HOOPLICATOR

Peter Nygaard

peter.a.nygaard

@RetepAdam

Background

In the National Basketball Association (NBA), teams operate under a salary cap, meaning there is a finite amount of money they are allowed to spend on players. Because of this, being able to spend efficiently and derive the most "bang for your buck" on player salaries is of the utmost importance.

Where this becomes difficult is when a team has a player who fits its system really well but suddenly, due to contract expiry, becomes much pricier. Ideally, the team would be able to replicate the costlier player's production without having to break the bank to keep them by finding a 'Hooplicate.'

Objectives

With the above in mind, the goals for the Hooplicator are twofold:

1. *Create a model that enables teams to identify players who will be able to replicate the production of a given current player.*
2. *Identify player skills that, based on the model, are more easily replaceable to prioritize where teams should be willing to spend and where they can expect production from replacement-level talent.*

Approach

In order to aggregate the data necessary to construct the model, I scraped the online basketball data basketball-reference to retrieve both career college statistics and rookie-year NBA statistics, using rookie-year data to get the most accurate reflection of out-of-the-box production for college players. By starting with college statistics rather than trying to translate from other various leagues (D-League, international, etc.), I was able to tap into an immense data set on the college side — and since the NCAA remains the largest point of entry for incoming NBA players, I guaranteed myself a large enough set of NBA rookies for the output end of the model.

To model production from college career to NBA rookie year, I filtered the database to only include players who had been NBA rookies since 2000 and appeared in the college database as well (i.e. no international players). Additionally, I set a threshold of 500 minutes played among rookies in order to reduce the amount of noise from who hadn't seen the floor enough for their numbers to begin to stabilize. After starting with a threshold of 100 minutes played, I continued raising the threshold until finding comfortable middle ground between improvement of the model's accuracy and maintaining a large enough sample size to avoid overfitting at 500 MP.

I decided to fit a model for each individual skill we were looking to replicate so as to keep each individual skill wholly separate. This meant training 25 different models on the 57 features I had culled or engineered.

Modeling

For each statistic, I trained standard linear models, random forest regressors and XGBoost on the data in order to generate predictions, with the latter two yielding the best results across the board.

However, since my goal was not just to generate predictions but to gauge the probabilities of a player being able to replace another player, I used infinitesimal jackknife to derive error bars on the predictions from the random forest. Since the error bars were normally distributed across a 95% confidence interval, this allowed me to reverse engineer probabilities for each player against a given threshold for every single statistic.

