**Game of Trades: Using simulation to get an edge in the NBA trade market**

Basketball

ID: 5643

1. **Introduction**

In this paper I present a method to evaluate the impact of trades for NBA teams. Assessing the potential impact of trades is a complex task as it affects not only the composition of the teams involved but also the rest of the league. For instance, the Kyrie Irving – Isaiah Thomas trade will change the number of wins Cleveland and Boston obtain but also has ripple effects and will alter the number of wins for any team playing these 2 teams on the regular season.

Briefly, this is the idea: first, I need to predict how good each player will be for the upcoming season and what will their playing time be. Second, I will estimate the offensive and defensive power of each team given the team roster and players’ minute share. Assuming the number of points a team scores follows a Normal distribution, with means the offensive and defensive powers and variance estimated from historical data. Once the density distribution is fully determined, I can calculate the probability of any matchup and thus the total number of wins in the regular season.

In the first part of the paper I will describe the proposed model and methodology. The second part is mostly devoted to examples and use cases. I developed an R Shiny dashboard[[1]](#footnote-1) that I use to interact with the model and the data. I recommend the reader to play along with it as you read the paper for a better experience. All the code is available on my Github account: asRodelgo/NBA.

1. **The Data**

The basis of every statistical analysis in the paper is players’ main measurable stats. My only source of data is basketball-refence.com. Although there are plenty of advanced statistics available on basketball-reference and other sites, my goal was to have as many years of consistent stats as possible for all NBA, European and College players so to have a more accurate learning from the data. For this purpose, I collect data from season 1979-1980 where 3-point stats were first recorded. In addition, I avoid stats that are not measurable from a player direct action, like box plus-minus. So, here are the list of stats I use[[2]](#footnote-2) throughout the paper:

Age, G, MP, FG, FGA, FG%, 3PM, 3PA, 3P%, 2PM, 2PA, 2P%, eFG%, FT, FTA, FT%, ORB, DRB, TRB, AST, STL, BLK, TOV, PF, PTS

These variables measure per game stats but my interest and the key to the whole analysis is in stats per minute. After adjusting from per game to per minute stats, I use the following nomenclature:

Age, FGPer, FG3Per, FG2Per, effFGPer, FTPer, effMin, effFG, effFGA, eff3PM, eff3PA, eff2PM, eff2PA, effFTM, effFTA, effORB, effDRB, effTRB, effAST, effSTL, effBLK, effTOV, effPF, effPTS

1. **The Model**

My objective is to predict how many games an NBA team will win in the regular season, given a set of players. Because wins depend on the capacity of teams to score points and to not receive them (Gelman, 2014)[2], I will instead predict the number of points an NBA team will score (offensive power) or allow (defensive power) on average during the regular season.

* 1. **Neural Network model**[[3]](#footnote-3)

For each offensive and defensive power I will use the output of a Neural Network. My inputs are the weighted average of players' projected per minute stats, where the weights are their share of minutes of play. Here are the steps I take:

1. Read historical players stats per game starting with season 1979-1980[[4]](#footnote-4)
2. Player name is my primary key so I need to differentiate players with the same name. I add a number after the name in ascending order (the younger the higher the number). Example: Tim Hardaway who played in the 90s vs. Tim Hardaway 2 (current NYK player)[[5]](#footnote-5).
3. Calculate stats per minute of play (effBLK, eff3PM, etc) for each player I of a total of 496[[6]](#footnote-6).

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

And effective minutes as MP (minutes played) over total possible minutes (48 minutes in 82 games):

|  |  |  |
| --- | --- | --- |
|  |  | (2) |

Finally, adjust effMin for each player relative to total minutes played by team Tm:

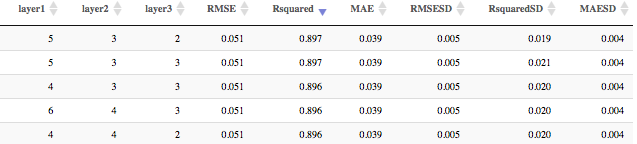
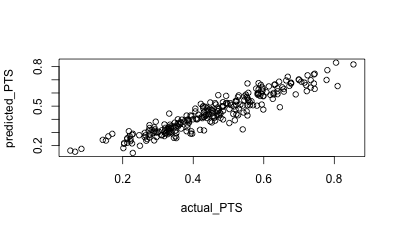
|  |  |  |
| --- | --- | --- |
|  |  | (3) |

1. The input vectors for the neural network are the weighted average of all stats per team per season. The weights are the effective minutes. The total size of the resulting input vector is 1,063.

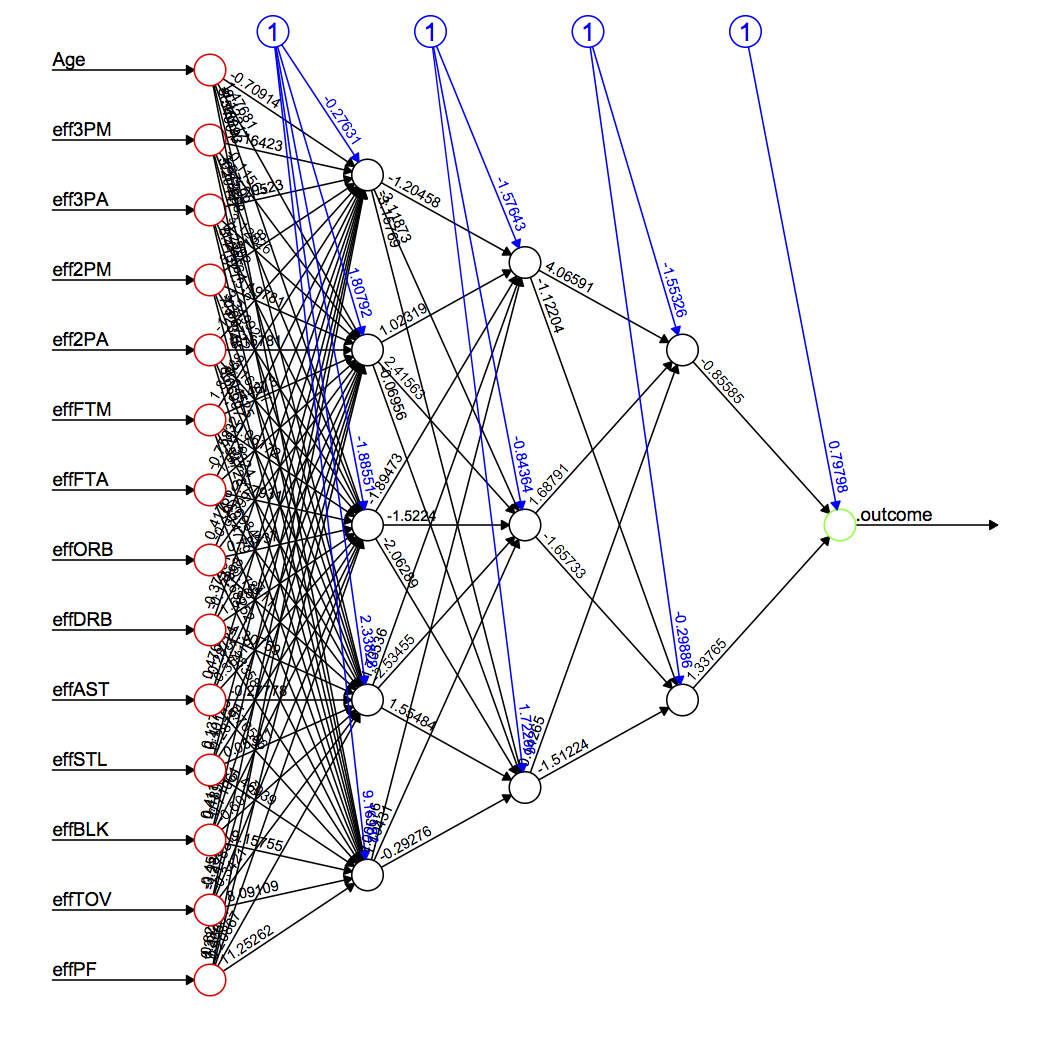
|  |  |  |
| --- | --- | --- |
|  |  | (4) |

1. Remove columns that have linear dependencies with others: FG, FGA, FG%, 3P%, 2P%, FT%, effFG%, effPTS and scale the data to [0,1] for easier convergence of back-propagation algorithm [4].
2. I use a 75-25% split for the training-testing samples and a 10-fold cross-validation with 10 repetitions (leave one out).
3. I train a regression neural network with 3 layers using the neuralnet [12] R package under default parameters defined by the caret [11] package. See Appendix A.1 for details.
   * 1. **Results: Offensive power**

The best 3 models for points scored (offense) based on R-squared. Selected model (5-3-2) on top:



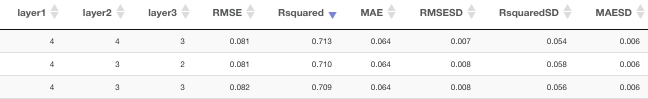
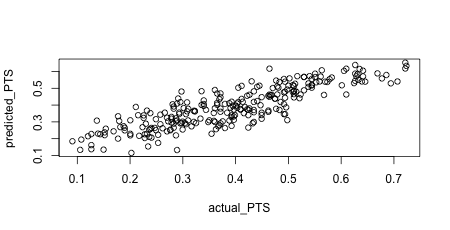
Actual vs. predicted points per game on out of sample (testing) data (25% = 266 observations). Scaled to [0,1]



The Network: Visit the Model tab on the R Dashboard for details on the weights connecting each of the nodes.

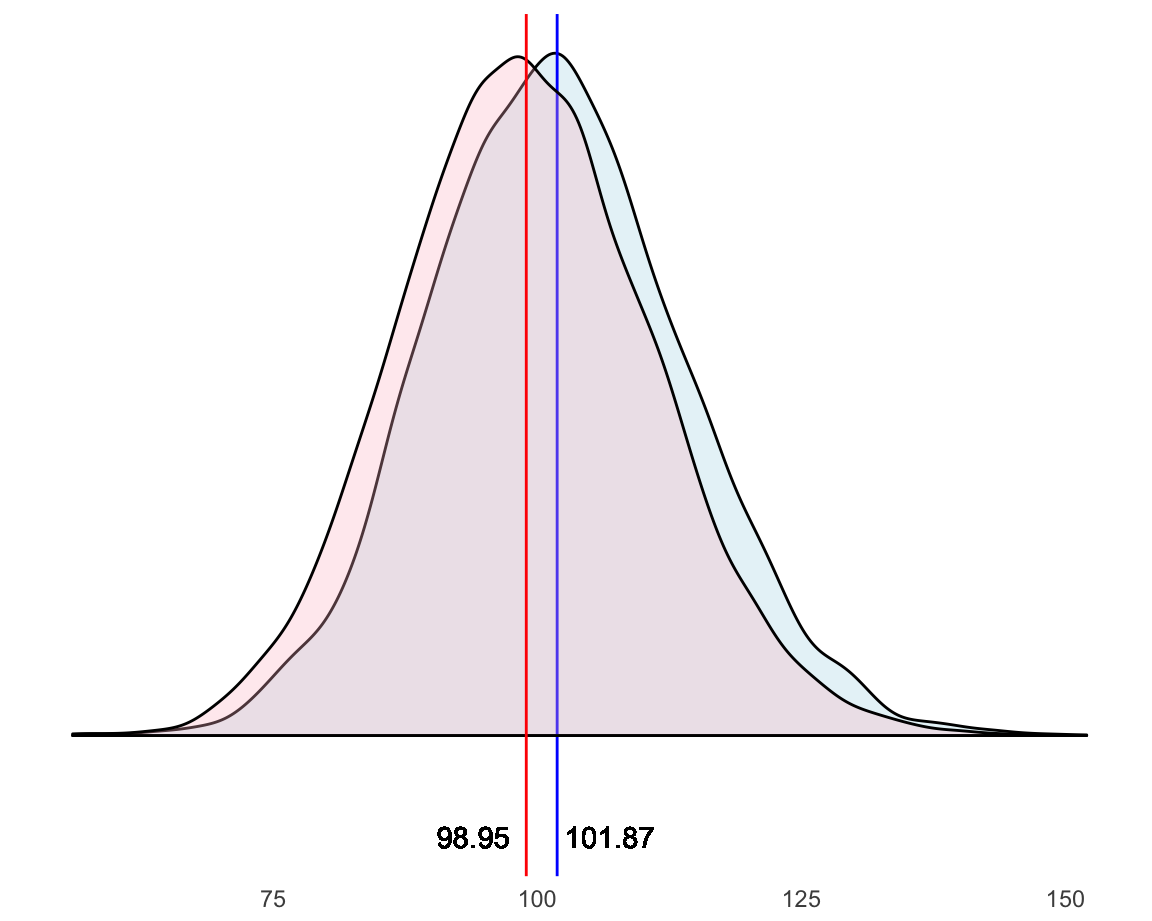
* + 1. **Results: Defensive power**

The best 3 models for points scored (offense) based on R-squared. Selected (4-4-3) on top:



Actual vs. predicted points allowed per game on out of sample (testing) data (25% = 266 observations). Scaled to [0,1]. Clearly not as good a fit as the offensive model but still picking up a good enough signal.

