Estimating NBA Player Career Accomplishments

Authors:
Sameer Sapre
Austin Gongora

Introduction

At the culmination of each NBA season, there is almost always a debate over the league's top performers across the whole season. Naturally, the debate among fans, journalists, media personalities, and players themselves culminates in the annual selection of the All-NBA teams. The All-NBA teams are official, league accepted selections of the best NBA players for the given year, i.e. the All-NBA teams for 2018-19 are the selections for the top performers for the 2018-19 season. There is, however, no defined criteria for a player to be deemed worthy of receiving an All-NBA selection. It is dependent completely on what the voters (writers selected by the NBA) feel. However, these awards are what separate the superstars from the rest of the NBA. So defining a criteria for this process would give us insight into what it takes to be a superstar in the NBA. A data-driven approach to this has been done before, specifically using a neural network to estimate the probability of a player winning an award for the most recent year[1]. However, this data has the potential to be used to forecast the careers of NBA players. Essentially, we are proposing a different methodology to research the question behind what makes an NBA player legendary. Our approach combines inferential statistics, natural language processing, and dimension reduction algorithms, to estimate how long current elite NBA could continue playing at their elite levels, and which younger players we expect to become superstars.

Data

1. Obtaining Performance Data

As expected, when writers select the best players for the season, they are selecting the best performers. Therefore, we had to get data on the performance of players for each season. To do this we scraped statistics off of the site Basketball-Reference.com. We scraped multiple tables, those containing the general statistics that are often mentioned on television and in the media, i.e) points per game, assists, rebounds, Minutes played. We also scraped more advanced statistics like BPM, Win Shares, value over replacement, and more. Finally, we gathered data on where players were selected in the NBA draft, the annual process in which players enter the league via college or high school. Often you'll see the most talented players are drafted towards the top of the draft and are therefore more likely to become elite NBA players. All the statistics and draft data were collected between 1980-2019 since the three-point line was first introduced in 1980. Data from 2020 is not included since the season was suspended. To scrape

1

Boger, Tal. "Using Machine Learning to Predict the 2019 MVP and All-NBA Teams: End of Season Predictions." *Dribble Analytics*, 12 Apr. 2019, dribbleanalytics.blog/2019/04/ml-mvp-all-nba-predict-2019/

data from Basketball-Reference, we used the BeautifulSoup library in Python. The code can be found in the Basketball-Reference Scrape.ipynb and NBA Draft Scrape.ipynb.

2. Motivation to extract NBA Articles

An NBA player's statistics determines how good an NBA basketball player is and the numbers are essentially irrefutable by any fan or sports analyst. Sports journalists heavily rely on these statistics to convey their perspective of the league and how they choose to word their stories. We are working under the assumption that ESPN NBA journalists write about the hottest news and updates for the league at that present time. If so, would it be possible to process the language of an article to predict the following All-NBA team? The idea would use text mining and natural language processing techniques to create a count of players and the number of times they were mentioned.

Obtaining NBA Articles

The program *NBA_Scraper.ipynb* is a python file that scrapes relevant NBA articles from ESPN archives (http://www.espn.com/nba/news/archive). The required input is desired month and year. The program extracts {Title, Contents, Date Published} per article and saves the information as a pandas Dataframe and converted to a usable *.csv file.* The main library used for web-scraping in the package *BeautifulSoup*. The average time to process a single month is approximately 4.5 minutes, but depending on the month it can take as long as 12 minutes or a little as 2 minutes to completely process. Roughly speaking, at first it took nearly an hour of scraping to get a single year of data. The program was written as a Jupyter Notebook, so the realization came that we can rely on Google Colab to assist with processing. Changing our run time session to use Google's advanced GPU hardware option cut the processing time of 60 minutes to about 20 minutes.

3. Analyzing NBA Articles

1. Motivation

The program *NBA-Analysis.ipynb* is a python file that analyzes unstructured text and works in parallel with *NBA_Scraper.* The original intention was to use article contents as our method of analysis, but after working closely with the NBA articles it was a realization that there would be a processing problem (due to the large number of articles), and by introducing so many different forms of unstructured documents it would require more dedication and time towards cleaning and normalizing. Although the contents of the articles were dense with information, oftentimes the titles served as an appropriate summary and would mention the one or two NBA players that the article focused on. Ultimately, as a team, we agreed to focus our attention on an article's title for further analysis.

