Predicting the Improvement of NBA players

Mohit Khandelwal

March 19, 2021

1. **Introduction**

1.1 Background National Basketball Association (NBA) is the best basketball league in the world with millions of fans worldwide. How players on a team perform is the most important factorthatdetermines which team wins the championship. Players' pay are largely based on their past performances. However, player performance change from season to season. Each year there are a number of players who improve dramatically over last year. Those players bring a lot of value, both competitively and economically, to the teams they belong to. Their importance is widely recognized by the NBA in that the player who improved the most over last season is awarded Most Improved Player (MIP) Award. Therefore, it is advantageous for teams to accurately predict whether and how much a player will improve in the next season. For example, this information can be used to target players to acquire in trades or signings.

1.2 Problem Data that might contribute to determining player improvement might include his performance last season, his age,hisdraftstatus,hisposition,andmetricsthatdescribewhatkindofplayerhe is. This project aims to predict whether and how much a player will improve the next season based on these data.

1.3 Interest Obviously, NBA teams would be very interested in accurate prediction of player improvement, for competitive advantage and business values. Others who are interested in NBA such as fans and fantasy basketball players may also be interested.

2. Data acquisition and cleaning

2.1 Data sources Most player stats, position, age, anddraftpositiondatacanbefoundintwoKaggledatasetshere andhere. These two datasets, however, lack data for certain years. Forexample,theplayerstats

datasetendsin2017,andtheplayerdraftdatasetstartsin1978andendsin2015.Tocomplement these two datasets, I scrapedbasketball-reference.com forplayerseasonstatsof2018andplayer draft positions of 1965-1977 and 2016-2017 (players drafted in 2018 has yet to play in NBA).

2.2 Data cleaning Data downloaded or scraped from multiple sources were combined into one table. Therewerea lot of missing values from earlier seasons, because of lack of record keeping. I decided to only use data from 1980 season and after, because of later seasons have fewer missing values and basketball was a lot different in the early years from today's game. There are several problems with the datasets. First, players were identified by their names. However, there were different players with the same names, which cause their data to mix with each other's. Though it was possible to separate some of them based on the years, teams,andpositionstheyplayed,Idecidedthatitwasnotworththelargeeffortto doso,because such players only accounted for ~1% of the data. Therefore, players with duplicate names were removed. Second, multiple entries existed for players who changed teams mid-season. This cause their seasonal data to represent multiple samples with incomplete data. I wrote script to extract total season stats for these players, and discarded partial season rows. Third, there were two short seasons in recent NBA history, during which less than the normal 82 games were played. This has caused stats in those seasons to be artificially smaller than other seasons. To correct that, I normalized cumulative features such as points, rebounds, etc. as if 82 games were played. After fixing these problems, I checked for outliers in the data. I found there were some extreme outliers, mostly caused bysometypesofsmallsamplesizeproblem.Forexample,some players had only played a few games or a few minutes the entire season, and had performed extremely well or poor in those minutes. Therefore, seasons during whichlessthan20gamesor 100 minutes were played were dropped from the dataset.Similarly,therewereplayerswhoonly took one 3-point shot, but made it, therefore had 100% shot accuracy. I changed the shot accuracies for players who shot less than 10 shots to missing values. There were 4 features which had missing values. Games started were imputed from minutes played because starters usually play more minutes. Missing 3-point accuracies were imputedwithaverysmallvalue(0.05)becauseifaplayerrarelyshoots3s,itisprobablybecause he is not very good at it. Missing free throw accuracies were imputed using the mean of all players. Missing draft positions, meaning undrafted, were imputed using position 61 (the position after the last position in the draft, 60th).

