Rocket League Analysis

Katie Lee, Hugo Leo, Linda Wang

March 14, 2022

Contents


ROCKET
LEAGUE

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. The dataset is from Ball Chasing, which includes data from various Rocket League game replays.

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 the xG model, we explored:

1. What does it mean for a player to "outperform"? Is outperformance a random process, or is it largley attribuable to a player's ability to position and utilize ball features well?
2. Even though Rocket League does not assign specific positions as soccer does, do players adopt a strictly offensive or strictly defensive position? Or are skilled players good at both offense and defense?
3. What were the playstyles of the players in the Rocket League Championship Series games? Was the outcome expected given aggregate game statistics, like accumulated xG?

2 Data Source

We obtained our data from Ball Chasing, which is a website that records various Rocket League game replay data, along with the carball Python package that combines various tools for decompiling Rocket League replays. For our expected goals model, our raw dataset included random frames from 4,328 Rocket League 2v2 games over Platinum 3 level, which yielded 657,897 frames in total. For our analysis of the top two Rocket League champions, we scraped data from Ball Chasing.

3 Data Cleaning

Because our analysis is on intended shot attempts, we filtered the data for frames that included shot attempts only, which yielded 60,341 observations that we used for our expected goals models. While the raw dataset included various features, such as car features (like dodging, double jumping, ball cam), we transformed the dataset to include location data (ball, shot taker, teammate, and both opponent's locations) and various velocity measurements.

4 Exploratory Data Analysis

4.1 Field Set Up

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 $(0, 5000, 0)$.

