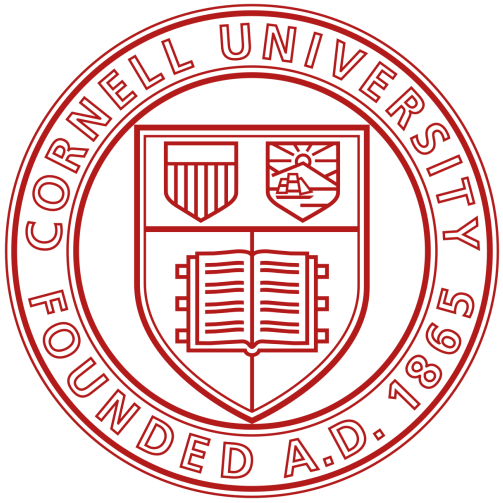


ORIE 4741

Project Final Report

Advice for players to improve game experience in a League of Legends match

Jin Cui (jc3362)
Yifan Zhu (yz2522)
Yuqi Hong (yh854)

December 2019

1. Introduction

From the league of legends official website. "*League of Legends* is a fast-paced, competitive online game that blends the speed and intensity of an RTS with RPG elements. Two teams of powerful champions, each with a unique design and playstyle, battle head-to-head across multiple battlefields and game modes. With an ever-expanding roster of champions, frequent updates and a thriving tournament scene, *League of Legends* offers endless replayability for players of every skill level." League of Legends was released in 2009 and has since grown in popularity. By 2012, it became the most played PC game in North America and Europe in terms of the number of hours played. Most of the players are students. Our analysis may solve the biggest problem that League of Legend players may concern: How to win the game? By using big messy data processing and analysis, players may know items with the highest win ratio and may also know how the trinket plays the essential role. There are more science behind shaping player behavior -- avoiding unreasonable group fights, the best places to place the wards and the order of building items. Player may win the game easily by playing scientifically and efficiently. In the end, this project will help players improve their win ratio and also save their time so they can go to study after the games as soon as possible.

2. Description of dataset

The data used for analysis is from Kaggle dataset, which provide different elements of each match of the game and also the result of the game. For example, the champions selected by players, different roles played by players, damage/heal made by champions, kills/deaths of each player, etc. To be specific, seven csv files are provided for analysis, they are:

(1) Teambans.csv: information about the team ban, which contains 'match_id', 'team_id', 'champion_id', 'ban_turn'
(2) Champs.csv: information about champions used by player, which contains 'name', 'id'
(3) Participants.csv: information about participants, which contains 'id', 'match_id', 'player', 'champion_id', 'ss1', 'ss2', 'role', 'position'
(4) Matches.csv: information about each match and game version, which contains 'id', 'game_id', 'platform_id', 'queue_id', 'season_id', 'duration', 'creation', 'version'
(5) Stats_1.csv + Stats_2.csv: which contains specific detailed information about each game match, for example, 'win', 'trinket', 'kills', 'deaths', 'assists', 'largest_killingspree', etc.
(6) Teammates.csv: information about teammates' performance, which contains 'first_blood', 'first_tower', 'first_inhib', 'first_baron', 'first_dragon', 'first_harry', etc.

To make full analysis of what elements may make a team win a match, we will try to implement as many as data we have acquired to make the prediction from different aspects. Some important histograms of data are shown on the next page.

