BASKETBALL SHOT SUCCESS CLASSIFICATION

STAT471 Final Project

Abstract

Although many statistics are collected for basketball players, application of modern statistical techniques on that data has been rare. In this study, classification methods are applied to a dataset of attempted shots in the career of Kobe Bryant to predict their success. They are evaluated on training misclassification error, as well as log loss on a test dataset of 5,000 shots. It is found that the logistic classifier with variables chosen by hand outperforms random forest as well as logistic regression using LASSO-selected variables. Finally, a discussion and attempt to deal with data leakage is presented.

Bill Guo

STAT 471 Spring 2016

1. Introduction

Many industries today have been revolutionized by statistics, making use of data in order to produce valuable insights. Approaching what once appeared to be qualitative problems with quantitative methods is quickly becoming the norm. Yet surprisingly, despite the ample opportunities in sports for the application of statistical tools, they largely lag behind. Only baseball is known for widespread use of statistics through sabermetrics, and even baseball has not fully embraced the use of modern statistical techniques.

This study analyzes a dataset of shots attempted by recently retired basketball great, Kobe Bryant, in order to classify each as a success or failure. Three model approaches will be presented: random forest, logistic regression, and an iterative logistic regression algorithm, explained in detail below. Through this exercise we hope to gain insight into Kobe's stellar on-court performance, and perhaps more importantly, demonstrate the viability and use of rigorous statistical methods in sports.

2. Data Description

2.1 Dataset

The dataset, obtained from Kaggle, was sourced directly from the NBA. It contains every goal attempted by Kobe in his 20-year career, a total of 30,697 shots. Of these, 5,000 were randomly selected by Kaggle to serve as a test set, with their shot success labels removed. The remainder serves as training data.

The data contains a wealth of information, with 25 variables total. A list of accompanying variables for every attempted shot is provided:

1. shot_made_flag: 1 if successful shot, 0 if not.

2. action_type: The type of shot attempted, such as jump shot, dunk, etc. There are 57.
3. combined_shot_type: Classifies the shots under 6 larger categories: Bank Shot, Dunk, Hook Shot, Jump Shot, Layup, and Tip Shot.
4. game_event_id: The ID of the event (attempted shot) in the specific match being played. Discarded for our purposes.
5. game_id: The ID of the specific match. Also discarded.
6. lat: The latitude of Kobe's position during the shot attempt.
7. lon: The longitude.
8. loc_x: The x-location on the court.
9. loc_y: The y-location on the court.
10. minutes_remaining: The minutes remaining in the specific match.
11. period: The period in the specific match.
12. playoffs: Indicator variable whether the match was in the playoffs or not.
13. season: The basketball season (2000, 2001, etc.)
14. seconds_remaining: The seconds remaining in the specific match.
15. shot_distance: The distance from which the shot was attempted, in ft.
16. shot_type: 2pt or 3pt.
17. shot_zone_area: Area from which shot was attempted (Right, Left, Center, Back Court, Right Center, Left Center)
18. shot_zone_basic: Further area information (Mid-range, restricted area, in the paint, above the break 3, backcourt, left corner 3, right corner 3)
19. shot_zone_range: Range (<8 ft, 8-16, 16-24, 24+, backcourt)
20. team_id: ID of Kobe's team. Always the Lakers, so discarded.

21. team_name: Name of Kobe's team, the Lakers, so discarded.
22. game_date: Date of the specific match. Discarded.
23. matchup: The two teams in the specific match. Since Kobe was always on the Lakers, opponent contains all the information in matchup. Matchup is thus discarded.
24. opponent: Opponent in the specific match.
25. shot_id: ID (from 1 to 30,697) of the attempted shot.

As mentioned, several variables were removed from consideration. They include game_event_id, game_id, team_id, team_name, game_date, and matchup. This leaves us with 18 predictors and one response, shot_made_flag.

The dataset is further cleaned by combining the 5 least attempted shots in action_type into an "other" category. This was because R's random forest function can only handle factors with 53 levels.

2.2 Data Exploration

Now we explore the remaining predictors, to find meaningful relationships with shot_made_flag. From here, we will proceed to take two approaches:

1. Using LASSO to select variables.
2. Using our findings here and our intuition to guide our variable selection.

First, we note Kobe's overall accuracy, 44.61%. Now we look at action_type. Note that while there are 57 types of shots, Kobe makes several much more often than the rest. In fact, two of the types are never shot (Turnaround Fadeaway Bank Jump Shot and Cutting Finger Roll Layup Shot). Of the rest, Kobe attempted 15,836 jump shots, 2,154 layup shots, and 1,628 driving layup shots.

The natural question is how accuracy varies by shot type. As it turns out, it varies substantially (Fig 1):

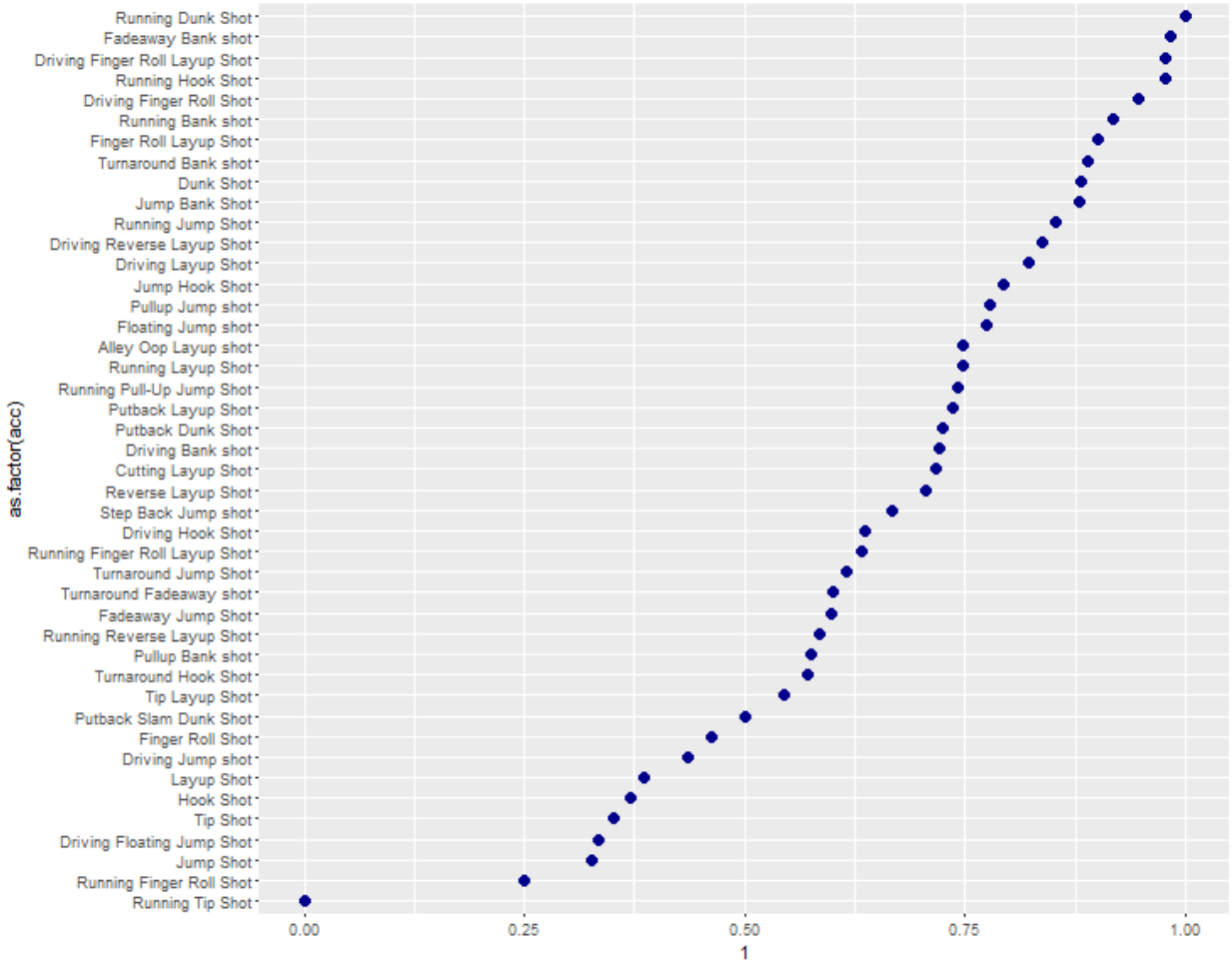


Figure 1 - Accuracy by Shot Type

The Y-axis depicts the type of shot, and the X-axis the accuracy. Kobe's jump shot accuracy is ~33%, and his layup shot accuracy is ~38%. More importantly, there is enormous variation here, suggesting the importance of action_type.

