ORIE 5751 Project 3

Yuekun Wang yw2222@cornell.edu
Nian Ji nj282@cornell.edu

May 4, 2019

1. Baseline Models

(a) We are using FGM as the dependent variable in the logistic regression. In this model, label zero which stands for shot not made occurs more frequently in the data. The percentage of label zero in the data is 0.625. Our baseline logistic model includes the all the features except totally correlated features and ID features. The baseline logistic model's accuracy is 0.605.

We are trying to predict the shot clock for each shot in the game using the linear regression. Our baseline linear model includes the all the features except totally correlated features and ID features. In this model, the mean prediction for the remaining shot clock is 0.6 second for each shot with a squared mean error of 0.07415.

(b) For the classification problem, 0-1 is the right classification. We are trying study what affects a shot made or not. There are only two possible outcome, 1 is made, 0 is miss. We don't lose right classification for this problem so 0-1 is the right classification.

(c) The logistic regression we fit with only few covariates is called *base_ few_ clf*. In this model, we include only SHOT_CLOCK, DRIBBLES, TOUCH_TIME, SHOT_DIST, PTS_TYPE and CLOSE_DEF_DIST these features. The accuracy of *base_ few_ clf* is 0.606 which is almost same with baseline model 0.605.

The new linear regression with a few covariates includes PERIOD, GAME_CLOCK, DRIBBLES, TOUCH_TIME, CLOSE_DEF_DIST, and FGM. This model gives an in-sample r-squared of 0.00818 wich is smaller than the original baseline model of 0.00876. Also, the new model gives a mean squared error of 0.07517 which is larger than the baseline model. Therefore, this is a worse model than the baseline model indicating the possibility of all features being significant.

2. Building your model

2.1. Logistic Regression

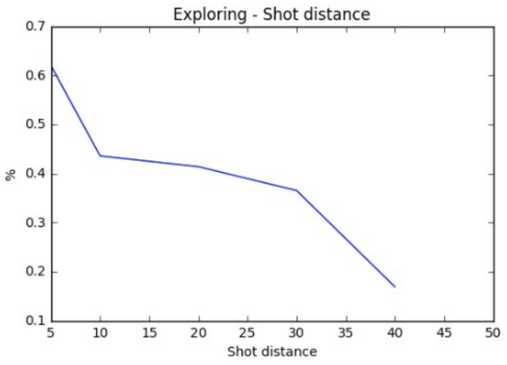
(i) **Transformations**

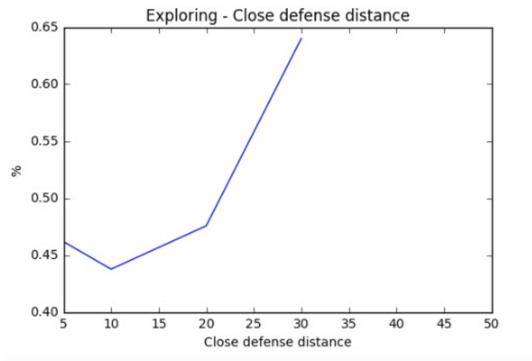
In our baseline and few variate model, we didn't normalized the numerical variables. So the first thing we do is to test whether normalized numerical variables will help improving the model performance. We normalized all the numerical variables. We built a simple model based on the normalized train data and found the accuracy is 0.603. We found transformations does not help with improving the classification problem very much and somehow can decrease the model performance. So we decided to not include transformation in our logistic model.

(ii) **Feature selections**

Our baseline model basically includes all the features and few variable model includes the variables that we think are most important. We think we need to do a feature importance analysis. The method we use is Recursive Feature Elimination. Recursive Feature Elimination (RFE) is based on the idea to repeatedly construct a model and choose either the best or worst performing feature, setting the feature aside and then repeating the process with the rest of the features. After multiple test, we decided to drop LOCATION_H, WIN_LOSE_W, PERIOD, GAME_CLOCK, FINAL_MARGIN because those features are shown less significant. From the following regression output and the figure we can tell SHOT_DIST and CLOSE_DEF_DIST are the most importance features of the model. The shot percentage is strong negative correlated with the shot distance and strong positive correlated with the close defence distance.

