Can Machine Learning be used to effectively forecast the performance of aspiring NBA draft picks?

Joshua Adebayo

April 25, 2024

I, Joshua Adebayo, certify that all material in this dissertation which is not my own work has been identified.

Abstract

Accurately predicting promising prospects in the NBA draft is crucial for team strategy and long-term success. This study explores the efficiency of various Machine Learning (ML) models in forecasting the potential of collegiate basketball players transitioning to the NBA. I collected a comprehensive dataset comprising player performance metrics during their collegiate career. Using this dataset, I then implemented several predictive models at increasing levels of complexity, multivariate logistic regression, and neural networks, to evaluate their precision in identifying likely successful athletes. I then analyze the results from these models and consider if other approaches could perform better.

My methodology involved training each model on historical data from 2009 to 2016 and testing their predictive accuracy on the 2010-2017 draft classes All-Rookie selections then using standard performance metrics. In addition, shapley values were also calculated which helped to make each model more interpretable and allowed me to estimate the impact of each variable working in conjunction with Principal Component Analysis. The results demonstrate that neural networks outperform other techniques, indicating a strong potential for ML in this domain. Moreover, I discuss the implications of predictive accuracy in terms of team composition and financial implications, providing a novel insight into the optimization of the NBA draft selection process.

This paper contributes to the sports analytics field by showcasing how ML can transform decision-making in sports drafts and make it scaleable for other professional sports leagues aiming to improve their drafting decisions through data science.

Contents

1 Introduction

The increased access to data is changing the world as we know it, according to Statisca [1], between 2022 and 2025 we should expect the volume of data (created, captured, copied, and consumed) globally to increase by 100% (from 97 zettabytes to 1̃81) which is quite staggering considering from 2010 to the present day there has been a surge of 735% (2 zettabytes to 1̃47). By having this data decision makers around the world have been brought into an age of 'data-driven decision making' (DDDM).

DDDM is defined as 'using facts, metrics, and data to guide strategic business decisions' [2]. Within sports analytics, DDDM is taking root holistically driving customer engagement, expanding partnerships, and most importantly, helping teams win. During my project, I will focus on the sport of basketball (in the NBA) and investigate if the use of Machine learning can aid the draft pick selection process which I believe is the biggest decision each team has to make every year.

1.1 The NBA Draft and its importance

The NBA draft is the annual selection process of the world's brightest talent by teams based on the success in the previous season. Picks 1-14 are assigned by a lottery process only involving the teams that didn't make the NBA playoffs in the previous season and the remaining teams are arranged by their win/loss %. The first 14 teams are given assigned odds for being given the No.1 draft pick in reverse order of their regular season record. The odds are as follows Team 1,2,3: 14.0% ,Team 4: 12.5% ,Team 5: 10.5% ,Team 6: 9.0% ,Team 7: 7.5% ,Team 8: 6.0% ,Team 9: 4.5% ,Team 10: 3.0% ,Team 11: 2.0% ,Team 12: 1.5% ,Team 13: 1.0% ,Team 14: 0.5% [3]. The odds are used to give the most struggling teams a higher chance of getting a higher draft pick to select the better talent. The raffle assigns the first pick and then all subsequent 13 picks. This is known as the NBA draft lottery. The remaining picks are in order of success and repeats giving each team at least 2 draft picks.

The NBA draft is a pivotal opportunity for struggling teams who want to find their future leaders to turn their franchises around, mediocre teams who are looking for one more piece to help them start winning more games, and for the top teams, finding the diamonds in the rough is key for continued success. From the first NBA draft in 1947, the list of draft successes and failures is long but what's clear is that the organizations that drafted well have been able to climb to the top sustainably and remain there. In Table 1 there is data on the championship teams from the past 11 years and their number of drafted players who contributed to their playoff success along with the ranking of the teams' financial expenditure in Salary Cap and Luxury Tax. Data on the table was extracted from [4].

The NBA Salary Cap is a limit on the total amount of money that NBA teams are allowed to pay their players. This cap is designed to help maintain competitive balance within the league by preventing teams with higher revenues from outspending others on player salaries. The salary cap is calculated each year based on a percentage of the league's revenue from the previous season, including income from broadcast rights, ticket sales, and merchandising. The cap can fluctuate from year to year, reflecting changes in the league's overall financial health.

The Luxury Tax is a mechanism that works in conjunction with the salary cap to further ensure

Year	Team	Drafted Players in Playoff Roster	Salary Cap (Rank)	Luxury Tax (Rank)
2022-23	Nuggets	6	21	8
2021-22	Golden SW	8	30	1
2020-21	Bucks	3	17	7
2019-20	LA Lakers	0	5	22
2018-19	Toronto Raps	2	18	3
2017-18	Golden SW	5	1	1
2016-17	Golden SW	6	1	14
2015-16	Cavaliers	3	1	1
2014-15	Golden SW	6	4	15
2013-14	SA Spurs	5	17	20
2012-13	Miami Heat	2	3	3

Table 1: NBA Championship Teams in relation to Drafted Players in Playoff Roster, Salary Cap and Luxury Tax Ranking

competitive balance. It is a tax that teams must pay for the portion of their payroll that exceeds a certain threshold, set higher than the salary cap. The luxury tax is designed to be punitive and progressive, meaning the more a team exceeds the threshold, the higher the tax rate it pays on the excess amount. This system discourages teams from excessive spending on purchasing/trading players and paying large salaries, even if they are willing to pay the salaries themselves. The revenue generated from the luxury tax is typically redistributed to teams with lower payrolls, further promoting competitive balance.

Of course, there are exceptions, named after NBA legend Larry Bird, Bird Rights allows teams to exceed the salary cap to re-sign their draftees, provided the player has been with the team for three full seasons without being waived or changing teams as a free agent. This exception enables teams to pay their players more money for a longer duration than any other team can offer. This is another reason why drafting right is so crucial financially.

In Table 1, the team that appears the most is the Golden State Warriors, and by no surprise, they have a high amount of drafted talent compared to the other championship teams. As time has gone on their Luxury Tax payments have had to increase to retain their talent but most recently in the 2021-22 season, they had the lowest Salary Cap payment in the league during their Championship run due to having so many draftees in their squad. The other 7 teams who have managed to win the Championship have been unable to replicate this level of success.

1.2 Machine Learning

The rapid development of Machine Learning and AI has seen that the deployment of ML has the potential to assist teams all over the NBA to better select young players, making the league a lot more competitive and helping teams to rise to the top sustainably and stay competitive for years to come. Additionally, this heightened level of competitiveness would be great for the fan experience, seeing teams that have been synthesized perfectly go head-to-head would surely bring more fans to the games along with endless financial benefits for teams individually and the league as a whole.

