NBA Shot Result Prediction

Alex Dai

***Abstract*— This report will detail the usage of statistical learning methods to predict whether a basketball shot is made or missed in the 2014-2015 NBA season. We will attempt to predict the result based on information about the game scenario at the time of the shot.**

I. INTRODUCTION

The usage of data analysis to influence decisions in basketball analytics has increased in popularity exponentially over the past few years. This trend is due to two main reasons. First being the increased hardware process in handling big data tasks. The second is the advent of advanced play-by-play detection technology facilitated by a camera system hung from the rafters collects data on the movements of each player. Now that teams are able to gather a plethora of detailed and descriptive data, they are shifting to relying more on statistical and quantitative measures to influence game time decisions.

Obviously, the objective of basketball is to score more points than your opponent. The point of this analysis is to determine whether we can build a model that can predict whether a shot will fall or not based on advanced data gathered by the SportsVU camera system mentioned above. A model that can do this provides invaluable information to the coaching staff of a team about how the conditions of the game at the time of the shot influences the outcome of the shot.

II. DATA

A. Data Source

The data we will be looking at shot log data gathered from the SportsVU camera systems for every game spanning a majority of the 2014-2015 NBA season, from Oct 2014-Mar 2015. The data was taken from `https://www.kaggle.com/dansbecker/nba-shot-logs/home`, which compiled this dataset by using the NBA API. The reason we are looking at data from the 2014-2015 season is that the NBA API stopped providing this data permanently after the 2015 season due to cost concerns.

B. Data Characteristics

The Shot Log dataset contains information about every single shot taken in 904 games spanning the aforementioned time period, totaling 128,069 rows with 21 features in each row. Figure 1 shows the features available, along with their corresponding data types.

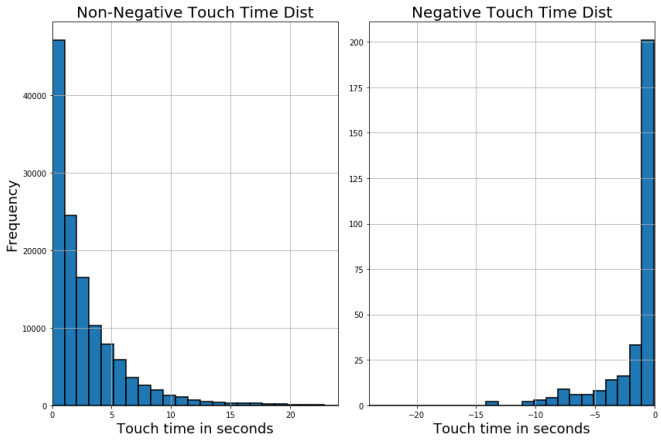
```
Data columns (total 21 columns):
GAME_ID                     128069 non-null int64
MATCHUP                     128069 non-null object
LOCATION                    128069 non-null object
W                           128069 non-null object
FINAL_MARGIN                128069 non-null int64
SHOT_NUMBER                 128069 non-null int64
PERIOD                      128069 non-null int64
GAME_CLOCK                  128069 non-null object
SHOT_CLOCK                  122502 non-null float64
DRIBBLES                    128069 non-null int64
TOUCH_TIME                  128069 non-null float64
SHOT_DIST                   128069 non-null float64
PTS_TYPE                    128069 non-null int64
SHOT_RESULT                 128069 non-null object
CLOSEST_DEFENDER            128069 non-null object
CLOSEST_DEFENDER_PLAYER_ID  128069 non-null int64
CLOSE_DEF_DIST              128069 non-null float64
FGM                         128069 non-null int64
PTS                         128069 non-null int64
player_name                 128069 non-null object
player_id                   128069 non-null int64
dtypes: float64(4), int64(10), object(7)
memory usage: 20.5+ MB
```