Model Prediction Scores

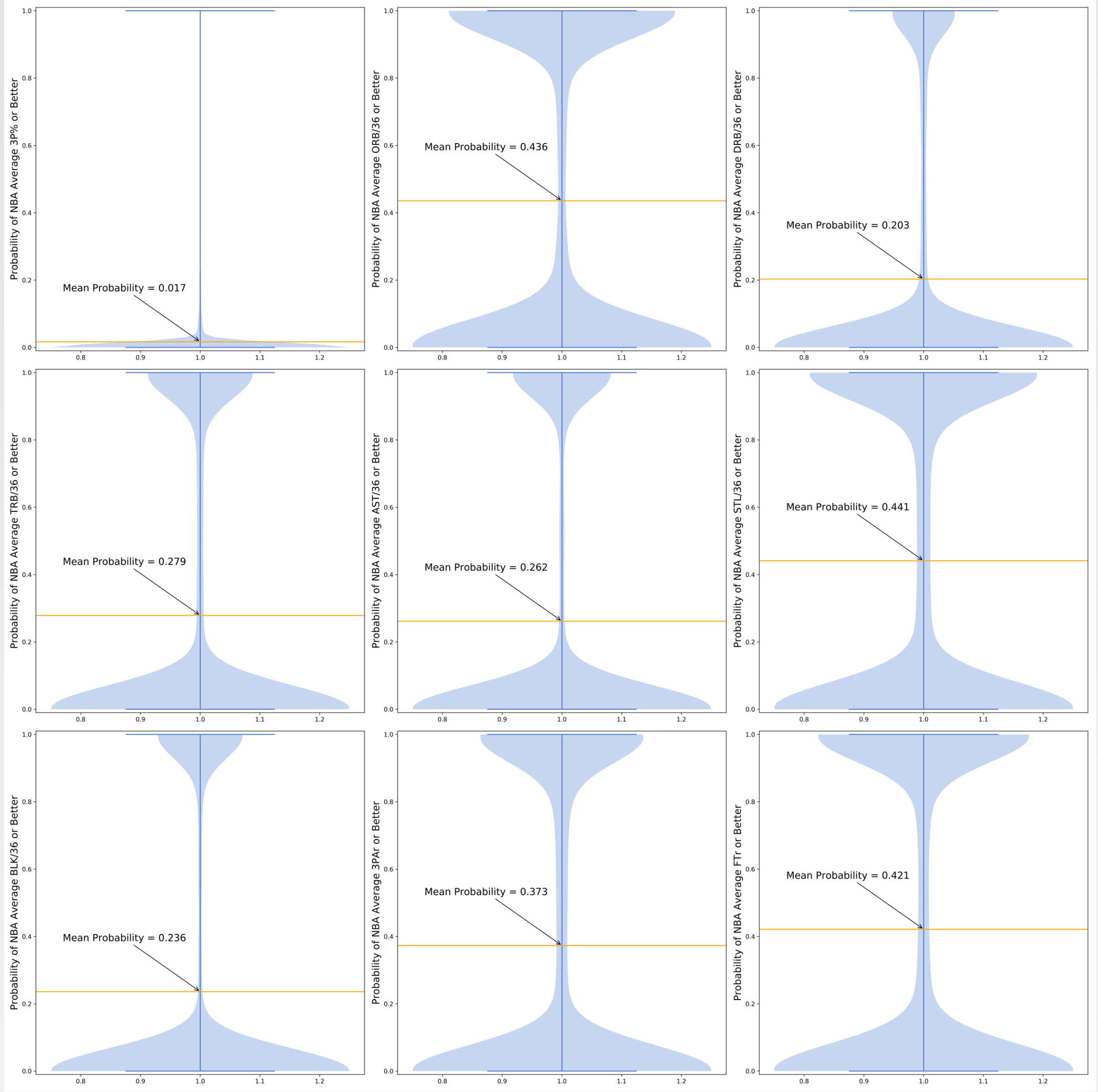
Model	FG%	2P%	3P%	FT%	eFG%	FG/36	FGA/36	2P/36	2PA/36
Linear Model	0.3122	0.0875	0.3627	0.4133	-0.0062	0.1069	0.2847	0.2792	0.2715
Random Forest	0.4000	0.2298	0.5935	0.4241	0.1365	0.2366	0.3273	0.3621	0.3468
XGBoost	0.3689	0.2353	0.5940	0.4416	0.1415	0.2518	0.4040	0.3326	0.3580

Model	3P/36	3PA/36	FT/36	FTA/36	ORB/36	DRB/36	TRB/36	AST/36
Linear Model	0.5442	0.6035	0.1504	0.2283	0.6720	0.5680	0.7114	0.6049
Random Forest	0.6776	0.6895	0.2842	0.3334	0.7141	0.6213	0.7438	0.6667
XGBoost	0.6771	0.6852	0.3283	0.3013	0.7442	0.6557	0.7431	0.6909

Model	STL/36	BLK/36	TOV/36	PF/36	PTS/36	TS%	3PAr	FTr
Linear Model	0.3652	0.6061	-0.1500	0.2946	0.0753	-0.0672	0.5393	0.4977
Random Forest	0.4796	0.7751	0.2548	0.4323	0.2260	0.1361	0.6680	0.5186
XGBoost	0.5039	0.8094	0.2866	0.4491	0.2134	0.1319	0.6762	0.5912

