Data Analysis Report of League of Legends Ranked Games

Jingyao Yang
Jingyao.yang@marquette.edu

Abstract

With the rapid development of data science, many methods of data analysis, data visualization and machine learning have been widely introduced in the area of competitive videogames. Targeted analysis of video game data may enable teams to better prepare for the game, including selection of heroes, choice of game strategies, and so on. The main storyline of this report is the online multiplayer fighting arena game "League of Legends", where we conducted a series of analyses using 50,000 match data from open sources from the Kaggle platform. The main components of the report are as follows: I. Data pre-processing methods, including reading, extracting and converting various data source files, mapping fields and extracting game data. II. Data visualization methods, including the selection of different figure forms for different types of data indicators to show the best visualization effect. III. Data prediction methods, including the use of 50,000 historical data to train classifiers for predicting win rates and to analyze which features have a particularly significant impact on match wins and losses.

1 Introduction

With the popularity of competitive online games and e-sports, there is also a need to better understand the performance of games and gamers. Due to the rapid development of data science, numerous methods of data analysis, data visualisation and machine learning have been widely introduced into the competitive space of video games. As a result, researchers began to develop data analysis and visualization tools to generate statistics and highlight interesting facts to gain a more in-depth understanding of the game quotedrachen2013game. A successful example is "Echo", a visual analysis tool in the online multiplayer game "Dota 2" [1], a tool that collects statistics during gameplay and displays information at the top of the game to enhance the viewing experience of the audience. Another similar tool is "Scelight", developed for StarCraft II, which provides gamers with graphics and stats on the game, such as actions performed per minute, duration of play, speed and specific information about other players [2].

The main study scenario for this report is the online multiplayer combat arena game "League of Legends", which is similar to "DotA 2" in many ways. LOL is a highly multiplayer online game, but with a larger player base and numerous professional tournament events throughout the year. The game also has increased complexity with 134 characters to choose from, each with different abilities, powers and weaknesses, over 400 rune tree paths to choose from, and over 250 items to help each character in the game. In addition to the objective complexity, the decisions of each player in a game can lead to entirely different results: all players have different play styles and behaviors. As a result,

Submitted to 34th Conference on Neural Information Processing Systems (NeurIPS 2020). Do not distribute.

Figure 1: Summoner's Rift Map

the game is very noisy and any extraction of statistics needs to take this into account and find a way to deal with the huge differences that exist in the data. However, this complexity and diversity also makes it a very interesting problem to analyze.

The components of this report include: Section 2 presents a basic overview of LOL, Section 3 presents some related research work and algorithmic models, Section 4 presents data pre-processing methods, Section 5 presents data visualization methods, Section 6 presents data prediction methods, and Section 7 presents the results of the experimental analysis.

2 League of Legends Background

"League of Legends" [3] is a multiplayer online arena game (MOBA) in which two teams of 3 or 5 players face off against each other, the data studied in this paper focus only on the 5-player mode. Each player chooses a character (champion) from the 134 currently available, and each character has four different abilities. Players can further customize their champions by spending *gold* on items in-game. Each champion can have up to 6 inventory slots, with an additional 7th inventory slot for visual items. All items have an effect on various statistics of the character: movement speed, attack speed, attack range, physical damage, magic damage, armor, magic resistance, critical hit chance and health, with a few exceptions. There are several items that increase the amount of gold earned in certain situations, as well as vision items that can be placed on the ground in order to remove the fog of war from the area (only for teams that have vision items). Different types of vision items exist that we will not discuss in detail, as this is only one area of future work and is not discussed further in the current project. Each champion will also increase their stats as they level up (by gaining enough experience or XP).

