NBA Player Value: Using Machine Learning to Help GMs Better Run Their Teams

Grant Cox

github.com/gcox32/NBA_Player_Value

***Abstract*—NBA General Managers are given the often-thankless task of constructing or reconstructing a professional basketball franchise. More often than not, they get it wrong and overpay players due to social pressure or underpay players who don't meet poorly defined features. Using gradient boosting models, we were able to predict team success (defined as a season ending rank of 1 to 4) with an accuracy of** 82% **and an ROCAUC of** 0.92**. Our best salary regression model could only predict with an R-squared of** 0.52**. The differences in model strengths and feature importances in the models told the story best regarding differences between player "value" (contributing to team success) and player "worth" (what someone is willing to pay).**

I. INTRODUCTION

GENERAL Managers (GM) in the NBA are tasked with the responsibility of constructing and paying a roster composed of high-level athletes—every one of which feels he ought to be paid what he is worth. All too often, these GMs succumb to external pressures: to sign the player the fans want or that the media has placed on a pedestal, to pay a player more than he is worth out of fear of missing out on a player, offering too little to a player out of pride or fear (or both) and missing out—the reasons go on and on. None of these pressures is based in anything objective. GMs need something on which to fall to make such costly and high stakes decisions, a means of defending the decisions they make. This project began with Michael Lewis' "Moneyball," a philosophy of valuing features that actually move the success needle over features that are perhaps gratuitous. As we saw in Lewis' book, the money that powers professional sports leagues can be ruled by forces other than logic or math. Using machine learning, we wanted to fix that, or at least give general managers the opportunity to see it from a data driven perspective.

Our process and insights hinged on the examination of historical NBA data from two angles. First, at the team-level, what is a successful team and does a successful team look like? Can we find trends for success? 2) At the player-level, which individual players are/were paid the most and what do they have in common? Can we distinguish similar looking players better than just using the traditional labels of point guard through center (using principal component analysis)? By determining what is valuable (team-level analysis) and stacking it up against what seems to define player worth (player-level salary analysis), we both looked to assess what has worked and prescribe what would work better. Using machine learning, we looked to give any GM an upper hand who is willing to take it.

II. INSIGHTS

Our machine learning efforts culminated in a classifier analysis on the team-level and regression analysis on the player-level, predicting success and salaries respectively. Machine learning regression models have a hard time pinning down just what exactly gets a player paid. Because these models were able to rely on 30+ features and still had a difficult time predicting, we must conclude that we did not have the features that would predict salary well (i.e. they aren't found on a stats sheet). Classifier models did not have the same trouble with predicting team success. Our models performed well at predicting whether a team would finish the season ranked in the top 4 based on things like how well the team show 3-pointers. Feature Importance between our salary regression and our team success classifier varied enough to raise flags—GMs aren't using success predictors to determine player worth. The player that deserves to get paid (i.e. who is the most valuable) is the one who can shoot, both accurately and often.

III. DATA AND FEATURES

The data was acquired via (extensive) web scraping of a few different websites using the BeautifulSoup package. The primary site, basketball-reference.com, provided both team-level and player-level data. The website

hoopshype.com was scraped for salary data: both player salaries and team salary caps for each year.

IV. SCRAPING AND CLEANING

Team-level data was composed on separate scraping efforts, first for win-loss records, then for team-level stats such as descript-tive statistics for team shooting or team defense. Player-level features were aggregated from four separate sources. 1) standard stats per 100 possessions, 2) advanced stats per 100 possessions, 3) season level shooting stats such as average distance from the basket, and 4) our predictor feature season salary. Scraping player data was much more involved as we scraped, created, and ultimately merged four separate pandas data frames. The data that needed the most cleaning was the salary data from hoopshype.com. For example, a datapoint we might read as "$20,000,000" was actually written in the HTML as "\n\t\t\t$20,000,000\t\t\t\n"—a very messy string. This required me to identify the characters that needed to be dropped before changing the datatype to something we could manipulate (i.e. string to float/integer). The data from basketball-reference.com was for the most part very clean, handling null values was an important component of this project. Depending on the feature, some data was filled using the population mean, and for some data, a null value in the wrong feature meant dropping the record entirely.

V. EXPLORATORY DATA ANALYSIS

On the team-level, we wanted to identify success, so that required of us that we define our terms. Our definition of success is a critical consideration, and for our project, we would define it first as "making playoffs," or ranking 1 through 8 at season's end, but ultimately we would define success as a slightly higher bar: ranking 1 through 4 at season's end. We began our EDA efforts with some scatter plots (below) that could most simply demonstrate appropriate win totals.

