Apply machine learning to NBA shot prediction

Fan Yang  
Pennslyvania State University Penn State College of IST State College   
fvy5046@psu.edu

*Abstract*—At any level of basketball, including NBA, shot selection is vital to wining to game of basketball. Shot selection is evaluated based on many features. Such as the shot distance, the distance of the nearest defender, minutes remaining on the clock. It’s difficult for coaches and players to know which shots are good and which are not. However, the process of understanding shot selection can be significantly enhanced by using machine learning. This paper will implement 6 machine learning algorithms including Logistic Regression, Naïve Bayes, Neural Network, Random Forest, XGBoost and SVM on player’s performance data to help quantifying shot quality in the NBA.

Keywords—NBA, Machine learning, shot, prediction, XGboost

# Introduction

Computer-based prediction of the outcome of sports events has a long history. Especially since the advent of deep learning, the application of advanced statistical methods to sports events have become increasingly important, such as the injury rate of baseball players next season, such as shooting a three-pointer in a basketball game. Artificial intelligence is much better than us when it comes to making sports predictions by leveraging historical data. Morey, the general manager of the Houston Rockets, the most profitable franchise in the NBA, firmly believes that decisions made with the support of data are the best decisions. Morey adapt advanced data analysis into a scouting and training framework. In recent years, few concepts have seen as much discussion as Moreyball[2], the title given to NBA analytics as popularized by staffer Daryl Morey.

At any level of basketball, including NBA, shot selection is vital to wining to game of basketball. Shot selection is evaluated based on many features. Such as the shot distance, the distance of the nearest defender, minutes remaining on the clock. It’s difficult for coaches and players to know which shots are good and which are not. However, the process of understanding shot selection can be significantly enhanced by using machine learning.

In this paper I will try to predict whether an NBA player's shot will hit or not based on his shooting stats. To achieve this, this paper will implement six machine learning algorithms: logistic regression, SVMs, Naïve Bayes, Neural Networks, Random Forests, and boosting.

The data set includes nearly 130,000 shooting data of 281 players in 904 regular season games of 30 NBA teams in the 2014-15 season. The data includes 21 variables: both sides of the game, home and away games, wins and losses, shooting players, defensive players, shooting distance, and number of hits, etc.

# Literature review

## LSTM model in sports analitics

This Paper uses a popular variant of RNN with long short-term memory (LSTM) units. The unique value of RNN is that it can efficiently process sequence data. For example, article content, audio, and stock price trends.

The reason why RNN can process sequence data is that the previous input in the sequence also affects the subsequent output, which is equivalent to having a "memory function". But RNN has serious short-term memory problems, and long-term data has little impact (even if it is important information). So based on RNN, there are variants of LSTM and other algorithms. These variant algorithms mainly have several characteristics: long-term information can be effectively retained. In this study, the input to the LSTM is the XYZ data and the game clock. After the RNN predicts both the probability of a successful shot and parameters for the mixture density network (MDN), The probability comes from a SoftMax layer and is trained based on cross-entropy error.

The results for the classification model used in this paper is AUC which is the area under the receiver-operating characteristic curve. The RNN model outperforms the baseline models at all distances which shows that the RNN was able to learn and classify basketball trajectories. This paper confirmed that RNN model are able to learn non-linear behavior like predicting basketball shots.

## The importance of Cross-validation

The aim of the research is to predict every field goal attempt during Kobe’s 20-year career, whether he made the shot or not. The goal is to assess the machine learning models’ performances on predicting whether basketball shots were made using trajectory and performance data. This paper first applies PCA for column transformation, followed by minmax scalar normalization, and then pass this scaled data to 7 machine learning algorithms (Logistic Regression, Random Forest, Linear Discriminant Analysis, Naïve Bayes, Gradient Boosting, Adaboost and Neural Network were used in this paper) for classification whether Kobe made the shot or not. The experimental results show that Adaboost performs best as compared to other methods with both hold-out and 5-fold cross-validation in shots prediction. Results of all methods have shown improvement when using cross-validation as compared to the hold-out method. This paper concluded that by using cross-validation, the result of all methods increases as compared to the hold-out method. By comparing the results, XGboost has a higher accuracy as compared to the Random Forest. The result from this paper has confirmed that regarding predicting shots, XGBoost performs relatively higher than Random Forest.

