Best Ways To Get An Early Advantage In League of Legends

STA304 student

December 9, 2020

Keywords: logistic regression, League of Legends, Early game advantage

Abstract

Winning a game of League of Legends is difficult and requires a great amount of decision making. While there so many things to do in the game, it is common to see players not know what to prioritize and that will often cause them to fall behind in the game. Particularly in the first 10 minutes of the game, it is important to get an early lead over the opponent to secure big objectives later on. We studied the data collected from over thousands of previous high-ranked games. By using the logistic regression to predict the whether a team will win based on what they have accomplished, we found out that while there are many aspects to the game that can make a team stronger, there is only so much that can be done in the given time and teams will need to prioritize enemy kills and early dragons to maximize their advantage over the other team. This advice does not only apply to the high-ranked players, but also all players of any rank as we have adjusted our model to fit the general skill level of the playerbase.

Introduction

If you have played or looked at any video games recently, you may have heard of League of Legends. League of Legends (LoL) is one of the most popular online multiplayer games with an enormous e-sports scene. Chances are that you have played the game before, or you may know people that play the game, or you may be curious on what this game is all about. In any case, LoL is a strategic game that involves a great amount of teamwork and coordination. While games typically last 20-40 minutes, the first 10 minutes of the game (the early game phase) is arguably the most important part of the game because it sets up small advantages that snowball into large advantages, ultimately leading to victory. The challenging part of the game is decision making because every game is different and you will need to adapt to various situations. By analyzing the data of previous games, we will research about what are the general goals and actions that should be prioritized in the early game to get ahead.

This report will be broken down into: a data section where the datasets are described, a model section, a results section which will show summary statistics and model performance, and a discussion section to interpret the findings of the model results as well as why the results are the way they came to be and what are some caveats and weaknesses to consider for future work.

Data

The dataset used for this report will be the first 10 minutes statistics of each team (team red vs team blue) for diamond-master ranked games (https://www.kaggle.com/bobbyscience/league-of-legends-diamond-ranked-games-10-min). This is the rank where many streamers/pros play at and it is where most players take the game seriously and play very often with good understanding of the game. This is important because it will improve the quality of data as there will be less variance of high-level play versus low-level play that is present in lower ranks, making it easier to do an accurate analysis.

The data is collected through the Riot API. Riot is the company the runs the game and they store information of every game and makes this data available through their API. The data was filtered through the ranks (diamond 1 to master) and 9879 games were randomly selected. Each game is unique and has their own gameId.

The population being studied is the upper-half the entire playerbase and the sampling frame is rank Diamond 1 to Master which represents on about 5% of the playerbase. The reason why I am using this frame to represent the population is because many players in the mid-tier divisions (platinum and even gold ranks) were previously diamond players from past years or spend enough time and effort in the game to be close enough to reach Diamond rank.

The columns consists of whether blue team won the game, team comparisons in the first 10 min which is gold difference and experience difference, and finally 16 features of what the each team has accomplished (2x16 columns): # of wards placed, # of wards destroyed, first blood, # of kills, # of deaths, # of assists, # of elite monsters slain, # of dragons slain, # of heralds slain, # of towers destroyed, total gold, average level of the team, total experience, total minions killed, total jungle minions killed, creep score (CS) per min, gold per min.

The dependent variable is blue team win/loss.

Below are the distributions for each feature.

As shown in the distribution, the blue side has won 50% of the games, and looking at the gold/exp differences, the normal distribution centered around zero is to be expected as the team with experience/gold advantages are stronger and more likely to win (logically speaking). Normal distributions are also present in most other features, meaning those features should have an impact on the winning chances of a team.

However, there are some features that do not look like they are relevant to winning a game when prioritized in the first 10 minutes. For example, the # of towers destroyed is at 0 in over 8000 of 9879 games, so this feature is likely not useful objective to prioritize in the first 10 games to win the game.

Other variables that will be excluded as inputs are total gold, total experience, and # of elite monsters killed by both teams. The reason why the gold and exp columns are being excluded is that every game has a different pace to it so total experience/gold obtained within the first 10 minutes varies and so this is not a good indicator of predicting a win. The more important part about gold and exp is to have more than the other team, and these features are already included: gold difference and experience difference. After all, the goal is just to be stronger than the other team. As for # of elite monsters killed, this feature is actually a superset of dragons and heralds. Since the research question is to see what objectives to prioritize, there is not much value in studying a superset, especially since dragons and heralds are already going to be inputs to the model. Additionally, CSPerMin for both teams will be ignored because we are already using minionKills (also known as “creep score” or CS) which is essentially CSPerMin*10. Since we only care about the statistics 10 minutes in, CSPerMin is a redundant feature.

The dataset will be split into a training set and a test set. The training set will be used to train the model and the test set will be used to evaluate how well the model performs. 1000 games will be used for testing and the rest of the games will be used for training. We want to as much data as possible for training because the model will be able to learn more and better capture the trend in data to avoid overfitting into a specific patterns of data that might not be the case for the general situation. However, we would still like a fair amount of data for testing to get the a meaningful prediction accuracy.

Model

The research question is to find out which parts of the game to prioritize. One way tackle this problem is to individually analyze each feature of the game and compare them with other features. However, due to the large number of features and how the game is based on snowballing advantages (eg. high KDA leads to securing dragons/heralds), it is difficult to determine whether an individual aspect of the game is the significant to victory, or it is just a by-product of the other advantages the team has gained.