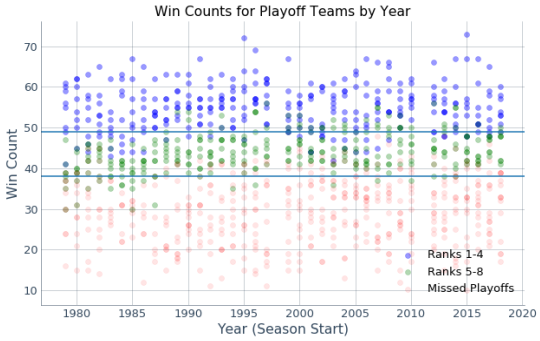

Fig. 1: Successful Team Designations

The previous plot showed a clearly delineation from missing playoffs and making playoffs at the 39 win mark; it showed a similarly clear delineation at the 49 win mark for our metric for success. This mark looks to have held true even across 40 years of team data. Our next step was to search for correlates with the feature of win-count (W). Where we saw the most notable strong correlation was with the feature was three-point shooting percentage (3P%). Teams that were more accurate from the 3-point line won more games. Interestingly, when we first looked at this correlate, we saw a coefficient of only 0.18, but this included every team-season dating back to 1979. When we limited our data span, only included team-seasons dating back 5 years, we saw a correlation value of 0.58 (below)

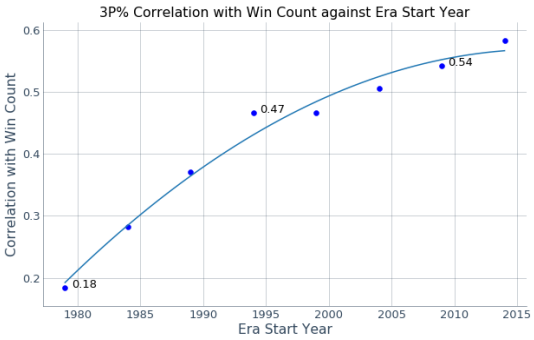

Fig. 2: 3P% Correlation

Player Clustering with Linear Discriminant Analysis

One of our primary objectives was to accurately determine player value. In order to best do this, it was necessary to distinguish differences between players who are listed as similar. Our hypothesis was that there is a more robust method for describing a player's position than just as one of five positions (point guard, shooting guard, small forward, power forward, or center). For example, compare the play styles of Ben Simmons and Steph Curry. Both are listed as Point Guards, but their contributions could not look more different.

In order to do this, we used our extensive player-level data (per 100 possessions stats, advanced stats, and shooting stats) to cluster players into 8 distinct positions that better described their contributions to the team.

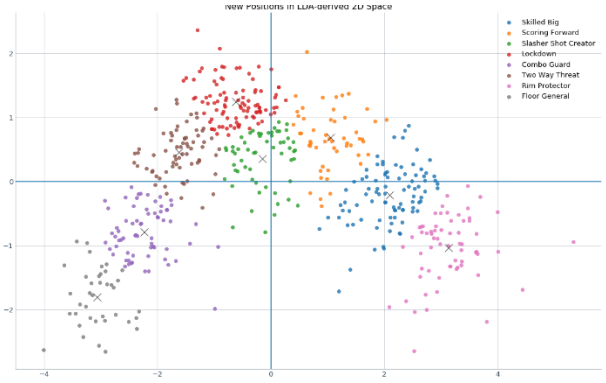

Fig. 2: LDA Clusters

Once new labels were merged with our player-level data, we were able to add another layer to our EDA and inferential statistics as we could then consider a player's cluster when asking questions of value.

VI. INFERENTIAL STATISTICS

With our team-level inferential statistics, two sample t-tests of independence showed us that money doesn't buy wins--at least not at a statistically significant level. The differences in mean Win count for teams paying over 120% of the salary cap for any given year, and teams paying under 120% of the salary cap for any given year varied greatly--44 and 38, respectively. However, when we sampled from these groups and ran a few t-tests (read: ten thousand t-tests), we saw that that this difference was not in fact significant at even a 0.10 confidence. This is incredibly "significant" in a practical sense as it bolsters our "Moneyball" approach that success can be afforded thanks to misplaced resources elsewhere. Players that are costing some teams more aren't necessarily translating that value to winning.

