League of Legends - Which Resource In The Summoner’s Rift Will Most Likely To Carry You to The Victory?

We build a logistic regression model to see what is the most significant resource in a game that contribute the most to win a game

Dingyi Yu

17 December 2020

Abstract

As one of the most popular video games in the world, League of Legends had nourished a significantly profitable industry. From the original purpose of personal relaxation to those e-sports players who pursue championship in world-wide competitions, the game propersies among people in a generation and had created more than a million job oppotinities and immeasurable value. Dividing into two groups of five, each group aims to destroy the Nexus of the other group to win the game and thus all players need to gain as much resource as possible to keep the lead position throughout the game. That leads to the key question to be researched in this study: Among all those avaliable resources, what is the most significant attribution that will lead to the final victory of a single game? To make furthur analysis, a logistics regression model is constructed to rank each of the element that attribute to the game result.

Keywords: League of Legends; Logistics Regression Model; Ranked Game; Attribution to Win
Code: Code and data are avaliable at https://github.com/eddieyyy/League-of-Legends (https://github.com/eddieyyy/League-of-Legends)

Introduction

In the past few years, the e-sport community was under an accelerated stage. The e-sport industry, which used to focus on leisure and entertainment, expect to create value of more than 1.8 billions dollar by 2022. Despite many video games thriving in the industry, few games could maintain their leading position in the community like League of Legends. The Season 9 World Championship in 2019, which was the most recent annual world-wide competition of League of Legends (LoL), had attracted more than 100 millions audiences globally.

Launched in 2009, League of Legends is a free Multi-player Online Battle Arena (MOBA) video game developed by Riot Games. The game quickly captured the interests of youth and kept thriving for ten years remaining the top popular video game world-wide. In each single game of League of Legends, ten players are divided into two groups of five. Each player controls a unique champion and the goal is to destroy a construction called “Nexus” of the other group to win a game. During the game, players need to obtain and allocate resources to maximize their uses. Those sources are available in many ways, such as killing the hostile champions from the other group, killing minions, killing dragons and barons to obtain buffs and destroying constructions such as turrets and inhibitors. To win a game, at least five turrets and one inhibitor needed to be destroyed until the team could reach to the hostile Nexus. Other targets, such as the amount of kills, money and buffs, are not indispensable but are undoubtedly beneficial indicators.

The diverse mechanism of the game leads to a new occupation, the gaming analysts. Similar to traditional sports team, the e-sport professional teams also require such analysts who could provide useful strategies that help the team to win the game in the competition. In this study, we had randomly collected data of 5000 games from different levels of players to see what is the most significant resource that contributes to the final win. By ranking all resources based on their level of significance, we could hopefully provide constructive idea for the analysts in professional teams to develop strategies in competition.

Data

Data Variables

The data used in the analysis consists of following 14 variables:

matchId is a categorical variable indicates each unique match. We will use it only for identification of certain matches, not for analysis.

tier is a categorical variable indicates the match’s ranked level which shows the level of the player participated in the match. It has 5 categories including BRONZE, SILVER, GOLD, PLATINUM and DIAMOND in an ascending order.

blue_kill is a numeric variable indicates the total number of kills for blue team.

red_kill is a numeric variable indicates the total number of kills for red team.

blue_gold is a numeric variable indicates the total number of gold earned by blue team.

red_gold is a numeric variable indicates the total number of gold earned by red team.

blue_firstBlood is a logical variable indicates whether blue team gets the first blood (get the first kill of the game), if the value is False meaning the red team gets the first blood.

blue_firstTower is a logical variable indicates whether blue team destroy the first tower of the game , if the value is False meaning the red team destroys the first tower.

blue_firstInhibitor is a logical variable indicates whether blue team destroy the first inhibitor of the game , if the value is False meaning the red team destroys the first inhibitor.

blue_firstBaron is a logical variable indicates whether blue team kills the first Baron of the game, if the value is False meaning the red team kills the first Baron.

blue_firstDragon is a logical variable indicates whether blue team kills the first dragon of the game, if the value is False meaning the red team kills the first dragon.

blue_firstRiftHerald is a logical variable indicates whether blue team kills the first Rift Herald of the game, if the value is False meaning the red team gets the first Rift Herald.

gameDuration is a numeric variable indicates the total length of the game in seconds.

result is a categorical variable shows the winner of the match.

