NBA Roster Statistics, Similarity, Segmentation Prediction

MAST 679 - Sports Analytics

Aymen Rumi
27879229

December 6, 2021

Abstract

Estimating how a group of players will mesh together is a ongoing challenge among team sports. With recent advancements in machine learning algorithms and abundance of team and player statistics available, we can now approach player based team metric estimation numerically. This paper presents a Convolution Neural Network based approach to predicting team statistics given a set of hypothetical NBA players. Historical NBA teams from 1997 to 2020 are analyzed and clustered into categorical segments. Sets of neural networks are used to associate hypothetical teams to segments as well as identify similar historical teams for the hypothetical team. A Python module is presented for users to conduct hypothetical roster analysis.

1 Introduction

Team sports work under a different set of rules than individual sports. Leagues have teams which operate as businesses that have partial control over their players. Players from team sports sign contracts to play for a team under a specified duration of time. Player movement is not uncommon in team sports, players often move around for a variety of reasons. General managers may trade players away for other players, players may get released by their teams, and teams can sign free agents from the player market.

In basketball, singing or trading for dominant players is a no-brainer. A high performing player in this sport always changes the dynamics of a team; however, given that teams are constraint by salary caps, a salary budget set by the league that each team must stick to, general managers face great difficulty when deciding which players to sign or trade for. When looking for new players to add to the roster, often times estimates on how new players or new a roster will perform are solely assessed with the "eye test", a way to judge an athlete based on intuition and observations. This is a highly biased practiced as it is prone to human error, thus a more objective assessment would greatly aid in the team roster building decision making process.

With abundance of player and team statistics available in the NBA, can quantitative analysis be leveraged to produce a statistical model-based team assessment method that removes human judgement? In this paper I will be presenting an algorithmic approach to quantitatively assess hypothetical NBA team rosters. Given a hypothetical roster of player, my algorithm will estimate.

i Team statistics

ii Team segmentation

iii Team similarity with historical NBA teams

As the great statistician George E. P. Box has said, "all models are wrong but some models are useful". The algorithm presented in this paper is designated to help aid in the decision making process with quantitative assessment. I will give an overview of my algorithm in 2, a Python module will be presented for users to conduct their own statistical roster analysis in 2.5. Experiments and results will be shown in 3, lastly limitations and areas of improvements will be discussed in 4.1.

2 Methodology

The model presented in this paper uses a convolutional neural network approach to predict team statistics and segments, using player data as input. I will break down my methodology into 3 parts.

i Clustering 2.2

ii Convolutional Neural Network 2.3

iii Sampling 2.4

2.1 Data

Data is gathered using a web scaper, data gathering functions are shared on the Python module discussed in 2.5. NBA team data and NBA player data from 1999 - 2020 is used for analysis in this paper. Individual team and individual player data are both average statistics for the regular season, a combination of traditional statistics and advanced statistics is used. Sample data with sample metrics are shown in 2.1.1 and 2.1.2.

2.1.1 Team

TEAM	YEAR	GP	W	L	WIN%	MIN	PTS	FGM	FGA
Atlanta Hawks	1999-00	82	28	54	0.341	48.4	94.3	36.6	83.0
Boston Celtics	1999-00	82	35	47	0.427	48.1	99.3	37.2	83.9

2.1.2 Player

PLAYER	YEAR	POS	GP	MIN	PTS	FGM	FGA	FG%	3PM
Shaquille O'Neal	1999-00	C	79	40.0	29.7	12.1	21.1	57.4	0.0
Allen Iverson	1999-00	SG	70	40.8	28.4	10.4	24.8	42.1	1.3

2.2 Clustering

All NBA Teams from 1999 to 2020 were analyzed to highlight statistical differences between them. Dimensionality reduction with principal component analysis was performed on 37 variables (shown below) to visualize teams in 2D space as shown in Figure 1:

WIN%, PTS, FGM, FGA, FG%, 3PM, 3PA, 3P%, FTM, FTA, FT%, OREB, DREB, REB, AST, TOV, STL, BLK, BLKA, PF, PFD, +/-, OFFRT, DEFRTG, NETRTG, AST%, AST/TO, AST/RATIO, OREB%, DREB%, REB%,TOV%, EFG%,TS%, PACE, PIE, POSS

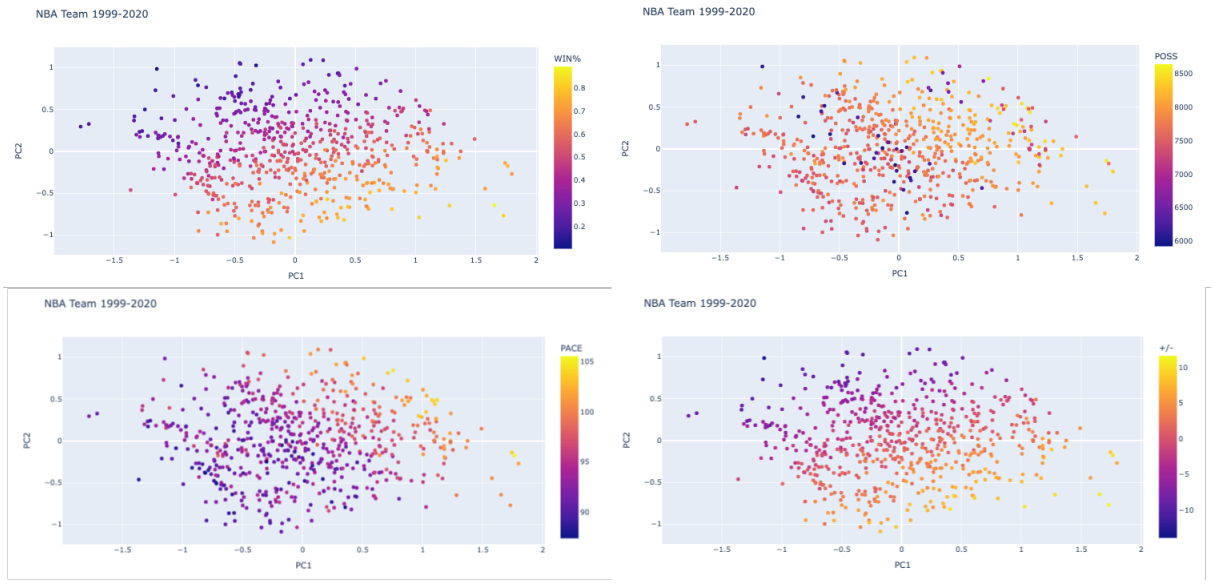

Figure 1: NBA Teams colored by team metrics shown in reduced to 2 dimensions

KMeans clustering was performed to create 8 distinct cluster segmentation, clustering was performed by normalizing all the team statistics with a min max scaler. When performing clustering, all metrics were given a weight of 1 with the exception of WIN% which was given a weight of 2.5 to add significance to winning. Clusters are shown in Figure 2.

These clusters represent distinct segments that NBA teams could be categorized into, the choice of K in this instance very arbitrary and can change depending on which team metric general manager may want to deem as significant. As seen in Figure 1, many team metrics can be used to assess teams, weight to any particular team metric may be added to cluster team segments. To tackle this problem of segment arbitrariness, we will also focus on ways to not only assign a hypothetical roster to a NBA team cluster, but also use a K nearest neighbor method to find NBA teams most

similar to our hypothetical NBA roster.

