

PSTAT 131 Final Project

University of California, Santa Barbara

Author: Chenhan Xu

Supervisor: Prof. Katie Coburn

November 11, 2022

By Chenhan Xu

Introduction

League of Legends is one of the most famous competitive computer games in the world nowadays. This project aims to find an accurate machine learning model to predict whether a team loses or wins based on the game stat in the first 10 minutes of the game. The dataset I chose records a number of game statistics for the blue team at 10 minutes of the game. All data were collected from high-diamond to master ranked games.

What is League of Legends?

League of Legends is an online competitive video game that requires 10 players per game. The 10 players are separated into two teams (blue and red) to fight each other. Each player is able to control a champion with unique abilities and features. The goal of the teams is to destroy the enemy base. During the match, players are able to gain gold and experience from multiple sources including killing minions, destroying enemy defensive measures, and killing enemy champions, etc. to enhance the ability of their champions. Also, there are special monsters in the map that can provide additional buff for the team that killed them. Players need to build up strategies and gain advantages through various ways in order to win the game.

Why is the model relevant?

As one of the most famous video games in the world, there are countless League of Legends tournaments every year around the globe. Due to the complexity of the game, one cannot

easily predict the result of the game. However, the audiences are often curious about which team has more chance of winning. A well-performing machine learning model might provide a reliable result for that question. Moreover, by analyzing the importance of different factors that affect the process of the game, professional teams can decide their strategy based on the model and focus on the objectives that increase the probability of winning most.

Loading Data and Packages

```
library(readr)
library(dplyr)
library(tidyverse)
library(tidymodels)
library(ggplot2)
library(corrplot)
```

```
setwd("C:/Users/Adam/Desktop/PSTAT 131/PSTAT-131/Final Project")
original_data <- read_csv("high_diamond_ranked_10min.csv")
```

```
## Rows: 9879 Columns: 40
## -- Column specification -----
## Delimiter: ","
## dbl (40): gameId, blueWins, blueWardsPlaced, blueWardsDestroyed, blueFirstBl...
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

Data Cleaning

```
lol <- original_data %>% dplyr::select(blueWins, blueFirstBlood, blueKills, blueDeaths,
                                       blueDragons, blueHeralds, blueTotalMinionsKilled,
                                       blueGoldDiff, blueExperienceDiff, blueWardsDestroyed)
head(lol)
```

```
## # A tibble: 6 x 10
##   blueWins blueFirstBl~1 blueK~2 blueD~3 blueD~4 blueH~5 blueT~6 blueG~7 blueE~8
##   <dbl>      <dbl>    <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
## 1      0          1      9      6      0      0     195     643     -8
## 2      0          0      5      5      0      0     174    -2908   -1173
```

```
## 3      0      0      7      11      1      0      186      -1172      -1033
## 4      0      0      4       5      0      1      201      -1321        -7
## 5      0      0      6       6      0      0      210      -1004       230
## 6      1      0      5       3      1      0      225       698       101
## # ... with 1 more variable: blueWardsDestroyed <dbl>, and abbreviated variable
## #   names 1: blueFirstBlood, 2: blueKills, 3: blueDeaths, 4: blueDragons,
## #   5: blueHeralds, 6: blueTotalMinionsKilled, 7: blueGoldDiff,
## #   8: blueExperienceDiff
```

predictors:

blueFirstBlood: whether the blue team gets the first kill of the game or not (0 or 1)

blueKills: the number of champion kills that the blue team gets by 10 minutes of the game

blueDeaths: the number of deaths of blue team champions

blueDragons: whether the blue team gets the first dragon or not (0 or 1)

blueHeralds: whether the blue team gets the first Heralds or not (0 or 1)

blueTotalMinionsKilled: the number of minions killed by blue team champions at 10 minutes

blueGoldDiff: the difference in total gold between blue team and red team

blueExperienceDiff: the difference in total champion experience between blue team and red team

blueWardsDestroyed: the number of red team wards destroyed by blue team at 10 minutes