2. Text-Preprocessing

To provide a sense of scale there are 8,752 articles for the years 2018 & 2019. The goal was to eliminate as much noise as possible by accurately counting player mentions per article title. A

good example is Lebron James. Article titles used a variety or expressions to reference the athlete such as [Lebron's, Lebron:, Lebron, Lebron;]. For the algorithm to function properly it must classify these instances as a single Lebron, rather than 4 versions of him.

To try and get complete names we used a Named-Entity Recognition from a python library called Spacy. Spacy allows us to use a neural network, trained on hundreds of thousands of articles, to help us find the full names of players. Once we iterated each article we could get access to, we were able to then extract them and create a count for each player for each season. For example, if Dwayne Wade was mentioned in articles 3 times during the year of 2004, he would have a "Mention" value of 3 for 2003. This "Mention" value actually became a prospective feature for our poisson model when we joined the data we gathered from the web-scraped articles with the player's corresponding performance statistics.

3. <u>Language Processing</u>

Once the titles are in the proper format, natural language processing is used to obtain the sentences Parts-of-Speech, PoS tagging. The python package *nltk* is used for PoS tagging. PoS tagging labels each word in a sentence with its appropriate part of speech.

```
s = "Lebron James is the best player in the league and is better than Michael Jordan"
nltk.tag.pos_tag(s.split())
```

```
[('Lebron', 'NNP'),
 ('James', 'NNP'),
 ('is', 'VBZ'),
 ('the', 'DT'),
 ('best', 'JJS'),
 ('player', 'NN'),
 ('in', 'IN'),
 ('the', 'DT'),
 ('league', 'NN'),
 ('and', 'CC'),
 ('is', 'VBZ'),
 ('better', 'JJR'),
 ('than', 'IN'),
 ('Michael', 'NNP'),
 ('Jordan', 'NNP')]
```

(Figure 1. PoS Tagging Example)

Each word is classified with a PoS. In our case we are only interested in the classifications *NNP & NNPS,* or respectively Proper Noun Singular/Proper Noun Plural. Logic is written to filter for these classifications and a frequency table is created for the counts by month and saved as a *.csv* file.

Methodology

1. Poisson Regression and Model Output

There have been a few attempts to use NBA statistics to predict or forecast future success for a given player, especially in the basketball analytics community[2]. Many revolve on just looking at incoming collegiate players that are entering the draft and predicting their statistics for their NBA careers. However, what about the players already in the NBA? Is there a way to use their performance at a certain point in their career to find out if they will be a superstar?

We decided to use a Poisson Regression to estimate the number of All-NBA selections a player could be expected to amass over the rest of his career. Since All-NBA selections are only given to the season's elite performers , they are a reliable measure of success and we concluded that it was the best variable to use as our response. In addition, since the variable was a count of

[2] Cheema, Ahmed. "Using Machine Learning to Predict Careers of 2019 NBA Draft Picks." *The Spax*, 9 Sept. 2019, www.thespax.com/nba/using-machine-learning-to-predict-careers-of-2019-nba-draft-picks/.

some event (being selected for All-NBA) occurring over a specific interval of time (the career span of an NBA player) we decided that it would be best to use a poisson regression model.

