KOBE BRYANT SHOT PREDICTION ANALYSIS

November 13th , 2016

By

Dheeraj Gadhiraju

Table of contents

1. INTRODUCTION

Kobe Bryant shot prediction is one of the Kaggle project and I am required to predict the outcomes of Kobe scoring a basket or not based on the given data for my project

2. Problem statement and purpose of analysis

Kobe Bryant is a famous basketball player who took the retirement in April 2016. The problem statement was to find the basket shots made by Kobe Bryant through out his career depending upon various features related to the field demographics and player's style. Depending on the given data points related to diverse variables, the predictions are to be made, whether Kobe has made a shot or not.

This task can be achieved by fitting significant explanatory variables in the machine learning algorithms. The performance of the algorithms is compared by using different performance matrices like validation set approach and cross-validation. After a thorough comparison of the models, predictions will be made on the best performed model. The final output can be achieved in various phases.

- Preparing Data and Data visualization
- variable selection
- Model Building and comparison
- Prediction
- Summary

Let us dive into the data!

3. About Dataset

The dataset contains 20 years of Kobe's misses and swishes. The data contains the location and circumstances of every field goal attempted by Kobe Bryant during his 20 years' career. The data has 30,697 observations in total, out of which predictions are to be made on 5000 observations.

The dataset contains 25 variables out of which 1 variable is response variable (shot_made_flag) and 24 variables are independent.

The independent variables are self-explanatory:

- action_type
- combined_shot_type
- game_event_id
- game_id
- lat
- loc_xloc_y

- lon
- minutes_remaining
- period
- playoffs
- season
- opponent
- shot_id
- seconds_remaining
- shot_distance
- shot_made_flag (target)
- shot_type
- shot_zone_area
- shot_zone_basic
- shot_zone_range
- team_id
- team_name
- game_date
- matchup

```
> summary(Data)
              action_type    combined_shot_type game_event_id      game_id             lat             loc_x              loc_y              lon
 Jump Shot            :18880   Bank Shot:  141    Min.   :  2.0   Min.   :20000012   Min.   :33.25   Min.   :-250.000   Min.   :-44.00   Min.   :-118.5
 Layup Shot           : 2567   Dunk     : 1286    1st Qu.:110.0   1st Qu.:20500077   1st Qu.:33.88   1st Qu.: -68.000   1st Qu.:  4.00   1st Qu.:-118.3
 Driving Layup Shot   : 1978   Hook Shot:  153    Median :253.0   Median :20900354   Median :33.97   Median :   0.000   Median : 74.00   Median :-118.3
 Turnaround Jump Shot: 1057   Jump Shot:23485    Mean   :249.2   Mean   :24764066   Mean   :33.95   Mean   :   7.111   Mean   : 91.11   Mean   :-118.3
 Fadeaway Jump Shot   : 1048   Layup    : 5448    3rd Qu.:368.0   3rd Qu.:29600474   3rd Qu.:34.04   3rd Qu.:  95.000   3rd Qu.:160.00   3rd Qu.:-118.2
 Running Jump Shot    :  926   Tip Shot :  184    Max.   :659.0   Max.   :49900088   Max.   :34.09   Max.   : 248.000   Max.   :791.00   Max.   :-118.0
 (Other)              : 4241
 minutes_remaining     period         playoffs           season       seconds_remaining shot_distance   shot_made_flag          shot_type
 Min.   : 0.000    Min.   :1.000   Min.   :0.0000   2005-06: 2318   Min.   : 0.00     Min.   : 0.00   Min.   :0.000    2PT Field Goal:24271
 1st Qu.: 2.000    1st Qu.:1.000   1st Qu.:0.0000   2008-09: 2242   1st Qu.:13.00     1st Qu.: 5.00   1st Qu.:0.000    3PT Field Goal: 6426
 Median : 5.000    Median :3.000   Median :0.0000   2002-03: 2241   Median :28.00     Median :15.00   Median :0.000
 Mean   : 4.886    Mean   :2.519   Mean   :0.1466   2007-08: 2153   Mean   :28.37     Mean   :13.44   Mean   :0.446
 3rd Qu.: 8.000    3rd Qu.:3.000   3rd Qu.:0.0000   2009-10: 2080   3rd Qu.:43.00     3rd Qu.:21.00   3rd Qu.:1.000
 Max.   :11.000    Max.   :7.000   Max.   :1.0000   2001-02: 2028   Max.   :59.00     Max.   :79.00   Max.   :1.000
                                                    (Other):17635                                     NA's   :5000
              shot_zone_area               shot_zone_basic          shot_zone_range      team_id                   team_name          game_date
 Back Court(BC)        :   83   Above the Break 3    : 5620   16-24 ft.      :8315   Min.   :1.611e+09   Los Angeles Lakers:30697   2016-04-13:   50
 Center(C)             :13455   Backcourt            :   71   24+ ft.        :6275   1st Qu.:1.611e+09                              2002-11-07:   47
 Left Side Center(LC) : 4044   In The Paint (Non-RA): 4578   8-16 ft.       :6626   Median :1.611e+09                              2006-01-22:   46
 Left Side(L)          : 3751   Left Corner 3        :  280   Back Court Shot:  83   Mean   :1.611e+09                              2006-12-29:   45
 Right Side Center(RC): 4776   Mid-Range            :12625   Less Than 8 ft.:9398   3rd Qu.:1.611e+09                              2007-03-30:   44
 Right Side(R)         : 4588   Restricted Area      : 7136                          Max.   :1.611e+09                              2008-01-14:   44
                                Right Corner 3       :  387                                                                         (Other)   :30421
        matchup          opponent          shot_id
 LAL @ SAS  : 1020   SAS    : 1978   Min.   :    1
 LAL vs. SAS:  936   PHX    : 1781   1st Qu.: 7675
 LAL @ SAC  :  889   HOU    : 1666   Median :15349
 LAL vs. HOU:  878   SAC    : 1643   Mean   :15349
 LAL @ DEN  :  873   DEN    : 1642   3rd Qu.:23023
 LAL @ PHX  :  859   POR    : 1539   Max.   :30697
 (Other)    :25242   (Other):20448
```

Pairs plot:

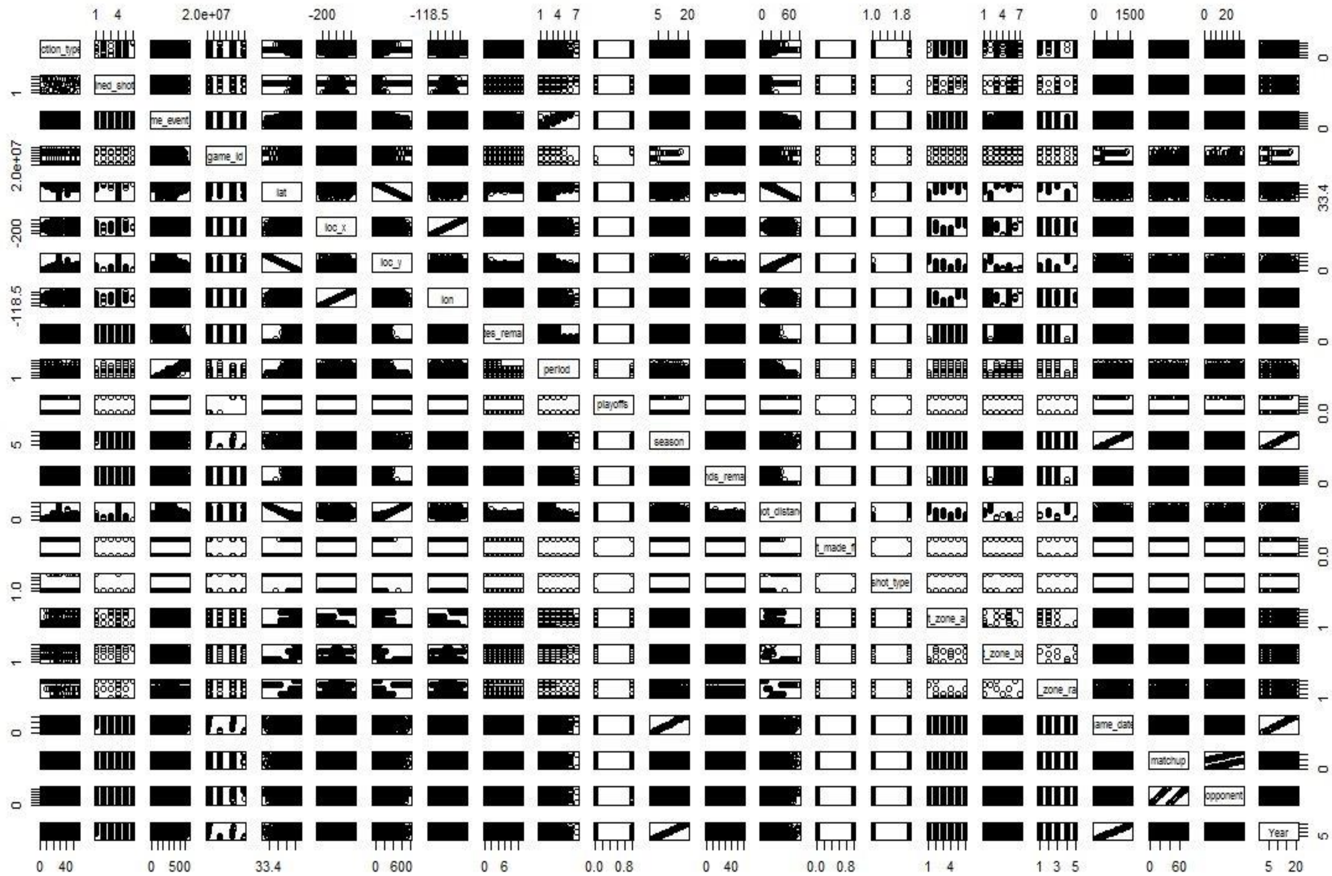


Fig.1 Pairs plot of the entire data

The above pairs plot indicates that there is a correlation between lon and loc_x, lat and loc_y. Conducting exploratory analysis will provide further insights about the data.

4. Preparing the Data

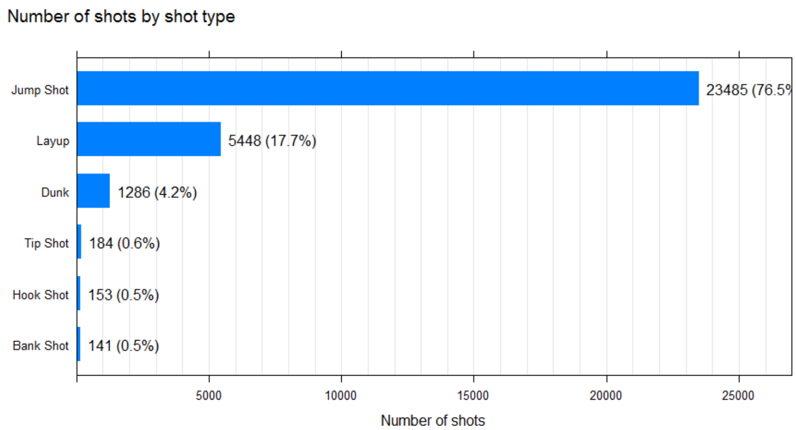
4.1 Initial Cleaning

The independent variables are self-explanatory and few variables like game_id, team_id, game_event_id, shot_id, game_date are related to different id's and unique dates. Also, Kobe has always played for LA Lakers and team_name is redundant and these variables are redundant when it comes to model building. Action_type and combined_shot-type represent the style of the player, while shooting the ball. Various action types are grouped into combined shot types and hence I have decided to remove the action_type variable too. Minutes remaining and seconds remaining represent the time remaining in each period. Since, both the variables represent time, I have decided to remove both the variables by creating a new variable, total-seconds_remaining in a period.

4.2 Data Visualization:

Let us explore the independent variables and their frequency.

4.2.1 Combined_shot_type:



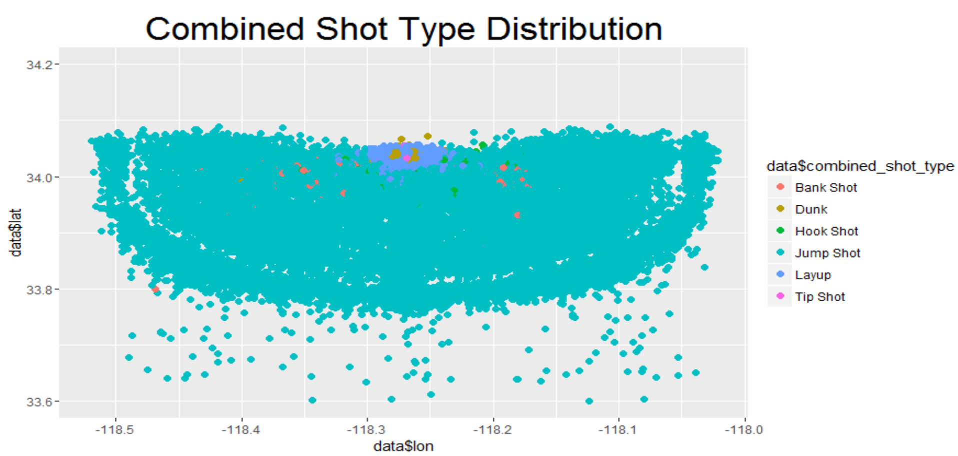


Fig. 1 and 2 above display the frequency of the shot types and the distribution on the court.

From the above figures, there is a clear difference in the choice of shots and Jump shots were attempted 76.5 % times. The second plot shows a better view of the shot types on the basketball court. The third plot shows the accuracy of the shots. Dunk shots were less attempted, but has the highest success rate.

