6372 Group Project 2

Collaborators: Shon Mohsin, Daniel Serna, William Trevino

Introduction

In this paper, our team will conduct a thorough analysis of Kobe Bryant’s 20 year shot record. We will perform an exploratory data analysis to prepare our input for effective processing prior to building out our models. An explanation of our data cleansing and outlier identification will be presented to the reader in order to aid with reproducibility for future research. We will conclude by presenting a model for predicting whether Kobe Bryant made or missed a shot based on a number of explanatory factors. An explanation of our model assumptions, construction, and analysis will be presented to the reader to aid in further research and discussion.

Data Description

Our input data consists of 25,697 records which account for Kobe Bryant's 20 year shot record. Each record contains 28 explanatory data points which we will analyse to predict the response - whether Kobe Bryant missed or made the shot (shot_made_flag). An explanation of the data points can be found in Appendix 1 - Variable Description.

Exploratory Data Analysis

At a high level, we found the game date field not be very useful as it is a large categorical variable. Instead we decided to target the season and account for playoff effects.

Outliers

From initial investigations of the data and histograms, it does not appear we have significant outliers that need to be removed. However, it does look like action_type, combined_shot_type,lat, loc_y, and shot_distance data points might benefit from transformations. Please see Figure 1 in the appendix.

Transformations

As mentioned in the Outliers section, it appears the following fields might benefit from a transformation:action_type, combined_shot_type,lat, loc_y, and shot_distance. Performing a log transformation on the numeric fields lat, loc_y, and shot_distance had some benefits for the loc_y datapoint as seen below. However, log transformations of lat and shot_distance had no benefit. Please note, to deal with negative values, we added a constant during our log transformation of lat and loc_y. To deal with 0 values in shot_distance we have added a small value of 0.001 to the data point. Please see Figure 2 in the Appendix.

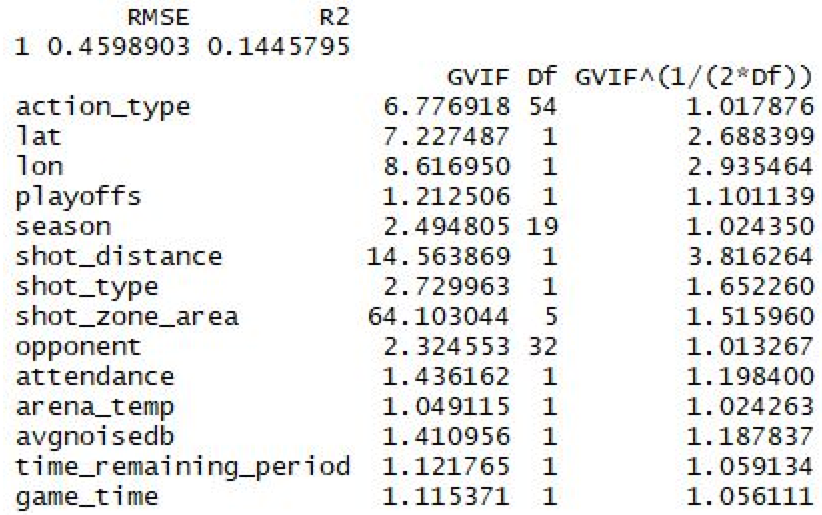
We added 2 additional variables to better model the time at which shots were taken. We combined minutes_remaining and seconds_remaining and applied some basic math to create a field called time_remaing_period which indicates the total seconds remaining in a period at which a shot was taken. Similarly, we combined our time_remaining_period field with the period field, applied some basic math to create a game_time field which indicates the seconds into a game at which a shot was taken.

Multicollinearity

As we conducted our variance inflation factor analysis, we received the following error: "there are aliased coefficients in the model." Upon further investigation, we discovered this error occurs when variables have perfect collinearity with each other. So we isolated the problem variables for removal and our results are as follows:

The action_type and combined_shot_type fields displayed perfect collinearity with each other, so we chose to drop combined_shot_type from our model since it has the lesser amount of information. Shot_zone_range and shot_zone_basic fields also displayed perfect collinearity with each other so we chose to drop shot_zone_basic from our model since it has the lesser amount of information. Lat and Lon displayed perfect collinearity with loc_x and loc_y respectively, so we chose to drop loc_x and loc_y since they contain the lesser amounts of information. Shot_zone_range and shot_zone_area also displayed perfect collinearity with each other and we will isolate one of these variables for removal after further VIF analysis.

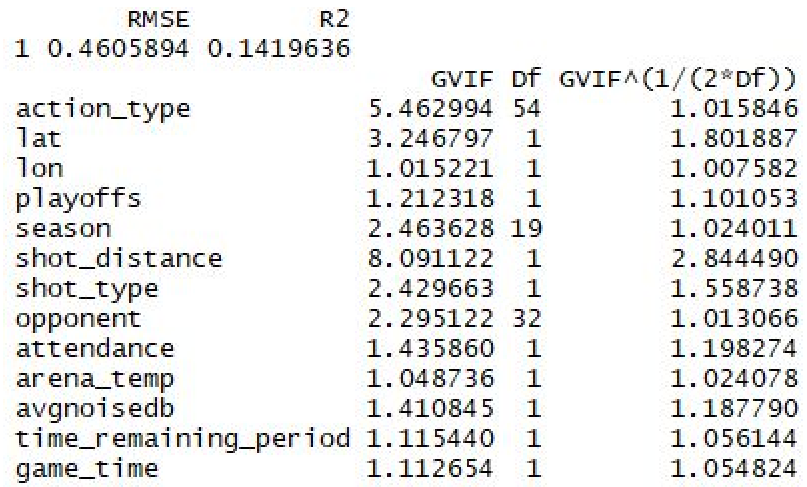
After the removal of the above mentioned variables, we are able to successfully run a VIF analysis. Please see Figure 3 for results.



```
       RMSE        R2
1 0.4598903 0.1445795
                            GVIF Df GVIF^(1/(2*Df))
action_type             6.776918 54        1.017876
lat                     7.227487  1        2.688399
lon                     8.616950  1        2.935464
playoffs                1.212506  1        1.101139
season                  2.494805 19        1.024350
shot_distance          14.563869  1        3.816264
shot_type               2.729963  1        1.652260
shot_zone_area         64.103044  5        1.515960
opponent                2.324553 32        1.013267
attendance              1.436162  1        1.198400
arena_temp              1.049115  1        1.024263
avgnoisedb              1.410956  1        1.187837
time_remaining_period   1.121765  1        1.059134
game_time               1.115371  1        1.056111
```

Figure 3

We are going to view a VIF > 5 as a problem data point and target the variable for removal. First we will attempt to remove shot_zone_area. After the removal of shot_zone_area we are provided results from our VIF analysis seen in Figure 4.

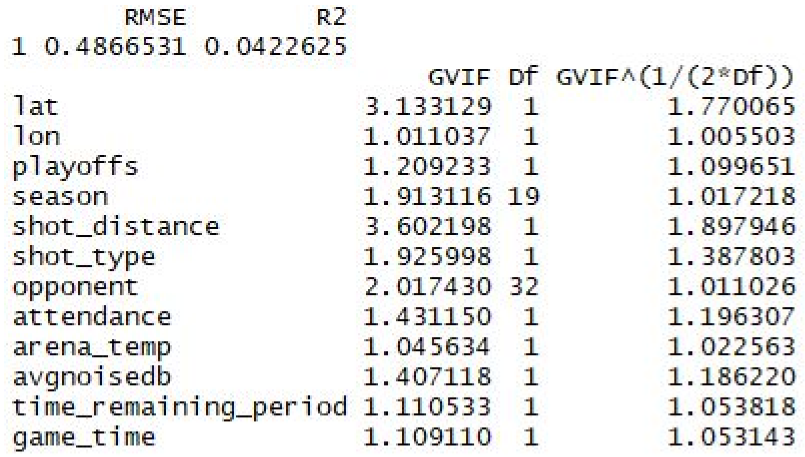


```
        RMSE        R2
1 0.4605894 0.1419636
                           GVIF Df GVIF^(1/(2*Df))
action_type            5.462994 54        1.015846
lat                    3.246797  1        1.801887
lon                    1.015221  1        1.007582
playoffs               1.212318  1        1.101053
season                 2.463628 19        1.024011
shot_distance          8.091122  1        2.844490
shot_type              2.429663  1        1.558738
opponent               2.295122 32        1.013066
attendance             1.435860  1        1.198274
arena_temp             1.048736  1        1.024078
avgnoisedb             1.410845  1        1.187790
time_remaining_period 1.115440  1        1.056144
game_time              1.112654  1        1.054824
```

Figure 4

As we can see, action_type still has a VIF > 5 which may be problematic, however we chose to leave this in our model and we will run additional analysis with and without this data point for comparison.

