# math_158_semesterproject_part4

Kevin Loun

5/9/2022

## Introduction

The dataset for this project contains 10,000 League of Legends ranked matches from the North American region with 775 variables offered through the Riot Games API, provided on Kaggle [@riot][@james_2020]. Each match is pulled from players who rank Gold in the League system, a ranking system that matches players of a similar skill level to play with and against each other. Amongst North American players, the Gold skill level was the second most common tier, achieved by 27.7 percent of players, or approximately 49.86 million players when considered against Riot Games' player base of 180 million [@statista_2021][@riot_tweet]. This dataset will be referred to as `lol10`.

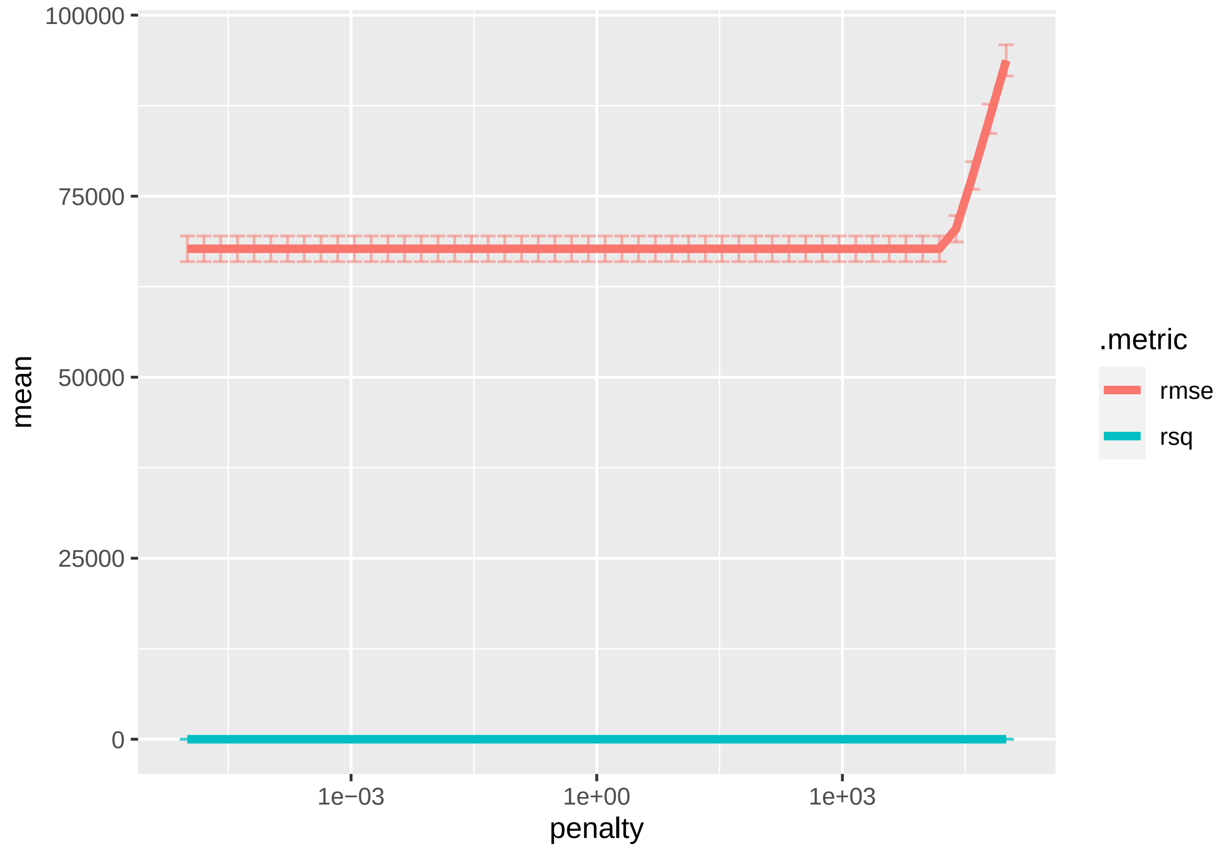
For this project, the following variables are of interest: time spent crowd controlling others, map side, longest time spent living, kills, gold earned, and total damage dealt. A figure including all the relevant variables and their description is attached at the end.

## Normalizing Data

Since we are running a Ridge Regression and LASSO model on our data we need to ensure that our data is normalized to ensure that all variables contribute equally to the penalized coefficients in our models.

