Predicting NBA Impact in Rookie Year from College Data

Daniel Varivoda

Contents

I Introduction

1.1 Project Description:

Every NBA general manager strives to draft the most effective player they can, trying to select the best possible player available. In the past, this used to be done through the use of the “eye test” and a legion of scouts, and while many sports industries are slow to change, the modern era of sports analytics has brought about many new techniques for discovering hidden gems. The goal of this project is to attempt to classify and predict players that can provide immediate value to a team in their rookie NBA season utilizing their college statistics. This is a daunting task, as many teams who have whole analytics departments dedicated to finding the best possible prospects still often manage to drafts busts. As such, being able to predict an impactful rookie with any relatively decent accuracy would be an interesting find. The data sets utilized for this project were a dataset of college basketball individual player statistics from 2008-2021 acquired from “https://barttorvik.com/”, an advanced metrics dataset and a player dictionary scraped from “www.basketball-reference.com” using the nbastatR package found on “https://github.com/abresler/nbastatR”, as well as a dataset of NBA player statistics scraped from the NBA stats website using python code found on “https://github.com/cjporteo/ml-NBA-asg-predictor”.

1.2 Data Dictionary:

College Statistics from trank.csv - 1,048,575 Rows x 65 Columns

Relevant variables:

"Player_name" - Player’s Name

"GP" - Games Played for Current Season
"Min_per" - % of Total Minutes Played
"ORtg" - Offensive Rating
"usg" - Usage Rate
"TS_per" - True Shooting Percentage
"ORB_per" - Percent of Possible Offensive Rebounds Grabbed
"DRB_per" - Percent of Possible Defensive Rebounds Grabbed
"AST_per" - Estimate of the percentage of teammate field goals a player assisted
"TO_per" - Estimate of turnovers per 100 plays
"FTM" - Free Throws Made
"FTA" - Free Throws Attempted
"FT_per" - Free Throw Make Percent
"twoPM" - Two Point Shots Made
"twoPA" - Two Point Shots Attempted
"twoP_per" - Two Point Make Percent
"TPM" - Three Point Shots Made
"TPA" - Three Point Shots Attempted
"TP_per" - Three Point Make Percent
"blk_per" - Estimate of the percent of opponent 2-point field goal attempts blocked
"stl_per" - Estimate of the percent of opponent possessions that end with a steal
"ftr" - Free Throws Attempted/ Field Goals Attempted
"yr" - Grade of Student
"ht" - Height (in.)
"porpag" - Points Over Replacement Per Adjusted Game
"adjoe" - Adjusted Offensice Efficiency
"pfr" - Personal Foul Rate
"year" - Season
"Ast_tov" - Assist to Turnover Ratio
"drtg" - Defensive Rating
"adrtg" - Adjusted Defensive Rating
"dporpag" - Defensive Points Over Replacement Per Adjusted Game
"stops" - # of Defensive Posessions that result in 0 points
"bpm" - Box Plus/Minus
"obpm" - Offensive Box Plus/Minus
"dbpm" - Defensive Box Plus/Minus
"gbpm" - Adjusted Box Plus/Minus
"mp" - Minutes Player per Game
"ogbpm" - Adjusted Offensive Box Plus/Minus
"dgbpm" - Adjusted Defensive Box Plus/Minus
"oreb" - Offensive Rebounds per Game

"dreb" - Defensive Rebounds per Game
"treb" - Total Rebounds per Game
"ast" - Assists per Game
"stl" - Steals per Game
"blk" - Blockers per Game
"pts" - Points per Game

NBA Statistics from basketball-refernce.com - 8478 Rows x 42 Columns

Relevant variables:

“slugSeason” - Current NBA Season (e.g. 2020-2021)

“namePlayer” - Player Name

“groupPosition” - Either C, Center; G, Guard; or F, Forward

“slugPlayerBREF” - Unique BBref Player Identifier

“ratioBPM” - Overall Box Plus/Minus

“ratioVORP” - Overall Value Over Replacement

NBA Statistics from ASG_19.csv & ASG_to_predict.csv - 8792 Rows x 25 Columns

Relevant variables:

“Year” - Current NBA Season (e.g 2020)

“PLAYER” - Player Name

“PIE” - Player Impact Estimate

II Data Preprocessing & Feature Engineering

2.1 Creating the Response Variable

One issue when attempting to estimate the impact of new NBA players that arose with these datasets was that college basketball data I was able to locate only dated as far back as 2008. While the college dataset has over a million rows due to the huge number of college basketball players, the number of these players that actually make it to the NBA level is incredibly minuscule, with only 60 players chosen each draft. Moreover, since 2020 was such a unique year with a shortened season, only statistics through 2019 were considered.

Ideally, one would want to look at a player's multi-year averaged impact estimate in order to get a grasp of their effectiveness, as some players may have one good year and then regress while others make a huge leap relatively early in their career. However, due to the limited amount of data available only the impact for player's NBA rookie season was considered.

The first step was to get a binary response variable for our classifier to train on and later attempt to predict. For this purpose Player Impact Estimate (PIE)

$$\left(\frac{PTS + FGM + FTM - FGA - FTA + DREB + (.5 * OREB) + AST + STL + (.5 * BLK) - PF - TO}{GmPTS+GmFGM+GmFTM-GmFGA-GmFTA+GmDREB+ (.5 * GmOREB)+GmAST+GmSTL+(.5 * GmBLK) - GmPF - GmTO)}\right)$$

was chosen as one of our measures of a players impact as it attempts to take defence into account and utilizes the raw score for most metrics, as weights for these types of statistics tend to be arbitrary. While advanced statistics are improving year by year, it is a well known fact any one alone statistic has inadequacies and fails to tell the whole story. Relying on anyone advanced metric by itself can introduce significant bias in a

model and result in some odd classifications. To this end Box Plus/Minus (BPM) and Value Over Replacement (VORP) were utlized in conjunction with PIE to create an averaged impact metric.

Since the scales of the variables were so different I had to normalize them to bring them into a (0,1) range before I could combine and average them. I chose to scale the variables in such a manner as opposed to standardizing them since it would keep the distributions intact. These scores were also normalized within their respective season groups to adjust for differences in season. Then our dataframe of basketball reference scores was combined with our nba.com stats dataframe by player and year (more details on processing of the dataframes will be provided in the next section). Once these data frames were combining, I could create my custom impact metric by averaging the normalized score of each of the advanced stats together for each player.

To create our binary classification variable, for each year, I found the median impact score and assigned a 1 to players that were above that score and 0 to players were under the median impact level under a new variable name “strong_impact”.

2.2 Preparing And Combining the Dataframes

Our final dataframe used for analysis had only 444 unique rows containing college statistics for rookie NBA players, which is a far cry from the over one million rows contained in the college statistics data frame. After using the ‘nbastatR’ package to query basketball reference for advanced player statistics and reading in our other stats data frames from a csv feel, the first step was to clean our dataframes so they can be successfully combined.

The basketball reference data frame (we will refer to this as bbref df) utlized a character vector to denote the season “2007-08”, while the nba.com stats dataframe (nba df) utlized an integer corresponding to the first four digits in the character vector. To rectify this we used the strsplit function to split the string along the “-” character and then transformed the first element of each split into an integer, making that our new season column. Also in the bbref df there would be duplicate rows for player and year if that players was traded throughout the year (one for each team and one for the total) so we grouped the dataframe by player identifier and year then removed all rows that were not teh total for years where a player had duplicates. Once that was done we could remove all extraneous columns leaving just the year, player name, unique player id, position grouping, VORP, and BPM. The VORP and BPM were then normalized following the procedure above after removing players with extremely low sample size (< 25% of games played in a season) to a skewing of data.

When looking at the distribution for PIE in our bbref df we found some extreme outliers as showing in the quantiles below.

```
> quantile(players_total$PIE)
    0%    25%    50%    75%   100%
-400.0    5.9    8.5   10.9  300.0
```

This was due to a very small amount of games played by these players leading to some odd sampling, and as such we had to limit both dataframes to just those players that had played more than 25% of the season. After rechecking the distribution of the overall

PIE it was found that this change eliminated the extreme outliers as can be seen below.

