Frank Rodriguez

Michael Seman

Data Mining – Final Project

Dr. Ernesto Lee

12/16/22

League of Legends Game Outcome Predictor

League of Legends (LOL), also commonly referred to as just League, is a multiplayer online battle arena video game developed by Riot Games. The game features a map, called Summoner's Rift, with two bases on opposite ends of the map. Each base is defended by a team (Red or Blue Team) of 5 players each, and each player chooses a "champion" to play with. Each champion is unique and has a different style of play. Champions become more powerful throughout a game by collecting experience points to gain levels and gold to obtain special equipment which augments each champion's abilities and make them more powerful as the game continues. The goal of each team is to reach the opposing teams base and destroy their opponent's "nexus", which is a large structure located in the deepest parts of the base near that team's spawn point.

League of Legends is probably one of the most popular games on the planet. The 2022 World Championships were aired for a total of just over 143 hours and the average viewership per hour was about 980,000 unique users, peaking with the Grand Final match having 5.15 million concurrent viewers (Note: The viewership numbers do not consider the largest viewing market of LOL, which is China. Some estimates have China as doubling the numbers previously mentioned). With the crowds that LOL draws worldwide, the prize pool for the 2022 Worlds was $2,225,000. This amount of popularity and money involved in the competitive scene of this game is what would make people put resources into finding the most optimal ways to win matches. This is where our analysis comes in.

We found a binary classification dataset in Kaggle that had 10,000 Diamond Rank League matches. Diamond rank and above is the top 2% of all ranked players. In each row of the dataset, key statistics for each team were recorded at the 10-minute mark of the match including kill, death, gold differential, experience differential and so on. Each match in LOL can be played for much longer than 10 minutes, especially at the higher levels of competitive play (Diamond Rank and above), which has an average match length of around 26 minutes. The target variable in our dataset is whether the blue team won (1) or the red team won (0). Our goal was to try and create supervised machine learning models that could use these statistics at the 10-minute mark of matches to accurately predict the outcome of new match statistics.

The first part of any data science project is doing some Exploratory Data Analysis (EDA) and ours was no exception. We started by loading the Kaggle CSV file into a python pandas dataframe and took a look a bit deeper to see what we can glean from the data. One of the key things we saw right away, which would be important later, is how there were the same columns to record the statistics of both teams red and blue. We looked at the number of unique values to see if we could find any categorical columns, normally those would be noticed as they would have a lower number of unique values which would represent different categories. The issue with our dataset specifically, is that most of our columns that had a lower number of unique values were because the records only show the statistics at the 10-minute mark of a match. In the case of red/blue team towers destroyed, there are only 11 total towers in the game and destroying the first couple are the hardest, as that is the time when both teams are the most even. The other columns that looked categorical, but weren't, dealt with "elite" monsters. The elite monster are the "Herald" and the "Dragon". The Herald appears on the map at the 8-minute mark, while the dragon shows up every 5 minutes or so starting at exactly at the 5-minute mark. At most, the number of dragons killed in the first 10 minutes of a match would be 1, the number of Heralds 1, and the total for both or the "elite monsters" columns would be 2. The only column that is categorical, showing a low number of unique values, is blueFirstBlood. First blood is given when either the blue team or the red team get the first kill of the match.

Now getting back to the whole "there were the same columns to record the statistics of both teams red and blue" and how that can cause a problem for any ML modeling later. We noticed that there were quite a few redundant columns because of this fact. Each record had the number of kills and deaths for both teams. Although this makes sense from a record keeping perspective, when you notice that the number of blue team kills is the exact same as red team deaths and vice-versa, then it makes sense to only keep one side of it as it is a zero-sum game. We did the same for red/blue goldDiff and red/blue experienceDiff columns, each being the difference in gold/experience points for each team where one would be the inverse of the other. We removed other because they basically existed in other forms. Some of those columns were related to gold like red/blue totalGold and GoldPerMin which would just be easier to use gold differential. The other columns were related to red/blue team experience difference. Total experience points for each team, CSPerMin (Creep Score is the amount of minions, monsters and other things killed which give experience points) which is another way of calculating experience points per minute, and average level of each team, which can be easier relayed as the experience difference between the teams.

Once we chose the features we would continue with, we also saw that we did not have any missing values and that all our variables were numeric, mostly integers and a couple of floating-point values. We did see some outliers which we removed be calculating the z-scores of the values in each column and removing anything above a z-score of 4. We saw that this was almost a perfectly balanced dataset when it came to the outcome variable with only a difference of 13 between blue and red team wins. We did see how some variables had a very high correlation with blueWins some positive: blueGoldDiff, blueExperienceDiff and blueAssists (assist while blue team kills a member of the red team) and some negative: blueDeaths, redAssists, and redDragon. Gold shows as slightly more important than experience points for winning a match. The rest of the numeric variables had enough unique values that when we created histograms, they would each show a normal distribution.

Once we were able to go through the dataset, we created a split where 80% of it would train the model and the other 20% would be to test it with new values and see how well the model performed. We then ran the data through an algorithm harness that included 10 different model types and we decided we would do further exploration on the 3 best performers. The 3 best performers that we chose were XG Boost which is a type of decision tree, Linear Discriminant Analysis, and finally we chose a classic Logistic Regression. With each one we tried different methods of tuning to see if we can get a bit more out of each model. With XGBoost (XGB) we used Bayesian Optimization to try and not only speed up the tuning process, but to try and have the tuning learn from itself to find better values. In Linear Discriminant Analysis (LDA) we tuned the solver and shrinkage to try and squeeze out more performance. Using Logistic Regression (LR), we tuned some hyper parameters and additionally tried to see if any reduced dimensionality in the dataset can help LR by using Principal Component Analysis.

As we continued our quest to try and get better performance out of each model. We started to come to a similar conclusion, that most of the tuning did not do much to change our models' predictions from the original ones given by just using each model with no tuning involved. In the end, the models performed somewhat similarly, but one did stand above the rest. LDA was the best performing model with an AUC (area under the curve) of 0.7343 compared to LR that had an AUC of 0.7323 and the XGB post tuning model that had an AUC of 0.7309. All in all, we were able to predict accurately a blue win rate of 74% and a red win rate of 73%. We ended up using streamlit to create a web app and implement our LDA model's prediction using new entries for the match data that could be entered by the app user.

Overall, we were satisfied with the results we were able to obtain. Predicting the outcome of a game that may last up to an hour in length at times with an almost 74% accuracy rate, using only stats from 10 minutes into the contest is very impressive. Future research would necessitate more complete data, including champions select and individual statistics. Another avenue could be to a model that shows how the chance of a given team winning changes as time and team statistics change. In the end this was an exciting project to work on, and we are eager to share our findings and further develop our work.