## Ridge Regression

```
## # A tibble: 50 x 7
##      penalty .metric .estimator   mean     n std_err .config
##        <dbl> <chr>   <chr>       <dbl> <int>   <dbl> <fct>
##  1 0.00001   rmse    standard   67734.    10   1772. Preprocessor1_Model01
##  2 0.0000160 rmse    standard   67734.    10   1772. Preprocessor1_Model02
##  3 0.0000256 rmse    standard   67734.    10   1772. Preprocessor1_Model03
##  4 0.0000409 rmse    standard   67734.    10   1772. Preprocessor1_Model04
##  5 0.0000655 rmse    standard   67734.    10   1772. Preprocessor1_Model05
##  6 0.000105  rmse    standard   67734.    10   1772. Preprocessor1_Model06
##  7 0.000168  rmse    standard   67734.    10   1772. Preprocessor1_Model07
##  8 0.000268  rmse    standard   67734.    10   1772. Preprocessor1_Model08
##  9 0.000429  rmse    standard   67734.    10   1772. Preprocessor1_Model09
## 10 0.000687  rmse    standard   67734.    10   1772. Preprocessor1_Model10
## # ... with 40 more rows
```




```
## # A tibble: 1 x 2
##   penalty .config
##     <dbl> <fct>
## 1 0.00001 Preprocessor1_Model01
```

```
## # A tibble: 2,504 x 1
##       .pred
##       <dbl>
##  1 520316.
##  2 636201.
##  3 595533.
##  4 750419.
##  5 590899.
##  6 471284.
##  7 597319.
##  8 253291.
##  9 574866.
## 10 509356.
## # ... with 2,494 more rows
```

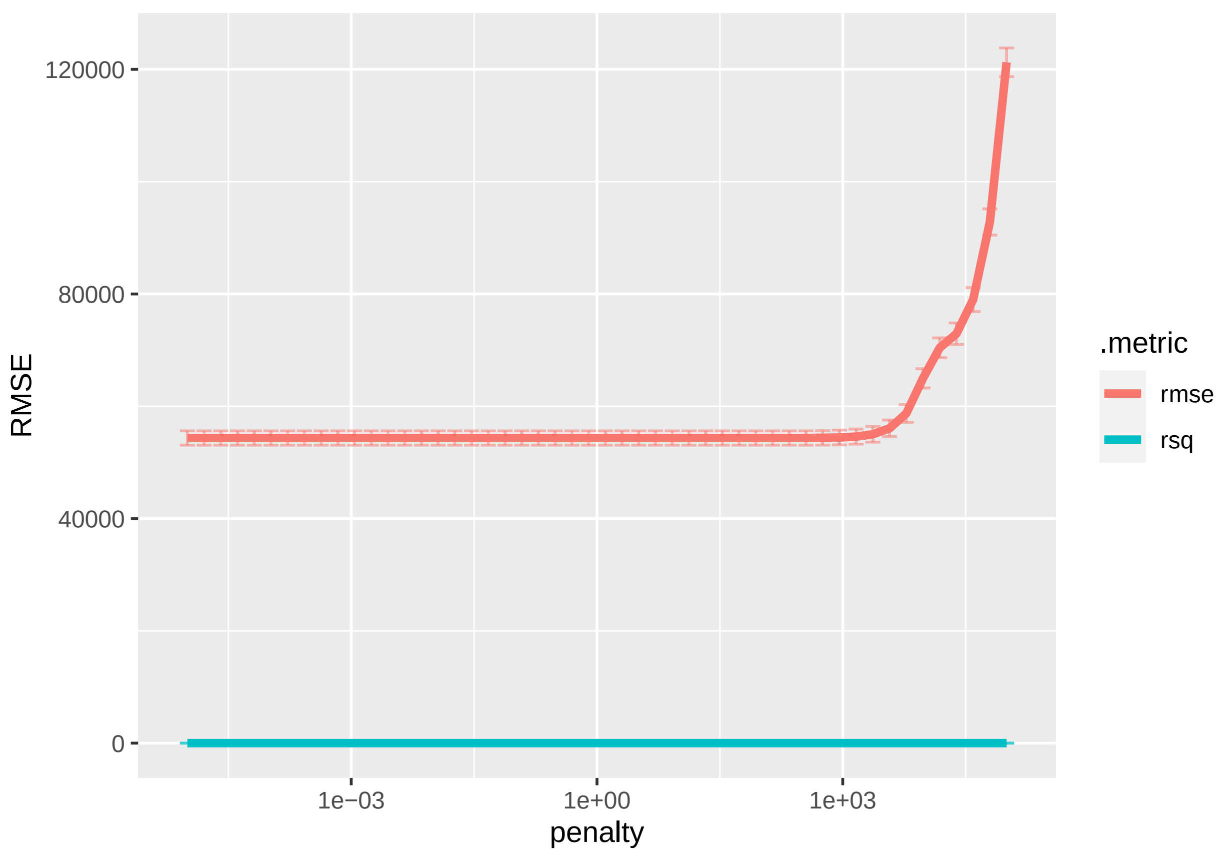
```
# Final Model
finalize_workflow(ridge_wf %>% add_model(ridge_spec_tune), best_rr) %>%
  fit(data = lol10_test) %>% tidy()
```

```
## # A tibble: 5 x 3
##   term                          estimate penalty
##   <chr>                            <dbl>   <dbl>
```

```
## 1 (Intercept)                   551340. 0.00001
## 2 b_gold_earned                 180281. 0.00001
## 3 b_kills                       -17192. 0.00001
## 4 b_time_c_cing_others           16638. 0.00001
## 5 b_longest_time_spent_living    15542. 0.00001
```