```
lol$blueWins <- as.factor(lol$blueWins)
lol$blueFirstBlood <- as.factor(lol$blueFirstBlood)
lol$blueDragons <- as.factor(lol$blueDragons)
lol$blueHeralds <- as.factor(lol$blueHeralds)
```

Data Split

```
set.seed(3435)
lol_split <- initial_split(lol, prop = 0.75,
                           strata = blueWins)

lol_train <- training(lol_split)
lol_test <- testing(lol_split)
```

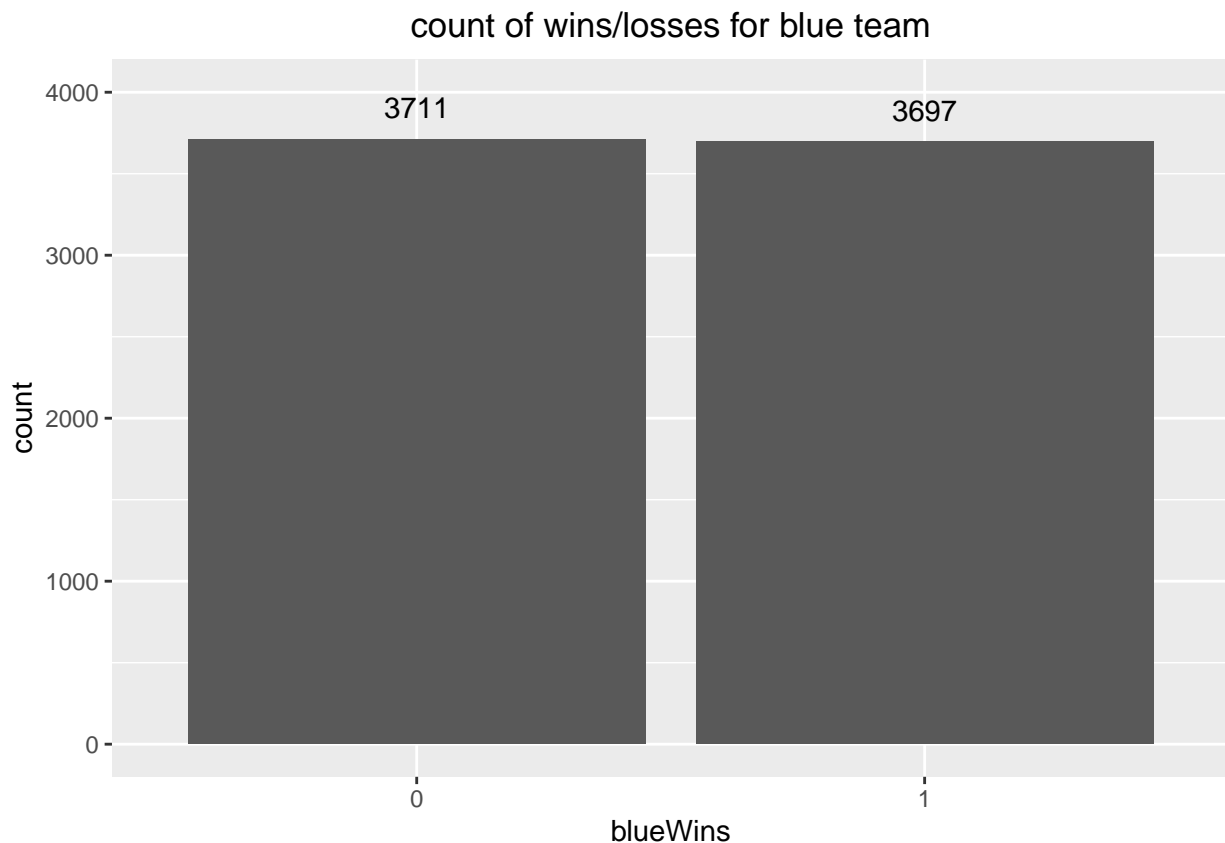
I chose to split the data with proportion of 0.75 so that the training set contains 7408 observations while the testing set contains 2471 observations. The amount of observations should be enough for my model.

Exploratory Data Analysis

The entire exploratory data analysis part will only based on the training set.

Barplot

