Esports Prediction of the Professional League of Legends Scene

周廷紘 Ting-Hong CHOU
108060002
Undergraduate of the Interdisciplinary Program of Electrical Engineering and Computer Science
jonathanchou99@gmail.com

許至誠 Chih-Cheng, HSU
109060015
Undergraduate of the Interdisciplinary Program of Electrical Engineering and Computer Science
waderhsu@gmail.com

賴彥呈 Yen-Cheng, LAI
109060031
Undergraduate of the Interdisciplinary Program of Electrical Engineering and Computer Science
ethanlai0928@gmail.com

陳璽正 Hsi-Cheng CHEN
109060032
Undergraduate of the Interdisciplinary Program of Electrical Engineering and Computer Science
peteraaa5@gmail.com

温柏崴 Bo-Wei, WEN
109060035
Undergraduate of the Interdisciplinary Program of Electrical Engineering and Computer Science
forhchs@gmail.com

***Abstract*—This report presents the implementations of different models used to predict Esports prediction using public League of Legends matches as examples. The data was retrieved from the website "Oracle's Elixir" which contains data from world tournaments. The models we used include linear regression, support vector classification (SVC), K-nearest neighbors (KNN), and decision trees with different modifiers. The data is split into a different dataset using 2022 data with improving practicality in mind. Further testing on data from the years 2015 to 2019 was also done using transfer learning from the model of 2022. The result showed that less information heavily affects the accuracy, the factors of importance do correlate with common sense, and transfer learning is feasible on the data.**

Keywords—Esports, classifiers, decision tree

I. Introduction

Esports is a type of sport where the game is conducted over electronic devices, which is mostly on mobile phones and computers nowadays. Similar to other sports, analyzing every game to determine the reason for either winning or losing the game is important for increasing the winning rate of a team. Additionally, the world also saw a thriving betting scene for the sport over the years. Therefore, we implemented several different machine learning approaches to predict the result, the winner of the game, and analyzed the major factors that affect the game result.

League of Legends is a team-based strategy game where two teams of five powerful champions face off to destroy the other's base. Each player can choose from over 140 champions, along with their teammates, to make plays, secure kills, and take down towers as they endeavor for victory [1]. There are several reasons behind choosing League of Legends to do research on among hundreds of competitive video games. Firstly, it is a game where the winning condition is simple, a team will control five characters and destroy towers until eventually the nexus of the opponent team to win the game. Since the nexus of two teams won't be destroyed simultaneously by game design, the result for a game is always either winning or losing, or a zero-sum game, which makes analyzing the game less complex. Secondly, several world championship of League of Legends has been held annually since 2011, which ensures the amount of match data for training the models. Last but not least, League of Legends has a lot of aspects to analyze whether a player is making good plays or not, such as kill count, amount of resources, etc.

For the method of obtaining data from the game matches, manually recording the statistics directly from the replay video is not efficient and also error-prone. Unfortunately, Riot, the officials of League of Legends, chose not to release the data of the game matches. So the source of the data is instead a third-party website, Oracle's Elixir.

II. Methods

A. Data Description

According to the official website hosting Oracle's Elixir, they are the premier source for advanced League of Legends esports stats [2]. The data for each year consists of numerous features, including results and statistics for players and the teams of the match. Also, values in most of those stats are calculated for each game, then averaged across games. Since the game received changes over the years, we stuck with data from the year 2022 and only consider using other data during transfer learning analysis.

The data directly from the source is not suited for training [3]. As our model focuses on predicting the results of the game, each entry of the training data should also represent data collected from a specific game. Notwithstanding, the source organized the data of a game by first recording statistics of each player in the game followed by totaling each feature for the two teams. In other words, it takes 12 rows to fully represent a game. With different intentions in mind, we split the data with different approaches, making three different datasets.

Due to the format constraint of this report and the file size constraint of GitHub, the data with 247 labels cannot be represented in any way on the report. Alternatively, it will be uploaded to a google drive for the report to refer to. For a clearer representation of the labels, please go to https://drive.google.com/drive/folders/1kDdjlNJ03mEezfscHwcmdbNGvgEINYer

1) Dataset 1: Almost every feature

At first, we wanted to make sure there is some relation between the result and some stats, moreover, which features affect the most. Therefore, we kept most of the features from the source, only deleting incomplete data and information unrelated to the game, such as URL and date. In addition, we

combined the data of 5 players into an overall team format, changed picked and banned champions to one-hot data, and organized it such that each row represented an independent game.

The features are spitted into team1, team2, and shared used in this dataset. You may find the data on the google drive we provided in the folder "Dataset 1". The meaning of each label can be found at Oracle's Elixir, but this report will only explain those referenced [4].