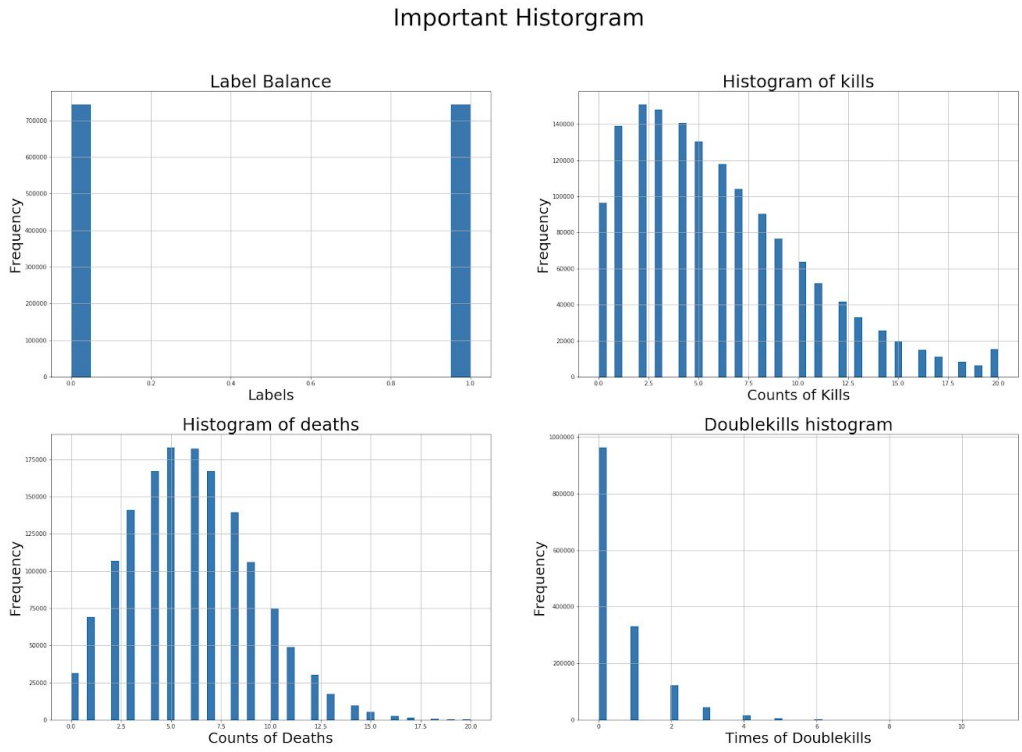


Figure 1 important histograms

3. Data preprocessing

(1) Raw data is collected in seven separate csv files, which represents different features of each match recorded. In order to train the model with all features, five tables are joined by user id and match id, and the other two tables are discarded because the recorded information is not for each match but contains general information, which will not be helpful to achieve our goal.

(2) Several string type features are transformed in order to make information consistent. For example, ‘Role’ and ‘Position’ are combined because ‘Role’ only contains information about ‘DUO_SUPPORT’ and ‘DUO_CARRY’, but other position information are contained in ‘Position’. These two features are combined as ‘adjposition’.

(3) Raw data is collected by team member, but every five players are in the same team, which is not shown in the raw data. For convenience, a new feature is added.

(4) Some features, for example, id type of features could be noisy for the model, because no information is contained. These kinds of features are removed.

(5) In order to remove outliers, limitations are set for some features, for example, kills cannot be too high or negative. Such kind of limitations are adjusted on deaths, assists, wards placed, etc..

(6) There are only 3 records are missing after the joint table. Since it relatively super small compared to the number of data, we just discards these three missing records. Also, repeated are checked and discarded.

4. Feature selection

After preprocessing, we have 1486362 examples with 76 features. These features contains both numeric, string and boolean data, while model only can be trained by numeric data. Thus, string type of data should be encoded. However, the dimensionality of data could explode if we encode every string type data. Also, numeric should also be selected to reduce the complexity.

(1) Only ‘role’ and ‘name’(champion) are used through all string data. The rest features, for example, item information will lead dimension explosion after encoded. Besides, the goal of the project will not rely on items, but role and champion will be quite important.

(2) Numeric data is mainly selected by correlation matrix.
A correlation matrix is a table showing correlation coefficients between variables. Each cell in the table shows the correlation between two variables. A correlation matrix is used to summarize data, as an input into a more advanced analysis, and as a diagnostic for advanced analyses. Correlation matrix of all numeric variables are shown below. It is quite a huge matrix.

