

CSE 190C Assignment 2

Andrew Conegliano and Jonathan Ho

I. DATASET

The data set used for this assignment is from the popular online game League of Legends(LoL) created by Riot. LoL is a team based game: two teams of five players fight each other on a map with the end goal of destroying the main building in each team's base. Each player chooses a champion, which has special and unique abilities, and goes to one of 4 lanes:top, mid, bottom, jungle. Players gain gold from killing other players and creeps, or monsters on the map, and buy items that make the player stronger. Experience is also gained from killing creeps and players and allows the player to gain new abilities.

Data from this game however is not public, so we had to create our own script that mined data using Riot's API. Our script does three main steps:

- Gets a list of the top ~750 players in North America.
- For each player, gets the last 10 matches played.
- For each match, gets the match data.
- Repeat every 5 hours. (average game length is 30 minutes)

The players chosen are from Master and Challenger leagues. LoL has a ranking system based on leagues, with these two leagues being the highest rank. Our script has been running since Friday, March 15, 2015. Basic statistics are shown in the table below.

TABLE I
BASIC STATISTICS

Number of Games	10,000
Unique Players	812
Average length of match	30.85 minutes

The reason for the low amount of games is because the players are at the top of the ranking system, the games they are matched with are with people of similar rank. So alot of these players are playing against each other, and the amount of unique games with these players decreases dramatically.

One of the first statistics we were curious to see is the which champion is the most popular. As seen in

Fig. 1, the top 10 most picked champions are shown. Thresh and Janna have been considered some of the best support champions for a while, so its not surprising to see them in the list. It is surprising though to see them as the highest picked champions; players are prioritizing these two support champions instead of mid or top lane champions. It can be reasoned that because these two champs have abilities that stun/knock-up your opponent, they are great for making plays (a series of actions that benefits the team). The rest of the champions on the list make sense because they are considered very strong this season and are champions that can carry games.

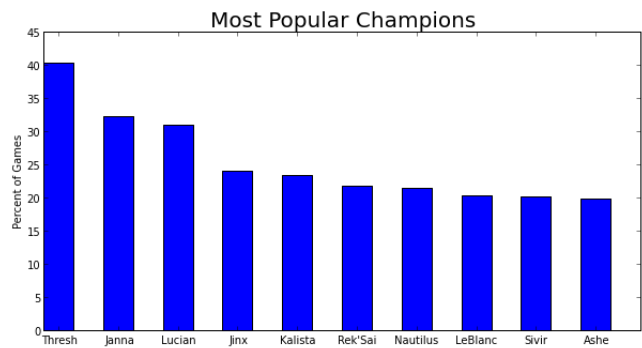


Fig. 1. Most popular champions

Another statistic we created was to see which champion leads to winning. Fig. 2 shows the champions with the highest win rates. Whats interesting here is that depending on champion popularity, win rates change dramatically. We can see that Karthus has the highest win rate, but there are only 263 games played with him, out of 10,000. And Quinn is even lower, with only 49 games. We can reason that people who play these champions have a very high expertise with them, and because of that are more likely to do well, and win. Champions like Ashe and Nidalee have a very high number of games, and we can conclude that champions like these are relatively better than other champions; you don't need to have a high expertise like with Karthus or Quinn.

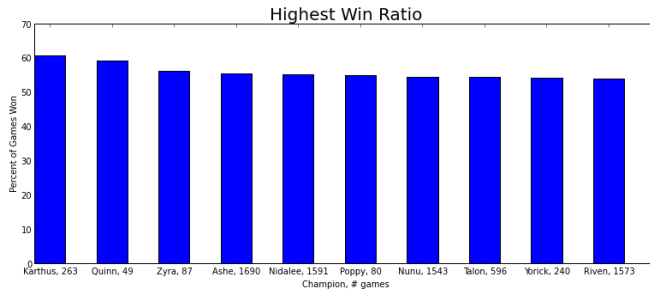


Fig. 2. Champions with highest winrates

Lastly, there are two heat maps in the Appendix. We wanted to see which items are most popular for each champion. The data was normalized so non-popular champions will have the same scale as popular champions. For champions like Vladimir and Irelia, it is well known that every player should get Will of the Ancients and TriForce respectively. This can be seen on our graph, which agrees with our intuition.

