**Big Time Ballers:**

**Predicting the 2017 All-NBA Roster**

**1 - Introduction:**

The NBA season can be a grueling stretch of 82 games (minimum) in a season, but it also allows the best players in the world the chance to display their skills and talents on the highest level. Success is typically valued on a team scale, but is also evaluated on an individual scale as well. Besides from winning the NBA MVP for the season, making the All-NBA team is considered to be the highest award for the best players in the league (based on personal performance, culminating the entire season). Along with championships, all-star appearances and MVP awards, this honor plays a critical role in personal accolades when being considered for the hall of fame when it is all said and done for that player. I stumbled on an article that breaks down each position by the numbers and assesses which players are considered the best playmakers in the league (<http://bleacherreport.com/articles/2690988-nba-metrics-101-the-best-playmakers-in-the-nba-according-to-the-numbers>).

**2 - The Problem:**

What caught my eye is that a majority of these players may be deemed as the “best playmaker” leading their position from a statistical standpoint, but that doesn't necessarily equate to making the All-NBA team. Wouldn’t it be great as a basketball fan to get a better sense of which players are truly playing at an elite level based solely on a cumulative score that accurately determines who is the best of the best? Ie) what the All-NBA roster *SHOULD* really be all about.

**3 - The Approach:**

These are the following topics which I covered:

1. Classification Models to determine what “classifies” a Player to be All-NBA status
2. Regression Analysis on the dataset
3. The various Cluster Analysis methods incorporated into this project.
4. Adding the innovative “Playmaker” and “Scorer” Rating into the mix.
5. Analyzing the accuracy of the Playmaker and Scorer Rating to previous All-NBA rosters.

**3.1 Initial Wrangling:**

The standard metrics such as: *Points, Rebounds, Assists and Player Efficiency Ratings*, although the predominantly conventional route to determine a player’s ranking, do carry some inherent inaccuracies in weighing the individual player’s overall performance from a quantitative standpoint. These tend to provide a one-dimensional detailed explanation, but to provide more volume and an in-depth look at player performance, further metrics need to be taken into account.

This revised format would help add additional accuracy to which players are the best at each position. Here are some of the factors (not in any particular order) that structure a player’s Scorer and Playmaker Rating:

|  |  |  |  |
| --- | --- | --- | --- |
| Games Played | Three’s Made/gm | Two’s Made/gm | Assisted Three’s% |
| Team FGA/gm | Team FTA/gm | Team Turnovers/gm | Team Points/gm |
| Assisted Two’s % | Free Throw% | Off. Reb % | Off. Team Rating |
| Team points/possession | Assisted Points/gm | Unassisted Points/gm | FG missed/gm |
| Pts/gm | Ast/gm | TO/gm | FTA/gm |
| FGA/gm | Usage |  |  |

At first glance this seemed like a daunting task, particularly with how/where I would be able to find this type of data, but after giving it some thought, and doing a bit of further research, I realized that I could calculate most of these variables by using basic player statistics, then use the amended variables to formulate my Scorer and Playmaker Ratings. Essentially, my plan was to take the basic player stats, use them to calculate the more complex variables, and take those variables that were derived and apply to the Scorer Rating formula.

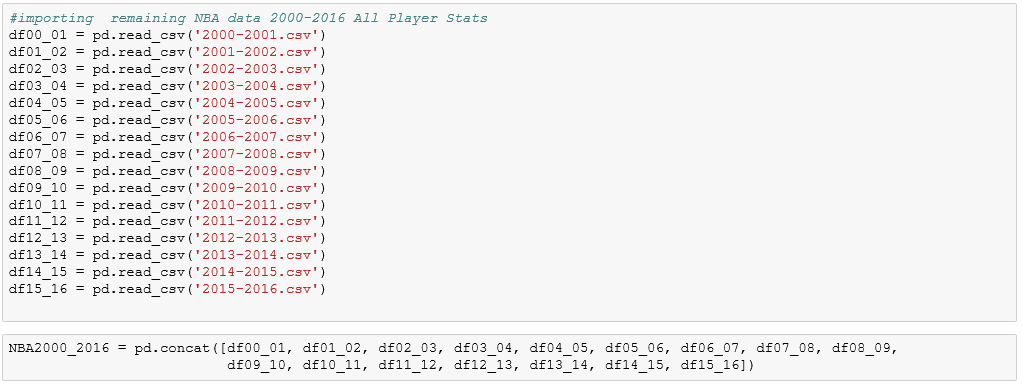
I was able to find all of my data online with the various popular NBA stats websites, but they didn’t consist of data put together on one table. My next plan was to scrape the player data from each API by using Pythons package “Beautiful Soup”. I was able to find various helpful YouTube links which provided the proper steps for web-scraping. I seemed to be making some progress in web scraping, but for one reason or another, eventually hit a roadblock with being able to complete this task. I brought this up to my mentor in case he would be able to provide some insight and troubleshoot this issue. He then brought up an interesting idea, perhaps I was overthinking this and if I had the ability to copy and paste the data on to a spreadsheet, then I should just give that a try, especially for time efficacy purposes. So I did this, and fortunately it worked. It took making 6 separate tables from three websites to painstakingly combine all player and team data together.

I took the combined initial player data and imported as a csv on python where I would then have my table and use the various python packages to evaluate the numbers and do any further data cleaning. I noticed that the outliers consisted of mostly unpopular players that I know should not necessarily have a particularly high rating. After looking a little closer, I realized that these players had a ridiculously high rating because they had not played in many games this season and when they did play, their game stats were based on very low averages with higher numbers. Had they theoretically played in as many games as a starter or role player, these outliers would have eventually regressed towards the mean. To solve this problem, I decided that to qualify for the rankings, players must have suited up in at least 65 games by the end of the regular season.

**3.2 - Dataset:**

For my first Springboard Capstone project, I collected all NBA player data from 2000-2016, roughly 7300 players with 89 variables. When the data was eventually organized I was able to delve into the more glamorous side of investigating the problem.

The original NBA data consists of 16 csv file and combined with a concat command:



**4 - Analysis:**

Because this required a way to choose between models, and find out if my data/sample size was feasible for making prediction models, I began with a few model evaluation methods. This would help answer any questions based on model types, tuning parameters and features. To start, I imported various Scikit\_learn packages into python, which would help proceed with developing training and testing data on my NBA players stats in hopes to reward overly complex models that “overfit” the training data would not give me a generalized assessment of this data. Additionally, to analyze the various metrics for precision to see how my model is shaping out.

To proceed with any further analysis, it would require me to convert this classification of All-NBA vs Non-All NBA players into a numerical binary model of 0’s and 1’s, respectively.

* Classification Accuracy

The dataset was split into two pieces so the model could eventually be trained and tested on different data. After tuning the model in making classification prediction for the testing set I reviewed the accuracy score to make sure that everything ran smoothly. Scikit-learn on python gave this model an accuracy score of .97 (97% accuracy), which is considered extremely high and effective. As for null accuracy scores for binary classification problems coded as 0/1, and null accuracy scores for multi-class classification problems, I received both scores of ~.93 (93% accuracy).