Features	coefficients
SHOT_NUMBER	0.0016
SHOT_CLOCK	0.0001
DRIBBLES	0.0053
TOUCH_TIME	-0.0352
SHOT_DIST	-0.0604
PTS_TYPE	0.0093
CLOSE_DEF_DIST	0.0917





(iii) **Interactions & Polynomials**

Based on the previous analysis, we decided to add two polynomial features, $SHOT\_DIST^2$ and $CLOSE\_DEF\_DIST^2$ into our model. We also think dribbles interactive with close defence distance is also an important feature to study since we can study the effect of catch and shot. We have tried adding several other interaction variables, which comes out those other interaction variables are not important to our model.

(iv) **Estimation Test Error**

The best model of all the models we tried includes features SHOT_NUMBER, SHOT_CLOCK, DRIBBLES, TOUCH_TIME, SHOT_DIST, PTS_TYPE, CLOSE_DEF_DIST, $SHOT\_DIST^2$, $CLOSE\_DEF\_DIST^2$, and $CLOSE\_DEF\_DIST * DRIBBLES$. We think this is the best model because based on the previous analysis, we delete all the irrelevant features and add few more relevant features. Our Baseline model's accuracy is 0.605. The estimate of our test error is 0.35, which means the accuracy should improve to around 0.65. We made this estimation because we think the features we add would help with explaining some of variances but not much. So our model will improve slightly but not significantly.

2.2. Linear Regression

(i) **Transformation and Regularization**

We already normalized all numerical variables in our baseline model so we want to see if transformation would help.

After calculating the covariances of the data set, we found little correlation between features. Therefore, not much transformation could help improving the model. In this case, we want to add interactions to improve r-squared and mean value.

(ii) **Two-way Interaction Regression**

We start with the two-way interaction regression by adding interactions of every two features. This gives a better in-sample r-squared value of 0.01244 and a better out-of-sample r-squared value of 0.0100. However, the mean squared error is 0.07419 which is still larger than the baseline model.

(iii) **Stepwise Regression**

Stepwise regression is a method of fitting regression models in which the choice of predictive variables is carried out by an automatic procedure by comparing the AIC scores by including or excluding each interactions. A higher AIC score indicates a higher quality of the model. After finishing the forward and backward regression, we are left with 25 variables (More details can be found in the ipynb). This model gives the highest in-sample r-squared value of 0.0121 and out-of-sample r-squared value of 0.0132, as well as the smallest mean squared error of 0.07407.

(iv) **LASSO**

LASSO regression is a type of linear regression that uses shrinkage. Shrinkage is where data values are shrunk towards a central point, like the mean. The lasso procedure encourages simple, sparse models. However, this model gives a r-squared value of 0.008298 and not optimal.