Next, we examine location data (Figure 2). By plotting loc_y v. loc_x a visualization of the shots Kobe made and missed by location is depicted. It is rather difficult at first glance to discern any differences. One point that becomes obvious, however, is the impact of range. There are many more misses than makes at longer ranges, meaning the 3-point line and beyond. Within

the 3-point area the data is too noisy to analyze. Keeping this in mind, we will revisit range later.

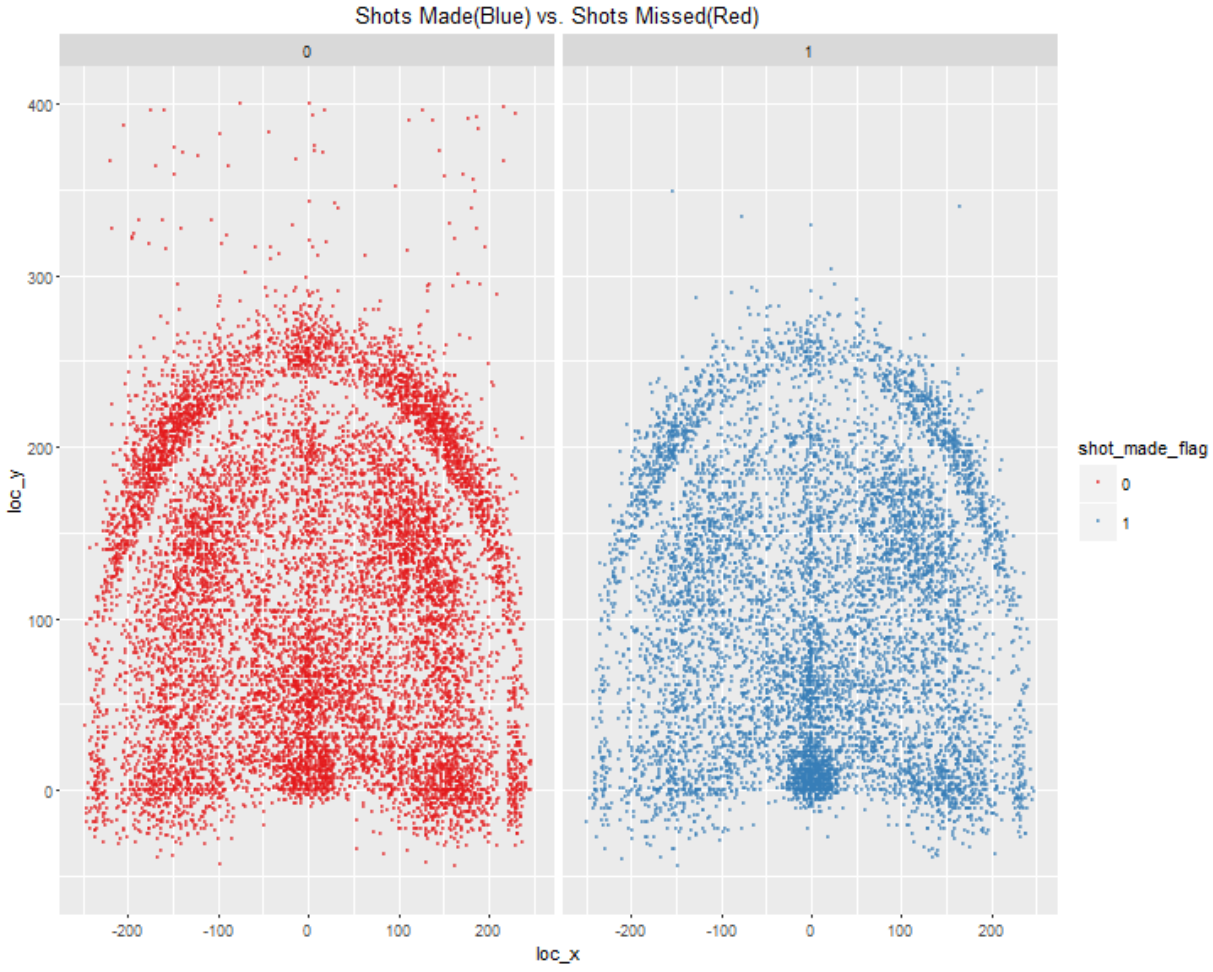


Figure 2 - Shots made v. missed, by location

Figure 3 provides a visualization of shot_zone_area, showing the on-court representation of each zone. Figure 4 shows the accuracy and number of shots in each zone. As expected, shots from the backcourt are at such great range that the accuracy is extremely low.

Though the variation in accuracy appears large, we must note that backcourt shots are rare. Of course, such shots are hardly ever attempted by any player, and Kobe is no exception. Among the remaining zones, Kobe appears to slightly prefer the right, and highly prefers the center. He also shoots much better in the center, and slightly better in the right zones. A quick search confirms that Kobe shoots with either hand, but that his right is dominant.