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-16.700   6.500   8.800   8.769  11.000  28.300
```

Small sample size had a similar but less extreme effect on VORP and BPM so a similar restriction was put in place. After obtaining a normalized metric for each dataset, the bbref df and nba df were joined by player name and season year to obtain a single data frame that contained all 3 key stats. To this data frame our binary response variable was added as detailed above. After this using a player dictionary containing unique identifiers and the rookie year of each basketball player that was obtaining the nbastatR package was utilized to filter out all non-rookie year nba statlines from our combined data set. Our college statistics data set was grouped by player name and sliced by the max year for each player to get only the college statistics at draft year to account for the fact that college players get drafted after different length of stays at their colleges. Then all columns except player year, name, player id, player position, and the binary impact variable were removed from the dataset and combined with our college stats data set to create our final data set for analysis. In order to prepare our categorical player position variable for analysis it was transformed into a matrix of three columns (C, F, G)

2.3 Cleaning the Dataframe

After the final dataframe was assembled it was necessary to remove extraneous columns in order to prepare it for future use in our machine learning algorithm.

First we begin by counting the number of NA values that appear in each column of our data frame. As shown below, it can be seen the vast majority of NA values are present

in columns 35-43, which correspond to very specific college statistics such as dunks, shots at the rim, and midradrange shots made/attempted.