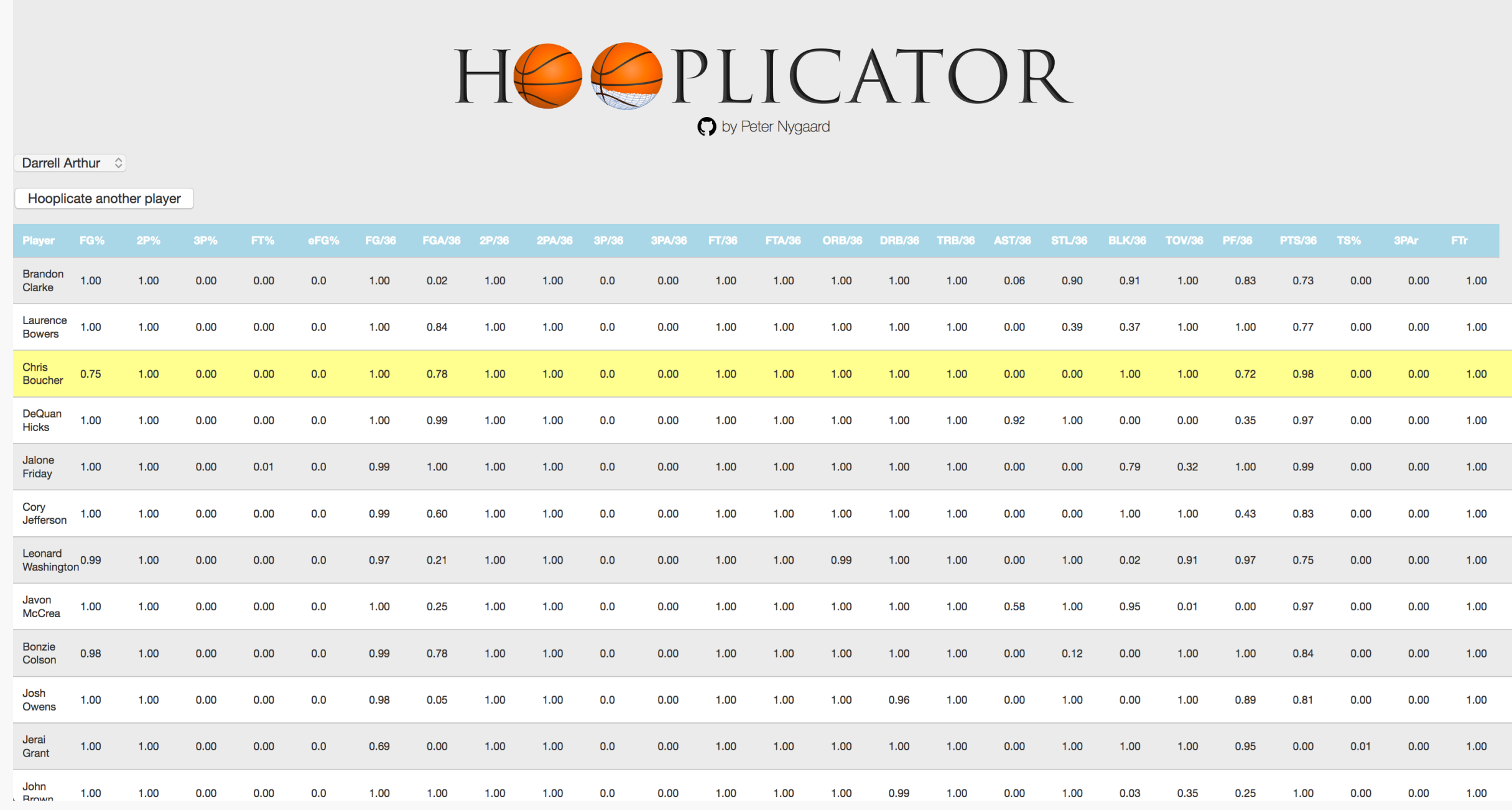
Results

Probability Distribution of Recent College Players Per Statistic Against NBA Average



Categories (L-R): 3-Point Field Goal Percentage, Offensive Rebounds Per 36 Minutes, Defensive Rebounds Per 36 Minutes; Total Rebounds Per 36 Minutes, Assists Per 36 Minutes, Steals Per 36 Minutes; Blocks Per 36 Minutes, 3-Point Field Goal Attempt Rate, Free Throw Attempt Rate