4.2.2 lat and lon vs loc_x and loc_y

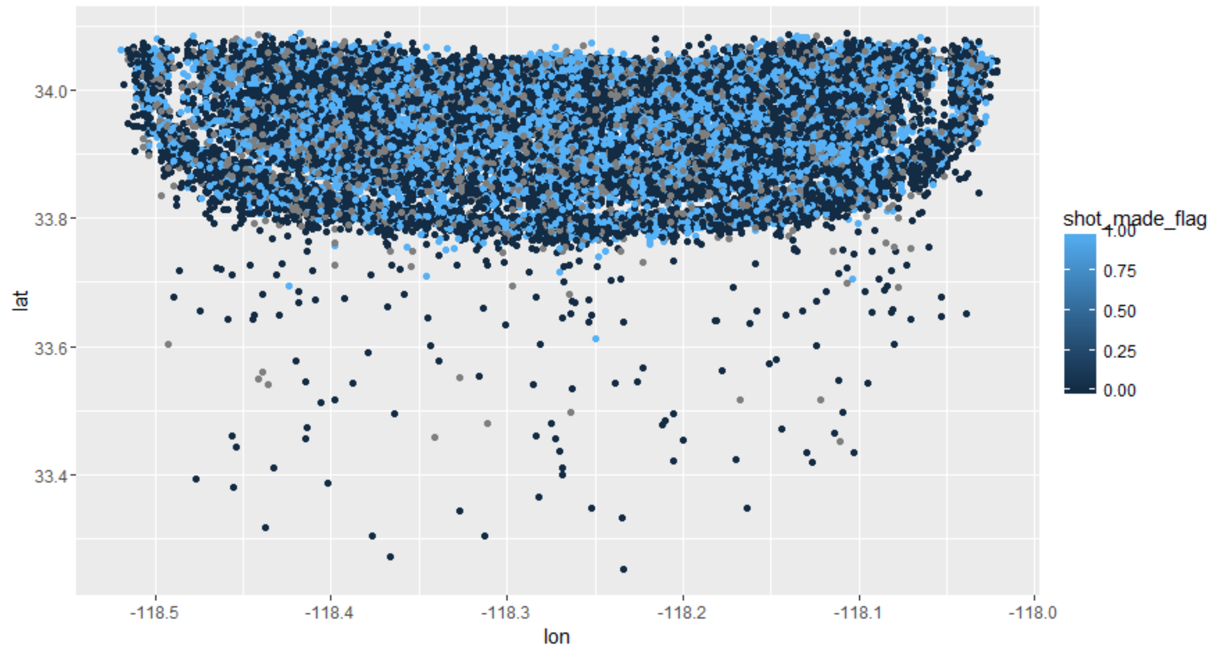


Fig. 3 Above plot represents the distribution of shots made using lat and lon

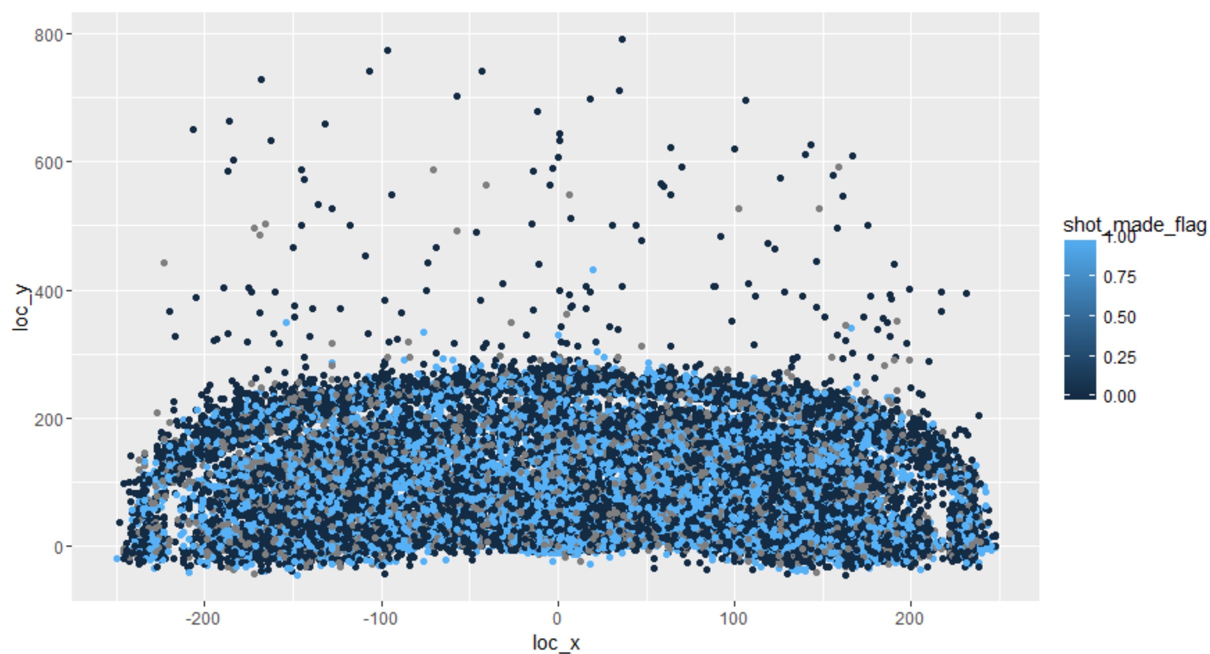


Fig. 4 Above plot represents the distribution of shots made using loc_x and loc_y

Both the plots are similar and they provide and represent the same data related to shots made on the basketball court. I have decided to remove lat and lon

4.2.3 Period

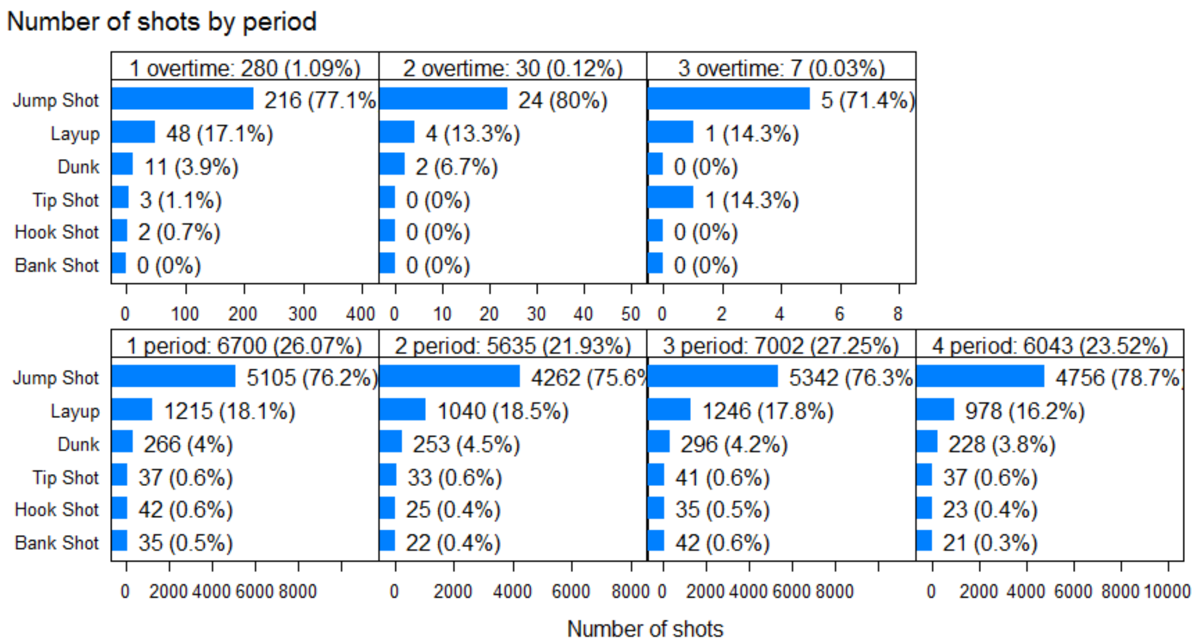


Fig. 5 Number of shots takes by period.

