

Rocket League Analysis

Katie Lee, Hugo Leo, Linda Wang

March 4, 2022

Contents

1	Executive Summary	2
2	Data Source	3
3	Data Cleaning	3
4	Exploratory Data Analysis	3
5	Expected Goals Models	3
5.1	Analysis Goals	4
5.2	Feature Engineering	4
5.3	Logistic Regression	4
5.4	Penalized Regressions	6
5.5	XGBoost	6
5.6	Model Evaluation	7
6	Excess Goals	7
7	Appendix	7



1 Executive Summary

Rocket League is a multi-platform game where players control cars with a rocket booster and aim to score as many goals as possible within 5 minutes using an oversized ball. The ball never leaves the field and is a team game, but for our analysis, we focused on 2 versus 2 player mode. Players have the ability to control the direction, velocity, and rotation of their cars and can jump to hit the ball. This allows players the ability to take shots or control the car to dribble, pass, and block incoming shots.

Our dataset includes random frames of 2 versus 2 player Rocket League games that includes location data of all 4 players, car features, ball position, and whether or not a particular shot resulted in a goal.

For our analysis, we built an expected goals (xG) model using location data to predict goals for a given frozen game frame. xG is important in that it is an estimator of goals a team is expected to score in the long run. Using our xG model, we explored individual player ability and analyzed the question: are good players good because they use more attempts, or are good players good because they are truly better skilled?

2 Data Source

[to write]

3 Data Cleaning

- dropped car features, only used location data
- because we are interested in expected goals and evaluating shot ability, we filtered for shot = true only

4 Exploratory Data Analysis

As a first step, we explored the location data to get a sense of the placement of the goal. Based on Figure 1, it seems that the goal post is located at $(0, 5000, 0)$.

