Prediction of NBA FanDuel Points

Jordan Levy, Yingyin Xiao

1. Introduction

When looking for datasets to study, we looked at subjects we were both interested in. We both like basketball, so we decided to look at NBA data from the past three years and make an insightful prediction on that data.
Over the course of any given NBA season, there are a variety of awe-inspiring standout performances from players across the league. As aspiring data scientists, we wanted to see if we could predict a player's performance before it happens.
Before beginning to come up with a model to solve this problem, we thought about the problem from a human perspective: what conditions lead to a successful or unsuccessful performance. We noticed that many players tend to go on hot and cold streaks that can last any number of games. In addition, teams usually play better at home in what is known as 'home-court advantage.' If the team is more likely to win at home, then the player's performance will likely be better at home. These factors and more were all things to consider when first tackling this problem.

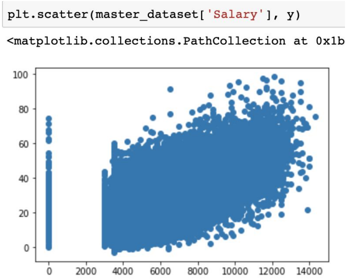
Our first course of action was to start collecting data. We scraped data from two different websites: stats.nba.com and rotoguru1.com. Specifically, we looked at the following sites: stats.nba.com/players/boxscores/, stats.nba.com/teams/boxscores-advanced/, stats.nba.com/players/boxscores/, and rotoguru1.com/cgi-bin/hyday.pl?game=fd&mon=10&day=25&year=2016. For the last link, we noticed that the URL contains the month, day, and year. This allowed us to scrape all data for the last three years by iterating through the start and end dates for each season. We then were able to scrape data on stats.nba.com by recording network activity under developer tools and downloading the .json file that generated the table. In total, we had scraped all player boxscores, team boxscores, fantasy points scored for each game in the last three seasons; as well as player bios for each of the players. Player bios entailed draft position, height, weight, and what college they went to among other information.
Once we had scraped our data, we loaded it into pandas data frames and joined the different datasets by 'PLAYER_ID' and 'GAME_ID' that was provided in all tables from stats.nba.com. However, when joining the data from rotoguru1.com, we had to match the names and dates for each game. This involved using regular expressions to match names that otherwise were recorded differently

from the two sites. Once we were done collecting and joining our data, we performed several analyses to get familiar with our dataset.

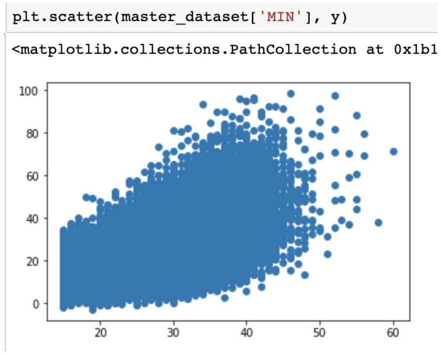
b. Exploratory Data Analysis (EDA)

The original dataset has 65 columns and 59214 rows; we narrowed it down to having 26 columns that interest us to study. The columns included the players' information and their performance on each game. Since the dataset is closely connected to FanDuel Points(FD PTS), other than exploring the interesting phenomenon in the dataset, we explored the relationship between FanDuel Points and other attributes.

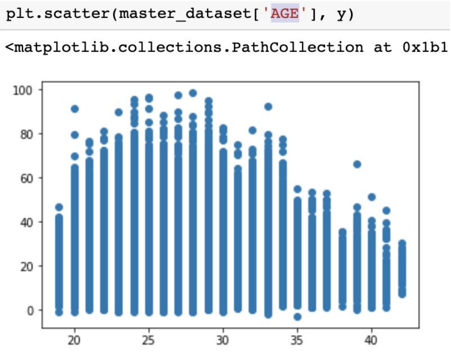
After drawing multiple graphs to imply insights, we found that some attributes are linearly related to FD PTS, as could be seen in the following scatter plots. We saved the 17 related attributes for future usage.



