NBAPositionify: Leveraging Data Mining Techniques to Classify Professional Basketball Players into Positions

Term Project Report *for CS 2756: Principles of Data Mining*

SHINWOO KIM, ROBBIE FISHEL, and BIRJU PATEL

1 BACKGROUND AND MOTIVATION

In professional basketball, the role of a player in a game is not solely defined by their physical stature or position on the court. Rather, it is a dynamic interplay of various skill sets, athletic abilities, and tactical understanding. Traditionally, players are categorized into the positions of point guards (PG), shooting guards (SG), small forwards (SF), power forwards (PF), and centers (C), as shown in figure 1. However, as basketball has evolved, the delineation between these positions has become increasingly blurred.

In recent years, many teams have introduced data analysis techniques for evaluating players and determining team strategies. By harnessing the power of data, teams gain deeper insights into player performance, and they can optimize their lineup on-the-fly in order to enhance overall team efficiency.

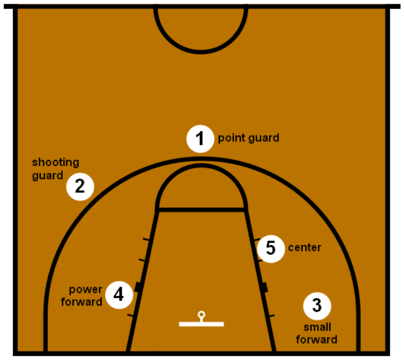


Fig. 1. Basketball positions with the numbers as they are known: (1) Point Guard, (2) Shooting Guard, (3) Small Forward, (4) Power Forward, and (5) Center. Illustration from Wikipedia.

2 PROBLEM STATEMENT

One such application of data analytics in basketball is the classification of players into specific positions based on their attributes. Each position on the court requires a different set of skills, and players gravitate towards the positions that best match their style. As an example, taller players typically play closer to the net, where their height makes it easier for them to get rebounds and make dunks. And often, a player will start his career in one position and transition to another over time. In recent years, several prominent players have followed this trajectory. Kevin Durant, who joined the league in 2008, started as a shooting guard before later transitioning to become a power forward. Giannis Antetokounmpo, who started in 2016 as a point guard, also became a power forward later in his career.

Both of these players spent the first few years of their careers playing in a suboptimal position. Since player salaries run in the millions of dollars, coaches want to get the highest return on their investment by putting players in the position where they will perform the best. Our project attempts to examine decades of historical data in order to develop a classification model capable of accurately assigning players to their respective positions. Using this model, NBA coaches would easily be able to use a prospective player's career statistics to estimate which position they would be best suited for.

Prior to this project, we expected certain trends—like forwards being generally taller than guards—to be reflected in the data. We presumed that we would likely discover more trends as our analysis progressed. One goal of this project was to visualize such trends by conducting clustering analysis on our data. Our aim is to show how different clusters of players vary according to different statistics. In addition, we also identify outliers that do not fit into any clusters; we predict that these outliers may correspond to certain notable players.

3 RELATED WORK

Works such as [1] and [5] use in-game spatial and topological data in order to classify players into positions. Unfortunately, these work do not incorporate player statistics into their decisions. There is preliminary work by Song and Wang that uses naive clustering methods in order to fit players to positions [6]; however, their work is limited to NBA players in the 2015-2016 season only. We conduct a more overarching analysis with data that spans many more seasons.

4 METHOD

4.1 Classification of Players by Position

Classification, the task of assigning objects to one of several predefined categories, is a pervasive problem that encompasses many diverse applications. In the case of our problem, the goal is to categorize basketball players into positions (point guard, shooting guard, small forward, power forward, or center) based on their various features and season statistics. To accomplish this goal, we consider various classification methods, outlined below.

4.1.1 Decision Trees. A decision tree is a simple but popular classification technique. In this technique, the tree has a flowchart-like structure, where each internal node represents a "test" on a feature, each branch represents the outcomes of the test, and the leaf nodes represent a class label. An example of a decision tree is shown in figure 2.

Decision trees are useful since they are easy to understand and interpret. They can also handle both numerical and categorical data, which makes them particularly useful for this project. Unfortunately, decision trees are prone to overfitting if not pruned properly, meaning they might learn the training data too well and perform poorly on unseen data.

