League of Legends Match Outcome Prediction

ANLY 590

Zikai Zhu, Ling Yang, Xunge Jiang, Jingrong Tian

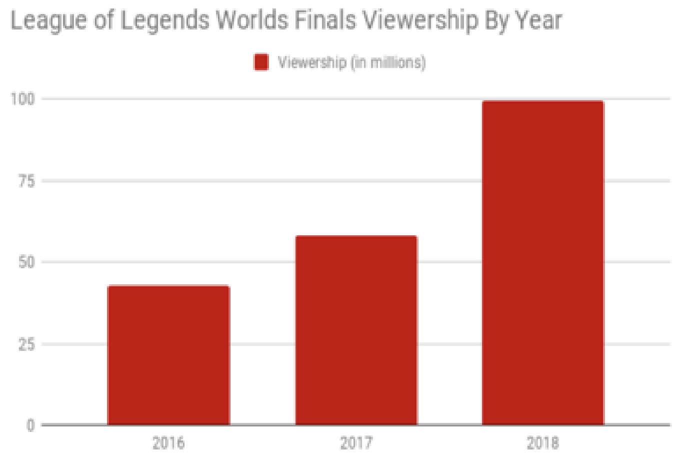
I. Abstract

League of Legends (LoL) is a popular multiplayer online battle arena (MOBA) video game. Two teams of five players control characters called "champions" and compete to complete objectives. With the growing number of people entering the esport betting, more and more money is thrown into each world cup game match. Therefore, bettors are expecting higher returns from their bets. The research is to make predictions on the outcome of LoL World Cup matches based on historical match data, to help numerous bettors know which side is much closer to champion, and make more accurate decisions on where their money should go. The results compare gradient boosted trees and artificial neural network to predict the match outcomes of LoL. The dataset is scripted from Riot Games API--attributes including both team and player stats. The paper is going to introduce the match outcome prediction, some related works, and also the methodologies for the project.

II. Introduction, Background and Motivation

League of Legends (LoL) is a popular multiplayer online battle arena video game developed and published by Riot Games. The game is played between two teams of five players each. Each player controls his or her own champion with its own economy and champion specific abilities.The win requirement is to kill the other team's nexus, which is a specific building. To achieve this condition, the whole team should collaborate together to gather resources (gold), experience and equipments for their champions so that they can gain an edge versus the other team.[1]

[1] "Bet on League of Legends (LoL) ᐅ Esports betting." https://esportbetting.eu/games/league-of-legends. Accessed 2 Oct. 2019.



During the past years, the number of players for LoL keeps increasing, and in 2019, League's active player base has grown rapidly with over 120 million players worldwide. The number of viewers for annual World's Championship of LoL also kept increasing and reached 100 million last year. Most viewers enjoy the match online from many resources including Youtube, Twitch, Facebook and other language platforms.

As one of the most popular video games, League of Legends also draws a large amount of money through esports betting. As time goes by, people with previous experience on betting and gambling side of American football have been enjoyed with their returns and excitement coming from betting on the outcomes of the games. And with esports expanding into more and more people's entertaining lives, bettors crowded into video games field to continue their exciting game..

A large amount of online betting providers have popped up offering betting chanced to those interested in the bet. Gamblers can view live matches and place bets through online entrances on League of Legends and other matches. Beyond that, addictive eSports is becoming more prominent. Participants earn points and cash back based on how those players perform during games, as some of them also play the game, they may have chances to obtain surprise gift in the game for their success.[2]

The total amount of money put into the betting is unimaginable, so the research aims at helping those unprofessional bettors to earn back more from their bet, by making more accurate predictions on the match outcome to help bettors make decisions.

[2] "The Rise of eSports Betting & Gambling - Business Insider." 9 Jan. 2018, https://www.businessinsider.com/the-rise-of-esports-betting-and-gambling-2018-1. Accessed 2 Oct. 2019.

