt per field goal attempt	Total free throw attempts divided by total field goal attempts
Minutes played per three point attempt	Total minutes played divided by total three point shot attempts
Assist per turnover	Total assists divided by total turnovers

3.2.1 Data Scaling

Scaling methods were applied to the input data to increase the effectiveness of the models [26]. Two scaling techniques are investigated: standard scaling and min-max scaling. Standard scaling in performed on each feature by removing the mean and scaling to unit variance, in accordance with Equation 3.1 [26]:

$$Z_i = \frac{X_i - \mu}{\sigma} \tag{3.1}$$

where Z_i is the scaled output, X_i is the input feature, and μ and σ are the feature column mean and standard deviation, respectively. Min-max scaling normalizes by mapping each input feature to a range between 0 and 1, in accordance with Equation 3.2:

$$Z_i = \frac{X_i - min(X)}{max(X) - min(X)} \tag{3.2}$$

where $min(X)$ and $max(X)$ are the minimum and maximum values of the feature column X [26]. Both methods were employed and tested to optimize the model.

3.2.2 Drafting Team Features

In addition to the base feature vector, two sets of drafting team-specific features were created for each player, both focused on comparing the draft candidate to the current roster of the drafting team. One set focused on comparing the draft candidate to the starting roster of the drafting team, and the other set focused on comparing the draftee to similar-styled players on the drafting roster. Starting players for each roster were assumed to be the five players with the greatest number of minutes played. These features are summarized in Table 3.4.

Table 3.4: A description of the features specific for each drafting team. Two versions of these features (excluding label count) were created for each draftee: one calculated over the five players with the most minutes (starting players) on the drafting team, and one over the subset of players belonging to the same label as the draftee, resulting in 11 new features. Distance metrics are passed as a model parameter.

Feature	Description
Average distance	The average distance value between the draftee and each player in the subset
Distance dot minutes	The dot product of the distance value between the draftee and a given player in the subset, and that players respective total minutes played.
Distance dot win shares	The dot product of the distance value between the draftee and a given player in the subset, and that players respective win shares.
Minimum distance	The minimum distance metric value between the draftee and each player in the subset
Minimum distance win shares	The win shares of the player from the subset with the minimum distance value between them and the draftee
Label count	The number of players on the drafting teams roster belonging to the same cluster as the draftee

The subset of similar styled players was determined from the results of k-means clustering. The algorithm was trained on NBA players and applied on the draftee data set to generate a label for each draft candidate. Draftee labels can be compared to the labels of the drafting roster to identify players who accomplish similar roles. Figure 3.2 shows the results of k-means clustering applied to NBA player statistics with three clustering groups, summarized on a subset of the features.

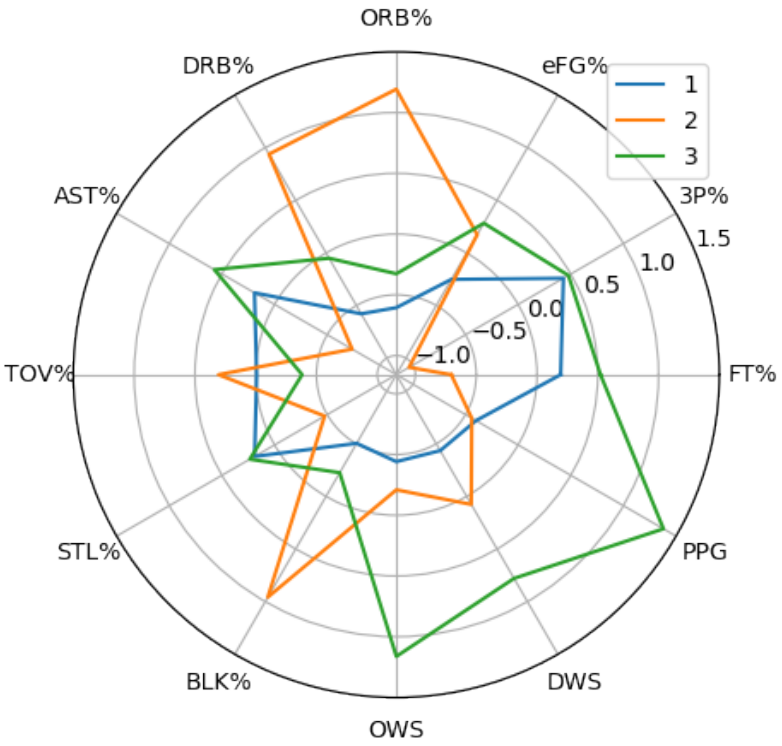

Figure 3.2: A plot demonstrating the differences between the three clusters generated by k-means clustering. Each line shows the average values of a cluster, normalized using standard scaling. See Table 3.1 for variable descriptions.