Fig. 1. Feature Summary in Shot Log Dataset

C. Data Preprocessing

First we cleaned the dataset by handling any NaN values as well as any non-sensical values. After inspection, only the $SHOT\_CLOCK$ column contained any NaNs, with around 5,500 out of 128,000 entries missing. Instead of deleting all of the rows with no $SHOT\_CLOCK$ information, We found a way to impute this value without losing too much valuable information. After some close inspection, it looks like there are only 10 entries in the dataset where the $SHOT\_CLOCK$ is equal to the game clock. This is a lot less than what is expected, due to the fact that there should be many instances where a team gets possession of the ball with less than 24 seconds on the game clock, causing the shot clock to be equal to the time remaining on the game clock. After re-reading the NBA rules, it says that the shot clock is actually turned off when a team gets possession of the ball when the game clock is under 24 seconds. Therefore, we can replace the shot clock with the value of the game clock if the shot clock is NaN and the game clock is less than 24 seconds. This accounts for 3,554 of our 5,500 NaN entries. The rest of the 2,000 NaNs, however, don't have as easy as an explanation, and are probably due to some recording error within the camera system. Thus, we will delete those rows from our analysis, as $SHOT\_CLOCK$ intuitively should be an important predictor.

The next preprocessing we did involved handling non-sensical values. The $TOUCH\_TIME$ feature describes how long the shooter has the ball before releasing his shot. There



are 300 values in which this value is negative, which makes no sense in this context. Figure 2 shows the distribution of the negative $TOUCH\_TIME$ values compared to the normal $TOUCH\_TIME$ values. It appears as if the negative values, once flipped, follow the same distribution as the normal ones. This seems to suggest that the negative values was a random user entry error that was sampled from the original distribution. Therefore, it made sense to flip these negative values in order to preserve as much information as possible.

Lastly, before doing any further analysis, we will factor our response variable, $SHOT\_RESULT$, into $0/1$ values, as well as other binary categorical response variables in the features set.

III. Data Exploration

A. Feature Distribution

Figure 3 shows the distributions of the features that may be important for prediction in the final model to serve as a launchpad for exploration. It also serves as interesting information about the current status of the NBA that may help in decision making in structuring our model. Approximately 45% of the shots in the 2014-2015 season went in, giving us a good baseline to compare our models against. If our model guessed the majority class of "MISS" every single time, we should achieve an accuracy of approximately 55%

One important feature that we are missing is whether or not the shooter was fouled during the shot. This will drastically affect the chances of a miss or a make, and represents a latent source of bias we must be aware of.

Some interesting things that the distribution plot reveals is the shot selection looks in the modern NBA. Ever since 2010, more and more emphasis has been placed on the 3-pt shot. As can be seen in the $SHOT\_DIST$ plot, a significant amount of shots came from the $22-24$ft area, which is right at the border of the 3-pt line.

Another interesting thing is the $SHOT\_CLOCK$ distribution, which looks relatively normally distributed, except for a large of spike of shots at the 24-second mark, which probably makes sense due to players chucking up shots at the end of the $SHOT\_CLOCK$ timer to avoid a 24-second violation. One important feature this dataset lacks is whether

