1. Introduction

Every professional athletes will say “I have myself fully devoted to my club”. But is that true? For many basketball players in the history of NBA (National Basketball Association), their professionalism is a long-debating issue. There are always sayings that some players only play at high level in the seasons prior to their free agency eligibility, after then their placements and salaries will be finalized. But once they sign the new contract, their performance return to the previous level.. By constructing this project, we aimed to uncover the mystery of such “contract year phenomenon”.

This project firstly examined in general the existence of statistical significant difference between contact year performance and non-contract year performance. Then we constructed “slippery index”, a statistics attempting to quantify the contract year phenomenon for each individual player (how bad the player is to only perform well in contract years). At the end part, we built statistical learning models to predict players’ season salaries and evaluated whether such “slippery index” has additional predictive power on the contract salaries.

2. Expletory Data Analysis

The dataset we used was downloaded from https://www.basketball-reference.com/, a centralized database for NBA player statistics, including their season performance and salary. We randomly picked 100 players (50 from the East and 50 from the West) having complete league experience over 5 seasons with at least one contract year season. The performance statistics was measured by 23 features (Table 1):

(1) Minutes played per game
(2) Field goals per game
(3) Field goal attempts per game
(4) Field goals percentage
(5) 3_Points field goals per game
(6) 3_Points field goal attempts per game
(7) 3_Points field goal percentage
(8) 2_point field goals per game
(9) 2_point field goal attempts per game
(10) 2_point field goal percentage
(11) Effective field goal percentage
(12) Free throws per game
(13) Free throws attempts per game
(14) Free throws percentage
(15) Offensive rebounds per game
(16) Defensive rebounds per game
(17) Total rebounds per game
(18) Assists per game
(19) Steals per game
(20) Blocks per game
(21) Turnovers per game
(22) Personal fouls per game
(23) Points per game

Table 1: player performance statistics features

Sanity check and data cleaning:

We firstly examined the dataset generally. Totally we had 100 players’ data with 989 season sample, average each player 9.89 seasons. The data occurred several minor data missing at features “3_Points field goal percentage” and “Free throws percentage” (Figure 1). After sanity check, the missing was due to the absence of 3 points field goal and free throw attempt, thus the percentage ratios were undefined in the dataset. We filled the missing values with 0.

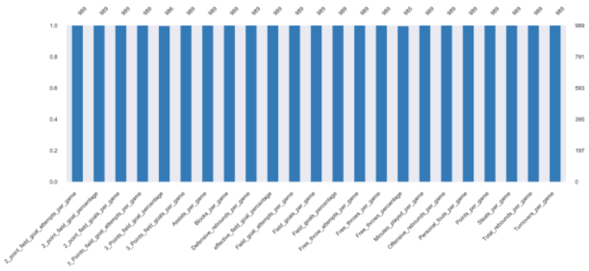

Figure 1: data missing sanity check

Feature distribution and correlation

By visualization, we examined the general distribution of some important features including "minutes played per game" (Figure 2), "points per data" (Figure 3) and "total rebounds per game" (Figure 4).