Figure 3.2 demonstrates three distinct clusters containing players of varying skill sets and abilities. Cluster 1 contains below-average players (presumably bench players) with no dominant skill set. Cluster 2 contains players with a focus on rebounding and blocks, while lacking in shooting ability (most likely centers). Cluster 3 shows scoring-focused players that are talented shooters and passers. The cluster sizes were 3203, 1586, and 2061 respectively. This is important, as clusters should be well distributed in order to have sufficient samples on NBA rosters. Rosters missing a cluster type have features set to the data set minimum for that particular feature.

3.2.3 Model Target

As the goal of this thesis is to develop an algorithm to identify the highest potential draftees, it is necessary to quantify the contribution a player provides to their team. This is done using the wins made metric, as defined by Maymin in [6] using Equation 3.3:

$$WinsMade = \frac{WS + WP + EWA}{3} \tag{3.3}$$

where WS is win shares, WP is wins made, and EWA is earned win average. These variables and their sources are detailed in Table 3.2. Wins made is a measure of the number of wins a player is responsible, calculated by averaging the output of three metrics designed to quantify the same. Wins made is the target variable each machine learning model is designed to predict.

3.3 Models

The structure of the final predictor was crafted to facilitate design iteration for optimization and to work for different core predictor types. Design features that were varied and tested for effectiveness include core model type, data scaling method, number of clustering groups, distance metrics, and oversampling strategies. The core model used to generate predictions for each draftee was trained using LOO training, as done by Maymin in [6]. This training procedure resulted in 655 unique models, each one excluding a specific player from the training data. These models could then be used to predict on the excluded entry using the most training data possible. Therefore, after training, each draftee has an associated model prediction.

3.3.1 Model Validation

In order to validate a given model, a simulation was designed to replicate an actual draft with a single team utilizing model predictions to make their choices. A drafting simulation allows for the conversion of model predictions to actual team benefit. This is important, as model error (in the form of mean squared error) is not representative of model effectiveness. This is of note, as the purpose of this thesis is not to predict draftee performance, but rather to develop a drafting algorithm that maximizes drafting efficiency for the controlling team. That is, the model presented here is designed to identify the highest potential draft candidate, rather than to predict the exact performance of a given candidate. This work uses the results of a drafting simulation (added wins made per draft pick) as the dominant validation metric.

For a given draft, a simulation is performed by controlling a single team's choices, with selections generated by the model. For teams not utilizing the model, it is assumed that they select the draftee that was chosen earliest, out of the remaining players. This is not entirely realistic, as different teams prioritize players differently, but is a necessary assumption in the context of this work. Teams controlled by the model select the player with the highest predicted wins made, and drafting order in the simulation reflects that which is found in the actual draft process. When the simulation is complete, added wins is calculated as the difference between the original and new wins made of a teams draft choices.

3.3.2 Design Iteration

The model used to generate player predictions was designed for optimization over a set of parameters and model structures. Model parameters were optimized based on

added wins per draft pick, generated from drafting simulations. As it would have taken over 100000 unique models to extensively test all parameter options, optimization was done in series in the following order: core model type, distance metric, scaling method, number of clustering categories, and oversampling designs. Core model type was addressed first, as it was assumed to be the most meaningful choice. Distance metric, scaling method, and number of clusters was done in no particular order, as they were approximated as independent of each other. Finally, different oversampling strategies were tested. This was achieved by varying the oversampling range and the number of duplicated entries, as these two features are closely related. The default parameters used to begin optimization are shown in Table 3.5.

Table 3.5: Default parameter choices used to begin design optimization.

Parameter	**Default Value**
Distance metric	Euclidean distance
Scaling method	Standard scaling
Number of clusters	3
Oversampling	None

Core Model

Prior to assessing each of the four core models, initial parameter choices were required to facilitate accurate comparison. The number of neighbours used in the KNN model and the learning rate of the GB model were chosen according to the most added wins provided. The plots for this are shown in Figures 3.3 and 3.4.

