# Evaluating and Classifying NBA Free Agents

Shanwei Yan

In this project, I applied machine learning techniques to perform multiclass classification on free agents by using game statistics, which is useful to evaluate players' performances. To do so, I implemented and compared a collection of supervised and unsupervised learning algorithms on a dataset consisting of 282 free agent samples. No previous work was accessible, and depending on the features I selected, Naïve Bayes outperformed the other algorithms and achieved an accuracy of 76%.

## Introduction

NBA, as a sports league, is different from many other leagues in rest of the world, because there is a "salary cap" for each team. This salary cap regulates that the team whose total payroll is above this cap will be penalized a huge amount of "luxury tax". Considering the heavy excessive finance burden, only a few teams among 30 are willing to pay luxury tax like Miami Heat and Brooklyn Nets. This regulation stops the basketball games from becoming a financial competition instead of an athletic one.

As a general rule, star players' annual salary range from 10 million to 30 million, while inept or rookie players' annual salary can be as low as 600 thousand. Since the sum of players' salaries of one team can be considered as a fixed number, assuming every team controls its payroll under the salary cap, in order to improve a team's performance, it is very important for a manager to sign good rookies and those who seem to be maimed player and get low paid, but can contribute a lot to the game. Therefore, it is surprising but also understandable that a team which spent 40 million can rank higher than a team which spent 80 million. A good example is Houston Rockets, which use statistical analysis to evaluate a player's skills, and sign low cost but good players. It is interesting and inspiring to dig out how they do that, and what the secrets are.

Though many kinds of game statistics are available online, there is few prior work open to the public. In this report, I investigated different machine learning algorithms to build multiclass classification models to fairly evaluate players' values, and find highly cost effective players. I am able to achieve high precisions in classifying inept and super star players.

## Data and Feature Selection

The first question is what salary can represent a player's value. NBA regulates that once a player signs a contract, no matter how his performances are in the following year, his salary won't be adjusted. For example, suppose LeBron James signs a contract which states that he can earn 15 million for the following 5 years. Then next year, he gets severe injured and plays as bad as a player who earn 1 million annually. He can still earn 15 million, but not 1 million. Therefore, it is not suitable to use all players' salaries as samples to train, because they can be highly biased. Instead, I believe that it is fair to use

players' statistics whose contracts are expiring, who are called free agents. The reason is that free agents earn the new salary for year t+1 depending on their performances in year t. Therefore, I gathered and cleaned the data for the latest three regular seasons (2010-2011, 2011-2012 and 2012-2013), and got 282 samples. The explanatory variables are game statistics, like PPG (points per game), RPG (rebounds per game), TS (true shooting rate), etc. And the responsible variable is salary.

Depending on the distribution of salary range, I decided to build multiclass classification models. Then I classified the salary data into 4 classes. The salary range for each class and priors are listed in Table 1. There are not many samples, which prevents me using more classes. On the other hand, too few classes (i.e. 2 classes) make the classification problem meaningless.

Table I - Salary Ranges

| Name | Range | Prior |
|---|---|---|
| Low Salary | <= 2 million | 124 (44%) |
| Middle Class | 2 million ~ 5 million | 94 (33%) |
| Good Player | 5 million ~ 8 million | 32 (11%) |
| Super Star | > 8 million | 32 (11%) |

In the dataset, there are hundreds of attributes indicating players' performances, either per game or overall season. Due to the high dimensionality of the feature space, techniques such as forward or backward searches are not effective. Therefore, I turned to use a batch of attributes, the so-called "efficiency" to measure players' values. According to tens of NBA data analysis reports, different experts defined "efficiency" by using different attributes and parameters, based on different points of views.

In order to find the efficiency with most predictive power, as a first step, I scored the existing efficiencies according to the correlation with class labels. As a result, 5 efficiencies ranked top among 17. They are defined similarly and put more weights on offensive statistics but also take defensive skills into account. Next step is to select additional attributes so that player's potentials are considered, and more defensive statistics are used to balance offensive ones included in efficiency. Finally I decided to use the 5 "efficiencies" combining with other potential/defense features which achieved the highest correct rate when implementing the Naive Bayes algorithms on the overall dataset.

## Models and Results

I explored both supervised and unsupervised learning algorithms for the classification task. In particular, I used the following techniques: Naive Bayes, SVMs and K-mean Clustering. The accuracies of the learning algorithms are evaluated using 10-fold cross-validation. Figure 1 shows the classification pipeline.

