League of Legends Win Rate Logistic Regression Analysis

Johnny Weng

Word Count: 1470

League of Legends Win Rate Logistic Regression Analysis

In this following study based on the vastly popular E-Sports game League of Legends, we will be using Logistic Regression to predict the win percentage of teams based on a number of in game factors and occurrences. League of Legends is a MOBA (Multiplayer Online Battle Arena) style game played between two teams of five people each, battling each other to achieve objectives in order to destroy the opposition's base and win the game. With the game being played professionally all around the world, we can use our statistical data to analyze which in-game course of action players should pursue in order to improve their chances of winning. Our model will use several objectives within the game to see which ones are significant to achieve and which will influence winning percentage the most. Through the use of this regression model, team managers and players can redefine in game goals and optimize their playing strategies to maximize their chances of winning their matches. The dataset takes objectives that are easily defined and quantified to help convert information into match prediction. By winning more matches in their domestic leagues, teams have the opportunity to qualify for World Championships and potentially play for glory and reputation on the International Stage.

The data from the regression model was gathered based on a list of Ranked games from a single season of League of Legends in the European Servers. Ranked games allow players to play competitively to improve your official ranking on an international stage. The game is played and updated on a per season basis, as we look at just a sample of 500 games from a single season based in the West European region. Ranked games allows players to compete and test your skills in order to help them advance up the rankings on the League of Legends ladder. Players take Ranked games very seriously because it allows anyone to prove their skills to advance their placement, potentially drawing attention to professional and college teams. In this sense, we can assume that players take these matches earnestly because there is pride in advancing your ranking. The data was tracked through in game metrics and would be extremely accurate based on artificial intelligence tracking software and lack of human error. Every data point measured happens within the game itself while following player decisions and in game scenarios.

For this particular set of data, only binary and dummy variables were used for the model, and therefore do not include a statistic summary worth mentioning. The independent variables used in our dataset include: firstBlood, firstDrag, firstTower, firstInhib, and firstBaron. Our dependent variable is a binary 1 or 0 based on whether the team wins or loses. We can, however, do a basic analysis on variables that have multicollinearity and correlates with our independent variables that affects how the data is viewed. Some variables that were included in the original dataset but not included in the model are: dragKills, towerKills, InhibKills, and baronKills. Since our independent variables directly affect these factors, there is high correlation between the two, such that a firstDrag for either team will immediately add a +1 to dragKills, which can greatly affect our model.

Table 1
Descriptive Statistics of Variables

Variable	N	Maximum	Mean	SD
towerKills	500	11	5.547	3.767
inhibKills	500	6	0.960	1.186
baronKills	500	3	0.367	0.579
dragonKills	500	5	1.321	1.196
Wins Team1	248			
Wins Team2	252			

From *Table 1,* we see that number of wins were almost identical in our data, with 248 for team 1 and 252 for team 2. Looking at these statistics, we can identify that towerKills has the highest average and baronKills the lowest. This infers that towerKills has the most occurrences since “towers” have the highest in game volume of any of the variables listed above. We can also identify that the average for inhibKills and baronKills are below 1, since they are factors that occur later in the game and may not necessarily be achieved before the game is potentially forfeited by one team. Later, we will look at the significance level of why achieving firstInhib and firstBaron will be much more significant than the other independent variables based on their frequency and importance later on in the matches.

The model created from our regression is as follows:

.512Constant - 0.801firstTower2 + 2.508firstInhib1 – 1.540firstInhib2 – 0.659firstDrag2 - 1.351firstBaron2.

Based on *Table* 6, our final regression output would involve only the significant variables listed above. In order to understand the entire model and for the regression to make sense, all the variables will be explained. Looking at *Table 2*, the variable firstBlood denotes which team achieved the first kill within the game. If our reference team (which we predetermined as team 1) gets the kill, we coded it as 1, and if the alternate team gets the kill we code it as 0. For all of our variables, we made dummy variables based on whether our reference team or the alternate team achieved the objective first. For example, if our reference team achieved firstTower1, we coded it as 1 and the variables firstTower2 and firstTower3 as 0. If the alternative team achieved it, we coded firstTower2 as 1. The third variable, firstTower3 (also firstInhib3, firstBaron3, etc.) was coded as the variable when neither team achieved the objective. None of these “3” variables appeared in our final output since they were being used as the reference category for the “1” and “2” variables. The only significant variables in this model ended up being firstTower2, firstInhib1, firstInhib2, firstDrag2, and firstBaron2. The firstInhib and firstBaron variable occurs towards the end game, and acts as an important turning point within the game, which explains their higher significance levels.

Table 2

Coefficients for Independent Variables

Variable	B	S.E.	Wald	df	Sig.	Exp(B)
firstBlood	0.046	0.305	0.022	1	0.881	1.047
firstTower1	0.065	1.064	0.004	1	0.951	1.067
firstTower2	-0.686	1.076	0.406	1	0.524	0.504
firstInhib1	2.362	0.458	26.653	1	0.000	10.613
firstInhib2	-1.644	0.479	11.790	1	0.001	0.193
firstDrag1	-0.727	0.880	0.683	1	0.409	0.483
firstDrag2	-1.273	0.861	2.184	1	0.139	0.280
firstBaron1	0.713	0.395	3.261	1	0.071	2.040
firstBaron2	-0.978	0.404	5.861	1	0.015	0.376
Constant	0.816	0.722	1.276	1	0.259	2.260

