**NBA Roster Construction with STAR: SRS Team Adjusted Rating**

Austin Hartman, Vish Narayan

Track: Basketball

ID: 7V7B0BZ38

1. **Abstract**
   1. **Introduction:**

The question of ideal roster construction strategy has long plagued NBA front offices. Should executives prioritize stars or depth? Talent or fit? In past studies of similar topics, researchers have made use of K-Means clustering to classify players and predict lineup success. However, simple clustering does not provide a granular view of a player’s skills, which is necessary for strategic roster management. As such, a more nuanced version of player summarization would prove useful in predicting team success.

* 1. **Methods:**

In order to summarize player skillsets, we made use of PCA and SVD compression on relevant per-game and advanced statistics (gathered from Basketball-Reference.com). Along with K-Means clustering of the same data, this technique provided archetypes and 8-factor summary statistics for each NBA player dating back to 1985. These summary statistics were combined to form vectors of player information for each team year and were run through a random-forest decision tree to predict team SRS. This unique combination of player statistical profiles resulted in our model: STAR (SRS Team Adjusted Rating). Once completed, the model provides analysts with multiple methods of analyzing ideal roster construction:

• Analyzing the importance of each factor in the decision tree by means of the Increase in Node Purity metric (INC) to establish new logical heuristics

• The capability to build teams from scratch and predict their SRS

* 1. **Results:**

The combination of individual player summary statistics proved useful in predicting overall team success, as plotting historical team SRS values (training data) against the model’s predictions resulted in a very strong R2 of 0.9853 and a residual standard error of 0.55 SRS points, while the testing data (2019-20 season) was also predicted quite accurately with a R2 of 0.9439 and a residual standard error of 1.19 SRS points. Analysis of the resulting INC scores revealed that the top four player VORP values for a team each had scores of 2900 or above, while the fifth player’s VORP INC was only 1537, the sixth 895, and so on with continued decline in importance (see figure). Additionally, all PCA condensed player-summary factors had INC scores under 100, significantly less than the VORP and BPM components. Finally, utilizing the model for team building and prediction results in an infinite sample space of potential combinations and results.

* 1. **Conclusion:**

Analysis of the INC scores of each factor in the decision tree allows us to build basic heuristics such as a team’s core being defined by its top four players, or talent/impact trumping fit/role. Furthermore, STAR’s capability to build teams from scratch and then predict their SRS provides more room for insight and exploration. STAR allows teams to experiment with adding and removing specific players and their corresponding statistical footprints. This ability to explore player and team combinations in an unbounded manner opens doors to creative new roster construction methods, less reliant on subjective beliefs and instead backed by machine learning and historical trends.

1. **Introduction**

In order to judge the composition of an NBA roster effectively, one must first be able to judge and summarize the abilities of a single player. However, making use of 38 statistics for every player on every roster in the NBA is overly complex, and would result in messy, incomprehensible data. As such, we made use of PCA reduction and K-Means clustering in order to provide brief, accurate summary reports detailing a player’s contributions. K-Means clustering is an unsupervised machine learning method whose goal is to classify data points into K unique clusters. After presenting the model with a chosen value of K, the

1. **Clustering and Five Factors**

2.1 Cluster Creations

In order to judge the composition of an NBA roster effectively, one must first be able to judge and summarize the abilities of a single player. However, making use of 38 statistics for every player on every roster in the NBA is overly complex, and would result in messy, incomprehensible data. As such, we made use of PCA reduction and K-Means clustering in order to provide brief, accurate summary reports detailing a player’s contributions. K-Means clustering is an unsupervised machine learning method whose goal is to classify data points into K unique clusters. Once the value of K has been selected, the k-means method continually updates the cluster centroids and resulting cluster assignments in order to minimize the within-cluster variances. Each point is assigned a cluster label according to the cluster centroid to which it has the shortest Euclidian distance. Often, K-Means clustering is combined with PCA reduction, in which the 38 statistics we chose as variables can be reduced to a lower dimensionality while still retaining most of the distinguishing information. In order to emphasize information retention, we attempted PCA reductions at each dimensionality from 2 to 37, and then chose to reduce the data to the smallest dimensionality that would still retain 95% of the variance of the original data (Table 1). As a result, our data was condensed to 15 principal components, which were then provided to the K-Means algorithm for cluster assignment.

