PCA and Sparse PCA for NBA Player Data Analysis

Kentaro Tanaka

December 22nd, 2020

Abstract

This paper will be looking into the classification of NBA all star players based on their season statistics. It will find the best feature transformation and classifier that has the highest accuracy for correctly classifying all star players from their statistics. The second problem the paper will be addressing is if all star players can be separated into groups based on their characteristics. This paper will be focusing on Sparse PCA and PCA feature representations for classification and visualization.

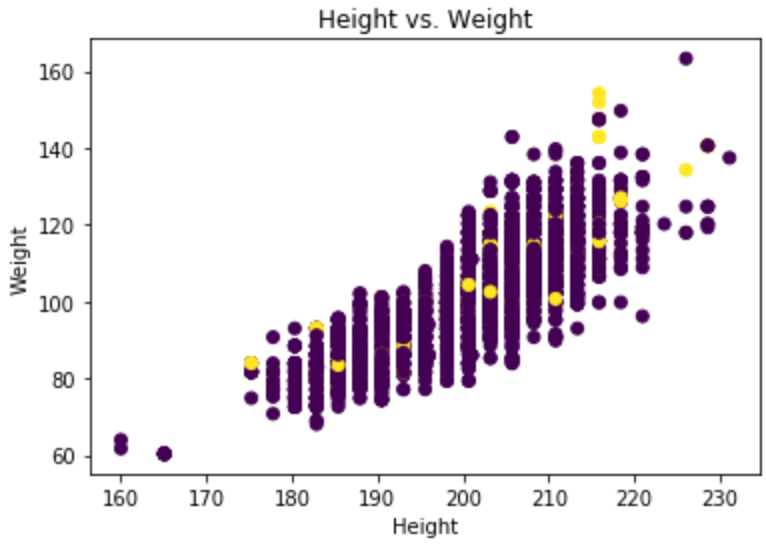
1 Introduction

The National Basketball Association (NBA) is the world's most elite basketball league. It houses the top 400 players on the planet and brings in billions of dollars of revenue yearly. Every year out of the 400 players in the league, 24 players are chosen to be all stars. These all star players are regarded as the best players who contribute most to their teams and to winning basketball games. However every basketball player isn't the same. They play a variety of positions and have different physical attributes such as height and weight. Throughout this project I aim to answer the question: What kind of players become all star players?

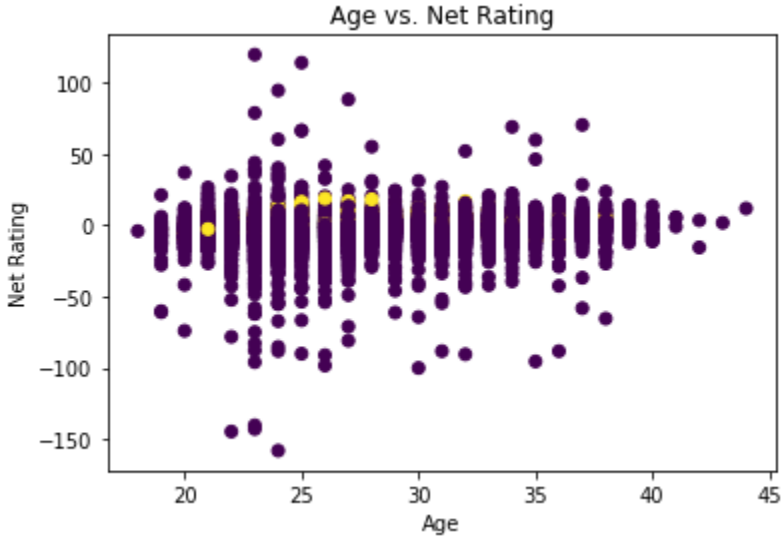
The data being used for this project is two different sets from Kaggle. The first dataset is a dataset based on each player's information and statistics. Examples for the variables included in this dataset are height, weight, points per game (PPT), assists per game (AST), rebounds per game (REB) and many others. The other dataset is information on players who made the all star team. This dataset will only be used to denote whether the player was an all star that season. [2][3]

I took a few steps to clean the data for further analysis. I first dropped variables that I thought wouldn't be related to whether a player made the all-star team that season. Some examples of these variables are: season, draft number, college and country. From there I matched the names from the all-star dataset and made a column of binary values (1 for all-star 0 for not all-star). I will be using this column as the output data to compare my predictions against.

