DS100 Final Project: Analyzing the Relationship Between NCAA Statistics and NBA Points Per Game to Predict the Success of a Player Entering the NBA from the NCAA

Author: Swadhin Nalubola

Abstract

This paper attempts to model and predict a player's points per game (ppg) average in the NBA, given solely their NCAA statistics and basic information (height, weight, age, etc.). This paper starts by processing a provided "college" csv dataset using Pandas – the data is cleaned to remove unnecessary columns (NBA statistics), reformat unworkable columns (categorical variables), and add desired columns (determined through exploratory data analysis). Throughout this process, Seaborn visualizations are utilized to better understand the data and decide what information is relevant or irrelevant. Following the cleaning process, a Lasso and Ridge Regularization model are both used to model the data; the ideal hyperparameter for each is chosen through 5-fold cross validation error. After optimizing each, a final choice is made based on whether the Lasso or Ridge has a lower cross validation error. The paper then concludes with an evaluation of the chosen model, methods used, and decisions made. Finally, suggestions for future studies are made.

[1] Although it is impossible to know the exact number of years a player will spend in the NBA when they first enter, it is fairly reasonable to include this in our model. For future prediction purposes, an NBA rookie would be asked for the amount of time they plan to spend in the league, and this number would become the input for the model.

Introduction

Prior to the start of every NBA season, all 30 NBA teams participate in the NBA draft. During the draft, every team chooses new players that will be joining them in the coming season. Considering the competitiveness of the league and the amount of money involved in NBA contracts, it is incredibly important for a team to draft well – an extensive amount of research goes into analyzing each draft prospect to ensure that a given team can draft exactly who they need and spend their money responsibly. As many of these draft prospects played college basketball in the NCAA prior to declaring for the NBA, NCAA statistics are often consulted to assess the potential of a given player.

This paper examines the NCAA and NBA statistics of all players in the NBA, going all the way back to the beginning of the league. We will attempt to predict the success of a player in the NBA given solely their NCAA statistics, along with some other relevant information. Here, success is measured by points per game (ppg) in the NBA – this paper will characterize and attempt to model the relationship between NCAA statistics and NBA ppg. In the following pages, we will go through a description of the data used (including exploratory data analysis and data cleaning procedures), a description of the methods (including an explicit definition of our model), a summary of the results, and a discussion of our findings.

Description of Data

To start, there were multiple datasets available for analysis. Ultimately, only the "college" dataset was consulted, as the others did not contain relevant information for our purpose. The "college" dataset includes NBA statistics for every player going back to the very beginning of the league in 1946. Along with this, each player's NCAA statistics are included. For players that did not play in the NCAA, did not attend college, or played in a college not part of the NCAA, there are no values for their NCAA statistics. Each row of the table corresponds to a different NBA player, and these are organized by alphabetical order. The columns of the table include basic information, NBA statistics, and NCAA statistics. For a full list of columns and explanations of their abbreviations, please consult the appendix. Our csv file was read into a Pandas DataFrame, and all future processes were performed in a Jupyter Notebook. A DataFrame named "raw" was used for the raw data from the csv file, and a DataFrame named "cleaned" was used for the cleaned data that would be utilized in modeling processes ("cleaned" was initialized as a mere copy of "raw").

Even though this data includes a plethora of columns, only some of them are useful to us. Many of them, in fact, need to be interpreted in a different form to become useful for predicting NBA ppg. To start, the active_from and active_to columns indicate the years in which an NBA player is active in the NBA – although potentially useful information, the specific years of an NBA player's career could skew our predictions for ppg. The playstyle of the NBA has changed significantly over time, and the ppg averages in the league have certainly reflected that (Wilt Chamberlain averaged 38.4 ppg in 1960, but such a number would be completely absurd in today's game). For this reason, a new column called "years_in_NBA" was added, based on active_to minus active_from. This column represents the number of years a player is active in the NBA and is much more useful than the specific years that somebody joined and left the league.[1]

The next column that was modified was the "birth_date" column. As it is written out (ex: June 12th, 1986), this column is largely useless for a computational model. That being said, the

[1] Although it is impossible to know the exact number of years a player will spend in the NBA when they first enter, it is fairly reasonable to include this in our model. For future prediction purposes, an NBA rookie would be asked for the amount of time they plan to spend in the league, and this number would become the input for the model.

age that someone enters the league can often be an indicator of their maturity and ability to adapt to the fast-paced play of the NBA. For this reason, a column "age_entering_NBA" was created, which subtracts the year element of birth_date from active_from to calculate the age at which a given player entered the NBA. With this modification, the birth_date column is no longer important, and the relative age of players can still be factored into our model.