A brief overview of the statistical information for the numerical data is shown below

```
##    blue_kill        red_kill       blue_gold         red_gold     
##  Min.   : 0.00   Min.   : 0.00   Min.   :  6773   Min.   :  7364  
##  1st Qu.:17.00   1st Qu.:16.00   1st Qu.: 38617   1st Qu.: 37626  
##  Median :25.00   Median :25.00   Median : 49134   Median : 49226  
##  Mean   :25.49   Mean   :24.81   Mean   : 49205   Mean   : 48853  
##  3rd Qu.:34.00   3rd Qu.:33.00   3rd Qu.: 59969   3rd Qu.: 59586  
##  Max.   :69.00   Max.   :69.00   Max.   :110544   Max.   :110531  
##   gameDuration 
##  Min.   : 479  
##  1st Qu.:1289  
##  Median :1595  
##  Mean   :1602  
##  3rd Qu.:1894  
##  Max.   :3641  
```

Data features and strengths

The data is collected through the bronze to diamond ranked-game record from 2020-08-07 to 2020-09-07 in Korean server which is retrieved from RIOT API - https://developer.riotgames.com/ (https://developer.riotgames.com/).

The time of the data is relatively closed to today, which is a remarkably strength of the data. Since the version of the game updates frequently and any significant update may change the way of winning the game.

The data is recorded from ranked-game, which is more relevant to common players comparing to analyze data from professional competitions. Moreover, ranked-game data also has an advantage to observe a more generalized trend of winning a game comparing to data from professional competitions as professional teams tend to play in designed ways for certain strategies.

The data also records players of a wide range from bronze to diamond, which makes the result representative to most players. Since players at different levels tend to perform differently in a game, it is better to include data from different levels to make sure that the characteristics of most players can be captured by the model.

Data weaknesses

As mentioned above, the data is from the Korean server only, which makes it hard to be generalized to players from other regions. It had been shown that players in different regions tend to have different playing style, which could affect their competition to certain resources such as Baron and dragons. Besides, the killing number and the duration of a game could also be affected by game style as game in

Korean server do not favor team-fighting but focus on destroying constructions such as towers and inhibitors.

Furthermore, due to certain restrictions, the size of the sampled data is relatively small, as we need to perform model validation for training and testing our model. In order to effectively utilize the limited data, the cross-validation will be performed while building the model which potentially increase the cost by doing calculation.