By looking at the above plots, there does not seem to be a much difference in shot_selection, between the 4 periods. Considering a basketball game, a game hardly enters a 3rd overtime period. The overtime periods give a clear sense that most of the overtime games were won/lost in the 1st over time.

4.2.4 Playoffs

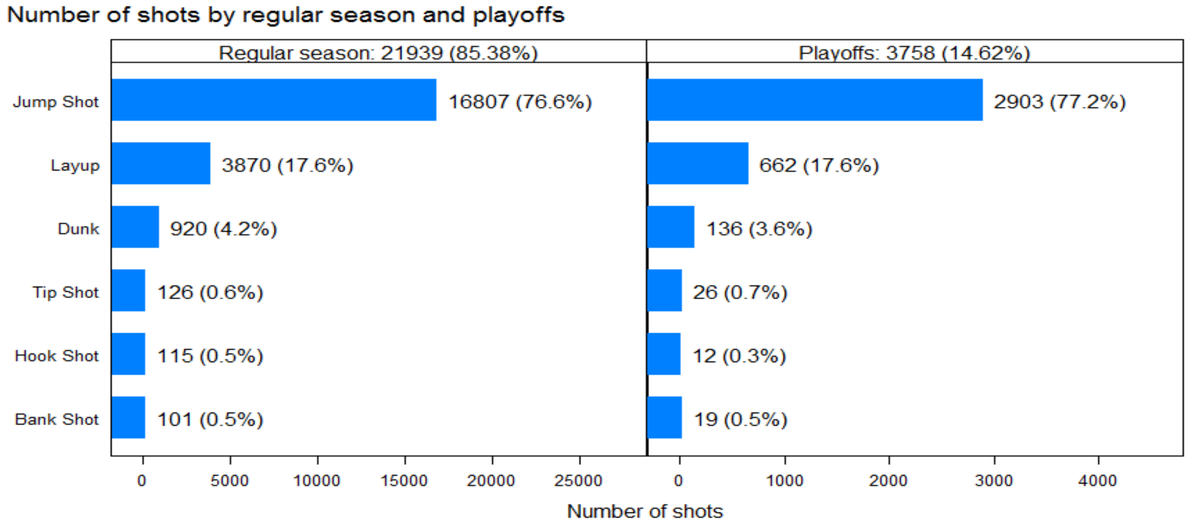


Fig. 6 Number of shots by regular season vs playoffs

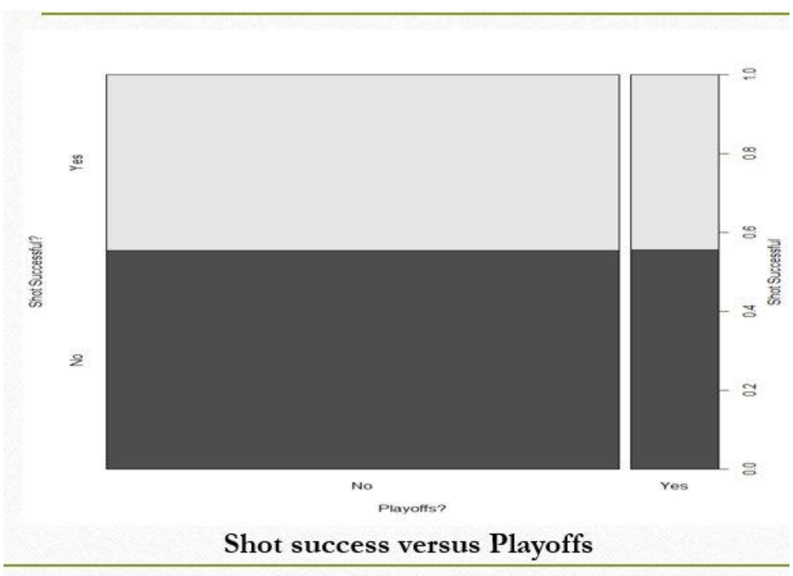


Fig. 7 Number of shots by shot_made_flag

The above plots show that the success rate doesn't change between playoffs and regular season games. But it makes sense that more shots were made in regular season.

4.2.5 Season

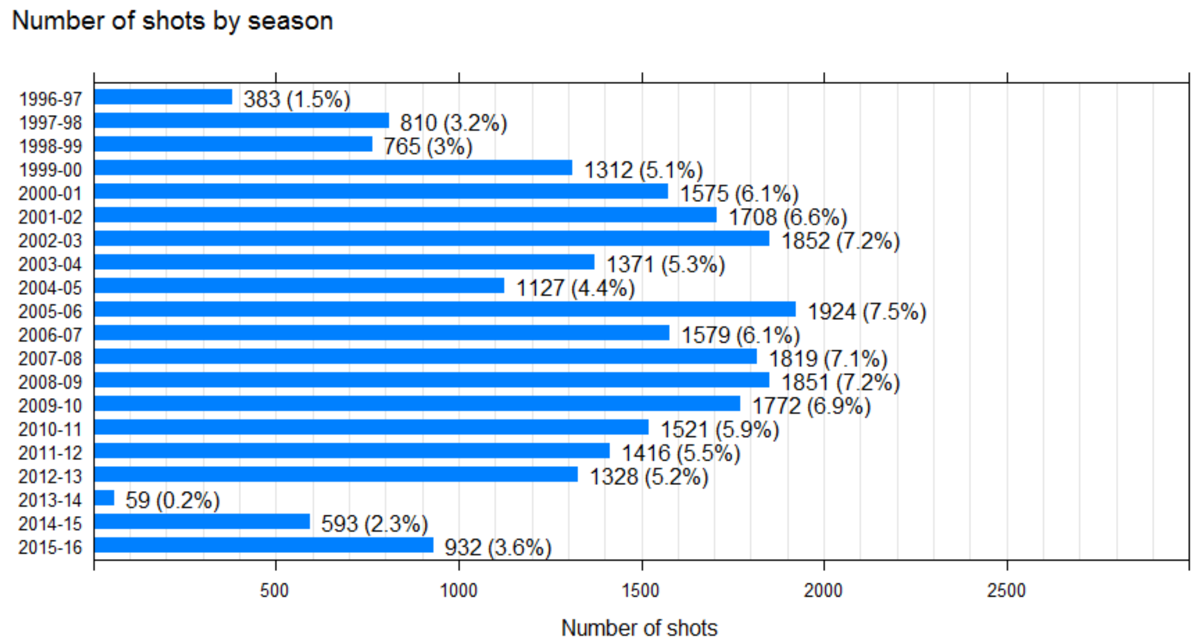


Fig. 8 Number of shots made by season

The above plot shows the % of shots made in each season. For example, Kobe suffered from left knee fracture in 2013-14 season and missed the season and hence he made only 59 shots in the entire season. 2004-2005 season was very miserable for LA Lakers.

