Rocket League Analysis

Katie Lee, Hugo Leo, Linda Wang

March 14, 2022

Contents

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

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 $(0, 5000, 0)$.

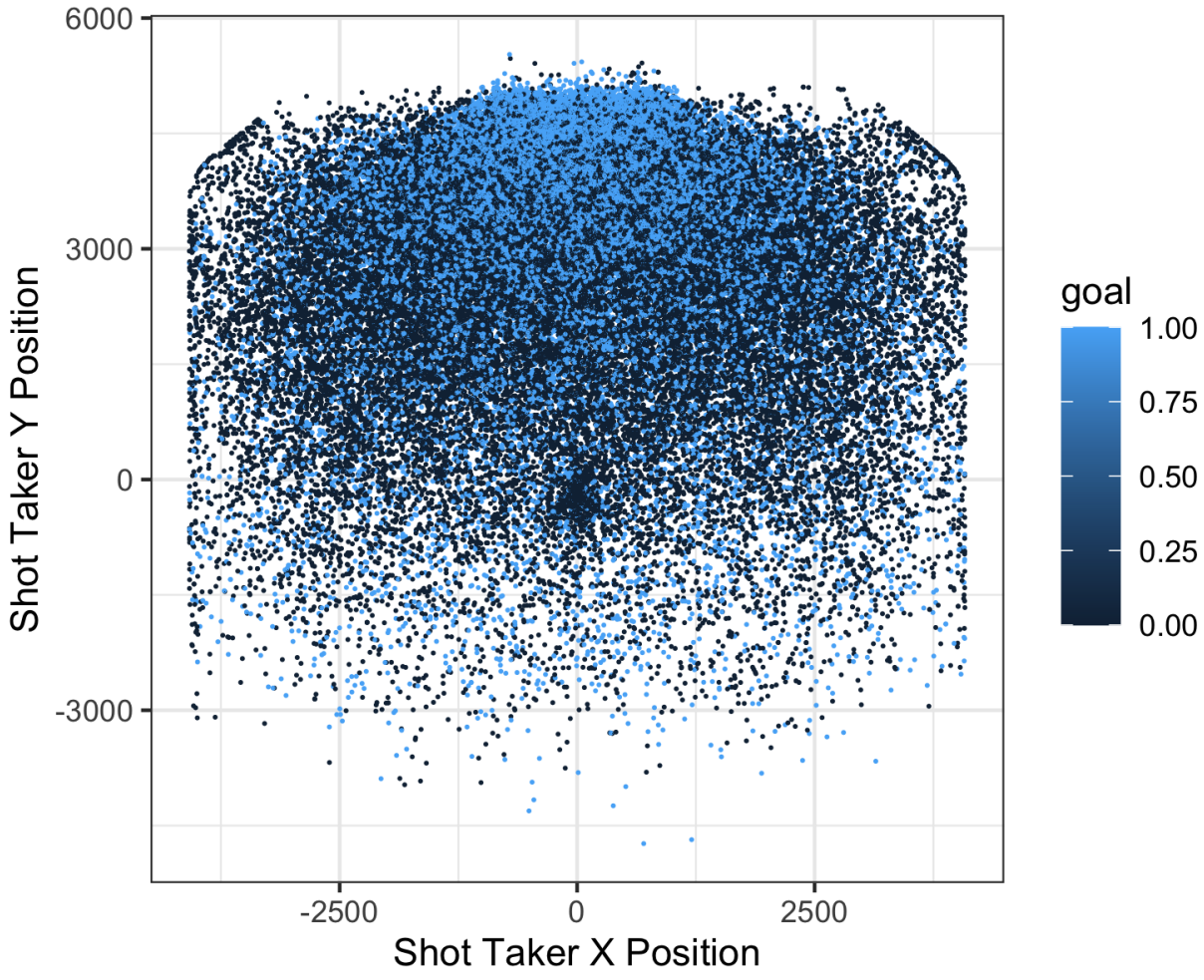

Figure 1: X-Y Field and Shot Outcome

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.

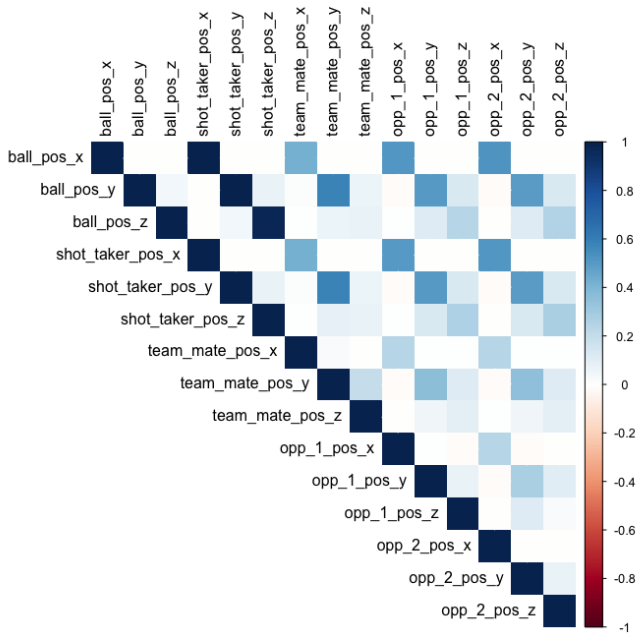

Figure 2: Position Features Correlation Plot

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 noticable 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.

