Forecasting Fantasy Basketball Player Value for Season Long Leagues

Daniel Biales

December 10, 2019

Abstract

Fantasy sports have been growing in popularity ever since Dan Okrent started the first fantasy baseball league, in 1980 [1]. "In fantasy sports, the fans pose as both general manager and field manager of their team, building a roster through a draft and trades and making lineups in pursuit of the greatest statistical production [1]". This paper aims to examine how machine learning can be applied to NBA season data, in order to predict a player's average fantasy output per game, for the next year.

Introduction

In 2019 the market size of fantasy sports was $13.9 billion and expected to grow to $26.4 billion by 2024 [2]. Fantasy sports participants require accurate player performance predictions in order to make decisions about their fantasy teams. Whoever has the best predictions will often win their league.

Since 2013, when the NBA installed player-tracking cameras in every arena, there has been a wealth of data about players and many different ways to measure performance [3]. Therefore, this area is a prime candidate to apply machine learning. The goal of the model is to predict a player's average fantasy points per game over the course of an upcoming season. This would be useful data for fantasy managers to have at the beginning of the season when drafting their fantasy team.

The rest of the paper is organized as follows. I start with a discussion of previous work and how that shaped my investigation into the topic. Then I cover how the data was collected and cleaned. Next, I describe my baseline model. After, I detail the procedure for improving the model, including the results of several interesting experiments. Next, I summarize the final results and finally, I provide a reflection on my exploration of the topic.

Related Work

Due to the direct correlation between data and performance, and the financial incentives in sports gambling and daily fantasy sports, there is a wealth of research related to sports predictions. However, much of the research is not focused on season-long predictions. It primarily focuses on predicting game outcomes, as that is important to bettors and professional team management, and predicting single-game player performance, as this is pertinent to the daily fantasy sports industry. Additionally, much of the research is the proprietary information of companies profiting from fantasy sports, such as Draftkings player valuation model [5]. Despite these obstacles, there were a number of papers that examined player predictions with the goal of beating the proprietary algorithms, in order to make money. Even though the research specifics are not the same as my project the ideas behind these daily predictions are very similar.

The prior research into daily fantasy predictions was helpful for choosing a set of machine learning models to evaluate. Linear models such as support vector machines and linear regression were the most common models, used in studies by Hermann and Ntoso [5], Lutz[6], Harrison [7], and Shivakumar [8]. Additionally, Steenkiste [9] and Shivakumar [8] assessed the performance of random forests. I used this information when assessing which types of models to compare when building the baseline model.

I also researched findings in other fantasy sports, such as football, since certain player prediction insights have applications across sports. In Lutz's paper on predicting player's fantasy football scores, he refined his problem space specifically to quarterbacks as his research showed that, "modeling positions separately improves the accuracy of a model [6]". The position-specific modeling informed my decision to use position based multilevel modeling in order to add position informed data to my feature space.

Data Collection

A cursory glance at stat.nba.com or basketball-reference.com will give you an idea of the many different basketball statistics available and different ways to slice the data. In order to gather the necessary data, I wrote NodeJS scripts to collect data from stats.nba.com using the nba (https://www.npmjs.com/package/nba) library written by Nick Bottomley. My scripts are stored on GitHub (https://github.com/bialesdaniel/ml-fantasy-prediction).

I gathered data on all NBA players starting with the 2010-2011 season and ending with the 2017-2018 season. The 2018-2019 season was captured but only to calculate the actual outcome for the instances from the 2017-18 season and the 2009-2010 season was captured for the purpose of providing previous season data for the 2010-2011 instances, but player instances were not used from either of these seasons. This amounted to 3893 player instances. I randomly divided these instances into three data sets. The cross-validation set of 2713 instances was used to train models and test the significance of different models, feature sets, and parameter tuning results. The development set of 775 instances was used for error analysis and was closely examined to understand the problem space. Finally, the final set of 387 instances was used to test the final model against new data.

For each player, I collected season-specific data and player bio data. For season data I collected gp, min, fgm, fga, ftm, fta, reb, ast, stl, blk, tov, pts, fG3M, fG3A, oreb, dreb, blka, pf, pfd, plusMinus, dD2, tD3, fgPct, fg3Pct, ftPct, wPct, and fppg (see glossary) for various time periods. Those statistics were gathered for the per game values of the current season, the per game values for the current season post-all-star break, the difference of the per game values pre and post all-star break, the total values for the previous season, the total values for the current season, the difference of the current season's total values and the previous season's total values, and the per-minute values for the current season. For bio data I collected the player's age, team, height, weight, draft number, position, the "current" season the statistics were collected for, years in the NBA, and college.

