# Analysis of Machine Learning Models Predicting Basketball Shot Success

Max Murakami-Moses
The American School in Japan; Tokyo, Japan
Email: 22murakamimosesm@asij.ac.jp

**Abstract - The most critical aspect of winning a basketball game is shot selection. However, due to the multitude of factors that come into play when deciding if a shot was a good shot selection (a shot that has a high chance of going in is a good shot selection), it is difficult for a human to make a reasonable assumption. Because of this, we used and analyzed a variety of machine learning techniques to predict shot success.**

**Here, we perform an analysis of the best machine learning models for predicting shot success as well as comparing the performance of these models using different features. Our models (neural network, logistic regression, and gradient boosting) were able to predict shot success between 64.9% - 65.1% accuracy.**

**Our models performed with about equal accuracy; however, when we altered the features, the accuracy significantly decreased. This highlighted the importance of good quality data as well as the importance of certain features, such as the type of shot, in making a shot.**

*Keywords - Deep neural network, machine learning, shot selection, and accuracy.*

## BACKGROUND

Shot selection is vital to winning the game of basketball—good shot selection results in increased points and a better chance at winning. Shot selection is evaluated across many domains, such as the shooter's positions with respect to the basket; positioning of the on-ball defender, rotating defenders, and teammates; minutes remaining on the clock; players skill level; shot type. Though shot selection can be judged based upon observable factors, it is difficult for coaches and players to understand if the shot was good beyond whether or not the ball went in. This is due to the fast pace of the game and the human eye's limited ability to evaluate the multitude of factors that play into a good shot selection.

However, the process of understanding shot selection can be significantly enhanced using machine learning. A model, such as a neural network, can be created and trained to understand what is a good shot selection based upon all of the mentioned factors. This can be of great use for all levels of players as they can learn what shots to pass upon and take, independent of the shot going in. Furthermore, coaches would be able to understand better which players can make the best decisions and give them the ball during crucial moments of the game. Over the last decade, progress in machine learning and access to large amounts of data have allowed data scientists to give sports teams a deeper insight into how to win and how to keep their players healthy. For example, in sports analytics, Machine learning has been applied in the game of soccer. A recent paper in the M edicine and Science in Sports and Exercise outlined how a statistical model was able to predict the likelihood of a hamstring injury [1]. When selecting two players at random, the probability that the player with a higher injury risk would get injured first was 75%. Furthermore, with the model, researchers were able to identify three factors significantly associated with hamstring injury: Seven genetic variants, previous hamstring injuries, and age(with players over 24 being more likely to be injured).

In this paper, we propose building and training a model using a machine learning and data-driven approach to give percentages indicating the shot's success rate. The percentage can be used to analyze whether the player selected a good shot, as a high percentage would signal that it was a good shot and vice-versa. Utilizing a deep neural network, the model could predict if a shot would go in with 65% accuracy.

## RELATED WORKS

In an article titled "NBA Shot Prediction and Analysis," Raymond Cen, Harrison Chase, Carlos Pena-Lobel, and Daniel Silberwasser attempted to predict shot success [2]. Their data set included all of the same features of our dataset. However, an important distinction is that they also included the players' FG% from each zone on the court (Figure [1]).

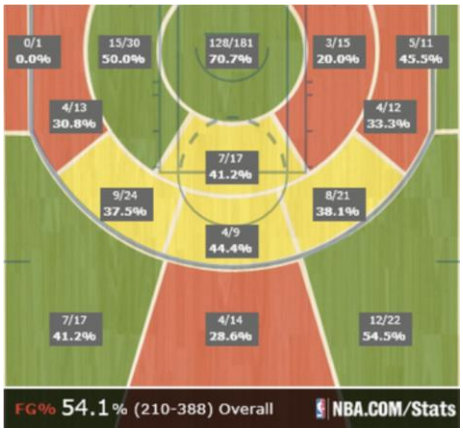



Figure [1]: Image from "NBA Shot Prediction and Analysis."