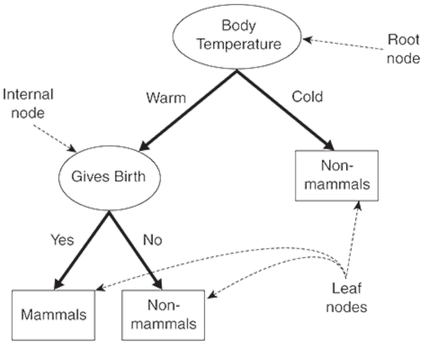


Fig. 2. An example of a decision tree for classifying mammals.

4.1.2 Boosted Trees. Boosting is an ensemble technique that combines the predictions from multiple decision trees to create a more powerful model. It employs an iterative procedure to change the distribution of training data by focusing more on previously misclassified records. This is accomplished by assigning weights to each training example, and are adaptively changing them at the end of each boosting round.

Boosted trees are highly flexible and powerful models that can capture complex relationships in the data. Furthermore, boosted trees tend to perform very well on a wide range of datasets and are less prone to overfitting compared to individual decision trees. For this project, we use a popular Python-based implementation of boosted trees called *XGBoost* which uses gradient descent to optimize the loss function [3].

4.1.3 Logistic Regression. The logistic regression classifier works by learning a simple linear function from the dataset to map input features to output classes. When given a training example, the model calculates the difference between the output from its current weights and the true class. It then uses this loss to modify its initial weights. This is done iteratively for each example in the training set.

4.1.4 Random Forests. Random Forest is another ensemble technique that attempts to improve the generalization performance by constructing an ensemble of "decorrelated" decision trees. In other words, the random forest technique creates multiple decision trees during training and combines their predictions to produce a more accurate and robust model. Random Forests are advantageous in that they provide reduced overfitting, show better generalization performance, and provide a measure of feature importance.

4.2 Clustering

Clustering is a data mining technique meant to group similar data points together into groups, or clusters. Given the nature of our goal being to group players within their positions (i.e., to find similarities between players who are in certain positions), clustering was an avenue which we saw as having potential for identifying and confirming what attributes best determine player position. Further, these clusters could give us a good sense of outliers to these normal trends seen. For instance, if a player who is a small forward is classified within the point guard cluster, this player could be seen as an outlier.

Two clustering algorithms were chosen to run experiments on: **K-Means** and **Agglomerative** clustering. We chose K-Means as a baseline algorithm, with the underlying assumption that players with certain relative stats would be close to one another if they play the same position. Having "centroids" representing the position groups seems to be a solid foundational baseline for determining what group players belong in. Agglomerative clustering was chosen as an alternative given the nature of positions in basketball becoming relatively more fluid in recent years. Agglomerative clustering is hierarchical in the sense that it starts with one large cluster, and narrows itself down into smaller, more distinct clusters with each clustering iteration. We want to see how similar players of similar positions are, while also finding distinct groups for them. We found agglomerative clustering to be successful in accomplishing this task and giving us another clustering perspective.

To best determine how many clusters to group our data in, we used as our metric Within Cluster Sum-Of-Squares (WCSS) which determines how close together the points within a cluster are; a lower WCSS value means more tightly packed clusters. We ran k-means clustering on 1 through 5 clusters in separate iterations to determine the amount of clusters which would be optimal, which we then used to perform our experiments and analysis for both k-means and agglomerative.

A subset of features was chosen from the dataset to perform the k-means clustering to maximize the likelihood of identifying a correlation between cluster and position. This subset contained the most important features found for determining position from our feature analysis. Any column which had more than a 5% relation to position was included for clustering. Features included in the subset were *total rebound percentage*, *total assist percentage*, *height* and *weight*. Data normalization was also to be performed before clustering, where each of the features chosen to train the clustering on would be subtracted by the column's mean and then divided by the standard deviation. This made it so that all the different features, despite them containing different values and being within different ranges, were on similar scales.

5 EXPERIMENT AND RESULTS

5.1 Dataset

We construct our classifiers and perform clustering analysis based on data from the Basketball Reference [4], a website that collects and aggregates player and season data. The data we use spans NBA seasons from 1950 to 2016 and has information on 4,500 different players. In addition, the dataset contains each player's individual season statistics, with a total of 24,691 player-season data.

5.2 Exploratory Data Analysis

The dataset we used is split into separate "player" data and "season" data files. The player data file contains the player's name, height, weight, years played, college attended, date of birth, and birth city/state. The season data file contains over 50 game statistics including percentage of assists, number of blocks, points scored, and most importantly, the position they played. Since we wanted to perform our analysis on all possible features, we needed to merge the two datasets into one.

