League of legends important feature extraction and win prediction using various classification models

Josip Kopršina

December 2020

1 Abstract

While playing League of Legends most players believe that matches are won and lost in the first 10 or so minutes of the game. Using data models such as logistic regression and decision trees we show how even though this is often the case it is not common enough that players should lose morale when they start losing early, as the chances of turning things around are not so low. Furthermore, using the same methods we show that contrary to most players opinion, the focus in the match should not be on killing other players of the opposite team but on all aspects of the game that give experience and gold as they all influence winning the match approximately the same amount.

2 Introduction

League of legends is an online multi player video game that was released by Riot Games in 2009 [1]. Through the years it has grown to one of the most played games ever, with a very consistent player base. To play, you pick a character ("champion") which you play as for the reminder of the match. You are then put into a match with nine other players to make two teams of five (called blue and red team) which are battling against each other. Each team has their side of the map and a base. The goal of the game is to destroy the enemy base while protecting your own. While the game goes on, your characters get stronger as they gain experience and gold. This can be achieved by killing the opposing teams characters, killing "minions and monsters" (non playable characters that are automatically placed on the map by the game) and destroying enemy structures. All of these and more will be features that we will look at. Our goal is to find out which features influence the game the most so that players know what they should focus on to maximize their chances of winning. Another thing we will be focusing on, is how difficult is it to predict the winner from the first 10 minutes of playing. A lot of players often complain that even though the matches usually last 30 to 40 minutes, the winner is mostly decided in the first 10 minutes. A list of word explanations has been added to the Appendix A.

RQ1: Is the winner decided in the first 10 minutes of the match?
We will take a look at how often we manage to get the prediction right for who will win just from looking at the information from the first 10 minutes of a match. I will assume that if the prediction rate using only the data from the first 10 minutes of the match is very good, then it really is true that most games are won or lost before the 10 minute mark. It's likely to end up true since we can say from experience that so called, "comebacks", (winning while you're behind in gold and experience) seem to be rare.

RQ2: What are the features that influence a team winning the game the most?
We assume that actions that give characters gold and experience influence it the most as those actions make the characters stronger and make it easier to destroy the enemy base.

The jupyter notebook used in the project can be found on my github repository[1].

3 Related work

There already exist a big amount of apps and web sites that help you plan for your match ahead of time such as mobanalytics[2] or Blitz[3]. These apps tell you statistics about your teammates and opponent just before entering a match, as well as how good certain champions are. All of this helps you plan how to play the coming match or even in some cases to leave and search for another one with better chances of victory. There are many studies and articles that focus on this aspect of calculating win possibility depending on what champion you are going to play and how good your teammates are [2, 3]. This won't be our focus. I want to emphasize the importance of playing well when the match has already started and the "cards have already been dealt". We will look at how important things like being the first one to take a certain objective or slay a certain respawnable monster are. Our goal is to discourage players from leaving and giving up after 10 minutes, claiming the match has already been lost, and promote giving it your best as we think matches would be much more enjoyable if people didn't just give up. Figure 1 shows how often people leave or just refuse to play matches in different skill categories and regions.

[1] *https://github.com/jkoprcina/LeagueOfLegendsDataMiningProject*
[2] *https://mobalytics.gg/*
[3] *https://blitz.gg/*