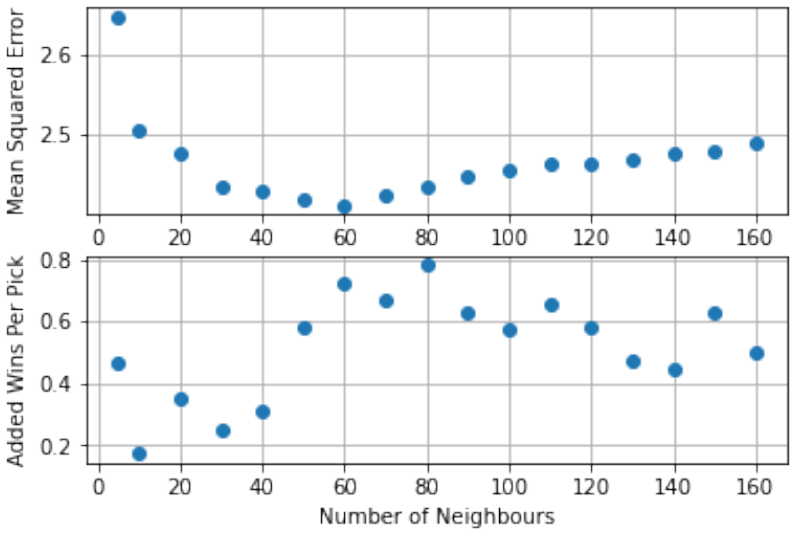

Figure 3.3: K-nearest-neighbour model mean squared error and added wins per pick as a function of the number of neighbours.

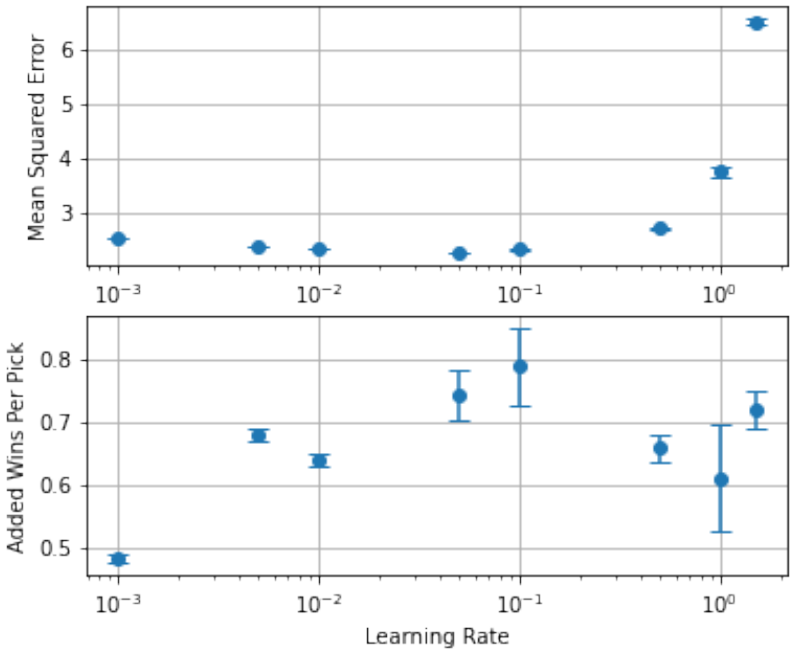

Figure 3.4: Gradient boosting model mean squared error and added wins per pick as a function of model learning rate.

The base neural network chosen was a single-layer perception model. This is the simplest form of neutral networks, with an input layer densely connected to a single output node. The gradient boosted and random forest models were initialized with 100 individual estimators. More estimators would likely result in marginally better results, but given computational restraints, 100 was selected as a starting value.

The core machine learning model was selected after performing 25 simulation runs over 15 years of data, using default parameters for each model (a single KNN simulation was sufficient, as there is no randomness present). The results of this are shown in Table 3.6. As the neural network outperformed all other models, it was chosen as the core model.

Table 3.6: Base model results for each core model type. Values are averaged over 25 training sessions over the entire data set. As KNN is a deterministic algorithm, it was only performed once.

Model	**Mean Squared Error**	**Added Wins**
Neural Network	2.27 +/- 0.01	1.0 +/- 0.1
KNN	2.435	0.780
Random Forest	2.34 +/- 0.02	0.76 +/- 0.05
Gradient Boosting	2.34 +/- 0.01	0.78 +/- 0.07

Distance Metric

Distance metric refers to the method of computing the difference (or distance) between two data entries. The two options include Euclidean distance and Manhattan distance. Both look to measure the distance between two points in N-dimensional space, where N is the feature vector size. Euclidean distance calculates the shortest possible distance, where Manhattan paths can only follow directions orthogonal to the coordinate system. The difference between these two algorithms is shown in

Figure 3.5.

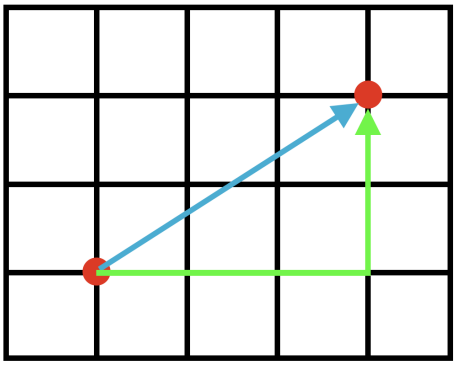

Figure 3.5: An example demonstrating the differences between Euclidean (blue line) and Manhattan (green line) distances in two-dimensional space.

