Ronald Tun A12458764

Louise Jensen A12596562

Phillip Lagoc A13469618

League of Logistic Regression

Abstract

Our dataset, obtained from Kaggle, contains thousands of observations from professional games of *League of Legends*. Each observation has 57 predictors which tell us how well each team is doing at the end of the game. All of this data will be used to fit a model that will help us view the data in a way that allows us to answer relational problems, such as predicting which team is going to win, as well as identify what factors matter most in victory on the rift. We will utilize PCA, K-folds cross validation, logistic regression, and SVMs in order to find the best model to work with. From there, we found that the most important factors are towers destroyed and gold discrepancy, and both models are as effective in making predictions.

Introduction

Thousands of people at any given moment are playing the popular Massively Online Battle Arena (MOBA) video-game *League of Legends*, which was developed by Riot Games and has since gained millions of players worldwide. In *League of Legends*, two teams of 5 battle each other to destroy the other team's Nexus. Each team's Nexus is guarded by Inhibitors and Towers, which must be destroyed first. In addition to destroying these objectives and fighting each other, players can also defeat the monsters (Baron, Dragon, and Rift Herald) to give them a better advantage in winning the game. Communication is key in *League of Legends*, but what should they be prioritizing? Every seasoned player understands what is needed to take down the other team's Nexus, but what will they be going for? Every advantage leads to victory, but the question is: what are these advantages? Our model will be able to predict whether or not Red Team won based on predictors concerning the statistics of the game, and show us which variables contribute the most to victory. We hypothesize that gold discrepancy between the two teams, along with destroying monsters (otherwise known as neutral objectives), will be key predictors in determining if Red team won. Gold is a crucial component of the game, as it can be used to power up each player, giving a clear advantage. For monster objectives, each monster gives a buff, which are traits that empower a player such as increased health, that can help one team dominate another. Also, we predict that SVMs will be more accurate given their robustness.

Materials and Methods

The dataset we are training our model from has been growing for three years, containing data from professional games of *League of Legends*. Over these years, it has accumulated data from 7620 games, with 57 predictors for each and every one of them. Most of these predictors contain quantitative values, such as gold discrepancies and objectives captured, while others contain categorical data, like what champions were chosen for which role.

To pre-process the data, it was necessary to understand the data first. While some of the variables were continuous (eg. the length of the game), many of the other variables were strings made to look like lists. For instance, the variable *golddiff* represents the difference in gold gained by each team per minute. So, it was a string that looked like this: “[0, 50, -10, 60]”. We want to pre-process features like this in such a way as to extract the values inside. So, we split the string by commas, converted each value into an integer, and appended it to a new list. We then calculated the mean of this new list, and concatenated it to the existing data. We processed many of the other features this way, separating the string using the appropriate delimiter, extracting the values, then appending these new values in a meaningful into the data. Below is a table showing the specific steps we took to pre-processing certain data:

Table 1. Table of Pre-Processing Steps. For each of the features in the original data that was a string made to look like a list of values, we pre-processed them in such a way so that they are appropriate to perform analysis on. Below is a table depicting the feature's name, content, and how it was pre-processed.

Feature Name	**Overview**	**How it was pre-processed**
golddiff	A String that contains the gold difference (Blue Team - Red Team) per minute.	- Delimit the String, and extract the gold differences per minute. - Compute the mean gold difference for the entire game.
goldred & goldblue	A String that contains the total gold per minute for Red Team and Blue Team, respectively.	- Delimit the String, and extract the total gold per minute. - Subtract each value by the previous to get the gold earned. - Compute the mean of the gold earned for the entire game.
rKills & bKills *rInhibs & bInhibs* *rTowers & bTowers* *rHeralds & bHeralds* *rDragons & bDragons* *rBarons & bBarons*	A String showing a list of each Team's kills, inhibitors taken per minute, towers taking per minute, Rift Heralds taken per minute, Dragons taken per minute, and Barons taken per minute.	- Delimit the String - Count the number of occurrences of the desired item, which is the length of the delimited String.
rTopChamp & bTopChamp *rJungleChamp & bJungle Champ* *rMiddleChamp & bMiddleChamp* *rADCChamp & bADCChamp* *rSupportChamp & bSupportChamp*	A String showing the Champion chosen for each role for each Team.	- Find the unique Champions that showed up in the data. - Label if the Champion was used with a 1. Otherwise, label with a 0.