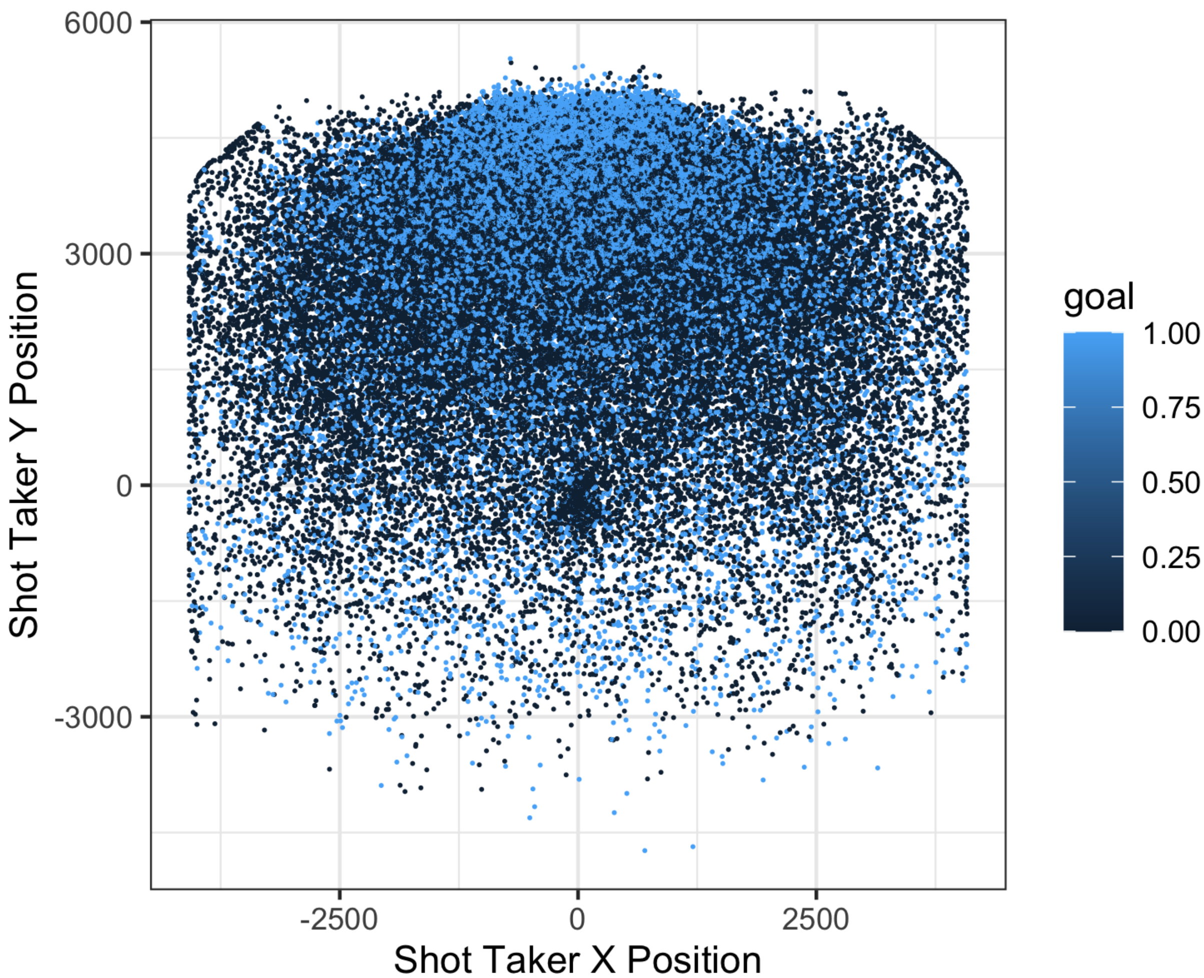

Figure 1: X-Y Field and Shot Outcome

4.2 Feature-Feature and Feature-Response Correlation

Next, we explored high-level relationships and correlations of the predictor variables with other predictor variables. We first looked at correlations between a few school-specific features, as shown in Figure 2. We observe a positive correlation between the (x, y, z) positions of the ball and the shot taker, as well as between the position of the shot taker, the teammate, and the opponents. Ultimately, we chose to include all of the correlated features, as we believe it to be informative from what angle the shot taker is striking the ball at (i.e. it could be that more skilled players are striking the ball from below or from above), as well as the locations of all players in the field.

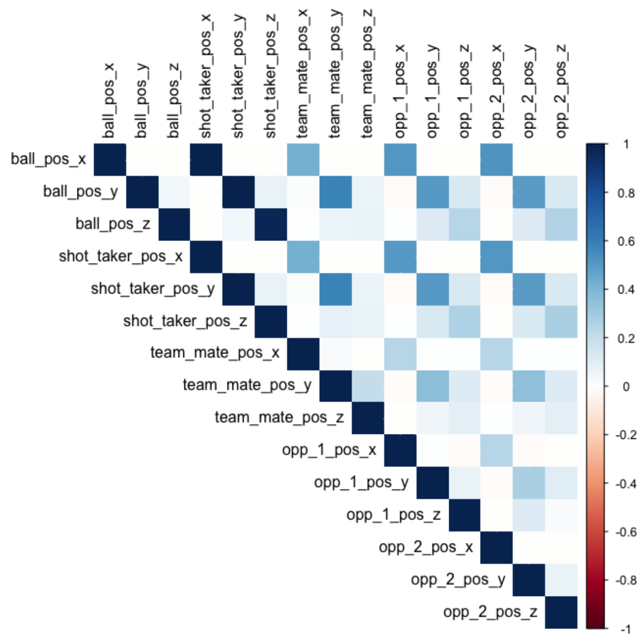

Figure 2: Position Features Correlation Plot

In addition, we looked relationships of some predictor variables with the response variable. Specifically, we examined how distances to the goal, to the shot taker's teammate, and to the two opponents were associated with the goal outcome. As shown in Figure 3, the log of the distance to the goal has a notable difference, with shorter distances resulting in more goals. The shot taker's distance to the teammate and the two opponents have less of a difference between those shots that resulted in goals and those that did not.

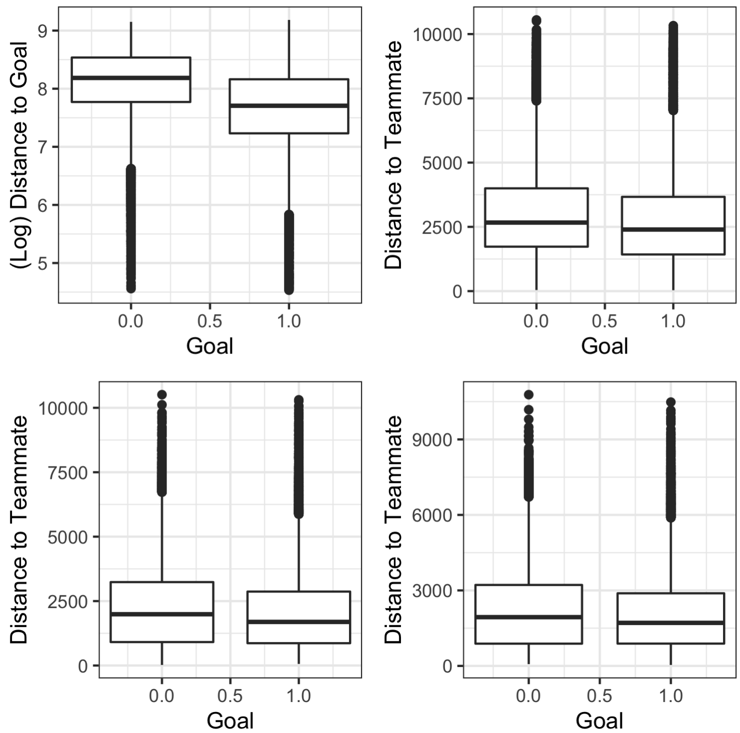

Figure 3: Position Features Correlation Plot

5 Expected Goals Models

We built four expected goals models using logistic regression, penalized regressions (ridge and lasso), and boosted trees. We included two penalized regressions to account for the high dimensionality of the dataset and for variance reduction purposes. 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

Expected goals (xG) is the expected probability of the total number of goals in a soccer match, given the number of attempts and the probability of a goal, computed by considering several factors such as the distance to the goal, the angle between the shot taker and the goal, as well as the part of the body used to score. The xG model was first implemented by bookmakers, but soon was used by data analysts in soccer clubs. By comparing expected goals with the actual goals scored, soccer clubs can evaluate individual player performance (and discover high and low efficiency players) as well as overall team analysis. In the same vein, xG models can be used to evaluate the defensive performances of individual players (such as goalkeepers) and teams. Players and teams with higher opponent xG's compared to actual opponent goals scored are more skilled defensively and vice versa. xG models are important because it quantifies the value of scoring chances, and allows coaches, teams and viewers to identify players and teams that are under-performing as well as over-performing relative to their expected goals.