The number my model aimed to predict was the average fantasy points per game of a player, for the next season. A player's fantasy points on a given night are calculated based on certain common statistics. Their average fantasy points are the average of all their fantasy points and the games that they played in, during a season. In order to calculate fantasy points, I used a common scoring system used by ESPN for their standard points leagues. The score is calculated by aggregating the following stats based on their point value: fg = 1, fga = -1, ft = 1, fta = -1, reb =1, ast =1, stl =1, tov = -1, pts=1. A fppg (fantasy points per game) value was assigned to each instance in order to assess the accuracy of the model.

In order to normalize the data, I normalized all the numeric features with the following formula: $normalized = (value - min)/(max - min)$ and converted the nominal features into binary features. The features and instances were then converted from JSON format to an ARFF file using the NodeJS arff library (https://www.npmjs.com/package/arff) created by Julien Fontanet.

	Linear Regression	Support Vector Machine	Random Forest
Correlation Coefficient	0.8399	0.8397	0.8215
RMSE	3.7208	3.7318	3.9124

Figure 1: baseline model comparison

The difficulty of this problem space is choosing the most relevant features to train the model on since there are nearly endless options and combinations. I chose the most basic features to train the baseline model, which is explained in the next section. Then I chose different sets of the other features to run experiments on, which is explained in the model improvements section.

Baseline model

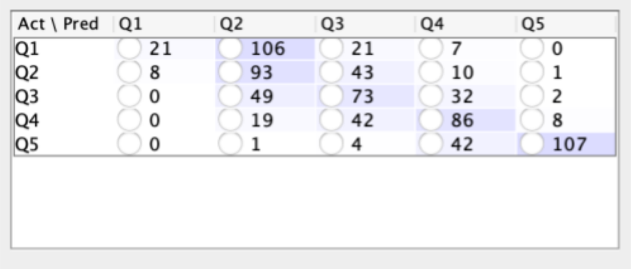
In order to train a baseline model, I chose a simple feature space based on the calculation of fantasy points per game. The features I chose were the player's per-game fg, fga, fta, ast, stl, tov, and pts. I also added the player's age, based on the concept of a player's prime years[10], games played and minutes, because this speaks to the player's experience, and the season the data was gathered for.

Using these features and the cross-validation set, I assessed three different models: linear regression, support vector machines (SMOreg), and random forests using Weka's basic configurations for each model. The results of this ten-fold cross validation experiment are listed in figure 1.

The linear regression model had the best correlation coefficient and root mean squared error, so that is the model that I used for further feature refinement.

Model Improvements

In order to improve upon the baseline model, I used my development set to perform an error analysis on the model. The largest source of error was differentiating between very low fppg values, as evident in figure 2. For example, predicting 4.1 when the real value was 0.0. I ignored this type of error since players with average fantasy points as low as that are not good enough to be rostered in most fantasy leagues. Instead, I focused on instances that were predicted to be in Q4 but were actually in Q5. This means those players were predicted to have a lower fppg than they actually had.



Act \ Pred	Q1	Q2	Q3	Q4	Q5
Q1	○ 21	○ 106	○ 21	○ 7	○ 0
Q2	○ 8	○ 93	○ 43	○ 10	○ 1
Q3	○ 0	○ 49	○ 73	○ 32	○ 2
Q4	○ 0	○ 19	○ 42	○ 86	○ 8
Q5	○ 0	○ 1	○ 4	○ 42	○ 107

Figure 2: baseline confusion matrix in lightside

Upon analysis, I noticed that two problematic features were pts and fga. The instances that were correctly predicted in Q5 had much higher points and field goals attempts than the instances that were incorrectly predicted in Q4. Some of the players that were misclassified were Nikola Vucevic, Al Horford, and Paul Millsap. One thing they have in common is that they have a high field goal percentage so they can score the same amount with fewer shots. They are also centers or power forwards. Therefore, they make up for their lack of scoring with rebounds and blocks. I started to consider that the weight my model was putting on some statistics was generalizing the weight across all positions and not factoring in the relevance of that statistic to specific positions. This hypothesis also motivated by Lutz's work on fantasy football predictions, where he only modeled quarterback data, because he found that combining the positions confused the model.