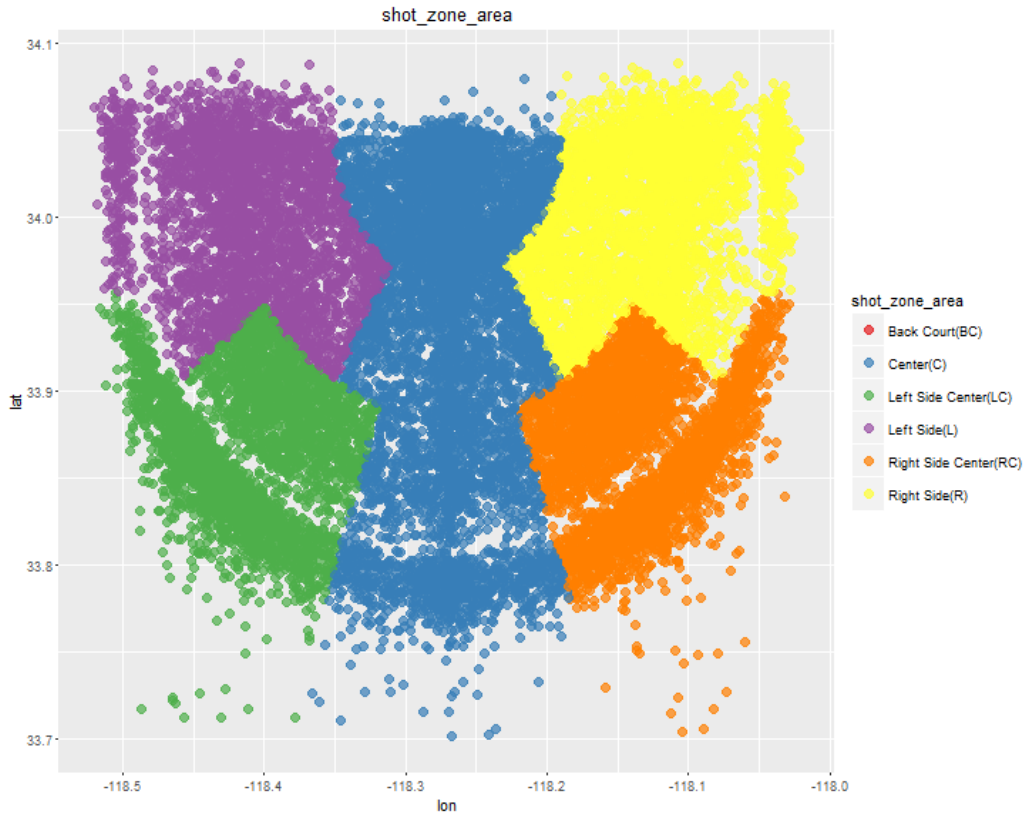
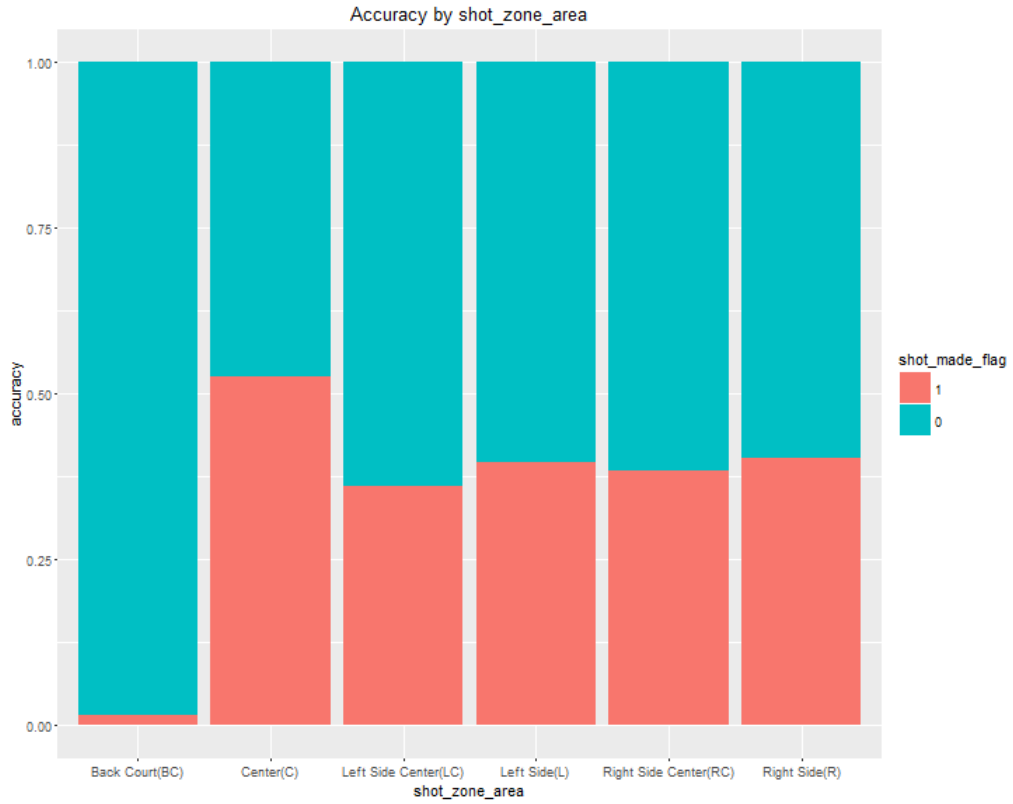


Figure 3 - Shot zone area



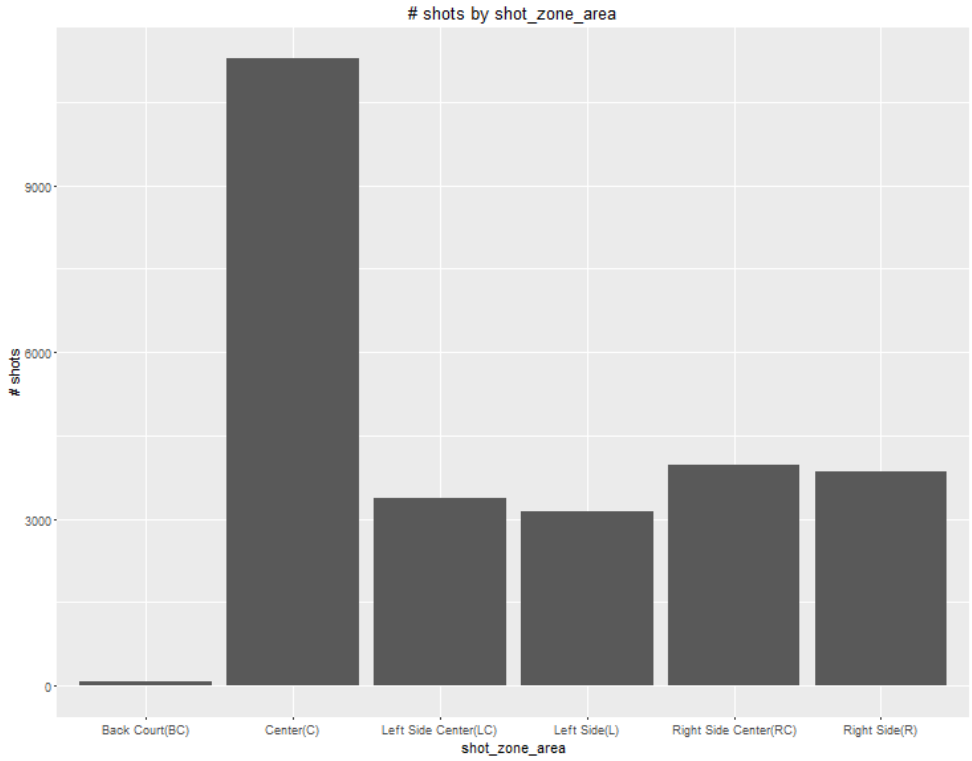


Figure 4 - Accuracy and # of shots by shot_zone_area

Shot_zone_area does not vary as much as action_type, but it is probably useful for prediction to some degree.

Next we tackle shot_zone_basic. Figure 5 provides a visualization of the on-court locations, and Figure 6 the accuracy and number of shots by location.

Kobe's accuracy by shot_zone_basic actually varies substantially even after we account for the fact that shots from the corners and the backcourt are very rare. Surprisingly, Kobe's left corner accuracy is higher than right corner accuracy.

We are tempted to conclude shot_zone_basic should be included in our model. However, in actuality the variable is just describing the influence of range on accuracy. We will need to analyze shot_zone_range in and shot_distance in order to decide on shot_zone_basic's inclusion. For instance, it could well be the case that shot_zone_range contains all the information of shot_zone_basic and more, or vice-versa.