4.2.6 Shot_zone_area, shot_zone_basic, shot_zone_range

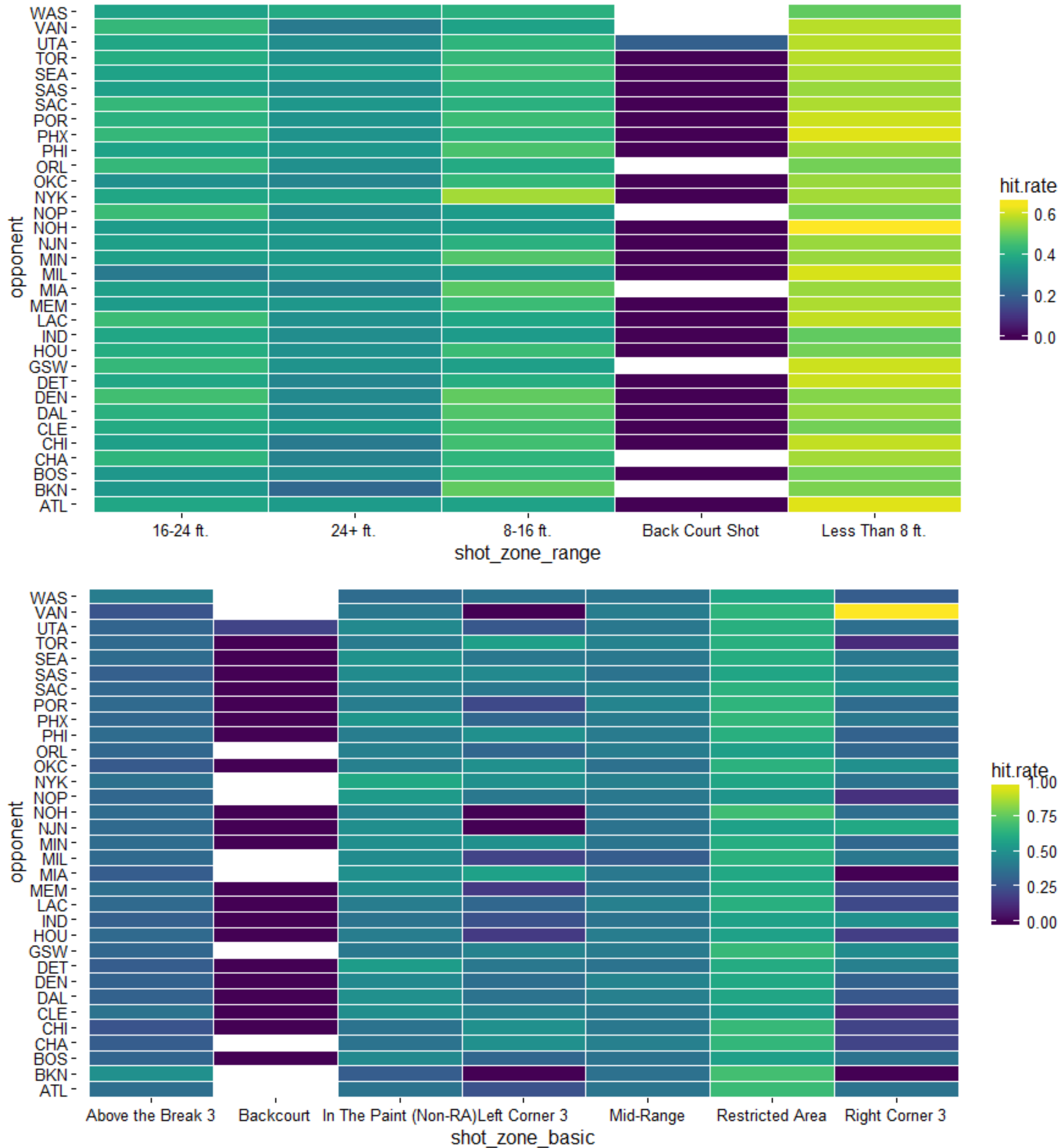


Fig.9 Heat maps of shot zone area, shot zone basic and shot zone range compared to opponents

The above plots show the success rates of the shots with respect to shot zone range, shot zone area, shot zone basic on opponents.
Kobe had a better success rate on center shot zone area and was bad with back court shots.
He had better hit rate when the shot zone range was less than 8ft.
He had a better hit rate when the shots were taken from restricted areas
The heat maps make sense logically.

4.2.7 Shot_Distance

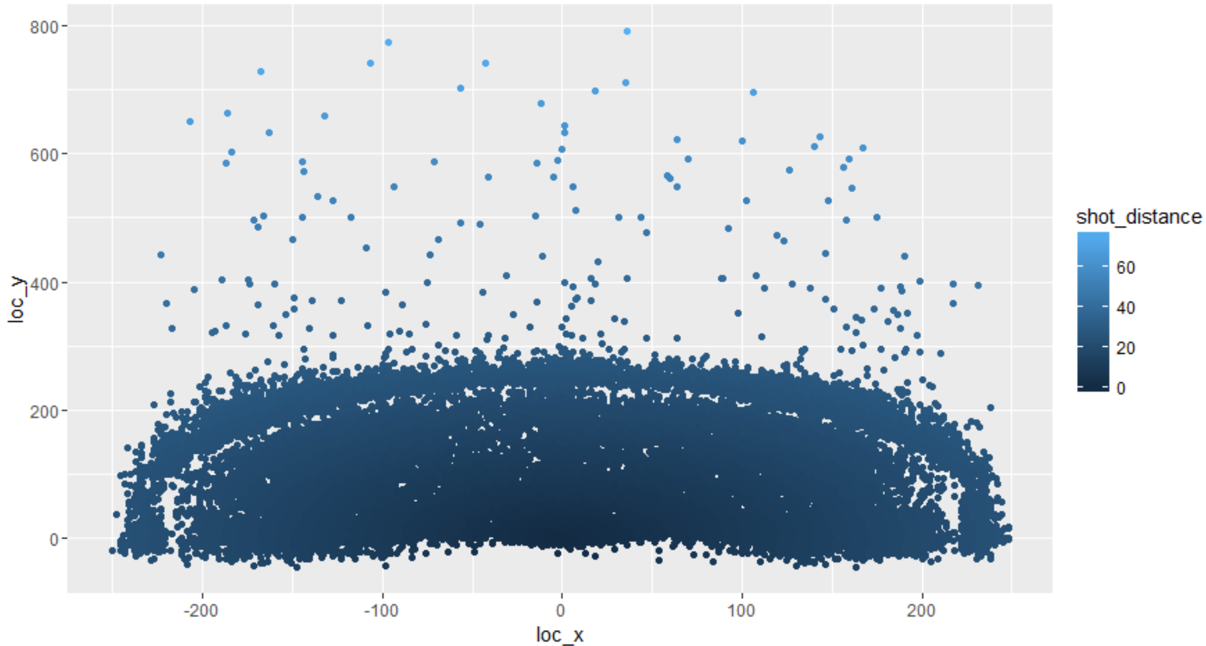


Fig. 10 Shots made by shot_distance

Above plot shows the shots made from various distances.

4.3 Final Cleaning

By intuition the variable opposition, the opponent team and the variable matchup has opponent as well as home game or away game. Since these both have similar data, I chose to keep matchup, since, it is giving more detail

5. Variable Selection

So far, the selected set of variables are combined_shot_type, loc_x, loc_y, period, playoffs, season, shot_distance, shot_made_flag, shot_type, shot_zone_area, shot_zone_basic, shot_zone_range, matchup, opponent, total_seconds_remaining.

A three way variable selection was performed using LASSO, stepwise and bestsubset regression methods.

5.1 Interaction terms

Often, while building models, two or more predictor variables might be insignificant and to capture an effect in the model, interaction terms were added in the variable selection methods to make better models and also, to increase the prediction accuracy. With a little theoretical knowledge and intuition, I have added the following interaction terms:

Period:total_seconds_remaining, shot_distance:shot_zone_range, loc_y:period and combined_shot_type:period are added to the variable selection methods.