* 1. **Probability model**

Now that I have a way to predict a team’s offensive and defensive powers, I can use these as the estimated mean parameters of a Normal probability model. The choice for the Normal distribution is obvious if we explore the density of points scored by teams in the last 8 seasons: (since 2009-2010)

Where the blue density corresponds to points scored by home teams and the red density to away teams. The sample average is 100.41 with a sample standard deviation of 12.16. Now, the variance is pretty consistent across the sample (12.13 for home teams and 12.0 for away teams) and for simplicity I take the overall sample standard deviation as the estimation for the standard deviation of both Offense and Defense Normal distributions:

|  |  |  |
| --- | --- | --- |
|  |  | (5) |

* 1. **Predicting Wins**

So far I can predict how many points a team will score or allow on average, and I know the probability family that is behind the model. I have all the pieces I need to compute how many wins those offensive and defensive powers award: Suppose team A plays against team B and suppose team A is the home team. I know how many points team A scores on average, call it: *ptsA*. Empirically I know from 3.2 that home teams score 1.46 more points on average. Call it: *home\_court\_coeff*. Next, I need to plug in team B’s defensive power, call it: *pts\_agB*. The higher the defensive power is, the worst the defense is (as opposed to offense). So, I will add this number to team A’s offensive power. This will give me an astronomical number of points scored for a basketball game, which will be cancelled out when I calculate team B’s offense. However, in order to be able to simulate realistic game scores (and eventually bet in Vegas, but that’s a different story) I will subtract the overall average points (100.41) from that number. So, here is how to calculate a team A’s points adjusted by home court and opponent:

|  |  |  |
| --- | --- | --- |
|  |  | (6) |

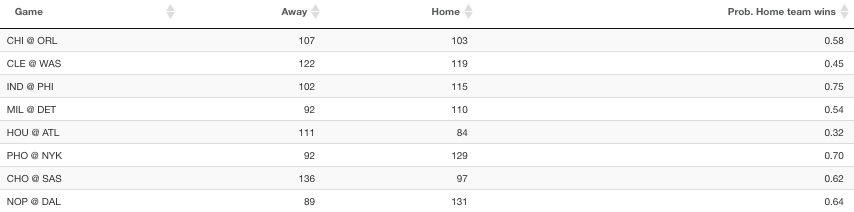
And for team B as the away team:

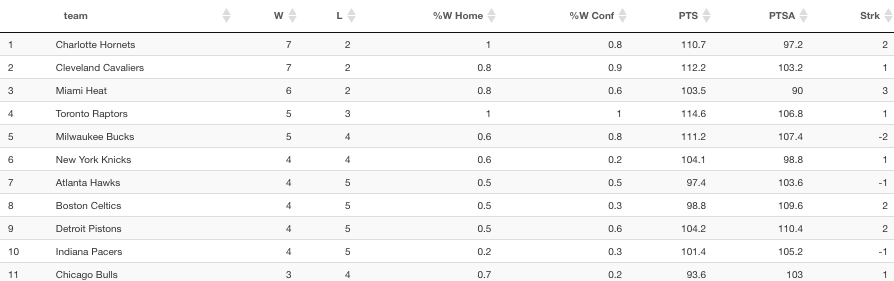
|  |  |  |
| --- | --- | --- |
|  |  | (7) |

Now I can plug these 2 measures as the estimated means of Normal distributions with standard deviation 12.16 and thus calculate the probability of team A beating team B:

Let and be random variables following a Normal distribution and P a probability measure defined on the space of all possible matchups:

|  |  |  |
| --- | --- | --- |
|  | P(team A beats team B) | (8) |

If I calculate these probabilities for all games in a regular season, I will be able to obtain expected wins for each team analytically. I will show the details in the results section but, as an example, take one random day in the regular season say, November 3rd. Here’s an extract from the R dashboard showing probabilities and simulated outcomes (Home and Away columns) for some of the games:

I can easily then simulate an entire season and compute standings at any given point. Below the Eastern Conference standings as of November 3rd according to my simulation:

* 1. **Player similarity: t-SNE**

A quick aside before I continue describing the methodology. One of the key aspects of this model consists on being able to compare and classify players, and eventually teams. I use the t-SNE algorithm (Van der Maaten, L.)[5] to reduce the multidimensional space defined by players’ measurable stats (see 2. The Data) into 2 dimensions. In a 2-D space I can easily calculate Euclidean distances between players and at the same time visualize them.

The inputs to this algorithm are players skills and the output are the (x, y) coordinates which we can map using a scatter plot. I use t-SNE in a variety of situations which will be exemplified further in the paper but, in brief:

* Similar players by age: To predict how players’ skills evolve with age
* Similar players historically: To see players’ skills evolution over time
* Similar players for the current or any particular season: For clustering purposes

Wattenberg, et al. (2016)[7] provides a good description of how the algorithm performs and what the parameters mean. I’ll leave the details to the reader to not interrupt the flow of the narrative.

* 1. **Building the roster**

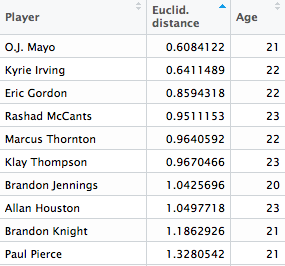
So far, I used historical data to build a model that calculates win probabilities based on a team’s offensive and defensive predicted powers. However, for this to be useful, I need to be able to predict these offensive and defensive powers based on the players that make each team at a given point. I need to do 2 things: Predict players’ skills, how good they will be, and their usage or how much time they will spend on the court. At the start of a new season we find 4 types of players:

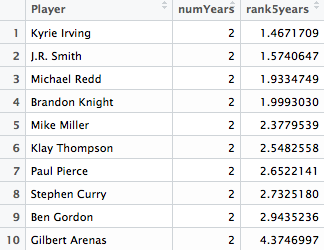
1. Returning NBA players who played in the previous season
2. Returning NBA players who didn’t play in the previous season
3. Rookies drafted from American schools
4. Rookies from international teams

For each of those I will use a different approach mostly determined by the disparity of data sources. What follows is a brief account of the methodology, for a more details and the R code please see Appendix A.2.

* + 1. **Returning players (cases a and b)**

The algorithm runs through all players who played in the last NBA season, obtains the closest 10 players for the past 5 years of his career and calculates how their skills changed when they turned one year older. This is inspired by FiveThirtyEight’s CARMELO (Silver, McCann, 2017)[3] and it’s better explained through an example:

* 1. Take Bradley Beal from the Washington Wizards and consider his last 5 years of NBA stats (from 19 to 23 years old).
  2. Calculate the closest players to Bradley Beal for each age. That is, calculate five t-SNE 2D maps: for all the 19, 20, etc year old players. If I keep the top 20 for each age, the output is a 100 record file like this[[7]](#footnote-7):
  3. For each of the players calculate the average Euclidean distance and the number of appearances in the list. The more appearances and the lowest distance the higher in the rank. Take the top 10:



* 1. Bradley Beal will turn 24 this new regular season so for each player i in this list and for each of their effStats[[8]](#footnote-8) calculate the variation when they went from 23 to 24 years old:

|  |  |  |
| --- | --- | --- |
|  |  | (9) |

And I obtain the median of all the 10 players variations for each stat:

|  |  |  |
| --- | --- | --- |
|  |  | (10) |

Finally, update Bradley Beal’s stats according to this variation:

|  |  |  |
| --- | --- | --- |
|  |  | (11) |

* 1. The only constraint to the above approach is that players younger than 20 or older than 39 are so scarce that their t-SNE maps don’t really make sense so I assign the latter to the class of 20 year olds and the former to the 39 year olds.

Sometimes a player didn’t record stats for the previous season, he played in another league or was injured. In this case I use his stats in his last recorded season in the NBA as the base for their predicted stats.

For players with no previous experience in the NBA, I use the average player[[9]](#footnote-9) stats.

* + 1. **Rookie players (cases c and d)**

I retrieve rookie players from current rosters[[10]](#footnote-10). They are labeled as “R” in the Experience column. This file does not contain rookie stats. I will merge this file with College Players’ stats[[11]](#footnote-11) and European stats[[12]](#footnote-12) for international players. Here are the steps I follow:

1. Query college players who played at least 15 games and 7 min/game last season. Totally arbitrary numbers under the assumption that if a player is fit for the NBA he would have played enough. For college players who played in multiple seasons, I take the average of their college stats.
2. These 2 files are retrieved from different tables in basketball-reference and sometimes their names are spelled differently. I have to manually change a few. See Appendix A.2 for details.
3. There are always a few players for whom I couldn’t find readily available stats, if they are international players (playing in Europe or not) they are assigned the average of all the stats of players coming from Europe. For the rest (college players who for some reason were not matched or found) I assign the means of the stats of rookie college players.
4. Finally, because college and European players play fewer minutes per game than NBA players, I assume their effective per minute stats are based on 48 minute games, that is: effMin = MP/3936. Still, this is not realistic in most cases, as college players tend to play way fewer minutes on their rookie year than they would in college. To account for this discrepancies we can take a look at the percentage of minute reduction experienced by rookie NBA college players by draft pick.



Data spans the last 20 years and it’s limited to players who played at least 30 games in their rookie year. Clearly, a lower draft pick guarantees a higher usage on his rookie year. I subsequently add the pick round to my dataset and adjust effective minutes based on it.