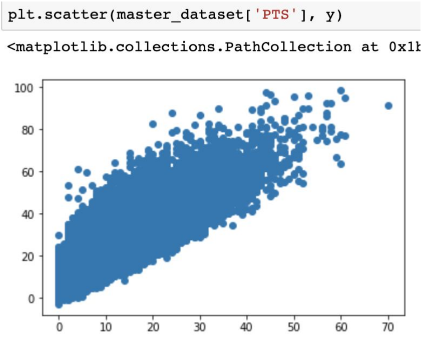
```
plt.scatter(master_dataset['Salary'], y)
<matplotlib.collections.PathCollection at 0x1b
```



```
plt.scatter(master_dataset['MIN'], y)
<matplotlib.collections.PathCollection at 0x1b1
```

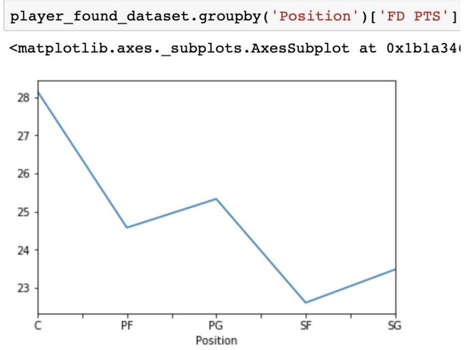


```
plt.scatter(master_dataset['AGE'], y)
<matplotlib.collections.PathCollection at 0x1b1
```



```
plt.scatter(master_dataset['PTS'], y)
<matplotlib.collections.PathCollection at 0x1b
```

We also created a new table to exclusively examine the columns with types int and float. From the table, we grouped by player's position to see if positions and the FD points are connected, so we found that positions indeed would affect the FanDuel points.



```
player_found_dataset.groupby('Position')['FD PTS']
<matplotlib.axes._subplots.AxesSubplot at 0x1b1a34
```

Then, we saved the attributes as a potential categorical feature.

2. Problem Statement

After the EDA, we found our interest in knowing how we would predict the FD Points as accurately as possible under the situation that we've found features that have a linear relationship with FD Point. We would like to perform predictive analysis on the statement that, given information on players and games for the past 3 years, we could predict their FD Points on the next game.

3. Objective

Identifying a predictive task

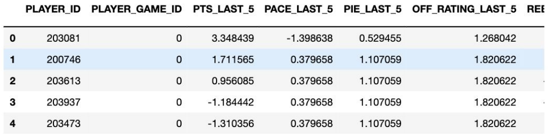
Going into this project, we already had an idea of what we wanted to predict. We were interested in predicting the performance of players game-to-game using fantasy points as the metric by which we measure their performance. We decided to use the formula for fantasy points provided by FanDuel, which is as follows:

$$FanDuelPoints = PointsScored + 1.2 * Rebounds + 1.5 * Assists + 3 * Steals + 3 * Blocks - Turnovers$$

We had a wide assortment of features to use in our model, as we had at least 50 columns of quantitative statistics and advanced statistics to choose from. In order to find the most predictive statistics, we found the r-squared scores for each column and our label, FanDuel points scored. This was done in part during EDA, when we generated a series of scatter plots showing linear correlation (or lack thereof) between different statistics and FanDuel points scored.

Feature Engineering

We did lots of feature engineering when evaluating our predictive task. In particular, we were careful to look at what features would be available for a game that has not happened yet. Keeping this in mind, we needed to implement features based on previous data. We decided to generate a rolling average of the last 5 games for the statistics most linearly correlated with FanDuel points scored. This involved using the pandas.rolling function, as well as generating game ids for each player's set of games.



	PLAYER_ID	PLAYER_GAME_ID	PTS_LAST_5	PACE_LAST_5	PIE_LAST_5	OFF_RATING_LAST_5	REI
0	203081	0	3.348439	-1.398638	0.529455	1.268042	
1	200746	0	1.711565	0.379658	1.107059	1.820622	
2	203613	0	0.956085	0.379658	1.107059	1.820622	
3	203937	0	-1.184442	0.379658	1.107059	1.820622	
4	203473	0	-1.310356	0.379658	1.107059	1.820622	

Figure An example of a list of player IDs with their corresponding game id. Since this is the head of the dataset sorted by date, all game ids are 0 (their first game).