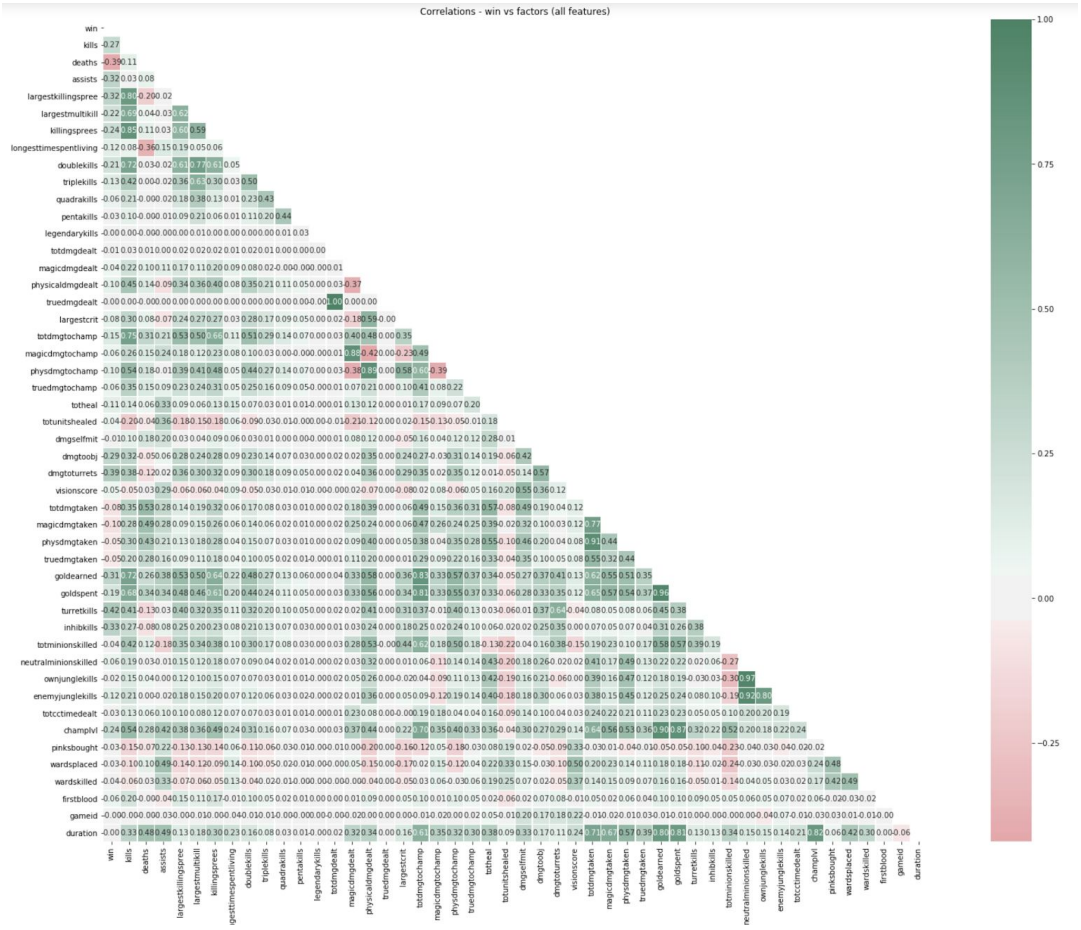


Figure 2 Whole correlation matrix for numeric data

The variables with higher correlation coefficients are picked for preliminary model built in midterm in order to predict whether the player will win the game or not. The correlation matrix is shown below.