The "height" column is listed in the format of "feet-inches." Again, this information is not fit to be inputted directly into a computational model. Using a regular expression, a function was created to read a string in the format of feet-inches and return the total height in inches (feet * 12 + inches). This function was applied to the height column and returned as a new column in the DataFrame named "height_inches." This column contains an integer and is useful for our model.

Finally, the "position" column needed to be addressed. As a categorical variable, position serves little to no purpose without some sort of data cleaning. In this case, one-hot-encoding was used to convert the singular column of position into multiple columns, one for each possible category of position (C, C-F, F, F-C, F-G, G, and G-F). In each column, a value of 0 indicates that a player is not this position, while a value of 1 indicates that a player is this position (naturally, a 1 is present in only one of these seven columns with a 0 in the remaining six). The function to one-hot-encode the position column was largely sourced from hw6 of the DS100 course UC Berkeley and utilizes the DictVectorizer class from the Scikit-learn package.

At this point, the "cleaned" DataFrame contains all of the raw data from our "college" csv file, along with the addition of multiple columns for years in the NBA, age entering the NBA, height in inches, and position. To conclude our initial phase of data cleaning, all irrelevant columns were removed. This included columns that had already been accounted for, all NBA statistics except for ppg, and some other completely irrelevant columns.

Although the dataset had now been cleaned to include only the desired information, it was still rife with NaN values that would be unusable by our model. Before inputting our data into a model, this information needed to be removed. In this cleaning process, we started by identifying all the columns with a NaN value and noticed that the following columns had such values: "height_inches", "weight", "age_entering_NBA", "college", and all NCAA statistics columns. For each column, it made sense to identify the specific rows with NaN values and determine if these rows were worth salvaging. As it would turn out, every row that had a NaN value in "height_inches", "weight", "age_entering_NBA", or "college" had absolutely no information on NCAA statistics – these rows were all removed, and the number of NaN values in columns was recounted.

At this point, there were still plenty of NaN values in the NCAA columns, all of which needed to be addressed before using our model. We started by removing all rows with a NaN value for "NCAA_ppg", as these rows have no NCAA statistics at all. In the resulting table, there were 16 players with some NCAA statistics, but without many of the more important statistics (fgpct, fgapg, ft, ftapg). In removing these 16 players, there were now only 3 columns with NaN values – "NCAA__3ptpg", "NCAA__3ptapg", and "NCAA__3ptpct". All our NaN values were concentrated in columns relating to NCAA 3-point statistics. Closer examination of the raw dataset and some research provides some insights into this – the NCAA 3-point line was not implemented until 1986, and so 711 players have no statistics for

[1] Although it is impossible to know the exact number of years a player will spend in the NBA when they first enter, it is fairly reasonable to include this in our model. For future prediction purposes, an NBA rookie would be asked for the amount of time they plan to spend in the league, and this number would become the input for the model.

their NCAA 3-point shots made per game (3ptpg), attempted per game (3ptapg), or percentage (3ptpct). There are 3 players with a 0 in the 3ptpg column, but a NaN in the 3ptapg. There are 142 more players with a 0 in the 3ptpg and 3ptapg columns, but a NaN in the 3ptpct column. The 711 players with no NCAA 3-point statistics were placed into a separate DataFrame and saved for possible future use (ultimately, these players went unused but could be useful in future studies). The 3 players were removed without conservation, as there did not seem to be a reasonable explanation for their information (or lack thereof). The 142 players were given a 0 in the "NCAA__3ptpct" column, as they attempted no 3 pointers and made no 3 pointers, yielding a whopping 0% 3-point percentage. With these revisions, our data was now free of NaN values and almost ready for our model.

Every column in the "cleaned" DataFrame was now numerical and without NaN values, but there was still one categorical variable: "college." Surely, this should be a useful feature for our model – many colleges are known for their ability to consistently churn out promising NBA prospects (Kentucky, Kansas, and Duke to name a few). To start examining the potential relationship between NBA ppg and the college that a player comes from, we constructed a Seaborn Boxplot that displays the NBA ppg of players from colleges with at least twenty NBA players. Additionally, this plot includes a red line to represent the median ppg of all NBA players in our dataset at this point, and a blue line to represent the median ppg of solely NBA players from this list of common colleges (at least twenty NBA players). Figure 1 in the appendix shows this plot.

Looking at our plot, we can see that the median ppg for players from these colleges is slightly higher than our overall median ppg. Some colleges have significantly higher ppgs, but we cannot factor this in without considering the number of players from a given college. Before analyzing which colleges have high median ppgs, we first looked at a potential relationship between number of NBA players and median ppg. Does producing more NBA players mean that a college tends to produce players with a higher ppg? We answered this question through a Seaborn Jointplot of each college with one axis representing number of NBA players and the other representing median ppg of these players. For viewability and interpretation purposes, this plot only includes colleges that have produced at least five NBA players. Figure 2 in the appendix shows this plot.

