DSC 478: Predicting League of Legends Games with Machine Learning

Daniel Kwan

DePaul University

Executive Summary

League of Legends is a team-based strategy game published by Riot Games where two teams of five face off to destroy the other's base.

This project attempts to discover the most important winning factors for each position, as well as predict winning games based on in-game statistics at the 10-minute mark. To do so, I have assembled data from nearly 17,000 games played at a high level of competition by interfacing with the Riot Games API.

I evaluated the following machine learning techniques to make basic predictions about winning games as well as role-specific analyses: K-Nearest Neighbors, Random Forest, AdaBoost, and Logistic Regression.

The AdaBoost and Logistic Regression models appear to indicate that early game performance and simple champion kills—both seen as the most desirable expressions of skill in the game, and often prompters of “early surrenders” are not the most important. Rather, objectives such as towers and late-game champion level are key winning factors.

DSC 478: Predicting League of Legends Games with Machine Learning

About League of Legends

On each team there are five positions that compete against their direct counterpart in different parts of the map: **Top Lane**, **Mid Lane**, **Bot Lane**, and **Jungle**. Each player selects a unique character from a roster of nearly 150 and attempts to accumulate more resources (**gold** and **experience**) in order to become stronger than their opponent and destroy their opposing base (known as a **nexus**).

Groups of computer-controlled characters known as **minions** or **creeps** march from each base and meet in the center of each lane, and players can destroy those minions to earn gold and experience. Gold can be used to upgrade the player's character with special equipment, while enough experience makes the player stronger by leveling up.

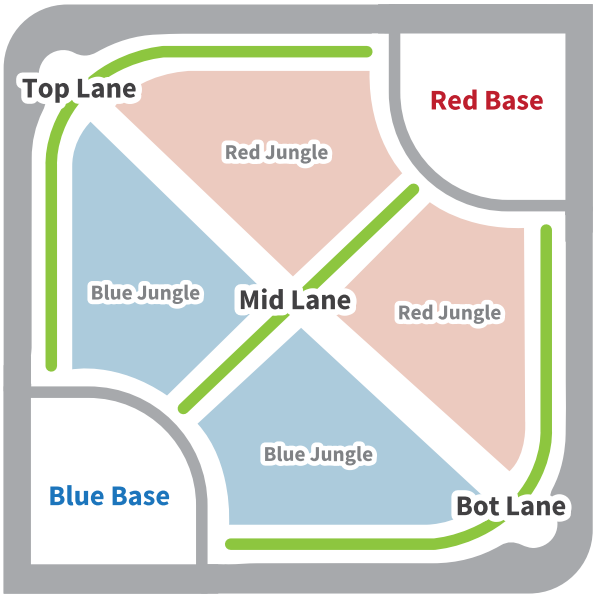


Figure 1: League of Legends Map

One player from each team will play in the mid and top lanes, while two players will face off against an opposing pair in the bot lane. The final player on each team plays in the **jungle** and is known as the **jungler**. The jungle houses static collections of neutral minions, which the jungler collects instead of staying in a lane.

