Classification Models for Predicting NBA Shot Success

Rauf Iftikhar, Alan Liang

December 2022

1 Abstract

Here we present an investigation into the efficacy of 4 supervised learning classification models for the binary classification of NBA shots from the 2014-15 season. We tested Random Forest, Logistic Regression, KNN, and Naïve Bayes models, achieving about 60% accuracy on all models. Moreover, we identified the most relevant features of our models, which can provide useful strategic insights for basketball teams.

2 Introduction

Basketball shot quality is a valuable but difficult to quantify metric. Effective methods of evaluating the quality of a shot, i.e., its likelihood of being successful, can help teams develop effective strategies that improve the quality of shots and consequently improve overall success on offense. However, it is difficult to untangle what variables influence shot outcome the most, as well as how significant the role of luck is. Ideally, we could quantify every aspect of a shot down to minutiae. This would consist of data such as the shooter's body angle, release angle, and the quality of the defense being played. While we do not yet have metrics that capture such information, we do have quantitative metrics that start to grasp aspects of these abstract variables. For example, we can, to a certain extent, quantify the abstract 'defense' with distance of the closest defender. Here, we propose a supervised learning approach to a classification problem of predicting whether individual shots will be successful. Using available features, we can develop models that learn the important features of a shot. Our data is likely not robust enough yet to be used to reliably predict shot outcomes (for betting purposes, for example), but we can observe the most relevant features of our models to gain insights into the features that contribute most to shot quality. We combined multiple datasets to curate a new dataset of features as inputs for our classification models. Our features included location (home/away), game result (win or loss), final margin of victory/defeat, shot

number, time on the shot clock, number of dribbles taken, time spent possessing the ball, shot distance, shot type, name distance of closest defender, number of shots attempted per game, percent of successful shots, percent of successful three point shots, offensive rating of shooter (team points per 100 possessions with player on the floor), and defensive rating of defender (opponent team points per 100 possessions with player on the floor). In this paper, we will discuss four different ML algorithms that can be used to predict the outcome of a basketball shot: K Nearest Neighbors (KNN), Naive Bayes, Random Forest, and Decision Trees.

3 Related Work

There has been significant research into predicting basketball shots, whether through visual-spatial data or player statistics. "Analysis of Machine Learning Models Predicting Basketball Shot Success" discusses how advances in basketball analytics have been enabled by player tracking data and the application of machine learning techniques to process this data [1]. Work has also been done on other basketball classification problems, such as playcall classification from player tracking data [3]. This is a key component in understanding a team's strategies and interactions between players. Through the use of pictorial representation of player position sequences, neural networks and recurrent neural networks, the authors are able to achieve high precision and accuracy in classifying the playcalls. The authors also show that their system is able to achieve good recognition rates when trained on one season and tested on the next, demonstrating the transferability of their method across seasons. We later use some of these techniques including the specific data collected such as offensive rating and defensive rating along with some of the algorithms presented.

"Predicting Shot Making in Basketball Learnt from Adversarial Multiagent Trajectories", is about predicting the likelihood of a player making a shot in basketball from multiagent trajectories [2]. A convolutional neural network (CNN) approach is presented, where the multiagent behavior is represented as an image. To capture the adversarial nature of basketball, a multichannel image is used and fed into the CNN. Additionally, to capture the temporal aspect of the trajectories, a "fading" technique is used. Results show that this approach is superior to a traditional FFN model, and that a combined FFN+CNN is the best performing network with an error rate of 39 percent. This paper provides insight into the nature of shot positions and the importance of certain features in predicting whether a shot will result in a basket. Overall it helps us understand the different approaches we could take in order to boost our current model. Although we looking into SortVu modelling and attempted to use a CNN, we ultimately decided against it due to continued poor performance and the additional complexities of visual data.

4 Methods

4.1 Preprocessing

4.1.1 Exploratory Data Analysis

Our exploratory data analysis consisted of loading our main dataset and viewing the features so as to decide how to manipulate the data. We also looked at the distribution of our response variable, FGM (whether or not the shot was made), to see if our dataset was balanced. The main dataset we used was a Kaggle matrix with 21 variables for each of over 120,000 shots from the 2014-15 NBA season. These columns included numerical spatial shot tracking data such as shot distance and distance to closest defender. There were also categorical data such as shooting player id, defender player id, whether the shooting player was on the home or away team, and what quarter of the game the shot was taken in.