Because expected goals is a popular concept in soccer, we built an expected goals model for Rocket League, which follows many of the same rules as soccer. The goal of the analysis is ultimately to determine the probability of scoring a goal using a number of location and ball features.

5.2 Feature Engineering

On top of the original features in our dataset, which can be found in the Appendix, we conducted feature engineering to extract meaningful features from our data. These are summarized below.

`logDistanceToGoal`: Log distance between player and the goal

`distanceToOpp1`: Distance between player and opponent 1

`distanceToOpp2`: Distance between player and opponent 2

`distanceToTeam`: Distance between player and team mate

`cos_theta_opp_1`: Cosine of theta, where theta represents the angle between the ball's velocity vector and the distance vector between the ball and opponent 1's position

`cos_theta_opp_2`: Cosine of theta, where theta represents the angle between the ball's velocity vector and the distance vector between the ball and opponent 2's position

These engineered features further introduced collinearity, which influenced our model-making decisions.

5.3 Models

We ran 4 models to predict expected goals: 1) a simple logistic regression with engineered features, 2) a ridge logistic regression, 3) a lasso logistic regression, and 4) an xgBoost classification model. The response variable was `goal`, which is represented in the dataset by a binary variable, and the explanatory variables include various location and positioning features.

5.3.1 Logistic Regression

As a starting point, we built a simple logistic regression with 59 explanatory variables. We removed teammate positioning and car features as including the teammate features resulted in model non-convergence. While we are unsure why including teammate-specific data leads to non-convergence, we assume that this is partially attributable to the collinearity in the dataset. As displayed in the correlation plot (Figure 2), there are correlations between the ball position, the shot taker position, and the teammate position. Extended results are reported in the Appendix.

However, there were a number of significant variables, including: the y and z coordinates of the ball position, the ball velocity, the shot taker location, the opponent's velocity, the distance between the shot taker and the two opponents, and the angle between the shot taker and the two opponents. While the coordinates of the ball position, the shot taker's location, the distance between the shot taker and the two opponents, as well as the angle between the shot taker and the opponents would influence goal probability as expected, it is interesting that ball velocity also impacts xG. The coefficient for the ball velocity in the y and z directions are positive, which implies that shot takers who strike the ball when it's moving faster are more likely to score a goal.

5.3.2 Penalized Regressions

Despite identifying significant variables from the logistic regression, the method utilized 59 explanatory variables (with high intercorrelations), which could lead to a cost in variance, and thus inaccurate predictions. To combat these issues, we built and evaluated shrinkage models, namely ridge regression and lasso regression, with the goal of fitting a more parsimonious and interpretable model. Specifically, ridge regression is more stable when handling correlated features, as it "splits the credit" among correlated features, and lasso regression penalizes many features to 0, contributing to increased interpretability. For both penalized regression methods, we ran a 10-fold cross validation to optimize the choice of regularization parameters (λ).

The lasso regression trace plot is shown in Figure 4 and the selected features and respective coefficients are displayed in Table 1. We applied the one standard error rule to select the optimal λ value, and we notice that the lasso regression selects around 29 variables, including various ball position features, log of the shot taker's distance to the goal, the opponents' positions, and the angle between the shot taker and the goal. There seems to be some overlap in the most important features per the lasso trace plot and the statistically significant features from the logistic regression.