I added position data to my data set in order to perform multilevel modeling. Instead of using the eight positions (guard, guard-forward, forward-guard, forward, forward-center, center-forward, center, unknown) given by nba.com, I transformed the data into four positions that are based on the positionless style of basketball currently played in the NBA[11]. As Brad Stevens, the coach of the Boston Celtics, said, “It may be as simple as three positions now, where you're either a ball-handler, a wing or a big [12]“. Therefore, I categorized players as guard, wing, big, or unknown.

To understand which statistics might be affected by position I examined scatter plots of my instances where the x-axis was the feature I was examining, the y-axis was fppg and the color of the point was based on player position. As visible in Figure 3, there was clear separation, by position, among some of the statistics, especially between guards and bigs.

With this information in mind, I used LightSide to create a multilevel model based on position for the ast, stl, pts, reb, and tov features. I also added fgPct and ftPct to account for the errors with fga. Then I retrained the model over my cross-validation set. The resulting model showed insignificant improvement over the baseline with a p-value= 0.981. The correlation coefficient was 0.8428 and the root mean squared error was 3.692. Despite the lack of statistically significant improvement, these results were promising, because, upon further inspection, they reduced some of the errors that I was seeing, in the development set. It seemed that some of the position features were more impactful than others so I could use some of them in further experiments to build out a more robust set of features.

One other interesting feature was position::unknown_teamAbbreviation=none. When a player's teamAbbreviation was equal to none it meant that the player was not on a team at the end of the season. Due to the way the data was stored on nba.com all the unknown positions with teamAbbreviation=none had a fppg value of 0, in the development set. Therefore, this feature helped the model predict players whose fppg would equal zero in the following year. This incrementally improved the problem mentioned earlier where the model was misclassifying players whose fppg was actually 0.