* Confusion Matrix

This will describe the performance of a classification model. After setting the first argument to be true values (y\_test) and the second argument to be predicted values (y\_predict\_class), I went ahead and saved the confusion matrix and sliced into four pieces: True Positive, True Negative, False Positive, False Negative (TP, TN, FP, FN). When I set the confusion matrix I wanted to then know the metrics that are computed from this classification to see overall how often is the classifier correct. The classification accuracy model received a score of ~.97 (97% accuracy)

* Classification error

I then want to know overall how often is the classifier incorrect, also known as “misclassification rate”. This received a score ~0.03 (3% inaccuracy)

* Remaining metrics computed from a confusion matrix

The remaining metrics that I wanted to compute for analysis consisted of Sensitivity, Specificity, False Positive Rate and Precision.

The role of a Sensitivity metric is to determine when the actual value is positive, how often will the prediction be correct? Albeit, how “sensitive” is the classifier to detecting positive instance, also known as True Positive Rate. The True Positive Rate received a score ~.99

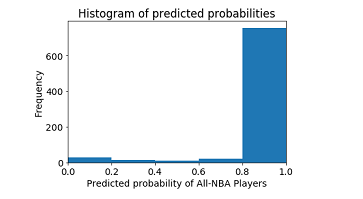
The specificity metric plays a role in situations when the actual value is determined to be negative and we want to know how often is the prediction correct? It is the determining factor of how “specific” is the classifier in predicting positive instances, receiving a score ~.71

False Positive Rate is described as when the actual value is negative, how often is the prediction incorrect. FPR prediction rate here came out to ~.29

The precision metric is calculation for how “precise” the classifier is when predicting positive instances. More specifically, when a positive value is predicted and we want to know how often is the prediction correct. Precision received a score ~.98

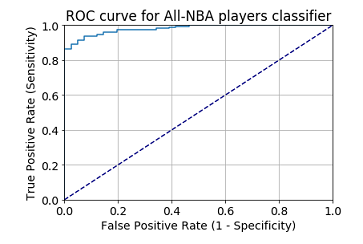
Overall, we can conclude from the aforementioned list of metrics above that the classifier for this model is performing at a relatively high level of accuracy.

For experimental purposes, I decided to modify the performance of this classifier by adjusting the classification threshold. Below is the distribution of the predicted probabilities of class 1.



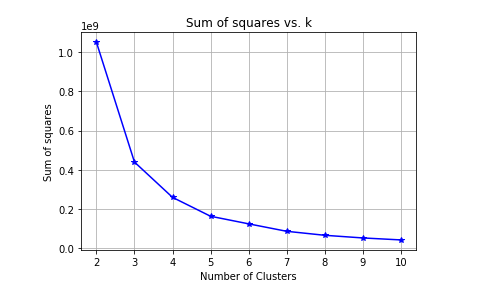
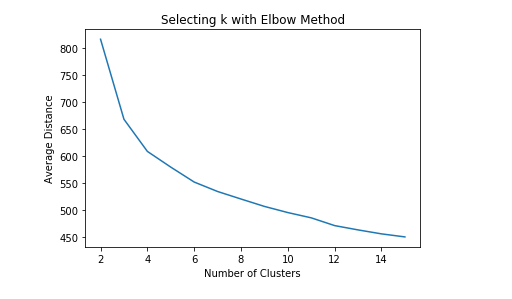
* Roc Curve and Area Under the Curve

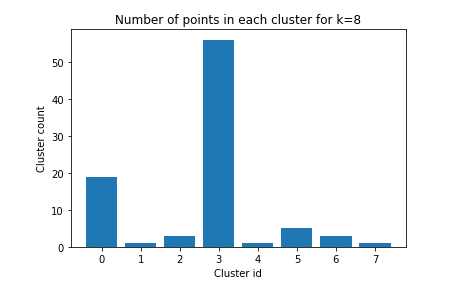
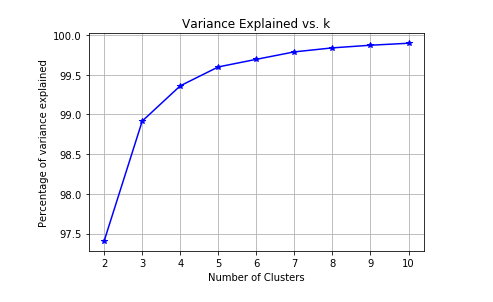
Finally, I decided to compile my data to perform an ROC curve which ultimately serves the purpose of helping to choose a threshold that balances sensitivity and specificity for this particular data. ROC curve was plotted as follows:



From these observations, we can note that an ideal classifier would be located in the upper left-hand corner of this plot, which is what this ROC plot above visually describes. An ideal classifier for the area under the ROC curve (AUC curve) illustrates that an ideal classifier is indicative of an overall better classifier. As such, AUC is used as a summary of the performance of a classifier as an alternative to classification accuracy. After calculating the AUC for this model, it received a score ~.98

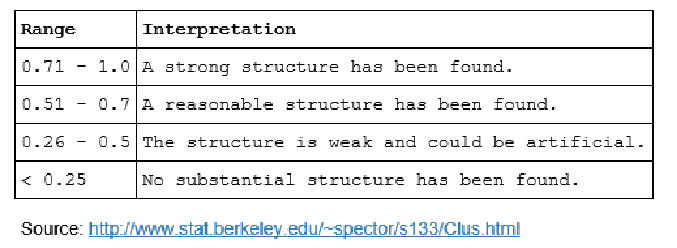
Now that we’ve covered evaluating the model and the classification threshold and accuracy, we can begin to start examining if there are any trends or patterns in the data collected. To start, I began with a few popular methods known as the Elbow Method, Silhouette Method and K-Means Clustering. All of which serve the purpose of visually determining cluster sizes within the accumulation of data points.



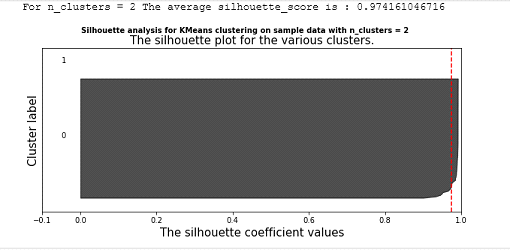
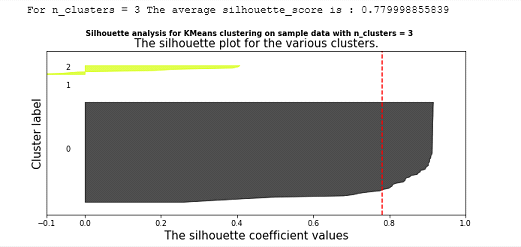


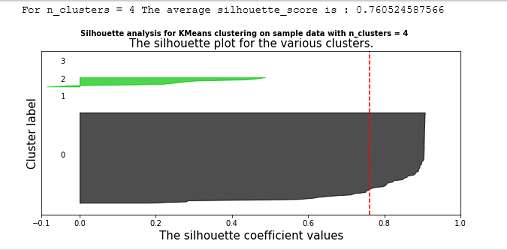
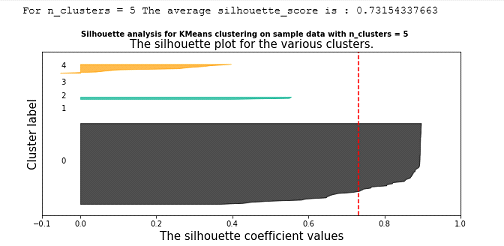
Based on the figures above, the graph begins to “elbow” and curve where K=3

