Predictive Modeling of NBA Player Value Using AdaBoost and Stochastic Gradient Descent for Applications to Fantasy Basketball Betting

Thomas Kalnik, MBA/MS Finance, Northeastern University

Abstract

This study attempts to predict the expected future value (FV) of individual NBA players based on historical performance during the typical NBA season. Specifically, this study looks at the performance of players throughout the 2015-2016 and 2016-2017 seasons. Various predictive variables are used to extrapolate expected future performance, evaluated as *number of points scored per game.*

A variety of classification and prediction models were tested on the data; AdaBoost and Stochastic Gradient Descent exhibited superior performance, Support Vector Regression performed adequately. Player point predictions are classified according to discrete point intervals, binned into identical length intervals with the lowest scoring players removed from further analysis. A sample of the AdaBoost, SGD Regression and SVM Regression results are reproduced in *Exhibit 3,* SGD Regression and AdaBoost were the algorithms that displayed greatest prediction accuracy on the data.

Introduction

The growing popularity of Fantasy Sports betting websites like *DraftKings.com* and *FanDuel.com* represent both a unique challenge for Machine Learning and Operations Research practitioners and a potentially lucrative inefficient market with risky arbitrage potential. The relatively small size of the DraftKings betting pool for any particular tournament and the amateurish nature of many market participants lends DraftKings several advantages as a speculative asset marketplace to the astute investor and data scientist that larger, more liquid, and more transparent markets such as the stock exchange lack.

The goal of this research is to evaluate what historical variables effectively predict the points a player will score in any given game. Several features are excluded from analysis (such as opponent) because of the complicated systems dynamics nature of the

necessary analysis, but could be extensible to future research. Multiple analytical tools were used for preprocessing, modeling the data and analyzing the results. To scrape the source data, webscraper.io was used. Tableau was used for some of the data visualizations presented in this paper. R was used to plot correlation matrices of the data. The Orange3 data-mining toolkit and a python script were used for preprocessing and modeling the data and analyzing the results.

Background

Daily fantasy sports games represent a trend in fantasy sports games that have been gaining in popularity since the founding of firms FanDuel, founded in New York in 2009, and DraftKings, founded in Boston in 2012[i]. The two firms currently dominate daily fantasy sports betting in the US.

In daily fantasy sports tournaments, players compete for cash prizes by creating a lineup or team of professional athletes while adhering to a salary cap, gaining points based on the real-world statistical performance of players in daily competitions, NBA in-season games are an example of one such contest. Each player is assigned a salary value which is updated regularly by the daily fantasy sports firm. For the purposes of this research, DraftKings salary data are utilized.

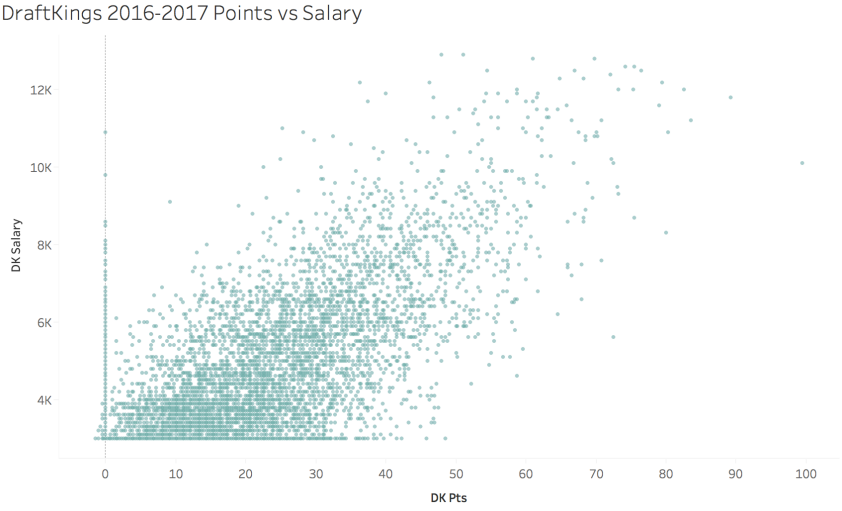
Basketball is an ideal sport for predictive modeling because it is a large and active market with many market participants to wager against and the frequency of games (and number of points scored in a game) is much higher than many other sports, 82 games are played in a basketball season compared to 16 games in a football season. The lower frequency of games in football suggests a relative paucity of data relative to basketball, thus it may be comparably easier to train a DraftKings prediction model for basketball than football. Because there are more games there are also more chances to bet in a season, which allows a betting strategy based on high probability (>50%) trades more time to recover from a string of sustained losses.

It is widely accepted that a minority of daily fantasy sports players win most contest prizes. *Bloomberg Businessweek* writes that "elaborate statistical modeling and automated tools that can manage hundreds of entries at once and identify the weakest opponents" allow skilled professional players to dominate the competition[ii]. In an effort to level the playing field, DraftKings has gone so far as to ban off-site programming scripts and create designated contests for "elite-level" players.[iii] This research is not concerned with monetary gain but instead the application of Machine Learning principles to an interesting prediction problem, we will disregard legal and ethical considerations for the remainder of this paper.

Problem Statement

A logical initial consideration for analysis of the problem is the correlation of *points* scored by players to the assigned DraftKings salary. If the DraftKings market were efficient, it stands to reason that a linear correlation would exist between the actual number of points scored in a game and the assigned player salary, higher scoring players are assumed to have a higher salary cost. *Figure 1* plots player points vs DraftKings salary.

Figure 1



Visual analysis of **Figure 1** suggests there is no clear linear correlation between player point performance and DraftKings salary. The problem can thus be assumed to be a nonlinear multiclass classification problem, the classes represented are the point range outcomes predicted by the historical player performance data.

Initial Hypothesis

H0: Historical NBA statistical data cannot be used to predict future expected points scored per player

Linear Hypothesis

$$h_\theta(x) \neq \theta_0 + \theta_1 x$$

H1: Historical NBA statistical data is predictive of future expected points scored per player

Linear Hypothesis

$$h_\theta(x) = \theta_0 + \theta_1 x$$

Data Selection, Collection and Preprocessing

The 2016-2017 NBA player data was collected from Basketball-reference.com[iv] using appropriate large instance data mining techniques. The processed data set had 18,100 instances and 31 columns. The 2015-2016 data is slightly larger, with 26,053 instances. In addition to the player NAME and the DATE, the following 29 features are represented in the data:

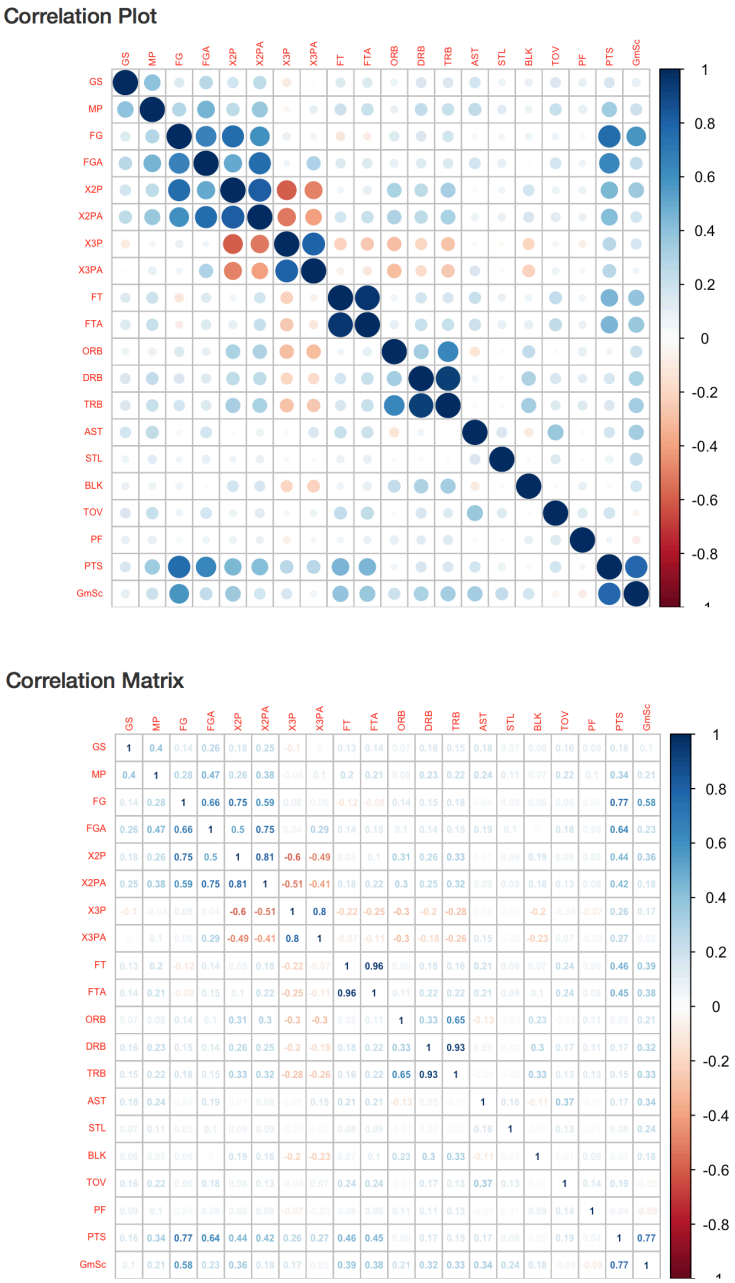
Rk -- Rank
Age -- Age of Player at the start of February 1st of that season.
Pos -- Position
Tm -- Team
Opp -- Opponent
GS -- Games Started
MP -- Minutes Played
FG -- Field Goals
FGA -- Field Goal Attempts
FG% -- Field Goal Percentage
2P -- 2-Point Field Goals
2PA -- 2-point Field Goal Attempts
2P% -- 2-Point Field Goal Percentage
3P -- 3-Point Field Goals
3PA -- 3-Point Field Goal Attempts
3P% -- 3-Point Field Goal Percentage
FT -- Free Throws
FTA -- Free Throw Attempts
FT% -- Free Throw Percentage
ORB -- Offensive Rebounds
DRB -- Defensive Rebounds
TRB -- Total Rebounds
AST -- Assists
STL -- Steals
BLK -- Blocks
TOV -- Turnovers
PF -- Personal Fouls
PTS -- Points
GmSc -- Game Score

A sample of the data is represented in *Exhibit 2*. After collection, the data was binned according to the number of points the player scored per individual game (not the average number of games played in the season, the individual game value for that date). Different binning intervals were tested, an interval of 7 was found to be preferable to smaller or larger bucket sizes.

Feature selection was tested using a correlation matrix and Principal Component Analysis (PCA). Important features were retained for analysis, unimportant features or features deemed difficult to model were disregarded. Various tests determined conclusively that some of the most predictive features are: MP, GmSc, FG, FGA, FT and FTA. Future research could incorporate more difficult to model 'systems-level' variables, such as intra-team player

correlations, team matchups and individual opponent matchups. For the purposes of this research, players were analyzed in isolation with no consideration of the effect of team or opponent matchups.

Figure 2, Feature Correlations, 2016-2017 Data



Individual position feature correlations are presented in *Exhibit 5*

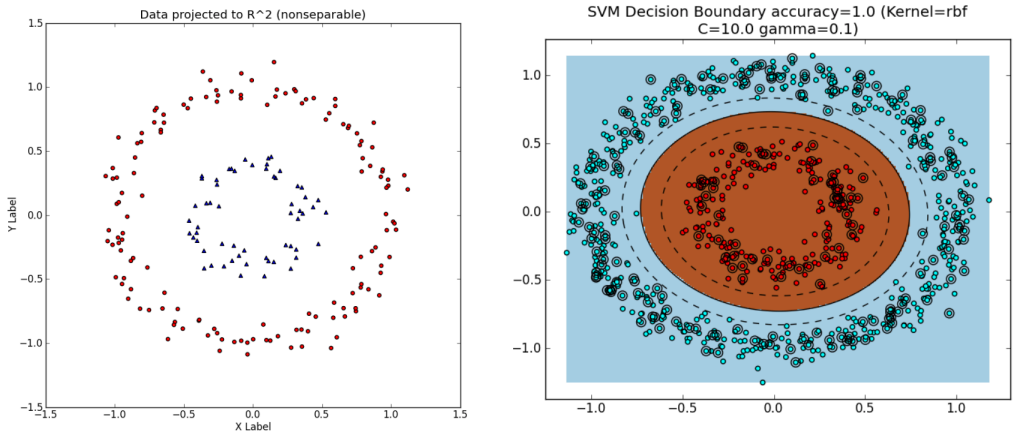
Brief Overview of Statistical Methods of Analysis

Support Vector Regression Algorithm

Support Vector Machines classify a training dataset according to a decision boundary known as the "margin" or "maximal margin classifier". This is essentially an attempt by the algorithm to divide the data into the most closely related groups. Support Vector Machines allow extensibility of linear algorithms to nonlinear decision boundaries by using a mathematical technique known as a "kernel trick", which essentially maps the data into a higher n-dimensional space.

Support Vector Machine classification is generally focused on binary classification problems, however a variety of techniques exist to utilize Support Vector Machines on multiclass classification problems, such as one-versus-one and one-versus-all classification[v].

Because the dataset presented is not clearly linearly separable, in this research a Support Vector Machine trained on the nonlinear multiclass classification problem presented in the *problem statement* section is used. A brief visual representation of the application of SVM classification to a nonlinear classification problem is presented below.