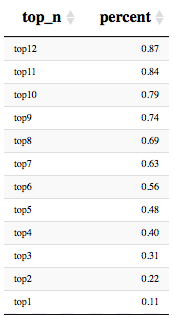
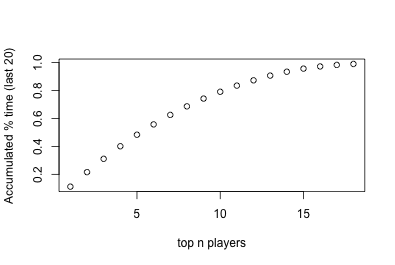
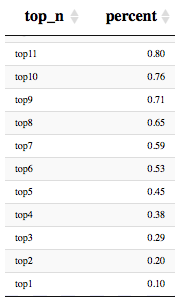
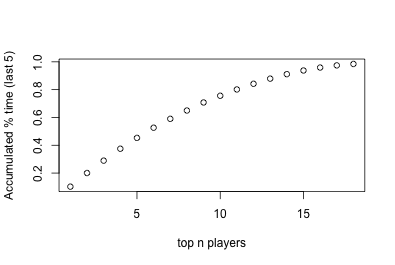
* + 1. **Putting it all together**

At the start of a new season[[13]](#footnote-13), rosters are mostly final so the starting point is the most current NBA rosters. Merging this file with the previously calculated stats for returning NBA players and rookie players described in 3.5.1 and 3.5.2 will create the full list of predicted stats by player. Again, for details see Appendix A.2.

* + 1. **Adjusting player usage at team level**

So far the computed statistics and effective minutes of playing time were computed on an individual basis for each player regardless of the team they play for. Obviously, teams formed of players who play heavy minutes will create an unbalance. To correct for this unbalance I transform the effective minutes of playing time into percentage with respect with total effective team minutes.

This method will balance out the minutes but may not yet reflect a realistic distribution of minutes. I wanted to make sure so I looked at the distribution of playing time on average. What I found is that, in recent years, the share of playing time becomes more spread throughout a roster. For instance, the top 7 players accounted for 59% of total minutes on average in last 5 years while for the last 20 years, it accounts for 63%.



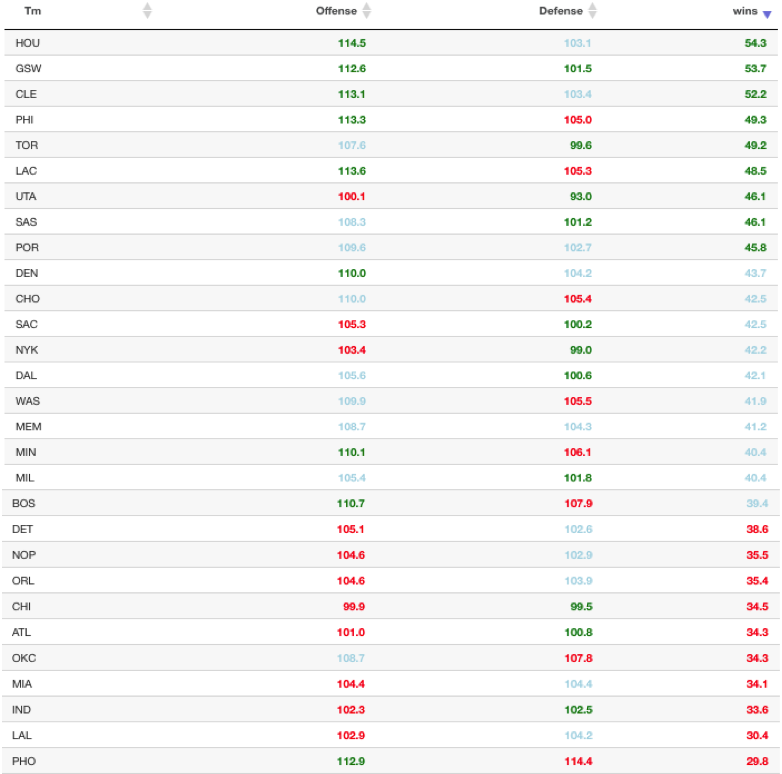
Considering this, I adjust playing time with the following parameters: top 7 players in each team sum up to 60% (round number in between 59% and 63% giving more weight to recent trends) of total playing time with no player accumulating more than 10.5 % of it (averaging out top 1 percentage from both examples above). For instance, in a team with 3 star players, their total share of minutes may be as high as 31.5% leaving 28.5% for the remaining 4 players.

1. **The Results**

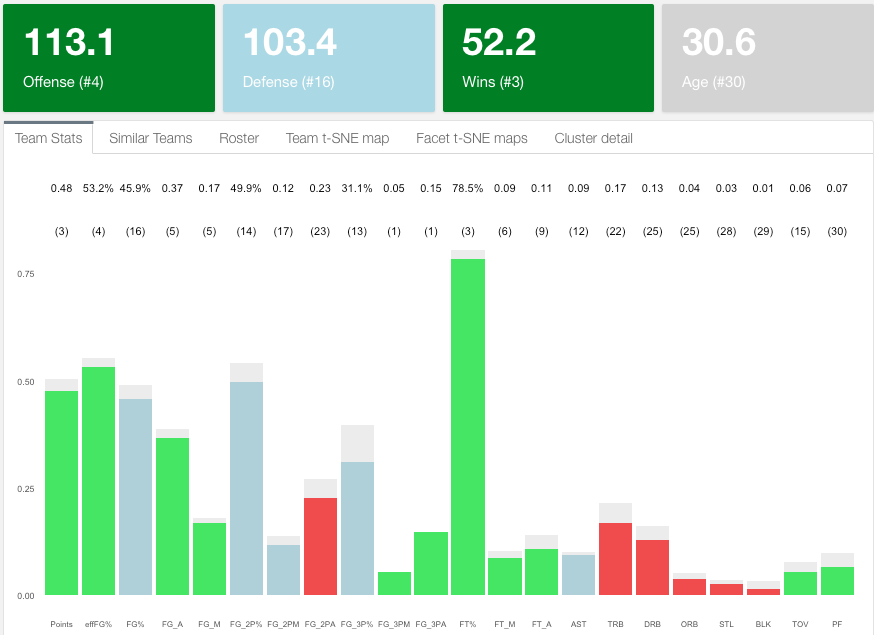
I have defined the probability model, predicted how good players will be, how much time they will spend on the court and I know how to calculate expected wins running a simulation of the regular season. So it’s time to show the actual results and have some fun. One final thought: what if I use the described methods above to evaluate the offensive and defensive strength of players under the assumption that they are “teams” composed of clones of the same player? How good offensively and defensively would a team be with 5 Kevin Durants on the court? Keep reading to know more. I recommend readers follow along with the R Shiny dashboard[[14]](#footnote-14).

* 1. **The Teams**

Here are the predicted results for all 30 NBA teams. In what follows I will explain the details.



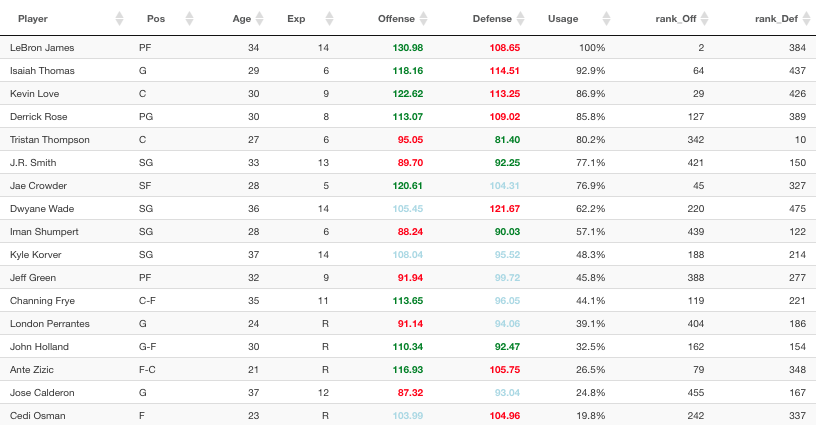
* + 1. **Team stats**



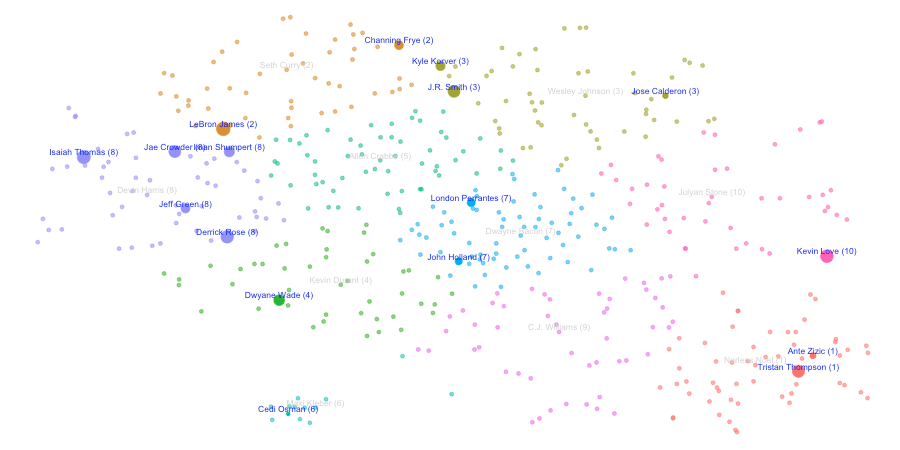
The dashboard above corresponds to the Cleveland Cavaliers. I rank teams in every category and split them in 3: top third, middle third and bottom third assigning, respectively, colors green, light blue and red. I can quickly spot strengths and weaknesses. My model gives them 52.2 wins in the regular season, they rank #4 in offense and #16 in defense and they are expected to be #1 in 3 points made and attempted although only #13 in 3-point percentage. Clearly their weaknesses are rebounding, blocks and steals. Unsurprisingly, 2-point attempts ranks low as they are expected to be a 3-point shooting team. Finally, they are the oldest team in the league averaging 30.6 years old.

* + 1. **Roster stats**

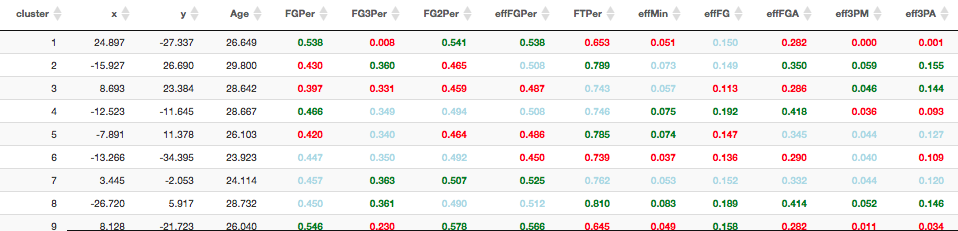
Below is Cleveland’s roster. I can show offensive and defensive individual contributions and usage (playing time). Again, green means top third and red bottom third. LeBron, Isaiah and Love are predicted to be good in offense in the NBA and they all have a high usage rate as expected. Defensively, Tristan Thompson is their best player ranking #10 in the league. I provide further analysis on players on **4.2. The Players**.