Looking at this plot, there does not seem to be a reliable relationship between number of NBA players produced and median ppg of these players. Since we could not generalize the NBA ppg of players from a college based on the number of players from this college (our only quantitative differentiator of colleges), we chose to instead assess only a few select colleges. Any college that produced at least five NBA players and maintained an median NBA ppg of at least 6.5 for these players was considered one of the best colleges. We then created a column in our DataFrame called "from_best_colleges" that served as an indicator for whether or not a given player was from one of the best colleges. At this point, there was more than enough information on our players, and we were ready to move on to constructing a model for predicting NBA ppg.

Description of Methods

In choosing a model, we wanted to select one that would be able to accommodate the large number of features and avoid overfitting. For that reason, we decided to assess both a Lasso and Ridge Regularization model, comparing them based on their lowest cross validation error. To start, the "cleaned" DataFrame was split into "X" and "Y" – X included all numerical information that would be input to the

[1] Although it is impossible to know the exact number of years a player will spend in the NBA when they first enter, it is fairly reasonable to include this in our model. For future prediction purposes, an NBA rookie would be asked for the amount of time they plan to spend in the league, and this number would become the input for the model.

model, and Y included solely NBA ppg, as this is what is trying to be predicted. The "train_test_split" function from Scikit-learn's model selection package was used to randomly split the data into a 90% and 10% split of training and testing data, respectively. Additionally, normalized versions of the training and testing X data were created through the StandardScaler class from Scikit-learn's preprocessing package. Normalization of our data is necessary to ensure that the range of values in a given column does not impact the weight of this column – percentage columns are always between 0 and 1, but this does not mean it should not be as important as some of the other columns present. Finally, a function to calculate root mean squared error was created for use in the "cross_val_score" function from Scikit-learn's model selection package.

Using the Pipeline class from Scikit-learn's Pipeline package, a Lasso Regularization model with a StandardScaler transformer and maximum iteration of 1000000 was created. This model was tested with a variety of possible hyperparameters, ranging from 0.001 to 0.5 (this range was chosen through brief iterations of trial and error to identify the range of hyperparameters with the lowest cross validation error). For each hyperparameter, the cross validation error was calculated with a cross validation of 5, and the hyperparameter with the lowest cross validation error was chosen as the best for the Lasso Regularization. Figure 3 in the appendix shows the cross validation error for varying hyperparameter (alpha) values. The lowest cross validation error for the Lasso Regularization model was 3.092, but more importantly the ideal hyperparameter was as close to 0 as possible. This indicated that our model was approaching linearity, and some columns were effectively useless in this model. Before identifying these columns, we analyzed the Ridge Regularization to see if the Lasso model was worth pursuing – if the Ridge model results in a lower cross validation error, the Lasso model need not be analyzed further.

A Ridge Regularization model was created in a similar fashion, using the Pipeline and StandardScaler classes. This model was also tested with varying hyperparameters, but the range was 0.1 to 2.5 (determined in a similar fashion). Cross validation error was again calculated with a cross validation of 5, and Figure 4 shows a similar plot of cross validation error against varying hyperparameters. The lowest cross validation error was 3.090 for a hyperparameter of 1.280. This error is lower than that of our Lasso model, and our Ridge model is thus the optimal choice moving forward.

Summary of Results

Having concluded that a Ridge Regularization is our best choice for our model, we wanted to briefly assess the accuracy of our model. To do this, we calculated the root mean squared error for both our training and testing datasets. Along with this, Figure 5 plots our testing data, showing the predicted NBA ppg against the actual NBA ppg. Additionally, the red line represents perfect prediction in which the actual and predicted ppg are the exact same. Our RMSEs of 3.053 and 3.523 for the training and testing datasets, respectively, are by no means perfect. However, for a student-constructed model of a hotly debated topic in today's league, it isn't bad! That being said, our scatter plot highlights some obvious shortcomings of our model – to start, the scaling of the axes for our predicted and actual ppgs are different. The predicted ppg ranges from -2.5 to 17.5, while the actual ppg ranges from 0 to 20. As a negative ppg is impossible, our model needs to be adjusted to not predict negative values. Additionally, the model needs to be able to predict values above 17.5, as a fair amount of

[1] Although it is impossible to know the exact number of years a player will spend in the NBA when they first enter, it is fairly reasonable to include this in our model. For future prediction purposes, an NBA rookie would be asked for the amount of time they plan to spend in the league, and this number would become the input for the model.

data points are above 20 for their actual ppg. Finally, we can observe that our model does not tend to over or under predict, as there is a roughly equal number of points both above and below the red line. With this, we have constructed a somewhat useful model for predicting the NBA ppg of a player given their basic information and NCAA statistics.

