Alex Damiecki, Nathan Nakahara, Mukul Tekchandani, Grace Wang, John Wen
Professor Zafari
QTM2000-04: Case Studies in Business Analytics
04 May 2018

# NBA All Star Analysis Final Report

## *Model Introduction*

We chose to create a model to predict all stars in the NBA for three primary reasons. The first reason was because the majority of our team was extremely passionate about basketball and more specifically the NBA league. Since we loved the sport so much, we knew that by studying this data set, we would obtain a better understanding of the advanced statistics collected, and what indications these variables have on certain aspects of each player's game. We also wanted to learn about which variables most significantly impact a player's chances of being selected as an all star. Choosing a dataset that our team was passionate about not only increased our productivity but also yielded results that were valuable to our team.

The second rationale for our team's decision to form an NBA all star predictive model was because of the popularity of NBA. We knew that since the NBA was such a renowned subject, we would be able to create a relevant model that future interviewers and recruiters would be able to relate to. We also knew that since the NBA is increasing in popularity every year and is projected to overtake football as America's number one sport, we could build a model that would have value to an increasing amount of stakeholders.

Lastly, we chose to predict NBA all stars with this dataset because the dataset was extremely comprehensive. We had a categorical objective variable which allowed us to develop KNN, Logistic Regression, and Classification Tree models. With over fifty numerical predictor variables we were able to convert some of these variables to categorical variables through a

manual binning process. With these newly created categorical variables, we were also able to generate a Naïve Bayes model. Because we had such a comprehensive data set, we were able to produce extensive models with options for constant improvement.

### *Description of Dataset and Pre-Processing*

Our dataset included all collected stats from every NBA player from years 1950 to 2017, and contained nearly 25,000 records. In order to make the dataset easier to manage, as well as abiding by the rubric (no more than 10,000 records), we trimmed it down to around 10,000 records, starting from the year 2000. We wanted to make sure that we had the entirety of 2000 and onwards contained within the dataset since we wanted to split training and validation by year. Additionally, the game of basketball has evolved with the rise of elite three-point shooters such as Stephen Curry and James Harden. With a more relevant dataset, we would be able to generate models that more accurately predict current and future all stars. In addition to this, some of the more advanced statistics such as Games Started, Usage Rate, and Minutes Played were not collected in previous years therefore removing the earlier years resulted in a cleaner dataset.

After the dataset was trimmed down, we began to remove certain columns and records that did not contribute any additional information. A notable example was a group of columns called blank1, blank2, blank3, etc, that were used for formatting by separating groups of variables that were similar (such as 3-points, 3-points attempted, 3-points per game). Also, between each year a blank record existed that contained no information, which we believed was used for formatting. Finally, we also set N/A values equal to zero, with the original thought being that the players did not register enough of the statistic to get a value for that specific variable. We wanted to consider these situations in the model, so the missing values were

replaced. To confirm this suspicion, we double checked N/A values in the dataset for the years we were looking at to see what was missing. Every N/A found was a either free throws/game or 3-points/game, and 100% of them were due to the player not having any free throws/3-points. However, we do realize that if there is an inappropriate N/A for a zero we could have taken a different approach by looking at the variable more closely and possibly trying to predict what it would have been given the statistics of the player.

To create our variable of interest, we manually added a column in the dataset called All.Star. We then set every value in the column to zero, which would be changed to a one if the player was an all star for that year. We divided the years from 2000 to 2017 ( 3 - 4 years per team member) and then got to work by manually changing the zeros to one for each year. Every member went to the NBA's historical records site for their year, found every all star that played, and added them to our column. To double check our work, each member then took another members years and checked whether the all stars for that year were correct.

Before we ran any analysis, we first removed five variables from the dataset: Year, Player Name, Team, Position, and Index. Our team believed that these variables were irrelevant in deciding whether a player would become an all star. None of these variables were included in the generation of our models.

### *Training and Validation*

Because of the nature of our project, we felt that using a seed and splitting training/validation 80%/20% would not have worked well for our models. Since there are only 25 players elected as all stars per year, compared to the nearly 600 players in the league, there was a good chance that randomly selecting players to use in the training and validation datasets

would give vastly different results as the seeds changed. Instead, we opted to split it by year, starting by using all of 2017's records as validation and 2000-2016 as training. Then, we swapped 2017 with 2016 and ran the model again, and continued to do this for until 2014. This acted as a manual four-fold cross validation for our models. Through this method of testing, we always had the same amount of all stars to work with, while also keeping players within the same year together.