III. High-level Problem Definition

This project is built for people who watch League of Legends matches and are willing to bet. To make the problem more detailed, our project is going to predict the match outcome. That is, through training the model by match data, the model can tell people which team will win in this game and the probability of winning.

IV. Detailed Problem and Challenges

People who bet on the match result usually make decisions simply based on their personal preference or the historical match results of each team, so if the two teams in one match are on a similar level, the correct rate would be 50%. The research builds and trains machine learning models to figure out the internal logical relationships among detailed statistical data coming from previous records for each team and their team members. With the help of our model, bettors could pick their side with higher winning probability.

Our major challenges happen in our dataset. Our dataset contains all the match data from all the regions including the World Championship from 2017 to 2019. There are about 100,000 data in total, including about 20,000 team data and 80,000 player data. The dataset has 96 attributes, which brings in our first challenge, which is the dimensionality. Every column looks reasonable so it is impossible to delete irrelevant columns by hand. However, the model does not need high dimensionalities. So we need to work on feature selection and engineer to narrow down the attributes first. Also, since we have both player data and team data together in one dataset. For example, one match has 12 rows of data. 10 of them are players data such as kill, assists, damage, damage per minute, gold earned, etc. Two of them are team data for red team and blue team with similar columns. With team data and player data, we build our model through either team data or player data. That makes our project differentiates in some ways but also gives us more challenges. What's more, a match contains 12 rows, which makes it difficult to clarify the format of each data sample. For example, the problem is, we should have two teams and related parameters as input so that we can get each team's win probability. It is ambiguous that which type of train data we should use. For this step, the choice is to train the model using all the players data and we get a list of parameters after we input all the players in two teams for the match we want to predict. And then, we have a second predicting step for the whole team.

V. State of Art (Related Work)

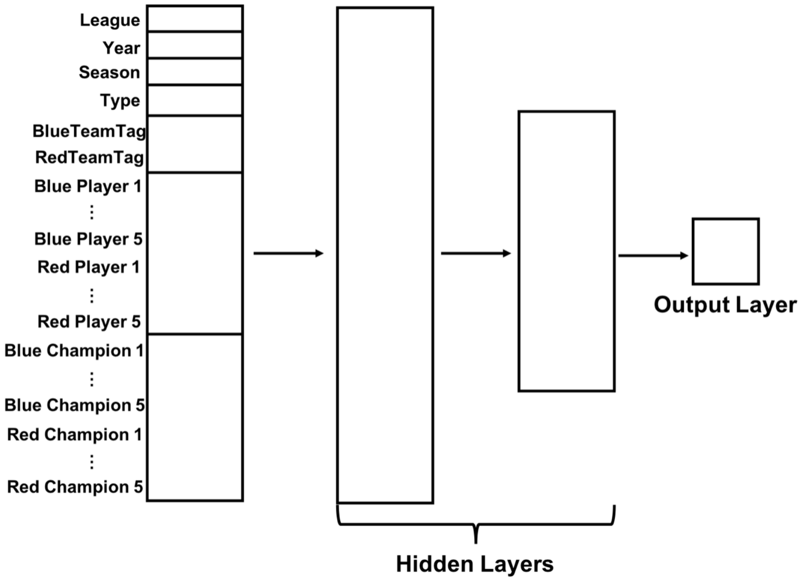
Due to the popularity of League and Legends and its related betting activity, many people have conducted analysis on how to predict the match outcome more accurately, from different aspects like statistical side and technical side. There are several papers using embedding and deep learning methods, especially using Neural Network. The most advanced research is conducted by Yuan Tian using dataset from Kaggle, named “Predicting League of Legends Match Outcome with Embeddings and Deep Learning”, to make pre-match outcome prediction by giving out a winning possibility in percentage.

In the research, originally one-hot encoding is used in feature engineering process to represent categorical features. However, there is a disadvantage of one-hot encoding because sparse representation can use a lot of storage if a feature has many different values, and the inner product of any two values is zero, thus one-hot encoding would not be able to capture the relationship between them.[3] This problem could also be a potential challenge for our research, as there would be categorical features including so many categories.

