Analysis of the Effects of Positional Features on NBA Shot Efficiency

Rahul Bhatt, Rohan Suresh, Shloak Jain, Aditya Bollam, Lovish Murjal
Sports Analytics Group at Berkeley (SAGB)
University of California, Berkeley

ABSTRACT

We present a scoring efficiency classifier for the NBA. Our model takes in player movement data from NBA games and uses it to predict makes and misses based on various features involving player movement in each offensive possession. Our methodology looks at player positions during attempted shots as well as other features such as the movement of players on both teams for each possession. We look at various movement features in the context of a possession which we define as from when the offensive team brings the ball past the half court line to when a shot is made or defensive rebound is recorded. Using logistic regression, we train our classifier to predict makes and misses based on a series of different movement features in each possession. The reason we decided to use logistic regression as our model was because the main point of our research was to determine which features mattered the most in predicting shot efficiency. If we had used another type of model, like a neural net, that information would have been lost.

After creating the logistic regression classifier we assessed the accuracy of the model and iteratively improved it with the addition of other player tracking features. Once we had an accurate model for predicting scoring efficiency based on the various player tracking features we consider, we then observed the coefficients of each of our features. This allows us to quantitatively determine which features correspond to the greatest increase in scoring efficiency. In order to control for biases in the data we analyzed a single team at a time to determine the most powerful features in our model. This allowed us to account for the raw offensive ability of a team.

After conducting our analysis, we found the following results for our features. The feature most correlated with shot efficiency improvement was number of players outside the arc, which had a weight of 0.045. A feature we found that had little bearing on shot efficiency was distance to the closest teammate from the shooter, which had a weight of -0.0041. The precision, recall, and f1-score of our classifier was 0.31, 0.56, and 0.40 respectively.

BACKGROUND

We explored a set of games in the 2015-16 NBA regular season involving the Portland Trail Blazers and used player tracking data to build a shot making classifier based on different

features. We chose the Portland Trail Blazers because we felt that their roster and play style in 2015-16 was an accurate representation of a modern NBA roster with high production from their guards and their perimeter style pick and roll offense. The game data was presented in 2 files: a game file and an events file. The game file contained a large table of the player and ball positions sampled every 0.04 seconds. The events file contained information about each play that occurred in the game, each characterized by an event type. We focused on event types 0 and 1, made and missed field goals, in order to build our classifier. Rather than trying to expand the classifier to all NBA teams, we decided to focus on one (Portland) so that we could use features that were personalized to the particular team and its style. After cleaning our data set and getting it into usable form, we then developed a set of utility functions such as finding shot times, player locations at the time of a shot, and distances traveled by players. We could use these utilities to build out features and to build relevant visualizations. After deciding on our features, we used our utilities to generate our training data and test data to analyze our classifier's performance.

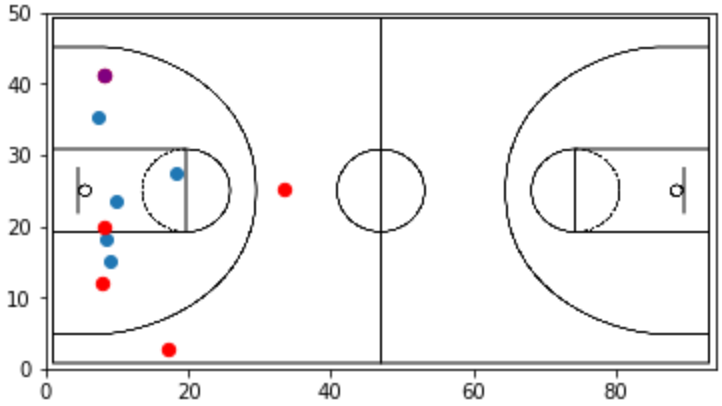

Figure 1

The above figure shows the locations of all 10 players at the time of a shot. The player in purple is the shooter, and his 4 red teammates are the Portland off-ball players. The 5 blue players are the Memphis players on defense.