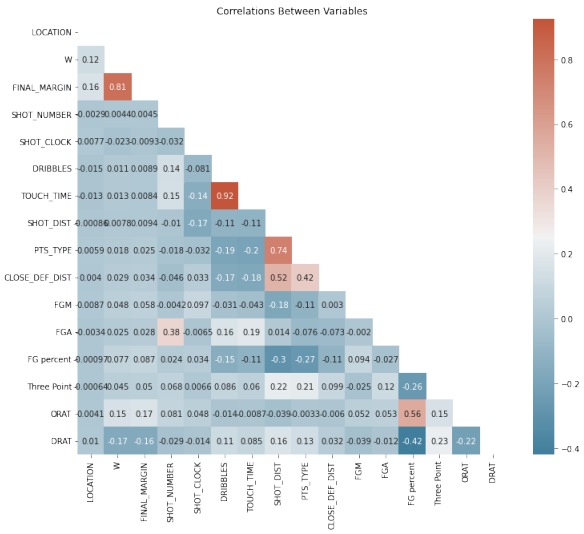
4.1.2 Feature Extraction

We felt that this dataset crucially lacked variables that attempted to quantify player skill. Thus we used player ids as look-up indexes to a second dataset with player averages for the total season. Thus, for every shot taken, we added the shooting player's average shot attempts per game, shooting efficiency, offensive rating, and the defending player's defensive rating as features of the shot. After this feature extraction, we began preprocessing the data by removing null valued observations. We next handled categorical features by removing them or converting them to numerical features. We removed a number of categorical variables that did not provide relevant data. These included date, game ID, player IDs, quarter of the game, player names. Other categorical variables we converted to numerical features. For example, we converted home/away, 3 point/2 point shot, and win/loss strings into binary encoded features. After preprocessing, we were left with 16 numerical features, 1 binary response variable, and 115,000 observations.

4.1.3 Feature Selection

We proceeded to further reduce this dataset by performing feature selection on our variables. We used Pearson correlation to measure correlations between each feature.

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$



Assuming highly correlated features would only provide redundant information to our models, we dropped features with high correlations. We dropped Win/Loss, 3 point/2 point shot type, and number of dribbles taken. These features were redundant as they were latent in other variables. In the final margin, the sign of the margin signifies whether the game was won or lost. The distance of the shot highly correlates with what kind of shot is being taken, and the number of dribbles taken is related to the touch time, because players tend to dribble while possessing the ball rather than hold still. Thus, we confidently assumed that we did not lose much relevant information in our feature selection.

4.1.4 Normalization and Dimension Reduction

Since one of the models we implemented (KNN) calculates distance between observations, we scaled our data using a Standard scaler to evenly distribute the data across features. We then performed PCA to further reduce our data. However, PCA revealed that our features were not highly correlated and did not explain the same variability in the data. It required 12 principle components to explain 95% of the variability in the data. Thus, we did not reduce the dimensions in our data.

4.2 Model Selection

4.2.1 KNN

K Nearest Neighbors (KNN) is a supervised ML algorithm that classifies a data point by looking at its k closest neighbors. Closest neighbors are determined by computing a distance metric for every features and selecting the observations from training with the smallest overall distance.

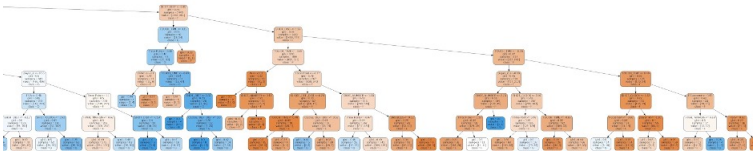
4.2.2 Naïve Bayes

Naive Bayes is a supervised ML algorithm that estimates the probability of an event given the outcome of previous events. The algorithm is based on Bayes Theorem and assumes that all features are independent of each other.

$$P(\theta|\mathbf{B}) = P(\theta)\frac{P(\mathbf{B}|\theta)}{P(\mathbf{B})} \tag{1}$$

4.2.3 Random Forest

Random Forest is an ensemble ML algorithm that uses multiple decision trees to make a prediction. The prediction is based on the majority vote of the individual trees in the forest.



4.2.4 Logistic Regression

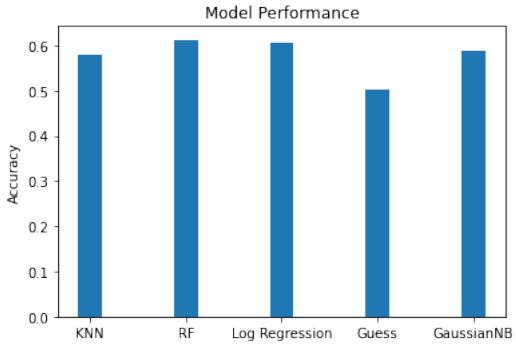
Logistic regression is a classification model that fits a sigmoid function to estimate the probability of a class.