If we remove action_type, we receive the output from our VIF analysis seen in Figure 5.



```
        RMSE        R2
1 0.4866531 0.0422625
                           GVIF Df GVIF^(1/(2*Df))
lat                    3.133129  1        1.770065
lon                    1.011037  1        1.005503
playoffs               1.209233  1        1.099651
season                 1.913116 19        1.017218
shot_distance          3.602198  1        1.897946
shot_type              1.925998  1        1.387803
opponent               2.017430 32        1.011026
attendance             1.431150  1        1.196307
arena_temp             1.045634  1        1.022563
avgnoisedb             1.407118  1        1.186220
time_remaining_period 1.110533  1        1.053818
game_time              1.109110  1        1.053143
```

Figure 5

After removing action_type, the R^2 has dropped significantly. Because of this, we will not remove this datapoint from our model. We will test with and without this datapoint for comparison in further analysis.

Models

Proposition Analysis

Odds Shot Distance

Running a logistic model with shot_distance as the only explanatory variable for shot_made_flag produces the results seen in Figure 6 of the Appendix.

The parameter estimate for shot_distance is a negative value. This indicates the odds of Kobe successfully making a shot decreases as shot_distance increases. At a p-value < 0.0001, there is strong statistical evidence to indicate this is the case.

Probability Shot Distance

By running a regression analysis on a model with shot_distance as the only explanatory variable for shot_made_flag, we are provided the results seen in Figure 7 of the Appendix.

The parameter estimate for shot_made_probability is a negative number. This indicates the probability of Kobe successfully making a shot decreases as shot_distance increases. At a p-value < 0.0001, there is strong statistical evidence to indicate this is the case.

Odds Shot Distance in Playoff

To determine if the odds of Kobe making a shot related to shot_distance is different between playoff and non-playoff games, we ran two logistic regression analyses on two datasets. The logistic regression analysis for non playoff games is seen in Figure 8 of the Appendix and the logistic regression analysis for playoff games is seen in Figure 9 of the Appendix.

In non-playoff games, the parameter estimate for shot_distance is -0.0448 whereas in playoff games the parameter estimate for shot_distance is -0.0402.The odds of Kobe making a shot when taking shot_distance into account is higher in playoff games (-0.0402 < -0.0448).

In addition to the above analysis, we also decided to conduct an odds ratio analysis to test if there is difference in shot_made percentage between playoff and non-playoff games. The results are shown in Figure 10 of the Appendix.

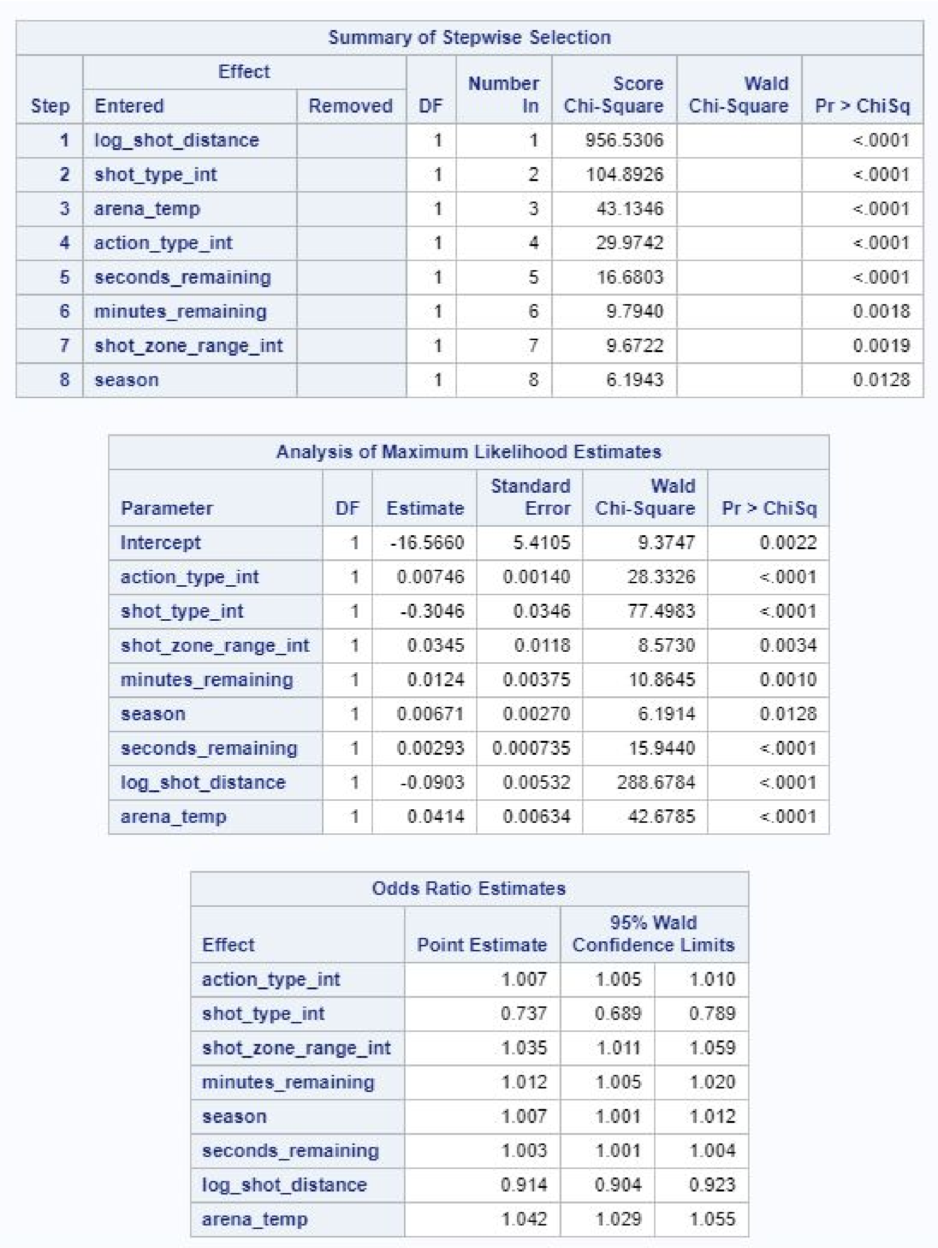
As we can see, 1 is a possible value in the 95% confidence intervals, thus we fail to reject the null hypothesis that there is a difference in shot_made odds between playoff and non-playoff games. Although we see a slight decrease in shot_made percentage for playoff games, it is not statistically significant.

Predictive Analysis

Logistic Regression Model

Please see Figure 11 below for the results of our logistic regression model. We initially started with a model stripped of many variables. However, we were not achieving effective success rates. We decided to try again by including all variables and proceed with a stepwise elimination to achieve better success rates. After significant efforts, we were able to increase our success rate to 60.8% as seen in the below classification table. Our final logistic regression model is:

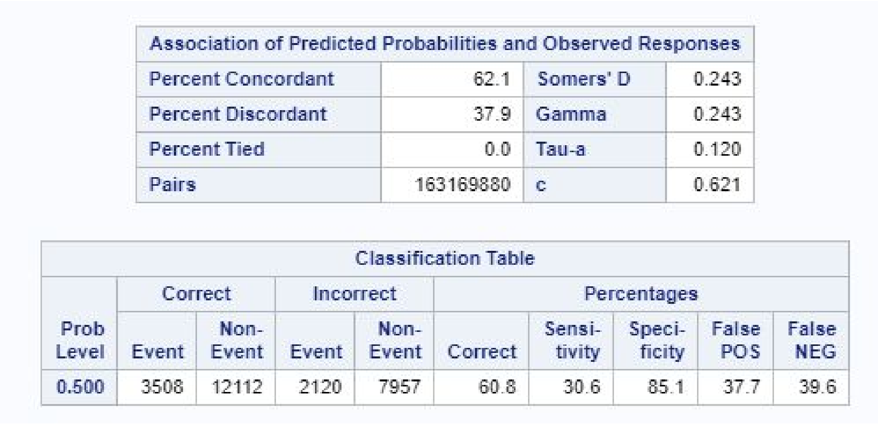
```
model shot_made_flag(event="1") = action_type_int shot_type_int shot_zone_range_int
opponent_int log_lat log_loc_y lon minutes_remaining playoffs season
seconds_remaining log_shot_distance arena_temp
```