The 5-on-5 game is played on a square map called "Summoner's Rift". Each team has a base in the top right corner ("red team") or bottom left corner ("blue team") that includes a large laser turret that spawns in each team's corner, a nexus (once destroyed, the owning team loses the game), 2 turrets to

protect the nexus, and 3 inhibitors at the top, bottom, and middle of the map (which do not attack the opponent and reappear after 5 minutes when destroyed). From each inhibitor to the corresponding inhibitor on the other side of the map, 3 lanes are formed (named "top", "middle" or "middle" and "bottom" or "robot"). Each lane has 3 turrets of each team to protect the blockers. These turrets must be destroyed in turn from the furthest away (otherwise they are immune to damage). Between the lanes is a "jungle" which is covered by the fog of war (players cannot see what is there unless another player in their party is in the area) and contains several neutral monsters (they will only attack if attacked first), as well as two epic monster camps (Rift Herald / Baron after 30 minutes of game time; and Dragon). Epic level monsters have increased life and deal high damage, requiring a high level champion and a group of champions to defeat them. Figure 1 shows a visualization of this.

3 Related Work

There have been several previous studies on victory prediction in multiplayer online games. Yang et al. studied combat patterns to predict the outcome of matches in "DotA 2" [6]. Hodge et al. showed that analysis of datasets with mixed levels (including players of different skill levels, or ranks) could be successfully used to predict the outcome of "DotA 2" professional competitive matches; although the accuracy was somewhat decreased, but this raises the possibility of obtaining more data and still achieving positive results [4]. Kim et al. conducted an interesting analysis of team performance in "League of Legends" using the concept of collective intelligence (CI) [5]. Their hypothesis was based on the idea that teams that work better together have a better chance of winning the game, regardless of individual skills. Their findings suggest that it is indeed CI that leads to higher win rates in the later stages of the game when players are not separated but are expected to cooperate more, and also correlates with the social perception of individual team members.

4 Preprocessing Data

This section will introduce our pre-processing process for the game data. In the first part, we will introduce the basic information and characteristics of the LOL game data provided by Kaggle. In the second section, we will introduce our read and load operations on the data files. In the third part, we will introduce the information of the main fields of the game data and some basic assumptions about these fields. In the fourth part, we will introduce how to calculate and generate some procedural data structures for advanced data analysis.

4.1 Description of LOL game data

There are three League of Legends data files provided by Kaggle: `champion_info_2.json`, `summoner_spell_info.json`, and `games.csv`. `games.csv` file has `51,490` rows and `61` columns, representing `51,490` matches and `61` columns of feature information, respectively. The `champion_info_2.json` file is a JSON file that records detailed information about each champion. The `summoner_spell_info.json` is another JSON file that records detailed information about each skill. The `games.csv` file only has the champion id and skill id, so it needs to work with the other two files.

4.2 Loading and extracting data

In this part, we use `Pandas` to read and load data files. First, we use Pandas' function `read_csv()` to read `games.csv`, and then we use `read_json()` to read the other two files. We have designed two mapping functions in order to convert the id represented by an integer into a real name represented by a string, so as to improve the convenience of readers.

gameId	champId	win
3326086514	Vladimir	True
3326086514	Bard	True
...	...	...
3317333020	Renekton	False

Table 1: Champion Record DataFrame

4.3 Description of Importance Columns

Although the game file has a total of 61 features, after analyzing the characteristics of the LOL game, we can divide these 61 features into several parts. The first part is the characteristics of the heroes of both teams, there are 10 in total, which can be simply expressed as `t[1,2]_champ[1-5]id`; The second part is the characteristics of the skills of both teams, there are 20 in total, which cane be expressed as `t[1-2]_champ[1-5]_sum[1-2]`; The third part is the hero's Ban information, simply expressed as `t[1-2]_ban[1-5]`; The fourth part is the first kill statistics for important elements, simply expressed as `firstXXX`; The fifth part is the offensive situation of each team on the key elements such as defensive towers. Other features include the duration of the game.

We can find that some of the data is category data, and some of the data is numerical data, so different methods need to be used when processing, for example, categories need to be onehot encoded, etc.

We can assume that the heroes between the same team have no significant influence on the order relationship, that is, the heroes between the same team can be regarded as a collection instead of other ordered queues. Under this hypothesis, we can arrange and combine the heroes of the same team more flexibly, thereby obtaining a more complex procedural data structure.