Because of all the metrics recorded in the game of basketball Neural Networks may be able to thrive here. Neural networks are capable of modeling complex, non-linear relationships between inputs and outputs. This capability allows them to solve problems that are difficult or impossible for traditional algorithms to handle. Additionally, NN can be quite robust to noise in their input data, making them useful in real-world applications. In the world of sports, this is particularly important because of the inherent randomness of having humans in the game. Injuries, mental health, and relationships, all add to the noise that can affect a player's performance NN's are best equipped to somewhat account for this.

2 Literature review overview and aim of project

2.1 How ML is used in team selections in sports

The use of data analytics in sports is already widespread within the industry and even everyday fans are becoming more adept at sports statistics just from listening to broadcasters trying to breakdown games/ players. Internally, teams have been investing in this as well for decision-making. The critically acclaimed movie 'Moneyball' (based on the 2002 baseball season for the Oakland Athletics) shows the early beginnings of using data for player acquisition. Twelve years later, this is a lot more commonplace. Regarding Machine Learning, the sporting world is yet to fully adopt its practices and techniques into team decisions. The majority of the papers with substantial work related to this topic are in the application of ML in football, this is no surprise since football is the world's most popular sport in viewership and market share commercially. Nonetheless, the use of ML in recruitment in football and other sports may bring insights translatable to basketball.

2.1.1 Neural Network Regressions

From my research in ML used in team selections, Neural Networks have been very popular. This comes as no surprise for all the reasons mentioned in the last chapter. With the use of NN's, the researchers approach their given problem with either a regression, an attempt to quantify the performance of a player/team given the parameters, alternatively, there is classification. When a classifier is being used, the aim is typically to group players by performance or by position.

Al-Shboul et al [5], and Iyer et al [6] implement NNs that at some point quantify player performance, in [6] players were ranked and the highest scoring players from each country were selected to predict the team selections in the 2019 cricket world cup. During this investigation, the authors broke down the game of cricket into batting and bowling and trained separate NNs to produce ratings for each aspect and selected players based on the joint score. This process was fairly accurate as 71% of the players predicted by the model were then picked for their national team. Although systematic, the clear shortcoming in this approach is that fielding, another aspect of the game, wasn't considered in this model. By missing this, a comprehensive rating of a player's performance can't be conclusive. In the sport of basketball, there are two clear parts of the game, offense, and defense, so this approach may be effective; however, within offense and defense, there are subcategories respectively. For offense alone, there's three-point shooting, mid-range scoring, and finishing by the rim so generating an accumulative rating may be unfeasible.

Similarly, Al-Shboul et al [5] attempt to pick the best squad for each premier league team but consider player rating using just one score. But with this methodology, the number of parameters was vast, additionally, because there are several positions in football it makes the relationships with each parameter and their weighting difficult to gauge. In this study, they came across this issue and struggled with poor accuracy (just 54%) even though the model used 10 years of English Premier League data with over 2500 players sampled.

Because of all of this, it will be interesting to see how using a classifier may help to produce a more viable product.

2.1.2 Neural Network Classifications

When trying to find future successful players in football, Musa et al [7] used NNs. During their study, dimensionality reduction was performed via Principal Component Analysis (PCA). When using just three (out of nine-teen) components, alternative classification algorithms were used (k-Nearest Neighbours and Random Forests) as control experiments to the NN. From this comparison, the authors found that with a much-reduced number of parameters, the accuracy of a simpler model can rival the accuracy of a complete NN, although both are mediocre (60%). However, other metrics may have been able to bring more insights out of the study (F1-Score, precision, and recall). In a similar study with Norwich FC [8] there is also a focus where again, unsupervised k-means clustering is used to group players by play style when in lower dimensions. This could prove useful as it could allow managing staff on the team to see what roles current players may be thriving or struggling in and from there be able to find a potential replacement if the player was to become injured, lose form, or leave the team. This could be applicable in basketball as well.

2.2 How NBA teams draft players

ML is becoming more integrated within the NBA and most NBA teams now have at least one Data Scientist to accompany their analytics teams. ML is typically used to understand a team's performance and that can be of their opponents by breaking down plays, shot efficiency, and the literal science behind the ball going into the hoop (angle, velocity, and ball rotation). In terms of the draft there is no (public) information on the use of ML, just the traditional recruitment techniques. Currently, the NBA drafting process is a year-long process with multiple steps but it all starts with eligibility.

For a player to declare for the NBA draft, they must be at least nineteen years old within the same calendar year of the draft or must have graduated high school at least one year prior [9]. Players remain eligible for the draft till they are 23 years old, this leaves a four-year window for young players to impress teams enough to be deserving of a draft pick.

During the years of eligibility, each team has scouts look out for talented players. Although the college season starts in August, NBA interest doesn't peak till during the months of February and March. During February, there are regional tournaments in each district, and the winners/best-performing teams in these tournaments get invited to 'March Madness' where the winning team is crowned as the College National Champion. By the end of the tournaments, recruiters will contact top-performing players from participating teams and the most elite players in less successful teams and invite them to the NBA Combine.

During the NBA combine the players are evaluated on their athletic capabilities, physical attributes, and their on-court skill capabilities [10]. The athletic testing consists of:

- No step vertical jump
- Max vertical jump
- Bench Press
- Lane Agility (lateral movement, stop-start acceleration)
- 3/4 Court Sprint

The physical testing measures:

- Height without shoes
- Height with shoes
- Weight
- Wingspan
- Standing Reach
- Body Fat%
- Hand size/width

Regarding the skills evaluation, this is done differently for each position group (split into guards and forwards) but then they are put into random teams to compete. In a way, it's a very high-level basketball trial. During the combine and in the months ahead of the NBA draft, teams are able to invite up to 20 players to their facilities for additional individual workouts for a closer inspection of a player's game. From all of this, the staff of each organization is trying to determine if a player passes the 'eye test', that the player has the capability to transfer their success in college or overseas to the big leagues.