Discussion

In cleaning and analyzing the provided "college" dataset, we made many different observations regarding both the data itself and the decisions we made in coming to our final conclusions. In constructing our model, we added a plethora of features, ranging from simple NCAA statistics to a one-hot-encoding of position. One especially interesting addition was the "from_best_colleges" feature, which served as an indicator for whether or not a player was from a specific list of elite colleges. This list was based on our analysis of the data, but it ended up as quite dissimilar to many of the usual lists of top colleges and presented a variety of challenges.

In most NBA scouting, the top colleges are widely considered to be those that have produced the most NBA players. But in our analysis, we wanted to make sure that these colleges were resulting in more successful NBA players. At the very least, we wanted to include colleges in our model in some form due to their importance in assessing NBA prospects. To analyze the quality of player produced by a college, we looked at the median of a college's players' ppgs (thereby mitigating the influence of outliers). We did not want to use averages, as this would be more akin to assessing colleges based on their probability of producing a superstar player; rather, we wanted to assess whether these colleges consistently produced players that were better than the rest of the league. When the median ppgs of colleges were compared, few colleges had a ppg significantly higher than the overall median ppg of players. As it would turn out, two colleges (UCLA at 5.7 and Kanas at 4.9) that fell into this category had produced over forty NBA players. With this discovery, we realized that we could no longer assess each individual college. Originally, we had planned to assign each college a score based on its number of NBA players and median ppg. This score would then be transferred to players from that college, and we would be able to utilize this score in predicting a player's NBA ppg. However, Figure 2 clearly demonstrated that count and ppg could not be strongly correlated, thus rendering our earlier plan ineffective. Instead, we constructed the list of elite colleges, judging them as those with a median ppg of at least 6.5 and a count of at least 5 NBA players. As it forced us to both reconsider earlier notions of prestigious basketball colleges and modify our plan for including colleges in our model, "from_best_colleges" served as both a challenging and rewarding feature to implement.

With regards to limitations of our analysis, there is no doubt that there are plenty of opportunities for improvement. As mentioned earlier, model's difficulty in avoiding negative values and predicting higher ppgs is quite obvious. For example, in Figure 5, it can be seen that nearly all actual ppgs that were above 15.0 were in fact predicted to be below 15.0. To address this, these datapoints should be examined to identify and incorporate similarities into the model. Another possible limitation of the model lies in its use of the "years_in_NBA" feature. This feature can only be completely accurate for players that have already completed their NBA career – naturally, a player entering the league cannot give a perfectly accurate answer for the number of years they will spend in the NBA. For new players, this feature would be almost entirely arbitrary, as a player would simply be

[1] Although it is impossible to know the exact number of years a player will spend in the NBA when they first enter, it is fairly reasonable to include this in our model. For future prediction purposes, an NBA rookie would be asked for the amount of time they plan to spend in the league, and this number would become the input for the model.

providing an estimate for how long they want to spend in the NBA. Improving this presents a new challenge, as it is impossible to know this information for a player entering the NBA. Instead, this model could be modified to predict the NBA ppg for any player in the NBA, and this feature would instead represent the number of years a player has spent in the NBA thus far (of course, this model would also include NBA statistics for players already in the league, and some NCAA data could be rendered moot). Finally, the model itself is a limitation – Lasso and Ridge Regularization may not be the best choice, as the data does not necessarily favor a linear model. A Decision Tree could be used instead, but this would most likely require slight modifications to the input data. As opposed to its current format, the decision tree data would have more indicator variables, and the list of colleges would almost certainly be separated into multiple tiers. Additionally, information regarding a player's accolades while in college (national awards, tournament championships, etc.) could be incorporated to greatly assist the model with more indicator variables.

For future studies, this model should be supplemented with more information. NCAA statistics alone cannot predict a player's NBA ppg with significant reliability. This does not come as much of a surprise, as many players do not flourish until the NBA. Along with NCAA statistics, the model should include any individual or team accomplishments of the player, their statistics in tournament or rivalry games (to assess performance under pressure), and the average statistics of their teammates (to determine whether they were a standout on their team). Additionally, the model could be modified to predict more than just ppg in the NBA. It could attempt to predict virtually any other NBA statistic, but some notable ones include 3-point percentage, assists per game, rebounds per game, steals per game, and blocks per game. Successful 3-point shooters in the NCAA will most likely continue their success in the NBA, and the same concept applies for assists, rebounds, steals, and blocks.

