Sebastian Crowell

Comp 562

Final Project

Does height change success of basketball shots?

Abstract:

The project examines a vast data set of basketball statistics from NBA players, creates a subset of the players and data, and uses classification models (gradient boosted decision trees) to predict the success of shots made from different x, y coordinates in the court to see how height of players affects the shot being made. Boosting is the choice because of the limited computing of a laptop with many high level features (after testing quadratic kernels for days), basically meaning the project has limited features.

Motivation:

A primary reason for the choice of NBA shots is that sports statistics are plentiful and easy to obtain as sites make them widely available. At first, the data consisted of millions of pieces of information, 18 features and hundreds of thousands of players, because it spanned from 2000 to 2018, CSV [1]. The first step was to condense this so that the computer could open the data. The data was narrowed onto 2017-2018 players and the features limited to 4, CSV [2].

The features used in the vectors fed to the machine learning portion of the project are comprised of: (1) position x, (2) position y, (3) outcome of shot, (4) height of player. Once the vectors are fed in and the data has been processed the boosted trees use a baseline to determine if a location on the court is a successful one and apply a value of 1 if a successful location or a 0 is unsuccessful. The cutoff for a successful location has to be at least 60% of shots made from that x, y position, determined from the Boolean outcome of a shot. This obviously makes a lot of assumptions about a player and the situation but is useful for a baseline of good shooting effectiveness for the height of the player.

Related Work:

A basketball court tends toward a quadratic kernel of player shots being made for 3 point shots, but personally not watching basketball caused a neglecting idea of how often a 3 point shot is attempted compared to a 2 point shot and a shot made under the hoop. It also turned out that the data is only interesting when considering missed shots as

well as shots made, because a lot of shots are missed at the 3 point line and under the hoop because many are made in those areas, Chang et al [1]. This changed to a Gaussian kernel, Silva [2] based on what we had studied in class, which was indecipherable despite averaging portions of the graph to try and create lower percentages figure 1. The problem up to this point was unnoticeable but relevant, there are particular spots where many players shoot consistently due to rules and distance, one such place is the penalty shot area which has a significant amount of shots and misses. This causes a Mt. Everest sized peak in the graph and when averaged, a sea of equally likely shots as well as a load time of 3 hours. Having little real machine learning practice, I sought out a common solution and found gradient boosting with trees, XGBoost [1] from among the kaggle challenges Castilla [3] & Meehan [4], as a way to bypass at the very least my time problem and possibly the issue of areas like the penalty shot area.

Approach & Result:

From here, there was a clear path forward. First the data needed to be crunched to remove many of the non-numerical values that served little purpose. Next, the chance of a shot needed to have an average value to compare to so that the prediction from XGB could have real value, and so that XGB could identify what features were affecting it figure 2(besides the obvious shots made and missed). Once these were combined a more clear image of the shots was by figures 3 & 4 (for figure 4 the boxes are 50 by 100), and it was possible to explore the features x and y to see how the distance in each direction affected the chance of success figure 5 & 6. The average shot success from the total missed and total sunk was 44.95% (rounded). That number represents a sub ~50% to guess if a shot will go in or not at any spot on the court, which we intend to do better than using XGBoost, considering it is worse than a coin flip. Defining our training and data with 47729 pieces of information, the cutoff for a successful spot on the court to shot from was set to .6 or 60% success. This was done to eliminate any continuous variables from multiple shots happening at the same location. The confusion matrix, Math [1], produced with the precision score of .415 or 41.50% meant that the prediction of the algorithm needed optimization first in order to improve the quality of the prediction. The parameters used were $\alpha = .001$, max depth 4, 3 estimators and a child weight of .0001, resulting in a .5150 or 51.50% prediction, a massive increase of 10% confidence in predictions using these parameters. With this new prediction, the height feature was applied to the data set to sort between players > 200 cm and < 200 cm using a larger set of players than the training set, with a final step to average the player shots in each spot to create a similar image as what we started with, just for different heights. This allows a conclusion that the height of a player affects when they take a shot but not the outcome of

a shot, showing that most players that are shorter shoot from further away while taller players get closer to the hoop figure 7 & figure 8.

References (CSV, People, Sites, Images, Math):

CSV:

[1] NBA_Shots_2000_to_2018.csv

[2] fullvariable-basketball-QueryResult.csv

People:

[1] Y. Chang, R. Maheswaran, J. Su, S. Kwok, T. Levy, A. Wexler,

and K. Squire. MIT Sloan Sports Analytics Conference, 2014.