2.3 Feature selection After data cleaning, there were 13,378 samples and 49 features in thedata.Uponexaminingthe meaning of each feature, it was clear that there was some redundancy in the features. For

example, there wasafeatureofthenumberofreboundsaplayercollected,andanotherfeatureof the rate of rebounds he collected. These two features contained very similar information (a player’s ability to rebound), with the difference being that the former feature increased with playing time, while thelatterfeaturedidnot.Suchtotalvs.raterelationshipalsoexistedbetween other features. These features are problematic for two reasons: (1) A player’s certain abilities were duplicated intwofeatures.(2)Aplayer’splayingtimewereduplicatedinmultiplefeatures. In order to fix this, I decided to keep all features that were rates in nature, and drop their cumulative counterparts (Table 1). There were also other redundancies, such as that total rebounds are the sumofoffensive rebounds and defensive rebounds. For features that can be calculated bysumofotherfeatures,I decided to drop them (Table 1). After discarding redundant features, I inspected the correlation of independentvariables, and found several pairs that were highly correlated (Pearson correlation coefficient > 0.9). For example, shots attempted, shots made, and points scored were highly correlated. This makes sense, after all, you score points by making shots. From these highly correlated features, only one was kept, others were dropped from the dataset. After all, 24 features were selected.

Table 1. Simple feature selection during data cleaning. Kept features Dropped features Reason for dropping features

TRB%, ORB%, AST%, STL%, BLK%, TOV%,

TRB, ORB, AST, STL, BLK, TOV

Two similar features (one being total, one being rates) depicting the same ability of players. TRB%, ORB%, WS, OWS DRB%, DRB, DWS Total = offense + defense. Dropped defense.

TS%, FGA, 3P%, 3PA 2PA, 2P, 2P%

Field goal = 2-point shots + 3-point shots. Dropped 2-point shots.

TS%, WS FG%, eFG%, VORP, BPM, OBPM, DBPM

Slightly different features that depict the same overall abilities of players.

3. Exploratory Data Analysis

3.1 Calculation of target variable Player improvement year over year was not a feature in the dataset, and had to be calculated. I chose to calculate the difference of win shares between two consecutive years as the target

variable. Win shares were chosen out of a few metrics because it is the mostinterpretable,after all, we play basketballtowin.Calculatedplayerimprovementhadanormaldistributioncentered around 0, with most values between -6 and 6. To verify if this calculation is consistent with people's eye-test of player improvement, I plotted the rank of improvement of past Most Improved Players winners among all players, andfoundthatinmostcases,theywereamongthe most improved players (Figure 1). Thissuggestedthatthechosenmetricofplayerimprovement, was a reasonable one.

3.2 Relationship between improvement and age It is widely accepted that younger players are more likely to improve than older players, and it was indeed supported by our data. Players' median improvement declined as players' age increased (Figure 2), and the mean improvement of different age groups (<25, 25-29, 30-34, >35) were all significantly different from each other (z-test, $p<0.001$, except for 30-34 vs. >35, $p=0.002$).

Figure 1. Rank of delta-win-share of Most Improved Players winners among all players of each year

Figure 2. Box plot of improvement of players of different ages.

3.3 Relationship between improvement and overall ability The hypothesis here is that players who are already stars don't have much room to improve, while a mediocre player can still improve. Our data were consistent with this hypothesis.Using win share per 48 minutes (WS/48) as a measure of a player's overall ability, I observed a negative relationship between a player'soverallabilityandhisimprovementnextseason(Figure 3). The mean improvement of star players (WS/48>0.2),solidplayers(WS/48between0.1and 0.2), rotational players (WS/48 between 0 and 0.1), and "scrubs" (WS/48 below 0) were significantly different from each other (z-test, $p<0.001$) (Figure 4).

3.4 Relationship between improvement and minutes played I hypothesized that players with less playing time might be more likely to improve. If a team recognizes a player's positive contribution during his limited time, he is likely to get more playing time, and therefore increase his production and/or improvehisskills.Ontheotherhand, if a good player is already a starter, he is already playing a lot of minutes and can't get more playing time.Afterinspectingthedata,itwastruethatplayerswhoplayedlessthan25minutesa game had statistically higher improvement than those who played more than25minutesagame (z-test, p<0.001). However, the actual difference of mean between the two groups was small (~0.7).