This study raises few to no ethical concerns or dilemmas, so plenty of additional research and analysis can be conducted to further optimize the predictions made by this model.

[1] Although it is impossible to know the exact number of years a player will spend in the NBA when they first enter, it is fairly reasonable to include this in our model. For future prediction purposes, an NBA rookie would be asked for the amount of time they plan to spend in the league, and this number would become the input for the model.

Appendix

Column Label	Description
active_from	Year that the player started playing in the NBA
active_to	Year that the player stopped playing in the NBA
birth_date	Birth date of the player in the format "Month Day, Year"
college	College(s) the player attended
height	Height of the player in the format "Feet-Inches"
name	Name of the player
position	Primary position of the player
url	Url to access the player's full information
weight	Weight of the player in pounds
NBA__3ptapg	NBA 3-point shots attempted per game by the player
NBA__3ptpct	NBA 3-point shooting percentage of the player
NBA__3ptpg	NBA 3-point shots made per game by the player
NBA_efgpct	NBA effective field goal shooting percentage of the player
NBA_fg%	NBA field goal shooting percentage of the player
NBA_fg_per_game	NBA field goals made per game by the player
NBA_fga_per_game	NBA field goals attempted per game by the player
NBA_ft%	NBA free throw shooting percentage of the player
NBA_ft_per_g	NBA free throws made per game by the player
NBA_fta_p_g	NBA free throws attempted per game by the player
NBA_g_played	NBA games played by the player
NBA_ppg	NBA points per game average of the player
NCAA__3ptapg	NCAA 3-point shots attempted per game by the player
NCAA__3ptpcg	NCAA 3-point shooting percentage of the player
NCAA__3ptpg	NCAA 3-point shots made per game by the player
NCAA_efgpct	NCAA effective field goal shooting percentage of the player (null for all rows, not calculated)
NCAA_fgapg	NCAA field goals attempted per game by the player
NCAA_fgpct	NCAA field goal shooting percentage of the

[1] Although it is impossible to know the exact number of years a player will spend in the NBA when they first enter, it is fairly reasonable to include this in our model. For future prediction purposes, an NBA rookie would be asked for the amount of time they plan to spend in the league, and this number would become the input for the model.

	plaer
NCAA_fgpg	NCAA field goals made per game by the player
NCAA_ft	NCAA free throw shooting percentage of the player
NCAA_ftapg	NCAA free throws attempted per game by the player
NCAA_ftpg	NCAA free throws made per game by the player
NCAA_games	NCAA games played by the player
NCAA_ppg	NCAA points per game average of the player

Table 1. The names and descriptions of all columns present in the "college" csv file that was used as the raw data.

