DS 4400: Machine Learning and Data Mining I
Spring 2022
Team Members: Cameron Gleichauf (@CamGleichauf), Justin Chen (@jstinchen)
Video

NCAA Basketball Prospect Grader

Problem Description

Every year NBA teams spend countless resources and time trying to determine the best college and international prospects that have entered the NBA draft. Draft selections can shape the course of a franchise for years to come and make for some of the most important decisions a franchise can make. Take the 2018 draft for example, where 4 of the top 5 picks have ranged from above average starters to bonafide franchise players. Then look at the #2 overall pick in the draft, Marvin Bagley, who struggled to find his footing in Sacramento and was eventually traded to Detroit for rotation pieces Trey Lyles and Josh Jackson. The importance of drafting well is showcased here, as each of the four other teams drafting in the top 5 made the playoffs this year, while Sacramento missed the playoffs for the 16th consecutive year.

Even after putting large amounts of time, energy and effort into scouting, many franchises still fail to choose the 'best' players possible or the 'best' players for their team needs. These decisions affect the success of the franchise for years to come and should be made with the most possible context information possible. Our machine learning algorithm aims to resolve this issue by collecting a large set of relevant feature variables that contribute to the success of draft prospects in the NBA. Our goal is to predict the quality of a potential NCAA D1 prospect's NBA career on a continuous scale based on evidence from recent prior drafts and patterns in the data.

The problem we are aiming to resolve is a regression problem that predicts a continuous output variable representing the estimated NBA win shares for each NCAA prospect. We will use regularized linear regression, decision tree regression, random forest regression, adaboost ensemble regression and a multi-layer perceptron to attempt to predict each training row's value on a continuous scale.

The resources, money and time that are spent on scouting is enormous and to solely use 'the eye test' in evaluating players will almost certainly lead to suboptimal draft selections. This is why the use of machine learning as an aid for decision making is incredibly important for NBA front offices. Machine learning is by no means a perfect system either, but its use as a supplemental factor when drafting players can provide huge benefits in selecting optimal players that may be valued less if only the eye test is used. Furthermore, the application of this machine learning algorithm can be used to revolutionize the way draft prospects are valued in the NBA by allowing us to make unbiased evaluations based on player talent that are repeatable over a large period of time. The use of machine learning algorithms to quantify the quality of a player can provide benefits in the range of millions and millions of dollars to teams if used correctly and in supplementation of classic scouting techniques.

Dataset and Exploratory Data Analysis

Dataset statistics

- Around 23,000 records of collegiate players and their NBA contributions (win shares)
- Our preliminary dataset before implementing recursive feature elimination and regularization includes about 46 different features
- Another set of 73 prospects and their statistics for the 2022 NBA Draft were collected. This list was based off all the collegiate prospects on ESPN's Top 100 Draft Board
- Statistics and biographical information for the two datasets were scraped from Sports Reference using BeautifulSoup4 and wrangled with Pandas

Feature definition and some insights

- Features are split into three main categories: counting statistics, advanced statistics and metadata regarding the player's physical attributes and position
- Positional labels are encoded
- We analyzed the covariance between various feature variables and attempted to eliminate unnecessary variables in an attempt to make the model easier to train and interpret.