Both images retrieved from: http://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html#%5B5%5D

AdaBoost Learning Algorithm

AdaBoost is a classification algorithm that uses an ensemble of weak learners (classifiers) to systematically update and improve the model in an iterative process. Weak learners are predictors with prediction accuracy just better than random chance (just above 50%). Through many iterations and the addition of new predictors that improve the model by decreasing the overall model prediction error, an ensemble AdaBoost model is created that can classify the data with a high degree of accuracy, as represented by low MSE and RMSE. AdaBoost is a desirable algorithm to use for this research problem because of its "out-of-the-box" nature, requiring minimal preprocessing of the data, and its resistance to overfitting.
The general equation for the algorithm is represented below:

$$\begin{aligned}\sum_i w_i e^{-y_i h_i \alpha_t} &\leq \sum_i \left(\frac{1 - y_i h_i}{2}\right) w_i e^{\alpha_t} + \sum_i \left(\frac{1 + y_i h_i}{2}\right) w_i e^{-\alpha_t} \\ &= \left(\frac{1 + \epsilon_t}{2}\right) e^{\alpha_t} + \left(\frac{1 - \epsilon_t}{2}\right) e^{-\alpha_t}\end{aligned}$$

For this research, AdaBoost is utilized to address a multiclass classification problem. AdaBoost is trained on 75% of the data and tested on 25% of the data using a shuffle split of 10 random samples.
Python's SKLearn Machine Learning package comes with two almost identical AdaBoost algorithms designed for use on multiclass classification problems, described here: "SAMME.R uses the probability estimates to update the additive model, while SAMME uses the classifications only…the SAMME.R algorithm typically converges faster than SAMME, achieving a lower test error with fewer boosting iterations."vi
SAMME is an acronym for *Stagewise Additive Modeling using a Multi-class Exponential loss function.*

One advantage of using the AdaBoost algorithm for this research is its robustness to overfitting[vii] with lower complexity decision stumps, data that exhibits minimal outliers and lower dimensional data. The data collected for this research does not contain gross outliers (see *Exhibit 4* on outlier testing). 29 potentially predictive features could cause the model to suffer from *the curse of dimensionality,* however accurate prediction results were obtained from the feature space, suggesting that the dataset is of a manageable dimensional size, particularly when the number of features used as predictors was isolated to the most correlated variables (namely MP, GmSc, FG, FGA, FT and FTA).

Stochastic Gradient Descent Algorithm

Stochastic Gradient Descent is an iterative learning algorithm that attempts to minimize the cost of an objective function, defining a global minimum, generally improving the prediction error with each iteration. Generalized, Stochastic Gradient Descent can be interpreted as:

Repeat until convergence

{

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

}

https://www.r-bloggers.com/regression-via-gradient-descent-in-r/

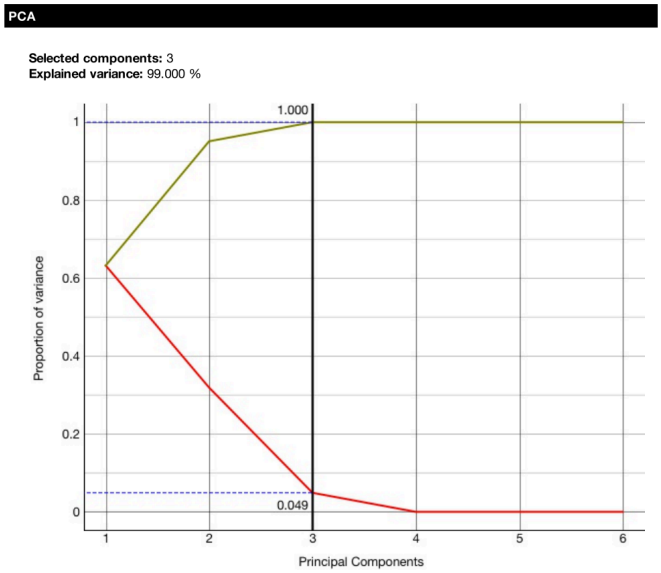
where:

Theta is the estimated parameter that minimizes the objective function
Alpha is the learning rate

Results

2015-2016 NBA Player Data Analysis

Principal Component Analysis



Principal Component Analysis was conducted on the retained 119 highest scoring instances.

GmSC Test Results

Sampling type: Shuffle split, 10 random samples with 75% data

Scores

Method	MSE	RMSE	MAE	R2
AdaBoost	0.815	0.903	0.616	0.983
SGD Regression	0.012	0.110	0.064	1.000
SVM Regression	1.016	1.008	0.301	0.979

Points Test Results

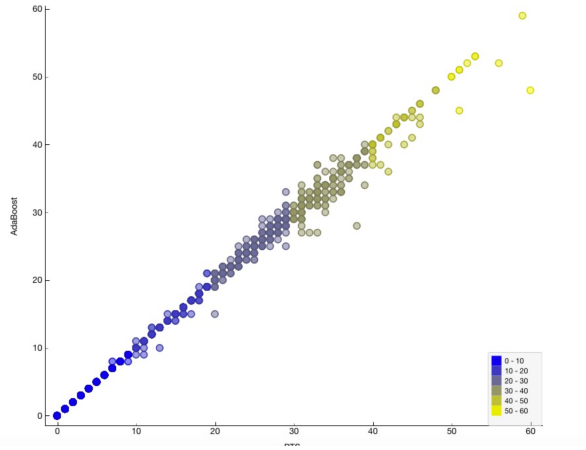
Sampling type: Shuffle split, 10 random samples with 75% data

Scores

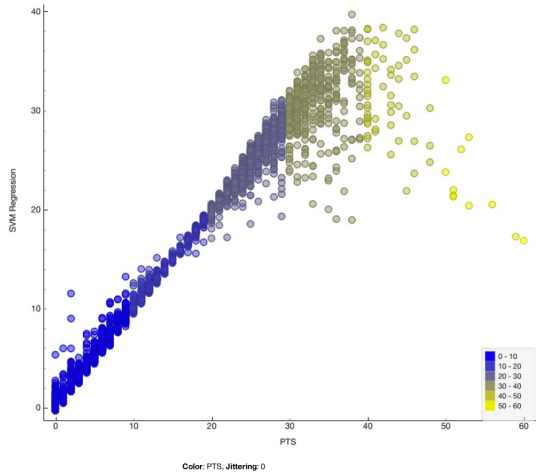
Method	MSE	RMSE	MAE	R2
AdaBoost	0.107	0.327	0.051	0.998
SGD Regression	0.001	0.032	0.019	1.000
SVM Regression	1.326	1.151	0.284	0.978

Visualization of model fit using scatterplots

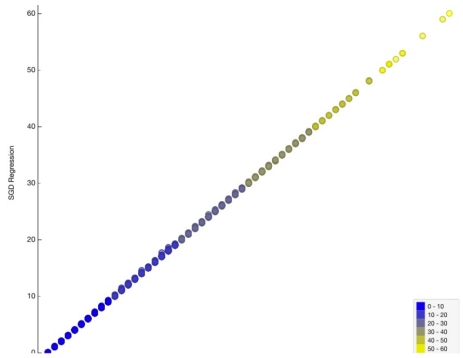
AdaBoost Regression Scatterplot



Support Vector Regression Scatterplot



Stochastic Gradient Descent Scatterplot



2016-2017 NBA Player Data Analysis

Ranks

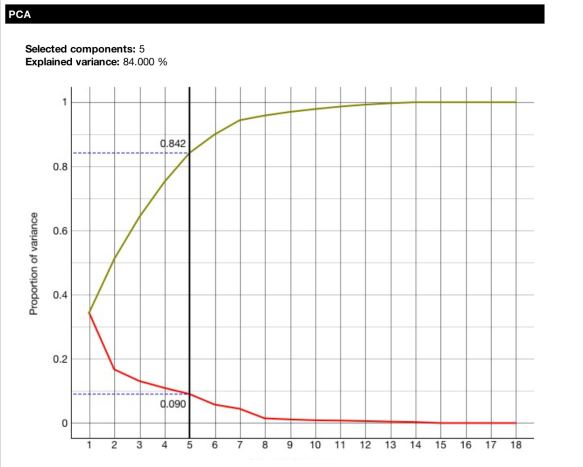
	#	Univar. Lin. Reg.	RReliefF
GmSc	C	4226.197925218353	0.0841048501413946
FG	C	4173.882454426905	0.07655800346215563
FGA	C	1978.5003412284736	0.05286237089871204
FT	C	768.1020431254251	0.06560539158898664
FTA	C	735.2445366763026	0.06850410598947143
2P	C	710.9803461788368	0.06631828218172792
2PA	C	624.5976438211173	0.05517022874146578
MP	C	384.02858892179404	0.05933736545989416
3PA	C	232.0674617633983	0.06988198861165966
3P	C	216.1452396004446	0.06149007776116981
TOV	C	107.2325821847236	0.08798297675046421
AST	C	88.42930403640155	0.07856387703503455
DRB	C	82.46813510912126	0.06572493686097662
TRB	C	70.0530196465861	0.05845767822789566
STL	C	18.36977025827523	0.0783468674737131
ORB	C	7.477660699964339	0.05257885351047492
PF	C	5.7292479246668595	0.09123609044036182
BLK	C	1.8898458104651492	0.0652006200254566