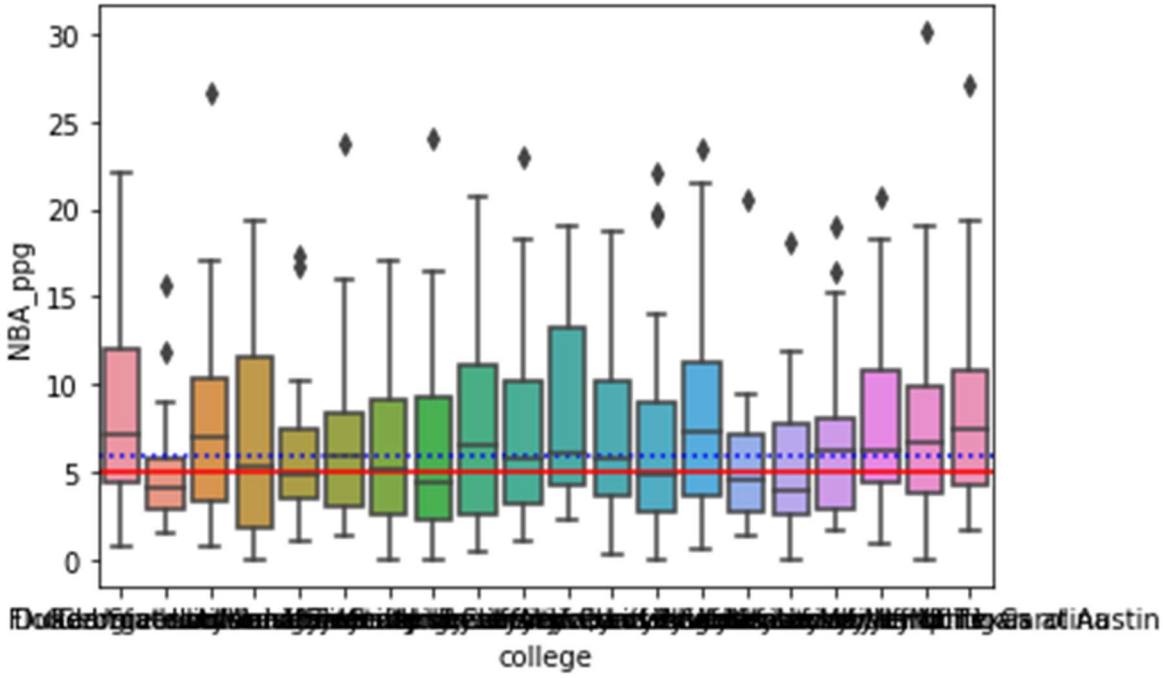

Figure 1. Seaborn Boxplot of NBA ppg for players coming from a given college, provided the college has produced at least twenty NBA players. Red line represents the median ppg of all NBA players in the current dataset; blue line represents the median ppg of solely NBA players from the colleges displayed in the graph.

[1] Although it is impossible to know the exact number of years a player will spend in the NBA when they first enter, it is fairly reasonable to include this in our model. For future prediction purposes, an NBA rookie would be asked for the amount of time they plan to spend in the league, and this number would become the input for the model.

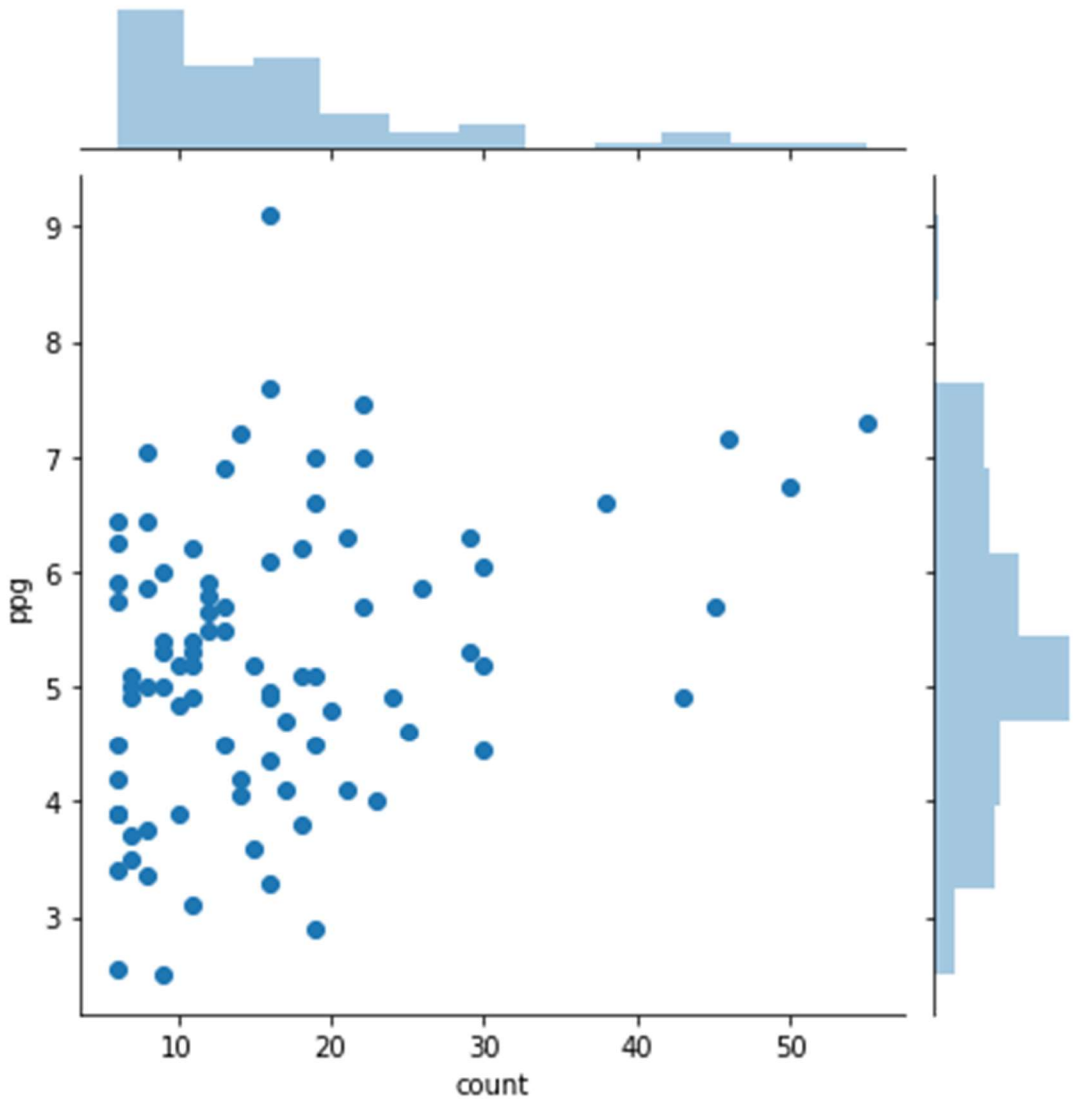

Figure 2. Seaborn Jointplot of median NBA ppg and count of NBA players for all colleges that produced at least five NBA players.