The concept of the 'eye test' is common among all major sports and is the idea that not everything is measurable and not all measurements matter, and the limit of a player's potential can be seen through watching them play enough [11]. Basketball purists rely heavily on the eye test and even argue to ignore clear stats if it doesn't go along with their intuition. At times this works well for organisations but more times than not it leads to unsuccessful drafting. Four-time NBA Champion and three-time Most Valuable Player Stephen Curry has often spoken about how coaches told him early in his career that they did not see him playing a significant role in a winning team,[12], even after three successful seasons in college averaging 20+ points each year and receiving many collegiate accolades. Inversely, the list is long of players who were thought of as the next big thing but then went on to have average or even poor careers examples being Kwame Brown, Greg Oden, and Sam Bowie who were all drafted #1 in their respective drafts but weren't able to bring their teams into championship contention [13].

2.3 Project steps and objectives

From my research into the NBA draft process and the various applications of ML in team selections/talent acquisition in sports, I think it is clear that the proper use of machine learning can help teams to better select players to improve their teams and financially sustainably build squads.

My dissertation aims to assess the applicability of machine learning models in predicting the performance of NBA draft picks based on college statistics. This involves the development and comparative analysis of two types of models: a neural network and a logistic regression model. The classification task will group the players into All-Rookie, and Non-All-Rookie groups. Each season the NBA selects 20 newly drafted players who made the biggest impact on their teams during their rookie season. I will develop these models using historical data and train them using advanced machine learning libraries (Scikit-Learn). This foundational phase is critical and has been prioritized due to its significance in supporting all other aspects of the research.

Following the model development, I will conduct a detailed evaluation of each model's performance, assessing metrics such as accuracy, recall, precision, and the F-1 score. This analysis, planned for the mid-phase of the project, will help determine the most effective model for practical application. This structured approach, aligned with SMART criteria, provides a solid framework for achieving meaningful and impactful results.

3 Experimental Design

3.1 Technology choice

During my project, I will be using Python within a Google Colab virtual environment. Python was selected for its widespread adoption within the machine learning community, particularly for neural network development, attributed to its user-friendly syntax. The efficiency it offers in prototyping sets it apart from alternatives like Java or C. Over time, Python has amassed an extensive array of libraries catering to various applications, particularly boasting advanced tools for data analysis, scientific computing, and machine learning which are advantageous within my project (Scikit-Learn, Numpy, Pandas, Matplotlip, Shap).

Scikit-Learn is pivotal for data mining and analysis within the Python ecosystem and supports classification, clustering, dimensionality reduction, and other preprocessing tasks. Leveraging Numpy for its computational tasks, Scikit-Learn offers a considerable performance improvement over purely Python-implemented methodologies.

Numpy and Pandas enhance Python's data manipulation, especially for vectors and matrices, alongside providing elementary statistical and linear algebra tools useful across different applications. Its use of C++ for computation underpinnings significantly accelerates processing speeds. Matplotlib and Shap are used for plotting, Shap has the additional functionality of being able to further analyze' and graph the behind-the-scenes of models, helping to make the processes more interpretable as opposed to the typical black-box methodology.

The key reason for working within Google Colab was to allow me to connect to a custom Google Compute Engine (GCE) virtual machine and therefore be able to utilize the GPU to train neural networks significantly enhancing the efficiency and speed of the training process. GPUs are designed for high throughput of calculations because of their parallel structure. This makes them efficient for the matrix and vector operations that are commonly used in deep learning

3.2 The Data

The data used came from merging two independent datasets, both of which I found on Kaggle. Firstly, there's the Advanced College statistics, [14]. The raw data had 25719 entries and had 65 columns comprised of players' stats and information about a player such as their college team and the conference they play in. All entries are from 2009 to 2021. The second dataset was a collection of rookie data from 1980 to 2016 with 2014 entries and 23 columns [15].

3.3 Data preprocessing

The College dataset had an extensive number of parameters, a lot of which aren't recognized by the NBA so I dropped these columns as well as the columns that aren't impacted by a player's performance. This left just 14 columns from 65 initially. Next, I dealt with duplicate entries from players who played multiple years of college basketball so I dropped all extra entries just leaving the stats from each player's final year of draft eligibility (61061 remaining entries).

From here, I merged (inner) the college and rookie datasets to eliminate college players who didn't make it to the NBA. this join revealed that from 2009 - 2016, 346 college players were drafted into the NBA. Since I have a classification task, I dropped the rookie stats of all players and instead added a column of whether a player who got drafted then went on to make the All-Rookie team that year (All-Rookie data was extracted from the NBA website [16]).

3.3.1 The Final Dataset

GP	FT	FTA	FT	3P	3PA	MIN	OREB	DREB	REB	AST	STL	BLK	PT	Rookie
33	64	114	0.56	0	0	22.91	1.42	3.30	4.73	0.85	0.45	0.33	7.58	False
35	66	82	0.81	29	94	30.29	1.31	3.23	4.54	3.82	1.71	0.69	7.06	False
31	183	223	0.82	30	94	35.03	0.90	3.84	4.74	4.32	1.42	0.45	19.51	False
35	213	257	0.83	42	114	34.03	1.31	4.43	5.74	2.46	1.23	0.51	18.31	False
37	124	174	0.71	69	180	32.27	3.03	4.65	7.68	2.401	1.54	1.02	16.46	True

Table 2: Final NBA Draft Pick Dataset(top 5 entries)

After preprocessing, I had a dataset ready to use, Table 2. The dataset contained the advanced stats of drafted college basketball players between 2009 and 2016 during their final year of eligibility for the NBA draft and if they went on to make the NBA All-Rookie team during their rookie year (58 All-Rookie players to 287). There are 346 entries and 14 columns. The columns are as follows.

GP is the number of games a player played in that season. FT is the number of free throws a player attempted in a year while FTA is the number of free throws attempted. FT is the percentage of free throws a player makes out of the total attempted. 3P indicates the total number of successful three-point field goals the player makes. 3PA is the total number of three-point field goal attempts by the player.MIN denotes the total number of minutes a player has spent on the court during games. OREB is the number of rebounds grabbed by the player on the offensive end of the court. DREB is the number of rebounds grabbed by the player on the defensive end of the court. REB is the total number of rebounds (both offensive and defensive) caught by the player. AST indicates the total number of assists made by the player. STL represents the total number of steals made by the player. BLK is the total number of shots blocked by the player. PT is the total number of points scored by the player over the season. Rookie is a boolean variable indicating whether the player was selected for the NBA All-Rookie team (True) or not (False).

3.4 Dimensionality Reduction

Dimensionality reduction is a key component of my task. For this, there are several techniques that I could've used but I found Principal Component Analysis (PCA) best for my study. If I aimed to use dimensionality reduction to better visualize the data I could have used t-distributed stochastic neighbor embedding (t-sne) or Uniform Manifold Approximation (UMAP) but with these techniques, the topological characteristics of the data would be lost which would be at the detriment of future modeling. Additionally, there are hyperparameters within t-sne and U-MAP that would need to be tuned for optimal performance.