In contrast to one-hot encoding, embedding uses a set of features to represent each value, and this is the choice the author has chosen. Embedding could save storage as the number of features is usually smaller than cardinality. What’s more, embedding could learn the relationships between values. For example, we can learn a champion embedding matrix where each row represents a champion and the columns may indicate their defense, health, speed and so on. We learn an embedding matrix for each of the categorical variables including league, year, season, type, team tag, players and champions.

When it comes to data modeling part, extract the embedding vector for the values of the 26 feature columns, vertically stack them together in one dimension and then feed the stacked vector through a neural network. Thus, the following diagram represents the final model:

[3] "Predicting League of Legends Match Outcome with ... - Medium." 24 May. 2018, https://medium.com/@yuan_tian/predicting-league-of-legends-match-outcome-with-embeddings-and-deep-learning-b7d1446c710f. Accessed 2 Oct. 2019.



[4]

The paper shows that after training the model for epochs, the prediction accuracy for the training set and validation set is 73% and 67%, respectively. From the number, we could see that the performance of neural network is not very high, and there is still a lot to be done to improve the accuracy.

Another popular research is also using machine learning, neural network to make predictions on pre-match results, based on 50 variables from previous matches, named "League of Legends Match Prediction". This paper is using similar modeling idea compared to the previous one, but the author has raised a crucial point about how the matching system from Riot's matchmaking system would limit the accuracy of prediction system.[5]

The available research online mostly use datasets with fewer columns than us as we have 96 features total while most of other existing research only have fewer than 30 features. So with more detailed features, this research is expected to produce more accurate prediction of outcome. Since there is an intelligent agent selecting the matchups for League of Legends, the matchmaking system has to balance creating fair team matchups with doing so in a timely manner. Therefore, the author inferred that Riot's matchmaking algorithm is likely not optimized entirely for matchup fairness. This means that a prediction system with a sufficient amount of data may be able to achieve greater than 50% accuracy at win prediction. If Riot's matchmaking system is indeed optimized well enough that it can match the

[4] "Predicting League of Legends Match Outcome with ... - Medium." 24 May. 2018, https://medium.com/@yuan_tian/predicting-league-of-legends-match-outcome-with-embeddings-and-deep-learning-b7d1446c710f. Accessed 2 Oct. 2019.

[5] "minihat/LoL-Match-Prediction: Win probability ... - GitHub." https://github.com/minihat/LoL-Match-Prediction. Accessed 2 Oct. 2019.

performance of a large neural network on win prediction, then he expected the neural network will not achieve much greater than 50% accuracy at win prediction.

VI. Methods

A. Data Cleaning and Feature Engineering

- *Filling missing values*

First, we delete some rows with more than 30% missing values. This method is suitable only for dataset with enough rows, and we decide to perform it as we have 96 features and more than 40,000 rows. For those numerical rows with fewer missing values, replace them with median or mean.

- *Feature engineering*

Current dataset for the research contains 96 features including almost all raw counts from different aspects of the match, including total kills, total assists, earned gold per min, first blood time and etc. There are columns that are meaningless for the prediction process, like url, split, week, date, game, patchno, champion and etc. After deleting unwanted features, one-hot encoding is implemented to processing categorical features like positions in the match to convert them into numerical values, as most machine learning methods cannot deal with categorical data.

Considering all raw counts in the dataset, the feature engineering step produces more high-level professional parameters to analyze the game. For the in-game knowledge, the research extracts and calculates from each match statistics like total damage spread within the team, differences in gold among all lanes, gold spread within a team through mean and standard deviations, kills to deaths ratio, first kill ratio from historical matches and other similar statistics. After applying statistical calculation on the raw data, we generated 3 main features to analyze player performance.