Once we have a set of chronologically ordered game ids for each player, where 0 is the first game played by the player in the dataset, we then were able to implement rolling averages.

We generated rolling averages for the last 5 to 10 games for 'PTS', 'FD PTS', as well as rolling averages for the past 5 games for 'REB', 'AST', 'OFF RATING', and whether or not the team played at home.

Once we had our list of quantitative features, we looked at encoding several categorical/nominal features. We one-hot

Using this set of features, we were ready to move on to designing our model.

4. Preliminary Literature Review

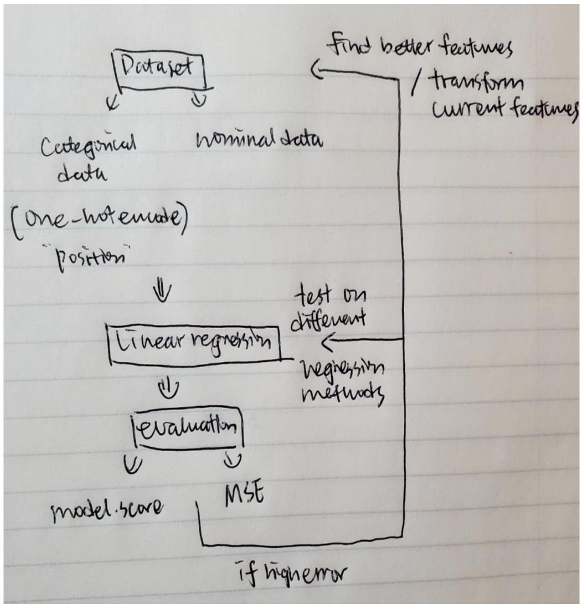
When creating our model, we looked at other existing solutions for inspiration. While we didn’t find anyone else with the exact same dataset as we had, we found other articles describing their approach in predicting player performance using historical statistics. We looked at one blog post, titled Using Automated Machine Learning to Predict NBA Player Performance. In this article, the author predicts Game Score, another metric used to measure player performance based on the player’s boxscore. They used a tool on their website to automate machine learning based on the previous season’s boxscores. In total, they had approximately 30-40 different modeling approaches, of which five were selected based on the cross-validation score. In addition to quantitative statistics, they scraped daily player reports and saw which words led to the biggest change in performance. The word “out” had the largest negative impact, as this usually means a player will not be playing that game, and the words “points” and “rebounds” seemed to have a large positive impact, as they are more likely to be used when noting impressive performances in these categories.

The biggest takeaway from this article for us was how they conducted feature engineering. They utilized lagging or rolling averages in several different statistical categories, such as average game score, points, and field goals over the last 30 games, as well as more abstract features such as the over-under of that game (how many points are predicted to be scored by both team) and the money line (which team is favored and by how much). This inspired us when designing our own features as we implemented moving averages of our own.

We found another project by a student at Stanford who used Linear Regression and an assortment of player-specific and team-specific features. The student took a similar approach as we did and further inspired us to use Linear Regression in our model.

Additionally, we found inspiration through the Week 10 Monday lecture, when Professor McAuley talked about making predictive models with time-series data.