```
  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   3  76  76
 37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61
 76  76  76  77  76  76 112   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
```

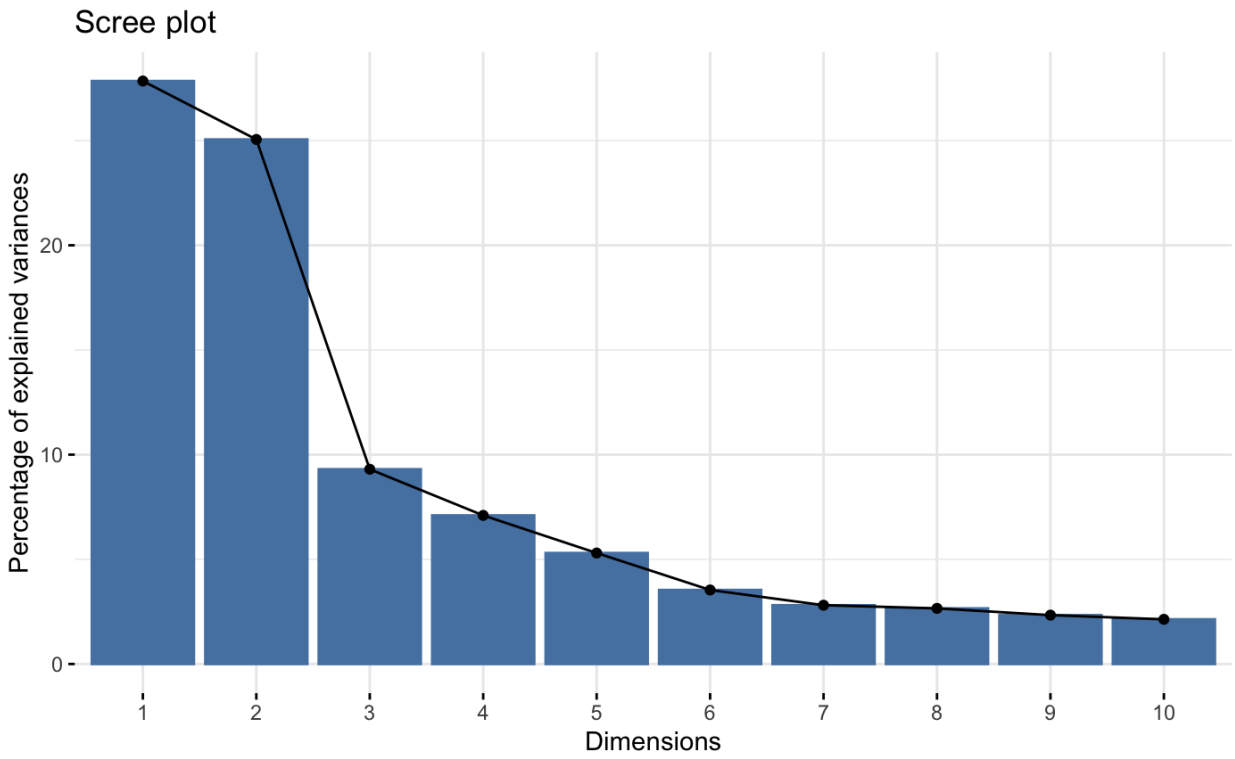
Since there are so many missing values within those columns they were removed. After utilizing the remaining data for exploratory data analysis (described in the following section), more variables were removed to prepare the dataframe for usage in the machine learning algorithm. Variables deemed not relevant to the classification, such as the players name, id, year drafted, player position, and college grade were removed. While college grade (e.g. freshman, sophomore, etc.) may have predictive power, as those nba players drafted as freshman may be more likely to be impactful rookies, however they were likely drafted as freshmen due to team believing them to be strong nba ready players rather than vise versa. Also after using a binary matrix transformation for the remaining relevant categorical variable, player position, to prepare it for use in the algorithm, the factor column was also removed. Variables that were a basic combination of other stats in the dataframe were also removed due to their extreme correlation with each other. For example, total rebounds is just a sum of offensive rebounds and defensive rebounds, or three pointers made is a multiple of three point percentage and three pointers attempted. The last category of variables removed were stats that had an adjusted version present, such as box plus minus in favor of adjusted box plus minus. The adjusted version of these statistics is utlized as takes the difficulty of opponents into account, which helps remove the inflation effect that occurs when an average player plays against weak teams or when a good player plays against mainly

strong opponents. This finally leaves with a dataframe of 438 unique nba rookies ready for analysis

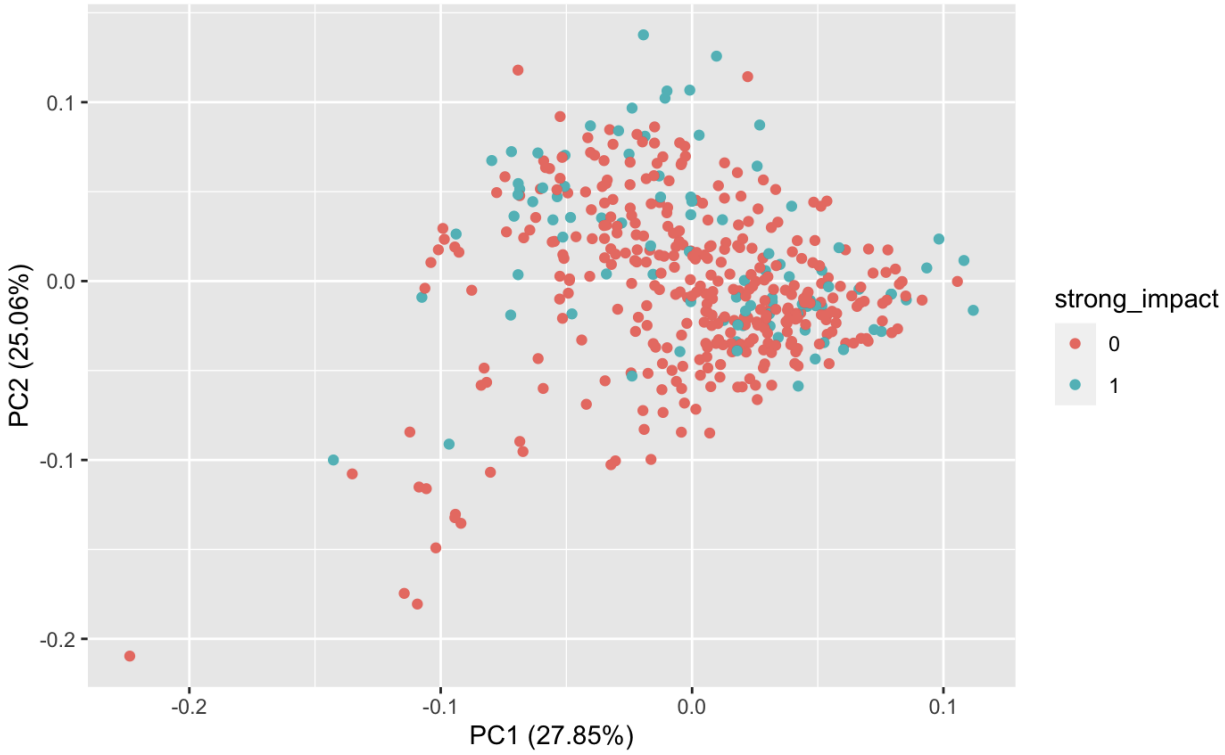
III Exploratory Data Analysis

3.1 Principle Components Analysis

Due tot he high number of possible predictors, our first step in exploring our dataset of nba rookies is to use a principle components analysis to decrease the dimensionality of our data and see if a distinct difference between the groups can be seen that way. This was done using the factoextra and caret libraries for R. After transforming our data, we can see from the chart below approximately half of the variation in our data can be explained using just the first two PCA dimensions.



Then, utlizing these two PCA dimensions the data was plotted and broken up by impact level as show below.



As expected, category for each player type will be difficult to distinguish as the overlap between the two types in the data is very high, which is why this a problem that has stumped so many sports analysts. It seems that the weak impact players may tend to appear more toward the bottom left corner of the chart while the high impact players tend to appear higher up on the y axis, which does give some indiciation of a distinction between the two groups, but for the most part the data is clumped together closely.