[1] Although it is impossible to know the exact number of years a player will spend in the NBA when they first enter, it is fairly reasonable to include this in our model. For future prediction purposes, an NBA rookie would be asked for the amount of time they plan to spend in the league, and this number would become the input for the model.

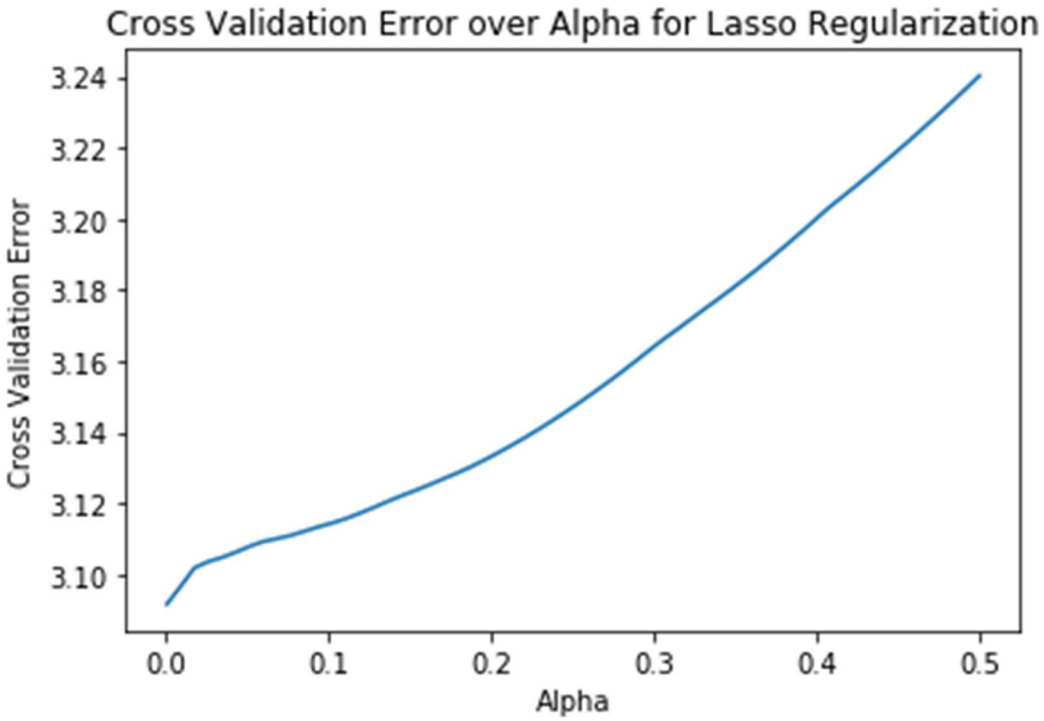

Figure 3. Cross validation error for varying hyperparameters (values of alpha, ranging from 0.001 to 0.5) in a lasso regularization model. The best alpha value is 0.001 with a cross validation error of 3.092.