Data Visulization

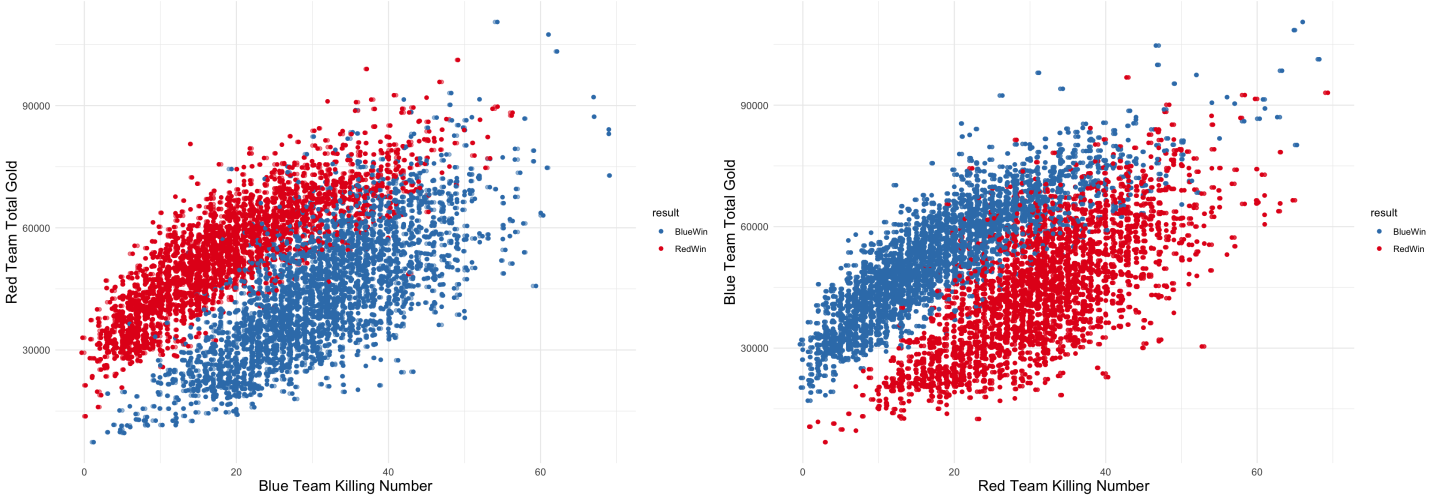

Figure 1: Winning team tend to earn more gold and get more killings compared to the losing game..

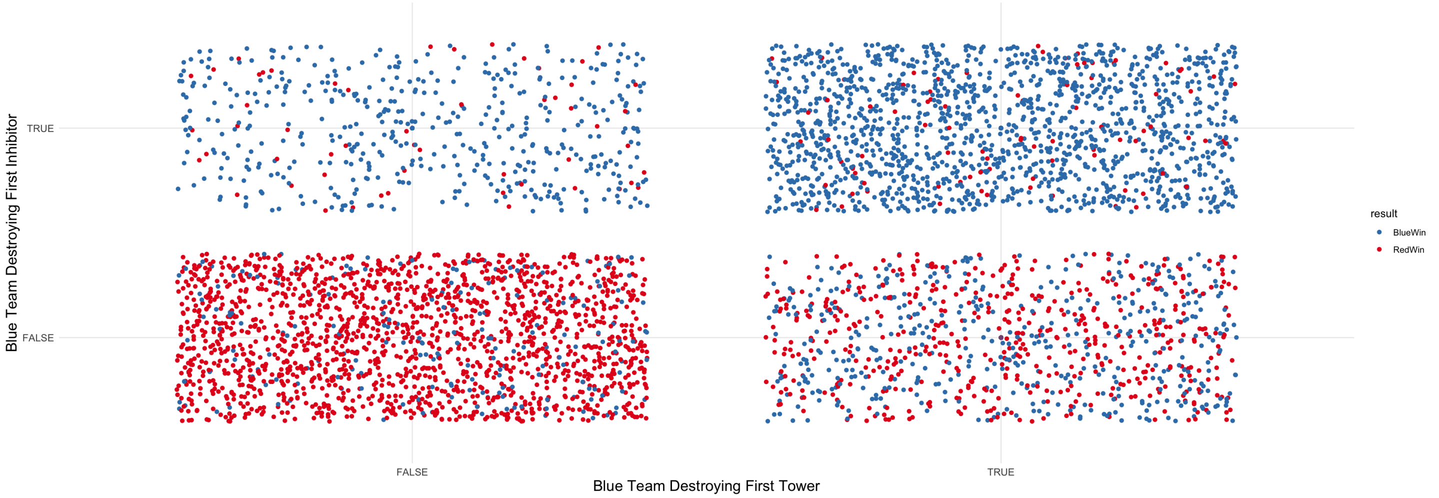

Figure 2: Destroying first inhibitor is a remarkable indicator for winning a game, while first tower occupation is not that significant.

By observing Figure 2, which combines the first tower and the first inhibitor occupation together as they are both structural resources of the game, we could see that first inhibitor is a remarkable indicator for winning a game, especially when the team also destroys the first tower of the game (Figures in top-right and bottom-left corners). As of the team which does not destroy the first inhibitor but destroys the first tower, one interesting phenomenon is that the it is harder for red teams to win in this case comparing to blue teams (Figures in top-left and bottom-right corners).

