A Machine Learning Study of NBA Data

Guanhua He
Molecular Biology
ghe@princeton.edu

Kengran Yang
Civil and Environmental Engineering
kengrany@princeton.edu

Pingping Zhao
Department of Geosciences
pingping@princeton.edu

Abstract

As one of the most successful commercial sports league, NBA is famous for its advertisement strategy, live TV contract negotiation and data. Tons of data has been collected and analyzed by NBA official to present the fans with all the perspectives to see a game, a player or a team. However, machine learning, an emerging field of data processing and visualization, has not been introduced much in the context of NBA. We made some explorations regarding NBA standing predictions, NBA player salary predictions as well as clustering the team/player using dimension reduction tools in this report. We found that l1-regularized linear regression stands upon all other models for best prediction, and it can predict the standings of the good and bad teams fairly well. In the meantime, we clustered players and teams using principal component analysis and found that features like position can categorize players based on principal components (PCs) while age cannot. For team clustering results, we found that 2-point field goals is the leading feature that explains the most variance among all teams in terms of PC1 and PC2.

1 Introduction

As the biggest and most successful basketball league in this world, NBA has 30 teams and hundreds of players and holds over 1000 games every season, producing a huge amount of data which are relatively easily accessed online. By applying various machine learning algorithms on the NBA data, we can better appreciate the beauty of NBA games and obtain different insight into it. Moreover, the prediction of NBA outcomes is a huge part in the gambling industry, indicating that an accurate prediction model would be potentially profitable.

Because of the popularity of the NBA games, there are quite a few studies on the NBA data [7, 9], even though the focuses of them vary. In a quite sophisticated study, Cheng et al. employ the maximum entropy principle to predict an incoming game using the stats from past 6 games, and their result seems to be better than those from other popular machine learning techniques such as logistic regression and random forest [8]. Other studies include prediction of score margins [6] and win-loss of a game [5]. However, none of the studies have perform a analysis on the prediction of the NBA standings. An in-depth analysis of the NBA players is also lacking in the literature. Hence, we would like to address these problems by identifying the following sub-tasks: 1. prediction of the standings of both the east and west conference, 2. prediction of the salaries of the players, and 3. an analysis on the player- and team-based data using dimension reduction and clustering techniques. Here the standing means the ranking of the teams based on their win rates. For the first two tasks, various regression models will be tried, while dimension reduction and clustering techniques will be employed to tackle task 3. The goal of this project is to explore the NBA data from a new angle, and examine the applicability of the machine learning technique on NBA data.

2 Related Work

2.1 Data processing

2.1.1 Data for the prediction of NBA standings

The raw data for this sub-task is crawled from the website basketballreference.com, using a crawler program found on the Internet [1]. The format of the raw data is per-game statistics, including performance of each player, as well as the whole team for both home team and away team. The performance is quantified by 46 measures, the detail of which can be found on the basketballreference website [3]. Each team will play 82 games in each regular season, and 4 - 28 more games depending on whether it enters the play-off season or not. So, there are roughly 1250 games played in each season. Games from 2003 to 2016 (13 seasons) were crawled (∼ 13,000 games), and grouped into different seasons. For each season, the games were further grouped by different teams and the averages of the stats for these games are used as features for each team. In other words, the average stats of one particular season constitute one data point for each team, so each team would have 11 (2003 - 2014) or 12 (2003 - 2015) data points as training data. The corresponding label for each training data point is the win rate, defined as the ratio between the number of winning games and the total number of games played, in that particular season.

Now we have a set of training data for each team, so in total we have 30 sets of data. In order to predict the standings which is the relative performance of the teams, the win rate of each team needs to be calculated. Our approach here is to build a machine learning model for each team, using the past performance for each team to predict its performance in the future (win rate in 2016 in this case). Once the predicted win rate for each team is obtained, the predicted standings (or rankings of win rate) can be easily calculated.

2.1.2 Data for the prediction of player salaries

The salary data is from starcrunch.com including all player salary data in 2015 as features. The player performance data of 2013-2015 is from a github project [4]. The biggest issue in data processing is that the players in the feature matrix and label matrix are not identical. Therefore, we used fuzzywuzzy [2], a github fuzzy string matching toolkit to calculate the "distance" between two players in both matrix and reserve those with 90 or higher matching values. We compared the result making sure there is not false positive match. For those who didn't match, we just eliminate the players.