Scoring by Inter-cluster distance

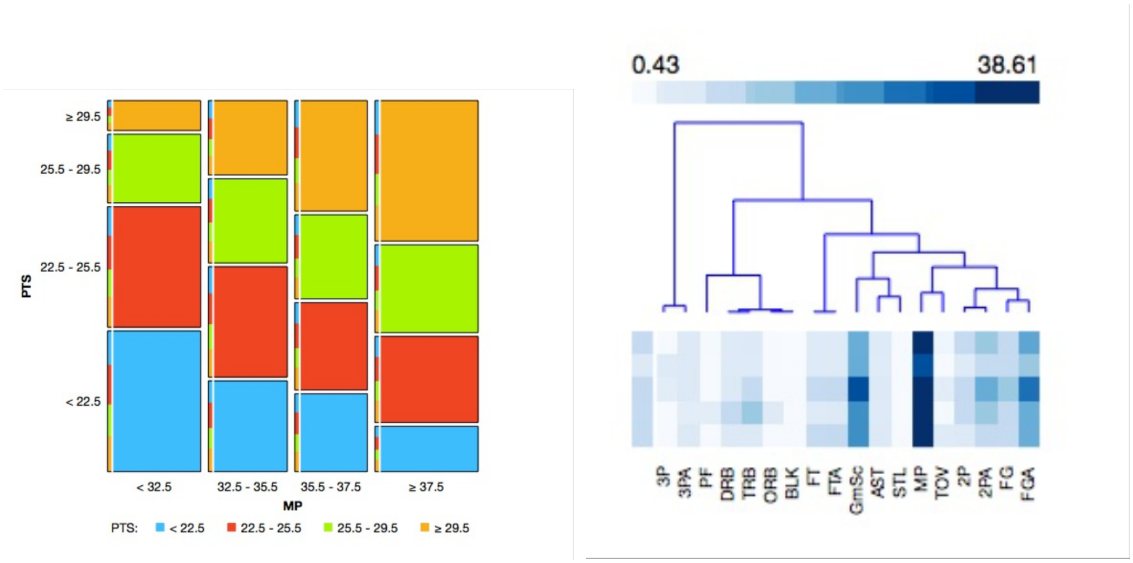
k	Score
2	13.7411
3	13.7080
4	14.5152
5	14.7426
6	14.5849
7	15.4799
8	15.6492

Univariate Linear Regression, RReliefF and Inter-cluster distance scoring. Univariate Linear Regression and RReliefF ranking suggest that GmSc, FG, FGA, FT and FTA account for the majority of variability in the data.

PCA 2016-2017 data, not normalized



Results from Hierarchical Clustering and Mosaic Display on the *Minutes Played* variable in Orange, 2016-2017 Player Data



Points Test Results, 2016-2017 Data

Test & Score

Settings

Sampling type: Stratified Shuffle split, 10 random samples with 75% data

Scores

Method	MSE	RMSE	MAE	R2
AdaBoost	1.072	1.035	0.503	0.967
SGD Regression	0.011	0.107	0.077	1.000
SVM Regression	3.914	1.978	0.663	0.879
Random Forest Regression	10.070	3.173	2.400	0.688
Nearest Neighbors	3.141	1.772	1.351	0.903

Discussion of Results

Of the models tested, AdaBoost and SGD Regression significantly outperformed SVM Regression, Random Forest Regression and k Nearest Neighbors. The AdaBoost model displayed an R2 of 0.967 with a low Mean-Squared Error of 1.072, a Root Mean-Squared Error of 1.035, and a Mean Average Error of 0.503. The SGD Regression model displayed a perfect fit R2 of 1 with a Mean-Squared Error of 0.011, a Root Mean-Squared Error of 0.107, and a Mean Average Error of 0.077. Considering the scale of the data (integer point values between 0-70), the SGD Regression results are quite good, while the AdaBoost Regression results are still acceptable.

Exhibit 3 is a representative sample of the prediction results obtained in Orange for the highest scoring players in the 2016-2017 dataset, in this case, players who scored more than 55 points in a game.

The accuracy of the prediction results is promising for future analysis and prediction of player performance and lineup optimization.

Conclusion and Suggestions for Future Research

The research presented in this paper could be expanded in several ways. Various other features could be modeled in more complex ways and matchups for teams could be assigned in an attempt to tease out the complex nuances of player performance, taking into consideration opponent physical characteristics such as weight and height, home vs away games, and season-to-date performance of both teams.

To create a reliable trading model for financial gain based on this research, the additional step of optimizing points scored per lineup needs to be addressed. This step asks the modeler to compare the modeled player intrinsic value (much as one compares the discounted expected future cash flows of a publically traded company) to the market asking price, in this case the daily

published DraftKings salary amount. Player salaries are currently calculated using a "Value multiplier", two examples of which are reproduced in *Exhibit 1.* Long-term collection and analysis of this data could result in more accurate predictions; suggesting players that are periodically undervalued, to include in a fantasy lineup, or periodically overvalued, to avoid.

Future research could compare the predicted points per player residuals to actual DraftKings salary data to evaluate market inefficiencies in the pricing of players and identify optimal DraftKings lineups using various methods in linear optimization- maximizing the number of points scored while minimizing the salary cost of a high-scoring lineup.

Appendix

Exhibit 1

Value Multipliers

The easiest way to consider calculating NBA value is to set goals for each player based on their salary. For example, you may want "X" player to score "Y" points based on "Z" salary. NBA stats are uniform among all positions (unlike in a sport like baseball where pitchers and hitters are scored differently or even football where players accumulate different point for various yardage). This allows you to create a uniform value formula across all players on your 8-man DraftKings roster. There are two multipliers you can use to calculate value as well, and they both lead to the same **target score of 300 fantasy points**.

DK Value Calculation #1: 4X+10

The first way to calculate value is a little more useful in head to head or 50-50 style contests where you don't necessarily need players with a really high ceiling, but rather you need guys with a low floor. The constant of 10 fantasy points above serves as the quickest way to account for that floor. Value is then calculated with the following formula, for example:

Value = ((Salary/1000) x 4) + 10

Player: Stephen Curry
Salary: $10,000
Value= ((10,000/1000) x 4) + 10 = (1000 × 4) +10 = 50 fantasy points

So for Steph Curry in a head to head or 50/50 game, you should ask yourself – "Can he realistically score 50 fantasy points tonight?"