### *KNN - K Nearest Neighbor*

The first model that was created was a KNN model with all fifty of the predictor variables. We ran our model with four validation datasets from 1NN to 9NN and found that 3NN proved to be the most consistent model. This model had an average accuracy of 0.973 and a sensitivity of 0.521. Even though the accuracy measure was extremely high, it did not have much value because of the skewness of our data. Since most players in the NBA were not all stars, classifying the true negatives right did not mean much. The low sensitivity rate showed that this model was not serviceable at accurately predicting all stars. This was most likely due to the standardization of the variables. From prior knowledge we knew that some statistics were naturally weighted more in the all star selection process such as Points Per Game, Win Shares, etc. Since all fifty predictor variables were weighted equally, it was foreseeable that KNN would not be a serviceable model.

After we created our logistic regression and classification tree model we returned to our KNN model to see if running KNN with limited variables would improve this model. Based off of our logistic regression and classification tree models, we chose variables that had high impact such as Win Shares, Points Per game, Usage, Box Plus Minus, etc, and reran KNN with just

these variables. Surprisingly, the accuracy and sensitivity of this model did not change from our previous KNN model. The predicted results were exactly the same. We thought that since some of the variables used such as Win Shares, Box Plus Minus were weighed calculations of other variables, this model's predictive power did not change much even with the removal of many of the variables.

### *Logistic Regression*

Because logistic analysis takes any type of variable, and is used to predict a binary variable, no data transformations were required other than removing the five variables mentioned previously.. We first ran a stepwise, forward, and backward to determine which variables to use. But we immediately ran into a problem with these functions. Each analysis would take up to ten minutes to complete. To combat this, we deleted columns of basic statistics such as minutes played and rebounds in favor of more advanced statistics such as total rebounds and usage to create a new dataset. In addition to making R run exponentially faster, we thought this would also eliminate some of the collinearity between some of our variables.

After doing this, we began running a summary of the variables to find which ones affected the odds of being selected as an all star. The variables that are the most important were: Win Share, Box Plus Minus, Usage, True Shooting, and Blocks. We found it interesting that variables such as points were not included in the model. We then later realized that variables such as points were accounted for in variables like Box Plus Minus and Win Share as weighted averages..

In setting the cutoff of our model, we originally had a cutoff of 0.5. This caused our sensitivity to be quite low. In response to this, we decided to lower it to 0.1 in order to capture

more all star caliber players. Every year there are a number of players who are worthy of all star selection, but are left out due to a lack of space. Rather than classify players like these as not all stars, we decided it would be better to classify these players as all stars since they could likely become all stars if a few voters changed their minds. After running logistic regression we got a model with an accuracy of 0.961, sensitivity of 0.932 and AUC of 0.938.

### *Classification Tree*

Another predictive model we constructed from our dataset was a classification tree. For this model, we decided to create two trees: pruned and unpruned. Our pruned tree was generated using a simple rpart function in which we used all of the variables we decided to keep in our dataset. After generating the tree, we noticed that seven rules were generated and five variables made the cut for being included in the model. These variables were Win Shares, Points, Usage Percentage, Free Throws, and Box Plus/Minus.

After generating our pruned tree, we then moved on to creating an unpruned tree and manipulated it through the use of stopping rules. Using the rpart.control function, we set our cost complexity value to 0.006, and our minsplit and minbucket values to three and two respectively. As a result, we got an unpruned tree which was made from 16 rules and included six more variables than our pruned model for a total of 11 variables. Our unpruned tree contained all of the variables from our pruned tree with the addition of Value Over Replacement, Defensive Rebounds, Win Shares per 48 Minutes, Minutes Played, Games Started, and Personal Fouls.

After creating the ROC charts for both the pruned and unpruned trees, we noticed that there was virtually no difference between the two in terms of AUC, despite their differences in appearance. This trend continued as we validated using different years, and shows when we

average all the accuracy measures across the four years, which only shows a difference of a few thousandths. The same pattern also shows for both accuracy and sensitivity, although both of these were slightly higher when validated against 2017 than when they were averaged. Since the averages are quite close to the 2017 validation, we are confident that our model would work well on future years where the data is most relevant. One other note about specificity: since our model attempted to correctly predict all stars, but did not care about those who were not qualified, we chose to deliberately not include it.

