Predicting NBA Performance of NCAA Prospects

Jonathan Copeland Williams
jonwill@stanford.edu

Albert Cai
albertc4@stanford.edu

1 Introduction

With two rounds and sixty total selections, the NBA draft aims to promote parity by granting higher picks to underperforming teams. While early selections offer the promise of securing top prospects, expectation often deviates from reality. Consider the 2013 draft. Anthony Bennett, the 1st overall pick, failed to translate his college success to the NBA and was sent to the G-League. Conversely, Giannis Antetokounmpo, selected 15th, quickly rose to stardom, becoming a two-time MVP. This example highlights the unpredictable nature of the draft and emphasizes the need for reliable player evaluation. This project aims to enhance draft decision-making by utilizing statistical models to predict NCAA prospects' NBA performance.

2 Related Work

Extensive research has explored predicting how a *current* NBA player will perform in the future. A team from DataRobot utilizes a cohort of 30-40 models to predict NBA players' Game Scores from their prior stats [3]. Kevin Wheeler - a prior CS 229 student - utilized linear models to predict NBA players' points scored [4]. Similarly, Data Scientist Mario Hoxha employed linear regression to predict NBA Players' points per game and 3-point shooting % stats [5]. To our knowledge, no research into predicting *NCAA* player performance in the NBA has been conducted.

3 Dataset

3.1 Defining Over/Under-performance

Our models aim to utilize NCAA stats to predict if a player will either 1) over-perform in the NBA, or 2) under-perform in the NBA. This modeling objective naturally forced us to define what it means for a player to over/under-perform in the NBA. To label players as over/under-performers, we chose to rely upon FiveThirtyEight's Robust Algorithm (using) Player Tracking (and) On/Off Ratings (RAPTOR) summary statistic [2].

RAPTOR is a publicly available statistic that takes advantage of modern NBA data - specifically player tracking and play-by-play data that is not available in traditional box scores. Compared to other statistics, RAPTOR differentiates itself by highly weighing nuanced statistics, such as "floor spacing, defense and shot creation" that make a player valuable. FiveThirtyEight provides RAPTOR scores for every season in a player's career. By listing the top RAPTOR scores, however, a clear problem emerges.

Table 1: 3 Players with Highest RAPTOR Scores

Name	Minutes Played	RAPTOR
Naz Mitrou-Long	1	72.6
Udonis Haslem	3	47.5
Tyler Ulis	1	45.7

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

The MVP this season, Joel Embiid (who we would expect to have a high RAPTOR), only had a score of + 8.0. Some players with limited playing time accumulate extremely high, yet unsustainable, RAPTOR scores. Therefore, we cannot quantify a player's NBA success by their maximum, mean, or median RAPTOR scores. To address this issue without filtering out players with low playing time, we derived our own summary statistic, *power*.

Power To round out the data, we compute r, the average of each player's three highest single-season RAPTOR scores, and m, the average minutes played in these three seasons. We then compute a player's power, which we defined as,

$$\text{power} := \frac{r * m + 6.24m}{10,000} + a,$$

where a is a constant so that the worst player's power is equal to 0. If a player has not played three seasons, the RAPTOR and minutes averages are computed with respect to the seasons they have played. The coefficient 6.24 was selected so that a baseline (consistently good, but not elite) player - such as Derrick Williams - would have power ≈ 1.

We see that the distribution of power matches natural intuition. There are many mediocre NBA players and few truly exceptional NBA players.

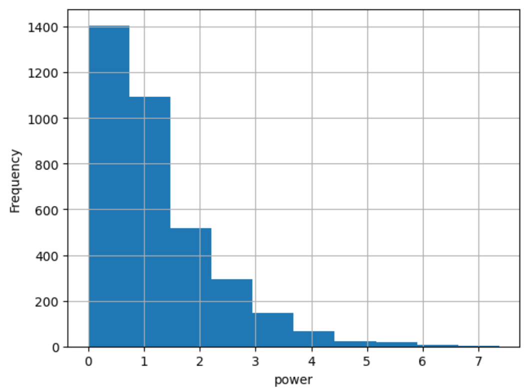


Figure 1: Power Distribution

If we rank players by power, we see that it is a much better metric of NBA performance than raw RAPTOR:

Table 2: 3 Players with Highest Power

Name	Power
Michael Jordan	7.37
LeBron James	6.69
Larry Bird	6.34

To accurately label a player as an over/under-performer, we must consider that player's draft context. Players who are drafted earlier tend to be more talented, hence their power metric is higher. Conversely, players who are drafted later tend to be less talented, hence their power metric is lower. In order to account for draft context, we bin players as follows:

- High First Round Players: Drafted in Slots [1,10]
- Mid First Round Players: Drafted in Slots [11,20]
- Late First Round Players: Drafted in Slots [21,30]
- High Second Round Players: Drafted in Slots [31,40]
- Mid-Late Second Round Players: Drafted in Slots [41,60]

For each bin, we compute the power metric distribution. We label an NCAA player as an over-performer if they fell within the top 1/4 quartile of their bin's power metric distribution. Conversely, we label an NCAA player as an under-performer if they fell within the bottom 1/4 quartile of their bin's power metric distribution.