In addition, due to a lack of record-keeping in earlier seasons (circa 1950), many records in the dataset contained null values, as shown in table **??**. These null-valued records proved problematic, as they could not be used with many classification and clustering algorithms. Thus, in our data preprocessing stage, these records needed to be removed.

Year	Player	Pos	Age	Tm	G	GS	MP	...	DRB	TRB	STL	BLK	TOV	PF
1950	Curly Armstrong	G-F	31	FTW	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	217

Table 1. An example row in the dataset

Similarly, the representation of the "position" column was inconsistent throughout the dataset. While most entries contained only one position, for players who played as multiple positions in one season the data was concatenated into one column (e.g., "G-F" for a player who played as both a guard and forward in one season). In addition, some position labels lacked specificity, writing merely "G" (guard), rather than specifying "PG" (point guard), or "SG" (shooting guard). Thus, in the data preprocessing stage, hyphenated positions needed to be separated in to multiple data rows and generic position labels were extended into their more specific classes. While this did increase the likelihood of misclassification, it allowed us to represent a player that is capable of playing in more than one position in a usable format.

5.2.1 Feature Selection. As part of our data exploration, we attempted to naïvely fit classifiers, mainly a decision tree, with little to no restrictions. Unfortunately, this proved meritless, as we were left with a very overfit tree which did not reveal much.

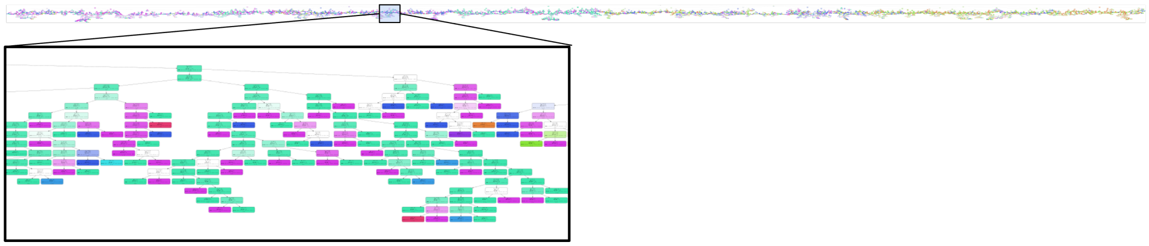
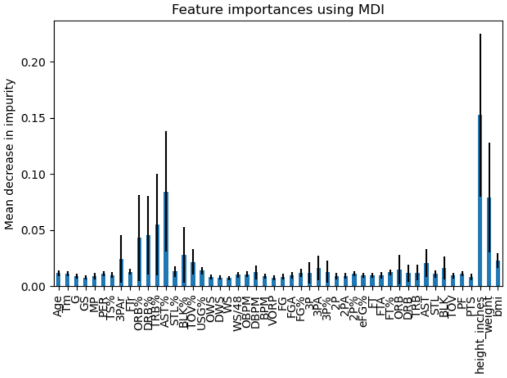


Fig. 3. Overfitted Decision Tree

Thus, we attempted to perform feature selection, in order to identity the most relevant predictors for player positions. To do this, we used a random forest classifier and ranked each feature based on their *Mean Decrease in Impurity*. The results of this analysis are given in figure 4.



Features	MDI
Height	0.150492
AST%	0.085989
Weight	0.079264
TRB%	0.054982
DRB%	0.044862
ORB%	0.043170
BLK%	0.028269
3PAr	0.025008
BMI	0.023135
TOV%	0.021247

(a) Top 10 Most Important Features

Fig. 4. Features Importance based on Mean Decrease in Impurity

And indeed, the importance of this features can be explained with a general understanding of the game. Players with high assists (the "quarterback" of offense) are typically the ones that call the plays and have control of the ball most in an offensive posessions. Hence, they are likely to be point guards. Power Forwards and Centers usually play close to the rim, ready to grab rebounds at a higher rate than all other players. Thus, it makes sense that that the total rebound percentage (TRB%), as well as offensive and defensive rebound percentage (ORB% and DRB%) are relevant to position. Finally, there is a clear hierarchy in height and weight from Point Guards to Centers, with Point and Shooting Guards being generally shorter and smaller than power forwards and centers.

5.3 Classification Performance

After running the various classification techniques on our dataset and attempting to optimize each model in order to get better performance, we found decision trees to be the worst performing model, as shown in table 2. This was as expected, since decision trees are usually simpler models that are not the most effective at identifying fine-grained differences between training samples.