Exploratory data analysis

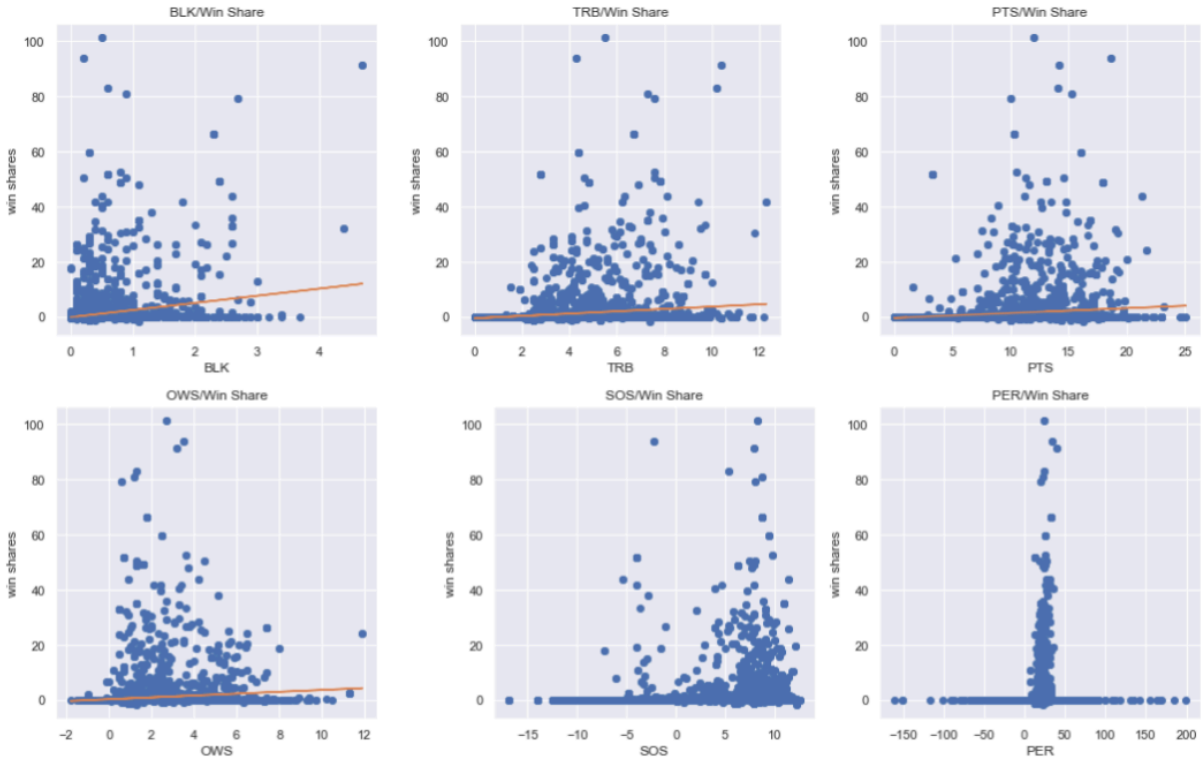
In order to apply recursive feature elimination and to determine the covariance between various feature variables, we plotted the Pearson correlation to the features dataframe. Here is the resulting graph depicting a subset of the features:

	G	GS	MP	FG	FGA	FG%	2P	2PA	2P%	3P	3PA	3P%	FT	FTA	FT%	ORB	DRB	TRB	AST	STL	BLK	TOV	PF	PTS
G	1.000000	0.807792	0.667566	0.574480	0.562754	0.276800	0.511037	0.520429	0.184017	0.384208	0.378523	0.183351	0.495241	0.501229	0.271318	0.407096	0.553585	0.529291	0.453983	0.507524	0.290726	0.515629	0.607178	0.572648
GS	0.807792	1.000000	0.812048	0.759709	0.745659	0.244987	0.691062	0.707737	0.164586	0.478285	0.475105	0.187714	0.689638	0.691591	0.291893	0.485240	0.694867	0.655319	0.608095	0.654361	0.346130	0.689371	0.656610	0.761309
MP	0.667566	0.812048	1.000000	0.903718	0.920166	0.262144	0.788534	0.829478	0.169346	0.639178	0.650726	0.264568	0.814008	0.808579	0.414684	0.518404	0.805357	0.743839	0.744202	0.809814	0.341025	0.865617	0.782229	0.913656
FG	0.574480	0.759709	0.903718	1.000000	0.973151	0.329104	0.922387	0.933537	0.233619	0.603637	0.605750	0.242722	0.869360	0.864146	0.374314	0.590488	0.828655	0.785831	0.617424	0.726275	0.412823	0.826513	0.708272	0.988893
FGA	0.562754	0.745659	0.920166	0.973151	1.000000	0.199009	0.836632	0.885274	0.129053	0.714267	0.730882	0.268115	0.860277	0.836792	0.422815	0.466969	0.762620	0.695938	0.675934	0.760343	0.296445	0.842950	0.668299	0.983564
FG%	0.276800	0.244987	0.262144	0.329104	0.199009	1.000000	0.420829	0.338880	0.797233	-0.045968	-0.093920	0.277539	0.236507	0.277850	0.003014	0.443412	0.378865	0.421581	0.053745	0.141858	0.385931	0.207097	0.391446	0.279581
2P	0.511037	0.691062	0.788534	0.922387	0.836632	0.420829	1.000000	0.981297	0.295442	0.250016	0.258506	0.077208	0.832067	0.869561	0.221174	0.753032	0.852787	0.860445	0.496218	0.626645	0.552678	0.776665	0.727270	0.876782
2PA	0.520429	0.707737	0.829478	0.933537	0.885274	0.338880	0.981297	1.000000	0.202511	0.316133	0.329880	0.098682	0.866839	0.894721	0.256418	0.698615	0.835734	0.829200	0.564052	0.675280	0.489001	0.829798	0.732944	0.902056
2P%	0.184017	0.164586	0.169346	0.233619	0.129053	0.797233	0.295442	0.202511	1.000000	-0.026960	-0.035365	-0.000495	0.144484	0.168674	0.007570	0.293397	0.266730	0.290379	0.021664	0.096057	0.264998	0.107579	0.244010	0.193591
3P	0.384208	0.478285	0.639178	0.603637	0.714267	-0.045968	0.250016	0.316133	-0.026960	1.000000	0.986220	0.441412	0.463266	0.372617	0.463925	-0.074561	0.317366	0.193859	0.526088	0.529021	-0.106997	0.471007	0.274063	0.670648
3PA	0.378523	0.475105	0.650726	0.605750	0.730882	-0.093920	0.258506	0.329880	-0.035365	0.986220	1.000000	0.393618	0.474622	0.385945	0.464871	-0.076804	0.322210	0.196369	0.544575	0.553055	-0.115682	0.493561	0.281053	0.673127
3P%	0.183351	0.187714	0.264568	0.242722	0.268115	0.277539	0.077208	0.098682	-0.000495	0.441412	0.393618	1.000000	0.166282	0.116748	0.312324	-0.068448	0.122906	0.061795	0.228386	0.211768	-0.073535	0.173507	0.098521	0.271969
FT	0.495241	0.689638	0.814008	0.869360	0.860277	0.236507	0.832067	0.866839	0.144484	0.463266	0.474622	0.166282	1.000000	0.980265	0.390615	0.517907	0.734974	0.695200	0.637700	0.696129	0.330945	0.826261	0.656782	0.908793
FTA	0.501229	0.691591	0.808579	0.864146	0.836792	0.277850	0.869561	0.894721	0.168674	0.372617	0.385945	0.116748	0.980265	1.000000	0.281400	0.606813	0.774422	0.754255	0.605522	0.685618	0.403038	0.833746	0.699847	0.888056
FT%	0.271318	0.291893	0.414684	0.374314	0.422815	0.003014	0.221174	0.256418	0.007570	0.463925	0.464871	0.312324	0.390615	0.281400	1.000000	-0.001772	0.225998	0.155575	0.335268	0.324338	-0.044479	0.321766	0.214950	0.421555
ORB	0.407096	0.485240	0.518404	0.590488	0.466969	0.443412	0.753032	0.698615	0.293397	-0.074561	-0.076804	-0.068448	0.517907	0.606813	-0.001772	1.000000	0.791037	0.904700	0.092079	0.329805	0.699153	0.450773	0.682142	0.525279
DRB	0.553585	0.694867	0.805357	0.828655	0.762620	0.378865	0.852787	0.835734	0.266730	0.317366	0.322210	0.122906	0.734974	0.774422	0.225998	0.791037	1.000000	0.975838	0.453822	0.604728	0.605840	0.726033	0.770192	0.797784
TRB	0.529291	0.655319	0.743839	0.785831	0.695938	0.421581	0.860445	0.829200	0.290379	0.193859	0.196369	0.061795	0.695200	0.754256	0.155575	0.904700	0.975838	1.000000	0.348262	0.537184	0.669761	0.664884	0.778009	0.741245
AST	0.453983	0.608095	0.744202	0.617424	0.675934	0.053745	0.496218	0.564052	0.021664	0.526088	0.544575	0.228386	0.637700	0.605522	0.335268	0.092079	0.453822	0.348262	1.000000	0.780325	-0.003298	0.789771	0.457169	0.655840
STL	0.507524	0.654361	0.809814	0.726275	0.760343	0.141858	0.626645	0.675280	0.096057	0.529021	0.553055	0.211768	0.696129	0.685618	0.324338	0.329805	0.604728	0.537184	0.780325	1.000000	0.181403	0.777708	0.607863	0.746266
BLK	0.290726	0.346130	0.341025	0.412823	0.296445	0.385931	0.552678	0.489001	0.264998	-0.106997	-0.115682	-0.073535	0.330945	0.403038	-0.044479	0.699153	0.605840	0.669761	-0.003298	0.181403	1.000000	0.280424	0.524904	0.352236
TOV	0.515629	0.689371	0.865617	0.826513	0.842950	0.207097	0.776665	0.829798	0.107579	0.471007	0.493561	0.173507	0.826261	0.833746	0.321766	0.450773	0.726033	0.664884	0.789771	0.777708	0.280424	1.000000	0.727006	0.839149
PF	0.607178	0.656610	0.782229	0.708272	0.668299	0.391446	0.727270	0.732944	0.244010	0.274063	0.281053	0.098521	0.656782	0.699847	0.214950	0.682142	0.770192	0.778009	0.457169	0.607863	0.524904	0.727006	1.000000	0.689062
PTS	0.572648	0.761309	0.913656	0.988893	0.983564	0.279581	0.876782	0.902056	0.193591	0.670648	0.673127	0.271969	0.908793	0.888056	0.421555	0.525279	0.797784	0.741245	0.655840	0.746266	0.352236	0.839149	0.689062	1.000000

