**Predicting NBA fantasy scores using Linear Regression**

**Ioannis Sakkaris, Alexandros Lotsos**

**ABSTRACT**

This project aims to use linear regression in various ways in order to predict an NBA player’s performance in a given match and by extension his Fanduel fantasy score. The predictions will be made using the player’s performance and the opposing team’s general pattern of play over the past five games as input for the regression. We will attempt to fit linear models to various subsets of players ranging from individual players to subsets organized by position (Point guards, Shooting guards etc.). We will then use our predictions to draft an optimal lineup for a specific format of fantasy basketball. We expect to see the accuracy of our results vary in accordance with the size of the player subset each regression applies to.

**INTRODUCTION**

Fantasy sports and sports statistics in general are fields that embrace data mining and statistical analysis because of the ease in quantifying a player performance as well as the mass appeal to the general public. Fantasy sports specifically, have been growing in popularity in the past year with just about every major sport and league having representation. Currently, there is a number of platforms on which fantasy sports can be played, including DraftKings and Fanduel.

The main objective of this project is to devise a way to predict a player’s performance on a given match day and to use that prediction in order to pick an optimal lineup for Fanduel’s head to head (H2H) format. In the H2H format, two users compete against each other using all the players from a given match day. For example, if there are 3 matches and each team has 12 players, then the two users must draft 9-man lineups from a pool of 72 players. After the lineups have been drafted, they are judged based on their real-life performance where each player receives a Fanduel fantasy score which is calculated as follows:

After scoring, the team with the higher cumulative score is the winner of the session.

The predictions were based on a dataset of each player’s performance during the 2015-2016 regular season in each of their games. The predictions themselves were done using linear regression where the input variables were initially defined to be each statistic describing a player’s individual performance over his 5 most recent games, as well as statistics describing the opposing team’s performance in their 5 most recent games.

**TECHNIQUES**

The method of choice for predicting the Fanduel score of each player was linear regression because the predicted output is a continuous real value. When picking our predictors for each case we used a backward elimination which involved taking into account every statistic of each player and team and iteratively eliminating them based on their p-values.

We also incorporated some clustering in our project in two forms. The first form of clustering was a manual clustering based on player position which was done in an effort to improve our regression results. The second form of clustering was done using K-means and took equally into consideration a player’s individual performance and the time he spent playing in his last five games. That is, we separated the players into groups that were distinguished strictly by their position in the first case, and their performance over time in the second case.

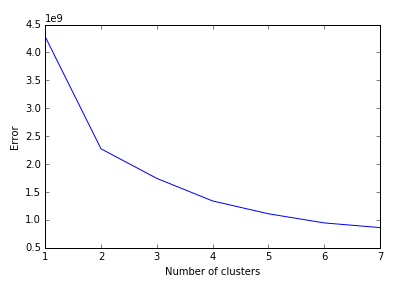
**DATASETS AND EXPERIMENTS**

Our dataset was scraped from Yahoo Sports using the python module BeautifulSoup. In detail, our dataset is organized in a dictionary whose keys correspond to players and values correspond to player statistics. Using the data of each player we were able to also piece together team statistics. In detail, the statistics for each player were minutes played, field goals, free throws and three pointers made and attempted as well as their rebounds, steals and assists. The statistics for each team were the same statistics allowed by the team (i.e. field goals allowed). Using backward elimination during the fitting of our linear regression we picked the optimal set of predictors depending on the situation.

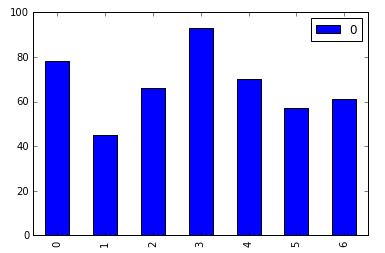
In practice we tried three different approaches so we could get our predictions, the most basic of which is a linear regression that is fitted for each individual player. That is, the optimal predictors and weights were unique to each player and each regression was trained on a maximum of 76 games that each player played during the regular season.

In an effort to improve our results, we decided to also arbitrarily cluster the players based on their position. This means that in this approach there were 9 different regressions, one for each position or combination of positions including point guard (PG), guard (G), shooting guard (SG), guard forward (GF), small forward (SF), forward (F), power forward (PF), forward-center (FC) and center (C). Each regression was trained on all the games that players of each position played, resulting in all players of one position having common weights and predictors.

Finally, we attempted to cluster the players using K-means based on the time that they play on average as well as their personal performance. This was done in order to separate players more organically so that we can account for differences of play styles (e.g. offensive point guards vs defensive point guards) rather than relying on an arbitrary position separation. By doing the below error analysis we quickly realized that 7 clusters is a good number for our model, as the error function gets more flat at that point. Also, when we tried more clusters, some of them had a really low number of players.