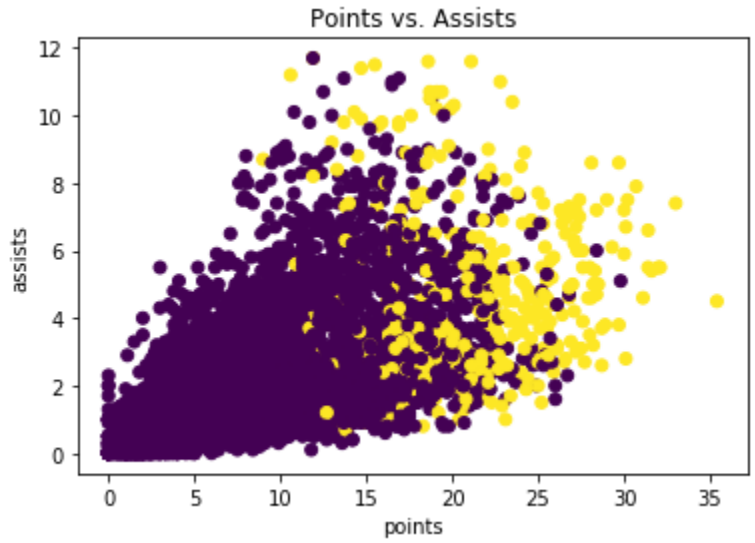
After cleaning the data, I explored the data to see if there are any variables that are correlated with making the all star team. To get started, I first made a scatterplot showing height vs weight.



The yellow dots denote that the player was an all star. The plot shows that being taller and weighing more does not necessarily correspond to making the all star team. The yellow dots are scattered throughout the different heights and weights. The next variable I tried plotting was net rating vs. age. Net rating is how many points the player contributes on average when they are playing.



This plot also shows that the two variables chosen do not necessarily correlate with whether a player was an all star or not. The last set of variables I tried plotting to explore the data would be most perceived to have correlation with whether they made the all star team, points and assists.



As predicted, there are many all star players that make up the right portion of the scatter plot that signifies high amounts of points and assists. However, there are a few outliers, the most questionable one being the yellow dot at around 12 pts and less than two assists on average. These numbers would regularly be one that a regular player in the league averages, how did this player make the all-star team? To answer these questions I did a further analysis of all the variables.

Throughout the project I utilized a dimensional reduction technique that we hadn't gone over in class before. This is Sparse PCA. In order to understand Sparse PCA, it may be helpful to define a sparse matrix. A sparse matrix is one that is composed of mainly zeros. Similarly in Sparse PCA, less variables are represented and more zeros are present. PCA is a linear combination of all the variables whereas Sparse PCA is only a combination of a select few of these. [1]

Some advantages of Sparse PCA is that it may be easier to interpret which variables affect the data the most since all variables aren't being considered. Mathematically, the Sparse PCA optimization problem is similar to the PCA optimization problem except it has one extra constraint. This extra constraint controls the number of non-zero loadings that need to be less than a value k. The higher the k value is, the less variables are part of the linear combination. If the k value is equal to the number of variables, this would be the same as PCA. I will be using Sparse PCA to address both of the problems, utilizing it as a feature transformation and visualization tool. [1]

This project yielded successful results as I was able to successfully classify what kind of players become allstars through classification and clustering. I will first highlight the problems that need to be solved, go over the method in which they were solved, and lastly go over the results and proper conclusions.

2 Problem

Throughout this project there are two problems that I try to solve.

The first problem is, which feature transformation and classifier can I use on player statistics to predict whether a player becomes an all star that season? To answer this problem we will be looking for a feature transformation and classifier combination that is able to predict whether a player becomes an all star at a high accuracy rate.

The second problem is, what kind of players become all stars? Can we categorize different types of all star players? This problem will be successfully answered if I am able to find clear clusters within the all star players data.

2.1 Methodology

To solve the first problem, I will be creating different pipelines that are combinations of feature transformations and classifiers. The feature transformations that I will be considering are Standard Scaler, PCA and Sparse PCA. The classifiers that I will be using are both binary. They are logistic regression and support vector classifier. I will take combinations of these feature transforms and classifiers and create several pipelines.

From there I will optimize for hyperparameters by using grid search cross validation. Hyperparameters that I will be optimizing for Logistic Regression are C, solver and penalty. The hyperparameter optimization on C will be particularly important to avoid overfitting. The parameters that I will be optimizing for the support vector classifier are C and gamma to avoid overfitting. After optimizing each pipeline, I will compare them based on their accuracy scores to find the best feature transformation classifier combination. The results will be observed to see if there is a classifier that can be fed player statistics and make a good prediction.