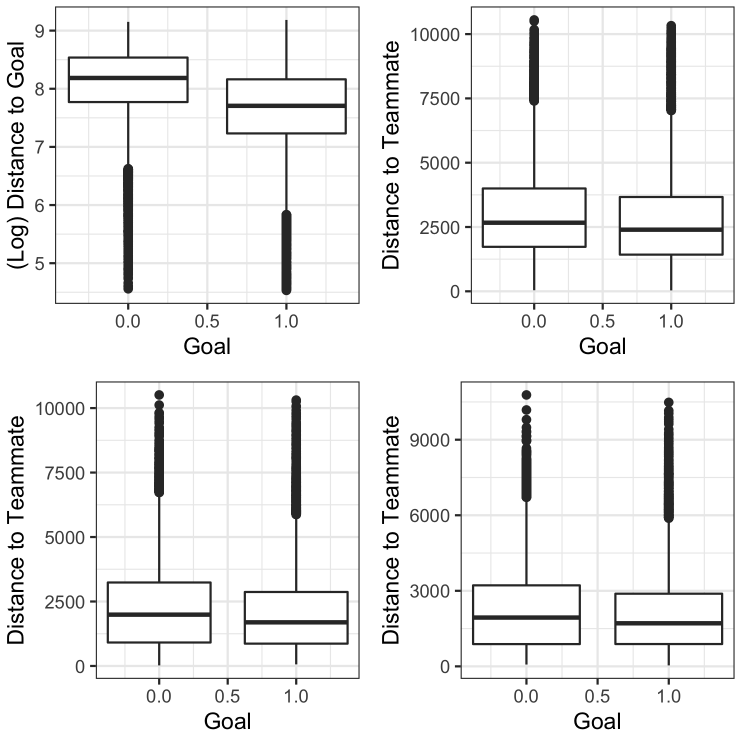

Figure 3: Position Features Correlation Plot

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

- 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 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 @ref(fig: position-corrplot)), 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 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.

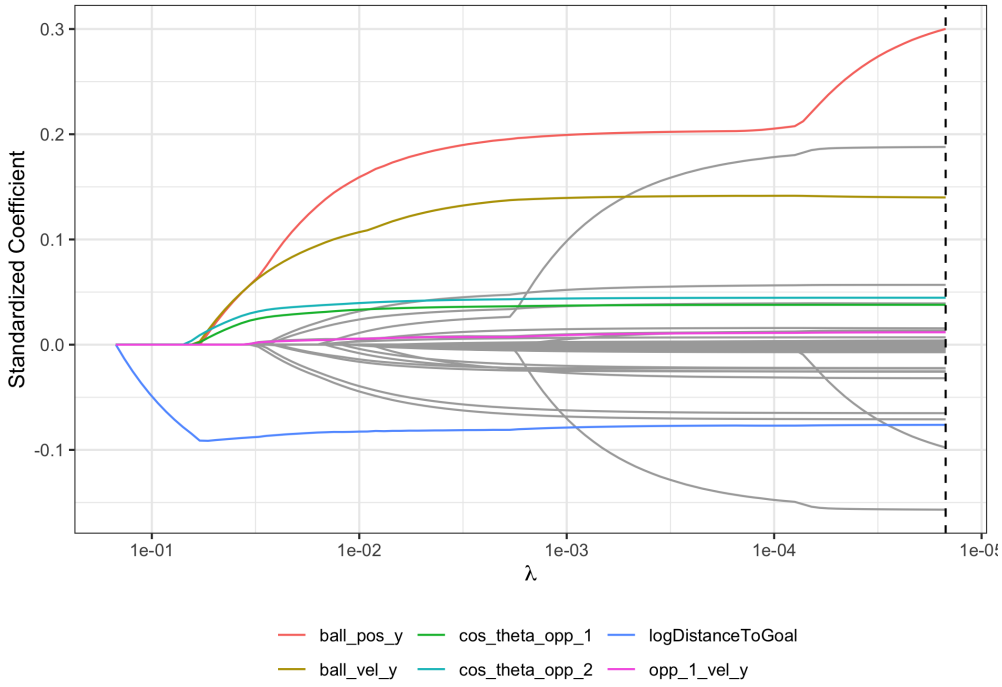

Figure 4: Lasso Regression Trace Plot

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:

Table 2: Regression Methods Summary

Model	Log Loss	Misclassification Rate
Logistic Regression, FE, no teammate data	0.52	0.29
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 method, the shot taker's distance to the goal, the angle between the shot taker and the goal, 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	0.29
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	0.46	0.22
Naive classifier	0.65	0.35

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.

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
76561198012337838	13.46	24	46	1.78	0.52
76561198002542052	13.81	22	26	1.59	0.85
76561199003527262	5.66	9	23	1.59	0.39
76561198164573553	8.22	13	25	1.58	0.52
b97fe6ef08b	10.28	16	36	1.56	0.44

7 Appendix

7.1 Subset of location 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 6: 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

7.2 Logistic regression summary output

```
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
```