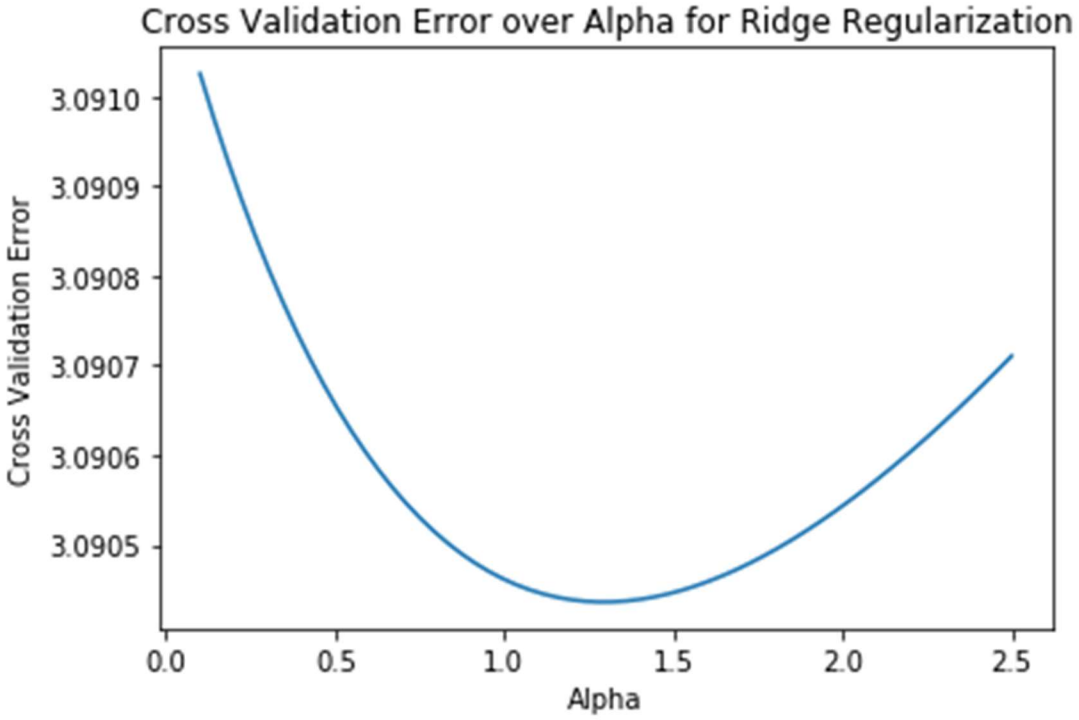

Figure 4. Cross validation error for varying hyperparameters (values of alpha, ranging from 0.1 to 2.5) in a ridge regularization model. The best alpha value is 1.280 with a cross validation error of 3.090.

[1] Although it is impossible to know the exact number of years a player will spend in the NBA when they first enter, it is fairly reasonable to include this in our model. For future prediction purposes, an NBA rookie would be asked for the amount of time they plan to spend in the league, and this number would become the input for the model.

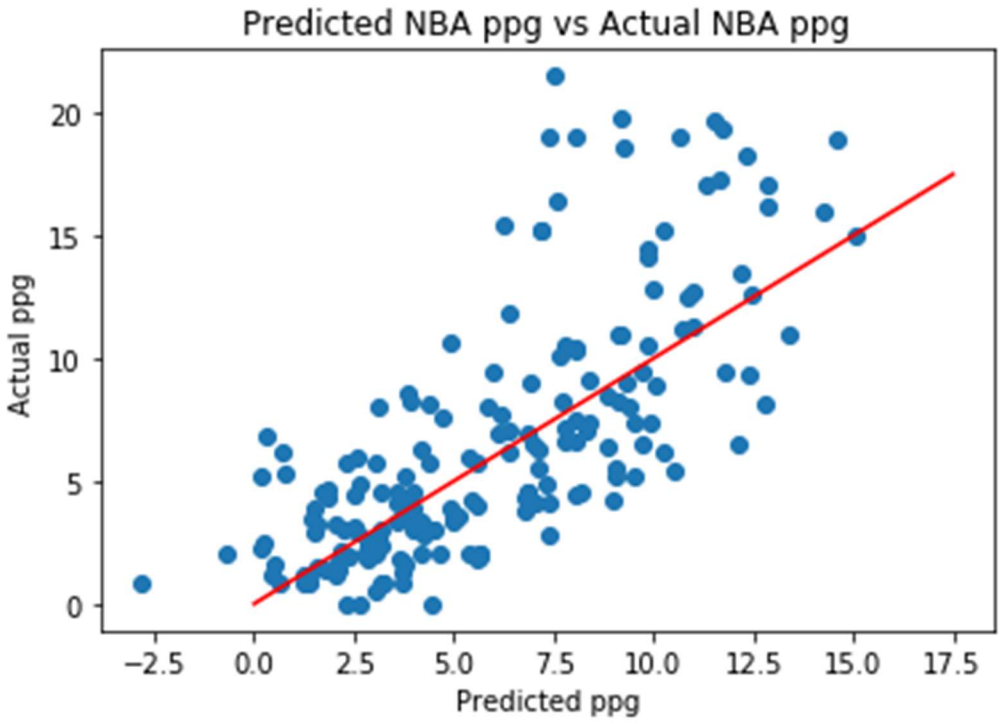

Figure 5. Scatter plot of predicted NBA ppg against actual NBA ppg for testing data (a randomly selected 10% of the overall dataset). These predictions have an RMSE of 3.523; red line represents perfect prediction.

[1] Although it is impossible to know the exact number of years a player will spend in the NBA when they first enter, it is fairly reasonable to include this in our model. For future prediction purposes, an NBA rookie would be asked for the amount of time they plan to spend in the league, and this number would become the input for the model.