Table1

|  |  |
| --- | --- |
| Dimensions | Variance Explained |
| 2 | 55.7% |
| 3 | 64.1% |
| 4 | 69.9% |
| 5 | 74.7% |
| 6 | 78% |
| 7 | 81% |
| 8 | 83.6% |
| 9 | 86% |
| 10 | 88.1% |
| 11 | 90% |
| 12 | 91.7% |
| 13 | 92.9% |
| 14 | 94.1% |
| 15 | 95.1% |

The next step was deciding on the optimal value of K to ensure accurate and useful clustering. While simple data sets may have distinct clusters when represented visually, allowing the researcher to easily determine the optimal value of K, NBA player statistics proved to be quite diverse. Although only 55.7% of the variance can be modeled within 2 dimensions and 64.1% within 3 dimensions, the visual representations of the PCA condensed data in these dimensions show no signs of a clear K value (Figures 2 and 3). As such, it became necessary to utilize another common method of K selection: inertia score analysis. In essence, inertia scores range from 0 to infinity and measure the distance between points within a cluster, meaning lower scores represent tighter and more useful clusters. Inertia scores can be used to select the optimal K value by performing PCA reduction at each of the possible dimensionalities and graphing the inertia scores at each dimensionality level. Then, the optimal point is chosen at the elbow point, a term used to represent the point at which the inertia scores begin to slow in their descent and the graph becomes nearly linear descent rather than polynomial/exponential. Within our data reduction analysis, the elbow point appeared around a dimensionality of 7, indicating an optimal clustering with 7 unique groups of players (Figure 4). This clustering is shown (in two dimensions for visibility) in Figure 5.

Figure 2

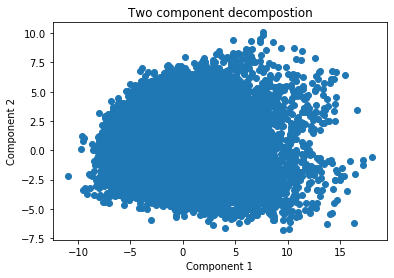


Figure 3

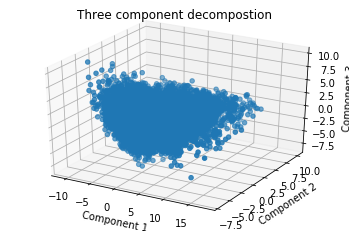


Figure 4

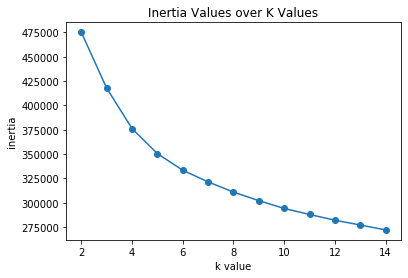
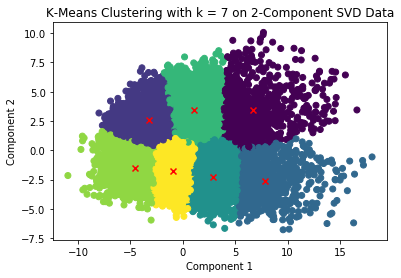


Figure 5



2.2. Five Factor Creation

In past similar papers, the researchers decided that some form of clustering (K-Means or Gaussian Mixture Model-based Clustering) was enough to provide accurate representations of a player’s skillset. However, we believed that just a single label (or probability distribution between cluster labels) did not provide an accurate, fine-grained description of a player’s statistical footprint. Of course, the overruling principle in these decisions was the idea that a model can only provide so much detail before becoming overly complex, resulting in difficulty of use and comprehension. However, we believed that responding with the oversimplicity of a single classification system was too extreme of a measure taken in the opposite direction. To balance model complexity with detail, in addition to a cluster assignment based on 15-dimension PCA reduction, we performed a second PCA reduction without clustering. The goal of this second reduction was to provide a multi-component summary report which could explain large amounts of variance in the data without losing its simplistic nature. As such, we revisited the chart shown in Figure 1 to determine the optimal dimensionality value for our reduced data set. In the end, we settled on a dimensionality of 5 for our newly condensed data, as 5 dimensions could explain nearly 75% of the variance in the original data, but was still small enough to be easily digestible when attempting to extract meaning from inherently nameless PCA component vectors. In addition, 5 is a simple rounded number that most of the population will intuitively latch onto, once again aiding in comprehensibility and user-friendliness.

2.3. Interpreting meaning of Five Factors

As noted earlier, while PCA reduction is incredibly useful for scaling down data to manageable sizes, it comes with the drawback of losing comprehensibility and meaning, as each of the new Principal components is a weighted combination of the original high-dimension variables with no inherent name. However, because we deemed user-understanding of the inner workings of the STAR model to be highly important, we analyzed each of the five factors to understand what traits and/or combinations of traits the factor represented. While some factors showed very strong correlations with certain variables from the original data (Figure 5), others seemed to have no relationship at all (Figure 6).