Web App



HOOPLICATOR

by Peter Nygaard

Player	FG%	2P%	3P%	FT%	eFG%	FG/36	FGA/36	2P/36	2PA/36	3P/36	3PA/36	FT/36	FTA/36	ORB/36	DRB/36	TRB/36	AST/36	STL/36	BLK/36	TOV/36	PF/36	PTS/36	TS%	3PAr	FTr
Brandon Clarke	1.00	1.00	0.00	0.00	0.0	1.00	0.02	1.00	1.00	0.0	0.00	1.00	1.00	1.00	1.00	1.00	0.06	0.90	0.91	1.00	0.83	0.73	0.00	0.00	1.00
Laurence Bowers	1.00	1.00	0.00	0.00	0.0	1.00	0.84	1.00	1.00	0.0	0.00	1.00	1.00	1.00	1.00	1.00	0.00	0.39	0.37	1.00	1.00	0.77	0.00	0.00	1.00
Chris Boucher	0.75	1.00	0.00	0.00	0.0	1.00	0.78	1.00	1.00	0.0	0.00	1.00	1.00	1.00	1.00	1.00	0.00	0.00	1.00	1.00	0.72	0.98	0.00	0.00	1.00
DeQuan Hicks	1.00	1.00	0.00	0.00	0.0	1.00	0.99	1.00	1.00	0.0	0.00	1.00	1.00	1.00	1.00	1.00	0.92	1.00	0.00	0.00	0.35	0.97	0.00	0.00	1.00
Jalone Friday	1.00	1.00	0.00	0.01	0.0	0.99	1.00	1.00	1.00	0.0	0.00	1.00	1.00	1.00	1.00	1.00	0.00	0.00	0.79	0.32	1.00	0.99	0.00	0.00	1.00
Cory Jefferson	1.00	1.00	0.00	0.00	0.0	0.99	0.60	1.00	1.00	0.0	0.00	1.00	1.00	1.00	1.00	1.00	0.00	0.00	1.00	1.00	0.43	0.83	0.00	0.00	1.00
Leonard Washington	0.99	1.00	0.00	0.00	0.0	0.97	0.21	1.00	1.00	0.0	0.00	1.00	1.00	0.99	1.00	1.00	0.00	1.00	0.02	0.91	0.97	0.75	0.00	0.00	1.00
Javon McCrea	1.00	1.00	0.00	0.00	0.0	1.00	0.25	1.00	1.00	0.0	0.00	1.00	1.00	1.00	1.00	1.00	0.58	1.00	0.95	0.01	0.00	0.97	0.00	0.00	1.00
Bonzie Colson	0.98	1.00	0.00	0.00	0.0	0.99	0.78	1.00	1.00	0.0	0.00	1.00	1.00	1.00	1.00	1.00	0.00	0.12	0.00	1.00	1.00	0.84	0.00	0.00	1.00
Josh Owens	1.00	1.00	0.00	0.00	0.0	0.98	0.05	1.00	1.00	0.0	0.00	1.00	1.00	1.00	0.96	1.00	0.00	1.00	0.00	1.00	0.89	0.81	0.00	0.00	1.00
Jerai Grant	1.00	1.00	0.00	0.00	0.0	0.69	0.00	1.00	1.00	0.0	0.00	1.00	1.00	1.00	1.00	1.00	0.00	1.00	1.00	1.00	0.95	0.00	0.01	0.00	1.00
John Brown	1.00	1.00	0.00	0.00	0.0	1.00	1.00	1.00	1.00	0.0	0.00	1.00	1.00	1.00	0.99	1.00	0.00	1.00	0.03	0.35	0.25	1.00	0.00	0.00	1.00

Conclusion

The Hooplicator model was best able to predict 3-point statistics, all three rebounding statistic rates, assist rate, block rate and shooting tendencies.

Cross-referencing these statistics against the probability distributions for each statistic, we can see that while it is easy to find players who fancy themselves 3-point shooters, it is much more difficult to find players who one can actually expect to make NBA 3-pointers at even a league average rate.

On the flip side, offensive rebounding is an area where there is a healthy distribution of players who could be expected to produce at an average level or better, same with creating steals and getting to the free throw line — though it must be stressed that this is in comparison to the league average mark. It would likely prove much more difficult to find players able to get to the free throw line with the frequency of a James Harden or Jimmy Butler.

In lieu of being able to continue to improve upon the model itself, the next step to utilize this information will be to either construct a similar type of model or use clustering to try to determine how much teams are paying for each individual skill/statistic. By assigning a cost variable to each, we can further elucidate which skills are being overvalued by teams and which may exist as market inefficiencies with a strong value-to-cost ratio.

References

[1] Wager, S., Hastie, T., & Efron, B. (2014). Confidence Intervals for Random Forests: The Jackknife and the Infinitesimal Jackknife. *Journal of Machine Learning Research.*

[2] Nichols, J. (2009). How Do NCAA Statistics Translate to the NBA? Retrieved from http://basketball-statistics.com/howdoncaastatisticstranslatetothenba.html

[3] Johnson, A. (2014). Predictions Are Hard, Especially About Three Point Shooting. *Counting The Baskets.*