The neural network model was run 20 times over the entire data set for both distance types, and the results are shown in Table 3.7. As Manhattan distance provided the most added wins, it was selected.

Table 3.7: Neural network model training results while using Euclidean and Manhattan distance metrics. Results were averaged over 20 training sessions.

Distance Metric	Mean Squared Error	Added Wins
Euclidean	2.28 +/- 0.01	0.99 +/- 0.08
Manhattan	2.27 +/- 0.01	1.15 +/- 0.08

Scaling Method

The procedure described in the preceding section was repeated to select the optimal feature scaling technique between standard scaling (see Equation 3.1) and min-max scaling (see Equation 3.2). The so-far-optimized neural network was implemented 20 times over the entire data set, once for each scaling method, with the results shown in Table 3.8. From this, standard scaling was chosen as the best method.

Table 3.8: A comparison of the results from standard scaling and min-max scaling methods. Results were averaged over 20 training sessions.

Scaling Method	Mean Squared Error	Added Wins
Standard Scaling	2.28 +/- 0.01	1.15 +/- 0.08
Min-max Scaling	2.443 +/- 0.008	0.6 +/- 0.1

Number of Clusters

The number of unique clusters used in the k-means clustering algorithm was also investigated to determine the best value. This was done by running the simulation ten times for integer values of clusters ranging from 1 to 9. The optimal core model, scaling method, and distance metric was used to evaluate cluster number. The results are shown in Table 3.6.

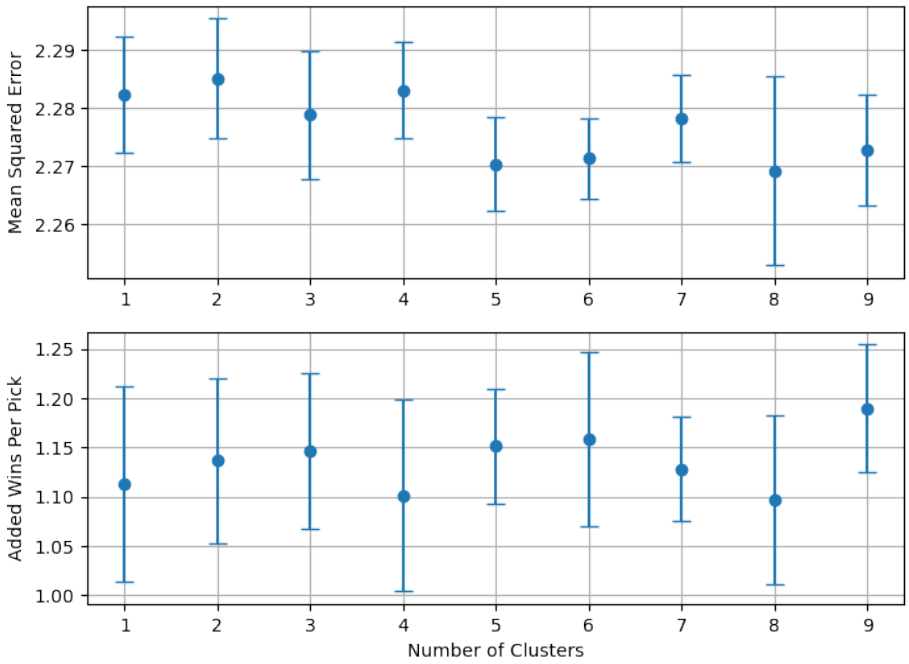

Figure 3.6: Model performance as a function of the number of clusters used for k-means clustering.

From these plots, it was determined that cluster number was not a key factor in model performance. Three clusters was chosen as it provides easy to understand groupings (see Figure 3.2), aiding in model interpretability.

Oversampling

Oversampling performed in this paper was defined by two parameters: oversampling value and oversampling size. Oversampling value refers to the value of the target variable over which data entries will be designated as the sample of interest, and oversampling size is the number of random duplication added to the data set. These parameters were optimized together, with the resulting mean squared error and added wins metrics shown in Figures 3.7 and 3.8.