3.2 Possible Strong Predictors

After completing our PCA Analysis, our next step was to create a correlation matrix for all of our predictors.

The first row of this correlation matrix is given below and was utilized to identify predictors that most strongly correlated with the response variable.

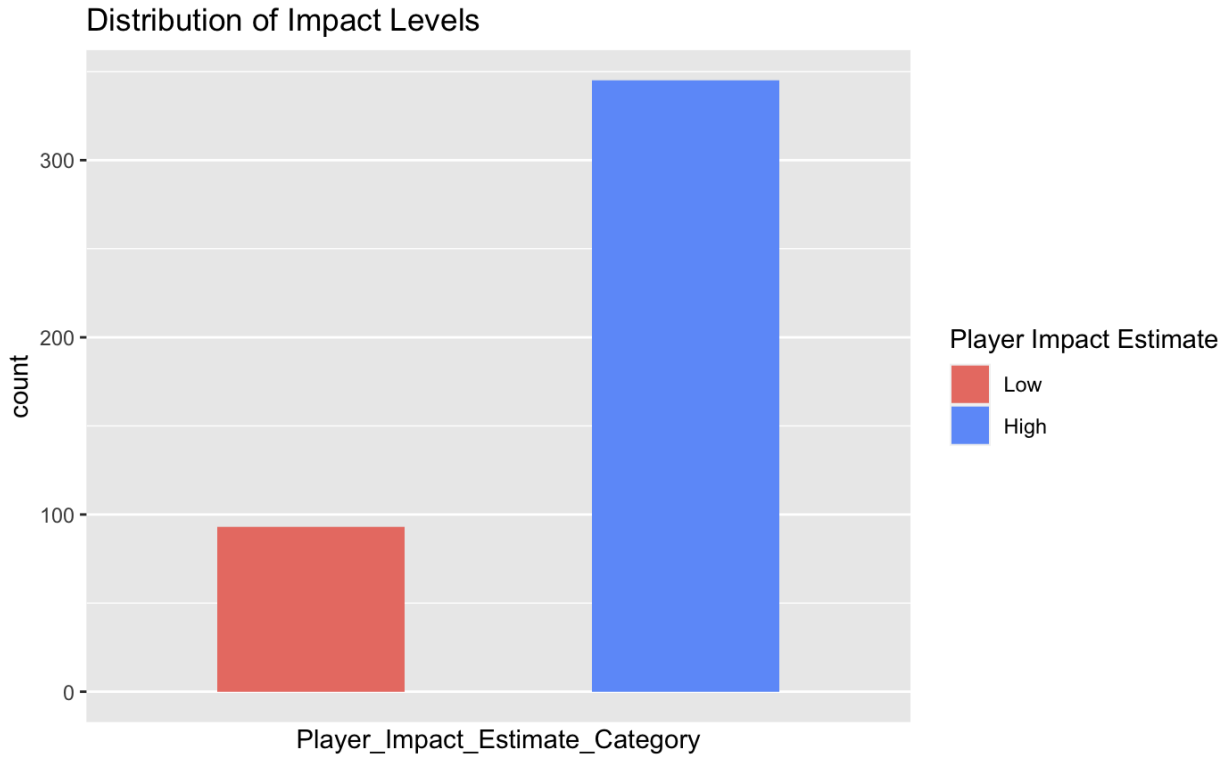
strong_impact	college_GP	college_Min_per	college_ORtg
1.00	-0.02	0.00	0.18
college_usg	college_eFG	college_TS_per	college_ORB_per
0.03	0.18	0.18	0.11
college_DRB_per	college_AST_per	college_TO_per	college_FTM
0.13	0.08	-0.01	0.07
college_FTA	college_FT_per	college_twoPM	college_twoPA
0.07	0.00	0.09	0.04
college_twoP_per	college_TPM	college_TPA	college_TP_per
0.19	-0.09	-0.10	-0.07
college_blk_per	college_stl_per	college_ftr	college_ht
0.18	0.06	0.11	0.07
college_porpag	college_adjoe	college_pfr	college_year
0.14	0.16	0.01	-0.11
college_ast_tov	college_drtg	college_adrtg	college_dporpag
0.04	-0.13	-0.12	0.09
college_stops	college_bpm	college_obpm	college_dbpm
0.08	0.22	0.14	0.18
college_gbpm	college_mp	college_ogbpm	college_dgbpm
0.21	0.02	0.15	0.18
college_oreb	college_dreb	college_treb	college_ast
0.09	0.10	0.10	0.08
college_stl	college_blk	college_pts	players_impact.groupPositionC
0.07	0.16	0.05	0.13
players_impact.groupPositionF	players_impact.groupPositionG		
-0.09	-0.01		

From this table it can be seen that Offensive Rating, True Shooting, Block Percentage, Offensive Efficiency, and Box Plus/Minus statistics are among some of the strongest correlated predictors. It is interesting that the center posistion seem to be relatively well correlated with a strong rookie impact, however the other positions are not. This may be because centers rely on raw athletic ability more than other positions, making it easier to distinguish strong prospects for NBA teams.

3.3 Graphical Analysis

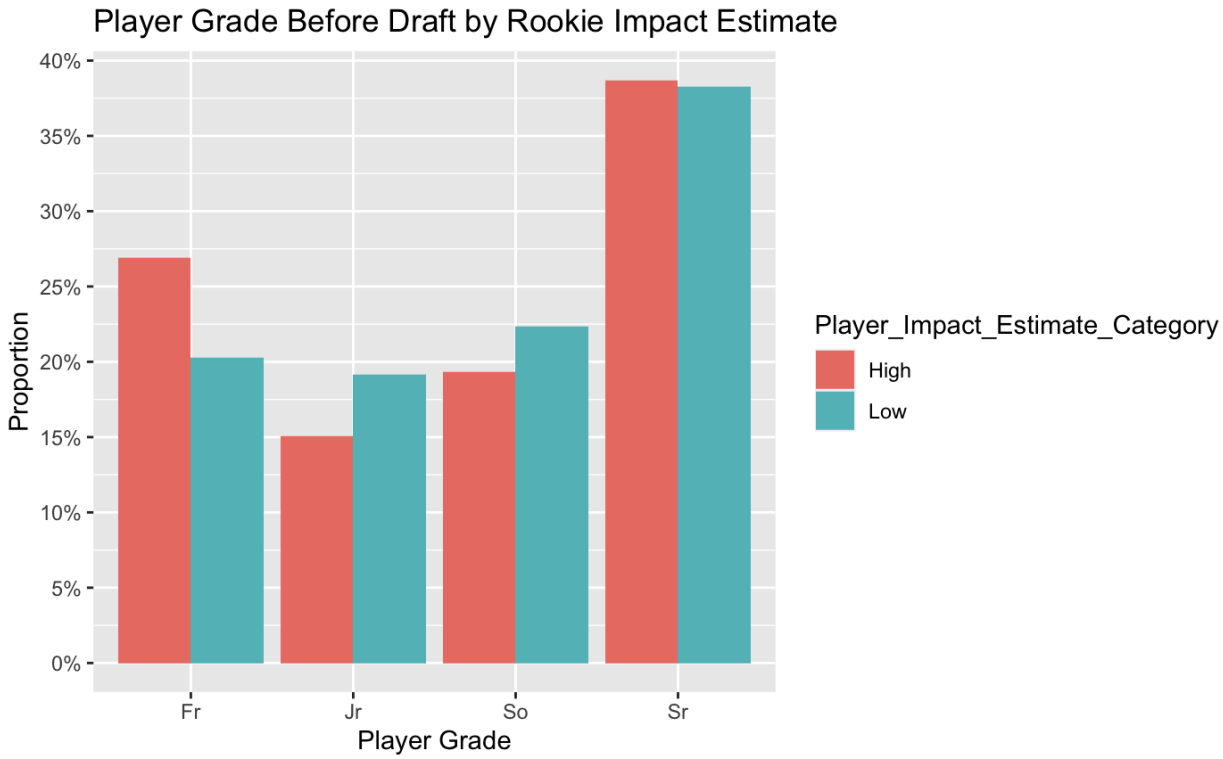
Using the correlation matrix as a guide, I delved into graphical views of the different predictors. I decided to use density histograms of our different predictors since they adjust for the size differences between the two data sets and provide a good view of the differences in distribution between the two data sets.

A basic histogram of the Player Impact categories shows us that impactful rookies are relatively rare, with only about 20% of the players present being considered impactful. This is to be expected as a vast majority of NBA rookies require a few years of development before becoming impactful players, while others end up unable to adapt the extreme skill level of the NBA.



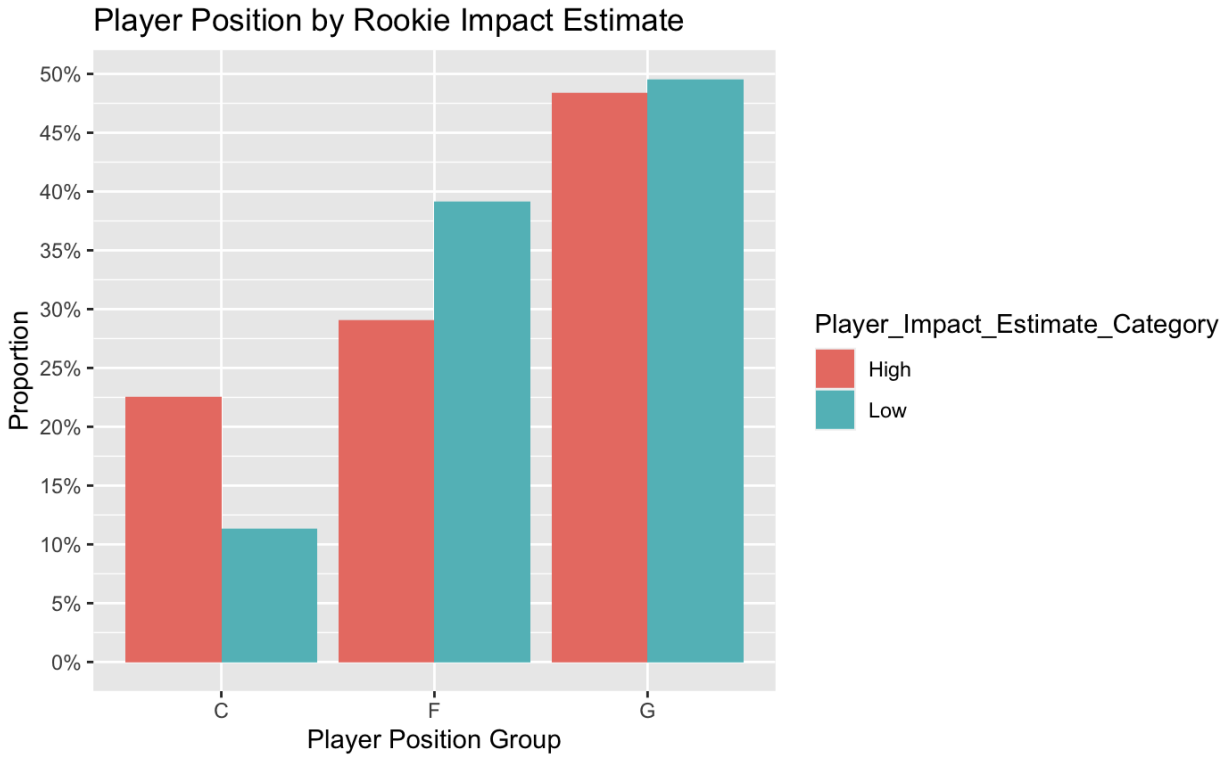