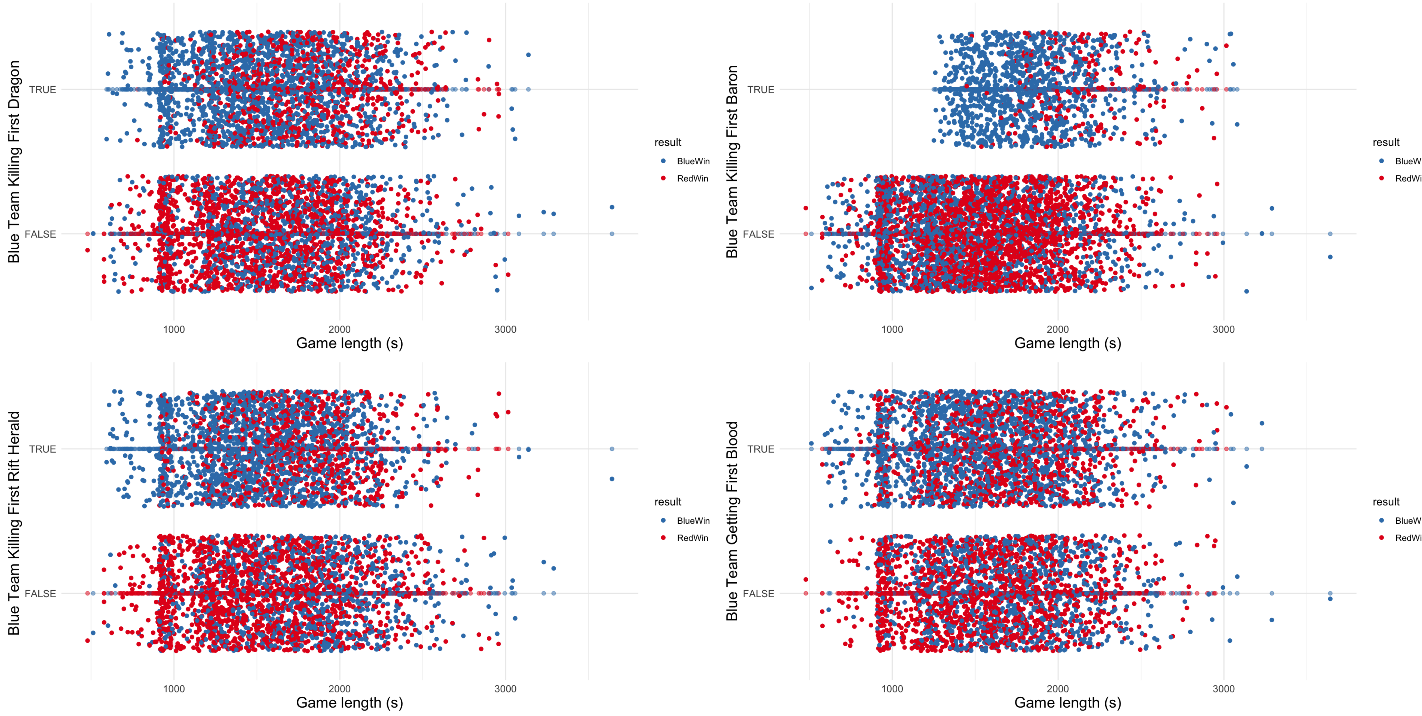

Figure 3: First Baron is a good indicator for winning a game, whereas first dragon, first Rift Herald and first blood do not show significant difference for the final outcome.

By observing Figure 3, which shows analysis to the trend that how occupation of first dragon, Baron, Rift Herald and first blood might attribute to the final outcome of the game. Obviously, all those four factors contribute to increase the probability of the final victory. However, among those factors, we could observe that killing the first Baron contributes the most to the final victory, as overturns seldom, if ever, happen when a team kills the first Baron, as it will maximize the superiority by providing the team with enormous power to destroy structures such as inhibitors, which is more essential to the final win.

One thing to notice is that as Baron appears in the game only after 20 minutes, therefore the points in Baron graph indicating Blue team kills the first Baron occurs only after approximately 1200 seconds. For game ends before 20 minutes, Baron is an invalid indicator for making predictions.