3.5 Relationship between improvement and games played I observed anegativerelationshipbetweenplayerimprovementandthegamesplayed(Figure5). If a good player missed significant numbers of games, it was probably because ofinjury,which might have negatively impacted his performance. He might return to his former form next season, and therefore improve. Players who played fewer than 50 games were more likely to improve than those who played more than 50 games. (z-test, p<0.001, difference of mean=1.3).

Figure 3. Scatter plot of improvement and player overall ability (measured by win share per 48 minutes)

Figure 4. Histogram of player improvement separated into 4 groups based on how good a player is.

Figure 5. Scatter plot of player improvement and games played.

3.6 Relationship between improvement and positions There is this myth among NBAfansthatfrontcourtplayerstakelongertoadapttotheNBAthan backcourt players, therefore they would have smaller improvement in the first few years. I transformed the feature of player position intoabinaryfeature(frontcourtvs.backcourtplayers) and found that there was no difference between frontcourt and backcourt players in their improvements, even in their first 2 years (z-test, p=0.34)

3.7 Relationship between improvement and last year's improvement I hypothesized that a player's improvement might be correlated with hispreviousimprovement, because younger players might improve continuously for a few years, and older players might decline for a few years straight. It turned out that the relationship between improvement and prior improvement was negative (Figure 6). In other words, more often than not, a player will "regress to the mean" rather than continuously improve or decline.

Figure 6. Scatter plot of player improvement and that of last season

3.8 Relationship between improvement and draft positions I, as many other basketball fans, thought that players drafted earlier are generally moretalented and therefore more likely to improve than players drafted later, at least in their early years. It turned out this was only true for a few really young and talented players (Figure 7) . Players under the age 20 with different draft positions did not have statistically different improvement (z-test, p=0.16).

3.9 Relationship between improvement and teams I engineered two features based on team information: was a player on a good or bad team, and did the player change team next season. Player improvement and team strength (measured by total win shares) hadaveryweaknegativerelationship.Playersthatchangedteamswereslightly more likely to improve than players that stayed on the same team(z-test,p<0.001,differenceof mean = 0.2).

Figure 7. Box plot of player improvement among different draft groups and ages

4. Predictive Modeling

There are two types of models, regression and classification, that can be used to predict player improvement. Regression models can provide additional information on the amount of improvement, while classification modelsfocusontheprobabilitiesaplayermightimprove.The underlying algorithms are similar between regression and classification models, but different

audience might prefer one over the other. For example, an NBA team executive might be more interested in the amount of improvement (regression models), butageneralNBAfanmightfind the results of classification models moreinterpretable.Therefore,inthisstudy,Icarriedoutboth regression and classification modeling.

4.1 Regression models 4.1.1 Applying standard algorithms and their problems I applied linear models (linear regression, Ridge regression, and Lasso regression), support vector machines (SVM), random forest, and gradient boost models to the dataset, using root mean squared error (RMSE) as the tuning and evaluation metric. The results all had the same problems. The predicted values had much narrow range thantheactualvalues(Figure8),andas a result, the prediction errors were larger as the actual values deviatedfurtherfromzero(Figure 9). These results were not acceptable, because players with large improvement/decline were arguably more important for NBA teams to predict than players with little change in performance. Having larger errors on those predictions was obviously not desirable.

4.1.2 Solution to the problems The reason behind these problems were the uneven distribution of player improvement, in that players with little improvement/decline were more common than players with big improvement/decline (Figure 8). Therefore, the models tried to prioritize minimizing errors on players with little improvement/decline when RMSE was used as the evaluation metric. My solution to thisproblemwastoassignweightstosamplesbasedontheinverseoftheabundances of target values. In other words, players with large improvement/decline would have higher weights in model training and evaluation because they were more rare. Using this method, all models predicted target values with similar range and distribution as the actual target values (Figure 10).