After pre-processing, our data had 7,620 observations and 157 features. The data will be split such that 80% of the data is used for cross-validation, and 20% is used for testing the model. The label we would want to predict would be whether or not Red Team won. Given the amount of features, it would be best to select a subset of them in order to avoid overfitting. So, we will use PCA and create 20 principal components. Afterwards, we will use 100-fold cross-validation to determine how many principal components we should use, using both logistic regression and SVMs as the model to compare the validation accuracies. A plot will be made for each model comparing the validation accuracy, and the model that achieves the highest validation accuracy with *k* principal components will then be trained on the entire cross-validation set. Then, the testing accuracy will be computed with the best logistic regression model and SVM, and finally compared to each other.

Results

The feature to give the greatest standard deviation was the mean gold discrepancy between the two teams. After performing PCA using logistic regression as the model comparing 1-30 principal components, the model to give the highest validation accuracy was when the first 5 principal components were used. Using more than 5 principal components led to overfitting, made evident by the large gap between the training accuracy and validation accuracy. This is shown in the plot given by Fig. 1. After training our logistic regression model on the entire cross-validation set using the the first 5 principal components, we tested it and got an accuracy of about 0.97.

Similarly, we performed PCA using SVMs as the model comparing 1-30 principal components. The model with the highest validation accuracy was when the first principal component was used. If more than about 5 principal components were used, there would be overfitting, which is shown in Fig. 2 by the extremely gap between the training accuracy line and the validation accuracy line. After training, we tested it and got a test accuracy of about 0.97. This is extremely similar to the accuracy we got using logistic regression. Using the first 5 principal components for logistic regression is as effective as using the first principal component for SVMs.

After training our models, we analyze statistics regarding each unique champion as well as statistics regarding each objective. The unique champion data is shown in Fig. 3 and Fig. 4, where we can see the top 10 champions used and top 10 champions with the highest win rates, respectively.

Fig. 3 shows the 5 most picked champions throughout the data. These numbers suggest that these champions are considered the best in their role, as professional players choose champions that they believe give the greatest advantages for victory.

Looking closely at Fig. 4, the champions do not necessarily match the ones shown in Fig. 3. In fact, none of the figures share champions. This is because the champions of Fig. 4 were rarely used, but when they were used, they helped win the game. Table 2 shows the number of times the champions in Fig. 4 were used.

Fig. 1: Plot of the Logistic Regression Accuracy vs. the number of principal components used. The highest point of the validation accuracy (blue) line would be k = 5 principal components. The best validation accuracy was achieved using the first 5 principal components, which is 0.966.