Due to the number of features, I extensively used Weka's WrapperSubsetEval with

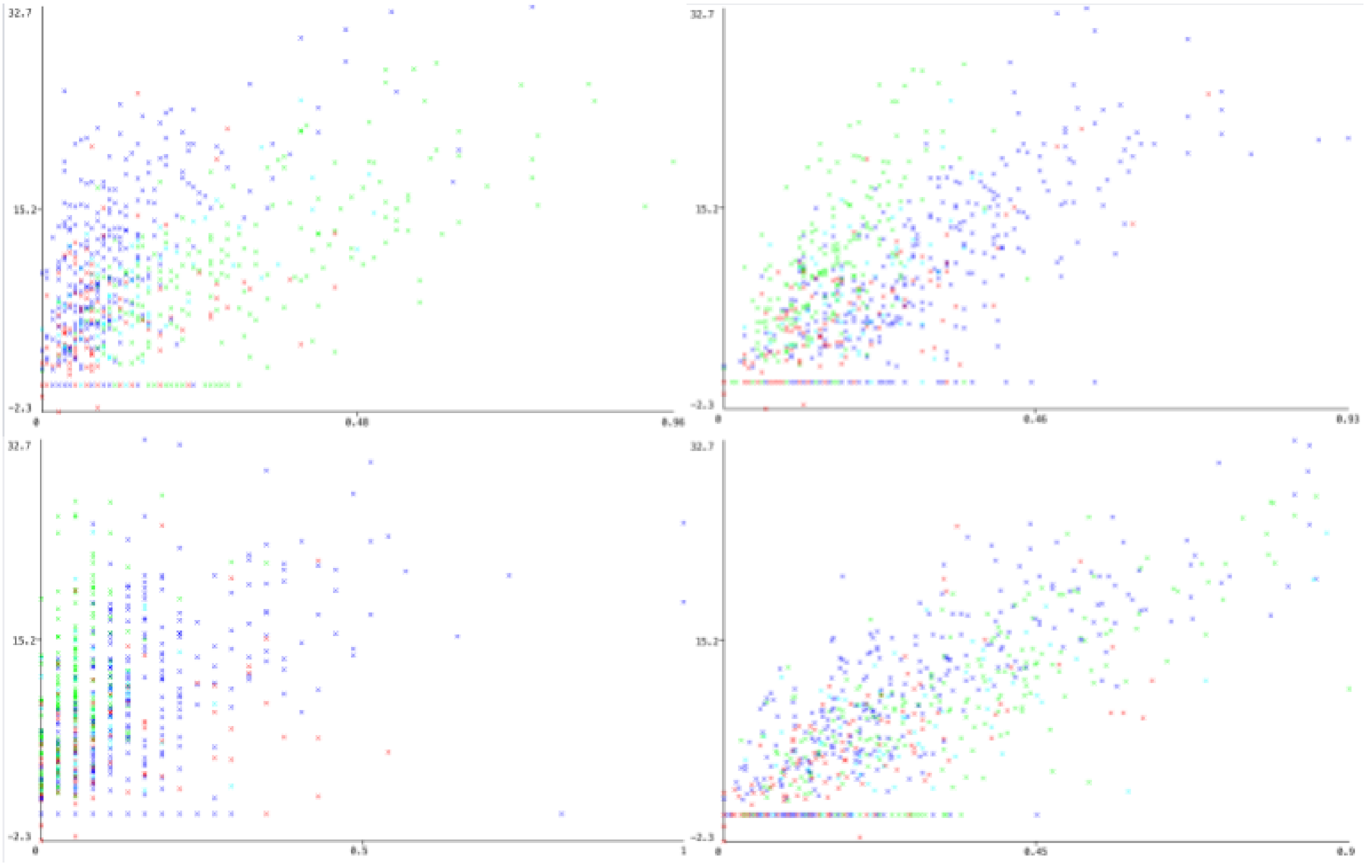


Figure 3: development dataset distribution by position. Positions - guard(green), wing(teal), big(blue), unknown(red). Clockwise from the top right - ast x fppg, reb x fppg, pts x fppg, stl x fppg

a GreedyStepwise forward search. I repeatedly added a number of features based on hypothesis or error analysis and ran the attribute selection to determine which features would have the biggest impact on my model. Through several iterations of this and other error analysis, I ended up with my final feature set. In addition to the baseline features, I added curr_tot_fppg, per_min_ftm, post_dreb, post_oreb, post_fga, post_fgm, post_pfd, post_pts, pre_post_diff_plusMinus, pre_tot_fgPct, prev_tot_fga, prev_tot_fppg, prev_tot_plusMinus, prev_tot_gp, prev_tot_pts, teamAbbreviation=none, position::big_curr_tot_reb, position::big_curr_tot_stl, position::big_per_min_pfd, position::big_prev_tot_dD2, position::guard_age, position::curr_tot_pts, position::guard_prev_tot_fG3A, and position::unknown_teamAbbreviation=none.

With the final feature space set, I wanted to reevaluate the three models that I used to select a baseline model. I expected linear regression to perform the best since the feature selection was based on the error analysis of a linear regression model, and it did have the best performance. However, SVM's performance was similar, as shown in figure 4. Due to the fact that these two models performed similarly, I chose to perform parameter tuning on SVM's complexity parameter to see if that would yield better results. My hypothesis was that an optimized SVM model

	Linear Regression	Support Vector Machine	Random Forest
Correlation Coefficient	0.8505	0.8505	0.8366
RMSE	3.6061	3.6194	3.7619

Figure 4: final feature set model comparison

would be better than linear regression because it is better at preventing overfitting. There could be some noisy features that confuse the linear model but can be ignored by SVM. Additionally, SVM does a better job of modeling non-linearity in high dimensionality feature spaces such as the numerous features included in the final feature set.

I used the CVParamterSelection in Weka to perform the parameter tuning. Lutz's research on fantasy football predictions found the c=0.25 performed best[6], so I tuned the model on complexity values of 0.25, 2.6, 5.0. The results confirmed Lutz's observation, that c=0.25 is the best setting for complexity when predicting player performance via SVM models. With c=0.25 the SVM correlation coefficient was 0.8508 and RMSE is 3.6147. Tuning this parameter did not show statistically significant improvement to the model.

Even though the linear regression and support vector machine models both seemed to be an improvement from the baseline, neither showed statistically significant improvement. The linear regression model had a p-value of 0.969 and the support vector machine had a value of 0.913.

Final Results

In order to asses the final results I used the final set of data that was held out over the course of the project. I never looked at or used any of the data so that I could test how the models perform against unseen data. Since both the linear regression and SVM models were so similar I decided to compare them both to the baseline model. As noted in figure 5 the support vector machine model was the best by a small margin. However, none of the models proved to be a statistically significant improvement over the baseline model.