- Since we have 46 feature variables we are starting with, the graph is quite cluttered and hard to make sense of. To simplify the model, we applied recursive feature elimination to reduce the total number of feature variables and get the most relevant features. Not many of the variables were eliminated after applying

recursive feature selection. One possible reason for this is that, based on our feature correlation map, most of the feature variables were positively correlated to the target variable and thus feature elimination was not highly suitable for this particular model.

- Another technique we utilized was one-hot encoding. This allowed us to turn categorical features such as the player's position into numerical features that could be used as part of our model.
- We also ranked all of the features by individual correlation to win shares to get a rudimentary idea of each of them's importance
    - Blocks (0.23818), rebounds (0.191122), points (0.182362), strength of schedule (0.142153), offensive win shares (0.117306), and player efficiency rating (0.118170) were a couple features that stood out



    - With these key, high-correlation statistics, we also looked at how different positions averaged in these categories. It can be seen that big-men triumph guards in blocks and player efficiency, while margins for rebounds, offensive win-share, and points aren't too major.

position	TRB	BLK	OWS	PTS	PER
Guard	2.152826	0.133528	1.08446	6.118033	10.809212
Forward/Center	3.424525	0.498334	1.024874	5.393287	12.979949

- All the features were positively correlated with win shares except for 3-point attempt rate (3PAr) and turnover percentage (TOV%).
    - 3PAr's negative correlation can be attributed to the idea of taking efficient shots and not settling for threes. Being able to score around the basket and from distance also shows a player's well-roundedness, so if a player is only taking 3-pointers that is not ideal.
    - TOV%'s negative correlation can be explained easily – you don't want a player giving the ball to the other team

Approach and Methodology

Machine learning models/Metrics:

We trained and tested 5 separate models after randomly splitting the training and testing set with sklearn.model_selection import train_test_split function. These were the metrics we obtained for each model, in decreasing order of quality on the testing set:

1. Random Forest Regression (Best Model)
    a. Training r2 score: 0.930
    b. Testing r2 score: 0.601
    c. Testing MSE: 7.659
2. Decision Tree Regression
    a. Training r2 score: 0.558
    b. Testing r2 score: 0.449
    c. Testing MSE: 10.574
3. AdaBoost Ensemble Regression
    a. Training r2 score: 0.467
    b. Testing r2 score: 0.320
    c. Testing MSE: 13.057
4. Multi-Layer Perceptron Regression
    a. Training r2 score: 0.467
    b. Testing r2 score: 0.307
    c. Testing MSE: 13.309
5. Ridge Regression (Worst Model)
    a. Training r2 score: 0.158
    b. Testing r2 score: 0.145
    c. Testing MSE: 16.41

Discussion and Result Interpretation

We found random forest regression to be the most effective model on the testing set, with an r^2 value of just over 0.6. Initially, the r^2 value for random forest regression was closer to 0.5 but after applying cross-validation to find the optimal max depth hyperparameter (which was 15), we were able to increase the testing r2 value by almost 0.1. Random forest regression models are composed of multiple decision trees and average the results of these trees together. In this way, we increase the previously small sample size of the decision trees that tend to catch much of the noise from the training data and the model is able to generalize much better. While decision trees tend to overfit to the training data, random forest regression is able to combat this by taking the average of a larger number of decision trees.