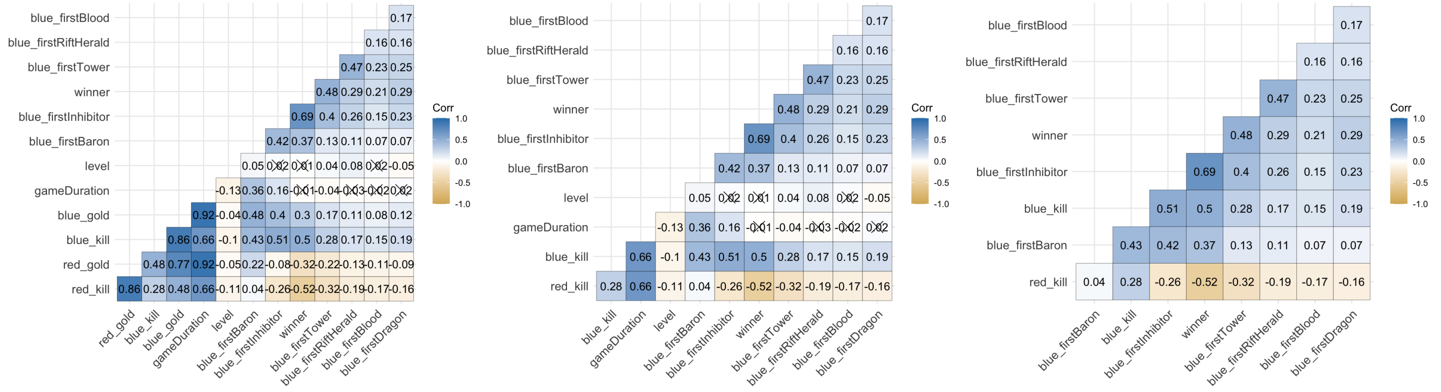

Figure 4: Correlation matrices of all variables

Figure 4 shows the process of filtering variables using correlation matrices. According to the first matrix, where we include all variables (except for the matchId because it makes no sense on predicting the result), we could see that certain variables are high-related, for example, gold versus gameDuration and gold versus kills, which are respectively high as 0.92 and 0.86. Such correlation is reasonable as the amount of gold earned will definitely increase as the game lasts longer or players take more killings (each killing is worth 300 gold in general). Such high correlation between variables might cause the issue of multi-collinearity, which might make it difficult to measure individual influence of certain predictors on the response variable and make the fitted model unstable.

To reduce the the potential multi-collinearity issue, we further remove the gold variable as it is commonly known that the winner side always end with more gold as it is the essential measurement of taking the lead. The second correlation matrix shows the result without the gold, and we would like to focus on the correlation between winner(our response) and other variables. Based on the value presented by the matrix, we could see that level(which is originally tier in raw data) and gameDuration had rare almost no correlation explaining the final outcome of a game, so we further eliminate the two variables.

The third correlation matrix is ideally constructed, as no correlations between any explanatory variables could exceed 0.5, which helps us avoid the potential multi-collinearity issue. Moreover, we could see that inhibitor and kill are most correlated to final outcome of the game as they have higher correlation to winner comparing to other variables.

Model

As the final outcome of winning a game is a binary result (1 stands for blue team winning the game and 0 stands for red team winning the game), logit model is commonly used to predict such information. We start by building a full model including all variables (except for the matchId because it makes no sense on predicting the result) and see how the model performs.