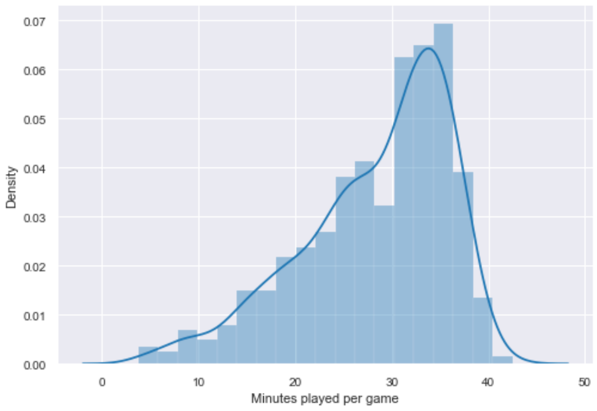

Figure 2: minutes played per game distribution

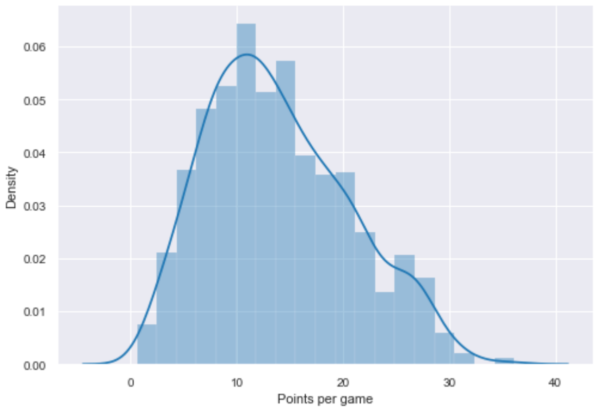

Figure 3: points per game distribution

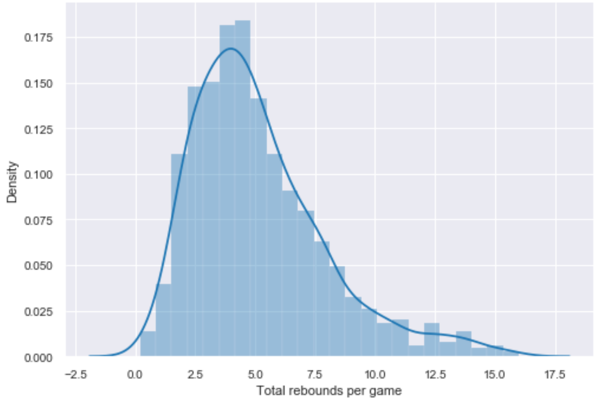

Figure 4: total rebounds per game

We see that the distributions were generally normal with skewness. And we plot the correlation matrix for all performance features (Figure 5).

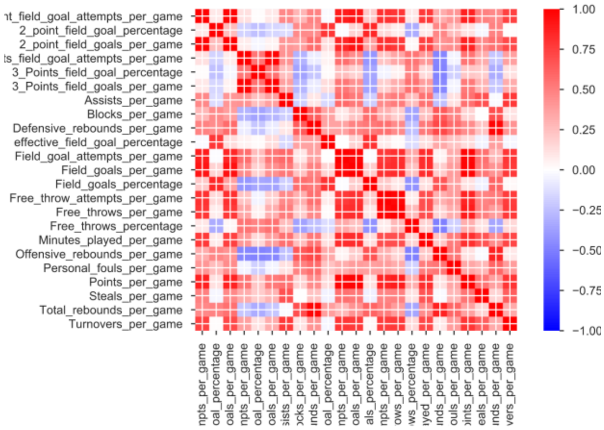

Figure 5: correlation matrix for all performance features

From the correlation matrix, we see that most performance statistics are highly correlated. Except for "percentage" statistics, as they are negatively correlated with "attempts". Thus in the "slippery index" construction, we will exclude all scoring attempts features as their information has been embedded in the percentage and scores.

3. Non-/Contract Year Performance

The first question we researched was whether there existed difference between players' contract year performance and non-contract year performance. Contract year season was generally defined as the seasons prior to players' free agency eligibility. We firstly want to see when contract year will usually occur in the players career. Thus we plot a histogram showing the happening of contract year seasons relative to entire career length (0 means the contract year happened at the player's first season; 1 means the contract year happened at the player's last season) (Figure 6)