2.3 Convolutional Neural Network

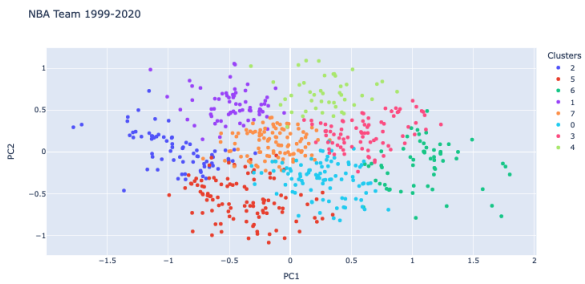

Figure 2: 8 clusters generated with KMeans

We have looked at historical NBA teams and how their statistics are distributed, as well as segments they fall into. A function that can approximate team statistics and team segments from an input of players and their associated individual player statistics (as shown below) is required.

POS, MIN, PTS, FGM, FGA, FG%, 3PM, 3PA, 3P%, FTM, FTA, FT%, REB, AST, STL, BLK, TO, DD2, TD, PER, AGE, OFFRTG, DEFRTG, NETRTG, AST%, AST/TO, AST RATIO, OREB%, DREB%, REB%, TO RATIO, EFG%, TS%, USG%, PACE, PIE, POSS

Since minutes played during games are mostly dominate among 8 players in the course of the season, we will limit our model to work with a roster of 8 players. In this instance, for an input data of 8 players, our model input is of shape 8x44, where players statistic POS is a categorical variable representing player position that is encoded using a dummy variable for the 8 different positions: C, F, G, GF, PF, PG, SF, SG.

2.3.1 CNN Training

Figure 3: 2D Input Data

Input data is preprocessed by normalizing all player data and encoding position with a dummy variable. A roster is taken by getting top 8 players (sorted by minutes played in descending order) with their normalized statistics for a given team. Input data for a specific roster can be represented as an image shown in Figure 3

38 different convolutional neural networks are trained for associated team statistics and clusters mentioned in 2.2. For models that predict numerical value, a mean squared error optimizer is used, and for models that predict categorical variables categorical cross entropy optimizer is used.

2.3.2 CNN Model Evaluation

All team metrics with the exception of FGA: field goals attempted, PFD: personal fouls drawn, and POSS: posessions have a mean square error rounded to 0 when predictions are made on testing data. Clusters are predicted accurately 50% of the time, and 73% when considering correct cluster neighborhood. With 8 clusters, a random choice would predict correctly with an accuracy of 12%, and a random choice in the correct neighborhood with 20%, thus the model's performance of 50% and 73% are adequate accuracy score.

2.4 Sampling

When conducting a hypothetical roster analysis, before we can feed player data into our convolutional neural network model, we need to make assumptions on players' individual statistics.

2.4.1 Minutes

Our model takes input of 8 players sorted by minutes played in descending order, to conduct a hypothetical roster analysis, an assumption of the minutes played by each player on a roster of 8 is required. Figure shows a distribution plot of minutes played by each player as well as a histogram plot, ordered by most minutes played to 8th most minutes played.

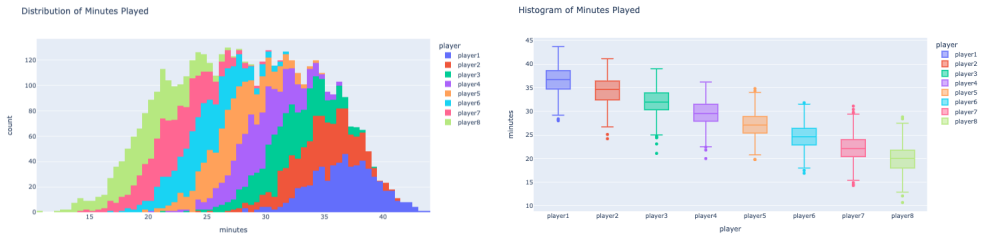

Figure 4: Distribution Hisrogram of minutes played

Minutes played by 8 players are not solely related linearly, their joint distrbution is multi-dimensional. To sample from the minutes, estimation of the joint distribution of minutes played by 8 players was found using kernel density estimation[2]. The kernel density estimator was estimated using all the minutes distributed for 8 players for all the rosters data.