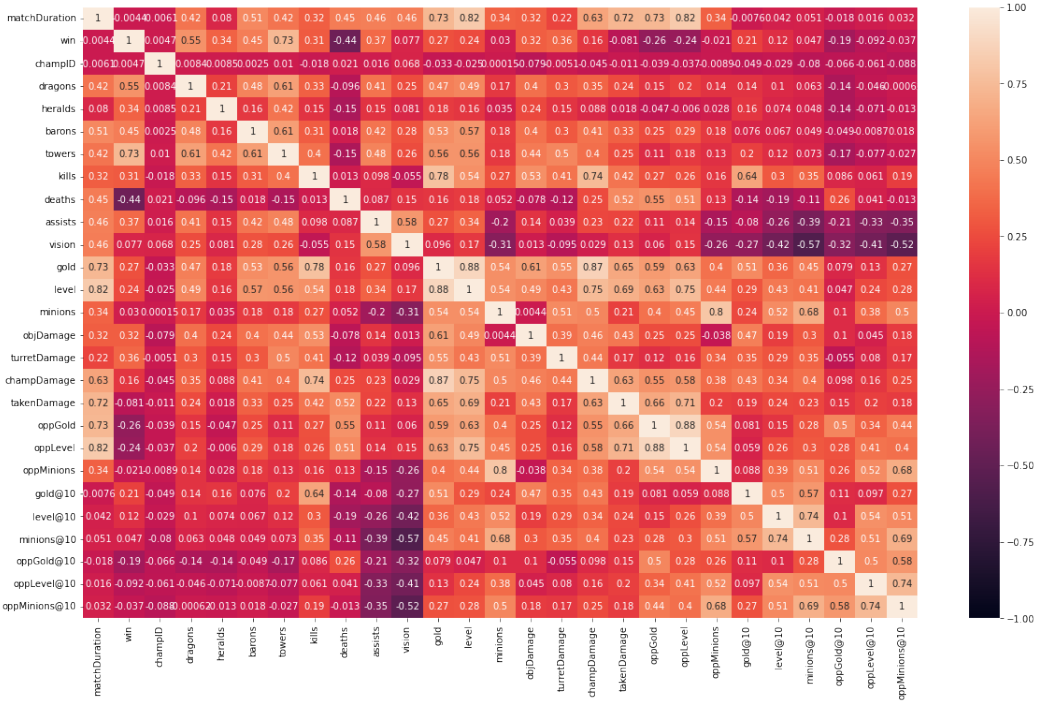
Lanes also feature defensive fortifications called **towers**, which deal damage to enemy minions and players, and can be destroyed for gold. Finally, there are three large monsters that grant boons when slain: **Dragons**, **Heralds**, and **Barons**. Collectively, these monsters, as well as towers, are called game objectives.

Data Set

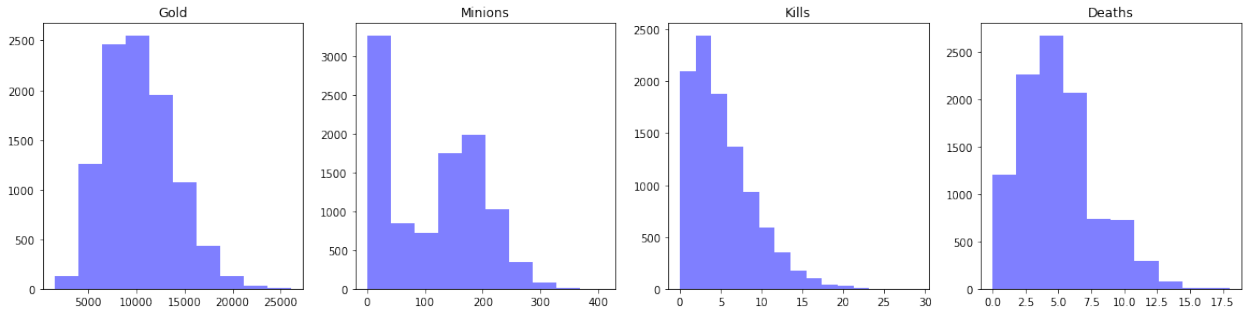
The data set for this analysis was collected directly from the game publisher, Riot Games, by interacting with its League of Legends API. The script first identified the top players from three regions (North America, Western Europe, and Korea). It then generated a list of identification numbers for the players' 75 most recent matches, filtering out duplicate matches in order to avoid conflation. Finally, the script pulled 33 columns of data for a total of 16,932 rows. The columns include a variety of game statistics from the perspective of one of the ten players in that game,

such as the amount of gold earned, minions killed, and character level at both the 10-minute mark and end of the game. Other stats tracked objectives taken and damage dealt.

Initial data exploration revealed some basic, intuitive correlations: match duration is strongly correlated with many counting stats that can only be increased over time, such as objectives taken and character levels. Gold and level are highly correlated, since most activities that grant the player gold also give experience. Additionally, the 10-minute stats are strongly correlated with the opponent's 10-minute stats, which tracks with the typical game experience: most lanes will be evenly matched.