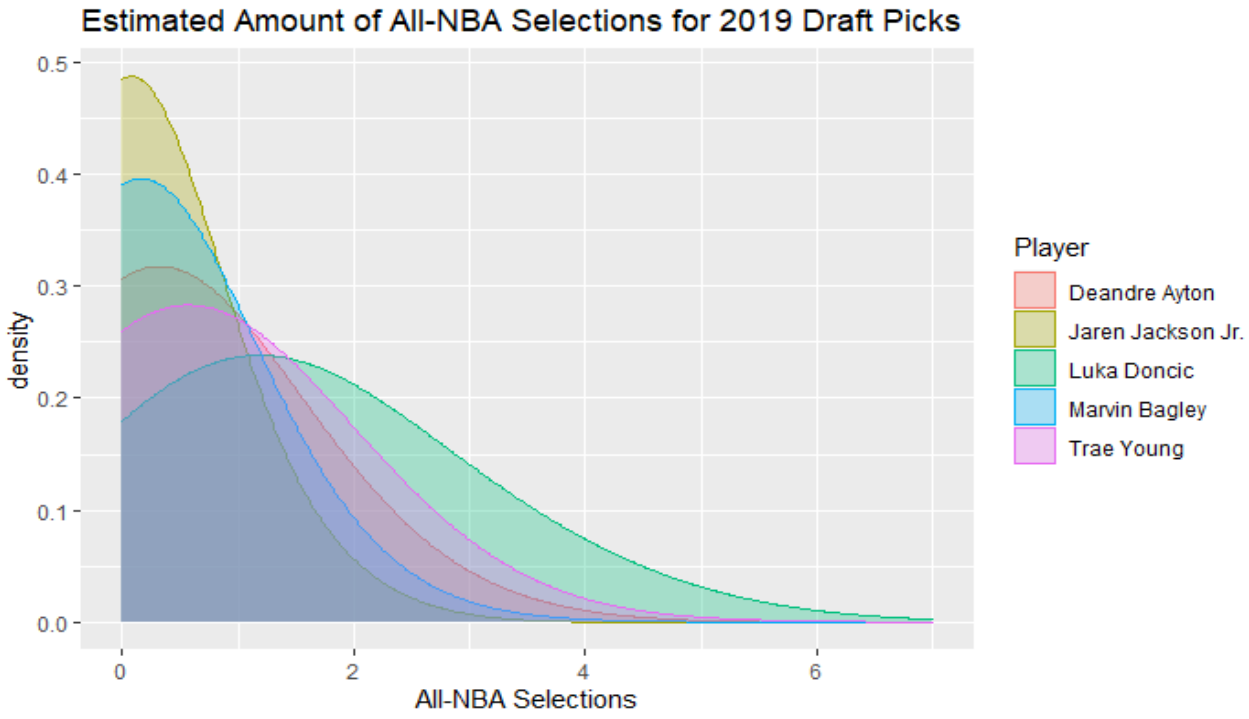
First, after obtaining the data from different sources on Basketball-Reference as well as from articles on ESPN, we had to go through an extensive process of engineering features. For example, we had to create a variable for All-NBA selections remaining in each player's career. In addition, we had to create a count variable for each player mentioned in ESPN articles by joining that data to our modeling data set. This feature creation can be found in more detail in the files 'More Preprocessing.ipynb', 'Player Name Extraction.ipynb', and 'Modeling.Rmd'. Then, we had to do more extensive cleaning like removing accents on names, correcting data quality issues, and splitting data. Finally, we decided to train our model on only retired players since current players are still eligible to be selected to an All-NBA team.

After the preprocessing was done, it was time to build the model. We had to choose between a group of around 60 features that could be highly correlated with each other. We attempted using a dimensionality reduction technique like t-SNE, as we will get into in further detail, but ended up using a correlation matrix and domain knowledge to build the model. Writers often cite points per game, assists, other "counting" stats, as well as newer advanced stats that better captures a player's impact. Since, we had gathered both advanced and traditional stats from Basketball-Reference.com we tried different combinations of advanced and traditional statistics. Eventually, we found a model in which every predictor was a significant predictor of All-NBA Selections remaining in a players career and had a residual deviance less than the degrees of freedom. The final features of the model were the players age, free throw rate, points per game, assists per game, total rebounds per game, three-point attempts, games played, position, box plus-minus, and blocks per game. The model is definitely not perfect, but when compared to the reduced model of the most basic stats, it fared much better.

During model evaluation, we also attempted cross validation similar to the class material on permutation, but the dataset was so large that it made it difficult to get through each sample in the training set.