4.4 Advanced Data Structures

In order to facilitate the calculation of the number of appearances of each hero, the winning rate, and the cooperation relationship between each hero, we need to calculate and generate some procedural data structures for advanced data analysis on the basis of raw data. According to the above assumption, the five heroes in each team can be regarded as equally important, so we split one row of data into ten rows, and each row records the information of a champion, as well as the corresponding game ID and game result. See table 1.

5 Visualizing Data and Case Study

This section visualizes several important statistical indicators. The most important ones are the number of pick of the champion, the number of hero banned, the hero's winning rate, and so on.

Figure 2 shows the three heroes selected the most and banned the most. The top three heroes picked by players are Thresh, Tristana and Vayne. According to the situation of LOL games, we analyzed the main reasons why these three champions are picked frequently. Thresh can be adapted to a variety of game scenarios, from giving opponents a very high damage early in a match to protecting other teammates early in a match. Thresh's versatility made it a favorite choice. Tristana is a very strong champion early in the match, often quickly killing the champion she meets, so many players who like to control the initiative early in the match like to choose her. Vayne is a champion who takes control of the game late in the game, and as long as she keeps accumulating her equipment and skills, Vayne is virtually untouchable late in the game.

The three most banned champion are Yasuo, Zed and Cho Gath. Yasuo is a champian who requires a lot of skill from the player. Players with good skills and those with poor skills will behave differently when using Yasuo, which leads to a huge fluctuation in Yasuo's performance in the game. Not sure if the enemy is a highly skilled player, and not sure if the teammate automatically matched by the game system is a less skilled player, players often disable the hero to get a better experience. Zed was a