3. Prediction on the test set

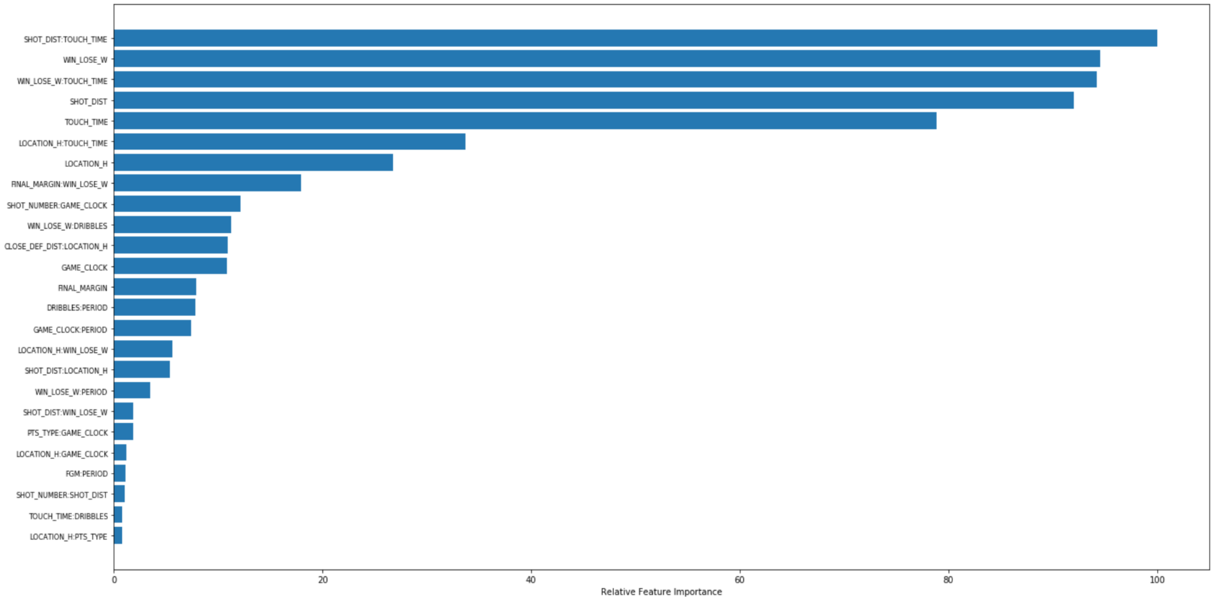
3.1. Logistic Regression

(a) The accuracy for the best model we built is 0.622. Our estimates of accuracy is 0.65. So our model is under-perform than my estimation.

(b) Our baseline model's accuracy is 0.605. Our model improved a little in terms of the accuracy. We also tested some other model's performance. 0.622 is the best accuracy so far. We observed that our best model does not significantly improved the accuracy. The reason might be FGM might need more complicated features to predict. Some player's FGM is not very sensitive to the defender and shooting distance. Furthermore, Different type of players like guard and forward may have their own comfortable shooting environments, making some features less informative.

3.2. Linear Regression

(a) We select best model of the stepwise regression with the largest r-squared value and the smallest mean squared error. Comparing to the baseline model, there is a slight improvement of the model when including some interaction terms in the model. The in-sample r-squared value increases from 0.00876 to 0.01210 and the out-of-sample increases from 0.01109 to 0.01323. The mean squared error decreases from 0.07415 to 0.07407.



(b) We computed feature importance in this model and found that the interaction between shot distance and touch time, winning the game, and interaction between winning the game and touch time, shot distance, and touch time are important in predicting the shot clock in the game. Comparing to the model we built before, this model is more accurate. However, the model does not explain much of the data set since there are still some parts of the information need to be considered, such as the player information, which we would discuss in the later section.

4. Discussion

4.1. Logistic Regression

Our logistic model can be used both for prediction and inference practically. This model can predict whether a shot will made or not based on game information, and also can study the relationship between the shot result and game technical statistics. We think this model can be hold up for a long time unless some rules of NBA and tactics of the game change. Otherwise, players' playing style of the game will not change much. We think a reasonable covariate should collect is the position the offensive player is playing. In particular, there are five positions in a basketball game, that is, PG, SG, SF, PF and C. If the offensive player is a center and he shots the ball very close to the basket, it is very likely that he will make this shot. But if a center shots the ball very far from the basket, it is very likely that he will miss this shot because center is generally more powerful inside. I will try other machine learning model like Random Forest or Decision Tree other than linear model if I were attack the same dataset again.

4.2. Linear Regression

Besides what was mentioned in the previous section which could also apply to linear regression, we would like to find an efficient way to include player information for future research. The player information should be quite influential of whom the closest defender is to the shot clock remaining. In common sense, the more powerful the defender is, the harder to make a shot. Due to the limitation of knowledge of handling unique player information, this part was omitted in the models we created, but should be definitely included in a more sophisticated model.