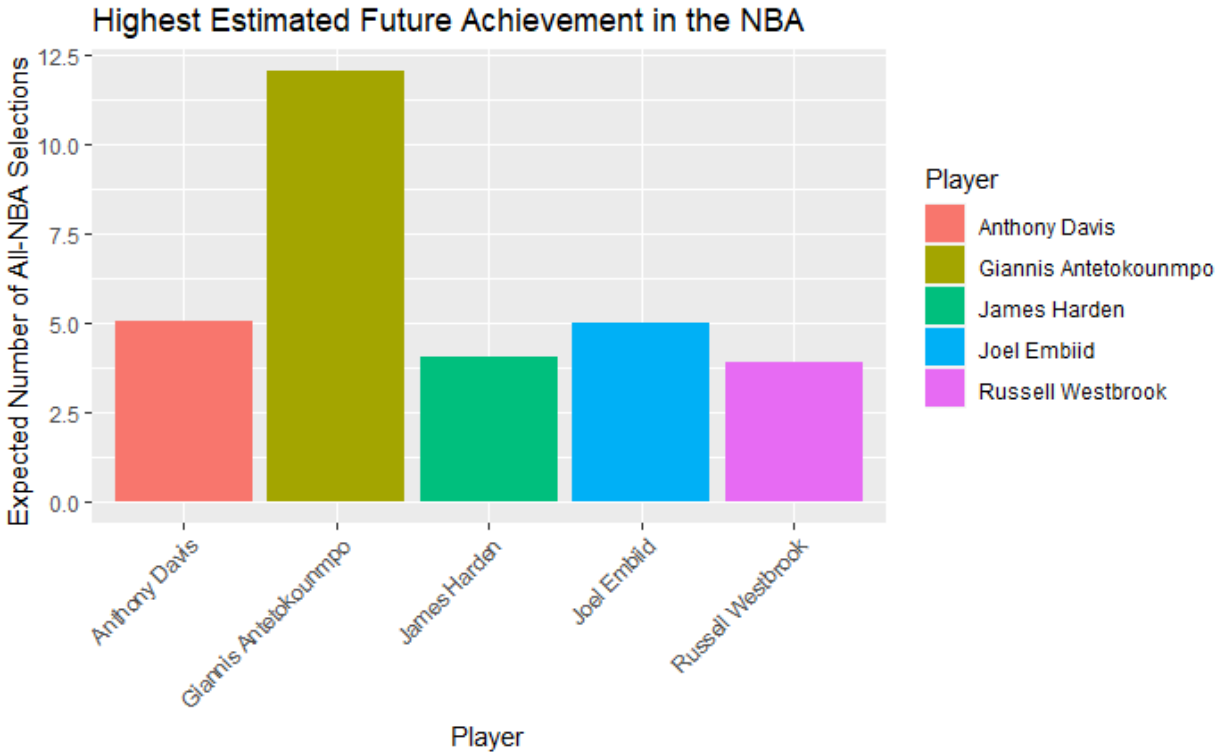
2. Model Output

Like many who try to predict the next wave of superstars in the NBA, we tried our hand at estimating the career success of the newest NBA draft picks. The following graph depicts the number of All-NBA selections we estimate each of the top 5 draft picks from 2018 to achieve.



Luka Doncic was rated most highly by the model as having the most future success.

In addition, out of all players during the 2018-19 season, Giannis Antetokounmpo is predicted to have the most future success by a wide margin.



Dimensionality Reduction

1. **TSNE**

T-distributed Stochastic Neighbor Embedding (T-SNE) is a machine learning algorithm for visualization. It is a nonlinear dimensionality reduction technique well-suited for embedding high-dimensional data for visualization in a low-dimensional space of two or three dimensions. Specifically, it models each high-dimensional object by a two- or three-dimensional point in such a way that similar objects are modeled by nearby points and dissimilar objects are modeled by distant points with high probability." The *jointplayers.csv* contains 53 variables and 16,594 observations, 49 of which are numeric variables. This would be an appropriate application to rely on dimension reduction to further understand any possible intricate relationships that exist in our data.