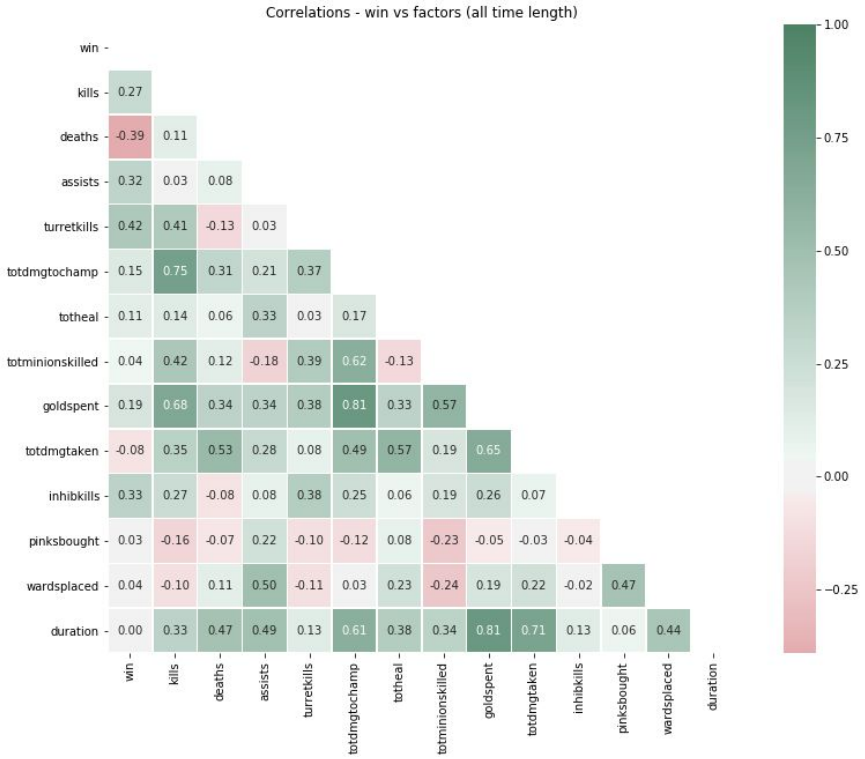


Figure 3 correlation matrix with fewer features

(3) After building the logistic regression model with few good features in figure 3, the result is both high speed(around 2 mins for training) and accurate(85% for both training and testing accuracy). In order to improve the accuracy, 20 best features(selected by the highest absolute value of correlation coefficients) plus the role and champion features are picked to train the logistic model. Although the training and testing accuracy can both achieve 89% near 90%, the time spent to train this single logistic regression model is around 6 minutes, which is much longer than the preliminary model.

(4) Tree models are also used for feature selection. Decision tree and random forest are used to train on the whole data, and features are selected by feature importance provided by sklearn(calculated by node gini impurity). However, correlation matrix is easier to understand and explain, so these strategy was tried but not used for training the model.

5. Algorithms used to solve the problem

(1) Logistic regression. This algorithm is both our preliminary model and final model. Although the goal of the project includes improving accuracy for predicting victory or defeat of the game, the main purpose is to dig out what elements may result in that, so algorithms and parameters tuning are not the most significant tasks. Besides, more than 14 million records with 140 features will cause the training process super low. A simple algorithm with high accuracy(logistic regression) is good enough for this project.
(2) Random forest.
(3) Decision tree.
(4) Support vector machine.

6. Results and how confident in results

(1) Victory prediction results(training/testing accuracy)
Logistic regression with fewer features: 0.8548/0.8547
Random forest with fewer features: 0.9760/0.8556
SVR with fewer features(without tuning): 0.6652/0.6574
Decision tree with fewer features: 0.8656/0.8341
Logistic regression with top features: 0.8993/0.8988
(2) Champion selection predicting victory: 0.545 accuracy
(3) Top 5 elements for winning a match: 'inhib kills', 'deaths', 'turret kills', 'pentakills', 'largest multikill'
(4) Top 5 elements for improving KDA: 'deaths', 'largest killing spree', 'killing sprees', 'assists', 'quadrakills'
(5) Top 2 elements for improving longer living time: 'team role', 'champion'
(6) For predicting the victory with match information, we are quite confident because how close the training and testing accuracy shows that our best model is not overfitting. Besides, the model itself can predict with a relatively high accuracy(around 90%) with quite few features.

For predicting the victory with champion information, we are not quite confident because the accuracy is a little bit higher than 50%, which means the results are a little bit better than randomness. However, it is still good to acquire information about the game right after the champion picking at the very beginning.

We are quite confident in top elements acquired for victory and longest time living because of the high accuracy. But for KDA elements, we are not super sure because the simple linear regression model can only yield 66 accuracy, so the confidence level of every coefficients could be high, which means they may not be so assured.