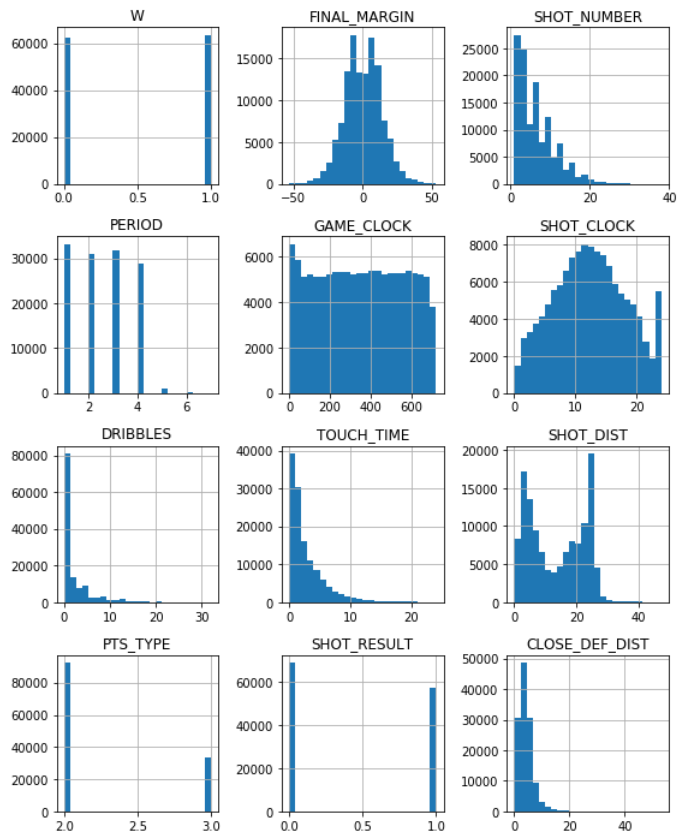


Fig. 3. Feature Distributions

the shot is the result of a initial posession, or a recovered possession, perhaps from a rebound or an early steal off of the entry pass. I believe shots coming off of second chance possessions should have a higher success rate, due to the fact that many of such shots are easy put-backs or tip-ins after the defense is scrambled by the initial shot. Since we don't have this information within the dataset, we also need to factor this latent source of bias as well.

B. Feature Correlations

In Figure 4, we created a correlation heatmap that is able to visualize the correlation between all of the features, as well as with the response variable, $SHOT\_RESULT$. There are quite a few variables that were heavily correlated with each other, however after closer inspection, it's clear that it's because there's some redundant information within the features. For example, $FINAL\_MARGIN$ and W are heavily correlated because when the $FINAL\_MARGIN$ is positive, the game obviously results in a win. Furthermore, neither of those features are something we wanted to use in a predictive model, since those features aren't ascertained until after a game is over.

Another set of features that is heavily correlated is PTS_TYPE and $SHOT\_DISTANCE$ for obvious reasons. Therefore, it made the most sense for our model to only utilize one of these.