Model	Decision Tree	XGBoost	Logistic Regression	Random Forest
Accuracy	68.78%	70.77%	76.80%	78.01%

Table 2. Classifier Performance Summary

On the other hand, the best performing models were logistic regression and random forest, with random forest being slightly better overall. We suspect that the random forest model, which works by learning a series of linear decision boundaries, had high performance, since the data is clustered by position. One surprising result was the performance of XGBoost, which was significantly lower than the Random Forest model. Since the two methods are similar decision tree based ensemble techniques, we expected them to show similar performance. However, this was not the case. We suspect that this is because the Random Forest classifier can better handle feature importance analysis, to minimize the impact of low-importance features.

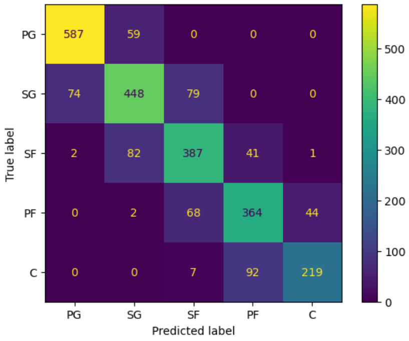


Fig. 5. Confusion Matrix for Random Forest Model

In addition, we found that the classification model typically confuses players with similar positions. That is, the model often misclassifies a shooting guard for a point guard, but rarely misclassifies it as a forward. This suggests that the data is clustered, and that guards are typically grouped into one cluster, while forwards are grouped into another.

5.4 Key Findings from Classification Methods

To extract useful "heuristic rules", we constructed a decision tree of depth 3 that can be used to summarize our findings, as shown in figure 6 . Such heuristic rules could prove important since they can directly and easily be applied by coaches and scouts on the field. Some key takeaways are summarized below:

- Point Guards and Shooting Guards are generally shorter.
- Power Forwards, Small Forwards, and Centers are generally taller.
- Point Guards have more assists than Shooting Guards.
- Power Forwards make more rebounds than Small Forwards.

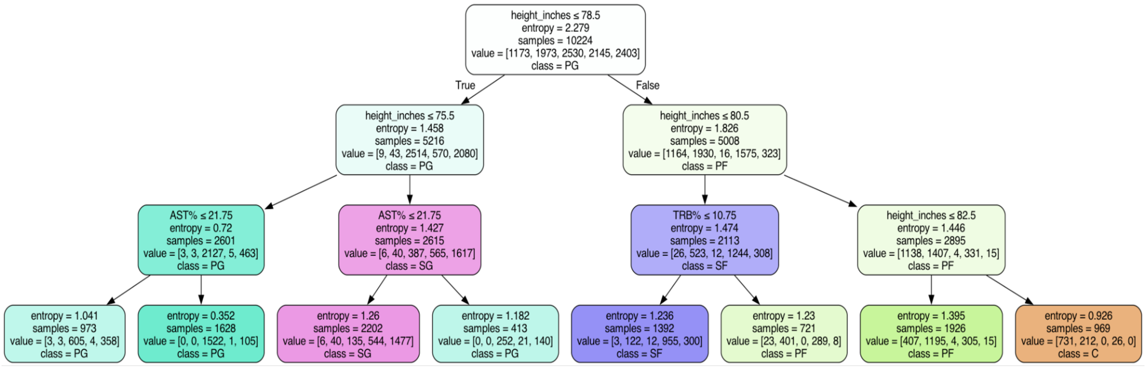


Fig. 6. A Simple Decision Tree Summarizing Key Findings.

5.5 Clustering

5.5.1 WCSS Analysis. Before we performed clustering, our first experiment was to determine the optimal number of clusters to use based on WCSS. To do so, we ran the k-means clustering algorithm for all number of clusters between 1 and 5, and calculated WCSS and observed the results. We chose 5 as the upper-bound for our clusters, since there are 5 distinct basketball positions. Our intuition was that selecting a number of clusters greater than 5 would surely overfit the data.

From this experiment, we determined that 5 clusters best fits the data. 3 and 4 clusters appear to be similar in terms of WCSS, indicating that we might be able to group positions into 3 or 4 larger groups instead of

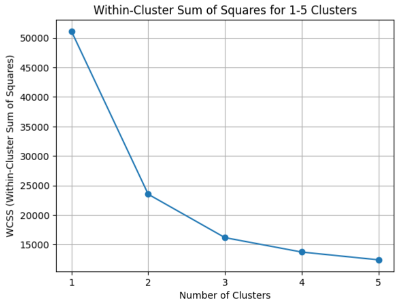


Fig. 7. A graph representing the Within Cluster Sum-of-Squares (WCSS) of 1 through 5 clusters from K-Means clustering.

the five traditional positional categories. However, for the purposes of attempting to differentiate players into the traditional 5 positions, this is the number of clusters we moved forward with in future experiments.