## LASSO Regression

```
## # A tibble: 50 x 7
##      penalty .metric .estimator   mean     n std_err .config
##        <dbl> <chr>   <chr>       <dbl> <int>   <dbl> <fct>
##  1 0.00001   rmse    standard   54345.    10   1272. Preprocessor1_Model01
##  2 0.0000160 rmse    standard   54345.    10   1272. Preprocessor1_Model02
##  3 0.0000256 rmse    standard   54345.    10   1272. Preprocessor1_Model03
##  4 0.0000409 rmse    standard   54345.    10   1272. Preprocessor1_Model04
##  5 0.0000655 rmse    standard   54345.    10   1272. Preprocessor1_Model05
##  6 0.000105  rmse    standard   54345.    10   1272. Preprocessor1_Model06
##  7 0.000168  rmse    standard   54345.    10   1272. Preprocessor1_Model07
##  8 0.000268  rmse    standard   54345.    10   1272. Preprocessor1_Model08
##  9 0.000429  rmse    standard   54345.    10   1272. Preprocessor1_Model09
## 10 0.000687  rmse    standard   54345.    10   1272. Preprocessor1_Model10
## # ... with 40 more rows
```




```
## # A tibble: 1 x 2
##   penalty .config
```

```
##     <dbl> <fct>
## 1 0.00001 Preprocessor1_Model01

## # A tibble: 2,504 x 1
##      .pred
##      <dbl>
##  1 542617.
##  2 643366.
##  3 584414.
##  4 761663.
##  5 626902.
##  6 445699.
##  7 524496.
##  8 263219.
##  9 552366.
## 10 515471.
## # ... with 2,494 more rows
```