Distributions ran largely as expected, with non-normal distributions matching in-game mechanics. The strange minion distribution is because of a specific role called **support**, which is one of the two players per team in the bot lane. These players intentionally forgo minions in order to give those resource to their partner (the **bot carry**, or **ADC**), whose character typically benefits from higher gold values.



Design Decisions

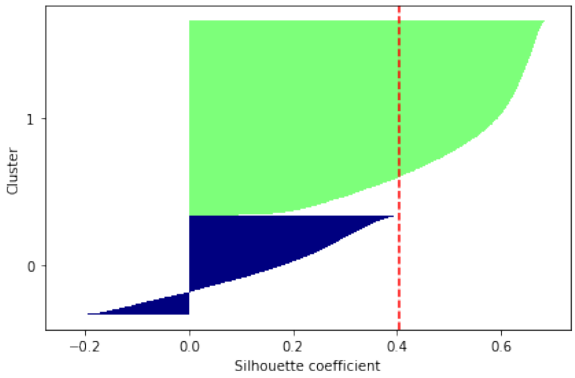
In the data collection process, I did not pull matches that lasted fewer than 5 minutes since games are expected to last 20-30 minutes (anything below the 20-minute mark is considered particularly one-sided). During additional cleaning I decided to also remove games that lasted fewer than 10 minutes in order to maximize the use of columns 27-32, which are intended to serve as a marker of how the early game proceeded.

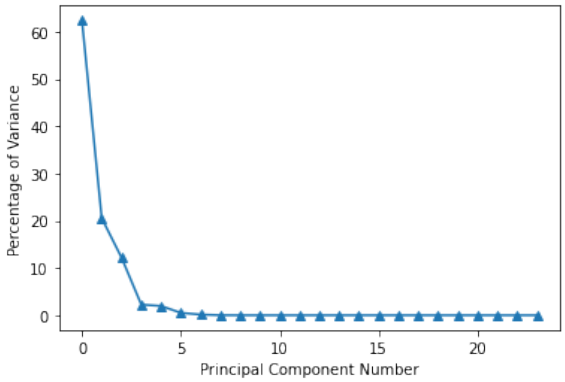
I converted the 10-minute statistics into single +/- stats by subtracting the opponent's statistics from the players. The League of Legends commonly uses combined Kills + Assists / Deaths as a single metric for overall player performance, but I wanted to tease out the effects of the individual effects, so I ended up excluding that column from my analysis.

For unsupervised learning I decided to try clustering and principal component analysis. For supervised learning, I decided to use K-Nearest Neighbors, Random Forest, AdaBoost, and Logistic Regression. I was interested to see how the diverging techniques might have better or worse results. Additionally, I suspected that KNN would be powerful for this dataset but wanted the ability to look at feature importance.

Data Analysis

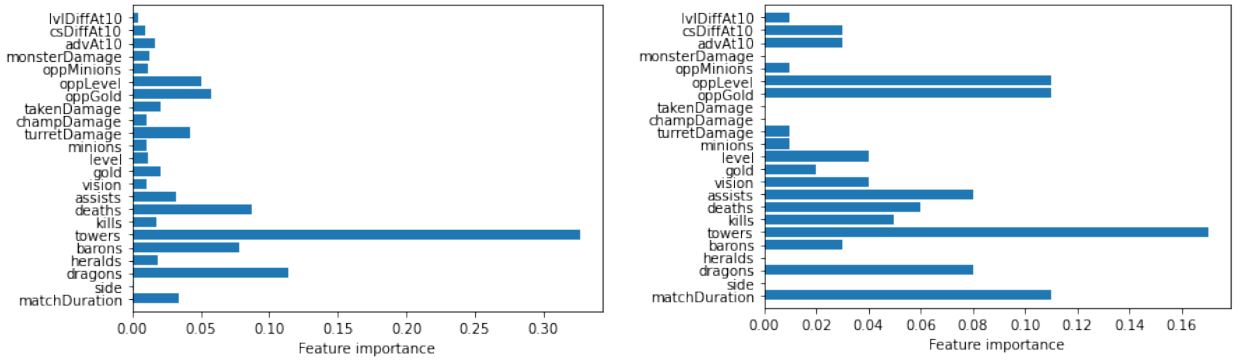
I began by running KMeans clustering, but quickly found that no value of ***k*** yielded good results. After applying Principal Component Analysis and reducing the dimensionality to 3 components, I still wasn't able to achieve strong clusters—the best result was a silhouette score of 0.4 for ***k*** = 2, but only one cluster was particularly meaningful.