This gives them a clear advantage in predicting shot success as they have access to the players' FG% in all zones, which allows them to rely on a player's history rather than in-the-moment factors. They utilized a logistic regression model to try to predict shot success. The model was able to achieve 65% accuracy on whether a shot would go in or not. In another paper titled, "Predicting NBA Shots" by Brett Meehan of Stanford University, shot success was able to be predicted between 54%-68% accuracy [3]. Brett used a variety of models, such as Naive Baynes, Random Forest, Boosting, Logistics Regression, and neural network, in an attempt to find the most suitable model for predicting shot success. His dataset came from the NBA via its SportVu tracking system similar to our dataset. However, his dataset differed slightly in features and was significantly larger than our dataset. Though our data set did not include any tracking data, Bretts' data set included features such as the shooters' distance to the basket and the nearest defender. The paper claims to achieve an accuracy of up to 68% using an XGBoosting with a tuning model. However, these results are contradictory to the confusion matrix presented in the same paper (Figure[2]). The max accuracy according to the confusion matrix is 63%, which aligns more closely with results given by other models and in experiments we briefly performed on the same data set.

| | Pred p | Pred n | total |
|---|---|---|---|
| Actual p | 1040 | 1875 | 2915 |
| Actual n | 496 | 2993 | 3489 |
| total | 1536 | 4868 | |

Figure [2]: Confusion matrix for Boosting

## METHOD

*1.Data Set and Features*

The data set this paper utilizes was initially provided by the NBA during the 2015-2016 season. However, due to the NBA's decision to restrict public access to their data, we loaded it from pages that hosted the data on their GitHub/Kaggle [4]. The data set consists of 419 NBA players from all 30 of the NBA's teams. Using cutting edge camera technology from SportVu, the NBA carefully recorded the players' movements and actions to provide data for the league, teams private and companies. In total, the dataset contained 84,466 data points. The mean percentage of a made shot per shot(FG%) across the data set was 44.8%. This means the clear accuracy benchmark for our network would be 55.2% percent, as that would be the result of naively predicting a failed shot attempt for all attempts.

*2.Data Collecting and Processing*

Initially, our dataset contained positional information of the player; however, we did not include it because it did not significantly affect our accuracy. We hypothesize this is due to the data of shot times being inaccurate by a few seconds. Though a few seconds may not seem significant, the positional data of players, even after a second, can dramatically change, m king the data unusable and inaccurate.

Besides positional data, we extracted what we believe to be the 12 most vital data columns in predicting shot success from the original basketball data set.

Four of the columns consisted of categorical data: "ACTION_TYPE" (a column composed of 53 different categorical listings of the type of shot such as dunk, layup, Hookshot, etc.), "PLAYER_NAME," "SHOT_ZONE_AREA," "SHOT_ZONE_BASIC"(a column consisting of 7 individual categorical listings of the location of a shot such as left corner three, in the paint, etc.). We one-hot encoded the categorical data in order to turn the categorical data into numerical data. This allowed for the models to take in only numerical data, which is optimal for performance. The remaining eight columns consisted of numerical data and are described in table 1. Additionally, our data set included a "flag" that noted if the shot was made or not. The main challenge with the data set was a limited amount of features describing an individual's performance. Although there are features that define the player's distance from the basket and XY coordinates on the court, most of the features consisted of high-level features such as "PERIOD" and "EVENT_TIME" which tell us much less about an individual's ability to make a shot. However, as these features could have a small correlation with shot accuracy (later in the game, the player may be more tired and thus may miss more frequently), we decided to include them in our data set (Table [1]).

Table [1]: A table describing each feature used.

| | |
|---|---|
| ACTION_TYPE | Categorical data labeling the type of shot such as layup, three pointing, and hook shot. |
| PLAYER_NAME | Categorical data that gives the players full name. |
| SHOT_ZONE_AREA | Categorical data describing which side of the court the player is on. For example, “Right Side”, “Back Court”, and “Ride Side Center”. |
| SHOT_ZONE_BASIC | Categorical data describing the area on the court relative to the basket. For example, “Mid-Range”, “Restricted Area”, and “In the Paint”. |
| EVENTTIME | Time remaining in the quarter. |
| LOC_X | Numerical data describing the players X coordinate on the court. |
| LOC_Y | Numerical data describing the players Y coordinate on the court. |
| MINUTES_REMAINING | Numerical data describing the minutes remaining in a period. |
| SECONDS_REMAINING | Numerical data describing the seconds remaining in a period. |
| PERIOD | Numerical data denoting which period, or quarter the game is in (1-4). |
| SHOT_TIME | The time of highest acceleration before the ball reaches its peak. |
| SHOT_DISTANCE | Numerical data describing how far the shooter is from the basket when attempting the shot. |

Lastly, we randomly split the data set into training data and validation data . 90% of our data was used as training data while the other 10% was used as validation data. This allowed us to properly evaluate our models as there is no inherent bias when testing our models because it has not had the chance to overfit to the validation model.