Figure 5

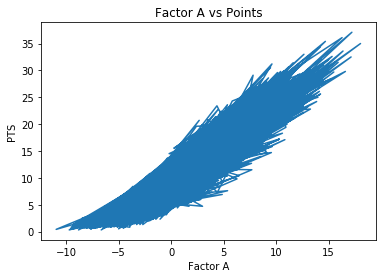
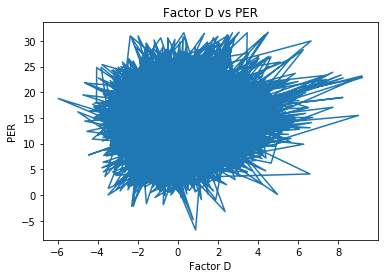


Figure 6



To analyze the extent of each of these relationships, we made use of a correlation matrix between the new Five Factors and all of the original numeric data (Figure 7). By identifying strongly correlated factors (both positive and negative), we were able to label each of the Five Factors with relevant real-world skills (Table 2). However, it is important to note that the additional variance explained by each factor decreases as more components are added. As such, it is not surprising to see many stronger correlations in Factor A than Factor E.

Figure 7

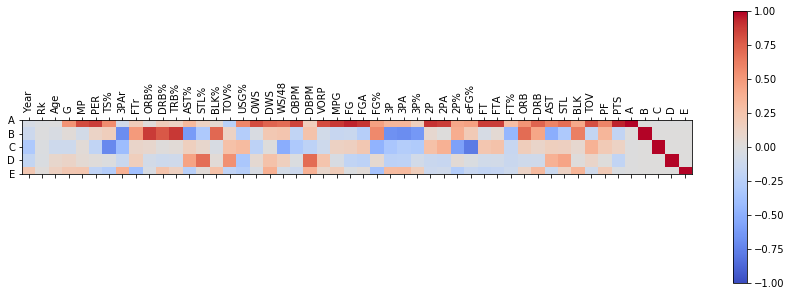


Table 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Original Component | Relevant Positive Correlations | Relevant Negative Correlations | Potential Skills | New Factor Name |
| A | PTS (0.944)  PER (0.873)  OBPM (0.844) | TOV% (-0.24) | Scoring  Playmaking | Offensive Talent (OT) |
| B | ORB% (0.877)  DRB% (0.781)  BLK% (0.718) | 3PA (-0.691)  3P% (-0.629)  AST% (-0.6109) | Rebounding  Interior Defense  Lack Shooting  Lack Playmaking | Interior Focus  (IF) |
| C | USG% (0.307)  2PA (0.368)  TOV% (0.356) | TS% (-0.707)  2P% (-0.585)  eFG% (-0.781) | Inefficiency | Inefficiency  (I) |
| D | STL% (0.695)  DBPM (0.703) | USG% (-0.32) | Perimeter Defense | Perimeter Defense  (PD) |
| E | BLK% (0.271)  DBPM% (0.37)  3P (0.311)  3PA (0.318) | FTr (-0.389) | Interior Defense  Three Point Shooting  Low Free Throws | Unicorn  (U) |

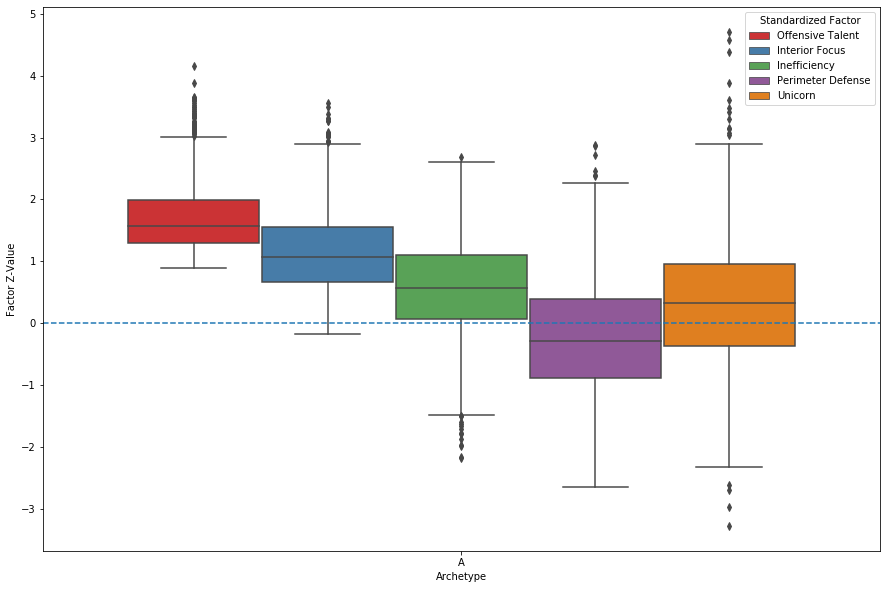
2.4. Interpreting the 7 Archetypes (Clusters)