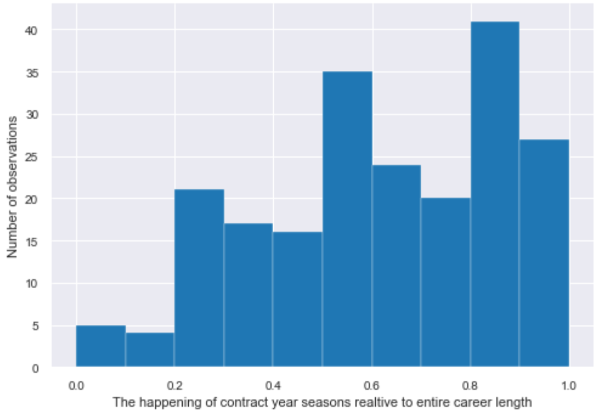

Figure 6: contract year happening time relative to career length

From the histogram we can see that more contract years happened in the twilight of the players' career. Thus a simple comparison between contract year and non-contract year will introduce age bias. Also at the end of a player's career, it could be often that the contract was extended on season based. To overcome these challenges, we focused on consecutive contract year and post-contract year comparison, as it mitigated the age bias. Also whether it presents performance reversion after the contract year should be the key to justify the existence of "contract year phenomenon".

We established our formal definition of contract/post-contract consecutive years model:

(1) contract year samples composed all contract year season which the next season was not a contract year;

(2) post-contract year samples composed all non-contract year seasons which the previous season was in contract year.

Totally we had 157 pairs of such contract/post-contract year seasons. Then we took the average of them (Table 2):

Average performance statistics	**Contract**	**Post-contract**
(1) Minutes played per game	**28.2879**	27.8
(2) Field goals per game	**5.189809**	5.13121
(3) Field goal attempts per game	**11.13057**	11.04459
(4) Field goals percentage	**0.468529**	0.464529
(5) 3_Points field goals per game	1.251592	**1.271338**
(6) 3_Points field goal attempts per game	3.417834	**3.53121**
(7) 3_Points field goal percentage	**0.320102**	0.31779
(8) 2_point field goals per game	**3.94586**	3.861146
(9) 2_point field goal attempts per game	**7.710191**	7.510191
(10) 2_point field goal percentage	0.509764	**0.510669**
(11) Effective field goal percentage	**0.524096**	0.522134
(12) Free throws per game	**2.592357**	2.57707
(13) Free throws attempts per game	**3.320382**	3.282166
(14) Free throws percentage	**0.774962**	0.768796
(15) Offensive rebounds per game	**1.235032**	1.13758
(16) Defensive rebounds per game	**4.089809**	4.068153

(17) Total rebounds per game	**5.313376**	5.205096
(18) Assists per game	3.216561	**3.296178**
(19) Steals per game	**0.954777**	0.929299
(20) Blocks per game	**0.575159**	0.569427
(21) Turnovers per game	1.740127	**1.761783**
(22) Personal fouls per game	**2.191083**	2.1
(23) Points per game	**14.22293**	14.09045

Table 2: average post-/contract year performance

From the table, we could see that 18 over 23 performance statistics, contract year season took a lead over post-contract year season.

Take the "points per game" this critical statistics as an example. We examined the distribution difference between contract year and post-contract year (Figure 7). And we could observe a clear shift to lower points per game after the contract year (blue line was the contract year; orange line was the post-contract year).

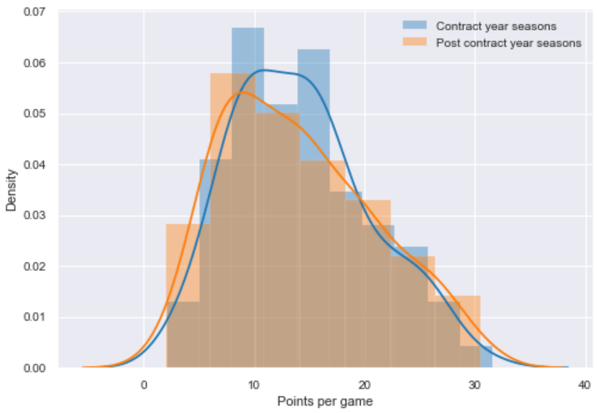

Figure 7: points per game distribution for two groups

From the above analysis, we made a preliminary conclusion that contract year phenomenon exists.

4. "Slippery Index" Construction

After reviewing the difference between contract year and non-contract year performance in general, we quantified the "contract year behavior" for individual player and contracted the "Slippery index" for each player (how slippery the player is to only perform well during contract years).