DK Value Calculation #1: 6X

This second way of calculating value is better served in tournaments since we remove the floor of 10 fantasy points as the constant and use a straight multiplier of 6X salary. This ends up increasing the score needed by some of the more expensive players and decreasing the value needed from cheaper players, although you should still hope for more than 6X from a minimum priced play. Here's an example of that calculation:

Value = ((Salary/1000) x 6)

Player: Stephen Curry
Salary: $10,000
Value= ((10,000/1000) x 6) = (1000 × 6) = 60 fantasy points

https://playbook.draftkings.com/uncategorized/estimating-value-for-nba-on-draftkings

Exhibit 2

Rk	Player	Team	H/A	Opp	W/L	MP	FG	FGA	FG%	3P	3PA	3P%	FT	FTA	FT%	ORB	DRB	TRB	AST	STL	BLK	TOV	PF	PTS	GmSc
1	Klay Thomps	26-301	G	42709	GSW		IND	W	1	29	21	33	0.636	13	19	0.684	8	14	0.571	10	11	0.909	0	2	2
2	Russell West	28-115	G	42801	OKC		POR	L	1	36	21	39	0.538	18	30	0.6	3	9	0.333	13	16	0.813	1	2	3
3	DeMarcus Cc	26-129	C-F	42724	SAC		POR	W	1	41	17	28	0.607	12	20	0.6	5	8	0.625	16	17	0.941	1	12	13
4	James Harde	27-127	G	42735	HOU		NYK	W	1	42	14	26	0.538	5	10	0.5	9	16	0.563	16	18	0.889	3	13	16
5	Jimmy Butle	27-110	G-F	42737	CHI		CHO	W	1	38	15	24	0.625	14	20	0.7	1	4	0.25	21	22	0.955	3	9	12
6	Isaiah Thom	27-327	G	42734	BOS		MIA	W	1	37	15	26	0.577	6	13	0.462	9	13	0.692	13	13	1	1	1	2
7	John Wall	26-091	G	42710	WAS		ORL	L	1	42	18	31	0.581	13	23	0.565	5	8	0.625	11	14	0.786	1	3	4
8	James Harde	27-154	G	42762	HOU	@	PHI	W	1	39	16	28	0.571	10	17	0.588	6	11	0.545	13	14	0.929	1	12	13
9	Russell West	27-351	G	42671	OKC		PHO	W	1	45	17	44	0.386	15	34	0.441	2	10	0.2	15	20	0.75	3	10	13
10	Anthony Dav	23-229	F-C	42669	NOP		DEN	L	1	41	17	34	0.5	17	32	0.531	0	2	0	16	17	0.941	1	14	15
11	Kyrie Irving	24-306	G	42758	CLE	@	NOP	L	1	42	15	28	0.536	7	14	0.5	8	14	0.571	11	11	1	0	2	2
12	Damian Lilla	26-247	G	42813	POR	@	MIA	W	1	36	14	21	0.667	5	9	0.556	9	12	0.75	12	12	1	0	1	1
13	Russell West	28-054	G	42740	OKC	@	HOU	L	1	38	16	34	0.471	8	19	0.421	8	15	0.533	9	11	0.818	1	7	8
14	Russell West	28-111	G	42797	OKC	@	PHO	L	1	38	14	30	0.467	11	18	0.611	3	12	0.25	17	18	0.944	1	16	17
15	Karl-Anthony	21-015	F-C	42704	MIN		NYK	L	1	42	15	22	0.682	15	19	0.789	0	3	0	17	20	0.85	3	15	18
16	Russell West	28-091	G	42777	OKC		GSW	L	1	37	14	26	0.538	11	16	0.688	3	10	0.3	16	18	0.889	1	10	11
17	Andrew Wig	21-264	F-G	42687	MIN		LAL	W	1	41	14	21	0.667	12	16	0.75	2	5	0.4	17	22	0.773	1	3	4
18	DeMarcus Cc	26-170	C-F	42765	SAC	@	PHI	L	1	32	11	16	0.688	7	10	0.7	4	6	0.667	20	22	0.909	4	11	15
19	Stephen Cur	28-238	G	42681	GSW		NOP	W	1	36	16	26	0.615	3	9	0.333	13	17	0.765	1	2	0.5	3	2	5
20	Anthony Dav	24-000	F-C	42805	NOP	@	CHO	W	1	45	18	31	0.581	14	26	0.538	4	5	0.8	6	8	0.75	4	17	21

Exhibit 3

Prediction Results for Point scores > 55, AdaBoost, SGD, SVM Regression predictions, n = 14

PTS	Player	AdaBoost	SGD Regression	SVM Regression
70.000	Devin Booker	58.000	69.852	29.315
60.000	Klay Thomp…	50.000	60.014	31.865
60.000	Klay Thomp…	50.000	59.981	31.723
60.000	Klay Thomp…	58.000	59.999	31.939
60.000	Klay Thomp…	58.000	59.996	32.137
60.000	Klay Thomp…	50.000	59.966	31.401
59.000	Damian Lillard	52.000	58.949	33.714
59.000	Damian Lillard	52.000	59.035	33.466
59.000	Damian Lillard	52.000	58.969	33.220
58.000	Russell West…	59.000	58.094	32.064
57.000	Russell West…	59.000	57.049	31.800
57.000	Russell West…	51.000	57.043	32.544
57.000	Russell West…	59.000	57.034	32.513
57.000	Russell West…	51.000	57.010	31.723

Exhibit 4

Outlier detection on a 2897 instance subset of the 2016-2017 data. This subset includes all point outcomes > 40. The chosen regularization parameter is 10%.

Data

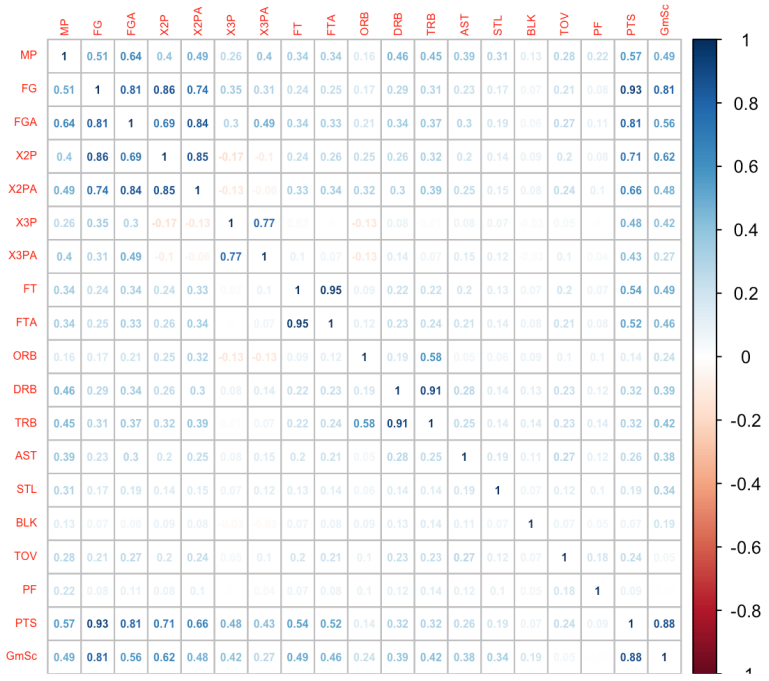
Input instances: 2897
Inliers: 2605
Outliers: 292