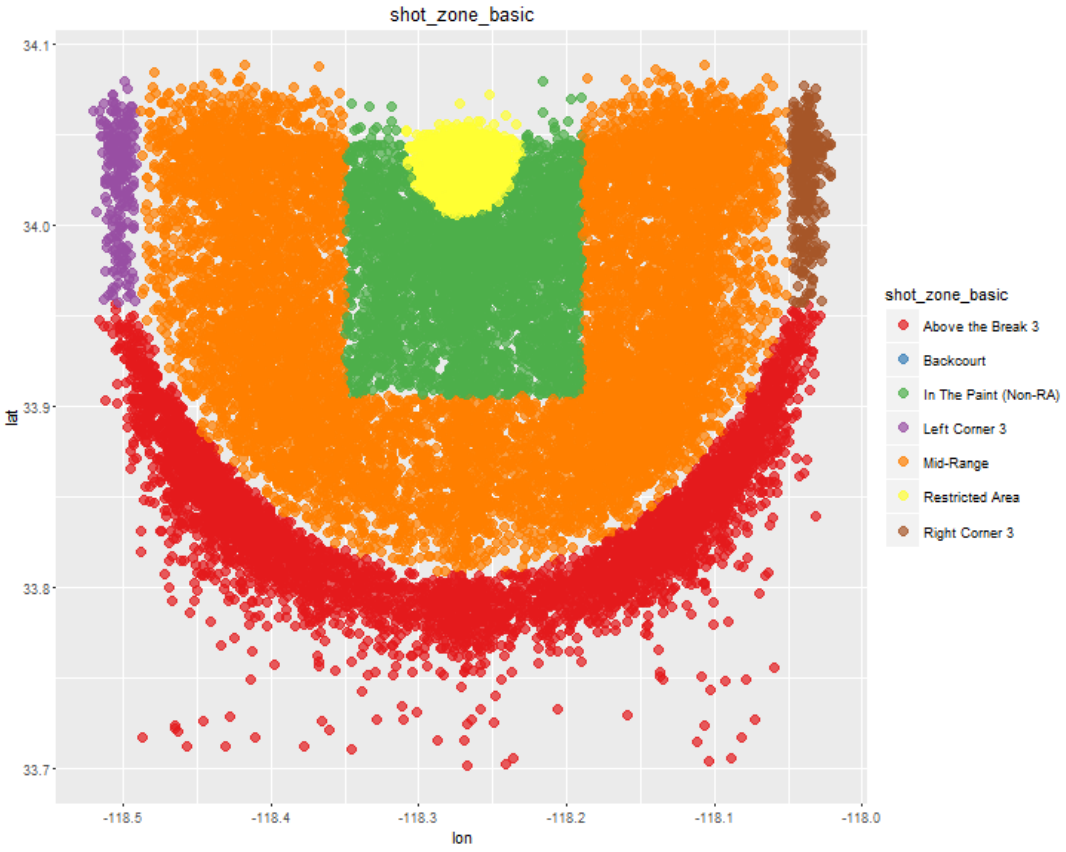
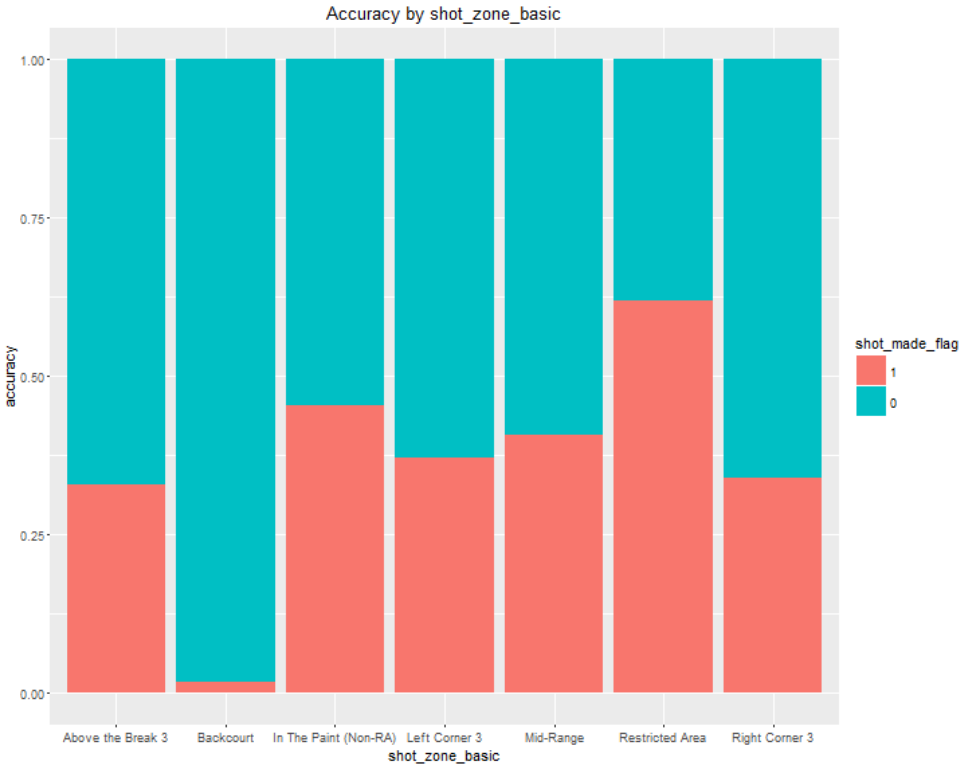


Figure 5 - Shot zone basic



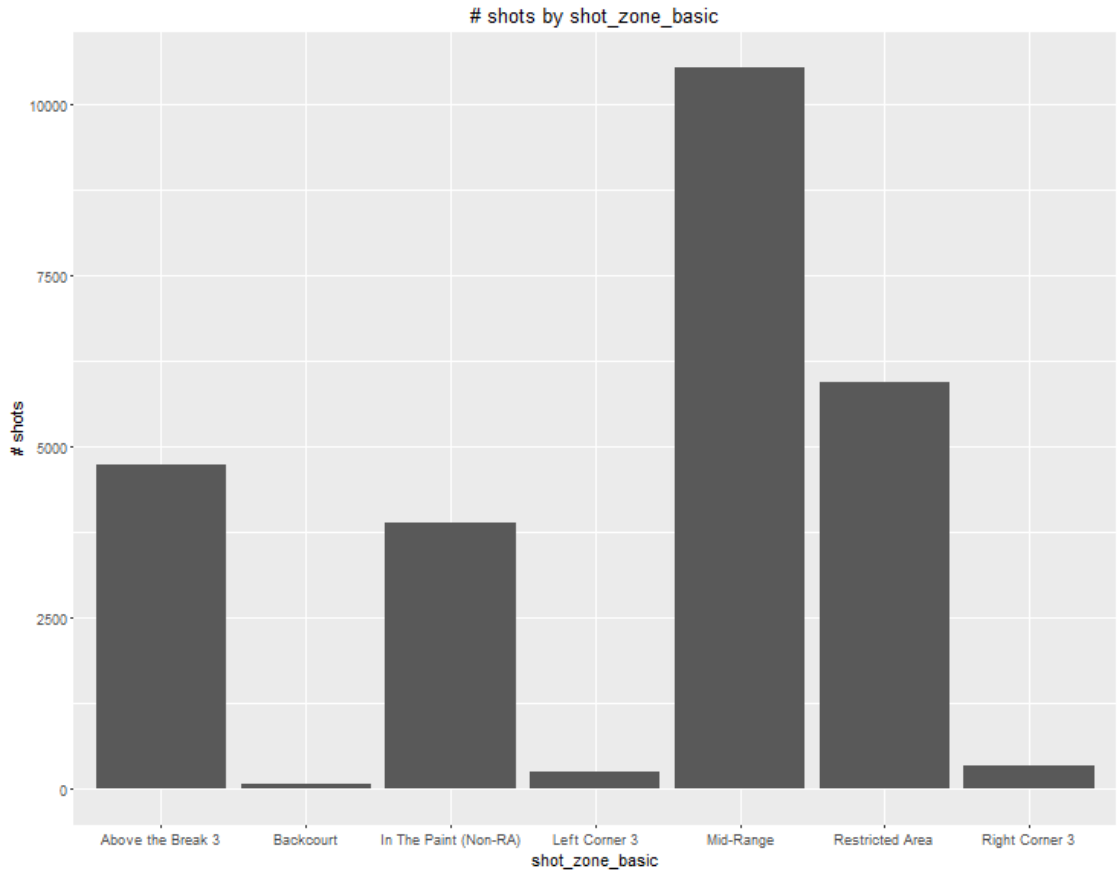


Figure 6 - Accuracy and # shots by shot_zone_basic

Figure 7 is a visualization of range on-court, and Figure 8 is the customary accuracy and # of shots plot. As suspected, the same conclusions from shot_zone_basic hold. Kobe shoots more < 8ft (restricted area) shots and 16-24ft (midrange) shots than others. His accuracy by range also corresponds exactly to the zone_basic accuracies.

That is, we should include only one or the other. Shot_zone_basic does contain additional information, however, such as breaking down shots by corner and separating < 8ft shots into restricted area and in the paint shots.

Shot_distance (Fig 9) provides more granularity than even shot_zone_basic, and the same conclusions. It may be advisable to include it over the other two. One thing is for certain, however; one variable must be included to account for range.