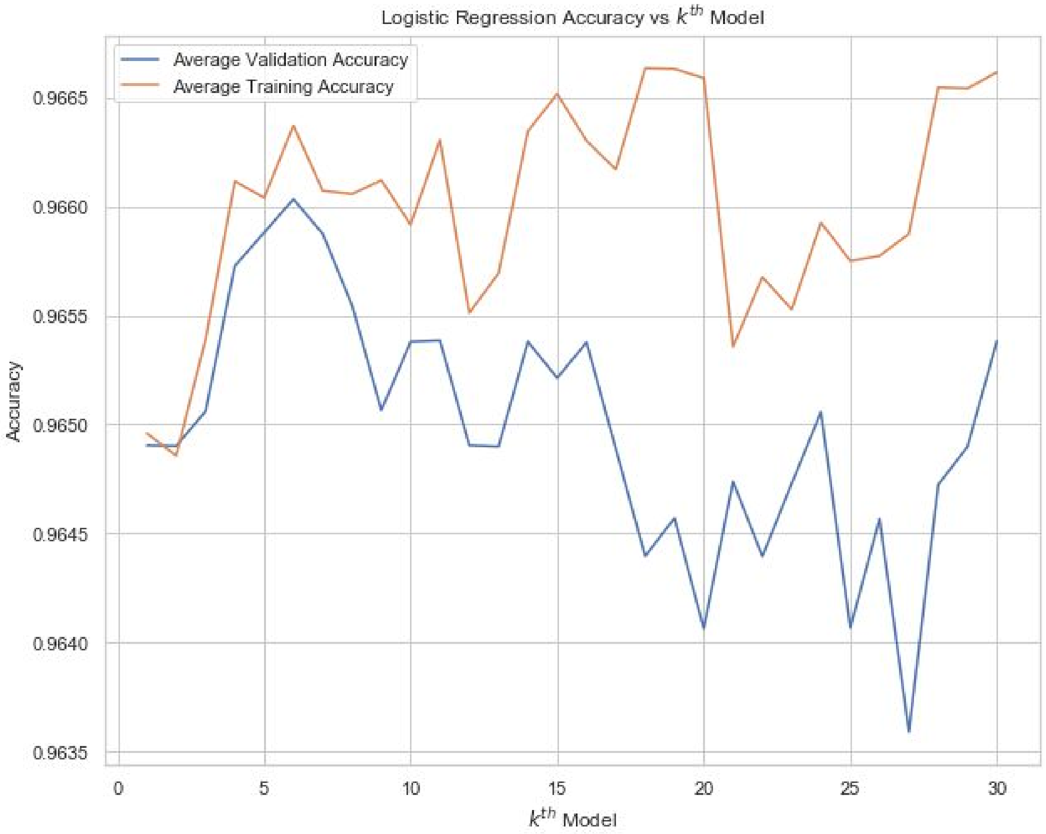


Fig. 4 and Table 2 both show an interesting trend in the data. These champions were not considered suitable for winning. What this means is that choosing them would put you at a clear disadvantage - other champions offer far greater advantages in terms of their abilities, and so you would waste your choice of choosing a better champion to play. Choosing such champions was considered “off-meta”. However, these champions were not completely useless, and could effectively counter the other team if the conditions were just right. These champions would have only been picked *if* conditions were favorable.

Fig. 2: Plot of the SVM Accuracy vs. the number of principal components used. The highest point of the validation accuracy (blue) line would be k = 1 principal components. The best validation accuracy was achieved using the first principal component, which is 0.965.

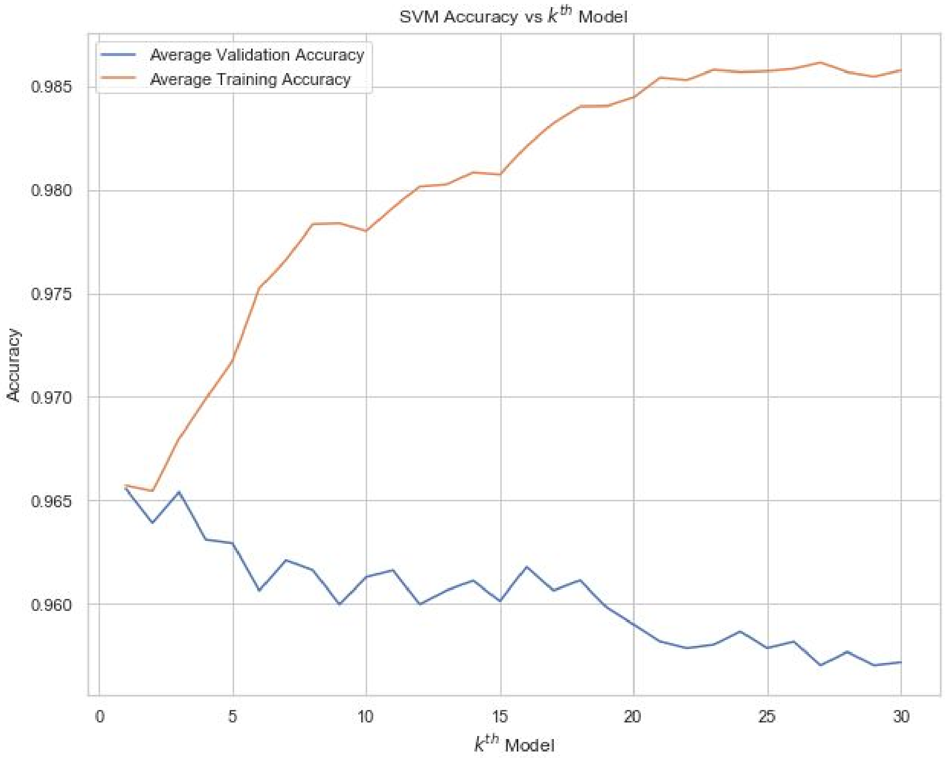


Fig. 5 shows a correlation heat map. The predictor with the highest correlation to victory was towers destroyed. This makes sense considering you need to destroy at the very least five out of the eleven towers on the enemy base before being able to destroy the nexus. This may differ from what one might find in real games, due to the fact that professional players are much less likely to forfeit. Those who have less at stake like casual players at home might find that towers aren't quite as vital to winning. If the skill gap is high enough, one team could gain a large enough advantage without killing towers to make the other team surrender. This would lead to less towers being taken down by winning teams. The predictor with the second highest correlation with victory was the one we hypothesized. A large gold discrepancy is highly correlated with winning the game.

Figure 3. Top 5 Champions Used. Shows the top 5 Champions used throughout the entire data.