2.1.3 Data for the analysis of NBA player and team stats

Team-based data for season 2015-2016 were picked up from the crawled data in 2.1.1. Of all 46 features for 30 teams, the string feature *Name* was dropped off and the rest 45 features were then used for principal component analysis.

2.2 Methods

The models we used for regression tasks are regularized linear regression, random forest and Gaussian Process Regression. Linear regression assumes multivariate linear relationship between features and labels, while Gaussian process accepts non-linear relationships, which could potentially be the case for the NBA time series data. Random forest is used because it is found to have decent performance in the Fragile Family Challenge study. Regularization is applied to the linear regression model as a common technique to avoid over-fitting.

PCA converts a set of observations of possibly correlated variables into a set of values of new linearly uncorrelated variables (called Principal Components) in such a way that those principal components capture largest variances in the data. PCA supplies us with a lower-dimensional picture, a projection of original data when viewed from the most informative viewpoint. Apart from producing derived variables for use in supervised learning problems, PCA also serves as a useful tool for data visualization. So to reduce the dimensionality of our NBA player- and team-based data and to interpret the data in a more meaningful form, Principal Component Analysis was conducted in this project.

2.3 Evaluation

For the prediction of NBA standings, two evaluation metrics are used: mean squared error and Spearman's rank-order correlation coefficient. The mean squared error is used here to offer a general sense of the accuracy of the regression models. The Spearman's rank-order correlation coefficient, on the other hand, is peculiar to this sub-task because it assesses the correlation between two rankings. It is defined as follows:

$$r_s = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \tag{1}$$

where d_i is the difference between the two ranks of each observation.

r_s ranges from 1 to -1, with 1 indicating two identical rankings and -1 indicating two opposite rankings.

To interpret PCA results of team-based dataset, each principal component's eigenvector was calculated and plotted in order to identify those features that are most contributing to the principal components. Eigenvectors are the component's linear combinations. They give the weights for the features by which we can know which feature have high/low impact on each principal component.

3 Spotlight Methods: Principal Component Analysis

Intuitively, PCA aims to represent the original features in the data with a linear combination of a (much) smaller amount of new features, or "principal components". Mathematically, it aims to decompose the original $p \times n$ matrix in the following way:

$$X = WZ \tag{2}$$

where W is an $p \times k$ matrix consisting of k principal component vectors, and Z is a $k \times n$ matrix containing the weights of each principal component vector. In addition, PCA aims to minimize the following objective function:

$$J(W, Z) = ||X - WZ^T||_F^2 \tag{3}$$

under the constraint that W is orthonormal [10]. Note that $||A||_F$ is the Frobenius form.

The solution of the above optimization problem is constructed on a vector-by-vector basis. Denoting the j'th principal component in W as w_j, the i'th column in X as x_i, and the i'th low-dimension representation in Z as z_i, the vector form of equation 3 is:

$$J(w_j, z_i) = ||x_i - w_j z_i||^2 \tag{4}$$

Starting from the first principal component, we have

$$J(w_1, z_1) = \frac{1}{N} \sum_{i=1}^{N} ||\mathbf{x_1} - z_{i1}\mathbf{w_1}||^2 \tag{5}$$

Taking the derivative of equation 5 and equating to 0 gives

$$z_{i1} = \mathbf{w_1^T x_i} \tag{6}$$

Plugging it back to equation 5 we get

$$J(w_1) = \frac{1}{N} \sum_{i=1}^{N} [x_i^T x_i - z_{i1}^2] = const - \frac{1}{N} \sum_{i=1}^{N} z_{i1}^2 = const - w_1^T C w_1 \tag{7}$$

where $C = X^T X$ which is the covariance matrix. Note that this is true only when X is centered.