****

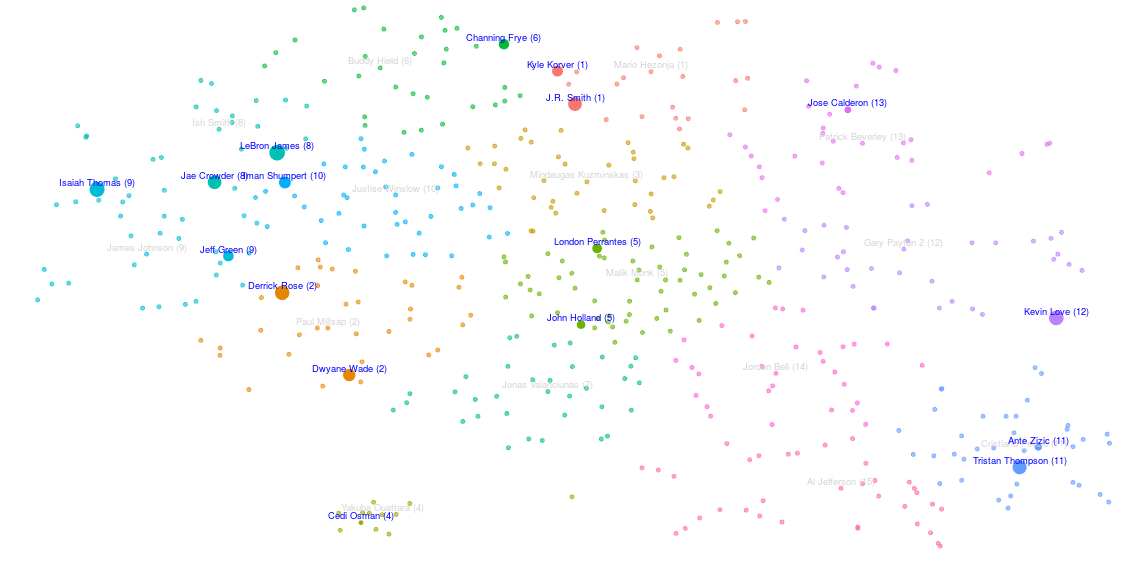
* + 1. **Team t-SNE map**

Using t-SNE I can plot each player on a 2-D map and perform different exploratory and analytical analysis. This is how Cleveland players are distributed based on t-SNE. Colors correspond to 10 K-means [6] clusters and the size of the dots to player usage.

I could add more clusters or reduce the number of them. Clusters represent different types of players[[15]](#footnote-15) and in order to make a more informed decision on how many clusters (or player types) exist in the league, I can look at the stats that define each of the clusters. If we look at 10 clusters:

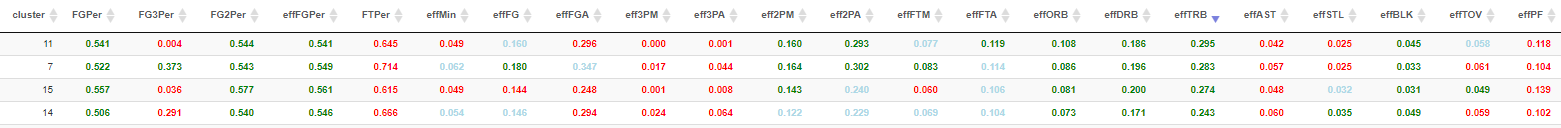
This classification puts LeBron in the same cluster (2) as Channing Frye. A closer inspection shows that this is the group of veteran players that play low minutes and shoot from long range. They have high assists but don’t have great shooting percentages and don’t rebound, steal or block very often:



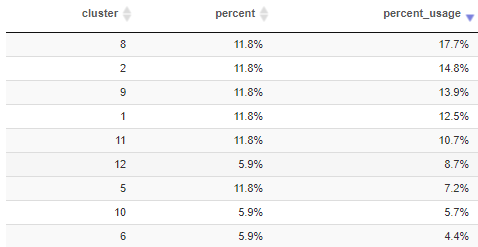
Looks like a higher number of clusters will possibly represent the player typology better. See now for 15 clusters how LeBron is no longer associated with Fry but now with Jae Crowder.

Still not what one would expect. An interesting analysis would be to identify weak areas in a team and the type of player in the market that would fill that gap. I already know Cleveland’s weakness is in the rebounding, blocking and stealing areas on the defensive end but also needs to improve field goal percentages on the offensive end. If we look at the 15-cluster k-means detail, these are the clusters with better averages of those:

Clusters 11, 7, 15 and 14 fit the bill perfectly as almost all of those categories are green:



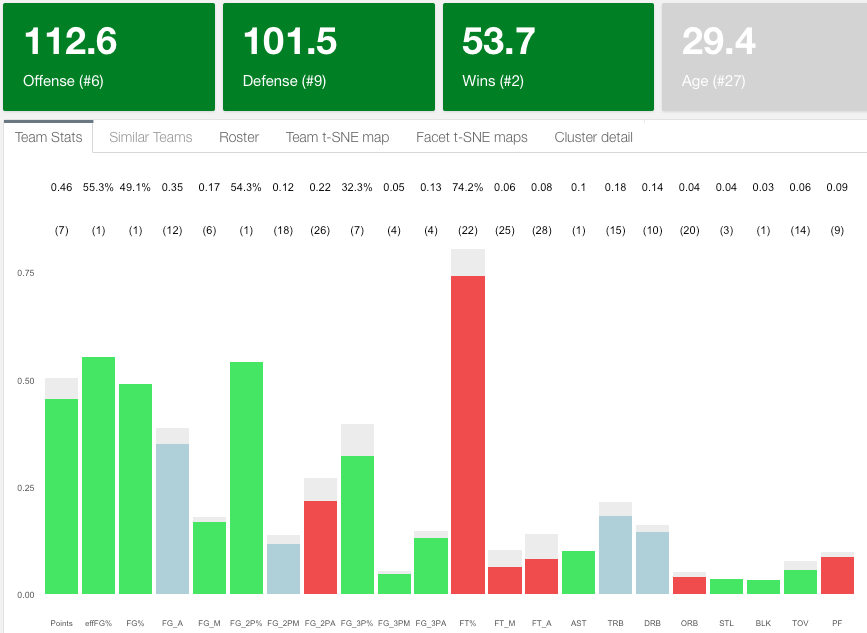
Now I look at how Cleveland players are distributed across clusters by headcount and by usage, (which matters more to predict wins)[[16]](#footnote-16):



Except for cluster 11, filled by Tristan Thompson and minimally by Ante Zizic, the rest of the clusters are not represented. I will go into more detail on how to look for players that fit a team’s needs. For now, I will move on to team similarities as an alternative way of assessing the winning potential of a team.

* + 1. **Team similarity**

When looking at the roster and structure of a team there is always a reference, the gold standard if you like. Arguably, in 2017, this is the Golden State Warriors and my model seems to back this up:

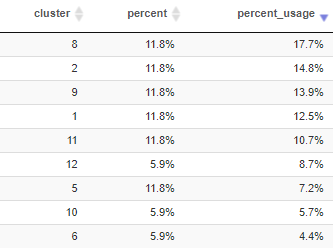
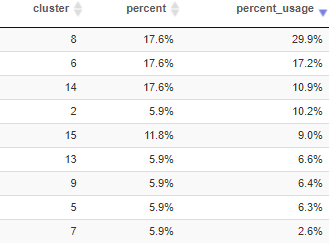
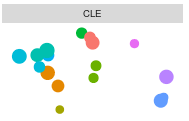


As expected it’s green almost across the board (except free throws and offensive rebounds) and they are number two in wins, closely behind the Houston Rockets. I would argue whatever shape they show on the t-SNE map should be close to an optimal distribution of talent.

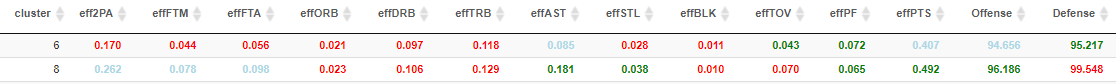
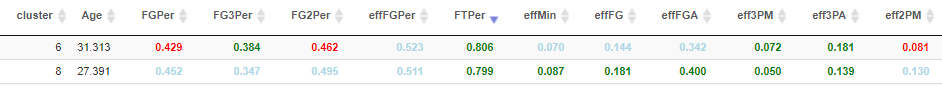


The above image shows Golden State’s distribution vs. the rest of the teams colored by 15 K-means clusters. Alternatively, I can use a gradient of greens and reds to spot offensive and defensive strengths or weaknesses. The image below corresponds to Offense (green is best):

Let’s take a closer look at GSW vs. Cleveland:



Totally different shapes, both winning teams but clearly GSW has been more successful lately. They seem to concentrate almost 50% of their usage on clusters 8 and 6 as opposed to barely 20% for Cleveland. A close inspection of the characteristics that define these two clusters:



Cluster 8 corresponds to heavily offensive players, high in scoring and field goal percentage who play top minutes. Unsurprisingly, Steph Curry, Klay Thompson and Draymond Green belong to it, as well as LeBron and Jae Crowder (this one maybe surprisingly but that’s what the model says). Cluster 6, on the other hand, is also characterized by 3-point shooters who play secondary minutes, they are more experienced players (Age over 31 on average) with good defensive skills. This is the territory of Shaun Livingston, David West or Nick Young versus Channing Fry in Cleveland. This analysis could continue in many different ways and the reader can definitely get the full experience using the dashboard.

But it’s really hard to measure the dissimilarity of shapes by just looking at the 2-D maps so I wanted to provide an analytical measure. I use a modified version of the Bhattacharyya distance [8] to compare probability distributions of 2 given teams A and B:

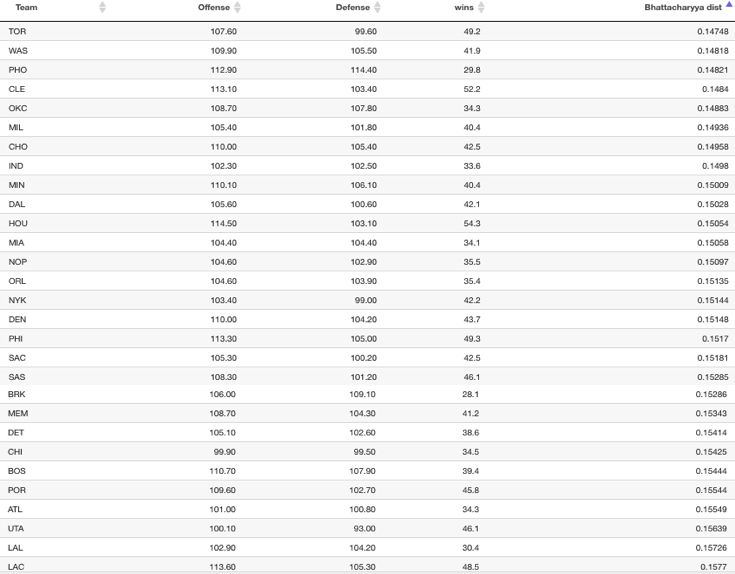
|  |  |  |
| --- | --- | --- |
|  |  | (12) |

Instead of counting the number of players that fall into each cluster, I use the accumulated usage percentage to better account for minutes played per position. The second modification is to use:

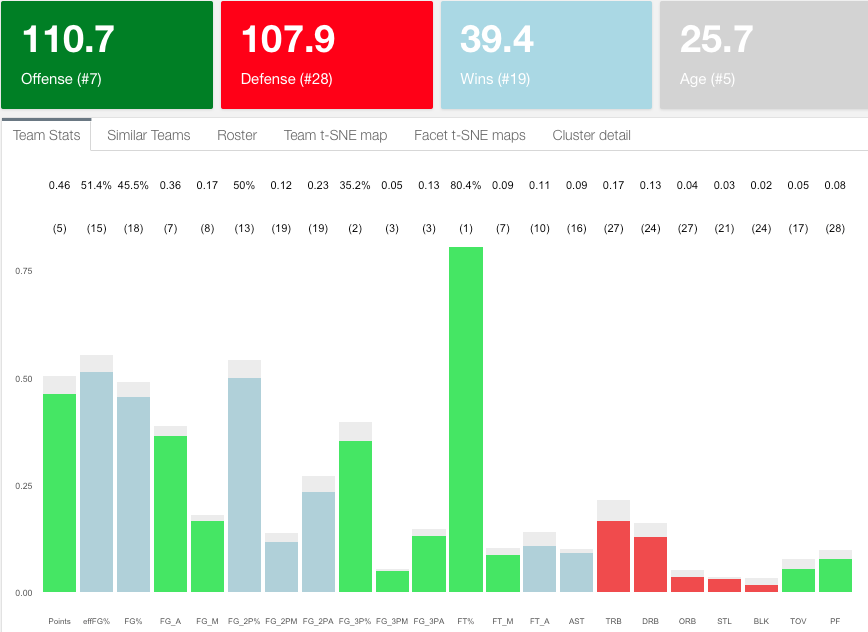
|  |  |  |
| --- | --- | --- |
|  |  | (13) |

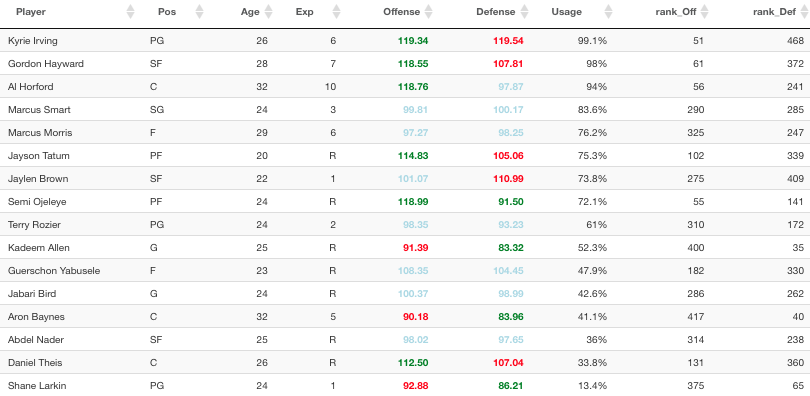
Rather than the original negative logarithm as this will always return a positive number.