Detection

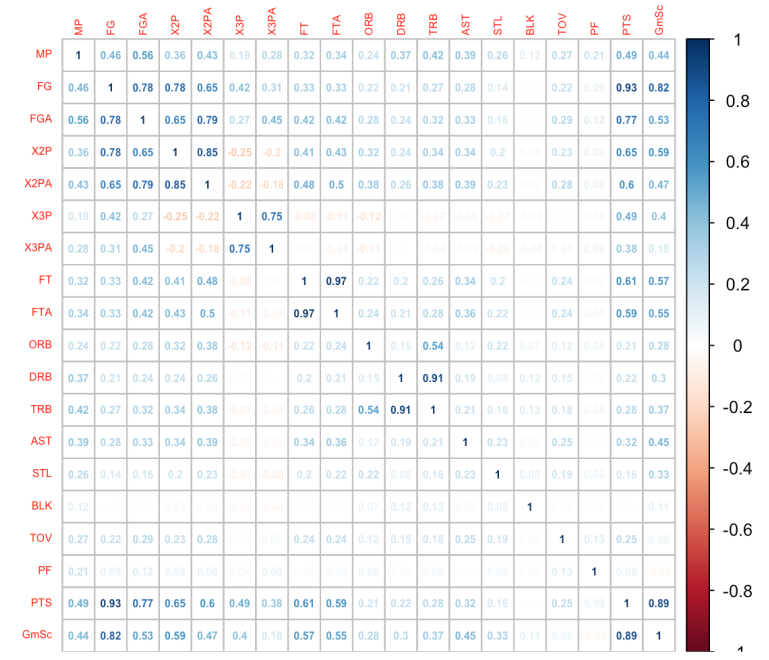
Detection method: One class SVM with non-linear kernel (RBF)
Regularization (nu): 10
Kernel coefficient: 0.01