Thus, to minimize equation (7) is equivalent to maximizing $w_1^T C w_1$, under the constraint $||w_1|| = 1$. This can be done by introducing a Lagrange multiplier and minimizing $w_1^T C w_1 - \lambda(w_1^T w_1 - 1)$. Differentiating, we obtain $Cw_1 = \lambda w_1$, which is the eigenvector equation. Since we want to maximize the variance, we pick the eigenvector that corresponds to the largest eigenvalue. The rest of w_j and z_i can be calculated by repeating the above procedure. These eigenvectors are the component's linear combinations. They give the weights for the features so that we can know which features have high/low impact on each principal component from the magnitude of eigenvalues.

4 Results

4.1 Prediction of NBA standings

After an evaluation of different regression models, it is decided to use Elastic net with built-in cross validation as the model for this sub-task because it offers the smallest mean square error. The l1 ratio of the model is determined by first finding the best one using a 3-fold cross validation for each team, repeating this process for all teams, and then picking the one used by the most teams. It is found that the distribution of the l1 ratio is highly polarized towards the two extreme values, 0 and 1. Eventually 1 is used because there are slightly more teams using this ratio.

4.1.1 Feature Selection

There are a number of ways to define the features and the test data, and they impact the performance of the prediction substantially, as shown in table 2. Two types of statistics are used as features: "self" in the "Stats" column means the statistics of the team itself will be used, while "difference between self and opponent" indicates that the difference of the statistics between the team itself and its opponent is used. "Number of games used for testing data" denotes the number of games used to calculate the average of the statistics for the testing data of year 2016. The reason why we tweak this variable is that in reality the prediction result will be the most meaningful when the regular NBA season is ongoing, where the statistics of all 82 games are not available. The standing would have been obvious if all the games had finished. Lastly, "Number of training data" is simply the number of data points used to train the model.

It is seen from table 2 that using the difference between self and the opponent generally gives slightly better performance compared to using just the statistics from the team itself, possibly due to the fact that using the difference takes extra information into account.

Two values are examined for the number of training data: 11 and 12. The reason we pick these two numbers is that to use 12 training data points would mean that the immediate next data point (year 2016) in the time series will be predicted, while using 11 points as training data would mean that the point to be predicted (year 2016) is a bit further away from the endpoint (year 2014) of the training data. Initially we have even less amount of training data (7 data points), and found that in that case, whether we are predicting the immediate neighbour in the time series or not makes a big difference in the prediction outcomes (not shown here). The larger the distance between training and testing data, the better the prediction outcome. This might be due to the fact that the impact of outliers becomes significant given a small amount of training data, and the model have been overfitted, such that it is not correctly reflecting the true situation when trying to extrapolate just at the outside of the training data. The performance of the model does improve when increasing the number of samples to 11, and the difference between using 11 or 12 data points becomes much reduced, because now the contribution from outliers to the model becomes less.

Lastly, the testing data is tweaked by using two different number of games to calculate the average stats in order to examine if solely using half of the games is predictive enough. Unfortunately this is not the case, where both the Spearman coefficient and MSE drop substantially compared to using all games as input. This is not unreasonable because some teams may not perform consistently in the whole regular season. For example, they might be good at the first half of the season, but getting worse due to change of personnel.

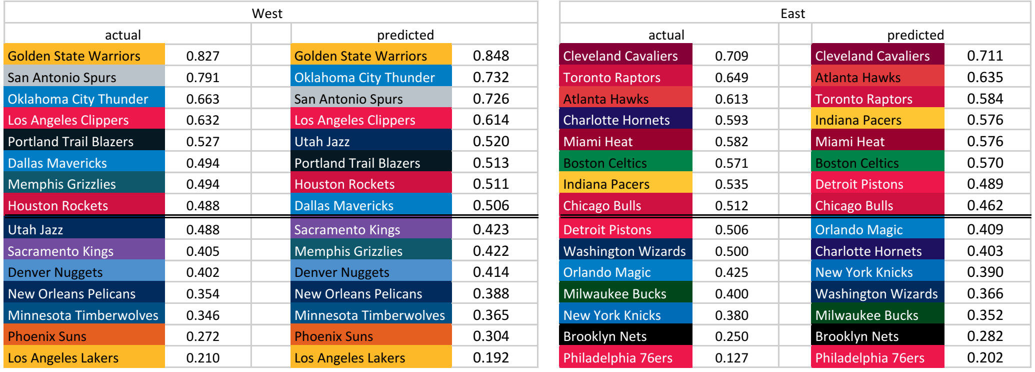
4.1.2 Comparison between Predicted and Actual Standings