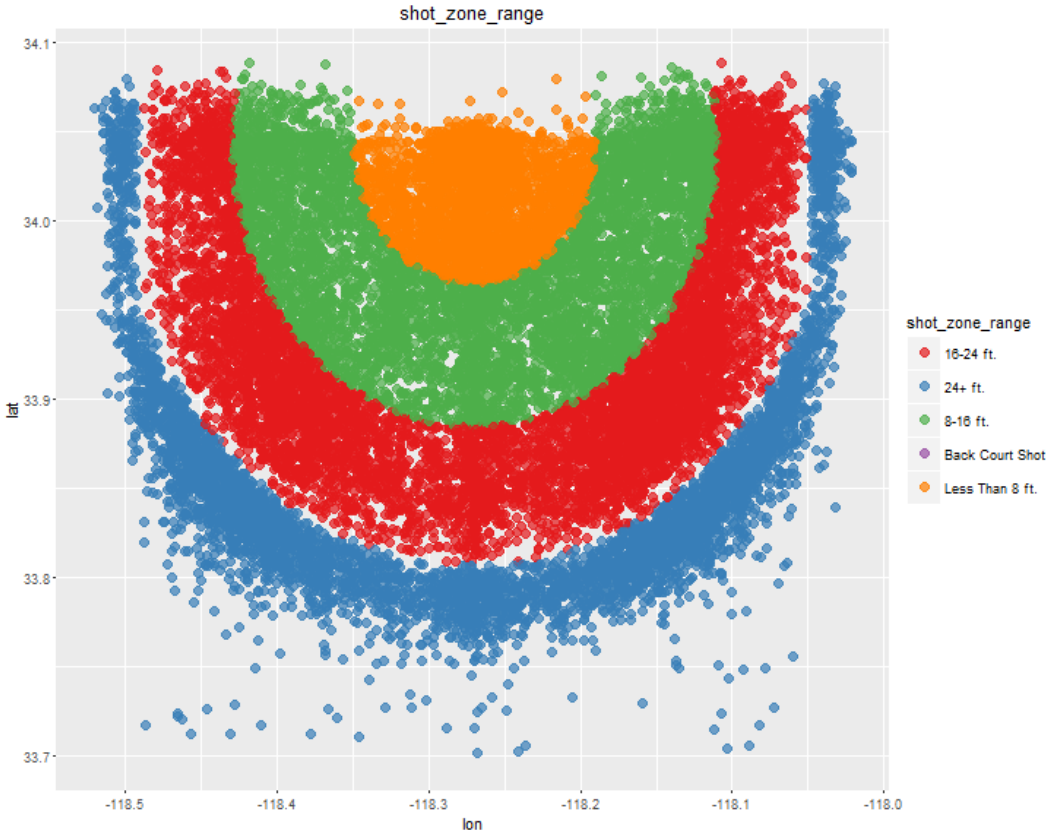
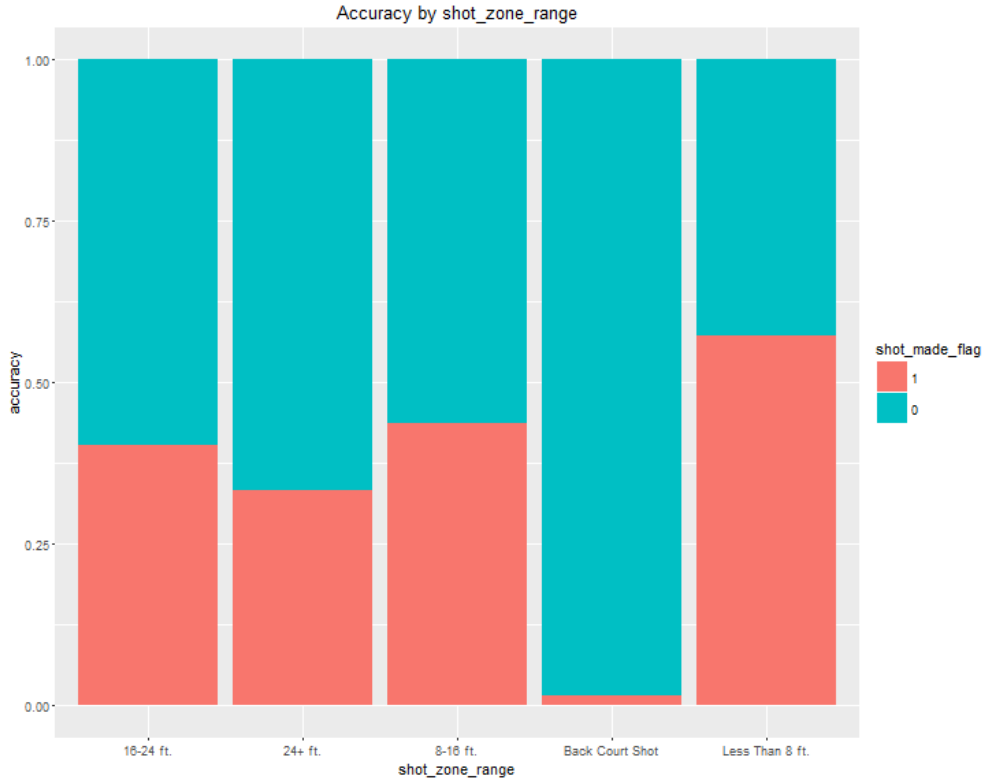


Figure 7 - Shot zone range



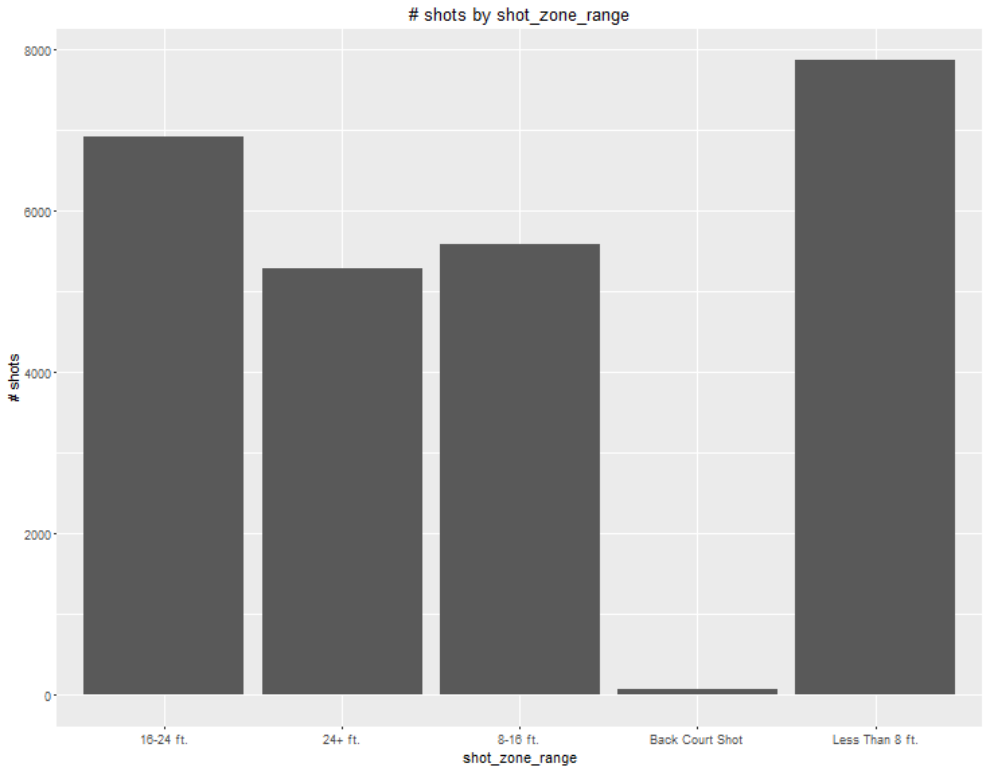
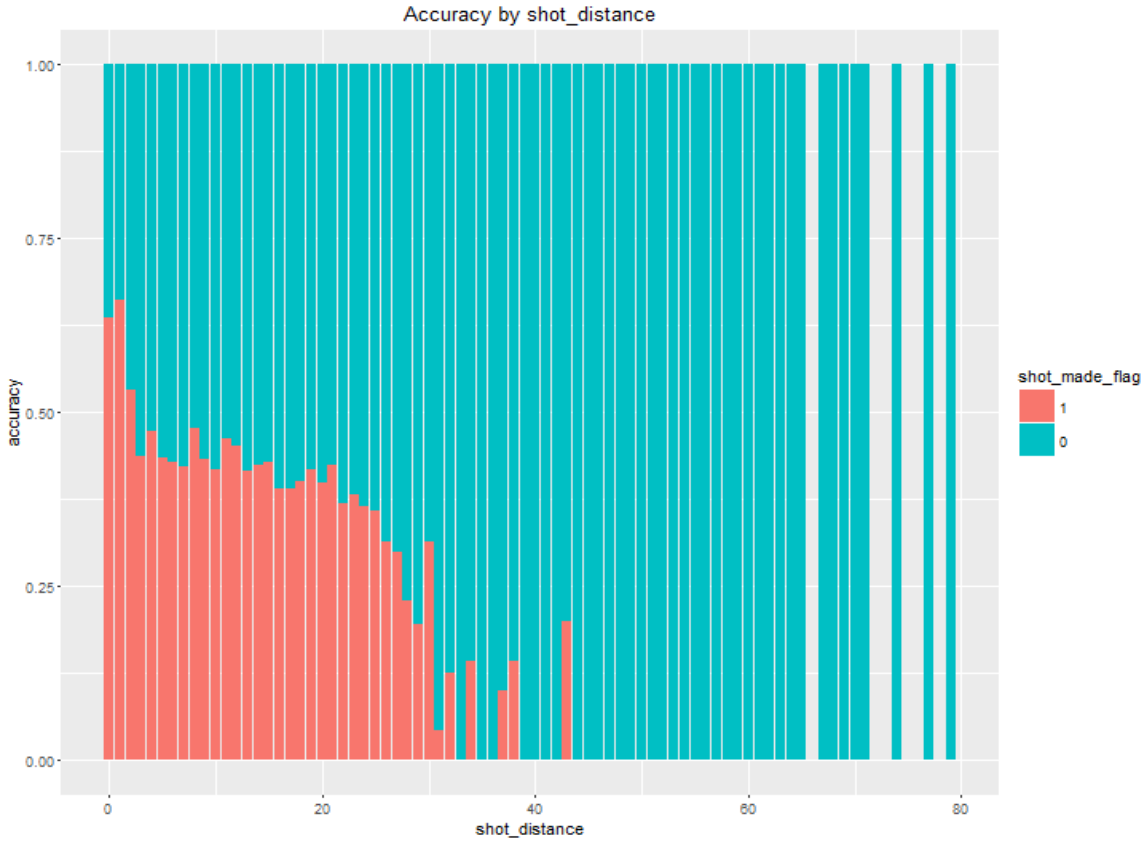