Additionally, $DRIBBLES$ and $TOUCH\_TIME$ are

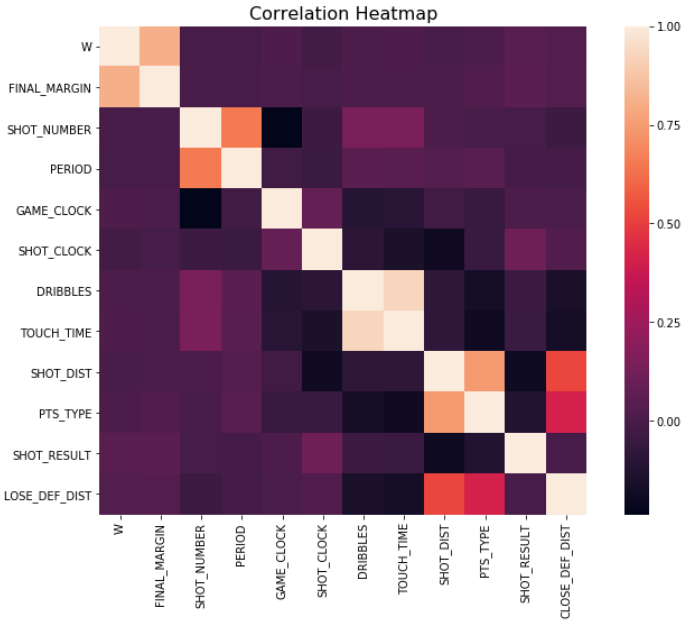


Fig. 4. Feature Correlation Heatmap

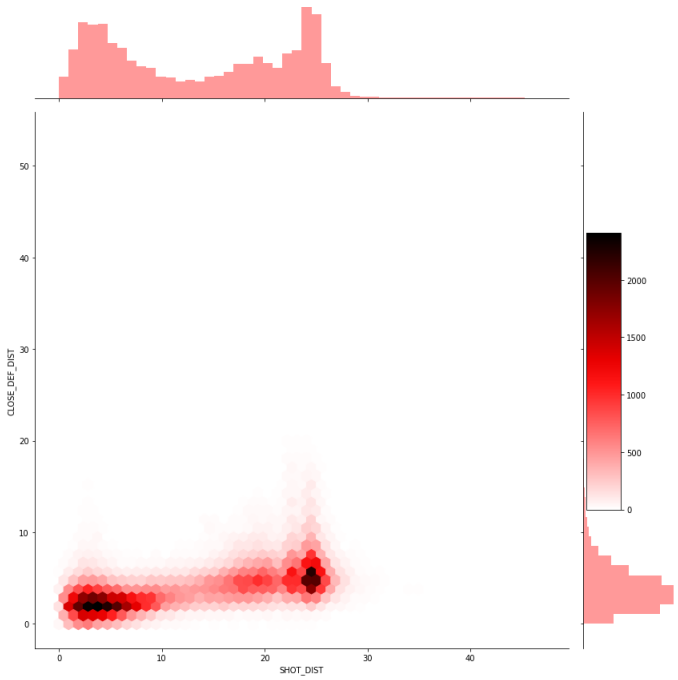


Fig. 5. Shot Distance vs. Closest Defender Distance Heatmap

very correlated with each other, but also contain non-overlapping information that could be useful to the predictor. For example, a possession by a player like Carmelo Anthony, who likes to hold on to the ball without dribbling for a long time before shooting should be treated differently than a player like Kyrie Irving, who tends to dribble a lot before each shot. Therefore, it would make sense to retain both in the model.

One interesting relationship was between $SHOT\_DIST$ and $CLOSE\_DEF\_DIST$. It would make sense that if the farther away a shooter is from the basket, the farther away they are from the defender as well. Furthermore, the farther away the shot occurs, the harder it is to make, incentivizing the defender to contest the shot less and less as the shot distance increases. Figure 5 shows a heatmap that visualizes this relationship. Most shots are either layups or 3-pt shots, which really highlights the trend the modern NBA is moving towards. If this plot were created for the early 2000s period, there would be much more red located in the mid-range portion. It seems that a majority of the 2-pt shots are defended tightly, with the closest defender within 2-5 feet. In contrast, 3-pt shots are defended much less tightly, with the closest defender distance centered around 5 feet away.

C. Feature Analysis

Figure 6 shows the relationship between $SHOT\_DISTANCE$ and our response, $SHOT\_RESULT$. Surprisingly, there is very little relationship between field goal percentage and distance from closest defender, unless the distance is absurdly high, something like 20ft+. This happens very rarely, probably only when a player gets a fast break off of a steal and is completely ahead of the pack. There seems to be a dip in field goal percentage at around 28 ft away down. We

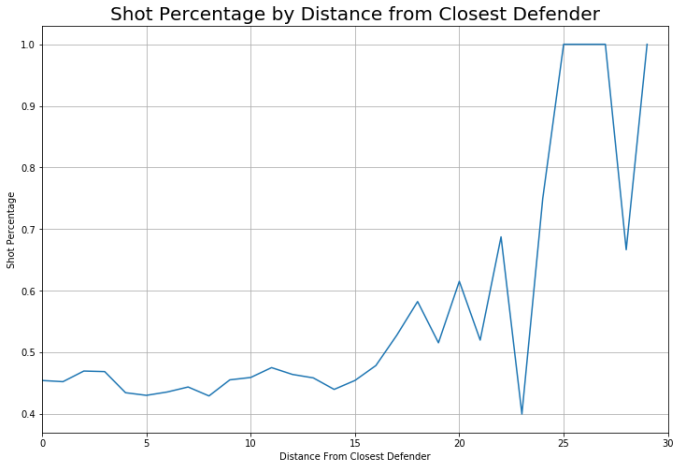


Fig. 6. Shot Distance Feature Analysis

don't want our model to overfit on these large distances, so instead of treating this predictor as a numerical feature, we bin this feature into 5 categories, which were recommended by `www.NBA.com`. The binning function is as follows:

- $0-2$ feet away - Very Tightly Defended
- $2-4$ feet away - Tightly Defended
- $4-6$ feet away - Open
- $6-10$ feet away - Very Open
- $10+$ feet away - Completely Open

Hopefully this encoding will reduce the variance of our model while minimizing the increase of potential bias it introduces.