The feature selection analysis shows that the scenario providing the best performance is the combination of using statistics from the difference between the team and its opponent, 82 games for testing data and 12 training data points. This combination is then used to predict the standings of the NBA teams in 2016, the result of which is provided in figure 4.1.2. It is seen that the predicted outcome roughly replicate the actual ones, with the prediction of west conference being slightly better than the one of east conference. For the west conference, both the best (first four) and worst (last five) teams are predicted quite accurately, the only mistake being the swap of position between San Antonio Spurs and Okalahoma City Thunder. However, the prediction of the teams in the mid-range becomes less accurate. It can be seen that Utah Jazz is predicted to be in the 5th position, but it

Table 1: Performance of the model using different features and different testing data.

Scenarios			Metrics	
Stats	Number of games used for testing data	Number of training data	Spearman Coefficient	MSE
difference between self and opponent	82	11	0.928571	0.003926
difference between self and opponent	82	12	**0.939286**	**0.003099**
self	82	11	0.910714	0.004089
self	82	12	0.935714	0.004099
difference between self and opponent	40	11	0.885714	0.005888
difference between self and opponent	40	12	0.885714	0.005574

actually ranks the 9th. Memphis Grizzlies, on the other hand, is underrated by our model by 3 positions. Other teams, such as Portland Trail Blazers and Houston Rockets, fluctuate around its actual position but not too much. We think that the two outliers (i.e., Utah Jazz and Memphis Grizzlies) occur because of the style of these two teams. They both are teams with traditional style, meaning that their stats might not be pretty, but they are still able to win games. Also they focus a lot on defense, which can not be easily reflected by the statistics. That being said, one way to improve the performance of the model would be to include features that quantify defense efforts, and features of players such that the presence of strong defensive players might impact the fitting of the model.



West				
actual			predicted	
Golden State Warriors	0.827		Golden State Warriors	0.848
San Antonio Spurs	0.791		Oklahoma City Thunder	0.732
Oklahoma City Thunder	0.663		San Antonio Spurs	0.726
Los Angeles Clippers	0.632		Los Angeles Clippers	0.614
Portland Trail Blazers	0.527		Utah Jazz	0.520
Dallas Mavericks	0.494		Portland Trail Blazers	0.513
Memphis Grizzlies	0.494		Houston Rockets	0.511
Houston Rockets	0.488		Dallas Mavericks	0.506
Utah Jazz	0.488		Sacramento Kings	0.423
Sacramento Kings	0.405		Memphis Grizzlies	0.422
Denver Nuggets	0.402		Denver Nuggets	0.414
New Orleans Pelicans	0.354		New Orleans Pelicans	0.388
Minnesota Timberwolves	0.346		Minnesota Timberwolves	0.365
Phoenix Suns	0.272		Phoenix Suns	0.304
Los Angeles Lakers	0.210		Los Angeles Lakers	0.192

East				
actual			predicted	
Cleveland Cavaliers	0.709		Cleveland Cavaliers	0.711
Toronto Raptors	0.649		Atlanta Hawks	0.635
Atlanta Hawks	0.613		Toronto Raptors	0.584
Charlotte Hornets	0.593		Indiana Pacers	0.576
Miami Heat	0.582		Miami Heat	0.576
Boston Celtics	0.571		Boston Celtics	0.570
Indiana Pacers	0.535		Detroit Pistons	0.489
Chicago Bulls	0.512		Chicago Bulls	0.462
Detroit Pistons	0.506		Orlando Magic	0.409
Washington Wizards	0.500		Charlotte Hornets	0.403
Orlando Magic	0.425		New York Knicks	0.390
Milwaukee Bucks	0.400		Washington Wizards	0.366
New York Knicks	0.380		Milwaukee Bucks	0.352
Brooklyn Nets	0.250		Brooklyn Nets	0.282
Philadelphia 76ers	0.127		Philadelphia 76ers	0.202

Figure 1: Comparison of predicted and actual NBA standings in 2016. The numbers represent win rates.