2. **Data Imputation**

Prior to applying any dimension reduction algorithms, it would be necessary to make sure that none of the required assumptions are violated. This includes accounting for [NA, NULL, or missing values] and converting leveled variables to a computationally useful form. We can account for NA values by counting the sum of NA's in our data frame.

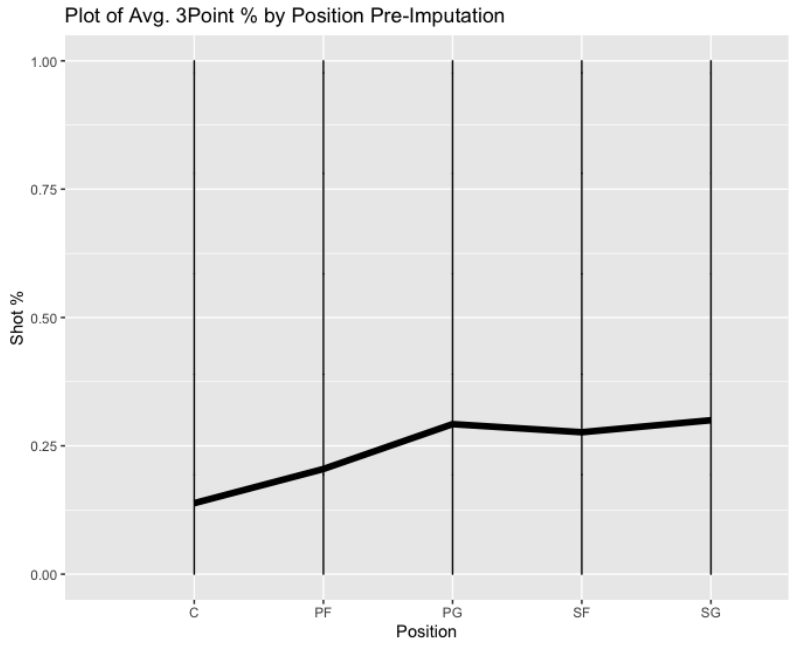
x = as.list(sapply(players, function(x) sum(is.na(x))))

z = as.data.frame(x[which(x>0)])

PER	TS.	X3PAr	FTr	ORB.	DRB.	TRB.	AST.	STL.	BLK.	TOV.	USG.	WS.48	X2P.	X3P.	FG.	FT.	GS	eFG.
3	44	51	51	3	3	3	3	3	3	36	3	3	82	2654	51	482	565	51

(*Fig. 2. – Count of NA values by variable)*

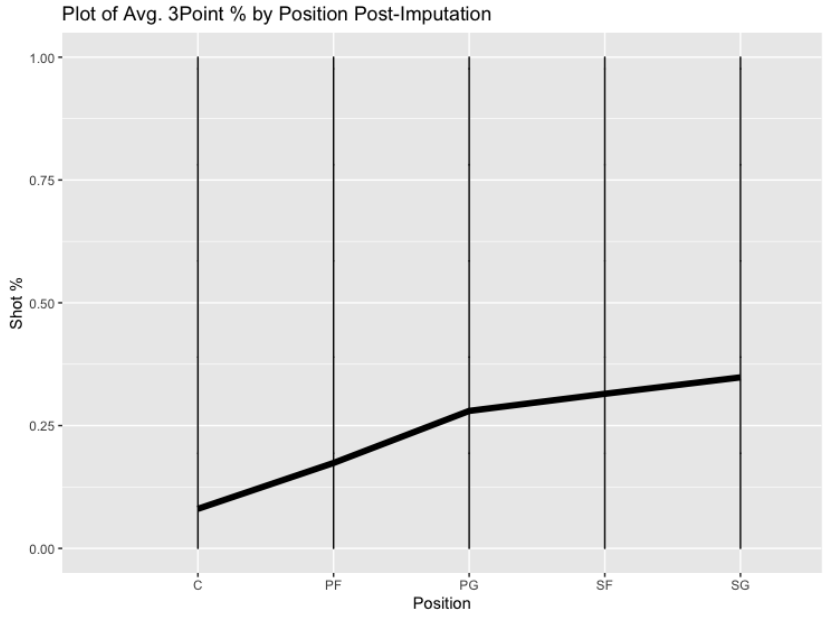
The 19 variables need to be dealt with accordingly and we propose to rely on data imputation methods to provide us with a better *ground truth* of our data. For example, feature X3P. (The players average 3-Point shot percentage) has 2,654 NA values. Because the imputation process is computationally expensive we randomly sampled 5,000 NBA players. The computation time to impute the dataset was over 2 hours.