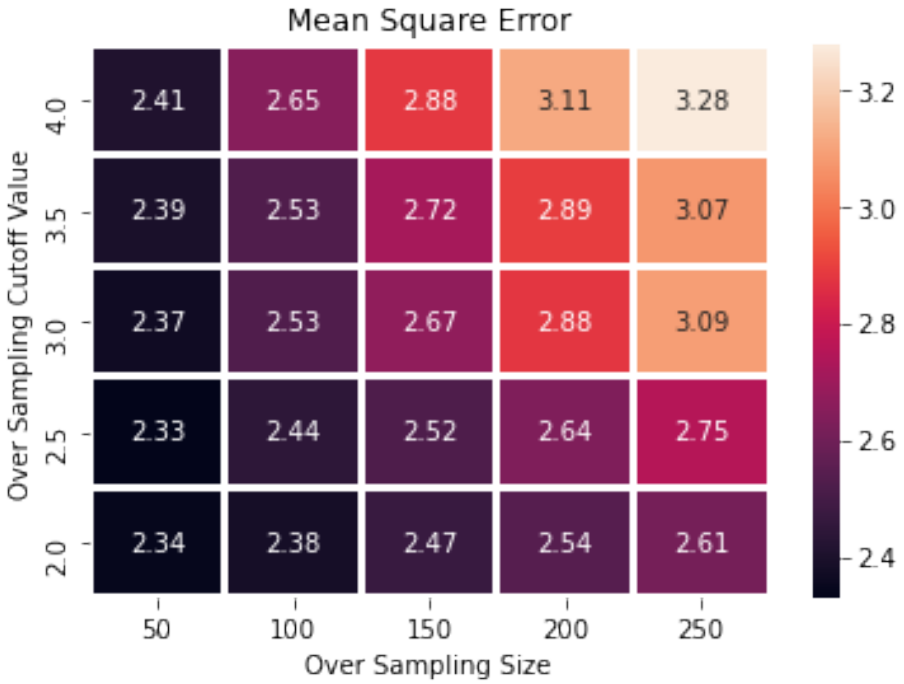

Figure 3.7: A heat-map showing the impact on mean squared error of varying oversampling cutoff and oversampling size parameters.

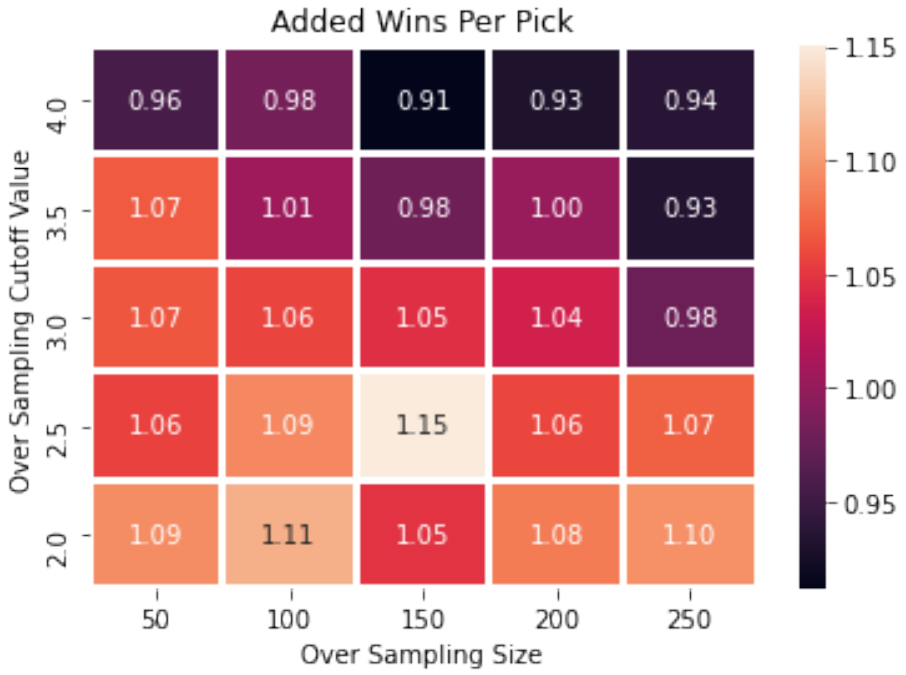

Figure 3.8: A heat-map showing the impact on average added wins per draft pick of varying oversampling cutoff and oversampling size parameters.

These data show that when when more aggressive oversampling is performed (either by increasing the replicated data size or selecting a more limited range of data to duplicate), the model shows no improvement in added wins, while mean square error is negatively impacted. Compared to the baseline values of MSE=2.28 and Added Wins=1.15, model performance is diminished by both increasing oversampling size and oversampling value.

Chapter 4

Results

The final model was chosen as the best performing model through design iteration to select the best, as measured by added benefit when implemented. Final model specifications are shown in Table 4.1.

Table 4.1: A summary of the design choices for the final algorithm.

Design Feature	**Final Choice**
Core model	Neural network
Distance metric	Euclidean distance
Scaling method	Standard scaling
Number of clusters	3
Oversampling	None

This model was implemented on the data set 30 times to assess its effectiveness, and the results are summarized in Table 4.2.

Table 4.2: A summary of final design results. Results are averaged over 30 training periods.