II. PREDICTIVE TASK

The predictive tasks we identified for our dataset are tasks concerned with match outcomes based on player performance after the match has happened and one where player profile averages are used to predict match outcome before it has been played. To predict a match outcome after it has been played we model a team based on the players that make up that team and make a player profile for each of them with features we select. To find the features to use to build this player profile we run Principal Component Analysis (PCA) on the statistics field for each player after a match is played. The statistics field holds 58 fields and after running PCA on those 58 fields we select the fields that have a variance greater than or equal to 0.01. This leaves us with 11 fields from statistics to build a player profile on. Some fields include player kills, largest killing spree, and magic damage dealt to champions. We kept these features with variance greater than or equal to 0.01 because skill level of top tier players vary less among each other compared to low to mid tier players. To predict a match outcome before it has been played based on the composition of players that make up a team we build player profiles that hold averages for the player for each of the 11 features we use to build our feature vector. If the player has never been seen before, we use a global average per feature of all the players we've seen.

Now that we have a set of features to build our feature vector we build 5 player profiles and group

them together to represent a team. Thus our feature vector length is 55. We chose Linear Regression, $X\theta = y$, because we are building a classifier and Linear Regression is good for fitting our features into a line similarly to what we learned in class. For our training set we use 90% of the data we collected to build it and the remaining 10% as our test set, and shuffled it before feeding it into the model. For a baseline comparison we refer to research done by Colin Feo, Jeremy Ma, and Allen Sirolly for their work on "Predicting Winning Teams in League of Legends (LoL)". Their use of Logistic Regression gives them a classification accuracy of 61.57% for their training set and 61.19% for their test set.

To assess the validity of our predictive tasks we need to see an improvement over guessing which is simply 50%. For our first predictive task on predicting match outcomes after the match has been played we need to see an improvement that is much better than guessing. Based on several important game scores post-match our model should be able to reasonably predict the winner. For our second predictive task where we use player averages to build their player profile we need to predict better than guessing again. Without the match being played and no available in-game information and based on past player performance we should be able to predict better than guessing. Guessing who will win a match before it's played is 50% and if our model performs worse than guessing then that does not validate our work.

III. RELATED WORK

We found two sources of literature that relate to our work. One paper by Jeremy Blackburn and Haewoon Kwak use predictive models to predict toxic behavior (behavior that negatively affects other player experiences). Another paper by Colin Feo, Jeremy Ma, and Allen Sirolly that aims to predict winning teams. While the latter is more closely related to our own study, The former is relevant to our study as well based on models and decisions making.

Blackburn and Kwak's paper collected data from Riot Games Tribunal cases. The data collected here are player reports against an offensive player. In total Blackburn and Kwak were able to collect 1,460,344 cases. Each case is reviewed by other players and punishment doled out by Riot Games. In each case are player reports that note the toxic behavior exhibited by the offensive player. Similar to our study this paper looks at in-game performance and models it

to predict behavior, while we model ours to predict match outcome therefore our conclusions won't match but it's interesting to note their approach to using in-game performance and Tribunal reports. Blackburn and Kwak use a supervised-learned classifier to answer their questions. Our approach is the same in that we look at well-labeled data to build a prediction model. One difference we should note though is that the data collected by Blackburn and Kwak do span across different regions such as Europe and Korea while our data set is exclusively in the North American region.

Now onto the piece of literature that relates most to our study. The research done by Feo, Ma, and Sirolly uses the same data collected from Riot Games. However our approach and data set differ greatly. Their approach was to look at rank matches in the middle tier where skill varies greatly. Our approach is to look at only the top tier of players to predict what makes them successful in game. The data set size and use of the data set differs greatly as well. They used 4,000 matches and split it evenly 50/50 for their training set and testing set. Our approach uses 10,000 matches and we split ours 90/10 for training and testing respectively. Their data collection method was to use a Python library that wrapped Riot Games REST API. For our data we collected it manually using only API endpoints and continuously running it on a server. Feo, Ma, and Sirolly used 6 different approaches to predict match outcome and they were Logistic Regression, Linear Support Vector Machine, Decision Tree, Naive Bayes, Random Forest, and Stacked Ensemble. For our study we only used Linear Regression and Stochastic Gradient Descent. In both our studies the way we built our feature vector started at the same by using Principal Component Analysis to reduce dimensionality of the data set we collected to find meaningful features. Our use of PCA differed from their approach because we selectively picked features and they used PCA on the entire space. This is necessary in our study because we are targeting features that highly skilled players rely on more than say a middle tier player would. In comparing the results of our findings we see that Feo, Ma, and Sirolly have better results because they predict on lower ranking players which have higher variances when running PCA which leads to more accurate results.