[2] J. Silva, Lecture notes chapter 17-21

[3] P. Castillo, https://www.kaggle.com/, 2016.

[4] B. Meehan, https://github.com/BrettMeehan/CS229_Final_Project, 2017.

Sites:

[1] = https://github.com/dmlc/xgboost

[2] = https://github.com/SebastianCrowell/MachineLearning562FinalProject

Images:

Figure 1, Overall_18.png

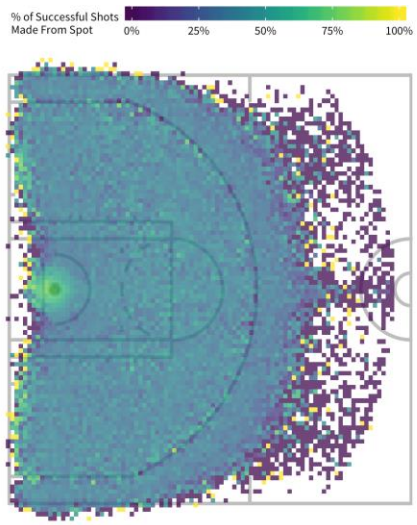


Figure 2, feature.png

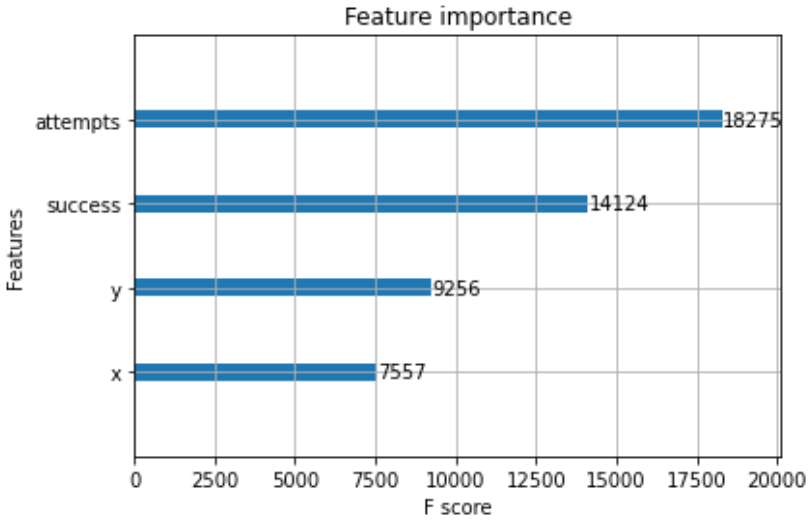