The next set of graphs detail the distribution of the categorical variables in the data set, player grade at draft and player position. In order to account for the different sizes of the two impact level groups the y axis measures the percentage make up of each category in its respective impact level grouping rather than overall counts. We can see in the distribution of the grades at draft that there is a large gap between the percentage of impactful freshmen and non-impactful freshmen. This was relatively expected as those

players that seem to be clearly playing at an elite level will get drafted as soon as possible rather than wasting time with developing their game in college and since they got drafted without needing the time to develop, teams likely consider them on the more talented end of the spectrum even amongst elite college players. It is interesting that such a high percentage of players are drafted in their senior year, though it is not unexpected as those players that arent top level stars likely need as much time as possible to develop before making the jump to the professional league.



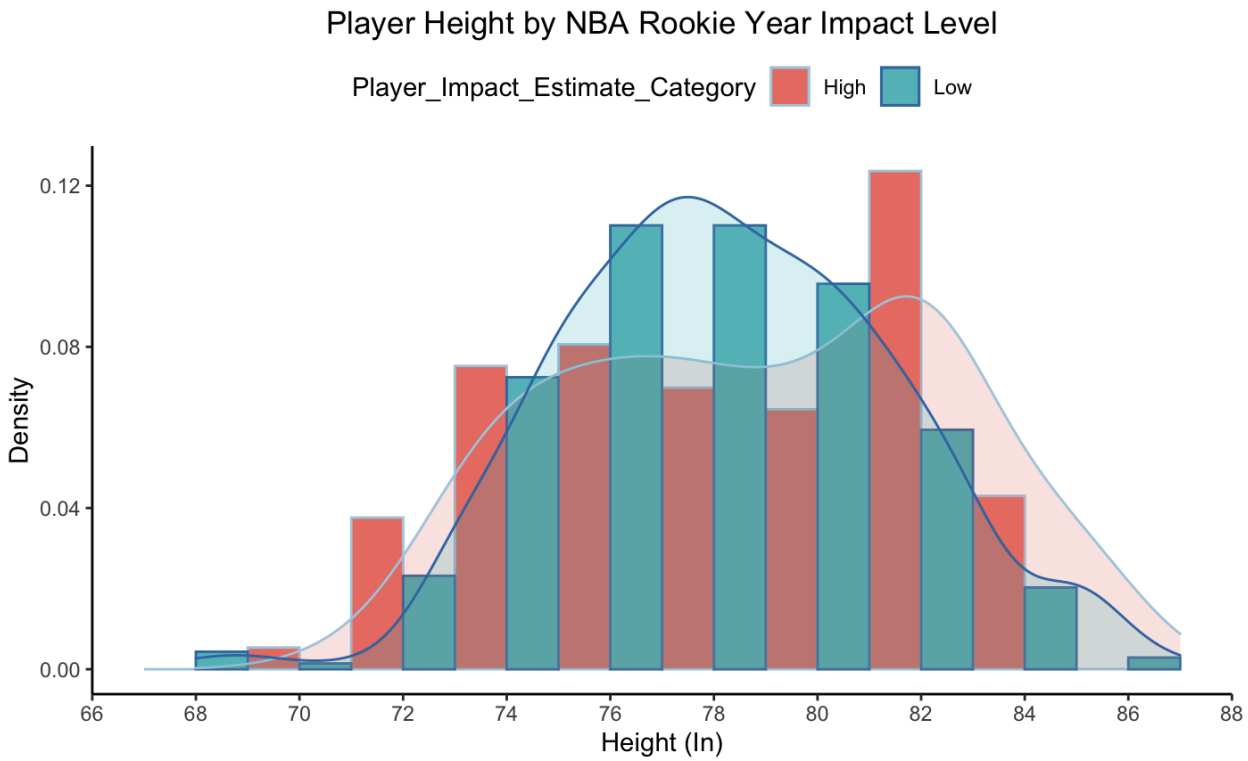
The breakdown of positions between the two impact levels is quite interesting as there is an over +10 point differential in the amount of impactful players that are centers vs non impactful players. As stated above, this is likely because centers, or the ‘big men’, rely on their easily witnessed physical assets more than other positions. While a guard may rely on learned skill with how to pull of certain shots or dribble around other

players, one cannot teach a center how to be tall, strong, or have a large wingspan, which can be measured easily. Since centers are so large they generally rely very little on many finesse skills and as such there is very low floor as compared to the other positions. It is also interesting that there is such a high percentage of non impactful forwards, this is likely due to the fact that forwards must have a delicate balance of strength, speed, and skill, which may make it more difficult to judge a player's NBA readiness based off of easily discernible physical metrics. For example, if a forward is too slow and bulky they may be unable to guard or get around other quicker players, however if they are fast but not strong enough they may get bullied by larger players.

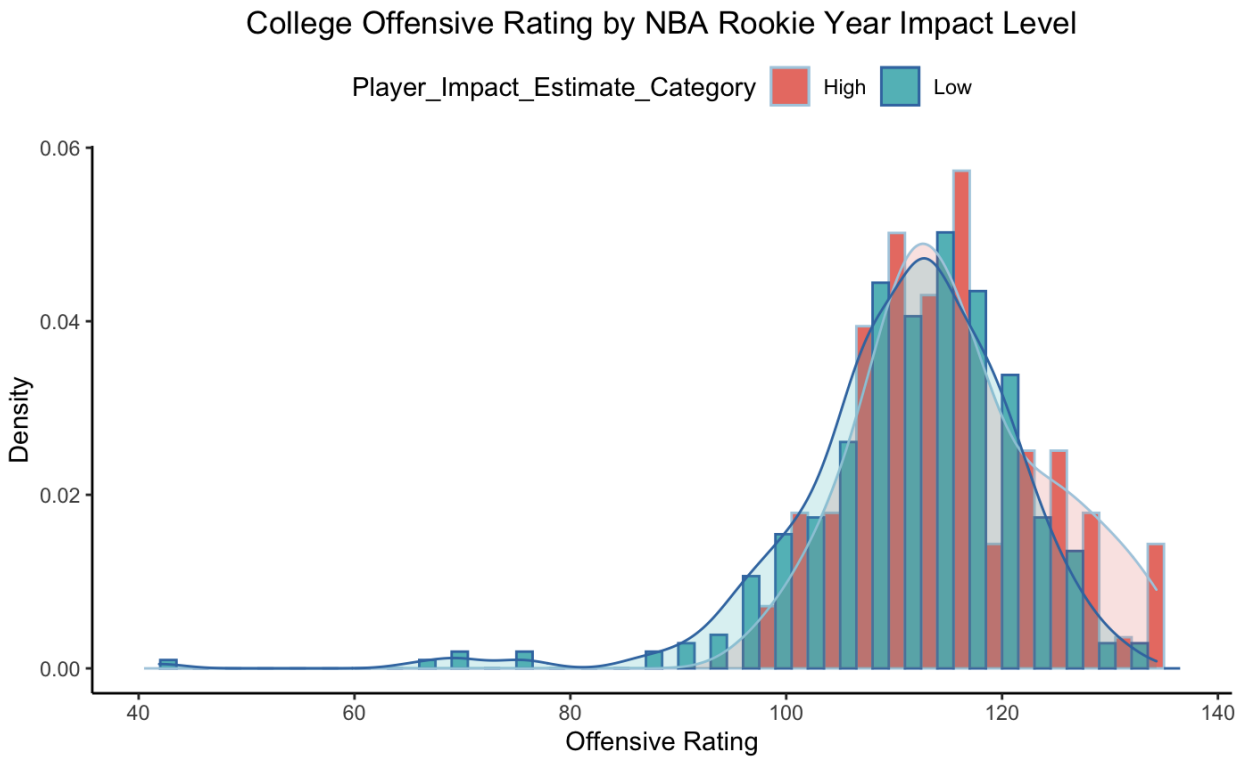


A further examination of the densities of heights of the NBA rookies in their last college year further shines light on the amount of impactful centers. While the distribution for heights of low impact players seems relatively bell shaped with the peak being at

approximately 76-78 inches, the distribution for high impact players is left skewed with the peak density occuring around the 82 inch mark. This further shows that the larger players, most likely centers, come in as higher impact players at a greater rate than their smaller counterparts.

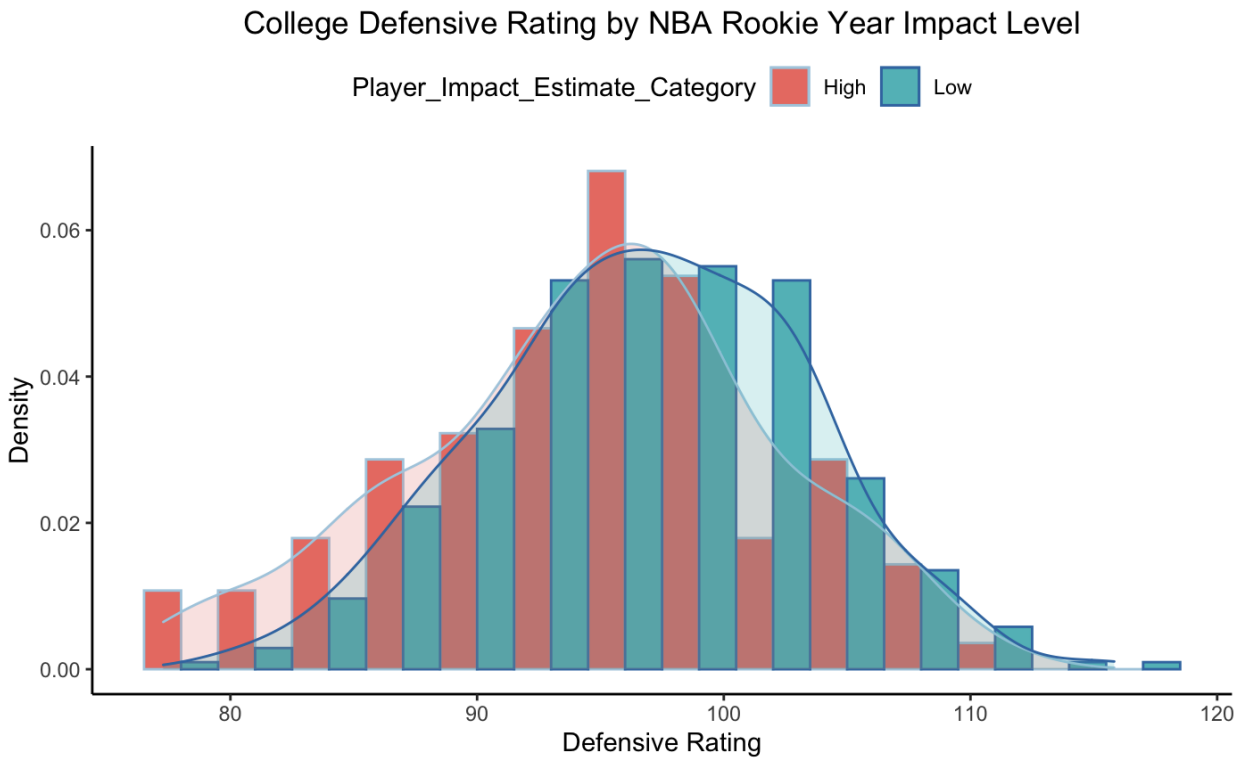