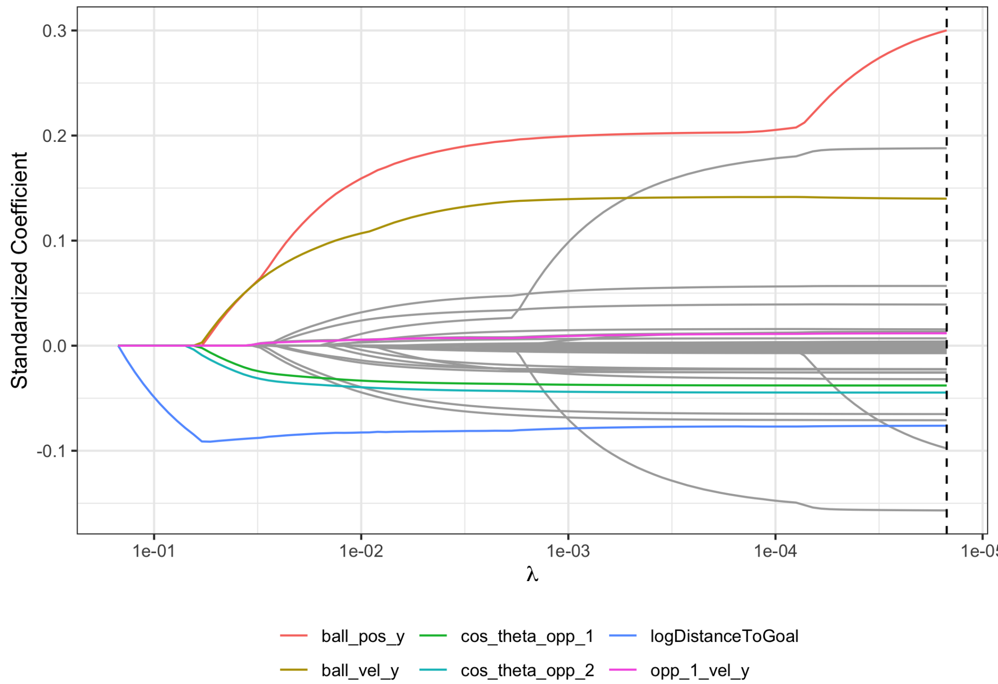

Figure 4: Lasso Regression Trace Plot

Table 1: Standardized coefficients for features in the lasso model based on the one-standard-error rule.

Feature	Coefficient
ball_pos_y	0.30
shot_taker_pos_z	0.19
ball_pos_z	-0.16
ball_vel_y	0.14
shot_taker_pos_y	-0.10
logDistanceToGoal	-0.08
opp_1_pos_y	-0.07
opp_2_pos_y	-0.07
ball_ang_vel_x	0.06
cos_theta_opp_2	-0.04

Following in Table 2 are the results for regression-based methods. All three models have similar misclassification rates, but the logistic regression performs best in terms of log loss, which implies that penalization did not reduce variance and improve predictive power.

Table 2: Regression Methods Summary

Model	Log Loss	Misclassification Rate
Logistic Regression, FE, no teammate data	0.52	0.26
Ridge Regression, FE, no teammate data	0.59	0.26
Lasso Regression, FE, no teammate data	0.62	0.26

5.3.3 XGBoost

We also implemented a gradient boosting model, which is a method of aggregating multiple decision trees to improve prediction performance over a traditional decision tree. Boosting grows shallow decision trees sequentially, by considering a low-complexity weak learner (a shallow decision tree) and boosting the performance of the weak learning by applying an iterative method. For our xgBoost model, we used 200 trees, an interaction depth of 3, and a shrinkage of 0.1 due to computing limitations.

With the given boosting model, we judged variable importance by comparing purity-based importance. The ranking of variables based on purity-based importance is given in Table 3. Similar to the regression-based methods, the shot taker's distance to the goal, the angle between the ball and the opponents, and various ball position and velocity features rank high in terms of variable importance.

Table 3: Boosting Important Variables

Variable	Relative Influence
distanceToGoal	19.80
cos_theta_opp_2	13.11
cos_theta_opp_1	12.62
ball_vel_y	11.28
ball_pos_y	10.97
ball_pos_z	8.62
shot_taker_pos_z	4.48
ball_vel_z	3.87
ball_vel_x	2.22
shot_taker_vel_z	1.65

5.4 Model Evaluation

The final model evaluation summary is given in Table 4. Based on the evaluation statistics, the xgBoost model yields the lowest log loss and lowest misclassification rate, so we select the xgBoost model as our final model to be used in our excess goals analysis.

Table 4: Model Evaluation

Model	Log Loss	Misclassification Rate
Logistic Regression, FE, no teammate data	0.52	0.26
Ridge Regression, FE, no teammate data	0.59	0.26
Lasso Regression, FE, no teammate data	0.62	0.26
xgBoost, FE, with teammate data	0.47	0.23
Naive classifier	0.65	0.35

To visualize the robustness of the model, we recreated images of the field with data points representing each shot taker's attempt location. [discuss: goals at the bottom of the field]