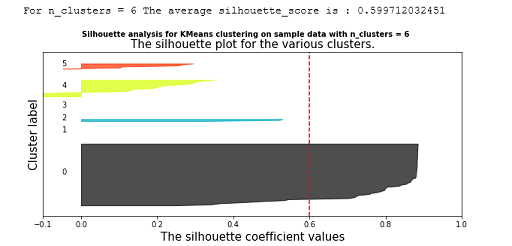
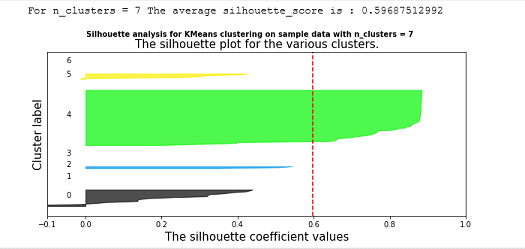
After getting an idea of the ideal cluster variance through the computed methods noted above, I chose to analyze the Silhouette Method which computes every data point on each cluster by averaging the distance to all other points between each cluster. Silhouette Scores, range from -1 to +1 and the table below summarizes the range of scores.

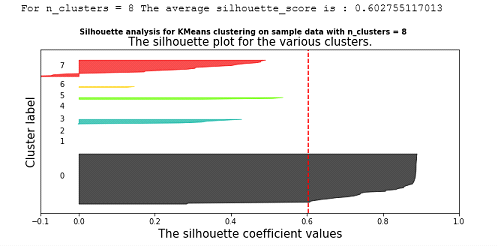
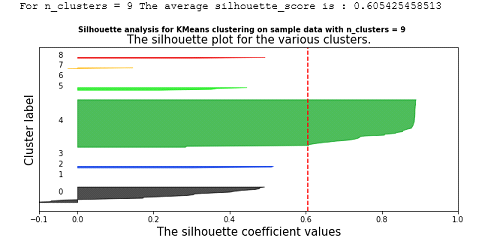


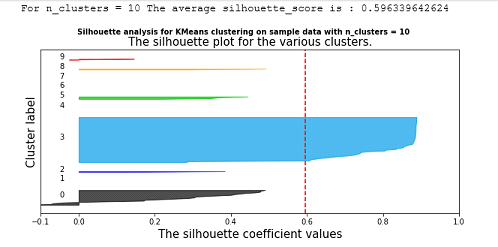
With Scikit-Learn I constructed a series of silhouette plots for the various clusters:

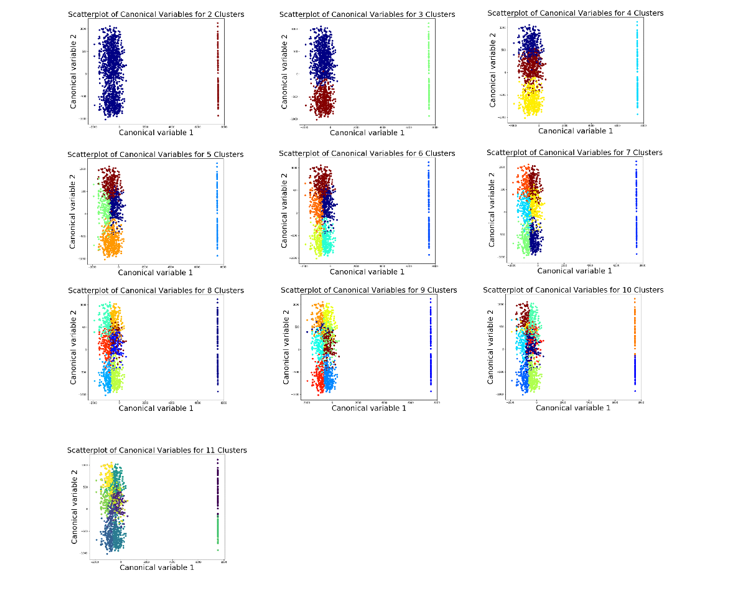
 



Based on the graphs above, I would use 2 as the ideal K Means Clustering.

Lastly, after observing the various methods which constructed a silhouette of every data point in every cluster and how its distance plays a role in the structure. The final method I wanted to construct involved taking the actual scatter plot data and computing a Clustering method called, Principal Component Analysis (PCA). PCA comes in handy when trying to visually reduce a large set of variables into smaller components. After computing clusters from 2 – 11, the scatterplot displayed a visualization as follows:



From a visual standpoint, there seems to be the clearest amount of grouping points separating data points for n\_cluster values of 6 and 7.

**4.1 Analysis of Method 1: Adding a Playmaker Rating**

I made a new variable called the “playmaker rating” which serves as a conduit in ranking the best players at each position with better precision. The typical basic in game stats that broadcasters, analysts and the associated press tend to focus on are of a player’s points per game, rebounds per game, steals and assists. This is not the case here. The Playmaker Rating takes an in depth look at each player’s performance in respect to individual weighted performance and assesses most, if not all relevant player stats recorded.

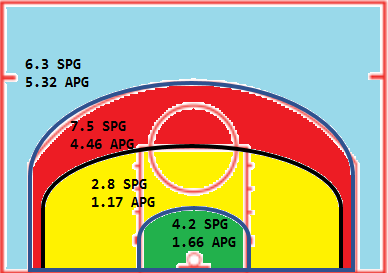
Here is what the Playmaker Rating formula looks like:



Where USG = Usage Rate Per Game, PPG = Points Per Game, APG = Assists Per Game, TPG = Turnovers Per Game, FGA = Field Goal Attempts Per Game, FTA = Free Throw Attempts Per Game[[1]](#footnote-1)

So, why the 2.26 in front of assists per game? It mostly has to do with the fact that not every assist is worth exactly two points of offense. Many assists lead to made three-pointers by teammates. To find out exactly how often this was the case, I turned to [HoopData.com](http://hoopdata.com/), a site that breaks down where on-court field goals are made and which ones are assisted.

During the first half of the 2015-2016 season, 15 shots per game have been made at the rim, and 52.4 percent were directly due to assists, demonstrating that 7.86 assists per game are generated by shots made at the rim. From 3 - 9 feet away, there are 4.2 made shots per game, and 39.6 percent of them are assisted, producing another 1.66 assists per game from this area. From 10 – 15 feet away from the hoop, there are 2.8 makes per game, and 41.9 percent of them are assisted: another 1.17 assists per game. From 16 - 23 feet, there are 7.5 makes, and 59.5 percent of them are assisted: 4.46 more assists per game. Subsequently, from behind the three-point arc, there are 6.3 makes per game, and 84.6 percent of them are the result of assists from teammates. That means that 5.32 assists per game lead to three points instead of two.[[2]](#footnote-2)