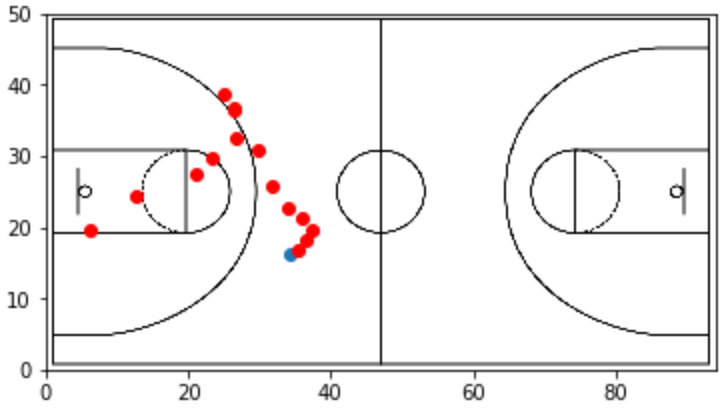

Figure 2a

The above figure shows the tracking of a single player over a possession. The player's starting point is the blue circle. This movement is broken down in Figure 2b and 2c.

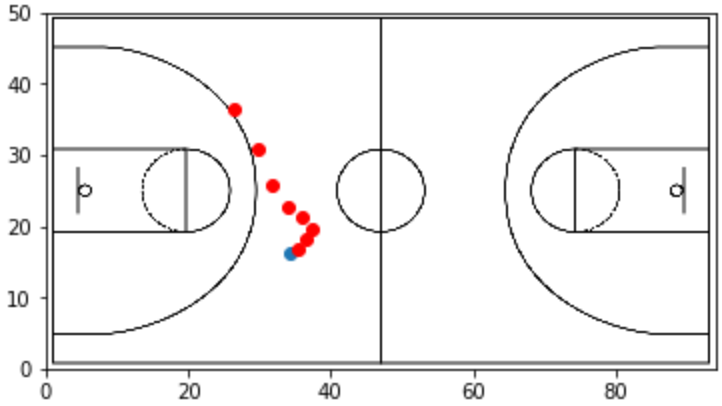

Figure 2b

The above figure shows the first part of the player's movement in the possession (starting at the blue circle) as he is moving along the perimeter to the right wing.

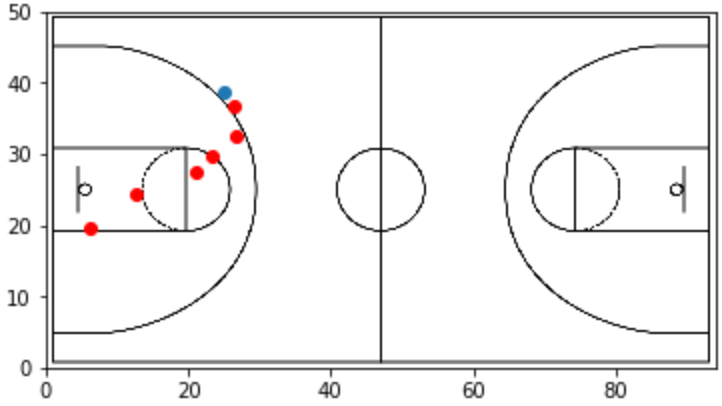

Figure 2c

The above figure shows the second part of the player's movement in Figure 2a. After the player moves to the right wing in figure 2b, he crosses back over to the left and cuts toward the basket.

Figures 2a-2c show how fine tuned the player tracking data is. This level of detail allowed us to generate very descriptive features.

TECHNOLOGIES

All of the coding for this project was done using python. We relied on a few core python libraries including numpy, pandas, and scikit-learn. The player location data was presented in very large csv files that gave the x, y position of all players on the court in 0.04 second intervals. Thus, to analyze features we had to write helper functions that would transform the raw data into features that we would analyze later on with the help of logistic regression. We also needed to scrape events files, which were play-by-play files and cross reference these files with the player position tracking files.

After we finished all the scraping, we wrote out our features and whether the associated shot at the time went in or not to a csv file. This was then split into a training and test data set. We used scikit-learn's LogisticRegression package to actually create the classifier and analyze its accuracy.

STATISTICAL TECHNIQUES

To conduct our analysis we used logistic regression to determine what features were correlated with increased shot efficiency. Logistic regression is a discriminative classifier, which means it works by maximizing a probability rather than generating a joint distribution. In our case, we were trying to maximize the probability of our observed samples, which included features related to the positions of players and an output class representing if the shot went in or not.