In NBA, a key statistics to evaluate the performance of NBA players is Player Efficiency Rating (PER). PER attempts to boil down all of a player's contributions into one number. Using the PER value, all the performances of players can be recorded, then weighted and integrated, and then players of different positions and different ages can be evaluated and compared. The detailed calculation of PER score can be viewed at https://www.basketballreference.com/about/per.html.

We construct our slipper index for players based on PER. In our project, we firstly calculated the PER value of all the players for all the seasons in our dataset. And for each player i, his "slippery index" is the average percentage change on PER for pair-wise contract year and consecutive post-contract year, as shown in the formula:

$$slipper\ index_i = \frac{(PER_{k+1} - PER_k)/PER_k}{n}$$

(k is a contract year and k+1 is its post-contract year; and player i experienced n contract years in his career)

Thus, we can get the slipper index for a player across his career, which will be considered as an additive feature in the salary-prediction model development.

5. Salary Prediction Model Development

We used three different techniques to build models to predict players' season salaries from their season performance statistics. When developing the models, there were two critical questions we attempted to justify:

(1) how to control the under/over-fitting and bias-variance-tradeoff to achieve higher precision accuracy?

(2) how important the "slippery index" feature is in our model and how much it could affect the salary?

Train and testing data preparation

In our models, the X are all performance statistics in a season. And we excluded all "XXX attempt" features as discussed in the

We spilt the all the data into training set and testing set by random. The training data will be used to train the models and the test data will only be used in the testing. Thus the testing performance should be a good estimation of generalized performance. For all three models, we used same ratios of train-test data split (70% for training and 30% for testing) and same random seed to maintain consistency and improve comparability.

Least-squares linear regression

The first question we were curious was how much the slippery index could affect the salary (Is the contract-year tricks players played positive or negative to their salaries expectation?). Thus the first model we fitted is the linear regression as it had nice statistical implication. After the model fitting, the coefficient $\beta_{\text{slipper index}}$ was positive and reached to 528350.25, which implied with 1% slippery index increase, the season salary will be expected to increase by 5283.5. That founding was, to some extent, in line with the phenomenon that although some players did only play exceptional good in contract year, the teams still offered them high to attract them or make them stay. Here is the bar chart for coefficients of all features (Figure 8):

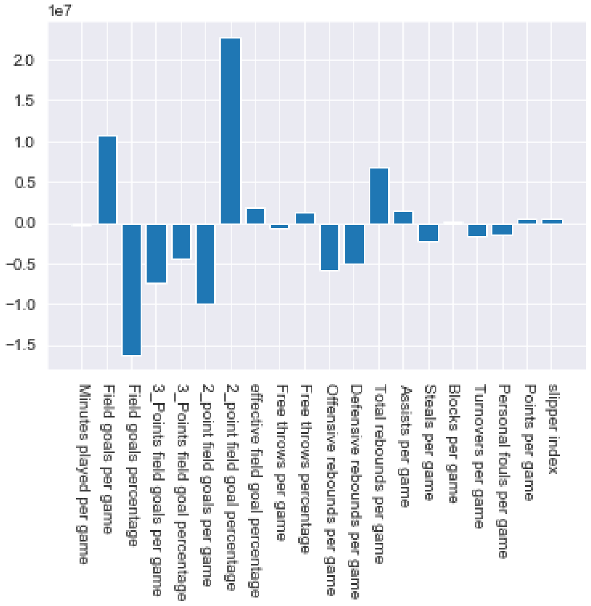

Figure 8: linear coefficients of all features

The prediction model using least squares regression achieved **27.85 R^2 score** and **$ 484,1484 main absolute error** in the testing dataset, which was not a very strong predictive model.

Ridge regression

As shown in the feature correlation analysis part, there existed median level of multicollinearity of input X. Thus to control the model variance, we attempted to use Ridge regression to shrink the coefficients magnitude. The ridge regression method is scale sensitive. Thus we firstly standardized the input to have all features with 0 mean and 1 standard deviation.