Exhibit 5

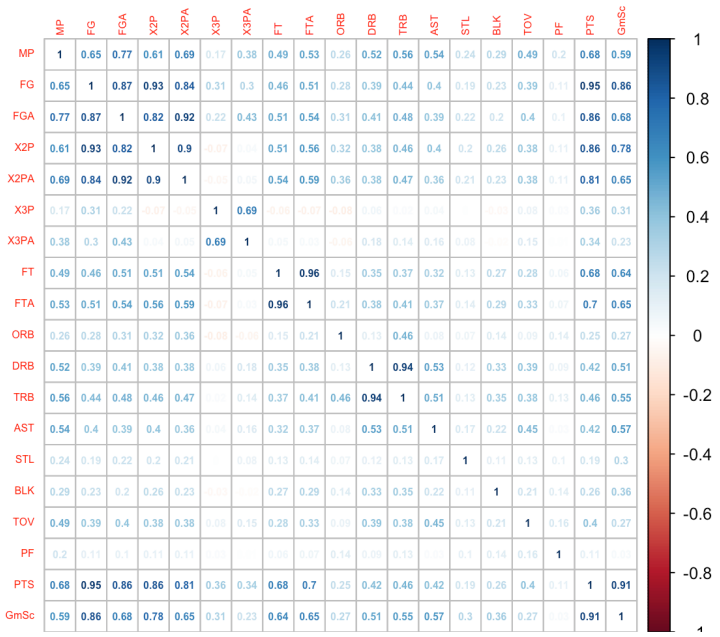
Position F Correlation Matrix



Position G_F Correlation Matrix

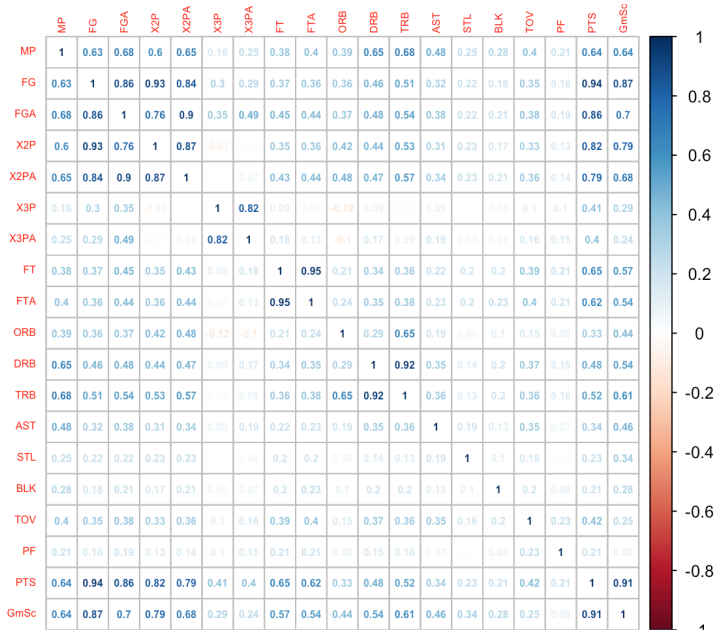


Position F_G Correlation Matrix



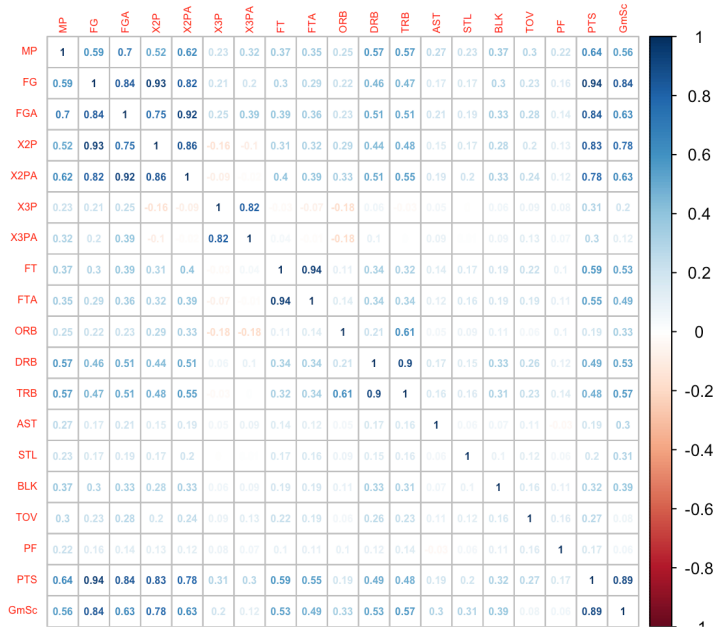
	MP	FG	FGA	X2P	X2PA	X3P	X3PA	FT	FTA	ORB	DRB	TRB	AST	STL	BLK	TOV	PF	PTS	GmSc
MP	1	0.65	0.77	0.61	0.69	0.17	0.38	0.49	0.53	0.26	0.52	0.56	0.54	0.24	0.29	0.49	0.2	0.68	0.59
FG	0.65	1	0.87	0.93	0.84	0.31	0.3	0.46	0.51	0.28	0.39	0.44	0.4	0.19	0.23	0.39	0.11	0.95	0.86
FGA	0.77	0.87	1	0.82	0.92	0.22	0.43	0.51	0.54	0.31	0.41	0.48	0.39	0.22	0.2	0.4	0.1	0.86	0.68
X2P	0.61	0.93	0.82	1	0.9	0.07	0.36	0.51	0.56	0.32	0.38	0.46	0.4	0.2	0.26	0.38	0.11	0.86	0.78
X2PA	0.69	0.84	0.92	0.9	1	0.05	0.55	0.54	0.59	0.36	0.38	0.47	0.36	0.21	0.23	0.38	0.11	0.81	0.65
X3P	0.17	0.31	0.22	-0.07	0.05	1	0.69	-0.06	-0.07	-0.08	0.06	0.03	0.09	0.1	0.08	0.08	0.08	0.36	0.31
X3PA	0.38	0.3	0.43	0.36	0.55	0.69	1	0.06	0.03	-0.08	0.18	0.14	0.16	0.08	0.13	0.15	0.09	0.34	0.23
FT	0.49	0.46	0.51	0.51	0.54	0.06	0.06	1	0.96	0.15	0.35	0.37	0.32	0.13	0.27	0.28	0.08	0.68	0.64
FTA	0.53	0.51	0.54	0.56	0.59	0.07	0.03	0.96	1	0.21	0.38	0.41	0.37	0.14	0.29	0.33	0.07	0.7	0.65
ORB	0.26	0.28	0.31	0.32	0.36	0.08	0.08	0.15	0.21	1	0.13	0.46	0.08	0.07	0.14	0.09	0.14	0.25	0.27
DRB	0.52	0.39	0.41	0.38	0.38	0.06	0.18	0.35	0.38	0.13	1	0.94	0.53	0.12	0.33	0.39	0.05	0.42	0.51
TRB	0.56	0.44	0.48	0.46	0.47	0.03	0.14	0.37	0.41	0.46	0.94	1	0.51	0.13	0.35	0.38	0.13	0.46	0.55
AST	0.54	0.4	0.39	0.4	0.36	0.09	0.16	0.32	0.37	0.08	0.53	0.51	1	0.17	0.22	0.45	0.12	0.42	0.57
STL	0.24	0.19	0.22	0.2	0.21	0.1	0.08	0.13	0.14	0.07	0.12	0.13	0.17	1	0.11	0.13	0.1	0.19	0.3
BLK	0.29	0.23	0.2	0.26	0.23	0.08	0.08	0.27	0.29	0.14	0.33	0.35	0.22	0.11	1	0.21	0.14	0.26	0.36
TOV	0.49	0.39	0.4	0.38	0.38	0.08	0.15	0.28	0.33	0.09	0.39	0.38	0.45	0.13	0.21	1	0.16	0.4	0.27
PF	0.2	0.11	0.1	0.11	0.11	0.08	0.09	0.08	0.07	0.14	0.09	0.13	0.12	0.1	0.14	0.16	1	0.11	0.09
PTS	0.68	0.95	0.86	0.86	0.81	0.36	0.34	0.68	0.7	0.25	0.42	0.46	0.42	0.19	0.26	0.4	0.11	1	0.91
GmSc	0.59	0.86	0.68	0.78	0.65	0.31	0.23	0.64	0.65	0.27	0.51	0.55	0.57	0.3	0.36	0.27	0.09	0.91	1

Position C_F Correlation Matrix

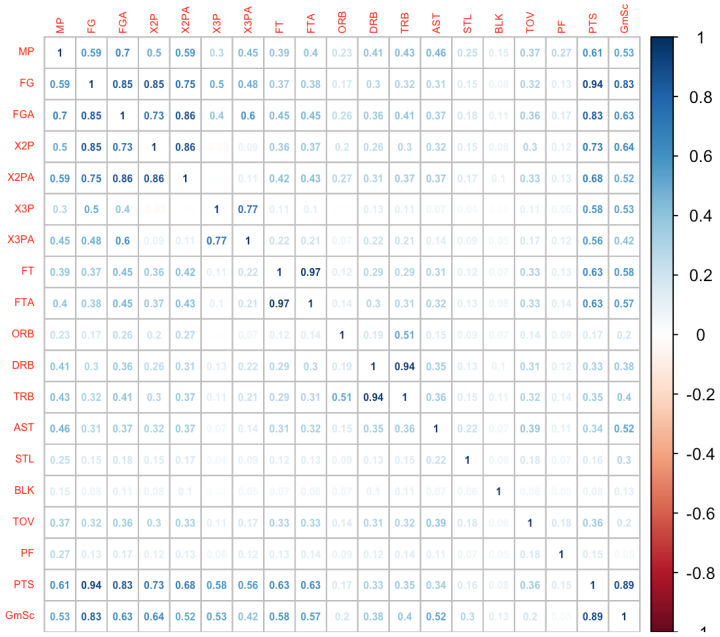


	MP	FG	FGA	X2P	X2PA	X3P	X3PA	FT	FTA	ORB	DRB	TRB	AST	STL	BLK	TOV	PF	PTS	GmSc
MP	1	0.63	0.68	0.6	0.65	0.16	0.25	0.38	0.4	0.39	0.65	0.68	0.48	0.25	0.28	0.4	0.21	0.64	0.64
FG	0.63	1	0.86	0.93	0.84	0.3	0.29	0.37	0.36	0.36	0.46	0.51	0.32	0.22	0.18	0.35	0.16	0.94	0.87
FGA	0.68	0.86	1	0.76	0.9	0.35	0.49	0.45	0.44	0.37	0.48	0.54	0.38	0.22	0.21	0.38	0.19	0.86	0.7
X2P	0.6	0.93	0.76	1	0.87	0.07	0.33	0.35	0.36	0.42	0.44	0.53	0.31	0.23	0.17	0.33	0.13	0.82	0.79
X2PA	0.65	0.84	0.9	0.87	1	0.04	0.53	0.43	0.44	0.48	0.47	0.57	0.34	0.23	0.21	0.36	0.14	0.79	0.68
X3P	0.16	0.3	0.35	-0.07	0.04	1	0.82	0.09	0.04	-0.12	0.08	0.12	0.09	0.1	0.08	0.1	0.1	0.41	0.29
X3PA	0.25	0.29	0.49	0.33	0.67	0.82	1	0.18	0.13	-0.1	0.17	0.09	0.19	0.08	0.07	0.16	0.15	0.4	0.24
FT	0.38	0.37	0.45	0.35	0.43	0.09	0.18	1	0.95	0.21	0.34	0.36	0.22	0.2	0.2	0.39	0.21	0.65	0.57
FTA	0.4	0.36	0.44	0.36	0.44	0.04	0.13	0.95	1	0.24	0.35	0.38	0.23	0.2	0.23	0.4	0.21	0.62	0.54
ORB	0.39	0.36	0.37	0.42	0.48	-0.12	-0.1	0.21	0.24	1	0.29	0.65	0.19	0.08	0.1	0.15	0.08	0.33	0.44
DRB	0.65	0.46	0.48	0.44	0.47	0.09	0.17	0.34	0.35	0.29	1	0.92	0.35	0.14	0.2	0.37	0.15	0.48	0.54
TRB	0.68	0.51	0.54	0.53	0.57	0.12	0.09	0.36	0.38	0.65	0.92	1	0.36	0.13	0.2	0.36	0.16	0.52	0.61
AST	0.48	0.32	0.38	0.31	0.34	0.09	0.19	0.22	0.23	0.19	0.35	0.36	1	0.19	0.13	0.35	0.07	0.34	0.46
STL	0.25	0.22	0.22	0.23	0.23	0.1	0.08	0.2	0.2	0.08	0.14	0.13	0.19	1	0.1	0.16	0.08	0.23	0.34
BLK	0.28	0.18	0.21	0.17	0.21	0.08	0.07	0.2	0.23	0.1	0.2	0.2	0.13	0.1	1	0.2	0.08	0.21	0.28
TOV	0.4	0.35	0.38	0.33	0.36	0.1	0.16	0.39	0.4	0.15	0.37	0.36	0.35	0.16	0.2	1	0.23	0.42	0.25
PF	0.21	0.16	0.19	0.13	0.14	0.1	0.15	0.21	0.21	0.08	0.15	0.16	0.07	0.08	0.08	0.23	1	0.21	0.08
PTS	0.64	0.94	0.86	0.82	0.79	0.41	0.4	0.65	0.62	0.33	0.48	0.52	0.34	0.23	0.21	0.42	0.21	1	0.91
GmSc	0.64	0.87	0.7	0.79	0.68	0.29	0.24	0.57	0.54	0.44	0.54	0.61	0.46	0.34	0.28	0.25	0.08	0.91	1