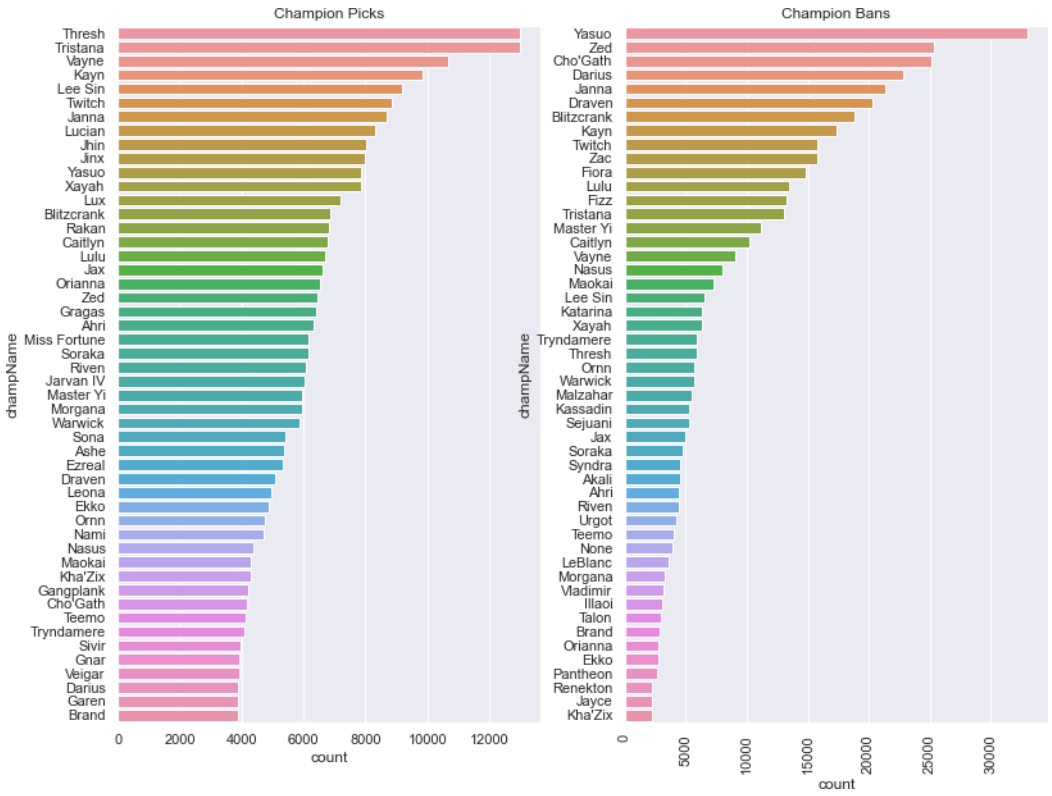


Figure 2: Freq Count of Champions (Pick vs Ban)

hero who didn't need any skill early on, but he was able to easily kill vulnerable heroes, which led to many players who preferred low health and high damage heroes to disable Zed. Cho Gath is a hero who is very defensive, has a lot of health, and has a lot of control skills. He doesn't require any skill, and many players want to disable him to improve the experience.

6 Modelling and Predicting

In this section, we will use some machine learning models to predict and analyze the winning percentage of LOL games. We used five classification models, namely SVM, KNN, Random Forest, Logistics Regression and Ridge Classifier.

6.1 Encoding

Before modeling the classification model, some basic data preparation and feature engineering are required. This is because the performance of the model depends on the feature engineering strategy. The original data set contains features of various data types, and various feature engineering techniques are required to convert data features of different data types into digital vectors.

There are many coding strategies to convert classification features into digital vector form including Count Encoder, One Hot Encoder, TF-IDF Encoder, etc. The `pd.get_dummies()` of the `Pandas` toolkit can perform the above coding in one line of code. Although many people use it for categorical feature coding, we do not recommend using it for Kaggle data competitions after we have experimented. This is because the `get_dummies()` in the Pandas library is a "static" behavioral technique, that is, the function does not learn the characteristics of the training data. If a feature appears in the training set but does not appear in the test set, the result of the function encoding will further lead to feature mismatch during prediction.

Algorithm	f1	accuracy	precision
SVM	0.968278	0.967684	0.958919
KNN	0.959736	0.959295	0.957675
Random Forest	0.964256	0.963723	0.958460
Logistic Regression	0.957176	0.956653	0.953954
Ridge Classifier	0.955036	0.954401	0.950015

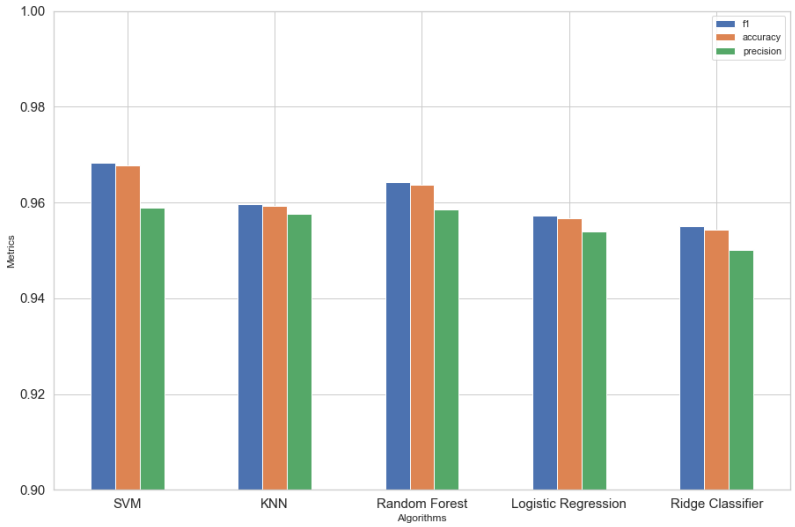


Figure 3: Performance of Predicting Algorithms

In order to solve this problem, we used One Hot Encoder in the scikit-learn toolkit in this step, which is a popular feature encoding strategy. The effect of this function is similar to `pd.get_dummies`, but it has a learning function. It encodes the classification features by assigning a binary column to each category of each classification feature, and the encoding features are saved when the classification variables of the training data are encoded.