1	2	3	4	5	6	7	8	9	10	11	12	13
7.666	3.076	0.973	0.812	0.686	0.538	0.422	0.369	0.207	0.140	0.087	0.044	0.013

Table 3: Eigenvalue for each component

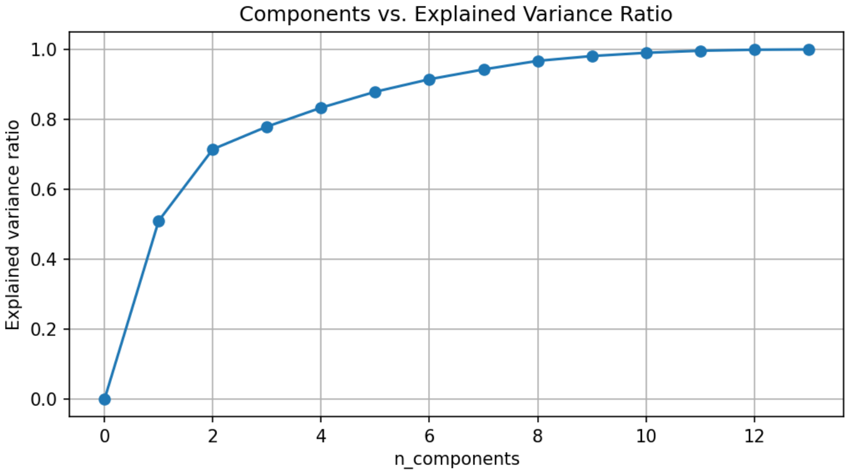

Figure 1: Explained variance Vs Number of components

3.4.1 Principal Component Analysis

PCA transforms the original dataset into a set of linearly uncorrelated variables known as principal components.

The first step involves standardizing the range of the continuous initial variables so that each one of them contributes equally to the analysis. This means subtracting the mean and dividing by the standard deviation for each value of each variable. Next, PCA computes the covariance matrix to understand how the variables of the input data vary from the mean with respect to each other, considering the covariance is a measure of the degree to which two variables are linearly associated. The covariance matrix is then decomposed into its eigenvectors and eigenvalues. The eigenvectors (principal components) determine the directions of the new feature space, and eigenvalues explain the variance of the data along the new feature axes ordered in descending order of eigenvalues. This entire process can be done using packages in Scikit-Learn.

For my investigation, PCA is perfect as I can train models with the data at different levels of dimensionality. Figure 1 illustrates my results from running PCA on my data. The graph shows that the first two components account for 70% of the explained variance in the data and the subsequent 12 components account for just 30%. The Kaiser criterion suggests that components with an eigenvalue less than 1 should be discarded (only leaving the first two components, Figure 2) but this mapping of the original dataset is very poor at separating the two classes. By rounding up and including the third component, however, Figure 3, there is a much better visualization to distinguish the two groups. Eigenvalues shown in Table 3.

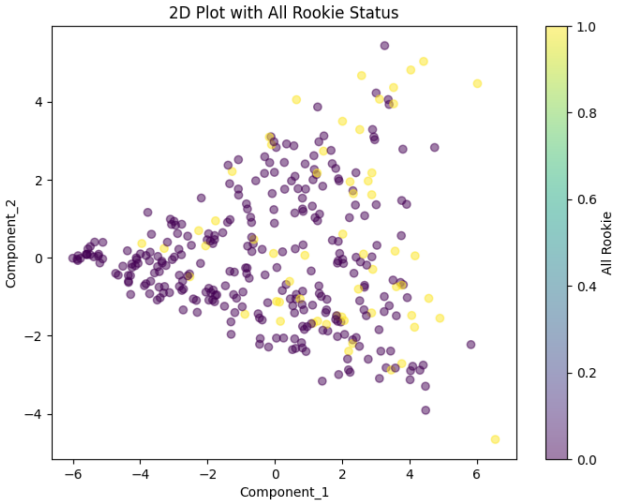

Figure 2: Visualisation of 2 components

3.4.2 Shapley Values

During PCA, the original features of the data are transformed into a set of linearly uncorrelated components. This transformation leads to the loss of original feature names, making it difficult to interpret which variables are most important in the model but I am training the NN and LR models I plan to calculate the shapley values within each model. Shapley values are a concept from cooperative game theory that has been adapted to explain the contribution of individual features to a prediction made by a machine learning model. This is achieved by calculating the average marginal contribution of each feature when it is included in versus excluded from a subset of features. This approach offers insight into how each feature influences the model's prediction, thus helping to make my results more interpretable [17]. The general representation of the formula for the Shapley value for the ith feature is as follows:

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} [v(S \cup \{i\}) - v(S)]$$

where $\phi_i(v)$ is the Shapley value for feature i, N is the set of all features, S is a subset of features without i, and $v(S)$ is the prediction model evaluated with the features in set S. The sum is taken over all subsets S of the set N that doesn't include i.

If a principal component is heavily influenced by a particular subset of features according to the Shapley values, I can infer that these features have strong relationships either in variance explained or in their impact on the model' s predictive accuracy. This methodological approach will be crucial for explaining model behavior in terms understandable by players,coaches, and recruiters who may not be familiar with machine learning intricacies.