After adding descriptive labels to each of the five factors, we decided to do the same for each of the 7 player archetypes identified by the K-Means clustering. To perform this exploratory data analysis, we compared the distributions of each of the five factors between each of the 7 archtypes. In order to make give the values more meaning, each of the five factors was scaled to have a mean of 0 and standard deviation of 1. Additionally, we viewed the raw clustering data, sorted by distance from the cluster centroid to see examples of NBA players that fit the archetype well. For example, in archetype A, the distribution of the five factors can be seen in Table 3 and Figure 9.

Table 3

|  |  |  |  |
| --- | --- | --- | --- |
| Factor | First Quartile Z-Score (Percentile) | Median Z-Score (Percentile) | Third Quartile Z-Score (Percentile) |
| Offensive Talent | 1.30 (90.2%) | 1.57 (94.1%) | 1.98 (97.6%) |
| Interior Focus | 0.66 (74.4%) | 1.07 (85.7%) | 1.56 (94.1%) |
| Inefficiency | 0.06 (52.4%) | 0.57 (71.7%) | 1.09 (86.3%) |
| Perimeter Defense | -0.89 (18.6%) | -0.29 (38.7%) | 0.38 (65%) |
| Unicorn | -0.36 (35.8%) | 0.32 (62.7%) | 0.96 (83.2%) |

Figure 9



These results indicate that players in archetype A very good to elite offensive scorers, with high interior focus, medium to high inefficiency and poor perimeter defense. The unicorn factor seems to have a wide range of outcomes, which makes sense given the uniqueness of its skill combinations.

Then, by looking at the players in this archetype (viewing only 2019 players for ease of comparison), sorted by their distance from the archetype centroid, we can confirm the previous findings given our existing knowledge of the players and their skillsets. The 5 most representative players for archetype A in 2019 are displayed in Table 4.

Table 4

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Player | Tm | Year | Pos | Archetype | Dist | OT | IF | I | PD | U |
| Domantas Sabonis | IND | 2019 | C | A | 3.62 | 1.90 | 1.15 | 0.20 | -0.13 | -0.05 |
| Bam Adebayo | MIA | 2019 | PF | A | 3.70 | 1.82 | 1.45 | 0.02 | 1.30 | -0.25 |
| Montrezl Harrell | LAC | 2019 | C | A | 3.81 | 1.57 | 1.35 | -0.43 | -0.70 | -1.31 |
| Julius Randle | NYK | 2019 | PF | A | 4.94 | 1.38 | 0.21 | 1.03 | -1.54 | 0.08 |
| LaMarcus Aldridge | SAS | 2019 | C | A | 5.06 | 1.29 | 0.31 | -0.40 | -1.64 | 0.76 |

Given the distributions of the five factors for the entire archetype, and the most representative players according to the distance metric, we decided to name the archetype “Skilled Big.” This process was then repeated for the other 6 archetypes (Figure 10 and Table 5).