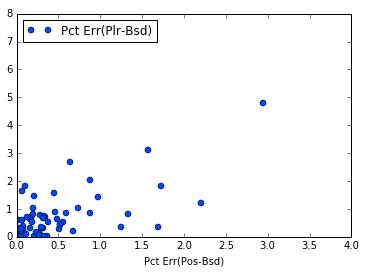


Below, you can also see the amount of players in each clusters.



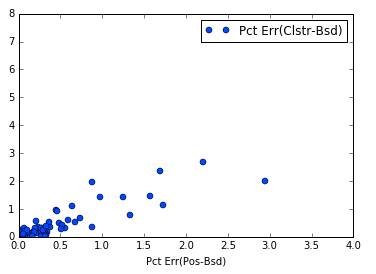
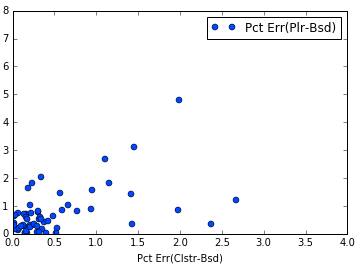
**RESULTS**

Below is a graph of the percent errors of both approaches as tested on the matches played during the 4/27/2016 game day. On the X-axis is the error of our Position based approach and on the Y-axis is the error of our Player based approach. Thus, points that are close to either axis but far from the other represent a case where one approach performed better than the other.



We can see that the player based approach generally had higher percentages of error compared to the position based approach which was expected since a mdoel that is trained on just 76 games (on average, it much less than that) will definitely underperform.

Below are two graphs similar to the previous one, comparing the k-means clustering approach with the position and player based approaches. The approaches were tested on the games of the same day as before and the coordinates of the points still represent the error of each approach.



We do notice that the k-means clustering outperforms the player based approach significantly while still pulling ahead from the position based approach which seems to indicate that clustering in some way shape or form to account for individual play styles would yield better predictions in the long run.

It should also be noted that outliers with large errors in all three approaches are players who played a very small amount of time each game or played few games in general. Those players tend to have low scores anyway and a player drafting a fantasy lineup who is aiming to win would rarely draft those players.

This can be seen in the following table where Marreese Speights has an error of more than 100% in each of the three methods.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Actl FanDuel Scr | Prdctd Scr(Pos Bsd) | Prdctd Scr(Plr-Bsd) | Prdctd Scr(Clstr-Bsd) | Pct Err(Pos-Bsd) | Pct Err(Plr-Bsd) | Pct Err(Clstr-Bsd) |
| J.J. Redick | 25.6 | 16.80152 | 7.117872 | 21.60445 | 0.343691 | 0.721958 | 0.156076 |
| Jeff Green | 32.2 | 15.40731 | 18.64653 | 19.9972 | 0.521512 | 0.420915 | 0.378969 |
| Al Jefferson | 25.9 | 26.20245 | 32.97145 | 22.26119 | 0.011677 | 0.273029 | 0.140495 |
| Mason Plumlee | 36 | 26.07178 | 7.958173 | 25.27604 | 0.275784 | 0.77894 | 0.297888 |
| Pablo Prigioni | 6.4 | 11.99206 | 19.51707 | 8.583147 | 0.87376 | 2.049542 | 0.341117 |
| Marreese Speights | 6.4 | 17.46827 | 17.94233 | 13.76717 | 1.729416 | 1.803489 | 1.15112 |
| Dwyane Wade | 32 | 30.26226 | 28.19626 | 33.97854 | 0.054304 | 0.118867 | 0.061829 |
| Josh McRoberts | 8.1 | 10.61055 | 5.520065 | 9.093986 | 0.309945 | 0.318511 | 0.122714 |
| Marvin Williams | 33.6 | 14.84164 | 15.51038 | 22.85564 | 0.558284 | 0.538381 | 0.319773 |

The full table of all our predictions can be found in the file “errors.csv” attached to this report.

**CONCLUSIONS**

During our testing process it became apparent that creating a model for each individual player and his matchups was unfeasible given the amount of games the average player plays during a single season. Perhaps collecting data on previous seasons for a given player would make player based approaches more feasible since, in theory, a player based approach would be more suitable in predicting a player’s performance given the wide variety of styles in each position (i.e. more defensive point guards vs. more offensive point guards). However, the helpfulness of collecting data from previous seasons is debatable given that a player can switch teams from season to season which might affect his role in a given lineup. For example, the star player of a lower end team might end up being a role player in a high end team.

It thus is slowly becoming apparent that the best way to predict a player’s performance is to identify his playstyle and to try and create a model that can accurately predict the performance of similar players against different kinds of opponents. In our case it seemed that rudimentary K-means clustering yielded linear models that outperformed both previous approaches even without taking into account the opposition that the player was facing in a given match. Perhaps more advanced methods of clustering or a more detailed choice of what to cluster by would yield even better results.

In future studies we would therefore like to explore different clustering methods and their effects on the regressions. It is feasible that more advanced clustering could allow for data from previous seasons to be used for better predictions.