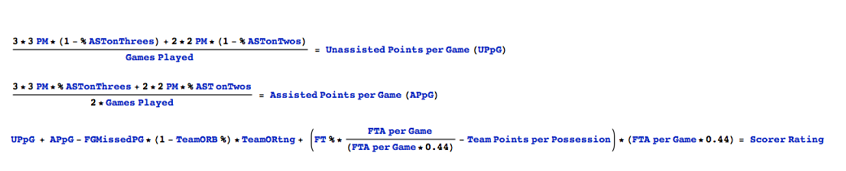
Adding it all up, there are 20.47 assists per game (7.86 + 1.66 + 1.17 + 4.46 + 5.32) by the average team in the NBA. 15.15 of them result in two-point shots, leading to 30.3 points per game. The remaining 5.32 come on three-pointers and thus lead to 15.96 points per game. Adding those two numbers up, we see that those 20.47 assists per game lead, on average, to 46.26 points per game. Simple division therefore tells us that each assist is worth ~2.26 points.[[3]](#footnote-3)

**4.2 - Analysis of Method 2: Adding a Scorer Rating**

The proverbial NBA saying goes a little something like, “Offense will win you games, defense will win you championships. Luckily here, I wasn’t trying to make a predictive model to assess which team will win an NBA championship. This project was all about the individual and not the team. Therefore, being an elite scorer will be advantageous when selecting a player to making the All-NBA team. Considering there 15 roster spots out of the entire league.

To get a better sense of a player’s offensive abilities I formulated another variable, which does not typically get used by sports broadcasters and analysts, called the “Scorer Rating”. Similarly, to the Playmaker Rating, it serves with exactness to get a better understanding by ranking each player. Rather in this scenario, the Scorer Rating has an emphasis on, you guessed it, scoring.

Here is what the Scorer Rating formula look like:

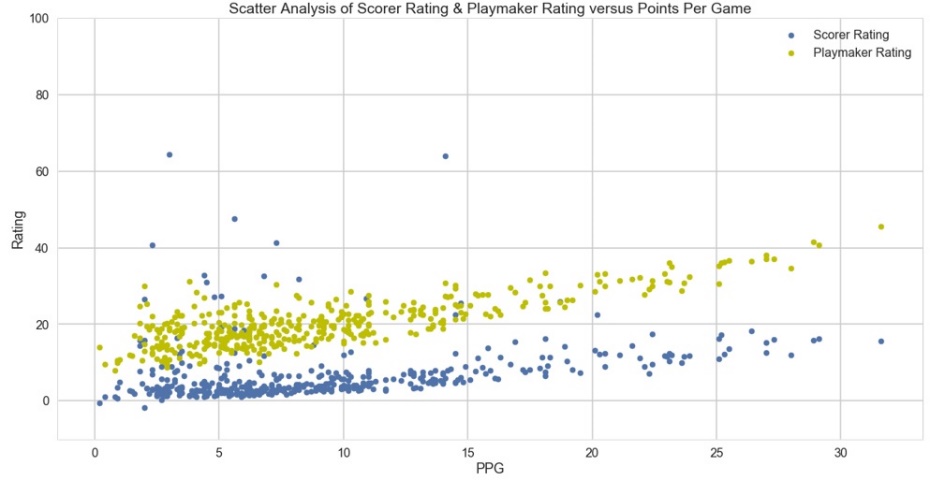
****

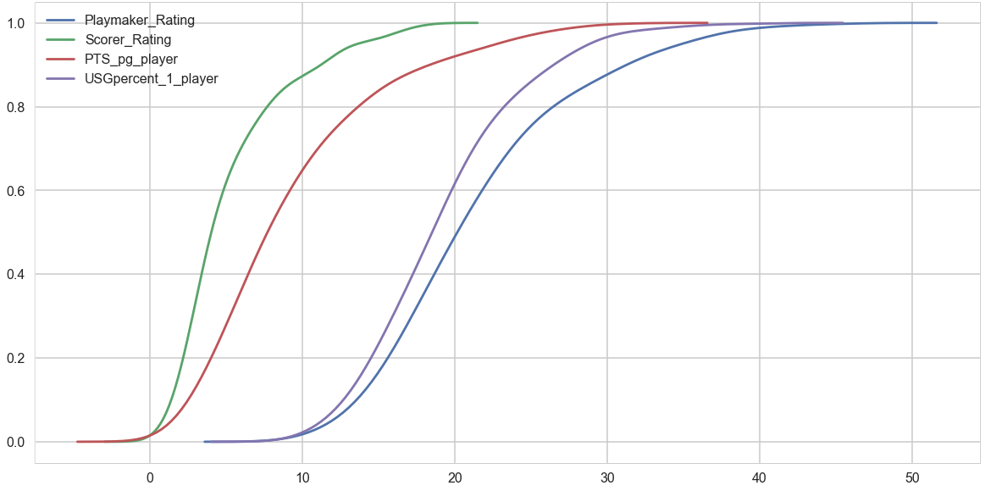
Where PM = Points Made, %ASTonThrees = Assist % on Threes, %ASTonTwos = Assist % on Twos, FGMissedPG = Field Goals Missed Per Game, TeamORtng = Team Offensive Rating, FTA = Free Throw Attempts Per Game, [[4]](#footnote-4)

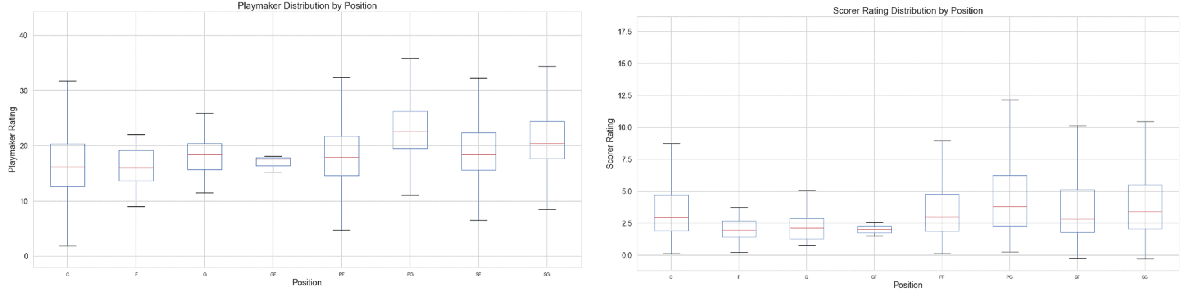
The number 0.44 in this equation stems from the fact that it is the consensus true shooting percentage, as a constant in the NBA.