Figure 10

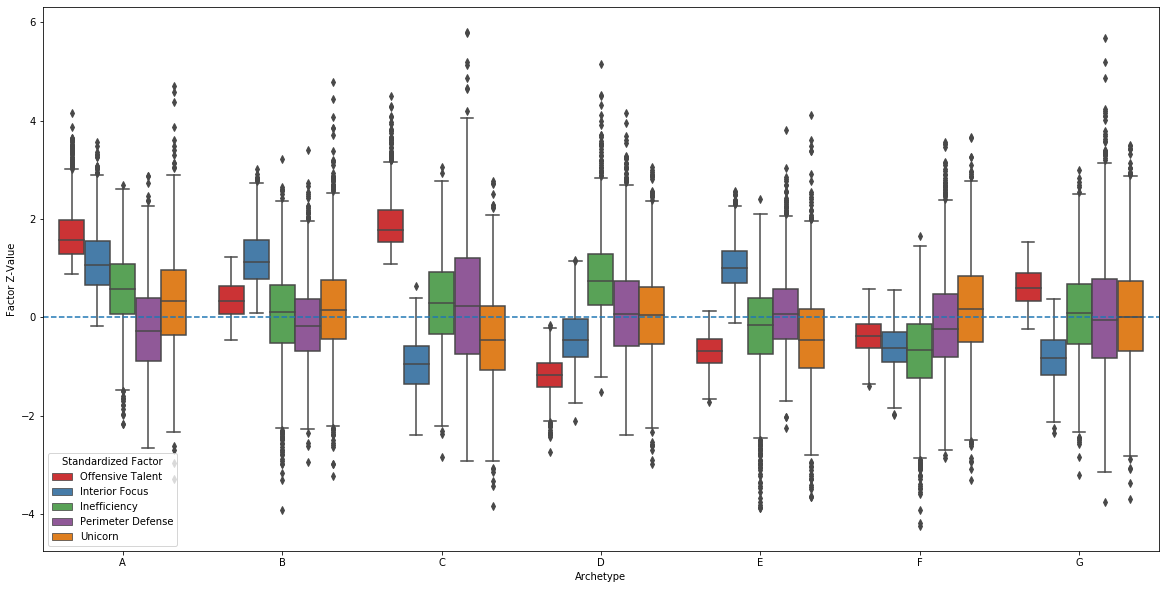
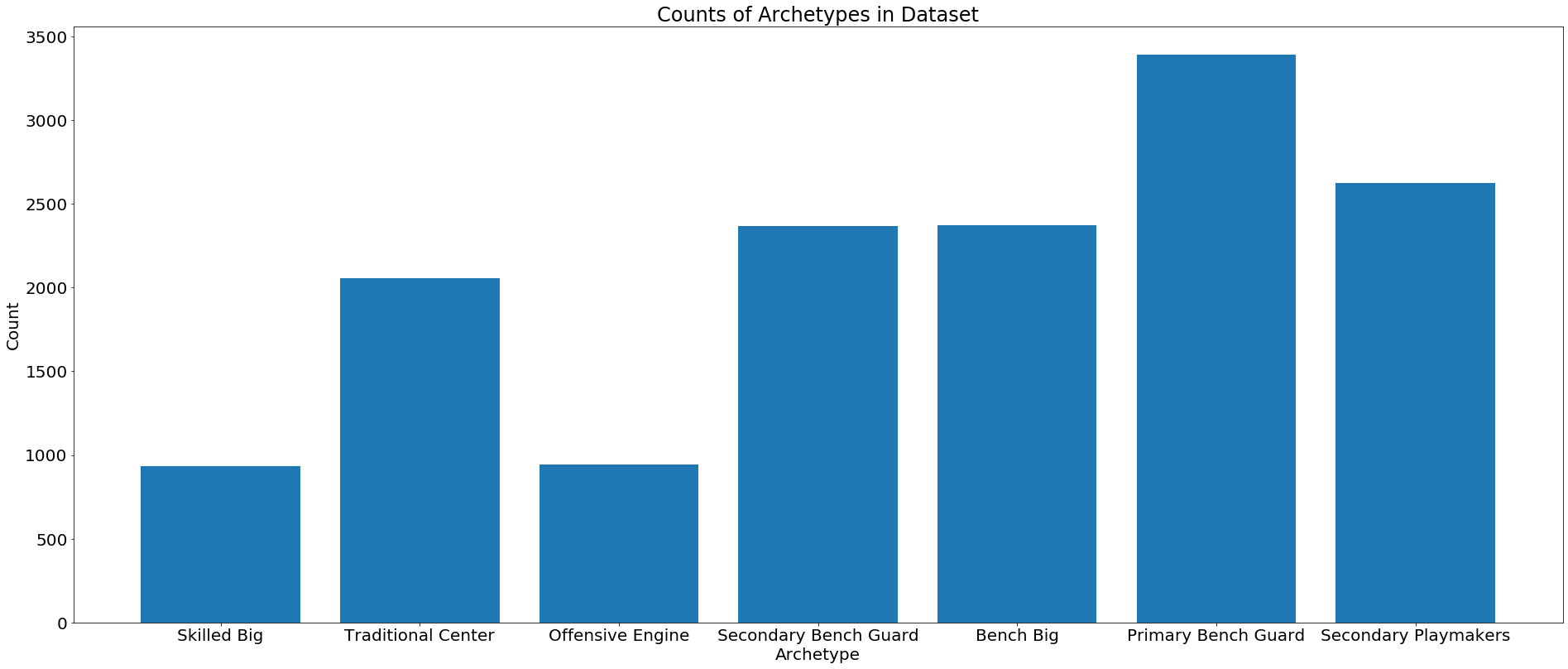