A visual exploration tells me that, for instance, San Antonio or the Clippers have a very different shape from Golden State. The analytical results confirm this, Clippers are the furthest apart team from Golden State although that doesn’t mean much in terms of wins, only that they have 2 different winning styles. Actually, one of the closest teams to Golden State is Phoenix which my model predicts a low number of wins.



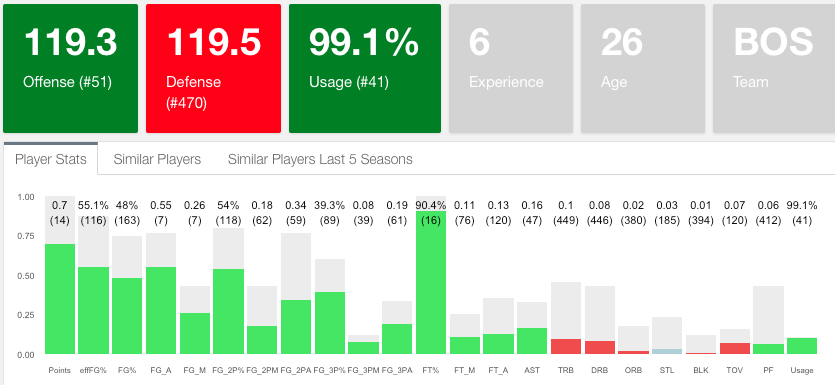
The final predicted number of wins for a team will depend on many different factors, one being the accuracy of player predictions, which is particularly hard the younger the roster is (like Boston). Sometimes it’s also a question of how to optimally distribute minutes of play. Let’s take a look at Boston:



Overall Boston looks pretty good. Not many reds, very consistent overall. Looking at just the individual team stats one would think they should be expected to win more games than just 39.4. Before jumping to conclusions let’s look at the minute share in Boston’s roster:

Boston’s offense ranks at #7 but their defense is the problem. According to the model, they are the third worst defense in the league. Looking at the roster, their best 3 defensive players: Allen, Baynes and Larkin are at or near the bottom in usage. This is a clear case in which a redistribution of minutes would benefit a team’s expected wins. I will actually test this hypothesis in the Trade section of the paper. Keep reading.

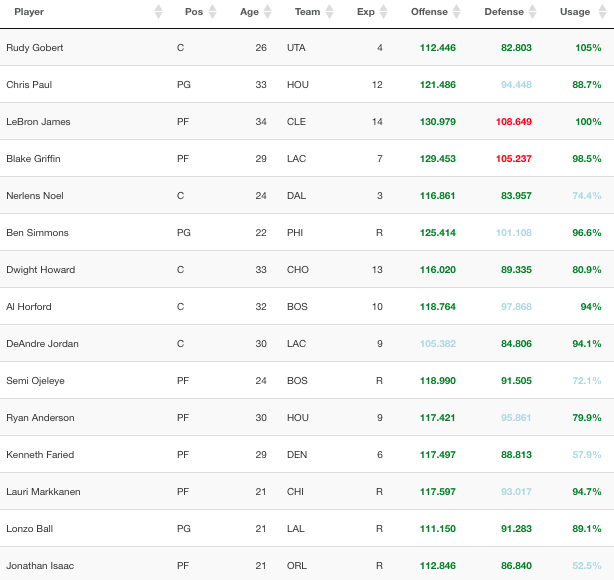
* 1. **The Players**

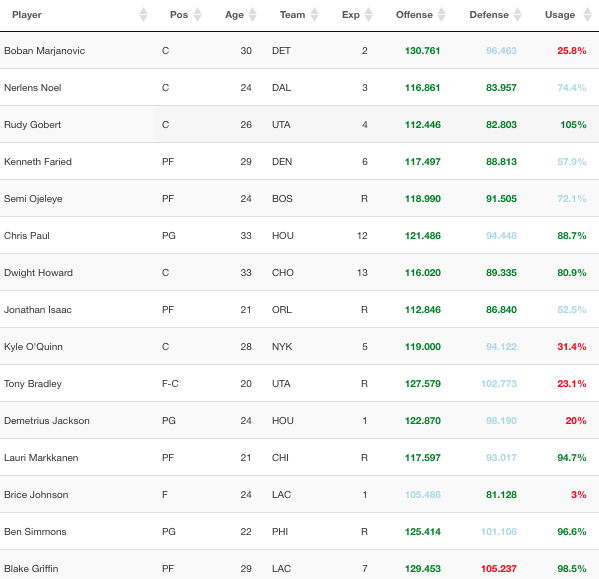
Similar to the analysis I did for teams, I can do for players. As discussed above, I can apply the Offense and Defense neural network models to players under the assumption that they are “teams” formed exclusively of his clones. Imagine a team with 5 Kyrie Irvings on the court and a bench full of Kyrie Irvings. Well, let’s look at what the model would say about it:

A team full of Kyrie Irvings competing against regular NBA teams would be pretty good offensively but will struggle defensively as nobody would be able to protect the rim, grab enough rebounds, etc. That’s my interpretation of what the model shows above.

* + 1. **Player plus minus**

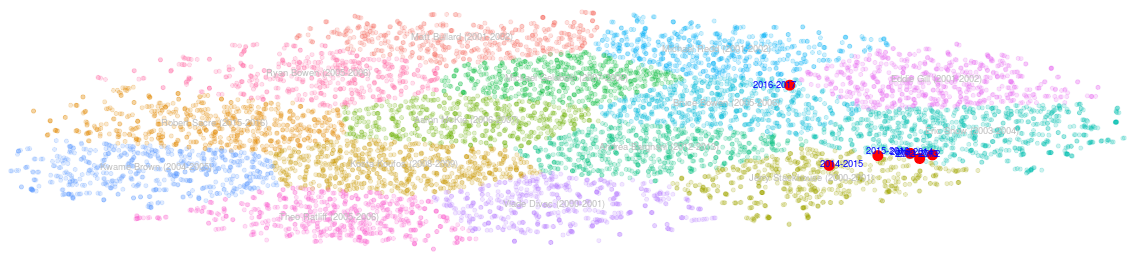
This leads me to define what I call a player’s plus-minus, nothing to do with the Box plus minus used on regular stat sheets. I define plus-minus as the predicted Offense minus the Defense, which allows me to rank the best 2-way players and identify potential improvements in the number of wins for a team. In addition, I also compute the adjusted plus-minus which accounts for the usage or minute share of each player. Here I rank the top 15 players using both plus-minus (table on the left) and adjusted plus-minus (right):

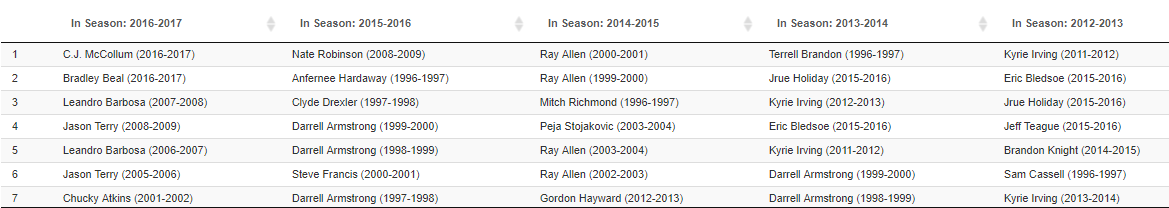


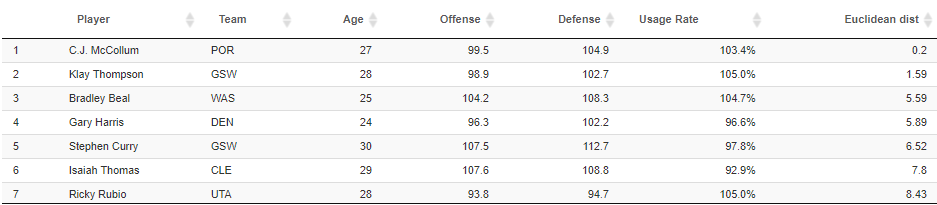


Boban Marjanovic is the most valuable 2-way player according to the model, followed by Nerlens Noel and Rudy Gobert. The team model applied to players returns some surprising results because big players will normally have high field goal percentages and also high rebounding and blocks which will boost their offense and defense. It’s hard for me to image 5 Boban Marjanovics on the same team. If we adjust by usage, the list seems a lot more reasonable with LeBron James, Chris Paul and Blake Griffin on the top of the list.

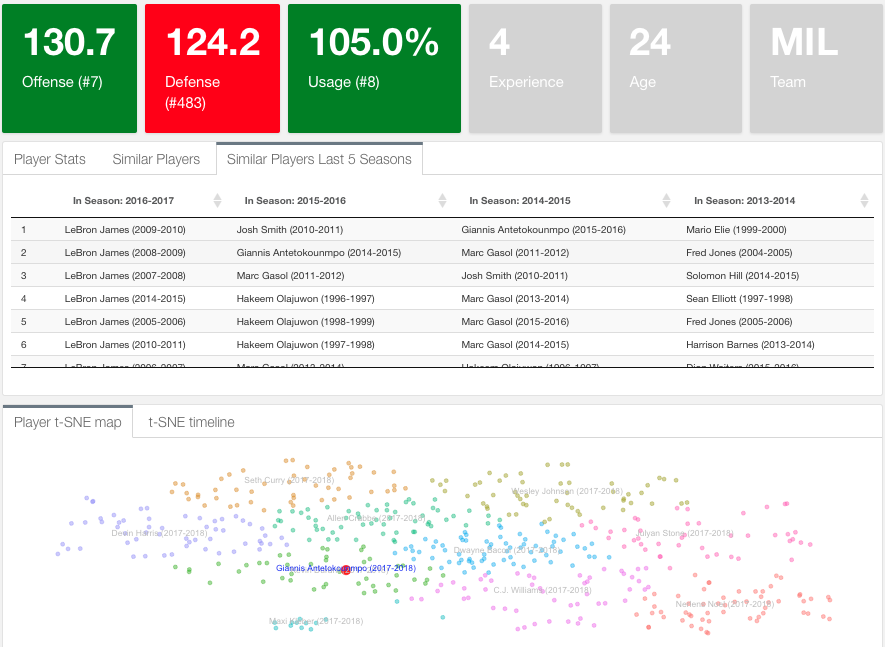
* + 1. **Player evolution**

But it would be premature to evaluate players by their plus-minus exclusively, especially players with little experience in the NBA as their per minute stats respond to a smaller sample size. However, for those with a little more experience, we can look at the consistency of their game over time. I do that by computing a t-SNE model with all players for the last 20 seasons. The result is a 2-D scatter plot similar to the previous one with the advantage of seeing how players evolve over time. Let’s look at Kyrie Irving once again:

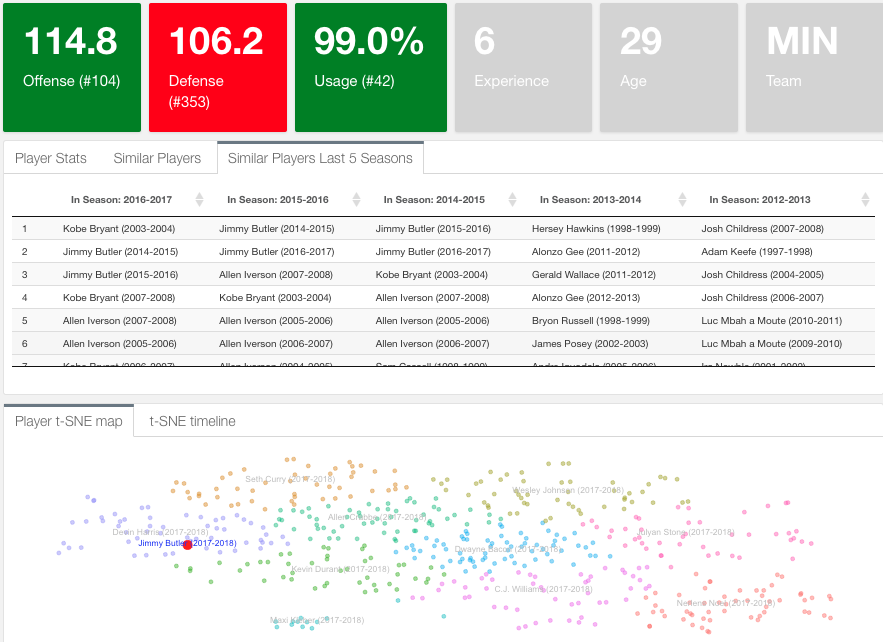
His stats look pretty consistent throughout his career except for last season in which he clearly departs from that territory. We can think of different reasons why this sudden change which definitely would have been a lot harder to spot by simply looking at his stat sheet, but let’s try to see this from the perspective of who were Kyrie’s neighbors (closest players by Euclidean distance) for the past 5 years:

From Clyde Drexler, Darrel Armstrong or Hardaway to C.J. McCollum, Bradley Beal or Leandro Barbosa. This reinforces the opinion of those who saw Kyrie becoming more like a sidekick type of player next to LeBron and wanting out to be back in the vicinity of the greats. Unsurprisingly, if we use the t-SNE map of current predicted season stats, these are the closest players to Kyrie:

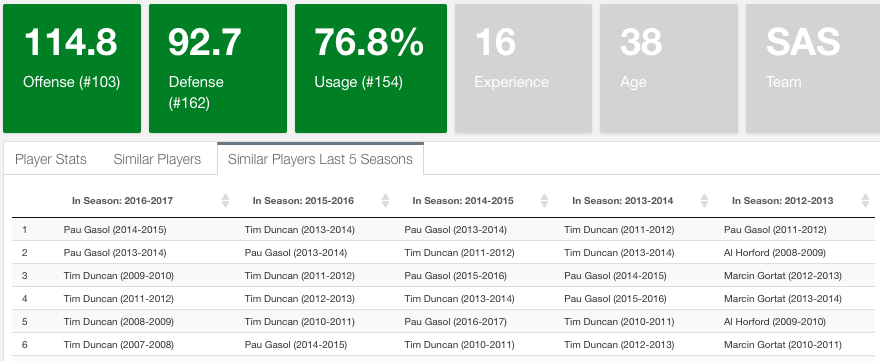
Let’s now look at a couple other examples of players who underwent big transformations. Here is Giannis’ Dashboard:



Giannis path has changed significantly: From Marc Gasol to Olajuwon to being the closest player to LeBron James. Another interesting evolution happened to Jimmy Butler who basically moved from mediocre stats to the closest mix between Allen Iverson and Kobe we see in today’s game. Below is Jimmy Butler’s dashboard:



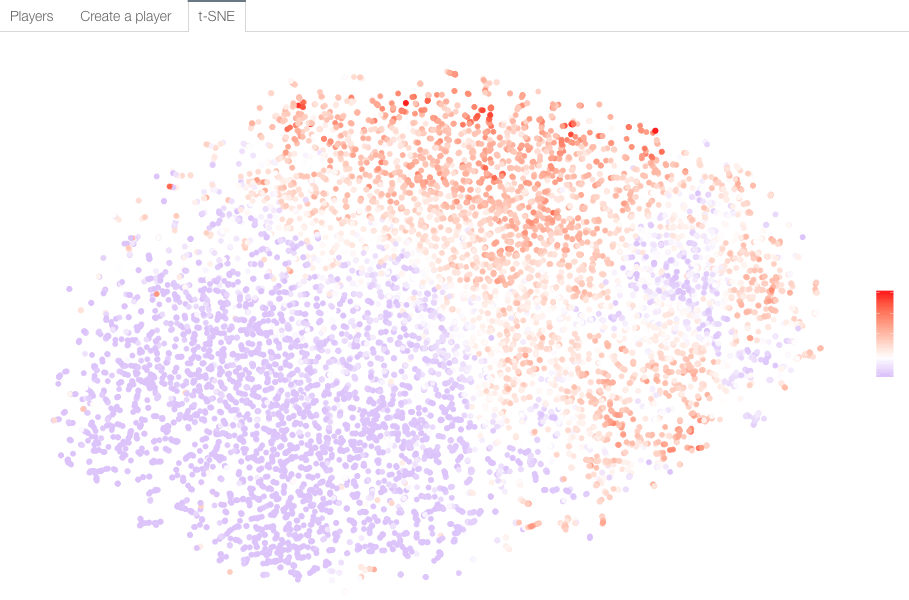
I will finalize this section with a real case scenario. When the San Antonio Spurs had to explore the market to replace Tim Duncan I’m sure they didn’t use my model but probably used a similar one. Just check out the closest players to Pau Gasol in his dashboard. No comments:

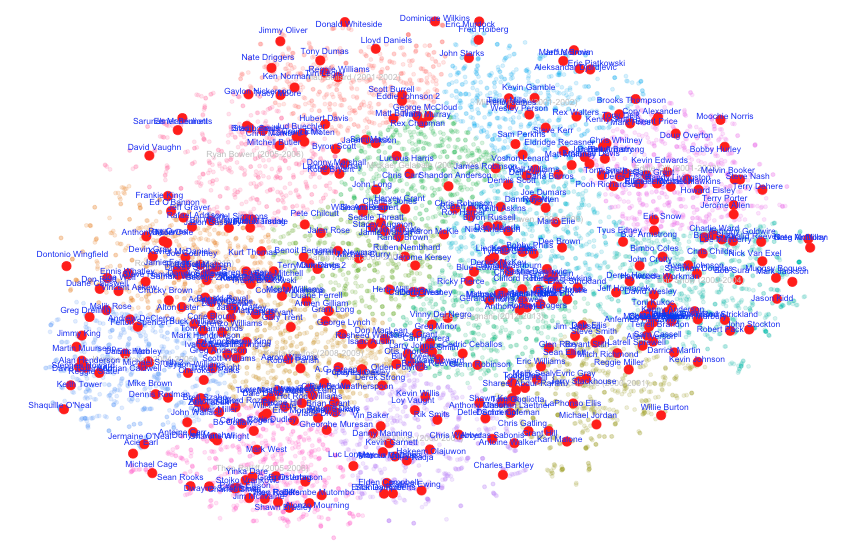
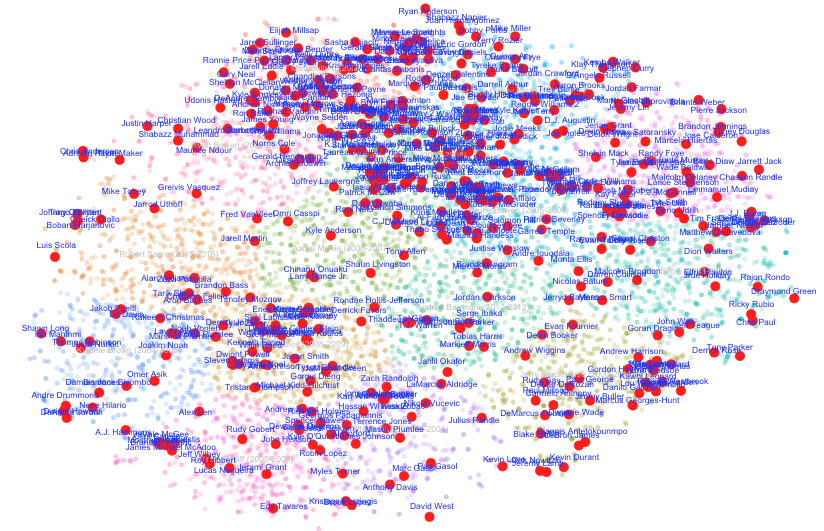


* 1. **The complete NBA t-SNE map**

Plotting all NBA players for the past 20 seasons on a 2-D t-SNE map provides a wide variety of possibilities. I have shown how we can trace a player over his career using this method. What we can also do is trace a team or a certain style of play over time. I will show how the 3-point shooting has shaped the game in recent years:

First of all, identify where does the 3-point shooting territory falls in the map: color by 3-point attempts where red is most:



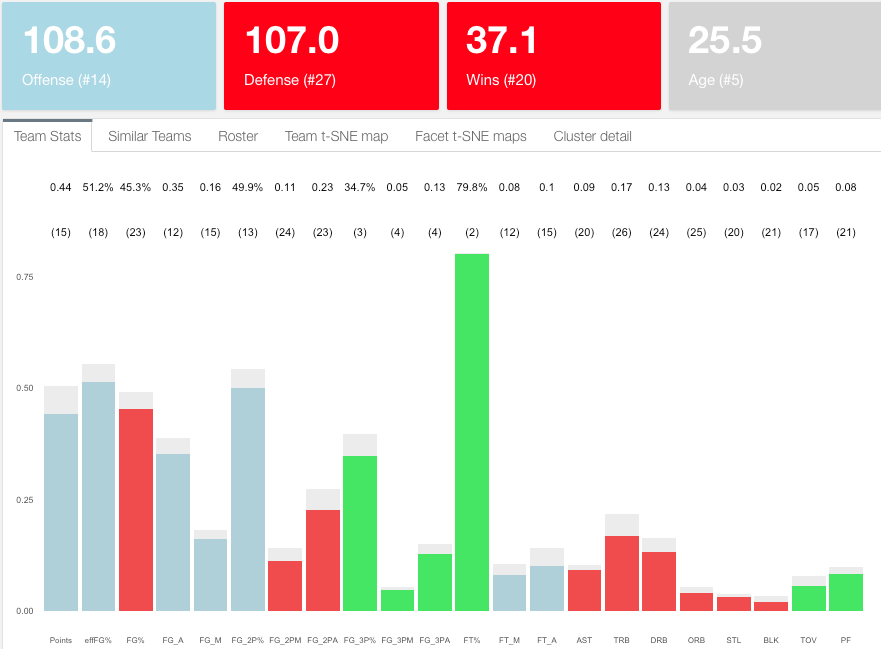
Now see how players lean towards the red region over the years. The left hand side picture corresponds to the 1996-1997 season. The image to the right to last season (2016-2017):