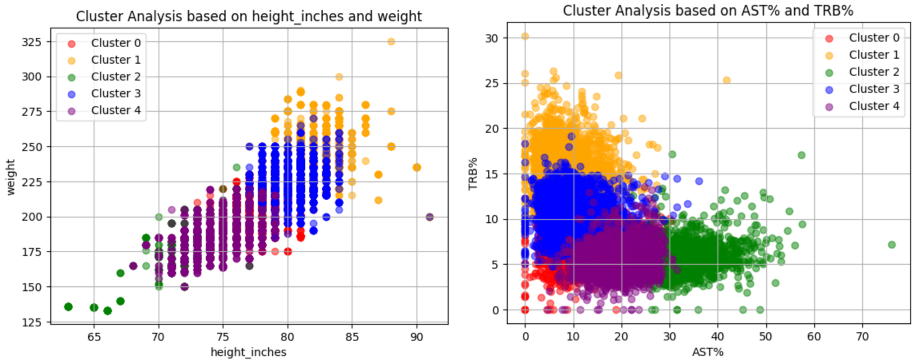


Fig. 8. Visual representation of k-means clusters based on the attributes clustering was performed on.

5.5.2 K-Means. Since we already performed k-means clustering with 5 clusters as part of the WCSS experiment, we examined these clusters for further analysis. We constructed a visual representation of the positions of players within each cluster, as well as visualizing the relationship between the variables used for clustering and what cluster each player ended up in.

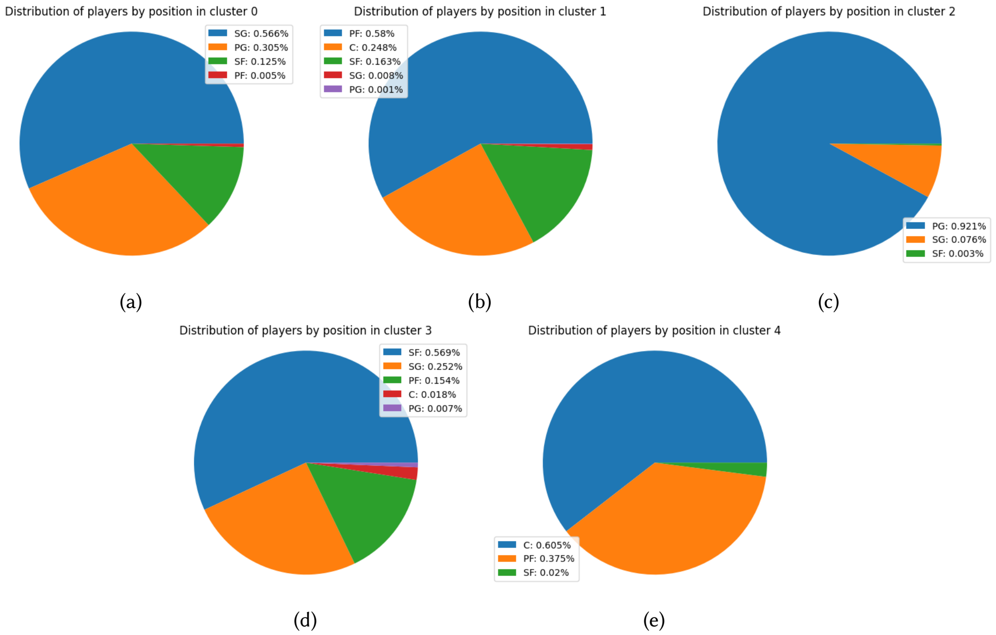
Each cluster contained mostly one specific position, except for the third cluster, which contained 49% small forwards and 38% power forwards. Clusters 2 and 4 both contained a majority of point guards, and no clusters contained a majority of centers. The clusters were relatively pure to one position each, but these specific cases meant that there were gaps left to be desired in using clustering as a way of grouping players together by position.

The positive to the k-means approach is, as seen in figure 8, there is a clear hierarchy when it comes to clusters and relevant statistics. As height and weight increase, the cluster that the data points are grouped in changes linearly. Similarly, high rebound percentage and low assist percentage are grouped in different clusters than high assist percentage and low rebound percentage.