Table 5

|  |  |  |  |
| --- | --- | --- | --- |
| Original Archetype | Important Skills | Example Players | New Archetype Name |
| A | +OT,IF | Domantas Sabonis  Bam Adebayo | Skilled Big |
| B | +IF | Cody Zeller  Steven Adams | Traditional Center |
| C | +OT  -IF | Chris Paul  Lebron James | Offensive Engine |
| D | +I  -OT, IF | PJ Dozier  Carsen Edwards | Secondary Bench Guard |
| E | +IF  -OT | Thon Maker  Jordan Bell | Bench Big |
| F | -OT,IF,I | Cory Joseph  Austin Rivers | Primary Bench Guard |
| G | +OT  -IF | Josh Richardson  Caris Levert | Secondary Playmakers |

With each of the archetypes cemented, the distribution of archetypes throughout the history of our dataset (year range) is seen below (Figure 11). While it may be interesting to explore how the NBA landscape has evolved over time, and analyze the distribution of archetypes for each year in our dataset, this is not the primary goal of our research. However, the potential for additional research in this domain exists.

Figure 11



1. **Random Forest Implementation**
   1. Introduction

After building our 7 clusters and 5 Factor summaries for every player in the league, we wanted to see how combinations of players would impact team success as a whole. The next step in the process was to create the STAR model: a mapping from player summary statistics to team SRS.

* 1. Data
  2. Roster Representation

Now that we have our 11 total factors to describe a player, the question becomes how to choose the players that represent a team in order to predict its SRS. There are numerous factors to consider here, including trades, signings and waivings, injuries, and a player’s place in the normal rotation. First, we decided to prioritize players that had played the most minutes with a team, as these are the players that represented the team throughout the year and are most definitive of the team’s identity. While this means a traded player who is no longer on the team is used to represent the team in the training data, we felt this was most representative of the team’s SRS given the minutes the player played for the team. Additionally we decided to include only the top 10 players from this group to define the overall roster. We reached this conclusion due to the average size of NBA rotations, which was estimated to be under 13.5 players for the regular season and under 9.4 players for the postseason (Bleacher Report). Because 10 is once again a simple round number, it was chosen for the sake of simplicity. As such, the final representation of a roster for any given year was a vector of length 113 (11 factors for 10 players, the team name, the team year, and the team SRS). To standardize the representation for improved learning, the player summaries were sorted by VORP in descending order.

* 1. Model Choice

Initially, we attempted to build the STAR model using a neural network, given their history of success in identifying hidden relationships between input variables. However, we immediately ran into issues, as neural networks are incapable of robustly handling categorical data such as the player archetype. This led us towards a more robust model that can handle both categorical and numeric data: decision trees. Decision trees offer similar strengths to neural networks, as they are easily capable of representing relationships within the input data. Additionally, they are capable of very high accuracy scores, as each node of the tree is built by selecting the attribute and cutoff that maximizes the information gain for the resulting branches. However, decision trees are often prone to overfitting to the training data, especially when they are not pruned after training, leading to a model that is not capable of generalizing to unseen data. In order to combat this issue, we made use of a Random Forest classifier, a common machine learning bagging algorithm which builds many separate decision trees on subsets of the training data. Then, anytime a new example is classified, it is presented to each of the decision trees in the forest, and the result is a voted average of the resulting decisions. The idea behind bagging algorithms such as random forests is to reduce variance in the model by providing accurate input from many different sources. While the new model will not necessarily reduce bias (i.e. increase accuracy on the training data), it reduces the tendency of the original single decision tree to overfit.