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

When predicting observations with logistic regression, we assign any sample point with greater than 0.5 probability to the 1 class, and the rest to 0 class.

5 Results

5.1 Accuracy



5.2 Confusion Matrices

5.2.1 Random Forest

		Shot Prediction		
		Made	Missed	Total
Shot Outcome	Made	16116	2903	19019
	Missed	10568	5197	15765
	Total	26684	8100	34784

5.2.2 Logistic Regression

		Shot Prediction		
		Made	Missed	Total
Shot Outcome	Made	13840	5179	19019
	Missed	8533	7232	15765
	Total	22373	12411	34784

5.2.3 Naïve Bayes

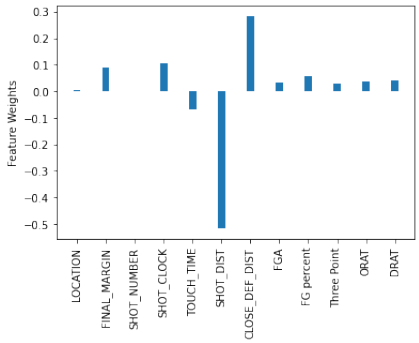
		Shot Prediction		
		Made	Missed	Total
Shot Outcome	Made	13748	5271	19019
	Missed	9066	6699	15765
	Total	22814	11970	34784

5.2.4 KNN

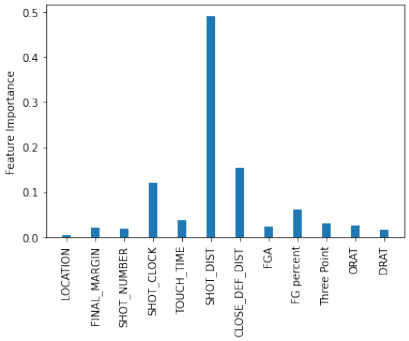
		Shot Prediction		
		Made	Missed	Total
Shot Outcome	Made	13639	5380	19019
	Missed	9213	6552	15765
	Total	22852	11932	34784

5.3 Feature Importances

5.3.1 Logistic Regression Weights



5.3.2 Random Forest Feature Importances



6 Discussion

Our best model, the random forest, achieves 61% accuracy, which is slightly worse than related work has achieved using similar methods. Logistic regression performed nearly as well at 60.5% accuracy. Thus, to answer our study question of which features are more important, we extracted and plotted the highest weighted variables for each model, and found the same 3 variables to be most significant: shot distance, closest defender distance, and shot clock. Of the features we added to the original Kaggle dataset, field goal percentage had the highest importance, which intuitively makes sense as it measures essentially the same thing as our response variable.

Our confusion matrices gave us some interesting insights into the differences between the models. Since the probability of a made or missed shot is about equal, we would perhaps expect to see even distribution of false positives and false negatives. However, there was a larger number of false positives than false negatives on the test data for each model. This was most apparent in the random forest, which had a massive disparity of 10568:2903 ratio of false positives to false negatives. Each model overestimates the number of makes relative to misses. These findings may indicate that the individual variability of shots is difficult to predict, and a more appropriate question may be how many shots out of a large sample we expect to be made/missed.

7 Conclusion

We implemented 4 machine learning classification models to predict the outcome of basketball shots. All 4 of our models achieved similar accuracy (around 60%), but differed on the types of errors made. Our best model was the random forest model, which is consistent with related literature that shows ensemble methods to be effective for predicting NBA shot outcomes. In extracting the important features from our models, we identified the most relevant predictors of shot success from the available data. In the future, we hope to get access to

datasets that are currently restricted by the NBA (such as SportVU) or are only accessible behind a paywall (such as ShotQuality.com data). These datasets will give us more robust spatial tracking data, which appears to be the most relevant type of information that we can collect.

8 Contributions

Code and csv files can be found at the following google drive folder:

https://drive.google.com/drive/folders/15dVokVwku5ivy8I044i2e6hCvaWeHmMq?usp=sharing

Both authors contributed to writing the final report. Rauf wrote the shot_logs.py script to handle the preprocessing, and both authors contributed to the python notebook that contains the model implementations and plot generation.

References

[1] "Analysis of Machine Learning Models Predicting Basketball Shot Success", Max Murakami-Moses, The American School in Japan; Tokyo, Japan

[2] "Predicting Shot Making in Basketball Learnt from Adversarial Multiagent Trajectories", Harmon et al. Northwestern University, January 2021

[3] "Classifying NBA Offensive Plays Using Neural Networks", Kuan-Chieh Wang, Richard Zemel,University of Toronto, Toronto, Ontario, Canada. March 2016