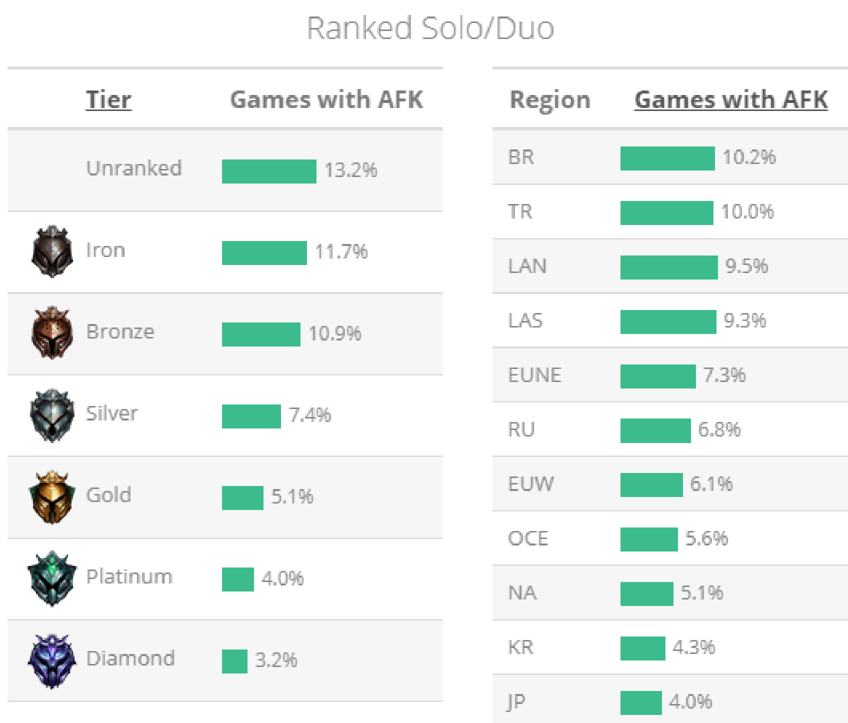

Figure 1: An image showing how often matches have people AFK in matches that affect their rank (which is why it's called Ranked at the top). AFK is a shorthand for "away from keyboard" which is a term meaning the player is not moving his character in the game. This could be because he left the match or is refusing to play. On the left side we have the information by player rank and on the right by region/server on which players can play. It is evident that better players are less likely to go AFK as the ranks are ordered from worst to best up to down same as the percentage of AFK players drops.

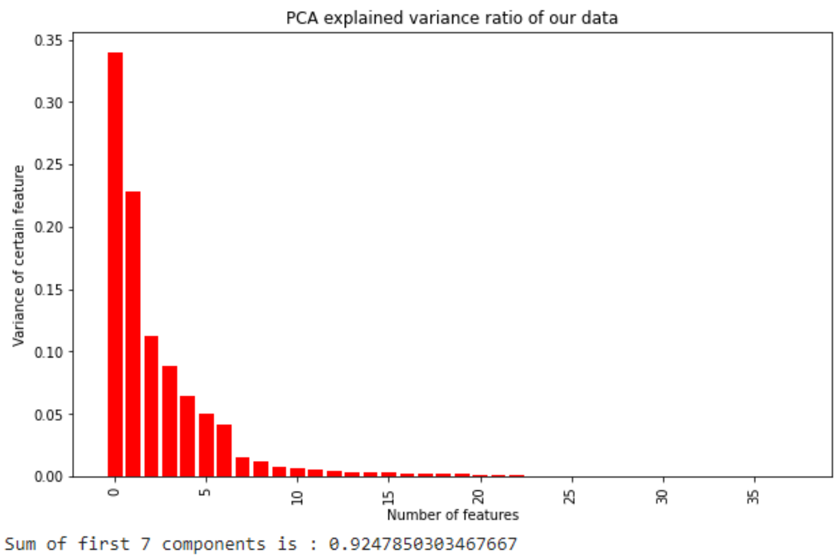

Figure 2: The first 7 components of the data account for around 92% of the variance

4 Data and research setup

The dataset we will use was taken from kaggle. It consists of 10,000 matches played in solo/duo queue diamond ranked matches. Ranked matches are those that influence your standing on the list of all players. "Solo/duo" is a subtype of Ranked games where you play with either 3 or 4 strangers on your own team (you can invite one person you know at most). "Diamond" is a stage on the list of players which only about 1% of the best players belong to. This means that all the conclusions we reach will only be valid for the players that are in the "diamond" category. All the information is taken from only the first 10 minutes of every match, along with a feature "blue_win" which shows us what team won at the end. We will use this as our y or label that we try to predict. If it's 1, blue team won, if it's 0, red team won. The rest of the features show the state of the game at the 10 minute mark. How much gold, kills, deaths, assists, exp each team has and more. The dataset was checked for missing values and it was shown it has none which made the preprocessing part much easier. The dataset was simply split into the data (X) and labels(y) and then normalized using sklearns methods for normalizing data to make it all between 0 and 1. Later, after going through most of the planned methods I realised that some features influence the match outcome a lot more than others. To maximize efficiency I applied PCA (Figure 2). This made rerunning the methods faster as they had to work with 7 features instead of 38.