Metric	Performance
Mean Squared Error	2.27 +/- 0.01
Average Wins Per Draft Pick	2.30 +/- 0.09
Average Added Wins Per Draft Pick	1.14 +/- 0.09

Figure 4.1 illustrates the comparison between current draft selections and model outputs using added wins per draft pick. Figure 4.2 shows the average benefit for each team.

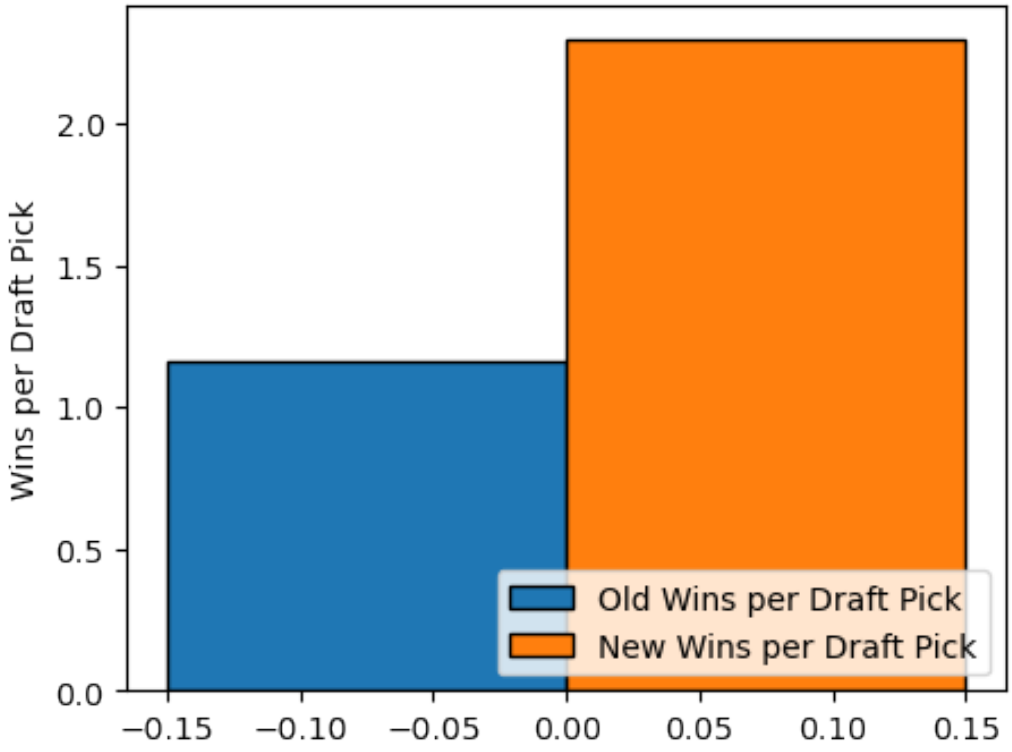

Figure 4.1: A bar graph comparing the average wins per draft pick from current selection habits (left) and the models draft choices (right).