The second most effective model was the decision tree regression model. Initially, the model overfitted the data by a significant margin, with the training data having an r^2 value of 1 and the testing data having an r^2 value of 0.13. However, after applying cross validation with the hyperparameter of maximum tree depth, we found the optimal value to be 6. Setting 6 as the max depth value, we achieved a significantly higher testing r^2 value of 0.449 but also a significantly lower training r^2 value of 0.558. This is acceptable because the goal is to have the highest testing r2 score, we do not care about optimizing the training r2 score.

The next best performing model was AdaBoost Ensemble Regression with a training r2 score of 0.467 and a testing r2 score of 0.320. We applied cross validation on this model as well to find the optimal number of estimators to be 10.

The second worst performing model was the Multi-Layer Perceptron Regression model, with a testing r2 score of 0.307 and training r2 score of 0.467.

The worst performing model by a wide margin was the Ridge Regression model. It had a training r2 score of 0.158 and a testing r2 score of 0.145. One possible reason why it performed so poorly is because it had an extremely high bias, assuming that the model was linear. This is likely an incorrect assumption given the complex nature of the problem and thus resulted in low r^2 scores.

Model's Draft Board (Top 20 Prospects)

Model Rank	name	position	school	ESPN ranking	NBA Win Share Prediction
1	Walker Kessler	Forward	UNC/Auburn	19	31.38
2	Mark Williams	Center	Duke	15	29.12
3	Chet Holmgren	Center	Gonzaga	1	14.43
4	Trevion Williams	Forward	Purdue	36	12.31
5	Jabari Smith	Forward	Auburn	3	12.31
6	Jalen Duren	Center	Memphis	6	11.97
7	Tari Eason	Forward	Cincinnati/LSU	13	10.82
8	Bryce McGowens	Guard	Nebraska	22	10.37
9	Malaki Branham	Guard	OSU	15	8.87
10	Kennedy Chandler	Guard	Tennessee	16	8.13
11	Scotty Pippen Jr	Guard	Vanderbilt	69	7.83
12	Blake Wesley	Guard	Notre Dame	17	7.66
13	Paolo Banchero	Forward	Duke	2	7.65
14	TyTy Washington	Guard	Kentucky	12	6.68
15	Ochai Agbaji	Guard	Kansas	10	6.44
16	Keegan Murray	Forward	Iowa	5	5.99
17	AJ Griffin	Forward	Duke	7	5.93
18	EJ Liddell	Forward	OSU	18	5.93
19	Justin Lewis	Forward	Marquette	29	5.87
20	Jabari Walker	Forward	Colorado	45	5.31

It is clear that the model favored forwards and centers, looking at both the pre-implementation data exploration and the results. Blocks and rebounds were highly correlated with win-shares and bigger players collect those statistics a lot easier than guards.

For the most part, all the prospects who entered our top-20 list were all ranked in the same range as ESPN. It was interesting to see Walker Kessler and Mark Williams, both mid-first round picks, ranked so highly. Another immediate observation was how the top 7 prospects were all big men – that's a testament to their ability to grab rebounds, block shots, and make close-range shots at a high percentage (and therefore efficiency).

There were two surprising outliers in our top-20 predictions: Trevion Williams and Scotty Pippen Jr. Both benefited from above average strength of schedules (with Williams' coming in at a whopping 10). Pippen excelled in efficiency, had the second most steals, and the most points in our dataset – all features that the model weighed heavily. Williams was a good rebounder, ranked 11th in offensive win shares, and 8th in player efficiency rating.

That follows the trend of the NBA, valuing their forwards being versatile – scoring both inside and out – as "stretch bigs." A lot of veteran forwards have also improved their three-point shooting as the league and analytics shifted to put a bigger emphasis on the skill. Some players that fit this description include 2021-22 Season's MVP finalists – Giannis Antetokounmpo, Joel Embiid, and defending MVP Nikola Jokic – with rising stars and prospects like Evan Mobley, Jaren Jackson Jr, and DeAndre Ayton.

Since almost all of the features were positively correlated, the model rewards players who collect higher statistics playing harder competition (strength of schedule). This is reflected in all of the top prospects coming out of traditional basketball powerhouses (Duke, Kentucky, Kansas, Gonzaga, etc) and being high-usage, star players on their teams during their collegiate career.