Figure 7 indicates that there is a strong relationship between $SHOT\_CLOCK$ and our response variable. It

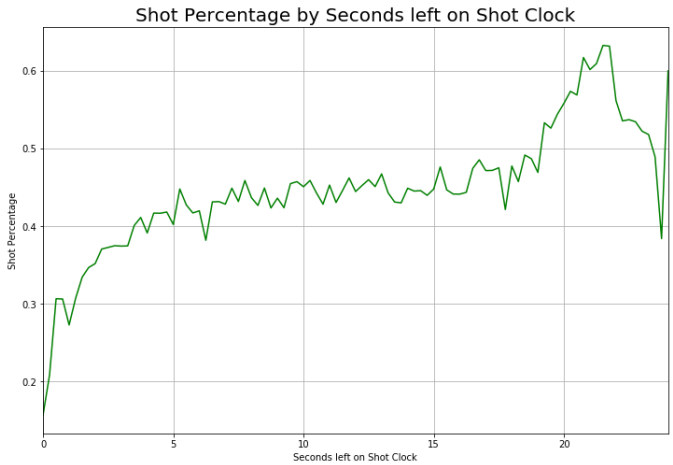


Fig. 7. Shot Distance vs. Closest Defender Distance Heatmap

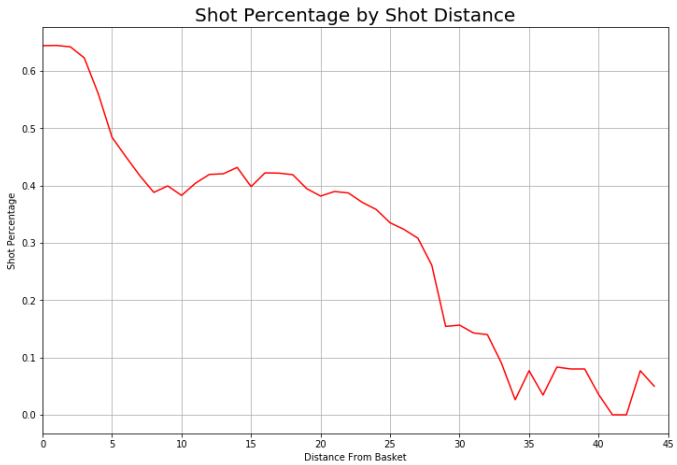


Fig. 8. Shot Distance vs. Accuracy

appears that shots that occur very early on in the shot clock have a significantly higher percentage than normal. This probably is aided by high percentage fast break plays that end up scoring very early into the shot clock. As the shot clock approaches 0, the accuracy dives. This is probably due to the reason mentioned earlier about the shooter's pressure to put up a shot before the shot clock expires. This feature thus is important to include in our model.

Figure 8 highlights the obvious relationship between $SHOT\_DISTANCE$ and our response. As the shooter gets further away from the basket, the chance of success drops rapidly. One surprising thing to notice is how the efficiency of a shot 10-ft away is almost the same as a shot 22-feet away. This trend explains why teams are moving away from the midrange shot. Though it is just as likely to fall as the 3-pt shot, it rewards the team only 2-pts, a 50% reduction per possession than a 3-pt shot. Furthermore, the shot percentage takes a dramatic drop from $0-5$ feet to 10 feet with no increase in points given per difficulty of shot. This trend further explains why the NBA is moving towards taking shots with a high expected value, namely layups and 3-pt shots. Furthermore, this trend shows that including $SHOT\_DISTANCE$ as a feature will improve our model.