The next two graphs are basic measures of offensive and defensive rating for these players. It can be seen that a strong offense may be a good indicator of players readiness for the NBA. It can be seen that the distribution for low impact players is right skewed with a trailing tail of low offensive rating players, while it can be seen that there is a larger density of high impact players on the higher end of the offensive rating scale. It seems like both groupings have their peak density at ~ 115 but the high impact grouping has a larger proportion of players than the low impact group in this range.



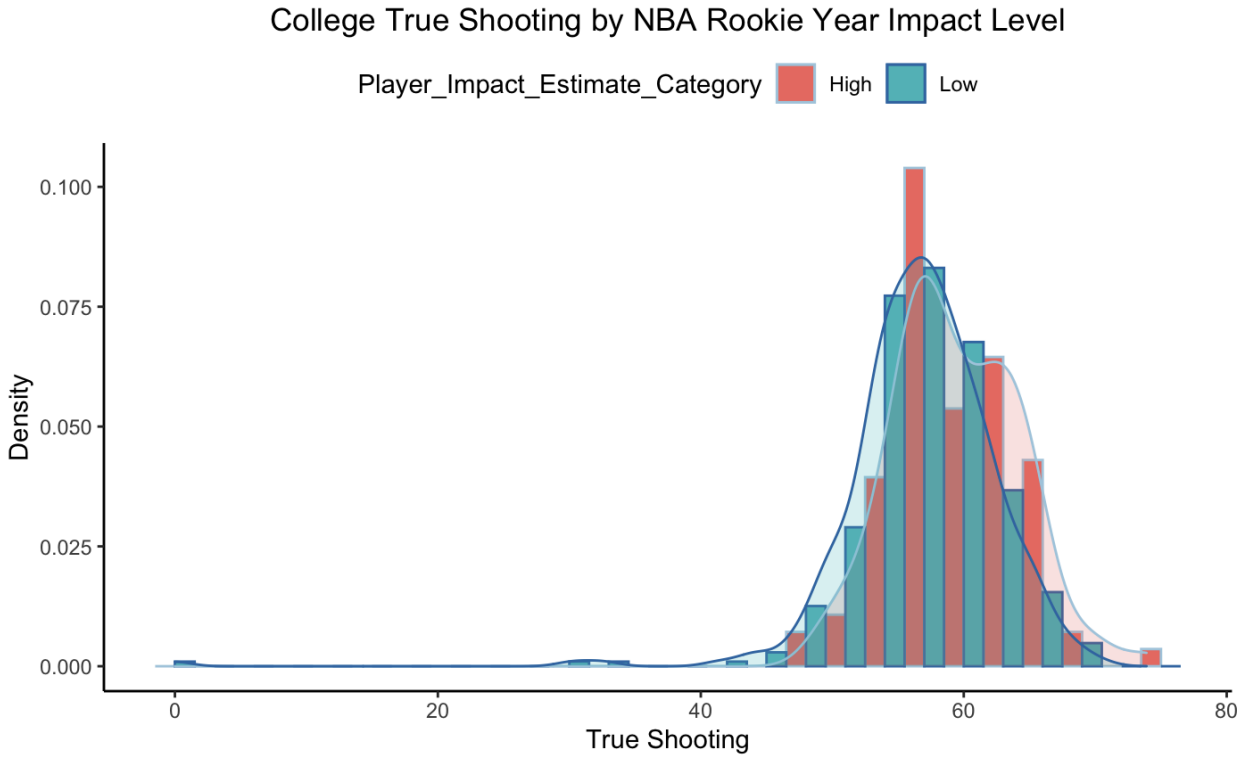
Unlike offensive rating it seems that for defensive rating the positions are switched with a higher density of the impact players being on the lower end of the defensive rating scale, while it seems like a much higher percentage of low impact players are on the 100+ range of the defensive rating scale. There is a large drop off in proportions of high impact players at the 100 point defensive rating mark with a large peak around the 95 point mark. It could be that defensive statistics do not transfer over as well to the nba as offensive statistics, which makes sense since defensive impact is notoriously hard to quantify. There are an extreme amount of intangibles which can be seen but have difficulty being translated to quantifiable statistics, like a defender causing players to change their behaviour to avoid them without directly guardian said player, that make defensive ratings difficult to gauge accurately at times. In contrast, offensive statistics are much more easily quantified as assists, free throws, and points scored are easy to

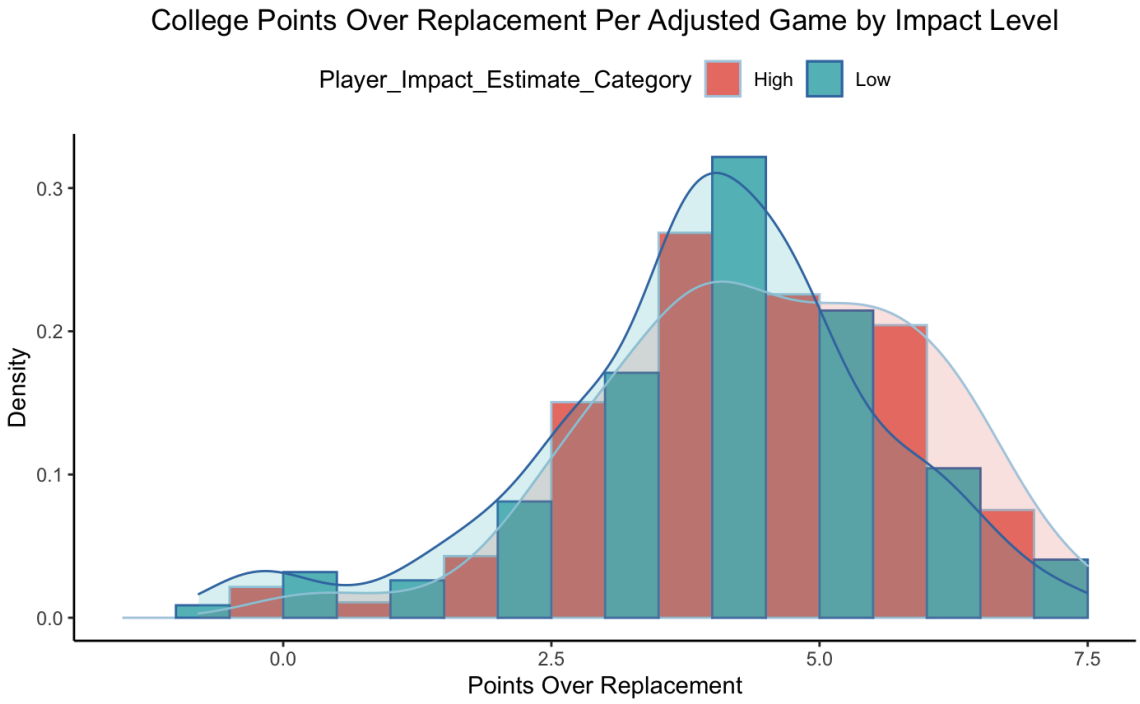
see in the box score and quantify. It could also be that it is more difficult translate defensive skills to the NBA against faster and stronger players as opposed to offensive skills, causing defensive rating to be a poor measure of players future impact.



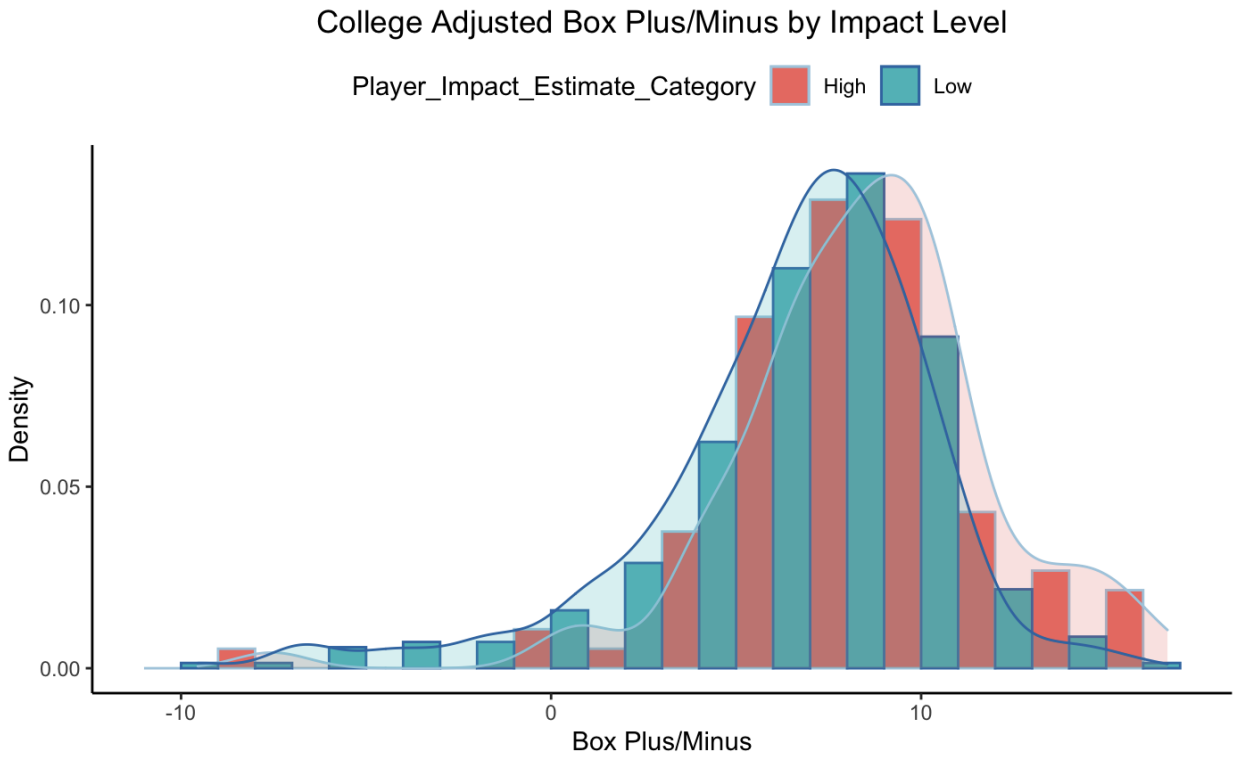
True shooting and Points Over Replacement are another two statistics that are generally regarded as good measures of a players offensive prowessor. For true shooting, much like offensive rating it can be seen the the low impact players have a long tail on the lower end of the distribution, though otherwise retaining a relatively bell curve shape. The high impact players again have a much higher density at the peak of their bell curve, but the bell shape does not hold for the right side with a much higher density of high true shooting players. For Points Over Replacement, the low impact players' distribution generally maintains its bell shape, while the distribution for the high

impact players flattens greatly looking like a hill rather than a bell with a larger proportion of players on the higher end of the spectrum.



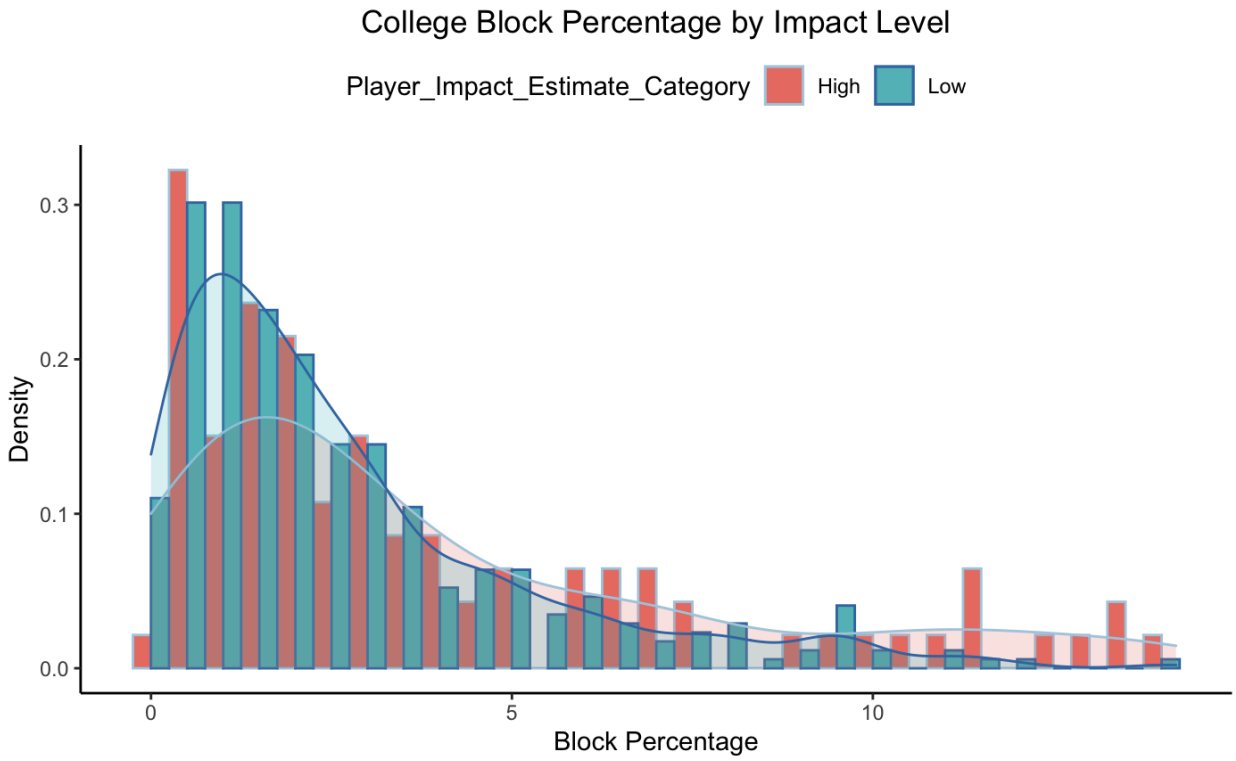