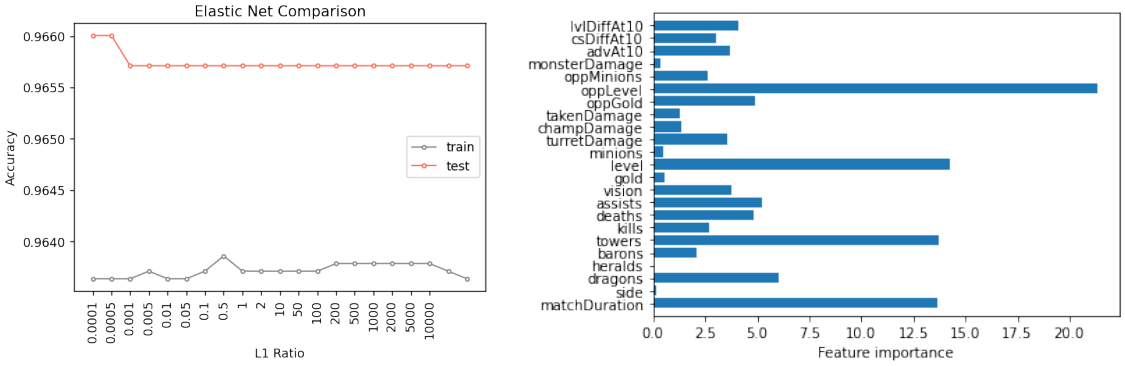
I moved on to K-Nearest Neighbors, and normalized the data with a MinMax transformation. The model achieved excellent accuracy at ***k*** = 15, which I used to pick distance weighting. There may be a small amount of overfitting, but even the test accuracy was quite high at almost 96%.

To determine feature importance, I then moved on to Random Forest and AdaBoost classifiers. For Random Forest, I used a randomized search cross validator to tune the hyperparameters, but found that initial model performance was identical. In the AdaBoost model I used a grid search cross validator to tune learning rate and the number of estimators and was able to increase performance by 1%. Both models achieved 96% accuracy, but had differing feature importances.



Left, Random Forest Features; Right, AdaBoost Features

The final machine learning process was logistic regression, for which I tuned the ***C*** value and attempted to apply regularization via different elastic net values. Overall performance did not change much, but I ended up selecting ***C*** = 50 and L1 = 0.0005.



I decided at this point to break my data down into five role-specific datasets and find feature importance using the best AdaBoost and Logistic Regression models. Since winning in League of Legends requires you to take a minimum of five towers, I discarded the Random Forest classifier, which appeared to have overindexed on tower importance. Towers are still important in the other models, but they place more emphasis on the opponent's level and match duration.

It is interesting that the models place such little importance on CS and gold advantages, considering the fact that obtaining minion kills (and the gold along with those kills) is a primary focus of the first third of the game. Statistical advantages at 10 minutes are also not important, which is a clue that late-game champion performance may be more relevant than early-game prowess.