6.2 Metric

In order to measure the effects of different models, this report chooses three metrics: Accuracy, Precision, and F1 to measure, and also draws the ROC curve corresponding to each algorithm and its corresponding AUC value. Experimental results show that SVM has the best effect, Random Forest ranks second, and the performance of the other three algorithms is similar.

7 Conclusion

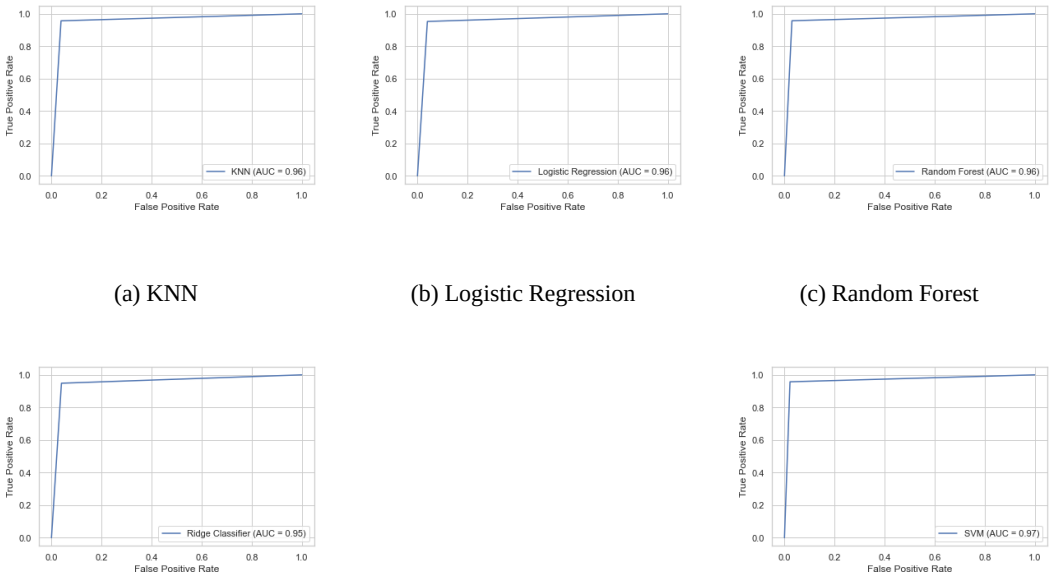
This report conducted a series of LOL game analysis using 50,000 pieces of open-source game data on the Kaggle platform. This article introduces in detail data preprocessing methods, data visualization results, and data prediction methods based on different machine learning models.

References

[1] Raluca Gaina and Charlotte Nordmoen. League of legends: A study of early game impact. 2018.

[2] Raluca D Gaina, Simon M Lucas, and Diego Pérez-Liébana. Vertigø: Visualisation of rolling horizon evolutionary algorithms in gvgai. In *Fourteenth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2018.

[3] Riot Games. League of legends [computer software]. *Los Angeles, Ca: Riot Games*, 2009.



(a) KNN (b) Logistic Regression (c) Random Forest

(d) Ridge Classifier (e) SVM

Figure 4: ROC and AUC of Predicting Algorithms

[4] Victoria Hodge, Sam Devlin, Nick Sephton, Florian Block, Anders Drachen, and Peter Cowling. Win prediction in esports: Mixed-rank match prediction in multi-player online battle arena games. *arXiv preprint arXiv:1711.06498*, 2017.

[5] Young Ji Kim, David Engel, Anita Williams Woolley, Jeffrey Yu-Ting Lin, Naomi McArthur, and Thomas W Malone. What makes a strong team? using collective intelligence to predict team performance in league of legends. In *Proceedings of the 2017 ACM conference on computer supported cooperative work and social computing*, pages 2316–2329, 2017.

[6] Pu Yang, Brent E Harrison, and David L Roberts. Identifying patterns in combat that are predictive of success in moba games. In *FDG*, 2014.