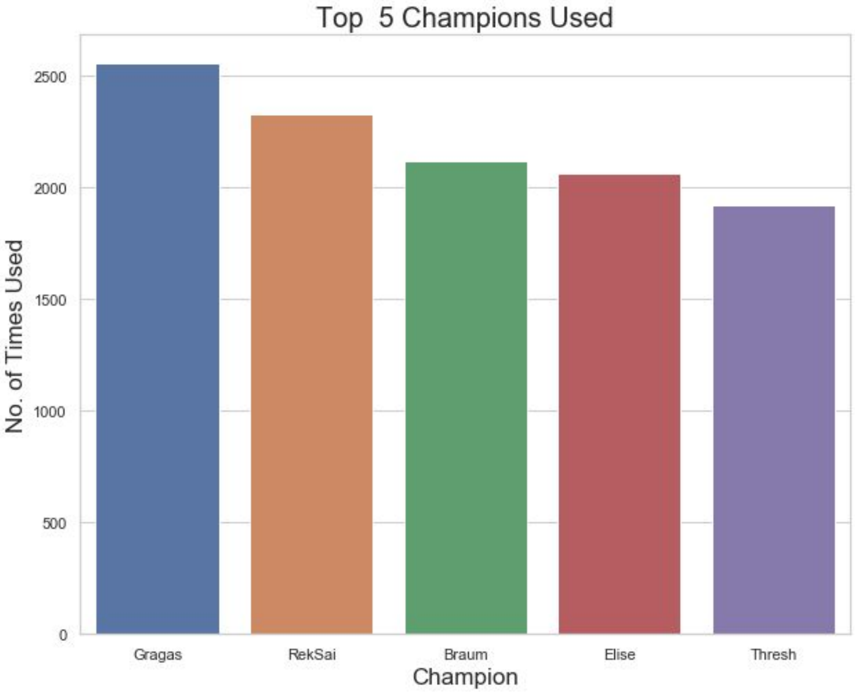


Figure 4. Top 5 Champions with the highest win rate. Shows the top 5 Champions with the highest win rate throughout the entire data.

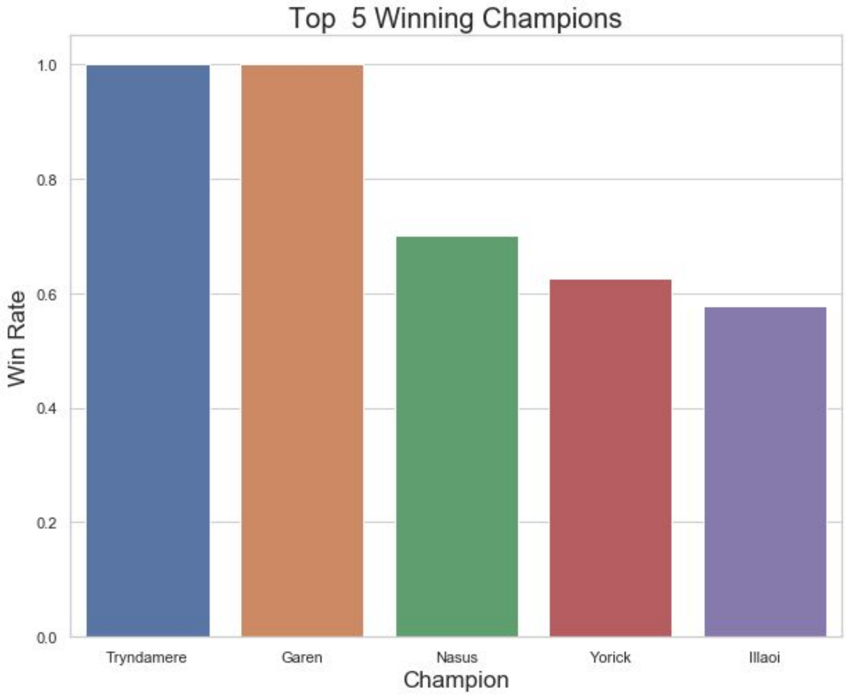
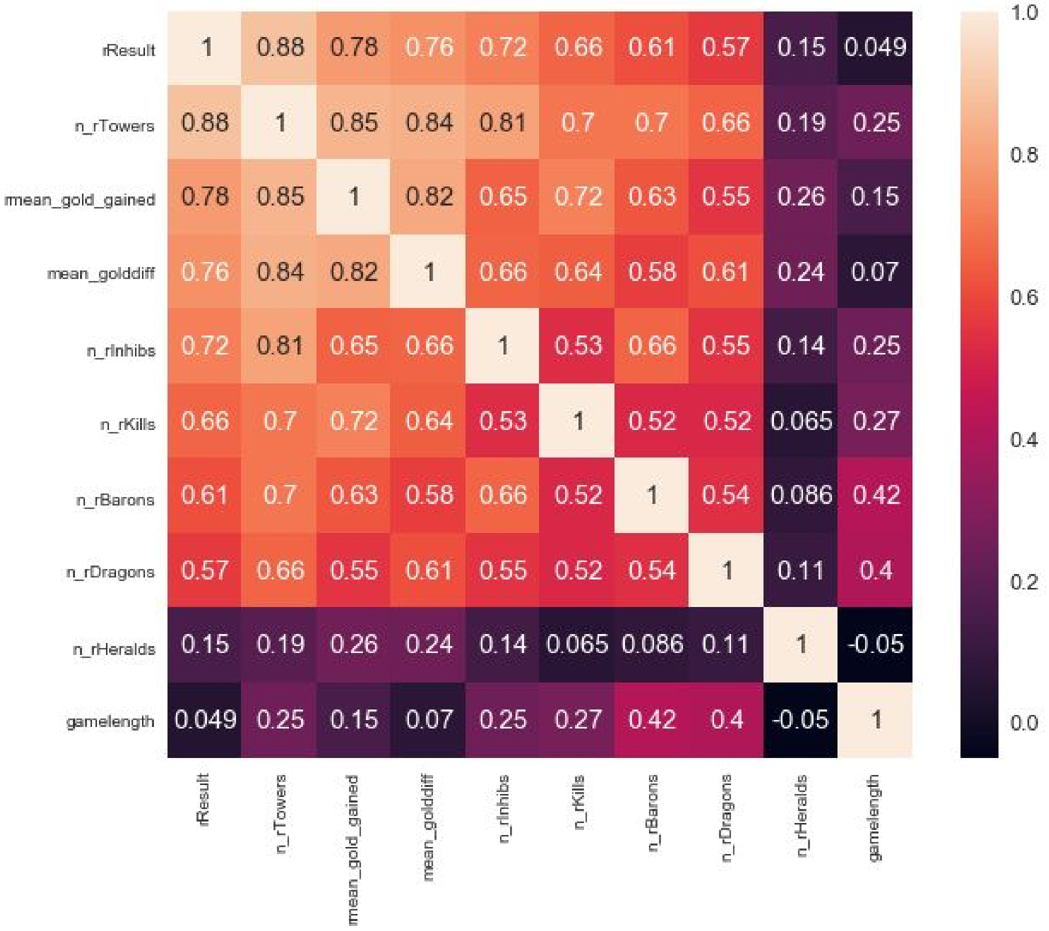