The models were built by using the below formula:

shot_made_flag~combined_shot_type+loc_x+loc_y+period+playoffs+season+shot_distance+shot_type+shot_zone_area+shot_zone_basic+shot_zone_range+matchup+total_seconds_remaining+period:total_seconds_remaining+shot_distance:shot_zone_range+loc_y:period+combined_shot_type:period

5.2 Stepwise selection

To build a better model and to get the most significant variables, all the variables that are related to the dependent variable should be added to the model along with the interaction terms.

Stepwise regression is a combination of both forward and backward selection techniques. After each step, the independent variables in the model are checked if the significance is reduced below specified tolerance level. If the variable is found to be insignificant, the variable is removed. The achieved cross-validation error using the significant variables from Step wise selection is **22.98%.** The significant variables from Step Wise are listed below.

```
shot_made_flag ~ combined_shot_type + loc_x + loc_y + period +
    season + shot_distance + shot_zone_basic + shot_zone_range+
    total_seconds_remaining + shot_distance:shot_zone_range
```

5.3 Least Absolute Shrinkage and Selector Operator (LASSO)

LASSO is widely used as variable selection technique and also to get the best independent variables in order to increase the prediction accuracy. The achieved cross-validation error using the significant variables from Step wise selection is **23.1825%.** The significant variables from LASSO are listed below.

shot_made_flag ~ combined_shot_type + shot_zone_range + shot_distance + period + total_seconds_remaining + shot_distance:shot_zone_range

5.4 Subset Selection

Regular Subset selection performs an exhaustive search for the best variables among the independent variables. The model returns a best model of each size. The achieved cross-validation error using the significant variables from Step wise selection is **23.1791%.** The significant variables from Subset selection are listed below.

shot_made_flag ~ combined_shot_type:period + shot_distance:shot_zone_range + matchup + season + combined_shot_type + shot_zone_area + shot_zone_basic + shot_zone_range + shot_distance

5.5 Finalizing the set of variables

Comparing the three variable selection methods, the best cross-validation error achieved was by Stepwise regression with 22.98%. I have fitted the subset of significant variables from Stepwise regression to get the final set of significant variables. The important thing to note is that the interaction term shotdistance:shot_zone_range consistanctly remained significant. Adding the significant term made loc_y as significant.The Final set of variables are displayed below:

shot_made_flag ~ combined_shot_type + loc_y + period + season + total_seconds_remaining + shot_distance:shot_zone_range

```
glm(formula = step_formula, family = "binomial", data = modDat)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3774  -1.0287  -0.8747   1.2775   2.3709

Coefficients:
                                            Estimate Std. Error z value Pr(>|z|)
(Intercept)                               -1.067e+12  1.713e+12  -0.623 0.533311
combined_shot_typeDunk                     1.006e+00  3.535e-01   2.845 0.004443 **
combined_shot_typeHook Shot               -1.326e+00  3.566e-01  -3.718 0.000201 ***
combined_shot_typeJump Shot               -1.777e+00  3.053e-01  -5.822 5.83e-09 ***
combined_shot_typeLayup                   -1.294e+00  3.400e-01  -3.807 0.000141 ***
combined_shot_typeTip Shot                -2.171e+00  3.867e-01  -5.615 1.97e-08 ***
loc_x                                      1.232e-04  1.251e-04   0.985 0.324602
loc_y                                      7.919e-04  3.356e-04   2.360 0.018300 *
period                                    -4.349e-02  1.139e-02  -3.817 0.000135 ***
season1997-98                              3.196e-02  1.321e-01   0.242 0.808750
season1998-99                              2.082e-01  1.322e-01   1.576 0.115131
season1999-00                              2.123e-01  1.229e-01   1.727 0.084137 .
season2000-01                              2.345e-01  1.205e-01   1.945 0.051721 .
season2001-02                              1.803e-01  1.197e-01   1.506 0.131978
season2002-03                              1.396e-01  1.189e-01   1.175 0.240162
season2003-04                              1.042e-01  1.222e-01   0.853 0.393597
season2004-05                              1.544e-01  1.251e-01   1.234 0.217177
season2005-06                              3.108e-01  1.181e-01   2.632 0.008483 **
season2006-07                              2.933e-01  1.202e-01   2.439 0.014716 *
season2007-08                              2.713e-01  1.188e-01   2.285 0.022335 *
season2008-09                              2.969e-01  1.185e-01   2.505 0.012255 *
season2009-10                              2.545e-01  1.189e-01   2.140 0.032327 *
season2010-11                              2.640e-01  1.210e-01   2.181 0.029160 *
season2011-12                              2.093e-01  1.218e-01   1.719 0.085702 .
season2012-13                              2.940e-01  1.225e-01   2.400 0.016403 *
season2013-14                              7.714e-02  2.909e-01   0.265 0.790855
season2014-15                              2.345e-02  1.387e-01   0.169 0.865780
season2015-16                             -6.990e-04  1.292e-01  -0.005 0.995683
shot_distance                              8.117e-03  1.394e-02   0.582 0.560245
shot_zone_basicBackcourt                   1.493e+02  9.423e+04   0.002 0.998736
shot_zone_basicIn The Paint (Non-RA)       1.067e+12  1.713e+12   0.623 0.533311
shot_zone_basicLeft Corner 3               4.602e-03  1.588e-01   0.029 0.976875
shot_zone_basicMid-Range                   1.067e+12  1.713e+12   0.623 0.533311
shot_zone_basicRestricted Area             1.067e+12  1.713e+12   0.623 0.533311
shot_zone_basicRight Corner 3             -2.128e-01  1.434e-01  -1.484 0.137879
shot_zone_range24+ ft.                     1.067e+12  1.713e+12   0.623 0.533311
shot_zone_range8-16 ft.                    2.088e-01  3.010e-01   0.694 0.487836
shot_zone_rangeBack Court Shot             1.067e+12  1.713e+12   0.623 0.533311
shot_zone_rangeLess Than 8 ft.             4.909e-01  3.100e-01   1.583 0.113325
total_seconds_remaining                    1.876e-04  6.350e-05   2.954 0.003137 **
shot_distance:shot_zone_range24+ ft.      -1.627e-01  2.483e-02  -6.555 5.56e-11 ***
shot_distance:shot_zone_range8-16 ft.     -4.338e-03  1.877e-02  -0.231 0.817241
shot_distance:shot_zone_rangeBack Court Shot -3.637e+01  7.275e+03  -0.005 0.996012
shot_distance:shot_zone_rangeLess Than 8 ft. -7.800e-02  3.231e-02  -2.414 0.015777 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 35325  on 25696  degrees of freedom
Residual deviance: 33270  on 25653  degrees of freedom
```

6. Model building and Selection

For model building evaluation purposes, 25% of data was reserved as validation data.

6.1 Logistic regression

Logistic regression is widely used for classification problems. Since the dependent variable is categorical in nature, ordinary least squares regression cannot be used as the assumptions of the normality of the responses will be violated and the probabilities can only lie between 0 and 1. The distribution is binomial in nature and the probability of Kobe Bryant scoring a basket, is to be modeled as the functions of independent variables. The function here is not going to be linear, but by using a non-linear transformation technique, log-odds is applied to the dependent variable and the predictions can lie anywhere between -∞ to +∞.

$$\ln\left(\frac{\hat{p}}{1-\hat{p}}\right) = B_0 + B_1 X$$

So, that the probabilities lie between 0 and 1. The logits can lie anywhere between -∞ to +∞. The probabilities can be obtained by transforming the estimated logit equation back into the probability form.

$$\hat{p} = \frac{\exp(B_0 + B_1 X)}{1+\exp(B_0 + B_1 x)} = \frac{e^{B_0+B_1x}}{1+e^{B_0+B_1x}}$$

In the above equations, x is the independent variable and to form an equation for our current model, the intercept and constants can be added to the equation along with the independent variables.