To solve the second problem, I will be utilizing PCA and K-means clustering. First I will get the first two principal components of the data and create a scatterplot visualizing the first component on the x-axis and the second component on the y-axis. From there I will run K-means clustering and experiment with a different number of clusters to see which number classifies the players in enough detail but not too much detail that the clusters have few players.

3 Results

To solve the first problem (choosing best feature transformation and classifier), I computed accuracy scores and runtime for different optimized pipelines. The results for accuracy and runtime are shown below.

Accuracy Values for Different Classifiers and Transformations

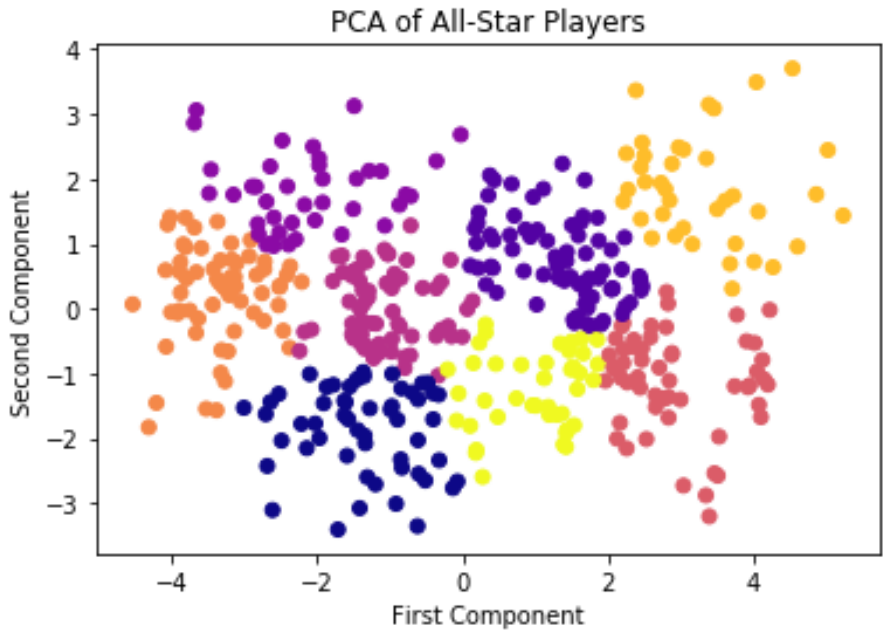
Feature Transformations	None	Standard Scaler	Standard Scaler & PCA	Standard Scaler & Sparse PCA
Logistic Regression	96%	95.94%	-	96.07%
SVC	-	96.07%	96.13%	96.13%

Runtime for Different Classifiers and Transformations

Feature Transformations	None	Standard Scaler	Standard Scaler & PCA	Standard Scaler & Sparse PCA
Logistic Regression	128 ms	1160 ms	-	5150 ms
SVC	-	311 ms	278 ms	4740 ms

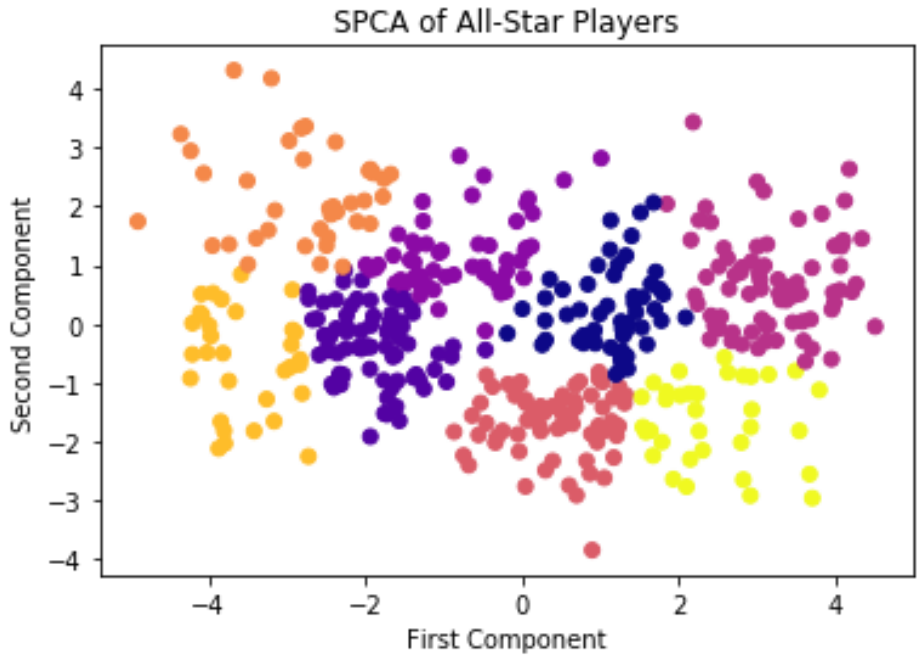
Clustering Results:

I ran k-means clustering on both PCA and Sparse PCA representations of the data. For both of the methods, I created a scatter plot of the first two principal components.