In rigid regression, a critical hyperparameter is the regularization strength index **α**, which controls the degree of underfitting and overfitting. To optimize the fittingness, we adopted the 3-folds cross validation techniques to implement grid search on hyperparameter turning. That is, for each value of α in grid searching space, we used 3-folds cross validation (split the training data into 3 equal size set; trained on 2 sets and validated on the remaining 1; then used another 2 sets for training and the rest 1 for validation) to calculated the average validation score. And we selected the α based on the best average validation score (Table 3 & Table 4).

Hyperparameter	Grid search space			
α	0.1	1	10	100

Table 3: grid search space for ridge regression

Hyperparameter	Best hyperparameter
α	10

Table 4: best hyperparameter for ridge regression

After we turned the hyperparameters, the ridge regression model was fitted. The prediction model using ridge regression achieved **28.54 R^2 score** and **$ 481,7343 main absolute error** in the testing dataset, which was an improvement to the least squares linear regression.

__Random forest regressor__

After fitting the least-squares linear model and ridge regression model, we could see the "slippery index" of the NBA players have positive impact on their salaries. But we still wanted to know how important this feature is. Also, from the low R^2 score of linear models, we can see there exist only limited amount of linear relationship between X and y. Thus we decided to implement the random forest method trying to capture some non-linearity, also to conduct feature importance analysis.

The random forest algorithm is an ensemble technique that combines multiple decision trees trained on random subset of features and subset of samples. A random forest usually has a better generalization performance than an individual tree due to randomness, which helps to control the overfitting and decrease the model's variance.

In this model, we attempted to analyses the importance of "slippery index" feature in predicting salary. Thus we prepared two random forests:

(1) reduced-form random forest, which only used performance statistics as X and did not include the "slippery index" feature;

(2) complete-form random forest, which included "slippery index" feature (same as least squares model and ridge model).

To best control over-fitting and underfitting, we optimized three important hyperparameters which dominate the fitness of model:

1. **Number of decision tree estimators**
2. **Minimum number of samples required to split an internal node;**
3. **Maximum depth of the tree.**

For both random forests, same as ridge regression model, we unutilized 3-folds cross validation to conduct grid search on hyperparameter turning:

(1) reduced-form random forest (Table 5 & Table 6):

Hyperparameters	Grid search space		
n_estimators	50	100	200
min_samples_split	2	10	20
max_depth	3	10	50

Table 5: grid search space for reduced-form random forest

Hyperparameters	Best hyperparameter
n_estimators	200
min_samples_split	2
max_depth	10

Table 6: best hyperparameter for reduced-form random forest

(2) complete-form random forest (Table 7 & Table 8):

Hyperparameters	Grid search space		
n_estimators	50	100	200
min_samples_split	2	10	20
max_depth	3	10	50

Table 7: grid search space for complete-form random forest

Hyperparameters	Best hyperparameter
n_estimators	200
min_samples_split	2
max_depth	50

Table 8: best hyperparameter for complete-form random forest

<u>Random forest model analysis</u>

In terms of random forest model performance, the reduced form random forest achieved **35.87 R^2 score** and **$ 432,7291 main absolute error** in the testing dataset. That was a large improvement compared with the linear models, which proved the random forest model did capture some additional non-linear relationship between the data (and here we even did not use the slippery index data).

After adding the "slippery index" feature, the complete-form random forest achieved **36.98 R^2 score** and **$ 431,8710 main absolute error** in the testing dataset. We could see the slippery index feature did improve the predive power of the model.