## CLASSIFICATION MODELS

In this paper, we utilized three different machine learning techniques and models to predict shot success. Each model predicted shot success within an accuracy of 64.9%-65.1% accuracy.

*1.Deep Neural Network*

A deep neural network is a machine learning technique modeled after the human brain and nervous system. The network can recognize individual rel tionships and/or patterns in sets of data in a process that mimics humans, allowing them to make intelligent predictions or classifications. The network consists of connected nodes that are organized into a certain amount of layers. Each node has a weight and threshold associated with it, which is initially set at random before the network is trained. As the network trains, it adjusts the parameters by passing the inputs into the network, calculating a prediction and comparing this prediction with the true known value. The error, or 'loss', between the true known value and the prediction, is back-propagated through the network using tools of calculus to update the parameters in a way that pushes the prediction towards the correct answer.

The final layer of nodes passes through an activation function, which in our network is a sigmoid function. This function takes the output of the network and produces a number between 0 and 1, which corresponds either to a missed shot or a made shot.

See the sigmoid function below (Figure [3]):

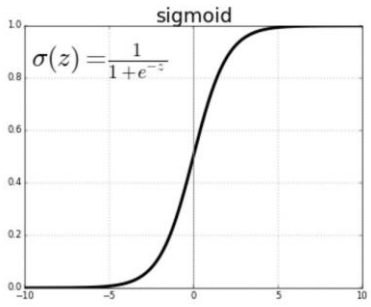



Figure [2]: Sigmoid function and equation.

The sigmoid function is a suitable activation model for a binary classification situation as it restrains the prediction between 0 and 1. Furthermore, it is also a

continuous function as opposed to a Heaviside function. A Heaviside function has an output of a 0 or 1 but has a sudden jump between the two values, making it unusable by a neural network.

The cross-entropy loss gives us the negative log-likelihood of our parameters. Simply put, it is a metric of how likely the true data was to be seen if the current probability predictions we have were true. Minimizing this binary cross-entropy loss (the negative log-likelihood) will ensure that our model assigns a high probability to what truly happened.

See the binary cross-entropy loss equation below:

$$CCE(p,t) = -\sum_{c=1}^{C} t_{o,c} \log\left(p_{o,c}\right)$$

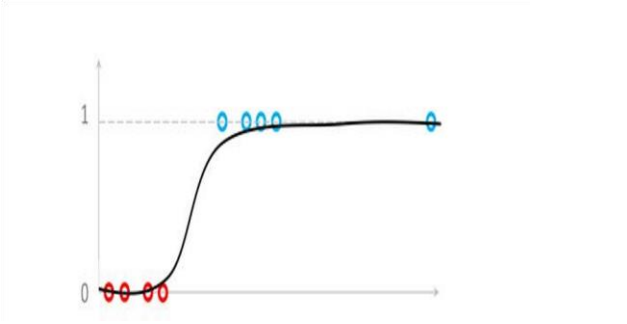
Figure [3]: Cross-entropy loss equation.

*2.Gradient Boosting*

Gradient Boosting is a machine learning technique that uses an ensemble of weak machine learners and combines them to create a strong learner. The most common learner is the decision tree. In this paper, we used Gradient Boosting from scikit-learn.org [5].

*3. Logistic Regression*

In Logistic regression, the dependent variable has a binary outcome (1 = success and 0 = failure), which makes it a good fit for this situation. Through a mathematical formula similar to the sigmoid function, it is able to classify data based upon multiple features. Below is a two feature representation of logistical regression, with the blue dots representing a positive classification and the red dots representing a negative classification.




Figure[4]: Illustration of a two feature logistic regression model. The X-axis presents a certain feature while the Y-axis presents if a dot has any correlation with that feature. The blue dots represent a positive result, while the red dots represents a negative result.

**Experiments and Results**

Our neural network consisted of 2 layers with each layer consisting of 20 nodes each. The activation used for each layer was a relu function; however, the last layer utilized a sigmoid activation function. The network had a learning rate with RMSprop of 0.001, a weight decay of 0.0005 and was tr ined for 10 epochs. Our six-layer neural network predicted shot success with an accuracy of 65%. To our knowledge, no other works have achieved a better accuracy than 65% on this or similar datasets. Given the highly stochastic nature of basketball shot attempts, 65% accuracy can be considered a well-performing model.

Furthermore, our experiments with gradient boosting and logistic regression models produced similar results. Our gradient boosting model was able to achieve 65.1% accuracy. Our logistic regression model was able to hit 64.9% accuracy [Table 2].