The model is evaluated based on the misclassification rate using validation set approach and 10-fold cross validation approach.

	True No	**True Yes**
Predicted No	2984	1991
Predicted Yes	537	913

Table 1. *Confusion matrix of the logistic regression model*

Validation set error: 39.34%

True Positive rate: 31.43%

True Negative Rate: 84.74%

10- fold Cross-validation error rate:38.28%

The model summary indicates that combined_shot_type, loc_y, period, season, total_seconds_remaining, and the interaction term, shot_distance:shot_zone_range are significant.

Model Summary:

```
> summary(logit.fit)

Call:
glm(formula = f, family = binomial, data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3577  -1.0302  -0.8727   1.2782   2.6851

Coefficients:
                                              Estimate Std. Error z value Pr(>|z|)
(Intercept)                                  1.550e+00  3.153e-01   4.917 8.80e-07 ***
combined_shot_typeDunk                       8.626e-01  3.152e-01   2.737 0.006204 **
combined_shot_typeHook Shot                 -1.403e+00  3.366e-01  -4.169 3.06e-05 ***
combined_shot_typeJump Shot                 -1.790e+00  2.707e-01  -6.613 3.77e-11 ***
combined_shot_typeLayup                     -1.377e+00  2.847e-01  -4.837 1.32e-06 ***
combined_shot_typeTip Shot                  -2.303e+00  3.495e-01  -6.591 4.38e-11 ***
loc_y                                        4.554e-04  3.084e-04   1.476 0.139852
period                                      -3.544e-02  1.318e-02  -2.690 0.007153 **
season1997-98                               -1.183e-01  1.556e-01  -0.761 0.446898
season1998-99                                1.302e-01  1.540e-01   0.845 0.397900
season1999-00                                1.846e-01  1.441e-01   1.281 0.200150
season2000-01                                1.555e-01  1.402e-01   1.109 0.267306
season2001-02                                5.135e-02  1.398e-01   0.367 0.713425
season2002-03                                7.910e-02  1.389e-01   0.570 0.569014
season2003-04                                3.745e-02  1.427e-01   0.262 0.792981
season2004-05                                1.241e-01  1.460e-01   0.850 0.395315
season2005-06                                2.023e-01  1.380e-01   1.466 0.142663
season2006-07                                2.105e-01  1.399e-01   1.504 0.132528
season2007-08                                2.082e-01  1.386e-01   1.502 0.133057
season2008-09                                2.343e-01  1.384e-01   1.692 0.090578 .
season2009-10                                1.590e-01  1.389e-01   1.145 0.252247
season2010-11                                2.176e-01  1.406e-01   1.547 0.121773
season2011-12                                1.226e-01  1.417e-01   0.865 0.386942
season2012-13                                1.945e-01  1.426e-01   1.364 0.172548
season2013-14                               -3.201e-01  3.574e-01  -0.896 0.370481
season2014-15                               -2.843e-02  1.625e-01  -0.175 0.861131
season2015-16                               -9.454e-02  1.505e-01  -0.628 0.529995
total_seconds_remaining                      2.586e-04  7.329e-05   3.528 0.000419 ***
shot_distance:shot_zone_range16-24 ft.      -1.944e-02  6.115e-03  -3.180 0.001474 **
shot_distance:shot_zone_range24+ ft.        -2.710e-02  4.987e-03  -5.434 5.52e-08 ***
shot_distance:shot_zone_range8-16 ft.       -1.691e-02  8.799e-03  -1.921 0.054693 .
shot_distance:shot_zone_rangeBack Court Shot -8.173e-02  2.206e-02  -3.706 0.000211 ***
shot_distance:shot_zone_rangeLess Than 8 ft. -4.966e-02  1.904e-02  -2.608 0.009100 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 26476  on 19271  degrees of freedom
Residual deviance: 24963  on 19239  degrees of freedom
AIC: 25029

Number of Fisher Scoring iterations: 6
```

6.2 Linear Discriminant Analysis:

LDA assumes that the distribution of the data is Gaussian, unlike Logistic regression. LDA can be used for both categorical and continuous response type variables. LDA, also, assumes that the attributes have same variance. If the response variable has more than 2 classes and if, the classes are separated well, LDA is preferred over Logistic regression. LDA makes predictions by estimating the probability of the new input observations belong to a class. LDA uses Bayes' theorem to obtain probability.

The model is evaluated based on the misclassification rate using validation set approach and 10-fold cross validation approach.

	True No	True Yes
Predicted No	2968	553
Predicted Yes	1992	912

Table 2. *Confusion matrix of the LDA model*

Validation set error: 39.61%

True Positive rate: 62.25%

True Negative Rate: 59.83%

10- fold Cross-validation error rate:38.52%

Model Summary:

```
Call:
lda(f, data = train)

Prior probabilities of groups:
        0         1 
0.5557804 0.4442196 

Group means:
  combined_shot_typeDunk combined_shot_typeHook Shot combined_shot_typeJump Shot combined_shot_typeLayup combined_shot_typeTip Shot
0            0.005414994                0.004481374                   0.8458594               0.1360284                0.006628699
1            0.084102324                0.006074057                   0.6716505               0.2248569                0.004321925
      loc_y   period season1997-98 season1998-99 season1999-00 season2000-01 season2001-02 season2002-03 season2003-04 season2004-05
0 103.65139 2.551769    0.03155634    0.02884885    0.04612081    0.06096536    0.06572682    0.07188871    0.05377649    0.04350668
1  76.11914 2.485457    0.02745006    0.03107114    0.05162948    0.06634739    0.06588015    0.07031889    0.05256395    0.04380329
  season2005-06 season2006-07 season2007-08 season2008-09 season2009-10 season2010-11 season2011-12 season2012-13 season2013-14
0    0.07431612    0.06180562    0.06834096    0.06880777    0.06862104    0.05993838    0.05797778    0.05218934   0.002614135
1    0.07405677    0.06436164    0.07557528    0.07604252    0.06938442    0.06190866    0.05338161    0.05291438   0.001635323
  season2014-15 season2015-16 total_seconds_remaining shot_distance:shot_zone_range16-24 ft. shot_distance:shot_zone_range24+ ft.
0    0.02492764    0.04322659                315.8220                               5.422556                             6.252731
1    0.01903983    0.02861815                330.6657                               4.530078                             3.743838
  shot_distance:shot_zone_range8-16 ft. shot_distance:shot_zone_rangeBack Court Shot shot_distance:shot_zone_rangeLess Than 8 ft.
0                              2.660723                                  0.301372421                                    0.5290823
1                              2.551805                                  0.005022778                                    0.5269244

Coefficients of linear discriminants:
                                                     LD1
combined_shot_typeDunk                      0.5709413077
combined_shot_typeHook Shot                -2.2663537756
combined_shot_typeJump Shot                -3.0160126723
combined_shot_typeLayup                    -2.1922036071
combined_shot_typeTip Shot                 -3.9435536987
loc_y                                       0.0008462019
period                                     -0.0629001216
season1997-98                              -0.1965273264
season1998-99                               0.2302611269
season1999-00                               0.3242342712
season2000-01                               0.2720976673
season2001-02                               0.0891445554
season2002-03                               0.1382275631
season2003-04                               0.0645065475
season2004-05                               0.2175487300
season2005-06                               0.3590379631
season2006-07                               0.3727698610
season2007-08                               0.3665747964
season2008-09                               0.4169185317
season2009-10                               0.2822249428
season2010-11                               0.3864372278
season2011-12                               0.2152379265
season2012-13                               0.3441538212
season2013-14                              -0.5635524656
season2014-15                              -0.0549014471
season2015-16                              -0.1683580990
total_seconds_remaining                     0.0004650647
shot_distance:shot_zone_range16-24 ft.     -0.0352159938
shot_distance:shot_zone_range24+ ft.       -0.0488076392
shot_distance:shot_zone_range8-16 ft.      -0.0298890460
shot_distance:shot_zone_rangeBack Court Shot -0.0677866695
shot_distance:shot_zone_rangeLess Than 8 ft. -0.0893702238
> 
```