We did not satisfied with only knowing "slippery index" had improved the model. Furthermore, we wanted to quantified the feature importance. As the random forest model is built from decision tree, one of its good property becomes its ability to rank feature importance by measuring information gain (IG) though data impurity improvement in each split. Here is the comparation of feature importance on both reduced-form model and complete-form model (Figure 9 & Figure 10):

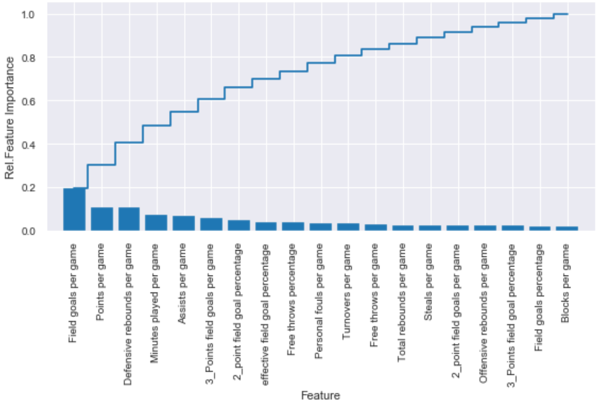

Figure 9: feature importance reduced-form random forest

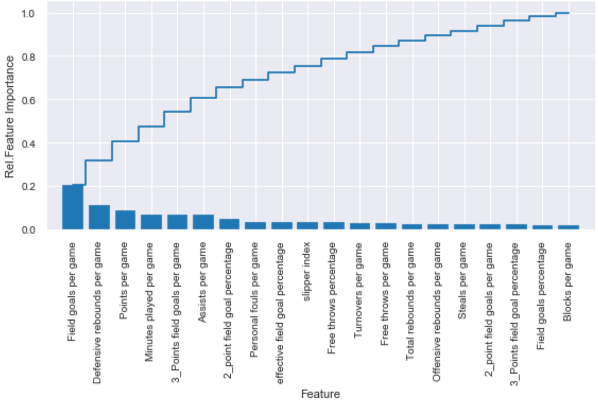

Figure 10: feature importance for complete-form random forest

For the feature importance charts, the difference was that we had "slippery index" in the complete-form model while did not have in the reduced-form model. From the charts, we can see that statistics of "field goals per game", "defensive rebounds per game" and "points per game" ranked top 3 on impact to salary for both. And other feature importance results are also pretty consistent for both models, which was a strong prove of model robustness.

By adding the "slippery index" in the complete form model, it ranked the 10th important features, which lied exactly at the medium. **The result illustrated that the tricks players played in their contract years had certain impact on their salaries, but not as important as other most significant performance statistics.**

6. Discussion and Future Outlook

How are the models?

We built total 4 machine learning models to predict NBA players' salaries using their performance statistics and/or slippery index which quantified their contract year behavior. Here is the summary of all models (Table 9):

Techniques	Slippery index usage	R^2 score on testing set	MAE score on testing set
Least-squares	Y	27.85	$ 484,1484
Ridge	Y	28.54	$ 481,7343
Random forest (reduced form)	N	35.87	$ 432,7291
Random forest (complete form)	Y	36.98	$ 431,8710

Table 9: summary of model performance

It could be seem that after a good control of underfitting and overfitting, the random forest model improved the model performance and it had better generalization ability. Also we discussed the medium effect of contract-year behavior on salary expectation.

Weapons of math destruction? Fairness?

Another critical question was whether our model could become a weapon of math destruction. The answer seemed to be yes. After seeing that the tricks some players played in their contract year actually increased their salary on average, the team managers and coaches might be more conservative on the contract salaries when being impressed by an exceptional performance happening in a contract year.

In terms of the fairness issue, the existence of WMD was under the scope of NBA league in general. And our model did not included any personal information about the players, as their names were excluded in out model. Thus it would not create privacy issue or affected any individual.

The league in the future

Maybe no one really likes contract year phenomenon. As a response, we can see that more and more NBA new contracts are not just about player's basic salary. For example, a few days ago, Adebayo signed a deal worth at least $163 million with Miami Heat. The contract could escalate to $195.6 million if Adebayo meets criteria such as making an All-NBA team. Those criteria could possibly relieve such phenomenon in the league. But it's hard for us to discuss whether this kind of criteria in the contract is meaningful on earth because the lack of such detailed data and contract information currently.