5.5.3 Agglomerative. Similar to k-means, we ran the agglomerative clustering algorithm with 5 clusters on the normalized dataset, and generated the same visualizations as k-means. We found that the relationship between the clusters and positions were more “pure” when creating the same visualizations as were created with the k-means clusters. As seen in Figure 9, each of the clusters has above 50% of records which are associated with one of the five specific positions, and each cluster’s position which has the majority of records is unique as compared to the other clusters. That is to say, there is 1 majority PG cluster, 1 majority SG cluster, etc. In other words, it is more likely for a player to be grouped with other players at their true position in agglomerative clustering than k-means.

5.6 Outliers

To determine major outliers in our clustering, we used visual observation of the clusters—specifically the position which appeared the least in each cluster. Below, we detail some notable players who appeared in a cluster which was not associated with their primary listed position.



(a) (b) (c)

(d) (e)

Fig. 9. The distribution of positions generated by the five clusters using agglomerative clustering.

Kevin Durant, 2016 and 2017. Clustered with C, but true position was SF. Durant was drafted out of college in 2008 as a shooting guard, and over time as he grew taller and stronger transitioned to Small Forward and Power Forward role. In 2016 and 2017, the two seasons in question, Durant averaged 8.2 and 8.3 rebounds per game, respectively. In 2016, this was good for 25th in the league in this category. He is a good shooter, but also does not produce many assists. His height, high rebounds and low assists gives solid evidence to him being misclassified in this case, and his elite scoring ability and athleticism, not taken into account by our clustering model, might be a reason why his true position is instead a small forward.

Jalen Rose, 1995. Clustered with PG, but true position was SF. Rose had the build of a forward, as he is the tallest and biggest player out of the 6 players to be misclassified as a PG while being an SF. However, his 33.3% assist percentage in his 1995 rookie year is what contributes to this misclassification. Rose was known to be a relatively good passer for someone in his position, and this is likely the main contributor to him being in the PG cluster. Later on in his career, Rose would go on to play some PG and SG from time to time, bringing further merit and reason behind this clustering.

Dwyane Wade, 2004, Clustered with SF, but true position was PG. In his prime, Wade was one of the best shooting guards in the game; a well-rounded scorer who had large enough of a build to be potentially confused with a small forward. The rookie year in which he was clustered with SFs was the only year in which his true position was not a shooting guard, and shooting guards and small forwards can often be interchanged for one another - especially if a shooting guard has a larger build, or can be considered more of a "slasher" (a perimeter player with the ability to drive to the basket).

6 CONCLUSION

In summary, we used several different data mining techniques—most notably classification and clustering—to help in guiding our understanding of professional basketball positions and what players fall into what groups. We built several classification models with high performance to place players given their statistics into the five positional categories. We also performed feature selection to determine the importance of each feature in predicting position. From this, we were able to construct a simple decision tree that can be used as a heuristic for coaches and scouts to place players in the position that would potentially best suits their skills. Lastly, we also used the selected features to cluster players for further analysis, and for determining noteworthy outliers.

A CONTRIBUTIONS & DIVISION OF WORK

Shinwoo Kim was responsible for preliminary exploratory data analysis (as described in section 5.2). Shinwoo Kim and Birju Patel worked collaboratively on implementing and analyzing the classification methods described in section 4.1. Robbie Fishel was responsible for clustering analysis (as described in section 4.2) and for outlier analysis. Importantly, Robbie Fishel provided background expertise on professional basketball. All team members equally contributed to drafting and revising this paper, and preparing for the presentation.

B PROJECT CODE

The code used during the production of this report is publicly available, and can be found on-line at https://github.com/shinwookim/NBAPositionify. For the original dataset, see [2].

REFERENCES

[1] Muthu Alagappan. [n. d.]. From 5 to 13: redefining the positions in basketball. https://www.sloansportsconference.com/research-papers/from-5-to-13-redefining-the-positions-in-basketball

[2] BasketballReference. 2000. Basketball Statistics and History. https://www.basketball-reference.com.

[3] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Francisco, California, USA) *(KDD '16)*. ACM, New York, NY, USA, 785–794. https://doi.org/10.1145/2939672.2939785

[4] Omri Goldstein. 2018. NBA Players Stats Since 1950. https://www.kaggle.com/datasets/drgilermo/nba-players-stats.

[5] Stephen M Shea. 2014. *Basketball analytics : spatial tracking*. Amazon Books.

[6] Tao Song, Zibo; Wang. 2017. Classification of NBA Players. https://chaspari.engr.tamu.edu/wp-content/uploads/sites/147/2018/01/2_7-1.pdf.