Reflection

This paper has examined the use of machine learning to make accurate predictions of future NBA player's fantasy value. This included data gathering, model evaluation, feature selection, and model tuning. The final result showed that a simple feature set performed best, as the baseline model achieved a correlation coefficient of 0.8705. Despite not making any marked improvements on the baseline model a

	Baseline	Linear Regression	Support Vector Machine
Correlation Coefficient	0.8705	0.8791	0.8795
RMSE	3.4365	3.3187	3.3181

Figure 5: final models performance over unseen data

correlation coefficient of 0.8705 is a very good value and proves that there are great opportunities to use machine learning to predict players' fantasy outlook in season-long leagues.

One area that I overlooked was instance selection. The full set of players that played in the NBA between 2010 and 2018 were included. In retrospect, this list could have been pruned to better fit the problem space. One problem with this many instances is that the data was highly skewed towards lower fppg values as shown in figure 6. By limiting the instances based on games or minutes played I could reduce the number of instances with lower fppg and prevent the model from over predicting lower fppg values.

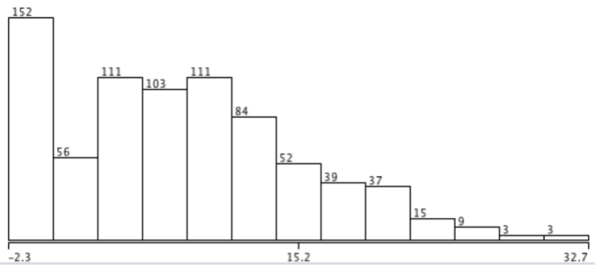


Figure 6: development set distribution of fppg

In addition, I divided the instances randomly between the development, cross-validation, and final sets which could lead to one set being more skewed than others. In order to fix this, I should have evenly distributed the fppg values among the different sets to make the sets more similar.

Over the course of this project, it became clear how important and difficult feature selection can be in the statistical analysis of sports. There are a multitude of possible features and virtually endless combinations of features. It required many iterations of error analysis, experimentation, and automated feature evaluation to select the features used in the final model and even those could be improved upon.

Additionally, as I came to understand the data better I realized some of the decisions I made when selecting my baseline features were poor. For example, including the season as a feature does independently help the model predict a player's future performance. It could help my model when I am using data for randomly distributed years, because one year could have higher average fantasy points, across all players. However, the application of the model would be to predict next year's fantasy points per game. Therefore, a model that factors in that 2015-2016 had higher than average fantasy points per game, would not help predict players' 2019-2020 fantasy value. Season data could be converted into a numeric feature and combined with other statistics to show how certain statistics become more relevant over time. For example, over the last ten years, three-pointers have gained importance across the NBA[13]. By combining a numeric season feature with three-pointers attempted the model could increasingly favor three-point shooters as the years progress.

One feature that could have helped the model is the pace of a player's team. The faster a team plays the more opportunities a player has to accrue fantasy-relevant statistics. Unfortunately, this data was not easily available so it could not be incorporated into the final feature set.

Furthermore, players do not operate in a vacuum they are greatly affected by other players on their team. Whether it is additional assist opportunities playing next to a good shooter or a reduction in rebounds playing with a larger player, the fantasy outlook for players is highly dependant on their teammates. Due to the complexity of measuring team composition, I did not include any features relating to the interdependence of teammates. However, this may be an interesting area for further research, as only one paper, I found, tried to model the impact of a player's team on their value [7].

When it came to model selection it made sense that linear regression worked best with the baseline features because calculating a player's fantasy points is an algebraic equation. Therefore, it logically follows that with simple features a linear model would perform best. When we added more dimensions I expected the SVM model to outperform the linear regression, due to its ability to model high dimensionality and non-linearity. SVM models can be especially helpful for problems like this one where it can be hard to understand the relationships between different attributes. In the end, my assumption proved false, as the SVM and linear models exhibited similar performance.

Given more compute resources it would have been interesting to investigate the use of Neural Networks in predicting fantasy points. Both Lutz[6] and Harrison[7] had varying levels of success with Neural Networks. This model could be advantageous due to the complexity of the feature space. Neural Networks have the advantages of adapting to the dataset and learning the hidden layers. This would be helpful for NBA data due to the extensive set of available features. However, it could turn out that a more complex model like Neural Networks do not perform better than SVM. Lutz did not find Neural Networks helpful in predicting NFL player performance; although, this could be due to his small number of instances.