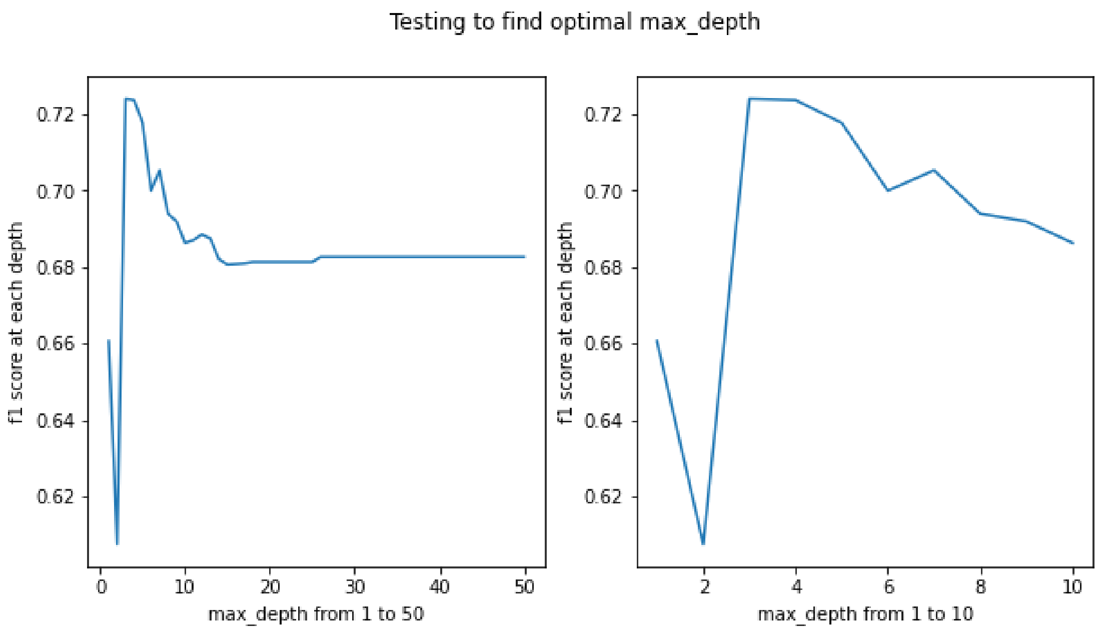

Figure 3: Searching for the optimal max_depth to use in our decision tree model. The same information is shown on the right and left side with the difference that the right side is more focused on the part that looks to be the peak point on the left.

5 Method and analysis

For our main methods of analysing RQ1 I used decision trees and logistic regression. Both were imported already finished from sklearn. The data was separated randomly into training and test set and then applied to the models. Before using the final version of the decision tree, different tree depths were tested to see which suits best (as is shown in figure 3). Each depth was tested 100 times and an average was taken. We can see from the figure that a depth of three proved to be the best. This way the decision tree model gave an average f1 score of around 74%. The logistic regression model was fitted for a few versions with different max iterations, but the basic 100 that the function sets up itself proved to be enough, probably because of the fact that the dataset had only 10,000 instances and, at this point using PCA, only 7 features (principal components). The logistic regression was run 100 times and an average f1 score of about 75% was reached. This meant that it worked about as good as the decision tree model. To see this better I plotted an ROC curve which can be seen in figure 4.