By dividing the data into 85% for model-training and 15% for model-testing, we use model validation by seeing the accuracy in the form of confusion matrix to evaluate the model performance.

```
##                               Estimate   Std. Error     z value     Pr(>|z|)
## (Intercept)              -0.6515280726 9.094557e-01  -0.7163935 4.737484e-01
## tierDIAMOND               0.8721562531 5.930485e-01   1.4706324 1.413906e-01
## tierGOLD                  0.7513116081 5.103471e-01   1.4721580 1.409782e-01
## tierPLATINUM              0.4775365055 5.434098e-01   0.8787779 3.795217e-01
## tierSILVER                0.1728902512 4.625015e-01   0.3738156 7.085415e-01
## blue_kill                 0.0057180748 3.173605e-02   0.1801760 8.570144e-01
## red_kill                 -0.0145935482 3.126149e-02  -0.4668218 6.406273e-01
## blue_gold                 0.0009096542 7.965690e-05  11.4196543 3.335582e-30
## red_gold                 -0.0009977198 8.333017e-05 -11.9730920 4.916178e-33
## blue_firstBloodTRUE      -0.4373178211 3.116488e-01  -1.4032392 1.605455e-01
## blue_firstTowerTRUE      -1.4715534251 3.542793e-01  -4.1536530 3.272093e-05
## blue_firstInhibitorTRUE   2.0516283820 3.495479e-01   5.8693769 4.374359e-09
## blue_firstBaronTRUE      -1.2254278455 3.472499e-01  -3.5289513 4.172098e-04
## blue_firstDragonTRUE      0.3239373349 3.060958e-01   1.0582875 2.899244e-01
## blue_firstRiftHeraldTRUE -0.2517616527 3.245681e-01  -0.7756821 4.379367e-01
## gameDuration              0.0034133613 2.041088e-03   1.6723248 9.446030e-02
```

The full model, as shown above, could be expressed in the following way:

$$Pr(result) \in \{1, 0\} =$$

$$logit^{-1}(\alpha_0 + \beta_{1[i]}(tier_i) + \beta_2(blue\_firstBlood) + \beta_3(blue\_firstTower) + \beta_4(blue\_firstInhibitor) + \beta_5(blue\_firstDragon)$$
$$+\beta_6(blue\_firstBaron) + \beta_7(blue\_firstRiftHerald) + \beta_8(gameDuration) + \beta_9(blue\_gold) + \beta_{10}(blue\_kill)$$
$$+\beta_{11}(red\_gold) + \beta_{12}(red\_kill) + \epsilon_i)$$

$$\epsilon_i \sim N(0, \sigma_e^2)$$

Using the usual 5% threshold significant level which is indicated by at least one * shown after the p-value of certain variables, we mark those variables as significant. Significant variables include gold, tower, inhibitor and Baron. The gold's correlation to the final outcome is clear as demonstrated above, and we could see destroying structures is more essential compared to resources from killing and other jungle resources. The Baron appears to be significant could also be explained by its powerful buff for destroying structures.

However, certain coefficients of the predictors in the model tend to be anti-intuitive. For example, teams with the first Baron and first tower are predicted be less possible to win with negative coefficients, which is very likely caused by multi-collinearity issue. To further test the performance of the model, we use confusion matrix to see the accuracy.

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   0   1
##          0 321   8
##          1   7 404
##                                           
##                Accuracy : 0.9797          
##                  95% CI : (0.9668, 0.9886)
##     No Information Rate : 0.5568          
##     P-Value [Acc > NIR] : <2e-16          
##                                           
##                   Kappa : 0.9589          
##                                           
##  Mcnemar's Test P-Value : 1               
##                                           
##             Sensitivity : 0.9787          
##             Specificity : 0.9806          
##          Pos Pred Value : 0.9757          
##          Neg Pred Value : 0.9830          
##              Prevalence : 0.4432          
##          Detection Rate : 0.4338          
##    Detection Prevalence : 0.4446          
##       Balanced Accuracy : 0.9796          
##                                           
##        'Positive' Class : 0               
## 
```

The confusion matrix shows that the accuracy of the prediction using the model is approximately 98%, which is extremely high. Of the 740 testing data, only 15 predictions get wrong. Despite the high accuracy, more improvement could be done through eliminating the multi-collinearity issue for a more reasonable predictors' interpretation of the model. We then build a reduced model by removing gold, gameDuration and tier variables.