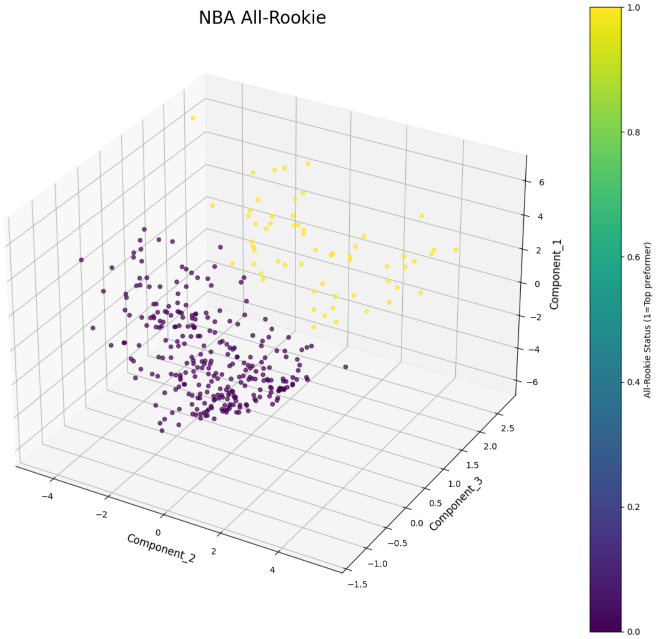

Figure 3: Visualisation of 3 components

3.5 Neural Network - MLP

The basic structure of a neural network is aimed to emulate some aspects of human brain function and is designed to recognize patterns and solve complex problems. It consists of three types of layers (input, hidden, and output) connected working with an activation function. The input layer is the initial phase where the neural network receives the raw data. Each neuron in this layer represents a feature of the input data. After the input layer, one or more hidden layers perform computations through their neurons. These layers transform the inputs into something that the output layer can use. They are called 'hidden' because they are not visible to the external systems, making this a black-box process. Each neuron in the hidden layers processes the inputs by performing weighted sums followed by a non-linear activation function. The final layer is the output layer, where the neural network produces its predictions or classifications based on the transformed inputs from the hidden layers. For a classification task, the output layer will typically use a softmax function to normalize the output to a probability distribution over predicted output classes. For regression tasks, it might consist of a single neuron that outputs a continuous value.

The layers are interconnected by nodes, or "neurons" which have associated weights and biases. During the training phase, the NN adjusts these weights and biases using an optimization algorithm like gradient descent, based on the difference between the network's prediction and the actual data (the loss or error). Over time, the network learns to reduce the error, and hence, improves its predictions or classifications. This learning process is iterative and involves forward propagation of inputs through the network to generate an output, and backward propagation of the error to update the weights and biases, a process known as backpropagation. Activation functions are crucial because they allow neural networks to approximate non-linear mappings between inputs and outputs. Common examples of activation functions include the sigmoid function, hyperbolic tangent (tanh), and rectified linear unit (ReLU).

I implemented an NN Classifier utilizing the Multi-Layered Perceptron - "MLPClassifier" function from Scikit-Learn, using the All-Rookie status as the target variable and the components as the input data. From the default parameters, I only changed the number of hidden layers as 100 is excessive for the complexity of the data and would likely lead to overfitting. Instead, I will be using 10 layers.

3.6 Multivariate-Logistical Regression Classifier

To run in tandem with the NN, I ran a multivariate-logistical with the same input and target variables as the MLP at the same level of dimensionality. I was able to use another function from Scikit-Learn for this.

Multivariate Logistic Regression extends the binary logistic regression model to accommodate multiple categorical outcomes. The core of this model involves a set of independent variables which can range from nominal to ratio-level, influencing the probability of the outcomes.

The computation of these probabilities in the multivariate case employs the softmax function, a generalization of the logistic function used in binary classification. The softmax function takes the linear combinations of the predictors, applies an exponential transformation, and then normalizes these values to ensure that the resulting probabilities across all classes sum up to one. The training of the model revolves around maximizing the likelihood of the observed data, which is typically achieved through iterative optimization techniques such as gradient descent.

Upon training, the model assigns weights to the features that reflect their importance in predicting the outcome. When new data is presented, the model uses these weights to predict class probabilities. The category with the highest probability is selected as the model' s output. These weights can be interpreted as the influence of each unit change in the predictor variables on the log odds of the outcomes.

3.7 Evaluation metrics

To evaluate the performance of the model I will calculate the accuracy, recall, precision, and the F-1 score.

Accuracy is the ratio of the number of correct predictions to the total number of input samples. Accuracy = $\frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$. Accuracy may return misleading results since there is an imbalance between the 2 classes in the dataset (58 All-Rookie players to 287 others). For example, a model that predicts False for all players would still achieve a misleadingly high accuracy of 83.2%. Recall measures the proportion of actual positives that are correctly identified by the

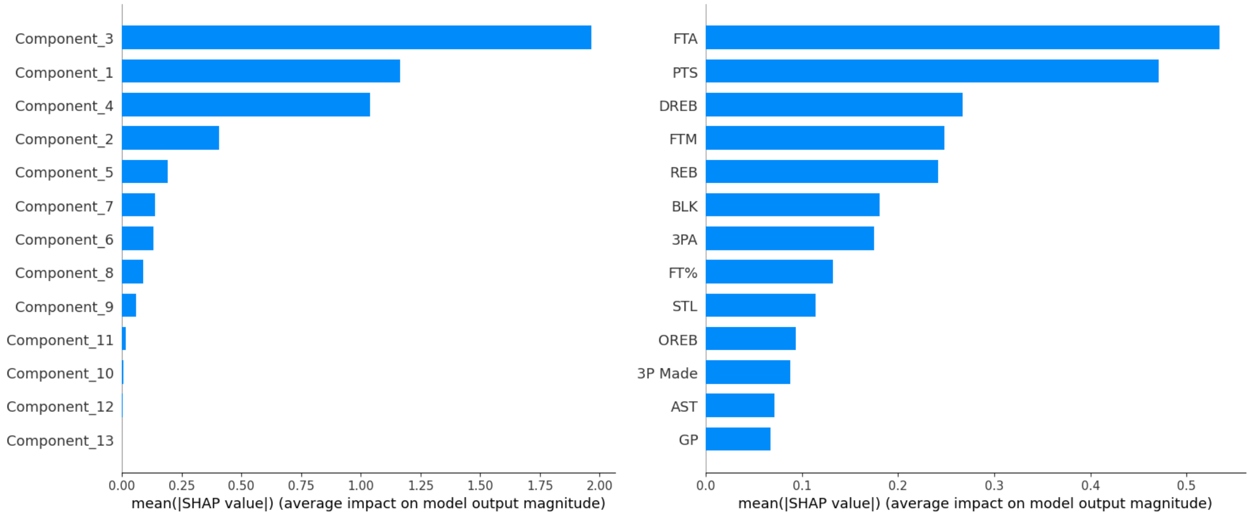

Figure 4: Approximating PCA true values with shapley values

model. Recall $= \frac{\text{True Positives}}{\text{True Positives+False Negatives}}$. Inversely, the precision is the ratio of true positives to all predicted positives. Precision $= \frac{\text{True Positives}}{\text{True Positives+False Positives}}$. Precision, Recall, and Accuracy are all expressed as a percentage. The F-1 score is a metric aimed at balancing precision and recall as a harmonic mean of precision and recall, taking both false positives and false negatives into account. It is a more robust measure than accuracy, especially for imbalanced datasets. The F-1 Score reaches its best value at 1 (perfect precision and recall) and its worst at 0. It's defined as: F-1 Score $= 2 \cdot \frac{\text{Precision} \times \text{Recall}}{\text{Precision+Recall}}$.