Table [2]: Table containing experiment results.

| | Accuracy | F1 Score |
|---|---|---|
| Neural Network | 65% | 0.65 |
| Logistic Regression | 64.9% | 0.72 |
| Gradient Boosting | 65.1% | 0.71 |

We also attempted to decipher which factors were the most important to the network to predict shot success. Removing player names from the dataset still allowed the network to have an accuracy of 65% [Table 3]. This shows that the network treated the “PLAYER_NAME” factor as almost obsolete.

However, it is well known that different players vary hugely in their ability to make a shot, and thus it would be intuitive that the player name would be an essential factor. We propose using a larger data set, so the network has more time to understand the player’s history of making - or not-making - shots for the network to truly take into account the player’s name.

Furthermore, removing the “SHOT_DISTANCE” factor resulted in a 64% accuracy [Table 3]. This was because the network could easily compensate for the “SHOT_DISTANCE” with the XY coordinates factor. The most crucial factor for the network was “ACTION_TYPE”. When removing that factor, the network performed at 62% accuracy [Table 3]. This finding suggests that players and coaches should focus on the category of shots they take (ACTION_TYPE) to dramatically increase the shot success.

Table [3]: Table containing accuracy when certain features are removed.

| | Accuracy |
|---|---|
| Original Features and Original Network Shape | 65% |
| Removing "SHOT_DISTANCE" and Original Network Shape | 64% |
| Removing "ACTION_TYPE" and Original Network Shape | 62% |
| Removing "PLAYER_NAME" and Original Network Shape | 65% |

We have included a link to our code in the references section.

## APPLICATION AND CONCLUSION

Overall, our networks were able to predict shot success with an accuracy between 64.9%-65.1%. Also, the neural network allowed us to understand that shot type(dunk, layup, hook shot) was the most significant factor in predicting shot success.

Our networks allow coaches to understand which players are taking the best quality shots. This has real game results as coaches can then give the ball to the player with the best shot selection during crucial moments of the game. This translates to a higher percentage of the shots going in, thus increasing the chances of winning.

For future avenues of further investigation, we suggest for researchers to use data that spans over entire seasons and possibly multiple seasons. The increased data points should hypothetically increase shot success rate prediction accuracy as the models will be able to take into account player names. Furthermore, the increased data points will allow the network to train for longer and further improve its accuracy.

## ACKNOWLEDGMENTS

I would like to thank James Borg for mentoring me in this research paper as well as for proof-reading this paper.

## REFERENCES

[1]: Larruskain, Jon, et al. "Genetic Variants and Hamstring Injury in Soccer: An Association and Validation Study." Medicine and Science in Sports and Exercise, U.S. National Library of Medicine, 2018, pubmed.ncbi.nlm.nih.gov/28976491/.

[2]: Cen, Raymond, et al. "NBA Shot Prediction and Analysis." NBA Shot Prediction and Analysis by hwchase17, hwchase17.github.io/sportvu/.

[3]: Meehan, Brett. Predicting NBA Shots, 2017, cs229.stanford.edu/proj2017/final-reports/5132133.pdf.

[4]: sealneaward(2018)nba-movement-data [Source Code] https://github.com/sealneaward/nba-movement-data

[5]:"3.2.4.3.5.Sklearn.ensemble.GradientBoostingClassifier ."Scikit ,scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html.

Figure [1]: Cen, Raymond, et al. "NBA Shot Prediction and Analysis." NBA Shot Prediction and Analysis by hwchase17, hwchase17.github.io/sportvu/.

Figure [2]: Neuronactivator.com. "What Even Is an Activation Function?" The Neuron Activator - Vaibhavsinh Vaghela, 13 Dec. 2019, www.neuronactivator.com/blog/what-even-is-activation-fu nction.

Figure[3]: Chris. "How to Use Binary & Categorical Crossentropy with Keras?" MachineCurve , 22 Oct. 2019, www.machinecurve.com/index.php/2019/10/22/how-to-use - binary-categorical-crossentropy-with-keras/.

Figure[4]: Narkhede, Sarang. "Understanding Logistic Regression." Medium , Towards Data Science, 26 May 2019, towardsdatascience.com/understanding-logistic-regression9b02c2aec102.

The code used throughout this paper can be found here: https://colab.research.google.com/drive/1Y1jup_Ub2KFUvrnYJCmke89nrmI7wngg?usp=sharing