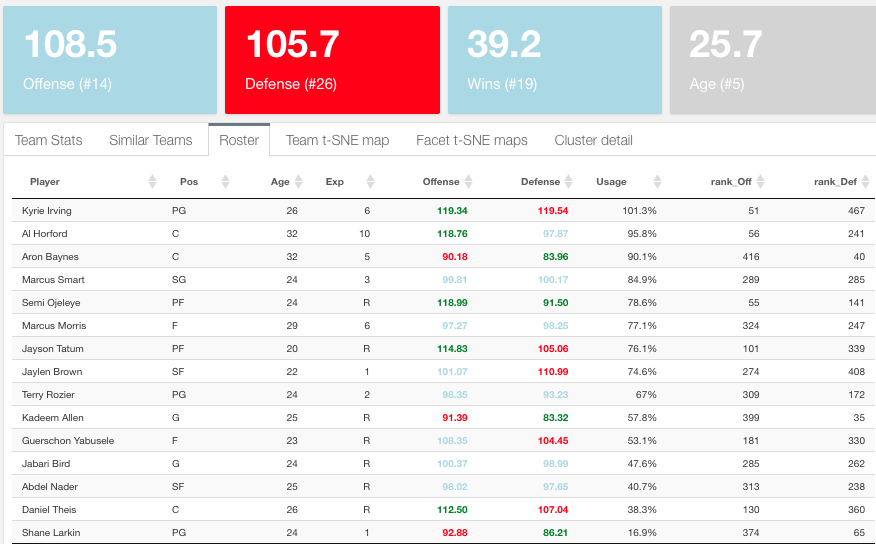
1. **The Trades**

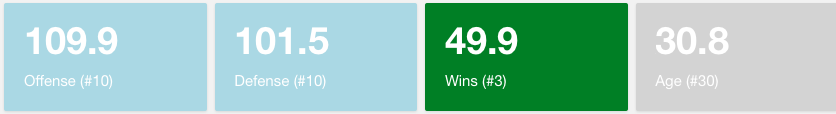
The results section set the tone for the type of analysis that can be performed with the described model. The next step if to test some of the hypothesis presented and the robustness of the model. I will start by defining what is a trade: I consider a trade any change effected in the roster of a team that modifies their predicted Offense and Defense powers. In particular, a trade may be the traditional trade of 2 or more players between 2 or more teams. Can also be a player leaving a team and the league (due to injury, retirement or traded to another league). Finally, it can also be a player who increases his usage within his team.

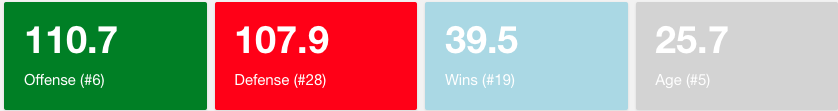
The key aspect of a trade is that when it does happen, the whole league stats and team rosters are re-calculated as a change in a team’s Offensive and Defensive power will affect not only their number of wins but also every other team’s number of wins. Now let’s get to the examples:

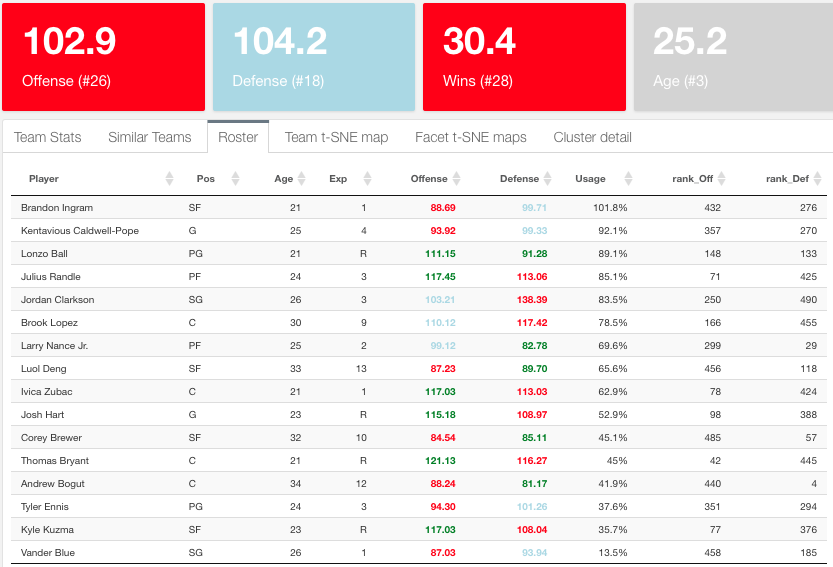
* 1. **Game of Trades**

Let’s focus on Boston again and let’s look at their stats dashboard without Gordon Hayward:

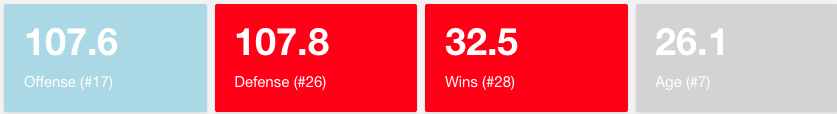
Boston gets worse as expected: Gordon Hayward only represents a couple of wins for Boston. The defense seems to improve though. Let’s now increase the usage of Aaron Baynes from his current 13.1 to 30 minutes per game[[17]](#footnote-17) and see how this further impacts the team:

Using more minutes of Baynes on the court improves Boston wins by 2, recuperating what they lost with Hayward, only this time through defense. I could continue this pattern to optimally tweak Boston to be a playoff team. Similarly, here is the impact of Isaiah Thomas’ injury on the Cavs:

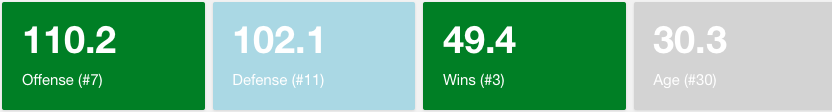
The biggest drop is in offense (-3.2 points), the defense gets better (-1.9 points) as players like Tristan Thompson will play more minutes but the balance is negative for Cleveland as they lose 2.3 wins. We can see the ripple effects of Isaiah’s injury for Boston. They improve by 0.1 wins:



What would happen if I trade LeBron to the Lakers? This is the Lakers before LeBron:

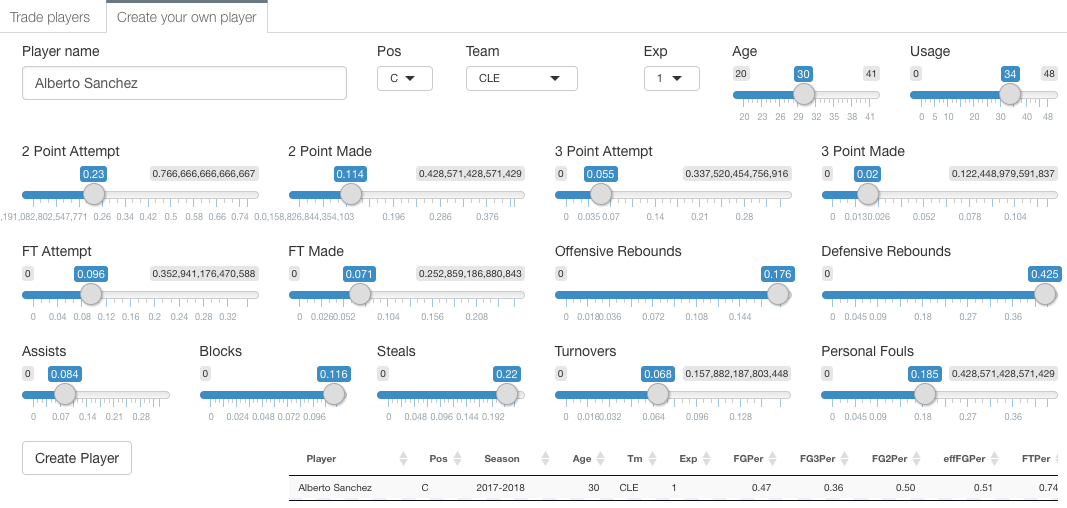
After acquiring LeBron James the offense improves greatly but the defense needs tweaking just as with Boston. Overall, the Lakers improve by 2.1 wins.

A quick look at the Lakers’ roster tells me Kyle Kuzma, Corey Brewer or Luol Deng should play more minutes. The reason Lonzo Ball has a much higher usage than Kuzma when both are rookies has to do with how the model uses the draft pick to predict playing time[[18]](#footnote-18).

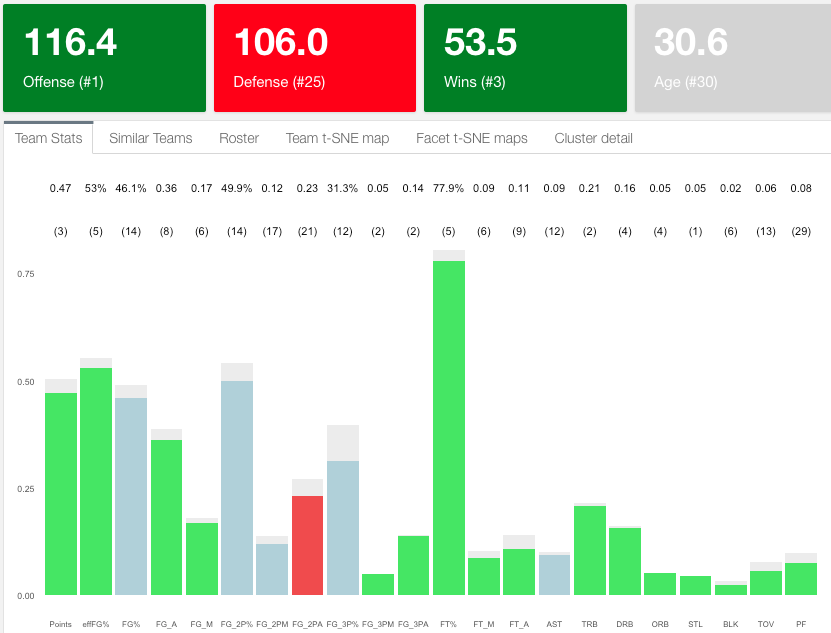
Finally, what happens to a LeBron-less Cleveland? -2.8 wins (vs. -2.3 wins without Isaiah Thomas)

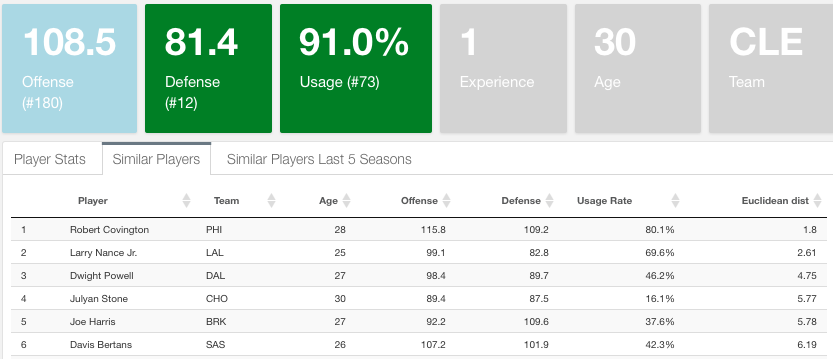
* 1. **Create your own player**

Why stop there? I can create my own player and plug it into any team. This will also allow me to improve my understanding in how the neural network works generates the offense and defense outputs. I am going to create a player with my name and make him really good in rebounds, steals and blocks. Precisely what Cleveland is lacking the most. Then I will trade myself to the Cavs and check out the results:

Create the player: Trade market section on the R dashboard, “Create a player” tab:

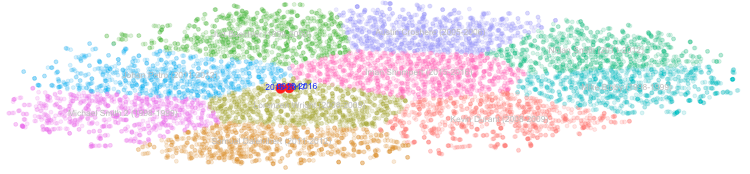
Check the stats dashboard for Cleveland below. Now almost all categories are green and predicted wins go up by 1.3.