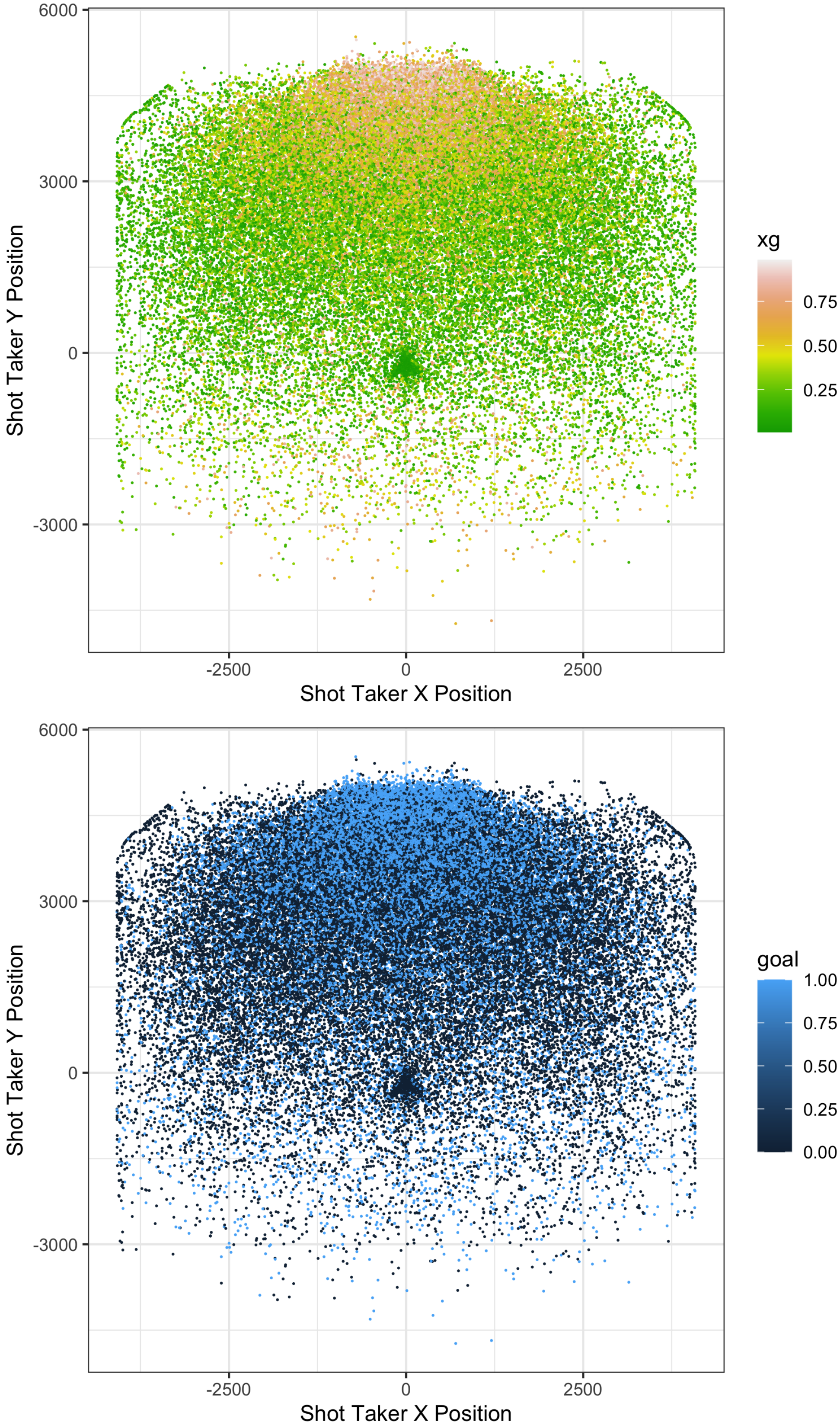

Figure 5: Outcome and xG Field Visualization

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 which are included in our xG boosting model.

For our analysis of excess goals and player outperformance, we selected for shot takers that appeared more than 20 data observations. We believe including shot takers that appeared less frequently will skew the analysis. Because we are interested in understanding what makes a shot taker successful, we looked at the sum of each players' expected goals over all shots in the dataset, the total number of goals actually scored, the number of goal attempts, outperformance, and the actual success rate. A subsample of the aggregated metrics dataset is shown in Table 5.

Table 5: Relevant Statistics for Outperformance Analysis

shot_taker_id	sum_xg	total_goals	count	outperformance	actual_goal_rate	avg_xg
76561198012337838	13.46	24	46	1.78	0.52	0.29
76561198002542052	13.81	22	26	1.59	0.85	0.53
76561199003527262	5.66	9	23	1.59	0.39	0.25
76561198164573553	8.22	13	25	1.58	0.52	0.33
b97fe6ef08b	10.28	16	36	1.56	0.44	0.29