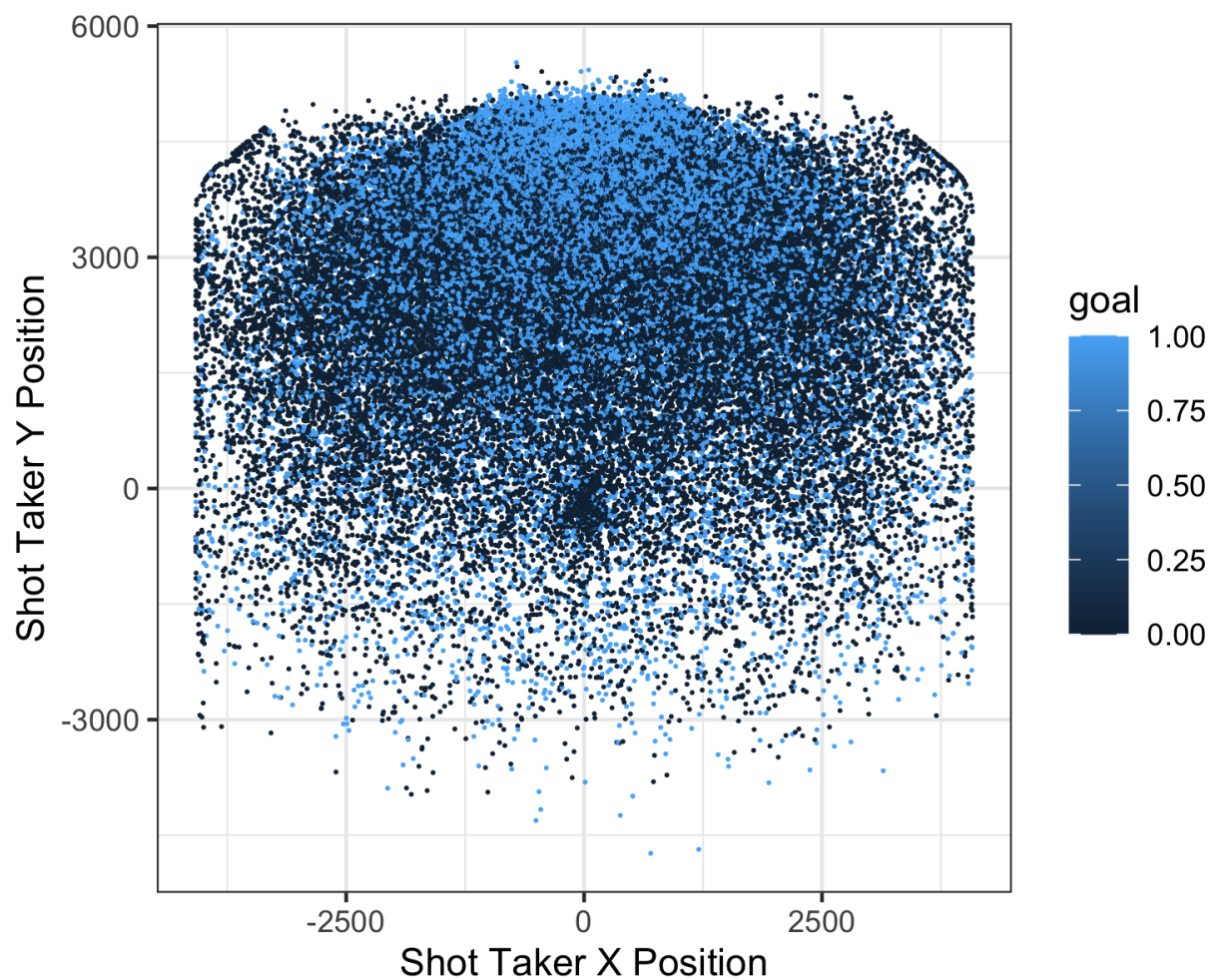


Figure 1: X-Y Field and Shot Outcome

5 Expected Goals Models

We built two expected goals models using logistic regression and boosted trees. Because of the nature of the dataset, we have data about the ball's trajectory, so we can easily compute whether a shot will result in a

goal or not. However, for our model, we used position, velocity, angular velocity, and rotation of the shot taker, defender, and teammate to build our model. To train our model, we predicted the outcome of a shot (goal or no goal) for a particular frame. This implies that our xG estimates are for a set of positions of the shot taker, teammate, and defenders.

5.1 Analysis Goals

- how is xg used in soccer
- why is xg important

5.2 Feature Engineering

We used feature engineering to transform the raw data into features that can be used in supervised learning.