On the player level, a much more robust data set, Python allowed us to scan for statistically significant differences in 6 different features across 8 subsets of the data, using 10,000 t-tests each time (48,000 in total) in about 2.5 minutes. From this scan, we learned the following: 1) the "Lockdown" player cluster was flagged for statistically lower mean Player Efficiency Rating (PER), 2) "Rim Protector" cluster was flagged for having statistically lower mean Usage Rate (i.e. they touch the ball less), 3) "Combo Guards" were conversely flagged for statistically higher mean Usage Rate, and 4) "Rim Protectors" were the only cluster with a lower significant difference in points scored. These player-level inferences added to the story critically, once we could frame them against our machine learning conclusions. Namely, they gave us some direction as to the type of player who might be overpaid. "Rim Protectors" for instance, contribute significantly less, on average, to points scoring.

VII. MODELING

As with the previous sections, we dealt with data on both the team level and the individual player level.

A. Player Salary Regression

We began our regression efforts with perhaps the most intuitive approach, Ordinary Least Squares from scikit-learn. Using OLS we were able to test many iterations of the available features (of which we had nearly 50). OLS however could only give us an R-squared value of 0.413.

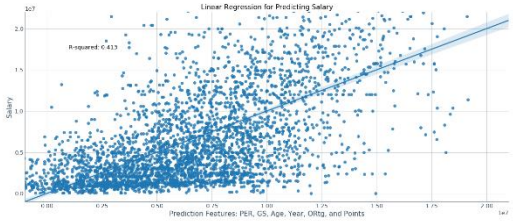

Fig. 3: OLS Regression

From the above regression, we saw just how difficult it is to apply a linear model to such varied data. If anything, this supports our underlying premise that players are paid on factors other than what they produce on the stats sheet. If we look towards the right of the plot, where stats are higher (read: better), we see there are points well below the regression line. Such players would be easy to designate as "underpaid." Players above this line could also just as easily be considered "overpaid." We also ran a Random Forest regression and an XGBoost regression and compared R-squared values across the three to find a best model.

TABLE I: Regression Models

Model	*R-squared*
OLS	0.41
Random Forest	0.51
XGBoost	0.52

Important to our efforts were the feature importance values of our best model (below).

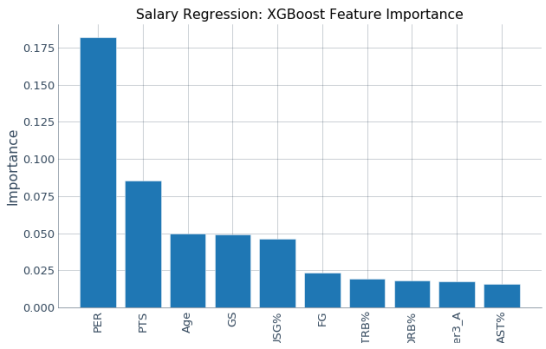

Fig. 3: XGBoost Regression Feature Importance

PER strives to measure a player's per-minute performance, while adjusting for pace. A league-average PER is always 15.00, which permits comparisons of player performance across seasons. PER considers accomplishments, such as field goals, free throws, 3-pointers, assists, rebounds, blocks and steals, and negative results, such as missed shots, turnovers and personal fouls. The formula adds positive stats and subtracts negative ones through a statistical point value system. The rating for each player is then adjusted to a per-minute basis so that, for example, substitutes can be compared with starters in playing time debates. It is also adjusted for the team's pace. In the end, one number sums up the players' statistical accomplishments for that season. PER is one of our better aggregate "value" statistics, but it potentially takes into account some statistics that aren't as relevant to success as others. PER is more or less a measure of how well a player fills up a stats sheet, not a measure of whether a team has a better chance of winning.

Points in any sport is probably the loudest stat, understandably the most important Age, Games Started (GS), and Usage (USG%) are intuitive as well but don't tell us anything about the player's production. FG made is comparable in importance to two rebounding metrics, "attempted corner threes," and assist percentage.

B. Team Success Classifier

Our approach first required of us that we define "success." Once again, this definition began with the bar of "making the playoffs" which corresponded with a season end rank of 1 to 8; we however decided to raise the bar for success to a season end rank of 1 to 4, as many teams live in that 5 to 8 season end rank year after year and become dissatisfied quickly. As with our regression modeling attempts, we started with a straightforward regression (this time logistic as we were dealing with probabilities of ones and zeroes) but proceeded to using both Random Forest and XGBoosting models as well. The logistic regression is demonstrated below.