Table3

Coefficients for Independent Variables excluding firstTower1

Variable	B	S.E.	Wald	df	Sig.	Exp(B)
firstBlood	0.045	0.305	0.022	1	0.882	1.046
firstTower2	-0.748	0.322	5.411	1	0.020	0.473
firstInhib1	2.364	0.457	26.785	1	0.000	10.630
firstInhib2	-1.642	0.478	11.801	1	0.001	0.194
firstDrag1	-0.692	0.656	1.112	1	0.292	0.501
firstDrag2	-1.238	0.638	3.762	1	0.052	0.290
firstBaron1	0.713	0.395	3.259	1	0.071	2.039
firstBaron2	-0.978	0.404	5.875	1	0.015	0.376
Constant	0.843	0.564	2.233	1	0.135	2.324

Table 4

Coefficients for Independent Variables excluding firstBlood

Variable	B	S.E.	Wald	df	Sig.	Exp(B)
firstTower2	-0.758	0.316	5.759	1	0.016	0.469
firstInhib1	2.364	0.457	26.798	1	0.000	10.637
firstInhib2	-1.643	0.478	11.810	1	0.001	0.193
firstDrag1	-0.692	0.656	1.112	1	0.292	0.501
firstDrag2	-1.241	0.638	3.786	1	0.052	0.289
firstBaron1	0.712	0.395	3.257	1	0.071	2.039
firstBaron2	-0.980	0.404	5.891	1	0.015	0.375
Constant	0.874	0.526	2.757	1	0.097	2.396

Table 5

Coefficients for Independent Variables excluding firstDrag1

Variable	B	S.E.	Wald	df	Sig.	Exp(B)
firstTower2	-0.774	0.314	6.072	1	0.014	0.461
firstInhib1	2.177	0.418	27.163	1	0.000	8.818
firstInhib2	-1.829	0.442	17.116	1	0.000	0.161
firstDrag2	-0.658	0.309	4.521	1	0.033	0.518
firstBaron1	0.672	0.394	2.908	1	0.088	1.957
firstBaron2	-1.012	0.403	6.315	1	0.012	0.363
Constant	0.460	0.331	1.928	1	0.165	1.584

Table 6

Coefficients for Independent Variables excluding firstBaron1

Variable	B	S.E.	Wald	df	Sig.	Exp(B)	VIF
firstTower	-0.801	0.311	6.635	1	0.010	0.449	1.328
firstInhib1	2.508	0.382	43.176	1	0.000	12.282	2.940
firstInhib2	-1.540	0.403	14.617	1	0.000	0.214	3.314
firstDrag2	-0.659	0.307	4.599	1	0.032	0.517	1.213
firstBaron2	-1.351	0.356	14.402	1	0.000	0.259	1.265
Constant	0.512	0.332	2.382	1	0.123	1.669	

For this particular regression, a significance level of .05 was used with a Confidence Interval of 95%. Starting from *Table 2* in our model, we can see that several of the independent variables were insignificant. Using backwards elimination, every variable that wasn't significant was taken out one at a time until our final regression in *Table 6* only includes significant variables at a .05 level. After running a multicollinearity test in SPSS, we were able to find our VIFS. As we can see in *Table* 6, our independent variable firstInhib2 has a VIF of 3.314, which states that it may have some level of multicollinearity within our model. Since we do have a reference variable of firstInhib3 where neither team achieves firstInhib, we are able to run both firstInhib1 and firstInhib2 as part of our regression. Looking deeper into our coefficients from *Table* 2, we can see that achieving every objective first will improve your logit and probability of winning with the exception of firstDrag1. This makes a lot of sense within the game itself as firstDrag is oftentimes the primary objective for teams and does not hold as much of a domino effect as the other factors. Through these basic calculations, we can conclude that firstInhib would play the biggest factor on predicting win percentage, regardless of the other objectives being achieved within the game.

Table 7

Win Percentage from achieving each Objective

Variable	Coefficient	firstTower2	firstInhib1	firstInhib2	firstDrag2	firstBaron2	All Objectives	fT2 & fD2	*fD2 & fB2*
firstTower2	-0.801	1	0	0	0	0	1	1	0
firstInhib1	2.508	0	1	0	0	0	0	0	0
firstInhib2	-1.540	0	0	1	0	0	1	0	0
firstDrag2	-0.659	0	0	0	1	0	1	1	1
firstBaron2	-1.351	0	0	0	0	1	1	0	1
Constant	0.512	-	-	-	-	-	-	-	-
	Logit	-0.289	3.02	-1.028	-0.147	-0.839	-3.839	-0.948	-1.498
	Odds	0.749	20.491	0.358	0.863	0.432	0.022	0.388	0.224
	Probability	0.4282	0.9535	0.2635	0.4633	0.3017	0.0211	0.2793	0.050

From *Table 7* we can make some predictions on win percentage based on the objectives that were achieved within the game. Since most of our final regression coefficients involve team 2 achieving the objective first, the probabilities which are in the output determine the win percentage for team 1. Looking at our variable of firstTower2, this indicates the alternate team achieving the objective first, which we can then use to determine our win percentage for team 1 falling to ~42.82% . When team 1 achieved the objective of firstInhib1, we can see that their win percentage jumped to ~95.35%. If team 2 achieved all of the objectives within the game first, then team 1's win percentage is ~2.11%.

Using the information from this dataset, we could with relative accuracy predict win outcomes based on a group of different factors that occur in game. Although the information provided was accurate and interesting, there could have been an added layer of complexity that dives deeper into other statistics within the game. For future collection of data, we could continue to measure ranked competitive matches within the game itself but with more data tracking. For example, the map itself is divided into top and bottom, with symmetrical dimensions but slightly different objectives on either side of the maps. By tracking which team played on which sides, we can see whether or not there are advantages to being on either side, which helps to measure the balance of the game. If we could track number of kills, deaths, and assists for each player position in game, we could correlate win percentage to the stats that matter most for each distinct position. More statistical measurements and data can always be added to our datasets so that we can add a level of accuracy to pinpoint the factors that most significantly influence game outcomes.