References

- Using Machine Learning to Predict Careers of 2019 NBA Draft Picks - An article detailing a similar project that attempts to classify possible NCAA prospects into one of several categories: MVP winners, All-NBA First Team players, All-NBA Second or Third Team players, and All-Stars,good starters, good role players, role players, fringe bench players, and players who flamed out of the league. This was one major difference between our model and the model outlined by this article, as our model used regression to predict total win shares in the NBA while their model attempted to classify players into one of a few groups. The article reviews the different types of models they attempted to use, such as decision trees or random forests. They ran into similar issues as us with decision trees, as the models tended to perform very well on the training sets but not as well on the testing sets, something known as overfitting. These issues were addressed by the random forest models because they are composed of multiple decision trees and average the results of these trees together. In this way, we reduce the small sample size of the decision trees that tend to catch much of the noise from the training data and the model is able to generalize much better.
- Advanced NBA Stats for Dummies: How to Understand the New Hoops Math - An article that gives an overview and summary of many advanced NBA metrics that we will use as feature variables in our model training. The article discusses the tendency of advanced metrics to overvalue big men as they tend to have higher shooting percentages, blocks and rebounds as compared to guards. This was a relevant reference for us because we were attempting to find the 'best' target variable to assign to NBA players to quantitatively measure their success in the league.
- NBA Win Shares - A deep dive into the statistic known as win shares. A system developed by mathematician Bill James where three win shares represent one win. Win shares are calculated via aggregations performed on various counting statistics as outlined in the article above. This was a relevant reference for us because we were attempting to decide on which advanced metric to use as the target variable to most effectively measure an NBA player's success in the league.
- Sloan Sports Analytics Conference Discussion Regarding Use of Advanced Statistics and Metrics in NBA for Drafting - A video of a panel from the Sloan Sports Analytics Conference discussing the use of advanced metrics and analytics as guides for drafting players in the NBA draft. It discusses the current use of analytics at the time (2014) and its possible uses in the future as it pertains to NBA front offices making decisions.

Conclusion

We set out to create a Machine Learning model that could predict the best players in a given NBA draft class coming from the NCAA. We tried various models throughout the process and found random forest regression to be the most effective with an overall r^2 value of just over 0.6. While this may not seem incredibly high in comparison to some other types of problems that are commonly addressed by machine learning algorithms, drafting in the NBA is an inherently uncertain enigma. Many things cannot be entirely quantified by statistics, such as a player's mental makeup or the circumstances around them while playing in college. Additionally, teams must be able to develop and play their players effectively in order to turn them into quality NBA players. Some teams have better track records of doing so than others, and this may in turn affect what the players ultimately turn into. Furthermore, basketball is a team sport and players can contribute meaningfully, like setting screens, being a good defender, and creating scoring opportunities for teammates, without it showing up in the statistics.

There are a few ideas we came up with to further improve the quality of the model from its current state. The idea that comes to mind first is to emphasize the importance of different statistics based on a player's position. For example, rebounding and blocks should be more important for a center than for a guard. Vice versa, three point shooting and assist rate should be valued more highly for point guards than centers. Another idea that comes to mind to improve the algorithm is to include a player's age as part of the algorithm. Older players tend to have a lower ceiling than younger players in college when it comes to NBA potential and this is something we failed to include in our model. Another feature variable that could be of use is to include a player's ranking coming out of high school, as this is yet another indicator of their overall talent level. Some features that would be nice to have but are difficult to quantify are mental makeup and defensive prowess, although metrics for defensive prowess are improving as of late. Another thing to consider is potentially using a different target variable instead of career Win Shares. One downside to career Win Shares is that they tend to favor big men over guards. Additionally, for recent players such as Anthony Edwards or Ja Morant who have been excellent but for a short period of time, their career win shares will be low and in turn not necessarily indicative of the quality of player they truly are. One remedy for this would be to use WS/40, although this has its downsides as well.

Overall, the model performs fairly well on testing data and projects two of ESPN's top three players in the 2022 NBA draft (Chet Holmgren and Jabari Smith) to be in the top five players of the draft, with the third player (Paolo Banchero) finishing 13th. The model does tend to favor big men, with projected mid-first round picks Walker Kessler and Mark Williams projected as the top two players in the draft. Two prospects that our model indicates as underrated are Trevion Williams of Purdue and Scotty Pippen Jr. of Vanderbilt.