The situation of the east conference is similar, there being slightly more jiggering of the teams, potentially due to the fact that the win rates of the teams in the east are closer to each other, and it is harder for the model to differentiate between them. An outlier, the Charlotte Hornets, also exists, possibly due to similar reasons mentioned above.

4.2 PCA and k-mean clustering on teams

For the benefit of visualization, we chose the first two components for principal component analysis which explain over 85% variance in the data (see Figure 3, left). Of the two chosen principal components, PC1 is most interesting as it explains over 75% of the variance in data (see Figure 3, left). Based on eigenvectors (see Figure 3, right), the 2nd feature (2-point field goals) is the highest weighted for PC1 and has a positive eigenvalue. That means that *Chicago Bulls, Boston Celtics, Brooklyn Nets* and *Atlantic Hawks*–those teams with high eigenvalues on PC1 (see Figure 2, left)–should have higher 2-point field goals than other teams. Therefore, we conclude from PCA and k-mean clustering that among all 45 features, the feature *2-point field goals* is the best one to distinguish one team from another.

Basing on PCA, k-mean cluster was further applied to classify all teams into 5 classes. As we can see, k-mean is clustering teams mostly according to PC1 values (Figure 2, right) which is contributed mainly by 2 point field goals.

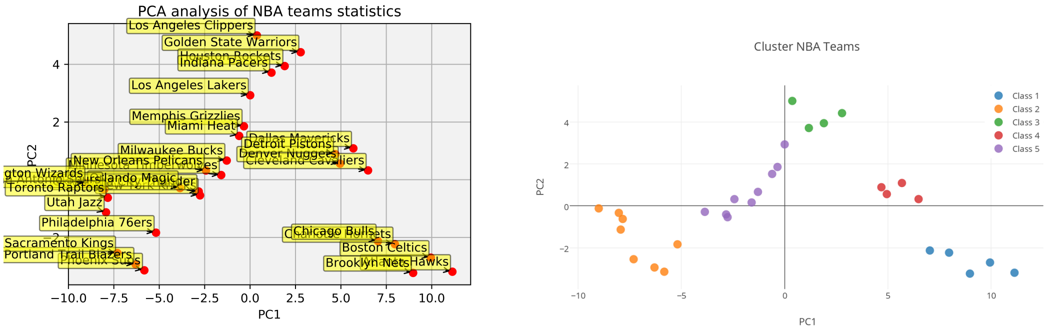


Figure 2: Left: PCA result of team-based data. Teams' names are marked in the yellow boxes. Right: k-mean clustering based on PCA

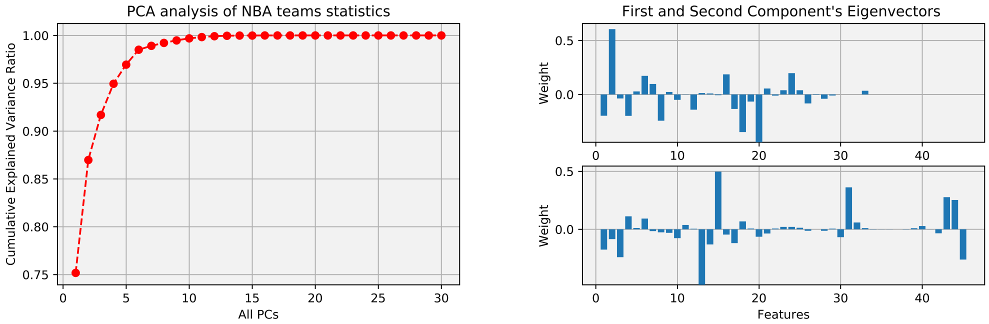


Figure 3: Left: Cumulative variance ratio explained by the number of principal components. The first two PCs together explain over 85% variance in the data. PC1 itself explains over 75%. Right: Based on eigenvectors which reflect the influence of each feature on the PCs, the 2nd feature (2-point field goals) has the highest impact on PC1. That means that as *Chicago Bulls, Boston Celtics, Brooklyn Nets* and *Atlantic Hawks* have rather high values on the first PC (PC1) (see Figure above, left), these teams will have higher 2-point field goals than other teams.