```
ggplot(lol_train, aes(x = blueWins)) + geom_bar() + geom_text(stat='count', aes(label =  
  coord_cartesian(ylim = c(0,4000)) +  
  ggtitle("count of wins/losses for blue team") +  
  theme(plot.title = element_text(hjust = 0.5)))
```

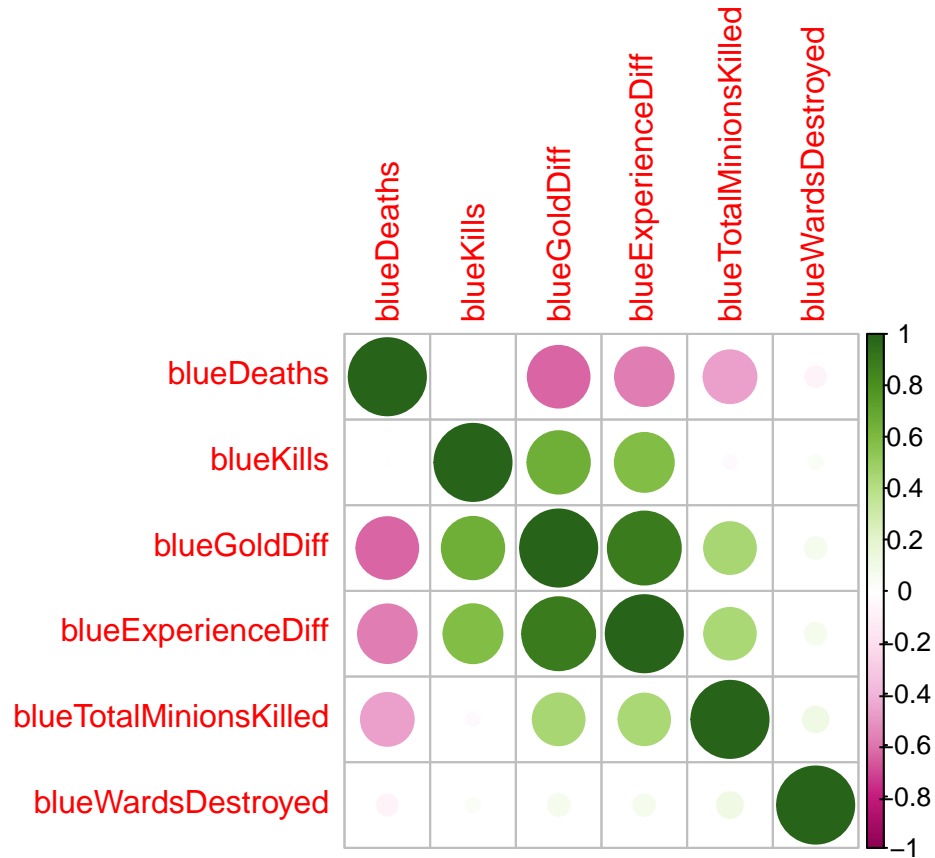


The bar plot shows that blue team wins 3697 games while losing 3711 games. The ratio of win v.s. lose is quite close to 1:1.

Correlation matrix