Figure 8 - Accuracy and # shots by shot_zone_range



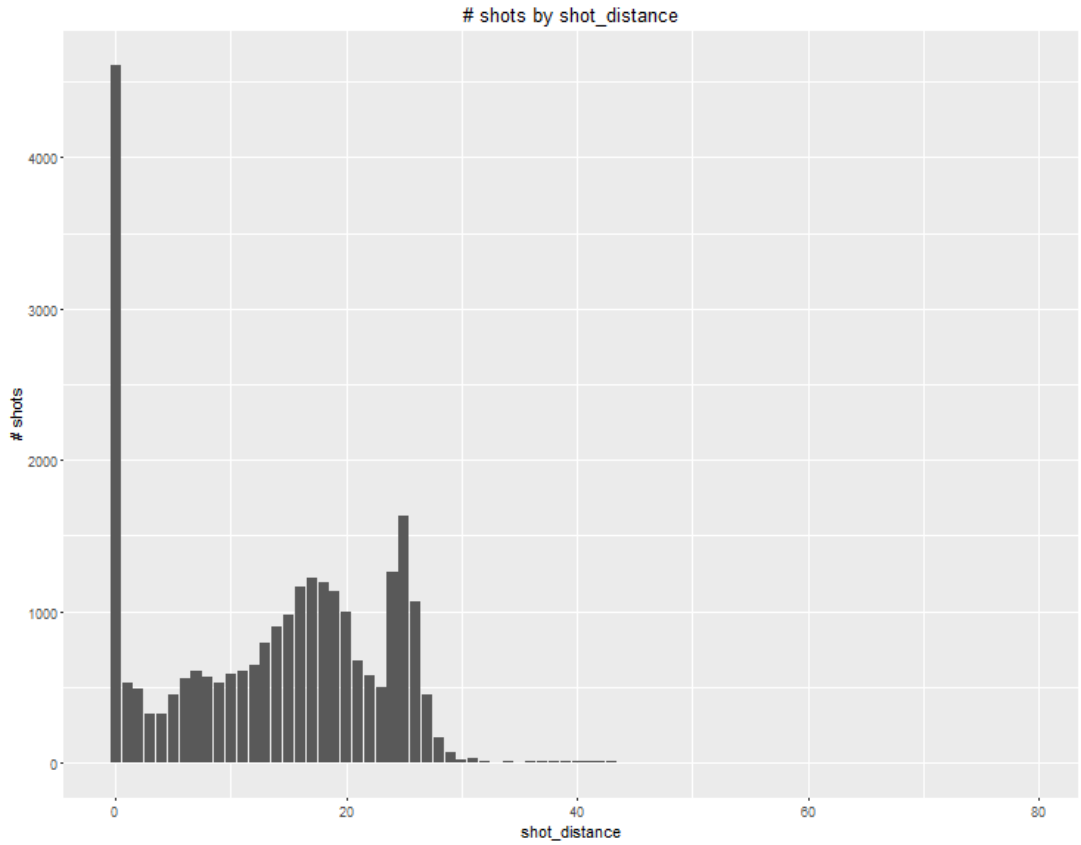
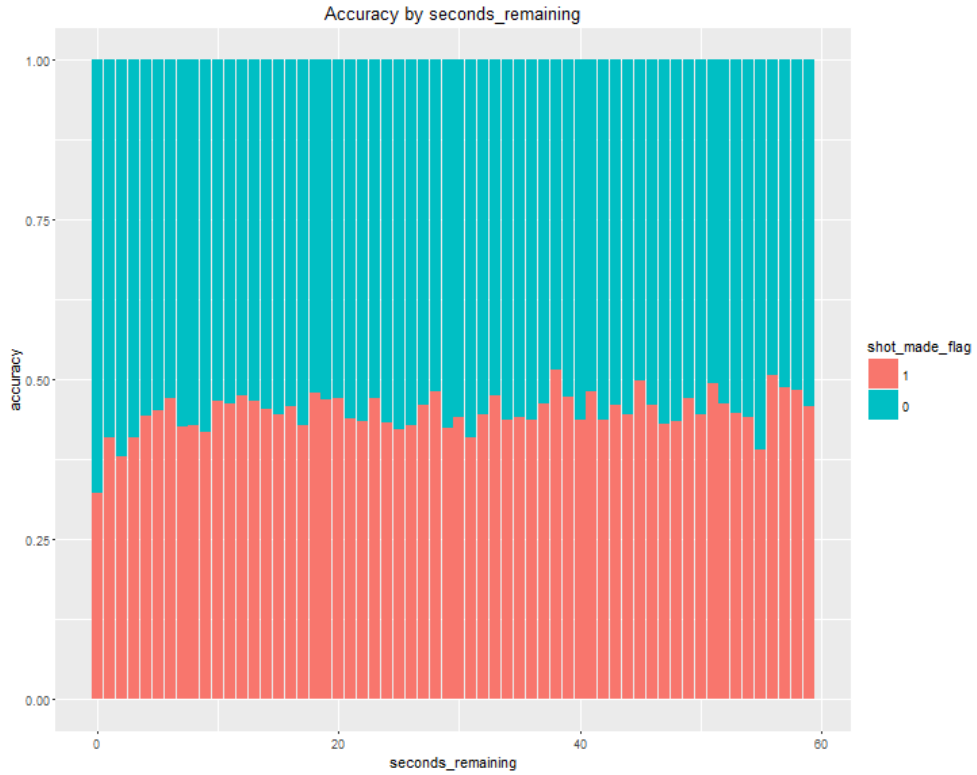


Figure 9 - Accuracy and # shots by shot_distance

Now for the analysis of time. Minutes remaining and seconds remaining are really the same thing, so only seconds_remaining will be analyzed.

Kobe's performance is very consistent with time, except for in the final seconds. This could be due to pressure, or more likely, due to riskier shots being taken in the final moments (such as backcourt shots). This is supported by a drastic spike in # shots at the end of the game – Kobe will take low probability shots at the end, which can be performed from longer ranges and more awkward circumstances. In normal situations Kobe would maneuver into more favorable positions before shooting.