4.3 Player salary prediction and clustering

Generally speaking, the salary of each NBA play depends on their performance. And currently there are a few metrics that try to quantify the overall performance of a player, including BPM, win share, and RAPM. The detailed computation of those metrics vary, but their core is similar: they compare the team performance with a given player and with a league-average player. If those metrics of one player is above zero, it means in every perspective this player's performance is above average, and vice versa.These metrics provide a quantitative way for NBA fans and experts to evaluate players, and they more or less affect the salary level. For example, Lebron James's BPM in 2008-09 season set a historic record: 13, which is even higher than Michael Jordan's 12.6 in 1989-90 season. Interestingly, both the legendary players didn't win the championship at that season. The players' performance should have a strong correlation with their salaries. However, we couldn't find a public model depicting the relationship between player performance and salary. Therefore, we collected all

NBA player salary in 2015 and their stats in 2013,2014 and 2015 as our raw dataset. We tried to use the player performance including points, minutes rebounds. etc as training features and player salary as labels. Here, we use different regression models and random forest to learn the training dataset and preditc. The prediction metric is Mean Squared Error(MSE). Here we can see that overall the performance of 2013 and 2014 do a better job in predicting salary in 2015. In contrast, 2015 player performance is less informative in predicting 2015 salary. This makes sense because the salary of a given year usually depends on the performance of past few years, instead of this year's.

4.3.1 Model selection

We also did model comparison here to select for the best prediction model. Linear regression is the most intuitive method since there might be linear correlation between player performance and salary. Because the dataset size is not big(around 400 samples), to avoid overfitting we employed L1 and L2 regularization term to reduce parameters. Also, it is possible that the relationship between feature and label is non-linear. Hence we applied Gaussian Process to depict the potential non-linear correlation. Besides, random forest tends to be a powerful tool in some prediction tasks. The result is as Table 2 shows: all linear regression models do a good job in prediction, especially lasso regression. One feature for lasso regression is that it significantly reduces parameter amount by 'arbitrary' feature selection. It turns out to be useful, since some of the features are indeed correlated like BPM and VORP.

Table 2: Performance of each prediction model

	linear regression	ridge	lasso	ElasticNet	GaussianProcess	random forest
2015	15.13883895	14.95976923	15.02415016	15.54846915	63.76736221	17.16184252
2014	15.36175068	14.82479303	14.69617764	16.11827034	73.54592385	19.37174522
2013	17.66958787	16.42312151	16.18829808	18.20396288	81.96848474	17.63762177

4.3.2 Prediction visualization and outliers

Then, we plot the player salary against VORP for observed and predicted samples in Fig 4. We could see that our prediction basically reflected the trend that better performance, higher salary. However, there are a few outliers that are hard to predict. One group of them is overpaid group, with Kobe Bryant, Derrick Rose and Joe Johnson. Another underpaid group is made of Stephen Curry, Damian Lillard and so on. These observations is quite consistent with the reality.

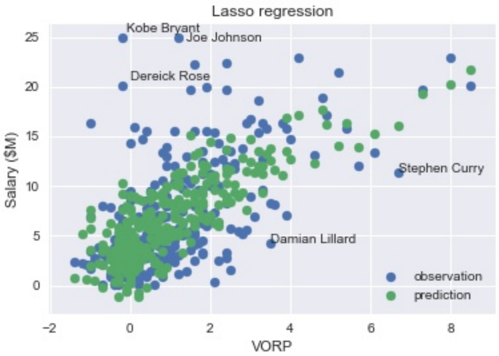


Figure 4: prediction of 2015 salary based on 2014 performance using lasso regression

4.3.3 Cluster players based on position and age

Then we wonder, can we apply PCA on these training dataset and see if it could cluster the players in some way? In Fig 5 , we plot PC1 vs PC2 and put all the samples on it. Besides, we labeled each sample(player) with their positions on the left panel or ages at right panel. We can see that PC2 and PC1 together separated center and shooting guard, whom should have distinct stats in terms of rebounds, stealing and etc. On the right panel, we separate each player's age into 3 groups: below 25, 25 to 29, and above 30. We can't really distinguish them based on the color, which suggests that the performance, or stats more specifically, are indistinguishable across different ages.