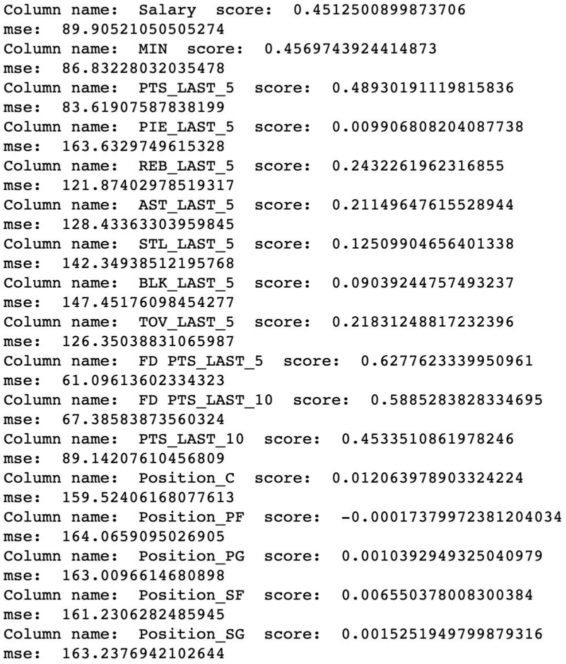
5. Methodology



a. After defining our problem that we are doing predictive analysis on the FanDuel points(target), we the related attributes first. Preliminary, we decided to use linear regression for the predictive task as the attributes showed a linear relationship with the target.

b. We split the dataset into a training set(0.65) and a test set(0.35), one-hot encoded the categorical feature “Position” and fit a linear regression model on the training set.

c. At first, the error was high; We then further narrowed down the attributes by printing out the R-squared values, MSE with the linear regression model and excluding the features with negative scores.



```
Column name:  Salary  score:  0.4512500899873706
mse:  89.90521050505274
Column name:  MIN  score:  0.4569743924414873
mse:  86.83228032035478
Column name:  PTS_LAST_5  score:  0.48930191119815836
mse:  83.61907587838199
Column name:  PIE_LAST_5  score:  0.009906808204087738
mse:  163.6329749615328
Column name:  REB_LAST_5  score:  0.2432261962316855
mse:  121.87402978519317
Column name:  AST_LAST_5  score:  0.21149647615528944
mse:  128.43363303959845
Column name:  STL_LAST_5  score:  0.12509904656401338
mse:  142.34938512195768
Column name:  BLK_LAST_5  score:  0.09039244757493237
mse:  147.45176098454277
Column name:  TOV_LAST_5  score:  0.21831248817232396
mse:  126.35038831065987
Column name:  FD PTS_LAST_5  score:  0.6277623339950961
mse:  61.09613602334323
Column name:  FD PTS_LAST_10  score:  0.5885283828334695
mse:  67.38583873560324
Column name:  PTS_LAST_10  score:  0.4533510861978246
mse:  89.14207610456809
Column name:  Position_C  score:  0.012063978903324224
mse:  159.52406168077613
Column name:  Position_PF  score:  -0.00017379972381204034
mse:  164.0659095026905
Column name:  Position_PG  score:  0.0010392949325040979
mse:  163.0096614680898
Column name:  Position_SF  score:  0.006550378008300384
mse:  161.2306282485945
Column name:  Position_SG  score:  0.0015251949799879316
mse:  163.2376942102644
```

d. To make sure that we have the most accurate model, we also tested on different regression methods: SVM, Logistic Regression, MLPRegression, none having better performance.

6. Results

Mean Squared Error: 47.960389248581116
R2-Score: 0.7038713865510818

Our MSE score of approximately 48 means that our model deviated from the actual FD points scored by an average of 6.928 points, or the square root of our MSE. This improved from our baseline model, which had an MSE of about 60. One hot encoding based on position, as well as adding rolling averages for the last 10 games as well as 5 helped our MSE decrease as we designed our model.

7. Improvements

While we were able to get pretty decent results out of a relatively simple model, we noted several things we could improve upon. First was that we could have spent more time designing features related to our dataset. For example, we could have implemented a feature that decayed the importance of stats over time instead of simply taking the average over a period of games. The most recent stats tend to be more important than those that happened long ago. Additionally, it would have been cool to implement natural language processing on player reports like the article mentioned above does, as well as collecting information about the opposing team, such as their defensive rating over the last X games and more.

As for designing our model, we did not have much experience in working with regression predictions. A more complex model such as a latent based model may have been more successful in predicting player performance.