3.2 Data Aggregation

We utilized Beautiful Soup to scrape the sports-reference college basketball website [1]. Both per-game and advanced stats were scraped from the last season of all drafted NCAA players between 2014 and 2021 (as RAPTOR data is most accurate post-2014). After setting aside 20% of data as a hold-out test set, we were left with a training set of 257 players. Each player is associated with a feature vector of 22 NCAA stats such as Points Per Game, 3Pt %, Steals, Turnovers, etc.

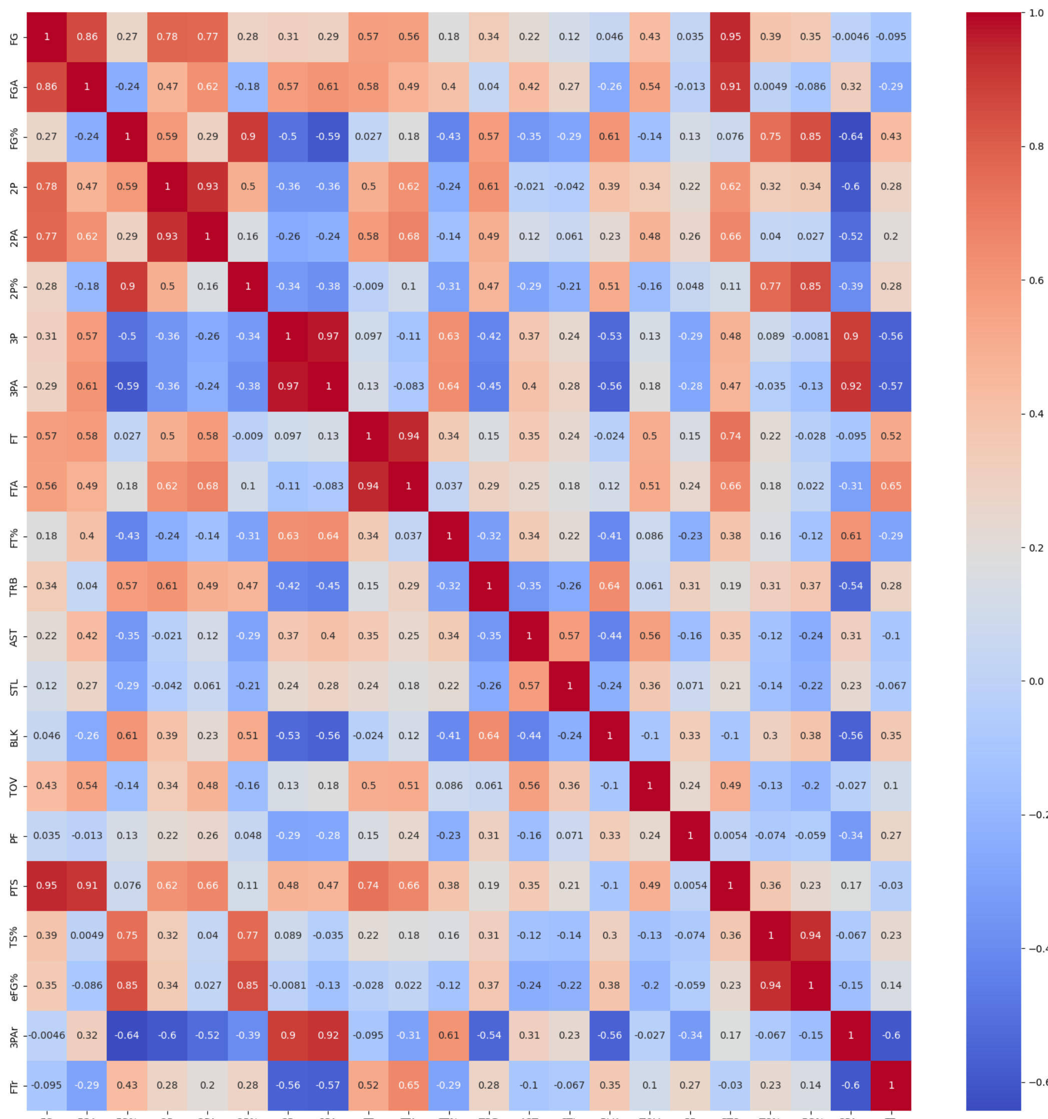


Figure 2: NCAA Feature Correlation Matrix

4 Methods

Our classification tasks are binary. Given the NCAA feature vector for a player, we wish to predict if they became an NBA over/under-performer. To this end, we experimented with 2 classical models - Logistic Regression and SVM. We also experimented with a deep-learning Multilayer Perceptron (MLP). The classical models were implemented via the Scikit-learn package. The MLP was implemented via the Keras package.