5.3 Logistic Regression

- discuss how we excluded teammate data? maybe because MLE doesn't converge because of high correlation across variables and the goals variable???

```
load("../results/glm_fit.rda")
summary(glm_fit)
```

```
##
## Call:
## glm(formula = goal ~ . - idx - distanceToGoal, family = binomial(link = "logit"),
##      data = shot_train)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -5.372  -0.796  -0.483   0.897   3.231
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)      1.65e+00   3.01e-01   5.46 4.8e-08 ***
## ball_pos_x      -1.09e-04   2.00e-04  -0.55 0.58485
## ball_pos_y       1.35e-03   1.86e-04   7.27 3.5e-13 ***
## ball_pos_z      -2.64e-03   2.40e-04 -11.02 < 2e-16 ***
## ball_vel_x      -2.13e-06   2.71e-06  -0.79 0.43182
## ball_vel_y       1.02e-04   2.64e-06  38.74 < 2e-16 ***
## ball_vel_z       1.17e-05   3.57e-06   3.27 0.00108 **
## ball_ang_vel_x    1.03e-04   4.39e-06  23.46 < 2e-16 ***
## ball_ang_vel_y   -4.43e-07   4.41e-06  -0.10 0.91988
## ball_ang_vel_z   -7.45e-06   4.55e-06  -1.64 0.10132
## ball_hit_team_no -1.54e-02   2.33e-02  -0.66 0.50877
## ball_rot_x        5.41e-03   1.68e-02   0.32 0.74747
## ball_rot_y      -1.98e-02   6.33e-03  -3.13 0.00177 **
## ball_rot_z      -1.61e-03   6.36e-03  -0.25 0.80004
## shot_taker_pos_x  1.15e-04   1.97e-04   0.58 0.55857
## shot_taker_pos_y  -6.93e-04   1.84e-04  -3.77 0.00017 ***
## shot_taker_pos_z  3.26e-03   2.44e-04  13.40 < 2e-16 ***
## shot_taker_vel_x  2.78e-06   2.32e-06   1.20 0.23092
## shot_taker_vel_y  -3.17e-05   2.65e-06 -11.94 < 2e-16 ***
## shot_taker_vel_z  8.85e-05   4.95e-06  17.89 < 2e-16 ***
## shot_taker_ang_vel_x 4.69e-06   4.69e-06   1.00 0.31733
## shot_taker_ang_vel_y 7.68e-06   4.50e-06   1.71 0.08772 .
```

```

## shot_taker_ang_vel_z      -1.06e-05  5.72e-06  -1.86  0.06308 .
## shot_taker_rot_x          8.09e-03  2.60e-02   0.31  0.75571
## shot_taker_rot_y          8.07e-02  1.12e-02   7.19  6.7e-13 ***
## shot_taker_rot_z         -1.11e-04  1.08e-02  -0.01  0.99185
## opp_1_pos_x              -1.04e-05  7.37e-06  -1.41  0.15923
## opp_1_pos_y              -1.95e-04  6.92e-06 -28.21 < 2e-16 ***
## opp_1_pos_z               5.41e-05  6.44e-05   0.84  0.40094
## opp_1_vel_x              -9.84e-07  1.21e-06  -0.81  0.41756
## opp_1_vel_y               5.13e-06  1.38e-06   3.72  0.00020 ***
## opp_1_vel_z              -5.68e-05  4.69e-06 -12.11 < 2e-16 ***
## opp_1_ang_vel_x          -2.31e-05  7.09e-06  -3.26  0.00113 **
## opp_1_ang_vel_y           5.61e-06  7.31e-06   0.77  0.44251
## opp_1_ang_vel_z           4.62e-06  7.54e-06   0.61  0.54040
## opp_1_rot_x               5.44e-03  3.23e-02   0.17  0.86614
## opp_1_rot_y               8.30e-03  8.07e-03   1.03  0.30352
## opp_1_rot_z              -5.92e-04  1.36e-02  -0.04  0.96520
## opp_2_pos_x               5.62e-06  7.31e-06   0.77  0.44230
## opp_2_pos_y              -1.72e-04  6.77e-06 -25.40 < 2e-16 ***
## opp_2_pos_z               1.88e-04  6.26e-05   3.01  0.00264 **
## opp_2_vel_x               9.26e-07  1.21e-06   0.76  0.44464
## opp_2_vel_y               6.20e-06  1.39e-06   4.47  7.8e-06 ***
## opp_2_vel_z              -5.03e-05  4.60e-06 -10.94 < 2e-16 ***
## opp_2_ang_vel_x          -2.65e-05  7.12e-06  -3.72  0.00020 ***
## opp_2_ang_vel_y          -7.87e-07  7.32e-06  -0.11  0.91439
## opp_2_ang_vel_z          -9.27e-06  7.51e-06  -1.23  0.21733
## opp_2_rot_x               3.13e-02  3.17e-02   0.99  0.32356
## opp_2_rot_y               4.82e-03  7.97e-03   0.60  0.54572
## opp_2_rot_z              -4.28e-03  1.36e-02  -0.31  0.75303
## shot_taker_boost_activeTrue 7.64e-02  2.28e-02   3.34  0.00083 ***
## logDistanceToGoal         -5.50e-01  3.38e-02 -16.28 < 2e-16 ***
## distanceToOpp1            -1.18e-04  8.35e-06 -14.18 < 2e-16 ***
## distanceToOpp2            -9.79e-05  8.20e-06 -11.93 < 2e-16 ***
## distanceToTeam            -1.40e-05  6.53e-06  -2.15  0.03148 *
## cos_theta_opp_1           2.85e+06  1.63e+05  17.42 < 2e-16 ***
## cos_theta_opp_2           3.40e+06  1.69e+05  20.09 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 60522 on 46785 degrees of freedom
## Residual deviance: 48396 on 46729 degrees of freedom
## AIC: 48510
##
## Number of Fisher Scoring iterations: 4