Moving on to RQ2, I again used logistic regression, but this time the .coef_ function of the sklearn model. It gave me a numeric value of how much each

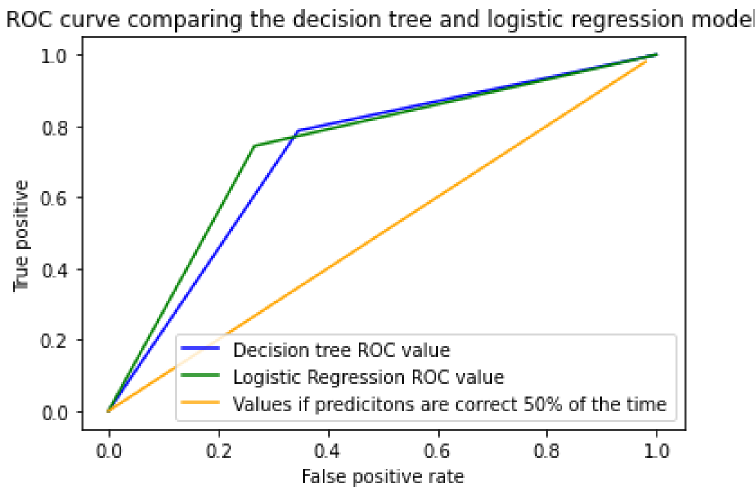

Figure 4: Comparing the quality of the decision tree and logistic regression using their ROC curves

feature influences the decision making (for this the original features were used, not the principal components). I took all of the values and split them into two categories. Those that influence blue_team winning and those that influence red_team winning positively. In figure 5 you can see a histogram of both.

6 Conclusion

If some phrases are hard to follow, a simple dictionary of league of legends related phrases is given in the appendix A. We predicted the outcome of a match with the information from the first 10 minutes. Both models have an f1 score of about 74% which is nothing amazing. I assume the calculations could be better if more data was used in the calculations. Moreover, the data was taken from matches in which highly skilled players took place, which should influence is the game lost or won quickly, as better players recover quicker from making mistakes and make less of them, so the games are more often decided in the later stages by small differences in player quality. The results could also be further improved if we used data from outside the match such as the win rate of the players and the characters they are playing as, but this was not in the scope of the project. As for RQ2, we of course found that features such as gold and experience gain were the most important factors in deciding who won. If we ignore the features that focus on having gold and experience, but focus on methods of achieving them, the number of jungle monsters killed makes the biggest difference for the blue team winning while placing wards makes the biggest difference for the red team. Blue team placing wards actually increases