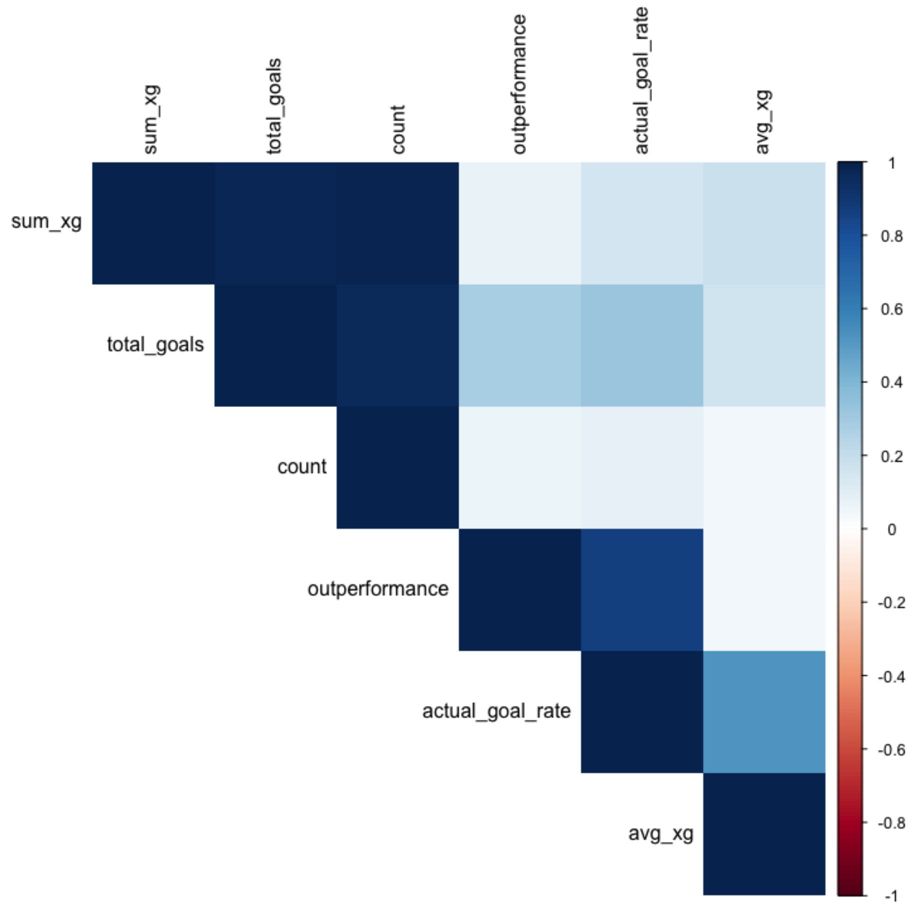

Figure 6: Player-Specific Aggregate Statistics Correlation Plot

To visualize relationships across these player-specific aggregate metrics, we created a correlation plot shown in Figure 6. The sum of the xG across all shot attempts is correlated with total goals and number of attempts, which makes sense as a higher xG is dependent on more shots, and more shots allows for more opportunities for goals. Outperformance is correlated with acual goal rate, which would be expected given a robust model. Moreover, the sum of xG and total goals is correlated, and actual goal rate is correlated with average xG, which are expected results. What's interesting is that outperformance is not strongly correlated with average xG or sum of xG, which suggests that outperformance is a random process, and scoring goals above what is expected can be attributed to covariates other than positioning, like luck and randomness. [CHECK WITH GROUP]

Table 6: Outperformance Mean and SD

Mean	Standard.Deviation
0.98	0.27

To further visualize the relationship between outperformance and average xG per player, we created scatterplots (Figure 7) between average xG and outperformance residual (defined by outperformance - average outperformance across all players) and average xG and outperformance z-score. The average and standard deviation of all players' outperformance scores are given in Table 6. Upon examination, the relationship between xG and the outperformance residual and z-score appear to be random, suggesting that scoring goals above what is expected given the location of all players and various ball features is largely due to luck.

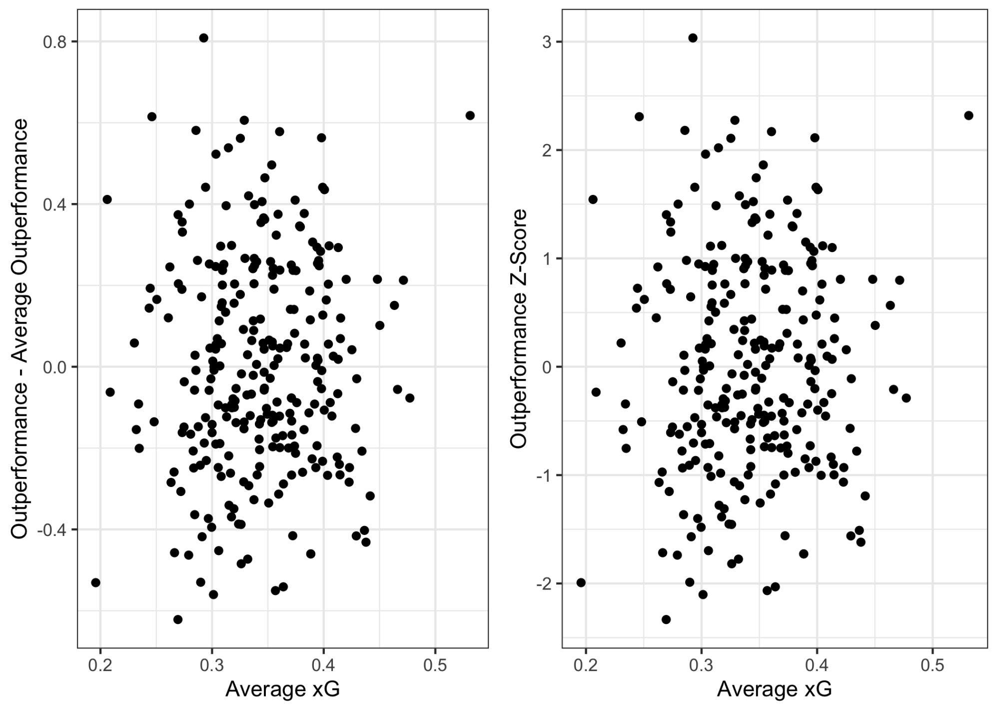

Figure 7: Outerperformance Scatterplots

7 Understanding Offensive and Defensive Playing Styles

In soccer, players are assigned specialized positions such as goal-keeper, midfielder, forward, and striker. Unlike its real-world counterpart, in Rocket League, players use a strategy called "rotations" to position themselves. A common rotation in 2v2 matches involves a player "challenging" the ball, then backing up to cover space in the field while their team mate moves up to receive and "challenge" a ball. Here, "challenging" the ball means to win control over the ball's direction. For our research paper, we will assume that these strategies can be understood as a rotation between defensive and offensive postures. We further assume that the two positions require similar yet slightly different skill sets.