2) Dataset 2: Early game related

However, there was little to no real-world application for the first dataset, as the data could only be recorded postgame, meaning in most cases, the data would appear complementary to the result. Therefore, this dataset focused on "Early Game statistics", or the data recorded at the first 10-to-15-minute mark. The result could help teams analyze the situation and possibly improve their strategies for the game.

Hence, we derive the dataset from Dataset 1, choosing the features related to the early game.

The features are spitted into team1, team2, and shared used in this dataset. You may find the data on the google drive we provided in the folder "Dataset 2".

3) Dataset 3: Pregame data with statistics per game

Last, we took it one step further and attempted at building a model that predicts before the game starts. Thus, the labels from the data source must be further modified to provide information from previous games. By manually sorting through each game and recording the averages up to that point, we generated new labels representing the average of past records from both teams not including the current game.

The features are spitted into team1, team2, and shared used in this dataset. You may find the data on the google drive we provided in the folder "Dataset 3".

B. Classification Methods

1) Voting Model

The selection of models started by reviewing related studies regarding this field. After a thorough reading of some papers in similar categories, we discovered that all of them used either neural networks or decision tree classifiers with different boosting methods [5] [6]. In order to practically compare the performances and see why they abandoned other classification methods, we selected some simpler schemes such as linear regression, support vector classification (SVC), decision tree, and something increasing in complexity, K nearest neighbor (KNN), to get prediction results with simple implementations. However, to make up for the less accurate result, we opted to use a pipeline to load all three models and produce the output by voting through different weights.

2) Decision Tree with Bagging

To address the possible negative impact of the high dimensionality observed in our input on methods such as the linear regression and the KNN classifier, we tried to improve the decision tree classifier, which is relatively less affected by aspects other than the dimensions, by performing bagging on the decision tree [7]. We will discuss the extent of improvement in the result section.

3) Decision Tree further modified with Boosting

Considering bagging randomly selects the parameters of the decision tree, we altered bagging with AdaBoost to ensure the improvement after each sampling based on the last implementation instead of random selection.

4) Transfer Learning of Data from Previous Years

We could use a particular year as training data to perform predictions of other years and observe the similarity of data by time. By comparing all the models we mentioned, we could single out the model with the better performance overall, and apply it to data from different years. Since Dataset 1 showed different numbers of features over the years, we directed the practice of transfer learning on only Dataset 2 and 3.

C. Feature Explainability

We could make the feature graph by getting all the feature importance and listing the top 10. With this information, we could determine whether there exist one or more features dominating the correlation of the data. We also wanted to figure out why these features were more decisive by considering their meaning throughout the game, further explaining how the model works and what it considers the most.

III. Results

The following results are accessible at:
https://github.com/CriticSuika/Machine-Learning-Final

A. Classification with Dataset 1

After fitting all the models through the different methods, we implemented with Dataset 1, we obtained the result shown in Tab. 1.

Tab. 1 shows that all the models showed similar accuracy across the board. Viewing voting as one model instead of three, the accuracy falls at 99.2% for all models.

We also ran a feature importance analysis and obtained Fig. 1 as the outcome. At the top of Fig. 1, we can see that the towers, followed by the inhibitors and the earned gold are the most important factors in the model.

B. Classification with Dataset 2

We fitted the models with Dataset 2 and acquired Tab. 2 as the result. Tab. 2 shows that all the models have accuracy at about 85%, with the exception of the KNN classifier performing at 68.8% which is further corrected by voting.

Running the model through importance analysis yields Fig. 2. As shown in the figure, Killing the baron before the other team is the most essential label in the dataset, followed by gold and XP differences 15 minutes into the game.

Last but not least, we attempt transfer analysis of 2015 through 2019 using the model trained for 2022 and the results are shown in Tab. 3. The evidence shows a similar if not better result compared to with the data from 2022.

C. Classification with Dataset 3

Fitting the data from Dataset 3 into the models, we got the result shown in Tab. 4. The accuracy dropped significantly compared to the results for Datasets 1 and 2, at about 62%.

The model's feature importance analysis is represented in Fig. 3, which shows that the most crucial factors are towers, total games, and win rates.

Similarly, we could place data from different years into the model for 2022 and obtain the result in Tab. 5. It shows similar results of about 62%, with the exception of the decision tree with bagging at an exceptional 92% accuracy.

IV. Discussion and Conclusion

In this report, we present an attempt on predicting the result of a League of Legends game match by different models and different partitions of the dataset. From the results, we can observe that using different datasets had a great impact on the accuracy, and thereby has significant implications behind it.