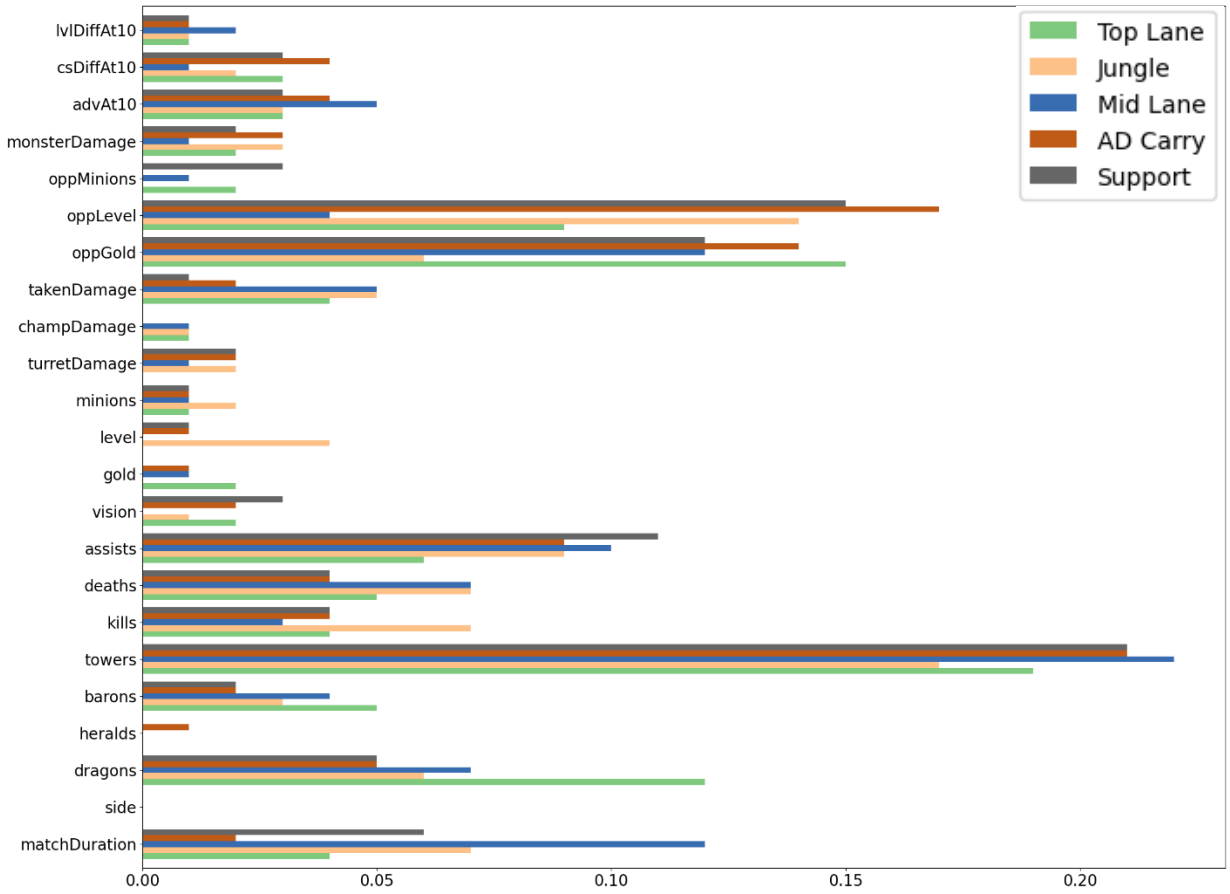


Figure 2: AdaBoost Feature Importance by Role

Rather, one of the most important factors in either model for any role is the opponent's level. In the logistic regression (not pictured), the opponent's level has the biggest magnitude, and is a negative influence on the player's ability to win. The opponent's level is also typically the statistic a player has the least control over, since it would require a significant resource advantage (in gold/experience, or through the involvement of teammates) to deny the opponent experience.

Another striking mismatch is the importance of match duration for Mid Lane players—and the relative unimportance for AD Carry players. This perhaps speaks to the type of champion that is played in those lanes—ones that perhaps benefit from long games where they have a greater chance of improving their strength over time.

The prevailing mindset of most League of Legends players is to defeat players on the opposing team, but the machine learning models indicate that kills are not effective for most roles. Rather, both models place the most importance player levels and towers. This indicates there may be a more optimal strategy than simply killing one's opponent—no matter how fun that aspect of the game may be.