Summary of Stepwise Selection							
	Effect						
Step	Entered	Removed	DF	Number In	Score Chi-Square	Wald Chi-Square	Pr > ChiSq
1	log_shot_distance		1	1	956.5306		<.0001
2	shot_type_int		1	2	104.8926		<.0001
3	arena_temp		1	3	43.1346		<.0001
4	action_type_int		1	4	29.9742		<.0001
5	seconds_remaining		1	5	16.6803		<.0001
6	minutes_remaining		1	6	9.7940		0.0018
7	shot_zone_range_int		1	7	9.6722		0.0019
8	season		1	8	6.1943		0.0128

Analysis of Maximum Likelihood Estimates					
Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	-16.5660	5.4105	9.3747	0.0022
action_type_int	1	0.00746	0.00140	28.3326	<.0001
shot_type_int	1	-0.3046	0.0346	77.4983	<.0001
shot_zone_range_int	1	0.0345	0.0118	8.5730	0.0034
minutes_remaining	1	0.0124	0.00375	10.8645	0.0010
season	1	0.00671	0.00270	6.1914	0.0128
seconds_remaining	1	0.00293	0.000735	15.9440	<.0001
log_shot_distance	1	-0.0903	0.00532	288.6784	<.0001
arena_temp	1	0.0414	0.00634	42.6785	<.0001

Odds Ratio Estimates			
		95% Wald Confidence Limits	
Effect	Point Estimate		
action_type_int	1.007	1.005	1.010
shot_type_int	0.737	0.689	0.789
shot_zone_range_int	1.035	1.011	1.059
minutes_remaining	1.012	1.005	1.020
season	1.007	1.001	1.012
seconds_remaining	1.003	1.001	1.004
log_shot_distance	0.914	0.904	0.923
arena_temp	1.042	1.029	1.055



Association of Predicted Probabilities and Observed Responses			
Percent Concordant	62.1	Somers' D	0.243
Percent Discordant	37.9	Gamma	0.243
Percent Tied	0.0	Tau-a	0.120
Pairs	163169880	c	0.621

Classification Table

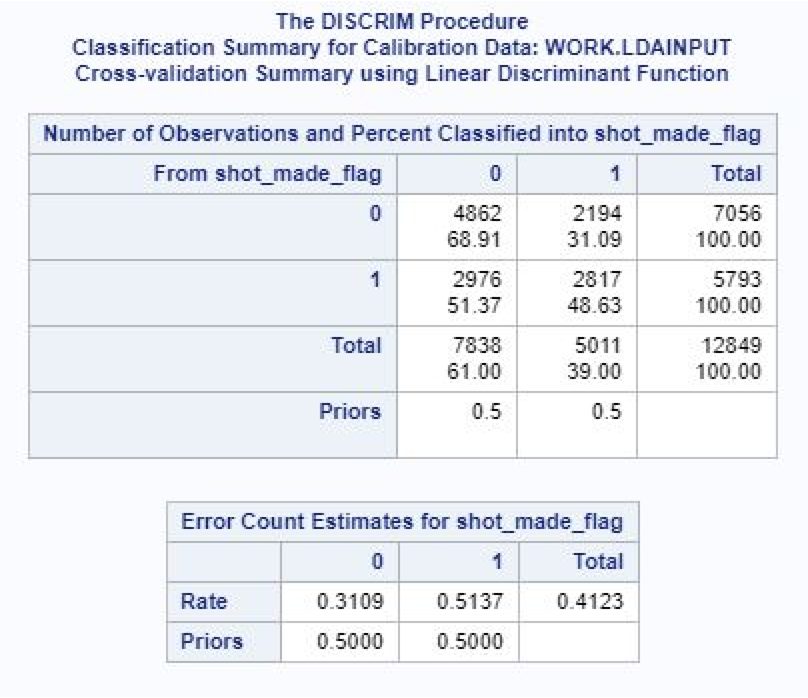
	Correct		Incorrect		Percentages				
Prob Level	Event	Non-Event	Event	Non-Event	Correct	Sensitivity	Specificity	False POS	False NEG
0.500	3508	12112	2120	7957	60.8	30.6	85.1	37.7	39.6

Figure 11

Linear Discriminant Analysis

Please see Figure 12 below for the results of our LDA model. While our LDA model is not very accurate at predicting successful shots made, it is fairly accurate at predicting shots missed. Our error rate for missed shots is only 31.1%, however our error rate for shots made is 51.4%. Our final LDA model is:

```
class shot_made_flag;
var action_type_int shot_type_int shot_zone_range_int opponent_int log_lat log_loc_y
lon minutes_remaining playoffs season seconds_remaining log_shot_distance
arena_temp;
priors "1" = 0.5 "0" = 0.5;
```



The DISCRIM Procedure
Classification Summary for Calibration Data: WORK.LDAINPUT
Cross-validation Summary using Linear Discriminant Function

Number of Observations and Percent Classified into shot_made_flag			
From shot_made_flag	0	1	Total
0	4862 68.91	2194 31.09	7056 100.00
1	2976 51.37	2817 48.63	5793 100.00
Total	7838 61.00	5011 39.00	12849 100.00
Priors	0.5	0.5	

Error Count Estimates for shot_made_flag			
	0	1	Total
Rate	0.3109	0.5137	0.4123
Priors	0.5000	0.5000	

Figure 12

Model Evaluation

The AUC for our logistic regression model is shown in Figure 13 of the Appendix.

The misclassification rate for the LDA model is (2194+2976)/12849 = 0.402. The misclassification rate for the logistic regression model is (2120+7957)/25697 = 0.392. So when looking at the misclassification rate, the logistic regression model is slightly more accurate.

The sensitivity for the LDA model is 48.63%. The sensitivity for the logistic regression model is 30.6%. So when looking at sensitivity, the LDA model performs better.

The specificity for the LDA model is 68.91%. The sensitivity for the logistic regression model is 85.1%. So when looking at specificity, the logistic regression model performs better.

The log loss value for the logistic regression model is 0.667. The log loss value for the LDA model is 0.668. So when comparing log loss values, the LDA model performs better.

After looking at all the above comparisons as a whole, the difference in performance between the two models is negligible. Both models appear to perform with relative similar accuracy.

Appendix 1

Variable Description

action_type: This field indicates the type of shot taken such as “Hook Shot” or “Running jump shot.” This field contains modifier indicators which may be useful for our analysis. Records containing the words “Dunk”, “Layup”, and “Tip” are classified as “short shots” and can be differentiated from other action_types.

combined _shot_type: This field indicates the overall classification of the shot type indicated in the action_type field.

matchup: This field indicates the team Kobe Bryant was playing against. This field is formatted as follows: LAL vs. {Opponent Team} or LAL @ {Opponent Team}..

opponent: This field is similar to matchup with a slightly different format. The LAL vs. and LAL @ indicator is removed which makes this field more useful for analysis.

season: This field indicates the 2 years the season spans in the format yyyy-yy.

shot_type: This field indicates whether the shot was a 2 point or 3 point shot. A 3 point shot is any shot greater than 23 feet 9 inches from the basket.

shot_zone_area: This field indicates where the shot was taken - back court, center, left side center, left side, right side center, or right side. This field does not indicate the distance from the basket.

shot_zone_basic: This field is another indicator for where the shot was taken but focuses more on the distance from the basket.

shot_zone_range: This field indicates the distance from the basket where the shot was taken.

team_name: This field simply indicates Kobe Bryant’s team name - Los Angeles Lakers and is not useful for data analysis.

team_id: This field is a record identifier for the Los Angeles Lakers team and is not useful for data analysis.

game_event_id: This field is a record identifier of some sort but we are not sure what it means.

game_id: This field is a record identifier for the game.

lat: This field indicates the latitude of where the shot was taken

loc_x: This field appears to indicate the x coordinate of where the shot taken using cartesian coordinates and has a range of -250 to 248.

loc_y: This field appears to indicate the y coordinate of where the shot taken using cartesian coordinates and has a range of -44 to 791.

lon: This field indicates the longitude of where the shot was taken

minutes_remaining: This field indicates the number of minutes remaining in the game period when the shot was taken and has a range of 0 to 11.

period: This field indicates the period of the game when the shot was taken.

playoff: This is a binary field indicating if the shot was taken during a playoff game.

seconds_remaining : This field, combined with the minutes_remaing field, indicates the time remaining in a game period when the shot was taken.

shot_distance: This field indicates the distance from the basket where the shot was taken. This appears to be in feet increments.

game_date: This field indicates the date of the game when the shot was taken.