Now I know a player like the one I created artificially would boost my wins and complement my team. If I am Cleveland’s manager, next thing I do is look at the closest actual players in the NBA and see if I can pull a trade. Below is the list of 6 closest players to Alberto Sanchez:

Now, putting monetary and contractual variables aside, suppose I short-listed Robert Covington and Larry Nance. Before I make one of them an offer I want to make sure I’m signing the most statistically consistent player. Let’s compare their stats over time.

Robert Covington’s 4 years in the NBA have been a copy of each other statistically:

Meanwhile, Larry Nance’s stats are also quite robust. However, he’s not as close to Alberto Sanchez and his sample size is smaller (only 2 years of experience).

I could continue with the analysis, the possibilities are almost infinite and it’s so much fun to play with it. It’s time to wrap it up.

1. **Conclusions**

I described a statistical model that allows me to predict how good NBA teams will be once I know the composition of their rosters. I have shown how I go about predicting each individual player’s stats and how they combine to determine each team’s offense, defense and number of wins.

I thought it was best to build a narrative around the main results derived from the model so for this purpose I developed an interactive R dashboard to generate the results and simulations. I hope I was able to convey the many possibilities this approach to trades has. I only scratched the surface of what can be done.

The game of trades can be a useful toolkit for managers and scouts to analyze players visually and analytically and either confirm their previous knowledge or discover hidden skills and patterns they didn’t know about. They can use the tool to plug and play players in any team, simulate trades and even create artificial players to understand the type of skills that make their team better and find the most similar in the actual market.

As I’m writing this paper I realized the many aspects I could improve in the future. For instance, including a more detailed analysis of the college and Euroleague basketball ecosystems so scouts can look beyond guys with NBA experience. Also, the way I assess how good offensively and defensively players are is based on neural networks trained on teams so I see some improvements there as well.

**References**

[1] Basketball Reference (2017). Retrieved from: https://www.basketball-reference.com/

[2] Gelman A. (2014). “Basketball Stats: Don’t model the probability of win, model the expected score differential”. Retrieved from: http://andrewgelman.com/2014/02/25/basketball-stats-dont-model-probability-win-model-expected-score-differential/

[3] Silver, McCann (2017). CARMELO NBA Player Projections. Retrieved from: https://projects.fivethirtyeight.com/carmelo/

[4] Alagappan, M. (2012). “From 5 to 13: Redefining the Positions in Basketball” (2012 EOS/Alpha Award Winner). Retrieved from: http://www.sloansportsconference.com/content/the-13-nba-positions-using-topology-to-identify-the-different-types-of-players/

[5] Van der Maaten, L. (2008, 2009, 2012, 2014). T-SNE. Retrieved from: https://lvdmaaten.github.io/tsne/

[6] Lloyd, Forgy (1957, 1965). K-means. Wikipedia. Retreived from: https://en.wikipedia.org/wiki/K-means\_clustering

[7] Wattenberg, et al. (2016). "How to Use t-SNE Effectively". Distill. Retrieved from: http://doi.org/10.23915/distill.00002

[8] Bhattacharyya, A. (1943). Bhattacharyya distance. Wikipedia. Retrieved from: https://en.wikipedia.org/wiki/Bhattacharyya\_distance

[9] Warren S. Sarle, Cary, NC (1997, 1998, 1999, 2000, 2001, 2002). “Should I standardize the input variables?”. Faqs.org. Retrieved from: http://www.faqs.org/faqs/ai-faq/neural-nets/part2/

[10] Sports Reference LLC (2017). Play Index. Retrieved from: https://www.sports-reference.com/cbb/play-index/

[11] Kuhn, M. (2017). The caret package. Retrieved from: https://www.sports-reference.com/cbb/play-index/

[12] Fritsch, Guenther, Suling, Mueller (2016). Training of neural nets. The neuralnet package. Retrieved from: https://cran.r-project.org/web/packages/neuralnet/neuralnet.pdf

[13] The R foundation: The R project for statistical computing. https://www.r-project.org/about.html

[14] RStudio: Open source professional software for R. https://www.rstudio.com

**Appendix A: R scripts**

**A.1. Neural Network model**

1. write\_playersHist.R returns: playersHist.csv
2. .rename\_PlayerName\_Duplicates.R() from rename\_PlayerName\_Duplicates.R
3. .team\_prepareAll() from prepare\_rosters.R
4. .computeModel\_neuralnet() from neural\_network.R

7. neuralnet default parameters:

neuralnet(formula, data, hidden = 1, threshold = 0.01,

stepmax = 1e+05, rep = 1, startweights = NULL,

learningrate.limit = NULL,

learningrate.factor = list(minus = 0.5, plus = 1.2),

learningrate=NULL, lifesign = "none",

lifesign.step = 1000, algorithm = "rprop+",

err.fct = "sse", act.fct = "logistic",

linear.output = TRUE, exclude = NULL,

constant.weights = NULL, likelihood = FALSE)

**A.2. Predicting Player Skills**

**A.2.1. Returning players (cases a and b)**

.computePredictedPlayerStats() from write\_teams\_predicted\_stats\_new\_season.R which returns: playersNewPredicted\_Oct20.csv

The algorithm runs through all players from last season and computes: .predictPlayer() (from similarityFunctions.R). In particular:

* Top 10 similar players: run .similarPlayers() (from similarityFunctions.R): Using .tSNE\_dist()
* Now calculate average variation in their stats when they went from current age to age + 1. Use .tSNE\_prepare from similarityFunctions.R).
* Final precaution, make sure no shooting percentages are > 1: playersNewPredicted <- mutate\_at(playersNewPredicted, vars(contains("Per")), function(x) ifelse(x >=1, quantile(x,.99), x))

**A.2.2. Rookie players (cases c and d)**

The basic algorithm is write\_RookieStats() from write\_rookiesDraft.R which returns rookieStats.csv. Here are the details:

To run the algorithm I need to prepare the following files:

* 1. rookies.csv (writeAllRookies() in write\_rookiesDraft.R): Retrieves all rookies in the current rosters (The Experience field = “R”) with no stats records. I need to match this file with Players in this file:
  2. collegePlayers.csv (write\_CollegePlayers() in write\_rookiesDraft.R)

These 2 files are retrieved from different tables in basketballreference and sometimes their names are spelled differently. I have to manually change these:

collegePlayers[grepl("Nazareth Mitrou-Long",collegePlayers$Player),]$Player <- "Naz Mitrou-Long"

collegePlayers[grepl("Royce O'Neale",collegePlayers$Player),]$Player <- "Royce O'Neal"

collegePlayers[grepl("Jacorey Williams",collegePlayers$Player),]$Player <- "JaCorey Williams"

collegePlayers[grepl("Andrew White III",collegePlayers$Player),]$Player <- "Andrew White"

collegePlayers[grepl("TJ Leaf",collegePlayers$Player),]$Player <- "T.J. Leaf"

collegePlayers[grepl("Frank Mason",collegePlayers$Player),]$Player <- "Frank Mason III"

collegePlayers[grepl("Akim Mitchell",collegePlayers$Player),]$Player <- "Akil Mitchell"

For unmatched players, I will query the European database for players for their stats in either the Euroleague, domestic league or overall and match them with players in rookies.csv.

For those still not matched, if they are international players (playing in Europe or not) they are assigned the means of all the stats of players coming from Europe. For the rest (college players who for some reason were not matched or found) I assign the means of the stats of rookie college players.

**A.2.3. Putting it all together**

current\_rosters.csv contains players and teams as of Otober 20 2017, that is right at the start of the new season and it’s obtained from: .getLatestRosters(thisSeason="2017",previousSeason = FALSE) from write\_teams\_predicted\_stats\_new\_season.R

Now I will merge this file with the previously calculated stats for returning NBA players and rookie players described in 3.5.1 and 3.5.2. If everything worked well, there will only be non-matching players because of spelling differences. In case not, and there are still players who were added last minute after we run 3.5.1 and 3.5.2, then we can run: .computePredictedPlayerStats\_Leftovers() # from compute\_PredictedLeftovers.R.

These are the unmatched players due to spelling differences:

current\_rosters[which(current\_rosters$Player == "Gary Payton II"),]$Player <- "Gary Payton 2"

current\_rosters[which(current\_rosters$Player == "Glenn Robinson III"),]$Player <- "Glenn Robinson 2"

current\_rosters[which(current\_rosters$Player == "Kelly Oubre Jr."),]$Player <- "Kelly Oubre"

current\_rosters[which(current\_rosters$Player == "Nene"),]$Player <- "Nene Hilario"

current\_rosters[which(current\_rosters$Player == "Taurean Prince"),]$Player <- "Taurean Waller-Prince"

current\_rosters[which(current\_rosters$Player == "Tim Hardaway"),]$Player <- "Tim Hardaway 2"

current\_rosters[which(current\_rosters$Player == "Sheldon Mac"),]$Player <- "Sheldon McClellan"

1. https://alberto-sanchez-rodelgo.shinyapps.io/Player\_Dashboard/ [↑](#footnote-ref-1)
2. For details on these stats see for example: https://www.basketball-reference.com/leagues/NBA\_2017\_per\_game.html [↑](#footnote-ref-2)
3. See Appendix A for details on the R code. Also available on my Github: [asRodelgo/NBA](https://github.com/asRodelgo/NBA) [↑](#footnote-ref-3)
4. <http://www.basketball-reference.com/leagues/NBA_2017_per_game.html> [↑](#footnote-ref-4)
5. In Basketball-reference both players have the same exact name so I have to edit it manually [↑](#footnote-ref-5)
6. Total size of NBA rosters as of October 20, 2017 [↑](#footnote-ref-6)
7. The interpretation is: O.J. Mayo is the closest player to Bradley Beal when both were 21, etc. [↑](#footnote-ref-7)
8. effBLK, eff2PA, etc. See: 2. The Data [↑](#footnote-ref-8)
9. Calculated as the mean of the skills of all NBA players from last season (2016-2017) [↑](#footnote-ref-9)
10. As of October 20, 2017. E.g.: https://www.basketball-reference.com/teams/BOS/2018.html [↑](#footnote-ref-10)
11. I use player finder: <https://www.sports-reference.com/cbb/play-index/> [10] [↑](#footnote-ref-11)
12. European players: https://www.basketball-reference.com/euro/players/ [↑](#footnote-ref-12)
13. I froze rosters as of October 20, 2017. There may have been a few trades since then. [↑](#footnote-ref-13)
14. https://alberto-sanchez-rodelgo.shinyapps.io/Player\_Dashboard/ [↑](#footnote-ref-14)
15. This idea is inspired by Ayasdi’s topological data analysis. See [4] [↑](#footnote-ref-15)
16. Usage (playing time) is the input weight to the model: The player’s contribution to each team stat. [↑](#footnote-ref-16)
17. 13.1 = 0.003328252/3936. See equation (2) [↑](#footnote-ref-17)
18. See **3.5.2. Rookie players (cases c and d)** [↑](#footnote-ref-18)