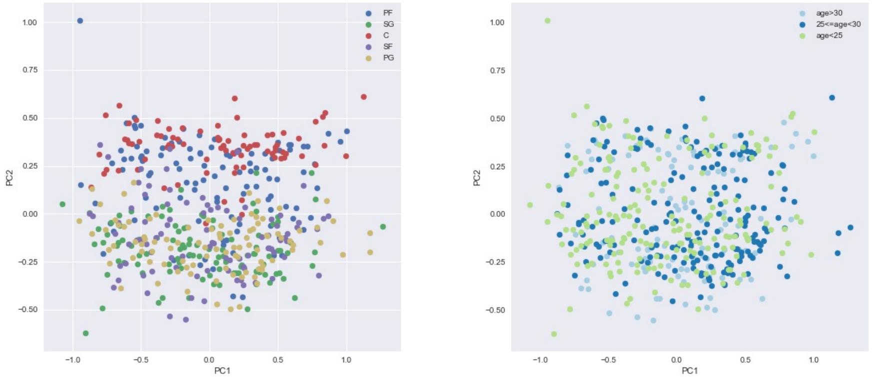


Figure 5: Cluster all players based on their position and age after PCA

5 Discussion and Conclusion

Three different tasks have been performed in this project: prediction of NBA standings, prediction of the salaries of the NBA players, and the PCA and clustering of NBA team- and player-based datasets. For the team standing prediction, it is found that the elastic net algorithm, coupled with careful feature selection, is able to predict the NBA standings with satisfactory accuracy. However, its predicting power is impacted if the number of games used to calculate the input data is insufficient. Also, the prediction of teams in the mid-range of the ranking is less accurate. One potential way to improve the model in the future is to include features that quantify the defense effort of the teams. The predicting power may be improved by assigning weights to the games such that more recent games have a bigger impact on the future performance of the team.

For the prediction of each player's salary, 6 different algorithms have been tried and lasso regression was found to be more accurate than others. Lasso gives good salary prediction for most players, but for some outliers (those well-known players who are overpaid or underpaid) even lasso is much less satisfactory. For the PCA and clustering of team-based dataset, the feature *2-point field goals* is found to be most important to the first principal component which explains over 75% variance and is concluded to be the best feature to distinguish one team from another. Players' position in basketball game was found to be a good feature to distinguish different players in the PCA analysis of player-based dataset.

In the future, there are a few ways to further explore this research. The first is to expand the sample size by using web crawler to grab more data and improve our model. Model-wise, when we use Gaussian Process to predict, the metrics are poor. We think that selecting the right kernel for Gaussian Process regressor might be critical.

References

[1] Basketball reference scraper. `https://github.com/FranGoitia/basketball_reference`. Accessed: 2017-05-15.

[2] Fuzzy string matching in python. `https://github.com/seatgeek/fuzzywuzzy`. Accessed: 2017-05-15.

[3] Glossary. `http://www.basketball-reference.com/about/glossary.html`. Accessed: 2017-05-15.

[4] Predict an nba player's per score. https://github.com/initFabian/NBA-Machine-Learning-Tutorial. Accessed: 2017-05-15.

[5] Prediction of nba games based on machine learning methods. https://homepages.cae.wisc.edu/~ece539/fall13/project/AmorimTorres_rpt.pdf. Accessed: 2017-05-15.

[6] Various machine learning approaches to predicting nba score margins. http://cs229.stanford.edu/proj2016/report/Avalon_balci_guzman_various_ml_approaches_NBA_Scores_report.pdf. Accessed: 2017-05-15.

[7] Richard Borghesi. An examination of prediction market efficiency: Nba contracts on tradesports. 2009.

[8] Ge Cheng, Zhenyu Zhang, Moses Ntanda Kyebambe, and Nasser Kimbugwe. Predicting the outcome of nba playoffs based on the maximum entropy principle. *Entropy*, 18(12), 2016.

[9] Edward R Hirt, Frank R Kardes, and Keith D Markman. Activating a mental simulation mindset through generation of alternatives: Implications for debiasing in related and unrelated domains. *Journal of Experimental Social Psychology*, 40(3):374–383, 2004.

[10] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.