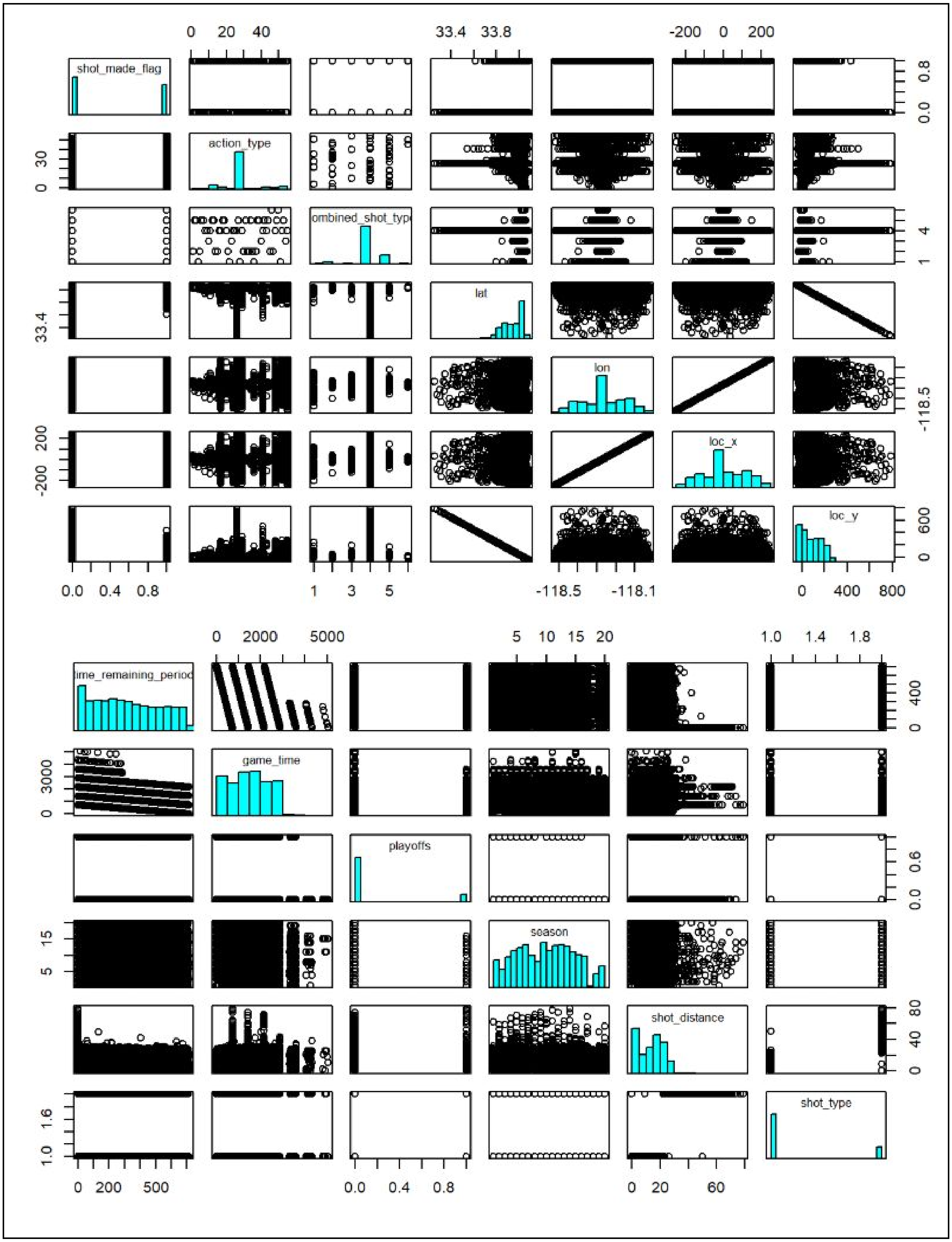
shot_id: This field is a record identifier for shot and is not useful for data analysis.

attendance: This field indicates the audience attendance when the shot was taken.

arena_temp: This field indicates the temperature of the arena (in fahrenheit) when the shot was taken.

avgnoisedb: This field indicates the average noise level (in decibels) when the shot was taken.

Figures


shot_made_flag
action_type
ombined_shot_typ
lat
lon
loc_x
loc_y
0 20 40
33.4 33.8
-200 0 200
0.0 0.8
0 30
1 4
33.4
-118.5
-200 200
0 600
0.0 0.4 0.8
1 3 5
-118.5 -118.1
0 400 800
ime_remaining_period
game_time
playoffs
season
shot_distance
shot_type
0 2000 5000
5 10 15 20
1.0 1.4 1.8
0 400
0 3000
0.0 0.6
5 15
0 40 80
1.0 1.6
0 200 500
0.0 0.4 0.8
0 20 60