(Figure 3. EDA of X3P. Pre-Imputation. N = 5,000)

The X3P. feature will be our test case to exemplify why data imputation is an important step for early analysis and more complex techniques like dimensionality reduction. To move forward with the imputation, we rely on the R library *Multivariate Imputation by Chained Equations* or *mice.* Specifically, we rely on the package because of its ability to create multiple imputations (replacement values) for multivariate missing data.

Once verifying the new data frame does not contain any missing values we test our X3P. variable for any new changes.



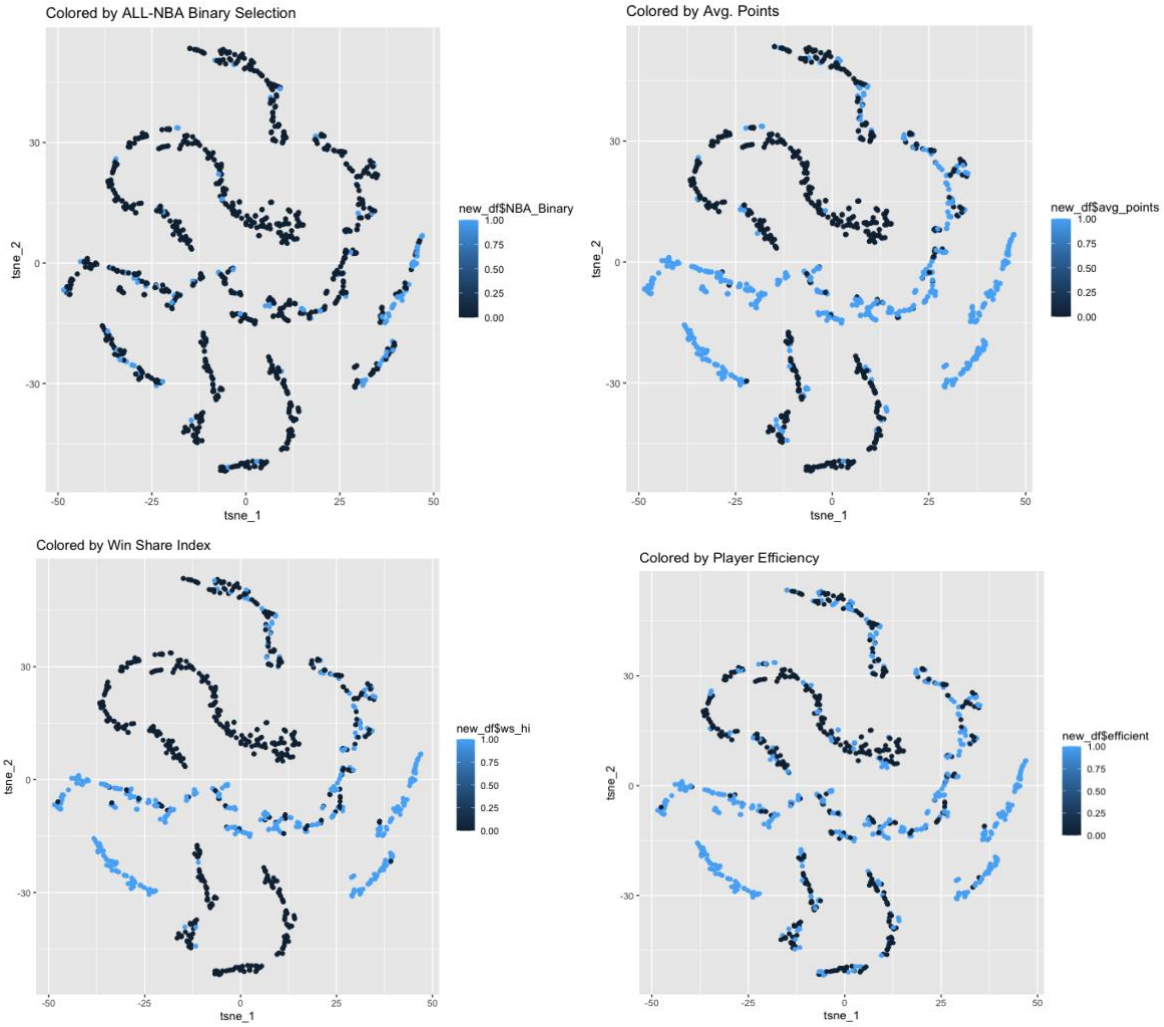
(Figure 4. EDA X3P. Post-Imputation, N = 5,000)