## The power of Gradient Boosting.

The goal of this paper is to predict whether NBA players make shots to analyze what factors affect the shooting percentage. By analyzing the past work, this paper points out spatial data is an important feature to improve accuracy. Implementing factorization machine model appears to be a method to improve the results.

Machine learning techniques including Logistic Regression, Random Forest, SVM, Naïve bayes, Neural Networks, and boosting were used in this paper. Among the 6 algorithms, boosting outperform the other methods. Random forest comes second in terms of performance. XGBoost achieved a 68% accuracy, Random Forest have a 61% accuracy while other algorithms scored below 60%. Boosting algorithms and Random Forest outperform other algorithms which confirm the status that Random Forest and Boosting are idea model for predicting shots.

Neural Network’s performance score failed to achieve the state-of-the-art result from Harmon et al. Removing categorical data appears to be the reason behind the result. By examining the importance level of each feature, this paper points out that the distance between a shooter and the closest defender is a strong predictor in terms of predicting shots.

## Tree based models in Shot selection

The purpose of this paper is by predicting whether NBA players make shorts to analyze what factors affect the shooting percentage. To achieve this, this paper compares the results performed by two machine learning method that were based on combining decision trees and then analyze the important features that affect the shot.

From analyzing the past work, this paper points out that the location of ball, offensive players, defensive players are important attributes that were used in relative research paper. Adding attributes like shot distance and the closest defender could improve the results. Murakami also believe that CNN and FFN might be better for image data inputs for predicting shots.

This paper compared two machine learning method that both are used for classification by building individual trees and combining their results-Random Forest and XGBoost. Random Forest is a bagging technique that contains several decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. Instead of relying on one decision tree, random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. And because of that, the con of using random forest is that predicted value won’t be optimized. In the boosting method, all the individual models are built sequentially. Which means the outcome of the first model passes to the next model etc.  In bagging, the models are built parallel, the error of each model is unknown. Whereas in boosting once the first model was built, the error of that model is available. So, when pass the first model to the next model, the intention here is to reduce the error further. Because the way of trees is built, boosting method tends to overfit. It needs special care on parameter tuning.

In this paper, XGboost without parameter tuning had a 62% accuracy score, 68% with parameter tuning, where Random Forest had a 57% accuracy score. The experiment result in this paper have shown that shot distance, shot clock and closet defenders turned out to have strong connection with shot prediction. XGboost had a better score than Random Forest for shot prediction.

# Data

1. Data Source

The data for this study were obtained from Kaggle【1】. It collects data on 128069 field goals made during the 2014-2015 NBA regular season. There are 21 features in the dataset and mainly provide four types of information: Game information, shot information, Time information and the defense information. Game information includes the matchup, location, and win/loss record of the game. Shot information includes shot number, shot distance, and point type. Time information are consist of game clock, shot clock, and touch time. Defense information are the closest defender of the shooter and the distance between the defender and the shooter. Once the raw data were loaded, they were processed using python.

1. Data processing

Unprocessed features have missing values, outliers, error values, data format and other issues. Discrete variables and time series need to be converted to the value that can be used as input. The first step in the data preprocessing phrase is to convert all the feature values into something we could use: Converting game clock to seconds, replacing abnormal values with NaNs, converting type of shots to categorical, Converting location information to categorical. The second stage of data preprocessing is scaling: Standardize attributes like shot clock, robust scaling on shot number, dribbles, touch\_time and closest defender, Min-Max transformation on period, game clock and shot distance, filling NaNs with mean. Last, dropping columns that will not provide useful information to modeling. The input features remained are listed below:

1. Home\_match: 1 represents home game, 0 represents away game.
2. Shot\_number: Numbers of field goals attempt. 1 represent 1st attempt, 2 represent 2nd attempt and so on.
3. Period: which period were played in.
4. Game\_clock: measure the time of each field goal attempt.
5. Shot\_clock: offensive team’s shooting time within 24 seconds.
6. Dribbles: how many dribbles were made before shooting.
7. Touch\_time: the number of times a player touched the ball in an attacking position on the floor.
8. Shot\_dist: distance from the shooting player to the rim.
9. 3pts\_shot: 1 represent 3 pointers were made and 0 represent 2 points shot.
10. Hit: classification output. 1 represent shot made successful, 0 represent shot missed.

