Jonathan Yun

Ronald Daskevich

Parallel and Distributed Computing

18 August 2022

Streaming ML Analysis of League of Legends

League of Legends is currently one of the most popular games in the video game scene with 150 million registered players. League of Legends is a multiplayer online battle arena, MOBA, where 5 players fight 5 other players to destroy the other team’s central nexus. During the game there are several objectives and incentives that can lead to one team having the advantage. These objectives such as first blood, first tower or first dragon gives one team advantages such as gold or experience that could snowball one team’s lead or give another team enough to comeback. Each player is in control of a character, or champion, that will gain experience points and gold as the game progresses. The player can then use the experience points to level up their champion and use the gold to buy stronger items. The faster they can gain experience points and gold, the more advantageous their team becomes when fighting the other team. The way that most teams gain the advantage is by accomplishing the objectives listed above. The game rewards the team that can achieve first blood or destroy the first tower with more gold and as a result gives the advantage to that team. In this assignment I wanted to see if we can purely predict which team would win by only depending on which team was able to capture which objective first.

The data comes from Mitchell J on Kaggle where the data contains the following information: Game ID, game creation time, game duration, season id, winner, first baron, first dragon, first tower, first blood, first inhibitor, first right herald, the champions, summoner spells, number of towers destroyed, number of inhibitors, destroyed, number of barons killed, number of dragons killed, number of rift heralds killed and the bans from each team. From the variables above I decided to narrow the variables to only first baron, dragon, tower, blood, inhibitor, and rift herald taken as the variables I would focus on. To do this, I created a python script that would cut off the excess variable information and only return a csv file of the variables I wanted. I also wanted to have a testing csv file and a training csv file that would replicate the data files in assignment 7. So, in the script I was able to keep the winner column but replace its values with 0 values rather than the default values. After running the script, it returned two files, testingGames.csv and trainingGames.csv, that I would use to apply the machine learning tools on the dataset.

Now that I had my two files, I was able to start my machine learning on the dataset. I first created a schema for the dataset and then imported the two csv files into databricks. Once the files were loaded, I decided to do a logistic regression on the dataset. After formatting the dataset further in databricks by creating indexers and bucketizers for all the variables I was able to perform the logistic regression on the data set using a pipeline. From there I fit the pipeline to the testing dataset and gathered my results. I then decided to test the accuracy of my results by using a multiclass classification evaluator to evaluate my prediction column to my winner column in databricks. From my results I see that the accuracy of the logistic regression was low at 48.76%. I wanted to see if there would be any difference in accuracy if I were to use a decision tree classifier and the accuracy did change but only slightly to 50.13%. After applying the machine learning to the dataset, I then applied streaming to the dataset as well. From our results we see that we are only able to correctly guess which team won about 50% of the time based on which team took an objective first.