* 1. Model Training

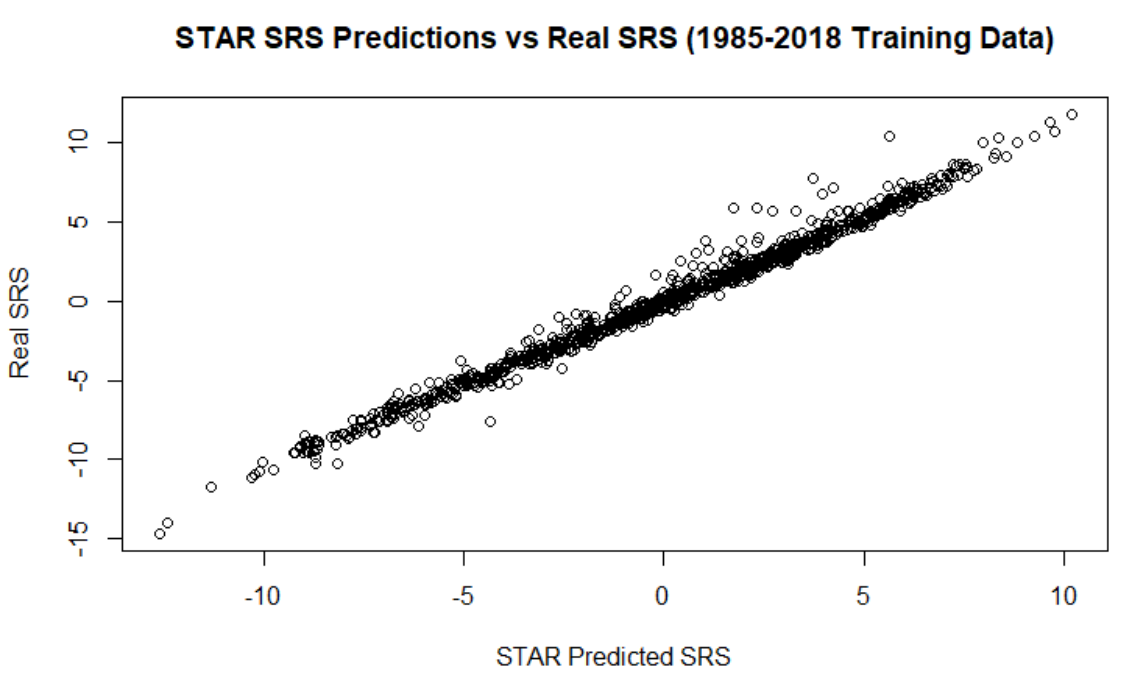
The random forest algorithm was provided with a dataset composed of team roster vectors: one vector for each NBA team year dating back to <year>.

* 1. Results
     1. Training on all data up to 2019

Our very first method of testing the effectiveness of the STAR model was to train on all the data we had available up to but not including the current season, and then validating on the single unseen year, as this is likely to be the most common method of practical application for the model. Because our data is not classification, we cannot present accuracy scores; however, by analyzing the correlation between our predicted values and true SRS values, we can evaluate the model’s performance.

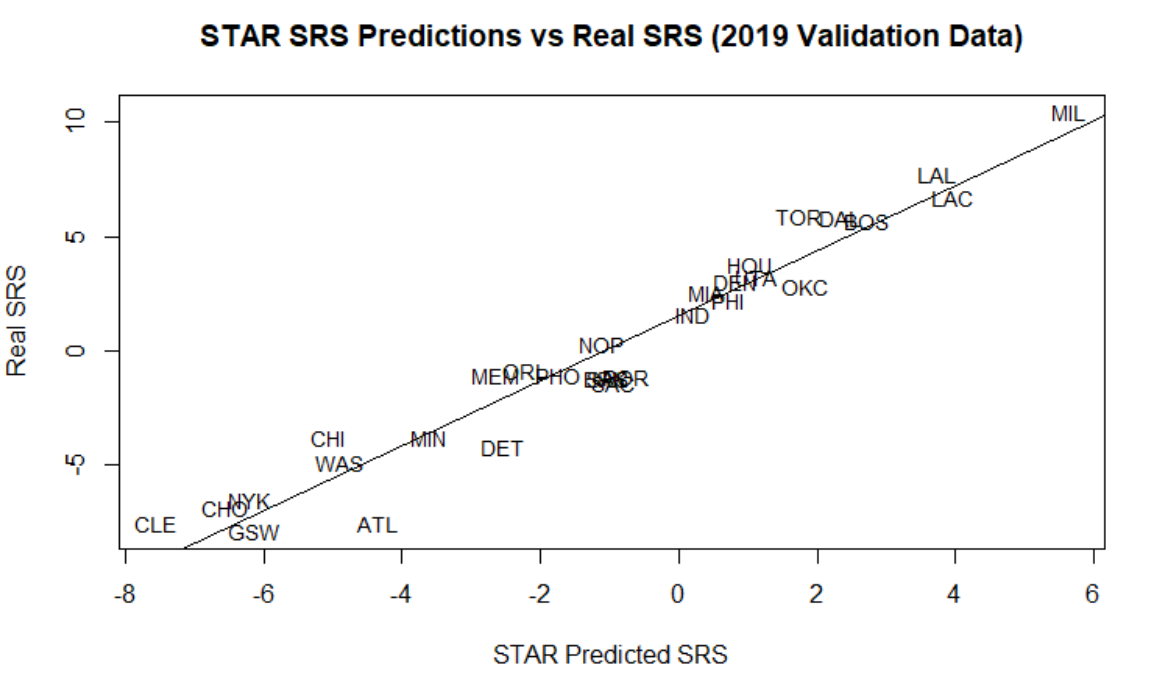
The predicted SRS values for examples training data (1985-2018) had an R2 of 0.9854 with the true SRS values, an incredibly strong correlation. The resulting scatterplot can be seen in Figure 12.