Position F_C Correlation Matrix



Position G Correlation Matrix



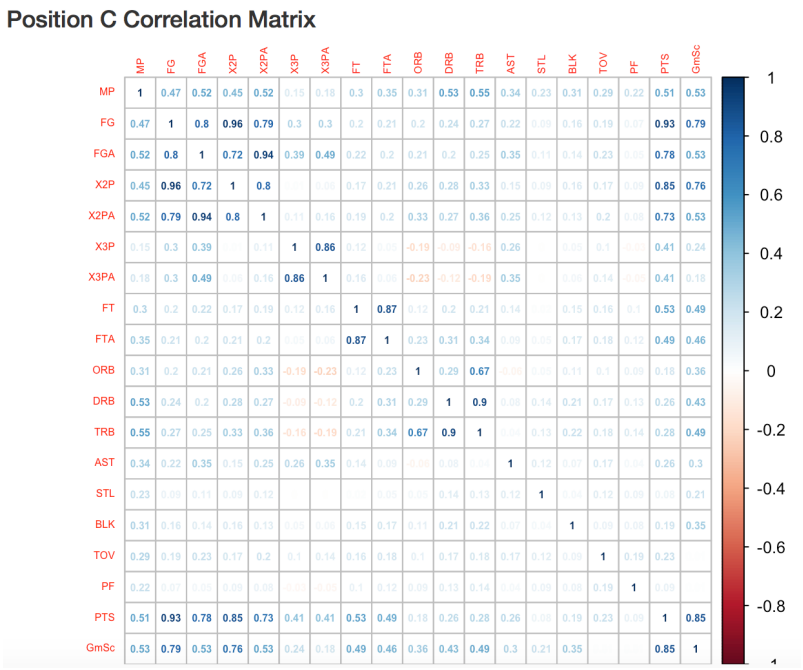


Exhibit 6

Python preprocessing (binning) script

```
import pandas as pd
import numpy as np
from datetime import datetime


NBA = pd.read_csv("/Users/thomaskalnik/Desktop/nba_2016_2017_master.csv", sep = ",")

NBA = pd.DataFrame(NBA, columns = ['Rk', 'Player', 'Age', 'Pos', 'Date', 'Tm', 'Opp', 'GS',  'MP',
'FG',  'FGA', 'FG%', '2P', '2PA', '2P%', '3P', '3PA', '3P%', 'FT',  'FTA', 'FT%', 'ORB', 'DRB',
'TRB', 'AST', 'STL', 'BLK',  'TOV', 'PF', 'PTS',  'GmSc'])

bins = [0, 10, 20, 30, 40, 50, 60, 70]

group_names = ['0-10', '10-20', '20-30', '30-40', '40-50', '50-60', '60-70']

categories = pd.cut(NBA['PTS'], bins, labels = group_names)
NBA['Binned PTS'] = pd.cut(NBA['PTS'], bins, labels = group_names)

print(NBA['Binned PTS'])

import os
path_d = '/Users/thomaskalnik/PycharmProjects/NBA ML FINAL'
NBA.to_csv(os.path.join(path,r"2016_2017_NBA_Binned.csv", sep = ","))
```

References

Fantasy basketball models/research papers:

http://www.sloansportsconference.com/wp-content/uploads/2014/06/DraftKings.pdf

https://playbook.draftkings.com/uncategorized/estimating-value-for-nba-on-draftkings

https://www.teamrankings.com/blog/nfl/fanduel-strategy-scoring-by-position

https://github.com/BenBrostoff/draft-kings-fun

http://datashoptalk.com/double-yo-money/

http://cs229.stanford.edu/proj2015/104_report.pdf

General fantasy basketball advice:

https://playbook.draftkings.com/uncategorized/4-expert-tips-to-building-a-winning-daily-fantasy-basketball-lineup

https://www.reddit.com/r/dfsports/comments/2mzqeg/anyone_interested_in_sharing_their_optimization/

Fantasy Football:

http://rotoviz.com/2014/12/using-machine-learning-create-daily-fantasy-football-projections-wild-card-round/

https://arxiv.org/abs/1505.06918

https://orbythebeach.wordpress.com/2015/09/28/how-to-build-the-best-fantasy-football-team/

https://rotogrinders.com/articles/what-s-needed-to-win-on-draftkings-368296

Optimization model:

http://www.mit.edu/~jvielma/publications/Picking-Winners.pdf

[i] **https://www.wsj.com/articles/daily-fantasy-sports-operators-await-reality-check-1441835630**

[ii] **https://www.bloomberg.com/news/articles/2015-09-10/you-aren-t-good-enough-to-win-money-playing-daily-fantasy-football**

iii http://www.espn.com/chalk/story/_/id/14623271/draftkings-bans-use-offsite-scripts-identify-experienced-players

iv http://www.basketball-reference.com/play-index/pgl_finder.cgi?request=1&player_id=&match=game&year_min=2017&year_max=2017&age_min=0&age_max=99&team_id=&opp_id=&is_playoffs=N&round_id=&game_num_type=&game_num_min=&game_num_max=&game_month=&game_day=&game_location=&game_result=&is_starter=&is_active=&is_hof=&pos_is_g=Y&pos_is_gf=Y&pos_is_f=Y&pos_is_fg=Y&pos_is_fc=Y&pos_is_c=Y&pos_is_cf=Y&c1stat=&c1comp=&c1val=&c2stat=&c2comp=&c2val=&c3stat=&c3comp=&c3val=&c4stat=&c4comp=&c4val=&is_dbl_dbl=&is_trp_dbl=&order_by=pts&order_by_asc=&offset=0

v http://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html#%5B5%5D
https://en.wikipedia.org/wiki/Kernel_method
http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html
http://scikit-learn.org/stable/auto_examples/svm/plot_svm_regression.html#sphx-glr-auto-examples-svm-plot-svm-regression-py

[vi] **http://scikit-learn.org/stable/auto_examples/ensemble/plot_adaboost_multiclass.html**

J. Zhu, H. Zou, S. Rosset, T. Hastie, "Multi-class AdaBoost", 2009.

[vii] **https://stats.stackexchange.com/questions/20622/is-adaboost-less-or-more-prone-to-overfitting**