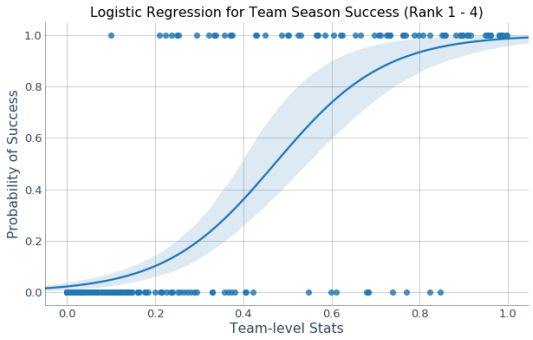

Fig. 4: Logistic Regression

We ultimately went with an XGBoost model as our classifier as it scored well in both prediction accuracy and ROC AUC (0.82 and 0.92, respectively).

TABLE 2: Classifier Models

Model	*Accuracy*	*ROCAUC*
Logistic Regression	0.89	0.92
Random Forest	0.81	0.84
XGBoost	0.82	0.92

As with the player-level salary regression, for our efforts, it was important that we observed feature importance from our best model, that we asked the question: what features best predict team success?

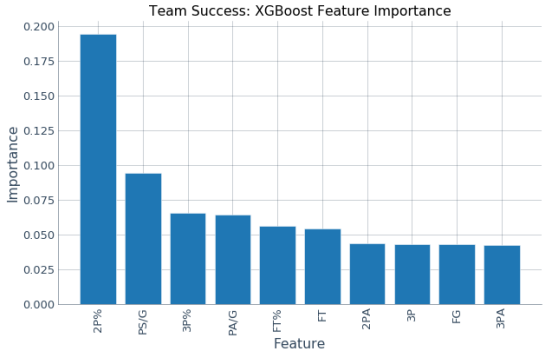

Fig. 5: XGBoost Classifier Feature Importance

Like our regression models, we can examine feature importance for our classifiers; we'll look specifically our XGBoost model. 2-point accuracy is the heaviest influencer of the model, followed by Points Scored, and then 3-point accuracy, then Points Against. Importantly, there are 3 separate 3-point related statistics (3P%, 3P, and

3PA) in our top ten. Other than the very general PS/G and PA/G, every feature is a shooting statistic.

VIII. CONCLUSIONS

The primary conclusion is two-fold, in line with every step of the project thus far. Predicting player salary is a daunting task with only the best models producing just over a 0.5 R-squared. Considering we had nearly 50 features available to predict one, we are forced to ask the question: are these even the appropriate features? Or are GMs looking at a stat sheet that isn't available by watching the game? A fundamental problem in data science is the procurement of the data needed to solve the problem—accrual of the features that help answer the question. Not only do we not seem to *have* such features, it would seem that such information does not exist in a codifiable way. In essence, GMs are, consciously or otherwise, paying players based on variables we don't have available. The team success predictions models behaved much differently, with much higher predictive capability. If it is easy to find the features that strongly predict a successful team, why is it not as easy to find the features that strongly predict a well-paid player? They should go hand in hand, and they do not.

Action Steps

Concerning practical and actionable steps, three things stand out from our team success feature importance bar plot: 1) two-point shooters win games, 2) three-point shooters win games, and 3) players who make their free throws (FT) win games. Very generally, a GM ought to pay the player who fits at least one of those.

In addition, if you want to pay a big man (traditionally, a center or power forward), get one who isn't a liability when he goes to the free throw line. If you must pay a player who can't shoot threes, ensure that he takes high percentage twos. Otherwise, invest in three-point shooters. Defense and rebounding are much further down the priorities list. Don't spend money on a player because he has impressive blocks or outrebounds other players because he "wants it more" or is a "harder worker." Again, most simply, pay players who can shoot. As the hallmark of what makes the sport unique, it is almost poetic that somehow this is the skill that is undervalued.

IX. FUTURE WORK

Certain clusters are probably more valuable than others, objectively, meaning our LDA and clustering was at least worth it in that regard. Since we know which features are strong predictors of team success, we should want to know which clusters test the strongest for those features.

In order to establish whether a player is overpaid or underpaid more specifically, we could set out to construct a coherent and intuitive algorithm for assigning value to a player's stats. We could then weigh that against what that player is/was actually paid (his worth).