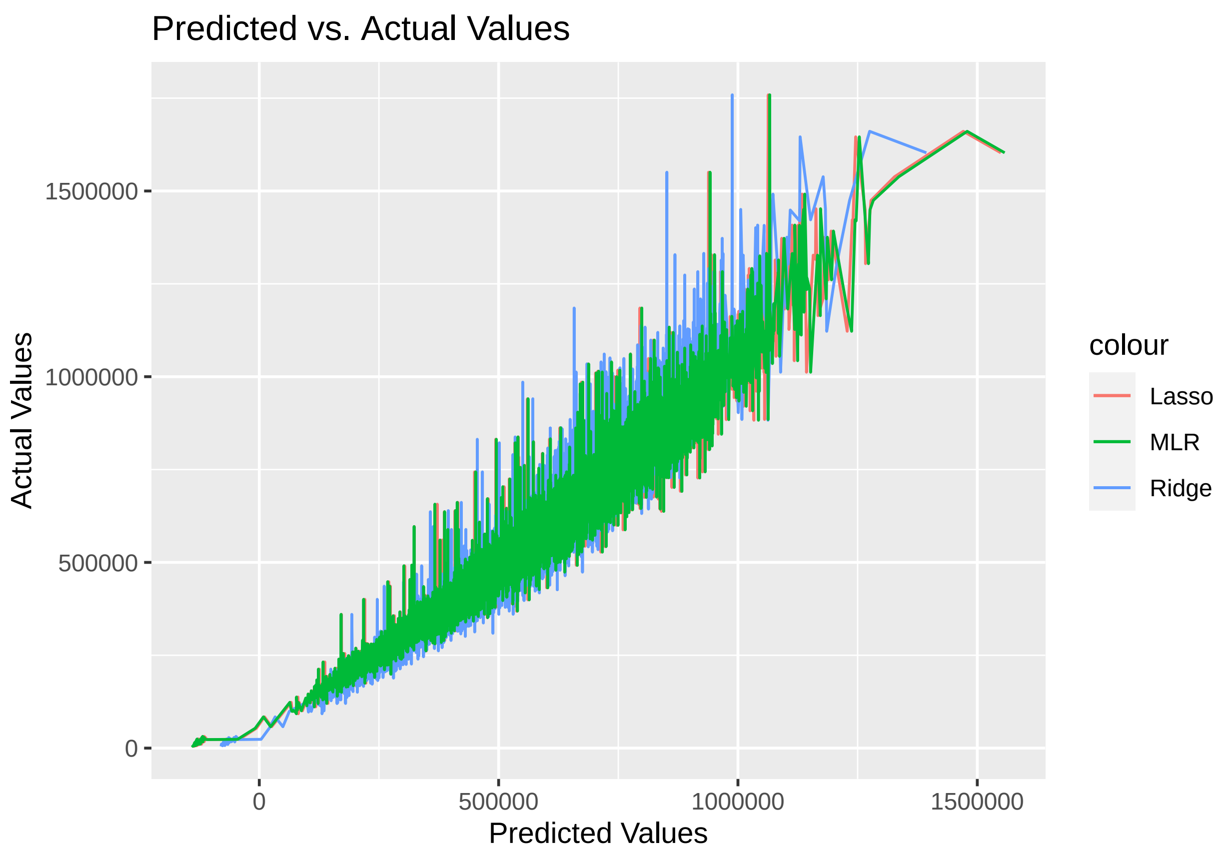
```
# Final Model Coefficients
finalize_workflow(lasso_wf %>% add_model(lasso_spec_tune), best_lasso) %>%
  fit(data = lol10_test) %>% tidy()
```

```
## # A tibble: 5 x 3
##   term                        estimate penalty
##   <chr>                          <dbl>   <dbl>
## 1 (Intercept)                  551340. 0.00001
## 2 b_gold_earned                269336. 0.00001
## 3 b_kills                      -79530. 0.00001
## 4 b_time_c_cing_others              0  0.00001
## 5 b_longest_time_spent_living    -659. 0.00001
```

## Comparing Models

## Plotting Predicted vs Actual for 3 Models

```
#MLR Model
MLR_lm <- lm(b_total_damage_dealt ~ b_gold_earned + b_kills + b_longest_time_spent_living + b_time_c_ci
ggplot() +
  geom_line(lol10, mapping=aes(x=unlist(predict(lasso_fit,lol10)), y=b_total_damage_dealt, color = "Las
  geom_line(lol10, mapping=aes(x=unlist(predict(ridge_fit,lol10)), y=b_total_damage_dealt, color = "Rid
  geom_line(lol10, mapping=aes(x=unlist(predict(MLR_lm, lol10)), y=b_total_damage_dealt, color= "MLR"))
  labs(x='Predicted Values', y='Actual Values', title='Predicted vs. Actual Values')
```




**TO DO**

The first section includes applications of the ideas from the mathematical optimization models covered after MLR (e.g., ridge regression, LASSO, smoothing splines, kernel smoothers). The report should include: ~~• Introduction (briefly refresh the reader's mind as to the variables of interest). Remember that you should include a reference for the original data source, and the reader should know to what population you are inferring your results.~~ • Run both ridge regression and LASSO on the full variable set (use cross validation to find lambda). Compare and contrast the models (i.e., coefficients) with the final MLR model from the previous project assignment. ~~• Make a single plot with the observed response variable on the x-axis and the predicted response variable on the y-axis. Overlay (using color with a legend) 3 different predictions: MLR, RR, LASSO. Comment on the figure.~~ • Choose a single variable and run both smoothing spline and kernel smoother models. Change the parameters so that you have at least four different models for each method. • Plot the (8+) smoothed curves on either one plot or two plots (depending on which looks better for your data. Comment on the figure(s). • Without cross validating, which of the 8 smoothed models would you choose to use for future predictions? Your argument might include smoothness, interpretation of coefficients, ability to include variability of the predictions, etc. • A Conclusion (Summarize your results. Comment on anything of interest that occurred. Were the data approximately what you expected or did some of the results surprise you? What other questions would you like to ask about the data?)