7. Production on real-world example

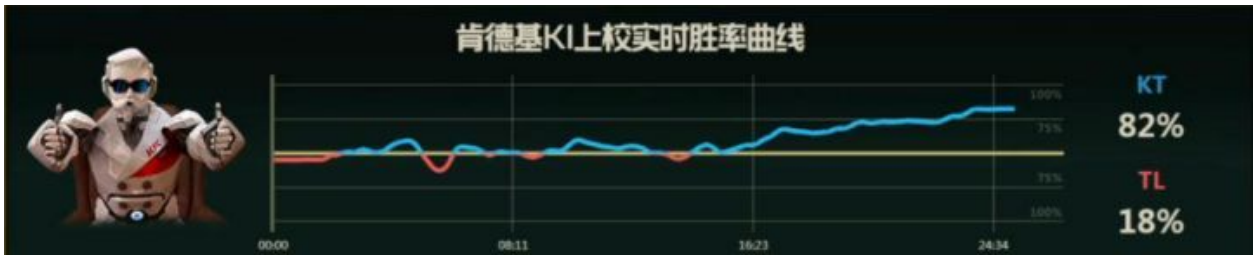
(1) One of the productions that we want to apply on real-world business is league of legends gambling game. There are always a lot gambling games for predicting the win or lose of League of Legends games, especially for world-wide championship. Therefore, it is meaningful for people who are interested in gambling and the analysis results can help people predict the game results more accurately.

The training data was built by reconstructing the original data, so each row is every match game, and there were 12 columns, 5 roles for each team and there were 2 teams each match, plus the platform ID and game season ID. We trained the model by using logistic regression, and the prediction accuracy is around 54.5%. We used the 2017 League of Legends world championship data as test dataset and apply the trained model to see whether the prediction results met the actual results. There were two teams called SKT and SSG, the prediction for SKT win rate is shown below:

```
Game 1 SKT win rate: 0.45234412013485226
Game 2 SKT win rate: 0.37559875300532575
Game 3 SKT win rate: 0.4796998416697271
```

Therefore, we would suggest people to predict SKT would lose and SSG would win for gambling. Actually, at 2017 world championship, SSG win 3 games in a row, which is exactly the same as we predicted, so we can say the prediction model can actually help people on gambling game.

(2) Real-time win rate prediction



Real-time win rate prediction has already been implemented by Chinese LOL match holder, which will be represented every few minutes in the match and make the decision based on the game data. This kind of model should be both fast to train and accurate. Although our model was trained only by the final data provided at the end of each match, yet if we were provided with such kind of data, it might be possible to produce a model very close to the production stated above. There might be some other elements to take into consideration, but it is a good direction to go and think about.

8. A list of all techniques used from class and outside class

(1) Data preprocessing
Since there were seven csv files in the original dataset, we need to concat similar data and join tables, then we also need to do grouping and aggregation to make several tables become single table and then it can be analysable for us.

(2) Data cleansing
We need to deal with missing data, incorrect data in the original dataset by using the methods we learned in class, such as directly dropping the missing data and filling missing data with mode.

(3) Data visualization
We plotted some histograms to help us better understand the data for each feature. We also used violin plot to analyze the data because it gives the quartile of the data, which makes the plot comparable for binary labels.

(4) Feature engineering
One hot encoding is also one of the techniques we learned in class and used in our project. Since our original data has 76 features, and would be much larger if apply one-hot-encoding method, it would significantly reduce the training speed. Therefore, we need to do feature selection first to select the top 20 most important features by using correlation matrix.

(5) Algorithms
We tried almost all machine learning algorithms we discussed in class, they are: logistic regression, decision tree, random forest, linear regression, support vector machine.

9. Limitation and fairness

Limitation: As we all know, the version of the game may be changed frequently. So does League of legends. League of legends always release a new patch at least once a month. For every patch, many champions and many items may be nerfed and some champions and items may be buffed. This change would influence the prediction for different patch. However, we only use the specific dataset from Kaggle. It is out of date so it could not predict the result accurately in the up-to-date patch.

Fairness: Different game server has different play style. Items in the different game styles may play a different important role. However, it is hard to define and describe the game style as the input of our model.

10. Reference

Kaggle data Analyst sample:
https://www.kaggle.com/laowingkin/lol-how-to-win-the-world-championship/notebook

Pandas dataframe: https://pandas.pydata.org/pandas-docs/stable/reference/frame.html

Seaborn violinplot: https://seaborn.pydata.org/generated/seaborn.violinplot.html

Seaborn heatmap: https://seaborn.pydata.org/generated/seaborn.heatmap.html

Scikit-learn package: https://scikit-learn.org/stable/

League of legends introduction:
https://na.leagueoflegends.com/en/game-info/get-started/what-is-lol/