All in all, it does not seem that Kobe's performance varies enough for time to be a significant variable. Without displaying the plots, performance by period (which is really the same thing) is the same story.



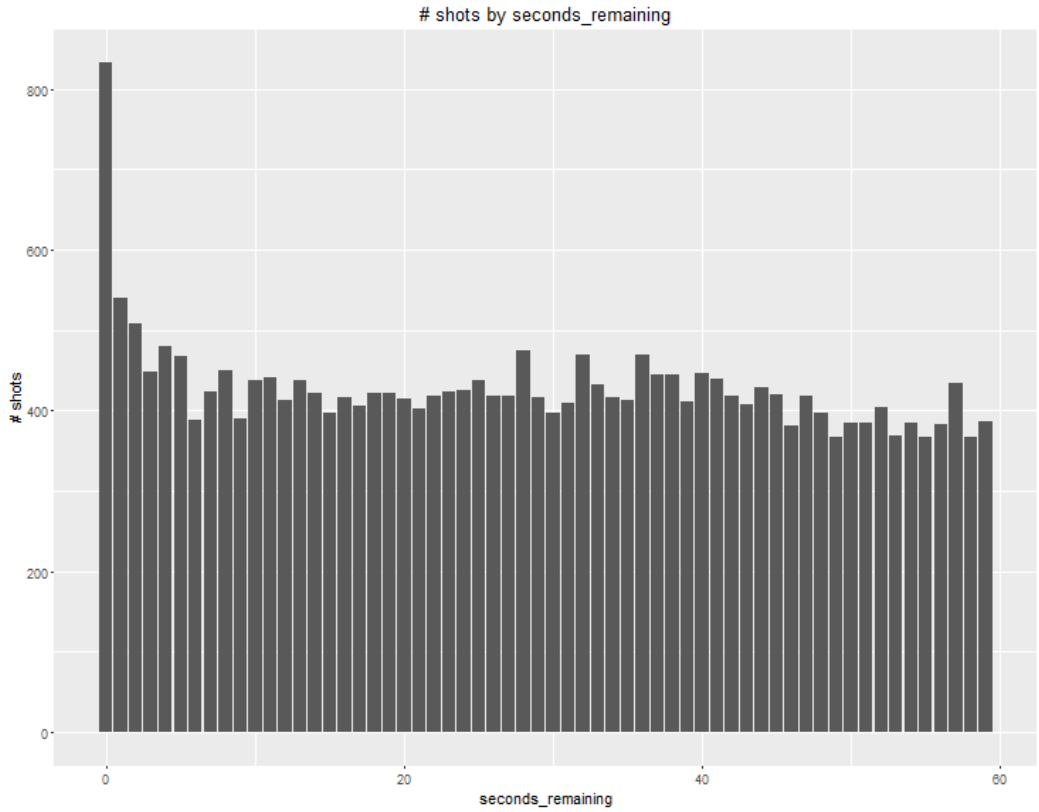


Figure 10 - Accuracy and # shots by seconds_remaining

Similarly, Kobe's performance by season and opponent do not vary either; he appears to be consistent against all opponents in all seasons. After all, if his performance dipped for some time, he would not be allowed to play by the coach. Finally, his playoffs performance does not vary either.

In conclusion, action_type, shot_zone_area, and shot_distance appear to be the best variables for predicting shot_made_flag.

3. Random Forest

Our first model approach will be to build a random forest classifier. To start off, all variables are allowed for consideration.

During tuning of the classifier, it turned out that 250 trees was enough for the OOB error to settle down. Furthermore, surprisingly, setting the number of variables considered at each split to 1 minimized OOB error. Figure 11 is the ROC. The AUC was 0.6102.

The training error performance was quite poor even using mtry=1, with misclassification error at a shocking 39.0%. Applying the model to the test set, 19.9% of shots were classified as successful – very different from Kobe's overall accuracy of 44.61%.

Looking at variable importance by decrease in GINI coefficient (Figure 12), we confirm that action_type is the most important variable. But surprisingly, opponent and season are the next best, even though our data exploration says otherwise.

All in all, the performance of the random forest is disappointing at best.

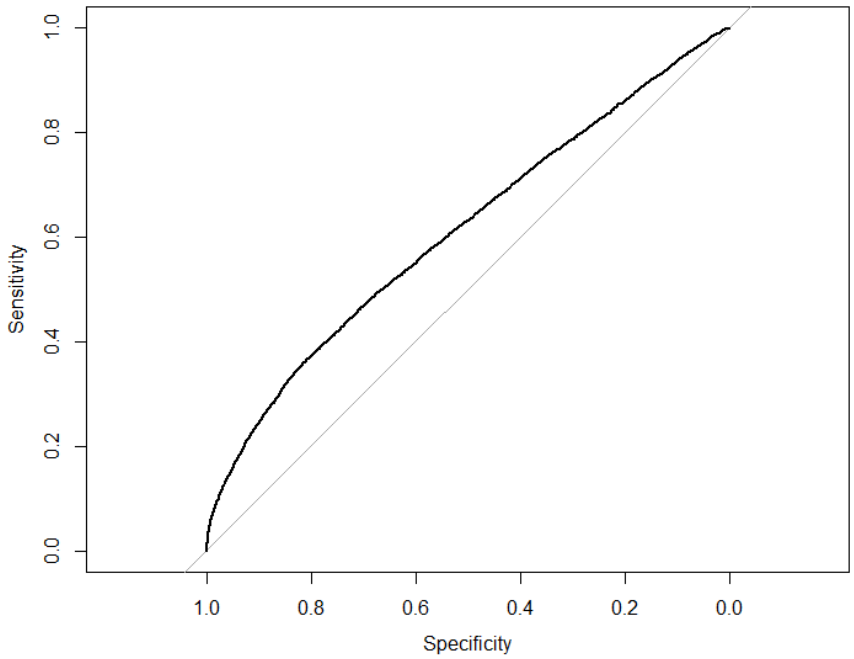


Figure 11 - ROC curve for random forest classifier, all variables allowed

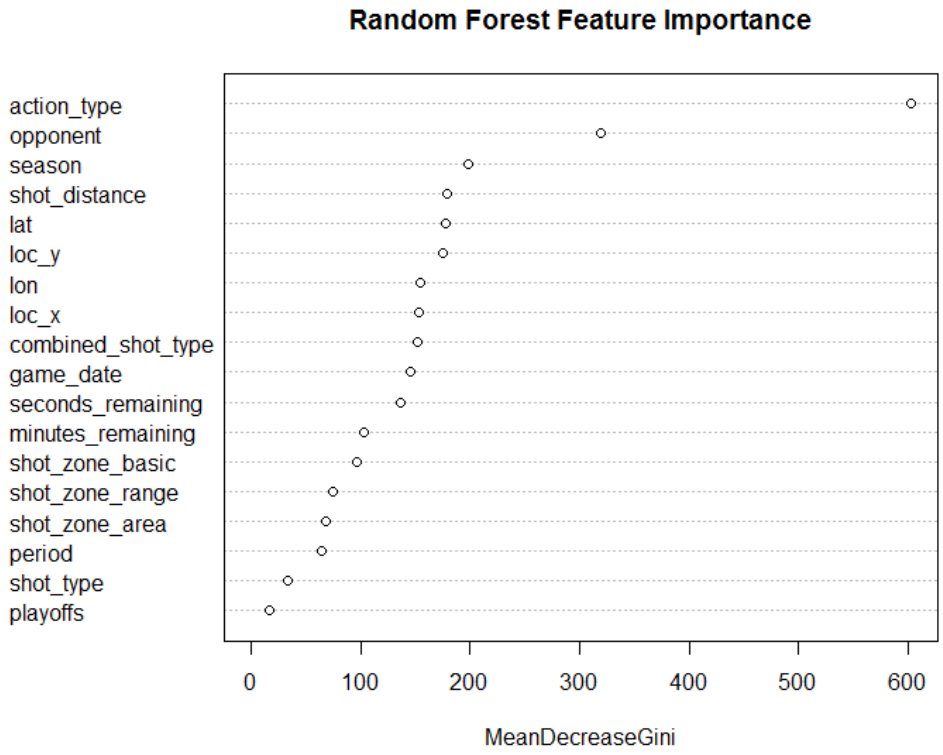


Figure 12 - Feature importance for random forest classifier, all variables allowed

4. Logistic Regression

Our second approach is to use logistic regression. Using LASSO with 10-fold cross-validation, a set of variables was selected. This set was winnowed down until all variables were statistically significant at the 0.05 level.