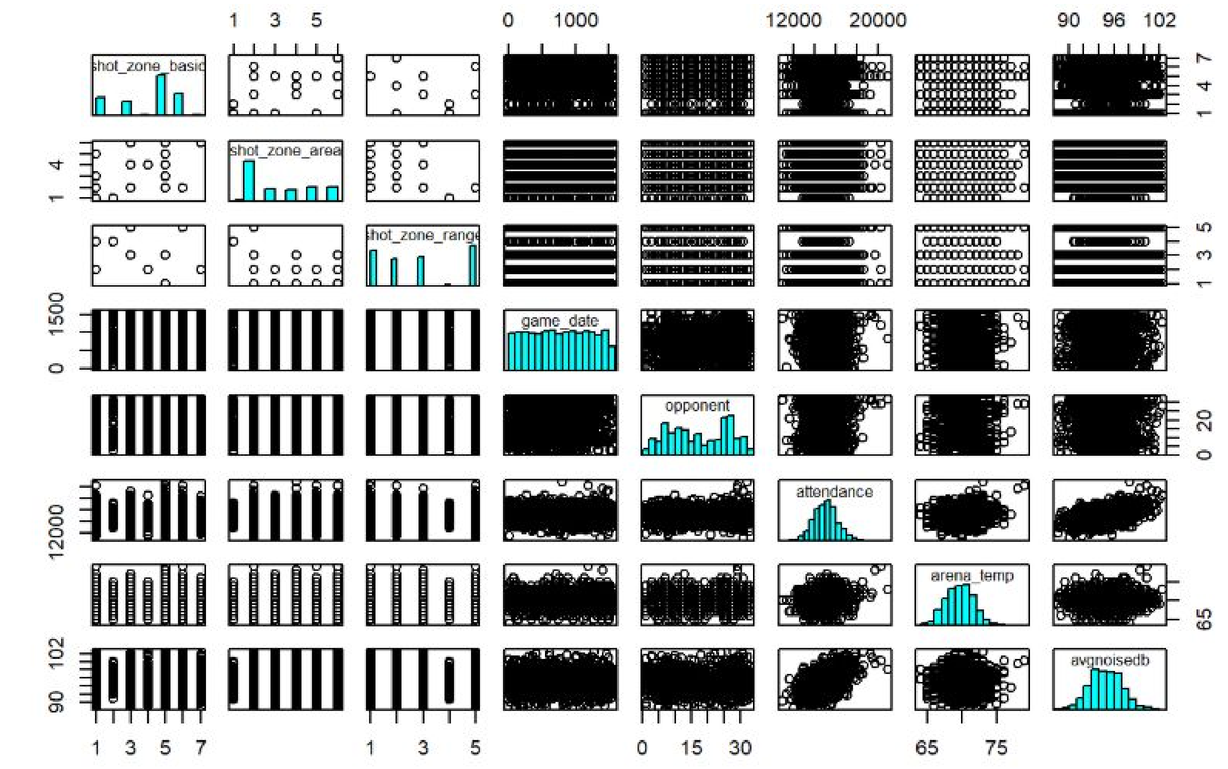


Figure 1

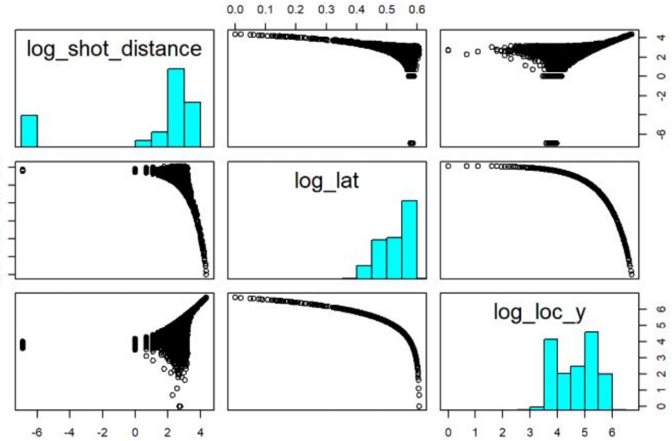
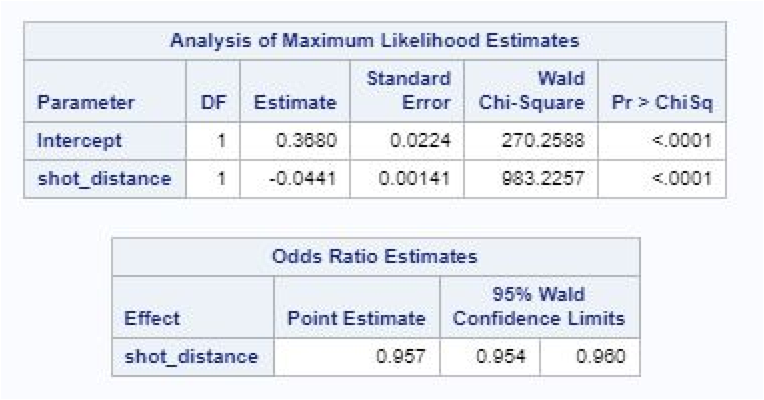


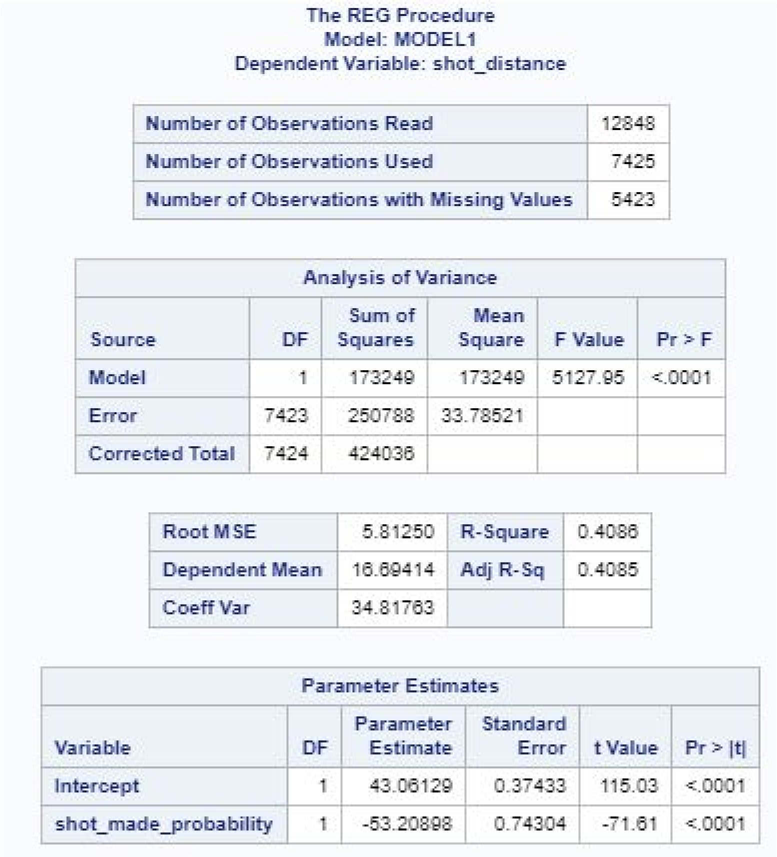
Figure 2



Analysis of Maximum Likelihood Estimates					
Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	0.3680	0.0224	270.2588	<.0001
shot_distance	1	-0.0441	0.00141	983.2257	<.0001

Odds Ratio Estimates			
Effect	Point Estimate	95% Wald Confidence Limits	
shot_distance	0.957	0.954	0.960

Figure 6



The REG Procedure
Model: MODEL1
Dependent Variable: shot_distance

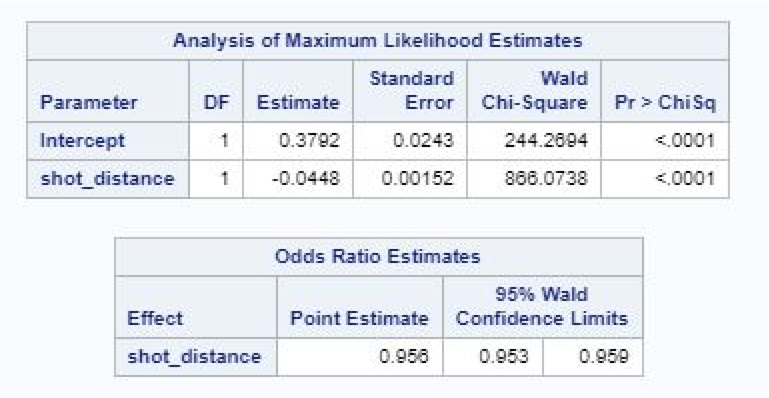
Number of Observations Read	12848
Number of Observations Used	7425
Number of Observations with Missing Values	5423

Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	173249	173249	5127.95	<.0001
Error	7423	250788	33.78521		
Corrected Total	7424	424036			

Root MSE	5.81250	R-Square	0.4086
Dependent Mean	16.69414	Adj R-Sq	0.4085
Coeff Var	34.81763		

Parameter Estimates					
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	43.06129	0.37433	115.03	<.0001
shot_made_probability	1	-53.20898	0.74304	-71.61	<.0001

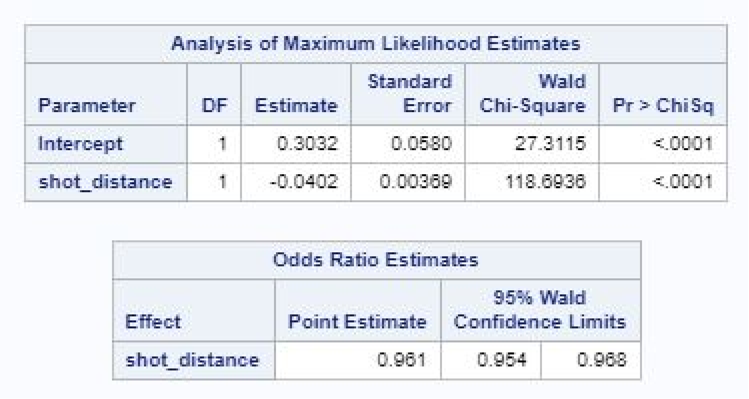
Figure 7



Analysis of Maximum Likelihood Estimates					
Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	0.3792	0.0243	244.2694	<.0001
shot_distance	1	-0.0448	0.00152	866.0738	<.0001

Odds Ratio Estimates			
Effect	Point Estimate	95% Wald Confidence Limits	
shot_distance	0.956	0.953	0.959