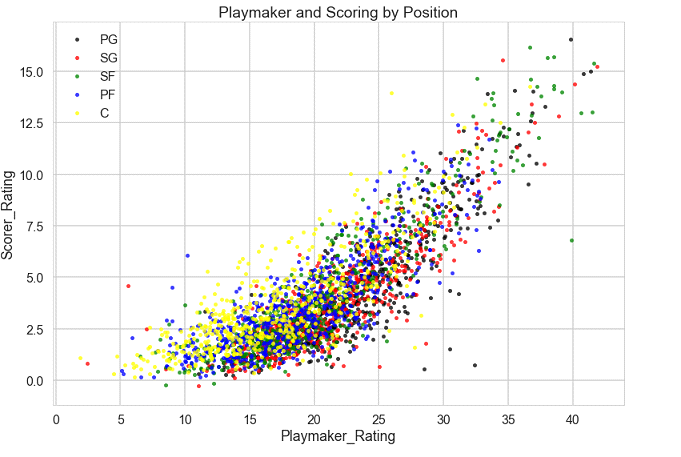
**4.3 - Comparing the 2 Ratings:**

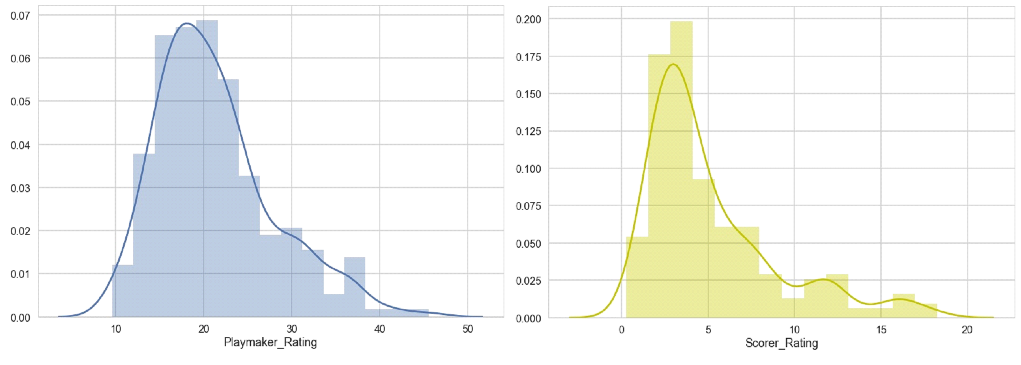
After creating the PR and SR variables, I wanted to see if there were any notable trends within the data points. The NBA player data in association with Playmaker and Scorer Ratings were as follows:

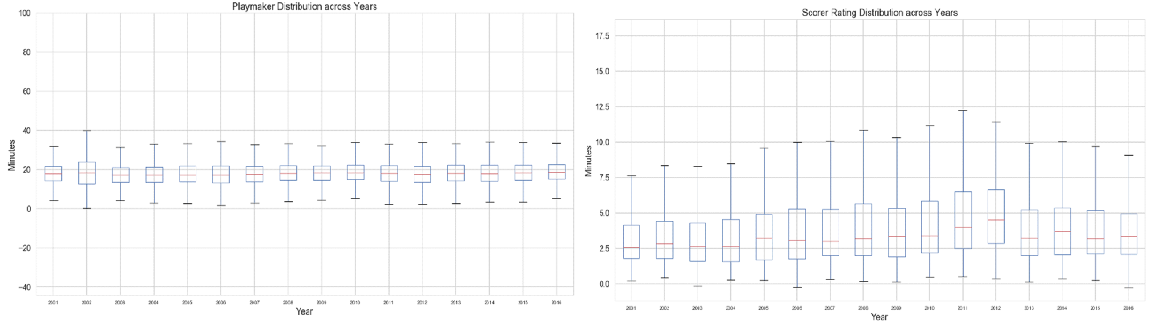


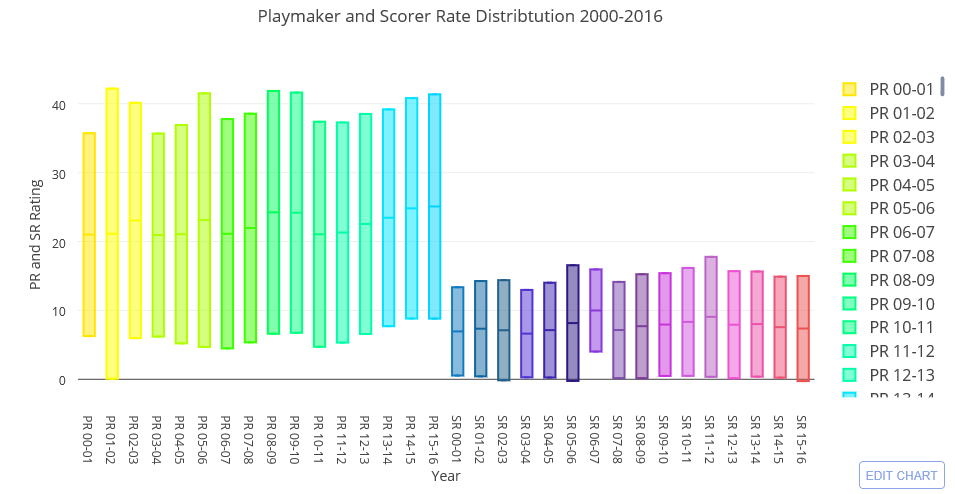


****





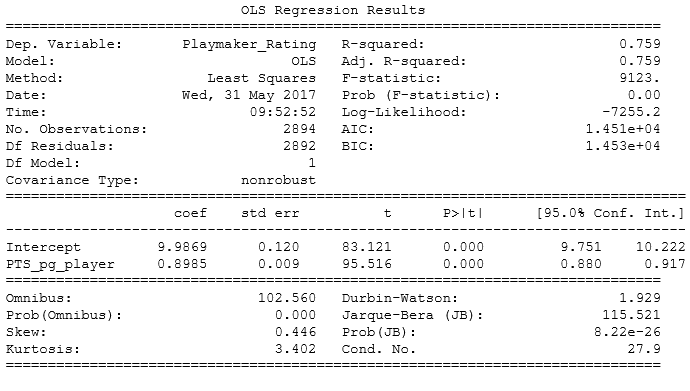
****



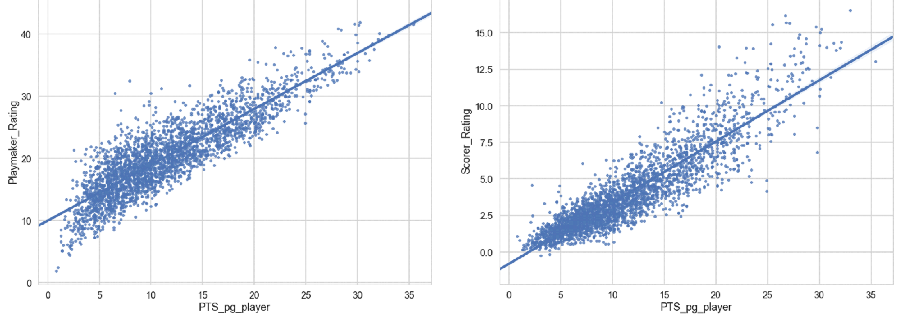
* Linear Regression

I may have all of these variables collected, but what good are they if I don’t interpret the relationships among coefficients and then design predictive models on continuous outcomes for my NBA player data.

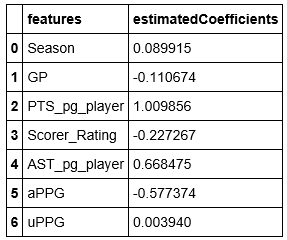
To commence, I imported a regression package called Ordinary Least Squares (OLS) which would display some of the basic Inferential Statistics for me. By using the Least Squares Method, it will help display the metrics of whether or not there could be a linear relationship between Points Per Game and PR Ratings. We can interpret the PPG coefficient (0.8985) that the p-Value (P > |t|) is so small that I should be more than okay with proceeding to fit this regression model.