The final selected set of variables is action_type, period, season, seconds_remaining, shot_zone_area, and shot_zone_range. Minutes_remaining and shot_zone_basic were also significant, but removed since seconds and range were already accounted for.

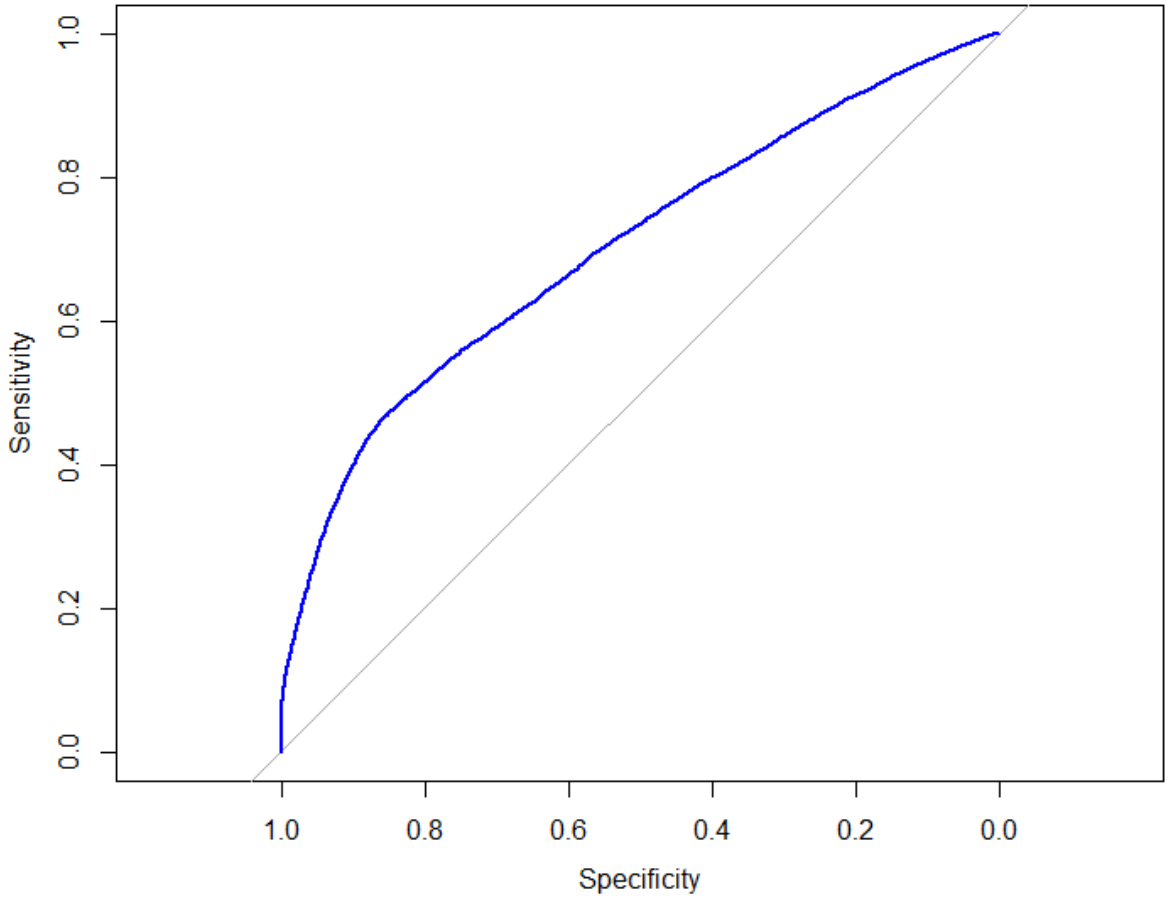


Figure 13 - Logistic classifier, ROC

Performance improved substantially, with AUC now being 0.701. Furthermore, using a naïve 0.5 threshold, the training MCE was 31.7%. Still not great, but a significant improvement from random forest.

A better threshold was obtained by using Kobe's actual accuracy of 44.61%. Since he misses more than he makes, the cost of misclassifying a make as a miss, a10, is higher than the alternative, a01. From this, using Bayes' rule, the threshold was chosen to be .4461.

The results were astonishing. Training error was reduced dramatically, down to 16.8%, using the Bayesian classifier. Furthermore, besides the improved performance over random forest, this model is also more interpretable and possesses greater explanatory power.

The predicted accuracy on the test set was 29.9%, closer to Kobe's overall accuracy.

Next, only action_type, shot_zone_area, and shot_distance were considered to build a logistic classifier. After all, our data exploration told us that period, season, and seconds remaining were all very weak predictors for shot_made_flag.

The AUC for this model was 0.688. The MCE under the threshold of 0.5 was 31.8%, and the Bayes MCE was 16.8%.

5. Dealing with Leakage

Finally, we attempt to deal with leakage. Leakage is the phenomenon where information we should not have access to is present in the training data, leading to overfit. A major challenge in machine learning, leakage will cause models to perform amazingly on training data but abysmally in test.

The problem faced here is that were we to use our models to predict Kobe's future shot successes, we should not be able to use information about shots that haven't even occurred yet. A

more intuitive example, given by Kaggle on this topic, is on a dataset relating to prostate cancer. The goal was to predict if a patient had prostate cancer. One variable in the dataset was an indicator for whether the patient had received prostate surgery. Clearly, it was highly predictive, yet useless for new patients. The resultant model was worthless on predicting whether a new patient had prostate cancer or not.

To attack this issue, a naïve algorithm was implemented. A logistic classifier was trained for each of the 5000 shots in the test set, but only on the subset of shots in the training set that occurred before the shot being predicted. This was accomplished using shot_ID to identify shots chronologically.

The logistic classifiers use the previously selected variables action_type, period, season, seconds_remaining, shot_zone_area, and shot_zone_range, under the assumption that the previously built model had explanatory power for shot_made_flag.

Due to the poor performance of random forest, as well as the time to estimate a random forest classifier, the algorithm was not implemented with it. However, that was also a viable option.

6. Results and Conclusion

To evaluate each model, the predictions on the test set were submitted to Kaggle. The output from Kaggle was the log loss. Using this metric, the performance of each model could be compared.

Model	**Log Loss**
Random Forest	13.51163
Logistic classifier, LASSO	11.07320
Logistic classifier, hand-chosen	11.01794
Leakage algorithm	n/a

The best model turned out to be the logistic classifier using only action_type, shot_distance, and shot_zone_area. Unfortunately, performance is not spectacular; the top submissions have log loss of around 0.6. But looking at those submissions, some achieved their high performance simply using random forests like done here, nothing more.

The result of the leakage algorithm was unavailable due to the computational complexity; the machine available was not capable of running it in reasonable time. One would expect it to have worse performance, however, as discussed in section 5.

One possible next step is to dive deeper into the data and build additional features. For instance, using matchup, it could be possible to identify whether the Lakers were playing at another stadium (away) or at home. An indicator variable for whether the Lakers were at home might have some predictive power.

Another logical step would be to look into more sophisticated methods of dealing with leakage. While little could be found in cursory online searches, it is doubtful that the naïve solution presented here is the optimal one.

Still, the work presented here goes a long way in showing how applicable modern statistics is to sports. One can easily imagine the implications. Team coaches can easily maintain a model for each of their players, and analyze which shots they need to improve on and which they excel at, whether their performance dips with less time remaining in the match and more pressure, who on the team should shoot longer ranges, where to position each player, etc. The possibilities are endless. Hopefully, this exercise will as a step towards more creative quantitative analyses in sports.