In this project, we want to create a shot probability model for the average NBA player to provide context to which players are better or worse shooters than average. The minimum baseline accuracy would naturally by 0.5, achievable by random guessing, as a shot can either miss or make. It is obviously hard to achieve 90% accuracy just because of all the different and complex factors that go into a shot we either don't have the data for or can't necessarily measure yet. So, there is a lot of value in evaluating the models for not just accuracy, but also with RMSE and R^2, while also picking out the most important features. We would split the data into a training and testing sample (80%-20% ratio) and calculate the RMSE for each. Had I had more time, it would also make sense to do cross validation. As aforementioned, a shot can either miss or make, and since we are predicting this, the target metric to use would be OUTCOME from the database table. The predictions will be 0 and 1 from a classification model, but we can also get the probability, where > 0.5 would be 1, and < 0.5 is 0. Of course, that is why calibration matters, as a prediction of 0.55 versus 0.95 on a made basket would both result in us predicting a make, but obviously we would feel better about the latter. To check calibration, we can calculate the Brier score, as well as plot the calibration curve. The features I decided to train on are listed in the README and are mainly quantifying the situation or state of the court leading up to the shot. Because we want the model to be player-agnostic, it didn't make sense to include player and team effects (i.e. shooter, passer, defender, team IDs). Also shot characteristics like RIMDEPTH or ARCANGLE would be more useful for evaluating something like shot consistency (or making a target metric that is less outcome-based and more process-oriented), but are less relevant for quantifying shot difficulty.

To clean some of the data, I took out shot clock values that were greater than 24. Without video, it was hard to tell why some of the null shot clock values were occurring, so I also removed those rows. I decided to use only shots where a foul didn't occur, or a foul did occur, but the shot was made, as only these are considered field goals attempted. Including them would've biased the predictions because fouls usually result in no chance of the shot being made. I also added a couple features – SHOTNUMBER, TOTALGAMECLOCK, and a pairwise interaction between DISTANCE and CLOSESTDEFDIST (definitions are in the README). I wanted to start with a simple logistic regression to get a baseline and get an idea for which features are the most impactful. I added a L2 regularization penalty term to avoid overfitting. The benefit to sticking with a linear model is that there is a lot more feature interpretability compared to other machine learning techniques, but I wanted to see if we could do better with a random forest model. I used a grid search to tune hyperparameters and prevent overfitting. A random forest should be able to handle interactions better and the high dimensionality of the dataset.

The grid search found that the ideal hyperparameters were 200 trees and 5 max depth. The accuracy of the random forest model was not that much better than the regression (0.62 vs 0.61). However, the random forest does have a slightly lower RMSE (~0.478 vs ~0.482), as well as better calibration and better R^2 (0.08 vs 0.066). The R^2 in general is quite low, and it seems like we might be missing some features that could increase it. I think the Brier score for the random forest is fairly low and satisfactory, but the calibration plot still shows that the higher predicted probabilities do not correspond as well to the actual make probability. The feature importances can be found in the README.