From here, we draw three hypotheses:

H_0 (null): There is no correlation between offensive skill and defensive skill. Since players rotate between offensive and defensive multiple times per game, we may expect that players are spending roughly equal amounts of time in the defensive and offensive positions and there is no relationship between offensive skill and defensive skill. [thus are equally skilled at both – discuss w group]

H_1: There is a positive correlation between offensive skill and defensive skill. Players who are skilled will tend to be better at both the offensive and defensive positions.

H_2: There is a negative correlation between offensive skill and defensive skill. Players have an inclination to be better than either offensive or defensive positions.

Using the xG we calculated, we will use average xG and average xG defended (defined below) as proxies for offensive and defensive skill. We will define xG defended for player u to be equal to the sum of the xG minus the goal outcome (where 1 is a goal and 0 is no goal) for each shot a player u had the opportunity to block.

Naturally, we will define average xG defended for player u to equal xG defended divided by total shots player u had the opportunity to block.

The use of these values as proxies is debatable, and there are strong limitations to both proxies. Average xG is a poor proxy for offensive skill as it only takes into account the player's ability to shoot a successful goal, and does not account for the player's skill in dribbling or passing during an offensive run. Similarly, average xG defended is a poor proxy for defensive skill as it only takes into account the player's ability to block a shot, and does not account for the player's skill in interception.

Our results of this analysis are shown below in Figure 8. Running a simple OLS regression to predict average xG using average xG saved yields an R^2 value of $5.456e^{-}05$, suggesting that there is little correlation. This means that we fail to reject the null hypothesis, and conclude that there is no correlation between offensive and defensive skill and players roughly spend equal amounts of time rotating between an offensive and defensive position.

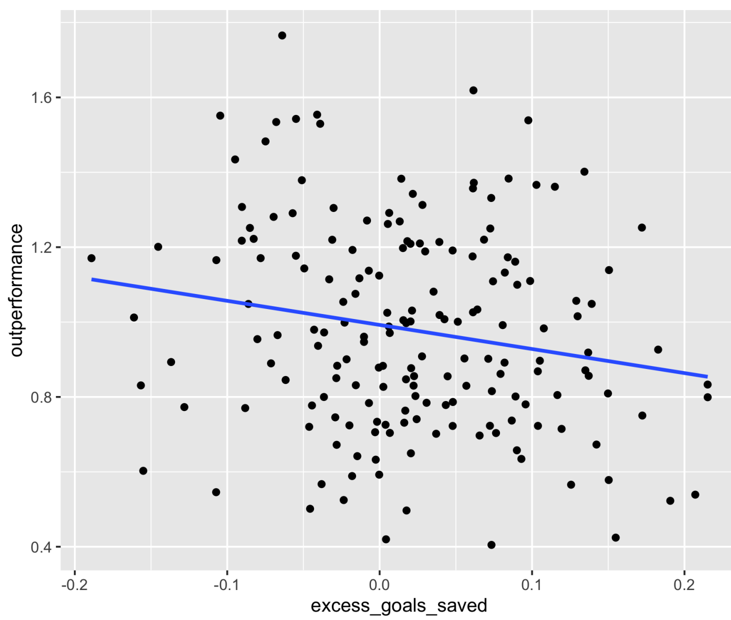

Figure 8: Average xG Defended vs. Average xG

8 Rocket League Championship Series Top Match Analysis

The Rocket League Championship Series (RLCS) is a world championship event hosted by Psyonix, the company that developed Rocket League. To examine top player play styles, we applied the boosting model on the 2020 2v2 Championship games in the Europe and North American region. Within each region, each game consisted of two teams competing against each other in a best-of-3 match-up. This gave us a total of 6 games worth of data to analyze. Note that we ignored goals whose last-hit occurred before the game-time was over, but resulted in a score after the game-time was over. This results in small differences between the official published goals and our goals.

For our case study, we will focus on a few surprising findings based on our analysis.

8.1 European 1st Match

The match details are shown in Figure 9.

European Match 1			
Team Tortilla		Team Dignitas	
Final Score: 2 v 7			
itachi	Stake	ApparentlyJack	joreuz

Figure 9: European 1st Match Details

Table 7: European 1st Match Player Statistics

shot_taker	sumXG	sumGoals	actualShots	goalRate	avgXG
ApparentlyJack	1.09	3	6	0.50	0.18
itachi	2.06	1	3	0.33	0.69
joreuz	0.57	4	4	1.00	0.14
Stake	2.01	1	4	0.25	0.50

With a final score of 7-2 for Team Dignitas, the first match of the European championship reflected poorly on Team Tortilla. The analysis of the players' xG, however, tells a very different story.

Notably, the players on the winning team Dignitas had significantly worse shot-quality than the players on the losing team Tortilla. Itachi and Stake have average xG scores that are factors of two to four times greater than their counterparts', indicating the high quality of their positioning, and yet the actual score does not reflect this. What might explain this discrepancy?