The next measure, Adjusted Box Plus/Minus, is considered to be decent measures of players overall performance. Adjusted Box Plus/Minus takes into account both defensive and offensive prowess as well as the difficulty of opponent faced. Here the difference in the two distributions is clearly visible where the peaks of the two bell shapes are distinctly separate. The low impact group has a longer tail to the lower side of the rating scale and the higher impact group distributions is shifted towards the higher end of the scale. There is also a higher density on the right tail compared to the drop off indesnity to the left of the median.



One defensive metric that actually seems to be distinct between the two groups is block percentage. Block percentage is the amount of estimated percentage of possible blocks that a player could make while he was on the floor. Both distributions here are strongly

right skewed however the high impact players have a much higher density on the tail and a lower peak.



IV Machine Learning Algorithm

4.1 XGBoost Algorithm

The XGBoost Algorirthm is an implementation of the gradient boosted decision tree model designed for speed and performance. Gradient boosting is a type of machine learning boosting, meaning that it relies on the intuition that the best possible next model, when combined with previous models, minimizes the overall prediction error. At its core the algorithm iterates over many decision trees, using the previous nodes to continually provide more information, using the gradient of the loss function to identify

shortcomings of week learners. The xgboost package in r was utilized in implementing the gradient boosting algorithm for this project.

4.2 Initial Model

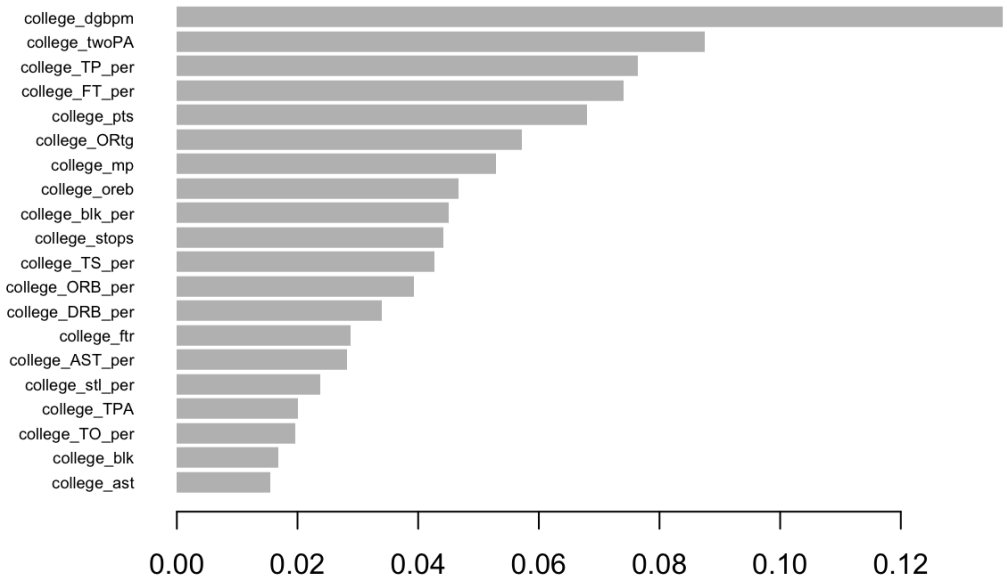
The first step in training our model was to divide the data into training, validation, and testing sets. A 70, 15, 15 split was used for each of the respective groupings. The training set was the data set aside for the model to learn on. When training the model utilized the validation set to target and asses how it was doing, essentially a test set for the training of the model, and a final test set containing 15% of the data points was set aside to asses the accuracy of predictions. Without the use of a validation set, the model may overfit when actually testing prediction performance.

The first step after we created our training, validation, and test subsets of the data, removed the labels, and transformed them into a labeled xgboost matrix is to utilize k-fold cross validation. The xgb.cv function allowed me to perform a 10-fold cross validation on the training data in order to find the optimal number of boosting iterations for training the model, which turned out to be 4. Once the cross validation was completed, it was time to begin training the model. For the initial iteration default learning parameters were kept and the maximum number of boosting iterations was set to 4. Once this model was trained on the training and validation data sets, tis performance could be measured.