4 Results and Analysis

4.1 My Results

Before running simulations to get results I calculated the shapley values for the standardized dataset and results from PCA, shown in Figure 4. Assuming that the variability in the gaps between each parameter is because of the calculation of eigenvectors, this data helps to retain the name of each component after PCA. By calculating the shapley values while I ran each model I was able to produce Figures 5 and 6 these show that in the NN and LR models component 3, free throw attempts, had by far the largest magnitude on each model. With 3 parameters, the impact of each component is almost identical as component 1 is 0.04 greater. But at the higher levels of complexity, the impact of each component becomes a lot more variable. The only constant is that component 3 has the most significant mean shapley value.

Tables 4 and 5 show my results from training NNs and Multinomial Logistical Regressions with the college data with 3,4 and 5 parameters. In this section, I will provide a detailed comparative overview focusing on each performance metric individually and provide insights into how each model

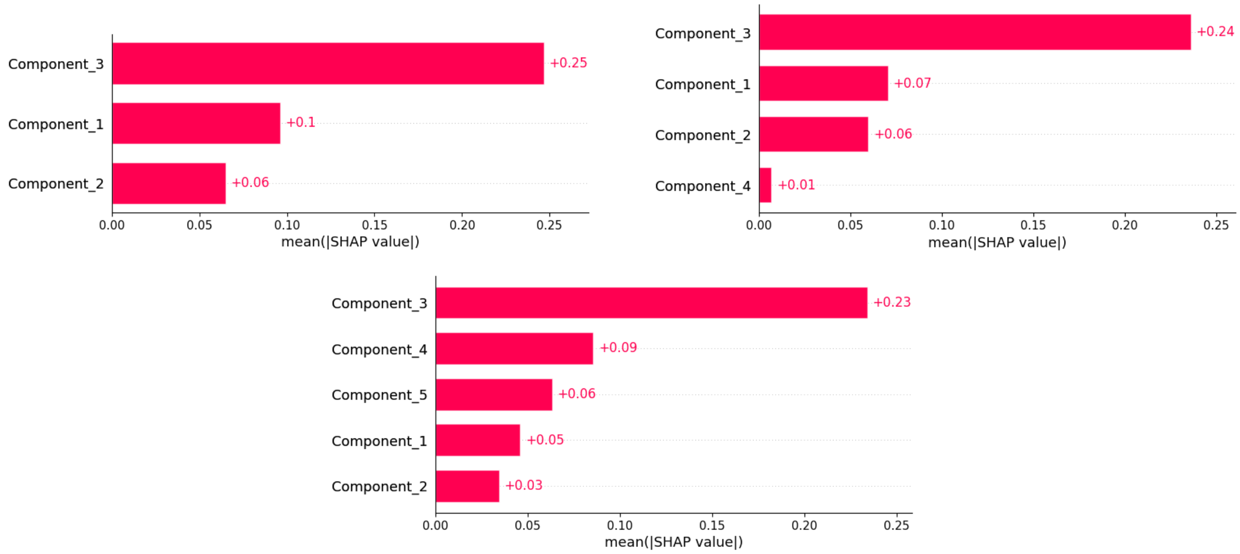

Figure 5: Shapley values in NN as complexity increases

Table 4: Neural Network Results table

Number of components	Accuracy	Precision	Recall	F1-Score
3	0.954	0.84	1	0.91
4	0.966	0.75	1	0.857
5	0.942	0.923	0.765	0.839

performs relative to the other, offering some potential explanations and implications of these findings.

The NN shows consistently high accuracy across all component configurations, peaking at 96.6% with 4 components. Even at its lowest, with 5 components, the accuracy remains high at 94.2%.On the other hand, in the LR the accuracy ranges from a low of 81.7% with 4 components to a high of 88.7% with 3 components, consistently underperforming relative to the Neural Network. From this, the Neural Network appears better suited for achieving higher overall prediction correctness regardless of the number of components used, suggesting robustness across various feature sets.

The Logistic Regression shows a very varied precision, achieving 100% with 4 components—although this is misleading due to the extremely low recall—and as low as 33.3% with 5 components

Table 5: Multivariate Logistical Regression Results table

Number of components	Accuracy	Precision	Recall	F1-Score
3	0.887	0.667	0.143	0.235
4	0.817	1	0.087	0.16
5	0.86	0.333	0.06	0.111

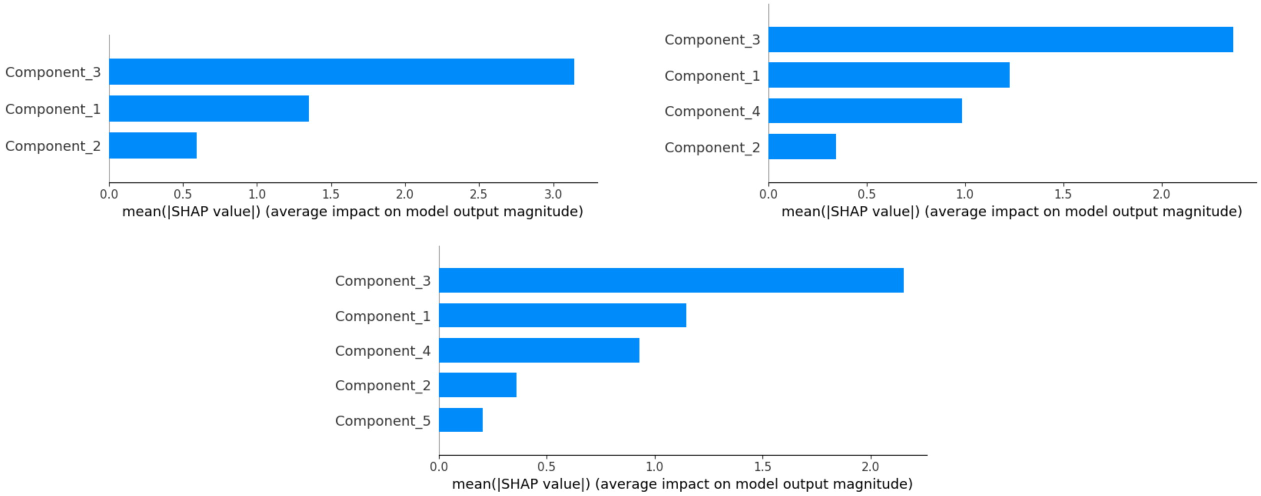

Figure 6: Shapley values in LR as complexity increases

The Neural Networks exhibit the highest precision of 92.3% with 5 components but drops to 75% with 4 components, indicating some variability based on feature set configuration. While the Neural Network results show better average precision, the variability with Logistic Regression suggests potential overfitting. However, this can also be because of the imbalances in the dataset as mentioned in 3.7 which the LR model has been victim to.