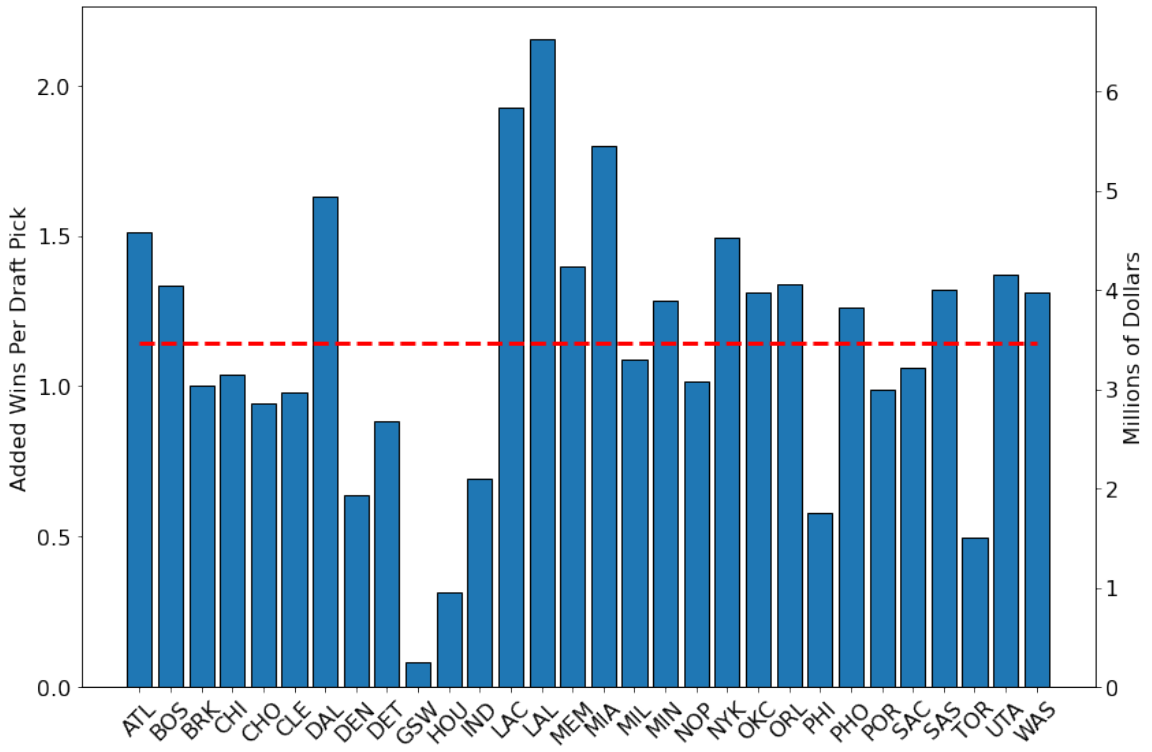

Figure 4.2: A bar graph showing the average added wins per draft pick for each team in the NBA. The left y-axis shows results in terms of average added wins. The right y-axis shows results in terms of the average added monetary value per draft pick, in millions of dollars. The red dotted line represents the average value across all teams: 1.1 +/- 0.5 in average added wins per pick or 3 +/- 1 in added millions of dollars per draft pick.

4.1 Discussion

The results shown in Figure 4.2 demonstrate that current draft selection methods implemented by NBA clubs are inefficient, with each selection costing teams millions of dollars annually. As Maymin does not provide explicit accuracy metrics in [6], it is difficult to compare his result to the algorithm presented in this paper. However, Maymin does claim that his model generates approximately one extra win per draft

pick [6]. Results achieved by a replication of his model developed for this paper provided 0.85 added wins per draft pick. This model utilized 100 estimators trained over the entire data set. As Maymin most likely implemented a more focused parameter selection process with increased training times, but would have been limited by a smaller data set, these results are considered consistent. Therefore, it is assumed that the model described in this paper outlines at minimum an approximate 14% increase in effectiveness over Maymin's original model.

With the assumption that each win is worth 3.02 million USD, this 14% increase represents an extra 425 thousand dollars per draft pick. Overall, the final model provides an extra 3.4 million dollars per draft pick when compared to current practices. Each club has an average of 2 draft picks per season, translating to losses of almost 7 million dollars annually.

4.2 Limitations

To begin, the algorithm introduced in this work was not designed to predict draftee potential. Rather, it was constructed to optimize the draft selection process. That is, this model is not purposed to generate predictions related to individual player performance.

The current model presented here assess only players who have played at least one season of college basketball, excluding international players and draftees who declare eligibility directly from high school. Eligibility for draftees directly out of high school is currently under review by the League [11]. If this comes to fruition, the number of players excluded from consideration would necessarily increase, reducing overall model effectiveness.

Finally, model evaluation assumes that all players would perform equally well on different teams. While this assumption allows for model testing, it may be flawed to the extent that teams have unique systems and interpersonal dynamics which impact player performance.

4.3 Future Work

Future work should focus on including players who did not play college basketball, as this will provide a more effective model. Beyond addressing this limitation, model improvement can also be explored in the following ways.

The model currently works by predicting draftee performance and selecting the candidate with the highest associated value. Substituting the core predictive model with a learn-to-rank algorithm (as used by search engines [27]) will allow model training to focus on the comparison of draftees, rather than the prediction of specific performance values. The core model could also replaced with a classification approach, where the model is trained to identify the highest potential draftee. Both approaches offer unique ways to recommend draft selections, and may support enhanced model performance.

Principle component analysis (PCA) could have been used, rather than a clustering algorithm, to identify similar players. PCA is designed to reduce the dimension of a data set while retaining as much important information as possible [28]. Cheng's work [29] demonstrates the efficacy of this approach in identifying players with similar styles. Implementation of PCA could include results by replacing discrete categorization with a continuous approach.

Chapter 5

Conclusion

The model introduced in this work outlines a strategy to remove human emotion from the NBA draft selection process while doubling the efficiency of draftee selections by NBA teams within the simulation timeline (2000-2017). This thesis sought to extend the work of Maymin in [6] by improving upon his results through an extended data set, accounting for the impact of existing rosters, and extensive model optimization. The model discussed in this paper provides an additional 1.14 wins per draft pick, translating to an added 3.4 million dollars of value per pick. This represents an improvement on Maymin's results, which are estimated to provide approximately 1 additional win per draft pick. Because the current model is unable to assess certain groups of draftees, it should be deployed in situations similar to those in which it was evaluated. Overall, the results prove that this model can provide significant benefit to potential users. Further, these results can serve a benchmark for future work in this area.