Player score: Average score from the player's historical matches

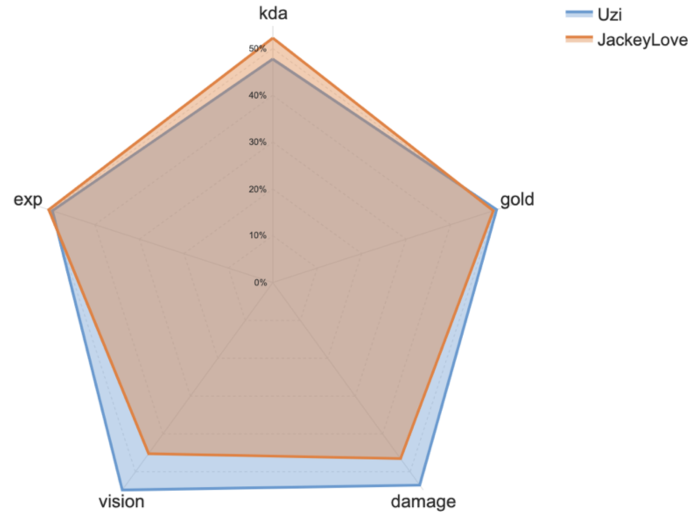
Champion score: Average score from the same champion's historical matches

Champion score by player: Average score from all player's historical matches using the specific champion (If the player has never use current champion before, then the feature takes average champion score - 1)

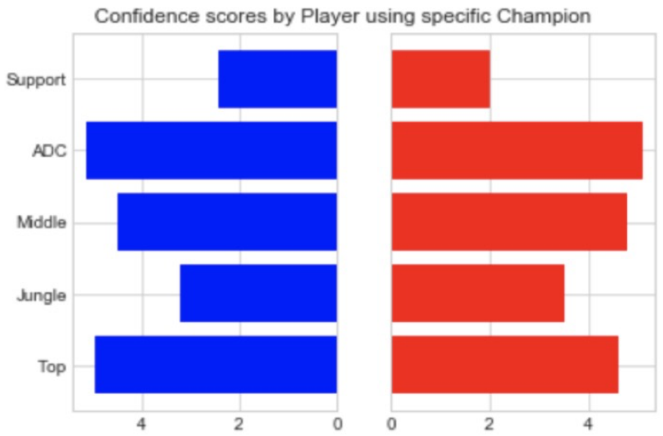
$$Score = \frac{(minion\_kills + monster\_kills) * 0.5 + kill * 3 + assist * 2 - death * 4 + wards * 2)}{game\_length} + \frac{dmg\_to\_champs}{total\_gold}$$

B. Feature Importances Visualizations

- Radar Chart: It is used to show the performance for a single player, depicting which component of the match each player show strength in.



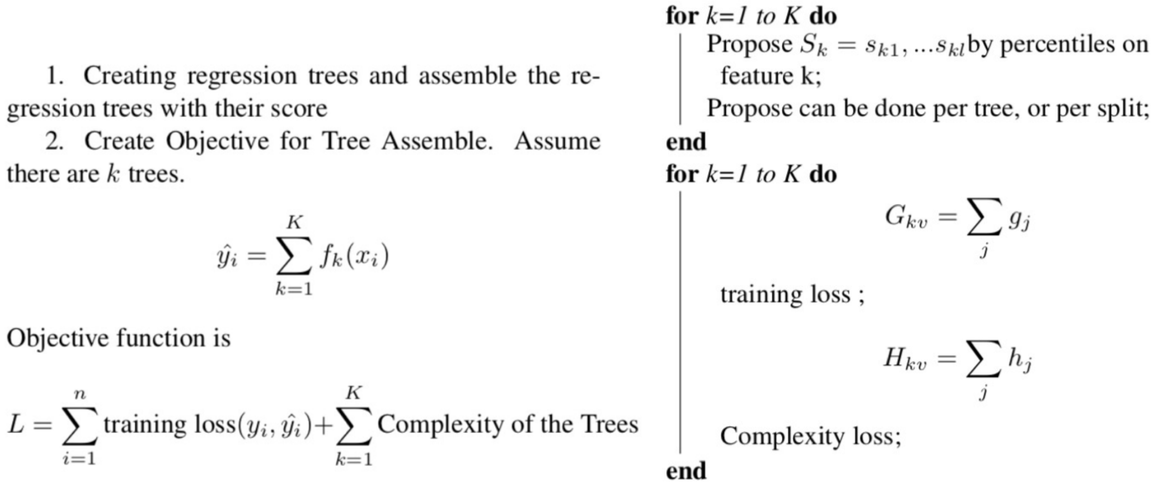
- Bar chart: It is used to compare historical performance of all players by teams and positions.