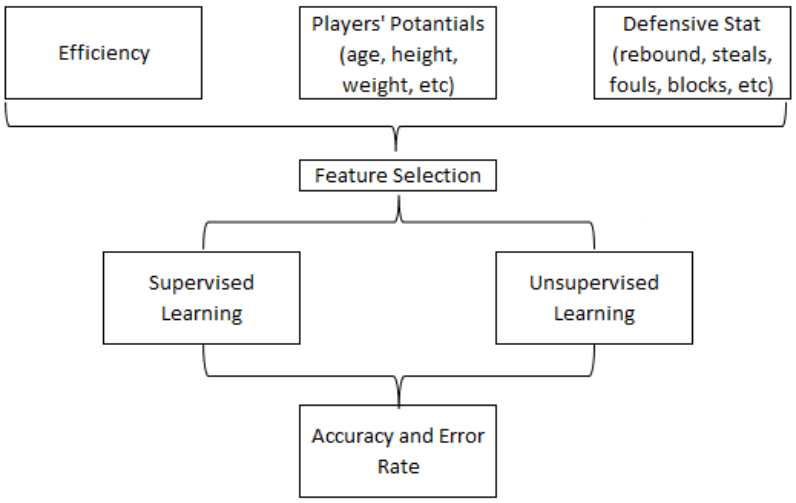


Figure 1 – Classification Pipeline

**Naive Bayes**

Randomly partition the original data set into 10 equal-size subsets (each set consists of 28 to 29 samples), and implement the Naive Bayes algorithm on the training set to build the classifier. Since the features I used were all continuous, then Gaussian distributions were generated for each predictive variable: one for each class (salary range). Ran 10-fold cross-validation for 100 times, and the results were listed in Table 2.

**SVM**

I tried SVMs with four different kernels, including linear (u' * v), polynomial ((1/dimension * u' * v)^3), radial basis (exp(-1/dimension * |u-v|^2)), and sigmoid. Ran 10-fold cross-validation for 100 times, and listed the results in Table 2. Among the 4 kernels used, radial basis had the highest accuracy. Though the performances of SVMs are generally worse than Naive Bayes Classifier, one possible reason is that I selected the features which have the best performance depending on Naive Bayes.

Table 2 – Error Rates of Supervised Learning

| Error Rate | Mean | Std | Min | Max |
|---|---|---|---|---|
| Naive Bayes | 26.65% | 2.69% | 19.60% | 33.20% |
| SVM - Linear Kernel | 29.43% | 2.74% | 22.86% | 37.50% |
| SVM - Polynomial Kernel | 32.45% | 3.04% | 26.79% | 41.07% |
| SVM - Radial Basis Kernel | 28.97% | 2.75% | 22.50% | 35% |
| SVM - Sigmoid Kernel | 40.93% | 3.07% | 32.50% | 48.93% |

**K-means Clustering**

I applied the unsupervised k-means approach to cluster samples into different groups. Though I previously classified the salaries into 4 ranges, here I tried k = 2 to 6, with starting points assigned randomly. By running the K-means Clustering algorithm for 100 times, the clustering patterns are shown in Table 3.

The error rates for k=2, 4 and 6 are respectively 25.18%, 28.9% and 37.1%. (According to the clustering patterns of different K, it is hard to calculate the error rates for k=3 or 5.) K=2 has the highest accuracies, but in real NBA world, it makes no sense to classify a free agent into only 2 class, whether he is a skilled or inept player. The error rate of K=4 is close to those of Naive Bayes and SVM. K=6 suffers from a very high error rate, which may be caused by the small sample size.

Table 3 – K-means Patterns

| K | Pattern |
|---|---|
| 2 | Class 1 and 2 are combined as one cluster, and Class 3 and 4 form the other. |
| 3 | Class 1 and some Class 2 samples form one cluster, Class 4 and Class 3 form another cluster, and the rest samples make up the other cluster. |
| 4 | Compared to the original classification, Class 1 and 4 enjoy the high accuracy, but Class 3 has relatively higher error rate. |
| 5 | Class 1, 2 and 3 are mixed and make up 4 clusters, while Class 4 exclusively makes the other cluster. |
| 6 | Class 1 and 2, each forms 2 clusters, and Class 3 and 4 forms 1 cluster respectively, which is expected. |

## Conclusion and Future Work

It can be seen that using efficiencies and potential / defense features in the classification models could generally get good prediction accuracies, which would help to evaluate basketball players' true values and dig out under-rated players. However it is not likely to improve the testing accuracy by elaborating efficiencies or re-selecting features, because the five efficiencies I used resulted in close error rates. On the other side, improving the parameter settings in certain machine learning algorithms like SVM, k-means can improve the accuracies a little bit. Another way to improve the accuracies I think is to increase sample size. This may also help to test whether the selected features are good enough to classify player's performances or not.

Another possibility of future work is to create an application tool, which uses the trained classifiers to evaluate all players' values, and then offers trade suggestions to team manager: players in the team who earned much more than their contributions should be traded out for those players who earn equivalent salary but better performances. This will be especially useful during the summer time (off season) or

mid-February (before the close date of trade), because it will help to establish a more competitive and cost effective team.

References

[1] A. Karatzoglou, D. Meyer and K. Hornik, Support vector machine in R. Journal of Statistical Software (2006).

[2] A. Karatzoglou, A. Smola, K. Hornik (2009)."kernlab An S4 Package for Kernel Methods in R".

[3] Machine Learning Lecture Notes, http://cs229.stanford.edu/materials.html

[4] Michael Lewis. Money Ball

[5] NBA Salary Cap, http://en.wikipedia.org/wiki/NBA_Salary_Cap

[6] Player Evaluation Metrics, http://www.nbastuffer.com/component/option,com_glossary/func,display/Itemid,90/catid,42/

[7] Player Stats, http://www.nbastuffer.com/player_stats