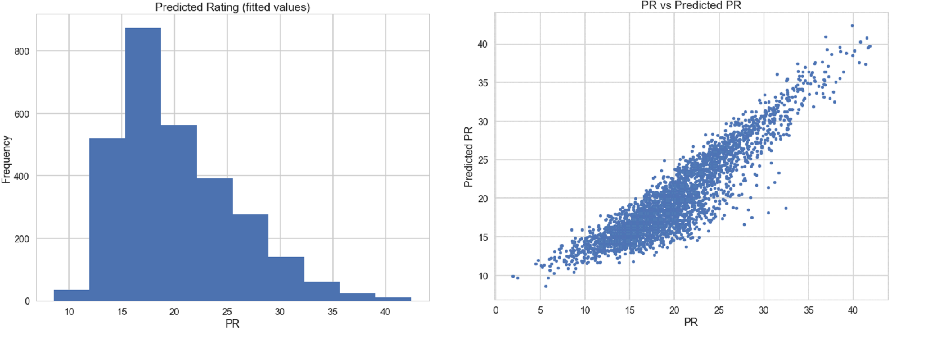
Next, I wanted to see how my model actual fits my data. I created a few scatter plots displaying the correlation between variables and then was able to make some interesting predictions with visual analysis. See figures below:



I then analyzed the linear model by building a table using pandas to see what the actual numerical estimates were of the coefficients. Here are the numerical features which I was able to display:

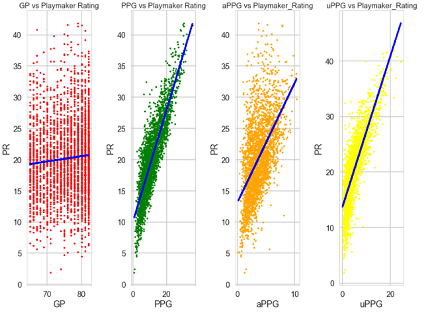


After gathering the estimated coefficients, I then plotted a histogram of a predicted ratings distribution of all NBA Players and made a scatter plot to observe the Actual Playmaker Ratings versus the Predicted Playmaker Rating. See figures below:



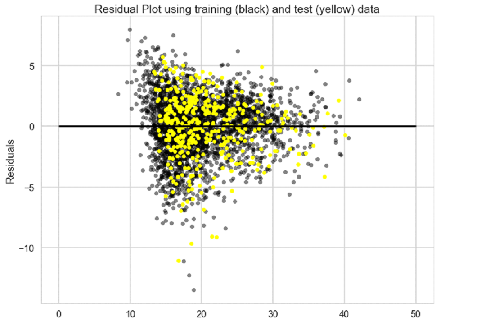
I then decided to fit a linear model using a few other independent variables. For this example, I chose to analyze the correlation between Games Played, Points Per Game, Assisted Points Per Game, Unassisted Points Per Game and how there is any correlation to the Playmaker Rating.

Here is what I was able to observe:



We can see from the figures above that there is a strong correlation between all variables and how they relate to PR and SR.

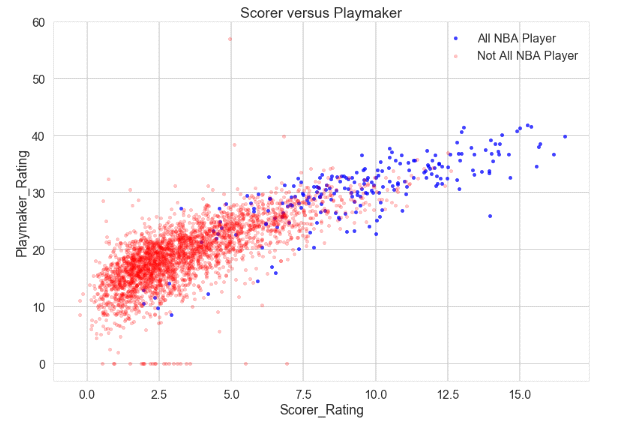
Lastly, to detect if the model fit the data well and any new data, I decided to split the data into random training and testing sets, build a linear regression model using this new training data and build a residual scatter plot of predicting the output on the test set. See figure below:



* Logistic Regression

Since I was able to get a good sense of predicting continuous outcomes for the various independent variables, it was now time to explore classification within this model, specifically, which of a small set of classes I can make observations about what it is that makes a player qualify to be an All-NBA Player. This method entails Logistic Regression.

In order to do so, I had to setup the data points in a binary classification. This entailed creating a variable for Players that were and were not All-NBA players, 0 = Yes, 1 = No. I then created a scatterplot to observe the distribution of All-NBA and Non-All NBA and their correlation to Playmaker and Scorer Ratings. Would the higher the Playmaker and Scorer Rating equate to that player being an All-NBA player? The answers were as follows:



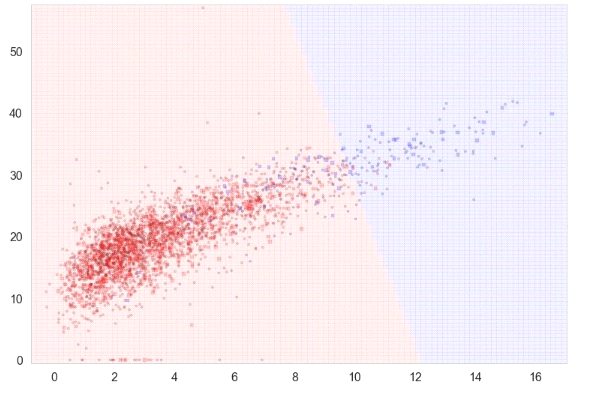
We can see that there was a tendency for All-NBA players to have the higher ratings among the distribution of all Players.

Next, I wanted to split the data into a train and test model, to evaluate the performance in fitting any unseen or new data and make any new predictions. With tuning the model for logistic regression and account for any hyperparameters, I used a mix of cross-validation and grid search. I then tuned the parameter with a method called regularization parameter, C, which is used to control for any unlikely high regression coefficients.

With Scikit\_Learn, I applied a cv\_score function to perform K-fold cross validation, and this score quantifies the accuracy of the model. The cv score received a valued of ~.95

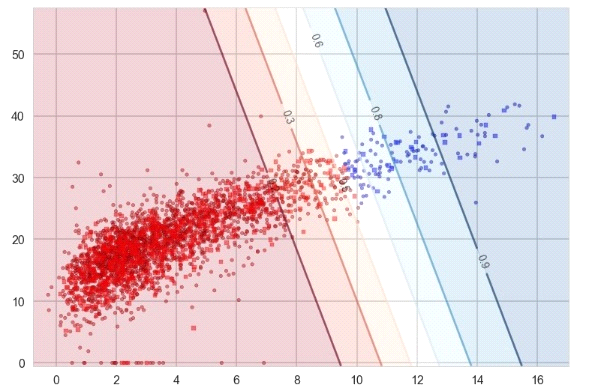
Next, I performed a grid search on the parameters. The best C values was 10, with a Maximum Score of ~.953. I took the best C that was obtained from the procedure and trained a Logistic Regression on the training data. The accuracy on the test data received a score of 94.9%.

The graph below computes the results of the logistic regression model. We can now see a line in feature space that divides the classification boundary between 0’s and 1’s which are Blue and Red, respectively.