Finally, the overall performance of the final model was very good. While there is much more that could be investigated, in order to improve the performance, it suffices to say this model could serve as one tool that a fantasy basketball manager uses to asses and select their players for an upcoming season.

Glossary

Prefixes

post_ - per game statistics for the current season, from the all-star game onward

pre_post_diff_ - the difference between pre-all-star break per game statistics and post-all-star game statistics

prev_tot_ - the cumulative statistics for the previous season

curr_tot_ - the cumulative statistics for the current season

curr_prev_diff_ - the difference between the cumulative statistics of the current season and previous season

per_min_ - the per-minute statistics for the current season

Statistics

ast - assists

blk - blocks

blka - blocks allowed (times the player’s shot was blocked)

dD2 - double-doubles

dreb -defensive rebounds

fg3A - three-point field goals attempted

fg3M - three point field goals made

fga - field goals attempted

fgm - field goals made

fgPct - field goal percentage

fta - free throws attempted

ftm -free throws made

ftPct - free throw percentage

gp - games played

min - minute played

oreb - offensive rebounds

pf - personal fouls

pfd - personal fouls drawn

plusMinus - the net score change while the player is on the court

pts - points

reb - rebounds

stl - steals

tD3 - triple-doubles

tov - turnovers

wPct - win percentage of the player’s team

Other

age - the age of the player

teamAbbreviation - an abbreviation of the player’s team (i.e. BOS)

position - the player's position (guard, wing, big, unknown)
draftNumber - what pick in the draft the player was selected
school - where the player attended college

References

[1]A. Augustyn and N. Zegura, "Fantasy Sports", *Encyclopedia Britanica*. 2016.

[2]"Global Fantasy Sports Market Growth (Status and Outlook) 2019-2024", *Market Study Report*, 2019. [Online]. Available: https://www.marketstudyreport.com/reports/global-fantasy-sports-market-growth-status-and-outlook-2019-2024 . [Accessed: 17- Oct- 2019].

[3]T. Ross, "This Isn't Your Dad's NBA: Thank Big Data", *The Atlantic*, 2015. [Online]. Available: https://www.theatlantic.com/entertainment/archive/2015/06/nba-data-analytics/396776/. [Accessed: 11- Dec- 2019].

[4]N. Dunnington, "Fantasy Football Projection Analysis". Honors Thesis. University of Oregon. 2015. Available: https://pdfs.semanticscholar.org/d91c/4de37816c9c712112dc49850a046f35ee80a.pdf

[5]E. Hermann and A. Ntoso, "Machine Learning Applications in Fantasy Basketball", Stanford University Department of Computer Science, 2017.

[6]R. Lutz, "Fantasy Football Prediction", University of Massachusetts Amherst College of Information and Computer Sciences, 2015.

[7]Z. Harrison, "NBA Game Prediction Using Neural Networks and Other Machine Learning Techniques", University of Wisconsin Madison Department of Electrical and Computer Engineering, 2018.

[8]C. Shivakumar, "Learning to Turn Fantasy Basketball Into Real Money", University of Pennsylvania, 2015.

[9]P. Steenkiste, "Finding the Optimal Fantasy Football Team", University of Stanford, 2015.

[10]"Age Effect to Basketball Players", *Nbaminer.com*, 2014. [Online]. Available: http://www.nbaminer.com/golden-ages-of-basketball-players/. [Accessed: 11- Dec- 2019].

[11]I. McMahan, "How (and why) position-less lineups have taken over the NBA playoffs", *the Guardian*, 2018. [Online]. Available: https://www.theguardian.com/sport/blog/2018/may/01/how-and-why-position-less-lineups-have-taken-over-the-nba-playoffs. [Accessed: 11- Dec- 2019].

[12]R. Goldberg, "Brad Stevens Says Celtics Have 3, Not 5, Positions Now", *Bleacher Report*, 2017. [Online]. Available: https://bleacherreport.com/articles/2720250-brad-stevens-says-celtics-have-3-not-5-positions-now. [Accessed: 11- Dec- 2019].

[13]S. Shea, "The 3-Point Revolution", *Shottracker.com*, 2019. [Online]. Available: https://shottracker.com/articles/the-3-point-revolution. [Accessed: 11- Dec- 2019].