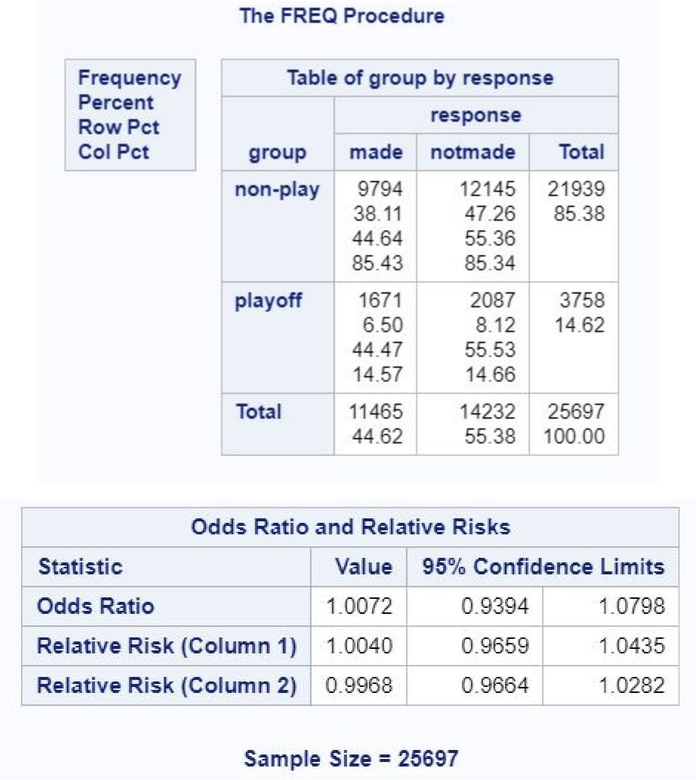
Figure 8



Analysis of Maximum Likelihood Estimates					
Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	0.3032	0.0580	27.3115	<.0001
shot_distance	1	-0.0402	0.00369	118.6936	<.0001

Odds Ratio Estimates			
Effect	Point Estimate	95% Wald Confidence Limits	
shot_distance	0.961	0.954	0.968

Figure 9



The FREQ Procedure

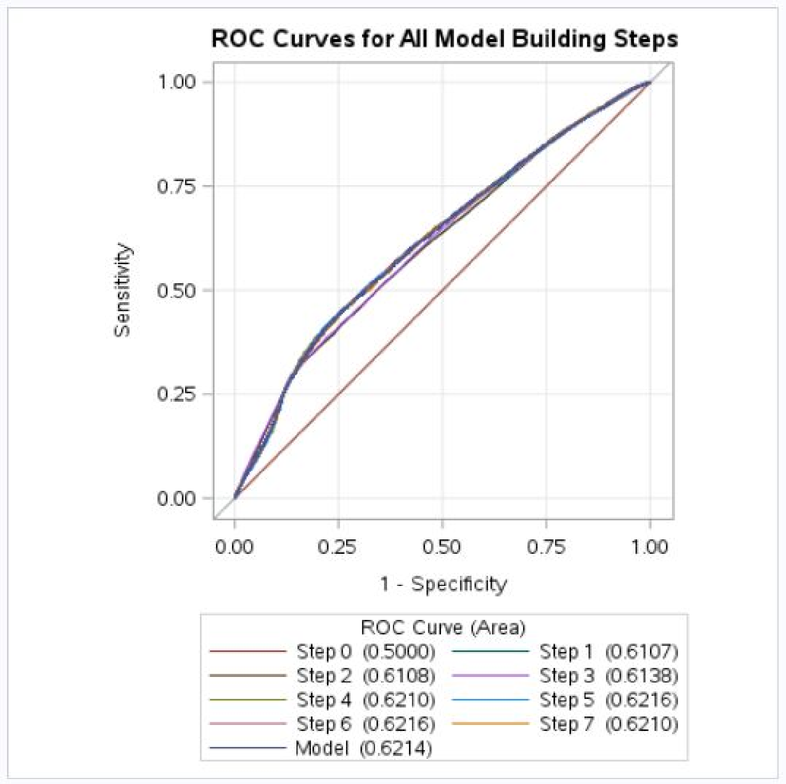
Frequency
Percent
Row Pct
Col Pct

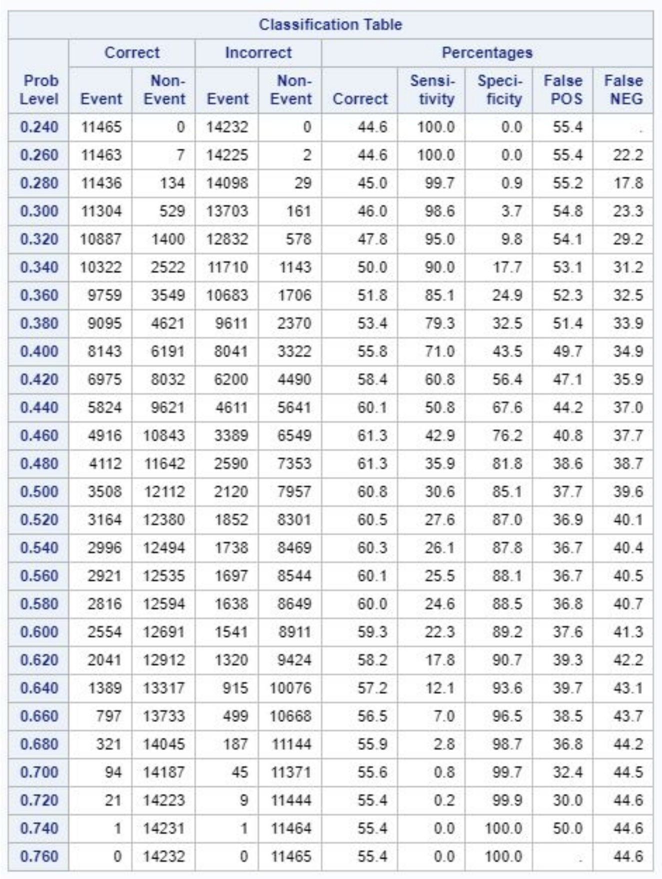
Table of group by response			
	response		
group	made	notmade	Total
non-play	9794 38.11 44.64 85.43	12145 47.26 55.36 85.34	21939 85.38
playoff	1671 6.50 44.47 14.57	2087 8.12 55.53 14.66	3758 14.62
Total	11465 44.62	14232 55.38	25697 100.00

Odds Ratio and Relative Risks			
Statistic	Value	95% Confidence Limits	
Odds Ratio	1.0072	0.9394	1.0798
Relative Risk (Column 1)	1.0040	0.9659	1.0435
Relative Risk (Column 2)	0.9968	0.9664	1.0282

Sample Size = 25697

Figure 10


ROC Curves for All Model Building Steps
Sensitivity
1.00
0.75
0.50
0.25
0.00
0.00
0.25
0.50
0.75
1.00
1 - Specificity
ROC Curve (Area)
Step 0 (0.5000)
Step 1 (0.6107)
Step 2 (0.6108)
Step 3 (0.6138)
Step 4 (0.6210)
Step 5 (0.6216)
Step 6 (0.6216)
Step 7 (0.6210)
Model (0.6214)



Classification Table									
	Correct		Incorrect		Percentages				
Prob Level	Event	Non-Event	Event	Non-Event	Correct	Sensi-tivity	Speci-ficity	False POS	False NEG
0.240	11465	0	14232	0	44.6	100.0	0.0	55.4	.
0.260	11463	7	14225	2	44.6	100.0	0.0	55.4	22.2
0.280	11436	134	14098	29	45.0	99.7	0.9	55.2	17.8
0.300	11304	529	13703	161	46.0	98.6	3.7	54.8	23.3
0.320	10887	1400	12832	578	47.8	95.0	9.8	54.1	29.2
0.340	10322	2522	11710	1143	50.0	90.0	17.7	53.1	31.2
0.360	9759	3549	10683	1706	51.8	85.1	24.9	52.3	32.5
0.380	9095	4621	9611	2370	53.4	79.3	32.5	51.4	33.9
0.400	8143	6191	8041	3322	55.8	71.0	43.5	49.7	34.9
0.420	6975	8032	6200	4490	58.4	60.8	56.4	47.1	35.9
0.440	5824	9621	4611	5641	60.1	50.8	67.6	44.2	37.0
0.460	4916	10843	3389	6549	61.3	42.9	76.2	40.8	37.7
0.480	4112	11642	2590	7353	61.3	35.9	81.8	38.6	38.7
0.500	3508	12112	2120	7957	60.8	30.6	85.1	37.7	39.6
0.520	3164	12380	1852	8301	60.5	27.6	87.0	36.9	40.1
0.540	2996	12494	1738	8469	60.3	26.1	87.8	36.7	40.4
0.560	2921	12535	1697	8544	60.1	25.5	88.1	36.7	40.5
0.580	2816	12594	1638	8649	60.0	24.6	88.5	36.8	40.7
0.600	2554	12691	1541	8911	59.3	22.3	89.2	37.6	41.3
0.620	2041	12912	1320	9424	58.2	17.8	90.7	39.3	42.2
0.640	1389	13317	915	10076	57.2	12.1	93.6	39.7	43.1
0.660	797	13733	499	10668	56.5	7.0	96.5	38.5	43.7
0.680	321	14045	187	11144	55.9	2.8	98.7	36.8	44.2
0.700	94	14187	45	11371	55.6	0.8	99.7	32.4	44.5
0.720	21	14223	9	11444	55.4	0.2	99.9	30.0	44.6
0.740	1	14231	1	11464	55.4	0.0	100.0	50.0	44.6
0.760	0	14232	0	11465	55.4	0.0	100.0	.	44.6