The full data pre-processing detail can be viewed at <https://github.com/Tudou77/DS340W>.

1. Feature importance

Figure 1 shows a visualization of feature importance generated by Random Forest model. Variables are ranked by relative importance for predicting shot among NBA players. The graph indicates that the most important variable is shot distance. The relative importance is expressed as a fraction based on the weight of each variable, with 1.0 being the most important and 0 having no contribution to the model. Chart, bar chart

Description automatically generated

Figure 1. Feature importance graph generated by Random Forest model.

# Implementation

Algorithms were developed to predict player’s shot. 6 classification machine learning models were built. Including Logistic Regression, Naïve Bayes, Neural Network, Random Forest, XGBoost and SVM. Each model will be evaluated with precisions and accuracy score, which is calculated as the ratio of correctly predicted positive examples divided by the total number of positive examples that were predicted. Since the dataset is relatively balanced, Accuracy will be the main evaluation to assess the performance of our model.

Data were split into train set and test set using an 80/20 split. Once train set and test set were split, 6 different model algorithms were created. Models were built using the scikit-learn Python library.

Table

Description automatically generated

Table 1: Performance score of each model evaluated by precision and accuracy.

Table 1 shows each model’s precision and accuracy score. Collectively, our results appear consistent with the state-of-the-art research paper. XGBoost model appears to have the highest accuracy score among all six models. Neural Network and Random Forest and Logistic Regression all have a accuracy higher than 60%.

# Novelty

1. Parameter tunning

XGBoost has a wide range of applications in machine learning and data mining. According to statistics, among the 29 winning proposals on the Kaggle platform in 2015, 17 teams used XGBoost; in the 2015 KDD-Cup, the top ten teams all used XGBoost, and the improvement that integrating other models brought are no better than just adjusting the parameters of the XGBoost model. These real-world examples show that XGBoost can achieve very good results on a variety of problems. Constructing a model that uses XGBoost is straightforward. However, improving the performance of this model is somewhat difficult. The XGboost algorithm uses several parameters. Therefore, in order to improve the performance of the model, it is necessary to adjust the parameters.

GridSearchCV is implemented here for parameter tuning. GridSearchCV has two functions, GridSearch and cv. GridSearch will adjust the parameters in turn within the specified parameter range, use the adjusted parameters to train the dataset, and find the parameter with the highest accuracy on the validation set from all the parameters. This is a process of training and comparison. CV is to use k-fold cross validation to divide all data sets into k parts, take one of them as the test set without repetition, and use the remaining k-1 parts as the training set to train the model, and then calculate the model's performance on the test set. Score, the k scores are averaged to get the final score.

The first step is to set the xgboost parameter search range of grid search. I set the main 6 parameters of XGBoost {min\_child\_weight, learning\_rate, n\_estimateors, subsamle, max\_depth, colsample\_bytree}

When the grid search is done, we get the ideal value of max\_depth is 5, the ideal value of min\_child\_weight is 0.0001, the ideal learning\_rate is 0.14, the idea n\_estimators is 153. Table 1 shows the performance score of XGboost before and after parameter tunning. As we can see model’s performance have great improvement after parameter tunning.

|  |  |  |
| --- | --- | --- |
| XGBoost | Precision | Accuracy |
| Before | 0.6568 | 0.6180 |
| After | 0.6880 | 0.6223 |

Table 2: Precision and accuracy score for XGboost model before and after parameter tunning.

One thing to note is that as the model improves, it becomes exponentially harder to improve further, especially if the performance is close to perfect. Several GridSearch are performed, and the improvement is subtle after achieving an 62% accuracy score.

1. K-fold cross validation

Cross-validation is a statistical method for evaluating generalization performance, which is more stable and comprehensive than the method of dividing the training set and the test set once. In cross-validation, the data is divided multiple times and multiple models need to be trained. The commonly used cross-validation is k-fold cross-validation, where k is a number specified by the user, usually 5 or 10. When performing 5-fold cross-validation, the data is first divided into (roughly) equal 5 parts, each called a fold. Next step is train a series of models. Use the 1st fold as the test set and the other folds (2~5) as the training set to train the first model. Use 2-5 folds of data to build the model, then evaluate accuracy at 1-fold. Then build another model, this time using 2 folds as the test set and data from 1, 3, 4, and 5 folds as the training set. Continue to repeat this process using 3, 4, and 5 folds as the test set. Accuracy is calculated for each of these 5 splits of the data into train and test sets. Table 2 shows all the 5 accuracy values of each model we implemented. Noted that since SVM model and XGboost model implemented GridSearchCV which provide internal cross-validation function, therefore SVM model and XGBoost model are not included in this process.