Table 2. Table showing the top 5 champions in Fig. 4 and how many times they were played, as well as the value of their win rate.

Champion	**No. Times Used**	**Win Rate**
Tryndamere	1	1.0
Garen	2	1.0
Nasus	10	0.7
Yorick	8	0.625
Illaoi	38	0.578947

Figure 5. Correlation Heat-Map for the top 10 features that have the highest correlation with whether or not Red Team won.



Discussion and Conclusion

Our predictions were based off of our own experiences with the game, and from them, we have predicted that the gold difference between each Team would be highly correlated with winning, which seems to have been validated in Fig. 5. Gold discrepancy seems to be strongly positively correlated with Red Team winning. This makes sense, as the entire game is spent trying to obtain gold. Also, gold discrepancies are caused by team skill, which is can lead to victory.

What we did not expect were the impact of champion picks. Often times, casual players just pick what champions they are comfortable with, while a professional is comfortable on much more champions than not. This puts a lot more importance in champion picks when looking at professional games, and as casual players, we overlooked this. We also failed to foresee the weak correlation between Red Team winning and the neutral objectives, which include Dragon, Baron, and Rift Herald. When taken, these grant buffs (advantages) to every player on a Team, who can then use these advantages to win the game. Based on personal experience, these buffs can drastically change who will win a game. As such, we hypothesized that they would be as strongly correlated with winning as gold discrepancy. However, this was not exactly the case. Finally, the similarity between the results of our trained logistic regression and trained SVM were surprising, as we had initially thought that SVMs would be more accurate given their robustness to outliers.

In the end, our experiment opens up more questions for analysis. Which champions, for each role, contributes more to winning? When a certain role starts to dominate, how much does this contribute to winning? Since we decided to ignore the human players, another experiment could look at their contribution to winning, such as which teams took the most wins during a season. Perhaps the biggest improvement that can be made to our experiment is the effectiveness of our model. Due to the amount of features, it is possible that our model is overfitting the data. As such, improved feature selection methods may prove helpful here, such as using best subset feature selection after performing PCA. For the SVMs, one might consider tuning the hyper-parameter *C* for even better results. Overall, our data shows that crucial game objectives, such as destroying towers and earning gold, are highly correlated with winning, and that logistic regression performs as well on the principal components of the data as SVMs do.