Bibliography

[1] The business of basketball: Forbes releases 22nd annual nba team valuations. https://www.forbes.com/sites/forbespr/2020/02/11/the-business-of-basketball-forbes-releases-22nd-annual-nba-team-valuations/?sh=4525618675ff, 2020. Accessed: 2021-02-28.

[2] Nba salaries. https://hoopshype.com/salaries/. Accessed: 2020-03-20.

[3] G. M. Wong and C. Deubert. National basketball association general managers: An analysis of the responsibilities, qualifications and characteristics. *Villanova Sports and Entertainment Law Journal*, 18, 2011.

[4] S. L. Brook D. J. Berri and A. J. Fenn. From college to the pros: Predicting the nba amateur player draft. *Journal of Productivity Analysis*, 35:25–35, 2011.

[5] M. Mudric. How the nba data and analytics revolution has changed the game. https://www.smartdatacollective.com/how-nba-data-analytics-revolution-has-changed-game/. Accessed: Fall 2020.

[6] P. Z. Maymin. The automated general manager: Can an algorithmic system for drafts, trades, and free agency outperform human front offices? *Journal of Global Sport Management*, 2:234–249, 2017.

[7] A. Jessop. The structure of nba rookie contracts. `https://www.forbes.com/sites/aliciajessop/2012/06/28/the-structure-of-nba-rookie-contracts/?sh=5c029b347299`, 2012. Accessed: 2021-01-10.

[8] University of Pennsylvania. The nba's adam silver: How analytics is transforming basketball. `https://knowledge.wharton.upenn.edu/article/nbas-adam-silver-analytics-transforming-basketball/`, 2017. Accessed: 2021-02-28.

[9] T. Reynolds. 10 things to know as the 2020 nba playoffs begin, 2020. Accessed: 2021-03-20.

[10] National Basketball Association. Nba draft lottery: Schedule, odds, and how it works. `https://www.nba.com/nba-draft-lottery-explainer`, 2020. Accessed: 2021-03-20.

[11] R. Maese. Nba comissioner adam silver says days of one-and-done players will soon be over. `https://www.washingtonpost.com/sports/2019/05/09/nba-commissioner-adam-silver-days-one-and-done-players-will-be-over-soon/`, 2019. Accessed: 2021-03-25.

[12] D. J. Livingstone, D. T. Manallack, and I. V. Tetko. Data modelling with neural networks: Advantages and limitations. *Journal of Computer-Aided Molecular Design*, 11, 1997.

[13] Z. Zhang. Introduction to machine learning: k-nearest neighbors. *Annals of Translational Medicine*, 4.

[14] Leo Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

[15] J. H. Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38:367–378, 2002.

[16] M. Brucher et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[17] T. Kanungo et al. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:881–892.

[18] Gustavo E. A. P. A. Batista, Ronaldo C. Prati, and Maria Carolina Monard. A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explor. Newsl.*, 6(1):20–29, June 2004.

[19] Octoparse web scraping software. `https://www.octoparse.com/`, 2020. Accessed: Fall 2020.

[20] National basketball association website. `NBA.com`.

[21] Real general manager website. `Basketball.realgm.com`.

[22] Draftexpress website. `Draftexpress.com`.

[23] Basketball reference website. `Basketball-reference.com`.

[24] Entertainment and sports programming network website. `ESPN.com`.

[25] O. Goldstein. Nba players stats since 1950. `https://www.kaggle.com/drgilermo/nba-players-stats`, 2018. Accessed: 2020-01-10.

[26] W. H. Atomi N. M. Nawi and M. Z. Rehman. The effect of data pre-processing on optimized training of artificial neural networks. *4th International Conference on Electrical Engineering and Informatics, ICEEI 2013*, 11:32–39, 2013.

[27] Hang Li. Learning to rank for information retrieval and natural language processing. *Synthesis Lectures on Human Language Technologies*, 4(1):1–113, 2011.

[28] K. Esbensen S. Wold and P. Geladi. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2:37–52, 1987.

[29] A. Cheng. Using machine learning to find the 8 types of players in the nba. `https://medium.com/fastbreak-data/classifying-the-modern-nba-player-with-machine-learning-539da03bb824`, 2017. Accessed: 2021-04-01.

Appendix A

Statement of Work and Contributions

No work on this thesis was contributed prior to September 1^st 2020. Work in the fall term was focused on replicating results demonstrated by Maymin in [6] and on developing the web-scraping framework required for data collection. Work in the winter term was aimed at improving upon the original model. This includes making the model dependent on drafting team roster and the design iteration steps described in Section 3.3.2.