2.4.2 Minute Adjusted Player Statistics

When players are chosen for a hypothetical roster, and their respective minutes per game is chosen, the next step is to sample their individual player statistics. There are many ways to do so, and more detail will be given in 2.5. The respective player's statistics must be adjusted according to the minutes played per game.

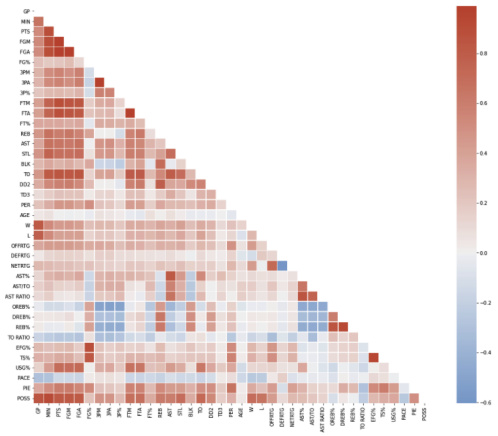

Figure 5: Correlation Matrix of Player Stats

Figure 5 shows a correlation matrix. The player statistics that have a correlation greater than 0.5 with minutes are adjusted to the minutes played. These include PTS: points, FGM: field goals made, FGA: field goals attempted, 3PM: 3 points made, 3PA: 3 pointa attempted, FTM: three throws made, FTA: free throws attempted, REB: rebounds, AST: assists, STL: steals, TO: turnovers, DD2: double doubles, PIE: player impact estimate, POSS: possessions.

2.5 Python Module

The steps presented are packaged into a Python module for users to conduct hypothetical roster analysis, shown in 3. The package allows users to download NBA team

data and player data for specified years, view clusters from 2, view team statistics for a hypothetical roster, view segments that a hypothetical roster belongs to, and view k nearest neighbor teams which are historical teams that the hypothetical roster will most resembles statistically.

2.5.1 Roster Analysis Options

Users have the option to select the minutes distribution of a team through.

i Sampling Minutes with Kernel Density Estimator

ii Average Minutes

For selecting player statistics, users may specify to select player stats based on their.

i Average stats for all seasons played

ii Stats for best reason played (chosen by highest PIE: player impact estimate)

iii Stats for prime playing years, with prime window default = 5

3 Experiments & Results

```
from NBA_Roster_Analysis import  NBA_Roster_Analysis
nba_analysis=NBA_Roster_Analysis()

team=['LeBron James','Kevin Durant',
      'Stephen Curry',"Shaquille O'Neal",
      'Carmelo Anthony','Kevin Garnett',
      'Paul Pierce','Ray Allen']

nba_analysis.predict_team_stats(team,minutes_selection_method='sample',
    stats_selection_method='prime',prime_window=3)

nba_analysis.k_nearest_neighbors(team,minutes_selection_method='sample',
    stats_selection_method='prime',prime_window=3,visualize=True)
```

Listing 1: Python module example

An experiment of a hypothetical lineup is done above with historically great NBA players, the parameters chosen are: minutes selection method done with kernel density estimator sampling, statistics of individual players are selected from their prime of 3 years. This hypothetical lineup's sample team stats are shown in Table 1 . The lineup is entered into 38 convolutional neural networks discussed in 2.3 and each team statistic is predicted. The team is also predicted to belong to cluster 3. When we look at figure 6 we can see the relative position this team belongs to and the close teams associated with it, we can also analyze the reactive distribution of their team metric in respect to all other teams if we color by that team metric.

WIN%	0.713406
PTS	116.494095
FGM	39.995895
FGA	89.250526
FG%	48.350498
3PM	11.375375
3PA	26.459307
3P%	38.104023
FTM	22.486904
FTA	31.103598
FT%	78.920700
OREB	11.863559
DREB	35.331051
REB	46.853794
AST	26.237303
TOV	15.523455

Table 1: Sample of estimated team statistics.