4.1 Logistic Regression

The logistic regression model assumes the hypothesis:

$$h_\theta(x) = g(\theta^\top x) = \frac{1}{1 + e^{-\theta^\top x}} \tag{1}$$

Where $g(.)$ is the sigmoid function. The probability of observing a single data sample is assumed to follow a Bernoulli distribution:

$$p(y \mid x; \theta) = (h_\theta(x))^y (1 - h_\theta(x))^{1-y} \tag{2}$$

Where x is a sample from the training dataset, and y is the associated label. The parameters θ of the model are learned via maximizing the following log-likelihood through gradient ascent:

$$\ell(\theta) = \sum_{i=1}^{n} y^{(i)} \log h(x^{(i)}) + (1 - y^{(i)}) \log(1 - h(x^{(i)})) \tag{3}$$

4.2 SVM

The SVM model assumes the hypothesis:

$$h_{w,b}(x) = g(w^\top x + b). \tag{4}$$

Where $g(z) = 1$ if $z \geq 0$, and $g(z) = -1$ otherwise. We define the geometric margin with respect to a single $(x^{(i)}, y^{(i)})$ training sample to be:

$$\gamma^{(i)} = y^{(i)} \left(\left(\frac{w}{\|w\|} \right)^\top x^{(i)} + \frac{b}{\|w\|} \right) \tag{5}$$

A large, positive geometric margin indicates that our model has yielded a correct prediction with high confidence. Let:

$$\gamma = \min_{i=1,\ldots,n} \gamma^{(i)}.$$

The parameters w, b of the model are learned via maximizing γ - our model's least confident prediction - via the constrained optimization problem:

$$\begin{aligned} \max_{\gamma,w,b} \quad & \gamma \\ \text{s.t.} \quad & y^{(i)}(w^\top x^{(i)} + b) \geq \gamma, \quad i = 1, \ldots, n \\ & \|w\| = 1. \end{aligned}$$

4.3 MLP

An MLP composed of r hidden layers assumes the hypothesis:

$$\text{MLP}(x) = \text{MM}_{W^{[r]},b^{[r]}}\left(\sigma\left(\text{MM}_{W^{[r-1]},b^{[r-1]}}\left(\sigma\left(\cdots \text{MM}_{W^{[1]},b^{[1]}}(x)\right)\right)\right)\right) \tag{6}$$

where $\sigma(\vec{z})$ is the nonlinear ReLU activation function applied element-wise to every entry in $\vec{z}$, and:

$$\text{MM}_{W,b}(z) = Wz + b \tag{7}$$

The parameters $\theta = W^{[r]}, b^{[r]} \cdots W^{[1]}, b^{[1]}$ are learned via minimizing the following loss function through gradient descent:

$$J(\theta) = -\frac{1}{n}\sum_{i=1}^{n} y^{(i)} \log(\text{MLP}(x^{(i)})) + (1 - y^{(i)}) \log(1 - \text{MLP}(x^{(i)})) \tag{8}$$

4.4 Baselines

Scikit-learn's Dummy Classifier was employed to establish a baseline performance threshold for all models. Two trivial classification approaches were utilized:

- Random: Randomly predict with $p \approx 0.25$ that each player belongs to the positive class.
- Constant: Predict that all players belong to the positive class

The proportion of labels in the negative and positive classes of both the over/under-performance datasets is roughly 0.75-0.25. In light of this class imbalance, Class Weights are incorporated into the loss function of all non-baseline models. Because we only care about model performance for the positive class, AUC-PR rather than AUC-ROC is used as the evaluation metric.

Reducing the number of features from 22 to 17 via PCA boosted performance for the classical models. For the MLPs, it was observed that dimensionality reduction harmed performance, hence all features were preserved. L2 Weight Decay is employed to regularize the classical models. Both L2 Weight Decay and 50% Dropout are employed to regularize the MLPs.

3-Fold stratified cross-validation was employed across both the classical and deep learning models.