Figure 3, Overall_D.png

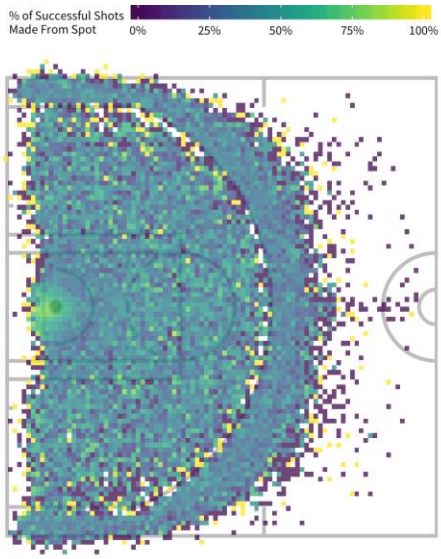


Figure 4, show_miss_cluster.png

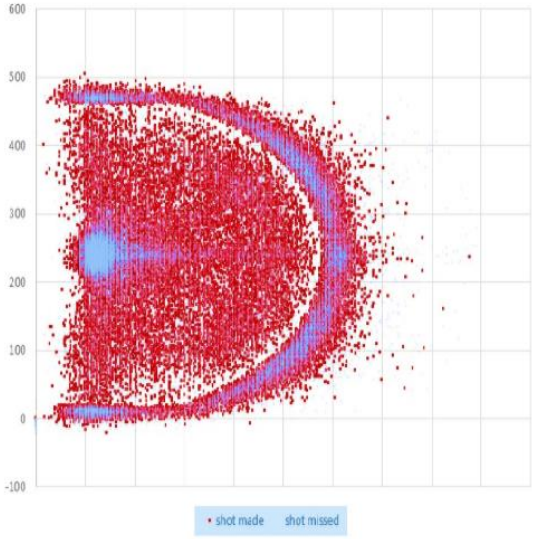


Figure 5, x_explore.png

Figure 6, y_explore.png

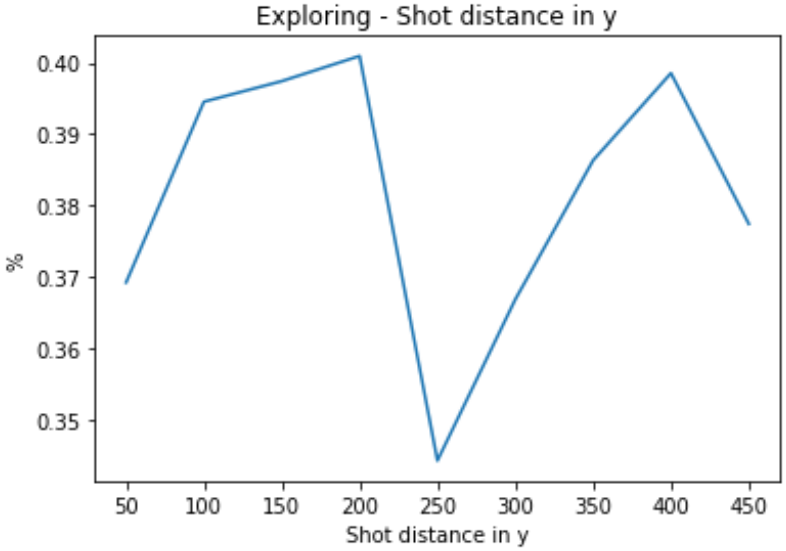


Figure 7, Shorter_17-18.png

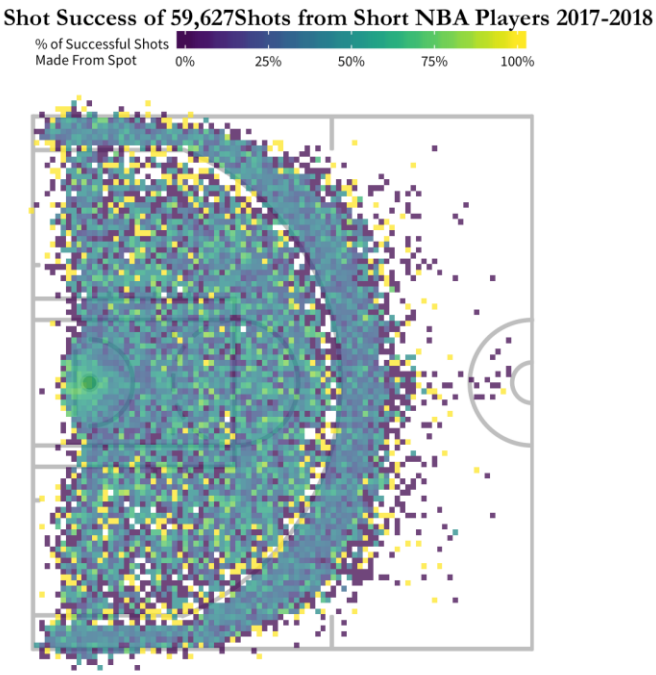
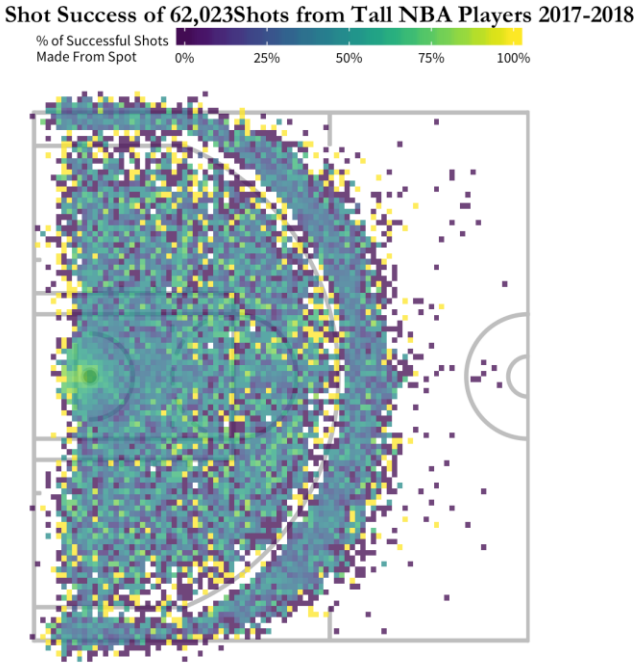


Figure 8, Taller_17-18.png



Math:

Confusion Matrix [1], [[16159 468]

[6906 332]]