```

5.4 Penalized Regressions

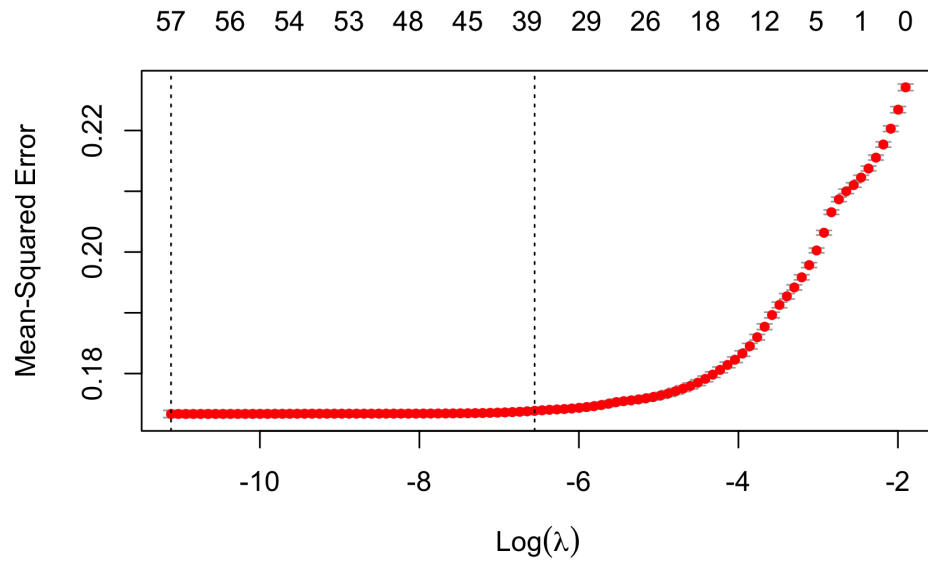


Figure 2: Cross Validation for Lasso Regression

5.5 XGBoost

```
load("../results/gbm_fit.Rda")
boost_feature_importance <- read.csv("../results/boost_feature_importance.csv") %>%
  as_tibble() %>%
  select(-c(X))

boost_feature_importance %>%
  kable(format = "latex", row.names = NA,
        booktabs = TRUE, digits = 2,
        col.names = c("Variable", "Relative Influence"),
        caption = "Boosting Important Variables") %>%
  kable_styling(position = "center", latex_options = "HOLD_position")
```

Table 1: Boosting Important Variables

Variable	Relative Influence
distanceToGoal	18.55
cos_theta_opp_2	12.84
cos_theta_opp_1	12.81
ball_pos_y	11.78
ball_vel_y	11.65
ball_pos_z	9.06
shot_taker_pos_z	4.31
ball_vel_z	4.05
ball_vel_x	2.07
team_mate_pos_y	1.89

5.6 Model Evaluation

Table 2: Model Evaluation

Model	Log Loss
Logistic Regression, FE, no teammate data	0.52
Ridge Regression, FE, no teammate data	0.59
Lasso Regression, FE, no teammate data	0.62
xgBoost, FE, with teammate data	0.46
Naive classifier	0.65

6 Excess Goals

To extent our Expected Goals model, we measured players' excess goals, or outperformance. Outperformance is given by dividing total goals scored divided by the sum of the player's expected goal probabilities. An outperformance ratio above one suggests the player is a "good" player in the sense that they are a good shooter and are able to make more goals than what is expected based on their location, their opponent's location, and various ball features.

7 Appendix

Subset of locational features used in models

```
data <- read.csv("../results/clean_shot_data.csv") %>%
  select(-c(X, idx)) %>%
  select(-contains("_ang_vel_")) %>%
  select(-contains("rot"))

names(data) %>%
  kable(format = "latex", row.names = NA,
        booktabs = TRUE, digits = 2,
        caption = "Subset of Location Features") %>%
  kable_styling(position = "center", latex_options = "HOLD_position")
```

Table 3: Subset of Location Features

x
goal
distanceToGoal
ball_pos_x
ball_pos_y
ball_pos_z
ball_vel_x
ball_vel_y
ball_vel_z
ball_hit_team_no
shot_taker_pos_x
shot_taker_pos_y
shot_taker_pos_z
shot_taker_vel_x
shot_taker_vel_y
shot_taker_vel_z
team_mate_id
team_mate_pos_x
team_mate_pos_y
team_mate_pos_z
team_mate_vel_x
team_mate_vel_y
team_mate_vel_z
opp_1_pos_x
opp_1_pos_y
opp_1_pos_z
opp_1_vel_x
opp_1_vel_y
opp_1_vel_z
opp_2_pos_x
opp_2_pos_y
opp_2_pos_z
opp_2_vel_x
opp_2_vel_y
opp_2_vel_z
shot_taker_boost_active