The accuracy of the models using Dataset 1 is extremely high, the accuracy of all the models except KNN reached 99%. And the reason is also indicated in Fig. 1, where we knew that the number of destroyed towers is the most important feature of the models. This is quite reasonable since the main objective of winning the game is to destroy towers and the nexus of the opponent team, and a specific requirement of destroyed towers must be met until the nexus is not invulnerable. However, the models using Dataset 1 are inconsequential in practical scenarios mainly due to two reasons. Firstly, the only information we gained from these models is that a team needs to destroy as many towers as they can to win a game. Secondly, these statistics are only obtainable by the end of the match, and as previously mentioned, under no circumstances will they appear without the results of the game. These are also the reasons why we tried to use different partitions of the whole dataset.

In Dataset 2, instead of using postgame statistics, we used the statistics 10 and 15 minutes after the game starts. As a result, the accuracy is around 90% on average, and the accuracy of KNN dropped about 20% in particular. This outcome is extremely reasonable, as we removed a huge portion of data significant to the previous result. Additionally, due to the differences, the information in Fig. 2 is totally different from the models using Dataset 1. This indicates that the most important factor of the game result in the early game is whether a team kills the first Baron of a game. In the game, Baron is a strong neutral opponent that if a team defeats it, they get some gold and their minions are strengthened. Despite the obvious advantage, it is not without its risks such as being killed by the opponent during the fight.

Finally, Dataset 3 includes only the statistics that can be obtained before the game starts, such as the average statistics of the history matches of a certain team. The accuracy of the models is around 60% on average. The most important feature of models using Dataset 3 is the average number of destroyed towers of both teams in their history matches. An important note is that the towers represent the team wins better than the win rates, as the percentages may be extremely biased when teams have only played a few games. For instance, winning a game out of one total game constitutes a 100% win rate. The low accuracy also indicates that the performance of a team in the past cannot fully represent the game performance of their next games. In real life, factors such as player morale, adrenaline levels, home court advantages, etc. all contribute to a result that is completely irrelevant to their records.

We tried three different datasets and gained models that can be applied to different circumstances. For instance, models using Dataset 2 are good for analyzing the gameplay advantages at different moments, and models using Dataset 3 are good for betting on the winning teams of a game. In addition, since the game has changed a lot over years, it might not be wise to include older data in the datasets. However, by conducting transfer learning, we could analyze the impact of changes to the game after every update instance. In conclusion, selecting different data can change the accuracy and the application of the models.

Author Contribution

108060002 Ting-Hong CHOU 周廷紘 (20%):
Study Design, Implementation, Result Analysis, Report
109060015 Chih-Cheng, HSU 許至誠 (20%):
Implementation, Result Analysis, Report
109060031 Yen-Cheng, LAI 賴彥呈 (20%):
Result Analysis, Presentation, Report
109060032 Hsi-Cheng CHEN 陳璽正 (20%):
Implementation, Result Analysis, Report
109060035 Bo-Wei, WEN 温柏崴 (20%):
Data Preprocessing, Result Analysis, Presentation, Report

References

[1] Riot Games, “League of Legends,” Riot Games, Available: https://www.leagueoflegends.com/.

[2] Oracle’s Elixir, “League of Legends Esports Stats – Oracle’s Elixir,” Oracle’s Elixir, Available: https://oracleselixir.com/.

[3] Tim Sevenhuysen, “OE Public Match Data,” Oracle’s Elixir, Available: https://drive.google.com/drive/folders/1gLSw0RLjBbtaNy0dgnGQDAZOHIgCe-HH.

[4] Oracle’s Elixir, “Oracle’s Elixir – Definitions,” Oracle’s Elixir, Available: https://oracleselixir.com/definitions.

[5] V. J. Hodge, S. Devlin, N. Sephton, F. Block, P. I. Cowling, and A. Drachen, "Win Prediction in Multiplayer Esports: Live Professional Match Prediction," in IEEE Transactions on Games, vol. 13, no. 4, pp. 368-379, Dec. 2021, DOI: 10.1109/TG.2019.2948469.

[6] A. Katona, R. Spick, V. J. Hodge, S. Demediuk, F. Block, A. Drachen, and J. A. Walker, "Time to Die: Death Prediction in Dota 2 using Deep Learning," in 2019 IEEE Conference on Games (CoG), London, UK, 2019, pp. 1-8, DOI: 10.1109/CIG.2019.8847997.

[7] K. U. Birant, “Multi-view rank-based random forest: A new algorithm for prediction in eSports,” in Expert Systems, vol 39, no. 2, DOI: 10.1111/exsy.12857.