The scatterplot above is a scatter plot displaying the first two principal components of regular PCA. After experimenting with a different number of clusters, I discovered that based on positions, and many other attributes of players, eight clusters divided player types well while still having a good number of players in each cluster. Next, I will be going over some of the notable clusters.

Players were successfully classified and split into clusters through PCA. Next, we looked into whether separating the all star players based on Sparse PCA would improve our clustering.



Some notable differences from PCA is that the Sparse PCA scatterplot has more values spread out to the upper left portion of the scatterplot. The points in each of the clusters seem to be closer to each other also, especially the ones closer to the middle of the scatterplot.

When clustering with Sparse PCA, the clusters characterized the players quite well also. To compare I will be comparing the Sparse PCA clusters and the PCA clusters.

4 Discussion

Discussion for Problem 1 (Feature Transformation and Classifier):

The accuracy of the pipelines were quite similar. Most of them had an accuracy of over 96% which shows that the pipelines are quite good at predicting whether a player makes it to the all star team. The pipeline with the best accuracy of 96.13% was Standard Scaler and PCA with the SVC classifier. Although it was only better by .13%, therefore all of the pipelines can be seen as great classifiers.

The runtime for each of the pipelines was drastically different. The pipeline that cost the most were the Sparse PCA ones by far. They both took around 5 seconds where the lower cost standard scaler and PCA pipelines only cost 200-300 ms. This is a large difference in computation cost.

Discussion for Problem 2 (Clustering):

I will be reviewing clusters from both the PCA and Sparse PCA clustering to discuss the results for the second problem.

PCA Cluster 1 - High Scoring Point/Shooting Guards

player_name	age	player_height	player_weight	gp	pts	reb	ast
Grant Hill	27.0	203.20	102.058200	74	25.8	6.6	5.2
Kobe Bryant	22.0	200.66	95.254320	68	28.5	5.9	5.0
Vince Carter	24.0	198.12	102.058200	75	27.6	5.5	3.9
Tracy McGrady	22.0	203.20	95.254320	77	26.8	7.5	4.6
Paul Pierce	24.0	198.12	104.326160	82	26.1	6.9	3.2

This cluster was made up of historical high scoring point guards and shooting guards who also had a moderate average of rebounds and assists. Their heights range from 195-200 cm.

PCA Cluster 5 - Dominant Power Forwards/ Centers

player_name	age	player_height	player_weight	gp	pts	reb	ast
David Robinson	34.0	215.90	113.398000	80	17.8	9.6	1.8
Alonzo Mourning	30.0	208.28	118.387512	79	21.7	9.5	1.6
Shaquille O'Neal	28.0	215.90	142.881480	79	29.7	13.6	3.8
Tim Duncan	24.0	213.36	117.933920	74	23.2	12.4	3.2
Tim Duncan	25.0	213.36	117.933920	82	22.2	12.2	3.0

This cluster characterizes power forwards and centers who were dominant in the “paint”. The “paint” means the area near the basketball hoop and scoring through layups or dunks. These players are very tall, 210-215 cm and very heavy 117+kg. They also score in heavy volumes and have historically high rebound averages (above 10 per game on average).

PCA Cluster 7 - Defensive Centers

player_name	age	player_height	player_weight	gp	pts	reb	ast
Dikembe Mutombo	34.0	218.44	118.387512	82	11.5	14.1	1.3
Antonio Davis	31.0	205.74	104.326160	79	11.5	8.8	1.3
Vlade Divac	32.0	215.90	117.933920	82	12.3	8.0	3.0
Theo Ratliff	27.0	208.28	102.058200	57	11.9	7.6	0.6
Dikembe Mutombo	35.0	218.44	120.201880	75	10.0	13.5	1.0

Cluster 7 shows the outlier that was referred to earlier. It was questionable how a player who only averages 12 points and less than two assists could possibly make the all star team. However, this cluster explains this outlier. Dikembe Mutombo is the exemplary player for this cluster. The player was a defensive threat with very high rebound averages. Although his offensive output wasn’t high, he is known as a very high level defensive player, thus making the all star team based on defensive qualities.