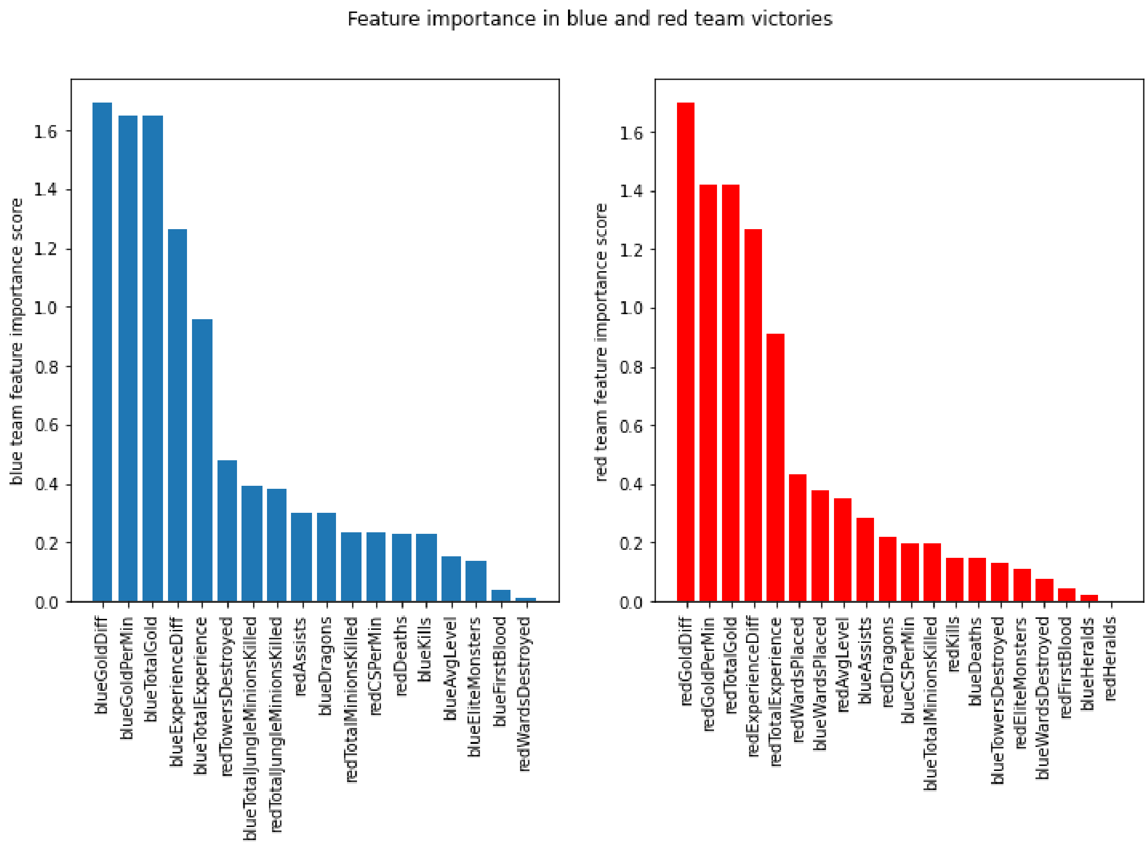

Figure 5: Looking at the importance the logistic regression model gives to the features when calculating the result. Left shows the features that favor blue team winning and right shows the features that favor red team winning.

the chance of blue team winning for some reason. This could be because of how the map is positioned so it's not completely symmetrical and could be an indication that trying to ward early is bad for the team on blue side as it probably exposes them more to being attacked by the red team. Both sides show minimal difference between the influence of killing enemy champions and gaining more cs. This is a great find as players often focus a lot more on killing enemy champions than they do on killing enemy minions. Sadly, some parts of the histogram such as blueCSPerMin and blueTotalMinionsKilled being a factor that helps red team win, make us question all the of the results. This could be a problem of the relatively small sample size. Nonetheless, the fact that the amount of kills influence the result similarly as the number of minions the team kills and the number of jungle monsters killed is a great find that proves people should focus all manners of gaining experience and gold instead of focusing just on getting as many kills as possible.

References

[1] League of Legends Basics
`https://na.leagueoflegends.com/en-us/how-to-play/`

[2] Lucas Lin, *League of Legends Match Outcome Prediction*. Stanford, 2016.

[3] Yihan Jhin, *League of Legends: Predicting wins in Chamption select with Machine Learning* https://medium.com/@jihan_yin/league-of-legends-predicting-wins-in-champion-select-with-machine-learning-6496523a7ea7. 2018.

A Appendix

Vocabulary that might make reading the work easier:
champion - a character the player can choose to play as in the match. Once chosen can't be changed for the rest of the match
experience - a collectable stat that makes your champions stronger
gold - a collectable stat that can be exchanged for items the player chooses which adds strength and utility to the champion
minions - monsters that spawn on both sides to help the team fight, they give experience and gold when killed
jungle monsters - neutral monsters that spawn on each side of the map in between the spaces where minions pass
neutral epic monsters - jungle monsters that spawn more rarely and give special bonuses to the player or team that kills it
dragon - a type of neutral epic monster
herald - a type of neutral epic monster
cs - amount of minions, jungle monsters and neutral epic monsters killed
csPerMin - amount of cs gained per minute of the match

champion - a character the player can choose to play as in the match. Once chosen can't be changed for the rest of the match
tower - a immobile object which automatically fires at enemy champions if it comes too close. Each team has their at symmetrical points
first blood - first kill of the game, gives extra experience and gold
ward - an item given every two minutes or bought with gold that gives vision of area when placed