So the approach that is used is to consider all the features together, and compare them based on the weights assigned to determine the importance of each feature. To do this, we will use a binary logistic regression model to predict whether a team wins based on what the team has achieved throughout the first 10 minutes of the game.

One data pre-processing that should be done is to consolidate a team's kills/deaths/assists into a value called into a KDA ratio. For this analysis, the KDA will be calculated as (kills+(assists/2))/deaths. The reason why assists is halved is because the gold obtained from assists is half of the kill gold and for the first ten minutes of the game, there is on average 1 assist per kill. So assuming there is one assist per kill, then for every two assists, a kill's worth of gold should be awarded. This KDA calculation is more useful than individual values of kills, deaths, assists because it gives a better picture of the advantages/disadvantages of previous fights and skirmishes in the game. For example, knowing that a team has 4 kills is not useful enough to gauge if it has an advantage over the other team because we do not know how many deaths that team had given to the other team. If the team has no deaths, then the calculation is calculated as (kills+(assists/2)), since we still want to know how many kills and assists the team is ahead by.

The output of the model is whether blue team wins or loses and the inputs of the model are the following features: blueKda, blueGoldDiff, blueExperienceDiff, blueFirstBlood, blueTowersDestroyed, blueTotalMinionsKilled, blueTotalJungleMinionsKilled, blueWardsPlaced, blueHeralds, blueDragons.

The model is constructed using the glm function from R.

Results

Running the model predictions against the test set, the accuracy of the predictions is 74.3%. This is significantly better than random guessing and there are definitely trends that can be deduced from the model.

Comparing the weights assigned to each feature that were used as inputs of the model, some features play a larger role in the prediction than others. The descending order of positive weights are as follows: first blood > heralds > jungle minions > dragons > KDA > gold difference > experience difference. The features that have little impact to snowball the game in the first 10 minutes are Wards and minions killed. In particular, "Towers destroyed" was assigned a large negative weight which hints that taking towers early actually hurts your chances of winning the game.

It is also important to mention that dragons, gold difference, exp difference, # of jungle minions killed are the features that are statistically significant (p-value < 0.05) which means these features have strong evidence to reject the null hypothesis, making them to be convincing factors in determining the dependent variable.

In these next few subsections, the most notable features will be further analyzed either individually or into smaller groups if there is an interesting connection between them.

First blood

First blood is a term that refers to the first kill of the game. It awards the player with a little extra gold on top of the regular kill gold. First blood has the largest positive weight out of all the other features which is interesting because players do not often make a big deal about first blood bonus gold. Below, we can see blue team wins with first blood 60% of the time, which is an impressive winrate for only taking the first of many kills the game.

```
##   win with first blood
## 1            0.6058824
```

Heralds and Towers

Herald is an elite monster that spawns in the middle of the map. It takes about a minute to kill in the early game so which is quite a long time. The reward of defeating Herald is a single-use ability to deal large damage to towers. However, the model summary shows that destroying towers this early in the game is does not set up a path to victory, so is this a contradiction? This question will be addressed on a player's perspective in the discussion section; however, it is interesting to note that the winrate of blue team when it defeats the herald, but does not take first tower in the first 10 minutes is very high at 81.9%.

```
##   herald no tower
## 1       0.8190045
```

KDA, Gold Difference, Experience Difference

The key to winning the game is to get a lead over the opponent. So KDA, gold difference, and experience difference are all features that compare the two teams unlike the other features. Hence it is makes sense for gold difference and experience difference to have a very high significance value. However, the weight estimates for these three features are not among the top 3. This matter will also further discussed from the player's perspective in the discussion section, but in terms of the stats themselves, this is could be the negative effects of multicollinearity. Multicollinearity occurs when independent variables are too correlated with one another. Although this does not affect the predictive ability of the model, it is reduces the inference ability of the estimates themselves. In other words, the estimates are not as trustworthy and so for our case, if these three features are correlated, then there is less certainty in making an inference based on their individual lower weights.

This can be verified by viewing the variance inflation factor and the correlation matrix (for goldDiff, expDiff, KDA only):

```
##                         blueKda                    blueGoldDiff
##                        1.933677                        4.796266
##              blueExperienceDiff                  blueFirstBlood
##                        3.436018                        1.180174
##              blueTowersDestroyed          blueTotalMinionsKilled
##                        1.134860                        1.175483
## blueTotalJungleMinionsKilled                 blueWardsPlaced
##                        1.116838                        1.005264
##                     blueHeralds                     blueDragons
##                        1.070894                        1.035692
```

```
##                      blueKda blueGoldDiff blueExperienceDiff
## blueKda            1.0000000    0.6890478          0.6070490
## blueGoldDiff       0.6890478    1.0000000          0.8947295
## blueExperienceDiff 0.6070490    0.8947295          1.0000000
```

The variance inflation values the all the features used in the model are under 5, which shows that all the variables are not too correlated with one another. However, with blueGoldDiff at 4.796 and blueExperienceDiff at 3.43, these higher VIF values show that there is indeed some correlation involved with the rest of the variables. The confusion matrix also agrees that there exists a fair amount of correlation among the three variables especially goldDiff and expDiff (0.8947). The higher the value, the more correlation exists between the variable belonging to that column of the matrix and the variable on that row of the matrix (min value of 0 which means not correlated at all, max value of 1 which means correlated to the point where they are redundant and multicollinearity is an issue). As such, since we know there is some collinearity, this explains why goldDiff and expDiff have lower weights. So these weights should not be taking too seriously because these features are logically-speaking the most important in winning the game.

Discussion

This section is a work in progress.