Sparse PCA Cluster 1 - High Scoring Power Forwards

player_name	age	player_height	player_weight	gp	pts	reb	ast
David Robinson	34.0	215.90	113.398000	80	17.8	9.6	1.8
Antonio McDyess	25.0	205.74	99.790240	81	19.1	8.5	2.0
Alonzo Mourning	30.0	208.28	118.387512	79	21.7	9.5	1.6
Chris Webber	27.0	208.28	111.130040	75	24.5	10.5	4.6
Karl Malone	36.0	205.74	116.119552	82	25.5	9.5	3.7

By looking at the cluster numbers it was made clear that they weren't the same numbering as PCA. Cluster 1 for Sparse PCA seemed to be made up of high scoring power forwards. This has the closest relation to cluster 2 or 5 from PCA. Therefore, it is clear that the numbers are different, but the classifications still separated the player types well.

Sparse PCA Cluster 5 - Defensive Centers

player_name	age	player_height	player_weight	gp	pts	reb	ast
Dikembe Mutombo	34.0	218.44	118.387512	82	11.5	14.1	1.3
Antonio Davis	31.0	205.74	104.326160	79	11.5	8.8	1.3
Vlade Divac	32.0	215.90	117.933920	82	12.3	8.0	3.0
Theo Ratliff	27.0	208.28	102.058200	57	11.9	7.6	0.6
Dikembe Mutombo	35.0	218.44	120.201880	75	10.0	13.5	1.0

Cluster 5 for Sparse PCA was similar to cluster 7 for PCA. This cluster characterizes the outliers of the all star players who are very good at defense but have a low offensive output.

Sparse PCA Cluster 6 - Dominant Power Forwards / Centers

player_name	age	player_height	player_weight	gp	pts	reb	ast
Shaquille O'Neal	28.0	215.90	142.881480	79	29.7	13.6	3.8
Tim Duncan	24.0	213.36	117.933920	74	23.2	12.4	3.2
Alonzo Mourning	31.0	208.28	118.387512	13	13.6	7.8	0.9
Tim Duncan	25.0	213.36	117.933920	82	22.2	12.2	3.0
Shaquille O'Neal	29.0	215.90	142.881480	74	28.7	12.7	3.7

For the last Sparse PCA cluster that I wanted to show, I wanted to find the cluster that was similar to the PCA cluster 5. For Sparse PCA it is cluster 6 and I chose this cluster to show the similarity between the PCA classifications and the Sparse PCA classifications. I wasn't able to find major differences in how the clusters were characterized. They mainly had the same clusters except were numbered differently. Therefore, both Sparse PCA and PCA are both valid methods for clustering and classifying what kind of players become all stars.

Other classifiers that could have been used in pipelines are tree based algorithms. These include decision trees and random forests. These may have also resulted in good accuracy scores, but generally logistic regression classifiers have better accuracy rates and take less computation cost. Especially with random forests, the classifiers that we used here are computational lighter. Although there are other algorithms that we could have used and can compare to, the ones we used here for binary classifications make the most sense I believe.

In terms of comparisons with other algorithms for clustering, spectral clustering could have also worked instead of K means clustering. However, by seeing the data, using K-means to separate the data makes more sense since using the distance between points would result in similar results. Another way to visualize the data better, or group them better was to use three components and use 3 dimensional visualizations. While working on this project I saw this as a possibility but did not explore it due to the time constraints. I believe that using three principal components for grouping would have made better clusters, but to visualize it is a complex process to complete.

5 Conclusion

Overall the best pipeline in terms of both computation cost and accuracy was PCA with the SVC classifier. This pipeline had significantly less computation cost than the Sparse PCA version of it and had just as good accuracy. The logistic regression with no feature transformation was also a good option since it has the lowest computation cost and still has a fairly high accuracy (near 96%).

For our clustering, clustering off of the first two components of both PCA and Sparse PCA are valid methods. However, as seen above the computation cost for Sparse PCA is higher, so in this case it may be better to use PCA since the results are similar.

In general, which players make the all star team can be predicted from player statistics at a high accuracy and all star players can also be characterized into different groups.

References

[1] Zou, Hui. *Sparse Principal Component Analysis* , 2006, web.stanford.edu/~hastie/Papers/spc_jcgs.pdf.

[2] Mejia, Felix. "NBA All Star Game 2000-2016." *Kaggle*, 4 Feb. 2020, www.kaggle.com/fmejia21/nba-all-star-game-20002016.

[3] Cirtautas, Justinas. "NBA Players." *Kaggle*, 8 Mar. 2020, www.kaggle.com/justinas/nba-players-data.