C. Model Construction

Since the match outcome is labeled (win=1, lose=0), we decided to use supervised machine learning model to predict labels. We applied K-fold Cross Validation for the dataset to make the outcome more precise. Although we have about 100,000 rows in total, we only have 20,000 data for teams. Thus, using K-fold help on tuning and model construction. We split the data into 80% training set and 20% test set to validate the model. For the method, we chose two machine learning methods, Gradient Boosting Trees and simple Artificial Neural Network.

Model 1: Gradient Boosting Trees. The model was appropriate for the data format we have since tree works well for classification and boosting is good to deal with very high dimensional data as well as the large amount of data we have for training set.[6]



1. Creating regression trees and assemble the regression trees with their score
2. Create Objective for Tree Assemble. Assume there are k trees.

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i)$$

Objective function is

$$L = \sum_{i=1}^{n} \text{training loss}(y_i, \hat{y}_i) + \sum_{k=1}^{K} \text{Complexity of the Trees}$$

```
for k=1 to K do
    Propose $S_k = s_{k1}, ...s_{kl}$by percentiles on feature k;
    Propose can be done per tree, or per split;
end
for k=1 to K do
    $G_{kv} = \sum_j g_j$
    training loss ;
    $H_{kv} = \sum_j h_j$
    Complexity loss;
end
```

Figure: Algorithm Demonstration for Gradient Boosting Trees

Model 2: Artificial Neural Network Model. It was constructed by adding three hidden layers with ReLU activation, hidden nodes and sigmoid activation for output. We evaluated the model performance by examining binary cross entropy loss function.

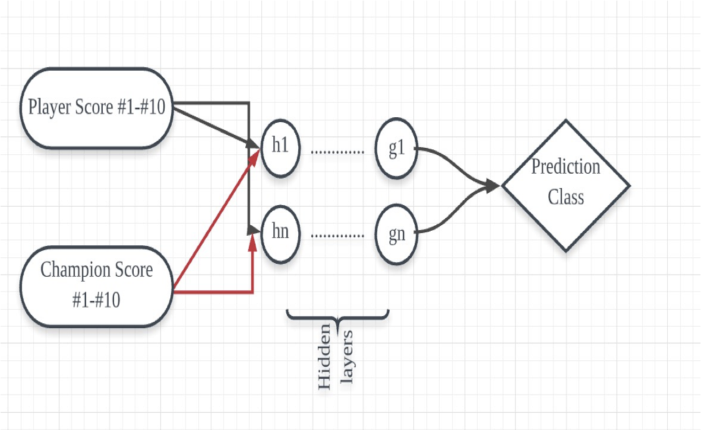
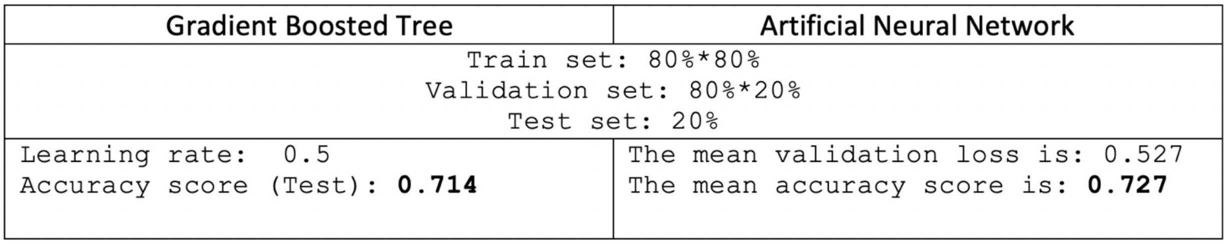