Figure 12



The predicted SRS values for the validation data (2019) had an R2 of 0.9429 with the true SRS values, also a very strong correlation. The resulting scatterplot can be seen below in Figure 13. Note that the predicted SRS in this set are all scaled down due to the reduced season length in 2019 caused by the Coronavirus pandemic.

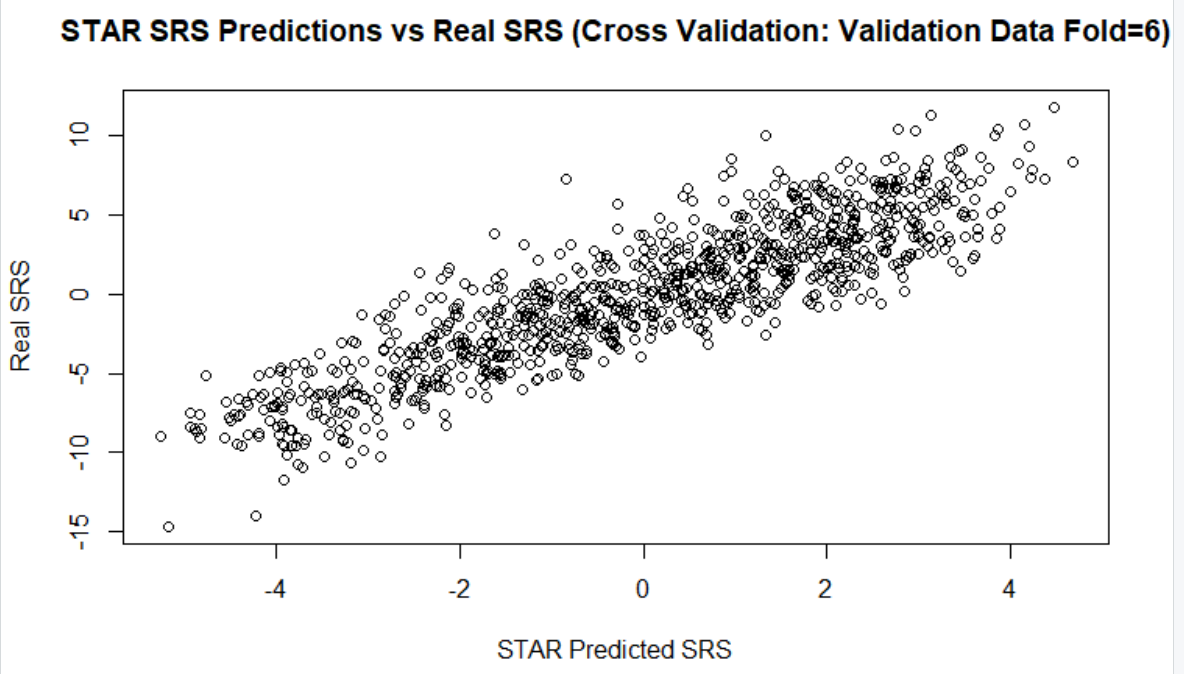
Figure 13



* + 1. K-Fold Cross Validation

Next we performed K-Fold Cross Validation to ensure that the model was able to generalize well to different sets of unseen data. After choosing K to be 6 (normal K values in 5-10 range and 6 allowed us to split the data evenly), we separated the original dataset into 6 subsets, and then performed training and validation 6 separate times, each time withholding one subset from the training data in order to be used for validation. Over the 6 folds, the average training R2 was 0.98 and the average validation R2was 0.772, still a very strong correlation. The scatterplot of one of the six folds can be seen below in Figure 14. Due to the use of the random forest algorithm, we can reasonably assume the difference in training and validation scores is not due to overfitting. While no further testing was performed, we have hypothesized that this difference was observed due to training and testing on such a wide time period, in which the game of basketball has evolved significantly. The style of play in 1985 is vastly different from that of the early 2000s, which is vastly different from today’s game. As such, there may be hidden factors regarding the evolution of the game that have not been accounted for within the model.

Figure 14



**References**

[1] Reference #1 cited using any mainstream citation style (e.g. APA, MLA).

[2] Reference #2

.

.

.

[n] Reference #n

**Appendix**

An appendix is not required, but if you have one please include it here.