```
## 
## Call:
## glm(formula = factor(result) ~ ., family = "binomial", data = train_data6)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -3.2108  -0.1024   0.0099   0.1010   3.2676  
## 
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)    
## (Intercept)              -0.53402    0.27687  -1.929 0.053755 .  
## blue_kill                 0.32654    0.01534  21.292  < 2e-16 ***
## red_kill                 -0.35219    0.01496 -23.548  < 2e-16 ***
## blue_firstBloodTRUE      -0.29032    0.17801  -1.631 0.102917    
## blue_firstTowerTRUE       0.40402    0.19445   2.078 0.037732 *  
## blue_firstInhibitorTRUE   2.30315    0.20195  11.404  < 2e-16 ***
## blue_firstBaronTRUE       0.75548    0.20119   3.755 0.000173 ***
## blue_firstDragonTRUE      0.26253    0.17580   1.493 0.135353    
## blue_firstRiftHeraldTRUE  0.07062    0.18762   0.376 0.706607    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 5799.93  on 4190  degrees of freedom
## Residual deviance:  964.88  on 4182  degrees of freedom
## AIC: 982.88
## 
## Number of Fisher Scoring iterations: 8
```

The reduced model, as shown above, could be expressed in the following way:

$Pr(result) \in \{1, 0\} =$

$$logit^{-1}(\alpha_0 + +\beta_1(blue\_firstBlood) + \beta_2(blue\_firstTower) + \beta_3(blue\_firstInhibitor) + \beta_4(blue\_firstDragon) + \beta_5(blue\_firstBaron) + \beta_6(blue\_firstRiftHerald) + \beta_7(blue\_kill) + \beta_8(red\_kill) + \epsilon_i)$$

$\epsilon_i \sim N(0, \sigma_e^2)$

Using the usual 5% threshold significant level which is indicated by at least one * shown after the p-value of certain variables, we mark those variables as significant. Significant variables include kills, tower, inhibitor and Baron. Similar explanation could be made as above, except for the fact that significance of gold variable transfers to the significance of the kills variable. A note-worthy change is that the significance of the first tower decreases in the reduced model.

The most remarkable improvement of the reduced model is to revise those unreasonable coefficients of the model, as one could see that acquiring first Baron and tower becomes positively related to the probability of final victory. Among the significant variables, we could see that the destroying the first inhibitor contribute the most to the final victory with an approximate weight of 2.30.

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 323  12
##          1  18 387
##
##                Accuracy : 0.9595
##                  95% CI : (0.9426, 0.9725)
##     No Information Rate : 0.5392
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9183
##
##  Mcnemar's Test P-Value : 0.3613
##
##             Sensitivity : 0.9472
##             Specificity : 0.9699
##          Pos Pred Value : 0.9642
##          Neg Pred Value : 0.9556
##              Prevalence : 0.4608
##          Detection Rate : 0.4365
##    Detection Prevalence : 0.4527
##       Balanced Accuracy : 0.9586
##
##        'Positive' Class : 0
##
```

Surprisingly, the confusion matrix shows an accuracy of approximately 96% for the reduced model, which is a bit lower than the full model. Of the 740 testing data, 30 predictions get wrong. However, since 96% accuracy is still much higher than expected, we could change our training and sampling data for fitting the model to check the performance of the reduced model, which will be left for future work.

For now, I would take the reduced model for further analysis, as it presents a more reasonable coefficient and even if lower but still high accuracy

Results

As the model comparison and interpretation shown above, the reduced model is selected for analyzing the final outcome of the game. The model includes variables of killing, first tower, first inhibitor, first blood, first dragon, first Baron and first Rift Herald as predictors.

Among the predictors, we could see that the most significant indicator that could be used to forecast the final victory is destroying the first inhibitor. It is reasonable as destroying the first inhibitor would cause the spawning of superior minions, which will cause much trouble to the other team. Moreover, having the first Baron is also a good sign for winning a game, as it will indirectly increase the probability of destroying more structures including inhibitors by providing allied minions powerfully buff while attacking structures. As of killing, despite the fact that killing more enemies also contribute to the final victory of the game, it is not wise to spend much time on fighting as destroying structures is more effective for winning a game.

In conclusion, according to the model, the team must play smartly and utilize the resources as efficiently as possible to destroy structures and pay extra attention on the first Baron to win a game. When it comes to make decisions on resource exchange with the enermy team, taking or defending structures should be in a top-priority.

Discussion

The overall outcome

Understanding Result

Small World vs Large World

Weaknesses and next steps

Next Steps

References