An interesting insight about the variables used in the two trees came up as well. Five variables were originally used and are present in both versions, however, we also wanted to investigate the other six that were unique to the manually pruned model. After further investigation, we realized that the six unique variables all are transformations of other variables used in the dataset. For example, Value over Replacement takes into account minutes played and box plus/minus. These transformation variables all were used to further refine the model, albeit only slightly.

### *Naïve Bayes*

We only had numerical variables in our data set, which meant that if we wanted to use Naïve Bayes, we had to bin some variables manually. Based on our basketball knowledge and online research, we set ranges for statistics of great players, good players and average players using the for loop function. After that, we checked the summary of the data set and found out the variables we changed all turned into characters. So we used as.factor function to turn the

variables into factors in order to use them in the Naïve Bayes model. The first thing we did after we finished our data processing was to pick some variables we thought would be useful in predicting future all-stars and binned all of them. We used Games Played, Games Started, Player Efficiency Rate, True Shooting Percentage, Total Rebound Percentage, Assists, Steals, Blocks, Points, and Field Goal Attempts. For example, we assumed that if a player scored a lot of points, the player would more likely be selected as an all-star. Also if the player started a lot of games, he or she is one of the best players on the team. The model returned an accuracy of 0.9555, a sensitivity of 0.826, and an AUC of 0.832. For the second model, we used the top 5 significant variables in the classification tree and logistic regression. We used Win Share, Points, Usage per Game, Box Plus Minus, Player Efficiency Rating, True Shooting per Game, Total Rebound Percentage Per Game, Field Goals per Game, Field Goals, and Minutes Played. It marginally improved our model by increasing accuracy to 0.958, sensitivity to 0.852 and AUC to 0.894. Overall, Naïve Bayes performed decently against our other models.

***Validation Analysis***

Among our four models, KNN had the highest accuracy, 0.973 and low sensitivity, 0.521. However, sensitivity is more important to our model than accuracy because we were trying to predict the outliers. Only 25 all stars are selected out of around 600 players every year. We wanted the model to predict as many true positives as it can. Consequently, KNN could not be considered a good model for our project. Because of the low sensitivity, we did not look further into the calculation of AUC for KNN.

The updated version of logistic regression had the highest AUC. However, our unpruned tree had a higher sensitivity, which is valued higher than accuracy in this case. Naïve Bayes is not a bad model but it doesn't stand out compared to the other two. Our final solution would be to average the results from unpruned classification tree and updated logistic regression since sensitivity and AUC are equally important to our model.

### *Conclusion and Lessons Learned*

Throughout our long and tedious process of creating the optimal predictive model, we had our fair share of successes with an equal amount of failures. Being able to learn from those mistakes allowed for our group to progress and finally create a meaningful model. From the beginning, we realized that data preprocessing and preparation takes quite a bit of time. Having to go in and manually change our binary target variable based on outside research was quite time consuming, but we were able to make this process a bit more manageable by diving up the data and assigning each team member a particular set of years to research. Another insight we gained from data preprocessing is that not all models will return accurate results like those in class. Many of the datasets we worked with in class were previously studied or used for educational purposes making them ideal for returning concise results for the different models taught in class. When working with raw data from the web, there were a lot more factors that we had to consider. Based on our results for the KNN model in particular, we realized that standardizing all of the variables may not always be useful like we learned in class. When players are considered for being an All-Star based on their qualifications, some of their statistics have more weight than others. For instance, 3-point percentages are considered more important than 2-point percentages. Given this, it makes sense why our KNN model was not returning optimal results

because of KNN's innate variable standardization. Finally, we realized that working with a copious amount of numerical variables turned out to be beneficial once we were able to bin them. Although our dataset seemed quite daunting at first, being able to categorize these numbers in bins allowed for us to work with a variety of models.

In regards to who our model could be useful for, we thought of a number of practical applications. First, our model could ideally be used for the NBA. The NBA essentially makes the decisions as to who will be in the next set of All-Stars, and we feel our model could be beneficial in helping out in this process by pointing them in the right direction and saying which player could be considered. Additionally, we felt that our model could also be beneficial to some of the sports analysts who work at media companies such as ESPN. Sports analysts are always set with the task for making predictions about the NBA season, and we feel that our model could aid in this predictive process as well. Finally, our model could be useful for the sports betting industry. Every year, large sums of money are invested within this industry, and our model could help participants feel comfortable with their data driven decisions.

"We pledge our honor that we have neither received nor provided unauthorized assistance during the completion of this work."

-Alex Damiecki, Nathan Nakahara, Mukul Tekchandani, Grace Wang, John Wen