Figure 13

Appendix 2

SAS Code

```
%web_drop_table(trainData);


FILENAME REFFILE '/home/dserna0/Code/6372/GroupProject2/trainDataFactorized.csv';

PROC IMPORT DATAFILE=REFFILE
	DBMS=CSV
	OUT=trainData;
	GETNAMES=YES;
RUN;

PROC CONTENTS DATA=trainData; RUN;
```

```
%web_open_table(trainData);

%web_drop_table(testData);


FILENAME REFFILE '/home/dserna0/Code/6372/GroupProject2/testDataFactorized.csv';

PROC IMPORT DATAFILE=REFFILE
	DBMS=CSV
	OUT=testData;
	GETNAMES=YES;
RUN;

PROC CONTENTS DATA=testData; RUN;


%web_open_table(testData);

/* data type conversions */
data trainData;
set trainData;
intSeason = input(season, BEST32.);
drop season;
rename intSeason = season;

data testData;
set testData;
intSeason = input(season, BEST32.);
drop season;
rename intSeason = season;
run;

data testData;
set testData;
int_shot_made_flag = input(shot_made_flag, BEST32.);
drop shot_made_flag;
rename int_shot_made_flag = shot_made_flag;
run;

data trainData;
set trainData;
```

```
Obs = _n_;
run;

data testData;
set testData;
Obs = _n_;
run;

/* Proposition 1 - Odds Ratio */
proc logistic data = trainData descending;
model shot_made_flag(event="1") = shot_distance / selection=stepwise ctable pprob = 0.5;
output out=logisticOut predprobs = I p=predprob resdev=resdev reschi=pearres;
run;

/* Proposition 3 - odds differnet in playoffs */
data trainDataNonPlayoffs;
set trainData;
if playoffs EQ 1 then delete;
run;

data manualOdds;
input group$ response$ n;
datalines;
non-playoff made 9794
non-playoff notmade 12145
playoff made 1671
playoff notmade 2087
;

proc freq data = manualOdds order = data;
weight n;
tables group*response / riskdiff(equal var=null cl=wald) relrisk;
run;


data trainDataPlayoffs;
set trainData;
if playoffs EQ 0 then delete;
run;

/*
proc print data = trainDataNonPlayoffs(OBS=50);run;
proc print data = trainDataPlayoffs(OBS=50);run;
```

```
*/

proc logistic data = trainDataNonPlayoffs descending;
model shot_made_flag(event="1") = shot_distance / selection=stepwise ctable pprob = 0.5;
output out=logisticOut predprobs = I p=predprob resdev=resdev reschi=pearres;
run;

proc logistic data = trainDataPlayoffs descending;
model shot_made_flag(event="1") = shot_distance / selection=stepwise ctable pprob = 0.5;
output out=logisticOut predprobs = I p=predprob resdev=resdev reschi=pearres;
run;


/* Predictive Models - Logistic Regression  */
/*train the model*/
proc logistic data = trainData descending;
model shot_made_flag(event="1") = action_type_int shot_type_int shot_zone_range_int
opponent_int log_lat log_loc_y lon minutes_remaining playoffs season seconds_remaining
log_shot_distance arena_temp / selection=stepwise ctable pprob = 0.5;
output out=logisticOut predprobs = I p=predprob resdev=resdev reschi=pearres;
run;

/*logistic prediction*/
data predictionData;
set trainData testData;
run;

proc logistic data = predictionData descending;
model shot_made_flag = action_type_int shot_type_int shot_zone_range_int opponent_int
log_lat log_loc_y lon minutes_remaining playoffs season seconds_remaining log_shot_distance
arena_temp / selection=stepwise ctable pprob = 0.5;
output out=logisticOut predprobs = I p=predprob resdev=resdev reschi=pearres;
run;

data logisticResults;
set logisticOut;
keep rannum shot_made_flag predprob;
where shot_made_flag EQ .;
run;

data logisticResults;
set logisticResults;
if predprob >= 0.50 then shot_made_flag = 1;
if predprob < 0.50 then shot_made_flag = 0;
run;

/* Predictive Models - Linear Discriminant Analysis */
/* train the models */
data ldaInput;
set trainData;
if mod(obs, 2) EQ 0 then delete;
run;

data ldaToCategorize;
set trainData;
if mod(obs, 2) NE 0 then delete;
run;

proc discrim data=ldaInput pool=YES crossvalidate testdata=ldatocategorize
testout=discrimOut;
class shot_made_flag;
var action_type_int shot_type_int shot_zone_range_int opponent_int log_lat log_loc_y lon
minutes_remaining playoffs season seconds_remaining log_shot_distance arena_temp;
priors "1" = 0.5 "0" = 0.5;
run;

data discrimOut;
set discrimOut;
rename '1'n = shot_made_probability;
rename '0'n = shot_not_made_probability;
run;

/* lda prediction */
proc discrim data=trainData pool=YES crossvalidate testdata=testData testout=ldaResults;
class shot_made_flag;
var action_type_int shot_type_int shot_zone_range_int opponent_int log_lat log_loc_y lon
minutes_remaining playoffs season seconds_remaining log_shot_distance arena_temp;
priors "1" = 0.5 "0" = 0.5;
run;

data ldaResults;
set ldaResults;
rename '1'n = shot_made_probability;
rename '0'n = shot_not_made_probability;
run;
```

```
data ldaResults;
set ldaResults;
if shot_made_probability >= 0.50 then shot_made_flag = 1;
if shot_made_probability < 0.50 then shot_made_flag = 0;
keep rannum shot_made_flag shot_made_probability shot_not_made_probability;
run;


/* Proposition 2 - probability shot_made decreases with distance */
proc reg data=discrimOut;
model shot_distance = shot_made_probability;
run;

/* model evaluation */
/*Logistic Model*/
proc logistic data = trainData descending plots=all;
model shot_made_flag(event="1") = action_type_int shot_type_int shot_zone_range_int
opponent_int log_lat log_loc_y lon minutes_remaining playoffs season seconds_remaining
log_shot_distance arena_temp / selection=stepwise ctable lackfit clparm=wald;
output out=logisticOut predprobs = I p=predprob resdev=resdev reschi=pearres;
run;

/*LDA Model*/
proc discrim data=trainData pool=YES crossvalidate testdata=testData testout=ldaResults;
class shot_made_flag;
var action_type_int shot_type_int shot_zone_range_int opponent_int log_lat log_loc_y lon
minutes_remaining playoffs season seconds_remaining log_shot_distance arena_temp;
priors "1" = 0.5 "0" = 0.5;
run;
```

R Code

~~~
---
title: "6372Project2"
author: "Shon Mohsin, Daniel Serna, William Trevino"
date: "November 3, 2018"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```
~~~

~~~
```{r libraryImports}
if(!require(tidyverse)) install.packages("tidyverse")
if(!require(caret)) install.packages("caret")
if(!require(knitr)) install.packages("knitr")
if(!require(MLmetrics)) install.packages("MLmetrics")
```