Figure 8. Distribution of actual and predicted improvement using linear regression with equal weights of samples.

Figure 9. Scatterplot of prediction errors vs. actual target values using linear regression with equal weights of samples.

Figure 10. Distribution of actual and predicted improvement using linear regression with different weights of samples based on inverse of sample abundance.

4.1.3 Performances of different models Using the new approach of different sample weights, I built linear regression, SVM, random forest, and gradient boost models using weighted root mean squared error as the evaluation metric. For each model, hyperparameters were tunedusingthesamemetricandcrossvalidation. For comparison, I also built a simple linear regression model withjustoneindependentvariable (age) as the benchmark model. SVM had the best performance among all models, which had ~26% less error than the benchmark model (Table 2). The predicted improvements had linear relationship with the actual improvements (Figure 11).

Table 2. Performance of the regression models. Benchmark (one feature) Linear Regression SVM Random Forest Gradient Boost

Weighted RMSE 3.84 2.98 2.86 2.93 2.96

4.2 Classification models The application of classification models was much more straightforward. I dividedthesamples into two classes (improvement>=0 or <0). The number of samples in each class were about the same.Ichoselogarithmiclossasthemetricherebecausetheresultswouldprobablybepresented with probabilities and logarithmic loss puts more emphasis on the probabilities than other metrics. Logistic regression, SVM, random forest, gradient boost models and a voting model weretunedandbuilt.Amongtheindividualmodels,theSVMmodelperformedthebest(~67.5%

accuracy), and voting model performed similarly as the SVM model (Table 3), though the differences between models were small.

Figure 11. Scatter plot of predicted and actual player improvements of the SVM model.

Table 3. Performance of classification models. Best performance labeled in red. Logistic Regression SVM Random Forest Gradient Boost Voting Model

Log Loss 0.605 0.603 0.612 0.613 0.603

Accuracy 0.675 0.675 0.672 0.672 0.675

No. of True Positives 835 830 810 815 838

No. of False Positives 413 406 396 400 416

No. of False Negatives 438 443 463 458 435

No. of True Negatives 929 936 946 942 926

Figure 12. A section of ROC curves of different classification models.

I also evaluated the models using their ROC curves. In this particular problem, lower false positive rate is moreimportantthanhighertruepositiverate.Inotherwords,itismoreimportant to be sure that a player will improve as predicted, rather than predict all players who will improve, simply because a team can only have limited number of players. In the ROC curves with low false-positive rate, the voting model had slightly higher true positive rates than other models (Figure 12).

5. Conclusions In this study, I analyzed the relationship between NBA players' improvement/decline and their performance and biographic data. I identified age, win share, minutes/games played, improvement last season among the most important features that affect a player's improvement next season. I built both regressionmodelsandclassificationmodelstopredictwhetherandhow much a player would improve/decline. These models can be very useful in helping NBA team management in a number of ways. For example, it could help identify players to acquire, estimate the sizeofthecontracttoofferplayers,planforperformancechangesofplayersalready on the team, etc.

6. Future directions I was able to achieve ~26% improvement from the benchmarkmodelintheregressionproblem, and ~68% accuracy in the classification problem. However, there was still significant variance that could not be predicted by the models in this study. I think the models could use more improvements on capturing players' individual traits. For example, two players might have similar performance metrics, but one might be more physical and the other might be more finesse. The future performance of these two types of players might be different. Another example is that players whose contracts are expiring might play harder/better than players who just signed hefty contracts. More data, especially data of different types, would help improve model performances significantly. Models in this study mainly focused on individual features. However, interactions with teammates, coaches, might also contribute to a player's performance. For example, if a player had a new teammate who is a superstar at the same position, his performance is likely to suffer because of competition. These interactions data are obviously more difficult to extract and quantify, but if optimized, could bring significant improvements to the models.