IV. Dimensionality Reduction

One important feature to include in our dataset is the specific player performing the shot, as well as the specific player defending the shot. Unfortunately, this presents a concern computationally, as there are 281 unique shooters in the dataset and 487 unique defenders in the shot logs. Once these are one-hot encoded and joined into the dataset, our dataset shape is (125,000, 705). This presents a challenge, as some models we tried to use, such as logistic regression, were unstable and had overflow errors. This section explores how we can reduce the dimensionality of our data. One idea was to use PCA in order to find a low rank approximation to our data matrix that might ease our computation woes. First we scaled all of our data to values between 0 and 1 using

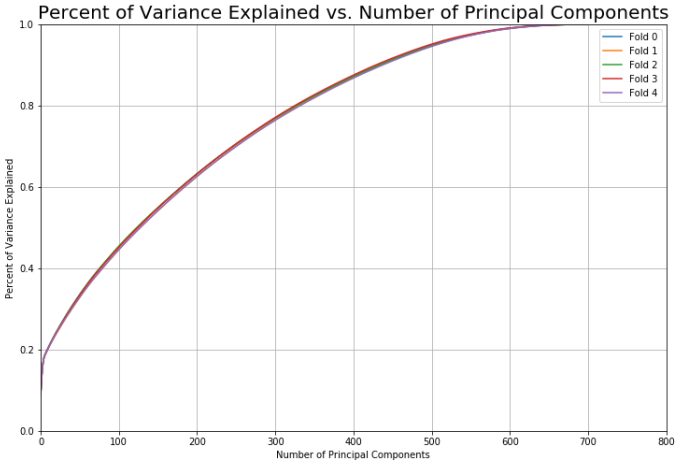


Fig. 9. Results vs. PCA

the following equation:

$$\frac{x - x_{min}}{x_{max} - x_{min}} \tag{1}$$

Figure 9 shows the results of the variance retained vs. the components used after performing 5-fold cross validation. It seems as if PCA isn't appropriate in this situation. I hypothesize that the nature of the sparsity of this dataset makes it hard to find a good low-rank approximation.

I talked to Su in class about possible alternatives, and one possible alternative included fitting a regression on the non-player specific features, and then treating each player as an intercept. I couldn't quite figure out how to do this for both the shooter and the defender as well. I should have at least tried to perform that method for just the shooter, as that is one of the most important predictors, however due to time constraints, I was unable to add this in. Thus for our model, we do not factor in the shooting player or closest defending player at all.

Logit Regression Results

Dep. Variable:	SHOT_RESULT	**No. Observations:**	126051
Model:	Logit	**Df Residuals:**	126040
Method:	MLE	**Df Model:**	10
Date:	Wed, 05 Dec 2018	**Pseudo R-squ.:**	0.04412
Time:	19:33:27	**Log-Likelihood:**	-82968.
converged:	True	**LL-Null:**	-86797.
		LLR p-value:	0.000

	coef	std err	z	P>|z|	[0.025	0.975]
SHOT_NUMBER	0.0030	0.001	2.362	0.018	0.001	0.006
SHOT_CLOCK	0.0174	0.001	17.174	0.000	0.015	0.019
GAME_CLOCK	-1.623e-05	2.88e-05	-0.564	0.573	-7.26e-05	4.02e-05
DRIBBLES	0.0290	0.005	6.257	0.000	0.020	0.038
SHOT_DIST	-0.0668	0.001	-74.531	0.000	-0.069	-0.065
TOUCH_TIME	-0.0628	0.005	-11.555	0.000	-0.073	-0.052
Very Tightly Defended	0.0838	0.025	3.331	0.001	0.035	0.133
Tightly Defended	0.4937	0.025	19.952	0.000	0.445	0.542
Open	0.8396	0.029	29.213	0.000	0.783	0.896
Very Open	1.0040	0.032	31.269	0.000	0.941	1.067
Completely Open	1.1566	0.043	27.094	0.000	1.073	1.240