The NN achieves perfect recall with 3 and 4 components but has a notable drop to 76.5% with 5 components. Inversely, the LR struggles significantly with recall, not exceeding 14.3% and going as low as 6% with 5 components. This suggests that the Neural Network is far superior in identifying all relevant cases in scenarios with fewer components, while the Logistic Regression fails to identify a substantial portion of positive cases, limiting its practical applicability.

As expected, the LR performs poorly, with F1-scores ranging from 11.1% to 23.5%, reflecting significant issues in balancing precision and recall. The NNs maintain relatively high F1-scores, with the best at 91% for 3 components and the lowest at 83.9% for 5 components. The higher F1 scores for the Neural Network indicate it better balances identifying relevant cases and minimizing false positives, making it more reliable for practical use.

With limited entries and a largely imbalanced dataset, the results show that the Neural Network consistently outperforms the Logistic Regression in all metrics across different levels of complexity. This superiority suggests that the Neural Network is better equipped to handle variations in the dataset and feature configurations, likely due to its ability to model complex relationships in the data more effectively than Logistic Regression. From analyzing the poor results from LR, if I were to redo my investigation I would consider another method to run in comparison to NNs such as Support Vector Machines (SVM).

4.2 Support Vector Machines

At the core of SVM operation is the concept of finding the best hyperplane that separates data points into distinct classes while maximizing the distance from the nearest data points on either side, known as the margin. The data points that influence the position of the hyperplane are called support vectors, hence the name of the method. SVM is particularly effective when dealing with high-dimensional spaces, making it suitable for datasets with a large number of features like what I am investigating. It's also memory efficient because it uses only a subset of the training data to make decisions; only the support vectors are needed to define the hyperplane. The adaptability of SVM is further enhanced by the kernel, which allows them to work well even with non-linear data separations. This is done by transforming the data into a higher-dimensional space where a linear hyperplane can be used to separate classes. In light of this, SVM sounds capable of competing with NNs in my research.

4.3 Regression Vs Classification

A pivotal decision during my research was to go forward with a classification task as opposed to a regression. The regression approach is commonly used, creating a metric to evaluate a player's performance such as [6] mentioned in Section 2.1.2 and may have been a better approach than using the NBA All-Rookie team selections. Firstly, regression tasks predict continuous variables, whereas classification tasks predict categorical variables. Predicting a player' s performance using regression (like scores, assists, rebounds per game, etc.) can provide more nuanced insights compared to classifying whether a player belongs to an All-Rookie team or not. This continuous output captures varying degrees of player performance rather than simply categorizing players, which might miss subtleties in their potential or skills.

Moreover, regression allows the modeling of a broader spectrum of outcomes, capturing the precise impact or contribution of a player. For instance, a player might not make the All-Rookie team but still have impressive stats that are better represented through regression. This can help in identifying undervalued players who might not be the best rookies but are still promising. Finally, regression analysis can also integrate more complex relationships and interactions between variables (like minutes played, team dynamics, etc.) that might affect a player's performance. This complexity can be crucial in sports analytics where numerous factors influence outcomes.

In summary, while classification might straightforwardly identify whether a player is considered a top rookie and is a measurement defined by the NBA itself, regression provides a detailed, quantitative measure of player performance and potential. This makes regression more suited for scouting and player development decisions during the NBA draft.

5 Conclusion

5.1 Aim of the project and intermediate steps

When starting this paper I set out to investigate if ML can be effectively used in selecting the most promising young players. The aforementioned steps involved finding the importance of each parameter and then running an NN and LR at increasing levels of complexity. I have been able to do this all and from the results can conclude that the implementation of machine learning within the NBA draft is promising. With more development, I believe that executives in NBA teams will be able to utilize NNs in making data-driven decisions in the NBA Draft. Also, I have been able to reveal aspects of the game that are statistically impactful on the chance of being selected for the All-Rookie team in a given year which young players can focus on developing (points, defensive rebounds, and free throws attempted and made).

In the years to come, I anticipate a league-wide switch to a more quantitative approach to the drafting process, and the 'Eye test' plays a much smaller part in the decision. This quantitative approach would allow teams to manage risk with more certainty when it comes to the NBA draft and the use of ML by a team would help them to build a cost-effective team to challenge for the NBA championship. Of course, there is always randomness such as career-ending injuries that can affect anyone which are difficult to preempt; however, machine learning can be used here as well. ML and computer vision (CV) are increasingly being applied across various domains, including sports, where they play significant roles in injury prevention and enhancing athlete performance. The integration of these technologies helps in early detection of potential injury conditions, improving training regimens, and even in real-time monitoring to prevent injuries using wearable technology running simulations and modeling during recovery and rehabilitation.

What's exciting is that the NBA isn't the only sports league that has a drafting process. The NFL, MBL, and the MLS all have their own drafts and also adhere to salary cap restrictions thus, this research can be transferred to American Football, Baseball, and Football assist teams.

5.2 Further steps

There are several potential avenues of future development, mentioned in Sections 4.1 & 4.2 including changing the ML task to a regression or potentially keeping the current experimental design but implementing another classification algorithm such as SVM instead of LR to compete with the performance from NNs.

Something to take note of is that the data I used was just with college players which has historically been the main avenue for players to get drafted so there was a lot of historical data to work with. However, over the years the game has increased in global reach and thus has seen players get drafted in other ways. Two of the last 10 winners of the Most Valuable Player Awards (Nikola Jokić and Giannis Antetokunmpo) weren't drafted from the college route but through the Euroleague instead. Additionally, 12 of the 58 drafted players in the most recent NBA Draft didn't attend college including number 1 pick Victor Wembenyama [18]. If I were to continue this investigation or if anyone is to continue the research I suggest first incorporating players with non-traditional paths to the NBA. Utilizing the Python Selenium library would help with web-scrapping to obtain the data of draft eligible players abroad. However, as the number of players is so small, I suspect that the models would be susceptible to overfitting.