6.3 Support Vector Machine (SVM)

Considering the response variable, shot_made_flag, it has 2 classes. SVM's are based on the concept of decision planes that define decision boundaries. SVM finds a hyerplane that separates both the classes as wide as possible. SVM is a non-probabilistic classifier when it comes to prediction. Out of n separating planes, SVM chooses a hyperplane which separates both the classes as wide as possible. SVM's work good when there is a clear margin of separation. It is also effective in high dimensional space. However, when the data set is large and when the target classes are overlapping each other SVM doen't perform well.

Model Summary:

```
> summary(svm_model)

Call:
svm(formula = f, data = train, kernel = "radial")


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  1
      gamma:  0.03030303

Number of Support Vectors:  15443

 ( 7472 7971 )


Number of Classes:  2

Levels:
 0 1
```

	True No	True Yes
Predicted No	2981	1990
Predicted Yes	540	914

Table 3. *Confusion matrix of the SVM model*

Validation set error: 39.37%

True Positive rate: 31.47%

True Negative Rate: 84.66%

10- fold Cross-validation error rate:38.93%

6.4 Bagging

Bagging is an ensemble technique in which multiple classifiers are trained using random sampling. In this way Bagging tries to reduce variance and helps avoid overfitting. It is a special type of model averaging approach. It aims in reducing the variance. Decision trees are usually sensitive to specific data and if the training data is changed, often, the resulting decision tree might be quite unique and in turn predictions might be different too.

Generally, we are not provided with multiple training sets, so, instead we bootstrap by taking repeated random samples from the same training set. Bagging is the application of Bootstrap technique to high variance decision trees. For each test observation, the predicted class is recorded and n trees and by applying majority vote, the overall prediction would be the most commonly occurring class. For example, in 10 decision trees, if an observation was predicted as 'yes', It applies the majority rule and it predicts as 'yes'.

```
> bag.mod

Call:
 randomForest(formula = as.factor(shot_made_flag) ~ combined_shot_type +      loc_y + period + season + total_seconds_remaining + shot_distance:shot_zone_range,      data = train, mtry = 6, importance = TRUE)

               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 6

        OOB estimate of  error rate: 42.61%
Confusion matrix:
     0    1 class.error
0 7495 3216   0.3002521
1 4995 3566   0.5834599
```

	True No	True Yes
Predicted No	2968	1979
Predicted Yes	553	925

Table 4. *Confusion matrix of the Bagging model*

Validation set error: 42.75%

True Positive rate: 39.97%

True Negative Rate: 71.40%

OOB error: 42.61%

10- fold Cross-validation error rate:40.86%

6.5 Random Forest

Random forests provide an improvement over Bagging. Bagging takes all the predictors available to form decision trees. And, these decision trees might have a lot of similarities and in turn might have high correlation in their predictions. Combining predictions work better if predictions from sub-models are un-correlated or weakly co-related.

Random Forest almost works like Bagging, except the trees are also built on randomly chosen predictors instead of choosing all the predictors. The training model can look through all the predictors and data to select the optimal split point. The number of predictors to be searched at each split point must be specified while building a classifier. By using a tune function, the optimal no. of predictors to be chosen while building a tree can be obtained.

Model Summary:

```
> frst.one

Call:
 randomForest(formula = f, data = train, mtry = 1, importance = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 1

        OOB estimate of  error rate: 38.36%
Confusion matrix:
     0    1 class.error
0 9127 1584   0.1478854
1 5809 2752   0.6785422
```

	True No	True Yes
Predicted No	2968	1979
Predicted Yes	553	925

Table 5. *Confusion matrix of the Random Forest model*

Validation set error: 39.40%

True Positive rate: 31.85%

True Negative Rate: 84.29%

OOB error: 38.36%

10- fold Cross-validation error rate:38.64%

6.6 Model Comparison

Model	VSA	True Positive	True Negative	10-fold	OOB*
Logistic	39.34%	31.43%	84.74%	38.28%	
LDA	39.61%	62.65%	59.83%	38.52%	
Bagging	42.75%	39.97%	71.40%	42.61%*	40.86%
RandomForest	39.40%	31.85%	84.29%	38.64%*	38.36%
SVM	39.37%	31.47%	84.66%	38.93%	

Table 6. *Comparison of models by Validation set and Cross-Validation error rates*

From the above table, there isn't a lot of difference in the performance of the models except for Bagging. However, Logistic regression stands out compared to the other models, having the least error rate in both Validation set and Cross-Validation.

6.6 Prediction

The prediction is to be made on the 5000 unlabeled observations in the response variable. After comparing the models with cross validation error and validation set error, Logistic Regression was found to be the best model and the predictions were made on the test data set. The predictions were submitted to Kaggle to find out the log loss and the achieved log loss was 0.64862

7 Conclusion

A proper exploratory analysis was conducted on all the independent variables by using different data visualization plots. The best subset of significant variables was selected using a 3- way variable selection methods, LASSO, stepwise and, regular subset selection. The interaction term shot_distance:shot_zone_range stood out as a significant term. Along, with the interaction term, combined_shot_type, loc_y, period and total_seconds_remaining were the final chosen predictors for model building.The best model was selected based on the lowest misclassification rate. Logistic regression has the lowest misclassification rate.

References:

https://ncss-wpengine.netdna-ssl.com/wp-content/themes/ncss/pdf/Procedures/NCSS/Stepwise_Regression.pdf

http://statweb.stanford.edu/~tibs/lasso/lasso.pdf

https://cran.r-project.org/web/packages/leaps/leaps.pdf

http://web.pdx.edu/~newsomj/da2/ho_logistic.pdf

https://www.reddit.com/r/MachineLearning/comments/15zrpp/please_explain_support_vector_machines_svm_like_i/

http://www.statsoft.com/Textbook/Support-Vector-Machines

http://www.slideshare.net/pierluca.lanzi/machine-learning-and-data-mining-16-classifiers-ensembles

https://lagunita.stanford.edu/c4x/HumanitiesScience/StatLearning/asset/trees.pdf

https://class.stanford.edu/c4x/HumanitiesScience/StatLearning/asset/cv_boot.pdf