IV. RESULTS

Using our Linear Regression model we get 91.7% classification accuracy for our training set and 90.6%

for our test set when predicting after the game finished and we have all the results. Our accuracy jumps very low when trying to predict the winner of a game before it happens. We have 55.6% accuracy for the training set, and 52% accuracy for the test set. The baseline we compare to have a 61.57% classification accuracy for their training set and 61.19% classification accuracy for their test set. The reason their model works so well is that they use data from lower ranking players, and have a more sophisticated model. Trying to predict the highest ranking players is extremely challenging, because the differences in the features are very small. Our results aren't surprising. If we could create a model that predicted wins with a higher accuracy, we would be able to use this information to bet against who will win and make lots of money, like in other sports. Predicting outcomes of sports, and e-sports in this case, is extremely hard.

APPENDIX

Most Popular Items per Champion

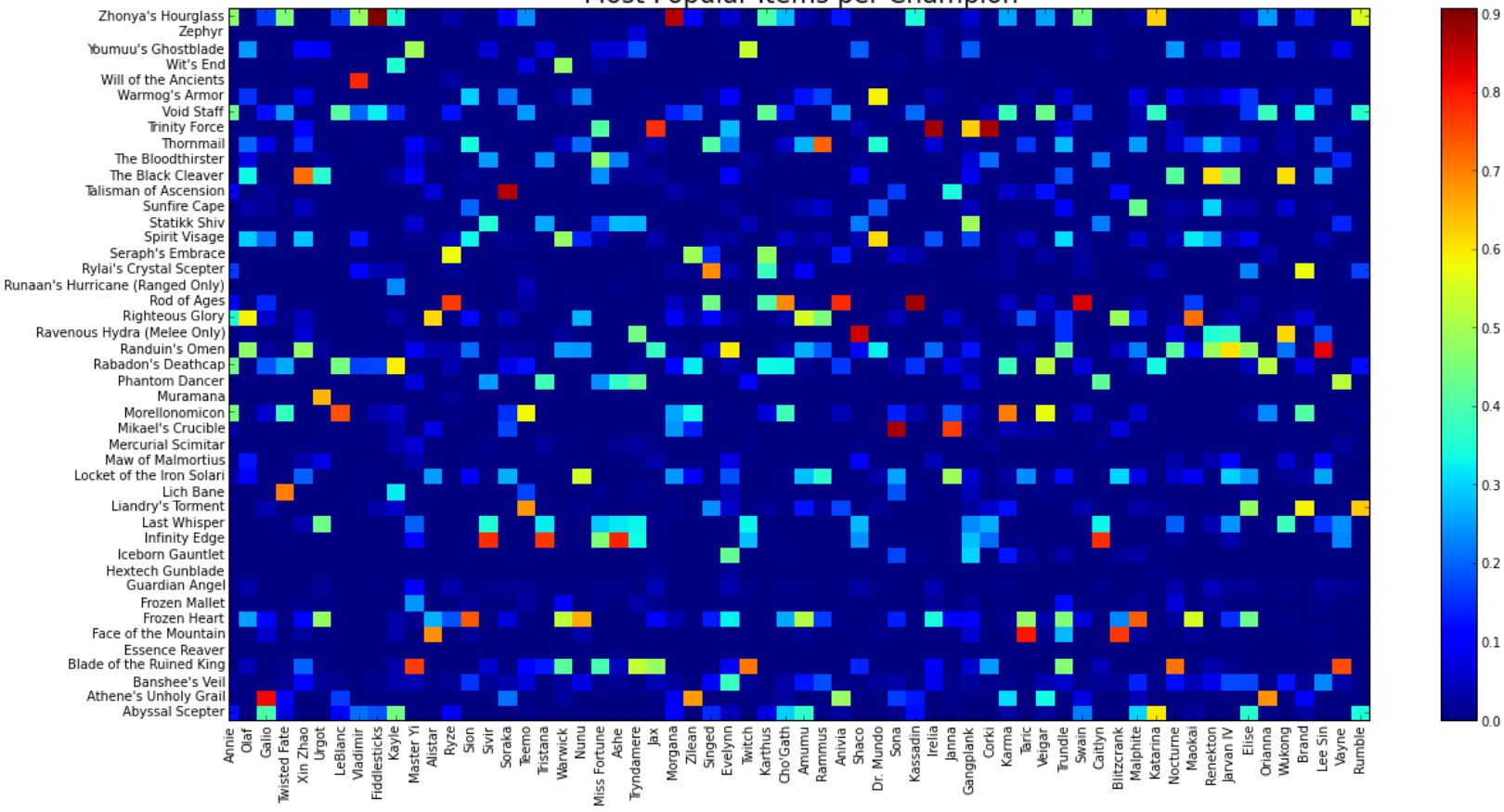


Fig. 3. Most popular items for reach champion, part 1

Most Popular Items per Champion

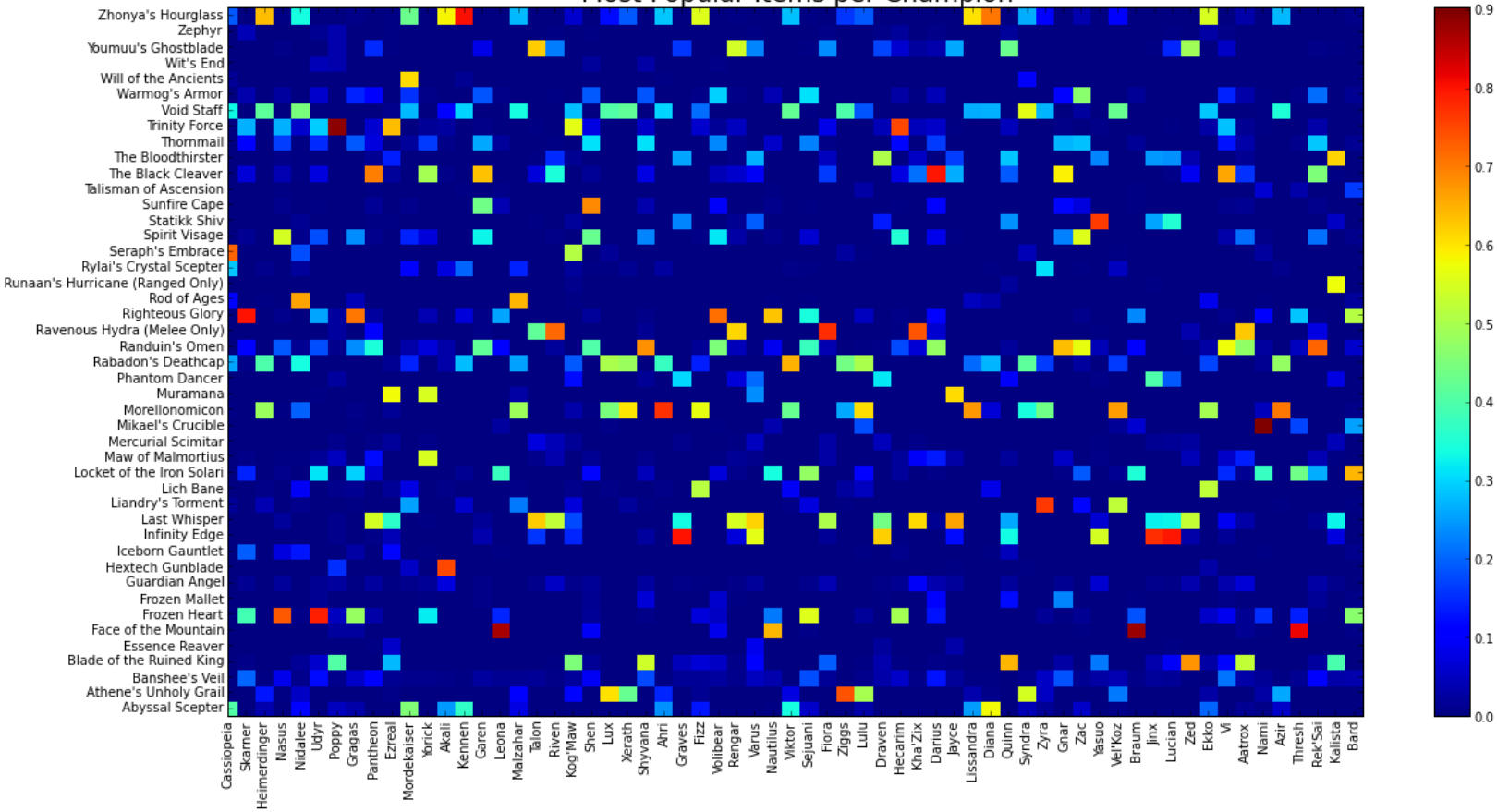


Fig. 4. Most popular items for reach champion, part 2