Fig. 10. Logistic Regression Summary

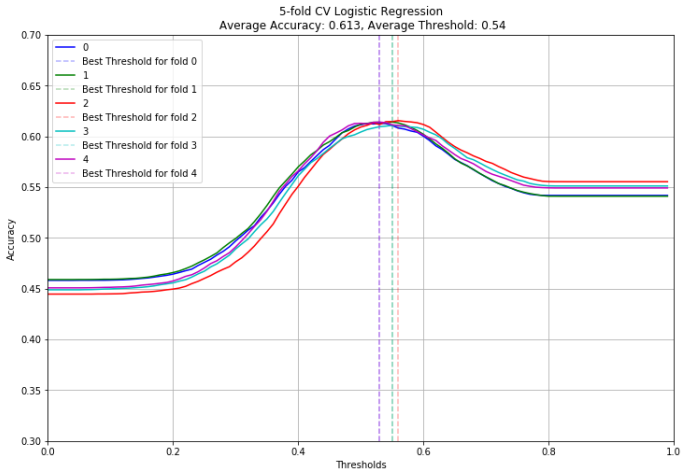


Fig. 11. Logistic Regression Threshold Selection

V. Model Results

A. Logistic Regression

Figure 10 shows the results of logistic regression over all of the data. With an R^2 score of 0.044, our model isn't that great at explaining the variability within the dataset. All of the predictors were significant, except for the $GAME\_CLOCK$ predictor. Some interesting results can be derived from this summary. When interpreting the coefficients, it can be seen that as the $SHOT\_DISTANCE$ increases with all other coefficients constant, the chance of shot success decreases as well. Furthermore, as shooters are less tightly defended, the chance of shot success increases. Interestingly, $TOUCH\_TIME$ and $DRIBBLES$ both seem to have a positive correlation with shot success. Personally, I would have imagined the opposite.

Figure 11 describes the results from 5-fold cross validation. After averaging the threshold that performs the best on all of the folds, we achieve a threshold of 0.54 with a cross-validated accuracy of 61.3%. Furthermore, we achieved a cross-validated precision of 64.2% and recall of 32.9%. These results aren't that impressive compared to the baseline accuracy of 55% by guessing the majority class.

B. Gradient Boosted Trees

In an effort to improve on the results of logistic regression, we tried applying gradient boosted trees using the XGBoost package on this dataset. Gradient boosted trees is one of the best performing architectures in Kaggle competitions for classification tasks. Since feature selection is a little bit more important in decision trees, we decided only to use $SHOT\_NUMBER$, $SHOT\_CLOCK$, $DRIBBLES$, $TOUCH\_TIME$, and the binned $CLOSE_DEF_DIST$ as features to input into my classifier. After using the default parameters for training, we achieved a cross-validated accuracy of 62.1% with a precision of 65.2% and a recall of 65.2%. The gradient boosted trees saw an improvement of all 3 metrics.

VI. Future Work

We were slightly disappointed with the lackluster results of our classifier, but given the limited numbers of information about each shot, we didn't expect much more. In terms of improving the quality of our classification, we feel like obtaining certain features such as whether or not the shooter was fouled, which possession the shot happened on, and general descriptions about the play leading up to the shot. The exclusion of these features puts a hard cap on how well our classifier can possibly do. Furthermore, being able to include the shooting player defending player in our model will potentially increase results, but couldn't be done do to lack of time. It may be possible to categorize each player into 5-6 offensive and defensive categories based on their offensive and defensive stats and used that as a lightweight alternative to treating each specific player as a feature. Despite that, this analysis was important in highlighting important trends in the current NBA landscape. Furthermore, if the base model were improved to 80% accuracy or so, it may provide a very powerful tool for shot evaluation for teams to use in the future.