Table

Description automatically generated

Table 3: Accuracy score after implement 5- fold cross validation

The accuracy score from Table 3 after implemented 5-fold cross validation are stable and consistent, which conveys the idea that our models is not biased and is unlikely to overfit.

1. LSTM model implementation

In“Applying Deep Learning to Basketball Trajectories” [3], researchers applied LSTM model in shot prediction using spatial data such as trajectories of the ball. The findings of this study can be understood as LSTM model performs better than tree-based model. Therefore, adding LSTM model for comparison is an important step in this topic.

Chart, line chart

Description automatically generatedFigure 2. LSTM model loss vs accuracy

Figure 2 shows the performance of the LSTM model with 12 epochs. The loss decreases and the accuracy increase as the training process goes on. Which indicates the training process is going well. The LSTM model is able to achieve a 63.75% accuracy. By implement LSTM model, I tested the hypothesis that LSTM model works well in shot prediction.

1. Principle component analysis

Principal components analysis (PCA) is one of the most important dimensionality reduction methods. It has a wide range of applications in the fields of data compression to eliminate redundancy and data noise removal. We performed PCA on the dataset during the preprocessing stage and remained a train\_test split using the PCA reduced dataset.

A picture containing chart

Description automatically generated

Figure 3. Cumulative sum of explained variance

Figure 3. shows how much variance is explained by adding a new component. We can see that fourth component includes nearly 90% of the information. The performance of PCA reduced dataset is shown in Figure 4.

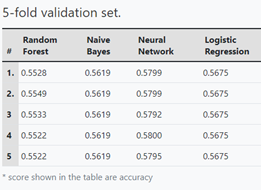


Figure 4. Accuracy score on PCA reduced dataset.

From the accuracy score on PCA reduced dataset, Random forest shows nearly 5% lower on accuracy score, 2% lower on Naïve bayes, 4% lower on Neural Network, and 5% lower on Logistic Regression. The analysis leads to the conclusion that scaled dataset lost a certain level of information after dimensionality reduction. Therefore, implement PCA is not a good choice when facing databases with relatively few variables.

# conclusion

In this paper, the main purpose is to predict basketball shot and compare the ability of different model regarding this topic. This task was formulated as a model optimization problem.

One of the contributions is to provide a detailed data processing method that could be used in future work.

A discussion on proposing a solution to overfitting based on k-fold cross validation has been provided. In particular, without introducing validation set, the performance score of each model tends to float between 5%. Implementation k-fold results in more stable and consistent score.

The main focus of this paper is implementing LSTM model and present its performance. The contribution of this implementation is twofold. First, it is the first time that LSTM has been applied on this type of data. Second, an experimental comparison of LSTM with other models was first performed. The experimental result provided in this paper confirms the ability of LSTM model handling shot classification.

Finally, another contribution relies in testing the unnecessity of dimension reduction when facing datasets with relatively few variables.

This study is an example of implementing ML into sports analytics and provides a foundation for future studies.

# Future work

This paper compared each classification model’s performance on shot predicting. Several methods are introduced to increase the accuracy of prediction. Alternative ideas, experiment and methods have been left behind for future due to time concerns.

There are three major directions in this study that could be addressed in future research:

1. Introduce more defender’s information to the dataset. In this paper, the attribute “closest defender” is not included as input data. Finding a method to utilize defender’s information is the key to improve model’s performance. One way to achieve this is replacing closest defender with defenders’ statistic attributes to the dataset, variables such as heights, standing reach, and wingspan. Another way is simply adding defensive rating of each players as one of the input data. Advantages of this approach are: First, it would result in great return if the added variables could give the model more information to learn. Second, player’s physical information is easy to acquire, NBA provides full statistic on each player’s physical information. Possible downside of this approach is it might add noise to the model, therefore causing bias on certain string which would result worse performance. What defender’s information brought to the algorithm is the defensive ability of the player, and the best optimization of this type of attributes is under the assumption that all players are actively playing defense all the time when they are on the court. Reality might differ from theory. In reality, if defenders were not actively playing defense, adding player’s defensive ability could provide irrelevant data which can significantly affect the result of prediction.
2. Introduce more spatial data such as the location (X, Y, Z) of each shot. Figure 1 indicates that the shot distance is an important feature in this topic. Adding more spatial data provide a potential mechanism for model to learn more information on each individual player. The most intuitive information provided by the location of each shot is each player’s shot probabilities made from different angle and different areas on the court. In theory, each player has their hot zones where in that position, they are most confident to make the shoot. Hot zone feature was introduced by NBA.

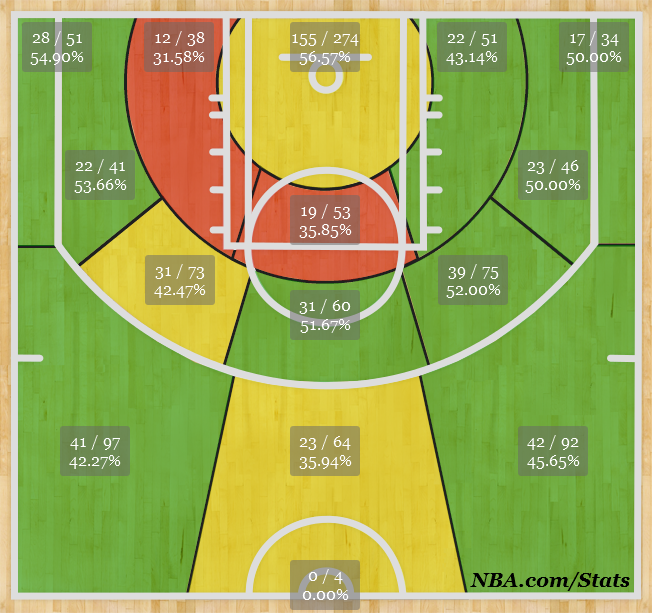


Figure 5. Stephen Curry’s shot selection from’10-’11 season [7]

In Figure 5, Player’s shooting ability was ranked in 3 levels in 14 spots in the courtroom, where green indicates premium field goal attempt, yellow indicates average field goal attempt and red indicate the worst field goal attempt. Hot zone data defined player’s confidence level of this specific shot. The advantage of this approach was discussed in [8], researchers concluded that spatial data such as the location of the ball and offensive and defensive players are important features in predicting shots. This is very much the key component in future attempts to improve the model performance. The limitation of this approach is spatial data is relatively difficult to acquire.

1. The implementation of BiLSTM model. Bi LSTM is an extension of traditional LSTM. BiLSTM is a combination of forward LSTM and backward LSTM, which can improve the performance of the model on sequence classification problems. In [9]

, Researchers concluded that bidirectional networks are significantly more effective than unidirectional ones and is not confined to time series data. This assumption might be addressed in the future studies.

# References

1. Kaggle nba shot logs. https://www.kaggle.com/ dansbecker/nba-shot-logs, 2016.
2. Bigdata, “Moreyball: the NBA an Big Data”
3. Rajiv C. Shah and Rob Romijnders, “Applying Deep Learning to Basketball Trajectories” 2018
4. Taimur Shahzad, Zahoor Ur Rehman, Eliza Batool and Zain Ullah, “Kobe Braynt Shot Prediction using Machine Learning”
5. Murakami-Moses, Max. "Analysis of Machine Learning Models Predicting Basketball Shot Success."
6. Oughali, Maram Shikh, Mariah Bahloul, and Sahar A. El Rahman. "Analysis of NBA players and shot prediction using random forest and XGBoost models." 2019 International Conference on Computer and Information Sciences (ICCIS). IEEE, 2019.
7. NBA.com. [S.l.: NBA Media Ventures, LLC, 2001] Software, E-Resource. <https://lccn.loc.gov/2004565108>.
8. M. Harmon, P. Lucey and D. Klabjan, "Predicting shot making in basketball learnt from adversarial multiagent trajectories", 2016.
9. Alex Graves and Jurgen Schmidhuber, “Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures”, 2005.

**IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove template text from your paper may result in your paper not being published.**