We obtained our samples by gathering positional data about the players at the time the shot was released by the shooter. After doing so, we split our samples into a training and test set, with the test set being 30% of all the samples. Finally, we looked at the coefficients on each feature as well as the precision and recall of our classifier. A coefficient with a high absolute value would mean that feature has a large effect on shot efficiency. For our classifier, the feature with the highest coefficient was the number of players outside the arc.

One thing we did do was we chose features that had minimal overlap with each other. This was done intentionally so that the features would be close to independent and thus we could truly see the effect one feature had on shot efficiency.

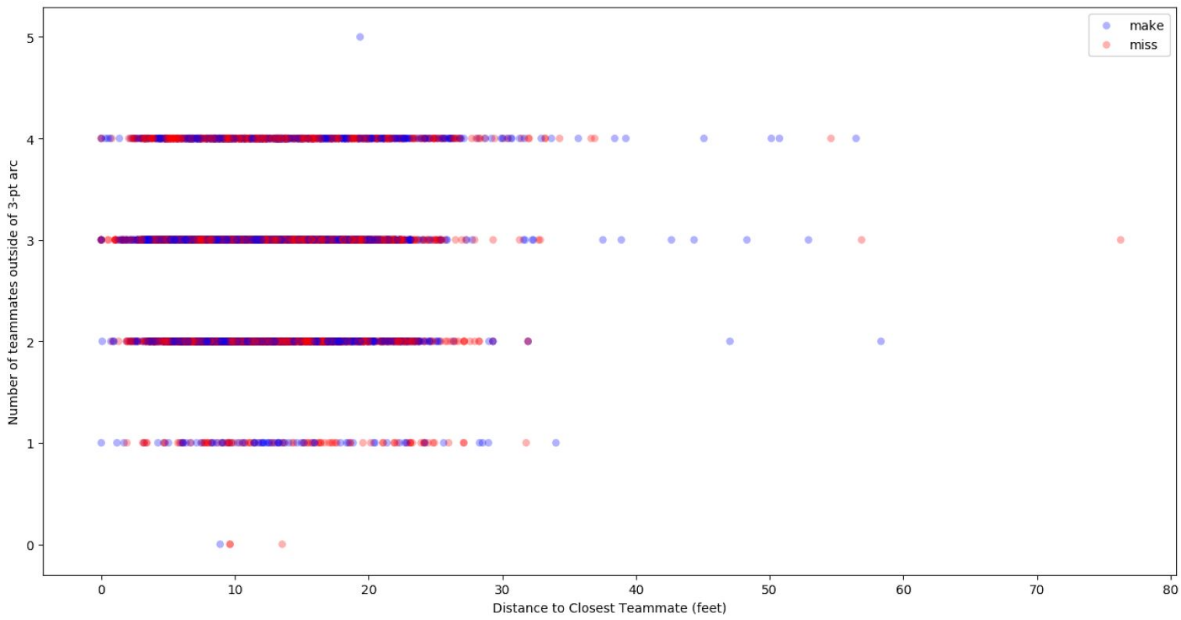

Figure 3

The scatterplot above visualizes all of the shot attempts taken by the Portland Trail Blazers. The x and y axis contain our 2 features (distance to closest teammate and number of teammates outside of 3pt-arc) and the color of the point represents a made or missed shot. Our model was trained using the data above; thus, giving us the ability to extract the relative importance of each feature.

CONCLUSION

After creating our model and analyzing its components and accuracy, we found that although our features were not very strongly correlated with shot efficiency, we did see that the number of players outside of the 3-point arc had the greatest influence on shot efficiency. From a spacing sense this makes sense since in today's NBA, it is nearly impossible to play zone defense. Thus, spreading out the offense will draw out defenders and give the shooter a higher chance of scoring if he can get past his defender. Additionally, we saw that distance to the closest teammate from the shooter had far less of an impact on shooting efficiency. This seems to suggest that other factors are more important than how close the shooter's closest teammate is. Finally, we found that our classifier had an f1-score of 0.40. F1-score is simply the harmonic mean of precision and recall. Although this f1-score is not as high as the f1-scores of classifiers used in other areas such as spam detection, we believe it is a relatively high score for a sport with as many variables as basketball. The fact that we were able to have some success classifying shots based solely on

player position data shows that player tracking data appears to have the potential to refine today's NBA offenses.