Comparing Figure 3 and Figure 4 we observe that the graphs differ. So much so that the positions now have a different mean with regards to average 3-Point shot percentage. Methods that simply remove or discard values such as [NA, NULL,0's] will not be as superior to methods that account for missing data by imputation.

3. Feature Selection

T-SNE was considered due in part that our dataset contained mostly numerical variables. Exactly 57 numerical features, some of which were engineered. Since neither of us have expert domain knowledge about basketball statistics the idea is to use T-SNE to provide some preliminary insight into features that are highly correlated with being picked to an ALL-NBA Team. To begin we created an additional 4 variables to assist in discovering commonalities. The features *{NBA_Binary, Avg_points, good_def, efficient, and ws_hi}* were added to our dataset. *NBA_Binary* assigns a 1 if a player has been selected at least once to an ALL-NBA team or a 0 if the condition does not hold. *Avg_points* assigns a 1 if a player has scored more points than the average for their position or a zero otherwise. Good_Def is also a binary variable except looks at a player's blocks, steals, and turnovers compared to their positions average. Efficient is an index that compares a players PER (Players Efficiency Rating) to the average PER by position. The variable *ws_hi* is an index that compares a player's *WS (Win Share), OWS (Offensive WIn Share), & DWS (Defensive Win Share)* respective to the mean for a player's position.

Rtsne was the primary R library used for running the t-sne algorithm. The process as a whole was time consuming and visualizing high dimensional data can sometimes be misleading or mysterious. The hyperparameters to run tsne is perplexity and number of iterations. It is common practice to choose a perplexity between 5 and 50, but as to my knowledge there is no sound way to optimize for this value. The first t-sne attempts focused on the entire imputed dataset, but after a lot of trial and error the visualizations that were being provided drew no clear connections. We decided to focus our attention by subsetting the data by positions and then running the T-SNE algorithm on the subsetted data. For the paper we will focus on the shooting guard position.



This approach was not as promising as we had hoped for. The figures represent a T SNE visualization of the shooting guard position with different features colored. Although not much is gained from the visualizations, they shed some light in how some of our features might be correlated and if we were to continue our work I am sure we could leverage this insight into constructing a better model.

Conclusion:

Though the model did provide some interesting insight as to what to expect to see out of NBA players in the years to come, there is still a great deal of room for improvement. The model heavily favors players who play Center, probably since three-point shooting was not nearly as common as it is today. Being able to take that into account would be very beneficial.

In addition, being able to gather a bit more data in the way of articles would be beneficial since we can get more than just perspectives from ESPN and, of course, provide the model with more data.

Finally, a better understanding of TSNE or dimension reduction in general would result in a more favorable model. As we mentioned earlier, we have around 60 features and using trial-and-error for feature selection may not be the optimal solution for this problem. While we did bring a good amount of domain knowledge to this project, dimensionality techniques like T-SNE or PCA could make up for anything we missed during our own evaluation of features. Dimensionality reduction would allow us to find relationships between variables that we didn't know existed, and allow us to simplify our model as much as possible. If we ever decide to bring in more data, for example, team results or media sentiment scores towards a player, this tool would be crucial.