Figure: ANN Graph for Model

VII. Result and Future Improvement

A. Result comparison

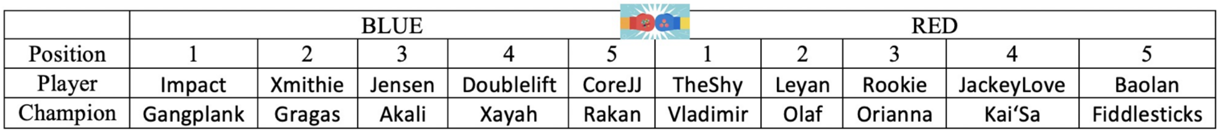
[6] "League of Legends Match Outcome Prediction - CS229 - Stanford ...." http://cs229.stanford.edu/proj2016/report/Lin-LeagueOfLegendsMatchOutcomePrediction-report.pdf. Accessed 2 Oct. 2019.



Gradient Boosted Tree	Artificial Neural Network
Train set: 80%*80% Validation set: 80%*20% Test set: 20%	
Learning rate: 0.5 Accuracy score (Test): **0.714**	The mean validation loss is: 0.527 The mean accuracy score is: **0.727**

Example Demonstration:

Here is the data for two teams of one match, we put the data into the ANN model we constructed to predict the winning probability.



	BLUE					RED				
Position	1	2	3	4	5	1	2	3	4	5
Player	Impact	Xmithie	Jensen	Doublelift	CoreJJ	TheShy	Leyan	Rookie	JackeyLove	Baolan
Champion	Gangplank	Gragas	Akali	Xayah	Rakan	Vladimir	Olaf	Orianna	Kai'Sa	Fiddlesticks

Result:

The win rate of Blue Team is: 0.457.

B. Future Development

This project demonstrates that using pre-match data, it is possible to predict the outcome of a League of Legends match with varying levels of success. We find that using the artificial neural network gave out a better performance than gradient boosted trees, with around a 72% success rate on prediction.

In the future, it would be interesting to develop a finer sense of how accurate the prediction system becomes given the amount of data at a certain point in a match. In other words, how much more accurate would the prediction becomes if data extracted from the first X minutes of each match is used as part of the feature vector as compared to the first Y minutes as well as how the accuracy changes with respect to X.

Reference

1. "Bet on League of Legends (LoL) ᐅ Esports betting. https://esportbetting.eu/games/league-of-legends. Accessed 2 Oct. 2019.
2. "The Rise of eSports Betting & Gambling - Business Insider." 9 Jan. 2018, https://www.businessinsider.com/the-rise-of-esports-betting-and-gambling-2018-1. Accessed 2 Oct. 2019.
3. "Predicting League of Legends Match Outcome with ... - Medium." 24 May. 2018, https://medium.com/@yuan_tian/predicting-league-of-legends-match-outcome-with-embeddings-and-deep-learning-b7d1446c710f. Accessed 2 Oct. 2019.
4. "Predicting League of Legends Match Outcome with ... - Medium." 24 May. 2018, https://medium.com/@yuan_tian/predicting-league-of-legends-match-outcome-with-embeddings-and-deep-learning-b7d1446c710f. Accessed 2 Oct. 2019.
5. "minihat/LoL-Match-Prediction: Win probability ... - GitHub." https://github.com/minihat/LoL-Match-Prediction. Accessed 2 Oct. 2019.