I plotted the actual labels of both training (circles) and test (squares) samples. Most of the samples appear to be classified well, but if you look closely, there remain to be some misclassified players on both sides, as evidence of leakage of dots or squares of one color on to the side of the other color.

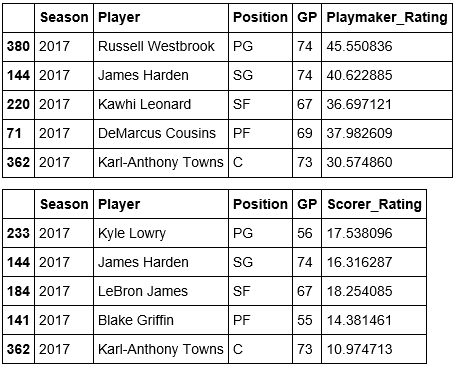
One way to resolve this issue would be to integrate a probabilistic interpretation into the graph. Training the model to have an intuitive notion that identifies the probability between boundary lines and the distribution in classifiers will certainly come in handy. After plotting the probabilities obtained, the graph came out as follows:



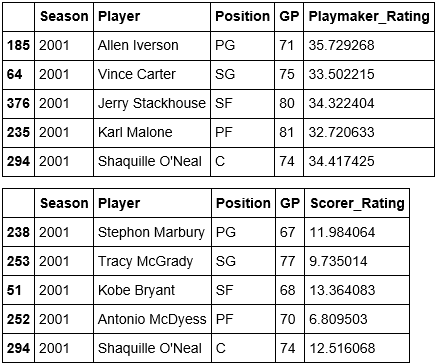
In the diagram above, I have plotted predicted values rather than the actual label of samples. Everything marked on the left of 0.5 illustrates the probability of a Non-All NBA Player and to the Right of 0.5 marks the probability of an All-NBA player. The line that creates a decision boundary is known as the discriminative classifier.

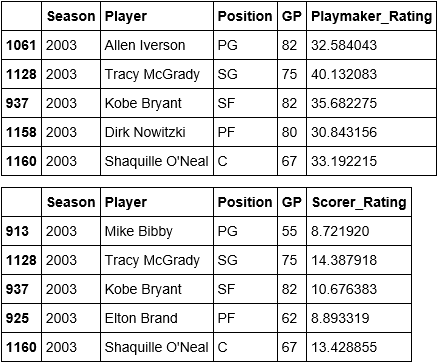
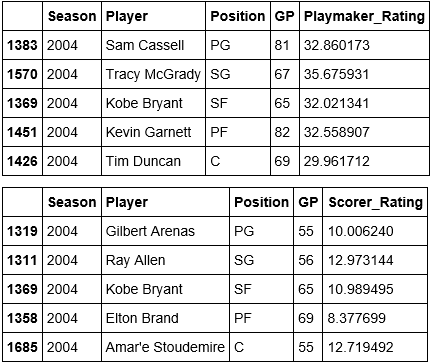
After familiarizing myself with analyzing the various trends and visualizations of which particular classifiers make an All-NBA Player, it was time to dive into the actual predictions of who will make the roster, based on PR and SR.

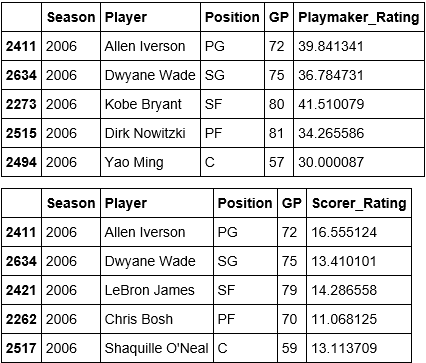
I filtered the results for the 2017 season, sorted by top Playmaker and Scorer Ratings at each position and these players were representative of the best of the best. You can see that some players overlap in both ratings, but based on the following, here are my predictions for the 2017 All-NBA Roster:

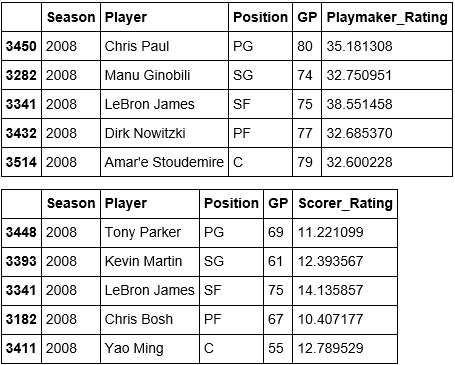


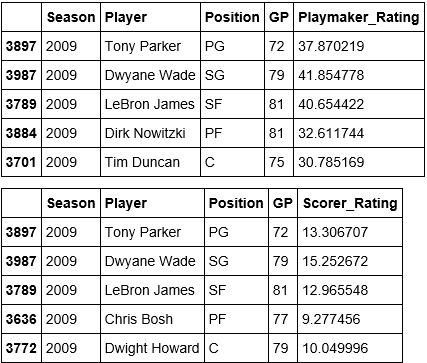
After diving into my predictions for the 2017 All-NBA roster selections, I wanted to evaluate previous seasons by using the Playmaker and Scorer Ratings and then assess if they were consistent with actual All-NBA selections. Firstly, I created tables of the top players from 2000-2016 seasons based on Playmaker and Scorer Ratings. The rosters consisted of the following players:

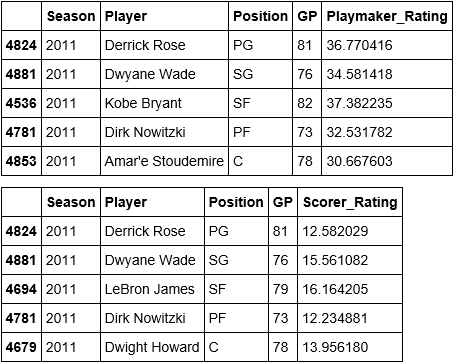
 

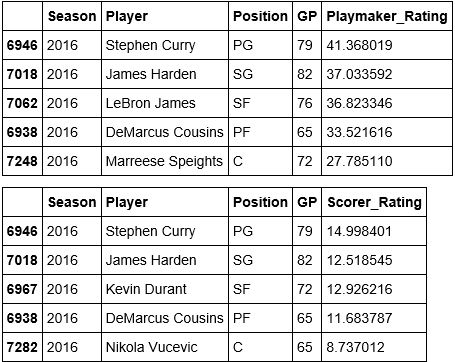
 

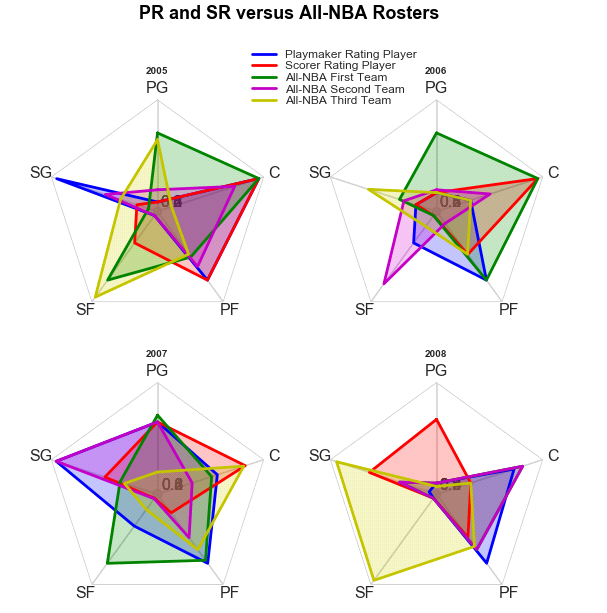
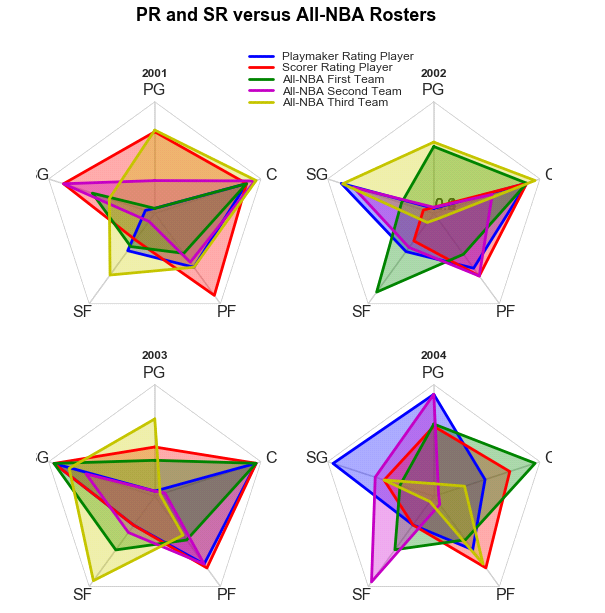
 

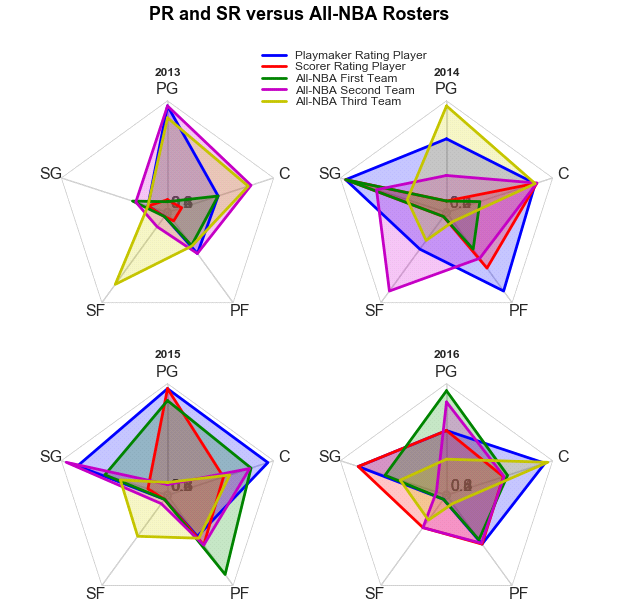
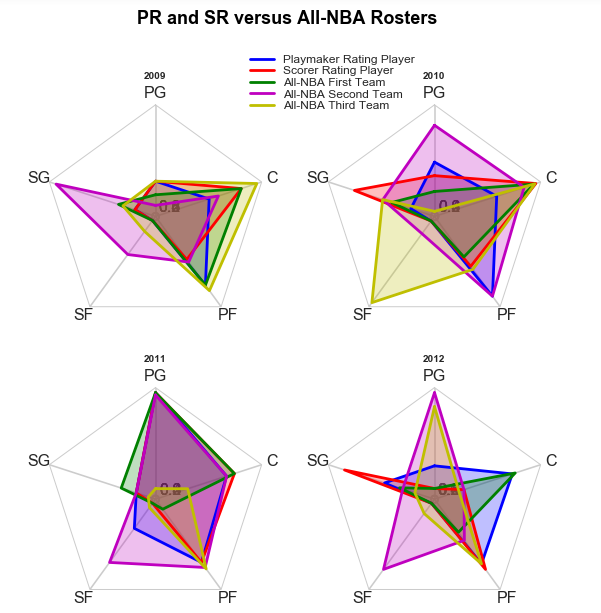
 

Now that we have all theoretical predictions for All-NBA based solely on Playmaker and Scorer Ratings. I wanted to see how they related to player data from the past 16 years of actual All-NBA teams. The figures below describe All-NBA First, Second and Third Teams and how they overlap with my predictions based on Playmaker and Scorer Ratings:





Based on the radar plots above, we can see that across seasons 2000-2016 there are many instances where my predictions of NBA players being selected to make the All-NBA team based on top Playmaker and Scorer Ratings overlap with actual players that were selected to All-NBA first, second, and third teams.

**5 - Recommended use of New Rating:**

Now that I have shed some light on the possibilities of a new rating system, the next question should be where do we go from here? How would this be implemented into the NBA? Here are a few suggestions:

1. Instead of focusing on the basic individual in-game stat line of ppg, apg, rpg, etc… the various websites that track real-time stats of games would highlight these ratings instead.
2. This could come of value when assessing which player’s one would decide to select when playing on the various web based fantasy sports betting games, like DraftKings or FanDuel.
3. The Associated Press use these metrics when voting for which players make the All-NBA roster, which would then provide greater credibility behind their logic, and erase any biased or preconceived notions hey may have had regarding a particular player.

**6 – Advantages/Disadvantages of New Rating:**

Because the Scorer Rating and Playmaker do provide clarity as a useful metric when trying to decide which players are the best of the best at each position, I believe these types of variables should play as a contributing factor when making the ultimate decisions. However, because the “playmaker” is philosophically more of an all-around evaluation on both sides of the ball (offensive and defensive abilities) and the “scorer” is more about dimensional analysis of a player’s offense, this can create some confusion when advocating solely on metrics whether or not one player should be selected versus another player that is on the cusp of making the All-NBA team, ie) is he a great all-around basketball player, or is he just very good with being a ball hog and scoring points for a team that may or may not be mediocre. All I am saying here is that in a certain context, the Playmaker Rating may have more of a weighted value than the Scorer Rating, and vice versa.

**7 - Potential Next Steps:**

I do think there is a need that for further accuracy to create some sort of a hybrid between both ratings that highlight the best player as a “playmaker” and as a “scorer” combined into a single rating system, possibly something I will explore in the near future.

1. Fromal, A. (2017, February 6). NBA Metrics 101: The Best Playmakers in the NBA, According to the Numbers. Retrieved from http://bleacherreport.com/articles/2690988-nba-metrics-101-the-best-playmakers-in-the-nba-according-to-the-numbers [↑](#footnote-ref-1)
2. Haley, J. (2011, October 11). An Introduction to Advanced Basketball Statistics: Understanding Possession Estimation and the Factors that Control Effeciency. Retrieved from http:// https://www.burntorangenation.com [↑](#footnote-ref-2)
3. Haley, J. (2011, October 11). An Introduction to Advanced Basketball Statistics: Understanding Possession Estimation and the Factors that Control Effeciency. Retrieved from http:// https://www.burntorangenation.com [↑](#footnote-ref-3)
4. Scaletta, K. (2014, September 19). Why Some NBA Points Mean More Than thers. Retrieved from http://bleacherreport.com/articles/2203628-why-some-nba-points-mean-more-than-others [↑](#footnote-ref-4)