```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 40 10
         1  9  7

               Accuracy : 0.7121
                 95% CI : (0.5875, 0.817)
    No Information Rate : 0.7424
    P-Value [Acc > NIR] : 0.7627

                  Kappa : 0.2326

 Mcnemar's Test P-Value : 1.0000

            Sensitivity : 0.8163
            Specificity : 0.4118
```

The confusion matrix above shows the predicted vs actual results from the initial base model. It can be seen to have a relatively high accuracy at 71% for what was expected, however it seems to have difficulty in identifying the high impact players. After testing out different percentage splits for what would be considered a prediction of high or low impact, it chosen that anything above 30% would be classified as a high impact prediction. This is because the model has decent amount of trouble classifying high impact players, generally giving low confidence levels for most predictions of high impact. This was somewhat to be expected as predicting high impact players from college statistics is an incredibly challenging task due to the high amount of important basketball skills that have difficulty being translated to the scoreboard, with the ‘eye test’ being the only real ability to measure these skills. Even then, many teams that have a large amounts of scouts on their payroll and an analytics department still fail to always draft the right talent. The model had a sensitivity of ~81.63% meaning that it was great at spotting the true negatives and not assigning false positives, however the specificity was much lower at ~41.18% showing a difficulty in identifying the true positives.

We can see below that the most important feature was identifying to be adjusted defensive box plus/minus, which is interesting as the main differences noted in the EDA were for the offensive stats. The next five statistics were all offensive denoting two point shot attempts, free throw percentage,three point percentage, points, and offensive rating. Offensive ratings importance was relatively in line with the differences seen in the graphical analysis, however it was interesting to see how important the base box score statistics were considering they weren’t as largely correlated with the response variable.



4.3 Tuning our Model

While the initial model provided interesting results, it is likely not the optimal possible configuration as the hyperparametrs used were just the default values. For the model to perform at an optimal level, it must be first be tuned, or have all the hyperparameters set to the values that allow for the best classification accuracy. To do this the mlr R package was utilized. The dataset was broken up into the exact same test, training, and validation sets as used in the base model and the subsets were loaded into a classification task object (matrix object with the targeted classification variable passed as an argument). Then a learner object is created, as well as parameter object which describes which parameters to tune and what range of values to tune them within. The random search technique was chosen to tune the hyperparamers which is where random combinations of the hyperparameters are used to find the best solution for the

built model. This technique was chosen because it has a 95% probability of finding a combination of parameters within the 5% optima with only 60 iterations. Once all of the necessary elements for tuning were properly set up they were passed to the tuning function which performed the random search and outputted what the function found to be the optimal set of hyperparameters. These hyperparameters were then passed to another model which trained on the same training data set as our base model. This new tuned model should in theory outperform the original base model. In order to test this belief another confusion matrix was created and compared to the one from the original model.

```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 44  9
         1  5  8

               Accuracy : 0.7879
                 95% CI : (0.6698, 0.8789)
    No Information Rate : 0.7424
    P-Value [Acc > NIR] : 0.2446

                  Kappa : 0.3992

 Mcnemar's Test P-Value : 0.4227

            Sensitivity : 0.8980
            Specificity : 0.4706
```

We can see from the confusion matrix above that the new model does indeed perform at a higher level than our original model with the base parameters as the accuracy increased by nearly 7% due to a jump in both specificity and sensitivity. This model is greatly preferable to the base model, as it is actually able to classify impactful rookies to some degree of accuracy and does not misscalssify non-impactful rookies to the same

degree as the previous model. It is more damaging to a sports franchise to draft a rookie who won't be impactful rather than miss on a possible impactful rookie and this model has a much smaller chance of that happeding as compared which incorrectly classified more non-impactful rookies than it correctly identified impactful ones.

variable <chr>	importance <dbl>
college_ORtg	0.03178175
college_usg	0.04419759
college_TS_per	0.04253564
college_ORB_per	0.01323344
college_DRB_per	0.02645657
college_AST_per	0.05382241

The above table shows the most important variables for the tuned model. It is interesting that the basic shot statistics were lost in favor of advanced statistics like usage rate and true shooting. It also seems that rebound and assist percentages have a much higher weight than in the previous model, which is more in line with what we saw in our exploratory data analysis.

V Conclusion

While a difficult challenge, I believe it has been shown that predicting the impact of NBA rookies before they are drafted is possible. Statistical analysis of sports is notoriously difficult due to all of the intangibles that fail to translate to box score statistics, however with the advancements in stat tracking machine learning, used in combination with traditional scouting, will be an increasingly powerful tool predicting for the usefulness of college players before they make the jump to the big leagues. Offensive measures seem to be translate to the professional best, likely due to the the difficult to quantify nature of defensive skills which advanced statistics still fail to capture well. Defensive statistics are important, as rebounds were shown to be one of the more significant factors in determining player impact, and will likely continue to become more important as more advanced measures of defensive prowess evolve. More research should still be done currently to refine the model presented in the study and a larger data set of college players would likely greatly improve the accuracy of the model. Furthermore, the model presented could be improved by a lengthier analysis of advanced metrics in order to create a composite metric that avoids the large biases present in many advanced statistics.

References

1. https://barttorvik.com/trank.php#
2. https://towardsdatascience.com/using-machine-learning-to-predict-nba-all-stars-part-1-data-collection-9fb94d386530
3. https://www.basketball-reference.com/about/bpm2.html
4. https://www.basketball-reference.com/about/glossary.html