```{r addUtilityFunctions}
panel.hist <- function(x, ...)
{
    usr <- par("usr"); on.exit(par(usr))
    par(usr = c(usr[1:2], 0, 1.5) )
    h <- hist(x, plot = FALSE)
    breaks <- h$breaks; nB <- length(breaks)
    y <- h$counts; y <- y/max(y)
    rect(breaks[-nB], 0, breaks[-1], y, col = "cyan", ...)
}

addConstantForLogTransform <- function(x)
{
  returnValue = x + (1 - min(x))
}
```

```{r importData}
trainData <- read.csv("project2Data.csv")
testData <- read.csv("project2Pred.csv")
```

```{r}
#changing factors to numeric for data analysis
trainData$rannum <- trainData$recId
trainData$recId <- NULL
allData <- rbind(trainData, testData)

allData$action_type_int <- as.numeric(allData$action_type)
allData$combined_shot_type_int <- as.numeric(allData$combined_shot_type)
allData$shot_type_int <- as.numeric(allData$shot_type)
allData$shot_zone_are_int <- as.numeric(allData$shot_zone_area)
allData$shot_zone_basic_int <- as.numeric(allData$shot_zone_basic)
allData$shot_zone_range_int <- as.numeric(allData$shot_zone_range)
allData$opponent_int <- as.numeric(allData$opponent)
~~~

~~~
allData$time_remaining_period <- allData$seconds_remaining
+(60*allData$minutes_remaining)
allData$game_time <- (12*60 - allData$time_remaining_period) + ((allData$period - 1)*12*60)
head(allData)

trainDataFactorized <- allData[is.na(allData$shot_made_flag) == FALSE,]
testDataFactorized <- allData[is.na(allData$shot_made_flag) == TRUE,]

trainDataFactorized$log_lat = log(trainDataFactorized$lat)
#we have 0s in the shot_distance field, so we need to add a small number so we can take the
log transform.
trainDataFactorized$log_shot_distance = log(trainDataFactorized$shot_distance + 0.001)
#we have negative values in the following fields, so we need to add a constant so we can take
the log tranform.
trainDataFactorized$log_loc_y = log(addConstantForLogTransform(trainDataFactorized$loc_y))

testDataFactorized$log_lat = log(testDataFactorized$lat)
#we have 0s in the shot_distance field, so we need to add a small number so we can take the
log transform.
testDataFactorized$log_shot_distance = log(testDataFactorized$shot_distance + 0.001)
#we have negative values in the following fields, so we need to add a constant so we can take
the log tranform.
testDataFactorized$log_loc_y = log(addConstantForLogTransform(testDataFactorized$loc_y))

trainDataFactorized$season <- substr(trainDataFactorized$season, 1,4)
testDataFactorized$season <- substr(testDataFactorized$season, 1,4)

write.csv(trainDataFactorized, "trainDataFactorized.csv", row.names = FALSE)
write.csv(testDataFactorized, "testDataFactorized.csv", row.names = FALSE)

```



```{r dataTransformations}
trainData$time_remaining_period <- trainData$seconds_remaining
+(60*trainData$minutes_remaining)
trainData$game_time <- (12*60 - trainData$time_remaining_period) + ((trainData$period -
1)*12*60)
```

```{r outlierAnalysis}
pairs(~shot_made_flag+action_type+combined_shot_type+lat+lon+loc_x+loc_y,data=trainData,
main="Simple Scatterplot Matrix", diag.panel=panel.hist)
~~~

~~~
pairs(~time_remaining_period+game_time+playoffs+season+shot_distance+shot_type,data=trainData,main="Simple Scatterplot Matrix", diag.panel=panel.hist)

pairs(~shot_zone_basic+shot_zone_area+shot_zone_range+game_date+opponent+attendance+arena_temp+avgnoisedb,data=trainData,main="Simple Scatterplot Matrix", diag.panel=panel.hist)

```

```{r logTransformations}
trainData$log_lat = log(trainData$lat)
#we have 0s in the shot_distance field, so we need to add a small number so we can take the log transform.
trainData$log_shot_distance = log(trainData$shot_distance + 0.001)
#we have negative values in the following fields, so we need to add a constant so we can take the log tranform.
trainData$log_loc_y = log(addConstantForLogTransform(trainData$loc_y))

#the above log transformations don't seem to help much at all.
pairs(~log_shot_distance+log_lat+log_loc_y,data=trainData,main="Simple Scatterplot Matrix", diag.panel=panel.hist)
```

```{r varianceInflationFactor}
trainData$team_name <- NULL #this is always Lakers
trainData$team_id <- NULL #this is always Lakers
trainData$game_date <- NULL
trainData$matchup <- NULL
trainData$recId <- NULL
trainData$game_event_id <- NULL
trainData$game_id <- NULL
trainData$shot_id <- NULL
trainData$combined_shot_type <- NULL
trainData$shot_zone_basic <- NULL
trainData$loc_x <- NULL
trainData$loc_y <- NULL
trainData$shot_zone_range <- NULL
trainData$log_lat <-NULL
trainData$log_loc_y <-NULL
trainData$log_shot_distance <-NULL
trainData$minutes_remaining <-NULL
trainData$seconds_remaining <-NULL
~~~

```
trainData$period <-NULL


set.seed(123)
trainData.samples <- trainData$shot_made_flag %>%
  createDataPartition(p = 0.8, list = FALSE)
train.data  <- trainData[trainData.samples, ]
test.data <- trainData[-trainData.samples, ]


# Build the model
model1 <- lm(shot_made_flag~., data = train.data)
# Make predictions
predictions <- model1 %>% predict(test.data)
# Model performance
data.frame(
  RMSE = RMSE(predictions, test.data$shot_made_flag),
  R2 = R2(predictions, test.data$shot_made_flag)
)

# Determine multicollinearity
car::vif(model1)

# Write variable VIF to csv file for analysis
datavif <- car::vif(model1)
write.csv(datavif, "datavif.csv")

#after successful vif run, we removed the following variable
trainData$shot_zone_area <- NULL

set.seed(123)
trainData.samples <- trainData$shot_made_flag %>%
  createDataPartition(p = 0.8, list = FALSE)
train.data  <- trainData[trainData.samples, ]
test.data <- trainData[-trainData.samples, ]


# Build the model
model1 <- lm(shot_made_flag ~., data = train.data)
# Make predictions
predictions <- model1 %>% predict(test.data)
# Model performance
data.frame(
```

~~~
  RMSE = RMSE(predictions, test.data$shot_made_flag),
  R2 = R2(predictions, test.data$shot_made_flag)
)

# Determine multicollinearity
car::vif(model1)

# Write variable VIF to csv file for analysis
datavif <- car::vif(model1)
write.csv(datavif, "datavif.csv")

trainData2 <- trainData
trainData2$action_type <- NULL

set.seed(123)
trainData2.samples <- trainData2$shot_made_flag %>%
  createDataPartition(p = 0.8, list = FALSE)
train2.data  <- trainData2[trainData2.samples, ]
test2.data <- trainData2[-trainData2.samples, ]


# Build the model
model2 <- lm(shot_made_flag ~., data = train2.data)
# Make predictions
predictions2 <- model2 %>% predict(test2.data)
# Model performance
data.frame(
  RMSE = RMSE(predictions2, test2.data$shot_made_flag),
  R2 = R2(predictions2, test2.data$shot_made_flag)
)

# Determine multicollinearity
car::vif(model2)

# Write variable VIF to csv file for analysis
datavif2 <- car::vif(model2)
write.csv(datavif2, "datavif2.csv")
```


```{r exportData}
trainData$season <- substr(trainData$season, 1,4)
write.csv(trainData, "trainDataClean.csv", row.names = FALSE)
```
~~~

~~~
```{r log loss function}
library(MLmetrics)
loglossdata <- read.csv("LOGISTICOUT.csv")
y_pred <- loglossdata$predprob
x_true <- loglossdata$shot_made_flag
loglossoutput <- LogLoss(y_pred, x_true)
loglossoutput

loglossdata1 <- read.csv("DISCRIMOUT.csv")
y_pred1 <- loglossdata1$shot_made_probability
x_true1 <- loglossdata1$shot_made_flag
loglossoutput1 <- LogLoss(y_pred1, x_true1)
loglossoutput1

```
~~~