All in all, predicting NBA player’s

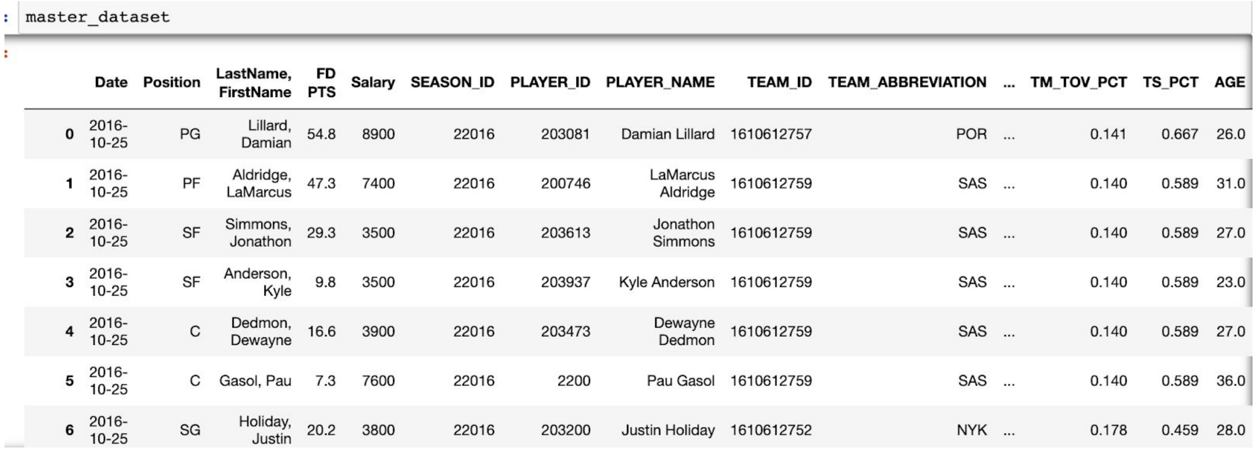
performances before they happen is a very challenging task due to the unpredictable nature of sports. but we can utilize powerful tools provided to us in data science packages to help predict the unpredictable. All models are wrong, but some can help predict NBA player performance.

8. References

Miller, Benjamin. “Using Automated Machine Learning to Predict NBA Player Performance.” *English*, 5 June 2018, blog.datarobot.com/using-datarobot-to-predict-nba-player-performance.

Wheeler, Kevin. *Predicting NBA Player Performance*. 2012 cs229.stanford.edu/proj2012/Wheeler-PredictingNBAPlayerPerformance.pdf.

PS.



```
master_dataset
```

	Date	Position	LastName, FirstName	FD PTS	Salary	SEASON_ID	PLAYER_ID	PLAYER_NAME	TEAM_ID	TEAM_ABBREVIATION	...	TM_TOV_PCT	TS_PCT	AGE
0	2016-10-25	PG	Lillard, Damian	54.8	8900	22016	203081	Damian Lillard	1610612757	POR	...	0.141	0.667	26.0
1	2016-10-25	PF	Aldridge, LaMarcus	47.3	7400	22016	200746	LaMarcus Aldridge	1610612759	SAS	...	0.140	0.589	31.0
2	2016-10-25	SF	Simmons, Jonathon	29.3	3500	22016	203613	Jonathon Simmons	1610612759	SAS	...	0.140	0.589	27.0
3	2016-10-25	SF	Anderson, Kyle	9.8	3500	22016	203937	Kyle Anderson	1610612759	SAS	...	0.140	0.589	23.0
4	2016-10-25	C	Dedmon, Dewayne	16.6	3900	22016	203473	Dewayne Dedmon	1610612759	SAS	...	0.140	0.589	27.0
5	2016-10-25	C	Gasol, Pau	7.3	7600	22016	2200	Pau Gasol	1610612759	SAS	...	0.140	0.589	36.0
6	2016-10-25	SG	Holiday, Justin	20.2	3800	22016	203200	Justin Holiday	1610612752	NYK	...	0.178	0.459	28.0