Similar historical nba teams are also found when comparing against the normalized team data and finding the euclidean distance between each one of them. These K nearest teams are outputting by the Python module.

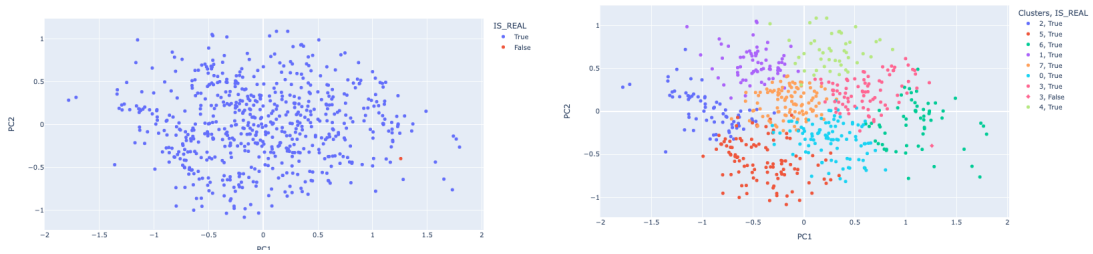

Figure 6: Location of hypothetical NBA roster in 2D space, and predicted cluster it belongs to.

4 Discussion

The algorithm performs adequately on predicting roster performance and estimating team statistics, if a general manager is looking to sign a free agent or trade for specific players but is unsure how each player he is thinking of acquiring may impact his team, this algorithm may be useful in finding the slight differences that may occur in trading for one player over another. A key team statistic may be shown to be weighted more significantly for an specific trade and not another, thus tilting a general managers decision making.

The algorithm to find hypothetical roster's predicted team statistics provides a useful behind the scenes analysis into player's statistics. Convolutional neural networks do an outstanding work of feature selecting as they optimize prediction, essentially pruning out player stats that are deemed as useless in predicting team statistics and segments. Some downfalls may exist however, neural networks have a very 'black-box' approach to estimation, which is unhelpful given that many general managers may need reasons and interpretability in their decision making, which a convolutional neural network may not be able to best explain. As useful as this proposed algorithm is there are still many limitations.

4.1 Limitations & Improvements

This model, although unbiased in a quantitative sense, is highly numbers driven. Sports are a domain where numbers are not the end all be all, it is only part of the larger equation. My current model cannot account for player chemistry; players with high statistical output may not perform as well when put into a lineup with other great performing players. The algorithm simulates player stats based off past season performance but not actual game to game, possession to possession data. A great area of improvement for this algorithm that I foresee is simulating actual game data to generate player statistics. Instead of sampling player individual stats from past seasons, a probabilistic graphical model can simulate a 48 minute game against defenders for a given hypothetical roster. A play by play data set would be required of each player, and sampling players decision making through a course of a game for every game in the season through a Monte Carlo Simulation on a Probabilistic Graphical Model would be a powerful tool. Once those individual player stats in the context of a team are estimated, then we can use a convolutional network to estimate team statistics as presented in this paper.

5 Conclusion

As the saying goes "all models are wrong but some models are useful". The convolutional neural network based approach to predicting team statistics given a roster of players whose individual stats are sampled and adjusted to minutes, serves as just that at the end, a model. However, the statistical insight provided to general managers is obvious, rather than an 'eye-test' a statistical based approach allows for more fine grain analysis and removes human bias and false judgements.

References

[1] Redefining nba player classifications using clustering. URL https://towardsdatascience.com/redefining-nba-player-classifications-using-clustering-36a348fa54a8.

[2] Density estimation. URL https://scikit-learn.org/stable/modules/density.html.

[3] Predicting the outcome of nba games with machine learning. URL https://towardsdatascience.com/predicting-the-outcome-of-nba-games-with-machine-learning-a810bgi=3c07a0e47917.