```
lol_num_var <- lol %>% select_if(is.numeric) # Select only the numeric variables
```

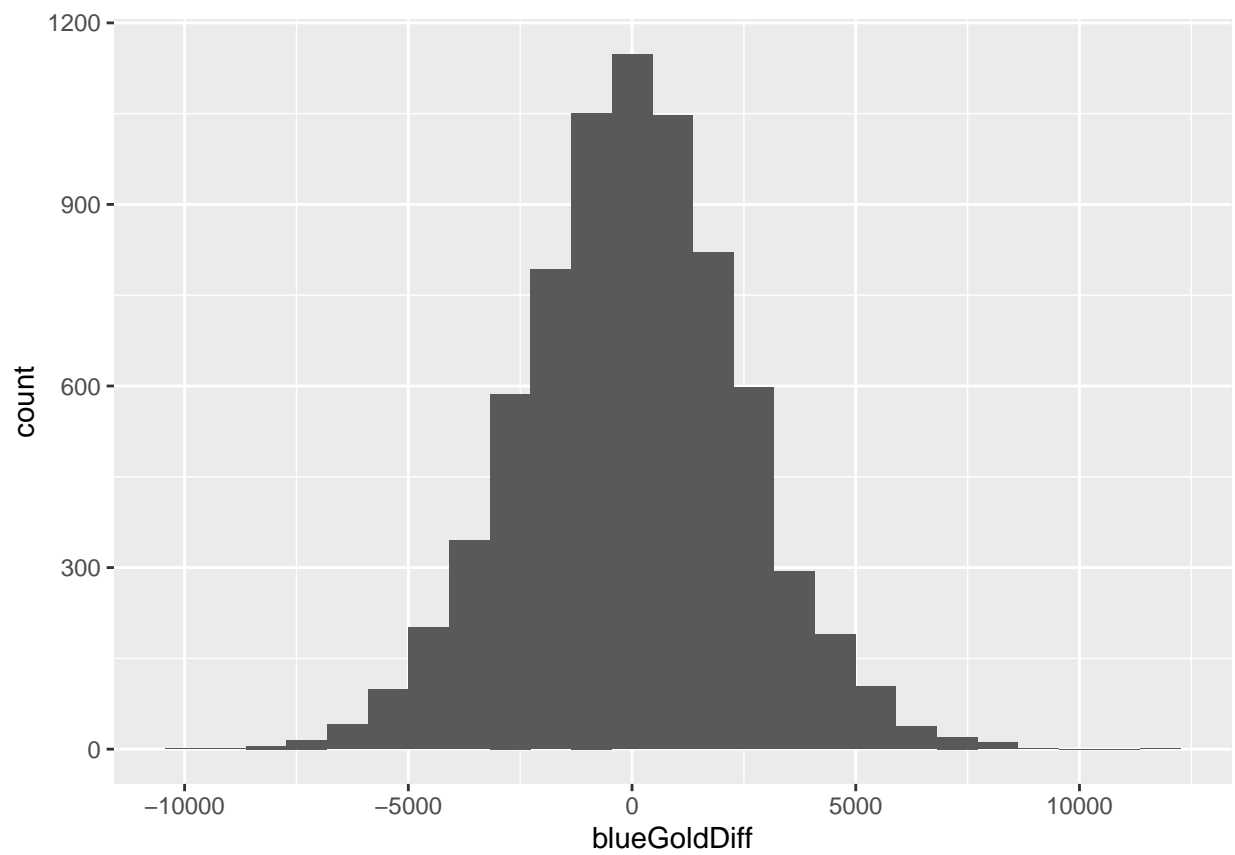
```
lol_cor <- cor(lol_num_var)
lol_cor_plt <- corrplot(lol_cor,
                        order = 'AOE',
                        col = COL2("PiYG")) # Plot correlation matrix
```



From the correlation matrix, we can see that blueGoldDiff and blueExperienceDiff have relatively strong positive correlation. It makes sense because gold and experience should be positively correlated as they both imply the advantage of the blue team. In my case, both gold and experience are represented by the difference between blue team and red team, so it is reasonable that the two variables have (strong) positive correlated.

Variable: blueGoldDiff

```
ggplot(lol_train, aes(x = blueGoldDiff)) + geom_histogram(bins = 25)
```



The variable blueGoldDiff

Vairable: blueExperienceDiff

```
ggplot(lol_train, aes(x = blueExperienceDiff)) + geom_histogram(bins = 25)
```