After that, a potential next step could be to optimize the parameters (and hyperparameters) in the NN particularly the weight of edges in the NN. Properly optimized weights enable the neural network to learn efficiently from the training data, converging to a solution that minimizes the error in predictions. Weights that are not optimized well may lead to slow convergence or no convergence at all, where the network fails to adjust adequately to the patterns in the data. Optimizing weights is also crucial for the model's ability to generalize to new, unseen data such as players coming from non-traditional routes before the draft. Evolutionary Algorithms, (EAs) can be utilized here, neuroevolution. EAs are algorithms that use mechanisms inspired by biological evolution, such as reproduction, mutation, recombination, and selection. Candidate solutions to an optimization problem play the role of individuals in a population, and the fitness function determines the quality of the solutions. The algorithm evolves the population of candidate solutions over many generations towards better solutions. There is extensive research in this field, notably "Evolutionary Algorithms and Neural Network" [19], that can be applied to NNs and the NBA Draft.

Other important parameters to look into would be the number of hidden layers and iterations (epochs) that the model would take, excessive of either would increase the risk of overfitting.

5.3 Final thoughts

Reflecting on this academic year, I am proud of the body of work that I have been able to produce in my dissertation. By applying knowledge from my academic career and learning additional pieces I have written and coded work that both basketball enthusiasts and data scientists can follow and has the potential to help shape the draft process within the NBA and other sports. Technically, I was able to plan and execute an exploratory study end to end which used/developed my prowess in data cleaning, manipulation, and analysis as well as debugging. Although I see this project as a success, by critically analyzing each part of the study I have made it easy for either myself or other contributors to carry on the investigation independently or with an institution or sports team. Code has been made open-source on GitHub [20].

A self-inflicted challenge that I had to overcome during the development of my dissertation was my time management and indecisiveness in crafting a problem statement. Spread out from September - May I might've been able to accomplish some of the future steps in Section 5.2; however, due to several substantial changes I essentially had to restart the project in January and had to develop at a much-accelerated timeline to be able to finish this substantial piece of work. In hindsight, thorough research within all available projects and asking intelligent questions to all potential supervisors would've led me to define a concrete project much earlier and may have led to greater insights. Nonetheless, powered by my genuine passion for basketball and regular check-in sessions I am very pleased with the development, results, and polish of my study.

References

[1] Petroc Taylor. “Amount of data created, consumed, and stores 2010-2020, with forecasts to 2025”. In: *https://www.statista.com/statistics/871513/worldwide-data-created/* (2023).

[2] Mark Nelson. “Beyond The Buzzword: What Does Data-Driven Decision-Making Really Mean?” In: *https://www.forbes.com/sites/tableau/2022/09/23/beyond-the-buzzword-what-does-data-driven-decision-making-really-mean/?sh=1fefdebb25d6* (2022).

[3] “2024 NBA Draft Lottery: Odds, history and how it works”. In: *https://www.nba.com/news/nba-draft-lottery-explainer* (2024).

[4] “NBA Team Salary Cap and Luxury Tax Tracker”. In: *https://www.spotrac.com/nba/tax/2012/* (2024).

[5] Rabah Al-Shboul et al. “Automated Player Selection for a Sports Team using Competitive Neural Networks”. In: *International Journal of Advanced Computer Science and Applications* 8.8 (2017). DOI: 10.14569/IJACSA.2017.080859. URL: http://dx.doi.org/10.14569/IJACSA.2017.080859.

[6] Subramanian Rama Iyer and Ramesh Sharda. “Prediction of athletes performance using neural networks: An application in cricket team selection”. In: *Expert Systems with Applications* 36.3, Part 1 (2009), pp. 5510–5522. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2008.06.088. URL: https://www.sciencedirect.com/science/article/pii/S095741740800420X.

[7] Rabiu Musa et al. *Machine Learning in Team Sports: Performance Analysis and Talent Identification in Beach Soccer Sepak-takraw*. Jan. 2020. ISBN: 978-981-15-3218-4. DOI: 10.1007/978-981-15-3219-1.

[8] Monica Benjamin et al. “Best Fit Player Acquisition and Squad Personnel System in Football for Sports”. In: *2022 1st International Conference on Computational Science and Technology (ICCST)*. 2022, pp. 1013–1018. DOI: 10.1109/ICCST55948.2022.10040369.

[9] “How are Draft Picks determined for the NBA Draft?” In: *NBA Draft Rules* (2023). URL: https://www.lines.com/nba/drafts/rules.

[10] Jonathan Waserman. "What Happens at the NBA Combine?" In: (2023). URL: https://bleacherreport.com/articles/1634735-what-happens-at-the-nba-combine.

[11] Drew Wolin. "Formalizing the Definition of "the Eye Test" ". In: (2023). URL: https://medium.com/@dwolin/formalizing-the-definition-of-the-eye-test-565d5fd9ae8e.

[12] Sky Sports. "Stephen Curry: I was undersized and failed the eye test but mindset made me an NBA great". In: (2024). URL: https://www.skysports.com/nba/news/36226/13075786/stephen-curry-i-was-undersized-and-failed-the-eye-test-but-mindset-made-me-an-nba-great.

[13] Justin Hussong. "15 Biggest NBA Draft Busts of All Time". In: *Bleachers/ Report* (2013). URL: https://bleacherreport.com/articles/1657535-15-biggest-nba-draft-busts-of-all-time.

[14] Willibrordus Bayu Pramudya. "College Basketball 2009-2021 + NBA Advanced Stats". https://www.kaggle.com/code/bayunova/college-basketball-2009-2021-nba-advanced-stats/input.

[15] Marilia Prata. "PCA on NBA Rookies Performance". https://www.kaggle.com/code/mpwolke/pca-on-nba-rookies-performance/input?select=NBA+Rookies+by+Year.csv.

[16] NBA. "Year-by-year NBA All-Rookie Teams". https://www.nba.com/news/history-all-rookie-teams.

[17] In: *Welcome to the SHAP documentation* (). URL: https://shap.readthedocs.io/en/latest/index.html.

[18] "2023 NBA Draft Results: Picks". In: *2023 NBA Draft* (2023). URL: https://www.nba.com/news/2023-nba-draft-order.

[19] Seyedali Mirjalili. *Evolutionary Algorithms and Neural Networks*. Springer Cham, 2019. URL: https://link.springer.com/book/10.1007/978-3-319-93025-1#bibliographic-information.

[20] Joshua Adebayo. "Can Machine Learning be used to effectively forecast the performance of aspiring NBA draft picks?" In: *GitHub* (2024). URL: https://github.com/JoshuaAdebayo/Neural-Networks-in-the-NBA.