5 Results

5.1 Over-performance Binary Classification Models

Table 3: Over-performance 3-Fold CV Results

Model	AUC-PR
Dummy Classifier (Random)	0.271
Dummy Classifier (Constant)	0.265
SVM (Linear Kernel)	0.316
Logistic Regression	0.282
MLP (50k Parameters)	**0.354**

Table 4: Over-performance Test Results

Model	AUC-PR
Dummy Classifier (Random)	0.326
Dummy Classifier (Constant)	0.259
MLP (50k Parameters)	**0.420**

5.2 Under-performance Binary Classification Models

Table 5: Under-performance 3-Fold CV Results

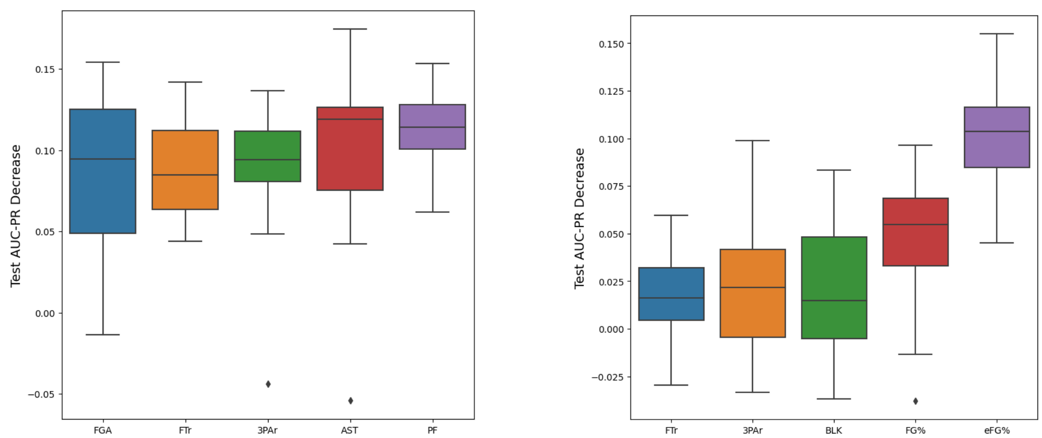
Model	AUC-PR
Dummy Classifier (Random)	0.260
Dummy Classifier (Constant)	0.253
SVM (Linear Kernel)	0.304
Logistic Regression	0.366
MLP (50k Parameters)	**0.430**

Table 6: Under-performance Test Results

Model	AUC-PR
Dummy Classifier (Random)	0.262
Dummy Classifier (Constant)	0.247
MLP (50k Parameters)	**0.406**

5.3 Feature Importance

To quantity which features most heavily influenced the MLP's inferences, a permutation feature importance test was run. In short, the values for a specific feature in the test set were shuffled, and then this permuted test set was fed to the trained MLP. The mean change in AUC-PR over 20 shuffle iterations was computed for each feature. The 5 features which elicited the largest mean decrease in AUC-PR were deemed most critical.



(a) Critical Over-performance Features (b) Critical Under-performance Features

Figure 3: Top 5 Most Critical Features for Predicting NCAA Prospect Over/Under-performance

6 Conclusions

For both predicting over/under-performance, the trained MLP outperformed data-agnostic approaches. Interestingly, it seems that it is an equally difficult task to predict if an NCAA player will over-perform or under-perform in the NBA. The test AUC-PR elicited by the under-performance MLP is within 1.4 percentage points of the test AUC-PR elicited by the over-performance MLP.

We further found it intriguing that the critical features used to predict over/under-performance fall in line with natural intuition. The over-performance critical feature set indicates that NCAA prospects who are elite Playmakers and 3-Point shooters tend to over-perform in the NBA. This player profile falls in line with recent gems such as Cade Cunningham and Jayson Tatum. Conversely, the under-performance critical feature set indicates that NCAA prospects who are elite inside finishers but poorer 3-Point shooters tend to underperform in the NBA. This player profile falls in line with recent disappointments such as Mo Bamba and Obi Toppin.

If given more time, we would like to explore including draft combine measurements (ex: height, weight, vertical jump, etc.), # of NCAA seasons played, and the NCAA conference strength as additional discriminatory features.

7 Contributions

Albert worked on deriving the power metric, and classifying players as over/under-performers. Jon worked on NCAA data aggregation and model training. Both members contributed in a satisfactory manner to the project.

References

[1] “Men’s and Women’s College Basketball Statistics and History: College Basketball at Sports.” Reference.Com, www.sports-reference.com/cbb/. Accessed 8 June 2023.

[2] “The Best NBA Players, According to Raptor.” FiveThirtyEight, 8 June 2023, projects.fivethirtyeight.com/nba-player-ratings/.

[3] Miller, Benjamin. “Using Automated Machine Learning to Predict NBA Player Performance.” DataRobot AI Platform, 7 July 2022, www.datarobot.com/blog/using-datarobot-to-predict-nba-player-performance.

[4] Wheeler, Kevin. Stanford University, "Predicting NBA Player Performance" cs229.stanford.edu/proj2012 Accessed 9 June 2023.

[5] Hoxha, Mario. “Predicting Individual Player Performance Based on Historical Trends.” Medium, 3 Jan. 2023, levelup.gitconnected.com/predicting-individual-player-performance-based-on-historical-trends-754235467c5f.