Team Tortilla and Dignitas made a total of 7 and 10 shot attempts, respectively. **The difference is small, so the number of goal attempts does not seem to have massively contributed to the large gap in score.** This means that the score differential is likely attributable to defensive ability: it could be that Team Tortilla was just lousy at defense, letting too many balls in that they could have likely blocked or that Team Dignitas may have had an excellent defense. Or, it could have been pure luck.

8.2 European 3rd Match

The match details are shown in Figure 10.

European Match 3			
Team Tortilla		Team Dignitas	
Final Score: 3 v 2			
itachi	Stake	ApparentlyJack	joreuz

Figure 10: European 3rd Match Details

Table 8: European 3rd Match Player Statistics

shot_taker	sumXG	sumGoals	actualShots	goalRate	avgXG
ApparentlyJack	0.87	1	2	0.50	0.44
itachi	3.49	3	10	0.30	0.35
joreuz	0.76	1	4	0.25	0.19
Stake	1.43	0	4	0.00	0.36

The 3rd match-up between team Tortilla and team Dignitas was a nail-biter. The two teams were tied at 2-2. At the very last moment of the 5-minute regulation period, with 0 seconds left on the clock, Team Tortilla's player itachi scored the team's 3rd goal, preventing an overtime period and winning the match. Throughout the game, the two teams seemed evenly-matched, neck-to-neck.

Analyzing the players' statistics gives us more insight into this game. Players Itachi and Stake of Team Tortilla had made far more shots than Team Dignitas, at 14 cumulative shots compared to 6. This was offset by the fact that Team Dignitas's player ApparentlyJack had better positioning with each shot, with the highest average xG per shot of 0.436 in the game. Though the two Dignitas teammates had the same number of goals, ApparentlyJack's average xG and thus his shot quality far exceeded his teammate's. At an average xG of 0.436, ApparentlyJack's xG was more than twice as high as his teammate's.

Ultimately, Team Tortilla had a cumulative xG of 4.92 versus Team Dignitas' 1.63, so perhaps Team Tortilla's eventual win is not too surprising and the narrow gap between the two teams was a result of luck on Dignitas's part.

8.3 North American 3rd Match

The summary results of the 3rd North American match (Figure 11) show that 2Piece and MaJicBear, both of whom were on the winning team 'LAST MINUTE', had an average xG that surpassed their actual goal rate.

<table>
<tr><td colspan="4">North American Match 3</td></tr>
<tr><td colspan="2">Team LAST MINUTE</td><td colspan="2">Team AFTERTHOUGHT</td></tr>
<tr><td colspan="4">Final Score:
1 v 0</td></tr>
<tr><td>2Piece</td><td>MaJicBear</td><td>Lj</td><td>Shock</td></tr>
</table>

Figure 11: North American 3rd Match Details

Table 9: European 3rd Match Player Statistics

shot_taker	sumXG	sumGoals	actualShots	goalRate	avgXG
2Piece	1.19	0	2	0.00	0.60
Lj	0.20	0	3	0.00	0.07
MaJicBear	1.90	1	3	0.33	0.63
Shock	0.15	0	3	0.00	0.05

The 3rd match-up in the North American championship was even more of a nail-biter than the last one. The two teams had each won 1 of the 3 matches, meaning the final match was the chamption-decider [check linda]. Neither teams had been able to score, and with the two teams were tied at 0-0 at the end of regulation time, the match went into sudden-death overtime. The first team to make a goal would win the match and thus the entire championship. Ultimately, Team LM was able to first shot and win.

The dearth of successful goals makes it difficult to analyze each players' performance throughout the match, but the analysis of their average xG enables us to value the shots that didn't result in a score.

One thing that is strikingly obvious is that the players on the winning Team LM had a much higher average xG than their counterparts on the losing team. 2Piece and MaJicBear's average xG per shot was more than a multiple of 10 larger than Lj and Shock's average xG, speaking to their ability to position themselves. The two teams only had a difference of one shot made, suggesting that the major factor in Team LM winning the game was their ability to position their shot.

9 Conclusions

In sum, we built an expected goals model using xgBoost to predict goal probability given various location and ball features. Then we extended our xG model to analyze the drivers behind outperformance and its relationship with the actual goal rate. We explored Rocket League rotations to understand whether skilled players good at both defense and offense or have a "niche" for defense or offense, or if there is no relationship. Then we concluded our analysis by applying our xG model to analyze top Rocket League Championship Series games. Our findings are as follows:

1. Outperformance, defined by the total goals scored divided by the sum of the player's xG, is not correlated with average xG, suggesting that there is no relationship between how many "excess" goals a player can score based on what we would expect given positioning features, and the average positioning across all of the player's attempts. This suggests that outperformance is largely due to luck or randomness.
2. Players rarely adopt a strictly offensive or strictly defensive position, and offensive and defensive ability do not correlate. This supports the idea that most play styles involve moving around a field and being a dynamic player.
3. Average xG and number of shot attempts are not always the only contributing factors to determining the outcome of a match. There are many other features, like luck and defensive ability, that influence the outcome.

10 Appendix

10.1 Subset of location features used in models

Table 10: 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

10.2 Logistic regression summary output

```
##
## 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
```